NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
60625-2

Deuxième édition
Second edition
1993-12

Instruments de mesurage programmables – Système d'interface (bits parallèles, octets série) –

Partie 2:
Codes, formats, protocoles et instructions communes

Programmable measuring instruments – Interface system (byte serial, bit parallel) –

Part 2:
Codes, formats, protocols and common commands

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Numéro de référence
Reference number
CEI/IEC 60625-2: 1993

Numéros des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60000.

Publications consolidées

Les versions consolidées de certaines publications de la CEI incorporant les amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles dans le Catalogue de la CEI.

Les renseignements relatifs à des questions à l'étude et des travaux en cours entrepris par le comité technique qui a établi cette publication, ainsi que la liste des publications établies, se trouvent dans les documents ci-dessous:

- **«Site web» de la CEI***
- **Catalogue des publications de la CEI**
  Publié annuellement et mis à jour régulièrement
  (Catalogue en ligne)*
- **Bulletin de la CEI**
  Disponible à la fois au «site web» de la CEI* et comme périodique imprimé

Terminologie, symboles graphiques et littéraux

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 60050: *Vocabulaire Electrotechnique International* (VEI).

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera la CEI 60027: *Symboles littéraux à utiliser en électrotechnique,* la CEI 60417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles,* et la CEI 60617: *Symboles graphiques pour schémas.*

* Voir adresse «site web» sur la page de titre.

Numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series.

Consolidated publications

Consolidated versions of some IEC publications including amendments are available. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available in the IEC catalogue.

Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is to be found at the following IEC sources:

- **IEC web site***
- **Catalogue of IEC publications**
  Published yearly with regular updates
  (On-line catalogue)*
- **IEC Bulletin**
  Available both at the IEC web site* and as a printed periodical

Terminology, graphical and letter symbols

For general terminology, readers are referred to IEC 60050: *International Electrotechnical Vocabulary* (IEV).

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications IEC 60027: *Letter symbols to be used in electrical technology*, IEC 60417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets* and IEC 60617: *Graphical symbols for diagrams.*

* See web site address on title page.

IECNORM.COM : Click to view the full PDF of IEC 60134:1993
Withdrawn

NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
60625-2

Deuxième édition
Second edition
1993-12

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Instruments de mesurage programmables – Système d'interface (bits parallèles, octets série) –

Partie 2:
Codes, formats, protocoles et instructions communes

Programmable measuring instruments – Interface system (byte serial, bit parallel) –

Part 2:
Codes, formats, protocols and common commands

© IEC 1993 Droits de reproduction réservés — Copyright - all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission
Telefax: +41 22 919 0300 e-mail: inmail@iec.ch
3, rue de Varembé Geneva, Switzerland
IEC web site http: //www.iec.ch

IEC
Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

SOMMAIRE

Pages

AVANT-PROPOS 20

INTRODUCTION 24

SECTION 1: CONCEPTS GÉNÉRAUX

Articles

1 Généralités 26
1.1 Domaine d'application 26
1.2 Objectifs 28
1.3 Notation 28
2 Références normatives 28
3 Considérations sur le système 30
3.1 Définitions 30
3.2 Trafic de messages du système 30
3.3 Couches fonctionnelles 34

SECTION 2: SPÉCIFICATIONS RELATIVES À L'APPAREIL

4 Critères de conformité de l'appareil 36
4.1 Prescriptions de la CEI 625-1 38
4.2 Prescriptions d'échange de messages 38
4.3 Prescriptions de syntaxe 40
4.4 Prescriptions d'indication de l'état 42
4.5 Instructions communes 46
4.6 Prescriptions de synchronisation 52
4.7 Capacités de configuration du système 54
4.8 Capacités des contrôleurs 54
4.9 Prescriptions concernant la documentation sur l'appareil 54
5 Prescriptions de fonction d'interface de l'appareil 58
5.1 Prescriptions d'établissement d'une liaison 58
5.2 Prescriptions d'adresses 58
5.3 Prescriptions parleur 60
5.4 Prescriptions écouteur 60
5.5 Prescriptions de demande de service 60
5.6 Prescriptions de mode de fonctionnement à distance ou local 62
5.7 Prescriptions concernant l'interrogation parallèle 68
5.8 Prescriptions d'appareils remis à zéro 68
5.9 Prescriptions de déclenchement de l'appareil 70
5.10 Prescriptions de fonction du contrôleur 70
5.11 Prescriptions électriques 70
5.12 Exigences de mise sous tension 70

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

CONTENTS

Page

FOREWORD 21

INTRODUCTION 25

SECTION 1: GENERAL CONCEPTS

Clause

1 General 27
1.1 Scope 27
1.2 Objectives 29
1.3 Notation 29
2 Normative references 29
3 System considerations 31
3.1 Definitions 31
3.2 System message traffic 31
3.3 Functional layers 35

SECTION 2: DEVICE RELATED SPECIFICATIONS

4 Device compliance criteria 37
4.1 IEC 625-1 requirements 39
4.2 Message exchange requirements 39
4.3 Syntax requirements 41
4.4 Status reporting requirements 43
4.5 Common commands 47
4.6 Synchronization requirements 53
4.7 System configuration capability 55
4.8 Controller capability 55
4.9 Device documentation requirements 55
5 Device interface function requirements 59
5.1 Handshake requirements 59
5.2 Address requirements 59
5.3 Talker requirements 61
5.4 Listener requirements 61
5.5 Service Request requirements 61
5.6 Remote/Local requirements 63
5.7 Parallel Poll requirements 69
5.8 Device Clear requirements 69
5.9 Device Trigger requirements 71
5.10 Controller function requirements 71
5.11 Electrical requirements 71
5.12 Power-on requirements 71

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Articles Pages

6 Protocole de contrôle des échanges de messages 72
6.1 Eléments fonctionnels 74
6.1.1 Bus CEI 625-1 74
6.1.2 Indication d'état 74
6.1.3 Interface d'échange de messages 74
6.1.4 Contrôle E/S 78
6.1.5 Tampon d'entrée 86
6.1.6 Analyseur syntaxique 90
6.1.7 Contrôle d'exécution 94
6.1.8 Fonctions de l'appareil 98
6.1.9 Formateur de réponse 100
6.1.10 File d'attente de sortie 100
6.1.11 Contrôle de déclenchement 102
6.2 Vue d'ensemble du protocole 104
6.2.1 Initialisation 106
6.2.2 Traitement des instructions 108
6.2.3 Traitement des interrogations 108
6.3 Fonctionnement du contrôle d'échanges de messages 108
6.3.1 Etats de contrôle d'échange de messages 108
6.3.2 Actions de transition de contrôle d'échange de messages 120
6.4 Règles de protocole 122
6.4.1 Transfert de messages de programmation 122
6.4.2 Indépendance de la source des messages 124
6.4.3 Séquence d'échange de messages 124
6.4.4 Interrogations complexes 126
6.4.5 Exigences concernant les instructions de messages 128
6.5 Exceptions au Protocole 130
6.5.1 Messages arrêtés prématurément 130
6.5.2 Adressé pour parler sans rien à dire 130
6.5.3 Pas d'écouteur sur le bus 132
6.5.4 Erreur d'instruction 132
6.5.5 Erreur d'exécution 132
6.5.6 Erreur spécifique à l'appareil 132
6.5.7 Erreur d'interrogation 132

7 Formats de l'appareil écouteur 136
7.1 Vue d'ensemble 136
7.2 Notation 140
7.3 Messages de programmation terminés – Syntaxe fonctionnelle 146
7.4 Eléments fonctionnels de séparation 156
7.4.1 <SÉPARATEUR D'UNITÉS DE MESSAGES DE PROGRAMMATION> 156
7.4.2 <SÉPARATEUR DE DONNÉES DE PROGRAMMATION> 158
7.4.3 <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> 160
7.5 <CARACTÈRE D'ARRÊT DE MESSAGES DE PROGRAMMATION> 160

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Clause Page

6 Message exchange control protocol 73
6.1 Functional elements 75
6.1.1 IEC 625-1 bus 75
6.1.2 Status reporting 75
6.1.3 Message Exchange Interface 75
6.1.4 I/O Control 79
6.1.5 Input Buffer 87
6.1.6 Parser 91
6.1.7 Execution Control 95
6.1.8 Device Functions 99
6.1.9 Response Formatter 101
6.1.10 Output Queue 101
6.1.11 Trigger Control 103
6.2 Protocol overview 105
6.2.1 Initialization 107
6.2.2 Command processing 109
6.2.3 Query processing 109
6.3 Message Exchange Control operation 109
6.3.1 Message Exchange Control states 109
6.3.2 Message Exchange Control transition actions 121
6.4 Protocol rules 123
6.4.1 Program message transfer 123
6.4.2 Message source independence 125
6.4.3 Message exchange sequence 125
6.4.4 Compound queries 127
6.4.5 Message order requirements 129
6.5 Protocol exceptions 131
6.5.1 Aborted messages 131
6.5.2 Addressed to talk with nothing to say 131
6.5.3 No listener on bus 133
6.5.4 Command Error 133
6.5.5 Execution Error 133
6.5.6 Device-specific Error 133
6.5.7 Query Error 133

7 Device listening formats 137
7.1 Overview 137
7.2 Notation 141
7.3 Terminated program messages – Functional syntax 147
7.4 Separator functional elements 157
7.4.1 <PROGRAM MESSAGE UNIT SEPARATOR> 157
7.4.2 <PROGRAM DATA SEPARATOR> 159
7.4.3 <PROGRAM HEADER SEPARATOR> 161
7.5 <PROGRAM MESSAGE TERMINATOR> 161

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Articles Pages

7.6 Eléments fonctionnels d'en-tête de programmation 162
7.6.1 <EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION> 162
7.6.2 <EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION> 170
7.7 Eléments fonctionnels de <DONNÉES DE PROGRAMMATION> 174
7.7.1 <DONNÉES DE PROGRAMMATION DE CARACTÈRES> 174
7.7.2 <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> 176
7.7.3 <DONNÉES DE PROGRAMMATION DE SUFFIXES> 180
7.7.4 <DONNÉES DE PROGRAMMATION NUMÉRIQUES NON DÉCIMALES> 188
7.7.5 <DONNÉES DE PROGRAMMATION DE CHAÎNES> 194
7.7.6 <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> 196
7.7.7 <DONNÉES DE PROGRAMMATION D'EXPRESSION> 200

8 Formats de l'appareil parleur 202
8.1 Vue d'ensemble 202
8.2 Notation 202
8.3 Messages de réponse terminés – Syntaxe fonctionnelle 202
8.4 Eléments fonctionnels de séparation 212
8.4.1 <SÉPARATEUR D'UNITÉS DE MESSAGES DE RÉPONSE> 212
8.4.2 <SÉPARATEUR DE DONNÉES DE RÉPONSE> 212
8.4.3 <SÉPARATEUR D'EN-TÊTES DE RÉPONSE> 214
8.5 <CARACTÈRE D'ARRÊT DE MESSAGES DE RÉPONSE> 214
8.6 <EN-TÊTE DE RÉPONSE> 214
8.7 Eléments fonctionnels de <DONNÉES DE RÉPONSE> 218
8.7.1 <DONNÉES DE RÉPONSE DE CARACTÈRES> 218
8.7.2 <DONNÉES DE RÉPONSES NUMÉRIQUES NR1> 220
8.7.3 <DONNÉES DE RÉPONSE NUMÉRIQUES NR2> 222
8.7.4 <DONNÉES DE RÉPONSE NUMÉRIQUES NR3> 222
8.7.5 <DONNÉES DE RÉPONSE NUMÉRIQUES HEXADÉCIMALES> 224
8.7.6 <DONNÉES DE RÉPONSE NUMÉRIQUES OCTALES> 226
8.7.7 <DONNÉES DE RÉPONSE NUMÉRIQUES BINAIRES> 228
8.7.8 <DONNÉES DE RÉPONSE DE CHAÎNES> 230
8.7.9 <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE> 232
8.7.10 <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE> 234
8.7.11 <DONNÉES DE RÉPONSE DE 7 BITS ISO ARBITRAIRE> 236

9 Codage des données de message 238
9.1 Codes ISO de 7 bits 238
9.2 Codes de nombre entier 8 bits binaire 238
9.3 Code en virgule flottante binaire 244

10 Instructions et interrogations communes 252
10.1 *AAD, Instruction d'acceptation d'adresse 256
10.2 *CAL?, Interrogation d'étalonnage 256

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Clause Page

7.6 Program Header functional elements 163
7.6.1 <COMMAND PROGRAM HEADER> 163
7.6.2 <QUERY PROGRAM HEADER> 171
7.7 <PROGRAM DATA> functional elements 175
7.7.1 <CHARACTER PROGRAM DATA> 175
7.7.2 <DECIMAL NUMERIC PROGRAM DATA> 177
7.7.3 <SUFFIX PROGRAM DATA> 181
7.7.4 <NON-DECIMAL NUMERIC PROGRAM DATA> 189
7.7.5 <STRING PROGRAM DATA> 195
7.7.6 <ARBITRARY BLOCK PROGRAM DATA> 197
7.7.7 <EXPRESSION PROGRAM DATA> 201

8 Device talking formats 203
8.1 Overview 203
8.2 Notation 203
8.3 Terminated response messages – Functional syntax 203
8.4 Separator functional elements 213
8.4.1 <RESPONSE MESSAGE UNIT SEPARATOR> 213
8.4.2 <RESPONSE DATA SEPARATOR> 213
8.4.3 <RESPONSE HEADER SEPARATOR> 215
8.5 <RESPONSE MESSAGE TERMINATOR> 215
8.6 <RESPONSE HEADER> 215
8.7 <RESPONSE DATA> functional elements 219
8.7.1 <CHARACTER RESPONSE DATA> 219
8.7.2 <NR1 NUMERIC RESPONSE DATA> 221
8.7.3 <NR2 NUMERIC RESPONSE DATA> 223
8.7.4 <NR3 NUMERIC RESPONSE DATA> 223
8.7.5 <HEXADECIMAL NUMERIC RESPONSE DATA> 225
8.7.6 <OCTAL NUMERIC RESPONSE DATA> 227
8.7.7 <BINARY NUMERIC RESPONSE DATA> 229
8.7.8 <STRING RESPONSE DATA> 231
8.7.9 <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> 233
8.7.10 <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> 235
8.7.11 <ARBITRARY ISO 7-bit RESPONSE DATA> 237

9 Message data coding 239
9.1 ISO 7-bit codes 239
9.2 Binary 8-bit integer codes 239
9.3 Binary floating point code 245

10 Common commands and queries 253
10.1 *AAD, Accept Address command 257
10.2 *CAL?, Calibration query 257

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Articles Pages

10.3 *CLS, Instruction d'effacement d'état 258
10.4 *DDT, Instruction de déclenchement défini par l'appareil 260
10.5 *DDT?, Interrogation de déclenchement défini par l'appareil 262
10.6 *DLF, Instruction de fonction écouteur invalide 264
10.7 *DMC, Définition d'une macro-instruction 264
10.8 *EMC, Validation d'une macro-instruction 270
10.9 *EMC?, Validation d'une macro-instruction 272
10.10 *ESE, Instruction de validation d'état d'événement standard 274
10.11 *ESE?, Interrogation de validation d'état d'événement standard 276
10.12 *ESR?, Interrogation de registre d'état d'événement standard 276
10.13 *GMC?, Lecture du contenu des macro-instructions 278
10.14 *IDN?, Interrogation d'identification 280
10.15 *IST?, Interrogation de lecture d'état individuel 282
10.16 *LMC?, Interrogation des macros définies 284
10.17 *LRN?, Interrogation de lecture de la configuration de l'appareil 286
10.18 *OPC, Instruction d'opération achevée 288
10.19 *OPC?, Interrogation d'opération achevée 288
10.20 *OPT?, Interrogation d'identification d'option 290
10.21 *PCB, Retour de passage de contrôle 292
10.22 *PMC, Elimination des macro-instructions 294
10.23 *PRE, Instruction de validation de registre d'interrogation parallèle 294
10.24 *PRE?, Interrogation de validation de registre d'interrogation parallèle 296
10.25 *PSC, Instruction d'effacement d'état à la mise sous tension 298
10.26 *PSC?, Interrogation d'effacement d'état à la mise sous tension 300
10.27 *PUD, Instruction de données utilisateur protégées 302
10.28 *PUD?, Interrogation de données utilisateur protégées 304
10.29 *RCL, Instruction de rappel 304
10.30 *RDT, Instruction de transfert de description de ressource 306
10.31 *RDT?, Interrogation de transfert de description de ressource 308
10.32 *RST, Instruction de remise à zéro 310
10.33 *SAV, Instruction de sauvegarde 312
10.34 *SRE, Instruction de validation de demande de service 314
10.35 *SRE?, Interrogation de validation de demande de service 314
10.36 *STB?, Interrogation de lecture de l'octet d'état 316
10.37 *TRG, Instruction de déclenchement 318
10.38 *TST?, Interrogation d'auto-test 318
10.39 *WAI, Instruction d'attente de poursuite 320

11 Indications sur l'état de l'appareil 322

11.1 Vue d'ensemble 322
11.1.1 Opération 322
11.1.2 Récapitulatif des instructions communes relatives 326
11.1.3 Opérations relatives définies dans la CEI 625-1 328

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Clause Page

10.3 *CLS, Clear Status command 259
10.4 *DDT, Define Device Trigger command 261
10.5 *DDT?, Define Device Trigger query 263
10.6 *DLF, Disable Listener Function command 265
10.7 *DMC, Define Macro command 265
10.8 *EMC, Enable Macro command 271
10.9 *EMC?, Enable Macro query 273
10.10 *ESE, Standard Event Status Enable command 275
10.11 *ESE?, Standard Event Status Enable query 277
10.12 *ESR?, Standard Event Status Register query 277
10.13 *GMC?, Get Macro Contents query 279
10.14 *IDN?, Identification query 281
10.15 *IST?, Individual Status query 283
10.16 *LMC?, Learn Macro query 285
10.17 *LRN?, Learn Device Setup query 287
10.18 *OPC, Operation Complete command 289
10.19 *OPC?, Operation Complete query 289
10.20 *OPT?, Option Identification query 291
10.21 *PCB, Pass Control back 293
10.22 *PMC, Purge Macros command 295
10.23 *PRE, Parallel Poll Enable Register Enable command 295
10.24 *PRE?, Parallel Poll Enable Register Enable query 297
10.25 *PSC, Power-on Status Clear command 299
10.26 *PSC?, Power-on Status Clear query 301
10.27 *PUD, Protected User Data command 303
10.28 *PUD?, Protected User Data query 305
10.29 *RCL, Recall command 305
10.30 *RDT, Resource Description Transfer command 307
10.31 *RDT?, Resource Description Transfer query 309
10.32 *RST, Reset command 311
10.33 *SAV, Save command 313
10.34 *SRE, Service Request Enable command 315
10.35 *SRE?, Service Request Enable query 315
10.36 *STB?, Read Status Byte query 317
10.37 *TRG, Trigger command 319
10.38 *TST?, Self-Test query 319
10.39 *WAI, Wait-to-Continue command 321

11 Device status reporting 323

11.1 Overview 323
11.1.1 Operation 323
11.1.2 Summary of related common commands 327
11.1.3 Related IEC 625-1 defined operations 329

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Articles Pages

11.2 Registre d'octets d'état 330
11.2.1 Définition 330
11.2.2 Lecture du registre d'octet d'état 332
11.2.3 Ecriture dans le registre d'octet d'état 336
11.2.4 Effacement du registre d'octet d'état 336
11.3 Validation de demande de service 336
11.3.1 Opération 336
11.3.2 Registre de validation de demande de service 336
11.3.3 Génération de demande de service 340
11.4 Structures de données d'état 352
11.4.1 Vue d'ensemble 352
11.4.2 Structure de données d'état – Modèle de registre 352
11.4.3 Structure de données d'état – Modèle de file d'attente 362
11.5 Structures de données d'état standard 366
11.5.1 Modèle de registre d'état d'événements standard 366
11.5.2 Modèle de file d'attente standard 378
11.6 Traitement des réponses à des interrogations parallèles 380
11.6.1 Registre de validation d'interrogation parallèle 382
11.6.2 Lecture de ist sans interrogation parallèle 382

SECTION 3: SPÉCIFICATIONS RELATIVES AU SYSTÈME

12 Techniques de synchronisation appareil/contrôleur 384
12.1 Vue d'ensemble 384
12.2 Instructions séquentielles et recouvrables 386
12.2.1 Illustration des instructions séquentielles 386
12.2.2 Illustration des instructions recouvrables 386
12.3 Signalisation "opération en cours" 388
12.4 Signalisation "pas d'opération en cours" 388
12.5 Instructions de synchronisation contrôleur/appareil 390
12.5.1 L'instruction commune *WAI 390
12.5.2 L'instruction commune *OPC 394
12.5.3 L'interrogation commune *OPC? 396
12.6 Synchronisation avec des signaux de commande extérieurs 400
12.7 Usage incorrect d'*OPC et d'*OPC? 400
12.8 Considérations de conception 402
12.8.1 Instructions recouvrables 402
12.8.2 Traitement de l'erreur d'exécution 408
12.8.3 Opération achevée 408

13 Configuration automatique du système 410
13.1 Introduction 410
13.2 Vue d'ensemble 410
13.3 Approche générique de la configuration automatique du système 320
13.3.1 Affectation d'adresse 420
13.3.2 Identification d'appareil 424

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Clause Page

11.2 Status Byte Register 331
11.2.1 Definition 331
11.2.2 Reading the Status Byte Register 333
11.2.3 Writing the Status Byte Register 337
11.2.4 Clearing the Status Byte Register 337
11.3 Service Request enabling 337
11.3.1 Operation 337
11.3.2 Service Request Enable register 337
11.3.3 Service Request generation 341
11.4 Status Data Structures 353
11.4.1 Overview 353
11.4.2 Status Data Structure – Register model 353
11.4.3 Status Data Structure – Queue model 363
11.5 Standard Status Data Structures 367
11.5.1 Standard Event Status Register model 367
11.5.2 Standard Queue model 379
11.6 Parallel Poll response handling 381
11.6.1 Parallel Poll Enable Register 383
11.6.2 Reading ist without a Parallel Poll 383

SECTION 3: SYSTEM-RELATED SPECIFICATIONS

12 Device/controller synchronization techniques 385
12.1 Overview 385
12.2 Sequential and Overlapped Commands 387
12.2.1 Illustration of Sequential Commands 387
12.2.2 Illustration of Overlapped Commands 387
12.3 Pending-Operation flag 389
12.4 No-Operation-Pending flag 389
12.5 Controller/device synchronization commands 391
12.5.1 The *WAI common command 391
12.5.2 The *OPC common command 395
12.5.3 The *OPC? common query 397
12.6 Synchronization with external control signals 401
12.7 Improper usage of *OPC and *OPC? 401
12.8 Design considerations 403
12.8.1 Overlapped commands 403
12.8.2 Execution Error handling 409
12.8.3 Operation complete 409

13 Automatic system configuration 411
13.1 Introduction 411
13.2 Overview 411
13.3 Generic approach to automatic system configuration 421
13.3.1 Address assignment 421
13.3.2 Device identification 425

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Articles Pages

13.4 Prescriptions détaillées des instructions de configuration automatique 426
13.4.1 Prescriptions d'instruction commune *DLF 426
13.4.2 Prescriptions d'instruction commune *AAD 428
13.5 Techniques de configuration automatique supplémentaires 440
13.6 Exemples 440

SECTION 4: SPÉCIFICATIONS RELATIVES AU CONTRÔLEUR

14 Critères de conformité du contrôleur 440
14.1 Prescriptions de la CEI 625-1 442
14.2 Prescriptions des échanges de messages 442
14.3 Protocoles 444
14.4 Traitement des éléments fonctionnels 446
14.5 Prescriptions de spécifications du contrôleur 446
15 Prescriptions du contrôleur CEI 625-2 448
15.1 Prescriptions de la fonction d'interface du contrôleur 448
15.1.1 Prescriptions du contrôleur 448
15.1.2 Prescriptions du parleur 448
15.1.3 Prescriptions de l'écouteur 448
15.1.4 Prescriptions de passage de contrôle 448
15.1.5 Prescriptions électriques 448
15.2 Prescriptions supplémentaires du contrôleur CEI 625-2 450
15.3 Recommandations du contrôleur CEI 625-2 450
15.3.1 Lignes de bus de surveillance 450
15.3.2 Temporisations 452
15.3.3 Interruptions SRQ 452
16 Protocoles d'échanges de messages du contrôleur 452
16.1 Définitions 452
16.2 Séquences de contrôle 456
16.2.1 ENVOI D'INSTRUCTION 458
16.2.2 ENVOI DE CONFIGURATION 458
16.2.3 ENVOI D'OCTETS DE DONNÉES 458
16.2.4 ENVOI 460
16.2.5 RÉCEPTION DE CONFIGURATION 462
16.2.6 RÉCEPTION DE MESSAGE DE RÉPONSE 462
16.2.7 RÉCEPTION 466
16.2.8 ENVOI DE IFC 466
16.2.9 APPAREIL REMIS À ZÉRO 466
16.2.10 VALIDATION DES CONTRÔLES LOCAUX 468
16.2.11 VALIDATION ÉLOIGNÉE 470
16.2.12 SÉLECTION DE RWLS 470
16.2.13 ENVOI DE LLO 472
16.2.14 PASSAGE DE CONTRÔLE 472
16.2.15 RÉALISER UNE INTERROGATION PARALLÈLE 472

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Clause Page

13.4 Detailed requirements of the auto configure commands 427
13.4.1 *DLF common command requirements 427
13.4.2 *AAD common command requirements 429
13.5 Additional automatic configuration techniques 441
13.6 Examples 441

SECTION 4: CONTROLLER RELATED SPECIFICATIONS

14 Controller compliance criteria 441
14.1 IEC 625-1 requirements 443
14.2 Message Exchange requirements 443
14.3 Protocols 445
14.4 Functional element handling 447
14.5 Controller specification requirements 447
15 IEC 625-2 controller requirements 449
15.1 Controller interface function requirements 449
15.1.1 Controller requirements 449
15.1.2 Talker requirements 449
15.1.3 Listener requirements 449
15.1.4 Passing control requirements 449
15.1.5 Electrical requirements 449
15.2 Additional IEC 625-2 controller requirements 451
15.3 IEC 625-2 controller recommendations 451
15.3.1 Monitoring bus lines 451
15.3.2 Timeouts 453
15.3.3 SRQ interrupts 453
16 Controller message exchange protocols 453
16.1 Definitions 453
16.2 Control sequences 457
16.2.1 SEND COMMAND 459
16.2.2 SEND SETUP 459
16.2.3 SEND DATA BYTES 459
16.2.4 SEND 461
16.2.5 RECEIVE SETUP 463
16.2.6 RECEIVE RESPONSE MESSAGE 463
16.2.7 RECEIVE 467
16.2.8 SEND IFC 467
16.2.9 DEVICE CLEAR 467
16.2.10 ENABLE LOCAL CONTROLS 469
16.2.11 ENABLE REMOTE 471
16.2.12 SET RWLS 471
16.2.13 SEND LLO 473
16.2.14 PASS CONTROL 473
16.2.15 PERFORM PARALLEL POLL 473

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Articles Pages

16.2.16 CONFIGURATION D'INTERROGATION PARALLÈLE 474
16.2.17 NON-CONFIGURATION D'INTERROGATION PARALLÈLE 474
16.2.18 LECTURE DE L'OCTET D'ÉTAT 476
16.2.19 DÉCLENCHEMENT 476

17 Protocoles communs du contrôleur 480

17.1 Protocole de remise à zéro 480
17.2 Protocole de recherche d'appareil demandant un service 484
17.3 Protocole d'interrogation série de tous les appareils 488
17.4 Protocole de passage de contrôle 490
17.5 Demande de contrôle 494
17.6 Protocole de recherche des appareils écouteurs 496
17.7 Protocole d'assignation d'adresses 498
17.8 Protocole d'auto-test du système 510

Annexes

A Utilisation et exemples d'en-têtes complexes 514
B Techniques de synchronisation appareil/contrôleur 522
C Exemple de configuration automatique de système 552

Figures

3-1 Trafic de messages habituels 32
3-2 Couches de protocoles fonctionnels CEI 625-1 et CEI 625-2 36

4-1 Capacités exigées d'indication d'état 44

6-1 Etat de l'appareil et vue d'ensemble des échanges de messages 76
6-2 Interface de contrôle d'échanges de messages, blocs fonctionnels 78
6-3 Schéma d'état de contrôle d'échanges de messages (simplifié) 106
6-4 Schéma d'état de contrôle d'échanges de messages (complet) 112

7-1 Syntaxe d'élément fonctionnel de <MESSAGE DE PROGRAMMATION TERMINÉ> 146
7-2 Syntaxe d'élément fonctionnel de <MESSAGE DE PROGRAMMATION> 148
7-3 Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE DE PROGRAMMATION> 148
7-4 Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE D'INSTRUCTION> 148
7-5 Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE D'INTERROGATION> 150
7-6 Syntaxe d'élément fonctionnel de <DONNÉES DE PROGRAMMATION> 152

8-1 Syntaxe d'élément fonctionnel de <MESSAGES DE RÉPONSE TERMINÉS> 204
8-2 Syntaxe d'élément fonctionnel de <MESSAGES DE RÉPONSE> 204
8-3 Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE DE RÉPONSE> 206
8-4 Syntaxe d'élément fonctionnel de <DONNÉES DE RÉPONSE> 208

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Clause Page

16.2.16 PARALLEL POLL CONFIGURE 475
16.2.17 PARALLEL POLL UNCONFIGURE 475
16.2.18 READ STATUS BYTE 477
16.2.19 TRIGGER 477

17 Common controller protocols 481

17.1 Reset protocol 481
17.2 Find device requesting service protocol 485
17.3 Serial Poll All Devices protocol 489
17.4 Pass control protocol 491
17.5 Requesting control 495
17.6 Find Listeners protocol 497
17.7 Set Address protocol 499
17.8 Test System protocol 511

Annexes

A Compound headers usage and examples 515
B Device/controller synchronization techniques 523
C Automatic system configuration example 553

Figures

3-1 Usual message traffic 33
3-2 IEC 625-1 and IEC 625-2 function protocol layers 37

4-1 Required status reporting capabilities 45

6-1 Device Status and Message Exchange overview 77
6-2 Message Exchange Control Interface functional blocks 79
6-3 Message Exchange Control State Diagram (simplified) 107
6-4 Message Exchange Control State Diagram (complete) 113

7-1 <TERMINATED PROGRAM MESSAGE> functional element syntax 147

7-2 <PROGRAM MESSAGE> functional element syntax 149
7-3 <PROGRAM MESSAGE UNIT> functional element syntax 149

7-4 <COMMAND MESSAGE UNIT> functional element syntax 149
7-5 <QUERY MESSAGE UNIT> functional element syntax 151
7-6 <PROGRAM DATA> functional element syntax 153

8-1 <TERMINATED RESPONSE MESSAGE> functional element syntax 205
8-2 <RESPONSE MESSAGE> functional element syntax 205
8-3 <RESPONSE MESSAGE UNIT> functional element syntax 207
8-4 <RESPONSE DATA> functional element syntax 209

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Figures Pages

11-1 Vue d'ensemble de la structure d'indication d'état CEI 625-2 326
11-2 Registre d'octet d'état 330
11-3 Validation de demande de service 338
11-4 Génération de demande de service 344
11-5 Schéma d'état rsv d'ensemble 346
11-6 Structure de données d'état – Modèle de registre 354
11-7 Structure de données d'état – Modèle de file d'attente 364
11-8 Vue d'ensemble des structures de données d'état standard 368
11-9 Modèle de registre d'état d'événements standard 370
11-10 Modèle de file d'attente standard 380
11-11 Structure de données de traitement de réponse à une interrogation parallèle 384

12-1 Schéma d'état d'opération en cours des instructions recouvrables sélectionnées 390
12-2 Diagramme des temps – Echantillon de message de programmation n'utilisant pas l'instruction *WAI 392
12-3 Diagramme des temps – Echantillon de message de programmation utilisant l'instruction *WAI 392
12-4 Schéma d'état d'instruction "opération achevée" 394
12-5 Diagramme des temps – Echantillon de message de programmation utilisant l'instruction *OPC 396
12-6 Schéma d'état d'interrogation "opération achevée" 398
12-7 Diagramme des temps – Opération incorrecte, exemple 1 404
12-8 Diagramme des temps – Opération incorrecte, exemple 2 406

13-1 Recherche poussée des appareils 412
13-2 Recherche d'appareils d'adresses non configurables 412
13-3 Recherche d'appareils d'adresses configurables 414
13-4 Protocole de recherche d'écouteurs 414
13-5 Protocole de sélection d'adresses 416
13-6 Instruction commune d'acceptation d'adresse 418

16-1 Modèle de contrôleur 452

A-1 Exemple d'organisation d'en-tête complexe utilisant un arbre 514
A-2 Organisation d'en-tête complexe utilisant un graphe, exemple 1 520
A-3 Organisation d'en-tête complexe utilisant un graphe, exemple 2 520

B-1 Exemple de topologie du système pour appareil répondeur et stimulateur 548

C-1 Circulation générale du protocole 552
C-2 Recherches d'identificateurs 554
C-3 Eléments de recherche de caractères 556
C-4 Corps de recherche 558

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figures Page

11-1 IEC 625-2 status reporting structure overview 327
11-2 Status Byte Register 331
11-3 Service Request enabling 339
11-4 Service Request generation 345
11-5 Set rsv State Diagram 347
11-6 Status Data Structure – Register model 355
11-7 Status Data Structure – Queue model 365
11-8 Standard Status Data Structures overview 369
11-9 Standard Event Status Register model 371
11-10 Standard Queue model 381
11-11 Parallel Poll response handling data structure 385

12-1 Pending operation state diagram of selected overlapped commands 391

12-2 Timing diagram – Sample program message not using *WAI command 393

12-3 Timing diagram – Sample program message using *WAI command 393

12-4 Operation Complete command state diagram 395
12-5 Timing diagram – Sample program message using the *OPC command 397

12-6 Operation Complete Query state diagram 399
12-7 Timing Diagram – Improper operation, example 1 405
12-8 Timing Diagram – Improper operation, example 2 407

13-1 Comprehensive Device Search 413
13-2 Non-Address-Configurable Device Search 413
13-3 Address-Configurable Device Search 415
13-4 Find Listeners protocol 415
13-5 Set Address protocol 417
13-6 Accept Address common command 419

16-1 Controller model 453

A-1 Compound header organization example using a tree 515
A-2 Compound header organization using a graph, example 1 521
A-3 Compound header organization using a graph, example 2 521

B-1 System topology for Response- and Stimulus-device example 549

C-1 Overall protocol flow 553
C-2 Identifier searches 555
C-3 Character search elements 557
C-4 Search body 559

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figures Pages

C-5 Action de recherche de caractère de fabricant et fin-de-zone, <eof> 560

C-6 Détection de la version élaborée de la zone BB 562

Tableaux

4-1 Prescriptions d'interfaces CEI 625-1 pour les appareils 38
4-2 Eléments fonctionnels nécessaires pour les appareils 40
4-3 Eléments fonctionnels facultatifs pour les appareils 42
4-4 Instructions communes d'indication d'état exigées 42
4-5 Instructions communes d'alimentation en circuit facultatives 46
4-6 Instructions communes d'interrogation parallèle facultatives 46
4-7 Instructions communes d'opérations internes exigées 46
4-8 Instructions communes facultatives de description des ressources 48
4-9 Instructions facultatives protégées de données de l'utilisateur 48
4-10 Instruction d'étalonnage facultative 48
4-11 Instruction de déclenchement facultative 50
4-12 Macro-instructions de déclenchement facultatives 50
4-13 Macro-instructions facultatives 50
4-14 Instruction facultative d'identification d'option 52
4-15 Instructions de positionnements mémorisés facultatives 52
4-16 Instruction d'apprentissage facultative 52
4-17 Instructions de synchronisation exigées 52
4-18 Instructions de configuration du système facultatives 54
4-19 Instruction de passage de contrôle facultative 54

7-1 Eléments d'<unités de suffixes> 184
7-2 Mnémoniques <multiplicateurs de suffixes> autorisés 186

9-1 Code ISO 7 bits 238
9-2 Tableau du code ISO 7 bits 242

10-1 En-têtes d'instructions communes CEI 625-2 252
10-2 Groupes d'instructions communes CEI 625-2 254

11-1 Instructions communes d'indication d'état 328
11-2 Indication d'état d'opérations définies dans la CEI 625-1 330
11-3 Schéma d'état rsv d'ensemble – Mnémoniques 346
11-4 Schéma d'état rsv d'ensemble – Sortie de messages 346

14-1 Prescriptions CEI 625-1 442
14-2 Séquences de contrôle exigées 444
14-3 Séquences de contrôle facultatives 444
14-4 Protocoles exigés 446
14-5 Protocoles facultatifs 446

17-1 Protocoles communs du contrôleur CEI 625-2 480
C-1 Identificateurs de l'appareil/Adresses affectées 552

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figures Page

C-5 Character search action for a manufacturer character and an end-of-field <eof> 561
C-6 Detection of the superset field BB 563

Tables

4-1 IEC 625-1 interface requirements for devices 39
4-2 Required functional elements for devices 41
4-3 Optional functional elements for devices 43
4-4 Required status reporting common commands 43
4-5 Optional power-on common commands 47
4-6 Optional Parallel Poll common commands 47
4-7 Required internal operation common commands 47
4-8 Optional resource description common commands 49
4-9 Optional protected user data commands 49
4-10 Optional calibration command 49
4-11 Optional trigger command 51
4-12 Optional trigger macro commands 51
4-13 Optional macro commands 51
4-14 Optional option identification command 53
4-15 Optional stored setting commands 53
4-16 Optional learn command 53
4-17 Required synchronization commands 53
4-18 Optional system configuration commands 55
4-19 Optional passing control command 55

7-1 <suffix unit> elements 185
7-2 Allowed <suffix mult.> mnemonics 187

9-1 ISO 7-bit code line relationship 239
9-2 ISO 7-bit code chart 243

10-1 IEC 625-2 common command headers 253
10-2 IEC 625-2 common command groups 255

11-1 Status reporting common commands 329
11-2 Status reporting IEC 625-1 defined operations 331
11-3 Set rsv State Diagram mnemonics 347
11-4 Set rsv State Diagram message output 347

14-1 IEC 625-1 requirements 443
14-2 Required control sequences 445
14-3 Optional control sequences 445
14-4 Required protocols 447
14-5 Optional protocols 447

17-1 IEC 625-2 common controller protocols 481
C-1 Device identifiers/Assigned addresses 553

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

INSTRUMENTS DE MESURAGE PROGRAMMABLES – SYSTÈME D'INTERFACE (BITS PARALLÈLES, OCTETS SÉRIE) –

Partie 2: Codes, formats, protocoles et instructions communes

AVANT-PROPOS

1) La CEI (Commission Electrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI, entre autres activités, publie des Normes internationales. Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par les comités d'études où sont représentés tous les Comités nationaux s'intéressant à ces questions, expriment dans la plus grande mesure possible un accord international sur les sujets examinés.
3) Ces décisions constituent des recommandations internationales publiées sous forme de normes, de rapports techniques ou de guides et agréées comme telles par les Comités nationaux.
4) Dans le but d'encourager l'unification internationale, les Comités nationaux de la CEI s'engagent à appliquer de façon transparente, dans toute la mesure possible, les Normes internationales de la CEI dans leurs normes nationales et régionales. Toute divergence entre la norme de la CEI et la norme nationale ou régionale correspondante doit être indiquée en termes clairs dans cette dernière.
5) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand un matériel est déclaré conforme à l'une de ses normes.

La Norme internationale CEI 625-2 a été établie par le sous-comité 65C: Communications numériques, du comité d'études 65 de la CEI: Mesure et commande dans les processus industriels.

Cette deuxième édition annule et remplace la première édition parue en 1980 et constitue une révision technique.

Le texte de cette norme est issu des documents suivants:

Règle des Six Mois	Rapport de vote
65C(BC)26	65C(BC)28

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PROGRAMMABLE MEASURING INSTRUMENTS – INTERFACE SYSTEM (BYTE SERIAL, BIT PARALLEL) –

Part 2: Codes, formats, protocols and common commands

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international cooperation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of the IEC on technical matters, prepared by technical committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.
3) They have the form of recommendations for international use published in the form of standards, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.

International Standard IEC 625-2 has been prepared by sub-committee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

This second edition cancels and replaces the first edition published in 1980 and constitutes a technical revision.

The text of this standard is based on the following documents:

Six Months' Rule	Report on voting
65C(CO)26	65C(CO)28

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

La CEI 625 comprend les parties suivantes, présentées sous le titre général *Instruments de mesurage programmables – Système d'interface (bits parallèles, octets série)*:

– Partie 1: 1993, Spécifications fonctionnelles, électriques, et mécaniques, applications du système et règles pour le constructeur et l'utilisateur

– Partie 2: 1993, Codes, formats, protocoles et instructions communes

Les annexes A, B et C sont données uniquement à titre d'information.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

IEC 625 consists of the following parts, under the general title *Programmable measuring instruments – Interface system (byte serial, bit parallel):*

- Part 1: 1993, Functional, electrical and mechanical specifications, system applications and requirements for the designer and user
- Part 2: 1993, Codes, formats, protocols and common commands.

Annexes A, B and C are for information only.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

INTRODUCTION

La première édition de la CEI 625-1 existe depuis plus de dix ans, période au cours de laquelle sa puissance et sa polyvalence ont fait leurs preuves de manière décisive. Sa contribution la plus importante à l'automatisation des systèmes d'essais a probablement été la normalisation de l'interconnexion des systèmes et de la communication. Toutefois, l'importance suivant laquelle cette normalisation s'est développée a été limitée par l'impossibilité de bien distinguer les besoins du marché. Sans expérience pour guider le processus de prise de décision, des efforts prématurés auraient été préjudiciables à l'acceptation de la norme.

Maintenant, le marché a mûri et de nombreux fabricants, intégrateurs de systèmes et utilisateurs ont connu les inconvénients de cette normalisation limitée. Heureusement, l'impact compensateur d'une grande base croissante d'expérience, d'étude et d'application est disponible pour poursuivre la normalisation du bus CEI 625-1. La présente partie est basée sur cette connaissance et favorisera un nouveau niveau de croissance et d'acceptation de la CEI 625-1.

Les concepts présentés dans la première partie viennent de plusieurs sources y compris de la première édition de la CEI 625-2. Depuis l'introduction de la CEI 625-1, les fabricants individuels d'appareils ont mis au point des normes internes qui adressent le code, les formats, le protocole, la syntaxe et les concepts sémantiques. Les intégrateurs de systèmes ont identifié les conditions requises utiles dans la configuration des systèmes. Toutes ces sources ont été utilisées et affinées dans la réalisation de la présente norme.

La normalisation supplémentaire recherchée par la CEI 625-2 était fondée sur une condition préalable: l'investissement existant en ressources de la CEI 625 doit être protégé. Ainsi, cette norme décrit une fonctionnalité qui complète et se base sur la CEI 625-1.

Au cours de la préparation de la présente édition CEI 625-2, des modifications mineures ont été apportées à quatre fonctions d'interface de la CEI 625-1: dialogue accepteur, demande d'intervention, disponibilité de l'appareil et commande locale/commande à distance. Il convient donc que les concepteurs, expérimentés ou non, revoient avec soin la CEI 625-1.

Tous les efforts ont été faits pour tirer profit des avantages de la normalisation sans limiter la liberté et la créativité du concepteur d'appareils.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

INTRODUCTION

The first edition of IEC 625-1 has been in existence for more than a decade during which time its power and versatility have been proven decisively. Probably its most important contribution to test system automation has been the standardization of system interconnection and communication. The extent, however, to which this standardization was developed was limited by the inability to clearly see the needs of the marketplace. Without experience to guide the decision-making process, early efforts would have been detrimental to the acceptance of the standard.

Now the marketplace has matured, and many manufacturers, system integrators, and users have experienced the negative side of this limited standardization. Fortunately, a counterbalancing force of a large and growing base of design and application experience is available to further standardize the IEC 625-1 bus. This part is based upon that knowledge and will encourage a new level of growth and acceptance of IEC 625-1.

The concepts presented in this part come from a variety of sources including the first edition of IEC 625-2. Since the introduction of IEC 625-1, individual manufacturers of devices have developed internal standards that address the code, formats, protocol, syntax, and semantic concepts. System integrators have identified requirements useful in configuring systems. All these sources have been used and refined in developing this standard.

The additional standardization sought by IEC 625-2 was founded on the premise that the existing investment in IEC 625-1 resources must be protected. Thus, this standard describes functionality that complements and is based upon IEC 625-1.

Minor changes were made to four interface functions in IEC 625-1 when writing this edition of IEC 625-2. The affected interface functions are: Acceptor Handshake, Service Request, Device Clear, and Remote/Local. New, as well as experienced, designers should carefully review IEC 625-1.

Every effort was made to reap the benefits of standardization without limiting the freedom and creativity of the device designer.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

INSTRUMENTS DE MESURAGE PROGRAMMABLES – SYSTÈME D'INTERFACE (BITS PARALLÈLES, OCTETS SÉRIE) –

Partie 2: Codes, formats, protocoles et instructions communes

SECTION 1: CONCEPTS GÉNÉRAUX

1 Généralités

1.1 *Domaine d'application*

La présente partie de la CEI 625 spécifie un ensemble de codes et de formats à utiliser par des dispositifs connectés par l'intermédiaire du bus CEI 625-1. Cette norme définit également les protocoles de communication nécessaires pour effectuer des échanges de messages indépendants de l'application et dépendants de l'appareil et définit en outre les instructions communes et caractéristiques utiles dans les applications de systèmes d'instruments.

La présente norme est destinée à s'appliquer directement aux systèmes d'instruments de petite à moyenne taille. Elle s'applique à des systèmes composés principalement d'appareils de mesure, de stimulation et d'interconnexion avec un contrôleur d'instrumentation. Cette norme peut s'appliquer également à certains appareils en dehors du domaine d'application dans l'environnement du système d'instruments.

Définissant également des messages variés dépendants de l'appareil, la présente norme étend et donne d'autres interprétations de certaines fonctions d'interface contenues dans la norme CEI 625-1, tout en restant compatibles avec cette norme.

La présente norme couvre les sujets suivants:

- Sous-ensembles de la norme CEI 625-1.
- Protocoles standards de traitement des messages y compris le traitement des erreurs.
- Structures syntaxiques non ambiguës pour les messages de programmation et de réponse.
- Instructions communes utiles sur une large gamme d'applications de systèmes d'instruments.
- Structures standards d'indication d'état.
- Configuration du système et protocoles de synchronisation.

L'utilisation de cette norme ne dégage pas l'utilisateur de sa responsabilité concernant la compatibilité du système au niveau application. L'utilisateur doit être familiarisé avec les caractéristiques de tous les composants du système afin de configurer un système optimal.

La présente norme est destinée à la fois aux concepteurs de contrôleurs et aux concepteurs d'appareils.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

PROGRAMMABLE MEASURING INSTRUMENTS – INTERFACE SYSTEM (BYTE SERIAL, BIT PARALLEL) –

Part 2: Codes, formats, protocols and common commands

SECTION 1: GENERAL CONCEPTS

1 General

1.1 *Scope*

This part of IEC 625 specifies a set of codes and formats to be used by devices connected via the IEC 625-1 bus. This standard also defines communication protocols necessary to effect application independent device-dependent message exchanges and further defines common commands and characteristics useful in instrument system applications.

This standard is intended to directly apply to small to medium-scale instrument systems. It applies to systems comprised mainly of measurement, stimulus, and interconnect devices with an instrumentation controller. The standard may also apply to certain devices outside the scope of the instrument system environment.

As well as defining a variety of device-dependent messages, this standard extends and further interprets certain interface functions contained in IEC 625-1 while remaining compatible with that standard.

This standard covers the following topics:

- Standard IEC 625-1 subsets.
- Standard message handling protocols including error handling.
- Unambiguous program and response message syntactic structures.
- Common commands useful in a wide range of instrument system applications.
- Standard status reporting structures.
- System configuration and synchronization protocols.

Use of this standard does not relieve the user from the burden of responsibility for system compatibility at the application level. The user must be familiar with the characteristics of all the system components in order to configure an optimum system.

The intended readers of this standard include both controller and device designers.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

1.2 *Objectifs*

Les objectifs de la présente norme sont les suivants:

1) Fournir une structure bien définie et non ambiguë de codes, formats, protocoles et instructions communes.

2) Conserver un caractère général pour s'adapter aux besoins d'une grande variété d'applications dans le domaine d'application de la présente norme.

3) Promouvoir le degré auquel des dispositifs provenant de différents fabricants peuvent être interconnectés et utilisés sans modification.

4) Permettre l'interconnexion des instruments et des appareils correspondants avec une capacité aussi bien limitée qu'étendue de créer, traiter et interpréter une variété de différents types de messages.

5) Définir les codes, formats, protocoles et instructions communes permettant de réduire les coûts de création de logiciels d'application et les coûts d'intégration de systèmes.

6) Permettre une communication directe parmi les appareils à systèmes d'instruments sans traduction et conversion extraordinaires de codes et de formats spéciaux.

1.3 *Notation*

La présente norme définit plusieurs mots communs ayant une signification particulière dans le contexte de cette norme. Ces mots ont différentes connotations dans la CEI 625-1, mais pour la lisibilité et la concision du texte, ce sont les mots qui sont utilisés de préférence dans la CEI 625-2. Pour éviter toute confusion, ces mots apparaissent toujours en caractères gras lorsqu'ils sont mentionnés dans le présent document. Les mots apparaissant en caractères gras sont: **système, appareil, contrôleur, bus de système**, et **interface du système**. Des messages locaux propres à la CEI 625-2 apparaîtront également en caractères gras minuscules.

Cette norme définit également les éléments syntaxiques utilisés pour décrire des messages transférés au bus. Les éléments syntaxiques sont mis entre crochets, par exemple <élément syntaxique>, pour les mettre en évidence par rapport aux messages locaux, aux messages éloignés, et au texte normal.

2 Références normatives

Les documents normatifs suivants contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente partie de la CEI 625. Au moment de la publication, les éditions indiquées étaient en vigueur. Tout document normatif est sujet à révision et les parties prenantes aux accords fondés sur la présente partie de la CEI 625 sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des documents normatifs indiqués ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur.

CEI 559: 1989, *Arithmétique binaire en virgule flottante pour systèmes à microprocesseurs*

CEI 625-1: 1993, *Instruments de mesurage programmables – Système d'interface (bits parallèles, octets série) – Partie 1: Spécifications fonctionnelles, électriques et mécaniques, application du système et règles pour le constructeur et l'utilisateur*

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

1.2 *Objectives*

The objectives of this standard are:

1) To provide a well-defined and unambiguous structure of codes, formats, protocols, and common commands.

2) To retain generality to accommodate the needs of a wide variety of applications within the scope of the standard.

3) To promote the degree to which devices from different manufacturers may be interconnected and used, without modification.

4) To enable the interconnection of instrumentation and related devices with both limited and extensive capability to generate, process and interpret a variety of different message types.

5) To define codes, formats, protocols, and common commands that will reduce the costs of generating application software and the costs of system integration.

6) To permit direct communication among instrument system devices without extraordinary translation and conversion of special codes and formats.

1.3 *Notation*

This standard defines several common English words to have special meaning in the context of this standard. These words have different connotations in IEC 625-1 but, for readability and conciseness of text, are the preferred words for use in IEC 625-2. To avoid confusion, these words always appear in bold type when referenced in this document. Words which appear in bold type are: **system, device, controller, system bus,** and **system interface**. Local messages peculiar to IEC 625-2 will also appear in lower case bold type.

This standard also defines syntactic elements used to describe messages transferred on the bus. Syntactic elements are enclosed by angle brackets, e.g. <syntactic element>, to set them off from local messages, remote messages, and normal text.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 625. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this part of IEC 625 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 559: 1989, *Binary floating-point arithmetic for microprocessor systems*

IEC 625-1: 1993, *Programmable measuring instruments – Interface system (byte serial, bit parallel) – Part 1: Functional, electrical and mechanical specifications, system applications and requirements for the designer and user*

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

ISO/IEC 646: 1991, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'informations*

ISO 2955: 1983, *Traitement de l'information – Représentation des unités du système international et d'autres unités dans des systèmes comprenant des jeux de caractères limités*

ISO 6093: 1985, *Traitement de l'information – Représentation des valeurs numériques dans les chaînes de caractères pour l'échange d'information*

3 Considérations sur le système

3.1 *Définitions*

Les définitions suivantes s'appliquent à la présente partie de la CEI 625. Cet article contient uniquement des définitions générales. Des définitions détaillées sont données dans les autres articles, comme il convient. Pour les définitions concernant spécifiquement la CEI 625-1, voir 1.3 de cette norme.

système: Groupe d'**appareils** et **contrôleur** interconnecté avec une **interface de système**.

appareil: Composant d'un **système** ne fonctionnant pas comme contrôleur de système, mais recevant typiquement des messages de programmation provenant du **contrôleur** et lui envoyant des réponses. Un **appareil** peut facultativement recevoir une commande du **contrôleur** et recevoir le contrôleur-en-charge du système*. Un **appareil** répond à toutes les exigences spécifiées dans l'article 4 de la présente norme.

contrôleur: Composant d'un **système** fonctionnant comme contrôleur de système. Un **contrôleur** envoie de manière typique des messages de programmation et reçoit des messages de réponses des **appareils**. Un **contrôleur** peut transmettre et recevoir une commande pour les protocoles de la présente norme. Un **contrôleur** répond à toutes les exigences spécifiées dans la Section 14 de la présente norme.

bus du système: Bus et protocoles CEI 625-1 interconnectant les **appareils** et les **contrôleurs** dans un **système**. Le contenu de cette norme s'applique au trafic dépendant de l'appareil passant par ce bus.

interface du système: Interface connectant un **appareil** ou un **contrôleur** au **bus du système**.

Une "interface de système non CEI 625-2" est toute interface autre que l'**interface du système** pouvant être connectée sur un **appareil** ou un **contrôleur**.

3.2 *Trafic de messages du système*

Cette norme est optimisée pour un **système** dans lequel les **appareils** ne deviennent pas contrôleur-en-charge. Le trafic de messages habituels est supposé allant du "contrôleur à l'appareil" ou de l'"appareil au contrôleur".

---

* Tous les composants CEI 625-2, c'est-à-dire les **appareils** et les **contrôleurs** dans un **système**, comprenant ce qui est défini comme un "appareil" dans la CEI 625-1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

ISO/IEC 646: 1991, *Information technology – ISO 7-bit coded character set for information interchange*

ISO 2955: 1983, *Information processing – Representation of SI and other units in systems with limited character sets*

ISO 6093: 1985, *Information processing – Representation of numerical values in character strings for information interchange*

3 System considerations

3.1 *Definitions*

The following definitions apply for the purpose of this part of IEC 625. This clause contains only general definitions. Detailed definitions are given in further clauses, as appropriate. For definitions relating specifically to IEC 625-1, see 1.3 of that standard.

system: A group of **devices** and a **controller** interconnected with a **system interface.**

device: A component of a **system** that does not function as the system-controller but typically receives program messages from, and sends response messages to, the **controller**. A **device** may optionally have the capability to receive control from the **controller** and become the controller-in-charge of the **system***. A **device** meets all the requirements stated in clause 4 of this standard.

controller: The component of a **system** that functions as the system controller. A **controller** typically sends program messages to and receives response messages from **devices**. A **controller** may pass and receive control per the protocols in this standard. A **controller** meets all the requirements stated in clause 14 of this standard.

system bus: The IEC 625-1 bus and protocols that interconnect the **devices** and **controllers** in a **system**. The content of this standard applies to device-dependent traffic over this bus.

system interface: An interface which connects a **device** or **controller** to the **system bus**.

A "non-IEC 625-2 system interface" is any interface other than the **system interface** which may happen to be connected to a **device** or **controller**.

3.2 *System message traffic*

This standard is optimized for a **system** in which the **devices** do not become controller-in-charge. The usual message traffic is assumed to be from "controller-to-device" or from "device-to-controller".

---

* All IEC 625-2 components, that is, **devices** and **controllers** in a **system** comprise what is defined in IEC 625-1 as "device".

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

La polyvalence de la syntaxe des messages de programmation (article 7) peut également permettre l'utilisation d'un composant CEI 625-2 dans un système contenant des composants non CEI 625-2.

Dans un système avec un **appareil** et **un contrôleur** non conforme, les formats écouteur polyvalents permettent l'obtention d'un degré de compatibilité plus élevé. Cette compatibilité est réalisée par des variations mineures de syntaxe que l'**appareil** doit accepter. Ces variations sont destinées à être des syntaxes faciles à générer sur une variété de **contrôleurs**. Elles peuvent également être la syntaxe par défaut du **contrôleur** non conforme. De tels systèmes sont en dehors du domaine d'application de la présente norme et peuvent ne pas exécuter les fonctions qui avaient été envisagées pour eux.

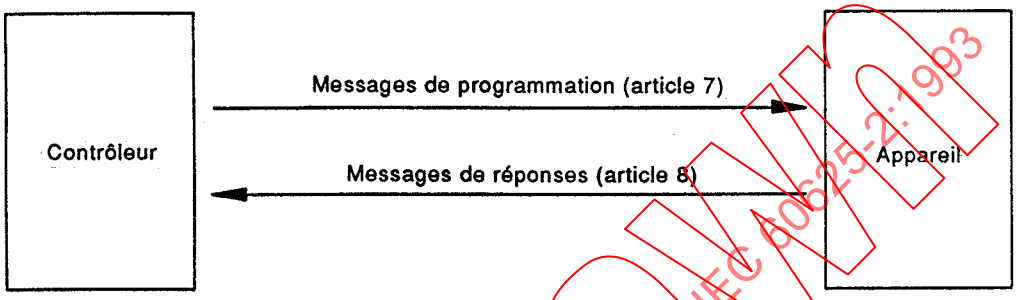


Figure 3-1 – Trafic de messages habituels

3.2.1 *Relations entre messages de programmation et messages de réponses*

Les articles qui suivent décrivent en détail la syntaxe et la sémantique autorisées pour un trafic de messages du système. La philosophie de cette norme est que les **appareils** reçoivent d'une manière plus polyvalente qu'ils n'émettent.

Ainsi, une gamme de variations de syntaxe définie par la présente norme doit être acceptée lorsque les **appareils** reçoivent. De même, une syntaxe précise, également identifiée par cette norme, est nécessaire lorsque les **appareils** émettent.

Ces messages de contrôleur à appareil (programmation) et d'appareil à contrôleur (réponse) sont composés d'éléments de syntaxe décrits aux articles 7 et 8, respectivement.

L'exemple qui suit démontre cette relation entre messages de programmation et messages de réponses pour un **appareil** qui a une gamme pouvant être programmée et questionnée. L'**appareil** a des gammes discrètes de 1.2, 12 et 120.

Voici un exemple de réception polyvalente de programmation de gammes; l'**appareil** reçoit.

Exemple de réception polyvalente

Le contrôleur envoie	*L'appareil interprète:*
GAMME 12.45	GAMME 12
ou GAMME 12	GAMME 12
ou GAMME 1.2E+1	GAMME 12

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The flexibility of the program message syntax (clause 7) also may allow the use of an IEC 625-2 component in a system which contains non-IEC 625-2 components.

In a system with a **device** and a non-compliant **controller**, the flexible listening formats allow the achievement of a higher degree of compatibility. This compatibility is realized by requiring minor variations in syntax to be accepted by the **device**. These variations are designed to be syntaxes that are easy to generate on a variety of **controllers**. They may even be the non-compliant **controller**'s default syntax. Such systems are beyond the scope of this standard and may not perform their intended functions.

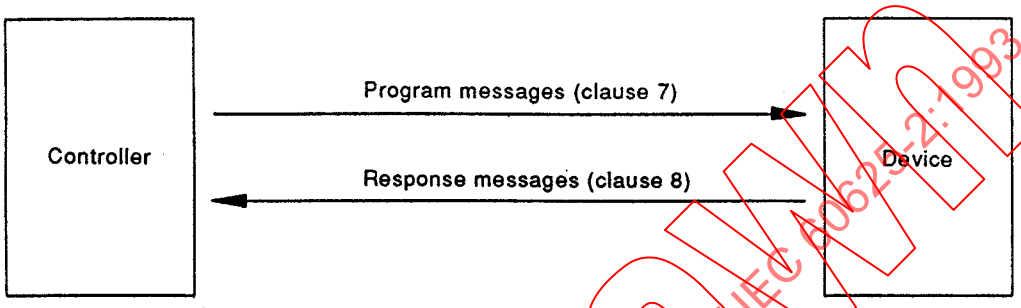


Figure 3-1 – Usual message traffic

3.2.1 *Program and response message relationships*

Subsequent clauses will describe in detail the allowed syntax and semantics for system message traffic. The philosophy of this standard is that **devices** receive in a more flexible manner than they send.

Thus, a range of variations of syntax defined by this standard must be accepted when **devices** listen. Likewise, a precise syntax, also identified by this standard, is required when **devices** talk.

These controller-to-device (program) and device-to-controller (response) messages are composed of syntactic elements described in clauses 7 and 8, respectively.

The following example demonstrates this relationship among program and response messages for a **device** that has a range that may be programmed and queried. The **device** has discrete ranges of 1.2, 12 and 120.

The following is an example of flexible reception of range programming with the **device** listening.

Flexible reception example

Controller sends	*Device interprets:*
RANGE 12.45	RANGE 12
or RANGE 12	RANGE 12
or RANGE 1.2E+1	RANGE 12

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Les règles de syntaxe de l'article 7, formats écouteur de l'appareil, assurent cette polyvalence d'interprétation.

Voici un exemple d'une réponse précise à la demande d'état des gammes; l'**appareil** émet.

Exemple d'une réponse précise

Contrôleur envoie	*L'appareil envoie*
GAMME?	12

Les règles de syntaxe de l'article 8, formats parleur de l'appareil, assurent une réponse de l'appareil avec un format précis.

3.2.2 *Autre trafic de messages*

En plus des messages de contrôleur à appareil et d'appareil à contrôleur, des protocoles sont définis pour faciliter le passage du contrôle, article 17.

Les protocoles d'appareil à appareil ne sont pas explicitement définis dans ce document. Toutefois, le trafic de messages entre les **appareils** suivra la syntaxe pour les <MESSAGES DE RÉPONSES> définis à l'article 8.

3.3 *Couches fonctionnelles*

Le but d'un système d'interface est de soutenir l'application du système de l'utilisateur. Dans le schéma dont cette norme fait partie, le système contient des "niveaux" successifs indépendants de protocole de communication comme indiqué à la figure 3-2.

Chaque couche représentée a des "messages" communs associés qui constituent des protocoles pour communiquer entre elle-même et la couche correspondante dans les éléments du système participant.

Les protocoles sont destinés à la communication entre des entités non-paires en raison des conditions imposées par la norme CEI 625-1. On n'a pas essayé d'associer la mise en couches des protocoles avec le modèle ISO d'interconnexion des systèmes ouverts.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The syntactic rules of clause 7, device listening formats, provide for this flexibility of interpretation.

The following is an example of precise response to range status request with the **device** talking.

Precise response example

Controller sends	*Device sends*
RANGE?	12

The syntactic rules of clause 8, Device talking formats, ensure the **device** responds with a precise format.

3.2.2 *Other message traffic*

In addition to controller-to-device and device-to-controller messages, protocols are defined to facilitate passing control, clause 17.

Device-to-device protocols are not explicitly defined in this document. However, message traffic between **devices** shall follow the syntax for <RESPONSE MESSAGES> defined in clause 8.

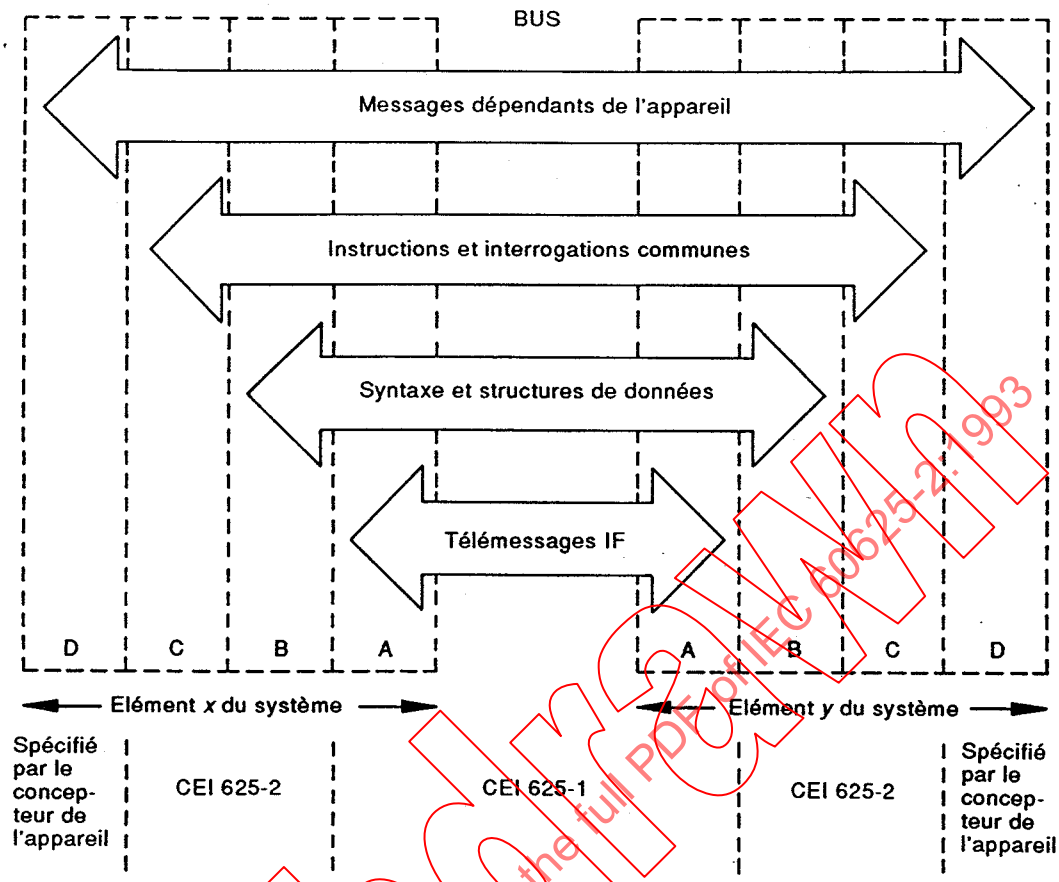
3.3 *Functional layers*

The purpose of an interface system is to support the system application of the user. In the schema of which this standard is a part, the system contains successive, independent "levels" of communication protocol as shown in figure 3-2.

Each layer shown has associated common "messages", which form protocols to communicate between itself and the corresponding layer in the participating system components.

The protocols are designed for communication between non-peer entities due to the requirements of the IEC 625-1 standard. No attempt has been made to associate the protocol layering with the ISO model for open systems interconnection.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



Les couches représentent les fonctions suivantes:

La couche D représente les fonctions de l'appareil

La couche C représente les fonctions du système commun

La couche B représente les fonctions de communication de messages

La couche A représente les fonctions d'interface (IF)

Où les couches D, C et B contiennent des messages dépendants de l'appareil CEI 625-1 et la couche A contient des messages d'interface CEI 625-1.

Figure 3-2 – Couches de protocoles fonctionnels CEI 625-1 et CEI 625-2

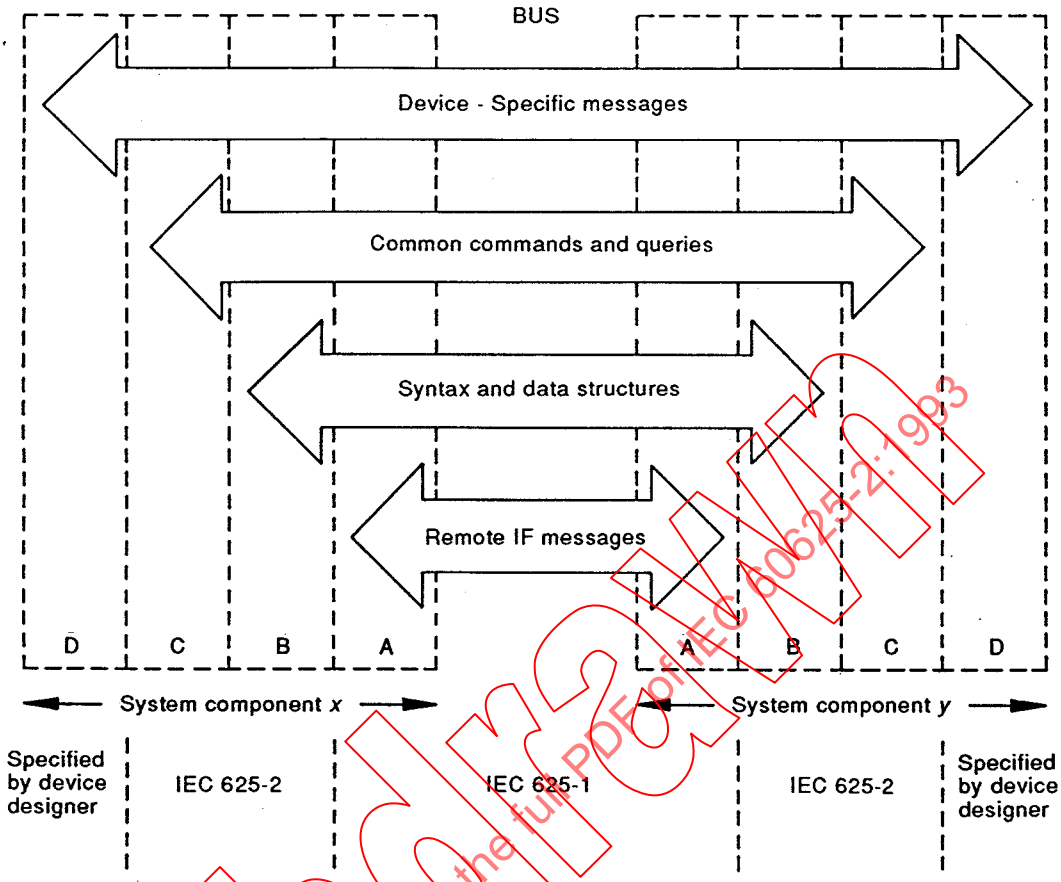
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

SECTION 2: SPÉCIFICATIONS RELATIVES À L'APPAREIL

4 Critères de conformité de l'appareil

Un **appareil** doit avoir certaines possibilités. Le présent article énumère les possibilités que cette norme exige sur un **appareil**. Un **appareil** peut facultativement comporter des possibilités supplémentaires. Toutes possibilités facultatives de l'**appareil** décrites dans la présente norme sont également énumérées dans cette section.

La conformité d'un **appareil** se subdivise en plusieurs domaines qui sont considérés séparément. Un **appareil** doit satisfaire à toute la fonctionnalité exigée dans chacun des domaines afin d'être conforme à la présente norme.



The layers have the following representations:

Layer D represents device functions

Layer C represents common system functions

Layer B represents message communication functions

Layer A represents interface functions (IF)

Where layers D, C and B contain IEC 625-1 device-dependent messages and layer A contains IEC 625-1 interface messages.

Figure 3-2 – IEC 625-1 and IEC 625-2 functional protocol layers

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

SECTION 2: DEVICE RELATED SPECIFICATIONS

4 Device compliance criteria

A **device** shall have certain capabilities. This clause lists the capabilities which this standard requires in a **device**. A **device** may optionally contain additional capabilities. Any optional **device** capabilities which are described by this standard are also listed in this section.

Compliance for a **device** is divided into several areas which are considered separately. A **device** shall satisfy all the required functionality in each of the areas in order to comply with this standard.

Cette section résume les conditions requises qui sont entièrement spécifiées dans l'article référencé. Le concepteur de l'appareil doit utiliser les prescriptions indiquées dans les articles ultérieurs lorsqu'il concevra effectivement un **appareil**.

4.1 *Prescriptions de la CEI 625-1*

Un **appareil** doit contenir les sous-ensembles CEI 625-1 énumérés dans le tableau 4-1 et pas d'autres.

Tableau 4-1 – Prescriptions d'interfaces CEI 625-1 pour les appareils

CEI 625-1 Fonction d'interface	CEI 625-1 Sous-ensembles	CEI 625-2 Paragraphes
Etablissement d'une liaison avec la source	SH1	5.1.1
Etablissement d'une liaison avec l'accepteur	AH1	5.1.2
Parleur	T5, T6, TE5, ou TE6	5.3
Ecouteur	L3, L4, LE3, ou LE4	5.4
Demande de service	SR1	5.5
Mode de fonctionnement à distance et local	RL0 ou RL1	5.6
Interrogation parallèle	PP0 ou PP1	5.7
Appareil remis à zéro	DC1	5.8
Déclenchement de l'appareil	DT0 ou DT1	5.9
Contrôleur	C0, ou C4 avec C5, C7, C9, ou C11	5.10
Interface électrique	E1 ou E2	5.11

Un **appareil** doit être conforme avec la CEI 625-1. Il doit également satisfaire à toutes les conditions requises à la section 5 de la présente norme.

4.2 *Prescriptions d'échange de messages*

Un **appareil** doit répondre à toutes les prescriptions énumérées à l'article 6 de cette norme.

Un tampon d'entrée de l'**appareil** peut prendre plusieurs formes. La longueur du tampon d'entrée peut être un nombre fixe d'octets. Il peut contenir un nombre fixe de <MESSAGES DE PROGRAMMATION> complets. Sa longueur en octets ou <MESSAGES DE PROGRAMMATION> peut varier avec l'état de l'**appareil**. Voir 6.1.5.

Certains messages d'interrogation peuvent générer le message de réponse effectif lorsque l'interrogation est reçue. D'autres messages d'interrogation peuvent générer la réponse effective lorsque le contrôleur lit la réponse. Voir 6.4.5.4.

L'exécution des éléments individuels analysables peut se faire au fur et à mesure de leur réception. L'**appareil** peut également attendre qu'un <SÉPARATEUR D'UNITÉS DE MESSAGES DE PROGRAMMATION> ou un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> soit analysé avant d'exécuter les éléments précédents d'<UNITÉ DE MESSAGE DE PROGRAMMATION>. Un **appareil** peut contenir un mélange de ces types d'instructions. Voir 6.4.5.1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

This clause summarizes requirements which are fully specified in the referenced clause. The device designer shall use the requirements stated in later clauses when actually designing a **device**.

4.1 *IEC 625-1 requirements*

A **device** shall contain these IEC 625-1 subsets listed in table 4-1 and no others.

Table 4-1 – IEC 625-1 interface requirements for devices

IEC 625-1 Interface function	IEC 625-1 Subsets	IEC 625-2 Subclause
Source Handshake	SH1	5.1.1
Acceptor Handshake	AH1	5.1.2
Talker	T5, T6, TE5, or TE6	5.3
Listener	L3, L4, LE3, or LE4	5.4
Service Request	SR1	5.5
Remote/Local	RL0 or RL1	5.6
Parallel Poll	PP0 or PP1	5.7
Device Clear	DC1	5.8
Device Trigger	DT0 or DT1	5.9
Controller	C0, or C4 with C5, C7, C9, or C11	5.10
Electrical Interface	E1 or E2	5.11

A **device** shall comply with IEC 625-1. It shall also meet all requirements stated in clause 5 of this standard.

4.2 *Message exchange requirements*

A **device** shall follow all the requirements listed in clause 6 of this standard.

A **device**'s input buffer may take on several forms. The input buffer's length may be a fixed number of bytes. It may contain a fixed number of complete <PROGRAM MESSAGES>. Its length in bytes or <PROGRAM MESSAGES> may vary with the **device**'s state. See 6.1.5.

Certain query messages may generate the actual response message when the query is received. Other query messages may generate the actual response when the controller reads the response. See 6.4.5.4.

Execution of individual parsable elements may be done as they are received. The **device** may also wait until either a <PROGRAM MESSAGE UNIT SEPARATOR> or a <PROGRAM MESSAGE TERMINATOR> is parsed before executing preceding <PROGRAM MESSAGE UNIT> elements. A **device** may contain a mix of these types of commands. See 6.4.5.1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.3 *Prescriptions de syntaxe*

Les articles 7 et 8 décrivent un ensemble d'éléments fonctionnels que le concepteur de l'appareil utilise pour décrire le langage de programmation d'un **appareil** spécifique. Toute la syntaxe décrite aux articles 7 et 8 n'est pas nécessaire sur chaque **appareil**. Certains éléments fonctionnels sont nécessaires, d'autres sont facultatifs.

4.3.1 *Eléments fonctionnels nécessaires*

Le tableau 4-2 énumère les éléments fonctionnels nécessaires.

Tableau 4-2 – Eléments fonctionnels nécessaires pour les appareils

Elément fonctionnel écouteur de l'appareil	Paragraphes
<MESSAGE DE PROGRAMMATION>	7.3.2
<CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>	7.5
<UNITÉ DE MESSAGE DE PROGRAMMATION>	7.3.2
<SÉPARATEUR D'UNITÉS DE MESSAGE DE PROGRAMMATION>	7.4.1
<UNITÉ DE MESSAGE D'INSTRUCTION>	7.3.2
<UNITÉ DE MESSAGE D'INTERROGATION>	7.3.2
<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION>*	7.6.1
<EN-TÊTE D'INTERROGATION DE PROGRAMMATION>*	7.6.2
<SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION>	7.4.3
<SÉPARATEUR DE DONNÉES DE PROGRAMMATION>	7.4.2
<DONNÉES DE PROGRAMMATION>	7.3.2
<DONNÉES DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>	7.7.2
Elément fonctionnel parleur de l'appareil	**Paragraphes**
<MESSAGE DE RÉPONSE>	8.3.2
<CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>	8.5
<UNITÉ DE MESSAGE DE RÉPONSE>	8.3.2
<SÉPARATEUR D'UNITÉS DE MESSAGE DE RÉPONSE>	8.4.1
<DONNÉES DE RÉPONSE>	8.7
<SÉPARATEUR DE DONNÉES DE RÉPONSE>	8.4.2
<DONNÉES DE RÉPONSE NUMÉRIQUE NR1>	8.7.2
<DONNÉES DE RÉPONSE ARBITRAIRE ISO 7-BIT>	8.7.11

* L'<en-tête d'instruction de programmation complexe> et l'<en-tête d'interrogation de programmation complexe> ne sont pas des éléments de codage nécessaires.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.3.2 *Eléments fonctionnels facultatifs*

Le tableau 4-3 énumère les éléments fonctionnels facultatifs.

4.3 *Syntax requirements*

Clauses 7 and 8 describe a set of functional elements which the device designer uses to describe the programming language for a specific **device**. The entire syntax described in clauses 7 and 8 is not required in every **device**. Some functional elements are required and some are optional.

4.3.1 *Required functional elements*

Table 4-2 lists the required functional elements.

Table 4-2 – Required functional elements for devices

Device listening functional element	Subclause
<PROGRAM MESSAGE>	7.3.2
<PROGRAM MESSAGE TERMINATOR>	7.5
<PROGRAM MESSAGE UNIT>	7.3.2
<PROGRAM MESSAGE UNIT SEPARATOR>	7.4.1
<COMMAND MESSAGE UNIT>	7.3.2
<QUERY MESSAGE UNIT>	7.3.2
<COMMAND PROGRAM HEADER>*	7.6.1
<QUERY PROGRAM HEADER>*	7.6.2
<PROGRAM HEADER SEPARATOR>	7.4.3
<PROGRAM DATA SEPARATOR>	7.4.2
<PROGRAM DATA>	7.3.2
<DECIMAL NUMERIC PROGRAM DATA>	7.7.2
Device talking functional element	**Subclause**
<RESPONSE MESSAGE>	8.3.2
<RESPONSE MESSAGE TERMINATOR>	8.5
<RESPONSE MESSAGE UNIT>	8.3.2
<RESPONSE MESSAGE UNIT SEPARATOR>	8.4.1
<RESPONSE DATA>	8.7
<RESPONSE DATA SEPARATOR>	8.4.2
<NR1 NUMERIC RESPONSE DATA>	8.7.2
<ARBITRARY ISO 7-bit RESPONSE DATA>	8.7.11

* <compound command program header> and <compound query program header> are not required encoding elements.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.3.2 *Optional functional elements*

Table 4-3 lists the optional functional elements.

Tableau 4-3 – Eléments fonctionnels facultatifs pour les appareils

Elément fonctionnel écouteur de l'appareil	Paragraphes
<EN-TÊTE DE PROGRAMMATION D'INSTRUCTION>*	7.6.1
<EN-TÊTE DE PROGRAMMATION D'INTERROGATION>*	7.6.2
<DONNÉES DE PROGRAMMATION DE CARACTÈRES>	7.7.1
<DONNÉES DE PROGRAMMATION DE SUFFIXES>	7.7.3
<DONNÉES DE PROGRAMMATION NUMÉRIQUES NON DÉCIMALES>	7.7.4
<DONNÉES DE PROGRAMMATION DE CHAÎNES>	7.7.5
<DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE>	7.7.6
<DONNÉES DE PROGRAMMATION D'EXPRESSION>	7.7.7
Elément fonctionnel parleur de l'appareil	**Paragraphes**
<SÉPARATEUR D'EN-TÊTES DE RÉPONSE>	8.4.3
<EN-TÊTE DE RÉPONSE>	8.6
<DONNÉES DE RÉPONSES DE CARACTÈRES>	8.7.1
<DONNÉES DE RÉPONSES NUMÉRIQUES NR2>	8.7.3
<DONNÉES DE RÉPONSES NUMÉRIQUES NR3>	8.7.4
<DONNÉES DE RÉPONSES NUMÉRIQUES HEXADÉCIMALES>	8.7.5
<DONNÉES DE RÉPONSES NUMÉRIQUES OCTALES>	8.7.6
<DONNÉES DE RÉPONSES NUMÉRIQUES BINAIRES>	8.7.7
<DONNÉES DE RÉPONSES DE CHAÎNES>	8.7.8
<DONNÉES DE RÉPONSES DE BLOCS ARBITRAIRES> <DE LONGUEUR DÉFINIE>	8.7.9
<DONNÉES DE RÉPONSES DE BLOCS ARBITRAIRES> DE LONGUEUR INDÉFINIE>	8.7.10

* L'<en-tête d'instruction de programmation complexe> et l'<en-tête d'interrogation complexe> sont des éléments de codage facultatifs bien que d'autres éléments de codage soient nécessaires.

4.4 *Prescriptions d'indication de l'état*

4.4.1 *Capacité d'indication de l'état exigée*

Un **appareil** doit suivre le modèle d'indication de l'état présenté à l'article 11. L'**appareil** doit comprendre les instructions concernant les indications de l'état décrites à l'article 10 et énumérées dans le tableau 4-4.

Tableau 4-4 – Instructions communes d'indication d'état exigées

Indication d'état	
Instruction	Paragraphes
*CLS	10.3
*ESE	10.10
*ESE?	10.11
*ESR?	10.12
*SRE	10.34
*SRE?	10.35
*STB?	10.36

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Table 4-3 – Optional functional elements for devices

Device listening functional element	Subclause
<COMMAND PROGRAM HEADER>*	7.6.1
<QUERY PROGRAM HEADER>*	7.6.2
<CHARACTER PROGRAM DATA>	7.7.1
<SUFFIX PROGRAM DATA>	7.7.3
<NON-DECIMAL NUMERIC PROGRAM DATA>	7.7.4
<STRING PROGRAM DATA>	7.7.5
<ARBITRARY BLOCK PROGRAM DATA>	7.7.6
<EXPRESSION PROGRAM DATA>	7.7.7
Device talking functional element	**Subclause**
<RESPONSE HEADER SEPARATOR>	8.4.3
<RESPONSE HEADER>	8.6
<CHARACTER RESPONSE DATA>	8.7.1
<NR2 NUMERIC RESPONSE DATA>	8.7.3
<NR3 NUMERIC RESPONSE DATA>	8.7.4
<HEXADECIMAL NUMERIC RESPONSE DATA>	8.7.5
<OCTAL NUMERIC RESPONSE DATA>	8.7.6
<BINARY NUMERIC RESPONSE DATA>	8.7.7
<STRING RESPONSE DATA>	8.7.8
<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>	8.7.9
<INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>	8.7.10
* <compound command program header> and <compound query program header> are optional encoding elements though other encoding elements are required.	

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.4 *Status reporting requirements*

4.4.1 *Required status reporting capability*

A **device** shall follow the status reporting model presented in clause 11. The **device** shall include those commands related to status reporting described in clause 10 and listed in table 4-4.

Table 4-4 – Required status reporting common commands

Status reporting	
Command	Subclause
*CLS	10.3
*ESE	10.10
*ESE?	10.11
*ESR?	10.12
*SRE	10.34
*SRE?	10.35
*STB?	10.36

Un **appareil** doit mettre en application le registre d'octets de l'état, le registre de validation de demande de service, le registre d'état d'événement standard et le registre de validation d'état d'événement standard illustrés à la figure 4-1.

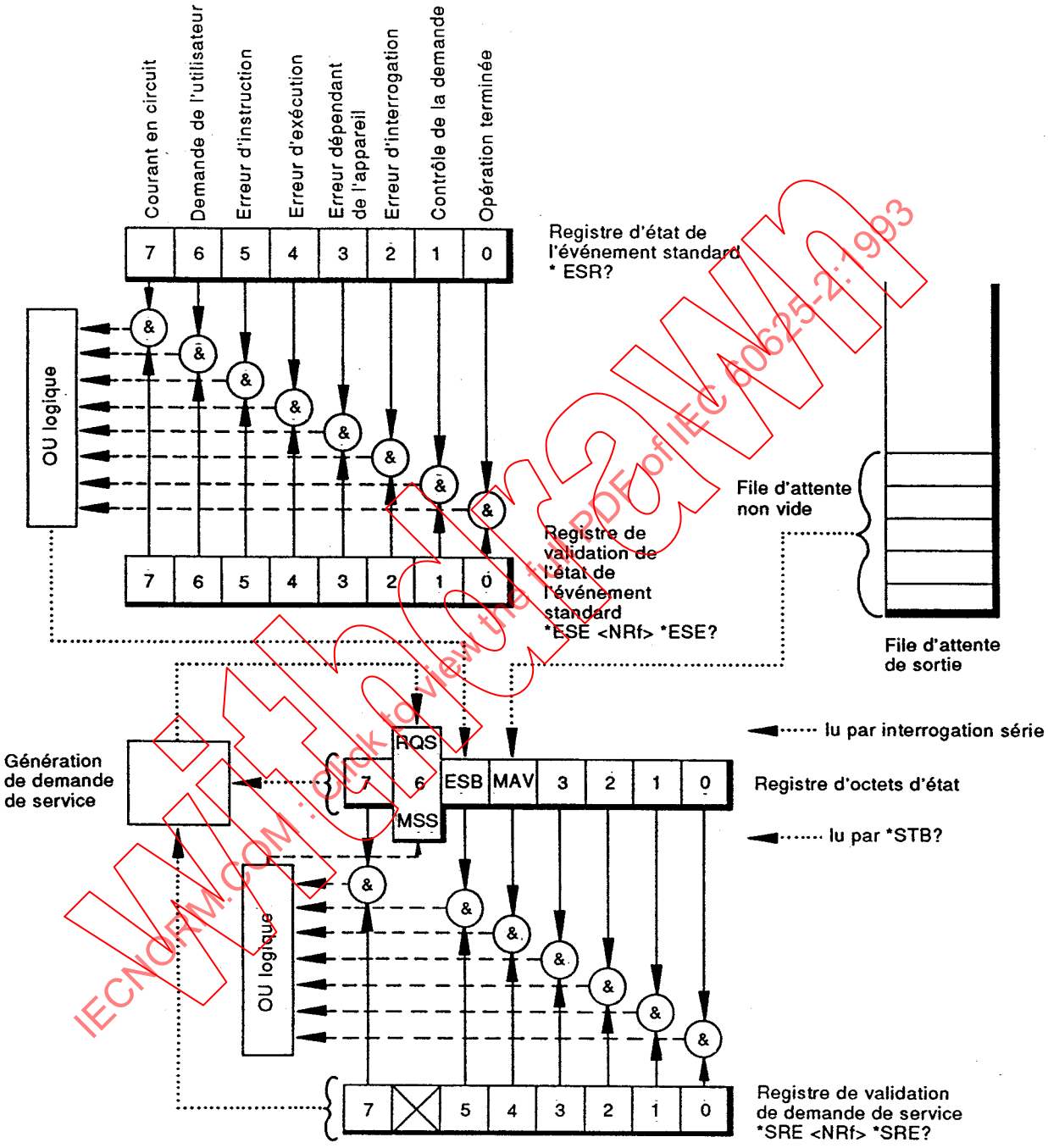


Figure 4-1 – Capacités exigées d'indication d'état

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

A **device** shall implement the status byte register, the Service Request Enable Register, the Standard Event Status Register, and the Standard Event Status Enable Register illustrated in figure 4-1.

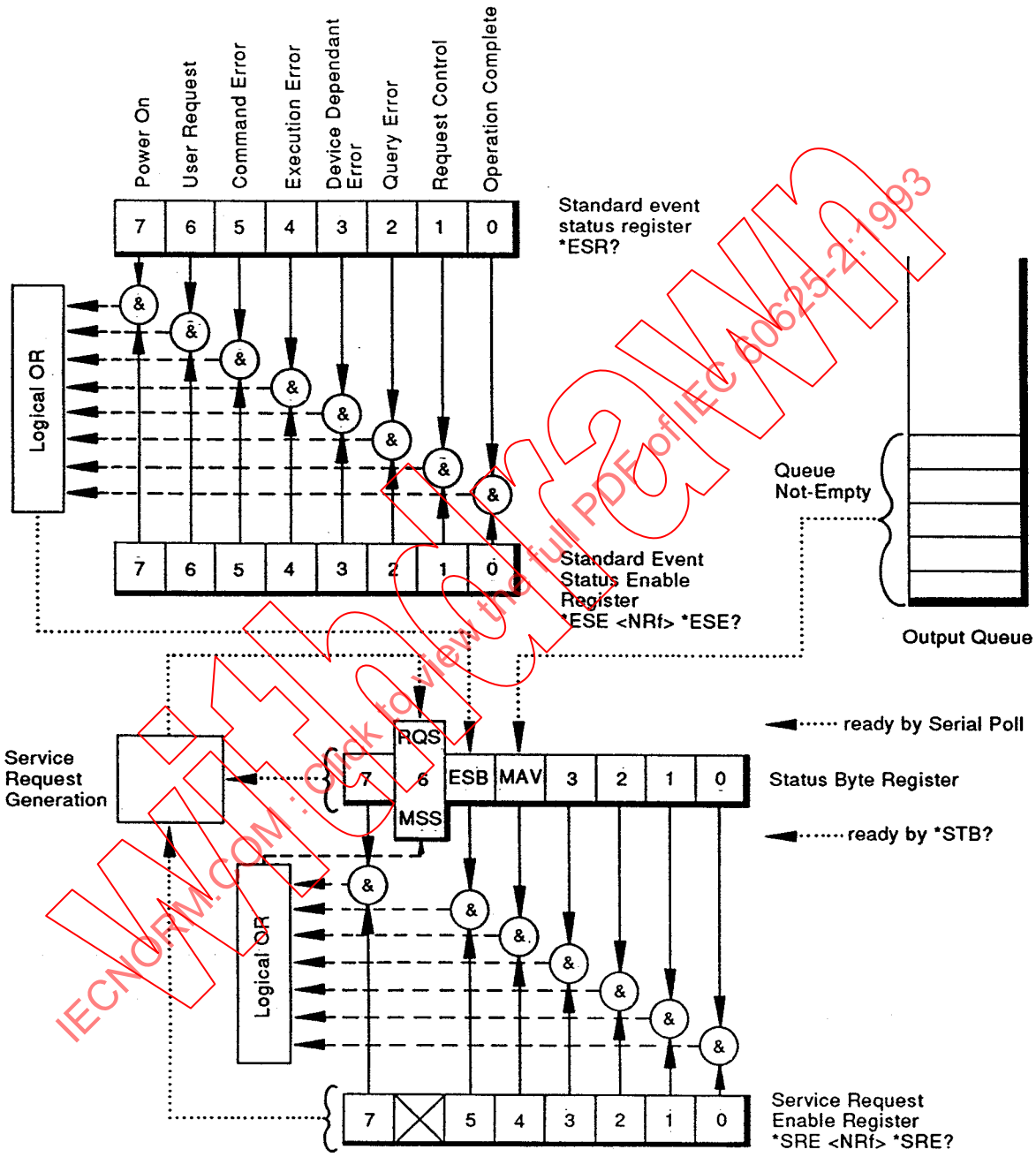


Figure 4-1 – Required status reporting capabilities

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.4.2 *Capacité d'indication d'état facultative*

Un **appareil** peut comporter n'importe quel nombre de registres des conditions, registres d'événements, registres de validation et files d'attente à condition qu'ils suivent le modèle présenté à l'article 11.

L'**appareil** peut facultativement comporter la possibilité de sauvegarder des registres de validation lorsque l'alimentation est coupée. Cette capacité exige également une mémoire non volatile et la totalité des instructions énumérées dans le tableau 4-5.

Tableau 4-5 – Instructions communes d'alimentation en circuit facultatives

Alimentation en circuit	
Instruction	Paragraphes
*PSC *PSC?	10.25 10.26

L'**appareil** peut facultativement comporter la possibilité de répondre à une interrogation parallèle. Cette capacité exige également le sous-ensemble CEI 625-1 PP1 et toutes les instructions énumérées dans le tableau 4-6.

Tableau 4-6 – Instructions communes d'interrogation parallèle facultatives

Interrogation parallèle	
Instruction	Paragraphes
*IST? *PRE *PRE?	10.15 10.23 10.24

4.5 *Instructions communes*

La présente norme énumère quelques instructions réservées. Certaines de ces instructions sont exigées, d'autres sont facultatives, et certaines doivent être implémentées par groupes.

4.5.1 *Instructions communes exigées*

Les instructions communes énumérées en 4.4.1, relatives aux indications d'état et les instructions communes énumérées en 4-6, relatives à la synchronisation sont exigées. Les instructions énumérées dans le tableau 4-7, relatives aux opérations internes, sont également exigées.

Tableau 4-7– Instructions communes d'opérations internes exigées

Opérations internes	
Instruction	Paragraphes
*IDN? *RST *TST?	10.14 10.32 10.38

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.4.2 *Optional status reporting capability*

A **device** may include any number of condition registers, event registers, enable registers, and queues provided they follow the model presented in clause 11.

The **device** may optionally include the ability to save enable registers when power is cycled. This capability also requires non-volatile memory and all of the commands listed in table 4-5.

Table 4-5 – Optional power-on common commands

Power-on	
Command	Subclause
*PSC *PSC?	10.25 10.26

The **device** may optionally include the ability to respond to a parallel poll. This capability also requires the IEC 625-1 subset PP1 and all of the commands listed in table 4-6.

Table 4-6 – Optional Parallel Poll common commands

Parallel Poll	
Command	Subclause
*IST? *PRE *PRE?	10.15 10.23 10.24

4.5 *Common commands*

This standard lists some reserved commands. Some of the commands are required, some are optional, and some shall be implemented in groups.

4.5.1 *Required common commands*

The common commands, listed in 4.4.1, related to status reporting and the common commands, listed in 4.6, related to synchronization are required. The commands, listed in table 4-7, related to internal operations are also required.

Table 4-7 – Required internal operation common commands

Internal operations	
Command	Subclause
*IDN? *RST *TST?	10.14 10.32 10.38

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.5.2 *Instructions communes facultatives*

Dans certains cas, l'implémentation d'une instruction commune est indépendante de toute autre instruction commune ou capacité de l'appareil. Dans d'autres cas, les instructions communes doivent être implémentées par groupes ou conjointement avec une autre capacité de l'appareil.

4.5.2.1 Instructions de description des ressources

Les instructions de description des ressources, énumérées dans le tableau 4-8, sont facultatives et ne sont pas liées entre elles de façon rigide. Si la description des ressources peut être écrite dans l'**appareil** (*RDT), l'**appareil** doit également prévoir la possibilité de lire la description des ressources (*RDT?). L'interrogation *RDT? peut toutefois être incluse sans l'instruction *RDT.

Tableau 4-8 – Instructions communes facultatives de description des ressources

Description des ressources	
Instruction	Paragraphes
*RDT *RDT?	10.30 10.31

4.5.2.2 Instructions protégées de données de l'utilisateur

Les instructions protégées de données de l'utilisateur, énumérées dans le tableau 4-9, sont facultatives. Toutefois, si des instructions de ce groupe sont implémentées, toutes les instructions de ce groupe doivent être implémentées.

Tableau 4-9 – Instructions facultatives protégées de données de l'utilisateur

Données protégées	
Instruction	Paragraphes
*PUD *PUD?	10.27 10.28

4.5.2.3 Instructions d'étalonnage

L'instruction d'auto-étalonnage, figurant dans le tableau 4-10, est facultative.

Tableau 4-10 – Instructions d'étalonnage facultative

Etalonnage	
Instruction	Paragraphe
*CAL?	10.2

4.5.2.4 Instructions de déclenchement

L'instruction de déclenchement, figurant dans le tableau 4-11, est facultative. Toutefois, elle est nécessaire si l'**appareil** a une capacité DT1. Si l'instruction de déclenchement est implémentée, l'**appareil** doit avoir également une capacité DT1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.5.2 *Optional common commands*

In some cases, the implementation of a common command is independent of other common commands or device capabilities. In other cases, common commands shall be implemented in groups or in conjunction with some other device capability.

4.5.2.1 Resource description commands

The resource description commands, listed in table 4-8, are optional and loosely coupled. If the resource description can be written into the **device** (*RDT), then the **device** shall also include the capability to read the resource description (*RDT?). The *RDT? query, however, may be included without the *RDT command.

Table 4-8 – Optional resource description common commands

Resource description	
Command	Subclause
*RDT *RDT?	10.30 10.31

4.5.2.2 Protected user data commands

The protected user data commands, listed in table 4-9, are optional. If any commands in this group are implemented, however, then all commands in this group shall be implemented.

Table 4-9 – Optional protected user data commands

Protected data	
Command	Subclause
*PUD *PUD?	10.27 10.28

4.5.2.3 Calibration command

The self-calibration command, listed in table 4-10, is optional.

Table 4-10 – Optional calibration command

Calibration	
Command	Subclause
*CAL?	10.2

4.5.2.4 Trigger command

The trigger command, listed in table 4-11, is optional. It is required, however, if the **device** has DT1 capability. If the trigger command is implemented the **device** shall also have DT1 capability.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993 Withdrawn

Tableau 4-11 – Instruction de déclenchement facultative

Déclenchement	
Instruction	Paragraphe
*TRG	10.37

4.5.2.5 Macro-instructions de déclenchement

Les macro-instructions de déclenchement, énumérées dans le tableau 4-12, sont facultatives. Toutefois, si des instructions de ce groupe sont implémentées, toutes les instructions de ce groupe devront être implémentées et l'**appareil** devra avoir une capacité DT1.

Tableau 4-12 – Macro-instructions de déclenchement facultatives

Macro-instructions de déclenchement	
Instruction	Paragraphes
*DDT	10.4
*DDT?	10.5

Si les macro-instructions de déclenchement sont implémentées, l'**appareil** devra également comporter les éléments fonctionnels de <DONNÉES DE PROGRAMMATION DE BLOCS ARBITRAIRES> et de <DONNÉES DE RÉPONSES DE BLOCS ARBITRAIRES DE LONGUEUR DÉFINIE>.

4.5.2.6 Macro-instructions

Les macro-instructions, énumérées dans le tableau 4-13, sont facultatives. Toutefois, si des instructions de ce groupe sont implémentées, toutes les instructions de ce groupe devront être implémentées.

Tableau 4-13 – Macro-instructions facultatives

Macro-instructions	
Instruction	Paragraphes
*DMC	10.7
*EMC	10.8
*EMC?	10.9
*GMC?	10.13
*LMC?	10.16
*PMC	10.22

Si les macro-instructions sont implémentées, l'**appareil** devra également comprendre les éléments fonctionnels de <DONNÉES DE PROGRAMMATION DE CHAÎNES>, <DONNÉES DE PROGRAMMATION DE BLOCS ARBITRAIRES>, <DONNÉES DE RÉPONSE DE CHAÎNES> et <DONNÉES DE RÉPONSES DE BLOCS ARBITRAIRES DE LONGUEUR DÉFINIE>.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Table 4-11 – Optional trigger command

Trigger	
Command	Subclause
*TRG	10.37

4.5.2.5 Trigger macro commands

The trigger macro commands, listed in table 4-12, are optional. If any commands in this group are implemented, however, then all commands in this group shall be implemented and the **device** shall have DT1 capability.

Table 4-12 – Optional trigger macro commands

Trigger macro	
Command	Subclause
*DDT *DDT?	10.4 10.5

If the trigger macro commands are implemented, the **device** shall also include the <ARBITRARY BLOCK PROGRAM DATA> and <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> functional elements.

4.5.2.6 Macro commands

The macro commands, listed in table 4-13, are optional. If any commands in this group are implemented, however, then all commands in this group shall be implemented.

Table 4-13 – Optional macro commands

Macros	
Command	Subclause
*DMC *EMC *EMC? *GMC? *LMC? *PMC	10.7 10.8 10.9 10.13 10.16 10.22

If the macro commands are implemented, the **device** shall also include the <STRING PROGRAM DATA>, <ARBITRARY BLOCK PROGRAM DATA>, <STRING RESPONSE DATA>, and <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> functional elements.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.5.2.7 Instruction d'identification d'option

L'instruction d'identification d'option, figurant dans le tableau 4-14, est facultative.

Tableau 4-14 – Instruction facultative d'identification d'option

Options	
Instruction	Paragraphe
*OPT?	10.20

4.5.2.8 Instructions de positionnements mémorisés

Les instructions de positionnements mémorisés, énumérées dans le tableau 4-15, sont facultatives. Toutefois, si une instruction de ce groupe est implémentée, toutes les instructions de ce groupe devront être implémentées.

Tableau 4-15 – Instructions de positionnements mémorisés facultatives

Positionnements mémorisés	
Instruction	Paragraphes
*RCL	10.29
*SAV	10.33

4.5.2.9 Instruction d'apprentissage

L'instruction d'apprentissage, figurant dans le tableau 4-16, est facultative.

Tableau 4-16 – Instruction d'apprentissage facultative

Apprentissage	
Instruction	Paragraphe
*LRN?	10.17

4.6 *Prescriptions de synchronisation*

L'article 12 décrit la capacité exigée sur tous les appareils pour la synchronisation. Un concepteur d'appareils peut choisir quelles sont les opérations qui comportent des signalisations d'opérations en cours associées avec celles-ci suivant les règles de l'article 12.

Les instructions communes exigées relatives à la synchronisation sont énumérées dans le tableau 4-17.

Tableau 4-17 – Instructions de synchronisation exigées

Synchronisation	
Instruction	Paragraphes
*OPC	10.18
*OPC?	10.19
*WAI	10.39

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.5.2.7 Option identification command

The option identification command, listed in table 4-14, is optional.

Table 4-14 – Optional option identification command

Options	
Command	Subclause
*OPT?	10.20

4.5.2.8 Stored setting commands

The stored setting commands, listed in table 4-15, are optional. If any commands in this group are implemented, however, then all commands in this group shall be implemented.

Table 4-15 – Optional stored setting commands

Stored settings	
Command	Subclause
*RCL *SAV	10.29 10.33

4.5.2.9 Learn command

The learn command, listed in table 4-16, is optional.

Table 4-16 – Optional learn command

Learn	
Command	Subclause
*LRN?	10.17

4.6 *Synchronization requirements*

Clause 12 describes capability which is required in all **devices** for synchronization. A device designer may choose which operations have pending operation flags associated with them, according to the rules in clause 12.

The required common commands related to synchronization are listed in table 4-17.

Table 4-17 – Required synchronization commands

Synchronization	
Command	Subclause
*OPC *OPC? *WAI	10.18 10.19 10.39

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.7 *Capacités de configuration du système*

La section 13 décrit les capacités facultatives qui sont utilisées pour configurer les adresses CEI 625-1 des **appareils**. Ces capacités sont implémentées en prévoyant les instructions communes dans le tableau 4-18.

Tableau 4-18 – Instructions de configuration du système facultatives

Configuration automatique	
Instruction	Paragraphes
*AAD *DLF	10.1 10.6

Si l'une ou l'autre de ces instructions est implémentée, les deux instructions devront être implémentées.

Ces instructions font que l'**appareil** suspend momentanément les protocoles d'échange de messages normaux et utilise des protocoles différents. Pour assurer la compatibilité avec d'autres appareils, le concepteur de l'appareil devra respecter strictement ces différents protocoles.

4.8 *Capacités des contrôleurs*

4.8.1 *Capacités exigées des contrôleurs*

Un **appareil** ne doit pas nécessairement avoir une capacité de contrôleur.

4.8.2 *Capacités facultatives des contrôleurs*

Si un **appareil** a une capacité de contrôleur, il devra posséder le sous-ensemble CEI 625-1 C4, l'instruction de passage du contrôle figurant dans le tableau 4-19, il devra obéir au protocole de passage de contrôle décrit en 17.4, et il devra obéir au protocole de demande de contrôle décrit en 17.5.

Tableau 4-19 – Instruction de passage de contrôle facultative

Passage de contrôle	
Instruction	Paragraphe
*PCB	10.21

4.9 *Prescriptions concernant la documentation sur l'appareil*

Tous les **appareils** doivent fournir des informations à l'utilisateur sur la façon dont l'**appareil** a implémenté cette norme. Ces informations comprendront:

1) Une liste des sous-ensembles de fonctions d'interfaces CEI 625-1 implémentés, article 5.
2) Une description du comportement de l'**appareil** lorsque l'adresse est assignée en dehors de la gamme de 0 à 30, voir 5.2.
3) Une description lorsqu'un changement d'adresse lancé par un utilisateur est reconnu par l'**appareil**.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

4.7 *System configuration capability*

Clause 13 describes optional capability which is used to configure the IEC 625-1 addresses of **devices**. This capability is implemented by including the common commands in table 4-18.

Table 4-18 – Optional system configuration commands

Auto configure	
Command	Subclause
*AAD *DLF	10.1 10.6

If either command is implemented then both commands shall be implemented.

These commands cause the **device** temporarily to suspend normal message exchange protocols and to use different protocols. To ensure compatibility with other devices, the device designer must carefully adhere to these different protocols.

4.8 *Controller capability*

4.8.1 *Required controller capability*

A **device** is not required to have any controller capability.

4.8.2 *Optional controller capability*

If a **device** has any controller capability, it shall include the IEC 625-1 C4 subset, the pass control back command listed in table 4-19, and it shall obey the passing control protocol described in 17.4, and it shall obey the requesting control protocol described in 17.5.

Table 4-19 – Optional passing control command

Passing control	
Command	Subclause
*PCB	10.21

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4.9 *Device documentation requirements*

All **devices** shall supply information to the user about how the **device** has implemented this standard. This information shall include:

1) A list of IEC 625-1 interface functions subsets implemented, clause 5.

2) A description of **device** behaviour when the address is set outside the range 0-30, see 5.2.

3) A description of when a user-initiated address change is recognized by the **device**.

4) Une description du réglage de l'**appareil** à la mise sous tension, voir 5.12. Toutes les instructions modifiant les réglages à la mise sous tension devront également être incluses.

5) Une description des options d'échange de messages:

   a) la taille et le comportement du tampon d'entrée, voir 6.1.5;

   b) quelles sont les interrogations qui renvoient plus d'une <UNITÉ DE MESSAGE DE RÉPONSE>, voir 6.4.3;

   c) quelles sont les interrogations qui génèrent une réponse lorsqu'elles sont analysées, voir 6.4.5.4;

   d) quelles sont les interrogations qui génèrent une réponse lorsqu'elles sont lues, voir 6.4.5.4;

   e) quelles sont les instructions qui sont couplées, voir 6.4.5.3.

6) Une liste des éléments fonctionnels utilisés dans la création des instructions spécifiques à l'appareil. Le fait que des éléments d'<en-tête de programmation d'instruction complexe> soient utilisés doit être également pris en compte, voir 7.1.1 et 7.3.3.

7) Une description de toute limitation de la taille des mémoires tampons relatives aux données de blocs, voir 7.7.6.5.

8) Une liste des éléments de <DONNÉES DE PROGRAMMATION> qui peuvent apparaître à l'intérieur d'une expression de même que la profondeur d'emboîtement de sous-expression maximale. Toute restriction de syntaxe que l'**appareil** peut placer sur l'<expression> doit également être incluse.

9) Une description de la syntaxe de réponse pour chaque interrogation, voir 8.

10) Une description de tout trafic de transfert de messages d'un appareil à l'autre qui ne suit pas les règles pour les éléments de <MESSAGES DE RÉPONSE>, voir 8.1.

11) La longueur de toutes réponses de données de blocs, voir 8.7.9.4.

12) Une liste des instructions et interrogations communes qui sont implémentées, voir 10.

13) Une description de l'état de l'**appareil** après l'exécution réussie de l'interrogation d'étalonnage, voir 10.2.

14) La longueur maximale du bloc utilisé pour définir la macro-instruction de déclenchement, si *DDT est implémenté, voir 10.4.

15) La longueur maximale des étiquettes de macro, la longueur maximale du bloc utilisé pour définir une macro-instruction, et comment la récurrence est traitée pendant l'expansion des macros, si les macro-instructions sont implémentées, voir 10.7.

16) Une description de la réponse à l'interrogation commune d'identification, *IDN?, voir 10.14.

17) La longueur de la zone de mémoire des données utilisateur protégées, *PUD, voir 10.27.

18) La longueur de la description des ressources, si l'instruction *RDT ou l'interrogation *RDT? sont implémentées, voir 10.30 et 10.31.

19) Une description des états affectés par *RST (10.32), *LRN? (10.17), *RCL (10.29), et *SAV (10.33).

20) Une description de l'étendue de l'auto-test réalisé par l'interrogation *TST?, voir 10.38.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

4) A description of the **device** setting at power-on, see 5.12. Any commands which modify the power-on settings shall also be included.

5) A description of message exchange options:

   a) the size and behaviour of the Input Buffer, see 6.1.5;

   b) which queries return more than one <RESPONSE MESSAGE UNIT>, see 6.4.3;

   c) which queries generate a response when parsed, see 6.4.5.4;

   d) which queries generate a response when read, see 6.4.5.4;

   e) which commands are coupled, see 6.4.5.3.

6) A list of functional elements used in constructing the device-specific commands. Whether <compound command program header> elements are used must also be included, see 7.1.1 and 7.3.3.

7) A description of any buffer size limitations related to block data, see 7.7.6.5.

8) A list of the <PROGRAM DATA> elements which may appear within an <expression> as well as the maximum sub-expression nesting depth. Any additional syntax restrictions which the **device** may place on the <expression> shall also be included.

9) A description of the response syntax for every query, see 8.

10) A description of any device-to-device message transfer traffic which does not follow the rules for <RESPONSE MESSAGE> elements, see 8.1.

11) The size of any block data responses, see 8.7.9.4.

12) A list of common commands and queries which are implemented, see 10.

13) A description of the state of the **device** after successful completion of the calibration query, see 10.2.

14) The maximum length of the block used to define the trigger macro, if *DDT is implemented, see 10.4.

15) The maximum length of macro labels, the maximum length of the block used to define a macro, and how recursion is handled during macro expansion, if the macro commands are implemented, see 10.7.

16) A description of the response to the identification common query, *IDN?, see 10.14.

17) The size of the protected user data storage area, *PUD, see 10.27.

18) The size of the resource description, if the *RDT command or *RDT? query are implemented, see 10.30 and 10.31.

19) A description of the states affected by *RST, (10.32), *LRN?, (10.17), *RCL, (10.29), and *SAV, (10.33).

20) A description of the scope of the self-test performed by the *TST? query, see 10.38.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

21) Une description des structures de données d'état supplémentaires utilisées dans l'indication d'état de l'**appareil**, article 11.

22) Pour chaque instruction, un état indiquant si elle est recouvrable ou séquentielle.

23) Pour chaque instruction, la documentation de l'appareil devra spécifier les critères fonctionnels qui sont remplis lorsqu'un message "opération terminée" est généré en réponse à cette instruction, voir 12.8.3.

5 Prescriptions de fonction d'interface de l'appareil

Cet article décrit les prescriptions de fonction d'interface CEI 625-1 d'un **appareil**. Il spécifie les prescriptions supplémentaires d'un **appareil** directement associées aux dix fonctions d'interface CEI 625-1. Ces prescriptions sont destinées à compléter la spécification CEI 625-1 pour l'environnement du système CEI 625-2 tel qu'il est décrit dans la présente norme.

5.1 *Prescriptions d'établissement d'une liaison*

5.1.1 *Prescriptions d'établissement d'une liaison avec la source*

Un **appareil** doit contenir le sous-ensemble SH1 (capacité complète). Il doit réaliser la transition de l'état de temporisation de source (SDYS) à l'état de transfert de source (STRS) seulement si le message "prêt pour données" (RFD) est VRAI et le message "données acceptées" (DAC) est FAUX, voir 6.5.3.

5.1.2 *Prescriptions d'établissement d'une liaison avec l'accepteur*

Un **appareil** doit contenir le sous-ensemble AH1 (capacité complète) et doit suivre en plus l'exigence du paragraphe suivant.

Un **appareil** doit entrer en AIDS (état inactif de l'accepteur) de la fonction AH en une milliseconde après qu'ATN soit entré dans l'état FAUX si l'**appareil** était en LIDS (état inactif de l'écouteur) lorsque ATN était VRAI. Cette exigence assure que l'**appareil** fonctionnera de façon fiable avec le protocole de contrôleur commun de recherche des écouteurs, voir 17.6.

5.2 *Prescriptions d'adresses*

Un **appareil** doit avoir les mêmes adresses parleur et écouteur. Les 5 bits inférieurs des codes MTA et MLA, l'adresse primaire de l'**appareil**, doivent être identiques (voir CEI 625-1, 39). Si l'**appareil** utilise un adressage étendu, les adresses MSA ("my secondary addresses" = mes adresses secondaires) parleur et écouteur doivent également être identiques.

Un **appareil** doit avoir une adresse primaire unique pouvant être réglée par l'utilisateur à n'importe quelle valeur sur une plage de 0 à 30. Si l'**appareil** utilise un adressage étendu, l'utilisateur devra également être en mesure de régler l'adresse secondaire unique sur la même plage.

L'adresse d'un **appareil** doit être réglable indépendamment de toute autre adresse de l'appareil même si les appareils partagent des ressources physiques communes telles que la même rubrique de description.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

21) A description of additional status data structures used in the **device's** status reporting, see 11.

22) For each command, a statement describing whether it is overlapped or sequential.

23) For each command the device documentation shall specify the functional criteria that are met when an operation complete message is generated in response to that command, see 12.8.3.

5 Device interface function requirements

This clause describes the IEC 625-1 interface function requirements of a **device**. It specifies the additional requirements of a **device** that are directly associated with the ten IEC 625-1 interface functions. These requirements are intended to supplement the IEC 625-1 specification for the IEC 625-2 system environment as described in this standard.

5.1 *Handshake requirements*

5.1.1 *Source handshake requirements*

A **device** shall contain the SH1 (complete capability) subset. It shall make the transition from Source Delay State (SDYS) to Source Transfer State (STRS) only if the ready for data (RFD) message is TRUE and the data accepted (DAC) message is FALSE, see 6.5.3.

5.1.2 *Acceptor handshake requirements*

A **device** shall contain the AH1 (complete capability) subset and shall additionally follow the requirement of the following paragraph.

A **device** shall enter AIDS (Acceptor Idle State) of the AH function within one millisecond after ATN assumes the FALSE state if the **device** was in LIDS (Listener Idle State) when ATN was TRUE. This requirement ensures that the **device** will operate reliably with the Find Listeners common controller protocol, see 17.6.

5.2 *Address requirements*

A **device** shall have the same talk and listen addresses. The lower 5 bits of the MTA and MLA codes, the primary address of the **device**, shall be identical (see IEC 625-1, 39). If the **device** utilizes extended addressing, the talk and listen MSA's (my secondary addresses) shall also be identical.

A **device** shall have a single primary address which may be set by the user to any value within the range of 0 to 30. If the **device** utilizes extended addressing, the user shall also be able to set the single secondary address within the same range.

The address of a **device** shall be settable independent from any other device's address even if the devices share common physical resources such as the same enclosure.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Un **appareil** doit avoir un moyen local de sélection de son adresse primaire CEI 625-1 (et de son adresse secondaire si on utilise un adressage étendu) qui devra être conservé pendant les coupures de courant. L'opérateur de l'appareil doit pouvoir modifier cette adresse.

La configuration physique, l'étiquetage et le positionnement du mécanisme de sélection d'adresse doivent suivre les directives de CEI 625-1, annexe L.

Il est recommandé que le concepteur de l'appareil évite d'utiliser des réglages d'adresses en dehors de la plage de 0 à 30. Toute exception à cette recommandation doit être justifiée par des documents en indiquant le comportement à suivre.

Si l'on utilise des commutateurs matériels pour régler l'adresse et si l'adresse est apparemment réglée sur 31, l'**appareil** devra fonctionner de manière à ne pas interrompre le trafic de bus du système.

La documentation de l'appareil doit signaler quand un changement d'adresse lancé par un utilisateur est reconnu par l'**appareil**. Un **appareil** doit mettre à jour son adresse à la mise sous tension.

5.3 *Prescriptions parleur*

Un **appareil** doit contenir les sous-ensembles T5, T6, TE5, ou TE6. Ces sous-ensembles exigent le parleur de base avec interrogation série et sans adresse si on utilise le code MLA.

Cette norme suppose que le contrôleur exécute tout l'adressage du parleur par l'intermédiaire du code MTA et le non-adressage par l'intermédiaire de MLA, OTA, et UNT. La capacité du mode parleur uniquement n'est donc pas une condition requise dans cette spécification. La présence d'un mode de fonctionnement parleur uniquement (par exemple, dans un système sans contrôleur) est admissible mais le fonctionnement dans ce mode est considéré comme dépassant le cadre de la présente norme.

5.4 *Prescriptions écouteur*

Un **appareil** doit contenir les sous-ensembles L3, L4, LE3 ou LE4. Ces sous-ensembles exigent un écouteur de base sans adresse si on est en code MTA.

Cette norme suppose que le contrôleur exécute tout l'adressage de l'écouteur par l'intermédiaire de MLA et le non-adressage par l'intermédiaire de MTA et UNL. La capacité de mode écouteur uniquement n'est donc pas une condition imposée par cette spécification. La présence d'un mode de fonctionnement écouteur uniquement, par exemple, dans un système sans contrôleur, est acceptable mais le fonctionnement dans ce mode est considéré comme dépassant le cadre de la présente norme.

Noter la prescription supplémentaire pour rendre inactive la fonction d'établissement d'une liaison avec l'accepteur pendant une transition ATN FAUX en LIDS (état inactif de l'écouteur). Voir 5.1.2.

5.5 *Prescriptions de demande de service*

Un **appareil** doit contenir le sous-ensemble SR1 (capacité complète) et doit être conforme aux prescriptions du traitement de l'état de l'article 11.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

A **device** shall have a local means of selecting its IEC 625-1 primary address (and secondary address if extended addressing is used) that shall be retained during power-off. The device operator shall be capable of altering this address.

The physical configuration, labelling, and positioning of the address selection mechanism shall follow the guidelines of IEC 625-1 annex L.

The device designer should avoid using address settings outside the range 0 to 30. Any exceptions to this recommendation shall be documented with the associated behaviour.

If hard switches are used to set the address and the address is apparently set to 31, the **device** shall operate in a manner that does not disrupt system bus traffic.

Device documentation shall state when a user-initiated address change is recognized by the **device**. A **device** shall update its address at power-on.

5.3 *Talker requirements*

A **device** shall contain either the T5, T6, TE5, or TE6 subsets. These subsets require the basic talker with serial poll and unaddress if MLA.

This standard assumes that the controller performs all talker addressing via MTA and unaddressing via MLA, OTA, and UNT. Talk-only mode capability is therefore not a requirement of this specification. The presence of a talk-only mode of operation (for example, in a controller-less system) is allowable, but operation in this mode is considered beyond the scope of this standard.

5.4 *Listener requirements*

A **device** shall contain either the L3, L4, LE3, or LE4 subsets. These subsets require the basic listener with unaddress if MTA.

This standard assumes the controller performs all listener addressing via MLA and unaddressing via MTA and UNL. Listen-only mode capability is therefore not a requirement of this specification. The presence of a listen-only mode of operation (for example, in a controller-less system) is allowable, but operation in this mode is considered beyond the scope of this standard.

Note the additional requirement for inactivating the Acceptor Handshake function on an ATN FALSE transition in LIDS (Listener Idle State). See 5.1.2.

5.5 *Service request requirements*

A **device** shall contain the SR1 (complete capability) subset and shall conform to the status handling requirements of clause 11.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.6 *Prescriptions de mode de fonctionnement à distance ou local*

5.6.1 *Définitions du contrôle et des opérations*

Les fonctions de contrôle et les opérations CEI 625-2 sont classées en utilisant la source de contrôle et la méthode d'annonce.

5.6.1.1 Fonctionnement à distance CEI 625-2

Une opération à distance CEI 625-2 est une opération d'une fonction d'un appareil dans un **système**, s'effectuant par des messages de programmation par l'intermédiaire de l'**interface du système**.

5.6.1.2 Fonctionnement local CEI 625-2

Une opération locale CEI 625-2 est une opération d'une fonction d'un appareil qui n'est pas une opération à distance CEI 625-2.

Par exemple, la manoeuvre d'un interrupteur accessible à l'utilisateur, d'un bouton, d'un emplacement sur l'écran tactile, etc. physiquement solidaire d'un **appareil** et utilisé pour commander localement ou programmer l'**appareil**.

Le fonctionnement d'un **appareil** par n'importe quel autre bus ou interface reliés à l'**appareil** est considéré comme un fonctionnement local du point de vue du **système**.

5.6.1.3 Commande locale

Une commande locale réalise une opération locale CEI 625-2 d'un **appareil**.

Les commandes locales comprennent les entrées sur l'**appareil** destinées à sa commande.

Les commandes locales et les prescriptions correspondantes de mode de fonctionnement à distance ou local indiquées dans cette section sont destinées à s'appliquer seulement aux commandes locales dont la fonction peut également être effectuée par fonctionnement à distance CEI 625-2.

Les commandes locales dont la fonction ne peut pas être effectuée par l'intermédiaire d'un fonctionnement à distance CEI 625-2 se trouvent en dehors du cadre de la présente norme.

Les fonctions de commande suivantes sont explicitement exclues de la catégorie des commandes locales.

1) La commutation de l'alimentation
2) La génération du message (de demande de l'utilisateur) URQ.

5.6.1.4 Signal de commande extérieur

Un signal de commande extérieur sollicite des actions de l'**appareil** en utilisant un point d'accès sur l'appareil autre que le **système d'interface**. Les signaux de commande extérieurs dépassent le cadre de la présente norme, mais peuvent facultativement fonctionner comme une commande locale. Les signaux de commande extérieurs, toutefois, ne doivent pas enfreindre les protocoles d'échange de messages de l'article 6.

Exemple de signal de commande extérieur: un déclenchement extérieur de voltmètre numérique qui déclenche une mesure. Les résultats peuvent alors être lus sur l'appareil.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.6 *Remote/Local requirements*

5.6.1 *Control and operation definitions*

IEC 625-2 control functions and operations are classified using the source of control and method of annunciation.

5.6.1.1 IEC 625-2 remote operation

An IEC 625-2 remote operation is any operation of a device function in a **system** that is effected via program messages over the **system interface**.

5.6.1.2 IEC 625-2 local operation

An IEC 625-2 local operation is any operation of a device function that is not an IEC 625-2 remote operation.

> An example is the actuation of a user-accessible switch, knob, button, touch-screen location, etc. that is physically attached to a **device** and used to control or program the **device** locally.

Operation of a **device** via any other bus or interface connected to the **device** is considered to be local operation from the standpoint of the **system**.

5.6.1.3 Local control

A local control effects IEC 625-2 local operation of a **device**.

Local controls include **device** inputs which are designed for control.

Local controls and the accompanying remote/local requirements of this section are intended to apply only to local controls whose function can also be effected via IEC 625-2 remote operation.

Local controls whose function cannot be effected via IEC 625-2 remote operation are beyond the scope of this standard.

The following control functions are explicitly excluded from categorization as local controls.

1) Switching line power.
2) The generation of the URQ (user request) message.

5.6.1.4 External-control-signal

An external-control-signal invokes **device** actions using a device port other than the **system interface**. External-control-signals are beyond the scope of this standard but may optionally function as a local control. External-control-signals, however, shall not violate the message exchange protocols of clause 6.

> An example of an external-control-signal is a digital voltmeter's external trigger which initiates a measurement. The results may then be read from the device.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.6.1.5 Commande locale permanente

Les commandes locales permanentes sur un **appareil** ont des indicateurs (mécanique, positionnel, etc.) qui ne peuvent pas être changés par le fonctionnement à distance CEI 625-2 de l'**appareil**.

Par exemple, un instrument peut avoir un commutateur rotatif mécanique utilisé pour faire une sélection parmi trois modes de fonctionnement différents. L'état de ce commutateur est indiqué par des étiquettes imprimées sur la platine de l'**appareil**, mais les messages à distance ne peuvent pas faire tourner le commutateur.

5.6.1.6 Commande locale passagère

Toute commande locale qui n'est pas une commande locale permanente est une commande locale passagère.

Par exemple, une clé physique (contact momentané) sur un **appareil** peut être utilisée pour faire tourner une fonction de l'appareil pour la mettre en circuit ou hors circuit par des pressions alternées. L'état de marche/arrêt est indiqué par une lampe témoin positionnée par le commutateur. Cette lampe est contrôlée par l'état effectif de la fonction de l'appareil.

5.6.1.7 Commande locale programmable

Une commande locale d'une fonction d'un **appareil** pouvant être affectée par le fonctionnement à distance de la CEI 625-2 ainsi que par le fonctionnement local de la CEI 625-2 est une commande locale programmable. Les commandes locales programmables peuvent être des commandes locales permanentes ou passagères.

5.6.2 *Prescriptions de sous-ensembles CEI 625-1*

Un **appareil** capable d'un fonctionnement local et à distance en CEI 625-2 et utilisant des commandes locales programmables doit contenir le sous-ensemble RL1 (capacité complète y compris verrouillage local) de la fonction à distance ou locale CEI 625-1.

Un **appareil** incapable d'un fonctionnement local en CEI 625-2 ou n'utilisant pas les commandes locales programmables peut contenir le sous-ensemble RL0 (pas de capacité) de la fonction de commande à distance ou locale CEI 625-1. Un **appareil** contenant le RL0 ne tiendra pas compte de tous les messages d'interface CEI 625-1 relatifs aux changements d'état des fonctions à distance ou locale.

5.6.3 *Prescriptions de transition d'état de fonctionnement local à fonctionnement à distance*

La transition de LOCS (état local) ou LWLS (état local avec verrouillage) à REMS (état de fonctionnement à distance) ou RWLS (état de fonctionnement à distance avec verrouillage) doit invalider le fonctionnement local de toutes les commandes locales programmables.

5.6.4 *Prescriptions de transition d'état de fonctionnement à distance à fonctionnement local*

La transition du REMS ou RWLS au LOCS ou LWLS doit permettre le fonctionnement local de toutes les commandes locales programmables de l'**appareil**.

La transition doit également modifier toutes les fonctions de commande locale permanente de façon qu'elles puissent s'adapter à l'indicateur de commande locale permanente. Après la transition, la platine de l'**appareil** et l'état de commande doivent être en conformité.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.6.1.5 Hard local control

Hard local controls on a **device** have indicators (mechanical, positional, etc.) which cannot be changed by IEC 625-2 remote operation of the **device**.

For example, an instrument may have a mechanical rotary switch that is used to select among three different modes of operation. The state of this switch is indicated by printed labels on the front panel of the **device** but remote messages cannot rotate the switch.

5.6.1.6 Soft local control

Any local control that is not a hard local control is a soft local control.

For example, a physical (momentary contact) key on a **device** may be used to turn a device function on or off with alternate pushes. The on/off state is indicated by a light located by the switch. The light is controlled by the actual state of the device function.

5.6.1.7 Programmable local control

A local control of a **device** function that can be affected by IEC 625-2 remote operation as well as by IEC 625-2 local operation is a programmable local control. Programmable local controls may be hard or soft local controls.

5.6.2 *IEC 625-1 subset requirements*

A **device** that is capable of IEC 625-2 local and remote operation and utilizes programmable local controls shall contain the RL1 (complete capability including local lockout) subset of the IEC 625-1 Remote/Local function.

A **device** that is incapable of IEC 625-2 local operation or does not utilize programmable local controls may contain the RL0 (no capability) subset of the IEC 625-1 Remote/Local function. A **device** containing the RL0 subset shall ignore all IEC 625-1 interface messages related to Remote/Local function state changes.

5.6.3 *Local-to-remote state transition requirements*

The transition from LOCS (Local State) or LWLS (Local With Lockout State) to REMS (Remote State) or RWLS (Remote With Lockout State) shall disable local operation of all programmable local controls.

5.6.4 *Remote-to-local state transition requirements*

The transition from REMS or RWLS to LOCS or LWLS shall enable local operation of all programmable local controls of the **device**.

The transition shall also alter any hard local control functions so they match the hard local control indicator. After the transition, the **device's** front panel and state of control shall agree.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

La transition ne doit pas modifier une fonction de commande locale passagère.

Le réglage à la valeur VRAIE du message local rtl avec LLO faux ou ACDS (acceptation de l'état des données) inactif fera passer un **appareil** se trouvant en REMS en LOCS (voir CEI 625-1, 11.3.3.3).

5.6.5 *Fonctionnement à l'état local*

En LOCS ou LWLS, un **appareil** doit traiter tous les messages de programmation de façon à maintenir une cohérence entre l'information présentée à l'utilisateur local et l'état effectif de l'**appareil**.

Pour éviter une incohérence, un **appareil** devra exécuter l'une des opérations suivantes:

1) Corriger l'information incohérente présentée à l'utilisateur local.

2) Supprimer l'information incohérente.

3) Générer une erreur d'exécution et supprimer l'<UNITÉ DE MESSAGE DE PROGRAMMATION> responsable. (Voir protocoles d'indication d'état CEI 625-2 à l'article 11).

L'option 1 est l'option à choisir de préférence.

En LOCS ou LWLS, on ne doit pas empêcher un **appareil** d'envoyer un <MESSAGE DE RÉPONSE> en réponse à une interrogation préalablement reçue par l'**appareil**.

Lorsqu'on envoie à un **appareil** des messages de programmation alors qu'il est en mode local, des problèmes d'arbitrage peuvent éventuellement se poser si les commandes du panneau avant sont manipulées. Les programmes d'application peuvent éviter de tels problèmes en plaçant l'appareil en mode "à distance".

5.6.6 *Fonctionnement à l'état de fonctionnement à distance*

En REMS ou RWLS, un appareil doit traiter tous les messages de programmation.

En REMS ou RWLS, les messages dépendants de l'appareil (non définis par la présente norme) peuvent être envoyés pour permettre sélectivement des commandes locales spécifiques. Dans ce cas, les commandes locales spécifiées ne sont plus inopérantes localement. Si des commandes locales permanentes spécifiques sont validées, l'état de commande de l'**appareil** et ces commandes locales permanentes devront être en conformité.

En REMS ou RWLS, on ne doit pas empêcher un **appareil** d'envoyer un <MESSAGE DE RÉPONSE> en réponse à une interrogation préalablement reçue par l'**appareil**.

5.6.7 *Fonctionnement indépendant du mode de fonctionnement à distance ou local*

La fonction à distance ou locale doit être associée seulement avec la validation et l'invalidation des commandes locales, tel que cela est défini dans cette section. Les autres opérations de l'appareil, telles que la mise à jour de l'état de l'appareil et l'annonce associée, doivent fonctionner indépendamment de l'état de fonctionnement à distance ou local de l'**appareil**.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The transition shall not alter any soft local control functions.

Setting the rtl local message TRUE with LLO false or ACDS (Accept Data State) inactive shall cause a **device** in REMS to change to LOCS (see IEC 625-1, 11.3.3.3).

5.6.5 *Local state operation*

In LOCS or LWLS, a **device** shall process all program messages so as to maintain consistency between the information presented to the local user and the actual state of the **device**.

To avoid an inconsistency, a **device** shall do one of the following:

1) Correct the inconsistent information presented to the local user.
2) Remove the inconsistent information.
3) Generate an execution error and discard the responsible <PROGRAM MESSAGE UNIT>. (See IEC 625-2 Status reporting protocols in clause 11.)

Alternative 1 is the preferred choice.

In LOCS or LWLS a **device** shall not be inhibited from sending a <RESPONSE MESSAGE> in response to a query previously received by the **device**.

When a **device** is sent program messages while in local there are potential arbitration problems if front panel controls are manipulated. Application programs can avoid such problems by placing the device in remote.

5.6.6 *Remote state operation*

In REMS or RWLS, a **device** shall process all program messages.

In REMS or RWLS, device-specific messages (not defined by this standard) may be sent to selectively enable specific local controls. Under this circumstance, the specified local controls are no longer locally inoperative. If specific hard local controls are enabled, the **device's** state of control and these hard local controls shall agree.

In REMS or RWLS a **device** shall not be inhibited from sending a <RESPONSE MESSAGE> in response to a query previously received by the **device**.

5.6.7 *Operation independent of remote/local state*

The Remote/Local function shall be associated only with the enabling and disabling of local controls as defined in this section. Other device operation, such as updating the device state and associated local annunciation shall function independent of the remote/local state of the **device**.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.6.8 *Prescriptions pour l'indicateur à distance ou local*

Les **appareils** avec commandes locales permanentes doivent comporter un indicateur de fonctionnement "à distance" (CEI 625-1, annexe L).

Pour les **appareils** sans commandes locales permanentes, l'indicateur de fonctionnement à distance est facultatif. Tous les indicateurs concernant les transitions du mode de fonctionnement à distance au mode local doivent être étiquetés et fonctionner conformément aux exigences de CEI 625-1, annexe L.

5.7 *Prescriptions concernant l'interrogation parallèle*

Un **appareil** doit avoir soit le sous-ensemble PPO (pas de capacité), soit le sous-ensemble PP1 (configuration pour fonctionnement à distance). Un **appareil** avec sous-ensemble PP1 doit être conforme aux exigences de 11.6.

5.8 *Prescriptions d'appareils remis à zero*

Un **appareil** doit contenir le sous-ensemble DC1 (capacité complète y compris la remise à zéro sélective de l'appareil). Les exigences spécifiques CEI 625-2, pour la réponse de l'**appareil** en vue d'entrer dans l'état DCAS, sont spécifiées à l'article 6 et résumées ci-dessous.

Un **appareil** entrant en DCAS (état actif de remise à zéro), par l'intermédiaire d'une instruction DCL ou SDC,

doit:

1) Supprimer le tampon d'entrée et la file d'attente de sortie.

2) Remettre à zéro l'analyseur syntaxique, le contrôle d'exécution, et le formateur de réponse (voir 6.1.4.2.6).

3) Supprimer toute instruction empêchant le traitement d'une instruction *RST ou d'autres instructions de l'appareil.

4) Supprimer toutes les instructions et interrogations différées en raison de paramètres couplés (voir 6.4.5.3).

5) Mettre l'**appareil** en OCIS (Etat inactif d'instruction "opération achevée") et en OQIS (Etat inactif d'interrogation "opération achevée") (voir 12.5.2.1.1 et 12.5.3.1.1).

6) Arrêter les instructions communes *AAD et *DLF (voir l'article 13).

7) Mettre l'**appareil** dans l'état INACTIF d'échange de messages (voir 6.3.1.1).

ne doit pas:

1) Changer un réglage ou des données mémorisées dans l'**appareil**, sauf comme indiqué plus haut.

2) Interrompre le panneau avant E/S.

3) Interrompre ou affecter une opération quelconque en cours dans l'**appareil**, sauf comme indiqué plus haut.

4) Changer l'octet de l'état autre que l'effacement du bit MAV, par suite de l'effacement de la file d'attente de sortie, voir 11.2.1.2.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

5.6.8 *Remote/local indicator requirements*

Devices with hard local controls shall incorporate a "Remote" indicator (IEC 625-1, annex L).

For **devices** without hard local controls, the Remote indicator is optional. All indicators relating to remote/local transitions shall be labelled and function in accordance with the requirements of IEC 625-1, annex L.

5.7 *Parallel Poll requirements*

A **device** shall have either the PP0 (no capability) or the PP1 (remote configuration) subset. A **device** with the PP1 subset shall conform to the requirements of 11.6.

5.8 *Device Clear requirements*

A **device** shall contain the DC1 (complete capability including selective device clear) subset. Specific IEC 625-2 requirements for the **device's** response to entering DCAS are specified in clause 6 and summarized below.

A **device** entering DCAS (Device Clear Active State) via a DCL or SDC command

Shall:

1) Clear the Input Buffer and Output Queue.

2) Reset the Parser, Execution Control, and Response Formatter (see 6.1.4.2.6).

3) Clear any command that prevents processing a *RST or other device commands.

4) Discard all commands and queries deferred due to coupled parameters (see 6.4.5.3).

5) Put the **device** into OCIS (Operation Complete Command Idle State) and into OQIS (Operation Complete Query Idle State) (see 12.5.2.1.1 and 12.5.3.1.1).

6) Abort the *AAD and *DLF common commands (see clause 13).

7) Put the **device** in the message exchange IDLE state (see 6.3.1.1).

Shall not:

1) Change any settings or stored data in the **device** except as stated above.

2) Interrupt front panel I/O.

3) Interrupt or affect any **device** operation in progress except as stated above.

4) Change the status byte other than clearing the MAV bit as a consequence of clearing the Output Queue, see 11.2.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.9 *Prescriptions de déclenchement de l'appareil*

Un **appareil** peut contenir soit le DT0 (pas de capacité), soit le DT1 (capacité totale) de la fonction de déclenchement de l'appareil CEI 625-1. Un **appareil** contenant le sous-ensemble DT1 doit mettre en application l'instruction commune associée, *TRG comme cela est défini en 10.37. L'utilisation de l'instruction commune *DDT facultative (voir 10.4), a une influence sur le fonctionnement de *TRG et le Déclenchement d'Exécution de Groupe (GET).

5.10 *Prescriptions de fonction du contrôleur*

Un **appareil** doit, soit ne contenir aucune capacité de contrôleur (C0), soit contenir une fonction de contrôleur avec les options de sous-ensembles suivantes: C4 avec C5, C7, C9 ou C11. Ces sous-ensembles fournissent la capacité de répondre à SRQ, d'envoyer des messages IF, de recevoir le contrôle, de faire passer le contrôle et de prendre le contrôle d'une manière synchrone. Ils interdisent spécifiquement la capacité de contrôleur du système, c'est-à-dire l'envoi de messages d'interface IFC et REN. Facultativement, ils permettent la capacité de faire passer le contrôle à eux-mêmes et de faire une interrogation parallèle.

Si un **appareil** contient un sous-ensemble de contrôleur autre que C0, il doit également être capable de transmettre et de recevoir le contrôle au moyen des protocoles décrits dans 17.4 et 17.5.

Voir l'article 15 pour les exigences du **contrôleur** (par opposition à l'**appareil**).

5.11 *Prescriptions électriques*

L'implémentation de l'option d'interface électrique E2 (CEI 625-1, annexe C) est recommandée. Des circuits de commande à collecteurs ouverts doivent être utilisés pour attaquer les lignes de signalisation SRQ, NRFD et NDAC. Il est recommandé d'utiliser des circuits de commande à trois états pour attaquer les lignes de signalisation DAV, EOI et ATN. Un **appareil** n'est pas autorisé à attaquer REN ou IFC. Si l'**appareil** est en PPAS (état actif d'interrogation parallèle), l'**appareil** doit utiliser des circuits de commande à collecteurs ouverts pour attaquer les lignes de signalisation DIO1-8. Lorsqu'il n'est pas en PPAS, l'**appareil** devrait utiliser des circuits de commande à trois états pour attaquer les lignes de signalisation DIO1-8. Si un **appareil** a besoin de fournir des octets de données d'établissement de liaison à une cadence supérieure à 250 000 octets par seconde, l'option E2 est nécessaire, CEI 625-1, article 31.

NOTE – Certains circuits intégrés peuvent nécessiter un matériel auxiliaire pour éviter d'attaquer les lignes DIO pendant une interrogation parallèle, même si le sous-ensemble PP0 est implémenté.

5.12 *Prescriptions de mise sous tension*

A la mise sous tension, les réglages de l'**appareil** doivent être rétablis à leurs valeurs lorsque l'**appareil** a été en dernier lieu mis hors circuit, réglés à des états connus explicitement indiqués dans la documentation concernant l'**appareil**, ou réglés à un état défini par l'utilisateur et stockés dans une mémoire locale rémanente.

Les instructions spécifiques à l'appareil peuvent être fournies pour sélectionner soit le rétablissement des réglages de l'appareil, soit leur réglage à un état connu à la mise sous tension. Cette norme définit une telle instruction, effacement d'état à la mise sous tension, (*PSC, voir 10.25) qui règle et efface l'indicateur d'effacement d'état à la mise sous tension.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.9 *Device Trigger requirements*

A **device** may contain either the DT0 (no capability) or the DT1 (complete capability) IEC 625-1 device trigger function. A **device** containing the DT1 subset shall implement the associated common command, *TRG as defined in 10.37. Use of the optional *DDT common command (see 10.4) affects the operation of *TRG and Group Execute Trigger (GET).

5.10 *Controller function requirements*

A **device** shall contain either no Controller function capability (C0) or contain a Controller function with the following subset options: C4 with C5, C7, C9, or C11. These subsets provide the capability to respond to SRQ, send IF messages, receive control, pass control, and take control synchronously. They specifically prohibit system controller capability, i.e., sending the IFC and REN interface messages. They optionally permit the capability of passing control to self and conducting a parallel poll.

If a **device** contains any Controller function subset other than C0, then it shall also be able to pass and receive control by means of the protocols described in 17.4 and 17.5.

See clause 15 for **controller** (as contrasted to **device**) requirements.

5.11 *Electrical requirements*

Implementation of the E2 electrical interface option (IEC 625-1, annex C) is recommended. Open collector drivers shall be used to drive the SRQ, NRFD, and NDAC signal lines. Three-state drivers should be used to drive the DAV, EOI, and ATN signal lines. A **device** is not allowed to drive REN or IFC. If the **device** is in PPAS (Parallel Poll Active State), the **device** shall use open collector drivers to drive the DIO1-8 signal lines. When not in PPAS, the **device** should use three-state drivers to drive the DIO1-8 signal lines. If a **device** needs to source handshake data bytes at a rate greater than 250 000 bytes per second, the E2 option is required, IEC 625-1 clause 31.

NOTE – Some integrated circuits may require external hardware to avoid driving the DIO lines during a parallel poll, even though the PP0 subset is implemented.

5.12 *Power-on requirements*

At power-on, **device** settings shall be either restored to their values when the **device** was last powered off, set to known states which are explicitly stated in the **device** documentation, or set to a state defined by the user and stored in local non-volatile memory.

Device-specific commands may be provided to select whether device settings are restored or set to a known state at power-on. This standard defines one such command, Power-on Status Clear, (*PSC, see 10.25) which sets and clears the power-on-status-clear flag.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.12.1 *Articles non affectés par la mise sous tension*

Un **appareil** ne doit pas modifier les paramètres suivants du fait d'une mise sous tension seulement (réception du message local de mise sous tension CEI 625-1).

- L'adresse transitant par le bus.
- Les données appropriées de calibrage.
- Les données ou les états de l'appareil causant un changement dans la réponse aux interrogations communes suivantes:

  *IDN? (Interrogation d'identification, voir 10.14)

  *OPT? (Interrogation d'identification de l'option, voir 10.20)

  *PSC? (Interrogation d'effacement d'état à la mise sous tension, voir 10.26)

  *PUD? (Interrogation sur les données de l'utilisateur protégées, voir 10.28)

  *RDT? (Interrogation sur le transfert de la description des ressources, voir 10.31)

5.12.2 *Articles dépendant de l'indicateur d'effacement d'état à la mise sous tension*

Si l'indicateur d'effacement d'état à la mise sous tension est FAUX, le registre de validation de demande de service, le registre de validation d'état d'événement standard et le registre de validation d'interrogation parallèle ne doivent pas être affectés par la mise sous tension. Voir instruction *PSC, article 10.25.

Un **appareil** doit effacer le registre de validation de demande de service, le registre de validation d'état d'événement standard et le registre de validation d'interrogation parallèle à la mise sous tension si l'indicateur d'effacement d'état à la mise sous tension est VRAI ou si l'instruction *PSC n'est pas implémentée.

5.12.3 *Articles pouvant être affectés par la mise sous tension*

A la discrétion du concepteur de l'appareil, un **appareil** peut modifier les paramètres suivants à la mise sous tension:

- Etats des fonctions de l'appareil en vigueur.
- Informations sur l'état.
- Registres *SAV/*RCL, voir 10.33 et 10.29.
- Définition d'une macro définie avec l'instruction *DDT, voir 10.4.
- Définitions de macros définies avec l'instruction *DMC, voir 10.7.
- Macro validant avec l'instruction *EMC, voir 10.8.
- Adresse reçue avec la dernière instruction *PCB, voir 10.21.

6 Protocole de contrôle des échanges de messages

Le protocole d'échange de messages de l'appareil spécifie comment un **appareil** traite les messages de programmation et de réponse.

Le terme "**contrôleur**" désigne l'élément du système qui échange des messages avec l'**appareil**. Les prescriptions de cette section s'appliquent également à la communication d'un appareil à un autre.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5.12.1 *Items not affected by power-on*

A **device** shall not alter the following due solely to a power-on (receipt of the IEC 625-1 pon local message).

- The bus address.
- Pertinent calibration data.
- Data or device states which cause a change in the response to the following common queries:

  *IDN? (Identification Query, see 10.14)

  *OPT? (Option Identification Query, see 10.20)

  *PSC? (Power-on Status Clear Query, see 10.26)

  *PUD? (Protected User Data Query, see 10.28)

  *RDT? (Resource Description Transfer Query, see 10.31)

5.12.2 *Items dependent upon power-on-status-clear flag*

If the power-on-status-clear flag is FALSE, the Service Request Enable Register, the Standard Event Status Enable Register, and the Parallel Poll Enable Register shall not be affected by power-on. See *PSC command, clause 10.25.

A **device** shall clear the Service Request Enable Register, the Standard Event Status Enable Register, and the Parallel Poll Enable Register at power-on if the power-on-status-clear flag is TRUE or if the *PSC command is not implemented.

5.12.3 *Items that may be affected by power-on*

At the discretion of the device designer, a **device** may alter the following at power-on.

- Current device function states.
- Status information.
- *SAV/*RCL registers, see 10.33 and 10.29.
- Macro definition defined with the *DDT command, see 10.4.
- Macro definitions defined with the *DMC command, see 10.7.
- Macro enabling with the *EMC command, see 10.8.
- Address received with last *PCB command, see 10.21.

6 Message exchange control protocol

The device message exchange protocol specifies how a **device** handles program and response messages.

The term "controller" refers to the system element that is exchanging messages with the **device**. The requirements of this clause also apply to device-to-device communication.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

La phrase ". . . l'**appareil** (ne) doit (pas) . . ." définit une prescription de cette norme. La phrase ". . . il est recommandé que le **contrôleur** (ne) . . . (pas). . ." s'applique à des conditions qui (n') interviendront (pas) dans des conditions normales. La réponse de l'**appareil** aussi bien à des conditions normales qu'exceptionnelles est définie par cette norme.

Le terme "message d'interrogation", tel qu'il est utilisé dans la présente section, est un <MESSAGE DE PROGRAMMATION> contenant un ou plusieurs éléments d'<UNITÉ DE MESSAGE D'INTERROGATION>. Noter que tous les éléments syntaxiques entre parenthèses sont définis en 7 et 8.

Les termes "erreur d'instruction", "erreur d'exécution", "erreur d'interrogation", et "registre d'état d'événement standard" sont définis à l'article 11.

6.1 *Eléments fonctionnels*

La figure 6-1 présente les relations entre le bus CEI 625-1, l'interface d'échange de messages décrits dans cet article, les fonctions d'indication d'état décrites à l'article 11, et les fonctions spécifiques à l'appareil prévues par le concepteur de l'appareil.

6.1.1 *Bus CEI 625-1*

Le bus CEI 625-1, indiqué dans les figures 6-1 et 6-2, représente le câble d'interface physique CEI 625-1 et les circuits d'attaque, tels que définis dans la CEI 625-1.

6.1.2 *Indication d'état*

Le bloc d'indication d'état apparaît seulement à la figure 6-1. La figure 6-2, toutefois, visualise bien les messages d'état envoyés à partir de l'interface d'échange de messages et les messages envoyés au contrôle E/S.

Le bloc d'indication d'état reçoit des messages d'erreur et d'état provenant de l'interface d'échange de messages et des blocs de fonctions de l'appareil. Il envoie également les messages STB, **reqt**, **reqf**, et CEI 625-1 ist au contrôle E/S, de sorte qu'il puisse répondre aux demandes d'interrogation série et d'interrogation parallèle CEI 625-1. Le bloc d'indication d'état est entièrement décrit à l'article 11.

6.1.3 *Interface d'échange de messages*

L'interface d'échange de messages traite tous les messages dépendants de l'appareil CEI 625-1 envoyés à l'**appareil** et en provenance de l'**appareil**.

La figure 6-2 présente un modèle logique décrivant le fonctionnement de l'interface d'échange de messages de l'**appareil**. Son objectif n'est pas d'impliquer une implémentation physique particulière. On n'a pas représenté certains signaux d'établissement de liaison entre les blocs, ceci afin de réduire la complexité du schéma.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The phrase ". . . the **device** shall (not) . . ." defines a requirement of this standard. The phrase ". . . the **controller** should (not) . . ." refers to conditions which will (not) happen under normal conditions. The **device's** response to both normal and exceptional conditions is defined by this standard.

The term "Query Message", as used in this clause, is a <PROGRAM MESSAGE> which contains one or more <QUERY MESSAGE UNIT> elements. Note that all bracketed syntactic elements are defined in clauses 7 and 8.

The terms "Command Error", "Execution Error", "Query Error", and "Standard Event Status Register" are defined in clause 11.

6.1 *Functional elements*

Figure 6-1 presents the relationship among the IEC 625-1 bus, the Message Exchange Interface described in this clause, the Status Reporting functions described in clause 11, and the device-specific functions provided by the device designer.

6.1.1 *IEC 625-1 bus*

The IEC 625-1 bus, shown in both figures 6-1 and 6-2, represents the physical IEC 625-1 interface cable and drivers as defined in IEC 625-1.

6.1.2 *Status reporting*

The status reporting block appears only in figure 6-1. Figure 6-2 does, however, display the status messages sent from the Message Exchange Interface and the messages sent to the I/O Control.

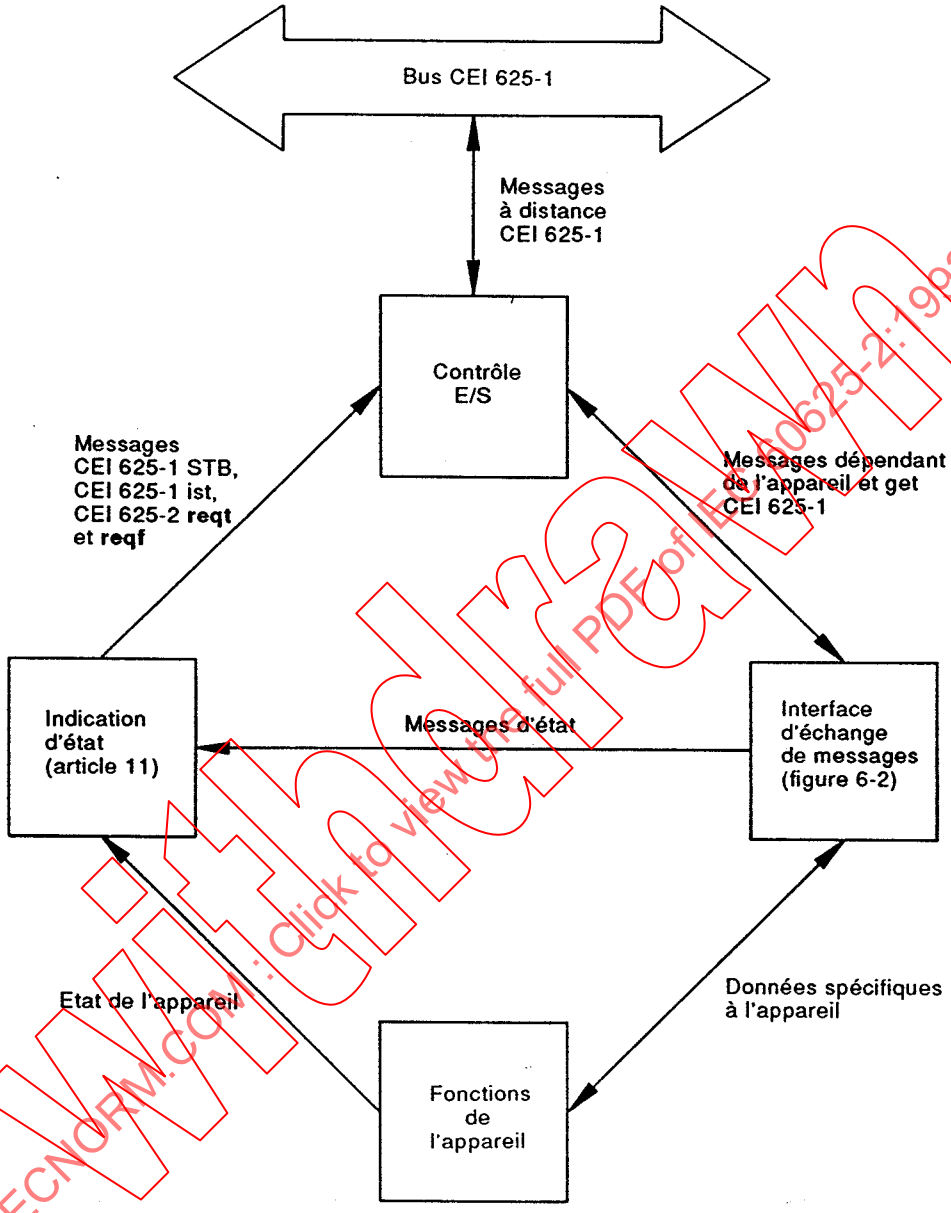
The status reporting block receives error and status messages from the Message Exchange Interface and Device Functions blocks. It also sends the STB, **reqt**, **reqf**, and IEC 625-1 ist messages to the I/O Control, so that it can respond to IEC 625-1 Serial Poll and Parallel Poll requests. The Status Reporting block is completely described in clause 11.

6.1.3 *Message Exchange Interface*

The Message Exchange Interface deals with all IEC 625-1 device-dependent messages to and from the **device**.

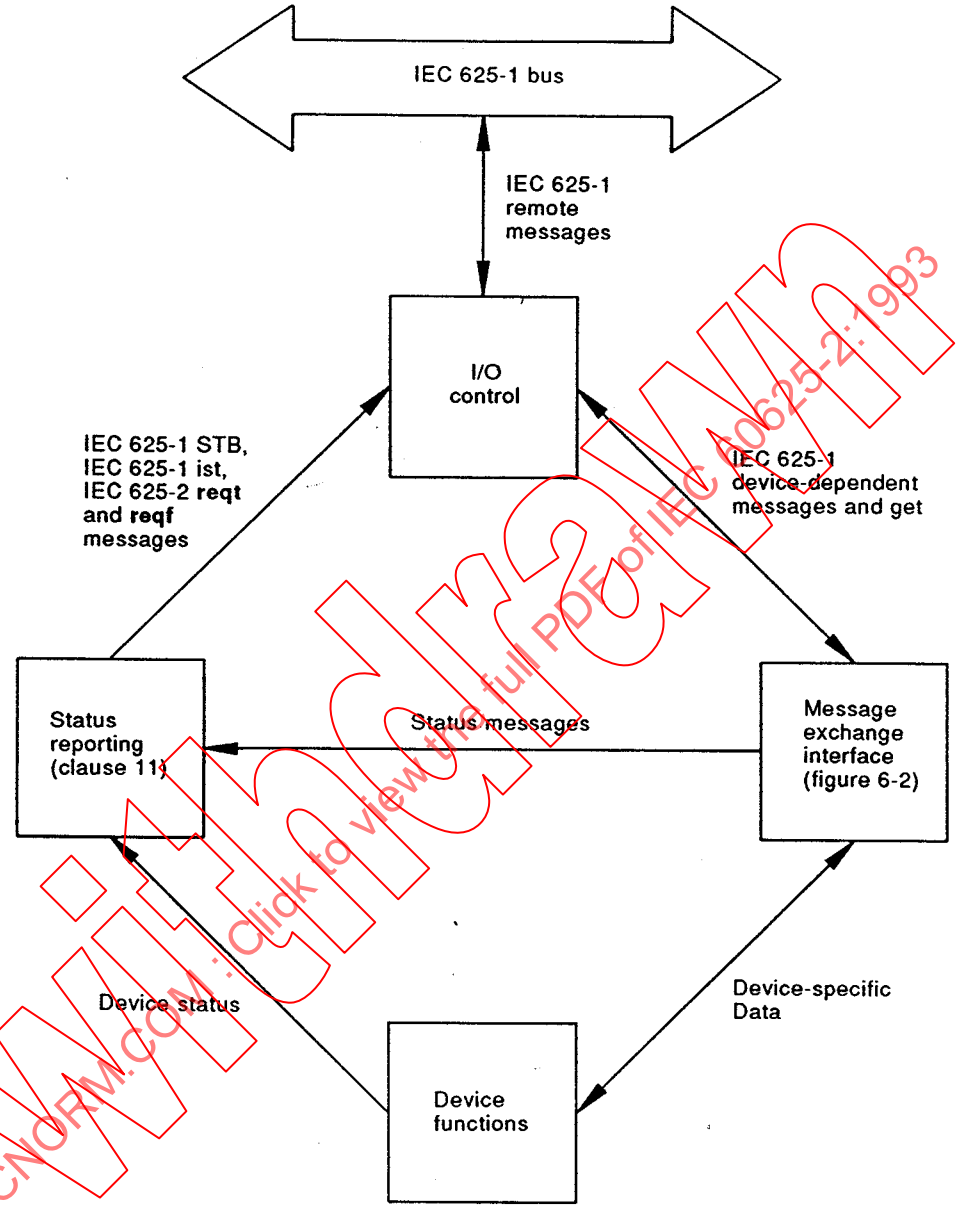
Figure 6-2 presents a logical model which describes the operation of the **device**'s Message Exchange Interface. It is not intended to imply a particular physical implementation. Some handshaking signals between blocks have been omitted to reduce the complexity of the diagram.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Figure 6-1 – Etat de l'appareil et vue d'ensemble des échanges de messages



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figure 6-1 – Device Status and Message Exchange overview

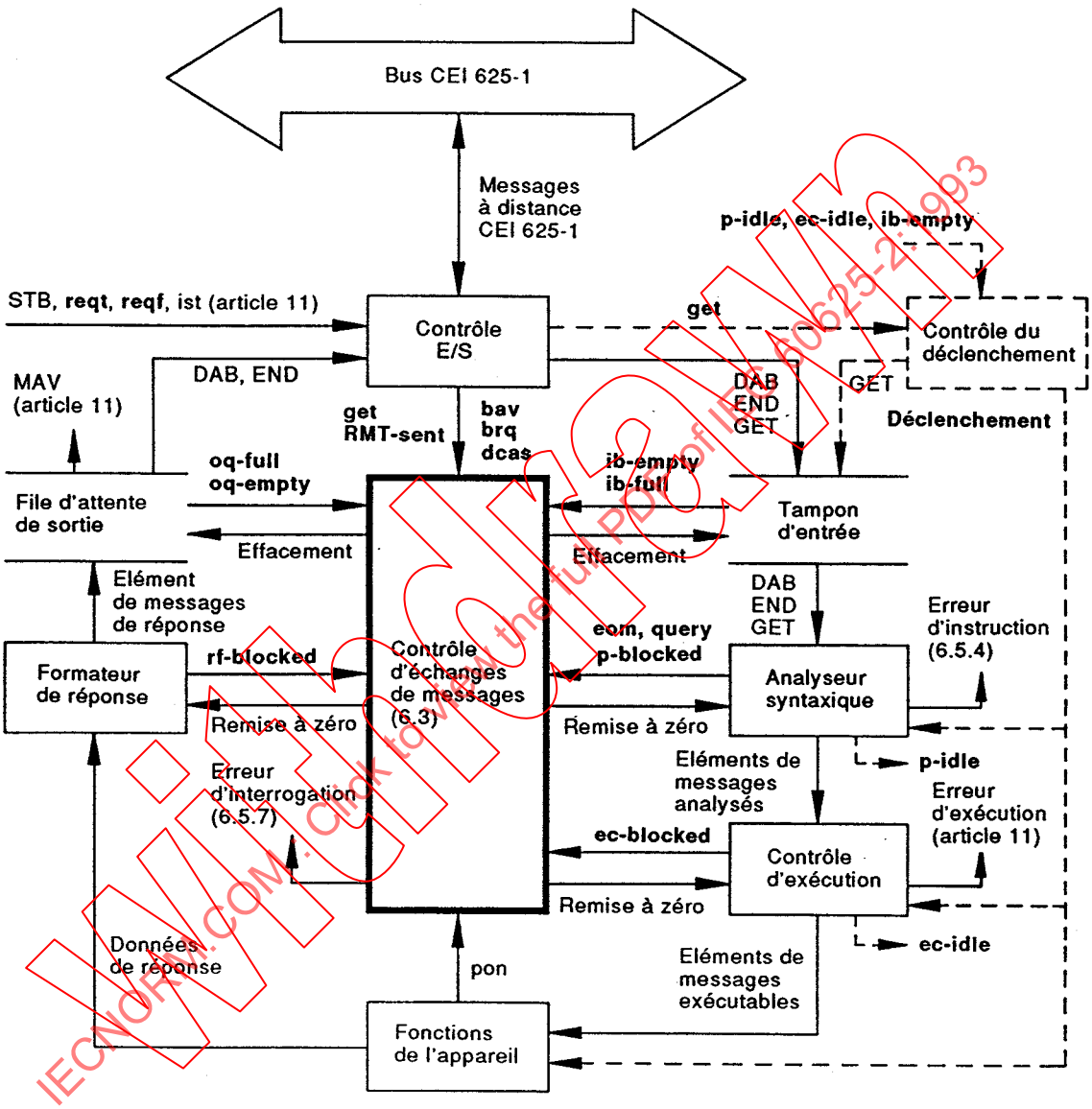


Figure 6-2 – Interface de contrôle d'échanges de messages, blocs fonctionnels

6.1.4 *Contrôle E/S*

Le contrôle E/S, représenté aux figures 6-1 et 6-2, interprète et décode les messages à distance CEI 625-1. Un grand nombre des fonctions de ce bloc sont fournies par des circuits intégrés commerciaux qui ont été conçus pour implémenter les exigences de la CEI 625-1.

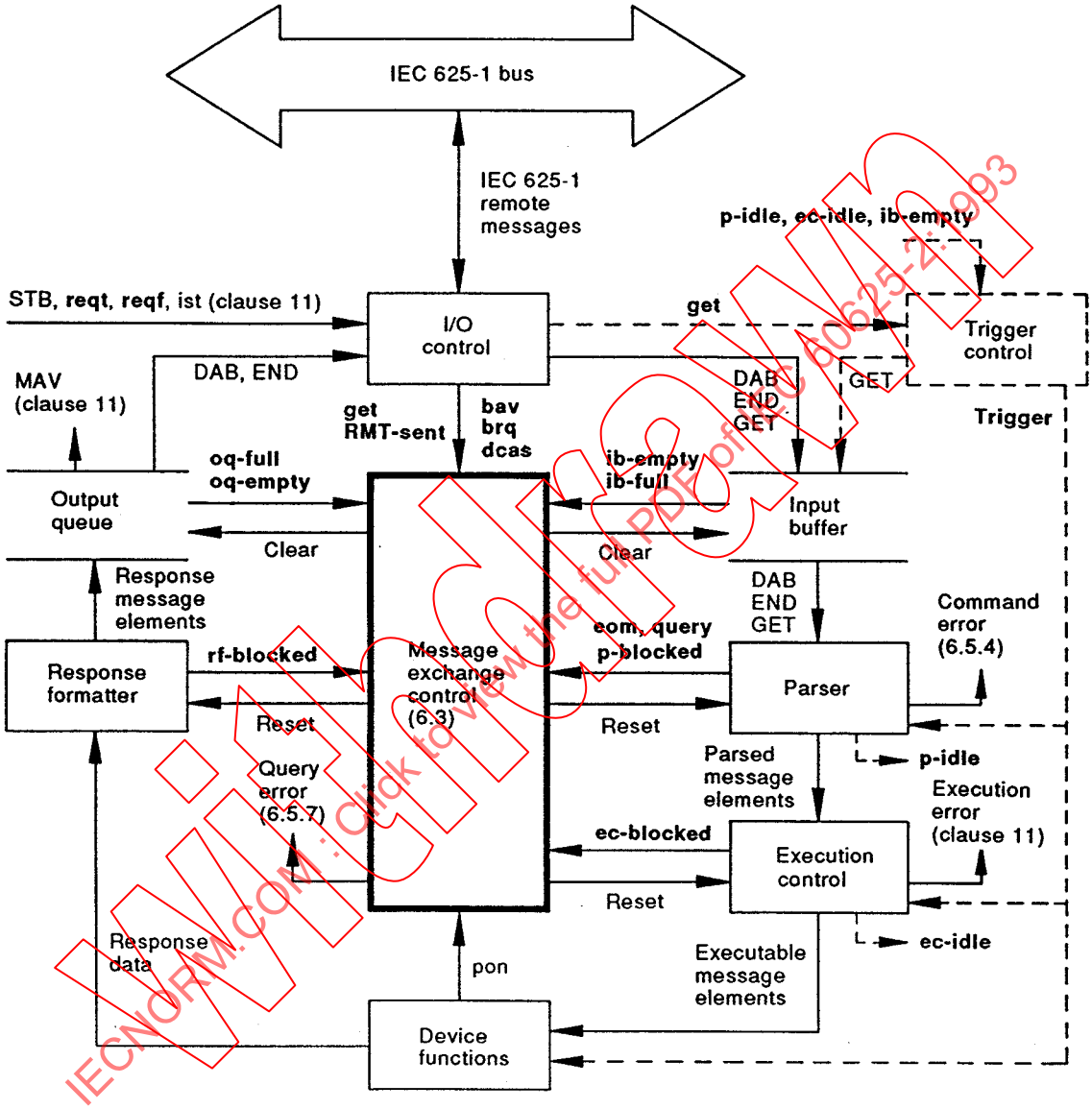


Figure 6-2 – Message Exchange Control Interface functional blocks

6.1.4 *I/O Control*

The I/O Control, shown in figures 6-1 and 6-2, interprets and decodes IEC 625-1 remote messages. Many of the functions of this block are provided by commercial integrated circuits that have been designed to implement the requirements of IEC 625-1.

Le contrôle E/S répond aux demandes d'interrogation série et d'interrogation parallèle CEI 625-1, conformément aux messages STB, **reqt**, **reqf**, et ist du bloc d'indication d'état, article 11. Il génère les messages **dcas**, DAB, END, **get**, **bav**, et **brq** qui sont envoyés à l'interface d'échanges de messages, et reçoit les messages DAB et END de l'interface d'échanges de messages.

6.1.4.1 Règles du contrôle E/S

6.1.4.1.1 Contrôle E/S envoyant des octets de données

Lorsque **brq** est VRAI et **oq-empty** est FAUX, le contrôle E/S doit enlever un octet de la file d'attente de sortie, et doit mettre **brq** FAUX, et doit envoyer l'octet au contrôleur en utilisant la fonction SH (établissement d'une liaison avec la source) CEI 625-1. Voir 6.1.4.2.3.

6.1.4.1.2 Contrôle E/S recevant des octets de données

Lorsque **bav** est VRAI et **ib-full** est FAUX, le contrôle E/S doit mettre **bav** FAUX, puis doit placer le DAB associé et tout message END d'accompagnement dans le tampon d'entrée en utilisant la fonction AH (établissement d'une liaison avec l'accepteur) CEI 625-1. Voir 6.1.4.2.4.

6.1.4.1.3 Contrôle E/S recevant le message GET CEI 625-1

Lorsque **get** est VRAI et **ib-full** est FAUX, le contrôle E/S doit mettre **get** FAUX, puis doit placer un message GET dans le tampon d'entrée.

NOTE - Si le contrôle de déclenchement facultatif est implémenté, le message **get** doit être envoyé au contrôle de déclenchement plutôt qu'au tampon d'entrée. Le **get** peut être exécuté directement dans certaines conditions, voir 6.1.11.

6.1.4.1.4 Classement des messages reçus par le contrôle E/S

Le contrôle E/S doit maintenir l'ordre des messages reçus, y compris octets de données (DAB's), messages END, déclenchement d'exécution de groupe, remise à zéro de l'appareil, et remise à zéro sélectionnée de l'appareil.

L'exemple suivant montre une technique de conception pour maintenir l'ordre des octets de données reçus et les messages d'interface à distance CEI 625-1: DCL, SDC et GET.

Concevoir l'interface de messages de l'**appareil** pour l'opération de déclenchement de l'appareil comme suit:

Concevoir la fonction AH CEI 625-1 de sorte qu'elle attende en ACDS (état d'acceptation des données) lorsque la fonction DT (déclenchement de l'appareil) entre en DTAS (état actif de déclenchement de l'appareil) après avoir reçu un GET et jusqu'à ce que le contrôle E/S mette le message **get** VRAI. Cette action est souvent appelée "blocage du bus NDAC".

Faire mettre **bav** FAUX par le contrôle E/S et placer l'octet de données dans le tampon d'entrée avant d'accepter le message GET lorsque les messages **bav** et **get** sont tous les deux VRAIS. Etant donné que le nouvel octet de données associé avec le message **bav** a dû être reçu en premier, le bus étant en état de blocage pendant la réception du message GET, un séquencement approprié est garanti.

S'assurer que le contrôle E/S accepte le message GET (met **get** VRAI) avant qu'il libère le blocage du bus et permette à la fonction AH CEI 625-1 de l'appareil d'accepter un nouvel octet de données.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The I/O Control responds to IEC 625-1 Serial Poll and Parallel Poll requests as directed by the STB, **reqt**, **reqf**, and ist messages from the status reporting block, clause 11. It generates the **dcas**, DAB, END, **get**, **bav**, and **brq** messages which are sent to the Message Exchange Interface, and receives the DAB and END messages from the Message Exchange Interface.

6.1.4.1 I/O Control rules

6.1.4.1.1 I/O Control sending data bytes

When **brq** is TRUE and **oq-empty** is FALSE, the I/O Control shall remove one byte from the Output Queue, set **brq** FALSE, and send the byte to the controller using the IEC 625-1 SH (Source Handshake) function. See clause 6.1.4.2.3.

6.1.4.1.2 I/O Control receiving data bytes

When **bav** is TRUE and **ib-full** is FALSE, the I/O Control shall set **bav** FALSE and then place the associated DAB and any accompanying END message into the Input Buffer using the IEC 625-1 AH (Acceptor Handshake) function. See clause 6.1.4.2.4.

6.1.4.1.3 I/O Control receiving IEC 625-1 GET message

When **get** is TRUE and **ib-full** is FALSE, the I/O Control shall set **get** FALSE and then place a GET message in the Input Buffer.

NOTE - If the optional Trigger Control is implemented, the **get** message shall be sent to the Trigger Control rather than the Input Buffer. The **get** can be executed directly under certain conditions, see 6.1.11.

6.1.4.1.4 I/O Control ordering of received messages

The I/O control shall preserve the order of received messages, including data bytes (DAB's), END messages, Group Execute Trigger, Device Clear, and Selected Device Clear.

The following example shows a design technique for preserving the order of received data bytes and the IEC 625-1 remote interface messages: DCL, SDC, and GET.

Design the **device's** message interface for Device Trigger handling as follows:

Design the IEC 625-1 AH function to wait in ACDS (Accept Data State) when the DT (Device Trigger) function enters the DTAS (Device Trigger Active State) after a GET is received and until the I/O Control sets the **get** message TRUE. This action is often called "NDAC bus holdoff".

Have the I/O Control set **bav** FALSE and place the data byte in the Input Buffer before accepting the GET message when the **bav** and **get** messages are both TRUE. Since the new data byte associated with the **bav** message must have been received first with the bus being in a holdoff state during the receipt of the GET message, proper sequencing is guaranteed.

Ensure the I/O Control accepts the GET message (sets **get** TRUE), before it releases the bus holdoff and allows the **device's** IEC 625-1 AH function to accept a new data byte.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Concevoir l'interface de messages de l'appareil pour l'opération de remise à zéro de l'appareil comme suit:

Concevoir la fonction AH CEI 625-1 de sorte qu'elle attende en ACDS lorsque la fonction DC (remise à zéro de l'appareil) entre en DCAS (état actif de remise à zéro de l'appareil) jusqu'à ce que le contrôle E/S mette le message **dcas** VRAI.

S'assurer que le contrôle E/S met **bav** FAUX et supprime le DAB avant qu'il libère le blocage du bus et permette à la fonction AH de l'**appareil** d'accepter un nouvel octet de données. Etant donné que le nouvel octet de données associé avec le message **bav** a dû être reçu en premier, le bus étant en blocage pendant la réception des messages DCL ou SDC, un séquencement approprié est garanti.

6.1.4.2 Messages du contrôle E/S

Ce paragraphe décrit l'action des messages locaux CEI 625-2 du contrôle E/S.

6.1.4.2.1 Message d'octet de données (DAB)

Le message DAB est défini dans CEI 625-1. Il est composé de huit bits qui sont soit envoyés, soit reçus par l'**appareil** pendant que ATN est FAUX.

6.1.4.2.2 Message de fin (END)

Le message END est défini dans CEI 625-1. Il est envoyé avec un DAB en mettant EOI VRAI et ATN FAUX.

6.1.4.2.3 Message de demande d'octet (brq)

Le message **brq** indique que le contrôle E/S est prêt à envoyer un octet de données au **contrôleur**.

Le contrôle E/S doit mettre **brq** VRAI une fois que la fonction SH CEI 625-1 est entrée dans l'état de génération de source (SGNS) CEI 625-1 et que la fonction T ou TE est dans l'état actif parleur (TACS). Le message **brq** doit être mis VRAI une fois, et une fois seulement, pour chaque passage en SGNS.

Le contrôle E/S doit mettre **brq** FAUX lorsque l'une des conditions suivantes se produit:

1) Le contrôle E/S envoie un octet de données de la file d'attente de sortie au **contrôleur** (voir 6.1.4.1.1).

2) L'**appareil** exécute la procédure "INTERROMPU" décrite en 6.3.2.3.

3) L'**appareil** exécute la procédure "NON TERMINÉ" décrite en 6.3.2.2.

4) Le contrôle E/S met le message **dcas** VRAI.

5) Le contrôle d'échanges de messages entre dans l'état INACTIF à partir des états EXÉCUTÉ ou BLOQUÉ.

Aucune autre condition de l'appareil ou d'interface ne doit rendre **brq** FAUX.

6.1.4.2.4 Message d'octet disponible (bav)

Le message **bav** indique que le contrôle E/S a reçu un octet de données et est prêt à le placer dans le tampon d'entrée.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Design the device's message interface for Device Clear handling as follows:

Design the IEC 625-1 AH function to wait in ACDS when the DC (Device Clear) function enters the DCAS (Device Clear Active State) until the I/O Control sets the **dcas** message TRUE.

Assure the I/O Control sets **bav** FALSE and discards the DAB before it releases the bus holdoff and allows the **device's** AH function to accept a new data byte. Since the new data byte associated with the **bav** message must have been received first with the bus being in a holdoff state during the receipt of the DCL or SDC messages, proper sequencing is guaranteed.

6.1.4.2 I/O Control messages

This subclause describes the action of the I/O Control IEC 625-2 local messages.

6.1.4.2.1 Data byte message (DAB)

The DAB message is defined in IEC 625-1. It consists of eight bits that are either sent or received by the **device** while ATN is FALSE.

6.1.4.2.2 End message (END)

The END message is defined in IEC 625-1. It is sent with a DAB by setting EOI TRUE and ATN FALSE.

6.1.4.2.3 Byte requested message (brq)

The **brq** message indicates the readiness of the I/O Control to send a data byte to the **controller**.

The I/O Control shall set **brq** TRUE after the IEC 625-1 SH function enters the IEC 625-1 Source Generate state (SGNS) and the T or TE function is in the Talker Active State (TACS). The **brq** message shall be set TRUE once and only once for each transition into SGNS.

The I/O Control shall set **brq** FALSE when any of the following conditions occur:

1) The I/O Control sends a data byte from the Output Queue to the **controller** (see clause 6.1.4.1.1).
2) The **device** performs the INTERRUPTED action described in 6.3.2.3.
3) The **device** performs the UNTERMINATED action described in 6.3.2.2.
4) The I/O Control sets the **dcas** message TRUE.
5) The Message Exchange Control enters the IDLE state from the DONE or DEADLOCK states.

No other device or interface condition shall cause **brq** to be set FALSE.

6.1.4.2.4 Byte available message (bav)

The **bav** message indicates that the I/O Control has received a data byte and is ready to place it in the Input Buffer.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Le contrôle E/S doit mettre **bav** VRAI une fois que la fonction L ou LE CEI 625-1 est dans l'état actif écouteur (LACS) et que la fonction d'établissement de liaison avec l'accepteur CEI 625-1 est entrée dans l'état d'acceptation des données (ACDS). Le message **bav** doit mettre VRAI une fois, et une fois seulement, à chaque passage de l'état accepteur prêt (ACRS) à ACDS.

Le contrôle E/S doit mettre **bav** FAUX dans l'une des conditions suivantes:

1) Un octet de données est placé dans le tampon d'entrée. (Si l'**appareil** a un tampon d'entrée de longueur zéro, le contrôle E/S mettra **bav** FAUX lorsque l'octet de données est envoyé à l'analyseur syntaxique.) Voir 6.1.4.1.2.

2) Le message **dcas** est mis VRAI. L'octet de données associé doit être supprimé.

Aucune autre condition de l'appareil ou d'interface ne doit rendre **bav** FAUX. Voir 6.1.4.1.3.

6.1.4.2.5 Message de déclenchement d'exécution de groupe (get)

Le message **get** CEI 625-2 signale le passage de la fonction de déclenchement de l'appareil (DT) CEI 625-1 en DTAS (état actif de déclenchement de l'appareil). Cette transition se produit lorsqu'un message d'interface à distance GET (Déclenchement d'Exécution de Groupe) CEI 625-1 est reçu alors que l'**appareil** était adressé pour écouter. Le message **get** peut être utilisé pour lancer une action définie par l'appareil ou l'action définie par l'instruction commune *DDT, voir 10.4.

L'**appareil** doit traiter les octets de données et les messages GET dans l'ordre reçu, voir 6.1.4.1.4.

La norme spécifie deux méthodes pour le déclenchement de l'appareil, l'instruction commune *TRG, 10.37, et le message d'interface GET CEI 625-1. D'une manière typique, le déclenchement de l'appareil est utilisé dans deux buts distincts:

1) Pour le lancement correctement séquencé d'une action définie par l'appareil ou définie par l'utilisateur dans le cadre d'un appareil à une adresse.

2) Pour le lancement synchronisé d'actions définies par l'appareil ou définies par l'utilisateur dans le cadre d'appareils à adresses multiples.

Pour l'objectif 1, on peut utiliser soit GET, soit *TRG pour le déclenchement de l'appareil. Le stockage de l'un ou l'autre dans le tampon d'entrée garantit que le déclenchement sera exécuté dans la séquence correcte par rapport aux autres instructions.

Pour l'objectif 2, où la synchronisation demandée pourraît être compromise par le système de traitement logiciel/micrologiciel, le message **get** peut être utilisé conjointement avec le bloc de contrôle de déclenchement facultatif. **get** générera un message local de **déclenchement** pour exécution immédiate par l'**appareil** à condition que le tampon d'entrée soit vide et que l'analyseur syntaxique et le contrôle d'exécution soient inactifs. Si le tampon d'entrée n'est pas vide, le message GET doit être placé dans le tampon d'entrée, et l'exécution du déclenchement différée jusqu'à ce que les instructions préalables aient été exécutées.

Si le bloc de contrôle de déclenchement, article (6.1.11) est implémenté, le contrôle E/S devra lui envoyer le message **get**. Sinon, le contrôle E/S doit placer le message GET directement dans le tampon d'entrée.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The I/O Control shall set **bav** TRUE after the IEC 625-1 L or LE function is in the Listener Active State (LACS) and the IEC 625-1 Acceptor Handshake function enters the Accept Data State (ACDS). The **bav** message shall be set TRUE once and only once for each transition from the Acceptor Ready State (ACRS) to ACDS.

The I/O Control shall set **bav** FALSE when any of the following conditions occur:

1) A data byte is placed in the Input Buffer. (If the **device** has a zero length Input Buffer, the I/O Control shall set **bav** FALSE when the data byte is sent to the Parser.) See 6.1.4.1.2.

2) The **dcas** message is set TRUE. The associated data byte shall be discarded.

No other device or interface condition shall cause **bav** to be set FALSE. See 6.1.4.1.3.

6.1.4.2.5 Group execute trigger message (get)

The IEC 625-2 **get** message signals the transition of the IEC 625-1 Device Trigger (DT) function to the DTAS (Device Trigger Active State). This transition occurs when an IEC 625-1 GET (Group Execute Trigger) remote interface message is received while the **device** was addressed to listen. The **get** message can be used to initiate a device-defined action or the action defined by the *DDT common command, see 10.4.

The **device** shall process data bytes and GET messages in the order received, see 6.1.4.1.4.

The standard specifies two methods for device triggering, the common command *TRG, (10.37), and the IEC 625-1 interface message GET. Typically, device triggering is used for two distinct purposes:

1) For the properly sequenced initiation of a device-defined or user-defined action within a single addressed device.

2) For the synchronized initiation of device-defined or user-defined actions within multiple addressed devices.

For purpose 1, either GET or *TRG may be used for device triggering. The storage of either one in the Input Buffer ensures that the trigger will be executed in the correct sequence in relation to other commands.

For purpose 2, where the required synchronization could be compromised by the software/firmware processing overhead, the **get** message may be used in conjunction with the optional Trigger Control Block. **get** will generate a **trigger** local message for immediate execution by the **device** provided that the Input Buffer is empty and that the Parser and Execution Control are idle. If the Input Buffer is not empty, the GET message must be placed in the Input Buffer, and execution of the trigger deferred until previous commands have been executed.

If the optional Trigger Control block, clause 6.1.11, is implemented, the I/O Control shall send the **get** message to it. Otherwise, the I/O Control shall place the GET message directly in the Input Buffer.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

6.1.4.2.6 Message d'état actif de remise à zéro de l'appareil (dcas)

Le contrôle E/S doit mettre le message **dcas** VRAI une fois que l'**appareil** est entré dans l'état actif de remise à zéro (DCAS) défini par CEI 625-1. L'**appareil** entre en DCAS chaque fois qu'il reçoit le télémessage de remise à zéro (DCL) ou qu'il reçoit le télémessage de remise à zéro de l'appareil sélectionné (SDC) pendant qu'il est adressé pour écouter.

Lorsque **dcas** est mis VRAI, le tampon d'entrée et la file d'attente de sortie doivent être effacés, l'analyseur syntaxique, le contrôle d'exécution et le formateur de réponse doivent être remis à zéro, et l'**appareil** doit entrer dans l'état INACTIF d'échanges de messages. L'**appareil** entre également en OCIS (état inactif d'instruction "opération achevée") et OQIS (état inactif d'interrogation "opération achevée"), selon 12.5, et revient en fonctionnement normal après les instructions *AAD ou *DLF, article 13.

Le contrôle E/S doit mettre le message **dcas** FAUX lorsque ces actions sont terminées et le contrôle d'échange de messages entre dans l'état INACTIF. Les exigences de remise à zéro de l'appareil sont résumées en 5.8.

L'**appareil** doit assurer que les octets de données et que DCL ou SDC sont traités dans l'ordre reçu. Si l'**appareil** reçoit un nouvel octet de données alors que **dcas** est VRAI, il ne doit pas supprimer l'octet de données ou ne doit pas mettre **bav** FAUX.

6.1.4.2.7 Message "caractère d'arrêt de message de réponse envoyé" (RMT-envoyé)

Le bloc de contrôle E/S doit mettre le message RMT-envoyé VRAI lorsqu'il a envoyé le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>.

La réception des messages **bav** ou **brq** VRAIS doit faire mettre au contrôle E/S le message RMT-envoyé FAUX.

6.1.5 *Tampon d'entrée*

Le tampon d'entrée, représenté à la figure 6-2, stocke les messages DAB, GET et END. Le tampon d'entrée livre alors ces messages à l'analyseur syntaxique dans l'ordre dans lequel ils ont été reçus du contrôle E/S. Le message **dcas** ne doit pas être stocké dans le tampon d'entrée car cela pourrait provoquer une impossibilité de l'**appareil** à répondre immédiatement au **dcas**.

Le tampon d'entrée doit être implémenté comme une structure de données "premier entré, premier sorti". Les octets de données, messages END et GET sont placés dans le tampon d'entrée par le contrôle E/S au fur et à mesure que l'**appareil** les saisit dans le bus CEI 625-1. Les détails pour savoir quand et comment les articles sont placés dans le tampon d'entrée sont décrits dans 6.1.4.1.2, 6.1.4.1.3 et 6.1.4.1.4 concernant les réceptions par le contrôle E/S. Les messages DAB, END et GET sont enlevés du tampon d'entrée par l'analyseur syntaxique à une cadence compatible avec les capacités de l'analyseur syntaxique et l'exécution associée des instructions.

6.1.5.1 Règles du tampon d'entrée

6.1.5.1.1 Longueur du tampon d'entrée

Le tampon d'entrée doit être égal à ou supérieur à zéro en longueur. La longueur du tampon d'entrée peut être fixée à un certain nombre d'octets ou de messages de programmation, ou sa longueur peut varier. La longueur du tampon d'entrée doit être spécifiée dans la documentation de l'utilisateur.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.1.4.2.6 Device clear active state message (dcas)

The I/O Control shall set the **dcas** message TRUE after the **device** enters the Device Clear Active State (DCAS) defined by IEC 625-1. The **device** enters DCAS whenever it receives the Device Clear (DCL) remote command or it receives the Selected Device Clear (SDC) remote command while it is addressed to listen.

When **dcas** is set TRUE, the Input Buffer and Output Queue shall be cleared, the Parser, Execution Control, and Response Formatter shall be reset, and the **device** shall enter the Message Exchange IDLE state. The **device** also enters OCIS (Operation Complete Command Idle State) and OQIS (Operation Complete Query Idle State), as in 12.5, and returns to normal operation after the *AAD or *DLF commands, clause 13.

The I/O Control shall set the **dcas** message FALSE when these actions are completed and the Message Exchange Control enters the IDLE state. Device Clear requirements are summarized in clause 5.8.

The **device** shall ensure that data bytes and DCL or SDC are processed in the order received. If the **device** receives a new data byte while **dcas** is TRUE, it shall not discard the data byte or set **bav** FALSE.

6.1.4.2.7 Response message terminator sent message (RMT-sent)

The I/O Control block shall set the RMT-sent message TRUE when it has sent the <RESPONSE MESSAGE TERMINATOR>.

Receipt of TRUE **bav** or **brq** messages shall cause the I/O Control to set the RMT-sent message FALSE.

6.1.5 *Input Buffer*

The Input Buffer, shown in figure 6-2, stores DAB's, GET messages, and END messages. The Input Buffer then delivers these messages to the parser in the order they were received from the I/O Control. The **dcas** message shall not be stored in the Input Buffer as this could result in the **device** not being able to respond immediately to **dcas**.

The Input Buffer shall be implemented as a first-in first-out data structure. Data bytes, END, and GET messages are placed into the Input Buffer by the I/O Control as the **device** handshakes them from the IEC 625-1 bus. The details of when and how items are placed into the Input Buffer are described in 6.1.4.1.2, 6.1.4.1.3, and 6.1.4.1.4 on I/O Control Receiving. DAB's, END, and GET messages are removed from the Input Buffer by the parser at a rate consistent with the capabilities of the parser and the associated execution of commands.

6.1.5.1 Input Buffer rules

6.1.5.1.1 Input Buffer length

The Input Buffer shall be equal to or greater than zero in length. The Input Buffer's length may be fixed at some number of bytes or program messages, or its length may vary. The Input Buffer length shall be specified in the user documentation.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Le fonctionnement du tampon d'entrée est transparent aux programmes d'application, sauf comme amélioration des performances. Il est recommandé que les **appareils** qui peuvent accepter des messages de programmation plus vite qu'ils ne peuvent les exécuter aient des tampons d'entrée suffisamment longs pour permettre le stockage d'un message de programmation d'une longueur raisonnable. Cette mise en mémoire tampon permettra aux programmes d'application d'envoyer des messages à l'**appareil** et ensuite de continuer à utiliser le **bus du système** pour d'autres objectifs pendant que l'**appareil** prend le temps qu'il lui faut pour répondre.

EXEMPLE: Un traceur numérique répond lentement aux instructions, en raison de la nécessité de permettre le déplacement du stylo. Il est recommandé que, si un message de programmation de longueur typique pour l'**appareil** est de 80 octets, l'**appareil** ait un tampon d'entrée d'au moins 80 octets.

Le concepteur de l'appareil doit considérer l'application de l'**appareil** en choisissant une longueur de tampon d'entrée. Un tampon d'entrée plus long est utile dans les instruments de test sophistiqués, très indépendants, qui utilisent des éléments de <MESSAGES DE PROGRAMMATION> complexes. Le traitement de ces éléments de <MESSAGES DE PROGRAMMATION> complexes peut prendre beaucoup de temps. Dans certains systèmes, ce temps peut être utilisé pour des tâches autres que d'attendre que cet **appareil** ait fini de traiter un <MESSAGE DE PROGRAMMATION>. L'augmentation de la longueur du tampon, toutefois, nécessite que l'utilisateur soit plus attentif à la synchronisation à l'intérieur du système.

Si on utilise un tampon d'entrée de longueur zéro, l'**interface du système** fournit une source de synchronisation. Avec ce tampon d'entrée très court, l'**appareil** ne peut jamais se laisser distancer par le **contrôleur**. Si un DAB ou et GET est envoyé, il sera traité immédiatement par l'**appareil**.

NOTE - La plupart des circuits intégrés commerciaux qui implémentent les exigences de la CEI 625-1 fournissent au moins un tampon de données interne égal à un octet.

6.1.5.1.2 Vue d'ensemble du tampon d'entrée

Le tampon d'entrée ne doit pas dépasser sa capacité. Si le tampon est plein, l'appareil ne doit pas entrer dans l'état accepteur prêt (ACRS) CEI 625-1 pendant qu'il est dans l'état actif écouteur (LACS) (retenue NRFD). Une fois que l'analyseur syntaxique a enlevé un article du tampon d'entrée, le contrôle E/S peut accepter un autre article de l'**interface du système**. Aucune erreur ne doit être signalée lorsque le tampon d'entrée se remplit, sauf si l'**appareil** est bloqué, voir 6.3.1.7 et 6.5.7.4.

6.1.5.1.3 Effacement du tampon d'entrée

Le tampon d'entrée doit être effacé lorsque pon ou **dcas** est VRAI. Aucune condition de l'appareil ou d'interface ne doit empêcher l'opération d'effacement d'avoir lieu. A l'exception du vidage pendant le traitement normal des messages et les actions de pon et **dcas**, aucune autre condition d'interface ou de l'appareil ne doit effacer le tampon d'entrée. Le tampon d'entrée est effacé pour garantir que l'**appareil** sera prêt à recevoir et à exécuter de nouveaux éléments de <MESSAGES DE PROGRAMMATION> suivant **dcas**.

6.1.5.2 Messages du tampon d'entrée

6.1.5.2.1 Message "tampon d'entrée vide" (ib-empty)

Le message "tampon d'entrée vide", **ib-empty**, doit être VRAI lorsque le tampon d'entrée est vide. Le message **ib-empty** doit être FAUX dans tous les autres cas.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The operation of the Input Buffer is transparent to application programs, except as a performance improvement. **Devices** which can accept program messages faster than they can execute them should have Input Buffers long enough to allow storing a reasonably sized program message. This buffering will allow application programs to send messages to the **device** and then continue to use the **system bus** for other purposes while the **device** is taking the time it needs to respond.

EXAMPLE: A digital plotter responds to commands slowly due to the need to allow for pen movement. If a typical length program message for the **device** is 80 bytes, the **device** should have an Input Buffer of at least 80 bytes.

The device designer must consider the application of the **device** when choosing an Input Buffer length. A larger Input Buffer is useful in smart, highly independent test instruments which use complex <PROGRAM MESSAGE> elements. These complex <PROGRAM MESSAGE> elements may require a significant time to process. In some systems, this time can be used for tasks other than waiting for this **device** to finish processing a <PROGRAM MESSAGE>. Increasing buffer length, however, requires the user to be more careful about synchronization within the system.

If a zero length Input Buffer is used, the **system interface** provides a source of synchronization. With this very short Input Buffer, the **device** can never fall behind the **controller**. If a DAB or GET is sent it will be processed immediately by the **device**.

NOTE - Most commercial integrated circuits which implement the requirements of IEC 625-1 provide at least a one-byte internal data buffer.

6.1.5.1.2 Input Buffer overflow

The Input Buffer shall not overflow. If the buffer is full, the device shall not enter the IEC 625-1 Acceptor Ready State (ACRS) while in Listener Active State (LACS) (NRFD holdoff). After the Parser removes an item from the Input Buffer, the I/O control may accept another item from the **system interface**. No error shall be reported when the Input Buffer fills unless the **device** becomes deadlocked, see 6.3.1.7 and 6.5.7.4.

6.1.5.1.3 Input Buffer clearing

The Input Buffer shall be cleared when pon or **dcas** is TRUE. No device or interface condition is allowed to keep the clear operation from taking place. Except for the emptying during normal message processing and the actions of pon and **dcas**, no other interface or device condition shall be allowed to clear the Input Buffer. The Input Buffer is cleared to ensure that the **device** will be ready to receive and execute new <PROGRAM MESSAGE> elements following **dcas**.

6.1.5.2 Input Buffer messages

6.1.5.2.1 Input Buffer empty message (ib-empty)

The Input Buffer empty message, **ib-empty**, shall be TRUE when the Input Buffer is empty. The **ib-empty** message shall be FALSE at all other times.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

6.1.5.2.2 Message "tampon d'entrée plein" (ib-full)

Le message "tampon d'entrée plein", **ib-full**, doit être VRAI lorsque le tampon d'entrée est plein. Le message **ib-full** doit être FAUX dans tous les autres cas.

6.1.6 *Analyseur syntaxique*

L'analyseur syntaxique, représenté à la figure 6-2, est la partie logique d'un **appareil** qui prend les messages DAB, END et GET dans le tampon d'entrée et les analyse en séparant les divers éléments syntaxiques CEI 625-2. Il signale une syntaxe ou des en-têtes incorrects au bloc d'indication d'état en tant qu'erreurs d'instruction. L'analyseur syntaxique convertit les éléments syntaxiques en une représentation interne qui est envoyée au contrôle d'exécution. L'analyseur syntaxique génère également les messages **eom** et d'**interrogation** lorsqu'il reconnaît ces éléments syntaxiques.

Un <MESSAGE DE PROGRAMMATION> ou une <UNITÉ DE MESSAGE DE PROGRAMMATION> est considérée comme "analysée" lorsqu'elle a été analysée par l'analyseur syntaxique et l'analyseur syntaxique est prêt à poursuivre l'analyse d'autres éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION>.

6.1.6.1 Règles de l'analyseur syntaxique

6.1.6.1.1 Erreurs de l'analyseur syntaxique

L'analyseur syntaxique doit détecter les erreurs d'instruction et les signalera au bloc d'indication d'état en mettant le bit d'erreur d'instruction dans le registre d'état d'événement standard, voir 11.5.1.1.4.

L'analyseur syntaxique doit signaler une erreur d'instruction s'il trouve une erreur de syntaxe dans un <MESSAGE DE PROGRAMMATION>, s'il trouve un en-tête non reconnu, s'il trouve un paramètre (élément de <DONNÉES DE PROGRAMMATION>) qui ne convient pas à son en-tête associé, ou s'il rencontre un message GET entre le premier octet d'un <MESSAGE DE PROGRAMMATION> et le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>.

Lorsqu'une erreur d'instruction est détectée, l'appareil doit déterminer ce que l'on devra faire des éléments analysables précédents du même <MESSAGE DE PROGRAMMATION>. L'**appareil** est autorisé soit à supprimer de tels éléments analysables, soit à essayer de les exécuter.

Lorsqu'il se produit une erreur d'instruction, l'analyseur syntaxique doit supprimer tous les messages DAB et GET consécutifs jusqu'à ce que:

1) **dcas** soit VRAI, ou
2) **pon** soit VRAI, ou
3) **eom** soit VRAI, ou
4) **brq** soit VRAI et **ib-empty** soit VRAI.

Le concepteur de l'appareil peut également choisir d'arrêter la suppression des messages DAB et GET pour d'autres conditions définies par l'appareil, telles que réception d'une virgule, d'un point-virgule, ou de NL. L'analyseur syntaxique est remis à zéro lorsque l'une quelconque de ces conditions est rencontrée.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.1.5.2.2 Input Buffer full message (ib-full)

The Input Buffer full message, **ib-full**, shall be TRUE when the Input Buffer is full. The **ib-full** message shall be FALSE at all other times.

6.1.6 *Parser*

The Parser, shown in figure 6-2, is the logical portion of a **device** which takes DAB's, END messages, and GET messages from the Input Buffer and analyzes them by separating out the various IEC 625-2 syntactic elements. It reports invalid syntax or headers to the Status Reporting block as Command Errors. The Parser converts syntactic elements into an internal representation which is sent to the Execution Control. The Parser also generates the **eom** and **query** messages when it recognizes these syntactic elements.

A <PROGRAM MESSAGE> or <PROGRAM MESSAGE UNIT> is considered "parsed" when it has been analyzed by the Parser and the Parser is ready to continue analyzing other <PROGRAM MESSAGE UNIT> elements.

6.1.6.1 Parser rules

6.1.6.1.1 Parser errors

The Parser shall detect Command Errors and report them to the Status Reporting block by setting the Command Error bit in the Standard Event Status Register, see 11.5.1.1.4.

The Parser shall report a Command Error if it finds a syntax error in a <PROGRAM MESSAGE>, finds an unrecognized header, finds a parameter (<PROGRAM DATA> element) that is of the wrong type for its associated header, or encounters a GET message between the first byte of a <PROGRAM MESSAGE> and the <PROGRAM MESSAGE TERMINATOR>.

When a Command Error is detected the device shall determine what will be done with any prior parsable elements of the same <PROGRAM MESSAGE>. The device is allowed to either discard or attempt to execute any such parsable elements.

When a Command Error occurs the Parser shall discard all subsequent DAB's and GET messages until either:

1) **dcas** is TRUE, or
2) **pon** is TRUE, or
3) **eom** is TRUE, or
4) **brq** is TRUE and **ib-empty** is TRUE.

The device designer may also choose to stop discarding DAB's and GET messages for other device defined conditions such as receipt of a comma, semicolon, or NL. The Parser is reset when any of these conditions is met.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

NOTE - L'**appareil** ne vide pas le tampon d'entrée après une erreur d'instruction. L'analyseur syntaxique supprime les messages DAB et GET jusqu'à ce qu'il ait détecté l'une des conditions ci-dessus. L'**appareil** reprendra alors l'analyse et l'exécution normales des messages.

6.1.6.1.2 Remise à zéro de l'analyseur syntaxique

Lorsque l'analyseur syntaxique est remis à zéro, il doit interpréter le prochain octet de données qu'il recevra comme le premier octet d'un <MESSAGE DE PROGRAMMATION>. L'analyseur syntaxique doit être remis à zéro lorsque **dcas** ou pon est VRAI. Aucune condition de l'appareil ou d'interface ne doit empêcher l'opération de remise à zéro d'avoir lieu. D'autres conditions pouvant remettre à zéro l'analyseur syntaxique sont décrites en 6.1.6.1.1.

6.1.6.2 Messages de l'analyseur syntaxique

6.1.6.2.1 Elément de message analysé

Un élément de message analysé représente la représentation interne de l'**appareil** de tout ou partie d'un <MESSAGE DE PROGRAMMATION>.

6.1.6.2.2 Message "Fin de message détectée" (eom)

L'analyseur syntaxique doit mettre le message "Fin de message détectée", **eom**, VRAI dans l'une des conditions suivantes:

1) Lorsqu'il reçoit un message END ou une séquence d'octets de données constituant un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>, (7.5) du tampon d'entrée, ou

2) Après avoir analysé un message GET, sauf lorsque le GET intervient entre le premier octet d'un <MESSAGE DE PROGRAMMATION> et le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>, (6.1.6.1.1) ou

3) Lorsque le bloc de contrôle de déclenchement a transmis le message de **déclenchement** au bloc de fonctions de l'appareil.

L'analyseur syntaxique doit mettre le message eom FAUX dans l'une des conditions suivantes:

1) Lorsqu'il est remis à zéro, ou

2) Lorsqu'il reçoit un autre message DAB ou GET du tampon d'entrée.

Lorsque l'analyseur syntaxique a reçu un message GET du tampon d'entrée ou le bloc de contrôle de déclenchement a transmis un message de **déclenchement** au bloc de fonctions de l'appareil, tout message d'**interrogation** associé doit être mis VRAI avant de mettre le message **eom** VRAI. Ceci assure une séquence correcte d'exécution des transitions des blocs d'échanges de messages (figure 6-4).

6.1.6.2.3 Message "Interrogation détectée" (interrogation)

L'analyseur syntaxique doit mettre le message "Interrogation détectée", **interrogation**, VRAI lorsqu'il reçoit ce qui suit:

1) Une séquence d'octets de données constituant un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct qui n'est pas une étiquette de macro, ou

2) Un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct qui est une étiquette de macro, où la macro est définie pour inclure une <UNITÉ DE MESSAGE D'INTERROGATION>, ou

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

NOTE - The **device** does not empty the Input Buffer after a Command Error. The Parser discards DABs and GET messages until it has detected one of the above conditions. The **device** will then resume normal parsing and executing of messages.

6.1.6.1.2 Parser resetting

When the Parser is reset it shall interpret the next data byte it receives as the first byte of a <PROGRAM MESSAGE>. The Parser shall be reset when **dcas** or pon is TRUE. No device or interface condition is allowed to keep the reset operation from taking place. Other conditions that can reset the Parser are described in 6.1.6.1.1.

6.1.6.2 Parser messages

6.1.6.2.1 Parsed message element

A Parsed message element represents the **device's** internal representation of all or part of a <PROGRAM MESSAGE>.

6.1.6.2.2 End of message detected message (eom)

The Parser shall set the end of message detected message, **eom**, TRUE when any of the following conditions occurs:

1) When it receives an END message or a sequence of data bytes making up a <PROGRAM MESSAGE TERMINATOR> (7.5) from the Input Buffer, or

2) After it parses a GET message, except when the GET occurs between the first byte of a <PROGRAM MESSAGE> and the <PROGRAM MESSAGE TERMINATOR> (6.1.6.1.1), or

3) When the Trigger Control block has passed the **trigger** message to the Device Functions block.

The Parser shall set the eom message FALSE when any of the following conditions occurs:

1) When it is reset, or

2) When it receives any other DAB or GET message from the Input Buffer.

When the Parser has received a GET message from the Input Buffer or the Trigger Control block has passed a **trigger** message to the Device Functions block, any associated **query** message shall be set TRUE prior to setting the **eom** message TRUE. This ensures a proper sequence of execution of Message Exchange block transitions (figure 6-4).

6.1.6.2.3 Query Detected message (query)

The Parser shall set the Query Detected message, **query**, TRUE when it receives any of the following:

1) A sequence of data bytes making up a valid <QUERY PROGRAM HEADER> which is not a macro label, or

2) A valid <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> which is a macro label, where the macro is defined to include a <QUERY MESSAGE UNIT>, or

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

3) Un message GET ou une instruction *TRG (ou, s'il est implémenté, le message de déclenchement), où l'appareil implémente l'instruction commune *DDT et l'action de déclenchement de l'appareil définie par l'instruction *DDT comprend une <UNITÉ DE MESSAGE D'INTERROGATION>, ou

4) Un message GET ou une instruction *TRG (ou, s'il est implémenté, le message de déclenchement), où l'appareil n'implémente pas l'instruction commune *DDT et l'action de déclenchement de l'appareil spécifiée par le concepteur de l'appareil est de générer un <MESSAGE DE RÉPONSE>.

Un "<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct" signifie que tous les caractères jusqu'à et y compris le "?" représentent un mnémonique d'interrogation ou une étiquette de macro corrects, et les caractères suivant immédiatement le "?" forment un séparateur ou caractère d'arrêt autorisé.

Un "<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> correct" signifie que l'en-tête représente un mnémonique d'instruction ou une étiquette de macro corrects, et les caractères suivant immédiatement le mnémonique forment un séparateur ou caractère d'arrêt autorisé.

EXEMPLE:

*XYZ? n'est pas un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct parce qu'il n'y a pas d'interrogation commune "*XYZ?"

*ESR?X n'est pas un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct parce que "X" n'est pas un séparateur ou caractère d'arrêt

*IDN?42 est un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct bien que l'argument soit incorrect.

Une erreur d'instruction doit être générée pour l'ensemble des trois conditions énumérées ci-desssus.

L'analyseur syntaxique doit mettre **query** FAUX lorsqu'il est remis à zéro ou lorsqu'il reçoit un autre message DAB ou GET du tampon d'entrée.

6.1.6.2.4 Message "Analyseur syntaxique inactif" (p-idle)

L'analyseur syntaxique doit mettre le message "analyseur syntaxique inactif", **p-idle**, VRAI lorsque l'analyseur syntaxique a analysé tous les messages antérieurs et n'est pas en train de traiter activement un <MESSAGE D'INSTRUCTION DE PROGRAMMATION> correct ou un <MESSAGE D'INTERROGATION DE PROGRAMMATION> correct. L'analyseur syntaxique doit mettre **p-idle** FAUX dans tous les autres cas.

6.1.6.2.5 Message "Analyseur syntaxique bloqué" (p-blocked)

L'analyseur syntaxique doit mettre le message "Analyseur syntaxique bloqué", **p-blocked**, VRAI lorsque l'analyseur syntaxique attend que le contrôle d'exécution ait fini de traiter le précédent élément de message analysé. L'analyseur syntaxique doit mettre **p-blocked** FAUX dans tous les autres cas.

6.1.7 *Contrôle d'exécution*

Le bloc de contrôle d'exécution, représenté à la figure 6-2, traite les paramètres couplés, voir 6.4.5.3, les interrogations qui nécessitent une action de l'appareil avant de pouvoir générer un <MESSAGE DE RÉPONSE>, et les instructions de synchronisation de l'**appareil** décrites à la section 12. Le contrôle d'exécution détermine quand l'**appareil** a assez d'informations pour exécuter une instruction. Lorsque le contrôle d'exécution a regroupé assez d'informations pour lancer une action de l'**appareil**, il envoie un élément

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

3) A GET message or *TRG command (or, if implemented, the trigger message), where the device implements the *DDT common command and the device trigger action defined by the *DDT command includes a <QUERY MESSAGE UNIT>, or

4) A GET message or *TRG command (or, if implemented, the trigger message), where the device does not implement the *DDT common command and the device trigger action specified by the device designer is to generate a <RESPONSE MESSAGE>.

"Valid <QUERY PROGRAM HEADER>" means that all characters up to and including the "?" represent a valid query mnemonic or macro label, and the characters immediately following the "?" form a legal separator or terminator.

"Valid <COMMAND PROGRAM HEADER>" means that the header represents a valid command mnemonic or macro label, and the characters immediately following the mnemonic form a legal separator or terminator.

EXAMPLE:

*XYZ? is not a valid <QUERY PROGRAM HEADER> because there is no "*XYZ?" common query

*ESR?X is not a valid <QUERY PROGRAM HEADER> because "X" is not a separator or terminator

*IDN? 42 is a valid <QUERY PROGRAM HEADER> although the argument is incorrect.

A Command Error will be generated for all of the three conditions listed above.

The Parser shall set **query** FALSE when it is reset or when it receives any other DAB or GET message from the Input Buffer.

6.1.6.2.4 Parser idle message (p-idle)

The Parser shall set the Parser idle message, **p-idle**, TRUE when the Parser has parsed all prior messages and is not actively processing a valid <COMMAND PROGRAM MESSAGE> or a valid <QUERY PROGRAM MESSAGE>. The Parser shall set **p-idle** FALSE at all other times.

6.1.6.2.5 Parser Blocked message (p-blocked)

The Parser shall set the Parser Blocked message, **p-blocked**, TRUE when the Parser is waiting for the Execution Control to finish processing the previous Parsed Message Element. The Parser shall set **p-blocked** FALSE at all other times.

6.1.7 *Execution Control*

The Execution Control block, shown in figure 6-2, deals with coupled parameters, see 6.4.5.3, queries which require a device action before they can generate a <RESPONSE MESSAGE>, and the **device** synchronization commands described in clause 12. The Execution Control determines when the **device** has enough information to execute a command. When the Execution Control has gathered enough information to initiate a **device** action, it sends an Executable Message Element to the Device Functions

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

de message exécutable au bloc de fonctions de l'appareil. Le contrôle d'exécution peut attendre que les actions de l'**appareil** qui en résultent soient terminées (instruction séquentielle) ou peut continuer à traiter des instructions (instruction recouvrable), suivant l'instruction. Voir 12.2. Une <UNITÉ DE MESSAGE DE PROGRAMMATION> est considérée comme "exécutée" lorsqu'elle a été analysée et toutes les opérations de l'appareil correspondantes ont été soit terminées (instruction séquentielle), soit lancées (instruction recouvrable).

6.1.7.1 Règles du contrôle d'exécution

6.1.7.1.1 Erreurs du contrôle d'exécution

Le contrôle d'exécution doit détecter les erreurs d'exécution et les signaler au bloc d'indication d'état, voir 11.5.1.1.5.

6.1.7.1.2 Remise à zéro du contrôle d'exécution

Lorsque le contrôle d'exécution est remis à zéro, l'historique antérieur n'aura aucun effet sur l'exécution de nouveaux éléments de <MESSAGES DE PROGRAMMATION> reçus après **dcas** ou pon. Le contrôle d'exécution doit être remis à zéro lorsque **dcas** ou pon est VRAI. Aucune condition de l'**appareil** ou d'interface ne doit empêcher l'opération de remise à zéro d'avoir lieu. Aucune autre condition d'interface ou de l'**appareil** ne remettra à zéro le contrôle d'exécution.

6.1.7.2 Messages du contrôle d'exécution

6.1.7.2.1 Elément de message exécutable

Un élément de message exécutable est une pseudo-instruction au bloc de fonctions de l'appareil pour exécuter une action spécifique à l'appareil. Ces éléments peuvent prendre la forme de demandes d'interrogations correctes ou d'éléments de messages exécutables non-interrogation.

Une demande d'interrogation correcte est un élément de message exécutable qui fait envoyer au bloc de fonctions de l'appareil des données au formateur de réponse. Un élément non-interrogation exécute une action spécifique à l'appareil mais n'envoie pas de données de réponse au formateur de réponse. Une seule <UNITÉ DE MESSAGE DE PROGRAMMATION> peut faire envoyer plus d'un élément de message exécutable.

EXEMPLE Une interrogation qui lance une mesure et renvoie les résultats lorsque la mesure est terminée pourrait nécessiter deux éléments de messages exécutables: un élément non-interrogation pour lancer la mesure et une demande d'interrogation correcte pour envoyer les résultats au formateur de réponse.

Le contrôle d'exécution doit maintenir également l'ordre de création des éléments d'<UNITÉ DE MESSAGE DE RÉPONSE> (comme demandé par 6.4.5.4), en envoyant les demandes d'interrogations correctes, associées aux <UNITÉS DE MESSAGE D'INTERROGATION> précédemment reçues qui sont déjà envoyées au bloc de fonctions de l'appareil.

6.1.7.2.2 Message "Contrôle d'exécution inactif" (ec-idle)

Le contrôle d'exécution doit mettre le message "Contrôle d'exécution inactif", **ec-idle**, VRAI dans les conditions suivantes:

1) Toutes les instructions différées ont été exécutées, et

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

block. The Execution Control may wait for the resulting **device** actions to complete (Sequential Command) or may continue processing commands (Overlapped Command), depending on the command. See 12.2. A <PROGRAM MESSAGE UNIT> is considered "executed" when it has been parsed and all corresponding device operations have been either completed (Sequential Command) or initiated (Overlapped Command).

6.1.7.1 Execution Control Rules

6.1.7.1.1 Execution Control Errors

The Execution Control shall detect Execution Errors and report them to the Status Reporting block, see 11.5.1.1.5.

6.1.7.1.2 Execution Control Resetting

When the Execution Control is reset, prior history will not affect the execution of new <PROGRAM MESSAGE> elements received after **dcas** or pon. The Execution Control shall be reset when **dcas** or pon is TRUE. No **device** or interface condition is allowed to keep the reset operation from taking place. No other interface or **device** condition shall reset the Execution Control.

6.1.7.2 Execution Control messages

6.1.7.2.1 Executable Message Element

An Executable Message Element is a directive to the Device Functions block to perform some device-specific action. These may take the form of Valid Query Requests or non-query Executable Message Elements.

A Valid Query Request is an Executable Message Element which causes the Device Functions block to send data to the Response Formatter. A non-query element performs device-specific action but does not send Response Data to the Response Formatter. A single <PROGRAM MESSAGE UNIT> may cause more than one Executable Message Element to be sent.

EXAMPLE: A query which starts a measurement and returns the results when the measurement is complete might require two Executable Message Elements: a non-query element to start the measurement and a Valid Query Request to send the results to the Response Formatter.

The Execution Control also shall maintain the order of creating <RESPONSE MESSAGE UNIT> elements (as required by 6.4.5.4), by sending the Valid Query Requests, associated with previous received <QUERY MESSAGE UNITS> that are already sent to the Device Functions block.

6.1.7.2.2 Execution Control Idle message (ec-idle)

The Execution Control shall set the Execution Control Idle Message, **ec-idle**, TRUE when the following conditions occur:

1) All deferred commands have been executed, and

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

2) Le contrôle d'exécution est prêt à accepter une nouvelle instruction ou interrogation de l'analyseur syntaxique, et

a) Le contrôle d'exécution est remis à zéro, ou

b) Toutes les instructions précédentes ont été exécutées.

Le contrôle d'exécution doit mettre **ed-idle** FAUX dans tous les autres cas.

Le contrôle d'exécution ne doit pas mettre le message **ec-idle** VRAI.

6.1.7.2.3 Message "Contrôle d'exécution bloqué" (ec-blocked)

Le contrôle d'exécution doit mettre le message "Contrôle d'exécution bloqué", **ec-blocked**, VRAI lorsqu'il est prêt à envoyer une demande d'interrogation correcte au bloc de fonctions de l'appareil mais doit attendre que le formateur de réponse et le bloc de fonctions de l'appareil traitent une demande d'interrogation correcte antérieure. Le contrôle d'exécution doit mettre **ec-blocked** FAUX dans tous les autres cas.

6.1.8 *Fonctions de l'appareil*

Le bloc des fonctions de l'appareil, représenté dans les figures 6-1 et 6-2, contient toutes les caractéristiques et fonctions spécifiques à l'appareil. Il accepte les instructions exécutables par l'appareil du contrôle d'exécution et réalise les opérations associées.

6.1.8.1 Règles des fonctions de l'appareil

Le bloc des fonctions de l'appareil doit accepter les demandes d'interrogation correctes du contrôle d'exécution et envoyer toute donnée de réponse demandée au formateur de réponse. Toutes les instructions d'interrogation communes CEI 625-2 implémentées et toutes les instructions d'interrogation spécifiques à l'appareil doivent être traitées de cette façon. Le bloc de fonctions de l'**appareil** doit donner des informations sur l'état de l'appareil au bloc d'indication d'état, article 11.

6.1.8.2 Messages des fonctions de l'appareil

6.1.8.2.1 Message "Mise sous tension" (pon)

Le bloc de fonctions de l'appareil doit générer le message local pon comme décrit dans CEI 625-1.

Lorsque pon est mis VRAI, le tampon d'entrée et la file d'attente de sortie doivent être effacés, l'analyseur syntaxique, le contrôle d'exécution et le formateur de réponse doivent être remis à zéro, et l'**appareil** entrera dans l'état INACTIF d'échanges des messages. L'appareil entre également en OCIS (état inactif d'instruction "opération achevée") et OQIS (état inactif d'interrogation "opération achevée"), voir 12.5. Le message pon peut également affecter les réglages spécifiques à l'appareil, voir 5.12. Le bloc de fonctions de l'appareil doit mettre le message pon FAUX lorsque ces actions sont terminées et le contrôle d'échanges de messages entre dans l'état INACTIF (voir 6.3.1.1).

6.1.8.2.2 Données de réponse

Les données de réponse sont des données fournies par l'**appareil** en réponse à des éléments d'<UNITÉ DE MESSAGE D'INTERROGATION> qui ont été analysés et envoyés à l'**appareil** sous la forme d'éléments de messages exécutables (demandes d'interrogations correctes). Les données de réponse sont non formatées et peuvent être uniquement des jetons représentant les données réelles à formater.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

2) Execution Control is ready to accept a new command or query from the Parser, and

   a) Execution Control is reset, or

   b) All previous commands have been executed.

The Execution Control shall set **ec-idle** FALSE at all other times.

The Execution Control shall not set the **ec-idle** message TRUE.

6.1.7.2.3 Execution Control Blocked message (ec-blocked)

The Execution Control shall set the Execution Control Blocked message, **ec-blocked**, TRUE when it is ready to send a Valid Query Request to the Device Functions block but must wait for the Response Formatter and Device Functions block to process a prior Valid Query Request. The Execution Control shall set **ec-blocked** FALSE at all other times.

6.1.8 *Device Functions*

The Device Functions block, shown in both figures 6-1 and 6-2, contains all device-specific features and functions. It accepts Executable Message Elements from the Execution Control and performs the associated operations.

6.1.8.1 Device Functions rules

The Device Functions block shall accept Valid Query Requests from the Execution Control and shall send any required Response Data to the Response Formatter. All implemented IEC 625-2 common query commands and all device-specific query commands shall be handled in this manner. The Device Functions block shall report device status information to the Status Reporting block, clause 11.

6.1.8.2 Device Functions messages

6.1.8.2.1 Power-On message (pon)

The Device Functions block shall generate the pon local message as described in IEC 625-1.

When pon is set TRUE, the Input Buffer and Output Queue shall be cleared, the Parser, Execution Control, and Response Formatter shall be reset, and the **device** shall enter the Message Exchange IDLE state. The device also enters OCIS (Operation Complete Command Idle State) and OQIS (Operation Complete Query Idle State), see 12.5. The pon message may also affect device-specific settings, see 5.12. The Device Functions block shall set the pon message FALSE when these actions are completed and the Message Exchange Control enters the IDLE state (see 6.3.1.1).

6.1.8.2.2 Response Data

Response Data is data provided by the **device** in response to <QUERY MESSAGE UNIT> elements that have been parsed and sent to the **device** in the form of Executable Message Elements (Valid Query Requests). Response Data is unformatted and may only be tokens representing the actual data to be formatted.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.1.9 *Formateur de réponse*

Le formateur de réponse, représenté à la figure 6-2, constitue un <MESSAGE DE RÉPONSE> fait d'éléments de message de réponse à partir de demandes d'interrogation correctes et de données de réponse. Le <MESSAGE DE RÉPONSE> est placé dans la file d'attente de sortie. La première responsabilité du formateur de réponse est de convertir la représentation interne des éléments de données en une séquence d'octets de données conformément aux règles de syntaxe de l'article 8.

6.1.9.1 Règles du formateur de réponse

Le formateur de réponse doit délimiter toutes les <UNITÉS DE MESSAGE DE RÉPONSE>, sauf la dernière, dans le <MESSAGE DE RÉPONSE> avec un <SÉPARATEUR D'UNITÉS DE MESSAGE DE RÉPONSE> (";", voir 8.4.1), et la dernière <UNITÉ DE MESSAGE DE RÉPONSE> dans le <MESSAGE DE RÉPONSE> avec le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>.

Le formateur de réponse doit être remis à zéro lorsque **dcas** ou pon est VRAI. Aucune condition de l'appareil ou d'interface ne doit empêcher l'opération de remise à zéro d'avoir lieu. Le formateur de réponse est remis à zéro de façon que tout élément nouveau de <MESSAGE DE RÉPONSE> ne soit pas affecté par les conditions qui existaient avant **dcas** ou pon.

6.1.9.2 Messages du formateur de réponse

6.1.9.2.1 Elément de message de réponse

Un élément de message de réponse représente une représentation interne de l'**appareil** de tout ou partie d'un <MESSAGE DE RÉPONSE>.

6.1.9.2.2 Message "Formateur de réponse bloqué" (rf-blocked)

Le formateur de réponse doit mettre le message "Formateur de réponse bloqué", **rf-blocked**, VRAI lorsqu'il a une <UNITÉ DE MESSAGE DE RÉPONSE> à placer dans la file d'attente de sortie et le message "File d'attente de sortie pleine", **eq-full**, est VRAI. Le formateur de réponse doit mettre **rf-blocked** FAUX dans tous les autres cas.

6.1.10 *File d'attente de sortie*

La file d'attente de sortie, représentée à la figure 2, stocke des messages de l'appareil au contrôleur jusqu'à ce que le **contrôleur** les lise. Le formateur de réponse place les messages DAB et END dans la file d'attente de sortie en réponse aux instructions d'interrogation. Ces octets sont enlevés de la file d'attente de sortie au fur et à mesure qu'ils sont lus par le **contrôleur**.

La file d'attente de sortie représentée à la figure 6-2 est un modèle logique. La file d'attente de sortie dans un appareil réel peut stocker les octets individuels du <MESSAGE DE RÉPONSE> et/ou les jetons qui représentent les éléments de <MESSAGES DE RÉPONSE>. Le concepteur de l'appareil peut également utiliser d'autres moyens pour déterminer la réponse à générer.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.1.9 *Response Formatter*

The Response Formatter, shown in figure 6-2, builds a <RESPONSE MESSAGE> out of Response Message Elements from Valid Query Requests and response data. The <RESPONSE MESSAGE> is placed into the Output Queue. The Response Formatter's primary responsibility is to convert the internal representation of data elements into a sequence of data bytes according to the syntax rules in clause 8.

6.1.9.1 Response Formatter rules

The Response Formatter shall delimit all but the last <RESPONSE MESSAGE UNIT> in the <RESPONSE MESSAGE> with a <RESPONSE MESSAGE UNIT SEPARATOR> (";", see 8.4.1), and the last <RESPONSE MESSAGE UNIT> in the <RESPONSE MESSAGE> with the <RESPONSE MESSAGE TERMINATOR>.

The Response Formatter shall be reset when **dcas** or pon is TRUE. No device or interface condition is allowed to keep the reset operation from taking place. The Response Formatter is reset so that any new <RESPONSE MESSAGE> elements will not be affected by conditions that existed before **dcas** or pon.

6.1.9.2 Response Formatter messages

6.1.9.2.1 Response Message Element

A Response Message Element represents a **device's** internal representation of all or part of a <RESPONSE MESSAGE>.

6.1.9.2.2 Response Formatter Blocked message (rf-blocked)

The Response Formatter shall set the Response Formatter Blocked message, **rf-blocked**, TRUE when it has a <RESPONSE MESSAGE UNIT> to be placed in the Output Queue and the Output Queue Full message, **oq-full**, is TRUE. The Response Formatter shall set **rf-blocked** FALSE at all other times.

6.1.10 *Output Queue*

The Output Queue, shown in figure 6-2, stores device-to-controller messages until the **controller** reads them. The Response Formatter places DAB and END messages into the Output Queue in response to query commands. These bytes are removed from the Output Queue as they are read by the **controller**.

The Output Queue shown in figure 6-2 is a logical model. The Output Queue in a real device may store the individual bytes of the <RESPONSE MESSAGE> and/or tokens which represent <RESPONSE MESSAGE> elements. The device designer may also use other means of determining the response to be generated.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Il est recommandé que le concepteur de l'appareil prévoie une file d'attente de sortie suffisamment longue pour traiter des éléments de <MESSAGE DE RÉPONSE> d'une longueur raisonnable. La longueur effective choisie dépendra des caractéristiques de l'**appareil**. Afin d'éviter de dépasser la capacité d'un tampon physique, le concepteur de l'appareil peut différer le formatage des éléments de <MESSAGE DE RÉPONSE> assez longs jusqu'à ce que le **contrôleur** en demande la sortie. Le concepteur de l'appareil doit éventuellement s'assurer que la longueur et l'exploitation de la file d'attente sont appropriées pour rendre un blocage improbable, voir 6.5.7.4.

6.1.10.1 Règles de la file d'attente de sortie

La file d'attente de sortie doit être effacée lorsque pon ou **dcas** est VRAI. Noter que ceci n'est pas une erreur d'interrogation. Une erreur d'interrogation doit être signalée si le contenu de la file d'attente de sortie est supprimé pour toute autre raison, voir 6.3.1.7, 6.3.2.2 et 6.3.2.3.

6.1.10.2 Messages de la file d'attente de sortie

6.1.10.2.1 Message "Message disponible" (MAV)

La file d'attente de sortie doit envoyer le message MAV au bloc d'indication d'état. Tant que la file d'attente de sortie contiendra un ou plusieurs octets, MAV sera VRAI. MAV doit être FAUX lorsque la file d'attente de sortie sera vide, sauf comme indiqué ci-après. Voir 12.2.1.2.

Lorsque l'**appareil** diffère la génération des données de réponse jusqu'à ce que **brq** soit VRAI, le message MAV doit être mis VRAI lorsque l'appareil sera prêt à générer les données de réponse. Voir 6.4.5.4.

6.1.10.2.2 Message "File d'attente de sortie pleine" (oq-full)

Le message "File d'attente de sortie pleine", oq-full, doit être VRAI lorsque la file d'attente de sortie sera pleine. Le message oq-full doit être FAUX dans tous les autres cas.

6.1.11 *Contrôle de déclenchement*

Le contrôle de déclenchement, représenté à la figure 6-2, est facultatif. Il permet l'exécution des messages GET sur matériel à grande vitesse, sans enfreindre les règles de l'ordre d'exécution.

Le message GET est communément utilisé pour déclencher une action spécifique à l'appareil ou un jeu d'actions dans le cadre d'un appareil à une adresse. Il est également utilisé pour déclencher des actions synchronisées par des appareils à adresses multiples. Dans ce cas et dans d'autres cas, où il peut être nécessaire d'éviter le temps système d'exécution de logiciels et micrologiciels passé dans les blocs de tampon d'entrée, d'analyseur syntaxique, et de contrôle d'exécution, le concepteur de l'appareil peut choisir d'implémenter le bloc de contrôle de déclenchement.

Si le bloc de contrôle de déclenchement est implémenté, les messages GET ne sont pas envoyés directement au tampon d'entrée du contrôle E/S. Au lieu de cela, **get** est envoyé au bloc de contrôle de déclenchement qui soit envoie un message de **déclenchement** directement aux fonctions de l'appareil, soit place un GET dans le tampon d'entrée.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The device designer should provide an Output Queue large enough to handle reasonably long <RESPONSE MESSAGE> elements. The actual length chosen will depend on the characteristics of the **device**. To avoid overflowing a physical buffer, the device designer may defer the formatting of lengthy <RESPONSE MESSAGE> elements until the **controller** requests output. The device designer should ensure that the size and operation of the Output Queue are adequate to make a deadlock unlikely, see 6.5.7.4.

6.1.10.1 Output Queue Rules

The Output Queue shall be cleared when pon or **dcas** is TRUE. Note that this is not a Query Error. A Query Error shall be reported if the contents of the Output Queue are discarded for any other reason, see 6.3.1.7, 6.3.2.2, and 6.3.2.3.

6.1.10.2 Output Queue messages

6.1.10.2.1 Message Available message (MAV)

The Output Queue shall send the MAV message to the Status Reporting block. As long as the Output Queue contains one or more bytes, MAV shall be TRUE. MAV shall be FALSE when the Output Queue is empty except as indicated below. See 11.2.1.2.

When the **device** defers generation of Response Data until **brq** is TRUE, the MAV message shall be set TRUE when the device is ready to generate the Response Data. See 6.4.5.4.

6.1.10.2.2 Output Queue Full message (oq-full)

The Output Queue Full message, oq-full, shall be TRUE when the Output Queue is full. The oq-full message shall be FALSE at all other times.

6.1.11 *Trigger Control*

The Trigger Control block, shown in figure 6-2, is optional. It allows for high-speed hardware execution of GET messages without violating the rules of execution order.

The GET message is commonly used to trigger a device-specific action or set of actions within a single addressed device. It is also used to trigger synchronized actions across multiple addressed devices. In this and other cases where it may be necessary to avoid the software/firmware execution time overhead incurred in the Input Buffer, Parser, and Execution Control blocks, the device designer may choose to implement the Trigger Control block.

If the Trigger Control block is implemented, GET messages are not sent directly to the Input Buffer from the I/O Control. Instead **get** is sent to the Trigger Control Block which either sends a **trigger** message directly to the Device Functions, or places a GET in the Input Buffer.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

6.1.11.1 Règles du contrôle de déclenchement

Le bloc de contrôle de déclenchement doit détecter le message **get** et vérifier **ib-empty** (du tampon d'entrée), **p-idle** (de l'analyseur syntaxique), et **ec-idle** (du contrôle d'exécution). Si ces messages sont tous les quatre VRAIS, le bloc de contrôle de déclenchement doit envoyer le message de **déclenchement** aux fonctions de l'appareil. Si **ib-empty**, ou **p-idle**, ou **ec-idle** est FAUX, le bloc de contrôle de déclenchement doit placer immédiatement le message GET dans le tampon d'entrée.

NOTE - L'implémentation de ce bloc de contrôle de déclenchement facultatif n'affecte pas la réponse de l'**appareil** à GET et ne mitige pas non plus l'exigence d'un traitement séquentiel des messages GET et DAB. Ce bloc facilite seulement l'exploitation à grande vitesse.

6.1.12 *Contrôle d'échanges de messages*

Le contrôle d'échanges de messages représente l'interconnexion de messages de contrôle entre la file d'attente de sortie, le formateur de réponse, le tampon d'entrée, l'analyseur syntaxique, le contrôle d'exécution, le contrôle E/S, et les blocs de fonctions des appareils.

6.2 *Vue d'ensemble du protocole*

Cette vue d'ensemble du protocole décrit le fonctionnement normal du protocole d'échange de messages de l'appareil. Elle n'est pas une spécification détaillée du protocole, pas plus qu'elle ne couvre les exceptions du protocole. La spécification détaillée du protocole d'échange de messages de l'appareil commence en 6.3.

La figure 6-3 représente les états et transitions d'échanges de messages rencontrés pendant le fonctionnement normal de l'appareil. Elle ne représente pas certains états et transitions nécessaires associés à un redressement d'erreur. La figure 6-4 et le paragraphe 6.3 décrivent le schéma complet d'état d'échanges de messages, les lignes en pointillé indiquant les cas d'exception ou d'erreur.

Les figures 6-3 et 6-4 utilisent un format différent de la CEI 625-1 pour souligner que le contrôle d'échange de messages s'effectue à un niveau supérieur à la CEI 625-1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.1.11.1 Trigger Control rules

The Trigger Control block shall detect the **get** message and check **ib-empty** (from the Input Buffer), **p-idle** (from the Parser), and **ec-idle** (from the Execution Control). If all four of these messages are TRUE, the Trigger Control block shall send the **trigger** message to the Device Functions. If either **ib-empty**, **p-idle**, or **ec-idle** is FALSE, the Trigger Control block shall place the GET message in the Input Buffer.

NOTE - The implementation of this optional Trigger Control block does not affect the **device's** response to GET nor relax the requirement of sequential processing of GET messages and DAB's. This block only facilitates high-speed operation.

6.1.12 *Message Exchange Control*

The Message Exchange Control represents the interconnection of control messages between the Output Queue, Response Formatter, Input Buffer, Parser, Execution Control, I/O Control, and Device Functions blocks.

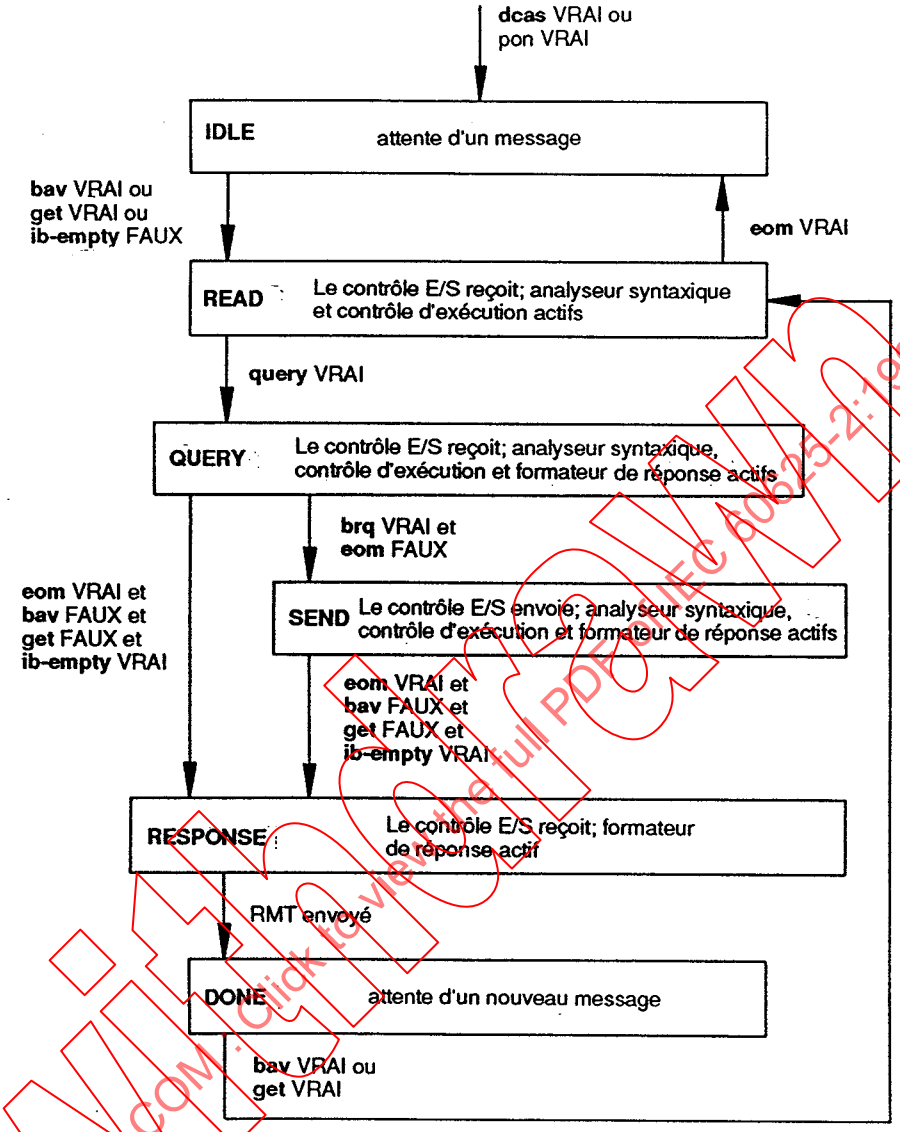
6.2 *Protocol overview*

This protocol overview describes the normal operation of the device message exchange protocol. It is not a detailed specification of the protocol, nor does it cover any protocol exceptions. The detailed specification of the device message exchange protocol begins in 6.3.

Figure 6-3 shows the Message Exchange states and transitions encountered during normal operation of the device. It does not show certain required states and transitions associated with error recovery. Figure 6-4 and 6.3 describe the complete Message Exchange State Diagram with the dotted lines indicating the exception or error cases.

Figures 6-3 and 6-4 use a different format from IEC 625-1 to reinforce that Message Exchange Control is done at a level higher than IEC 625-1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figure 6-3 – Schéma d'état de contrôle d'échanges de messages (simplifié)

6.2.1 *Initialisation*

Après la mise sous tension ou **dcas**, le contrôle d'échanges de messages attendra dans l'état INACTIF un <MESSAGE DE PROGRAMMATION> ou un message GET du **contrôleur**. L'appareil ne doit pas envoyer de <MESSAGE DE RÉPONSE> au **contrôleur** avant d'avoir reçu un message d'interrogation correct du **contrôleur**. Le contrôleur n'essaiera pas normalement de lire les données de l'**appareil** avant d'avoir envoyé un message d'interrogation à l'appareil.

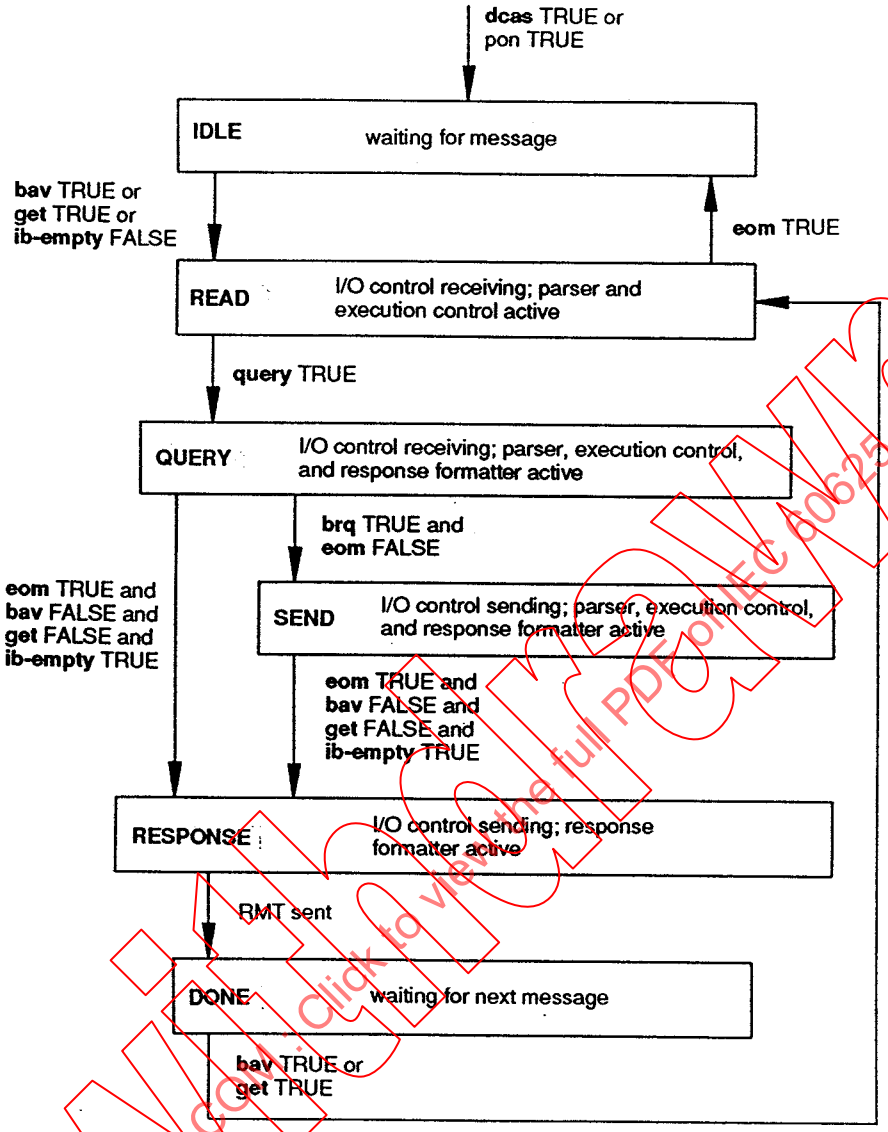


Figure 6-3 – Message Exchange Control State Diagram (simplified)

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.2.1 *Initialization*

After power on or **dcas**, the Message Exchange Control shall wait in the IDLE state for a <PROGRAM MESSAGE> or GET message from the **controller**. The device shall not send a <RESPONSE MESSAGE> to the **controller** until it receives a valid Query Message from the **controller**. The controller will not normally attempt to read data from the **device** until it has sent a Query Message to the device.

6.2.2 *Traitement des instructions*

Lorsque l'**appareil** reçoit un <MESSAGE DE PROGRAMMATION> du contrôleur pendant qu'il est dans l'état de LECTURE:

1) Le contrôle E/S place les octets du message dans le tampon d'entrée.

2) L'analyseur syntaxique enlève les octets du tampon d'entrée et identifie les composants syntaxiques du message suivant les règles de syntaxe de l'article 7.

3) Le contrôle d'exécution ordonne au bloc de fonctions de l'appareil d'exécuter les actions associées au message.

L'**appareil** peut accepter les octets d'un autre <MESSAGE DE PROGRAMMATION> et les placer dans le tampon d'entrée avant d'avoir fini de traiter les éléments de <MESSAGES DE PROGRAMMATION> antérieurs.

6.2.3 *Traitement des interrogations*

Le traitement des interrogations s'effectue dans les états QUERY, SEND, RESPONSE, et DONE. Si un <MESSAGE DE PROGRAMMATION> contient une ou plusieurs interrogations, l'**appareil** prépare un <MESSAGE DE RÉPONSE> pour le **contrôleur** et le place dans la file d'attente de sortie. Une fois que le contrôleur a fini d'envoyer le message d'interrogation à l'appareil, il adressera l'**appareil** pour parler et commencera à lire le <MESSAGE DE RÉPONSE>. Le contrôleur n'enverra pas normalement un autre <MESSAGE DE PROGRAMMATION> à l'**appareil** avant d'avoir fini de lire le <MESSAGE DE RÉPONSE>.

6.3 *Fonctionnement du contrôle d'échanges de messages*

Le schéma d'état de contrôle d'échanges de messages, figure 6-4, définit précisément comment les messages, décrits à 6.1.4, du contrôle E/S, font recevoir ou envoyer des messages à l'**appareil**. Les interactions entre les blocs dans l'interface de contrôle d'échanges de messages, figure 6-2, sont représentées en détail. La figure 6-3, représentée précédemment, comprend seulement les états et transitions rencontrés pendant le fonctionnement "normal" de l'appareil. La figure 6-4 comprend les états et transitions supplémentaires nécessaires pour détecter les erreurs du protocole d'échanges de messages. (Ces états et transitions supplémentaires sont représentés par un trait discontinu.)

6.3.1 *Etats de contrôle d'échange de messages*

A tout moment, l'**appareil** se trouve dans l'un des états d'échanges de messages. En fonction de l'état, divers éléments fonctionnels de la figure 6-2 sont actifs ou inactifs. Les transitions sont provoquées par la combinaison logique de messages des éléments fonctionnels.

6.3.1.1 Etat INACTIF

Dans l'état INACTIF, l'**appareil** attend un message du contrôleur. Le prochain octet de données reçu doit être interprété comme le début d'un nouveau <MESSAGE DE PROGRAMMATION>. La file d'attente de sortie est vide.

Le contrôle d'échange de messages doit entrer dans l'état LECTURE dans l'un des cas suivants:

1) Le contrôle E/S met **bav** VRAI, indiquant qu'un octet de données est disponible, ou

2) Le contrôle E/S met **get** VRAI, ou

IECNORM.COM : Click to view the full PDF of 625-2:1993

6.2.2 *Command processing*

When the **device** receives a <PROGRAM MESSAGE> from the controller while in the READ state:

1) The I/O Control places the bytes of the message into the Input Buffer.

2) The Parser removes the bytes from the Input Buffer and identifies the syntactic components of the message according to the syntax rules in clause 7.

3) The Execution Control directs the Device Functions block to perform the actions associated with the message.

The **device** may accept the bytes of another <PROGRAM MESSAGE> and place them in the Input Buffer before it has finished processing prior <PROGRAM MESSAGE> elements.

6.2.3 *Query processing*

Query processing is handled in the QUERY, SEND, RESPONSE, and DONE states. If a <PROGRAM MESSAGE> contains one or more queries, the **device** prepares a <RESPONSE MESSAGE> for the **controller** and places it in the Output Queue. After the controller finishes sending the Query Message to the device, it will address the **device** to talk and start reading the <RESPONSE MESSAGE>. The controller will not normally send another <PROGRAM MESSAGE> to the **device** until it has finished reading the <RESPONSE MESSAGE>.

6.3 *Message Exchange Control operation*

The Message Exchange Control State Diagram, figure 6-4, precisely defines how messages, described in 6.1.4, from the I/O Control cause the **device** to either receive or send messages. The interactions between the blocks in the Message Exchange Control Interface, figure 6-2, are shown in detail. Figure 6-3, shown before, includes only those states and transitions encountered during "normal" operation of the device. Figure 6-4 includes the additional states and transitions necessary to detect Message Exchange Protocol errors. (These additional states and transitions are shown with dashed lines.)

6.3.1 *Message Exchange Control states*

At any time, the **device** is in one of the Message Exchange states. Depending on the state, various functional elements from figure 6-2 are active or inactive. Transitions are caused by the logical combination of messages from the functional elements.

6.3.1.1 IDLE state

In the IDLE state the **device** is waiting for a message from the controller. The next data byte received shall be interpreted as the beginning of a new <PROGRAM MESSAGE>. The Output Queue is empty.

The Message Exchange Control shall enter the READ state when either:

1) The I/O control **sets bav** TRUE, indicating that a data byte is available, or

2) the I/O Control **sets get** TRUE, or

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

3) **ib-empty** est FAUX. Cette condition peut se produire lorsque l'**appareil** est entré dans l'état INACTIF à partir de l'état LECTURE ou BLOCAGE.

Le contrôle d'échanges de messages doit rester dans l'état INACTIF lorsque:

1) **brq** est VRAI, et
2) **bav** est FAUX, et
3) **get** est FAUX, et
4) **ib-empty** est VRAI.

Ceci se produit lorsque l'**appareil** a été adressé pour parler et n'a rien à dire. L'**appareil** doit exécuter l'action "NON TERMINÉ" décrite en 6.3.2.2.

Le contrôle d'échanges de messages doit rester dans l'état INACTIF lorsque **dcas** ou pon est VRAI. L'**appareil** doit exécuter l'action INITIALISER décrite en 6.3.2.1.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

3) **ib-empty** is FALSE. This condition can occur when the **device** had entered the IDLE state from the READ or DEADLOCK state.

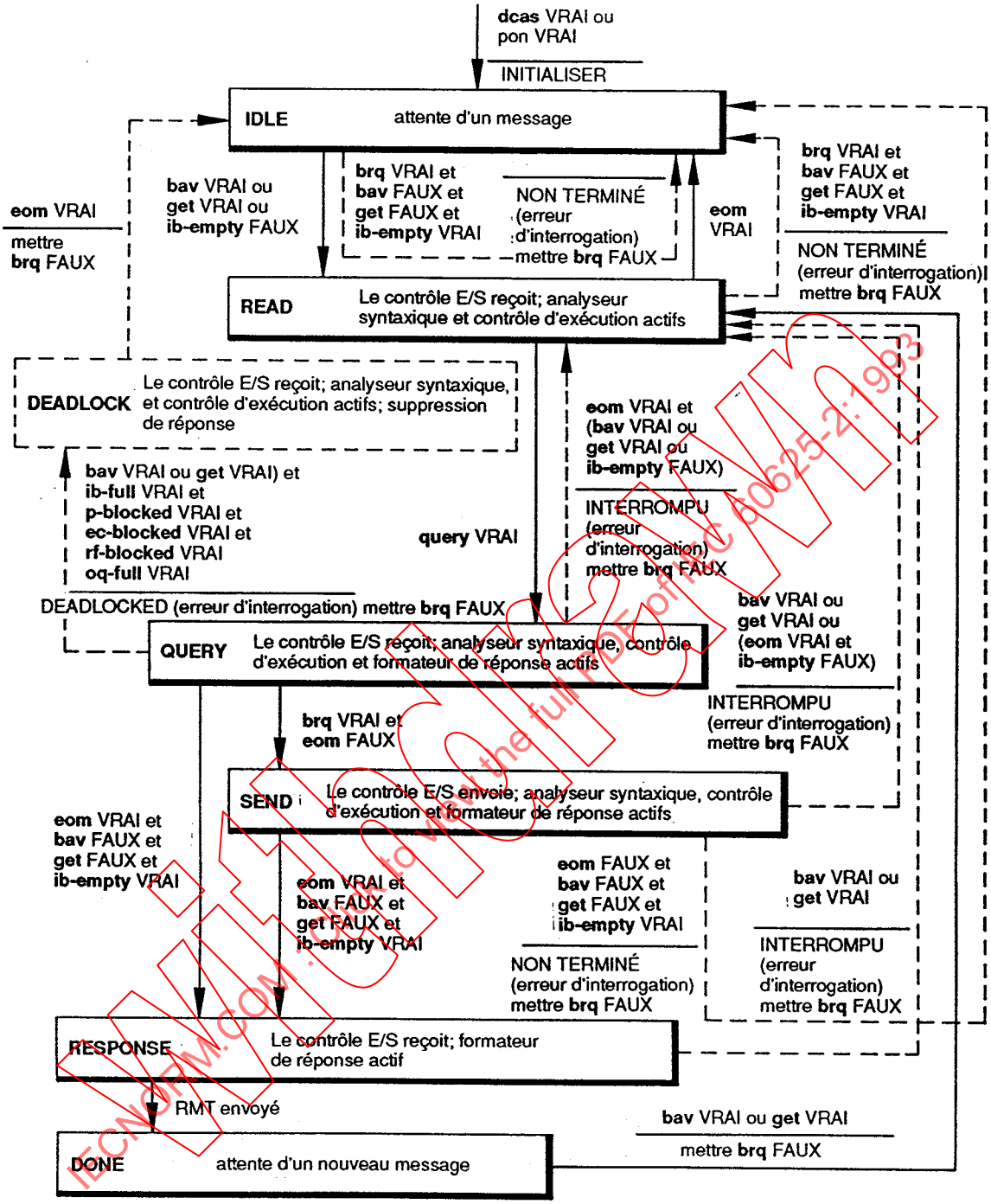
The Message Exchange Control shall stay in the IDLE state when:

1) **brq** is TRUE, and
2) **bav** is FALSE, and
3) **get** is FALSE, and
4) **ib-empty** is TRUE.

This occurs when the **device** has been addressed to talk and has nothing to say. The **device** shall perform the UNTERMINATED action described in 6.3.2.2.

The Message Exchange Control shall stay in the IDLE state when either the **dcas** or pon is TRUE. The **device** shall perform the INITIALIZE action described in 6.3.2.1.

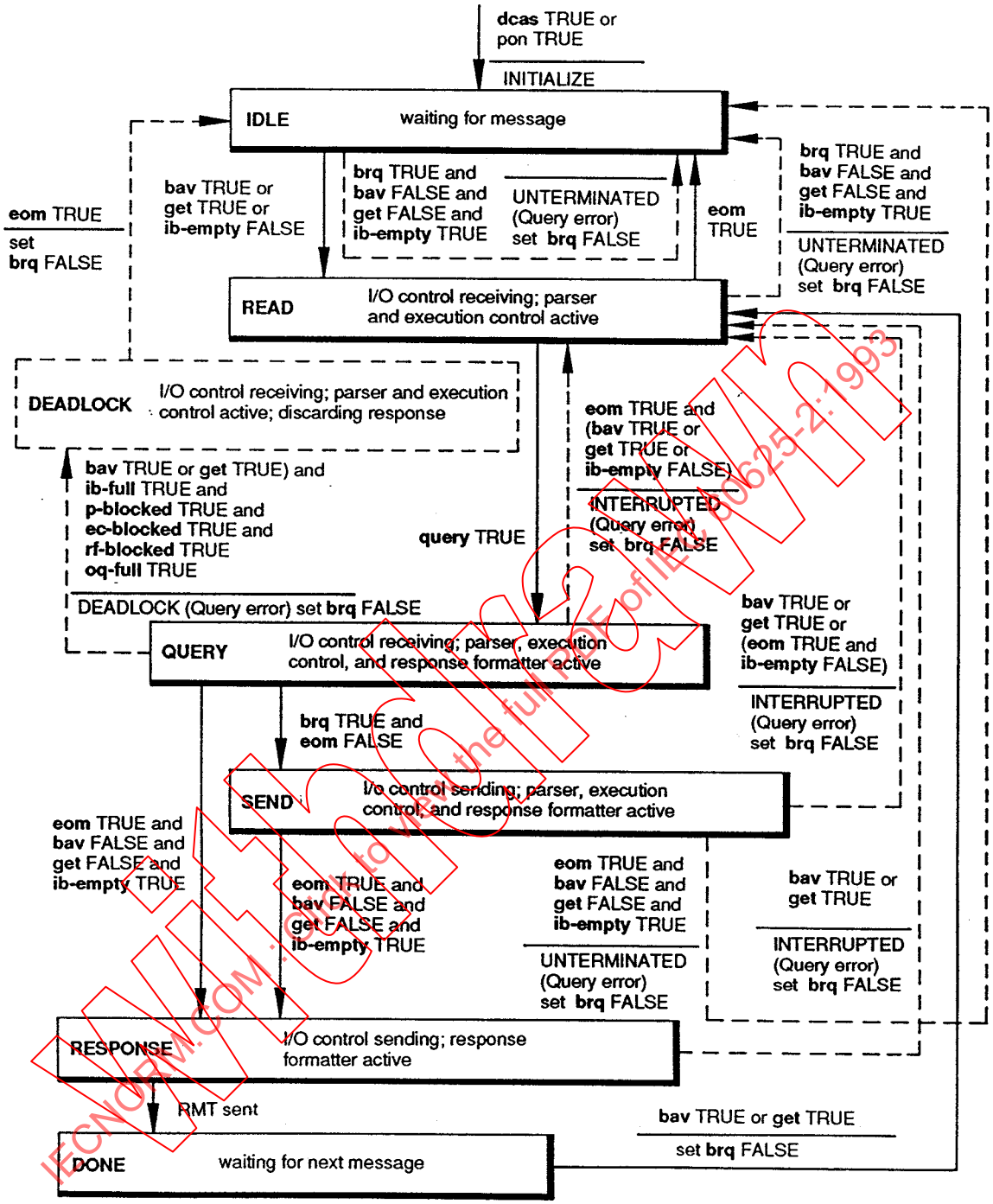
Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

NOTE - Le texte au-dessus de la ligne pour les lignes de transition en pointillé indique l'action que le dispositif accomplit pendant la transition.

Figure 6-4 – Schéma d'état de contrôle d'échanges de messages (complet)



NOTE - Text below the line for the dotted line transitions indicates an action which the device performs during the transition.

Figure 6-4 – Message Exchange Control State Diagram (complete)

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.3.1.2 Etat LECTURE

Dans l'état LECTURE, le contrôle E/S doit lire les octets de données, les messages GET et END du bus CEI 625-2 et les placera dans le tampon d'entrée, comme décrit à 6.1.4. L'analyseur syntaxique et le contrôle d'exécution sont actifs et la file d'attente de sortie est vide.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF lorsque **dcas** ou pon est VRAI. L'**appareil** exécutera l'action INITIALISER décrite à 6.3.2.1.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF dans l'une des conditions suivantes:

1) **brq** est VRAI, et
2) **bav** est FAUX, et
3) **get** est FAUX, et
4) **ib-empty** est VRAI.

L'appareil doit exécuter l'action "NON TERMINÉ" décrite à 6.3.2.2.

NOTE - Le fait d'entrer dans l'état INACTIF sans terminer le <MESSAGE DE RÉPONSE> pourrait résulter en un délai d'attente du contrôleur lorsque les données attendues ne sont pas envoyées. Lorsque ceci se produit, on peut utiliser un *ESR? (interrogation du registre d'état d'événements standard, 10.12) pour déterminer qu'aucune donnée n'a été envoyée parce qu'une erreur d'interrogation s'était produite.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF lorsque l'analyseur syntaxique met **eom** VRAI.

Le contrôle d'échanges de messages doit entrer dans l'état INTERROGATION lorsque l'analyseur syntaxique met **query** VRAI. Voir en 6.1.6.2.3.

6.3.1.3 Etat INTERROGATION

Dans l'état INTERROGATION, l'analyseur syntaxique a reconnu une interrogation à l'intérieur d'un MESSAGE DE PROGRAMMATION. Le contrôle E/S doit continuer à lire les octets de données du bus CEI 625-1 et les placer dans le tampon d'entrée, comme décrit à 6.1.4.2.4. L'analyseur syntaxique et le contrôle d'exécution sont actifs. L'**appareil** peut formater son <MESSAGE DE RÉPONSE> à ce moment-là, ou peut attendre jusqu'à ce que le contrôle d'échanges de messages soit dans les états ENVOI ou RÉPONSE; voir en 6.4.5.4.

Le contrôle d'échanges de messages doit entrer dans l'état ENVOI lorsque **brq** est VRAI et qu'un **eom** n'a pas été analysé. Cette transition indique que le contrôleur a commencé à lire la réponse.

Le contrôle d'échanges de messages doit entrer dans l'état RÉPONSE lorsque l'analyseur syntaxique met **eom** VRAI, **bav** est FAUX, **get** est FAUX, et **ib-empty** est VRAI. Cette transition indique que l'**appareil** a fini de traiter le message d'interrogation et attend que le **contrôleur** lise la réponse.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.3.1.2 READ state

In the READ state the I/O Control shall read data bytes, GET, and END messages from the IEC 625-2 bus and place them in the Input Buffer as described in 6.1.4. The Parser and Execution Control are active and the Output Queue is empty.

The Message Exchange Control shall enter the IDLE state when either **dcas** or pon is TRUE. The **device** shall perform the INITIALIZE action described in 6.3.2.1.

The Message Exchange Control shall enter the IDLE state when each of the following conditions occur:

1) **brq** is TRUE, and
2) **bav** is FALSE, and
3) **get** is FALSE, and
4) **ib-empty** is TRUE.

The device shall perform the UNTERMINATED action described in 6.3.2.2.

NOTE - Entering the IDLE state without terminating the <RESPONSE MESSAGE> could result in a controller time out when the expected data is not sent. When this happens an *ESR? (Standard Event Status Register Query, 10.12) can be used to determine that no data was sent because a Query Error had occurred.

The Message Exchange Control shall enter the IDLE state when the Parser sets **eom** TRUE.

The Message Exchange Control shall enter the QUERY state when the Parser sets **query** TRUE. See 6.1.6.2.3.

6.3.1.3 QUERY state

In the QUERY state the Parser has recognized a valid query within a <PROGRAM MESSAGE>. The I/O Control shall continue to read data bytes from the IEC 625-1 bus and place them in the Input Buffer as described in 6.1.4.2.4. The Parser and Execution Control are active. The **device** may format its <RESPONSE MESSAGE> at this time, or may wait until the Message Exchange Control is in the SEND or RESPONSE states; see 6.4.5.4.

The Message Exchange Control shall enter the SEND state when **brq** is TRUE and an **eom** has not been parsed. This transition indicates that the controller has started to read the response.

The Message Exchange Control shall enter the RESPONSE state when the Parser sets **eom** TRUE, **bav** is FALSE, **get** is FALSE, and **ib-empty** is TRUE. This transition indicates that the **device** has finished processing the Query Message and is waiting for the **controller** to read the response.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Le contrôle d'échanges de messages doit entrer dans l'état BLOCAGE dans toutes les conditions suivantes:

1) le contrôle E/S met **bav** ou **get** VRAI, indiquant que le **contrôleur** attend d'envoyer des données à l'**appareil**, et
2) **ib-full** est VRAI, et
3) **p-blocked** est VRAI, et
4) **ec-blocked** est VRAI, et
5) **rf-blocked** est VRAI, et
6) **oq-full** est VRAI.

Cette transition doit faire mettre à l'**appareil** le bit d'erreur d'interrogation dans le registre d'état d'événements standard, effacer la file d'attente de sortie, et remettre à zéro le formateur de réponse.

Le contrôle d'échanges de messages doit entrer dans l'état LECTURE dans les conditions suivantes:

1) **eom** est VRAI, et
2) **bav** est VRAI, et
3) **get** est VRAI, et
4) **ib-empty** est FAUX.

L'**appareil** doit exécuter l'action "INTERROMPU" décrite en 6.3.2.3.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF lorsque **dcas** ou pon est VRAI. L'**appareil** doit exécuter l'action INITIALISER décrite en 6.3.2.1.

6.3.1.4 Etat ENVOI

Dans l'état ENVOI, le **contrôleur** a commencé à lire le <MESSAGE DE RÉPONSE> de l'**appareil**. L'**appareil** continue à analyser et à exécuter le message d'interrogation stocké dans le tampon d'entrée. Le formateur de réponse prépare le <MESSAGE DE RÉPONSE>, et le place dans la file d'attente de sortie. Le contrôle E/S doit envoyer les octets de données de la file d'attente de sortie au **contrôleur**, comme décrit en 6.1.4.2.3.

Le contrôle d'échanges de messages doit entrer dans l'état de RÉPONSE lorsque **eom** est VRAI, **bav** est FAUX, **get** est FAUX, et **ib-empty** est VRAI.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF dans toutes les conditions suivantes:

1) **eom** est VRAI, et
2) **bav** est FAUX, et
3) **get** est FAUX, et
4) **ib-empty** est VRAI.

L'**appareil** doit exécuter l'action "NON TERMINÉ" décrite en 6.3.2.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The Message Exchange Control shall enter the DEADLOCK state when all of the following occur:

1) the I/O Control sets **bav** or **get** TRUE, indicating that the **controller** is waiting to send data to the **device**, and
2) **ib-full** is TRUE, and
3) **p-blocked** is TRUE, and
4) **ec-blocked** is TRUE, and
5) **rf-blocked** is TRUE, and
6) **oq-full** is TRUE.

This transition shall cause the **device** to set the Query Error bit in the Standard Event Status Register, clear the Output Queue, and reset the Response Formatter.

The Message Exchange Control shall enter the READ state when the following conditions occur:

1) **eom** is TRUE and either
2) **bav** is TRUE, or
3) **get** is TRUE, or
4) **ib-empty** is FALSE.

The **device** shall perform the INTERRUPTED action described in 6.3.2.3.

The Message Exchange Control shall enter the IDLE state when either **dcas** or pon is TRUE. The **device** shall perform the INITIALIZE action described in 6.3.2.1.

6.3.1.4 SEND state

In the SEND state the **controller** has started to read the <RESPONSE MESSAGE> from the **device**. The **device** continues to parse and execute the Query Message stored in the Input Buffer. The Response Formatter prepares the <RESPONSE MESSAGE>, and places it in the Output Queue. The I/O Control shall send data bytes from the Output Queue to the **controller** as described in 6.1.4.2.3.

The Message Exchange Control shall enter the RESPONSE state when **eom** is TRUE, **bav** is FALSE, **get** is FALSE, and **ib-empty** is TRUE.

The Message Exchange Control shall enter the IDLE state when all of the following conditions occur:

1) **eom** is TRUE, and
2) **bav** is FALSE, and
3) **get** is FALSE, and
4) **ib-empty** is TRUE.

The **device** shall perform the UNTERMINATED action described in 6.3.2.2.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Le contrôle d'échanges de messages doit entrer dans l'état LECTURE dans l'une des conditions suivantes:

1) **bav** est VRAI, ou
2) **get** est VRAI, ou
3) L'analyseur syntaxique met **eom** VRAI et **ib-empty** est FAUX.

Cette condition indique que le **contrôleur** a commencé à envoyer un nouveau message avant de lire la totalité de la réponse. L'**appareil** doit exécuter l'action "INTERROMPU" décrite en 6.3.2.3.

L'**appareil** doit entrer dans l'état INACTIF lorsque **dcas** ou pon est VRAI. L'appareil doit exécuter l'action INITIALISER décrite en 6.3.2.1.

6.3.1.5 Etat RÉPONSE

Dans l'état RÉPONSE, l'analyseur syntaxique a reçu un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> et l'**appareil** envoie le <MESSAGE DE RÉPONSE> complet au **contrôleur**. Le formateur de réponse formate le <MESSAGE DE RÉPONSE> et le contrôleur E/S envoie les octets de données de la file d'attente de sortie au **contrôleur**, comme décrit en 6.1.4.2.3.

Le contrôle d'échanges de messages doit entrer dans l'état EXÉCUTÉ lorsque le <MESSAGE DE RÉPONSE> complet, y compris le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>, a été envoyé.

Le contrôle d'échanges de messages doit entrer dans l'état LECTURE dans l'une des conditions suivantes:

1) **bav** est VRAI, ou
2) **get** est VRAI.

Cette transition indique que le **contrôleur** a essayé d'envoyer un nouveau message avant de lire le <MESSAGE DE RÉPONSE> complet. L'**appareil** doit exécuter l'action "INTERROMPU" décrite en 6.3.2.3.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF lorsque **dcas** ou pon est VRAI. L'**appareil** doit exécuter l'action INITIALISER, décrite en 6.3.2.1.

Un **appareil** peut n'avoir rien à dire pendant que le contrôle d'échanges de messages est dans l'état RÉPONSE si toutes les interrogations dans le message d'interrogation n'ont pas généré de <DONNÉES DE RÉPONSE> en raison d'erreurs de syntaxe (instruction), sémantique (exécution), ou spécifiques à l'appareil. L'**appareil** ne doit pas envoyer de <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE> au **contrôleur**. Le contrôle d'échanges de messages doit rester dans l'état RÉPONSE jusqu'à ce qu'il reçoive un octet de données ou un message GET du contrôleur ou que le message pon ou **dcas** devienne VRAI.

6.3.1.6 Etat EXÉCUTÉ

Dans l'état EXÉCUTÉ, l'**appareil** a fini d'envoyer un <MESSAGE DE RÉPONSE> au **contrôleur** et attend un nouveau <MESSAGE DE PROGRAMMATION> ou GET du **contrôleur**. L'analyseur syntaxique, le contrôle d'exécution et le formateur de réponse sont inactifs et le tampon d'entrée et la file d'attente de sortie sont vides.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The Message Exchange Control shall enter the READ state when any of the following conditions occur:

1) **bav** is TRUE, or
2) **get** is TRUE, or
3) the Parser sets **eom** TRUE and **ib-empty** is FALSE.

This condition indicates that the **controller** started sending a new message before reading all of the response. The **device** shall perform the INTERRUPTED action described in 6.3.2.3.

The **device** shall enter the IDLE state when either **dcas** or pon is TRUE. The device shall perform the INITIALIZE action described in 6.3.2.1.

6.3.1.5 RESPONSE state

In the RESPONSE state the Parser has received a <PROGRAM MESSAGE TERMINATOR> and the **device** is sending the complete <RESPONSE MESSAGE> to the **controller**. The Response Formatter formats the <RESPONSE MESSAGE> and the I/O Control sends data bytes from the Output Queue to the **controller** as described in 6.1.4.2.3.

The Message Exchange Control shall enter the DONE state when the complete <RESPONSE MESSAGE> including the <RESPONSE MESSAGE TERMINATOR> has been sent.

The Message Exchange Control shall enter the READ state when either of the following conditions occur:

1) **bav** is TRUE, or
2) **get** is TRUE.

This transition indicates that the **controller** attempted to send a new message before reading the complete <RESPONSE MESSAGE>. The **device** shall perform the INTERRUPTED action described in 6.3.2.3.

The Message Exchange Control shall enter the IDLE state when either **dcas** or pon is TRUE. The **device** shall perform the INITIALIZE action described in 6.3.2.1.

A **device** may have nothing to say while the Message Exchange Control is in the RESPONSE state if all of the queries in the Query Message failed to generate any <RESPONSE DATA> because of syntax (Command), semantic (Execution), or Device-specific errors. The **device** shall not send a <RESPONSE MESSAGE TERMINATOR> to the controller. The Message Exchange Control shall remain in the RESPONSE state until it either receives a data byte or GET message from the **controller** or the pon or **dcas** message becomes TRUE.

6.3.1.6 DONE state

In the DONE state the **device** has finished sending a <RESPONSE MESSAGE> to the **controller** and is waiting for a new <PROGRAM MESSAGE> or GET from the **controller**. The Parser, Execution Control, and Response Formatter are inactive and the Input Buffer and Output Queue are empty.

IECNORM.COM : Click to view the full PDF of IEC 625-2:1993
Withdrawn

Le contrôle d'échanges de messages doit entrer dans l'état LECTURE et mettre **brq** FAUX dans l'une des conditions suivantes:

1) **bav** est VRAI, indiquant qu'un nouveau <MESSAGE DE PROGRAMMATION> a été commencé, ou

2) **get** est VRAI.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF lorsque **dcas** ou pon est VRAI. L'**appareil** doit exécuter l'action INITIALISER décrite en 6.3.2.1.

6.3.1.7 Etat BLOCAGE

Dans l'état BLOCAGE, on a demandé à l'**appareil** de mettre en mémoire tampon plus de données qu'il n'a de place pour les stocker. La file d'attente de sortie était pleine, bloquant le formateur de réponse, le contrôle d'exécution et l'analyseur syntaxique. Le tampon d'entrée était plein, et le **contrôleur** attendait d'envoyer plus d'octets de données à l'**appareil**.

Le contrôle d'échanges de messages interrompt le blocage en effaçant la file d'attente de sortie et en remettant à zéro le formateur de réponse au moment où il entre dans l'état BLOCAGE (transition d'INTERROGATION à BLOCAGE, voir 6.3.1.3). Le contrôle d'échanges de messages met également le bit d'erreur d'interrogation dans le registre d'état d'événements standard. Pendant qu'il est dans l'état BLOCAGE, le contrôle d'échanges de messages doit continuer à analyser et à exécuter les éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> comme d'habitude, sauf qu'il doit supprimer les réponses aux interrogations plutôt que de les placer dans la file d'attente de sortie.

Le contrôle d'échanges de messages doit entrer dans l'état INACTIF et mettre **brq** FAUX dans l'une des conditions suivantes:

1) L'analyseur syntaxique met **eom** VRAI. Lorsque le **contrôleur** essaie de lire le <MESSAGE DE RÉPONSE>, l'**appareil** n'envoie pas d'octets de données. Noter que le fait de ne pas envoyer d'octets de données peut mettre le **contrôleur** en attente. Lorsque ceci se produit, on peut utiliser un *ESR? (interrogation du registre d'état d'événements standard, voir 10.12) pour déterminer qu'aucune donnée n'a été envoyée parce qu'une erreur d'interrogation s'était produite.

2) Soit **dcas**, soit pon est VRAI. L'**appareil** doit exécuter l'action INITIALISER décrite en 6.3.2.1.

6.3.2 *Actions de transition de contrôle d'échange de messages*

Certaines transitions entre les états sont assez importantes pour être citées. Les actions provoquées par les transitions sont essentielles pour la fiabilité de la détection, de la signalisation et du redressement des erreurs.

6.3.2.1 Action INITIALISER

L'action INITIALISER doit être exécutée lorsque la voie de transmission, CEI 625-2, doit être effacée.

L'**appareil** exécutera chacune des actions suivantes:

1) Effacer le tampon d'entrée, et

2) Effacer la file d'attente de sortie, et

3) Remettre à zéro l'analyseur syntaxique, et

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

The Message Exchange Control shall enter the READ state and set **brq** FALSE when either:

1) **bav** is TRUE, indicating that a new <PROGRAM MESSAGE> has been started, or

2) **get** is TRUE.

The Message Exchange Control shall enter the IDLE state when either **dcas** or pon is TRUE. The **device** shall perform the INITIALIZE action described in 6.3.2.1.

6.3.1.7 DEADLOCK State

In the DEADLOCK state the **device** has been asked to buffer more data than it has room to store. The Output Queue was full, blocking the Response Formatter, Execution Control, and Parser. The Input Buffer was full, and the **controller** was waiting to send more data bytes to the device.

The Message Exchange Control breaks the deadlock by clearing the Output Queue and resetting the Response Formatter as it enters the DEADLOCK state (transition from QUERY to DEADLOCK, see 6.3.1.3). The Message Exchange Control also sets the Query Error bit in the Standard Event Status Register. While in the DEADLOCK state the Message Exchange Control shall continue to parse and execute <PROGRAM MESSAGE UNIT> elements as usual, except that it shall discard query responses rather than placing them into the Output Queue.

The Message Exchange Control shall enter the IDLE state and set **brq** FALSE when either of the following conditions occur:

1) The Parser sets **eom** TRUE. When the **controller** attempts to read the <RESPONSE MESSAGE>, the **device** shall not send any data bytes. Note that not sending any data bytes may cause the **controller** to time out. When this happens, an *ESR? (Standard Event Status Register Query, see 10.12) can be used to determine that no data was sent because a Query Error had occurred.

2) Either **dcas** or pon is TRUE. The **device** shall perform the INITIALIZE action described in 6.3.2.1.

6.3.2 *Message Exchange Control transition actions*

Certain transitions among the states are important enough to be named. The actions caused by the transitions are essential for reliable error detection, reporting, and recovery.

6.3.2.1 INITIALIZE action

The INITIALIZE action is executed when the IEC 625-2 communication channel is to be cleared.

The **device** shall perform each of the following actions:

1) Clear the Input Buffer, and
2) Clear the Output Queue, and
3) Reset the Parser, and

IECNORM.COM : Click to view the full PDF of IEC 625-2:1993

4) Remettre à zéro le contrôle d'exécution, et
5) Remettre à zéro le formateur de réponse.

Aucune condition d'erreur ne doit être signalée.

6.3.2.2 Action "NON TERMINÉ"

L'action "NON TERMINÉ" est exécutée lorsque le **contrôleur** essaie de lire un <MESSAGE DE RÉPONSE> de l'**appareil** sans avoir d'abord envoyé un message d'interrogation complet, y compris le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>, à l'**appareil**.

L'**appareil** doit exécuter chacune des actions suivantes:

1) Mettre le bit d'erreur d'interrogation dans le registre d'état d'événement standard.
2) Effacer la file d'attente de sortie.
3) Exécuter facultativement des éléments quelconques d'<UNITÉ DE MESSAGE DE PROGRAMMATION> du message incomplet. S'il exécute l'un quelconque des éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION>, il doit également exécuter tous les éléments antérieurs d'<UNITÉ DE MESSAGE DE PROGRAMMATION> du même message.
4) Supprimer toute <UNITÉ DE MESSAGE DE PROGRAMMATION> partiellement analysée de sorte que l'analyseur syntaxique sera prêt à analyser une nouvelle <UNITÉ DE MESSAGE DE PROGRAMMATION>.
5) Mettre **brq** FAUX.

6.3.2.3 Action "INTERROMPU"

L'action "INTERROMPU" est exécutée lorsque l'**appareil** est interrompu par un nouveau <MESSAGE DE PROGRAMMATION> avant d'avoir fini d'envoyer un <MESSAGE DE RÉPONSE>.

L'**appareil** doit exécuter chacune des actions suivantes:

1) Mettre le bit d'erreur d'interrogation dans le registre d'état d'événement standard, et
2) Effacer la file d'attente de sortie, et
3) Remettre à zéro le contrôle d'exécution et le formateur de réponse de sorte que, lorsque l'appareil recevra une nouvelle <UNITÉ DE MESSAGE D'INTERROGATION>, le <MESSAGE DE RÉPONSE> correct sera envoyé, et
4) Mettre **brq** FAUX.

6.4 *Règles de protocole*

6.4.1 *Transfert de messages de programmation*

L'**appareil** doit échanger des messages avec le **contrôleur** conformément à CEI 625-1 et aux règles de syntaxe des articles 7 et 8.

L'**appareil** ne doit pas traiter IFC ou toute instruction ou adresse CEI 625-1 conforme à ATN, sauf SDC adressé à l'**appareil**, DCL ou GET, comme un caractère d'arrêt de message. Le **contrôleur** peut librement intercaler des instructions (sauf DCL, SDC et GET) et messages conformes à ATN adressés aux autres appareils parmi les octets de messages adressés à ou venant de l'**appareil** sans affecter le contenu ou l'interprétation d'un message.

IECNORM.COM : Click to view the full PDF of IEC 625-2:1993
Withdrawn

4) Reset the Execution Control, and

5) Reset the Response Formatter.

No error condition shall be reported.

6.3.2.2 UNTERMINATED action

The UNTERMINATED action is executed when the **controller** attempts to read a <RESPONSE MESSAGE> from the **device** without first having sent a complete Query Message, including the <PROGRAM MESSAGE TERMINATOR>, to the **device**.

The **device** shall perform each of the following actions:

1) Set the Query Error bit in the Standard Event Status Register.

2) Clear the Output Queue.

3) Optionally execute any <PROGRAM MESSAGE UNIT> elements from the incomplete message. If it executes any of the <PROGRAM MESSAGE UNIT> elements, it shall also execute all previous <PROGRAM MESSAGE UNIT> elements from the same message.

4) Discard any partially parsed <PROGRAM MESSAGE UNIT> so that the Parser will be ready to parse a new <PROGRAM MESSAGE UNIT>.

5) Set **brq** FALSE.

6.3.2.3 INTERRUPTED action

The INTERRUPTED action is executed when the **device** is interrupted by a new <PROGRAM MESSAGE> before it finishes sending a <RESPONSE MESSAGE>.

The **device** shall perform each of the following actions:

1) Set the Query Error bit in the Standard Event Status Register, and

2) Clear the Output Queue, and

3) Reset the Execution Control and Response Formatter so that, when the device receives a new <QUERY MESSAGE UNIT>, the correct <RESPONSE MESSAGE> will be sent, and

4) Set **brq** FALSE.

6.4 *Protocol rules*

6.4.1 *Program message transfer*

The **device** shall exchange messages with the **controller** in accordance with IEC 625-1 and the syntax rules of clauses 7 and 8.

The device shall not treat IFC or any ATN-true IEC 625-1 command or address, except for SDC addressed to the **device**, DCL or GET, as a message terminator. The **controller** may freely intersperse ATN-true commands (except DCL, SDC, and GET) and messages addressed to other devices among the bytes of messages to or from the **device** without affecting the contents or interpretation of a message. Such commands may cause the

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

De telles instructions peuvent faire interrompre à l'appareil l'envoi d'un message de l'appareil au contrôleur, mais lorsque l'**appareil** reviendra en TACS, l'**appareil** doite reprendre l'envoi du reste du message. L'**appareil** ne doit pas non plus être perturbé en traitant un message du contrôleur à l'appareil qui est interrompu en raison des conditions mentionnées. L'**appareil** doit reprendre l'acceptation du reste du message lorsque l'**appareil** sera de nouveau adressé pour écouter et que ATN sera FAUX.

6.4.2 *Indépendance de la source des messages*

L'**appareil** doit interpréter les messages reçus sans faire référence à la source du message.

Les **appareils** (par exemple, répéteurs de bus ou instruments de diagnostic) peuvent enregistrer des adresses parleur et écouteur et peuvent envoyer un tel enregistrement en réponse à une interrogation, mais ne devront pas ignorer les messages ou les interpréter différemment sur la base de telles informations.

6.4.3 *Séquence d'échange de messages*

Le **contrôleur** et l'**appareil** échangent des éléments de <MESSAGES DE PROGRAMMATION> et <MESSAGES DE RÉPONSE> complets, à l'exception de ceux décrits dans le cadre des "Exceptions au Protocole", voir 6.5.

Les échanges de messages sont lancés par le **contrôleur**. L'**appareil** ne doit pas générer de <MESSAGE DE RÉPONSE> avant d'avoir reçu l'un des messages d'interrogation suivants:

1) Une séquence d'octets de données constituant un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct qui n'est pas une étiquette de macro.

2) Un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct qui est une étiquette de macro, où la macro est définie pour inclure une <UNITÉ DE MESSAGE D'INTERROGATION>.

3) Un message GET ou une instruction *TRG, où l'appareil implémente l'instruction commune *DDT et l'action de déclenchement de l'appareil définie par l'instruction *DDT comprend une <UNITÉ DE MESSAGE D'INTERROGATION>.

4) Un message GET ou instruction *TRG, ou l'appareil n'implémente pas l'instruction commune *DDT et l'action de déclenchement de l'appareil spécifiée par le concepteur de l'appareil doit générer un message de réponse.

Voir 6.1.6.2.3.

L'**appareil** doit générer exactement un <MESSAGE DE RÉPONSE> pour chaque message d'interrogation qu'il recevra. Une <UNITÉ DE MESSAGE D'INTERROGATION> peut renvoyer un ou plusieurs éléments d'<UNITÉ DE MESSAGE DE RÉPONSE>.

EXEMPLE:

On demande à un DVM de prendre cinq lectures séparées et de les renvoyer en un seul <MESSAGE DE RÉPONSE>.

Instruction: MEAS_FIVE? <PMT>

Réponse: 6; 2; 9; 2; 4 <RMT>

On demande à un DVM de renvoyer toutes ses informations de réglage concernant sa fonction OHMS.

Instruction: OHMS_SET? <PMT>

Réponse: RANGE 100; AVGS 10;. . .; AUTOZERO 1 <RMT>

La documentation de l'appareil doit signaler clairement toute <UNITÉ DE MESSAGE D'INTERROGATION> qui renvoie plus d'une <UNITÉ DE MESSAGE DE RÉPONSE>.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

device to suspend sending a device-to-controller message, but when the **device** re-enters TACS, the **device** shall resume sending the rest of the message. The **device** shall also be undisturbed in processing a controller-to-device message that is suspended due to the conditions mentioned. The **device** shall resume accepting the rest of a message when the **device** is again addressed to listen and ATN is set FALSE.

6.4.2 *Message source independence*

The **device** shall interpret received messages without reference to the source of the message.

Devices (e.g. bus repeaters or diagnostic instruments) may record talker and listener addresses and may return such a record in response to a query, but shall not ignore messages or interpret them differently on the basis of such information.

6.4.3 *Message exchange sequence*

The **controller** and **device** exchange complete <PROGRAM MESSAGE> and <RESPONSE MESSAGE> elements except as described under "Protocol exceptions", see 6.5.

Message exchanges are initiated by the **controller**. The **device** shall not generate a <RESPONSE MESSAGE> until it receives one of the following Query Messages:

1) A sequence of data bytes making up a valid <QUERY PROGRAM HEADER> which is not a macro label.
2) A valid <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> which is a macro label, where the macro is defined to include a <QUERY MESSAGE UNIT>.
3) A GET message or *TRG command, where the device implements the *DDT common command and the device trigger action defined by the *DDT command includes a <QUERY MESSAGE UNIT>.
4) A GET message or *TRG command, where the device does not implement the *DDT common command and the device trigger action specified by the device designer is to generate a response message.

See 6.1.6.2.3.

The **device** shall generate exactly one <RESPONSE MESSAGE> for each Query Message it receives. A <QUERY MESSAGE UNIT> may return one or more <RESPONSE MESSAGE UNIT> elements.

EXAMPLE:

A DVM is asked to take five separate readings and return them in a single <RESPONSE MESSAGE>.

Command: MEAS_FIVE? <PMT>
Response: 6; 2; 9; 2; 4 <RMT>

A DVM is asked to return all its setting information about its OHMS function.

Command: OHMS_SET? <PMT>
Response: RANGE 100; AVGS 10;. . .; AUTOZERO 1 <RMT>

Device documentation shall clearly indicate any <QUERY MESSAGE UNIT> which returns more than one <RESPONSE MESSAGE UNIT>.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

L'**appareil** doit interpréter un message GET de la même façon qu'un élément de MESSAGE DE PROGRAMMATION. L'**appareil** doit indiquer une erreur d'instruction s'il rencontre un message GET entre le premier octet d'un <MESSAGE DE PROGRAMMATION> et le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>.

Il n'est pas recommandé que le **contrôleur** essaie de lire une partie quelconque d'un <MESSAGE DE RÉPONSE> de l'**appareil** avant d'avoir envoyé le message d'interrogation entier à l'**appareil**. Le **contrôleur** ne devrait pas envoyer une partie quelconque d'un autre <MESSAGE DE PROGRAMMATION> à l'**appareil** avant d'avoir lue le <MESSAGE DE RÉPONSE> entier de l'**appareil**. L'**appareil** doit détecter que le **contrôleur** ne suit pas ce protocole et agit comme cela est décrit dans le cadre des "Exceptions au Protocole", voir 6.5.

Après pon ou après avoir reçu **dcas**, l'**appareil** doit attendre un <MESSAGE DE PROGRAMMATION> du **contrôleur**. Au cours de cette période, il peut exciter le signal SRQ (demande de service) CEI 625-1 et doit interpréter et répondre aux messages d'interface CEI 625-1 (par exemple, interrogation série), mais ne doit pas générer automatiquement des éléments de <MESSAGE DE RÉPONSE>.

A n'importe quel moment donné, l'**appareil** effectue l'une des actions suivantes:

1) L'**appareil** est inactif, attendant un message du **contrôleur**. Le contrôleur peut être inactif ou peut être en train de communiquer avec quelque(s) autre(s) appareil(s) sur le **bus du système**. (Le contrôle d'échanges de messages est dans l'état INACTIF ou EXÉCUTÉ).

2) L'**appareil** est en train de recevoir un message du **contrôleur**. Le **contrôleur** peut interrompre le transfert pour exécuter une interrogation série ou pour communiquer avec d'autres appareils, mais ne doit pas, en principe, essayer de lire des octets de données de l'**appareil** avant d'avoir envoyé un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>, voir 7.5, à l'**appareil**. (Le contrôle d'échanges de messages est dans l'état LECTURE, INTERROGATION, ou BLOCAGE).

3) L'**appareil** est en train d'envoyer un <MESSAGE DE RÉPONSE> au **contrôleur**. Le **contrôleur** peut interrompre le transfert pour exécuter une interrogation série ou pour communiquer avec d'autres appareils, mais ne doit pas, en principe, essayer d'envoyer des octets de données à l'**appareil** avant d'avoir reçu un <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>, voir 8.5, de l'**appareil**. (Le contrôle d'échanges de messages est dans l'état ENVOI ou RÉPONSE).

6.4.4 *Interrogations complexes*

Un ou plusieurs éléments d'<UNITÉ DE MESSAGE D'INTERROGATION> peuvent être placés dans un seul <MESSAGE DE PROGRAMMATION>, article 7. De même, l'appareil est autorisé à générer un ou plusieurs éléments d'<UNITÉ DE MESSAGE DE RÉPONSE> pour chaque <UNITÉ DE MESSAGE D'INTERROGATION> reçue. Le <SÉPARATEUR D'UNITÉS DE MESSAGE DE RÉPONSE> (point-virgule ";" voir 8.4.1) doit être utilisé pour séparer les éléments d'<UNITÉ DE MESSAGE DE RÉPONSE> à l'intérieur du <MESSAGE DE RÉPONSE> complexe complet. Le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>, défini en 8.5 par NL^END, doit terminer la dernière <UNITÉ DE MESSAGE DE RÉPONSE>. Etant donné que le caractère d'arrêt nécessaire pour une <UNITÉ DE MESSAGE DE RÉPONSE> dépend de la présence d'une <UNITÉ DE MESSAGE DE RÉPONSE> consécutive dans le même <MESSAGE DE RÉPONSE>, l'**appareil** n'est pas en mesure de terminer chaque <UNITÉ DE MESSAGE DE RÉPONSE> avant que l'analyseur syntaxique n'ait reçu une autre <UNITÉ DE MESSAGE D'INTERROGATION> ou un autre <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The **device** shall interpret a GET message in the same way as a <PROGRAM MESSAGE> element. The **device** shall report a Command Error if it encounters a GET message between the first byte of a <PROGRAM MESSAGE> and the <PROGRAM MESSAGE TERMINATOR>.

The **controller** should not attempt to read any part of a <RESPONSE MESSAGE> from the **device** until it has sent the entire Query Message to the **device**. The **controller** should not send any part of another <PROGRAM MESSAGE> to the **device** until it has read the entire <RESPONSE MESSAGE> from the **device**. The **device** shall detect the **controller's** failure to follow this protocol and shall act as described under "Protocol exceptions", see 6.5.

After pon or after receiving **dcas**, the **device** shall wait for a <PROGRAM MESSAGE> from the **controller**. During this time, it may assert the IEC 625-1 SRQ (service request) signal and shall interpret and respond to IEC 625-1 interface messages (e.g. Serial Poll) but shall not automatically generate any <RESPONSE MESSAGE> elements.

At any given time, the **device** is doing one of the following things:

1) The **device** is idle, waiting for a message from the **controller**. The **controller** may be idle or may be communicating with some other device(s) on the **system bus**. (The Message Exchange Control is in the IDLE or DONE state.)

2) The **device** is receiving a message from the **controller**. The **controller** may interrupt the transfer to perform a serial poll or to communicate with other devices, but should not attempt to read data bytes from the **device** until it has sent a <PROGRAM MESSAGE TERMINATOR>, see 7.5, to the **device**. (The Message Exchange Control is in the READ, QUERY, or DEADLOCK state.)

3) The **device** is sending a <RESPONSE MESSAGE> to the **controller**. The **controller** may interrupt the transfer to perform a serial poll or to communicate with other devices, but should not attempt to send data bytes to the **device** until it has received a <RESPONSE MESSAGE TERMINATOR>, see 8.5, from the **device**. (The Message Exchange Control is in the SEND or RESPONSE state.)

6.4.4 *Compound queries*

One or more <QUERY MESSAGE UNIT> elements can be placed in a single <PROGRAM MESSAGE>, clause 7. Also, the **device** is allowed to generate one or more <RESPONSE MESSAGE UNIT> elements for each <QUERY MESSAGE UNIT> received. The <RESPONSE MESSAGE UNIT SEPARATOR> (semicolon ";", see 8.4.1) shall be used to separate <RESPONSE MESSAGE UNIT> elements within the complete compound <RESPONSE MESSAGE>. The <RESPONSE MESSAGE TERMINATOR>, defined in 8.5 as NL^END, shall terminate the last <RESPONSE MESSAGE UNIT>. Since the required terminator for a <RESPONSE MESSAGE UNIT> is dependent upon whether a subsequent <RESPONSE MESSAGE UNIT> will be in the same <RESPONSE MESSAGE>, the **device** is not able to terminate each <RESPONSE MESSAGE UNIT> until the Parser receives either another <QUERY MESSAGE UNIT> or <PROGRAM MESSAGE TERMINATOR>.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Les interrogations complexes ne doivent pas couvrir plus d'un <MESSAGE DE PROGRAMMATION> complet. Si une interrogation complexe dépasse la capacité d'un **appareil** à maintenir le <MESSAGE DE RÉPONSE> associé dans la file d'attente de sortie, le formateur de réponse de l'**appareil**, le contrôle d'exécution et l'analyseur syntaxique peuvent devenir bloqués jusqu'à ce qu'il y ait de la place disponible. Si le tampon d'entrée devient plein pendant que l'analyseur syntaxique est interrompu, il se produira un blocage du tampon, voir 6.5.7.4.

NOTE - Si l'un quelconque des éléments d'<UNITÉ DE MESSAGE D'INTERROGATION> d'une interrogation complexe renvoie plus d'une <UNITÉ DE MESSAGE DE RÉPONSE>, un programme d'application ne peut pas utiliser le point-virgule ";", séparateur, pour distinguer la fin de la réponse à chaque <UNITÉ DE MESSAGE D'INTERROGATION>.

6.4.5 *Exigences concernant les instructions de messages*

6.4.5.1 Instruction d'exécution des messages de programmation

L'appareil doit exécuter les éléments de <MESSAGE DE PROGRAMMATION> dans l'ordre reçu. (Voir 6.1.7 pour la définition de l'exécution). L'exécution d'éléments analysables individuels peut être faite au fur et à mesure qu'ils sont reçus. L'**appareil** peut également attendre jusqu'à ce que, soit un <SÉPARATEUR D'UNITÉS DE MESSAGE DE PROGRAMMATION>, soit un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> soit analysé avant d'exécuter les éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION> précédents. L'**appareil** ne doit pas exécuter d'<UNITÉ DE MESSAGE DE PROGRAMMATION> avant d'exécuter tous les éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION> antérieurs, à l'exception de ceux décrits dans le cadre de "Couplages des paramètres de l'appareil", voir 6.4.5.3. L'**appareil** doit exécuter tous les éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION> d'un <MESSAGE DE PROGRAMMATION> donné avant d'exécuter des éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION> d'un <MESSAGE DE PROGRAMMATION> consécutif.

6.4.5.2 Instruction d'exécution de messages GET mélangés et d'éléments de MESSAGE DE PROGRAMMATION

L'**appareil** doit exécuter les messages GET dans le même ordre que les éléments de <MESSAGE DE PROGRAMMATION>. Si l'**appareil** reçoit un message GET pendant l'analyse ou l'exécution d'un message antérieur, l'exécution du message GET doit être retardée jusqu'à ce que les messages antérieurs aient été exécutés.

6.4.5.3 Couplages des paramètres de l'appareil

Les paramètres couplés sont les fonctions de l'appareil qui interagissent l'une avec l'autre d'une façon spécifique à l'appareil. Des problèmes peuvent surgir parce que l'état antérieur de l'**appareil** peut affecter la réponse de l'**appareil** à la programmation d'un paramètre couplé. L'idéal est que l'**appareil** garantisse qu'un <MESSAGE DE PROGRAMMATION> complet contenant un jeu de positionnements corrects pour les paramètres couplés soit accepté indépendamment du positionnement préalable de l'un quelconque des paramètres couplés.

EXEMPLE: Une alimentation électrique peut avoir son courant et sa tension réglés sur une large gamme de valeurs avec la nécessité que le produit des deux paramètres soit inférieur à 100 watts. Supposons que la tension soit réglée sur 100 volts et le courant sur 1 ampère. L'**appareil** devrait permettre au programmeur d'application d'envoyer le message suivant et de ne pas signaler une erreur.

COURANT 100; TENSION 1 NL^END

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Compound queries shall not span more than one complete <PROGRAM MESSAGE>. If a compound query exceeds a **device's** ability to hold the associated <RESPONSE MESSAGE> in the Output Queue, the **device's** Response Formatter, Execution Control, and Parser may become blocked waiting until room is available. If the Input Buffer becomes full while the Parser is suspended, a buffer deadlock will occur, see 6.5.7.4.

NOTE - If any of the <QUERY MESSAGE UNIT> elements of a compound query returns more than one <RESPONSE MESSAGE UNIT>, an application program cannot use the semicolon, ";", delimiter to distinguish the end of the response to each <QUERY MESSAGE UNIT>.

6.4.5 *Message order requirements*

6.4.5.1 Execution order of program messages

The device shall execute <PROGRAM MESSAGE> elements in the order received. (See 6.1.7 for definition of execute.) Execution of individual parsable elements may be done as they are received. The **device** may also wait until either a <PROGRAM MESSAGE UNIT SEPARATOR> or a <PROGRAM MESSAGE TERMINATOR> is parsed before executing preceding <PROGRAM MESSAGE UNIT> elements. The **device** shall not execute any <PROGRAM MESSAGE UNIT> before executing all prior <PROGRAM MESSAGE UNIT> elements, except as described under "Device parameter couplings", see 6.4.5.3. The **device** shall execute all the <PROGRAM MESSAGE UNIT> elements of a given <PROGRAM MESSAGE> before executing any <PROGRAM MESSAGE UNIT> elements of a successive <PROGRAM MESSAGE>.

6.4.5.2 Execution order of intermixed GET messages and <PROGRAM MESSAGE> elements

The **device** shall execute GET messages in the same order as <PROGRAM MESSAGE> elements. If the **device** receives a GET message while parsing or executing a prior message, execution of the GET message shall be delayed until the prior messages have been executed.

6.4.5.3 Device parameter couplings

Coupled parameters are device functions which interact with each other in a device-specific manner. Problems can arise because the prior state of the **device** can affect the **device's** response to the programming of a coupled parameter. Ideally, the **device** shall guarantee that a complete <PROGRAM MESSAGE> containing a set of valid settings for coupled parameters will be accepted independent of the prior setting of any of the coupled parameters.

EXAMPLE: A power supply can have its current and voltage set to a wide range of values with the requirement that the product of the two parameters is less than 100 watts. Assume the voltage is set to 100 volts, and the current to 1 amp. The **device** should allow the application programmer to send the following message and not have an error reported.

CURRENT 100; VOLTAGE 1 NL^END

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Pour accomplir l'indépendance de séquence des paramètres couplés, le contrôle d'exécution peut mettre en mémoire tampon les unités de messages analysées et différer l'exécution jusqu'à ce que **eom** ou une instruction spécifique à l'appareil soit reçue. L'**appareil** peut également utiliser d'autres techniques pour atteindre le même objectif. La nécessité que les éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION> soient exécutés dans l'ordre de leur réception est supprimée dans ces circonstances.

Si les éléments de <MESSAGES DE PROGRAMMATION> de paramètres couplés ne sont pas contigus, les résultats finals sont alors non spécifiés. Dans ce cas, l'**appareil** peut signaler une erreur d'exécution si le critère de couplage de l'appareil n'est pas respecté avant que le jeu complet de paramètres couplés ait été exécuté.

Certains **appareils** peuvent avoir des couplages de paramètres n'entrant pas dans le cadre des exigences du présent article concernant le couplage des paramètres de l'appareil.

6.4.5.4 Génération des données de messages de réponse

Les concepteurs d'appareils peuvent choisir de générer les octets d'un <MESSAGE DE RÉPONSE> au moment où le **contrôleur** lit un message plutôt qu'immédiatement après l'exécution d'un message d'interrogation. L'envoi de grandes quantités de données de réponse peut nécessiter cette technique. Dans de tels cas, le contenu du <MESSAGE DE RÉPONSE> représentera l'état de l'**appareil** au moment où le **contrôleur** lit la réponse, plutôt qu'au moment où l'interrogation a été analysée. La documentation de l'appareil doit signaler les réponses aux interrogations qui seront évaluées au moment où les données seront effectivement lues.

Lorsque le concepteur de l'appareil choisit de différer la génération des données de réponse jusqu'à ce que **brq** soit VRAI, l'**appareil** doit mettre le message MAV VRAI au moment où l'**appareil** est prêt à répondre à la demande du **contrôleur** pour des données. Ainsi, un programme d'application peut toujours compter recevoir le message MAV, par l'intermédiaire du potentiel d'indication d'état, comme un signal pour commencer un transfert de l'appareil au contrôleur, voir 11.5.2.1.

Les éléments de <MESSAGE DE RÉPONSE> doivent être envoyés au contrôleur dans le même ordre selon lequel leurs éléments d'<UNITÉ DE MESSAGE D'INTERROGATION> associés ont été envoyés à l'**appareil**. L'**appareil** ne devra en aucun cas envoyer la réponse à une <UNITÉ DE MESSAGE D'INTERROGATION> avant de répondre à une <UNITÉ DE MESSAGE D'INTERROGATION> antérieure, sauf si la réponse antérieure a été arrêtée prématurément en raison de l'une des exceptions au protocole décrites dans 6.5.

6.5 *Exceptions au Protocole*

6.5.1 *Messages arrêtés prématurément*

Des <MESSAGES DE PROGRAMMATION> et <MESSAGES DE RÉPONSE> doivent être arrêtés prématurément par **dcas** ou pon.

6.5.2 *Adressé pour parler sans rien à dire*

Si le contrôle E/S met **brq** VRAI pendant que le contrôle d'échanges de messages est dans l'état LECTURE ou INACTIF, l'**appareil** a été adressé pour parler sans rien à dire. L'**appareil** doit exécuter, comme indiqué à 6.3.2.2, une action "NON TERMINÉ".

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

To accomplish sequence independence of coupled parameters, the Execution Control may buffer parsed message units and defer execution until **eom** or some device-specific command is received. The **device** may also use other techniques to achieve the same goal. The requirement that <PROGRAM MESSAGE UNIT> elements are executed in order of reception is relaxed under these circumstances.

If the coupled parameter <PROGRAM MESSAGE> elements are not contiguous then the final results are unspecified. In this case the **device** may report an Execution Error if the parameter coupling criterion is violated before the complete set of coupled parameters has been executed.

Some **devices** may have parameter couplings that are beyond the scope of the device parameter coupling requirements of this clause.

6.4.5.4 Generation of Response Message data

Device designers may choose to generate the bytes of a <RESPONSE MESSAGE> at the time the **controller** reads a message rather than immediately following the execution of a Query Message. Sending large amounts of response data may require this technique. In such cases, the contents of the <RESPONSE MESSAGE> will represent the state of the **device** at the time the **controller** reads the response, rather than at the time the query was parsed. Device documentation shall indicate the query responses which will be evaluated at the time data is actually read.

When the device designer chooses to defer the generation of response data until **brq** is TRUE, the **device** shall set the MAV message TRUE at the point when the **device** is ready to respond to the **controller's** request for data. Thus, an application program can always rely on receiving the MAV message, via the status reporting capability, as a signal to begin a device-to-controller transfer, see 11.5.2.1.

<RESPONSE MESSAGE> elements shall be sent to the controller in the same order that their associated <QUERY MESSAGE UNIT> elements were sent to the **device**. Under no condition shall the **device** send the response to a <QUERY MESSAGE UNIT> before responding to a prior <QUERY MESSAGE UNIT>, except when the prior response was aborted due to one of the protocol exceptions described in 6.5.

6.5 *Protocol exceptions*

6.5.1 *Aborted messages*

<PROGRAM MESSAGE> and <RESPONSE MESSAGE> elements shall be aborted by **dcas** or pon.

6.5.2 *Addressed to talk with nothing to say*

If the I/O Control sets **brq** TRUE while the Message Exchange Control is in the READ or IDLE state, the **device** has been addressed to talk with nothing to say. The **device** shall perform as indicated in 6.3.2.2, UNTERMINATED action.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Si le contrôle E/S met **brq** VRAI pendant que le contrôle d'échanges de messages est dans l'état RÉPONSE, mais toutes les interrogations dans le message d'interrogation n'ont généré aucune <DONNÉE DE RÉPONSE> en raison d'erreurs de syntaxe (instruction), sémantique (exécution), ou spécifiques à l'appareil, l'**appareil** a été adressé pour parler sans rien à dire. L'**appareil** doit agir comme indiqué à 6.3.1.5.

NOTE - L'appareil peut être incapable d'envoyer des données lorsqu'il est adressé pour parler même s'il n'existe aucune condition d'erreur. Ceci peut se produire parce que l'appareil a différé la préparation des données de réponse jusqu'à ce qu'une opération interne soit complète ou qu'un "signal de contrôle externe" soit reçu. Ceci n'est pas une condition d'erreur et une erreur d'interrogation ne sera pas signalée. Si le contrôleur continue à attendre les données, l'appareil les enverra lorsqu'elles seront disponibles.

6.5.3 *Pas d'écouteur sur le bus*

Si la fonction d'établissement d'une liaison avec la source CEI 625-1 de l'**appareil** est dans l'état "Retard Source" (SDYS) et les signaux "Pas de données acceptées" (NDAC) et "Pas prêt pour données" (NRFD) sont tous les deux passifs et faux, l'**appareil** ne doit pas envoyer d'octets de données ni entreprendre une action qui affecterait l'activité ultérieure sur le bus CEI 625-1. L'appareil doit attendre qu'un écouteur excite NDAC VRAI ou que le **contrôleur** prenne le contrôle. Le concepteur de l'appareil peut signaler cette condition à l'utilisateur par un moyen autre que l'**interface du système**.

NOTE - Cette règle s'applique uniquement aux **appareils** qui ne sont pas le contrôleur valide. Les **contrôleurs** sont obligés de signaler cette condition à l'application (voir 15.2).

6.5.4 *Erreur d'instruction*

Une erreur d'instruction est générée dans les conditions définies à l'article 6.1.6.1.1. Lorsque le contrôle d'indication d'état reçoit une erreur d'instruction, il doit mettre le bit d'erreur d'instruction dans le registre d'état d'événement standard, voir 11.5.1.1.4.

6.5.5 *Erreur d'exécution*

Une erreur d'exécution se produit dans les conditions définies en 11.5.1.1.5. Lorsque le contrôle d'exécution détecte une erreur d'exécution, le contrôle d'indication d'état doit mettre le bit d'erreur d'exécution dans le registre d'état d'événement standard. L'**appareil** doit continuer à analyser et à exécuter les instructions.

6.5.6 *Erreur spécifique à l'appareil*

Les erreurs spécifiques à l'appareil sont définies en 11.5.1.1.6. Ces erreurs ne doivent avoir aucun effet sur le protocole d'échange de messages défini dans cette section.

6.5.7 *Erreur d'interrogation*

Les erreurs d'interrogation sont signalées en mettant le bit d'erreur d'interrogation dans le registre d'état d'événement standard tel qu'il est défini en 11.5.1.1.7. L'**appareil** doit signaler une erreur d'interrogation lorsque le **contrôleur** ne suit pas le protocole de contrôle d'échange de messages dans l'une quelconque des conditions définies aux 6.3.1.7, 6.3.2.2 et 6.3.2.3.

6.5.7.1 Instruction ou interrogation reçue incomplète

L'**appareil** doit détecter une instruction ou interrogation incomplète lorsque le contrôle E/S met **brq** VRAI et l'analyseur syntaxique a reçu au moins un octet d'un <MESSAGE DE PROGRAMMATION>, mais pas **eom**, et le tampon d'entrée est vide. L'**appareil** doit répondre comme indiqué à 6.3.2.2, NON TERMINÉ.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

If the I/O Control sets **brq** TRUE while the Message Exchange Control is in the RESPONSE state but all of the queries in the Query Message failed to generate any <RESPONSE DATA> because of syntax (Command), semantic (Execution), or device-specific errors, the **device** has been addressed to talk with nothing to say. The **device** shall perform as indicated in 6.3.1.5.

NOTE - The device may be unable to send data when it is addressed to talk even if no error condition exists. This can occur because the device has deferred the preparation of the response data until an internal operation is complete or an "External Control Signal" is received. This is not an error condition and a Query Error shall not be reported. If the controller continues to wait for the data, the device will send it when it is available.

6.5.3 *No listener on bus*

If the **device's** IEC 625-1 Source Handshake function is in the Source Delay State (SDYS) and the Not Data Accepted (NDAC) and Not Ready For Data (NRFD) signals are both passive false, the **device** shall not send any data bytes or take any action that would affect subsequent activity on the IEC 625-1 bus. The **device** shall wait for a listener to assert NDAC TRUE or for the **controller** to take control. The device designer may report this condition to the user by means other than the **system interface**.

NOTE - This rule applies only to **devices** which are not the controller-in-charge. **Controllers** are required to report this condition to the application (see 15.2).

6.5.4 *Command Error*

A Command Error is generated under the conditions defined in 6.1.6.1.1. When the Status Reporting Control receives a Command Error, it shall set the Command Error bit in the Standard Event Status Register, see 11.5.1.1.4.

6.5.5 *Execution Error*

An Execution Error occurs under the conditions defined in 11.5.1.1.5. When the Execution Control detects an Execution Error, the Status Reporting Control shall set the Execution Error bit in the Standard Event Status Register. The **device** shall continue parsing and executing commands.

6.5.6 *Device-specific Error*

Device-specific Errors are defined in 11.5.1.1.6. These errors shall have no effect on the message exchange protocol defined in this section.

6.5.7 *Query Error*

Query errors are reported by setting the Query Error bit in the Standard Event Status Register as defined in 11.5.1.1.7. The **device** shall report a Query Error when the **controller** fails to follow the Message Exchange Control Protocol under any of the conditions defined in 6.3.1.7, 6.3.2.2, and 6.3.2.3.

6.5.7.1 Incomplete command or query received

The **device** shall detect an incomplete command or query when the I/O Control sets **brq** TRUE and the Parser has received at least one byte of a <PROGRAM MESSAGE>, but not **eom**, and the Input Buffer is empty. The **device** shall respond as indicated in 6.3.2.2, UNTERMINATED.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

6.5.7.2 Réponse interrompue

Si l'**appareil** reçoit un octet de données ou un message GET suivant un message d'interrogation mais avant que l'**appareil** ait envoyé le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>, l'action "INTERROMPU" doit être exécutée. Lorsque ceci se produit, l'**appareil** doit arrêter prématurément le reste de la réponse et répondre comme indiqué à 6.3.2.3, INTERROMPU.

6.5.7.3 Unités de messages d'interrogation séparées par le caractère d'arrêt de message de programmation

Lorsqu'un programme d'application envoie un message d'interrogation à un appareil, terminé correctement par un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>, et envoie un autre <MESSAGE DE PROGRAMMATION> sans lire d'abord le <MESSAGE DE RÉPONSE> en entier, l'appareil doit répondre comme indiqué à 6.3.2.3, action "INTERROMPU".

Attention: Si le programme d'application envoie deux messages d'interrogation sans lire la réponse du premier, puis tente de lire la réponse au second, il peut recevoir certains octets de données de la première réponse suivis par la deuxième réponse complète. Le résultat peut être ou non un <MESSAGE DE RÉPONSE> syntaxiquement correct. Ceci se produit lorsque la synchronisation du système permet au **contrôleur** d'envoyer le deuxième message d'interrogation et de commencer son opération de lecture avant que l'analyseur syntaxique détecte la fin du premier message d'interrogation. Dans ce cas, le contrôle d'échanges de messages entre dans l'état ENVOI et commence à envoyer la première réponse au **contrôleur**. Lorsque l'analyseur syntaxique arrive à la fin du premier message d'interrogation, l'**appareil** est INTERROMPU et commence à traiter la deuxième interrogation.

Afin d'éviter la condition ci-dessus, il est recommandé que les programmes d'application n'envoient pas de messages d'interrogation à un **appareil** sans lire le <MESSAGE DE RÉPONSE> associé. Lorsque ceci ne peut pas être évité comme dans un traitement d'exception, il est recommandé de procéder à un effacement d'appareil avant d'envoyer un nouveau message d'interrogation. Ceci assurera que toute réponse fragmentaire, d'un <MESSAGE DE RÉPONSE> antérieur, ne sera pas envoyée par l'appareil.

6.5.7.4 Blocage du tampon

Dans certaines circonstances, un **appareil** peut ne pas être en mesure de générer complètement un <MESSAGE DE RÉPONSE>. L'**appareil** est débloqué lorsqu'il ne peut pas accepter un autre octet de données d'un <MESSAGE DE PROGRAMMATION> du contrôleur parce que son tampon d'entrée et sa file d'attente de sortie sont tous les deux pleins et le **contrôleur** ne peut pas lire le <MESSAGE DE RÉPONSE> de l'**appareil** avant d'avoir fini d'envoyer son message d'interrogation.

L'**appareil** doit répondre à un blocage du tampon comme indiqué à 6.3.1.7, BLOCAGE.

IECNORM.COM : Click to view the full PDF of IEC 625-2:1993

6.5.7.2 Interrupted Response

If the **device** receives a data byte or a GET message following a Query Message but before the **device** has sent the <RESPONSE MESSAGE TERMINATOR>, the INTERRUPTED action shall be performed. When this happens, the **device** shall abort the rest of the response and respond as indicated in 6.3.2.3, INTERRUPTED.

6.5.7.3 Query Message Units separated by the Program Message Terminator

When an application program sends a device a Query Message, properly terminated with a <PROGRAM MESSAGE TERMINATOR>, and sends another <PROGRAM MESSAGE> without first reading the entire <RESPONSE MESSAGE>, the device shall respond as indicated in 6.3.2.3, INTERRUPTED action.

Warning: If the application program sends two Query Messages without reading the response from the first, and then attempts to read the response to the second, it may receive some data bytes from the first response followed by the complete second response. The result may or may not be a syntactically correct <RESPONSE MESSAGE>. This happens when the system timing allows the **controller** to send the second Query Message and begin its read operation before the Parser detects the end of the first Query Message. In this case, the Message Exchange Control enters the SEND state and starts to send the first response to the **controller**. When the Parser reaches the end of the first Query Message, the **device** is INTERRUPTED and starts to process the second query.

To avoid the above condition, application programs should not send Query Messages to a **device** without reading the associated <RESPONSE MESSAGE>. When this cannot be avoided as in exception handling, a device clear should be used before a new Query Message is sent. This will assure that any fragmentary response, from a prior <RESPONSE MESSAGE>, will not be sent by the device.

6.5.7.4 Buffer Deadlock

Under some circumstances a **device** may be unable to completely generate a <RESPONSE MESSAGE>. The **device** is deadlocked when it cannot accept another data byte of a <PROGRAM MESSAGE> from the controller because its Input Buffer and Output Queue are both full and the **controller** cannot read the **device's** <RESPONSE MESSAGE> until it has completed sending its Query Message.

The **device** shall respond to a Buffer Deadlock as indicated in 6.3.1.7, DEADLOCK.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.5.7.5 Interrogation après réponse indéfinie

Certaines interrogations complexes peuvent chercher à générer un <MESSAGE DE RÉPONSE> illégal. Si une <UNITÉ DE MESSAGE D'INTERROGATION> qui génère une réponse indéfinie (soit une DONNÉE DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE, soit une DONNÉE DE RÉPONSE ARBITRAIRE ACSII) n'est pas la dernière <UNITÉ DE MESSAGE D'INTERROGATION> dans un <MESSAGE DE PROGRAMMATION>, l'**appareil** doit signaler une erreur d'interrogation et ne doit pas envoyer la réponse des éléments suivants d'<UNITÉ DE MESSAGE D'INTERROGATION>. L'**appareil** peut en option soit formater, soit ne pas formater la donnée de réponse qui est supprimée. Une fois que l'**appareil** a reconnu le **eom** et a envoyé le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>, le protocole d'échange des messages est rétabli, voir 6.3.

7 Formats de l'appareil écouteur

7.1 *Vue d'ensemble*

Cet article traite du formatage des messages de programmation reçus par un **appareil** en provenance de son **système d'interface**. Ce formatage se produit sur deux niveaux. Le plus élevé est le niveau "fonctionnel" où chaque élément a une importance fonctionnelle. Ce niveau fonctionnel est nécessaire pour les concepteurs du jeu d'instructions de l'appareil. Le niveau le moins élevé de formatage représente le "codage" réel par bus nécessaire pour transmettre un élément fonctionnel. Cette dernière information est nécessaire pour les concepteurs des analyseurs syntaxiques de l'appareil.

7.1.1 *Génération du jeu d'instructions de l'appareil*

Les messages de programmation CEI 625-2 admissibles sont composés de séquences d'unités de messages de programmation, chaque unité représentant une instruction ou une interrogation de programmation. Chaque instruction ou interrogation de programmation est composée d'une séquence d'éléments syntaxiques fonctionnels. Les instructions et interrogations de programmation autorisées CEI 625-2 sont créées à partir de séquences d'éléments fonctionnels générées en traversant le diagramme de syntaxe fonctionnelle illustré aux figures 7-1 à 7-6.

Quelques instructions et interrogations d'application du système d'instruments universel ont été définies par cette norme. Ce sont les instructions communes décrites à la section 10. Ces instructions et interrogations communes suivent des branches spécifiques de sélection à travers le diagramme de syntaxe fonctionnelle. Les instructions restantes sont spécifiques à l'appareil et sont générées par le concepteur de celui-ci, utilisant le diagramme de syntaxe fonctionnelle et les besoins de l'**appareil**. Les éléments fonctionnels comprennent les caractères de séparation, les caractères d'arrêt, les en-têtes et les types de données. Chaque type d'élément est traité en détail dans 7.4 à 7.7.

Les structures des instructions de programmation autorisées CEI 625-2 au niveau fonctionnel devront être générées seulement en traversant le diagramme de syntaxe fonctionnelle.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

6.5.7.5 Query after indefinite response

Certain compound queries may attempt to generate an illegal <RESPONSE MESSAGE>. If a <QUERY MESSAGE UNIT> which generates an indefinite response (either an INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA or ARBITRARY ISO 7-bit RESPONSE DATA) is not the last <QUERY MESSAGE UNIT> in a <PROGRAM MESSAGE>, the **device** shall report a Query Error and not send the response from the following <QUERY MESSAGE UNIT> elements. The **device** may optionally either format or not format the response data which is discarded. Once the **device** recognizes the **eom** and sends the <RESPONSE MESSAGE TERMINATOR>, the message exchange protocol is re-established, see 6.3.

7 Device listening formats

7.1 *Overview*

This clause discusses the formatting of programming messages received by a **device** from its **system interface**. This formatting occurs on two levels. The higher is the "functional" level where each element has functional importance. This functional level is required for designers of the device command set. The lower-level formatting represents the actual bus "encoding" required to transmit a functional element. This latter information is necessary for designers of device parsers.

7.1.1 *Device command set generation*

Allowable IEC 625-2 program messages are composed of sequences of program message units, each unit representing a program command or query. Each program command or query is composed of a sequence of functional syntactic elements. Legal IEC 625-2 program commands and queries are created from functional element sequences generated by traversing the functional syntax diagram illustrated in figures 7-1 to 7-6.

Some commands and queries of universal instrument system application have been defined by this standard. They are the common commands described in clause 10. These common commands and queries are specific path selections through the functional syntax diagram. The remaining commands are device-specific and are generated by the device designer using the functional syntax diagram and the needs of the **device**. The functional elements include separators, terminators, headers, and data types. Each element type is discussed in detail in 7.4 to 7.7.

Legal IEC 625-2 program command structures at the functional level shall only be generated by traversing the functional syntax diagram.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.1.2 *Syntaxe de codage*

7.1.2.1 Syntaxe admissible

Une instruction ou une interrogation de programmation, définie par l'appareil ou commune, est envoyée à l'**appareil** par l'**interface du système** sous la forme d'une séquence d'octets de données. Les séquences d'octets admissibles pour chaque élément fonctionnel sont définies dans le diagramme de syntaxe de codage accompagnant chaque élément. Des branches alternées, mais sémantiquement équivalentes, sont représentées sur chacun de ces diagrammes de syntaxe de codage. Les "champs de données" sont définissables par l'appareil à l'intérieur de limites de codage spécifiées. Les séquences admissibles pour chaque élément fonctionnel sont générées en traversant le diagramme de syntaxe de codage respectif.

Pour les séquences d'éléments fonctionnels admissibles, un **appareil** devra accepter et interpréter correctement les éléments fonctionnels qui suivent la séquence de codage. Toutes les branches équivalentes, comme elles auront été définies pour chaque élément fonctionnel, devront être acceptées par l'analyseur syntaxique de l'**appareil**.

7.1.2.2 Syntaxe illégale – Erreur d'instruction

Une séquence d'octets qui ne suit pas les règles de la syntaxe de codage comme cela est défini en 7.1.2.1 ne devra pas être interprétée comme un élément fonctionnel, mais générera une erreur d'instruction.

Une séquence d'éléments fonctionnels qui n'est pas un message de programmation CEI 625-2 admissible comme défini en 7.1.1 ne doit pas être interprétée comme une instruction ou une interrogation de programmation, mais aboutira à la génération d'une erreur d'instruction.

La manière dont les erreurs d'instruction sont indiquées est décrite à l'article 11: Indications sur l'état de l'appareil. L'interaction de l'analyseur syntaxique avec les erreurs d'instruction et le redressement des erreurs d'instruction sont décrits à l'article 6, protocoles d'échange de messages.

7.1.2.3 Caractère rationnel de la flexibilité de la syntaxe de codage

Pour les données de programmation, les diagrammes de syntaxe de codage sont conçus pour autoriser une variété de codages sémantiquement équivalents d'un élément fonctionnel.

Cette flexibilité a été mise en place pour:

1) Possibilité de lecture humaine. Les instructions de programmation apparaissent de manière typique en code de programmation et de cette façon nécessitent un dégré raisonnable de lisibilité.

2) Génération humaine. Les instructions de programmation sont générées de manière typique par le programmeur et non le contrôleur. De ce fait, un degré de flexibilité de formatage est désirable pour permettre les variations de formatage communément utilisées.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.1.2 *Encoding syntax*

7.1.2.1 Allowable syntax

A device-defined or common program command or query is sent to the **device** over the **system interface** as a sequence of data bytes. The allowable byte sequences for each functional element are defined in the encoding syntax diagram accompanying each element. Alternate, but semantically equivalent paths are shown in each of these encoding syntax diagrams. "Data Fields" are device-definable within specified encoding limits. Allowable sequences for each functional element are generated by traversing the respective encoding syntax diagram.

For allowable functional element sequences, a **device** shall accept and correctly interpret functional elements which follow the encoding sequence. All equivalent paths, as defined for each functional element, shall be accepted by the **device's** parser.

7.1.2.2 Illegal syntax – Command Error

A byte sequence which does not follow encoding syntax rules as defined in 7.1.2.1 shall not be interpreted as a functional element, but shall generate a Command Error.

A functional element sequence that is not an allowable IEC 625-2 program message as defined in 7.1.1 shall not be interpreted as a program command or query but shall result in the generation of a Command Error.

How Command Errors are reported is described in clause 11, Device Status Reporting. Parser interaction with, and recovery from, Command Errors are described in clause 6, Device Message Exchange Protocols.

7.1.2.3 Rationale for encoding syntax flexibility

For program data, the encoding syntax diagrams are designed to allow a variety of semantically equivalent encodings of a functional element.

This flexibility is included for:

1) Human-readability. Program commands typically appear in program code and thus need a reasonable degree of readability.

2) Human-generation. Program commands are typically generated by the programmer and not the controller. Thus, a degree of formatting flexibility is desirable to allow commonly used formatting variations.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.2 *Notation*

7.2.1 *Présentation sous forme de diagramme de la circulation syntaxique*

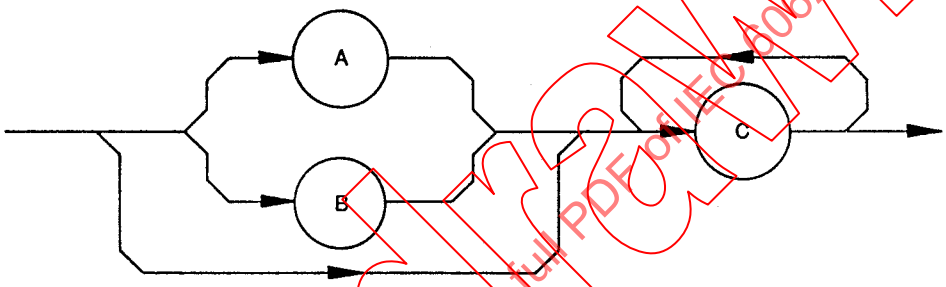
Les éléments syntaxiques sont reliés par des lignes avec des symboles directionnels pour indiquer la circulation. La circulation à travers les diagrammes de syntaxe va généralement de gauche à droite.

Quand un ou plusieurs éléments peuvent être répétés, un chemin inverse, de droite à gauche, sera représenté autour et au-dessus du (ou des) élément(s).

Quand un ou plusieurs éléments peuvent être inutilisés, un chemin de gauche à droite sera représenté autour et au-dessous du (ou des) élément(s).

Le chemin se divise en plusieurs branches quand il y a un choix d'éléments.

EXEMPLE:



Permet les combinaisons suivantes des éléments A, B et C.

AC
C
BCCCCC
CCC

NOTE - L'élément C peut être répété indéfiniment.

7.2.2 *Eléments syntaxiques*

7.2.2.1 Composants élémentaires syntaxiques

Les composants élémentaires sont les constructions syntaxiques de base, indivisibles. Ils représentent soit une fonction de base dans les diagrammes de syntaxe fonctionnelle, soit un codage particulier par octets dans les diagrammes de syntaxe de codage.

Dans les diagrammes de syntaxe fonctionnelle, tous les éléments sont représentés en lettres majuscules, entre crochets, à l'intérieur d'un ovale.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.2 *Notation*

7.2.1 *Diagraming syntactic flow*

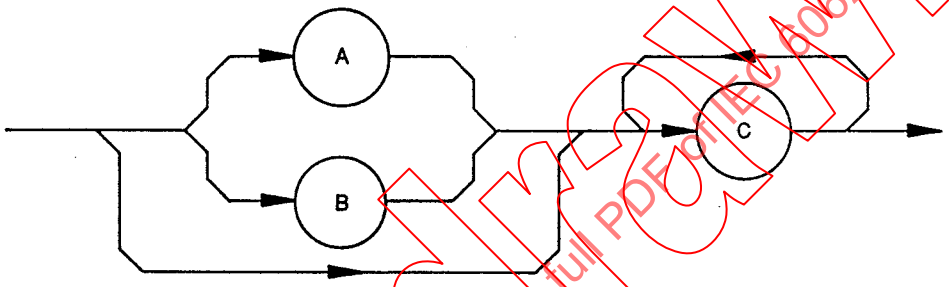
Syntactic elements are connected by lines with directional symbols to indicate the flow. Flow through the syntax diagrams generally proceeds from left to right.

When an element or elements is repeatable, a reverse, right-to-left path will be shown around and above the element(s).

When an element or elements can be bypassed, a left-to-right path will be shown around and below the element(s).

The path branches when there is a choice of elements.

EXAMPLE:



Allows the following combinations of elements A, B and C.

AC
C
BCCCCC
CCC

NOTE - The element C may be repeated indefinitely.

7.2.2 *Syntactic elements*

7.2.2.1 Terminal syntactic elements

Terminal elements are the basic, indivisible syntactic constructs. They represent either a basic function in the functional syntax diagrams or a particular byte-encoding in the encoding syntax diagrams.

All elements in the functional syntax diagrams are represented by an all upper-case description between brackets inside an oval.

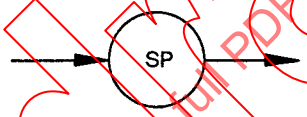
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

EXEMPLE:



Dans les diagrammes de syntaxe de codage, les composants élémentaires sont directement définis comme des octets de données ou par référence au tableau de code de 7 bits ISO représenté à la figure 9-2. Lorsqu'un **appareil** reçoit des caractères codés de 7 bits ISO, la lecture du huitième bit (DIO8) ne sera pas prise en compte.

EXEMPLE: Un espace est représenté comme suit:

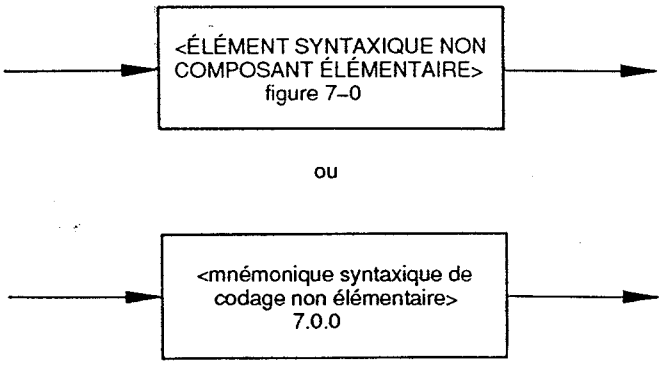


IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.2.2.2 Autres éléments syntaxiques

Les éléments qui ne sont pas des composants élémentaires sont des constructions syntaxiques intermédiaires présentées pour la clarté ou pour insister. Ces éléments sont présentés entre des barres verticales ou dans des cadres pour les distinguer des composants élémentaires. Cet élément est toujours ramené à un diagramme de composants élémentaires ou (pour la syntaxe de codage) est explicable dans un énoncé rapportant l'élément à un code particulier. En-dessous du nom de l'élément se trouve la référence de l'article de l'emplacement du diagramme d'expansion.

EXEMPLE:

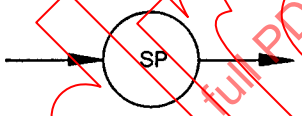


EXAMPLE:



In the encoding syntax diagrams, terminal elements are directly defined as 8-bit data bytes or by reference to the ISO 7-bit Code Chart shown in figure 9-2. When a device receives ISO 7-bit characters, the sense of the eighth bit (DIO8) shall be ignored.

EXAMPLE: A space is represented by the following:

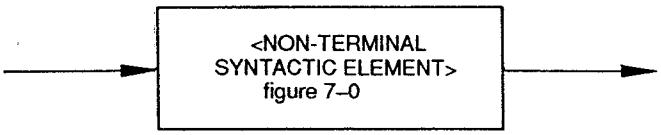


Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

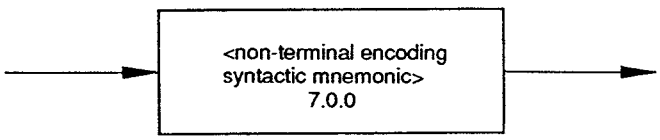
7.2.2.2 Non-terminal syntactic elements

Non-terminal elements are intermediate syntactic constructs presented for clarity or emphasis. Non-terminals are presented in boxes to distinguish them from terminals. A non-terminal is always expandable to a diagram of terminal elements or (for encoding syntax) is explainable in a statement relating the element to a particular code set. Below the element name is a clause reference to the expansion diagram's location.

EXAMPLE:



or



Dans la notation, les éléments fonctionnels sont écrits en majuscules et les éléments de codage sont écrits en minuscules entre crochets à moins qu'un code de 7 bits ISO particulier en soit indiqué.

7.2.2.3 Règles

Plusieurs diagrammes de syntaxe de codage nécessitent un texte supplémentaire pour expliquer les restrictions qui ne peuvent pas facilement être représentées dans le diagramme. Ces restrictions sont incluses dans une section de "Règles" suivant immédiatement le diagramme.

7.2.3 *Symboles spéciaux*

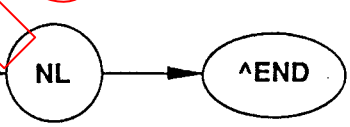
7.2.3.1 *Equivalence entre majuscules et minuscules*

Pour améliorer la présentation, une notation spéciale est utilisée pour indiquer la représentation alternativement en majuscules ou en minuscules. Cette représentation est indiquée en mettant dans un cercle le symbole alpha ISO en majuscule, une barre verticale, et le symbole alpha ISO en minuscule.

EXEMPLE



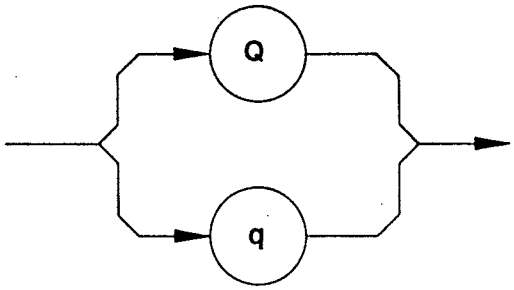
est équivalent à



7.2.3.2 Message de FIN

Un ovale contenant "^END" indique la transmission simultanée du message de FIN avec l'octet de donnée précédent. Ce symbole n'a pas de chemin de sortie car il représente toujours la fin d'un message.

EXEMPLE:



représente l'envoi de NL (interligne) avec EOI VRAI et ATN FAUX. Dans le texte, ceci est représenté par NL^END.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

In the notation, functional elements are written in upper case and encoding elements are written in lower case between brackets unless a particular ISO 7-bit code is indicated.

7.2.2.3 Rules

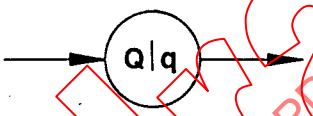
Several encoding syntax diagrams require additional text to explain restrictions not easily shown in the diagram. These restrictions are included in a "Rules" section immediately following the diagram.

7.2.3 *Special symbols*

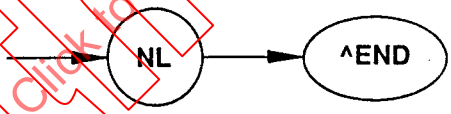
7.2.3.1 Upper/lower case equivalence

To improve presentation, a special notation is used to indicate either upper-or lower-case alternative representation. This representation is indicated by enclosing in a circle the upper-case ISO 7-bit alpha symbol, a vertical bar, and the lower-case ISO 7-bit alpha symbol.

EXAMPLE



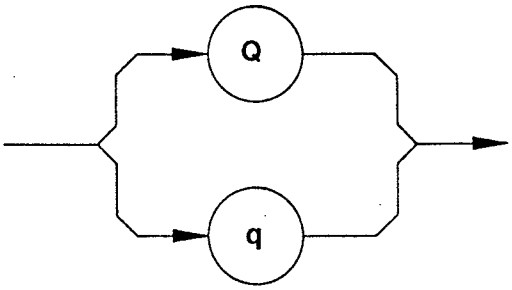
is equivalent to



7.2.3.2 END message

An oval containing "^END" indicates concurrent transmission of the END message with the preceding data byte. This symbol has no exit path as it always represents the end of a message.

EXAMPLE:



represents sending NL (newline) with EOI set TRUE and ATN FALSE. In text this is shown as NL^END.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.2.3.3 Commencement d'un message

Le commencement d'un message est indiqué par un cercle sans entrée sur la gauche.

7.2.3.4 Expansion de diagramme

Des flèches à gauche et à droite indiquent que le diagramme représente une expansion en vue détaillée d'une partie d'un diagramme de plus haut niveau.

7.3 *Messages de programmation terminés – Syntaxe fonctionnelle*

7.3.1 *Fonction*

Les messages de programmation terminés sont des messages complets de "contrôleur à appareil". Ce sont des séquences de zéro ou plusieurs éléments d'<UNITÉ DE MESSAGES DE PROGRAMMATION>. L'élément d'<UNITÉ DE MESSAGE DE PROGRAMMATION> représente une instruction ou une donnée de programmation envoyée à l'appareil depuis le contrôleur. Le message terminé représente une transmission "complète" et, en tant que telle, possède certains sens sémantiques supplémentaires, voir 7.5.

7.3.2 *Syntaxe*

La syntaxe fonctionnelle d'instruction devra correspondre à l'ensemble des six diagrammes suivants dans les figures 7-1 à 7-6.

NOTE - L'utilisation de la forme indéfinie de l'élément de <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE>, voir 7.7.6, impose un arrêt immédiat du <MESSAGE DE PROGRAMMATION>.

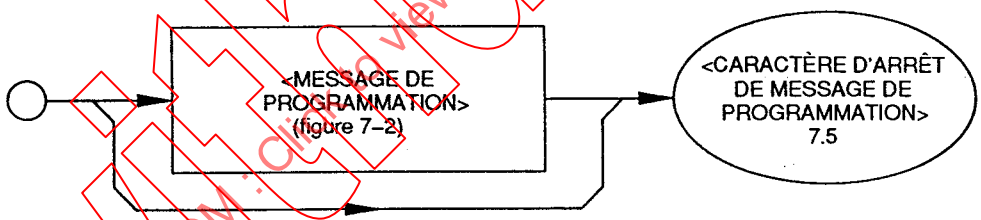


Figure 7-1 – Syntaxe d'élément fonctionnel de <MESSAGE DE PROGRAMMATION TERMINÉ>

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.2.3.3 Beginning of message

The beginning of a message is indicated by a circle with no entry from the left.

7.2.3.4 Diagram expansion

Leading and trailing arrows indicate that the diagram represents an expansion to a detailed view of a portion of a higher level diagram.

7.3 *Terminated program messages – Functional syntax*

7.3.1 *Function*

Terminated program messages are complete 'controller-to-device' messages. They are sequences of zero or more <PROGRAM MESSAGE UNIT> elements. The <PROGRAM MESSAGE UNIT> element represents a programming command or data sent to the device from the controller. The terminated message represents a "complete" transmission and, as such, has certain additional semantic meanings, see 7.5.

7.3.2 *Syntax*

The command's functional syntax shall match the following set of six diagrams in figures 7-1 to 7-6.

NOTE - The use of the indefinite form of the <ARBITRARY BLOCK PROGRAM DATA> element, see 7.7.6, forces an immediate termination of the <PROGRAM MESSAGE>.

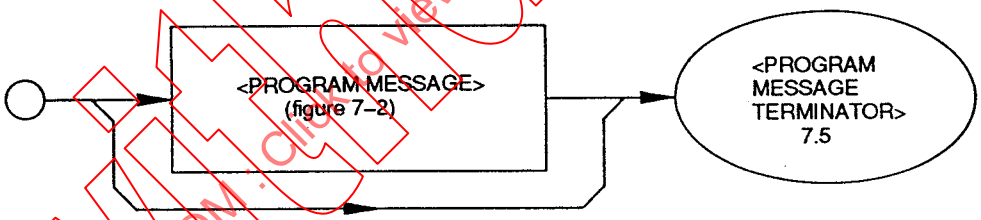


Figure 7-1 – <TERMINATED PROGRAM MESSAGE> functional element syntax

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

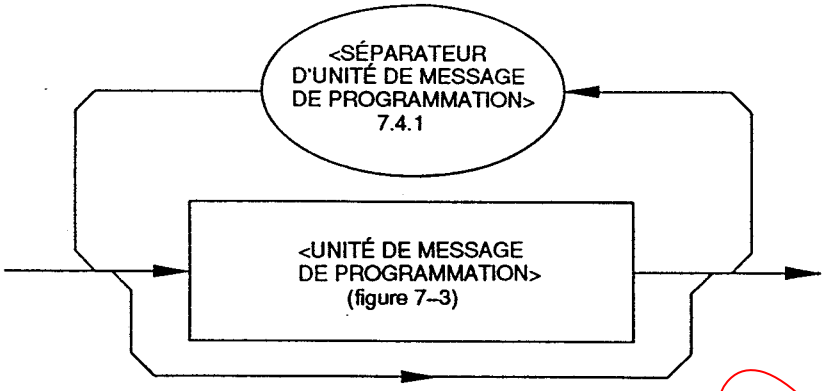


Figure 7-2 – Syntaxe d'élément fonctionnel de <MESSAGE DE PROGRAMMATION>

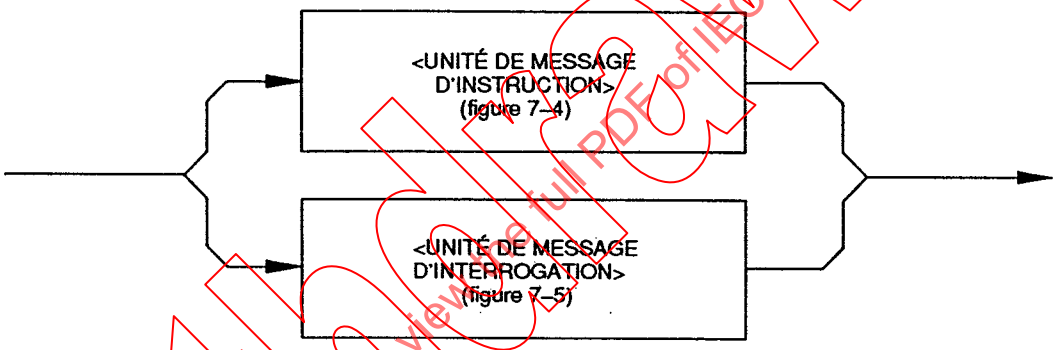


Figure 7-3 – Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE DE PROGRAMMATION>

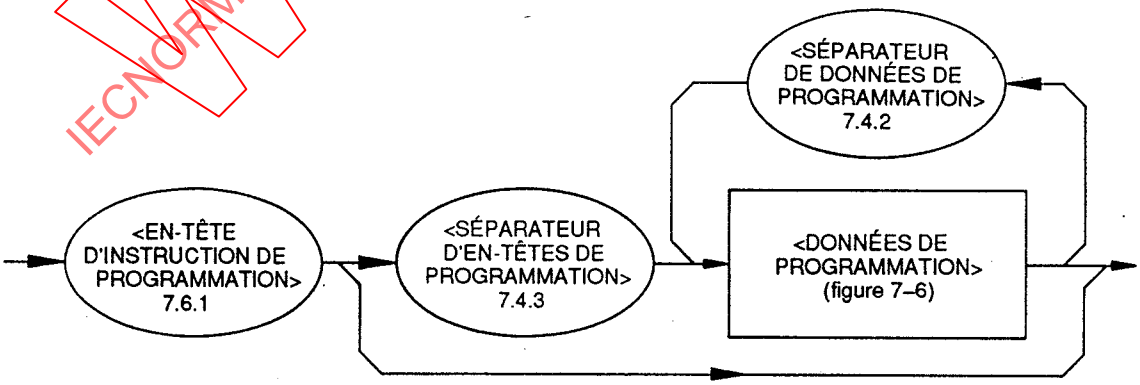


Figure 7-4 – Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE D'INSTRUCTION>

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

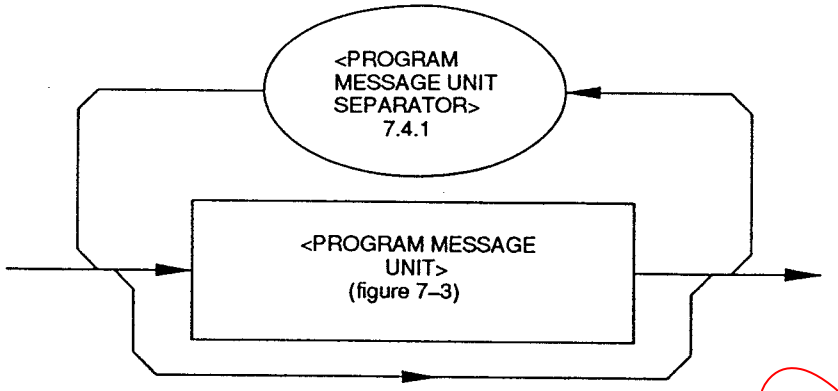


Figure 7-2 – <PROGRAM MESSAGE> functional element syntax

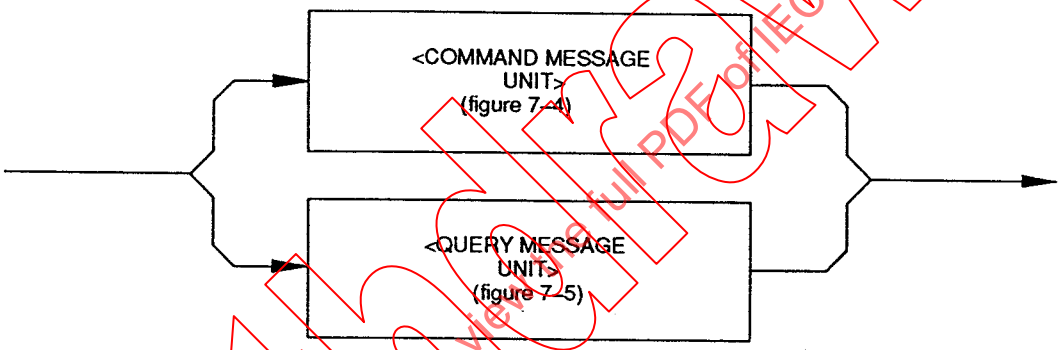


Figure 7-3 – <PROGRAM MESSAGE UNIT> functional element syntax

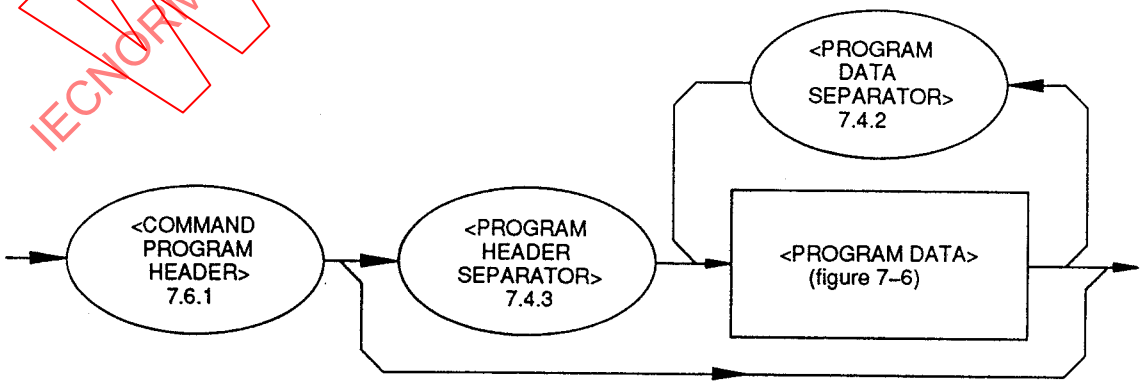


Figure 7-4 – <COMMAND MESSAGE UNIT> functional element syntax

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

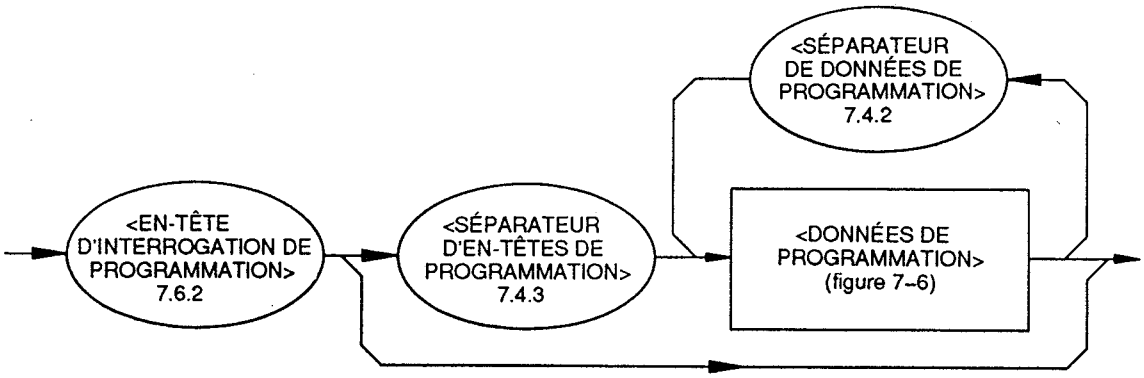


Figure 7-5 – Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE D'INTERROGATION>

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

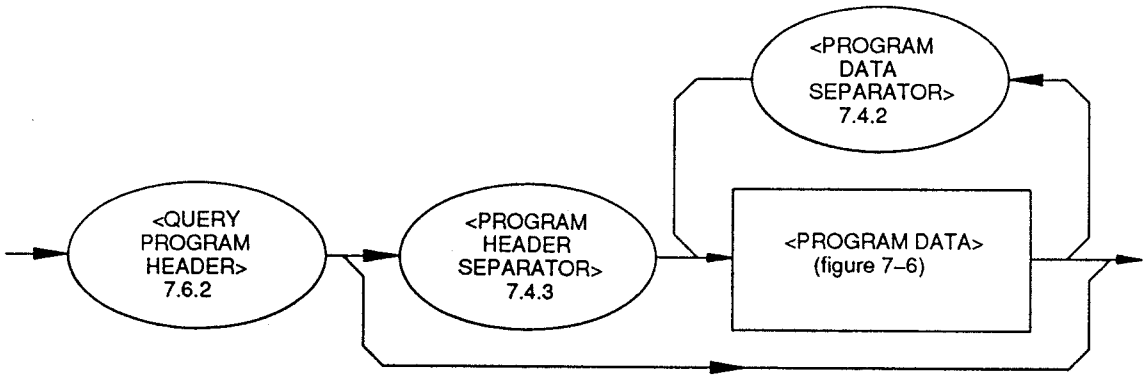
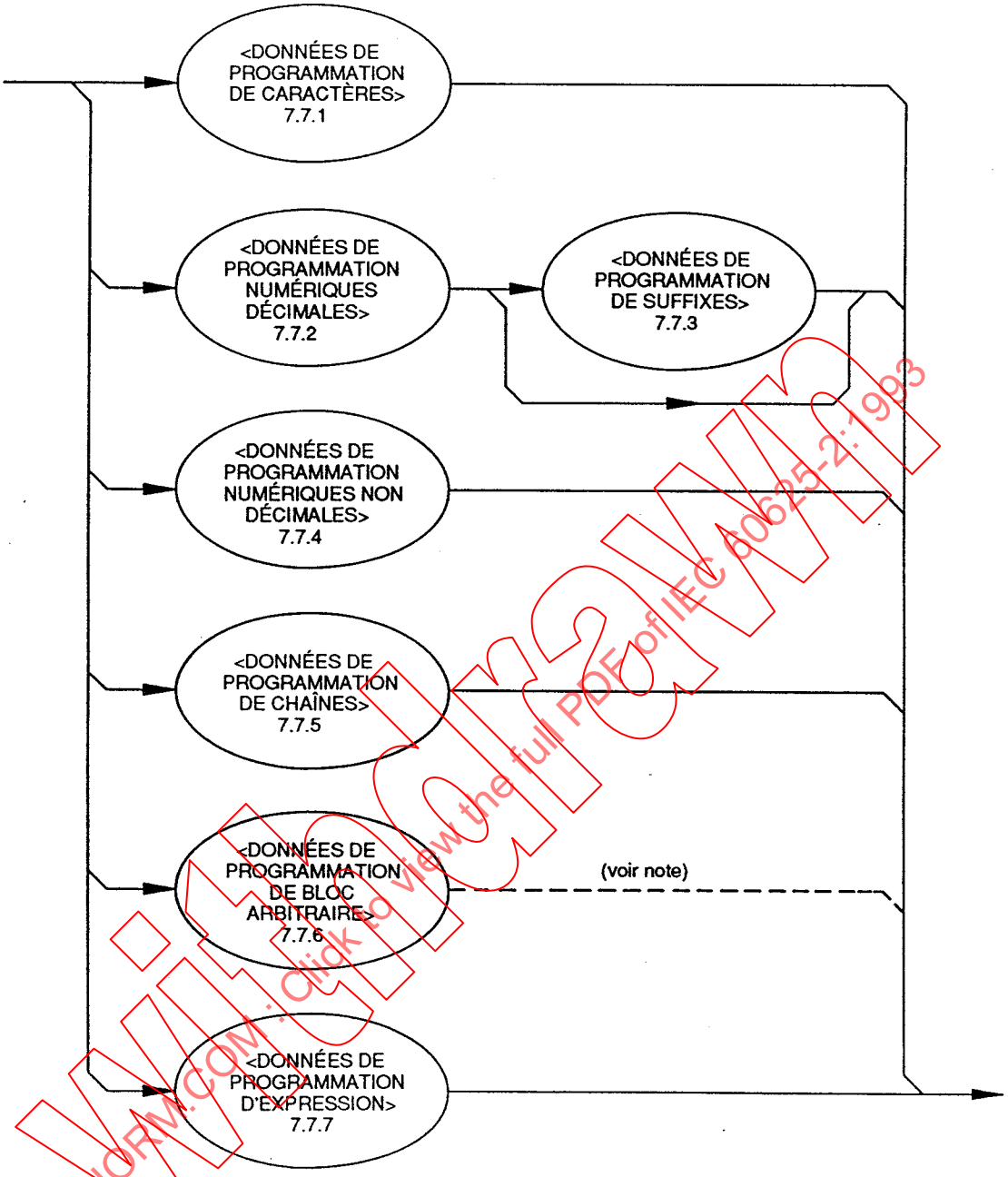


Figure 7-5 – <QUERY MESSAGE UNIT> functional element syntax

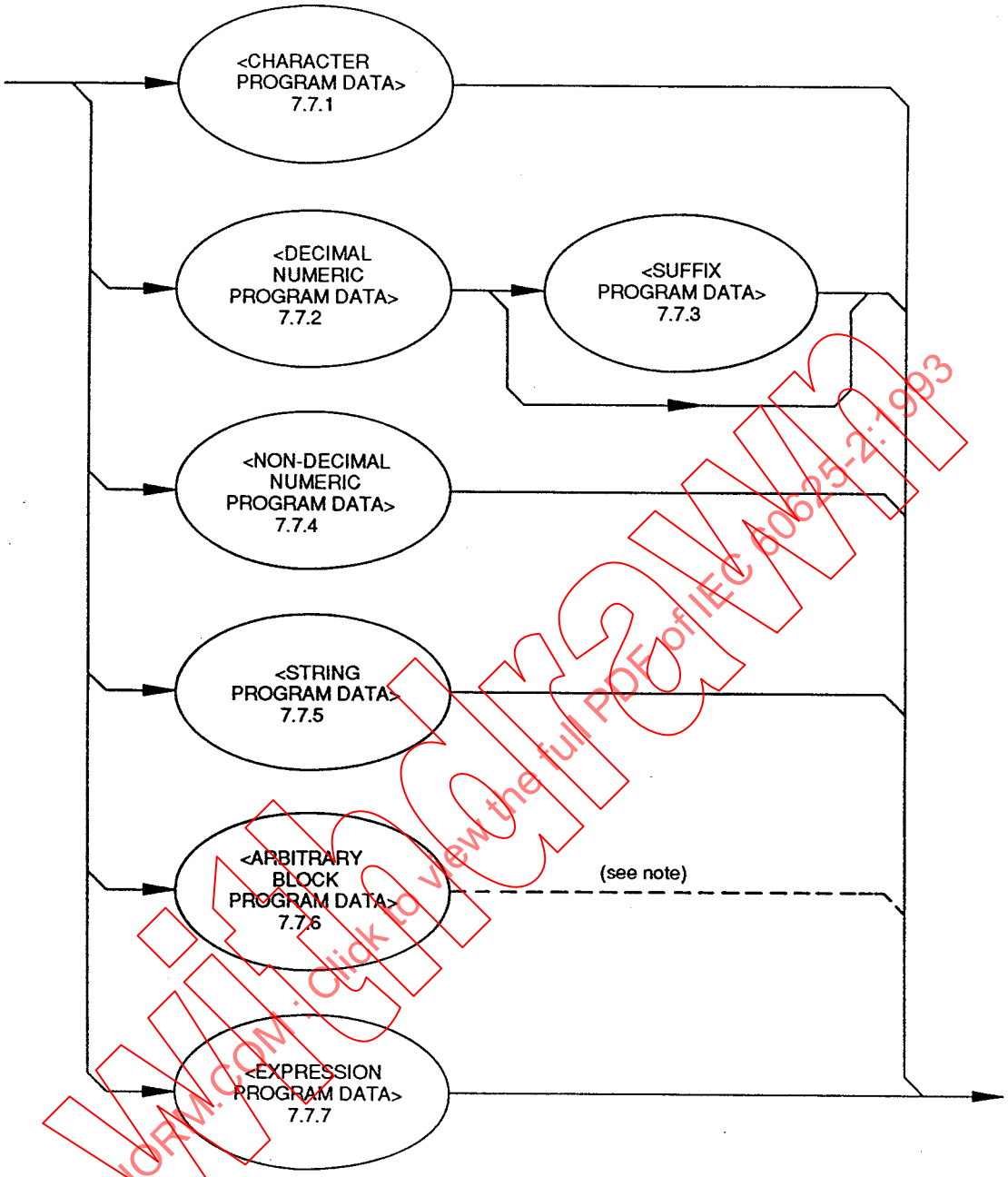
Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

NOTE - Les <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> utilisant le format indéfini se terminent avec un caractère d'arrêt de message implicite. Dans ce cas, le message se termine sans sortie. Voir 7.7.6.

Figure 7-6 – Syntaxe d'élément fonctionnel de <DONNÉES DE PROGRAMMATION>



NOTE - <ARBITRARY BLOCK PROGRAM DATA> using the indefinite format ends with an implicit message terminator. In this case the message terminates with no exit. See 7.7.6.

Figure 7-6 – <PROGRAM DATA> functional element syntax

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.3.3 *Récapitulatif des éléments fonctionnels*

Elément	*Fonction*
<MESSAGE DE PROGRAMMATION> figure 7-2	Représente une séquence de zéro ou plusieurs éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> séparés par des éléments <SÉPARATEURS D'UNITÉS DE MESSAGES DE PROGRAMMATION>.
<UNITÉ DE MESSAGE DE PROGRAMMATION> figure 7-3	Représente une instruction simple, donnée de programmation, ou interrogation reçue par l'**appareil**.
<UNITÉ DE MESSAGE D'INSTRUCTION> figure 7-4	Représente une instruction simple ou donnée de programmation reçue par l'**appareil**.
<UNITÉ DE MESSAGE D'INTERROGATION> figure 7-5	Représente une interrogation simple envoyée du contrôleur à l'**appareil**.
<DONNÉES DE PROGRAMMATION> figure 7-6	Représente n'importe lequel des six différents types de données de programmation.
<SÉPARATEUR D'UNITÉS DE MESSAGES DE PROGRAMMATION> 7.4.1	Sépare les éléments d'<UNITÉ DE MESSAGES DE PROGRAMMATION> les uns des autres dans un <MESSAGE DE PROGRAMMATION>
<SÉPARATEUR DE DONNÉES DE PROGRAMMATION> 7.4.2	Sépare les éléments séquentiels de <DONNÉES DE PROGRAMMATION> relatifs au même en-tête.
<SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> 7.4.3	Sépare l'en-tête de n'importe quelle <DONNÉE DE PROGRAMMATION ASSOCIÉE>.
<CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> 7.5	Termine un <MESSAGE DE PROGRAMMATION>.
<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> 7.6.1	Spécifie la fonction ou l'opération. Utilisé avec n'importe quel(s) élément(s) de <DONNÉE DE PROGRAMMATION> associé(s).
<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> 7.6.2	Semblable à l'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> excepté un indicateur d'interrogation (?) montrant qu'une réponse est attendue de la part de l'appareil.
<CHARACTER PROGRAM DATA> 7.7.1	Type de donnée convenant pour envoyer de courtes données mnémoniques, généralement quand une donnée de type numérique ne convient pas.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.3.3 *Functional element summary*

Element	*Function*
<PROGRAM MESSAGE> figure 7-2	Represents a sequence of zero or more <PROGRAM MESSAGE UNIT> elements separated by <PROGRAM MESSAGE UNIT SEPARATOR> elements.
<PROGRAM MESSAGE UNIT> figure 7-3	Represents a single command, programming data, or query received by the **device**.
<COMMAND MESSAGE UNIT> figure 7-4	Represents a single command or programming data received by the **device**.
<QUERY MESSAGE UNIT> figure 7-5	Represents a single query sent from the controller to the **device**.
<PROGRAM DATA> figure 7-6	Represents any of the six different program data types.
<PROGRAM MESSAGE UNIT SEPARATOR> 7.4.1	Separates the <PROGRAM MESSAGE UNIT> elements from one another in a <PROGRAM MESSAGE>
<PROGRAM DATA SEPARATOR> 7.4.2	Separates sequential <PROGRAM DATA> elements that are related to the same header.
<PROGRAM HEADER SEPARATOR> 7.4.3	Separates the header from any associated <PROGRAM DATA>.
<PROGRAM MESSAGE TERMINATOR> 7.5	Terminates a <PROGRAM MESSAGE>.
<COMMAND PROGRAM HEADER> 7.6.1	Specifies function or operation. Used with any associated <PROGRAM DATA> element(s).
<QUERY PROGRAM HEADER> 7.6.2	Similar to <COMMAND PROGRAM HEADER> except a query indicator (?) shows that a response is expected from the device.
<CHARACTER PROGRAM DATA> 7.7.1	A data type suitable for sending short mnemonic data, generally where a numeric data type is not suitable.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Elément	*Fonction*
<DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> 7.7.2	Type de donnée convenant pour envoyer des nombres entiers décimaux ou des fractions décimales avec ou sans exposants.
<DONNÉES DE PROGRAMMATION DE SUFFIXES> 7.7.3	Espace facultatif après une <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> utilisé pour indiquer les multiplicateurs et les unités associés.
<DONNÉES DE PROGRAMMATION NUMÉRIQUES NON DÉCIMALES> 7.7.4	Type de donnée convenant pour envoyer la représentation de nombres entiers en base 16, 8 ou 2. Utile pour les données qui sont plus facilement interprétées lorsqu'elles sont exprimées directement dans un format non décimal.
<DONNÉES DE PROGRAMMATION DE CHAÎNES> 7.7.5	Type de donnée convenant pour envoyer des chaînes de caractères de 7 bits ISO dans lesquelles le contenu nécessite d'être "caché" (par des séparateurs).
<DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> 7.7.6	Type de donnée convenant pour envoyer des blocs d'informations arbitraires de 8 bits.
<DONNÉES DE PROGRAMMATION D'EXPRESSION> 7.7.7	Type de donnée convenant pour envoyer des données qui, avant l'analyse syntaxique, sont évaluées en un ou plusieurs éléments de données, qui ne forment pas une expression.

7.4 *Eléments fonctionnels de séparation*

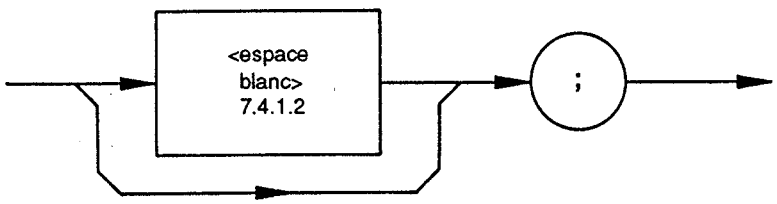
7.4.1 *<SÉPARATEUR D'UNITÉS DE MESSAGES DE PROGRAMMATION>*

7.4.1.1 Fonction

Le <SÉPARATEUR D'UNITÉS DE MESSAGES DE PROGRAMMATION> sépare les éléments séquentiels d'<UNITÉ DE MESSAGES DE PROGRAMMATION> les uns des autres à l'intérieur d'un <MESSAGE DE PROGRAMMATION>.

7.4.1.2 Syntaxe de codage

Un <SÉPARATEUR D'UNITÉS DE MESSAGES DE PROGRAMMATION> est défini par



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Element	*Function*
<DECIMAL NUMERIC PROGRAM DATA> 7.7.2	A data type suitable for sending decimal integers or decimal fractions with or without exponents.
<SUFFIX PROGRAM DATA> 7.7.3	An optional field following <DECIMAL NUMERIC PROGRAM DATA> used to indicate associated multipliers and units.
<NON-DECIMAL NUMERIC PROGRAM DATA> 7.7.4	A data type suitable for sending integer numeric representations in base 16, 8, or 2. Useful for data that is more easily interpreted when directly expressed in a non-decimal format.
<STRING PROGRAM DATA> 7.7.5	A data type suitable for sending ISO 7-bit character strings where the content needs to be "hidden" (by delimiters).
<ARBITRARY BLOCK PROGRAM DATA> 7.7.6	A data type suitable for sending blocks of arbitrary 8-bit information.
<EXPRESSION PROGRAM DATA> 7.7.7	A data type suitable for sending data that is evaluated as one or more non-expression data elements before further parsing.

7.4 *Separator functional elements*

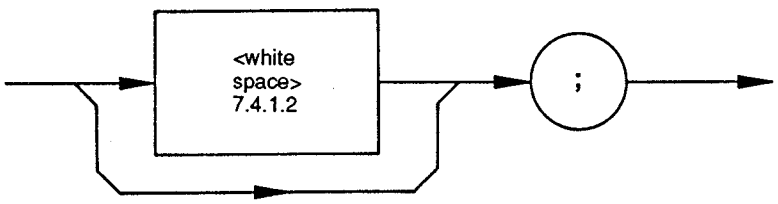
7.4.1 *<PROGRAM MESSAGE UNIT SEPARATOR>*

7.4.1.1 Function

The <PROGRAM MESSAGE UNIT SEPARATOR> separates sequential <PROGRAM MESSAGE UNIT> elements from one another within a <PROGRAM MESSAGE>.

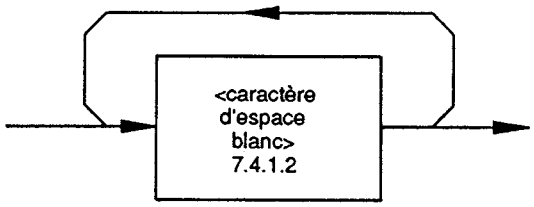
7.4.1.2 Encoding syntax

A <PROGRAM MESSAGE UNIT SEPARATOR> is defined as



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Où <espace blanc> est défini par



Où <caractère d'espace blanc> est défini comme un octet simple codé de 7 bits ISO dans la gamme 00-09, 0B – 20 (0-9, 11-32 décimal). Cette gamme inclut les caractères de contrôle de 7 bits ISO et l'espace mais exclut l'interligne.

7.4.1.3 Equivalence sémantique

Un **appareil** traitera un <espace blanc> sans interprétation sémantique.

Un **appareil** devra interpréter le point-virgule (;) comme le <SÉPARATEUR d'UNITÉS DE MESSAGES DE PROGRAMMATION>. Aucune variante de codage n'est autorisée.

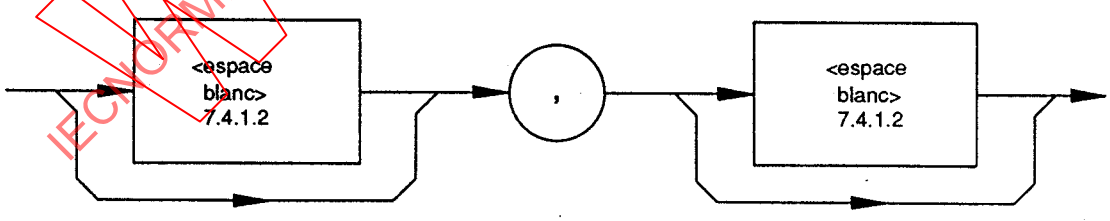
7.4.2 *<SÉPARATEUR DE DONNÉES DE PROGRAMMATION>*

7.4.2.1 Fonction

Le <SÉPARATEUR DE DONNÉES DE PROGRAMMATION> sépare les éléments séquentiels de <DONNÉES DE PROGRAMMATION> les uns des autres après un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. Il est utilisé lorsqu'un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> possède de multiples paramètres.

7.4.2.2 Syntaxe de codage

Un <SÉPARATEUR DE DONNÉES DE PROGRAMMATION> est défini par



Où <espace blanc> est défini en 7.4.1.2.

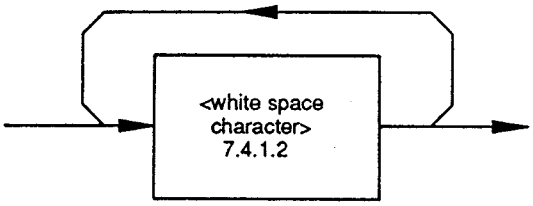
7.4.2.3 Equivalence sémantique

Un **appareil** traitera un <espace blanc> sans interprétation sémantique.

Un **appareil** devra interpréter la virgule (,) comme le <SÉPARATEUR DE DONNÉES DE PROGRAMMATION>. Aucune variante de codage n'est autorisée.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Where <white space> is defined as



Where <white-space character> is defined as a single ISO 7-bit encoded byte in the range 00-09, 0B – 20 (0-9, 11-32 decimal). This range includes the ISO 7-bit control characters and the space but excludes the newline.

7.4.1.3 Semantic equivalence

A **device** shall process <white space> without semantic interpretation.

A **device** shall interpret the semicolon (;) as the <PROGRAM MESSAGE UNIT SEPARATOR>. No alternative encodings are allowed.

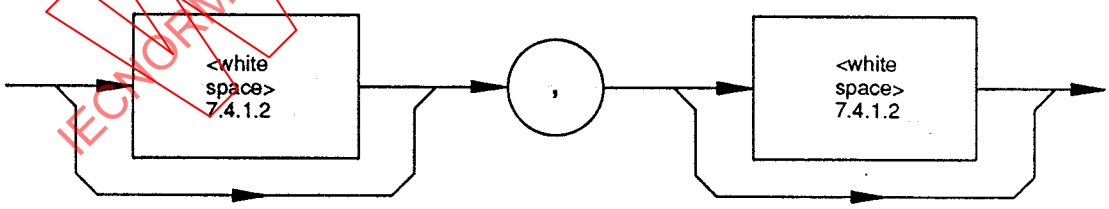
7.4.2 *<PROGRAM DATA SEPARATOR>*

7.4.2.1 Function

The <PROGRAM DATA SEPARATOR> separates sequential <PROGRAM DATA> elements from one another after a <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER>. It is used when a <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> has multiple parameters.

7.4.2.2 Encoding syntax

A <PROGRAM DATA SEPARATOR> is defined as



Where <white space> is defined in 7.4.1.2.

7.4.2.3 Semantic equivalence

A **device** shall process <white space> without semantic interpretation.

A **device** shall interpret the comma (,) as the <PROGRAM DATA SEPARATOR>. No alternative encodings are allowed.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

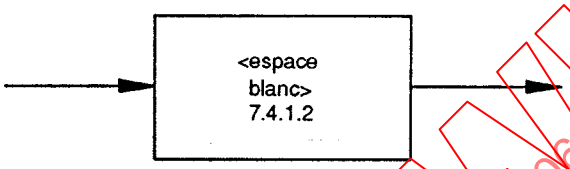
7.4.3 *<SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION>*

7.4.3.1 Fonction

Le <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> sépare l'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou l'<EN-TÊTE D'NTERROGATION DE PROGRAMMATION> des éléments de <DONNÉES DE PROGRAMMATION>.

7.4.3.2 Syntaxe de codage

Un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> est défini par



Où <espace blanc> est défini en 7.4.1.2.

7.4.3.3 Equivalence sémantique

Un **appareil** doit interpréter le premier <caractère d'espace blanc> comme le <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION>. Aucune variante de codage n'est autorisée.

Aucun <espace blanc> supplémentaire ne doit avoir de valeur syntaxique ou sémantique.

7.5 *<CARACTÈRE D'ARRÊT DE MESSAGES DE PROGRAMMATION>*

7.5.1 *Fonction*

Un <CARACTÈRE D'ARRÊT DE MESSAGES DE PROGRAMMATION> termine une séquence d'un ou plusieurs éléments d'<UNITÉ DE MESSAGES DE PROGRAMMATION> d'une longueur définie. Un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> est abrégé dans la présente norme par <PMT>.

Le caractère d'arrêt apporte un sens sémantique supplémentaire en ce qui concerne l'ordre d'exécution des instructions interactives ou "couplées", voir 6.4.5.3, l'opération de file d'attente de sortie, voir 6.1.10, et l'utilisation syntaxique des en-têtes complexes, voir 7.6.1.5 et 7.6.2.5.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.4.3 *<PROGRAM HEADER SEPARATOR>*

7.4.3.1 Function

The <PROGRAM HEADER SEPARATOR> separates the <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> from the <PROGRAM DATA> elements.

7.4.3.2 Encoding syntax

A <PROGRAM HEADER SEPARATOR> is defined as



Where <white space> is defined in clause 7.4.1.2.

7.4.3.3 Semantic equivalence

A **device** shall interpret the first <white-space character> as the <PROGRAM HEADER SEPARATOR>. No alternative encodings are allowed.

Any additional <white-space> shall have no syntactic nor semantic value.

7.5 *<PROGRAM MESSAGE TERMINATOR>*

7.5.1 *Function*

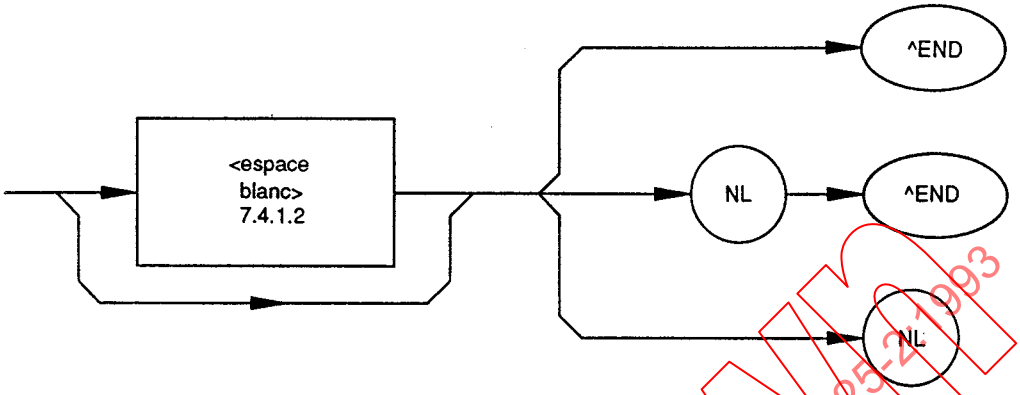
A <PROGRAM MESSAGE TERMINATOR> terminates a sequence of one or more definite length <PROGRAM MESSAGE UNIT> elements. A <PROGRAM MESSAGE TERMINATOR> is abbreviated in this standard as <PMT>.

The terminator conveys additional semantic meaning with regard to the execution order of interactive or "coupled" commands, see 6.4.5.3, to the operation of the Output Queue, see 6.1.10, and to the syntactic use of compound headers, see 7.6.1.5 and 7.6.2.5.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.5.2 *Syntaxe de codage*

Un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> est défini par



Où <espace blanc> est défini en 7.4.1.2.

Où NL est défini par un seul octet codé ISO OA (10 décimal).

7.5.3 *Equivalence sémantique*

Un **appareil** traitera un <espace blanc> sans interprétation sémantique.

Un **appareil** devra interpréter chacun des trois caractères d'arrêt (c'est-à-dire: message END (FIN) sur le dernier octet de données, interligne avec message END, ou interligne) comme sémantiquement équivalents. Aucune variante de codage n'est autorisée.

7.6 *Eléments fonctionnels d'en-tête de programmation*

7.6.1 *<EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION>*

7.6.1.1 Fonction

L'<EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION> représente l'opération à exécuter dans un **appareil**. Cet élément peut être facultativement suivi de paramètres associés codés comme des éléments de <DONNÉES DE PROGRAMMATION>, voir 7.7.

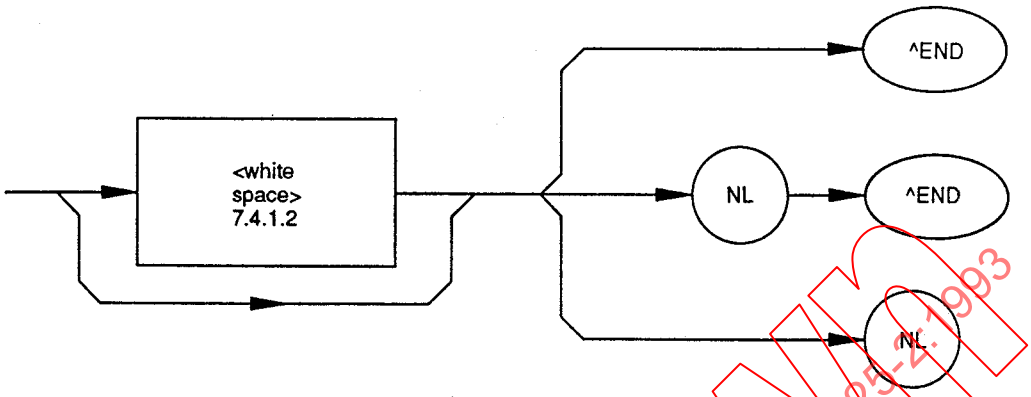
Noter qu'une étiquette de macro (voir *DMC, voir 10.7) peut se présenter sous forme d'un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> mais peut annoter une séquence d'éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> qui contient des éléments d'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> corrects (voir 7.6.2.1) et générerait ainsi une réponse lorsqu'elle serait traitée. L'instruction commune *TRG (voir 10.37) peut également être spécifiée pour générer une réponse.

Un <en-tête d'instructions de programmation complexe> autorise une structure interne dans l'en-tête. Cette structure est généralement utilisée par des appareils plus complexes pour limiter le nombre d'en-têtes uniques et aussi pour organiser logiquement le jeu d'instructions de l'appareil.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.5.2 *Encoding syntax*

A <PROGRAM MESSAGE TERMINATOR> is defined as



Where <white space> is defined in 7.4.1.2.

Where NL is defined as a single ISO 7-bit encoded byte 0A (10 decimal).

7.5.3 *Semantic equivalence*

A **device** shall process <white space> without semantic interpretation.

A **device** shall interpret any and all of the three terminators (that is: END message on last data byte, newline with END message, or newline) as semantically equivalent. No alternative encodings are allowed.

7.6 *Program Header functional elements*

7.6.1 *<COMMAND PROGRAM HEADER>*

7.6.1.1 Function

The <COMMAND PROGRAM HEADER> represents the operation to be performed in a **device**. This element may be optionally followed by associated parameters encoded as <PROGRAM DATA> elements, see 7.7.

Note that a macro label (see *DMC, see 10.7) may have the form of a <COMMAND PROGRAM HEADER> but may label a sequence of <PROGRAM MESSAGE UNIT> elements that contains valid <QUERY PROGRAM HEADER> elements (see 7.6.2.1) and thus would generate a response when processed. The *TRG common command (see 10.37) may also be specified to generate a response.

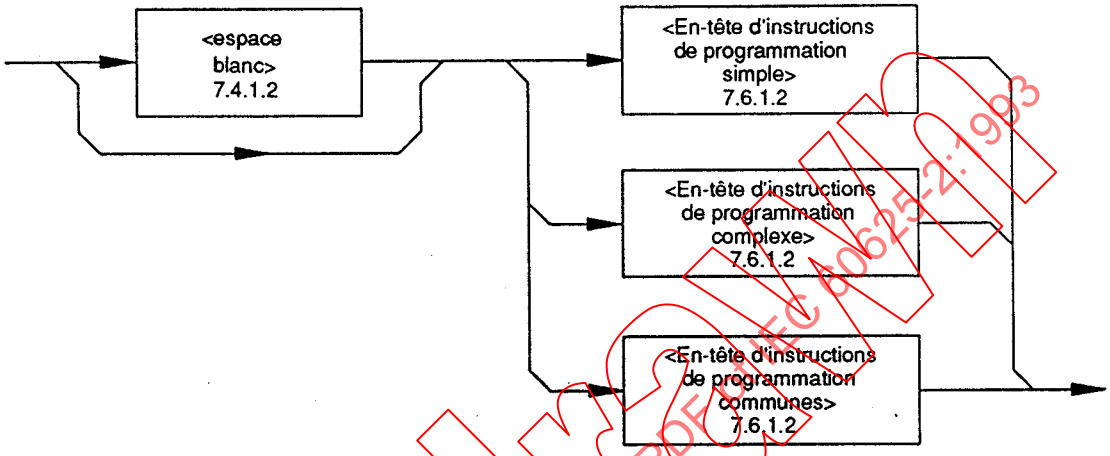
A <compound command program header> allows for internal structure in the header. This structure is generally used by more complex devices to limit the number of unique headers and also to organize logically the device command set.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Les éléments de l'<en-tête d'instructions de programmation complexe> sont utiles pour le traitement de structures hiérarchisées d'instructions. Les éléments de l'<en-tête d'instructions de programmation complexe> peuvent fournir des notations abrégées de mnémoniques lorsqu'ils sont utilisés à l'intérieur d'un <MESSAGE DE PROGRAMMATION> contenant plusieurs éléments d'<UNITÉ DE MESSAGES DE PROGRAMMATION>.

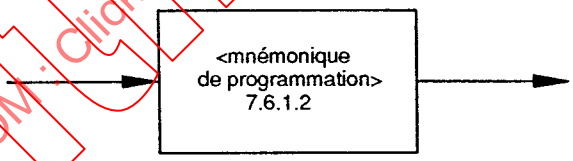
7.6.1.2 Syntaxe de codage

Un <EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION> est défini par

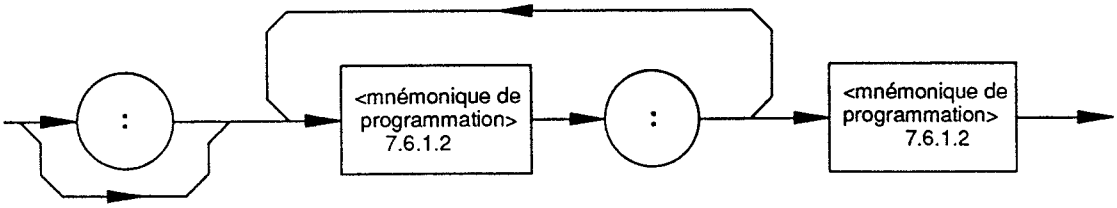


Où <espace blanc> est défini en 7.4.1.2.

Où <en-tête d'instructions de programmation simples> est défini par



Où <en-tête d'instructions de programmation complexes> est défini par

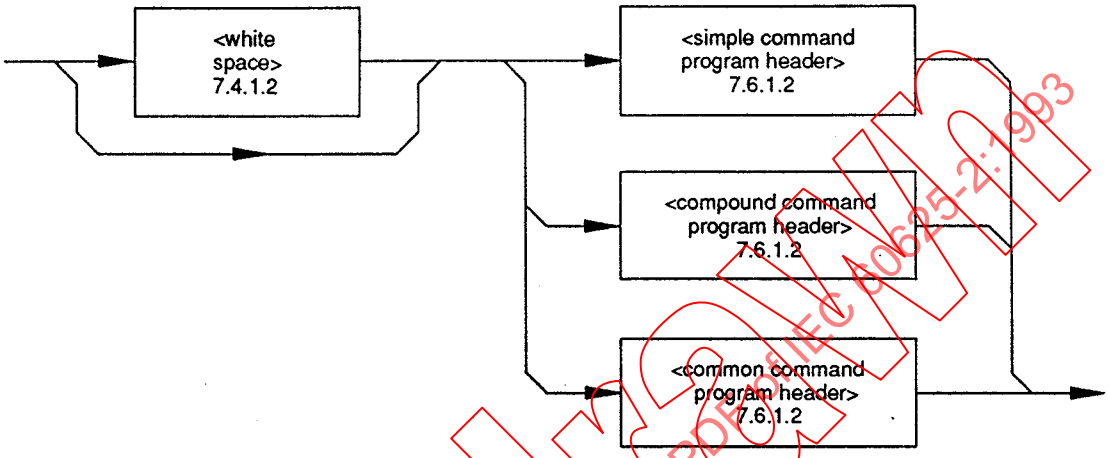


IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

<compound command program header> elements are useful for handling hierarchically related command structures. <compound command program header> elements can provide shorthand mnemonic notation when used within a <PROGRAM MESSAGE> containing several <PROGRAM MESSAGE UNIT> elements.

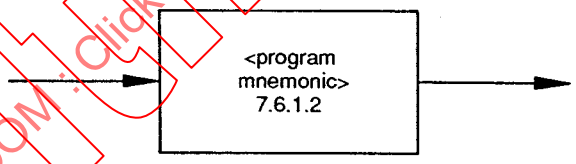
7.6.1.2 Encoding syntax

A <COMMAND PROGRAM HEADER> is defined as

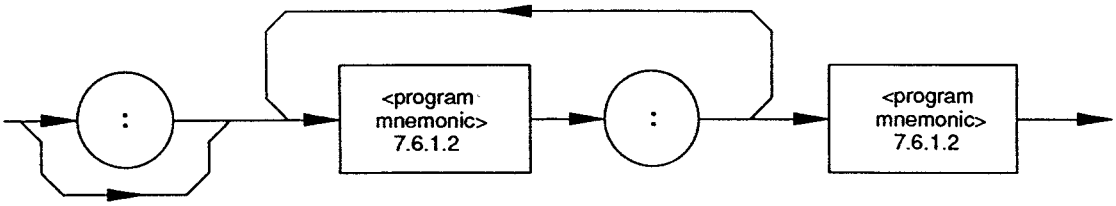


Where <white space> is defined in 7.4.1.2.

Where <simple command program header> is defined as

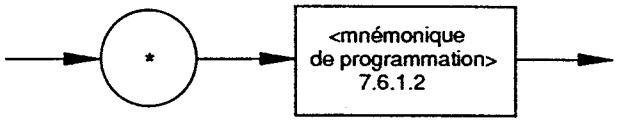


Where <compound command program header> is defined as

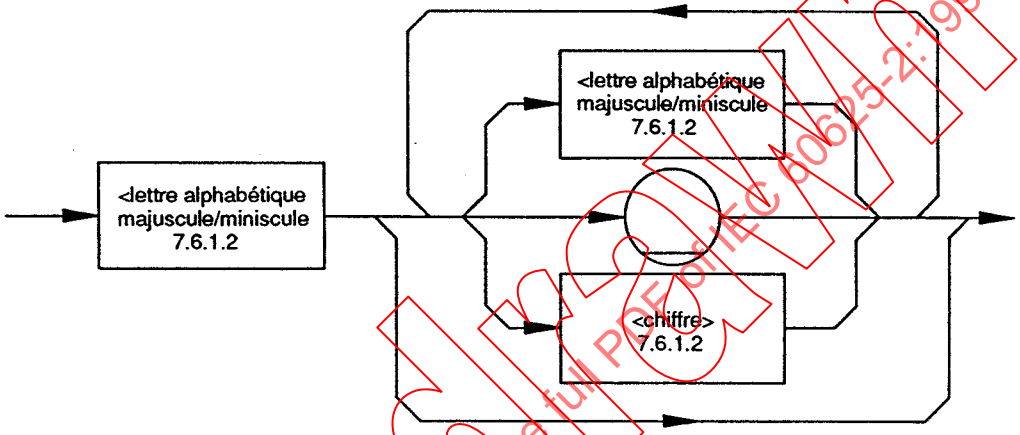


IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Où <en-tête d'instructions de programmation communes> est défini par



Où <mnémonique de programmation> est défini par



Où <lettre alphabétique majuscule/minuscule> est définie comme un octet simple codé ISO dans la gamme 41 – 5A, 61 – 7A (65-90, 97 – 122 en décimal).

Où <chiffre> est défini comme un octet simple codé ISO dans la gamme 30 – 39 (48 – 57 décimal).

Où (_) représente un "soulignement", un octet simple codé ISO avec la valeur 5F (95 en décimal).

7.6.1.3 Equivalence sémantique

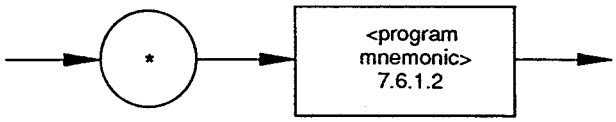
Un **appareil** doit interpréter la <lettre alphabétique majuscule/minuscule> contenue dans le mnémonique de programmation sans attacher de sens sémantique à la casse (majuscules/minuscules) des caractères alphabétiques.

Un **appareil** doit interpréter les deux points (:) comme le séparateur du <mnémonique de programmation> dans l'élément <en-tête de programmation d'instructions complexe>. Aucune variante de codage n'est autorisée.

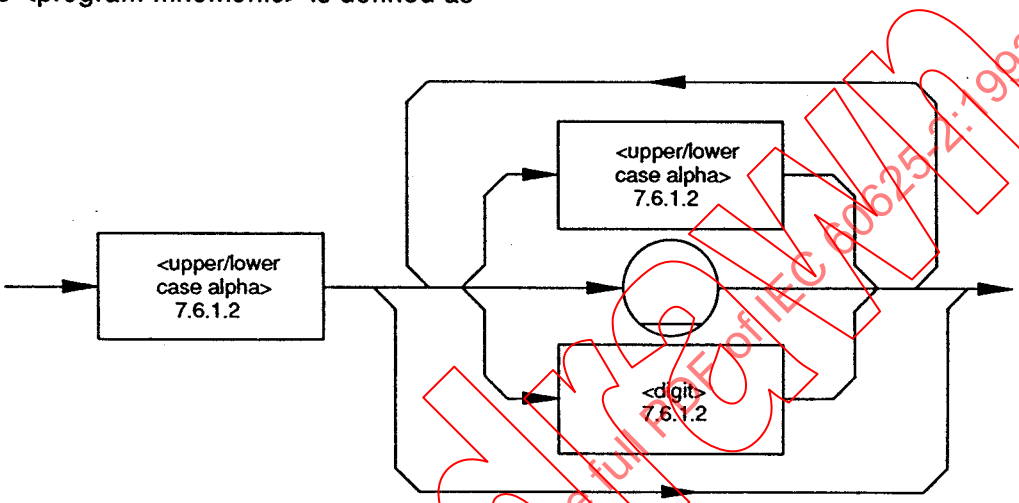
Un **appareil** doit traiter un <espace blanc> contenu dans un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> sans interprétation sémantique.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Where <common command program header> is defined as



Where <program mnemonic> is defined as



Where <upper/lower-case alpha> is defined as a single ISO 7-bit encoded byte in the range 41 – 5A, 61 – 7A (65-90, 97-122 decimal).

Where <digit> is defined as a single ISO 7-bit encoded byte in the range 30 – 39 (48 – 57 decimal).

Where (_) represents an "underscore", a single ISO 7-bit encoded byte with the value 5F (95 decimal).

7.6.1.3 Semantic equivalence

A **device** shall interpret <upper/lower-case alpha> contained within the <program mnemonic> without attaching semantic meaning to the case of the alpha characters.

A **device** shall interpret the colon (:) as the <program mnemonic> separator in the <compound command program header> element. No alternative encodings are allowed.

A **device** shall process <white space> contained within a <COMMAND PROGRAM HEADER> without semantic interpretation.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.6.1.4 Règles

7.6.1.4.1 Longueur

Le <mnémonique de programmation> doit avoir au maximum une longueur de 12 caractères avec une longueur préférentielle de quatre caractères.

7.6.1.4.2 Directives de désignation des <EN-TÊTES D'INSTRUCTIONS DE PROGRAMMATION>

La relation entre le mnémonique de l'<EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION> et sa fonction associée devra être évidente. Ces mnémoniques devront être indépendants de l'emplacement du contrôle sur la platine, mais devront se rapporter à la platine ou à la nomenclature affichée comme il convient.

7.6.1.4.3 Règles de l'<en-tête d'instructions de programmation communes>

La syntaxe d'<en-tête d'instructions de programmation communes> est d'usage réservé par cette norme et les futures révisions de celle-ci. Des instructions et interrogations communes définies utilisant cette syntaxe sont traitées à l'article 10.

Les concepteurs d'appareils ne devront pas utiliser l'<en-tête d'instructions de programmation communes> défini à la section 10 pour d'autres intentions que celles définies à l'article 10.

7.6.1.5 Règles de composition des en-têtes

La partie d'un <en-tête d'instructions de programmation complexe>, qui exclut l'élément <mnémonique de programmation> suivant, est définie comme étant la "branche d'en-tête". Un <en-tête d'instruction de programmation complexe> ou <en-tête d'instruction de programmation simple> devra être interprété par un **appareil** comme si la branche d'en-tête, dans l'<en-tête d'instruction de programmation complexe> ou l'<en-tête d'interrogation de programmation complexe> immédiatement précédent, était envoyée immédiatement avant cet <en-tête d'instruction de programmation complexe> ou <en-tête d'instruction de programmation simple>. La branche d'en-tête fournie devra inclure à la fois toute branche d'en-tête explicitement spécifiée dans l'instruction immédiatement précédente et toute branche d'en-tête fournie à l'instruction immédiatement précédente par une application préalable de la règle.

Cette règle ne doit PAS être appliquée à:

1) Un <en-tête d'instruction de programmation complexe> qui commence un message de programmation.

2) Un <en-tête d'instruction de programmation complexe> qui commence par deux points (:).

La présence d'un <en-tête d'instruction de programmation commune> ou <en-tête d'interrogation de programmation commune> n'a aucun effet sur la branche d'en-tête.

Voir annexe A pour des exemples.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.6.1.4 Rules

7.6.1.4.1 Length

The <program mnemonic> shall have a maximum length of 12 characters with a preferred length of four characters.

7.6.1.4.2 <COMMAND PROGRAM HEADER> naming guidelines

The relation of a <COMMAND PROGRAM HEADER> mnemonic and its associated function shall be readily apparent. These mnemonics shall be independent of control location on the front panel but shall relate to front panel or display nomenclature as appropriate.

7.6.1.4.3 <common command program header> rules

The <common command program header> syntax is reserved for use by this standard and future revisions of this standard. Defined common commands and queries using this syntax are discussed in clause 10.

Device designers shall not use the <common command program header> defined in clause 10 for purposes other than those defined in clause 10.

7.6.1.5 Header compounding rules

That part of a <compound command program header> that excludes the trailing <program mnemonic> element is defined as the "header-path". A <compound command program header> or <simple command program header> shall be interpreted by a **device** as if the header-path in the immediately preceding <compound command program header> or <compound query program header>, was sent immediately preceding that <compound command program header> or <simple command program header>. The supplied header-path shall include both any header-path explicitly stated in the immediately preceding command and any header-path supplied to the immediately preceding command by a previous application of the rule.

This rule shall NOT be applied to:

1) A <compound command program header> that begins a <PROGRAM MESSAGE>.

2) A <compound command program header> that begins with a colon (:).

The presence of a <common command program header> or <common query program header> has no effect on the header-path.

See annex A for examples.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.6.2 *<EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION>*

7.6.2.1 Fonction

L'<EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION> représente l'opération à réaliser dans un **appareil**. Un <EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION> fait générer une réponse à l'**appareil**. Cet élément peut être facultativement suivi de paramètres associés codés comme des éléments de <DONNÉES DE PROGRAMMATION>, voir 7.7.

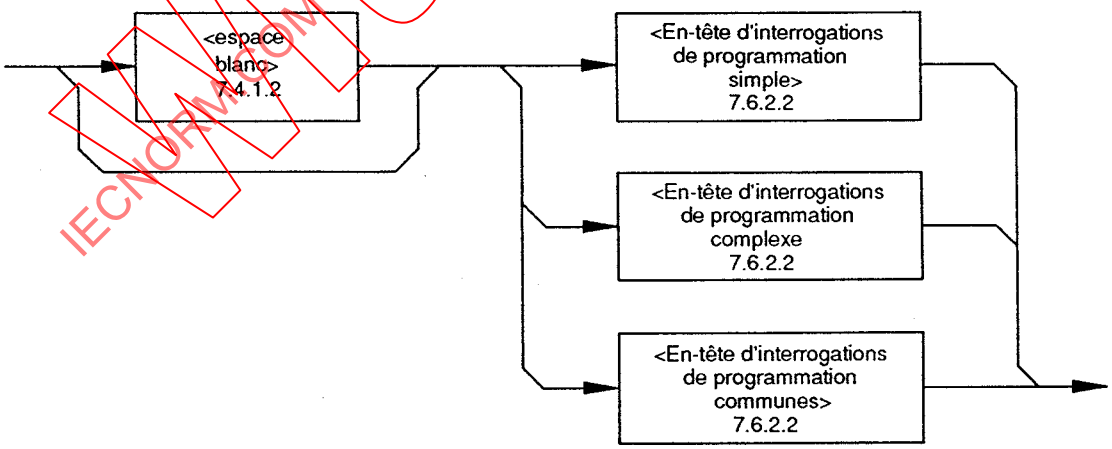
Noter qu'une étiquette de macro (voir *DMC, voir 10.7) peut se présenter sous la forme d'un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>, mais peut annoter une séquence d'éléments d'<UNITÉ DE MESSAGE DE PROGRAMMATION> qui ne contient aucun élément d'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> correct et ainsi ne générerait aucune réponse lorsqu'elle serait traitée.

L'<en-tête d'interrogations de programmation complexe> représente la fonction à réaliser conjointement avec les éléments de données possibles suivants pour former une instruction de programmation. Un <en-tête d'interrogations de programmation complexe> autorise une structure interne dans l'en-tête.

Cette structure est généralement utilisée par des **appareils** plus complexes pour limiter le nombre d'en-têtes uniques et aussi pour organiser logiquement le jeu d'instructions de l'appareil pour un contrôle plus facile et un code plus lisible. Les éléments d'<en-tête d'interrogations de programmation complexe> sont utiles pour traiter les structures d'instructions hiérarchisées. Les éléments d'<en-tête d'interrogations de programmation complexe> sont capables de fournir des notations abrégées de mnémoniques lorsqu'ils sont utilisés à l'intérieur d'éléments de <MESSAGES DE PROGRAMMATION> d'une unité multi-messages.

7.6.2.2 Syntaxe de codage

Un <EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION> est défini par



Où <espace blanc> est défini en 7.4.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.6.2 *<QUERY PROGRAM HEADER>*

7.6.2.1 Function

The <QUERY PROGRAM HEADER> represents the operation to be performed in a **device**. A <QUERY PROGRAM HEADER> causes the **device** to generate a response. This element may be optionally followed by associated parameters encoded as <PROGRAM DATA> elements, 7.7.

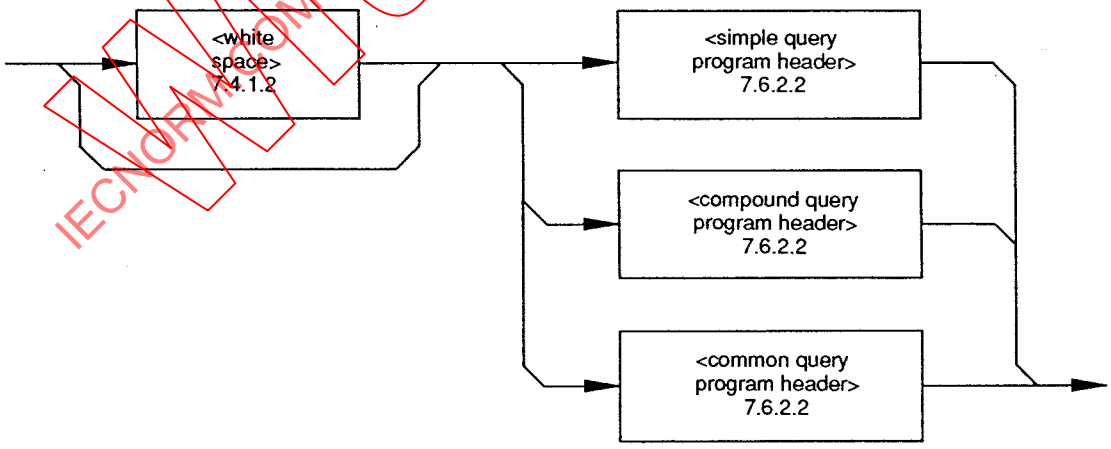
Note that a macro label (see *DMC, 10.7) may have the form of a <QUERY PROGRAM HEADER> but may label a sequence of <PROGRAM MESSAGE UNIT> elements that contains no valid <QUERY PROGRAM HEADER> elements and thus would generate no response when processed.

The <compound query program header> represents the function to be performed in conjunction with possible following data elements to form a program command. A <compound query program header> allows for internal structure in the header.

This structure is generally used by more complex **devices** to limit the number of unique headers and also to organize logically the device command set for easier control and more readable code. <compound query program header> elements are useful for handling command structures that are hierarchically related. <compound query program header> elements are capable of providing a shorthand mnemonic notation when used within multi-message-unit <PROGRAM MESSAGE> elements.

7.6.2.2 Encoding syntax

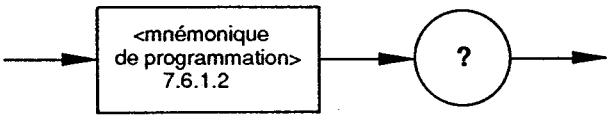
A <QUERY PROGRAM HEADER> is defined as



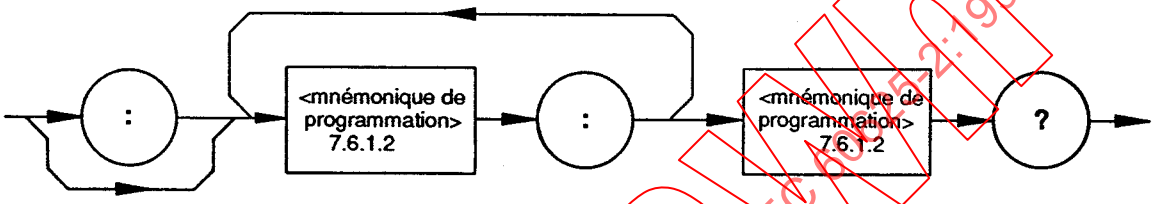
Where <white space> is defined in 7.4.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

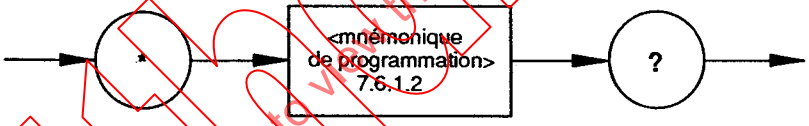
Où <en-tête d'interrogations de programmation simple> est défini par



Où <en-tête d'interrogations de programmation complexe> est défini par



Où <en-tête d'interrogations de programmation commune> est défini par



Où <mnémonique de programmation> est défini en 7.6.1.2.

7.6.2.3 Equivalence sémantique

Les règles d'équivalence sémantique pour l'<EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION> sont les mêmes que pour l'<EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION>, voir 7.6.1.3.

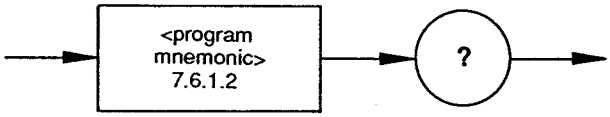
Un **appareil** doit traiter un <espace blanc> contenu dans un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> sans interprétation sémantique.

7.6.2.4 Règles

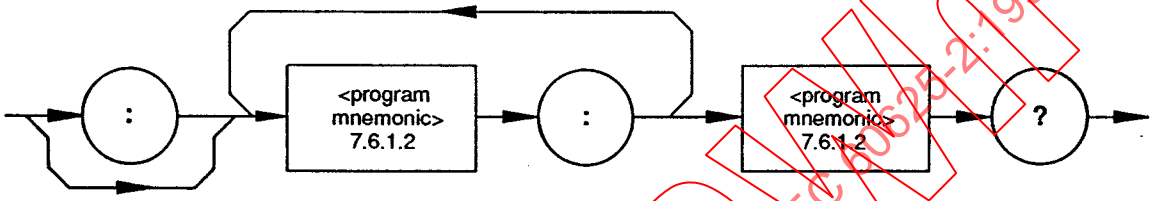
Les règles générales pour l'<EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION> sont les mêmes que pour l'<EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION>, voir 7.6.1.4.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

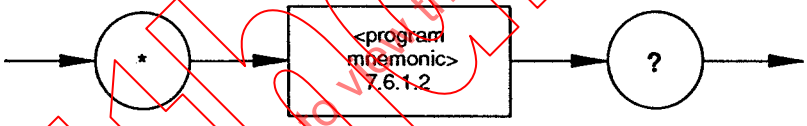
Where <simple query program header> is defined as



Where <compound query program header> is defined as



Where <common query program header> is defined as



Where <program mnemonic> is defined in 7.6.1.2.

7.6.2.3 Semantic equivalence

Semantic equivalence rules for <QUERY PROGRAM HEADER> are the same as for <COMMAND PROGRAM HEADER>, see 7.6.1.3.

A **device** shall process <white space> contained within a <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> without semantic interpretation.

7.6.2.4 Rules

General rules for <QUERY PROGRAM HEADER> are the same as for <COMMAND PROGRAM HEADER>, see 7.6.1.4.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.6.2.5 Règles de composition des en-têtes

Les règles pour les <en-têtes d'interrogations de programmation complexe> sont les mêmes que les règles pour les <en-têtes d'instructions de programmation complexe>, voir 7.6.1.5.

Voir l'annexe A pour des exemples.

7.7 *Eléments fonctionnels de <DONNÉES DE PROGRAMMATION>*

Un élément fonctionnel de <DONNÉES DE PROGRAMMATION> est utilisé pour acheminer une variété de types de paramètres d'informations relatifs à l'en-tête de programmation.

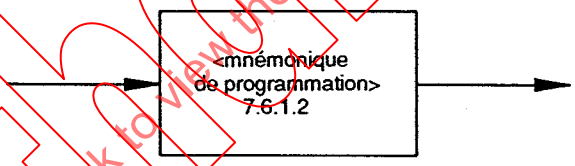
7.7.1 *<DONNÉES DE PROGRAMMATION DE CARACTÈRES>*

7.7.1.1 Fonction

L'élément fonctionnel de <DONNÉES DE PROGRAMMATION DE CARACTÈRES> est utilisé pour acheminer des paramètres d'informations mieux exprimés mnémoniquement comme une courte chaîne alphabétique ou alphanumérique. Il est utile dans les cas où les paramètres numériques sont inappropriés.

7.7.1.2 Syntaxe de codage

Un élément de <DONNÉES DE PROGRAMMATION DE CARACTÈRES> est défini par



Où <mnémonique de programmation> est défini en 7.6.1.2.

7.7.1.3 Equivalence sémantique

Le <mnémonique de programmation>, tel qu'il est utilisé dans l'élément fonctionnel de <DONNÉES DE PROGRAMMATION DE CARACTÈRES>, devra suivre les mêmes règles sémantiques d'équivalence que lorsqu'il est utilisé dans l'élément d'<EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION>, voir 7.6.1.3.

7.7.1.4 Règles

Le <mnémonique de programmation>, tel qu'il est utilisé dans l'élément fonctionnel de <DONNÉES DE PROGRAMMATION DE CARACTÈRES>, devra suivre les mêmes règles générales que lorsqu'il est utilisé dans l'élément d'<EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION>, voir 7.6.1.4.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.6.2.5 Header compounding rules

Rules for <compound query program headers> are the same as the rules for <compound command program headers>, see 7.6.1.5.

See annex A for examples.

7.7 *<PROGRAM DATA> functional elements*

A <PROGRAM DATA> functional element is used to convey a variety of types of parameter information related to the program header.

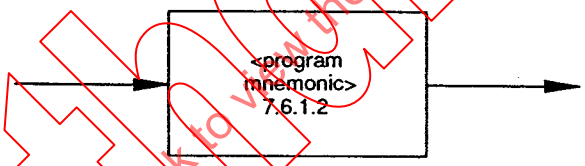
7.7.1 *<CHARACTER PROGRAM DATA>*

7.7.1.1 Function

The <CHARACTER PROGRAM DATA> functional element is used to convey parameter information best expressed mnemonically as a short alpha or alphanumeric string. It is useful in those cases where numeric parameters are inappropriate.

7.7.1.2 Encoding syntax

A <CHARACTER PROGRAM DATA> element is defined as



Where <program mnemonic> is defined in 7.6.1.2.

7.7.1.3 Semantic equivalence

The <program mnemonic> as used in the <CHARACTER PROGRAM DATA> functional element shall follow the same semantic equivalence rules as when it is used in <COMMAND PROGRAM HEADER> element, see 7.6.1.3.

7.7.1.4 Rules

The <program mnemonic> as used in the <CHARACTER PROGRAM DATA> functional element shall follow the same general rules as when it is used in the <COMMAND PROGRAM HEADER> element, see 7.6.1.4.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.2 *<DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES>*

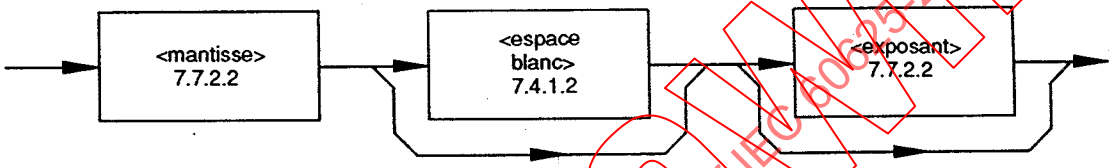
7.7.2.1 Fonction

La <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> est une version flexible des trois représentations numériques (NR1, 2 et 3) comme défini dans ISO 6093.

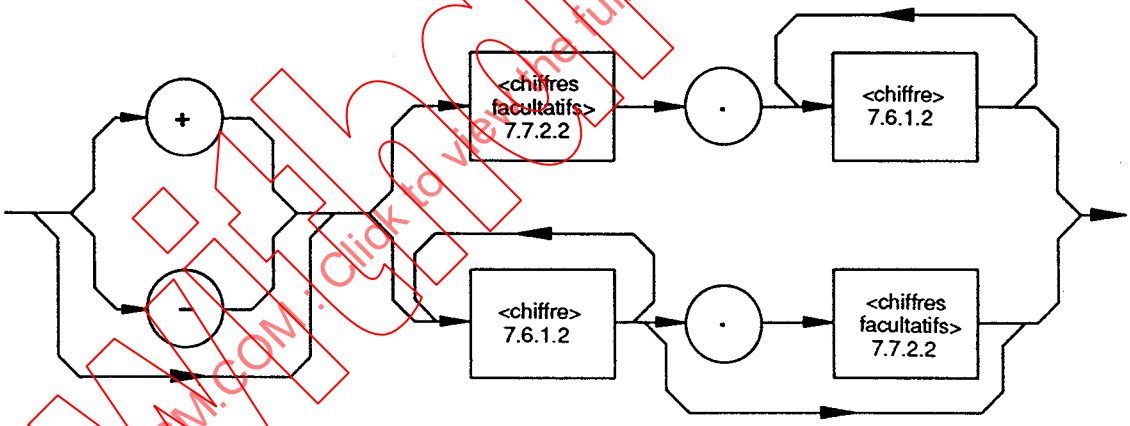
Cet élément est aussi noté NRf (représentation numérique flexible) dans les sections suivantes.

7.7.2.2 Syntaxe de codage

Un élément de <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> est défini par

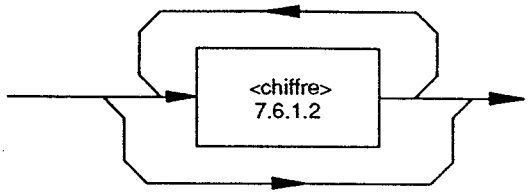


Où <mantisse> est défini par



Où <chiffre> est défini en 7.6.1.2.

Où <chiffres facultatifs> est défini par



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.2 *<DECIMAL NUMERIC PROGRAM DATA>*

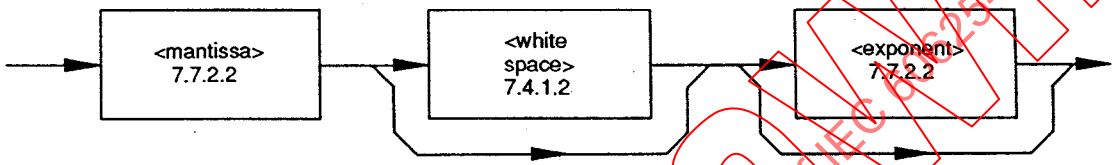
7.7.2.1 Function

The <DECIMAL NUMERIC PROGRAM DATA> is a flexible version of the three numeric representations (NR1, 2 and 3) as defined in ISO 6093.

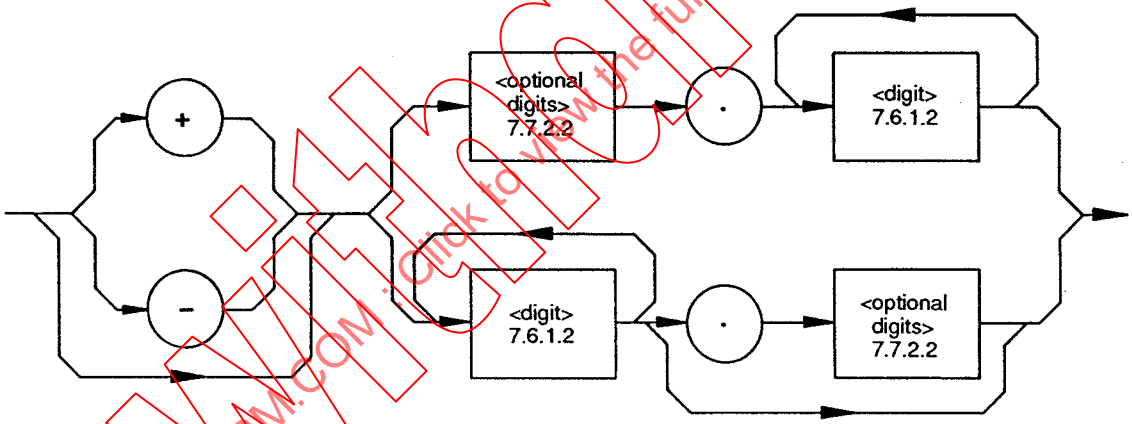
This element is also denoted by NRf (flexible numeric representation) in subsequent sections.

7.7.2.2 Encoding syntax

A <DECIMAL NUMERIC PROGRAM DATA> element is defined as

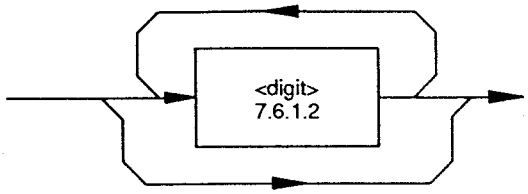


Where <mantissa> is defined as



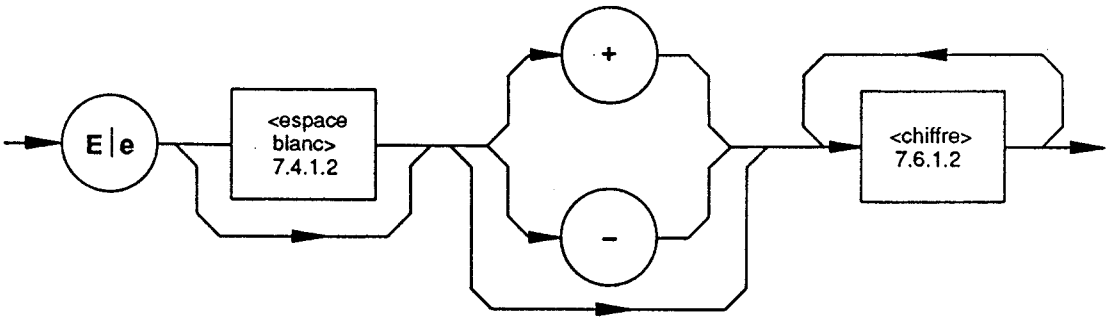
Where <digit> is defined in 7.6.1.2.

Where <optional digits> is defined as



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Où <exposant> est défini par



Où <chiffre> est défini en 7.6.1.2.

Où <espace blanc> est défini en 7.4.1.2.

7.7.2.3 Equivalence sémantique

Un **appareil** doit traiter un <espace blanc> sans interprétation sémantique.

Un **appareil** devra interpréter tous les autres caractères codés de 7 bits ISO comme comprenant un format numérique décimal suivant généralement n'importe quel format NR1, NR2 ou NR3 de l'ISO 6093, comme cela est autorisé par le diagramme en 7.7.2.2. La valeur interprétée devra être soumise à la règle d'arrondi qui se trouve en 7.7.2.4.2.

7.7.2.4 Règles

7.7.2.4.1 Gamme

La mantisse d'un élément de <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> ne devra pas contenir plus de 255 caractères, à l'exclusion de tous les zéros de tête.

Les éléments de <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> devront avoir une valeur d'exposant dans la gamme de –32 000 à +32 000.

Si l'élément de <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> enfreint l'une quelconque de ces règles, une erreur d'instruction doit être générée.

7.7.2.4.2 Arrondi des éléments numériques

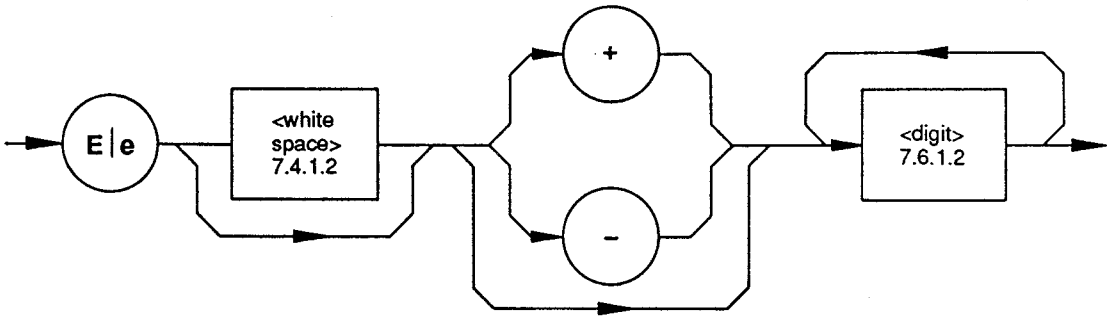
Un **appareil** peut recevoir un élément de <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> ayant une plus grande précision que celle que l'**appareil** peut traiter intérieurement. Dans ce cas, l'**appareil** devra arrondir le nombre, et ne pas le tronquer, avant interprétation. Lors de l'arrondi, l'**appareil** ne devra pas tenir compte du signe du nombre et arrondir à des valeurs supérieures ou égales aux demi-valeurs exactes. Les valeurs plus petites que les demi-valeurs exactes devront être arrondies à la valeur inférieure.

7.7.2.4.3 Indication d'erreur

Toutes les conditions d'erreurs indiquées devront être basées sur le résultat de la valeur interprétée après arrondi.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Where <exponent> is defined as



Where <digit> is defined in 7.6.1.2.

Where <white space> is defined in 7.4.1.2.

7.7.2.3 Semantic equivalence

A **device** shall process <white space> without semantic interpretation.

A **device** shall interpret all other ISO 7-bit encoded characters as comprising a decimal numeric format generally following any of the NR1, NR2 and NR3 formats of ISO 6093 as allowed by the diagram in 7.7.2.2. The interpreted value shall be subject to the rounding rule as stated in 7.7.2.4.2.

7.7.2.4 Rules

7.7.2.4.1 Range

The mantissa of a <DECIMAL NUMERIC PROGRAM DATA> element shall contain no more than 255 characters excluding any leading zeros.

<DECIMAL NUMERIC PROGRAM DATA> elements shall have an exponent value in the range –32000 through +32000.

If the <DECIMAL NUMERIC PROGRAM DATA> element violates either of these rules, a Command Error shall be generated.

7.7.2.4.2 Numeric element rounding

A **device** may receive a <DECIMAL NUMERIC PROGRAM DATA> element which has greater precision than the **device** can handle internally. In this case, the **device** shall round off the number, not truncate the number, before interpretation. When rounding off, the **device** shall ignore the sign of the number and round up on values greater than or equal to exact half values. Values less than exact half values shall be rounded down.

7.7.2.4.3 Error reporting

Any error conditions reported shall be based on the result of the interpreted value after rounding.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.7.2.4.4 Directives pour les valeurs numériques hors gamme

Si la valeur d'un élément de <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> est en dehors des limites autorisées pour l'en-tête associé, une erreur d'exécution devra être indiquée.

7.7.3 *<DONNÉES DE PROGRAMMATION DE SUFFIXES>*

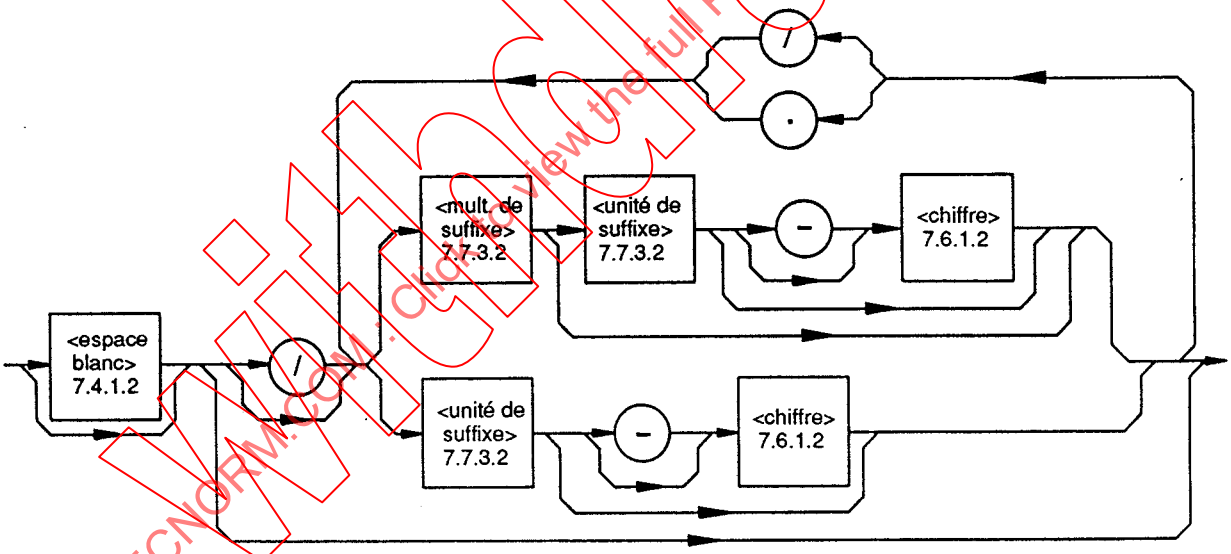
7.7.3.1 Fonction

Un élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES> permet l'utilisation d'un suffixe suivant les <DONNÉES DE PROGRAMMATION NUMÉRIQUES DÉCIMALES> (NRf). Le suffixe exprime les unités associées et les multiplicateurs (facultatifs) qui modifient la façon dont la NRf est interprétée par l'**appareil**.

Comme cela est indiqué dans la syntaxe fonctionnelle à la figure 7-6, la présence d'une <DONNÉE DE PROGRAMMATION DE SUFFIXES> après une NRf est toujours facultative. Dans un **appareil**, aucun <EN-TÊTE D'INSTRUCTIONS DE PROGRAMMATION> ou <EN-TÊTE D'INTERROGATIONS DE PROGRAMMATION> particulier ne doit nécessiter l'utilisation d'un élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES>.

7.7.3.2 Syntaxe de codage

Un élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES> est défini par



Où <chiffre> est défini en 7.6.1.2.

Où <mult. de suffixe>, multiplicateur de suffixe, est défini comme une séquence d'un, deux ou trois caractères <lettre alphabétique majuscule/minuscule>.

Où <unité de suffixe> est définie comme une séquence d'un à quatre caractères <lettre alphabétique majuscule/minuscule>.

<lettre alphabétique majuscule/minuscule> est définie en 7.6.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.2.4.4 Numeric out-of-range guidelines

If a <DECIMAL NUMERIC PROGRAM DATA> element's value is outside the range allowed for the associated header, an Execution Error shall be reported.

7.7.3 *<SUFFIX PROGRAM DATA>*

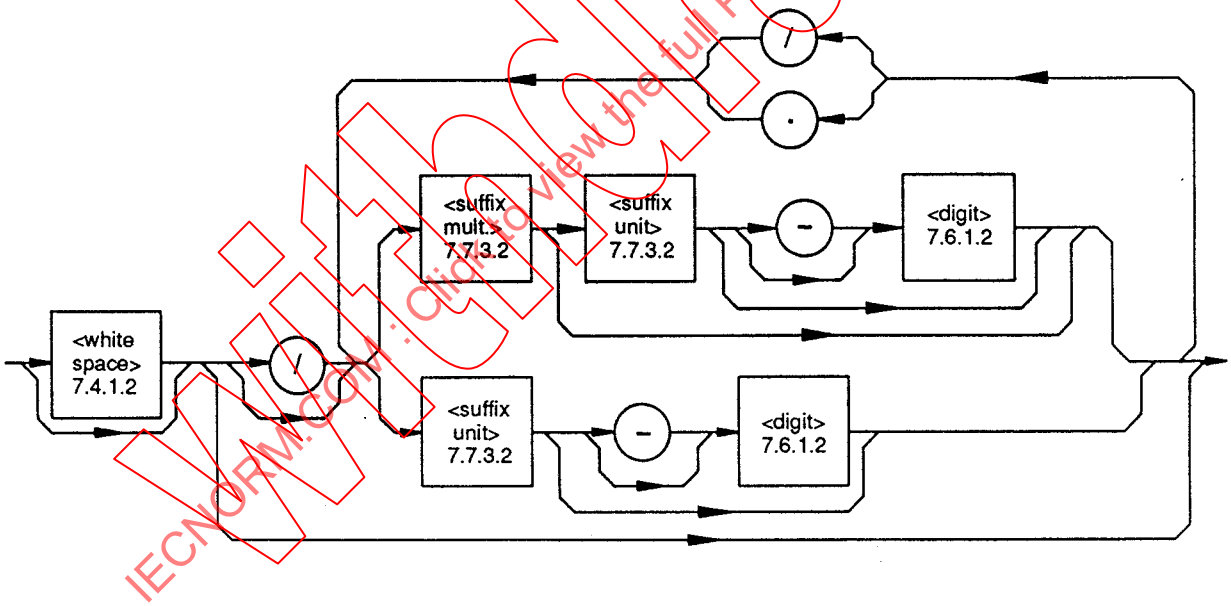
7.7.3.1 Function

A <SUFFIX PROGRAM DATA> element permits the use of a suffix following the <DECIMAL NUMERIC PROGRAM DATA> (NRf). The suffix expresses associated units and (optional) multipliers that modify how the NRf is interpreted by the **device**.

As shown in the functional syntax in figure 7-6, the presence of a <SUFFIX PROGRAM DATA> after an NRf is always optional. No particular <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> in a **device** shall require the use of a <SUFFIX PROGRAM DATA> element.

7.7.3.2 Encoding syntax

A <SUFFIX PROGRAM DATA> element is defined as



Where <digit> is defined in 7.6.1.2.

Where <suffix mult.>, suffix multiplier, is defined as a sequence of one, two, or three <upper/lower-case alpha> characters.

Where <suffix unit> is defined as a sequence of one to four <upper/lower-case alpha> characters.

<upper/lower case-alpha> is defined in 7.6.1.2.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.7.3.3 Equivalence sémantique

Un **appareil** doit traiter un <espace blanc> sans interprétation sémantique.

Un **appareil** doit interpréter tous les caractères alphabétiques avec une équivalence sémantique lettre majuscule/lettre minuscule.

Tous les autres octets codés 7 bits ISO dans cet élément fonctionnel peuvent avoir ou non une équivalence sémantique au choix du concepteur de l'appareil.

La déviation dans l'élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES> devra être interprétée par l'**appareil** comme si <multiplicateur de suffixes> et une <unité de suffixe> implicites avaient été reçus. Ce suffixe implicite est spécifié par le concepteur de l'appareil.

7.7.3.4 Règles

Si un **appareil** utilise des suffixes, il devra faire usage des éléments de <multiplicateur de suffixes> et d'<unité de suffixe> définis dans cette section. Les directives présentées ici pour l'utilisation d'éléments de <DONNÉES DE PROGRAMMATION DE SUFFIXES> ont été définies dans la norme ISO 2955 et ont été étendues pour inclure les unités dérivées du système non international.

N'importe quel élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES> non spécifiquement répertorié dans cette norme devra contenir une valeur mnémonique pour aider l'utilisateur dans la mise en relation de la conversion associée avec le suffixe.

L'arrondi numérique peut être fait avant ou après l'application de l'algorithme de conversion.

Un élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES> ne devra pas contenir plus de 12 caractères, à l'exclusion de l'<espace blanc> en tête.

7.7.3.4.1 Sélection des unités de suffixes

Le tableau 7-1 présente quelques-unes des <unités de suffixes> les plus communément utilisées dans les **appareils** avec plusieurs éléments d'<unités de suffixes> secondaires préférés et autorisés pour une variété d'applications.

Les éléments d'<unités de suffixes> devront utiliser les mnémoniques d'"unités primaires" du tableau 7-1. Cependant, des variantes d'éléments d'<unités de suffixes> sont incluses dans le tableau 7-1 comme "unités secondaires" destinées à un utilisateur fortuit qui tendrait à penser en unités de mesure. Si une unité n'est pas spécifiée, les normes de référence doivent être consultées.

L'utilisation de "E<chiffre>" ou "E-<chiffre>" comme un élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES> ou comme les caractères en tête d'un élément de <DONNÉES DE PROGRAMMATION DE SUFFIXES> est spécifiquement interdite en raison d'une confusion possible avec l'exposant de la précédente NRf.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.3.3 Semantic equivalence

A **device** shall process <white space> without semantic interpretation.

A **device** shall interpret all alpha characters with upper/lower-case semantic equivalence.

All other ISO 7-bit encoded bytes in this functional element may or may not have semantic equivalence at the discretion of the device designer.

The bypass within the <SUFFIX PROGRAM DATA> element shall be interpreted by the **device** as if a default <suffix multiplier> and <suffix unit> had been received. This default suffix is specified by the device designer.

7.7.3.4 Rules

If a **device** uses suffixes, it should make use of the <suffix multiplier> and <suffix unit> elements defined in this clause. Guidelines presented here for the use of <SUFFIX PROGRAM DATA> elements, have been defined in ISO 2955 and have been expanded to include non-SI derived units.

Any <SUFFIX PROGRAM DATA> element not specifically listed in this standard shall contain mnemonic value to assist the user in relating the associated conversion to the suffix.

Numeric rounding may be done before or after the conversion algorithm is applied.

A <SUFFIX PROGRAM DATA> element shall contain no more than 12 characters, excluding the leading <white space>.

7.7.3.4.1 <suffix unit> selection

Table 7-1 presents some of the most common <suffix unit> elements used in **devices** along with several preferred and allowed secondary <suffix unit> elements for a variety of applications.

<suffix unit> elements should use the 'primary unit' mnemonics from table 7-1. Alternate <suffix unit> elements, however, are included in table 7-1 as 'secondary units' for use by the casual user who tends to think in measurement units. If a unit is not specified, the referenced standards should be consulted.

The use of "E<digit>" or "E-<digit>" as a <SUFFIX PROGRAM DATA> element or as the leading characters of a <SUFFIX PROGRAM DATA> element is specifically disallowed because of possible confusion with the exponent of the preceding NRf.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Tableau 7-1 – Eléments d'<unités de suffixes>

Classe	Suffixe préféré (unité primaire)	Suffixe autorisé (unité secondaire)	Unité de référence
Courant	A		Ampère
Pression	ATM		Atmosphère
Charge	C		Coulomb
Eclairage	CD		Candela
Rapport	DB (voir 7.7.3.4.2)		Décibel
Puissance	DBM		Décibel milliwatt
Capacitance	F		Farad
Masse		G	Gramme (le kilogramme est l'unité du SI pour les unités complexes)
Inductance	H		Henry
Fréquence	HZ		Hertz
Pression	INHG		Pouces de mercure
Energie	J		Joule
Température	K		Degré Kelvin
		CEL	Degré Celsius
		FAR	Degré Fahrenheit
Volume	L		Litre
Eclairage	LM		Lumen
Eclairage	LX		Lux
Longueur	M		Mètre
		FT	Pied
		IN	Pouce
Fréquence		MHZ	Mégahertz
Résistance		MOHM	Mégohm
Force	N		Newton
Résistance	OHM		Ohm
Pression	PAL		Pascal
Rapport	PCT		Pour cent
Angle	RAD		Radian
		DEG	Degré
		MNT	Minute (d'arc)
		SEC	Seconde
Temps	S		Seconde
Conductance	SIE		Siemens
Densité magnétique	T		Tesla
Pression	TORR		Torr
Amplitude	V		Volt
Puissance	W		Watt
Flux magnétique	WB		Weber
Flux lumineux	LM		Lumen

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.3.4.2 Référencement de DB

Pour référencer une unité relative (dB) à un niveau absolu, le niveau absolu est rattaché aux DB selon la syntaxe suivante.

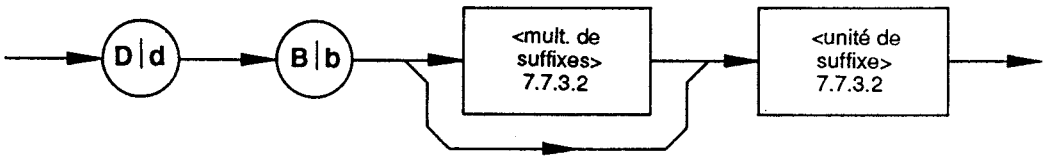
TABLE 7-1 – <suffix unit> elements

Class	Preferred suffix (primary unit)	Allowed suffix (secondary unit)	Referenced unit
Current	A		Ampere
Pressure	ATM		Atmosphere
Charge	C		Coulomb
Illumination	CD		Candela
Ratio	DB (see 7.7.3.4.2)		Decibel
Power	DBM		Decibel milliwatt
Capacitance	F		Farad
Mass		G	Gram (kilogram is the SI unit for compound units)
Inductance	H		Henry
Frequency	HZ		Hertz
Pressure	INHG		Inches of mercury
Energy	J		Joule
Temperature	K		Degree Kelvin
		CEL	Degree Celsius
		FAR	Degree Fahrenheit
Volume	L		Litre
Illumination	LM		Lumen
Illumination	LX		Lux
Length	M		Metre
		FT	Foot
		IN	Inch
Frequency		MHZ	Megahertz
Resistance		MOHM	Megohm
Force	N		Newton
Resistance	OHM		Ohm
Pressure	PAL		Pascal
Ratio	PCT		Percent
Angle	RAD		Radian
		DEG	Degree
		MNT	Minute (of arc)
		SEC	Second
Time	S		Second
Conductance	SIE		Siemens
Mag density	T		Tesla
Pressure	TORR		Torr
Amplitude	V		Volt
Power	W		Watt
Mag flux	WB		Weber
Luminous flux	LM		Lumen

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.3.4.2 DB referencing

To reference a relative unit (dB) to an absolute level, the absolute level is appended to the DB according to the following syntax.



De cette façon, les décibels référencés à 1 milliwatt deviennent DBMW. Pour des raisons historiques, DBM est considéré comme un équivalent de DBMW.

EXEMPLE:

DBUW signifie DB référencés à 1 microwatt

DBUV signifie DB référencés à 1 microvolt

NOTE - Une recherche de pratique courante montre que la forme proposée ici est d'usage courant et ne présente aucune ambiguïté.

7.7.3.4.3 Règles de sélection des <mult. de suffixes>

Le <mult. de suffixes> est utilisé pour modifier la valeur de l'<unité de suffixe>. Le <mult. de suffixes> permet à l'utilisateur d'entrer des valeurs en unités de mesure courantes. Ainsi, dans quelques applications, 1 KHZ est préférable à 1E3 HZ. Le tableau 7-2 dresse la liste des <mult. de suffixes> autorisés.

NOTE - Seuls les multiplicateurs d'unités techniques sont autorisés.

Tableau 7-2 – Mnémoniques <multiplicateurs de suffixes> autorisés

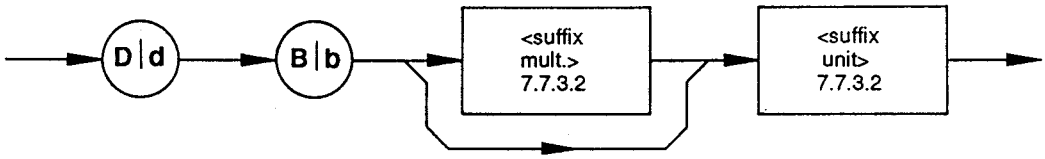
Définition	Mnémonique
1E18	EX
1E15	PE
1E12	T
1E9	G
1E6	MA (voir note)
1E3	K
1E-3	M (voir note)
1E-6	U
1E-9	N
1E-12	P
1E-15	F
1E-18	A

NOTE - Les unités de suffixes, MHZ et MOHM, sont des cas particuliers qu'il ne faut pas confondre avec <mult. de suffixes> HZ ou <mult. de suffixes> OHM. Ces cas particuliers d'<unités de suffixes> sont inclus dans le tableau 7-1 en raison de la pratique courante dans l'industrie.

7.7.3.4.4 Utilisation d'unités fabriquées à partir de combinaisons de règles SI

Quand un besoin d'unités complexes apparaît (c'est-à-dire newton-mètre ou mètre par seconde), les règles suivantes doivent être appliquées. Quand l'unité complexe est le quotient de deux unités, le caractère "/" doit alors être utilisé pour séparer les unités. Donc, M/S est la représentation autorisée pour mètre/seconde.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



Thus, decibels referenced to 1 milliwatt become DBMW. For historical reasons, DBM is allowed as an alias for DBMW.

EXAMPLE:

DBUW means DB referenced to 1 microwatt

DBUV means DB referenced to 1 microvolt

NOTE - A search of common practice shows that the form proposed here is the common usage and does not introduce ambiguity.

7.7.3.4.3 <suffix mult.> selection rules

The <suffix mult.> is used to modify the value of the <suffix unit>. The <suffix mult.> allows the user to enter values in common measurement units. Thus in some applications 1 KHZ is preferable to 1E3 HZ. Table 7-2 lists the allowed <suffix mult.>'s.

NOTE - Only engineering unit multipliers are allowed.

TABLE 7-2 – Allowed <suffix mult.> mnemonics

Definition	Mnemonic
1E18	EX
1E15	PE
1E12	T
1E9	G
1E6	MA (see note)
1E3	K
1E-3	M (see note)
1E-6	U
1E-9	N
1E-12	P
1E-15	F
1E-18	A

NOTE - The suffix units, MHZ and MOHM, are special cases which should not be confused with <suffix mult.>HZ or <suffix mult.>OHM. These special case <suffix units> are included in table 7-1 because of accepted industry practice.

7.7.3.4.4 Use of units made up of combinations of SI rules

When a need arises for complex units (i.e. newton-meter or meter per second), the following rules should apply. When the complex unit is the quotient of two units, then the character "/" should be used to separate the units. Thus M/S is the allowed representation for metre/second.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Quand l'unité complexe est le produit de deux unités, le caractère "." doit être utilisé pour séparer les unités. Donc, N.M est la représentation autorisée pour newton-mètre.

Les exposants suffixes sont représentés par un <chiffre>. Ils sont utilisés pour des unités qui apparaissent plus d'une fois dans une unité complexe. Lorsqu'un exposant suffixe est utilisé, il doit apparaître comme un chiffre et être rattaché à l'unité sans espace. Un exemple: l'unité SI d'accélération, mètre par seconde carré, s'écrit M/S2.

7.7.4 *<DONNÉES DE PROGRAMMATION NUMÉRIQUES NON DÉCIMALES>*

7.7.4.1 Fonction

Les <DONNÉES de PROGRAMMATION NUMÉRIQUES NON DÉCIMALES> permettent le passage d'informations numériques dans des bases autres que la base 10. Ces formats facilitent l'interprétation directe par les utilisateurs des systèmes et les programmes d'applications pour les applications où les nombres non décimaux ont une signification directe. L'utilisation de ce format peut simplifier les programmes de conversion dans les logiciels d'applications.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

When the complex unit is the product of two units, the character "." should be used to separate the units. Thus N.M is the allowed representation for newton-metre.

Suffix exponents are represented by a <digit>. They are used for units which appear more than once in a complex unit. When a suffix exponent is used, it should appear as a <digit> and be appended to the unit without a space. An example is the SI unit of acceleration, metres per second squared, written as M/S2.

7.7.4 *<NON-DECIMAL NUMERIC PROGRAM DATA>*

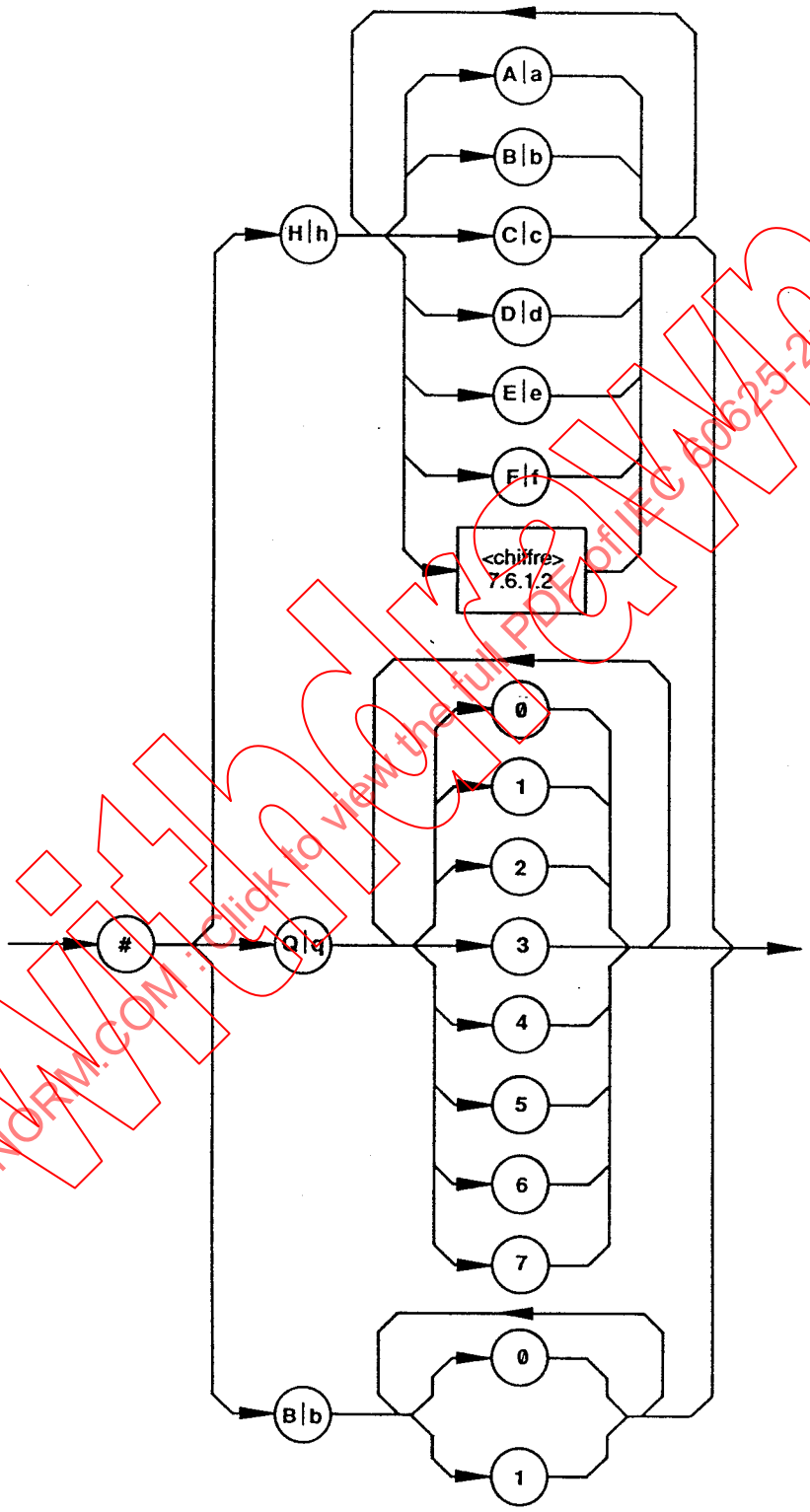
7.7.4.1 Function

The <NON-DECIMAL NUMERIC PROGRAM DATA> element allows passing numeric information in bases other than 10. These formats facilitate direct interpretation by system users and application programs for those applications where non-decimal numbers have direct significance. The use of this format may simplify conversion routines in application software.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.7.4.2 Syntaxe de codage

Un élément de <DONNÉES DE PROGRAMMATION NUMÉRIQUES NON DÉCIMALES> est défini par

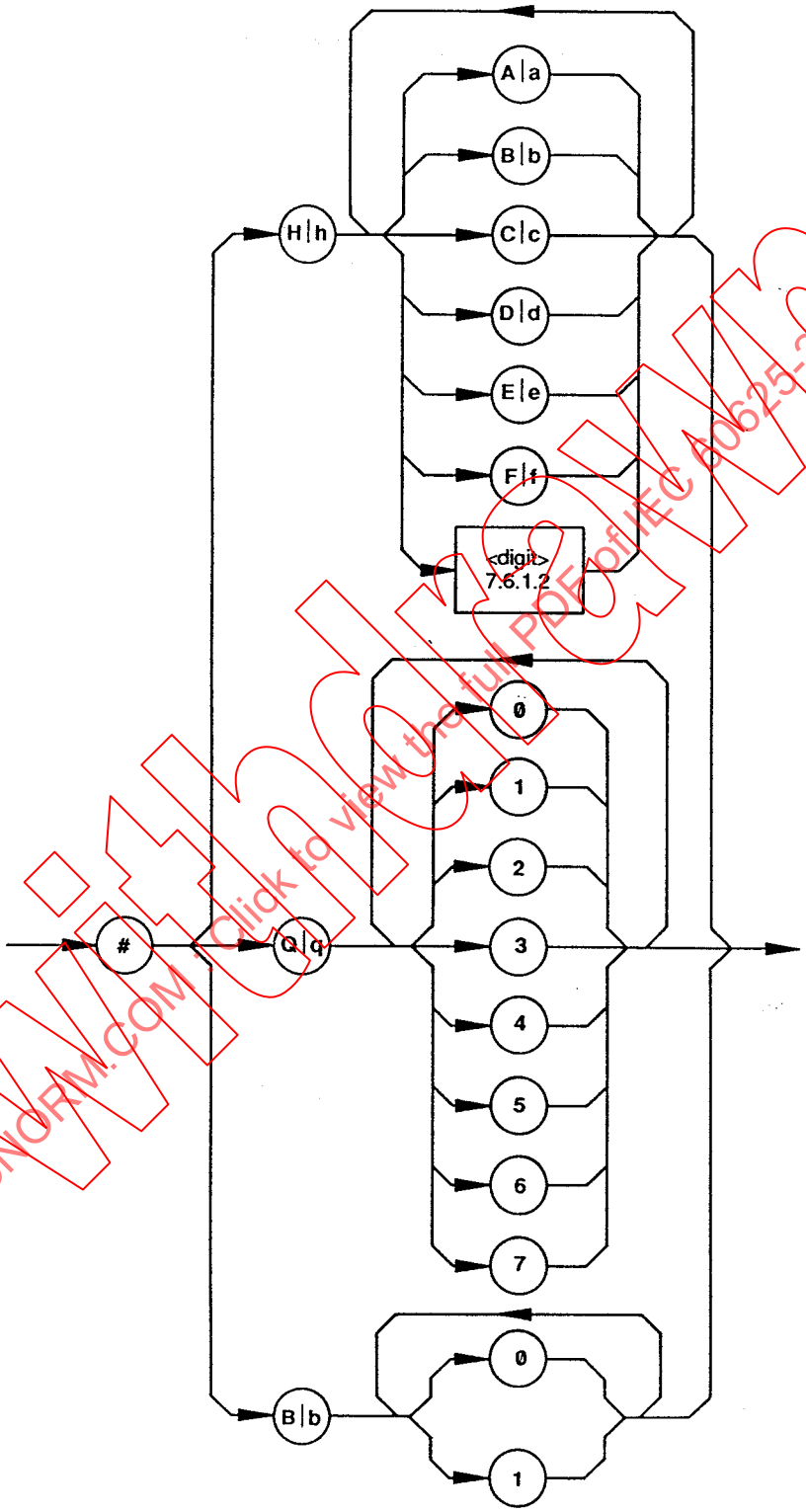


Où <chiffre> est défini en 7.6.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.4.2 Encoding syntax

A <NON-DECIMAL NUMERIC PROGRAM DATA> element is defined as



Where <digit> is defined in 7.6.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.7.4.3 Equivalence sémantique

Un **appareil** doit interpréter n'importe lequel et l'ensemble des trois formats numériques non décimaux spécifiés comme ayant des valeurs numériques sémantiquement équivalentes.

Un **appareil** doit interpréter <lettre alphabétique majuscule/minuscule> sans attacher de sens sémantique à la casse (majuscules/minuscules).

7.7.4.4 Règles

7.7.4.4.1 Codage hexadécimal

Les caractères suivant un préambule #H ou #h doivent être interprétés comme un nombre hexadécimal sans signe à emplacement de virgule implicite.

La base pour les nombres hexadécimaux est 16. Les valeurs décimales et les représentations codées de 7 bits relatives sont les suivantes:

Valeur décimale:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Code de 7 bits ISO:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Variante du Code de 7 bits ISO:	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

Les caractères non hexadécimaux doivent générer une erreur d'instruction.

7.7.4.4.2 Codage octal

Les caractères suivant un préambule #Q ou #q doivent être interprétés comme un nombre octal sans signe à emplacement de virgule implicite.

La base pour les nombres octaux est 8. Les valeurs décimales et les représentations codées ISO relatives sont les suivantes:

Valeur décimale:	0	1	2	3	4	5	6	7
Code de 7bits ISO:	0	1	2	3	4	5	6	7

Les caractères non octaux doivent générer une erreur d'instruction.

7.7.4.4.3 Codage binaire

Les caractères suivant un préambule #B ou #b doivent être interprétés comme un nombre binaire sans signe à emplacement de virgule implicite.

La base pour les nombres binaires est 2. Les valeurs décimales et les représentations codées de 7 bits relatives sont les suivantes:

Valeur décimale:	0	1
Code de 7 bits ISO:	0	1

Les caractères non binaires doivent générer une erreur d'instruction.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7.7.4.3 Semantic equivalence

A **device** shall interpret any and all of the three specified non-decimal numeric formats as semantically equivalent numeric values.

A **device** shall interpret <upper/lower-case alpha> without attaching semantic meaning to the case of the alpha characters.

7.7.4.4 Rules

7.7.4.4.1 Hexadecimal encoding

The characters following a #H or #h preamble shall be interpreted as an unsigned implicit radix point hexadecimal number.

The radix for hexadecimal numbers is 16. The decimal values and related ISO 7-bit code representations are as follows:

Decimal value:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ISO 7-bit code:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Alternate ISO 7-bit Code:	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

Non-hex characters shall generate a Command Error.

7.7.4.4.2 Octal encoding

The characters following a #Q or #q preamble shall be interpreted as an unsigned implicit radix point octal number.

The radix for octal numbers is 8. The decimal values and related ISO 7-bit code representations are as follows:

Decimal value:	0	1	2	3	4	5	6	7
ISO 7-bit code:	0	1	2	3	4	5	6	7

Non-octal characters shall generate a Command Error.

7.7.4.4.3 Binary encoding

The characters following a #B or #b preamble shall be interpreted as an unsigned implicit radix point binary number.

The radix for binary numbers is 2. The decimal values and related ISO 7-bit code representations are as follows:

Decimal value:	0	1
ISO 7-bit code:	0	1

Non-binary characters shall generate a Command Error.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

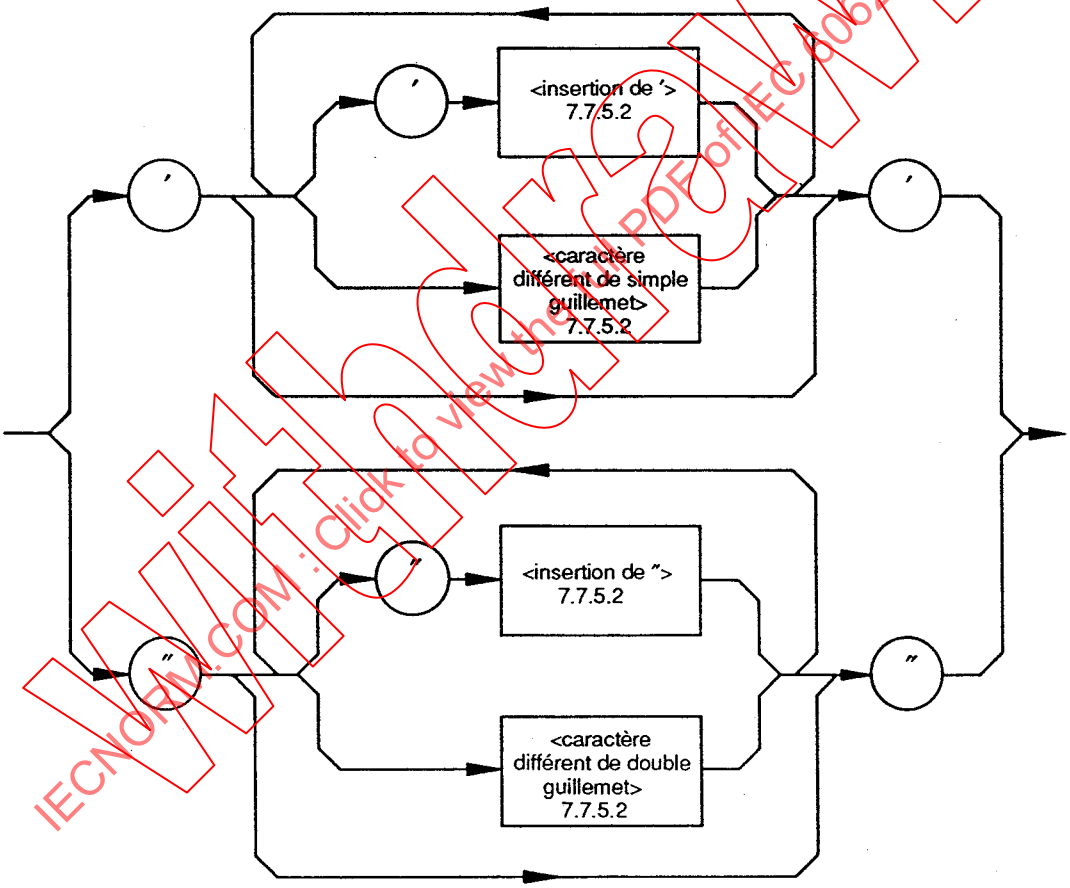
7.7.5 *<DONNÉES DE PROGRAMMATION DE CHAÎNES>*

7.7.5.1 Fonction

L'élément de <DONNÉES DE PROGRAMMATION DE CHAÎNES> permet à n'importe quel caractère dans le code de 7 bits ISO (y compris caractères non imprimables) d'être transmis comme un message. Ce champ de données est particulièrement utile lorsque le texte doit être affiché (par exemple, sur une imprimante ou sur un appareil type CRT). Les <DONNÉES DE PROGRAMMATION DE CHAÎNES> permettent l'utilisation de fichiers-formats, tels que retour de chariot, interligne ou espace, pour formater correctement le texte.

7.7.5.2 Syntaxe de codage

Un élément de <DONNÉES DE PROGRAMMATION DE CHAÎNES> est défini par



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Où <insertion de '> est défini comme un caractère ISO simple ayant la valeur 27 (39 en décimal).

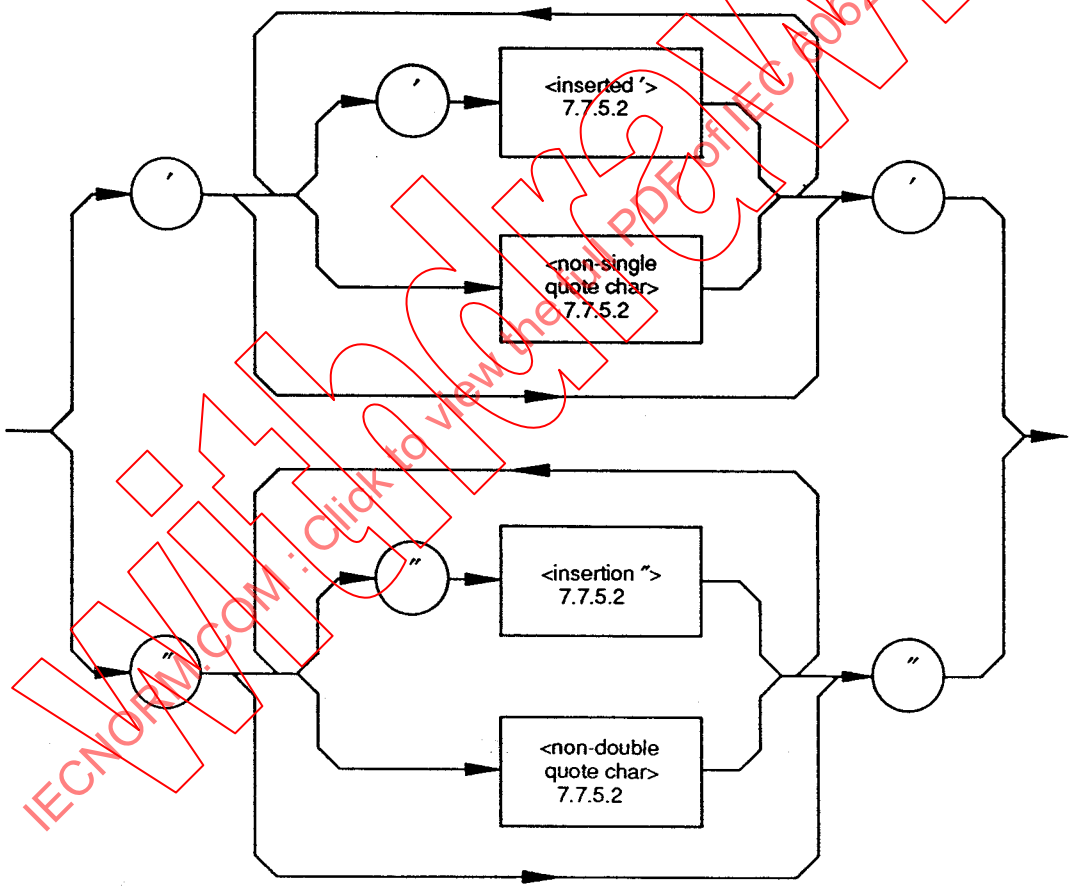
7.7.5 *<STRING PROGRAM DATA>*

7.7.5.1 Function

The <STRING PROGRAM DATA> element allows any character in the ISO 7-bit code (including non-printable characters) to be transmitted as a message. This data field is particularly useful where text is to be displayed (for example, on a printer or CRT type device). The <STRING PROGRAM DATA> permits the use of format effectors, such as carriage return, newline, or space, to format text correctly.

7.7.5.2 Encoding syntax

A <STRING PROGRAM DATA> element is defined as



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Where <inserted '> is defined as a single ISO 7-bit character with the value 27 (39 decimal).

Où <caractère différent de simple guillemet> est défini comme un caractère ISO simple ayant n'importe quelle valeur sauf 27 (39 en décimal).

Où <insertion de "> est défini comme un caractère ISO simple ayant la valeur 22 (34 en décimal).

Où <caractère différent de double guillemet> est défini comme un caractère ISO simple ayant n'importe quelle valeur sauf 22 (34 en décimal).

7.7.5.3 Equivalence sémantique

Un **appareil** doit recevoir les données de chaînes englobées dans n'importe lequel des deux formats à chaînes délimitées (c'est-à-dire l'utilisation de simple ou double guillemet) et interpréter les données de chaînes comme sémantiquement équivalentes.

7.7.5.4 Règles

7.7.5.4.1 Délimitation entre simples guillements

Un simple guillemet (') directement suivi par une <insertion de'> est utilisé pour représenter l'envoi d'un seul ' à l'intérieur d'une <DONNÉE DE PROGRAMMATION DE CHAÎNES> délimitée par de simples guillemets.

7.7.5.4.2 Délimitation entre doubles guillemets

Un double guillemet (") directement suivi par une <insertion de "> est utilisé pour représenter l'envoi d'un seul " à l'intérieur d'une <DONNÉE DE PROGRAMMATION DE CHAÎNES> délimitée par des doubles guillemets.

7.7.6 *<DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE>*

7.7.6.1 Fonction

L'élément de <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> permet à n'importe quel octet (y compris les codes de 7 bits ISO étendus) d'être transmis dans un message. Cet élément est particulièrement utile pour envoyer de grandes quantités de données.

Cet élément représente une solution générale à la transmission de données binaires de 8 bits. L'écriture binaire spécifique est décrite en 9.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Where <non-single quote char> is defined as a single ISO 7-bit character of any value except 27 (39 decimal).

Where <inserted "> is defined as a single ISO 7-bit character with the value 22 (34 decimal).

Where <non-double quote char> is defined as a single ISO 7-bit character of any value except 22 (34 decimal).

7.7.5.3 Semantic equivalence

A **device** shall interpret string data encapsulated in either of the two specified delimited-string formats (that is, the use of double quotes or single quotes) as semantically equivalent.

7.7.5.4 Rules

7.7.5.4.1 Single quote delimiting

A single quote (') followed directly by an <inserted '> is used to represent a single ' within a <STRING PROGRAM DATA> delimited by single quotes.

7.7.5.4.2 Double quote delimiting

A double quote (") followed directly by an <inserted"> is used to represent a single" within a <STRING PROGRAM DATA> delimited by double quotes.

7.7.6 *<ARBITRARY BLOCK PROGRAM DATA>*

7.7.6.1 Function

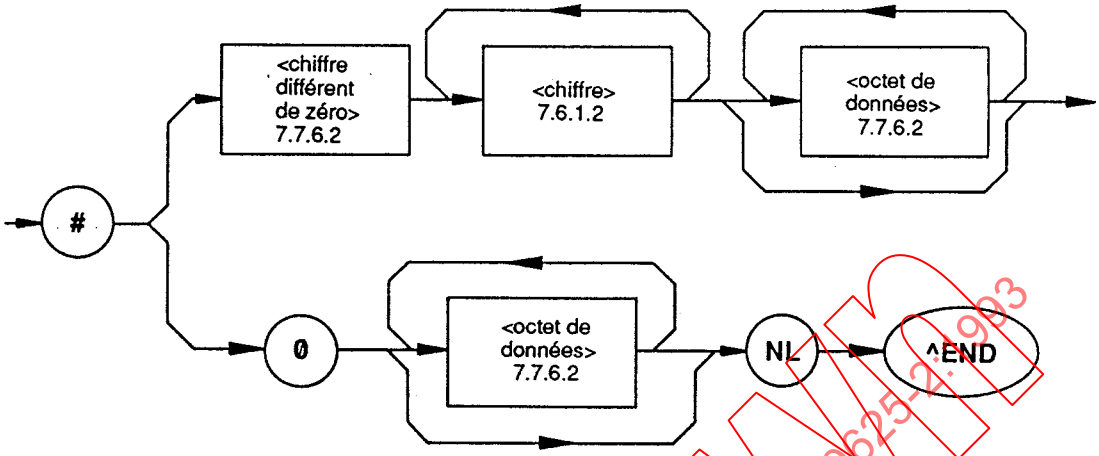
The <ARBITRARY BLOCK PROGRAM DATA> element allows any 8-bit bytes (including extended ISO 7-bit codes) to be transmitted in a message. This element is particularly useful for sending large quantities of data.

This element represents a general purpose solution to the transmission of 8-bit binary data. Specific binary coding is described in 9.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.6.2 Syntaxe de codage

Un élément de <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> est défini par



Où <chiffre> est défini en 7.6.1.2.

Où <chiffre différent de zéro> est défini comme un simple octet codé ISO dans la gamme 31-39 (49-57 en décimal).

Où <octet> est défini comme un octet dans la gamme 00 – FF (0 – 255 décimal).

NOTE - L'utilisation d'un format indéfini (#0) nécessite NL^END et impose un arrêt immédiat du <MESSAGE DE PROGRAMMATION>.

7.7.6.3 Equivalence sémantique

Un **appareil** doit interpréter <octets de données> dans n'importe lequel des deux formats spécifiés (c'est-à-dire avec #<chiffre différent de zéro> ou #0 avec l'arrêt du message NL^END) comme sémantiquement équivalents.

7.7.6.4 Règles

La valeur de l'élément <chiffre différent de zéro> sera égale au nombre d'éléments <chiffre> qui suivent. La valeur des éléments <chiffre> pris ensemble comme un nombre entier décimal sera égale au nombre d'éléments <octet de données> qui suivent. Si le message CEI 625-1 END est reçu avant que le nombre d'octets spécifié ait été reçu, une erreur d'instruction sera signalée.

7.7.6.5 Notes et exemples

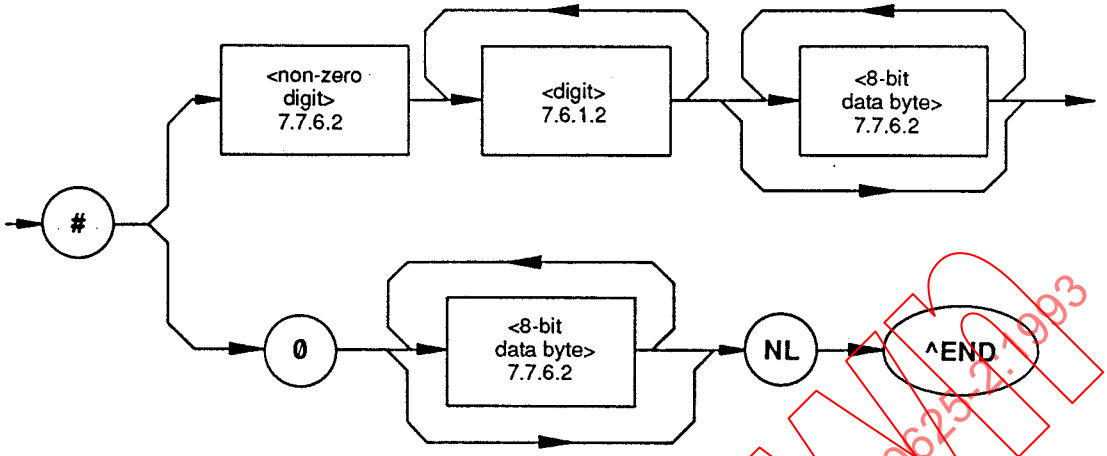
Il est recommandé aux concepteurs de considérer les implications de la taille et de la disponibilité des mémoires tampons, particulièrement dans les récepteurs. Le concepteur ne devrait pas supposer que tous les appareils seront capables d'accepter des messages longs. Voir 4.9 pour les exigences de spécifications de l'**appareil**.

Le format indéfini est utile lorsque la longueur de la transmission n'est pas connue ou lorsque la vitesse de transmission ou d'autres considérations empêchent le découpage en blocs d'une longueur définie.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7.7.6.2 Encoding syntax

An <ARBITRARY BLOCK PROGRAM DATA> element is defined as



Where <digit> is defined in 7.6.1.2.

Where <non-zero digit> is defined as a single ISO 7-bit encoded byte in the range 31-39 (49-57 decimal).

Where <8-bit byte> is defined as an 8-bit byte in the range 00 – FF (0 – 255 decimal).

NOTE - The use of the indefinite format (#0) requires NL^END and forces immediate termination of the <PROGRAM MESSAGE>.

7.7.6.3 Semantic equivalence

A **device** shall interpret <8-bit data bytes> in either of the two specified formats (that is, with #<non-zero digit> or #0 with the NL^END message termination) as semantically equivalent.

7.7.6.4 Rules

The value of the <non-zero digit> element shall equal the number of <digit> elements which follow. The value of the <digit> elements taken together as a decimal integer shall equal the number of <8-bit data byte> elements which follow. If the IEC 625-1 END message is received before the specified number of bytes has been received, a Command Error shall be reported.

7.7.6.5 Notes and examples

Designers should consider the implications of buffer size and availability, particularly in the intended receiving devices. The designer should not assume that all devices will be capable of accepting lengthy messages. See 4.9 for **device** specification requirements.

The indefinite format is useful when the length of the transmission is not known or when transmission speed or other considerations prevents segmentation into definite length blocks.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Par exemple, quatre octets de données (DAB's) peuvent être envoyés en utilisant:

1) #14<DAB><DAB><DAB><DAB>
2) #3004<DAB><DAB><DAB><DAB>
3) #0<DAB><DAB><DAB><DAB> NL^END.

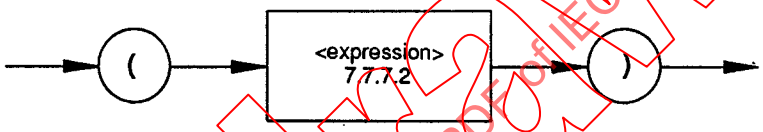
7.7.7 *<DONNÉES DE PROGRAMMATION D'EXPRESSION>*

7.7.7.1 Fonction

L'élément de <DONNÉES DE PROGRAMMATION D'EXPRESSION> est un élément qui donne un scalaire, un vecteur, une matrice ou une valeur de chaînes. Il permet la manipulation des paramètres par l'**appareil**.

7.7.7.2 Syntaxe de codage

Un élément de <DONNÉES DE PROGRAMMATION D'EXPRESSION> est défini par



Où <expression> est définie comme une séquence:

– d'octets de données codés ISO dans la gamme 20 à 7E (32 à 126 en décimal), à l'exception des doubles guillemets, du signe "fagot", des simples guillemets, de la parenthèse gauche, de la parenthèse droite, et du point-virgule (caractères 22, 23, 27, 28, 29, et 3B en hexadécimal ou 34, 35, 39, 40, 41 et 59 en décimal respectivement), et/ou

– un ensemble défini par l'appareil d'éléments de <DONNÉES DE PROGRAMMATION> décrits à la section 7.7, à l'exception de la forme indéfinie des <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE>. Ceci permet la représentation des expressions emboîtées.

7.7.7.3 Équivalence sémantique

Un **appareil** doit interpréter <lettre alphabétique majuscule/minuscule> contenue dans un élément d'<expression> sans attacher de sens sémantique à la casse (majuscule/minuscule) des caractères alphabétiques.

Les octets codés ISO contenus dans un élément d'<expression> peuvent avoir d'autres règles ou conditions d'équivalence sémantique au choix du concepteur de l'appareil.

7.7.7.4 Règles

Les sémantiques d'<expression> sont entièrement dépendantes de l'appareil, mais l'**appareil** doit être capable de transformer l'<expression> en l'équivalent d'un ou plusieurs éléments de <DONNÉES DE PROGRAMMATION> qui ne sont pas des éléments d'<expression>.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

For example, four data bytes (DAB's) may be sent using:

1) #14<DAB><DAB><DAB><DAB>
2) #3004<DAB><DAB><DAB><DAB>
3) #0<DAB><DAB><DAB><DAB> NL^END.

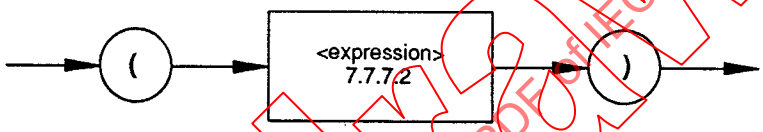
7.7.7 *<EXPRESSION PROGRAM DATA>*

7.7.7.1 Function

The <EXPRESSION PROGRAM DATA> element evaluates to a scalar, vector, matrix, or string value. It allows parameters to be manipulated by the **device**.

7.7.7.2 Encoding syntax

An <EXPRESSION PROGRAM DATA> element is defined as



Where <expression> is defined as a sequence:

– of ISO 7-bit encoded data bytes in the range 20 to 7E (32 to 126 decimal) except the double quote, number sign, single quote, left parenthesis, right parenthesis, and semicolon (characters 22, 23, 27, 28, 29, and 3B hexadecimal or 34, 35, 39, 40, 41, and 59 decimal respectively), and/or

– a device-defined set of the <PROGRAM DATA> elements described in clause 7.7 except the indefinite form of the <ARBITRARY BLOCK PROGRAM DATA>. This allows the representation of nested expressions.

7.7.7.3 Semantic equivalence

A **device** shall interpret <upper/lower-case alpha> contained in the <expression> element without attaching semantic meaning to the case of the alpha characters.

ISO 7-bit encoded bytes contained in the <expression> element may have other rules or conditions for semantic equivalence at the discretion of the device designer.

7.7.7.4 Rules

The <expression> semantics are completely device-dependent, but the **device** shall be able to transform the <expression> into the equivalent of one or more <PROGRAM DATA> elements that are not <expression> elements.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Les **appareils** peuvent autoriser des sous-expressions correctement emboîtées à l'intérieur d'une <expression> jusqu'à une profondeur d'emboîtement maximum définie par l'appareil. Une sous-expression est une <expression> syntaxiquement correcte qui est entièrement renfermée à l'intérieur d'une <expression>.

La documentation de l'appareil doit clairement indiquer quels sont les éléments de <DONNÉES DE PROGRAMMATION> qui peuvent apparaître à l'intérieur d'une <expression> de même que la profondeur d'emboîtement de sous-expression maximale. La documentation de l'**appareil** doit clairement indiquer toute restriction de syntaxe supplémentaire que l'**appareil** peut apporter à l'<expression>.

Les caractères énumérés en 7.7.7.2 (doubles guillemets, signe "fagot", simples guillemets, parenthèses gauche et droite, et point-virgule) peuvent ne pas apparaître dans le corps de l'expression, mais peuvent apparaître dans un élément de <DONNÉE DE PROGRAMMATION DE CHAÎNE> ou de <DONNÉE DE PROGRAMMATION DE BLOC ARBITRAIRE> dans une <expression> ou une sous-expression.

8 Formats de l'appareil parleur

8.1 *Vue d'ensemble*

Cet article traite du formatage des éléments de <MESSAGES DE RÉPONSE> envoyés d'un **appareil** par l'intermédiaire de son **système d'interface**. Comme à la section 7, le formatage se produit sur deux niveaux, le fonctionnel et le codage. Le matériel d'ensemble pour l'article 7 s'applique ici sauf que les codes sont générés par l'**appareil** et non humainement, et incorporent donc une somme minimum de flexibilité dans la syntaxe de codage. Pour la lisibilité et la compatibilité humaines, la syntaxe de codage est, dans la plupart des cas, obtenue à partir d'un sous-ensemble de la syntaxe contrôleur à appareil.

Des besoins particuliers peuvent nécessiter qu'un **appareil** parle directement, c'est-à-dire pas par l'intermédiaire d'un contrôleur, à un appareil ou à une catégorie d'appareils qui ne comprennent pas la syntaxe de <MESSAGE DE RÉPONSE> CEI 625-2. Par exemple, un **appareil** peut être capable de s'étalonner en donnant l'ordre à une source de modèle particulière de fournir une stimulation. Un exemple plus courant pourrait être un **appareil** qui est capable d'utiliser le bus pour donner l'ordre à une imprimante ou à un traceur d'imprimer ou de représenter sous forme de graphique des données enregistrées. Toutes les déviations par rapport à la CEI 625-2 pour obtenir un transfert de message d'un appareil à l'autre, comme cela est décrit ci-dessus, seront minimisées et explicitement documentées.

8.2 *Notation*

La notation du diagramme de syntaxe suit le même format que décrit en 7.2.

8.3 *Messages de réponse terminés – Syntaxe fonctionnelle*

8.3.1 *Fonction*

Un <MESSAGE DE RÉPONSE> terminé est un message "appareil à contrôleur". Il contient de manière typique des résultats de mesures, des réglages, ou des informations d'état.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Devices may allow properly nested subexpressions within an <expression> up to a device-defined maximum nesting depth. A subexpression is a syntactically correct <expression> which is completely contained within an <expression>.

The device documentation shall clearly indicate which <PROGRAM DATA> elements may appear within an <expression> as well as the maximum subexpression nesting depth. The **device** documentation shall clearly indicate any additional syntax restrictions which the **device** may place on the <expression>.

The characters listed in 7.7.7.2 (double quote, number sign, single quote, left and right parentheses, and semicolon) may not appear within the body of the <expression> but may appear within a <STRING PROGRAM DATA> or <ARBITRARY BLOCK PROGRAM DATA> element in an <expression> or subexpression.

8 Device talking formats

8.1 *Overview*

This clause discusses the formatting of <RESPONSE MESSAGE> elements sent from a **device** via its **system interface**. As in clause 7, the formatting occurs on two levels, the functional and the encoding. The overview material for clause 7 applies here except that the codes are **device**, and not human, generated and thus embody a minimum amount of flexibility in the encoding syntax. For human readability and consistency, the encoding syntax is, in most cases, obtained from a subset of the controller-to-device syntax.

Particular needs may require a **device** to talk directly, i.e. not through a controller, to a device or class of devices which do not understand the IEC 625-2 <RESPONSE MESSAGE> syntax. For example, a **device** may be able to calibrate itself by commanding a particular model source to supply a stimulus. A more common example might be a **device** which is able to use the bus to command a printer or plotter to print or graph internally stored data. All deviations from IEC 625-2 to achieve device-to-device message transfer, as described above, shall be minimized and shall be explicitly documented.

8.2 *Notation*

Syntax diagram notation follows the same format as is described in 7.2.

8.3 *Terminated response messages – Functional syntax*

8.3.1 *Function*

A terminated <RESPONSE MESSAGE> is a 'device-to-controller' message. It typically contains measurement results, settings, or status information.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Un <MESSAGE DE RÉPONSE> est interprété par un contrôleur exécutant un programme d'application et, en tant que tel, nécessite d'acheminer ses informations précisément pour un fonctionnement compatible avec une large gamme de contrôleurs. Un <MESSAGE DE RÉPONSE>, par conséquent, a un format plus restrictif qu'un <MESSAGE DE PROGRAMMATION>, voir l'article 7.

8.3.2 *Syntaxe*

La syntaxe fonctionnelle de la réponse de l'appareil doit être générée en suivant le groupe des quatre diagrammes dans les figures 8-1 à 8-4. L'utilisation de l'élément de <DONNÉE DE RÉPONSE DE 7 BITS ARBITRAIRE>, voir 8.7.11, ou bien de l'élément de <DONNÉE DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE>, voir 8.7.10, impose un arrêt immédiat du <MESSAGE DE RÉPONSE>.

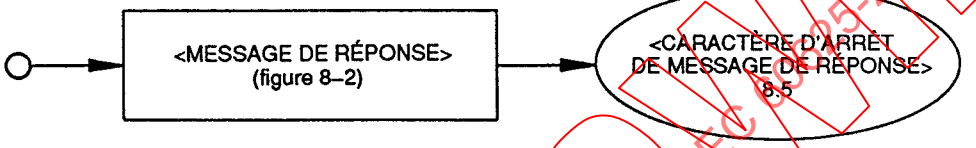


Figure 8-1 – Syntaxe d'élément fonctionnel de <MESSAGE DE RÉPONSE TERMINÉ>

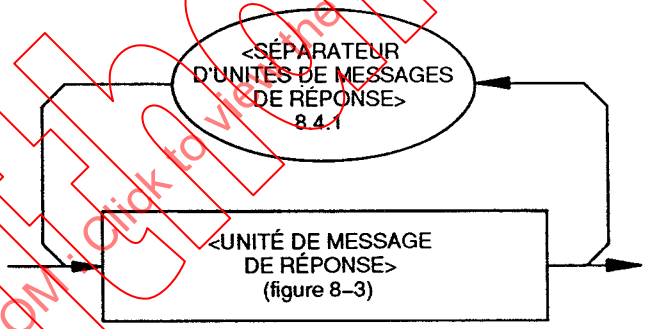


Figure 8-2 – Syntaxe d'élément fonctionnel de <MESSAGE DE RÉPONSE>

L'<UNITÉ DE MESSAGE DE RÉPONSE> a deux syntaxes basiques. La première est une version "précise" de la syntaxe d'<UNITÉ DE MESSAGE DE PROGRAMMATION>, et est généralement utilisée pour les informations d'ajustement de retour. La seconde est "sans en-tête" et est typiquement utilisée pour retourner les résultats de mesures avec un minimum de temps de système de bus.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

A <RESPONSE MESSAGE> is interpreted by a controller running an application program, and as such, needs to convey its information precisely for consistent operation with a wide range of controllers. A <RESPONSE MESSAGE>, therefore, has a more restrictive format than a <PROGRAM MESSAGE>, see clause 7.

8.3.2 *Syntax*

The functional syntax of the device's response shall be generated by following the four-diagram set in figures 8-1 to 8-4. The use of either the <ARBITRARY ISO 7-bit RESPONSE DATA> element, see 8.7.11, or the <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element, see 8.7.10, forces an immediate termination of the <RESPONSE MESSAGE>.

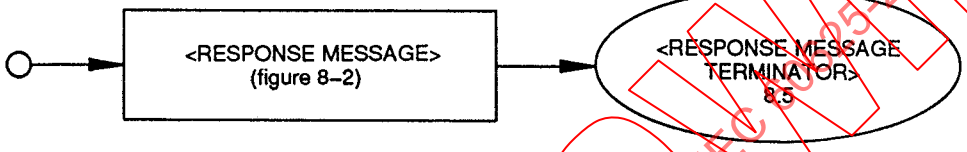


Figure 8-1 – <TERMINATED RESPONSE MESSAGE> functional element syntax

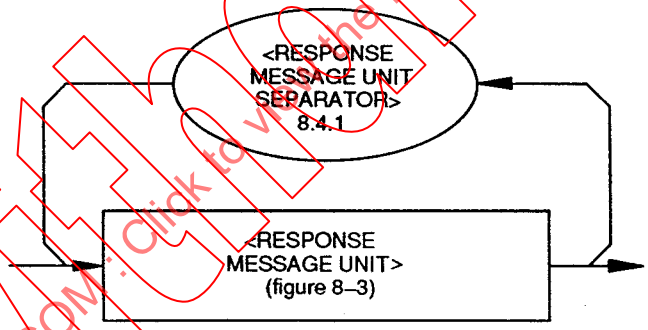


Figure 8-2 – <RESPONSE MESSAGE> functional element syntax

The <RESPONSE MESSAGE UNIT> has two basic syntaxes. The first is a "precise" version of the <PROGRAM MESSAGE UNIT> syntax, and is generally used for returning setting information. The second is "headerless" and is typically used to return measurement results with a minimum of bus overhead.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

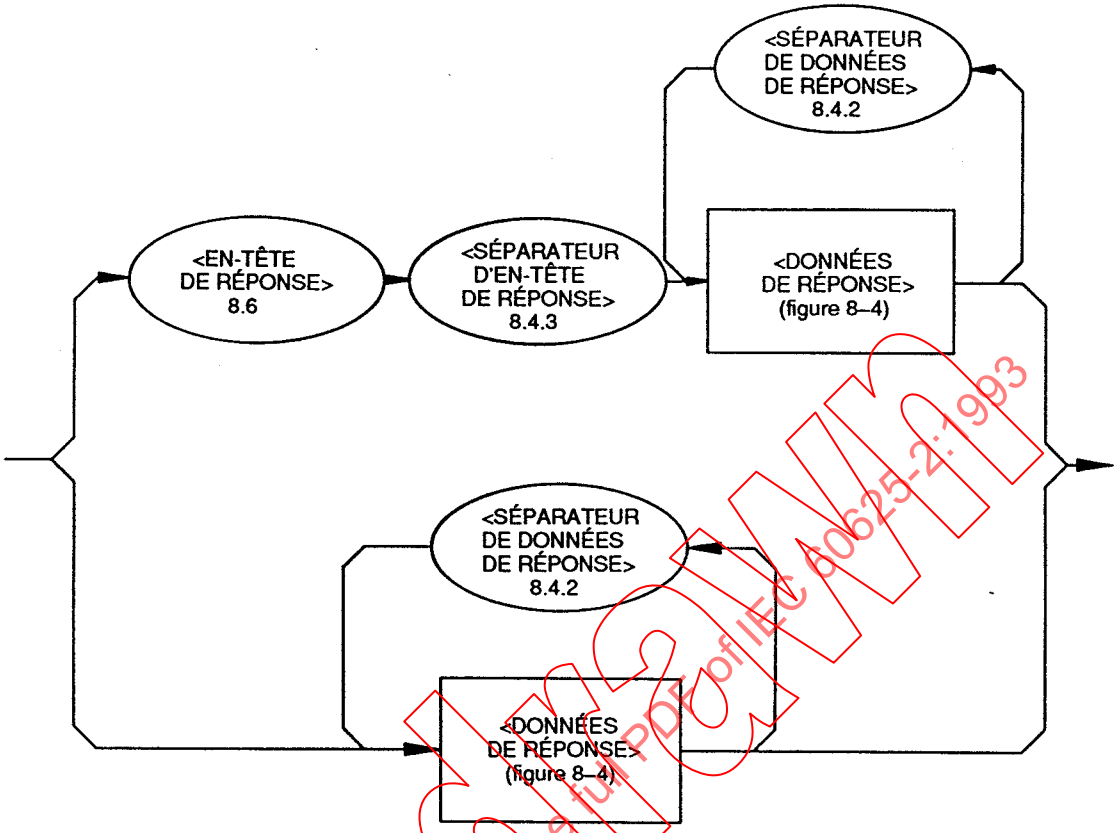


Figure 8-3 – Syntaxe d'élément fonctionnel d'<UNITÉ DE MESSAGE DE RÉPONSE>

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

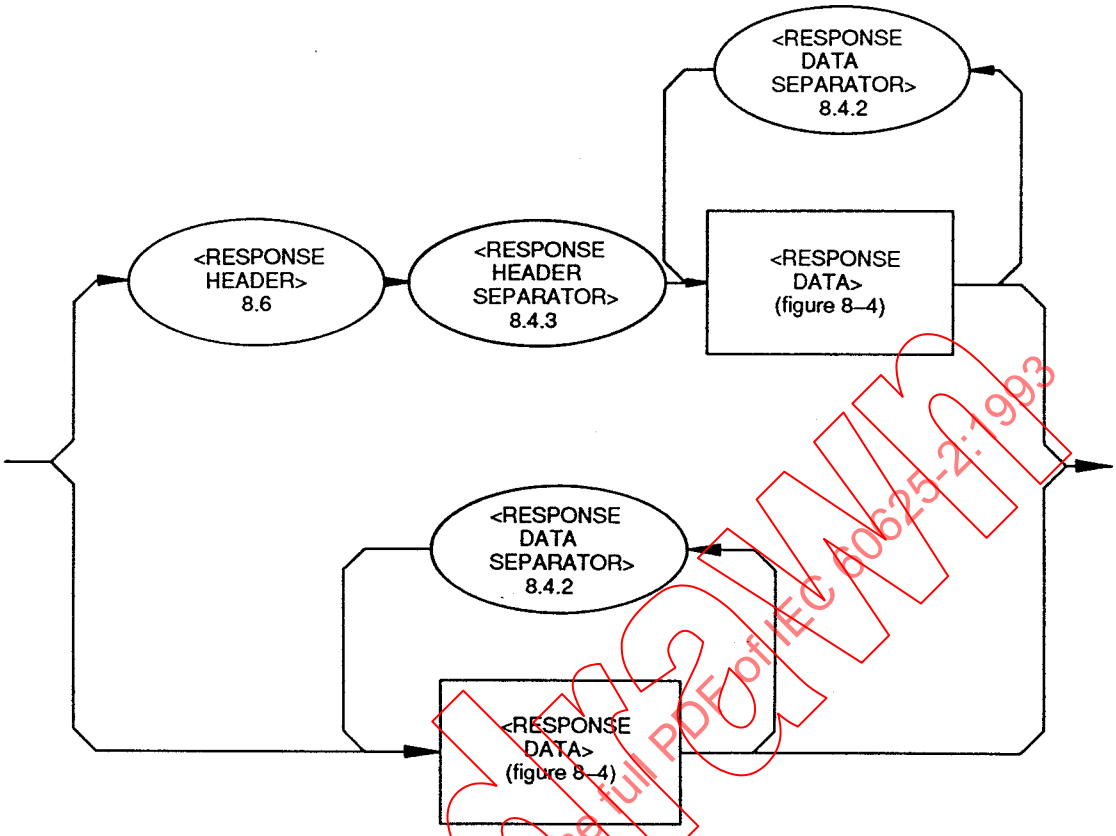
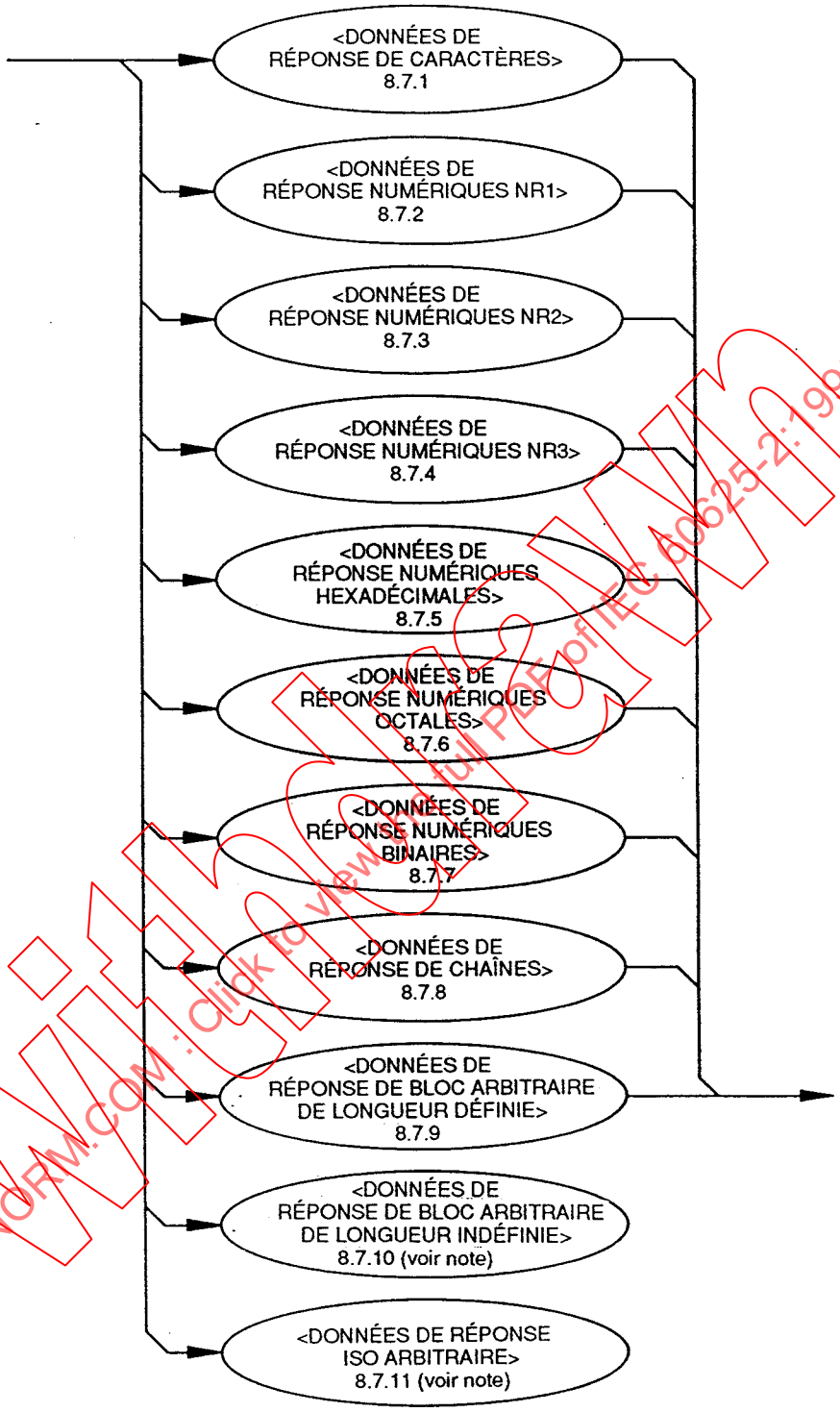


Figure 8-3 – <RESPONSE MESSAGE UNIT> functional element syntax

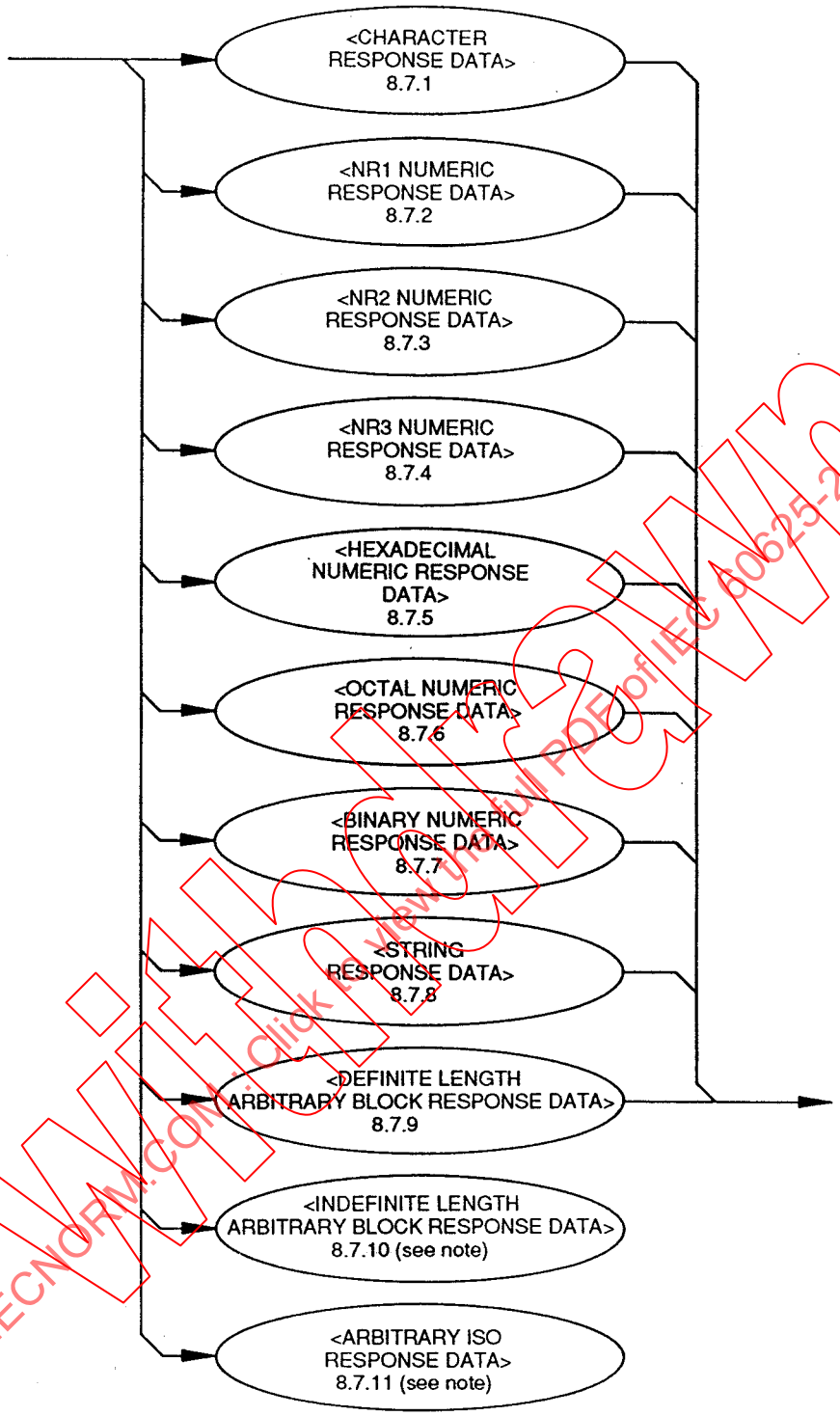
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figure 8-4 – Syntaxe d'élément fonctionnel de <DONNÉES DE RÉPONSE>

NOTE - Les éléments de <DONNÉES DE RÉPONSE ISO ARBITRAIRE> et de <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE> se terminent par un caractère d'arrêt de message implicite. Dans ce cas, le message se termine sans sortie.



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figure 8-4 – <RESPONSE DATA> functional element syntax

NOTE - The <ARBITRARY ISO 7-bit RESPONSE DATA> and <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> elements end with an implied message terminator. In this case, the message terminates with no exit.

8.3.3 *Récapitulatif des éléments fonctionnels*

Elément	*Fonction*
<MESSAGE DE RÉPONSE> figure 8-2	Représente une séquence d'un ou plusieurs éléments d'<UNITÉS DE MESSAGES DE RÉPONSE> séparés par des éléments <SÉPARATEURS D'UNITÉS DE MESSAGES DE RÉPONSE>.
<UNITÉ DE MESSAGE DE RÉPONSE> figure 8-3	Représente une unité de message simple envoyée de l'**appareil**.
<DONNÉES DE RÉPONSE> figure 8-4	Représente un des 11 éléments de <DONNÉES DE RÉPONSE>
<SÉPARATEUR D'UNITÉS DE MESSAGES DE RÉPONSES> 8.4.1	Sépare les éléments d'<UNITÉS DE MESSAGES DE RÉPONSE> les uns des autres dans un <MESSAGE DE RÉPONSE>.
<SÉPARATEUR DE DONNÉES DE RÉPONSE> 8.4.2	Sépare les éléments séquentiels de <DONNÉES DE RÉPONSE> qui sont relatifs au même en-tête ou entre eux.
<SÉPARATEUR D'EN-TÊTES DE RÉPONSE> 8.4.3	Sépare l'en-tête des <DONNÉES DE RÉPONSE>.
<CARACTÈRE D'ARRÊT DE MESSAGES DE RÉPONSE> 8.5	Termine un <MESSAGE DE RÉPONSE>.
<EN-TÊTE DE RÉPONSE> 8.6	Spécifie la fonction du (ou des) élément(s) de <DONNÉES DE RÉPONSE> associé(s). Les caractères alphabétiques et numériques indiquent mnémoniquement la fonction. La structure interne est fournie pour la hiérarchisation d'en-tête.
<DONNÉES DE RÉPONSE DE CARACTÈRE> 8.7.1	Type de données convenant pour l'envoi de courtes chaînes de caractères mnémoniques. Utilisées généralement lorsqu'un paramètre numérique ne convient pas.
<DONNÉES DE RÉPONSES NUMÉRIQUES NR1> 8.7.2	Convenant pour l'envoi de valeurs décimales implicites.
<DONNÉES DE RÉPONSES NUMÉRIQUES NR2> 8.7.3	Convenant pour l'envoi de valeurs décimales explicites.
<DONNÉES DE RÉPONSES NUMÉRIQUES NR3> 8.7.4	Convenant pour l'envoi de valeurs décimales explicites avec un exposant.
<DONNÉES DE RÉPONSE NUMÉRIQUES HEXADÉCIMALES> 8.7.5	Convenant pour l'envoi de valeurs hexadécimales implicites.
<DONNÉES DE RÉPONSE NUMÉRIQUES OCTALES> 8.7.6	Convenant pour l'envoi de valeurs octales implicites.
<DONNÉES DE RÉPONSE NUMÉRIQUES BINAIRES> 8.7.7	Convenant pour l'envoi de valeurs binaires implicites.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.3.3 *Functional element summary*

Element	*Function*
<RESPONSE MESSAGE> figure 8-2	Represents a sequence of one or more <RESPONSE MESSAGE UNIT> elements separated by <RESPONSE MESSAGE UNIT SEPARATOR> elements.
<RESPONSE MESSAGE UNIT> figure 8-3	Represents a single message unit sent from the **device**.
<RESPONSE DATA> figure 8-4	Represents one of the 11 defined <RESPONSE DATA> elements.
<RESPONSE MESSAGE UNIT SEPARATOR> 8.4.1	Separates <RESPONSE MESSAGE UNIT> elements from one another in a <RESPONSE MESSAGE>.
<RESPONSE DATA SEPARATOR> 8.4.2	Separates sequential <RESPONSE DATA> elements that are related to the same header or to each other.
<RESPONSE HEADER SEPARATOR> 8.4.3	Separates the header from the <RESPONSE DATA>.
<RESPONSE MESSAGE TERMINATOR> 8.5	Terminates a <RESPONSE MESSAGE>.
<RESPONSE HEADER> 8.6	Specifies the function of the associated <RESPONSE DATA> element(s). Alpha and numeric characters mnemonically indicate the function. Internal structure is provided for hierarchical header structuring.
<CHARACTER RESPONSE DATA> 8.7.1	A data type suitable for sending short mnemonic character strings. Generally used when a numeric parameter is not suitable.
<NR1 NUMERIC RESPONSE DATA> 8.7.2	Suitable for sending implicit radix decimal values.
<NR2 NUMERIC RESPONSE DATA> 8.7.3	Suitable for sending explicit radix decimal values.
<NR3 NUMERIC RESPONSE DATA> 8.7.4	Suitable for sending explicit radix decimal values with an exponent.
<HEXADECIMAL NUMERIC RESPONSE DATA> 8.7.5	Suitable for sending implicit radix hexadecimal values.
<OCTAL NUMERIC RESPONSE DATA>8.7.6	Suitable for sending implicit radix octal values.
<BINARY NUMERIC RESPONSE DATA> 8.7.7	Suitable for sending implicit radix binary values.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Elément	*Fonction*
<DONNÉES DE RÉPONSE DE CHAÎNES> 8.7.8	Type de donnée convenant pour l'envoi de chaînes de caractères ISO où le contenu a besoin d'être "caché" par des séparateurs. Cet élément est généralement utilisé pour envoyer des données pour affichage direct sur un appareil.
<DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE> 8.7.9	Type de donnée convenant pour l'envoi de blocs d'informations binaires de 8 bits lorsque la longueur est connue avant traitement.
<DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE> 8.7.10	Type de donnée convenant pour l'envoi de blocs d'informations binaires de 8 bits lorsque la longueur n'est pas connue avant traitement ou lorsque le calcul de la longueur avant traitement n'est pas désirable.
<DONNÉES DE RÉPONSE ISO ARBITRAIRES> 8.7.11	Convenant pour l'envoi d'octets ISO arbitraires lorsque des types de variantes de données ne sont pas utilisables.

8.4 *Eléments fonctionnels de séparation*

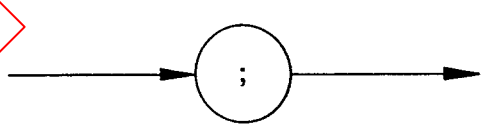
8.4.1 *<SÉPARATEUR D'UNITÉS DE MESSAGES DE RÉPONSE>*

8.4.1.1 Fonction

Le <SÉPARATEUR D'UNITÉS DE MESSAGES DE RÉPONSE> sépare les éléments d'<UNITÉS DE MESSAGES DE RÉPONSE> séquentiels les uns des autres lorsque des éléments d'<UNITÉS DE MESSAGES DE RÉPONSE> multiples sont envoyés dans un <MESSAGE DE RÉPONSE>.

8.4.1.2 Syntaxe de codage

Un <SÉPARATEUR D'UNITÉS DE MESSAGES DE RÉPONSE> est défini par:



8.4.2 *<SÉPARATEUR DE DONNÉES DE RÉPONSE>*

8.4.2.1 Fonction

Le <SÉPARATEUR DE DONNÉES DE RÉPONSE> sépare les éléments séquentiels de <DONNÉES DE RÉPONSE> les uns des autres lorsque des éléments de données multiples sont envoyés.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993 Withdrawn

Element	*Function*
<STRING RESPONSE DATA> 8.7.8	A data type suitable for sending ISO 7-bit character strings where the contents need to be "hidden" by delimiters. This element is generally used to send data for direct display on a device.
<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> 8.7.9	A data type suitable for sending blocks of 8-bit binary information when the length is known beforehand.
<INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> 8.7.10	A data type suitable for sending blocks of 8-bit binary information when the length is not known beforehand or when computing the length beforehand is undesirable.
<ARBITRARY ISO 7-bit RESPONSE DATA> 8.7.11	Suitable for sending arbitrary ISO 7-bit codes when alternate data types are unworkable.

8.4 *Separator functional elements*

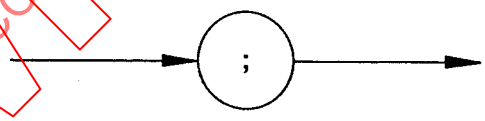
8.4.1 *<RESPONSE MESSAGE UNIT SEPARATOR>*

8.4.1.1 Function

The <RESPONSE MESSAGE UNIT SEPARATOR> separates sequential <RESPONSE MESSAGE UNIT> elements from one another when multiple <RESPONSE MESSAGE UNIT> elements are sent in a <RESPONSE MESSAGE>.

8.4.1.2 Encoding syntax

A <RESPONSE MESSAGE UNIT SEPARATOR> is defined as



8.4.2 *<RESPONSE DATA SEPARATOR>*

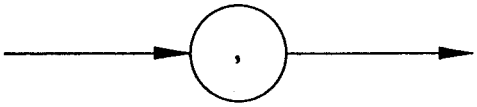
8.4.2.1 Function

The <RESPONSE DATA SEPARATOR> separates sequential <RESPONSE DATA> elements from one another when multiple data elements are sent.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

8.4.2.2 Syntaxe de codage

Un <SÉPARATEUR DE DONNÉES DE RÉPONSE> est défini par



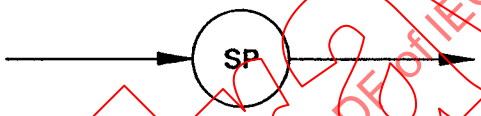
8.4.3 *<SÉPARATEUR D'EN-TÊTES DE RÉPONSE>*

8.4.3.1 Fonction

Le <SÉPARATEUR D'EN-TÊTES DE RÉPONSE> sépare sans ambiguïté l'<EN-TÊTE DE RÉPONSE> des <DONNÉES DE RÉPONSE>.

8.4.3.2 Syntaxe de codage

Un <SÉPARATEUR D'EN-TÊTES DE RÉPONSE> est défini par



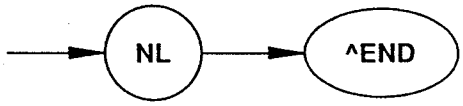
8.5 *<CARACTÈRE D'ARRÊT DE MESSAGES DE RÉPONSE>*

8.5.1 *Fonction*

La fonction de l'élément <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE> est de terminer une séquence d'un ou plusieurs éléments d'<UNITÉS DE MESSAGES DE RÉPONSE>.

8.5.2 *Syntaxe de codage*

Un <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE> est défini par



8.6 *<EN-TÊTE DE RÉPONSE>*

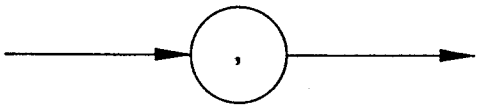
8.6.1 *Fonction*

L'<EN-TÊTE DE RÉPONSE> est utilisable par le concepteur de l'appareil pour des réponses spécifiques à l'appareil. Il peut être utilisé, par exemple, pour créer des réponses en format d'<UNITÉ DE MESSAGE DE PROGRAMMATION> directement renvoyable ou pour identifier des données de réponse au contrôleur.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.4.2.2 Encoding syntax

A <RESPONSE DATA SEPARATOR> is defined as



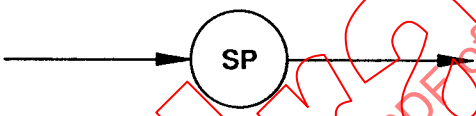
8.4.3 *<RESPONSE HEADER SEPARATOR>*

8.4.3.1 Function

The <RESPONSE HEADER SEPARATOR> unambiguously separates the <RESPONSE HEADER> from the <RESPONSE DATA>.

8.4.3.2 Encoding syntax

A <RESPONSE HEADER SEPARATOR> is defined as



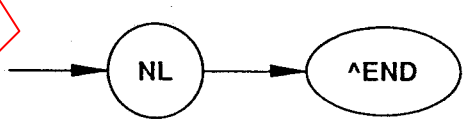
8.5 *<RESPONSE MESSAGE TERMINATOR>*

8.5.1 *Function*

The <RESPONSE MESSAGE TERMINATOR> element's function is to terminate a sequence of one or more <RESPONSE MESSAGE UNIT> elements.

8.5.2 *Encoding syntax*

A <RESPONSE MESSAGE TERMINATOR> is defined as



8.6 *<RESPONSE HEADER>*

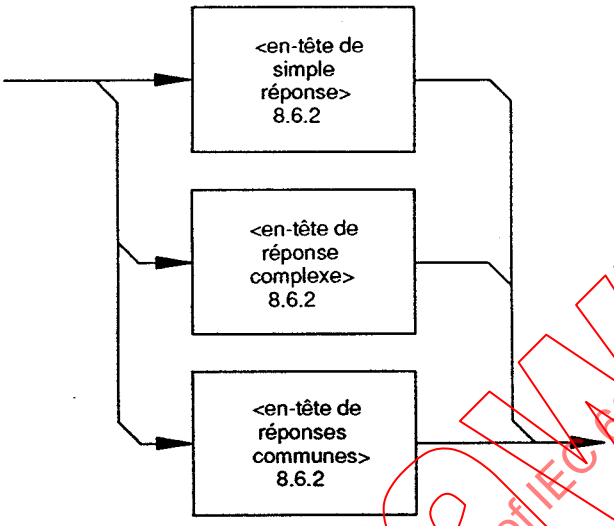
8.6.1 *Function*

The <RESPONSE HEADER> is available for use by the device designer for device-specific responses. It may be used, for example, to create responses in directly resendable <PROGRAM MESSAGE UNIT> format or to identify response data to the controller.

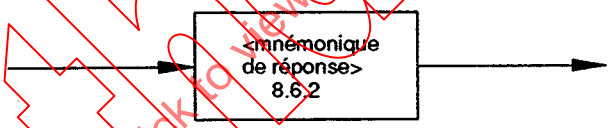
Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

8.6.2 *Syntaxe de codage*

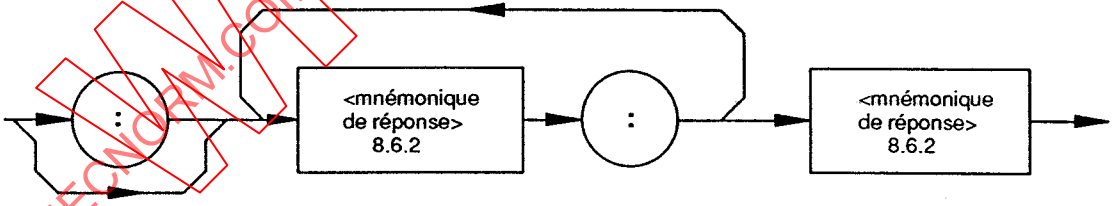
Un <EN-TÊTE DE RÉPONSE> est défini par



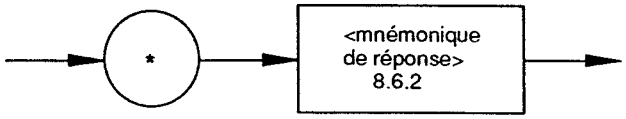
Où <mnémonique de simple réponse> est défini par



Où <en-tête de réponses complexes> est défini par



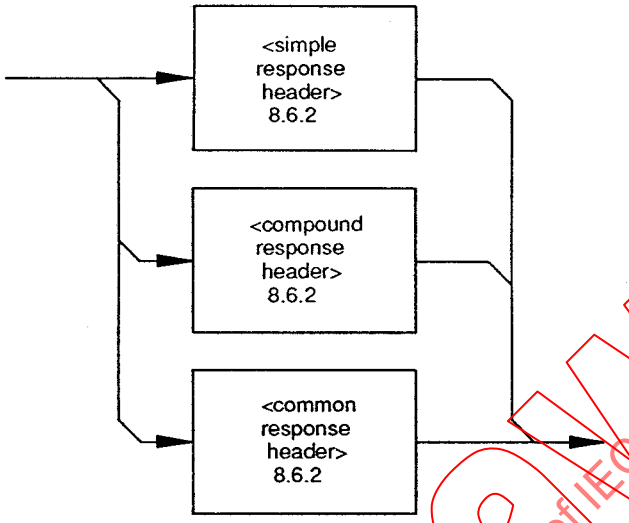
Où <en-tête de réponses communes> est défini par



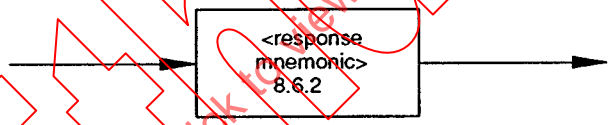
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.6.2 *Encoding syntax*

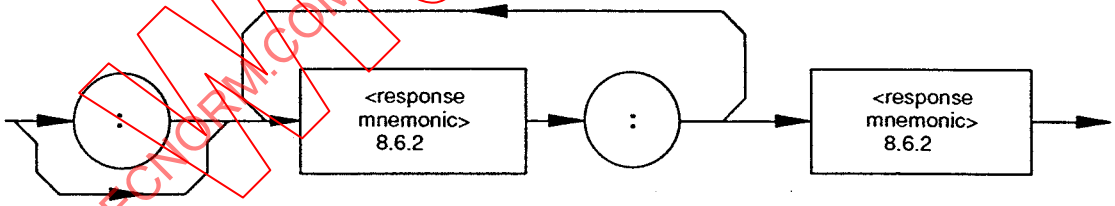
A <RESPONSE HEADER> is defined as



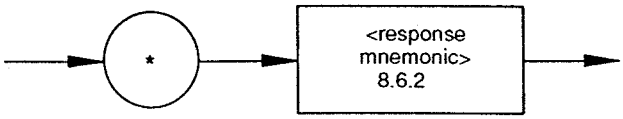
Where <simple response mnemonic> is defined as



Where <compound response header> is defined as

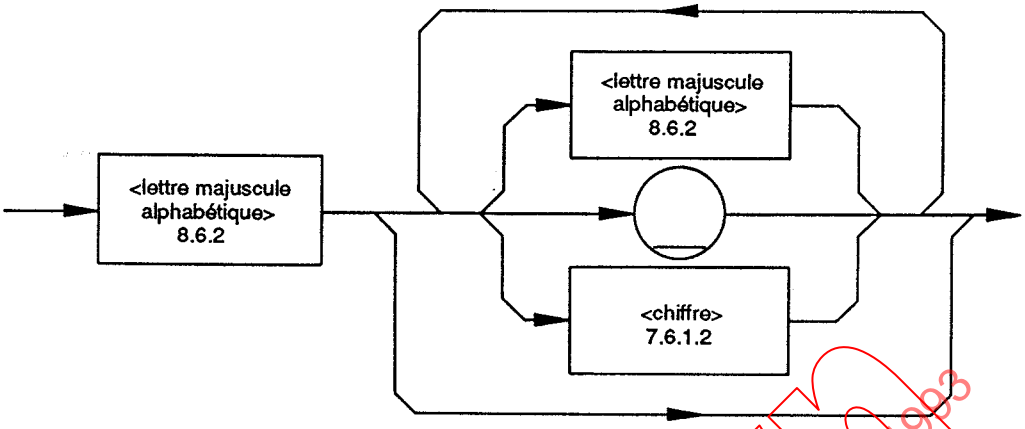


Where <common response header> is defined as



Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Où <mnémonique de réponse> est défini par



Où <lettre majuscule alphabétique> est défini comme un octet codé ISO simple dans la gamme 41 – 5A (65-90 en décimal).

Où <chiffre> est défini en 7.6.1.2.

Où (_) représente un "soulignement", un octet codé ISO simple avec la valeur 5F (95 en décimal).

8.6.3 *Règles*

Le <mnémonique de réponse> devra avoir une longueur maximale de 12 caractères.

La relation entre un <mnémonique de réponse> et la fonction qui s'y rattache doit être évidente.

Les sémantiques des éléments mnémoniques de réponse complexes utilisés dans un <EN-TÊTE DE RÉPONSE> sont au-delà de la portée de cette norme.

Les éléments d'<en-têtes de réponses communes> ne peuvent être utilisés que pour représenter les éléments d'<EN-TÊTES D'INSTRUCTIONS DE PROGRAMMATION> définis à l'article 10.

8.7 *Eléments fonctionnels de <DONNÉES DE RÉPONSE>*

Un élément fonctionnel de <DONNÉES DE RÉPONSE> véhicule une variété d'informations réponses. Le type d'élément est déterminé par l'interrogation pour éclaircissement ou l'équivalent d'interrogation (par exemple, développement d'une macro-instruction).

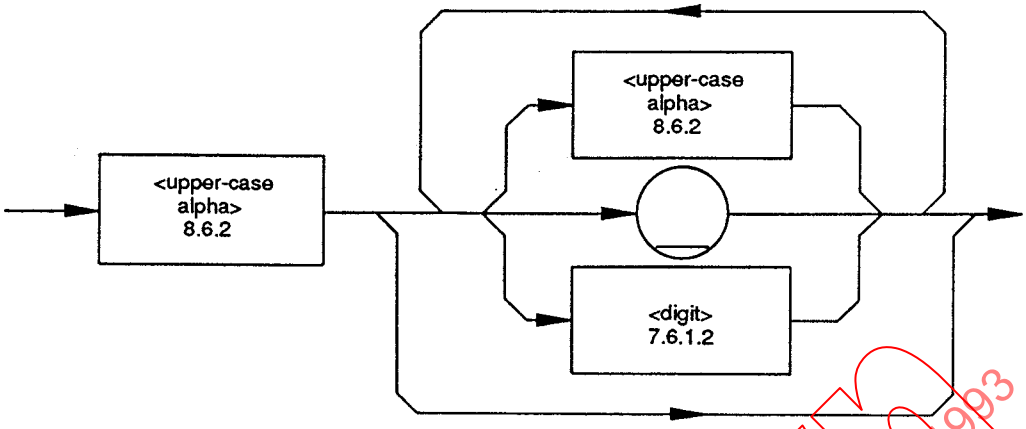
8.7.1 *<DONNÉES DE RÉPONSE DE CARACTÈRES>*

8.7.1.1 Fonction

L'élément fonctionnel de <DONNÉES DE RÉPONSE DE CARACTÈRES> est utilisé pour acheminer des informations mieux exprimées mnémoniquement comme une chaîne alphabétique ou alphanumérique courte. Ceci est utile lorsque les paramètres numériques sont inappropriés.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Where <response mnemonic> is defined as



Where <upper-case alpha> is defined as a single ISO 7-bit encoded byte in the range 41 – 5A (65-90 decimal).

Where <digit> is defined in 7.6.1.2.

Where (_) represents an "underscore", a single ISO 7-bit encoded byte with the value 5F (95 decimal).

8.6.3 *Rules*

The <response mnemonic> shall have a maximum length of 12 characters.

The relation of a <response mnemonic> and its related function shall be readily apparent.

The semantics of <compound response mnemonic> elements used in a <RESPONSE HEADER> is beyond the scope of this standard.

<common response header> elements may only be used to represent <COMMAND PROGRAM HEADER> elements defined in clause 10.

8.7 *<RESPONSE DATA> functional elements*

A <RESPONSE DATA> functional element conveys a variety of response information. The element type is determined by the eliciting query or query equivalent (e.g. macro expansion).

8.7.1 *<CHARACTER RESPONSE DATA>*

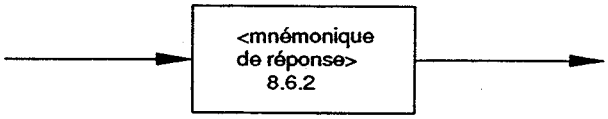
8.7.1.1 Function

The <CHARACTER RESPONSE DATA> functional element is used to convey information best expressed mnemonically as a short alpha or alphanumeric string. It is useful when numeric parameters are inappropriate.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.1.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE DE CARACTÈRES> est définie par



Où <mnémonique de réponse> est défini en 8.6.2.

8.7.1.3 Règles

Le <mnémonique de réponse> doit se conformer aux règles fixées de 7.6.1.4.1 à 7.6.1.4.3.

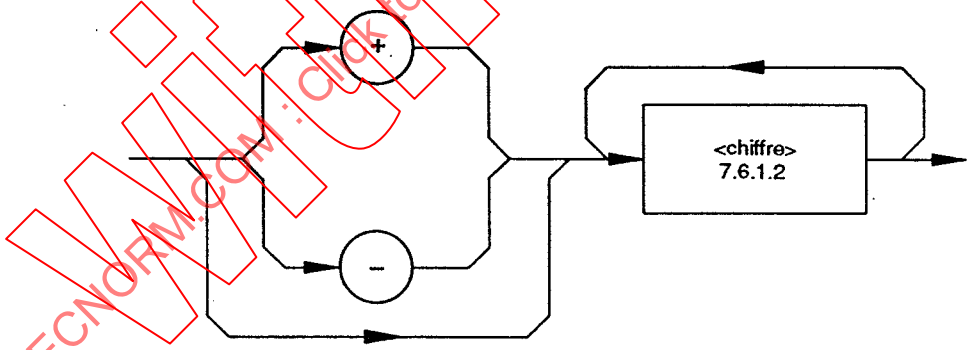
8.7.2 *<DONNÉES DE RÉPONSES NUMÉRIQUES NR1>*

8.7.2.1 Fonction

Les éléments de <DONNÉES DE RÉPONSE NUMÉRIQUES NR1> consistent en un ensemble de représentations à virgule implicites de valeurs numériques. C'est-à-dire qu'un emplacement de virgule est tenu d'être placé (fixé mais non transmis) à la fin à droite de la chaîne de chiffres.

8.7.2.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSES NUMÉRIQUES NR1> est définie par



8.7.2.3 Règles

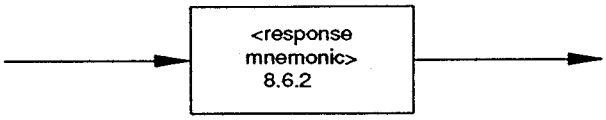
8.7.2.3.1 Hors gamme

La représentation numérique d'informations hors gamme doit être envoyée avec le même type de données numériques que pour les informations de la gamme avec une valeur en dehors de la gamme normale de mesures. De même, le bit DDE sera mis dans le registre d'état d'événements standards. Voir 11.5.1.1.6.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.1.2 Encoding syntax

A <CHARACTER RESPONSE DATA> is defined as



Where <response mnemonic> is defined in 8.6.2.

8.7.1.3 Rules

The <response mnemonic> shall comply with the rules as stated in 7.6.1.4.1 to 7.6.1.4.3.

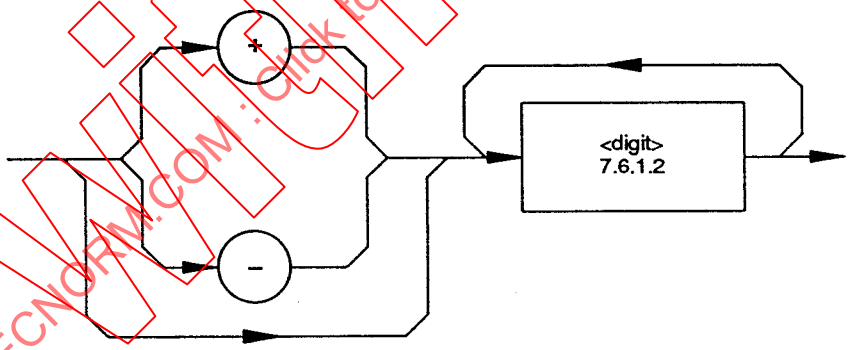
8.7.2 *<NR1 NUMERIC RESPONSE DATA>*

8.7.2.1 Function

<NR1 NUMERIC RESPONSE DATA> elements consist of a set of implicit point representations of numeric values. That is, a radix point is implicitly considered to be placed (fixed but not transmitted) at the right end of the string of digits.

8.7.2.2 Encoding syntax

An <NR1 NUMERIC RESPONSE DATA> is defined as



8.7.2.3 Rules

8.7.2.3.1 Out-of-range

Numeric representation of out-of-range information shall be sent with the same numeric data type as for in-range information with a value outside of the normal range of measurements. Also, the DDE bit shall be set in the Standard Event Status Register. See 11.5.1.1.6.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Par exemple, un voltmètre avec une capacité de mesure maximale de 100 V pourrait indiquer qu'il mesurait au-dessus de 100 V en renvoyant une réponse avec une valeur "hors gamme" prédéfinie au-dessus de 100. Cette valeur devra être renvoyée comme une <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> si les réponses normales de la gamme sont renvoyées comme des éléments de <DONNÉES DE RÉPONSE NUMÉRIQUES NR1>.

8.7.2.3.2 Gamme admissible

La gamme admissible pour NR1 est la même que la gamme pour les nombres à virgule flottante en double précision (+/– 179,6 . . . E+306). Voir CEI 559 et l'article 9 pour les détails des formats en virgule flottante.

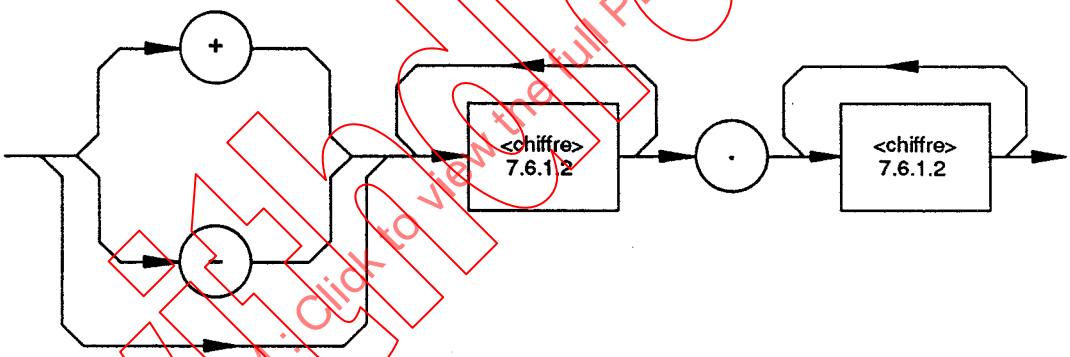
8.7.3 *<DONNÉES DE RÉPONSE NUMÉRIQUES NR2>*

8.7.3.1 Fonction

Les éléments de <DONNÉES DE RÉPONSE NUMÉRIQUES NR2> sont les représentations de valeurs numériques à virgule explicite.

8.7.3.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE NUMÉRIQUE NR2> est définie par



Où <chiffre> est défini en 7.6.1.2.

8.7.3.3 Règles

La gamme admissible pour NR2 est la même que la gamme pour les nombres à virgule flottante en double précision (+/– 179,76. . . E+306). Voir CEI 559 et l'article 9 pour les détails des formats en virgule flottante.

Les règles de codage hors gamme sont identiques à celles de 8.7.2.3.

8.7.4 *<DONNÉES DE RÉPONSE NUMÉRIQUES NR3>*

8.7.4.1 Fonction

Les <DONNÉES DE RÉPONSE NUMÉRIQUES NR3> sont des représentations de valeurs numériques mises à l'échelle à emplacement de virgule explicite, avec une notation exponentielle.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

For example, a voltmeter with a maximum measurement capability of 100 V could indicate that it was measuring above 100 V by returning a response with a pre-defined "overrange" value above 100. This value shall be returned as <NR1 NUMERIC RESPONSE DATA> if the normal in-range responses are returned as <NR1 NUMERIC RESPONSE DATA> elements.

8.7.2.3.2 Allowable range

The allowable range for NR1's is the same as the range for double precision floating point numbers (+/– 179.76. . . E+306). See IEC 559 and clause 9 for details of floating point formats.

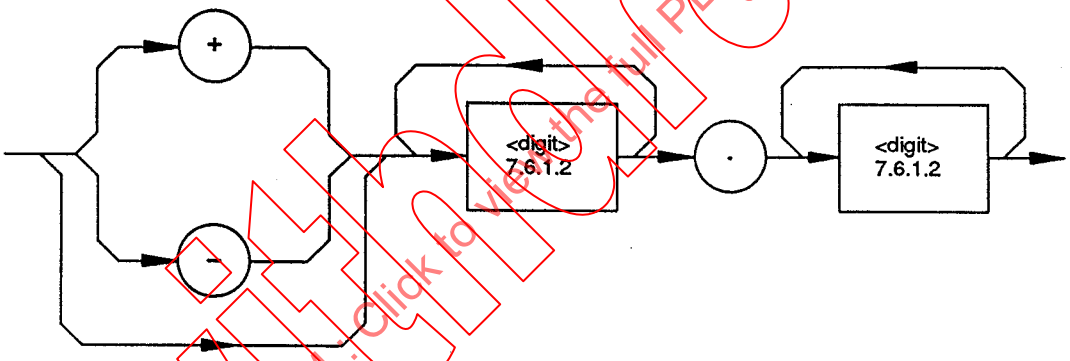
8.7.3 *<NR2 NUMERIC RESPONSE DATA>*

8.7.3.1 Function

<NR2 NUMERIC RESPONSE DATA> elements are the representations of explicit point numeric values.

8.7.3.2 Encoding syntax

An <NR2 NUMERIC RESPONSE DATA> is defined as



Where <digit> is defined in 7.6.1.2.

8.7.3.3 Rules

The allowable range for NR2's is the same as the range for double precision floating point numbers (+/– 179.76. . . E+306). See IEC 559 and clause 9 for details of floating point formats.

Out-of-range encoding rules are identical to those of 8.7.2.3.

8.7.4 *<NR3 NUMERIC RESPONSE DATA>*

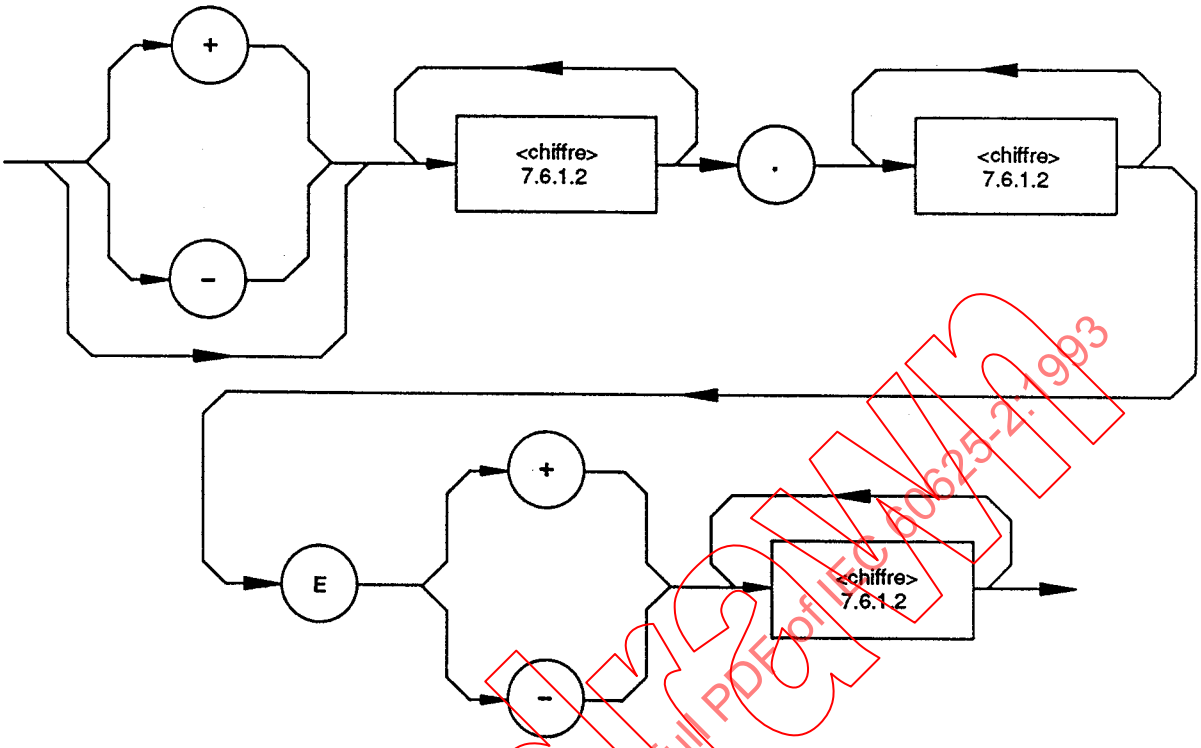
8.7.4.1 Function

<NR3 NUMERIC RESPONSE DATA> elements are representations of scaled explicit radix point numeric values together with an exponent notation.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.4.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE NUMÉRIQUE NR3> est définie par



Où <chiffre> est défini en 7.6.1.2.

8.7.4.3 Règles

La gamme admissible pour NR3 est la même que la gamme pour les nombres à virgule flottante en double précision (+/– 179,76. . . E+306). Voir CEI 559 et l'article 9 pour les détails des formats en virgule flottante.

Les règles de codage hors gamme sont identiques à celles de 8.7.2.3.

8.7.5 *<DONNÉES DE RÉPONSE NUMÉRIQUES HEXADÉCIMALES>*

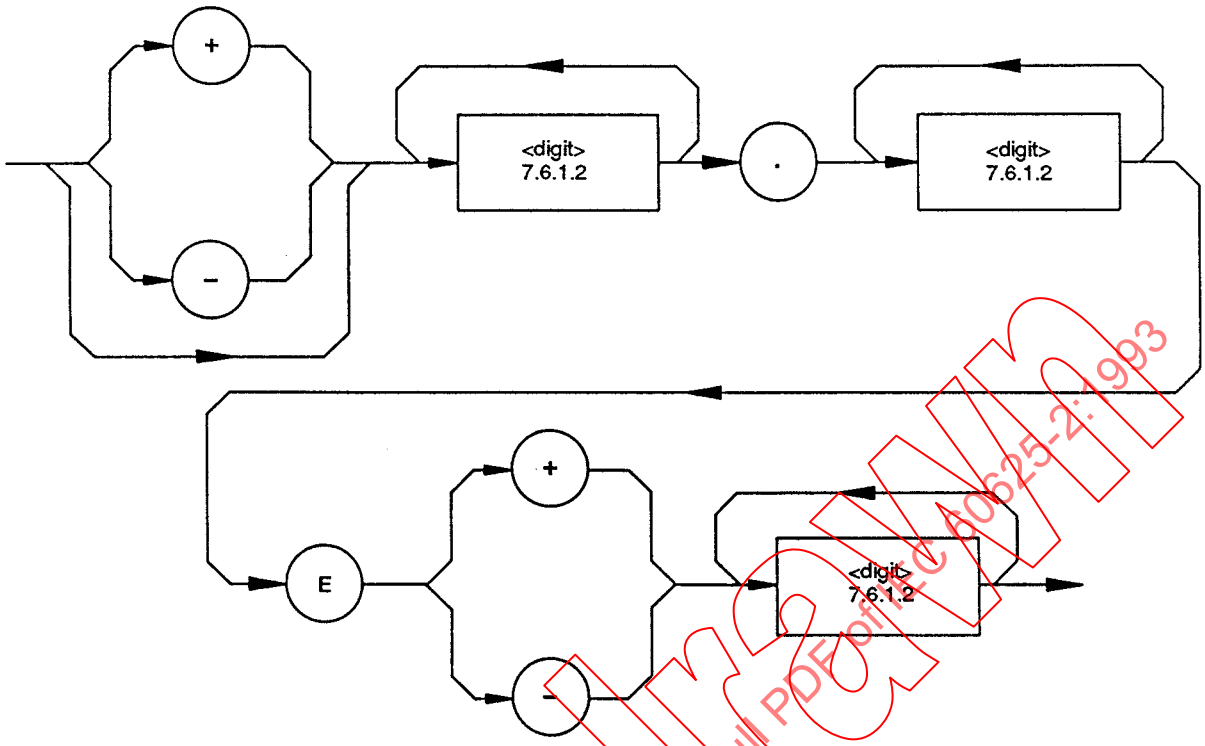
8.7.5.1 Fonction

L'élément de <DONNÉES DE RÉPONSES NUMÉRIQUES HEXADÉCIMALES> est utilisé pour représenter en base 16 des informations numériques de base implicite.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.4.2 Encoding syntax

An <NR3 NUMERIC RESPONSE DATA> is defined as



Where <digit> is defined in 7.6.1.2.

8.7.4.3 Rules

The allowable range for NR3's is the same as the range for double precision floating point numbers (+/– 179.76. . . E+306). See IEC 559 and clause 9 for details of floating point formats.

Out-of-range encoding rules are identical to those of 8.7.2.3.

8.7.5 *<HEXADECIMAL NUMERIC RESPONSE DATA>*

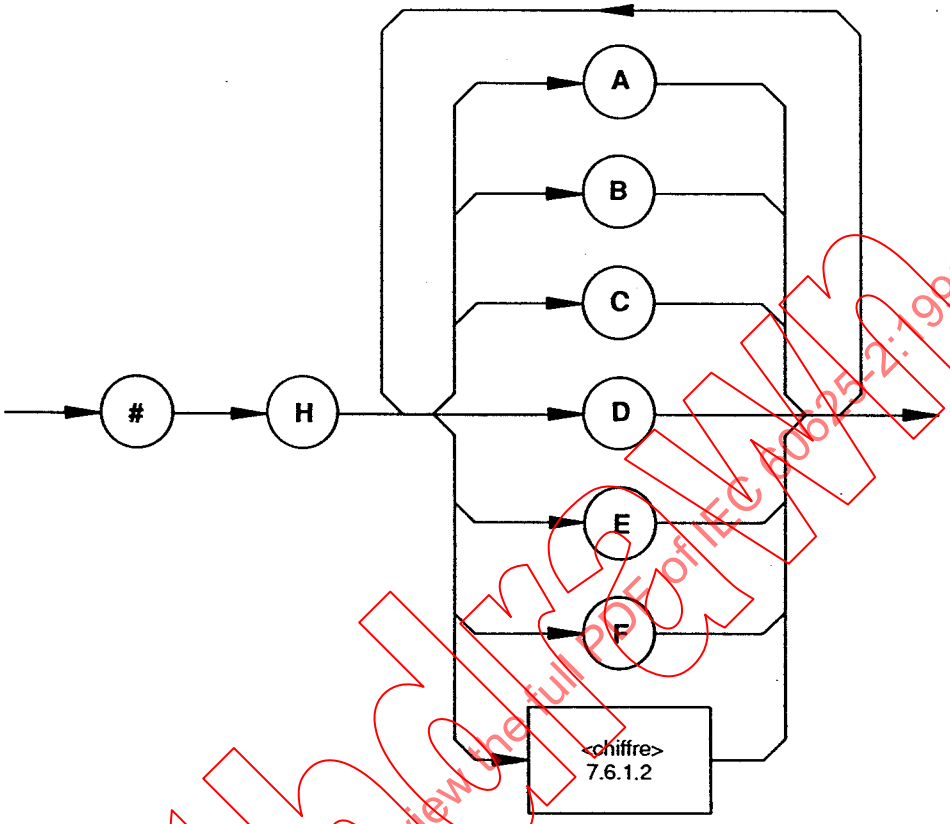
8.7.5.1 Function

The <HEXADECIMAL NUMERIC RESPONSE DATA> element is used to represent implicit radix, base 16 numeric information.

IECNORM.COM : Click to view the full PDF of iec 60625-2:1993
Withdrawn

8.7.5.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE NUMÉRIQUE HEXADÉCIMALE> est définie par



Où <chiffre> est défini en 7.6.1.2.

8.7.5.3 Règles

Les règles de codage sont identiques à celles de 7.7.4.4.1, sauf l'option en variante du code de 7 bits ISO (minuscule) qui n'est pas autorisée.

8.7.6 *<DONNÉES DE RÉPONSE NUMÉRIQUES OCTALES>*

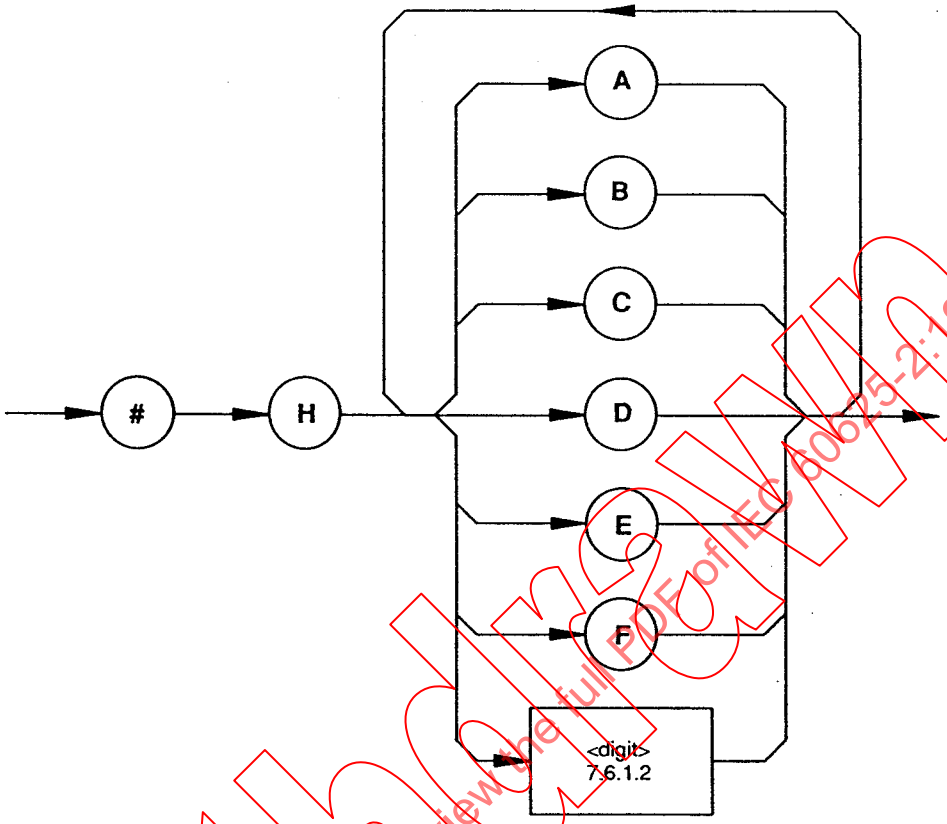
8.7.6.1 Fonction

L'élément de <DONNÉES DE RÉPONSE NUMÉRIQUES OCTALES> est utilisé pour représenter en base 8 des informations numériques de base implicite.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.5.2 Encoding syntax

A <HEXADECIMAL NUMERIC RESPONSE DATA> is defined as



Where <digit> is defined in 7.6.1.2.

8.7.5.3 Rules

The encoding rules are identical to those in 7.7.4.4.1, except that the alternate (lower-case) ISO 7-bit code option is disallowed.

8.7.6 *<OCTAL NUMERIC RESPONSE DATA>*

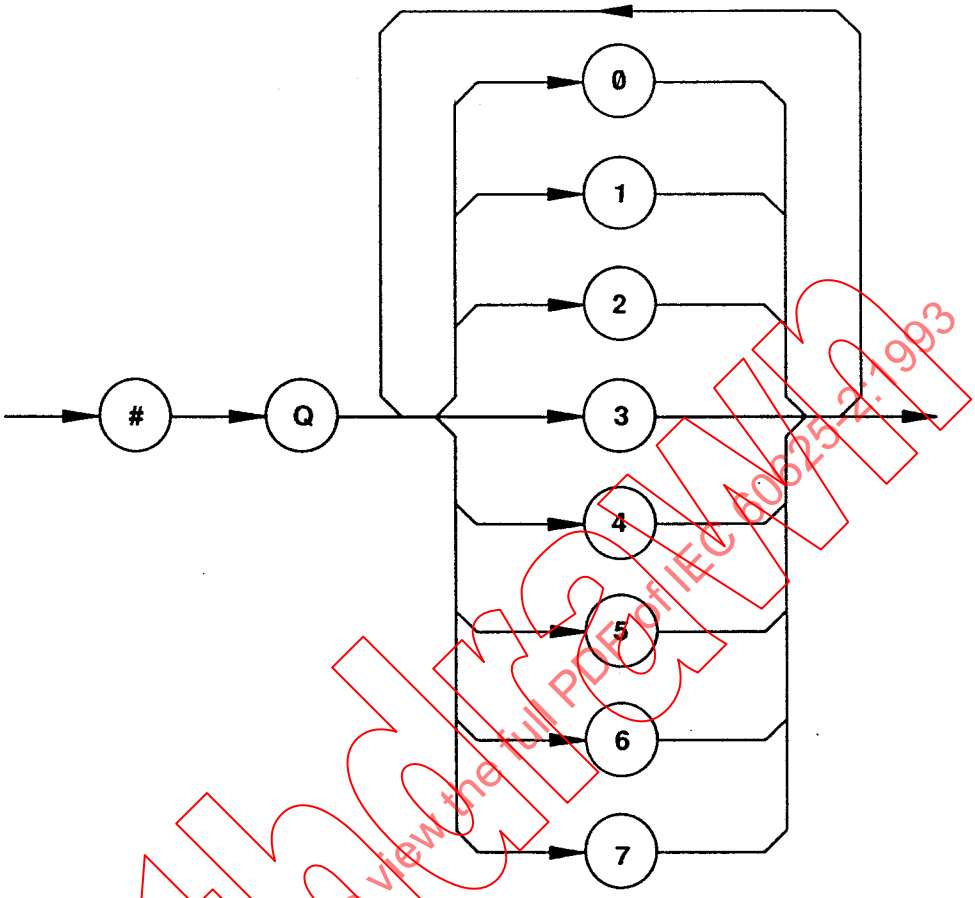
8.7.6.1 Function

The <OCTAL NUMERIC RESPONSE DATA> element is used to represent base 8, implicit radix numeric information.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.6.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE NUMÉRIQUE OCTALE> est définie par



8.7.6.3 Règles

Les règles de codage sont identiques à celles de 7.7.4.4.2, sauf l'option en variante du code de 7 bits ISO (minuscule) qui n'est pas autorisée.

8.7.7 *<DONNÉES DE RÉPONSE NUMÉRIQUES BINAIRES>*

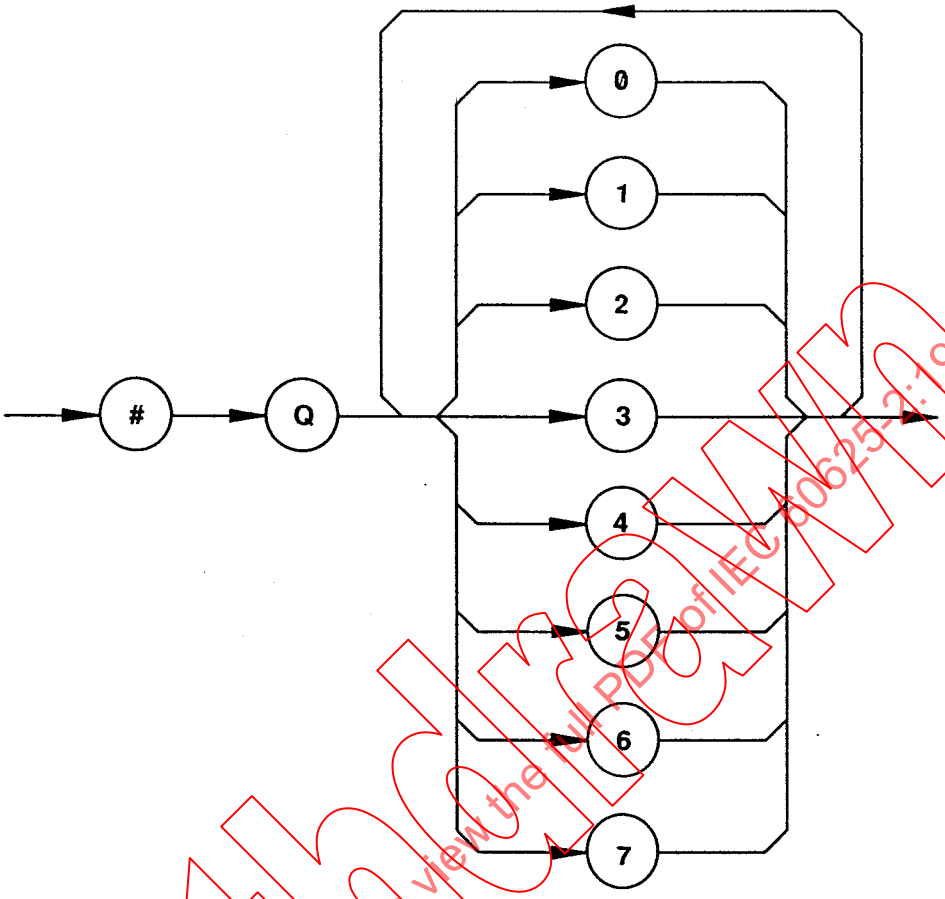
8.7.7.1 Fonction

L'élément de <DONNÉES DE RÉPONSE NUMÉRIQUES BINAIRES> est utilisé pour représenter en base 2 des informations numériques de base implicite.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.6.2 Encoding Syntax

An <OCTAL NUMERIC RESPONSE DATA> is defined as



8.7.6.3 Rules

The encoding rules are identical to those in 7.7.4.4.2, except that the alternate (lower-case) ISO 7-bit code option is disallowed.

8.7.7 *<BINARY NUMERIC RESPONSE DATA>*

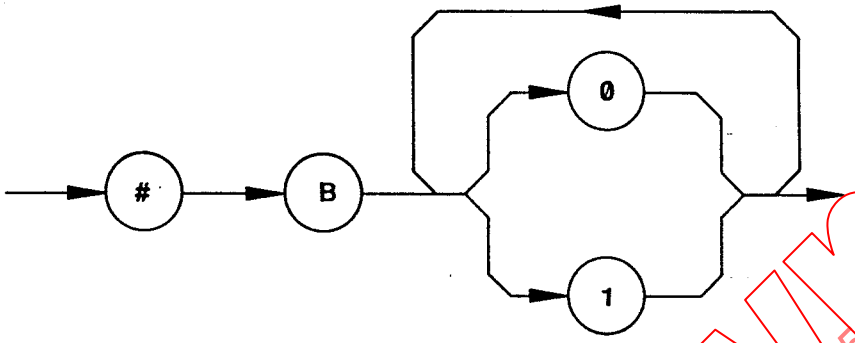
8.7.7.1 Function

The <BINARY NUMERIC RESPONSE DATA> element is used to represent base 2, implicit radix numeric information.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.7.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE NUMÉRIQUE BINAIRE> est définie par



8.7.7.3 Règles

Les règles de codage sont identiques à celles de 7.7.4.4.3, sauf l'option en variante du code de 7 bits ISO (minuscule) qui n'est pas autorisée.

8.7.8 *<DONNÉES DE RÉPONSE DE CHAÎNES>*

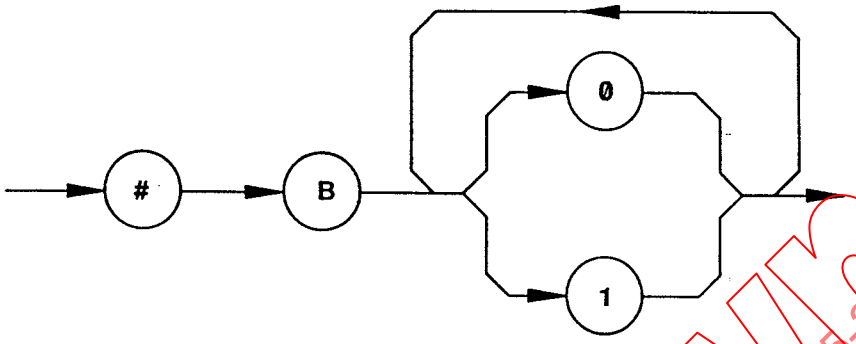
8.7.8.1 Fonction

Les éléments de <DONNÉES DE RÉPONSE DE CHAÎNES> permettent à N'IMPORTE QUEL caractère dans le code de 7 bits ISO (comprenant des caractères non imprimables) d'être transmis comme partie d'un message. Ce champ de données est particulièrement utile lorsque le texte doit être affiché (par exemple, sur une imprimante ou un appareil type CRT). L'élément de <DONNÉES DE RÉPONSE DE CHAÎNES> permet l'utilisation de fichiers-formats tels que retour chariot, interligne ou espace pour formater correctement le texte.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.7.2 Encoding syntax

A <BINARY NUMERIC RESPONSE DATA> is defined as



8.7.7.3 Rules

The encoding rules are identical to those in clause 7.7.4.4.3, except that the alternate (lower-case) ISO 7-bit code option is disallowed.

8.7.8 *<STRING RESPONSE DATA>*

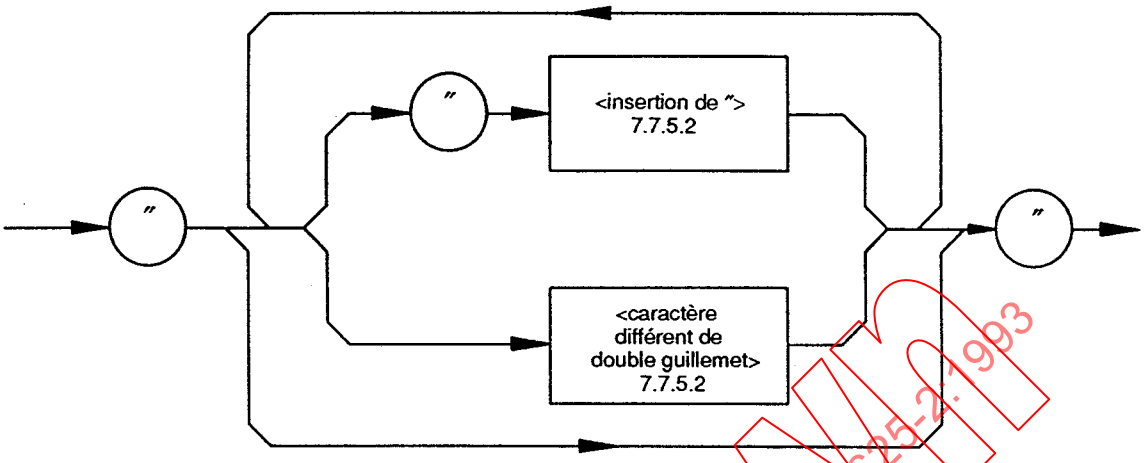
8.7.8.1 Function

The <STRING RESPONSE DATA> elements allow any character in the ISO 7-bit code (including non-printable characters) to be transmitted as a part of a message. This data field is particularly useful when text is to be displayed (for example, on a printer or CRT type device). The <STRING RESPONSE DATA> element permits the use of format effectors such as carriage return, newline, or space to format text correctly.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

8.7.8.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE DE CHAÎNE> est définie par



Où <insertion de "> est définie en 7.7.5.2.

Où <caractère différent de double guillemet> est défini en 7.7.5.2.

8.7.8.3 Règles

Les règles de codage sont les mêmes que pour <DONNÉES DE PROGRAMMATION DE CHAÎNES>, voir 7.7.5.4.

Le bit 8 doit être mis faux lors de la transmission de données de 7 bits ISO.

8.7.8.4 Notes et exemples

Des variantes de méthodes pour englober le texte de 7 bits ISO, limité à une réponse de l'appareil, sont l'élément de <DONNÉES DE RÉPONSE de 7 BITS ISO ARBITRAIRES>, voir 8.7.11, l'élément de <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE>, voir 8.7.9, et l'élément de <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE>, voir 8.7.10. Ce dernier élément ne doit être utilisé que si des variantes de formats ne sont pas utilisables.

8.7.9 *<DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE>*

8.7.9.1 Fonction

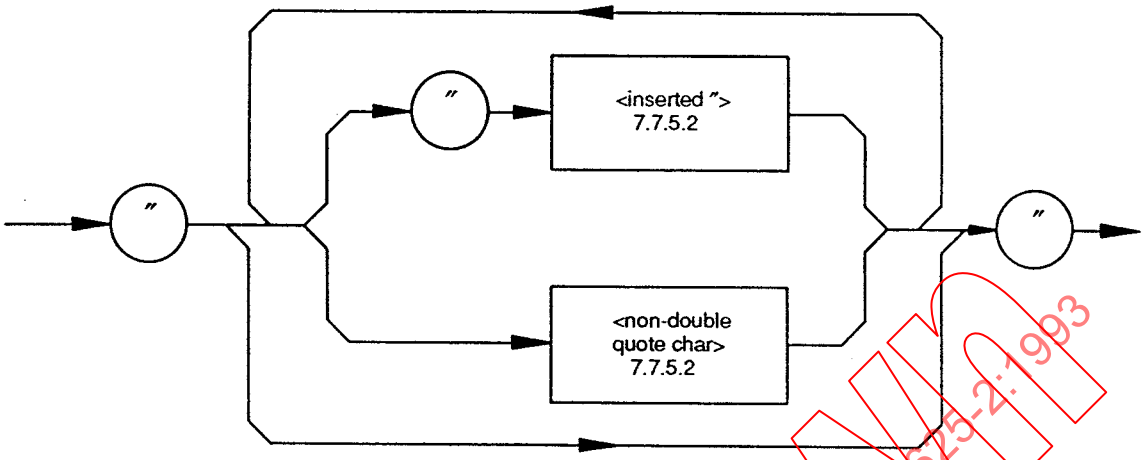
L'élément de <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE> permet à n'importe quel type de donnée dépendant de l'appareil d'être transmis par le **système d'interface** comme une série d'octets de données. Cet élément est particulièrement utile pour envoyer de grandes quantités de données, codes de 7 bits ISO étendus à 8 bits, ou d'autres données qui ne sont pas directement affichables.

Des codages binaires recommandés spécifiques à utiliser avec cet élément sont décrits à l'article 9.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.8.2 Encoding syntax

A <STRING RESPONSE DATA> is defined as



Where <inserted "> is defined in 7.7.5.2.

Where <non-double quote char> is defined in 7.7.5.2.

8.7.8.3 Rules

The encoding rules are the same as for <STRING PROGRAM DATA>, see 7.7.5.4.

Bit 8 shall be set false when transmitting ISO 7-bit data.

8.7.8.4 Notes and examples

Alternate methods of encapsulating ISO 7-bit text, limited to response from a device, are the <ARBITRARY ISO 7-bit RESPONSE DATA> element, see 8.7.11, the <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element, see 8.7.9, and the <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element, see 8.7.10. The last element should only be used if alternate formats are unworkable.

8.7.9 *<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>*

8.7.9.1 Function

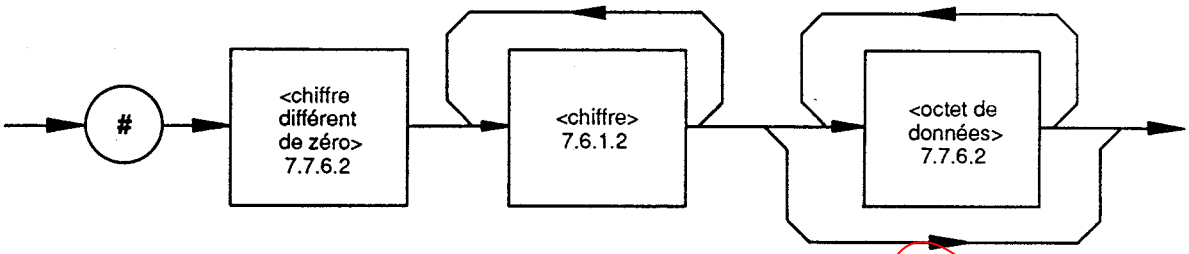
The <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element allows any type of device-dependent data to be transmitted over the **system interface** as a series of 8-bit data bytes. This element is particularly useful for sending large quantities of data, 8-bit extended ISO 7-bit codes, or other data which are not directly displayable.

Specific recommended binary encodings for use with this element are described in clause 9.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.9.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE> est définie par



Où <chiffre différent de zéro> est défini en 7.7.6.2.

Où <chiffre> est défini en 7.6.1.2.

Où <octets de données> est défini en 7.7.6.2.

8.7.9.3 Règles

Les règles sont identiques à celles de 7.7.6.4.

8.7.9.4 Notes et exemples

Les concepteurs d'appareils doivent considérer les implications de la taille et de la disponibilité des mémoires tampons dans les contrôleurs récepteurs. Il est recommandé aux concepteurs de ne pas supposer que tous les contrôleurs seront automatiquement capables d'accepter des messages longs. Voir l'article 14 pour les exigences de spécifications de l'appareil.

8.7.10 *<DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE>*

8.7.10.1 Fonction

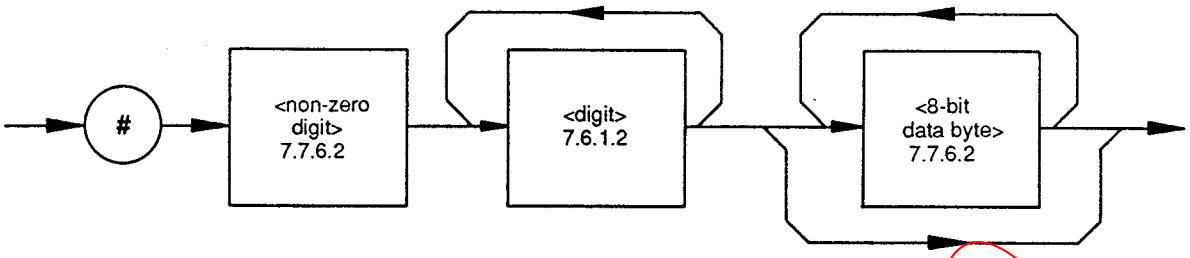
L'élément de <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE> est utile lors de la transmission d'octets de données quand la longueur de la transmission n'est pas connue ou lorsque la vitesse de transmission ou d'autres conditions empêchent la segmentation de la sortie en blocs de longueur connue.

Les diagrammes de syntaxe fonctionnelle permettent l'utilisation de cet élément uniquement comme l'élément final dans un <MESSAGE DE RÉPONSE TERMINÉ> (voir figure 8-4).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.9.2 Encoding syntax

A <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> is defined as



Where <non-zero digit> is defined in 7.7.6.2.

Where <digit> is defined in 7.6.1.2.

Where <8-bit data bytes> is defined in 7.7.6.2.

8.7.9.3 Rules

The rules are identical to those of 7.7.6.4.

8.7.9.4 Notes and examples

Device designers should consider the implications of buffer size and availability in the intended receiving controllers. Designers should not assume that all controllers will automatically be capable of accepting lengthy messages. See clause 14 for controller specification requirements.

8.7.10 *<INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>*

8.7.10.1 Function

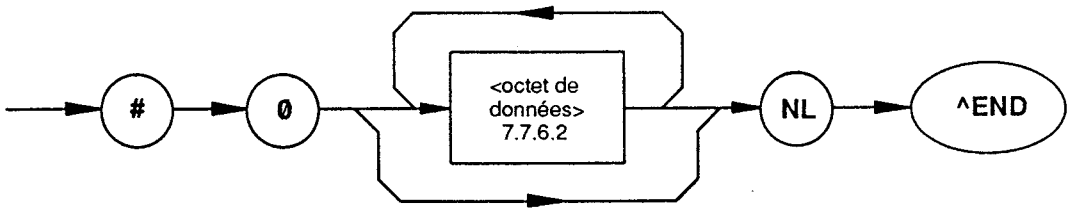
The <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element is useful when transmitting 8-bit data bytes when the length of the transmission is not known or where transmission speed or other conditions prevent segmenting the output into known length blocks.

The functional syntax diagrams permit the use of this element only as the final element in a <TERMINATED RESPONSE MESSAGE> (see figure 8-4).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

8.7.10.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR INDÉFINIE> est définie par



Où <octet de donnée> est défini en 7.7.6.2.

NOTE - Le message de FIN CEI 625-1 a la double fonction de terminer cet élément et de terminer aussi le <MESSAGE DE RÉPONSE>. Il est envoyé seulement une fois avec le dernier octet de donnée de bloc indéfini. Le NL est présent pour la compatibilité avec le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>.

8.7.10.3 Règles

Ce format ne doit être utilisé que lorsque les autres formats d'éléments de <DONNÉES DE RÉPONSE> ne sont pas utilisables.

8.7.11 *<DONNÉES DE RÉPONSE DE 7 BITS ISO ARBITRAIRE>*

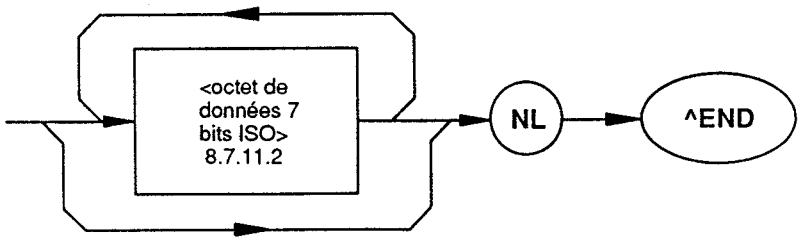
8.7.11.1 Fonction

L'élément de <DONNÉES DE RÉPONSE DE 7 BITS ISO ARBITRAIRE> permet aux appareils de répondre à un texte de 7 bits ISO non délimité. Dans certains cas, cet élément facilite le traitement par le contrôleur de l'information de l'appareil, comme un texte affiché.

Les diagrammes de syntaxe fonctionnelle ne permettent l'utilisation de cet élément qu'à la fin d'un <MESSAGE DE RÉPONSE TERMINÉ> (voir figure 8-4).

8.7.11.2 Syntaxe de codage

Une <DONNÉE DE RÉPONSE 7 BITS ISO ARBITRAIRE> est définie par

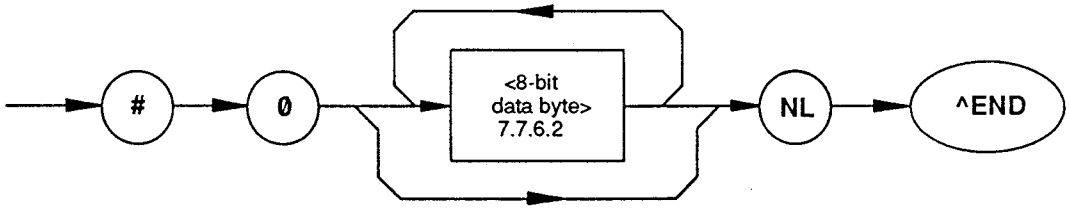


Où <octet de données 7 bits ISO> représente n'importe quel octet de données codées ISO, sauf NL (0A, 10 en décimal).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

8.7.10.2 Encoding syntax

An <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> is defined as



Where <8-bit data byte> is defined in 7.7.6.2.

NOTE - The IEC 625-1 END message serves the dual function of terminating this element as well as terminating the <RESPONSE MESSAGE>. It is only sent once with the last byte of the indefinite block data. The NL is present for consistency with the <RESPONSE MESSAGE TERMINATOR>.

8.7.10.3 Rules

This format should only be used where other <RESPONSE DATA> element formats are unworkable.

8.7.11 *<ARBITRARY ISO 7-bit RESPONSE DATA>*

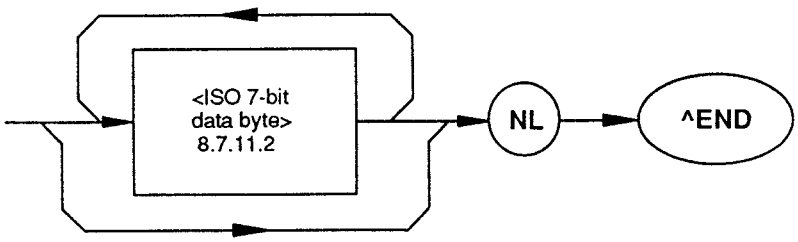
8.7.11.1 Function

The <ARBITRARY ISO 7-bit RESPONSE DATA> element allows devices to respond with undelimited ISO 7-bit text. In some cases this element facilitates controller handling of device information, such as displayed text.

The functional syntax diagrams permit the use of this element only at the end of a <TERMINATED RESPONSE MESSAGE> (see figure 8-4).

8.7.11.2 Encoding syntax

An <ARBITRARY ISO 7-bit RESPONSE DATA> is defined as



Where <ISO 7-bit data byte> represents any ISO 7-bit encoded data byte except NL (0A, 10 decimal).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

NOTES

1 Le message de FIN fournit une fin sans ambiguïté à un élément qui contient des caractères ISO arbitraires.

2 Le message de fin CEI 625-1 a la double fonction de terminer cet élément et de terminer aussi le <MESSAGE DE RÉPONSE>. Il est envoyé seulement une fois avec le dernier octet de données de bloc indéfini. Le NL est présent pour la compatibilité avec le <CARACTÈRE D'ARRÊT DE MESSAGE DE RÉPONSE>.

9 Codage des données de message

Cette partie de la CEI 625 précise les formes spécifiques de codage pour les messages spécifiques aux appareils. Le code de 7 bits ISO est le code utilisé dans tout le document pour la majorité des définitions syntaxiques et sémantiques. Le codage sur 8 bits est autorisé dans des éléments de messages de blocs spéciaux: <DONNÉES DE PROGRAMMATION ARBITRAIRE>, voir 7.7.6, <DONNÉES DE RÉPONSE ARBITRAIRE DE LONGUEUR DÉFINIE>, voir 8.7.9, et <DONNÉES DE RÉPONSE ARBITRAIRE DE LONGUEUR INDÉFINIE>, voir 8.7.10.

9.1 *Codes ISO de 7 bits*

Le code de 7 bits ISO 646 (1991) est le code courant de représentation des données pour la communication de messages spécifiques à l'appareil, comme décrit tout au long des articles 7 et 8.

Les bits du code de 7 bits ISO devront être assignés aux lignes du signal DIO comme cela est indiqué au tableau 9-1. Le tableau du code de 7 bits ISO est représenté au tableau 9-2.

Tableau 9-1 – Code de 7 bits ISO/Correspondance des lignes

Code ISO:	**	B7	B6	B5	B4	B3	B2	B1
Ligne DIO:	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1
** Le bit 8 (DIO8) devra être envoyé FAUX et ignoré à la réception par les appareils.								

9.2 *Codes de nombre entier 8 bits binaire*

Les codes suivants sont les formats recommandés pour une utilisation avec des éléments de réponse et de programmation de bloc, voir 7.7.6, 8.7.9, et 8.7.10. Bien que les autres codes ne soient pas interdits par cette norme, on utilisera de préférence les codes suivants.

9.2.1 *Relations entre la ligne de données et l'ordre des octets*

Les codes de bits binaires utilisent des bits de poids forts et faibles pour représenter directement des nombres binaires (base 2). Chaque nombre binaire est envoyé dans une zone de données consistant en autant d'octets que désiré. Le contenu des zones de données devra être envoyé octet de poids fort en premier.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

NOTES

1 The END message provides an unambiguous termination to an element that contains arbitrary ISO 7-bit characters.

2 The IEC 625-1 END message serves the dual function of terminating this element as well as terminating the <RESPONSE MESSAGE>. It is only sent once with the last byte of the indefinite block data. The NL is present for consistency with the <RESPONSE MESSAGE TERMINATOR>.

9 Message data coding

This clause of IEC 625 specifies specific forms of coding for device-specific messages. The ISO 7-bit code is the code used throughout this document for the majority of the syntactic and semantic definitions. Eight-bit coding is allowed in special block message elements: <ARBITRARY BLOCK PROGRAM DATA>, see 7.7.6, <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>, see 8.7.9, and <INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>, see 8.7.10.

9.1 *ISO 7-bit codes*

ISO 646 (1991) ISO 7-bit code is the common data representation code for communication of device-specific messages as described throughout clauses 7 and 8.

The ISO 7-bit code bits shall be assigned to the DIO signal lines as shown in table 9-1. The ISO 7-bit code chart is shown in table 9-2.

Table 9-1 – ISO 7-bit code/line relationship

ISO 7-bit code:	**	B7	B6	B5	B4	B3	B2	B1
DIO line:	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1
** Bit 8 (DIO8) shall be sent FALSE and ignored on reception by **devices.**								

9.2 *Binary 8-bit integer codes*

The following codes are the recommended formats for use with block program and response elements, see 7.7.6, 8.7.9, and 8.7.10. Although other codes are not prohibited by this standard, the use of the following codes is preferred.

9.2.1 *Byte order and data line relationships*

Bit binary codes utilize high and low bits to directly represent binary (radix 2) numbers. Each binary number is sent in a data field consisting of as many 8-bit bytes as desired. The contents of the data field shall be sent most significant byte first.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Une zone de données typique est illustrée ci-dessous.

Séquence:	|← octet 1 →|	|← octet 2 →|
	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1
	^	^
	MSB (most significant bit) (Position du bit de poids le plus fort)	LSB (least significant bit) (Position du bit de poids le plus faible)

Les codes binaires devront avoir une étendue de 1, 2, 3, 4 ou 8 octets.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

A typical data field is illustrated below.

Sequence:	|← byte 1 →|	|← byte 2 →|
	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1
	^	^
	MSB (most significant bit position)	LSB (least significant bit position)

Binary codes shall be 1, 2, 4 or 8 bytes wide.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Tableau 9-2 – Tableau de code de 7 bits ISO

La zone en trait discontinu correspond à la gamme des codes définis par <espace blanc>.

B7 B6 B5 / BITS / B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTRÔLE		NOMBRES SYMBOLES		MAJUSCULE		MINUSCULE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 SP 32 20	60 0 30 48	100 @ 40 64	120 P 50 80	140 ` 60 96	160 p 70 112
0 0 0 1	1 GTL SOH 1 1	21 LLO DC1 11 17	41 ! 21 33	61 1 31 49	101 A 41 65	121 Q 51 81	141 a 61 97	161 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 " 22 34	62 2 32 50	102 B 42 66	122 R 52 82	142 b 62 98	162 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 # 23 35	63 3 33 51	103 C 43 67	123 S 53 83	143 c 63 99	163 s 73 115
0 1 0 0	4 SDC EOT 4 4	24 DCL DC4 14 20	44 $ 24 36	64 4 34 52	104 D 44 68	124 T 54 84	144 d 64 100	164 t 74 116
0 1 0 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 % 25 37	65 5 35 53	105 E 45 69	125 U 55 85	145 e 65 101	165 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 & 26 38	66 6 36 54	106 F 46 70	126 V 56 86	146 f 66 102	166 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 ' 27 39	67 7 37 55	107 G 47 71	127 W 57 87	147 g 67 103	167 w 77 119
1 0 0 0	10 GET BS 8 8	30 SPE CAN 18 24	50 (28 40	70 8 38 56	110 H 48 72	130 X 58 88	150 h 68 104	170 x 78 120
1 0 0 1	11 TCT HT 9 9	31 SPD EM 19 25	51) 29 41	71 9 39 57	111 I 49 73	131 Y 59 89	151 i 69 105	171 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 * 2A 10 42	72 : 3A 26 58	112 J 4A 10 74	132 Z 5A 26 90	152 j 6A 106	172 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 + 2B 11 43	73 ; 3B 27 59	113 K 4B 11 75	133 [5B 27 91	153 k 6B 107	173 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 , 2C 12 44	74 < 3C 28 60	114 L 4C 12 76	134 \ 5C 28 92	154 l 6C 108	174 | 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 - 2D 13 45	75 = 3D 29 61	115 M 4D 13 77	135] 5D 29 93	155 m 6D 109	175 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 . 2E 14 46	76 > 3E 30 62	116 N 4E 14 78	136 ^ 5E 30 94	156 n 6E 110	176 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 / 2F 15 47	77 UNL ? 3F 63	117 O 4F 15 79	137 UNT _ 5F 95	157 o 6F 15 111	177 DEL (RUBOUT) 7F 127

CLÉ

Instructions adressées | Instructions Universelles | Adresses écouteur | Adresses parleur | Adresses ou instructions secondaires

octal 25 PPU NAK
hex 15 21

Code CEI 625-1
Caractère de 7 bits ISO
decimal

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Table 9-2 – ISO 7-bit code chart

The dashed area corresponds to the range of codes defined as <white space>.

B7 B6 B5 / BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 SP 20 32	60 0 30 48	100 @ 40 64	120 P 50 80	140 ` 60 96	160 p 70 112
0 0 0 1	1 GTL SOH 1 1	21 LLO DC1 11 17	41 ! 21 33	61 1 31 49	101 A 41 65	121 Q 51 81	141 a 61 97	161 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 " 22 34	62 2 32 50	102 B 42 66	122 R 52 82	142 b 62 98	162 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 # 23 35	63 3 33 51	103 C 43 67	123 S 53 83	143 c 63 99	163 s 73 115
0 1 0 0	4 SDC EOT 4 4	24 DCL DC4 14 20	44 $ 24 36	64 4 34 52	104 D 44 68	124 T 54 84	144 d 64 100	164 t 74 116
0 1 0 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 % 25 37	65 5 35 53	105 E 45 69	125 U 55 85	145 e 65 101	165 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 & 26 38	66 6 36 54	106 F 46 70	126 V 56 86	146 f 66 102	166 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 ' 27 39	67 7 37 55	107 G 47 71	127 W 57 87	147 g 67 103	167 w 77 119
1 0 0 0	10 GET BS 8 8	30 SPE CAN 18 24	50 (28 40	70 8 38 56	110 H 48 72	130 X 58 88	150 h 68 104	170 x 78 120
1 0 0 1	11 TCT HT 9 9	31 SPD EM 19 25	51) 29 41	71 9 39 57	111 I 49 73	131 Y 59 89	151 i 69 105	171 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 * 2A 42	72 : 3A 58	112 J 4A 74	132 Z 5A 90	152 j 6A 106	172 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 + 2B 43	73 ; 3B 59	113 K 4B 75	133 [5B 91	153 k 6B 107	173 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 , 2C 44	74 < 3C 60	114 L 4C 76	134 \ 5C 92	154 l 6C 108	174 | 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 - 2D 45	75 = 3D 61	115 M 4D 77	135] 5D 93	155 m 6D 109	175 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 . 2E 46	76 > 3E 62	116 N 4E 78	136 ^ 5E 94	156 n 6E 110	176 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 / 2F 47	77 UNL ? 3F 63	117 O 4F 79	137 UNT _ 5F 95	157 o 6F 111	177 DEL (RUBOUT) 7F 127

KEY

ADRS CMDS | UNIV CMDS | LISTEN ADDRESSES | TALK ADDRESSES | SECONDARY ADDRESSES OR COMMANDS

octal 25 PPU / hex 15 NAK 21 — IEC 625-1 code / ISO 7-bit character / decimal

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

9.2.2 *Code de nombre entier binaire*

Le nombre entier binaire est supposé cadré à droite dans la zone de donnée avec l'emplacement de la virgule à droite de l'emplacement du bit de poids le plus faible. On suppose la représentation en complément à 2 et que le contenu du bit de poids le plus fort est interprété comme le bit de signe. Si le mot de donnée de l'appareil n'est pas aussi large que la zone, le bit (de signe) de poids le plus fort devra alors être étendu pour remplir la zone.

9.2.3 *Code de nombre entier binaire sans signe*

Le nombre entier binaire sans signe est cadré à droite comme décrit ci-dessus. Le contenu de la zone de donnée du bit de poids le plus fort n'est pas interprété comme un signe, mais comme le bit de poids le plus fort d'un nombre positif. Donc, chaque bit de donnée de poids le plus fort inutilisé dans la zone de donnée doit être rempli de zéros.

9.3 *Code en virgule flottante binaire*

La représentation en virgule flottante incluse ici est un sous-ensemble de la CEI 559 sur les virgules flottantes arithmétiques.

9.3.1 *Zones de code de la virgule flottante*

Les nombres à virgule flottante devront être représentés par trois zones. Les zones sont:

1) La zone du signe
2) La zone de l'exposant
3) La zone de fraction

La taille des zones dépend de la précision du nombre.

Pour des nombres avec précision simple:

Largeur de la zone du signe	1 bit	*E*(max)	+127
Largeur de la zone de l'exposant	8 bits	*E*(min)	–126
Largeur de la zone de fraction	23 bits	Polarisation de l'exposant	+127
Largeur totale	32 bits		

Pour des nombres avec précision double:

Largeur de la zone du signe	1 bit	*E*(max)	+1023
Largeur de la zone de l'exposant	11 bits	*E*(min)	–1022
Largeur de la zone de fraction	52 bits	Polarisation de l'exposant	+1023
Largeur totale	64 bits		

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

9.2.2 *Binary integer code*

Binary integer is assumed to be right-justified in the data field, with the radix point to the right of the least significant bit position. 2's complement representation is assumed and the contents of the most significant bit position is interpreted as the sign bit. If the device data word is not as wide as the field, then the most significant (sign) bit shall be extended to fill the field.

9.2.3 *Binary unsigned integer code*

Binary unsigned integer is right-justified as described above. The contents of the data field's most significant bit is not interpreted as a sign, but as the most significant bit of a positive number. Thus any most-significant unused data bits in the data field should be zero filled.

9.3 *Binary floating point code*

The floating point representation included here is a subset of the IEC 559 on Floating Point Arithmetic.

9.3.1 *Floating point code fields*

Floating point numbers shall be represented by three fields. The fields are:

1) The sign field
2) The exponent field
3) The fraction field

The size of the fields depends on the precision of the number.

For single precision numbers:

Sign field width	1 bit	*E*(max)	+127
Exponent field width	8 bits	*E*(min)	–126
Fraction field width	23 bits	Exponent bias	+127
Total width	32 bits		

For double precision numbers:

Sign field width	1 bit	*E*(max)	+1023
Exponent field width	11 bits	*E*(min)	–1022
Fraction field width	52 bits	Exponent bias	+1023
Total width	64 bits		

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

9.3.2 *Formats basiques*

Les nombres en simple et double formats sont composés des trois zones suivantes:

1) Signe de 1 bit s (s = 0 = positif) (s = 1 = négatif)
2) Exposant polarisé $e = E$ + polarisation
3) Fraction $f = .b(1)\ b(2)\ \ldots\ b(p-1)$

où

p est le nombre de bits significatifs

b(n) = 0 ou 1

La gamme de l'exposant E non polarisé devra inclure tous les nombres entre et y compris deux valeurs E(min) et E(max), et aussi deux autres valeurs réservées E(min) – 1 pour coder ±0 et les nombres dénormalisés, et E(max) + 1 pour coder ±∞ et NaNs (Pas un symbole Nombre). Les paramètres précédents sont donnés en 9.3.1. Chaque valeur numérique qui n'est pas zéro ne possède qu'un seul codage. Les zones sont interprétées comme suit:

9.3.2.1 Simple

Un nombre X de format simple de 32 bits est divisé comme représenté en 9.3.2. La valeur v de X est déduite de ses zones constituantes de la façon suivante:

1) Si $e = 255$ et $f \neq 0$, alors v est NaNs sans tenir compte de s
2) Si $e = 255$ et $f = 0$, alors $v = (-1)^S \infty$
3) Si $0 < e < 255$, alors $v = (-1)^S 2^{e-127} (1.f)$
4) Si $e = 0$ et $f \neq 0$, alors $v = (-1)^S 2^{-126} (0.f)$
(nombres dénormalisés)
5) Si $e = 0$ et $f = 0$, alors $v = (-1)^S 0$ (zero)

9.3.2.2 Double

Un nombre X de format double de 64 bits est divisé comme représenté en 9.3.2. La valeur v de X est déduite de ses zones constituantes de la façon suivante:

1) Si $e = 2047$ et $f \neq 0$, alors v est NaNs sans tenir compte de s
2) Si $e = 2047$ et $f = 0$, alors $v = (-1)^S \infty$
3) Si $0 < e < 2047$, alors $v = (-1)^S 2^{e-1023} (1.f)$
4) Si $e = 0$ et $f \neq 0$, alors $v = (-1)^S 2^{-1022} (0.f)$
(nombres dénormalisés)
5) Si $e = 0$ et $f = 0$, alors $v = (-1)^S 0$ (zéro)

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

9.3.2 *Basic formats*

Numbers in the single and double formats are composed of the following three fields:

1) 1-bit sign s (s = 0 = positive) (s = 1 = negative)
2) Biased exponent e = E + bias
3) Fraction f = .b (1) b (2). . . b (p-1)

where

p is the number of significant bits

b(n) = 0 or 1

The range of the unbiased exponent E shall include every integer between two values E(min) and E(max), inclusive, and also two other reserved values E(min)–1 to encode ±0 and denormalized numbers, and E(max)+1 to encode ±∞ and NaNs (Not a Number symbol). The foregoing parameters are given in 9.3.1. Each non-zero numerical value has just one encoding. The fields are interpreted as follows:

9.3.2.1 Single

A 32-bit single format number X is divided as shown in 9.3.2. The value v of X is inferred from its constituent fields, thus

1) If $e = 255$ and $f \neq 0$, then v is NaNs regardless of s
2) If $e = 255$ and $f = 0$, then $v = (-1)^S \infty$
3) If $0 < e < 255$, then $v = (-1)^S 2^{e-127} (1.f)$
4) If $e = 0$ and $f \neq 0$, then $v = (-1)^S 2^{-126} (0.f)$
(denormalized numbers)
5) If $e = 0$ and $f = 0$, then $v = (-1)^S 0$ (zero)

9.3.2.2 Double

A 64-bit double format number X is divided as shown in clause 9.3.2. The value v of X is inferred from its constituent fields, thus

1) If $e = 2047$ and $f \neq 0$, then v is NaNs regardless of s
2) If $e = 2047$ and $f = 0$, then $v = (-1)^S \infty$
3) If $0 < e < 2047$, then $v = (-1)^S 2^{e-1023} (1.f)$
4) If $e = 0$ and $f \neq 0$, then $v = (-1)^S 2^{-1022} (0.f)$
(denormalized numbers)
5) If $e = 0$ and $f = 0$, then $v = (-1)^S 0$ (zero)

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

9.3.3 *Ordre de transmission*

9.3.3.1 Nombres avec précision simple

Les nombres avec précision simple doivent être transmis sur quatre octets. La transmission doit être structurée selon les relations suivantes entre les lignes de signaux DIO et les zones.

DIO	–	8	7	6	5	4	3	2	1	
		S	E	E	E	E	E	E	E	Premier octet envoyé
			^							
		MSBE								
		E	F	F	F	F	F	F	F	Deuxième octet envoyé
		^	^							
	LSBE		MSBF							
		F	F	F	F	F	F	F	F	Troisième octet envoyé
		F	F	F	F	F	F	F	F	Quatrième octet envoyé
									^	
									LSBF	

Où

MSBE (most significant bit of exponent) est le bit de poids le plus fort de l'exposant.

LSBE (least significant bit of exponent) est le bit de poids le plus faible de l'exposant.

MSBF (most significant bit of fraction) est le bit de poids le plus fort de la fraction.

LSBF (least significant bit of fraction) est le bit de poids le plus faible de la fraction.

S est le bit de signe.

E est un bit d'exposant.

F est un bit de fraction.

9.3.3.2 Nombres avec double précision

Les nombres avec double précision doivent être transmis sur huit octets. La transmission doit être structurée selon les relations suivantes entre les lignes de signaux DIO et les zones.

DIO	–	8	7	6	5	4	3	2	1	
		S	E	E	E	E	E	E	E	Premier octet envoyé
			^							
		MSBE								
		E	E	E	E	F	F	F	F	Deuxième octet envoyé
					^	^				
				LSBE		MSBF				
		F	F	F	F	F	F	F	F	Troisième à septième octet envoyé
		F	F	F	F	F	F	F	F	Dernier octet envoyé
									^	
									LSBF	

Où

MSBE (most significant bit of exponent) est le bit de poids le plus fort de l'exposant.

LSBE (least significant bit of exponent) est le bit de poids le plus faible de l'exposant.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

9.3.3 *Order of transmission*

9.3.3.1 Single precision numbers

Single precision numbers shall be transmitted in four bytes. The transmission shall be structured according to the following relationships between DIO signal lines and the fields.

DIO	–	8	7	6	5	4	3	2	1	
		S	E	E	E	E	E	E	E	First byte sent
			^							
		MSBE								
		E	F	F	F	F	F	F	F	Second byte sent
		^	^							
	LSBE		MSBF							
		F	F	F	F	F	F	F	F	Third byte sent
		F	F	F	F	F	F	F	F	Fourth byte sent
									^	
								LSBF		

Where

MSBE is the most significant bit of the exponent.

LSBE is the least significant bit of the exponent.

MSBF is the most significant bit of the fraction.

LSBF is the least significant bit of the fraction.

S is the sign bit.

E is an exponent bit.

F is a fraction bit.

9.3.3.2 Double precision numbers

Double precision numbers shall be transmitted in eight bytes. The transmission shall be structured according to the following relationships between DIO signal lines and the fields.

DIO	–	8	7	6	5	4	3	2	1	
		S	E	E	E	E	E	E	E	First byte sent
			^							
		MSBE								
		E	E	E	E	F	F	F	F	Second byte sent
					^	^				
				LSBE		MSBF				
		F	F	F	F	F	F	F	F	Third through seventh byte sent
		F	F	F	F	F	F	F	F	Last byte sent
									^	
								LSBF		

Where

MSBE is the most significant bit of the exponent.

LSBE is the least significant bit of the exponent.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

MSBF (most significant bit of fraction) est le bit de poids le plus fort de la fraction.

LSBF (least significant bit of fraction) est le bit de poids le plus faible de la fraction.

S est le bit de signe.

E est un bit d'exposant.

F est un bit de fraction.

9.3.4 *Exemple de nombre avec simple précision*

Un exemple de nombre avec simple précision pourrait être codé en utilisant ces quatre octets:

01000010	11100000	00000000	00000000
se---------	ef---------	----------	----------f

où

Le nombre donne alors:		*binaire*	*décimal*
v	$= (-1)^S 2^{e-127} (1.f)$	$s =$	0 = 0
	$= (-1)^0 2^6 (1.75)$	$e =$	10000101 = 133
	$= (64) (1.75)$	$f =$	.110 = .75
	$= 112$		

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

MSBF is the most significant bit of the fraction.

LSBF is the least significant bit of the fraction.

S is the sign bit.

E is an exponent bit.

F is a fraction bit.

9.3.4 *Example of single precision number*

An example of a single precision number could be encoded using these four bytes:

01000010	11100000	00000000	00000000
se---------	ef---------	----------	----------f

Where

The number then evaluates to:

			binary	*decimal*
v	$= (-1)^S 2^{e-127} (1.f)$	$s =$	0	= 0
	$= (-1)^0 2^6 (1.75)$	$e =$	10000101	= 133
	$= (64) (1.75)$	$f =$	.110	= .75
	$= 112$			

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10 Instructions et interrogations communes

Cet article décrit les instructions et interrogations communes. Les informations descriptives et les exigences de conformité sont incluses pour chaque instruction. Le tableau 10-1 ci-dessous fait la liste de tous les en-têtes d'instructions communes définis dans cette norme. Le tableau 10-2 énumère ces mêmes instructions regroupées suivant leur fonction. La syntaxe des instructions et interrogations suit les conventions de 7.1.

Tableau 10-1 – En-têtes d'instructions communes CEI 625-2

Mnémonique	Nom	Paragraphe
*AAD	Instruction d'acceptation d'adresse	10.1
*CAL?	Interrogation d'étalonnage	10.2
*CLS	Instruction d'effacement d'état	10.3
*DDT	Instruction de déclenchement défini par l'appareil	10.4
*DDT?	Interrogation de déclenchement défini par l'appareil	10.5
*DLF	Instruction de fonction écouteur invalidée	10.6
*DMC	Définition d'une macro-instruction	10.7
*EMC	Validation d'une macro-instruction	10.8
*EMC?	Validation d'une macro-interrogation	10.9
*ESE	Instruction de validation d'état d'événement standard	10.10
*ESE?	Interrogation de validation d'état d'événement standard	10.11
*ESR?	Interrogation de registre d'état d'événement standard	10.12
*GMC?	Lecture du contenu des macro-instructions	10.13
*IDN?	Interrogation d'identification	10.14
*IST?	Interrogation de lecture d'état individuel	10.15
*LMC?	Interrogation des macros définies	10.16
*LRN?	Interrogation de lecture de la configuration de l'appareil	10.17
OPC	Instruction d'opération achevée	10.18
*OPC?	Interrogation d'opération achevée	10.19
*OPT?	Interrogation d'identification d'option	10.20
*PCB	Instruction de retour de passage de contrôle	10.21
*PMC	Élimination des macro-instructions	10.22
*PRE	Instruction de validation de registre d'interrogation parallèle	10.23
*PRE?	Interrogation de validation de registre d'interrogation parallèle	10.24
*PSC	Instruction d'effacement d'état à la mise sous tension	10.25
*PSC?	Interrogation d'effacement d'état à la mise sous tension	10.26
*PUD	Instruction de données utilisateur protégées	10.27
*PUD?	Interrogation de données utilisateur protégées	10.28
*RCL	Instruction de rappel	10.29
*RDT	Instruction de transfert de description de ressources	10.30
*RDT?	Interrogation de transfert de description de ressources	10.31
*RST	Instruction de remise à zéro	10.32
*SAV	Instruction de sauvegarde	10.33
*SRE	Instruction de validation de demande de service	10.34
*SRE?	Interrogation de validation de demande de service	10.35
*STB?	Interrogation de lecture de l'octet d'état	10.36
*TRG	Instruction de déclenchement	10.37
*TST?	Interrogation d'auto-test	10.38
*WAI	Instruction d'attente de poursuite	10.39

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10 Common commands and queries

This clause describes common commands and queries. Descriptive information and compliance requirements are included for each command. Table 10-1 below lists alphabetically all the common command headers defined in this standard. Table 10-2 lists the same commands grouped according to function. Syntax of the commands and queries follows the conventions of 7.1.

Table 10-1 – IEC 625-2 common command headers

Mnemonic	Name	Subclause
*AAD	Accept Address Command	10.1
*CAL?	Calibration Query	10.2
*CLS	Clear Status Command	10.3
*DDT	Define Device Trigger Command	10.4
*DDT?	Define Device Trigger Query	10.5
*DLF	Disable Listener Function Command	10.6
*DMC	Define Macro Command	10.7
*EMC	Enable Macro Command	10.8
*EMC?	Enable Macro Query	10.9
*ESE	Standard Event Status Enable Command	10.10
*ESE?	Standard Event Status Enable Query	10.11
*ESR?	Standard Event Status Register Query	10.12
*GMC?	Get Macro Contents Query	10.13
*IDN?	Identification Query	10.14
*IST?	Individual Status Query	10.15
*LMC?	Learn Macro Query	10.16
*LRN?	Learn Device Setup Query	10.17
*OPC	Operation Complete Command	10.18
*OPC?	Operation Complete Query	10.19
*OPT?	Option Identification Query	10.20
*PCB	Pass Control Back Command	10.21
*PMC	Purge Macro Command	10.22
*PRE	Parallel Poll Register Enable Command	10.23
*PRE?	Parallel Poll Register Enable Query	10.24
*PSC	Power On Status Clear Command	10.25
*PSC?	Power On Status Clear Query	10.26
*PUD	Protected User Data Command	10.27
*PUD?	Protected User Data Query	10.28
*RCL	Recall Command	10.29
*RDT	Resource Description Transfer Command	10.30
*RDT?	Resource Description Transfer Query	10.31
*RST	Reset Command	10.32
*SAV	Save Command	10.33
*SRE	Service Request Enable Command	10.34
*SRE?	Service Request Enable Query	10.35
*STB?	Read Status Byte Query	10.36
*TRG	Trigger Command	10.37
*TST?	Self-Test Query	10.38
*WAI	Wait-to-Continue Command	10.39

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Tableau 10-2 – Groupes d'instructions communes CEI 625-2

Mnémonique	Groupe	Conformité	Paragraphe
*AAD	Configuration automatique	Facultative[1)]	10.1
*DLF	Configuration automatique	Facultative[1)]	10.6
*IDN?	Données du système	Obligatoire	10.14
*OPT?	Données du système	Facultative	10.20
*PUD	Données du système	Facultative	10.27
*PUD?	Données du système	Facultative	10.28
*RDT	Données du système	Facultative	10.30
*RDT?	Données du système	Facultative	10.31
*CAL?	Opérations internes	Facultative	10.2
*LRN?	Opérations internes	Facultative	10.17
*RST	Opérations internes	Obligatoire	10.32
*TST?	Opérations internes	Obligatoire	10.38
*OPC	Synchronisation	Obligatoire	10.18
*OPC?	Synchronisation	Obligatoire	10.19
*WAI	Synchronisation	Obligatoire	10.39
*DMC	Macro-instruction	Facultative[1)]	10.7
*EMC	Macro-instruction	Facultative[1)]	10.8
*EMC?	Macro-instruction	Facultative[1)]	10.9
*GMC?	Macro-instruction	Facultative[1)]	10.13
*LMC?	Macro-instruction	Facultative[1)]	10.16
*PMC	Macro-instruction	Facultative[1)]	10.22
*IST?	Interrogation parallèle	Obligatoire si PP1	10.15
*PRE	Interrogation parallèle	Obligatoire si PP1	10.23
*PRE?	Interrogation parallèle	Obligatoire si PP1	10.24
*CLS	Etat et événement	Obligatoire	10.3
*ESE	Etat et événement	Obligatoire	10.10
*ESE?	Etat et événement	Obligatoire	10.11
*ESR?	Etat et événement	Obligatoire	10.12
*PSC	Etat et événement	Facultative	10.25
*PSC?	Etat et événement	Facultative	10.26
*SRE	Etat et événement	Obligatoire	10.34
*SRE?	Etat et événement	Obligatoire	10.35
*STB?	Etat et événement	Obligatoire	10.36
*DDT	Déclenchement	Facultative si DT1	10.4
*DDT?	Déclenchement	Facultative si DT1	10.5
*TRG	Déclenchement	Obligatoire si DT1	10.37
*PCB	Contrôleur	Obligatoire si autre que C0	10.21
*RCL	Positionnements stockés	Facultative[1)]	10.29
*SAV	Positionnements stockés	Facultative[1)]	10.33

[1)] Si une des instructions dans les groupes de configuration automatique, ou de macro-instructions, ou de positionnements stockés, est implémentée, toutes les instructions de ce groupe doivent alors être implémentées.

L'article 6.1.6.1.1 exige qu'un **appareil** génère une erreur d'instruction si une instruction commune non implémentée est reçue.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Table 10-2 – IEC 625-2 common command groups

Mnemonic	Group	Compliance	Subclause
*AAD	Auto Configure	Optional[1)]	10.1
*DLF	Auto Configure	Optional[1)]	10.6
*IDN?	System Data	Mandatory	10.14
*OPT?	System Data	Optional	10.20
*PUD	System Data	Optional	10.27
*PUD?	System Data	Optional	10.28
*RDT	System Data	Optional	10.30
*RDT?	System Data	Optional	10.31
*CAL?	Internal Operations	Optional	10.2
*LRN?	Internal Operations	Optional	10.17
*RST	Internal Operations	Mandatory	10.32
*TST?	Internal Operations	Mandatory	10.38
*OPC	Synchronization	Mandatory	10.18
*OPC?	Synchronization	Mandatory	10.19
*WAI	Synchronization	Mandatory	10.39
*DMC	Macro	Optional[1)]	10.7
*EMC	Macro	Optional[1)]	10.8
*EMC?	Macro	Optional[1)]	10.9
*GMC?	Macro	Optional[1)]	10.13
*LMC?	Macro	Optional[1)]	10.16
*PMC	Macro	Optional[1)]	10.22
*IST?	Parallel Poll	Mandatory if PP1	10.15
*PRE	Parallel Poll	Mandatory if PP1	10.23
*PRE?	Parallel Poll	Mandatory if PP1	10.24
*CLS	Status and Event	Mandatory	10.3
*ESE	Status and Event	Mandatory	10.10
*ESE?	Status and Event	Mandatory	10.11
*ESR?	Status and Event	Mandatory	10.12
*PSC	Status and Event	Optional	10.25
*PSC?	Status and Event	Optional	10.26
*SRE	Status and Event	Mandatory	10.34
*SRE?	Status and Event	Mandatory	10.35
*STB?	Status and Event	Mandatory	10.36
*DDT	Trigger	Optional if DT1	10.4
*DDT?	Trigger	Optional if DT1	10.5
*TRG	Trigger	Mandatory if DT1	10.37
*PCB	Controller	Mandatory if other than C0	10.21
*RCL	Stored Settings	Optional[1)]	10.29
*SAV	Stored Settings	Optional[1)]	10.33

[1)] If any commands in either the Auto Configure, Macro, or Stored Settings groups are implemented then all the commands in that group shall be implemented.

Clause 6.1.6.1.1 requires that a **device** shall generate a Command Error if an unimplemented common command is received.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.1 **AAD, Instruction d'acceptation d'adresse*

10.1.1 *Fonction et exigences*

Cette instruction, conjointement avec le protocole du jeu d'adresses, permet au contrôleur de détecter tous les **appareils** d'adresse configurable (c'est-à-dire les **appareils** qui implémentent cette instruction) et d'affecter une adresse CEI 625-1 à chacun de ces **appareils**. Un appareil d'adresse configurable est détecté lorsque le contrôleur a achevé une recherche octet par octet de l'identificateur de l'**appareil**, après quoi l'adresse est affectée. Le protocole du jeu d'adresses provoque la répétition de cette recherche d'identificateur jusqu'à ce que tous les appareils d'adresse configurable aient été détectés. Voir 13.4.2, Exigences d'instruction commune *AAD, pour les exigences d'implémentation détaillée de cette instruction.

10.1.2 *Syntaxe*

La syntaxe pour l'instruction d'affectation d'adresse est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> immédiatement suivi par un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*AAD'.

10.1.3 *Sémantique*

Voir 13.2.2.1.

10.1.4 *Instructions communes relatives*

L'implémentation de *DLF nécessite cette instruction facultative.

10.1.5 *Conformité à la norme*

Facultative.

10.1.6 *Traitement de l'erreur*

Voir 13.2.2.1.

10.2 **CAL?, Interrogation d'étalonnage*

10.2.1 *Fonctions et exigences*

L'interrogation d'étalonnage fait exécuter à un **appareil** un auto-étalonnage interne et génère une réponse qui indique si l'**appareil** a ou n'a pas achevé l'auto-étalonnage sans erreur. Des informations supplémentaires sur des erreurs d'étalonnage peuvent être contenues dans la réponse, voir 10.2.3.

L'interrogation d'étalonnage ne devra nécessiter l'interaction d'aucun opérateur local pour fonctionner. Elle ne devra pas créer de conditions de bus qui sont des violations des normes CEI 625-1 ou CEI 625-2. Par ailleurs, la portée de l'auto-étalonnage est complètement à la discrétion du concepteur de l'appareil.

Après achèvement de *CAL?, il faut faire revenir l'**appareil** à l'état juste avant le cycle d'étalonnage, ou une autre condition d'exploitation spécifiée, précisée dans la documentation, voir 4.9.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.1 **AAD, Accept Address command*

10.1.1 *Function and requirements*

This command, in conjunction with the Address Set protocol, allows the controller to detect all address-configurable **devices** (i.e. **devices** which implement this command) and assign an IEC 625-1 address to each of those **devices**. An address-configurable device is detected when the controller has completed a byte-by-byte search of the **device's** identifier after which time the address is assigned. The Address Set protocol causes this identifier search to be executed repeatedly until all address-configurable devices have been detected. See 13.4.2, *AAD common command requirements, for detailed implementation requirements of this command.

10.1.2 *Syntax*

The syntax for the Assign Address command is defined as a <COMMAND PROGRAM HEADER> followed immediately by a <PROGRAM MESSAGE TERMINATOR>. The <COMMAND PROGRAM HEADER> is defined as '*AAD'.

10.1.3 *Semantics*

See 13.2.2.1.

10.1.4 *Related common commands*

*DLF – Implementation requires this optional command.

10.1.5 *Standard compliance*

Optional

10.1.6 *Error handling*

See 13.2.2.1.

10.2 **CAL?, Calibration query*

10.2.1 *Function and requirements*

The Calibration query causes a **device** to perform an internal self-calibration and generates a response that indicates whether or not the **device** completed the self-calibration without error. Additional information about any calibration errors may be contained in the response, see 10.2.3.

The Calibration query shall not require any local operator interaction to function. It shall not create bus conditions that are violations to the IEC 625-1 or IEC 625-2 standards. Otherwise, the scope of the self-calibration is completely at the discretion of the device designer.

Upon completion of *CAL?, the **device** should be returned to the state just prior to the calibration cycle, or other specified operational condition stated in the documentation, see 4.9.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.2.2 *Structure d'interrogation*

10.2.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation d'étalonnage est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*CAL?'.

10.2.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation d'étalonnage est définie comme une simple <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> sera dans la gamme –32767 à 32767.

10.2.3 *Sémantique de réponse*

La valeur zéro indique que l'étalonnage s'est achevé sans aucune erreur détectée.

Une valeur différente de zéro indique que l'étalonnage ne s'est pas achevé ou s'est achevé avec des erreurs détectées.

La sémantique de réponse différente de zéro doit être complètement à la discrétion du concepteur de l'appareil.

10.2.4 *Instructions communes relatives*

Aucune.

10.2.5 *Conformité à la norme*

Facultative, ne nécessite pas d'autres instructions facultatives.

10.3 **CLS, Instruction d'effacement d'état*

10.3.1 *Fonction et prescriptions*

L'instruction d'effacement d'état efface les structures de données d'état, voir 11.1.2, et impose à l'appareil l'état inactif d'instruction "opération achevée" et l'état inactif d'interrogation "opération achevée", voir 12.5.2 et 12.5.3.

Si l'instruction d'effacement d'état suit immédiatement un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>, la file d'attente de sortie et le bit MAV seront effacés, étant donné que n'importe quel nouveau <MESSAGE DE PROGRAMMATION> après un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> efface la file d'attente de sortie, voir 6.3.

10.3.2 *Syntaxe*

La syntaxe pour l'instruction d'effacement d'état est définie comme uniquement un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*CLS'.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.2.2 *Query structure*

10.2.2.1 Query syntax

The syntax for the Calibration query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*CAL?'.

10.2.2.2 Response syntax

The response syntax for the Calibration query is defined as a single <NR1 NUMERIC RESPONSE DATA>. The <NR1 NUMERIC RESPONSE DATA> shall be in the range –32767 to 32767.

10.2.3 *Response semantics*

A value of zero indicates the calibration completed without any detected errors.

A value not equal to zero indicates the calibration did not complete or completed with errors detected.

The semantics of the non-zero response shall be completely at the discretion of the device designer.

10.2.4 *Related common commands*

None.

10.2.5 *Standard compliance*

Optional, requires no other optional commands.

10.3 **CLS, Clear Status command*

10.3.1 *Function and requirements*

The Clear Status command clears status data structures, see 11.1.2, and forces the device to the Operation Complete Command Idle State and the Operation Complete Query Idle State, see 12.5.2 and 12.5.3.

If the Clear Status command immediately follows a <PROGRAM MESSAGE TERMINATOR>, the Output Queue and the MAV bit will be cleared since any new <PROGRAM MESSAGE> after a <PROGRAM MESSAGE TERMINATOR> clears the Output Queue, see 6.3.

10.3.2 *Syntax*

The syntax for the Clear Status command is defined as only a <COMMAND PROGRAM HEADER>. The <COMMAND PROGRAM HEADER> is defined as '*CLS'.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.3.3 *Sémantique*

Non applicable.

10.3.4 *Instructions communes relatives*

Aucune.

10.3.5 *Conformité à la norme*

Obligatoire.

10.4 **DDT, Instruction de déclenchement défini par l'appareil*

10.4.1 *Fonction et prescriptions*

L'instruction de déclenchement définie par l'appareil enregistre une séquence d'instructions qui est exécutée lorsqu'un message d'interface CEI 625-1 de déclenchement d'exécution de groupe (GET) ou une instruction commune *TRG sont reçus, voir 10.37. L'instruction commune *RST mettra la séquence d'instructions dans un état défini par l'appareil, voir 10.32.

10.4.2 *Syntaxe*

La syntaxe pour l'instruction de déclenchement définie par l'appareil est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION DE BLOC ARBITRAIRE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*DDT'.

10.4.3 *Sémantique*

La <DONNÉE DE PROGRAMMATION DE BLOC ARBITRAIRE> contient soit la séquence d'éléments d'<UNITÉ DE MESSAGES DE PROGRAMMATION> à exécuter, soit une zone de données de longueur zéro. Une zone de longueur zéro indique qu'aucune action ne sera entreprise lorsqu'un message d'interface CEI 625-1 de déclenchement d'exécution de groupe (GET) ou une instruction commune *TRG, voir 10.37, sont reçus.

10.4.4 *Instructions communes relatives*

*DDT? est l'interrogation associée.

10.4.5 *Conformité à la norme*

Facultative mais nécessite le sous-ensemble DT1.

10.4.6 *Traitement de l'erreur*

10.4.6.1 Bloc trop long

Une erreur d'exécution, voir 6.1.7 et 11.5.1.1.5, devra être indiquée si le bloc est trop long pour être accepté par l'**appareil**. L'exigence de longueur maximum doit être spécifiée dans la documentation de l'appareil.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.3.3 *Semantics*

Not applicable.

10.3.4 *Related common commands*

None.

10.3.5 *Standard compliance*

Mandatory.

10.4 **DDT, Define Device Trigger command*

10.4.1 *Function and requirements*

The Define Device Trigger command stores a command sequence which is executed when a group execute trigger (GET) IEC 625-1 interface message or a *TRG common command is received, see 10.37. The *RST common command shall set the command sequence to a device-defined state, see 10.32.

10.4.2 *Syntax*

The syntax for the Define Device Trigger command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by an <ARBITRARY BLOCK PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*DDT'.

10.4.3 *Semantics*

The <ARBITRARY BLOCK PROGRAM DATA> contains either the sequence of <PROGRAM MESSAGE UNIT> elements to be executed or a zero-length data field. A zero-length field indicates that no action will be taken when a group execute trigger (GET) IEC 625-1 interface message or a *TRG common command, see 10.37, is received.

10.4.4 *Related common commands*

*DDT? – This is the companion query.

10.4.5 *Standard compliance*

Optional but requires the DT1 subset.

10.4.6 *Error handling*

10.4.6.1 Oversized block

An Execution Error, see 6.1.7 and 11.5.1.1.5, shall be reported if the block is too long for the **device** to accept. The maximum length requirement shall be specified in the device's documentation.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.4.6.2 Erreurs de données de bloc

N'importe quelle erreur d'instruction ou d'exécution contenue dans le bloc peut être soit détectée immédiatement, soit différée jusqu'à ce que la macro-instruction soit exécutée.

10.4.6.3 Récurrence

Si la séquence d'instructions mémorisées contient *TRG ou l'étiquette d'une macro-instruction validée dont l'expansion conduit à l'exécution de *TRG, il se produira alors une récurrence. Le programme d'application ne peut arrêter cette récurrence qu'en envoyant DCL ou SDC. L'**appareil** peut soit permettre que la récurrence se produise indéfiniment, soit signaler une erreur d'exécution à un niveau de récurrence défini par l'appareil. La façon dont la récurrence est traitée doit être spécifiée dans la documentation de l'appareil. voir 4.9.

10.4.7 *Exemple*

*DDT #217TRIG WFM; MEASWFM?

Un GET déclenchera immédiatement un signal et exécutera une interrogation qui place le signal acquis dans la file d'attente de sortie de l'**appareil**.

10.5 **DDT?, Interrogation de déclenchement défini par l'appareil*

10.5.1 *Fonction et prescriptions*

L'interrogation de déclenchement défini par l'appareil permet au programmeur d'examiner la séquence d'instructions qui sera exécutée lorsqu'un déclenchement d'exécution de groupe ou une instruction *TRG sont reçus.

10.5.2 *Structure d'interrogation*

10.5.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de déclenchement défini par l'appareil est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*DDT?'.

10.5.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de déclenchement défini par l'appareil est définie comme un élément de <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE>.

10.5.3 *Sémantique de réponse*

L'élément de <DONNÉES DE RÉPONSE DE BLOC ARBITAIRE DE LONGUEUR DÉFINIE> contient la séquence d'éléments d'<UNITÉ DE MESSAGES DE PROGRAMMATION> qui sont exécutés lorsqu'un GET ou un *TRG est reçu.

La condition de "pas de séquence d'instructions enregistrée" aboutit à une réponse de bloc de longueur zéro.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.4.6.2 Block Data Errors

Any Command or Execution Errors contained in the block may be either detected immediately or deferred until the macro is executed.

10.4.6.3 Recursion

If the stored command sequence contains *TRG or the label of an enabled macro whose expansion leads to the execution of *TRG then recursion will occur. The application program can only stop this recursion by sending DCL or SDC. The **device** may either allow recursion to take place indefinitely or report an Execution Error at a device-defined level of recursion. The manner in which recursion is handled shall be stated in the device's documentation. See 4.9.

10.4.7 *Example*

*DDT #217TRIG WFM; MEASWFM?

A GET will immediately trigger a waveform and perform a query which places the acquired waveform into the **device's** Output Queue.

10.5 **DDT?, Define Device Trigger query*

10.5.1 *Function and requirements*

The Define Device Trigger query allows the programmer to examine the command sequence which will be executed when a group execute trigger or *TRG command is received.

10.5.2 *Query structure*

10.5.2.1 Query syntax

The syntax for the Define Device Trigger query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*DDT?'.

10.5.2.2 Response syntax

The response syntax for the Define Device Trigger query is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

10.5.3 *Response semantics*

The <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element contains the sequence of <PROGRAM MESSAGE UNIT> elements which are executed when a GET or *TRG is received.

The condition of "no command sequence stored" results in a response of a zero-length block.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.5.4 *Commandes communes relatives*

L'implémentation de "*DDT" nécessite cette instruction facultative. Ceci est l'instruction associée.

10.5.5 *Conformité à la norme*

Facultative mais nécessite le sous-ensemble DT1.

10.6 **DLF, Instruction de fonction écouteur invalidée*

10.6.1 *Fonction et prescriptions*

L'instruction de fonction écouteur invalidée fait qu'un **appareil** cesse d'être un écouteur (changement pour le sous-ensemble L0). Si cette instruction est le premier message spécifique à l'appareil reçu après que l'**appareil** a quitté DCAS, l'**appareil** devra cesser d'être un écouteur dans les 100 millisecondes après l'acceptation du <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>. Voir 13.4.1, Prescriptions d'instruction commune *DLF, pour les prescriptions d'implémentation détaillée de cette instruction. Un message DCL ultérieur devra rétablir la capacité écouteur. L'**appareil** reprendra l'écoute en moins de 100 millisecondes après être entré en DCAS.

10.6.2 *Syntaxe*

La syntaxe pour l'instruction de fonction écouteur invalidée est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> immédiatement suivi par un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par "*DLF".

10.6.3 *Sémantique*

L'instruction *DLF doit être immédiatement suivie par <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> sans paramètres.

10.6.4 *Instructions communes relatives*

L'implémentation de *AAD nécessite cette instruction facultative.

10.6.5 *Conformité à la norme*

Facultative.

10.6.6 *Traitement de l'erreur*

Voir 13.2.1 pour de plus amples informations.

10.7 **DMC, Définition d'une macro-instruction*

10.7.1 *Fonction et prescriptions*

La définition d'une macro-instruction permet au programmeur d'affecter une séquence de zéro ou plusieurs éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> à une étiquette de macro. La séquence est exécutée lorsque l'étiquette est reçue comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.5.4 *Related common commands*

*DDT – Implementation requires this optional command. This is the companion command.

10.5.5 *Standard compliance*

Optional but requires the DT1 subset.

10.6 **DLF, Disable Listener Function command*

10.6.1 *Function and requirements*

The Disable Listener Function command causes a **device** to cease being a listener (change to L0 subset). If this command is the first device-specific message received after the **device** leaves DCAS, the **device** shall cease being a listener within 100 milliseconds after the acceptance of the <PROGRAM MESSAGE TERMINATOR>. See 13.4.1, *DLF Common Command requirements, for detailed implementation requirements of this command. A subsequent DCL message shall restore listener capability. The **device** shall resume listening within 100 milliseconds after entering DCAS.

10.6.2 *Syntax*

The syntax for the Disable Listener Function command is defined as a <COMMAND PROGRAM HEADER> followed immediately by a <PROGRAM MESSAGE TERMINATOR>. The <COMMAND PROGRAM HEADER> is defined as '*DLF'.

10.6.3 *Semantics*

The *DLF command shall be immediately followed by <PROGRAM MESSAGE TERMINATOR> with no parameters.

10.6.4 *Related common commands*

*AAD – Implementation requires this optional command.

10.6.5 *Standard compliance*

Optional.

10.6.6 *Error handling*

See 13.2.1 for further information.

10.7 **DMC, Define Macro command*

10.7.1 *Function and requirements*

The Define Macro command allows the programmer to assign a sequence of zero or more <PROGRAM MESSAGE UNIT> elements to a macro label. The sequence is executed when the label is received as a <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER>.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.7.2 *Syntaxe*

La syntaxe pour la définition d'une macro-instruction est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par une <DONNÉE DE PROGRAMMATION DE CHAÎNE> suivie par un <SÉPARATEUR DE DONNÉES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION DE BLOC ARBITRAIRE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*DMC'.

10.7.3 *Sémantique*

La <DONNÉE DE PROGRAMMATION DE CHAÎNE> contient l'étiquette de macro. L'étiquette devra être un <en-tête d'instruction de programmation simple>, ou un <en-tête d'instruction de programmation complexe>, ou un <en-tête d'interrogation de programmation simple>, ou un <en-tête d'interrogation de programmation complexe>. Toutefois, l'étiquette de macro ne devra pas être un <en-tête d'instruction de programmation commune>, voir 7.6.1.2, ou un <en-tête d'interrogation de programmation commune>, voir 7.6.2.2. Noter qu'un <espace blanc> avant le premier caractère de la <DONNÉE DE PROGRAMMATION DE CHAÎNE> ne sera pas considéré comme faisant partie de l'étiquette de macro.

L'étiquette de macro peut être pareille à un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> spécifiques à l'appareil. Dans ce cas, à condition que les macro-instructions soient validées, voir 10.8, l'étiquette de macro reçue dans le cadre d'un <MESSAGE DE PROGRAMMATION> sera exécutée comme une expansion de macro, et non l'instruction du même nom spécifique à l'appareil. Cette instruction spécifique à l'appareil peut encore être exécutée en invalidant les macro-instructions.

La <DONNÉE DE PROGRAMMATION DE BLOC ARBITRAIRE> contient la séquence des éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> étiquetés.

Des paramètres peuvent être entrés dans la séquence en cours d'exécution. Les emplacements pour les paramètres apparaissent dans la séquence avec un signe dollar (code de 7 bits ISO 24, 36 en décimal) suivi par un chiffre simple dans la gamme de 1 à 9 inclus (codes de 7 bits ISO 31-39, 49-57 en décimal). Le premier élément de <DONNÉES DE PROGRAMMATION> suivant l'étiquette de macro, lorsqu'il est utilisé comme en-tête, est substitué à l'emplacement de paramètre étiqueté $1. Le second élément de <DONNÉES DE PROGRAMMATION> est substitué à $2. Ce processus de substitution se poursuit jusqu'à neuf paramètres.

L'emplacement doit apparaître dans la séquence comme s'il était un élément de <DONNÉE DE PROGRAMMATION> complet. Voir 7.3.2 pour une description des endroits où des éléments de <DONNÉES DE PROGRAMMATION> sont permis dans une <UNITÉ DE MESSAGE DE PROGRAMMATION>. 7.7.7 décrit comment des <DONNÉES DE PROGRAMMATION> peuvent apparaître dans une <expression>.

10.7.4 *Instructions communes relatives*

L'implémentation de *PMC nécessite cette instruction facultative.

L'implémentation de *GMC? nécessite cette interrogation facultative.

L'implémentation de *LMC nécessite cette instruction facultative.

L'implémentation de *EMC nécessite cette instruction facultative.

L'implémentation de *EMC? nécessite cette interrogation facultative.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.7.2 *Syntax*

The syntax for the Define Macro command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <STRING PROGRAM DATA> followed by a <PROGRAM DATA SEPARATOR> followed by an <ARBITRARY BLOCK PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*DMC'.

10.7.3 *Semantics*

The <STRING PROGRAM DATA> contains the macro label. The macro label shall be either a <simple command program header>, a <compound command program header>, a <simple query program header>, or a <compound query program header>. The macro label shall not, however, be either a <common command program header>, see 7.6.1.2, or a <common query program header>, see 7.6.2.2. Note that any <white space> before the first character of the <STRING PROGRAM DATA> shall not be considered to be a part of the macro label.

The macro label may be the same as a device-specific <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER>. In this case, provided macros are enabled, see 10.8, the macro label received within a <PROGRAM MESSAGE> shall be executed as the macro expansion, not the device-specific command of the same name. That device-specific command may still be executed by disabling macros.

The <ARBITRARY BLOCK PROGRAM DATA> contains the sequence of <PROGRAM MESSAGE UNIT> elements being labelled.

Parameters may be passed to the sequence during execution. Placeholders for parameters appear in the sequence as a dollar sign (ISO 7-bit code 24, 36 decimal) followed by a single digit in the range 1 to 9 inclusive (ISO 7-bit codes 31-39, 49-57 decimal). The first <PROGRAM DATA> element following the macro label, when used as a header, is substituted for the parameter placeholder labelled $1. The second <PROGRAM DATA> element is substituted for $2. This substitution process is continued for up to nine parameters.

The placeholder must appear in the sequence as if it were a complete <PROGRAM DATA> element. See 7.3.2 for a description of where <PROGRAM DATA> elements are permitted in a <PROGRAM MESSAGE UNIT>. 7.7.7 describes how <PROGRAM DATA> elements may appear in an <expression>.

10.7.4 *Related common commands*

*PMC – Implementation requires this optional command.

*GMC? – Implementation requires this optional query.

*LMC – Implementation requires this optional command.

*EMC – Implementation requires this optional command.

*EMC? – Implementation requires this optional query.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.7.5 *Conformité à la norme*

Facultative.

10.7.6 *Traitement de l'erreur*

10.7.6.1 Bloc trop long

Une erreur d'exécution, voir 6.1.7 et 11.5.1.1.5, devra être indiquée si le bloc est trop long pour être accepté par l'**appareil**.

La prescription de longueur maximum est spécifique à l'appareil et devra être spécifiée dans la documentation de l'appareil.

10.7.6.2 Erreurs d'étiquette de macro

Un **appareil** devra indiquer une erreur d'exécution si l'étiquette de macro est trop longue pour être acceptée par l'**appareil**.

La longueur maximum de l'étiquette est spécifique à l'appareil et devra être spécifiée dans la documentation de l'appareil.

L'**appareil** devra indiquer une erreur d'exécution si l'étiquette de macro n'adhère pas à la syntaxe d'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> ou d'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> dans 7.6.

10.7.6.3 Erreurs de données de bloc

Toute erreur d'instruction ou d'exécution contenue dans le bloc peut être soit détectée immédiatement, soit différée jusqu'à ce que la macro-instruction soit exécutée.

10.7.6.4 Redéfinition des macro-instructions existantes

La redéfinition d'une macro-instruction existante devra entraîner une erreur d'exécution.

10.7.6.5 Erreurs de paramètres

Si la séquence d'instructions nécessite un nombre de paramètres différent du nombre de paramètres derrière l'étiquette de macro, l'**appareil** devra générer une erreur d'instruction.

L'**appareil** devra réaliser une vérification des paramètres normaux après la substitution des paramètres.

10.7.6.6 Récurrence

Si la séquence d'éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> contient l'étiquette d'une macro-instruction validée dont l'expansion entraîne l'exécution de la macro définie, il y aura alors récurrence. Le programme d'application ne peut arrêter cette récurrence qu'en envoyant DCL ou SDC. L'**appareil** peut soit permettre que la récurrence se produise indéfiniment, soit signaler une erreur d'exécution à un niveau de récurrence défini par l'appareil. La façon dont la récurrence est traitée doit être spécifiée dans la documentation de l'appareil. Voir 4.9.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.7.5 *Standard compliance*

Optional

10.7.6 *Error handling*

10.7.6.1 Oversized block

An Execution Error, see 6.1.7 and 11.5.1.1.5, shall be reported if the block is too long for the **device** to accept.

The maximum length requirement is device-specific and shall be specified in the device's documentation.

10.7.6.2 Macro label errors

A **device** shall report an Execution Error if the macro label is too long for the **device** to accept.

The maximum label length is device-specific and shall be specified in the device's documentation.

The **device** shall report an Execution Error if the macro label does not adhere to the <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> syntax in 7.6.

10.7.6.3 Block data errors

Any Command or Execution Errors contained in the block may be either detected immediately or deferred until the macro is executed.

10.7.6.4 Redefining existing macros

Redefining an existing macro shall cause an Execution Error.

10.7.6.5 Parameter errors

If the command sequence requires a different number of parameters than follow the macro label, the **device** shall generate a Command Error.

The **device** shall perform normal parameter checking after parameter substitution.

10.7.6.6 Recursion

If the sequence of <PROGRAM MESSAGE UNIT> elements contains the label of an enabled macro whose expansion leads to execution of the macro being defined then recursion will occur. The application program can only stop this recursion by sending DCL or SDC. The **device** may either allow recursion to take place indefinitely or report an Execution Error at a device-defined level of recursion. The manner in which recursion is handled must be stated in the device documentation. See 4.9.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.7.7 *Exemples*

1) *DMC "HOME", #18MOVE 0,0

La macro-instruction étiquette de façon descriptive une instruction qui envoie un traceur à sa position initiale.

2) *DMC "SLO", #210SPEED SLOW

La macro-instruction définit une abréviation d'une instruction plus longue, réduisant ainsi le trafic du bus.

3) *DMC "SETUP1", #222VOLT 14.5; CURLIM 2E-3

SETUP1 sera traduit par l'**appareil** par VOLT 14.5;CURLIM 2E-3

4) *DMC "TEST_A" #221BEGFREQ $1; ENDFREQ $2

Cet exemple illustre l'utilisation des paramètres. Cette macro-instruction attend deux paramètres. L'**appareil** interprète TEST_A 1000,2000 de la même façon que BEGFREQ 1000;ENDFREQ 2000.

5) *DMC "CLK_EXT", #219HORIZ: CLOCK: EXT: NEG

Instruction qui sélectionne une horloge extérieure pour le déclenchement horizontal d'un oscilloscope à sensibilité sur front descendant, donnée en notation abrégée.

6) *DMC "SWEEP_SET", #234START $1; MARK1 $1; STOP $2; MARK2 $2

Cette macro-instruction montre la réutilisation des paramètres. Le premier paramètre est utilisé pour sélectionner à la fois la fréquence de départ et la position d'un des marqueurs. Le second paramètre positionne la fréquence d'arrêt et un autre marqueur. SWEEP_SET 1E6,5E6 est équivalent à START 1E6;MARK1 1E6;STOP 5E6;MARK2 5E6.

10.8 **EMC, Validation d'une macro-instruction*

10.8.1 *Fonction et prescriptions*

Cette instruction valide ou invalide l'expansion de macro-instructions. Les définitions des macro-instructions ne sont pas affectées par cette instruction. Une utilisation de cette instruction est la mise hors fonction d'une expansion de macro-instruction afin d'exécuter une instruction spécifique à l'appareil avec le même nom qu'une macro-instruction.

10.8.2 *Syntaxe*

La syntaxe pour la validation d'une macro-instruction est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*EMC'.

10.8.3 *Sémantique*

La valeur de la <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> détermine si les macro-instructions définies sont validées ou invalidées.

Une <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> qui s'arrondit à une valeur entière de zéro invalide toute macro-instruction définie.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.7.7 *Examples*

1) *DMC "HOME", #18MOVE 0,0

The macro descriptively labels a command that sends a pen plotter to its home position.

2) *DMC "SLO", #210SPEED SLOW

The macro defines an abbreviation of a longer command, thus reducing bus traffic.

3) *DMC "SETUP1", #222VOLT 14.5; CURLIM 2E-3

SETUP1 will be expanded by the **device** into VOLT 14.5; CURLIM 2E-3

4) *DMC "TEST_A", #221BEGFREQ $1; ENDFREQ $2

This example illustrates the use of parameters. This macro expects two parameters. The **device** interprets TEST_A 1000, 2000 the same as BEGFREQ 1000; ENDFREQ 2000.

5) *DMC "CLK_EXT", #219HORIZ: CLOCK: EXT: NEG

The command which selects an external clock for the horizontal trigger of an oscilloscope with negative edge sensitivity is given a shorthand notation.

6) *DMC "SWEEP_SET", #234START $1; MARK1 $1; STOP $2; MARK2 $2

This macro demonstrates the re-use of parameters. The first parameter is used to set both the start frequency as well as the position of one of the markers. The second parameter sets the stop frequency and another marker. SWEEP_SET 1E6, 5E6 is equivalent to START 1E6; MARK1 1E6; STOP 5E6; MARK2 5E6.

10.8 **EMC, Enable Macro command*

10.8.1 *Function and requirements*

The Enable Macro command enables and disables expansion of macros. Macro definitions are not affected by this command. One use of this command is to turn off macro expansion in order to execute a device-specific command with the same name as a macro.

10.8.2 *Syntax*

The syntax for the Enable Macro command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*EMC'.

10.8.3 *Semantics*

The value of the <DECIMAL NUMERIC PROGRAM DATA> determines whether the defined macros are enabled or disabled.

A <DECIMAL NUMERIC PROGRAM DATA> that rounds to an integer value of zero disables any defined macros.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Une <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> qui s'arrondit à une valeur entière différente de zéro valide toute macro-instruction définie. La gamme autorisée doit être de –32767 à +32767.

10.8.4 *Instructions communes relatives*

L'implémentation de *PMC nécessite cette instruction facultative.

L'implémentation de *GMC? nécessite cette interrogation facultative.

L'implémentation de *DMC nécessite cette instruction facultative.

L'implémentation de *LMC nécessite cette instruction facultative.

L'implémentation de *EMC? nécessite cette interrogation facultative.

10.8.5 *Conformité à la norme*

Facultative.

10.8.6 *Traitement de l'erreur*

Si l'élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> ne s'arrondit pas à un nombre entier de la gamme –32767 à +32767 inclus, une erreur d'exécution devra être indiquée.

10.8.7 *Exemple*

Supposons qu'une alimentation électrique en courant alternatif a une instruction de "OUTPUT" ("SORTIE") qui nécessite deux paramètres: tension et fréquence. La nouvelle définition suivante redéfinit "SORTIE" pour prendre un seul paramètre, la tension. La fréquence est toujours de 1 kilohertz.

*DMC "OUTPUT", #228*EMC 0;OUTPUT $1,1000;*EMC 1

10.9 **EMC?, Validation d'une macro-interrogation*

10.9.1 *Fonction et prescriptions*

Cette interrogation permet au programmeur de demander si les macros sont validées. Une valeur renvoyée de 0 indique que les macros sont invalidées. Une valeur renvoyée de 1 indique que les macros sont validées.

10.9.2 *Structure d'interrogation*

10.9.2.1 Syntaxe d'interrogation

La syntaxe pour la validation d'une macro-interrogation est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*EMC?'.

10.9.2.2 Syntaxe de réponse

La syntaxe de réponse pour la validation d'une macro-interrogation est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE DÉCIMALE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE DÉCIMALE NR1> devra être soit un simple octet codé ISO DE 7 BITS pour "0" (30, 48 en décimal), soit "1" (31, 49 en décimal).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

A <DECIMAL NUMERIC PROGRAM DATA> that rounds to an integer value not equal to zero enables any defined macros. The allowed range shall be –32767 to +32767.

10.8.4 *Related common commands*

*PMC – Implementation requires this optional command.

*GMC? – Implementation requires this optional query.

*DMC – Implementation requires this optional command.

*LMC – Implementation requires this optional command.

*EMC? – Implementation requires this optional query.

10.8.5 *Standard compliance*

Optional

10.8.6 *Error handling*

If the <DECIMAL NUMERIC PROGRAM DATA> element does not round to an integer in the range –32767 to +32767 inclusive, an Execution Error shall be reported.

10.8.7 *Example*

Assume an AC power supply has a command of "OUTPUT" which requires two parameters: voltage and frequency. The following new definition redefines "OUTPUT" to take only one parameter, voltage. The frequency is always 1 kilohertz.

```
*DMC "OUTPUT", #228*EMC 0; OUTPUT $1,1000; *EMC 1
```

10.9 **EMC?, Enable Macro query*

10.9.1 *Function and requirements*

The Enable Macro query allows the programmer to query whether the macros are enabled. A returned value of 0 indicates that macros are disabled. A returned value of 1 indicates that macros are enabled.

10.9.2 *Query structure*

10.9.2.1 Query syntax

The syntax for the Enable Macro query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*EMC?'.

10.9.2.2 Response syntax

The response syntax for the Enable Macro query is defined as a <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> shall be either a single ISO 7-bit encoded byte for "0" (30, 48 decimal) or "1" (31, 49 decimal).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.9.3 *Sémantique de réponse*

Une valeur de 0 indique que les macros sont invalidées.

Une valeur de 1 indique que les macros sont validées.

10.9.4 *Instructions communes relatives*

L'implémentation de *PMC nécessite cette instruction facultative.

L'implémentation de *GMC? nécessite cette interrogation facultative.

L'implémentation de *DMC nécessite cette instruction facultative.

L'implémentation de *LMC nécessite cette instruction facultative.

L'implémentation de *EMC? nécessite cette instruction facultative.

10.9.5 *Conformité à la norme*

Facultative.

10.10 **ESE, Instruction de validation d'état d'événement standard*

10.10.1 *Fonction et prescriptions*

L'instruction de validation d'état d'événement standard sélectionne les bits du registre de validation d'état d'événement standard, comme défini en 11.5.1.3.

10.10.2 *Syntaxe*

La syntaxe pour l'instruction de validation d'état d'événement standard est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*ESE'.

10.10.3 *Sémantique*

La <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>, lorsqu'elle est arrondie à une valeur entière et exprimée en base 2 (binaire), représente les valeurs des bits du registre de validation d'état d'événement standard. Voir 11.4.2.3.

La valeur du nombre entier envoyé devra être dans la gamme de 0 à 255.

10.10.4 *Instructions communes relatives*

*ESE? – est l'interrogation associée.

*PSC – détermine si le registre de validation d'état d'événement standard est effacé à la mise sous tension.

10.10.5 *Conformité à la norme*

Obligatoire.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.9.3 *Response semantics*

A value of 0 indicates that macros are disabled.

A value of 1 indicates that macros are enabled.

10.9.4 *Related common commands*

*PMC – Implementation requires this optional command.

*GMC? – Implementation requires this optional query.

*DMC – Implementation requires this optional command.

*LMC – Implementation requires this optional command.

*EMC – Implementation requires this optional command.

10.9.5 *Standard compliance*

Optional

10.10 **ESE, Standard Event Status Enable command*

10.10.1 *Function and requirements*

The Standard Event Status Enable command sets the Standard Event Status Enable Register bits as defined in 11.5.1.3.

10.10.2 *Syntax*

The syntax for the Standard Event Status Enable command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*ESE'.

10.10.3 *Semantics*

The <DECIMAL NUMERIC PROGRAM DATA> when rounded to an integer value and expressed in base 2 (binary), represents the bit values of the Standard Event Status Enable Register. See 11.4.2.3.

The value of the integer shall be in the range of 0 through 255.

10.10.4 *Related common commands*

*ESE? – This is the companion query.

*PSC – Determines whether the Standard Event Status Enable Register is cleared at power-on.

10.10.5 *Standard compliance*

Mandatory

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.10.6 *Traitement de l'erreur*

L'envoi d'un nombre entier hors gamme entraînera une erreur d'exécution, voir 11.5.1.1.5.

10.11 **ESE? Interrogation de validation d'état d'événement standard*

10.11.1 *Fonction et prescriptions*

L'interrogation de validation d'état d'événement standard permet au programmeur de déterminer le contenu courant du registre de validation d'état d'événement standard. Voir 11.5.1.3.

10.11.2 *Structure d'interrogation*

10.11.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de validation d'état d'événement standard est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*ESE?'.

10.11.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de validation d'état d'événement standard est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> devra être dans la gamme de 0 à 255.

10.11.3 *Sémantique de réponse*

La valeur entière de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> exprimée en base 2 (binaire) représente les valeurs des bits du registre de validation d'état d'événement standard. Voir 11.4.2.3.2.

10.11.4 *Instructions communes relatives*

*ESE – est l'instruction associée.

10.11.5 *Conformité à la norme*

Obligatoire.

10.12 **ESR?, Interrogation de registre d'état d'événement standard*

10.12.1 *Fonction et prescriptions*

Interrogation de registre d'état d'événement standard permet au programmeur de déterminer le contenu courant du registre d'état d'événement standard. Le fait de lire le registre d'état d'événement standard le met à zéro. Voir 11.5.1.2.

10.12.2 *Structure d'interrogation*

10.12.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de registre d'état d'événement standard est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*ESR?'.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.10.6 *Error handling*

An out-of-range integer shall cause an Execution Error, see 11.5.1.1.5.

10.11 **ESE?, Standard Event Status Enable query*

10.11.1 *Function and requirements*

The Standard Event Status Enable query allows the programmer to determine the current contents of the Standard Event Status Enable Register. See 11.5.1.3.

10.11.2 *Query structure*

10.11.2.1 Query syntax

The syntax for the Standard Event Status Enable query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*ESE?'.

10.11.2.2 Response syntax

The response syntax for the Standard Event Status Enable query is defined as a <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> shall be in the range 0 through 255.

10.11.3 *Response semantics*

The <NR1 NUMERIC RESPONSE DATA> integer value expressed in base 2 (binary) represents the bit values of the Standard Event Status Enable register. See 11.4.2.3.2.

10.11.4 *Related common commands*

*ESE – This is the companion command.

10.11.5 *Standard compliance*

Mandatory.

10.12 **ESR?, Standard Event Status Register query*

10.12.1 *Function and requirements*

The Standard Event Status Register query allows the programmer to determine the current contents of the Standard Event Status Register. Reading the Standard Event Status Register clears it. See 11.5.1.2.

10.12.2 *Query structure*

10.12.2.1 Query syntax

The syntax for the Standard Event Status Register query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*ESR?'.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.12.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de registre d'état d'événement standard est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> devra être dans la gamme de 0 à 255.

10.12.3 *Sémantique de réponse*

La valeur entière de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> exprimée en base 2 (binaire) représente les valeurs des bits du registre d'état d'événement standard. Voir 11.5.1.

10.12.4 *Instructions communes relatives*

Aucune.

10.12.5 *Conformité à la norme*

Obligatoire.

10.13 **GMC?, Lecture du contenu des macro-instructions*

10.13.1 *Fonction et prescriptions*

Cette interrogation permet de retrouver une définition courante d'une macro-instruction dans un **appareil**.

10.13.2 *Structure d'interrogation*

10.13.2.1 Syntaxe d'interrogation

La syntaxe pour la lecture du contenu des macro-instructions est définie comme un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION DE CHAÎNE>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*GMC?'.

10.13.2.2 Syntaxe de réponse

La syntaxe de réponse pour la lecture du contenu des macro-instructions est définie comme un élément de <DONNÉE DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE>.

10.13.3 *Sémantique*

10.13.3.1 Sémantique des paramètres

La donnée dans l'élément de <DONNÉE DE PROGRAMMATION DE CHAÎNE> doit être une étiquette de macro couramment définie.

10.13.3.2 Sémantique de réponse

L'élément de <DONNÉE DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE> contient la séquence d'<UNITÉ DE MESSAGE DE PROGRAMMATION> qui est exécutée lorsque l'étiquette de macro est reçue.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.12.2.2 Response syntax

The response syntax for the Standard Event Status Register query is defined as an <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> shall be in the range 0 through 255.

10.12.3 *Response semantics*

The <NR1 NUMERIC RESPONSE DATA> integer value expressed in base 2 (binary) represents the bit values of the Standard Event Status Register. See 11.5.1.

10.12.4 *Related common commands*

None

10.12.5 *Standard compliance*

Mandatory

10.13 **GMC?, Get Macro Contents query*

10.13.1 *Function and requirements*

The Get Macro Contents query allows the current definition of a macro to be retrieved from a **device**.

10.13.2 *Query structure*

10.13.2.1 Query syntax

The syntax for the Get Macro Contents query is defined as a <QUERY PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <STRING PROGRAM DATA> element. The <QUERY PROGRAM HEADER> is defined as '*GMC?'.

10.13.2.2 Response syntax

The response syntax for the Get Macro Contents query is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

10.13.3 *Semantics*

10.13.3.1 Parameter semantics

The data in the <STRING PROGRAM DATA> element must be a currently defined macro label.

10.13.3.2 Response semantics

The <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element contains the <PROGRAM MESSAGE UNIT> sequence which is executed when the macro label is received.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

La condition de "pas de séquence d'instruction enregistrée" engendre une réponse de bloc de longueur zéro.

10.13.4 *Instructions communes relatives*

L'implémentation de *PMC nécessite cette instruction facultative.

L'implémentation de *EMC nécessite cette instruction facultative.

L'implémentation de *EMC? nécessite cette interrogation facultative.

L'implémentation de *DMC nécessite cette instruction facultative.

L'implémentation de *LMC nécessite cette instruction facultative.

10.13.5 *Conformité à la norme*

Facultative.

10.13.6 *Traitement de l'erreur*

Si le programmeur tente de rappeler le contenu d'une macro-instruction non définie, l'**appareil** devra retourner un bloc de longueur zéro et signaler une erreur d'exécution.

10.14 **IDN?, Interrogation d'identification*

10.14.1 *Fonction et prescriptions*

Le but de l'interrogation d'identification est uniquement l'identification d'**appareils** par le **système d'interface**.

10.14.2 *Structure d'interrogation*

10.14.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation d'identification est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*IDN?'.

10.14.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation d'identification est définie comme un élément de <DONNÉE DE RÉPONSE ISO ARBITRAIRE>. Ceci implique que l'interrogation *IDN? devrait être la dernière <UNITÉ DE MESSAGE D'INTERROGATION> dans un <MESSAGE DE PROGRAMMATION TERMINÉ> (voir 8.7.11 et 6.4.3.).

10.14.3 *Sémantique de réponse*

La réponse est organisée en quatre zones séparées par des virgules. Les définitions des zones se font de la façon suivante:

Zone 1	Constructeur	exigé
Zone 2	Modèle	exigé
Zone 3	Numéro de série	caractère 0 ISO si non disponible
Zone 4	Niveau de microprogrammation ou équivalent	caractère 0 ISO si non disponible

NOTE – Le caractère 0 ISO représente un octet codé ISO simple avec la valeur 30 (48 en décimal).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The condition of "no command sequence stored" results in a response of a zero-length block.

10.13.4 *Related common commands*

*PMC – Implementation requires this optional command.

*EMC – Implementation requires this optional command.

*EMC? – Implementation requires this optional query.

*DMC – Implementation requires this optional command.

*LMC – Implementation requires this optional command.

10.13.5 *Standard compliance*

Optional

10.13.6 *Error handling*

If the programmer attempts to retrieve the contents of an undefined macro, the **device** shall return a zero-length block and report an Execution Error.

10.14 **IDN?, Identification query*

10.14.1 *Function and requirements*

The intent of the Identification query is for the unique identification of **devices** over the **system interface**.

10.14.2 *Query structure*

10.14.2.1 Query syntax

The syntax for the Identification query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*IDN?'.

10.14.2.2 Response syntax

The response syntax for the Identification query is defined as an <ARBITRARY ISO 7-bit RESPONSE DATA> element. This implies that the *IDN? query should be the last <QUERY MESSAGE UNIT> in a <TERMINATED PROGRAM MESSAGE> (see 8.7.11 and 6.4.3).

10.14.3 *Response semantics*

The response is organized into four fields separated by commas. The field definitions are as follows:

Field 1	Manufacturer	required
Field 2	Model	required
Field 3	Serial Number	ISO 7-bit character 0 if not available
Field 4	Firmware Level or equivalent	ISO 7-bit character 0 if not available

NOTE - ISO 7-bit character 0 represents a single ISO 7-bit encoded byte with value 30 (48 decimal).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

La présence de données dans toutes les zones est obligatoire. Si l'une des zones 3 ou 4 n'est pas disponible, le caractère 0 devra être retourné à cette zone.

Une zone peut contenir n'importe quel octet codé ISO de 7 bits dans la gamme 20 à 7E (32 à 126 en décimal) sauf les virgules (2C, 44 en décimal) et les points-virgules (3B, 59 en décimal).

La longueur totale de la réponse *IDN? devra être inférieure ou égale à 72 caractères.

Le format précis des zones (sauf caractère 0 ISO signifiant zone nulle) est au choix du concepteur de l'appareil.

Si des numéros de série uniques sont utilisés, le caractère 0 ISO ne devra alors pas être utilisé comme un numéro de série.

10.14.4 *Instructions communes relatives*

Aucune.

10.14.5 *Conformité à la norme*

Obligatoire.

10.14.6 *Exemple*

Une réponse hypothétique *IDN? apparaîtrait comme suit pour un produit modèle 246B d'une société XYZ Co qui a choisi le mnémonique "XYZCO" pour se représenter avec le numéro de série S000-123-02, et aucune révision de micrologiciel n'étant spécifiée.

XYZCO, 246B, S000-0123-02,0

10.15 **IST?, Interrogation de lecture d'état individuel*

10.15.1 *Fonction et prescriptions*

Cette interrogation permet au programmeur de lire l'état du message local 'ist' défini par CEI 625-1 dans l'**appareil**, voir 11.6.2.

10.15.2 *Structure d'interrogation*

10.15.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de lecture d'état individuel est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*IST?'.

10.15.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de lecture d'état individuel est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> devra être un octet simple codé ISO codé comme un '0' ou un '1' (30 ou 31, 48 ou 49 en décimal).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

The presence of data in all the fields is mandatory. If either field 3 or 4 is not available, the ISO 7-bit character 0 shall be returned for that field.

A field may contain any ISO 7-bit encoded bytes in the range 20 through 7E (32 through 126 decimal) except commas (2C, 44 decimal) and semicolons (3B, 59 decimal).

The overall length of the *IDN? response shall be less than or equal to 72 characters.

The precise format of fields (except ISO 7-bit character 0 as meaning empty field) is left to the device designer.

If unique serial numbers are used, then ISO 7-bit character 0 shall not be used as a serial number.

10.14.4 *Related common commands*

None

10.14.5 *Standard compliance*

Mandatory

10.14.6 *Example*

A hypothetical *IDN? response might appear as follows for a model 246B product from a company XYZ Co that has chosen the mnemonic "XYZCO" to represent itself with serial number S000-123-02, and no firmware revision specified.

XYZCO, 246B, S000-0123-02,0

10.15 **IST?, Individual Status query*

10.15.1 *Function and requirements*

The Individual Status query allows the programmer to read the current state of the IEC 625-1 defined 'ist' local message in the **device**, see 11.6.2.

10.15.2 *Query structure*

10.15.2.1 Query syntax

The syntax for the Individual Status query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*IST?'.

10.15.2.2 Response syntax

The response syntax for the Individual Status query is defined as an <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> shall be a single ISO 7-bit encoded byte encoded as a '0' or '1' (30 or 31, 48 or 49 decimal).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.15.3 *Sémantique de réponse*

Un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> avec la valeur 0 (zéro) indique que le message local ist est FAUX.

Un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> avec la valeur 1 (un) indique que le message local ist est VRAI.

La réponse à cette interrogation dépend de l'état courant de l'instrument.

10.15.4 *Instructions communes relatives*

Aucune

10.15.5 *Conformité à la norme*

Obligatoire pour les appareils implémentant le sous-ensemble PP1.

10.16 **LMC?, Interrogation des macros définies*

10.16.1 *Fonction et prescriptions*

Cette interrogation renvoie les étiquettes des macros définies.

10.16.2 *Structure d'interrogation*

10.16.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation des macros définies est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*LMC?'.

10.16.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation des macros définies est définie comme une séquence d'un ou plusieurs éléments de <DONNÉES DE RÉPONSE DE CHAÎNES> séparés par des <SÉPARATEURS DE DONNÉES DE RÉPONSE>.

10.16.3 *Sémantique de réponse*

Les éléments de <DONNÉES DE RÉPONSE DE CHAÎNES> contiennent des étiquettes de macros définies dans l'**appareil**, en utilisant l'instruction *DMC, voir 10.7.

Si aucune étiquette de macro n'est définie, la réponse devra alors être une <DONNÉE DE RÉPONSE DE CHAÎNE> simple avec une donnée de chaîne nulle, c'est-à-dire deux guillemets(") consécutifs.

La réponse est indépendante de l'état macro-instruction validée ou macro-instruction invalidée de l'**appareil**.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.15.3 *Response semantics*

An <NR1 NUMERIC RESPONSE DATA> element with the value 0 (zero) indicates the ist local message is FALSE.

An <NR1 NUMERIC RESPONSE DATA> element with the value 1 (one) indicates the ist local message is TRUE.

The response of this query is dependent upon the current status of the instrument.

10.15.4 *Related common commands*

None

10.15.5 *Standard compliance*

Mandatory for devices implementing the PP1 subset.

10.16 **LMC?, Learn Macro query*

10.16.1 *Function and requirements*

This query returns the currently defined macro labels.

10.16.2 *Query structure*

10.16.2.1 Query syntax

The syntax for the Learn Macro query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*LMC?'.

10.16.2.2 Response syntax

The response syntax for the Learn Macro query is defined as a sequence of one or more <STRING RESPONSE DATA> elements separated by <RESPONSE DATA SEPARATORS>.

10.16.3 *Response semantics*

The <STRING RESPONSE DATA> elements contain macro labels currently defined in the **device**, using the *DMC command, see 10.7.

If no macro labels are defined, then the response shall be a single <STRING RESPONSE DATA> with null string data, that is, two consecutive double quote (") marks.

The response is independent of the macro-enabled or macro-disabled state of the **device.**

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.16.4 *Instructions communes relatives*

Voir instruction *PMC, 10.22.

L'implémentation de *PMC nécessite cette instruction facultative.

L'implémentation de *DMC nécessite cette instruction facultative.

L'implémentation de *EMC nécessite cette instruction facultative.

L'implémentation de *EMC? nécessite cette interrogation facultative.

L'implémentation de *GMC? nécessite cette interrogation facultative.

10.16.5 *Conformité à la norme*

Facultative

10.17 **LRN?, Interrogation d'apprentissage de la configuration de l'appareil*

10.17.1 *Fonction et prescriptions*

L'interrogation d'apprentissage de la configuration de l'appareil permet au programmeur d'obtenir une séquence d'éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> qui peut être utilisée plus tard pour placer l'**appareil** dans l'état où il était lorsque l'interrogation commune *LRN? a été faite.

10.17.2 *Structure d'interrogation*

10.17.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation d'apprentissage de la configuration de l'appareil est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*LRN?'.

10.17.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation d'apprentissage de la configuration de l'appareil est définie comme une séquence d'un ou plusieurs éléments d'<UNITÉS DE MESSAGES DE RÉPONSE> séparés par des éléments <SÉPARATEURS D'UNITÉS DE MESSAGES DE RÉPONSE>. L'<UNITÉ DE MESSAGE DE RÉPONSE> suivra le parcours représenté en 8.3.2 qui contient l'<EN-TÊTE DE RÉPONSE>.

10.17.3 *Sémantique de réponse*

Le <MESSAGE DE RÉPONSE> est une séquence d'un ou plusieurs éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION> directement renvoyables comme des éléments d'<UNITÉS DE MESSAGES DE PROGRAMMATION>.

L'**appareil** devra être placé dans son état "appris" à partir de n'importe quel autre état courant à réception de ce <MESSAGE DE RÉPONSE> comme un <MESSAGE DE PROGRAMMATION>.

L'étendue des placements en position est la même que pour l'instruction *RST, voir 10.32.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.16.4 *Related common commands*

See *PMC command, 10.22.

*PMC – Implementation requires this optional command.

*DMC – Implementation requires this optional command.

*EMC – Implementation requires this optional command.

*EMC? – Implementation requires this optional query.

*GMC? – Implementation requires this optional query.

10.16.5 *Standard compliance*

Optional

10.17 **LRN?, Learn Device Setup query*

10.17.1 *Function and requirements*

The Learn Device Setup query allows the programmer to obtain a sequence of <PROGRAM MESSAGE UNIT> elements that may later be used to place the **device** in the state it was in when the *LRN? common query was made.

10.17.2 *Query structure*

10.17.2.1 Query syntax

The syntax for the Learn Device Setup query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*LRN?'.

10.17.2.2 Response syntax

The response syntax for the Learn Device Setup query is defined as a sequence of one or more <RESPONSE MESSAGE UNIT> elements separated by <RESPONSE MESSAGE UNIT SEPARATOR> elements. The <RESPONSE MESSAGE UNIT> shall use the path shown in 8.3.2 which contains the <RESPONSE HEADER>.

10.17.3 *Response semantics*

The <RESPONSE MESSAGE> is a sequence of one or more <RESPONSE MESSAGE UNIT> elements that are directly resendable as <PROGRAM MESSAGE UNIT> elements.

The **device** shall be restored to its "learned" state from any other current state upon receipt of this <RESPONSE MESSAGE> as a <PROGRAM MESSAGE>.

The scope of the restored settings is the same as the *RST command, see 10.32.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.17.4 *Instructions communes relatives*

Aucune

10.17.5 *Conformité à la norme*

Facultative

10.18 **OPC, Instruction d'opération achevée*

10.18.1 *Fonction et prescriptions*

L'instruction d'opération achevée fait générer à l'**appareil** le message d'opération achevée dans le registre d'état d'événement standard lorsque toutes les opérations en cours ont été terminées.

Voir 12.5.2 pour les détails de l'opération.

10.18.2 *Syntaxe*

La syntaxe pour l'instruction d'opération achevée est définie comme uniquement un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*OPC'.

10.18.3 *Sémantique*

Non applicable

10.18.4 *Instructions communes relatives*

*OPC?, *WAI

10.18.5 *Conformité à la norme*

Obligatoire

10.19 **OPC?, Interrogation d'opération achevée*

10.19.1 *Fonction et prescriptions*

L'interrogation d'opération achevée place un caractère 1 ISO dans la file d'attente de sortie de l'appareil lorsque toutes les opérations de l'**appareil** sélectionnées en cours ont été terminées.

Voir 12.5.3 pour les détails de l'opération.

10.19.2 *Structure d'interrogation*

10.19.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation d'opération achevée est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*OPC?'.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.17.4 *Related common commands*

None

10.17.5 *Standard compliance*

Optional

10.18 **OPC, Operation Complete command*

10.18.1 *Function and requirements*

The Operation Complete command causes the **device** to generate the operation complete message in the Standard Event Status Register when all pending selected device operations have been finished.

See 12.5.2 for details of operation.

10.18.2 *Syntax*

The syntax for the Operation Complete command is defined as only a <COMMAND PROGRAM HEADER>. The <COMMAND PROGRAM HEADER> is defined as '*OPC'.

10.18.3 *Semantics*

Not applicable

10.18.4 *Related common commands*

*OPC?, *WAI

10.18.5 *Standard compliance*

Mandatory

10.19 **OPC?, Operation Complete query*

10.19.1 *Function and requirements*

The Operation Complete query places an ISO 7-bit character 1 into the **device's** Output Queue when all pending selected device operations have been finished.

See 12.5.3 for details of operation.

10.19.2 *Query structure*

10.19.2.1 Query syntax

The syntax for the Operation Complete query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*OPC?'.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.19.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation d'opération achevée est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> devra être un simple octet codé ISO pour '1' (31, 49 en décimal).

10.19.3 *Sémantique de réponse*

Aucune

10.19.4 *Instructions communes relatives*

*OPC, *WAI

10.19.5 *Conformité à la norme*

Obligatoire

10.20 **OPT?, Interrogation d'identification d'option*

10.20.1 *Fonction et prescriptions*

L'interrogation d'identification d'option sert à l'identification des options de l'appareil qu'il est possible d'indiquer par le **système d'interface**.

10.20.2 *Structure d'interrogation*

10.20.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation d'identification d'option est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*OPT?'.

10.20.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'instruction d'opération achevée est définie comme un élément de <DONNÉE DE RÉPONSE ISO ARBITRAIRE>. Ceci implique que l'interrogation *OPT? devrait être la dernière <UNITÉ DE MESSAGE D'INTERROGATION> dans un <MESSAGE DE PROGRAMMATION TERMINÉ> (voir 8.7.11 et 6.4.3).

10.20.3 *Sémantique de réponse*

La réponse consiste en un nombre indéterminé de zones séparées par des virgules. Les zones doivent suivre ces règles:

1) Chaque zone doit correspondre à une option qu'on peut indiquer, présente dans l'appareil. Le format précis des zones est au choix du concepteur de l'appareil.

2) Les zones vides sont interdites. Les options qu'on peut indiquer manquantes répondent avec un caractère 0 ISO (30, 48 en décimal).

3) La donnée d'option peut être indiquée d'une manière dépendante de la position.

4) Un caractère 0 ISO simple doit être renvoyé comme réponse si l'appareil ne contient pas d'options qu'on peut indiquer. Le caractère 0 ISO ne doit pas être utilisé comme une identification d'option.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.19.2.2 Response syntax

The response syntax for the Operation Complete query is defined as an <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> shall be a single ISO 7-bit encoded byte for '1' (31, 49 decimal).

10.19.3 *Response semantics*

None

10.19.4 *Related common commands*

*OPC, *WAI

10.19.5 *Standard compliance*

Mandatory

10.20 **OPT?, Option Identification query*

10.20.1 *Function and requirements*

The Option Identification query is for identifying reportable device options over the **system interface**.

10.20.2 *Query structure*

10.20.2.1 Query syntax

The syntax for the Option Identification query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*OPT?'.

10.20.2.2 Response syntax

The response syntax for the Option Identification query is defined as an <ARBITRARY ISO 7-bit RESPONSE DATA> element. This implies that the *OPT? query should be the last <QUERY MESSAGE UNIT> in a <TERMINATED PROGRAM MESSAGE> (see 8.7.11 and 6.4.3).

10.20.3 *Response semantics*

The response consists of any number of fields separated by commas. The fields shall follow these rules:

1) Each field must correspond to a reportable option present in the device. The precise format of the fields is left to the device designer.

2) Empty fields are prohibited. Missing reportable options respond with an ISO 7-bit character 0 (30, 48 decimal).

3) Option data may be reported in a position-dependent manner.

4) A single ISO 7-bit character 0 shall be returned as a response if the **device** contains no reportable options. ISO 7-bit character 0 may not be used as an option identification.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

5) Une zone peut contenir n'importe quel octet codé ISO de 7 bits dans la gamme 20 à 7E (32 à 126 en décimal), sauf les virgules (2C, 44 en décimal) et les points-virgules (3B, 59 en décimal).

La longueur totale de la réponse *OPT? devra être inférieure ou égale à 255 caractères.

10.20.4 *Instructions communes relatives*

Aucune

10.20.5 *Conformité à la norme*

Facultative

10.21 **PCB, Retour de passage de contrôle*

10.21.1 *Fonction et prescriptions*

L'instruction de retour de passage de contrôle est utilisée pour dire à un contrôleur potentiel quelle adresse doit être utilisée lors d'un retour de passage de contrôle.

Chaque **appareil** qui contient cette instruction devra posséder une capacité de contrôleur. Chaque **appareil** qui possède une capacité de contrôleur devra implémenter cette instruction.

10.21.2 *Syntaxe*

La syntaxe pour l'instruction de retour de passage de contrôle est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> suivi facultativement par un <SÉPARATEUR DE DONNÉES DE PROGRAMMATION> et un autre élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*PCB'. La première <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> devra s'arrondir à une valeur entière dans la gamme de 0 à 30. La deuxième <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> si elle est présente, devra s'arrondir à une valeur entière dans la gamme de 0 à 30.

10.21.3 *Sémantique*

Le premier élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> est arrondi à un entier dont la valeur est interprétée comme l'adresse primaire du contrôleur envoyant l'instruction.

Le deuxième élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> est arrondi à un entier dont la valeur est interprétée comme l'adresse secondaire du contrôleur envoyant l'instruction.

Une deuxième adresse manquante indiquera que le contrôleur envoyant cette instruction ne possède pas d'adressage étendu.

10.21.4 *Instructions communes relatives*

Aucune

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

5) A field may contain any ISO 7-bit-encoded bytes in the range 20 through 7E (32 through 126 decimal) except commas (2C, 44 decimal) and semicolons (3B, 59 decimal).

The overall length of the *OPT? response shall be less than or equal to 255 characters.

10.20.4 *Related common commands*

None

10.20.5 *Standard compliance*

Optional

10.21 **PCB, Pass Control Back*

10.21.1 *Function and requirements*

The Pass Control Back command is used to tell a potential controller which address should be used when passing control back.

Any **device** which implements this command shall have controller capability. Any **device** which has controller capability shall implement this command.

10.21.2 *Syntax*

The syntax for the Pass Control Back command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <DECIMAL NUMERIC PROGRAM DATA> element optionally followed by a <PROGRAM DATA SEPARATOR> and another <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*PCB'. The first <DECIMAL NUMERIC PROGRAM DATA> shall round to an integer value in the range of 0 to 30. The second <DECIMAL NUMERIC PROGRAM DATA>, if present, shall round to an integer value in the range of 0 to 30.

10.21.3 *Semantics*

The first <DECIMAL NUMERIC PROGRAM DATA> element is rounded to an integer whose value is interpreted as the primary address of the controller sending the command.

The second <DECIMAL NUMERIC PROGRAM DATA> element is rounded to an integer whose value is interpreted as the secondary address of the controller sending the command.

A missing second address shall indicate the controller sending this command does not have extended addressing.

10.21.4 *Related common commands*

None

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.21.5 *Conformité à la norme*

Obligatoire pour les **appareils** qui ont la capacité contrôleur (appareils non CO). Voir 15.1.4.

10.21.6 *Traitement de l'erreur*

10.21.6.1 Impossibilité de passage de contrôle

Un **appareil** ne possédant pas la capacité contrôleur (CO) devra générer une erreur d'instruction s'il reçoit l'instruction commune *PCB.

10.21.6.2 Paramètres hors gamme

L'**appareil** recevant une instruction *PCB devra générer une erreur d'exécution si la valeur du premier ou du second paramètre de données est en dehors de sa gamme spécifiée.

10.22 **PMC, Elimination des macro-instructions*

10.22.1 *Fonction et prescriptions*

L'instruction d'élimination des macro-instructions fait effacer à l'**appareil** toutes les macro-instructions qui peuvent avoir été préalablement définies en utilisant l'instruction *DMC. Toutes les séquences et étiquettes de macro-instructions enregistrées devront être retirées de la mémoire de l'**appareil** par cette instruction.

10.22.2 *Syntaxe*

La syntaxe pour l'instruction d'élimination des macro-instructions est définie comme uniquement un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*PMC'.

10.22.3 *Sémantique*

Non applicable

10.22.4 *Instructions communes relatives*

L'implémentation de *EMC nécessite cette instruction facultative.

L'implémentation de *EMC? nécessite cette interrogation facultative.

L'implémentation de *GMC? nécessite cette interrogation facultative.

L'implémentation de *DMC nécessite cette instruction facultative.

L'implémentation de *LMC nécessite cette instruction facultative.

10.22.5 *Conformité à la norme*

Facultative

10.23 **PRE, Instruction de validation de registre de validation d'interrogation parallèle*

10.23.1 *Fonction et prescriptions*

L'instruction de validation de registre de validation d'interrogation parallèle positionne les bits de validation du registre de validation d'interrogation parallèle comme défini en 11.6.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.21.5 *Standard compliance*

Mandatory for **devices** that have any controller capability (non-CO devices). See 15.1.4.

10.21.6 *Error handling*

10.21.6.1 Unable to Pass Control

A **device** with no controller capability (CO) shall issue a Command Error if it receives the *PCB common command.

10.21.6.2 Parameters out of range

The **device** receiving the *PCB command shall issue an Execution Error if the value of either the first or second parameter is out of its specified range.

10.22 **PMC, Purge Macros command*

10.22.1 *Function and requirements*

The Purge Macros command causes the **device** to delete all macros that may have been previously defined using the *DMC command. All stored macro command sequences and labels shall be removed from the **device's** memory by this command.

10.22.2 *Syntax*

The syntax for the Purge Macros command is defined as only a <COMMAND PROGRAM HEADER>. The <COMMAND PROGRAM HEADER> is defined as '*PMC'.

10.22.3 *Semantics*

Not applicable

10.22.4 *Related common commands*

*EMC – Implementation requires this optional command.

*EMC? – Implementation requires this optional query.

*GMC? – Implementation requires this optional query.

*DMC – Implementation requires this optional command.

*LMC – Implementation requires this optional command.

10.22.5 *Standard compliance*

Optional

10.23 **PRE, Parallel Poll Enable Register Enable command*

10.23.1 *Function and requirements*

The Parallel Poll Enable Register Enable command sets the Parallel Poll Register Enable bits as defined in 11.6.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.23.2 *Syntaxe*

La syntaxe pour l'instruction de validation de registre de validation d'interrogation parallèle est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*PRE'. La valeur de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> sera arrondie à un entier dans la gamme de 0 à 65535.

10.23.3 *Sémantique*

La <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>, lorsqu'elle est arrondie à une valeur entière et exprimée en base 2 (binaire), représente les valeurs des bits du registre de validation d'interrogation parallèle. Voir 11.6.1.3.

10.23.4 *Instructions communes relatives*

*PRE? – est l'interrogation associée.

10.23.5 *Conformité à la norme*

Obligatoire pour les appareils implémentant le sous-ensemble PP1.

10.23.6 *Traitement de l'erreur*

Si le nombre entier est en dehors de la gamme de 0 à 65535 inclus, une erreur d'exécution devra être indiquée, voir 11.5.1.1.5.

10.24 **PRE?, Interrogation de validation de registre de validation d'interrogation parallèle*

10.24.1 *Fonction et prescriptions*

L'interrogation de validation de registre de validation d'interrogation parallèle permet au programmeur de déterminer le contenu courant de validation de registre de validation d'interrogation parallèle.

10.24.2 *Structure d'interrogation*

10.24.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de validation de registre de validation d'interrogation parallèle est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*PRE?'.

10.24.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de validation de registre de validation d'interrogation parallèle est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La valeur de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> est un nombre entier dans la gamme de 0 à 65535.

10.24.3 *Sémantique de réponse*

La valeur entière de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> exprimée en base 2 (binaire) représente les valeurs des bits du registre de validation d'interrogation parallèle. Voir 11.6.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.23.2 *Syntax*

The syntax for the Parallel Poll Enable Register command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*PRE'. The <DECIMAL NUMERIC PROGRAM DATA> value shall round to an integer in the range 0 to 65535.

10.23.3 *Semantics*

The <DECIMAL NUMERIC PROGRAM DATA> when rounded to an integer value and expressed in base 2 (binary) represents the bit values of the Parallel Poll Enable Register. See 11.6.1.3.

10.23.4 *Related common commands*

*PRE? – This is the companion query.

10.23.5 *Standard compliance*

Mandatory for devices implementing the PP1 subset.

10.23.6 *Error handling*

If the integer is outside the range 0 to 65535 inclusive, an Execution Error shall be reported, see 11.5.1.1.5.

10.24 **PRE?, Parallel Poll Enable Register Enable query*

10.24.1 *Function and requirements*

The Parallel Poll Enable Register Enable query allows the programmer to determine the current contents of the Parallel Poll Enable Register.

10.24.2 *Query structure*

10.24.2.1 Query syntax

The syntax for the Parallel Poll Enable Register query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*PRE?'.

10.24.2.2 Response syntax

The response syntax for the Parallel Poll Enable Register query is defined as an <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> value is an integer in the range 0 to 65535.

10.24.3 *Response semantics*

The <NR1 NUMERIC RESPONSE DATA> integer value expressed in base 2 (binary) represents the bit values of the Parallel Poll Enable Register. See 11.6.1.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.24.4 *Instructions communes relatives*

*PRE – est l'instruction associée.

10.24.5 *Conformité à la norme*

Obligatoire pour les appareils implémentant les sous-ensembles PP1.

10.25 **PSC, Instruction d'effacement d'état à la mise sous tension*

10.25.1 *Fonction et prescriptions*

L'instruction d'effacement d'état à la mise sous tension commande l'effacement automatique à la mise sous tension du registre de validation de demande de service, du registre de validation d'état d'événement standard et du registre de validation d'interrogation parallèle, voir 11.1.3. Aucune autre fonction de l'appareil ne devra être rattachée à cette instruction. Pour les exigences supplémentaires de mise sous tension CEI 625-2, voir 5.12.

10.25.2 *Syntaxe*

La syntaxe pour l'instruction d'effacement d'état à la mise sous tension est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*PSC'. La valeur de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> devra être dans la gamme de –32767 à 32767.

10.25.3 *Sémantique*

La valeur de la <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> détermine si l'indicateur d'effacement à la mise sous tension de l'**appareil** est VRAI ou FAUX.

Une <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> qui s'arrondit à une valeur entière de zéro positionne l'indicateur d'effacement à la mise sous tension FAUX. Par conséquent, envoyer *PSC 0 permet aux instruments d'activer SRQ après mise sous tension.

Une <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> qui s'arrondit à une valeur entière différente de zéro positionne l'indicateur d'effacement à la mise sous tension VRAI. Par conséquent, envoyer *PSC 1 valide l'effacement à la mise sous tension et ne permet plus aucune activation de SRQ après mise sous tension.

10.25.4 *Instructions communes relatives*

L'implémentation de *PSC? nécessite cette instruction facultative. Ceci est l'interrogation associée.

10.25.5 *Conformité à la norme*

Facultative

10.25.6 *Traitement de l'erreur*

Si le nombre entier est en dehors de la gamme de –32767 à 32767 inclus, une erreur d'exécution sera signalée, voir 11.5.1.1.5.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.24.4 *Related common commands*

*PRE – This is the companion command.

10.24.5 *Standard compliance*

Mandatory for devices implementing the PP1 subset.

10.25 **PSC, Power-on Status Clear command*

10.25.1 *Function and requirements*

The Power-on Status Clear command controls the automatic power-on clearing of the Service Request Enable Register, the Standard Event Status Enable Register, and the Parallel Poll Enable Register, see 11.1.3. No other device functions shall be tied to this command. For additional IEC 625-2 power-on requirements, see 5.12.

10.25.2 *Syntax*

The syntax for the Power-on Status Clear command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*PSC'. The <DECIMAL NUMERIC PROGRAM DATA> value shall be in the range –32767 to 32767.

10.25.3 *Semantics*

The value of the <DECIMAL NUMERIC PROGRAM DATA> determines whether the power-on-clear flag of the **device** is TRUE or FALSE.

A <DECIMAL NUMERIC PROGRAM DATA> that rounds to an integer value of zero sets the power-on-clear flag FALSE. Sending *PSC 0, therefore, allows instruments to assert SRQ after power-on.

A <DECIMAL NUMERIC PROGRAM DATA> that rounds to an integer value not equal to zero sets the power-on-clear flag TRUE. Sending *PSC 1, therefore enables the power-on clear and disallows any SRQ assertion after power-on.

10.25.4 *Related common commands*

*PSC? – Implementation requires this optional command. This is the companion query.

10.25.5 *Standard compliance*

Optional

10.25.6 *Error handling*

If the integer is outside the range –32767 to 32767 inclusive, an Execution Error shall be reported, see 11.5.1.1.5.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993 Withdrawn

10.25.7 *Exemple*

La séquence d'instruction

*PSC 0; *SRE 32; *ESE 128;

permettra à un **appareil** d'activer SRQ à l'achèvement de la mise sous tension.

10.26 **PSC?, Interrogation d'effacement d'état à la mise sous tension*

10.26.1 *Fonction et prescriptions*

L'interrogation d'effacement d'état à la mise sous tension permet au programmeur d'interroger l'indicateur d'effacement à la mise sous tension de l'**appareil**. Le renvoi d'une valeur 0 indique que le registre de validation d'état d'événement standard, le registre de validation de demande de service, et le registre de validation parallèle, conserveront leur état lorsque l'alimentation sera rétablie dans l'**appareil**. Le renvoi d'une valeur 1 indique que les registres énumérés ci-dessus seront remis à zéro lorsque l'alimentation sera rétablie dans l'**appareil**, voir 11.1.3.

10.26.2 *Structure d'interrogation*

10.26.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation d'effacement d'état à la mise sous tension est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*PSC?'.

10.26.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation d'effacement d'état à la mise sous tension est définie comme uniquement un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE DÉCIMALE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE DÉCIMALE NR1> devra être un octet simple codé ISO pour "0" (30, 48 en décimal), ou "1" (31, 49 en décimal).

10.26.3 *Sémantique de réponse*

Une valeur 0 indique que l'indicateur d'effacement à la mise sous tension de l'appareil est FAUX.

Une valeur 1 indique que l'indicateur d'effacement à la mise sous tension de l'appareil est VRAI.

10.26.4 *Instructions communes relatives*

L'implémentation de *PSC nécessite cette instruction facultative. Ceci est l'instruction associée.

10.26.5 *Conformité à la norme*

Facultative

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.25.7 *Example*

The command sequence

*PSC 0; *SRE 32; *ESE 128;

allows a **device** to assert SRQ upon completion of power-on.

10.26 **PSC?, Power-on Status Clear query*

10.26.1 *Function and requirements*

The Power-on Status Clear query allows the programmer to query the **device's** power-on-clear flag. A returned value of 0 indicates that the Standard Event Status Enable Register, Service Request Enable Register, and the Parallel Poll Enable Register, will retain their status when power is restored to the **device**. A returned value of 1 indicates that the registers listed above will be cleared when power is restored to the **device**, see 11.1.3.

10.26.2 *Query structure*

10.26.2.1 Query syntax

The syntax for the Power-on Status Clear query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*PSC?'.

10.26.2.2 Response syntax

The response syntax for the Power-on Status Clear query is defined as only an <NR1 DECIMAL NUMERIC RESPONSE DATA> element. The <NR1 DECIMAL NUMERIC RESPONSE DATA> shall be either a single ISO 7-bit encoded byte for "0" (30, 48 decimal) or "1" (31, 49 decimal).

10.26.3 *Response semantics*

A value of 0 indicates the device's power-on-clear flag is FALSE.

A value of 1 indicates the device's power-on-clear flag is TRUE.

10.26.4 *Related common commands*

*PSC – Implementation requires this optional command. This is the companion command.

10.26.5 *Standard compliance*

Optional

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993 Withdrawn

10.27 **PUD, Instruction de données utilisateur protégées*

10.27.1 *Fonction et prescriptions*

L'instruction de données utilisateur protégées enregistre des données uniques dans l'**appareil**, comme la date d'étalonnage, la durée d'utilisation, les conditions ambiantes et le nombre de contrôles d'inventaire. Un minimum de 63 octets sera prévu. La longueur de cette zone sera spécifiée dans la documentation de l'appareil, voir 4.9.

Les données devront être protégées par certains procédés (tels qu'un mot de passe ou un interrupteur en retrait dont le trou d'accès peut être recouvert par une protection adhésive). Le mécanisme de protection exact doit être décidé par le concepteur de l'appareil.

Les données peuvent être enregistrées seulement lorsque le mécanisme de protection est invalidé. Un mécanisme de protection validé devra générer une erreur d'exécution lorsqu'une instruction *PUD sera reçue.

Le contenu de cette mémoire sert seulement pour le stockage des données et ne devra pas avoir d'effet sur les opérations de l'appareil sauf dans la réponse à l'interrogation *PUD?, voir 10.28.

10.27.2 *Syntaxe*

La syntaxe pour l'instruction de données utilisateurs protégées est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*PUD'.

10.27.3 *Sémantique*

La sémantique de l'élément de <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> est au-delà de l'étendue de cette norme.

10.27.4 *Instructions communes relatives*

L'implémentation de PUD? nécessite cette instruction facultative. Ceci est l'interrogation associée.

10.27.5 *Conformité à la norme*

Facultative

10.27.6 *Traitement de l'erreur*

Lorsque le nombre d'octets de données de 8 bits dans l'élément de <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> dépasse la longueur prévue, une erreur d'exécution devra alors être générée.

Un mécanisme de protection validé générera une erreur d'exécution lorsqu'une instruction *PUD sera reçue.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.27 **PUD, Protected User Data command*

10.27.1 *Function and requirements*

The Protected User Data command stores data unique to the **device** such as calibration date, usage time, environmental conditions, and inventory control numbers. A minimum of 63 bytes shall be provided. The size of this area shall be specified in the device documentation, see 4.9.

The data shall be protected by some means (such as a password or a recessed switch whose access hole can be covered by a protective sticker). The exact protection mechanism is a device designer decision.

Data can be stored only when the protection mechanism is disabled. An enabled protection mechanism shall cause an Execution Error to be generated when a *PUD command is received.

The contents of this memory is for data storage only and shall have no effect on device operation except in the response to the *PUD? query, see 10.28.

10.27.2 *Syntax*

The syntax for the Protected User Data command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by an <ARBITRARY BLOCK PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*PUD'.

10.27.3 *Semantics*

The semantic meaning of the <ARBITRARY BLOCK PROGRAM DATA> element is beyond the scope of this standard.

10.27.4 *Related common commands*

*PUD? – Implementation requires this optional command. This is the companion query.

10.27.5 *Standard compliance*

Optional

10.27.6 *Error handling*

When the number of <8-bit data bytes> in the <ARBITRARY BLOCK PROGRAM DATA> element is more than the size provided, then an Execution Error shall be generated.

An enabled protection mechanism shall cause an Execution Error to be generated when a *PUD command is received.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.28 **PUD?, Interrogation de données utilisateur protégées*

10.28.1 *Fonction et prescriptions*

L'interrogation de données utilisateur protégées permet au programmeur d'extraire le contenu de la zone de stockage *PUD. Voir instruction *PUD, 10.27.

10.28.2 *Structure d'interrogation*

10.28.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de données utilisateur protégées est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*PUD?'.

10.28.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de données utilisateur protégées est définie comme un élément de <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE>.

10.28.3 *Sémantique de réponse*

Le contenu des <DONNÉES DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE> devra être la dernière donnée envoyée à, et acceptée par l'**appareil** en tant que paramètre de l'instruction *PUD. Voir 10.27.

10.28.4 *Instructions communes relatives*

L'implémentation de *PUD nécessite cette instruction facultative. Ceci est l'instruction associée.

10.28.5 *Conformité à la norme*

Facultative

10.29 **RCL, Instruction de rappel*

10.29.1 *Fonction et prescriptions*

L'instruction de rappel remet l'**appareil** dans un état initial à partir d'une copie enregistrée dans une mémoire locale. L'étendue de l'état sauvegardé devra figurer explicitement dans la documentation de l'appareil. L'étendue de l'état sauvegardé devra être identique à l'étendue de l'état affecté par une instruction *RST, voir 10.32.

10.29.2 *Syntaxe*

La syntaxe pour l'instruction de rappel est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*RCL'. La valeur de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> devra être dans la gamme 0 à limite supérieure spécifiée par l'appareil.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.28 **PUD?, Protected User Data query*

10.28.1 *Function and requirements*

The Protected User Data query allows the programmer to retrieve the contents of the *PUD storage area. See *PUD command, 10.27.

10.28.2 *Query structure*

10.28.2.1 Query syntax

The syntax for the Protected User Data query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*PUD?'.

10.28.2.2 Response syntax

The response syntax for the Protected User Data query is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

10.28.3 *Response semantics*

The data content of the <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> shall be the data last sent to, and accepted by, the **device** as a parameter to the *PUD command. See 10.27.

10.28.4 *Related common commands*

*PUD – Implementation requires this optional command. This is the companion command.

10.28.5 *Standard compliance*

Optional

10.29 **RCL, Recall command*

10.29.1 *Function and requirements*

The Recall command restores the state of the **device** from a copy stored in local memory. The scope of the saved state shall be explicitly stated in the device documentation. The scope of the saved state shall be identical with the scope of the state affected by an *RST command, see 10.32.

10.29.2 *Syntax*

The syntax for the Recall command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*RCL'. The <DECIMAL NUMERIC PROGRAM DATA> value shall be in the range 0 through a device specified upper bound.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.29.3 *Sémantique*

L'élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> devra être arrondi à un nombre entier avant d'être interprété par l'appareil.

L'**appareil** devra rappeler l'état de l'**appareil** à partir d'un registre de mémoire associé au nombre entier reçu. L'étendue de l'état rétabli est la même que celle associée à l'instruction *RST.

La limite supérieure pour le nombre de registres devra être la même pour les deux instructions *SAV et *RCL, voir 10.33.

La valeur zéro est recommandée pour tout registre spécial, tel qu'une configuration prédéfinie à la mise sous tension.

10.29.4 *Instructions communes relatives*

*RST – L'étendue définie par l'appareil de *RST détermine l'étendue de l'instruction *RCL.

L'implémentation de *SAV nécessite cette instruction facultative.

10.29.5 *Conformité à la norme*

Facultative

10.29.6 *Traitement de l'erreur*

Une valeur interprétée en dehors de la gamme admissible entraînera une erreur d'exécution.

10.30 **RDT, Instruction de transfert de description de ressource*

10.30.1 *Fonction et prescriptions*

L'instruction de transfert de description de ressource permet à une description de ressource d'être enregistrée dans un **appareil**.

Cette instruction rappelle des informations décrivant des caractéristiques telles que performances de l'**appareil**, codes de programmation, etc. dans une structure standard.

Les données seront protégées par certains procédés (tels qu'un mot de passe ou un interrupteur en retrait dont le trou d'accès peut être recouvert par une protection adhésive). Le mécanisme de protection exact doit être décidé par le concepteur de l'appareil.

Les données ne peuvent être enregistrées que lorsque le mécanisme de protection est invalidé.

Cette mémoire sert uniquement pour le stockage des données et son contenu ne devra pas avoir d'effet sur les opérations de l'appareil, sauf dans la réponse à l'interrogation *RDT?, voir 10.31.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.29.3 *Semantics*

The <DECIMAL NUMERIC PROGRAM DATA> element shall be rounded to an integer before interpretation by the device.

The **device** shall recall the state of the **device** from a memory register associated with the received integer. The scope of the restored state is the same as the one associated with an *RST command.

The upper bound for the register number shall be the same for both the *SAV and *RCL commands, see 10.33.

The value zero is recommended for any special purpose register, such as a predefined power-on setup.

10.29.4 *Related common commands*

*RST – The device-defined scope of *RST determines the scope of the *RCL command.

*SAV – Implementation requires this optional command.

10.29.5 *Standard compliance*

Optional

10.29.6 *Error handling*

An interpreted value outside of the allowable range shall cause an Execution Error.

10.30 **RDT, Resource Description Transfer command*

10.30.1 *Function and requirements*

The Resource Description Transfer command allows a Resource Description to be stored in a **device**.

This command retrieves information describing such characteristics as **device** performance, programming codes, et cetera, in a standard structure.

The data shall be protected by some means (such as a password or a recessed switch whose access hole can be covered by a protective sticker). The exact protection mechanism is a device designer decision.

Data can be stored only when the protection mechanism is disabled.

This memory is for data storage only and its contents shall have no effect on device operation except in the response to the *RDT? query, see 10.31.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.30.2 *Syntaxe*

La syntaxe pour l'instruction de transfert de description de ressource est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*RDT'.

10.30.3 *Sémantique*

Le contenu en données des <DONNÉES DE PROGRAMMATION DE BLOC ARBITRAIRE> est au-delà de la portée de cette norme.

10.30.4 *Instructions communes relatives*

L'implémentation de *RDT? nécessite cette instruction. Ceci est l'interrogation associée.

10.30.5 *Conformité à la norme*

Facultative

10.30.6 *Traitement de l'erreur*

10.30.6.1 Données protégées

Un mécanisme de protection validé devra générer une erreur d'exécution lors de la réception d'une instruction *RDT.

10.30.6.2 Bloc surdimensionné

Un bloc plus long que ce que l'**appareil** peut accepter devra générer une erreur d'exécution. La documentation de l'appareil doit spécifier la longueur maximale de bloc.

10.31 **RDT?, Interrogation de transfert de description de ressource*

10.31.1 *Fonction et prescriptions*

L'interrogation de transfert de description de ressource permet à une description de ressource d'être extraite d'un **appareil**. La description de ressource peut être stockée dans une mémoire morte ou une mémoire vive sélectionnables par l'instruction *RDT.

10.31.2 *Structure d'interrogation*

10.31.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de transfert de description de ressource est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*RDT?'.

10.31.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de transfert de description de ressource est définie comme un élément de <DONNÉE DE RÉPONSE DE BLOC ARBITRAIRE DE LONGUEUR DÉFINIE>.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.30.2 *Syntax*

The syntax for the Resource Description Transfer command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by an <ARBITRARY BLOCK PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*RDT'.

10.30.3 *Semantics*

The data content of the <ARBITRARY BLOCK PROGRAM DATA> is beyond the scope of this standard.

10.30.4 *Related common commands*

*RDT? – Implementation requires this command. This is the companion query.

10.30.5 *Standard compliance*

Optional

10.30.6 *Error handling*

10.30.6.1 Protected data

An enabled protection mechanism shall cause an Execution Error to be generated when an *RDT command is received.

10.30.6.2 Oversized block

A block length greater than the **device** can accept shall generate an Execution Error. The device's documentation must specify the maximum block length.

10.31 **RDT?, Resource Description Transfer query*

10.31.1 *Function and requirements*

The Resource Description Transfer query allows a Resource Description to be retrieved from a **device**. The Resource Description may be stored in a read-only memory or in a read-write memory settable by the *RDT command.

10.31.2 *Query structure*

10.31.2.1 Query syntax

The syntax for the Resource Description Transfer query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*RDT?'.

10.31.2.2 Response syntax

The response syntax for the Resource Description Transfer query is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.31.3 *Sémantique de réponse*

Un bloc de longueur différente de zéro devra contenir des données de description de ressources, c'est-à-dire soit la description définie de ressources incorporées, soit exactement ce qui a été envoyé à l'**appareil** avec la dernière instruction commune *RDT.

Un bloc de longueur zéro indique que l'**appareil** ne contient pas de données de description de ressources.

10.31.4 *Instructions communes relatives*

*RDT – est l'instruction associée.

10.31.5 *Conformité à la norme*

Facultative

10.32 **RST, Instruction de remise à zéro*

10.32.1 *Fonction et prescriptions*

L'instruction de remise à zéro réalise une remise à l'état initial de l'appareil. L'instruction de remise à zéro est le troisième niveau de remise à zéro dans une stratégie de remise à zéro qui en compte trois, voir 17.1.2.

L'instruction de remise à zéro devra réaliser les actions suivantes:

1) Mettre les fonctions spécifiques à l'appareil dans un état connu indépendant de l'historique passé de l'**appareil**. Des instructions spécifiques à l'appareil peuvent être fournies pour programmer un état de remise à zéro différent de l'original fourni en usine.
2) Mettre les macro-instructions définies par *DDT dans un état défini par l'appareil, voir 10.4.
3) Invalider les macro-instructions, voir 10.8.
4) Faire entrer l'appareil dans l'état OCIS, voir 12.5.2.
5) Faire entrer l'appareil dans l'état OQIS, voir 12.5.3.

L'instruction de remise à zéro ne devra PAS affecter explicitement:

1) L'état de l'interface CEI 625-1.
2) L'adresse CEI 625-1 sélectionnée de l'appareil.
3) La file d'attente de sortie.
4) Le positionnement de validation de registre d'état standard.
5) Le positionnement de validation d'état d'événement standard.
6) Le positionnement de l'indicateur d'effacement à la mise sous tension.
7) Les macro-instructions définies avec l'instruction de définition du contenu des macro-instructions.
8) Les données d'étalonnage qui affectent les spécifications de l'appareil.
9) La réponse à l'interrogation de données utilisateur protégées.
10) La réponse à l'interrogation de transfert de description de ressource.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.31.3 *Response semantics*

A non-zero length block shall contain Resource Description data that is either the defined built-in Resource Description or exactly what was sent to the **device** with the last *RDT common command.

A zero-length block indicates the **device** contains no Resource Description data.

10.31.4 *Related common commands*

*RDT – This is the companion command.

10.31.5 *Standard compliance*

Optional

10.32 **RST, Reset command*

10.32.1 *Function and requirements*

The Reset command performs a device reset. The Reset command is the third level of reset in a three-level reset strategy, see 17.1.2.

The Reset command shall do the following:

1) Set the device-specific functions to a known state that is independent of the past-use history of the **device**. Device-specific commands may be provided to program a different reset state than the original factory-supplied one.

2) Set the macro defined by *DDT to a device-defined state, see 10.4.

3) Disable macros, see 10.8.

4) Force the device into the OCIS state, see 12.5.2.

5) Force the device into the OQIS state, see 12.5.3.

The Reset command explicitly shall NOT affect the following:

1) The state of the IEC 625-1 interface.

2) The selected IEC 625-1 address of the device.

3) The Output Queue.

4) The Standard Status Register Enable setting.

5) The Standard Event Status Enable setting.

6) The power-on-clear flag setting.

7) Macros defined with the Define Macro Contents command.

8) Calibration data that affects device specifications.

9) The Protected User Data query response.

10) The Resource Description Transfer query response.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.32.2 *Syntaxe*

La syntaxe pour l'instruction de remise à zéro est définie comme uniquement un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*RST'.

10.32.3 *Sémantique*

Non applicable

10.32.4 *Instructions communes relatives*

Voir 10.32.1.

10.32.5 *Conformité à la norme*

Obligatoire

10.33 **SAV, Instruction de sauvegarde*

10.33.1 *Fonction et prescriptions*

L'instruction de sauvegarde enregistre l'état courant de l'**appareil** en mémoire locale. L'étendue de l'état sauvegardé devra figurer explicitement dans la documentation de l'appareil. L'étendue de l'état sauvegardé devra être identique à l'étendue de l'état affecté par une instruction *RST.

10.33.2 *Syntaxe*

La syntaxe pour l'instruction de sauvegarde est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*SAV'. La valeur de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> devra être dans la gamme 0 à limite supérieure spécifiée par l'appareil.

10.33.3 *Sémantique*

L'élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> devra être arrondi à un nombre entier avant interprétation par l'appareil.

L'**appareil** devra sauvegarder l'état courant de l'**appareil** dans un registre de mémoire associé au nombre entier interprété. L'étendue de l'état sauvegardé est la même que celle associée à une instruction *RST ou *LRN?.

La limite supérieure pour le nombre de registres devra être la même pour les deux instructions *SAV et *RCL, voir 10.29.

La valeur zéro est recommandée pour tout registre spécial, tel qu'une configuration prédéfinie à la mise sous tension.

10.33.4 *Instructions communes relatives*

L'implémentation de *RCL nécessite cette instruction facultative.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.32.2 *Syntax*

The syntax for the Reset command is defined as only a <COMMAND PROGRAM HEADER>. The <COMMAND PROGRAM HEADER> is defined as '*RST'.

10.32.3 *Semantics*

Not applicable

10.32.4 *Related common commands*

See 10.32.1.

10.32.5 *Standard compliance*

Mandatory

10.33 **SAV, Save command*

10.33.1 *Function and requirements*

The Save command stores the current state of the **device** in local memory. The scope of the saved state shall be explicitly stated in the device documentation. The scope of the saved state shall be identical with the scope of the state affected by an *RST command.

10.33.2 *Syntax*

The syntax for the Save command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*SAV'. The <DECIMAL NUMERIC PROGRAM DATA> value shall be in the range 0 through a device specified upper bound.

10.33.3 *Semantics*

The <DECIMAL NUMERIC PROGRAM DATA> element shall be rounded to an integer before interpretation by the device.

The **device** shall save the current state of the **device** in a memory register associated with the interpreted integer. The scope of the saved state is the same as the one associated with an *RST or *LRN? command.

The upper bound for the register number shall be the same for both the *SAV and *RCL commands, see 10.29.

The value zero is recommended for any special purpose register, such as a predefined power-on setup.

10.33.4 *Related common commands*

*RCL – Implementation requires this optional command.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.33.5 *Conformité à la norme*

Facultative

10.33.6 *Traitement de l'erreur*

Une valeur en dehors de la gamme admissible entraînera une erreur d'exécution.

10.34 **SRE, Instruction de validation de demande de service*

10.34.1 *Fonction et prescriptions*

L'instruction de validation de demande de service positionne les bits du registre de validation de demande de service comme défini en 11.3.2.

10.34.2 *Syntaxe*

La syntaxe pour l'instruction de validation de demande de service est définie comme un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> suivi par un <SÉPARATEUR D'EN-TÊTES DE PROGRAMMATION> suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DECIMALE>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*SRE'. La valeur de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> est dans la gamme de 0 à 255.

10.34.3 *Sémantique*

La <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>, lorsqu'elle est arrondie à une valeur entière et exprimée en base 2 (binaire), représente les valeurs des bits du registre de validation de demande de service. Voir 11.3.2.3.

Pour tous les bits sauf le bit 6, un bit de valeur 1 devra indiquer une condition de validation. Un bit de valeur 0 indiquera une condition d'invalidation. Voir 11.3.2.3. La valeur du bit 6 devra être ignorée.

10.34.4 *Instructions communes relatives*

*PSC – détermine si le registre de validation de demande de service est effacé à la mise sous tension.

*SRE? – est l'interrogation associée.

10.34.5 *Conformité à la norme*

Obligatoire

10.34.6 *Traitement de l'erreur*

L'**appareil** générera une erreur d'exécution si un paramètre hors gamme est reçu.

10.35 **SRE?, Interrogation de validation de demande de service*

10.35.1 *Fonction et prescriptions*

L'interrogation de validation de demande de service permet au programmeur de déterminer le contenu courant du registre de validation de demande de service, voir 11.3.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.33.5 *Standard compliance*

Optional

10.33.6 *Error handling*

A value outside the allowable range shall cause an Execution Error.

10.34 **SRE, Service Request Enable command*

10.34.1 *Function and requirements*

The Service Request Enable command sets the Service Request Enable Register bits as defined in 11.3.2.

10.34.2 *Syntax*

The syntax for the Service Request Enable command is defined as a <COMMAND PROGRAM HEADER> followed by a <PROGRAM HEADER SEPARATOR> followed by a <DECIMAL NUMERIC PROGRAM DATA> element. The <COMMAND PROGRAM HEADER> is defined as '*SRE'. The <DECIMAL NUMERIC PROGRAM DATA> value is in the range of 0 to 255.

10.34.3 *Semantics*

The <DECIMAL NUMERIC PROGRAM DATA> when rounded to an integer value and expressed in base 2 (binary) represents the bit values of the Service Request Enable Register. See 11.3.2.3.

For all bits except bit 6, a bit value of one shall indicate an enabled condition. A bit value of zero shall indicate a disabled condition. See 11.3.2.3. The bit value of bit 6 shall be ignored.

10.34.4 *Related common commands*

*PSC – Determines whether the Service Request Enable Register is cleared at power-on.

*SRE? – This is the companion query.

10.34.5 *Standard compliance*

Mandatory

10.34.6 *Error handling*

The **device** shall generate an Execution Error if an out-of-range parameter is received.

10.35 **SRE?, Service Request Enable query*

10.35.1 *Function and requirements*

The Service Request Enable query allows the programmer to determine the current contents of the Service Request Enable Register, see 11.3.2.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.35.2 *Structure d'interrogation*

10.35.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de validation de demande de service est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*SRE?'.

10.35.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de validation de demande de service est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> devra être dans la gamme 0 à 63 ou 128 à 191.

10.35.3 *Sémantique de réponse*

Convertie en binaire (base 2), la <DONNÉE DE RÉPONSE NUMÉRIQUE RN1> représente les valeurs courantes des bits du registre de validation de demande de service. Le bit 6 de la représentation binaire devra toujours être envoyé avec la valeur zéro.

10.35.4 *Instructions communes relatives*

*SRE – est l'instruction associée.

10.35.5 *Conformité à la norme*

Obligatoire

10.36 **STB?, Interrogation de lecture de l'octet d'état*

10.36.1 *Fonction et prescriptions*

L'interrogation de lecture de l'octet d'état permet au programmeur de lire l'octet d'état et les bits du récapitulatif principal.

10.36.2 *Structure d'interrogation*

10.36.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation de lecture de l'octet d'état est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*STB?'.

10.36.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation de lecture de l'octet d'état est définie comme un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE RN1>. La valeur de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> devra être dans la gamme de 0 à 255.

10.36.3 *Sémantique de réponse*

La valeur entière de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> exprimée en base 2 (binaire) représente les valeurs des bits du registre d'octets d'état, voir 11.2.2.2.

Le bit MSS (Master Summary Status = Etat récapitulatif principal), et non le message RQS, est indiqué sur le bit 6, voir 11.2.2.3.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.35.2 *Query structure*

10.35.2.1 Query syntax

The syntax for the Service Request Enable query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*SRE?'.

10.35.2.2 Response syntax

The response syntax for the Service Request Enable query is defined as an <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> shall be in the range 0 to 63 or 128 to 191.

10.35.3 *Response semantics*

When converted to binary (base 2), the <NR1 NUMERIC RESPONSE DATA> represents the current bit values of the Service Request Enable Register. Bit 6 of the binary representation shall always be sent with value zero.

10.35.4 *Related common commands*

*SRE – This is the companion command.

10.35.5 *Standard compliance*

Mandatory

10.36 **STB?, Read Status Byte query*

10.36.1 *Function and requirements*

The Read Status Byte query allows the programmer to read the status byte and Master Summary Status bit.

10.36.2 *Query structure*

10.36.2.1 Query syntax

The syntax for the Read Status Byte query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*STB?'.

10.36.2.2 Response syntax

The response syntax for the Read Status Byte query is defined as an <NR1 NUMERIC RESPONSE DATA> element. The <NR1 NUMERIC RESPONSE DATA> value shall be in the range of 0 to 255.

10.36.3 *Response semantics*

The <NR1 NUMERIC RESPONSE DATA> integer value expressed in base 2 (binary) represents the bit values of the Status Byte Register, see 11.2.2.2.

The MSS (Master Summary Status) bit, not the RQS message, is reported on bit-6, see 11.2.2.3.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.36.4 *Instructions communes relatives*

Aucune

10.36.5 *Conformité à la norme*

Obligatoire

10.37 **TRG, Instruction de déclenchement*

10.37.1 *Fonction et prescriptions*

L'instruction de déclenchement est l'analogue spécifique à l'appareil du message d'interface (GET) de déclenchement d'exécution de groupe défini dans la CEI 625-1, et possède exactement les mêmes effets qu'un GET lorsqu'elle est reçue, analysée syntaxiquement et exécutée par l'**appareil**. L'opération GET est traitée en détail dans 6.1.4.2.5.

Noter que l'instruction *TRG sans *DDT implémenté (voir 10.4) peut générer une réponse.

10.37.2 *Syntaxe*

La syntaxe pour l'instruction de déclenchement est définie comme uniquement un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*TRG'.

10.37.3 *Sémantique*

Non applicable

10.37.4 *Instructions communes relatives*

L'instruction facultative *DDT peut être utilisée pour définir l'action de *TRG.

10.37.5 *Conformité à la norme*

Obligatoire pour les **appareils** implémentant le sous-ensemble DT1.

10.38 **TST?, Interrogation d'auto-test*

10.38.1 *Fonction et prescriptions*

L'interrogation d'auto-test entraîne un auto-test interne et place une réponse dans la file d'attente de sortie indiquant si l'**appareil** a ou n'a pas achevé l'auto-test sans aucune erreur détectée. Facultativement, l'information sur la raison pour laquelle l'auto-test ne s'est pas terminé peut être contenue dans la réponse. L'étendue de l'auto-test interne doit figurer dans la documentation de l'appareil, voir 4.9.

L'interrogation *TST? ne devra nécessiter l'interaction d'aucun opérateur local. Elle ne devra pas créer de conditions de bus qui sont des violations des normes CEI 625-1 ou CEI 625-2. Par ailleurs, l'étendue de l'auto-test est complètement à la discrétion du concepteur de l'appareil.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.36.4 *Related common commands*

None

10.36.5 *Standard compliance*

Mandatory

10.37 **TRG, Trigger command*

10.37.1 *Function and requirements*

The Trigger command is the device-specific analog of the IEC 625-1 defined Group Execute Trigger (GET) interface message, and has exactly the same effect as a GET when received, parsed, and executed by the **device**. GET operation is discussed in detail in 6.1.4.2.5.

Note that the *TRG command without *DDT implemented (see 10.4) may generate a response.

10.37.2 *Syntax*

The syntax for the Trigger command is defined as only a <COMMAND PROGRAM HEADER>. The <COMMAND PROGRAM HEADER> is defined as '*TRG'.

10.37.3 *Semantics*

Not applicable

10.37.4 *Related common commands*

*DDT – This optional command may be used to define the action of *TRG.

10.37.5 *Standard compliance*

Mandatory for **devices** implementing the DT1 subset.

10.38 **TST?, Self-Test query*

10.38.1 *Function and requirements*

The Self-Test query causes an internal self-test and places a response into the Output Queue indicating whether or not the **device** completed the self-test without any detected errors. Optionally, information on why the self-test was not completed may be contained in the response. The scope of the internal self-test shall appear in the device documentation, see 4.9.

The *TST? query shall not require any local operator interaction. It shall not create bus conditions that are violations to the IEC 625-1 or IEC 625-2 standards. Otherwise, the scope of the self-test is completely at the discretion of the device designer.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Après exécution réussie de *TST?, les positionnements de l'**appareil** doivent être remis à leurs valeurs avant *TST?, réglés à des valeurs fixes, connues, qui sont spécifiées dans la documentation de l'appareil, ou réglés à des valeurs définies par l'utilisateur et stockées dans la mémoire locale.

10.38.2 *Structure d'interrogation*

10.38.2.1 Syntaxe d'interrogation

La syntaxe pour l'interrogation d'auto-test est définie comme uniquement un <EN-TÊTE D'INTERROGATION DE PROGRAMMATION>. L'<EN-TÊTE D'INTERROGATION DE PROGRAMMATION> est défini par '*TST?'.

10.38.2.2 Syntaxe de réponse

La syntaxe de réponse pour l'interrogation d'auto-test est définie comme une <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. La valeur de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> devra être dans la gamme –32767 à 32767.

10.38.3 *Sémantique de réponse*

Une <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> avec la valeur 0 indique que l'auto-test a été achevé sans erreurs détectées.

Une <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> avec la valeur 1 indique que l'auto-test n'a pas été achevé ou a été achevé avec des erreurs détectées.

La sémantique de réponse différente de zéro devra par ailleurs être complètement à la discrétion du concepteur de l'appareil.

10.38.4 *Instructions communes relatives*

Non applicables

10.38.5 *Conformité à la norme*

Obligatoire

10.39 **WAI, Instruction d'attente de poursuite*

10.39.1 *Fonction et prescriptions*

L'instruction d'attente de poursuite empêchera l'exécution par l'**appareil** de toute autre instruction ou interrogation jusqu'à ce que la signalisation "pas d'opération en cours" soit VRAIE. Voir 12.5.1.

10.39.2 *Syntaxe*

La syntaxe pour l'instruction d'attente de poursuite est définie comme uniquement un <EN-TÊTE D'INSTRUCTION DE PROGRAMMATION>. L'<EN-TÊTE D'INSTRUCTION DE PROGRAMMATION> est défini par '*WAI'.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Upon successful completion of *TST?, the **device** settings shall be restored to their values prior to the *TST?, set to fixed, known values which are stated in the device documentation, or set to values defined by the user and stored in local memory.

10.38.2 *Query structure*

10.38.2.1 Query syntax

The syntax for the Self-Test query is defined as only a <QUERY PROGRAM HEADER>. The <QUERY PROGRAM HEADER> is defined as '*TST?'.

10.38.2.2 Response syntax

The response syntax for the Self-Test query is defined as an <NR1 NUMERIC RESPONSE DATA>. The <NR1 NUMERIC RESPONSE DATA> value shall be in the range –32767 to 32767.

10.38.3 *Response semantics*

An <NR1 NUMERIC RESPONSE DATA> with the value of 0 indicates the self-test has completed without errors detected.

An <NR1 NUMERIC RESPONSE DATA> with the value not equal to zero indicates the self-test was not completed or was completed with errors detected.

The semantics of the non-zero response otherwise shall be completely at the discretion of the device designer.

10.38.4 *Related common commands*

Not applicable

10.38.5 *Standard compliance*

Mandatory

10.39 **WAI, Wait-to-Continue command*

10.39.1 *Function and requirements*

The Wait-to-Continue command shall prevent the **device** from executing any further commands or queries until the No-Operation-Pending flag is TRUE. See 12.5.1.

10.39.2 *Syntax*

The syntax for the Wait-to-Continue command is defined as only a <COMMAND PROGRAM HEADER>. The <COMMAND PROGRAM HEADER> is defined as '*WAI'.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

10.39.3 *Sémantique*

Non applicable

10.39.4 *Instructions communes relatives*

*OPC, *OPC? dans les actions de protocole de synchronisation.

10.39.5 *Conformité à la norme*

Obligatoire

11 Indications sur l'état de l'appareil

Cet article spécifie les exigences de l'**appareil** entraînant les fonctions de demande de service et interrogation parallèle CEI 625-1. Ces exigences sont fondées sur les spécifications CEI 625-1 et les étendent pour fournir une structure d'indication d'état détaillée.

Un modèle complet est défini pour toutes les indications d'état. L'octet d'état CEI 625-1 est une partie de ce modèle. Des messages d'état spécifiques et exigés sont définis. Un modèle d'indication d'état spécifique à l'appareil convenant pour des exigences spécifiques, et spécifiques à l'appareil, est présenté.

Cette section fournit une méthode pour transférer l'octet d'état CEI 625-1 au **contrôleur** en utilisant soit l'interrogation série CEI 625-1, soit une interrogation commune définie CEI 625-2. Des instructions et interrogations communes supplémentaires sont fournies pour obtenir des informations d'état plus détaillées.

11.1 *Vue d'ensemble*

11.1.1 *Opération*

La figure 11-1 représente le diagramme de blocs de la structure d'indication d'état CEI 625-2.

L'indication d'état CEI 625-2 utilise l'octet d'état CEI 625-1 avec des règles et structures de données supplémentaires.

Le registre d'octets d'état est composé de sept "messages récapitulatifs" de bits simples (figure 11-1). Chaque message récapitulatif résume une structure de données d'état de recouvrement.

Les messages récapitulatifs suivent toujours l'état en cours de la structure de données d'état associée. Les messages récapitulatifs sont seulement effacés par une action du programme d'application qui efface la structure de données d'état associée.

La génération et la validation de demande de service détermine si un ou plusieurs des sept messages récapitulatifs générera le message rsv et obligera ainsi l'**appareil** à demander un service en utilisant sa fonction de demande de service CEI 625-1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

10.39.3 *Semantics*

Not applicable

10.39.4 *Related common commands*

*OPC, *OPC? in synchronization protocol actions.

10.39.5 *Standard compliance*

Mandatory

11 Device status reporting

This clause specifies **device** requirements involving IEC 625-1 Service Request and Parallel Poll functions. These requirements build upon and extend the IEC 625-1 specifications to provide a detailed status reporting structure.

A complete model is defined for all status reporting. The IEC 625-1 status byte is part of the model. Specific and required status messages are defined. A device-specific status reporting model suitable for specific device-specific requirements is presented.

This section provides a method to transfer the IEC 625-1 status byte to the **controller** using either the IEC 625-1 serial poll or an IEC 625-2-defined common query. Additional common commands and queries are provided to obtain more detailed status information.

11.1 *Overview*

11.1.1 *Operation*

Figure 11-1 shows the block diagram of the IEC 625-2 Status Reporting Structure.

IEC 625-2 status reporting utilizes the IEC 625-1 status byte with additional data structures and rules.

The Status Byte Register is composed of seven single-bit "summary-messages" (figure 11-1). Each summary-message summarizes an overlaying Status Data Structure.

Summary-messages always track the current status of the associated Status Data Structure. The summary-messages are cleared only by some action of the application program which clears the associated Status Data Structure.

Service Request Enabling and Generation determines if one or more of the seven summary-messages will generate the rsv message and thus cause the **device** to request service using its IEC 625-1 Service Request function.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Les structures de données d'état associées aux messages récapitulatifs sont définies dans cet article. Elles suivent, soit:

– un ensemble de registres pour enregistrer les événements et les conditions de l'appareil (modèle de registre);

– soit une file d'attente pour enregistrer l'état séquentiel ou d'autres informations (modèle de file d'attente).

Seulement deux des sept structures de données d'état disponibles sont complètement définies par cette norme:

– la "file d'attente de sortie" est une structure de modèle de file d'attente qui est résumée par le message récapitulatif MAV (Message Available) (= message disponible) sur le bit 4. (DIO5 – Voir figure 11-2);

– l'"état d'événements standard" est une structure de modèle de registre qui est résumée par le message récapitulatif ESB (Event Status Bit) (= bit d'événement standard) sur le bit 5. (DIO6 – Voir figure 11-2).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

The Status Data Structures associated with summary-messages are defined in this clause. They follow either:

- a set of registers to record device events and conditions (register-model);

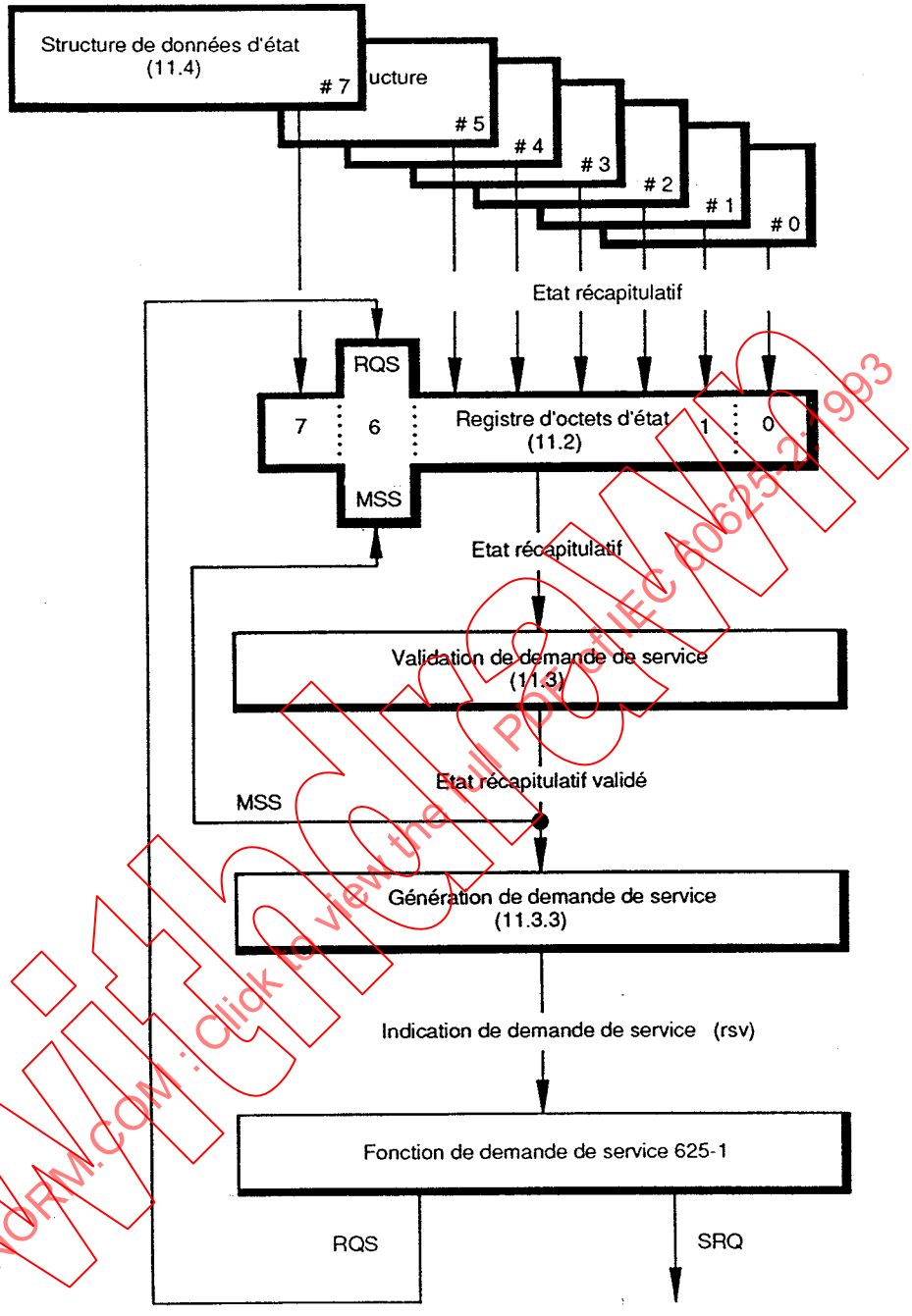
- or a Queue to record sequential status or other information (queue model).

Only two of the seven available Status Data Structures are completely defined by this standard:

- the "Output Queue" is a queue model structure which is summarized by the MAV (Message Available) summary-message on bit 4. (DIO5 – See figure 11-2);

- the "Standard Event Status" is a register-model structure which is summarized by the ESB (Event Status Bit) summary-message on bit 5. (DIO6 – See figure 11-2).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

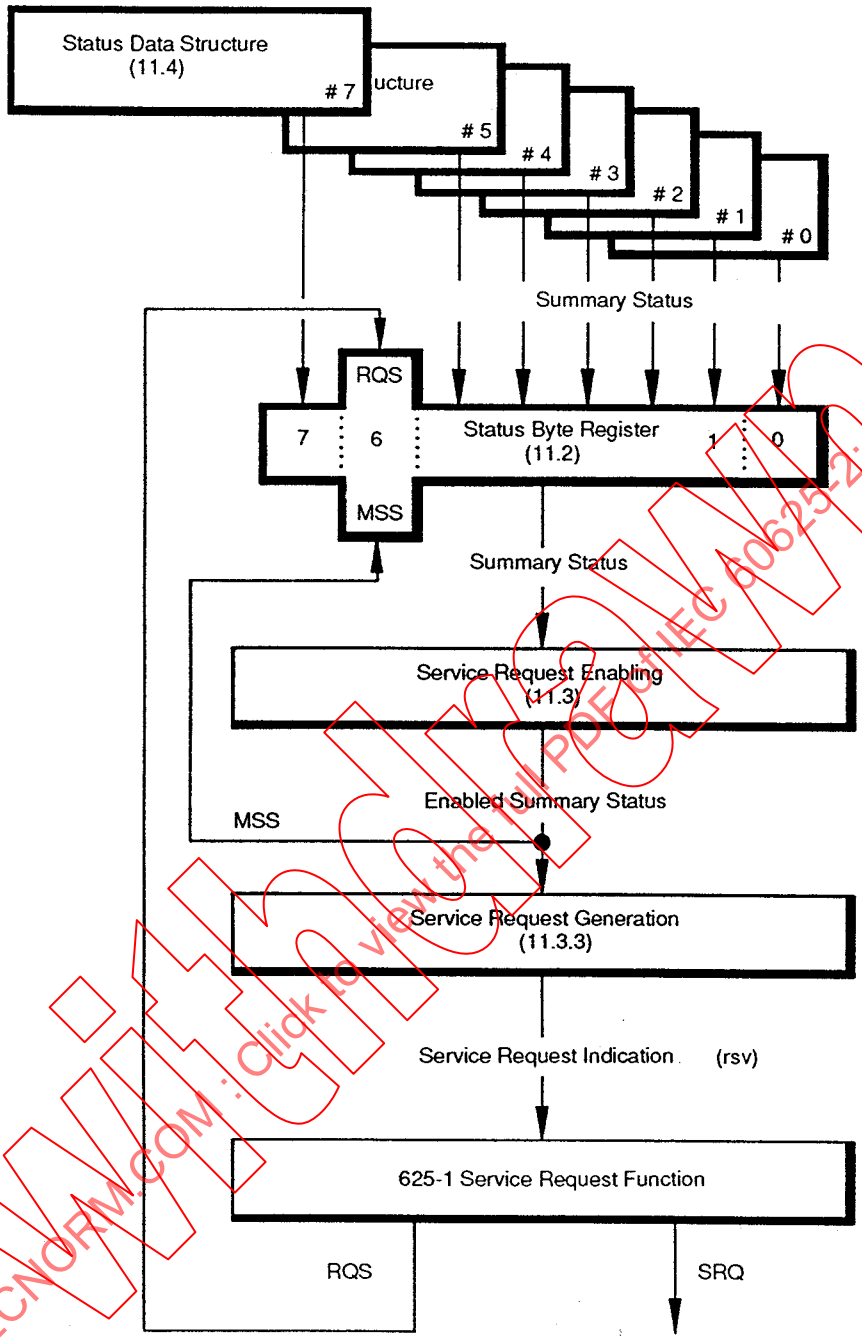


IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figure 11-1 – Vue d'ensemble de la structure d'indication d'état CEI 625-2

11.1.2 *Récapitulatif des instructions communes relatives*

Le tableau 11-1 fait la liste des instructions communes définies dans la CEI 625-2 relatives à cet article. Une courte description et les références des articles pour plus ample information sont données pour chaque instruction.



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Figure 11-1 – IEC 625-2 status reporting structure overview

11.1.2 *Summary of related common commands*

Table 11-1 lists the IEC 625-2 defined common commands related to this clause. A short description and clause references for further information are given for each command.

Tableau 11-1 – Instructions communes d'indication d'état

*STB?	Renvoie une NR1, qui est la valeur de l'octet d'état CEI 625-1 et du message récapitulatif MSS (Master Summary Status) (= état récapitulatif principal). Voir 10.36 et 11.2.2.2.
*OPC	Positionne le bit d'événement "opération achevée" dans le registre d'état d'événements standards lorsque toutes les opérations de l'**appareil** sélectionnées en cours ont été terminées. Voir 10.18 et 12.5.2.
*OPC?	Place un "1" dans la file d'attente de sortie lorsque toutes les opérations sélectionnées en cours sont terminées, ce qui entraîne alors la génération du message récapitulatif MAV (message disponible). Voir 10.19 et 12.5.3.
*CLS	Efface tous les registres d'événements résumés dans l'octet d'état. Toutes les files d'attente, sauf la file d'attente de sortie, qui sont résumées dans l'octet d'état sont vidées. On impose à l'appareil l'état inactif d'instruction "opération achevée" et l'état inactif d'interrogation "opération achevée". (Voir 10.3, 11.2.4, 12.5.2 et 12.5.3).
*ESR?	Renvoie une NR1, qui est la valeur du registre d'état d'événements standards. Voir 10.12 et 11.5.1.
*ESE NRf	Positionne les bits du registre de validation d'état d'événements standards. Voir 10.10 et 11.4.2.3.3.
*ESE?	Renvoie une NR1, qui est la valeur du registre de validation d'état d'événements standards. Voir 10.11 et 11.4.2.3.2.
*IST?	Renvoie une NR1, qui est la valeur du message local ist (individual status) (= état individuel) CEI 625-1. Voir 10.15 et 11.6.2.
*SRE NRf	Positionne les bits du registre de validation de demande de service. Voir 10.34 et 11.3.2.3.
*SRE?	Renvoie une NR1, qui est la valeur du registre de validation de demande de service. Voir 10.35 et 11.3.2.2.
*PRE NRf	Positionne les bits du registre de validation d'interrogation parallèle. Voir 10.23 et 11.6.1.3.
*PRE?	Renvoie une NR1, qui est la valeur du registre de validation d'interrogation parallèle. Voir 10.24 et 11.6.1.2.
*PSC NRf	Renvoie une NR1, qui est la valeur de l'état de l'indicateur d'effacement d'état à la mise sous tension. Voir 10.26, 11.1.3 et 5.12.
*PSC?	Renvoie une NR1, qui est la valeur de l'état de l'indicateur d'effacement d'état à la mise sous tension. Voir 10.26, 11.1.3 et 5.12.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.1.3 *Opérations relatives définies dans la CEI 625-1*

Le tableau 11-2 fait la liste des opérations définies dans la CEI 625-1 relatives à cet article. Une courte description et les références des articles pour plus ample information sont incluses avec chaque opération.

Table 11-1 – Status reporting common commands

*STB?	Returns an NR1, which is the value of the IEC 625-1 status byte and the MSS (Master Summary Status) summary-message. See 10.36 and 11.2.2.2.
*OPC	Sets the "Operation Complete" event bit in the Standard Event Status Register when all selected pending **device** operations have been completed. See 10.18 and 12.5.2.
*OPC?	Places a "1" in the Output Queue when all selected pending operations are completed which in turn causes the MAV (Message Available) summary-message to be generated. See 10.19 and 12.5.3.
*CLS	Clears all Event Registers summarized in the status byte. All Queues, except the Output Queue, which are summarized in the status byte are emptied. The **device** is forced into the Operation Complete Command Idle State and the Operation Complete Query Idle State. (See 10.3, 11.2.4, 12.5.2 and 12.5.3).
*ESR?	Returns an NR1, which is the value of the Standard Event Status Register. See 10.12 and 11.5.1.
*ESE NRf	Sets the bits of the Standard Event Status Enable Register. See 10.10 and 11.4.2.3.3.
*ESE?	Returns an NR1, which is the value of the Standard Event Status Enable Register. See 10.11 and 11.4.2.3.2.
*IST?	Returns an NR1, which is the value of the IEC 625-1 ist (individual status) local message. See 10.15 and 11.6.2.
*SRE NRf	Sets the bits of the Service Request Enable Register. See 10.34 and 11.3.2.3.
*SRE?	Returns an NR1, which is the value of the Service Request Enable Register. See 10.35 and 11.3.2.2.
*PRE NRf	Sets the bits of the Parallel Poll Enable Register. See 10.23 and 11.6.1.3.
*PRE?	Returns an NR1, which is the value of the Parallel Poll Enable Register. See 10.24 and 11.6.1.2.
*PSC NRf	Sets the power-on-status-clear flag. When set FALSE (value 0), power-on service requests are possible. See 10.25, 11.1.3, and 5.12.
*PSC?	Returns an NR1, which is the value of the state of the power-on-status-clear flag. See 10.26, 11.1.3, and 5.12.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.1.3 *Related IEC 625-1 defined operations*

Table 11-2 lists the IEC 625-1 defined operations related to this clause. A short description and clause references for further information are included with each operation.

Tableau 11-2 – Indications d'état d'opérations définies dans la CEI 625-1

Mise sous tension	Efface le registre de validation de demande de service, le registre d'état d'événement standard, et le registre de validation d'interrogation parallèle si l'indicateur d'effacement d'état à la mise sous tension est VRAI ou si l'instruction *PSC (voir 10.25) n'est pas implémentée. Voir 5.12 pour les exigences supplémentaires à la mise sous tension.
Interrogation série	Il s'agit de l'opération d'interrogation série CEI 625-1. La valeur des messages STB et RQS définis CEI 625-1 est renvoyée. La valeur de l'octet d'état n'est pas affectée comme résultat de l'interrogation série. 'rsv', cependant, est mis FAUX par l'**appareil**, entraînant l'effacement du message RQS lorsque l'**appareil** quitte APRS (Affirmative Poll Response State) (= état de réponse affirmative à une interrogation).
Interrogation parallèle	Il s'agit de l'opération d'interrogation parallèle CEI 625-1. Un message d'état d'un bit est renvoyé (voir 11.6).

11.2 *Registre d'octets d'état*

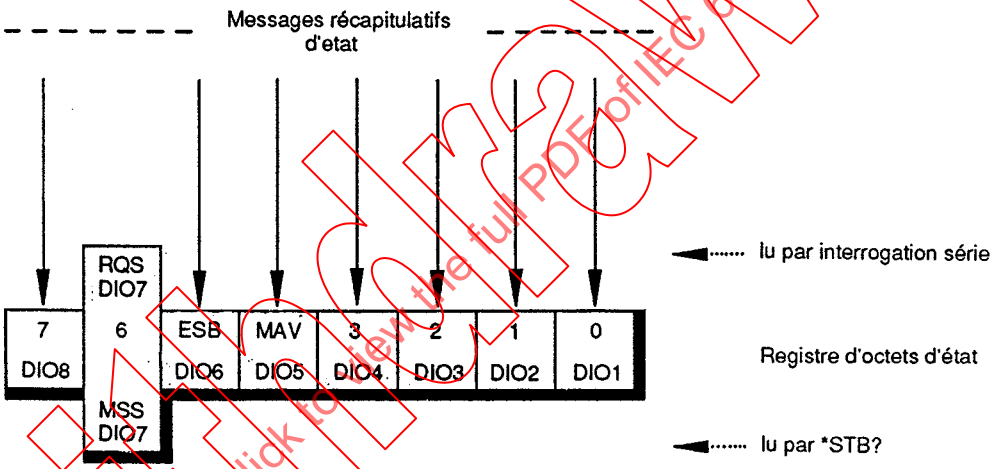


Figure 11-2 – Registre d'octets d'état

11.2.1 *Définition*

Le registre d'octets d'état contient les messages STB et RQS (ou MSS) de l'appareil. CEI 625-1 définit la méthode d'indication des messages STB et RQS, mais laisse indéfinis les protocoles de positionnement et d'effacement et la sémantique pour le message STB. Cette norme définit en outre les messages récapitulatifs STB spécifiques à l'**appareil**. Un message MSS (état récapitulatif principal) est aussi fourni et est sorti comme bit 6 avec STB en réponse à une interrogation commune *STB?. Voir 11.2.2.2.

11.2.1.1 Message récapitulatif (ESB) du bit d'état d'événement standard défini dans la CEI 625-2

Le message récapitulatif ESB est un message défini dans la CEI 625-2 et apparaît dans le bit 5 du registre d'octets d'état. Son état indique si l'un ou plusieurs des événements définis dans la CEI 625-2 validés se sont produits ou non depuis la dernière lecture ou le dernier effacement du registre d'état d'événements standards (voir 11.5.1).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Table 11-2 – Status reporting IEC 625-1 defined operations

Power-on	Clears the Service Request Enable Register, the Standard Event Status Register, and the Parallel Poll Enable Register if the power-on-status-clear flag is TRUE or if the *PSC command (see 10.25) is not implemented. See 5.12 for additional power-on requirements.
Serial poll	This is the IEC 625-1 serial poll operation. The value of the IEC 625-1 defined STB and RQS messages is returned. The value of the status byte is not altered as a result of the serial poll. "rsv", however, is set FALSE by the **device** causing the RQS message to be cleared when the **device** leaves APRS (Affirmative Poll Response State).
Parallel poll	This is the IEC 625-1 parallel poll operation. A one-bit status message is returned (see 11.6).

11.2 *Status Byte Register*

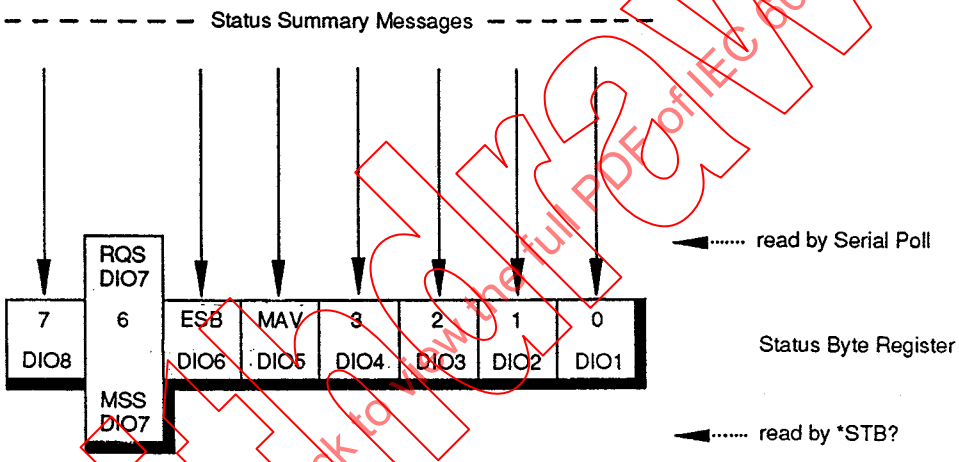


Figure 11-2 – Status Byte Register

11.2.1 *Definition*

The Status Byte Register contains the device's STB and RQS (or MSS) messages. IEC 625-1 defines the method of reporting the STB and RQS message, but leaves the setting and clearing protocols and semantics for the STB message undefined. This standard further defines specific **device** STB summary-messages. A MSS (Master Summary Status) message is also provided which is output as bit 6 with the STB in response to a *STB? common query. See 11.2.2.2.

11.2.1.1 IEC 625-2 defined Standard Event Status Bit (ESB) summary-message

The ESB summary-message is an IEC 625-2 defined message which appears in bit 5 of the Status Byte Register. Its state indicates whether or not one or more of the enabled IEC 625-2 defined events have occurred since the last reading or clearing of the Standard Event Status Register (see 11.5.1).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Le message récapitulatif ESB est VRAI quand un événement validé dans le registre d'état d'événements standard est mis VRAI. Réciproquement, le message récapitulatif ESB est FAUX lorsque des événements invalidés sont VRAIS. Voir 11.5.1.2 et 11.5.1.3 pour l'opération détaillée.

11.2.1.2 Message récapitulatif de file d'attente MAV défini par la norme

Le message récapitulatif MAV (message disponible) est un message défini CEI 625-2 qui apparaît dans le bit 4 du registre d'octets d'état. L'état du message indique si la file d'attente de sortie (voir 6.1.10) est vide ou non. Chaque fois que l'**appareil** est prêt à accepter une demande du **contrôleur** pour sortir des octets de données, le message récapitulatif MAV devra être VRAI. Le message récapitulatif MAV devra être FAUX lorsque la file d'attente de sortie est vide.

Ce message est utilisé pour synchroniser l'échange d'informations avec le **contrôleur**. Le **contrôleur** peut, par exemple, envoyer une instruction d'interrogation à l'**appareil** et attendre alors que MAV devienne VRAI. Le bus CEI 625-1 est disponible pour une autre utilisation pendant qu'un programme d'application attend la réponse d'un **appareil**. Si un programme d'application commence une opération de lecture de la file d'attente de sortie sans premièrement tester MAV, toute l'activité du **bus du système** est arrêtée jusqu'à ce que l'**appareil** réponde. Voir l'article 12 et l'annexe B pour les discussions et exemples de synchronisation des programmes d'application et de l'**appareil**.

11.2.1.3 Messages récapitulatifs définis par l'appareil

Les bits 0 à 3 et 7 sont disponibles pour une utilisation comme messages récapitulatifs par le concepteur de l'appareil. Chaque message récapitulatif défini par l'appareil devra avoir une structure de données d'état associée qui suive un des deux modèles décrits à 11.4. Cette structure est soit un ensemble de registres pour événement parallèle et indication de condition, soit une file d'attente d'indication d'état séquentiel et d'information. Le message récapitulatif résume l'état en cours de la structure de données d'état associée. C'est-à-dire qu'un ou plusieurs bits d'événements validés et VRAI dans le modèle de registre ou dans une file d'attente non vide dans le modèle de file d'attente sont indiqués par un message récapitulatif VRAI. Voir 11.4.2 et 11.4.3.

11.2.2 *Lecture du registre d'octets d'état*

Le registre d'octets d'état peut être lu avec soit une interrogation série, soit l'interrogation commune *STB?. Ces deux méthodes lisent le message STB CEI 625-1. La valeur envoyée pour la position du bit 6 est, cependant, dépendante de la méthode utilisée.

11.2.2.1 Lecture avec une interrogation série

Lorsqu'il sera interrogé en série, selon la CEI 625-1, l'**appareil** devra renvoyer l'octet d'état de 7 bits plus le message RQS CEI 625-1 de bit simple. Le message RQS est ajouté par le contrôle E/S (voir 6.1.4) en accord avec le diagramme d'état CEI 625-1 SR. L'octet d'état et le message RQS sont renvoyés au **contrôleur** comme un octet de données simple. Selon la CEI 625-1, le message RQS indique si l'**appareil** envoyait un SRQ VRAI. Les **appareils** ne devront pas envoyer le message END pendant une interrogation série.

Pour l'application de cette norme, les lignes de signaux DIO CEI 625-1 DIO1-6 et 8 correspondent respectivement aux bits 0-5 et 7 du registre d'octets d'état. Le message RQS est envoyé sur la ligne DIO7 (bit 6) selon les exigences de la CEI 625-1.

IECNORM.COM : Click to view the full PDF of 60625-2:1993
Withdrawn

The ESB summary-message is TRUE when an enabled event in the Standard Event Status Register is set TRUE. Conversely, the ESB summary-message is FALSE when no enabled events are TRUE. See 11.5.1.2 and 11.5.1.3 for detailed operation.

11.2.1.2 Standard-defined MAV Queue summary-message

The MAV (Message Available) summary-message is an IEC 625-2 defined message which appears in bit 4 of the Status Byte Register. The state of the message indicates whether or not the Output Queue (see 6.1.10) is empty. Whenever the **device** is ready to accept a request by the **controller** to output data bytes, the MAV summary-message shall be TRUE. The MAV summary-message shall be FALSE when the Output Queue is empty.

This message is used to synchronize information exchange with the **controller**. The **controller** can, for example, send a query command to the **device** and then wait for MAV to become TRUE. The IEC 625-1 bus is available for other use while an application program is waiting for a **device** to respond. If an application program begins a read operation of the Output Queue without first checking for MAV, all **system bus** activity is held up until the **device** responds. See clause 12 and annex B for device and application program synchronization discussion and examples.

11.2.1.3 Device-defined summary-messages

Bits 0 through 3 and bit 7 are available for use by the device designer as summary-messages. Every device-defined summary-message shall have an associated Status Data Structure that follows one of the two models described in 11.4. This structure is either a register set for parallel event and condition reporting, or a Queue for sequential status and information reporting. The summary-message summarizes the current status of the associated Status Data Structure. That is, one or more TRUE and enabled event bits in the register-model or a non-empty Queue in the queue model is indicated by a TRUE summary-message. See 11.4.2 and 11.4.3.

11.2.2 *Reading the Status Byte Register*

The Status Byte Register can be read with either a serial poll or the *STB? common query. Both of these methods read the IEC 625-1 STB message. The value sent for the bit 6 position is, however, dependent upon the method used.

11.2.2.1 Reading with a serial poll

When serial polled according to IEC 625-1, the **device** shall return the 7-bit status byte plus the single bit IEC 625-1 RQS message. The RQS message is added by the I/O Control (see 6.1.4) in accordance with the IEC 625-1 SR state diagram. The status byte and RQS message are returned to the **controller** as a single data byte. Per IEC 625-1, the RQS message indicates if the **device** was sending SRQ TRUE. **Devices** shall not send the END message during a serial poll.

For the purpose of this standard, the IEC 625-1 DIO signal lines DIO1-6 and 8 correspond to bits 0-5 and 7 of the status byte register, respectively. The RQS message is sent on line DIO7 (bit 6) according to the requirements of IEC 625-1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Un message VRAI signifie que la ligne de signaux DIO associée devra être excitée (mise à une tension basse) lorsque l'octet d'état est envoyé. Cette convention est la logique négative de la CEI 625-1. (CEI 625-1, article 18).

La partie de message STB du registre d'octets d'état est lue de façon non destructive. La valeur de l'octet d'état ne devra pas être altérée par une interrogation série.

L'appareil doit mettre le message rsv FAUX lorsqu'il est interrogé (voir 11.3.3) de sorte que le message RQS sera FAUX si l'**appareil** est à nouveau interrogé avant qu'une nouvelle raison de service (voir 11.3.3) ne soit apparue.

11.2.2.2 Lecture avec l'interrogation *STB?

L'interrogation commune *STB? (voir 10.36) entraînera l'envoi par l'**appareil** du contenu du registre d'octets d'état et du message récapitulatif MSS (état récapitulatif principal) (voir 11.2.2.3) comme un élément de <DONNÉES DE PROGRAMMATION NUMÉRIQUES NR1> simple.

La réponse devra représenter la somme des valeurs par poids binaires des bits 0-5 et 7 (respectivement de poids 1, 2, 4, 8, 16, 32, et 128) du registre d'octets d'état et du message récapitulatif MSS (de poids 64). De cette façon, la réponse à *STB?, lorsqu'elle est considérée comme une valeur binaire, est identique à la réponse à une interrogation série sauf que le message récapitulatif MSS apparaît dans le bit 6 à la place du message RQS.

L'interrogation commune *STB? ne devra en aucune façon modifier directement l'octet d'état, le message récapitulatif MSS, le message RQS ou le message local rsv.

NOTE – Les messages MAV et RQS peuvent être indirectement affectés, voir 11.2.1.2 et 11.3.3 respectivement.

11.2.2.3 Récapitulatif d'état principal

Le message MSS (récapitulatif d'état principal) indique que l'appareil a au moins une raison pour demander un service. Bien que le message MSS soit envoyé dans la position du bit 6 de la réponse de l'appareil à l'interrogation *STB?, il n'est pas envoyé en réponse à une interrogation série et ne doit pas être considéré comme une partie de l'octet d'état CEI 625-1.

MSS est le OU inclusif de la combinaison bit à bit (à l'exclusion du bit 6) du registre d'octets d'état (SB) et du registre de validation de demande de service (SRE) (voir 11.3.2), c'est-à-dire:

MSS est défini par

(bit 0 du registre SB ET bit 0 du registre SRE)
OU
(bit 1du registre SB ET bit 1 du registre SRE)
OU
(bit 2 du registre SB ET bit 2 du registre SRE)
OU
(bit 3 du registre SB ET bit 3 du registre SRE)
OU

IECNORM.COM : Click to view the full PDF of 60625-2:1993

A TRUE message means that the associated DIO signal line shall be asserted (pulled to a low voltage) when the status byte is sent. This convention is standard IEC 625-1 negative logic (IEC 625-1, clause 18).

The STB message portion of the Status Byte Register is read non-destructively. The value of the status byte shall not be altered by a serial poll.

The **device** shall set the rsv message FALSE when polled (see 11.3.3) so that the RQS message will be FALSE if the device is polled again before a new reason for service (see 11.3.3) has occurred.

11.2.2.2 Reading with the *STB? query

The *STB? common query (see 10.36) shall cause the **device** to send the contents of the Status Byte Register and the MSS (Master Summary Status) summary message (see 11.2.2.3) as a single <NR1 NUMERIC RESPONSE DATA> element.

The response shall represent the sum of the binary-weighted values of the Status Byte Register bits 0-5 and 7 (weights 1, 2, 4, 8, 16, 32, and 128 respectively) and the MSS summary-message (weight 64). Thus, the response to *STB?, when considered as a binary value, is identical to the response to a serial poll except that the MSS summary-message appears in bit 6 in place of the RQS message.

The *STB? common query shall not directly alter the status byte, the MSS summary message, the RQS message, or the rsv local message.

NOTE - The MAV and RQS messages may be indirectly affected, see 11.2.1.2 and 11.3.3 respectively.

11.2.2.3 Master Summary Status

The MSS (Master Summary Status) message indicates that the device has at least one reason for requesting service. Although the MSS message is sent in bit position 6 of the device's response to the *STB? query, it is not sent in response to a serial poll and should not be considered part of the IEC 625-1 status byte.

MSS is the inclusive OR of the bitwise combination (excluding bit 6) of the Status Byte (SB) Register and the Service Request Enable (SRE) Register (see 11.3.2); that is:

MSS is defined as

(SB Register bit 0 AND SRE Register bit 0)
OR
(SB Register bit 1 AND SRE Register bit 1)
OR
(SB Register bit 2 AND SRE Register bit 2)
OR
(SB Register bit 3 AND SRE Register bit 3)
OR

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

(bit 4 du registre SB ET bit 4 du registre SRE)
OU
(bit 5 du registre SB ET bit 5 du registre SRE)
OU
(bit 7 du registre SB ET bit 7 du registre SRE)

Noter que la définition de MSS ignore l'état du bit 6 aussi bien dans le registre d'octets d'état que dans le registre de validation de demande de service. Dans le but de calculer la valeur de MSS, les implémenteurs peuvent choisir de traiter l'octet d'état comme une valeur de 8 bits, le bit 6 étant toujours "0".

11.2.3 *Ecriture dans le registre d'octets d'état*

Le registre d'octets d'état change seulement lorsque l'état de la structure de données d'état de recouvrement change. Ceci est illustré à la figure 11-1.

NOTE – Des changements dans l'octet d'état peuvent affecter l'état du message local rsv, du message récapitulatif MSS, et du message d'interface SRQ, comme cela est décrit à 11.3.3.

11.2.4 *Effacement du registre d'octets d'état*

L'instruction commune *CLS (voir 10.3) entraînera l'effacement de toutes les structures de données d'état (c'est-à-dire leurs registres d'événements et leurs files d'attente), de sorte que les messages récapitulatifs correspondants soient clairs. La file d'attente de sortie et son message récapitulatif MAV sont une exception et ne sont pas affectés par *CLS.

L'octet d'état tout entier peut être effacé en envoyant à l'**appareil** l'instruction *CLS après un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> et avant tout élément d'<UNITÉ DE MESSAGE D'INTERROGATION>. On effacera de la file d'attente de sortie tous les messages non lus en utilisant cette méthode (voir 6.3.2.3). Avec la file d'attente de sortie vide, le message récapitulatif MAV sera FAUX. Le message MSS dans la réponse *STB? deviendra également FAUX. Le message RQS dans le registre d'octets d'état sera FAUX à moins que l'**appareil** ne soit en APRS (état de réponse à une interrogation affirmatif).

11.3 *Validation de demande de service*

11.3.1 *Opération*

L'opération de validation de demande de service est représentée à la figure 11-3.

La validation de demande de service permet à un programmeur d'application de sélectionner quel message récapitulatif, dans le registre d'octet d'état, peut causer des demandes de service. Le registre de validation de demande de service, illustré à la figure 11-3, est utilisé pour sélectionner les messages récapitulatifs.

11.3.2 *Registre de validation de demande de service*

11.3.2.1 Fonction

Le registre de validation de demande de service est un registre de 8 bits qui valide les messages récapitulatifs correspondants dans le registre d'octet d'état. Ainsi, le programmeur d'application peut sélectionner les raisons, pour un **appareil**, d'émettre une demande de service en modifiant le contenu du registre de validation de demande de service.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

(SB Register bit 4 AND SRE Register bit 4)
OR
(SB Register bit 5 AND SRE Register bit 5)
OR
(SB Register bit 7 AND SRE Register bit 7)

Note that the MSS definition ignores the state of bit 6 in both the Status Byte Register and the Service Request Enable Register. For purposes of computing the value of MSS, implementors may choose to treat the status byte as an 8-bit value with bit 6 always "0".

11.2.3 *Writing the Status Byte Register*

The Status Byte Register is altered only when the state of the overlaying Status Data Structure is altered. This is illustrated in figure 11-1.

NOTE - Changes in the status byte may affect the state of the rsv local message, MSS summary message, and the SRQ interface message as described in 11.3.3.

11.2.4 *Clearing the Status Byte Register*

The *CLS common command (see 10.3) shall cause all Status Data Structures (that is, their Event Registers and Queues) to be cleared so that the corresponding summary messages are clear. The Output Queue and its MAV summary message are an exception and are unaffected by *CLS.

The entire status byte can be cleared by sending the *CLS command to the **device** after a <PROGRAM MESSAGE TERMINATOR> and before any <QUERY MESSAGE UNIT> elements. The Output Queue will be cleared of any unread messages using this method (see 6.3.2.3). With the Output Queue empty, the MAV summary message will be FALSE. The MSS message in the *STB? response will also go FALSE. The RQS message in the Status Byte Register will be FALSE unless the **device** is in APRS (Affirmative Poll Response State).

11.3 *Service Request enabling*

11.3.1 *Operation*

Service Request enabling operation is shown in figure 11-3.

Service Request enabling allows an application programmer to select which summary messages in the Status Byte Register may cause service requests. The Service Request Enable Register, illustrated in figure 11-3, is used to select the summary messages.

11.3.2 *Service Request Enable Register*

11.3.2.1 Function

The Service Request Enable Register is an 8-bit register that enables corresponding summary messages in the Status Byte Register. Thus, the application programmer can select reasons for a **device** to issue a service request by altering the contents of the Service Request Enable Register.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.3.2.2 Lecture du registre de validation de demande de service

On lit le registre de validation de demande de service avec l'interrogation commune *SRE? (voir 10.35). Le message de réponse à cette interrogation devra être un élément de <DONNÉES DE RÉPONSE NUMÉRIQUES NR1> qui représente la somme des valeurs de poids binaire du registre de validation de demande de service (2 élevé à la puissance du nombre de bits). La valeur du bit inutilisé 6 devra toujours être zéro.

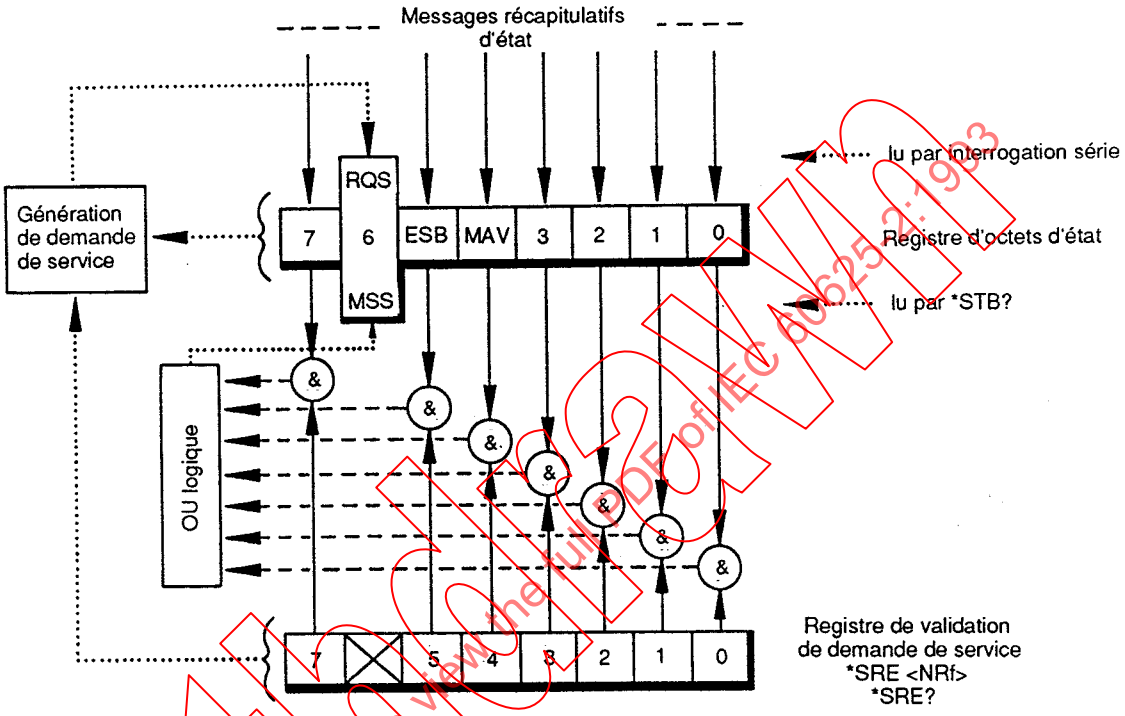


Figure 11-3 – Validation de demande de service

11.3.2.3 Ecriture dans le registre de validation de demande de service

On écrit dans le registre de validation de demande de service avec l'instruction commune *SRE (voir 10.34). *SRE est suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>.

La <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>, arrondie à une valeur entière et exprimée en base 2 (binaire), devra représenter les valeurs des bits du registre de validation de demande de service. Une valeur de bit 1 devra indiquer une condition validée. Une valeur de bit 0 indiquera une condition invalidée.

L'**appareil** devra toujours ignorer la valeur du bit 6.

11.3.2.4 Effacement du registre de validation de demande de service

L'envoi de l'instruction commune *SRE avec un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> de valeur zéro efface le registre de validation de demande de service. Un registre effacé ne permet à aucune information d'état de générer un message local rsv et ainsi, aucune demande de service n'est émise.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.3.2.2 Reading the Service Request Enable Register

The Service Request Enable Register is read with the *SRE? common query (see 10.35). The response message to this query shall be an <NR1 NUMERIC RESPONSE DATA> element that represents the sum of the binary-weighted values of the Service Request Enable Register (2 raised to the power of the bit number). The value of unused bit 6 shall always be zero.

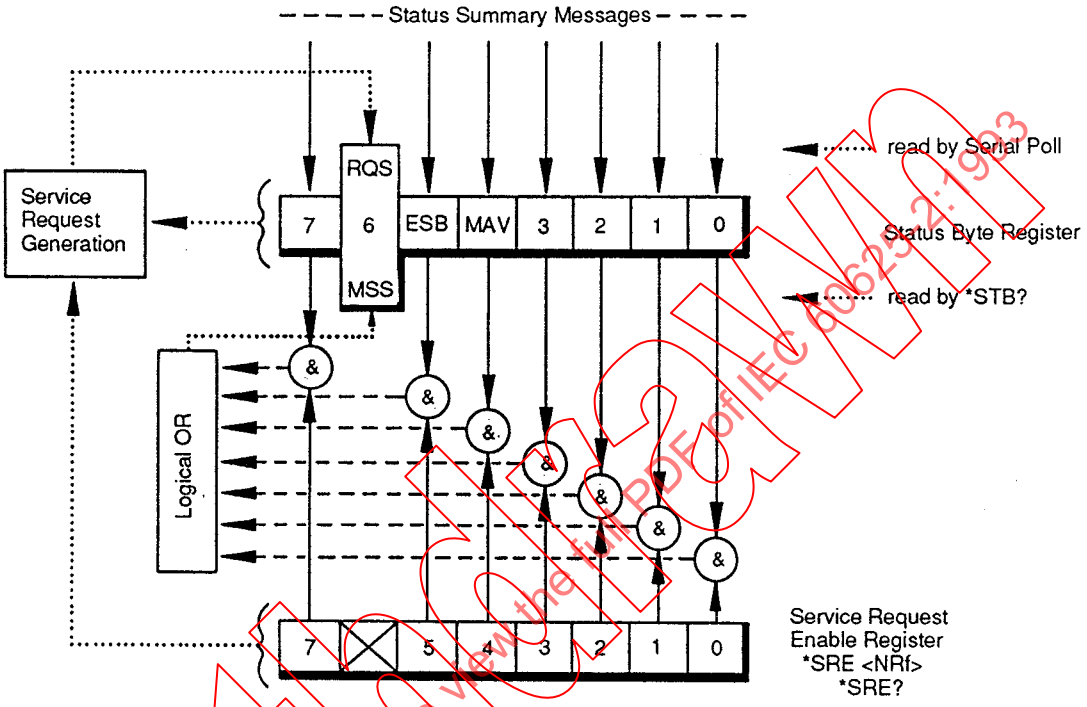


Figure 11-3 – Service Request enabling

11.3.2.3 Writing the Service Request Enable Register

The Service Request Enable Register is written with the *SRE common command (see 10.34). The *SRE is followed by a <DECIMAL NUMERIC PROGRAM DATA> element.

The <DECIMAL NUMERIC PROGRAM DATA>, when rounded to an integer value and expressed in base 2 (binary), shall represent the bit values of the Service Request Enable Register. A bit value of one shall indicate an enabled condition. A bit value of zero shall indicate a disabled condition.

The **device** shall always ignore the value of bit 6.

11.3.2.4 Clearing the Service Request Enable Register

Sending the *SRE common command with a <DECIMAL NUMERIC PROGRAM DATA> element value of zero clears the Service Request Enable Register. A cleared register does not allow status information to generate an rsv local message and thus, no service requests are issued.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Si l'indicateur d'effacement d'état à la mise sous tension est VRAI ou si *PSC n'est pas implémenté (voir 11.1.3), le registre de validation de demande de service devra être effacé à la mise sous tension. L'instruction *PSC est utilisée pour sélectionner l'état de l'indicateur d'effacement d'état à la mise sous tension (voir 10.25). 5.12 contient les exigences supplémentaires à la mise sous tension.

Le registre de validation de demande de service ne devra pas être changé par la réception du message **dcas**, ni par n'importe quelle condition de l'**appareil** autre que celles spécifiées dans cet article.

11.3.3 *Génération de demande de service*

Tous les **appareils** devront implémenter la fonction de demande de service comme cela est décrit dans la CEI 625-1. Cette fonction fournit un **appareil** avec les capacités de demander des services au **contrôleur** via le message d'interface SRQ (demande de service) et d'indiquer qu'il a demandé un service via le message ROS (demande de service), qui est envoyé avec l'octet d'état en réponse à une interrogation série.

La génération de demandes de services est contrôlée par le message local rsv (demande de service) CEI 625-1 qui est représenté sous forme de schéma fonctionnel à la figure 11-4. Cet article décrit le couplage entre l'octet d'état et le message rsv (demande de service) CEI 625-1. Cet article donne aussi les exigences supplémentaires (au-delà de celles spécifiées dans la CEI 625-1) sur l'interaction de l'établissement d'une liaison avec l'accepteur CEI 625-1 et le message rsv. Ces exigences assurent qu'un **appareil** doit:

1) Exciter SRQ quand une condition préalablement "validée" se produira.

2) Maintenir SRQ excité jusqu'à ce que le contrôleur ait reconnu la demande de service et interrogé l'appareil ou ait entrepris une action spécifique pour annuler la demande (par exemple, instruction *CLS).

3) Libérer SRQ lorsqu'il est interrogé de façon que le **contrôleur** puisse détecter un SRQ d'un autre appareil.

4) Exciter à nouveau SRQ si une autre condition se produit, que le **contrôleur** ait ou n'ait pas effacé la première condition.

11.3.3.1 Nouvelle raison de service

Chaque fois que les contenus du registre de validation d'octets d'état ou de demande de service sont changés, l'**appareil** doit déterminer si le changement affecte l'état de demande de service de l'**appareil**. Les transitions d'état de l'appareil n'affectent pas directement l'état du message d'interface SRQ. Au lieu de cela, les changements de l'octet d'état et du registre de validation de demande de service génèrent les messages locaux **reqt** (demande de rsv VRAI) et **reqf** (demande de rsv FAUX), qui sont des entrées dans le bloc de synchronisation de demande de service à la figure 11-4. La sortie du bloc de synchronisation de demande de service est le message local rsv qui contrôle le message d'interface SRQ, comme cela est décrit dans la CEI 625-1.

L'**appareil** devra générer une nouvelle demande de service (mettre le message **reqt** VRAI) quand:

1) Un bit d'octet d'état passe de FAUX à VRAI alors que le bit correspondant dans le registre de validation de demande de service est VRAI.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

If the power-on-status-clear flag is TRUE or if *PSC is not implemented (see 11.1.3), the Service Request Enable Register shall be cleared upon power-on. The *PSC command is used to set the state of the power-on-status-clear flag (see 10.25). 5.12 contains additional power-on requirements.

The Service Request Enable Register shall not be changed by the receipt of the **dcas** message, nor by any **device** condition other than as stated in this section.

11.3.3 *Service Request generation*

All **devices** shall implement the Service Request function as described in IEC 625-1. This function provides a **device** with the capabilities of requesting service from the **controller** via the SRQ (Service Request) interface message and reporting that it has requested service via the RQS (Request Service) message, which is sent with the status byte in response to a serial poll.

The generation of service requests is controlled by the IEC 625-1 rsv (request service) local message which is shown in block diagram form in figure 11-4. This clause describes the coupling between the status byte and the IEC 625-1 rsv (request service) message. This clause also places additional requirements (beyond those specified in IEC 625-1) on the interaction of the IEC 625-1 Acceptor Handshake and the rsv message. These requirements ensure that a **device** shall:

1) Assert SRQ when a previously "enabled" condition occurs.

2) Keep SRQ asserted until the **controller** has recognized the service request and polled the **device** or has taken specific action to cancel the request (e.g. *CLS command).

3) Release SRQ when polled so that the **controller** can detect an SRQ from another device.

4) Assert SRQ again if another condition occurs, whether or not the **controller** has cleared the first condition.

11.3.3.1 New reason for service

Whenever the contents of the status byte or Service Request Enable Register are changed, the **device** must determine whether the change affects the service request state of the **device**. Device status transitions do not affect the state of the SRQ interface message directly. Instead, changes to the status byte and the Service Request Enable Register generate the local messages **reqt** (Request rsv TRUE) and **reqf** (Request rsv FALSE), which are inputs to the Service Request Synchronization block in figure 11-4. The output of the Service Request Synchronization block is the local rsv message which controls the SRQ interface message as described in IEC 625-1.

The **device** shall generate a new service request (set the **reqt** message TRUE) when:

1) A bit in the status byte changes from FALSE to TRUE while the corresponding bit in the Service Request Enable Register is TRUE.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

2) Un bit du registre de validation de demande de service passe de FAUX à VRAI alors que le bit correspondant de l'octet d'état est VRAI.

3) Un bit de l'octet d'état passe de FAUX à VRAI et le bit correspondant dans le registre de validation de demande de service passe simultanément de FAUX à VRAI.

L'**appareil** doit mettre le message **reqt** FAUX quand:

1) Le message MSS passe de VRAI à FAUX.

2) L'**appareil** entre dans l'état SRWS du diagramme d'état rsv d'ensemble, figure 11-5.

L'**appareil** devra cesser de demander des services (mettre le message **reqf** VRAI) lorsque le message MSS passe de VRAI à FAUX.

L'**appareil** devra mettre le message **reqf** FAUX lorsqu'il entre dans l'état SRIS (état inactif de demande de service) du diagramme d'état rsv d'ensemble à la figure 11-5 ou à la mise sous tension.

L'**appareil** peut aussi mettre **reqt** VRAI chaque fois que les contenus de l'octet d'état sont changés et que MSS est VRAI. Du fait que des demandes de services intempestives peuvent être générées, cette méthode n'est pas recommandée. Elle est autorisée, toutefois, afin de recevoir certains circuits d'interface CEI 625-1 commercialement disponibles. Voir 11.3.3.4 sur les techniques d'implémentation.

En général, le programme d'application du **contrôleur** ne doit jamais supposer que SRQ indique qu'une nouvelle raison de service s'est produite, mais seulement qu'une nouvelle raison de service peut s'être produite et que le programme d'application devrait vérifier l'octet d'état de l'**appareil** pour déterminer si c'est vraiment le cas.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

2) A bit in the Service Request Enable Register changes from FALSE to TRUE while the corresponding bit in the status byte is TRUE.

3) A bit in the status byte changes from FALSE to TRUE and the corresponding bit in the Service Request Enable Register changes from FALSE to TRUE simultaneously.

The **device** shall set the **reqt** message FALSE when:

1) The MSS message changes from TRUE to FALSE.

2) The **device** enters the SRWS of the Set rsv State Diagram in figure 11-5.

The **device** shall stop requesting service (set the **reqf** message TRUE) when the MSS message changes from TRUE to FALSE.

The **device** shall set the **reqf** message FALSE when it enters SRIS (Service Request Idle State) of the Set rsv State Diagram in figure 11-5 or at power-on.

The **device** may also set **reqt** TRUE whenever the contents of the status byte are changed and MSS is TRUE. Because extraneous service requests may be generated, this method is not recommended. It is permitted, however, to accommodate certain commercially available IEC 625-1 interface circuits. See 11.3.3.4 on implementation techniques.

In general, the **controller** application program must never assume that SRQ indicates that a new reason for service has occurred, but only that a new reason for service may have occurred, and that the application program should check the **device** status byte to determine whether this is indeed the case.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

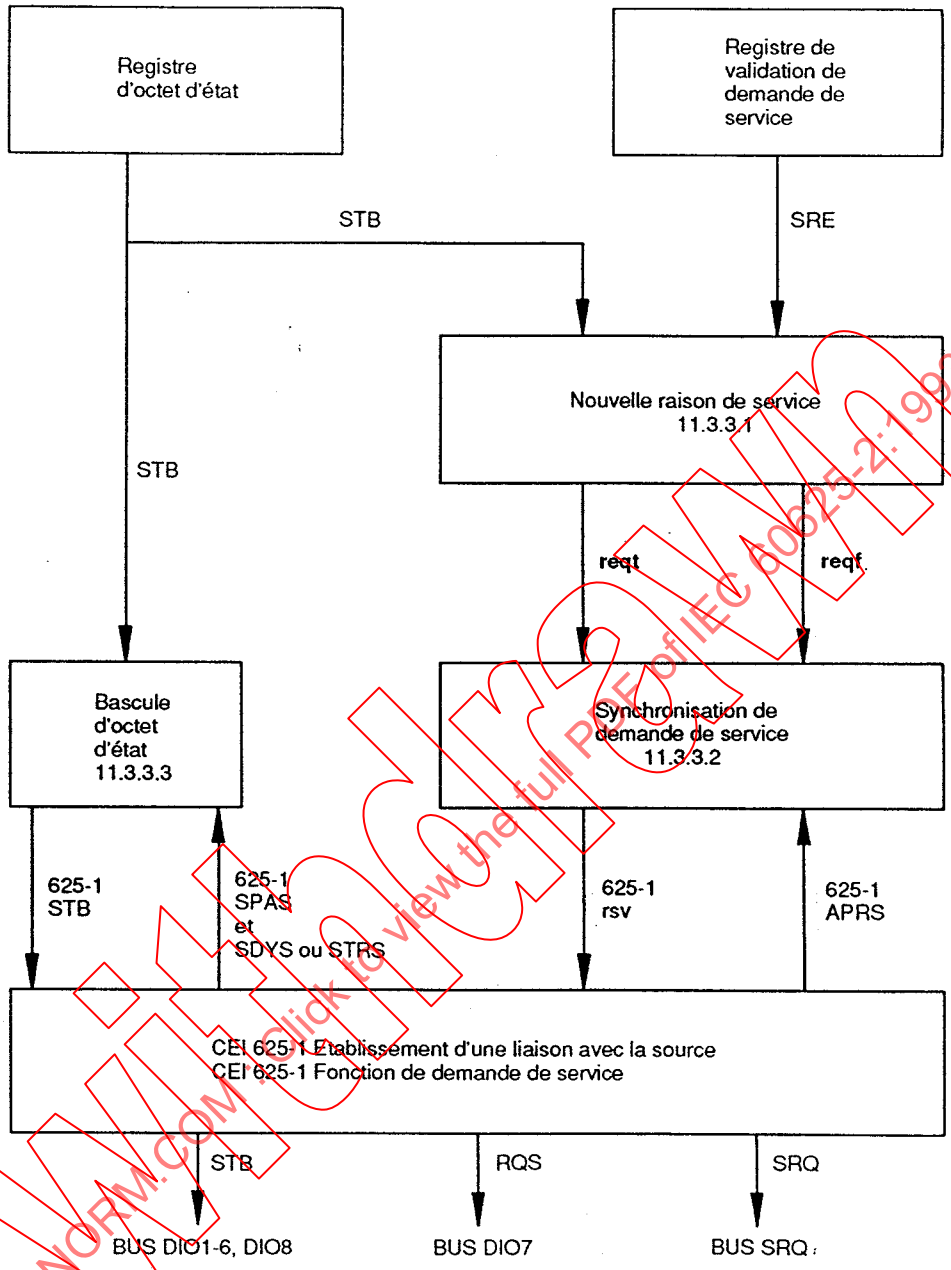


Figure 11-4 – Génération de demande de service

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

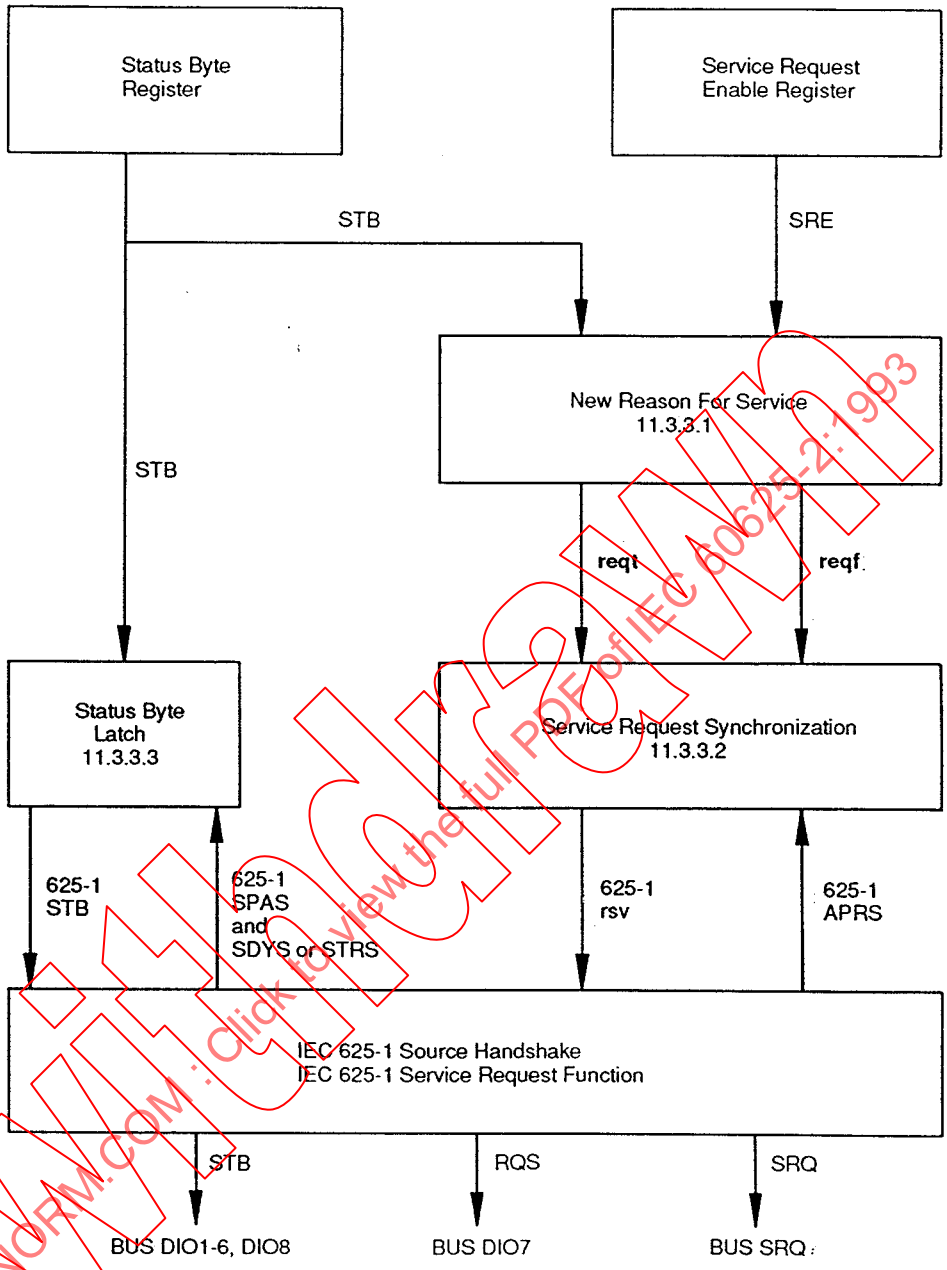


Figure 11-4 – Service Request generation

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

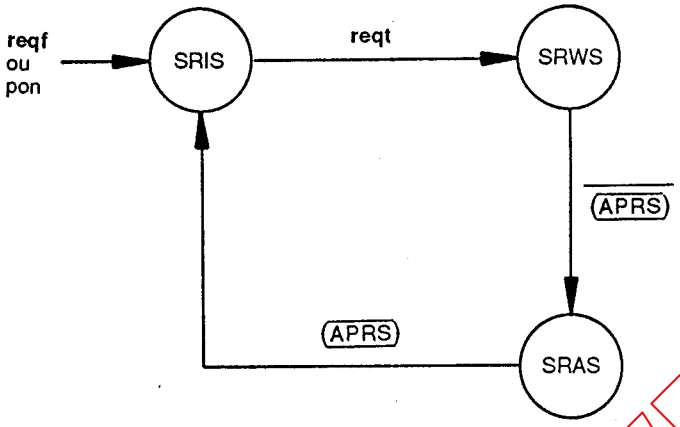


Figure 11-5 – Schéma d'état rsv d'ensemble

Tableau 11-3 – Schéma d'état rsv d'ensemble – Mnémoniques

Messages		Etats d'interface	
reqt	Demande rsv VRAI	SRIS	Etat inactif de demande de service
reqf	Demande rsv FAUX	SRWS	Etat d'attente de demande de service
rsv	Demande de service	SRAS	Etat actif de demande de service
		(APRS)	Etat affirmatif de réponse à une interrogation

Tableau 11-4 – Schéma d'état rsv d'ensemble – Sortie des messages

Etat rsv d'ensemble	Message rsv envoyé	Interaction de l'**appareil**
SRIS	F	ne demande pas de service
SRWS	F	ne demande pas de service
SRAS	T	demande de service

11.3.3.2 Synchronisation de demande de service

Le diagramme d'état rsv d'ensemble, représenté à la figure 11-5, synchronise les changements du registre de validation de demande de service et de l'octet d'état de l'appareil avec l'état de la fonction SR CEI 625-1. Cela assure que l'**appareil** ne change pas les messages d'interface SRQ ou RQS alors qu'il est interrogé par **contrôleur**, mais que l'**appareil** "se rappellera" que le changement d'état s'est produit et changera l'état de SRQ et RQS à la sortie de APRS (Etat de réponse à une interrogation affirmatif).

A la mise sous tension, ou lorsque le message local **reqf** est VRAI, l'**appareil** devra entrer dans l'état SRIS (état inactif de demande de service). Pendant cet état, l'**appareil** devra envoyer le message rsv local FAUX.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

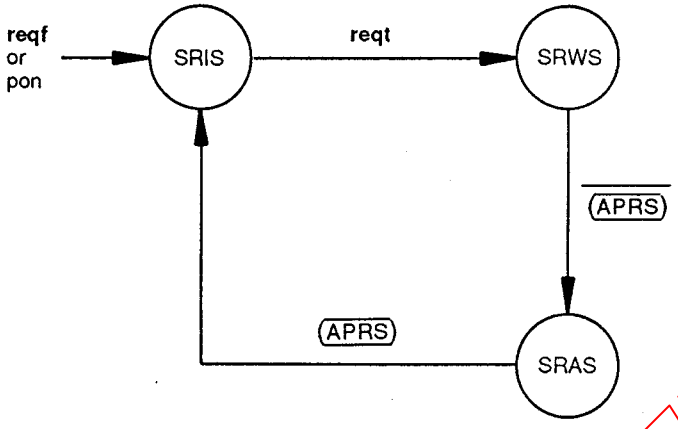


Figure 11-5 – Set rsv State Diagram

Table 11-3 – Set rsv State Diagram mnemonics

Messages		Interface states	
reqt	Request rsv TRUE	SRIS	Service Request Idle State
reqf	Request rsv FALSE	SRWS	Service Request Wait State
rsv	request service	SRAS	Service Request Active State
		(APRS)	Affirmative Poll Response State

Table 11-4 – Set rsv State Diagram message output

Set rsv state	rsv message sent	**Device** interaction
SRIS	F	not requesting service
SRWS	F	not requesting service
SRAS	T	requesting service

11.3.3.2 Service Request synchronization

The Set rsv State Diagram, shown in figure 11-5, synchronizes changes to the device's status byte and Service Request Enable Register with the state of the IEC 625-1 SR function. It ensures that the **device** does not change the SRQ or RQS interface messages while being polled by the **controller**, but that the **device** will 'remember' that the status change occurred and will change the state of SRQ and RQS on exit from APRS (Affirmative Poll Response State).

At power-on, or when the **reqf** local message is TRUE, the **device** shall enter SRIS (Service Request Idle State). While in this state, the **device** shall send the rsv local message FALSE.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Quand l'**appareil** mettra le message local **reqt** VRAI, l'**appareil** devra entrer dans l'état SRWS (état d'attente de demande de service). L'**appareil** devra rester dans l'état SRWS et continuer à envoyer le message local rsv FAUX jusqu'à ce que les messages pon ou **reqf** local deviennent VRAIS ou que l'**appareil** quitte l'APRS CEI 625-1. Le fait de rester dans l'état SRWS garantit que l'**appareil** ne changera pas les messages d'interface RQS ou SRQ de FAUX à VRAI pendant qu'il est interrogé par le **contrôleur**.

Si l'**appareil** n'est pas en APRS, ou lorsqu'il quitte APRS, il devra entrer dans l'état SRAS (état actif de demande de service). Pendant qu'il est dans cet état, l'**appareil** devra envoyer le message local rsv VRAI. L'**appareil** devra rester dans cet état jusqu'à ce que les messages pon ou **reqf** local deviennent VRAIS ou que l'**appareil** entre en APRS.

Après entrée en APRS, l'**appareil** devra retourner à l'état SRIS et mettre rsv FAUX, indiquant qu'il a été interrogé. Noter que l'**appareil** continuera à envoyer le message d'interface RQS VRAI jusqu'à ce qu'il ait quitté APRS, selon la CEI 625-1.

11.3.3.3 Bascule d'octet d'état

Si les contenus du registre d'octet d'état devaient être placés directement sur le bus lorsque l'**appareil** est interrogé, un changement dans l'octet d'état, alors que l'**appareil** est en SPAS CEI 625-1(Serial Poll Active State) (= état actif d'interrogation série) pourrait entraîner une lecture incorrecte de l'octet d'état par le **contrôleur**, due au changement d'état des lignes E/S des données du bus. Pour empêcher ce problème, les **appareils** doivent stocker les contenus de l'octet d'état de l'**appareil** et du message RQS comme cela est indiqué sur les lignes DIO pendant que l'**appareil** est en SPAS CEI 625-1 et que l'établissement d'une liaison avec la source CEI 625-1 est en SDYS (état de retard de la source) ou STRS (état de transfert de la source). La bascule garantit que l'octet d'état ne changera pas sur le bus au moment où le diagramme SR CEI 625-1 assure que le message RQS ne changera pas. Cette exigence n'implique pas que les contenus du registre d'octet d'état ne peuvent pas être changés pendant cette période, mais seulement que la valeur présentée au **contrôleur** sur la ligne DIO1-8 sera stable. A la figure 11-4, l'entrée à partir de l'établissement d'une liaison avec la source CEI 625-1/bloc de demande de service CEI 625-1 détermine si la bascule d'octet d'état est transparente ou non.

11.3.3.4 Techniques d'implémentation

De nombreux circuits intégrés CEI 625-1 commercialement disponibles implémentent le diagramme d'état rsv d'ensemble au lieu de, ou en plus de fournir un contrôle direct par le message local rsv. Ces circuits comprennent aussi une mémoire tampon d'octet d'état qui permet au circuit de contrôle CEI 625-1 de répondre à une interrogation série sans interruption de la microprogrammation de l'**appareil**. Cette section décrit comment il est possible de répondre aux exigences du modèle d'indication d'état CEI 625-2 en utilisant de tels circuits. Elle ne sanctionne pas l'utilisation des composants spécifiques ou n'interdit pas d'autres implémentations qui répondent aux exigences de cette norme, mais elle est considérée comme une discussion sur les techniques et les pièges communs de l'implémentation.

11.3.3.4.1 Implémentation préférée

L'implémentation préférée, du point de vue d'un concepteur d'appareil, serait de procéder à ce qui suit lorsque le contenu de la mémoire tampon de l'octet d'état est mis à jour:

1) mettre **reqt** VRAI si la "nouvelle raison de service" est VRAIE, ou
2) mettre **reqf** VRAI si MSS est FAUX.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

When the **device** sets the **reqt** local message TRUE, the **device** shall enter SRWS (Service Request Wait State). The **device** shall remain in SRWS and continue to send the rsv local message FALSE until the pon or **reqf** local messages become TRUE or the **device** leaves the IEC 625-1 APRS. Staying in SRWS ensures that the **device** will not change the RQS or SRQ interface messages from FALSE to TRUE while it is being polled by the **controller**.

If the **device** is not in APRS, or when it leaves APRS, it shall enter SRAS (Service Request Active State). While in this state, the **device** shall send the rsv local message TRUE. The **device** shall remain in this state until the pon or **reqf** local messages become TRUE or the **device** enters APRS.

Upon entry into APRS, the **device** shall return to SRIS and set rsv FALSE, indicating that it has been polled. Note that the **device** will continue to send the RQS interface message TRUE until it has left APRS, according to IEC 625-1.

11.3.3.3 Status Byte Latch

If the contents of the Status Byte Register were to be placed directly on to the bus when the **device** was polled, a change in the status byte while the **device** was in IEC 625-1 SPAS (Serial Poll Active State) could cause the **controller** to read the status byte incorrectly, due to the changing state of the bus data I/O lines. To prevent this problem, **devices** shall latch the contents of the **device** status byte and RQS message as presented on the DIO lines while the **device** is in IEC 625-1 SPAS and the IEC 625-1 source handshake is in SDYS (Source Delay State) or STRS (Source Transfer State). The latch guarantees that the status byte will not change on the bus just as the IEC 625-1 SR diagram assures that the RQS message will not change. This requirement does not imply that the contents of the Status Byte Register may not be changed during this time, only that the value presented to the **controller** on DIO1-8 will be stable. In figure 11-4, the input from the IEC 625-1 Source Handshake/IEC 625-1 Service Request block determines whether or not the Status Byte Latch is transparent.

11.3.3.4 Implementation techniques

Many commercially available IEC 625-1 integrated circuits implement the Set rsv State Diagram in place of, or in addition to, providing direct control over the rsv local message. These circuits also include a status byte buffer which allows the IEC 625-1 control circuit to respond to a serial poll without the intervention of **device** firmware. This section describes how the requirements of the IEC 625-2 status reporting model may be met using such circuits. It does not sanction the use of specific components or prohibit alternate implementations which meet the requirements of this standard, but is meant as a discussion of common implementation techniques and pitfalls.

11.3.3.4.1 Preferred implementation

The preferred implementation from a device designer's point of view, would be to do the following when the contents of the status byte buffer is updated:

1) set **reqt** TRUE if "New Reason for Service" is TRUE, or
2) set **reqf** TRUE if MSS is FALSE.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Ici, la "nouvelle raison de service" est un argument supplémentaire à l'instruction, généré par la microprogrammation de l'**appareil** suivant 11.3.3.1. Cette implémentation évite la condition de course de 11.3.3.4.2 et les demandes de services parasites de 11.3.3.4.3, mais exige que neuf bits d'information soient spécifiés avec l'instruction (sept bits d'état, MSS, et la nouvelle raison de service).

11.3.3.4.2 Contrôle indépendant autorisé de STB, **reqt**, et **reqf**

Certains circuits intégrés permettent au contenu de la mémoire tampon de l'octet d'état du circuit d'être mis à jour à n'importe quel moment sans changer l'état de rsv. De tels circuits ont trois entrées:

1) Une instruction qui met à jour le contenu des mémoires tampons de l'octet d'état du circuit.
2) Une instruction qui met **reqt** VRAI.
3) Une instruction qui met **reqf** VRAI.

Les transitions de VRAI à FAUX de **reqt** et **reqf** sont générées de façon interne par le circuit selon le diagramme d'état rsv d'ensemble. (2) et (3) peuvent être implémentés comme des instructions séparées ou comme une seule instruction avec un argument.

Lorsque l'octet d'état ou le registre de validation d'état de l'**appareil** change, la micro-programmation de l'**appareil** doit:

1) Mettre à jour la mémoire tampon de l'octet d'état du circuit.
2) S'il y a une nouvelle raison de service (voir 11.3.3.1), mettre **reqt** VRAI.
3) S'il n'y a pas de raison de service (MSS FAUX), mettre **reqf** VRAI.

Noter qu'il y a une course entre la microprogrammation de l'appareil positionnant **reqt** ou **reqf** VRAI et le **contrôleur** interrogeant l'**appareil** en réponse à une demande de service antérieure. Si le **contrôleur** interroge l'**appareil** une fois qu'il a mis à jour la mémoire tampon de l'octet d'état, mais avant qu'il ait mis **reqt** VRAI, l'**appareil** générera une demande de service une fois que le contrôleur aura lu le nouvel octet d'état. Si le **contrôleur** interroge l'**appareil** une fois qu'il a mis à jour la mémoire tampon de l'octet d'état, mais avant qu'il ait mis **reqf** VRAI, le **contrôleur** recevra le nouvel octet d'état (qui montre qu'il n'y a aucune raison de demande de service) avec RQS VRAI (indiquant une demande de service). Cette course ne pose pas de problèmes en pratique, puisque le **contrôleur** reçoit dans tous les cas l'état de l'**appareil** le plus récent. Les implémentations qui mettent **reqt** VRAI avant de mettre à jour la mémoire tampon de l'octet d'état ne sont pas autorisées car elles peuvent provoquer une violation de 11.3.3, règle 2.

11.3.3.4.3 Contrôle couplé autorisé de STB, **reqt** et **reqf**

Certains circuits intégrés nécessitent que **reqt** ou **reqf** soit mis VRAI chaque fois que la mémoire tampon de l'octet d'état est mise à jour. De tels circuits ont une entrée.

Lorsque le contenu de la mémoire tampon de l'octet d'état est mis à jour, le circuit met:

1) **reqt** VRAI si MSS est VRAI, ou
2) **reqf** VRAI si MSS est FAUX.

Cette implémentation est autorisée par cette norme (avant-dernier alinéa de 11.3.3.1), mais peut générer une demande de service lorsqu'un bit de l'octet d'état change, même si ce bit n'a pas été validé pour générer une demande de service.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Here, "New Reason for Service" is an additional argument to the command, generated by the device firmware in accordance with 11.3.3.1. This implementation avoids the race condition of 11.3.3.4.2 and the spurious service requests of 11.3.3.4.3, but requires that nine bits of information be specified with the command (seven bits of status, MSS, and New Reason for Service).

11.3.3.4.2 Allowed Independent Control of STB, **reqt**, and **reqf**

Some integrated circuits allow the contents of the circuit's status byte buffer to be updated at any time without changing the state of rsv. Such circuits have three inputs:

1) A command which updates the contents of the circuit's status byte buffer.
2) A command which sets **reqt** TRUE.
3) A command which sets **reqf** TRUE.

TRUE to FALSE transitions of **reqt** and **reqf** are generated internally by the circuit according to the Set rsv State Diagram. (2) and (3) may be implemented as separate commands or as a single command with one argument.

When the **device** status byte or Status Enable Register changes, the **device** firmware must:

1) Update the circuit's status byte buffer.
2) If there is a New Reason for Service (see 11.3.3.1), set **reqt** TRUE.
3) If there is no reason for service (MSS FALSE), set **reqf** TRUE.

Note that there is a race between the device's firmware setting **reqt** or **reqf** TRUE and the **controller's** polling the **device** in response to a previous service request. If the **controller** polls the **device** after it has updated the status byte buffer but before it has set **reqt** TRUE, the **device** will generate a service request after the **controller** has read the new status byte. If the **controller** polls the **device** after it has updated the status byte buffer but before it has set **reqf** TRUE, the **controller** will receive the new status byte (which shows no reason for service request) with RQS TRUE (indicating a request for service). This race does not cause problems in practice, as the **controller** receives the most recent **device** status in either case. Implementations which set **reqt** TRUE before updating the status byte buffer are not allowed, as they could cause a violation of 11.3.3, rule 2.

11.3.3.4.3 Allowed Coupled Control of STB, **reqt**, and **reqf**

Some integrated circuits require that one of **reqt** or **reqf** be set TRUE each time the status byte buffer is updated. Such circuits have one input.

When the contents of the status byte buffer are updated, the circuit sets:

1) **reqt** TRUE if MSS is TRUE, or
2) **reqf** TRUE if MSS is FALSE.

This implementation is allowed by this standard (penultimate paragraph of 11.3.3.1) but can generate a service request when a bit in the status byte changes, even if that bit has not been enabled to cause a service request.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.3.3.4.4 Autres implémentations autorisées

On peut utiliser d'autres techniques d'implémentation qui répondent aux exigences de la présente norme.

11.4 *Structures de données d'état*

11.4.1 *Vue d'ensemble*

Toutes les structures de données d'état d'**appareil** devront suivre soit le modèle de registre, soit le modèle de file d'attente, comme défini en 11.4. Les structures de données d'état fournissent une voie commune au concepteur d'appareil pour réaliser l'indication d'état. Chaque structure de données d'état a une "sortie" unique, un message récapitulatif qui résume l'état de la structure. Un état récapitulatif FAUX signifie qu'il n'y a pas d'état à indiquer et un état récapitulatif VRAI indique qu'il y a un état à indiquer.

11.4.2 *Structure de données d'état – Modèle de registre*

La structure de données d'état de modèle de registre permet au concepteur d'appareil de résumer de multiples événements dans un message récapitulatif unique dans le registre d'octet d'état. La figure 11-6 illustre une structure généralisée de données d'état de modèle de registre.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.3.3.4.4 Other allowed implementations

Other implementation techniques that satisfy the requirements of this standard may be used.

11.4 *Status Data Structures*

11.4.1 *Overview*

All **device** Status Data Structures shall follow either the register model or the queue model as defined in 11.4. The Status Data Structures provide a common way for the device designer to perform status reporting. Each status data structure has a single "output", a summary message which summarizes the structure's state. A FALSE summary-status means there is no status to report and TRUE summary status indicates there is status to report.

11.4.2 *Status Data Structure – Register model*

The register model Status Data Structure allows the device designer to summarize multiple events in a single summary message in the Status Byte Register. Figure 11-6 illustrates a generalized register model Status Data Structure.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

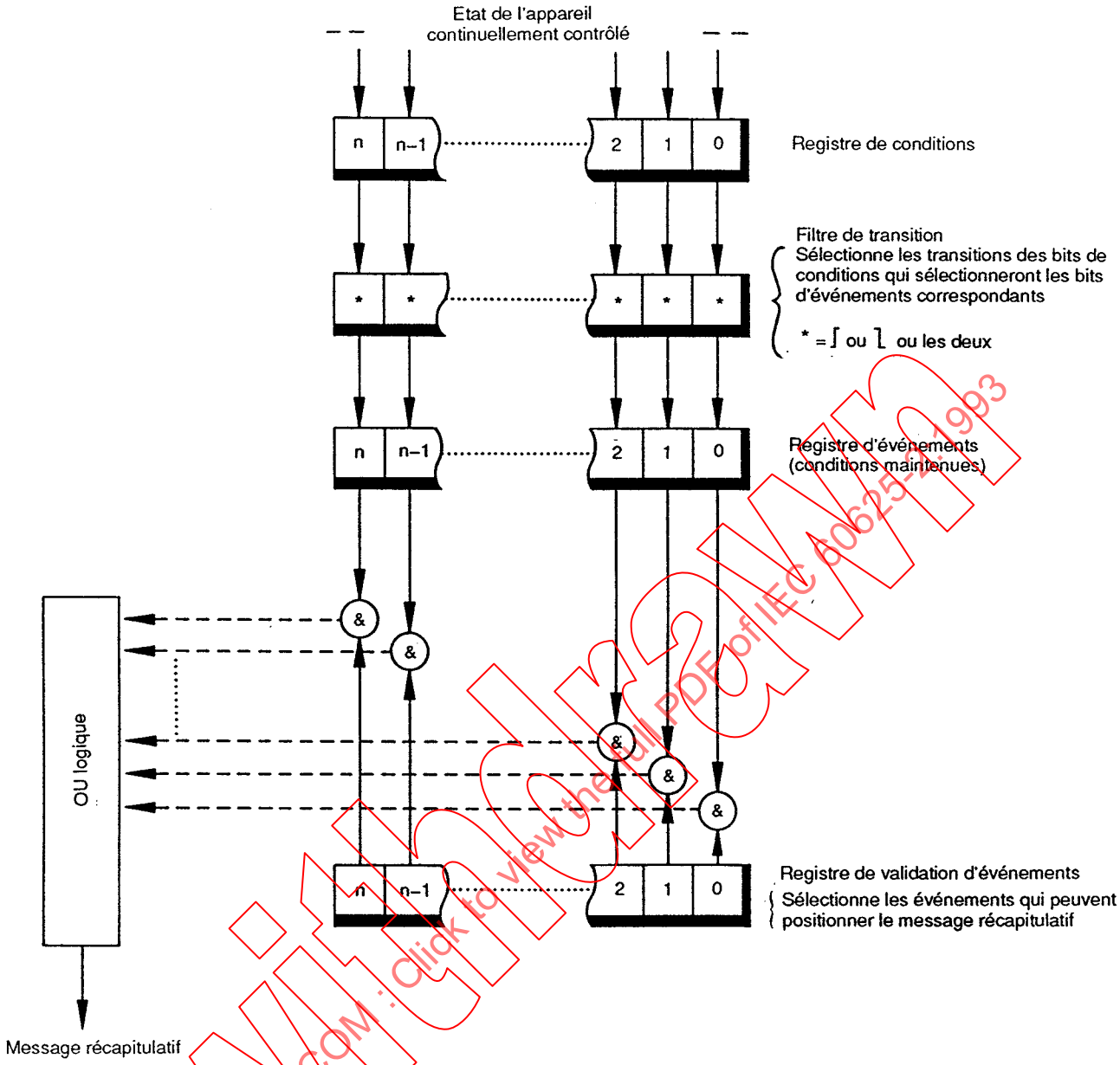


Figure 11-6 – Structure de données d'état – Modèle de registre

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.4.2.1 Registres de conditions

11.4.2.1.1 Fonction

Une condition est un état de l'**appareil** qui est soit FAUX, soit VRAI. Un registre de conditions reflète ces états dans ses bits de conditions. Un registre de conditions peut être compris dans une gamme de longueurs allant de 1 à 16 bits maximum et peut contenir des bits inutilisés. Les bits inutilisés devront renvoyer une valeur zéro lorsqu'ils sont lus.

NOTE – Certains **contrôleurs** peuvent avoir des difficultés à lire un nombre entier non signé de 16 bits dans leur format interne.

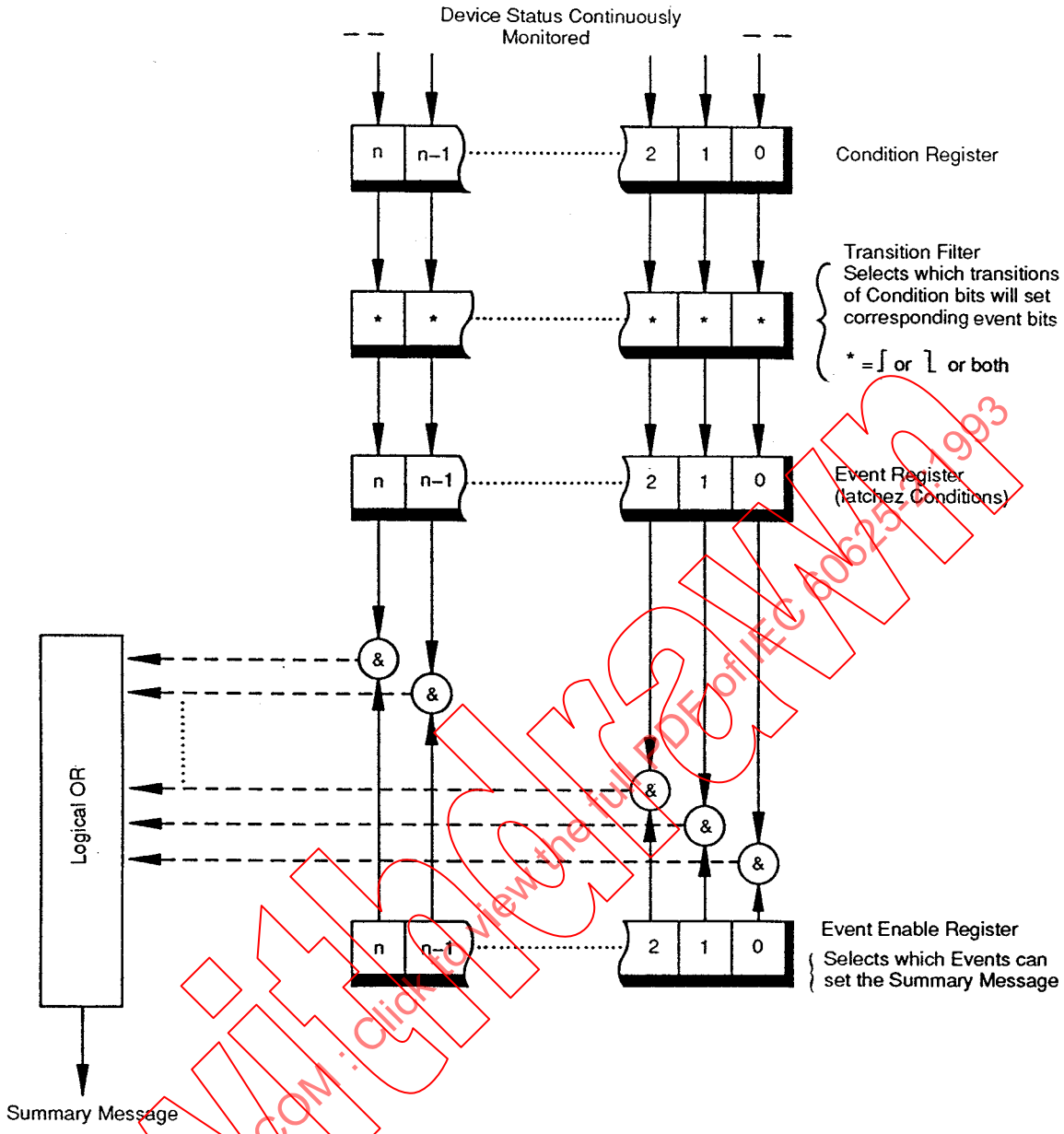


Figure 11-6 – Status Data Structure – Register model

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.4.2.1 Condition Registers

11.4.2.1.1 Function

A condition is a **device** state which is either TRUE or FALSE. A Condition Register reflects these states in its condition bits. A Condition Register may range in length from 1 to a maximum of 16 bits and may contain unused bits. Unused bits shall return a zero value when read.

NOTE - Some **controllers** may have difficulty reading a 16-bit unsigned integer into their internal format.

Typiquement, les bits de conditions représentent des états spécifiques de l'**appareil**. En alternative, les bits de conditions peuvent représenter des informations récapitulatives d'une structure de données d'état secondaire de recouvrement. Cette structure de données d'état secondaire devra être l'une des structures de données d'état définies à 11.4, soit le modèle de registre, soit le modèle de file d'attente. Cette technique de recouvrement peut être étendue à des couches supplémentaires, si nécessaire.

L'implémentation des registres de conditions est facultative puisque le seul but d'un registre de conditions est de grouper les états de l'**appareil** relatifs ou les messages récapitulatifs de recouvrement pour qu'ils puissent être lus ensemble.

Cette norme ne définit aucune structure de données d'état spécifique qui contienne un registre de conditions.

11.4.2.1.2 Lecture des registres de conditions

L'**appareil** peut fournir à lui-même des instructions spécifiques pour lire un ou tous les registres de conditions de l'**appareil**. Le fait de lire un registre de conditions ne devra pas changer son contenu.

11.4.2.1.3 Ecriture dans les registres de conditions

Aucune instruction spécifique à l'appareil ne devra directement écrire dans un registre de conditions. Seuls les changements dans l'état de l'**appareil** ou dans une structure de données d'état secondaire de recouvrement peuvent modifier le contenu d'un registre de conditions.

11.4.2.1.4 Effacement des registres de conditions

Aucune instruction spécifique à l'appareil ne devra directement effacer un registre de conditions. Seuls les changements dans l'état de l'**appareil** ou l'effacement d'une structure de données d'état secondaire de recouvrement peuvent effacer un bit de condition.

11.4.2.2 Registres d'événements

11.4.2.2.1 Fonction

Les registres d'événements détectent les changements dans les conditions. Chaque bit d'événement dans un registre d'événements devra correspondre à un bit de condition dans un registre de conditions ou (s'il n'y a pas de registre de conditions) à une condition spécifique dans l'**appareil**. Un événement deviendra VRAI lorsque la condition associée de l'**appareil** créera certaines transitions définies par l'appareil.

La définition du registre d'événements garantit que l'application ne peut pas manquer un événement qui se produit entre la lecture et l'effacement d'un registre d'événements. Un bit de registre d'événements (bit d'événement) devra être mis VRAI lorsqu'un événement associé se produit. Ces bits, une fois positionnés, sont "bloqués". C'est-à-dire qu'ils ne peuvent pas être effacés, même s'ils ne reflètent pas l'état courant d'une condition relative, jusqu'à ce qu'ils soient lus par l'application.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Typically, the condition bits represent specific **device** states. Alternately, the condition bits may represent summary information from an overlaying secondary Status Data Structure. This secondary Status Data Structure shall be one of the Status Data Structures defined in 11.4, either the register model or the queue model. This overlaying technique may be extended to additional layers as required.

Implementation of Condition Registers is optional, since the sole purpose of a Condition Register is to group-related **device** states or overlaying summary messages so that they may be read together.

This standard does not define any specific Status Data Structure that contains a Condition Register.

11.4.2.1.2 Reading Condition Registers

The **device** may provide device-specific commands to read any or all of the **device's** Condition Registers. Reading a Condition Register shall not change its contents.

11.4.2.1.3 Writing Condition Registers

No device-specific commands shall directly write to a Condition Register. Only changes in **device** state or in an overlaying secondary Status Data Structure may alter the contents of a Condition Register.

11.4.2.1.4 Clearing Condition Registers

No device-specific commands shall directly clear a Condition Register. Only changes in **device** state or clearing of an overlaying secondary Status Data Structure may clear a condition bit.

11.4.2.2 Event Registers

11.4.2.2.1 Function

Event Registers capture changes in conditions. Each event bit in an event register shall correspond to a condition-bit in a Condition Register or (if there is no Condition Register) to a specific condition in the **device**. An event shall become TRUE when the associated **device** condition makes certain device-defined transitions.

The Event Register definition guarantees that the application cannot miss an event which occurs between the reading and clearing of an Event Register. An Event Register bit (event bit) shall be set TRUE when an associated event occurs. These bits, once set, are "sticky". That is, they cannot be cleared even if they do not reflect the current status of a related condition, until they are read by the application.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Cette norme autorise les trois critères de transition suivants pour mettre les bits d'événements VRAIS (voir filtre de transition figure 11-6):

1) Transition positive. L'événement devient VRAI lorsque sa condition associée produit une transition de FAUX à VRAI seulement.

2) Transition négative. L'événement devient VRAI lorsque sa condition associée produit une transition de VRAI à FAUX seulement.

3) Transition positive ou négative. L'événement devient VRAI lorsque sa condition associée produit une transition soit de FAUX à VRAI, soit de VRAI à FAUX.

Le concepteur de l'appareil devra spécifier lequel des critères de transition ci-dessus doit être utilisé pour chaque bit d'événement. L'**appareil** peut fournir des instructions spécifiques à l'appareil pour changer la transition associée avec chaque condition et bit d'événement.

Dans tous les cas, le changement d'une condition de l'**appareil** ne devra pas causer le changement d'un bit d'événement de VRAI à FAUX.

Les registres d'événements peuvent être compris dans une gamme de longueurs allant de 1 à 16 bits maximum et peuvent contenir des bits "inutilisés". Les bits inutilisés devront être à zéro lorsque le registre est lu.

NOTE – Certains **contrôleurs** peuvent avoir des difficultés à lire un nombre entier sans signe de 16 bits dans leur format interne.

Cette norme spécifie complètement un seul registre d'événements, le registre d'état d'événements standards (voir 11.5.1).

11.4.2.2.2 Lecture des registres d'événements

L'appareil peut fournir des interrogations spécifiques à l'appareil pour lire un ou tous les registres d'événements de l'appareil. La seule interrogation commune qui lit un registre d'événements est l'interrogation *ESR?, qui lit le registre d'état d'événements standards (voir 10.12 et 11.5.1.2.2).

11.4.2.2.3 Ecriture dans les registres d'événements

Aucune instruction dépendante de l'appareil ne devra autoriser le programmeur d'application à écrire directement des bits dans un registre d'événements. Des instructions peuvent, par contre, effacer le registre en entier.

11.4.2.2.4 Effacement des registres d'événements

Les registres d'événements devront être effacés lors d'une lecture par une interrogation. Seul(s) le (ou les) registre(s) d'événements standards interrogé(s) devra (devront) être effacé(s).

En outre, des instructions spécifiques, par contraste avec les interrogations, peuvent être implémentées pour effacer (et non lire) les registres d'événements.

L'instruction commune *CLS est une instruction obligatoire (voir 10.3) qui efface tous les registres d'événements. Tous les registres d'événements définis par l'appareil de même que le registre d'état d'événements standards (voir 11.5.1) devront être effacés par *CLS.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

This standard allows the following three transition criteria for setting event bits TRUE (see transition filter in figure 11-6):

1) Positive transition. The event becomes TRUE when its associated condition makes a FALSE to TRUE transition only.

2) Negative transition. The event becomes TRUE when its associated condition makes a TRUE to FALSE transition only.

3) Positive or negative transition. The event becomes TRUE when its associated condition makes either a FALSE to TRUE or a TRUE to FALSE transition.

The device designer shall specify which of the above transition criteria is to be used for each event bit. The **device** may provide device-specific commands to change the transition associated with each condition and event bit.

In all cases, the change of a **device** condition shall not cause an event bit to go from TRUE to FALSE.

Event Registers may range in length from 1 to a maximum of 16 bits and may contain "unused" bits. Unused bits shall be zero when the register is read.

NOTE - Some **controllers** may have difficulty reading a 16-bit unsigned integer into their internal format.

This standard completely specifies only one Event Register, the Standard Event Status Register (see 11.5.1).

11.4.2.2.2 Reading Event Registers

The **device** may provide device-specific queries to read any or all of the device's Event Registers. The only common query which reads an Event Register is the *ESR? query, which reads the Standard Event Status Register (see 10.12 and 11.5.1.2.2).

11.4.2.2.3 Writing Event Registers

No device-specific command shall allow the application programmer to directly write bits in an Event Register. Commands may, however, clear the entire register.

11.4.2.2.4 Clearing the Event Registers

Event Registers shall be cleared when read by a query. Only the Event Register(s) queried shall be cleared.

Additionally, specific commands, in contrast to queries, may be implemented to clear (and not read) Event Registers.

The *CLS common command is a mandatory command (see 10.3) which clears all Event Registers. All device-defined Event Registers as well as the Standard Event Status Register (see 11.5.1) shall be cleared by *CLS.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Les registres d'événements peuvent être effacés à la mise sous tension. Les registres d'événements ne devront pas être effacés d'une autre façon que ce qui est indiqué ci-dessus.

11.4.2.3 Registres de validation d'événements

11.4.2.3.1 Fonction

Les registres de validation d'événements sélectionnent les bits d'événements, dans le registre d'événements correspondant, qui provoqueront un message récapitulatif VRAI lors de leur sélection. Chaque bit d'événement devra avoir un bit de validation correspondant dans le registre de validation d'événements. En utilisant les bits de validation, un programmeur d'application peut programmer un **appareil** pour demander un service pour un événement unique ou un OU inclusif de n'importe quel groupe d'événements.

Chaque registre de validation d'événements devra avoir la même longueur que le registre d'événements correspondant. Tous les bits inutilisés dans le registre de validation d'événements devront correspondre avec les bits inutilisés dans le registre d'événements. La valeur des bits inutilisés devra être zéro lorsque le registre de validation d'événements est lu et devra être ignorée lorsqu'il est écrit par des instructions.

Cette norme ne spécifie complètement qu'un registre de validation d'événements, le registre de validation d'événements standard (voir 11.5.1.3).

11.4.2.3.2 Lecture des registres de validation d'événements

L'**appareil** peut fournir des interrogations définies par l'appareil pour lire un ou tous les registres de validation d'événements de l'**appareil**. Si un **appareil** fournit une instruction pour écrire dans un registre de validation d'événements et une interrogation pour lire le même registre de validation d'événements, l'<EN-TÊTE D'INTERROGATION> devra être formé en plaçant un "?" après l'<EN-TÊTE D'INSTRUCTION> qui sélectionne le registre (voir 11.4.2.3.3).

La réponse à l'interrogation devra être renvoyée comme un élément de <DONNÉES DE RÉPONSE NUMÉRIQUES DÉCIMALES NR1>. Les poids binaires des bits devront être utilisés pour former la valeur NR1. Un bit inutilisé devra avoir la valeur zéro.

La seule interrogation commune définie qui lit un registre de validation d'événements est *ESE? qui lit le registre de validation d'état d'événements standards (voir 10.11 et 11.5.1.3.2).

11.4.2.3.3 Ecriture dans les registres de validation d'événements

L'**appareil** peut fournir des instructions spécifiques à l'appareil pour écrire dans un ou la totalité des registres de validation d'événements de l'**appareil**. Si une telle instruction est implémentée, elle devra suivre le format décrit dans cette section.

Le format de l'instruction consistera en un en-tête défini par l'appareil suivi par un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>. L'**appareil** peut également accepter facultativement un élément de <DONNÉE D'EXPRESSION> qui évalue un élément de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>.

La <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE>, lorsqu'elle est arrondie à une valeur entière et exprimée en base 2 (binaire), devra représenter les valeurs individuelles des bits du registre de validation d'événements. Une valeur de bit de 1 indiquera une condition validée. Une valeur de bit de zéro indiquera une condition invalidée.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Event Registers may be cleared at power-on. Event Registers shall not be cleared in any way other than as stated above.

11.4.2.3 Event Enable Registers

11.4.2.3.1 Function

Event Enable Registers select which event bits in the corresponding Event Register will cause a TRUE summary message when set. Each event bit shall have a corresponding enable bit in the Event Enable Register. By use of the enable bits, an application programmer can program a **device** to request service for a single event or an inclusive OR of any group of events.

Each Event Enable Register shall be the same length as the corresponding Event Register. Any unused bits in the Event Enable Register shall correspond with unused bits in the Event Register. The value of unused bits shall be zero when the Event Enable Register is read and shall be ignored when written to by commands.

This standard completely specifies only one Event Enable Register, the Standard Event Status Enable Register (see 11.5.1.3).

11.4.2.3.2 Reading Event Enable Registers

The **device** may provide device-defined queries to read any or all of the **device's** Event Enable Registers. If a **device** provides a command to write to an Event Enable Register and a query to read the same Event Enable Register, the <QUERY HEADER> shall be formed by placing a "?" after the <COMMAND HEADER> that sets the register (see 11.4.2.3.3).

The query response shall be returned as an <NR1 NUMERIC RESPONSE DATA> element. Binary-weighting of the bits shall be used to form the NR1 value. An unused bit shall have the value zero.

The only common query which reads an Event Enable Register is *ESE? which reads the Standard Event Status Enable Register (see 10.11 and 11.5.1.3.2).

11.4.2.3.3 Writing Event Enable Registers

The **device** may provide device-specific commands to write to any or all of the **device's** Event Enable Registers. If such a command is implemented, it shall follow the format as described in this section.

The command format shall consist of a device-defined header followed by a <DECIMAL NUMERIC PROGRAM DATA> element. The **device** may also optionally accept an <EXPRESSION DATA> element that evaluates to a <DECIMAL NUMERIC PROGRAM DATA> element.

The <DECIMAL NUMERIC PROGRAM DATA> when rounded to an integer value and expressed in base 2 (binary) shall represent the individual bit values of the Event Enable Register. A bit value of one shall indicate an enabled condition. A bit value of zero shall indicate a disabled condition.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Les instructions spécifiques à l'appareil supplémentaires qui utilisent d'autres formats sont aussi autorisées. Si aucune instruction n'est fournie pour écrire dans un registre de validation d'événements, chaque bit utilisé dans le registre aura alors une valeur de 1 et chaque bit inutilisé aura une valeur de 0.

La seule instruction commune définie qui écrit dans un registre de validation d'événements est l'instruction commune *ESE, qui écrit dans le registre de validation d'état d'événements standards (voir 10.10).

11.4.2.3.4 Effacement des registres de validation d'événements

Si l'**appareil** fournit une instruction spécifique à l'appareil pour écrire dans un registre de validation d'événements (voir 11.4.2.3.3), le registre peut être effacé par l'envoi de cette instruction avec un argument zéro (format NRf).

Les registres de validation d'événements seront effacés à la mise sous tension si l'instruction commune facultative *PSC n'est pas implémentée, ou si l'indicateur d'effacement d'état à la mise sous tension est VRAI (voir 5.12).

Aucune autre condition d'**appareil** ou d'interface, y compris la réception du message **dcas**, ne devra modifier un registre de validation d'événements.

11.4.3 *Structure de données d'état – Modèle de file d'attente*

La structure de données d'état de modèle de file d'attente permet au concepteur de l'appareil d'indiquer des états séquentiels ou une autre information. La présence de telles informations dans une file d'attente est résumée dans un message récapitulatif.

Une structure de données d'état utilisant le modèle de file d'attente, la file d'attente de sortie avec son message récapitulatif associé MAV (bit 4 de l'octet d'état), est définie par la présente norme (voir 11.5.2).

11.4.3.1 Fonction

La figure 11-7 représente un modèle de file d'attente pour une structure de données d'état. La file d'attente est une structure de données contenant une liste séquentielle d'informations. La file d'attente est vide lorsque toutes les informations de la liste ont été lues.

Les articles peuvent être placés dans la file d'attente dans n'importe quel ordre. Les articles devront être enlevés de la file d'attente dès qu'ils sont lus. La file d'attente peut être effacée dans les conditions spécifiées à 11.4.3.4. Le message récapitulatif associé est VRAI si la file d'attente n'est pas vide et FAUX si elle est vide.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Additional device-specific commands which use other formats are also allowed. If no command is provided to write to an Event Enable Register then every used bit in the register shall have the value of one and every unused bit shall have a value of zero.

The only defined common command which writes to an Event Enable Register is the *ESE common command, which writes to the Standard Event Status Enable Register (see 10.10).

11.4.2.3.4 Clearing Event Enable Registers

If the **device** provides a device-specific command to write to an Event Enable Register (see 11.4.2.3.3), the register may be cleared by sending that command with an argument of zero (NRf format).

Event Enable Registers shall be cleared at power-on if the optional *PSC common command is not implemented, or if power-on status clear flag is TRUE (see 5.12).

No other **device** or interface conditions, including receipt of the **dcas** message, shall alter any Event Enable Register.

11.4.3 *Status Data Structure – Queue model*

The queue model Status Data Structure allows the device designer to report sequential status or other information. The presence of such information in a Queue is summarized in a summary message.

One Status Data Structure using the queue model, the Output Queue, with its associated MAV summary message (bit 4 of the status byte), is defined by this standard (see 11.5.2).

11.4.3.1 Function

Figure 11-7 represents the queue model for a Status Data Structure. The Queue is a data structure containing a sequential list of information. The Queue is empty when all information has been read from the list.

Items may be placed in the Queue in any order. Items shall be removed from the Queue as they are read. The Queue may be cleared under the conditions specified in 11.4.3.4. The associated summary message is TRUE if the Queue is not empty and FALSE if it is empty.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

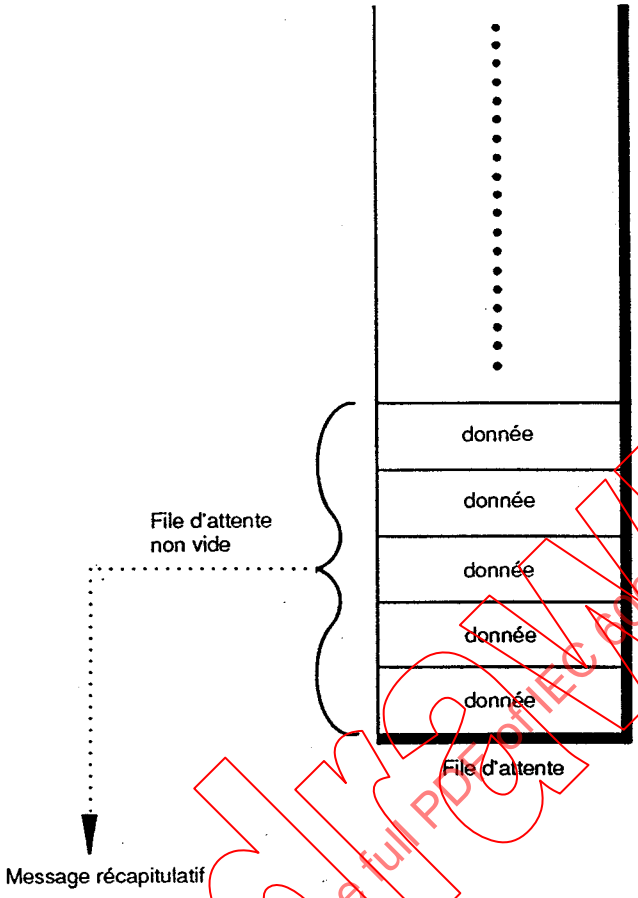


Figure 11-7 – Structure de données d'état – Modèle de file d'attente

Tous les éléments d'<UNITÉ DE MESSAGE DE RÉPONSE> dans une file d'attente (autre que la file d'attente de sortie) devront être du même type (voir figure 8-4). Le contenu spécifique de l'élément de donnée envoyé est défini par l'appareil. Cette règle assure qu'un programme d'application peut savoir d'avance le format de n'importe quelle donnée lue.

EXEMPLE: Une file d'attente pourrait contenir une liste d'éléments de <DONNÉES DE RÉPONSE DE CARACTÈRES>. Une autre file d'attente pourrait contenir une liste d'éléments de <DONNÉES DE RÉPONSE NUMÉRIQUES NR3>. Une autre encore pourrait contenir une unité de message avec deux éléments séparés par une virgule comme: <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>, <DONNÉE DE RÉPONSE DE CHAÎNE>.

L'ordre dans lequel les articles sont lus dans une file d'attente (autre que la file d'attente de sortie) est en dehors du domaine d'application de cette norme. Une file d'attente définie par l'appareil ne nécessite pas d'être lue suivant le système premier entré, premier sorti (FIFO).

Le concepteur de l'appareil peut imposer une longueur maximale de la structure de la file d'attente. Les règles gouvernant les conditions de dépassement de capacité d'une file d'attente, autre que la file d'attente de sortie, sont en dehors du domaine d'application de cette norme.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

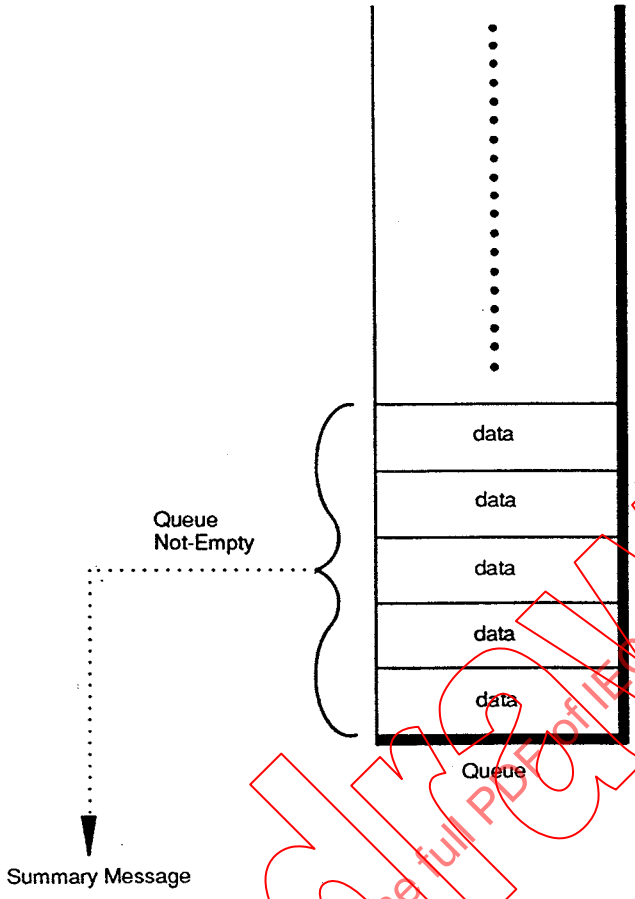


Figure 11-7 – Status Data Structure – Queue model

All <RESPONSE MESSAGE UNIT> elements in a Queue (other than the Output Queue) shall be of the same type (see figure 8-4). The specific content of the data element sent is device defined. This rule assures that an application program can know, in advance, the format of any data read.

EXAMPLE: A Queue could contain a list of <CHARACTER RESPONSE DATA> elements. Another Queue could contain a list of <NR3 NUMERIC RESPONSE DATA> elements. Yet another could contain a message unit with the two elements separated by a comma as: <NR1 NUMERIC RESPONSE DATA>, <STRING RESPONSE DATA>.

The order that the items are read from a Queue (other than the Output Queue) is beyond the scope of this standard. A device-defined Queue need not be read in a first-in, first-out (FIFO) manner.

The device designer may impose a maximum length on the Queue structure. Rules governing a Queue overflow condition, other than the Output Queue, are beyond the scope of this standard.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.4.3.2 Lecture d'une file d'attente

Une file d'attente (autre que la file d'attente de sortie) peut être lue par des interrogations facultatives définies par l'appareil. De telles interrogations définies par l'appareil feront enlever l'article lu de la file d'attente et le feront placer dans la file d'attente de sortie.

11.4.3.3 Ecriture dans une file d'attente

Aucune file d'attente ne devra être écrite directement par des éléments de <MESSAGES DE PROGRAMMATION>. A part la file d'attente de sortie, la file d'attente représente une information codée de l'**appareil**, telle qu'un historique d'état de l'**appareil** ou une autre information, comme défini par le concepteur de l'appareil.

11.4.3.4 Effacement d'une file d'attente

Les files d'attente, sauf la file d'attente de sortie, devront être effacées lorsque l'un des faits suivants se produit:

1) Réception de l'instruction *CLS.

2) Lecture de tous les articles dans la file d'attente. Seules les données de la file d'attente interrogée doivent être enlevées.

3) Autres moyens spécifiques à l'appareil.

NOTE – La file d'attente de sortie est un cas spécial et n'est pas effacée par *CLS mais peut être effacée dans d'autres conditions (voir 6.3.2.3 et 10.3.1), y compris la méthode 2.

11.5 *Structures de données d'état standard*

La figure 11-8 représente un schéma de vue d'ensemble des structures de données d'état – état d'événement standard. Le modèle de registre d'état d'événement standard est traité en détail en 11.5.1. Les exigences concernant la file d'attente de sortie sont données en 11.5.2.

11.5.1 *Modèle de registre d'état d'événements standards*

La figure 11-9 représente le fonctionnement du modèle de registre d'état d'événements standards. Cette structure de données est une implémentation spécifique du modèle général de registre de structure de données d'état (voir 11.4.2) et est exigée par tous les **appareils**.

11.5.1.1 Définitions des bits du registre d'état d'événements standards

Cette norme affecte des événements spécifiques définis CEI 625-2 à des bits spécifiques dans le registre d'état d'événements standards.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.4.3.2 Reading a Queue

A Queue (other than the Output Queue) may be read by optional device-defined queries. Such device-defined queries shall cause the item read to be removed from the Queue and placed into the Output Queue.

11.4.3.3 Writing to a Queue

No Queue shall be written to directly by <PROGRAM MESSAGE> elements. Except for the Output Queue, the Queue represents encoded **device** information, such as a **device** status history or other information as defined by the device designer.

11.4.3.4 Clearing a Queue

Queues, except the Output Queue, shall be cleared when any of the following occur:

1) Receipt of the *CLS command.

2) Reading all the items in the Queues. Only the Queue that is queried shall have data removed.

3) Other device-specific means.

NOTE - The Output Queue is a special case and is not cleared by *CLS but can be cleared under other conditions (see 6.3.2.3 and 10.3.1), including method 2.

11.5 *Standard Status Data Structures*

Figure 11-8 represents an overview diagram of the Standard Event Status – Status Data Structures. The Standard Event Status Register model is discussed in detail in 11.5.1. Output Queue requirements are given in 11.5.2.

11.5.1 *Standard Event Status Register model*

Figure 11-9 represents the operation of the Standard Event Status Register Model. This data structure is a specific implementation of the general Status Data Structure register model (see 11.4.2) and is required by all **devices**.

11.5.1.1 Standard Event Status Register Bit Definitions

This standard assigns specific IEC 625-2 defined events to specific bits in the Standard Event Status Register.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

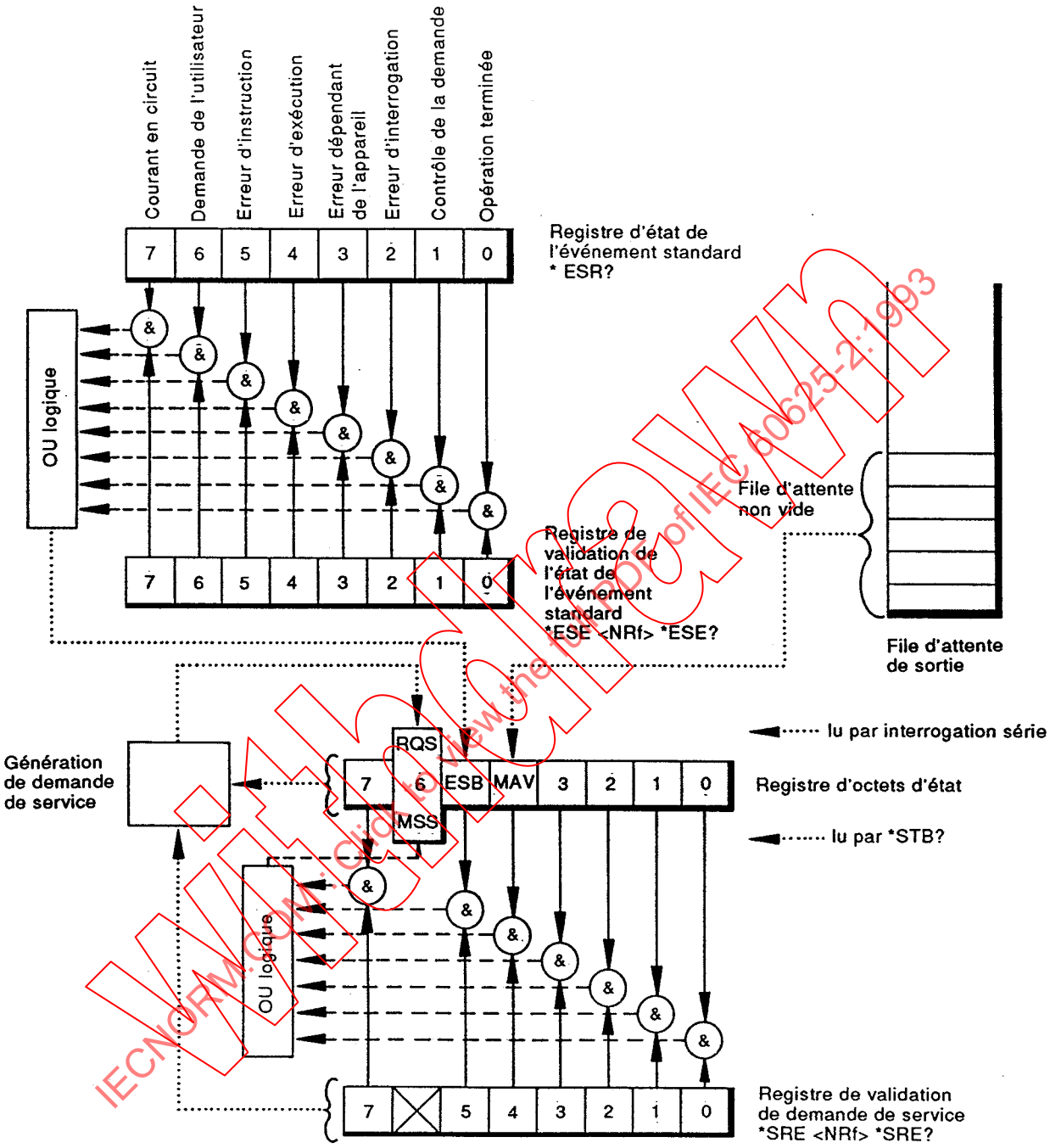


Figure 11-8 – Vue d'ensemble des structures de données d'état standard

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

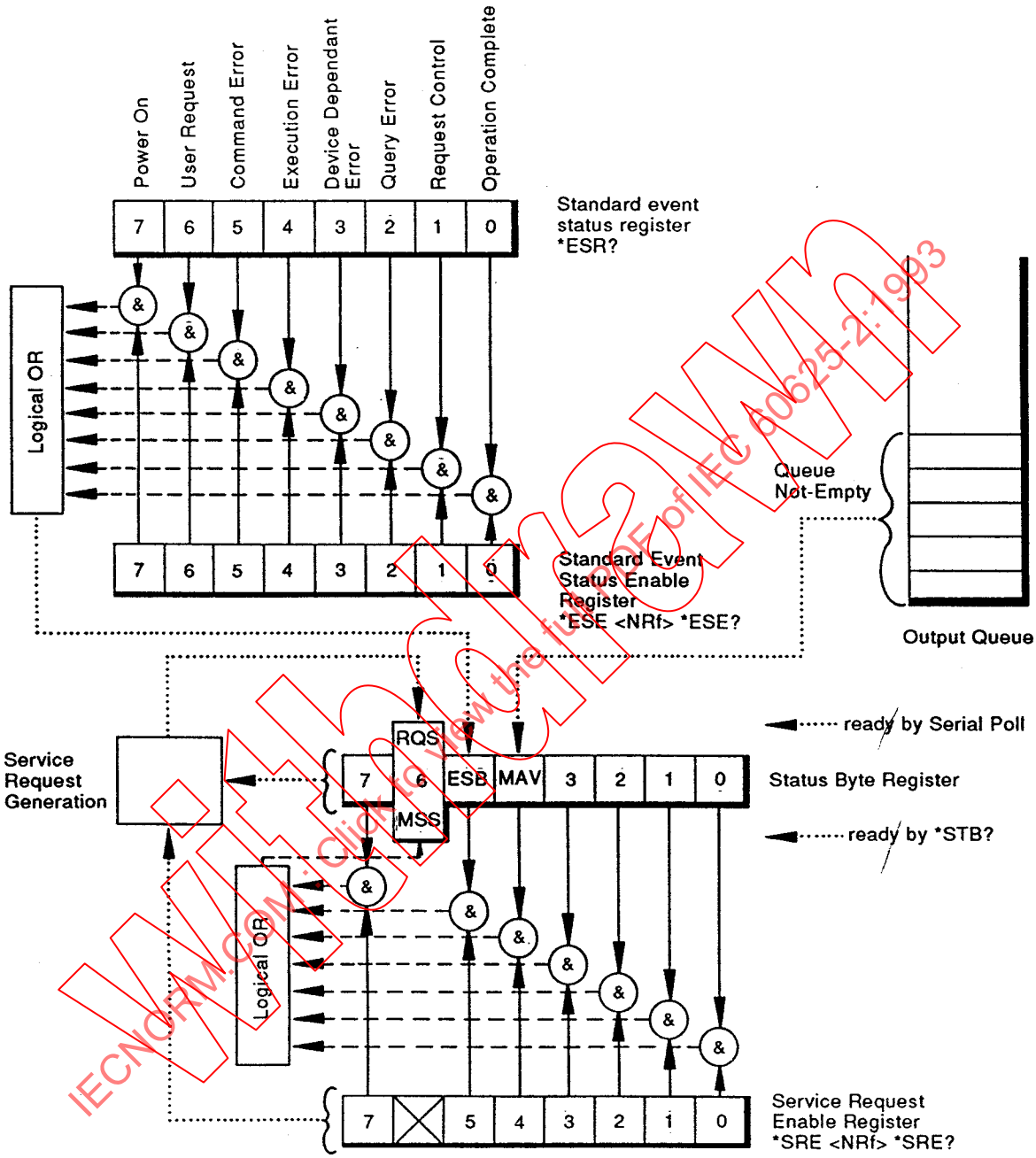


Figure 11-8 – Standard Status Data Structures overview

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

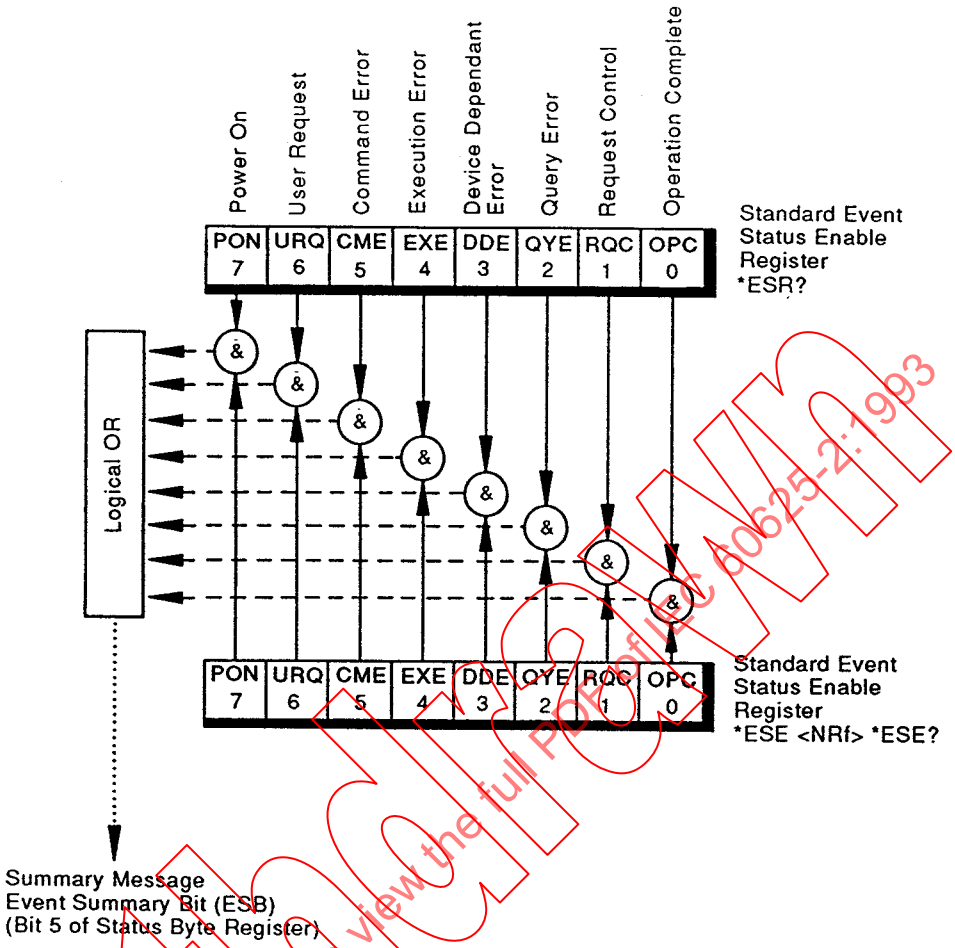


Figure 11-9 – Standard Event Status Register model

11.5.1.1.1 Bits 15 through 8 – Reserved

Standard Event Status Register bits 15 through 8 are reserved for possible future use by IEC. These bit values shall be reported as zero.

11.5.1.1.2 Bit 7 – Power-on (PON)

This event bit indicates that an off-to-on transition has occurred in the device's power supply.

11.5.1.1.3 Bit 6 – User Request (URQ)

This event bit indicates that one of a set of local controls (see 5.6.1.3) defined by the device designer as a User Request control has been activated. The setting of this event bit shall occur regardless of the IEC 625-1 Remote/Local state of the **device**.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.5.1.1.4 Bit 5 – ERREUR d'instruction (CME)

Les erreurs d'instruction sont détectées par l'analyseur syntaxique (voir 6.1.6). Ce bit d'événement indique que l'un des événements suivants s'est produit:

1) Une erreur de syntaxe CEI 625-2 a été détectée par l'analyseur syntaxique. C'est-à-dire qu'un message du contrôleur à l'appareil a été reçu, en violation avec cette norme. Les violations possibles comportent un élément de donnée violant les formats écouteur de l'**appareil** ou dont le type n'est pas acceptable par l'**appareil** (voir 7.1.2.2).

2) Une erreur sémantique s'est produite, indiquant qu'un en-tête inconnu a été reçu. Les en-têtes inconnus comprennent les en-têtes spécifiques à l'appareil incorrects et les instructions communes CEI 625-2 incorrectes ou non implémentées (article 10).

3) Un déclenchement d'exécution de groupe (GET) a été introduit dans la mémoire tampon d'entrée à l'intérieur d'un <MESSAGE DE PROGRAMMATION> (voir 6.1.4.5 et 6.1.6.1.1).

Lorsqu'un **appareil** détecte une erreur d'instruction, la synchronisation des analyseurs syntaxiques peut être perdue. Voir 6.5.4 pour la discussion de l'action de l'analyseur syntaxique après une erreur d'instruction.

Le bit d'erreur d'instruction ne devra pas être sélectionné pour indiquer toute autre condition spécifique à l'appareil. Les événements qui sont signalés comme erreurs d'instruction ne devront pas être également signalés comme des erreurs d'exécution, erreurs d'interrogation ou erreurs spécifiques à l'appareil. Voir les autres définitions de bit dans cet article.

11.5.1.1.5 Bit 4 – ERREUR d'exécution (EXE)

Les erreurs d'exécution sont détectées par le bloc de contrôle d'exécution (voir 6.1.7). Ce bit d'événement indique que:

1) Un élément de <DONNÉE DE PROGRAMMATION> suivant un en-tête a été évalué par l'**appareil** comme étant en dehors de sa gamme légale d'entrée ou sinon est incompatible avec les capacités de l'**appareil**.

2) Un message de programmation valide ne pourrait pas être traité correctement à cause d'une condition de l'**appareil**.

A la suite d'une erreur d'exécution, l'**appareil** devra continuer à traiter le train de travaux en entrée.

Les erreurs d'exécution devront être signalées par l'**appareil** une fois que les opérations d'arrondi et d'évaluation d'expression auront eu lieu. L'arrondi d'un élément de données numériques, par exemple, ne devra pas être signalé comme une erreur d'exécution.

Les événements qui génèrent des erreurs d'exécution ne devront pas aussi générer des erreurs d'instruction, des erreurs d'interrogation, ou des erreurs spécifiques à l'appareil. Voir les autres définitions de bit dans cet article.

11.5.1.1.6 Bit 3 – ERREUR spécifique à l'appareil (DDE)

Les erreurs spécifiques à l'appareil sont détectées par les fonctions de l'appareil (voir 6.1.8). Ce bit d'événement indique qu'une erreur s'est produite, qui n'est ni une erreur d'instruction, ni une erreur d'interrogation, ni une erreur d'exécution.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

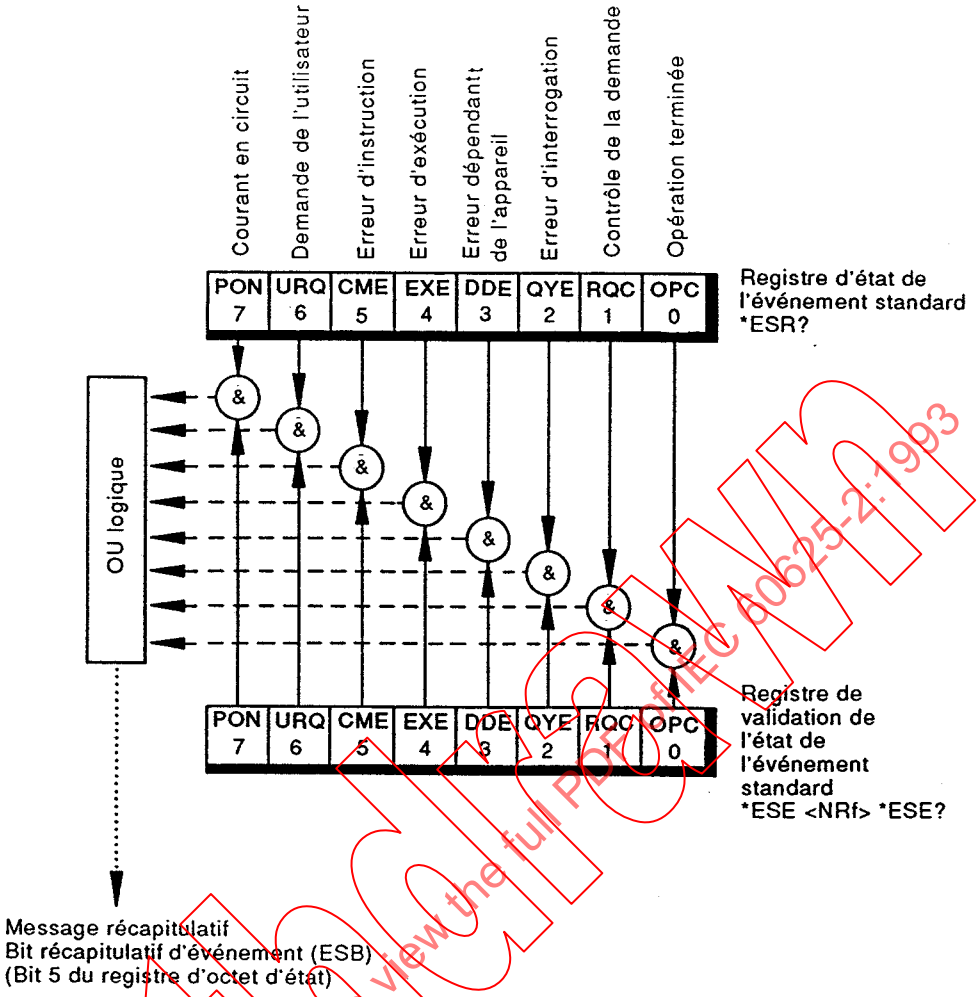


Figure 11-9 – Modèle de registre d'état d'événements standards

11.5.1.1.1 Bits 15 à 8 – réservés

Les bits 15 à 8 du registre d'état d'événements standards sont réservés pour une utilisation possible future par la CEI. Ces valeurs de bits doivent être indiquées comme zéro.

11.5.1.1.2 Bit 7 – Mise sous tension (PON)

Ce bit d'événement indique qu'une transition d'"éteint" à "allumé" s'est produite dans l'alimentation de l'appareil.

11.5.1.1.3 Bit 6 – Demande d'utilisateur (URQ)

Ce bit d'événement indique qu'un contrôle parmi le groupe de contrôles locaux (voir 5.6.1.3) défini par le concepteur de l'appareil comme un contrôle de demande d'utilisateur a été lancé. La sélection de ce bit d'événement devra se produire sans tenir compte de l'état éloigné/local CEI 625-1 de l'**appareil**.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.5.1.1.4 Bit 5 – Command ERROR (CME)

Command Errors are detected by the Parser (see 6.1.6). This event bit indicates that one of the following events has occurred:

1) An IEC 625-2 syntax error has been detected by the parser. That is, a controller-to-device message was received which is in violation of this standard. Possible violations include a data element which violates the **device** listening formats or whose type is unacceptable to the **device** (see 7.1.2.2).

2) A semantic error has occurred indicating that an unrecognized header was received. Unrecognized headers include incorrect device-specific headers and incorrect or unimplemented IEC 625-2 common commands (clause 10).

3) A Group Execute Trigger (GET) was entered into the Input Buffer inside a <PROGRAM MESSAGE> (see 6.1.4.5 and 6.1.6.1.1).

When a **device** detects a Command Error, parser synchronization may be lost. See 6.5.4 for discussion of parser action after a Command Error.

The Command Error bit shall not be set to report any other device-specific condition. Events that are reported as Command Errors shall not also be reported as Execution Errors, Query Errors, or Device-specific Errors. See other bit definitions in this clause.

11.5.1.1.5 Bit 4 – Execution ERROR (EXE)

Execution Errors are detected by the Execution Control Block (see 6.1.7). This event bit indicates that:

1) A <PROGRAM DATA> element following a header was evaluated by the **device** as outside of its legal input range or is otherwise inconsistent with the **device's** capabilities.

2) A valid program message could not be properly executed due to some **device** condition.

Following an Execution Error, the **device** shall continue to process the input stream.

Execution Errors shall be reported by the **device** after rounding and expression evaluation operations have taken place. Rounding a numeric data element, for example, shall not be reported as an Execution Error.

Events that generate Execution Errors shall not also generate Command Errors, Query Errors, or Device-specific Errors. See other bit definitions in this clause.

11.5.1.1.6 Bit 3 – Device-specific ERROR (DDE)

Device-specific Errors are detected by Device Functions (see 6.1.8). This event bit indicates that an error has occurred which is neither a Command Error, a Query Error, nor an Execution Error.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Une erreur spécifique à l'appareil est toute opération de l'appareil exécutée qui ne s'est pas terminée correctement en raison d'une certaine condition, telle qu'un dépassement de capacité.

A la suite d'une erreur spécifique à l'appareil, l'**appareil** devra continuer à traiter le train de travaux en entrée.

Les erreurs spécifiques à l'appareil doivent être utilisées au choix du concepteur de l'appareil.

Les événements qui génèrent des erreurs spécifiques à l'appareil ne devront pas aussi générer des erreurs d'instruction, des erreurs d'interrogation ou des erreurs d'exécution.

11.5.1.1.7 Bit 2 – ERREUR d'interrogation (QYE)

Les erreurs d'interrogation sont détectées par le contrôle de file d'attente de sortie (voir 6.1.10). Ce bit d'événement indique que, soit:

1) une tentative est faite pour lire les données de la file d'attente de sortie alors qu'aucune sortie n'est présente ou en cours,
2) soit une donnée dans la file d'attente de sortie a été perdue.

Voir 6.5.7 pour une description complète.

Le bit d'erreur d'interrogation ne devra pas être sélectionné pour signaler toute autre condition. Les événements qui génèrent des erreurs d'interrogation ne devront pas aussi générer des erreurs d'exécution, des erreurs d'instruction ou des erreurs spécifiques à l'appareil.

11.5.1.1.8 Bit 1 – Demande de contrôle (RQC)

Ce bit d'événement indique au **contrôleur** que l'**appareil** demande la permission de devenir le contrôleur en charge CEI 625-1 actif.

11.5.1.1.9 Bit 0 – Opération achevée (OPC)

Ce bit d'événement est généré en réponse à l'instruction *OPC?. Il indique que l'**appareil** a achevé toutes les opérations en cours sélectionnées. Voir 12.5.2 pour discussion sur la synchronisation de l'**appareil** utilisant ce bit d'événement.

11.5.1.2 Opération de registre d'état d'événements standards

11.5.1.2.1 Fonction

L'opération de registre d'état d'événements standards suit les règles pour registres d'événements dans le modèle de registre de structure de données d'état général décrit à 11.4.2.

11.5.1.2.2 Lecture

Le registre d'état d'événements standards est lu de façon destructive (c'est-à-dire lu et effacé) avec l'interrogation commune *ESR? (voir 10.12).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

A Device-specific Error is any executed device operation that did not properly complete due to some condition, such as overrange.

Following a Device-specific Error, the **device** shall continue to process the input stream.

Device-specific Errors are to be used at the discretion of the device designer.

Events that generate Device-specific Errors shall not also generate Command Errors, Query Errors, or Execution Errors.

11.5.1.1.7 Bit 2 – Query ERROR (QYE)

Query Errors are detected by the Output Queue Control (see 6.1.10). This event bit indicates that either:

1) an attempt is being made to read data from the Output Queue when no output is either present or pending;

2) or data in the Output Queue has been lost.

See 6.5.7 for a complete description.

The Query Error bit shall not be set to report any other condition. Events that generate Query Errors shall not also generate Execution Errors, Command Errors, or Device-specific Errors.

11.5.1.1.8 Bit 1 – Request Control (RQC)

This event bit indicates to the **controller** that the **device** is requesting permission to become the active IEC 625-1 controller-in-charge.

11.5.1.1.9 Bit 0 – Operation Complete (OPC)

This event bit is generated in response to the *OPC command. It indicates that the **device** has completed all selected pending operations. See 12.5.2 for a discussion of **device** synchronization using this event bit.

11.5.1.2 Standard Event Status Register operation

11.5.1.2.1 Function

Standard Event Status Register operation follows the rules for event registers in the general Status Data Structure register model described in 11.4.2.

11.5.1.2.2 Reading

The Standard Event Status Register is destructively read (that is, read and cleared) with the *ESR? common query (see 10.12).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.5.1.2.3 Ecriture

On ne peut pas écrire à distance dans le registre d'état d'événements standards, sauf pour l'effacer.

11.5.1.2.4 Effacement

Le registre d'état d'événements standards devra seulement être effacé par:

1) une instruction *CLS;

2) une mise sous tension si l'indicateur d'effacement d'état à la mise sous tension est mis VRAI (voir nota);

3) une interrogation *ESR?

NOTE – Un **appareil** subissant une séquence de mise sous tension devra au début effacer le registre d'état d'événements standards, mais ensuite enregistrer tous les événements qui suivront pendant la séquence de mise sous tension de l'appareil, y compris le positionnement du bit d'événement PON (voir 11.5.1.1.2).

Le registre d'état d'événements standards ne devra être effacé par aucune autre condition de l'**appareil**.

11.5.1.3 Opération de registre de validation d'état d'événements standards

11.5.1.3.1 Fonction

Le registre de validation d'état d'événements standards permet à un ou plusieurs événements dans le registre d'état d'événements standards d'être reflétés dans le bit de message récapitulatif ESB. Ce registre suit les règles d'opération du modèle de registre de structure de données d'état pour les registres de validation d'événements (voir 11.4.2.3).

Ce registre est défini sur 8 bits, chacun correspondant aux bits du registre d'état d'événements standard. Les bits 8 à 15 sont réservés par la CEI pour une utilisation future.

11.5.1.3.2 Lecture

Le registre de validation d'état d'événements standards est lu avec l'interrogation commune, *ESE?. La donnée est renvoyée comme une <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> à poids binaire (voir 10.11).

11.5.1.3.3 Ecriture

On écrit dans le registre de validation d'état d'événements standards avec l'instruction commune, *ESE. La donnée est codée comme une <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> (voir 10.10).

11.5.1.3.4 Effacement

Le registre de validation d'état d'événements standards devra être effacé par les opérations suivantes:

1) L'envoi de *ESE avec une valeur de donnée égale à zéro.

2) Un événement de mise sous tension si l'indicateur d'effacement d'état à la mise sous tension est VRAI ou si l'instruction *PSC n'est pas implémentée. Voir instruction *PSC, 10.25 pour de plus amples détails.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.5.1.2.3 Writing

The Standard Event Status Register cannot be written remotely except to clear it.

11.5.1.2.4 Clearing

The Standard Event Status Register shall only be cleared by:

1) a *CLS command;

2) a power-on if the power-on-status-clear flag is set TRUE (see note);

3) a *ESR? query.

NOTE - A **device** undergoing a power-on sequence shall initially clear the Standard Event Status Register but then record any subsequent events during the device's power-on sequence including setting the PON event bit (see 11.5.1.1.2).

The Standard Event Status Register shall not be cleared by any other **device** condition.

11.5.1.3 Standard Event Status Enable Register operation

11.5.1.3.1 Function

The Standard Event Status Enable Register allows one or more events in the Standard Event Status Register to be reflected in the ESB summary-message bit. This register follows the rules of operation of the Status Data Structure register model for Event Enable Registers (11.4.2.3).

This register is defined for 8 bits, each corresponding to the bits in the Standard Event Status Register. Bits 8 through 15 are reserved by the IEC for future use.

11.5.1.3.2 Reading

The Standard Event Status Enable Register is read with the common query, *ESE?. Data is returned as a binary-weighted <NR1 NUMERIC RESPONSE DATA> (10.11).

11.5.1.3.3 Writing

The Standard Event Status Enable Register is written to by the common command, *ESE. Data is encoded as <DECIMAL NUMERIC PROGRAM DATA> (see 10.10).

11.5.1.3.4 Clearing

The Standard Event Status Enable Register shall be cleared by the following:

1) Sending *ESE with a data value of zero.

2) A power-on event if the power-on-status-clear flag is TRUE or the *PSC command is not implemented. See *PSC command, 10.25 for further details.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

Aucune autre condition de l'**appareil** ou d'interface, y compris la réception du message **dcas**, ne doit modifier ce registre.

Le registre de validation d'état d'événements standards ne devra spécifiquement pas être affecté par les opérations suivantes:

1) Un changement d'état de fonction d'effacement d'appareil CEI 625-1.
2) L'envoi de l'instruction commune *RST.

11.5.2 *Modèle de file d'attente standard*

11.5.2.1 Fonction

La file d'attente de sortie stocke les messages de réponse jusqu'à ce qu'ils soient lus. La disponibilité de la sortie est résumée par le message récapitulatif MAV (message disponible) (bit 4 de l'octet d'état). Voir figure 11-10.

Le message récapitulatif MAV est utilisé pour synchroniser l'échange d'informations avec le **contrôleur**. Voir l'article 12 pour la discussion sur la synchronisation de l'**appareil** et l'article 6 pour les détails sur l'opération de la file d'attente de sortie.

11.5.2.2 Lecture

La file d'attente de sortie doit être une file d'attente premier entré, premier sorti (FIFO). La file d'attente est lue en établissant des liaisons d'octets à l'extérieur pendant que l'**appareil** est en TACS (état parleur actif).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

No other **device** or interface conditions, including receipt of the **dcas** message, shall alter this register.

The Standard Event Status Enable Register shall specifically not be affected by the following:

1) An IEC 625-1 Device Clear function state change.
2) Sending the *RST common command.

11.5.2 *Standard Queue model*

11.5.2.1 Function

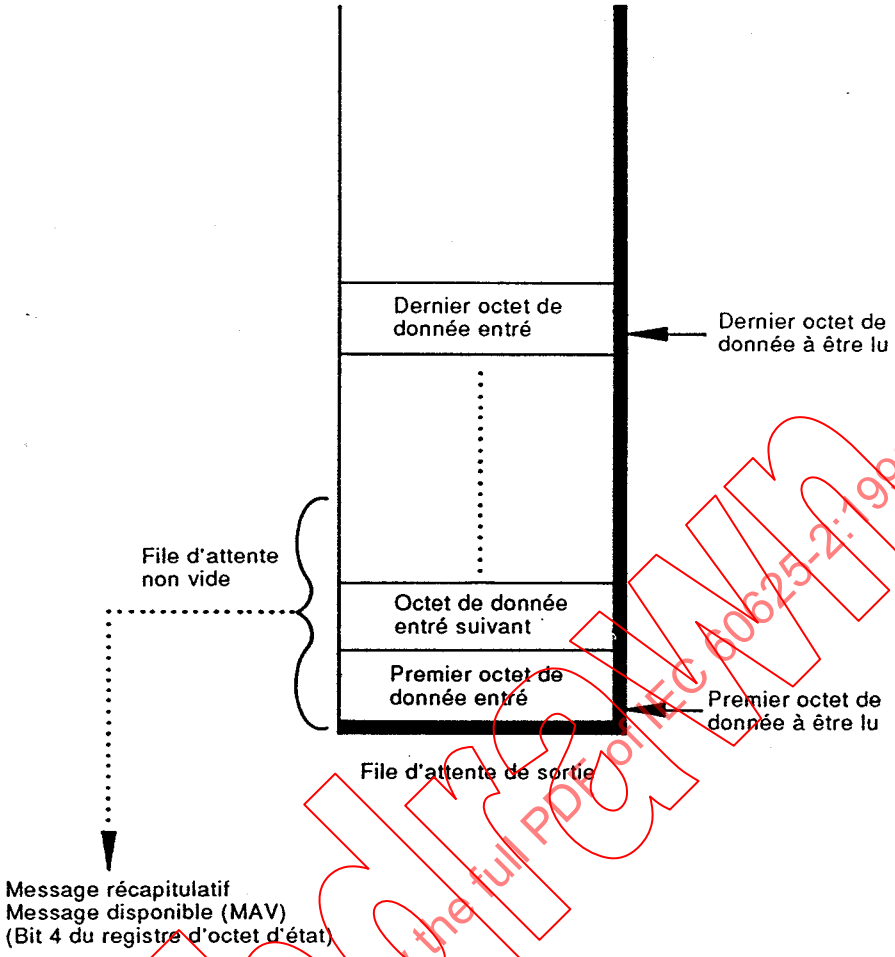
The Output Queue stores response messages until they are read. The availability of output is summarized by the MAV (message available) summary message (bit 4 of the status byte). See figure 11-10.

The MAV summary message is used to synchronize information exchange with the **controller**. See clause 12 for a discussion of **device** synchronization and clause 6 for details on the operation of the Output Queue.

11.5.2.2 Reading

The Output Queue shall be a first-in, first-out (FIFO) Queue. The Queue is read by handshaking bytes out while the **device** is in TACS (Talker Active State).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993



NOTE – MAV devient VRAI chaque fois que des données sont prêtes et pas obligatoirement dans la file d'attente de sortie, 6.4.5.3 et 6.1.10.2.1.

Figure 11-10 – Modèle de file d'attente standard

11.5.2.3 Ecriture

La file d'attente de sortie ne se relie à l'**interface du système** que par les protocoles définis à l'article 6.

11.5.2.4 Effacement

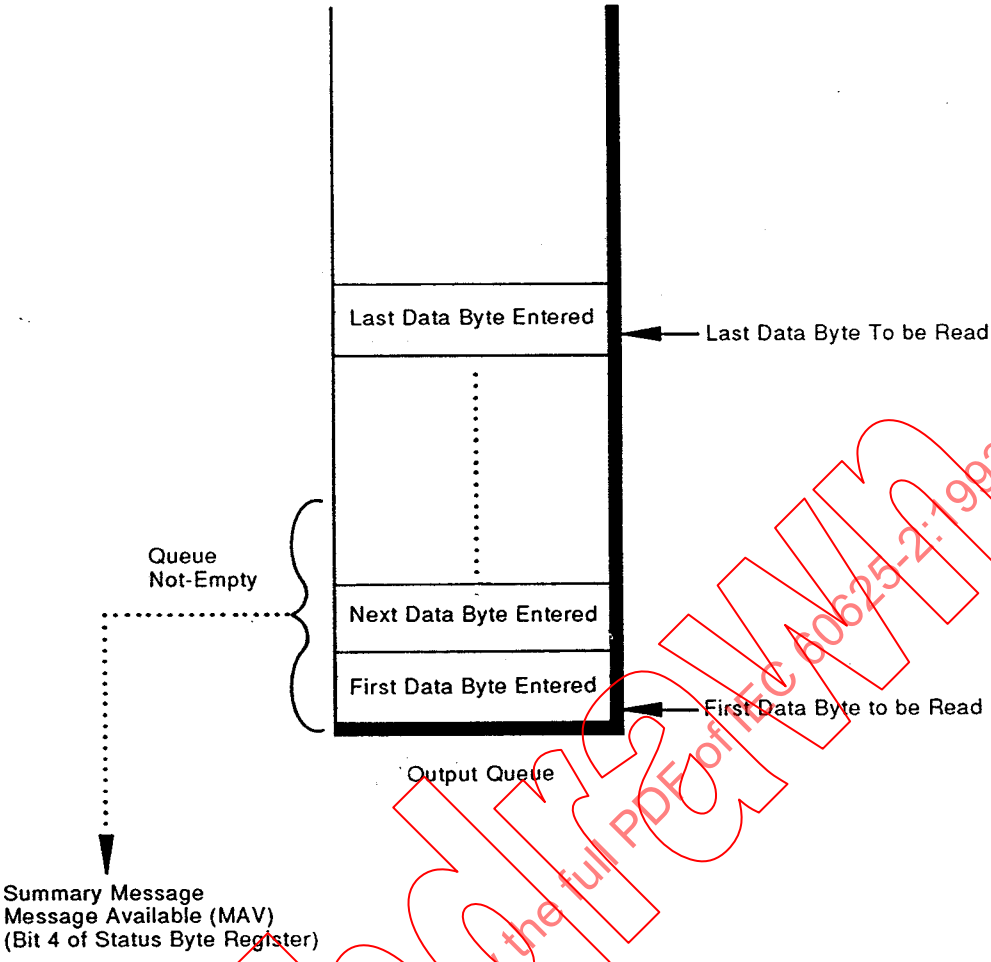
La file d'attente de sortie devra être effacée en accord avec les protocoles de l'article 6.

L'instruction *CLS spécifiquement ne devra PAS effacer la file d'attente de sortie sauf comme cela est exigé en 6.3.2.3 pour tout <MESSAGE DE PROGRAMMATION>.

11.6 *Traitement des réponses à des interrogations parallèles*

Cette section décrit une méthode pour générer et contrôler le message local ist (état individuel) CEI 625-1. Ce message est le bit d'état envoyé durant une opération d'interrogation parallèle. Si un **appareil** possède la capacité (complète) PP1, l'**appareil** devra implémenter la structure de données et les instructions communes associées de cet article (voir 11.6).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



NOTE - MAV becomes TRUE whenever data is ready and not necessarily in the Output Queue, 6.4.5.3 and 6.1.10.2.1.

Figure 11-10 – Standard Queue model

11.5.2.3 Writing

The Output Queue interfaces to the **system interface** only through the protocols defined in clause 6.

11.5.2.4 Clearing

The Output Queue shall be cleared in accordance with the protocols of clause 6.

The *CLS command specifically shall NOT clear the Output Queue except as required in 6.3.2.3 for any <PROGRAM MESSAGE>.

11.6 *Parallel Poll response handling*

This section describes a method for generating and controlling the IEC 625-1 ist (individual status) local message. This message is the status bit sent during a parallel poll operation. If a **device** has PP1 (complete) capability, the **device** shall implement the data structure and associated common commands of this clause (see 11.6).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

11.6.1 *Registre de validation d'interrogation parallèle*

11.6.1.1 Fonction

Le registre de validation d'interrogation parallèle représenté à la figure 11-11 est un registre large de huit à seize bits. Chaque bit de ce registre correspond à un bit de l'octet d'état ou à une condition définie par l'appareil. Tous les bits dans l'octet d'état doivent être utilisés. On peut associer jusqu'à huit conditions supplémentaires dans l'**appareil** avec les huit bits supérieurs du registre de validation d'interrogation parallèle.

NOTE – Certains **contrôleurs** peuvent avoir des difficultés à lire un nombre entier sans signe de 16 bits dans leur format interne.

Chaque bit du registre de validation d'interrogation parallèle fait un ET avec sa condition correspondante ou son bit récapitulatif. Les bits résultants font un OU pour générer ist. L'utilisation du registre de validation d'interrogation parallèle permet à n'importe quel bit unique ou combinaison de bits de contrôler ist.

Le message MSS (état récapitulatif principal) est utilisé à la place de RQS pour signaler le bit 6 du registre d'octet d'état. MSS est le message renvoyé avec l'octet d'état en réponse à une interrogation commune *STB? (voir 11.2.2.2).

11.6.1.2 Lecture du registre de validation d'interrogation parallèle

Le contenu du registre de validation d'interrogation parallèle peut être interrogé par l'envoi de *PRE?. L'**appareil** répond avec un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1> (voir 10.24).

11.6.1.3 Ecriture dans le registre de validation d'interrogation parallèle

On écrit dans le registre de validation d'interrogation parallèle en envoyant l'instruction *PRE suivie d'un nombre sous forme de <DONNÉE DE PROGRAMMATION NUMÉRIQUE DÉCIMALE> à l'appareil. Le nombre est converti en binaire et placé dans le registre (voir 10.23).

11.6.1.4 Effacement du registre de validation d'interrogation parallèle

Le registre de validation d'interrogation parallèle est effacé en:

1) mettant la valeur du registre à zéro;

2) réalisant une mise sous tension, si l'indicateur d'effacement d'état à la mise sous tension est VRAI ou si *PSC n'est pas implémenté.

11.6.2 *Lecture de ist sans interrogation parallèle*

L'état de ist peut être interrogé en envoyant l'interrogation commune *IST?. L'**appareil** devra renvoyer soit un "0", si ist est FAUX, soit un "1", si ist est VRAI. Le nombre renvoyé devra être un élément de <DONNÉE DE RÉPONSE NUMÉRIQUE NR1>. (Voir 10.15).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

11.6.1 *Parallel Poll Enable Register*

11.6.1.1 Function

The Parallel Poll Enable Register shown in figure 11-11 is an 8- to 16-bit wide register. Each bit in this register corresponds to a bit in the status byte or a device-defined condition. All the bits in the status byte must be used. Up to eight more conditions in the **device** may be associated with the upper eight bits of the Parallel Poll Enable Register.

NOTE - Some **controllers** may have difficulty reading a 16-bit unsigned integer into their internal format.

Each bit in the Parallel Poll Enable Register is ANDed with its corresponding condition or summary bit. The resulting bits are ORed together to generate ist. Using the Parallel Poll Enable Register allows any single bit or combination of bits to control ist.

The MSS (Master Summary Status) message is used in place of RQS to report bit 6 of the Status Byte Register. MSS is the message returned with the status byte in response to an *STB? common query (see 11.2.2.2).

11.6.1.2 Reading the Parallel Poll Enable Register

The contents of the Parallel Poll Enable Register can be queried by sending *PRE?. The **device** responds with an <NR1 NUMERIC RESPONSE DATA> element (see 10.24).

11.6.1.3 Writing the Parallel Poll Enable Register

The Parallel Poll Enable Register is written to by sending the *PRE command followed by a number in <DECIMAL NUMERIC PROGRAM DATA> format to the device. The number is converted to binary and placed in the register (see 10.23).

11.6.1.4 Clearing the Parallel Poll Enable Register

The Parallel Poll Enable Register is cleared by:

1) setting the value of the register to zero;

2) performing a power-on, if the power-on-status-clear flag is TRUE or if *PSC is not implemented.

11.6.2 *Reading ist without a Parallel Poll*

The state of ist can be queried by sending the *IST? common query. The **device** shall return either a "0", if ist is FALSE, or a "1", if ist is TRUE. The number returned shall be an <NR1 NUMERIC RESPONSE DATA> element (see 10.15).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

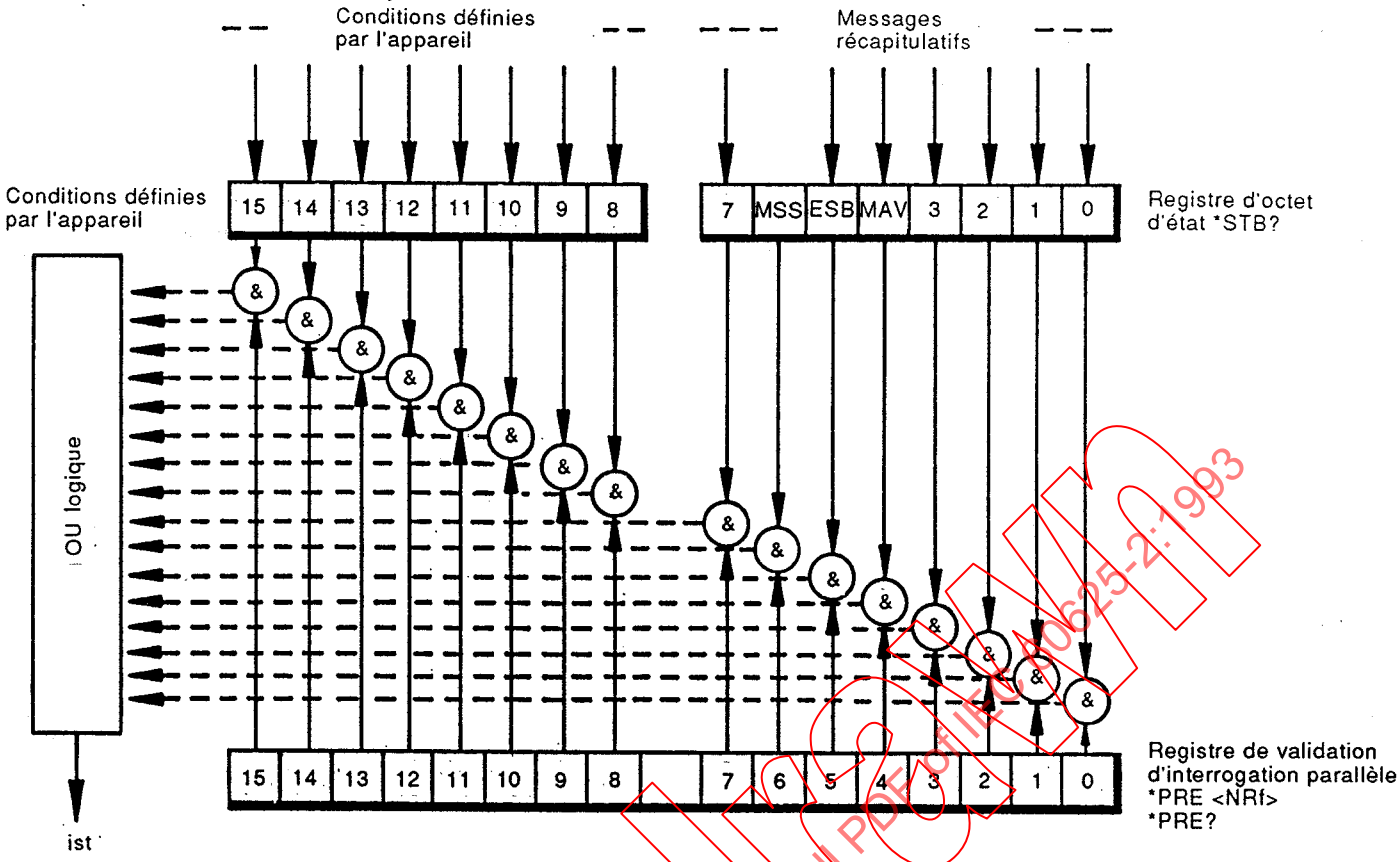


Figure 11-11 – Structure de données de traitement de réponse à une interrogation parallèle

SECTION 3: SPÉCIFICATIONS RELATIVES AU SYSTÈME

12 Techniques de synchronisation appareil/contrôleur

12.1 *Vue d'ensemble*

Cet article décrit les techniques qui peuvent être utilisées pour assurer la synchronisation entre un **appareil** et un **contrôleur**. Trois techniques de base sont indiquées:

- Forçage d'exécution séquentielle
- Attente de réponse dans la file d'attente de sortie de l'**appareil**
- Attente de demande de service

Des exemples détaillés de ces techniques apparaissent dans l'annexe B.

Le problème suivant peut se poser avec les instructions qui mettent un temps appréciable à prendre fin: le programme d'application a besoin de savoir quand elles sont terminées. Prenons le cas d'un **appareil** en cours d'essai connecté à une alimentation programmable et à un voltmètre numérique programmable (DVM). Le programme d'application a besoin de savoir que la sortie de l'alimentation a atteint la tension désirée avant qu'il n'ordonne au DVM de faire une mesure. Cette confirmation que la sortie est valide est une illustration de ce qu'on entend par synchronisation appareil/contrôleur.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

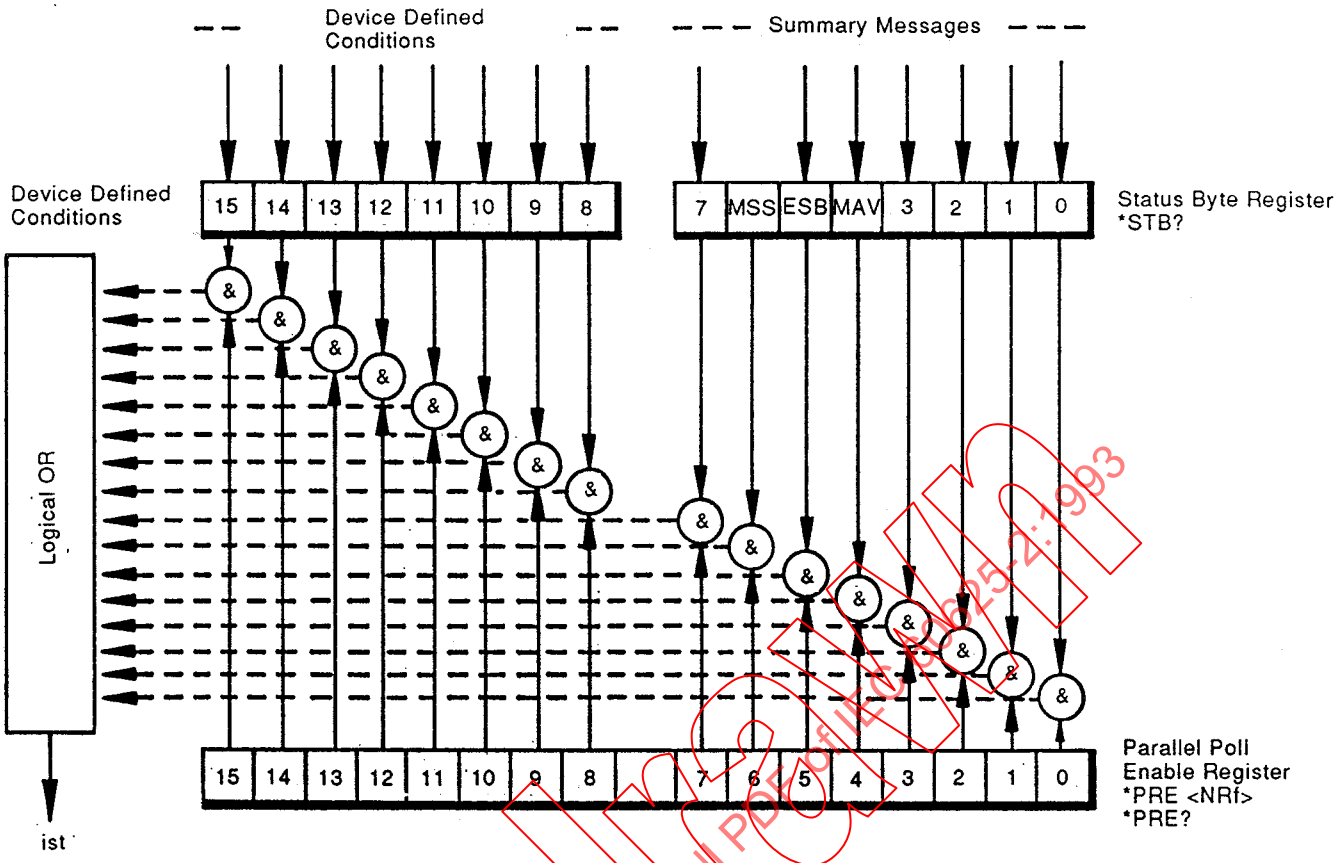


Figure 11-11 – Parallel Poll response handling data structure

SECTION 3: SYSTEM RELATED SPECIFICATIONS

12 Device/controller synchronization techniques

12.1 *Overview*

This clause describes techniques which may be used to insure synchronization between a **device** and a **controller**. Three basic techniques are shown:

- Force sequential execution
- Wait for response in **device's** Output Queue
- Wait for a Service Request

Detailed examples of these techniques appear in annex B.

One potential problem with commands that take appreciable time to finish is that the application program needs to know when they have finished. Consider the case of a unit under test connected to a bus-controllable power supply and a bus-controllable digital voltmeter (DVM). The application program needs to know that the power supply output has reached the desired voltage setting before it commands the DVM to take a measurement. This confirmation that the output is valid is an illustration of what is meant by device/controller synchronization.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

12.2 *Instructions séquentielles et recouvrables*

Les instructions d'**appareil** se classent en deux grandes catégories. La première est la catégorie des "Instructions séquentielles". Une instruction séquentielle est une instruction qui finit toujours avant que l'instruction suivante ne soit exécutée.

L'autre catégorie d'instructions est celle des "Instructions recouvrables". Une instruction recouvrable est une instruction qui autorise l'exécution d'instructions subséquentes alors que les opérations de l'**appareil** lancées par cette instruction recouvrable sont encore en cours.

Le concepteur de l'appareil choisit si une instruction spécifique à l'appareil est séquentielle ou recouvrable. Se référer à 12.8.1 pour de plus amples informations. Le concepteur peut également choisir d'implémenter des instructions spécifiques à l'appareil qui changent une instruction séquentielle en instruction recouvrable et vice versa.

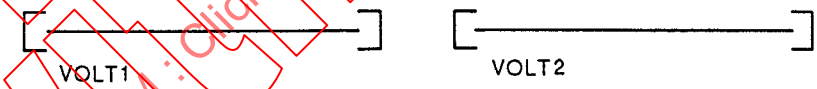
La documentation de l'appareil spécifiera pour chaque instruction s'il s'agit d'une instruction recouvrable ou d'une instruction séquentielle.

12.2.1 *Illustration des instructions séquentielles*

Comme illustration, prenons le cas d'une alimentation programmable avec deux sorties. L'instruction VOLT1 NRf détermine la tension d'une sortie et VOLT2 NRf détermine la tension de l'autre. Si les instructions VOLT sont des instructions séquentielles, on a alors le <MESSAGE DE PROGRAMMATION>

VOLT1 1.23; VOLT2 1.00 <PMT>

(où <PMT> représente le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>) qui réglerait la première sortie à 1,23 V et, une fois que ceci a été fait, réglerait la seconde sortie à 1,00 V. Sur un diagramme des temps, les instructions apparaissent comme suit:



Le symbole "[" représente le commencement des opérations associées de l'**appareil** et "]" représente l'achèvement de ces opérations. Noter que l'achèvement des opérations de l'**appareil** n'est pas le même que l'achèvement de l'exécution de l'instruction; voir 6.1.7 pour la différence.

12.2.2 *Illustration des instructions recouvrables*

Si les instructions VOLT sont des instructions recouvrables, l'instruction VOLT2 commencerait l'exécution avant que les opérations de l'appareil lancées par l'instruction VOLT1 soient terminées. Le diagramme des temps est le suivant:

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

12.2 *Sequential and Overlapped Commands*

Device commands fall into two broad classes. The first is the class of "Sequential Commands". A Sequential Command is a command that always finishes before the next command is executed.

The other class of commands is "Overlapped Commands". An Overlapped Command is a command that allows execution of subsequent commands while the **device** operations initiated by that Overlapped Command are still in progress.

The device designer chooses whether a device-specific command is Sequential or Overlapped. Refer to 12.8.1 for further information. The designer may also choose to implement device-specific commands which change a command's type from Sequential to Overlapped and vice versa.

Device documentation shall specify for each command whether it is an Overlapped or Sequential Command.

12.2.1 *Illustration of Sequential Commands*

As an illustration, consider the case of a bus-controllable power supply with two outputs. The command VOLT1 NRf sets the voltage of one output and VOLT2 NRf sets the voltage of the other. If the VOLT commands are Sequential Commands, then the <PROGRAM MESSAGE>

VOLT1 1.23; VOLT2 1.00 <PMT>

(where <PMT> stands for the <PROGRAM MESSAGE TERMINATOR>) would set the first output to 1,23 V, and then after that was done, set the second output to 1,00 V. In a timeline, the commands appear as

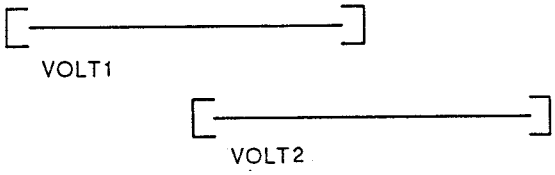


The symbol "[" stands for initiation of the associated **device** operations and "]" stands for the completion of those operations. Note that completion of **device** operations is not the same as completion of execution of the command; see 6.1.7 for the distinction.

12.2.2 *Illustration of Overlapped Commands*

If the VOLT commands are Overlapped Commands, the VOLT2 command would begin execution before the device operations initiated by the VOLT1 command have completed. The timeline is:

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



Il faut de la méthode pour montrer que toutes les instructions sont finies. Les instructions séquentielles sont faciles car l'instruction suivante n'est pas exécutée tant que l'instruction séquentielle précédente n'est pas finie. Les instructions recouvrables sont plus difficiles parce que des opérations de l'appareil, qui ont été lancées plusieurs instructions en arrière, peuvent ne pas être encore finies.

12.3 *Signalisation "opération en cours"*

Le but d'une signalisation "opération en cours" est d'informer le programme d'application que toutes les opérations commencées par une instruction recouvrable sont achevées et que leur résultat est valide ou que les opérations ont été suspendues.

Chaque instruction recouvrable possède une signalisation "opération en cours". Une signalisation "opération en cours" est mise VRAI lorsque l'instruction correspondante est envoyée par le bloc de contrôle d'exécution (voir 6.1.7) au bloc de fonctions de l'appareil (voir 6.1.8). La signalisation est mise FAUX lorsque toutes les opérations de l'**appareil** lancées par cette instruction ont été achevées ou suspendues. (Par exemple, par l'exécution de l'instruction *RST).

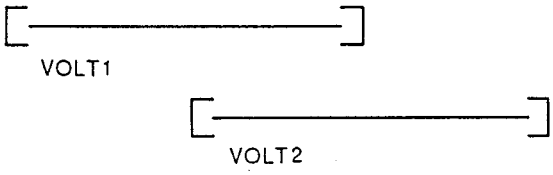
Par exemple, l'opération de changement de la tension de sortie d'une source d'alimentation n'est achevée que lorsqu'on sait que la sortie a atteint la tension programmée dans la tolérance spécifiée par le fabricant. L'opération de changement du calibre d'un DVM est achevée seulement quand le DVM est prêt à effectuer une mesure sur ce calibre. La figure 12-1 décrit le diagramme d'état pour ces signalisations.

12.4 *Signalisation "pas d'opération en cours"*

La signalisation "pas d'opération en cours" est utilisée pour indiquer si l'**appareil** a terminé les instructions recouvrables sélectionnées. Le concepteur de l'appareil peut choisir de faire signaler par la signalisation "pas d'opération en cours" l'état de la totalité ou d'un sous-ensemble de toutes les instructions recouvrables. Le concepteur de l'appareil peut prévoir des instructions spécifiques à l'appareil pour sélectionner quelles sont les instructions recouvrables dont l'état sera signalé par la signalisation "pas d'opération en cours".

Les signalisations "opération en cours" associées aux instructions recouvrables sélectionnées feront un OU et le résultat sera inversé pour générer la signalisation "pas d'opération en cours". La signalisation "pas d'opération en cours" est VRAIE si l'**appareil** n'a pas d'instructions recouvrables sélectionnées.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



Some method is needed to show that all commands have finished. Sequential Commands are easy because the next command is not executed until the previous Sequential Command is finished. Overlapped Commands are more difficult because device operations that were initiated several commands ago may still not be finished.

12.3 *Pending-Operation flag*

The purpose of a Pending-Operation flag is to inform the application program that all operations started by an Overlapped Command are completed and their result is valid or that the operations were aborted.

Each Overlapped Command shall have a corresponding Pending-Operation flag. A Pending-Operation flag is set TRUE when the corresponding command is sent by the Execution Control block (6.1.7) to the Device Function block (6.1.8). The flag is set FALSE when all the **device** operations initiated by that command have been completed or aborted (e.g., by execution of the *RST command).

For instance, the operation of changing the output voltage of a power supply is complete only when the output is known to have reached the programmed voltage within manufacturer specified tolerance. The operation of changing the range on a DVM is complete only when the DVM is ready to make a measurement in that range. Figure 12-1 describes control of these flags.

12.4 *No-Operation-Pending flag*

The No-Operation-Pending flag is used to indicate whether the **device** has completed operations initiated by selected Overlapped Commands. The device designer may choose to have the No-Operation-Pending flag report the status of all or a subset of all Overlapped Commands. The device designer may provide device-specific commands to select which Overlapped Commands shall have their status reported by the No-Operation-Pending flag.

The Pending-Operation flags associated with the selected Overlapped Commands shall be OR'd together and the result inverted to generate the No-Operation-Pending flag. The No-Operation-Pending flag is TRUE if the **device** has no selected Overlapped Commands.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

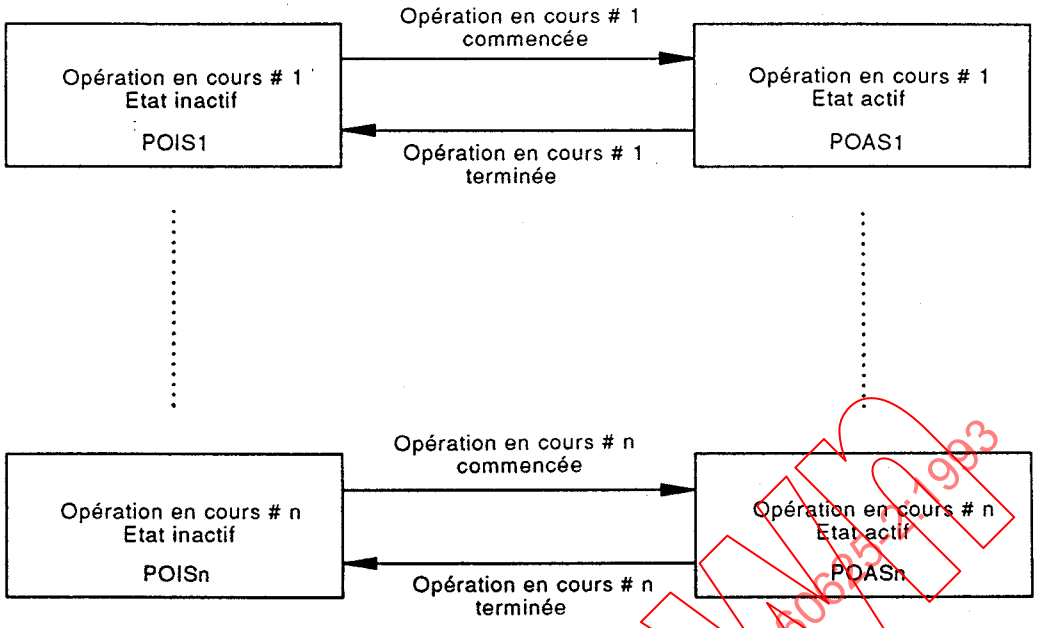


Figure 12-1 – Schéma d'état d'opération en cours des instructions recouvrables sélectionnées

12.5 *Instructions de synchronisation contrôleur/appareil*

Les instructions *OPC et *WAI et l'interrogation *OPC? permettent au programmeur d'application de maintenir la synchronisation contrôleur/appareil.

12.5.1 *L'instruction commune *WAI*

L'instruction commune *WAI empêche l'exécution par l'**appareil** de toute instruction ou interrogation suivante jusqu'à ce que la signalisation "pas d'opération en cours" soit VRAIE. Aucune spécification spéciale n'est faite pour les instructions séquentielles parce que toute instruction séquentielle précédente achèvera son exécution avant que l'analyseur syntaxique ne traite la prochaine instruction (y compris l'instruction *WAI) du bloc de contrôle d'exécution.

12.5.1.1 Un échantillon de message de programmation sans l'instruction *WAI

Une illustration de l'utilisation de l'instruction *WAI est la suivante: considérons un **appareil** à enregistrement de données ayant reçu l'instruction de faire une mesure avec l'instruction dépendant de l'appareil START et de lire le temps actuel avec TIME?. L'instruction START est une instruction recouvrable et prend un temps appréciable pour se réaliser. Le <MESSAGE DE PROGRAMMATION TERMINÉ>

START; TIME? <PMT>

fait faire une mesure à l'**appareil** et le fait revenir au moment où la mesure a été commencée. Le diagramme de temps et les signalisations sont représentés à la figure 12-3.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

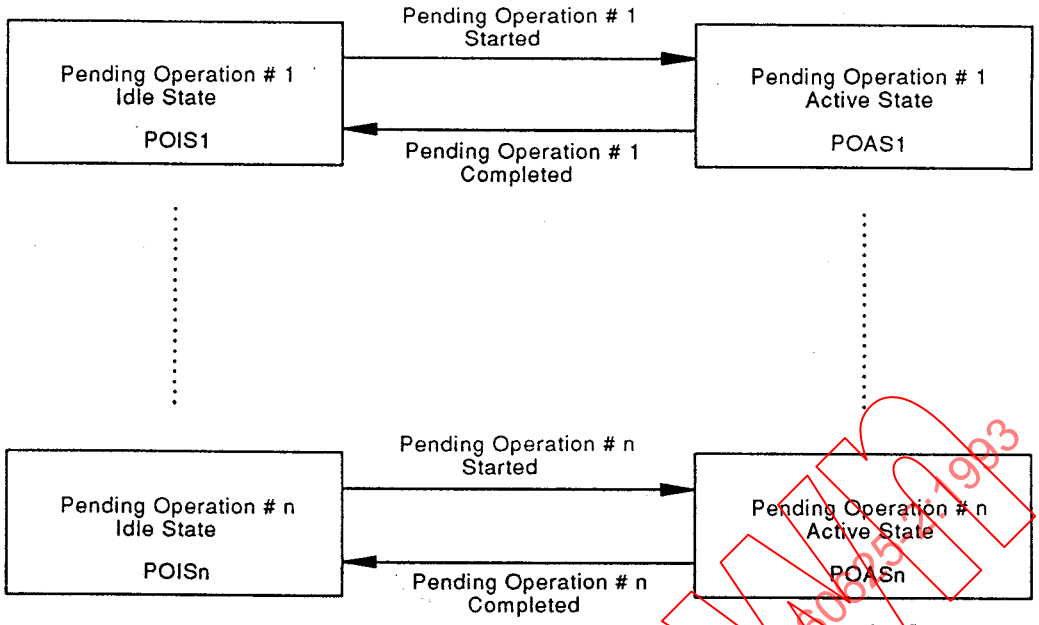


Figure 12-1 – Pending operation state diagram of selected overlapped commands

12.5 *Controller/device synchronization commands*

The *OPC and *WAI commands and the *OPC? query allow the application programmer to maintain controller/device synchronization.

12.5.1 *The *WAI common command*

The *WAI command shall prevent the **device** from executing any further commands or queries until the No-Operation-Pending flag is TRUE. No special provision is made for Sequential Commands because any preceding Sequential Command will complete execution before the Parser hands the next command (including the *WAI command) to the Execution Control block.

12.5.1.1 A sample program message without the *WAI command

An illustration of the use of the *WAI command is the following: Consider a data-logging **device** which is commanded to take a measurement with the device-specific command START and to read the present time by TIME?. The START command is an Overlapped Command and takes appreciable time to perform. The <TERMINATED PROGRAM MESSAGE>

START; TIME? <PMT>

causes the **device** to take a measurement and return the time at which the measurement was begun. The timeline and flags are shown in figure 12-3.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

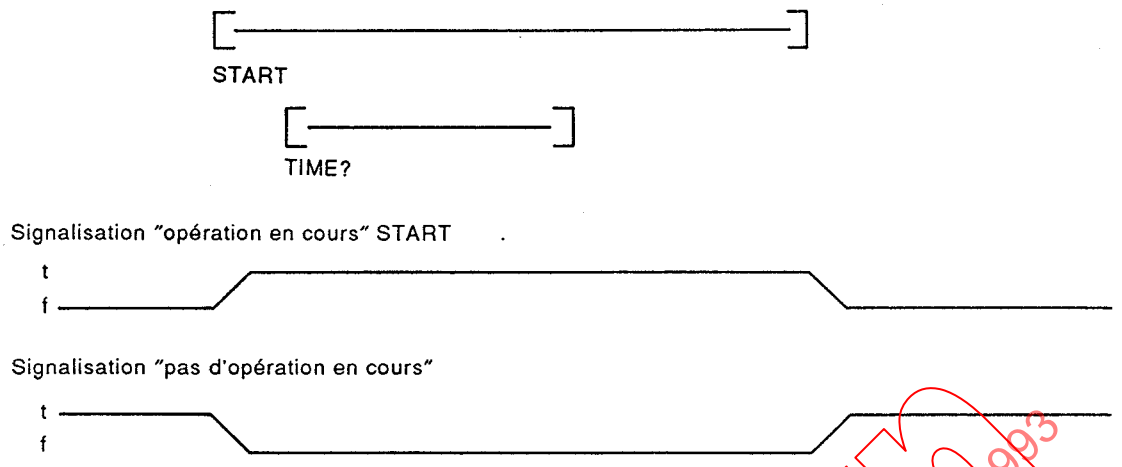


Figure 12-2 – Diagramme de temps – Echantillon de programmation n'utilisant pas l'instruction *WAI

12.5.1.2 Un échantillon de message de programmation avec l'instruction *WAI

Le <MESSAGE DE PROGRAMMATION TERMINÉ>

START; *WAI; TIME? <PMT>

oblige la réponse à l'interrogation TIME? à être le moment auquel la mesure a été finie. De cette manière, l'instruction commune *WAI permet au programmeur d'application d'imposer une exécution séquentielle des instructions recouvrables. Le diagramme de temps et les signalisations sont représentés à la figure 12-3.

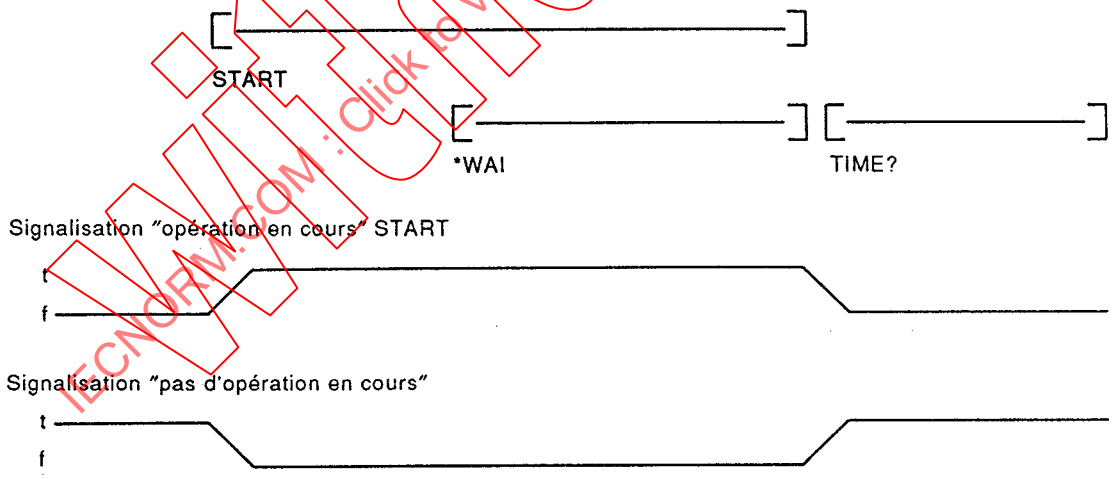


Figure 12-3 – Diagramme de temps – Echantillon de message de programmation utilisant l'instruction *WAI

Des illustrations détaillées de l'utilisation de l'instruction *WAI apparaissent dans l'annexe B.

L'action d'arrêt de l'instruction commune *WAI sera effacée par la mise sous tension et par le message **dcas**. A partir du moment où l'instruction *WAI suspend le contrôle d'exécution de l'**appareil** jusqu'à ce que la signalisation "pas d'opération en cours" soit VRAIE, les instructions *RST et *CLS ne peuvent avoir aucun effet sur l'opération de *WAI.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

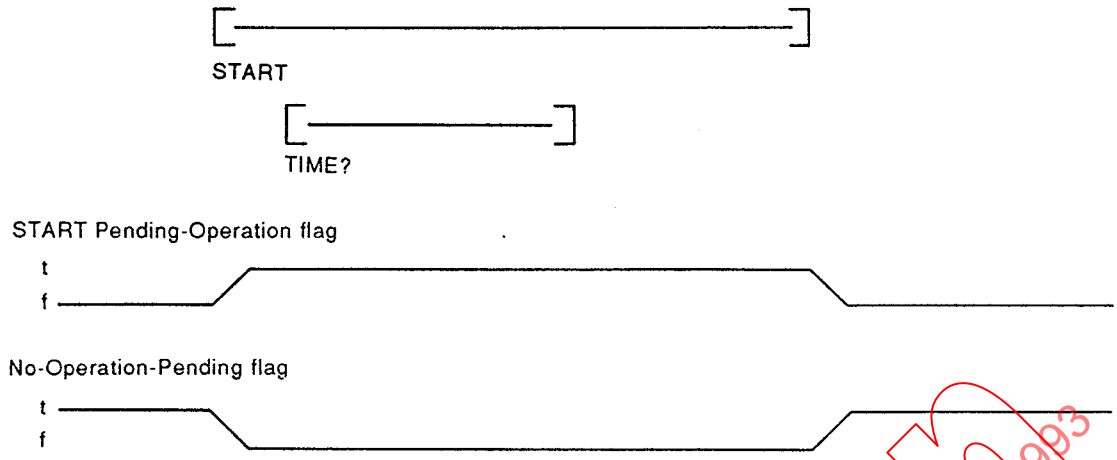


Figure 12-2 – Timing diagram – Sample program message not using *WAI command

12.5.1.2 A sample program message with the *WAI command

The <TERMINATED PROGRAM MESSAGE>

START; *WAI; TIME? <PMT>

forces the TIME? query response to be the time at which the measurement was finished. Thus the *WAI command allows the application programmer to force sequential execution of Overlapped Commands. The timeline and flags are shown in figure 12-3.

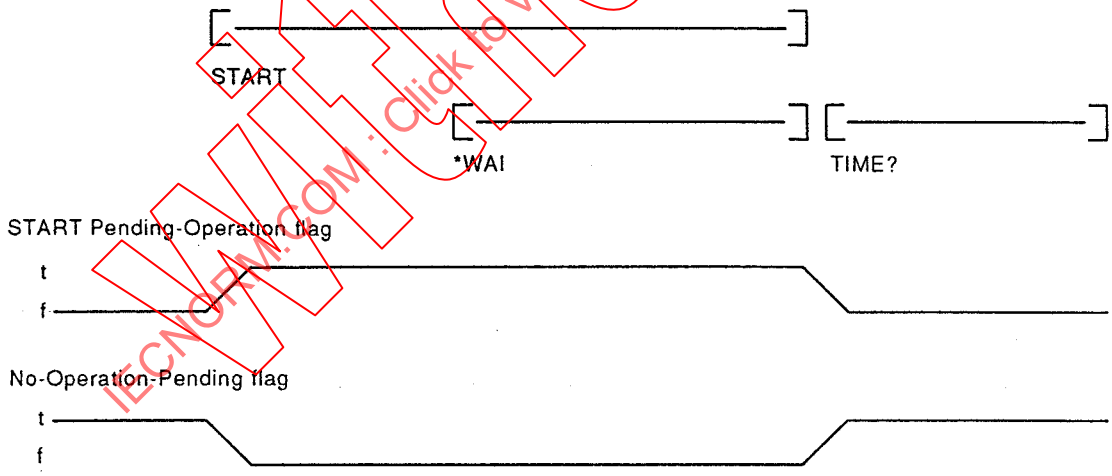


Figure 12-3 – Timing diagram – Sample program message using *WAI command

Detailed illustrations of the use of the *WAI command appear in annex B.

The *WAI command's holdoff action shall be cancelled by power-on and by the **dcas** message. Since the *WAI command holds off the **device** execution control until the No-Operation-Pending flag is TRUE, the commands *RST and *CLS can have no effect on the operation of *WAI.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

12.5.2 *L'instruction commune *OPC*

L'instruction *OPC permet la synchronisation entre un **contrôleur** et plusieurs **appareils**. Le bit d'événement OPC dans l'ESR (Registre d'état d'événements standards) est utilisé pour effectuer la synchronisation. Des illustrations détaillées de l'utilisation de l'instruction *OPC apparaissent en annexe B.

12.5.2.1 Le diagramme d'état d'instruction "opération achevée"

L'instruction *OPC agira comme cela est décrit dans le diagramme d'état donné à la figure 12-4 et les descriptions données tout au long de 12.5.2.

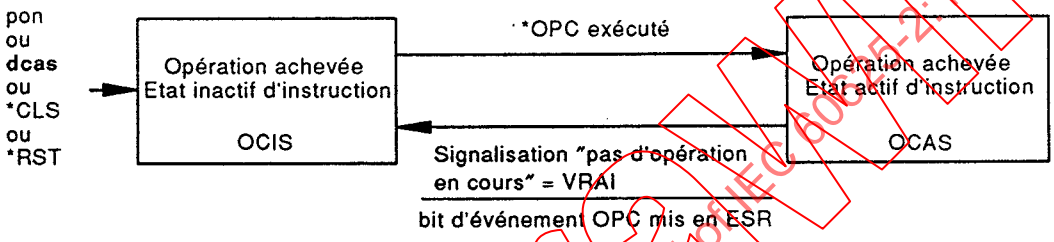


Figure 12-4 – Diagramme d'état d'instruction "opération achevée"

12.5.2.1.1 OCIS (Etat inactif d'instruction "opération achevée")

Les conditions suivantes placent l'**appareil** dans l'état OCIS:

- Mise sous tension (voir 5.12)
- Réception du message **dcas** (voir 5.8)
- *CLS est exécuté (voir 10.3)
- *RST est exécuté (voir 10.32).

L'**appareil** sortira de OCIS et entrera en OCAS (état actif d'instruction "opération achevée") lorsque l'instruction *OPC est exécutée.

12.5.2.1.2 OCAS (Etat actif d'instruction "opération achevée")

En OCAS, l'**appareil** détecte continuellement la signalisation "pas d'opération en cours". Lorsque la signalisation "pas d'opération en cours" est détectée VRAIE, l'**appareil** mettra le bit d'événement OPC dans l'ESR (Registre d'état d'événements standards), sortira de OCAS, et entrera en OCIS (Etat inactif d'instruction "opération achevée").

12.5.2.2 Echantillon d'utilisation de l'instruction *OPC

L'exemple précédent à 12.1 de l'alimentation, DVM, et **appareil** en cours d'essai, illustre l'effet de *OPC. Le problème est de programmer le DVM pour faire une mesure uniquement une fois que l'alimentation a fini de changer sa tension de sortie.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

12.5.2 *The *OPC common command*

The *OPC command allows synchronization between a **controller** and several **devices**. The OPC event-bit in the ESR (Standard Event Status Register) is used to effect the synchronization. Detailed illustrations of the use of the *OPC command appear in annex B.

12.5.2.1 The Operation Complete command state diagram

The *OPC command shall perform as described in figure 12-4 and the descriptions given throughout clause 12.5.2.

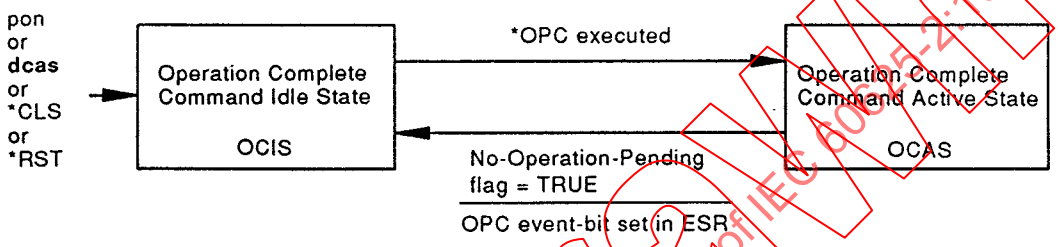


Figure 12-4 – Operation Complete command state diagram

12.5.2.1.1 OCIS (Operation Complete Command Idle State)

The following conditions place the **device** into OCIS:

- Power-on (see 5.12)
- Receipt of the **dcas** message (see 5.8)
- *CLS is executed (see 10.3)
- *RST is executed (see 10.32)

The **device** shall exit OCIS and enter OCAS (Operation Complete Command Active State) when the *OPC command is executed.

12.5.2.1.2 OCAS (Operation Complete Command Active State)

In OCAS the **device** continuously senses the No-Operation-Pending flag. When the No-Operation-Pending flag is sensed TRUE the **device** shall set the OPC event-bit in the ESR (Standard Event Status Register), exit OCAS, and enter OCIS (Operation Complete Command Idle State).

12.5.2.2 Sample use of the *OPC command

The previous example in 12.1 of the power supply, DVM, and **device** under test illustrates the effect of *OPC. The problem is to program the DVM to take a measurement only after the power supply has finished changing its output voltage.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Le programmeur d'application peut valider le bit 0 (correspondant au bit d'événement OPC) dans le registre de validation d'état d'événement standard et aussi le bit 5 (correspondant au message récapitulatif ESB) dans le registre de validation de demande de service, de sorte que l'appareil demandera un service chaque fois que le bit d'événement OPC devient VRAI. Un <MESSAGE DE PROGRAMMATION TERMINÉ> de

*CLS; VOLT 5.20; *OPC <PMT>

donne alors l'instruction à l'alimentation d'effacer son registre d'état d'événement standard et son octet d'état, de mettre sa tension de sortie à 5,20 V, puis de demander un service une fois que la tension de sortie est valide. Une fois que la demande de service a été détectée, le programme d'application peut alors ordonner au DVM de faire une mesure. Le diagramme de temps et les signalisations sont représentés à la figure 12-5.

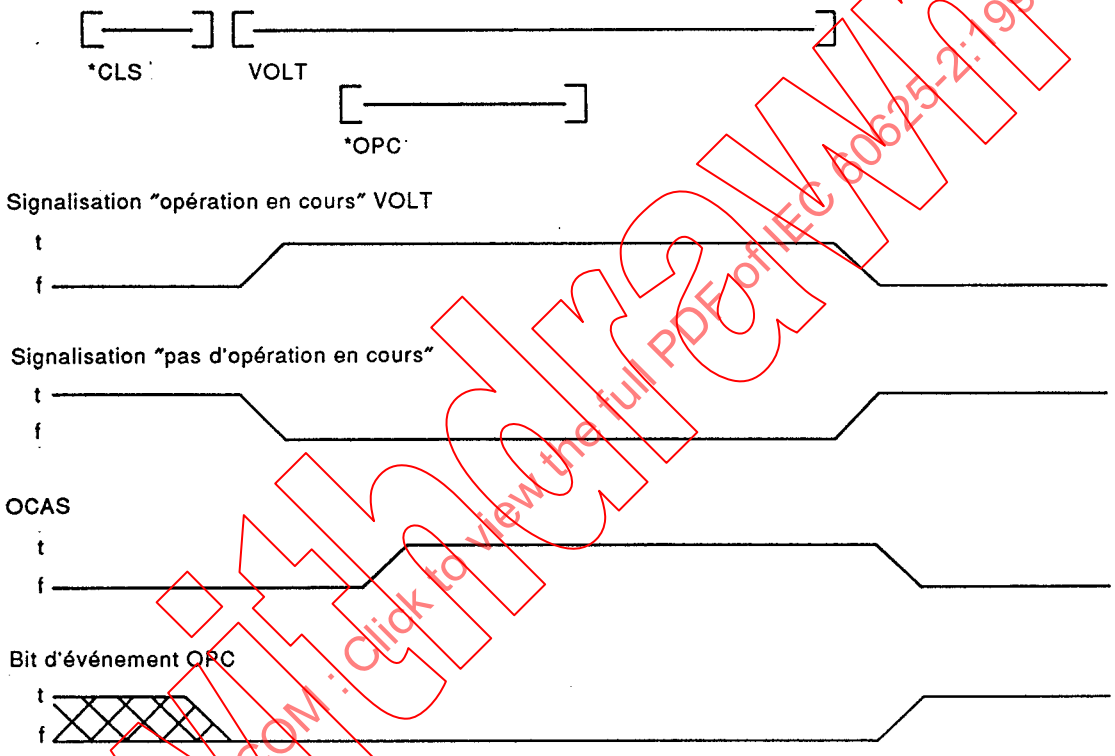


Figure 12-5 – Diagramme de temps – Echantillon de message de programmation utilisant l'instruction *OPC

12.5.3 *L'interrogation commune *OPC?*

L'interrogation *OPC? permet la synchronisation entre un **contrôleur** et un **appareil** en utilisant le bit MAV dans l'octet d'état ou une lecture de la file d'attente de sortie. Des illustrations détaillées de l'utilisation de cette fonction apparaissent en annexe B. Noter que, contrairement à l'instruction *OPC décrite en 12.5.2, l'interrogation *OPC? n'affecte en aucune façon le bit d'événement OPC dans l'ESR (Registre d'état d'événements standards).

12.5.3.1 Le diagramme d'état d'interrogation *OPC?

L'interrogation *OPC? interviendra comme cela est décrit dans le diagramme d'état donné à la figure 12-6 et les descriptions données tout au long de 12.5.3.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The application programmer may enable bit 0 (corresponding to the OPC event-bit) in the Standard Event Status Enable Register and also bit 5 (corresponding to the ESB summary message) in the Service Request Enable Register so that the **device** will request service whenever the OPC event-bit becomes TRUE. A <TERMINATED PROGRAM MESSAGE> of

*CLS; VOLT 5.20; *OPC <PMT>

then commands the power supply to clear its Standard Event Status Register and Status Byte, set its output voltage to 5.20 V, and then request service once the output voltage is valid. After the service request is detected the application program can then command the DVM to take a measurement. The timeline and flags are shown in figure 12-5.

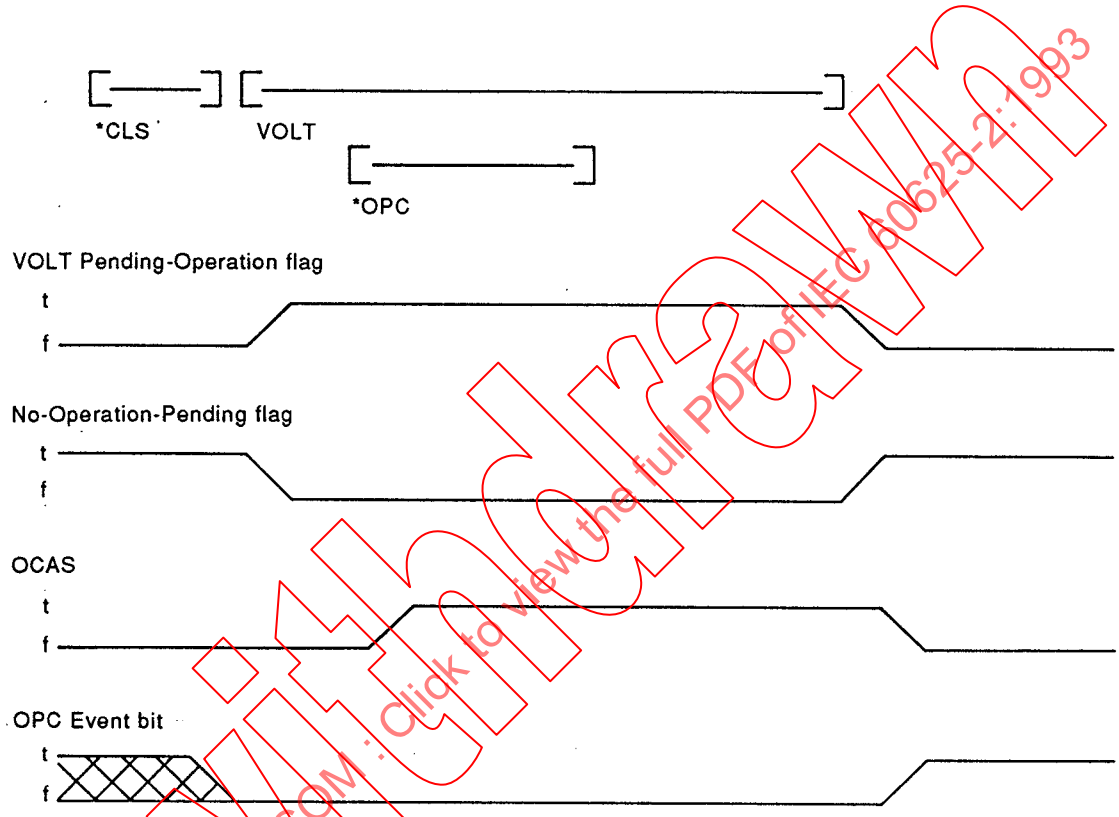


Figure 12-5 – Timing diagram – Sample program message using the *OPC command

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

12.5.3 *The *OPC? common query*

The *OPC? query allows synchronization between a **controller** and a **device** using the MAV bit in the Status Byte or a read of the Output Queue. Detailed illustrations of the use of this function appear in annex B. Note that unlike the *OPC command described in 12.5.2, the *OPC? query does not in any way affect the OPC Event bit in the ESR (Standard Event Status Register).

12.5.3.1 The *OPC? query diagram

The *OPC? query shall perform as described in figure 12-6 and the descriptions given throughout 12.5.3.

12.5.3.1.1 OQIS (Etat inactif de l'interrogation "opération achevée")

Les conditions suivantes placent l'appareil en OQIS:

- Mise sous tension (voir 5.12)
- Réception du message **dcas** (voir 5.8)
- *CLS est exécuté (voir 10.3)
- *RST est exécuté (voir 10.32)

L'**appareil** doit sortir de OQIS et entrer en OQAS (Etat actif d'interrogation "opération achevée") lorsque l'interrogation *OPC? est exécutée.

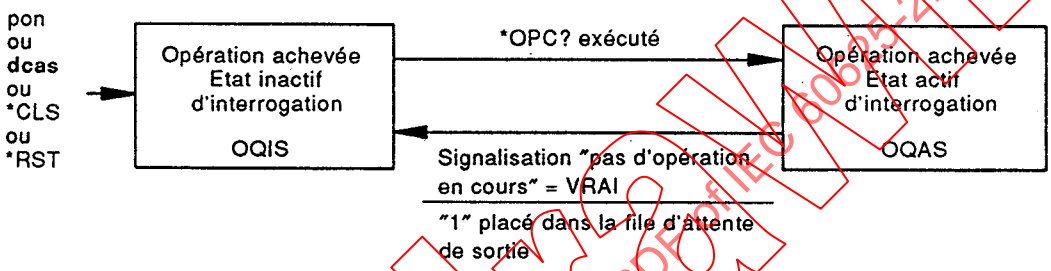


Figure 12-6 – Diagramme d'état d'interrogation "opération achevée"

12.5.3.1.2 OQAS (Etat actif d'interrogation "opération achevée")

En OQAS, l'**appareil** détecte continuellement la signalisation "pas d'opération en cours". Lorsque la signalisation "pas d'opération en cours" est détectée VRAIE, l'**appareil** doit placer un caractère 1 ISO 7 bit dans la file d'attente de sortie. Une conséquence de cette action est que le bit MAV dans l'octet d'état va devenir VRAI. L'**appareil** sortira également de OQAS et entrera en OQIS.

12.5.3.2 Echantillon d'utilisation de l'interrogation *OPC? – Pas de demande de service

La technique suivante d'utilisation de l'interrogation *OPC? évite l'utilisation de demandes de service et d'interrogations série. Le bit 4 (correspondant au message récapitulatif MAV) dans le registre de validation de demande de service est réglé à une valeur de zéro (invalidé). Le <MESSAGE DE PROGRAMMATION TERMINÉ>

VOLT 5.20; *OPC? <PMT>

est alors envoyé à l'alimentation. Puis le programme d'application tente de lire la réponse de l'alimentation à l'interrogation *OPC?. L'**appareil** n'introduira pas de réponse à *OPC? dans la file d'attente de sortie tant que la tension n'est pas correcte. Une fois que la réponse est reçue, le **contrôleur** ordonne au DVM de faire une mesure.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

12.5.3.1.1 OQIS (Operation Complete Query Idle State)

The following conditions place the device in OQIS:

- Power-on (see 5.12)
- receipt of the **dcas** message (see 5.8)
- *CLS is executed (see 10.3)
- *RST is executed (see 10.32)

The **device** shall exit OQIS and enter OQAS (Operation Complete Query Active State) when the *OPC? query is executed.

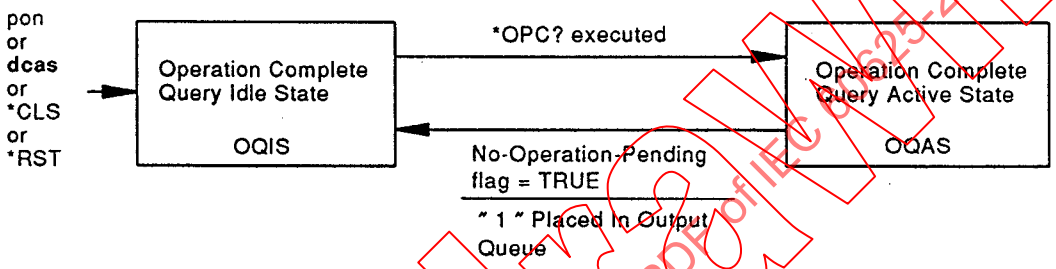


Figure 12-6 – Operation Complete query state diagram

12.5.3.1.2 OQAS (Operation Complete Query Active State)

In OQAS the **device** continuously senses the No-Operation-Pending flag. When the No-Operation-Pending flag is sensed TRUE the **device** shall place an ISO 7-bit character 1 in the Output Queue. A consequence of this action is that the MAV bit in the Status Byte will become TRUE. The **device** shall also exit OQAS and enter OQIS.

12.5.3.2 Sample use of the *OPC? query – No Service Request

The following technique of using the *OPC? query avoids the use of service requests and serial polls. Bit 4 (corresponding to the MAV summary-message) in the Service Request Enable Register is set to a value of zero (disabled). The <TERMINATED PROGRAM MESSAGE>

VOLT 5.20; *OPC? <PMT>

is then sent to the power supply. The application program then attempts to read the *OPC? query response from the power supply. The **device** will not put a response to the *OPC? into the Output Queue until the voltage is valid. After the response is received the **controller** commands the DVM to take a measurement.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

12.5.3.3 Echantillon d'utilisation de l'interrogation *OPC? – Méthode de demande de service

La technique suivante utilise la demande de service et l'interrogation série. Le bit 4 (correspondant au message récapitulatif MAV) dans le registre de validation de demande de service est réglé à une valeur de un (validé) et le <MESSAGE DE PROGRAMMATION TERMINÉ>

VOLT 5.20; *OPC? <PMT>

est envoyé à l'alimentation. L'alimentation demande un service lorsque la tension de sortie est correcte. Le DVM reçoit alors l'ordre de faire une mesure.

Bien que la lecture de la réponse à l'interrogation *OPC? supprime la complication de traiter les demandes de service et l'interrogation série relative, elle présente l'inconvénient suivant: aussi bien le **bus du système** que l'établissement d'une liaison avec le **contrôleur** sont dans un état d'arrêt temporaire, pendant que le **contrôleur** attend de lire la réponse à l'interrogation *OPC?. La méthode utilisant l'instruction *OPC et le bit d'événement OPC dans le registre d'état d'événement ou la méthode utilisant l'interrogation *OPC? et le message récapitulatif MAV dans le registre d'octet d'état nécessitent toutes les deux l'utilisation d'une interrogation série. Toutefois, elles permettent aussi bien au **bus du système** qu'au **contrôleur** de réaliser d'autres opérations en attendant la demande de service.

12.6 *Synchronisation avec des signaux de commande extérieurs*

Les techniques décrites dans cette section peuvent être utilisées pour fournir une synchronisation entre des signaux de commande extérieurs, comme défini en 5.6.1.4 et dans le programme d'application. Le concepteur de l'appareil peut fournir des instructions recouvrables spécifiques à l'appareil qui dépendent de la réception de signaux de commande extérieurs. Une opération de l'appareil peut être lancée par l'instruction et terminée en conséquence de la réception d'un signal de commande extérieur.

Par exemple, considérons un DVM possédant l'instruction "TRG:EXT" pour sélectionner un déclenchement extérieur, et l'instruction (et interrogation) recouvrable "MEAS?" pour effectuer une mesure lorsqu'il est déclenché. Le DVM mettra la signalisation d'opération en cours VRAIE à la réception du <MESSAGE DE PROGRAMMATION TERMINÉ> suivant:

*CLS; TRG:EXT; MEAS? <PMT>

Après l'application du signal de déclenchement extérieur, la mesure est effectuée, puis la signalisation "opération en cours" est mise FAUSSE.

Toutes les méthodes de synchronisation utilisant les instructions *WAI, *OPC et l'interrogation *OPC? peuvent être utilisées avec des instructions recouvrables qui utilisent des signaux de commande extérieurs.

12.7 *Utilisation incorrecte de *OPC et *OPC?*

L'instruction *OPC et l'interrogation *OPC? sont destinées à apparaître comme dernière <UNITÉ DE MESSAGE DE PROGRAMMATION> dans un <MESSAGE DE PROGRAMMATION>. Le résultat du <MESSAGE DE PROGRAMMATION TERMINÉ>

*CLS; OL1; OL2; *OPC; OL3 <PMT>

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

12.5.3.3 Sample use of the *OPC? query – Service Request method

The following technique makes use of service request and serial poll. Bit 4 (corresponding to the MAV summary-message) in the Service Request Enable register is set to a value of one (enabled) and the <TERMINATED PROGRAM MESSAGE>

VOLT 5.20; *OPC? <PMT>

is sent to the power supply. The power supply requests service when the output voltage is valid. Then the DVM is commanded to take a measurement.

While reading the response to the *OPC? query removes the complication of dealing with service requests and the attendant serial poll, it has the penalty that both the **system bus** and the **controller** handshake are in a temporary holdoff state while the **controller** is waiting to read the *OPC? query response. The method using the *OPC command and the OPC event-bit in the Standard Event Status Register or the method using the *OPC? query and the MAV summary-message in the Status Byte Register both require the use of a serial poll. However, they allow both the **system bus** and the **controller** to perform other operations while waiting for the service request.

12.6 *Synchronization with External-Control-signals*

The techniques described in this section may be used to provide synchronization between External Control signals as defined in 5.6.1.4 and the application program. The device designer may provide device-specific Overlapped Commands that depend upon receipt of external-control-signals. A **device** operation may be initiated by the command and completed as a consequence of receipt of an external-control-signal.

For example, consider a DVM that has the command "TRG:EXT" to select external trigger, and the Overlapped Command (and query) "MEAS?" to take a measurement when triggered. The DVM will make the Pending-Operation flag TRUE upon receipt of the following <TERMINATED PROGRAM MESSAGE>

*CLS; TRG:EXT; MEAS? <PMT>

After the external trigger signal is applied the measurement is taken, and then the Pending-Operation flag is made FALSE.

All the synchronization methods using the *WAI, *OPC commands and the *OPC? query may be used with Overlapped Commands that use external-control-signals.

12.7 *Improper usage of *OPC and *OPC?*

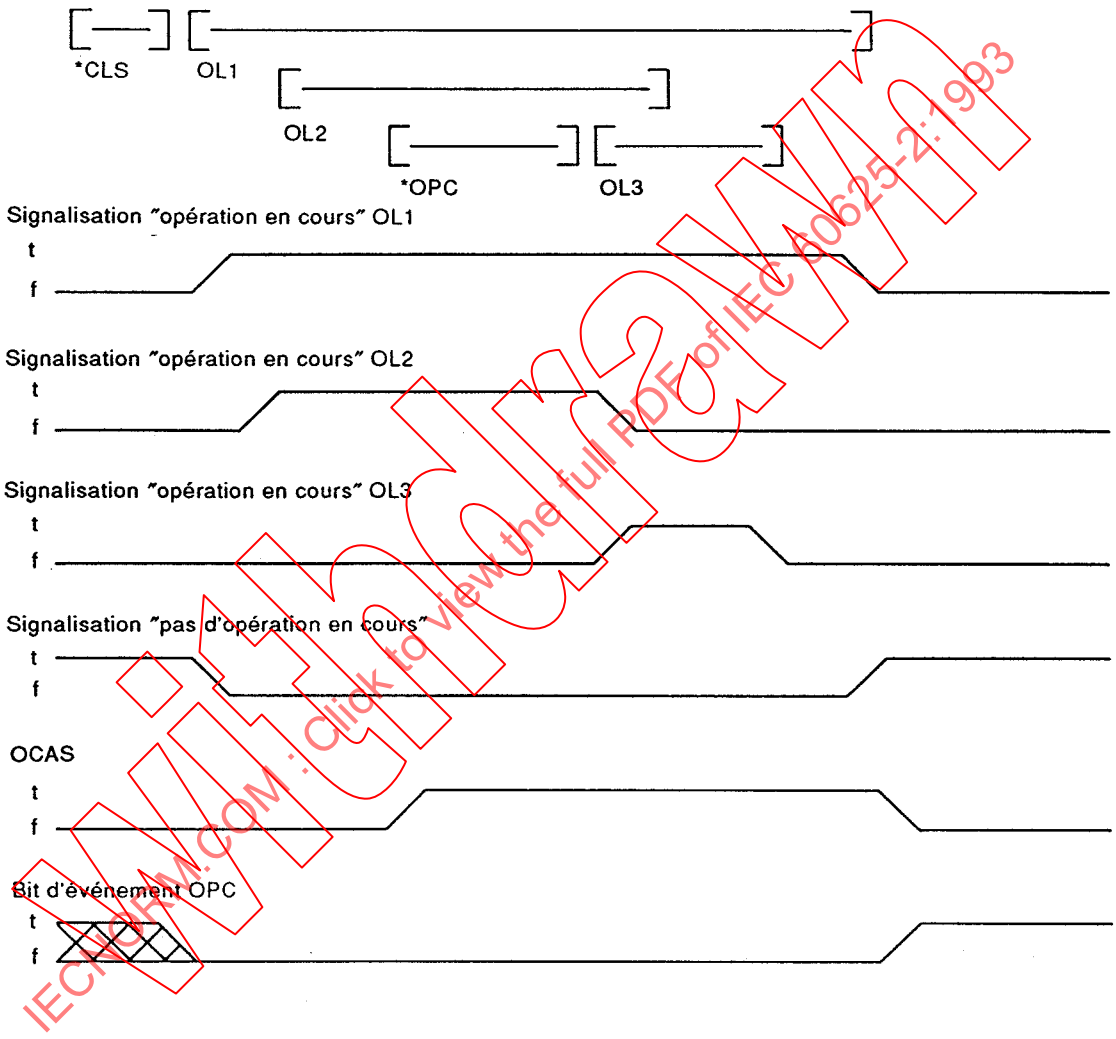
The *OPC command and *OPC? query are intended to appear as the last <PROGRAM MESSAGE UNIT> in a <PROGRAM MESSAGE>. The result of the <TERMINATED PROGRAM MESSAGE>

*CLS; OL1; OL2; *OPC; OL3 <PMT>

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

dépendra de la durée relative d'accomplissement des opérations de l'**appareil** démarrées par les instructions recouvrables "OL1" et "OL2" et l'exécution de l'instruction recouvrable "OL3".

La signalisation "pas d'opération en cours" sera peut-être encore FAUSSE lorsque l'instruction recouvrable "OL3" aura été exécutée. Dans ce cas, le bit d'événement OPC devient VRAI lorsque l'instruction recouvrable "OL3" est terminée, puisque la signalisation "pas d'opération en cours" ne devient pas VRAIE avant que la signalisation "opération en cours" correspondant à l'instruction "OL3" ne devienne VRAIE. Le diagramme de temps et les signalisations dans ce cas sont représentés à la figure 12-7.



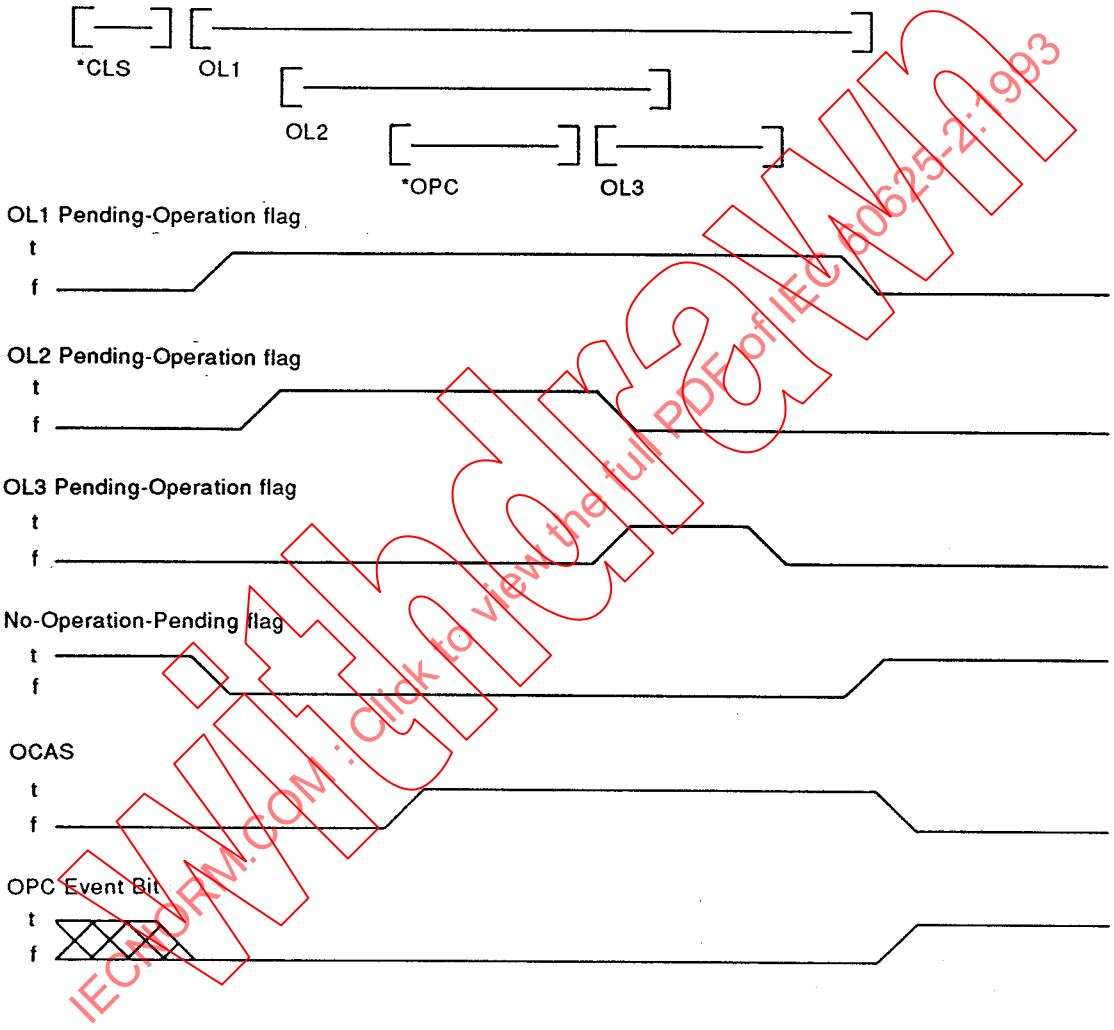
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figure 12-7 – Diagramme de temps – Opération incorrecte, exemple 1

Cependant si la signalisation "pas d'opération en cours" devient VRAIE avant que l'instruction recouvrable "OL3" ne soit exécutée, le bit d'événement OPC sera alors déjà VRAI. Le diagramme de temps et les signalisations dans ce cas sont représentés à la figure 12-8.

will depend upon the relative timing of the completion of the **device** operations started by the "OL1" and "OL2" Overlapped Commands and the execution of the "OL3" Overlapped Command.

The No-Operation-Pending flag may still be FALSE when the Overlapped Command "OL3" is executed. In this case the OPC event-bit becomes TRUE when the Overlapped Command "OL3" is finished, since the No-Operation-Pending flag does not go TRUE before the Pending Operation flag corresponding to the "OL3" command goes TRUE. The timeline and flags for this case are as shown in figure 12-7.



IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Figure 12-7 – Timing diagram – Improper operation, example 1

However, if the No-Operation-Pending flag becomes TRUE before the Overlapped Command "OL3" is executed then the OPC event-bit will already be TRUE. The timeline and flags for this case are as shown in figure 12-8.

12.8 *Considérations de conception*

Cet article est destiné à guider le concepteur de l'appareil pour choisir les instructions qui devraient être des instructions recouvrables et celles qui devraient être séquentielles. Il fournit aussi un guide au concepteur de l'appareil sur la façon dont la signalisation "pas d'opération en cours" devra être générée.

12.8.1 *Instructions recouvrables*

En général, un concepteur d'appareil voudra rendre les instructions séquentielles si elles se terminent en un laps de temps court, et recouvrables si elles mettent beaucoup de temps à se terminer. "Temps court" et "temps long" doivent être considérés dans le contexte d'autres instructions de l'appareil.

Par exemple, considérons un enregistreur de données qui peut restituer ses données sur bande magnétique sans interrompre l'enregistrement de nouvelles données. Cette restitution est terminée en une heure, donc l'opération de restitution devrait être recouvrable. Supposons que l'enregistrement de nouvelles données soit fait en une minute. Ceci est une longue durée par comparaison avec de nombreuses opérations de l'appareil (par exemple, répondre à l'nterrogation *STB?) mais une courte durée par rapport à l'opération de restitution.

Considérons le cas où le concepteur de l'appareil choisit de rendre recouvrable l'opération d'enregistrement de données. Lorsque le programme d'application ordonne à l'enregistreur de données de faire une mesure, la signalisation "pas d'opération en cours" deviendra FAUSSE pendant une minute, puis VRAIE. Si le programme d'application ordonne à l'enregistreur de données de faire une mesure alors qu'une opération de restitution est en cours, la signalisation "pas d'opération en cours" restera FAUSSE jusqu'à ce que la restitution soit terminée, cette durée pouvant atteindre une heure. Pendant ce temps, le programme d'application n'a aucun moyen de savoir si la mesure est terminée.

Considérons maintenant le cas où le concepteur de l'appareil choisit de rendre séquentielle l'opération d'enregistrement de données. Ceci peut être une solution acceptable, puisque l'instruction "MESURER" par elle-même ne provoquera pas de retard. Si le **contrôleur** essaie d'envoyer des instructions supplémentaires avant que l'opération d'enregistrement de données ne soit terminée, le **contrôleur** et le bus seront alors "bloqués" jusqu'à ce que l'enregistreur de données soit prêt à exécuter les nouvelles instructions. Noter que cette méthode est très sensible à l'implémentation de l'appareil. Si l'**appareil** n'a pas de tampon d'entrée, l'envoi du <MESSAGE DE PROGRAMMATION TERMINÉ>

MESURER;<espace><PMT>

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

12.8 *Design considerations*

This clause is intended to guide the device designer in choosing which commands should be Overlapped Commands and which should be Sequential. It also provides guidance to the device designer about how the No-Operation-Pending flag shall be generated.

12.8.1 *Overlapped commands*

In general, a device designer will want to make commands sequential if they complete in a short time and overlapped if they take a long time to complete. "Short time" and "long time" must be considered in the context of other device commands.

For example, consider a data logger that can back up its data to tape without interrupting the logging of new data. This back-up takes one hour to complete, so the back-up operation should be overlapped. Assume that logging new data takes one minute. This is a long time compared to many of the device operations (e.g. responding to *STB? query) but a short time compared to the back-up operation.

Consider the case where the device designer chooses to make the data-logging operation overlapped. When the application program commands the data logger to take a measurement, the No-Operation-Pending flag will go FALSE for one minute and then TRUE. If the application program commands the data logger to take a measurement while a back-up operation is in progress, the No-Operation-Pending flag will remain FALSE until the back-up completes, possibly as long as one hour. During this time, the application program has no way of knowing whether the measurement is complete.

Now consider the case where the device designer chooses to make the data logging operation sequential. This may be an acceptable solution since the "MEASURE" command by itself will not cause a delay. If the **controller** attempts to send additional commands before the data-logging operation is complete, then the **controller** and bus will be "stalled" until the data logger is ready to execute the new commands. Note that this method is very sensitive to device implementation. If the **device** has no Input Buffer, sending the <TERMINATED PROGRAM MESSAGE>

MEASURE;<space><PMT>

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

pourrait faire que l'**appareil** attende le point-virgule jusqu'à ce que la mesure soit terminée et ensuite il serait prêt à accepter le reste du <MESSAGE DE PROGRAMMATION TERMINÉ>.

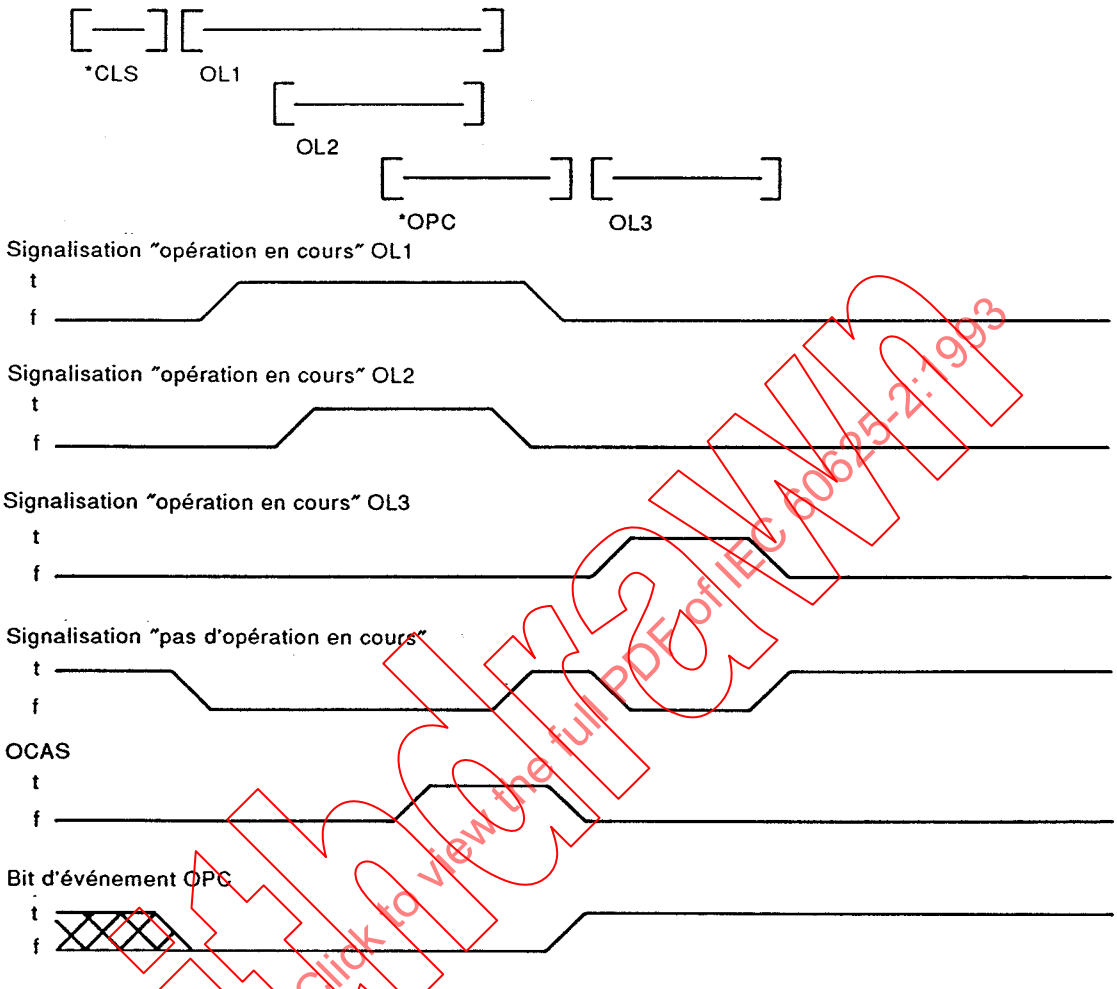


Figure 12-8 – Diagramme de temps – Opération incorrecte, exemple 2

NOTE – Si l'opération OL1 ou OL2 est étendue au-delà de *OPC, OL3 décidera alors du moment où l'événement OPC se produit.

Un tampon d'entrée de "longueur raisonnable" (voir 6.1.5) serait assez grand pour maintenir l'instruction, l'espace blanc supplémentaire, et le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> de sorte que l'établissement d'une liaison avec le bus ne serait pas bloqué pendant la mesure.

Si l'enregistreur de données hypothétique devait renvoyer certaines données au **contrôleur** à la fin de chaque mesure, une opération séquentielle ne serait pas acceptable, puisque le **contrôleur** et le bus seraient bloqués pendant la durée de chaque mesure. Dans ce cas, le concepteur de l'appareil pourrait choisir d'ignorer la signalisation d'opération en cours "RESTITUTION" en déterminant l'état de la signalisation "pas d'opération en cours". (Le "sous-ensemble défini par l'appareil" à 12.4 n'inclurait pas la signalisation d'opération en cours "RESTITUTION"). Ceci permettrait au programmeur d'application d'utiliser les instructions standard *WAI, *OPC et *OPC? pour synchroniser les mesures. Le concepteur de l'appareil devrait fournir d'autres moyens spécifiques à l'appareil pour déterminer quand l'opération de restitution est terminée.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

could cause the **device** to wait after the semicolon until the measurement completes and it is ready to accept the rest of the <TERMINATED PROGRAM MESSAGE>.

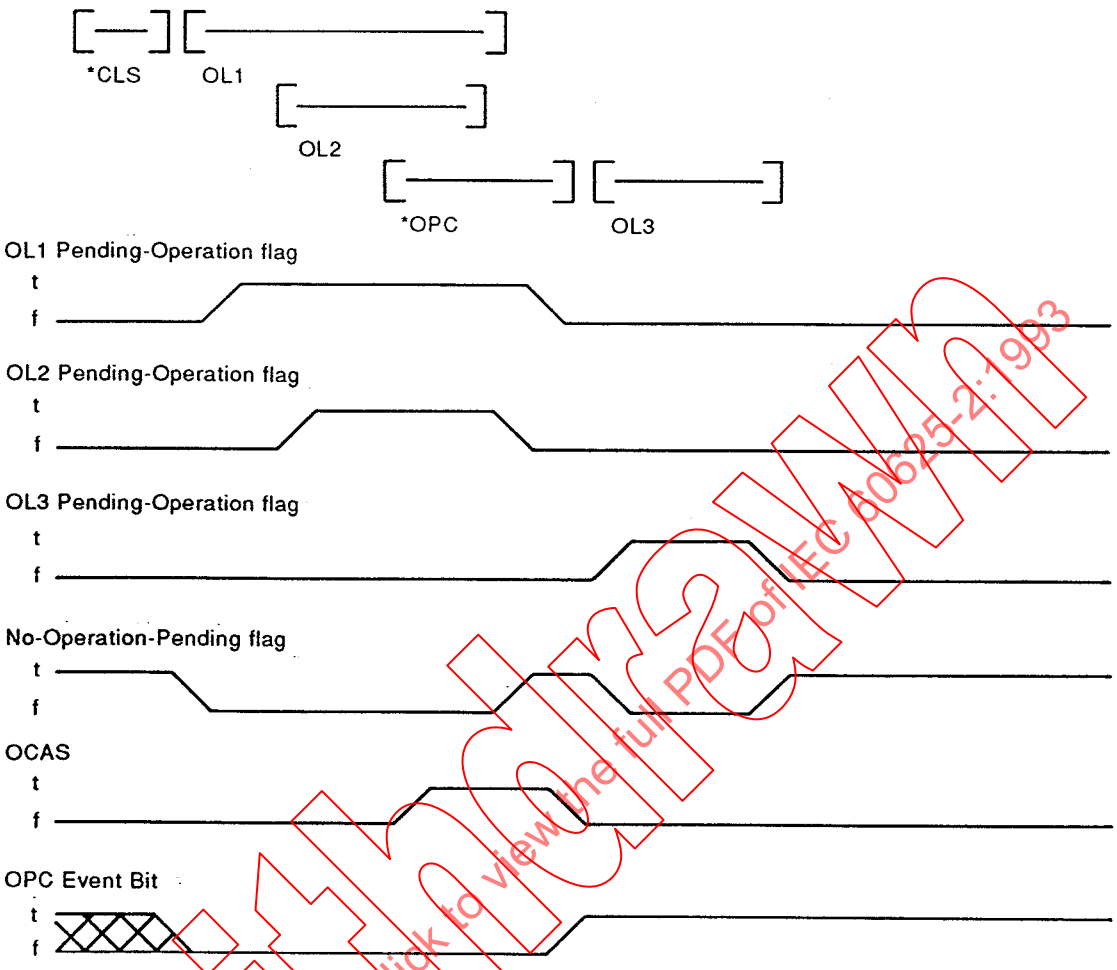


Figure 12-8 – Timing diagram – Improper operation, example 2

NOTE – If OL1 or OL2 operation is extended beyond *OPC, then OL3 will decide the time at which the OPC event occurs.

A "reasonable length" Input Buffer (see 6.1.5) would be large enough to hold the command, extra white space, and <PROGRAM MESSAGE TERMINATOR> so that the bus handshake would not be stalled during the measurement.

If the hypothetical data logger had to return some data to the **controller** at the end of each measurement, sequential operation would not be acceptable since the **controller** and bus would be stalled for the duration of each measurement. In this case, the device designer might choose to ignore the "BACKUP" Pending-Operation flag when determining the state of the No-Operation-Pending flag. (The "device-defined subset" in 12.4 would not include the "BACKUP" Pending-Operation flag.) This would allow the application programmer to use the standard *WAI, *OPC, and *OPC? commands to synchronize measurements. The device designer would have to provide some other device-specific means of determining when the bac-kup operation had completed.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Le concepteur de l'appareil pourrait également créer une instruction spécifique à l'appareil pour sélectionner les signalisations d'opérations en cours qui affecteraient la signalisation "pas d'opération en cours". Ceci permettrait au programmeur d'application d'utiliser les instructions *WAI, *OPC et *OPC? pour attendre que la mesure soit terminée, que la restitution soit terminée, ou les deux.

12.8.2 *Traitement de l'erreur d'exécution*

Si un <MESSAGE DE PROGRAMMATION> entraîne une erreur d'exécution, l'opération associée est suspendue, la signalisation "opération en cours" est mise FAUSSE et l'**appareil** signale une erreur d'exécution.

12.8.3 *Opération achevée*

Le concepteur de l'appareil devra déterminer le moment dans le temps où les instructions recouvrables terminent leurs opérations de l'**appareil** associées. Les principes suivants devront être suivis pour la détermination du moment où la signalisation "opération en cours" doit être mise FAUSSE pour chaque instruction recouvrable associée définie par l'appareil.

Le concepteur de l'appareil doit se rendre compte de l'importance des exigences de cet article. Ces exigences permettent au logiciel d'application de déterminer quand des composants externes d'un système peuvent utiliser les résultats d'une opération de l'**appareil**. Ainsi, l'**appareil** ne devra pas indiquer prématurément qu'une opération est achevée. L'**appareil** peut indiquer qu'une opération est terminée à tout moment une fois que l'opération est réellement terminée.

Le concepteur de l'appareil peut inclure un matériel spécial, par exemple des capteurs, pour que l'**appareil** détermine quand une opération est terminée. De plus, le concepteur de l'appareil peut concevoir l'**appareil** de façon qu'il attende qu'un laps de temps approprié s'écoule. Toute méthode est acceptable tant que les résultats d'une opération sont corrects chaque fois que l'**appareil** indique que cette opération est achevée. Un **appareil** indiquant qu'une opération est achevée, alors qu'en fait elle ne l'est pas, est mal utilisé, ou hors étalonnage, ou défectueux. Un générateur de signal connecté à un court-circuit est un exemple de mauvaise utilisation. Pour chaque instruction, la documentation de l'**appareil** devra spécifier les critères fonctionnels qui sont satisfaits lorsqu'un message "opération terminée" est généré en réponse à cette instruction.

Les critères qui devront être satisfaits avant qu'une opération ne soit signalée achevée sont indiqués séparément pour deux catégories d'appareils: les appareils stimulateurs (voir 12.8.3.1) et les appareils répondeurs (voir 12.8.3.2). Un **appareil** peut contenir ces deux types de fonctions.

12.8.3.1 Appareils stimulateurs

Un appareil stimulateur génère un signal qui est utilisé pour stimuler un élément d'équipement externe dans un système. Exemple d'appareil stimulateur: une alimentation.

Lorsqu'une instruction recouvrable spécifique à l'appareil fait générer un signal à l'**appareil**, l'appareil ne devra pas indiquer que l'opération en cours associée est terminée tant que:

1) le signal sur son point de sortie n'est pas dans un état valide,
2) l'opération n'a pas été suspendue.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The device designer could also create a device-specific command to select which pending-operation flags would affect the No-Operation-Pending flag. This would allow the application programmer to use the *WAI, *OPC, and *OPC? commands to wait for the measurement to complete, for the back-up to complete, or both.

12.8.2 *Execution Error handling*

If a <PROGRAM MESSAGE> causes an Execution Error, the associated operation is aborted, the Pending-Operation flag is made FALSE, and the **device** reports an Execution Error.

12.8.3 *Operation complete*

The device designer shall determine the point in time when Overlapped Commands complete their associated **device** operations. The following principles shall be followed in determining when to set the Pending-Operation flag FALSE for each associated device-defined Overlapped Command.

The device designer must realize the importance of the requirements of this clause. These requirements allow application software to determine when external components of a system may utilize the results of a **device** operation. Thus the **device** shall not prematurely report that an operation is complete. The **device** may indicate that the operation is complete anytime after the operation really is complete.

The device designer may include special hardware, such as sensors, for the **device** to determine when an operation is complete. Alternatively, the device designer may design the **device** to wait for an appropriate amount of time to elapse. Any method is acceptable as long as the results of an operation are valid whenever the **device** reports that operation to be complete. A **device** that reports an operation to be complete when, in fact, it is not, is either misapplied, out of calibration, or defective. A signal generator connected to a short circuit is an example of a misapplication. For each command the **device** documentation shall specify the functional criteria that are met when an operation complete message is generated in response to that command.

The criteria that shall be satisfied before an operation is reported complete are shown separately for two classes of devices: stimulus-devices (see 12.8.3.1) and response-devices (see 12.8.3.2). A **device** may contain functions of both types.

12.8.3.1 Stimulus-devices

A stimulus-device generates a signal which is used to stimulate an external piece of equipment in a system. An example of a stimulus-device is a power supply.

When a device-specific Overlapped Command causes the **device** to generate a signal the device shall not indicate that the associated pending operation is complete until one of the following has occurred:

1) the signal on its output port is in a valid state,
2) or the operation has been aborted.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

L'**appareil** peut fournir des instructions spécifiques à l'appareil qui permettent au programmeur d'application de sélectionner quelles caractéristiques du signal appliqué sur le point de sortie de l'**appareil** mettront la signalisation "opération en cours" FAUSSE. Par exemple, ces instructions pourraient sélectionner l'exactitude à laquelle un signal de sortie doit se conformer avant que l'**appareil** n'indique qu'une opération est achevée.

12.8.3.2 Appareils répondeurs

Un appareil répondeur est un **appareil** qui répond à un signal fourni extérieurement. Exemple d'appareil répondeur: un DVM qui mesure un signal extérieur.

Lorsqu'une instruction recouvrable spécifique à l'appareil fait répondre l'**appareil** à un signal extérieur, l'**appareil** ne devra pas indiquer que l'opération en cours associée est achevée tant que:

1) la réponse n'est pas achevée, ou
2) l'opération n'a pas été suspendue, ou
3) l'**appareil** n'est pas prêt pour que le **contrôleur** commence une opération de lecture en réponse à une instruction recouvrable d'interrogation.

L'appareil peut fournir des instructions spécifiques à l'appareil qui permettent au programmeur d'application de sélectionner quelle(s) caractéristique(s) de la réponse positionne(nt) la signalisation "opération en cours" FAUSSE. Par exemple, ces instructions pourraient sélectionner le degré d'exactitude auquel le signal est mesuré.

13 Configuration automatique du système

13.1 *Introduction*

Lorsqu'un système de test est configuré ou reconfiguré, des adresses doivent être assignées aux appareils pour que le programme d'application du **contrôleur** puisse envoyer des messages aux appareils désirés et recevoir des messages de ces appareils. Cet article fournit des outils sous la forme de protocoles, d'instructions, et de directives pour automatiser cette tâche pour les **appareils**, sans se préoccuper de la conformité avec la CEI 625-2. Elle traite aussi de la façon dont une information générée par les outils peut être utilisée par un programme d'application pour s'adapter automatiquement aux changements matériels du système de test.

13.2 *Vue d'ensemble*

Les combinaisons de protocoles et instructions communes suivantes prennent en charge les configurations automatiques:

Protocole de contrôleur	Instruction commune	But de la combinaison
Recherche des écouteurs (voir 17.6)	Invalider l'écouteur (*DLF) (voir 10.6 et 13.4.1)	Trouver des appareils d'adresse non configurable CEI 625-1 et CEI 625-2 (*AAD/*DLF non implémenté)
Sélectionner l'adresse (voir 17.7)	Accepter l'adresse (*AAD) (voir 10.1 et 13.4.2)	Trouver des appareils d'adresse configurable (*AAD/*DLF implémenté)

Une vue d'ensemble de l'utilisation de ces combinaisons est donnée ci-dessous et ne représente pas une implémentation réelle. La figure 13-1 illustre une recherche poussée des appareils sur un bus CEI 625-1.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The **device** may provide device-specific commands which allow the application programmer to select which characteristics of the signal applied at the **device's** output port will set the Pending-Operation flag FALSE. For example, these commands might select the accuracy to which an output signal must settle before the **device** indicates that an operation is complete.

12.8.3.2 Response-devices

A response-device is a **device** which responds to an externally provided signal. An example of a response-device is a DVM which measures an external signal.

When a device-specific Overlapped Command causes the **device** to respond to an external signal the **device** shall not indicate that the associated pending operation is complete until one of the following has occurred:

1) the response is complete,
2) or the operation has been aborted,
3) or the **device** is ready for the **controller** to begin a read operation in response to a query Overlapped Command.

The **device** may provide device-specific commands which allow the application programmer to select which characteristic(s) of the response sets the Pending-Operation flag FALSE. For example, these commands might select the degree of accuracy to which the signal is measured.

13 Automatic system configuration

13.1 *Introduction*

When a test system is configured or reconfigured, addresses must be assigned to the devices so that the **controller** application program can send messages to and receive messages from the desired devices. This clause provides tools in the form of protocols, commands, and guidelines to automate this task for **devices** regardless of compliance with IEC 625-2. It also discusses how the information generated by the tools can be used by an application program to automatically adjust to test system hardware changes.

13.2 *Overview*

The following protocol and common command combinations support automatic configuration:

Controller protocol	Common command	Purpose of combination
Find Listeners (see 17.6)	Disable Listener (*DLF) (see 10.6 and 13.4.1)	To find non-address-configurable IEC 625-1 and IEC 625-2 devices (*AAD/*DLF not implemented)
Set Address (see 17.7)	Accept Address (*AAD) (see 10.1 and 13.4.2)	To find address-configurable IEC 625-2 devices (*AAD/*DLF implemented)

An overview of the use of these combinations is given below and does not represent an actual implementation. Figure 13-1 illustrates a comprehensive search of devices on an IEC 625-1 bus.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

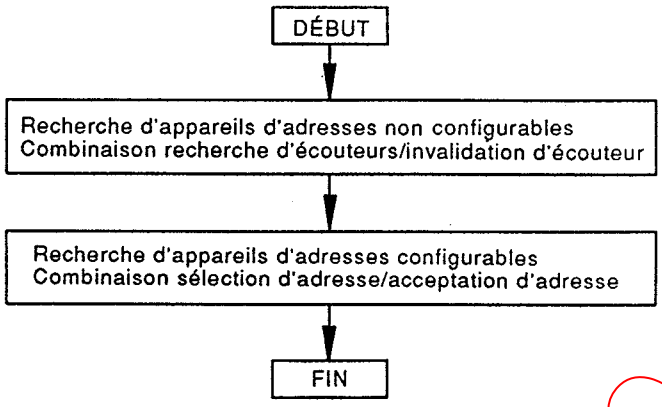


Figure 13-1 – Recherche poussée des appareils

La recherche d'appareils d'adresses non configurables est composée des éléments représentés à la figure 13-2, suivis par les éléments de recherche d'appareils d'adresses configurables à la figure 13-3.

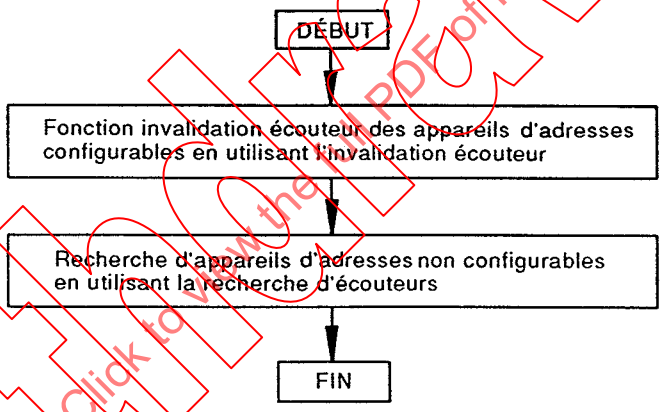


Figure 13-2 – Recherche d'appareils d'adresses non configurables

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

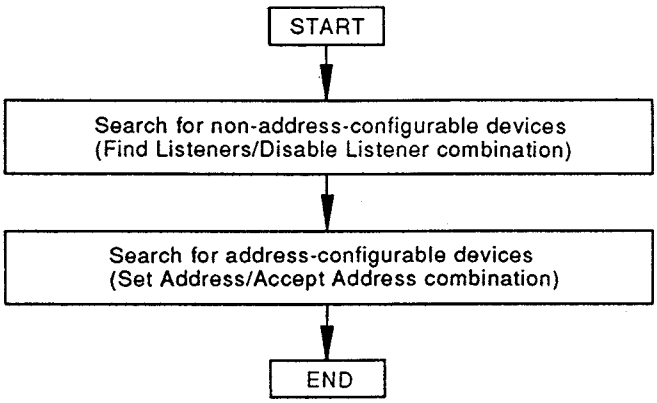


Figure 13-1 – Comprehensive Device Search

The Non-Address-Configurable Device Search in turn is comprised of the elements shown in figure 13-2 followed by the Address-Configurable Device Search elements in figure 13-3.

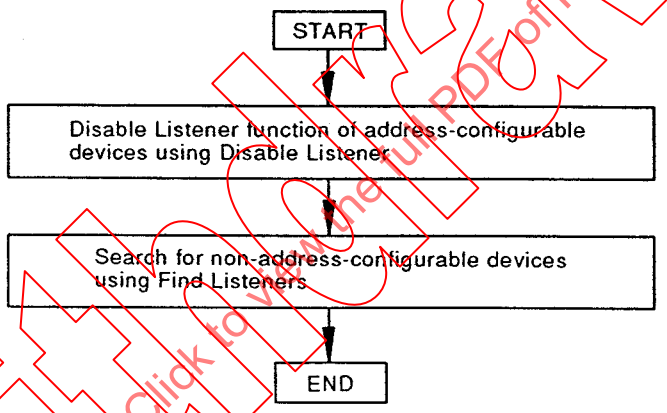


Figure 13-2 – Non-Address-Configurable Device Search

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

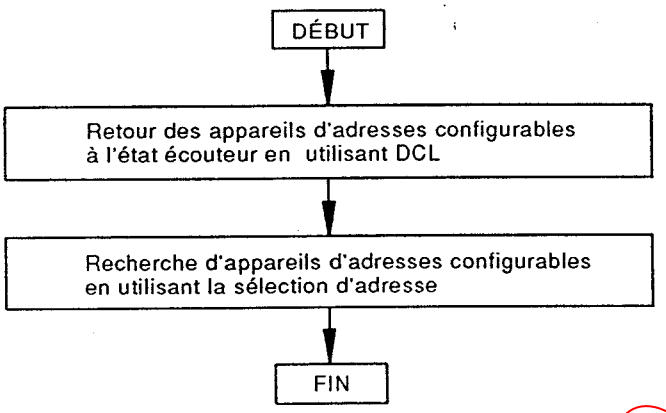


Figure 13-3 – Recherche d'appareils d'adresses configurables

Le protocole du contrôleur de recherche des écouteurs mentionné à la figure 13-2 et la sélection d'adresse à la figure 13-3 sont plus amplement décrits aux figures 13-4 et 13-5 respectivement. Le protocole de sélection d'adresse dialogue avec l'instruction commune d'acceptation d'adresse de la figure 13-6 pour implémenter la recherche d'appareils d'adresses configurables.

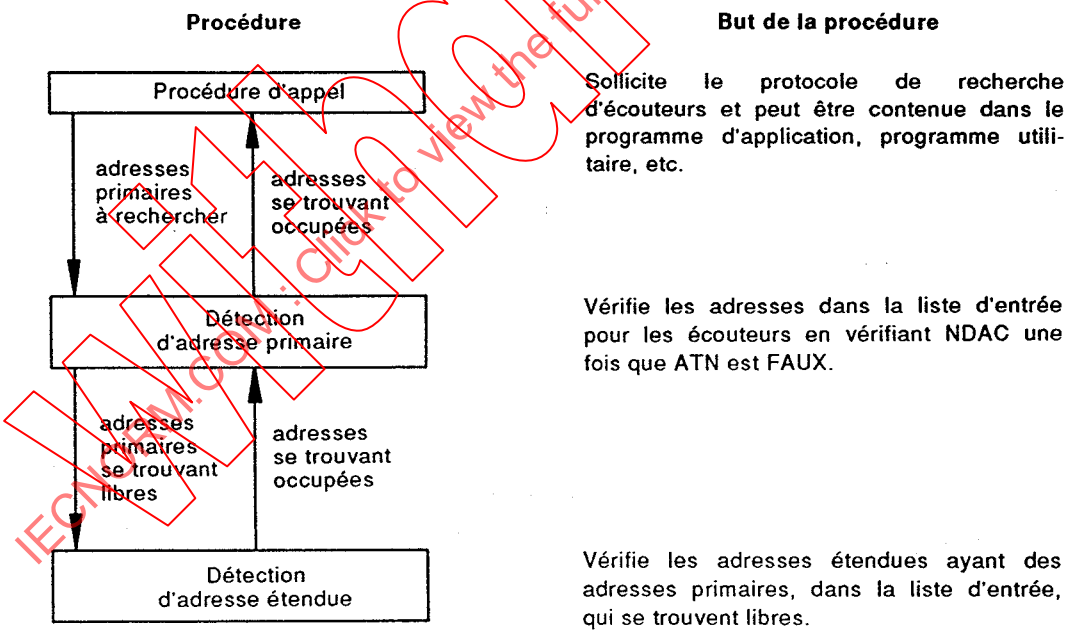


Figure 13-4 – Protocole de recherche d'écouteurs

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

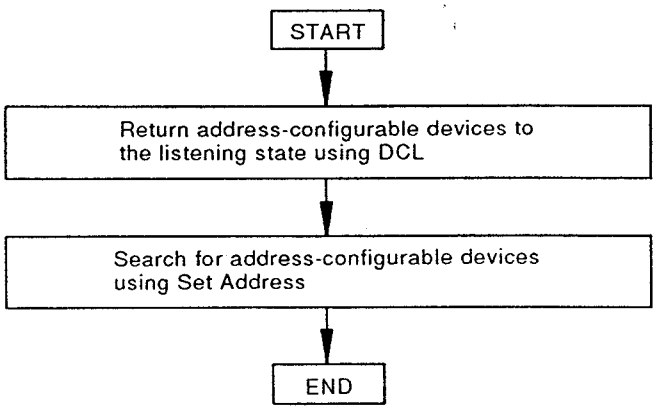


Figure 13-3 – Address-Configurable Device Search

The Find Listeners controller protocol mentioned in figure 13-2 and Set Address in figure 13-3 are further described in figures 13-4 and 13-5 respectively. Set Address protocol interacts with the Accept Address common command of figure 13-6 to implement the address configurable Device Search.

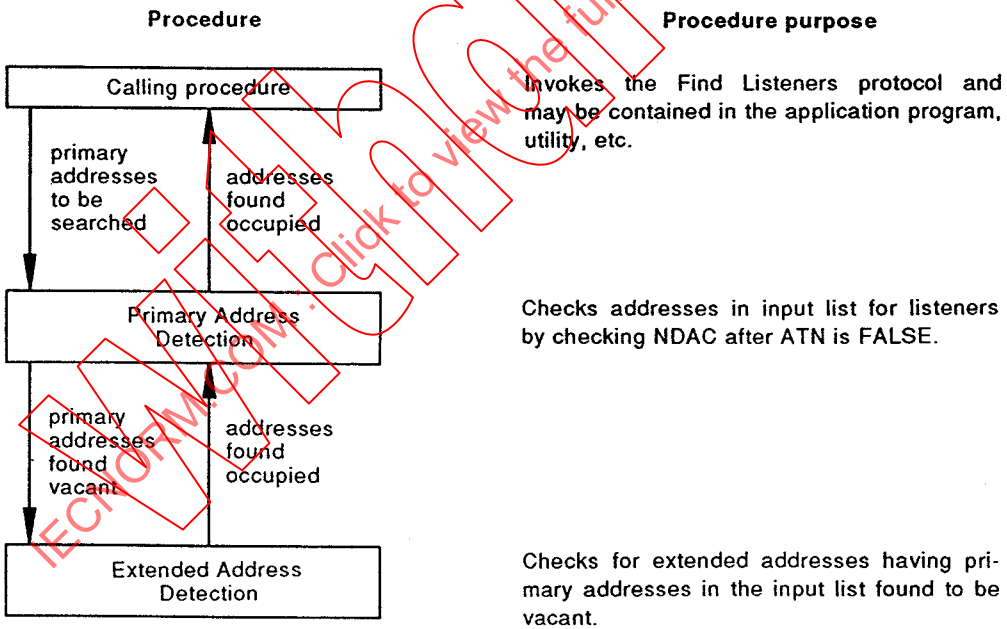
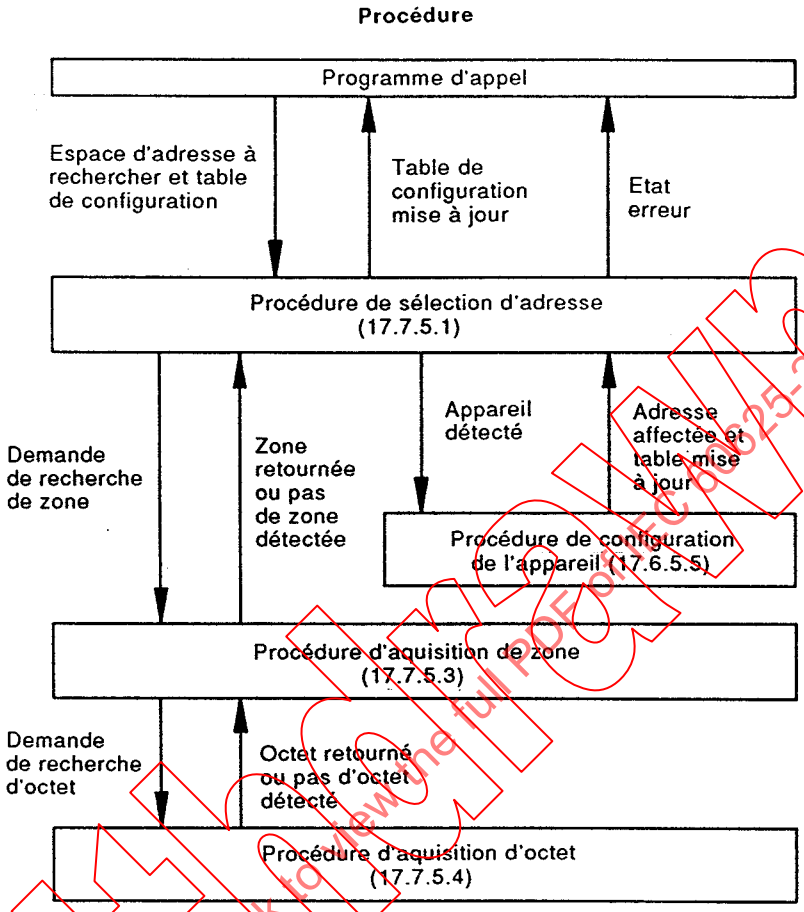


Figure 13-4 – Find Listeners protocol

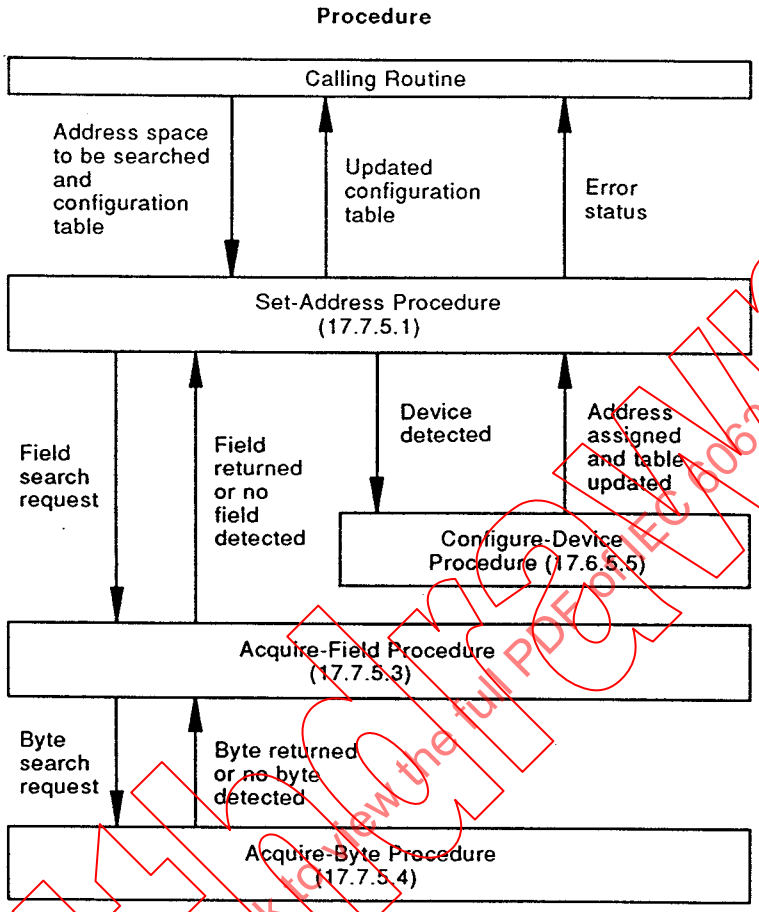
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



Procédure	But de la procédure
Programme d'appel	Sollicite le protocole de sélection d'adresses et peut être contenu dans le programme d'application, ou un programme utilitaire, etc.
Procédure de sélection d'adresses (17.7.5.1)	Recherche les appareils d'adresses configurables dans l'espace d'entrée d'adresse, gère l'acquisition de zones d'identificateur de l'appareil (voir 13.4.2.2) et configure l'**appareil**.
Procédure de configuration d'appareil (17.7.5.5)	Met à jour la table de configuration pour inclure l'**appareil** détecté et affecte une adresse à cet **appareil**.
Procédure d'acquisition de zone (17.7.5.3)	Gère l'acquisition d'octets dans une zone d'identificateur de l'appareil (voir 13.4.2.2.).
Procédure d'acquisition d'octet (17.7.5.4)	Exécute une recherche binaire pour identifier l'octet de donnée d'identificateur de l'appareil à taux de mouvement le plus faible présenté par un (ou des) appareil(s) et ordonner seulement à cet **appareil** de continuer à participer.

Figure 13-5 – Protocole de sélection d'adresses

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn



Procedure	Procedure purpose
Calling routine	Invokes the Set Address protocol and may be contained in the application program, or a utility, etc.
Set-Address procedure (17.7.5.1)	Searches for address-configurable devices in the input address space, manages the acquisition of device identifier fields (see 13.4.2.2) and configures the **device**.
Configure-Device procedure (17.7.5.5)	Updates configuration table to include the **device** detected and assigns an address to that **device**.
Acquire-Field procedure (17.7.5.3)	Manages the acquisition of bytes within a device identifier field (see 13.4.2.2.).
Acquire-Byte procedure (17.7.5.4)	Performs a binary search to identify the lowest active device identifier data byte being presented by a device(s) and instruct only that **device** to continue participating.

Figure 13-5 – Set Address protocol

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

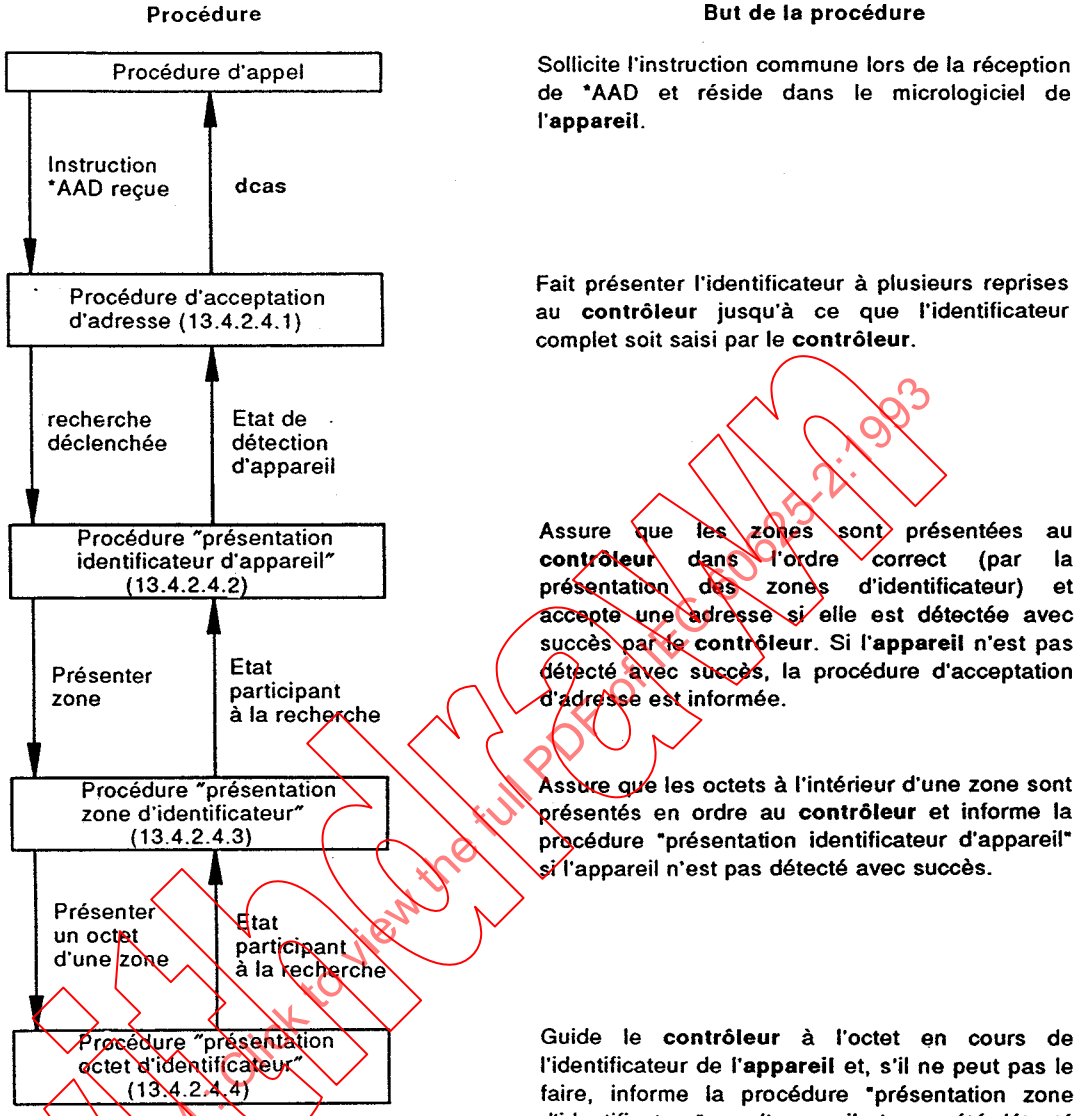


Figure 13-6 – Instruction commune d'acceptation d'adresse

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Le protocole du contrôleur de sélection d'adresse et l'instruction commune d'acceptation d'adresse interagissent comme suit.

La procédure "présentation octet d'identificateur" dans chaque appareil s'efforce de guider le **contrôleur** à son octet en cours qui est détecté par la procédure de saisie d'octet dans le **contrôleur**. La procédure de saisie d'octet saisit chaque octet par une recherche binaire jusqu'à ce qu'un identificateur complet soit saisi (voir annexe C). A ce stade, on ne laisse participer qu'un seul **appareil**, et la procédure de configuration d'appareil du **contrôleur** affecte à l'**appareil** une adresse qui est acceptée par la procédure "présentation identificateur d'appareil" de l'**appareil**. Ce scénario se répète jusqu'à ce que tous les appareils aient été détectés et que des adresses leur aient été affectées.

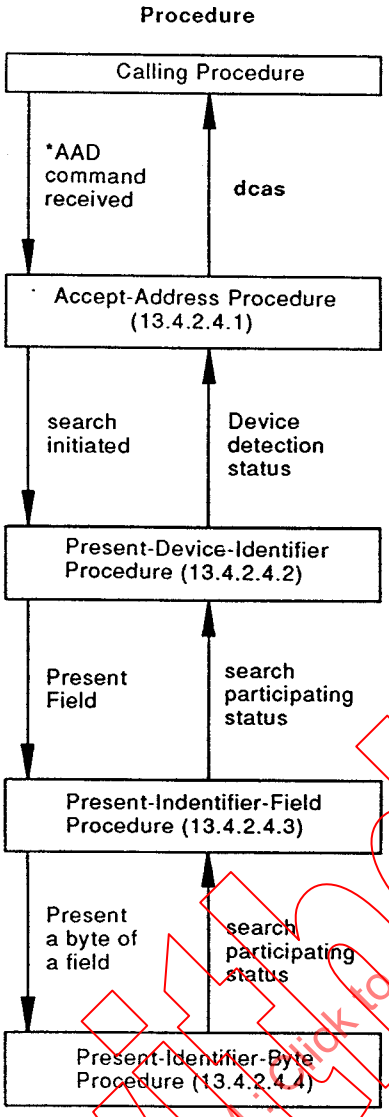


Figure 13-6 – Accept Address common command

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

The Set Address controller protocol and Accept Address common command interact as follows.

The Present-Identifier-Byte procedure in each device endeavours to guide the **controller** to its current byte which is detected by the Acquire-Byte procedure in the **controller**. The Acquire-Byte procedure acquires each byte via a binary search until one entire identifier is acquired (see annex C). At this point, only one **device** is left participating, and the **controller's** Configure-Device procedure assigns the **device** an address which is accepted by the **device's** Present-Device-Identifier procedure. This scenario repeats until all devices have been detected and have been assigned addresses.

13.3 *Approche générique de la configuration automatique du système*

13.3.1 *Affectation d'adresse*

La méthode d'affectation d'adresse décrite ci-dessous est une approche générique et peut être adaptée pour des exigences spécifiques. Les appareils d'adresses non configurables sont détectés les premiers, suivis par les appareils d'adresses configurables. Le programme d'application est responsable de la fourniture d'une liste d'adresses à chercher pour la présence des appareils et d'un tableau décrivant la configuration finale désirée.

13.3.1.1 Types d'appareils

Trois types d'appareils peuvent se trouver sur un bus CEI 625-1:

- les appareils non CEI 625-2
- les **appareils** d'adresses non configurables (pas de *AAD)
- les **appareils** d'adresses configurables (implémentant *AAD).

Les deux premiers types sont définis comme étant d'adresse non configurable. Le troisième type est défini comme étant d'adresse configurable.

13.3.1.2 Détection des appareils d'adresses non configurables

Les **appareils** d'adresses non configurables sont détectés par un programme de sondage d'adresse (protocole de recherche des écouteurs du contrôleur, voir 17.6) qui recherche chaque adresse spécifiée dans la liste d'adresses d'entrée pour un ou plusieurs **appareils** écouteurs. Pendant l'interrogation, les capacités écouteur des **appareils** d'adresses configurables sont invalidées par l'instruction commune *DLF (fonction invalidation écouteur, voir 10.6) pour les empêcher de répondre à l'interrogation. Le protocole exécute la recherche en notant si la ligne NDAC est VRAIE une fois que l'adresse a été envoyée. Dans l'affirmative, un (ou des) appareil(s) existe(nt). Sinon, l'emplacement d'adresse est vide. Si l'emplacement d'adresse est occupé, l'adresse est stockée dans un tableau d'adresses non configurables.

13.3.1.3 Détection des appareils d'adresses configurables

Les **appareils** d'adresses configurables sont détectés via un programme de recherche binaire (protocole de sélection d'adresses du contrôleur, voir 17.7, et l'instruction commune *AAD). Chaque **appareil** recherche la détection en guidant le **contrôleur** à son identificateur avec ses fonctions écouteur et demande de service. L'identificateur est composé des trois premières zones de la réponse à l'interrogation *IDN? (fabricant, numéro de modèle et numéro de série) et l'adresse d'écoute en cours de l'**appareil**. Une fois qu'un identificateur a été détecté, le **contrôleur** affecte une adresse à l'**appareil** associé suivant un tableau de configuration fourni par l'utilisateur. Le tableau exclura normalement les adresses occupées par les **appareils** d'adresses non configurables pour éviter les conflits de communication.

IECNORM.COM : Click to view the PDF of IEC 60625-2:1993
Withdrawn

13.3 *Generic approach to automatic system configuration*

13.3.1 *Address assignment*

The method of address assignment described below is a generic approach and may be tailored to specific requirements. The non-address-configurable devices are detected first, followed by address-configurable devices. The application program is responsible for supplying a list of addresses to be searched for the presence of devices and a table describing the desired final configuration.

13.3.1.1 Types of devices

Three types of devices may reside on an IEC 625-1 bus:

- non-IEC 625-2 devices
- non-address-configurable **devices** (no *AAD)
- address-configurable **devices** (implement *AAD).

The first two types are defined as non-address-configurable. The third type is defined as address-configurable.

13.3.1.2 Non-address-configurable device detection

Non-address-configurable **devices** are detected by an address probing routine (controller Find Listeners protocol, see 17.6) which searches each address specified in the input address list for a listener **device** or **devices**. During the poll, the listener capabilities of the address-configurable **devices** are disabled by the *DLF common command (Disable Listener Function, see 10.6) to prevent them from responding to the poll. The protocol performs the search by noting if the NDAC line is TRUE after the address is sent. If so, a device(s) exist(s). Otherwise, the address location is empty. If the address location is occupied, the address is stored in a non-address-configurable address table.

13.3.1.3 Address-configurable device detection

Address-configurable **devices** are detected via a binary searching routine (controller Set Address protocol, see 17.7, and the *AAD Common Command). Each **device** seeks detection by guiding the **controller** to its identifier with its Listener and Service Request functions. The identifier is made up of the first three fields of the *IDN? query response (manufacturer, model number and serial number) and the **device's** current listen address. Once an identifier is detected, the **controller** assigns an address to the associated **device** according to a user supplied configuration table. The table will normally exclude addresses occupied by the non-address-configurable **devices** to prevent communication conflicts.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

13.3.1.4 Tableau de configuration du système

Une fois que tous les **appareils** ont été détectés, le programme d'application peut créer un tableau de configuration du système qui fait la liste de chacun des appareils détectés par ces protocoles. Ce programme d'application peut aussi placer d'autres données dans le tableau, comme celles renvoyées par les interrogations communes *RDT? ou *PUD?. Certains protocoles du **contrôleur** décrits à l'article 17 nécessitent une liste d'entrée des adresses. Le programme d'application peut utiliser le tableau de configuration du système pour fournir ces adresses aux protocoles. Ainsi, le logiciel d'application peut se régler automatiquement en réponse à n'importe quelle configuration ou reconfiguration du système d'essai qu'il contrôle.

13.3.1.5 Limitations

Le protocole de recherche des écouteurs ne peut pas détecter plusieurs appareils à la même adresse. De cette façon, l'utilisateur doit prendre la responsabilité de s'assurer que chaque **appareil** d'adresse non configurable réside à une adresse unique. Le protocole ne fonctionnera pas non plus s'il y a dans le système des appareils qui prennent plus de 1 ms pour cesser d'être en écoute une fois que le **contrôleur** a envoyé ATN FAUX lorsqu'ils ne sont pas adressés.

Le protocole de sélection d'adresses détectera sans ambiguïté chaque **appareil** d'adresse configurable sans s'occuper de son adresse, sauf quand un identificateur n'est pas unique. Une telle situation peut se produire si deux **appareils** semblables ont "0" comme numéro de série et que leurs adresses sont les mêmes au commencement de la recherche. Si l'utilisateur change l'une de ces adresses, la méthode de recherche s'opérera alors correctement.

Le protocole de sélection d'adresses ne fonctionnera pas si l'un des **appareils** d'adresses non configurables du système met SRQ VRAI pendant le protocole de sélection d'adresses. Dans certains cas, l'interrogation série de tous les **appareils** trouvés par le protocole de recherche d'écouteur, avant de démarrer le protocole de sélection d'adresses, peut être suffisante pour effacer la ligne SRQ. Dans d'autres cas, d'autres moyens doivent être utilisés pour que ces appareils mettent SRQ FAUX pendant la durée du protocole.

13.3.1.6 Procédure

1) Envoyer toutes les adresses d'écoute primaires et étendues, ce qui fait que tous les **appareils** du système écoutent.

2) Envoyer le message **dcas** *DLF <NL^END> et attendre 100 ms, ce qui met tous les **appareils** d'adresses configurables hors d'écoute.

3) Envoyer *CLS <NL^END>. Cette instruction efface l'erreur d'interrogation dans tous les **appareils** d'adresses non configurables, causée par la réception de l'instruction non reconnue, *DLF, et qui leur fait libérer SRQ.

4) Réaliser le protocole de recherche d'écouteurs à toutes les adresses. Chaque adresse est testée pour la présence d'appareils d'adresses non configurables. Après quoi, tous ces appareils sont désadressés.

5) Placer les appareils d'adresses non configurables (trouvés lors de la phase précédente) dans le tableau de configuration du système.

6) Envoyer DCL et attendre 100 ms pour qu'il rétablisse les **appareils** d'adresses configurables en opération d'écoute normale (LACS ou LADS).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

13.3.1.4 System configuration table

After all **devices** have been detected, the application program may create a system configuration table that lists each of the devices detected by these protocols. This application program may also place other data into the table, such as that returned by the *RDT?" or *PUD? common queries. A number of the **controller** protocols described in clause 17 requires an input list of addresses. The application program may use the system configuration table to provide these addresses to the protocols. Thus, the application software can automatically adjust itself in response to any configuration or reconfiguration of the test system which it controls.

13.3.1.5 Limitations

The Find Listeners protocol cannot detect multiple devices at the same address. Thus, the user is responsible for ensuring that each non-address-configurable **device** resides at a unique address. The protocol also will not work if there are devices in the system which take more than 1 ms to cease listening after the **controller** sends ATN FALSE when they are not addressed.

The Set Address protocol will unambiguously detect every address-configurable **device** regardless of its address except when an identifier is not unique. Such a situation can occur if two like **devices** have "0" for a serial number and their addresses are the same at the start of the search. If the user changes one of these addresses, then the searching method will operate correctly.

The Set Address protocol will not work if any non-address-configurable **device** in the system asserts SRQ TRUE during the Set Address protocol. In some cases, serial polling all **devices** found by the Find Listener protocol before starting the Set Address Protocol may be sufficient to clear the SRQ line. In other cases, some other means must be used to make these devices set SRQ FALSE for the duration of the protocol.

13.3.1.6 Procedure

1) Send all primary and extended listen addresses which cause all **devices** in the system to listen.

2) Send **dcas** message *DLF <NL^END> and wait 100 ms which disables all address-configurable devices from listening.

3) Send *CLS <NL^END>. This command clears the Query Error in all non-address-configurable devices caused by receiving the unrecognized command, *DLF, and causes them to release SRQ.

4) Perform Find Listeners protocol at all addresses. Each address is tested for the presence of non-address-configurable devices. Afterwards, all of those devices are unlistened.

5) Place the non-address-configurable devices (found in the previous step) into the system configuration table.

6) Send DCL and wait 100 ms which restores address-configurable **devices** to normal listening operation (LACS or LADS).

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

7) Réaliser le protocole de sélection d'adresses sur tous les **appareils** actuellement adressés en écoute. Ce protocole identifie tous les **appareils** d'adresses configurables, mais ne change pas leurs adresses en cours.

8) Placer les **appareils** d'adresses configurables dans le tableau de configuration du système.

9) Réaliser à nouveau le protocole de sélection d'adresses pour affecter de nouvelles adresses aux **appareils** d'adresses configurables. Le programme d'application ne devrait inclure que des **appareils** qui ont besoin d'une affectation de nouvelles adresses dans cette phase. (Si le programme d'application ne nécessite pas d'affecter des adresses spécifiques à des appareils spécifiques, cette phase peut être omise).

A la conclusion de cette procédure, le tableau de configuration du système contient une liste:

- d'**appareils** d'adresses configurables, comportant leur fabricant, numéro de modèle, numéro de série, et adresse.
- d'adresses d'**appareils** d'adresses non configurables.

13.3.2 *Identification d'appareil*

Le fabricant et le numéro du modèle d'un appareil peuvent être utilisés pour affecter un type d'**appareil** (par exemple DVM, alimentation) à une adresse de bus particulière. Dans certaines applications, il est important de garder un enregistrement des éléments spécifiques de l'équipement d'essai qui ont été utilisés pour un essai donné. Par exemple, ce renseignement peut être utilisé pour mettre à jour un historique d'étalonnage d'un **appareil**.

13.3.2.1 Addition des appareils non CEI 625-2

A la conclusion de la procédure précédente, le tableau de configuration du système comprend le fabricant, le numéro de modèle, le numéro de série, et une adresse pour chaque **appareil** d'adresse configurable. Le tableau contient seulement l'adresse de chaque **appareil** d'adresse non configurable. L'application doit obtenir les trois renseignements qui restent pour les appareils non CEI 625-2, à partir du clavier, d'un fichier ou d'une autre source. Les appareils non CEI 625-2 doivent être identifiés dans le tableau de configuration du système, de sorte qu'ils peuvent être exclus des procédures spécifiques CEI 625-2.

13.3.2.2 Addition d'appareils d'adresses non configurables CEI 625-2

Une fois que les appareils non CEI 625-2 ont été identifiés, l'application peut utiliser l'interrogation commune *IDN? pour remplir le reste des valeurs du tableau pour les **appareils** d'adresses non configurables.

13.3.2.3 Instructions de configuration supplémentaires

Certaines instructions et interrogations communes sont disponibles pour fournir des informations utiles à l'intérieur d'un système de test automatique. Ces instructions comprennent *PUD et *PUD? (données utilisateur protégées), *RDT et *RDT? (transfert de description de ressources), *CAL? (étalonnage), et *OPT? (interrogation d'identification d'option).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

7) Perform the Set Address protocol on all currently listen-addressed **devices**. This protocol identifies all address-configurable **devices**, but does not change their current addresses.

8) Place address-configurable **devices** into the system configuration table.

9) Perform the Set Address protocol again to assign new addresses to address-configurable **devices**. The application program should include only **devices** which need to be assigned new addresses in this step. (If the application program does not need to assign specific addresses to specific **devices**, this step may be omitted.)

At the conclusion of this procedure, the system configuration table contains a list of

- address-configurable **devices**, including their manufacturer, model number, serial number, and address
- non-address-configurable **device** addresses.

13.3.2 *Device identification*

The manufacturer and model number of a **device** may be used by an application to assign a type of **device** (e.g. DVM, power supply) to a particular bus address. In some applications, keeping a record of which specific pieces of test equipment that were used for a given test is important. For example, this information can be used to maintain a calibration history of a **device**.

13.3.2.1 Addition of non-IEC 625-2 devices

At the conclusion of the previous procedure, the system configuration table includes the manufacturer, model number, serial number, and address for each address-configurable **device**. The table contains only the address for each non-address-configurable **device**. The application must obtain the remaining three pieces of information for non-IEC 625-2 devices from the keyboard, a disk file, or other source. Non-IEC 625-2 devices must be identified in the system configuration table so that they can be excluded from IEC 625-2 specific procedures.

13.3.2.2 Addition of IEC 625-2 non-address-configurable devices

After the non-IEC 625-2 devices are identified, the application may use the *IDN?" query to fill in the rest of the table entries for non-address-configurable **devices**.

13.3.2.3 Additional configuration commands

A number of common commands and queries are available to provide information useful within an automatic test system. These commands include *PUD and *PUD? (Protected User Data), *RDT and *RDT? (Resource Description Transfer), *CAL? (Calibration), and *OPT? (Option Identification Query).

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Du fait que ces instructions communes et d'autres sont facultatives, le programme d'application doit être capable de déterminer si un **appareil** particulier a implémenté une instruction ou une interrogation donnée. L'application peut extraire cette information à partir de ses propres connaissances des capacités d'un **appareil**, à partir d'une description de ressources. Une autre méthode consiste à envoyer l'instruction ou l'interrogation, et à utiliser les ressources d'indication d'état nécessitées par l'**appareil** pour déterminer si l'**appareil** n'implémente pas cette instruction. Par exemple, le programme d'application pourrait déterminer si l'**appareil** implémente l'interrogation commune *RDT? avec la procédure suivante:

NOTE – Les guillemets entourant les chaînes de données ne font pas partie de la donnée envoyée.

1) Envoyer "*ESE 1; *SRE 32; *CLS<NL^END>". Ce <MESSAGE DE PROGRAMMATION TERMINÉ> valide l'indication de l'événement OPC (seulement) et efface les registres d'état.

2) Envoyer "*RDT?; *OPC<NL^END>". L'**appareil** peut ou non implémenter l'interrogation *RDT?, mais est obligé d'implémenter l'instruction *OPC (opération achevée). L'**appareil** sélectionnera le bit d'événement OPC dans le registre d'état d'événement standard lorsqu'il exécutera l'instruction *OPC. (On suppose que l'**appareil** n'a aucune opération en cours, ce qui est une supposition raisonnable durant l'initialisation du système). L'**appareil** sélectionnera finalement le bit d'état d'événement (ESB, bit 5) dans le registre d'octet d'état de l'**appareil**, et générera une demande de service.

3) L'application peut attendre le signal de demande de service ou peut interroger de façon répétitive l'**appareil** jusqu'à ce que l'octet d'état de celui-ci indique une demande de service (RQS VRAI). (L'application ne devrait pas supposer que la demande de service est entraînée par cet **appareil**, mais devrait toujours vérifier l'octet d'état de l'**appareil** avant de continuer).

4) Envoyer "*ESR? <NL^END>" sans tenter de lire la réponse à l'interrogation *RDT?. Noter que cette séquence est une violation du protocole d'échange de messages d'appareil normal (voir l'article 6), et peut faire indiquer à l'**appareil** une erreur d'interrogation, que le programme d'application choisit d'ignorer.

5) Lire la réponse à *ESR?. Le bit d'opération achevée (OPC, bit 0) devrait être VRAI (valeur de 1). Le bit d'erreur d'instruction (CME, bit 5) devra être FAUX (valeur de 0) si l'**appareil** implémente l'interrogation commune *RDT? et VRAI (valeur de 1) dans le cas contraire. Le bit d'application devrait ignorer le bit d'événement d'interrogation d'erreur (QYE, bit 2), qui peut ou non être sélectionné.

13.4 *Prescriptions détaillées des instructions de configuration automatique*

13.4.1 *Prescriptions de l'instruction commune *DLF*

L'instruction commune *DLF doit invalider la fonction écouteur normale de l'**appareil** jusqu'à ce qu'un message d'interface DCL (remise à zéro de l'appareil), un message d'interface IFC (remise à zéro d'interface), ou pon (mise sous tension) soit reçu. L'**appareil** entrera en LIDS (état écouteur inactif) dans les 100 ms qui suivent l'entrée en AWNS (attente par l'accepteur d'un nouvel état de cycle) de la fonction d'établissement d'une liaison avec l'accepteur de l'**appareil**, sur le dernier octet du <MESSAGE DE PROGRAMMATION TERMINÉ> suivant:

*DLF<NL^END>

Où "DLF" peut se présenter sous la forme d'une combinaison quelconque de majuscules ou de minuscules. Cette exigence de temps ne s'applique pas si:

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Because these and other related common commands are optional, the application program must be able to determine if a particular **device** has implemented a given command or query. The application may derive this information from its own knowledge of a **device's** capabilities from a Resource Description. Another way is to send the command or query, and use the **device's** required status reporting facilities to determine if the **device** does not implement the command. For example, the application program could determine if the **device** implements the *RDT? common query with the following procedure:

NOTE - Quotes around data strings are not part of the data sent.

1) Send "*ESE 1;*SRE 32;*CLS<NL^END>". This <TERMINATED PROGRAM MESSAGE> enables reporting of the OPC event (only) and clears the status registers.

2) Send "*RDT?;*OPC<NL^END>". The **device** may or may not implement the *RDT? query, but is required to implement the *OPC (operation complete) command. The **device** will set the OPC event bit in the Standard Event Status Register when it executes the *OPC command. (An assumption is made that the **device** has no pending operations, which is a reasonable assumption during system initialization.) The **device** will eventually set the Event Status Bit (ESB, bit 5) in the **device's** Status Byte Register, and cause a service request.

3) The application may wait for the service request signal or may repeatedly poll the **device** until the **device's** status byte indicates a service request (RQS TRUE). (The application should not assume that a service request is caused by this **device**, but should always check the **device** status byte before continuing.)

4) Send "*ESR?<NL^END>" without attempting to read the *RDT? query response. Note that this sequence is a violation of the normal device message exchange protocol (see clause 6), and may cause the **device** to indicate a Query Error, which the application program is choosing to ignore.

5) Read the *ESR? response. The Operation Complete (OPC, bit 0) event bit should be TRUE (value of 1). The Command Error (CME, bit 5) event bit will be FALSE (value of 0) if the **device** does implement the *RDT? common query and TRUE (value of 1) if the **device** does not. The application program should ignore the Query Error (QYE, bit 2) event bit, which may or may not be set.

13.4 *Detailed requirements of the auto configure commands*

13.4.1 **DLF common command requirements*

The *DLF common command shall disable the **device's** normal Listener function until a DCL (Device Clear) interface message, an IFC (interface clear) interface message, or pon (power-on) is received. The **device** shall enter LIDS (Listener Idle State) within 100 ms after the **device's** acceptor handshake enters AWNS (Acceptor Wait for New Cycle State) on the last byte of the following <TERMINATED PROGRAM MESSAGE>:

*DLF<NL^END>

Where 'DLF' may be in any mixture of upper-case or lower-case. This timing requirement does not apply if:

IECNORM.COM : Click to view the full PDF of IEC 625-2:1993
Withdrawn

1) ce n'est pas la première <UNITÉ DE MESSAGE DE PROGRAMMATION> reçue après le message **dcas**,

2) si tout autre élément syntaxique est envoyé avec l'instruction *DLF avant le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>,

3) ou si tout <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> autre que <NL^END> est utilisé.

Si l'**appareil** reçoit une instruction *DLF qui n'est pas la première <UNITÉ DE MESSAGE DE PROGRAMMATION> reçue après le message **dcas**, ou qui n'est pas la seule <UNITÉ DE MESSAGE DE PROGRAMMATION> dans un <MESSAGE DE PROGRAMMATION>, il doit indiquer une erreur d'exécution et n'exécutera pas l'instruction *DLF.

L'**appareil** ne doit pas indiquer une erreur uniquement parce que le message *DLF est précédé ou suivi par un espace blanc ou parce qu'un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> autre que <NL^END> est utilisé, mais il peut ne pas répondre à l'exigence de temps.

Après avoir reçu *DLF, la fonction écouteur doit rester invalidée jusqu'à ce que l'**appareil** reçoive un message **dcas**. Pendant ce temps, la fonction parleur, le mécanisme d'indication d'état et la fonction contrôleur (si présents) peuvent continuer à fonctionner normalement ou peuvent être invalidés. L'**appareil** doit revenir en fonctionnement normal avec la fonction écouteur en LADS (état adressé écouteur) ou LACS (état actif écouteur) dans les 100 ms qui suivent la réception du message **dcas**. L'**appareil** doit également revenir en fonctionnement normal en LIDS (état inactif écouteur) après la mise sous tension ou le message d'interface IFC (remise à zéro de l'interface), mais ne doit pas revenir pas en fonctionnement normal en raison de toute autre condition de bus CEI 625-1.

13.4.2 *Prescriptions de l'instruction commune *AAD*

L'instruction commune *AAD prépare l'**appareil** à participer au protocole de sélection d'adresses du **contrôleur**. Le tampon d'entrée normale, le traitement des instructions, et les mécanismes d'indication d'état de l'**appareil** sont invalidés pendant le protocole.

13.4.2.1 Lancement du protocole de l'appareil

L'appareil doit être prêt à participer au protocole de sélection d'adresses dans les 100 ms qui suivent l'entrée en AWNS de la fonction d'établissement d'une liaison avec l'accepteur de l'appareil, sur le dernier octet du <MESSAGE DE PROGRAMMATION TERMINÉ> suivant:

*AAD<NL^END>

où "AAD" peut se présenter sous la forme d'une combinaison quelconque de majuscules ou de minuscules. L'exigence de temps ne s'applique pas si:

1) ce n'est pas la première <UNITÉ DE MESSAGE DE PROGRAMMATION> reçue après le message **dcas**,

2) tout autre élément syntaxique est envoyé avec l'instruction *AAD avant le <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION>,

3) ou si tout <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> autre que <NL^END> est utilisé.

IECNORM.COM : Click to view the full PDF of 625-2:1993
Withdrawn

1) this is not the first <PROGRAM MESSAGE UNIT> received after the **dcas** message,

2) if any other syntactic elements are sent with the *DLF command before the <PROGRAM MESSAGE TERMINATOR>,

3) or if any <PROGRAM MESSAGE TERMINATOR> other than <NL^END> is used.

If the **device** receives a *DLF command which is not the first <PROGRAM MESSAGE UNIT> received after the **dcas** message, or is not the only <PROGRAM MESSAGE UNIT> in a <PROGRAM MESSAGE>, it shall indicate an Execution Error and shall not execute the *DLF command.

The **device** shall not indicate an error solely because the *DLF is preceded or followed by white space or because a <PROGRAM MESSAGE TERMINATOR> other than <NL^END> is used, but it may not meet the timing requirement.

After receiving *DLF, the Listener function shall remain disabled until the **device** receives a **dcas** message. During this time, the Talker function, status reporting mechanism, and Controller function (if present) may continue normal operation or may be disabled. The **device** shall return to normal operation with the Listener function in LADS (Listener Addressed State) or LACS (Listener Active State) within 100 ms after receiving the **dcas** message. The **device** shall also return to normal operation in LIDS (Listener Idle State) following power-on or the IFC (Interface Clear) interface message, but shall not return to normal operation because of any other IEC 625-1 bus condition.

13.4.2 **AAD common command requirements*

The *AAD common command prepares the **device** to participate in the **controller's** Set Address protocol. The **device's** normal input buffering, command processing, and status-reporting mechanisms are disabled during the protocol.

13.4.2.1 Device protocol initiation

The **device** shall be ready to participate in the Set Address protocol within 100 ms after the **device's** Acceptor Handshake function enters AWNS on the last byte of the following <TERMINATED PROGRAM MESSAGE>:

*AAD<NL^END>

where 'AAD' may be in any mixture of upper-case or lower-case. This timing requirement does not apply if:

1) this is not the first <PROGRAM MESSAGE UNIT> received after the **dcas** message,

2) any other syntactic elements are sent with the *AAD command before the <PROGRAM MESSAGE TERMINATOR>,

3) or if any <PROGRAM MESSAGE TERMINATOR> other than <NL^END> is used.

IECNORM.COM : Click to view the full PDF of IEC 625-2:1993

Si l'**appareil** reçoit une instruction *AAD qui n'est pas la première <UNITÉ DE MESSAGE DE PROGRAMMATION> reçue après le message **dcas**, ou qui n'est pas la seule <UNITÉ DE MESSAGE DE PROGRAMMATION> dans un <MESSAGE DE PROGRAMMATION>, il doit indiquer une erreur d'exécution et ne doit pas exécuter l'instruction *AAD.

L'**appareil** ne doit pas indiquer une erreur uniquement parce que le message *AAD est précédé ou suivi par un espace blanc ou un <CARACTÈRE D'ARRÊT DE MESSAGE DE PROGRAMMATION> autre que <NL^END> est utilisé, mais il peut ne pas répondre à l'exigence de temps.

L'**appareil** doit revenir en fonctionnement normal à la fin du protocole de sélection d'adresses ou à la suite de pon (mise sous tension), du message d'interface IFC (remise à zéro de l'interface), ou du message **dcas**, mais ne doit pas revenir en fonctionnement normal en raison de toute autre condition de bus CEI 625-1. Si un **appareil** reçoit un message pon, IFC, ou **dcas** avant qu'une adresse lui soit affectée, il doit conserver l'adresse qu'il possédait avant l'exécution de cette instruction.

13.4.2.2 Identificateur d'appareil

Chaque **appareil** qui implémente cette instruction doit avoir un identificateur comportant quatre zones:

- fabricant
- numéro de modèle
- numéro de série
- adresse en cours

Pour un fonctionnement correct de cette instruction, chaque octet de l'identificateur doit avoir une valeur décimale supérieure à 31 et inférieure à 127, et l'identificateur doit être unique parmi tous les appareils dans le système. Les trois premières zones de cet identificateur doivent être identiques aux trois premières zones de la réponse à l'interrogation *IDN? (interrogation d'identification). La quatrième zone doit consister en un seul octet de données (valeur décimale 32 à 62) identique à l'adresse écouteur en cours de l'**appareil** et, si l'**appareil** implémente un adressage étendu, un autre octet de données (valeur décimale 96 à 126) identique à l'adresse secondaire en cours de l'**appareil**.

13.4.2.3 Fonctionnement du protocole

13.4.2.3.1 Vue d'ensemble

Pendant le protocole de sélection d'adresses, le **contrôleur** tente d'identifier chaque **appareil** d'adresse configurable à l'intérieur du système en recherchant un identificateur unique. Le **contrôleur** identifie un **appareil** à chaque passage par le protocole. L'ordre de la recherche est lexicographique, par zones, avec des caractères classés suivant leurs codes de 7 bits ISO. Si une zone dans un identificateur d'**appareil** est une sous-chaîne initiale de la même zone dans un autre identificateur d'**appareil**, l'**appareil** ayant la zone la plus longue sera trouvé en premier. En cours de passage, les appareils sont retirés de la recherche jusqu'à ce que seul le premier **appareil** (dans l'ordre ci-dessus) subsiste. En fin de passage, le **contrôleur** affecte une adresse à cet **appareil**. Une fois qu'une adresse lui a été affectée, un **appareil** ignore la circulation ultérieure sur le bus jusqu'à ce que tous les appareils aient été identifiés. Le **contrôleur** recommence alors la recherche jusqu'à ce que les appareils d'adresses configurables aient été identifiés et qu'on leur ait affecté de nouvelles adresses.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

If the **device** receives a *AAD command which is not the first <PROGRAM MESSAGE UNIT> received after the **dcas** message, or is not the only <PROGRAM MESSAGE UNIT> in a <PROGRAM MESSAGE>, it shall indicate an Execution Error and shall not execute the *AAD command.

The **device** shall not indicate an error solely because the *AAD is preceded or followed by white space or a <PROGRAM MESSAGE TERMINATOR> other than <NL^END> is used, but it may not meet the timing requirement.

The **device** shall return to normal operation at the end of the Set Address protocol or following pon (power-on), the IFC (Interface Clear) interface message, or the **dcas** message, but shall not return to normal operation because of any other IEC 625-1 bus condition. If a **device** receives a pon, IFC, or **dcas** prior to being assigned an address, it shall retain the address it possessed prior to the execution of this command.

13.4.2.2 Device identifier

Each **device** which implements this command shall have an identifier made up of four fields:

- manufacturer
- model number
- serial number
- current address

For proper operation of this command, each byte of the identifier must have a decimal value greater than 31 and less than 127, and the identifier must be unique among all devices in the system. The first three fields of this identifier shall be identical to the first three fields of the *IDN? (Identification Query) query response. The fourth field shall consist of a single data byte (decimal value 32 through 62) which is identical to the **device's** current listen address and, if the **device** implements extended addressing, another data byte (decimal 96 through 126) that is identical to the **device**'s current secondary address.

13.4.2.3 Protocol operation

13.4.2.3.1 Overview

During the Set Address protocol, the **controller** attempts to identify each address-configurable **device** within the system by searching for a unique identifier. The **controller** identifies one **device** in each pass through the protocol. The search ordering is lexicographic by fields, with characters ordered according to their ISO 7-bit codes. If a field in one **device's** identifier is an initial substring of the same field in another **device's** identifier, the **device** with the longer field will be found first. During the pass, devices drop out of the search until only the first **device** (in the above ordering) is left. At the end of the pass, the **controller** assigns an address to that **device**. Once assigned an address, a **device** ignores subsequent bus traffic until all devices have been identified. The **controller** then restarts the search with the remaining devices. This process continues until all address-configurable devices have been identified and assigned new addresses.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

13.4.2.3.2 Fonction de l'interface SR

Le **contrôleur** utilise une technique de recherche binaire pour trouver chaque octet dans l'identificateur. L'**appareil** utilise sa fonction d'interface SR pour guider le **contrôleur** dans la recherche. L'**appareil** devra envoyer le message rsv FAUX si la valeur de l'octet de données présenté par le **contrôleur** est inférieure ou égale à la valeur de l'octet dans l'identificateur de l'**appareil**. L'**appareil** devra envoyer le message rsv VRAI si la valeur de l'octet de données présenté par le **contrôleur** est plus grande que la valeur de l'octet dans l'identificateur de l'**appareil**. L'**appareil** devra envoyer le message rsv FAUX lorsque la recherche aura atteint la fin d'une zone de l'identificateur, ou lorsque la recherche aura atteint la fin de l'identificateur.

13.4.2.3.3 Exigences de temps

L'**appareil** devra envoyer la valeur correcte du message rsv en 1 ms de l'établissement d'une liaison avec l'accepteur de l'**appareil** entrant en AWNS. L'**appareil** ne devra pas passer de ANRS à ACRS (en préparation pour le prochain octet) avant que le message rsv ne soit stable.

13.4.2.3.4 Affectation d'adresse

Lorsque l'**appareil** aura été identifié, il devra accepter un octet de données du **contrôleur**. La valeur de cet octet est la nouvelle adresse écouteur de l'**appareil**. L'**appareil** devra régler son adresse primaire à la valeur des cinq bits de poids le plus faible de cet octet, et devra régler son adresse primaire parleur de sorte que les cinq bits de poids le plus faible de MLA et MTA soient identiques.

Un **appareil** utilisant l'adressage étendu devra accepter un octet de données supplémentaire du **contrôleur**. L'**appareil** devra régler son adresse secondaire à la valeur des cinq bits de poids le plus faible de cet octet.

La fonction écouteur de l'**appareil** ne devra pas passer en LIDS (état écouteur inactif) lorsque les nouvelles adresses auront été affectées, mais devra rester en LACS (état écouteur actif).

13.4.2.4 Algorithmes

Les algorithmes suivants définissent une implémentation correcte de l'instruction *AAD. L'**appareil** devra implémenter ces algorithmes ou une procédure fonctionnellement équivalente.

13.4.2.4.1 Procédure d'acceptation d'adresse

L'acceptation d'adresse attend le caractère STX, caractère de 7 bits ISO avec la valeur 02 (2 en décimal), qui marque le début d'une séquence de recherche. La "présentation d'identificateur d'appareil" est appelée pour présenter l'identificateur de l'**appareil**. Si l'**appareil** n'est pas le premier dans l'ordre de recherche, la "présentation d'identificateur d'appareil" revient avec la signalisation d'appareil détecté FAUSSE. Si la signalisation d'appareil détecté est FAUSSE, l'acceptation d'adresse continue à lire et à rejeter des octets de données jusqu'à ce qu'un caractère STX soit reçu, indiquant le départ d'une nouvelle séquence de recherche. Si la signalisation d'appareil détecté est VRAIE, cet **appareil** a été identifié. L'acceptation d'adresse lit et rejette toutes les données jusqu'à ce qu'un message **dcas** rétablisse le fonctionnement normal.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

13.4.2.3.2 SR interface function

The **controller** uses a binary search technique to find each byte in the identifier. The **device** uses its SR interface function to guide the **controller** through the search. The **device** shall send the rsv message FALSE if the value of the data byte presented by the **controller** is less than or equal to the value of the byte in the **device's** identifier. The **device** shall send the rsv message TRUE if the value of the data byte presented by the **controller** is greater than the value of the byte in the **device's** identifier. The **device** shall send the rsv message FALSE when the search has reached the end of an identifier field, or when the search has reached the end of the identifier.

13.4.2.3.3 Timing requirements

The **device** shall send the correct value of the rsv message within 1 ms of the **device's** acceptor handshake entering AWNS. The **device** shall not make a transition from ANRS to ACRS (in preparation for the next byte) until the rsv message is stable.

13.4.2.3.4 Address assignment

When the **device** has been identified, it shall accept one data byte from the **controller**. The value of this byte is the **device's** new listen address. The **device** shall set its primary address to the value of the lower five bits of this byte, and shall set its primary talk address so that the lower five bits of MLA and MTA are identical.

A **device** utilizing extended addressing shall accept an additional data byte from the **controller**. The **device** shall set its secondary address to the value of the lower five bits of this byte.

The **device's** Listener function shall not enter LIDS (Listener Idle State) when the new addresses are assigned, but shall remain in LACS (Listener Active State).

13.4.2.4 Algorithms

The following algorithms define a correct implementation of the *AAD command. The **device** shall implement these algorithms or a functionally equivalent procedure.

13.4.2.4.1 Accept-Address procedure

Accept-Address watches for the STX character, ISO 7-bit character with the value 02 (2 decimal), which marks the start of a search pass. Present-Device-Identifier is called to present the **device's** identifier. If the **device** is not first in the search order, Present-Device-Identifier returns with the device-detected flag set FALSE. If the device-detected flag is FALSE, Accept-Address continues reading and discarding bytes until another STX character is received, indicating the start of a new search pass. If the device-detected flag is TRUE, this **device** has been identified. Accept-Address reads and discards all data until a **dcas** message restores normal operation.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

```
BEGIN acceptation d'adresse

   Invalidation de la mémoire tampon d'entrée
   Invalidation du traitement d'instructions normales
   Invalidation de l'indication d'état de l'appareil
   Mettre le message rsv FAUX

   Mettre l'appareil détecté FAUX
   WHILE appareil détecté est FAUX
    BEGIN
      REPEAT
       Lire l'octet de données
     UNTIL octet de données est de 7 bits ISO STX (*valeur décimale de 2*)
      Exécuter "présentation identificateur d'appareil"
      (*La "présentation identificateur d'appareil" mettra l'appareil détecté VRAI
      si le contrôleur a identifié avec succès cet appareil*)
    END

   WHILE l'on n'est pas en DCAS
     Lire et rejeter les octets de données

   Valider la mémoire tampon d'entrée
   Valider le traitement d'instructions normales
   Valider l'indication d'état de l'appareil

END acceptation d'adresse
```

13.4.2.4.2 Procédure "présentation identificateur d'appareil"

La "présentation identificateur d'appareil" présente les quatre zones de l'identificateur de l'**appareil** au **contrôleur**. Si les quatre zones correspondent, la "présentation identificateur d'appareil" accepte la nouvelle adresse de l'**appareil** venant du **contrôleur**. Si l'une des zones ne convient pas, la "présentation identificateur d'appareil" revient immédiatement.

```
   BEGIN "présentation identificateur d'appareil"
     Sélectionner la zone en cours du fabricant
     Mettre participation VRAIE
     WHILE participation est VRAIE
       BEGIN
         Exécuter la "présentation zone d'identificateur"
         (*La "présentation zone d'identificateur" mettra participation FAUSSE
           s'il y a sur le bus un autre appareil dont l'identificateur vient avant
           l'identificateur de cet appareil dans l'ordre de la recherche.*)
        IF participation est VRAIE
          THEN
            BEGIN
              IF zone en cours est adressée
               THEN
                 BEGIN
                   Mettre appareil détecté VRAI
                   Mettre participation FAUSSE
                 END
               ELSE
```

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

```
BEGIN Accept-Address

   Disable input buffer
   Disable normal command processing
   Disable device status reporting
   Set rsv message FALSE

   Set device-detected to FALSE
   WHILE device-detected is FALSE
     BEGIN
       REPEAT
         Read data-byte
       UNTIL data-byte is ISO 7-bit STX (*decimal value of 2*)
       Perform Present-Device-Identifier
       (*Present-Device-Identifier will set device-detected TRUE if
       the controller has successfully identified this device*)
     END

   WHILE NOT in DCAS
     Read and discard data bytes

   Enable input buffer
   Enable normal command processing
   Enable device status reporting
END Accept-Address
```

13.4.2.4.2 Present-Device-Identifier procedure

Present-Device-Identifier presents the four fields of the **device's** identifier to the **controller**. If all four fields match, Present-Device-Identifier accepts the **device's** new address from the **controller**. If any field does not match, Present-Device-Identifier returns immediately.

```
BEGIN Present-Device-Identifier
   Set current-field to manufacturer
   Set participating TRUE
   WHILE participating is TRUE
     BEGIN
       Perform Present-Identifier-Field
       (*Present-Identifier-Field will set participating FALSE if there is
         another device on the bus whose identifier comes prior
         to this device's identifier in the search order.*)
       IF participating is TRUE
         THEN
           BEGIN
             IF current-field is address
             THEN
               BEGIN
                 Set device-detected to TRUE
                 Set participating to FALSE
               END
             ELSE
```

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

```
                CASE zone en cours OF
                  BEGIN
                    Fabricant: Mettre la zone en cours au numéro de modèle
                    Numéro de modèle: Mettre la zone en cours au numéro de série
                    Numéro de série: Mettre la zone en cours à l'adresse
                  END
            END
      END
  IF appareil détecté est VRAI
   THEN
    BEGIN
      Lire l'octet de données (*signale fin d'identificateur par SRQ FAUX*)
      IF octet de données est 127
       THEN
        BEGIN
          Lire l'octet de données
          Prendre l'adresse primaire indiquée par l'octet de données
          IF appareil implémente l'adressage étendu
           THEN
            BEGIN
              Lire l'octet de données
              Prendre l'adresse secondaire indiquée par l'octet de données
            END
        END
       ELSE
        Mettre appareil détecté FAUX
        (*Un autre appareil a une zone d'adresses qui est un sous-ensemble
        de la zone d'adresses de cet appareil*)
      END
  END "présentation identificateur d'appareil"
```

13.4.2.4.3 Procédure "présentation zone d'identificateur"

La "présentation zone d'identificateur" présente des octets successifs de la zone de recherche en cours au **contrôleur**.

```
  BEGIN "Présentation zone d'identificateur"
    Mettre le pointeur d'octet de l'identificateur au commencement de la zone en cours
    Mettre fin de zone FAUSSE
  WHILE participation est VRAIE ET fin de zone est FAUSSE
    BEGIN
      Exécuter la "présentation octet d'identificateur"
      (*La "présentation octet d'identificateur" mettra participation FAUSSE
        s'il y a un autre appareil sur le bus dont l'identificateur vient avant
        l'identificateur de cet appareil dans l'ordre de la recherche.*)
        IF le pointeur d'octet de l'identificateur est à la fin de la zone courante
        THEN
        Mettre fin de zone VRAIE
      ELSE
        Incrémenter le pointeur d'octet de l'identificateur
    END
```

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

```
                CASE current-field OF
                  BEGIN
                    manufacturer: Set current-field to model number
                    model number: Set current-field to serial number
                    serial number: Set current-field to address
                  END
            END
      END
  IF device-detected is TRUE
   THEN
    BEGIN
      Read data-byte (*Signals end of identifier through SRQ being FALSE*)
      IF data-byte is 127
       THEN
        BEGIN
          Read data-byte
          Assume primary address indicated by data-byte
          IF device implements extended addressing
           THEN
            BEGIN
              Read data-byte
              Assume secondary address indicated by data-byte
            END
        END
        ELSE
          Set device-detected to FALSE
          (*Another device has an address field which is a superset
          of this device's address field*)
      END
  END Present-Device-Identifier
```

13.4.2.4.3 Present-Identifier-Field procedure

Present-Identifier-Field presents successive bytes of the current search field to the **controller**.

```
  BEGIN Present-Identifier-Field
    Set identifier-byte-pointer to beginning of current-field
    Set end-of-field to FALSE
  WHILE participating is TRUE AND end-of-field is FALSE
    BEGIN
      Perform Present-Identifier-Byte
      (*Present-Identifier-Byte will set participating FALSE if
        there is another device on the bus whose identifier
        comes prior to this device's identifier in the search order.*)
      IF identifier-byte-pointer is at end of current-field
        THEN
        Set end-of-field to TRUE
       ELSE
        Increment identifier-byte-pointer
    END
```

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

```
IF participation est VRAIE
    THEN
      BEGIN
       (*Signale la fin de cette zone en renvoyant SRQ FAUX
          pour la recherche du prochain caractère.*)
       Lire l'octet de données
       Mettre le message rsv FAUX
      END
END "présentation zone d'identificateur"
```

13.4.2.4.4 Procédure "présentation octet d'identificateur"

La "présentation octet d'identificateur" présente un octet de l'identificateur de l'**appareil** au **contrôleur**. Elle utilise la fonction d'interface SR pour guider le **contrôleur** dans la recherche binaire de l'octet de l'identificateur de plus faible valeur dans le système. Le **contrôleur** signale la fin de la recherche en présentant l'octet de l'identificateur avec le groupe de bits de poids le plus fort. La "présentation octet d'identificateur" vérifie que cet octet correspond à l'octet d'identificateur de l'**appareil**. La signalisation de participation est mise FAUSSE si l'octet ne correspond pas. Cette non-correspondance entraînera un retour à l'acceptation d'adresses, qui rejettera les octets de données jusqu'à la prochaine séquence.

```
BEGIN "présentation octet d'identificateur"
  Lire l'octet de données
  IF octet de données est 127
    THEN
      BEGIN
        Mettre le message rsv VRAI
        Mettre détection de FI08 FAUSSE
        WHILE détection DIO8 est FAUSSE
          BEGIN
           Lire l'octet de données
           IF octet de données est entre 0 et 127
             THEN
               BEGIN
                (*Le contrôleur est guidé dans la recherche binaire par
                l'appareil avec l'octet de plus petite valeur à la position
                courante dans l'identificateur.*)
                IF l'octet de données est PLUS GRAND QUE l'octet d'identificateur en cours
                   THEN
                     Mettre le message rsv VRAI
                   ELSE
                     Mettre le message rsv FAUX
                END
             ELSE
                BEGIN
                 (*Le contrôleur a mis le bit de poids le plus fort de l'octet
                 pour indiquer que la recherche est faite et que cette valeur
                 est l'octet de plus faible valeur dans cette position.
                 Il place cet octet sur le bus pour que l'appareil puisse
                 déterminer s'il devrait continuer de participer à la recherche.*)
               Mettre le message rsv FAUX
```

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

```
IF participating is TRUE
    THEN
      BEGIN
       (*Signals end of this field by returning SRQ FALSE for the
          search of the next character.*)
       Read data-byte
       Set rsv message FALSE
    END
END Present-Identifier-Field
```

13.4.2.4.4 Present-Identifier-Byte procedure

Present-Identifier-Byte presents one byte of the **device's** identifier to the **controller**. It uses the SR interface function to guide the **controller** through a binary search for the lowest-value identifier byte in the system. The **controller** signals the end of the search by presenting the identifier byte with the most significant bit set. Present-Identifier-Byte checks that this byte matches the **device's** identifier byte. The participating flag is set FALSE if the byte does not match. This mismatch will cause a return to Accept-Address, which will discard data bytes until the next pass.

```
BEGIN Present-Identifier-Byte
  Read data-byte
  IF data-byte is 127
    THEN
      BEGIN
        Set rsv message TRUE
        Set DIO8-detect FALSE
        WHILE DIO8-detect is FALSE
          BEGIN
           Read data-byte
           IF data-byte is between 0 and 127
             THEN
               BEGIN
                (*The controller is guided through the binary search
                by the device with the lowest-value byte at
                the current position within the identifier.*)
                IF data-byte is GREATER THAN current identifier byte
                  THEN
                    Set rsv message TRUE
                  ELSE
                    Set rsv message FALSE
                END
             ELSE
               BEGIN
                (*The controller has set the most significant bit of the byte
                to indicate that it is done searching and this is the value
                of the lowest valued byte in this position.
                It is placing that byte on the bus so that the device can
                determine if it should continue participating in the search.*)
              Set rsv message FALSE
```

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993

```
                Mettre la détection de DI08 VRAIE
                IF l'octet de données n'est PAS ÉGAL à
                    (octet de l'identificateur en cours + 128)
                    THEN
                      Mettre participation FAUSSE
                END
              END
            END
          ELSE (*Une zone de super-ensemble, à l'exclusion de l'adresse, d'un autre
                appareil est détectée*)
            BEGIN
              Mettre le message rsv FAUX
              Mettre participation FAUSSE
            END

    END "présentation octet d'identificateur"
```

13.5 *Techniques de configuration automatique supplémentaires*

Les adresses et les identifications des appareils du système de test ne représentent qu'une partie de l'information dont a besoin le **contrôleur** afin de réaliser les tâches exigées par l'utilisateur. C'est-à-dire que les dates d'étalonnage, les codes de programmation, etc., sont exigés par le **contrôleur** pour garantir que les exigences de test de l'utilisateur sont transformées par le système de test en actions valides.

Les techniques de base fournies dans cette section peuvent être utilisées par le programmeur d'application pour faciliter la tâche de configuration et de reconfiguration automatique des systèmes de test. Cette section décrit seulement quelques-unes des techniques de configuration possibles. Par exemple, le tableau de configuration du système peut être étendu pour comprendre les masques de demande de service qui sont automatiquement envoyés aux appareils lorsque le tableau est créé ou remis à jour.

13.6 *Exemples*

Voir annexe C, configuration automatique de système, pour exemples d'application.

SECTION 4: SPÉCIFICATIONS RELATIVES AU CONTRÔLEUR

14 Critères de conformité du contrôleur

Un **appareil** doit avoir certaines capacités, comme cela est décrit dans les articles précédents. Le présent article fait l'inventaire des capacités qu'exige cette norme dans un **contrôleur de système**. Un **contrôleur** peut facultativement contenir des capacités supplémentaires. Toutes les capacités facultatives qui sont décrites par cette norme sont aussi répertoriées.

La conformité pour un **contrôleur** est divisée en plusieurs zones qui sont considérées séparément. Un **contrôleur** doit satisfaire toute la fonctionnalité requise dans chaque zone afin de se conformer à cette norme.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

```
            Set DIO8-detect TRUE
            IF data-byte is NOT EQUAL to
            (current identifier byte + 128)
                THEN
                  Set participating FALSE
            END
          END
        END
      ELSE (*A superset field, excluding address, of another device is
            detected*)
        BEGIN
          Set rsv message FALSE
          Set participating FALSE
        END

  END Present-Identifier-Byte
```

13.5 *Additional automatic configuration techniques*

Addresses and identifications of test system devices represent only a portion of the information the **controller** needs in order to perform tasks requested by the user. That is, calibration dates, programming codes, etc., are required by the **controller** to ensure that the test requirements of the user are transformed into valid actions by the test system.

The basic techniques provided in this section may be utilized by the application programmer to ease the task of configuring and reconfiguring automatic test systems. This section describes only a few of the configuration techniques that are possible. For example, the system configuration table can be expanded to include service request masks that are automatically sent to the devices when the table is created or updated.

13.6 *Examples*

See annex C, Automatic system configuration, for application examples.

SECTION 4: CONTROLLER RELATED SPECIFICATIONS

14 Controller compliance criteria

A **device** must have certain capabilities as described in previous clauses. This clause lists the capabilities which this standard requires in a **system controller**. A **controller** may optionally contain additional capabilities. Any optional **controller** capabilities which are described by this standard are also listed.

Compliance for a **controller** is divided into several areas which are considered separately. A **controller** must satisfy all the required functionality in each of the areas in order to comply with this standard.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

14.1 *Prescriptions de la CEI 625-1*

Un **contrôleur** devra contenir ces sous-ensembles CEI 625-1 énumérés dans le tableau 14-1 et aucun autre:

Tableau 14-1 – Prescriptions de la CEI 625-1

Fonction d'interface CEI 625-1	Sous-ensembles CEI 625-1	Paragraphe CEI 625-2
Etablissement d'une liaison avec la source	SH1	15.1.2
Etablissement d'une liaison avec l'accepteur	AH1	15.1.3
Parleur	si C5, C7, C9, C11, C17, C19, C21, ou C23 alors T5, T6, TE5, ou TE6 ou bien T7, T8, TE7, ou TE8	15.1.2
Ecouteur	L3, L4, LE3, ou LE4	15.1.3
Demande de service	si C5, C7, C9, C11, C17, C19, C21, ou C23 alors SR1 ou bien SR0	
Eloigné local	RL0 ou RL1	
Interrogation parallèle	PP0 ou PP1	
Effacement d'appareil	DC0 ou DC1	
Déclenchement d'appareil	DT0 ou DT1	
Contrôleur		15.1.1
Contrôleur de système	C1	
Envoyer IFC et prendre en charge	C2	
Envoyer REN	C3	
Répondre à SRQ	C4	
Envoyer des messages d'interface prendre le contrôle synchroniquement	un parmi C5, C7, C9, C11, C13, C15, C17, C19, C21, C23, C25, ou C27	

Un **contrôleur** devra répondre aux prescriptions de la norme CEI 625-1. Il devra aussi répondre à toutes les prescriptions présentées à l'article 15 de cette norme.

14.2 *Prescriptions des échanges de messages*

14.2.1 *Séquences de contrôle exigées*

Chacune des séquences de contrôle énumérées dans le tableau 14-2 et décrites à l'article 16 devra être implémentée individuellement dans un **contrôleur**.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

14.1 *IEC 625-1 requirements*

A **controller** shall contain these IEC 625-1 subsets listed in table 14-1 and no others:

Table 14-1 – IEC 625-1 requirements

IEC 625-1 Interface function	IEC 625-1 Subsets	IEC 625-2 Subclause
Source Handshake	SH1	15.1.2
Acceptor Handshake	AH1	15.1.3
Talker	if C5, C7, C9, C11, C17, C19, C21, or C23 then T5, T6, TE5, or TE6 else T7, T8, TE7, or TE8	15.1.2
Listener	L3, L4, LE3, or LE4	15.1.3
Service Request	if C5, C7, C9, C11, C17, C19, C21, or C23 then SR1 else SR0	
Remote Local	RL0 or RL1	
Parallel Poll	PP0 or PP1	
Device Clear	DC0 or DC1	
Device Trigger	DT0 or DT1	
Controller		15.1.1
System Controller	C1	
Send IFC and Take Charge	C2	
Send REN	C3	
Respond to SRQ	C4	
Send Interface Messages and Take Control Synchronously	one of C5, C7, C9, C11, C13, C15, C17, C19, C21, C23, C25, or C27	

A **controller** shall meet the requirements of IEC 625-1. It shall also meet all requirements stated in clause 15 of this standard.

14.2 *Message Exchange requirements*

14.2.1 *Required control sequences*

Each of the control sequences listed in table 14-2 and described in clause 16, shall be implemented individually in a **controller**.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Tableau 14-2 – Séquences de contrôle exigées

Séquence de contrôle	Paragraphes	Exige le sous-ensemble CEI 625-1
ENVOI D'INSTRUCTION	16.2.1	un parmi C5-C28
ENVOI DE CONFIGURATION	16.2.2	un parmi C5-C28
ENVOI D'OCTETS DE DONNÉES	16.2.3	un parmi C5-C28
ENVOI	16.2.4	un parmi C5-C28
RÉCEPTION DE CONFIGURATION	16.2.5	un parmi C5-C28
RÉCEPTION D'UN MESSAGE DE RÉPONSE	16.2.6	un parmi C5-C28
RÉCEPTION	16.2.7	un parmi C5-C28
ENVOI DE IFC	16.2.8	C2
APPAREIL REMIS À ZÉRO	16.2.9	un parmi C5-C28
VALIDATION DES CONTRÔLES LOCAUX	16.2.10	C3
VALIDATION ÉLOIGNÉE	16.2.11	C3
SÉLECTION DE RWLS	16.2.12	C3 et un parmi C5-C28
ENVOI DE LLO	16.2.13	C3 et un parmi C5-C28
LECTURE DE L'OCTET D'ÉTAT	16.2.18	un parmi C5-C28
DÉCLENCHEMENT	16.2.19	un parmi C5-C28

14.2.2 *Séquences de contrôle facultatives*

Chacune des séquences de contrôle énumérées dans le tableau 14-3 et décrites à l'article 16 peut être implémentée individuellement dans un **contrôleur**.

Tableau 14-3 – Séquences de contrôle facultatives

Séquence de contrôle	Paragraphe	Exige le sous-ensemble CEI 625-1
PASSAGE DE CONTRÔLE	16.2.4	C5, C7, C9, ou C11
RÉALISER UNE INTERROGATION PARALLÈLE	16.2.5	C5, C9, C13, C17, C21, ou C25
CONFIGURATION D'INTERROGATION PARALLÈLE	16.2.6	C5, C9, C13, C17, C21, ou C25
NON-CONFIGURATION D'INTERROGATION PARALLÈLE	16.2.7	C5, C9, C13, C17, C21, ou C25

14.3 *Protocoles*

14.3.1 *Protocoles exigés*

Chacun des protocoles de **contrôleur** énumérés dans le tableau 14-4 et décrits à l'article 17 doivent être implémentés individuellement dans un **contrôleur**:

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Table 14-2 – Required control sequences

Control sequence	Subclause	Requires IEC 625-1 Subset
SEND COMMAND	16.2.1	one of C5-C28
SEND SETUP	16.2.2	one of C5-C28
SEND DATA BYTES	16.2.3	one of C5-C28
SEND	16.2.4	one of C5-C28
RECEIVE SETUP	16.2.5	one of C5-C28
RECEIVE RESPONSE MESSAGE	16.2.6	one of C5-C28
RECEIVE	16.2.7	one of C5-C28
SEND IFC	16.2.8	C2
DEVICE CLEAR	16.2.9	one of C5-C28
ENABLE LOCAL CONTROLS	16.2.10	C3
ENABLE REMOTE	16.2.11	C3
SET RWLS	16.2.12	C3 and one of C5-C28
SEND LLO	16.2.13	C3 and one of C5-C28
READ STATUS BYTE	16.2.18	one of C5-C28
TRIGGER	16.2.19	one of C5-C28

14.2.2 *Optional control sequences*

Each of the control sequences listed in table 14-3 and described in clause 16, may be implemented individually in a **controller**.

Table 14-3 – Optional control sequences

Control sequence	Subclause	Requires IEC 625-1 Subset
PASS CONTROL	16.2.4	C5, C7, C9, or C11
PERFORM PARALLEL POLL	16.2.5	C5, C9, C13, C17, C21, or C25
PARALLEL POLL CONFIGURE	16.2.6	C5, C9, C13, C17, C21, or C25
PARALLEL POLL UNCONFIGURE	16.2.7	C5, C9, C13, C17, C21, or C25

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

14.3 *Protocols*

14.3.1 *Required protocols*

Each of the **controller** protocols listed in table 14-4 and described in clause 17, shall be implemented individually in a **controller**:

Tableau 14-4 – Protocoles exigés

Mot clé du protocole	Paragraphes
RESET	17.1
ALLPOLL	17.3

14.3.2 *Protocoles facutatifs*

Les protocoles de **contrôleur** énumérés dans le tableau 14-5 et décrits à l'article 17 peuvent être implémentés dans un **contrôleur**:

Tableau 14-5 – Protocoles facultatifs

Mot clé du protocole	Paragraphes
FINDRQS	17.2
PASSCTL	17.4
REQUESTCTL	17.5
FINDLSTN	17.6
SETADD	17.7
TESTSYS	17.8

14.4 *Traitement des éléments fonctionnels*

Un **contrôleur** devra avoir la capacité d'envoyer tous les éléments syntaxiques fonctionnels CEI 625-2 décrits à l'article 7. Un **contrôleur** devra avoir la capacité de recevoir tous les éléments syntaxiques fonctionnels CEI 625-2 décrits à l'article 8.

14.5 *Prescriptions de spécifications du contrôleur*

Des informations sur la façon dont le fabricant a implémenté les exigences du **contrôleur** de la présente norme doivent être fournies avec la documentation du **contrôleur** par le fabricant. Ces informations doivent comprendre:

1) Une liste des sous-ensembles implémentés de la fonction d'interface CEI 625-1, l'article 15.

2) Une liste des séquences de contrôle qui peuvent être exécutées par le **contrôleur**, voir 16.2.

3) Une liste des protocoles qui peuvent être exécutés par le **contrôleur**, l'article 17.

4) Une description de la façon dont les divers éléments fonctionnels sont envoyés et reçus par le **contrôleur**, voir 14.4.

5) Toute limitation de dimensions du tampon devra être décrite, voir 7.7.6 et 8.7.9.

6) Une description du fonctionnement des délais d'attente, voir 15.2 et 15.3.2, y compris:

- comment les délais d'attente sont mis en service et hors service,
- si la temporisation est effectuée sur la base d'un octet ou d'un message,
- les valeurs admissibles de temps pour le délai d'attente.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

Table 14-4 – Required protocols

Protocol keyword	Subclause
RESET	17.1
ALLSPOLL	17.3

14.3.2 *Optional protocols*

The **controller** protocols listed in table 14-5 and described in clause 17, may be implemented in a **controller**:

Table 14-5 – Optional protocols

Protocol keyword	Subclause
FINDRQS	17.2
PASSCTL	17.4
REQUESTCTL	17.5
FINDLSTN	17.6
SETADD	17.7
TESTSYS	17.8

14.4 *Functional element handling*

A **controller** shall have the capability of sending all the IEC 625-2 functional syntactic elements described in clause 7. A **controller** shall have the capability to receive all the IEC 625-2 functional syntactic elements described in clause 8.

14.5 *Controller specification requirements*

Information about how the manufacturer implemented **controller** requirements of this standard shall be supplied with the **controller** documentation by the manufacturer. This information shall include:

1) A list of IEC 625-1 Interface Function subsets implemented, clause 15.

2) A list of control sequences which can be performed by the **controller**, see 16.2.

3) A list of protocols which can be performed by the **controller**, clause 17.

4) A description of how the various functional elements are sent and received by the **controller**, see 14.4.

5) Any buffer size limitations shall be described, see 7.7.6 and 8.7.9.

6) A description of how timeouts work, see 15.2 and 15.3.2, including:

- how timeouts are turned on and off,
- if the timing is done on a byte or message basis,
- the allowable time values for the timeout.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

15 Prescriptions du contrôleur CEI 625-2

15.1 *Prescriptions de la fonction d'interface du contrôleur*

Cet article décrit les prescriptions de la fonction d'interface CEI 625-1 d'un **contrôleur**. Elle spécifie également les prescriptions supplémentaires d'un **contrôleur** qui sont directement associées aux dix fonctions d'interface CEI 625-1. Ces prescriptions sont destinées à compléter la spécification CEI 625-1 pour l'environnement du système d'instruments, comme décrit dans cette norme.

15.1.1 *Prescriptions du contrôleur*

Le **contrôleur** devra avoir les capacités de fonction de contrôleur suivantes: contrôleur de système (C1), envoyer IFC et prendre en charge (C2), envoyer REN (C3), répondre à SRQ (C4), envoyer des messages d'interface (un parmi C5-C27), et prendre contrôle synchroniquement (un parmi C5, C7, C9, C11, C13, C15, C17, C19, C21, C23, C25, C27). Contrôle de réception (un parmi C5-C15), passage de contrôle (un parmi C5-C11 ou C17 C23), passage de contrôle à lui-même (un parmi C5, C7, C17, C19), et interrogation parallèle (un parmi C5, C9, C13, C17, C21, C25) sont des capacités facultatives.

15.1.2 *Prescriptions du parleur*

Pour une transmission de message spécifique de l'appareil CEI 625-2 le **contrôleur** devra contenir un des sous-ensembles parleur T5-T8 ou TE5-TE8. Ces sous-ensembles donnent un mandat sans adresse au parleur de base dans le cas de MLA. La capacité facultative comprend la réponse à une interrogation série et la possession d'un mode parole seulement. La capacité d'établissement d'une liaison avec la source complète (SH1) est aussi exigée pour soutenir le sous-ensemble parleur.

15.1.3 *Prescriptions de l'écouteur*

Pour une réception de message dépendant de l'appareil CEI 625-2 le **contrôleur** devra contenir un des sous-ensembles écouteur L3-L4 ou LE3-LE4. Ces sous-ensembles donnent un mandat sans adresse à l'écouteur de base dans le cas de MTA avec la capacité facultative de contenir un mode écouteur seulement. La capacité d'établissement d'une liaison avec l'accepteur complète (AH1) est aussi exigée pour soutenir le sous-ensemble écouteur.

15.1.4 *Prescriptions de passage de contrôle*

Si on attend qu'un **contrôleur** exécute le protocole décrit dans 17.4 et 17.5 pour le passage de contrôle et le retour de contrôle, il devra avoir la capacité de sous-ensemble de "passage de contrôle" et "prise de contrôle synchroniquement". C'est-à-dire que les sous-ensembles C1, C2, C3 et C4 sont exigés avec le choix de C5, C7, C9 ou C11. Un **contrôleur** pouvant faire un passage de contrôle devra aussi contenir les sous-ensembles de fonction parleur T5, T6, TE5 ou TE6 qui soutiennent l'interrogation série.

15.1.5 *Prescriptions électriques*

Un **contrôleur** devra implémenter l'option d'interface électrique E2 (CEI 625-1, annexe C). Les circuits à collecteurs ouverts devront être utilisés pour attaquer les lignes de signaux SRQ, NRFD et NDAC. Les circuits à trois états devront être utilisés pour attaquer les lignes de signaux DAV, EOI, ATN, REN et IFC. Lorsqu'un appareil CEI 625-1 dans un système CEI 625-1 n'est pas en PPAS, le **contrôleur** devra utiliser des circuits à trois états pour attaquer les lignes de signaux DI01-8. Si un appareil CEI 625-1 du système CEI 625-1

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

15 IEC 625-2 controller requirements

15.1 *Controller interface function requirements*

This clause describes the IEC 625-1 interface function requirements of a **controller**. It also specifies the additional requirements of a **controller** that are directly associated with the ten IEC 625-1 interface functions. These requirements are intended to supplement the IEC 625-1 specification for the instrument system environment as described in this standard.

15.1.1 *Controller requirements*

The **controller** shall have the following Controller function capabilities: system controller (C1), send IFC and take charge (C2), send REN (C3), respond to SRQ (C4), send interface messages (one of C5-C27), and take control synchronously (one of C5, C7, C9, C11, C13, C15, C17, C19, C21, C23, C25, C27). Receive control (one of C5-C15), pass control (one of C5-C11 or C17-C23), pass control to self (one of C5, C7, C17, C19), and parallel poll (one of C5, C9, C13, C17, C21, C25) are optional capabilities.

15.1.2 *Talker requirements*

For IEC 625-2 device-specific message transmission, the **controller** shall contain one of the Talker subsets T5-T8 or TE5-TE8. These subsets mandate the basic talker with unaddress if MLA. Optional capability includes responding to a serial poll and having a talk-only mode. Complete Source Handshake (SH1) capability is also required to support the Talker subset.

15.1.3 *Listener requirements*

For IEC 625-2 device-specific message reception, the **controller** shall contain one of the Listener subsets L3-L4 or LE3-LE4. These subsets mandate the basic listener with unaddress if MTA with optional capability to contain a listen-only mode. Complete Acceptor Handshake (AH1) capability is also required to support the Listener subset.

15.1.4 *Passing control requirements*

If a **controller** is expected to execute the protocol described in clauses 17.4 and 17.5 for passing and returning control, it shall have the subset capability to 'Pass Control' and 'Take Control Synchronously'. That is, subsets C1, C2, C3 and C4 are required along with a choice of either C5, C7, C9 or C11. A **controller** which can pass control shall also contain the T5, T6, TE5, or TE6 Talker function subsets that support serial poll.

15.1.5 *Electrical requirements*

A **controller** shall implement the E2 electrical interface option (IEC 625-1, annex C). Open collector drivers shall be used to drive the SRQ, NRFD and NDAC signal lines. Three-state drivers shall be used to drive the DAV, EOI, ATN, REN, and IFC signal lines. When an IEC 625-1-device in an IEC 625-1 system is not in PPAS, the **controller** shall use three-state drivers to drive the DIO1-8 signal lines. If an IEC 625-1-device in the IEC 625-1 system is in PPAS, the **controller** shall use open collector drivers or the high-impedance state

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

est en PPAS, le **contrôleur** devra utiliser des circuits à collecteurs ouverts ou l'état haute impédance des circuits à trois états pour attaquer les lignes de signaux DI01-8 convenant pour le fonctionnement en cours du **contrôleur**.

15.2 *Prescriptions supplémentaires du contrôleur CEI 625-2*

Un **contrôleur** devra avoir:

1) un moyen de fournir un contrôle de bus de bas niveau par un logiciel d'application pour permettre indépendamment à l'utilisateur:
   - d'émettre des impulsions IFC VRAI pendant plus de 100 microsecondes,
   - de mettre les lignes de signaux REN soit VRAI, soit FAUX,
   - d'envoyer séparément, ou dans n'importe quelle combinaison, tout message d'interface défini dans CEI 625-1,
   - d'envoyer et de détecter le message END CEI 625-1 (c'est-à-dire la ligne de signaux E01 VRAIE et la ligne de signaux ATN FAUSSE liées avec un octet de données);
2) la facilité (soit directement, soit par un logiciel d'application) de rentrer et de sortir les codes, formats, protocoles et instructions communes, comme défini dans cette norme.
3) la facilité de détecter l'état (VRAI ou FAUX) de la ligne SRQ.
4) la facilité de détecter les transitions de FAUX à VRAI de la ligne SRQ. La durée des actions du programme d'application suivant la transition est au-delà de l'étendue de cette norme.
5) la facilité (soit directement, soit par un programme d'application) d'examiner l'octet d'état sur la base d'un bit.
6) la facilité de détecter une condition d'erreur du **contrôleur** tentant d'établir une liaison de source avec un octet alors que tous les autres appareils sont en AIDS et relayent l'erreur jusqu'au programme d'application.
7) la facilité de temporiser l'échange de messages de contrôleur à appareil et d'appareil à contrôleur et avoir la capacité de relayer le déroulement d'une temporisation jusqu'au programme d'application. Une temporisation se produit lorsque l'établissement d'une liaison simple ou une séquence d'établissements de liaisons ne sont pas terminés dans le temps imparti. Le programme d'application devra être capable de valider et d'invalider cette facilité. La documentation du contrôleur devra comporter l'information sur la façon dont les temporisations sont contrôlées.

15.3 *Recommendations du contrôleur CEI 625-2*

15.3.1 *Lignes de bus de surveillance*

Pour le diagnostic, les **contrôleurs** peuvent contrôler les lignes de signaux CEI 625-1 et indiquer leur état et/ou leurs transitions au programme d'application.

Si le **contrôleur** n'implémente pas les protocoles communs de contrôleur SETADD et FINDLSTN, le programmeur d'application doit tester NRFD et NDAC pour exécuter ces protocoles au niveau de l'application.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

of three-state drivers to drive the DIO1-8 signal lines as appropriate to the **controller's** current operation.

15.2 *Additional IEC 625-2 controller requirements*

A **controller** shall have:

1) a means of providing low-level bus control via application software to allow the user independently to:
   - pulse IFC TRUE for greater than 100 ms,
   - set the REN signal line either TRUE or FALSE,
   - send singly or in any combination any interface message defined in IEC 625-1,
   - send and detect the IEC 625-1 END message (i.e, the EOI signal line TRUE and the ATN signal line FALSE handshook with a data byte);
2) the facility (either directly or via the application program) to input and output the codes, formats, protocols, and common commands as defined in this standard;
3) the facility to sense the state (TRUE or FALSE) of the SRQ line;
4) the facility to sense FALSE to TRUE SRQ line transitions. Timing of application program actions following the transition is beyond the scope of this standard;
5) the facility (either directly or via the application program) to examine the status byte on a bit basis;
6) the facility to detect the error condition of the **controller** attempting to source handshake a byte while all other devices are in AIDS and relay the error to the application program;
7) the facility to timeout on controller-to-device and device-to-controller message exchange and have the ability to relay the occurrence of a timeout to the application program. A timeout occurs when a single handshake or a sequence of handshakes are not completed within an allocated time. The application program shall be able to enable and disable this facility. The controller documentation shall include information on how timeouts are controlled.

15.3 *IEC 625-2 controller recommendations*

15.3.1 *Monitoring bus lines*

For diagnostic use, **controllers** may monitor the IEC 625-1 signal lines and report their status and/or transitions to the application program.

If the **controller** does not implement the SETADD and FINDLSTN common controller protocols, the application programmer needs to monitor NRFD and NDAC to perform these protocols at the application layer.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn

15.3.2 *Temporisations*

La possibilité de faire varier la valeur de la temporisation est recommandée. La gamme et la résolution sont dépendantes du contrôleur, mais devront figurer dans la documentation.

15.3.3 *Interruptions SRQ*

Le paragraphe 15.2 prescrit qu'un **contrôleur** est capable de détecter et d'indiquer les transitions de FAUX à VRAI de SRQ. De plus, un **contrôleur** peut facultativement fournir à un programme d'application un moyen de se brancher sur une section spécifiée du programme d'application lorsque SRQ passe de FAUX à VRAI. Le programme de prise en charge de SRQ peut retarder l'exécution jusqu'à ce que le message en cours ait été complètement transmis ou reçu, son exécution pourra commencer immédiatement après que l'octet d'état en cours eut été complètement transféré, ou il pourra retarder l'exécution jusqu'à un moment convenant au **contrôleur**.

16 Protocoles d'échanges de messages du contrôleur

La figure 16-1 représente le modèle logique d'un **contrôleur**.

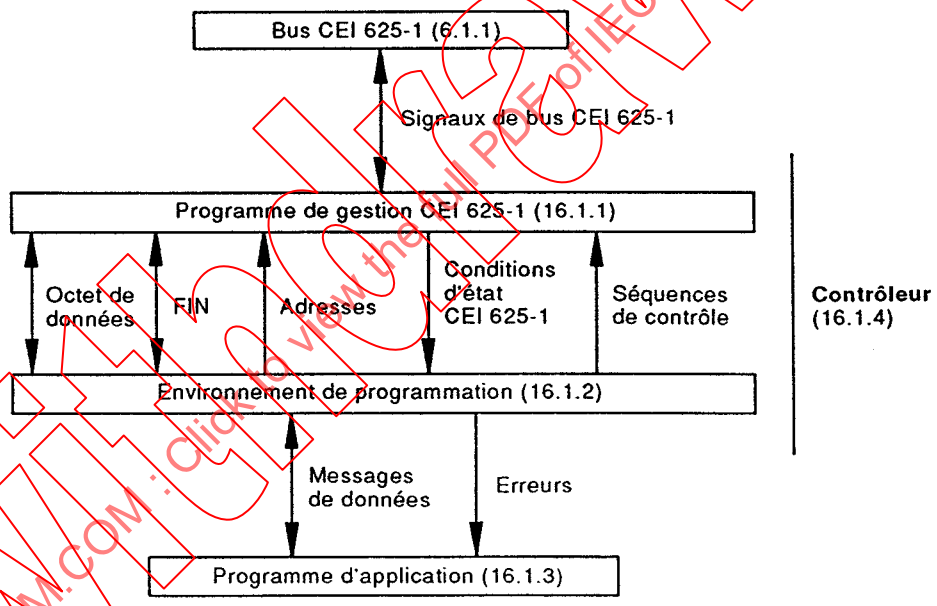


Figure 16-1 – Modèle de contrôleur

16.1 *Définitions*

16.1.1 *Programme de gestion CEI 625-1*

Cet élément contient les machines d'état décrites dans CEI 625-1. Il contrôle également une grande partie du séquencement de messages éloignés.

16.1.2 *Environnement de programmation*

Ce bloc fournit les capacités relatives à l'envoi et à la réception de messages sur le bus CEI 625-1 du programme d'application. Cette norme décrit les capacités qui devront être fournies au programme d'application.

IECNORM.COM : Click to view the full PDF of IEC 60625-2:1993
Withdrawn