INTERNATIONAL STANDARD

ISO 11073-91064

First edition
2009-05-01

Health informatics — Standard communication protocol —

Part 91064: Computer-assisted electrocardiography

Informatique de santé — Communication entre dispositifs médicaux sur le site des soins —

Partie 91064: Protocole de communication standard pour l'électrocardiographie assistée par ordinateur

Reference number
ISO 11073-91064:2009(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword ... iv
Introduction ... v
1 Scope ... 1
2 Normative references ... 1
3 Terms and definitions ... 1
4 Abbreviations ... 3
5 Definition of the data contents and format ... 4
5.1 General considerations ... 4
5.2 Specifications for the data structure ... 5
5.3 Pointer section – Section 0 ... 8
5.4 Header information – Patient data/ECG acquisition data – Section 1 ... 10
5.5 Huffman tables – Section 2 ... 23
5.6 ECG lead definition – Section 3 ... 24
5.7 QRS locations, reference beat subtraction zones and protected areas – Section 4 ... 30
5.8 Encoded type 0 reference beat data – Section 5 ... 32
5.9 Rhythm data – Section 6 ... 34
5.10 Global measurements – Section 7 ... 36
5.11 Storage of full text interpretive statements – Section 8 ... 41
5.12 Storing manufacturer specific interpretive statements and data related to the overreading trail – Section 9 ... 43
5.13 Lead measurement block – Section 10 ... 43
5.14 Storage of the universal ECG interpretive statement codes – Section 11 ... 46
6 Minimum requirements for encoding and compression of the ECG signal data ... 48
6.1 Scope and field of application ... 48
6.2 Introduction ... 48
6.3 ECG compression methodology ... 49
6.4 Main results from investigations on ECG data compression in the SCP-ECG Project ... 50
6.5 Minimum requirements for ECG data compression ... 51
Annex A (normative) Encoding of alphanumeric ECG data in a multilingual environment ... 53
Annex B (normative) Definition of compliance with the SCP ECG standard ... 66
Annex C (normative) Methodology and conformance testing of the recommended ECG signal compression technique ... 74
Annex D (informative) Definition of a minimum set of control and query messages for the interchange of ECG data ... 106
Annex E (informative) Standard low-level ECG-Cart to host protocol ... 121
Annex F (informative) Universal ECG interpretation statements codes ... 132
Annex G (informative) Glossary ... 154
Bibliography ... 156

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 11073-91064 was prepared by Technical Committee ISO/TC 215, *Health informatics*.

Introduction

The electrocardiogram (ECG) is a recording of voltage changes transmitted to the body surface by electrical events in the heart muscle, providing direct evidence of cardiac rhythm and conduction, and indirect evidence of certain aspects of myocardial anatomy, blood supply and function. During its propagation to the surface, extracardiac tissues may intervene and influence the ECG.

Electrocardiography has been used for many years as a key, non-invasive method in the diagnosis and early detection of coronary heart disease, which is the leading cause of mortality in western countries. In 1993, it was estimated that more than 100 million standard ECGs are recorded yearly in the European Community (EC) for routine diagnostic and screening purposes at an estimated cost of more than 1,2 billion € per year.

Almost all newer electrocardiographs nowadays use digital recording, interpretation and communication techniques. These stand-alone, microcomputer based machines can be connected to each other, and to larger minicomputer-based management servers for long-term storage and serial comparison. To this end, various manufacturers have used different techniques.

It is in the general public interest for users not to be restricted in their options by incompatible technical features and services of different systems. ECG processing is increasingly being integrated with various other data processing in health care. This evolution shall have considerable impact on the storage and communication of ECG data. There are many different end-users who for different purposes (support of patient care, management, research and education) want to obtain a copy of the signal data, of the interpretive report and/or measurement results. Being one of the very first systems for medical decision support, computerized ECG interpretation stretches from departments of cardiology in hospitals, to general practitioners in primary care and health care centres. In life-threatening acute myocardial infarction, ECGs are being used in ambulances by paramedical personnel to assess the necessity for administering thrombolytic agents, with long-distance monitoring whenever possible.

To enable the exchange of information between various systems it was of utmost importance that a standard communications protocol for computer-aided electrocardiography (SCP-ECG) had to be established, as defined in this document. The primary aim of this document is to specify a data format for transferring ECG reports and data from any vendor's computerized ECG recorder to any other vendor's central ECG management system. The same standard should also allow standardized transfer of digitized ECG data and results between various computer systems.

Under the standard communication protocol (SCP) the contents and format of the ECG waveform data and the measurements from ECG devices of different manufacturers are not expected to be identical. As a result, the determination of the suitability of a device and/or system for any particular application remains with the user/purchaser. The following possible uses of ECG records require special attention:

— serial comparison of ECGs and interpretations;

— plot formats of ECGs;

— maintaining audit trail of edits;

— bi-directional communication and remote query.

The user is cautioned to make sure that the data contents and format of the waveform data, measurements, and the interpretive statements meet his or her specific needs. If more than one type of ECG device and/or database management system are interconnected, the user is also advised to verify with the manufacturers that the data from different systems are compatible with each other and with the user's needs.

In order to understand this document, the reader needs some basic understanding of electrocardiology, electrocardiography and signal processing.

This part of ISO 11073 relates to the conventional recording of the electrocardiogram, i.e. the so-called standard 12-lead electrocardiogram and the vectorcardiogram (VCG). Initially, the electrical connections used for recording the ECG were made to the limbs only. These connections to the right arm (RA), left arm (LA), left leg (LL) and right leg (RL) were introduced by Einthoven. The electrical variations detected by these leads are algebraically combined to form the bipolar leads I, II and III. Lead I, for example, records the difference between the voltages of the electrodes placed on the left arm and the right arm. The unipolar electrocardiographic leads (aVR, aVL, aVF and the precordial leads V1 to V6) were introduced much later, starting in 1933. In these leads, potentials are recorded at one location with respect to a level which does not vary significantly in electrical activity during cardiac contraction. The “augmented” limb lead potentials are recorded with reference to the average potential of (L+F), (R+F) and (L+R) respectively. The unipolar chest leads are recorded with reference to the average potential of $\frac{(RA+RL+LL)}{3}$ which is called the Wilson “central terminal” (CT). In vectorcardiography, recordings are made of three mutually perpendicular leads, running parallel to one of the rectilinear coordinate axes of the body. The axes are the X-axis going right to left, the Y-axis with a top to bottom orientation and the Z or front to back axis.

In some research centres, so-called body surface maps are obtained by placing many (from 24 to 124 or even more) closely-spaced electrodes around the torso. This part of ISO 11073 has not been designed to handle exchanges of such recordings, although future extensions could be made to this end. This part of ISO 11073 has also not been designed to exchange specialized recordings of intracardiac potentials or of the so-called Holter or other long-term ECG recordings made for monitoring cardiac rhythm. This part of ISO 11073 also does not address exercise ECG recordings.

ECG computer processing can be reduced to three principal stages:

1) data acquisition, encoding, transmission and storage;

2) pattern recognition and feature extraction, i.e. ECG measurement;

3) diagnostic classification.

In each of these stages there are important needs for standardization and quality assurance testing. The scope of this part of ISO 11073 is confined to the first of these three stages.

The various data sections that shall be transmitted by means of the standard ECG communications protocol are defined in Clause 5. Minimum requirements for data encoding and compression are defined in Clause 6.

The compliance categories defined in Annex B provide users and manufacturers of ECG devices and/or systems with a relatively simple codification of SCP-ECG related features and information content that may be provided by a specific device. Two data format categories have been defined based on information content as in Table 1.

Table 1 — Data format categories for compliance specifications

Category	Data sections required	Content description
I	0, 1, [2], 3, 6, (7), (8), (10)	Demographics, and ECG rhythm data (uncompressed or with lossless compression)
II	0, 1, [2], 3, 4, 5, 6, (7), (8), (10)	Demographics, ECG rhythm data (uncompressed, with lossless compression or with high compression), and reference beats

NOTE 1 Square brackets [ ] indicate that data section 2 is required if Huffman encoding has been used.

NOTE 2 Parentheses ( ) indicate that these data sections are optional for export.

A further category may be added in future versions in order to fulfil the specific needs of ECG devices used in other applications (such as telemedicine or homecare).

All devices stating an SCP-ECG data format category shall import at minimum data sections 0, 1, 3, 6, 7 and 8. All categories may have additional sections added (e.g. 9, 10, 11). Manufacturer-specific data shall be optionally included only in manufacturer-specific fields, bytes and data blocks that have been defined in the document. Reserved, unspecified and undefined fields, bytes or data blocks shall not be used for manufacturer-specific data.

For a particular device, an SCP-ECG compliance statement lists data format category(ies) for export (i.e. acquiring and making available an SCP-ECG record) and import (i.e. accepting, and making available to a user, an SCP-ECG record). A device may also state its ability to transfer (i.e. making available an SCP-ECG record without changing its data format, for example, exporting a record that was previously imported). (These terms are precisely defined in Annex B for the purpose of this part of ISO 11073).

The selection and definition of ECG-specific high-level syntaxes for transfer of messages and data host-to-hosts, such as EDIFACT or ASN.1, are beyond the scope of this part of ISO 11073.

Health informatics — Standard communication protocol —

Part 91064: Computer-assisted electrocardiography

1 Scope

This part of ISO 11073 specifies the common conventions required for the cart-to-host as well as cart-to-cart interchange of specific patient data (demographic, recording, ...), ECG signal data, ECG measurement and ECG interpretation results.

This part of ISO 11073 specifies the content and structure of the information that is to be interchanged between digital ECG carts and computer ECG management systems, as well as other computer systems where ECG data can be stored.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

ISO/IEC 2022:1994, *Information technology — Character code structure and extension techniques*

ISO/IEC 4873, *Information technology — ISO 8-bit code for information interchange — Structure and rules for implementation*

ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

JIS X 0201-1976, *Code for Information Interchange*

JIS X 0208-1997, *Code of the Japanese Graphic Character Set for Information Interchange*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
acquiring cardiograph
cardiograph recording the original ECG signal

3.2
bimodal compression
use of low pass filtering and sample decimation outside of a protected zone containing the QRS complex, with no decimation or filtering within the protected zone, indicated by 5.9.3 byte 6

3.3
confirming
process whereby a trained and experienced cardiologist reviews the computer-generated (or overread) interpretation of an ECG in order to confirm the computer-generated (or overread) interpretation or to make the final changes to the interpretation text

NOTE The confirmed ECG is the final clinically acceptable version for diagnosis and treatment.

3.4
CSE Project
project supported by DG XII of the European Commission aiming at the development of Common Standards for (Quantitative) Electrocardiography

3.5
downsampling factor
decimation factor
factor that gives the reduction of samples in data sections where the sampling rate is reduced with reference to the original sampling rate.

NOTE This applies for bimodal data compression.

EXAMPLE Original sampling rate 500 S/s (equivalent to a sample interval of 2 ms) is reduced to 125 S/s (equivalent to a sample interval of 8 ms). The downsampling factor is then 4.

3.6
interpretive device
device (cart, computer) analysing the ECG signal

3.7
message
textual body of information

3.8
overreading
process whereby a cardiologist or a cardiology fellow reviews the computer-generated interpretation of an ECG in order to verify the accuracy or to make changes to the interpretation text

NOTE An overread ECG is generally not the final clinically acceptable version for diagnosis and treatment. Usually, the overreading process precedes the confirming process.

3.9
record
entire data file to be transmitted, including the ECG data and associated information, such as patient identification, demographic and other clinical data

3.10
reference beat
reference/representative ECG cycle computed through any (but not specified) algorithm comprising the P, QRS and the ST-T waves

3.11
residual data
remaining original ECG data after "proper" subtraction of the reference beat where the adjective "proper" refers to accurate beat alignment

3.12
rhythm data
full original ECG data, or the decompressed and reconstructed ECG data at reduced resolution

NOTE Rhythm data is typically 10 s in length.

3.13
section
aggregate of data elements related to one aspect of the electrocardiographic recording, measurement or interpretation

3.14
universal statement codes
ECG interpretation codes

See Annex F.

NOTE See Glossary in Annex G for other technical terms related to this part of ISO 11073.

4 Abbreviations

AAMI	American Association for the Advancement of Medical Instrumentation
AC	Alternating current
AHA	American Heart Association
AIM	Advanced Informatics for Medicine Programmes of the European Commission Directorate General XIII
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
AVM	Amplitude Value Multiplier (see 5.8.3)
BS	Backspace (control character)
CCITT	International Telegraph and Telephone Consultative Committee
CEN	Comité Européen de Normalisation/European Committee for Standardization
CR	Carriage return (control character)
CRC	Cyclic redundancy check
CSE	Common standards for quantitative electrocardiography
DG	Directorate General (of the European Commission)
EC	European Community
ECG	Electrocardiogram
ECU	European currency unit (€)
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
EN	Europäische Norm (European Standard)
ENV	Europäische Norm Vorausgabe (European Pre-standard)
ESC	Escape (control character)

FF	Form feed (control character)
HT	Horizontal Tab (control character)
ICD	International Classification of Diseases
ID	Identification
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IMIA	International Medical Informatics Association
ISO	International Organization for Standardization
JIS	Japanese Industrial Standard
LF	Line feed (control character)
LSB	Least significant bit
MSB	Most significant bit
RMS	Root mean square
SCP	Standard Communications Protocol
SCP-ECG	Standard Communications Protocol for Computerized Electrocardiography
TC	Technical Committee
VCG	Vectorcardiogram
VT	Vertical tab (control character)

5 Definition of the data contents and format

5.1 General considerations

5.1.1 The data record which is to be interchanged shall be divided into different sections. The contents and format of each of these sections are defined in this part of ISO 11073.

5.1.2 All text data (character strings) shall comply to the limited conformance requirements of ISO/IEC 2022, described in Annex A. Latin-1 (ISO/IEC 8859-1) shall be the default character set.

5.1.3 All character strings shall be NULL terminated (not part of ISO/IEC 2022).

5.1.4 For all signed binary values 2's-complement coding shall be applied.

5.1.5 All single and multiple byte binary values are regarded as unsigned integers, if not otherwise specified.

5.1.6 Binary values spanning more than 1 byte shall be transmitted in ascending order of significance (the least significant byte is transmitted first, the most significant byte last).

5.1.7 Consecutive bytes are numbered from left to right (starting with 1). Bits of a byte are numbered from right to left (0 = LSB, 7 = MSB).

5.1.8 The first byte in the record (i.e. the first byte of the checksum) is defined as Byte 1.

5.1.9 ECG samples are indexed and numbered starting with sample number 1. Sample index 0 is not used in this part of ISO 11073. The sample index is a ones-based 16-bit index. The first sample starts at time 0. The second sample is at time (0 + 2) ms in case of 500 samples/s sampling rate.

5.1.10 Sections are numbered starting from 0 (the Pointer Section) to 32 767.

5.1.11 The term “Reference Beat” used in this part of ISO 11073 refers to an ECG complex that is chosen as representative of a class of such complexes. No specific statistical meaning is implied by this term; for example, it may be an averaged beat, a “Median Beat”, a selected or any other representative single cycle taken from the total ECG recording. This “Reference Beat” does include the P-wave if present (not in case of atrial fibrillation), the ST-T segment and the T wave of this beat.

An ECG may have multiple reference beats. The term “Beat type” used in this part of ISO 11073 refers to any one of an ordered list of reference beats, starting with reference beat type 0 (zero). Reference beat type 0 is, by definition, the reference beat used for classification of the ECG, and for reference beat subtraction, if reference beat subtraction is used in compression. The ordering of the list of reference beats does not imply a temporal sequence within the rhythm data.

The term “Rhythm Data” is used to indicate the ECG recording over the entire recording time, usually 10 s in most recorders. A description of these terms and of the recommended data compression methodology, including numerical examples and the methods for conformance testing on the minimum requirements of data compression and signal distortion are given in Clause 6, Annex B and Annex C.

Reference Beat type 0 data in 5.8 are intended to be used for display, (re)analysis and, if reference beat subtraction has been used for data compression, for Rhythm Data reconstruction.

5.1.12 All indexes or pointers to a field are defined in bytes and are ones-based (start at 1) if not otherwise specified.

5.1.13 1 KByte = 1 024 bytes.

5.2 Specifications for the data structure

5.2.1 All sections shall start on an odd index (even offset) boundary. This implies that all sections shall contain an even number of bytes. A padding byte has to be added to the end of any section containing an odd number of bytes. Padding bytes shall always be set to NULL. Blocks of data within a section may contain either odd or even numbers of bytes. Padding occurs only at the end of a section if needed.

5.2.2 All sections are given identification numbers. Section ID numbers 0 to 11 are currently defined in the SCP-ECG protocol, numbers 12 to 127, as well as numbers above 1 024 are reserved for future use. Numbers 128 to 1 023 are for manufacturer-specific sections. The combination of the manufacturer code (see 5.4.3.1, tag 14) and section numbers 128 to 1 023 uniquely defines the content of the manufacturer-specific sections. There are no specific rules for the layout and format of these sections. However, use of the structure defined in 5.2.7 is recommended.

5.2.3 Inclusion of Sections 2, 4, 5, 7 to 11 (5.2.7 and 5.2.8) is optional. Any SCP-ECG data record shall contain Section 0 (Pointers), Section 1 (Header), Section 3 (ECG Lead Definition) and Section 6 (Rhythm Data). No other consistency checking among the presence of different sections is assumed. Specifically, if any of the Sections 8, 9 or 11 is present, it is not assumed that all three shall be present.

5.2.4 The ECG record starts with a 6-byte record header, consisting of a 2-byte CRC followed by a 4-byte record length. These are defined as follows:

1) the 2-byte cyclic redundancy check (CRC) is calculated as a CRC-CCITT, the algorithm of which is described in E.5.5, and is calculated over the entire range starting with the first byte following the CRC and ending with the last byte in the record;

2) the 4-byte record length denotes the number of bytes in the total record, including the 6 bytes of this record header.

5.2.5 Record overview:

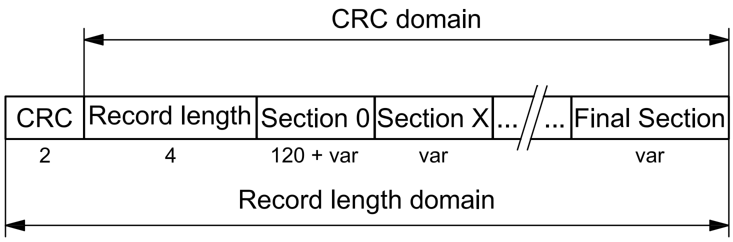


Figure 1 — Record overview

5.2.6 The sequence order of the sections of a record is free, with the exception of Section 0 (zero) which shall immediately follow the record header. However, a maximum of one instance of any section is allowed in an SCP-ECG data record.

5.2.7 Each section consists of:

1) a Section Identification Header (Section ID Header);

2) a Section Data Part.

Any section shall start with a “Section ID Header” (16 bytes) defined below:

Bytes	**Contents**
1 to 2	16 bit CRC-CCITT over the entire section except these 2 bytes.
3 to 4	**Section ID number** as defined in 5.2.2 (see also 5.3.3.1).
5 to 8	**Section length** in bytes <u>including</u> the “Section ID Header” (5.3.3.2).
9	**Version number of the Section**.
10	**Version number of the Protocol** (see 5.4.3.1 tag 14, byte 15).
11 to 16	**Reserved** (for data section 0 see 5.3.1).

Each section shall have a Section Protocol Version Number (see bytes 9 and 10) which may be used to specify different levels of compatibility with the standard when this is updated in the future (see Annex B). For data Sections 1 to 11, Section Version Numbers (byte 9) shall be the Protocol Version under which the section was approved. For data Sections 12 to 1 023, Section Version shall refer to the manufacturer’s version for that section, independent of the Protocol version.

5.2.8 Reserved fields shall always be set to NULL (zero).

5.2.9 Section layout overview:

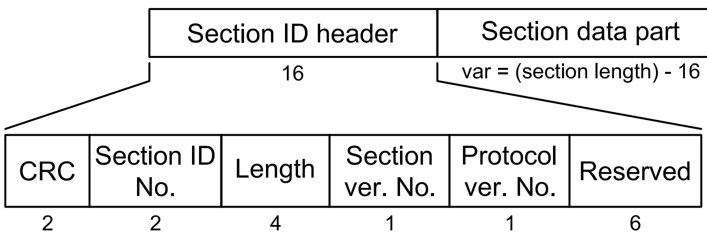


Figure 2 — Section layout overview

5.2.10 The numbers in italic in the layout overviews (in 5.2.5, 5.2.9 and below) indicate the length in bytes of the corresponding field or indicated block (var = variable length).

5.2.11 A global overview of the SCP-ECG data structure is presented in Table 2.

Table 2 — SCP-ECG data structure

Requirement status	Content
Required	2 BYTES - CHECKSUM - CRC - CCITT OVER THE ENTIRE RECORD (EXCLUDING THIS WORD)
Required	4 BYTES - (UNSIGNED) SIZE OF THE ENTIRE ECG RECORD (IN BYTES)
Required	(Section 0) POINTERS TO DATA AREAS IN THE RECORD
Required	(Section 1) HEADER INFORMATION – PATIENT DATA/ECG ACQUISITION DATA
Dependent	(Section 2) HUFFMAN TABLES USED IN ENCODING OF ECG DATA (IF USED)
Required	(Section 3) ECG LEAD DEFINITION
Optional	(Section 4) QRS LOCATIONS (IF REFERENCE BEATS ARE ENCODED)
Optional	(Section 5) ENCODED REFERENCE BEAT DATA IF REFERENCE BEATS ARE STORED
Required	(Section 6) “RESIDUAL SIGNAL” IF REFERENCE BEAT SUBTRACTION AND REFERENCE BEATS STORAGE ARE PERFORMED, OTHERWISE ENCODED RHYTHM DATA
Optional	(Section 7) GLOBAL MEASUREMENTS
Optional	(Section 8) TEXTUAL DIAGNOSIS FROM THE “INTERPRETIVE” DEVICE
Optional	(Section 9) MANUFACTURER-SPECIFIC DIAGNOSTIC AND OVERREADING DATA FROM THE “INTERPRETIVE” DEVICE
Optional	(Section 10) LEAD MEASUREMENT RESULTS
Optional	(Section 11) UNIVERSAL STATEMENT CODES RESULTING FROM THE INTERPRETATION

5.2.12 The following remarks apply to the data areas identified above.

0 This section contains pointers to the start of each of the following sections. This section is required.

1 This section contains information of general interest concerning the patient (e.g. patient's name, patient's ID, age, etc.) and the ECG (acquisition date, time, etc.). This section is required.

2 This section contains all of the Huffman tables used in the encoding of rhythm (or "residual signal") and reference beat data. The tables shall be referenced by Sections 5 and 6 by their numerical order in this section. Thus, when reference is made in the reference beat encoding section to Table 2, this shall refer to the second table defined in Section 2. This section is required, dependent upon Huffman encoding being used in the encoding of rhythm (or "residual signal") and of reference beat (if stored).

3 This section specifies which ECG leads are contained within the record. This section is required.

4 If reference beats are encoded, then this section shall identify the position of these reference beats relative to the "residual" signal contained in Section 6. This section is optional.

5 Reference beats for each lead are encoded if the originating device has identified those complexes. This section is optional.

6 This section contains the "residual" signal that remains for each lead after the reference beats have been subtracted, or if no reference beats have been subtracted, the entire rhythm signal. This section is required.

7 This section contains global measurements for each reference beat type or for each QRS contained in the record and a list of possible pacemaker spikes in the record. This section is optional.

8 This section contains the latest actual text of the diagnostic interpretation of the recorded ECG data, including all overreadings if performed. Only the text of the most recent interpretation and overreading shall be included in this section. No manufacturer-specific codes should be used in the text. Mnemonic codes as listed in the Universal Statement Codes may be used if necessary. The data contained in this section shall be consistent with the data in Section 9 and Section 11. This section is optional.

9 This section contains the manufacturer-specific diagnostic statements of the analysing device and overreading trails of the interpretations. The source of the analysing device and the name of the latest confirming physician (or device) are defined in the "Header section" (Section 1). This section is optional.

10 A set of basic measurements and manufacturer-specific measurements (if any) for each recorded lead are presented in this section. This section is optional.

11 This section contains the most recent interpretation and overreading data, coded according to the Universal Statement Codes and Coding rules (Annex F). The data contained in this section shall be consistent with the data in Sections 8 and 9. This section is optional.

5.3 Pointer section – Section 0

5.3.1 The purpose of this section is to store pointers at the remaining sections in the record. All sections are given identification numbers, as listed in 5.2.2.

5.3.2 The section starts with a "Section ID Header" as defined in 5.2.7. Bytes 11 to 16 of the Section ID Header shall contain the six-character ASCII string: "SCPECG".

5.3.3 To provide a flexible way of section management, the data part of the pointer section are defined as follows.

- One pointer field for each of Sections 0 to 11 defined by SCP-ECG protocol shall be provided in the pointer section, whether the optional sections are present or not. For any optional section not included in an SCP-ECG data record, the special codes defined in 5.3.3.2 and 5.3.3.3 for the pointer field shall be used.
- Manufacturer specific sections, if present, shall have a pointer field in Section 0.
- The first pointer field included in this section shall be the field for Section 0 (this section).

Each pointer field contains three parts:

a) A Section Identification (see 5.3.3.1).

b) A Section Length (see 5.3.3.2).

c) An Index to the Section (see 5.3.3.3).

5.3.3.1 The Section Identification number is stored in 2 bytes containing the section number, as listed in 5.2.2. Section ID numbers 0 to 11 are currently defined in this SCP-ECG protocol, numbers 12 to 127, as well as numbers above 1 023 are reserved for future use, numbers 128 to 1 023 are codes for manufacturer-specific sections.

5.3.3.2 The length, in bytes, of a section (= an even number, see 5.2.1) is presented in this unsigned 4 byte integer field part. The length includes the “Section ID Header” bytes (see 5.2.7). The 4 byte integer is necessary to allow sections to be larger than 32 KBytes. For data Sections 2 to 11 a pointer field shall be included. If no data are transmitted for any of these sections, set the section length to 0.

5.3.3.3 An index to the first byte of a section shall be presented in this unsigned 4 byte integer field part. The index is calculated relative to the start of the record, i.e. byte 1 of the record (first byte of the CRC). For example, if Section 11 begins at an offset of 128 900 bytes from the beginning of the ECG record, the index to Section 11 would be set to 128 901. The 4 byte integer is necessary to allow an SCP-ECG record to be larger than 32 KBytes. If a section is not included in the SCP-ECG record the index shall be set to NULL (0). The index to Section 0 shall always be set to 7, since Section 0 is always preceded by the Checksum (2 bytes) and the Record Length (4 bytes).

5.3.3.4 The pointer fields shall be in numerical order. However, the sections themselves do not necessarily have to follow in numerical order.

5.3.4 Pointer section data part overview:

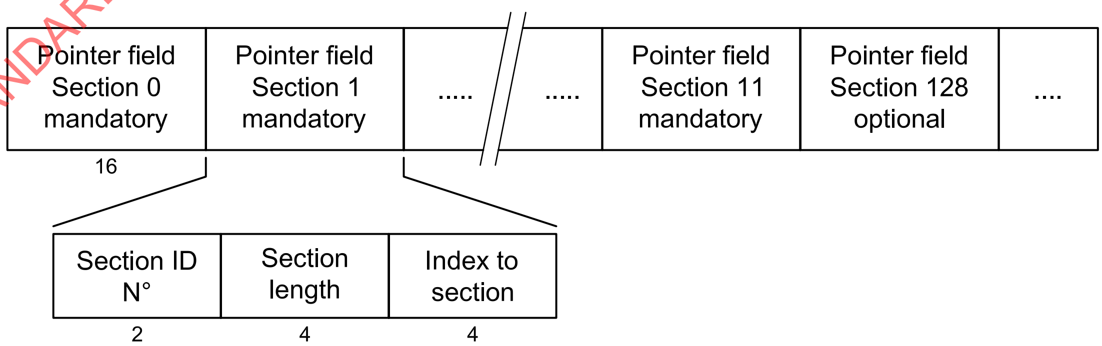


Figure 3 — Pointer section data part overview

5.4 Header information – Patient data/ECG acquisition data – Section 1

5.4.1 General

The section shall start with a "Section ID Header" as defined in 5.2.7.

5.4.2 Introduction to the section data part

The following layout details the format that should be used to transmit patient demographic and ECG administrative data as part of the standard (SCP-ECG) communications protocol for digital ECG data.

5.4.3 Basic methodology

5.4.3.1 It is recognized that, although a large number of parameters may be transmitted, most devices will only send a subset of that number. As a result, it was agreed that the format of the patient demographic area should be made flexible.

Each parameter shall be stored in a separate field. Including a field in this section shall be optional, with the single provision that the following parameters (1 to 4) shall be present.

	Tag	Parameter
1	2	Patient ID (used as primary key in the ECG management database)
2	14	ID of the acquiring device
3	25	Date of acquisition
4	26	Time of acquisition

In addition, the SCP-ECG Working Group highly recommends the following parameters for uniquely identifying the patient and time of acquisition.

	Tag	Parameter
5	0	Patient's last name
6	1	Patient's first name
7	5	Patient's date of birth (the date of birth shall in principle be given AD)
8	8	Patient's sex
9	15	ID of the analysing device
10	34	Date time zone

5.4.3.2 Flexibility is achieved by identifying each field by the following means.

a) One leading specification byte, referred to as "tag", indicating the contents of the parameter field.

b) A 2-byte unsigned integer, referred to as "length", containing the length of the field value in bytes, allowing variable length text entries and use of multiple byte character sets (as the Japanese character set). The NULL terminator character of a text string shall be included to calculate the field length. For example, for the last name "Menuel" the length shall be listed as 7, the NULL included. A length field value of 0 is allowed, which is equivalent to "not defined".

c) Zero or more parameter bytes, referred to as "value", containing the actual parameter data.

The field tag (1 binary byte) permits a total of 255 different field types to be defined (0 to 254; 255 is used as terminator), of which 55 (200 to 254) are reserved for use by an individual manufacturer. Any field identified by a value of 200 to 254 is not defined within the specification of this protocol and permits a manufacturer to define its own set of fields.

The field length (2 binary bytes) shall contain the actual length of the field value. The tag and length bytes (first 3 bytes of any field) are not included in the field length. The maximum possible length of each field value is 65 535 bytes (unsigned 2 bytes). However, for practical reasons the maximum length of a field shall not exceed 64 bytes, except for the free text items (see 5.4.3.5).

The field value, containing the actual parameter data, can be of any combination of binary bytes and text characters.

5.4.3.3 A maximum of one instance of any field defined in 5.4.5 is allowed to be included in the "Header" section, except for the following fields listed below:

Tag	**Value description**	**Max. instances**
10	Drugs	no limit
13	Diagnosis or referral indication	no limit
30	Free text	no limit
32	History diagnostic codes	no limit
35	Free-text Medical History	no limit

5.4.3.4 The first 16 characters of the patient identification number shall be unique.

5.4.3.5 In order to facilitate the implementation of the protocol, a maximum field length, i.e. 64 (except for tag 13, tag 30 and tag 35, where the limit is 80) and reasonable values for the length of the different free text fields have been determined, as shown in Table 3.

Table 3 — Maximum and reasonable length of free text fields

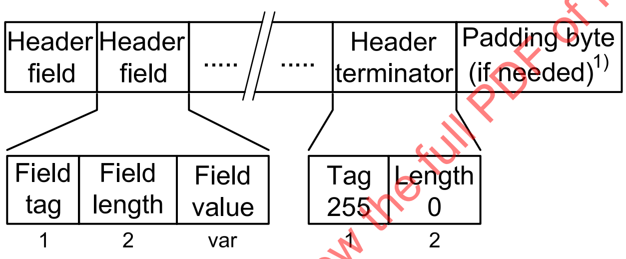
Section	Tag	Contents	Instance > once	Reasonable length
1	0	Last name	–	40
1	1	First name	–	40
1	2	Patient identification number	–	40
1	3	Second last name	–	40
1	10	Drugs	yes	40[a]
1	13	Diagnosis or referral indication	yes	80[a]
1	14	Acquiring device identification number	–	40
1	15	Analysing device ID number	–	40
1	16	Acquiring institution description	–	40
1	17	Analysing institution description	–	40
1	18	Acquiring department description	–	40
1	19	Analysing department description	–	40

Table 3 (*continued*)

Section	Tag	Contents	Instance > once	Reasonable length
1	20	Referring physician	–	60
1	21	Latest confirming physician	–	60
1	22	Technician description	–	40
1	23	Room description	–	40
1	30	Free text field	yes	80[a]
1	31	ECG sequence number	–	12
1	35	Free-text medical history	yes	80[a]

[a] Multiple instances are allowed for these fields, each with 40 or 80 characters, NULL terminated.

5.4.4 An overview of the “Header” section data part is presented below:



NOTE Padding bytes (if any) should be set to zero. This applies to all sections, but will not be shown in all the following diagrams.

Figure 4 — Overview of the “Header” section data part

5.4.5 For specification of the defined parameters, see Table 4.

Table 4 — Specification of the defined parameters

Tag	Length	Value (parameter data)
0	length	**Last name** (text characters) This shall also be used to transmit the entire name if the originating unit does not explicitly determine a first name.
1	length	**First name** (text characters)
2	length	**Patient ID** (text characters)
3	length	**Second last name** (text characters) The field value may be defined as appropriate for the country or area where the ECG-device is used. For instance in the USA this field may hold the family member prefix code, in France it may contain the patient's maiden name, and in Portugal as well as in Spain and several Latin American countries, the second last name of the patient.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
4	3	**Age** (binary) This field has the following format: Byte Contents 1 to 2 Binary: Age in units as indicated in byte 3. 3 Binary: Units of age as defined below: Value Unit 0 Unspecified 1 Years 2 Months 3 Weeks 4 Days 5 Hours If all 3 bytes are zero, then age is not specified.
5	4	**Date of birth** (binary) This field has the following format: Byte Contents 1 to 2 Binary: Year (full integer notation, as in 1 990). 3 Binary: Month (range 01 to 12; 01 = January). 4 Binary: Day (range 01 to 31). If all 4 bytes are zero, then date of birth is not specified.
6	3	**Height** (binary) This field has the following format: Byte Contents 1 to 2 Binary: Height in units as indicated in byte 3. 3 Binary: Units of height as defined below: Value Unit 0 Unspecified 1 Centimetres 2 Inches 3 Millimetres If all 3 bytes are zero, then height is not specified.
7	3	**Weight** (binary) This field has the following format: <u>Byte</u> <u>Contents</u> 1 to 2 Binary: Weight in units as indicated in byte 3. 3 Binary: Units of weight as defined below: <u>Value</u> <u>Unit</u> <u>Value</u> <u>Unit</u> 0 Unspecified 3 Pound 1 Kilogram 4 Ounce 2 Gram If all 3 bytes are zero, then height is not specified.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
8	1	**Sex** (binary) This has the following format: Byte Contents 1 Binary: Sex indication defined as: Value Sex Value Meaning 1 Male 0 Not Known 2 Female 9 Unspecified
9	1	**Race** (binary) This has the following format: Byte Contents 1 Binary: Race indication defined as: Value Race 0 Unspecified 1 Caucasian 2 Black 3 Oriental 4 to 9 Reserved 10 to 255 Other (manufacturer specific)
10	length	**Drugs** (binary bytes and text characters) Each drug entered in the patient demographic area shall be described by the following structure: Byte Contents 1 Binary: Drug code table indicator. If byte 1 is set to 0 then the following table applies. 2 Binary: Class code. 3 Binary: Specific drug code within the specified class. 4 to *** Text characters: Text description of the drug (optional). The following classes are defined: 0 - Unspecified 5 - Antianginal 1 - Digitalis 6 - Antithrombotic 2 - Antiarrhythmic 7 - Beta blockers 3 - Diuretics 8 - Psychotropic 4 - Antihypertensive 9 - Calcium blockers 10 - Antihypotensive 100 - Not taking drugs 11 - Anticholesterol 101 - Drugs, but unknown type 12 - ACE- Inhibitors 102 - Other medication 13 to 99 - Reserved 103 to 255 - Manufacturer specific codes
		NOTE 1 A class code of 0 is always followed by a drug code of 0, indicating that the drug is undefined within this document, and that the text in bytes 4 to *** is the only description available. NOTE 2 A non-zero class code together with a drug code of 0 always indicates that a drug of that particular class has been applied, but that it is either unknown or not defined within this document. Even if a non-zero class and drug code are applied, a text description of the drug may also be sent in bytes 4 to ***. There are no standardized naming conventions. NOTE 3 There is no limit on the number of drugs which can be coded. NOTE 4 Within each class, subcode 9 shall be used for “other”.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
10, *cont.*		CLASS 1: DIGITALIS PREPARATION 0 – Unspecified 1 – Digoxin-Lanoxin 2 – Digitoxin-Digitalis CLASS 2: ANTIARRHYTHMIC 0 – Unspecified 1 – Dysopyramide 2 – Quinidine 3 – Procainamide 4 – Lidocaine 5 – Phenytoin 6 – Dilantin 7 – Amiodarone 8 – Tocainide 9 – Other 10 – Encainide 11 – Mexitil/Mexilitine 12 – Flecainide 13 – Lorcainide CLASS 3: DIURETICS 0 – Unspecified 1 – Thiazide 2 – Furosemide (Lasix) 3 – Potassium Chloride CLASS 4: ANTIHYPERTENSIVE 0 – Unspecified 1 – Clonidine 2 – Prasozin 3 – Hydralazine CLASS 5: ANTIANGINAL 0 – Unspecified 1 – Isosorbide 2 – Calcium Blockers 3 – Nitrates CLASS 6: ANTITHROMBOTIC AGENTS 0 – Unspecified 1 – Aspirin 2 – Coumadin 3 – Heparin 4 – Warfarin 5 – Streptokinase 6 – t-PA CLASS 7: BETA BLOCKERS 0 – Unspecified 1 – Propranolol 2 – Corgard 3 – Atenolol 4 – Metoprolol 5 – Pindolol 6 – Acebutolol CLASS 8: PSYCHOTROPIC 0 – Unspecified 1 – Tricyclic antidepressant 2 – Phenothiazide 3 – Barbiturate CLASS 9: CALCIUM BLOCKERS 0 – Unspecified 1 – Nifedipine 2 – Verapamil CLASS 10: ANTIHYPOTENSIVE 0 – Unspecified 1 – Asthmatic drug 2 – Aminophyline 3 – Isuprel CLASS 11: ANTICHOLESTEROL 0 – Unspecified 1 – Colestid 2 – Lovastatin 3 – Simvastatin 4 – Fibrates CLASS 12: ACE- INHIBITORS 0 – Unspecified 1 – Captopril
11	2	**Systolic blood pressure (binary)** Byte Contents 1 to 2 Binary: Systolic blood pressure in millimetres of mercury.
12	2	**Diastolic blood pressure** (binary) Byte Contents 1 to 2 Binary: Diastolic blood pressure in millimetres of mercury.
13	length	**Diagnosis or referral indication** (text characters) This field contains a text description of the patient's diagnosis or the referral indication.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
14	length	**Machine ID acquiring device** (binary bytes and text characters) This field uniquely identifies the device that acquired the ECG. It uses the following generic data structure for device characterization, which is also used in tag 15: Byte — Contents 1 to 2 — Binary: Institution number 3 to 4 — Binary: Department number 5 to 6 — Binary: Device ID 7 — Binary: Device type Value — Type 0 — Cart 1 — System (or Host)
		8 — Binary: set equal to 255 - see manufacturer character string at end of tag 14. NOTE — Legacy devices used this byte to specify a manufacturer code. These codes shall no longer be used except for identifying legacy files. For historical purposes the assigned codes were as follows: 0 – Unknown 1 – Burdick 2 – Cambridge 3 – Compumed 4 – Datamed 5 – Fukuda 6 – Hewlett-Packard 7 – Marquette electronics 8 – Mortara instruments 9 – Nihon Kohden 10 – Okin 11 – Quinton 12 – Siemens 13 – Spacelabs 14 – Telemed 15 – Hellige 16 – ESA-OTE 17 – Schiller 18 – Picker-Schwarzer 19 – et medical devices (ex Elettronica-Trentina) 20 – Zwönitz 21 to 99 – Reserved 100 – Other
		9 to 14 — Text characters: Text model description. Up to 5 bytes of text and NULL terminator.
		15 — Binary: SCP-ECG protocol revision number (the decimal point shall be deleted; version 1.0 becomes 10; the revisions shall as far as possible be backward compatible). This number shall exactly refer to the written document describing the actual protocol revision.
		16 — Binary: SCP-ECG protocol compatibility level (1 byte). Detailed specifications are given in Annex B (see B.4).

Table 4 (*continued*)

Tag	Length	Value (parameter data)
14, *cont.*		17 Binary: Language support code (1 byte). This bit map indicates the supported character sets:

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Meaning
0	x	x	x	x	x	x	x	8-bit ASCII-only
1	0	x	x	x	x	x	x	ISO-8859-1 Latin-1
1	1	0	0	0	0	0	0	ISO-8859-2 Latin-2 (Central and Eastern European)
1	1	0	1	0	0	0	0	ISO-8859-4 Latin-4 (Baltic)
1	1	0	0	1	0	0	0	ISO-8859-5 Cyrillic
1	1	0	1	1	0	0	0	ISO-8859-6 Arabic
1	1	0	0	0	1	0	0	ISO-8859-7 Greek
1	1	0	1	0	1	0	0	ISO-8859-8 Hebrew
1	1	0	0	1	1	0	0	ISO-8859-11 Thai
1	1	0	1	1	1	0	0	ISO-8859-15 Latin-9 (update to Latin-1, also called “Latin-0”)
1	1	1	0	0	0	0	0	ISO/IEC 10646
1	1	1	1	0	0	0	0	JIS X 0201-1976 (Japanese)
1	1	1	0	1	0	0	0	JIS X 0208-1997 (Japanese)
1	1	1	1	1	0	0	0	JIS X 0212-1990 (Japanese)
1	1	1	0	0	1	0	0	GB 2312-80 (Chinese)
1	1	1	1	0	1	0	0	KS C5601-1987 (Korean)
1	1	1	0	1	1	0	0	Reserved
1	1	1	1	1	1	0	0	Reserved
x	x	x	x	x	x	0	1	Reserved
x	x	x	x	x	x	1	0	Reserved
x	x	x	x	x	x	1	1	Reserved (except for following entry)
1	1	1	1	1	1	1	1	Manufacturer-specific

Table 4 (*continued*)

Tag	Length	Value (parameter data)
14, *cont.*		18 Binary: Capabilities of the ECG Device (1 byte bit map). This bit map indicates the supported functions: Bit / Contents: Reset (0) / Set (1) 0 to 3: Reserved / Reserved 4: No printing / Can print ECG reports 5: No analysis / Can interpret ECG 6: No storage / Can store ECG records 7(MSB): No acquisition / Can acquire ECG data 19 Binary: AC mains frequency environment (1 byte): Value / Meaning 0: Unspecified 1: 50 Hz 2: 60 Hz 20 to 35 Reserved for future use. 36 Binary: Length of the string for analysing program revision number. Byte 36 shall be equal to or greater than 1. The character strings following byte 36 are required. If a particular character string is empty, a single NULL is required to be present. 37 to *** Character string: Analysing program revision number. The string has to be NULL terminated. *** to *** Character string: Serial number of the acquisition device. The character string has to be NULL terminated. *** to *** Character string: Acquisition device system software identifier. The character string has to be NULL terminated. *** to *** Character string: Acquisition device SCP implementation software identifier (maximum 24 characters plus NULL terminator). The character string has to be NULL terminated. *** to *** Character string: Manufacturer of the acquisition device. Contains the manufacturer's registered trade name. The character string has to be NULL terminated.
15	length	**Machine ID analysing device** (binary bytes and text characters) This field uniquely identifies the device that analysed the ECG (if other than the acquiring cardiograph). The format of this field is identical to that utilized by tag 14 above.
16	length	**Acquiring institution description** (text characters) This field provides a text description of the institution where the ECG was acquired.
17	length	**Analyzing institution description** (text characters) This field provides a text description of the institution where the ECG was analysed.
18	length	**Acquiring department description** (text characters) This field provides a text description of the department where the ECG was acquired.
19	length	**Analyzing department description** (text characters) This field provides a text description of the department where the ECG was analysed.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
20	length	**Referring physician** (text characters) This field provides a text description of the referring physician.
21	length	**Latest confirming physician** (text characters) This field provides a text description of the latest confirming physician.
22	length	**Technician description** (text characters) This field provides a text description of the technician.
23	length	**Room description** (text characters) This field provides a text description of the room where the ECG was recorded.
24	1	**Stat code** (binary) Byte Contents 1 Binary: Level of emergency. Value 0 refers to “routine” and higher values to increasing levels of emergency as defined by the user. For this, codes in the range of 1 to 10 are recommended.
25	4	**Date of acquisition** (binary) This field has the following format: Byte Contents 1 to 2 Binary: year (full integer notation, as in 1990). 3 Binary: month (range 01 to 12; 01 = January). 4 Binary: day (range 01 to 31).
26	3	**Time of acquisition** (binary) This field has the following format: Byte Contents 1 Binary: hours (range 0 to 23). 2 Binary: minutes (range 0 to 59). 3 Binary: seconds (range 0 to 59). The time of acquisition shall be expressed as local time in the time zone of acquisition (see tag 34).
27	2	**Baseline filter** (binary) This field contains the “cut-off” frequency (–3 db) of the high pass baseline filter in units of (1/100) Hertz.
28	2	**Low-pass filter** (binary) This field contains the “cut-off” frequency (-3 db) of the low pass filter in units of Hertz.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
29	1	**Filter bit map** (binary) This field indicates if other filters, which were not defined within tag 27 and 28, have been used during the processing of the ECG. The definition of these bits are: 0 60 Hertz notch filter 1 50 Hertz notch filter 2 Artifact filter 3 Baseline filter (e.g. adaptive filter or spline filter) 4 to 7 Undefined If all bits are zero then the filter setting was not specified.
30	length	**Free text field** (text characters) This field permits free text comments to be carried along with the ECG.
31	length	**Sequence number** (text characters) ECG sequence number.
32	length	**Medical history codes** (binary) This field contains a description of the patient's clinical problems and diagnoses. There is no limit on the number of diagnoses. Each diagnosis shall be represented by 1 byte. Byte 1 is used to designate the Medical History Code Table which is applied. If Byte 1 is not equal to zero, the designated Medical History Code Table is undefined. In case this byte is equal to zero (0) then the following set of codes apply: <u>Value</u> <u>Contents</u> 0 Diagnoses or clinical problems not specified 1 Apparently healthy 10 Acute myocardial infarction 11 Myocardial infarction 12 Previous myocardial infarction 15 Ischemic heart disease 18 Peripheral vascular disease 20 Cyanotic congenital heart disease 21 Acyanotic congenital heart disease 22 Valvular heart disease 25 Hypertension 27 Cerebrovascular accident 30 Cardiomyopathy 35 Pericarditis 36 Myocarditis 40 Post-operative cardiac surgery 42 Implanted cardiac pacemaker 45 Pulmonary embolism 50 Respiratory disease 55 Endocrine disease 60 Neurological disease 65 Alimentary disease 70 Renal disease 80 Pre-operative general surgery 81 Post-operative general surgery 90 General medical 100 Other 128 to 255 Manufacturer specific The missing numbers in the series from 1 to 100 have been reserved for future extension of some categories.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
33	2	Electrode configuration code (binary) This field is used to identify the placement and system of electrodes: Byte — Contents 1 — Binary: Code representing the definitions for system of electrode placements for 12-lead ECG (standard, Mason-Likar, Omnitrode, etc.). Value — Electrode placement system 0 — Unspecified Note: carts that do not record the electrode placement information should use 0. 1 — Standard 12-lead positions: RA, RL, LA, and LL are placed at limb extremities. V1 to V6 at standard positions on the chest. All electrodes are placed individually. 2 — RA, RL, LA, and LL are placed on the torso (Mason-Likar positions). V1 to V6 are placed at standard positions on the chest. All electrodes are placed individually. 3 — RA, RL, LA, and LL are placed on the torso (Mason-Likar positions). These limb electrodes are individually placed. V1 to V6 are placed on the chest as part of a single electrode pad (V1 to V6 are NOT placed individually). 4 — RA, RL, LA, LL, and V1 to V6 (all electrodes) are placed on the chest in a single electrode pad (such as Omnitrode). (None of the electrodes is placed individually). 5 — 12-lead ECG is derived from Frank XYZ leads. 6 — 12-lead ECG is derived from non-standard leads. 7 to 255 Undefined now. Reserved for later use. 2 — Binary: Code representing the definitions for system of electrode placements for XYZ leads such as Frank, Cube, McFee-Parungao, Bipolar, etc. [see chapter 1 of Vectorcardiography by Alberto Benchimol (Williams & Wilkins, Baltimore, 1973) for location of electrodes on the torso and weighting resistors]. Value — Electrode placement system 0 — Unspecified Note: carts that do not record the electrode placement information should use 0. 1 — Frank lead system (Frank, 1956; 13:737). 2 — McFee-Parungao lead system (see Benchimol, Vectorcardiography, Williams & Wilkins, Baltimore, 1973, fig. 1.6 on page 6). 3 — Cube lead system (Grishman et al, *Amer. Heart J*. 1951; **41**, p. 483). 4 — Bipolar uncorrected XYZ lead system. 5 — Pseudo-orthogonal XYZ lead system (as used in Holter recording). 6 — XYZ leads derived from standard 12-lead ECG. 7 to 255 Undefined now. Reserved for later use.

Table 4 (*continued*)

Tag	Length	Value (parameter data)
34	length	**Date time zone** (binary bytes and text characters) The contents of this tag identify the global time zone in which the acquired data were obtained, thus allowing the date/time value specified by tags 25 and 26 to be converted to any time zone (e.g. UTC). The following parameter bytes of this tag provide three ways to indicate time zone. Byte — Name — Type — Notes 1 to 2 — Offset — signed binary — Time zone specified as an offset from UTC in minutes [Note 1]. 3 to 4 — Index — unsigned binary — Time zone specified by a manufacturer-defined mapping (until a consensus mapping is defined using bytes 1 to 1 000) using this value as a lookup-table index [Note 2]. Value — Meaning 0 = Index not used 1 to 1 000 = Reserved for future use 1 001 to 32 766 = Manufacturer-specific 32 767 = Reserved 5 to *** — Description — Text string — Time zone specified by a null-terminated string [Note 3]. NOTE 1 Allowable values for offset are -780 to 780 (i.e., ± 13 h) and hexadecimal 0x7FFF. 0x7FFF indicates that the field is not initialized or is unused. If the offset field contains an allowed value other than 0x7FFF, the index and description fields are considered redundant and may be ignored. NOTE 2 The index value specifies time zone only if the offset value is 0x7FFF. An index value of zero indicates that the field is not used or not initialized. Use of this byte as currently defined is manufacturer-specific. NOTE 3 The description field specifies time zone only if the offset value is 0x7FFF. This string should be in the format of the TZ environment variable as standardized by Posix (Unix). Reference: 'C/C++' language subroutine name tzset(), environment variable “TZ”, and associated data structures. The description field must be 1 byte at a minimum (i.e., the null terminator). NOTE 4 If time zone is not defined for the device, Tag 34 may be omitted from the data record. Similarly, an instance of Tag 34 containing values for the offset = 0x7FFF, index = 0, and description = null terminator is also allowed if time zone is not defined.
35	length	**Free text medical history** (text characters) This field permits free text for entering the medical history.
255	0	**None (demographic section terminator).**

5.5 Huffman tables – Section 2

5.5.1 If present, the section shall start with a “Section ID Header” as defined in 5.2.7.

5.5.2 The scheme for signal compression presented below is based on the Huffman method of encoding. This method is not the only one possible, but is the recommended one.

5.5.3 This section of the ECG record contains the definition of the Huffman Code Tables that were used to encode the ECG. The provision of a number of tables permits optimum encoding of the data (e.g. the reference beats and rhythm data will probably use different tables). It shall be assumed that the encoded data within each entity shall be encoded using table # 1 (i.e. the first table defined within this Section 2). Escape codes should be provided within each table to enable a change to another table.

5.5.4 The following basic values are used:

1) The fundamental sample time, as defined in the section containing the coded data (i.e. Section 5 “Reference Beat data” and Section 6 “Rhythm data”).

2) The fundamental of ECG amplitude, as defined in the section containing the coded data (i.e. Section 5 “Reference Beat data” and Section 6 “Rhythm data”).

The structure of the data part of this Section is then as follows:

Byte	Contents
1 to 2	Number of Huffman Tables defined (if 19 999 then the default table, defined in C.3.7.6, is used).
3 to 4	Number of code structures in table # 1.
5 to ***	The structures defining each code in table # 1. Each structure has the following layout:

1 byte	Number of bits in prefix
1 byte	Number of bits in entire code
1 byte	Table mode switch

Value	Content
0	Switch to another Huffman table
1	Huffman coding if # of bits in prefix = # of bits in entire code
1	Original data if # of bits in prefix < # of bits in entire code

2 bytes	Base value represented by base code (in AVM units).
4 bytes	Base code - 1st bit in code represented by least significant bit of the 4 byte area.

***+1 to ***+2 Number of structures in table # 2.

***+3 to *** Structures representing table # 2.

etc.

5.5.5 The Huffman codes have been defined to permit a single structure described above to specify a series of consecutive amplitude values. The “prefix” mentioned above is common to all of the codes describing the consecutive values - the remaining bit field changes by 1 LSB in incrementing through the indicated range.

Structures that define the Huffman code for just one value shall have no remainder and hence the prefix length shall equal the total code length.

An example byte structure of the Huffman code is then:

Table 5 — Example byte structure (Huffman code)

	MSB							**LSB**
Received Byte 1	P1	P2	P3	P4	C1	C2	C3	C4
Received Byte 2	C5	C6	P5	P6	P7	C7	C8	C9
Received Byte 3	C10	C11	P8	P9	P10	C12	C13	C14
etc.	…	…	…	…	…	…	…	…

NOTE This represents the following coded values:

P1 P2 P3 P4 C1 C2 C3 C4 C5 C6

4 bit prefix with total code length of 10.

P5 P6 P7 C7 C8 C9 C10 C11

3 bit prefix with total code of 8.

P8 P9 P10 C12 C13 C14

3 bit prefix with total code length of 6.

It shall be seen that the "picking" of bits in a given byte proceeds from the most significant bit to the least significant bit and that the bytes are processed in the order received.

5.5.6 Escape codes, i.e. codes that shall dictate a change of Huffman table, shall include a zero (0) value for the "Table mode switch". The "Base Value" shall then contain the number of the table to which a switch is desired.

5.5.7 An overview of the data part of this section is presented in Figure 5.

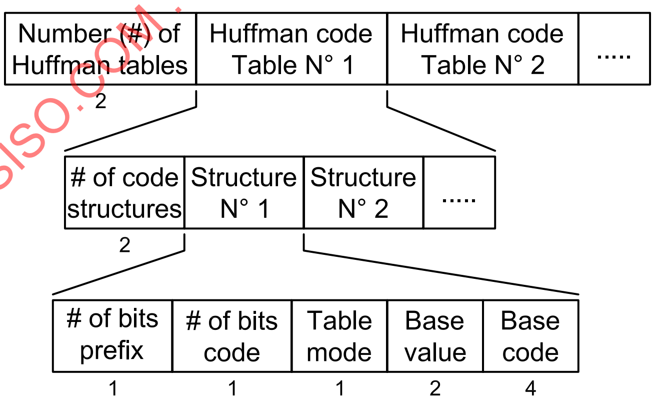


Figure 5 — Overview of the data part of the Huffman tables section

5.6 ECG lead definition – Section 3

5.6.1 This section defines the leads that are transmitted, together with some general administrative information.

5.6.2 If present, the section shall start with a "Section ID Header" as defined in 5.2.7.

5.6.3 The section data part is defined below:

Byte	Contents	
1	Number of leads enclosed	
2	Flag byte:	
	Bit 0 (LSB)	Set = Reference beat subtraction used for compression Reset = Reference beat subtraction not used for compression
	Bit 1	Reserved
	Bit 2	Set = Leads all simultaneously recorded Reset = Leads not all simultaneously recorded
	Bits 3 to 7	The number of simultaneously recorded leads
3 to 11	Detail for first lead (see 5.6.4)	
12 to 20	Detail for second lead (see 5.6.4)	
etc.		

In case not all leads are recorded simultaneously, the leads shall be presented in groups corresponding to those recorded simultaneously.

EXAMPLE Three leads are recorded simultaneously: e.g. first leads I, aVF, V2; second leads X, Y, Z, etc. Lead details should be listed in above listed order: Lead I in the first segment (3 to 11), Lead aVF in the second segment (12 to 20), Lead V2 in the third segment (21 to 29), Lead X in the fourth segment (30 to 38), etc.

5.6.4 The detailed information for each lead is as follows:

Byte	**Contents**
1 to 4	(Unsigned) Starting sample number
5 to 8	(Unsigned) Ending sample number
9	Lead identification. The numbering scheme shown in Table 6 shall be used.

Table 6 — Lead identification codes

SCP-ECG name	SCP-ECG Code	Description	VITAL[12] Ref ID MDC_ECG_LEAD_xxx
	0	Unspecified lead	MDC_ECG_LEAD_CONFIG
I	1	Lead I	MDC_ECG_LEAD_I
II	2	Lead II	MDC_ECG_LEAD_II
V1	3	V1	MDC_ECG_LEAD_V1
V2	4	V2	MDC_ECG_LEAD_V2
V3	5	V3	MDC_ECG_LEAD_V3
V4	6	V4	MDC_ECG_LEAD_V4
V5	7	V5	MDC_ECG_LEAD_V5
V6	8	V6	MDC_ECG_LEAD_V6
V7	9	V7	MDC_ECG_LEAD_V7
V2R[a]	10	V2R	MDC_ECG_LEAD_V2R

Table 6 (*continued*)

SCP-ECG name	SCP-ECG Code	Description	VITAL[12] Ref ID MDC_ECG_LEAD_xxx
V3R	11	V3R	MDC_ECG_LEAD_V3R
V4R	12	V4R	MDC_ECG_LEAD_V4R
V5R	13	V5R	MDC_ECG_LEAD_V5R
V6R	14	V6R	MDC_ECG_LEAD_V6R
V7R	15	V7R	MDC_ECG_LEAD_V7R
X	16	X[b]	MDC_ECG_LEAD_X
Y	17	Y[b]	MDC_ECG_LEAD_Y
Z	18	Z[b]	MDC_ECG_LEAD_Z
CC5[c]	19	CC5, per V5 and V5R placement	MDC_ECG_LEAD_CC5
CM5	20	CM5, per V5 placement	MDC_ECG_LEAD_CM5
LA	21	Left arm	MDC_ECG_LEAD_LA
RA	22	Right arm	MDC_ECG_LEAD_RA
LL	23	Left leg	MDC_ECG_LEAD_LL
fI[d]	24	fI	MDC_ECG_LEAD_fI
fE	25	fE	MDC_ECG_LEAD_fE
fC	26	fC	MDC_ECG_LEAD_fC
fA	27	fA	MDC_ECG_LEAD_fA
fM	28	fM	MDC_ECG_LEAD_fM
fF	29	fF	MDC_ECG_LEAD_fF
fH	30	fH	MDC_ECG_LEAD_fH
dI	31	Derived lead I	MDC_ECG_LEAD_dI
dII	32	Derived lead II	MDC_ECG_LEAD_dII
dV1	33	Derived lead V1	MDC_ECG_LEAD_dV1
dV2	34	Derived lead V2	MDC_ECG_LEAD_dV2
dV3	35	Derived lead V3	MDC_ECG_LEAD_dV3
dV4	36	Derived lead V4	MDC_ECG_LEAD_dV4
dV5	37	Derived lead V5	MDC_ECG_LEAD_dV5
dV6	38	Derived lead V6	MDC_ECG_LEAD_dV6
dV7	39	Derived lead V7	
dV2R	40	Derived lead V2R	
dV3R	41	Derived lead V3R	
dV4R	42	Derived lead V4R	
dV5R	43	Derived lead V5R	
dV6R	44	Derived lead V6R	
dV7R	45	Derived lead V7R	
dX	46	Derived lead X	
dY	47	Derived lead Y	
dZ	48	Derived lead Z	
dCC5	49	Derived lead CC5	
dCM5	50	Derived lead CM5	
dLA	51	Derived lead LA	
dRA	52	Derived lead RA	
dLL	53	Derived lead LL	
dfI	54	Derived lead fI	
dfE	55	Derived lead fE	
dfC	56	Derived lead fC	

Table 6 (*continued*)

SCP-ECG name	SCP-ECG Code	Description	VITAL[12] Ref ID MDC_ECG_LEAD_xxx
dfA	57	Derived lead fA	
dfM	58	Derived lead fM	
dfF	59	Derived lead fF	
dfH	60	Derived lead fH	
III	61	Lead III	MDC_ECG_LEAD_III
aVR	62	aVR, augmented voltage, right	MDC_ECG_LEAD_AVR
aVL	63	aVL, augmented voltage, left	MDC_ECG_LEAD_AVL
aVF	64	aVF, augmented voltage, foot	MDC_ECG_LEAD_AVF
aVRneg	65	aVRneg	MDC_ECG_LEAD_AVRneg
V8	66	V8	MDC_ECG_LEAD_V8
V9	67	V9	MDC_ECG_LEAD_V9
V8R	68	V8R	MDC_ECG_LEAD_V8R
V9R	69	V9R	MDC_ECG_LEAD_V9R
D	70	D (Nehb – Dorsal)	MDC_ECG_LEAD_D
A	71	A (Nehb – Anterior)	MDC_ECG_LEAD_A
J	72	J (Nehb – Inferior)	MDC_ECG_LEAD_J
Defib	73	Defibrillator lead: anterior-lateral	MDC_ECG_LEAD_DEFIB
Extern	74	External pacing lead: anterior-posterior	MDC_ECG_LEAD_EXTERN
A1	75	A1 (Auxiliary unipolar lead #1)	MDC_ECG_LEAD_A1
A2	76	A2 (Auxiliary unipolar lead #2)	MDC_ECG_LEAD_A2
A3	77	A3 (Auxiliary unipolar lead #3)	MDC_ECG_LEAD_A3
A4	78	A4 (Auxiliary unipolar lead #4)	MDC_ECG_LEAD_A4
dV8	79	Derived lead V8	
dV9	80	Derived lead V9	
dV8R	81	Derived lead V8R	
dV9R	82	Derived lead V9R	
dD	83	Derived lead D (Nehb – Dorsal)	
dA	84	Derived lead A (Nehb – Anterior)	
dJ	85	Derived lead J (Nehb – Inferior)	
Chest	86	Chest lead	MDC_ECG_LEAD_C
V	87	Precordial lead	MDC_ECG_LEAD_V
VR	88	VR, nonaugmented voltage, vector of RA	MDC_ECG_LEAD_VR
VL	89	VL, nonaugmented voltage, vector of LA	MDC_ECG_LEAD_VL
VF	90	VF, nonaugmented voltage, vector of LL	MDC_ECG_LEAD_VF
MCL	91	Modified chest lead (left arm indifferent)	MDC_ECG_LEAD_MCL
MCL1	92	MCL, per V1 placement	MDC_ECG_LEAD_MCL1
MCL2	93	MCL, per V2 placement	MDC_ECG_LEAD_MCL2
MCL3	94	MCL, per V3 placement	MDC_ECG_LEAD_MCL3
MCL4	95	MCL, per V4 placement	MDC_ECG_LEAD_MCL4
MCL5	96	MCL, per V5 placement	MDC_ECG_LEAD_MCL5
MCL6	97	MCL, per V6 placement	MDC_ECG_LEAD_MCL6
CC	98	Chest lead (symmetric placement)	MDC_ECG_LEAD_CC
CC1	99	CC1, per V1 and V1R placement	MDC_ECG_LEAD_CC1
CC2	100	CC2, per V2 and V2R placement	MDC_ECG_LEAD_CC2
CC3	101	CC3, per V3 and V3R placement	MDC_ECG_LEAD_CC3
CC4	102	CC4, per V4 and V4R placement	MDC_ECG_LEAD_CC4
CC6	103	CC6, per V6 and V6R placement	MDC_ECG_LEAD_CC6
CC7	104	CC7, per V7 and V8R placement	MDC_ECG_LEAD_CC7

Table 6 (*continued*)

SCP-ECG name	SCP-ECG Code	Description	VITAL[12] Ref ID MDC_ECG_LEAD_xxx
CM	105	Chest-manubrium	MDC_ECG_LEAD_CM
CM1	106	CM1, per V1 placement	MDC_ECG_LEAD_CM1
CM2	107	CM2, per V2 placement	MDC_ECG_LEAD_CM2
CM3	108	CM3, per V3 placement	MDC_ECG_LEAD_CM3
CM4	109	CM4, per V4 placement	MDC_ECG_LEAD_CM4
CM6	110	CM6, per V6 placement	MDC_ECG_LEAD_CM6
dIII	111	Derived lead III	MDC_ECG_LEAD_dIII
daVR	112	Derived lead aVR	MDC_ECG_LEAD_daVR
daVL	113	Derived lead aVL	MDC_ECG_LEAD_daVL
daVF	114	Derived lead aVF	MDC_ECG_LEAD_daVF
daVRneg	115	Derived lead aVRneg	
dChest	116	Derived lead Chest	
dV	117	Derived lead V	
dVR	118	Derived lead VR	
dVL	119	Derived lead VL	
dVF	120	Derived lead VF	
CM7	121	CM7, per V7 placement	MDC_ECG_LEAD_CM7
CH5	122	CH5	MDC_ECG_LEAD_CH5
CS5	123	Negative: right infraclavicular fossa	MDC_ECG_LEAD_CS5
CB5	124	Negative: low right scapula	MDC_ECG_LEAD_CB5
CR5	125	CR5	MDC_ECG_LEAD_CR5
ML	126	ML, modified limb lead, ~ Lead II	MDC_ECG_LEAD_ML
AB1	127	AB1 (auxiliary bipolar lead #1)	MDC_ECG_LEAD_AB1
AB2	128	AB2 (auxiliary bipolar lead #2)	MDC_ECG_LEAD_AB2
AB3	129	AB3 (auxiliary bipolar lead #3)	MDC_ECG_LEAD_AB3
AB4	130	AB4 (auxiliary bipolar lead #4)	MDC_ECG_LEAD_AB4
ES	131	EASI™ ES[e]	MDC_ECG_LEAD_ES
AS	132	EASI AS	MDC_ECG_LEAD_AS
AI	133	EASI AI	MDC_ECG_LEAD_AI
S	134	EASI upper sternum lead	MDC_ECG_LEAD_S
dDefib	135	Derived lead Defib: Defibrillator lead: anterior-lateral	
dExtern	136	Derived lead Extern: External pacing lead: anterior-posterior	
dA1	137	Derived lead A1 (Auxiliary unipolar lead #1)	
dA2	138	Derived lead A2 (Auxiliary unipolar lead #2)	
dA3	139	Derived lead A3 (Auxiliary unipolar lead #3)	
dA4	140	Derived lead A4 (Auxiliary unipolar lead #4)	
dMCL1	141	Derived lead MCL1: MCL, per V1 placement	
dMCL2	142	Derived lead MCL2: MCL, per V2 placement	
dMCL3	143	Derived lead MCL3: MCL, per V3 placement	
dMCL4	144	Derived lead MCL4: MCL, per V4 placement	
dMCL5	145	Derived lead MCL5: MCL, per V5 placement	
dMCL6	146	Derived lead MCL6: MCL, per V6 placement	
RL	147	Right leg	MDC_ECG_LEAD_RL
CV5RL	148	Canine, fifth right intercostal space near the edge of the sternum at the most curved part of the costal cartilage	MDC_ECG_LEAD_CV5RL

Table 6 (*continued*)

SCP-ECG name	SCP-ECG Code	Description	VITAL[12] Ref ID MDC_ECG_LEAD_xxx
CV6LL	149	Canine, sixth left intercostal space near the edge of the sternum at the most curved part of the costal cartilage	MDC_ECG_LEAD_CV6LL
CV6LU	150	Canine, sixth left intercostal space at the costochondral junction	MDC_ECG_LEAD_CV6LU
V10	151	Canine, over dorsal spinous process of the seventh thoracic vertebra	MDC_ECG_LEAD_V10
dMCL	152	Derived lead MCL: Modified chest lead (left arm indifferent)	
dCC	153	Derived lead CC: Chest lead (symmetric placement)	
dCC1	154	Derived lead CC1, per V1 and V1R placement	
dCC2	155	Derived lead CC2, per V2 and V2R placement	
dCC3	156	Derived lead CC3, per V3 and V3R placement	
dCC4	157	Derived lead CC4, per V4 and V4R placement	
dCC6	158	Derived lead CC6, per V6 and V6R placement	
dCC7	159	Derived lead CC7, per V7 and V8R placement	
dCM	160	Derived lead CM Chest-manubrium	
dCM1	161	Derived lead CM1, per V1 placement	
dCM2	162	Derived lead CM2, per V2 placement	
dCM3	163	Derived lead CM3, per V3 placement	
dCM4	164	Derived lead CM4, per V4 placement	
dCM6	165	Derived lead CM6, per V6 placement	
dCM7	166	Derived lead CM7, per V7 placement	
dCH5	167	Derived lead CH5	
dCS5	168	Derived lead CS5: negative: right infraclavicular fossa	
dCB5	169	Derived lead CB5: negative: low right scapula	
dCR5	170	Derived lead CR5	
dML	171	Derived lead ML, modified limb lead, ~ Lead II	
dAB1	172	Derived lead AB1 (auxiliary bipolar lead #1)	
dAB2	173	Derived lead AB2 (auxiliary bipolar lead #2)	
dAB3	174	Derived lead AB3 (auxiliary bipolar lead #3)	
dAB4	175	Derived lead AB4 (auxiliary bipolar lead #4)	
dES	176	Derived lead ES: EASI[e] ES	
dAS	177	Derived lead AS: EASI AS	
dAI	178	Derived lead AI: EASI AI	
dS	179	Derived lead S: EASI upper sternum lead	
dRL	180	Derived lead RL: right leg	
dCV5RL	181	Derived lead CV5RL: Canine, fifth right intercostal space near the edge of the sternum at the most curved part of the costal cartilage	
dCV6LL	182	Derived lead CV6LL: Canine, sixth left intercostal space near the edge of the sternum at the most curved part of the costal cartilage	
dCV6LU	183	Derived lead CV6LU: Canine, sixth left intercostal space at the costochondral junction	

Table 6 (*continued*)

SCP-ECG name	SCP-ECG Code	Description	VITAL[12] Ref ID MDC_ECG_LEAD_xxx
dV10	184	Derived lead V10: Canine, over dorsal spinous process of the seventh thoracic vertebra	
	185 to 199	Reserved for future expansion	
	200 to 255	Manufacturer specific	

[a] V2R is identical to lead V1. Similarly, lead V1R (not listed in the lead table) is identical to lead V2.

[b] Leads X, Y and Z can be recorded by an orthogonal system, such as Frank or McFee lead systems, etc.

[c] CM5, CH5, CS5, CC5, CB5, CR5 bipolar leads used in conjunction with stress testing. Macfarlane, Volume 1, page 323. [CX5?].

[d] Frank leads indicated by 'f' for clarity and label uniqueness.

[e] EASI™ trademark owned by Philips, invented by Dr. Gordon Dower. Leads: S, upper sternum; E, lower sternum (Frank lead fE); A, under left arm, above V6 (Frank lead fA); I, under right arm, above V6R (Frank lead fI).

NOTE 1 Extension of the lead numbering scheme may be done in future revisions of the protocol.

NOTE 2 Users of this part of ISO 11073 are advised to refer to documents [6], [12] and other current standards in the ISO/IEEE nomenclature series to avoid unintended duplication in these code ranges.

5.6.5 The sample numbering shall start with sample number 1 and refers to all leads recorded simultaneously. In order to convert these values to time, the sampling rate of the proper data section (see 5.8.3) should be consulted.

For example, if 8 leads (I, II, V1 to V6) are recorded simultaneously over 10 seconds at 500 samples/s and stored this way, then each lead begins with sample number 1 and ends with sample number 5 000.

If the leads are recorded in groups of three, for example over 2,5 s at 500 samples/s, then leads I, II and III begin with sample number 1 to sample 1 250, and leads aVR, aVL and aVF begin with sample number 1 251.

5.6.6 An overview of the data part of this section is presented in Figure 6.

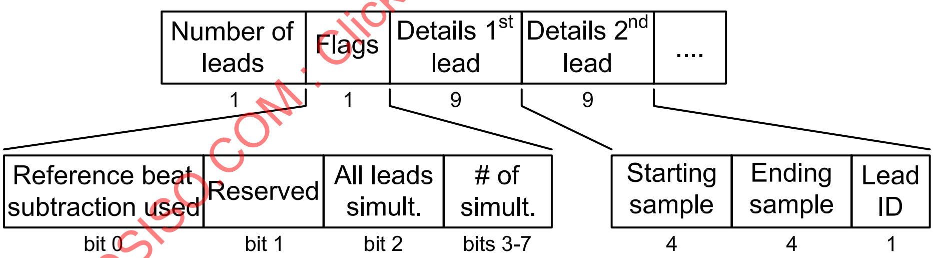


Figure 6 — Overview of the data part of the ECG lead definition section

5.7 QRS locations, reference beat subtraction zones and protected areas – Section 4

5.7.1 If present, this section defines the locations and width of the various QRS complexes. For a definition of reference beats, beat types and the significance of reference beat type 0, see 5.1.11. For a detailed description of the overall process, see Annex C.

5.7.2 The section shall start with a "Section ID Header" as defined in 5.2.7.

5.7.3 The header area of the section data part defines certain quantities that are common to the type 0 reference beat for all leads. The remaining data indicate the reference beat type and location of each QRS relative to the “residual” signal. The section data part header area has the following contents:

Byte **Contents**

1 to 2 Length of reference beat type 0 data in milliseconds.

NOTE The number of samples N is obtained by dividing the length L of the reference beat (in milliseconds) by the sample time interval SI (in microseconds, see 5.9.3, bytes 3 to 4) using the following equation:

$$N = \text{truncation}\left[\frac{(1\,000\,\mu\text{s/ms} \times L)}{\text{SI}}\right]$$

The manufacturer shall assign a length (in bytes 1 to 2) such that when this equation is used, the intended number of samples in the reference beat is obtained. For example, 1 000 ms of data at 2 000 µs per sample results in an N of 500 samples.

3 to 4 (Unsigned) sample number of the fiducial (QRS trigger point) relative to the beginning of reference beat type 0. This location is abbreviated as fcM in Annex C. The first sample is numbered 1.

5 to 6 Total number of QRS complexes within the entire ECG record.

5.7.4 The following information on location of reference beat subtraction zones is stored, consisting of one block of 14 bytes for each QRS complex. The total number of blocks is equal to the number of QRSs stored in 5.7.3 bytes 5 to 6.

Byte **Contents**

1 to 2 Beat type of 1st QRS (see 5.1.11 for definition of “Beat type”)

3 to 6 (Unsigned) sample number[1)] on residual data for the start of subtraction/addition of reference beat 0 for 1st QRS, if the QRS is of type 0, otherwise a value of 0 (zero).[2), 3)]

7 to 10 (Unsigned) sample number[1)] on residual data for location of fiducial point for 1st QRS.[4)] This location is abbreviated as fc(1) in Annex C.

11 to 14 (Unsigned) sample number[1)] on residual data for end of subtraction/addition of reference beat 0 for 1st QRS, if the QRS is of type 0, otherwise a value of 0 (zero).[2,3)]

15 to 16 Beat type of 2nd QRS etc.

1) All sample numbers in this clause refer to the original samples before processing them for decimation and/or compression. The first sample of the original data is numbered 1.

2) If bytes 1 to 2 indicate reference beat type 0, then bytes 3 to 6 and 11 to 14 bound the area around the QRS for reference beat type 0 subtraction or addition, as specified and illustrated in Annex C. These locations are abbreviated in Annex C as SB(k) and SE(k), respectively.

3) If bytes 1 to 2 indicate a reference beat type other than 0, then reference beat subtraction is not used, in which case bytes 3 to 6 and 11 to 14 contain 0 (zero).

4) 5.7.4 and 5.7.5 may also be used to indicate location of the protected zones in case bimodal compression is used but not reference beat subtraction. In this case, 5.7.4 bytes 3 to 6 and 11 to 14 shall be set to zero.

5.7.5 The following information on location of protected areas (QRS complexes) is stored, consisting of one block of 8 bytes for each QRS complex. The total number of blocks is equal to the number of QRSs stored in 5.7.3 bytes 5 to 6.[4)]

Byte	Contents
1 to 4	(Unsigned) sample number[1)] on residual data for the start of the protected area of the 1st QRS. This location is abbreviated as QB(1) in Annex C.
5 to-8	(Unsigned) sample number[1)] on residual data for the end of the protected area of the 1st QRS. This location is abbreviated as QE(1) in Annex C.
9 to 12	(Unsigned) sample number[1)] on residual data for the protected area of the start of the 2nd QRS. This location is abbreviated as QB(2) in Annex C.
13 to 16	(Unsigned) sample number[1)] on residual data for the end of the protected area of the 2nd QRS. This location is abbreviated as QE(2) in Annex C.
etc.	

5.7.6 An overview of the data part of this section is presented in Figure 7.

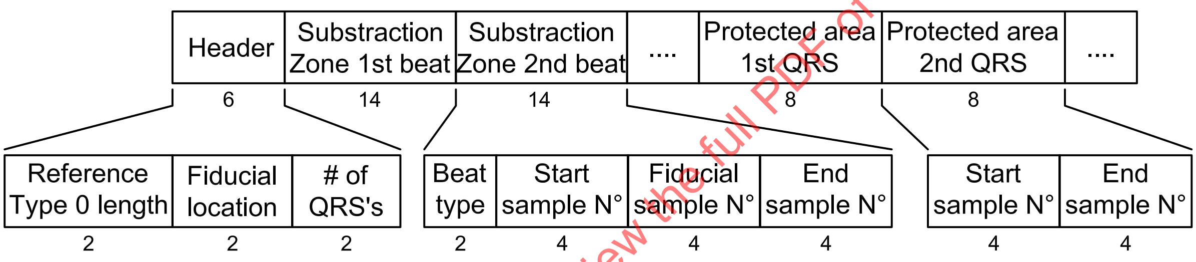


Figure 7 — Overview of the data part of Section 4

5.8 Encoded type 0 reference beat data – Section 5

5.8.1 This section provides details of reference beat type 0. For a definition of reference beats, reference beat types, and the significance of reference beat type 0, see 5.1.11. For a detailed description of the overall process see Annex C.

5.8.2 If present, the section shall start with a “Section ID Header” as defined in 5.2.7.

5.8.3 The section data part begins with a header that has the following format:

Byte **Contents**

1 to 2 Multiplier for amplitude value (AVM). This operates as follows:

The amplitude value multiplier is expressed in nanovolt (1×10^{-9} V).

Example: 1 250 → 1 amplitude quantum = 1,250 µV
2 441 → 1 amplitude quantum = 2,441 µV

3 to 4 The sample time interval for this section in microseconds (1×10^{-6} s).

Example: 4 000 → 250 samples/s.
1 250 → 800 samples/s.

5 This value indicates the encoding of the sample data as follows:

0 = Real (zero difference) data used for reference beat 0 data.

1 = First difference data used for reference beat 0 data.

2 = Second difference data used for reference beat 0 data.

6 Reserved.

NOTE 1 Difference data are defined as: [sample value (difference) for time t] – (sample value (difference) for time t–1).

NOTE 2 For the first two samples in each lead, second differences are not computed in the SCP-ECG protocol. The original value amplitudes of these samples are retained. The first sample value is similarly retained in the encoded data stream using first differences.

NOTE 3 An example of the encoded results using second differences is given in Table 8 for a series of eight sample data.

NOTE 4 An example of the encoded results using first differences is given in Table 9 using the same series of eight sample data.

Table 7 — Difference data

Original data	First differences	Second differences
X(1)	D1(1)=X(1)	D2(1)=X(1)
X(2)	D1(2)=X(2)-X(1)	D2(2)=X(2)
X(3)	D1(3)=X(3)-X(2)	D2(3)=D1(3)-D1(2)=X(3)-2*X(2)+X(1)
X(4)	D1(4)=X(4)-X(3)	D2(4)=D1(4)-D1(3)=X(4)-2*X(3)+X(2)

So the general formula for the first difference is as follows: $D1(n) = X(n) - X(n-1)$.

The general formula for the second difference is as follows: $D2(n) = X(n) - 2*X(n-1) + X(n-2)$.

Decoding of the 2nd difference data is performed using the following formula: $X(n) = D2(n) + 2*X(n-1) - X(n-2)$.

Table 8 — Example of encoded results using 2nd differences

Sample Number:	n	1	2	3	4	5	6	7	8
Sample Value:	X(n)	10	12	13	15	18	22	20	15
2nd Difference:	D2(n)	–	–	–1	1	1	1	–6	–3
Encoded data:		10	12	–1	1	1	1	–6	–3

Table 9 — Example of encoded results using 1st differences

Sample Number:	n	1	2	3	4	5	6	7	8
Sample Value:	X(n)	10	12	13	15	18	22	20	15
1st Difference:	D1(n)	–	2	1	2	3	4	–2	–5
Encoded data:		10	2	1	2	3	4	–2	–5

5.8.4 The section data part contains the byte lengths of the encoded leads. Its format is as follows:

Byte	Contents
1 to 2	(Unsigned) number of bytes in compressed reference beat 0 data for first encoded lead.
3 to 4	(Unsigned) number of bytes in compressed reference beat 0 data for second encoded lead.
etc.	

5.8.5 The encoded reference beat 0 data then follows. If Section 2 has been provided, the data is coded as a series of Huffman codes taken from Section 2. The leads are encoded in the order specified in Section 3. If Section 2 is not provided, ECG data (either differenced or non-differenced) shall be formatted as signed, two-byte integers.

Other formats may be accommodated by providing a “dummy” Huffman table with one code structure. The number of bits in the prefix shall be set to zero. The number of bits in the entire code shall be set to the desired number of bits per sample.

5.8.6 An overview of the data part of this section is presented in Figure 8.

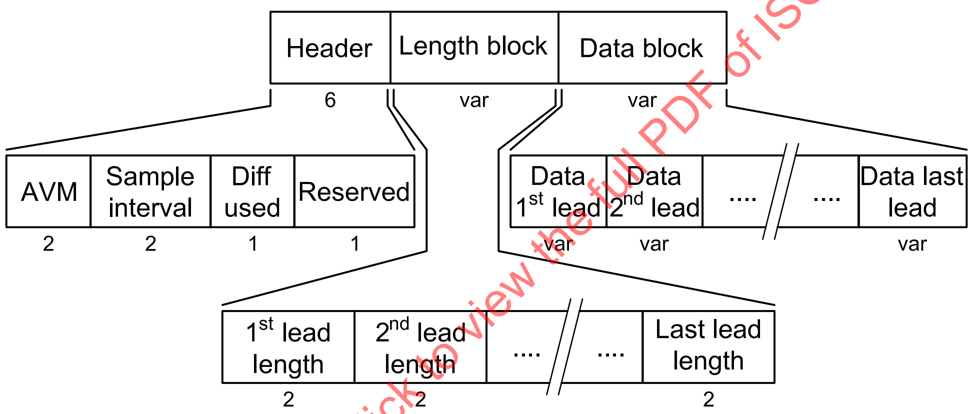


Figure 8 — Overview of the data part of the encoded type 0 reference beat section

5.9 Rhythm data – Section 6

5.9.1 This section contains either:

i) the entire ECG rhythm data, if no reference beats have been subtracted (5.6.3, byte 2),

or

ii) the residual signal after reference beats have been subtracted.

5.9.2 If present, the section shall start with a “Section ID Header” as defined in 5.2.7.

5.9.3 The section data part begins with a header that has the following format:

Byte	Contents
1 to 2	Multiplier for amplitude value (AVM). This operates as follows: The amplitude value multiplier is expressed in nanovolt (1×10^{-9} Volt).

Example: 1 250 → 1 amplitude quantum = 1,250 µV
2 441 → 1 amplitude quantum = 2,441 µV

3 to 4 The sample time interval for this section in microseconds (1×10^{-6} s).

Example: 4 000 → 250 samples/s.
1 250 → 800 samples/s.

5 This value indicates the encoding of the sample data as follows:

0 = Real (zero difference) data used for rhythm data.
1 = First difference data used for rhythm data.
2 = Second difference data used for rhythm data.

6 This value indicates how rhythm data is compressed, as follows:

0 = Bimodal compression not used.
1 = Bimodal compression used.

NOTE If bimodal compression is used, the protected region sample time interval is as in 5.8.3, but AVM is as in 5.9.3 bytes 1 to 2. Outside the protected region, AVM and sample time interval are as in 5.9.3 bytes 1 to 2 and 3 to 4.

5.9.4 The section data part contains the byte lengths of the encoded leads. Its format is as follows:

Byte	Contents
1 to 2	(Unsigned) number of bytes in compressed rhythm data for first encoded lead.
3 to 4	(Unsigned) number of bytes in compressed rhythm data for second encoded lead.
etc.	

5.9.5 The rhythm data then follows. If Section 2 has been provided, the data is coded as a series of Huffman codes taken from Section 2. The leads are encoded in the order specified in Section 3. If Section 2 is not provided, ECG data (either differenced or non-differenced) shall be formatted as signed, 2-byte integers.

Other formats may be accommodated by providing a “dummy” Huffman table with one code structure. The number of bits in the prefix shall be set to zero. The number of bits in the entire code shall be set to the desired number of bits per sample.

5.9.6 An overview of the data part of this section is presented in Figure 9.

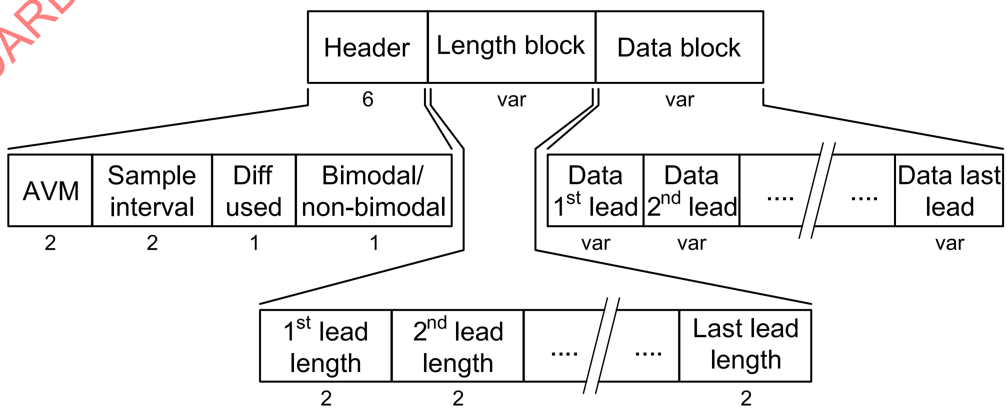


Figure 9 — Overview of the data part of the rhythm data section

5.10 Global measurements – Section 7

5.10.1 General

This section contains either global measurements for each reference beat type or for each QRS in the record, and a list of pacemaker spikes in the record. If measurements are provided for each QRS, then the first measurement block shall contain global measurements of beat type 0. The term "global" refers to measurements taken across all leads of the ECG, but not necessarily representing more than one individual beat. See 5.1.11 for a discussion of beat types.

5.10.2 Section Header ID

If present, the section shall start with a "Section ID Header" as defined 5.2.7.

5.10.3 Global ECG measurement data and pacemaker spike measurement data

5.10.3.1 General

The section data part contains global ECG measurement data and pacemaker spike measurement data if any.

Special codes, as defined in the CSE Project, have been reserved to indicate:

29 999 (decimal)	Measurement not computed by the program.
29 998 (decimal)	Measurement result not found due to rejection of the lead by the measurement program.
19 999 (decimal)	Measurement not found because wave was not present (e.g. P wave during atrial fibrillation).

These codes shall replace the measurement data when appropriate.

5.10.3.2 Global ECG measurement data

Byte	Contents
1	This byte contains either the number of reference beat types or the number of QRS's + 1 (compare to 5.10.3.5 byte 1 to 2). This byte refers to the number of measurement blocks stored, where the first block (bytes 7 to 22) always contains measurements for reference beat type 0. If this byte contains the number of reference beat types (i.e., it is not equal to 5.10.3.5 byte 1 to 2 plus 1), then each subsequent block contains the global measurements for each subsequent reference beat type. If this byte contains the number of QRS's + 1, then the subsequent blocks contain the measurements for each individual beat in sequence.
2	The number of pacemaker spikes for which location times are sent.
3 to 4	Average RR interval in milliseconds for all QRS's.
5 to 6	Average PP interval in milliseconds for all QRS's.
7 to 22	Measurements for reference beat type 0 (see 5.10.4).
23 to 38	Measurements for reference beat type 1, or for first QRS (see byte 1).
etc.	

5.10.3.3 Pacemaker spike measurement data (if any)

Byte	Contents
1 to 2	1st spike time in milliseconds from start of rhythm record (unsigned integer).
3 to 4	1st spike amplitude in microvolts (1×10^{-6} V) (signed integer).
5 to 6	2nd spike time in milliseconds from start of rhythm record (unsigned integer).
7 to 8	2nd spike amplitude in microvolts (1×10^{-6} V) (signed integer).
etc.	

Time and amplitude of these pacemaker spikes shall be given as signed quantities, which give a range of 0 to 65,535 s and ± 32,767 mV, respectively.

The time resolution for the pacemaker spikes shall be less than or equal to 2 ms.

5.10.3.4 Pacemaker spike information

For each pacemaker spike identified in 5.10.3.2, byte 2, and in 5.10.3.3, this section shall contain one 6-byte block providing additional information about the pacemaker spike. The order of the blocks corresponds to the order of the spikes identified in 5.10.3.3.

Byte **Contents**

1 Spike type of Pacemaker Spike #1:

0	Unknown
1	Spike triggers neither P-wave nor QRS
2	Spike triggers a QRS
3	Spike triggers a P-wave
4 to 127	Reserved
128 to 254	Manufacturer-specific
255	No spike type analysis performed

2 Source of Pacemaker Spike #1:

0	Unknown
1	Internal
2	External
3 to 255	Reserved

3 to 4 Index of triggered QRS complex for pacemaker spike #1:

0	No link	
1	Link to QRS #1	first QRS complex
2	Link to QRS #2	second QRS complex etc.

5 to 6 Pulse width in microseconds – 0 is unknown or uncomputed (unsigned).

5.10.3.5 QRS type information

This section identifies the reference beat type for each QRS complex in the ECG. Complexes are addressed in order. Reference beat types are numbered according to their appearance in the Global ECG measurement data section (5.10.3.2).

Byte	Contents
1 to 2	Number of QRS complexes.
3	Reference beat type of first QRS complex (0-??).
4	Reference beat type of second QRS complex (0-??).
etc.	

5.10.3.6 Additional global measurements

This section provides for additional measurements beyond those defined 5.10.3.2. It is placed here so as not to render inoperable any implementations of previous versions of the protocol.

Byte	Contents
1 to 2	Ventricular rate, in beats per minute (unsigned integer).
3 to 4	Atrial rate, in beats per minute (unsigned integer).
5 to 6	QT corrected (milliseconds) (unsigned integer).
7	Formula type used for HR correction: 0 Unknown or unspecified 1 Bazett 2 Hodges 3 to 127 Reserved 128 to 254 Manufacturer-specific 255 Measurement not available
8 to 9	Number of bytes in tagged fields, which follow (zero if no tagged fields).
10 to ***	Tagged fields, as in the following table. Valid tags are 0 to 254, tag 255 is a terminator. Each tag has at least a 1-byte tag identifier and a 1-byte length specifier (tag 255 length is 0).

Table 10 — Tagged fields

Tag	Length	Value (parameter data)
0	5	**QT end All-lead Dispersion** (Binary) QT Intervals measured in milliseconds, from QRS onset to T wave offset. All ECG leads are used in measurement. Valid values are 0 to 254 (milliseconds); 255 = measurement not provided. Byte Contents 1 Dispersion = maximum QT interval – minimum QT interval. 2 Heart rate corrected Dispersion: max.–min. 3 Dispersion = standard deviation of the QT intervals. 4 Heart rate corrected Dispersion: standard deviation. 5 Heart rate correction formula (see definition of byte 7 for valid values).
1	5	**QT peak All-lead Dispersion** (Binary) QT Intervals measured in milliseconds, from QRS onset to T wave peak. All ECG leads are used in measurement. Valid values are 0 to 254 (milliseconds); 255 = measurement not provided. Byte Contents 1 Dispersion = maximum QT peak interval – minimum QT peak interval. 2 Heart rate corrected Dispersion: max.–min. 3 Dispersion = standard deviation of the QT peak intervals. 4 Heart rate corrected Dispersion: standard deviation. 5 Heart rate correction formula (see definition of byte 7 for valid values).
2	5	**QT end Precordial Dispersion** (Binary) QT Intervals measured in milliseconds, from QRS onset to T wave offset. Precordial ECG leads only are used in measurement. Valid values are 0 to 254 (milliseconds); 255 = measurement not provided. Byte Contents 1 Dispersion = maximum QT interval – minimum QT interval. 2 Heart rate corrected Dispersion: max.–min. 3 Dispersion = standard deviation of the QT intervals. 4 Heart rate corrected Dispersion: standard deviation. 5 Heart rate correction formula (see definition of byte 7 for valid values).
3	5	**QT peak Precordial Dispersion** (Binary) QT Intervals measured in milliseconds, from QRS onset to T wave peak. Precordial ECG leads only are used in measurement. Valid values are 0 to 254 (milliseconds); 255 = measurement not provided. Byte Contents 1 Dispersion = maximum QT peak interval – minimum QT peak interval. 2 Heart rate corrected Dispersion: max.–min. 3 Dispersion = standard deviation of the QT peak intervals. 4 Heart rate corrected Dispersion: standard deviation. 5 Heart rate correction formula (see definition of byte 7 for valid values).
4 to 254	(none)	**Reserved**
255	0	**None (section terminator)**

5.10.4 The format of the measurement block for each reference beat type or for each individual QRS

Byte	Contents
1 to 2	P onset
3 to 4	P offset
5 to 6	QRS onset
7 to 8	QRS offset
9 to 10	T offset
11 to 12	P axis in the frontal plane (angular degrees, 999 if undefined)
13 to 14	QRS axis in the frontal plane (angular degrees, 999 if undefined)
15 to 16	T axis in the frontal plane (angular degrees, 999 if undefined)

NOTE 1 If the measurement block contains measurements for a reference beat type, then measurements for onset/offset are given in milliseconds from the beginning of the reference beat. If the measurement block contains measurements for an individual beat, then measurements for onset/offset are given in milliseconds from the beginning of the ECG record. Wave durations and intervals can be computed from wave or interval offset minus onset.

NOTE 2 For the axes (P, QRS, T) in the frontal plane the convention shown in Figure 10 is used.

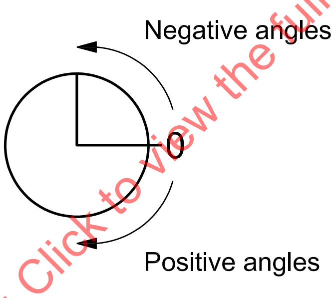


Figure 10 — Angle definition

5.10.5 Manufacturer specific global measurement block

— A block with variable length for manufacturer-specific global measurements can be added to this section, after the data on pacemaker spikes.

— The start of the manufacturer-specific block (counting from the beginning of the Section ID Header) shall be derived from the information given for the global ECG measurement data. For example, if the measurement blocks contain global measurements for each reference beat type, the start of the manufacturer specific block will be 16 (i.e. 5.2.7) + 6 + (Number of reference beat types times 16) + (Number of pacemaker spikes times 4) + (Number of pacemaker spikes times 6) + (2 + Number of QRSs) + (9 + the number of bytes in tagged global measurements) + 1. The end shall be given in the Section ID Header by the total length of the section, including the Section ID Header (see 5.2.7).

5.10.6 An overview of the data part of global measurements

An overview of the data part of Section 7, global measurements, is presented in Figure 11.

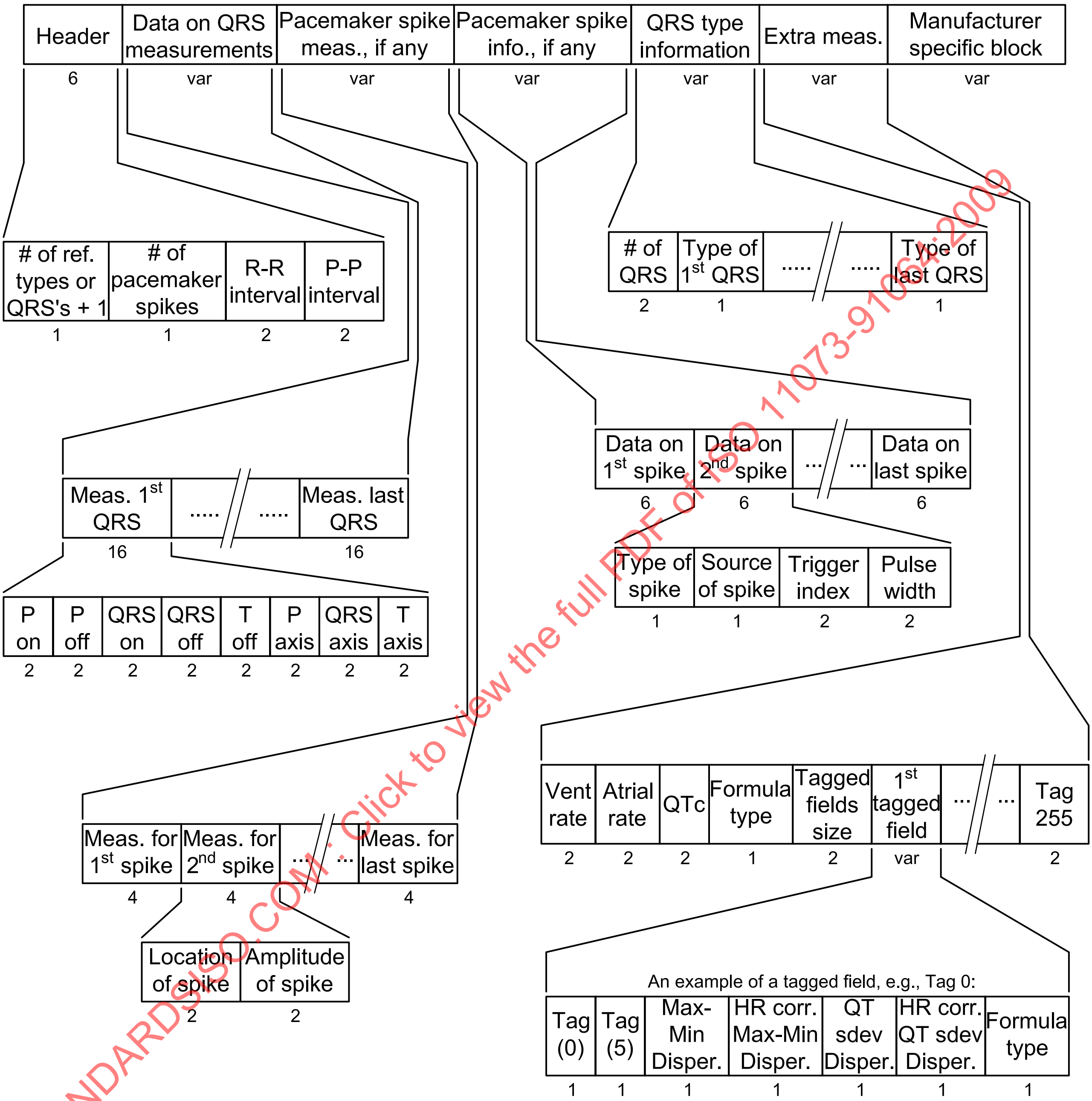


Figure 11 — Overview of the data part of the global measurements section

5.11 Storage of full text interpretive statements – Section 8

5.11.1 This section contains a text version of the latest diagnostic interpretation of the ECG.

5.11.2 If present, the section shall start with a “Section ID Header” as defined in 5.2.7.

5.11.3 The data portion of this section includes a data header followed by multiple statements.

5.11.4 Header:

Byte	Contents
1	Binary: Confirmed/Non-confirmed report:

Value	Type
0	Original report (not overread).
1	Confirmed report.
2	Overread report, but not confirmed.

Byte	Contents
2 to 3	Binary: year (full integer notation, as in 1 990).
4	Binary: month (range 01 to 12; 01 = January).
5	Binary: day (range 01 to 31).
6	Binary: hours (range 00 to 23).
7	Binary: minutes (range 00 to 59).
8	Binary: seconds (range 00 to 59).
9	Binary: number of statements in this section.

5.11.5 Statement data:

Byte	Contents
1	Binary: Statement sequence number, starting with 1.
2 to 3	Binary: Statement field length (number of bytes in the statement, starting with the first byte following, and including, the NULL terminator).
4 to ***	Statement body: text terminated by NULL.

5.11.6 No codes are allowed in the statements, unless accompanied by descriptive text.

5.11.7 The section data part lay-out is shown in Figure 12.

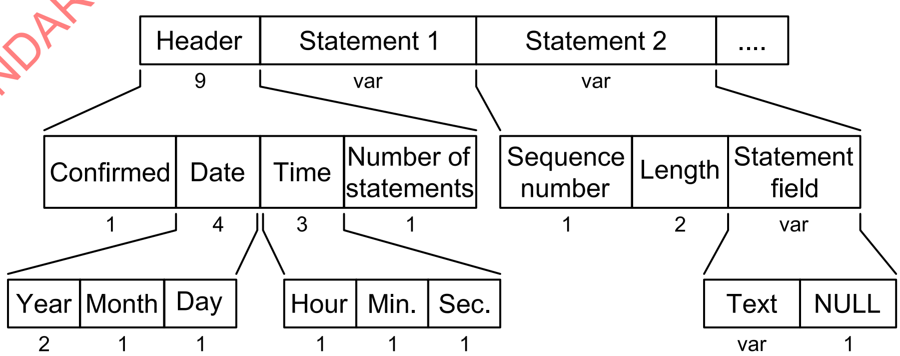


Figure 12 — Overview of the data part of Section 8

5.12 Storing manufacturer specific interpretive statements and data related to the overreading trail – Section 9

5.12.1 This section is reserved for manufacturer-specific diagnostic statements of the analysing device and overreading trail of the interpretation. The source of the analysing device and the name of the overreading physician (or device) are defined in the "Header Section" (Section 1).

5.12.2 If present, the section shall start with a "Section ID Header" as defined in 5.2.7.

5.12.3 The structure and format of the data part of this section are manufacturer specific.

5.13 Lead measurement block – Section 10

5.13.1 This section contains the measurements of each recorded lead separately. The mandatory measurements and their format are listed below. A manufacturer specific area, and escape codes for special conditions have been provided.

5.13.2 If present, the section shall start with a "Section ID Header" as defined in 5.2.7.

5.13.3 The lead measurement section data part shall consist of one record for each measured lead. Each record shall consist of four fields:

a) Lead identifier (Binary 2 bytes). Refer to 5.6.4, byte 9, for lead numbering scheme.

b) Length (unsigned integer) of record in bytes, excluding bytes 1 to 4 (Binary; 2 bytes).

c) Up to 50 basic measurements (signed integers) (Binary fields; 2 bytes each).

d) Manufacturer measurement area, starting from byte 105 on (Binary). No specific guidelines are included for the layout or format of this block.

5.13.4 Special codes, as defined in the CSE Project, have been reserved to indicate:

29 999 (decimal)	Measurement not computed by the program.
29 998 (decimal)	Measurement result not found due to rejection of the lead by measurement program.
19 999 (decimal)	Measurement not found because wave (e.g. Q wave) was not present in the corresponding lead.

5.13.5 The header of the data part of this section contains the number of leads for which a measurement block is transmitted (binary, 2 bytes), followed by 2 bytes of manufacturer-specific information.

5.13.6 Each lead measurement block shall consist of:

Byte	Contents
1 to 2	Lead ID
3 to 4	Length of record
5 to 6	P-duration (ms) (total P-duration, including P+ and P- components)
7 to 8	PR-interval (ms)
9 to 10	QRS-duration (ms)
11 to 12	QT-interval (ms)
13 to 14	Q-duration (ms)
15 to 16	R-duration (ms)
17 to 18	S-duration (ms)

19 to 20	R'-duration (ms)
21 to 22	S'-duration (ms)
23 to 24	Q-amplitude (µV)
25 to 26	R-amplitude (µV)
27 to 28	S-amplitude (µV)
29 to 30	R'-amplitude (µV)
31 to 32	S'-amplitude (µV)
33 to 34	J-point-amplitude (µV) (amplitude of the J-point = amplitude of end of QRS)
35 to 36	P(+)-amplitude (µV)
37 to 38	P(-) -amplitude (µV)
39 to 40	T(+)-amplitude (µV)
41 to 42	T(-)-amplitude (µV)
43 to 44	ST-slope (µV/s)
45 to 46	P morphology description, as defined below
47 to 48	T morphology description, as defined below
49 to 50	Iso-electric segment at onset of QRS (in milliseconds) (Segment I)
51 to 52	Iso-electric segment at the end of QRS (in milliseconds) (Segment K)

NOTE For the definition of the iso-electric segments I and K see European Heart Journal 1985, **vol.6**, pp. 815-825. Briefly, “I” is the interval between the global onset of QRS derived from all simultaneously recorded leads and the onset of QRS in a specific lead. Conversely, “K” is the time between the end of QRS in a specific lead and the global end of QRS.

53 to 54	Intrinsicoid deflection (in ms)
55 to 56	Quality code reflecting ECG recording conditions, as defined below
57 to 58	ST-amplitude at the J-point plus 20 ms
59 to 60	ST-amplitude at the J-point plus 60 ms
61 to 62	ST-amplitude at the J-point plus 80 ms
63 to 64	ST-amplitude at the J-point plus 1/16 average R-R interval
65 to 66	ST-amplitude at the J-point plus 1/8 average R-R interval
67 to 104	Reserved for future use
105 to ***	Manufacturer specific block for measurement results

All measurements shall be expressed as signed integers. The amplitudes of the Q, S, S', T(–) and P(–) waves shall be expressed as negative integers, as well as the J-point amplitude and J+20, J+60 and J+80 amplitudes, when they are negative. Note that the J-point (J) is the same as the QRS-end location.

Bytes 67 to 104 have to be set to zero and need to be transmitted if a manufacturer specific measurement block is included.

5.13.6.1 The P and T morphology description codes (Bytes 45 to 48) are defined as follows:

Value	Contents
0	unknown
1	positive
2	negative
3	positive/negative
4	negative/positive
5	positive/negative/positive
6	negative/positive/negative
7	notched M-shaped
8	notched W-shaped

5.13.6.2 The Quality Code (Bytes 55 to 56) is defined as follows:

2 binary bytes per lead, consisting of 8 2-bit fields. Each 2-bit pair represents the noise level in one of four classes.

The least significant bit of byte 55 is defined as bit 0. The most significant bit of byte 56 is defined as bit 15.

Bit	Contents	Level	Class
0 to 1	AC (mains) noise	0	none/no
2 to 3	overrange	1	moderate/yes
4 to 5	wander	2	severe
6 to 7	tremor or muscle artifact	3	unknown
8 to 9	spikes or sudden jumps		
10 to 11	electrode loose or off		
12 to 13	pacemaker		
14 to 15	interchanged lead		

5.13.7 An overview of the data part of this section is presented in Figure 13.

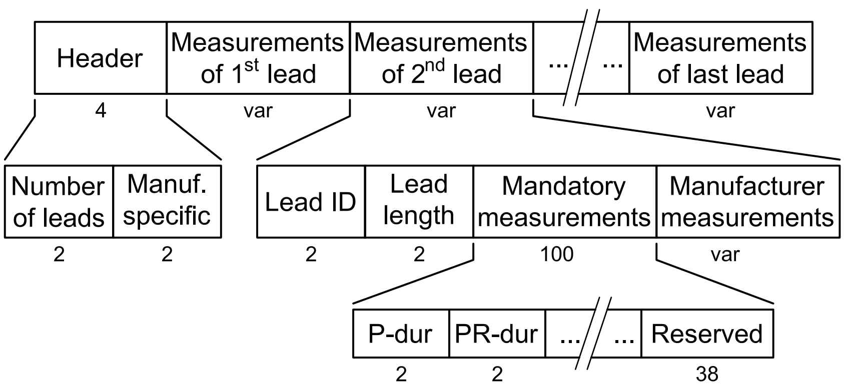


Figure 13 — Overview of the data part of the lead measurement block section

5.14 Storage of the universal ECG interpretive statement codes – Section 11

5.14.1 This section contains the most recent interpretation and overreading data, coded according to the Universal Statement Codes and Coding Rules, defined in Annex F. The data contained in this section shall be consistent with the data in Sections 8 and 9.

5.14.2 If present, the section shall start with a “Section ID Header” as defined in 5.2.7.

5.14.3 Structure and format definition:

Data shall be stored on a statement-by-statement basis. There are three possible types of statement:

1) Universal Statement Codes (as defined in Annex F).

2) Full Text (as used in Section 8).

3) Statement Logic (identifying logical relationships between statements).

To store the three types of statement, three separate statement fields have been defined.

Only one field of type “Statement Logic” is allowed to identify the logical relationships between statements of the other types. If no “Statement Logic” type field is included in the section, it is assumed that all statements are equally valid, and have no “special” relationship to each other, except for what is declared in the statement. The number of fields of the types “Universal Statement Codes” and “Full Text” are not restricted.

5.14.4 The layout of the data part of this section is presented in 5.14.5, and is explained below.

5.14.4.1 Header

Byte	Contents
1	Binary: Confirmed/Non confirmed report: Value — Type 0 — Original report (not overread). 1 — Confirmed report. 2 — Overread report, but not confirmed.
2 to 3	Binary: year (Full integer notation, as in 1 990).
4	Binary: month (range 01 to 12; 01 = January).
5	Binary: day (range 1 to 31).
6	Binary: hours (range 0 to 23) (time is always local time).
7	Binary: minutes (range 0 to 59).
8	Binary: seconds (range 0 to 59).
9	Binary: number of statements in this section.

5.14.4.2 Statement data

Byte	Contents
1	Binary: Statement sequence number. Each statement has been given a sequence number to allow easy binding by the Type 3 logical operands.
2 to 3	Binary: Statement field length (number of bytes in the statement, starting with the statement field type byte, and including the NULL terminator).
4 to ***	Statement field:

Byte — Contents

1 — Binary: Statement field type:

Value	Type
1	Coded statement type, using the Universal Statement Codes.
2	Full text type, as used in Section 8.
3	Statement logic type, as described below.

2 to *** — Data depending on the field type, terminated by NULL (0).

5.14.4.3 Statement type

Type 1: This type shall contain one coded statement optionally followed by one or more modifiers, according to the Universal Statement Codes. The coded statement and modifiers shall each occupy one data part terminated by a NULL (0). The data parts are of variable length and the number of data parts is not restricted. The only restriction is the total length of all data parts together, which is 65 535 bytes maximum.

Type 2: This type has one data part containing only text characters, and is NULL (0) terminated.

Type 3: This type has one data part containing only text characters, and is NULL (0) terminated. The contents of this field identify the logical relationships between statements by using logical operands acting on statements referred to by their sequence number.

EXAMPLE “(1+2);3” with “+” = or, “;” = and, “(...)” marking precedence this means: statement No. 1 OR statement No. 2. AND statement No. 3.

5.14.5 Layout of the data part of Section 11

See Figure 14.

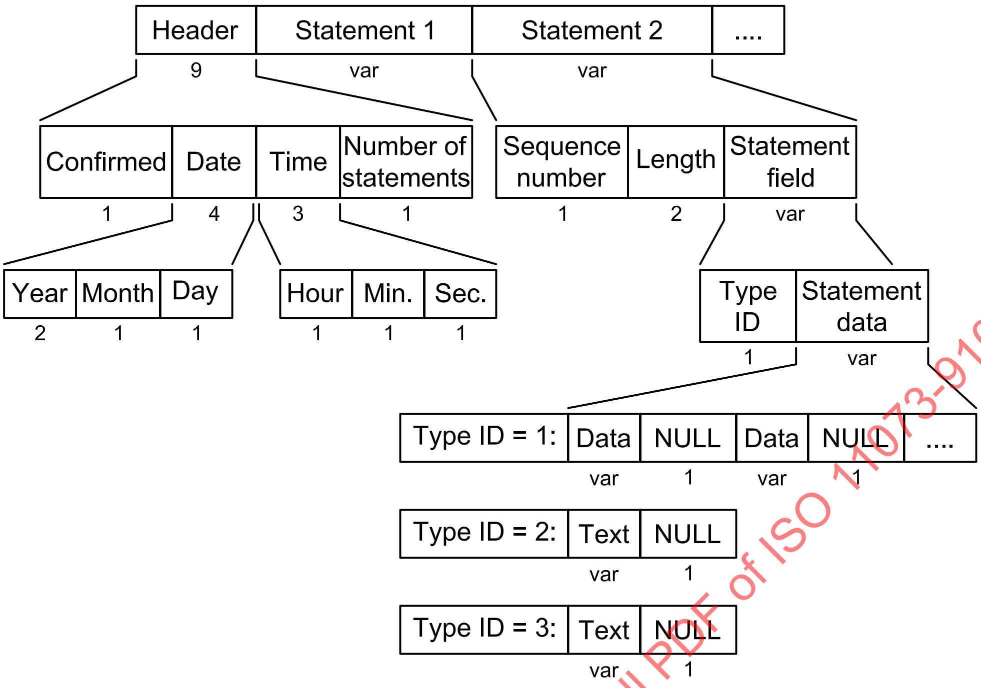


Figure 14 — Overview of the data part of Section 11

6 Minimum requirements for encoding and compression of the ECG signal data

6.1 Scope and field of application

As described in Clause 1, ECGs are taken in routine clinical practice with the patient being at rest, during defined periods of exercise or over long-term periods during regular daily activity, i.e. during ambulatory (so-called Holter) monitoring or in intensive care units for cardiac arrhythmia monitoring. Recommendations for compression of data from long-term ECG recordings have been excluded from this part of ISO 11073. The specifications for encoding and compression described are restricted to the routine resting electrocardiogram.

6.2 Introduction

ECG recordings made at rest have typically a length of 10 s. If digitized within accuracy limits recommended by the International Committees on Electrocardiology of AHA, AAMI, CSE and others, namely 500 samples/s and maximum 5 µV/LSB, for each ECG lead per second, 1 000 bytes of data is obtained with 16 bits/sample. This results in 80 000 bytes of data for a standard 12-lead 10-second ECG (in this case, redundancy of limb leads which can be reconstructed from lead I and lead II has already been removed).

Although less than medical imaging, electrocardiography thus results in large amounts of digital data compared to other medical data such as patient history, diagnostic codes and biomedical laboratory data. Although technology has nowadays significantly increased the available capacity and transmission speed, data reduction is still desirable, if not an economic necessity, especially when transmission is performed over the normal telephone network.

Indeed, there are two compelling reasons for compressing digital ECG data:

1) to reduce the (magnetic) storage requirements for medical databases and hospital information systems;

2) to reduce the time (and telephone expense) of ECG data transmission.

Although the semantic diagnostic ECG information can in theory be stored in a few bytes, re-analysis and comparison with consecutive ECG recordings is a frequent need, particularly for cardiac patients. Small changes in the morphology of an ECG may be sensitive indicators for the course of a disease. They may be recognizable within ECG serial comparison, but they may be lost if only major diagnostic findings of ECG analysis are documented as textual information.

Various ECG compression algorithms have been developed during the past decades. Compression of ECGs was, except where in specific environment only redundancy reduction was applied, always accompanied by some deterioration of the original record. Since there are no standards for accuracy limits, data formats or compression methods, compressed ECG data could be re-analysed and compared only within processing systems of the same manufacturer. Neither could data be exchanged between systems of different manufacturers nor was quantitative information available on the deterioration of the data from compression and decompression procedures.

The major goal and achievement obtained under the activities of Workpackage 2 within the SCP-ECG Project of the Preliminary AIM Programs (1989-1990) of the European Community was to define and to agree on accuracy limits and specifications for compression of ECG data, and to define encoding of data in such a way that compressed ECGs can be transmitted and re-analysed in systems of different manufacturers. The recommendations resulting from this activity form the basis of this document.

6.3 ECG compression methodology

The following principles applied to ECG data compression can be identified:

1) redundancy reduction within the digital record (reproducible data compression);

2) bandwidth limitation and reduction of sampling rate;

3) information reduction by irreversible data compression with and without defined accuracy limits.

A review of various algorithms, developed during the past 20 years, for compression of resting as well as Holter ECGs has been prepared during the SCP-ECG Project. Relevant contributions to ECG data compression can be found in the references listed in the Bibliography.

The major conclusions from this review were the following.

1) Redundancy reduction is the only method to avoid deterioration of the signal. Redundancy reduction results in compression ratios in the order of magnitude of 2 to 5.

2) Other compression schemes allow compression between 5 and 12.

3) The error measures for signal reconstruction are mainly given in RMS values. The RMS figures, however, are averages and even at small RMS figures unpredictably high absolute errors in significant signal parts may occur.

4) Validation of compression schemes should always include the analysis of absolute amplitude errors since they may be 10 to 20 times larger than the RMS figures.

5) Signal transformation (discrete Fourier transformation, Karhunen-Loeve transformation, Walsh transformation) has also been applied. The compression ratios reported range from 6 to 12. However, for these methods the reconstruction fidelity is given only in terms of RMS figures. Results reported indicate that transform compression may be used for data reduction of one typical ECG cycle rather than compression of complete ECG records.

6) Measurability of minimum waves and reproducibility of notches/slurs determine the possible compression of ECGs. At present the CSE recommendations require that minimum waveforms down to 20 µV/6 ms be measurable. Clinically significant notches of ~ 4 ms have been reported. However, as a result of recent experiments, the duration requirement for minimum waves has, for practical reasons, been adjusted to 10 ms. With a sampling rate of 500 samples/s shorter waves can indeed not be reliably measured.

7) Standardization of ECG data compression requires:

   a) harmonization of hardware specifications, amplifier noise, input voltage range, analog to digital converter precision, LSB-value, etc.;

   b) definition of reproducibility limits and maximum error tolerances.

6.4 Main results from investigations on ECG data compression in the SCP-ECG Project

6.4.1 Proposed solution for compression

Except for the redundancy reduction, any compression results in reduction of signal resolution. However, from a practical point of view, not every µV resolution and not every bit is significant. More important is that measurements, obtained at a reasonable degree of amplitude and time resolution, match with each other before and after compression.

A compression scheme was therefore developed by investigators of some companies, where a reference beat is processed from the original data and is stored or transmitted in such a way that reprocessing would provide the same measurements as before. This could be achieved:

a) by a compression which makes use of redundancy reduction only;

b) by transmission of pointers (wave recognition points) which allow re-measurement based on exactly the same references.

Besides the reference beat, a residual record is stored and transmitted. The residual record is obtained by subtracting the reference beat from the original ECG record at each cycle location. The residual record is low-pass filtered, truncated and the sampling rate is reduced. From this record first or second amplitude differences are Huffman encoded and stored or transmitted. Reconstitution of the ECG record for visual inspection is possible by adding the reference beat to the residual record at the respective cycle locations. For this method overall compression ratios in the order of twenty could be obtained.

A detailed analysis of this compression scheme in the SCP-ECG Project has revealed that within the reconstructed record relatively large errors may occur within QRS. These errors result from low-pass filtering. Particularly the QRS sections of the residual record contain high frequency components which are removed by low-pass filtering, truncation and sample rate reduction by the methods proposed and used in some commercial programs.

In the SCP-ECG Project a method was therefore designed which applies basically the same compression scheme, but whereby the QRS-section is protected from low pass filtering and sample rate reduction. In this way all the details of the QRS-T complex are maintained, on which the main morphology diagnosis shall be made. In the residual record needed to reconstruct the ECG recording on which the rhythm diagnosis has to be made, only almost invisible changes to the human eye can be seen with the compression scheme developed in SCP-ECG. These changes had no effect on the rhythm diagnosis of the studied cases.

The major advantage of the new compression scheme is that essential details of sensitive QRS data sections are protected and that the overall accuracy could be substantially improved. A detailed description of the methodology is given in Annex C, with numerical examples.

6.4.2 Testing methodology for the proposed solutions

To investigate the behaviour of this and various other compression algorithms at first a set (test set 1) of 11 ECGs from the CSE database for assessment of wave recognition accuracy has been used. These ECGs were selected according to rhythm (sinus rhythm, atrial fibrillation, atrial flutter), “normal” beats and extrasystoles, with different QRS-morphology including normal, myocardial infarction, left ventricular hypertrophy and others. These ECGs represent various types of ECG waveforms and contain low and high levels of low frequency (baseline) and high frequency (line frequency, muscle tremor) noise. Essential observations could be made on this data set. To get statistically relevant data a second set (test set 2) of 89 + 19 ECGs from the CSE diagnostic studies has been evaluated. These data were selected according to amplitude distributions, which had to be multiples of 1 μV. It also consists of cases with normal and abnormal rhythm, extrasystoles and normal and pathological QRS-morphology.

In some situations it was helpful to use artificial waveforms (e.g. simulating atrial flutter waves) and a number of artificial ECGs with normal P-QRS-T morphology. To avoid unrealistic results because of signal edges etc., care was taken that in all leads first and second derivatives are continuous. The advantage of the synthetic signals is that true amplitudes and intervals are known and that results from compression experiments are not masked by effects from noise removal or noise reproduction.

In Annex C a set of test ECGs is presented by which conformance testing and error evaluation of ECG compression algorithms can be done with respect to the requirements described in this part of ISO 11073.

6.5 Minimum requirements for ECG data compression

6.5.1 General

Based upon the results from the investigations in the SCP-ECG Project the following quantization and error limits have been chosen as standard for digital ECG encoding and compression.

6.5.2 Categories of compression schemes

Basically three categories of compression of ECG information can be identified:

Category A: Only a set of measurement parameters and diagnostic statements shall be stored or transmitted from an original ECG record.

Category B: A set of measurements, diagnostic statements and a Reference Beat with the residual record from which the original ECG can be reconstituted within error limits is stored/transmitted.

Category C: A set of measurements and diagnostic statements and the only redundancy reduced compressed original ECG is stored/transmitted. Besides, a Reference Beat (as processing result) may be compressed, stored and transmitted as well.

6.5.3 Minimum requirements for ECG data encoding and compression

6.5.3.1 It is left open to the manufacturers in which way they carry out the data compression. However, reference beat type 0 and the residual record shall be provided if reference beat subtraction is applied. The reconstruction RMS error and the absolute errors shall be verifiable on the SCP test set. Error measures shall be calculated from the beginning of the first subtraction to the end of the last subtraction. For pure redundancy reduction, the reconstruction errors (RMS and absolute errors – except the quantization error) shall be 0 with reference to a 500 samples/s and 5 μV/LSB record.

6.5.3.2 If reference beat subtraction is used for data compression, all leads of an ECG record shall be recorded simultaneously.

6.5.3.3 Digitization: SR $\geqslant$ 500 samples/s; LSB $\leqslant$ 5 μV.

6.5.3.4 Reference Beat: SR ⩾ 500 samples/s; LSB ⩽ 5 µV.

6.5.3.5 Residual Record: Truncation Error ⩽ ±15 µV.

6.5.3.6 Residual Record: Sampling Interval ⩽ 8 ms.

6.5.3.7 Reconstruction Error: RMS ⩽ 10 µV.

6.5.3.8 Absolute Error: ⩽ 100 µV in a single sample outside P-QRS-T.

6.5.3.9 Absolute Error within QRS: ⩽ 15 µV in a single sample.

Annex A
(normative)

Encoding of alphanumeric ECG data in a multilingual environment

A.1 General

The encoding method to be used for text fields in SCP-ECG is explained in this annex. The method is taken from the ISO/IEC standards that describe the use of multiple character sets. For a better understanding of the present annex, the reader should consult the structure and extension rules of the ISO 8-bit code, documented in ISO/IEC 4873 and ISO/IEC 2022.

Multiple character sets are required, not only in the case of the Japanese but also in European languages which use non-ASCII characters (e.g., “ ä ” in German, “ ç ” in French, “ å ” in Swedish etc.). Thus, use of extended character sets shall ensure that the SCP-ECG protocol can be accepted internationally.

A text string, which includes unknown character sets, may cause serious problems with patient identification in some machines. Therefore, a method for handling unsupported character sets shall also be described in this annex.

A.2 Scope

Text encoding should be applied to the fields of the SCP-ECG protocol listed in Table A.1.

Table A.1 — Fields of the SCP-ECG protocol

Section	Tag	Contents
1	0	Last name
1	1	First name
1	2	Patient identification number
1	3	Second last name
1	10	Drugs
1	13	Diagnosis or referral indication
1	14	Acquiring device ID number
1	15	Analysing device ID number
1	16	Acquiring institution description
1	17	Analysing institution description
1	18	Acquiring department description
1	19	Analysing department description
1	20	Referring physician
1	21	Overreading physician
1	22	Technician description
1	23	Room description
1	30	Free text field

Table A.1 (*continued*)

Section	Tag	Contents
1	31	ECG sequence number
1	32	History diagnostic codes
1	33	Electrode configuration code
1	34	Date time zone
1	35	Free text medical history
8+11	–	Free text diagnostic report information
9	–	Manufacturer specific diagnostic report information

This encoding is intended as an interchange format, not as an internal representation. It is expected (but not required) that each ECG computer system shall convert this format to some internal representation for processing and rendering, and convert from that internal representation to this format for data communication.

A.3 References and definitions

A.3.1 References

See Clause 2 as well as the following.

ISO/IEC 2375, *Information technology — Procedure for registration of escape sequences and coded character sets*

ISO/IEC 6429:1992, *Information technology — Control functions for coded character sets*

ISO/IEC 8859-2:1999, *Information technology — 8-bit single-byte coded graphic character sets — Part 2: Latin alphabet No. 2*

JIS X 0202, *Code Extension Techniques for Use with the Code for Information Interchange; Japanese Industrial Standard (JIS)*

SCHEIFLER, R. W., *Compound Text Encoding version 1.1*, MIT X Consortium Standard

A.3.2 Terms and definitions

For the purpose of this annex, the following definitions were taken over from ISO/IEC 4873 and JIS C 6228, that on March 1, 1987 was renamed JIS X 0202.

Latin-1 defined in ISO/IEC 8859-1 shall be used as the default character set in this part of ISO 11073.

A.3.2.1
coded character set
code
set of unambiguous rules which establishes a character set and the one-to-one relationship between each character of the set and its coded representation by one or more bit combinations

A.3.2.2
control character
control function, the coded representation of which consists of a single bit combination

A.3.2.3
graphic character
character, other than a control function, that has a visual representation normally handwritten, printed or displayed, and that has a coded representation consisting of one or more bit combinations

A.3.2.4
control function
action that affects the recording, processing, transmission or interpretation of data, and that has a coded representation consisting of one or more bit combinations

A.3.2.5
format effector
control character that controls the layout and positioning of information on character-imaging devices such as printing and display devices

A.3.2.6
to designate
to identify a set of characters that are to be represented, in some cases immediately and in others on the occurrence of a further control function, in a prescribed manner

A.3.2.7
to invoke
to cause a designated set of characters to be represented by the prescribed bit combinations whenever those bit combinations occur, until an appropriate extension function occurs

A.3.2.8
to represent
a to use a prescribed bit combination with the meaning of a character in a set of characters that has been designated and invoked or

b to use an escape sequence with the meaning of an additional control function

A.4 Values

Byte values are represented in this annex as two decimal numbers in the form column/row as defined in the ISO/IEC standards. This means that the value can be calculated as (col*16) + row; e.g. 01/11 corresponds to the value 27 (1B hex).

The byte encoding space is divided into four ranges:

C0 bytes from 00/00 to 01/15

GL bytes from 02/00 to 07/15

C1 bytes from 08/00 to 09/15

GR bytes from 10/00 to 15/15

C0 and C1 are “control character” sets, while GL and GR are “graphic character” sets. For GL, byte 02/00 is always defined as SPACE, and byte 07/15 (normally DELETE) is never used.

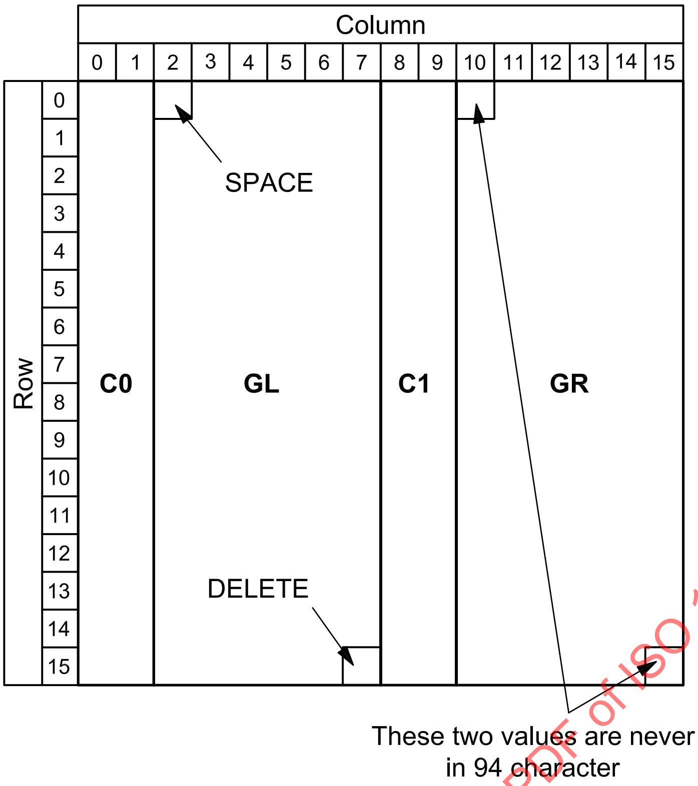


Figure A.1 — Ranges of the byte encoding space

The 16-by-16 array of columns numbered 00 to 15, and rows numbered 0 to 15 contains 256 code positions. Columns 00 to 07 of this array contain 128 character positions which are in one-to-one correspondence with the characters of the 7-bit set. Their coded representation is the same as in the 7-bit environment with the addition of an eight bit that is ZERO. Columns 08 to 15 contain 128 more positions. The eighth bit of their coded representation is ONE. Columns 08 and 09 are used to indicate control characters, and columns 10 to 15 graphic characters, with the exception of positions 10/0 and 15/15.

The 8-bit code table can be extended through various code extension facilities, one of which is by means of an appropriate escape sequence, as summarised in A.6.2.

A.5 Control characters

The definition of each control character comes from ISO/IEC 646 and ISO/IEC 4873.

As shown in Table A.2, only a subset of the control characters of C0 shall be used for the encoding of free text in SCP-ECG. No values of the control set C1 shall be used.

The interchange of text string data (such as Sections 8, 10 and 11) may require some formatting information. To this end format effectors (as listed in Table A.2) should be used, but their usage should be kept to a minimum since some machines may handle them inappropriately.

Table A.2 — Control characters

Category	Acronym	Name	Coded Presentation
Format effector	BS	BACKSPACE	00/08
Format effector	HT	HORIZONTAL TAB	00/09
Format effector	LF	LINE FEED	00/10
Format effector	VT	VERTICAL TAB	00/11
Format effector	FF	FORM FEED	00/12
Format effector	CR	CARRIAGE RETURN	00/13
Code extension	ESC	ESCAPE	01/11

1) BS

Backspace may be used for making composed characters by overstriking (ISO/IEC 646 allows this kind of representation). However, some machines have no overstriking facility, so it is better to use adequate character sets (e.g. ISO/IEC 8859-1) to represent compound characters.

2) HT, VT

Specification of tabulation width settings is not part of this proposal.

3) CR

CARRIAGE RETURN may also be used for composite characters. The same can be said for BACKSPACE.

4) LF

Some machines (such as UNIX based machines) may interpret LINEFEED (00/10) as a NEWLINE. This can be thought of as using the (deprecated) NEW LINE mode, described in E.1.3 in ISO/IEC 6429:1992.

In this proposal, NEWLINE is represented as CR+LF. So under such an environment, it is expected that this format is converted to an internal representation (i.e. convert CR+LF to LF).

NULL is used for a string terminator in the SCP-ECG protocol. But it is not a part of the text string. This is consistent with ISO/IEC 646 and ISO/IEC 4873.

A.6 Character set encoding

A.6.1 General principle of multi-lingual text encoding

In order to realize a multi-lingual environment, a switching facility for different character sets in a text string is required. Some graphic characters have different visual representation with the same code, e.g. both lower case "a" with a tilde in ISO/IEC 8859-1 and lower case "a" with a brave in ISO/IEC 8859-2 have the same code i.e. 14/03. In addition, there are some multi-byte character sets such as Japanese Kanji and Chinese Hanzi that have thousands of characters, in which case single byte encoding is impossible.

ISO/IEC 2022 defines the code extension techniques (see Figure A.2) making it possible to use multiple graphic character sets in a text string thereby achieving a multi-lingual environment. In ISO/IEC 2022, the code extension is realized by mapping the desired graphic character set into the coding space by using the following two steps:

1) the desired graphic character set is designated to one of G0, G1, G2 or G3; this is done by using escape sequences;

2) the designated set (G0, G1, G2 or G3) is invoked to the area in the coding space 02/00 to 07/15 or 10/00 to 15/15; this is done by using shift functions.

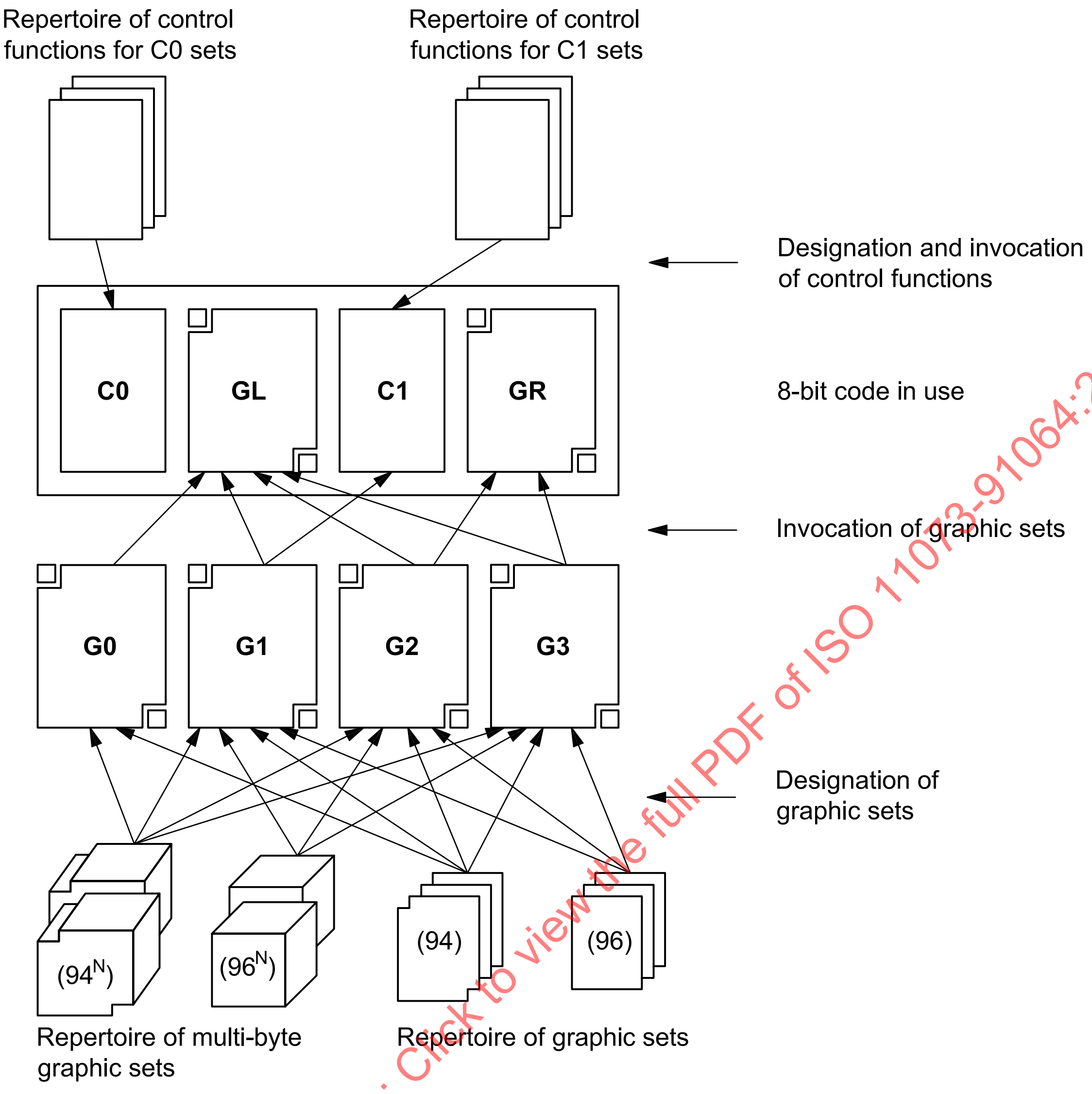


Figure A.2 — Code extension techniques (see ISO/IEC 2022)

A.6.2 Default and initial setting

A.6.2.1 We shall use only an 8-bit environment, and shall always use G0 for GL and G1 for GR (see Figure A.3). This is done by the announcement of the necessary extension facilities, as described in Clause 9 of ISO/IEC 2022:1994. Under this condition, designation also invokes, so it is not necessary to explicitly invoke.

A 96-character set can only be designated/invoked to GR. This is because 02/00 is SPACE and 07/15 is DELETE so that GL can contain only 94 characters in 02/01 to 07/14. GR can contain 96 characters in 00/00 to 15/15. Hence GR can be invoked by different 94-character sets as well as 96-character sets.

In a 94-character set no characters shall be allocated to the positions 02/00, 07/15, 10/00 and 15/15, while in a 96-character set these positions can contain characters. The three-dimensional block in the figure below represents a multi-byte character set. A 94-n character set uses n (n >1) bytes for each character as shown in Figure A.3.

A.6.2.2 The default GL set corresponds to the left half of ISO/IEC 8859-1 (Latin 1). This is the same as ASCII (ANSI X3.4-1968).

A.6.2.3 The default GR set corresponds to the right half of ISO/IEC 8859-1 (Latin 1).

These specifications determine the default character set which can be used without an escape sequence as specified in 6.4 “Initial designation and invocation” of ISO/IEC 2022. The subset of ISO/IEC 2022 for SCP-ECG is shown in Figure A.3.

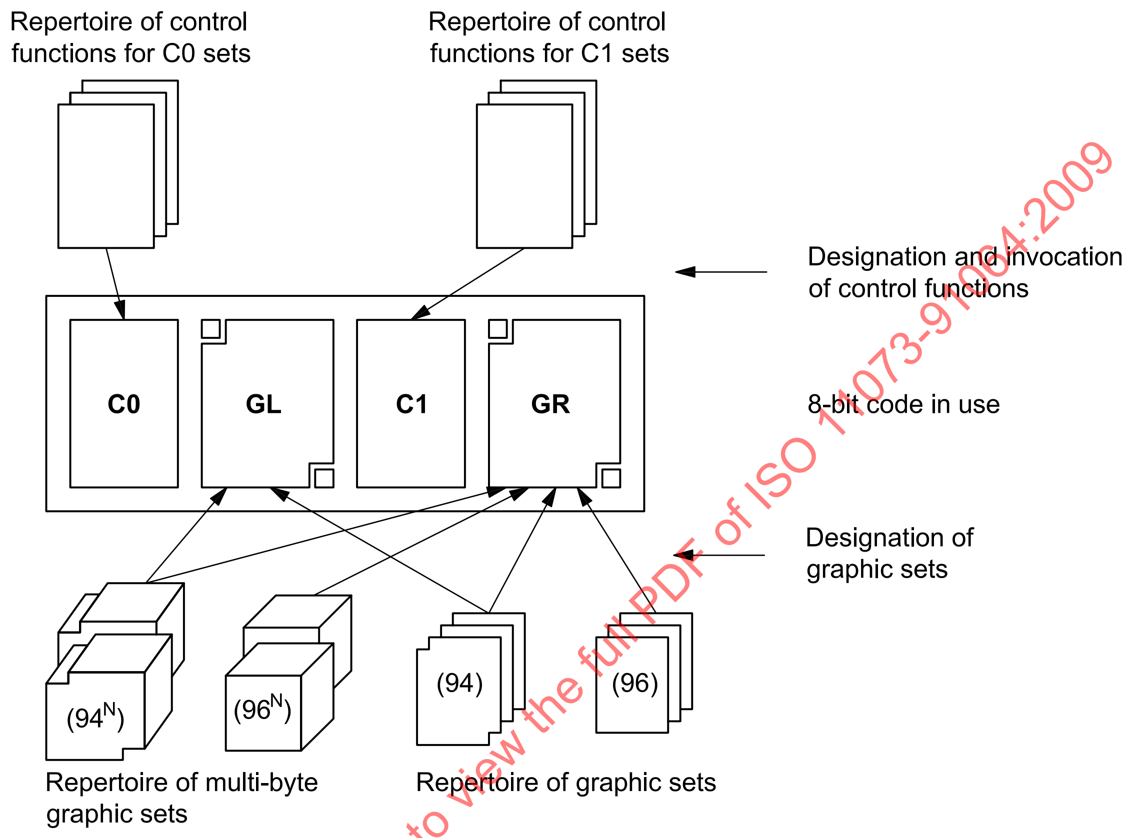


Figure A.3 — The default GR set (subset of ISO/IEC 2022)

A.6.2.4 The implied initial state in ISO/IEC 2022 is defined by the sequence shown in Table A.3.

Table A.3 — Implied initial state in ISO/IEC 2022

Escape sequence	Description
ESC 02/00 04/03	GO and G1 in 8-bit environment only. Designation also invokes.
ESC 02/00 04/07	In an 8-bit environment, C1 represented as 8-bits.
ESC 02/00 04/09	Graphic character sets can be 94 or 96.
ESC 02/00 04/11	8-bit code is used.
ESC 02/08 04/02	Designate ASCII into G0.
ESC 02/13 04/01	Designate right-handed part of ISO Latin-1 into G1.
ESC 02/01 04/00	Designate ISO/IEC 646 into C0.

The default character set for SCP-ECG is shown in Figure A.4.

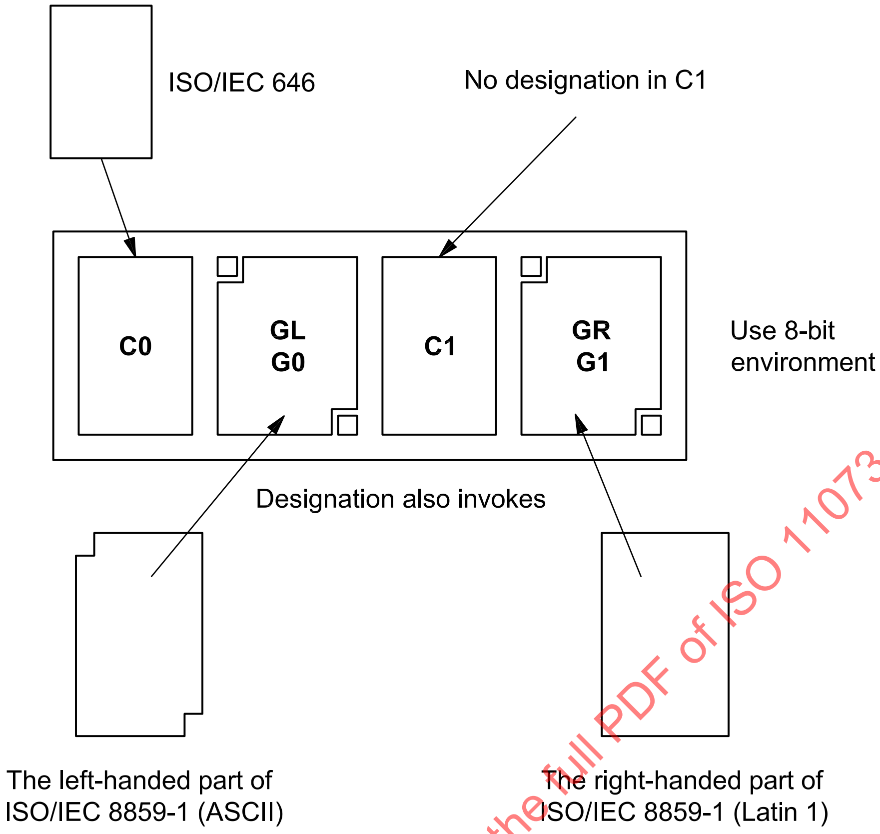


Figure A.4 — The default character set for SCP-ECG

A.6.3 Character sets extension

A.6.3.1 Figure A.5 shows the string field structure encoded in accordance with ISO/IEC 2022. The text string with such encoding contains escape sequences or shift function characters, as well as the announcement of extension facilities at the beginning of the string.

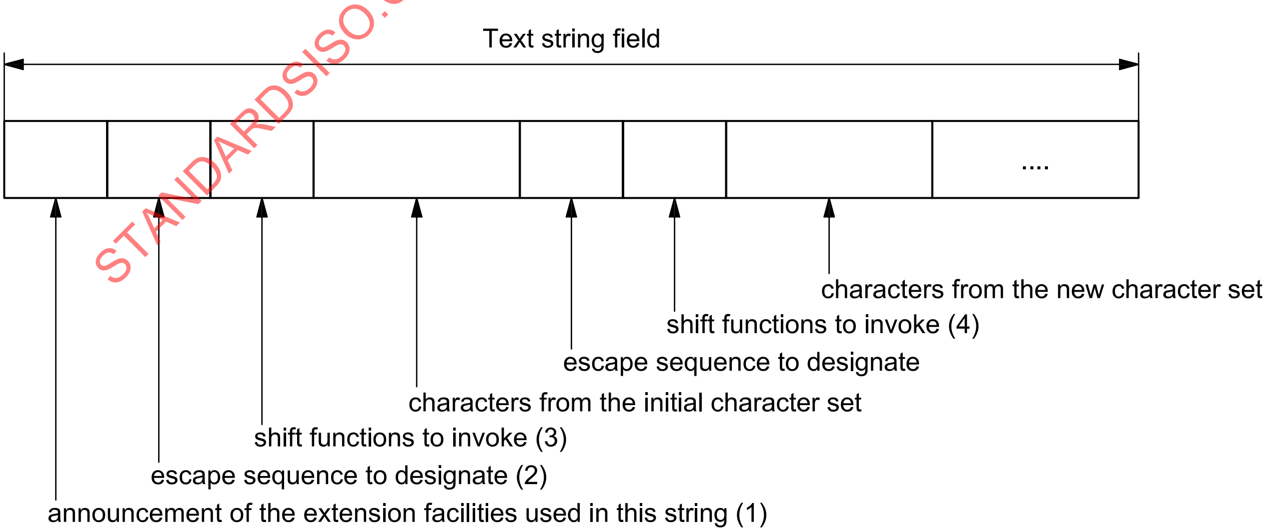


Figure A.5 — The text string field structure (ISO/IEC 2022)

A.6.3.2 The ISO/IEC 2022 method is flexible and powerful, but the full implementation may become too complicated for the SCP-ECG protocol. Some escape sequences, shift functions and announcements can be omitted in accordance with the standard.

The following rules are applied in order to omit some escape sequences and offer a simpler method for multi-lingual encoding.

1) Announcers (1) in Figure A.5 shall be omitted by agreement on extension facilities between interchanging parties.

2) Shift functions (3) and (4) shall be omitted by the escape sequences which designate and also invoke the G0 and G1 into GL and GR respectively, as defined by "ESC 02/00 04/03".

3) The escape sequence (2) is omitted by the following initial default setting:

   — Designate ASCII (left-handed part of ISO/IEC 8859-1) into G0 and also invoke to GL.

   — Designate Latin-1 (right-handed part of ISO/IEC 8859-1) into G1 and also invoke to GR.

   — Designate ISO/IEC 646 into C0.

   — No designation to C1.

No escape sequences shall be found in a text string when only ISO/IEC 8859-1 is used. This is the normal ASCII 8-bit text string. If another character set is required in a string, an escape sequence for switching the character set is necessary before the characters of the other character set.

The format of a text field with multiple character sets is summarised in Table A.4.

Table A.4 — Format of an encoded text string in SCP-ECG

Characters from the default character set	Escape sequence	Characters from the new character set	Escape sequence to change to another character set	–

Each string in the transmission shall start with the default character set. A new field or the trailing NULL shall reset to the default character set.

A.6.3.3 The format of an escape sequence for an extension character set is:

ESC {I} F

* "ESC" is the escape character. Its code representation is 01/11.

* "{I}" stands for one or more "intermediate characters", which are in the range 02/00 to 02/15. It shows the function of the escape sequence.

* "F" stands for "Final character", which is always in the range 04/00 to 07/14. These are registered by an International Registration Authority. Some of the final characters are listed in A.6.4.

A.6.3.4 To define another character set encoding to be the GL set, one of the following escape sequences is used:

ESC 02/08 F	94 character set
ESC 02/04 02/08 F	94N character set
ESC 02/04 F	some special 94N character set

For the final characters, 04/01 and 04/02 can be used in the form "ESC 02/04 F". This exception comes from 6.3.9 of ISO/IEC 2022:1994. The sequence "ESC 02/04 04/02", for the Japanese character set, is commonly used in Japan.

A.6.3.5 To define one of the other character set encodings to be the GR set, one of the following control sequences is used:

ESC 02/09 F 94 character set

ESC 02/13 F 96 character set

ESC 02/04 02/09 F 94N character set

A.6.3.6 The following intermediate characters are allowed: 02/00, 02/01, 02/04, 02/09 and 02/13. The following intermediate characters are not permitted in SCP-ECG:

02/02, 02/03, 02/05, 02/06, 02/07, 02/10, 02/11, 02/12, 02/14 and 02/15.

A.6.3.7 Final characters for private encoding (in the range of 03/00 to 03/15) are not permitted in SCP-ECG.

A.6.3.8 The other escape sequences described in ISO/IEC 2022 shall not be used for SCP-ECG.

A.6.3.9 A 94N character set uses N bytes ($N > 1$) for each character. The value of N is derived from the column value of the final character F, specified above:

column 04 2 bytes

column 05 2 bytes

column 06 3 bytes

column 07 4 or more bytes

In a 94N encoding, the byte values 02/00 and 07/15 (in GL) and 10/00 and 15/15 (in GR) are never used (the column definitions come from ISO/IEC 2022).

A.6.4 Final characters

In SCP-ECG only internationally registered final characters shall be used. The list in Table A.5 shows some of the final characters which may be used.

Table A.5 — Final characters

F	94/96	Description
04/02	94	7-bit ASCII graphics (ANSI X3.4-1968) Left half of ISO/IEC 8859 sets
04/09	94	Right half of JIS X 0201-1976 (reaffirmed 1984) 8-bit Alphanumeric-Katakana Code
04/10	94	Left half of JIS X 0201-1976 (reaffirmed 1984) 8-bit Alphanumeric-Katakana Code
04/01	96	Right half of ISO/IEC 8859-1, Latin alphabet No.1
04/02	96	Right half of ISO/IEC 8859-2, Latin alphabet No.2
04/03	96	Right half of ISO/IEC 8859-3, Latin alphabet No.3
04/04	96	Right half of ISO/IEC 8859-4, Latin alphabet No.4
04/06	96	Right half of ISO/IEC 8859-7, Latin/Greek alphabet
04/07	96	Right half of ISO/IEC 8859-6, Latin/Arabic alphabet

Table A.5 (*continued*)

F	94/96	Description
04/08	96	Right half of ISO/IEC 8859-8, Latin/Hebrew alphabet
04/12	96	Right half of ISO/IEC 8859-5, Latin/Cyrillic alphabet No.1
04/13	96	Right half of ISO/IEC 8859-9, Latin alphabet No.5
04/01	942	GB2312-1980, China (PRC) Hanzi
04/02	942	JIS X 0203-1983, Japanese Graphic Character Set
04/03	942	KS C5601-1987, Korean Graphic Character Set

The sets listed as “Left half of ...” shall always be defined as GL. The sets listed as “Right half of ...” shall always be defined as GR. Other sets can be defined either as GL or GR.

If 04/01, 04/02 and 04/03 are used in the form “ESC 02/04 F”, they are always defined as GR.

A.7 Language code

The “Language code” field of Section 2 in the SCP-ECG protocol is used to identify the class of character sets used. This field is coded according to Table A.6.

Table A.6 — Language code

Value	Description
Bit 0 reset to 0	uses ASCII only
Bit 0 set to 1	uses non-ASCII character set (includes the right half of ISO/IEC 8859-1)
Bit 1 reset to 0	uses ISO/IEC 8859-1 only
Bit 1 set to 1	uses other than ISO/IEC 8859-1
Bit 2 reset to 0	does not use multi-byte character set
Bit 2 set to 1	uses multi-byte character set
Bits 3 to 7	always reset to 0

Note that bit 0 shall be set to be 1, when using the right-handed part of ISO/IEC 8859-1. In this case, no escape sequence shall appear in the string.

A.8 Method for handling unsupported character sets

A.8.1 General

The use of unsupported character sets in the string may cause serious problems in patient identification. The method for handling unsupported character sets is described below. It is assumed that one system can input all characters being displayed/printed.

A.8.2 ISO/IEC 8859-1 based machine

Print, display and input all GR and GL characters which cannot be handled in the following manner.

* replace all “\” characters (05/02) with two characters “\\”.

* replace the escape character (01/11), which precedes unknown escape sequences, with four characters “\033”. Note that known escape sequences should be processed adequately.

A.8.3 ASCII based machine

* replace all “\” characters (05/02) with two characters “\\”.

* replace the escape character (01/11), which precedes unknown escape sequences, with four characters “\033” (note that known escape sequences should be processed adequately).

* replace all GR set characters with four characters “\nnn”, where 'nnn' is the 3 column octal representation of each byte.

Checking the language code is helpful for this method. However, parsing of the text string is still required because there might be supported and unsupported character sets in one string.

NOTE This proposal does not define the actions for over-flowing of display, print or input fields.

A.9 Summary of the Escape sequences described in this annex for the encoding of free text in SCP-ECG

At the beginning of an information interchange, it may be required to announce the code extension facilities used in the data which follow. If such an announcement is to be embedded within the character coded information, one or more of the class of three-character escape sequences ESC 02/00 F shall be used. Subject to agreement between the interchanging parties, such an announcement sequence may be omitted.

Escape sequences that are not listed here shall not be used in text fields. “Announcer” shall not appear in the field explicitly.

Table A.7 — Escape sequences

Escape sequence	Description	Note
ESC 02/00 04/03	G0 and G1 in 8-bit environment only; designation also invokes	Announcer[a]
ESC 02/00 04/07	In an 8-bit environment, C1 represented as 8-bits	Announcer[a]
ESC 02/00 04/09	Graphic character sets can be 94 or 96	Announcer[a]
ESC 02/00 04/11	8-bit code is used	Announcer[a]
ESC 02/08 04/02	Designate ASCII left half of ISO/IEC 8895-1 into G0	Default
ESC 02/13 04/01	Designate right half of ISO/IEC 8895-1 Latin-1 into G1	Default
ESC 02/01 04/00	Designate ISO/IEC 646 into C0	Default
ESC 02/08 F	Designate 94 character set into G0	
ESC 02/04 02/08 F	Designate 94N character set into G0	
ESC 02/04 F	Designate some special 94N character set into G0	
ESC 02/09 F	Designate 94 character set into G1	
ESC 02/13 F	Designate 96 character set into G1	
ESC 02/04 02/09 F	Designate 94N character set into G1	

[a] The final characters (04/03, 04/07, 04/09, 04/11) announce the code extension that is actually used, hence the term “Announcer” and also “announcing sequence”. The final character of the announcing sequence indicates the facilities for representing graphic sets and some control sets in the 7-bit or 8-bit environments.

A.10 Examples of encoded text strings

Some examples of encoded strings are shown below. The representation in an unsupported environment is also shown.

EXAMPLE 1 ISO/IEC 8859-1 string

text: ä ç å

code representation: 14/04 14/07 14/05

in unsupported environment: \344\347\345

EXAMPLE 2 mixture of ASCII and Japanese Kanji string (JIS X 0208)

text: a b c 日 本 語 1 2 3

code representation: as shown in Table A.8.

Table A.8 — Code representation

Character encoding	Comments
06/01	character “a”
06/02	character “b”
06/03	character “c”
01/11 02/04 04/02	escape sequence to change to Kanji
04/06 07/12	1st character of Kanji string shown above (日)
04/11 05/12	2nd character of Kanji string shown above (本)
03/08 06/12	3rd character of Kanji string shown above (語)
01/11 02/08 04/02	escape sequence to change to ASCII
03/01	character “1”
03/02	character “2”
03/03	character “3”

Annex B
(normative)

Definition of compliance with the SCP ECG standard

B.1 General

The SCP-ECG standard specifies means by which ECG devices and systems may exchange information. The Data Format Categories defined in this annex provide users and manufacturers of ECG devices and/or systems with a relatively simple codification of SCP-ECG related features and information content that may be provided by a specific device. The ways in which ECG data may be encoded are well defined, but are also flexible. In implementing this document, a manufacturer may choose to implement only a subset of all possible ways of encoding the ECG data. Therefore, Clause B.3 defines a testing procedure that shall be followed by manufacturers who state SCP-ECG compliance. Because of the flexibility allowed in information content and encoding, the user/purchaser shall determine the suitability of a device and/or system for a particular application.

Manufacturers who state SCP-ECG compliance for their devices and/or systems shall follow the specifications and definitions of this annex. Data format includes those items specified in Clauses 5 and 6 (and referenced annexes) of this document. Compliance for communication support is not a mandatory part of this document. In case the exchange of ECG records between two different carts and/or systems is made through a serial line connection (direct cable or modem), then a possible solution could be to use query messaging functions as specified in Annex D and data transport as specified in Annex E. If different communication supports are used (i.e. Bluetooth, wireless, LAN, etc.), the query messaging and the data transport should be clearly described in a similar way to Annex D and Annex E and made freely available to the cart/system potential user/purchaser upon request.

At this time, there is no recognised tool/method to allow testing compliance with the specifications of this document. At best such a tool would be useful to manufacturers in their efforts to assure compatibility of their devices. Because of the flexibility allowed in information content and encoding, compatibility between devices made by different manufacturers shall be determined in each case, even if both devices could be shown to be compliant with the SCP-ECG standard. A statement of compliance with the standard alone would be of little use to a user/purchaser. Therefore, a statement of compliance with the SCP-ECG standard shall be made with an accompanying statement of compatibility with a device or devices of another manufacturer or manufacturers, uniquely identified by manufacturer trade name, model description and SCP implementation software identifier.

B.2 Compliance specification

B.2.1 Data format categories

Table B.1 defines data format categories that shall be used in specifying compliance with the SCP-ECG standard. The information content provided by each category is summarized under “Content description”. Compliance implies that each data section shall be encoded as specified in the applicable subclause of Clause 5.

Table B.1 — Data format categories for compliance specifications

Category	Data sections required	Content description
I	0, 1, [2][a], 3, 6, (7)[b], (8)[b], (10)[b]	Demographics, and ECG rhythm data (uncompressed or with lossless compression)
II	0, 1, [2][a], 3, 4, 5, 6, (7)[b], (8)[b], (10)[b]	Demographics, ECG rhythm data (uncompressed, with lossless compression or with high compression), and reference beats

[a] Square brackets [.] indicate that data section 2 is required if Huffman encoding has been used.

[b] Parentheses (.) indicate that these data sections are optional for export. “Import” and “Export” are defined in B.2.2.

A further category may be added in future versions in order to fulfil the specific needs of ECG devices used in other applications (such as telemedicine or homecare).

All devices stating SCP-ECG compliance shall import at minimum data sections 0, 1, 3, 6, 7 and 8. All Categories may have additional sections added (e.g. 9, 10, 11). Manufacturer specific data shall be optionally included only in manufacturer specific fields, bytes and data blocks that have been defined in the document. Reserved, unspecified and undefined fields, bytes or data blocks shall not be used for manufacturer specific data.

B.2.2 Data exchange functions

B.2.2.1 Export

The ability to make available to a communications channel an SCP-ECG record with a specified data format category. The SCP-ECG record is created from raw ECG data as per the SCP-ECG specification, with data compression performance specified in 6.5.3.

B.2.2.2 Import

The ability to accept from a communications channel, extract and make available to the user, information from an SCP-ECG record with a specified data format category. An importing device shall minimally import data sections 0, 1, 3, 6, 7, and 8 [e.g. from a standard 10-second, 12-lead ECG (coded as I, II, V1, V2, V3, V4, V5, V6)] and make available to the user the extracted information (requirements in 6.5.3 have to be satisfied).

B.2.2.3 Transfer

Export of a previously imported record, at the same data format, with or without modification to the individual sections. ECG records stored in formats other than SCP can be converted in compliance with SCP-ECG format. Waveform data imported in a SCP-ECG format shall not be subjected to losses in compression processes for transfer.

The purpose of the transfer definition is to preserve components in the record (for example, a manufacturer-specific section) that the Importing device is unable to process. The requirement that ECG data shall not be subjected to further losses during transfer implies that either the original compressed data be sent, or that further compression be without loss. Editing by the user of demographic data, measurements and interpretations, or user chosen data reduction/losses are beyond the scope of this part of ISO 11073.

B.2.2.4 Communication channel

Any mechanism capable of making the SCP-ECG record available externally.

NOTE An example of a communication channel is RS-232 with the query messaging described in Annex D and the data transport described in Annex E. This part of ISO 11073 neither prohibits nor supports the use of any other mechanism.

B.2.3 SCP-ECG Messaging/transport protocol

In case the exchange of ECG records between different carts and/or systems is made through a serial line connection (direct cable or modem), then Annex D specifies a possible solution for the query messaging and Annex E a possible solution for data transport. A device shall specify the communication support(s) (RS-232, Bluetooth, wireless, Lan, etc.) and state that the communication protocol information for the query messaging and data transport are available upon request. Generically the manufacturer shall disclose the mechanism(s) by which SCP-ECG data formatted files may be accessed.

If a cart/system uses the RS-232 communication support, “Query Messaging supported” implies compliance with Annex D, but via a transport protocol not specified by this part of ISO 11073, and which the manufacturer shall disclose. “Query Messaging and Data Transport supported” implies compliance with Annex D and Annex E. If a device does not support all functions specified, it will be considered compliant if it responds to requests in a manner defined in this part of ISO 11073. For example, if a “Request for patient list” (D.2.2.2) is not supported, the request may be answered with “processing request not supported” [D.2.6, 11)]. A device that fails to respond to an unsupported request, or responds in a manner unspecified in this document will not be considered compliant.

It is recognized that other mechanisms for transfer of a SCP-ECG formatted file exist. This document neither prohibits nor supports the use of any mechanism other than SCP-ECG query messaging and data transport defined in Annex D and Annex E.

B.2.4 Specification for statement of compliance

A statement of SCP-ECG compliance shall have the following form and contents:

The specified device is compliant with SCP-ECG Standard Version x.x as follows:

Device: Manufacturer's trade name.

Model description.

SCP-ECG implementation software identifier.

Export: Data format category(ies), or "Not supported."

Data sections with content description.

List of manufacturers (and devices) specifying categories and optional sections with which export compatibility has been verified by testing for each device; or the following statement: "A list of manufacturers (and devices) specifying categories and optional sections with which SCP-ECG export compatibility has been verified by testing for each device is available on request".

Import: Data format category(ies), or "Not supported".

Data sections with content description.

List of manufacturers (and devices) specifying categories and optional sections with which import compatibility has been verified by testing for each device or the following statement: "A list of manufacturers (and devices) specifying categories and optional sections with which SCP-ECG import compatibility has been verified by testing for each device is available on request".

Transfer: "Supported" or "Not Supported".

Communication Channel:

Communication support: RS-232

Transmit: "Query Messaging and Data Transport not supported", or

"Query Messaging supported", or

"Query Messaging and Data Transport supported".

Receive: "Query Messaging and Data Transport not supported", or

"Query Messaging supported", or

"Query Messaging and Data Transport supported".

In case the Communication Support is different by RS-232 or transmit and/or receive are different by "Query Messaging and Data Transport supported", then the following sentence should be added: "The communication protocol information for the query messaging and data transport is available upon request" or "The mechanism(s) by which SCP-ECG data formatted files may be accessed is(are) available upon request".

In case of "Query Messaging and Data Transport Support" for transmit and/or receive then a List of manufacturers (and devices) with which SCP messaging/transport layer compatibility has been verified by testing should be added; or the following statement: "A list of manufacturers (and devices) with which SCP-ECG messaging/transport layer compatibility has been verified by testing is available on request".

B.2.5 Hypothetical examples

B.2.5.1 Cardiograph

	SCP-ECG Standard Version 2.x Statement of Compliance
Device:	MyECG Company
	Model Top1
	SCP-ECG implementation MyECG SCP version 2.
Export:	Data format category II.
	Data sections 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, containing demographics, ECG rhythm data, reference beats, global measurements, lead measurements and interpretation.
	Export compatibility at Category II has been verified by testing with:
	- Xzq Manufacturing, Inc. Models LB1577, LB1755 and ZM922 (SCP-ECG version 1.3, software implementation version 6.1);
	- BestECG, LTD. Model PQRST2 (SCP-ECG version 1.3, software implementation version 3.0).
Import:	Data format category I:
	Data sections 0, 1, 2, 3, 6, 7, 8, 10, containing demographics, ECG rhythm data, global measurements, lead measurements and interpretation.
	Data format category II:
	Data Sections 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, containing demographics, ECG rhythm data, reference beats, global measurements, lead measurements and interpretation.
	A list of manufacturers (and devices) specifying categories and optional sections with which SCP-ECG import compatibility has been verified by testing for each device is available on request.

B.2.5.2 Management System

	SCP-ECG Standard Version 2.x Statement of Compliance:
Device:	MyECG Company.
	Model Top2.
	SCP_ECG implementation version MyECG SCP version 2.
Export:	Not supported.
Import:	Data format category I:
	Data sections 0, 1, 2, 3, 6, 7, 8, 10, containing demographics, ECG rhythm data, global measurements, lead measurements and interpretation.
	Data format category II:
	Data sections 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, containing demographics, ECG rhythm data, reference beats, global measurements, lead measurements and interpretation.
	A list of manufacturers (and devices) specifying categories and optional sections with which SCP-ECG Import compatibility has been verified by testing for each device is available on request.

B.2.5.3 Defibrillator 12-lead ECG

SCP-ECG Standard Version 2.x Statement of Compliance:

Device: MyECG Company.

Model Top3.

SCP_ECG implementation version MyECG SCP version 2.

Export: Data format category I:

Data sections 0, 1, 2, 3, 6 containing demographics and ECG rhythm data.

Export compatibility at Category I has been verified by testing with:

- Xzq Manufacturing, Inc. Models LB1577, LB1755 and ZM922 (SCP-ECG version 1.3, software implementation version 6.1);
- BestECG, LTD. Model PQRST2 (SCP-ECG version 1.3, software implementation version 3.0).

B.3 Testing/validation of SCP-ECG data format compatibility

B.3.1 Overview

Testing/validation of SCP-ECG data format compliance/compatibility may be done by individual manufacturers. The requirements for testing/validation are shown in Figure B.1 and are detailed in B.3.2. In brief, each manufacturer that states export compliance makes publicly and freely available example SCP-ECG formatted records for each ECG generating device and for each data format level claimed, and additional supporting data files to allow an importing manufacturer to validate accurate decoding of the SCP-ECG records. By reading Manufacturer A's files, Manufacturer B can validate that A's files could be Imported. If in a SCP-ECG compliance statement, Manufacturer B states validation of import compatibility with A then Manufacturer A is notified in writing by Manufacturer B. Manufacturer A may then state export validation with B. Manufacturers therefore "cooperatively self-validate" compatibility with each other.

Import validation may be diagrammed as follows:

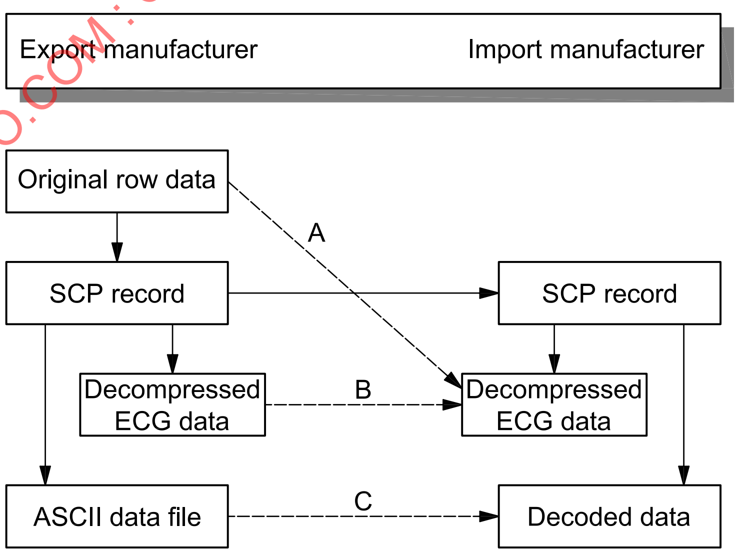


Figure B.1 — Import validation diagram

B.3.2 Requirements

An export manufacturer shall make five file types publicly and freely available for each test case provided, for each exporting device type and for each compliance level.

1) An SCP-ECG formatted binary file (***.ECG).

2) An original ECG raw data file from which the SCP-ECG file was compiled, and in the binary format defined in B.3.3 (***.EC0).

3) A decompressed ECG data file from the SCP-ECG file using the export manufacturer's decompression process, and in the binary format defined in B.3.3. (***.EC1).

4) A text file as defined in B.3.3 specifying the data in the original and the decompressed ECG binary files (***.EC2).

5) An ASCII data file (for non-waveform data) containing all demographic, measurement and interpretation data (***.EC3).

An export manufacturer shall provide test cases consisting of the required five file types for at least those cases specified in Clause C.5 (and Table C.13). The export manufacturer may also provide additional test cases in the required file formats. The set of test cases provided by the export manufacturer shall include files with data content at the limits implemented by the export manufacturer.

For all test cases provided by the export manufacturer, the import manufacturer shall import the SCP-ECG records, then decode and decompress them. For each test case, the decompressed ECG data shall be compared to the original ECG data (comparison "A" in the diagram), and the decompressed ECG data shall meet the requirements for quantization and for error limits specified in 6.5. Comparison "B" between the export manufacturer's decompressed data and the import manufacturer's decompressed data is to aid the import manufacturer in evaluating differences seen in comparison "A". Comparison "B" is optional.

For each test case, the import manufacturer shall compare the decoded demographic, measurement and interpretation data with the data in the ASCII file provided by the export manufacturer (comparison "C" in the diagram). This comparison shall be exact for all data fields decoded by the import manufacturer.

A manufacturer may state Import compatibility for each manufacturer's device and for each data format category that has been validated. If import compliance is stated for another manufacturer's exported files, then the exporting manufacturer shall be notified in writing, and the exporting manufacturer shall be allowed to state export compatibility with the importing manufacturer.

B.3.3 ECG Binary File Format (***.EC0, ***.EC1)

Each test ECG shall be provided with the following information.

1) A text file (***.EC2) containing:

   i) comma delimited descriptors for each lead of ECG data (which may be more leads or less leads than the typical 8 stored for a resting 12-lead ECG);

   ii) the total number of samples for each lead;

   iii) the sample rate (per second) or sample interval (microseconds);

   iv) the number of nanovolts per least significant bit.

2) Binary files (***.EC0, ***.EC1) with ECG data stored as 16 bit signed words, stored in Intel format (low-byte, high-byte). The sequence of the samples (S1, S2 … Sn) for leads (L1, L2 … Lm) is:

   S1L1, S1L2 … S1Lm, S2L1, S2L2 …SnL1, SnL2 … SnLm

EXAMPLE The following example is for 8 ECG leads, all identical to each other, with alternating samples of ± 1,0 mV for each lead. In this case, ± 1 000, hexadecimal values of 03E8 and FC18.

***.EC2 - Text file contains the following:

Leads: I, II, V1, V2, V3, V4, V5, V6

5 000 samples per lead; 500 samples per second; 1 000 nanovolts per LSB.

***.EC0 or ***.EC1 - Binary file (in hexadecimal for each byte)

Table B.2 — Example for 8 ECG leads

		Leads								
		I	II	V1	V2	V3	V4	V5	V6	
Bytes	00 to 0F	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	Sample 1
	10 to 1F	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	Sample 2
	20 to 2F	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	Sample 3
	30 to 3F	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	Sample 4
	40 to 4F	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	E8 03	Sample 5
	50 to 5F	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	18 FC	Sample 6

B.4 Coding of SCP-ECG compliance

5.4.5, Tag 14 (Machine ID Acquiring Device), byte 16 contains the coded compatibility categories. The upper four bits of byte 16 shall be used to characterize the data format category, and the lower 4 bits of the same byte are reserved, as depicted in Table B.3. Although a particular device may state a variety of data format categories (B.2.4), the upper 4 bits of byte 16 shall indicate the data format category for the record in which the tag is embedded.

Table B.3 — Specification of 5.4.5, Tag 14, Byte 16

Specification of 5.4.5, Tag 14, Byte 16:		**Upper 4 bits: SCP-ECG Data Format Category (I to II)**				**Reserved**			
	bit:	7	6	5	4	3	2	1	0
Upper 4 bits:	Category I:	1	1	0	1				
Data Content Compatibility	Category II:	1	1	1	0				
Lower 4 bits: Reserved						0	0	0	0

Annex C
(normative)

Methodology and conformance testing of the recommended ECG signal compression technique

C.1 General

The methodology of the recommended SCP-ECG signal data compression techniques is explained in this annex. The principles of the methodology are presented first in general in various diagrams (Clause C.2). Subsequently, a detailed description is given of the recommended data compression and decompression methodology, with corresponding mathematical definitions (see Clauses C.3 and C.4). Finally, a description is given of a test set for conformance testing (see Clause C.5).

A test set ad hoc was created to assess the reconstruction errors of ECG compression methods and to test the absolute accuracy of an ECG compression implementation. This test set is available for public use. The set provides a useful tool for validation and performance testing.

C.2 Principles of “HIGH” SCP-ECG data compression

C.2.1 Original ECG - “raw data”

a) Locate a reference point inside all ECG complexes, e.g. the time of QRSmax or any other marker:

— fiducials for reference beat subtraction (see fc1 to fc7 in Figure C.1).

b) Identify ECG complex types, i.e. normal type, different extrasystoles:

— QRS-type 0, 1, etc.

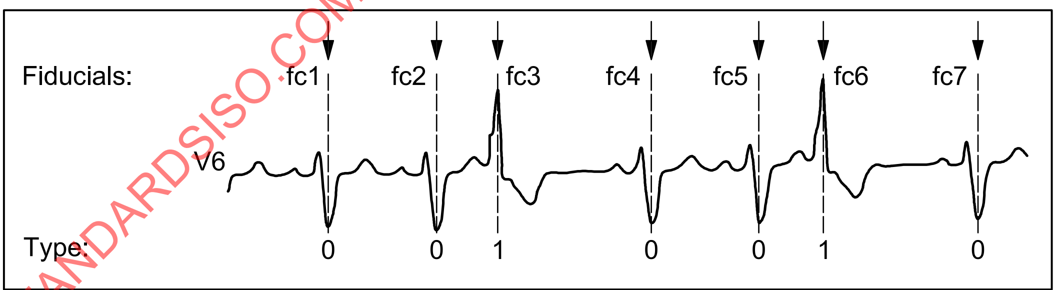


Figure C.1 — Example of raw data, fiducials and QRS typing

C.2.2 Reference beat 0

a) Compute the “Reference beat Type 0”, e.g. representative average cycle, median cycle, modal beat.

b) Identify the wave onsets and offsets of the Reference beat 0 (see Figure C.2):

— length of the reference beat 0 to be subtracted;

— pointers for QRS data segments p (see Figures C.3 and C.4) to be protected from filtering and decimation.

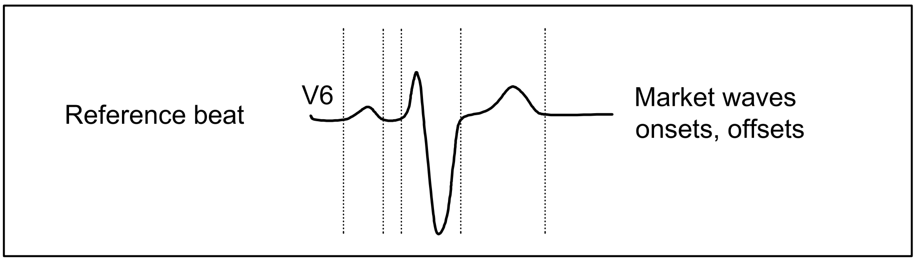


Figure C.2 — Example of a reference beat

C.2.3 Residual record

Subtract the Reference Beat Type 0 from all ECG complexes of type 0 using the fiducial locations.

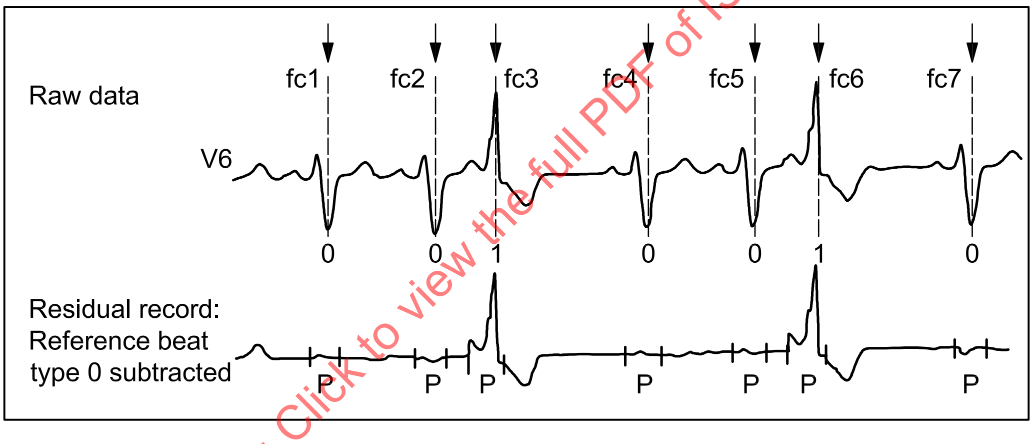


Figure C.3 — Example of a residual record

C.2.4 Compressed data

a) Low pass filtering of the residual record (excluding the protected areas p).

b) Sample decimation 2 ms to 8 ms (excluding the protected areas p).

c) Compute 2nd successive differences on all the resulting data.

Residual record:

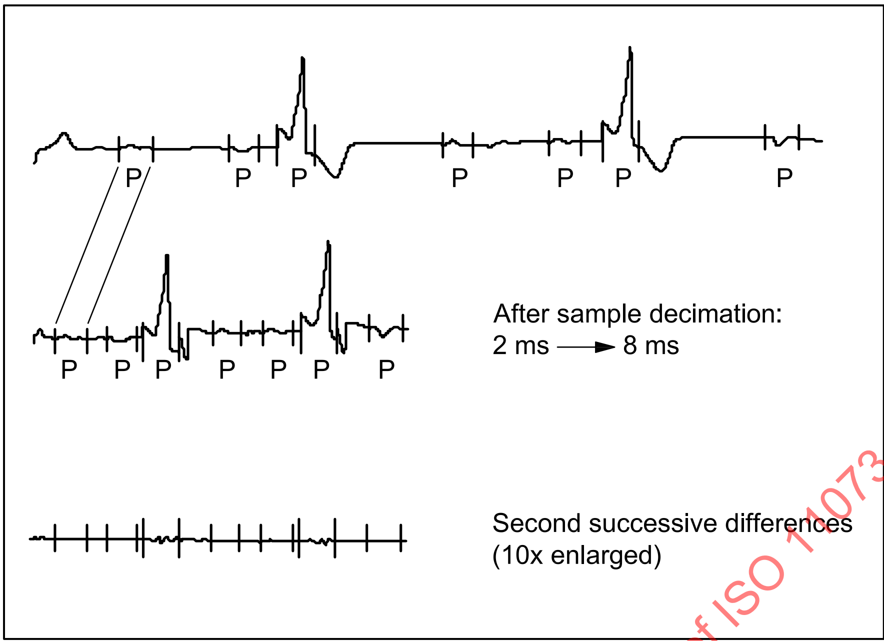


Figure C.4 — Example of residual data compression

C.2.5 Encoding

The second successive differences are then Huffman encoded, see numerical examples.

C.2.6 Decompression of SCP-ECG data

For decompression of the ECG data the steps of compression are generally performed backwards. If “high compression” is used, the steps (a) and (b) described below have to be performed on the Residual Record and the Representative Beat 0. Steps (c) and (d) apply to the Residual Record only, whereas (e) and (f) apply to both the Representative Beat 0 and the Residual Record.

The steps of decompression are:

a) decoding with Huffman tables;

b) reconstitution of the data from the first or second differences;

c) reconstitution of decimated samples;

d) low pass filtering of the reconstructed Residual Record outside protected areas;

e) multiplication with AVM (the amplitude value modifier);

f) in case of high compression: addition of the Representative Beat 0 to the Residual Record at all complex type 0 locations. For this the stored pointers fc(k), SB(k) and SE(k) of the Residual Record and fcM of the Reference Beat 0 shall be used.

Steps C.2.1 to C.2.6 are explained in detail in Clauses C.3 and C.4.

C.3 Equations for SCP-ECG data compression

C.3.1 Definitions

C.3.1.1 Raw data

Elements of raw data : $X_r(m,n)$

Subscript r : raw data

Lead : m; $1 \leqslant m \leqslant M$

Sample : n; $1 \leqslant n \leqslant N$

Amplitude quantization: LSB [µV] = AVM * 10^{-3}

(AVM, the amplitude value multiplier specified in Section 6, Header bytes 1 to 2)

Sampling interval : SI[ms] = STM * 10^{-3}

(STM, the sampling time multiplier specified in Section 6, Header bytes 3 to 4)

C.3.1.2 Sample number and time relationship

a) By definition: the first sample, n=1, is at time t=0.

b) A sampling interval (SI) of 2 ms is specified as a minimum requirement for SCP-ECG compatible data.

c) With SI = 2 ms, then the relationship between sample, n, and time, t, within the record is:

$t = (n-1)$*SI[ms]=2*$(n-1)$[ms].

C.3.1.3 Examples of denomination and indexing of ECG data

C.3.1.3.1 Raw data

Leads I, II, V1, ..., V6 $\Rightarrow M = 8$

Record length 10 s, 2 ms sampling interval

Number of samples $\Rightarrow N = 5\ 000$

Lead I: $X_r(1,1)$, $X_r(1,2)$, ... $X_r(1,n)$, ... $X_r(1,5\ 000)$

Lead II: $X_r(2,1)$, $X_r(2,2)$, ... $X_r(2,n)$, ... $X_r(2,5\ 000)$

Lead V1: $X_r(3,1)$, $X_r(3,2)$, ... $X_r(3,n)$, ... $X_r(3,5\ 000)$

Lead V6: $X_r(8,1)$, $X_r(8,2)$, ... $X_r(8,n)$, ... $X_r(8,5\ 000)$

C.3.1.3.2 Reference beat

Leads I, II, V1,..., V6 $\Rightarrow M = 8$

Record length 1 second, 2 ms sampling interval

Number of samples $\Rightarrow p = 500$

Lead I:	$Y_r(1,1)$, $Y_r(1,2)$, ... $Y_r(1,p)$, ... $Y_r(1,500)$
Lead II:	$Y_r(2,1)$, $Y_r(2,2)$, ... $Y_r(2,p)$, ... $Y_r(2,500)$
Lead V1:	$Y_r(3,1)$, $Y_r(3,2)$, ... $Y_r(3,p)$, ... $Y_r(3,500)$
Lead V6:	$Y_r(8,1)$, $Y_r(8,2)$, ... $Y_r(8,p)$, ... $Y_r(8,500)$

NOTE The data (potentials V) for leads III, aV_R, aV_L and aV_F can be computed according to the formula of Einthoven and Goldberger, where

$$\mathrm{III} = \mathrm{II} - \mathrm{I} = V_F - V_L \qquad V_{aVR} = \frac{-(\mathrm{I}+\mathrm{II})}{2} = \frac{2V_R - V_L - V_F}{2}$$

$$V_{aVL} = \mathrm{I} - \frac{\mathrm{II}}{2} = \frac{2V_L - V_R - V_F}{2} \qquad V_{aVF} = \mathrm{II} - \frac{\mathrm{I}}{2} = \frac{2V_F - V_L - V_R}{2}$$

C.3.1.4 Pointers

C.3.1.4.1 Raw data

Pointer to the fiducial point of cycle, k, in the raw data (number of QRS complexes in raw data: K): fc(k)

C.3.1.4.2 Reference beat

Pointer to the fiducial point in the reference beat data (the fiducial point may be the spatial maximum, QRS-onset or any other point of the QRS complex.): fcM

Pointer to the beginning of P in the reference beat data: PBM

Pointer to P end in the reference beat data: PEM

Pointer to QRS beginning in the reference beat data: QBM

Pointer to QRS end in the reference beat data: QEM

Pointer to T end in the reference beat data: TEM

C.3.1.4.3 Residual record

Pointer to the beginning of subtraction of reference beat 0 for cycle, k, in the raw data: SB(k)

Pointer to the end of subtraction of reference beat 0 for cycle, k, in the raw data: SE(k)

Pointer to the beginning of the protected area for complex, k, in the raw data and in the residual data: QB(k)

Pointer to the end of the protected area for complex, k, in the raw data and in the residual data: QE(k)

NOTE The pointers QB(k), QE(k) are explicitly stored in SCP-ECG section 4 (5.7.5). The protected area includes the interval from QRSonset until QRSoffset, but it may be larger (e.g. because of N-sample boundaries for decimated samples, see C.3.4.2). QRSonset(k) and QRSoffset(k) can be calculated from the fiducial fcM of Reference Beat 0, its QRS duration and the intervals bM and eM (see C.3.3.1), and from the fiducials fc(k) for each beat.

$$\mathrm{QRSonset}(k) = \mathrm{fc}(k) - \mathrm{bM}$$

$$\mathrm{QRSoffset}(k) = \mathrm{fc}(k) - \mathrm{eM}$$

C.3.2 Truncation of all values to 5 µV resolution

C.3.2.1 General

The amplitude quantization (resolution, precision) of the raw data and the reference beat data should be LSB $\leqslant$ 5 µV. The following equations describe the truncation of the data to 5 µV/LSB based upon the assumption that the original data are recorded with 1 µV/LSB resolution.

NOTE 1 Rounding and truncation are not handled by all compilers in the same way. During design of a compression program it should be made sure that rounding of positive and negative numbers is correct. For example, in Intel processors, truncation is always towards zero, i.e., –9/5 = –1.8 truncates to –1. In contrast, for calculations done in offset binary, the answer would truncate to a value equivalent to –2.

NOTE 2 Index t: truncated data.

C.3.2.2 Raw data

$$X_{\mathrm{t}}(m, n) = \frac{X_{\mathrm{r}}(m, n) + 2}{5} \quad \text{for } X_{\mathrm{r}}(m, n) \geqslant 0$$

$$X_{\mathrm{t}}(m, n) = \frac{X_{\mathrm{r}}(m, n) - 2}{5} \quad \text{for } X_{\mathrm{r}}(m, n) < 0$$

C.3.2.3 Reference beat data

$$Y_{\mathrm{t}}(m, p) = \frac{Y_{\mathrm{r}}(m, p) + 2}{5} \quad \text{for } Y_{\mathrm{r}}(m, p) \geqslant 0$$

$$Y_{\mathrm{t}}(m, n) = \frac{Y_{\mathrm{r}}(m, p) - 2}{5} \quad \text{for } Y_{\mathrm{r}}(m, p) < 0$$

C.3.3 Subtraction of the reference beat from the raw signal data

C.3.3.1 General

The SCP-ECG protocol was designed to handle ECG signal data which may be compressed and encoded by different methods:

a) pure redundancy reduction;

b) “high compression” by using a “Reference Beat 0” which is encoded separately; it may be subtracted from the raw data and the residual data may be filtered, sample decimated, etc.

The following text and equations describe in detail the computation and specification of the residual record for a compression according to b), described above as “high compression” scheme.

C.3.3.2 Truncated raw data

Given truncated raw data of lead V6 with located ECG complexes (fiducial points fck) as indicated in Figure C.5.

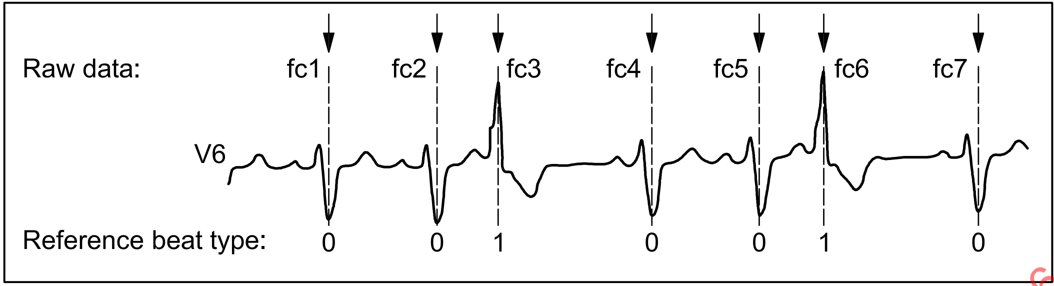


Figure C.5 — Example of raw data, fiducials and QRS typing

Table C.1 gives the locations and types of the QRS complexes located in the raw data depicted in Figure C.5 from $t = 0$ to $t = 5$ s (samples 1 to 2 500).

Table C.1 — Locations and types of QRS complexes

Sample	Pointer	QRS type	Data
1			$X_t(m,1)$
2			$X_t(m,2)$
333	fc1	0	$X_t(m,333)$
675	fc2	0	$X_t(m,675)$
840	fc3	1	$X_t(m,840)$
1 357	fc4	0	$X_t(m,1\ 357)$
1 702	fc5	0	$X_t(m,1\ 702)$
1 869	fc6	1	$X_t(m,1\ 869)$
2 373	fc7	0	$X_t(m,2\ 373)$
5 000			$X_t(m,5\ 000)$

C.3.3.3 Computation of the residual data

C.3.3.3.1 Align ("synchronize") the fiducial point fcM of reference beat 0 data with each of the fiducial points fc1, fc2, ..., fc(k) for beat type 0 complexes of the raw data. See Figure C.6.

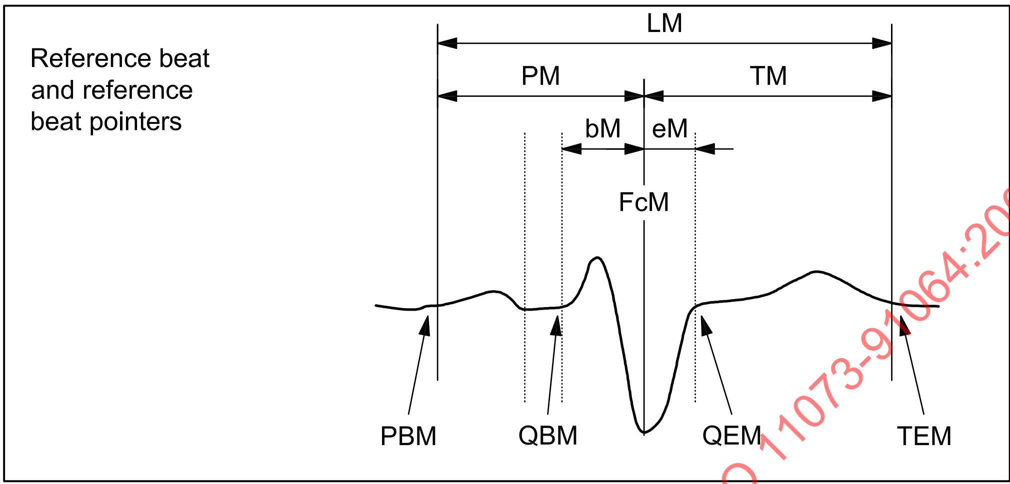


Figure C.6 — Example of a reference beat and reference beat pointers

Table C.2 presents the direct storage location of the pointers and the formula to calculate them from the data in the SCP-ECG record.

Table C.2 — Direct storage locations of pointers

Pointer	Subclause	Section	Byte	Direct	Calculation
LM	5.10.4	7		no	TEM-PBM
PM	5.7.3/5.10.4	4,7		no	fcM-PBM
TM	5.7.3/5.10.4	4,7		no	TEM-fcM
BM	5.7.3/5.10.4	4,7		no	fcM-QBM
EM	5.7.3/5.10.4	4,7		no	QEM-fcM
FcM	5.7.3	4	3 to 4	yes	
PBM	5.10.4	7	1 to 2	yes	
PEM	5.10.4	7	3 to 4	yes	
QBM	5.10.4	7	5 to 6	yes	
QEM	5.10.4	7	7 to 8	yes	
TEM	5.10.4	7	9 to 10	yes	

C.3.3.3.2 Subtract sample by sample the reference beat data from the raw data at the respective cycle location fc(k).

NOTE The length of the data segment to be subtracted is not necessarily the total length LM of the reference beat. In addition it may vary from cycle to cycle within the raw data. To accommodate this, within Section 4 of the SCP-ECG protocol under 5.7.4, bytes 3 to 6 are reserved to store pointers for the beginning of reference beat data subtraction. Bytes 11 to 14 are reserved to store the end of reference beat data subtraction for each cycle respectively.

C.3.3.3.3 For practical reasons it is most convenient to constantly subtract the "complete" reference beat data from PBM to TEM (this segment has the length LM). The pointers for the beginning and end of the subtraction of the reference beat data for cycle, k, are then found by:

$$\text{SB}(k) = \text{fc}(k) - PM$$

$$\text{SE}(k) = \text{fc}(k) + TM$$

These pointers shall be used during the subtraction procedure and they shall be stored within the QRS field data of Section 4.

NOTE Subtraction of the reference beat ("normal" Type 0) is not effective if the raw data cycle is not of the same type, e.g. an extrasystole (Type 1). However, the QRS data segments of the extrasystoles should be protected from low pass filtering and sample decimation as well.

C.3.3.3.4 The data remaining after subtraction of the reference beat at all suitable complex locations fc(k) are called "**Residual Record**".

C.3.4 Low-pass filtering

C.3.4.1 Low-pass filtering of the residual record effectively improves the compression ratio. Since high frequency components are usually found only within the QRS, all data segments except those where QRS complexes were located, can be filtered and sample decimated.

C.3.4.2 Pointers to the protected data segment of cycle k, are computed as follows:

$$\text{QB}(k) = \text{QRSonset}(k) - \text{eB}(k)$$

$$\text{QE}(k) = \text{QRSoffset}(k) + \text{eE}(k)$$

If reference beat global measurements (see 5.10.4) are present for the beat type of beat k, pointers to the QRSonset and QRSoffset are computed as follows:

$$\text{QRSonset}(k) = \text{fc}(k) - bM$$

$$\text{QRSoffset}(k) = \text{fc}(k) + eM$$

where bM denotes the interval from fiducial point fcM to QRS beginning with the reference beat and eM is the interval from fcM to QRS-end of the reference beat. The values bM and eM can be calculated from the location of the fiducial point of the reference beat (stored in bytes 3 to 4 of Section 4, see 5.7.3) and the QRS onset and offset pointers (stored in bytes 5 to 6 and 7 to 8 of Section 7, see 5.10.4).

The values eB(k) and eE(k) in the equations are adjustments to the protected data segment to avoid problems with sample decimation/reconstruction. A practical solution is to set boundaries for QB and QE so that the interval for sample decimation between beat k–1, and beat k, [= QB(k)–QE(k–1) – 1], is a multiple of the sample interval of the residual record.

Thus, QB(k) can be reduced of a quantity eB(k) (from 0 to downsampling factor – 1) to take into account the need of maintaining the previous non-protected interval as a multiple of the downsampling factor. Moreover the last QE(k) can be increased by a quantity eE(k) (from 0 to downsampling factor – 1) to take into account the need of maintaining the last non-protected interval as a multiple of the downsampling factor. The fiducials eB(k) or eE(k) may additionally be decreased or increased by a multiple of the downsampling factor due to a "safety margin" for possible errors in locating QRS onset and QRS offset.

C.3.4.3 A simple non-recursive moving average filter has given sufficient results for low pass filtering of the residual record. The filter length, L, is 9 samples.

Filter values for an odd filter length, L, are calculated as follows:

$$F(m,a) = X(m,a)$$

$$F(m,a+1) = \frac{X(m,a) + X(m,a+1) + X(m,a+2) + 1}{3}$$

$$F(m,a+2) = \frac{X(m,a) + X(m,a+1) + X(m,a+2) + X(m,a+3) + X(m,a+4) + 2}{5}$$

$$F(m,a+3) = \cdots$$

$$F(m,n) = \frac{X\{m,n-[(L-1)/2]\} + \ldots + X(m,n) + \ldots + X\{m,n+[(L-1)/2]\} + (L-1)/2}{L}$$

$$F(m,b-3) = \cdots$$

$$F(m,b-2) = \frac{X(m,b-4) + X(m,b-3) + X(m,b-2) + X(m,b-1) + X(m,b) + 2}{5}$$

$$F(m,b-1) = \frac{X(m,b-2) + X(m,b-1) + X(m,b) + 1}{3}$$

$$F(m,b) = X(m,b)$$

NOTE The methods for rounding are defined in C.3.2.2 and C.3.2.3. The rounding constants 1, 2, …, $(L-1)/2$ in the following filter equations should be negated in case the filtered values are negative.

After subtraction of reference beat type 0, there can be steps in the residual data at the boundaries of the subtraction zones, SB(k) and SE(k). These steps shall not be filtered, so that it is possible to reproduce them in the reconstituted residual data.

There are three filter intervals, a and b, for each complex k, and an additional one from the end of complex K, to the end of the data stream.

1) From the end of the reference beat type 0 subtraction of the preceding complex (k–1) to the beginning of subtraction of the current complex k.

   a = SE(k–1)+1 SE(0) = 0

   b = SB(k)–1

2) From the beginning of the reference beat type 0 subtraction of the current complex k, to the QRS onset of the current complex k.

   a = SB(k)

   b = QB(k)–1

3) From the QRS offset of the current complex k, to the end of the reference beat type 0 subtraction of the current complex k.

   a = QE(k)+1

   b = SE(k)

4) The filter interval for the remaining samples until the end of the data is:

   a = SE(K)+1

   $b = N$

where

m is the lead number;

n is the sample number;

K is the number of QRS complexes;

N is the number of the last sample;

k is the QRS complex number.

Pointer to begin subtraction of reference beat type 0 for cycle (k) in raw data: SB(k)

Pointer to end subtraction of reference beat type 0 data for cycle (k) in raw data: SE(k)

Pointer to beginning of protected area for complex (k) in raw data and in residual data: QB(k)

Pointer to end of protected area for complex (k) in raw data and in residual data: QE(k)

C.3.5 Sample decimation

Besides low pass filtering, sample decimation can be applied to "slowly changing" data. The SCP-ECG protocol specifies a minimum sampling rate of 125 samples/s for the residual record, which is equivalent to an 8 ms sampling interval.

Decimation of sampling is permitted only outside the protected QRS data segments. In SCP, sample decimation is called "bimodal compression".

For sample decimation an averaging algorithm is applied. This is different from sample decimation with subsequent four sample averaging.

For the implementation of SCP-ECG, manufacturers may choose the method with which they want to perform sample decimation, by averaging or sample skipping. It should be noted that the best results are obtained if compression and decompression algorithms are adjusted to each other.

Although no particular method of decimation and reconstruction is required, transients can be created at the subtraction zone boundaries by the reconstruction process unless certain conditions are met. Therefore SCP-ECG recommends the following boundaries for reference beat type 0 subtraction:

— Place SB(k) at the beginning of a decimated sample interval (e.g. sample 1,5,9,13,17,21,...)

— SE(k) is at the end of a decimated sample interval (e.g. sample 4,8,12,16,20,…)

If these conditions are met, it is not necessary to know which decimating algorithm is used to avoid transients at the subtraction zone boundaries during reconstruction.

A sampling interval of 8 ms is the maximum which is allowed for the residual record in SCP-ECG. So 125 samples/s is the lowest effective sampling rate within the residual record. Bimodal compression is indicated by 5.9.3 byte 6.

To avoid problems with sample decimation/reconstruction a practical solution is to set boundaries for QB and QE so that the interval for sample decimation between beat k–1 and beat k (= QB(k)–QE(k–1)–1) is a multiple of the sample interval of the residual record.

It is not necessary to update the pointers for beginning and end of the protected area after sample decimation, because they are pointers within the residual record and not within the sample decimated residual record (see specifications for Section 4 in 5.7).

There is no universal algorithm for the reconstruction of the decimated samples. It is left to the manufacturers as to which way they perform the reconstruction (interpolation or repetition of samples). However, the reconstruction RMS error and the absolute errors between the original and decompressed ECG shall be verifiable on the SCP-ECG test set. This test set is described in C.5.

EXAMPLE Filtering and decimation of the residual record resulting from ECG data is depicted in C.3.3.

Table C.3 — Example of filtering and decimation

Sample number	Pointer	Protected segment	Data filtered	Data decimated	Comment
1		no	yes	yes	
2		no	yes	yes	
3		no	yes	yes	
178	SB1	no	yes	yes	P1 onset
264		no	yes	yes	
265	QB1	yes	no	no	Begin protected area 1st QRS
267	QRSonset1	yes	no	no	QRS begin 1st QRS
333	fc1	yes	no	no	Fiducial 1st QRS
361	QE1 = QRSoffset1	yes	no	no	End protected area and QRS end 1st QRS
362		no	yes	yes	
495	SE1	no	yes	yes	T1 end
520	SB2	no	yes	yes	P2 onset
605		no	yes	yes	
606	QB2	yes	no	no	Begin protected area 2nd QRS
609	QRSonset2	yes	no	no	QRS begin 2nd QRS
675	fc2	yes	no	no	Fiducial 2nd QRS

NOTE QB2 is not equal to QRSonset2 because in this case the difference for sample decimation (= QB2−QE1−1) would be 609−361−1= 247 samples, which is not a multiple of the 4 sample decimation. The sample 606 for QB2 is suitable, because 606−361−1= 244 samples, which is a multiple of the 4 sample decimation.

C.3.6 Computation and storage of the difference data

C.3.6.1 General

To take advantage of inter-sample correlation, first or second successive differences of the respective data can be stored and encoded instead of the “original” values of the reference beat and of the residual record data. Usually the word length (in bits) of an original sample is much larger than that of the first or second successive differences.

The SCP-ECG protocol leaves it to the user, whether first or second differences (or even original data) are stored/transmitted. This shall be specified in byte 5 of the header of Section 5 (for reference beat data) and of Section 6 (for residual data), respectively.

C.3.6.2 Computation of first and second successive differences

It is assumed that of the residual record, resulting from the raw data recording of N samples, Q samples remain in the decimated record.

We denote these samples for lead, m, as follows:

$$Z(m,1),\ Z(m,2),\ \ldots\ Z(m,q),\ \ldots\ Z(m,Q)$$

The differences are then calculated in the following way:

First differences:

$$\Delta 1(m,q) = Z(m,q+1) - Z(m,q); q = 1\ \ldots\ (Q-1)$$

NOTE $\Delta 1(m,1) \equiv Z(m,1)$, where $\equiv$ means, “is defined as”.

Second differences:

$$\Delta 2(m,q) = Z(m,q+2) - 2 * Z(m,q+1) + Z(m,q);\ q = 1\ \ldots\ (Q-2)$$

NOTE $\Delta 2(m,1) \equiv Z(m,1), \Delta 2(m,2) \equiv Z(m,2)$

Equivalently the first and/or second differences for the reference beat can be calculated from the reference beat data $Y_t(m,p)$.

C.3.6.3 Reconstitution of the data from the differences

Reconstitution of the data from the first differences requires storage/transmission of one (the first) original value because:

$$Z(m,q+1) = Z(m,q) + \Delta 1(m,q);\ q = 1\ \ldots\ (Q-1)$$

where $Z(m,1)$ is the necessary value to restore $Z(m,2)$,

$$Z(m,2) = Z(m,1) + \Delta 1(m,1)$$

NOTE $Z(m,1) = \Delta 1(m,1)$

Reconstitution of the data from the second differences requires storage and transmission of two original values:

$$Z(m,q+2) = 2 * Z(m,q+1) - Z(m,q) + \Delta 2(m,q);\ q = 1\ \ldots\ (Q-2)$$

where $Z(m,1)$ and $Z(m,2)$ are the necessary values to restore $Z(m,3)$,

$$Z(m,3) = 2 * Z(m,2) - Z(m,1) + \Delta 2(m,1)$$

NOTE $Z(m,1) = \Delta 2(m,1), Z(m,2) = \Delta 2(m,2)$

The respective original values should be stored at the beginning of the encoded data (see 5.8.3).

C.3.7 Huffman encoding within SCP-ECG

C.3.7.1 General

Huffman encoding provides a minimum redundancy encoding for data with non-uniform probability of occurrence. An encoding scheme is applied where the most frequently occurring values are assigned the shortest bit code length. The most seldomly occurring data shall be given the longest bit code length. Huffman encoding changes word-oriented data into a bit-oriented data stream.

With the Huffman tables provided in the SCP-ECG protocol, pure Huffman encoding and initial encoding are possible. It is possible to use more than one Huffman table and to switch between different tables during the encoding for better compression efficiency if the distribution of the data word lengths changes.

C.3.7.2 Pure Huffman encoding

In pure Huffman encoding one data word is represented in the Huffman table with the "entire code", which is stored as the "base code" or the "prefix". For encoding, the prefix shall be used and for the decoding, the base value from the table shall be used. The length of the entire code in bits after encoding is equal to the length of the prefix.

EXAMPLE In the Huffman table # 1 (Table C.4) the code for "0" is 0_b in the prefix. The entire code length is 1 bit which is equal to the length of the prefix. For the value "1" the prefix is 100_b and the bit code length is 3, i.e. the length of the prefix.

C.3.7.3 Initial encoding

If the data value is not contained in the Huffman table the value shall be initially encoded. The base value contains, in this case, no reasonable and useful value (should be "0"). But the table contains information about the number of bits of a "remainder" in the bit stream which follows the prefix. From this remainder the reconstructed data word shall be calculated. The number of these bits is equal to the difference between the number of bits of the entire case and the prefix.

EXAMPLE Bit stream: 1110000000101_b

1) The comparison of the bit stream to the Huffman table # 1 (Table C.4) provides structure 6 for decoding. Therefore the first 4 bits are used and this is a switch to Huffman table # 2.

2) For decoding, the method of initial encoding shall be used, because in table # 2 the number of bits of the prefix 1 (prefix is 0_b) is less than that of the entire code (9 bit – 1 bit = 8 bit > 0).

3) The length of the remainder is equal to the difference between the number of bits of the entire code and the prefix. In this case these are 8 bits (00000101_b), used for reconstruction of the data word.

4) To complete the data word (2 bytes), the missing bits shall be filled with the most significant bit of the remainder.

EXAMPLE Reconstructed data word: $0000000000000101_b = 5_h$

C.3.7.4 Huffman tables used in SCP-ECG

C.3.7.4.1 General

Tables C.4 and C.5 are used in the examples for encoding of the ECG data.

C.3.7.4.2 Structure of the Huffman tables

Huffman table # 1 contains 7 structures, 5 structures for pure Huffman encoding, 1 for 8-bit initial encoding and 1 for changing to table # 2. Data values $\leqslant$ 2 are pure Huffman encoded. Data values > 2 are initial encoded. In this case the first 4 bits of 12 bits of entire code are the prefix (1111_b). The following 8 bits are the remainder.

If the third byte (table mode switch) of a structure (in this case No. 6) is reset to 0 this indicates that the actual Huffman table should change to the Huffman table with the number entered in the base value. In this case the prefix for changing is 1110_b and the number of the new table is 2.

Table C.4 — Huffman table # 1

No. of code structures	Number of bits		Table mode	Base value	Prefix code (in bits)
	Entire code	Prefix			
1	1	1	1	0	0
2	3	3	1	1	100
3	3	3	1	–1	101
4	4	4	1	2	1100
5	4	4	1	–2	1101
6	4	4	0	2 (switch to table # 2)	1110
7	12	4	1	No entry (0), 8-bit original	1111

Huffman table # 2 contains 5 structures, 3 structures for pure Huffman encoding, 1 for 8-bit initial encoding and 1 for changing to table # 1. Data values $\leqslant$ 1 are pure Huffman encoded. Data values > 1 are initial encoded. The first bit of the nine-bit code is the prefix.

If the third field (table mode switch) of the structure (in this case No. 5) is reset to 0 this indicates that the actual Huffman table should change to the Huffman table with the number entered in the base value. In this case the prefix for changing is 111_b and the number of the new table is 1.

Table C.5 — Huffman table # 2

No. of code structures	Number of bits		Table mode	Base value	Prefix code (in bits)
	Entire code	Prefix			
1	9	1	1	no entry (0), 8-bit original	0
2	3	3	1	0	100
3	3	3	1	1	101
4	3	3	1	–1	110
5	3	3	0	1 (switch to table # 1)	111

C.3.7.4.3 Huffman tables without table mode switch

Let us consider an example of the following succession of byte-oriented data values:

1, 2, –1, 0, 3, 0, 4, 1, 0, –2,

0, 15, –1, 0, 13, 0,1, –2, –1, 1

These data shall be encoded with the Huffman table # 1. For each data value the corresponding bit code from the Huffman table is picked up and connected to the bit stream. If a value is not found in the table it shall be encoded with the initial encoding as described above.

Table C.6 — Alignment of byte oriented data values and Huffman code bits

No.	Value	Entire code	Comments
1	1	100	3 bit prefix, no remainder
2	2	1100	4 bit prefix, no remainder
3	–1	101	3 bit prefix, no remainder
4	0	0	1 bit prefix, no remainder
5	3	111100000011	4 bit prefix, 8 bit remainder
6	0	0	1 bit prefix, no remainder
7	4	111100000100	4 bit prefix, 8 bit remainder
8	1	100	3 bit prefix, no remainder
9	0	0	1 bit prefix, no remainder
10	–2	1101	4 bit prefix, no remainder
11	0	0	1 bit prefix, no remainder
12	15	111100001111	4 bit prefix, 8 bit remainder
13	–1	101	3 bit prefix, no remainder
14	0	0	1 bit prefix, no remainder
15	13	111100001101	4 bit prefix, 8 bit remainder
16	0	0	1 bit prefix, no remainder
17	1	100	3 bit prefix, no remainder
18	–2	1101	4 bit prefix, no remainder
19	–1	101	3 bit prefix, no remainder
20	1	100	3 bit prefix, no remainder

Resulting bit stream:

10011001010111100000011011110000010010001101

0111100001111101011110000110101001101101100

C.3.7.4.4 Huffman tables with table mode switch

To get better compression efficiency it is possible to encode data with more than one Huffman table. For example it is reasonable to encode the reference beat and the residual record with different tables, or to switch to another Huffman table within the encoding exercise because the probability of occurrences has changed with time.

For example, should the following succession of byte oriented data values be picked up (including the switch statements):

1, 2, –1, 0, 3, 0, 4, 1, 0, –2, switch to table 2

0, 15, –1, 0, 13, switch to table 1, 0, 1, –2, –1, 1

The encoding begins with Huffman table # 1. After the 10th value, the encoding changes to Huffman table # 2, and after the encoding of the 16th value, it changes back to table # 1.

For alignment of byte oriented data values and Huffman bit codes see Table C.7.

Table C.7 — Alignment of byte oriented data values and Huffman code bits

No.	Value	Entire code	Table	TMS	Comments
1	1	100	1	1	3 bit prf., no rem.
2	2	1100	1	1	4 bit prf., no rem.
3	–1	101	1	1	3 bit prf., no rem.
4	0	0	1	1	1 bit prf., no rem.
5	3	111100000011	1	1	4 bit prf., 8 bit rem.
6	0	0	1	1	1 bit prf., no rem.
7	4	111100000100	1	1	4 bit prf., 8 bit rem.
8	1	100	1	1	3 bit prf., no rem.
9	0	0	1	1	1 bit prf., no rem.
10	–2	1101	1	1	4 bit prf., no rem.
11	(2)	1110	1	0	4 bit prf. → table # 2
12	0	100	2	1	3 bit prf., no rem.
13	15	000001111	2	1	1 bit prf., 8 bit rem.
14	–1	110	2	1	3 bit prf., no rem.
15	0	100	2	1	3 bit prf., no rem.
16	13	000001101	2	1	1 bit prf., 8 bit rem.
17	(1)	111	2	0	3 bit prf. → table # 1
18	0	0	1	1	1 bit prf., no rem.
19	1	100	1	1	3 bit prf., no rem.
20	–2	1101	1	1	4 bit prf., no rem.
21	–1	101	1	1	3 bit prf., no rem.
22	1	100	1	1	3 bit prf., no rem.

TMS: table mode switch;

prf.: prefix;

rem.: remainder;

"→ table #": switch to table #;

(#): # shall not be transferred from the base value to the reconstituted data field;

#: the number of the new Huffman table.

Resulting bit stream:

1001100101011110000001101111000001001000110111110

10000000111111010000000110111101001101101100

C.3.7.5 Definition and storage of the Huffman tables in Section 2

To explain how Huffman tables (for base code/prefix see Note 1) have to be stored in Section 2, Tables C.6 and C.7 are presented like the layout overview in the SCP-ECG protocol in the main document.

Table C.8 — Example of storage of two Huffman tables in Section 2

Section header:	**CRC**	**ID**	**length**	**reserved**
	2 a)	*2*	*4*	*8*

Number of tables:	2
	2
Number of structures:	7 (number for the first table)
	2

Structure 1	1	1	1	0	0
Structure 2	3	3	1	1	1
Structure 3	3	3	1	–1	5
Structure 4	4	4	1	2	3
Structure 5	4	4	1	–2	11
Structure 6	4	4	0	2	7
Structure 7	4	12	1	0	15
a) *for each structure*	*1*	*1*	*1*	*2*	*4*

Number of structures:	5 (number for the second table)
	2

Structure 1	1	9	1	0	0
Structure 2	3	3	1	0	1
Structure 3	3	3	1	1	5
Structure 4	3	3	1	–1	3
Structure 5	3	3	0	1	7
a) *for each structure*	*1*	*1*	*1*	*2*	*4*

NOTE 1 Base code/prefix 1st bit in code represented by the least significant bit of the 4 byte area. It means that the code is stored in the 4-byte field in its bit-reversed format. Thus, store code 100_b (decimal 4) as 1_b (decimal 1), code 1100_b (decimal 12) as 0011_b (decimal 3) and so on. Compare for this the last two columns in the default Huffman table (Table C.9 – see C.3.7.6).

NOTE 2 The numbers in italics indicate the length in bytes of the corresponding fields.

C.3.7.6 Definition of the default SCP-ECG Huffman table

Theoretically, different ECG data sets may require different Huffman code tables. However, through extended experiments with different data sets it was found that the following Huffman table could be used for practically all types of ECG data without much loss in data compression efficiency.

If, on the other hand, a manufacturer wants to apply different tables to different ECGs (ECG data sets) he can do so by specifying these tables in Section 2 of the SCP-ECG protocol.

Table C.9 is the default SCP-ECG Huffman table referred to in the main document under 5.5.4, bytes 1 to 2. The table has been used for the encoding of the data depicted in the Examples 1 and 2 in C.4.1 and C.4.2, respectively.

Table C.9 — Default SCP-ECG Huffman table

No.	Number of bits		Table mode	Base value	Prefix code (in bits)	Store binary as
	Entire code	Prefix				
1	1	1	1	0	0	0d
2	3	3	1	1	100	1d
3	3	3	1	–1	101	5d
4	4	4	1	2	1100	3d
5	4	4	1	–2	1101	11d
6	5	5	1	3	11100	7d
7	5	5	1	–3	11101	23d
8	6	6	1	4	111100	15d
9	6	6	1	–4	111101	47d
10	7	7	1	5	1111100	31d
11	7	7	1	–5	1111101	95d
12	8	8	1	6	11111100	63d
13	8	8	1	–6	11111101	191d
14	9	9	1	7	111111100	127d
15	9	9	1	–7	111111101	383d
16	10	10	1	8	1111111100	255d
17	10	10	1	–8	1111111101	767d
18	18	10	1	8 bit orig.	1111111110	511d
19	26	10	1	16 bit orig.	1111111111	1023d

In order to identify a switch to another Huffman table a TMS shall be used (see 5.6.4, byte 7 in the main document). This switch shall be inserted in the Huffman encoded data stream. It identifies the change of the Huffman table. The new Huffman table that shall be used to decode the data is identified in the structure (see 5.6.4, bytes 8-9 in the main document).

C.3.8 Decoding of compressed ECG data

C.3.8.1 General

To make the explanation of the compression and decompression logic consistent, the same notation has been used for variables and indices. The variables and indices used in the decompression algorithms get a prime (′).

EXAMPLES compression: $X(m,n)$, $Z(m,q)$, $\Delta 1(m,q)$, etc.

decompression: $X'(m,n)$, $Z'(m,q)$, $\Delta 1'(m,q)$, etc.

C.3.8.2 Decoding with Huffman tables

The method of decoding Huffman encoded data is as follows.

The first bit from the bit stream of every encoded lead is picked up and compared with the first (or default) Huffman table. If an equal prefix is found, the corresponding value shall be entered in the field for the decoded data. This value is in case of pure Huffman coding the corresponding base value or in case of initial encoding from the remainder reconstructed data value. The length of the remainder is equal to the difference in length of the entire code and prefix. If there is no equal prefix code found, the first two bits are picked up and compared, if necessary, with the first three and so on. If the code is found and the data value entered in the field, the comparison continues with the next bit.

If a prefix of a structure for changing to another Huffman table is found, the actual table shall be quit. This is indicated by the table mode switch of this structure. It is reset to 0. The number of the new table is read from the base value of this structure. Inside the new table the decoding can continue as described.

C.3.8.3 Reconstitution of the first and second differences

C.3.8.3.1 General

In the SCP-ECG protocol it is left to the user whether first or second differences are stored/transmitted (or even original data). The kind of data is specified in the headers of Section 5 (for the reference beat) and Section 6 (for the residual record) in byte 5. It is assumed that the residual record, resulting from Huffman decoding, contains a total of Q values.

C.3.8.3.2 First differences

We denote the Huffman decoded samples for lead m:

$$\Delta 1'(m,1),\ \Delta 1'(m,2),\ldots\Delta 1'(m,q),\ldots\Delta 1'(m,Q)$$

The “original” data $Z'(m,n)$ can be calculated in the following way:

$$Z'(m,q) = Z'(m,q-1) + \Delta 1'(m,q);\ q = 2 \text{ to } Q$$

Reconstitution of the “original” data from first differences requires storage/transmission of one (the first) original data value:

$$Z'(m,1) = \Delta 1'(m,1)$$

$$Z'(m,2) = Z'(m,1) + \Delta 1'(m,2)$$

where $\Delta 1'(m,1)$ is the necessary “original” value.

C.3.8.3.3 Second differences

We denote the Huffman decoded samples for lead, m:

$$\Delta 2'(m,1),\ \Delta 2'(m,2),\ldots\Delta 2'(m,q),\ldots\Delta 2'(m,Q)$$

The “original” data $Z'(m,n)$ can be calculated in the following way:

$$Z'(m,q) = 2{*}Z'(m,q-1) - Z'(m,q-2) + \Delta 2'(m,q);\ q = 3 \text{ to } Q$$

Reconstitution of the "original" data from second differences requires storage and transmission of two original values:

$$Z'(m,1) = \Delta 2'(m,1)$$

$$Z'(m,2) = \Delta 2'(m,2)$$

$$Z'(m,3) = 2 * Z'(m,2) - Z'(m,1) + \Delta 2'(m,3)$$

where $\Delta 2'(m,1)$ and $\Delta 2'(m,2)$ are the necessary "original" values.

Equivalently the reference beat data $Y'(m,p)$ can be calculated from the first and/or second differences of the reference beat.

The respective original values should be stored at the beginning of the encoded data (see 5.8.3).

C.3.8.4 Reconstitution of decimated samples

For "slowly changing" data, a sample decimation (up to a maximum sample interval of 8 ms, which is equivalent to 125 samples/s) can be applied. The use of sample decimation is called bimodal compression, and is indicated by 5.9.3, byte 6.

There is no exact algorithm for reconstitution of the decimated samples. The best results of sample decimation are obtained if compression and decompression algorithms are adjusted to each other.

An averaging algorithm with sufficient results for compression is:

first average value:

$$Z_{\mathrm{av}}(m,1) = \frac{X(m,1) + X(m,2) + X(m,3) + X(m,4)}{4}$$

second average value:

$$Z_{\mathrm{av}}(m,2) = \frac{X(m,5) + X(m,6) + X(m,7) + X(m,8)}{4}$$

…

and for decompression:

$$X'(m,i) = \frac{Z'_{\mathrm{av}}(m,2) - Z'_{\mathrm{av}}(m,1)}{4} * (i-1) + Z'_{\mathrm{av}}(m,1)$$

where $1 \leqslant i \leqslant 4$.

$Z'_{\mathrm{av}}(m,1)$ is the left-sided average value [equal to $X'(m,1)$] and $Z'_{\mathrm{av}}(m,2)$ is the right-sided average value of the four recalculated values of the interval. This reconstruction moves over the complete non-protected area, a,b, ($b-a$ is the number of the non-decimated samples). The first two reconstructed samples of this area are set equal to the first average value and the last two samples equal to the last average value.

Interval, a,b:

$$X'(m,a) \quad = Z'_{\mathrm{av}}(m,a),$$

$$X'(m,a+1) \quad = Z'_{\mathrm{av}}(m,a),$$

Interpolation over $(b–a–1)/4$ intervals.

$$X'(m, b–1) = Z'_{av}(m,(b–a)/4),$$

$$X'(m,b) = Z'_{av}(m,(b–a)/4)$$

EXAMPLE Table C.10 gives an example of decimation and reconstitution of 100 samples. For easier presentation, the steps with difference data and Huffman encoding are omitted. The values of Z_{av} are calculated as described above.

Table C.10 — Example of decimation and reconstitution of 100 samples

No.	Original	Decimated	Reconstructed
1	$X'(m,1)$	$Z'_{av}(m,1)$	$Z'_{av}(m,1)$
2	$X'(m,2)$		$Z'_{av}(m,1)$
3	$X'(m,3)$		$Z'_{av}(m,1)$
4	$X'(m,4)$		$Z'_{av}(m,1)+1*[Z'_{av}(m,2)–Z'_{av}(m,1)]/4$
5	$X'(m,5)$	$Z'_{av}(m,2)$	$Z'_{av}(m,1)+2*[Z'_{av}(m,2)–Z'_{av}(m,1)]/4$
6	$X'(m,6)$		$Z'_{av}(m,1)+3*[Z'_{av}(m,2)–Z'_{av}(m,1)[/4$
7	$X'(m,7)$		$Z'_{av}(m,2)$
8	$X'(m,8)$		$Z'_{av}(m,2)+1*[Z'_{av}(m,3)–Z'_{av}(m,2)]/4$
9	$X'(m,9)$	$Z'_{av}(m,3)$	$Z'_{av}(m,2)+2*[Z'_{av}(m,3)–Z'_{av}(m,2)]/4$
10	$X'(m,10)$		$Z'_{av}(m,2)+3*[Z'_{av}(m,3)–Z'_{av}(m,2)]/4$
11	$X'(m,11)$		$Z'_{av}(m,3)$
12	$X'(m,12)$		$Z'_{av}(m,3)+1*[Z'_{av}(m,4)–Z'_{av}(m,3)]/4$
......			
93	$X'(m,93)$	$Z'_{av}(m,24)$	$Z'_{av}(m,23)+2*[Z'_{av}(m,24)–Z'_{av}(m,24)]/4$
94	$X'(m,94)$		$Z'_{av}(m,23)+3*[Z'_{av}(m,24)–Z'_{av}(m,24)]/4$
95	$X'(m,95)$		$Z'_{av}(m,24)$
96	$X'(m,96)$		$Z'_{av}(m,24)+1*[Z'_{av}(m,25)–Z'_{av}(m,24)]/4$
97	$X'(m,97)$	$Z'_{av}(m,25)$	$Z'_{av}(m,24)+2*[Z'_{av}(m,25)–Z'_{av}(m,24)]/4$
98	$X'(m,98)$		$Z'_{av}(m,24)+3*[Z'_{av}(m,25)–Z'_{av}(m,24)]/4$
99	$X'(m,99)$		$Z'_{av}(m,25)$
100	$X'(m,100)$		$Z'_{av}(m,25)$

C.3.8.5 Low pass filtering of reconstructed residual record

Low pass filtering of the residual record outside the protected areas after reconstruction of the decimated samples smoothes noise which has risen during reconstitution. A simple non-recursive moving average filter has given sufficient results for low pass filtering of the residual record.

The filter length is three samples. The filter values are calculated as follows:

$$F'(m,a) - X'(m,a)$$

$$F'(m,a+1) = \frac{X'(m,a) + X'(m,a+1) + X'(m,a+2) + 1}{3}$$

$$F'(m,a+2) = \frac{X'(m,a+1) + X'(m,a+2) + X'(m,a+3) + 1}{3}$$

$$\vdots$$

$$F'(m,n) = \frac{X'(m,n-1) + X'(m,n) + X'(m,n+1) + 1}{3}$$

$$\vdots$$

$$F(m,b-1) = \frac{X'(m,b-2) + X'(m,b-1) + X'(m,b) + 1}{3}$$

$$F'(m,b) = X'(m,b)$$

for $k = 0$ to K

$a = \text{QE}(k)+1,$ $\qquad$ $\text{QE}(0) = 0$

$b = \text{QB}(k+1)-1,$ $\qquad$ $\text{QB}(K+1) = N+1$

$a+1 \leqslant n \leqslant b-1$

NOTE The methods for rounding are defined in C.3.2.2 and C.3.2.3. The rounding constant “1” in the previous equation should be negated in case the filtered values are negative.

C.3.8.6 Multiplication of raw data with AVM

C.3.8.6.1 General

This step means calibration. For this purpose the data are multiplied by the AVM to get the raw data, and in case of high compression, the residual record and the reference beat in equal resolution for addition.

C.3.8.6.2 Raw data

$$X'_r(m,n) = X'(m,n) * \text{AVM}, \quad \text{for} \quad 1 \leqslant n \leqslant N$$

$$1 \leqslant m \leqslant M$$

C.3.8.6.3 Reference beat data

$$Y'_r(m,p) = Y'(m,p) * \text{AVM}, \quad \text{for} \quad 1 \leqslant p \leqslant P$$

$$1 \leqslant m \leqslant M$$

C.3.8.7 Addition of the reference beat to the residual record

The SCP-ECG protocol was designed to handle data that may be compressed and encoded by different methods:

a) pure redundancy reduction;

b) “high SCP-ECG compression”.

In case of “high” compression it is necessary to add the reference beat again to the residual record at all locations were it was subtracted during compression. Therefore the reference beat shall be synchronized with the residual record with the fiducial pointers stored in the SCP-ECG record as fc(k) and fc(M). Pointers to the QRS protected zones are calculated from the stored pointers. The area for addition of reference beat to cycle, k, is known through the pointers SB(k) (addition beginning) and SE(k) (addition end).

NOTE 1 Within Section 4 of the SCP-ECG protocol under 5.7.4, bytes 3 to 6 are reserved to store pointers for the beginning of reference beat data addition [SB(k)]. Bytes 7 to 10 are reserved to store the fiducial pointer of cycle, k [fc(k)], and bytes 11 to 14 are reserved to store the end of reference beat data addition [SE(k)] for each cycle. Addition on a sample by sample basis of the reference beat data to the residual record at the respective cycle location fc(k) will get the original sample data back.

NOTE 2 If the QRS type is non-zero (bytes 1 to 2, under 5.7.4) then reference beat 0 has not been subtracted from this cycle [see footnote 3) on page 38].

C.3.8.8 Default SCP-ECG decompression parameters

— reference beat subtraction used (Section 3, byte 2, bit 0 = 1);

— AVM for reference beat data = 5 µV (Section 5, bytes 1+2 = 5 000);

— sample time interval for reference beat data = 2 ms (Section 5, bytes 3+4 = 2 000);

— second difference data used for reference beat data (Section 5, byte 5 = 2);

— AVM for residual record = 20 µV (Section 6, bytes 1+2 = 20 000);

— sample time interval for residual record outside protected areas (QRS) = 8 ms (Section 6, bytes 3+4 = 8 000);

— second difference data used for residual record (Section 6, byte 5 = 2);

— sample interpolation to 2 ms sampling interval (for algorithm see C.3.8.9);

— low-pass filtering after sample interpolation with filter length of three samples (for algorithm see C.3.8.10);

— Huffman table for decoding of reference beat data and residual record (see Table C.9).

C.3.8.9 Default method for interpolation of decimated samples

In compressed data only the protected areas have the original sampling interval. The areas between the protected areas and the parts before the first and after the last protected area have to be expanded by an interpolation algorithm in order to get the original sampling interval.

An algorithm that gives sufficient results for sample interpolation in the interval, a,b, is as follows:

$Z'(m,a')$ = decimated samples

$X'(m,a)$ = interpolated samples

m = lead number

$X'(m,a) = Z'(m,a')$

$X'(m,a+1) = Z'(m,a')$

$X'(m,a+2) = Z'(m,a') \quad \Delta = [Z'(m,a'+1)-Z'(m,a')]/4$

$X'(m,a+3) = Z'(m,a') + 1*\Delta$

$X'(m,a+4) = Z'(m,a') + 2*\Delta$

$X'(m,a+5) = Z'(m,a') + 3*\Delta$

$X'(m,a+6) = Z'(m,a'+1) \quad \Delta = [Z'(m,a'+2)-Z'(m,a'+1)]/4$

. .

$X'(m,a+9) = Z'(m,a'+1) + 3*\Delta$

. .

. .

. .

$X'(m,b-1) = Z'(m,b')$

$X'(m,b) = Z'(m,b')$

These computations are performed for all $K+1$ data segments outside the protected areas:

for $k = 0$ to K

$a = \mathrm{QE}(k)+1 \qquad \mathrm{QE}(0) = 0$

$b = \mathrm{QB}(k+1)-1 \qquad \mathrm{QB}(K+1) = N+1$

C.3.8.10 Default method for three sample point moving average

Low-pass filtering of the reconstituted residual record is done for smoothing of the truncation steps. Low-pass filtering can only be done outside the protected areas, because inside the QRS the reconstruction error is strictly limited to 15 µV. Special care shall be taken of the boundaries of the subtraction areas. After the subtraction of the reference beat data there can be steps at these boundaries and those steps shall be present again after the addition of the reference beat data. Therefore they cannot be eliminated by filtering. A simple non-recursive moving averaging filter has given sufficient results for low pass filtering of the reconstituted residual record.

The filter length, L, shall be three samples. The filter values for an odd filter length, L, are calculated as follows:

$$F'(m,a) = X'(m,a)$$

$$F'(m,a+1) = \frac{X'(m,a) + X'(m,a+1) + X'(m,a+2) + 1}{3}$$

$$\vdots$$

$$F'(m,n) = \frac{X'(m,n-(L-1)/2) + \cdots + X'(m,n) + \cdots + X'(m,n+(L-1)/2) + (L-1)/2}{L}$$

$$\vdots$$

$$F'(m,b-1) = \frac{X'(m,b-2) + X'(m,b-1) + X'(m,b) + 1}{3}$$

$$F'(m,b) = X'(m,b)$$

There are three filter intervals, a,b, for each complex k, and an additional one from the end of complex K, to the end of the data stream:

1) From the end of the reference beat subtraction of the preceding complex (k–1) to the beginning of subtraction of the current complex k.

   a = SE(k–1)+1   SE(0)= 0

   b = SB(k)–1

2) From the beginning of the reference beat subtraction of the current complex k, to the QRS onset of the current complex k.

   a = SB(k)

   b = QB(k)–1

3) From the QRS offset of the current complex k, to the end of the reference beat subtraction of the current complex k.

   a = QE(k)+1

   b = SE(k)

4) The filter interval for the remaining samples until the end of the data is:

   a = SE(K)+1

   b = N

where

m is the lead number;

n is the sample number;

K is the number of QRS complexes;

N is the number of the last sample;

k is the QRS complex number;

Pointer to begin subtraction of Reference Beat for cycle k in raw data: SB(k)

Pointer to end subtraction of Reference Beat data for cycle k, in raw data: SE(k)

Pointer to the beginning of the protected area for complex k, in the raw data and in the residual data: QB(k)

Pointer to end of the protected area for complex k, in the raw data and in the residual data: QE(k)

NOTE The methods for rounding are defined in C.3.2.2 and C.3.2.3. The rounding constants 1, ... (L-1)/2 in the previous equation should be negated in case the filtered values are negative.

C.4 Numerical examples for SCP-ECG data compression

C.4.1 Example 1

This example shows the different data obtained during SCP-ECG high compression for the first 28 samples of an ECG record.

- RAW: raw data with 1 µV/LSB, sampling interval 2 ms (500 samples/s)
- TRU: truncated raw data 1 µV/LSB → 5 µV/LSB
- RES: residual record, after subtraction of the reference beat
- FIL: non-recursive moving average filter over nine samples
- DEC: sampling rate decimation to 8 ms with averaging (125 samples/s)
- 2D: second differences, first two original values
- HUF: Huffman encoding (default code Table C.9)

Table C.11 — Data obtained during SCP-ECG high compresssion

Sample number	RAW	TRU	RES	FIL	DEC	2D	HUF
1	63	13	13	13	14	14	111111111000001110
2	70	14	14	14			
3	74	15	15	14			
4	71	14	14	15			
5	79	16	16	16	18	18	111111111000010010
6	89	18	18	17			
7	96	19	19	19			
8	102	20	20	20			
9	108	22	22	21	22	0	0
10	112	22	22	22			
11	114	23	23	22			
12	116	23	23	23			
13	116	23	23	23	22	–4	111101
14	112	22	22	22			

Table C.11 (*continued*)

Sample number	RAW	TRU	RES	FIL	DEC	2D	HUF
15	110	22	22	21			
16	100	20	20	19			
17	87	17	17	18	13	–9	11111111011110111
18	74	15	15	14			
19	59	12	12	12			
20	42	8	8	9			
21	28	6	6	6	3	–1	101
22	13	3	3	4			
23	5	1	1	2			
24	–1	0	0	0			
25	–8	–2	–2	–1	–2	5	1111100
26	–11	–2	–2	–2			
27	–13	–3	–3	–3			
28	–17	–3	–3	–3			
...	...	...	...	...	...	...	...

56 bytes = 448 bits → reduced to 71 bits

The first 28 samples of the raw data lead are compressed to this bit stream:

1111111110000011101111111110000100100

1111011111111110111101111011111100...

which is hexadecimal represented by these bytes:

FF 83 BF E1 27 BF F7 BD ...

C.4.2 Example 2

This example shows the different data obtained during SCP-ECG pure redundancy reduction for the first 28 samples of an ECG record.

— RAW: raw data with 1 µV/LSB, sampling interval 2 ms (500 samples/s)

— TRU: truncated raw data 1 µV/LSB → 5 µV/LSB

— RES: residual record, after subtraction of the reference beat

— 2D: second differences, first two data are original values

— HUF: Huffman encoding with default Huffman table (Table C.9)

Table C.12 — SCP-ECG pure redundancy reduction

Sample number	RAW	TRU	RES	2D	HUF
1	63	13	13	13	111111111000001101
2	70	14	14	14	111111111000001110
3	74	15	15	0	0
4	71	14	14	–2	1101
5	79	16	16	3	11100
6	89	18	18	0	0
7	96	19	19	–1	101
8	102	20	20	0	0
9	108	22	22	1	100
10	112	22	22	–2	1101
11	114	23	23	1	100
12	116	23	23	–1	101
13	116	23	23	0	0
14	112	22	22	–1	101
15	110	22	22	1	100
16	100	20	20	–2	1101
17	87	17	17	–1	101
18	74	15	15	1	100
19	59	12	12	–1	101
20	42	8	8	–1	101
21	28	6	6	2	1100
22	13	3	3	–1	101
23	5	1	1	1	100
24	–1	0	0	1	100
25	–8	–2	–2	–1	101
26	–11	–2	–2	2	1100
27	–13	–3	–3	–1	101
28	–17	–3	–3	1	100
...	...	...	...	...	...

56 bytes = 448 bits → reduced to 113 bits

The first 28 samples of the lead are compressed to this bit stream:

1111111110000011011111111110000011100110111101010101001101100 1

0101011001101101100101101110010110010010111001011 00...

which is hexadecimal represented by these bytes:

FF 83 7F E0 E6 F5 53 65 59 B6 5B 96 4B 96 …

C.5 Test set of ECGs for conformance testing

Table C.13 lists the ECGs selected for testing SCP-ECG compression and decompression errors. The data are provided as 10 s digital records with 500 samples/s and 5 µV/LSB quantization level. There are ECGs with sinus rhythm, with atrial fibrillation and atrial flutter, with polyform and monoform ventricular extrasystoles, and with a supraventricular extrasystole. Two cases with major intraventricular conduction defects are included. To verify the absolute calibration, an artificial ECG with low noise and a heart rate of 120 bpm and sinus rhythm has also been included.

These data are selected from the CSE multilead reference database for wave recognition testing and from the CSE diagnostic reference database. A few data have been slightly smoothed to remove some noise. All these ECG recordings have been compressed and decompressed within the standards proposed by the SCP-ECG Working Group. The RMS error figures and the absolute maximum errors found after reconstitution of the original signals are listed in Table C.13.

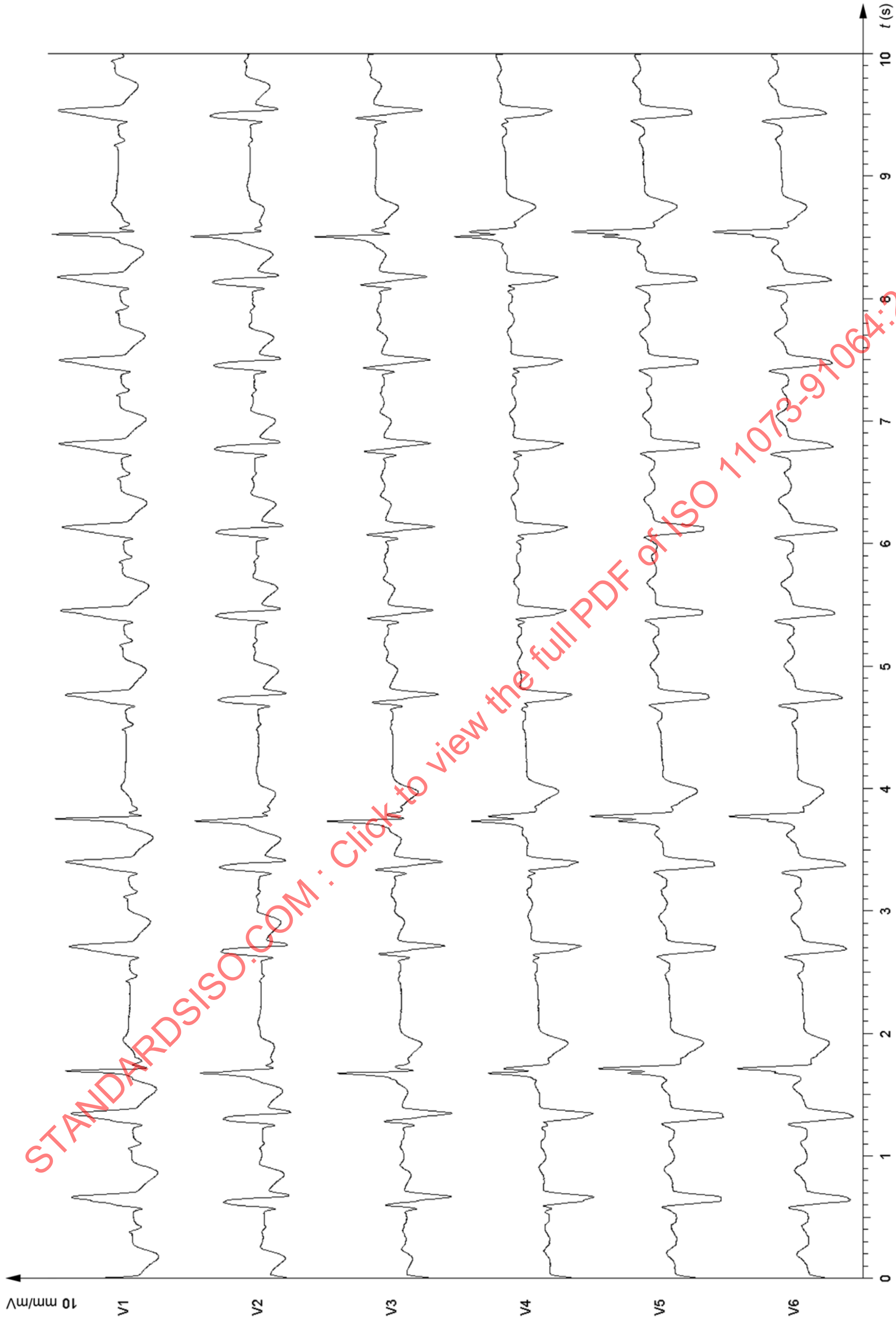


Figure C.7 — ECG for the example data (lead V1 to V6)

Table C.13 — Test set for error verification of SCP-ECG compression requirements

Patient	Rhythm, QRS morphology	Noise
PD2-003 Decompressed	Sinus rhythm, anterior infarction, QRS with notch Absolute maximum error: 64 µV; RMS: 7,9 µV	5 µV
PD2-010 Decompressed	Sinus tachycardia; Absolute maximum error: 67µV; RMS: 8,1 µV	6 µV
PD2-051 Decompressed	Sinus rhythm, infarction Absolute maximum error: 44 µV; RMS: 7,6 µV	3 µV
PD2-078 PF2-078 Decompressed	Atrial flutter, muscle tremor in leads I, and II, large amplitudes (prefiltered) Absolute maximum error: 43 µV; RMS: 8,8 µV	16 µV 10 µV
PD2-217 Decompressed	Sinus rhythm, intraventr. conduction defect, anterior infarction Absolute maximum error: 39 µV; RMS: 7,5 µV	3 µV
PD2-313 Decompressed	Atrial flutter/fibrillation Abs. maximum error: 62 µV; RMS: 8,8 µV	8 µV
PD3-145 PF3-145 Decompressed	Polymorphic ventricular extrasystoles (prefiltered) Absolute maximum error: 77 µV; RMS: 8,6 µV	18 µV 8 µV
PD3-471 Decompressed	Atrial flutter Absolute maximum error: 52 µV; RMS: 8,1 µV	4 µV
PD3-1207 PF3-1207 Decompressed	Supraventricular extrasystoles (prefiltered) Absolute maximum error: 89 µV; RMS: 7,7 µV	6 µV 4 µV
PWE-103 PFE-103 Decompressed	Polymorphic ventricular extrasystoles (prefiltered) Absolute maximum error: 41 µV; RMS: 8,6 µV	26 µV 8 µV
PWE-105 PFE-105 Decompressed	Ventricular extrasystoles, complete right bundle branch block (prefiltered) Absolute maximum error: 54 µV; RMS: 7,7 µV	19 µV 12 µV
P120-N00 Decompressed	Sinus tachycardia, normal QRS complex-mathematically constructed Absolute maximum error: 16 µV; RMS: 5,0 µV	0 µV 3 µV

Annex D
(informative)

Definition of a minimum set of control and query messages for the interchange of ECG data

D.1 General

The messaging part of the SCP-ECG standard describes the type of information that can be requested and transmitted, at the application layer level, between devices, and what the format of the message headers are. This document defines the structures for data messages and gives the sequences needed for data transfers and queries required by the protocol along with the format of each message type. Also described is the use of advisory messages. The data content is described in Clause 5. Knowledge of the data content for the protocol is necessary to understand the messaging part.

D.2 Message formats

D.2.1 General

Each message block is 256 bytes long. The first byte of each message block is an ASCII character specifying the message type. The types currently defined (minimum set) are I, R, S, A and D. Default binary data and reserved fields should be filled with zeros. Text character strings should be NULL terminated. The implementer is responsible for insuring that all messages have a length of 256 bytes or less. Data fields containing free-format strings (Section 1, Tag 14, for example) may need to be truncated to comply with this restriction. Messages longer than 256 bytes will be treated as improperly formatted.

All superscripts in Tables D.1 to D.8 refer to the notes [a) to l)] listed in D.2.7.

D.2.2 Identification data interchange (Message type = “I”)

Table D.1 — Identification data interchange (Message type = “I”)

	Byte number
Message type = “I” (ASCII, 1 byte)	1
Institution identification number [a] (binary)	2 to 3
Department identification number [a] (binary)	4 to 5
Device identification number (ID) [a] (binary)	6 to 7
Device type [a] (0=Cart, 1=System) (binary)	8
Manufacturer code [b] (binary)	9
Model description [b] (up to 5 ASCII characters e.g., “0107B”,“MAC15”,“4760A”+ NULL)	10 to 15
SCP-ECG protocol revision number (binary)	16
SCP-ECG protocol compatibility level (binary)	17
Language support code [c] (1-byte bit mapped)	18
Capabilities [d] (binary)	19
AC mains frequency environment (binary)	20
Reserved	21 to 128
Acquisition device SCP implementation software identifier. Maximum 24 characters plus NULL terminator from 5.4.5, tag 14.	129 to 153
Manufacturer of acquisition device – registered trade name. Maximum 24 characters plus NULL terminator from 5.4.5, tag 14.	154 to 178
SPARE [l]	179 to 256

D.2.3 Request (Message type = "R")

Table D.2 — Request (Message type = "R")

	Byte number
Generic request contents (bytes 1 to 14)	
Message type = "R" (ASCII, 1 byte)	1
Processing request [e] (ASCII, 1 byte)	2
Sub-request [f] (1 binary byte)	3
Request sequence number (1 unsigned integer; 1 – 65 535)	4 to 5
Password, 9 ASCII characters (optional) [g]	6 to 14
[See notes a) to l) in D.2.7 and D.2.3.1 to D.2.3.3 for definitions]	15 to 256

NOTES

— For security reasons, any request should be processed only if the ID messages of the 2 communicating systems (cart and host) have been exchanged successfully.

— Unknown or don't care fields are set to NULL.

— Each request message starts with a 14 byte long "Generic Request" field followed by a 242 byte field (bytes 15 to 256) which contains patient data as specified in D.2.2.1 to D.2.2.3. Each parameter shall be stored in a separate field, each identified by a leading specification byte, referred to as a "tag", followed by the length (an unsigned integer) referred to as "length", followed by zero or more parameter bytes, referred to as "value" (see D.2.2.1 to D.2.2.3).

— A request sequence number counter should be incremented for each "request" message and may rollover to a value of 1 (one). Reset to 1 (one) with each "ID" message from receiving machine. A value of 0 (zero) is not allowed.

— Request sequence numbers are intended to be used to detect redundant and/or missing "Request" messages.

D.2.3.1 Search rules for subrequest types “E” or “L”

Table D.3 — ECG list request (subrequest type “E” or “L”)

	Field length in bytes
Generic request (see D.2.2)	14
Institution number (1 binary integer)	2
Department number (1 binary integer)	2
Patient ID, tag = 2 (binary)	1
Patient ID length (binary)	1
Patient ID, text (text characters)	variable
Patient last name, Tag = 0 (binary)	1
Patient last name length (binary)	1
Patient last name (text characters)	variable
Patient first name, tag = 1 (binary)	1
Patient first name length (binary)	1
Patient first name (text characters)	variable
Patient sex tag = 8 (binary)	1
Patient sex length = 1 (binary)	1
Patient sex (byte)	1
Patient date-of-birth (DOB), tag = 5 (binary)	1
Patient DOB length = 4 (binary)	1
Patient DOB (4 byte binary) [h]	4
SPARE [l]	variable

Total 256 Bytes

D.2.3.2 Search rules for Subrequest Types “I” or “P”

Table D.4 — Patient List Request (Subrequest type “I” or “P”)

	Field length in bytes
Generic request (see D.2.2)	14
Institution number (binary)	2
Department number (binary)	2
Patient ID, tag = 2 (binary)	1
Patient ID length (binary)	1
Patient ID, text (text characters)	variable
Patient last name, tag = 0 (binary)	1
Patient last name length (binary)	1
Patient last name (text characters)	variable
Patient first name, tag = 1 (binary)	1
Patient first name length (binary)	1
Patient first name (text characters)	variable
SPARE [l]	variable
	Total 256 Bytes

D.2.3.3 Search rules for Subrequest Types “R” or “S”

Table D.5 — Test Request (Subrequest type “R” or “S”)

	Field length in bytes
Generic request (see D.2.2)	14
Institution number (binary)	2
Department number (binary)	2
Patient ID, tag = 2 (binary)	1
Patient ID length (binary)	1
Patient ID (text characters)	variable
Patient last name, tag = 0 (binary)	1
Patient last name length (binary)	1
Patient last name (text characters)	variable
Patient first name, tag = 1 (binary)	1
Patient first name length (binary)	1
Patient first name (text characters)	variable
Patient sex, tag = 8 (binary)	1
Patient sex length = 1 (binary)	1
Patient sex (binary)	1
Patient date of birth (DOB), tag = 5 (binary)	1
Patient DOB length = 4 (binary)	1
Patient DOB (4 byte binary)	4
Date of acquisition, Tag = 25 (binary)	1
Date of acquisition length = 4 (binary)	1
Date of acquisition (4 byte binary, format same as DOB)	4
Time of acquisition, tag = 26 (binary)	1
Time of acquisition length = 3 (binary)	1
Time of acquisition (3 byte binary) [i]	3
SPARE [l]	variable

Total 256 bytes

The data sent in response to “L” and “P” requests consists of a series of SCP-ECG header sections, consisting of Sections 0 and 1 only, for each report or patient which matches the search criteria.

D.2.3.4 Search rules for subrequest types “L” and “P”

1) When requesting to receive an ECG list (subrequest type “L”) or a patient list (subrequest type “P”) the following search rules are applied:

   i) any field may or may not be specified;

   ii) if a field is unspecified, it is assumed to be “don't care” and not used as part of the search criteria;

   iii) if a field is specified without wildcards [see iv)], an exact match is required;

   iv) the patient ID and name fields may include “wildcards”, these are determined as follows:
   ? : matches exactly one character to that location,
   * : matches zero or more characters at that location,
   both * and ? may be literally interpreted by preceding either with a backslash (\);

   v) all searches are case insensitive (upper or lower case characters are equally accepted).

EXAMPLE 1 Patient ID = 123, patient last name and first name not specified, matches all patient names with patient ID=123

EXAMPLE 2 Patient last name = “M*CFARL*” matches “MCFARLEY”, “MacFarlane”, “Macfarlane”, “Mcfarlane”, etc.

EXAMPLE 3 Patient ID = “*FAR?” matches “*FAR1”, “*FAR2”, but not “*FAR99”.

2) These search rules apply only to requests for lists, and do not apply to requests for ECGs (subrequest type “R”) for which the search criteria are exact for the required parameters (institution, department, patient ID, date and time of acquisition).

3) A Tag of type 255 with length 0 is used to mark the end of the Tag data in a “list request” message.

D.2.4 Status (Message type = “S”)

The format of the status Message is as follows.

Table D.6 — Status (Message type = or “S”)

	Byte number
Message type = “S” (ASCII character)	1
Status flag [j] (ASCII character)	2
Error-reason code [k] (binary)	3
Request sequence number (binary): 1 to 65 535	4 to 5
SPARE [l]	6 to 256

The request sequence number (RSN) in this message is the RSN of the “request” message to which the Status message responds.

Spurious “STATUS OK” messages received prior to an expected “DONE” message will be ignored.

D.2.5 Advisory (Message type = "A")

The format of the Advisory Message is as follows.

Table D.7 — Advisory (Message type "A")

	Byte number
Message type = "A" (ASCII character)	1
Reserved	2 to 3
Advisory message (text character string, NULL terminated)	4 to 256

Guidelines for the use of Advisory Messages are given in D.5.

D.2.6 Done (Message type = "D")

A "DONE" message indicates the completion of a command. The format of the done message is as follows.

Table D.8 — Done (Message type "D")

	Byte number
Message type = "D" (ASCII character)	1
Reserved	2 to 3
SPARE [l]	4 to 256

Notes on the done message:

— a "pending termination" (PT) flag is maintained by each device;

— a master device changes its status to slave by sending the "D" message and setting its PT flag;

— a slave device changes its status to master and sets its PT flag on the reception of a "D" message;

— the reception by a slave device of a "D" message, without an intervening "R" message and with its PT flag set, will cause the end of the session and the termination of the link;

— a slave device without the PT flag set may send a "D" message without changing to a master device and without modifying the state of the PT flag in either the master or the slave device;

— a master device, on receipt of a "D" message without its PT flag set, will set its PT flag;

— a master device with its PT flag set, on receipt of a "D" message, will terminate the link.

D.2.7 Notes for D.2.2 to D.2.6

a) These fields uniquely identify the requesting device and its location.

b) Manufacturer codes and the ECG recorder's model designation are defined in the patient demographic and acquisition data fields of Section 1 (see 5.4.3.1 and 5.4.5).

c) The language support code is as defined in 5.4.5, Section 1, tag 14, byte 17.

d) Device capabilities are bit-encoded as follows:

1 (LSB)	Spare
2	Spare
4	Spare
8	Spare
16	Print ECG
32	Interpret ECG
64	Store ECG
128 (MSB)	Acquire ECG

e) Processing request is one ASCII character, defined as follows:

"E" -	Request to Send ECG List for Specified Patient
"I" -	Request to Send Patient List for Specified Name
"L" -	Request to Receive ECG List for Specified Patient
"P" -	Request to Receive Patient List for Specified Name
"R" -	Request to receive ECGs
"S" -	Request to send ECGs
"X" -	Escape to Vendor specific request

f) This field is only used for "R" and "X" requests. For these types of request the receiver looks here for the request subcode.

For "R" requests, this field allows multiple tests to be requested without the need for multiple requests, or the need to necessarily know the date and time of the test. This field is a bit map. Values are defined as follows:

0 (no bits set)	= No request mask needed: send all ECGs.
1 (LSB)	= Requested ECG, will have date and time.
2	= Latest ECG.
4	= 1st previous ECG.
8	= 2nd previous ECG.
16	= Baseline ECG if present, otherwise oldest ECG.
32	= All since, will have date and time.

For "X" requests, subrequest codes are vendor-specific extensions and are not defined here.

g) This field contains a nine-character NULL terminated ASCII password, if required by the receiving system.

h) The format for dates (see 5.4.5, Section 1, tag 5 and 25) is as follows:

Bytes 1 to 2	Binary: year.
Byte 3	Binary: month (01 to 12).
Byte 4	Binary: day of month (01 to 31).

i) The Format for Time (see 5.4.5, Section 1, tag 26) is as follows:

Byte 1	Binary: hours (0 to 23).
Byte 2	Binary: minutes (0 to 59).
Byte 3	Binary: seconds (0 to 59).

j) The status flag is one character (ASCII) defined as follows:

"G" - OK (go ahead); error code not used. Receiver should resume and send the next message.

"E" - Not OK, error:
- if ID error, call is terminated immediately.
- if Request or ECG data error, receiver may send another request or send done.

k) For a status “E” (error status) flag, this field contains a binary code indicating the reason for the error. Codes are as follows:

0 = non-specific error.

1 = last message type not recognised.

2 = last message invalid.

3 = invalid location code; the location (i.e. the institution or department number) sent is not defined.

4 = invalid password.

5 = processing request not recognised: processing device does not understand request.

6 = processing request not supported: processing device does not support request.

7 = processing request not supported: processing device has insufficient memory.

10 = patient ID invalid.

11 = patient name invalid.

12 = patient demographic data invalid; data other than ID or name is not of a valid type or range.

13 = patient demographic data inconsistent; the data do not agree with that currently stored for the patient.

These codes all pertain to the signal data and the support for it in the receiving device:

20 = incorrect sample rate.

21 = incorrect lead combination.

22 = incorrect lead duration.

23 = incorrect data compression.

24 = other ECG data error.

30 = measurements invalid; one or more of the measurements was/were invalid.

31 = diagnosis invalid.

40 = output device not ready.

41 = storage device not ready.

42 = database error.

43 = other system error.

44 = insufficient memory error.

50 = no ECGs for this location.

60 = no data matches request.

Codes in the range 128 to 255 are reserved for manufacturer-specific error codes.

l) Areas defined as “SPARE” in message data blocks are available for manufacturer-specific implementation.

D.2.8 Minimum functionality

To be compliant with the specifications of this section, a system consisting of a cart and host is able to:

1) Exchange “ID” messages.

2) Respond to an “RR” request with a subcode of 0 as defined in D.2.7, f).

3) Transfer “All” ECGs as defined in D.2.7, e) or f).

D.3 State diagrams

D.3.1 Establishment of session state diagram

Figure D.1 describes the process by which ID messages are exchanged by cart and host devices in order to establish an SCP-ECG Query-Messaging session. Note that the session does not enter the “connected” state until ID messages (D.2.2) have been successfully exchanged. See D.3.2 for a description of operation after the session has been established.

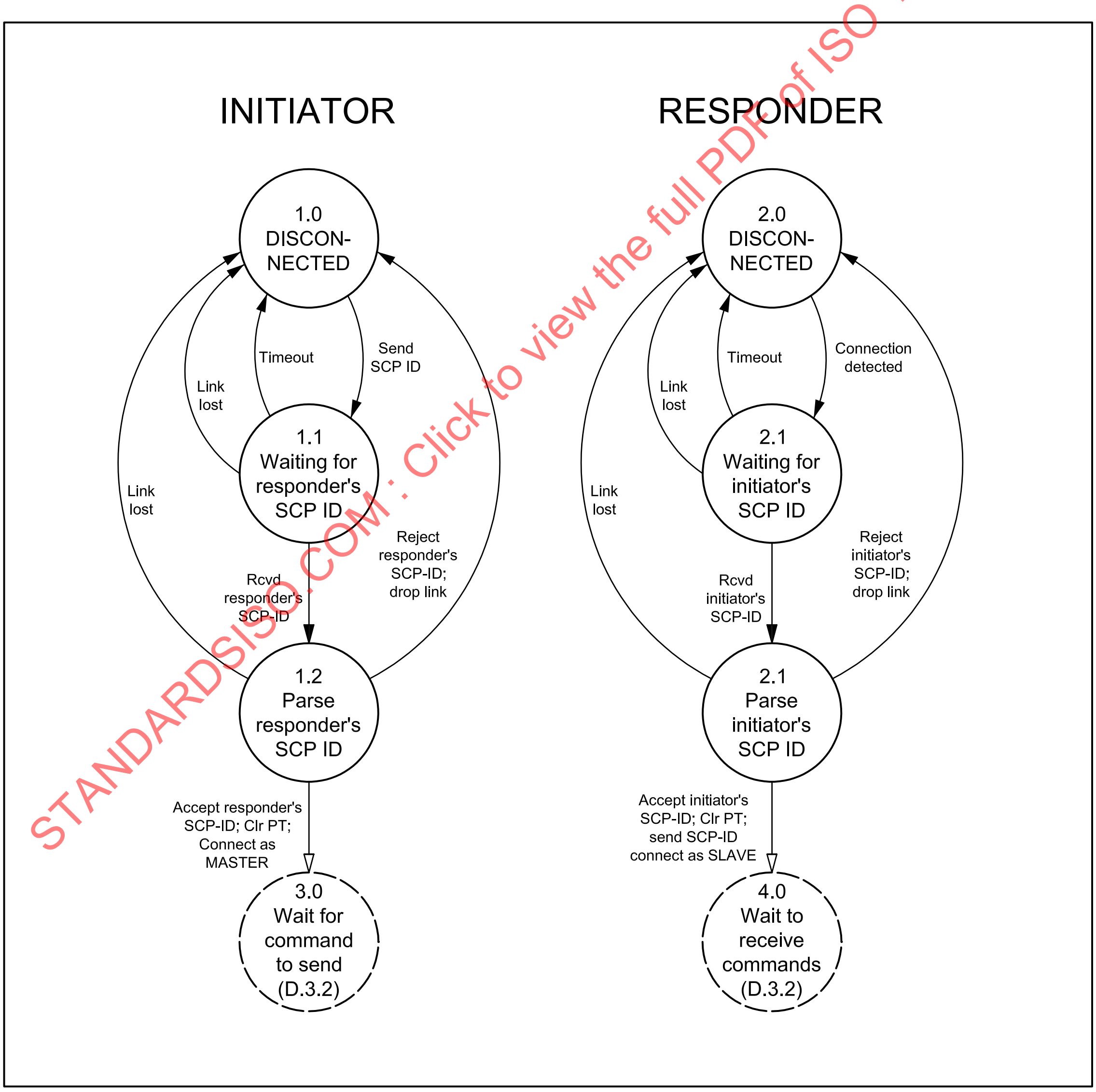


Figure D.1 — Session state diagram

D.3.2 Query messaging system state diagram

The following state diagram (Figure D.2) illustrates the proper operation of the query messaging system following the exchange of “ID” messages as described in D.3.1. A device exists in one of two states while executing the query messaging system: master or Slave. A master device is allowed to send commands. A slave device may only respond to commands.

Manufacturer-specific messages and responses are not required to adhere to this diagram.

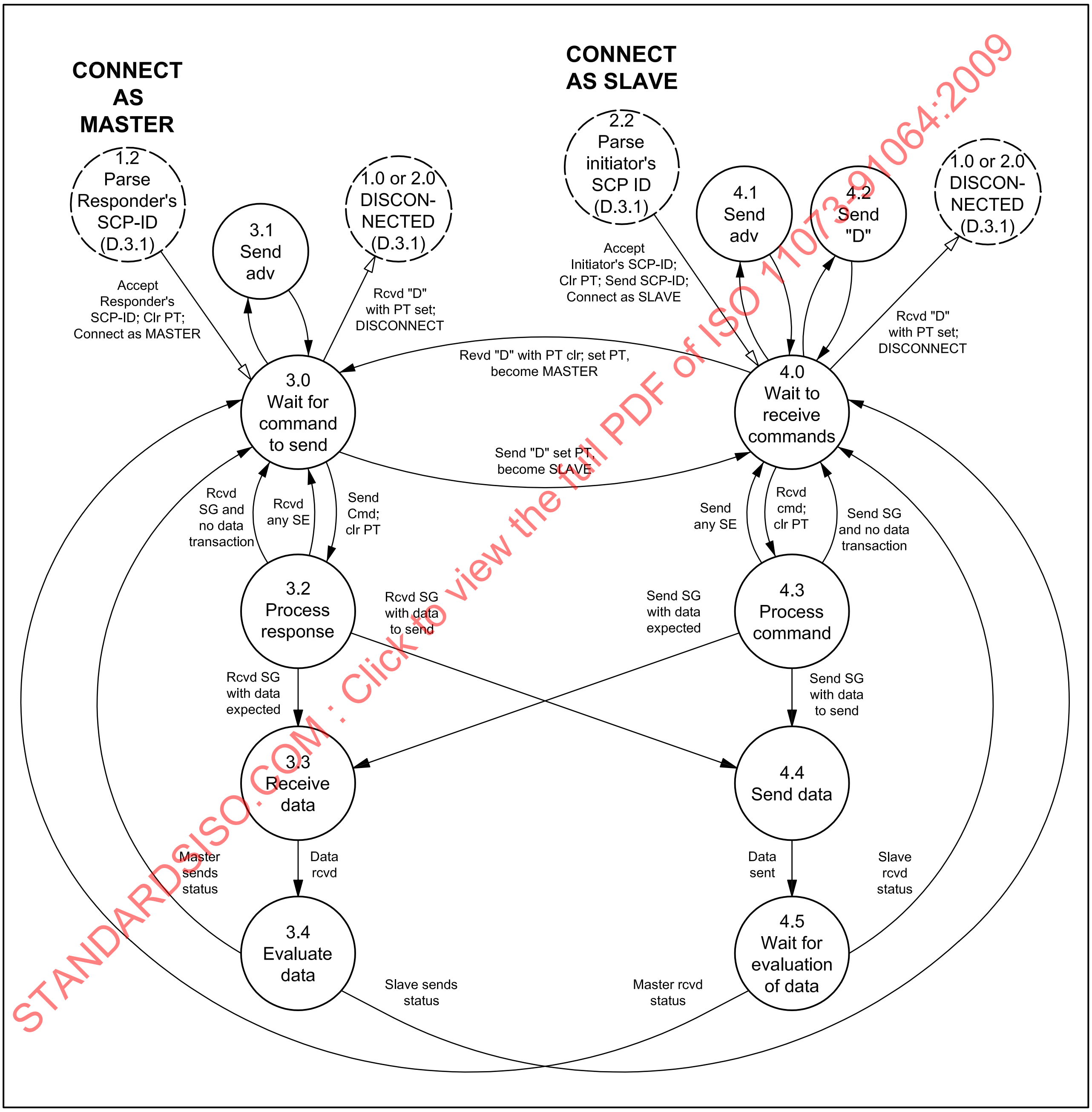


Figure D.2 — The query messaging system state diagram

D.4 Message sequence examples

D.4.1 General

In the following examples, “ECG Data” refers to an SCP-ECG formatted data file, and “ID”, “Request”, “Status”, “Done” and “Advisory” refer to SCP-ECG messages defined in D.2.

D.4.2 ECG Transfer

The normal message sequence for sending an ECG from an initiator to a responder is as follows:

	Initiator	Responder	Comment
a)	ID		Log-on: Exchange of Identification Data
b)		ID	via message type “I”.
c)	Request		Initiator requests to send an ECG (message type “R”).
d)		Status	Responder replies: ready to receive (message type “S”).
e)	ECG Data		Initiator sends data.
f)		Status	Responder replies: received data are OK (see Note 1 under D.4.5).
g)	Request		Initiator sends another request (see Note 2 under D.4.5) [go to d) above or D.4.3 to D.4.5]
	or		or
	Done		says “no more data” (message type “D”).
h)		Request	Responder sends its own request (go to D.4.5, below)
		or	or
		(done)	terminates the call.

D.4.3 Patient list request

The normal message sequence for requesting a list of patients who match the request mask is as follows:

	Initiator	Responder	Comment
a)	ID		Log-on, exchange of identification data.
b)		ID	
c)	Request		Initiator requests a patient list.
d)		Status	Responder replies OK. Responder keeps line alive until request can be fulfilled.
e)		Request	Responder requests to send a patient list.
f)	Status		Initiator replies ready to receive.
g)		Data	Responder sends ECG data.
h)	Status		Initiator replies received data are OK (see Note 1 under D.4.5).
i)		Done	Responder says no more data.
j)	Request		Initiator sends another request (see Note 2 under D.4.5) [go to d) above or D.4.4 and D.4.5]
	or		or
	Done		says “no more data”.
k)		Request	Responder sends its own request (go to D.4.5, below)
		or	or
		(done)	terminates the call.

D.4.4 ECG list request

The message sequence for requesting a list of ECG reports for a known patient from a responder is as follows:

	Initiator	Responder	Comment
a)	ID		Log-on, exchange of identification data.
b)		ID	
c)	Request		Initiator requests a list of ECGs.
d)		Status	Responder replies OK. Responder keeps line alive until request can be fulfilled.
e)		Request	Responder requests to send the list of ECGs.
f)	Status		Initiator replies ready to receive data.
g)		Data	Responder sends data.
h)	Status		Initiator replies received data are OK (see Note 1 under D.4.5).
i)		Done	Responder says “no more data”.
j)	Request or Done		Initiator sends another request [go to d) above or to D.4.3 to D.4.5] (see Note 2 under D.4.5) or says “no more data”.
k)		Request or (done)	Responder sends its own request (go to D.4.5) or terminates the call.

D.4.5 ECG report request

The message sequence for requesting a specific report or reports for a known patient from a responder is as follows:

	Initiator	Responder	Comment
a)	ID		Log-on, exchange of identification data.
b)		ID	
c)	Request		Initiator requests to receive an ECG.
d)		Status or Request	Responder returns an error status if no ECG to send or a request to send an ECG.
e)	Status		Initiator replies ready to receive.
f)		ECG Data	Responder sends data.
g)	Status		Initiator replies received data are OK (see Note 1).
h)		Request or Done	Responder sends another request [go to e) above] (see Note 2) or says “no more data”.
i)	Request (done)		Initiator sends its own request (go to D.4.2 to D.4.4) or terminates the call.

NOTE 1 If the received data are not OK, the responder replies with an error code (see D.2.4). The initiator then requests g) above to retransmit the same data, other (new) data or send “Done”.

NOTE 2 Through this mechanism a “batch” of ECGs or a “set of patient lists” can be transmitted. See also Example 2 in D.5.

D.5 Use of advisory messages

At any point in the above sequence where a control message (ID, Request, Status or Done) could be sent, one or more Advisory messages may be interjected. An Advisory message has no effect on the processing sequences. It is a means of providing additional information to a human operator.

An Advisory message does not end with a line turn around. When an Advisory message is received instead of another control message, the receiver processes the Advisory (normally by displaying a message to the operator) and continues in the same state, waiting for the originally expected control message before proceeding.

Examples of the use of Advisory messages are as follows:

EXAMPLE 1

	Initiator	Responder	Comment
a)	ID		Log-on, exchange identification data.
b)		ID	
c)	Request		Initiator requests to send an ECG.
d)		Status	Responder replies ready to receive.
e)	ECG Data		Initiator sends data.
		Advisory	“Default used for patient age”.
		Advisory	“Please wait 30 seconds”.
f)		Status	Responder replies received data are OK.
g)	Done		Initiator says “no more data”.
h)		(done)	Responder terminates the call.

EXAMPLE 2

	Initiator	Responder	Comment
a)	ID		Log-on, exchange of identification data.
b)		ID	
c)	Request		Initiator requests to receive an ECG.
		Advisory	“Sending #1 of 1 ECGs cued”.
d)		Request	Responder requests to send an ECG.
e)	Status		Initiator replies ready to receive.
f)		ECG Data	Responder sends data.
g)	Status		Initiator replies received data are OK.
h)		Done	Responder says “no more requests”.
i)	(done)		Initiator terminates the call.

Annex E
(informative)

Standard low-level ECG-Cart to host protocol

E.1 General

The specification for this low-level transport protocol for communication of ECGs between digital ECG carts and computerized ECG management (storage) systems consists of two functional layers:

— the data link function layer;

— the physical function layer.

Communication between two ECG management systems, and communication between these systems and other hosts is not within the scope of the specification.

E.2 Data link and physical functional layers

In Clause E.3 a brief description is given of the minimum requirements for local and remote connection and transfer of ECG related data, permitting the use of low-cost, high-speed asynchronous modems or simple RS-232-C local lines. Its purpose is to ensure that devices utilizing this standard protocol are able to communicate.

Clause E.5 describes the methods used to ensure that the two devices communicating are synchronized and that the data are not corrupted during the transfer process. The document describes the states necessary to handle the exception conditions.

When communicating data between devices, especially between compressed binary data which are particularly susceptible to corruption by errors during transmission, it is important to employ some device or software by which the integrity of the data and data link is ensured. Such devices or software are available in many formats, e.g. lower-level network protocols, error-correcting modems, etc. In this annex a low-level protocol is described, which can be used when no other adequate protocol is available or appropriate.

Also included are state transition diagrams to aid in the understanding of this layer, the algorithm for the CRC of each data packet and the format of each of the defined data blocks. Basically the data link layer is an enhanced XMODEM protocol.

E.3 Physical functional layer

E.3.1 General description

In this clause a description is given of the physical layer of the standard protocol for the transmission of ECGs. Minimum requirements for both “local” and “remote” connections are given. Local connections are defined as a point-to-point direct connection. Remote connections are those that utilize public switched telephone networks or the equivalent.

Individual manufacturers may decide to exceed the requirements given here or use other physical layers. The standards described here are not meant to impede future system development or degrade system performance but rather to provide a common interface which provides reasonable performance with minimal cost of implementation while maintaining a common method of ECG transmission between vendors.

E.3.2 Local connections

The local connection consists of an RS-232-C link. The link is able to support the following parameter set:

9 600 BAUD
8 Data Bits
1 Stop Bit
No Parity

E.4 Remote connection

The minimum requirement for a remote connection is a V.22bis modem with the appropriate government approval(s) for the installed country. The V.22bis standard calls for a 2 400 baud full duplex asynchronous modem. MNP level 5 error detection and correction is not a requirement since the error handling layer is designed to be able to handle that function. Other requirements are the same as those for local connections.

E.5 Data link functional layer

E.5.1 General description

E.5.1.1 This clause defines the error correcting and line arbitration layer of the standard protocol for the transmission of ECGs. The purpose of this clause is to define a minimal standard to allow carts and systems from different manufacturers to communicate. It is not intended as a definition or suggested method for system-to-system communications. This protocol level is referred to as the data link layer.

The data link layer is a modified (enhanced) version of XMODEM. It is intended to be easy to implement using "standard" asynchronous communications devices, and yet to address the performance requirements for ECG communications.

Specifically these are as described in E.5.1.2 to E.5.1.6.

E.5.1.2 Timeouts were reduced from 10 s to 2,5 (t1) s and 3,5 (t2) s, thereby reducing execution time when either machine's performance is less than optimum.

E.5.1.3 A temporary text delay was introduced to allow a machine to keep the link alive when there are no data to send. This supports the situation where a transmitting device is expected to transmit a data block, but the data are not ready due to processing delays. Under normal XMODEM the receiver would timeout several times waiting for data, then abort the link.

E.5.1.4 The ability for a transmitting device to request retransmission of the latest data block acknowledgement message was added, to cover the situation in which the receiver's ACK or NAK was garbled. Under standard XMODEM, the transmitting station simply retransmits its data block, a potentially lengthy process.

E.5.1.5 The block size is larger, 256 bytes.

E.5.1.6 The protocol allows "line turn around" so that data can be transferred in both directions during a single session.

For the purposes of identifying the participating station in the following description, the two devices are referred to as the "transmitter" and the "receiver". The device that is transmitting data or that last transmitted data is the master. These roles can change during a communications session (see also Figure E.2).

During each state described below, except for TRANSMIT WAKE-UP and RECEIVE WAKE-UP, both stations have the capability of receiving or transmitting an EOT termination code. EOT capability at any time is necessary to handle abnormal terminations. The same mechanism may be used for normal termination because there is no difference between the two methods at the level of the data link layer.

E.5.2 Transmit machine

E.5.2.1 For graphical representation of the transmit machine, see Figures E.3 to E.5.

E.5.2.2 The transmit machine consists of three states: TRANSMIT WAKE-UP, NO DATA WAIT and WAIT FOR ACKNOWLEDGE.

E.5.2.2.1 TRANSMIT WAKE-UP: This is the entry state for the calling device. The answering device is expected to send an ASCII 'ENQ'. Upon receipt of an ENQ, if data are ready to be sent, the caller sends the data and switches to the WAIT FOR ACK state. If there is no data ready when the ENQ is received, then NO DATA WAIT becomes the current state. If no ENQ is received within 1 min the process aborts.

E.5.2.2.2 NO DATA WAIT: In this state the transmitter is waiting for data to send. When this occurs, the transmit station is expected to send a data block but the data block has not yet been prepared by the higher level application. This state has only a 2,5 s time-out.

If the 2,5 s timer goes off, the transmit machine checks for data. If they exist, the data are sent, and the current state becomes the WAIT FOR ACK state. If no data are present, a temporary text delay is sent to prevent the receiving station from timing out, and control remains in the NO DATA WAIT state. If data become available before the 2,5 s have elapsed, they may be sent as described above.

E.5.2.2.3 WAIT FOR ACK: In this state the transmitting device waits for a response from the receiver to the last data block sent. Proper responses are ACK or NAK.

E.5.2.2.3.1 If an ACK is received and the transmitter is instructed by the application layer to turn the line around, it transmits an ENQ and control is transferred to the receive state, WAIT FOR TURNAROUND.

E.5.2.2.3.2 If an ACK is received and the transmitter is not expecting turnaround, it checks for the presence of data to send. If data are present they are sent and the transmit machine remains in the WAIT FOR ACK state. If no data are available, the current state becomes the NO DATA WAIT state.

E.5.2.2.3.3 If a NAK is received the last data block is retransmitted. The state remains WAIT FOR ACK. If ten consecutive NAKs have been received, an EOT with error code 006 will be sent, and the link aborted.

E.5.2.2.3.4 If 2,5 s pass without a response, the last packet is retransmitted. After ten transmissions of the same packet have been sent with no response, an EOT is sent with error code 001 and the link will be aborted.

E.5.3 Receive machine

E.5.3.1 For graphical representation of the receive machine see Figures E.6 to E.9.

E.5.3.2 The receive machine has four states: RECEIVE WAKE-UP, WAIT FOR TURNAROUND, WAIT ON DATA and BAD DATA.

E.5.3.2.1 RECEIVE WAKE-UP: This is the entry state for the receive machine. Upon wake-up, the device transmits an ENQ and enters the WAIT FOR TURNAROUND state.

E.5.3.2.2 WAIT FOR TURNAROUND: In this state the receiver waits for the transmission of the first data block from the transmitting device. Two inputs are valid: a data block or a TTD.

E.5.3.2.2.1 If data are received, they are checked for accuracy based on their block number and their CRC. If the data are good an ACK is sent and the current state becomes WAIT ON DATA. If the data are bad a NAK is sent and the receive machine enters the BAD DATA state.

E.5.3.2.2.2 If a TTD is received, no action is taken (other than resetting the time-out clock).

E.5.3.2.2.3 If no data are received within 3,5 s, the ENQ is retransmitted. If after ten ENQs have been sent with no response, the link is aborted and an EOT is sent with an error code 002.

E.5.3.2.3 WAIT ON DATA: This is the normal receive state. The receiver waits here while receiving data blocks from the transmitting device. Four inputs are valid: data, TTD, SYN and ENQ. Non-valid inputs are discarded.

E.5.3.2.3.1 If data are received they are checked for accuracy based on block number and CRC. If the block is good an ACK is sent and the machine remains in the WAIT ON DATA state. If the data are bad a NAK is sent and the current state becomes BAD DATA.

E.5.3.2.3.2 If the new packet just received is the same as the previous packet, send an ACK and discard the packet. After receiving ten consecutive copies of the same packet, an EOT is sent with an error code of 005 and the link is aborted.

E.5.3.2.3.3 If a TTD is received no action is taken (other than resetting the time-out clock).

E.5.3.2.3.4 If a SYN is received an ACK is retransmitted and the WAIT ON DATA state remains valid.

E.5.3.2.3.5 If an ENQ is received, the receiver is about to become a transmitter. The presence of data to send is checked. If no data are present, the receive machine becomes a transmit machine, in the NO DATA WAIT state. If data are present, it is sent and control passes to the transmit state, WAIT FOR ACK.

E.5.3.2.3.6 If no data are received within 3,5 s, a NAK is transmitted. This ensures that the transmitting station resends the last data block if it was lost. A NAK is sent rather than an ACK because resending an ACK may result in a loss of data, if the transmitting station sent a data block which the receiver failed to receive. The machine enters the BAD DATA state.

E.5.3.2.4 BAD DATA: After the receiver sends a NAK it waits in the BAD DATA state. Valid inputs are data, SYN and ENQ.

E.5.3.2.4.1 If good data are received an ACK is sent and the machine enters the WAIT ON DATA state. If the data are bad, a NAK is sent. After ten NAKs have been sent and another bad data block is received, an EOT is sent with an error code 003 and the link is aborted.

E.5.3.2.4.2 If a SYN is received the last NAK is retransmitted. The state remains BAD DATA.

E.5.3.2.4.3 If no data are received within 3,5 s a NAK is transmitted. If ten NAKs have been sent due to lack of input as opposed to bad data and another 3,5 s pass without input, the link is aborted and an EOT is sent with error code 004.

E.5.3.2.4.4 If an ENQ is received, the receiver is about to become a transmitter. The presence of data to send is checked. If no data are present, the receive machine becomes a transmit machine in the NO DATA WAIT state. If data are present, they are sent and control passes to the transmit state, WAIT FOR ACK.

E.5.4 Format of the data blocks

E.5.4.1 The control codes should be sent as follows:

ENQ:	ENQ DC2
ACK:	ACK DC2
NAK:	NAK DC2
SYN:	SYN DC2
TTD:	CAN DC2
EOT:	EOT $FF CHAR CHAR CHAR DC2

where the three characters (CHAR's) are ASCII digits representing the termination code, i.e. the error codes 001 to 006 mentioned in E.5.2 and E.5.3. For normal termination, an error code of **000** will be sent.

E.5.4.2 The format of the data blocks are as follows:

a) Message: SOH
Block # (1 to 255, incremental, recirculating)
255 - Block# (1's complement of Block #)
data (256 bytes of data)
CRCHI
CRCLO

b) Data: STX
Block # (1 to 255, incremental, recirculating)
255 - Block# (1's complement of Block #)
data (256 bytes of data)
CRCHI
CRCLO

NOTE The same block counter is used for message and data blocks, and should be incremented for each occurrence of either type of block.

E.5.5 CRC error detection algorithm

The CRC is based on CRC-CCITT (X16 + X12 + X5 + 1). The CRC is a 16-bit quantity and should be preset to all 1s (FFFF hex) at the start of the calculation for each block of data. The CRC is calculated over the entire data block up to the CRC itself, as shown in Figure E.1.

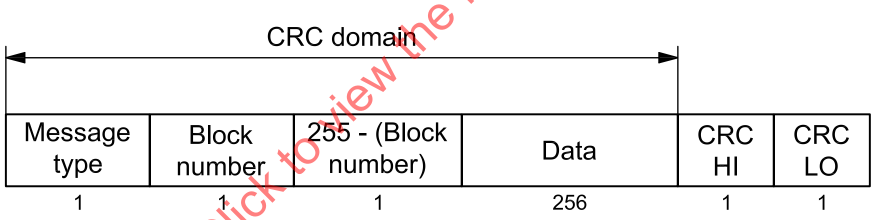


Figure E.1 — CRC-CCITT error detection

The algorithm for the CRC-CCITT is below described. Note that all operations are on bytes.

A = new byte

B = temp byte

CRCHI = High byte (most significant) of the 16-bit CRC

CRCLO = Low byte (least significant) of the 16-bit CRC

START:

FOR A = FIRST_BYTE TO LAST_BYTE IN BLOCK DO:

A = A XOR CRCHI

CRCHI = A

SHIFT A RIGHT FOUR TIMES (ZERO FILL)

A = A XOR CRCHI { I J K L M N O P }

CRCHI = CRCLO	{ swap CRCHI, CRCLO }
CRCLO = A	
ROTATE A LEFT 4 TIMES	{ M N O P I J K L }
B = A	{ temp save }
ROTATE A LEFT ONCE	{ N O P I J K L M }
A = A AND $1F	{ 0 0 0 I J L L M }
CRCHI = A XOR CRCHI	
A = B AND $F0	{ M N O P 0 0 0 0 }
CRCHI = A XOR CRCHI	{ CRCHI complete }
ROTATE B LEFT ONCE	{ N O P 0 0 0 0 M }
B = B AND $ E0	{ N O P 0 0 0 0 0 }
CRCLO = B XOR CRCLO	{ CRCLO complete }

DOEND;

FINISH.

Final check on the CRC is accomplished by adding or concatenating CRCHI and CRCLO at the end of the data stream. Calculating the CRC of the resulting data stream will result in a zero CRC if the data were correctly received.

E.5.6 State transition diagrams (STD)

E.5.6.1 Data flow diagram (not an STD)

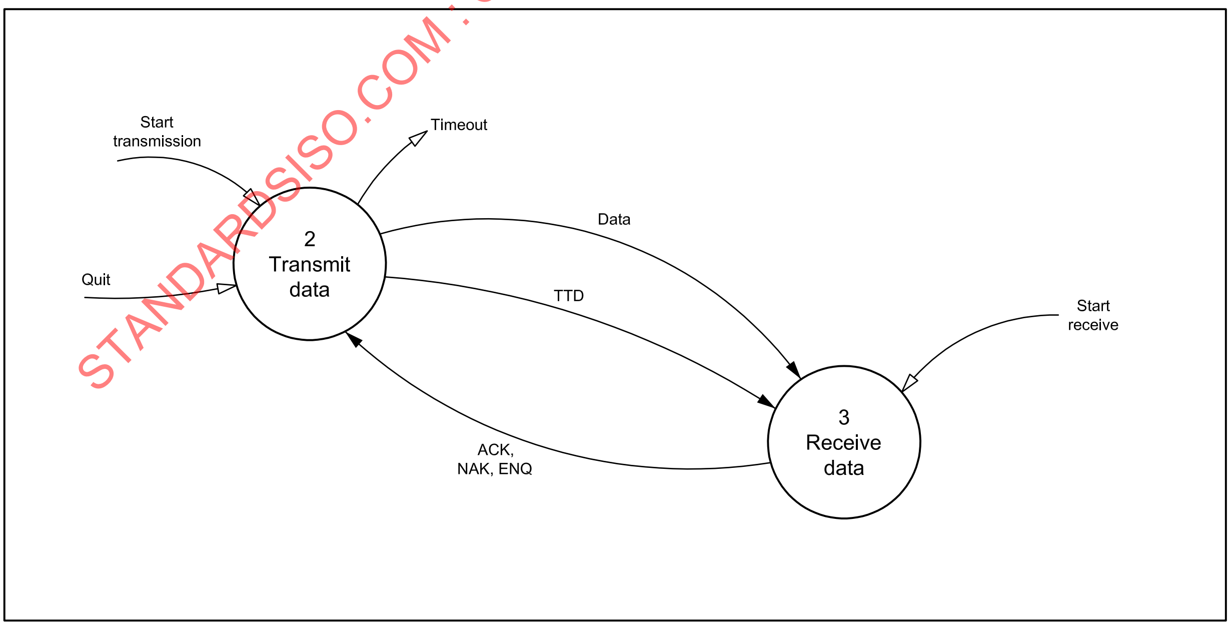


Figure E.2 — Data flow diagram