SURFACE VEHICLE INFORMATION REPORT	J3054	JAN2015
	Issued 2015-01	

TP1.6 Vehicle DIAGNOSTIC Protocol

RATIONALE

Some Volkswagen of America and Audi of America vehicles are equipped with ECU(s), in which a TP1.6 proprietary diagnostic communication protocol is implemented. This document is needed to specify the requirements necessary to implement the TP1.6 communication protocol in an SAE J2534 interface.

TABLE OF CONTENTS

1. SCOPE

This Technical Information Report defines the diagnostic communication protocol TP1.6. This document should be used in conjunction with SAE J2534-2 in order to fully implement the communication protocol in an SAE J2534 interface.

Some Volkswagen of America and Audi of America vehicles are equipped with ECU(s), in which a TP1.6 proprietary diagnostic communication protocol is implemented. The purpose of this document is to specify the requirements necessary to implement the communication protocol in an SAE J2534 interface.

This Technical Information Report describes how a tester can be connected to a vehicle to perform diagnostics using the TP1.6 protocol. Details regarding ECU to ECU communication have been left out.

2. REFERENCES

2.1 Applicable Documents

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or 724-776-4970 (outside USA), www.sae.org.

SAE J2534-1 Recommended Practice for Pass-Thru Vehicle Programming

OSEK Communication V1.0: Fault-Tolerant Communication

3. TERMS AND ACRONYMS

3.1 Ack

Acknowledge message

3.2 AR

Acknowledge request

3.3 BS

Block size

3.4 CA

Connection acknowledge: Dynamic connection configuration response

3.5 CAN Telegram

A CAN message with 11 address bits and up to 8 data bytes used to request, respond or establish a connection.

3.6 ChA

Channel Acknowledge: Dynamic connection establishment response

3.7 ChId

Channel Identifier without offset

3.8 ChS

Channel Setup: Dynamic connection establishment request

3.9 CS

Connection setup: Dynamic connection configuration request

3.10 DC

Disconnect message

3.11 DDC

Data direction change

3.12 DT

Data message

3.13 ECU

Electronic Control Module

3.14 EOM

End of message

3.15 ID or Identifier

Fixed CAN ID assigned to a module.

3.16 MNT

Maximum value during communication

3.17 MNTC

Maximum value with connection structure

3.18 OC

OpCode

3.19 RNR

Passive ECU (Receiver) not ready

3.20 RR

Passive ECU (Receiver) ready

3.21 RS

Receive status

3.22 SN

Sequence number

3.23 TAa

Target Address of Active module

3.24 TAp

Target Address of Passive module

3.25 T_E

Time-out for channel structure

3.26 T1, T3

Parameter ECU (CS / CA)

3.27 TP

Transport protocol

3.28 TPCI

Transport protocol control information byte

3.29 TPDU

Transport protocol data unit

3.30 “Active”

An active module is the module that is sending data to a passive module.

3.31 “Passive”

A passive module is the module that is receiving data from an active module.

3.32 “Identifier”

Each message in TP1.6 has an identifier which is the 11 bit Address of the CAN message. Each module on the CAN bus has been assigned fixed Identifiers. The fixed Identifier is used to set-up a dynamic channel through which the data exchange can flow.

3.33 “Dynamic Channel”

A Dynamic Channel uses dynamic assigned CAN addresses which are set-up when a channel is established. The dynamic channel CAN IDs are used to send the data between nodes.

4. OVERVIEW

This document describes the transport protocol TP1.6 which is an exclusive connection between two CAN (11 bit IDs only) participants for the transmission of large amounts of data.

The CAN-transport protocol includes an agreement for the dynamic assignment of bi-directional transport channels between control modules. It is an extension of the transport protocol that was standardized in the OSEK-communication V1.0.

The generalization of the OSEK- connection is necessary to make dynamic assignments of identifiers to transport connections and to make a discontinuation of a running data transmission and additional timings possible.

For the dynamic identifier a unique address was assigned to each control module for all vehicles and a firm question or answer address channel. By exchanging messages, the systems assign on these channels the transport channels that must then be used.

The major attributes of the TP1.6 transport protocol are:

- Control bytes for channel structure, connection structure, structure confirmation, connection control, data transfer and confirmation
- Half duplex channels
- Confirmation of each telegram or major block of telegrams including error correction
- Interruption of a running data transmission

5. CAN MESSAGE FORMAT

5.1 CAN-Telegram Overview for Channel Setup, Channel Acknowledge

The first step in creating a Transport Protocol connection is to setup a channel. Two messages, the Channel Setup (ChS) message and Channel Acknowledge (ChA) message are used to setup a channel. Channel Setup and Channel Acknowledge messages have a fixed length of 3 data bytes (CAN DLC = 3).

The CAN-telegram for Channel Setup and Channel Acknowledge has the following structure:

Identifier	1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte
11 bits Source	Destination	Opcode	ChId	Not used			

Source:	Fixed Source address of the sending ECU.
Destination:	TAp for Channel Setup, TAa for Channel Acknowledge - target address of the recipient
Opcode:	Channel Set-up 0xC0 Channel Ack-Positive Reply 0xD0 Channel Ack-Negative Reply 0xD8
ChId:	Channel ID, requested lower 8 bits of the CAN ID to be used for the dynamic communication channel.

5.1.1 Channel setup message structure

Channel Setup messages are used to establish a data channel between 2 ECUs (or the tester and an ECU) for the purpose of sending large blocks of data. The Channel Setup message is sent from the active ECU to a passive ECU to see if it can start a Dynamic channel. The receiving passive ECU must send a positive or negative response to the message. The sending active ECU will wait for the response or time-out.

The responding passive ECU must be able to accept the ChId sent by the requester as part of the CAN ID the responder will use to send messages to the requester. The responding passive ECU must fill in the ChId in the Channel Acknowledge message when returning a Positive Reply.

After receiving a Channel Acknowledge-Positive Reply, the sending and responding ECU use the Dynamic Channel to send and receive Transport Protocol Data Unit Telegrams which will be used to send data and manage the Dynamic Channel.

Each ECU, either active or passive, shall support multiple Dynamic channels simultaneously if it has the resources to do so. Each established Dynamic channel will have a unique ChId pair and will operate independently of any other Dynamic channels established on the ECU. An ECU that does not have the resources to support additional Dynamic channels will reply to the Channel Setup message with Channel Acknowledge-Negative Reply.

5.1.2 Channel setup request

5.1.2.1 Source identifier

Each ECU in a vehicle has been assigned a fixed Source identifier which must be used during the channel setup process. The default Channel Setup tester Source identifier for communicating with Drive ECUs is 0x200, for communicating with Comfort ECUs is 0x2D0 and for communicating with Infotainment ECUs is 0x2D0.

The assigned fixed Source identifier is sometimes referred to as the "opening identifier".

Table 1 - Tester source identifier

Tester Type	Tester Setup offset	Tester Target Address range	Tester Source range	Default Tester Target Address	Default Tester Source
Drive	0x200	0x00 – 0x1F	0x200 – 0x21F	0x00	0x200
Comfort	0x2D0	0x00 – 0x0F	0x2D0 – 0x2DF	0x00	0x2D0
Infotainment	0x2D0	0x00 – 0x0F	0x2D0 – 0x2DF	0x00	0x2D0

Table 2 - ECU source identifier

ECU Type	ECU Setup offset	ECU Target Address range	ECU Identifier Range
Drive	0x200	0x00 – 0x1F	0x200 – 0x21F
Comfort	0x2E0	0x00 – 0x1F	0x2E0 – 0x2FF
Infotainment	0x4A0	0x30 – 0x3F	0x4D0 – 0x4DF

5.1.2.2 Destination

The Destination is the Target Address of the passive control unit being addressed (TAp). Target Address values are fixed for each ECU in the vehicle. The active requester's Target Address (TAa) can not equal the Destination Target Address (TAp).

Table 3 - Destination target address range

ECU Type	Destination range
Drive	0x00 – 0x1F
Comfort	0x00 – 0x1F
Infotainment	0x30 – 0x3F

5.1.2.3 OpCode

For a Channel Setup request the OpCode is always 0xC0.

5.1.2.4 Channel ID

The Channel ID is the ID of the Dynamic Channel being requested. The Channel ID (ChId) is added to the Channel offset to yield the actual CAN ID used on the CAN network.

Each ECU type has a different formula for generating the Channel ID. The Device sending the Channel Setup request uses its own Target Address plus a constant or range of constants to generate the Channel ID.

The passive ECU will respond with a Channel Acknowledge message confirming the active ECUs Channel ID and specifying the passive ECU's Channel ID. Note: the Channel IDs from the active and passive devices will be added with the same offset to create the actual Dynamic CANID used to send data.

Table 4 - Channel setup channel IDs

ECU Type	ChId Formula Channel Setup (ChS)	ChId offset	CAN ID used if the communication channel is successfully established
Drive (4 channels defined)	0x40 + Target Address, 0x60 + Target Address, 0x80 + Target Address, 0xA0 + Traget Address	0x700	0x700 + ChId
Comfort	Range of 0x00 – 0x1F	0x300	0x300 + ChId
Infotainment (High priority)	Range of 0xB0 – 0xCF	0x400	0x400 + ChId
Infotainment (Low priority)	Range of 0x60 – 0x7F	0x600	0x600 + ChId

5.1.3 Channel acknowledge response

The Channel Acknowledge message is sent in response to the Channel Setup request message. The ECU receiving the Channel Setup request will decide if it can support the Channel Setup request and respond with a positive acknowledgement if the resources are available to support the Dynamic Channel or respond with a negative acknowledgement if the resources are not available.

5.1.3.1 CAN messages criteria for receiving a channel setup request.

An ECU or Tester will respond to a Channel Setup request when the CANID is in its correct range per the ECUs type and the Destination (byte 1) matches its Target Address.

ECU Type	Range of CANIDs to recieve messages	Requester's Destination = Byte 1
Drive	0x200 – 0x21F	Destination = Target Address
Comfort	0x2D0 – 0x2DF 0x2E0 – 0x2FF	Destination = Target Address
Infotainment	0x2D0 – 0x2DF 0x4D0 – 0x4DF	Destination = Target Address

When the ECU sees a Channel Setup request that has a CAN ID in the correct range and the Destination matches its Target Address then it will generate a Channel Acknowledge message.

The Channel Acknowledge message uses the same CAN Telegram as the Channel Setup request described in 5.1 CAN-Telegram Overview for Channel Setup.

5.1.3.2 Source identifier

The Source identifier used for the Channel Acknowledge message will be created as shown below.

ECU Type	Setup offset	Requester's Destination	Source identifier =
Drive	0x200	N	0x200 + N
Comfort	0x2E0	N	0x2E0 + N
Infotainment	0x4A0	N	0x4A0 + N

5.1.3.3 Destination

The Destination is the Target Address of the control unit being addressed. The responder's Destination (Target Address) is equal to the Request's Source identifier minus the offset.

ECU Type	Requesters Source identifer (CAN ID)	Setup Offset	Destination =
Drive	N	0x200	N - 0x200
Comfort	N (in 0x2D0 – 0x2DF range) N (in 0x2E0 – 0x2FF range)	0x2D0 0x2E0	N - 0x2D0 N - 0x2E0
Infotainment	N (in 0x2D0 – 0x2DF range) N (in 0x4D0 – 0x4DF range)	0x2D0 0x4A0	N - 0x2D0 N - 0x4A0

5.1.3.4 OpCode

For a Channel Acknowledge response the OpCode can either be 0xD0 for a positive response, all resource available to open the communication channel, or 0xD8 for a negative response, communication channel can not be opened.

5.1.3.5 Channel ID

The Channel ID (ChId) is added to the Channel offset to yield the actual Channel identifier (CAN ID).

Table 5 - Channel acknowledge channel IDs

CAN ECU Type	ChId offset	ChId range Channel Setup (ChS)	ChId range Channel Acknowledge (ChA)	CAN ID used if the communication channel is successfully established
Drive	0x700	0x40 – 0xBF	0x40 – 0xBF	0x700 + ChId
Comfort	0x300	0x00 – 0x1F	0x20 – 0x3F	0x300 + ChId
Infotainment (High priority)	0x400	0xE0 – 0xEF	0xF0 – 0xFF	0x400 + ChId
Infotainment (Low priority)	0x600	0x90 – 0x9F	0xB0 – 0xBF	0x600 + ChId

5.1.4 Static CAN-telegram parameters

The following static CAN-Telegram parameters are defined to maintain the link.

Table 6 - Static CAN telegram parameters

Description	**Parameter Name**	**Default Value**
Dynamic Channel structure messages		
Maximum repeats of the connecting structure telegrams	MNTC	20
Time-out for connection structure waiting for response	T_E	100 ms

5.1.5 CAN-telegram error handling

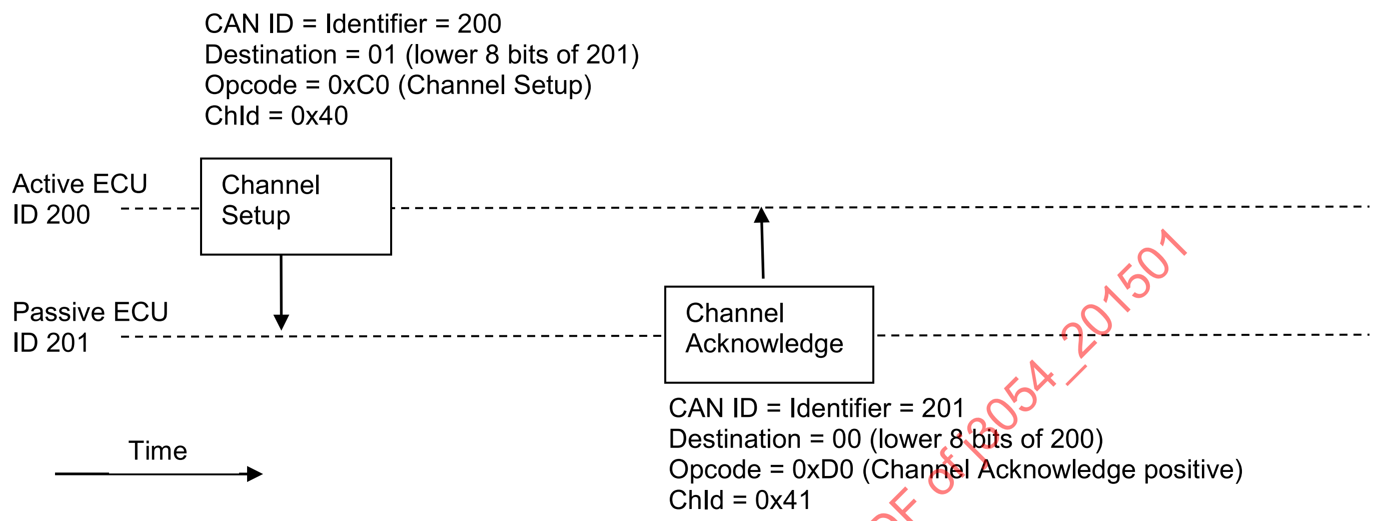
A CAN Telegram that has a parameter out of range or is not formatted correctly shall be ignored.

Table 7 - CAN telegram error handling

Condition	**Action**
Time-out T_E exceeded waiting for Channel Acknowledge	If retries < MNTC then send connection Channel Setup again
MNTC exceeded on waiting for Channel Acknowledge	Give up with a connection Failed

5.1.6 CAN-telegram establishing a channel and connection

Below is an example of how a channel is established.
.


CAN ID = Identifier = 200
Destination = 01 (lower 8 bits of 201)
Opcode = 0xC0 (Channel Setup)
ChId = 0x40
Active ECU ID 200
Channel Setup
Passive ECU ID 201
Channel Acknowledge
CAN ID = Identifier = 201
Destination = 00 (lower 8 bits of 200)
Opcode = 0xD0 (Channel Acknowledge positive)
ChId = 0x41
Time

The active ECU sends a Channel Setup request message to the passive ECU. The passive ECU responds with a Channel Acknowledge response message. After this exchange, both the active and passive ECUs have the channel CAN IDs which will be used by the new connection (0x740 and 0x741). A connection can now be established using the Connection Setup and Connection Acknowledge messages.

5.2 Transport Protocol Data Unit Telegrams on an Established Channel

Once a Dynamic Channel has been established, use the Transport Protocol Data Unit (TPDU) messages to establish a connection, send the data and maintain the channel. A device must maintain up to 4 Transport Protocol Channels simultaneously.

The coding of the Transport Protocol Data Unit (TPDU) Telegram message is as follows:

Identifier	1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte
10 -0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0
From Channel setup	TPCI	BS or Data or NA	Data or T1 or NA	Data or T2 or NA	Data or T3 or NA	Data or T4 or NA	Data or NA	Data or NA

5.2.1 Identifier (CAN ID)

The active ECU will always use the identifier it negotiated in the Channel Setup request (offset + ChId) to send messages to the passive ECU. The passive ECU will always use the identifier it negotiated in the Channel Acknowledge response (offset + ChId) to send messages to the active ECU.

5.2.2 Payload (CAN data bytes)

5.2.2.1 TPCI

The TPCI byte defines 5 different TPDU telegram types:.

- Connection Set-up: The Connection Set-up telegram is used to start a transfer of data. The sending ECU who sends the Connection Set-up is considered the active ECU and is in control of the channel. The sending ECU sends its T1 T2, T3 and T4 values to the receiving ECU. The receiving ECU must always space all its messages back to the sending ECU by T3.

- Connection Acknowledge: The Connection Acknowledge telegram is sent in response to a Connection Setup telegram or a Connection Test telegram. The receiving ECU sends its T1, T2, T3 and T4 values in this message. The Sending ECU must space its messages sent to the receiving ECU by at least T3.

- Data: The Data telegram is used to send 7 or less bytes of data. A bit in the Data TPCI tells the receiver (passive ECU) if an Ack is expected after the Data telegram.

- Acknowledge: The Ack telegram is used to acknowledge the receipt of the Data telegrams. The SN field in the Ack message is used to request retransmission of data telegram. Normally, SN is equal to the last Data Telegram's SN + 1. If the Ack's SN is not equal to the Data telegram's SN+1, the active ECU should re-send the data telegrams starting at the telegram whose SN was specified in the Ack.

- Disconnect: The Disconnect telegram is used by the active ECU to cancel (disconnect) a Transport Protocol channel.

TPDU message telegram types and their message size:

Table 8 - General TPDU formats

TPDU Type	**Acronym**	**TPDU Bytes**							
		0	1	2	3	4	5	6	7
Connection set-up	CS	TPCI = 0xA0	BS	T1	T2	T3	T4	-	-
Connection ack	CA	TPCI = 0xA1	BS	T1	T2	T3	T4	-	-
Data	DT	TPCI	D / -	D / -	D / -	D / -	D / -	D / -	D / -
Acknowledge	Ack	TPCI	-	-	-	-	-	-	-
Disconnect	DC	TPCI = 0xA8	-	-	-	-	-	-	-

'-' Bytes are not sent; the length of the telegram must be adapted for each TPDU type.

5.2.2.2 Control bytes

The control byte (TPCI) has the following structure:

Table 9 – TPCI byte 0

TPDU Type	**Acronym**	**TPCI Byte 0**							
		7	6	5	4	3	2	1	0
Data	DT	0	0	AR	EOM	SN			
Acknowledge	Ack	1	0	RS	1	SN			
Connection set-up	CS	1	0	1	0	0	0	0	0
Connection ack	CA	1	0	1	0	0	0	0	1
Disconnect	DC	1	0	1	0	1	0	0	0

AR - Acknowledge request: 0 = Request; 1 = No Request
EOM - End of Message: 0 = False; 1 = True

RS - Receive status: 0 = Receiver not ready; 1 = Receiver Ready If Receiver is not ready, insert a delay of greater than T3 but less than T1 before sending the next Data telegram.

SN - Sequence number: 0x0 through 0xF. If an Ack is requested, the SN for the Ack is the SN from the data message plus 1.

The control byte BS has the following structure:

Table 10 – BS byte 1

TPDU Type	**Acronym**	**BS Byte 1**							
		7	6	5	4	3	2	1	0
Connect set-up	CS	0	0	0	0	BS			
Connect ack	CA	0	0	0	0	BS			

BS - Block Size: Number of directly following telegrams after which an acknowledgement must be requested. Value range:0 < BS < 16. Use the smaller of either the Set-up BS or the Ack BS. Each ECU must always send the current maximum possible value for the transfer.

Sending priorities of the TPDU types

Table 11 lists the sending priorities of the individual TPDU types. In the case that different types have to be sent at the same time, the type with the highest priority must be transferred on the bus. Smaller numbers denote higher priority.

Table 11 – Sending priority

TPDU Type	Priority
DT	3
Ack	2
CS	4
CA	1
DC	4

5.2.2.3 Dynamic transport protocol timing parameters

Timings and block sizes of the channel are established with the connection set-up / acknowledge message. The transport layer must reset all variables (SN, counter, timer) after the exchange of connection set-up / acknowledge telegrams.

Description of the parameters:

T1	Acknowledge Time-out used by this ECU for received Acknowledge telegrams
T2	Data Time-out used by this ECU for received Data telegrams
T3	minimum time the sending ECU should put between consecutive telegrams being sent to this ECU. For a diagnostic tester, T3 must never be less than 10 ms.
T4	Channel Time-out the maximum time after an acknowledgement before which frames must be sent.

Bytes T1, T2, T3 and T4 are constructed as follows:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Time Base		Time					

Time Base	00 = 100 µsec
	01 = 1 msec
	10 = 10 msec
	11 = 100 msec
Time:	0.63

The time is calculated by multiplying time base by the time

Special cases:

T1 = 0xFF	No Acknowledge Time-out
T2 = 0xFF	No Data Time-out
T3 = 0x00	Immediate successively following telegrams are permissible
T4 = 0xFF	No Channel Time-out

For setting the timings that must be used, the following conditions must be observed because of the sending priorities:

$$T1 > 4 \times T3$$

5.2.3 Transport protocol rules and parameters

For all transport protocol channels and connections the following rules and parameters are statically established:

- Every communication operation is started by setting up a channel using the Channel Setup request and Channel Acknowledge response.

- After a successful channel is established, every communication connection must be configured with a Connection Setup request and Connection Acknowledge response.

- The device that requests the channel (sends the Channel Setup request) sends the first data frame after a connection has been setup.

- On an established connection, the interval between frames must never be shorter than the T3 value specified by the other node.

- On an established connection, the message counter (SN) starts at zero for the first data frame and is incremented with every new data frame. It rolls over from 15 (0x0F) to 0.
- During a data transfer, the active ECU must request an acknowledge (AR = 0) when the exchanged block size (BS) is reached.
- If an acknowledge is requested (AR = 0), the passive ECU (receiver of the data frames) must send the acknowledge frame within the exchanged timeout period T1.
- An Acknowledge can extend the timeout period of the connection by sending RNR. The next acknowledge (with RNR or RR) must be sent after T3 but before T1 expires. The acknowledge is completed when RR is sent.
- After a data transfer with EOM, the connection partners will switch direction of the data flow, the active ECU becomes the passive ECU and the passive ECU becomes the active ECU.
- When the last data message, with EOM, is sent which does request an acknowledge, the active ECU will wait for a acknowledge frame as the response for acknowledgement within timeout T1.
- When the last data message, with EOM, is sent which does not request an acknowledge, the active ECU will wait for a data frame as the response for acknowledgement within timeout T1.
- The SN in an acknowledge frame is always the SN of the last received Data frame plus 1 (the next expected frames SN number).
- With every successful change in data direction, the SN of the first data message is always zero.
- After a complete data transfer, with acknowledgement, the active ECU becomes the passive ECU and waits up to T4 for the first data message to arrive.
- A Channel is terminated when the Disconnect (DC) frame is sent. This frame is not acknowledged.
- The active ECU is the only node allowed to send the Disconnect (DC) frame.

Table 12 – Static transport protocol parameters

Description	**Parameter Name**	**Default Value**
Connection Timing		
Time-out used when waiting for Acknowledge telegrams	T1	50 ms
Time-out used when waiting for Data telegrams	T2	100 ms
Minimum time allowed between sending any 2 telegrams	T3	5 ms
Maximum time after an acknowledgement before which frames must be sent	T4	1000 ms
Request / Response		
Maximum repeats of data acknowledge requests	MNT	5

Repetition Count of ‘n’ means that the applicable telegram was sent n+1 in total.

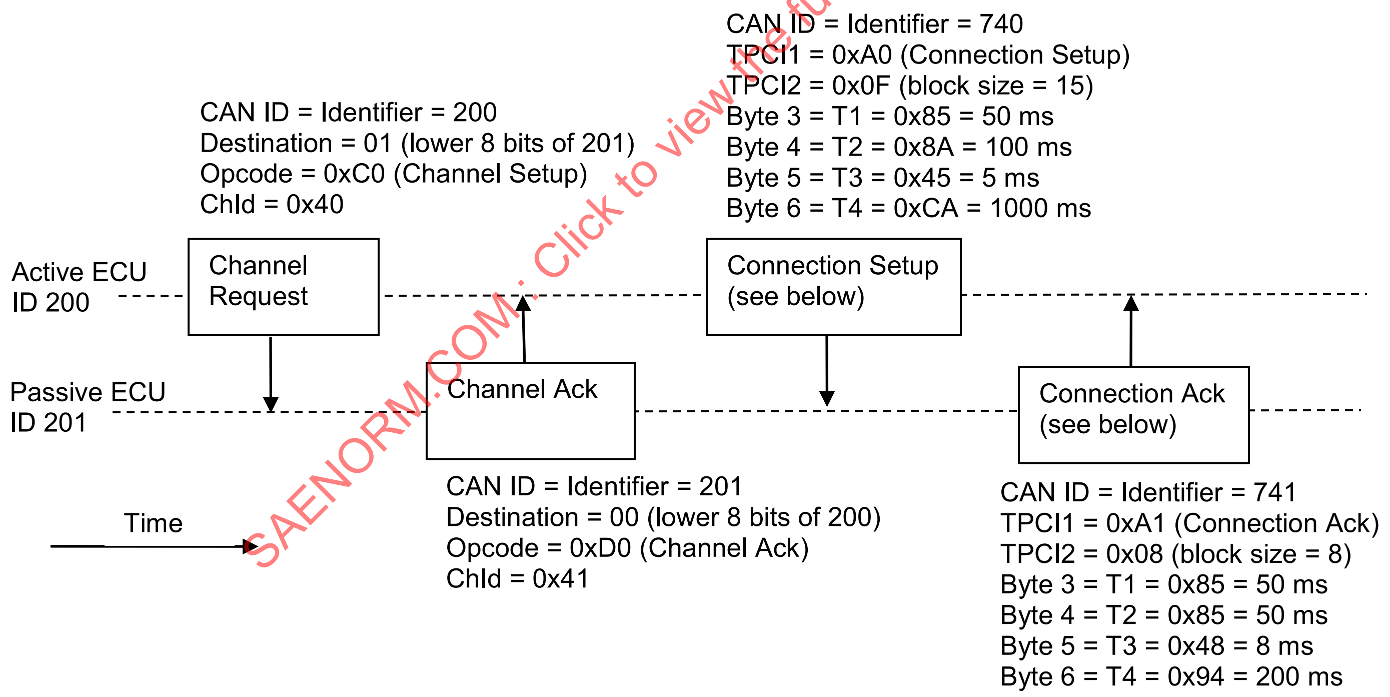
5.2.4 Transport protocol error handling

Table 13 – Transport protocol error handling

Condition	Action
Receive a TPCI byte that is invalid	Ignore this message
Passive ECU receives a Data telegram with an unexpected SN	Send Ack with the SN that is expected.
Active ECU time-out on T1 when expecting a Channel Ack response	Repeat the last Channel Setup Telegram for up to MNTC times then error by closing the channel (send Disconnect telegram)
Active ECU receives Ack with request for previous SN telegram	Send the asked for Data Telegram up to MNT times then error by closing the channel (send Disconnect telegram).
Active ECU time-out on T2 when expecting a Ack response	Repeat the last Data Telegram for up to MNT times then error by closing the channel (send Disconnect telegram)

5.2.5 CAN-Telegram Establishing a Channel and Connection

Below is an example of how a channel and connection are established.
.

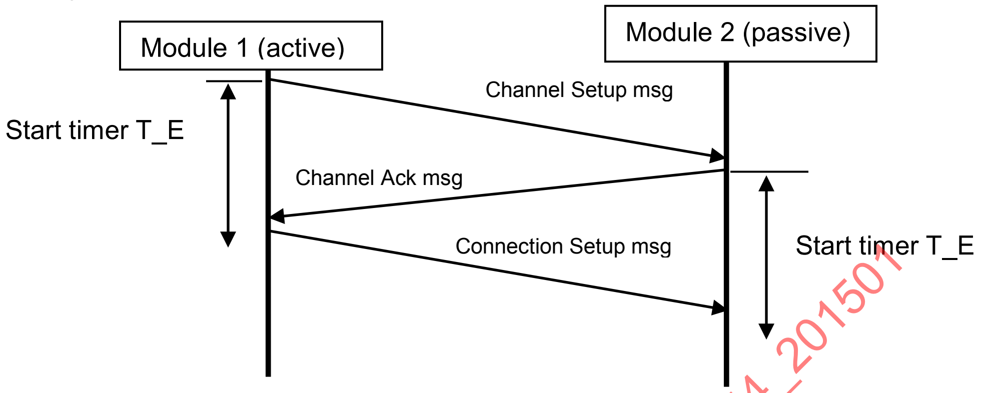

CAN ID = Identifier = 740
TPCI1 = 0xA0 (Connection Setup)
TPCI2 = 0x0F (block size = 15)
Byte 3 = T1 = 0x85 = 50 ms
Byte 4 = T2 = 0x8A = 100 ms
Byte 5 = T3 = 0x45 = 5 ms
Byte 6 = T4 = 0xCA = 1000 ms
CAN ID = Identifier = 200
Destination = 01 (lower 8 bits of 201)
Opcode = 0xC0 (Channel Setup)
ChId = 0x40
Active ECU
ID 200
Channel Request
Connection Setup (see below)
Passive ECU
ID 201
Channel Ack
Connection Ack (see below)
Time
CAN ID = Identifier = 201
Destination = 00 (lower 8 bits of 200)
Opcode = 0xD0 (Channel Ack)
ChId = 0x41
CAN ID = Identifier = 741
TPCI1 = 0xA1 (Connection Ack)
TPCI2 = 0x08 (block size = 8)
Byte 3 = T1 = 0x85 = 50 ms
Byte 4 = T2 = 0x85 = 50 ms
Byte 5 = T3 = 0x48 = 8 ms
Byte 6 = T4 = 0x94 = 200 ms

The active ECU sends a Channel Setup Request message to the passive ECU. The passive ECU responds with a Channel Acknowledge message. After this exchange, both the active and passive ECUs have the channel CAN IDs which will be used by the new connection (0x740 and 0x741). A connection can now be established using the Connection Setup and Connection Acknowledge messages.

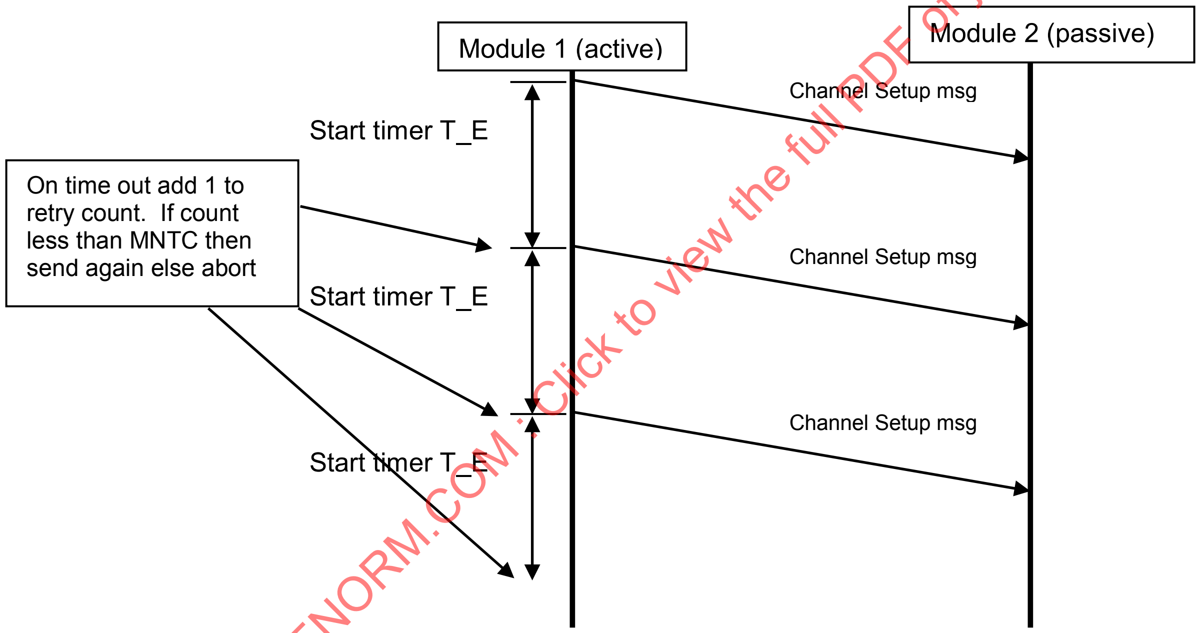
6. EXAMPLES:

6.1 CAN-Telegram Examples

6.1.1 Channel set-up with Ack

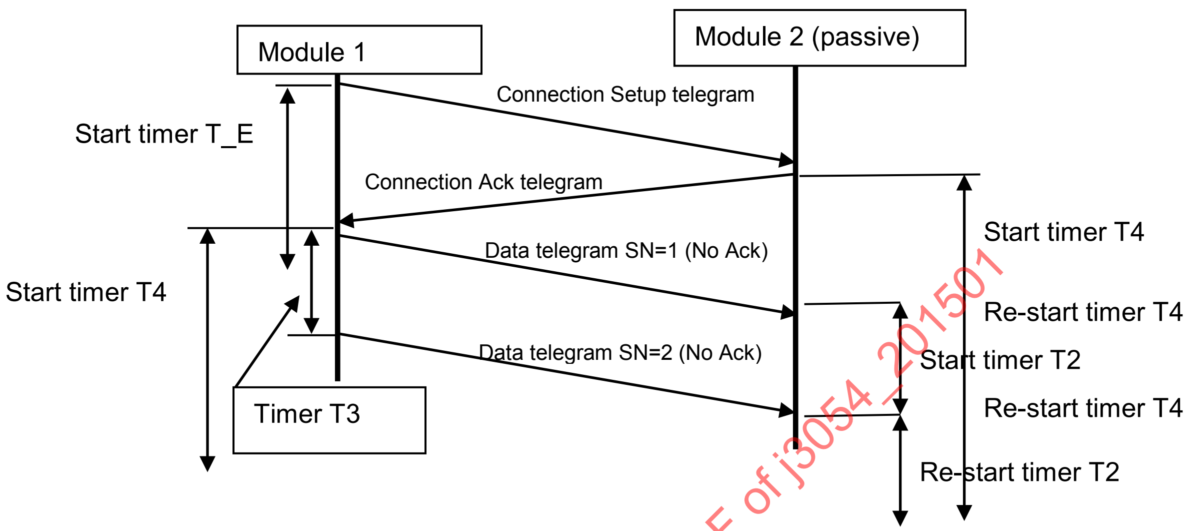


6.1.2 Channel set-up missing Ack

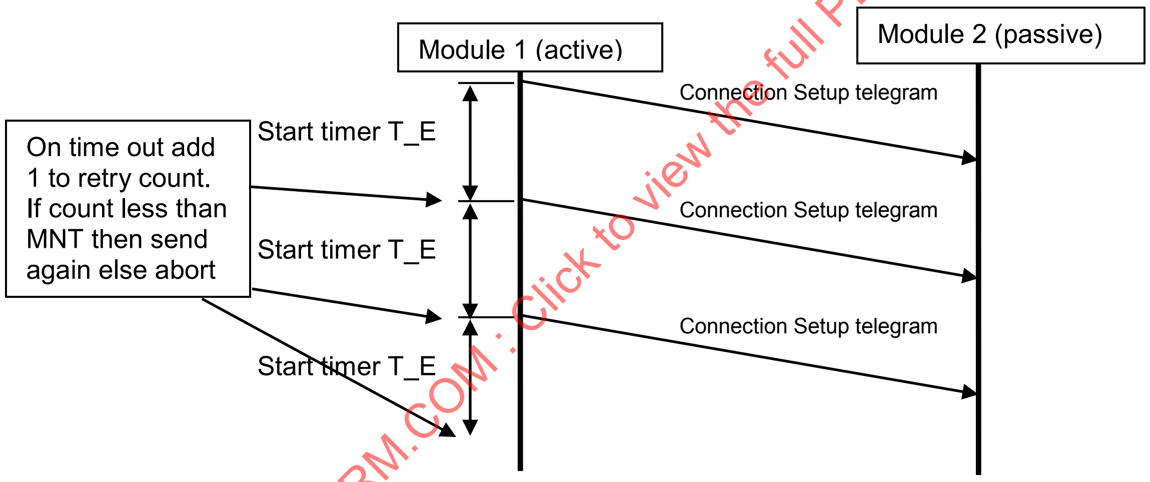


6.2 Transport Protocol Examples

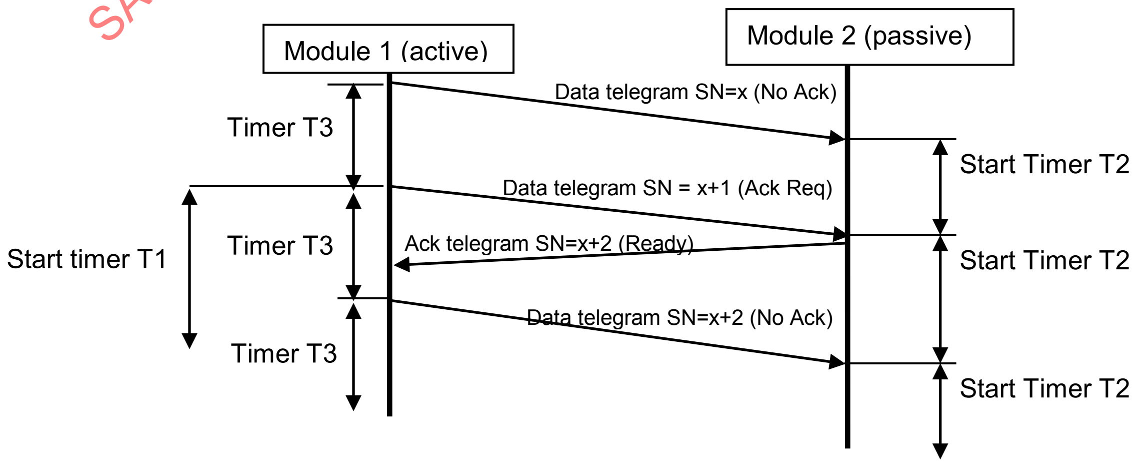
6.2.1 Connection set-up with Ack



6.2.2 Connection Set-up missing Ack


Module 1 (active)
Module 2 (passive)
Connection Setup telegram
On time out add 1 to retry count. If count less than MNT then send again else abort
Start timer T_E
Connection Setup telegram
Start timer T_E
Connection Setup telegram
Start timer T_E

6.2.3 Sending data with acknowledge request - ready response


Module 1 (active)
Module 2 (passive)
Data telegram SN=x (No Ack)
Timer T3
Start Timer T2
Data telegram SN = x+1 (Ack Req)
Start timer T1
Timer T3
Ack telegram SN=x+2 (Ready)
Start Timer T2
Data telegram SN=x+2 (No Ack)
Timer T3
Start Timer T2