



Research & Vehicle Technology
“Infotainment Systems Product Development”

A2B Command and Control API
Specification

Infotainment Subsystem Part Specific
Specification (SPSS)

Version: Revision 1.2.1
UNCONTROLLED COPY IF PRINTED

Version Date: 2/22/22

FORD CONFIDENTIAL



Revision History

Date	Ver	Notes	
June 24, 2021	0.1	Initial DRAFT Release	
August 5, 2021	0.2	Updated DRAFT Release	Modifications as follows: <ol style="list-style-type: none">1. Standardized message format to four bytes2. Added Mailbox definition3. Added Mailbox message sequence diagram and pseudocode example from ADI4. Added Message Send Type to all APIs defined5. Corrected errors in sending direction6. Updated all APIs for standard message size in order to accommodate.
November 4, 2021	1.0	Release 1.0	Modifications as follows: <ol style="list-style-type: none">1. Added new messages for Amplifier Diagnostic Information Request, Enable/Disable Frequency Hopping, Amplifier Direct Mute, Speaker Fault Status, Clip Detect Enable2. Corrected message send type for Amplifier Enable State Message3. Corrected data direction for A2B Stream Report Status4. Modified Amplifier State of Health message to combine NVM Error state, add Thermal Shutdown and create general Speaker 1-4 Fault Warnings5. Added D425 in combination with D245 in all sections.6. Removed DSP AMP from Frequency Hopping Message7. Removed ShortToExternalVoltageDetected as redundant in SOH Message8. Added FreqHoppingEnabled and ClipDetectEnabled status to SOH Message9. Added DID and DTC Mapping to CAC Message table
December 8, 2021	1.1	Release 1.1	Modifications as follows: <ol style="list-style-type: none">1. Modified AmpDiagStatus Command to allow for Amplifier Thermal Warning Status
February 21, 2022	1.2	Release 1.2	Modifications as follows: <ol style="list-style-type: none">1. Modified Error and Response Handling requirements2. Added section for message Flow Control for ECU Serial Number and ECU Part Number3. Modification to Read ECU Serial Number for Flow Control and CRC-16 modification4. Modification to Read ECU Part Number for Flow Control and CRC-16 modification
February 22, 2022	1.2.1	Release 1.2.1	Modifications as follows: <ol style="list-style-type: none">1. Addition of Flow Control Frame Section2. Modification to correct message format and description for Read ECU Part Number, Read ECU Serial Number, Message Format



Table of Contents

REVISION HISTORY	2
1.1 Scope	4
1.2 Abbreviations and Definitions	4
1.3 Protocol Overview	5
1.3.1 SWR-REQ-334780/D-Protocol Overview	5
1.3.2 SWR-REQ-435456/A-A2B Mailbox Communication Sequence	6
1.3.3 SWR-REQ-435457/A-A2B Broadcast Message Description	8
1.3.4 SWR-REQ-483777/B-Flow Control Message Definition	9
1.3.5 Message Format	10
1.3.5.1 SWR-REQ-335597/D-Message Format	10
1.3.6 Error and Response Handling	11
1.3.6.1 SWR-REQ-335602/C-Interface Module Handling of Response	11
1.3.6.2 SWR-REQ-425417/A-Error Handling	11
1.4 Messages	12
1.4.1 Commands Overview	12
1.4.2 Amplifier Enable State	13
1.4.2.1 SWR-REQ-371060/B-Amplifier Enable State Message Definition	13
1.4.3 A2B Broadcast State	14
1.4.3.1 SWR-REQ-387217/B-A2B Broadcast State Message Definition	14
1.4.4 A2B Node ID Report Out	15
1.4.4.1 SWR-REQ-387223/B-A2B Node ID Report Out Message Definition	15
1.4.5 AMP Speaker Output Status	16
1.4.5.1 SWR-REQ-387224/B-AMP Speaker Output Status Message Definition	16
1.4.6 A2B Stream Report Status	17
1.4.6.1 SWR-REQ-387225/B-A2B Stream Report Status Message Definition	17
1.4.7 Amplifier State of Health	18
1.4.7.1 SWR-REQ-387226/C-Amplifier State of Health Message Definition	18
1.4.8 Amplifier Diagnostic Info Request	19
1.4.8.1 SWR-REQ-456117/A-Amplifier Diagnostic Info Request Message Definition	19
1.4.9 Enable/Disable Frequency Hopping	20
1.4.9.1 SWR-REQ-456118/A-Enable/Disable Frequency Hopping Message Definition	20
1.4.10 Amplifier Directed Mute	21
1.4.10.1 SWR-REQ-456119/A-Amplifier Directed Mute Message Definition	21
1.4.11 Read ECU Part Number	22
1.4.11.1 SWR-REQ-456120/C-Read ECU Part Number Message Definition	22
1.4.12 Read ECU Serial Number	23
1.4.12.1 SWR-REQ-456121/C-Read ECU Serial Number Message Definition	23
1.4.13 Speaker Fault Status	24
1.4.13.1 SWR-REQ-456137/A-Speaker Fault Status Message Definition	24
1.4.14 Clip Detect Enable	25
1.4.14.1 SWR-REQ-456147/A-Clip Detect Enable Message Definition	25
1.4.15 Flow Control Frame for Multi-Frame Response	26
1.4.15.1 SWR-REQ-483891/A-Flow Control Frame Message Format	26



1.1 Scope

This specification describes the command and control for communication between the PAC and external peripherals communicating on the I2C back channel over A2B.

The scope of this specification will be limited to protocol description, API definition for messages supported, and general description of communication flow.

[Note: For detailed information on message usage in the system, please refer to the Global A2B SPSS. For detailed information on hardware setup and software driver requirements, please refer to ADI documentation.]

1.2 Abbreviations and Definitions

Abbreviation / Definition	Description
AM	Amplitude Modulation
AMP	Amplifier
ASCII	American Standard Code for Information Interchange
CAC	Command and Control
CRC	Cyclic Redundancy Check
dB	DeciBel
EOL	End of Line
EU	Europe
Fct	Function
FM	Frequency Modulation
GND	Ground
HW	Hardware
Hz	Hertz
ID	Identifier
imsbf	(signed) integer most significant bit first
JP	Japan
MCU	Micro Computer Unit
MSB	Most significant bit
PAC	Phoenix Audio Controller
PDC	Phoenix Domain Controller
ROW	Rest Of World
SW	Software or Switch

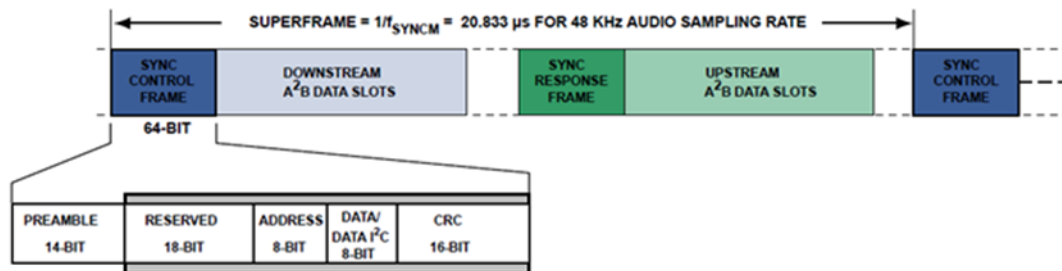


uimsbf	unsigned integer most significant bit first
US	United States (of America)

1.3 Protocol Overview

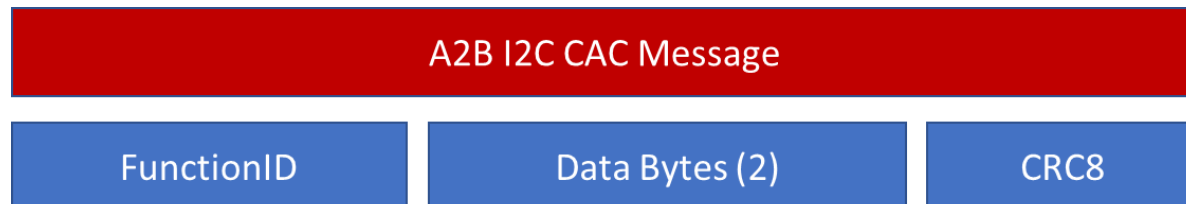
1.3.1 SWR-REQ-334780/D-Protocol Overview

All peripherals shall comply with the following message protocol and utilize A2B I2C mailbox communication (*Note: See page 58 of AD243x TRM for super-frame definition*).



4-3: SCF

The A2B I2C Command and Control (CAC) Message shall consist of a 1-byte Function ID, two bytes of subsequent Payload Data, and an 8-bit CRC for messages that do not require flow control.



Using the mailbox method, a setup would look like the below with an example for Mailbox 0 for a message that does not require flow control:

Mailbox Identifier	Data Definition
MBOX0B0	FunctionID
MBOX0B1	Data Byte 1
MBOX0B2	Data Byte 2
MBOX0B3	CRC-8

For messages that do require flow control, a TransactionID, FrameCtr, TotalFrames, and a 16-bit CRC are added (e.g. messages that cannot be sent and received within one full mailbox action).

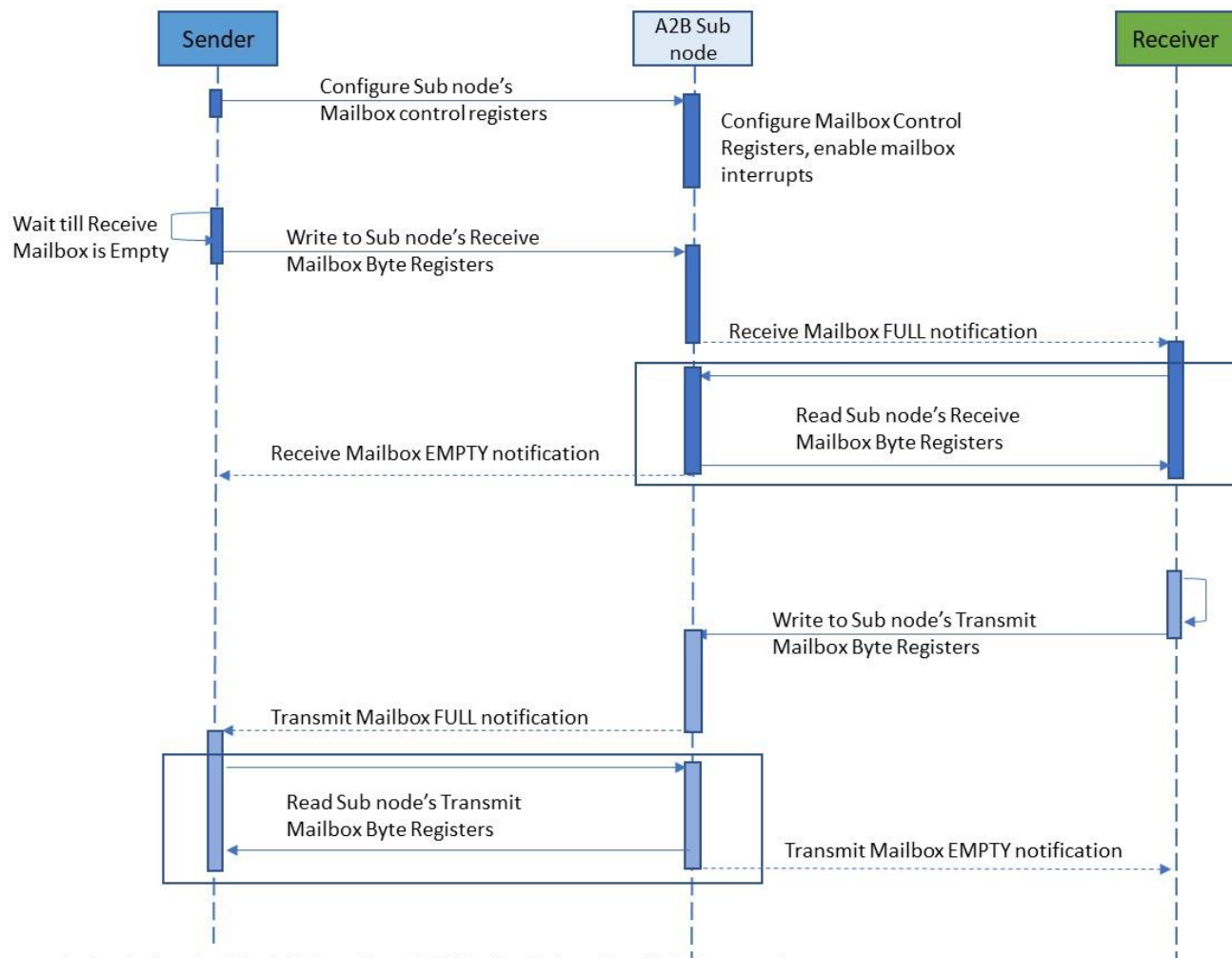
An example for the multi-frame response is shown below:

**A2B I2C CAC Message**

FunctionID	TransactionID	FrameCtr	TotalFrames	Response Frame 1
TransactionID	FrameCtr +1	TotalFrames	Data Byte (1)	Response Frame 2
TransactionID	FrameCtr + 2	TotalFrames	Data Byte (2)	Response Frame 3
TransactionID	FrameCtr + n	TotalFrames	Data Byte (n)	Response Frame n

1.3.2 SWR-REQ-435456/A-A2B Mailbox Communication Sequence

The A2B mailbox communication sequence shall be as follows:



Analog Devices Confidential Information. ©2021 Analog Devices, Inc. All rights reserved.

**Example pseudocode:**

```
//Mailbox communication - between Master and a Slave node
-----
MASTER SEND:
-----
//MBOXxCTL.MBxFIEN = 1 and MBOXxCTL.MBxEIEN = 1
FIRST TIME: //As there is no interrupt indication at the beginning
if (MBOXxSTAT.MBxEMPTY == 1)
{
    SEND MBOXxB0-(LEN-1) Bytes //all bytes written sequentially
}

NEXT TIME:
if (INTSTAT == 1)
{
    READ INTTYPE and INTSRC
    if (INTTYPE == MBOXx_EMPTY) && (INTSRC.INODE == SlaveNum)
    {
        SEND MBOXxB0-(LEN-1) Bytes //all bytes written sequentially
    }
}

-----
MASTER RECEIVE:
-----
//WHEN MBOXyCTL.MByFIEN = 1 and MBOXyCTL.MByEIEN = 1
if (INTSTAT.IRQ == 1)
{
    READ INTTYPE and INTSRC
    if (INTTYPE == MBOXy_FULL) && (INTSRC.INODE == SlaveNum)
    {
        READ MBOXyB0-(LEN-1) Bytes //all bytes read sequentially
    }
}

-----
SLAVE RECEIVE:
-----
// WHEN MBOXxCTL.MBxFIEN = 1 and MBOXxCTL.MBxEIEN = 1
if (MBOXxSTAT.MBxFULL == 1)
{
    if (LINTTYPE == MBOXx_FULL)
    {
        READ MBOXxB0-(LEN-1) Bytes //all bytes read sequentially
    }
}

-----
SLAVE SEND:
-----
//WHEN MBOXyCTL.MByFIEN = 1 and MBOXyCTL.MByEIEN = 1
FIRST TIME: //As there is no interrupt indication at the beginning
if (MBOXySTAT.MByEMPTY == 1)
{
    SEND MBOXyB0-(LEN-1) Bytes //all bytes written sequentially
}

NEXT TIME:
If (MBOXySTAT.MByEMPTY == 1)
{
    if (LINTTYPE == MBOXy_EMPTY)
    {
        SEND MBOXyB0-(LEN-1) Bytes //all bytes written sequentially
    }
}
```



1.3.3 SWR-REQ-435457/A-A2B Broadcast Message Description

To send a A2B broadcast message to all sub-nodes, the Main Node (sender) shall issue a broadcast write where the same message is sent to all A2B sub-nodes in the system.

The node closest to the Main Node shall generate the mailbox full interrupt first followed by the next node (e.g. Sub-node 0 shall generate the mailbox full interrupt first followed by Sub-node 1, Sub-node 2 and so on (assuming they all have mailbox configured and mailbox interrupts enabled)).

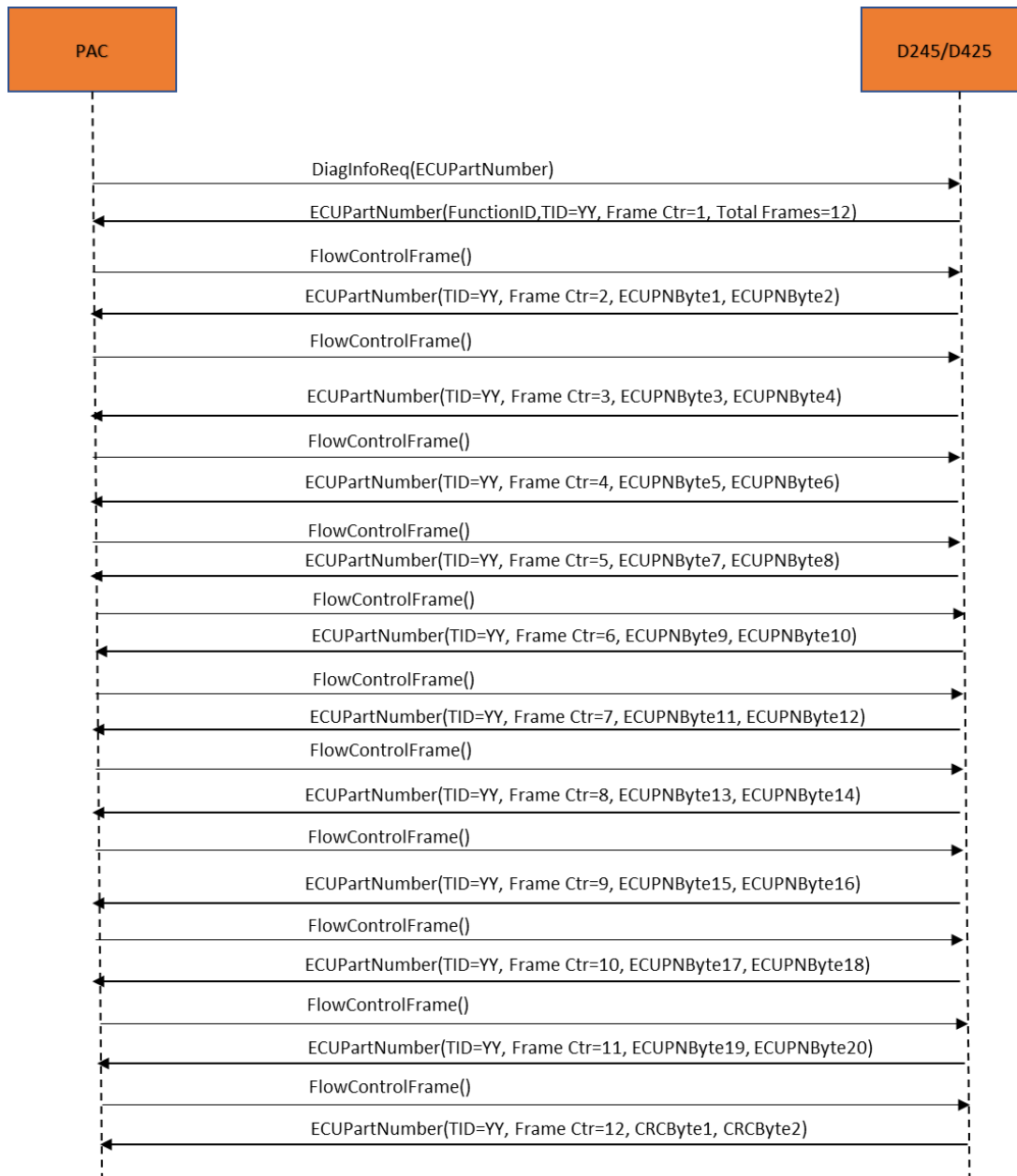


1.3.4 SWR-REQ-483777/B-Flow Control Message Definition

Message sequences that can not be accomplished within one mailbox read or write action shall require flow control.

An example is as shown below where special consideration is given to the flow control messages where:

- TransactionID is required
- Frame Counter is required
- Total Number of Frames is required
- Extension of CRC to CRC16 is required





1.3.5 Message Format

1.3.5.1 SWR-REQ-335597/D-Message Format

All peripherals shall support the common datagram representation as follows for messages that are not multi-frame:

Syntax	Nr of bits	
Message() { FunctionID for (i=0; i<N; i++) { PayloadDataByte1 PayloadDataByte2 } CRC8 }	8 8 8 8	uimsbf uimsbf uimsbf uimsbf

Name	Byte	Range	Description
FunctionID	1	Full range	Specifies the function
PayloadDataByte1	1	Full range	Payload Data
PayloadDataByte2	1	Full range	Payload Data
CRC8	1	Full range	1-byte CRC following polynomial: X^8+x^2+x+1

All peripherals shall support the common datagram representation as follows for multi-frame messages requiring flow control:

Syntax	Nr of bits	
MessageMultiFrame() { FunctionID TID FrameCtr TotalFrames for (i=0; i<N; i++) { PayloadDataByte1 PayloadDataByte2 PayloadDataByte(N) } CRC16 }	8 8 8 8 8 8 8 8 16	uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf

Name	Byte	Range	Description
FunctionID	1	Full range	Specifies the function



TID	1	0x1-0xFF	Specifies TransactionID for the multi-frame response as a means of identification to keep all frames of the multi-frame response together
FrameCtr	1	0x1-0xFF	Specifies current frame
TotalFrames		0x1-0xFF	Specifies total number of frames in message to be received (including CRC16 and preceding bytes)
PayloadDataByte1	1	Full range	Payload Data
PayloadDataByte2	1	Full range	Payload Data
PayloadDataByten	1	Full range	End of Payload Data
CRC16	1	Full range	2-byte CRC following polynomial: $X^{16}+x^{12}+x^5+1$

1.3.6 Error and Response Handling

1.3.6.1 SWR-REQ-335602/C-Interface Module Handling of Response

The PAC shall induce a retry strategy where it shall retry up to three times before determining that an A2B Communication Error exists IF:

- A message that is NOT a multiframe message AND a response is NOT received within 1000msec OR;
- A message is a multiframe message AND a response is NOT received within 50msec.

[Note: Multiframe messages are those which require Flow Control]

1.3.6.2 SWR-REQ-425417/A-Error Handling

A cyclic redundancy check (CRC) is done at the hardware level on each data byte. If the CRC doesn't match the data, the data is flagged with an error (i.e. CRC ERROR) and is discarded.

Any peripheral on the A2B bus which does not receive an entire message shall consider this an error and take no further action.



1.4 Messages

1.4.1 Commands Overview

The following command set is defined for control and diagnostic status between the A2B Main Node and Sub-Nodes for modules that communicate using the A2B CAC Protocol [Note: Mapping to the IDS Specification for DID or DTC is given for implementation]:

Function ID	Command	DID/DTC Mapping
0x00	Not used	N/A
0x01	Amplifier Enable State	DID FD52
0x02	A2B Broadcast State	DID FD53
0x03	A2B Node ID Report Out	DID FD54
0x04	AMP Speaker Output Status	DID FD55
0x05	A2B Stream Content Status	DID FD57
0x06	Amplifier State of Health	DID FD58
0x07	Amplifier Diagnostic Information Request	N/A
0x08	Enable/Disable Frequency Hopping	DID FD59
0x09	Amplifier Direct Mute	DID FD5B
0x0A	Read ECU Part Number	DID F129
0x0B	Read ECU Serial Number	DID F0E8
0x0C	Speaker Fault Status	DTC 9238xx
0x0D	Clip Detect Enable	DID FD5C
0x0E-0x7E	Reserved for Future Use	N/A
0x80	Message Flow Control ID	N/A
0x81-0xFF	Reserved for Future Use	N/A



1.4.2 Amplifier Enable State

1.4.2.1 SWR-REQ-371060/B-Amplifier Enable State Message Definition

The *AMPEnableSet* command shall be utilized as a directed command, based on event, from the PAC to Enable or Disable the DSP AMP Module, D425, and D245 AMP Module and to get status on the current Enable/Disable state.

Syntax Description:

Syntax	Nr of Bits	
AMPEnableSet() { FunctionID AMPEnableState ReservedByte CRC8 }	8 8 8 8	uimbsf uimbsf uimbsf uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x01	PAC to DSP AMP PAC to D245 PAC to D425	AMPEnableState	Event

Parameter Description:

Parameters	Bytes	Range	Description
AMPEnableState	1	0x00-0xFF	0x00: Disabled 0x01: Enabled 0x02-FF: Invalid/Reserved
ReservedByte	1	0x00-0xFF	Reserved Payload Data Byte



1.4.3 A2B Broadcast State

1.4.3.1 SWR-REQ-387217/B-A2B Broadcast State Message Definition

All peripherals shall support the *A2BBroadcast* command.

The PAC shall broadcast the *A2BBroadcast* command to all sub-nodes for communication of the:

- Muted state of the PAC speakers.
- Status of audio content (e.g. whether it has been sent or not).
- Periodic heartbeat for the system; OR
- Request for A2B Node ID.

Syntax Description:

Syntax	Nr of Bits	
A2BBroadcastState ()		
{		
FunctionID	8	uimbsf
A2BBroadcastState	8	uimbsf
ReservedByte	8	uimbsf
CRC8	8	uimbsf
}		

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x02	PAC to DSP AMP PAC to PDC PAC to D245 PAC to D425	A2BBroadcastState ReservedByte	Broadcast

Parameter Description:

Parameters	Bytes	Description
A2BBroadcastState	1	Bit 0: UnMuted (0x1), Muted (0x0) Bit 1: Audio Sent (0x1), Audio Not Sent (0x0) Bit 2: A2B Node Request (0x1), Default (0x0) Bits 3-7: Reserved
ReservedByte	1	Reserved Payload Data Byte



1.4.4 A2B Node ID Report Out

1.4.4.1 SWR-REQ-387223/B-A2B Node ID Report Out Message Definition

The *A2BNodeReportOut* command shall be supported by all sub-nodes to communicate the A2B Node ID to the PAC module as a response to the request of the *A2BBroadcastState* (Bit 2=1, A2B Node Request) command.

Syntax Description:

Syntax	Nr of Bits	
A2BNodeReportOut ()		
{		
FunctionID	8	uimsbf
A2BNodeID	8	uimsbf
ReservedByte	8	uimsbf
CRC8	8	uimsbf
}		

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x03	DSP AMP to PAC PDC to PAC D245 to PAC D425 to PAC	A2BNodeID ReservedByte	Event

Parameter Description:

Parameters	Bytes	Description
A2BNodeID	1	Node ID assigned to the Sub-Node
ReservedByte	1	Reserved Payload Data Byte



1.4.5 AMP Speaker Output Status

1.4.5.1 SWR-REQ-387224/B-AMP Speaker Output Status Message Definition

The *AMPSpeakerOutputStatus* shall be supported by the PAC and Amplifier modules as an event-based message. This message is sent from the AMP Module/Modules (DSP AMP or D245/D425) to give the PAC status on the Mute or Unmute of the system correlating to whether it is ready to play sounds.

Syntax Description:

Syntax	Nr of Bits	
AMPSpeakerOutputStatus() { FunctionID SpeakerStatus ReservedByte CRC8 }	 8 8 8 8	 uimbsf uimbsf uimbsf uimbsf

Command Description:

Function ID	Data Direction	Parameters	Message Send Type
0x04	DSP AMP to PAC D245 to PAC D425 to PAC	SpeakerStatus ReservedByte	Event

Parameter Description:

Parameters	Bytes	Description
SpeakerStatus	1	0x00: Default (Muted) 0x01: UnMuted 0x02-0xFF: Invalid/Reserved
ReservedByte	1	Reserved Payload Data Byte



1.4.6 A2B Stream Report Status

1.4.6.1 SWR-REQ-387225/B-A2B Stream Report Status Message Definition

The *A2BStreamReportStatus* message shall be supported by both the PDC and the PAC. This message is an event-based status command from the PDC to the PAC to communicate audio status.

Syntax Description:

Syntax	Nr of Bits	
A2BStreamReportStatus() { FunctionID AudioStatus ReservedByte CRC8 }	 8 8 8 8	 uimbsf uimbsf uimbsf uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x05	PDC to PAC	AudioStatus ReservedByte	Event

Parameter Description:

Parameters	Bytes	Description
AudioStatus	1	0x00: AudioNotSent 0x01: AudioSent 0x02-FF: Invalid/Reserved
ReservedByte	1	Reserved Payload Data Byte



1.4.7 Amplifier State of Health

1.4.7.1 SWR-REQ-387226/C-Amplifier State of Health Message Definition

The *AmplifierSOH* command shall be supported by the D245/D425 Amplifiers and the PAC. This message shall communicate diagnostic status (e.g. state of health) to the PAC on an event-periodic basis for purposes of logging amplifier DTCs and taking fault or functional action accordingly. The message is bit encoded where bit=1 indicates that the condition is ACTIVE and bit=0 indicate that the condition is INACTIVE.

Syntax Description:

Syntax	Nr of Bits	
AmplifierSOH ()		
{		
FunctionID	8	uimbsf
AmpDiagStatus	16	uimbsf
CRC-8	8	uimbsf
}		

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x06	D245 to PAC D425 to PAC DSP AMP to PAC (Optional)	AmpDiagStatus	Event-Periodic (5 sec period)

Parameter Description:

Parameters	Byte #	Description
AmpDiagStatus	Byte 1	Bit 0: OverTemperatureProtectionActive Bit 1: OverVoltageProtectionActive Bit 2: NVMEErrorActive Bit 3: MPUPeripheralErrorDetected Bit 4: WatchdogTimerResetDetected Bit 5: UnderVoltageProtectionActive Bit 6: FreqHoppingEnabled Bit 7: ClipDetectEnabled
	Byte 2	Bit 0: Speaker1FltActive Bit 1: Speaker2FltActive Bit 2: Speaker3FltActive Bit 3: Speaker4FltActive Bit 4: ThermalWarningActive Bit 5: Reserved Bit 6: Reserved Bit 7: Reserved



1.4.8 Amplifier Diagnostic Info Request

1.4.8.1 SWR-REQ-456117/A-Amplifier Diagnostic Info Request Message Definition

The *AmpDiagInfoReq* message shall be supported by both the PAC and the amplifier modules (D245/D425). This message is an event-based request command from the PAC to the amplifiers to request specific diagnostic information.

Note: As D245 amplifier will only support two channels, Speaker 3 and Speaker 4 fault requests are not supported.

Syntax Description:

Syntax	Nr of Bits	
AmpDiagInfoReq() { FunctionID DiagInfoReq ReservedByte CRC8 }	8 8 8 8	uimbsf uimbsf uimbsf uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x07	PAC to D245 PAC to D425	DiagInfoReq ReservedByte	Event

Parameter Description:

Parameters	Bytes	Description	Notes
DiagInfoReq	1	0x00: Invalid/Null 0x01: ECUPartNumber 0x02: ECUSerialNumber 0x03: Speaker1Faults 0x04: Speaker2Faults 0x05: Speaker3Faults (not supported for D245) 0x06: Speaker4Faults (not supported for D245) 0x07-0xFF: Invalid/Reserved	The DiagInfoReq byte is meant to handle one request at a time. It is not possible to request all diagnostic information at the same time or in combination.
ReservedByte	1	Reserved Payload Data Byte	



1.4.9 Enable/Disable Frequency Hopping

1.4.9.1 SWR-REQ-456118/A-Enable/Disable Frequency Hopping Message Definition

The *FrequencyHopSet* command shall be utilized as a direct event command from the PAC to Enable or Disable frequency hopping in the D245/D425 AMP modules and if enabled, the frequency to be used shall be given.

Syntax Description:

Syntax	Nr of Bits	
FreqHopSet() { FunctionID FreqHopState AssignedFreq CRC8 }	 8 8 8 8	 uimbsf uimbsf uimbsf uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x08	PAC to D245 PAC to D425	FreqHopState AssignedFreq	Event Message

Parameter Description:

Parameters	Bytes	Range	Description
FreqHopState	1	0x00-0xFF	0x00: Disabled 0x01: Enabled 0x02-FF: Invalid/Reserved
AssignedFreq	1	0x00-0xFF	0x00: 8 x fS (352.8 kHz/384 kHz) 0x01: 10 x fS (441 kHz/480 kHz) 0x02: Reserved 0x03: Reserved 0x04: Reserved 0x05: 38 x fS (1.68 MHz/1.82 MHz) 0x06: 44 x fS (1.94 MHz/2.11 MHz) 0x07: 48 x fS (2.12 MHz/ not supported) 0x08-0xFE: Reserved 0xFF: IF FreqHopState=Disabled



1.4.10 Amplifier Directed Mute

1.4.10.1 SWR-REQ-456119/A-Amplifier Directed Mute Message Definition

The *AMPDirectMute* message shall be supported by both the PAC, the DSP AMP, D425 and the D245 AMP modules. This message is an event-based status command from the PAC to the amplifiers to directly mute or unmute.

Syntax Description:

Syntax	Nr of Bits	
AMPDirectMute() { FunctionID DirectMute ReservedByte CRC8 }	 8 8 8 8	 uimbsf uimbsf uimbsf uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x09	PAC to DSP AMP PAC to D245 PAC to D245	DirectMute ReservedByte	Event

Parameter Description:

Parameters	Bytes	Description
DirectMute	1	0x00: AudioMute 0x01: AudioUnmute 0x02-FF: Invalid/Reserved
ReservedByte	1	Reserved Payload Data Byte





ECUPNByte[n]	1-20	ECU Part Number (20-Bytes) ASCII
--------------	------	----------------------------------

1.4.12 Read ECU Serial Number

1.4.12.1 SWR-REQ-456121/C-Read ECU Serial Number Message Definition

The *ECUSerialNumber* message shall be supported by both the PAC and D245/D425 modules. This message is an event-based command from the PAC from the D245/D425 module to communicate the ECU Serial Number based on a *DiagInfoRequest* where it is requested.

[Note: This message is a multi-frame response which requires the FlowControlID]

Syntax Description:

Syntax	Nr of Bits	
ECUSerialNumber() { FunctionID TID FrameCtr TotalFrames ECUSNByte1 ECUSNByte2 ECUSNByte3 ECUSNByte4 ECUSNByte5 ECUSNByte6 ECUSNByte7 ECUSNByte8 ECUSNByte9 ECUSNByte10 ECUSNByte11 ECUSNByte12 ECUSNByte13 ECUSNByte14 ECUSNByte15 ECUSNByte16 CRC16 }	 8 16	 uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x0B	D245 to PAC D425 to PAC	ECUSNByte1-16	Event

Parameter Description:

Parameters	Bytes	Description
TID	1	TransactionID for multi-frame response
FrameCtr	1	Count of current frame of multi-frame response
TotalFrames	1	Total count of all frames expected in multi-frame response
ECUSNByte[n]	1-16	ECU Serial Number (16-Bytes) ASCII



1.4.13 Speaker Fault Status

1.4.13.1 SWR-REQ-456137/A-Speaker Fault Status Message Definition

The *SpeakerFltStatus* command shall be supported by the PAC and the amplifier (D245/D425). This message shall be in response to a DiagInfoRqst where a specific speaker fault status is requested AND upon qualification of a speaker fault. This message is sent for a single speaker.

Syntax Description:

Syntax	Nr of Bits	
SpeakerFltStatus ()		
{		
FunctionID	8	uimbsf
SpeakerID	8	uimbsf
SpeakerStatus	8	uimbsf
CRC-8	8	uimbsf
}		

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x0C	D245 to PAC D425 to PAC	SpeakerID SpeakerStatus	Event

Parameter Description:

Parameters	Byte #	Description
SpeakerID	Byte 1	Bit 0: Speaker1 Bit 1: Speaker2 Bit 2: Speaker3 (not used for D245) Bit 3: Speaker4 (not used for D245) Bit 4-7: Reserved
SpeakerStatus	Byte 2	Bit 0: ShortAcrossLoadDetected Bit 1: ShortToSupplyDetected Bit 2: ShortToGroundDetected Bit 3: CurrentLimitingActive Bit 4: OutputDCOffsetDetected Bit 5: OpenCircuitDetected Bit 6: Reserved Bit 7: Reserved



1.4.14 Clip Detect Enable

1.4.14.1 SWR-REQ-456147/A-Clip Detect Enable Message Definition

The *ClipDetectSet* command shall be utilized as a direct event command from the PAC to Enable or Disable clip detect in the amplifier (D245/D425 module) and if enabled, one of the thirteen settings specified shall be utilized. These thirteen settings are defined by the EQ settings stored within the PAC.

Syntax Description:

Syntax	Nr of Bits	
FreqHopSet() { FunctionID ClipDetectState ClipDetectSetting CRC8 }	 8 8 8 8	 uimbsf uimbsf uimbsf uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x0D	PAC to D245 PAC to D425	ClipDetectState ClipDetectSetting	Event

Parameter Description:

Parameters	Bytes	Range	Description
ClipDetectState	1	0x00-0xFF	0x00: Disabled 0x01: Enabled 0x02-FF: Invalid/Reserved
ClipDetectSetting	1	0x00-0xFF	0x00: Invalid 0x01-0x0D: Setting1-Setting13 0x0E-0xFE: Reserved 0xFF: IF ClipDetectState=Disabled



1.4.15 Flow Control Frame for Multi-Frame Response

1.4.15.1 SWR-REQ-483891/A-Flow Control Frame Message Format

The *FlowCtrlFrame* command shall be utilized as a method for flow control for multi-frame responses from the D245/D425 to the PAC.

Syntax Description:

Syntax	Nr of Bits	
FlowCtrlFrame() { FunctionID FunctionIDResp TID CRC8 }	8 8 8 8	uimbsf uimbsf uimbsf uimbsf

Command Description:

FunctionID	Data Direction	Parameters	Message Send Type
0x80	PAC to D245 PAC to D425	FunctionIDResp TID	Event

Parameter Description:

Parameters	Bytes	Range	Description
FunctionIDResp	1	0x0A OR 0x0B	FunctionID of the multi-frame response that the Flow Control is for.
TID	1	0x00-0xFF	TransactionID of the multi-frame response that the Flow Control is for.