Research & Vehicle Technology
"Product Development"

Displaying Module Interface

Subsystem Part Specific Specification (SPSS)

Version 1.1

UNCONTROLLED COPY IF PRINTED

Version Date: August 26, 2021

FORD CONFIDENTIAL

Revision History

Date	Version	Notes	
June 30, 2021	**1.0**	**Initial release**	
August 26, 2021	**1.1**	**Updated Release**	
		IFS-MMI2C-REQ-421663/B-Multi-finger tracking	**<hzubert> removed 10 fingers preferred**
		IFS-MMI2C-REQ-422315/B-Touch Report Rate	**<hzubert> added 120 Hz acceptable for In cell touch displays**
		IFS-MMI2C-TMR-REQ-422302/B-t_BL_FADE	**<hzubert> changed value to 750 +- 50 ms**
		IFS-MMI2C-TMR-REQ-422302/B-t_BL_FADE	**<hzubert> changed value to 750 +- 50 ms**
		IFS-MMI2C-SR-REQ-197885/C-Temperature Derating	**<hzubert> clarification**
		STR-307924/C-Interrupt	**<hzubert> added new requirement: Fault Detection Timing**
		IFS-MMI2C-SR-REQ-436478/A-Fault Detection Timing	**<hzubert> initial release**
		IFS-MMI2C-SR-REQ-140561/E-Read from Subaddress	**<hzubert> corrrected wronly NACKs and ACKs**
		IFS-MMI2C-SR-REQ-197857/D-Write to Read-Only Subaddress	**<hzubert> corrrected wronly NACKs and ACKs**
		IFS-MMI2C-SR-REQ-140569/E-Write Underflow	**<hzubert> corrrected wronly NACKs and ACKs**
		IFS-MMI2C-SR-REQ-140570/E-Write Overflow	**<hzubert> corrrected wronly NACKs and ACKs**
		IFS-MMI2C-SR-REQ-197875/C-Read from Subaddress Beyond Defined Length+	**<hzubert> added example for functional safety support**
		IFS-MMI2C-SR-REQ-197875/D-Read from Subaddress Beyond Defined Length	**<hzubert> readded content due to database error with version C**
		IFS-MMI2C-SR-REQ-140565/E-Undefined / Unsupported Subaddress	**<hzubert> corrrected wronly NACKs and ACKs**
		STR-207381/F-I2C Messages	**<hzubert> added 0x09 for actual PWM value**
		IFS-MMI2C-SR-REQ-140617/C-0x02 LCD Backlight PWM Value	**<hzubert> added clarification about reading value**
		IFS-MMI2C-SR-REQ-435880/A-0x09 Momentary LCD Backlight PWM Status	**<hzubert> initial release**

Table of Contents

REVISION HISTORY **2**

1 ARCHITECTURAL DESIGN **5**

2 GENERAL REQUIREMENTS **6**

2.1 IFS-MMI2C-REQ-420265/A-Power Enable line Low *6*

2.2 IFS-MMI2C-REQ-420266/A-Undervoltage *6*

2.3 IFS-MMI2C-REQ-420267/A-Backlight set to Zero *6*

2.4 Touch Requirements *6*

2.4.1 IFS-MMI2C-REQ-421663/B-Multi-finger tracking 6
2.4.2 IFS-MMI2C-REQ-422797/A-Large object - detection 6
2.4.3 IFS-MMI2C-REQ-422799/A-Large object - four fingers side-by-side 6
2.4.4 IFS-MMI2C-TMR-REQ-421679/A-Touch accuracy at middle of touch panel 6
2.4.5 IFS-MMI2C-TMR-REQ-422796/A-Touch accuracy at edge of sensing area 6
2.4.6 IFS-MMI2C-REQ-422800/A-Jitter 7
2.4.7 IFS-MMI2C-REQ-422315/B-Touch Report Rate 7
2.4.8 IFS-MMI2C-REQ-422801/A-Minimum object (finger) size 7
2.4.9 IFS-MMI2C-REQ-422802/A-Hover 7
2.4.10 IFS-MMI2C-REQ-422803/A-Finger separation distance 7
2.4.11 IFS-MMI2C-REQ-422804/A-Finger separation – multi-finger hopping and dropouts 7
2.4.12 IFS-MMI2C-REQ-420198/A-Baseline Recovery 8
2.4.13 IFS-MMI2C-REQ-421658/A-Baseline temperature drift 8
2.4.14 IFS-MMI2C-REQ-421659/A-Touch performance – low ground mass 8
2.4.15 IFS-MMI2C-REQ-422789/A-Touch performance – moisture present on touch sensor 9
2.4.16 IFS-MMI2C-REQ-422791/A-Touch performance – mist present on touch sensor 9
2.4.17 IFS-MMI2C-REQ-422792/A-Touch performance – frost present on touch sensor 9
2.4.18 IFS-MMI2C-REQ-422793/A-Touch performance – noisy display pattern 10
2.4.19 IFS-MMI2C-REQ-421660/A-Self-test 10
2.4.20 IFS-MMI2C-REQ-422805/A-Response times – active mode 10
2.4.21 IFS-MMI2C-REQ-422806/A-Response times – idle to active 10
2.4.22 IFS-MMI2C-REQ-421661/A-Glove Touch 11
2.4.23 IFS-MMI2C-REQ-425877/A-Touch Controller E2E communication RC 11
2.4.24 IFS-MMI2C-REQ-425878/A-Touch Controller E2E communication CRC 11

2.5 Display Timing Requirements *11*

2.5.1 IFS-MMI2C-REQ-425879/A-Horizontal Timing Requirements 11

2.6 Display Dimming Requirements *11*

2.6.1 IFS-MMI2C-REQ-426497/A-Display Dimming Requirement 11

3 FUNCTIONAL DEFINITION **12**

3.1 IFS-MMI2C-REQ-418083/A-Start Up for Self Powered Displays *12*

3.1.1 Requirements 12
3.1.2 Use Cases 13
3.1.3 Sequence Diagrams 14

3.2 IFS-MMI2C-REQ-419617/A-Shutdown for Self Powered Displays normal operation/reset request *15*

3.2.1 Requirements 15
3.2.2 Use Cases 15
3.2.3 Sequence Diagrams 16

3.3 IFS-MMI2C-REQ-419625/A-Shutdown for Self Powered Displays for recovery *17*

3.3.1 Requirements 17

3.3.2 Use Cases 17
3.3.3 Sequence Diagrams 18
3.4 IFS-MMI2C-REQ-419637/A-Recovery for Self Powered Displays 19
3.4.1 Requirements 19
3.4.2 Use Cases 20
3.4.3 Sequence Diagrams 21
3.5 IFS-MMI2C-REQ-419629/A-Touch Reset for Self Powered Displays 22
3.5.1 Requirements 22
3.5.2 Use Cases 22
3.5.3 Sequence Diagrams 23
3.6 REQ-403507/A-I2C over LVDS Communication Protocol 24
3.6.1 IFS-MMI2C-FUN-REQ-140540/C-I2C Interface Displays 24
3.6.2 IFS-MMI2C-FUN-REQ-140607/D-Generation 2 Display Modules 35
3.6.3 Appendix A: Definitions / Acronyms 80
3.6.4 Appendix B: Reference Documents 80

4 APPENDIX: REFERENCE DOCUMENTS 81

5 APPENDIX A: I2C SEQUENCE EXAMPLE - STARTUP 82

6 APPENDIX A: I2C SEQUENCE EXAMPLE - SHUTDOWN 1 83

7 APPENDIX A: I2C SEQUENCE EXAMPLE - SHUTDOWN 2 84

8 APPENDIX A: I2C SEQUENCE EXAMPLE - RECOVERY 85

9 APPENDIX A: I2C SEQUENCE EXAMPLE - TOUCH RESET 86

10 APPENDIX A: I2C SEQUENCE EXAMPLE - ON DEMAND BACKLIGHT UPDATE 87

11 APPENDIX A: I2C SEQUENCE EXAMPLE - INTERRUPT SERVICE ROUTINE 88

12 APPENDIX A: I2C SEQUENCE EXAMPLE - PERIODIC CHECK 89

13 APPENDIX B: RC AND CRC EXAMPLE - SUBADDR 90

14 APPENDIX B: RC AND CRC EXAMPLE - MISSING ROLLING COUNTER OR CRC 91

15 APPENDIX B: RC AND CRC EXAMPLE - ROLLING COUNTER 92

16 APPENDIX B: RC AND CRC EXAMPLE - INVALID ROLLING COUNTER 94

17 APPENDIX B: RC AND CRC EXAMPLE - INVALID CRC 95

1 Architectural Design

2 General Requirements

2.1 IFS-MMI2C-REQ-420265/A-Power Enable line Low

When the „Display Power Enable“-line is low, backlighting and LCD shall be off.

2.2 IFS-MMI2C-REQ-420266/A-Undervoltage

When the display power supply is below defined range for defined time duration, backlighting and LCD shall be off.

2.3 IFS-MMI2C-REQ-420267/A-Backlight set to Zero

When the backlight is set to zero via I2C, backlighting shall be off.

2.4 Touch Requirements

2.4.1 IFS-MMI2C-REQ-421663/B-Multi-finger tracking

Touch Requirement	Specification Limit(s)	Requirement Description
Multi-finger tracking	5 fingers required	All fingers shall be tracked.

2.4.2 IFS-MMI2C-REQ-422797/A-Large object - detection

Touch Requirement	Specification Limit(s)	Requirement Description
Large object - detection	> 30mm diameter	A large object (palm) with a diameter of > 30mm shall suppress all touches.

2.4.3 IFS-MMI2C-REQ-422799/A-Large object - four fingers side-by-side

Touch Requirement	Specification Limit(s)	Requirement Description
Large object - four fingers side-by-side	N/A	Four fingers side by side shall not invoke a large object detection.

2.4.4 IFS-MMI2C-TMR-REQ-421679/A-Touch accuracy at middle of touch panel

Name	Description	Units	Range	Resolution	Default
Touch accuracy at middle of touch panel	A single 8 mm diameter object at the middle of the touch panel shall be reported with specified accuracy.	mm	-1.0 to +1.0	0.1	0

2.4.5 IFS-MMI2C-TMR-REQ-422796/A-Touch accuracy at edge of sensing area

Name	Description	Units	Range	Resolution	Default
Touch accuracy at edge of sensing area	A single 8 mm diameter object at the edge of the sensing area shall be reported with the following specified accuracy.	mm	-2.5 to + 2.5	0.1	0

2.4.6 IFS-MMI2C-REQ-422800/A-Jitter

Touch Requirement	Specification Limit(s)	Requirement Description
Jitter	± 0.5 mm	The position of a single stationary 8 mm diameter object shall be polled 50 times. The reported finger location shall not exceed specified jitter.

2.4.7 IFS-MMI2C-REQ-422315/B-Touch Report Rate

Touch Requirement	Specification Limit(s)	Requirement Description
Touch Report Rate	100 Hz preferred 120 Hz acceptable	Display module shall report the updated positions of a single finger in motion at the specified report rate.

2.4.8 IFS-MMI2C-REQ-422801/A-Minimum object (finger) size

Touch Requirement	Specification Limit(s)	Requirement Description
Minimum object (finger) size	5mm	A finger size of a minimum diameter of 5mm shall be continously reported without drop out, including with a coverlay or film (screen protector) on touch sensor.

2.4.9 IFS-MMI2C-REQ-422802/A-Hover

Touch Requirement	Specification Limit(s)	Requirement Description
Hover	< 2 mm	Hover is the height above touch sensor where touch is detected. A finger hovering higher than specified distance shall not report finger touch event.

2.4.10 IFS-MMI2C-REQ-422803/A-Finger separation distance

Touch Requirement	Specification Limit(s)	Requirement Description
Finger separation distance	< 13 mm center to center	Two 8 mm diameter objects (fingers) shall be reported separately until separation distance is less than specified limit.

2.4.11 IFS-MMI2C-REQ-422804/A-Finger separation – multi-finger hopping and dropouts

Touch Requirement	Specification Limit(s)	Requirement Description
Finger separation – multi-finger hopping and dropouts	N/A	Two, three and four fingers dragged side by side shall be reported with minimal index hopping and dropouts.

2.4.12 IFS-MMI2C-REQ-420198/A-Baseline Recovery

Touch Requirement	Specification Limit(s)	Requirement Description
Baseline recovery	N/A	Display module shall automatically recover from all forms of baseline corruption. Examples: • moisture • fingers/palms covering all or part of the sensor Baseline shall recover once all objects are removed from the sensor. Touch events shall be reported correctly across entire sensor area within 3 seconds of objects being removed. Examples of baseline corruption: • object placed on sensor during startup • cleaning the display with wet cloth • spraying the display with liquid for cleaning

2.4.13 IFS-MMI2C-REQ-421658/A-Baseline temperature drift

Touch Requirement	Specification Limit(s)	Requirement Description
Baseline temperature drift	N/A	Display module shall automatically compensate baseline during temperature drift. During and after temperature changes all fingers shall be tracked.

2.4.14 IFS-MMI2C-REQ-421659/A-Touch performance – low ground mass

Touch Requirement	Specification Limit(s)	Requirement Description
Touch performance – low ground mass	N/A	For low ground mass the display module shall guarantee touch performance. A large ungrounded finger with a touch diameter of 20mm (user is sitting on seat/chair) shall be reported as a single touch.

2.4.15 IFS-MMI2C-REQ-422789/A-Touch performance – moisture present on touch sensor

Touch Requirement	Specification Limit(s)	Requirement Description
Touch performance – moisture present on touch sensor	N/A	While moisture present on touch sensor the display module shall guarantee touch performance. No ghost touches shall be reported when 2.0 cm^3 water droplet is on touch sensor. The touch shall be reported correctly outside droplet.

2.4.16 IFS-MMI2C-REQ-422791/A-Touch performance – mist present on touch sensor

Touch Requirement	Specification Limit(s)	Requirement Description
Touch performance – mist present on touch sensor	N/A	While mist present on touch sensor the display module shall guarantee touch performance. No ghost touches shall be reported when 0.5 cm^3 water mist is covering sensor. One finger shall be reported correctly when 0.5 cm^3 water mist is covering sensor.

2.4.17 IFS-MMI2C-REQ-422792/A-Touch performance – frost present on touch sensor

Touch Requirement	Specification Limit(s)	Requirement Description
Touch performance – frost present on touch sensor	N/A	While frost present on touch sensor the display module shall guarantee touch performance. No ghost touches shall be reported when frost collects on the sensor due to cold start environment. One finger shall be reported correctly when frost collects on the sensor due to cold start environment.

2.4.18 IFS-MMI2C-REQ-422793/A-Touch performance – noisy display pattern

Touch Requirement	Specification Limit(s)	Requirement Description
Touch performance – noisy display pattern	N/A	For noisy display pattern the display module shall guarantee touch performance. No ghost touches shall be reported with a 1pixel x 1pixel black and white checker pattern displayed. One finger shall be reported correctly with a 1pixel x 1pixel black and white checker pattern displayed.

2.4.19 IFS-MMI2C-REQ-421660/A-Self-test

Touch Requirement	Specification Limit(s)	Requirement Description
Self-test	N/A	Display module should have the ability to conduct a self-test with results for: • analog voltages • opens/shorts • capacitance limits

2.4.20 IFS-MMI2C-REQ-422805/A-Response times – active mode

Touch Requirement	Specification Limit(s)	Requirement Description
Response times – active mode	12 msec max	Active mode means the change from an active touch to new touch position. Display Module shall report touch interrupt within the specified time. Typical and maximum response times should be measured.

2.4.21 IFS-MMI2C-REQ-422806/A-Response times – idle to active

Touch Requirement	Specification Limit(s)	Requirement Description
Response times – idle to active	25 msec max	Idle to active means the change from no touch to single touch position. Display Module shall report touch interrupt within the specified time. Typical and maximum response times should be measured.

2.4.22 IFS-MMI2C-REQ-421661/A-Glove Touch

Touch Requirement	Specification Limit(s)	Requirement Description
Glove touch	N/A	One or more fingers shall be reported correctly when user is wearing 2mm thick leather gloves (leather glove with two non-leather inner/ insulation layers). No recalibration shall be required for gloved touch operation. Any touch in gloved operation should be reported as a glove touch status.

2.4.23 IFS-MMI2C-REQ-425877/A-Touch Controller E2E communication RC

The touch controller protocol shall support RC (=rolling counter).

2.4.24 IFS-MMI2C-REQ-425878/A-Touch Controller E2E communication CRC

The touch controller protocol shall support CRC.

2.5 Display Timing Requirements

2.5.1 IFS-MMI2C-REQ-425879/A-Horizontal Timing Requirements

Display Source Timing Requirement	Source Protocol	Requirement Description
Horizontal active area width	MIPI DSI	shall be divisible by 4
Horizontal front porch width		
Horizontal back porch width		
Horizontal sync pulse width		
Total horizontal blanking period		shall be >= 40 pixel clocks
Total vertical blanking period		shall be >= 25 lines
Horizontal active area width	VESA DisplayPort	shall be divisible by 2
Horizontal front porch width		
Horizontal back porch width		
Horizontal sync pulse width		
Total horizontal blanking period		shall be >= 40 pixel clocks
Total vertical blanking period		shall be >= 25 lines

2.6 Display Dimming Requirements

2.6.1 IFS-MMI2C-REQ-426497/A-Display Dimming Requirement

For general illumination dimming, please refer to "*RQT-002004-021873 - General Illumination Dimming*".

3 Functional Definition

3.1 IFS-MMI2C-REQ-418083/A-Start Up for Self Powered Displays

3.1.1 Requirements

3.1.1.1 IFS-MMI2C-TMR-REQ-418092/A-t_INIT

Name	Description	Units	Range	Resolution	Default
t_INIT	Display microcontroller initialization from battery connect	msec	20	20	20

3.1.1.2 IFS-MMI2C-TMR-REQ-418093/A-t_STAB

Name	Description	Units	Range	Resolution	Default
t_STAB	Stabilization from Display Power Enable to IC power	msec	200	200	200

3.1.1.3 IFS-MMI2C-TMR-REQ-418094/A-t_RUN

Name	Description	Units	Range	Resolution	Default
t_RUN	Display Power Enable to Run Mode	msec	300	300	300

3.1.1.4 IFS-MMI2C-TMR-REQ-418095/A-t_DISP_EN

Name	Description	Units	Range	Resolution	Default
t_DISP_EN	Time to Enable LCD and show image	msec	200	200	200

3.1.1.5 IFS-MMI2C-TMR-REQ-420497/A-t_TSC_EN

Name	Description	Units	Range	Resolution	Default
t_TSC_EN	Time to Enable touch controller	msec	250	250	250

3.1.1.6 IFS-MMI2C-TMR-REQ-422302/B-t_BL_FADE

Name	Description	Units	Range	Resolution	Default
t_BL_FADE	Time range to fade in backlight.	msec	700-800	1	750

3.1.2 Use Cases

3.1.2.1 IFS-MMI2C-UC-REQ-418084/A-Start Up for Self Powered Displays

Actors	Host Display
Pre-conditions	Display battery supply is connected.
Scenario Description	Host is powering display-power-enable line. After I2C bus is started up, an ID check is performed. Then Serializer and Deserializer are initialized. Then DISP_EN and TSC_EN shall be sent. After that, the backlight shall be enabled.
Post-conditions	Display is operational. Display is showing an image.
List of Exception Use Cases	None
Interfaces	N/A

3.1.3 Sequence Diagrams

3.1.3.1 IFS-MMI2C-SD-REQ-418086/A-Start Up for Self Powered Displays

Scenarios

Normal Usage

Host is powering display-power-enable line. After I2C bus is started up, an ID check is performed. Then Serializer and Deserializer are initialized. Then DISP_EN and TSC_EN shall be sent. After that, the backlight shall be enabled.

Constraints

Pre-condition

Display battery supply is connected.

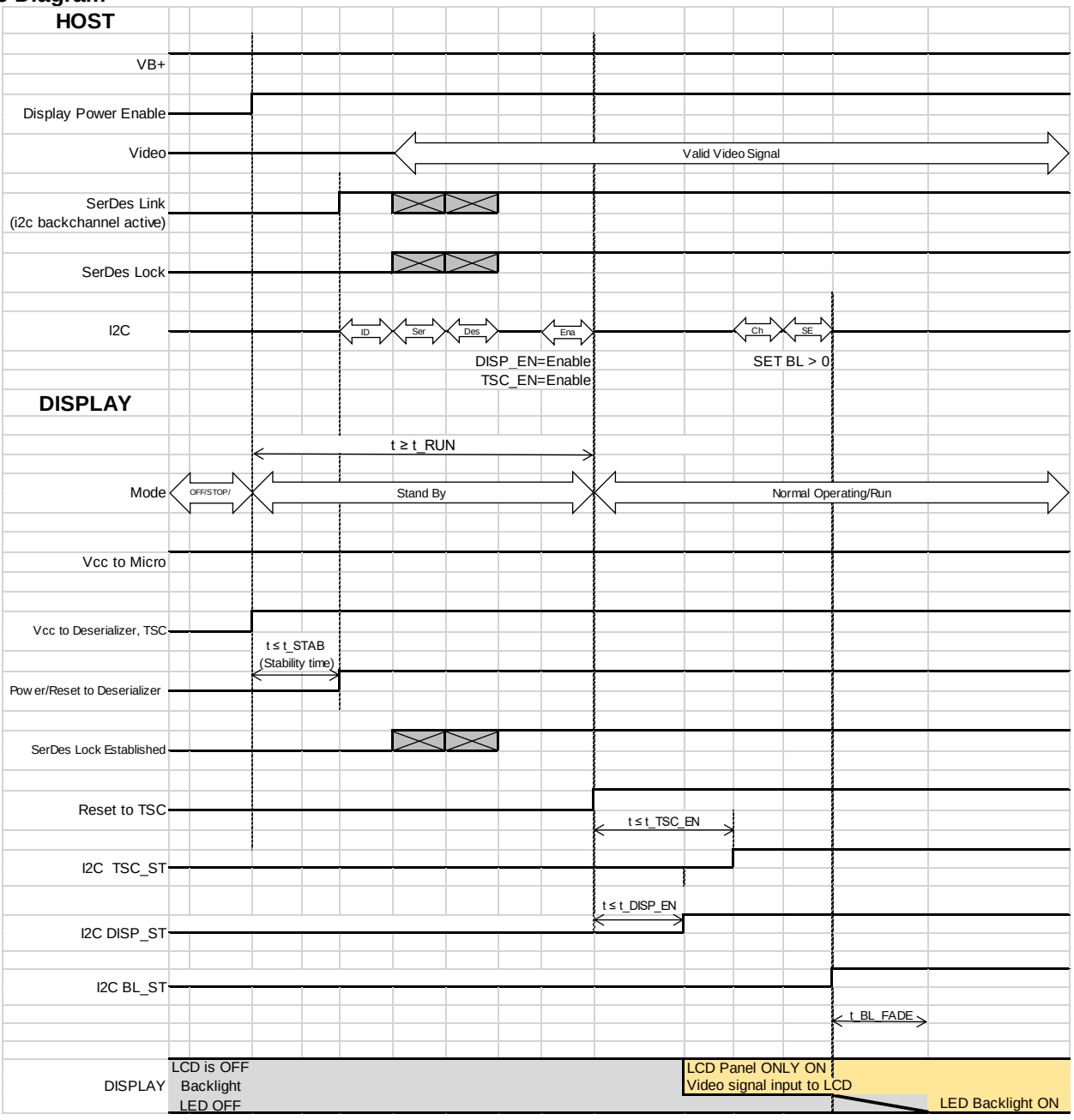
Post-condition

Display is operational.

Post-condition

Display is showing an image.

Sequence Diagram



3.2 IFS-MMI2C-REQ-419617/A-Shutdown for Self Powered Displays normal operation/reset request

3.2.1 Requirements

3.2.1.1 IFS-MMI2C-TMR-REQ-419622/A-t_Shutdown

Name	Description	Units	Range	Resolution	Default
t_Shutdown	Time to perform controlled shutdown	msec	500	500	500

3.2.1.2 IFS-MMI2C-TMR-REQ-419623/A-t_DISP_DIS

Name	Description	Units	Range	Resolution	Default
t_DISP_DIS	Time to Disable LCD and show Blank screen	msec	200	200	200

3.2.1.3 IFS-MMI2C-TMR-REQ-419624/A-t_LVDSSlave_RESET

Name	Description	Units	Range	Resolution	Default
t_LVDSSlave_RESET	Time to completely Reset Display as directed by Source	msec	1000	1000	1000

3.2.2 Use Cases

3.2.2.1 IFS-MMI2C-UC-REQ-419618/A-Shutdown for Self Powered Displays normal operation/reset request

Actors	Host Display
Pre-conditions	System is in normal operation.
Scenario Description	Host sends shutdown command and then Display is performing internal shutdown sequence by first switching off backlight. After “Display Power Enable”-line goes low the LCD goes in sleep/power off.
Post-conditions	Display is in sleep/power off. I2C bus is off.
List of Exception Use Cases	None
Interfaces	N/A

3.2.3 Sequence Diagrams

3.2.3.1 IFS-MMI2C-SD-REQ-419620/A-Shutdown for Self Powered Displays normal operation/reset request

Scenarios

Normal Usage

Host sends shutdown command and then Display is performing internal shutdown sequence by first switching off backlight. After “Display Power Enable”-line goes low the LCD goes in sleep/power off.

Constraints

Pre-condition

System is in normal operation.

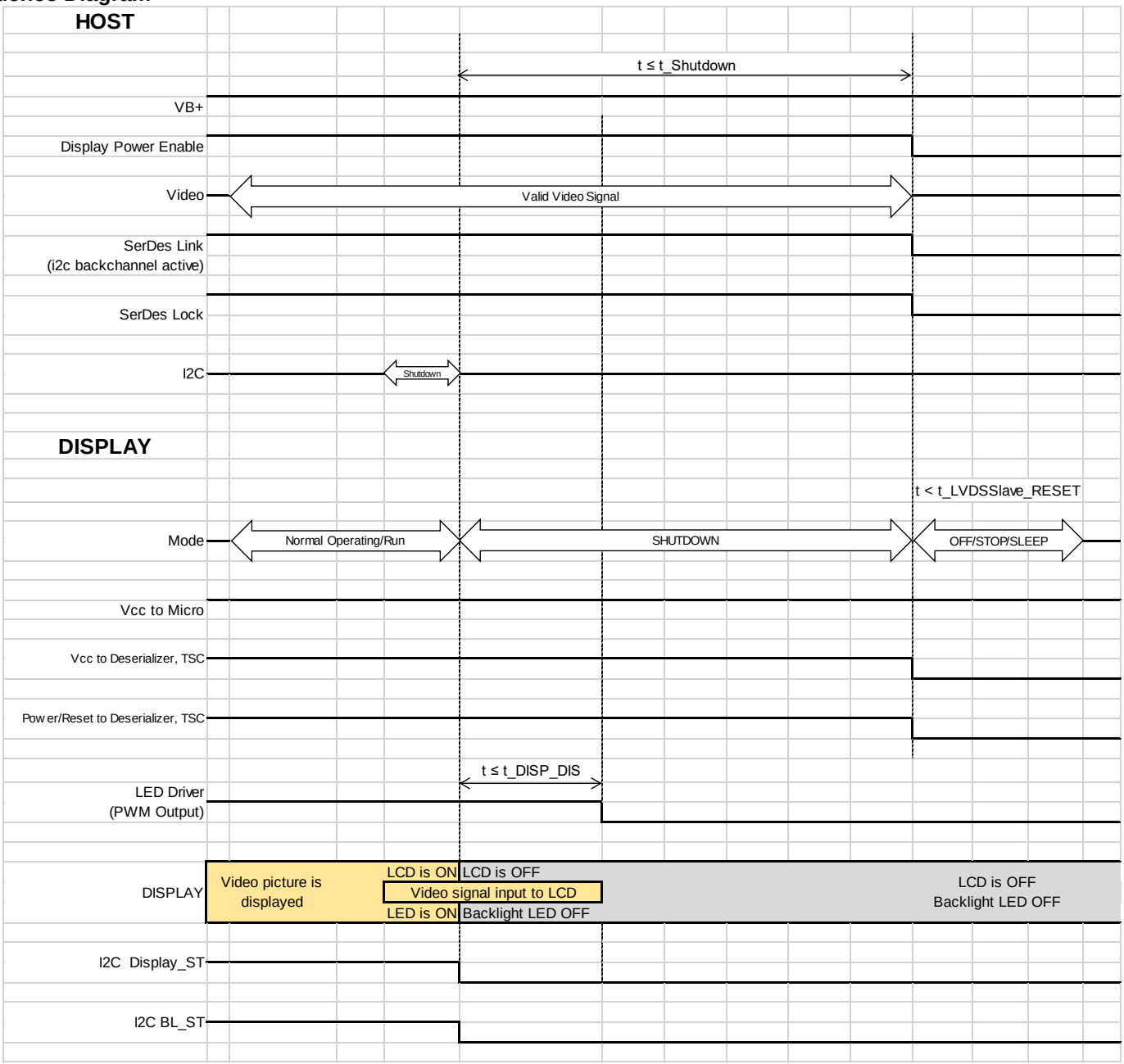
Post-condition

Display is in sleep/power off.

Post-condition

I2C bus is off.

Sequence Diagram



3.3 IFS-MMI2C-REQ-419625/A-Shutdown for Self Powered Displays for recovery

3.3.1 Requirements

3.3.1.1 IFS-MMI2C-TMR-REQ-419622/A-t_Shutdown

Name	Description	Units	Range	Resolution	Default
t_Shutdown	Time to perform controlled shutdown	msec	500	500	500

3.3.1.2 IFS-MMI2C-TMR-REQ-419623/A-t_DISP_DIS

Name	Description	Units	Range	Resolution	Default
t_DISP_DIS	Time to Disable LCD and show Blank screen	msec	200	200	200

3.3.1.3 IFS-MMI2C-TMR-REQ-419624/A-t_LVDSSlave_RESET

Name	Description	Units	Range	Resolution	Default
t_LVDSSlave_RESET	Time to completely Reset Display as directed by Source	msec	1000	1000	1000

3.3.2 Use Cases

3.3.2.1 IFS-MMI2C-UC-REQ-419626/A-Shutdown for Self Powered Displays for recovery

Actors	Host Display
Pre-conditions	System is in any state (normal or fault). Shutdown Message Missed or Not Sent.
Scenario Description	Host is switching off the “Display Power Enable”-line. Then Display is performing internal shutdown sequence by first switching off backlight. After that the LCD goes in sleep/power off.
Post-conditions	Display is in sleep/power off.
List of Exception Use Cases	None
Interfaces	N/A

3.3.3 Sequence Diagrams

3.3.3.1 IFS-MMI2C-SD-REQ-419628/A-Shutdown for Self Powered Displays for recovery

Scenarios

Normal Usage

Constraints

Pre-condition

System is in any state (normal or fault).

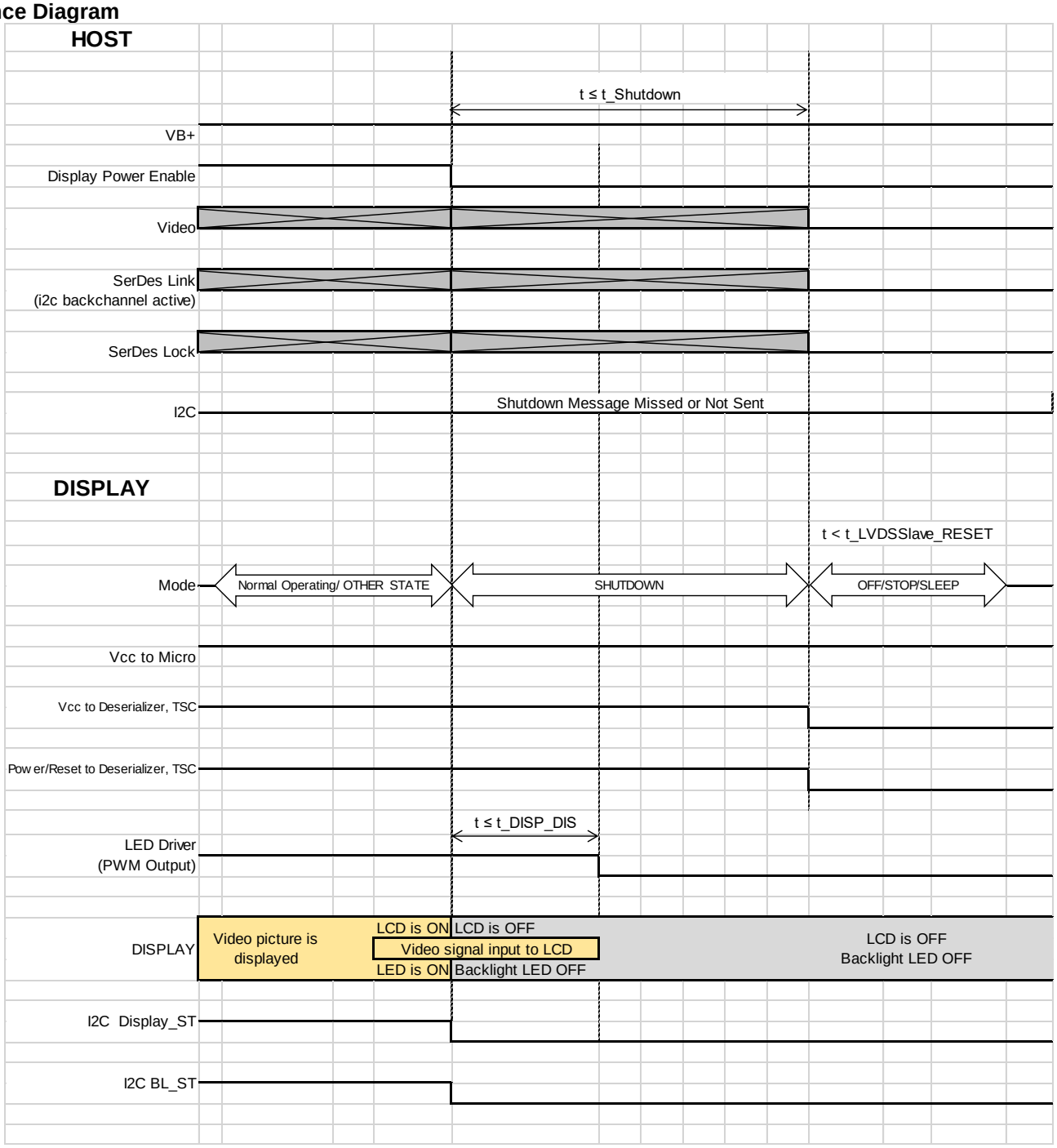
Pre-condition

Shutdown Message Missed or Not Sent.

Post-condition

Display is in sleep/power off.

Sequence Diagram



3.4 IFS-MMI2C-REQ-419637/A-Recovery for Self Powered Displays

3.4.1 Requirements

3.4.1.1 IFS-MMI2C-TMR-REQ-419622/A-t_Shutdown

Name	Description	Units	Range	Resolution	Default
t_Shutdown	Time to perform controlled shutdown	msec	500	500	500

3.4.1.2 IFS-MMI2C-TMR-REQ-419623/A-t_DISP_DIS

Name	Description	Units	Range	Resolution	Default
t_DISP_DIS	Time to Disable LCD and show Blank screen	msec	200	200	200

3.4.1.3 IFS-MMI2C-TMR-REQ-419624/A-t_LVDSSlave_RESET

Name	Description	Units	Range	Resolution	Default
t_LVDSSlave_RESET	Time to completely Reset Display as directed by Source	msec	1000	1000	1000

3.4.1.4 IFS-MMI2C-TMR-REQ-418095/A-t_DISP_EN

Name	Description	Units	Range	Resolution	Default
t_DISP_EN	Time to Enable LCD and show image	msec	200	200	200

3.4.1.5 IFS-MMI2C-TMR-REQ-420497/A-t_TSC_EN

Name	Description	Units	Range	Resolution	Default
t_TSC_EN	Time to Enable touch controller	msec	250	250	250

3.4.1.6 IFS-MMI2C-TMR-REQ-422302/B-t_BL_FADE

Name	Description	Units	Range	Resolution	Default
t_BL_FADE	Time range to fade in backlight.	msec	700-800	1	750

3.4.1.7 IFS-MMI2C-TMR-REQ-422308/A-t_PON

Name	Description	Units	Range	Resolution	Default
t_PON	Time for Power ON	msec	1000	1000	1000

3.4.2 Use Cases

3.4.2.1 IFS-MMI2C-UC-REQ-419642/A-Recovery for Self Powered Displays

Actors	Host Display
Pre-conditions	System is in normal operation.
Scenario Description	DISP_EN and TSC_EN set to disable either dropped or commanded.
Post-conditions	System is in normal operation, again.
List of Exception Use Cases	None
Interfaces	N/A

3.4.3 Sequence Diagrams

3.4.3.1 IFS-MMI2C-SD-REQ-419644/A-Recovery for Self Powered Displays

Scenarios

Normal Usage

DISP_EN and TSC_EN set to disable either dropped or commanded.

Constraints

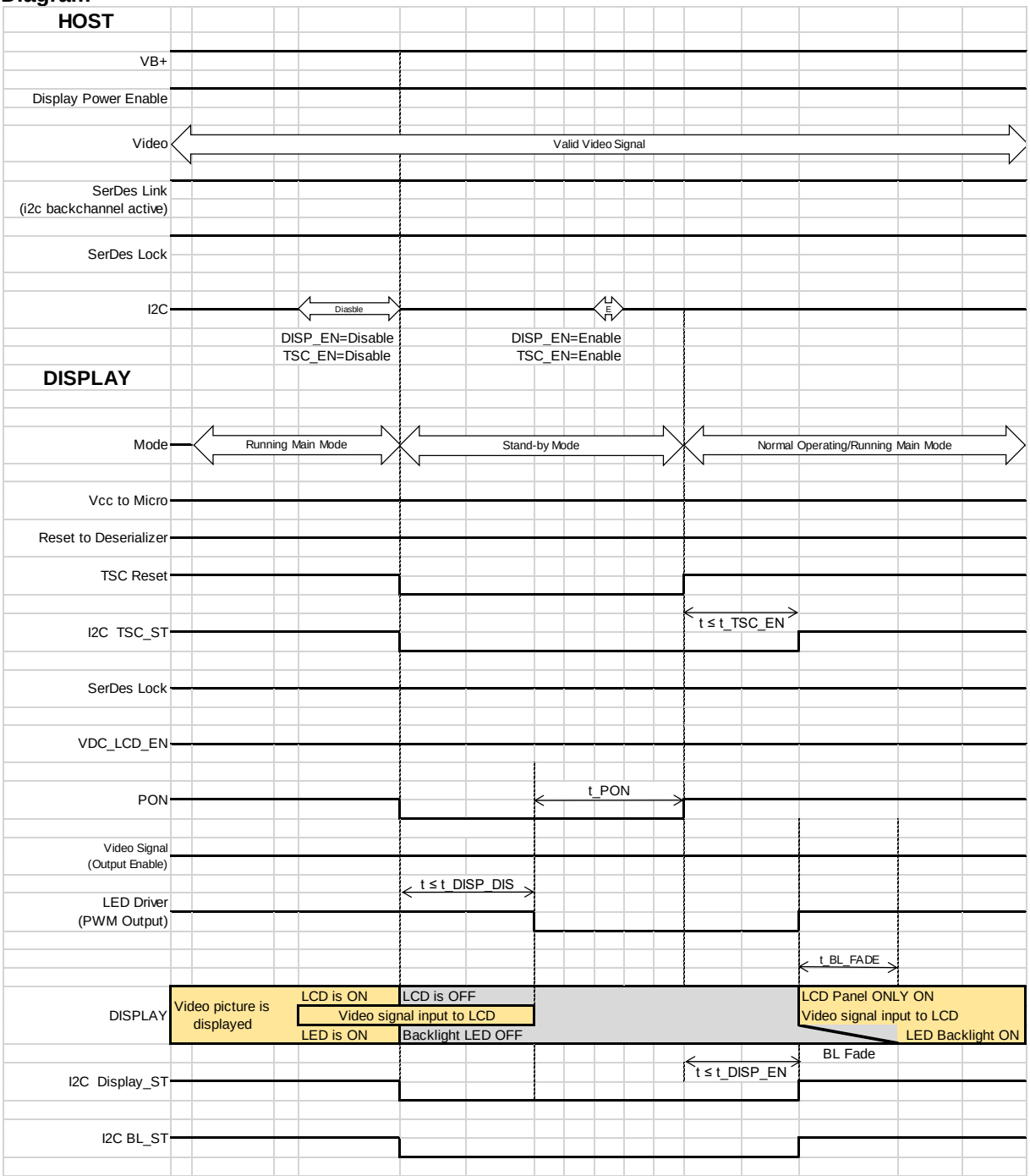
Pre-condition

System is in normal operation.

Post-condition

System is in normal operation, again.

Sequence Diagram



3.5 IFS-MMI2C-REQ-419629/A-Touch Reset for Self Powered Displays

3.5.1 Requirements

3.5.1.1 IFS-MMI2C-TMR-REQ-420497/A-t_TSC_EN

Name	Description	Units	Range	Resolution	Default
t_TSC_EN	Time to Enable touch controller	msec	250	250	250

3.5.1.2 IFS-MMI2C-TMR-REQ-422310/A-t_DON_TCHRST

Name	Description	Units	Range	Resolution	Default
t_DON_TCHRST	Time for Display ON and Touch Reset	msec	100	100	100

3.5.2 Use Cases

3.5.2.1 IFS-MMI2C-UC-REQ-419630/A-Touch Reset for Self Powered Displays

Actors	Host Display
Pre-conditions	Display is operational. I2C bus is operational.
Scenario Description	To reset touch controller Host is setting TSC_EN to disable and back to enable, again.
Post-conditions	Touch controller is operational.
List of Exception Use Cases	None
Interfaces	N/A

3.5.3 Sequence Diagrams

3.5.3.1 IFS-MMI2C-SD-REQ-419632/A-Touch Reset for Self Powered Displays

Scenarios

Normal Usage

To reset touch controller Host is setting TSC_EN to disable and back to enable, again.

Constraints

Pre-condition

Display is operational.

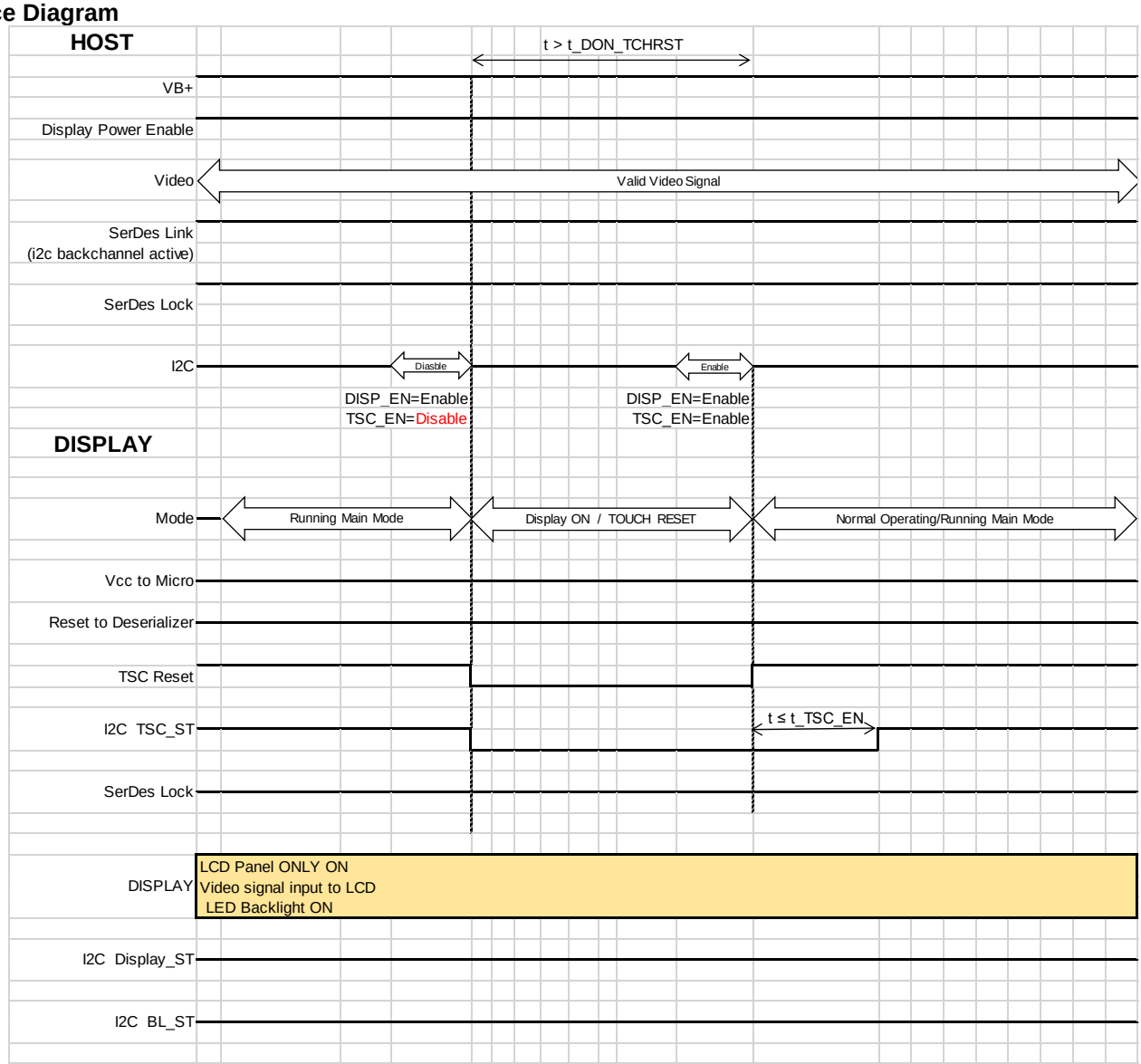
Pre-condition

I2C bus is operational.

Post-condition

Touch controller is operational.

Sequence Diagram

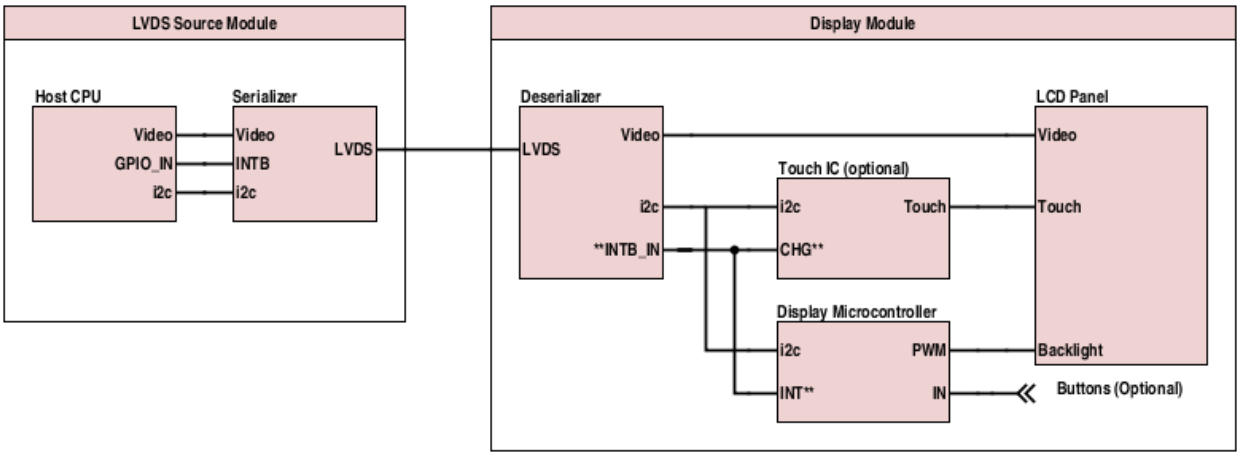


3.6 REQ-403507/A-I2C over LVDS Communication Protocol

3.6.1 IFS-MMI2C-FUN-REQ-140540/C-I2C Interface Displays

3.6.1.1 IFS-MMI2C-SR-REQ-140544/D-System Overview Displays

The LVDS interface block diagram looks like:



** Interrupt architecture varies by display

The LVDS Source Module contains:

- Host CPU, which acts as an I²C bus master
- LVDS Serializer IC

The Display Module contains:

- LVDS Deserializer IC
- Display Microcontroller, which acts as an I²C slave
- LCD Panel
- Touch IC (optional), which acts as an I²C slave
- Buttons (optional)
- Rotary (optional)

3.6.1.2 IFS-MMI2C-SR-REQ-199141/B-Bus Frequency

The I²C Master shall support a bus frequency of 400 kHz. All peripherals on the I²C Slave shall support a bus frequency of 400 kHz.

The I²C Master may configure the deserializer for, and operate at, any bitrate that meets overall system performance requirements. Refer to the TI user's guide for details on configuring the deserializer bitrate with "SCL High Time" and "SCL Low Time".

Note: Even if both sides use 400 kHz, the LVDS link has a lower effective-bitrate (~163 kHz) because each byte is buffered and regenerated. Refer to TI AN-2173 for a table of achievable net bitrates.

3.6.1.3 IFS-MMI2C-SR-REQ-199142/G-Slave Addresses

The following I²C Slave addresses shall be used (7 bit format):

Device	Generation 2 / IPC / HUD / EDM Displays	Generation 2 Cameras
Serializer (92x series)	0x15	0x5D
Serializer (UH92x series)	0x0C	---
Serializer (UH94x series)	0x0C	---
Deserializer (92x series)	0x35	0x30
Deserializer (94x series)	0x34	---
Deserializer (98x series)	0x38	---
Touch Screen Controller (mXT641T)	0x4B	---
Touch Screen Controller (mXT449T)	0x4B	---
Touch Screen Controller (mXT1188S)	0x4A	---
Touch Screen Controller (mxT1067TD)	0x4B	---
Touch Screen Controller (CYAT8268x)	0x24	---
Touch Screen Controller (CYAT827x)	0x24	---
Touch Screen Controller (S7880)	0x20	---
Touch Screen Controller (mXT2912T)	0x4B	---
Touch Screen Controller (TD7800)	0x2C	---
Touch Screen Controller (HX83192-A)	0x48	---
Slave Microcontroller	0x71	0x5E

I²C Slave shall not respond on any other slave address. All unspecified addresses are reserved for future expansion.

3.6.1.4 IFS-MMI2C-SR-REQ-199146/C-Clock Stretching

The I²C Master shall support clock stretching as defined by the I²C specification.

The I²C Slave shall minimize the required clock stretch time, and shall not stretch the clock for longer than 500µs at a time during normal operation (ie DISP_ST and TSC_ST).

Longer clock stretch is allowed outside normal operation (ie initialization) but should not exceed 25ms.

3.6.1.5 IFS-MMI2C-SR-REQ-140571/B-Device Drivers

LVDS Source module shall consider the dynamically-detected display module type, and load the correct device drivers for:
- Video Output
- Touch Input
- Display Control (backlight, status, etc)

For displays containing Atmel chipset, the LVDS Source module shall implement a software device driver based on the atmel_mxt_ts driver published by Atmel at https://github.com/atmel-maxtouch/linux. The driver must support all features of Atmel maXTouch E, S, and T series chips. The driver must support loading touch calibration in *.xcfg or OBP_RAW format. LVDS Source module shall implement Atmel's mxt-app published at https://github.com/atmel-maxtouch/obp-utils

3.6.1.6 IFS-MMI2C-SR-REQ-140549/B-Device Driver

LVDS Source Module shall consider the dynamically-detected display module type, and load the correct device drivers for:
- Video Output
- Touch Input (as applicable)
- Button Input (as applicable)
- Rotary Input (as applicable)
- Display Control (backlight, status, etc)

3.6.1.7 IFS-MMI2C-SR-REQ-140552/B-Display Microcontroller

The display microcontroller shall support the power up / power down requirements of the TSC. The display microcontroller shall provide the proper power sequencing and reset line controls for the TSC to power up / power down properly. The display microcontroller shall use the timing requirements of the TSC to determine that the TSC is ready.

The display microcontroller shall be robust to abrupt power removal.

The display microcontroller shall not update any data accessible over the I^2C interface while an I^2C access is in progress. An I^2C access is bounded by the START and STOP states as defined by the I^2C specification.

3.6.1.8 Touch

3.6.1.8.1 IFS-MMI2C-SR-REQ-140560/A-Touch Screen Pixel Mapping

Display shall be calibrated such that touch coordinates and display pixels have a 1:1 mapping.

3.6.1.8.2 IFS-MMI2C-SR-REQ-140551/B-Touch and Calibration

For a landscape display, the system shall be designed such that LCD [0,0] and Touch [0,0] are both in the top-left corner when viewed by the driver. This means that:

- LCD displays the video signal as top-to-bottom and left-to-right
- Touch is calibrated such that [0,0] is in the top-left corner

For a portrait display, the system shall be designed such that LCD [0,0] is in the bottom-left corner and Touch [0,0] is in the top-left corner when viewed by the driver. This means that:

- LCD is rotated counter-clockwise (-90 degrees) from the landscape orientation
- Touch is calibrated such that [0,0] is in the top-left corner

In both cases the display shall be responsible to control the direction of video (using HRV / VRV), based on the final orientation when installed in a vehicle.

3.6.1.8.3 Atmel Touch Controllers

3.6.1.8.3.1 IFS-MMI2C-SR-REQ-140556/B-Touch Screen Calibration (Atmel E-Series)

The display supplier shall calibrate:

- T9 instance 0 enabled
- T27 pinch, stretch enabled

The LVDS Source Module shall utilize these touch objects for single-touch and multi-touch detection.

The display supplier may utilize any other features to provide robust touch-detection. The LVDS Source module shall be robust against unexpected touch object reports.

3.6.1.8.3.2 IFS-MMI2C-SR-REQ-140558/B-Touch Screen Calibration (Atmel T-Series)

The display supplier shall calibrate:

- T100 instance 0 enabled
- T27 pinch, stretch enabled

The LVDS Source Module shall utilize these touch objects for single-touch and multi-touch detection.

The display supplier may utilize any other features to provide robust touch-detection. The LVDS Source module shall be robust against unexpected touch object reports.

3.6.1.8.3.3 IFS-MMI2C-SR-REQ-202034/A-Signal Limit Threshold

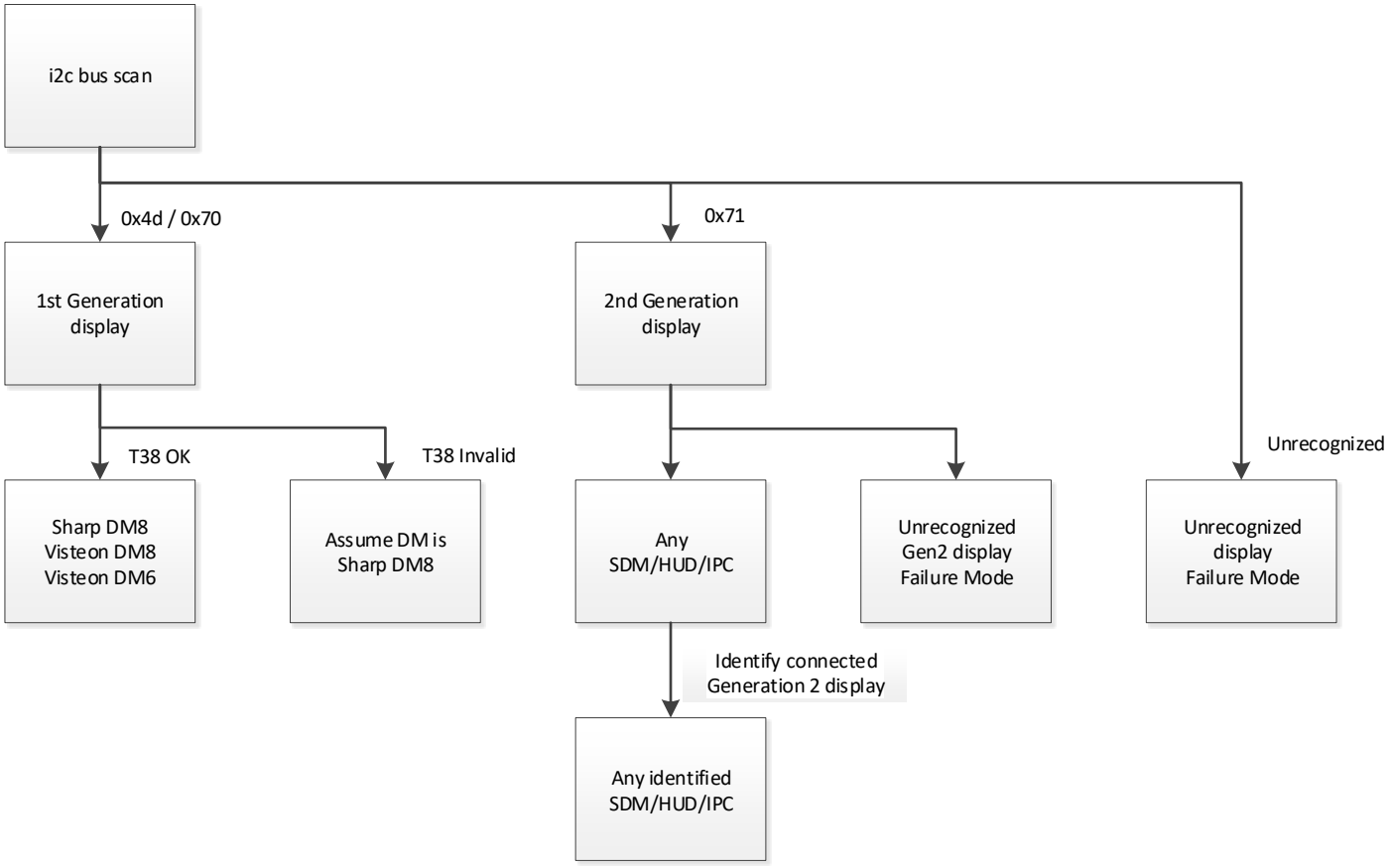
The display supplier shall define thresholds for the T25 signal limit test.

3.6.1.8.4 Cypress Touch Controllers

3.6.1.9 Initialization and Autodetect

3.6.1.9.1 IFS-MMI2C-SR-REQ-140547/C-Identifying Connected Display

At each power-up, I²C Master shall dynamically identify what kind of display is connected with the following strategy:

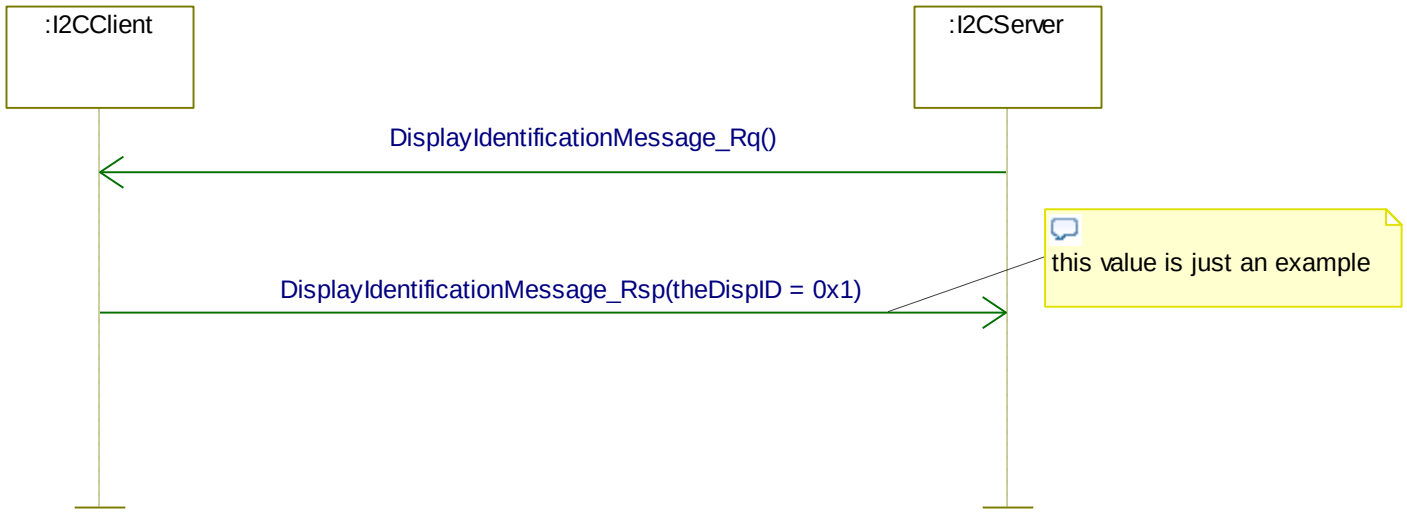


The I²C Master must detect I²C bus errors and restart the sequence until a decision is made.
The I²C Master may support (S)DM/HUD/IPC hotplug. In this case, the I²C Master shall re-run the display identification sequence each time the LVDS cable is disconnected / connected. All dependent steps must be re-evaluated: which device driver to use, which calibration file to load (if applicable) and which HMI to display.

3.6.1.9.2 IFS-MMI2C-SR-REQ-140548/A-Identifying Connected Generation 2 Display

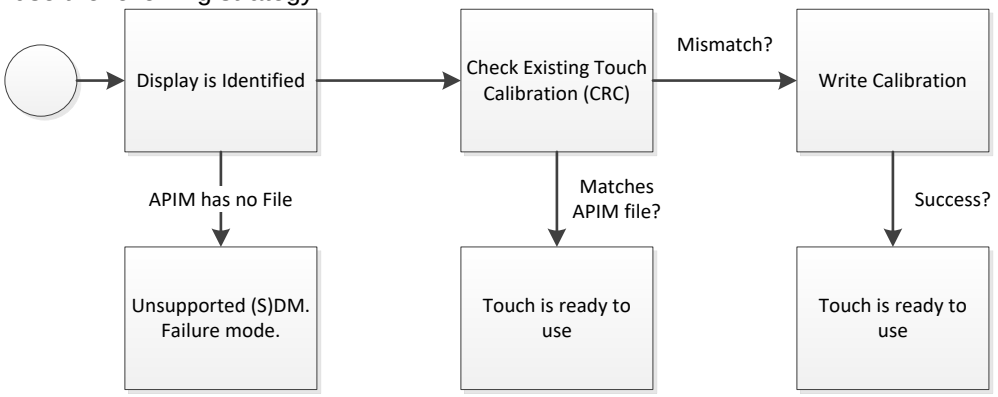
If the LVDS Source Module has been identified that a 2nd generation display is connected it has the possibility to request display ID for choosing e.g. correct calibration.

See requirement “Display Identification Table” for further information on display ID.



3.6.1.9.3 IFS-MMI2C-SR-REQ-140550/A-Touch Calibration

If applicable, the LVDS Source Module is responsible to write the correct touch calibration into the display module. LVDS Source Module shall use the following strategy:



LVDS Source Module shall examine the display, inspect the calibration written into the display, determine if a different calibration is available, and write the calibration if needed.

LVDS Source Module must use a technique that checks the entire touch calibration; computing a hash (CRC) is acceptable, but reading a version number or identifying-mark on the calibration is not acceptable.

LVDS Source Module must minimize flash-memory wear by rewriting the calibration only when required by the strategy.

3.6.1.10 IFS-MMI2C-SR-REQ-140554/C-Timer Settings

The timers described in this section shall have the following values:

Timer	Value
$t_{DISP\_EN(*1)}$	200ms
$t_{DISP\_DIS(*1)}$	200ms
t_{SHTDWN}	500ms
$t_{LVDSSlave\_RESET}$	1000ms

(*1) only valid in case of a display

- $t_{DISP\_EN(*1)}$: Maximum time to enable LCD Panel, LCD Backlight, and display an image on the LCD.
- $t_{DISP\_DIS(*1)}$: Maximum time to disable LCD Panel, LCD Backlight, and show a blank screen.
- t_{SHTDWN}: Maximum time to perform a controlled shutdown.

3.6.1.11 IFS-MMI2C-SR-REQ-199145/B-Time to Ready

All I^2C Slaves (e.g. TouchScreenController and display microcontroller) shall be capable of communicating on the I^2C bus 300ms after the filtered battery supply is enabled. Actual communication cannot begin until LOCK is achieved on the LVDS link between the LVDS Client and LVDS Source Module.

3.6.1.12 Diagnostics

This table summarizes the diagnostic requirements:

Requirement	Name
REQ-199350	Unsupported Display
REQ-197882	Unexpected Reset
REQ-197881	Reset Request
REQ-197883	Display Connection Error
REQ-197885	Thermistor Backlight De-rating
REQ-197884	Backlight Circuit Fault
REQ-197886	Touch Screen Error
REQ-199371	LVDS Link Fault
REQ-197887	Loss of Lock
REQ-199353	Lost Comm. Display Microprocessor
REQ-199355	Lost Communication Touch Controller
REQ-199354	Touch Circuit Fault
REQ-199369	Touch Panel Fault
REQ-199370	Touch Panel Range/Performance
REQ-266614	Touch Circuit Fault
REQ-266615	Touch Panel Fault
REQ-266616	Touch Panel Range/Performance

3.6.1.12.1 IFS-MMI2C-SR-REQ-199350/B-Unsupported Display

During initialization of a display module, the LVDS Source Module reads the electronic identifier from display microcontroller subaddress 0x01.

If the LVDS Source Module does not support the type of display which is connected, it shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC based on this error.

3.6.1.12.2 IFS-MMI2C-SR-REQ-197882/C-Unexpected Reset

The I²C Slave module bit INIT is cleared (=0) at power-on, and set (=1) at I²C Slave enable. Any transition from 1 -> 0 during normal operation indicates that I²C Master Module was operating normally but:

1. The I²C Slave was disconnected and same-or-different I²C Slave was connected.
2. Or the I²C Slave reset, for example: low-voltage, watchdog, etc.

After an I²C Slave is initialized, I²C Master Module shall monitor INIT and set appropriate DID if an unexpected reset was detected.

The I²C Master Module shall implement a counter and set appropriate DTC if there are greater than 5 events detected during any single ignition cycle.

3.6.1.12.3 IFS-MMI2C-SR-REQ-197881/C-Reset Request

The I²C Slave module is permitted to set RST_RQ to request a full power-cycle. I²C Slaves are known to make this request after detecting loss-of-lock, low-voltage dropout, and backlight fault. The I²C Slave shall only make this request if the fault can be fixed by cycling power.

The I²C Master Module shall monitor bit RST_RQ and set appropriate DID if a reset was requested.

The I²C Master Module shall implement a counter and set appropriate DTC if there are greater than 5 reset requests during any single ignition cycle.

3.6.1.12.4 IFS-MMI2C-SR-REQ-197883/B-LCD Connection

The display module shall monitor the flexible cable connecting PCB to LCD panel, and report the status with bit DCERR.

The LVDS Source Module shall monitor bit DCERR and set appropriate DID.

The LVDS Source Module shall set appropriate DTC based on this error. The diagnostic has detected a faulty connection inside the display, and the recommended action is to replace the display.

3.6.1.12.5 IFS-MMI2C-SR-REQ-197885/C-Temperature Derating

The display module shall monitor temperature and report when the module starts brightness derating due to an over-temperature condition with bit TERR.

The LVDS Source Module shall monitor bit TERR and message “0x09 Momentary LCD Backlight PWM Status“ and set appropriate DID and DTC.

3.6.1.12.6 IFS-MMI2C-SR-REQ-197884/B-LCD Backlight

The display module shall monitor the LCD backlight controller for any fault, and report the status with bit BLERR.

The LVDS Source Module shall monitor bit BLERR and set appropriate DID.

The LVDS Source Module shall set appropriate DTC based on this error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.7 IFS-MMI2C-SR-REQ-197886/B-LCD Module / Panel

If the display module is capable of monitoring the LCD panel for a fault, it shall report the status with bit LCDERR. Not all display modules are capable of this diagnostic.

The LVDS Source Module shall monitor bit LCDERR and set appropriate DID.

The LVDS Source Module shall set appropriate DTC based on this error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.8 IFS-MMI2C-SR-REQ-199371/D-LVDS Link Detect Fault

When the LVDS Source Module is providing power to the LVDS Slave Module, it shall monitor LVDS chip register “LINK Status“. If the LVDS Source Module detects an LVDS serial link fault it shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC based on this error. The diagnostic has detected a connection fault.

3.6.1.12.9 IFS-MMI2C-SR-REQ-197887/C-Loss of Lock

While enabled (DISP_EN=1), the LVDS Source Module shall monitor bit LLOSS and set appropriate DID if the deserializer reports a loss-of-lock event.

The LVDS Source Module shall implement a counter and set appropriate DTC if there are greater than 5 loss-of-lock events during any single ignition cycle. The diagnostic has detected a signal-quality problem with communication to the display module.

3.6.1.12.10 IFS-MMI2C-SR-REQ-199353/B-Loss of Communication with Display Microcontroller

During normal operation, the LVDS Source Module display driver shall determine loss-of-communication by monitoring for persistent I^2C NAK response.

If the LVDS Source Module detects a condition where the LVDS link is operational but the display microcontroller has a persistent NAK response (> 500ms), it shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for this error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.11 IFS-MMI2C-SR-REQ-199355/B-Loss of Communication with Touch IC

During normal operation, the LVDS Source Module display driver shall monitor for persistent I^2C communication faults.

If the LVDS Source Module detects a condition where the LVDS link is operational but the touch IC has a persistent NAK response, it shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for this error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.12 IFS-MMI2C-SR-REQ-199354/C-AVdd Power Test (Atmel-only)

During Self-Test [0202] the LVDS Source Module shall command the Atmel Touch IC to run T25 AVdd Power Test. Based on the test result, the LVDS Source Module shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for any error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.13 IFS-MMI2C-SR-REQ-266614/B-AVdd Power Test (Cypress-only)

The LVDS Source Module shall monitor the Cypress touch IC error register.
Based on the AVDD error result, the LVDS Source Module shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for any error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.14 IFS-MMI2C-SR-REQ-199369/C-Pin Fault Test (Atmel-only)

During Self-Test [0202] the LVDS Source Module shall command the Atmel Touch IC to run T25 Pin Fault Test. Based on the test result, the LVDS Source Module shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for any error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.15 IFS-MMI2C-SR-REQ-266615/B-Pin Fault Test (Cypress-only)

During Self-Test [BIST] the LVDS Source Module shall command the Cypress Touch IC to run Built In Self Test. Based on the test result for SHORTS, the LVDS Source Module shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for any error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.16 IFS-MMI2C-SR-REQ-199370/C-Signal Limit Test (Atmel-only)

During Self-Test [0202] the LVDS Source Module shall configure appropriate signal levels then command the Atmel Touch IC to run T25 Signal Limit Test. Based on the test result, the LVDS Source Module shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for any error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.12.17 IFS-MMI2C-SR-REQ-266616/B-Signal Limit Test (Cypress-only)

During Self-Test [BIST] the LVDS Source Module shall command the Cypress Touch IC to run Built In Self Test. Based on the test result for Cp/Cm Tests, the LVDS Source Module shall set appropriate DID.

The LVDS Source Module shall set appropriate DTC for any error. The diagnostic has detected a failure inside the display, and the recommended action is to replace the display.

3.6.1.13 Failure Mode Avoidance

3.6.1.13.1 IFS-MMI2C-SR-REQ-202030/B-Reset Request

The LVDS Slave module is permitted to set RST_RQ to request a full power-cycle. The LVDS Slave shall only make this request if the fault can be fixed by cycling power (e.g. Gen1 displays are known to make this request after detecting loss-of-lock, low-voltage dropout, and backlight fault).

The LVDS Source Module shall monitor bit RST_RQ. If bit RST_RQ=1, the LVDS Source Module shall perform a controlled power shutdown. After $t_{LVDSSlave\_RESET}$ expires the LVDS Source Module shall re-enable power to the LVDS Slave and perform a normal re-initialization sequence.

3.6.1.13.2 IFS-MMI2C-SR-REQ-202033/B-Loss of Communication

During normal operation, if the LVDS Source Module detects a condition where the LVDS Slave has become non-functional, either Link Detect Fault or Loss of Communication, it shall perform a full power-cycle as an attempt to recover the LVDS Slave.

The full power-cycle sequence is: disable power, wait $t_{LVDSSlave\_RESET}$, then perform a full re-initialization sequence.

3.6.1.13.3 IFS-MMI2C-SR-REQ-197874/A-Interrupt Polling

The LVDS chipset does not mirror interrupt status; it only asserts INTB on a falling-edge of INTB_IN. Because the system is edge-sensitive, it is possible for the system to enter an error-state where INTB_IN is asserted but INTB is deasserted.

The LVDS Source Module shall use a timer and run the normal interrupt handling if an interrupt has not been seen for 100ms. This is intended to recover the system from a “stuck interrupt” condition.

3.6.1.13.4 IFS-MMI2C-SR-REQ-199134/B-Restart AEQ Algorithm

The LVDS chipset has an undocumented auto equalization (AEQ) behavior. When attempting to establish an LVDS link, the deserializer will begin with the minimum EQ setting and try to lock. If unsuccessful, it increments EQ and tries again. It repeats this routine until lock is established. Resetting the deserializer forces a restart at the beginning of the algorithm.

In any situation where deserializer is powered and running the search algorithm before the serializer is ready, the EQ setting could lock to a larger-than-necessary value.

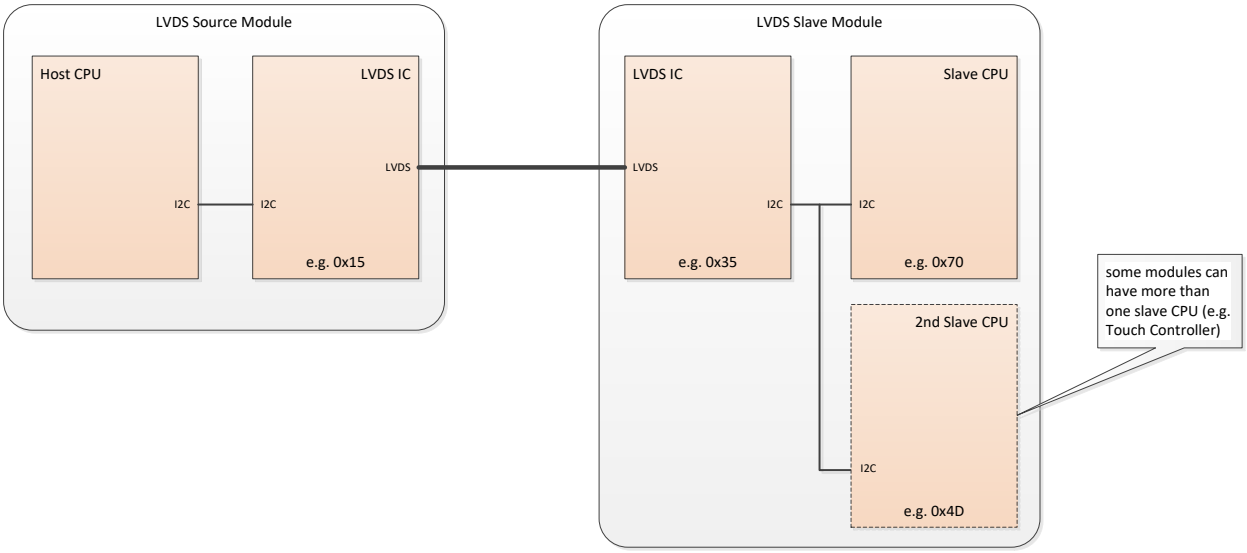
Therefore, the LVDS Source Module shall reset the deserializer:

- After resetting the serializer
- After changing LFMODE

3.6.1.13.5 IFS-MMI2C-SR-REQ-199348/B-Atomic Transaction

The LVDS chipset as an undocumented requirement regarding the sequence of I^2C messages across the LVDS link. Any transaction to the SER or DES must be performed in an “atomic” manner, because any I^2C message that flows across the link will overwrite the register offset.

For example:



The following sequences are permissible:

- Write offset to 0x15, read 0x15, write offset to 0x4D, read 0x4D
- Write offset to 0x4D, write offset to 0x70, read 0x4D, read 0x70

The following sequences are not permissible and result in **incorrect transactions**:

- Write offset to 0x15, write offset to 0x4D, **read 0x15**, read 0x4D
- Write offset to 0x35, write offset to 0x4D, **read 0x35**, read 0x4D

Note: All addresses mentioned in this requirement are just examples to be more descriptive.

3.6.1.13.6 IFS-MMI2C-SR-REQ-199357/A-Avoid driving INTB_IN during loss-of-lock

The LVDS chipset has an undocumented requirement regarding a falling-edge of INTB_IN during loss-of-lock. In this situation the interrupt may be missed, and the LVDS Source Module will not receive the signal.

The display module may use a “buffer-and-defer” strategy (STR-307941) to delay generating an interrupt request until lock is regained.

The system will recover when another interrupt arrives, or worst-case when LVDS Source Module performs interrupt polling (REQ-197874).

3.6.1.13.7 IFS-MMI2C-SR-REQ-226922/A-Write Configuration to Flash

Several versions of Atmel’s Touch IC have an undocumented requirement to maintain power when saving a configuration to non-volatile memory with the BACKUPNV command.

If the LVDS Source Module removes power during a BACKUPNV operation, the configuration region of the flash memory may become unstable. The Atmel Touch IC will detect the condition, will not enter the firmware, and will remain stuck in the bootloader until actions are taken to clear the configuration region.

Because the LVDS Source Module cannot guarantee an uninterrupted BACKUPNV operation, the LVDS Source Module shall support in-field recovery using Atmel’s recommended technique.

3.6.2 IFS-MMI2C-FUN-REQ-140607/D-Generation 2 Display Modules

3.6.2.1 IFS-MMI2C-SR-REQ-140609/A-Display Module Identification

The display module supplier shall program into the display microcontroller, at time of manufacturing, e.g. following information:

- Display ID
- Ford Part Number
- Electronic Serial Number
- Software Part Number (Firmware Version)

3.6.2.2 IFS-MMI2C-SR-REQ-140612/B-Touch Screen Controller Operation

The display module supports a touch interface on some versions. The touch screen controller shall use the standard I^2C communication protocol defined by the touch screen controller vendor. The interrupt lines shall be serviced as defined by both the touchscreen controller documentation and the TI FPD Link III documentation for handling the back-channel interrupt signal.

3.6.2.3 IFS-MMI2C-SR-REQ-140613/B-Button Controller Operation

The display module supports a button array interface on some versions. The Display Module shall trigger an interrupt whenever a button press with button debounce or a button release with hysteresis occurs. The button press information shall be transmitted via the standard I^2C communication protocol defined in this document.

3.6.2.4 IFS-MMI2C-SR-REQ-408696/A-Backlight Error

The display shall report any condition that prevents the backlight from turning on. While (DISP_EN=1) if any condition causes BL_ST=0 (even momentary recovery), the display shall

- Latch that a backlight error was detected (BLERR= 1)

3.6.2.5 IFS-MMI2C-SR-REQ-408694/A-Valid timing check

The display shall monitor valid video timing is received. For any instance, the video timing is incorrect while (DISP_EN=1), the display shall

- Automatically turn-off LCD backlight and panel to prevent showing bad video. This will not affect the commanded setting (DISP_EN), but must be reported in the actual status (DISP_ST =0).
- Latch that a DISPERR was detected (DISPERR = 1) indicating the display is not on.

3.6.2.6 IFS-MMI2C-SR-REQ-408695/A-Display Error

The display shall report any condition that prevents the display from enabling and shall report any condition that causes a dropped enable. While (DISP_EN=1) if any condition causes DISP_ST=0 (even momentary recovery), the display shall

- Latch that a display error was detected (DISPERR = 1)

3.6.2.7 Interrupt

In this design, multiple functions within the display are able to interrupt the LVDS Source Module. These include:

1. Any display status change / error event
2. CHG asserted by the Touch IC (for displays with touch)
3. Any button message (for displays with buttons)
4. Any rotary message (for displays with one or more rotary)

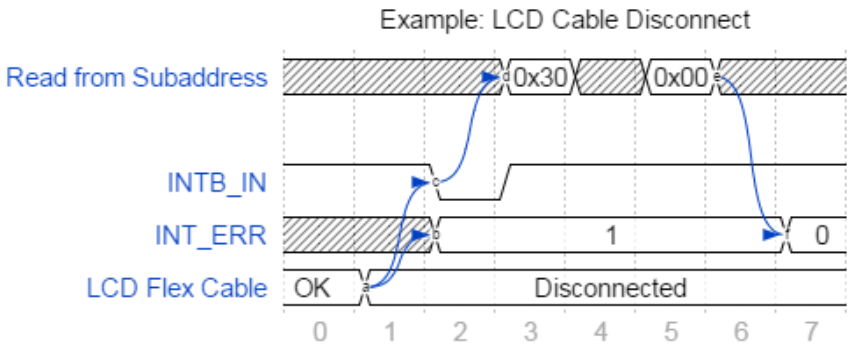
The display supplier may choose any hardware implementation that honors the interface requirements.

3.6.2.7.1 IFS-MMI2C-SR-REQ-197941/B-Interrupt Request

The LVDS chipset does not mirror interrupt status; it only asserts INTB on a falling-edge of INTB_IN. In this system, the interrupt request is an edge-triggered event.

The I²C Slave shall generate an interrupt request whenever an interrupt-generating event occurs. An interrupt-generating event is defined as any event that could cause a bit in subaddress 0x30 (ISR) to transition from 0 -> 1.

This diagram shows an example with the relationship of reading and clearing:



- Over temperature is an interrupt-generating event. The I²C Slave microcontroller sets corresponding bit in subaddress 0x30 (e.g. TERR, OVRTMP) here named as INT_ERR, then generates an interrupt request by driving INTB_IN (t=2).
  Note: The I²C Slave will generate an interrupt request for this event; it doesn't matter if INT_ERR =1 already from a previous unserved interrupt.
- Read of I²C Slave microcontroller subaddress 0x30 (t=3) to understand cause of interrupt.
- Read of I²C Slave microcontroller subaddress 0x00 (t=5) clears INT_ERR (t=7).
  Note: The fault still exists.

3.6.2.7.2 IFS-MMI2C-SR-REQ-436478/A-Fault Detection Timing

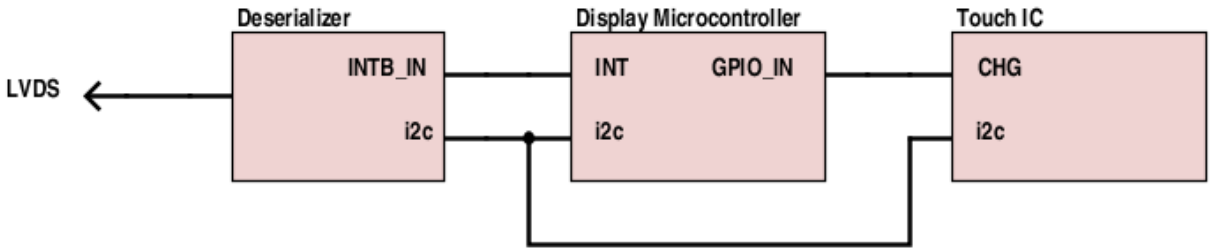
The display module microprocessor (MCU) shall detect a fault and report an interrupt to the LVDS source module within:

Time < 50 msec max

3.6.2.7.3 Single IC driving INTB_IN

The display may implement a circuit where display microcontroller is the only IC that drives INTB_IN. This is recommended as the simplest implementation.

For displays with touch, the block-diagram would look like:

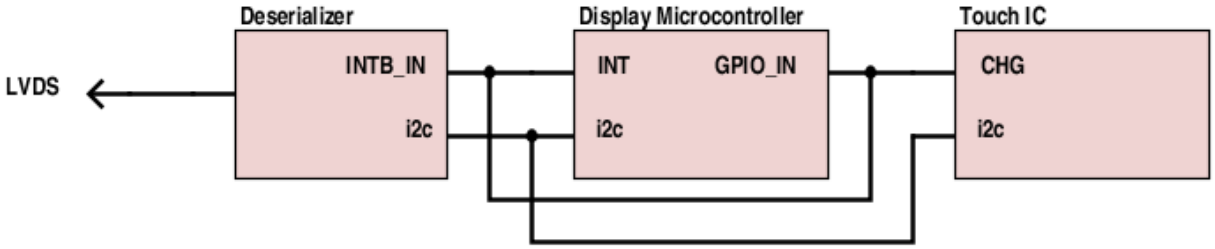


In this case, the display microcontroller monitors CHG and generates an interrupt request for any interrupt-generating event.

3.6.2.7.4 Multiple ICs driving INTB_IN

The display may implement a circuit where both display microcontroller and touch IC are able to drive the INTB_IN signal (as open-drain).

For displays with touch, the block-diagram would look like:



In this case, INTB_IN is driven by a different IC for each feature:

- Touch IC drives INTB_IN to generate an interrupt request for INT_TCH.
- Display microcontroller drives INTB_IN to generate an interrupt request for INT_BTN and INT_ERR.

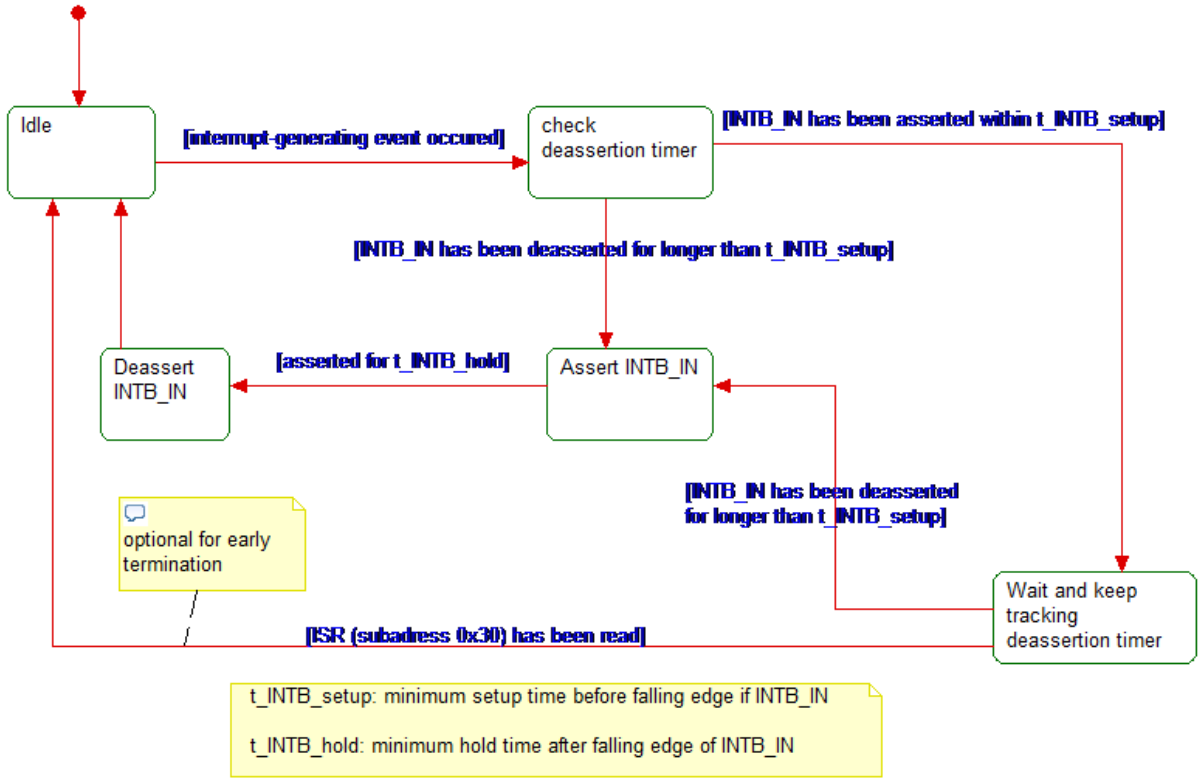
Display microcontroller reads INTB_IN to determine when the Touch IC is requesting an interrupt, so it can report an accurate INT_TCH value.

This circuit is vulnerable to a race-condition where two-or-more ICs attempt to generate an interrupt request at the same time. If multiple ICs drive INTB_IN an interrupt may be missed and the system may deadlock.

3.6.2.7.5 Interrupt Request Strategy

The display must implement a strategy to meet setup-time and hold-time requirements around an INTB_IN falling edge. This is applicable to both proposed hardware designs.

This diagram shows a strategy for the display microcontroller to “buffer-and-defer” the interrupt to meet timing constraints. The microcontroller monitors INTB_IN, verifies that INTB_IN is de-asserted for the required setup-time, drives INTB_IN for the required hold-time, then releases INTB_IN:



With two ICs driving INTB_IN (STR-307934), the display microcontroller may use this strategy to “buffer-and-defer” an interrupt request until after the touch IC releases INTB_IN.

3.6.2.7.6 IFS-MMI2C-SR-REQ-198936/B-Interrupt Service

When the LVDS Source Module receives the interrupt it shall:

1. Read SER subaddress 0xC7 (ISR).
   Note: This action causes serializer to deassert INTB, preparing the system to assert INTB again on the next falling-edge of deserializer INTB_IN.
2. Read display microcontroller subaddress 0x30 (ISR). Determine which sources have an interrupt pending.
3. Service each pending interrupt: touch, buttons, rotary, or status.

3.6.2.8 IFS-MMI2C-SR-REQ-197933/C-Loss of Lock Displays

The display shall monitor deserializer LOCK pin; even momentary loss-of-lock must be detected. Suggested to use interrupt on Display microcontroller. For any loss-of-lock, the display shall:

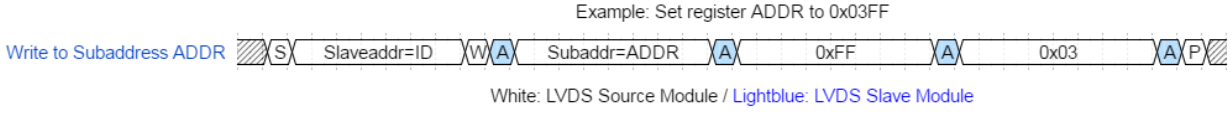
- Automatically turn-off LCD backlight and panel to prevent showing bad video. This will not affect the commanded setting (DISP_EN), but will be reported in the actual status (DISP_ST =0).
- Latch that a loss-of-lock was detected (LLOSS = 1).
- If lock is regained, consider the commanded setting (DISP_EN) and determine if the LCD backlight and panel need to be automatically turned back on.

This strategy allows the system to recover quickly and automatically from a momentary loss-of-lock, without a full power-cycle. It also allows the LVDS Source Module to distinguish between signal-quality problems and a fault that requires full reset.

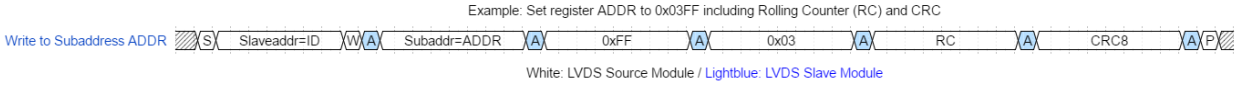
3.6.2.9 I2C Bus Interface

3.6.2.9.1 IFS-MMI2C-SR-REQ-140564/D-Write to Subaddress

This diagram shows a typical write by the I²C Master. Writes are implemented by writing the subaddress then one-or-more bytes of data:

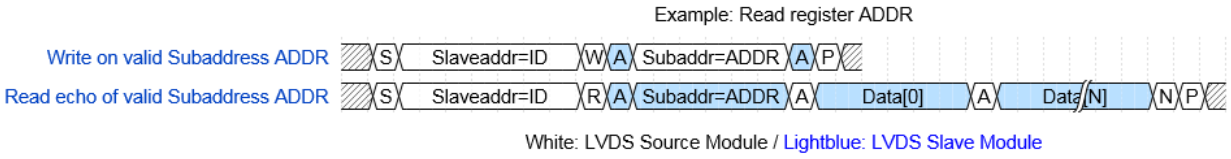


Same example showing functional safety support:

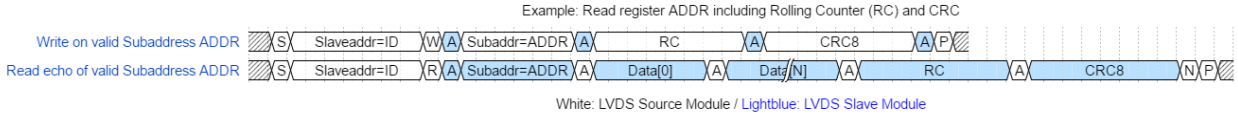


3.6.2.9.2 IFS-MMI2C-SR-REQ-140561/E-Read from Subaddress

This diagram shows a typical read by the I²C Master. Reads are implemented by writing the subaddress, then reading an echo of the subaddress followed-by the data:

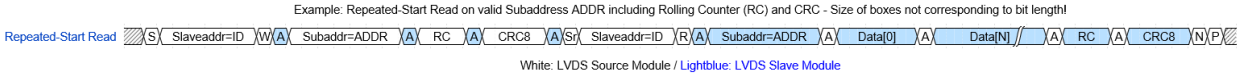


Same example showing functional safety support:



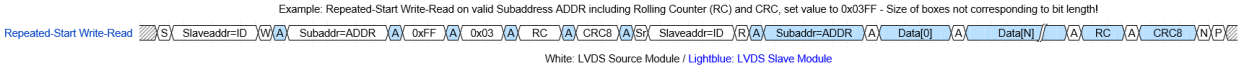
3.6.2.9.3 IFS-MMI2C-SR-REQ-403502/A-Repeated-Start Read from Subaddress

This diagram shows a typical read by the I²C Master using repeated start. Reads are implemented by writing the subaddress, then reading an echo of the subaddress followed-by the data:



3.6.2.9.4 IFS-MMI2C-SR-REQ-403503/A-Repeated-Start Write-Read from Subaddress

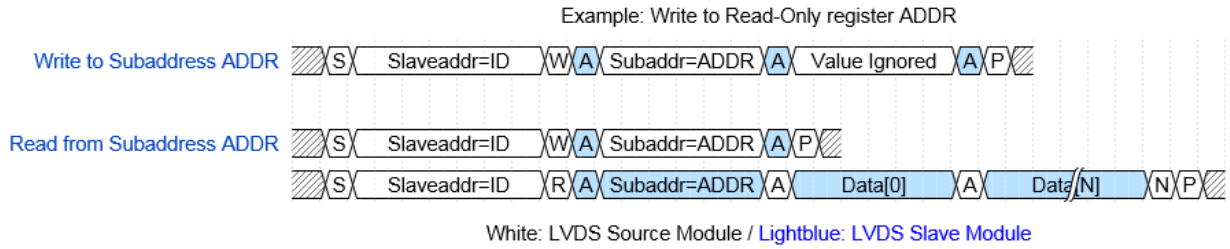
This diagram shows a typical write by the I²C Master using repeated-start. Writes are implemented by writing the subaddress then one-or-more bytes of data. Reads are implemented by writing the subaddress, then reading an echo of the subaddress followed-by the data:



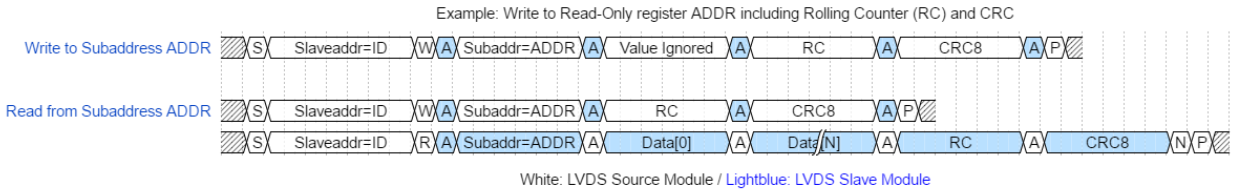
3.6.2.9.5 IFS-MMI2C-SR-REQ-197857/D-Write to Read-Only Subaddress

If I²C Master Module attempts to write to a read-only subaddress, the I²C Slave shall send ACK to indicate the bytes are received but make no state-change.

As an example with subaddress ADDR that will be read-only. This diagram shows that the I²C Slave ignores an attempt to write to the subaddress:



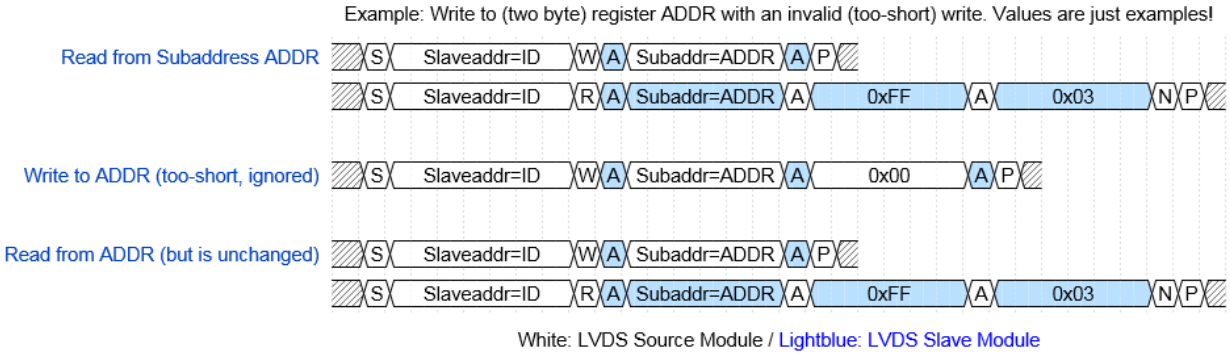
Same example showing functional safety support:



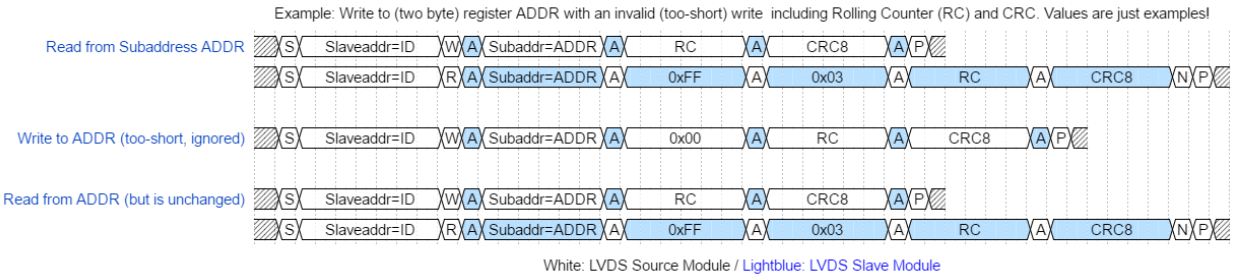
3.6.2.9.6 IFS-MMI2C-SR-REQ-140569/E-Write Underflow

If the I²C Master writes too-few bytes, the I²C Slave shall make no state-change.

As an example with subaddress ADDR that will accept exactly two bytes of data. This diagram shows I²C Master attempting to write only one byte of data, and the I²C Slave making no state-change:



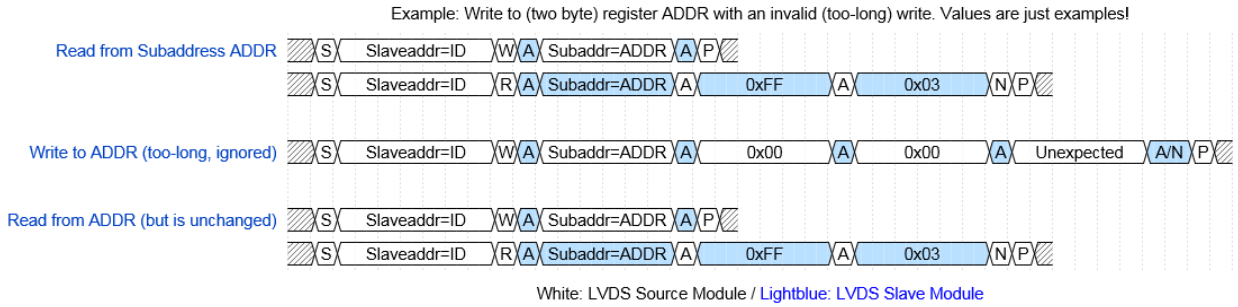
Same example showing functional safety support:



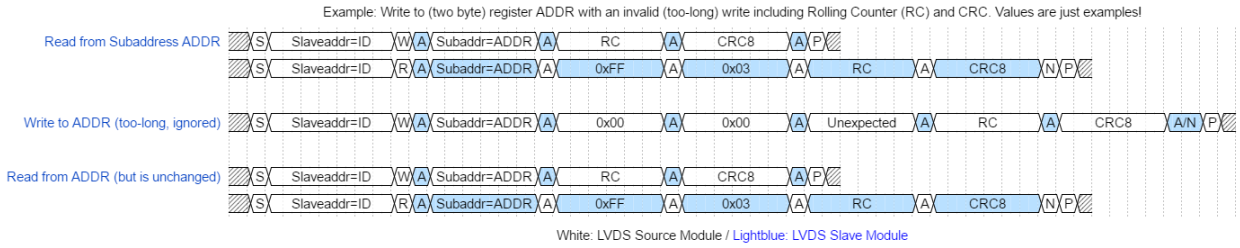
3.6.2.9.7 IFS-MMI2C-SR-REQ-140570/E-Write Overflow

If the I²C Master attempts to write too-many bytes, the I²C Slave shall continue sending ACK to indicate the bytes are received, and make no state-change.

As an example with subaddress ADDR that will accept exactly two bytes of data. This diagram shows I²C Master attempting to write three bytes, and the I²C Slave making no state-change:

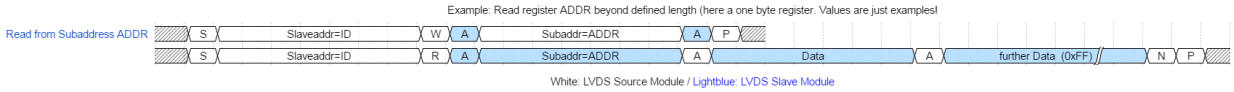


Same example showing functional safety support

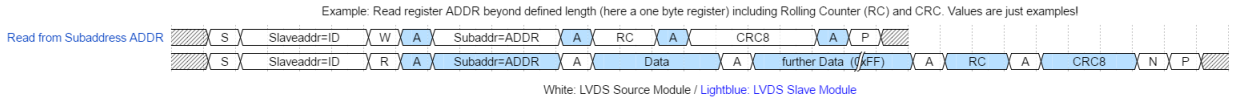


3.6.2.9.8 IFS-MMI2C-SR-REQ-197875/D-Read from Subaddress Beyond Defined Length

If I²C Master continues reading beyond the defined data-length of a subaddress, I²C Slave shall leave SDA undriven resulting in Data = 0xFF.



Same example showing functional safety support:



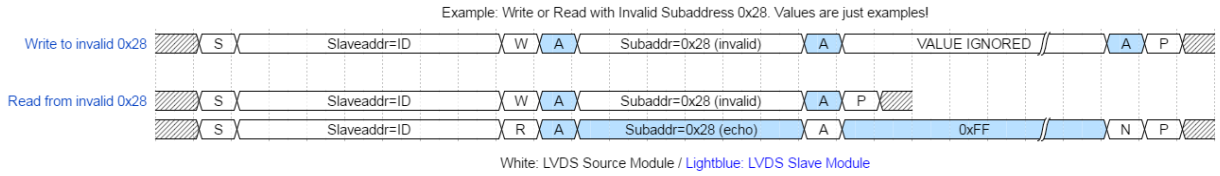
3.6.2.9.9 IFS-MMI2C-SR-REQ-140565/E-Undefined / Unsupported Subaddress

If the I²C Master attempts to write to an undefined subaddress, the I²C Slave shall:

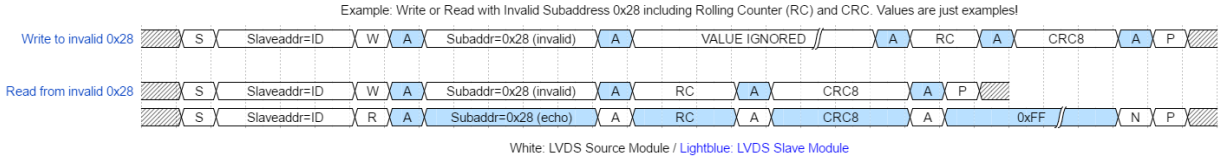
- ACK to indicate the byte was received
- Update the internal subaddress register (for echo purposes)
- Take no other action because the request was unrecognized.

If I²C Master attempts to read from an undefined subaddress, the I²C Slave shall leave SDA undriven resulting in Data = 0xFF.

For example, subaddress 0x28 is undefined. This diagram shows the I²C Master attempting to write to, and read from, the undefined subaddress:



Same example showing functional safety support:



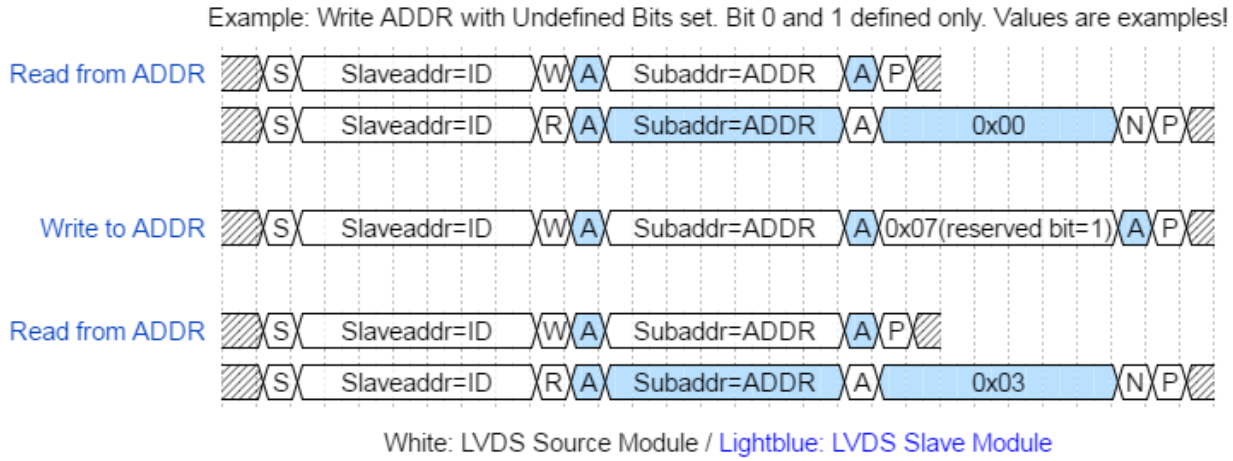
3.6.2.9.10 IFS-MMI2C-SR-REQ-140611/C-Reserved Bits

I²C Slave shall respond with “reserved” bits equal to zero.

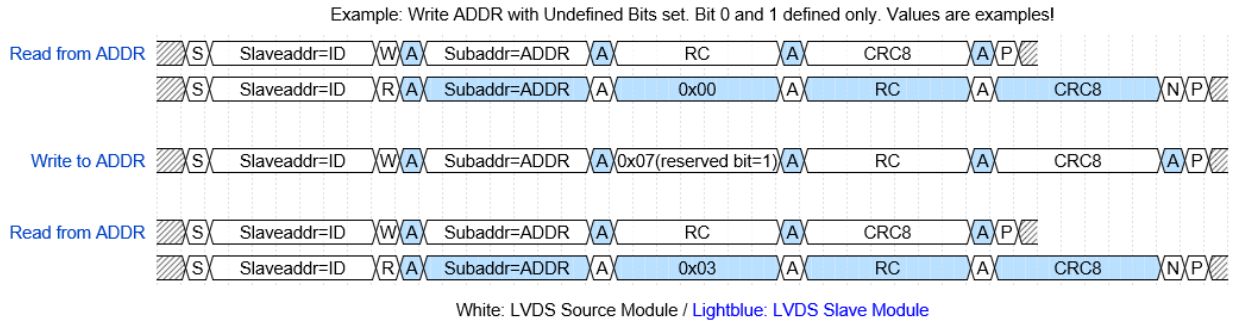
3.6.2.9.11 IFS-MMI2C-SR-REQ-140566/D-Reserved Bits

If I²C Master writes to a subaddress and any reserved bit is set, the I²C Slave shall treat the reserved bit as “don’t care” and shall act upon the defined bits.

As an example with subaddress ADDR that has several undefined bits. This diagram shows the I²C Slave changing value from 0x00 to 0x03, and ignoring a reserved bit:



Same example showing functional safety support:



3.6.2.9.12 IFS-MMI2C-SR-REQ-399913/A-Functional Safety Support

If a function has e.g. an ASIL rating for functional safety, I²C bus communication shall implement “IFS-MMI2C-SR-REQ-399914-Rolling Counter behavior” and “IFS-MMI2C-SR-REQ-399916-CRC8” requirements.

3.6.2.9.13 IFS-MMI2C-SR-REQ-399914/A-Rolling Counter behavior

The rolling counter shall be incremented by one, in each consecutive corresponding frame. As well, if all values of this frame are the same.
If the rolling counter reaches its maximum value, it shall be reset and begin with default value, again.

Hint: See “IFS-MMI2C-SR-REQ-399915-Rolling Counter default value”

Example of a rolling counter supporting CRC:

Startup rolling counter is 0. So, first rolling counter carries 0. Next will have 1, and so on until 255. For next the rolling counter is reset to default value. So, it will be 0, again.

Hint: This example assumes “IFS-MMI2C-SR-REQ-399915-Rolling Counter default value” defines default value to and size is eight bits.

3.6.2.9.14 IFS-MMI2C-SR-REQ-399915/A-Rolling Counter default value

default value for rolling counter is 0x0.

3.6.2.9.15 IFS-MMI2C-SR-REQ-399916/A-CRC8

For supporting CRC8, following requirements shall be implemented:

- "IFS-MMI2C-SR-REQ-399917-CRC8 algorithm",
- "IFS-MMI2C-SR-REQ-399918-CRC8 notation",
- "IFS-MMI2C-SR-REQ-399919-CRC8 initialization",
- "IFS-MMI2C-SR-REQ-399920-CRC8 computation",
- "IFS-MMI2C-SR-REQ-399921-CRC8 calculation",
- "IFS-MMI2C-SR-REQ-399922-CRC8 unused bits",
- "IFS-MMI2C-SR-REQ-399923-CRC8 unused bytes",
- "IFS-MMI2C-SR-REQ-399924-CRC8 data stream".

This means to add rolling counter field and CRC field to all defined messages directly behind.

3.6.2.9.16 IFS-MMI2C-SR-REQ-399919/A-CRC8 initialization

The CRC shall not return zero when the data set is all zero by initializing the CRC algorithm CRC to five (`uint8 crc = 5;--` as shown in the above algorithm).

3.6.2.9.17 IFS-MMI2C-SR-REQ-399917/A-CRC8 algorithm

Following look up table is used for 0x83 CRC algorithm:

```
uint8 CalcCRC8(uint8 data[], uint8 len)
{
uint8 crc = 5; // CRC is non-zero CRC when all data is zero
uint8 tmp;
uint8 i = 0;
// CRC Lookup Table for #0x83 = x^8 +x^2 +x +1 (0x107) <=> (0xe0; 0x1c1)
static uint8 CRC_table_0x83[256] = { // so array is not allocated on stack
0x00, 0x07, 0x0E, 0x09, 0x1C, 0x1B, 0x12, 0x15, 0x38, 0x3F, 0x36, 0x31, 0x24, 0x23, 0x2A, 0x2D,
0x70, 0x77, 0x7E, 0x79, 0x6C, 0x6B, 0x62, 0x65, 0x48, 0x4F, 0x46, 0x41, 0x54, 0x53, 0x5A, 0x5D,
0xE0, 0xE7, 0xEE, 0xE9, 0xFC, 0xFB, 0xF2, 0xF5, 0xD8, 0xDF, 0xD6, 0xD1, 0xC4, 0xC3, 0xCA, 0xCD,
0x90, 0x97, 0x9E, 0x99, 0x8C, 0x8B, 0x82, 0x85, 0xA8, 0xAF, 0xA6, 0xA1, 0xB4, 0xB3, 0xBA, 0xBD,
0xC7, 0xC0, 0xC9, 0xCE, 0xDB, 0xDC, 0xD5, 0xD2, 0xFF, 0xF8, 0xF1, 0xF6, 0xE3, 0xE4, 0xED, 0xEA,
0xB7, 0xB0, 0xB9, 0xBE, 0xAB, 0xAC, 0xA5, 0xA2, 0x8F, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9D, 0x9A,
0x27, 0x20, 0x29, 0x2E, 0x3B, 0x3C, 0x35, 0x32, 0x1F, 0x18, 0x11, 0x16, 0x03, 0x04, 0x0D, 0x0A,
0x57, 0x50, 0x59, 0x5E, 0x4B, 0x4C, 0x45, 0x42, 0x6F, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7D, 0x7A,
0x89, 0x8E, 0x87, 0x80, 0x95, 0x92, 0x9B, 0x9C, 0xB1, 0xB6, 0xBF, 0xB8, 0xAD, 0xAA, 0xA3, 0xA4,
0xF9, 0xFE, 0xF7, 0xF0, 0xE5, 0xE2, 0xEB, 0xEC, 0xC1, 0xC6, 0xCF, 0xC8, 0xDD, 0xDA, 0xD3, 0xD4,
0x69, 0x6E, 0x67, 0x60, 0x75, 0x72, 0x7B, 0x7C, 0x51, 0x56, 0x5F, 0x58, 0x4D, 0x4A, 0x43, 0x44,
0x19, 0x1E, 0x17, 0x10, 0x05, 0x02, 0x0B, 0x0C, 0x21, 0x26, 0x2F, 0x28, 0x3D, 0x3A, 0x33, 0x34,
0x4E, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5C, 0x5B, 0x76, 0x71, 0x78, 0x7F, 0x6A, 0x6D, 0x64, 0x63,
0x3E, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2C, 0x2B, 0x06, 0x01, 0x08, 0x0F, 0x1A, 0x1D, 0x14, 0x13,
0xAE, 0xA9, 0xA0, 0xA7, 0xB2, 0xB5, 0xBC, 0xBB, 0x96, 0x91, 0x98, 0x9F, 0x8A, 0x8D, 0x84, 0x83,
0xDE, 0xD9, 0xD0, 0xD7, 0xC2, 0xC5, 0xCC, 0xCB, 0xE6, 0xE1, 0xE8, 0xEF, 0xFA, 0xFD, 0xF4, 0xF3};


while (i <> len)
{
// XOR datat byte into CRC
tmp = (data[i] ^ crc);
// fetch CRC value from table
crc = CRC_table_0x83[tmp];
i++;
}
return crc;
```

3.6.2.9.18 IFS-MMI2C-SR-REQ-399918/A-CRC8 notation

CRC algorithm shall use x^8 +x^2 +x +1 (0x83 in "Koopman" notation) to calculate the 8-bit CRC of the data byte set.
It has Hamming Distance of four (HD=4) for 119 data bits.

3.6.2.9.19 IFS-MMI2C-SR-REQ-399920/A-CRC8 computation

The CRC shall be computed whenever:

- Any of the data is updated
- OR – whenever the message is transmitted. This option has less CPU load

3.6.2.9.20 IFS-MMI2C-SR-REQ-399921/A-CRC8 calculation

CRC calculation shall use all bytes before CRC byte (includes rolling counter, as well!).

3.6.2.9.21 IFS-MMI2C-SR-REQ-399922/A-CRC8 unused bits

Unused bits in the message frame shall be set to 0.

3.6.2.9.22 IFS-MMI2C-SR-REQ-399923/A-CRC8 unused bytes

Bytes with no signal/data (0x0) shall be used in CRC calculation. I.e. All 7 bytes are used in all CRC calculations.

3.6.2.9.23 IFS-MMI2C-SR-REQ-399924/A-CRC8 data stream

If CRC is not supported whole CRC8 shall be set to 0x00.

Hint: have a look in "BUTTON-SR-REQ-366706-Rolling Counter default value if CRC not supported".

3.6.2.10 I2C Messages

0x00	R	Display Status
0x01	R	Display Identification
0x02	R/W	LCD Backlight PWM Value
0x03	R/W	Display Scanning
0x04	R/W	Display Enable
0x05	R/W	Display Shutdown
0x06	R/W	Button Backlight PWM Value
0x07	R	Button Status
0x08	R	Rotary Status
0x09	R	Momentary LCD Backlight PWM Status
0x15	R	Module specific backlight capabilities
0x16	R/W	Encoded Backlight brightness Value
0x30	R	Interrupt Status Message (ISR)
0x31	R	Core Assembly
0x32	R	Delivery Assembly

0x33	R	Software Ford Part Number
0x34	R	Serial Number
0x35	R	Main Calibration Ford Part Number
0x40	R/W	Image Adjustment
0x41	R	Supplier Precalc Low Warping Table
0x42	R	Supplier Precalc Medium Warping Table
0x43	R	Supplier Precalc High Warping Table
0x44	R	Supplier EOL Low Warping Table
0x45	R	Supplier EOL Medium Warping Table
0x46	R	Supplier EOL High Warping Table
0x91	R/W	Light Ambient Sensor RAW Value
0x92	R/W	Forward Collision Warning Status
0xA0	R	Client specific High Priority Errors
0xA1	R	Client specific Medium Priority Errors
0xA2	R	Client specific Low Priority Errors
0xA3	R/W	Client specific diagnostic message
0xB0-0xFF	R/W	Reserved for Supplier

3.6.2.10.1 IFS-MMI2C-SR-REQ-140614/E-0x00 Display Status

The Display Status message provides a mechanism to transmit general I²C Client related status's back to the I²C Source Module.

Subaddress: 0x00
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	**TCERR**	**TSCERR**	**LLOSS**	**RST_RQ**	**DCERR**	**TERR**	**BLERR**	**LCDERR**
[1]	**DISPERR**	**BL_ST**	**LOWERR**	**MEDERR**	**HIGHERR**	**INIT**	**TSC_ST**	**DISP_ST**
[2]	-	-	-	-	-	-	-	-

- TCERR: Touch Connection Error (latched) (optional status)
  This bit reports latched status of the flexible printed circuit connecting PCB to Touch Panel.
  0 FPC is connected
  1 FPC is disconnected
- TSCERR: Touch Screen Controller Error (latched) (optional status)
  This bit reports latched status of touch panel controller
  0 No Fault
  1 Fault
- LLOSS: Loss of Lock (latched)
  This bit reports latched status of loss-of-lock, as indicated by the deserializer LOCK pin.
  0 Lock is established
  1 Lock is lost
- RST_RQ: Reset Request
  This bit defaults clear, and is set when the I²C Client requires a full power-cycle reset to resolve some problem.
  0 Normal operation
  1 Request is requested
- DCERR: Disconnect error (latched)
  This bit reports latched status of the flexible printed circuit connecting PCB to LCD Panel.
  0 FPC is connected
  1 FPC is disconnected
- TERR: Temperature Derating (latched)
  This bit reports latched status of temperature derating mode.
  0 Inactive
  1 Active
- BLERR: LCD Backlight Error (latched)
  This bit reports latched status of LCD backlight.
  0 No Fault
  1 Fault
- LCDERR: LCD Error (latched)
  This bit reports latched status of LCD Panel.
  0 No Fault
  1 Fault
- INIT: Display Initialized
  This bit defaults clear, and is set after the I²C Client has been enabled.
  0 I²C Client has not been enabled during this power-cycle.
  1 I²C Client has been enabled at least once during this power-cycle.
  This bit is used by the host to detect an unexpected reset. Any transition from 1 -> 0 during normal operation indicates that the I²C Client may need a complete re-initialization.

- TSC_ST: Touch Controller Status
  This bit reflects actual status. This may be different, due to delay or an error condition, from commanded value (TSC_EN).
  0 Touch Screen Controller is not ready (held in reset).
  1 Touch Screen Controller is ready for use.
  If I²C Client has no touch screen controller, report 0.

- DISP_ST: I²C Client Status (formerly Display Status)
  This bit reflects actual status. This may be different, due to delay or an error condition, from commanded value (DISP_EN).
  0 I²C Client is disabled.
  1 I²C Client is enabled.

- HIGHERR: High Priority Error Status (latched)
  This bit reports latched status if at least one of the high priority error(s) active. In some cases (dependent on definition of connected I²C Client), there exists an extra subaddress to read out more details, if I²C Source Module needs these.
  0 No high priority error.
  1 High priority error active.

- MEDERR: Medium Priority Error Status (latched)
  This bit reports latched status if at least one of the medium priority error(s) active. In some cases (dependent on definition of connected I²C Client), there exists an extra subaddress to read out more details, if I²C Source Module needs these.
  0 No high priority error.
  1 High priority error active.

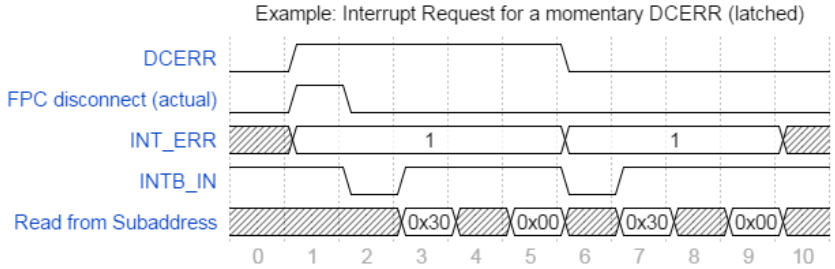
- LOWERR: Low Priority Error Status (latched)
  This bit reports latched status if at least one of the low priority error(s) active. In some cases (dependent on definition of connected I²C Client), there exists an extra subaddress to read out more details, if I²C Source Module needs these.
  0 No high priority error.
  1 High priority error active.

- BL_ST: Backlight Status
  This bit reflects actual status. This may be different, due to delay or an error condition from commanded value.
  0 Backlight is OFF (Backlight LEDs are off).
  1 Backlight is ON (Backlight LEDs are on).

- DISPERR: Display Error (latched)
  This bit reports latched status of Display Error.
  0 No Fault.
  1 Fault active.

Several bits in this I²C message have latched behavior, allowing the I²C Client to inform the host of a momentary event. The I²C Client microcontroller shall latch any value change until this subaddress is read by the host, then re-evaluate the current state.

For example, DCERR is a latched bit. This diagram shows latching behavior after a momentary FPC disconnect:



In this example:

- I²C Client detects a momentary FPC disconnect (t=1) and latches DCERR=1.
- I²C Client generates an interrupt request (t=2), and LVDS Source Module reads the latched value (t=5)
- I²C Client re-evaluates, determines FPC is now connected, and latches DCERR=0.
- I²C Client generates another interrupt request (t=6), and I²C Master reads the latched value (t=9)

3.6.2.10.2 IFS-MMI2C-SR-REQ-140615/C-0x01 Display Identification

The Display Identification message provides a mechanism to identify which kind of display is connected.

Subaddress: 0x01
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	**Display ID (see REQ-140616)**							
[1]	**Subrevision**							

- Display ID: ID of connected I²CSlave according to “IFS-MMI2C-SR-REQ-140616-Display Identification Table”.

- Subrevision: giving a more detailed information about evolution of connected I²CSlave module.
  I²C Slaves shall increase per 0x01 the subrevision on every hardware change, display micro firmware change or any compatibility break that not drives a major (means Display ID) change.
  I²C Masters should not require the use of the subrevision.
  Each Display ID begins with a subrevision of 0x00.

3.6.2.10.3 IFS-MMI2C-SR-REQ-140616/K-Display Identification Table

This table shows which display identifier shall be used by Generation 2 display:

Display ID		Display Type	Display Vendor	Disp. Size	Fu Sa	Rtr Btn	Display Orient.	Touch Panel Controller	Illumi. Strategy	Comments
0xFD*		SIMU	-	-	-	-	-	-	-	Reserved
0xFE*		SIMU	-	-	-	-	-	-	-	Reserved
0x01		SDM4	Visteon	4,2	-	Btn	Landscp.	-	PWM[1]	Initial
0x02		SDM4	Visteon	4,2	-	-	Landscp.	-	PWM[1]	Initial
0x03		SDM6	Sharp	6,5	-	-	Landscp.	mXT449T	PWM[1]	Initial
0x04		SDM8	Sharp	8,0	-	-	Landscp.	mXT641T	PWM[1]	Initial
0x05		SDM10	JDI	10,1	-	-	Landscp.	mXT1188S	PWM[1]	Initial
0x06		SDM10	JDI	10,1	-	-	Portrait	mXT1188S	PWM[1]	Initial
0x07		SDM6	Sharp	6,5	-	-	Landscp.	mXT449T	PWM[1]	w/ film on TP
0x08		SDM6	Sharp	6,5	-	-	Landscp.	mXT449T	PWM[1]	w/ modified TP glass
0x09		SDM8	Sharp	8,0	-	-	Landscp.	mXT641T	PWM[1]	w/ new TP supplier
0x0A		SDM8	Sharp	8,0	-	-	Landscp.	mXT641T	PWM[1]	w/ new TP supplier and inverted TP
0x0B		SDM6	Sharp	6,5	-	-	Landscp.	mXT449T	PWM[1]	w/ New Atmel Firmware
0x0C		SDM8	Sharp	8,0	-	-	Landscp.	mXT641T	PWM[1]	w/ New Atmel Firmware
0x0D		SDM8	JDI	8,0	-	-	Landscp.	CYAT8268X-100AS46	PWM[1]	In-cell touch
0x0E		SDM8	Sharp	8,0	-	-	Landscp.	mXT641T	PWM[1]	New LCD Module w/ carryover TP
0x0F		VDM8	Sharp	8,0	-	-	Landscp.	S7880	PWM[1]	Value DM8
0x10		SDM10	JDI	10,1	-	-	Portrait	CYAT8268X-100AS46	PWM[1]	In-cell touch
0x11		SDM12	Sharp	12,4	-	-	Landscp.	mxT1067T	PWM[1]	initial
0x12		DM12	Sharp	12,0	-	-	Landscp.	mxT1067T	PWM[1]	initial
0x13		DMD HUD0.3	tbd	0,3	-	-	Landscp.	-	RAW[1]	initial
0x14		DMD HUD0.6	tbd	0,55	-	-	Landscp.	-	RAW[1]	initial
0x15		TFT HUD2	tbd	1,8	-	-	Landscp.	-	tbd	initial
0x16		TFT HUD3	tbd	3,1	-	-	Landscp.	-	tbd	initial
0x17		IPC12	tbd	12,4	-	-	Landscp.	-	tbd	initial
0x18		SDM15	Preh	15,5	-	Cntr	Portrait	mxT2912TD	PWM[1]	initial
0x19		SDM10	JDI	10,1	-	-	Landscp.	CYAT8268X-100AS46	PWM[1]	In-cell touch
0x1A		MD23	JDI	23,4	-	-	Landscp.	-	PWM[1]	initial

0x1B		SDM15	JDI	15,5	-	-	Landscp.	CYAT827AZ S59-3200A	PWM[1]	initial
0x1C		DM12	Sharp	12,0	-	-	Portrait	mXT1067TD	PWM[1]	initial
0x1D		SDM15	JDI	15,5	-	Rtr	Portrait	CYAT827AZ S59-3200A	PWM[1]	initial
0x1E		DM12	Sharp	12,0	-	Rtr	Landscp.	mXT2912TD	PWM[1]	initial
0x1F		SDM13	Sharp	13,2	-	-	Lettersc.	TD7800	PWM[1]	initial
0x20		SDM10	JDI	10,1	-	-	Portrait	TD7800	PWM[1]	initial
0x21		SX12	JDI	12,3	Y	-	Landscp.	-	PWM[1]	initial
0x22		DM12	Sharp	12,0	-	-	Portrait	TD7850	PWM[1]	initial
0x23		LX12	JDI	12,4	-	-	Landscp.	-	PWM[1]	initial
0x24		DM12	AUO	12,0	-	-	Landscp.	TD7800	PWM[1]	initial
0x25		DM12	AUO	12,0	-	-	Landscp.	TD7800	PWM[1]	initial
0x26		SDM27	tbd	27,2	-	-	Landscp.	-	tbd	initial
0x27		SDM23	AUO	23,6	Y	-	Landscp.	-	PWM[1]	initial
0x28		DMD EDM0.2	Panasonic	0,2	-	-	-	-	ENC[1]	initial
0x29		SDM10	Viseton	10,1	-	-	Landscp	HX83192-A	PWM[1]	initial
0x2A		SDM10	Viseton	10,1	-	-	Landscp	HX83192-A	PWM[1]	initial
0x2B		LX12	Sharp	12,4	-	-	Landscp	-	PWM[1]	initial
0x2C		SDM13	Visteon/ BOE	13,2	-	-	Landscp	TD7800	PWM[1]	initial
0x2D		SDM13	Sharp	13,2	-	-	Letterscp	TD7800	PWM[1]	initial
0x2E		SDM13	AUO	13,2	Y	-	Letterscp	TD7800	PWM[1]	initial
0x2F		SDM11	Sharp	11,1	Y	-	Letterscp	TD7850	PWM[1]	initial
0x30		SDM11	Sharp	11,1	Y	-	Letterscp	TD7850	PWM[1]	Rotated 180°
0x31		SDM11	Tianma / Adayo	11,1	tbd	-	Letterscp	CYAT81688-AS77	PWM[1]	initial
0x32		DM12	Sharp	12,0	-	-	Landscp	TD7850	PWM[1]	initial
0x33		DM12	Sharp	12,0	tbd	-	Landscp	TD7850	PWM[1]	initial
0x34		LX12	Sharp	12,4	Y	-	Landscp	-	PWM[1]	Rotated 180°
0x35		SDM13	Sharp	13,2	Y	-	Letterscp	TD7800	PWM[1]	initial
0x36		LX12	JDI	12,4	Y	-	Landscp	-	PWM[1]	initial
0x37		SDM8	Sharp	8,0	-	-	Landscp	mXT641T	PWM[1]	New TP config

Note: Each display (ex. SDM6) may have multiple variants, each with a different display identifier. This is because the variants have physical differences that require e.g. a different touch calibration file.

Note*: Display IDs 0xFD and 0xFE do not need to be implemented by suppliers. However, shall not cause any problems.

Note 1: For illumination strategy:

- in case of “PWM” use requirement for 0x02 LCD Backlight PWM Value (see “IFS-MMI2C-SR-REQ-140617)”

- in case of “ENC” use requirement “0x16 Encoded Backlight brightness Value” (see “IFS-MMI2C-SR-REQ-312337).
- in case of “RAW” use requirement “0x91 Light Ambient Sensor RAW Value” (see “IFS-MMI2C-SR-REQ-323568). Please keep in mind, this a proprietary, not supported and private interface and not recommended to use!

3.6.2.10.4 IFS-MMI2C-SR-REQ-140617/C-0x02 LCD Backlight PWM Value

The LCD Backlight PWM message contains the brightness information for a 10 bit display backlight PWM generator. The PWM generator should use the complete range and resolution of 1024 steps with 0x000 = off and 0x3FF = 100% on.

Subaddress: 0x02
Access: Read-Write
Default Value: {0x00, 0x00}

	7	6	5	4	3	2	1	0
[0]	**BL_PWM[7:0]**							
[1]	**-**	-	-	-	-	**-**	**BL_PWM[9:8]**	

The SDM shall set a **default value** of 0x00 to the PWM message until set by the LVDS Source Module.
Display illumination shall be turned on/off with display enable/disable (DISP_EN).
Reference Illumination Specification for how to use.

Hint: If read, it returns set-point value.

3.6.2.10.5 IFS-MMI2C-SR-REQ-140618/B-0x03 Display Scanning

The Display Scanning message provides a mechanism to control the LCD scanning direction.

Subaddress: 0x03
Access: Read-Write
Default Value: 0x00

	7	6	5	4	3	2	1	0
[0]	-	-	**-**	**-**	**-**	**-**	**VSD**	**HSD**

- VSD: Vertical Scanning Direction.
  0 Top to Bottom
  1 Bottom to Top (engineering test-use only)
- HSD: Horizontal Scanning Direction
  0 Left to Right
  1 Right to Left (engineering test-use only)

Display shall adjust the VRV / HRV pins on the LCD panel according to the value in this subaddress.

Display supplier shall ensure that video is oriented correctly, with default value 0x00, when the display is installed in the vehicle. LVDS Source Module is not responsible to set this subaddress based on the orientation of the LCD Panel. The capability is provided only for engineering test.

3.6.2.10.6 IFS-MMI2C-SR-REQ-140619/C-0x04 Display Enable

The Display Enable message provides a mechanism for the I²C Master to tell the I²C Slave to enable the display output.

Subaddress: 0x04
Access: Read-Write
Default Value: 0x00

	7	6	5	4	3	2	1	0
[0]	-	-	**-**	**-**	**-**	**-**	**TSC_EN**	**DISP_EN**

- TSC_EN: Touch Screen Controller Enable
  0 Command touch screen controller disabled
  1 Command touch screen controller enabled
- DISP_EN: Display Enable.
  Note: This controls both the LCD Panel and the LCD Backlight.
  0 Command display disabled
  1 Command display enabled

This subaddress sets and reports the commanded status. The actual status (TSC_ST / DISP_ST) may be different, due to delay or an error condition.

The I²C Master shall not attempt to enable unless it is driving a valid pixel clock and video signal.

If the I²C Slave detects a loss of LOCK while enabled, the I²C Slave shall take any action necessary to prevent visible video problems. This may include disabling the backlight. If LOCK is re-established the I²C Slave shall take any steps necessary to resume showing video. This may include reset of the LCD panel and re-enabling the backlight.

If the Enable Display Output bit is set to enabled during a controlled shutdown the I²C Slave shall ignore this bit and complete the shutdown.

3.6.2.10.7 IFS-MMI2C-SR-REQ-140620/C-0x05 Display Shutdown

The Display Shutdown message provides a mechanism for the I²C Master to tell the I²C Slave that it will remove power from the I²C Slave and it should perform a controlled shutdown.

Subaddress: 0x05
Access: Read-Write
Default Value: 0x00

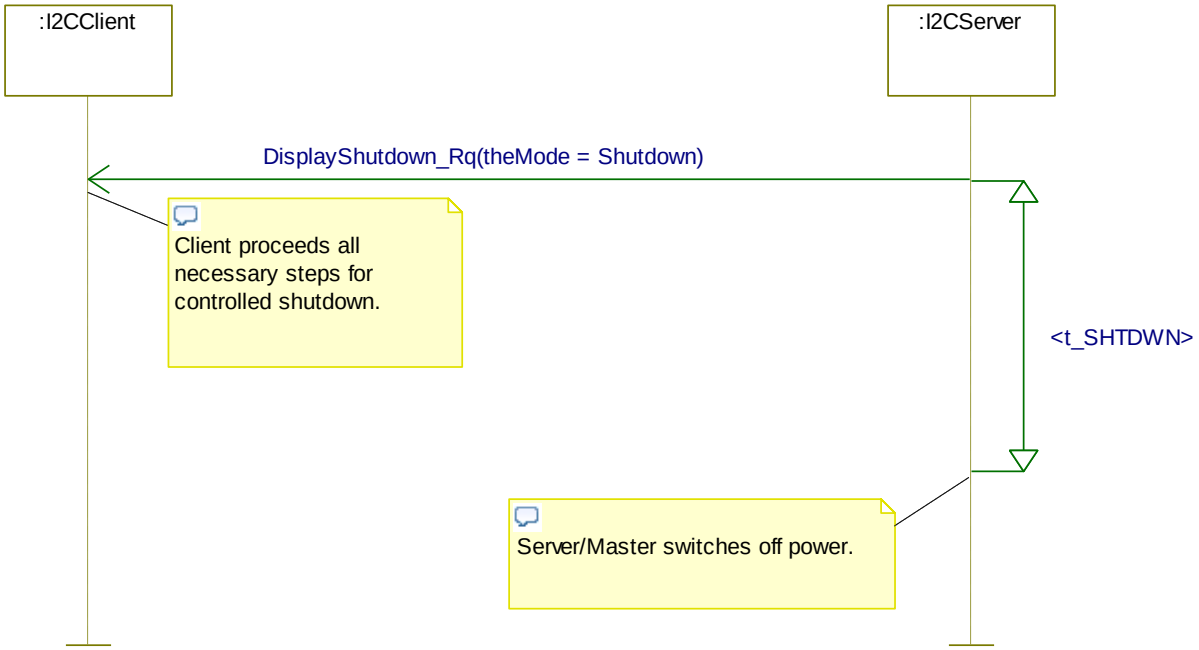
	7	6	5	4	3	2	1	0
[0]	-	-	-	-	-	-	-	**SHDWN**

- SHDWN: Display Shutdown
  - 0 Normal operation.
  - 1 Command display module to perform controlled shutdown.

After sending a controlled shutdown request, the I²C Master shall not remove power or stop driving video signal (including pixel clock) before t_{SHTDWN} expires. After the timer expires, the I²C Master shall remove power.

During an uncontrolled shutdown (i.e. battery removal, etc.) t_{SHTDWN} does not apply.

To enable function again, the I²C Master must perform the normal power initialization sequence.



3.6.2.10.8 IFS-MMI2C-SR-REQ-140621/B-0x06 Button Backlight PWM Value

The Display Button Backlight PWM message contains the brightness information for an 8-bit display backlight PWM generator. The PWM generator should use the complete range and resolution of 256 steps with 0x00 = off and 0xFF = 100% on.

Subaddress: 0x06
Access: Read-Write
Default Value: 0x00

	7	6	5	4	3	2	1	0
[0]	**BL_PWM[7:0]**							

- BL_PWM: Button Backlight PWM
  Value of the 8-bit PWM generator where 0x00 = fully off and 0xFF = fully on.

The SDM shall set a **default value** of 0x00 to the PWM message until set by the LVDS Source Module.
Button illumination shall be independent from the display enable/disable (DISP_EN).
Reference Illumination Specification for how to use.

3.6.2.10.9 IFS-MMI2C-SR-REQ-140622/B-0x07 Button Status

The Button Status message provides a mechanism to transmit button state status's back to the LVDS Source Module.

Subaddress: 0x07
Access: Read-Only
Default: n/a

	7	6	5	4	3	2	1	0
[0]	**ButtonID_A**							
[1]	**ButtonID_B**							
[2]	**ButtonID_C**							
[3]	**ButtonID_D**							
[4]	**ButtonCoding_A**		**ButtonCoding_B**		**ButtonCoding_C**		**ButtonCoding_D**	

* Reference Input Translation Matrix for ButtonID values. Reference "LVDS Button SPSS" for how to use Button ID messages and Button Coding messages.

The SDM shall report the state of each button as defined above. There shall be either a default value of '0' if button is not pressed, or a single bit with a value of '1' for the proper button state.

If multiple buttons are pressed at the same time, the SDM shall report the appropriate states for each button (concurrent button presses are allowed).

Button debounce time shall be set to $t_{button\_pressed}$ before the state is allowed to change.

State changes (not pressed to pressed & pressed to not pressed) shall be held for a minimum of $t_{button\_pressed}$ before the state is allowed to change again (hysteresis).

Hint: $t_{button\_pressed}$ is define in requirement "BUTTONv3-TMR-REQ-133003-T_reaction_time (I2C)" in Button SPSS.

3.6.2.10.10IFS-MMI2C-SR-REQ-324135/A-0x08 Rotary Status

The Rotary Status message provides a mechanism to transmit rotary turning information and status back to the I²C Master. The Display Identification table indicates which display modules support this feature.

Subaddress: 0x08
Access: Read-Only
Default: n/a

	7	6	5	4	3	2	1	0
[0]	-	-	-	-	**MT_CK**	**ERR_CK**	**PUSH_CK**	**HOR_CK**
[1]	-	-	-	-	**DETENTS_CK [3:0]**			
[2]	-	-	-	-	**MT_LK**	**ERR_LK**	**PUSH_LK**	**HOR_LK**
[3]	-	-	-	-	**DETENTS_LK [3:0]**			
[4]	-	-	-	-	**MT_RK**	**ERR_RK**	**PUSH_RK**	**HOR_RK**
[5]	-	-	-	-	**DETENTS_RK [3:0]**			

- HOR_CK: Center Knob Hand On Ring

This bit defaults clear, and is set when the user puts her/his hand onto the ring.
0 Untouched
1 Hand on Ring

- PUSH_CK: Center Knob Rotary pushed

This bit defaults clear, and is set when the user pushed the rotary.
0 Rotary released
1 Rotary pushed

- ERROR_CK: Center Knob Error on rotary

This bit defaults clear, and is set when a rotary error is detected.
0 Rotary ok
1 Rotary error or rotary unavailable

- MT_CK: Center Knob Middle Touched

This bit defaults clear, and is set when the user touches in the middle/center of the ring.
0 Untouched
1 Center/Middle of knob touched

- DETENTS_CK: Center Knob cumulated amount of rotary detents (steps)

These 4 Bits showing cumulated amount of rotary turning detents since last read of I2C Master.
Please keep in mind 0x7 = “0”

0x0-0x06	-7..-1	(7 detents in direction to left up to 1 detent in direction to left)
0x7	0	(inactive, means no detents)
0x8-0xD	+1..+7	(7 detents in direction to right up to 1 detent in direction to right)
0xE	Reserved	
0xF	Invalid	

- HOR_LK: Left Knob Hand On Ring

This bit defaults clear, and is set when the user puts her/his hand onto the ring.
0 Untouched
1 Hand on Ring

- PUSH_LK: Left Knob Rotary pushed

This bit defaults clear, and is set when the user pushed the rotary.
0 Rotary released
1 Rotary pushed

- ERROR_LK: Left Knob Error on rotary

This bit defaults clear, and is set when a rotary error is detected.
0 Rotary ok
1 Rotary error or rotary unavailable

- MT_LK: Left Knob Middle Touched

This bit defaults clear, and is set when the user touches in the middle/center of the ring.
0 Untouched
1 Center/Middle of knob touched

- DETENTS_LK: Left Knob cumulated amount of rotary detents (steps)

These 4 Bits showing cumulated amount of rotary turning detents since last read of I2C Master.
Please keep in mind 0x7 = “0”

0x0-0x06	-7..-1	(7 detents in direction to left up to 1 detent in direction to left)
0x7	0	(inactive, means no detents)
0x8-0xD	+1..+7	(7 detents in direction to right up to 1 detent in direction to right)
0xE	Reserved	
0xF	Invalid	

- HOR_RK: Right Knob Hand On Ring

This bit defaults clear, and is set when the user puts her/his hand onto the ring.
0 Untouched
1 Hand on Ring

- PUSH_RK: Right Knob Rotary pushed

This bit defaults clear, and is set when the user pushed the rotary.
0 Rotary released
1 Rotary pushed

- ERROR_RK: Right Knob Error on rotary

This bit defaults clear, and is set when a rotary error is detected.
0 Rotary ok
1 Rotary error or rotary unavailable

- MT_RK: Right Knob Middle Touched

This bit defaults clear, and is set when the user touches in the middle/center of the ring.
0 Untouched
1 Center/Middle of knob touched

- DETENTS_RK: Right Knob cumulated amount of rotary detents (steps)

These 4 Bits showing cumulated amount of rotary turning detents since last read of I2C Master.
Please keep in mind 0x7 = “0”

0x0-0x06	-7..-1	(7 detents in direction to left up to 1 detent in direction to left)
0x7	0	(inactive, means no detents)
0x8-0xD	+1..+7	(7 detents in direction to right up to 1 detent in direction to right)
0xE	Reserved	
0xF	Invalid	

3.6.2.10.11IFS-MMI2C-SR-REQ-435880/A-0x09 Momentary LCD Backlight PWM Status

The Momentary LCD Backlight PWM Status message contains the brightness information for a 10 bit display backlight PWM generator. The PWM generator should use the complete range and resolution of 1024 steps with 0x000 = off and 0x3FF = 100% on.

Subaddress: 0x09
Access: Read Only
Default Value: {0x00, 0x00}

	7	6	5	4	3	2	1	0
[0]	**LCD_BL_PWM[7:0]**							
[1]	**-**	-	-	-	-	**-**	**LCD_BL_PWM[9:8]**	

The SDM shall report the state of the Momentary LCD Backlight illumination as defined above.

3.6.2.10.12IFS-MMI2C-SR-REQ-395418/A-Exponential Backlight Brightness Values

Encoded backlight brightness concept shall be supported using exponential values.

3.6.2.10.13IFS-MMI2C-SR-REQ-395419/A-Brightness Timer Function

To ensure various transition timings, brightness timer shall be implemented/supported according to “IFS-MMI2C-SR-REQ-312696-Backlight Timer Table”.

3.6.2.10.14 IFS-MMI2C-SR-REQ-312336/A-0x15 Module specific backlight capabilities

The Module specific backlight capabilities message provides a mechanism to transmit various specific parameter values back to the I²C Master.

Subaddress: 0x15
Access: Read-Only
Default: n/a

	7	6	5	4	3	2	1	0
[0]	**BR_MIN [7:0]**							
[1]	**BR_EXP**	**BR_LIN**	**BR_TIM**	**-**	**-**	**-**	**-**	**-**
[2]	**BR_MAX [7:0]**							
[3]	**BR_MAX [15:8]**							
[4]	**BR_AREA [7:0]**							
[5]	**BR_AREA [15:8]**							

- BR_MIN: Brightness minimum
  Minimum value of brightness the module is capable.
  in 0.1 cd/m² (nit)
  (Hint: is first "on" value of e-curve (see *low value* in "IFS-MMI2C-SR-REQ-312701-Encoded Backlight Lookup Table")
  example: return value of 100 means 10.0 cd/m²
- BR_MAX: Brightness maximum
  Maximum value of brightness the module is capable.
  in 1 cd/m² (nit)
  example: return value of 1200 means 1200 cd/m²
- BR_TIM: Brightness timer available
  shows if I²C Slave supports brightness timer for smooth transition.
  0 not supported
  1 supported
  if supported, all values of Backlight Timer Table must be implemented (see "IFS-MMI2C-SR-REQ-312696-Backlight Timer Table" (Backlight Timer Table))
- BR_AREA: Brightness area
  size of lighted area.
  in 1 cm² (=0.0001 m²)
  example: return value of 184 means 184 cm² (=0.0184 m²)
- BR_LIN: brightness linear supported
  Shows if linear brightness is supported.
  0 not supported
  1 supported
  Note: at least one of BR_LIN or BR_EXP must be set!
- BR_EXP: brightness exponential supported
  Shows if exponential brightness is supported.
  0 not supported
  1 supported
  Note: at least one of BR_LIN or BR_EXP must to be set!

Note: for further definition of these values see display spec

Reference Illumination Specification for how to use.

3.6.2.10.15IFS-MMI2C-SR-REQ-312337/A-0x16 Encoded Backlight brightness Value

Note: Only applicable for I²C Slaves using “ENC” illumination strategy!

The Encoded Backlight brightness message contains an 11-bit value for supporting points for a logarithmic lookup-table (“e-curve”) and a timer value for local smooth transition (if supported)

Subaddress: 0x16
Access: Read-Write
Default Value: {0x00, 0x00, 0x00}

	7	6	5	4	3	2	1	0
[0]	**BL_ENC[7:0]**							
[1]	**BL_TIM_UP[5:0]**						**BL_ENC[10:8]**	
[2]	**BL_LIN_EXP**	**BL_TIM_DN[5:0]**						

- BL_TIM_DN: Backlight Timer down
  Time for I²C Slave to dim to stated value when downwards
  see lookup table for backlight timer “IFS-MMI2C-SR-REQ-312696-Backlight Timer Table”.
- BL_TIM_UP: Backlight Timer up
  Time for I²C Slave to dim to stated value when upwards
  see lookup table for backlight timer “IFS-MMI2C-SR-REQ-312696-Backlight Timer Table”.
- BL_ENC: Backlight Encoded value (known as N in formula)
  Showing supporting point (entry point for “e-curve” lookup table; see Encoded Backlight Lookup Table in Requirement “IFS-MMI2C-SR-REQ-312701-Encoded Backlight Lookup Table”)
- BL_LIN_EXP: encoded linear or exponential
  Showing if encoded value has to be interpreted in linear or exponential way.
  0 linear encoding
  1 exponential encoding

  The I²C Slave shall set a **default value** of 0x00 to the encoded value and timer message until set by the I²C Master.
  Means: reading this message returns last set values.

  Display illumination shall be turned on/off with display enable/disable (DISP_EN).
  Reference Illumination Specification for how to use.

3.6.2.10.16 IFS-MMI2C-SR-REQ-312701/A-Encoded Backlight Lookup Table

"e-curve" lookup table shall be calculated using following formula:

0 = 0.0000
1 = *low value* (-> first non-off value, =BR_MIN)
N = *low value* * (*high value* / *low value*) ^ ((*N*-1)/2046)
2047 = *high value* (-> 100%)

low value = minimum brightness value returned by I²C Slave
(see *BR_MIN* in "IFS-MMI2C-SR-REQ-312336-0x15 Module specific backlight capabilities")
high value = maximum brightness value returned by I²C Slave
(see "IFS-MMI2C-SR-REQ-312336-0x15 Module specific backlight capabilities")
N = supporting point with value range 2<= *N* <= 2047

Note: In case I²C Slave supports only lower backlight resolution, internal reduction is allowed (-> further definition see display spec)

3.6.2.10.17IFS-MMI2C-SR-REQ-312696/A-Backlight Timer Table

The Backlight Timer Table shows a lookup table that must be implemented with all values, if I²C Slave supports local smooth transition (see “IFS-MMI2C-SR-REQ-312336-0x15 Module specific backlight capabilities”)

To ensure smooth processing please refer to “IFS-MMI2C-SR-REQ-395417-Smooth Dimming Transition Time Steps”.

For steps please use following formula: *steps = round[1] (time / internal interrupt time)*
[1]use 4/5 rounding

Setting	Time [s]	comment
0	0,000	minimum - one interrupt step
1	0,250	
2	0,500	
3	0,750	
4	1,000	
5	1,250	
6	1,500	
7	1,750	
8	2,000	
9	2,250	
10	2,500	
11	2,750	
12	3,000	
13	3,250	
14	3,500	
15	3,750	
16	4,000	
17	4,250	
18	4,500	
19	4,750	
20	5,000	
21	5,250	
22	5,500	
23	5,750	
24	6,000	
25	6,250	
26	6,500	
27	6,750	
28	7,000	
29	7,250	
30	7,500	
31	7,750	
32	8,000	
33	8,250	
34	8,500	
35	8,750	
36	9,000	
37	9,250	
38	9,500	
39	9,750	
40	10,000	
41	10,250	
42	10,500	

43	10,750
44	11,000
45	11,250
46	11,500
47	11,750
48	12,000
49	12,250
50	12,500
51	12,750
52	13,000
53	13,250
54	13,500
55	13,750
56	14,000
57	14,250
58	14,500
59	14,750
60	15,000
61	15,250
62	15,500
63	15,750

For time values, round to closest possible time step.

3.6.2.10.18IFS-MMI2C-SR-REQ-395417/A-Smooth Dimming Transition Time Steps

To ensure smooth processing/transition the internal time steps shall be <= 20ms.

3.6.2.10.19 IFS-MMI2C-SR-REQ-140623/C-0x30 Interrupt Status Register

The Interrupt Status message provides a mechanism to check the reason of pulling the interrupt line. For that the LVDS Source Module requests this message to transmit the interrupt reason back to the LVDS Source Module.

Subaddress: 0x30
Access: Read-Only
Default: n/a

	7	6	5	4	3	2	1	0
[0]	-	-	**-**	**-**	**INT_ROT**	**INT_TCH**	**INT_BTN**	**INT_ERR**

- INT_TCH: Touch Interrupt
  Mirrors Atmel CHG signal.
  1 CHG is asserted
  0 CHG is deasserted
  Returns =0 if the display does not support touch.
- INT_BTN: Button Interrupt
  Set on button event: press, release, or repeated transmission.
  Cleared on reading subaddress 0x07.
  Returns =0 if the display does not support buttons.
- INT_ERR: Display Status Interrupt
  Set on display status change; any bit in subaddress 0x00 changing 0->1 or 1->0.
  Cleared on reading subaddress 0x00.
- INT_ROT: Rotary Interrupt
  Set on Rotary event; any change in bit in subaddress 0x08 changing 0->1 or 1->0 or change in DETENTS for any Rotary Knob.
  Cleared on reading subaddress 0x08.
  Returns =0 if the display does not support any rotary.

The display shall generate an interrupt request whenever an interrupt-generating event occurs. An interrupt-generating event is defined as any event that causes a bit in this register to transition from 0 -> 1.

Note: The LVDS chipset does not mirror interrupt status; it only asserts INTB on a falling-edge of INTB_IN. This system uses an edge-triggered interrupt request. Refer to REQ-197941 for requirements about driving INTB_IN to make an interrupt request.

3.6.2.10.20IFS-MMI2C-SR-REQ-140624/C-0x31 Core Assembly FPN

The I²C Slave Core Assembly message provides a mechanism to transmit a Ford Part Number back to the I²C Master.

Subaddress: 0x31
Access: Read-Only
Default: n/a

	7	6	5	4	3	2	1	0
[0]	Core Assembly character[0]							
…	…							
[24]	Core Assembly character[24]							

- Core Assembly: Released (or prototype) Ford Part Number
  Null-terminated string. For example “H1BT-14F180-FA”.
  Maximum length 24 characters plus NULL.

The I²C Master shall read a maximum of 25 bytes, be robust to receiving non-ASCII bytes, and be robust to receiving non-NULL terminated data.

If the I2C Slave is not released with this kind of Ford Part Number, the I²C Slave shall indicate that the subaddress is unsupported as described in REQ-140565. In this case the I²C Slave would leave SDA undriven resulting in Data = 0xFF.

3.6.2.10.21IFS-MMI2C-SR-REQ-140625/C-0x32 Delivery Assembly FPN

The Delivery Assembly message provides a mechanism to transmit a Ford Part Number back to the I²C Master.

Subaddress: 0x32
Access: Read-Only
Default: n/a

	7	6	5	4	3	2	1	0
[0]	Delivery Assembly FPN character[0]							
…	…							
[24]	Delivery Assembly FPN character[24]							

- Delivery Assembly FPN: Released (or prototype) Ford Part Number
  Null-terminated string. . For example “H1BT-18B955-FA”
  Maximum length 24 characters plus NULL.

The I²C Master shall read a maximum of 25 bytes, be robust to receiving non-ASCII bytes, and be robust to receiving non-NULL terminated data.

If the I²C Slave is not released with this kind of Ford Part Number, the I²C Slave shall indicate that the subaddress is unsupported as described in REQ-140565. In this case the I²C Slave would leave SDA undriven resulting in Data = 0xFF.

3.6.2.10.22IFS-MMI2C-SR-REQ-140626/C-0x33 Software FPN

The Software Part Number message provides a mechanism to transmit a Ford Part Number back to the I²C Master.

Subaddress: 0x33
Access: Read-Only
Default: n/a

	7	6	5	4	3	2	1	0
[0]	Software FPN character[0]							
…	…							
[24]	Software FPN character[24]							

- Software FPN: Released (or prototype) Ford Part Number
  Null-terminated string. For example “H1BT-14D358-FA”
  Maximum length 24 characters plus NULL.

The I²C Master shall read a maximum of 25 bytes, be robust to receiving non-ASCII bytes, and be robust to receiving non-NULL terminated data.

If the I²C Slave is not released with this kind of Ford Part Number, the I²C Slave shall indicate that the subaddress is unsupported as described in REQ-140565. In this case the I²C Slave would leave SDA undriven resulting in Data = 0xFF.

3.6.2.10.23IFS-MMI2C-SR-REQ-140627/D-0x34 Serial Number

The Serial Number message provides a mechanism to transmit an electronic serial number back to the I²C Master.

Subaddress: 0x34
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	Serial Number character[0]							
…	…							
[24]	Serial Number character[24]							

- Serial Number:
  Null-terminated string.
  Maximum length 24 characters plus NULL.

Note: This specification contains no functional requirement about the format of the serial number.

The I²C Master shall read a maximum of 25 bytes, be robust to receiving non-ASCII bytes, and be robust to receiving non-NULL terminated data.

If the I²C Slave contains no serial number, the I²C Slave shall indicate that the subaddress is unsupported as described in REQ-140565. In this case the I²C Slave would leave SDA undriven resulting in Data = 0xFF.

3.6.2.10.24IFS-MMI2C-SR-REQ-140628/C-0x35 Main Calibration Data FPN

The Main Calibration Data message provides a mechanism to transmit a Ford Part Number back to the I²C Master.

Subaddress: 0x35
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	Main Calibration Data FPN character[0]							
…	…							
[24]	Main Calibration Data FPN character[24]							

- Main Calibration Data FPN: Released (or prototype) Ford Part Number
  Null-terminated string. No example provided.
  Maximum length 24 characters plus NULL.

The I²C Master shall read a maximum of 25 bytes, be robust to receiving non-ASCII bytes, and be robust to receiving non-NULL terminated data.

If the I²C Slave is not released with this kind of Ford Part Number, the I²C Slave shall indicate that the subaddress is unsupported as described in REQ-140565. In this case the I²C Slave would leave SDA undriven resulting in Data = 0xFF.

3.6.2.10.25IFS-MMI2C-SR-REQ-307237/A-0x40 Image Adjustment

The Image Adjustment message provides a mechanism for the I²C Master to tell I²C Slave where to move the image and increase/decrease brightness of the image (e.g. related to user settings).

Subaddress: 0x40
Access: Read-Write
Default Value: 0xFFFFFFFF

	7	6	5	4	3	2	1	0
[0]	**Horizontal Position**							
[1]	**Vertical Position**							
[2]	**Rotation**							
[3]	**Brightness**							

- Horizontal Position: coordinate in steps of horizontal position of the image
  0x00-0xFD: valid range
  0xFE: invalid
  0xFF: no change (of horizontal position)

- Vertical Position: coordinate in steps of vertical position of the image
  0x00-0xFD: valid range
  0xFE: invalid
  0xFF: no change (of vertical position)

- Rotation: number of steps to rotate image
  0x00-0xFD: valid range
  0xFE: invalid
  0xFF: no change (of rotation)

- Brightness: number of steps to adjust brightness
  0x00-0xFD: valid range
  0xFE: invalid
  0xFF: no brightness change

3.6.2.10.26 IFS-MMI2C-SR-REQ-306750/A-0x41 Supplier Precalc Low Warping Table

The Supplier Precalculated Low Warping Table message provides a mechanism to get a data for distortion adjustment of the image from I²C Slave.

Subaddress: 0x41
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0] [1] [2] [3]	**Check field**							
[4] [5] [6] [7]	**Number of Bytes of Data**							
[8] […] […]	**Data** **……..**							

- Check field:
  Containing data for checking if content of data field is valid (XPY format)

- Number of Bytes of Data:
  Length of following data area in bytes

- Data:
  Table of warping/distortion data

3.6.2.10.27 IFS-MMI2C-SR-REQ-307232/A-0x42 Supplier Precalc Medium Warping Table

The Supplier Precalculated Medium Warping Table message provides a mechanism to get a data for distortion adjustment of the image from I^2C Slave.

Subaddress: 0x42
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0] [1] [2] [3]	**Check field**							
[4] [5] [6] [7]	**Number of Bytes of Data**							
[8] […] […]	**Data** **……..**							

- Check field:
  Containing data for checking if content of data field is valid (XPY format)

- Number of Bytes of Data:
  Length of following data area in bytes

- Data:
  Table of warping/distortion data

3.6.2.10.28 IFS-MMI2C-SR-REQ-307233/A-0x43 Supplier Precalc High Warping Table

The Supplier Precalculated High Warping Table message provides a mechanism to get a data for distortion adjustment of the image from I²C Slave.

Subaddress: 0x43
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0] [1] [2] [3]	**Check field**							
[4] [5] [6] [7]	**Number of Bytes of Data**							
[8] […] […]	**Data** **……..**							

- Check field:
  Containing data for checking if content of data field is valid (XPY format)

- Number of Bytes of Data:
  Length of following data area in bytes

- Data:
  Table of warping/distortion data

3.6.2.10.29 IFS-MMI2C-SR-REQ-307234/A-0x44 Supplier EOL Low Warping Table

The Supplier End Of Line Low Warping Table message provides a mechanism to get a data for distortion adjustment of the image from I²C Slave.

Subaddress: 0x44
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0] [1] [2] [3]	**Check field**							
[4] [5] [6] [7]	**Number of Bytes of Data**							
[8] […] […]	**Data** **……..**							

- Check field:
  Containing data for checking if content of data field is valid (XPY format)

- Number of Bytes of Data:
  Length of following data area in bytes

- Data:
  Table of warping/distortion data

3.6.2.10.30 IFS-MMI2C-SR-REQ-307235/A-0x45 Supplier EOL Medium Warping Table

The Supplier End Of Line Medium Warping Table message provides a mechanism to get a data for distortion adjustment of the image from I^2C Slave.

Subaddress: 0x45
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0] [1] [2] [3]	**Check field**							
[4] [5] [6] [7]	**Number of Bytes of Data**							
[8] […] […]	**Data** **……..**							

- Check field:
  Containing data for checking if content of data field is valid (XPY format)

- Number of Bytes of Data:
  Length of following data area in bytes

- Data:
  Table of warping/distortion data

3.6.2.10.31 IFS-MMI2C-SR-REQ-307236/A-0x46 Supplier EOL High Warping Table

The Supplier End Of Line High Warping Table message provides a mechanism to get a data for distortion adjustment of the image from I²C Slave.

Subaddress: 0x46
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0] [1] [2] [3]	**Check field**							
[4] [5] [6] [7]	**Number of Bytes of Data**							
[8] […] […]	**Data** **……..**							

- Check field:
  Containing data for checking if content of data field is valid (XPY format)

- Number of Bytes of Data:
  Length of following data area in bytes

- Data:
  Table of warping/distortion data

3.6.2.10.32 IFS-MMI2C-SR-REQ-323568/A-0x91 Light Ambient Sensor RAW Value

Note: Only applicable for I²C Slaves using “RAW” illumination strategy!

The RAW values light sensor message contains an 8-Bit raw value and 4 values showing a status.
Since this is not a recommended strategy to have the illumination strategy isolated and proprietary in receiving ECU, this is not fully supported or described.

Subaddress: 0x91
Access: Read-Write
Default Value: { 0x00, 0x0 }

	7	6	5	4	3	2	1	0
[0]	**BL_RAW[7:0]**							
[1]	**-**	**-**	**-**	**-**	**-**	**-**	**BL2_RAW[1:0]**	

Note: It is highly recommended NOT to use this message!

- BL_RAW: RAW value of sensor value
- BL2_RAW: status of sensor
  0x0 Null
  0x1 Low
  0x2 High
  0x3 Faulty

3.6.2.10.33IFS-MMI2C-SR-REQ-324467/A-0x92 Forward Collision Warning Status

The Forward Collision Warning Status message provides a mechanism to transmit if a Forward Collision Warning event occurs to the I²C Slave.

Subaddress: 0x92
Access: Read-Write
Default: n/a

	7	6	5	4	3	2	1	0
[0]	-	-	-	-	-	-	-	**FCW_Flag**

- FCW_Flag: Forward Collision Waning Flag

This bit defaults clear, and is set if a Forward Collision warning event occurs.
0 Off FCW visible warning event is not active
1 On FCW visible warning event is active

3.6.2.10.34IFS-MMI2C-SR-REQ-378557/A-0xA0 Client specific High Priority Errors

The Client specific High Priority Errors message provides a mechanism to read out errors of this category in more details, if needed.

Subaddress: 0xA0
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	-	-	**FAN2ERR**	**FAN1ERR**	**GRN_OVHT**	**AMB_OVHT**	**DMD_OVHT**	**DMD_PARK**

- DMD_PARK: Digital Micro-Mirror Device Parked (latched).
  0 Not Parked
  1 Still in safe Park Position (-> video not projected)
- DMD_OVHT: Digital Micro-Mirror Device Board Overheat (latched).
  0 No Overheat
  1 Overheat
- AMB_OVHT: Amber LED Junction Overheat (latched).
  0 No Overheat
  1 Overheat
- GRN_OVHT: Green LED Junction Overheat (latched).
  0 No Overheat
  1 Overheat
- FAN1ERR: Fan #1 failed (latched).
  0 No Failure
  1 Failure
- FAN2ERR: Fan #2 failed (latched).
  0 No Failure
  1 Failure

Several bits in this I²C message have latched behavior, allowing the I²C Client to inform the host of a momentary event. The I²C Client microcontroller shall latch any value change until this subaddress is read by the host, then re-evaluate the current state.

In case these are not read by the host they will be reset after shutdown, too. This means, they don't need to be stored to be available after shutdown cycle.

If a new error is evaluated, related bit in this message shall be set AND related bit of related category (here: Bit 3 for High Priority Error) in subaddress 0x00 Display Status (see: IFS-MMI2C-SR-REQ-140614-0x00 Display Status) shall be set, as well.

3.6.2.10.35IFS-MMI2C-SR-REQ-378558/A-0xA1 Client specific Medium Priority Errors

The Client specific Medium Priority Errors message provides a mechanism to read out errors of this category in more details, if needed.

Subaddress: 0xA1
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	-	**PRG_CHK**	**LOSTCOM2**	**CAL_ERR**	**SPLY_ERR**	**DMD_ERR**	**LOSTCOM1**	**VID_LOST**

- VID_LOST: Video Source Lost (latched).
  0 Video Source available
  1 Video Source Lost
- LOSTCOM1: (internal) Lost Communication (e.g. with Video Processor) (latched).
  0 Communication Active
  1 Communication Lost
- DMD_ERR: Malfunction/Error of Digital Micro-Mirrors Device (latched).
  0 No Error
  1 Error
- SPLY_ERR: Power Supply Error (latched).
  0 No Error
  1 Error
- CAL_ERR: Calibration Error (latched).
  0 No calibration/incorrect calibration
  1 Calibration Available
- LOSTCOM2: (internal) Lost Communication (e.g. with Main Processor) (latched).
  0 Communication Active
  1 Communication Lost
- PRG_CHK: program checksum error (latched).
  0 Checksum Ok
  1 Checksum Error

Several bits in this I^2C message have latched behavior, allowing the I^2C Client to inform the host of a momentary event. The I^2C Client microcontroller shall latch any value change until this subaddress is read by the host, then re-evaluate the current state.

In case these are not read by the host they will be reset after shutdown, too. This means, they don't need to be stored to be available after shutdown cycle.

If a new error is evaluated, related bit in this message shall be set AND related bit of related category (here: Bit 4 for Medium Priority Error) in subaddress 0x00 Display Status (see: IFS-MMI2C-SR-REQ-140614-0x00 Display Status) shall be set, as well.

3.6.2.10.36IFS-MMI2C-SR-REQ-378559/A-0xA2 Client specific Low Priority Errors

The Client specific Medium Priority Errors message provides a mechanism to read out errors of this category in more details, if needed.

Subaddress: 0xA2
Access: Read-Only
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	-	**B_LEDCRK**	**G_LEDCRK**	**A_LEDCRK**	**PHO_DIO**	**BLU_LEDO**	**GRN_LEDO**	**AMB_LEDO**

- AMB_LEDO: Amber LED Open (latched).
  0 Amber LED circuit Ok
  1 Amber LED has open circuit
- GRN_LEDO: Green LED Open (latched).
  0 Green LED circuit Ok
  1 Green LED has open circuit
- BLU_LEDO: Blue LED Open (latched).
  0 Blue LED circuit Ok
  1 Blue LED has open circuit
- PHO_DIO: Photo Diode Open (latched).
  0 Photo Diode circuit Ok
  1 Photo Diode has open circuit
- A_LEDCRK: Solder crack Amber LED (latched).
  0 Amber LED Ok
  1 Amber LED failure is imminent
- G_LEDCRK: Solder crack Green LED (latched).
  0 Green LED Ok
  1 Green LED failure is imminent
- B_LEDCRK: Solder crack Blue LED (latched).
  0 Blue LED Ok
  1 Blue LED failure is imminent

Several bits in this I^2C message have latched behavior, allowing the I^2C Client to inform the host of a momentary event. The I^2C Client microcontroller shall latch any value change until this subaddress is read by the host, then re-evaluate the current state.

In case these are not read by the host they will be reset after shutdown, too. This means, they don't need to be stored to be available after shutdown cycle.

If a new error is evaluated, related bit in this message shall be set AND related bit of related category (here: Bit 5 for Low Priority Error) in subaddress 0x00 Display Status (see: IFS-MMI2C-SR-REQ-140614-0x00 Display Status) shall be set, as well.

3.6.2.10.37 IFS-MMI2C-SR-REQ-408697/A-0xA3 Client Specific Diagnostic Message

The Client specific diagnostic message provides an interface for vendor defined diagnostics. This message is module specific. The format of the message shall be defined in the module's product specification. The content may be used for triage, analytics and verification of diagnostic mechanisms.

Subaddress: 0xA3
Access: Read-Write
Default Value: n/a

	7	6	5	4	3	2	1	0
[0]	DIAGNOSTIC MESSAGE							
[..]	...							

- DIAGNOSTIC MESSAGE:
  Client specific diagnostic message.

3.6.2.10.38 IFS-MMI2C-SR-REQ-140629/B-I2C Reserved Subaddresses

The read and write messages at subaddress 0xB0-0xFF shall be reserved for internal supplier uses.

3.6.3 Appendix A: Definitions / Acronyms

ADM – Auxiliary Display Module
DES – LVDS Deserializer
DM – Display Module
DMD – Digital Micro-Mirror Device
DTC – Diagnostic Trouble Code
EDM – External Display Module
ESN – Electronic Serial Number
FPC – Flexible Printed Circuit
FPN – Ford Part Number
Gen2 Cameras – e.g. Digital Rear View Camera
HUD – Head Up Display
IPC – Instrument Panel Cluster
IPMB – Image Processing Module B, Rear View Camera
ISR – Interrupt Status Register
SDM – Slim Display Module
SER – LVDS Serializer
TFT – Thin Film Transistor
TSC – Touch Screen Controller

3.6.4 Appendix B: Reference Documents

Reference #	Document Title
1	Button SPSS
2	Bezel Diagnostics SPSS
3	Hardware specification of related Module e.g. SDM, IPC, HUD
4	NXP UM102104, I2C-bus specification and user manual
5	Atmel mXT540E Protocol Guide
6	Atmel mXT641T Protocol Guide
7	TI AN-2173 I2C Communication Over FPD-Link III with Bidirectional Control Channel (Application Note)
8	TI SNLS407 DS90UB925Q (User's Guide)
9	TI SNLS422 DS90UB926Q (User's Guide)
10	Cypress Automotive TrueTouch Touch Screen Controller Technical Reference Manual

The requirements of the documents listed in the reference table above, of the latest revision level, form a part of this Engineering Specification

4 Appendix: Reference Documents

Reference #	Document Title
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

5 APPENDIX A: I2C sequence example - Startup

Startup

Sequence Description:	Gen2 sequence	Gen3 sequence
assert DISPLAY_EN_N	*gpio*	*gpio*
assert SER_PDB	*gpio*	*gpio*
wait 300ms	*wait*	*wait*
SERDES i2c passthough	*Serializer specific*	*Serializer specific*
check display id	I2C_SEND:71:[01]	I2C_SEND:71:[01][01][53]
check display id	I2C_RECV:71:[01][01][01]	I2C_RECV:71:[01][01][01][01][21]
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][02][4F]
check display status	I2C_RECV:71:[00][00][00]	I2C_RECV:71:[00][00][00][00][02][E3]
SERDES init	*Serializer / Deserializer specific / Includes Link and Lock confirmation*	*Serializer / Deserializer specific / Includes Link and Lock confirmation*
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][03][48]
check display status	I2C_RECV:71:[00][00][00]	I2C_RECV:71:[00][00][00][00][03][E4]
check display enable	I2C_SEND:71:[04]	I2C_SEND:71:[04][04][09]
check display enable	I2C_RECV:71:[04][00]	I2C_RECV:71:[04][00][04][77]
check display backlight pwm setpoint	I2C_SEND:71:[02]	I2C_SEND:71:[02][05][70]
check display backlight pwm setpoint	I2C_RECV:71:[02][00][00]	I2C_RECV:71:[02][00][00][05][79]
enable display	I2C_SEND:71:[04][03]	I2C_SEND:71:[04][03][06][46]
wait 200ms	wait	wait
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][07][54]
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][07][00][06][E9]
enable backlight - pwm setpoint	I2C_SEND:71:[02][ff][01]	I2C_SEND:71:[02][ff][01][08][64]
check display enable	I2C_SEND:71:[04]	I2C_SEND:71:[04][09][2A]
check display enable	I2C_RECV:71:[04][03]	I2C_RECV:71:[04][03][07][41]
check display backlight pwm setpoint	I2C_SEND:71:[02]	I2C_SEND:71:[02][0A][5D]
check display backlight pwm setpoint	I2C_RECV:71:[02][ff][01]	I2C_RECV:71:[02][ff][01][08][64]
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][0B][70]
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][47][00][09][42]

6 APPENDIX A: I2C sequence example - Shutdown 1

Shutdown_1

Sequence Description:	Gen2 sequence	Gen3 sequence	
shutdown	I2C_SEND:71:[05][01]	I2C_SEND:71:[05][01][0A][23]	eg rc=0A
wait 500ms	*wait*	*wait*	
deassert DISPLAY_EN_N	*gpio*	*gpio*	
deassert SER_PDB	*gpio*	*gpio*	
wait 1000ms	*wait*	*wait*	

7 APPENDIX A: I2C sequence example - Shutdown 2

Shutdown_2

Sequence Description:	Gen2 sequence	Gen3 sequence
deassert DISPLAY_EN_N	*gpio*	*gpio*
deassert SER_PDB	*gpio*	*gpio*
wait 1500ms	*wait*	*wait*

8 APPENDIX A: I2C sequence example - Recovery

Recovery

Sequence Description:	Gen2 sequence	Gen3 sequence	
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][01][46]	eg rc=01
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][47][00][01][7A]	
disable display	I2C_SEND:71:[04][00]	I2C_SEND:71:[04][00][02][65]	
wait 1000ms	*wait*	*wait*	
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][03][48]	
check display status	I2C_RECV:71:[00][00][04]	I2C_RECV:71:[00][00][04][00][02][48]	
enable display	I2C_SEND:71:[04][03]	I2C_SEND:71:[04][03][04][48]	
wait 250ms	wait	wait	
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][05][5A]	
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][47][00][03][74]	disp_st,tsc_st_bl_st

9 APPENDIX A: I2C sequence example - Touch Reset

Touch Reset

Sequence Description:	Gen2 sequence	Gen3 sequence	
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][01][46]	eg rc=01
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][47][00][01][7A]	
check display enable	I2C_SEND:71:[04]	I2C_SEND:71:[04][02][1B]	
check display enable	I2C_RECV:71:[04][03]	I2C_RECV:71:[04][03][02][5A]	
disable touch	I2C_SEND:71:[04][01]	I2C_SEND:71:[04][01][03][77]	
wait 100ms	*wait*	*wait*	
enable touch	I2C_SEND:71:[04][03]	I2C_SEND:71:[04][03][04][48]	
wait 250ms	wait	wait	
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][05][5A]	
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][47][00][03][74]	disp_st,tsc_st_bl_st

10 APPENDIX A: I2C sequence example - On demand backlight update

On demand backlight update

Sequence Description:	Gen2 sequence	Gen3 sequence
update backlight setpoint	I2C_SEND:71:[02][ff][01]	I2C_SEND:71:[02][ff][01][01][5B]
…		
update backlight setpoint	I2C_SEND:71:[02][ff][02]	I2C_SEND:71:[02][ff][02][02][6D]
…		
update backlight setpoint	I2C_SEND:71:[02][f0][00]	I2C_SEND:71:[02][f0][00][03][07]
…		
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][04][5D]
		I2C_RECV:71:[00][00][47][00][01][7A]
check display status	I2C_RECV:71:[00][00][07]	disp_st=1,tsc_st=1,bl_st=1
backlight off	I2C_SEND:71:[02][00][00]	I2C_SEND:71:[02][00][00][05][79]
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][06][53]
		I2C_RECV:71:[00][00][07][00][02][F5]
check display status	I2C_RECV:71:[00][00][07]	disp_st=1,tsc_st=1,bl_st=0

11 APPENDIX A: I2C sequence example - Interrupt service routine

Interrupt service routine

Sequence Description:	Gen2 sequence	Gen3 sequence
serializer interrupt occurs	I2C_SENDRECV:0c:S:[c7]:R:[21]	I2C_SENDRECV:0c:S:[c7]:R:[21]
check display interrupt source	I2C_SEND:71:[30]	I2C_SEND:71:[30][01][BF]
display interrupt source is display	I2C_RECV:71:[30][01]	I2C_RECV:71:[30][01][01][33]
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][02][4F]
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][47][00][02][73]
…		
serializer interrupt occurs	I2C_SENDRECV:0c:S:[c7]:R:[21]	I2C_SENDRECV:0c:S:[c7]:R:[21]
check display interrupt source	I2C_SEND:71:[30]	I2C_SEND:71:[30][01][BF]
display interrupt source is touch ic	I2C_RECV:71:[30][04]	I2C_RECV:71:[30][01][01][33]
check touch data	*touch ic specific*	*touch ic specific*

12 APPENDIX A: I2C sequence example - Periodic Check

Periodic Check

Sequence Description:	Gen2 sequence	Gen3 sequence
check display status	I2C_SEND:71:[00]	I2C_SEND:71:[00][08][79]
check display status	I2C_RECV:71:[00][00][07]	I2C_RECV:71:[00][00][47][00][06][6F]
check display enable	I2C_SEND:71:[04]	I2C_SEND:71:[04][09][2A]
check display enable	I2C_RECV:71:[04][03]	I2C_RECV:71:[04][03][07][41]
check display backlight pwm	I2C_SEND:71:[02]	I2C_SEND:71:[02][0A][5D]
check display backlight pwm	I2C_RECV:71:[02][ff][01]	I2C_RECV:71:[02][ff][01][08][64]
check serializer interrupt	I2C_SENDRECV:0c:S:[c7]:R:[21]	I2C_SENDRECV:0c:S:[c7]:R:[21]
check display interrupt source	I2C_SEND:71:[30]	I2C_SEND:71:[30][0B][89]
display interrupt source is touch ic	I2C_RECV:71:[30][04]	I2C_RECV:71:[30][01][09][0B]

13 APPENDIX B: RC and CRC example - Subaddr

Subaddr Examples:

subaddr write must contain 3 data bytes (sub address, rolling counter, crc)

Sequence Description:		Notes
check display id (write subaddr,rc,crc)	I2C_SEND:71:[01][01][53]	eg host rc = 1
check display id	I2C_RECV:71:[01][01][01][01][21]	eg display rc = 1 (eg ID=01, SR=01) host sent missing rc and missing crc.
check display status (underflow)	I2C_SEND:71:[00]	This is not accepted
check display status (but display id returned)	I2C_RECV:71:[01][01][01][02][28]	display returned display id (subaddr is still 0x01 based on last accepted value) host sent rc = 2 but missing crc.
check display status (underflow)	I2C_SEND:71:[00][02]	This is not accepted
check display status (but display id returned)	I2C_RECV:71:[01][01][01][03][2F]	display returned display id (subaddr is still 0x01 based on last accepted value)
check display status (valid)	I2C_SEND:71:[00][03][48]	host sent valid rc = 3 and crc
check display status (valid)	I2C_RECV:71:[00][00][00][00][04][F1]	display returned status

14 APPENDIX B: RC and CRC example - Missing Rolling Counter or CRC

Missing Rolling Counter or CRC Examples:

rolling counter and crc
must be present

Sequence Description:		Notes
check display enable	I2C_SEND:71:[04][01][12]	
check display enable	I2C_RECV:71:[04][00][01][6C]	disp_en, tsc_en are not set host sent missing rc and missing crc.
enable display (underflow)	I2C_SEND:71:[04][03]	This is not accepted
check display enable	I2C_SEND:71:[04][02][1B]	
check display enable	I2C_RECV:71:[04][00][02][65]	disp_en, tsc_en has not changed host sent rc = 2 but missing crc.
enable display (underflow)	I2C_SEND:71:[04][03][03]	This is not accepted
check display enable	I2C_SEND:71:[04][04][09]	
check display enable	I2C_RECV:71:[04][00][03][62]	disp_en, tsc_en has not changed
enable display	I2C_SEND:71:[04][03][05][4F]	valid rolling counter and crc
check display enable	I2C_SEND:71:[04][06][07]	
check display enable	I2C_RECV:71:[04][03][04][48]	disp_en, tsc_en are set

15 APPENDIX B: RC and CRC example - Rolling Counter

Rolling Counter Examples:

rolling counter must be different than the last rolling counter received
rolling counter will increment on each transaction, but any value different than last transaction is accepted

Sequence Description:		Notes
typical rolling counter increments		
assert DISPLAY_EN_N	*gpio*	init host rc = 0, display rc = 0
assert SER_PDB	*gpio*	
wait 300ms	*wait*	
SERDES i2c passthough	*Serializer specific*	
check display id	I2C_SEND:71:[01][01][53]	eg host rc = 1
check display id	I2C_RECV:71:[01][01][01][01][21]	eg display rc = 1 (eg ID=01, SR=01)
check display status	I2C_SEND:71:[00][02][4F]	host rc = 2
check display status	I2C_RECV:71:[00][00][00][00][02][E3]	display rc = 2
SERDES init	*Serializer / Deserializer specific / Includes Link and Lock confirmation*	
check display status	I2C_SEND:71:[00][03][48]	host rc = 3 display rc = 3
check display status	I2C_RECV:71:[00][00][00][00][03][E4]	I2C_RECV:71:[00][00][47][00][02][73]
check display enable	I2C_SEND:71:[04][04][09]	
check display enable	I2C_RECV:71:[04][00][04][77]	
check display backlight pwm	I2C_SEND:71:[02][05][70]	
check display backlight pwm	I2C_RECV:71:[02][00][00][05][79]	
enable display	I2C_SEND:71:[04][03][06][46]	host rc = 6
wait 200ms	wait	
check display status	I2C_SEND:71:[00][07][54]	host rc = 7
check display status	I2C_RECV:71:[00][00][07][00][06][E9]	display sent rc = 6
enable backlight	I2C_SEND:71:[02][ff][01][08][64]	
check display enable	I2C_SEND:71:[04][09][2A]	
check display enable	I2C_RECV:71:[04][03][07][41]	
check display backlight pwm	I2C_SEND:71:[02][0A][5D]	
check display backlight pwm	I2C_RECV:71:[02][ff][01][08][64]	
check display status	I2C_SEND:71:[00][0B][70]	
check display status	I2C_RECV:71:[00][00][47][00][09][42]	
…		
rolling counter roll over		
check display status	I2C_SEND:71:[00][FF][B2]	host sent rc = FF
check display status	I2C_RECV:71:[00][00][47][00][FF][8E]	display sent rc = FF
check display status	I2C_SEND:71:[00][00][41]	host sent rc = 0
check display status	I2C_RECV:71:[00][00][47][00][00][7D]	display sent rc = 0
…		
repeated i2c reads increment rolling counter		
check display id (no writes)	I2C_RECV:71:[01][01][01][01][21]	display sent rc = 1 (eg ID=01, SR=01)
check display id (no writes)	I2C_RECV:71:[01][01][01][02][28]	display sent rc = 2 (eg ID=01, SR=01)
check display id (no writes)	I2C_RECV:71:[01][01][01][03][2F]	display sent rc = 3 (eg ID=01, SR=01)
check display id (no writes)	I2C_RECV:71:[01][01][01][04][3A]	display sent rc = 4 (eg ID=01, SR=01)
…		
rolling counter is valid if it is different from last value		

example startup (not typical but still accepted)		
assert DISPLAY_EN_N	*gpio*	init host rc = 0, display rc = 0
assert SER_PDB	*gpio*	
wait 300ms	*wait*	
SERDES i2c passthough	*Serializer specific*	
check display id	I2C_SEND:71:[01][01][53]	eg host rc = 1
check display id	I2C_RECV:71:[01][01][01][01][21]	eg display rc = 1 (eg ID=01, SR=01) host sent rc = 0 (display sees last host rc = 01, new host rc = 00)
check display status	I2C_SEND:71:[00][00][41]	This is still accepted display sent rc = 0 (host sees last display rc = 01, new display rc = 00)
check display status	I2C_RECV:71:[00][00][00][00][00][ED]	This is still accepted
SERDES init	*Serializer / Deserializer specific / Includes Link and Lock confirmation*	
check display status	I2C_SEND:71:[00][01][46]	host sent rc = 1
check display status	I2C_RECV:71:[00][00][00][00][01][EA]	
check display enable	I2C_SEND:71:[04][00][15]	host sent rc = 0
check display enable	I2C_RECV:71:[04][00][00][6B]	
check display backlight pwm	I2C_SEND:71:[02][01][6C]	host sent rc = 1
check display backlight pwm	I2C_RECV:71:[02][00][00][01][65]	
enable display	I2C_SEND:71:[04][03][02][5A]	host sent rc = 2
wait 200ms	wait	
		host sent rc = 1 (display sees last host rc = 02, new host rc = 01)
check display status	I2C_SEND:71:[00][01][46]	This is still accepted
check display status	I2C_RECV:71:[00][00][07][00][00][FB]	
enable backlight	I2C_SEND:71:[02][ff][01][00][5C]	
check display enable	I2C_SEND:71:[04][01][12]	
check display enable	I2C_RECV:71:[04][03][01][53]	
check display backlight pwm	I2C_SEND:71:[02][00][6B]	
check display backlight pwm	I2C_RECV:71:[02][ff][01][00][5C]	
check display status	I2C_SEND:71:[00][01][46]	
check display status	I2C_RECV:71:[00][00][07][00][01][FC]	all still accepted

16 APPENDIX B: RC and CRC example - Invalid Rolling Counter

Invalid Rolling Counter Examples:

rolling counter must be different than the last rolling counter received
rolling counter will increment on each transaction, but any value different than last transaction is accepted

Sequence Description:		Notes
example startup		
assert DISPLAY_EN_N	*gpio*	init host rc = 0, display rc = 0
assert SER_PDB	*gpio*	
wait 300ms	*wait*	
SERDES i2c passthough	*Serializer specific*	
check display id	I2C_SEND:71:[01][01][53]	eg host rc = 1
check display id	I2C_RECV:71:[01][01][01][01][21]	eg display rc = 1 (eg ID=01, SR=01) host sent rc = 1 (display sees last host rc = 01, new host rc = 01)
check display status	I2C_SEND:71:[00][01][46]	This is not accepted (value ignored)
check display status (but display id returned)	I2C_RECV:71:[01][01][01][02][28]	display returned display id (subaddr is still 0x01 based on last accepted value) host sent rc = 2 (display sees last host rc = 01, new host rc = 02)
host repeats check display status	I2C_SEND:71:[00][02][4F]	This is accepted
host repeats check display status	I2C_RECV:71:[00][00][00][00][03][E4]	display sent display status
...		
example startup		
assert DISPLAY_EN_N	*gpio*	init host rc = 0, display rc = 0
assert SER_PDB	*gpio*	
wait 300ms	*wait*	
SERDES i2c passthough	*Serializer specific*	
check display id	I2C_SEND:71:[01][01][53]	eg host rc = 1
check display id	I2C_RECV:71:[01][01][01][01][21]	eg display rc = 1 (eg ID=01, SR=01)
check display status	I2C_SEND:71:[00][02][4F]	host rc = 2 display sent rc = 1 (display sees last host rc = 01, new host rc = 01)
check display status	I2C_RECV:71:[00][00][00][00][01][EA]	This is not accepted
host repeats check display status	I2C_SEND:71:[00][03][48]	host rc = 3 display sent rc = 2 (display sees last host rc = 02, new host rc = 01)
host repeats check display status	I2C_RECV:71:[00][00][00][00][02][E3]	This is accepted

17 APPENDIX B: RC and CRC example - Invalid CRC

Invalid CRC Examples:		
CRC must calculated on subaddr and data		
Sequence Description:		
		Notes
check display id	I2C_SEND:71:[01][01][53]	eg host rc = 1
check display id	I2C_RECV:71:[01][01][01][01][21]	eg display rc = 1 (eg ID=01, SR=01)
check display id	I2C_SEND:71:[01][02][5A]	
		display responds with rc=FF, crc=FF (expected crc = D5) (expected [01][01][01][FF][D5])
check display id	I2C_RECV:71:[01][01][01][FF][FF]	This is not accepted
…		
check display enable	I2C_SEND:71:[04][01][12]	
check display enable	I2C_RECV:71:[04][00][01][6C]	disp_en, tsc_en are not set rolling counter=02, crc=AA (expected crc = 5A) (expected [04][03][02][5A])
enable display	I2C_SEND:71:[04][03][02][AA]	This is not accepted
check display enable	I2C_SEND:71:[04][03][1C]	
check display enable	I2C_RECV:71:[04][00][02][65]	disp_en, tsc_en has not changed