



Research & Vehicle Technology
“Infotainment Systems Product Development”

Feature – Button Strategy

**LIN ICP Infotainment Subsystem Part
Specific Specification (SPSS)**

Version 1.1

UNCONTROLLED COPY IF PRINTED

Version Date: May 18, 2015

FORD CONFIDENTIAL



Revision History

Date	Ver	Notes	
December 8, 2014	1.0	Initial Release	
May 18, 2015	1.1	Updated Release	
		BUTTON-SR-REQ-096736/C-LongEvent	<jmyslin2 / hzubert> changed long press from 2.0 to 1.5 seconds



Table of Contents

REVISION HISTORY	2
1 ARCHITECTURAL DESIGN - LIN INTERFACE	4
1.1 <i>Button Interface Requirements - LIN</i>	4
1.1.1 BUTTONv2-IIR-REQ-096644/C-LIN BCP Button Press - Button Interface Requirements.....	4
1.1.2 BUTTON-IIR-REQ-107291/A-LIN setVolume - Button Interface Requirements.....	4
1.1.3 BUTTONv2-IIR-REQ-096643/C-LIN setVolume 2 - Button Interface Requirements.....	5
1.1.4 BUTTON-IIR-REQ-095295/E-LIN setRotarySteps - Button Interface Requirements	5
1.1.5 BUTTON-IIR-REQ-153850/A-LIN - ApplicationInformation0	6
1.1.6 BUTTON-IIR-REQ-153851/A-LIN - ApplicationInformation1	6
1.1.7 BUTTON-IIR-REQ-153852/B-LIN - ApplicationInformation2	6
1.1.8 BUTTON-IIR-REQ-153853/A-LIN - ApplicationInformation3	6
1.1.9 BUTTON-IIR-REQ-153854/A-LIN - ApplicationInformation4	6
1.1.10 BUTTON-IIR-REQ-107294/B-LIN - LINStatus	6
1.1.11 BUTTON-IIR-REQ-107295/A-LIN - IlluminationZone.....	7
1.1.12 BUTTON-IIR-REQ-107296/A-LIN - IlluminationLevel1	7
1.1.13 BUTTON-IIR-REQ-107297/A-LIN - IlluminationLevel2	7
1.1.14 BUTTON-IIR-REQ-107298/A-LIN - PartNumberXxxxx	7
1.1.15 BUTTON-IIR-REQ-107299/B-LIN - ConfigDataXxxx	7
1.1.16 BUTTON-IIR-REQ-117484/A-LIN - SerialNumberXxxx	7
1.2 <i>General Requirements</i>	8
1.2.1 BUTTON-SR-REQ-107308/A-LIN Scheduler turnaround frequency	8
1.2.2 BUTTON-SR-REQ-116454/A-LIN Scheduler turnaround default values.....	8
1.2.3 BUTTON-SR-REQ-116455/A-Reset ButtonID after button release.....	8
1.2.4 BUTTON-SR-REQ-107300/A-LimpHome state (Button Transmitter)	8
1.3 <i>BUTTONv2-FUN-REQ-095292/A-LIN Message Structure</i>	9
1.3.1 LIN - BCP (Button Control Panel) Button Press Message Structure	9
1.3.2 ICP Status Transferring.....	14
1.3.3 LIN Rotary Structure.....	14
1.4 <i>BUTTONv2-CLD-REQ-095293/A-Button Input Client (Button Transmitter) - LIN</i>	16
1.4.1 Push Button Activation - Transmitter Timing Requirements	16
1.4.2 Push / Touch Button Activation - Transmitter Functional Requirements	16
1.4.3 setRotarySteps signal (LIN)	17
1.4.4 setVolume signal (LIN)	17
1.5 <i>BUTTONv2-CLD-REQ-095294/A-Button Input Server (Button Receiver) - LIN</i>	18
1.5.1 Button Activation - Receiver Timing Requirements (LIN)	18
1.5.2 Button Activation - Receiver Functional Requirements (LIN).....	18
2 APPENDIX: REFERENCE DOCUMENTS.....	20



1 Architectural Design - LIN Interface

1.1 Button Interface Requirements - LIN

1.1.1 BUTTONv2-IIR-REQ-096644/C-LIN BCP Button Press - Button Interface Requirements

BCP_Button_Press (ICPBtnStateRotary, ICPBtnState...)	Method from the Button Input Client to the receiving modules.	<p>ButtonANameID (ICPBtnID_A) 0x0 - 0x255 see Input Translation Matrix</p> <p>ButtonAActivationState (ICPBtnCoding_A) 0x0 Inactive / Not_Pressed 0x1 Active / Pressed 0x2 ShortEvent / Pressed 0x3 ShortElapsed / Pressed 0x4 LongEvent / Pressed 0x5 Stuck 0xF Idle / Not_Pressed</p> <p>ButtonBNameID (ICPBtnID_B) 0x0 – 0x255 see Input Translation Matrix</p> <p>ButtonBActivationState (ICPBtnCoding_B) 0x0 Inactive / Not_Pressed 0x1 Active / Pressed 0x2 ShortEvent / Pressed 0x3 ShortElapsed / Pressed 0x4 LongEvent / Pressed 0x5 Stuck 0xF Idle / Not_Pressed</p> <p>ButtonCNameID (ICPBtnID_C) 0x0 – 0x255 see Input Translation Matrix</p> <p>ButtonCActivationState (ICPBtnCoding_C) 0x0 Inactive / Not_Pressed 0x1 Active / Pressed 0x2 ShortEvent / Pressed 0x3 ShortElapsed / Pressed 0x4 LongEvent / Pressed 0x5 Stuck 0xF Idle / Not_Pressed</p> <p>ButtonDNameID (ICPBtnID_D) 0x0 – 0x255 see Input Translation Matrix</p> <p>ButtonDActivationState (ICPBtnCoding_D) 0x0 Inactive / Not_Pressed 0x1 Active / Pressed 0x2 ShortEvent / Pressed 0x3 ShortElapsed / Pressed 0x4 LongEvent / Pressed 0x5 Stuck 0xF Idle / Not_Pressed</p>
---	---	---

1.1.2 BUTTON-IIR-REQ-107291/A-LIN setVolume - Button Interface Requirements



SetVolume (ICPVolumeCmd)	Method for incrementing / decrementing Volume (always used with a rotary knob)	0x0 -30 steps 0x1 -29 steps 0x2 -28 steps ... 0x1B -3 steps 0x1C -2 steps 0x1D -1 step (decrement volume) 0x1E Not Pressed / Inactive 0x1F +1 step (increment volume) 0x20 +2 steps 0x21 +3 steps ... 0x3A +28 steps 0x3B +29 steps 0x3C +30 steps
--------------------------	--	--

1.1.3 BUTTONv2-IIR-REQ-096643/C-LIN setVolume 2 - Button Interface Requirements

SetVolume (ICPVolumeCmd2)	Method for incrementing / decrementing Volume (always used with a rotary knob)	0x0 - 7 steps 0x1 - 6 steps 0x2 - 5 steps 0x3 -4 steps 0x4 -3 steps 0x5 -2 steps 0x6 -1 step (decrement volume) 0x7 Not Pressed / Inactive 0x8 +1 step (increment volume) 0x9 +2 steps 0xA +3 steps 0xB +4 steps 0xC +5 steps 0xD +6 steps 0xE +7 steps
---------------------------	--	---

1.1.4 BUTTON-IIR-REQ-095295/E-LIN setRotarySteps - Button Interface Requirements

SetRotarySteps (ICPRotary)	Method for incrementing / decrementing Tune/Rotary Browsing (always used with a rotary knob)	0x0 - 7 steps 0x1 - 6 steps 0x2 - 5 steps 0x3 -4 steps 0x4 -3 steps 0x5 -2 steps 0x6 -1 step 0x7 Not Pressed / Inactive 0x8 +1 step 0x9 +2 steps 0xA +3 steps 0xB +4 steps 0xC +5 steps 0xD +6 steps 0xE +7 steps
----------------------------	--	---

**1.1.5 BUTTON-IIR-REQ-153850/A-LIN - ApplicationInformation0**

ApplicationInformation0 (APINFO0)	Method for error reporting for the volume knob	See LIN Data Link and Physical Layer for further information (Chapter "Ford Standard Error Reporting")
-----------------------------------	--	--

1.1.6 BUTTON-IIR-REQ-153851/A-LIN - ApplicationInformation1

ApplicationInformation1 (APINFO1)	Method for error reporting for the rotary knob	See LIN Data Link and Physical Layer for further information (Chapter "Ford Standard Error Reporting")
-----------------------------------	--	--

1.1.7 BUTTON-IIR-REQ-153852/B-LIN - ApplicationInformation2

ApplicationInformation2 (APINFO2)	Method for error reporting for an under voltage error	See LIN Data Link and Physical Layer for further information (Chapter "Ford Standard Error Reporting")
-----------------------------------	---	--

1.1.8 BUTTON-IIR-REQ-153853/A-LIN - ApplicationInformation3

ApplicationInformation3 (APINFO3)	Method for error reporting for an over voltage error	See LIN Data Link and Physical Layer for further information (Chapter "Ford Standard Error Reporting")
-----------------------------------	--	--

1.1.9 BUTTON-IIR-REQ-153854/A-LIN - ApplicationInformation4

ApplicationInformation4 (APINFO4)	Method for reporting that the slave needs to be configured*	See LIN Data Link and Physical Layer for further information (Chapter "Ford Standard Error Reporting")
-----------------------------------	---	--

*not used for now and is always set to "No Configuration Needed"

1.1.10 BUTTON-IIR-REQ-107294/B-LIN - LINStatus

LINStatus (ICPLINStatus)	Method for error reporting	See LIN Data Link and Physical Layer for further information (Chapter "Ford Standard Error Reporting")
--------------------------	----------------------------	--

**1.1.11 BUTTON-IIR-REQ-107295/A-LIN - IlluminationZone**

IlluminationZone (DSPIlluZone)	Method for the active zones to illuminate	See LIN Illumination Specifications for further information
--------------------------------	---	---

1.1.12 BUTTON-IIR-REQ-107296/A-LIN - IlluminationLevel1

IlluminationLevel1 (DSPIlluLv1)	Method for Value for the 8-bit button backlight PWM generator.	See LIN Illumination Specifications for further information
---------------------------------	--	---

1.1.13 BUTTON-IIR-REQ-107297/A-LIN - IlluminationLevel2

IlluminationLevel2 (DSPIlluLv2)	Method for Value for the 8-bit knob backlight PWM generator.	See LIN Illumination specifications for further information
---------------------------------	--	---

1.1.14 BUTTON-IIR-REQ-107298/A-LIN - PartNumberXxxxx

PartNumberXxxxx (ICPPartNumXxxxx) Ex. ICPPartNumIndex, ICPPartNumData3,...	Method for transferring part number. See LIN Data Link and Physical Layer specification for further information.	See LIN Data Link and Physical Layer specification for further information.
---	--	---

1.1.15 BUTTON-IIR-REQ-107299/B-LIN - ConfigDataXxxx

ConfigDataXxxx (DSPLConfigXxxx) Ex. DSPLConfigIndex, DSPLConfigData2,...	Method for sending configuration data to the LIN slave. NOT USED.	See LIN Data Link and Physical Layer specification for further information.
---	---	---

1.1.16 BUTTON-IIR-REQ-117484/A-LIN - SerialNumberXxxx

SerialNumberXxxx (ICPSrNrDigitXX) Ex. ICPSrNrDigit00, ICPSrNrDigit00, ICPSrNrDigit12,...	Method for transferring serial number of the slave to the master for standard readout. Note: this shall be used equivalent to part number readout.	See LIN Bezel Diagnostics SPSS for further information.
---	---	---



1.2 General Requirements

1.2.1 BUTTON-SR-REQ-107308/A-LIN Scheduler turnaround frequency

It shall be configurable (e.g. via diagnosis) to send each scheduler a specified number of times (e.g. X, Y, ...) and then the next one a specified number of times (e.g. Y). After that it shall begin from start.

X, Y shall be possible to set from 0x0 to 0xF.

Note: This not applies for the configuration scheduler(s).

An example for clarification:

E.g. LIN11: X=0, LIN12: Y=1, LIN13: Z=5

This will result in: LIN11 is not sent, LIN12 will be sent 1 times and then LIN13 will be sent 5 times. After that it begins from start.

LIN11: X=0												
LIN12: Y=1												
LIN13: Z=5												

1.2.2 BUTTON-SR-REQ-116454/A-LIN Scheduler turnaround default values

LIN11: X=0,

LIN12: Y=1,

LIN13: Z=0

1.2.3 BUTTON-SR-REQ-116455/A-Reset ButtonID after button release

If a button is "released" after "pressed" and an "inactive/Not_Pressed " was sent at least once then the corresponding ICPBtnID/ButtonNameID Byte will be changed to "Idle". If the ButtonID-Code is "Idle" then the appropriate ICPBtnCoding/ButtonActivationState Nibble is a don't care and should always be "Inactive/Not_Pressed".

1.2.4 BUTTON-SR-REQ-107300/A-LimpHome state (Button Transmitter)

If a LIN Server (ex. LIN ICP) reaches the limp-home state it should:

- Activate battery saver mode
- Switch off illumination (if available)



1.3 BUTTONv2-FUN-REQ-095292/A-LIN Message Structure

1.3.1 LIN - BCP (Button Control Panel) Button Press Message Structure

1.3.1.1 LIN - Infotainment Button Press

1.3.1.1.1 BUTTONv2-SR-REQ-096645/B-LIN BCP message structure

Up to 4 infotainment push button press events can be activated simultaneously by the BCP. The LIN button press messages are periodic. The basic structure of these messages is shown below.

Note: The LIN ICP will always send the LIN Button Encoding (ex 0x1 Active, 0x2 ShortEvent...). For the receiving module of the LIN ICP button presses whether to use the LIN Button Encoding or the Button Encoding Normalization column depends on how a feature is specified. If the feature SPSS or HMI spec (ex. press and hold timer) uses pressed/not pressed then the Button Encoding Normalization column shall be used. If the SPSS or HMI spec (ex. press and hold timer) uses the LIN Button Encoding (ex 0x1 Active, 0x2 ShortEvent...) then the LIN Button Encoding values shall be used.

Button ID	LIN Button Encoding	Button Encoding Normalization
Button A Activation State	0x0 Inactive	0x0 Not Pressed
	0x1 Active	0x1 Pressed
	0x2 ShortEvent	0x2 Pressed
	0x3 ShortElapsed	0x3 Pressed
	0x4 LongEvent	0x4 Pressed
	0x5 Stuck	0x5 Stuck
Button B Activation State	0x0 Inactive	0x0 Not Pressed
	0x1 Active	0x1 Pressed
	0x2 ShortEvent	0x2 Pressed
	0x3 ShortElapsed	0x3 Pressed
	0x4 LongEvent	0x4 Pressed
	0x5 Stuck	0x5 Stuck
Button C Activation State	0x0 Inactive	0x0 Not Pressed
	0x1 Active	0x1 Pressed
	0x2 ShortEvent	0x2 Pressed
	0x3 ShortElapsed	0x3 Pressed
	0x4 LongEvent	0x4 Pressed
	0x5 Stuck	0x5 Stuck
Button D Activation State	0x0 Inactive	0x0 Not Pressed
	0x1 Active	0x1 Pressed
	0x2 ShortEvent	0x2 Pressed
	0x3 ShortElapsed	0x3 Pressed
	0x4 LongEvent	0x4 Pressed
	0x5 Stuck	0x5 Stuck

Infotainment Button Activation State Coding

BYTE X

BYTE X + 1

BYTE X + 2

BYTE X + 3

BYTE X + 4

BYTE X + 5



ICPBtnID_A (ie Button A Name ID)	ICPBtnID_B (ie Button B Name ID)	ICPBtnID_C (ie Button C Name ID)	ICPBtnID_D (ie Button D Name ID)	ICPBtnCoding_A (ie Button A Activation State)	ICPBtnCoding_B (ie Button B Activation State)	ICPBtnCoding_C (ie Button C Activation State)	ICPBtnCoding_D (ie Button D Activation State)
--	--	--	--	---	---	---	---

Note: position of x may vary from scheduler frame to scheduler frame

Infotainment Button Message Structure

1.3.1.1.2 BUTTONv2-SR-REQ-096646/B-LIN BCP message structure usage

When a button is activated it shall encode Button A first and if that position is already being used (ie pressed) shall move to the next Button position in the message. If all 4 buttons (A – D) are being used then any new button inputs shall be ignored until one of the 4 buttons are released.

Once a button press is active for a button ID (A – D) the BCP shall not move that same button press to another button ID location. For example if seek is being pressed and held and is assigned to 'Button C Name ID' it shall not change to 'Button A Name ID' while still being pressed.

The default to set Button Activation States A – D is 'Not Pressed / Inactive' unless there is a button activation event.

Example:

1. Button Module powers up and bus awake
2. Button Module sending: Button A Name ID = Inactive (Idle) AND Button Activation State = Not Pressed
3. User presses button X
4. Button Module sending: Button A Name ID = X AND Button Activation State = Pressed
5. User releases button X
6. Button Module sends: Button A Name ID = X AND Button Activation State = Not Pressed at least once.
7. Button Module sends: Button A Name ID = Inactive (Idle) AND Button Activation State = Not Pressed
8. Button Module continues to send (periodically) Button A Name ID = Inactive (Idle) AND Button Activation State = Not Pressed until the next button press. Any new button press changes Button A Name ID from Inactive (Idle) to the new button value.

1.3.1.1.3 BUTTONv2-SR-REQ-096647/B-LIN Multiple stuck buttons in BCP message structure

If all 4 BCP buttons A – D encoded in the BCP_Button_Press message are determined to be 'stuck' (not including setVolume / setRotarySteps signals – Byte 7 and 8) then one of the 4 button bytes (Button A – D) shall be released so that other buttons can be activated.

1.3.1.2 **LIN Signal Functionality (not normalized to Pressed / Not Pressed)**

LIN signals are sent out periodically at a pre-defined set of time defined in the LDF. The receiving module may choose to utilize the LIN specific signals such as ShortEvent, ShortElapsed, LongEvent, Stuck to reduce the variability of not knowing when a button was pressed since button press could have occurred +/- (LIN periodic rate) msec from the event.

Also the normalization values of Press/Not_Pressed for encodings can be used (unless noted otherwise) but there could be variability of +/- (LIN periodic rate) msec.

1.3.1.2.1 BUTTON-SR-REQ-096734/B-Active

The Active encoding is set when a LIN button is first pressed.

1.3.1.2.2 BUTTON-SR-REQ-096735/A-ShortElapsed

The ShortElapsed button encoding is set 250 msec after the LIN button press event.



1.3.1.2.3 BUTTON-SR-REQ-096736/C-LongEvent

The LongEvent is set 1.5 seconds after the LIN button press event.

1.3.1.2.4 BUTTON-SR-REQ-096650/B-ShortEvent

The ShortEvent encoding is set if button presses are pressed quicker than ShortElapsed time. This value shall never be overwritten before it is sent out on the bus.

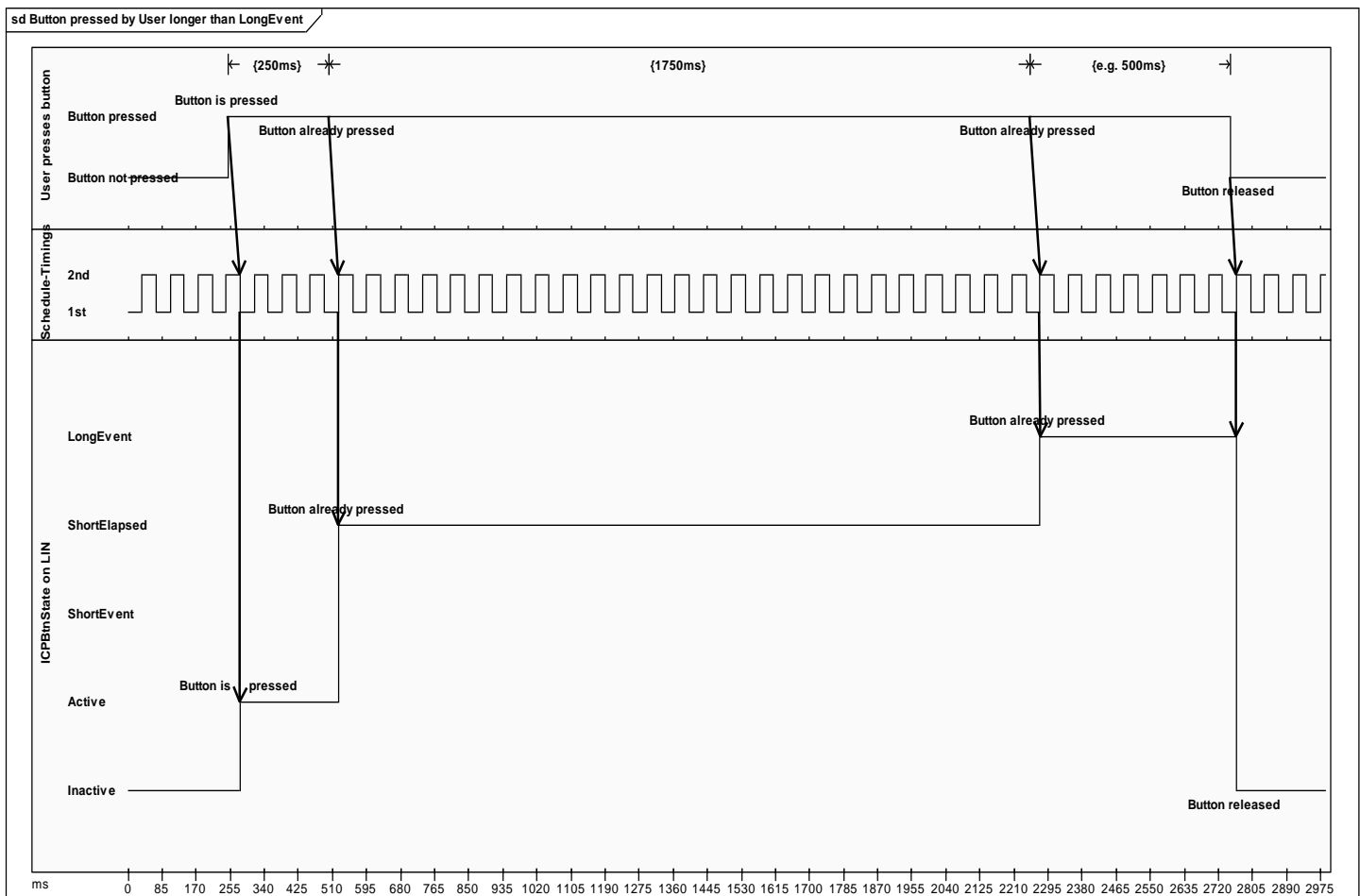
1.3.1.2.5 BUTTON-SR-REQ-096737/B-Stuck

The Stuck encoding is set 120 seconds after a press of a LIN button with no release.

Note: The receiving module of the LIN button press is the module responsible for setting the LIN stuck button DTC as defined in the IDS (infotainment diagnostic specification).

1.3.1.2.6 Button Sequences

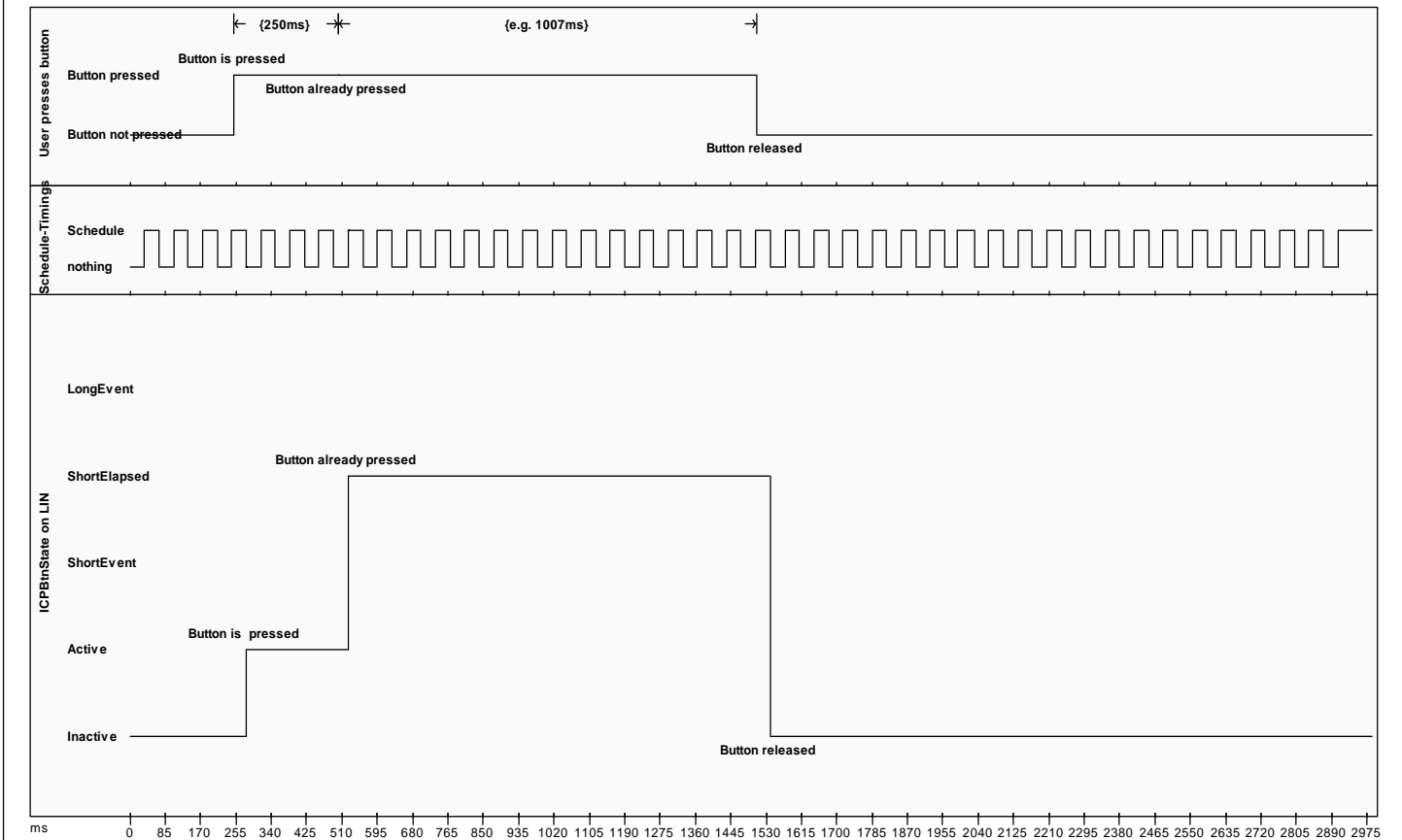
1.3.1.2.6.1 BUTTON-SR-REQ-107301/A-Button pressed by User longer than LongEvent - Button Sequence





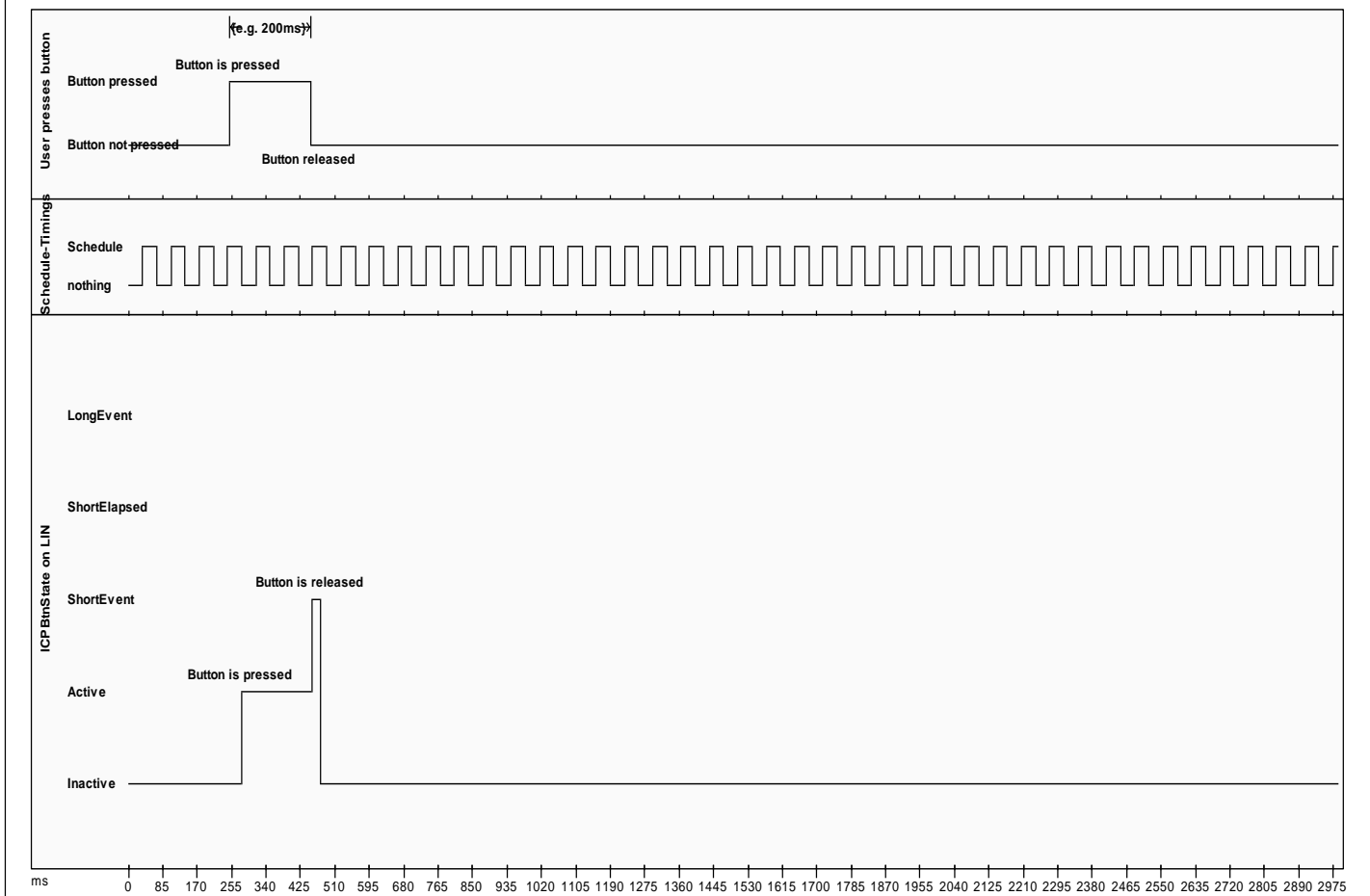
1.3.1.2.6.2 BUTTON-SR-REQ-107302/A-Button Pressed shorter than LongEvent but longer than ShortElapsed - Button Sequence

sd Button pressed shorter than LongEvent but longer than ShortElapsed



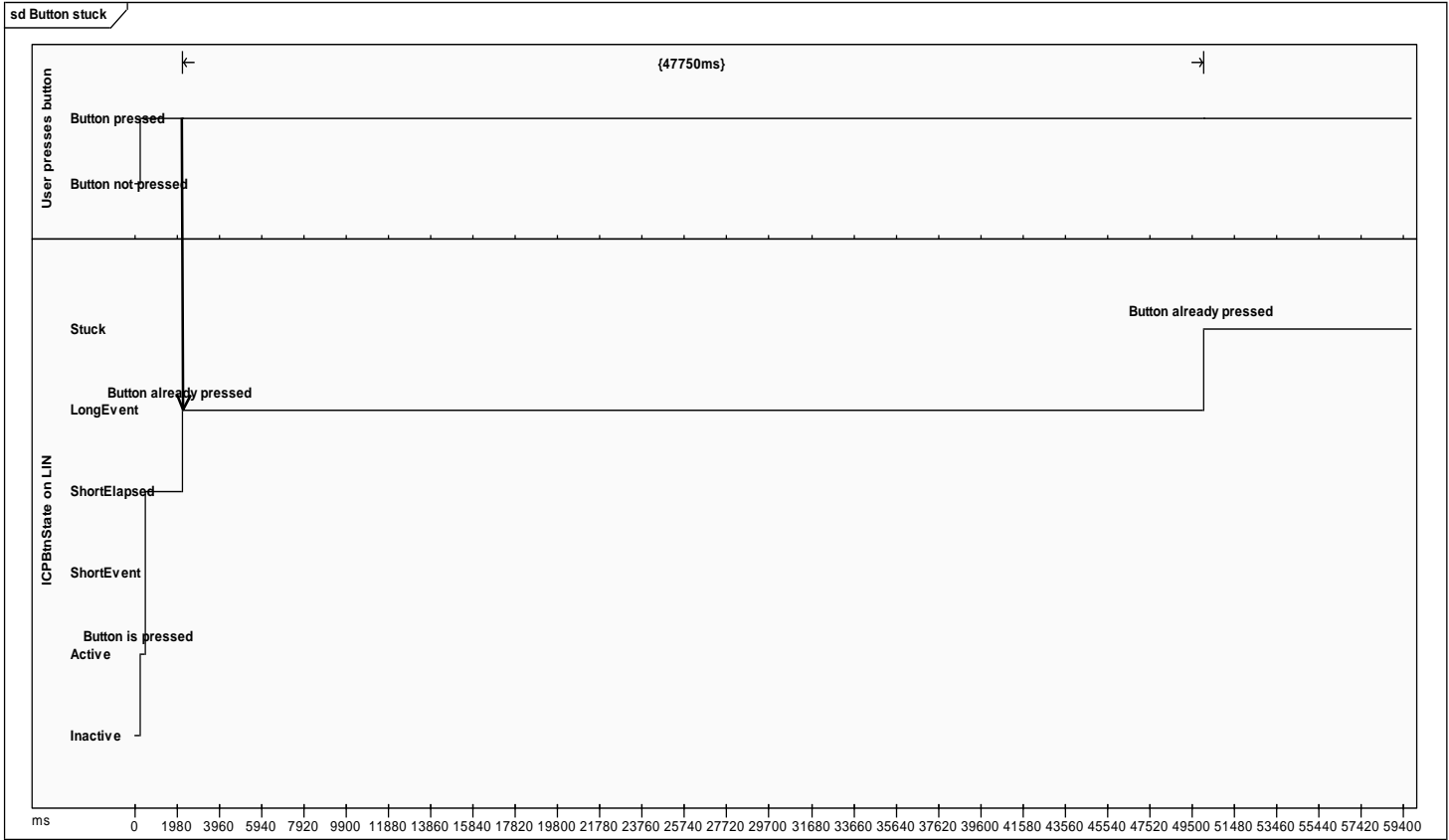
1.3.1.2.6.3 sd Button pressed shorter than ShortElapsed - Button Sequence

sd Button pressed shorter than ShortElapsed





1.3.1.2.6.4 BUTTON-SR-REQ-107304/B-Button Stuck - Button Sequence



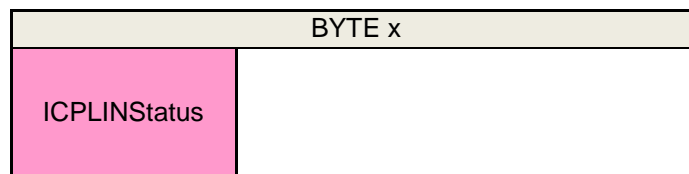
Note: this is an example only and reference the other parts of the SPSS for stuck button timer value to be used if the diagram above does not match what is called out elsewhere in the SPSS. Currently 120 seconds is used for stuck button timer.

1.3.2 ICP Status Transferring

1.3.2.1 BUTTON-SR-REQ-115765/A-LIN ICP Status Structure

This sub chapter describes how status information is transferred from the ICP to the Master. ICPLINStatus shows states or errors that are possible to detect but not supported in another way.

An example of the basic structure of this part of message is shown below:
(keep in mind this byte is not entire filled here)



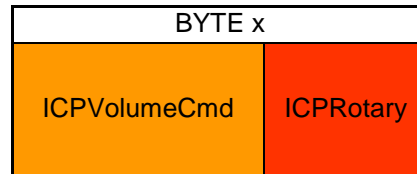
1.3.3 LIN Rotary Structure

1.3.3.1 BUTTON-SR-REQ-116456/A-LIN Rotary Structure

This requirement describes how rotary information is transferred from the ICP to the Master. Only delta counts from frame to frame will be sent.



An example of the basic structure of these parts of messages is shown below:





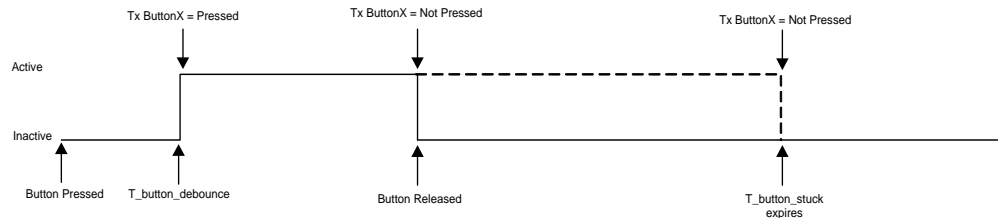
1.4 BUTTONv2-CLD-REQ-095293/A-Button Input Client (Button Transmitter) - LIN

The following sections define the Button Activation Strategy from the Transmitters perspective.

1.4.1 Push Button Activation - Transmitter Timing Requirements

1.4.1.1 BUTTONv2-SR-REQ-107305/A-Transmitter Button Activation Process timing (LIN)

The TBAP timing figure below will always remain true for any button activation event using the Pressed/Not Pressed normalization (ie not using LIN button sequences). The exception to this rule is for "Rotary Knobs" which is covered in the next section.



Transmitter Button Activation Process (TBAP) Timing

1.4.1.2 BUTTONv2-TMR-REQ-096739/B-T_reaction_time (LIN)

Name	Description	Units	Range	Resolution	Default
T_reaction_time (LIN)	The maximum transmitter reaction time from when a push button switch is closed until the push button message is put on the LIN bus.	msec	0-1000	10	70

1.4.2 Push / Touch Button Activation - Transmitter Functional Requirements

1.4.2.1 BUTTONv2-SR-REQ-107306/A-Button Pressed / Not Pressed transimission (LIN)

Once a button is pressed and debounced on the transmitter the button message will be sent to the receiver with the button signal coding set to "Pressed" (ie LongEvent, shortEvent...).

Once the button is released the transmitter will set the button coding as "Not Pressed / Inactive".

1.4.2.2 BUTTON-SR-REQ-014685/C-Button Press reaction time (TcSE ROIN-39764-1)

The transmitter (ex.EFP, ICP, SWC, SDM) reaction time from when a push button switch is closed until the push button message is put on the bus shall not exceed T_reaction_time.

Note: this does not apply to the touch sense buttons. Reference applicable specifications for touch sense debounce requirements.

1.4.2.3 BUTTONv2-SR-REQ-103693/A-Transmitter Stuck Button (LIN)

When the transmitter determines a button to be stuck:

- The transmitter will keep the button encoding status as 'Stuck' as long as the button is stuck. Upon a new ignition cycle the button encoding shall remain as 'Stuck' until the button is determined to be operational.
- Once a button becomes unstuck after previously being stuck then after remaining unstuck for T_button_unstuck the button shall become operational again.

1.4.2.4 BUTTON-TMR-REQ-014687/A-T_button_unstuck (TcSE ROIN-60370-1)

Name	Description	Units	Range	Resolution	Default
FILE:BUTTON STRATEGY LIN ICP SPSS v1.1 MAY 18 2015.DOCX	FORD MOTOR COMPANY CONFIDENTIAL The information contained in this document is Proprietary to Ford Motor Company.				Page 16 of 20



T_button_unstuck	Once a button is determined to be stuck then T_button_unstuck is the time the button has to be unstuck before the button can be operational again.	sec	0-100	1	10
------------------	--	-----	-------	---	----

1.4.3 setRotarySteps signal (LIN)

1.4.3.1 BUTTON-SR-REQ-095296/D-Tx SetRotarySteps (LIN)

The setRotarySteps signal could possibly be used to increment / decrement for the tune or fast browse function for example. Reference the applicable SPSS section for details. Each setRotarySteps step encoding shall be treated as a press event in the remainder of this document.

The Button Input Client (Button Transmitter) shall increment the setRotarySteps signal by 1 for every detected rotary knob detent in the clockwise direction.

- For example if the Button Transmitter detects 3 detents in the clockwise direction could send +3 (fast turn) or three +1 button press messages (slower turn) as long as no information is lost. Note for LIN do not need to see the Not_Press to act on one of the volume steps.

The Button Input Client (Button Transmitter) shall decrement the setRotarySteps signal by 1 for every detected rotary knob detent in the counter-clockwise direction.

- For example if the Button Transmitter detects 3 detents in the counter clockwise direction could send -3 (fast turn) or three -1 button press messages (slower turn) as long as no information is lost. Note for LIN do not need to see the Not_Press to act on one of the volume steps.

The Button Input Client (Button Transmitter) shall send out the delta counts accumulated since the last setRotarySteps signal sent out on the bus.

If delta counts are 0 then "Not Pressed / Inactive" shall be sent out.

1.4.4 setVolume signal (LIN)

1.4.4.1 BUTTON-SR-REQ-096743/B-Tx SetVolume (LIN)

The setVolume signal can be used to increment / decrement the Volume for a volume rotary knob for example. Reference the applicable SPSS section for details. Each setVolume step encoding shall be treated as a press event in the remainder of this document.

The Button Input Client (Button Transmitter) shall increment the setVolume signal by 1 for every detected rotary knob detent in the clockwise direction.

- For example if the Button Transmitter detects 3 detents in the clockwise direction could send +3 (fast turn) or three +1 button press messages (slower turn) as long as no information is lost. Note for LIN do not need to see the Not_Press to act on one of the volume steps.

The Button Input Client (Button Transmitter) shall decrement the setVolume signal by 1 for every detected rotary knob detent in the counter-clockwise direction.

- For example if the Button Transmitter detects 3 detents in the counter clockwise direction could send -3 (fast turn) or three -1 button press messages (slower turn) as long as no information is lost. Note for LIN do not need to see the Not_Press to act on one of the volume steps.

The Button Input Client (Button Transmitter) shall send out the delta counts accumulated since the last setVolume signal sent out on the bus.

If delta counts are 0 then "Not Pressed / Inactive" shall be sent out.



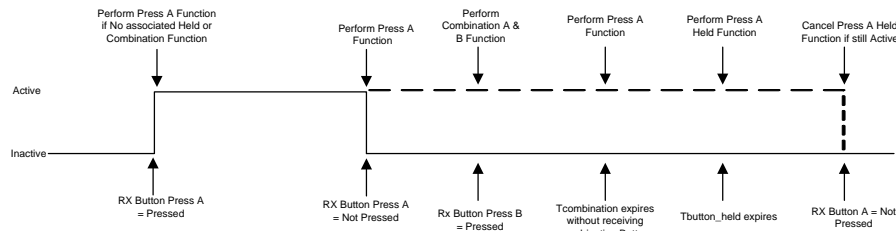
1.5 BUTTONv2-CLD-REQ-095294/A-Button Input Server (Button Receiver) - LIN

The following sections define the Button Activation Strategy from the Receivers perspective.

1.5.1 Button Activation - Receiver Timing Requirements (LIN)

1.5.1.1 BUTTONv2-SR-REQ-096745/B-Receiver Button Activation Process (RBAP) Timing (LIN)

All component receivers of button press information shall implement the Receiver Button Activation Process (RBAP) timing defined in the figure below.



Receiver Button Activation Process (RBAP) Timing

Exception: If a particular button supports press and hold function there may be times where a functional requirement may require the function to first be performed on the press (not wait for not pressed) and then take additional action when Tbutton_held expires and the buttons are determined to be held. This should only be performed if explicitly called out in a functional requirement otherwise the function shall be performed only on a Not Press or when Tbutton_held expires as shown in the RBAP above.

Exception 2: Rotary Knobs signals do not need to wait for the not pressed (ex setVolume, setRotarySteps).

1.5.2 Button Activation - Receiver Functional Requirements (LIN)

1.5.2.1 BUTTONv2-SR-REQ-096746/A-Receivers of Button Presses follow RBAP (LIN)

All receivers of button press information shall activate the RBAP upon receipt of the button press message.

1.5.2.2 BUTTONv2-SR-REQ-096747/B-Button Receiver Sampling Rate (LIN)

The sampling rate used by the receiver to read incoming button information messages shall be fast enough to read the multiple incoming messages.

1.5.2.3 BUTTON-SR-REQ-014699/A-Receivers of Held Button Presses (TcSE ROIN-39784-1)

The receiver shall determine whether the specific button has an associated held function.

1.5.2.4 BUTTON-SR-REQ-014700/B-Button Presses with no Held function or Combination with another button press (TcSE ROIN-39785-1)

If the button does not have an associated held or combination type function, then the receiver shall perform the function associated with a button press immediately upon receipt of the button press message and the RBAP process shall be exited.

1.5.2.5 BUTTON-SR-REQ-014701/B-Buttons with Held Function (TcSE ROIN-39786-1)

If the button does have an associated held function, the receiver shall start a hold timer (Tbutton_held) and wait for button 'not pressed' information.



-- If a button 'not pressed' message is not received prior to the expiration of Tbutton_held, the receiver shall perform the associated held function for that button press.

-- If a button 'not pressed' message is received prior to the expiration of Tbutton_held, then the receiver shall perform the associated press function.

1.5.2.6 BUTTON-SR-REQ-014702/A-Receivers of combination type button presses (TcSE ROIN-39787-1)

The receiver shall determine whether the button press has an associated combination type button press.

1.5.2.7 BUTTON-SR-REQ-014703/B-Combination Type button press operation (TcSE ROIN-39788-1)

When the receiver detects a button press (A) that has an associated combination the receiver must wait Tcombination before executing the button press function associated with the single button press (A).

- If a second button (B) is received prior to Tcombination expiring, the receiving module must now determine if this (B) is part of a valid combination with (A).
 - If the combination is valid, then the resulting combination (A + B) can be performed.
 - If the combination is invalid, then no combination function is performed and the button presses (A) & (B) can be processed independently if allowed by the receiving module.

-- If Tcombination expires, then button press (A) is now valid and the (A) function can be performed.

1.5.2.8 BUTTON-SR-REQ-014704/B-Cancelling RBAP (TcSE ROIN-39789-1)

The receiver shall cancel the RBAP upon:

- Receipt of the button 'Not Pressed' message.
- If the receiver does not receive a button 'Not Pressed' message within T_RBAP_Timeout (T_RBAP_Timeout as defined in applicable feature specs).

1.5.2.9 BUTTON-SR-REQ-014705/A-Cancelling RBAP when change to Standby (TcSE ROIN-39790-2)

For Infotainment receivers any event that shall cause a transition from Functional mode to Standby mode shall cancel a RBAP unless noted otherwise. The receiver shall cancel the operation, if active, and perform the required actions to enter Standby State.

1.5.2.10 BUTTONv2-SR-REQ-096749/A-Receivers of SetVolume Button presses (LIN)

The Button Input Server (Volume Setting Server) shall increment/decrement the volume based on the delta count of the volume steps received since the last SetVolume signal update.

1.5.2.11 BUTTON-SR-REQ-107307/A-Receivers of SetRotarySteps Button Presses (LIN)

The Button Input Server (Rotary Setting Server) shall increment/decrement the steps based on the delta count of the steps received since the last SetRotarySteps signal update.



2 Appendix: Reference Documents

Reference #	Document Title
1	Input Translation Matrix
2	LIN DVM specifications
3	LIN Data LINK and Physical Layer specifications
4	LIN Physical Layer Approved Components
5	SAE J2602-3
6	SAE J2602-2
7	SAE J2602-1
8	LIN Database File (LDF)
9	LIN Illumination Specification
10	LIN Netcom SOW
11	
12	
13	