

/IPB/20\_Software/SCDD Library/LVDC DSP\_C

SCDD\_ComServ

Software Component Detailed Design

Version: 0.1 (C0\_RfR)

Printed by: I-Ritesh.K

Printed on: Tuesday, July 23, 2024

<b>1</b>	<b>Software Component Design Description</b>	<b>1</b>
1.1	Introduction	1
<b>2</b>	<b>Attributes</b>	<b>2</b>
<b>3</b>	<b>Views</b>	<b>3</b>
<b>4</b>	<b>ComServ</b>	<b>4</b>
4.1	External Interfaces	4
4.2	Internal design	4
4.3	Requirements	5
4.3.1	ASIL	5
4.3.2	Data Define	5
4.3.3	ComServ Function	5
4.3.3.1	ASIL Function	5
4.3.3.1.1	void HsfbApp_GetRx10ms(HSFBAPP_S_RX *y)	5
4.3.3.1.2	void HsfbApp_SetTx10ms(const HSFBAPP_S_TX *u)	6
4.3.3.1.3	void ComServ_EmergencyMode(void)	6
4.3.3.1.4	void ComServ_AutoDisable(HSFBAPP_S_RX* y)	6
4.3.3.1.5	void ComServ_1msTask(void)	7
4.3.3.1.6	void ComServ_10msTask(void)	7
4.3.3.1.7	void CANDriver_Diag(uint16* const pu16ErrCode)	7
4.3.3.1.8	void CANDriver_HwCfg(void)	7
4.3.3.1.9	static void ComServ_ReadComData(void)	7
4.3.3.1.10	static inline u16 ConvertSM(bit SM)	7
4.3.3.1.11	static inline u16 ConvertAutoDiag(bit SM,bit Fin)	8
4.3.3.1.12	static u16 ConvertState(HSFBSTATE_E_STATE State)	8
4.3.3.1.13	static u16 ConvertTempToCan(s16 Temp)	8
4.3.3.1.14	static void UdsData_Param(u16 ReadWrite, u8* Buffer, u16* DataPtr, u16 DataSize)	8
4.3.3.1.15	uint8 UdsData_Parameter(uint8 ReadWrite, uint8 *RxCtxBuf, uint16 LenOfMsg)	8
4.3.3.1.16	uint8 UdsData_Calibration(uint8 ReadWrite, uint8 *RxCtxBuf, uint16 LenOfMsg)	9
4.3.3.1.17	uint32 Crc_CalculateCRC32(uint8 *Crc_DataPtr, uint16 Crc_Length)	9
4.3.3.1.18	uint8 Crc_CalculateCRC8(const uint16_t* DataPtr, uint16 Length, uint16 CrcByteIdx)	9

4.3.3.1.19	static void CANDriver_TxDataE2E(uint16 u16Index)	9
4.3.3.1.20	static void CANDriver_RxDataE2E(uint16 u16Index)	9
4.3.3.1.21	void HsfbMeas_GetCalibData(HSFBMEAS_S_CALIB *y)	10
4.3.3.1.22	void HsfbCtrl_GetParamData(HSFBCTRL_S_PARAM *y)	10
4.3.3.2	QM Function	10
4.3.3.2.1	void ComServ_Init(void)	10
4.3.3.2.2	static void ComServ_InitStatus(void)	10

ID	Software Component Detailed Design
SCDD_C omServ _1	<sup>1</sup> <b>Software Component Design Description</b>
SCDD_C omServ _2	<sup>1.1</sup> <b>Introduction</b>
SCDD_C omServ _3	This document describes the needed requirements for a SWC or BSWM.
SCDD_C omServ _4	<p>This is module is the Software Component Detail Description.  It contains each SW component of each SW architecture.  It is always structured in:</p> <ul style="list-style-type: none"> <li>External Interface</li> <li>Internal Design</li> <li>Requirements</li> </ul>

ID	Software Component Detailed Design
SCDD_C omServ _5	<sup>2</sup> <b>Attributes</b>
SCDD_C omServ _6	Agreed attributes for SWE.3 (ENG.6)

ID	Software Component Detailed Design
SCDD_C omServ _10	<sup>3</sup> <b>Views</b>
SCDD_C omServ _11	<b>SwConstructionView:</b> This view is used for the sw construction process.
SCDD_C omServ _12	<b>SCDD_EditView:</b> This view is used for creating the content of SCDD

ID	Software Component Detailed Design																																																																																														
SCDD_C omServ _13	4 <b>ComServ</b>																																																																																														
SCDD_C omServ _14	4.1 <b>External Interfaces</b>																																																																																														
SCDD_C omServ _15	<p>The function interface of this component are as following:</p> <table> <tr> <th>Function</th><th>Signal Name</th><th>Data Type</th><th>Direction</th></tr> <tr> <td>HsfbApp_GetRx10ms</td><td>y</td><td>HSFBAPP_S_RX*</td><td>Output</td></tr> <tr> <td>HsfbApp_SetTx10ms</td><td>u</td><td>HSFBAPP_S_TX*</td><td>Input</td></tr> <tr> <td>ComServ_EmergencyMode</td><td>N/A</td><td>N/A</td><td>N/A</td></tr> <tr> <td>ComServ_AutoDisable</td><td>y</td><td>HSFBAPP_S_RX*</td><td>Output</td></tr> <tr> <td>ComServ_1msTask</td><td>N/A</td><td>N/A</td><td>N/A</td></tr> <tr> <td>ComServ_10msTask</td><td>N/A</td><td>N/A</td><td>N/A</td></tr> <tr> <td>CANDriver_Diag</td><td>pu16ErrCode</td><td>uint16*</td><td>Input</td></tr> <tr> <td>CANDriver_HwCfg</td><td>N/A</td><td>N/A</td><td>N/A</td></tr> <tr> <td>ComServ_ReadComData</td><td>N/A</td><td>N/A</td><td>N/A</td></tr> <tr> <td>ConvertSM</td><td>SM (return value)</td><td>boolean uint16</td><td>Input Output</td></tr> <tr> <td>ConvertAutoDiag</td><td>SM Fin (return value)</td><td>boolean boolean uint16</td><td>Input Input Output</td></tr> <tr> <td>ConvertState</td><td>State (return value)</td><td>HSFBSTATE_E_STATE uint16</td><td>Input Output</td></tr> <tr> <td>ConvertTempToCan</td><td>Temp (return value)</td><td>sint16 uint16</td><td>Input Output</td></tr> <tr> <td>UdsData_Param</td><td>ReadWrite Buffer DataPtr DataSize</td><td>uint16 uint8* uint16* uint16</td><td>Input Input Output Input</td></tr> <tr> <td>UdsData_Parameter</td><td>ReadWrite RxTxBuf LenOfMsg</td><td>uint8 uint8* uint16</td><td>Input Output Input</td></tr> <tr> <td>UdsData_Calibration</td><td>ReadWrite RxTxBuf LenOfMsg</td><td>uint8 uint8* uint16</td><td>Input Output Input</td></tr> <tr> <td>Crc_CalculateCRC32</td><td>Crc_DataPtr Crc_Length</td><td>uint8* uint16</td><td>Input Input</td></tr> <tr> <td>Crc_CalculateCRC8</td><td>DataPtr Length CrcByteIdx</td><td>uint16* uint16 uint16</td><td>Input Input Input</td></tr> <tr> <td>CANDriver_TxDataE2E</td><td>u16Index</td><td>uint16</td><td>Input</td></tr> <tr> <td>CANDriver_RxDataE2E</td><td>u16Index</td><td>uint16</td><td>Input</td></tr> <tr> <td>HsfbMeas_GetCalibData</td><td>y</td><td>HSFBMEAS_S_CALIB*</td><td>Output</td></tr> <tr> <td>HsfbCtrl_GetParamData</td><td>y</td><td>HSFBCTRL_S_PARAM*</td><td>Output</td></tr> </table>			Function	Signal Name	Data Type	Direction	HsfbApp_GetRx10ms	y	HSFBAPP_S_RX*	Output	HsfbApp_SetTx10ms	u	HSFBAPP_S_TX*	Input	ComServ_EmergencyMode	N/A	N/A	N/A	ComServ_AutoDisable	y	HSFBAPP_S_RX*	Output	ComServ_1msTask	N/A	N/A	N/A	ComServ_10msTask	N/A	N/A	N/A	CANDriver_Diag	pu16ErrCode	uint16*	Input	CANDriver_HwCfg	N/A	N/A	N/A	ComServ_ReadComData	N/A	N/A	N/A	ConvertSM	SM (return value)	boolean uint16	Input Output	ConvertAutoDiag	SM Fin (return value)	boolean boolean uint16	Input Input Output	ConvertState	State (return value)	HSFBSTATE_E_STATE uint16	Input Output	ConvertTempToCan	Temp (return value)	sint16 uint16	Input Output	UdsData_Param	ReadWrite Buffer DataPtr DataSize	uint16 uint8* uint16* uint16	Input Input Output Input	UdsData_Parameter	ReadWrite RxTxBuf LenOfMsg	uint8 uint8* uint16	Input Output Input	UdsData_Calibration	ReadWrite RxTxBuf LenOfMsg	uint8 uint8* uint16	Input Output Input	Crc_CalculateCRC32	Crc_DataPtr Crc_Length	uint8* uint16	Input Input	Crc_CalculateCRC8	DataPtr Length CrcByteIdx	uint16* uint16 uint16	Input Input Input	CANDriver_TxDataE2E	u16Index	uint16	Input	CANDriver_RxDataE2E	u16Index	uint16	Input	HsfbMeas_GetCalibData	y	HSFBMEAS_S_CALIB*	Output	HsfbCtrl_GetParamData	y	HSFBCTRL_S_PARAM*	Output
Function	Signal Name	Data Type	Direction																																																																																												
HsfbApp_GetRx10ms	y	HSFBAPP_S_RX*	Output																																																																																												
HsfbApp_SetTx10ms	u	HSFBAPP_S_TX*	Input																																																																																												
ComServ_EmergencyMode	N/A	N/A	N/A																																																																																												
ComServ_AutoDisable	y	HSFBAPP_S_RX*	Output																																																																																												
ComServ_1msTask	N/A	N/A	N/A																																																																																												
ComServ_10msTask	N/A	N/A	N/A																																																																																												
CANDriver_Diag	pu16ErrCode	uint16*	Input																																																																																												
CANDriver_HwCfg	N/A	N/A	N/A																																																																																												
ComServ_ReadComData	N/A	N/A	N/A																																																																																												
ConvertSM	SM (return value)	boolean uint16	Input Output																																																																																												
ConvertAutoDiag	SM Fin (return value)	boolean boolean uint16	Input Input Output																																																																																												
ConvertState	State (return value)	HSFBSTATE_E_STATE uint16	Input Output																																																																																												
ConvertTempToCan	Temp (return value)	sint16 uint16	Input Output																																																																																												
UdsData_Param	ReadWrite Buffer DataPtr DataSize	uint16 uint8* uint16* uint16	Input Input Output Input																																																																																												
UdsData_Parameter	ReadWrite RxTxBuf LenOfMsg	uint8 uint8* uint16	Input Output Input																																																																																												
UdsData_Calibration	ReadWrite RxTxBuf LenOfMsg	uint8 uint8* uint16	Input Output Input																																																																																												
Crc_CalculateCRC32	Crc_DataPtr Crc_Length	uint8* uint16	Input Input																																																																																												
Crc_CalculateCRC8	DataPtr Length CrcByteIdx	uint16* uint16 uint16	Input Input Input																																																																																												
CANDriver_TxDataE2E	u16Index	uint16	Input																																																																																												
CANDriver_RxDataE2E	u16Index	uint16	Input																																																																																												
HsfbMeas_GetCalibData	y	HSFBMEAS_S_CALIB*	Output																																																																																												
HsfbCtrl_GetParamData	y	HSFBCTRL_S_PARAM*	Output																																																																																												
SCDD_C omServ _16	4.2 <b>Internal design</b>																																																																																														
SCDD_C omServ _17																																																																																															





ID	Software Component Detailed Design
SCDD_C omServ _95	This function will call following function: ComServ_AutoDisable()
SCDD_C omServ _27	4.3.3.1.2 <b>void HsfbApp_SetTx10ms(const HSFAPP_S_TX *u)</b>
SCDD_C omServ _29	This function is an external function called by HsfbApp with a cycle time of 10ms. In this function, the data in the variable corresponding to the pointer will be read and written to the array of DCDC_Measure, DCDC_Measure2, DCDC_Measure3, DCDC_Measure4, DCDC_Status, DCDC_Status_ErrCode and LVDC_SafetyMeasure. The input of this function is the pointer with data type HSFAPP_S_TX. This function has no output.
SCDD_C omServ _96	This function will call following function: ConvertSM() ConvertAutoDiag() ConvertState() ConvertTempToCan()
SCDD_C omServ _30	4.3.3.1.3 <b>void ComServ_EmergencyMode(void)</b>
SCDD_C omServ _34	This function is used for checking the CAN error flag and controlling the Enable, PeakCurrEn, default IoutRef and VoutRef variable in ComServ_Status when emergency mode is needed. This function has no input and no output.
SCDD_C omServ _36	If CanError is 0, the PeakCurrentEn and CanErrorCnt will be set to 0, as there is no CAN error happen.
SCDD_C omServ _38	If CanError is 1 and Operation is 0 (Buck mode), the CanErrorCnt will add 1 every 10ms. If CanErrorCnt < 15000 (150s), the PeakCurrentEn will be set to 1, otherwise the Enable, PeakCurrentEn will be set to 0.
SCDD_C omServ _31	4.3.3.1.4 <b>void ComServ_AutoDisable(HSFAPP_S_RX* y)</b>
SCDD_C omServ _35	This function one part of HsfbApp_GetRx10ms, which is used for make sure the LVDC will not change direction or topology by COM during normal operation. Before the Enable, Operation, Topology is written from ComServ_Status to the variable corresponding to the pointer, it will check if the command is changing or not. This function has no input. The output of this function is the pointer of HsfbApp_Rx10ms.
SCDD_C omServ _39	If the Enable is changing from 1 to 0, then the Enable will be written directly to the destination. Otherwise, if Operation or Topology change is detected, it will set Enable to 0 first, and write the value to the destination in the next cycle.
SCDD_C omServ _40	If no command is changing, the EnableCnt will add 1 every 10ms. If the Enable is changing from 0 to 1, only when EnableCnt is already reach the maximum value (10 for 100ms) can the Enable be written to the destination.

ID	Software Component Detailed Design
SCDD_C omServ _49	4.3.3.1.5 <b>void ComServ_1msTask(void)</b>
SCDD_C omServ _93	This function is the 1ms task of ComServ, mainly used for reading data from Rx buffer and writing data to Tx buffer. This function has no input and no output.
SCDD_C omServ _94	This function will call following function: CANDriver_TxDataE2E() CANDriver_RxDataE2E()
SCDD_C omServ _50	4.3.3.1.6 <b>void ComServ_10msTask(void)</b>
SCDD_C omServ _91	This function is the 10ms task of ComServ, mainly used for writing data from Rx buffer to local variable, and perform emergency mode. This function has no input and no output.
SCDD_C omServ _92	This function will call following function: ComServ_ReadComData() ComServ_EmergencyMode()
SCDD_C omServ _51	4.3.3.1.7 <b>void CANDriver_Diag(uint16* const pu16ErrCode)</b>
SCDD_C omServ _89	This function is used for checking if CAN bus has any error (e.g. bus off) from CAN register. If any CAN bus error happen, it will reset CAN configuration. This function has no input. The output of this function is the pointer of last error code from CAN register.
SCDD_C omServ _90	This function will call following function: CANDriver_HwCfg()
SCDD_C omServ _52	4.3.3.1.8 <b>void CANDriver_HwCfg(void)</b>
SCDD_C omServ _88	This function is used for configuration of CAN peripheral. This function has no input and no output.
SCDD_C omServ _53	4.3.3.1.9 <b>static void ComServ_ReadComData(void)</b>
SCDD_C omServ _79	This function will read data from buffer of DCDC_CMD and DCDC_DEV_CMD, and write data to ComServ_Status variable. This function has no input and no output.
SCDD_C omServ _54	4.3.3.1.10 <b>static inline u16 ConvertSM(bit SM)</b>

ID	Software Component Detailed Design
SCDD_C omServ _66	This function will convert safety status from HsfbSafety to safety status signal defined in DBC file. The input of this function is the value of safety status. The output of this function is the value of safety status signal.
SCDD_C omServ _55	4.3.3.1.11 <b>static inline u16 ConvertAutoDiag(bit SM,bit Fin)</b>
SCDD_C omServ _67	This function will convert safety status from HsfbSafety to safety status signal defined in DBC file. When AutoDiag is not finish, the signal will be set to 3, otherwise will be set to 1 or 2. The input of this function is the value of safety status and AutoDiag finish flag. The output of this function is the value of safety status signal.
SCDD_C omServ _56	4.3.3.1.12 <b>static u16 ConvertState(HSFBSTATE_E_STATE State)</b>
SCDD_C omServ _68	This function will convert state from HsfbState to state signal defined in DBC file. The input of this function is the enumerate value of state. The output of this function is the value of state signal.
SCDD_C omServ _57	4.3.3.1.13 <b>static u16 ConvertTempToCan(s16 Temp)</b>
SCDD_C omServ _69	This function will convert temperature from HsfbNtc to temperature signal defined in DBC file. The input of this function is the enumerate value of state. The output of this function is the value of temperature signal.
SCDD_C omServ _58	4.3.3.1.14 <b>static void UdsData_Param(u16 ReadWrite, u8* Buffer, u16 * DataPtr, u16 DataSize)</b>
SCDD_C omServ _72	This function will write data from Rx buffer to ParameterData or CalibrationData. The input of this function is ReadWrite command, pointer of buffer and DataSize of output data. The output of this function is the pointer of output data.
SCDD_C omServ _59	4.3.3.1.15 <b>uint8 UdsData_Parameter(uint8 ReadWrite, uint8 *RxTxBuf, uint16 LenOfMsg)</b>
SCDD_C omServ _73	This function is used for check data in Rx buffer, write it to ParameterData, and give response. In the function, the ATE CRC and COM CRC will be calculated. Only when both CRC is same with the data in Rx buffer, can data be written to ParameterData. The input of this function is ReadWrite command, pointer of buffer and length of message. This function has no output.
SCDD_C omServ _77	This function will call following function: Crc_CalculateCRC32() UdsData_Param()

ID	Software Component Detailed Design
SCDD_C omServ _60	4.3.3.1.16 <b>uint8 UdsData_Calibration(uint8 ReadWrite, uint8 *RxTxBuf, uint16 LenOfMsg)</b>
SCDD_C omServ _74	This function is used for check data in Rx buffer, write it to CalibrationData, and give response. In the function, the ATE CRC and COM CRC will be calculated. Only when both CRC is same with the data in Rx buffer, and IsCalibrated flag is ture, can data be written to CalibrationData. The input of this function is ReadWrite command, pointer of buffer and length of message. This function has no output.
SCDD_C omServ _78	This function will call following function: Crc_CalculateCRC32() UdsData_Param()
SCDD_C omServ _61	4.3.3.1.17 <b>uint32 Crc_CalculateCRC32(uint8 *Crc_DataPtr, uint16 Crc_Length)</b>
SCDD_C omServ _75	This function is used for calculating the CRC32 of data. The input of this function is the pointer of data and length of data. The output of this function is the CRC32 result.
SCDD_C omServ _62	4.3.3.1.18 <b>uint8 Crc_CalculateCRC8(const uint16_t* DataPtr, uint16 Length, uint16 CrcByteIdx)</b>
SCDD_C omServ _76	This function is used for calculating the CRC8 of data. The input of this function is the pointer of data, length of data and CRC8 byte index. The output of this function is the CRC8 result.
SCDD_C omServ _63	4.3.3.1.19 <b>static void CANDriver_TxDataE2E(uint16 u16Index)</b>
SCDD_C omServ _80	This function is used for calculating BZ and CRC data and writing to E2E protected Tx message. The input of this function is the index of CAN Tx message. This function has no output.
SCDD_C omServ _83	This function will call following function: Crc_CalculateCRC8()
SCDD_C omServ _82	The BZ will add 1 for every message cycle time, and will be written to the 8~11bit of message. The CRC will be calculated for every message cycle time, with the data start from bit 8 to end of message, and will be written to the 0~7bit of message.
SCDD_C omServ _81	The E2E protected Tx message include: DCDC_MEASURE DCDC_MEASURE2 DCDC_MEASURE4
SCDD_C omServ _64	4.3.3.1.20 <b>static void CANDriver_RxDataE2E(uint16 u16Index)</b>

ID	Software Component Detailed Design
SCDD_C omServ _84	This function is used for checking BZ and CRC data from E2E protected Rx message. The E2E check result will be stored in CanE2E variable. The input of this function is the index of CAN Rx message. This function has no output.
SCDD_C omServ _85	This function will call following function: Crc_CalculateCRC8()
SCDD_C omServ _86	The BZ data will be read from 8~11bit of message, and if it's different with local BZ data, the error counter will add 1. The local BZ data will be overwritten by BZ data from message, and add 1 every time before E2E check. The CRC data will be read from 0~7bit of message, and if it's different with local CRC calculation result, the error counter will add 1. If any error counter of BZ or CRC is not 0, the E2E error counter will add 1. If the value of E2E error counter reach 10, the E2E error flag will be set to 1.
SCDD_C omServ _87	The E2E protected Rx message include: DCDC_CMD
SCDD_C omServ _65	4.3.3.1.21 <b>void HsfbMeas_GetCalibData(HSFBMEAS_S_CALIB *y)</b>
SCDD_C omServ _70	This function is used for writing calibration data from CalibrationData in ComServ to HsfbMeas_Calib in HsfbMeas. This function has no input. The output of this function is the pointer of HsfbMeas_Calib.
SCDD_C omServ _32	4.3.3.1.22 <b>void HsfbCtrl_GetParamData(HSFBCTRL_S_PARAM *y)</b>
SCDD_C omServ _71	This function is used for writing parameter data from ParameterData in ComServ to HsfbCtrl_Param in HsfbCtrl. This function has no input. The output of this function is the pointer of HsfbCtrl_Param.
SCDD_C omServ _43	4.3.3.2 <b>QM Function</b>
SCDD_C omServ _44	4.3.3.2.1 <b>void ComServ_Init(void)</b>
SCDD_C omServ _45	This function is used for Initialization of ComServ component. This function has no input and no output.
SCDD_C omServ _46	This function will call following function: ComServ_InitStatus()
SCDD_C omServ _47	4.3.3.2.2 <b>static void ComServ_InitStatus(void)</b>

ID	Software Component Detailed Design
SCDD_C omServ _48	<p>This function will write initial value of AppDateTime, AppVersion and CfgVersion in ComServ_Status variable.</p> <p>This function has no input and no output.</p>