

SRS_P1_ECU

Contents

- 1 Scope 1
- 2 Internal_Block_Diagram = its_IBD() 2
- 3 Funct_Req_dscr = P1_ECU:ECU 2
 - 3.1 Black_Box_Interfaces = its_BB_IFs() 2
 - 3.1.1 Input_Signal1 = switch_signal:AN_IN 2
 - 3.1.2 Output_Message1 = switch_msg:CAN_OUT 3
 - 3.1.3 Output_Frame1 = DCU_status:CAN_OUT 3
 - 3.2 Event = its_Events() 3
 - 3.2.1 Event1 = Switch_Event 3
 - 3.2.2 SF = Measure_Switch_Status_SF 3
 - 3.2.3 SF = Evaluate_Signals_SF 6
 - 3.3 Supplement_Functions = P1_SUP 9
 - 3.3.1 Mode_Handling = ECU_Modes 9
 - 3.3.2 UC_Features = xx_UC_Features 13
 - 3.3.3 Operating_System = AUTOSAR 14
 - 3.3.4 (1..*)Ext_Power_Supply = XX:Ext_POWER_SUPPLY 15
 - 3.3.5 (1..*)Int_Power_Supply 16
 - 3.3.6 (0..*)BUS_Handling 16
 - 3.3.7 (0..*)DIAG_Handling 16
 - 3.3.8 (0..*)NVM_Handler 16
- 4 Nonfunctional Requirements 16
 - 4.1 Quality_Req 16
 - 4.2 Reusability_Req 16
 - 4.3 Criticalty 16
- 5 Abbreviations = Used_Abbreviations 16

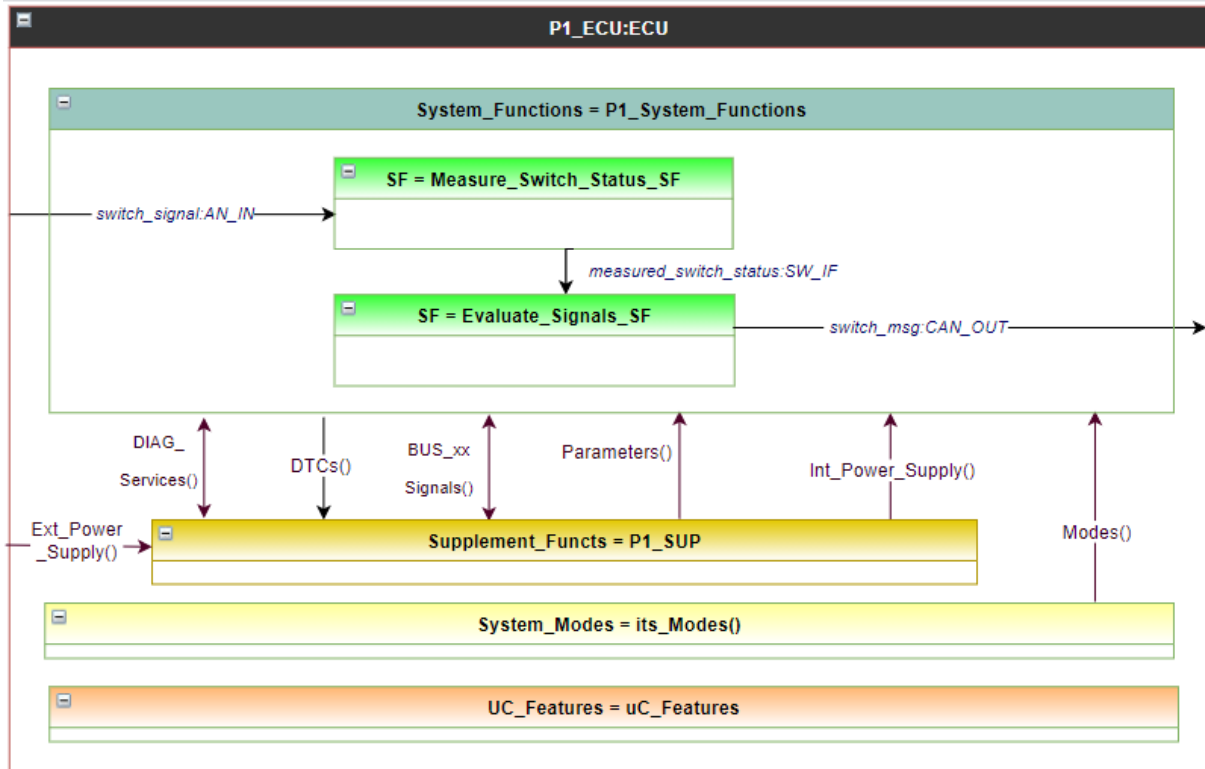
1 Scope

This document is describing the P1_ECU as White-Box.

SRS_P1_ECU

2 Internal_Block_Diagram = its_IBD()

(Note: This diagram is automatically generated out of information provided from the project specific RML classes)



3 Funct_Req_dscr = P1_ECU:ECU

3.1 Black_Box_Interfaces = its_BB_IFs()

(Note: All following IF requirements tables are automatically generated out of the imported IF Signals)

3.1.1 Input_Signal1 = switch_signal:AN_IN

SRDE-524 -

Name	Type_and_direction	Sender	Receiver	Intended usage	Update_rate	States	Connector_pin	Event
switch_signal	AN_IN	Analog_Switch	Measure_Switch_Status	The switch status shall be provided on the CAN bus	contin-uous	SWITCH_OPEN = 100 Ohm <= R < 200 Ohm SWITCH_CLOSED = 200 Ohm <= R < 300 Ohm SWITCH_SC_GND = R < 100 Ohm SWITCH_OP_CIRC = "R >= 300 Ohm	X1.3	Switch_Event

SRS_P1_ECU

3.1.2 Output_Message1 = switch_msg:CAN_OUT

SRDE-661 -

Name	Type_and_direction	Sender	Receiver	Intended usage	Update_rate	States	Frame	Event
switch_msg	CAN_OUT	Evaluate_Signals	BCM_ECU	Shall be used to trigger the LED	200 ms	SWITCH_OPEN = Bit 0 is set SWITCH_CLOSED = Bit 1 is set SWITCH_ERROR = Bit 2 is set	DCU_Status	no

3.1.3 Output_Frame1 = DCU_status:CAN_OUT

SRDE-728 -

Name	Type_and_direction	Sender	Receiver	Update_rate	Frame_sending_criteria	Frame_repetition_criteria	Message
DCU_status	CAN_OUT	Evaluate_Signals	BCM_ECU	200 ms	Key_Pos1-3	no repetition	byte 0 -> switch_msg

3.2 Event = its_Events()

(Note: All open Requirement Engineer Tasks are marked in red)

3.2.1 Event1 = Switch_Event

SRDE-677 - Ev_qualific_crit = tbd

SRDE-678 - Event_actions = tbd

SRDE-679 - Ev_reoccur_handle = tbd

SRDE-680 - Ev_reaction_time = tbd

SRDE-681 - (0..1)Ev_traceability_action = tbd

3.2.2 SF = Measure_Switch_Status_SF

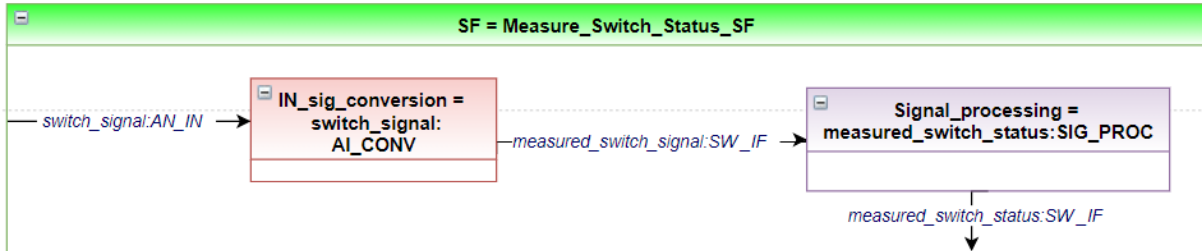
3.2.2.1 Scope

Description of switch measurement.

SRS_P1_ECU

3.2.2.2 Internal_Block_Diagram = its_IBD()

(Note: This diagram is automatically generated out of information provided from the project specific RML classes)



3.2.2.3 Interface_req = SF_its_IF_REQ()

(Note: All following IF requirements tables are automatically generated out of the imported IF Signals)

3.2.2.3.1 Input_Signal1 = switch_signal:AN_IN

SRDE-665 -

Name	Type_and_direction	Sender	Receiver	Intended usage	Update_rate	States	Connector_pin	Event
switch_signal	AN_IN	Analog_Switch	Measure_Switch_Status	If the switch is closed, the LED shall be on	continuous	SWITCH_OPEN = 100 Ohm <= R < 200 Ohm SWITCH_CLOSED = 200 Ohm <= R < 300 Ohm SWITCH_SC_GND = R < 100 Ohm SWITCH_OP_CIRC = "R >= 300 Ohm	X1.3	Switch_Event

3.2.2.3.2 Output_Signal1 = measured_switch_status:SW_IF

SRDE-667 -

Name	Type_and_direction	Sender	Receiver	Intended usage	Update_rate	(0..*) xx_STATE	(0..*) xx_VALUE	Default_Value
measured_switch_status	SW_IF	Measure_Switch_Status	Evaluate_Signals	Input for switch_msg proc.	tbd	tbd	tbd	tbd

3.2.2.3.3 SF_Parameter = its_parameter()

SRDE-675 - (0..1) xx_CONFIG_PARAMETER = tbd

SRDE-676 - (0..*) xx_SF_PARAMETER = tbd

SRS_P1_ECU

3.2.2.3.4 SF_Diag_Services = its_Diag_services();

SRDE-683 - (0..*) xx_Diag_Service_Routines = tbd

3.2.2.4 Execution_rate = its_Autosar_rate()

SRDE-702 - Execution_rate = tbd

3.2.2.5 IN_sig_conversion= switch_signal:AI_CONV

IN_sig_conversion=switch_signal:AI_CONV ->> SRS_P1_HSI

(Note: This function is described in a separate specification)

3.2.2.6 Signal_processing = measured_switch_status:SIGNAL_PROC

3.2.2.6.1 Relevant_Modes= its_mode_dependency()

(Note: All following IF requirements tables are automatically generated out of the defined Modes)

SRDE-688 -

ECU_Modes			Mode Dependency
Life_Cycle_Phases	System_Modes	SW_Modes	
[(1..*)ECU_production_mode, (1..*)System_assembly_modes, (1..*)System_transport_modes, (1..*)System_transport_modes]	[(1..*)Imported_System_Modes = tbd]	[Start_up_mode, (1..*)Operation_mode, (0..*)Energy_safe_mode, (0..*)Fail_safe_mode, (0..*)Diagnostic_mode, Power_down_mode]	
tbd	tbd	tbd	SF ENABLED
tbd	tbd	tbd	SF DISABLED

3.2.2.6.2 Processing_Goals= its_processing_goals()

(Note: Here comes an automatic created list of all processing goals that are connected to this SF and shall be intentionally

defined in the following logic_table)

SRDE-689 - Processing_Goal1 = If Bat_voltage < 9V, then the previous signal value shall be used

3.2.2.6.3 Logic_Table= its_logic_table()

(Note: the headings of the table is automatically created due to the

SRS_P1_ECU

- Signal_Processing its connected IO and error signals,
- Potential configuration parameters
- Modes in which the SF is enabled
- Potential usage as state machine

The table contents must be finalized manually.)

SRDE-691 - measured_switch_status_its_Logic_table

(0..1) CONFIG_ PARAM [enable, disable]	LIFE_CYCLE _PHASE [tbd]	SYSTEM_MODES [tbd]	SW_ MODES [tbd]	measured_ switch_signal [tbd]	Result: measured_ switch_status [tbd]	(0..*) Follow up State	(0..*) Condition for transition
tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd

3.2.2.6.4 Functional_Requirements = switch_status_its_text_conversion

(Note: Those textual requirements are automatically created out of the logic table)

SRDE-701 - tbd

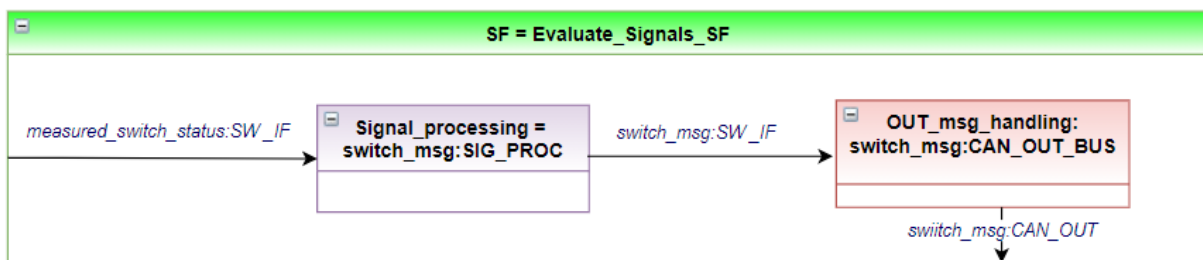
3.2.3 SF = Evaluate_Signals_SF

3.2.3.1 Scope

Description of switch_evaluation

3.2.3.2 Internal_Block_Diagram = its_IBD()

(Note: This diagram is automatically generated out of information provided from the project specific RML classes)



3.2.3.3 Interface_req = SF_its_IF_REQ()

(Note: All following IF requirements tables are automatically generated out of the imported IF Signals)

SRS_P1_ECU

3.2.3.3.1 Input_Signal1 = measured_switch_status:SW_IF

SRDE-711 -

Name	Type_and_direction	Sender	Receiver	Intended usage	Update_rate	(0..*) xx_STATE	(0..*) xx_VALUE	Default_Value
measured_switch_status	SW_IF	Measure_Switch_Status	Evaluate_Signals	Input for switch_msg proc.	tbd	tbd	tbd	tbd

3.2.3.3.2 Output_Message1 = switch_msg:CAN_OUT

SRDE-724 -

Name	Type_and_direction	Sender	Receiver	Intended usage	Update_rate	States	Frame	Event
switch_msg	CAN_OUT	Evaluate_Signals	BCM_ECU	Shall be used to trigger the LED	200 ms	SWITCH_OPEN = Bit 0 is set SWITCH_CLOSED = Bit 1 is set SWITCH_ERROR = Bit 2 is set	DCU_Status	no

3.2.3.3.3 SF_Parameter = its_parameter()

SRDE-698 - (0..1) xx_CONFIG_PARAMETER = tbd

SRDE-693 - (0..*) xx_SF_PARAMETER = tbd

3.2.3.3.4 SF_Diag_Services = its_Diag_services();

SRDE-717 - (0..') xx_Diag_Service_Routines = tbd

3.2.3.4 Execution_rate = its_Autosar_rate()

SRDE-718 - Execution_rate = tbd

3.2.3.5 Signal_processing = switch_msg: SIGNAL_PROC

3.2.3.5.1 Relevant_Modes= its_mode_dependency()

(Note: All following IF requirements tables are automatically generated out of the defined Modes)

SRDE-715 -

SRS_P1_ECU

ECU_Modes			Mode Dependency
Life_Cycle_Phases	System_Modes	SW_Modes	
[(1..*)ECU_production_mode, (1..*)System_assembly_modes, (1..*)System_transport_modes, (1..*)System_transport_modes]	[(1..*)Imported_System_Modes = tbd]	[Start_up_mode, (1..*)Operation_mode, (0..*)Energy_safe_mode, (0..*)Fail_safe_mode, (0..*)Diagnostic_mode, Power_down_mode]	
tbd	tbd	tbd	SF ENABLED
tbd	tbd	tbd	SF DISABLED

3.2.3.5.2 Processing_Goals= its_processing_goals()

(Note: Here comes an automatic created list of all processing goals that are connected to this SF and shall be intentionally

defined in the following logic_table)

SRDE-708 - Processing_Goal1 = The switch status shall be provided on the CAN bus

3.2.3.5.3 Logic_Table= its_logic_table()

(Note: the headings of the table is automaticly created due to the

- Signal_Processing its connected IO and error signals,
- Potential configuration parameters
- Modes in which the SF is enabled
- Potential usage as state machine

The table contents must be finalized manually.)

SRDE-697 - switch_msg_its_logic_table

(0..1) CONFIG_PARAM [enable, disable]	LIFE_CYCLE_PHASE [tbd]	SYSTEM_MODES [tbd]	SW_MODES [tbd]	measured_switch_signal [tbd]	Result: switch_status [tbd]	(0..*) Follow up State	(0..*) Condition for transition
tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd

3.2.3.5.4 Functional_Requirements = switch_status_its_text_conversion

(Note: Those textual requirements are automatically created out of the logic table)

SRDE-700 - tbd

SRS_P1_ECU

3.2.3.6 *OUT_msg_handling = switch_msg:CAN_OUT_BUS*

SRDE-729 - MSG_Update_Rate = 200 ms

SRDE-730 - MSG_Sending_criteria = Key_Pos1-3

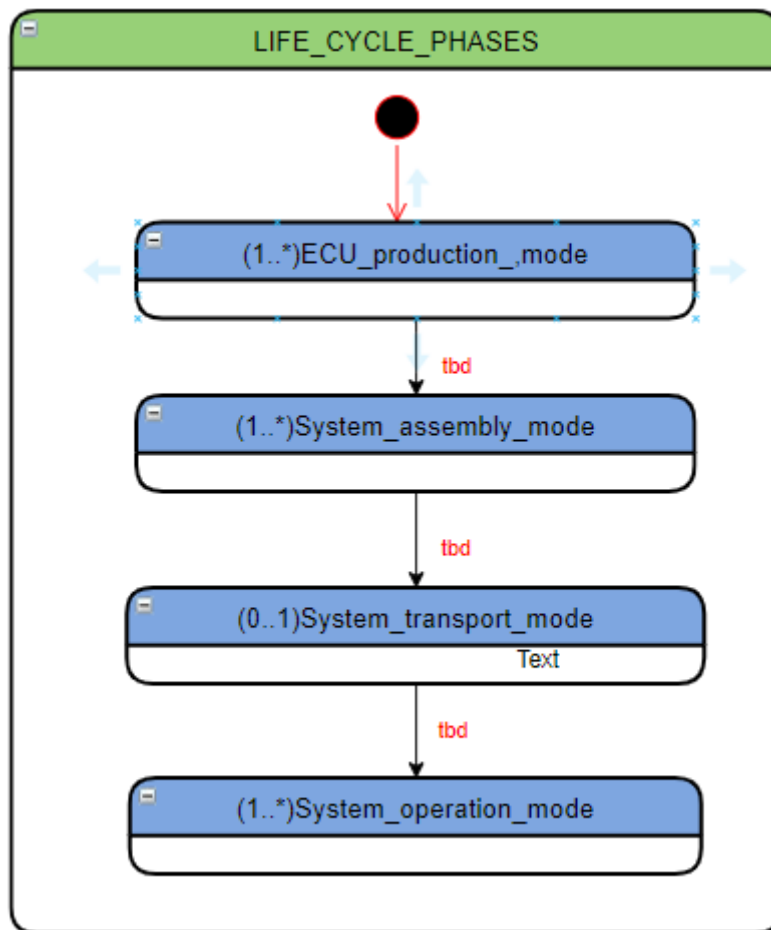
SRDE-731 - MSG_Repeation_ = no repetition

3.3 Supplement_Functions = P1_SUP

Scope = Overall supplement functions

3.3.1 Mode_Handling = ECU_Modes

3.3.1.1 *State_Diagram = its_state_diagram()*



3.3.1.2 *Life_Cycle_Phases = Life_cycle_phases*

3.3.1.2.1 (1..*)ECU_production_mode = xx_MODE

SRDE-752 - Mode_Entry = tbd

SRS_P1_ECU

SRDE-753 - (1..n)Mode_Task; = **tbd**

SRDE-754 - (0..n)SubModes = **tbd**

SRDE-755 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.2.2 (1..*)System_assembly_mode = xx_MODE

SRDE-756 - Mode_Entry = **tbd**

SRDE-757 - (1..n)Mode_Task; = **tbd**

SRDE-759 - (0..n)SubModes = **tbd**

SRDE-761 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.2.3 (0..1)System_transport_mode = xx_MODE

SRDE-751 - Mode_Entry = **tbd**

SRDE-750 - (1..n)Mode_Task; = **tbd**

SRDE-747 - (0..n)SubModes = **tbd**

SRDE-746 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.2.4 (1..*)System_operation_mode= xx_MODE;

SRDE-749 - Mode_Entry = **tbd**

SRDE-748 - (1..n)Mode_Task; = **tbd**

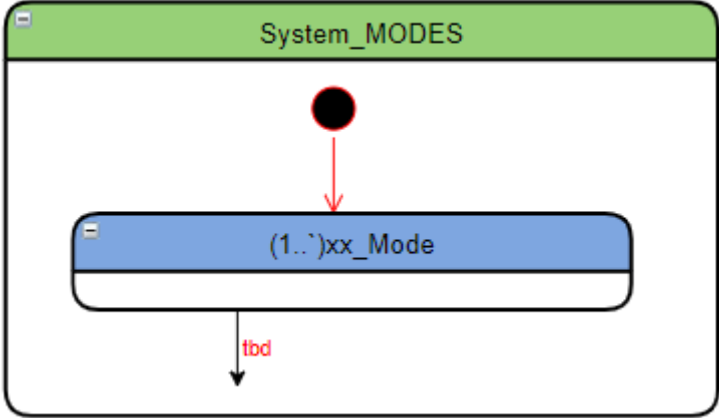
SRDE-758 - (0..n)SubModes = **tbd**

SRDE-760 - (1..n)Mode_Exit = *condition* --> *exit_mode*

SRS_P1_ECU

3.3.1.3 System_Modes = Imported_system_modes()

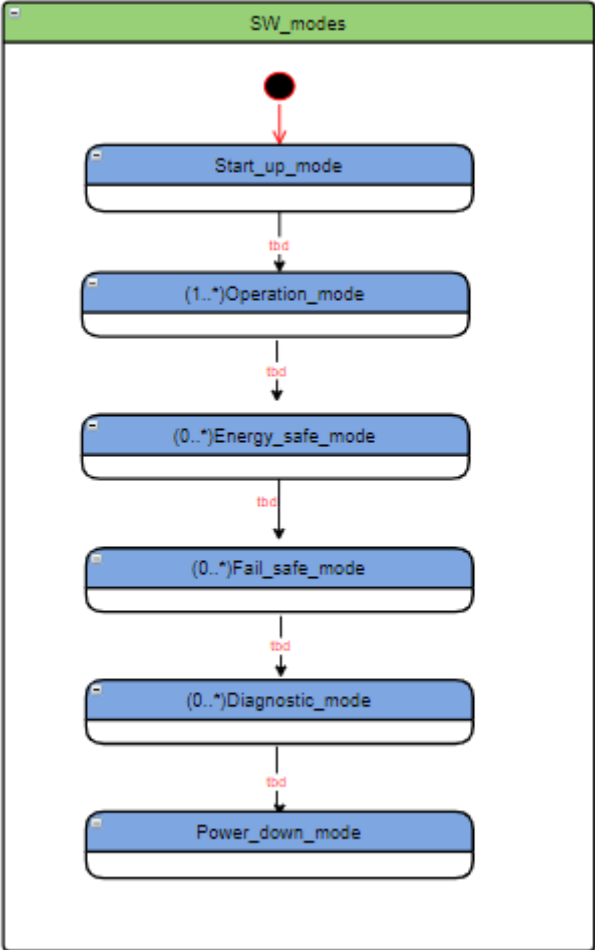
3.3.1.3.1 State_Diagram = its_state_diagram()



SRDE-802 - tbd

3.3.1.4 SW_Modes = SW_modes

3.3.1.4.1 State_Diagram = its_state_diagram()



SRS_P1_ECU

3.3.1.4.2 Start_up_mode= xx_MODE

SRDE-779 - Mode_Entry = **tbid**

SRDE-780 - (1..n)Mode_Task; = **tbid**

SRDE-781 - (0..n)SubModes = **tbid**

SRDE-782 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.4.3 (1..*)Operation_mode= xx_MODE

SRDE-783 - Mode_Entry = **tbid**

SRDE-784 - (1..n)Mode_Task; = **tbid**

SRDE-785 - (0..n)SubModes = **tbid**

SRDE-787 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.4.4 (0..*)Energy_safe_mode = xx_MODE

SRDE-789 - Mode_Entry = **tbid**

SRDE-777 - (1..n)Mode_Task; = **tbid**

SRDE-775 - (0..n)SubModes = **tbid**

SRDE-770 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.4.5 (0..*)Fail_safe_mode = xx_MODE

SRDE-768 - Mode_Entry = **tbid**

SRDE-773 - (1..n)Mode_Task; = **tbid**

SRDE-772 - (0..n)SubModes = **tbid**

SRDE-788 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.4.6 (0..*)Diagnostic_mode = xx_MODE

SRDE-786 - Mode_Entry = **tbid**

SRDE-791 - (1..n)Mode_Task; = **tbid**

SRDE-790 - (0..n)SubModes = **tbid**

SRS_P1_ECU

SRDE-774 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.1.4.7 Power_down_mode= xx_MODE

SRDE-778 - Mode_Entry = *tbd*

SRDE-776 - (1..n)Mode_Task; = *tbd*

SRDE-771 - (0..n)SubModes = *tbd*

SRDE-769 - (1..n)Mode_Exit = *condition* --> *exit_mode*

3.3.2 UC_Features = xx_UC_Features

(Note: The following Standard μ Controller Features are undefined, as no uC is selected)

Scope = Definition oh Standard μ Controller Features

3.3.2.1 Frequency

SRDE-853 - Frequency = *tbd*

3.3.2.2 Housing

SRDE-844 - Housing = *tbd*

3.3.2.3 NVM

SRDE-848 - NVM = *tbd*

3.3.2.4 RAM

SRDE-837 - RAM = *tbd*

3.3.2.5 EEPROM

SRDE-841 - EEPROM = *tbd*

3.3.2.6 Dig_IO_init

SRDE-834 - Dig_IO_init = *tbd*

3.3.2.7 ADC_init

SRDE-827 - Sampling_Frequency = *tbd*

SRDE-826 - Sampling_Mode = *tbd*

SRDE-829 - AD_Resolution = *tbd*

SRS_P1_ECU

SRDE-828 - Reference_Voltage = **tbd**

3.3.2.8 CAN_IF_init

SRDE-822 - Operation frequency = **tbd**

SRDE-825 - Protocol = **tbd**

SRDE-824 - Buffer_size = **tbd**

SRDE-852 - Buffer_location = **tbd**

SRDE-855 - Overflow_handling = **tbd**

3.3.2.9 PWM_IF_init

SRDE-847 - PWM_init = **tbd**

3.3.2.10 TIC_IF_init

SRDE-851 - TIC_IF_init = **tbd**

3.3.2.11 I2C_IF_init

SRDE-840 - I2C_IF_init = **tbd**

3.3.2.12 SPI_IF_init = All SPIs

SRDE-843 - SPI_init = **tbd**

3.3.2.13 UART_IF_init = All UARTSs

SRDE-830 - UART_IF_init = **tbd**

3.3.2.14 Interrupt_init

SRDE-833 - Interrupt_init = **tbd**

3.3.3 Operating_System = AUTOSAR

Scope = Autosar specific Properties

3.3.3.1 (1..)Service_Layer*

SRDE-811 - **tbd**

3.3.3.2 (1..)ECU_Abstraction_Layer*

SRDE-812 - **tbd**

SRS_P1_ECU

3.3.3.3 (1..)Microcontroller_Abstraction_Layer*

SRDE-813 - tbd

3.3.3.4 (0..)Complex_Devices*

SRDE-814 - tbd

3.3.3.5 (1..)RTE*

SRDE-815 - tbd

3.3.3.6 (1..)Application_Layer*

SRDE-816 - tbd

3.3.4 (1..)Ext_Power_Supply = XX:Ext_POWER_SUPPLY*

Scope = Power Supply specific Requirements

SRDE-809 - Nominal_Voltage =tbd

SRDE-807 - Max_Voltage =tbd

SRDE-803 - Min_Voltage_for_full_ECU_operation =tbd

SRDE-801 - Max_Peak_Consumption =tbd

SRDE-805 - Availibilty =tbd

SRDE-804 - Connector_pin=tbd

SRDE-819 - GND_Connection =tbd

SRDE-818 - Disturbances =tbd

3.3.4.1 (1..)Power_Consumption_Mode = XX:Power_Mode*

Scope = Power Mode specific Requirements

SRDE-821 - Max_current =tbd

SRDE-820 - Condition_for_Mode_Entry = tbd

SRDE-806 - Max_Mode_Readiness_Time =tbd

SRDE-810 - Activated_Functions = tbd

SRS_P1_ECU

3.3.5 (1..*)Int_Power_Supply

SRDE-808 - tbd

3.3.6 (0..*)BUS_Handling

SRDE-793 - tbd

3.3.7 (0..*)DIAG_Handling

SRDE-794 - tbd

3.3.8 (0..*)NVM_Handler

SRDE-817 - tbd

4 Nonfunctional Requirements

4.1 Quality_Req

SRDE-513 - The system shall have high quality.

4.2 Reusability_Req

SRDE-514 - All components of the system shall be reusable.

4.3 Criticalty

SRDE-656 - Crticality = tbd

5 Abbreviations = Used_Abbreviations

P1 = "Project 1 (Example Project)"

BCP = "Body Control Platform"

ECU = "Electronic Control Unit"

IF = "Interface"

SF = "System Function"

SRS_P1_ECU

SRD = "System Requirement Description"

SUPL = "Supplement"

UC = "Microcontroller"

PLCC = "Plastic Leaded Chip Carrier"

NVM = "Non-volatile memory"

RAM = "Random access memory"

EEPROM = "electrically erasable programmable read only memory"

Dig_IO = "Digital Input Output"

ADC = "Analog Digital converter"

CAN = "Controller Area Network"

tbd = "To be defined"

PWM = "pulse wide modulation"

I2C = "Inter-Integrated Circuit"

SPI = "Serial Peripheral Interface"

UART = "Universal Asynchronous Receiver Transmitter"

init = "initialization"

req = "requirement"

NF_Req = "non-functional requirement"