

/GDL_Playground/David Ricardo Cruz Juarez

Integration Test Plan

Template for the integration test plan

Contents

| | | |
|----------|--|----------|
| 1 | Integration Test Environment Description Global | 1 |
| 1.1 | ITS_TestCase_1 | 1 |
| 1.2 | ITS_TestCase_2 | 1 |
| 1.3 | ITS_TestCase_3 | 1 |
| 1.4 | ITS_TestCase_4 | 2 |
| 1.5 | ITS_TestCase_5 | 2 |
| 1.6 | ITS_TestCase_6 | 3 |
| 1.7 | ITS_TestCase_7 | 3 |
| 1.8 | ITS_TestCase_8 | 3 |
| 1.9 | ITS_TestCase_9 | 3 |
| 1.10 | ITS_TestCase_10 | 3 |

| ID | Template for the integration test plan | Requirements covered | Test_Description |
|----|---|---|---|
| 1 | 1 Integration Test Environment Description Global This module is to create 10 module test plan for the Window Control Module These test cases will be approved on a protoboard with the AURIX TC275_SB development board, with different buttons, B10R LED bar, and or VALUECAN4 connector | | |
| 12 | 1.1 ITS_TestCase_1 | | |
| 25 | Test_Architecture_Reference See Figure 1 on page 5. | 2.4.9 Window Position Determination SWR_171 2.4.9.1WINDOW_COMPLETELY_OPEN | Function used to report current Window Position to the network. |
| 11 | 1.2 ITS_TestCase_2 | 2.4.9 Window Position Determination 2.4.9.2WINDOW_COMPLETELY_CLOSE | Function used to report current Window Position to the network. |

| Test_Objective | Test Functionalit y | Expected_Res ults | Test_Fit_Criteria_and_test_exec ution_results |
|---|--|--|--|
| | | | |
| | | | |
| To verify if the window position is determined correctly when the SW_WINDOW_OPEN and SW_WINDOW_CLOSE states are determined as SW_INACTIVE. | Function used to determine current WINDOW_POSITION_COMPLETY_OPEN and report to other Functions. | The window position is determined correctly like Window Completely Open when SW_WINDOW_OPEN & SW_WINDOW_CLOSE states are determined as SW_INACTIVE | Is necessary the LEDs Bar B10R is completely turn off. |
| To verify if the window position is determined as COMPLETELY_CLOSE if SW_WINDOW_OPEN is determined as SW_ACTIVE and SW_WINDOW_CLOSE is determined as SW_ACTIVE. | Function used to determine current WINDOW_POSITION_COPLETY_CLOSE moreover shall report to other Functions. | The window shall be considered as COMPLETELY_CLOSE if SW_WINDOW_OPEN is determined as SW_ACTIVE & SW_WINDOW_CLOSE is determined as SW_ACTIVE | It is necessary the LEDs Bar B10R is completely turn on. |

| Test_Result | Comments | References |
|-------------|----------|------------|
| | | |
| | | |
| Passed | N/A | |
| Passed | N/A | |

| ID | Template for the integration test plan | Requirements covered | Test_Description |
|----|---|--|---|
| 27 | This picture is reference the ITS_TestCase 1,2,3,4 See Figure 2 on page 6. | 2.4.9 Window Position Determination 2.4.9.3 WINDOW_IN_BETWEEN | Function used to report current the window position to network, when the status is SW_WINDOW_OPEN equals 1 but SW_WINDOW equals 0 |
| 9 | 1.4 ITS_TestCase_4 | 2.4.9 Window Position Determination 2.4.9.3 WINDOW_ERROR | This function used for window show errors with relationship Close or Open |
| 29 | It is a reference for the antipinch ITS_TestCase_5 See Figure 3 on page 7. | | |
| 8 | 1.5 ITS_TestCase_5 | Antipinch Operation | Anti-Pinch Event Detected during Close Window Actuation furthermore Window is being closed, ANTIPINCH_SIGNAL is active |

| Test_Objective | Test Functionalit y | Expected_Res ults | Test_Fit_Criteria_and_test_exec ution_results |
|--|--|---|--|
| To verify If SW_WINDOW_OP EN is determined as SW_ACTIVE and SW_WINDOW_CL OSE is determined as SW_INACTIVE, then Window shall be considered as IN_BETWEEN. | Function used to determine current window position is middle SW_WINDOW _OPEN is activate && SW_WINDOW _CLOSE is not activate | The window shall considerer WINDOW_IN_B ETWEEN when the SW_WINDOW- OPEN is activate && SW_WINDOW_C LOSE is not activate | It is necessary that the LEDs Bar B10R shall be one led above the middle and at least the last permanent LED off. |
| Verify if WINDOW_ERROR when SW_WINDOW_CL OSE is determined as SW_ACTIVE and SW_WINDOW_OP EN is determined as SW_INACTIVE during 10 samples of 10 ms concurrently each. | The position window is no possible active Window Close if Window Open is desactive. | Window shall be considered as WINDOW_ERRO R if SW_WINDOW_C LOSE is SW_ACTIVE and SW_WINDOW_ OPEN is SW_INACTIVE, it is no probably that happen, because, the state window the SW_OPEN or SW_CLOSE can not jump state. | It is necessary that the LED bar remains off and only the last LED remains on. |
| | | | |
| When one hand or perturbation is detected on window and the CLOSE_WINDOW_ ACTUATION is active. and moreover verify the value analog input and | This action is for the security and the antipinch shall stop window | The system detects the anti-pinch event, cancels the current window actuation, executes GLOBAL_OPEN _ACTUATION and inhibits further window | For this, it is necessary to connect a phototransistor that detects the presence and makes active the STOP in the CLOSE_WINDOW_ACTUATION. |

| Test_Result | Comments | References |
|-------------|--------------------------|------------|
| Passed | N/A | |
| Error | N/A | |
| | | |
| Passed | the verify is no correct | |

| ID | Template for the integration test plan | Requirements covered | Test_Description |
|----|--|---|--|
| 7 | 1.6 ITS_TestCase_6 | Antipinch Operation | Anti-Pinch Event Detected during Global Close Window Actuation Multiple windows are being closed using GLOBAL_CLOSE_WINDOW_ACTUATION, ANTIPINCH_SIGNAL is active |
| 6 | 1.7 ITS_TestCase_7 | Antipinch Operation | Anti-Pinch Event Not Detected During Open Window Actuation when Window is being opened, ANTIPINCH_SIGNAL is inactive |
| 5 | 1.8 ITS_TestCase_8 | 2.4.10 Window Position Report Integration Test | The window position report the lock mechanism of each door (driver, passenger, rear left, and rear right) using the corresponding electronic lock. |
| 32 | 1.9 ITS_TestCase_9 | 2.4.7 Remote Operation Mode | All doors locked via CAN remotely |

| Test_Objective | Test Functionalit y | Expected_Res ults | Test_Fit_Criteria_and_test_exec ution_results |
|---|---|---|--|
| To verify the system detected the anti-pinch event, cancel all until WINDOW_ACTUATION execute GLOBAL_OPEN_ACTUATION, and around 15 seconds. | The function execute for when the window execute GLOBAL_WINDOW_ACTUATION shall stop when system detected Antipinch_Event | The system detects the anti-pinch event, cancels all ongoing WINDOW_ACTUATION, executes GLOBAL_OPEN_ACTUATION, and inhibits further window closure for 15 seconds. | This action shall be that execute CLOSE_WINDOW_ACTUATION and GLOBAL_CLOSE_WINDOW_ACTUATION are cancelled immediately |
| When the system execute the WINDOW_OPEN_ACTUATION, the system Antipinch shall activate. | When the GLOBAL_WINDOW_ACTUATION and OPEN_WINDOW_ACTUATION is execute the Antipinch event must inactive so, in the Led Bar B10R when LEDs turn off in descending order. | The system does not detect an anti-pinch event during GLOBAL_WINDOW_ACTUATION and the process continues uninterrupted. | This action Antipinch_Event must remain inactive when the push button open action the OPEN_WINDOW_ACTUATION moreover GLOBAL_OPEN_WINDOW_ACTUATION |
| To verify that the Window Position Report feature accurately reports the determined lock status of each door via the correct DCU. | The vehicle has four doors, each with their corresponding DCU. The Window Position Report feature is installed and functioning properly. | The Window Position Report correctly the variable the lock status for each door. The Window Position Report feature functions correctly for each door and its corresponding | The WindowPos variable of each DCU accurately reports the lock status of each door. The reported lock status in the WindowPos variable matches the actual lock status of the door. |
| Verify that all doors can be locked | When a signal send via CAN | All doors be locked remotely | Send a door lock request from BCM to the doors via CAN network. |

| Test_Result | Comments | References |
|-------------|----------|------------|
| Passed | N/A | |
| Passed | N/A | |
| Incomplete | N/A | |
| Incomplete | N/A | |

| ID | Template for the integration test plan | Requirements_ covered | Test_Descripti on |
|----|--|--------------------------|----------------------|
| 4 | | | |

| Test_Objective | Test Functionalit y | Expected_Res ults | Test_Fit_Criteria_and_test_exec ution_results |
|----------------|---------------------------|---|--|
| UnlockAllCmd. | | UnlockAllCmd. and moreover the OPEN_WINDOW _FUNTION executed. | |

| Test_Result | Comments | References |
|-------------|----------|------------|
| | | |

Figure 1: From object 25 on page 1.

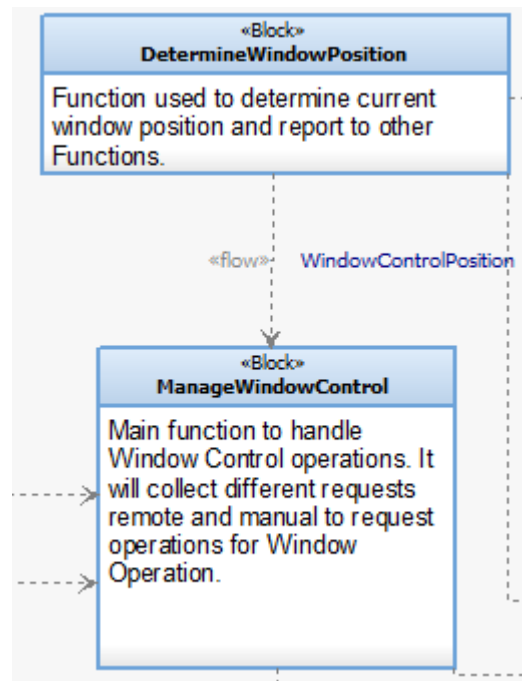


Figure 2: From object 27 on page 1.

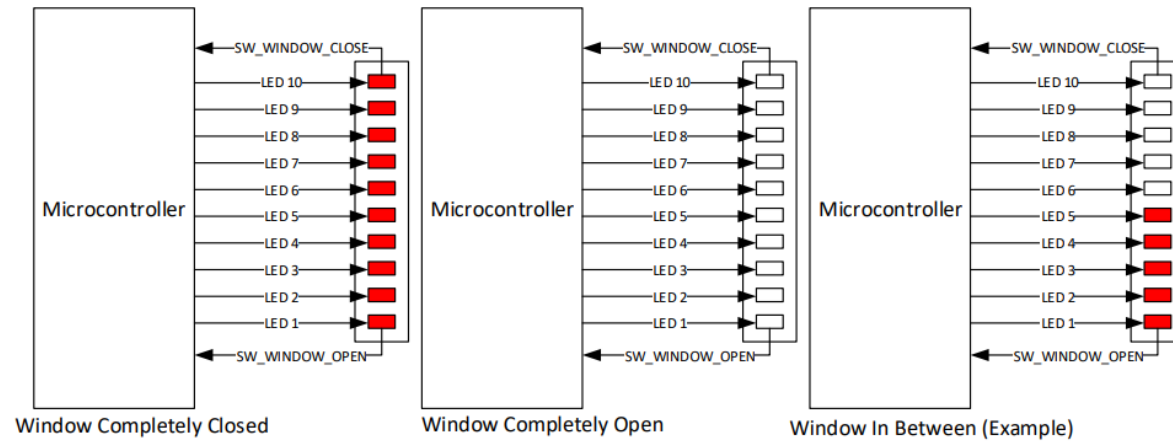
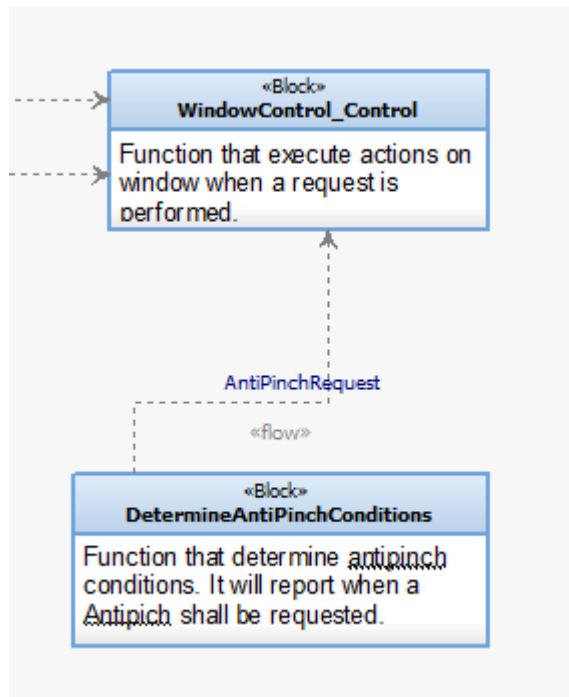


Figure 3: From object 29 on page 2.



/GDL_Playground/David Ricardo Cruz Juarez

Software Test Plan

Template for the Software Test Plan

Baseline: 1.0

Printed by: UIF87839

Printed on: Sunday, April 23, 2023

Contents

| | | |
|----------|--|----------|
| 1 | Software Test Environment Global Plan | 1 |
| 1.1 | SVT_TestCase_1 | 1 |
| 1.2 | SVT_TestCase_2 | 1 |
| 1.3 | SVT_TestCase_3 | 1 |
| 1.4 | SVT_TestCase_4 | 2 |
| 1.5 | SVT_TestCase_5 | 2 |

| ID | Template for the Software Test Plan | Test_Description | Test_Objective | Test_Requirement_Details |
|----|---|------------------------------|---|---|
| 2 | 1 Software Test Environment Global Plan These test cases will be approved on a protoboard with the AURIX TC275_SB development board, with different buttons, B10R LED bar, Buttons and circuit, moreover or VALUECAN4 connector | | | |
| 8 | 1.1 SVT_TestCase_1 | OPEN_WINDOW_ACTUATION | To verify that the OPEN_WINDOW_ACTUATION function follows the specified requirements and that each ECU correctly reports the window operations during the sequence. | When SHORT_Push_Open send the signal trough DIO ports and when signal is received for the Leds_Bar B10R must increased led for each pulse. |
| 7 | 1.2 SVT_TestCase_2 | GLOBAL_OPEN_WINDOW_ACTUATION | Verify that the GLOBAL_OPEN_WINDOW_ACTUATION function is executed correctly and that each ECU reports it's corresponding window operation in the corresponding CAN frame. | Until WINDOW_POSITION is different COMPLETELY_OPEN. When the LONG_BUTTON_PRESS OR SHORT_BUTTON_PRESS invocate GLOBAL_OPEN_WINDOW_ACTUATION. |
| 6 | 1.3 SVT_TestCase_3 | Antipinch | Verify that the system can detect and report an Anti-pinch event using ANTIPINCH_SIGNAL input. | The system is configured to use ANTIPINCH_SIGNAL input. The threshold reference value is set to 820 ADC counts. The system is in CLOSE_WINDOW_ACTUATION or GLOBAL_CLOSE_WINDOW_ACTUATION. |

| Test_Funcionality | Expected_Results | Test_Fit_Criteria_and_test_execution_results | Test_Result |
|--|---|--|-------------|
| | | | |
| Verify that each ECU reports proper window operations on the corresponding CAN frame during the sequence. DCU_X shall report WINDOW_DOWN operation for the driver door. | The OPEN_WINDOW_ACTUATION function executes only when WINDOW_POSITION is not COMPLETELY_OPEN and not ERROR. Each ECU reports the proper window operation on the corresponding CAN frame during the sequence. The system behaves correctly for different sets of WINDOW_POSITION values. | System setup with all relevant ECUs properly installed and connected Test data sets to simulate different WINDOW_POSITION values Test framework to log the required CAN frames. | Passed |
| Verify that ECU reports proper the Window operations on the corresponding CAN frame during the sequence Moreover each DCU_X Windows must report WINDOW_DOWN. | The GLOBAL_OPEN_WINDOW_ACTUATION function runs correctly. Each window is moved to correctly programmed positions. Each ECU reports it's corresponding window operation in the corresponding CAN frame. | All windows are closed and that WINDOW_POSITION is in a different position from WINDOW_COMPLETY_OPEN and ERROR. Configure the system to simulate the GLOBAL_OPEN_WINDOW_ACTUATION function moreover each ECU has reported it's corresponding window operation in the corresponding CAN frame. | |
| If ANTIPINCH_SIGNAL is present then an immediate CANCEL_WINDOW_ACTUATION shall be executed. | The system should report the Anti-pinch event through ANTIPINCH_SIGNAL. The system should execute an immediate CANCEL_WINDOW_ACTUATION. The system should execute a | Anti-pinch event by transitioning the Analog input signal from below the threshold to above the threshold. The system reports the Anti-pinch event through ANTIPINCH_SIGNAL. System executes an immediate CANCEL_WINDOW_ACTUATION. | Passed |

| Comments | References |
|----------|------------|
| | |
| | |
| | |
| | |

| ID | Template for the Software Test Plan | Test_Description | Test_Objective | Test_Requirement_Details |
|-----------|--|--|--|--|
| 6 | | | | <p>ATION mode.</p> <p>The Manual Mode or Remote Operation for the corresponding Door is enabled.</p> |
| 5 | 1.4 SVT_TestCase_4 | Manual Mode SHOT_BUTTON_PRESS | To verify if the Driver Door and Passenger Door execute the correct actuation when the OPEN_BTN and CLOSE_BTN states are equal to SHORT_BTN_PRESS. | <p>Driver Door and Passenger Door shall execute GLOBAL_OPEN_WINDOW_ACTUATION when its OPEN_BTN state is equal to SHORT_BTN_PRESS and Driver Door and Passenger Door shall execute GLOBAL_CLOSE_WINDOW_ACTUATION when its CLOSE_BTN state is equal to SHORT_BTN_PRESS and t</p> |
| 4 | 1.5 SVT_TestCase_5 | Remote Operation of Door Locking | To verify that remote operation of door locking works correctly on All Doors | <p>Send a BCM request via CAN network to remotely lock all doors.</p> <p>Verify that all doors are locked and that the BCM_2.ConfortCmd signal is set to LockAllCmd.</p> |

| Test_Funcionality | Expected_Results | Test_Fit_Criteria_and_test_execution_results | Test_Result |
|--|--|--|-------------|
| | <p>GLOBAL_OPEN_ACTUATION.</p> <p>The CLOSE_WINDOW_ACTUATION and GLOBAL_CLOSE_WINDOW_ACTUATION should be inhibited for 15 seconds for Manual Mode or Remote Operation for the corresponding Door.</p> | <p>Verify that the system executes a GLOBAL_OPEN_ACTUATION.</p> <p>Wait for the GLOBAL_OPEN_ACTUATION to finish. Moreover</p> <p>CLOSE_WINDOW_ACTUATION and GLOBAL_CLOSE_WINDOW_ACTUATION are inhibited for 15 seconds for Manual Mode or Remote Operation for the corresponding Door.</p> | |
| <p>When GLOBAL_OPEN_WINDOW_ACTUATION is executed when it's OPEN_BTN state is equal to SHORT_BTN_PRESS the window shall open.</p> | <p>The Driver Door window shall open moreover</p> <p>The Passenger Door window should open.</p> | <p>The driver and passenger doors are closed.</p> <p>The Driver Door and Passenger Door execute the correct actuation when the OPEN_BTN and CLOSE_BTN states are equal to SHORT_BTN_PRESS, with no unexpected effects on other systems in the vehicle.</p> | Passed |
| <p>When the signal send via CAN network the doors shall respond locked and unlocked more over when the execution the WINDOW_POSITION needed to know the status window, shall WINDOW_CLOSE.</p> | <p>All doors are locked and unlocked correctly via remote operation.</p> <p>The OPEN_WINDOW_ACTUATION and CLOSE_WINDOW_ACTUATION are executed</p> | <p>The BCM and DCU 1 are connected and communicating via CAN network.</p> <p>The vehicle doors are unlocked and closed.</p> | Incomplete |

| Comments | References |
|----------|------------|
| | |
| | |
| | |

/GDL_Playground/David Ricardo Cruz Juarez

Unit Test Plan

Development test case plans: Modules

Contents

| | | |
|----------|---|----------|
| 1 | Test Cases Environment Global Plan | 1 |
| 2 | Module Test Cases | 2 |
| 2.1 | UTS_TestCase_1 | 2 |
| 2.2 | UTS_TestCase_2 | 2 |
| 2.3 | UTS_TestCase_3 | 2 |
| 2.4 | UTS_TestCase_4 | 2 |
| 2.5 | UTS_TestCase_5 | 3 |
| 2.6 | UTS_TestCase_6 | 3 |
| 2.7 | UTS_TestCase_7 | 3 |
| 2.8 | UTS_TestCase_8 | 3 |
| 2.9 | UTS_TestCase_9 | 4 |
| 2.10 | UTS_TestCase_10 | 4 |

| ID | Development test case plans: Modules | Requirements_covered | Test_Description |
|----|---|----------------------|------------------|
| 16 | <p>1 Test Cases Environment Global Plan</p> <p>This module is for the crate 10 module test plan for the Window Control Module</p> <p>These test cases will be approved on a protoboard with the AURIX TC275_SB development board, with different buttons, B10R LED bar, and or VALUECAN4 connector</p> | | |

| Test_Objective | Test_Design_Details | Test_Operation | Expected_Results |
|----------------|---------------------|----------------|------------------|
| | | | |

| Test_Fit_Criteria_and_test_execution_results | Test_Result | Comments | References |
|--|-------------|----------|--|
| | | | <p>Manual KIT_AURIX_TC275 _ARD_SB</p> <p>AURIX™ Development Studio Infineon Technologies AG Version: 1.9.0</p> |

| ID | Development test case plans: Modules | Requirements_covered | Test_Description |
|----|--------------------------------------|---|---------------------------------------|
| 2 | 2 Module Test Cases | | |
| 36 | 2.1 UTS_TestCase_1 | 2.4 Window Functionality SWR_130 | Verification Window Control Function |
| 6 | 2.2 UTS_TestCase_2 | 2.4.1 Window Control IDLE Report SWR_468 | Verify IDLE Window Operation |
| 7 | 2.3 UTS_TestCase_3 | 2.4.2 Open window actuation SWR_177 | Verify OPEN_WINDOW_ACTUATION Function |
| 8 | 2.4 UTS_TestCase_4 | 2.4.3 GLOBAL_OPEN_WINDOW_ACTUATION | Verify GLOBAL_OPEN_WINDOW_A |

| Test_Objective | Test_Design_Details | Test_Operation | Expected_Results |
|---|---|---|--|
| | | | |
| Ensure that the Window Control Function can control the corresponding Window DC motor to open and close the window. | On a protoboard, connect the B10R LED bar with its running lands and using the AURIX_TC275_ARD_SB card, declare 10 DIO output pins, and evaluate the states CLOSE_OPEN_WINDOW | 1.-Trigger Window Control Actuation. 2.-Verify that the window opens or closes accordingly. 3.-Verify that the ECU reports the correct status (WINDOW_UP or WINDOW_DOWN) on the corresponding CAN frame. | 2.-The window shall open/close as expected. 3.- ECU should report the correct status on the CAN frame. |
| Ensure that if there is not a Window Control Actuation, the ECU reports IDLE Window operation on the corresponding CAN frame. | When the action the Window is do nothing the LED Diode must turn on moreover the LEDS Bar B10R the state must save last state, furthermore, the transmission with frame CAN must show. | 1.- Do not trigger Window Control Actuation. 2.-Verify that the ECU reports IDLE Window operation on the corresponding CAN frame. | 1.-The WINDOW_CONTROL_ACTUATION do not shall 2.-The ECU should report IDLE Window operation on the CAN frame. |
| Ensure that the OPEN_WINDOW_ACTUATION function is executed correctly when there are transitions from Picture in descendant order. | The leds Bar B10R receive an indicators to descendant the must turn off, when pulse Short o Long with a button_descendant. Moreover the status window must report the operation on the corresponding CAN Frame. | 1.- Set WINDOW_POSITION to a value that is not COMPLETELY_OPEN or ERROR. 2.- Trigger OPEN_WINDOW_ACTUATION. 3.- Verify that the window moves downwards. 4.- Verify that each ECU reports it's window operation on the corresponding CAN frame. | 3.- The window should move downwards this action show in leds bar B10R. 4.-Each ECU should report it's window operation on the CAN frame. |
| Ensure that the GLOBAL_OPEN_WINDOW_ACTUATION | This function is executed when there are transitions from Picture in descendant | 1.- Set WINDOW_POSITION to a value that is not | 3.-The window shall move downwards until it is completely ope. |

| Test_Fit_Criteria_and_test_execution_results | Test_Result | Comments | References |
|--|-------------|----------|---|
| | | | /GDL_Playground/Hector Nicolas Munoz Hernandez |
| The leds B10R worked under definitions in the template, the individual led turn on o turn off under 500 ms. | Passed | | Manual KIT_AURIX_TC275 _ARD_SB |
| Each ECU shall report on the CAN frame DCU_X.WindowOp (CAN frame DCU_Y BYTE X) | | | rtCANCheck Version 2.1.2.0 |
| The buttons Down shall send signals in when press the down button the leds bar turn off in pulse short & long. | Passed | N/A | Manual KIT_AURIX_TC275 _ARD_SB rtCANCheck Version 2.1.2.0 |
| The button_close is necessary for the test case because when the window is not complete open and trigger | Passed | N/A | Manual KIT_AURIX_TC275 _ARD_SB rtCANCheck Version 2.1.2.0 |

| ID | Development test case plans: Modules | Requirements_covered | Test_Description |
|----|--------------------------------------|-------------------------------------|---|
| 9 | 2.5 UTS_TestCase_5 | 2.4.4 Close_Window_Actuation | Verify Close_Window_Actuation |
| 10 | 2.6 UTS_TestCase_6 | 2.4.5 Global close window actuation | Verify GLOBAL_CLOSE_WINDOW_ACTUATION |
| 11 | 2.7 UTS_TestCase_7 | 2.4.6 Cancel Window Actuation | Verify CANCEL_WINDOW_ACTUATION |

| Test_Objective | Test_Design_Details | Test_Operation | Expected_Results |
|--|---|--|--|
| Ensure that the GLOBAL_CLOSE_WINDOW_ACTUATION shall be executed only when WINDOW_POSITION is different from COMPLETELY_CLOSED and WINDOW_POSITION is different from ERROR. | This function is executed when there are transitions from Picture in ascendant order. There is a 500 ms delay between every transition from one picture to another. Sequence will stop when conditions are no longer present. | 1.- Set WINDOW_POSITION to a value that is not COMPLETELY_CLOSED or ERROR. 2.- Trigger CLOSE_WINDOW_ACTUATION. 3.- Verify that the window moves go up. 4.- Verify that each ECU reports its window operation on the corresponding CAN frame. | 3.-The window shall move go up until it is completely close. 4.-Each ECU should report it's window operation on the CAN frame. |
| Ensure that the CLOSE_WINDOW_ACTUATION shall be executed only when WINDOW_POSITION is different from COMPLETELY_CLOSED and WINDOW_POSITION is different than ERROR | This function is executed when there are transitions from Picture in ascendant order. There is a 500 ms delay between every transition from one picture to another. Sequence will stop when window is considered as COMPLETELY_CLOSED | 1.- Set WINDOW_POSITION to a value that is not COMPLETELY_CLOSED or ERROR. 2.- Trigger GLOBAL_CLOSE_WINDOW_ACTUATION. 3.- Verify that the window moves go up until it is completely close. 4.- Verify that each ECU reports it's window operation on the corresponding CAN frame. | 3.-The window shall move go up until it is completely close. 4.-Each ECU should report it's window operation on the CAN frame. |
| Ensure that the CANCEL_WINDOW_ACTUATION shall be executed to move the Window Control to IDLE. This actuation can be executed only when there is a Window | This function is executed when all the LED stop keeping current position | 1.- Get WINDOW_POSITION to value that is COMPLETELY_CLOSE OR OPEN OR ERROR. 2.- When de button press for the cancel the move window these stop. | 1.-The WINDOW_POSTION shall or COMPLETEL_CLOSE or COMPLETEL_OPEN 2.-The action when press a button cancel window shall stop move. |

| Test_Fit_Criteria_and_test_execution_results | Test_Result | Comments | References |
|---|-------------|----------|---|
| The transmission is worked under definition in requirements. | Passed | | Manual KIT_AURIX_TC275 _ARD_SB rtCANCheck Version 2.1.2.0 |
| The transmission and time to close complete is worked under definitions in requirements | Passed | N/A | Manual KIT_AURIX_TC275 _ARD_SB rtCANCheck Version 2.1.2.0 |
| The BUTTON_STOCK shall be push to cancel window move. | Passed | | Manual KIT_AURIX_TC275 _ARD_SB rtCANCheck Version 2.1.2.0 |

| ID | Development test case plans: Modules | Requirements_covered | Test_Description |
|----|--------------------------------------|---|-------------------------|
| 13 | 2.9 UTS_TestCase_9 | 2.4.7 Manual Mode 2.4.7.2 Long Button Press | Verify Long Button Pres |
| 14 | 2.10 UTS_TestCase_10 | 2.4.7 Manual Mode 2.4.7.2 Stuck Button Press | Verify Button Stuck |

| Test_Objective | Test_Design_Details | Test_Operation | Expected_Results |
|---|--|--|---|
| Ensure that the the Long_Short_Button with an input > 500 ms && < 10000 ms is considerer as Long_Button_Press | Button transitions from BTN_NOT_PRESSED to BTN_PRESSED and back to BTN_NOT_PRESSED within a time > DID_0111.LongBtnCfg and <= DID_0111.StuckBtnCfg | 1.-Connects the button to DIO board Auirx TC275 2.-Implement Debounce System 3.-Press Short_Button with a duration > 500 ms && <10000 ms | The action Long_Button_Press most respect the time about > 500 ms && < 10000 ms |
| Ensure that the the Stuck_Button with an input > 10000 ms is considere as Button_Stuck | Button transitions from BTN_NOT_PRESSED to BTN_PRESSED and back to BTN_NOT_PRESSED within a time > DID_0111.StuckBtnCfg. | 1.- Conect the button to DIO board Auirx TC275 2.- Implement Debounce Sytem 3.- Press Short_Button with a duration > 10000 ms | The action Short_Button_Press most respect the time about > 10000 ms |

| Test_Fit_Criteria_and_test_execution_results | Test_Result | Comments | References |
|---|-------------|----------|-------------------------------|
| The functionlaty button is necessary integrate with Debounce | Passed | N/A | Manual KIT_AURIX_TC275_ARD_SB |
| The functionality button is necessary integrate with Debounce | Passed | N/A | Manual KIT_AURIX_TC275_ARD_SB |