|  |  |
| --- | --- |
| **Title:** | **Door Control Module**  **SW Component Window** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **History** | | | | |
| **Issue status**  (Index) | **Maturity/Date**  (draft/invalid/valid)  (dd-mmm-yyyy) | **Author**  Department | **Check/Release**  Department | **Description** |
| 1.0 | Draft  02-Sep-21 | Berenice Jimenez | Berenice Jimenez | Creation of the document |

**Table of Contents**

[1 Purpose 3](#_Toc81759604)

[2 Definitions and abbreviations 3](#_Toc81759605)

[3 Realization constraints and targets 3](#_Toc81759606)

[4 SW Conceptual design 3](#_Toc81759607)

[5 SW Component internal breakdown 5](#_Toc81759608)

[5.1 Functional Decomposition 5](#_Toc81759609)

[*5.2* Function WINDOW\_OPERATION *Window\_Get\_Operation()* 6](#_Toc81759610)

[*5.3* Function WINDOW\_STATUS *Window\_Get\_Status ()* 6](#_Toc81759611)

[*5.4* Function void *Window\_Init ()* 7](#_Toc81759612)

[*5.5* Function void *Window\_Run ()* 7](#_Toc81759613)

[*5.6* Function void *Window\_RunSafetyPath ()* 7](#_Toc81759614)

[*5.7* Function void *Window\_Set\_Request (WINDOW\_REQUEST request)* 8](#_Toc81759615)

# Purpose

The purpose of this document is to describe the detailed design of the Window module from Door Control Module project and the relation with other modules in the project.

# Definitions and abbreviations

**Abbreviations**

DIO Digital Input Output

Hw Hardware

CAN Controller Area Network

**References**

|  |  |  |
| --- | --- | --- |
| **N°** | **Document name** | **Reference** |
| 1  2 | Proyecto\_DoorControlModule.docx  Traceability Matrix Template.xls |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Realization constraints and targets

The Window Module is responsible of getting and setting the values for window actuation operation, these will be handled by the application module.

# SW Conceptual design

Window Module will interact with:

* ADC module
* DIO module
* HwConfig

Diagram

Description automatically generated

# SW Component internal breakdown

## Functional Decomposition



**Function Description and Dynamic Behavior**

## Function WINDOW\_OPERATION Window\_Get\_Operation()

|  |  |
| --- | --- |
| **Description** | Interface used to indicate the current window operation. |
| **Parameter** | NA |
| **Return Value** | Type: WINDOW OPERATION  Possible values:   * WINDOW\_OPERATION\_IDLE 0 * WINDOW\_OPERATION\_UP 1 * WINDOW\_OPERATION\_DOWN 2 |
| **Precondition** | WindowApp status app periodic execution. |
| **Post condition** | WindowOp of every Window CAN message, will be updated. |
| **Error Conditions** |  |
| **Requirements** |  |

## Function WINDOW\_STATUS Window\_Get\_Status ()

|  |  |
| --- | --- |
| **Description** | Interface used to indicate the current window status. |
| **Parameter** | NA |
| **Return Value** | Type: WINDOW STATUS  Valid values:   * WINDOW\_POSITION\_OPEN 0 * WINDOW\_ POSITION\_1 1 * WINDOW\_ POSITION\_2 2 * WINDOW\_ POSITION\_3 3 * WINDOW\_ POSITION\_4 4 * WINDOW\_ POSITION\_5 5 * WINDOW\_ POSITION\_6 6 * WINDOW\_ POSITION\_7 7 * WINDOW\_ POSITION\_8 8 * WINDOW\_ POSITION\_9 9 * WINDOW\_ POSITION\_CLOSED 10 * WINDOW\_ POSITION\_ERROR 255 |
| **Precondition** |  |
| **Post condition** | WindowPos of every Window CAN message, will be updated. |
| **Error Conditions** | If Window position is OPEN and CLOSED at the same time, the window position will be determined as WINDOW\_POSITION\_ERROR |
| **Requirements** | DCU\_SWR\_111, DCU\_SWR\_112, DCU\_SWR\_113, DCU\_SWR\_114 |

## Function void Window\_Init ()

|  |  |
| --- | --- |
| **Description** | Interface function to perform Window SWC initialization |
| **Parameter** | NA |
| **Return Value** | NA |
| **Precondition** | Execution of WindowHandler\_Init after power-up. |
| **Post condition** | Window module is initialized and ready to use. Execute WindowHandler\_Run |
| **Error Conditions** | NONE |
| **Requirements** | NONE |

**Dynamic Behavior**

After power-up, the WindowApp module will be on charge of execute the Initialization API for Window module, this will be executed from the WindowHandler\_Init.

## Function void Window\_Run ()

|  |  |
| --- | --- |
| **Description** | Interface function to perform Window SWC periodic operations. |
| **Parameter** | NA |
| **Return Value** | NA |
| **Precondition** | Execution of WindowHandler\_Run |
| **Post condition** | Values for Window periodic operations. |
| **Error Conditions** | NONE |
| **Requirements** | NONE |

**Dynamic Behavior**

After WindowHandler\_Init finish own execution, the WindowApp module will be on charge of execute the Run API for Window module, this will be executed from the WindowHandler\_Run.

## Function void Window\_RunSafetyPath ()

|  |  |
| --- | --- |
| **Description** | Interface function to perform Window SWC periodic safety operations. |
| **Parameter** | NA |
| **Return Value** | NA |
| **Precondition** | Execution of WindowHandler\_Run |
| **Post condition** | ANTIPINCH\_SIGNAL value obtained from DIO module. |
| **Error Conditions** |  |
| **Requirements** | DCU\_SWR\_110 |

**Dynamic Behavior**

After WindowHandler\_Init finish own execution, the WindowApp module will be on charge of execute the Run API for Window module, this will be executed from the WindowHandler\_Run.

## Function void Window\_Set\_Request (WINDOW\_REQUEST request)

|  |  |
| --- | --- |
| **Description** | Interface function to indicate a Window Operation |
| **Parameter**  <input> | Type: WINDOW\_REQUEST  Valid values:   * WINDOW\_ REQUEST\_IDLE 0 * WINDOW\_ REQUEST\_UP 1 * WINDOW\_ REQUEST\_DOWN 2 |
| **Return Value** | NA |
| **Precondition** | After a Close/open event from any window |
| **Post condition** | Excution of corresponding Window\_Actuation function. |
| **Error Conditions** |  |
| **Requirements** | DCU\_SWR\_111, DCU\_SWR\_112, DCU\_SWR\_113, DCU\_SWR\_114 |

**Dynamic Behavior**