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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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](#_Toc81629178)

[2 Definitions and abbreviations 3](#_Toc81629179)

[3 Realization constraints and targets 3](#_Toc81629180)

[4 SW Conceptual design 3](#_Toc81629181)

[5 SW Component internal breakdown 3](#_Toc81629182)

[5.1 Functional Decomposition 3](#_Toc81629183)

[*5.2* Function HW\_CONFIG *HwConfig\_Get()* 5](#_Toc81629184)

[*5.3* Function Boolean *HwConfig\_IsAny (uint8 config, const uint8 config\_mask)* 5](#_Toc81629185)

# Purpose

The purpose of this document is to describe the SDD of the module HwConfig, part of the Door Control Module project.

# Definitions and abbreviations

**Definitions**

|  |  |
| --- | --- |
| *Jumper* | Short length of conductor used to close, open or bypass a electronic circuit. |
|  |  |
|  |  |

**Abbreviations**

DIO Digital Inputs Outputs

Hw Hardware

SDD Software Detailed Design

**References**

|  |  |  |
| --- | --- | --- |
| **N°** | **Document name** | **Reference** |
| 1 | SWA\_DCU.rpxy | *ARCH design doc* |
| 2 | Traceability Matrix Template.xlsm | *Req doc* |
|  |  |  |
|  |  |  |
|  |  |  |

# Realization constraints and targets

The HWConfig module is responsible of report the current HW configuration during DIO initialization.

# SW Conceptual design

DIO module will provide the status of the jumpers digital values and the HwConfig module will interpret it, in order to report the DCU HW configuration. This will be done just one time in the entire operating cycle.

WindowApp, Window and Button can require the HW configuration value for their own purposes.

# SW Component internal breakdown

## Functional Decomposition



Graphical user interface, diagram

Description automatically generated

**Function Description and Dynamic Behavior**

## Function HW\_CONFIG HwConfig\_Get ()

|  |  |
| --- | --- |
| **Description** | Interface to get the current HW variant. |
| **Parameter** | NA |
| **Return Value** | Type: HW\_CONFIG  Range:   * HWCONFIG\_UNKNOWN 0 * HWCONFIG\_DRIVER 1 * HWCONFIG\_PASSENGER 2 * HWCONFIG\_REAR\_LEFT 4 * HWCONFIG\_REAR\_RIGHT 8 |
| **Precondition** | Digital value for Jumper 1&2 must be read by DIO interface |
| **Post condition** | HW config values is set for the rest of the operating cycle |
| **Error Conditions** | The interface will return HWCONFIG\_UNKNOWN when the hw configuration is different of the stablished configuration |
| **Requirements** | DCU\_SWR\_152 |

**Dynamic Behavior**

During DIO initialization, the signal values of J0 and J1 will be acquired, and HwConfig\_Get will be interpreted and return the config value.

## Function Boolean HwConfig\_IsAny (uint8 config, const uint8 config\_mask)

|  |  |
| --- | --- |
| **Description** | Interface to check for a specific HW configuration. |
| **Parameter 1** <input> | Type: uint8  Name: config  Description: Configuration reference |
| **Parameter 2**  <input> | Type: uint8  Name: config\_mask  Description: Group of configurations to be compared |
| **Return Value** | Type: Boolean  True (1): config\_mask match with one of the valid Hw configurations.  False (0): config\_mask does not match with any valid configuration. |
| **Precondition** | DIO Initialization |
| **Post condition** | Specific module configuration |
| **Error Conditions** | This interface will return FALSE in case an invalid configuration. |
| **Requirements** | DCU\_SWR\_152 |

**Dynamic Behavior**

This interface will be called from the following modules: WindowApp, Window and Button, in order to set configuration accordingly Hw Configuration.