****

# PO1\_DGC Calculator

**(GDD)**

**Status**: **Draft**

### **Document Status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Status** | **Author** | **Date** |
| 1.0 | Draft | May Abdelsalam | 26/2/2020 |
| 1.0 | Draft | Moamen Ahmed | 27/2/2020 |
| 1.0 | Draft | May Abdelsalam | 27/2/2020 |
| 1.0 | Draft | Areej Ayman Helal | 27/2/2020 |
| 1.0 | Draft | Bishoy Nabil | 27/2/2020 |
| 1.1 | Proposed | Moamen Ahmed | 2/3/2020 |

**History Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| 1.0 | May Abdelelsalam | 26/2/2020 | Initial creation |
| 1.0 | Moamen Ahmed | 27/2/2020 | Added HAL APIs |
| 1.0 | May Abdelelsalam | 27/2/2020 | Update on Input Output signals, Static Architecture |
| 1.0 | Areej Ayman Helal | 27/2/2020 | Added App APIs. |
| 1.0 | Bishoy Nabil | 27/2/2020 | Added MCAL APIs |
| 1.1 | Moamen Ahmed | 2/3/2020 | Added covers section to all requirements |

**Reference documents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference number** | **Document name** | **Version** | **Status** |
| 1 | PO1\_DGC Calculator  (SRS) | V1.7 | Released |

## **Table of contents**

1. **Project Description.**
2. **Input Output signals.**
3. **Software features.**
4. **Static Architecture.**
5. **APP Components APIs.**
6. **HAL Components APIs.**
7. **MCAL Components APIs.**

## Project Description

The Digital Calculator system will consist of a Power component in the APP layer to turn all the system ON whenever the ON switch is pressed.

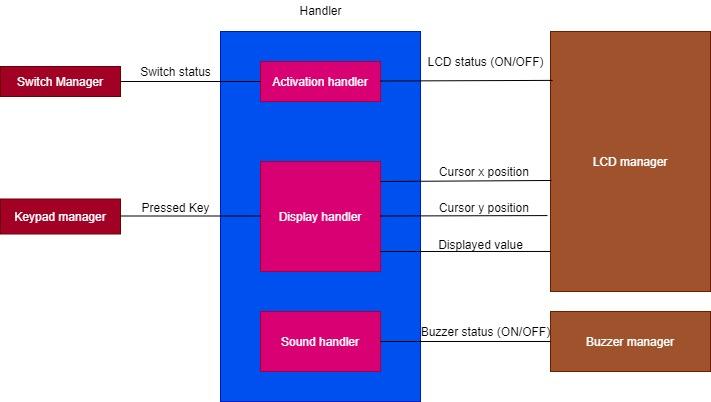
The system starts with the Keypad\_assignment component waiting for any of the Keypad keys to get pressed. When a switch of the keypad is pressed the Keypad\_assignment component will send signals to all of the following components:

1- Math\_calc: will calculate the entered operation and send the result to the Display component.

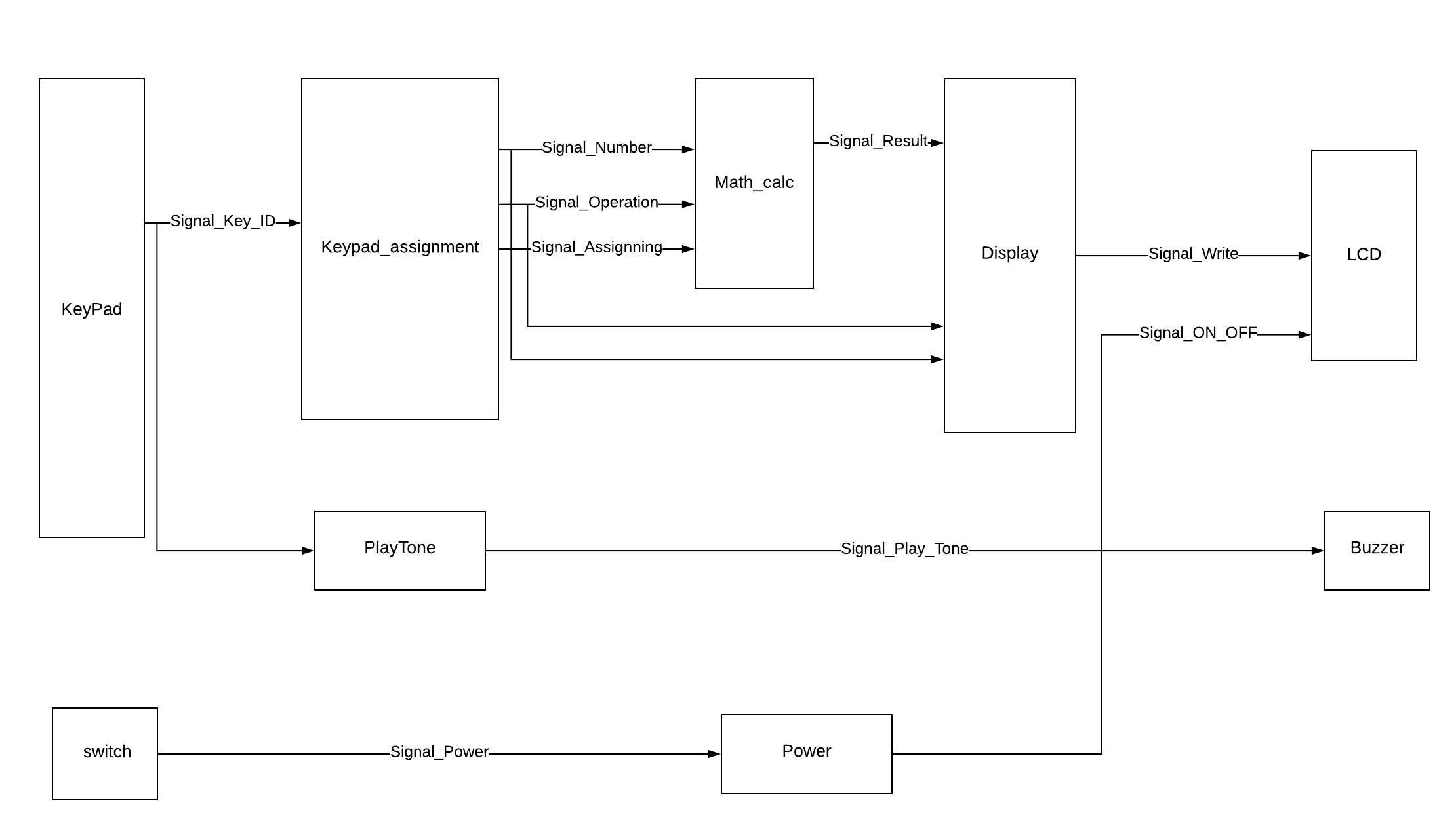
2- Display: will display the entered first operand, operation and second operand.

3- PlayTone: will play a tune every time keypad\_assignment component is used.

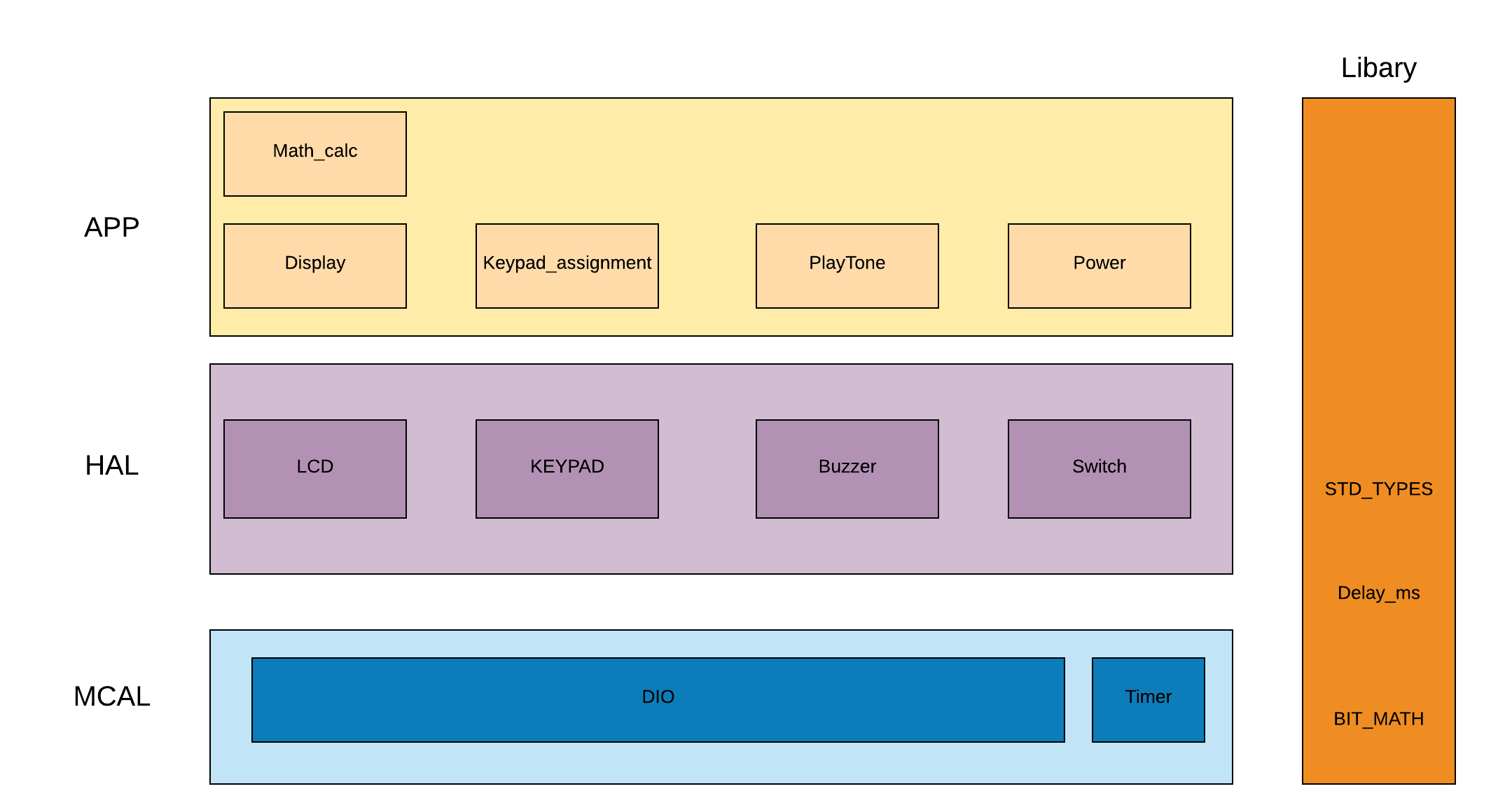
## Software context diagram:



**Input Output signals:**

****

**Static Architecture:**



### **APP Components APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_001 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8PressedKey  struct\* Copy\_structAssignedKeys |
| **Name** | DGC\_KeypadAssignment | | |
| **Description** | It takes the pressed key as an input and returns a struct of numbers and assignment operators. | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_001\_v1.3  Req\_PO1\_DGC\_SRS\_003\_v1.3  Req\_PO1\_DGC\_SRS\_005\_v1.3  Req\_PO1\_DGC\_SRS\_007\_v1.3  Req\_PO1\_DGC\_SRS\_008\_v1.3  Req\_PO1\_DGC\_SRS\_013\_v1.3  Req\_PO1\_DGC\_SRS\_014\_v1.3  Req\_PO1\_DGC\_SRS\_015\_v1.3  Req\_PO1\_DGC\_SRS\_016\_v1.3  Req\_PO1\_DGC\_SRS\_018\_v1.3  Req\_PO1\_DGC\_SRS\_019\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_002 V1.3 | | |
| **Return Type** | STD\_ERR | **Input arguments** | struct Copy\_structAssignedKeys  f32\* Copy\_f32Result |
| **Name** | DGC\_MathCalc | | |
| **Description** | It takes the struct of numbers and assignment operators and applies math operations on them , then returns the result . | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_0016\_v1.3  Req\_PO1\_DGC\_SRS\_0017\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_003 V1.3 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8Result , struct ,Copy\_structAssignedKeys . |
| **Name** | DGC\_Display | | |
| **Description** | It takes the struct of entered numbers and assignment operations , and the returned result from DGC\_u8MathCalc , And Displays all of them on the LCD. | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_002\_v1.3  Req\_PO1\_DGC\_SRS\_004\_v1.3  Req\_PO1\_DGC\_SRS\_006\_v1.3  Req\_PO1\_DGC\_SRS\_009\_v1.3  Req\_PO1\_DGC\_SRS\_010\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_004 V1.3 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8PressedKey |
| **Name** | DGC\_PlayTone | | |
| **Description** | It takes the pressed key as an input and applies a different tone on each key of them whenever it’s pressed. | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_011\_v1.3  Req\_PO1\_DGC\_SRS\_012\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_005 V1.3 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8PState |
| **Name** | DGC\_Power | | |
| **Description** | It turns the system on or off whether it's from the power button or if the system idles for 15 seconds . | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_020\_v1.3 | | |

### **HAL Components APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_006 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | void |
| **Name** | LCD\_Init | | |
| **Description** | Initialize LCD with determined configurations | | |
| **Covers** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_007 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8Command |
| **Name** | LCD\_WriteCommand | | |
| **Description** | Control the LCD using the commands found in the LCD datasheet | | |
| **Covers** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_008 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8Data |
| **Name** | LCD\_WriteData | | |
| **Description** | Write a single character to the LCD | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_002\_v1.3  Req\_PO1\_DGC\_SRS\_004\_v1.3  Req\_PO1\_DGC\_SRS\_006\_v1.3  Req\_PO1\_DGC\_SRS\_009\_v1.3  Req\_PO1\_DGC\_SRS\_010\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_009 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8\* Copy\_u8KeyPressed |
| **Name** | KEYPAD\_GetPressedKey | | |
| **Description** | Returns the current pressed key(1:16) or returns 0 if no key is pressed | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_001\_v1.3  Req\_PO1\_DGC\_SRS\_003\_v1.3  Req\_PO1\_DGC\_SRS\_005\_v1.3  Req\_PO1\_DGC\_SRS\_007\_v1.3  Req\_PO1\_DGC\_SRS\_008\_v1.3  Req\_PO1\_DGC\_SRS\_013\_v1.3  Req\_PO1\_DGC\_SRS\_014\_v1.3  Req\_PO1\_DGC\_SRS\_015\_v1.3  Req\_PO1\_DGC\_SRS\_016\_v1.3  Req\_PO1\_DGC\_SRS\_018\_v1.3  Req\_PO1\_DGC\_SRS\_019\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_010 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | void |
| **Name** | BUZZER\_SetBuzzerOn | | |
| **Description** | Turn on the buzzer by outputting a high signal on the buzzer pin | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_011\_v1.3  Req\_PO1\_DGC\_SRS\_012\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_011 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | void |
| **Name** | BUZZER\_voidSetBuzzerOff | | |
| **Description** | Turn off the buzzer by outputting a low signal on the buzzer pin | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_011\_v1.3  Req\_PO1\_DGC\_SRS\_012\_v1.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_012 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | void |
| **Name** | SWITCH\_GetSwitchStatus | | |
| **Description** | Returns 1 if the switch is pressed or 0 otherwise | | |
| **Covers** | - | | |

### **MCAL Components APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_013 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | void |
| **Name** | DIO\_Init | | |
| **Description** | Initialize all pins and ports to their operation mode and initial values | | |
| **Covers** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_014 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8Port u8 Copy\_u8Pin u8 Copy\_u8Mode |
| **Name** | DIO\_SetPinMode | | |
| **Description** | Checks which Port and which Pin and set its Mode if the function done its purpose return OK if not return NOK | | |
| **Covers** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_015 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8Port u8 Copy\_u8Pin u8 Copy\_u8Value |
| **Name** | DIO\_SetPinValue | | |
| **Description** | Checks which Port and which Pin and set it or reset it if the function done its purpose return OK if not return NOK | | |
| **Covers** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_016 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8Port u8 Copy\_u8Pin u8\* Copy\_u8ValuePtr |
| **Name** | DIO\_GetPinValue | | |
| **Description** | Checks which Port and which Pin and reads its value and returns it in “Copy\_u8ValuePtr” pointer if the function done its purpose return OK if not return NOK | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_017 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8TimerNum |
| **Name** | TIMER\_Init | | |
| **Description** | Checks which Timer that will be initialized and makes its initialization. if the function done its purpose return OK if not return NOK | | |
| **Covers** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_018 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8TimerNum u8 Copy\_u8Preload |
| **Name** | TIMER\_SetPreloadValue | | |
| **Description** | Checks which Timer and sets its preload value if the function done its purpose return OK if not return NOK | | |
| **Covers** | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_019 V1.1 | | |
| **Return Type** | STD\_ERR | **Input arguments** | u8 Copy\_u8TimerNum void (\*Copy\_CallBackPtr)(void) |
| **Name** | TIMER\_SetCallBack | | |
| **Description** | Checks which Timer and sets its Call Back Function. if the function done its purpose return OK if not return NOK | | |
| **Covers** | Req\_PO1\_DGC\_SRS\_0020\_v1.3 | | |