****

# PO1\_DGC Calculator

**(GDD)**

**Status**: **Draft**

### **Document Status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Status** | **Author** | **Date** |
| 1.0 | Draft | May Abdelsalam | 26/2/2020 |
| 1.1 | Draft | Moamen Ahmed | 27/2/2020 |
| 1.2 | Draft | May Abdelsalam | 27/2/2020 |
| 1.3 | Draft | Areej Ayman Helal | 27/2/2020 |
| 1.4 | Draft | Bishoy Nabil | 27/2/2020 |

**History Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| 1.0 | May Abdelelsalam | 26/2/2020 | Initial creation |
| 1.1 | Moamen Ahmed | 27/2/2020 | Added HAL APIs |
| 1.2 | May Abdelelsalam | 27/2/2020 | Update on Input Output signals, Static Architecture |
| 1.3 | Areej Ayman Helal | 27/2/2020 | Added App APIs. |
| 1.4 | Bishoy Nabil | 27/2/2020 | Added MCAL APIs |

**Reference documents**

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

## **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.

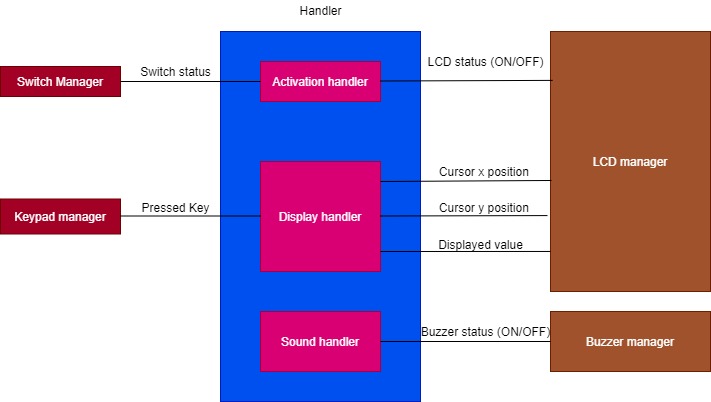
The system starts with the Keypad\_assignment component waiting for any of the Keypad keys to get pressed. When a switch of the keypad is pressed the Keypad\_assignment component will send signals to all of the following components:

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

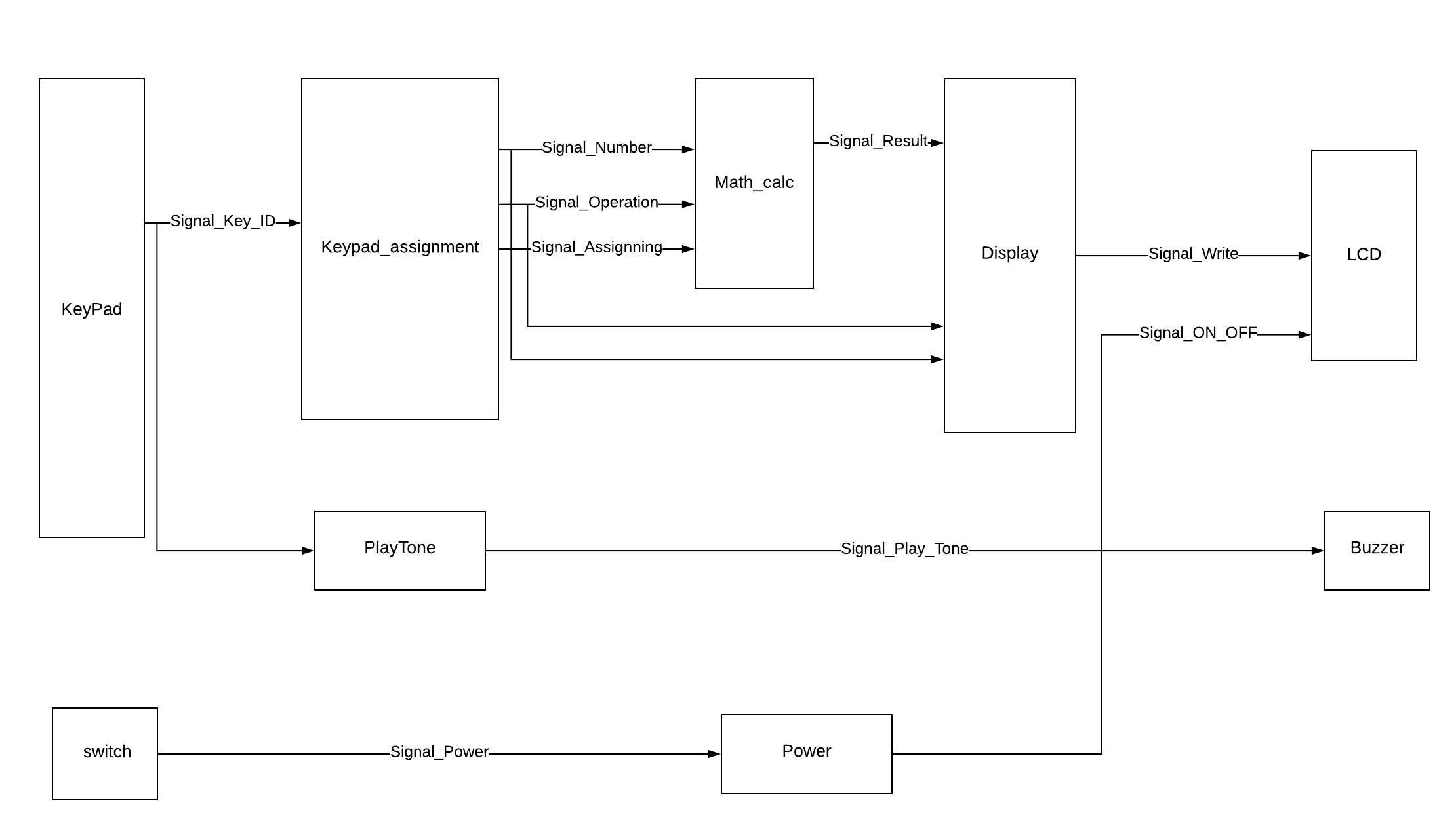
2- Display: will display the entered first operand, operation and second operand.

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

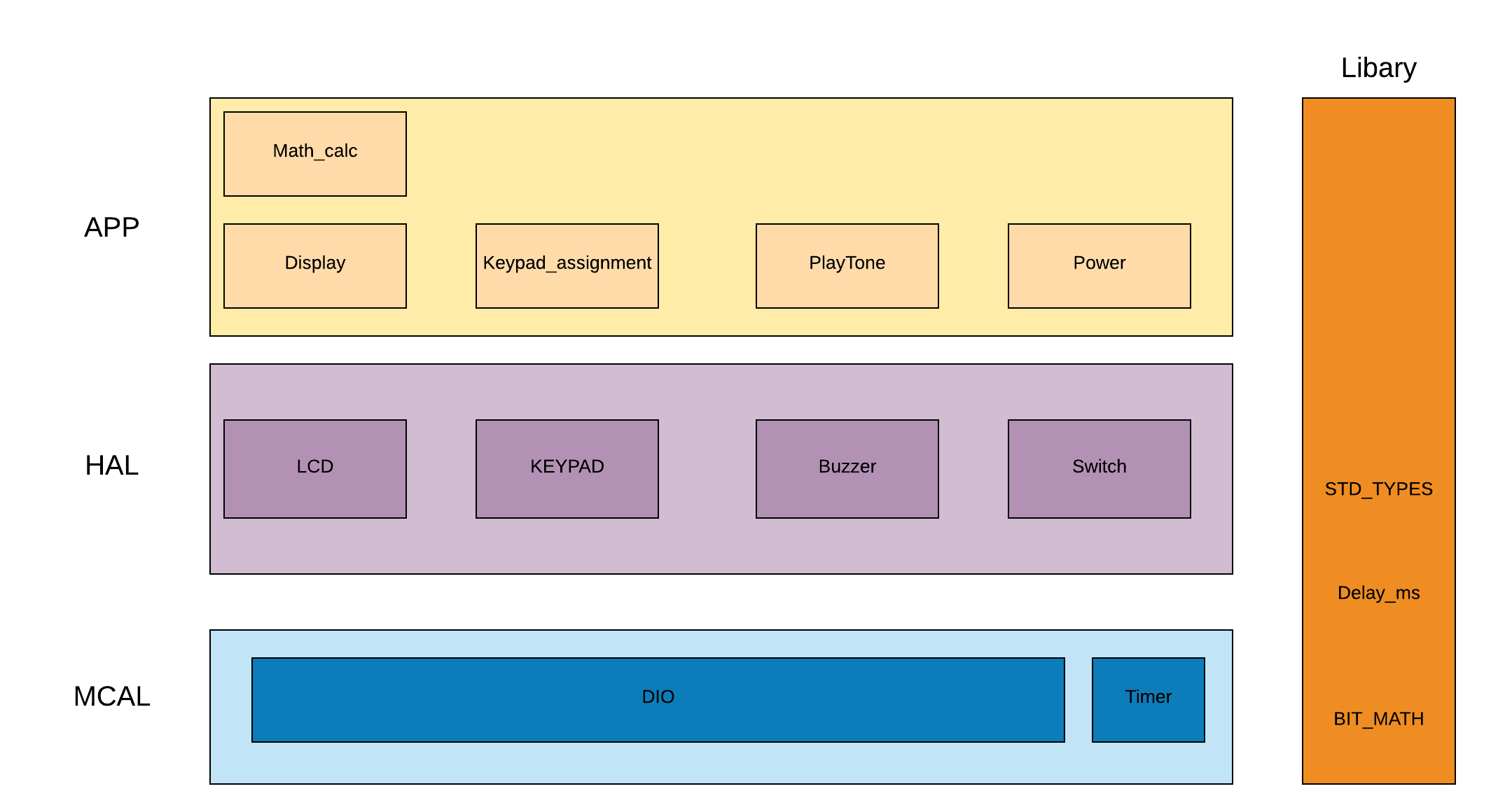
Software context diagram:



**Input Output signals:**

****

**Static Architecture:**



### **APP Components APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_APP\_Keypad\_assignment\_001 V1.3 | | |
| **Return Type** | struct | **Input arguments** | u8 Copy\_u8PressedKey |
| **Name** | DGC\_structKeypadAssignment | | |
| **Description** | It takes the pressed key as an input and returns a struct of numbers and assignment operators. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_APP\_Math\_Calc\_002 V1.3 | | |
| **Return Type** | u8 | **Input arguments** | struct Copy\_structAssignedKeys |
| **Name** | DGC\_u8MathCalc | | |
| **Description** | It takes the struct of numbers and assignment operators and applies math operations on them , then returns the result . | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_APP\_Display\_003 V1.3 | | |
| **Return Type** | void | **Input arguments** | u8 Copy\_u8Result , struct ,Copy\_structAssignedKeys . |
| **Name** | DGC\_voidDisplay | | |
| **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. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_APP\_Play\_Tone\_004 V1.3 | | |
| **Return Type** | void | **Input arguments** | u8 Copy\_u8PressedKey |
| **Name** | DGC\_voidPlayTone | | |
| **Description** | It takes the pressed key as an input and applies a different tone on each key of them whenever it’s pressed. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_APP\_Power\_005 V1.3 | | |
| **Return Type** | void | **Input arguments** | u8 Copy\_u8PowerPressedKey |
| **Name** | DGC\_voidPower | | |
| **Description** | It takes the power pressed key status as an input and then sets the system on or off depending on it . | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_APP\_Power\_006 V1.3 | | |
| **Return Type** | void | **Input arguments** | void |
| **Name** | DGC\_voidInit | | |
| **Description** | It initializes power pins to output. | | |

### **HAL Components APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_LCD\_001 V1.0 | | |
| **Return Type** | void | **Input arguments** | void |
| **Name** | LCD\_voidInit | | |
| **Description** | Initialize LCD pins to output and set the default configurations  (5:8, Blinking: off,Cursor: off, 2 lines: on ) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_LCD\_002 V1.0 | | |
| **Return Type** | void | **Input arguments** | u8 Copy\_u8Command |
| **Name** | LCD\_voidWriteCommand | | |
| **Description** | Control the LCD using the commands found in the LCD datasheet | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_LCD\_003 V1.0 | | |
| **Return Type** | void | **Input arguments** | u8 Copy\_u8Data |
| **Name** | LCD\_voidWriteData | | |
| **Description** | Write a single character to the LCD | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_LCD\_004 V1.0 | | |
| **Return Type** | void | **Input arguments** | u8 Copy\_u8Shape[8] |
| **Name** | LCD\_voidWriteToCGRAM | | |
| **Description** | Add a custom shape to CGRAM | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_KEYPAD\_001 V1.0 | | |
| **Return Type** | void | **Input arguments** | void |
| **Name** | KEYPAD\_voidInit | | |
| **Description** | Initialize keypad pins (1,2,3,4) to input and pins (5,6,7,8) to output | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_KEYPAD\_002 V1.0 | | |
| **Return Type** | u8 | **Input arguments** | void |
| **Name** | KEYPAD\_u8GetPressedKey | | |
| **Description** | Returns the current pressed key(1:16) or returns 0 if no key is pressed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_BUZZER\_001 V1.0 | | |
| **Return Type** | void | **Input arguments** | void |
| **Name** | BUZZER\_voidInit | | |
| **Description** | Initialize buzzer pin to output | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_BUZZER\_002 V1.0 | | |
| **Return Type** | void | **Input arguments** | void |
| **Name** | BUZZER\_voidSetBuzzerOn | | |
| **Description** | Turn on the buzzer by outputting a high signal on the buzzer pin | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_BUZZER\_003 V1.0 | | |
| **Return Type** | void | **Input arguments** | void |
| **Name** | BUZZER\_voidSetBuzzerOff | | |
| **Description** | Turn off the buzzer by outputting a low signal on the buzzer pin | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_SWITCH\_001 V1.0 | | |
| **Return Type** | void | **Input arguments** | void |
| **Name** | SWITCH\_voidInit | | |
| **Description** | Initialize the switch pin to input pull up | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_HAL\_SWITCH\_002 V1.0 | | |
| **Return Type** | u8 | **Input arguments** | void |
| **Name** | SWITCH\_u8GetSwitchStatus | | |
| **Description** | Returns 1 if the switch is pressed or 0 otherwise | | |

### **MCAL Components APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_MCAL\_DIO\_001 V1.0 | | |
| **Return Type** | Error\_Status | **Input arguments** | void |
| **Name** | DIO\_Init | | |
| **Description** | Initialize Port A Pins 0:7 to output Initialize Port B Pins 0:7 to Input Initialize Port D Pins 0:3 to output Initialize Port D Pin 4 to input if the function done its purpose return OK if not return NOK | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_MCAL\_DIO\_002 V1.0 | | |
| **Return Type** | Error\_Status | **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 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_MCAL\_DIO\_003 V1.0 | | |
| **Return Type** | Error\_Status | **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 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_MCAL\_DIO\_004 V1.0 | | |
| **Return Type** | Error\_Status | **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\_API\_MCAL\_TIMER\_001 V1.0 | | |
| **Return Type** | Error\_Status | **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 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_MCAL\_TIMER\_002 V1.0 | | |
| **Return Type** | Error\_Status | **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 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_API\_MCAL\_TIMER\_003 V1.0 | | |
| **Return Type** | Error\_Status | **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 | | |