|  |
| --- |
|  |
| SOVY |
| Digital Calculator PO1\_DGC\_GDD |

|  |
| --- |
|  |

Table of Contents

[1. Document Status 3](#_Toc68619986)

[2. Document History 4](#_Toc68619987)

[3. Reference Documents 5](#_Toc68619988)

[4. Project Description 5](#_Toc68619989)

[5. Software Context Diagram 6](#_Toc68619990)

[6. Software features 7](#_Toc68619991)

[7. Static Architecture 7](#_Toc68619992)

[8. MCAL Component APIs 8](#_Toc68619993)

[9. HAL Component APIs 14](#_Toc68619997)

[10. App Component APIs 18](#_Toc68620001)

# Document Status

|  |  |
| --- | --- |
| Name | SOVY |
| Version | 1.4 |
| Status | Released |
| Author | Hossam Magdy |
| Date | 15/4/2021 |

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Change |
| 1.0 | Abdullah Ayman | 3/4/2021 | Initial Creation. |
| 1.1 | Nihal khaled | 4/4/2021 | Software feature and Static diagrams are added |
| 1.2 | Habiba Mahmoud | 5/4/2021 | Added APP Component APIs   * Req\_PO1\_DGC\_GDD\_0011-V1.0 * Req\_PO1\_DGC\_GDD\_0012-V1.0 * Req\_PO1\_DGC\_GDD\_0013-V1.0 * Req\_PO1\_DGC\_GDD\_0014-V1.0 |
| 1.3 | Anrdew Ezzat | 6/4/2021 | Added APP Component APIs   * Req\_PO1\_DGC\_GDD\_0015-V1.0 * Req\_PO1\_DGC\_GDD\_0016-V1.0 * Req\_PO1\_DGC\_GDD\_0017-V1.0 * Req\_PO1\_DGC\_GDD\_0018-V1.0 * Req\_PO1\_DGC\_GDD\_0019-V1.0 * Req\_PO1\_DGC\_GDD\_0020-V1.0 * Req\_PO1\_DGC\_GDD\_0021-V1.0 |
| 1.4 | Hossam Magdy | 15/4/2021 | Edited Requirements:   * Req\_PO1\_DGC\_GDD\_PORT\_001-V1.0 * Req\_PO1\_DGC\_GDD\_PORT\_002-V1.0 * Req\_PO1\_DGC\_GDD\_PORT\_003-V1.0 * Req\_PO1\_DGC\_GDD\_DIO\_004-V1.0 * Req\_PO1\_DGC\_GDD\_DIO\_005-V1.0 * Req\_PO1\_DGC\_GDD\_TIMER\_006-V1.0 * Req\_PO1\_DGC\_GDD\_TIMER\_007-V1.0 * Req\_PO1\_DGC\_GDD\_TIMER\_008-V1.0 * Req\_PO1\_DGC\_GDD\_TIMER\_009-V1.0 * Req\_PO1\_DGC\_GDD\_TIMER\_0010-V1.0 * Req\_PO1\_DGC\_GDD\_KEYPAD\_0015-V1.0 * Req\_PO1\_DGC\_GDD\_BUZZER\_0016-V1.0 * Req\_PO1\_DGC\_GDD\_LCD\_0017-V1.0 * Req\_PO1\_DGC\_GDD\_LCD \_0018-V1.0 * Req\_PO1\_DGC\_GDD\_LCD \_0019-V1.0 * Req\_PO1\_DGC\_GDD\_LCD \_0020-V1.0 * Req\_PO1\_DGC\_GDD\_LCD \_0021-V1.0 * Req\_PO1\_DGC\_GDD\_MATH\_CALC\_0011-V1.0 * Req\_PO1\_DGC\_GDD\_OPERATION\_MANAGER\_0012 * Req\_PO1\_DGC\_GDD\_OPERATION\_MANAGER\_0013 * Req\_PO1\_DGC\_GDD\_KEYPAD\_MANAGER\_0014 |

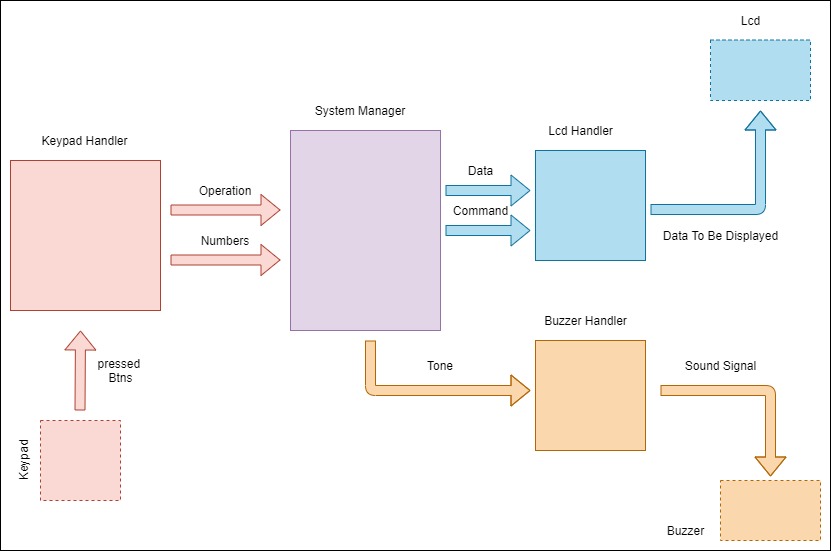
# Reference Documents

|  |  |  |  |
| --- | --- | --- | --- |
| Ref. No. | Doc. Name | Version | Status |
| 1 | PO1\_DGC\_SRS\_DigitalCalculator | 1.4 | Released |

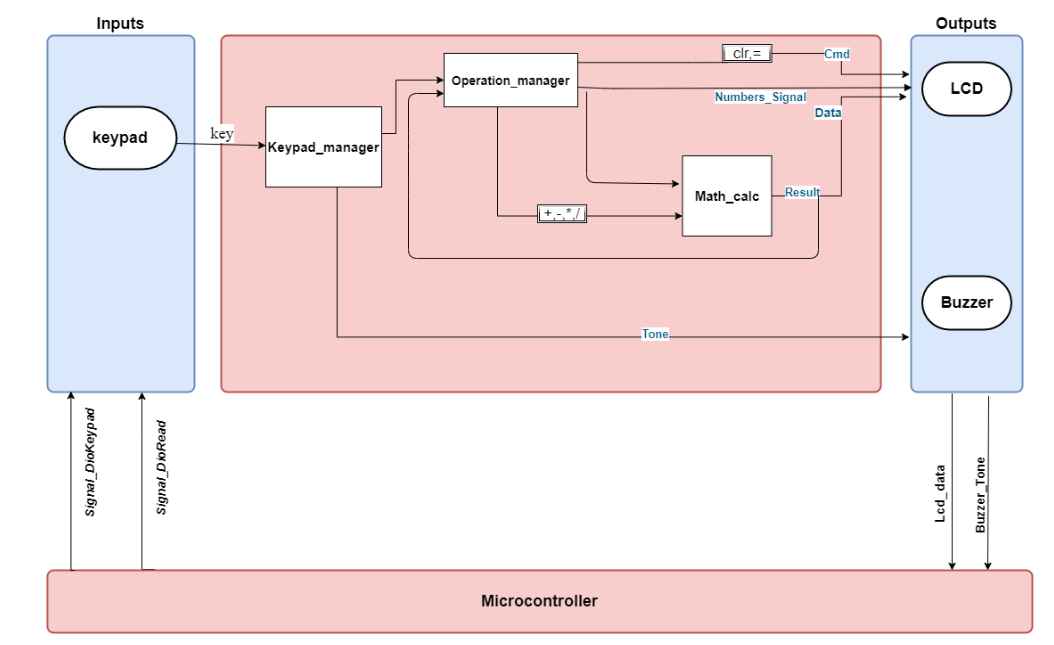
# Project Description

“Sovy” is a simple digital calculator that performs basic mathematical operations and displays the operation on an LCD-screen, with an ON/OFF button and keypad for user input.

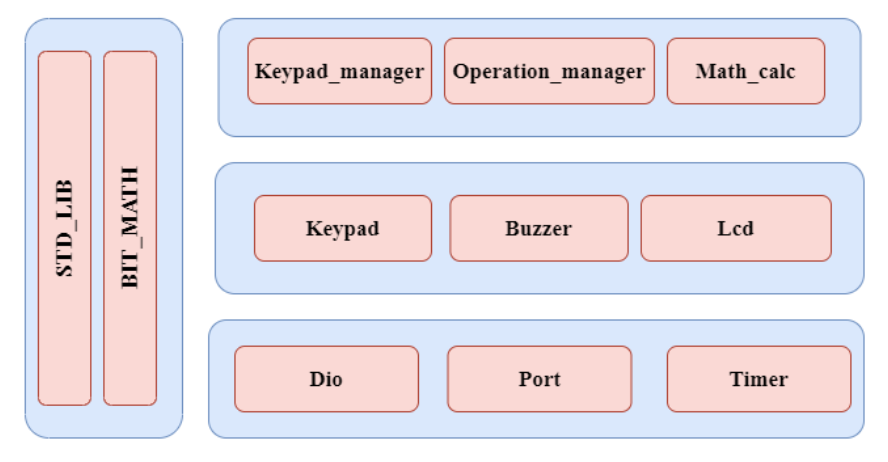
# Software Context Diagram



# 6. Software features



# 7.Static Architecture



# 8. MCAL Component APIs

## Port:

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_PORT\_001-V1.1 | | |
| **Component** | Port | | |
| **Return Type** | Void | **Input arguments** | void |
| **Name** | **Port\_vidInit** | | |
| **Description** | Initialize all pins and ports to their operation mode and initial values | | |
| **Covers** | -None | | |
| **Range** | - | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_PORT\_002-V1.1 | | | |
| **Component** | Port | | | |
| **Return Type** | Void | **Input arguments** | ***u8*** | Copy\_u8SwPinNo |
| ***u8*** | Copy\_u8PinMode |
| **Name** | **Port\_vidSetPinMode** | | | |
| **Description** | This function is used to change the Software pin Mode | | | |
| **Covers** | -None | | | |
| **Range** | **-** Copy\_u8SwPinNo**: {PORT\_u8PIN\_0 -> PORT\_u8PIN\_31}**  - Copy\_u8PinMode**: {PORT\_u8OUTPUT\_LOW, PORT\_u8OUTPUT\_HIGH, PORT\_u8INPUT\_PULL\_UP ,PORT\_u8INPUT\_HIGH\_IMP}** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_PORT\_003-V1.1 | | | |
| **Component** | Port | | | |
| **Return Type** | Void | **Input arguments** | ***u8*** | Copy\_u8SwPinNo |
| ***u8*** | Copy\_u8SwPinDir |
| **Name** | **Port\_vidSetPinDir** | | | |
| **Description** | This function is used to change the Software pin direction | | | |
| **Covers** | -None | | | |
| **Range** | **-** Copy\_u8SwPinNo**: {PORT\_u8PIN\_0 -> PORT\_u8PIN\_31}**  - Copy\_ u8SwPinDir**: {Port\_u8INPUT,Port\_u8OUTPUT}** | | | |

## Dio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_DIO\_004-V1.1 | | | |
| **Component** | Dio | | | |
| **Return Type** | u8 | **Input arguments** | ***u8*** | Copy\_u8SwPinNo |
| ***u8*** | Copy\_u8SwPinVal |
| **Name** | **Dio\_u8SetPinVal** | | | |
| **Description** | This function is used to change the Software pin Value | | | |
| **Covers** | -None | | | |
| **Range** | **-** Copy\_u8SwPinNo**: {PORT\_u8PIN\_0 -> PORT\_u8PIN\_31}**  - Copy\_ u8SwPinVal**: {DIO\_u8HIGH, DIO\_u8LOW}** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_DIO\_005-V1.1 | | | |
| **Component** | Dio | | | |
| **Return Type** | u8 | **Input arguments** | ***u8*** | Copy\_u8SwPinNo |
| **Name** | **Dio\_u8GetPinVal** | | | |
| **Description** | This function is used to get the Software pin Value | | | |
| **Covers** | -None | | | |
| **Range** | **-** Copy\_u8SwPinNo**: {PORT\_u8PIN\_0 -> PORT\_u8PIN\_31}**  **-** returnrange **: {DIO\_u8HIGH, DIO\_u8LOW}** | | | |

## Timer

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_TIMER\_006-V1.1 | | |
| **Component** | Timer | | |
| **Return Type** | Void | **Input arguments** | Void |
| **Name** | **Gpt\_vidInit** | | |
| **Description** | This function is used to initialize the Timer | | |
| **Covers** | -None | | |
| **Range** | **-** None | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_TIMER\_007-V1.1 | | |
| **Component** | Timer | | |
| **Return Type** | Void | **Input arguments** | Void |
| **Name** | **Gpt\_vidStartTimer** | | |
| **Description** | This function is used to start the Timer | | |
| **Covers** | -None | | |
| **Range** | **-** None | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_TIMER\_008-V1.1 | | |
| **Component** | Timer | | |
| **Return Type** | Void | **Input arguments** | Void |
| **Name** | **Gpt\_vidStopTimer** | | |
| **Description** | This function is used to stop the Timer | | |
| **Covers** | -None | | |
| **Range** | **-** None | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_TIMER\_009-V1.1 | | |
| **Component** | Timer | | |
| **Return Type** | Void | **Input arguments** | Void (\*pfun)(void) |
| **Name** | **Gpt\_vidSetCBF** | | |
| **Description** | This function is used to set the call back function of the Timer | | |
| **Covers** | -None | | |
| **Range** | **-** None | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_TIMER\_0010-V1.1 | | | |
| **Component** | Timer | | | |
| **Return Type** | Void | **Input arguments** | ***u8*** | Copy\_u8LoadVal |
| **Name** | **Gpt\_vidLoadTimer** | | | |
| **Description** | This function is used to load the timer with a value to start counting from it | | | |
| **Covers** | -None | | | |
| **Range** | **-** Copy\_u8LoadVal : {0 -> 255 } | | | |

## 

## 9. HAL Component APIs

## Switch Component:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_KEYYPAD\_0015-V1.1 | | **Covers** |  | Req\_1ST123\_SRS\_overall\_001-V1.1  Req\_1ST123\_SRS\_overall\_002-V1.1  Req\_1ST123\_SRS\_overall\_003-V1.2  Req\_1ST123\_SRS\_overall\_004-V1.2  Req\_1ST123\_SRS\_overall\_005-V1.2  Req\_1ST123\_SRS\_overall\_006-V1.2  Req\_1ST123\_SRS\_overall\_007-V1.1  Req\_1ST123\_SRS\_overall\_008-V1.1  Req\_1ST123\_SRS\_overall\_009-V1.2 |
| **Description:** | Get Switch state according to its config file that it configures its Port and pin to get its status | | | | |
| **API** | ReturnStatus\_e Switch\_GetState (u8 Copy\_u8SwId, pu8 Copy\_ptrState) | | | | |
| **Parameters** | **Type** | **Name** | | | **Description** |
| u8  pu8 | Copy\_u8SwId  Copy\_ptrState | | | Switch ID to get state of a certain switch  Pointer to state of the switch according to Switch\_config file  Pressed will represent 0 or 1  If RELEASE\_STATE defined as 1  So pressed will represent 0  If RELEASE\_STATE defined as 0  So pressed will represent 1 |
| **Return** | **Type** | **Name** | | | **Description** |
| ReturnStatus\_e | \_\_\_\_\_ | | | STD\_ERORR  E\_OK=0 -> If the SwNum sent is in the scope  E\_NOK=1 -> If SwNum sent is out of the scope |

# 2. Buzzer Component:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_BUZZER\_0016-V1.1 | | **Covers** |  | Req\_1ST123\_SRS\_overall\_026-V1.1  Req\_1ST123\_SRS\_overall\_027-V1.0 |
| **Description:** | It sets buzzer state according to its config file that it congiures its port and pin make sound when button is pressed. | | | | |
| **API** | ReturnStatus\_e Buzzer\_SetState(u8 State) | | | | |
| **Parameters** | **Type** | **Name** | | | **Description** |
| u8 | State | | | Buzzer state to set its pin to this state  High -> 1  Low -> 0 |
| **Return** | **Type** | **Name** | | | **Description** |
| ReturnStatus\_e | \_\_\_\_\_ | | | STD\_ERORR  E\_OK = 0 -> If State Sent is 0 or 1  E\_NOK = 1 -> If State send is not 0 nor 1 |

# 3. LCD Component:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_LCD\_0017-V1.1 | | **Covers** |  | Req\_1ST123\_SRS\_overall\_013-V1.1 |
| **Description:** | It initializing the Lcd by init sequence and the configuration of pins. | | | | |
| **API** | ReturnStatus\_e Lcd\_Init(void) | | | | |
| **Parameters** | **Type** | **Name** | | | **Description** |
| Void | \_\_\_\_\_ | | | \_\_\_\_\_ |
| **Return** | **Type** | **Name** | | | **Description** |
| ReturnStatus\_e | \_\_\_\_\_ | | | STD\_ERORR  E\_OK -> If the function cleared the LCD successfully  E\_NOK -> If any problem occurred while clearing the LCD |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_LCD\_0018-V1.1 | | **Covers** |  | Req\_1ST123\_SRS\_overall\_025-V1.1 |
| **Description:** | It clears the screen. | | | | |
| **API** | ReturnStatus\_e Lcd\_Clear(void) | | | | |
| **Parameters** | **Type** | **Name** | | | **Description** |
| Void | \_\_\_\_\_ | | | \_\_\_\_\_ |
| **Return** | **Type** | **Name** | | | **Description** |
| ReturnStatus\_e | \_\_\_\_\_ | | | STD\_ERORR  E\_OK = 0 -> If the function cleared the LCD successfully  E\_NOK = 1 -> If any problem occurred while clearing the LCD |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_LCD\_0019-V1.1 | | **Covers** |  | Req\_1ST123\_SRS\_overall\_022-V1.1  Req\_1ST123\_SRS\_overall\_021-V1.1  Req\_1ST123\_SRS\_overall\_014-V1.1  Req\_1ST123\_SRS\_overall\_015-V1.1  Req\_1ST123\_SRS\_overall\_016-V1.1  Req\_1ST123\_SRS\_overall\_017-V1.1  Req\_1ST123\_SRS\_overall\_018-V1.1  Req\_1ST123\_SRS\_overall\_019-V1.1  Req\_1ST123\_SRS\_overall\_012-V1.2  Req\_1ST123\_SRS\_overall\_011-V1.2 |
| **Description:** | It takes Pointer to char of the string that it will be written as whole onto the LCD. | | | | |
| **API** | ReturnStatus\_e Lcd\_WriteString(pu8 Copy\_ptrString); | | | | |
| **Parameters** | **Type** | **Name** | | | **Description** |
| pu8 | Copy\_ptrString | | | Pointer to char of the string to be written onto LCD |
| **Return** | **Type** | **Name** | | | **Description** |
| ReturnStatus\_e | \_\_\_\_\_ | | | STD\_ERORR  E\_OK = 0 -> If sent character pattern valid in the CGROM  E\_NOK = 1 -> If a pattern sent outside the scope of the CGROM |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_LCD\_0020-V1.1 | | **Covers** |  | Req\_1ST123\_SRS\_overall\_019-V1.1 |
| **Description:** | It sets new position for the LCD to start write from.  XPos -> which column  YPos -> which Row  To select which cell. | | | | |
| **API** | ReturnStatus\_e Lcd\_GotoXY(u8 Copy\_u8XPos, u8 Copy\_u8YPos) | | | | |
| **Parameters** | **Type** | **Name** | | | **Description** |
| u8  u8 | Copy\_u8XPos  Copy\_u8YPos | | | XPos -> which column  YPos -> which row |
| **Return** | **Type** | **Name** | | | **Description** |
| ReturnStatus\_e | \_\_\_\_\_ | | | STD\_ERORR  E\_OK = 0 -> If XPos is sent between 0-15 and YPos is sent between 0-1  E\_NOK = 1 -> If any other range is sent either in XPos or YPos |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req\_ID | Req\_PO1\_DGC\_GDD\_LCD\_0021-V1.1 | | **Covers** |  | Req\_1ST123\_SRS\_overall\_020-V1.1 |
| Description: | It writes number onto the LCD. | | | | |
| API | ReturnStatus\_e Lcd\_WriteNum (u16 Copy\_u16Num); | | | | |
| Parameters | **Type** | **Name** | | | **Description** |
| U16 | Copy\_u16Num | | | Number to be written onto the LCD |
| Return | **Type** | **Name** | | | **Description** |
| ReturnStatus\_e | \_\_\_\_\_ | | | STD\_ERORR  E\_OK = 0 -> If Numbers between 0-9 are sent  E\_NOK = 1 -> If anything except 0-9 is sent to the function |

# App Component APIs

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_MATH\_CALC\_0011-V1.1 | | |
| **Component** | Math\_Calc | | |
| **Return Type** | ReturnStatus\_e | **Input arguments** | f32\* Copy\_f32Result |
| **Name** | DGC\_CalcResult | | |
| **Description** | Calculate the operation and output the result and handle the corner cases. | | |
| **Covers** | -Req\_1ST123\_SRS\_overall\_003-V1.2  -Req\_1ST123\_SRS\_overall\_004-V1.2  -Req\_1ST123\_SRS\_overall\_005-V1.2  -Req\_1ST123\_SRS\_overall\_006-V1.2  -Req\_1ST123\_SRS\_overall\_008-V1.1  -Req\_1ST123\_SRS\_overall\_009-V1.2  -Req\_1ST123\_SRS\_overall\_010-V1.2  -Req\_1ST123\_SRS\_overall\_011-V1.2  -Req\_1ST123\_SRS\_overall\_012-V1.2 | | |
| **Range** | Copy\_f32Result: (Float range)  ReturnStatus\_e: (E\_OK = 0 , E\_NOK = 1) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_OPERATION\_MANAGER\_0012-V1.1 | | |
| **Component** | Operation\_Manager | | |
| **Return Type** | ReturnStatus\_e | **Input arguments** | u8 Copy\_u8Char |
| **Name** | DGC\_DisplayChar | | |
| **Description** | Display a character on the LCD screen | | |
| **Covers** | -Req\_1ST123\_SRS\_overall\_014-V1.1  -Req\_1ST123\_SRS\_overall\_015-V1.1  -Req\_1ST123\_SRS\_overall\_016-V1.1  -Req\_1ST123\_SRS\_overall\_017-V1.1  -Req\_1ST123\_SRS\_overall\_018-V1.1  -Req\_1ST123\_SRS\_overall\_019-V1.1  -Req\_1ST123\_SRS\_overall\_020-V1.1 | | |
| **Range** | Copy\_u8Char: ([0:9],+,-,\*,/,=)  ReturnStatus\_e: (E\_OK = 0, E\_NOK = 1) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_OPERATION\_MANAGER\_0013-V1.1 | | |
| **Component** | Operation\_Manager | | |
| **Return Type** | ReturnStatus\_e | **Input arguments** | u8 Copy\_kpau8Str |
| **Name** | DGC\_DisplayError | | |
| **Description** | Display error message when needed | | |
| **Covers** | -Req\_1ST123\_SRS\_overall\_021-V1.1  -Req\_1ST123\_SRS\_overall\_022-V1.1 | | |
| **Range** | Copy\_kpau8Str: (“Syntax Error”,” Math Error”)  ReturnStatus\_e: (E\_OK = 0, E\_NOK = 1) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_GDD\_KEYPAD\_MANAGER\_0014-V1.1 | | |
| **Component** | KeyPad\_Manager | | |
| **Return Type** | ReturnStatus\_e | **Input arguments** | void |
| **Name** | DGC\_PlayTone | | |
| **Description** | Plays a tone when a key is pressed on the keypad | | |
| **Covers** | -Req\_1ST123\_SRS\_overall\_026-V1.0 | | |
| **Range** | ReturnStatus\_e: (E\_OK = 0, E\_NOK = 1) | | |