**PO1\_DGC\_Digital Calculator**

**SRS Document**

**Version 1.10**

**Proposed**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Document Change History** | | |  |
| **Version** | **Author** | **Date** | **Change** | **Status** |
| 1.0 | Mina Helmi | 03/2/2020 | * Initial creation | Draft |
| 1.1 | Hazem Mekawy | 05/2/2020 | * Review | Proposed |
| 1.2 | Mina Helmi | 07/2/2020 | * Removed requirement **Req\_PO1\_DGC\_SRS\_010\_V01**, and appended its useful parts to requirement **Req\_PO1\_DGC\_SRS\_007\_V01** * Added a new requirement **Req\_PO1\_DGC\_SRS\_012\_V01** to indicate that the user input shall be received from the keypad. * Added the desired previous state of the switch in both requirements **Req\_PO1\_DGC\_SRS\_010\_V01** and **Req\_PO1\_DGC\_SRS\_011\_V01** * Changed requirement **Req\_PO1\_DGC\_SRS\_006\_V01** to be more explicit and clear regarding the operator and the operand * Added a new requirement **Req\_PO1\_DGC\_SRS\_013\_V01** to indicate that the software design should follow the described flow chart * Renamed State Machine to Flow Chart | Proposed |
| 1.3 | Mina Helmi | 08/2/2020 | * Corrected flow chart * Corrected required switch state in requirement **Req\_PO1\_DGC\_SRS\_011\_V01** * Corrected input of requirement **Req\_PO1\_DGC\_SRS\_012\_V01** | Proposed |
| 1.4 | Mina Helmi | 14/2/2020 | * Added 2 new requirements **Req\_PO1\_DGC\_SRS\_014\_V01** and **Req\_PO1\_DGC\_SRS\_015\_V01** to indicate the range of the horizontal cursor position and what should happen if they were exceeded * Added 2 new requirements **Req\_PO1\_DGC\_SRS\_016\_V01** and **Req\_PO1\_DGC\_SRS\_017\_V01** to indicate the range of the vertical cursor position and what should happen if they were exceeded * Added project description from the CYRS document (V1.2 Released) | Proposed |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.5 | Mina Helmi | 18/2/2020 | * Updated the requirements **Req\_PO1\_DGC\_SRS\_004\_V01** and **Req\_PO1\_DGC\_SRS\_005\_V01** to reflect the new requirements of the buzzer tones | Proposed |
| 1.6 | Mina Helmi | 20/02/2020 | * Updated requirements **Req\_PO1\_DGC\_SRS\_014\_V01** and **Req\_PO1\_DGC\_SRS\_016\_V01** to be more descriptive. | Proposed |
| 1.7 | Mina Helmi | 23/02/2020 | * Added Table of signals * Added Context diagram * Categorized requirements * Fixed some typos | Proposed |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.8 | Mina Helmi | 27/02/2020 | * Updated **Req\_PO1\_DGC\_SRS\_001\_V01** to be more atomic and specific and moved it to category **User input acceptance from the keypad** * Added new **Req\_PO1\_DGC\_SRS\_018\_V01**, previously part of **Req\_PO1\_DGC\_SRS\_001\_V01** * Updated **Req\_PO1\_DGC\_SRS\_009\_V01** to indicate that the result shall be displayed from the beginning of the second line * Updated **Req\_PO1\_DGC\_SRS\_010\_V01** to resolve a conflict with **Req\_PO1\_DGC\_SRS\_011\_V01** if the pressed the ON/OFF switch * Updated **Req\_PO1\_DGC\_SRS\_012\_V01** to reflect the keypad layout from the HSI document * Added **Req\_PO1\_DGC\_SRS\_019\_V01** and **Req\_PO1\_DGC\_SRS\_020\_V01** to specify the max limit of each operand * Added **Req\_PO1\_DGC\_SRS\_021\_V01** and **Req\_PO1\_DGC\_SRS\_022\_V01** to specify what should happen if each operand reached the maximum limit * Added **Req\_PO1\_DGC\_SRS\_023\_V01** and **Req\_PO1\_DGC\_SRS\_024\_V01** to specify what should happen if the user tried to enter more than 1 sign * Added **Req\_PO1\_DGC\_SRS\_025\_V01** to indicate what happen when pressing the clear switch | Proposed |
| 1.9 | Mina Helmi | 03/03/2020 | * Updated **Req\_PO1\_DGC\_SRS\_001\_V01** to be more atomic/specific, and only concerned with operands 1 and 2, and the operator | Proposed |
| 1.10 | Mina Helmi | 14/03/2020 | * Added 8 new requirements **Req\_PO1\_DGC\_HSI\_026\_V01**, **Req\_PO1\_DGC\_ HSI\_027\_V01**, **Req\_PO1\_DGC\_ HSI\_028\_V01**, **Req\_PO1\_DGC\_ HSI\_029\_V01**, **Req\_PO1\_DGC\_ HSI\_030\_V01**, **Req\_PO1\_DGC\_ HSI\_031\_V01**, **Req\_PO1\_DGC\_ HSI\_032\_V01**, **Req\_PO1\_DGC\_ HSI\_033\_V01** to cover the HSI requirement **PO1\_DGC\_HSI\_006\_V01** which is concerned with hardware connections * Updated reference documents: HSI V1.4, CYRS V1.3 * Fixed some typos | Proposed |

|  |  |  |
| --- | --- | --- |
| **Reference Document** | | |
| **Reference document** | **Version** | **Status** |
| CYRS | 1.3 | Released |
| HSI | 1.4 | Proposed |

**Table of Contents**

[Project Description 10](#_Toc35086303)

[Context diagram 11](#_Toc35086304)

[Requirements 12](#_Toc35086305)

[Table of signals 24](#_Toc35086306)

[Flow Chart 25](#_Toc35086307)

**Index of Figures:**

[Figure 1: Context diagram 11](file:///C:\Users\MH\Desktop\SRS.docx#_Toc35086313)

[Figure 2: Flow chart 25](file:///C:\Users\MH\Desktop\SRS.docx#_Toc35086314)

# Project Description

This project is a digital calculator that takes input from user and displays the input and the result on a screen.

The hardware used in the calculator is Keypad which takes input from user, LCD to display the result, buzzer to generate tunes on each key press and micro controller that performs all operations in the system.

# Context diagram

Software Main Handler

Figure : Context diagram

LCD  
Manager

Display On/Off

ON/OFF  
Handler

Buzzer  
Manager

Buzzer tone 2  
Tpulse = 4ms, duty = 50%

Buzzer tone 1  
Tpulse = 2ms, duty = 50%

Error message

Display Handler

Cur\_x

Cur\_y

Opr1

Result

Operator

Opr2

Key press

Switch press

Keypad  
Manager

ON/OFF  
Switch Manager

# Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Result calculation, and error checking and reporting** | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_018\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | The software shall store the result of the mathematical operation in <**Result>**.  An OK error state will be assigned to <**ErrState**>. | | |
| **Inputs** | * Opr1 * Opr2 * Operator | **Outputs** | * Result = <Opr1> <Operator> <Opr2> * ErrState = OK |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_002\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall report a bad error state <**NOK**> if an operand, either <**Opr1**> or <**Opr2**> already contained a decimal point and the user tried to type another decimal point. | | |
| **Inputs** | * Opr1 * Opr2 | **Outputs** | ErrState = NOK |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_003\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall report a bad error state <**NOK**> if any operand either <**Opr1**> or <**Opr2**> is preceded by more than one sign. | | |
| **Inputs** | * Opr1 * Opr2 | **Outputs** | ErrState = NOK |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_006\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_003\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall check the operator <**Operator**> if it’s equal to the ASCII symbol '**/**' and the second operand <**Opr2**> if it’s equal to 0, and report a bad error state <**NOK**> if a division by zero occurred. | | |
| **Inputs** | * Opr2 * Operator | **Outputs** | ErrState = NOK |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_007\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_003\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall display an error message if a bad error state <**NOK**> was reported at any point during the software execution for only 2 seconds on the first line of the LCD exclusively, then the retained user input shall be displayed again and the cursor is set back to its last saved location. | | |
| **Inputs** | ErrState = NOK | **Outputs** | Error message: "**ERR: Wrong Input**" |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **LCD state and display, and cursor positioning** | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_008\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_004\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall display the user operands <**Opr1**> and <**Opr2**> and the operator <**Operator**> on the first line of the LCD, then update the cursor position <**Cur\_x**> and <**Cur\_y**>. | | |
| **Inputs** | * Opr1 * Opr2 * Operator | **Outputs** | * Cur\_x * Cur\_y * Output on the first line of the LCD |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_014\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_004\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 02 / 2020 |
| **Description** | The software shall make sure that the horizontal cursor position of the LCD stored in <**Cur\_x**> never goes outside the range [0, 15] inclusively, where 0 means the first column of the LCD and 15 means the last column on the LCD regardless of the row. This indicator changes linearly by 1 step. | | |
| **Inputs** | Cur\_x | **Outputs** | Any value between [0, 15] inclusively |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_016\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_004\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 02 / 2020 |
| **Description** | The software shall make sure that the vertical cursor position of the LCD stored in <**Cur\_y**> never goes outside the range [0, 1] inclusively, where 0 means the first row of the LCD and 1 means the second/last row on the LCD regardless of the column. This indicator changes linearly by 1 step. | | |
| **Inputs** | Cur\_y | **Outputs** | Any value between [0, 1] inclusively |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_009\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_004\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall display the result of the mathematical operation <**Result**> from the beginning of the second line of the LCD, and update the cursor positions <**Cur\_x**> and <**Cur\_y**> correspondingly. | | |
| **Inputs** | Result | **Outputs** | * Cur\_x * Cur\_y * Output on the second line of the LCD |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_010\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_005\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall turn on and clear the display, reset the cursor location <**Cur\_x**> and <**Cur\_y**>, and start a new session if the user pressed the ON/OFF switch, and the previous switch state was <**OFF**> | | |
| **Inputs** | ON/OFF switch = OFF | **Outputs** | * LCD on * Clear LCD * Reset Cur\_x * Reset Cur\_y |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_011\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_005\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall turn off the display if the user pressed the ON/OFF switch, and the previous switch state was <**ON**> | | |
| **Inputs** | ON/OFF switch = ON | **Outputs** | Turn off LCD |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_025\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | The software shall clear the display, reset the cursor locations <**Cur\_x**> and <**Cur\_y**>, and start a new session if the user pressed the clear switch | | |
| **Inputs** | Clear switch is pressed | **Outputs** | * Clear LCD * Reset Cur\_x * Reset Cur\_y |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Buzzer tones** | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_004\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_002\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | The software shall send a signal to the buzzer for **1 second** on each press from the user on the keypad.  The period of the pulse **Tpulse = 2ms**, with a duty cycle = **50%**. | | |
| **Inputs** | Keypad press | **Outputs** | Buzzer tone 1 |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_005\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_002\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall send a signal to the buzzer for **1 second** when a bad error state is reported <**NOK**> at any point during the software execution.  The period of the pulse **Tpulse = 4ms**, with a duty cycle = **50%**. | | |
| **Inputs** | ErrState = NOK | **Outputs** | Buzzer tone 2 |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User input acceptance from the keypad** | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_012\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 07 / 02 / 2020 |
| **Description** | Software shall accept the user input from the keypad and store it in either one of the operands <**Opr1**> or <**Opr2**> or in the operator <**Operator**>. The keypad has the following keys layout:   * First Line [ 1, 2, 3, A ] * Second Line [ 4, 5, 6, B ] * Third Line [ 7, 8, 9, C ] * Fourth Line [ \*, 0, #, D ] | | |
| **Inputs** | Keypad press | **Outputs** | * Opr1 * Opr2 * Operator |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_001\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 03 / 02 / 2020 |
| **Description** | Software shall accept from the user 2 operands <**Opr1**> and <**Opr2**> and an operator <**Operator**> in the following order:   1. Operand 1 is a number, may contain decimal point character and/or a sign. 2. Operator, could be one of the following ASCII characters: '**+**' '**-**' '**/**' '**\***' 3. Operand 2 is a number, may contain decimal point character and/or a sign. | | |
| **Inputs** | Keypad press | **Outputs** | * Operand 1 <**Opr1**> * Operand 2 <**Opr2**> * Operator <**Opr**> * ErrState = OK |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_015\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_004\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 02 / 2020 |
| **Description** | The software shall reject the user input from the keypad if the horizontal cursor position stored in <**Cur\_x**> went outside the range [0, 15] inclusively. | | |
| **Inputs** | Cur\_x | **Outputs** | Reject user input from the keypad |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_017\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_004\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 02 / 2020 |
| **Description** | The software shall reject the user input from the keypad if the vertical cursor position stored in <**Cur\_y**> went outside the range [0, 1] inclusively. | | |
| **Inputs** | Cur\_y | **Outputs** | Reject user input from the keypad |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_019\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | Software shall accept operand 1 from the user <**Opr1**> as a number of maximum 7-characters, including the decimal point and/or the sign. | | |
| **Inputs** | Key Press | **Outputs** | <**Opr1**> |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_020\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | Software shall accept operand 2 from the user <**Opr2**> as a number of maximum 7-characters, including the decimal point and/or the sign. | | |
| **Inputs** | Key Press | **Outputs** | <**Opr2**> |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_021\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | The software shall stop accepting the user input from the keypad if operand 1 <**Opr1**> reached 7 characters. | | |
| **Inputs** | Opr1 | **Outputs** | Reject user input from the keypad |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_022\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | The software shall stop accepting the user input from the keypad if operand 2 <**Opr2**> reached 7 characters. | | |
| **Inputs** | Opr2 | **Outputs** | Reject user input from the keypad |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_023\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | The software shall reject the user input from the keypad if the user tried to enter more than 1 sign for operand 1 <**Opr1**>. | | |
| **Inputs** | Opr1 | **Outputs** | Reject user input from the keypad |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_024\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 27 / 02 / 2020 |
| **Description** | The software shall reject the user input from the keypad if the user tried to enter more than 1 sign for operand 2 <**Opr2**>. | | |
| **Inputs** | Opr2 | **Outputs** | Reject user input from the keypad |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Pins and connections** | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_026\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The data pins of the LCD shall be connected to Port A, pins [0:7] | | |
| **Inputs** | LCD data pins | **Outputs** | Connection to Port A[0:7] |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_027\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The control pins of the LCD shall be connected to Port B, pins [0:2] | | |
| **Inputs** | LCD control pins | **Outputs** | Connection to Port B[0:2] |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_028\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The rows pins of the keypad shall be connected to Port C, pins [0:3] | | |
| **Inputs** | Keypad rows pins | **Outputs** | Connection to Port C[0:3] |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_029\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The columns pins of the keypad shall be connected to Port C, pins [4:7] | | |
| **Inputs** | Keypad columns pins | **Outputs** | Connection to Port C[4:7] |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_030\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The Buzzer shall be connected to Port D, pin 0 | | |
| **Inputs** | Buzzer input pin | **Outputs** | Connection to Port D[0] |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_031\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The Tactile switch shall be connected to Port D, pin 1 | | |
| **Inputs** | Tactile switch pin | **Outputs** | Connection to Port D[1] |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_032\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The following pins shall be defined as an output:  Port A[0:7], Port B[0:2], Port C[0:3], Port D[0] | | |
| **Inputs** | Port A[0:7], Port B[0:2], Port C[0:3], Port D[0] | **Outputs** | Pins directions: output |
|  | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_033\_V01 | **Covers** | Covers PO1\_DGC\_HSI\_006\_V01 |
| **Author** | Mina Helmi | **Date** | 14 / 03 / 2020 |
| **Description** | The following pins shall be defined as an input:  Port C[4:7], Port D[1] | | |
| **Inputs** | Port C[4:7], Port D[1] | **Outputs** | Pins directions: input |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Overall software design** | | | |
| **Req\_ID** | Req\_PO1\_DGC\_SRS\_013\_V01 | **Covers** | Covers PO1\_DGC \_CYRS\_001\_V01 |
| **Author** | Mina Helmi | **Date** | 07 / 02 / 2020 |
| **Description** | Software shall follow the described flow chart | | |
| **Inputs** | Flow Chart | **Outputs** | Overall software design |
|  | | | |

# Table of signals

|  |  |
| --- | --- |
| **Signal name** | **Possible values** |
| ErrState | <**OK**>, <**NOK**> |
| Result | Any value with accuracy up to 2 decimal points |
| Opr1 | Any value with accuracy up to 2 decimal points |
| Opr2 | Any value with accuracy up to 2 decimal points |
| Operator | <**+**>, <**-**>, <**\***>, <**/**> |
| Cur\_x | [0, 15] |
| Cur\_y | [0, 1] |
| ON/OFF switch | <**ON**>, <**OFF**> |
| Buzzer tone 1 | **Tpulse = 2ms**, duty cycle = **50%**. |
| Buzzer tone 2 | **Tpulse = 4ms**, duty cycle = **50%**. |

# Flow Chart

Figure : Flow chart

**Yes**

Yes

No

No

Yes

No

Turn off  
display

Is calculator on?

Result = <Opr1> <Operator> <Opr2>

Trigger  
buzzer

Display error message

Display error message

Trigger  
buzzer

Is operand valid?

Get  
operand 2  
<Opr2>

Get  
operator

Is operand valid?

Turn on  
display

Reset  
cursor

Clear  
display

Get  
operand 1  
<Opr1>