

# **Design Document.**

***Task: #2.***

***By: Ahmed Taha***

## Contents

1. System Description.....	3
2. System Architectural Pattern.....	3
3. System Constrains.....	3
4. Layered Architecture.....	3
5. SW Data Type Tables.....	4
6. SW Layers.....	5
I. Component Layer.....	5
1. Switch Module.....	5
2. Keypad Module.....	8
3. LCD Module.....	11

## 1. System Description.

A simple Character LCD and Keypad Interface, where the Pressed key's Value should be Displayed on the Character LCD.

Each Press will put a key's value in the LCD's cursor position till the LCD is Full, then overwrite on the LCD's data from the First location.

## 2. System Architectural Pattern.

Monolithic Layered Architecture Pattern.

## 3. System Constrains.

The system shall not be blocked under any condition.

## 4. Layered Architecture.

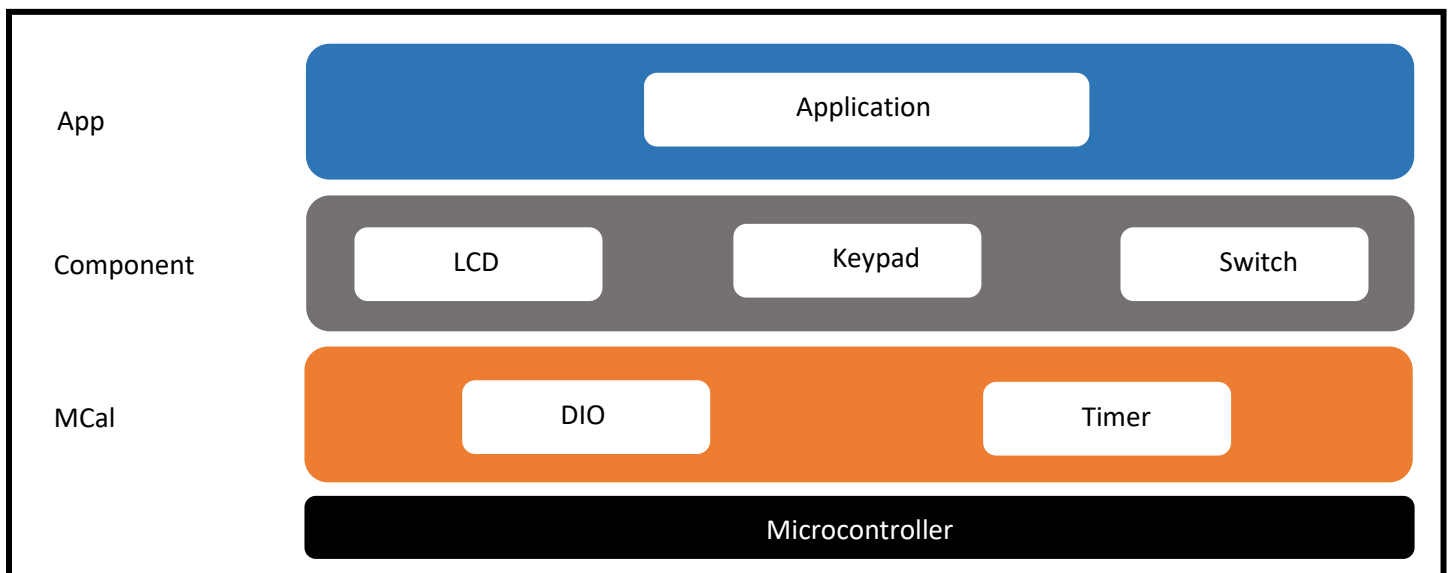


Figure 1: System Layered Architecture.

### ▪ MCal Layer.

- Microcontroller abstraction layer, directly control the HW peripherals inside the Controller.
- Closed Layer.
- Consist of 2 Modules:
  - Digital Input/Output.
  - Timer

▪ **Component Layer.**

- The Component layer controls all the HW devices (components) on Board
- Closed Layer.
- Consist of 3 Modules:
  - LCD
  - Switch
  - Keypad.

▪ **Application Layer.**

- The main app driver.
- Consist only of the main program.

## 5. SW Data Type Tables.

1		2	
Name	uint8_t	Name	int8_t
Type	unsigned char	Type	signed char
Range	0 : 255	Range	-128 : 127
Description	unsigned 8_bit integer	Description	signed 8_bit integer
3		4	
Name	uint16_t	Name	int16_t
Type	unsigned short	Type	signed short
Range	0 : 65535	Range	-32768 : 32767
Description	unsigned 16_bit integer	Description	signed 16_bit integer
5		6	
Name	uint32_t	Name	int32_t
Type	unsigned long int	Type	signed long int
Range	0 : 4294967296	Range	-2147483648 : 2147483647
Description	unsigned 32_bit integer	Description	signed 32_bit integer
7		8	
Name	uint64_t	Name	int64_t
Type	unsigned long long	Type	signed long long
Range	0 : (2^64)	Range	-(2^63) : (2^63) - 1
Description	unsigned 64_bit integer	Description	signed 64_bit integer
9		10	
Name	bool	Name	OpStatus_t
Type	unsigned char	Type	unsigned char
Range	false = 0 true = 1	Range	SUCCESS = 0 FAIL = 1
Description	Boolean type for comparison	Description	the operation status

## 6. SW Layers.

The Detailed Design of each layer.

### I. Component Layer.

Component layer is a closed layer that handles and controls the on-board HW component.

#### 1. Switch Module.

##### 1. Description.

The switch module is responsible for reading the switches status and store it into a buffer to be read with an API.

##### 2. Data Type Tables.

1	
Name	Switch_PressState_t
Type	Enumeration
Range	SWITCH_NULL_VALUE = -1
	KEY_NOT_PRESSED = 0
	KEY_PRESSED = 1
	DEBOUNCING = 2
Description	Describes the switch current status
2	
Name	SwitchStates_t
Type	Enumeration
Range	FIRST_DETECTION_STATE = 0
	DEBOUNCE_STATE = 1
Description	Describes the dispatcher current state
3	
Name	SwitchId_t
Type	unsigned char
Range	0 : 255
Description	describes the number of a switch

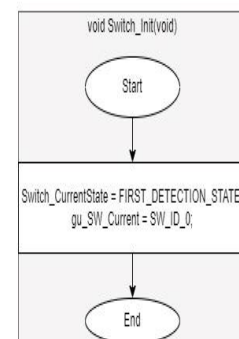
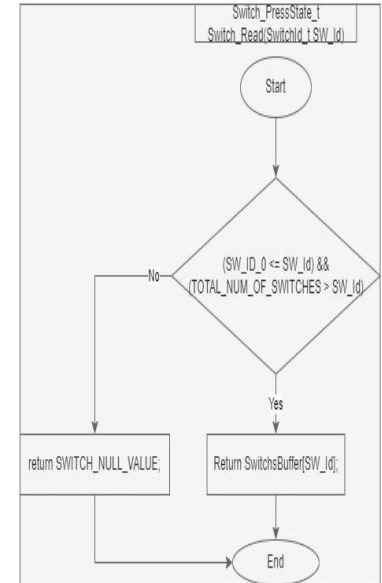
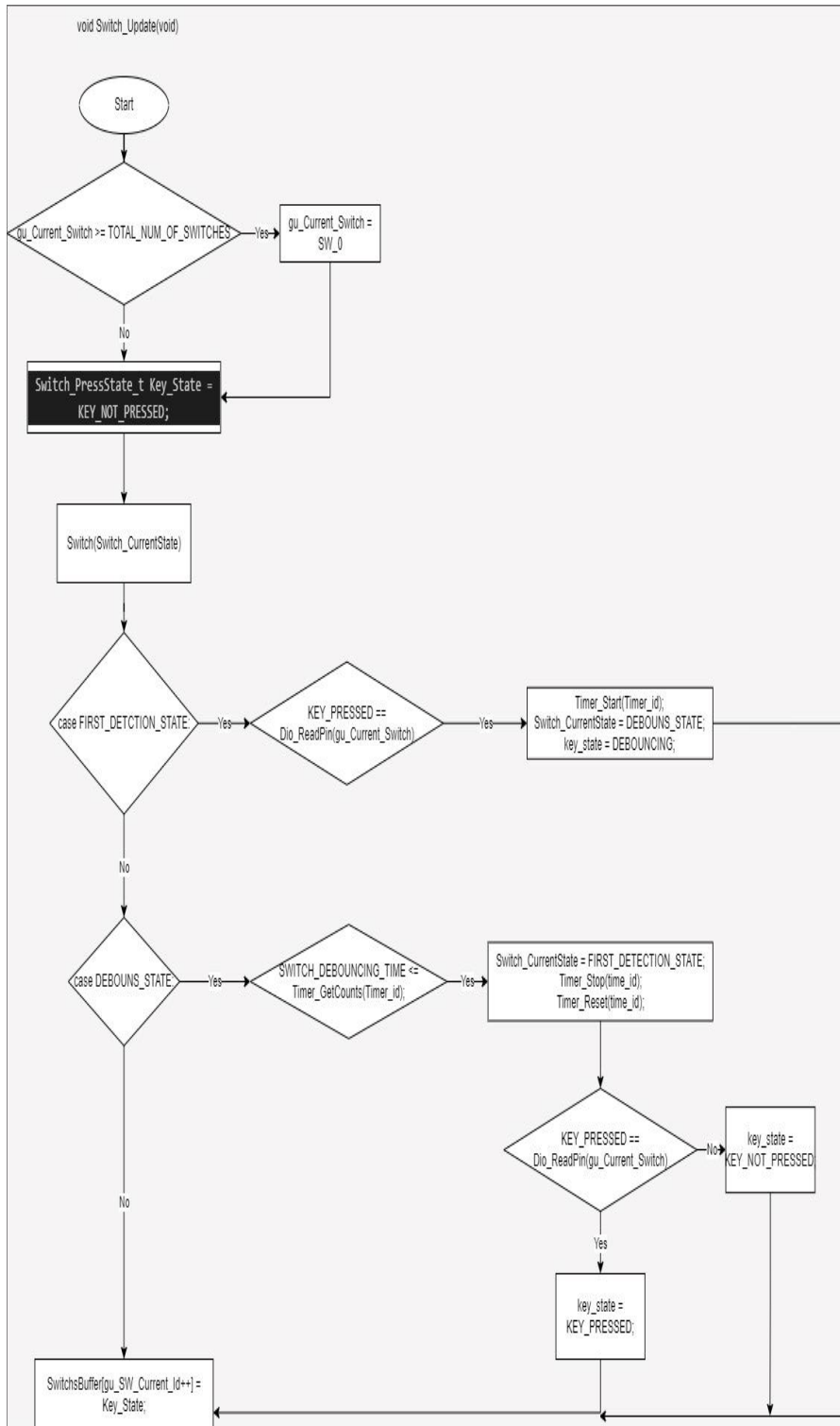
### 3. APIs List.

Function Name	Switch_Init
Arguments	None
Return	None
Synchronous	Yes
Reentrant	Yes
Description	Perform the Init sequence of the module by initializing the DIO Module with the required Switches' pins

Function Name	Switch_Update
Arguments	None
Return	None
Synchronous	Yes
Reentrant	No
Description	the Module dispatcher that responsible for getting the switch press state and store it in it's a propriate location in a buffer

Function Name	Switch_Read		
Arguments	Input	Name: SW_Id	Type: SwitchId_t
		Description: a Switch ID to return its status	
Return	SWITCH_NULL_VALUE		-1
	KEY_NOT_PRESSED		0
	KEY_PRESSED		1
	DEBOUNCING		2
Synchronous	Yes		
Reentrant	No		
Description	A getter API to get the status of a specific switch determined by the API argument.		

#### 4. Module Flow charts.



## 2. Keypad Module.

### 1. Description.

The keypad Module is responsible for detecting the pressed keys of the keypad through the switch module and return the pressed key's data.

### 2. Data Type Tables.

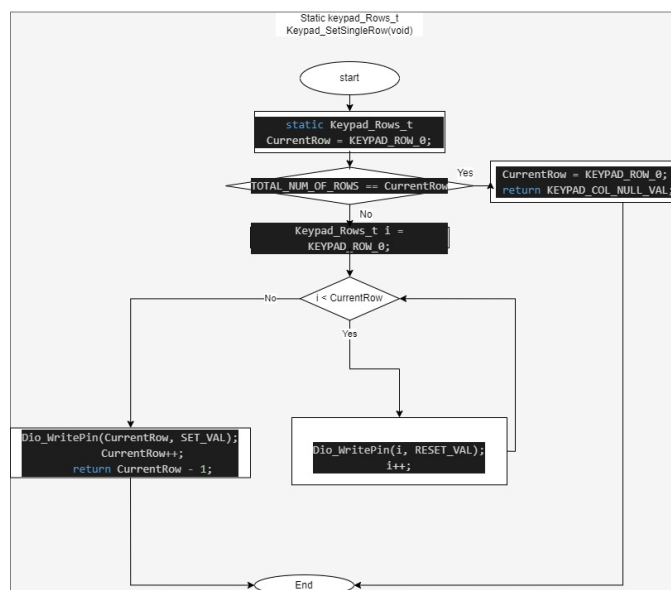
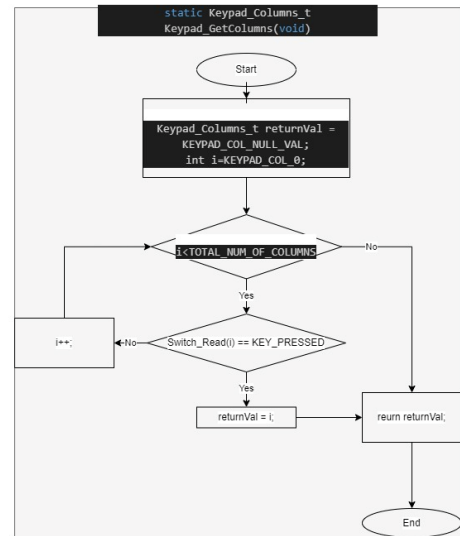
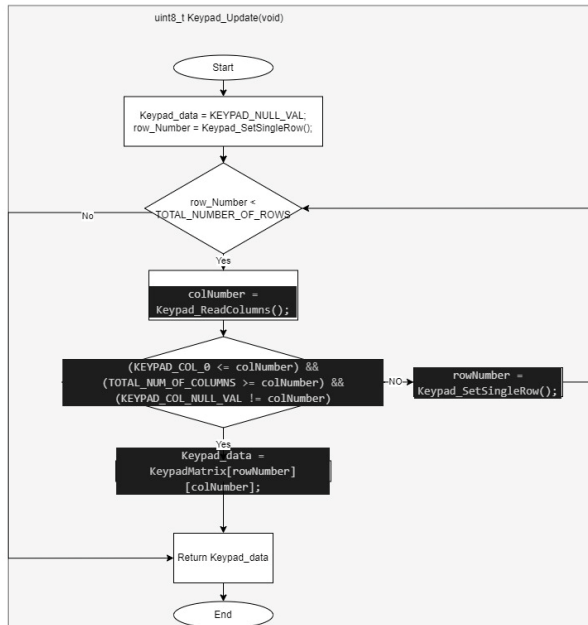
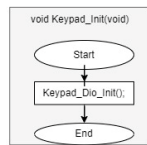
1	
Name	Keypad_dataMatrix_t
Type	uint8_t [][]
Range	0 : 255 For each cell
Description	A 2D array contains the keypad equivalent values
2	
Name	Keypad_Columns_t
Type	Enumeration
Range	From 0 : 255 according to the Columns number
Description	An Enumeration that contains the Rows Locations
3	
Name	Keypad_Rows_t
Type	Enumeration
Range	From 0 : 255 according to the Rows number2
Description	An Enumeration that contains the Rows Locations



### 3. APIs List.

<b>Function Name</b>	<b>Keypad_Update</b>
<b>Arguments</b>	None
<b>Return</b>	The Equivalent pressed Key data from [Keypad_dataMatrix_t]
<b>Synchronous</b>	Yes
<b>Reentrant</b>	No
<b>Description</b>	The Main APIs for the module which detects the pressed switch and returns its equivalent value

#### 4. Module Flow charts.



### 3. LCD Module.

#### 1. Description.

The LCD Module responsible for initializing the LCD and Display the required data on it.

#### 2. Data Type Table.

<b>Name</b>	<b>LCD_DataType_t</b>
<b>Type</b>	Enum
<b>Range</b>	CMD = 0
	DATA = 1
<b>Description</b>	describing the write operation type

<b>Name</b>	<b>LCD_States_t</b>
<b>Type</b>	Enum
<b>Range</b>	LCD_SEND_FIRST_NIBBLE_STATE = 0
	LCD_SEND_SECOND_NIBBLE_STATE = 1
	LCD_TRIGGER_DELAY_STATE = 2
	LCD_INIT_DELAY_STATE = 3
<b>Description</b>	the Module states

#### 3. APIs List.

Function Name	Lcd_Write		
Arguments	Input	Name: Data	Type: uint8_t
		the data to be sent to the LCD [CMD or DATA]	
	Input	Name: dataType	Type: LCD_DataType_t
		the type of the Transmitted data is it CMD or Data	
Return	NONE		
Synchronous	Yes		
Reentrant	No		
Description	A Setter API that puts a data to be sent in a buffer with its type		

<b>Function Name</b>	<b>Lcd_Update</b>
<b>Arguments</b>	None
<b>Return</b>	NONE
<b>Synchronous</b>	Yes
<b>Reentrant</b>	NO
<b>Describtion</b>	The Module Dispatcher that responsible for fetching the data to be sent to the LCD from a Buffer, and send it nibble by nibble.

#### 4. LCD Flow charts.

