**[Global Design Document ‘’GDD’’]**

\*\*

* **Static Architecture:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layer Architecture** | **Component** | **REQ\_GDD\_ID** | **Covers** | **Function prototype** | **Specification** |
| **Application** | 1) APP | Req\_GDD\_11 |  | void main(void); | The interface between the Drivers in HAL Layer as shown in the diagram |
| **HAL** | 2) LCD | Req\_GDD\_12 | Req\_SRS\_03 | u8 LCD\_u8Init (void);  u8 LCD\_u8SendCommand(u8 u8CmdCpy);  u8 LCD\_u8WriteString(u8 Copy\_\*Pu8StringCpy);  u8 LCD\_u8gotoxy(u8 u8ROW,u8 U8COLOUM); | -- Using the LCD Driver to:  - Initialize LCD Configurations:  1) Configure two lines.  2) Blinking cursor.  3) Clear display.  -Send command function used for sending commands.  -Write string function to write string on the LCD.  - Goto xy function to switch in the two lines of the LCD:  1) 2 Rows:  Addresses in first row from 0x80  Addresses in second row from 0x0C.  2) 16 Columns to switch in.  On a side note: return u8 in all functions to check the error state if it equals 1 or zero. |
| **HAL** | 3) Keypad | Req\_GDD\_13 | Req\_SRS\_03 | u8 KEYPAD\_u8init(void);  u8 KEYPAD\_u8Read (); | -- Using KEYPAD driver for:  - Initialize the configuration of the keypad by  1) define 3 pins as input and set their values to ‘1’  2) define 3 pins as output  - Use input pins for reading the pressed button on the keypad |
| **HAL** | 4) 7-segment | Req\_GDD\_14 | Req\_SRS\_05 | u8 SEVENSEG\_u8init(void);  u8 SEVENSEG\_u8Display (u8 Copy\_u8Value); | -- Using SEVENSEG driver for:  - SEVENSEG\_u8init() to set 8 pins as output  - display a number from (0-9) by passing its value to the function |
| **HAL** | 5) Button | Req\_GDD\_15 | Req\_SRS\_03  Req\_SRS\_05 | u8 BUTTON\_u8init(void);  u8 BUTTON\_u8Read(void); | -- Using the Button Driver to:  - Initialize Button Direction to Input and Value to High.  -Read the Button value by checking if it’s pressed or not. |
| **HAL** | 6) Servo Motor | Req\_GDD\_16 | Req\_SRS\_01  Req\_SRS\_02  Req\_SRS\_05 | u8 SERVOMOTOR\_u8init(void);  u8 SERVOMOTOR\_u8SetDirection (u8 Copy\_u8Angle); | -- Using the SERVOMOTOR Driver to:  - Initialize the configuration of servo by  1) make OC1A pin as output  2) set timer1 count to zero  3) set top count for timer1 in ICR1 1 register to 2499  4) Set Fast PWM, TOP in ICR1, Clear OC1A on compare match, clk/64  - set the direction by pass the angle’s value to the function |
| **HAL** | 7) Lock Sensor | Req\_GDD\_17 | Req\_SRS\_02 | u8 SWI\_u8readState (u8\* Copy\_u8State); | -- Using SWI DRIVER  --function passing to it a pointer and it checks the pin from the MCAL(DIO) if it is opened or closed |
| **HAL** | 8) Counter sensor | Req\_GDD\_18 |  | u8 COUNTERSENSOR\_u8readValue (u8\* Copy\_u8Value); | -- Using COUNTERSENSOR DRIVER  --function passing to it a pointer and it checks the pin from the MCAL(DIO) and increment the variable if its less than 4 else it return that the counter reaches maximum number |
| **LIB** | 9) Delay | Req\_GDD\_19 |  | Void Delay\_ms(u32 Time); | --Using the Delay Driver for:  - passing the required delay time in milliseconds |
| **LIB** | 10) STD\_TYPES | Req\_GDD\_20 |  | Library that included in every program file.  It contains the definition of (u8, u16, u32) |  |
| **LIB** | 11) BIT\_CALC | Req\_GDD\_21 |  | Library that included in MCAL (DIO) program file To set or toggle or clear bits. |  |
| **MCAL** | 12) DIO | Req\_GDD\_22 |  | u8 DIO\_u8init ();  u8 DIO\_u8SetPinValue (u8 Copy\_u8PinId, u8 Copy\_u8PinValue);  u8 DIO\_u8SetPinDir (u8 Copy\_u8PinNB, u8 Copy\_u8Direction);  u8 DIO\_u8GetPortValue (u8 Copy\_u8PortNB, u8 \*Copy\_Pu8Value); | -- DIO initialization and selection of port direction as input or output and port value as 5v or 0v. |

* **ELEVATOR Layered Architecture Design:**

**u8 LCD\_u8Init (void);**

**u8 LCD\_u8SendCommand(u8 u8CmdCpy);**

**u8 LCD\_u8WriteString(u8 Copy\_\*Pu8StringCpy);**

**u8 LCD\_u8gotoxy(u8 u8ROW,u8 U8COLOUM);**

**u8 SEVENSEG\_u8init(void);**

**u8 SEVENSEG\_u8Display (u8 Copy\_u8Value);**

APP

**u8 DIO\_u8SetPinDir (u8 Copy\_u8PinNB, u8 Copy\_u8Direction);**

**u8 DIO\_u8GetPortValue (u8 Copy\_u8PortNB, u8 \*Copy\_Pu8Value);**

DIO

**u8 DIO\_u8init ();**

**u8 DIO\_u8SetPinValue (u8 Copy\_u8PinId, u8 Copy\_u8PinValue);**

**u8 LOCKSENSOR\_u8readState (u8\* Copy\_u8State);**

**u8 COUNTERSENSOR\_u8readValue (u8\* Copy\_u8Value);**

Lock Sensor

Counter Sensor

**u8 BUTTON\_u8init(void);**

**u8 BUTTON\_u8Read(void);**

**u8 KEYPAD\_u8init(void);**

**u8 KEYPAD\_u8Read ();**

**u8 SERVOMOTOR\_u8init(void);**

**u8 SERVOMOTOR\_u8SetDirection (u8 Copy\_u8Angle);**

Buttons

7segment

Keypad

LCD

Servomotor

DELAY, STD\_TYEEPES, BIT\_CALC