[Design Document]

LED SEQUENCE V3.0

By Mohamed Sayed
SPRINTS

Table of Contents

TASK DESCRIPTION……………………………………………3

TASK FLOWCHART……………………………………………..4

HIGH LEVEL DESIGN…………………………………………..5

LAYERED ARCHITECTURE……………………………5

SYSTEM MODULES…...………………………………6

LOW LEVEL DESIGN……………………………………………7

APIs…………………………………………………………7

MCAL……………………………………………………7

DIO……………………………………………………7

Timer…………………………………………………7

External Interrupt………………………………7

ECUAL ………………………………………………..8

Buttons …………………………………………………8

LEDs …………………………………………………….9

APPLICATION…………………………………………10

Task Description:-

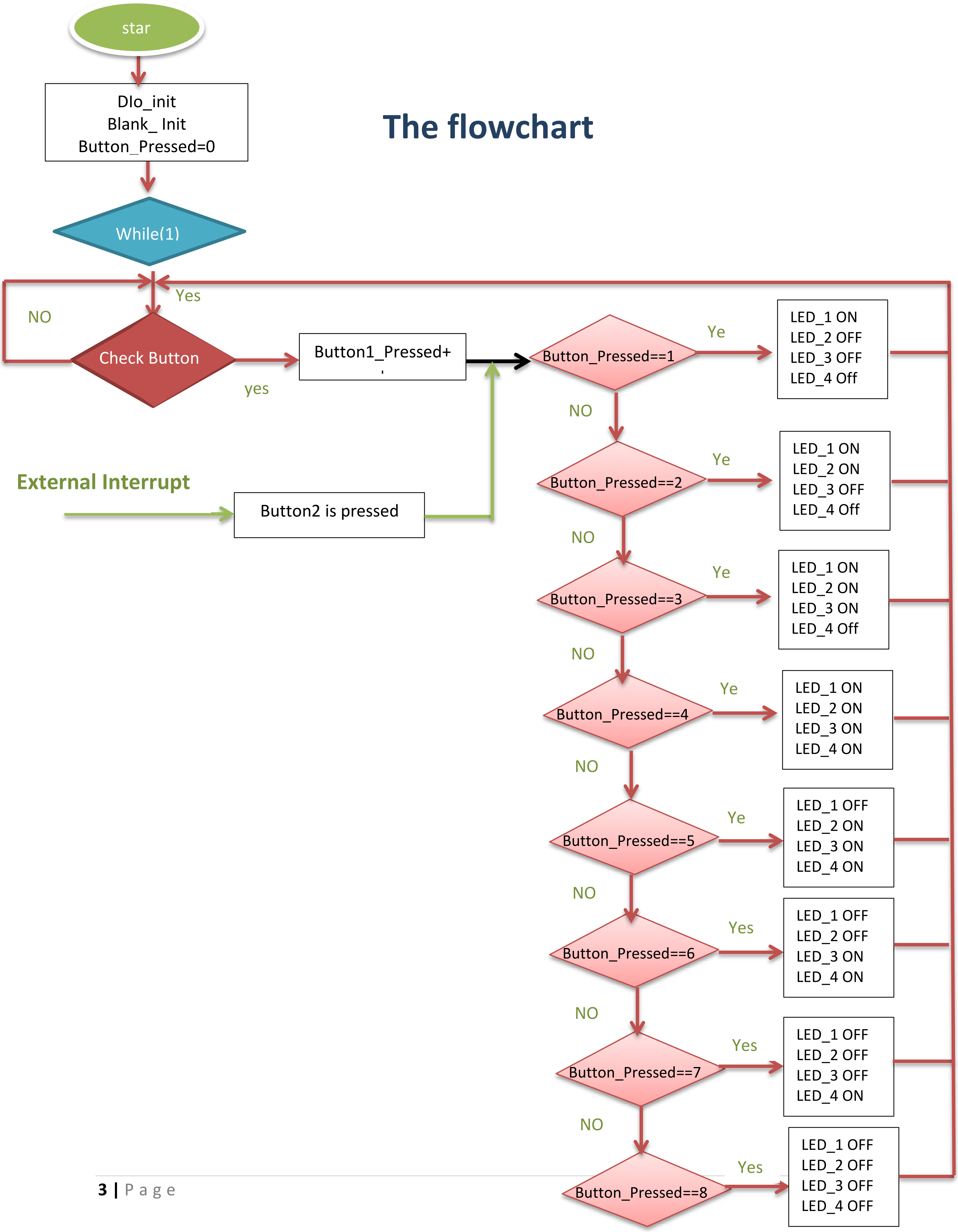
- **Hardware Requirements:**

1) Four LEDs (LED0, LED1, LED2, LED3)
2) Two Buttons (BUTTON0, BUTTON1)

- **Software Requirements:**

    1) Initially, all LEDs are OFF
    2) Once BUTTON1 is pressed, LED0 will be ON
    3) Each press further will make another LED is ON
    4) At the fifth press, LED0 will changed to be OFF
    5) Each press further will make only one LED is OFF
    6) This will be repeated forever
    7) The sequence is described below:-
        1) Initially (OFF, OFF, OFF, OFF)
        2) Press 1 (ON, OFF, OFF, OFF)
        3) Press 2 (ON, ON, OFF, OFF)
        4) Press 3 (ON, ON, ON, OFF)
        5) Press 4 (ON, ON, ON, ON)
        6) Press 5 (OFF, ON, ON, ON)
        7) Press 6 (OFF, OFF, ON, ON)
        8) Press 7 (OFF, OFF, OFF, ON)
        9) Press 8 (OFF, OFF, OFF, OFF)
        10) Press 9 (ON, OFF, OFF, OFF)
    8) When BUTTON1 has pressed the blinking on and off durations will be changed
        1) No press → BLINK_1 mode (ON: 100ms, OFF: 900ms)
        2) First press → BLINK_2 mode (ON: 200ms, OFF: 800ms)
        3) Second press → BLINK_3 mode (ON: 300ms, OFF: 700ms)
        4) Third press → BLINK_4 mode (ON: 500ms, OFF: 500ms)
        5) Fourth press → BLINK_5 mode (ON: 800ms, OFF: 200ms)
        6) Fifth press → BLINK_1 mode
    9) Use External Interrupt

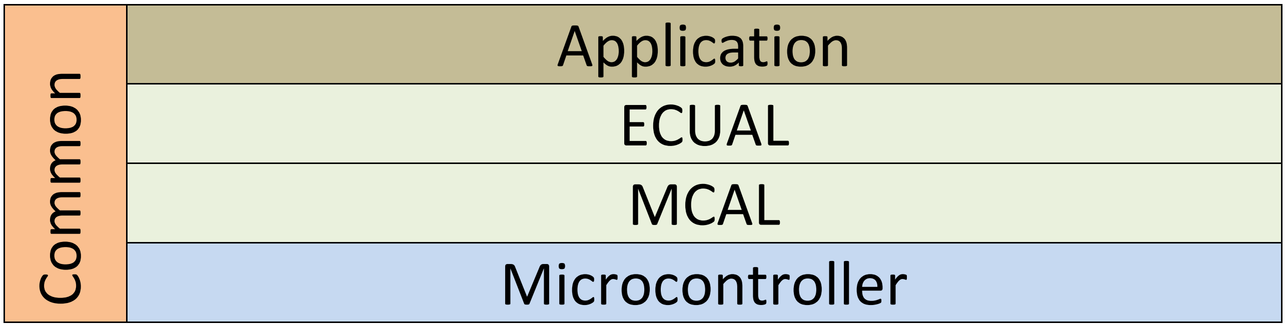
The flowchart



Layered architecture

The system may be divided to 4 layers:-

- Microcontroller
- MCAL
- ECUAl
- Application



Common

Application

ECUAL

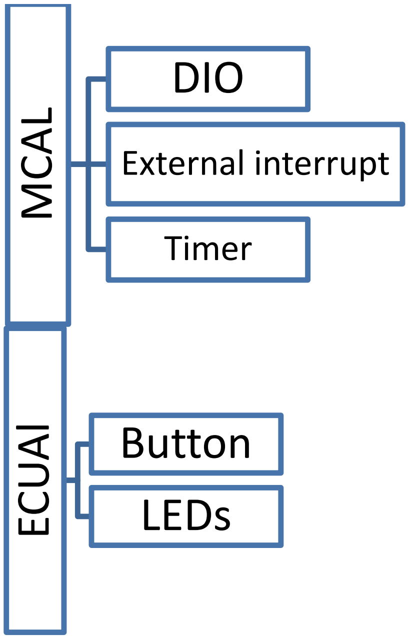
MCAL

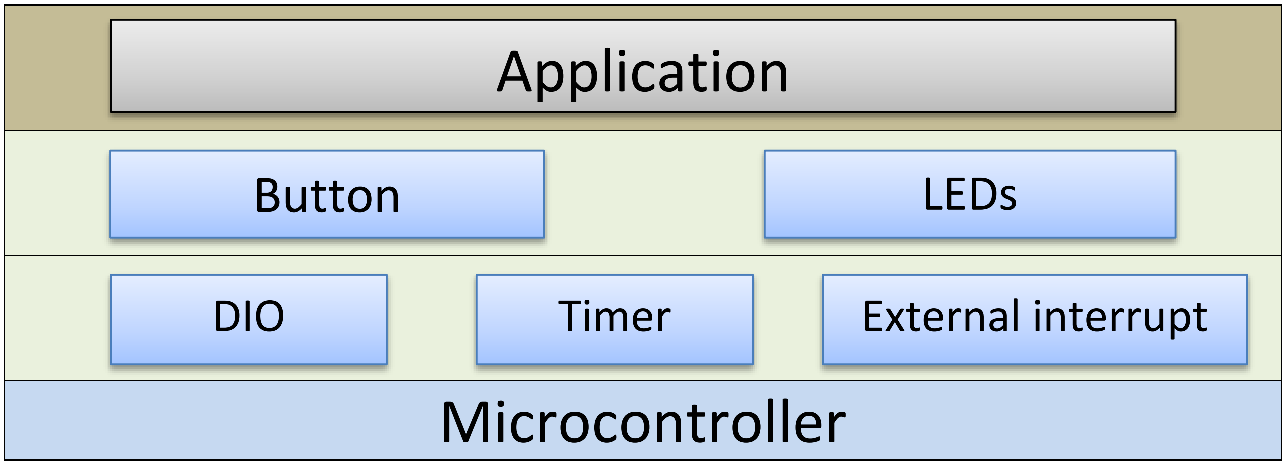
Microcontroller

High Level Design:-

System modules

The system may be divided into drivers:-





Low Level Design:-

MCAL

DIO

APIs

```
void DIO_InitPin (PIn_name pin ,PIN_Status status );
void DIO_init (void);
void DIO_WRitePin (PIn_name pin ,Voltage_type s);
Voltage_type DIO_ReadPin(PIn_name pin);
void DIO_WritePort(PORT_Type l,u8 k);
```

TIMER

APIs

```
void Timer0_init (Timer0Mode_type mode ,Timer0Scaler_type scaler);
void TIMER0_OC0Mode(OC0Mode_type mode);
void TIMER0_OV_InterruptEnable(void);
void TIMER0_OV_InterruptDisable(void);
void TIMER0_OC_InterruptEnable(void);
void TIMER0_OC_InterruptDisable(void);
void TIMER0_OV_SetCallBack(void(*local_fptr)(void));
void TIMER0_OCR_SetCallBack(void(*local_fptr)(void));
```

External interrupt

APIs

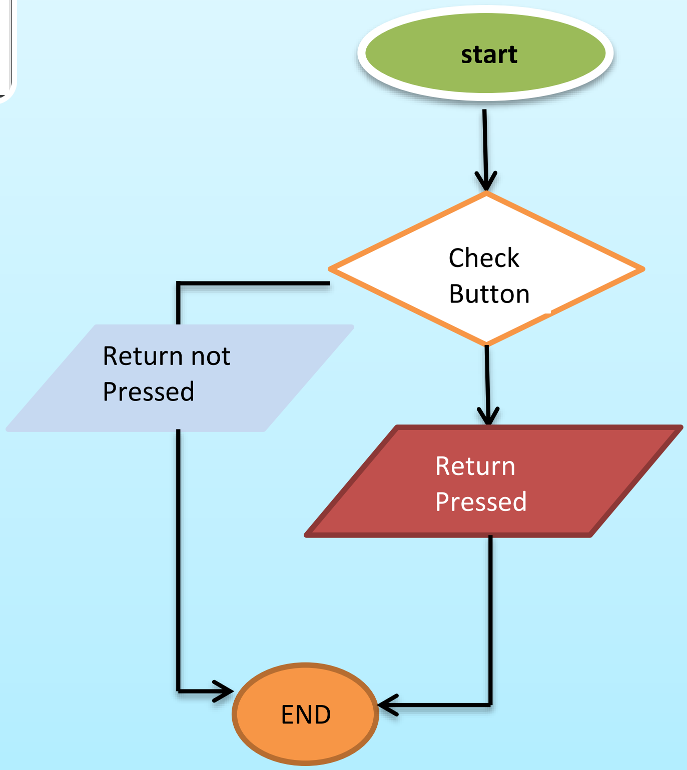
```
void EXI_Enable (ExInterruptSource_type Interrupt);
void EXI_Disable (ExInterruptSource_type Interrupt);
void EXI_Trigger(ExInterruptSource_type Interrupt,TriggerEdge_type trigger);
void EXI_SetCallBack(ExInterruptSource_type Interrupt,void(*pf)(void));
```

ECUAL

Button

APIs

Button_status Button_check(u8 ButtonNo);



LEDs

APIs

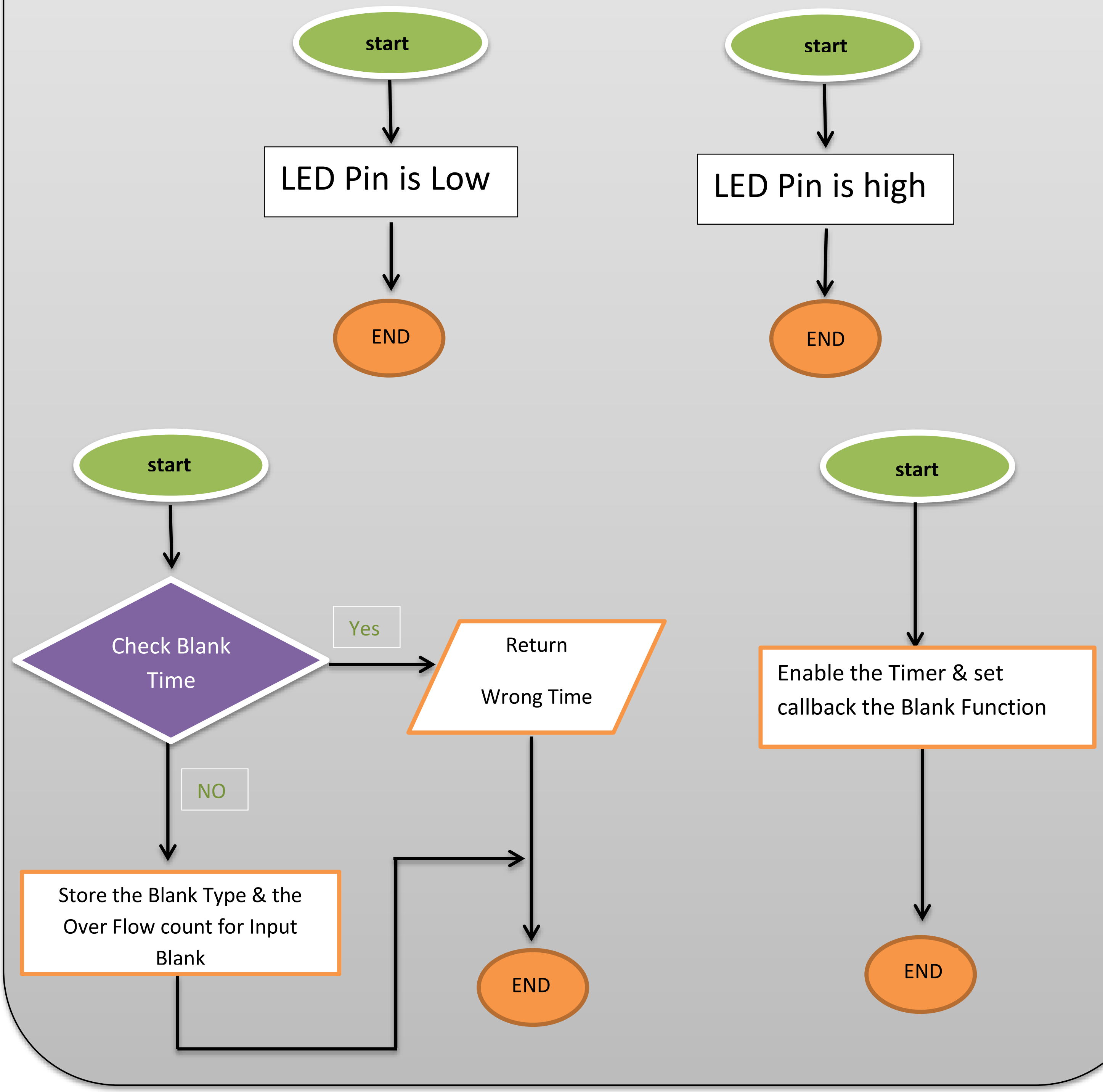
```
void LED_ON(u8 LEDno);

void LED_Off(u8 LEDno);

void LED_StopBlank(void);

Blank_Status LED_Blank_init(u32 Time_ON,u32
Time_Off,Blank_Type blank);

void LED_Blank(Blank_Type Blank);
```



start

LED Pin is Low

END

start

LED Pin is high

END

start

Check Blank Time

Yes

Return Wrong Time

NO

Store the Blank Type & the Over Flow count for Input Blank

END

start

Enable the Timer & set callback the Blank Function

END

Application

Application APIs

```
void APP_Init(void);
void APP_Start(void);
```