Sprints

LED Sequence Version 1.0

Assigned By: Mohamed Adel Abdelsalam

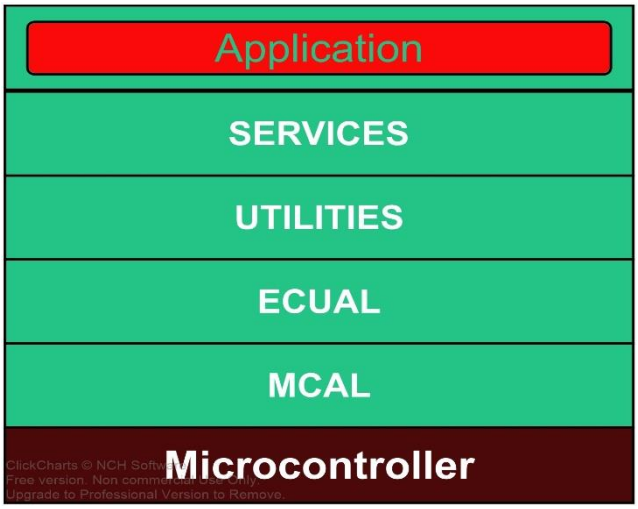
Firstly: Project Description:

1. ***Hardware Requirements***
    1. *Four LEDs (LED0, LED1, LED2, LED3)*
    2. One button (BUTTON0)
2. ***Software Requirements***
    1. Initially, all LEDs are OFF
    2. Once BUTTON0 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)

Sprints

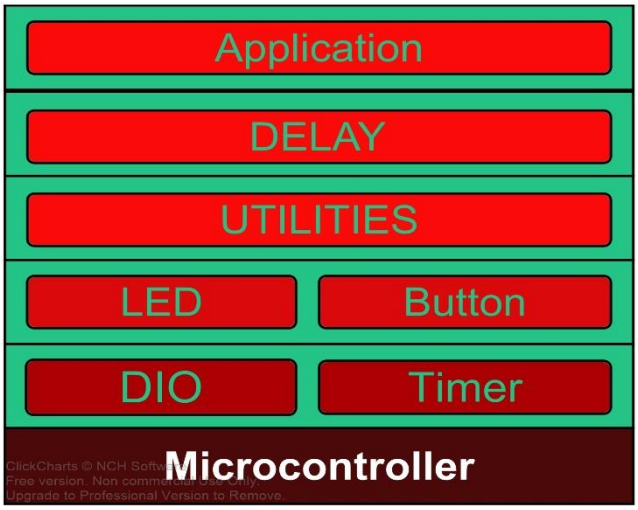
Secondly: Layered architecture:

1- Microcontroller
2- MCAL
3- ECUAL
4- UTILITIES
5- SERVICES
6- Application



Thirdly : System modules:

1- **Specify system modules/drivers:**
   - DIO, TIMER, LED, BUTTON, DELAY
2- **Assign each module to its related layer:**
   - By drawing



Forthly: APIs:

1- DIO APIs:

```
void DIO_init (uint8_t portNumber,uint8_t
      pinNumber,uint8_t direction);
void DIO_write (uint8_t portNumber,uint8_t
      pinNumber,uint8_t value);
void DIO_read (uint8_t portNumber,uint8_t
```

Sprints

2- TIMER APIs:

```
void TIMER_init (uint8_t Mode,uint8_t intial_value);
void TIMER_start (uint8_t prescaler_value);
void TIMER_set(uint8_t intial_value);
void TIMER_getStatus(uint8_t *value);
void TIMER_Stop (void);
```

3- LED APIs:

```
void LED_init (uint8_t port, uint8_t pin);
void LED_on (uint8_t port, uint8_t pin);
void LED_off (uint8_t port, uint8_t pin);
void LED_toggle (uint8_t port, uint8_t pin);
```

4- BUTTON APIs:

```
void BUTTON_init (uint8_t buttonport, uint8_t buttonpin);
void BUTTON_read (uint8_t buttonport, uint8_t buttonpin, uint8_t *value);
```

Sprints

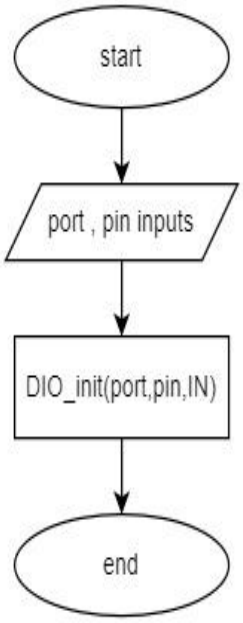
5- DELAY APIs:

```
void Delay(uint32_t milliseconds);
```

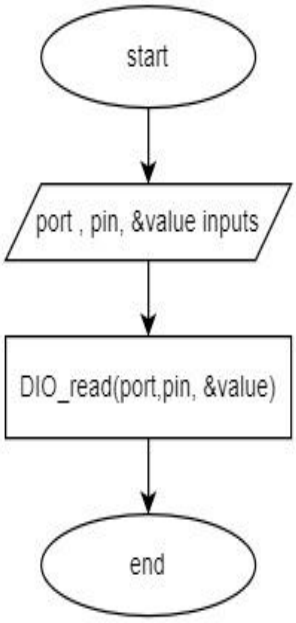
6- APPLICATION APIs:

```
void APP_init(void);
void APP_start(void);
void APP_stop(void);
```

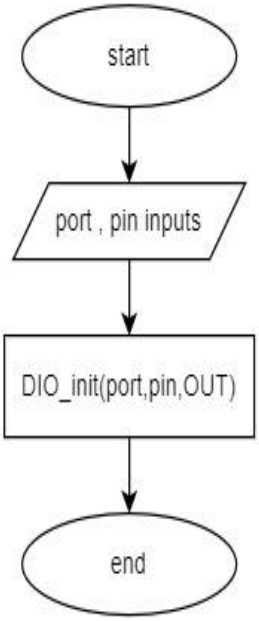
BUTTON_init(port,pin)



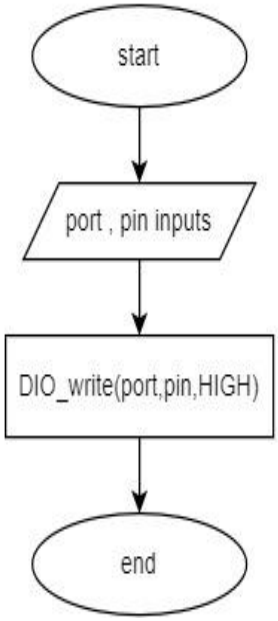
BUTTON_read(port,pin,&value)



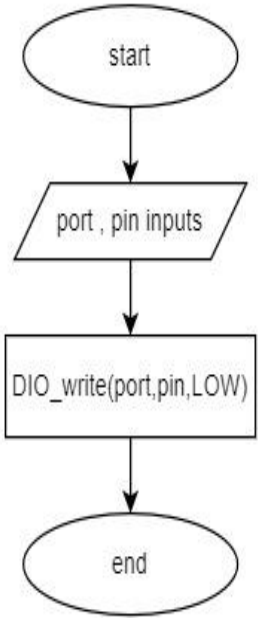
LED_init(port,pin)



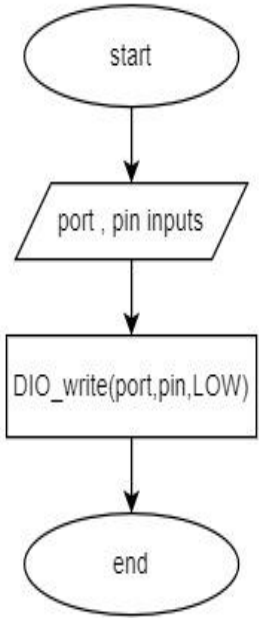
LED_on(port,pin)



LED_off(port,pin)

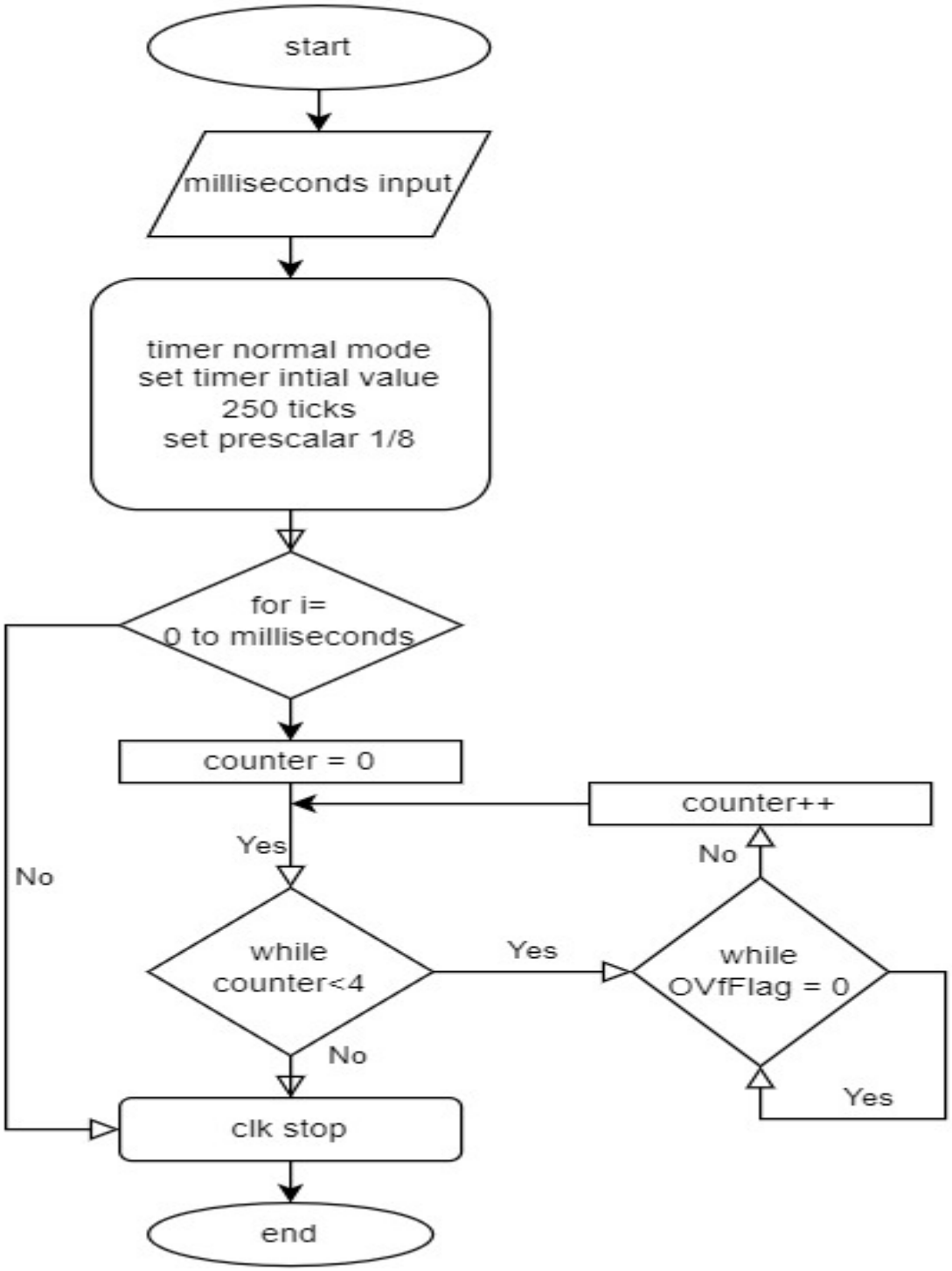


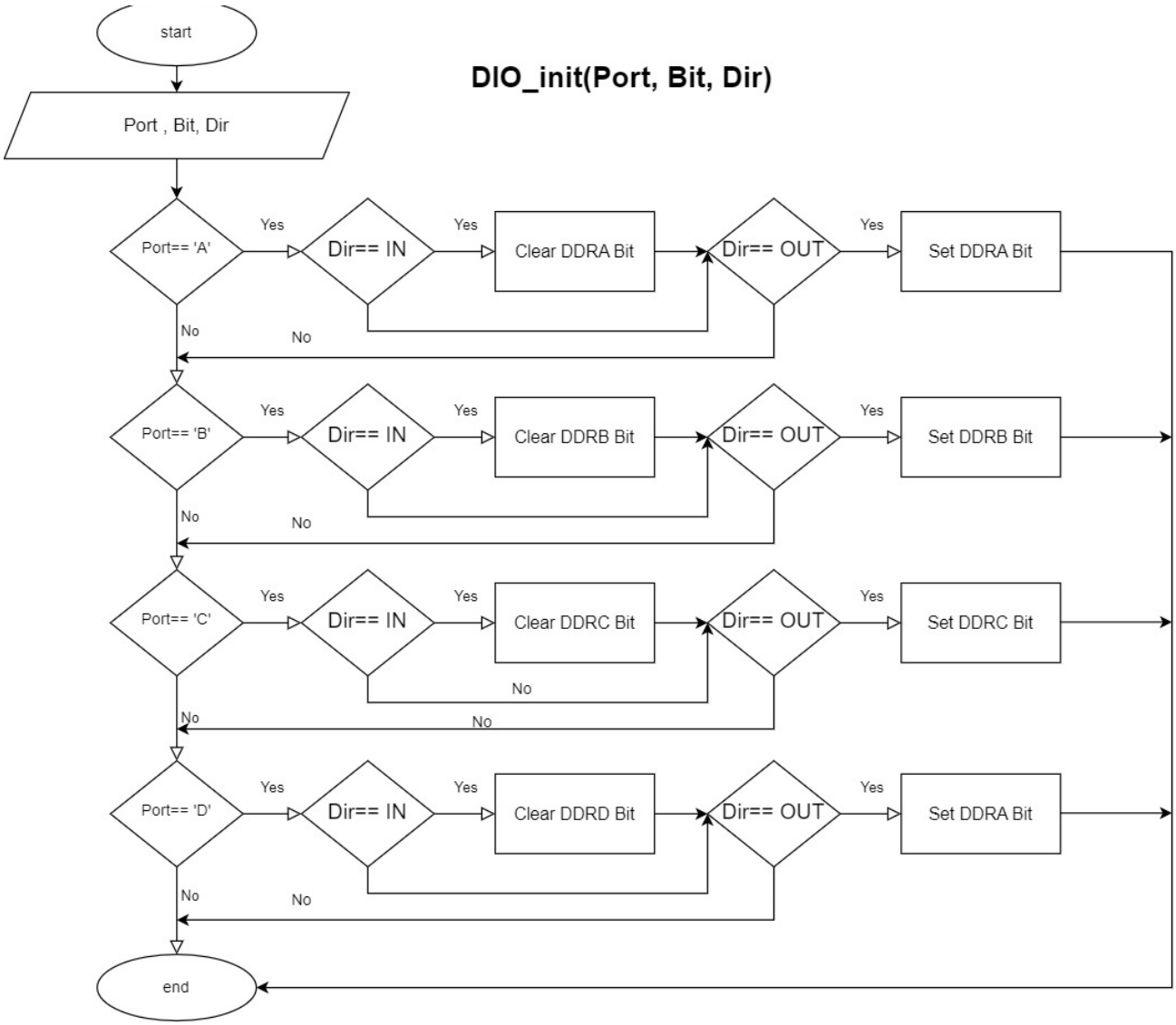
LED_toggle(port,pin)

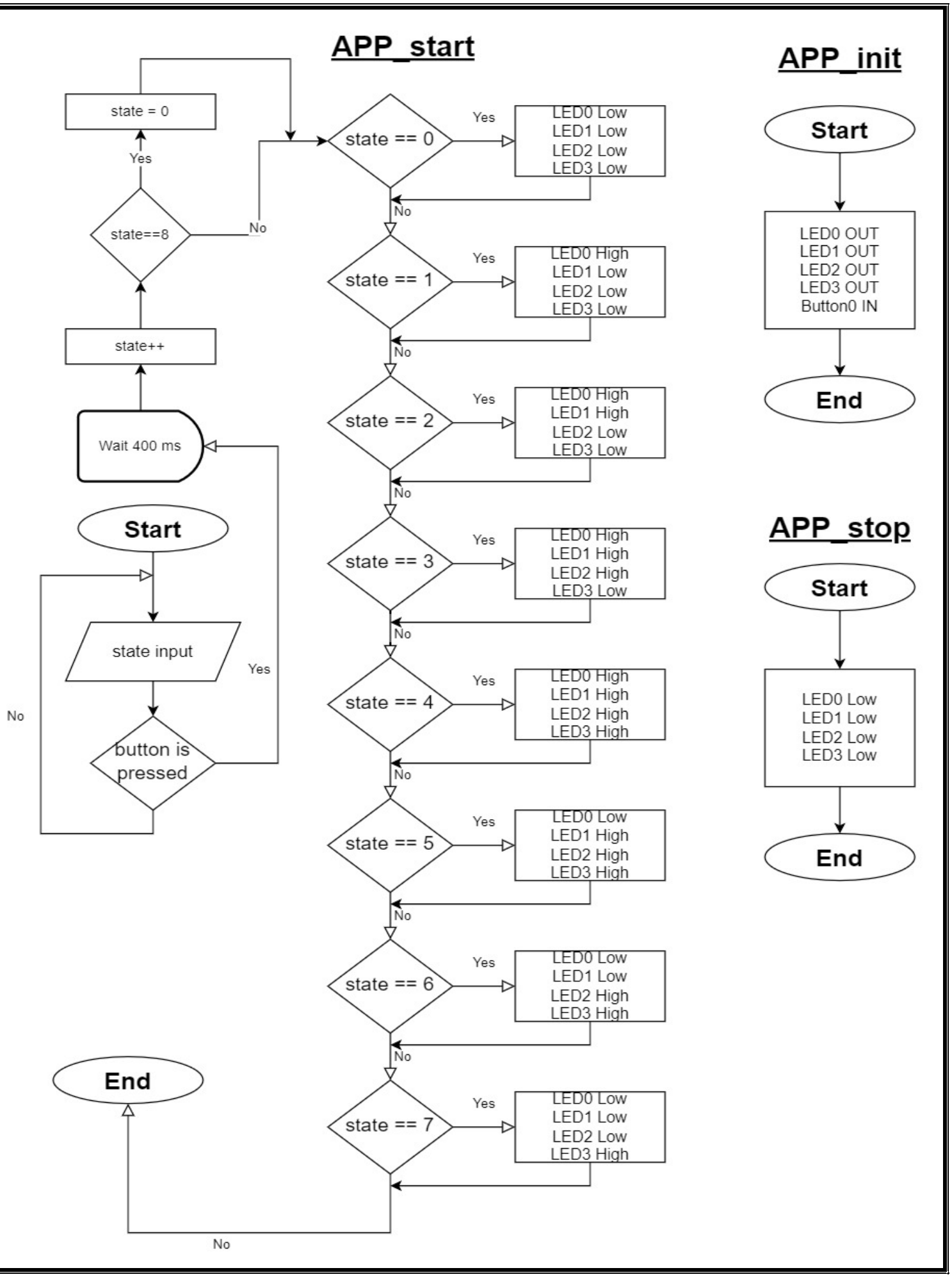


Fifthly: Flowcharts APIs:

Delay(uint8_t milliseconds)




DIO_init(Port, Bit, Dir)
start
Port , Bit, Dir
Port== 'A'
Yes
Dir== IN
Yes
Clear DDRA Bit
Dir== OUT
Yes
Set DDRA Bit
No
No
Port== 'B'
Yes
Dir== IN
Yes
Clear DDRB Bit
Dir== OUT
Yes
Set DDRB Bit
No
No
Port== 'C'
Yes
Dir== IN
Yes
Clear DDRC Bit
Dir== OUT
Yes
Set DDRC Bit
No
No
Port== 'D'
Yes
Dir== IN
Yes
Clear DDRD Bit
Dir== OUT
Yes
Set DDRA Bit
No
No
end


APP_start
state = 0
Yes
state==8
No
state++
Wait 400 ms
Start
state input
Yes
No
button is pressed
state == 0
Yes
LED0 Low
LED1 Low
LED2 Low
LED3 Low
No
state == 1
Yes
LED0 High
LED1 Low
LED2 Low
LED3 Low
No
state == 2
Yes
LED0 High
LED1 High
LED2 Low
LED3 Low
No
state == 3
Yes
LED0 High
LED1 High
LED2 High
LED3 Low
No
state == 4
Yes
LED0 High
LED1 High
LED2 High
LED3 High
No
state == 5
Yes
LED0 Low
LED1 High
LED2 High
LED3 High
No
state == 6
Yes
LED0 Low
LED1 Low
LED2 High
LED3 High
No
state == 7
Yes
LED0 Low
LED1 Low
LED2 Low
LED3 High
No
End
APP_init
Start
LED0 OUT
LED1 OUT
LED2 OUT
LED3 OUT
Button0 IN
End
APP_stop
Start
LED0 Low
LED1 Low
LED2 Low
LED3 Low
End