

LED Sequence Version 2.0

Assigned By : Mohamed Adel Abdelsalam

Firstly: Project Description:

1. *Hardware Requirements*

1. Four LEDs (**LED0**, **LED1**, **LED2**, **LED3**)
2. One button (**BUTTON1**)

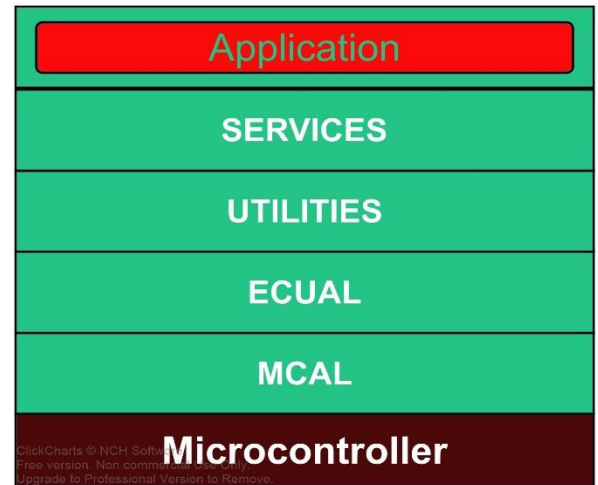
2. *Software Requirements*

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. **USE EXTERNAL INTERRUPTS**

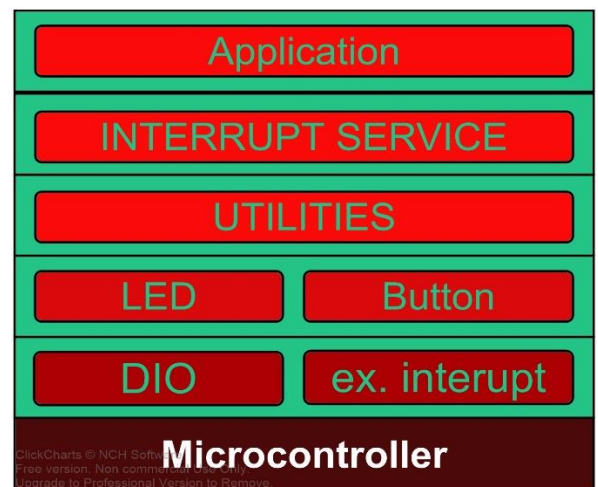
Secondly: Layered architecture:

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



Thirdly : System modules:

- 1- Specify system modules/drivers:
 - DIO, EX. INT, LED, BUTTON, APPLICATION
- 2- Assign each module to its related layer:
 - By drawing



Fourthly: 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 pinNumber,uint8_t *data);
void DIO_toggle (uint8_t portNumber,uint8_t pinNumber);
```

2- External interrupt APIs:

```
void INT_VECT(void) __attribute__((signal,used));  
void SIE(void);  
void CLI(void);  
void INT_SENSE(uint8_t inerrupt_number,uint8_t sense);  
void EX_INT_Enable(uint8_t inerrupt_number);  
void EX_INT_Disable(uint8_t inerrupt_number);  
void EX_INT0_SET_CALLBACK (void (*copyFuncptr) (void));  
void EX_INT1_SET_CALLBACK (void (*copyFuncptr) (void));  
void EX_INT_init(uint8_t interrupt , uint8_t sense);
```

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);
```

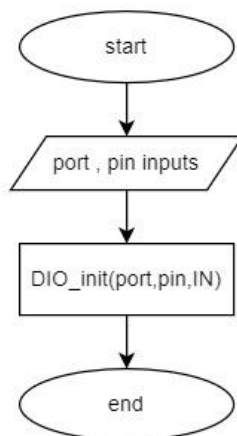
5- APPLICATION APIs:

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

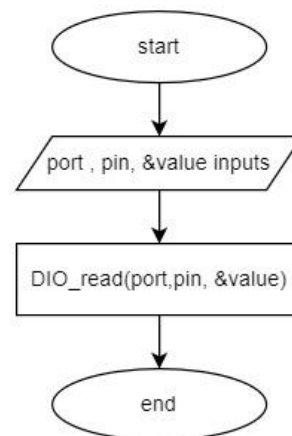
```
void APP_start(void);
void EX_INT0_ISR(void);
```

Fifthly: Flowcharts APIs:

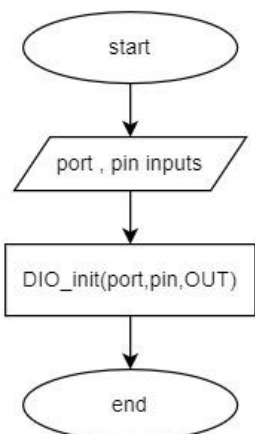
BUTTON_init(port,pin)



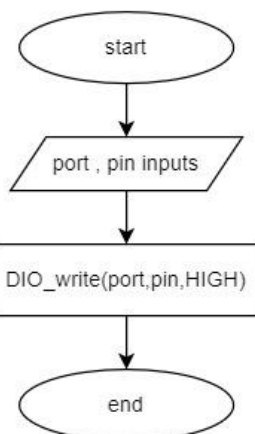
BUTTON_read(port,pin,&value)



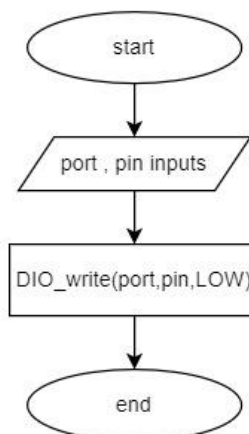
LED_init(port,pin)



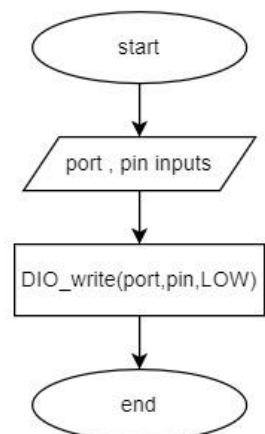
LED_on(port,pin)

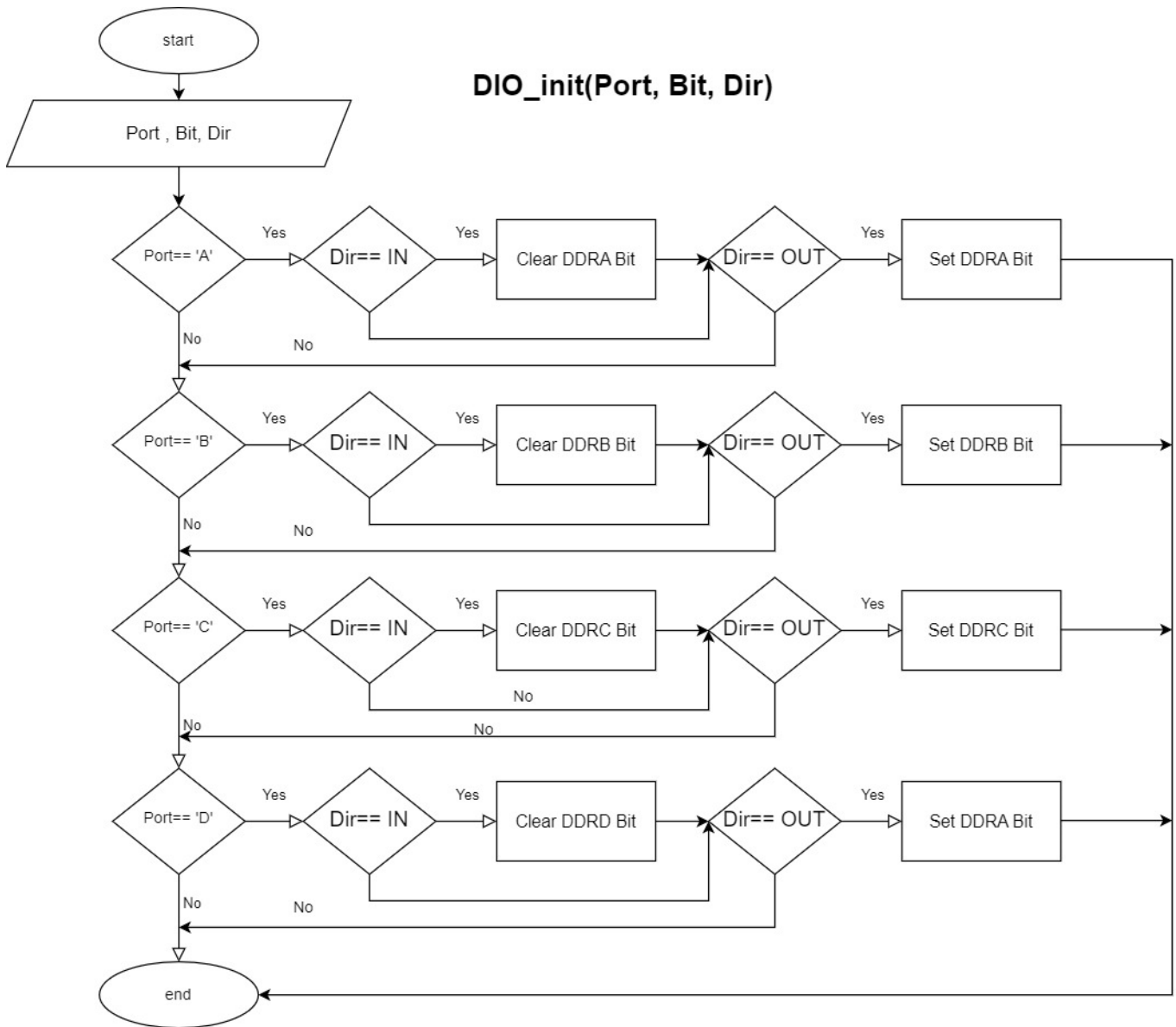


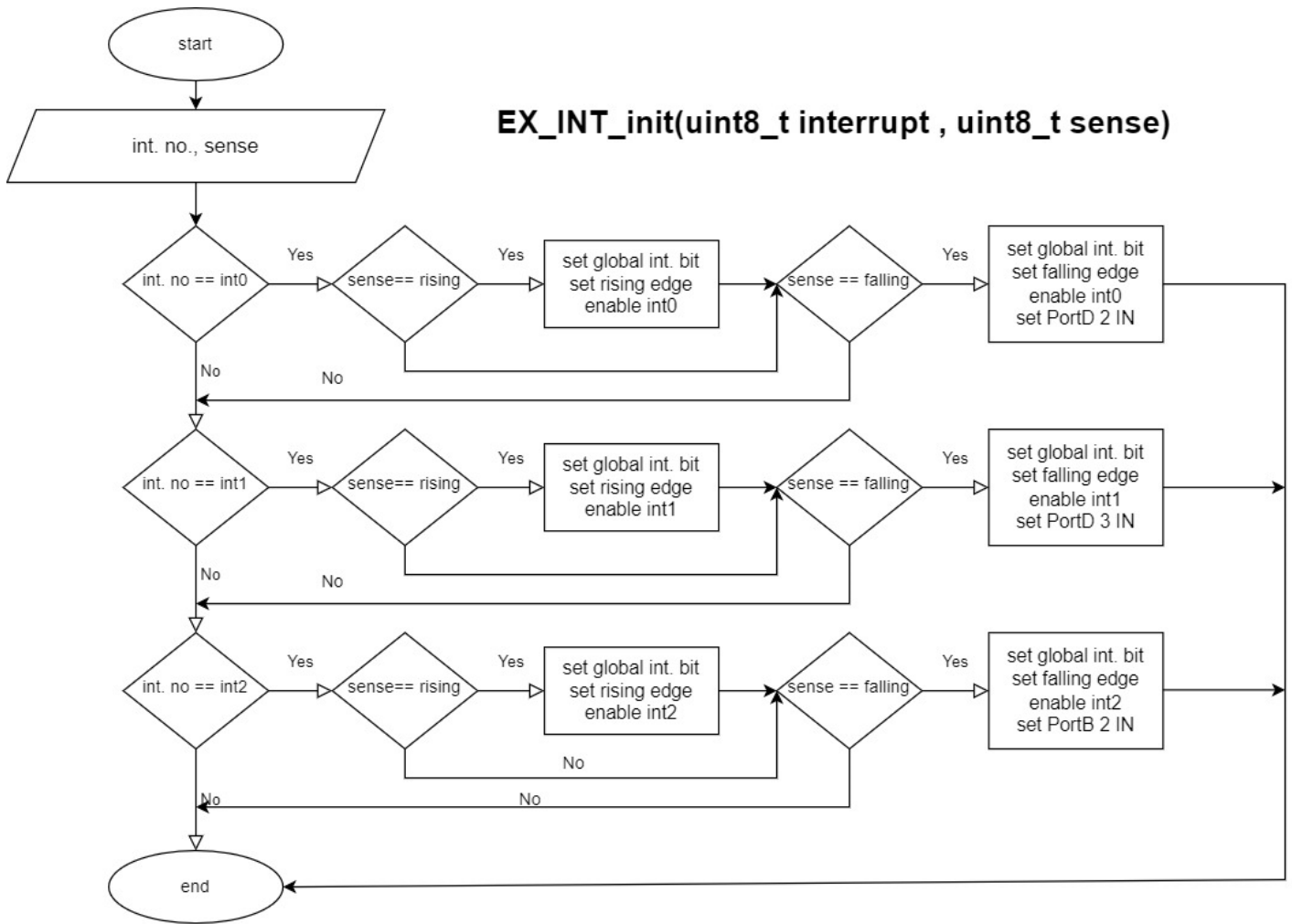
LED_off(port,pin)



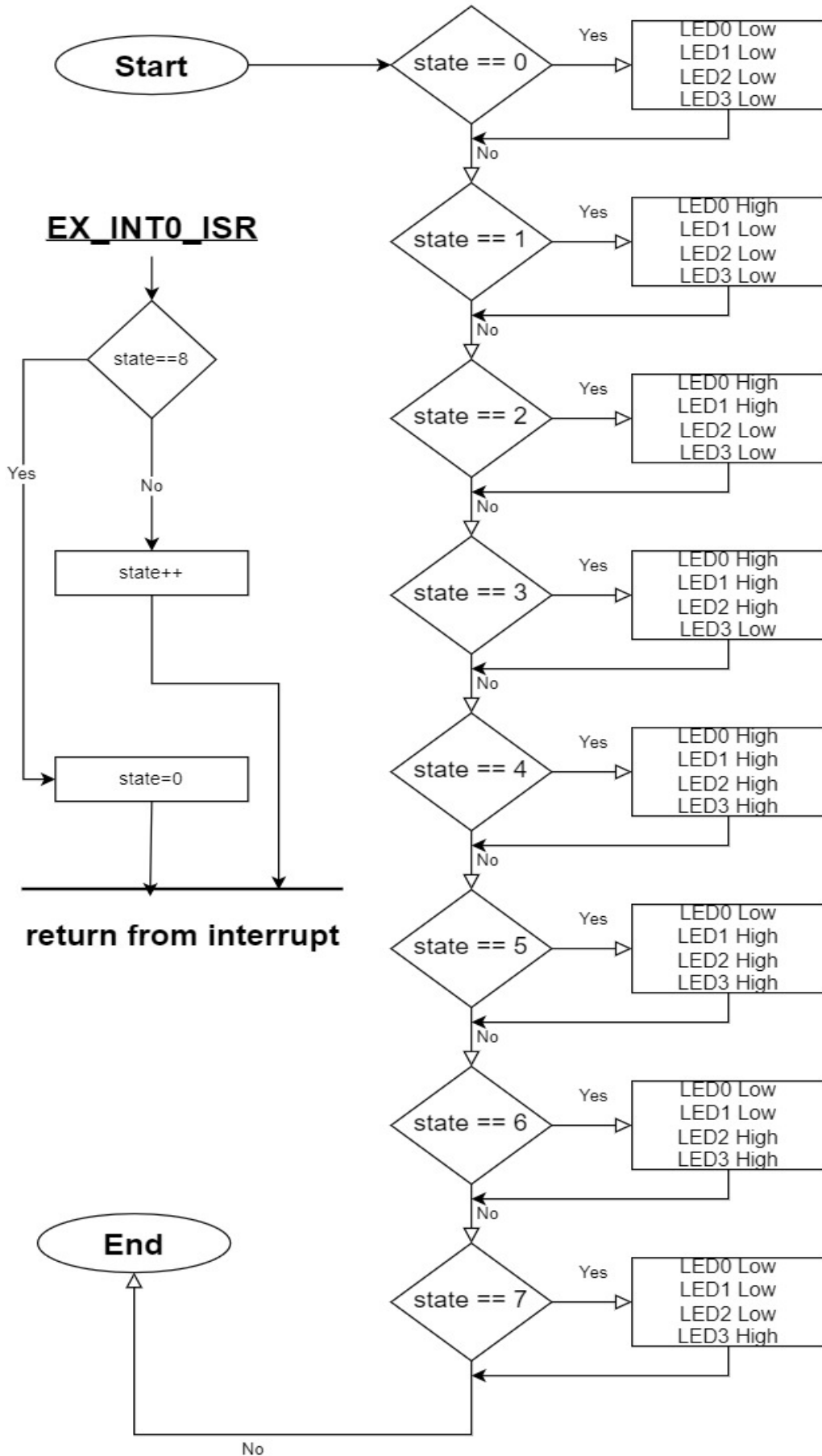
LED_toggle(port,pin)



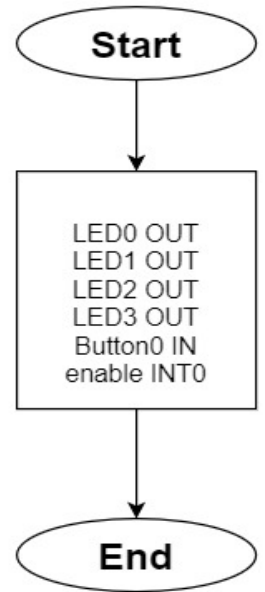




APP_start



APP_init



APP_stop

