Introduction

Project Title:

LED sequence V2.0

Name:

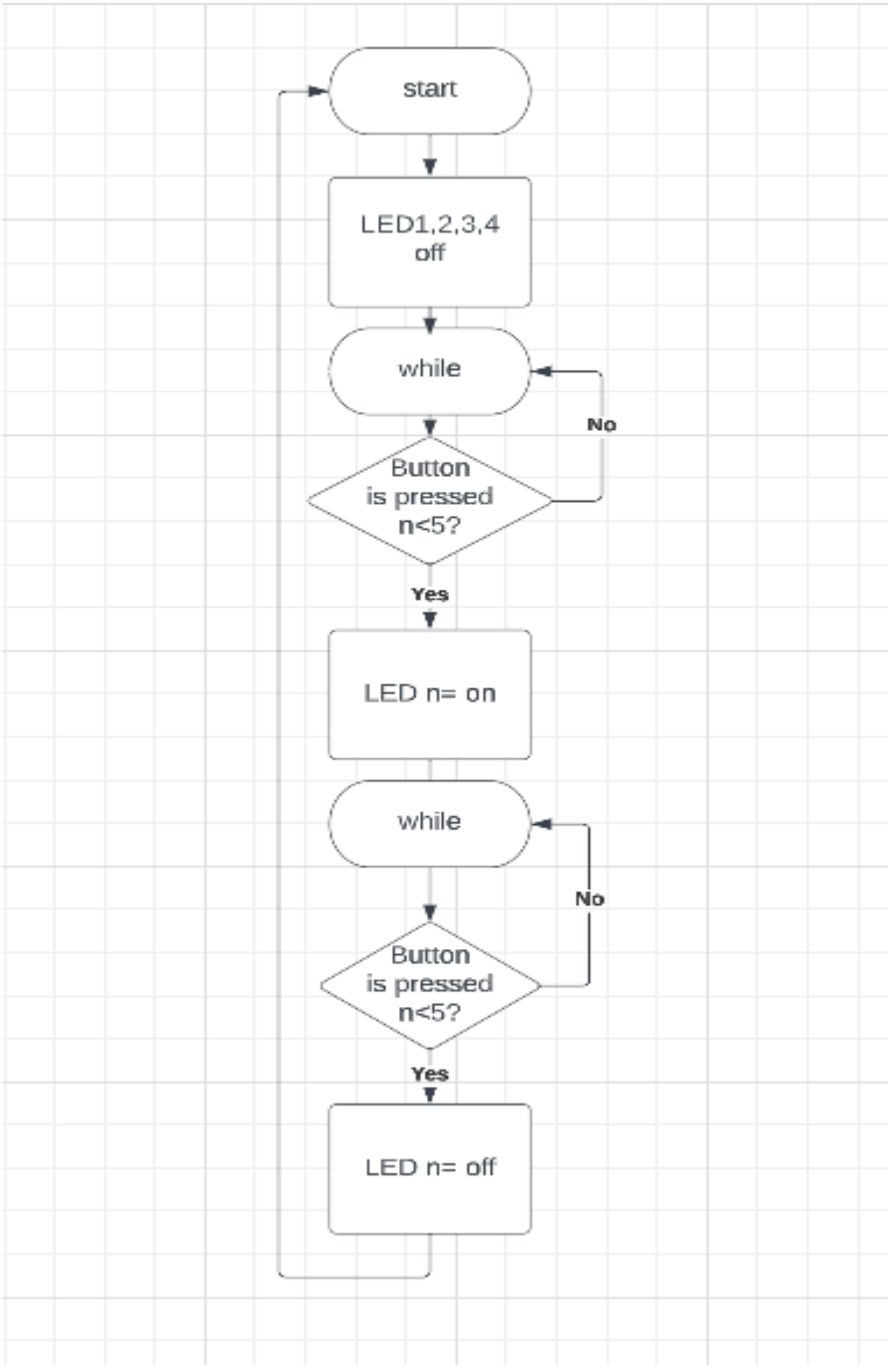
Nada Abdelazim Mohamed

Project Description:

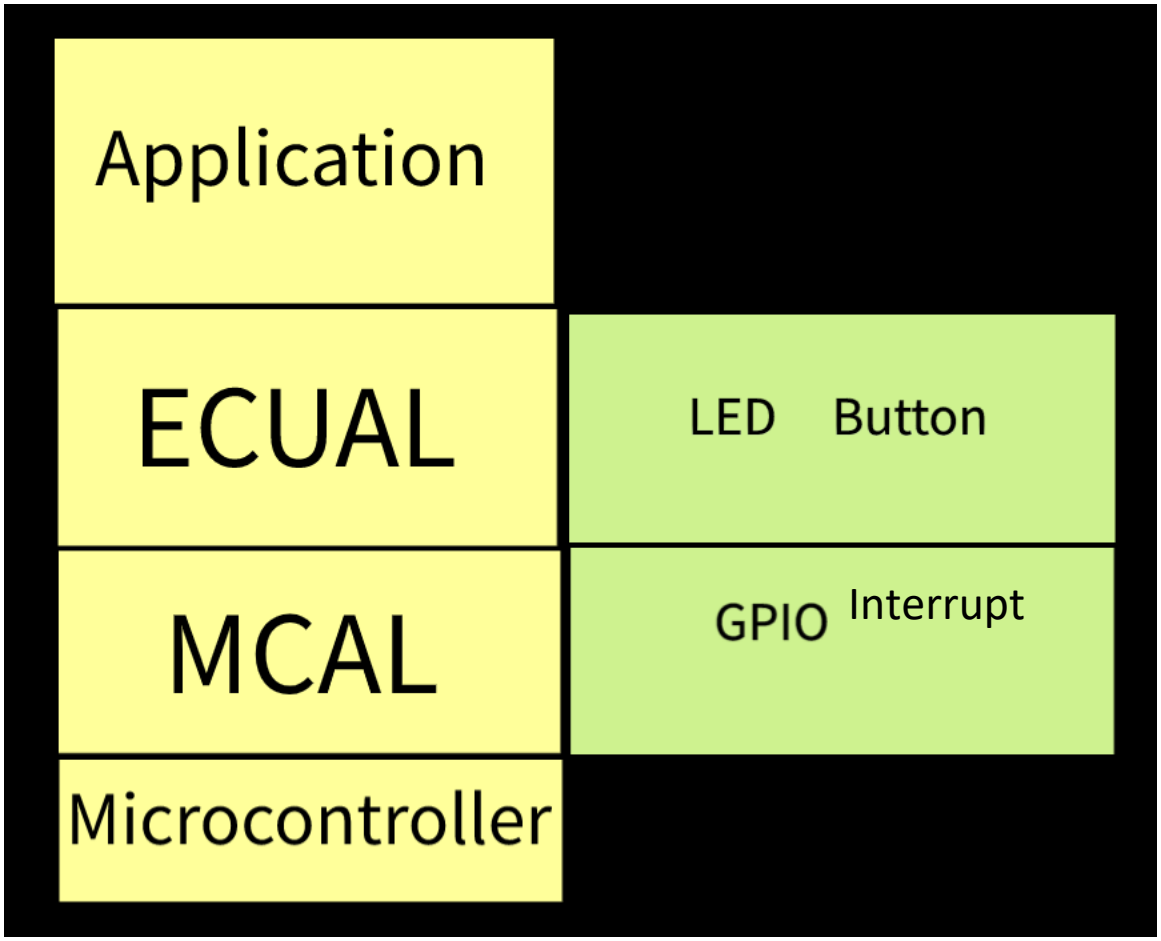
You are supposed to have a system that controls some LEDs lighting sequence according to button external interrupt.

1. ***Hardware :***
    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)

Project Flowchart



Layered architecture



Layers description:

(1)- Application layer:

Contains functions calls to implement the main project.

(2)- ECUAL: “Electronics Unit Abstraction Layer”

Contains Drivers of the external electronic devices which will be connected to the microcontroller and the system overall.

(3)-MCAL: “Microcontroller Abstraction Layer”

Contains interfaces of the microcontroller’s peripherals.

(4)-Microcontroller:

The microcontroller type that will be used to implement the project.

APIs

1- GPIO API:

Functions prototypes:

```
void DIO_init(uint8_t pinNumber, uint8_t portNumber, uint8_t direction);

void DIO_write(uint8_t pinNumber, uint8_t portNumber, uint8_t value);

void DIO_read(uint8_t pinNumber, uint8_t portNumber, uint8_t *value);
```

2- LED API:

Functions prototypes:

```
void LED_init(uint8_t ledPort, uint8_t ledPin);

void LED_on(uint8_t ledPort, uint8_t ledPin);

void LED_off(uint8_t ledPort, uint8_t ledPin);
```

3- Button API:

Functions prototypes:

```
void Button_init(uint8_t buttonPort, uint8_t buttonPin);

void Button_read(uint8_t buttonPort, uint8_t buttonPin, uint8_t *value);
```

4- Interrupt API:

Functions prototypes:

```
void Exit_enable0 (void);

void Exit_disable0 (void);

void Global_interrupt_enable (void);

void Global_interrupt_disable (void);

void Exit_init (void);

void External_interrupt_mode (uint8_t mode);
```