On demand traffic light system control

Project Description

The project aim is to make traffic light system for cars and pedestrian, So we have two modes normal mode and pedestrian mode. First the normal mode it works as follows the system consist of three LED (Green, Red, Yellow) for cars and another three for pedestrian in normal mode the green light for the cars will be on for 5 seconds then both yellow LED lights will blink for 5 seconds then the green led for the pedestrian will go on and the red LED light for car will also turns on. For the pedestrian mode there will be a push button if this push button is pressed the system will change to both yellow LED will blink and after 5 seconds the green led of the pedestrian will go on. But if someone pressed the button two times it will be ignored.

System Design

First, I had to make some drivers before writing the application like:

Register: I/O registration for PORTS (A, B, C, D).

Timer0 registration.

External interrupts registrations.

MCAL Layer:

1.DIO driver:

```
void DIO_init(uint8_t pinNumber, uint8_t portNumber , uint8_t direction)
(To initialize the Digital input output).
void DIO_write(uint8_t pinNumber, uint8_t portNumber , uint8_t value);
(This to write in the pin and set if ON or OFF).
void DIO_toggle(uint8_t pinNumber, uint8_t portNumber);
(Toggle the state of pin).
void DIO_read(uint8_t pinNumber, uint8_t portNumber , uint8_t *value);
(This function used to read the pin).
```

2.Interupt driver:

This library defines the external interrupt vectors, macros to set OR clear global register and ISR macro o Make function to initialize the interrupt

3. Timer driver:

```
void Timer0_init();
```
(This function is used to initialize timer 0).

```
void TIMER0_start(uint16_t preScalar, uint32_t number_of_overflow);
(This function is used to set the prescaler and get number of overflow).

void Timer_stop();
(This function is used to stop the timer)

void delay_5s(void);
(This function is used to set the timer delay for 5 seconds).

void Timer_delay(uint16_t delay);

(This function is used to give the timer the number of microseconds you want to
delay).
```

ECUAL Layer

LED Driver:

void LED_init(uint8_t ledPort ,uint8_t ledPin);

(This function is to initialize LED port and pin).

void LED_on(uint8_t ledPort ,uint8_t ledPin);

(This function is used to turn on the LED sends 1 the pin).

void LED_off(uint8_t ledPort ,uint8_t ledPin);

(This function is used to turn off the LED sends 1 the pin).

void LED_toggle(uint8_t ledPort ,uint8_t ledPin);

(This function is used to toggle the pin).

void toggle_LED_5s(uint8_t ledPort, uint8_t ledPin);

(This function is used to set a timer delay for 5 seconds and toggle the pin).

void toggle_2_LEDs_5s(uint8_t ledPort1, uint8_t ledPin1, uint8_t ledPort2, uint8_t ledPin2);

(This function is used to set a timer delay for 5 seconds and toggle the pin for two leds).

Button Driver:

void BUTTON_init(uint8_t buttonPort, uint8_t buttonPin);

(This function is used to initialize the button).

void BUTTON_read(uint8_t buttonPort, uint8_t buttonPin, uint8_t *value);

(This function is used to read the state of the button High or LOW.)