# On-demand Traffic light control Project Egypt-FWD

## **Content:**

- System description
- System design
- System flow chart
- System epics
- System Layers
- Solution Explorer
- Application FUNCTIONS
- System circuit

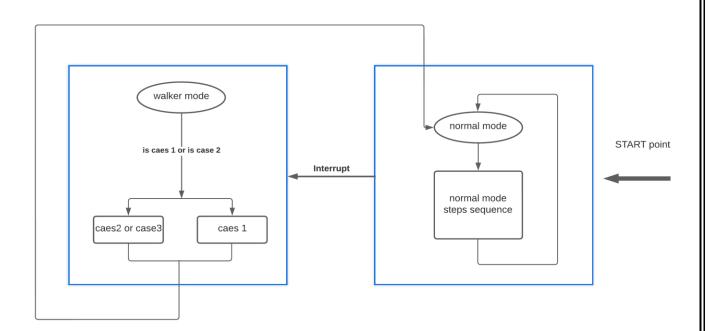
### **System description:**

traffic light system equipped with 6 led and one push button the first 3 led is for normal mode which dedicated for car movement and the second 3 led dedicated for pedestrian movement

### **System design:**

- The system is designed to run in one of two modes, Normal mode & Pedestrian mode.
- The system consists of 6 LEDs and one push button.
- It runs all the time in the normal mode unless the button is pressed, then it runs in the pedestrian mode.
- In normal mode Cars' LEDs will be changed every five seconds starting from Green then yellow then red then yellow then Green, The Yellow LED will blink for five seconds before moving to Green or Red LEDs.
- If entered pedestrian mode while car green led is on then the walker red led AND the yellow LED will be on for 5 secs then the car red and the pedestrian LED turns on.
- If entered pedestrian mode while car red led is on the walker green LED turns on for 5 seconds then the car yellow and the walker yellow LED blinks for 5 seconds then go to normal mode.

### **System flow chart:**



### **System epics:**

# [ walker mode >> case2 ,case3 ] [ walker mode >> case1 ] # it happen when car green led is on # or it happen when car yellow led is on # it happen when car red led is on turn on car green led & car yellow led & walker red led & walker yellow led (for delay 5 sec) turn on car red led & walker green led (for delay 5 sec) go to unified termination block 1- turn on car red led & walker green led (for delay 5 sec) 2- go to unified termination block [ unified termination block ] turn on car red led & car yellow led & walker green led & walker yellow led (for delay 5 sec) turn on car green led & walker red led (for delay 5 sec) tern on car green led & walker red led (for delay 5 sec)

#### [ normal mode ]

- 1- turn on car green led (for delay 5 sec)
  2- turn on car green led & blink car yellow led (for delay 5 sec)
  3- turn on car red led (for delay 5 sec)
  4- turn on car red led & blink car yellow led (for delay 5 sec)

# [ normal mode ]

- 1- turn on car green led (for delay 5 sec)
- 2- turn on car green led & blink car yellow led (for delay 5 sec)
- 3- turn on car red led (for delay 5 sec)
- 4- turn on car red led & blink car yellow led (for delay 5 sec)

#### [ walker mode >> case1 ]

- # it happen when car red led is on
- 1- turn on car red led & walker green led (for delay 5 sec)
- 2- go to unified termination block

### [ walker mode >> case2 ,case3 ]

- # it happen when car green led is on # or it happen when car yellow led is on
- 1- turn on car green led & car yellow led & walker red led & walker yellow led (for delay 5 sec)
- 2- turn on car red led & walker green led (for delay 5 sec)
- 3- go to unified termination block

### [ unified termination block ]

- 1- turn on car red led & car yellow led & walker green led & walker yellow led (for delay 5 sec)
- 2- turn on car green led & walker red led (for delay 5 sec)
- 3- return back to normal mode

# **System Layers:**

#### -APP

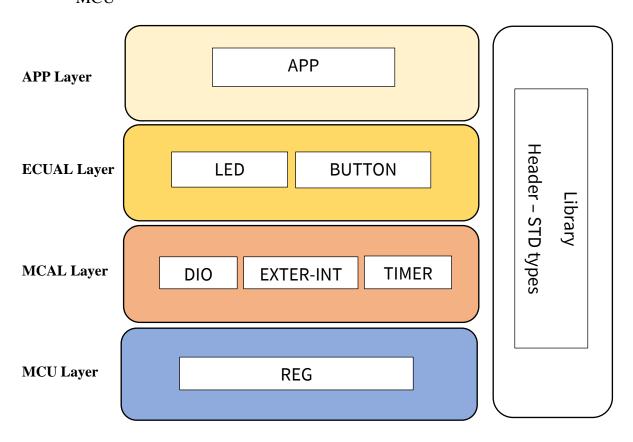
#### -ECUAL:

Electronic Unit Abstract layer, which contains any driver that represents the external hardware, such as LED, LCD, Motor or Seven segment.

#### -MCAL:

Microcontroller architecture layer, which contains any driver that represents the internal peripheral of the MCU, such as DIO, interrupt, ADC, UART... etc..

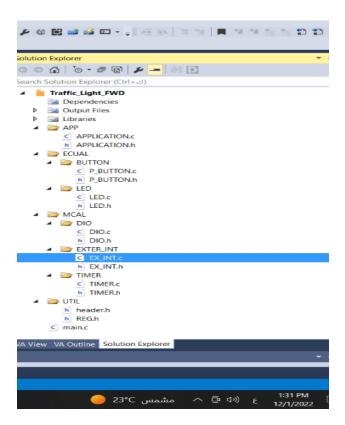
#### -MCU



### **System constrains:**

There are no constrains.

## **Solution Explorer:**



## **Application FUNCTIONS:**

```
1- delay_In_ms(uint32_t TIME)
2- LED_ON(uint8_t LED_PORT, uint8_t LED_PIN);
3- LED_OFF(uint8_t LED_PORT, uint8_t LED_PIN);
4- ON_Green_led_Walker();
```

# **System Circuit:**

