On-demand Traffic light control

-Content.

1.System description.
2.System design.
3.System flowchart.
4.Constraints.

1.System description.

- This system is a simulation of the traffic lights, with a pedestrian button to allow the pedestrian to cross the street, when the pedestrian press the button.

1.System description.

- For the regular mood the traffic lights changed every 5 seconds sequentially.
CARS :green ,yellow (blink 1s) ,red.
Pedestrian:Red , yellow (blink 1s),green
- If the Pedestrian Button is Pressed while the pedestrian red light is on the yellow light will Blinking for 5 s then the green led will be on for 5s if the pedestrian button is pressed again after 3s from turning the green led on it will reset the 5s again before the 3s it will do nothing.

1.System description.

- If the Pedestrian Button is Pressed while the pedestrian yellow is blinking it will stay until the 5s is finished then the green light becomes on if the pedestrian button is pressed again after 3s from turning the green led on it will reset the 5s again before the 3s it will do nothing.
- If the pedestrian green light is on and the button is pressed the 5s will reset again if the pedestrian button is pressed again after 3s from turning the green led on it will reset the 5s again before the 3s it will do nothing.

2.System design.

- The system is designed by two main layers

  .Microcontroller abstraction layer
  .Application layer.

- Microcontroller abstraction layer.

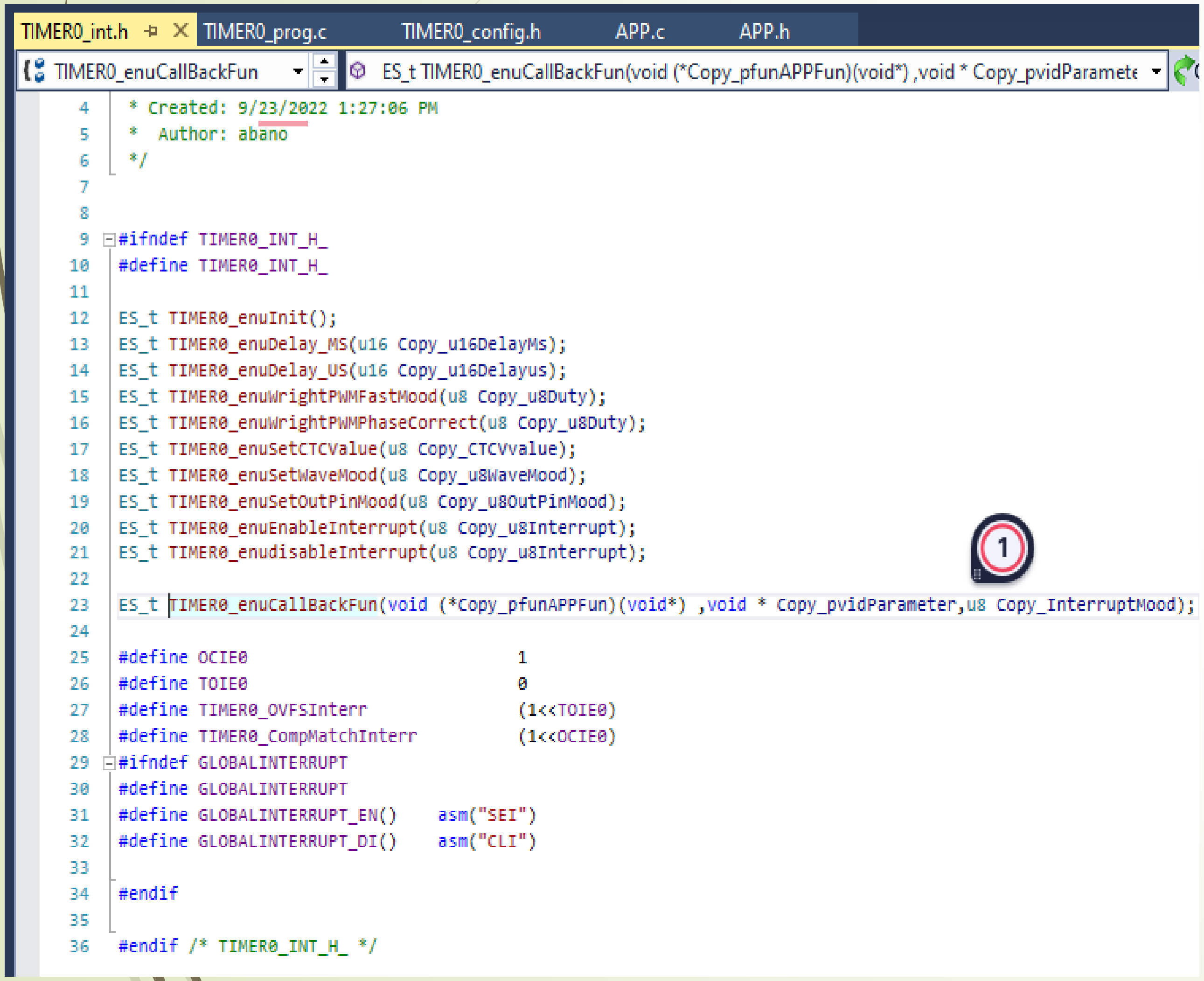
  this layer content of three Drivers (digital input-output ,external interrupt ,timer_0)

2.System design.

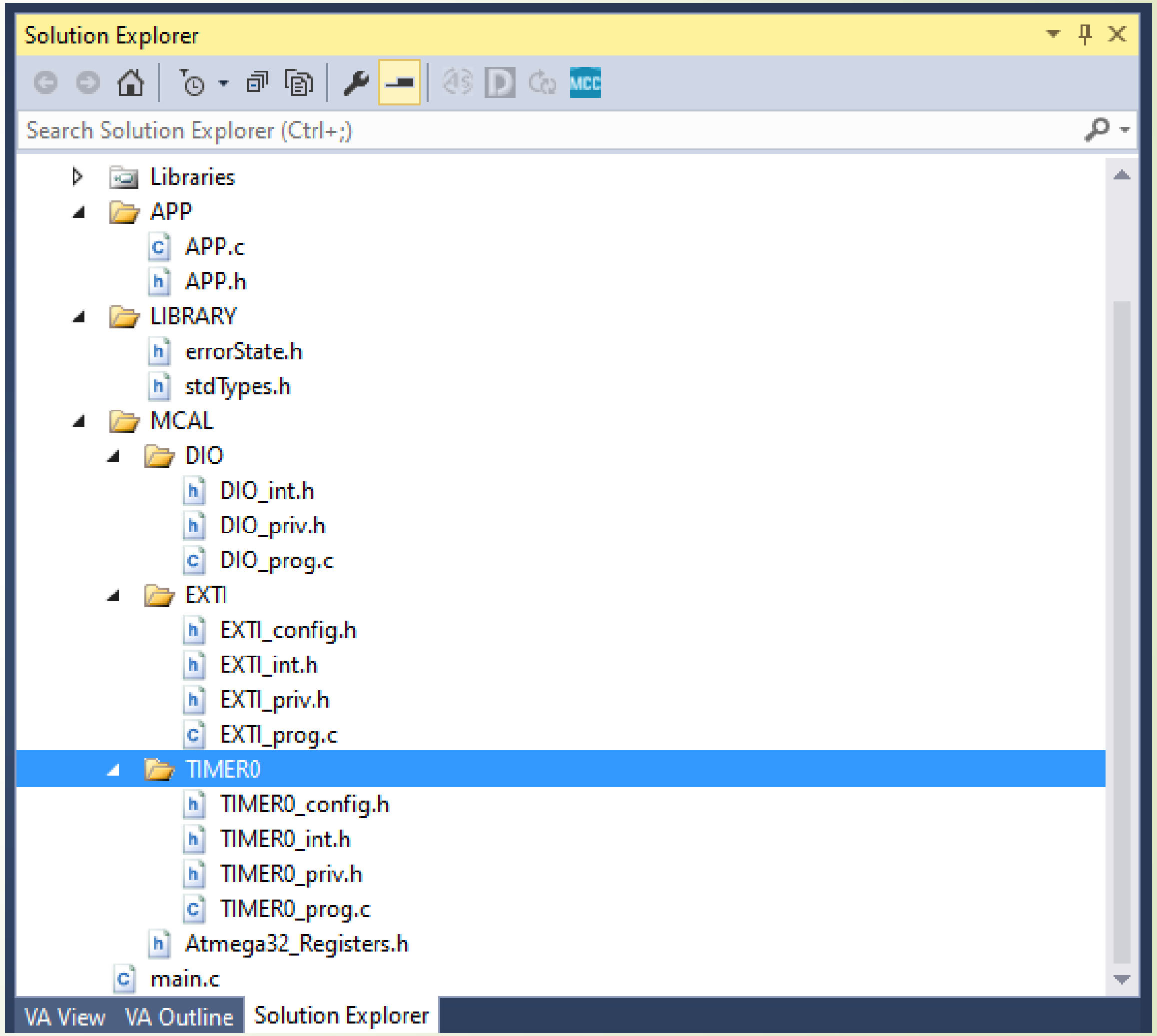
- Application layer.

  this layer content of two main functions initialization function that initializes the Drivers by calling the required functions from the MCAL layer, the second function is the ablation function is the app start function this function controls the output bins valued depending on which status is running. and two interrupt functions for the Pedestrian button and timer overflow interrupt function for counting the time.

2.System design.

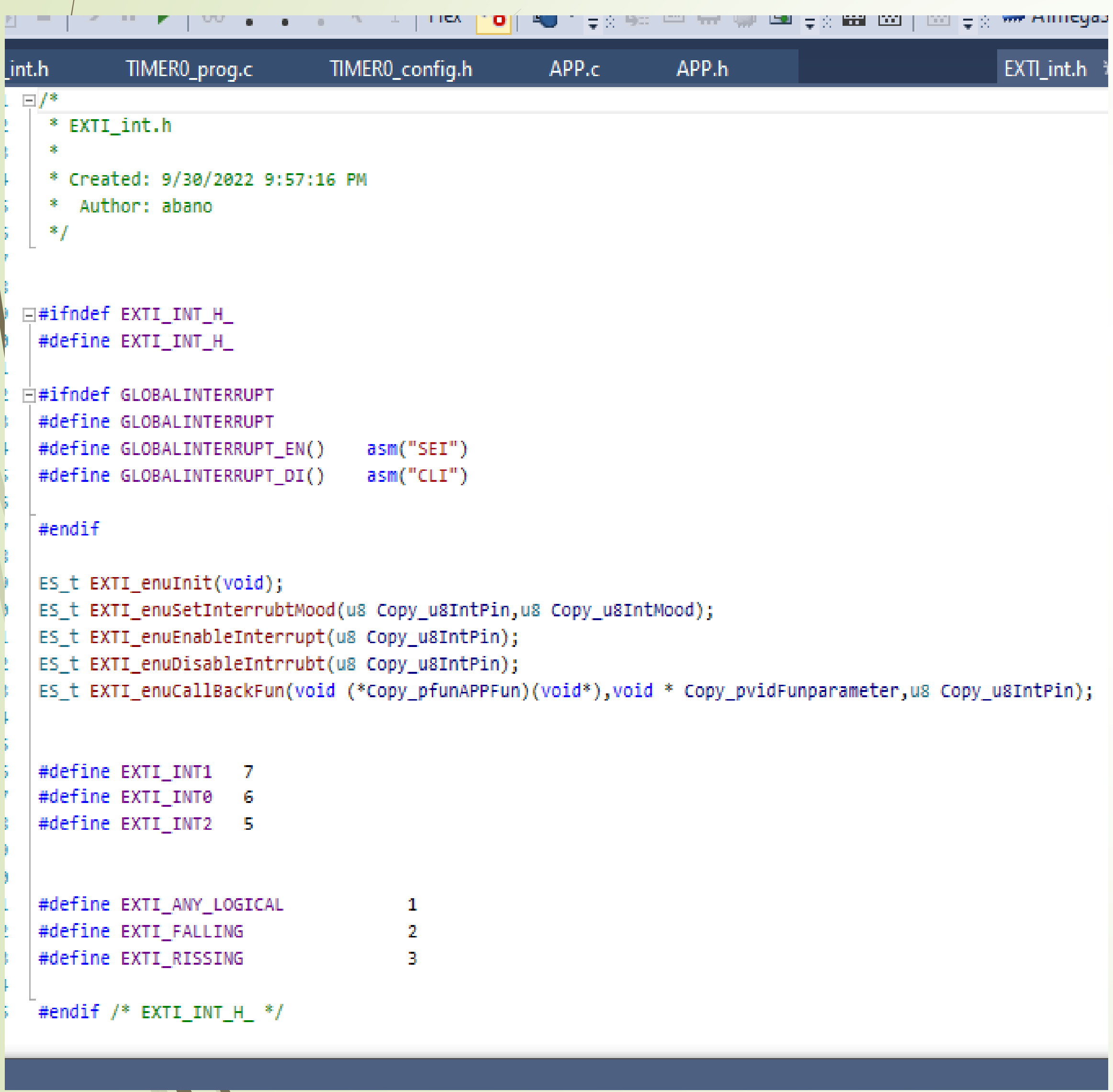

TIMER0_int.h
TIMER0_prog.c
TIMER0_config.h
APP.c
APP.h
TIMER0_enuCallBackFun
ES_t TIMER0_enuCallBackFun(void (*Copy_pfunAPPFun)(void*) ,void * Copy_pvidParamete
1

```c
 * Created: 9/23/2022 1:27:06 PM
 *  Author: abano
 */


#ifndef TIMER0_INT_H_
#define TIMER0_INT_H_

ES_t TIMER0_enuInit();
ES_t TIMER0_enuDelay_MS(u16 Copy_u16DelayMs);
ES_t TIMER0_enuDelay_US(u16 Copy_u16Delayus);
ES_t TIMER0_enuWrightPWMFastMood(u8 Copy_u8Duty);
ES_t TIMER0_enuWrightPWMPhaseCorrect(u8 Copy_u8Duty);
ES_t TIMER0_enuSetCTCValue(u8 Copy_CTCValue);
ES_t TIMER0_enuSetWaveMood(u8 Copy_u8WaveMood);
ES_t TIMER0_enuSetOutPinMood(u8 Copy_u8OutPinMood);
ES_t TIMER0_enuEnableInterrupt(u8 Copy_u8Interrupt);
ES_t TIMER0_enudisableInterrupt(u8 Copy_u8Interrupt);

ES_t TIMER0_enuCallBackFun(void (*Copy_pfunAPPFun)(void*) ,void * Copy_pvidParameter,u8 Copy_InterruptMood);

#define OCIE0                       1
#define TOIE0                       0
#define TIMER0_OVFSInterr           (1<<TOIE0)
#define TIMER0_CompMatchInterr      (1<<OCIE0)
#ifndef GLOBALINTERRUPT
#define GLOBALINTERRUPT
#define GLOBALINTERRUPT_EN()    asm("SEI")
#define GLOBALINTERRUPT_DI()    asm("CLI")

#endif

#endif /* TIMER0_INT_H_ */
```

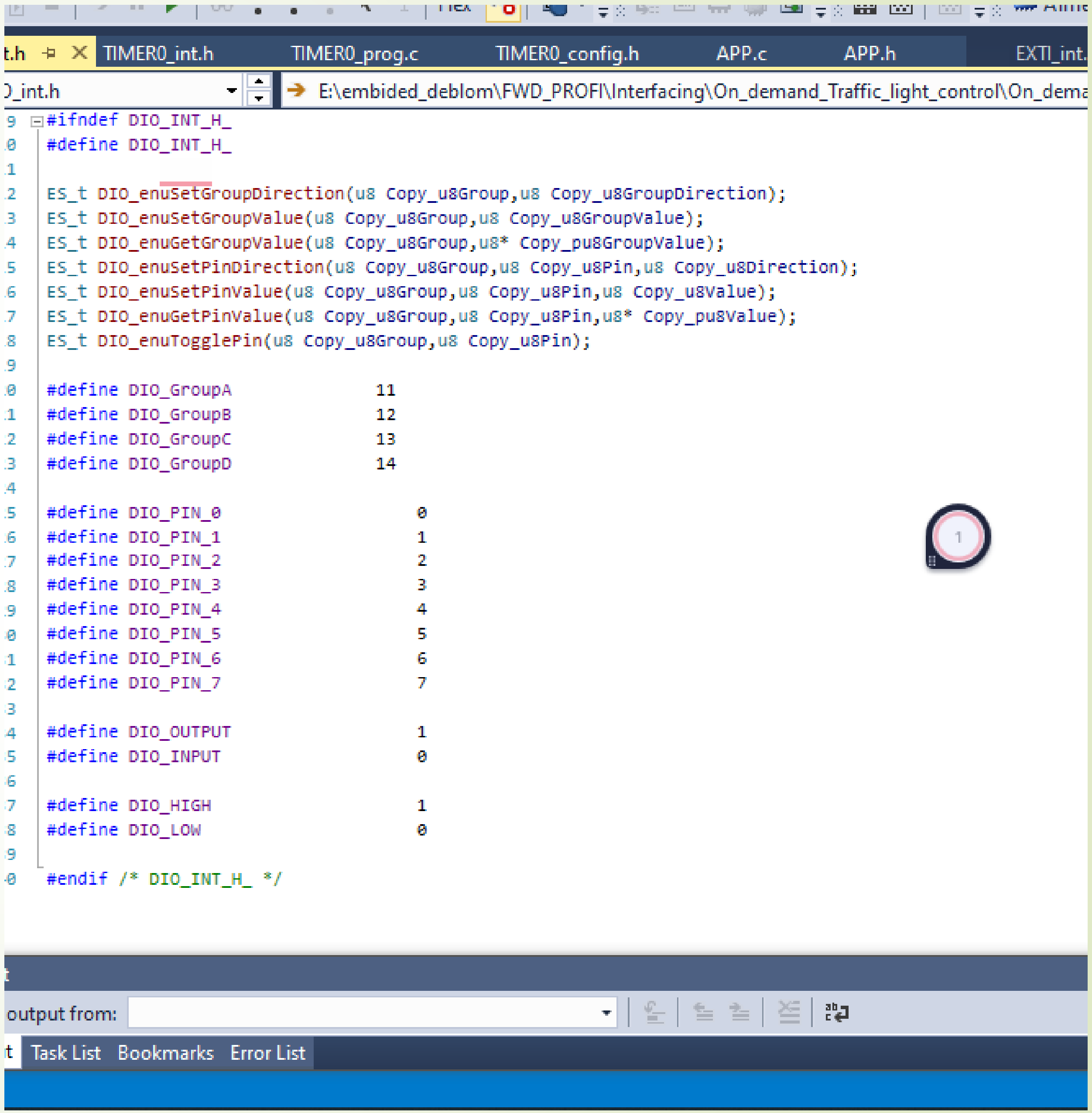


```
Libraries
APP
    APP.c
    APP.h
LIBRARY
    errorState.h
    stdTypes.h
MCAL
    DIO
        DIO_int.h
        DIO_priv.h
        DIO_prog.c
    EXTI
        EXTI_config.h
        EXTI_int.h
        EXTI_priv.h
        EXTI_prog.c
    TIMER0
        TIMER0_config.h
        TIMER0_int.h
        TIMER0_priv.h
        TIMER0_prog.c
    Atmega32_Registers.h
main.c
```

2.System design.

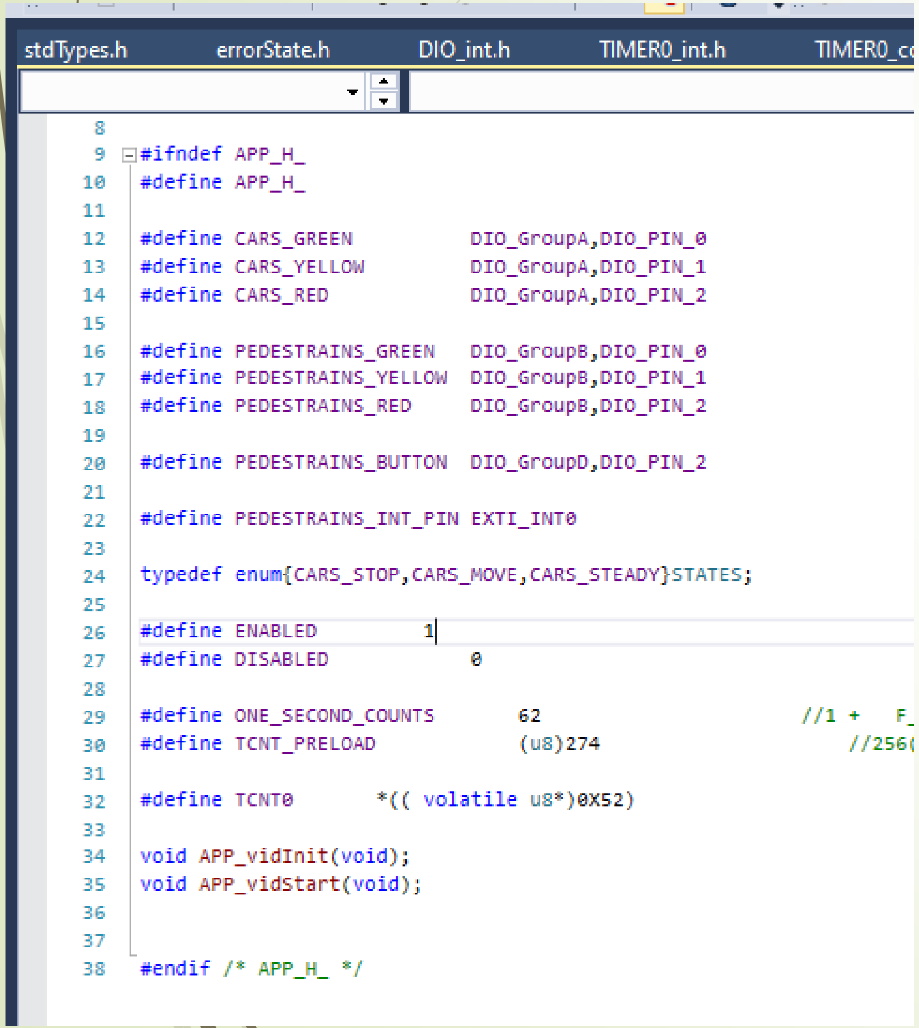


```c
/*
 * EXTI_int.h
 *
 * Created: 9/30/2022 9:57:16 PM
 *  Author: abano
 */


#ifndef EXTI_INT_H_
#define EXTI_INT_H_

#ifndef GLOBALINTERRUPT
#define GLOBALINTERRUPT
#define GLOBALINTERRUPT_EN()    asm("SEI")
#define GLOBALINTERRUPT_DI()    asm("CLI")

#endif

ES_t EXTI_enuInit(void);
ES_t EXTI_enuSetInterrubtMood(u8 Copy_u8IntPin,u8 Copy_u8IntMood);
ES_t EXTI_enuEnableInterrupt(u8 Copy_u8IntPin);
ES_t EXTI_enuDisableIntrrubt(u8 Copy_u8IntPin);
ES_t EXTI_enuCallBackFun(void (*Copy_pfunAPPFun)(void*),void * Copy_pvidFunparameter,u8 Copy_u8IntPin);


#define EXTI_INT1   7
#define EXTI_INT0   6
#define EXTI_INT2   5


#define EXTI_ANY_LOGICAL            1
#define EXTI_FALLING                2
#define EXTI_RISSING                3

#endif /* EXTI_INT_H_ */
```

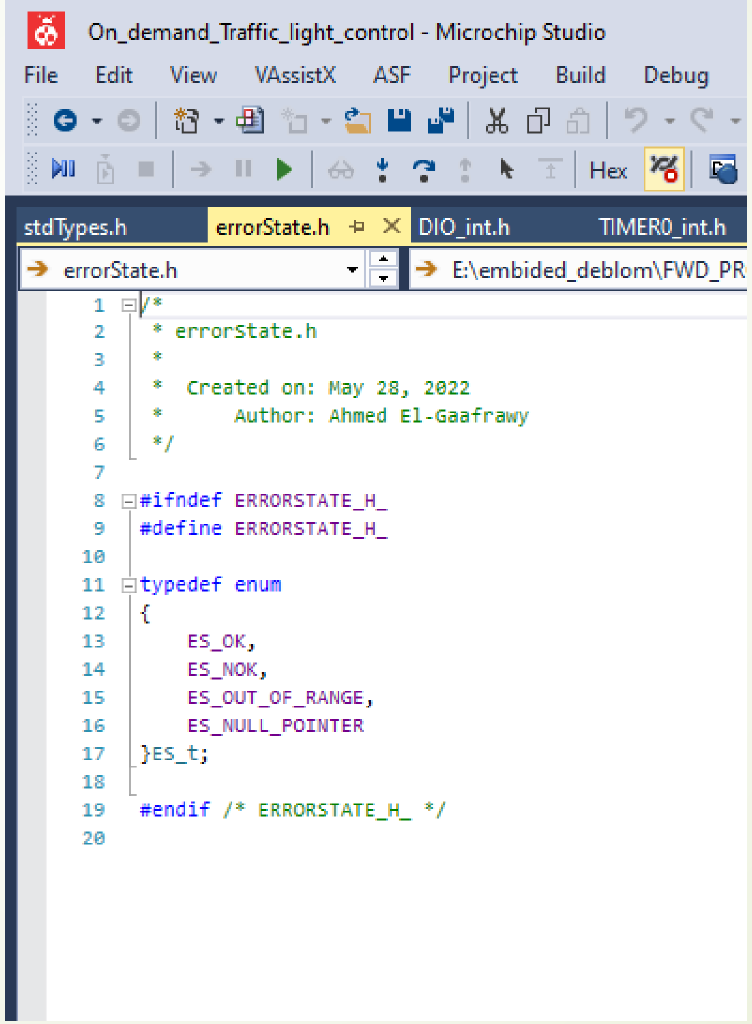


```c
#ifndef DIO_INT_H_
#define DIO_INT_H_

ES_t DIO_enuSetGroupDirection(u8 Copy_u8Group,u8 Copy_u8GroupDirection);
ES_t DIO_enuSetGroupValue(u8 Copy_u8Group,u8 Copy_u8GroupValue);
ES_t DIO_enuGetGroupValue(u8 Copy_u8Group,u8* Copy_pu8GroupValue);
ES_t DIO_enuSetPinDirection(u8 Copy_u8Group,u8 Copy_u8Pin,u8 Copy_u8Direction);
ES_t DIO_enuSetPinValue(u8 Copy_u8Group,u8 Copy_u8Pin,u8 Copy_u8Value);
ES_t DIO_enuGetPinValue(u8 Copy_u8Group,u8 Copy_u8Pin,u8* Copy_pu8Value);
ES_t DIO_enuTogglePin(u8 Copy_u8Group,u8 Copy_u8Pin);

#define DIO_GroupA              11
#define DIO_GroupB              12
#define DIO_GroupC              13
#define DIO_GroupD              14

#define DIO_PIN_0                   0
#define DIO_PIN_1                   1
#define DIO_PIN_2                   2
#define DIO_PIN_3                   3
#define DIO_PIN_4                   4
#define DIO_PIN_5                   5
#define DIO_PIN_6                   6
#define DIO_PIN_7                   7

#define DIO_OUTPUT                  1
#define DIO_INPUT                   0

#define DIO_HIGH                    1
#define DIO_LOW                     0

#endif /* DIO_INT_H_ */
```

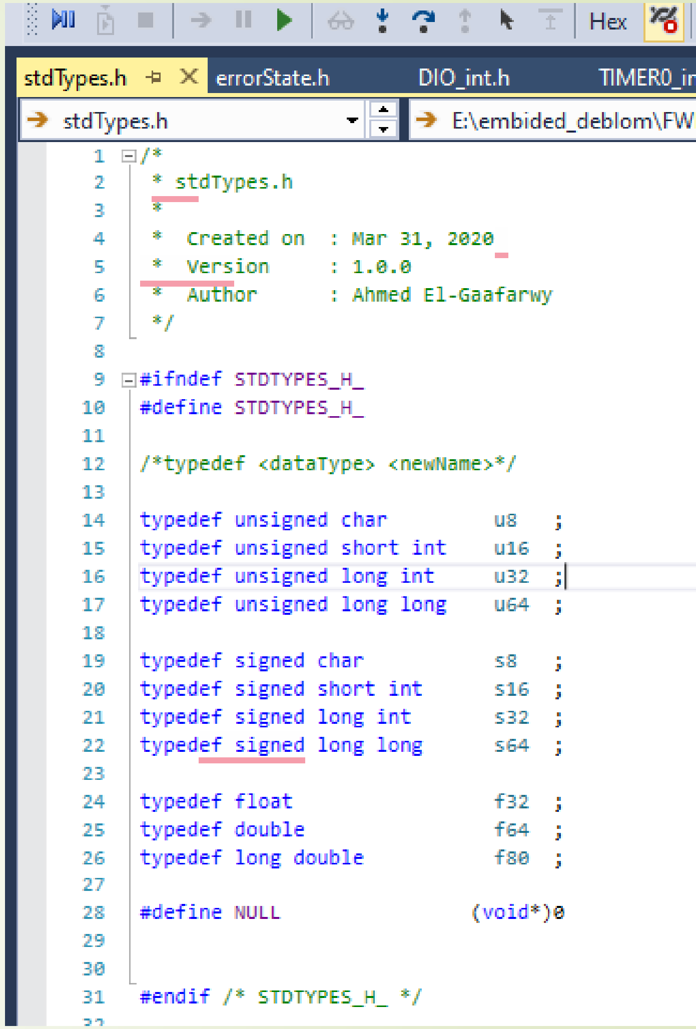
2.System design.



```c
#ifndef APP_H_
#define APP_H_

#define CARS_GREEN          DIO_GroupA,DIO_PIN_0
#define CARS_YELLOW         DIO_GroupA,DIO_PIN_1
#define CARS_RED            DIO_GroupA,DIO_PIN_2

#define PEDESTRAINS_GREEN   DIO_GroupB,DIO_PIN_0
#define PEDESTRAINS_YELLOW  DIO_GroupB,DIO_PIN_1
#define PEDESTRAINS_RED     DIO_GroupB,DIO_PIN_2

#define PEDESTRAINS_BUTTON  DIO_GroupD,DIO_PIN_2

#define PEDESTRAINS_INT_PIN EXTI_INT0

typedef enum{CARS_STOP,CARS_MOVE,CARS_STEADY}STATES;

#define ENABLED          1
#define DISABLED            0

#define ONE_SECOND_COUNTS      62                     //1 +   F_
#define TCNT_PRELOAD           (u8)274                    //256(

#define TCNT0       *(( volatile u8*)0X52)

void APP_vidInit(void);
void APP_vidStart(void);


#endif /* APP_H_ */
```

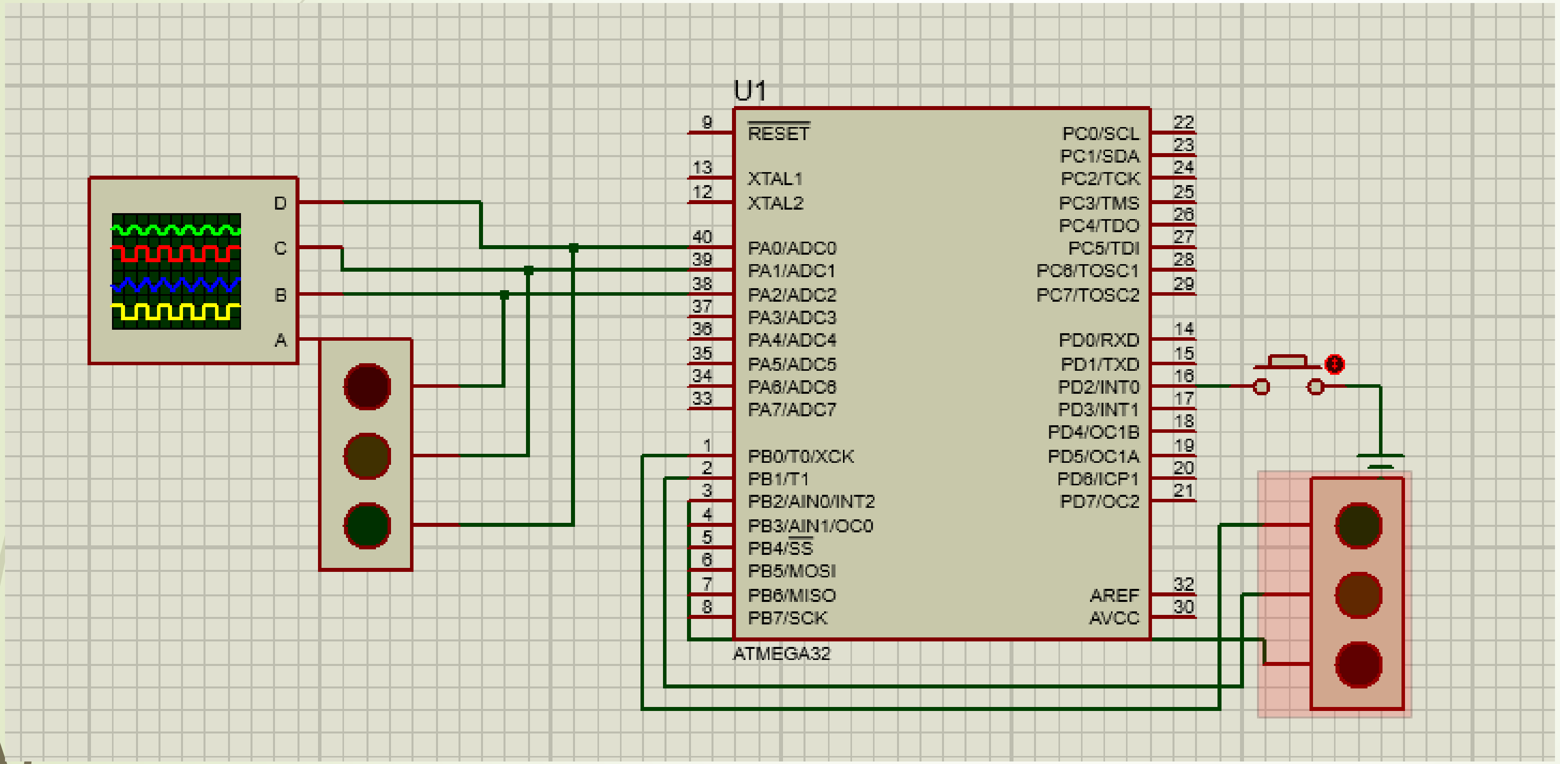


```c
/*
 * errorState.h
 *
 *  Created on: May 28, 2022
 *      Author: Ahmed El-Gaafrawy
 */

#ifndef ERRORSTATE_H_
#define ERRORSTATE_H_

typedef enum
{
    ES_OK,
    ES_NOK,
    ES_OUT_OF_RANGE,
    ES_NULL_POINTER
}ES_t;

#endif /* ERRORSTATE_H_ */
```

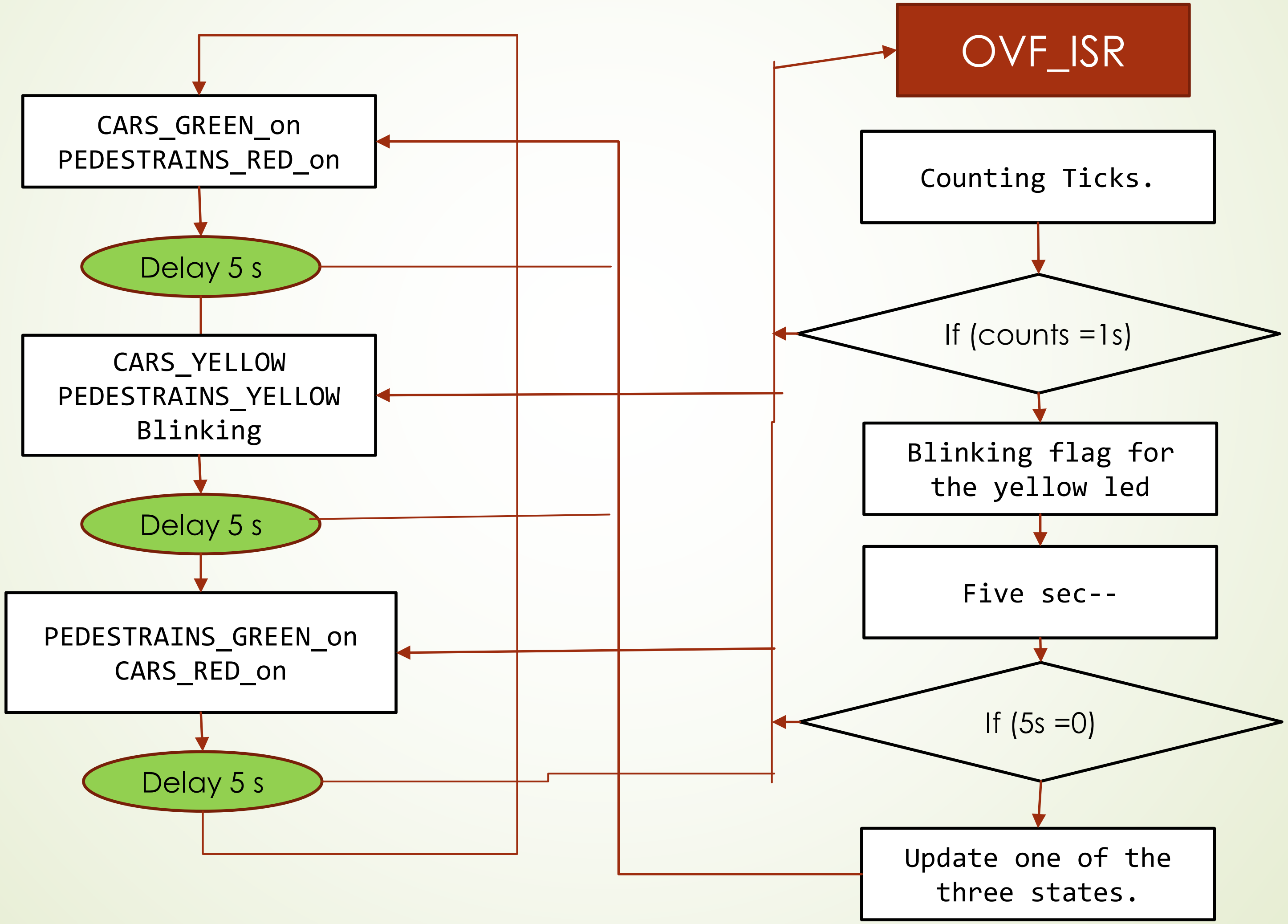


```c
/*
 * stdTypes.h
 *
 *  Created on  : Mar 31, 2020
 *  Version     : 1.0.0
 *  Author      : Ahmed El-Gaafarwy
 */

#ifndef STDTYPES_H_
#define STDTYPES_H_

/*typedef <dataType> <newName>*/

typedef unsigned char           u8   ;
typedef unsigned short int      u16  ;
typedef unsigned long int       u32  ;
typedef unsigned long long      u64  ;

typedef signed char             s8   ;
typedef signed short int        s16  ;
typedef signed long int         s32  ;
typedef signed long long        s64  ;

typedef float                   f32  ;
typedef double                  f64  ;
typedef long double             f80  ;

#define NULL                 (void*)0


#endif /* STDTYPES_H_ */
```

2.System design.



3.System flowchart.



CARS_GREEN_on PEDESTRAINS_RED_on

Delay 5 s

CARS_YELLOW PEDESTRAINS_YELLOW Blinking

Delay 5 s

PEDESTRAINS_GREEN_on CARS_RED_on

Delay 5 s

OVF_ISR

Counting Ticks.

If (counts =1s)

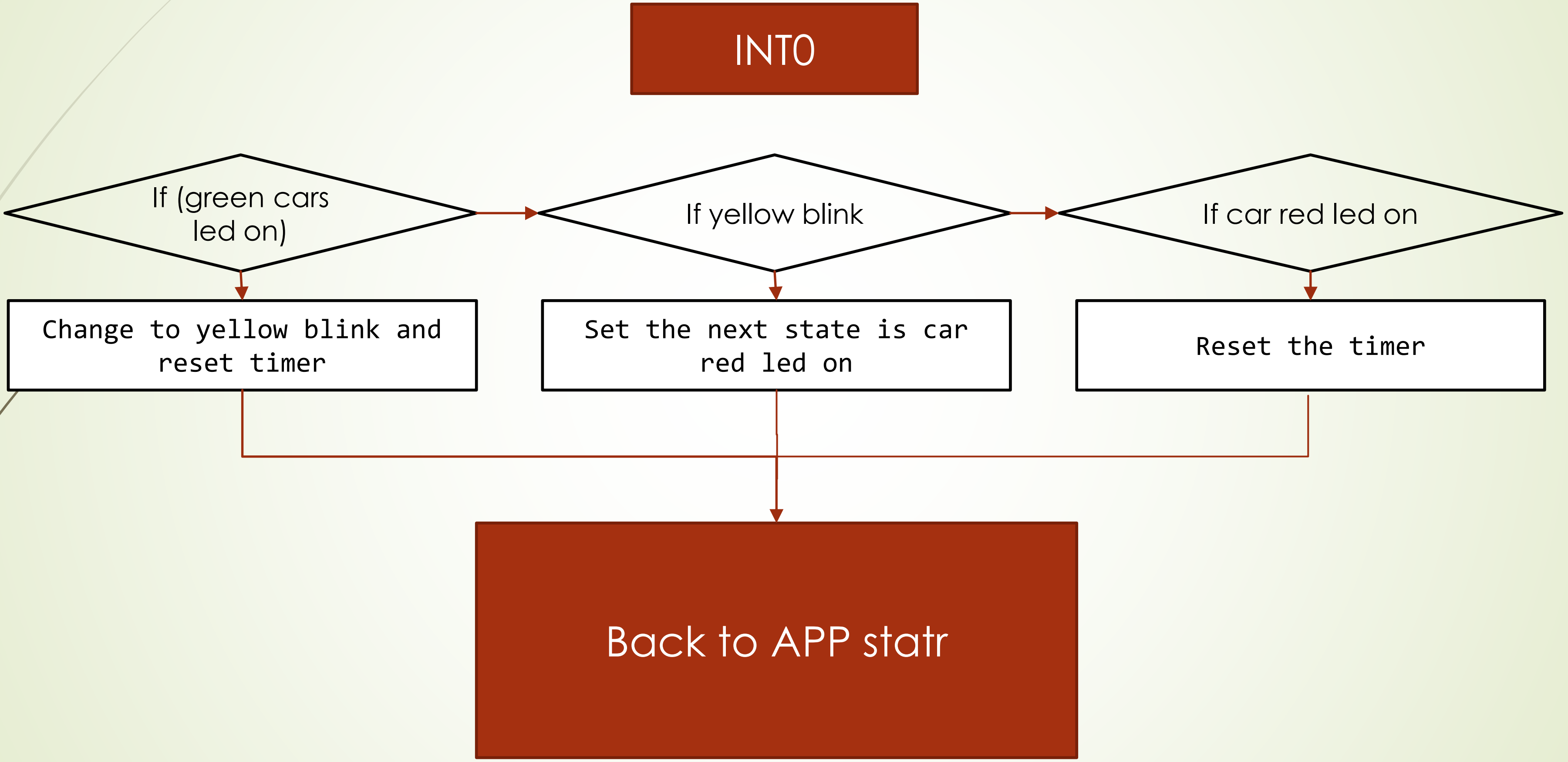
Blinking flag for the yellow led

Five sec--

If (5s =0)

Update one of the three states.

3.System flowchart.



4.System constraints .

- If the user holds the pedestrian button the system will never go into pedestrian mode.
- double click on the pedestrian button the first will operate all clicks that come after the first click will be neglected till the green light for the pedestrian is switched on for 3s.