

On-demand Traffic light control

-Content.

- 
- 1.System description.
 - 2.System design.
 - 3.System flowchart.



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.

1. 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)

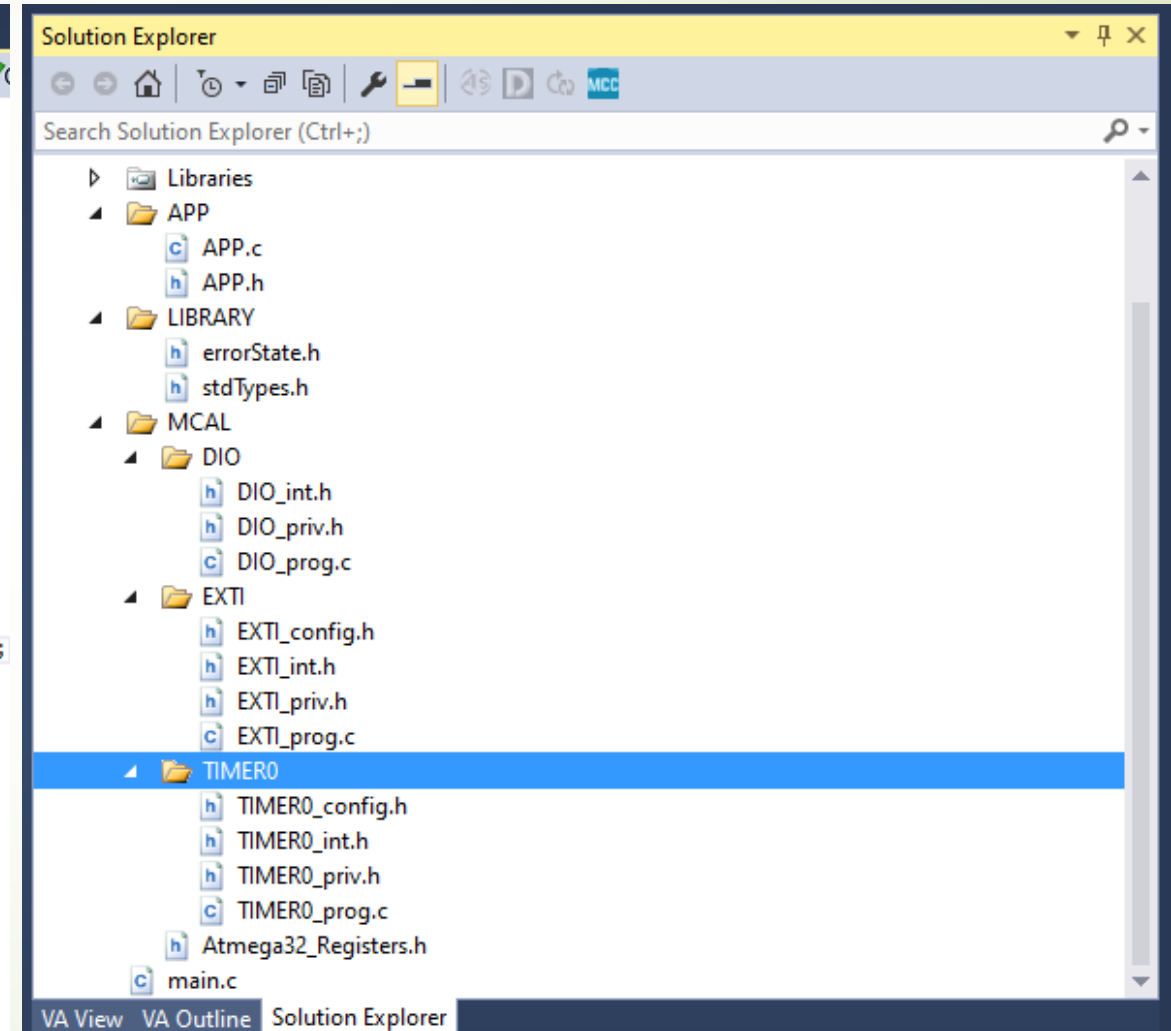
1. System design.

➤ 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.

1. System design.

```
TIMER0_int.h  X  TIMER0_prog.c  TIMER0_config.h  APP.c  APP.h
TIMER0_enuCallBackFun  ES_t TIMER0_enuCallBackFun(void (*Copy_pfunAPPFun)(void*),void * Copy_pvidParameter, u8 Copy_Interru...
4  * Created: 9/23/2022 1:27:06 PM
5  * Author: abano
6  */
7
8
9  #ifndef TIMER0_INT_H_
10 #define TIMER0_INT_H_
11
12 ES_t TIMER0_enuInit();
13 ES_t TIMER0_enuDelay_MS(u16 Copy_u16DelayMs);
14 ES_t TIMER0_enuDelay_US(u16 Copy_u16Delayus);
15 ES_t TIMER0_enuWrightPWMFastMood(u8 Copy_u8Duty);
16 ES_t TIMER0_enuWrightPWMPHaseCorrect(u8 Copy_u8Duty);
17 ES_t TIMER0_enuSetCTCValue(u8 Copy_CTCVvalue);
18 ES_t TIMER0_enuSetWaveMood(u8 Copy_u8WaveMood);
19 ES_t TIMER0_enuSetOutPinMood(u8 Copy_u8OutPinMood);
20 ES_t TIMER0_enuEnableInterrupt(u8 Copy_u8Interrupt);
21 ES_t TIMER0_enudisableInterrupt(u8 Copy_u8Interrupt);
22
23 ES_t TIMER0_enuCallBackFun(void (*Copy_pfunAPPFun)(void*),void * Copy_pvidParameter,u8 Copy_Interru...
24
25 #define OCIE0 1
26 #define TOIE0 0
27 #define TIMER0_OVFSInterr (1<<TOIE0)
28 #define TIMER0_CompMatchInterr (1<<OCIE0)
29 #ifndef GLOBALINTERRUPT
30 #define GLOBALINTERRUPT
31 #define GLOBALINTERRUPT_EN() asm("SEI")
32 #define GLOBALINTERRUPT_DI() asm("CLI")
33
34 #endif
35
36 #endif /* TIMER0_INT_H_ */
```



1. System design.

```
int.h  TIMER0_prog.c  TIMER0_config.h  APP.c  APP.h  EXTI_int.h
/*
 * EXTI_int.h
 *
 * Created: 9/30/2022 9:57:16 PM
 * Author: abano
 */

#ifndef EXTI_INT_H_
#define EXTI_INT_H_

#ifndef GLOBAL_INTERRUPT
#define GLOBAL_INTERRUPT
#define GLOBAL_INTERRUPT_EN() asm("SEI")
#define GLOBAL_INTERRUPT_DI() asm("CLI")
#endif

ES_t EXTI_enhInit(void);
ES_t EXTI_enhSetInterruptMood(u8 Copy_u8IntPin, u8 Copy_u8IntMood);
ES_t EXTI_enhEnableInterrupt(u8 Copy_u8IntPin);
ES_t EXTI_enhDisableInterrupt(u8 Copy_u8IntPin);
ES_t EXTI_enhCallBackFun(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_RISING 3

#endif /* EXTI_INT_H_ */
```

```
h.h  X  TIMER0_int.h  TIMER0_prog.c  TIMER0_config.h  APP.c  APP.h  EXTI_int.h
D_int.h  E:\embeded_deblom\FWD_PROFI\Interfacing\On_demand_Traffic_light_control\On_dema

9 #ifndef DIO_INT_H_
10 #define DIO_INT_H_
11
12 ES_t DIO_enhSetGroupDirection(u8 Copy_u8Group, u8 Copy_u8GroupDirection);
13 ES_t DIO_enhSetGroupValue(u8 Copy_u8Group, u8 Copy_u8GroupValue);
14 ES_t DIO_enhGetGroupValue(u8 Copy_u8Group, u8* Copy_pu8GroupValue);
15 ES_t DIO_enhSetPinDirection(u8 Copy_u8Group, u8 Copy_u8Pin, u8 Copy_u8Direction);
16 ES_t DIO_enhSetPinValue(u8 Copy_u8Group, u8 Copy_u8Pin, u8 Copy_u8Value);
17 ES_t DIO_enhGetPinValue(u8 Copy_u8Group, u8 Copy_u8Pin, u8* Copy_pu8Value);
18 ES_t DIO_enhTogglePin(u8 Copy_u8Group, u8 Copy_u8Pin);
19
20 #define DIO_GroupA 11
21 #define DIO_GroupB 12
22 #define DIO_GroupC 13
23 #define DIO_GroupD 14
24
25 #define DIO_PIN_0 0
26 #define DIO_PIN_1 1
27 #define DIO_PIN_2 2
28 #define DIO_PIN_3 3
29 #define DIO_PIN_4 4
30 #define DIO_PIN_5 5
31 #define DIO_PIN_6 6
32 #define DIO_PIN_7 7
33
34 #define DIO_OUTPUT 1
35 #define DIO_INPUT 0
36
37 #define DIO_HIGH 1
38 #define DIO_LOW 0
39
40 #endif /* DIO_INT_H_ */

output from:
Task List  Bookmarks  Error List
```


1. System design.

```
stdTypes.h  errorState.h  DIO_int.h  TIMER0_int.h  TIMER0_co

8
9 #ifndef APP_H_
10 #define APP_H_
11
12 #define CARS_GREEN      DIO_GroupA,DIO_PIN_0
13 #define CARS_YELLOW     DIO_GroupA,DIO_PIN_1
14 #define CARS_RED       DIO_GroupA,DIO_PIN_2
15
16 #define PEDESTRAINS_GREEN DIO_GroupB,DIO_PIN_0
17 #define PEDESTRAINS_YELLOW DIO_GroupB,DIO_PIN_1
18 #define PEDESTRAINS_RED  DIO_GroupB,DIO_PIN_2
19
20 #define PEDESTRAINS_BUTTON DIO_GroupD,DIO_PIN_2
21
22 #define PEDESTRAINS_INT_PIN EXTI_INT0
23
24 typedef enum{CARS_STOP,CARS_MOVE,CARS_STEADY}STATES;
25
26 #define ENABLED 1
27 #define DISABLED 0
28
29 #define ONE_SECOND_COUNTS 62 //1 + F_
30 #define TCNT_PRELOAD (u8)274 //256(
31
32 #define TCNT0 *(( volatile u8*)0X52)
33
34 void APP_vidInit(void);
35 void APP_vidStart(void);
36
37
38 #endif /* APP_H_ */
```

```
On_demand_Traffic_light_control - Microchip Studio
File Edit View VAssistX ASF Project Build Debug

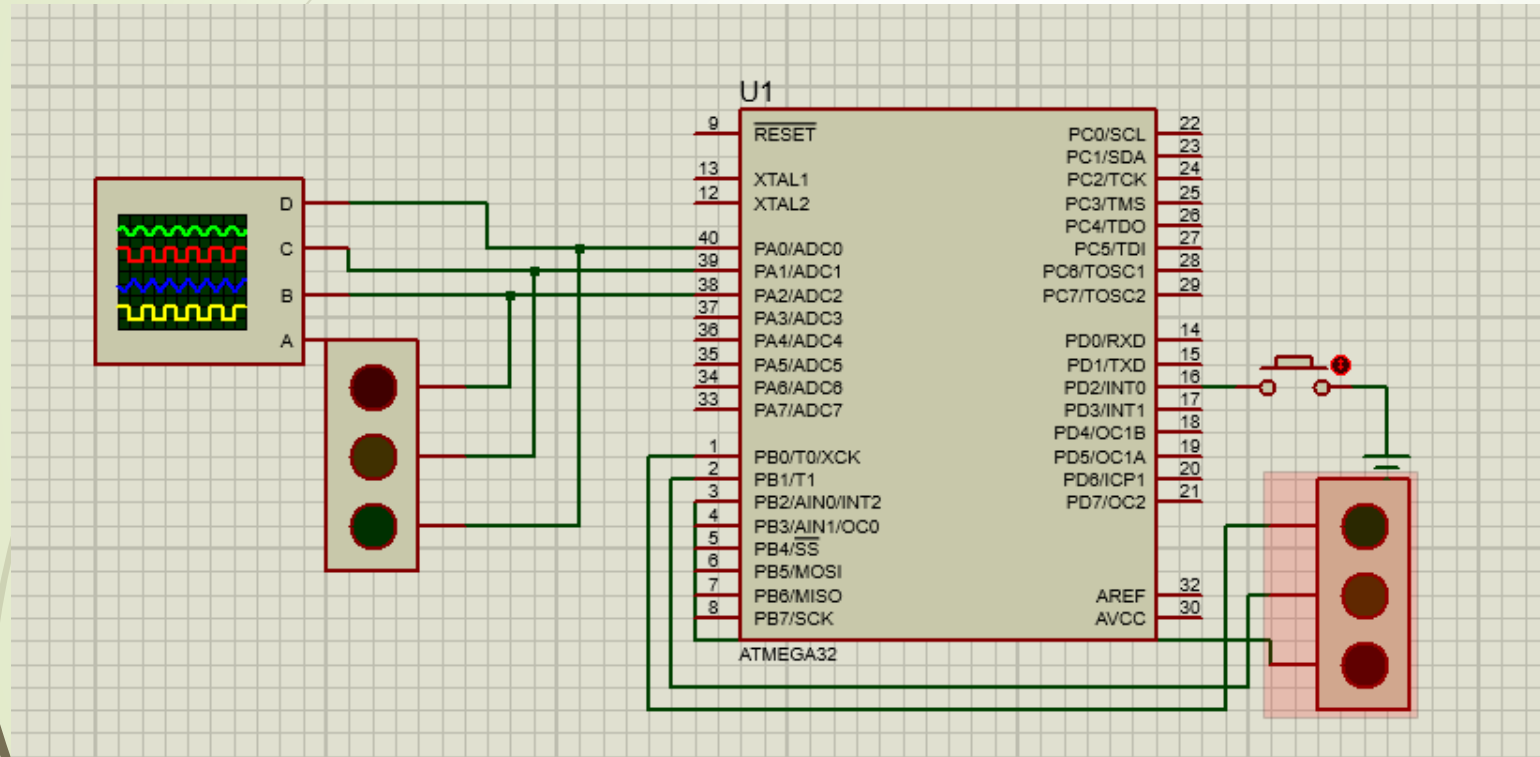
stdTypes.h  errorState.h  DIO_int.h  TIMER0_int.h
errorState.h  E:\embided_deblom\FWD_PR

1 /*
2  * errorState.h
3  *
4  * Created on: May 28, 2022
5  * Author: Ahmed El-Gaafrawy
6  */
7
8 #ifndef ERRORSTATE_H_
9 #define ERRORSTATE_H_
10
11 typedef enum
12 {
13     ES_OK,
14     ES_NOK,
15     ES_OUT_OF_RANGE,
16     ES_NULL_POINTER
17 }ES_t;
18
19 #endif /* ERRORSTATE_H_ */
20
```

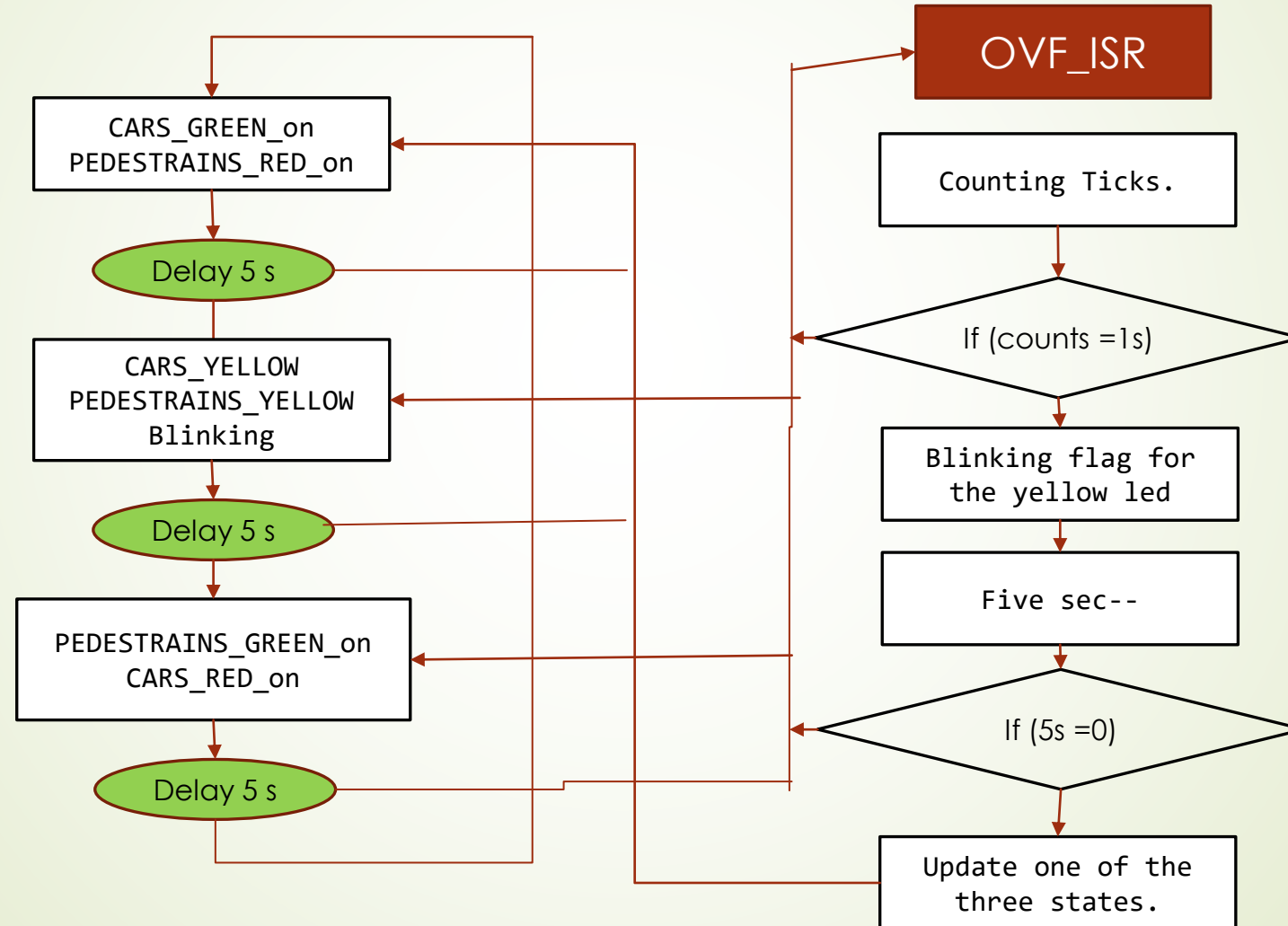
```
stdTypes.h  errorState.h  DIO_int.h  TIMER0_in
stdTypes.h  E:\embided_deblom\FW

1 /*
2  * stdTypes.h
3  *
4  * Created on : Mar 31, 2020
5  * Version : 1.0.0
6  * Author : Ahmed El-Gaafrawy
7  */
8
9 #ifndef STDYPES_H_
10 #define STDYPES_H_
11
12 /*typedef <dataType> <newName>*/
13
14 typedef unsigned char u8 ;
15 typedef unsigned short int u16 ;
16 typedef unsigned long int u32 ;
17 typedef unsigned long long u64 ;
18
19 typedef signed char s8 ;
20 typedef signed short int s16 ;
21 typedef signed long int s32 ;
22 typedef signed long long s64 ;
23
24 typedef float f32 ;
25 typedef double f64 ;
26 typedef long double f80 ;
27
28 #define NULL (void*)0
29
30
31 #endif /* STDYPES_H_ */
32
```

1. System design.



1. System flowchart.



1. System flowchart.

