**On Demand Traffic Light System**

**Project report**

**Name :** Mohamed Ashraf Elsayed Mahmoud

**Email add. :** mohamedashrafelsayed321@gmail.com

**August Cohort**

**System Description:**

The system is a traffic light with two modes:

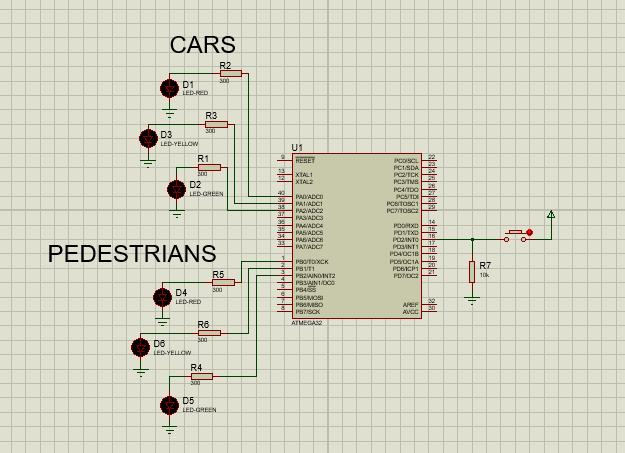
normal mode and pedestrian mode. Switching between the two modes is done using a push button. The system is controlled by microcontroller which is ATmega32a.

**NORMAL MODE:**

In normal mode , when the cars’ red light is on the pedestrians’ green light is also on . when cars’ yellow light blinks pedestrians’ yellow lights also blinks . when cars’ green light is on the pedestrians’ red light is also on .so if we considered cars it would go on a sequence of red/yellow/green/yellow/red…… and the pedestrians are its inverse except for the yellow color that is common between them.

**PEDESTRIAN MODE:**

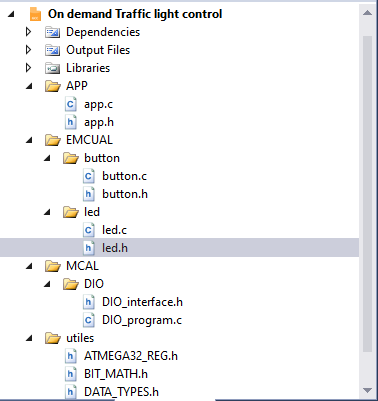
When switching to the pedestrian mode by pressing on the button, the system checks first which pair of light is on. If the cars’ red light is on and the pedestrian green light is on too. The system does nothing but if the cars’ green or yellow light is on the yellow light on both cars and pedestrians traffic blinks for five seconds then the cars’ red light will be on while pedestrians green light will be on.

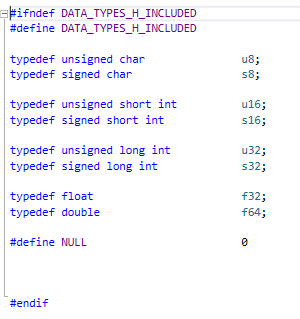


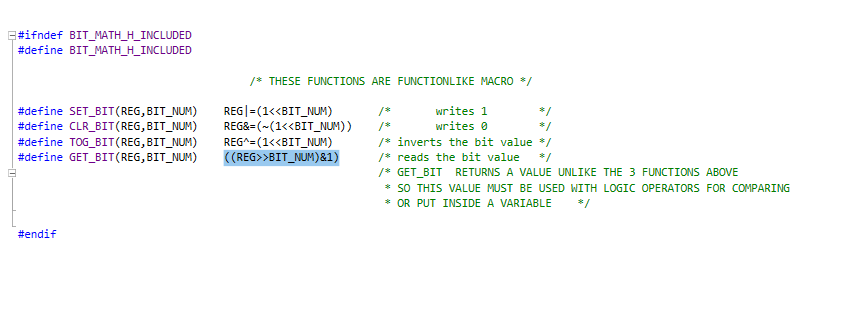
This is the simulation of the project on proteus

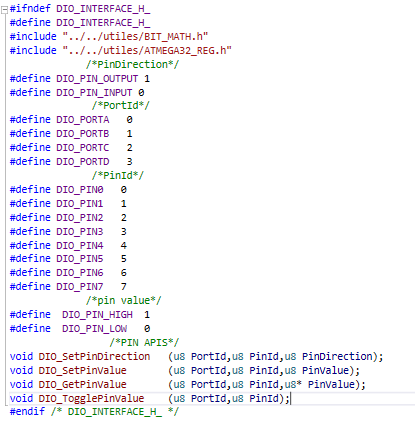
**System Design and Implementations:**

I used the static architecture and SOLID principles in creating and programming my project so I divided the whole project into layers like MCAL, ECUAL and APP in addition to a folder called utiles that contain 3 files which are (DATATYPES.H , BITMATH.H and ATMEGA32REG.h) that can be used in each and every layer and also can be used in any project concerning the use of atmega32 which is great for reusability.

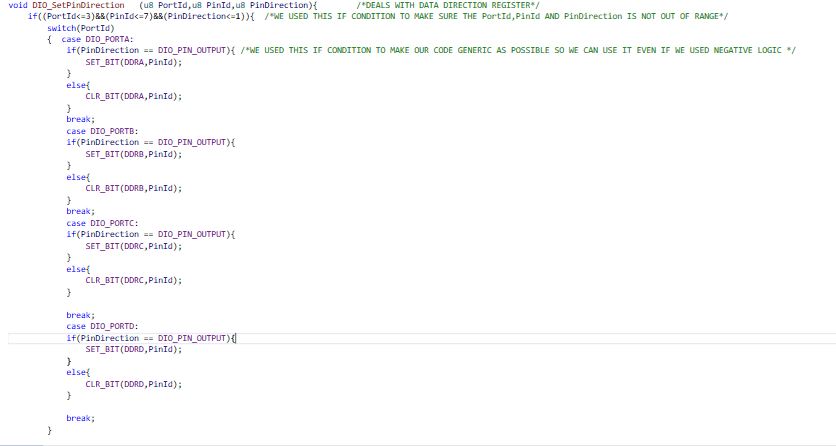
This is the solution explorer that shows the layers ,folders and files.

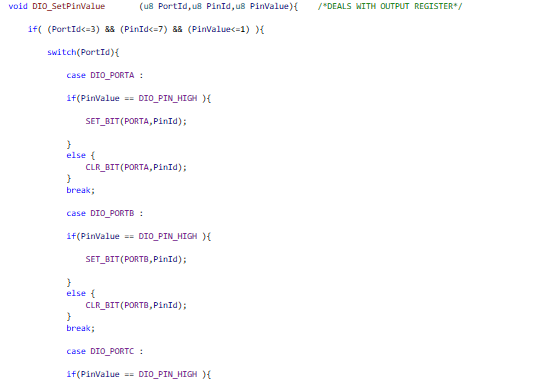
 This is the datatypes file that is inside the utiles folder and here I wrote all the typedefs

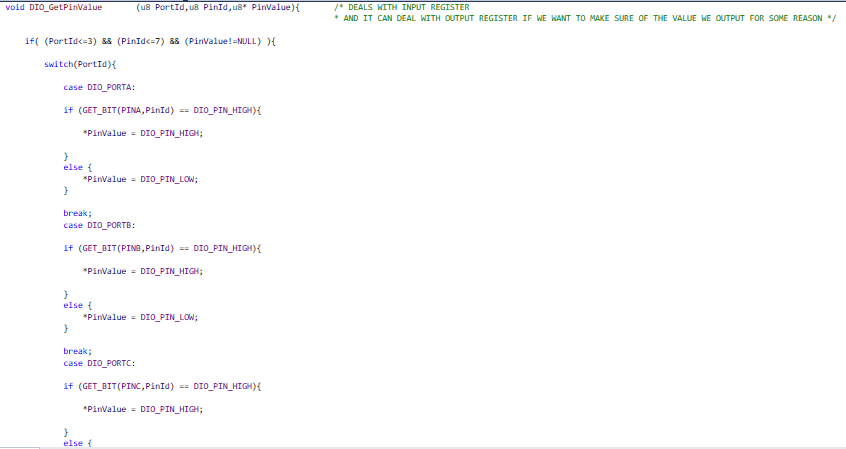
this is BIT\_MATH.H file which contains 4 function like macros that set, clear, toggle and read the value of the bit .

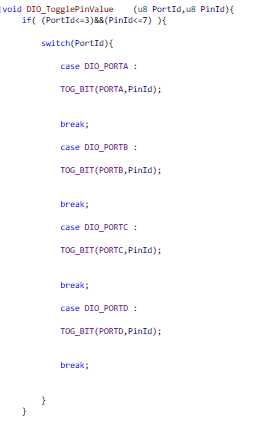


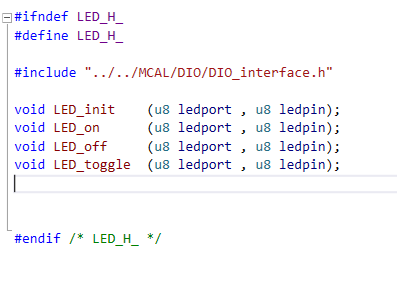
this is the dio header file where I defined macros like pin direction, port id, pin id and pin value in addition to the prototypes of 4 functions .

this is the first function in dio.c that sets the direction of the pin whether it is output or input and takes the port, pin and output or input .

this function sets the value of a bit to high or low according to the value passed to it.

this function reads the value of pin and stores it in the variable that its address is passed to the function.

this function toggles the pin that is passed to the function



This the led.h file which contains 4 functions that are implemented in the led.c file



This c file implements the 4 functions which are

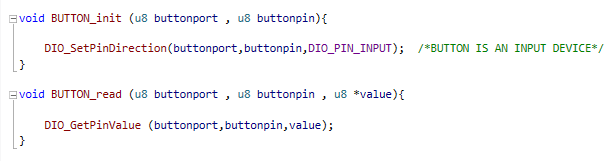
LED\_init sets the direction of the led to be out since it is an output device

LED\_on turns on the led

LED\_off turns off the led

LED\_toggle toggles the led

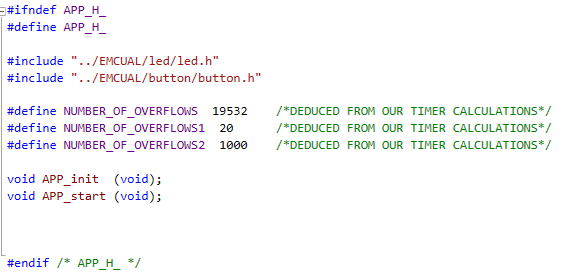
this file is button.h which contains two functions prototypes that are implemented in button.c file



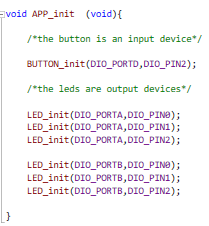
this is button.c which contains the implementations of two functions

BUTTON\_init it sets the button to be an input device

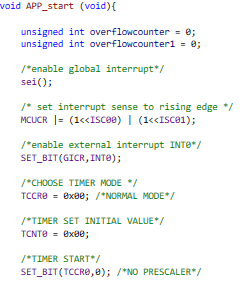
BUTTON\_read it reads the value of the button whether it is pressed or not



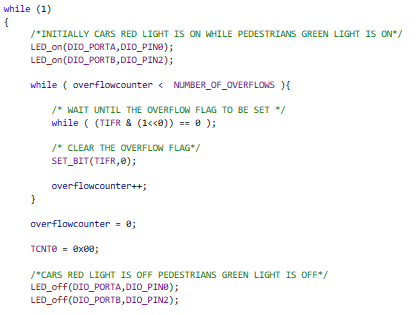
This file is app.h contains 2 functions which are APP\_init and APP\_start



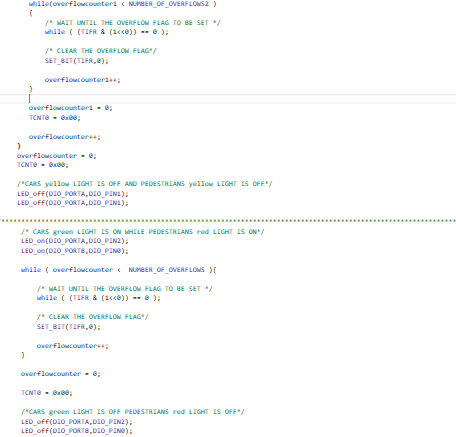
This is the implementation of the function APP\_init that sets the directions of the button to be input and the 6 leds to be output

this is the first part of the implementation of APP\_start function

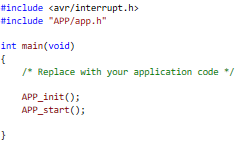
this part handles the interrupt and the timer

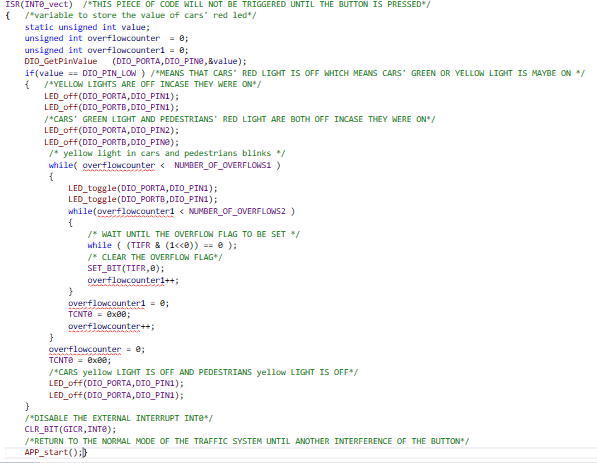


This part lights the cars’ red light on and pedestrians’ green light on



This part makes the yellow light of both blinks then after 5 seconds the cars’green light is on while the pedestrians’ red light is on

this is part of the main.c where we call the two functions in the app layer



this is the ISR which is related to the interrupts and is only triggered when the button is pressed and The function checks first for the state of pin 0 in port A (cars’ red light). If the red light is on nothing will happen. If the green or yellow light is on then the yellow light starts blinking for 5 seconds then the cars’ red light and pedestrians green light is turned on.

At the end the external interrupt is disabled to return to normal mode and the function app start is called in order for the traffic light system to work normally.

The cars’ red light and pedestrians’ green light are initially on.

delay 5 sec.

Both the cars’ and pedestrians’ yellow light starts blinking

Both the cars’ and pedestrians’ yellow light starts blinking

Both the cars’ and pedestrians’ yellow light starts blinking

The cars’ red light and pedestrians’ green light is on..

The cars’ green light and pedestrians’ red light is on.

delay 5 sec.

Is the button pressed

delay 5 sec. delay 5 sec.