Embedded Systems Professional Track Nano-Degree

EGYPT – FWD

On-demand Traffic light control

Author: Nour Alaa Eldin Hanafy Moharram

Project over-view:

In our cities there are a lot of traffic light and pedestrian lights that manage the process of moving of people and cars to prevent accidents and critical problems that may occur. In this project I have implemented a program to maintain these process by using six LEDS three for the traffic light (RED,GREEN and YELLOW) and three for pedestrian light (RED,GREEN and YELLOW) and a push button to allow the pedestrian to stop the passing cars and be able to cross the road safely.

Part 1: System Architecture:

Micro-controller Abstraction layer (MCAL):

- DIO Peripheral:

Used to access the general purpose input / output pins and determine which pins to be input to be used for push button and which pins to be output to be used by LEDS to give high output to light

- External interrupt enable peripheral:

Used to activate interrupt for the required pin connected with the push button so when it is pressed it will interrupt the processor to go and execute the Interrupt service routine function (ISR)

- General interrupt Enable peripheral:

Used to enable the interrupts in the system to be able to give interrupt when some occasion occurred

- Timer peripheral

By initializing the Timer peripheral and selecting the required mode (Normal mode) from the (TCCR0) register and initializing the counter register (TCNT0)

And then selecting the required PRESCALLER according to our calculations in order to determine the timer frequency, timer tick time and timer maximum delay time.

Hardware Abstraction Layer (HAL):

- LED Driver

Used to implement LED modes for traffic and pedestrian lights

- BUTTON Driver

Used to initialize the pin to be input to receive signal from output source PUSH BUTTON

Common layer:

- BIT WISE operators :

Used to sit, clear, toggle and get status of pins.

- Type definitions:

Used to implement type definitions for data types in our system.

APPLICATION LAYER:

Used to implement the program sequence according to the requirments and it has access to all the APIS in the Lower layers.

Main.c:

Used to call all the APIS in the Lower layers and Start the program.

Timer Calculations:

1-For 5 seconds delay:

- System frequency = 1MHz
- Tick time = PRESCALLER / system frequency = 1024/1000000=1.024 msec
- Timer maximum delay = 2^8 * Tick time = 262.144 msec
- Since Tdelay > Timer maximum delay “5000 msec > 262.144msec”
- Number of over flows = ceil(Tdelay/Timer maximum delay)= 20
- Timer initial value = 2^8 – (Number of overflows/20)= 255;

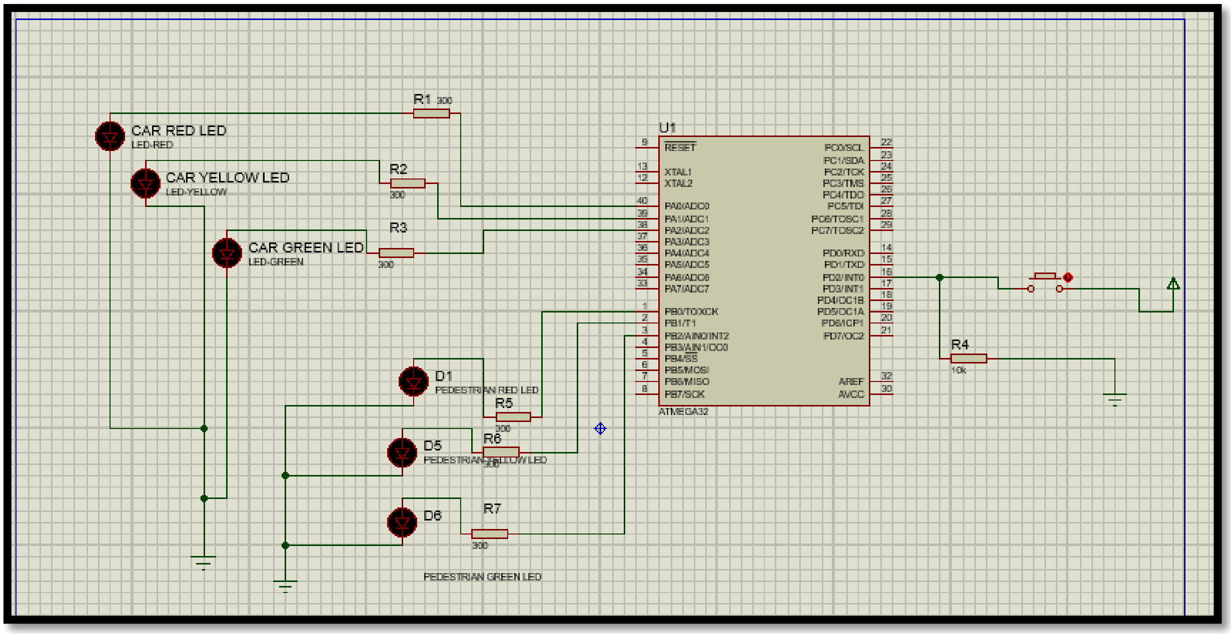
2- For 1 second delay:

- System frequency = 1MHz
- Tick time = PRESCALLER / system frequency = 1024/1000000=1.024 msec
- Timer maximum delay = 2^8 * Tick time = 262.144 msec
- Since Tdelay > Timer maximum delay “1000 msec > 262.144msec”
- Number of over flows = ceil(Tdelay/Timer maximum delay)= 4
- Timer initial value = 2^8 – (Number of overflows/4)= 255;

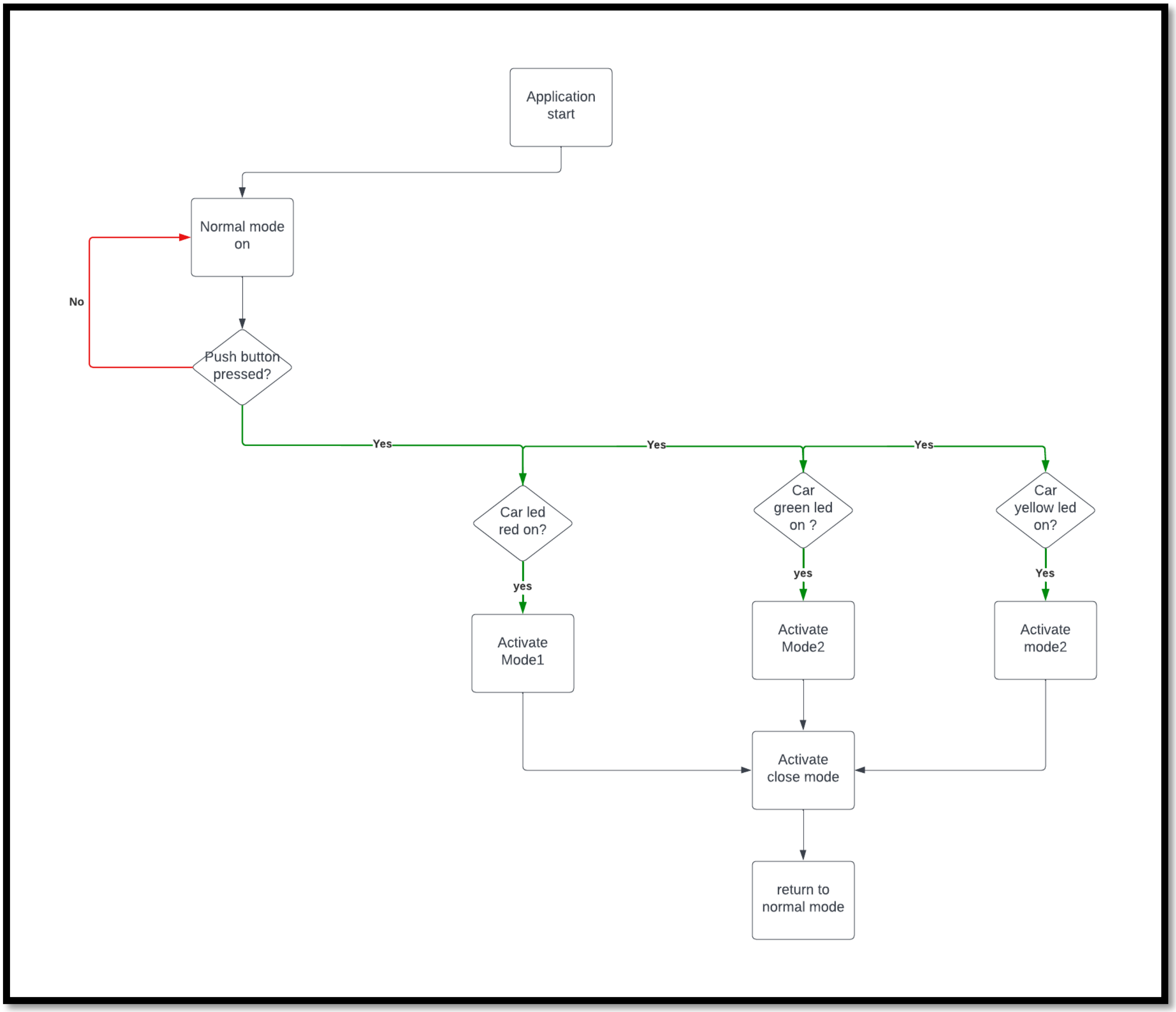
System Electric Design:

- 6 LEDS
- 6 resistors of 300 ohm
- 1 resistor of 10K ohm for Pull-Down
- 1 Push button
- 1 Micro-controller Atmega32 with Frequency 1MHz

System Simulation:



System Flow chart:


Application start
Normal mode on
No
Push button pressed?
Yes
Yes
Yes
Car led red on?
Car green led on ?
Car yellow led on?
yes
yes
Yes
Activate Mode1
Activate Mode2
Activate mode2
Activate close mode
return to normal mode