

Embedded Systems Professional Track

EgFWD - Udacity

On demand Traffic Lights Control Project

Presented by

Andrew Ashraf Waheed Guerguis

November Cohort

Contents

System Description	3
System Design	3
System requirements	3
Hardware requirements	3
Software requirements	3
System constrains	4
Software layers	4
Flowchart	5

System Description

Traffic signals consist of three colors: Red, Yellow, and Green. When the signal turns red, you must stop, when it turns yellow, slow down and wait, and when it turns green, go. While the red signal for cars is on the green signal for pedestrians is on too and vice versa.

We implemented a system that makes it easier for the pedestrians to cross the road. As our system, consists of a button that people crossing the road can use it to force the traffic lights to turn on the red signal for cars and the green one for pedestrians.

System Design

System requirements

Hardware requirements

1. ATmega32 microcontroller
2. One push button connected to INTO pin for pedestrian
3. Three LEDs for cars - Green, Yellow, and Red, connected on port A, pins 0, 1, and 2
4. Three LEDs for pedestrians - Green, Yellow, and Red, connected on port B, pins 0, 1, and 2

Software requirements

In normal mode:

1. Cars' LEDs will be changed every five seconds starting from Green then yellow then red then yellow then Green.
2. The Yellow LED will blink for five seconds before moving to Green or Red LEDs.

In pedestrian mode:

1. Change from normal mode to pedestrian mode when the pedestrian button is pressed.
2. If pressed when the cars' Red LED is on, the pedestrian's Green LED and the cars' Red LEDs will be on for five seconds, this means that pedestrians can cross the street while the pedestrian's Green LED is on.
3. If pressed when the cars' Green LED is on or the cars' Yellow LED is blinking, the pedestrian Red LED will be on then both Yellow LEDs start to blink for five seconds, then the cars' Red LED and pedestrian Green LEDs are on for five seconds, this means that pedestrian must wait until the Green LED is on.
4. At the end of the two states, the cars' Red LED will be off and both Yellow LEDs start blinking for 5 seconds and the pedestrian's Green LED is still on.
5. After the five seconds the pedestrian Green LED will be off and both the pedestrian Red LED and the cars' Green LED will be on.
6. Traffic lights signals are going to the normal mode again.

System constrains

When designing the system, we had many constrains that the system should follow.

1. In normal mode, the green LED must be turned on for 5 seconds, then the yellow LED should blink for another 5 seconds, then both must turn off and then the red one turns on, and vice versa for the pedestrians LEDs.
2. the second thing is that when the pedestrian's button is pressed, the LEDs change their sequence based on the requirements, as we have 3 requirements when the pedestrians' mode is pressed that could be found in the second point in the system requirements section.
3. The final constrain is about the button. As if a pedestrian pressed the button for a long time, nothing unexpected should happen and also if the button is double pressed the first press only should take action and nothing else for the second press.

Software layers

Application Layer				Utilities
ECUAL	LED driver	Delay Driver	Button Driver	
MCAL	Timer	DIO	Interrupts	
Microcontrollers				

We used an Atmega32 as our controller, 6 LEDS to mimic the traffic lights for both cars and pedestrians, and a push button to control the pedestrians' mode.

As you can see in the table on top. We implemented our systems as layers. Each layer consists of different modules to make easy for us to modify the system any time and in an easy way.

Flowchart



**FIGURE 1:
NORMAL MODE**

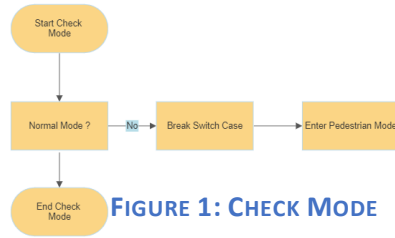
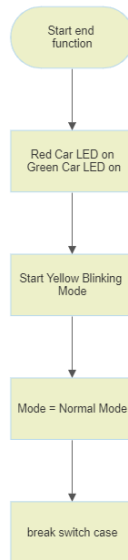


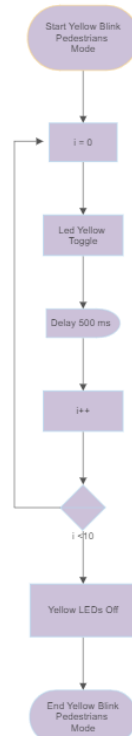
FIGURE 1: CHECK MODE



**FIGURE 3: BLINK
YELLOW NORMAL MODE**



**FIGURE 4: END
FUNCTION**



**FIGURE 5: YELLOW BLINKING PEDESTRIANS
MODE**

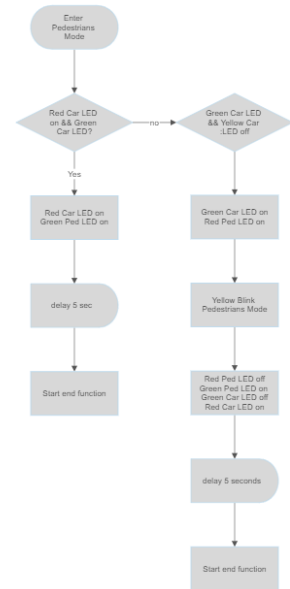


FIGURE 6: PEDESTRIANS MODE