**ON DEMAND**

**TRAFFIC LIGHTS SYSTEM**

# **Embedded System Professional Track**

# **EgFWD – Udacity**

# Project Documentation

***By: MennaAllah Gamal Mohamed***

# Simulation on Proteus

# Diagram, schematic Description automatically generated

# System Description

The system can detect when the button is pressed. Afterwards, based on current state it would decide what to do. It allows pedestrians to walk by making sure cars are stopped first.

# 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 Design

* *Hardware Design:*

• AVR Atmega32 (1MHz)

• 2 Green LEDs

• 2 Yellow LEDs

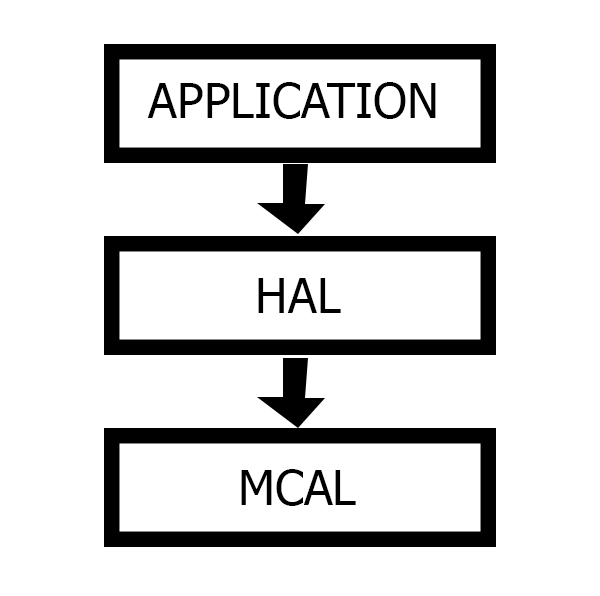
• 2 Red LEDs

• 6 (300) Ohm resistors

• 1 10k Ohm resistor

• 1 Push Button

* *Software Design:*

1. **MCAL:** Microcontroller Abstraction Layer

* Enables direct access to on-chip MCU peripheral modules and makes the upper software layer independent of the MCU.
* DIO Driver, Timer Driver, External Interrupt Driver.

1. **HAL:** Hardware Abstraction Layer

* Serves as an abstraction layer between a computer's physical hardware and its software. It provides a device driver interface allowing a program to communicate with the hardware.
* LED Driver, Button Driver.

1. **APP:** Application Layer

* It’s an application-specific so that it only provides support for a specific functionality
* *Solution Explorer*

*Graphical user interface, text, application

Description automatically generated*

# System Flowchart

Graphical user interface, application

Description automatically generated

A picture containing text

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

# System Constraints

* 1. ***Short Press:***

If short press is made on the crosswalk button, The system changes from Normal mode to Pedestrian mode.

* 1. ***Long Press:***

If long press is made on the crosswalk button, nothing should be done.

* 1. ***Double Press***

If double press is made on the crosswalk button,the first press will do the action and changes from normal mode to pedestrian mode. Nothing to be done after the second press.