

Egypt FWD – Udacity
Embedded systems professional track

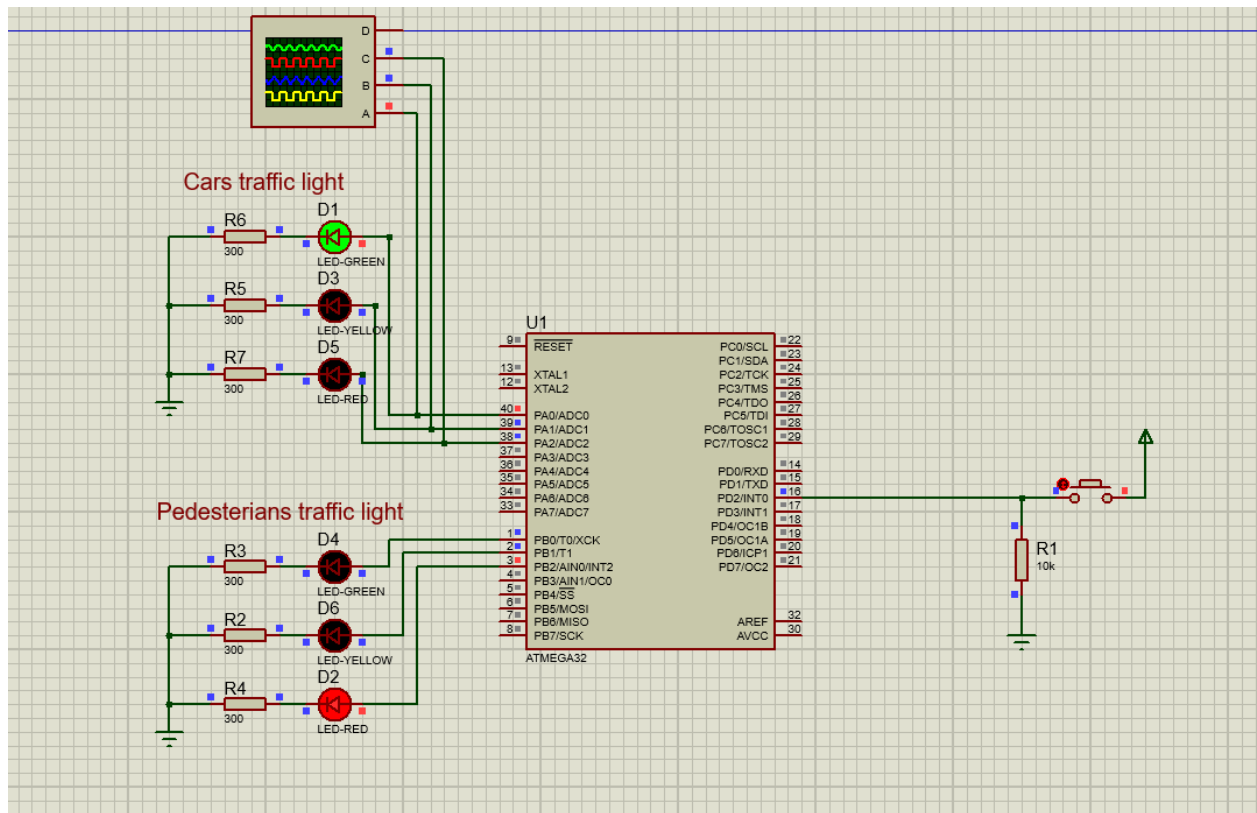
On-demand Traffic Light Control

Project documentation
By/ Mostafa Ayyad

Contents:

1. System description
 - a. About the project
 - b. Normal mode
 - c. Pedestrian mode
 - d. System components
 - Hardware requirements
 - Software requirements
2. Project design
 - a. System flowchart
 - b. System layers

1. System description



a. About the project:

This is an on-demand smart traffic light system. The main objective is to decrease waiting time for pedestrians as they have higher priority. The system has 2 modes: Normal mode, and pedestrian mode. In normal mode, the cars traffic light operates normally, and pedestrian traffic light are initially on red light on. Until the button is pressed, the system switches to pedestrian mode to allow pedestrians to cross the road safely and when the cycle is finished, the system switches back to normal mode.

b. Normal mode:

- 1) (Phase_1) This mode starts with cars green light initially on and pedestrian red light (always) on.
- 2) (Phase_2) After 5 seconds, cars yellow light starts blinking for another 5 seconds while cars green light is still on.
- 3) (Phase_3) After the 5 seconds, cars green and yellow lights are turned off then cars red light is turned on for 5 seconds.
- 4) (Phase_4) The last phase, the yellow light starts blinking again for 5 seconds with cars red light is still on.
- 5) At last, the system goes back to phase_1 again.

c. Pedestrian mode:

There is a special response for the pedestrian mode for every phase, depending on when the button is clicked.

- 1- If pressed in (**phase_1**) it skips green light remaining wait time and turns on car blinking yellow light, while cars green light is still on.
 - 2- If pressed in (**phase_2**) it resets cars blinking yellow light remaining wait time and turns on pedestrians blinking yellow light.
 - 3- if pressed in (**phase_3**) it resets cars red light remaining wait time, turns pedestrians green light on and red light off.
 - 4- if pressed in (**phase_4**) it skips cars blinking yellow light remaining wait time and returns back to phase_2.
- Then, it turns back the program to (**phase_2**), reversing the flow of the cars light from (yellow to green) to (yellow to red) again.

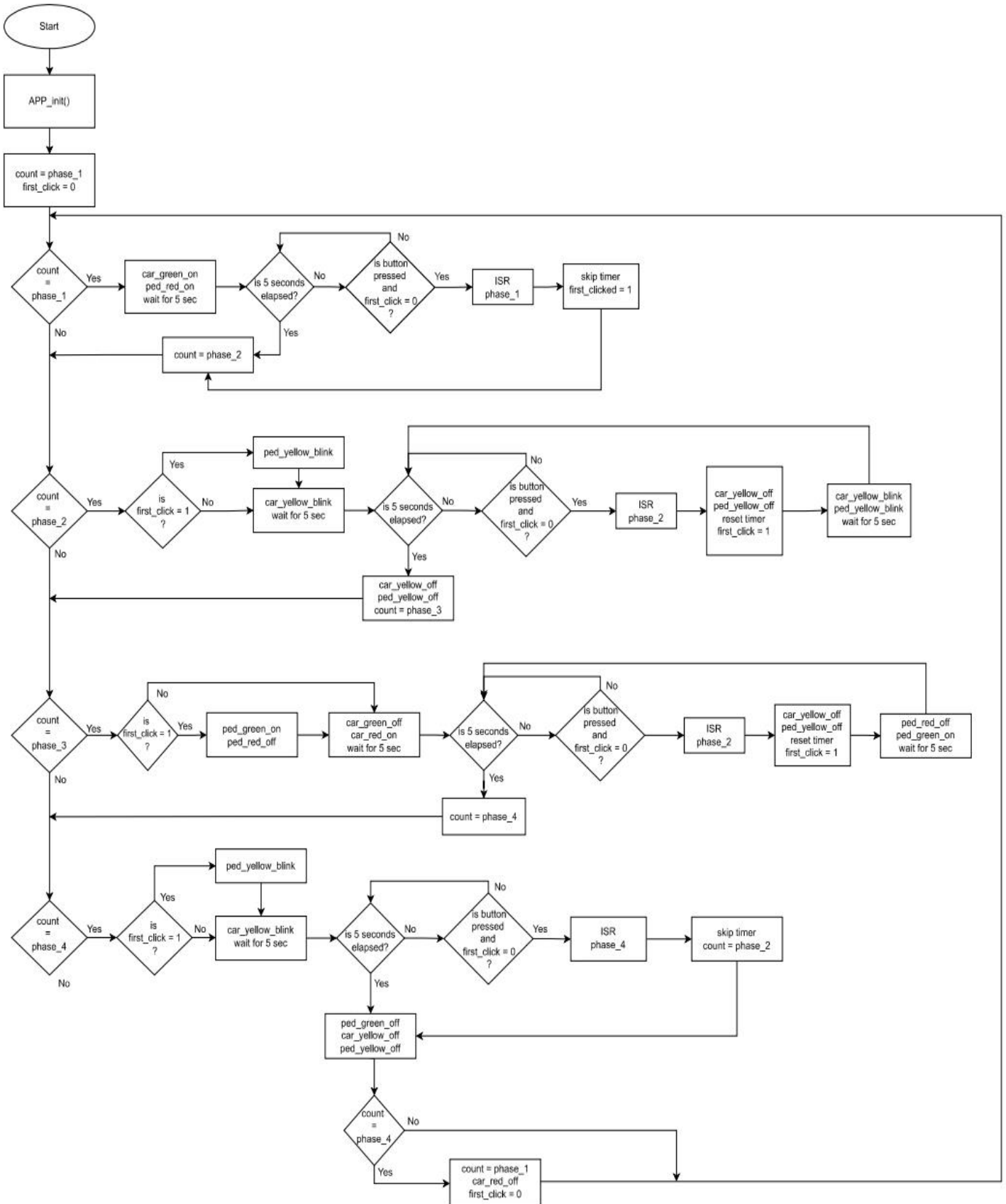
d. System requirements

- Hardware requirements
 - 1- AVR atmega32 microcontroller.
 - 2- 2 red LEDs, 2 yellow LEDs, 2 green LEDs.
 - 3- 6 (300 ohm) resistors.
 - 4- 1 (10K ohm) resistor. (pull-down resistor)
 - 5- 1 push button

- Software requirements
 - 1- Microchip studio IDE
 - 2- Proteus 8 simulator
 - 3- Draw.io (diagram designs)

2. Project design

a. System flowchart



b. System layers

