Smart Traffic Lights wit IoT

Kerem Atasoy
katasoy1@binghamton.edu
Department of Computer Science
Binghamton University

ABSTRACT

The main objective of this project is to create motion based and smarter traffic control environment. The smart street light system automatically switches the Leds based on the motion PIR sensor. Also, traffic lights can be switched from IOS application. Earlier version of the project I was able to create smart led but with this version I implied functionalities of traffic lights. In the prototype for a traffic light. The red led is turned on when there is no motion detected. When there is motion detected by the PIR sensor. The red light turn on until 3 seconds and system start to count down from 3 seconds. After 3 seconds red light turn off and yellow lights turn on1 second. After that yellow lights turn off and Green lights turn on 5 seconds. Then system turns off the green light and turns on the red light until motion detect.

KEYWORDS

PIR Sensor, LED, Internet of Things, Raspberry Pi, Blynk

INTRODUCTION

In a daily life we are all using traffic lights. Sometimes when crossing the street, sometimes while driving. This project of Smart Traffic Light can provide safer and advanced system for us. It can be used wherever there are regulations regarding traffic. Thanks to the IOT feature in this system, authorized public institutions such as the Police Department can stop or direct the entire traffic flow remotely. On some special days, it may be necessary to direct the pedestrian flow and vehicle flow to alternative routes or to distribute the density that may occur. With the possibilities provided by IoT, it is possible to do this process more effectively from a single center.

Thanks to the motion-based operation feature, which is one of the features of this system, pedestrians can confront safely within 8 seconds without touching any button. One of the benefits of being action-based is that it benefits the public health by reducing contact these days when there is a Corona virus epidemic. Because the density in traffic lights is very high at the beginning and end of business hours.

1. DESIGN ANDIMPLEMENTATION

In this part, I want to show the design of my Project which consist of Control Flow diagram and Data Flow diagram.

Then I can show the tools and the hardware that I used in my project with software implementation.

1.1 Control-Flow Diagram

The Control Flow Diagram shows the overall control of my project. It described in below steps:

- Program is booted in the Terminal of Raspberry Pi
- Motion is detected by the PIR
- First turn on the red LED for safety and count from 5.
- Second turn off the red LED and turn on the yellow.
- Then turn off the Yellow and turn on the Green for 5 seconds
- If motion is detected, then the red LED is ON.

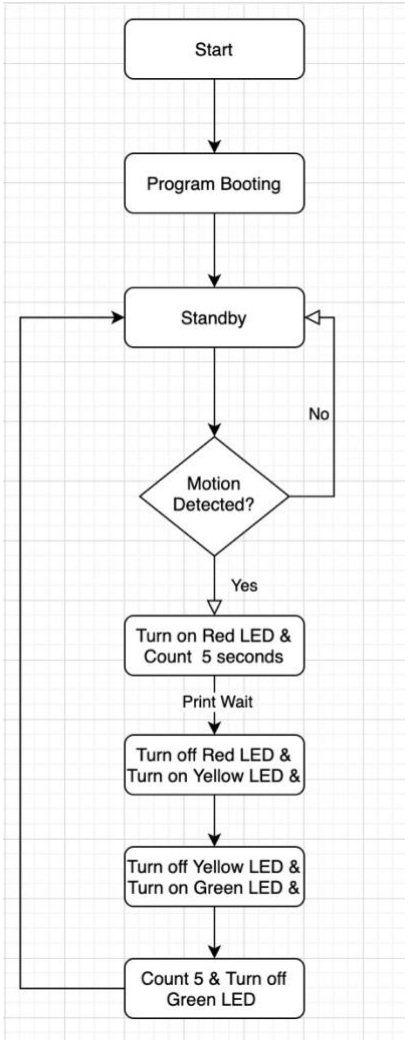
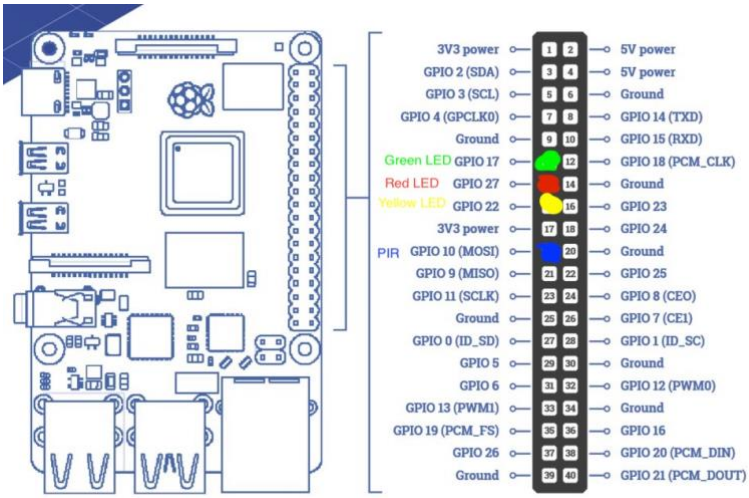


Figure 2: Dataflow diagram

1.2 Data-Flow Diagram

The data flow diagram is described below- The Green LED is connected to digital pin no 17 through a resistor to avoid excess flow of current through the circuit. The Red LED is connected to digital pin no 27 through a resistor to avoid excess flow of current through the circuit . The Yellow LED is connected to digital pin no 22 through a resistor to avoid excess flow of current through the circuit. The PIR is connected to digital pin no 10.



2. HARWDARE COMPONENTS

Figure 3.1 PIR Sensor

It is used to detect heat and motion of an object. It works entirely by detection of infrared radiation emitted or reflected by objects. Passive Infrared Sensor (PIR) refers to the fact that they do not radiate energy for detection purposes

Figure 3.2 Raspberry Pi

Raspberry Pi:

1. 4 USB ports
2. CPU : Quad-core 64 -bit ARM Cortex A53 clocked at 1.2 GHz
3. GPU: 400 MHz Video Core IV multimedia
4. Power source: 5V via MicroUSB or GPIO Header
5. Network: 10/100Mbps Ethernet and 802.11n Wireless LAN
6. It is programmable with Raspberry Pi IDE, Python

4. SOFTWARE REQUIREMENTS

I used following software for our project:

1. Raspberry Pi IDE for initialize Raspberry Pi
2. Python for running and debugging code on Raspberry Pi
3. Blynk libraries
4. Terminal for SSH connection

5. BLYNK PLATFORM

Blynk is software program that permits you to persistently construct interfaces for controlling and observing your equipment projects from your iOS and Android device. For using Blynk you should download it on your phone from App store or Play Store. You can create your project with controllers, displays, notifications, interfaces and some built-in sensors. Using the buttons, you can turn pins on and off or display data from sensors.

Blynk is perfect match for the Internet of Things. It can handle equipment distantly, it can show sensor information, it can store information, imagine it and do numerous other cool things.

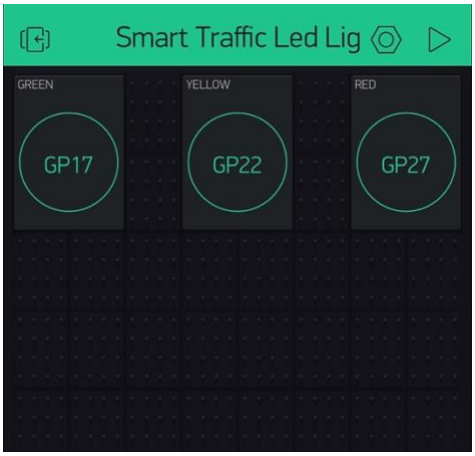
There are three major components in the platform:

- **Blynk App** - permits to you make useful interfaces for your activities utilizing different gadgets I give.

- **Blynk Server** - liable for every one of the

interchanges between the cell phone and equipment. You can utilize our Blynk Cloud or run your private Blynk worker locally. It is open source, could undoubtedly deal with a great many gadgets and can even be dispatched on an Arduino.

- **Blynk Libraries** –
  With Blynk Library you can associate more than 400 equipment models (counting ESP8266, ESP32, NodeMCU, all Arduinos, Raspberry Pi, Particle, Texas Instruments, etc.)to the Blynk Cloud. for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands



CONCLUSION

The important aim of this project is to create safer and smart traffic lights. This example project can be adapted to real life examples when necessary. It provides solutions wherever traffic lights are used and at the same time allows remote control of both vehicle and pedestrian traffic by changing the software.

REFERENCES

1. https://blynk.io/
2. https://www.youtube.com/watch?v=Tw0mG4YtsZk&t=496s/
3. https://www.youtube.com/watch?v=LJ3ic8C8CcA&t=390s