Smart Traffic Light Control System

Nicole Díaz
Facultad de Ingeniería
en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
ncdiaz@espol.edu.ec

Jorge Guerra
Facultad de Ingeniería
en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
jiguerra@espol.edu.ec

Juan Nicola
Facultad de Ingeniería
en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
jxnicola@espol.edu.ec

***Abstract*—This paper describes the design and implementation of an autonomous traffic light system, with traffic-based timing for improving traffic flow efficiency in urban roads. The importance of changing light timing lengths of current traffic lights to timing lengths that vary depending on the number of vehicles in the avenues, and of how the proposed smart traffic light system would help to constantly update these times of automatic way is presented. To achieve this, an IoT system based on the Raspberry Pi platform and PIR sensor will be designed, with expandability in mind, as the design for implementing camera-related functions will be layed out.**

Keywords— Traffic control system, Smart traffic lights, Raspberry Pi, PIR sensor, Internet of Things.

I. INTRODUCTION

Transportation is one of the fundamental anthropogenic activities for the development of societies. Modern terrestrial transportation is achieved thanks to all types of vehicles such as cars, motorcycles, bicycles or public transport. The use of these is subject to rules or regulations established by the government institutions based on the place where they are located. Traffic regulations were designed with the aim of safeguarding the life and integrity of people whether they are drivers or pedestrians, as well as organizing and controlling pedestrian traffic. Traffic Lights are common devices which used to control the flow of traffic, as well as providing pedestrians a safe way to cross streets. However, traffic lights are not perfect and the may cause additional delays in traffic.

A. Justification

Due to the exponential growth of ground travel, the number of cars traveling on each street is increasing. Considering that traffic lights installed on different roadways were configured to handle a fixed amount of traffic, it it clear that they do not adequately serve modern day traffic loads. This is because the light timing lenghts that were initially programmed are not efficient enough to handle today's traffic patterns. Light timing lengths should be dinamically updated in response to live traffic loads. In this paper, we will propose the use of Smart traffic lights that do so, depending on the number of vehicles located near each intersection where traffic lights are located.

978-1-5386-6657-9/18/$31.00 ©2018 IEEE

II. METHODOLOGY

A. City of study

According to the National Institute of Statistics and Censuses[1], the province of Guayas, Ecuador, where our city of interest, Guayaquil, is located, has more tan 362.857 vehicles on its streets in 2015, representing a 57% increase in respect to 2010.

Table I
REGISTERED CARS IN ECUADOR, 2015

Province	Number of vehicles
Pichincha	492.568
Guayas	362.857
Manabí	152.231

According to the information shown in Table 1, we note that, as of 2015, Guayas is the province with the second-highest number of registered vehicles, bringing a number of issues, particularly traffic congestion. Traffic lights have been implemented around the world since 1868 to try to mitigate this problem, with the most representative implementations shown below:

B. Traffic congestion

Traffic congestion is a condition on transport networks that occurs as use increases, and is characterized by slower speeds, longer trip times, and increased vehicular queuing. When traffic demand is great enough that the interaction between vehicles slows the speed of the traffic stream, this results in some congestion.

C. Traffic lights as traffic control devices

In recent years, technology has constantly evolved, bringing improvements to people's lives. Traffic control is one example of that. Traffic lights were designed with gas-based lightning, but it did not take too long for it to become a fully electrical system. Nowadays, conventional traffic lights are LED-based, due to its low energy requirements [2]. The 3 colors that signal the right of way allocated to users follow a universal color code:

- Red: Forbids passing of vehicular traffic and permits pedestrian crossing. Lasts between 28 to 40 seconds.
- Amber: Warns that the light is about to change from green to red. Lasts between 2 to 5 seconds.
- Green: Allows vehicular traffic to pass and indicates and forbids pedestrian crossing. Lasts between 28 to 40 seconds.

This system has prevailed since its introduction, due to its relatively low cost. Guayaquil is a city still maintains traditional traffic lights. Traffic congestion is generated at intersections due to the waiting times for each way. The system described below will solve this problem, dynamically allocating time to each way according to the realtime traffic situation.

Traffic lights color timing shouldn't be fixed, but allocated dynamically following some variables such as actual traffic flow, average speed of crossing, distance between vehicles, etc. Taking this into account, a more useful light timing could be determined and thus avoiding excess vehicles queuing at intersections for an extended period of time. The system proposed will build on realtime traffic data collected on-site, to achieve the installation of "Smart" traffic light on city streets.

D. Communication technologies in IoT

Automation is one of the main growth areas of technology, due to the constant increase in industrial production. In the vast majority of automation developments, the Internet is essential for their operation.

Sensors and actuators are fundamental components of IoT applications, because information is constantly obtained to be processed, stored or sent to other devices through wireless networks. [3].

Large number of devices are connected to the Internet, via TCP/IP. This model can be used for large networks, like the Internet, or small ones like a personal area network. However, energy consumption varies per application. [4]

The aforementioned project is based on the application of IoT design principles: a motion sensor is at the heart of the solution. It measures the infrared light radiated from the objects located in its field of vision [5], which in this case are the vehicles. Through the TCP/IP protocol, the traffic lights communicate with each other, and with the information obtained by the PIR sensor, the behavior of the traffic lights is determined, as it will be explained in this article.

III. Design & Implementation

For the development of this project, we made use of the following parts:

- Raspberry Pi 2 Model B
- 5 mm red-color LED
- 40-pin data bus
- 3 ohms resistor
- Common cathod RGB LED display
- PCB
- PIR motion detector sensor

The Raspberry Pi 2 microcontroller handles the LED lighting of the traffic light according to the times previously set with Python code. Depending on the output of the Pir motion detector sensor which, when detecting the infrared radiation change through its line of sight, sends a HIGH to the microcontroller. On the PCB board all the red-colored LEDs are solded in parallel. They will flash when the traffic light changes from red to green. The common cathod RGB LED display, solded in parallel on the PCB board, consists of 4 terminals. The longest terminal corresponds to the common cathod while the other 3 terminals correspond to the emission of red, green light and blue respectively. The microcontroller will send PWM signals to each of these terminals and depending on the work cycle that is applied to the PWM signal of each terminal, the RGB LED of a will light up with the correct color. A resistance of 100 ohms was used for each terminal per color of the RGB LED display, to avoid excess current that could burn these components. To connect the components of the PCB with the Raspberry, a 40-pin data bus welded on the PCB was used. On this bus, the male-female jumpers are connected to the Raspberry Pi 2.

IV. Design of Proposal

In Guayaquil, one-sided traffic lights are ubiquitous. They normally consist of 3 lenses, however there are traffic lights that have up to 4 faces due to the number of intersections.

The traffic control system described in this paper, in addition to having additional functions and actions to the traditional traffic lights.

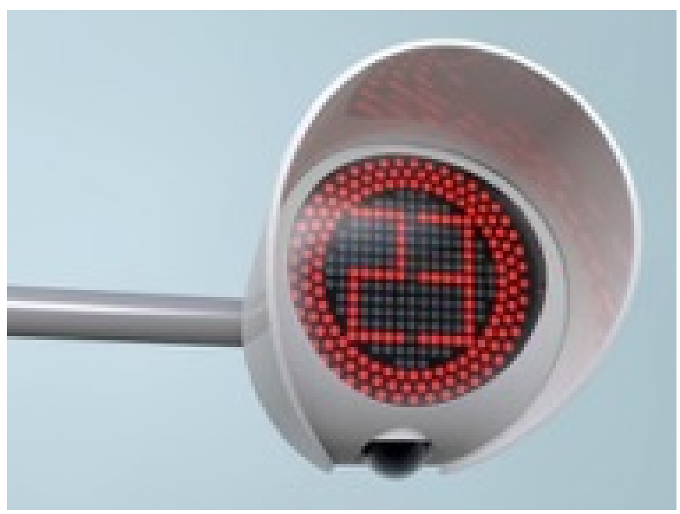

Figure 1. Traffic light

The design that shows in the Figure 1 proposes the use of 1 lens consisting of a single circumference of RGB LEDs that varies their color according to the traffic color codes (red, amber, or green)[6]. The center of the lens displays the number of seconds remaining in its current state, before a transition from red to green happens, with the purpose of accelerating traffic flow.

This design also incorporates a bar of red LEDs that indicate the countdown of the completion of the waiting stage of the vehicles, which is signaled as a red traffic light. This design will also have a PIR sensor, which will be located towards the vehicles that pass in the street controlled by that traffic light.

The functionality of the system is described below:

- The traffic light initially operates like any other traffic light, with usual fixed times for each state.
- At an intersection of two traffic lights, both are constantly communicating with each other as they are connected to the same network.
- Each traffic light receives information from the corresponding PIR sensors, during the normal operation of the traffic light.
- When the PIR Sensor of a traffic light captures a vehicle, it transmits a message to the other traffic light, making it change its state to red. The traffic light that captured the car and sent the message proceeds to change its state to green.
- Each traffic light has a minimum duration of green state, that is, if the other traffic light receives a vehicle through its way, but the traffic light that is in green state has not yet elapsed its minimum time, the other traffic light will have to wait that minimum time for changing its color and it will not do so instantaneously.
- While a traffic light is in red, the matrix of red LEDs display the number of remaining seconds before changing to green.
- With the design of this system there are a number of possibilities regarding the operation of the LED Matrix that can be configured according to the needs or preferences desired by the customer, or the intersection in which it will be placed.

V. Camera system

The PIR sensors will be replaced by video cameras as initially planned [7], which will process the images with the OpenCV open source library using Python to determine the number of vehicles, speed, distance between vehicles, and other important variables [8]. Thanks to this, the traffic lights in turn will generate data-based statistical reports, so that the colors of the traffic lights will be dinamically set. In this way the traffic lights will be set up with the best timing based on data for each time slot per weekday. Additionally, image recognition function will be added to capture the license plates of the vehicles circulating in the traffic light location, and also to be able to determine which vehicles violate traffic laws either by passing the Red, speeding, or any other contravention.

VI. Results

This first prototype of the intelligent traffic light manages to speed up the intersections in which one avenue is not transiting vehicles in a moment of time while in the other street there are cars waiting, since when not detecting vehicles in the first, it automatically gives way to the second. The operation was tested with two traffic lights that communicate with each other connected to the same network either by ethernet cable or with a wifi module. When the traffic light 1, in red, detects the arrival of a vehicle through the PIR sensor, it sends a message to the traffic light 2, in green, indicating that it finishes its time and goes to yellow to later be red, while the traffic light 1 starts a countdown through the array of LEDs and the indicator of the red LED bar. Upon ending the red light of traffic light 1, it sends another message to traffic light 2 to tell it to turn yellow and then red, while traffic light 1 turns green, with the respective delays necessary to avoid accidents. All of these times can be configured by programming in Python according to the needs of each intersection. All this was achieved successfully through the first prototype and for the next prototype it is expected to implement artificial vision and achieve correct vehicle counting.

The prototype that has been built consists of several parts: PCB, Circuit Schematic, Python Code and Basic design of the traffic light housing.

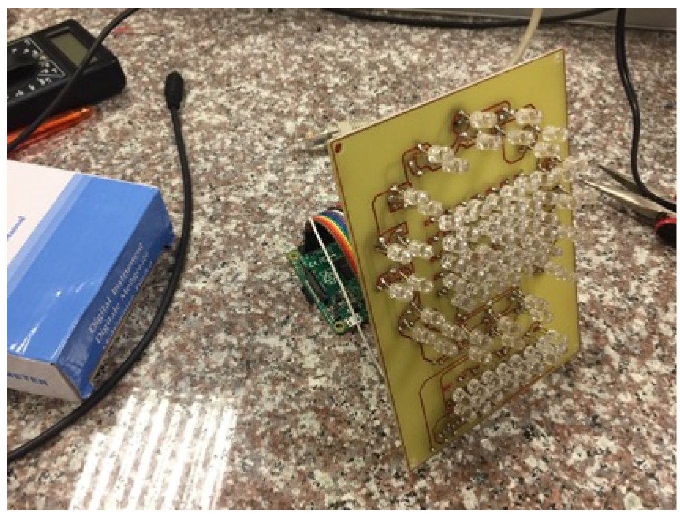

Figure 2. Prototype

A. PCB design

The components were layed out in the PCB in a cyclops-type form traffic light with a 5x7 array of leds at its center to write 2-digit numbers and different shapes and symbols such as X, seen, circle, ”ok” , among others. These forms are configurable in Python code. Below the circle of LEDs is the bar of red LEDs. The plate was printed on both sides. The dimensions of the plate are 145mm x 105mm. The diameter of the semaphore circle is 100 mm. The LED bar is 12.5mm x 60mm.

B. Design in PCB

The circuit basically consists of the connection between the Raspberry GPIOs to the female data bus soldered on-board, in such a way that the Raspberry feeds the RGB LEDs which are connected in parallel. The LED bar is powered with a single GPIO. The color displayed is controlled by the PWM pulses

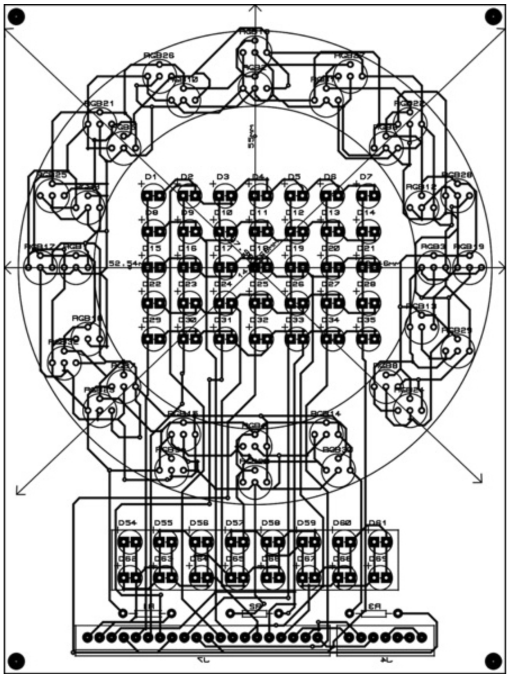

Figure 3. Design in PCB

sent to the RGB LEDs. In this prototype we used a common resistance of 330 ohms for each color of the RGB, that is, 3 resistors. This is not ideal since with this configuration a lower intensity in the LEDs is achieved. In the next prototype this will be improved.

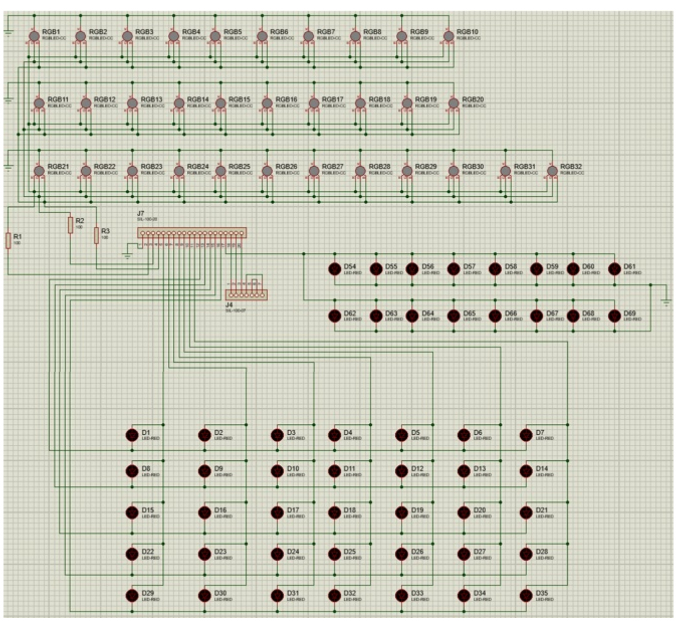

Figure 4. Schematic of the device.

C. Python Code

The Raspberry was programmed using the Python. At the time when the traffic lights are installed at the intersections, the timing of each color, delay times between colors, IP address of each traffic light, count to be made in the array of LEDs, must be previously configured in a user-friendly interface. All this is easy to configure thanks to the distribution by functions of the programming which is commented for easy understanding.

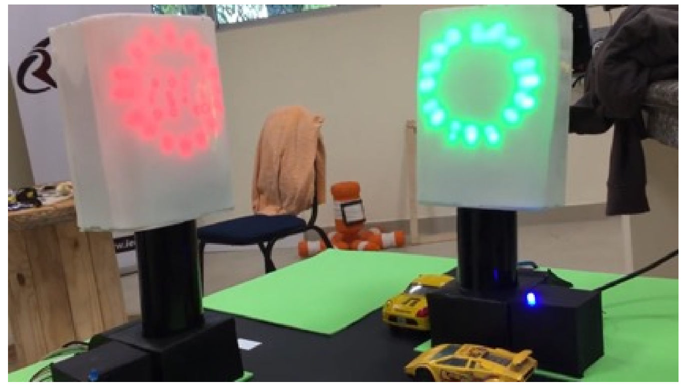

Figure 5. Lighting

D. Case basic design

The case of the traffic lights was made with white acrylic and molded by a heat gun, in such a way that the shapes presented in Figure 5 were obtained.

VII. CONCLUSIONS

This article serves as a foundation for a larger project involving total traffic control automation using the intelligent traffic lights. This first prototype as shown which will be enhanced with more functions and improving it over time to meet traffic demands. Some of the functions that will be added to the future in this project is the vehicle counting by artificial vision with Raspberry Pi and OpenCV, detection of vehicle plates, detection of vehicle speeds and sending data obtained by traffic lights to the cloud.

REFERENCES

[1] El Telégrafo, ”INEC: parque automotor de Ecuador creció 57% en 5 años”, 2016.

[2] V. Valencia Alaix, ”Principios sobre Semáforos”, PhD, Universidad Politécnica de Valencia, 2000.

[3] A. González García, ”IoT: Dispositivos, tecnologías de transporte y aplicaciones”, Msc, Universitat Oberta de Catalunya, 2017.

[4] J. K. and A. Desai, ”IoT: Networking Technologies and Research Challenges”, International Journal of Computer Applications, vol. 154, no. 7, pp. 1-6, 2016.

[5] ”Application of Raspberry Pi and PIR Sensor for Monitoring of Smart Surveillance System”, International Journal of Science and Research (IJSR), vol. 5, no. 2, pp. 736-737, 2016.

[6] K. Choi, ”Visible Light Communication with Color and Brightness Control of RGB LEDs”, ETRI Journal, vol. 35, no. 5, pp. 927-930, 2013.

[7] P. N R, ”Smart pi cam based Internet of things for motion detection using Raspberry pi”, International Journal Of Engineering And Computer Science, 2016.

[8] B. Gupta, A. Chaube, A. Negi and U. Goel, ”Study on Object Detection using Open CV - Python”, International Journal of Computer Applications, vol. 162, no. 8, pp. 17-21, 2017.