"Automated Traffic Control System using Wireless Sensor Networks and IoT"

Arpitha M D[1], Bindhya N[2], Reshma R[3], Impana Appa

[123]Computer Science & Engineering, Academy for Technical and Management Excellence college of Engineering, Mysore, Karnataka, India,

[4]Assistant Professor, Computer Science & Engineering, Academy for Technical and Management Excellence college of Engineering, Mysore, Karnataka, India,

***Abstract*-** An Automated microcontroller based traffic control system using sensors along with live web updates can be a helpful step in optimizing the traffic flow pattern in busy intersections. This intuitive design of the transport infrastructure can help alleviate the traffic congestion problem in crowded cities. This paper describes a system where ultrasonic sensors are integrated with the Raspberry Pi to operate the lanes of an intersection based on the density of traffic. The current condition of the intersection is updated on a user accessible website. This integration of traffic systems in an Internet of Things (IOT) fashion enables the addition of smart security and road safety devices. As a result, the improvement in traffic system can be incrementally enhanced, which can lead to eventually significant improvement in the overall traffic system.

***Keywords*-** Automated traffic system; Raspberry pi; Ultrasonic sensor; Internet of Things (IoT); Traffic congestion

I. INTRODUCTION

Internet of Things (IoT) is the recent ongoing phenomenon, which tells that in the near future all physical devices will be internetworked. As internetworking of physical devices with the Internet gets more popular day by day, these internetworked devices have the potential to integrate among themselves in terms of operation. The production of selfdriving cars is rising and it is said to be the next major technological advancement in the transport industry. This change in the transport ecosystem requires redesigning the transport infrastructure so that this development of embedded vehicles can be internetworked with a similar embedded traffic ecosystem. A typical traffic system consists of four lanes each having a signal with a fixed timer on each end that operates sequentially. The problem with this conventional system is that it cannot detect the density of traffic on each lane; therefore, time is wasted even when a lane is empty. Along with inefficient timing it cannot restrict the vehicles from blocking pedestrian crossing, cannot operate in accordance to the density of vehicles on each lane, there is no mechanism for commuters to check traffic condition of a route and the current infrastructure will bottleneck driverless cars from reaching its full potential.

There are several adaptive traffic control system using methods such as GPS tracking [1], RFID (radio frequency identification) technology [2], image processing using CCTV cameras [3], VANET (Vehicular Ad-Hoc Network) [4], text mining on data collected from social media and many systems use algorithm based systems for efficient traffic management [5][6][7]. IoT based traffic systems do exist but in the form of using networking based ideas and several software applications that take traffic update from the application users, based on their location [8][9][10]. The current systems available do not provide real time update of traffic conditions to the commuters.

Integrating multiple ultrasonic sensors with a website, a system is designed that will be more efficient, safe and open the path to future innovations. A Raspberry Pi microcomputer and multiple ultrasonic sensors are used in each lane to calculate the density of traffic and operate the lane based on that calculation. While operating the lanes, traffic data are simultaneously uploaded to a website, which can be accessed by any user. This feature can be used by users to get real-time information for a given road intersection and set their route accordingly.

The motivation behind this approach was to create an embedded system that will be effective, enable addition of sensors and make it very simple to implement. The system design, components and operation used to build and control the system are discussed in the following sections.

II. SYSTEM DESIGN

Fig.1: shows the overall design of the system. In this intersection, each outgoing lane has four ultrasonic sensors that calculate and report the traffic conditions of each lane to the Raspberry Pi.

The Raspberry Pi uses this information to set the signal timer according to the level of traffic and updates this information to a website. Sixteen ultrasonic sensors are used in total, out of which four are used for the buzzer alarm system.

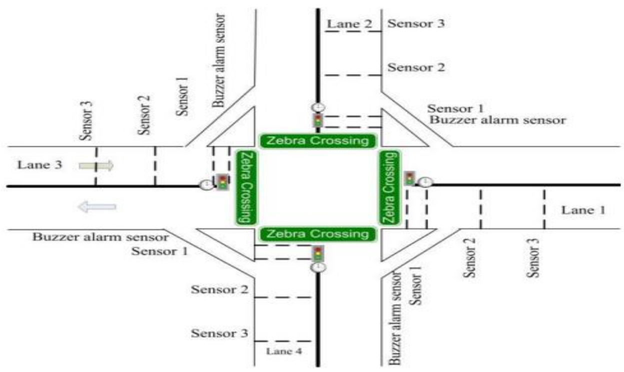

Fig. 1. The model of the system

III. COMPONENTS

The components used in this system are listed below.

A. Raspberry pi 3 Raspberry pi is a miniature computer with an operating system that can be used as a development tool for different software and hardware based projects. In this project, the Raspberry Pi 3rd generation was used for its superior processing power compared to other available microcontrollers.

B. MCP23S17 MCP23S17 is a 16-bit I/O extender integrated circuit (IC) using serial peripheral interface (SPI). This IC enables the addition of more devices to the system if necessary.

C. Ultrasonic Sensor HC-SR04 HC-SR04 is an ultrasonic sensor that can sense objects within the range of 2-400cm.

D. Buzzer The buzzer is used as an alarm in the pedestrian crossing as a safety device.

E. Light Emmiting Diode (LED) LED is used for traffic light operation. F. Dual 7-Segment Display The display is used to show the traffic timers.

IV. ASSEMBLY

The methods used to assemble all the components are discussed in this section. Table I shows the number of I/O pins used in the design and also how they are distributed among each component. It is also used to represent how the number of I/O pins were reduced to increase the efficiency of the system.

TABLE I. COMPONENT-WISE I/O REQUIREMENT

Component name	*Number of components used*	*Number of I/O pins required for each unit of component*	*Total number of I/O pins required*	*Number of I/O pins used*
Ultrasonic Sensor HC-SR04	16	2	32	17
Buzzer	4	1	4	4
Dual 7-Segment Display	4	9	36	15
LED	12	1	12	12
MCP23S17	2	4	8	4
Total				52

The ultrasonic sensors require a total of 32 I/O pins but it was reduced to only 17 by making the trigger I/O common for all sensors. The Dual 7-Segment Displays and MCP2317s combined require 44 I/O pins in total. But, by multiplexing the respective components, the number of I/O pins is reduced to only 19. Table I shows that the total number of I/O pins required are 52, while the Raspberry Pi 3 has 26 pins that can be used for I/O. This issue of shortage of I/O pins is solved using the MCP23S17 IC. Fig. 2 shows the connections from Raspberry Pi 3 to the MCP23S17s and the sensors. The ultrasonic sensors are connected directly with the Raspberry Pi.

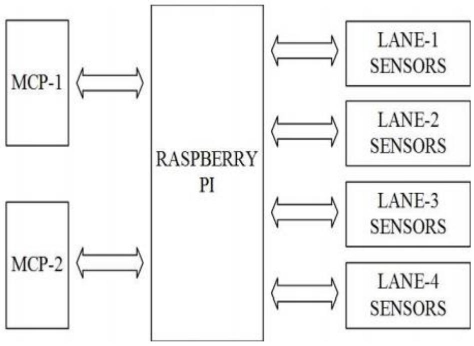

Fig. 2. Connection with Raspberry Pi

MCP23S17 is used to expand the number I/O pins in the system. Each MCP23S17 can add an extra 16 I/O pins to the system. The designed system consists of four sensors in each lane, which requires two MCP23S17s to balance the number of I/O pins needed. However, eight MCP23S17s can be multiplexed together at a time, offering 128 extra I/O pins. This is useful in case the system needs to be further upgraded to a better version using more components. The Raspberry Pi 3 checks the 3-bit control address of an MCP23S17 before communicating with it. Fig. 3 and Fig. 4 demonstrates the connections from the two MCP23S17s used in the design.

V. OPERATION OF THE SYSTEM

A. Calculation This system uses ultrasonic sensors to measure the level of traffic in each lane. When triggered, the ultrasonic sensor generates a pulse, transmitting an echo from the sensor that bounces off any object in its path and the echo is received back at the sensor. The time delay between the transmission and the reception of the echo is used to calculate the distance of the object from the sensor. Distance is calculated using the formula in (1),

where the speed of sound is 34000 cm/s.

Distance = (Time between Echo*34000)/2

The logic of the system uses the width of each lane as the threshold value to determine if a vehicle is present or not. If the distance calculated from an ultrasonic sensor is less than the threshold value, the system concludes that a vehicle is present in that position, else it concludes that the lane is empty at that position. The combination of outputs from multiple ultrasonic sensors placed at different positions along the lane is used to compute the density of traffic for each lane. B. Computing Traffic Density The density of traffic is computed to three fixed levels called Low, Medium and High. Each lane uses data from three ultrasonic sensors to examine the level of traffic.

If the distance calculated by using (1) is less than the threshold value, the output from that is sensor is taken as ‘1’, else the output is considered to ‘0’. An output of ‘1’ signifies that a vehicle is present at that location, while an output of ‘0’ signifies an empty lane at that position.

The traffic density for each lane is assigned based on the following conditions: x

Low: (Sensor 1=0, Sensor 2=0, and Sensor 3=0) or (Sensor 1=1, Sensor 2=0, and Sensor 3=0) x

Medium: (Sensor 1=1, Sensor 2=1, and Sensor 3=0) x

High: (Sensor 1=1, Sensor 2=1, and Sensor 3=1)

A fixed time is set in the program code for each level of traffic density. For testing the system, the following time periods have been used for each level of traffic: x Low: 10 seconds Medium: 20 seconds x High: 30 seconds
The program runs an endless loop where each lane is activated in a serial order. Data collected from the ultrasonic sensors is used to generate the traffic density. The information regarding the level of traffic congestion is stored in a file and it is updated to the designated website. The timer is set for that lane according to the level of traffic. When the timer for a lane is counting down, it continuously monitors if the other lanes, currently with red signals, have any vehicle blocking the zebra crossing. If vehicles in any other lane get too close to the zebra crossing, the system will sound an alarm, issuing the warning to back off and keep a safe distance. This whole process is applied in a loop for each lane in the intersection.
D. Online Update The traffic data collected at each step is updated to the webpage using Flask [11], which is a micro framework for python. The web server used is Apache 2. Traffic condition calculated for each lane is stored in files, where each lane has a file containing its information. Here flask is used to read the data from each file containing traffic data and pass it on as a parameter to the HTML file of the webpage.

Fig. 6. View of the webpage

Flask is very simple to implement, so it is fast and easy to update data to the web interface. The main purpose of the webpage is to display and update traffic data in real-time while the hardware system is running. Fig.6 shows the design of the webpage. It can be seen that the name and traffic condition of each lane is displayed at the edge of every road. In the webpage, alongside written text, the traffic densities are also represented with respective color codes (Green: Low traffic, Yellow: Medium traffic, Red: High traffic) to make the design more user friendly.

VI. RESULTS AND DISCUSSION

Table II shows a small comparison between conventional traffic system and automated traffic system to demonstrate the advantages of using an automated system. Lanes of an intersection can have many different combinations of traffic congestion. From Table II, it can see that for every case except one, the automated system takes less amount of time to complete a cycle. The range of time taken can vary from 33.3% to 83.3% of the time it would take in a conventional traffic system. These results show significant improvement in traffic time management.

VII. FUTURE WORK

More sensors can be used in each lane to make the system more accurate and sensitive to small changes in traffic density. Driverless cars can access the website to view the intensity of traffic at an intersection and choose the fastest route accordingly. Data mining techniques such as classification can be applied on traffic data collected over a long term to study the patterns of traffic in each lane at different times of the day. Using this information, different timing algorithms can be used at different points of the day according to the traffic pattern.

VIII. CONCLUSION

A more efficient and safe traffic ecosystem has been developed which provide commuters with live update of the road condition in a website. In the fixed timer traffic system, the lane signals are operated one after another based on a fixed set timer. This causes traffic congestion on a high traffic lane due to wasted time on a low traffic lane, and this also restricts the use of new technology. The automated traffic system discussed in this paper can reduce the time spent by commuters on traffic signals and reduce traffic congestion on busy lanes caused by unnecessary signals on empty lanes . It can also ensure safety of pedestrians using the buzzer alert if vehicles block the zebra crossing. It can also help commuters reroute their destinations using the live traffic update from the website. This IoT based automated traffic system can be a significant step towards the development of future smart cities.

IX. REFERENCES

[1]. R. Dhakad and M. Jain, "GPS based road traffic congestion reporting system," 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, 2014, pp. 1-6. doi: 10.1109/ICCIC.2014.7238547
[2]. Q. Xinyun and X. Xiao, "The design and simulation of traffic monitoring system based on RFID," The 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, 2014, pp. 4319-4322. doi: 10.1109/CCDC.2014.6852939
[3]. M. F. Rachmadi et al., "Adaptive traffic signal control system using camera sensor and embedded system," TENCON 2011 -

2011 IEEE Region 10 Conference, Bali, 2011, pp. 1261-1265. doi: 10.1109/TENCON.2011.6129009

[4]. X. Jiang and D. H. C. Du, "BUS-VANET: A BUS Vehicular Network Integrated with Traffic Infrastructure," in IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 2, pp. 47-57, Summer 2015.doi: 10.1109/MITS.2015.2408137

[5]. I. Septiana, Y. Setiowati and A. Fariza, "Road condition monitoring application based on social media with text mining system: Case Study: East Java," 2016 International Electronics Symposium (IES), Denpasar, 2016, pp. 148-153.doi: 10.1109/ELECSYM.2016.7860992
9/ICIEV.2016.7760025

[6]. E. D'Andrea, P. Ducange, B. Lazzerini and F. Marcelloni, "Real-Time Detection of Traffic From Twitter Stream Analysis," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 2269-2283, Aug.2015.doi: 10.1109/TITS.2015.2404431

[7]. F. Ferdous and M. S. Mahmud, "Intelligent traffic monitoring system using VANET infrastructure and ant colony optimization," 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, 2016, pp. 356-360. doi:10.110