International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

SMART PARKING SYSTEM USING IOT

Ojas Amit Pethe[1], Aditya Milind Bhabhe[1], Saurabh Kiran Ambardekar[1], Prof. Nilesh A. Lakade[2]

[1]Department of Electronics Engineering, K. J. Somaiya College of Engineering (Autonomous), (Affiliated to University of Mumbai), Mumbai, India
[2]Assistant Professor, Department of Electronics Engineering, K. J. Somaiya College of Engineering (Autonomous), (Affiliated to University of Mumbai), Mumbai, India

---

Abstract - *The objective of this paper is to give a better solution to the troublesome problem of car parking. An attempt has been made to make an IoT enabled parking system, thus reducing the human efforts, cost, time to search for an empty slot and also the fuel consumption of the vehicle. While developing this system, some existing techniques which are based on IoT were studied, this technique differs from those by means that there is a provision made to store the user entries and their car's number plate in the database. The entire flow of the process is controlled by Raspberry Pi. Lastly, this paper incorporates the images of working website displaying availability of slots in red and green and also the entries in database alongside the user number plate.*

Key Words: Open ALPR, Website, Database

1. INTRODUCTION

With rapidly changing technology, the concept of Smart City has become very popular around the world. 'Smart' means everything is connected to each other through Internet of Things(IoT) and data at each node is collected and is used to find different trends. As a small step towards such Smart City, we intend to make a parking arena without any human intervention. Especially, in huge parking spaces, it becomes very difficult to find a place to park without any human assistance. Not only it is a very time consuming process but also more fuel is consumed by the vehicle leading to air pollution. So if the driver already know, which slot is available it will be easier to park a vehicle also saving time and energy. Attempts have been made to give an effective solution to manage parking arenas automatically but none of them gives a completely integrated solution. One of the techniques Smart car parking system using IOT concept, used IR sensors to check the availability of the parking slots along with temperature and light sensors which were then connected to microcontroller. Ethernet shield and microcontroller formed connection between sensors and cloud. Simple webpage was created to check data from end nodes i.e sensors [1]. On the other hand, Smart Car Parking system was developed with help of Arduino, using Network Protocols to connect the device to the cloud. Cloud Computing was used to exchange the information or data in between the hardware and an end- user device. The proposed project also displayed temperature changes and varying light conditions in the parking slot [2]. Other techniques also involve the use of traditional controllers for making a smart parking system one of which is an IoT based system which uses a PIC microcontroller with IR sensors and WiFi module interfaced to it. It sends the sensor data over WiFi to the mobile connected to the same network to which the controller is connected. The entry to the parking lot in this case is password protected [3]. A system has been developed which focuses on Slot security asking for user credentials at the entrance. The user can see the available slots through an website only if the user enters the required details. It also captures the vehicle number plate and stores the number plate value in the database. If the number plate is not detected at the entrance, the user is not able to access the parking space; this ensures that all necessary details are stored. Number Plate Detection is done using Image Processing tools [4].

2. BLOCK DIAGRAM

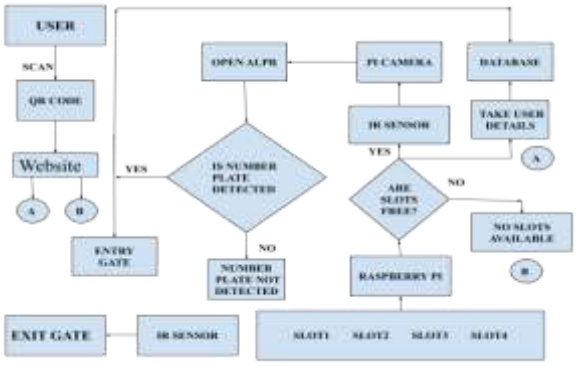

Fig-1: Flow Chart

Fig-1 shows the proposed technique. The car arrives at the entrance. It is detected by the IR sensor at the entrance. QR code is available at the entrance which contains the link of the website. The user now has to scan the QR code that directs the user to that particular website. User has to enter the details in the form of that Website and can see the slots both available/unavailable. Once the car is detected by the IR sensor the Pi-camera captures the image and sends the data to the Open ALPR cloud. The number plate is detected. Both user details and the results are stored in database for future access. Once the number plate from the

cloud is obtained it is indicated by the status code (for success = 200) when the code runs and once this status code is achieved, the entrance gate opens. The user enters the parking area and avails the slot. At the exit, IR sensor detects the car and till the time IR sensor gives high output, the exit gates are opened, the user now can exit the parking area.When all the parking slots are full the HTML page will now show a message indicating Unavailable Slots and also the car will not be detected at entrance.

3. HARDWARE DESCRIPTION

3.1 Microcontroller:

Raspberry pi is used as microcontroller for processing the data coming from sensors. Due to its Small size, high processing power, we preferred using Raspberry pi over other processors available in the market. Another advantage is that, Raspberry pi is a general purpose computer having its own operating system such as Raspbian, Windows 10 IOT core, Moebius etc. For our model we installed Raspbian which is the most popular operating system for IOT applications

Raspbian has been used extensively because it is based on Debian (Linux) which makes it easy to use and it also protects against malware. The output of the IR sensors is saved in the text file in Raspberry pi.

3.2 Sensors:

We used IR sensors at the parking slots and at both the gates. These sensors are connected to the microcontroller through wires. The output pins are connected to the GPIO pins of Raspberry pi. Raspberry pi has total 26 GPIO pins out of which 2 are used by the servo motors. Hence total 24 sensors can be connected to Raspberry Pi through wires. These connections can also be made wireless and the number of sensors integrated can be increased by using MUX [5].

3.3 Servo Motors:

Two servo motors are used at the ENTRANCE and EXIT gate which are interfaced using the python script. The angle of rotation of both the motors is specified in the python script. Like IR sensors servo motors are also connected with wired connection.

3.4 Pi-Camera Module:

The function of camera module is to capture the image and pass these pictures to the processing unit. In our model, pictures are captured when the camera module receives the signal from IR signal which is situated at the entrance gate.

4. SOFTWARE DESCRIPTION

4.1 Number Plate Detection using Open ALPR:

We have used Open ALPR (Automatic License Plate Recognition) tool to recognize the vehicle number plate. We are calling the API ,designed to convert an input image to the text document featuring the number plate from the input image, using a Python Script based JSON tool. Open ALPR is a cloud service and an open source platform. It operates as a pipeline based architecture where the input is a Base64 converted image file, in between stages are based on image processing and the final stage is to extract the text from the cloud platform using a Python code. The Image Processing steps include Detection: to find the license plate regions, Binarization: to convert the detected image from RGB to monochrome, Edge Detection: to detect the possible license plate edges, Character Segmentation: to segment the preprocessed image, OCR: Optical Character Recognition is an advanced tool to detect text from an image file, Post Processing: to arrange into a set of best possible OCR detected text files. The extracted output plate of great precision is stored and also used to as a parameter to control the entry gate[6].

4.2 Database:

We created database which has three columns for user's name, contact number and car number. User has to fill his name and contact number in the form created on website before entering the parking area. Car's number gets added in the third column once the camera module captures the image of the number plate and the required text from the captured image is extracted. Database was added in the model for security purposes.

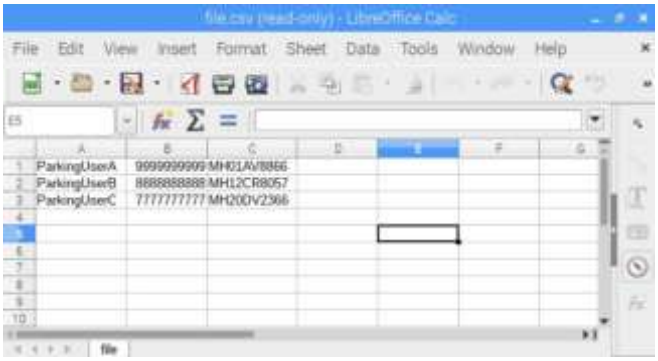

Fig-2: Database

4.3 Website and Server Implementation:

A user interface is created in this case a website through which sensor data can be seen by the user. Webpage was created using HTML and CSS and these pages are then linked to the database. Raspberry pi used has the ability to host a server locally. The required packages for installing apache server which is the local server were installed and the HTML and CSS scripts were then moved to the specified directory. When the user puts the link of the webpage on the search engine the webpage is displayed on the users phone. To simplify this process we created QR code linked to the webpage.

There are two separate servers for front-end and back-end programming. Apache server is a front-end server and node.js is the back-end server. The data of the parking slots is collected by Raspberry Pi and stored in a text file. An API has

been created to communicate between front-end and the back-end server. API was created using Express framework of node.js. This API call is made by the webpage displaying the availability of slots. AJAX i.e. Asynchronous Java script and XML is used to call the API after every second to update the HTML web page regularly.

For database creation, PHP coding was used. When the QR code is scanned and the user enters the details, the PHP code gets these details and stores them into CSV file. The next entered user credential is appended to the same csv file creating an offline database. This PHP code also reads the text file containing the scanned Number plate data and appends this string into the third column of the same csv file. So, a database has been created which stores all the database data which all is controlled by a PHP file.

5. COMPONENT INTERFACING

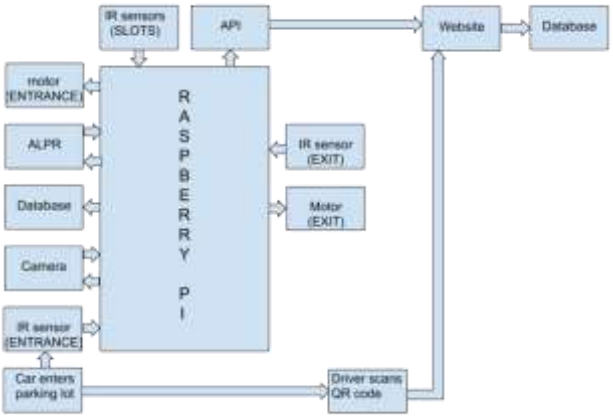

Fig-3: Component Interfacing

6. RESULTS

6.1 Sensor Outputs:

Fig-4 shows the change in response of IR sensor as the car enters or leaves the slot. This result is stored in a text document. This text document is then referred to read the slot values and change the indicating colors on the website. Sensors give output as 0 when it sense any object and 1 when object is absent.

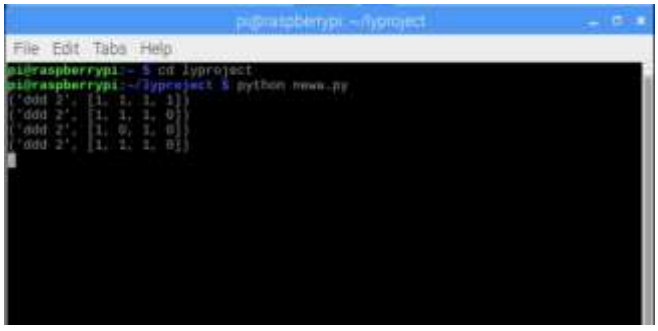

Fig-4: Response of IR Sensors

6.2 License Plate Detection

Fig-5 displays the output of the steps carried out at the entrance and also the number plate which is detected. This number plate is stored in the database alongside the user credentials.

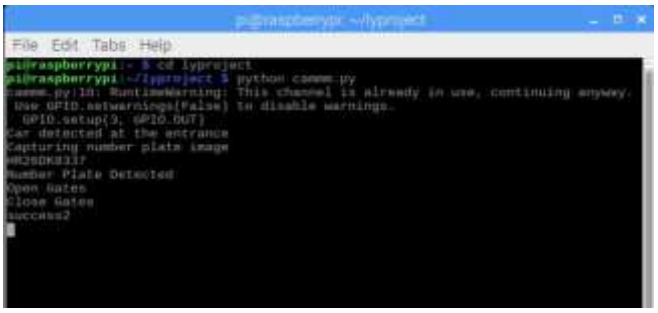

Fig-5: Detected Number Plate

6.3 Dynamic Coding:

The Javascript code, required to make a dynamic website, is run using the terminal command as shown in Fig-6. This also hits the webpage every one seconds using an API that makes the necessary changes.

Fig-6: API listening

6.4 Website:

a) Starting Page: This is an interactive webpage, as it is visible on a mobile screen. This page has two possible cases:

i) When some slots are empty, this HTML page is developed to acquire the user credentials and store the inputs to a database. The first tag i.e Name is set such that at least six characters can be entered, also the second tag i.e Contact is set to get only numeric values. This website is achieved after the user scans the QR code. This can be observed in Fig-7a.
ii) When all slots are occupied, the Webpage now will display “No Slots Available”. We can see this in Fig-7b

Fig-7a: Case i

Fig-7b: Case ii

b) Final Page: Fig-8 is the correct representation of slots as it is accessed from the user device. This ensures that the user is known about the availability of parking slot as he enters the parking area. These changes are made with change in respective IR sensors connected at every slot. All the changes are saved in a file and this file is referred by the website to make the variations visible. At this instance the webpage is showing Slot 1, Slot 2 and Slot 3 occupied and Slot 4 is free.

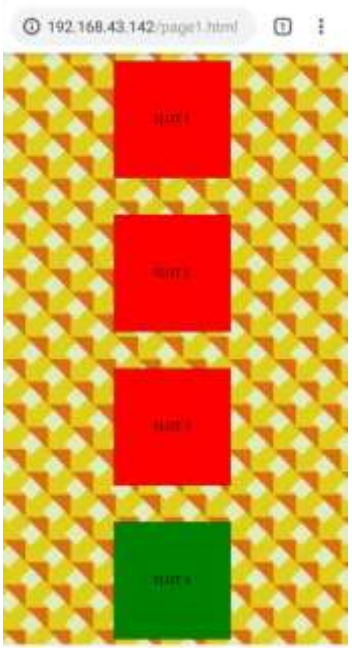

Fig-8: Slot Availability

CONCLUSION

A coherent approach to solve the issue of Car parking was developed overshadowing system security and robustness. Also the Vehicle license plate recognition based on real time captured images was achieved. This technique ensured security at operational levels and also made it easier for the user to avail the parking. A prototype was designed to illustrate the proposed model and the results are noted.

REFERENCES

[1] Gouhar Fati, Chanukya Ranima, Imtiyazunnisa Begum, "Smart Parking System Using IOT and WSN", International Journal & Magazine of Engineering, Management & Research, 2017,Volume 4, Issue No.3 (March), ISSN 2348-4845

[2] Vrushali D. Ichake , Priya D. Shitole, Mohsina Momin, Kanchan S Thakare (Assistant Professor), "Smart Car Parking System Based on IOT Concept",International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726, www.ijesi.org , Volume 5 Issue 3, March 2016, PP.48-54

[3] Dr. Y Raghavender Rao, "Automatic Car Parking System using Internet of Things (IOT)," International Journal of Engineering Technology Science and Research, IJETSR, www.ijetsr.com, ISSN 2394 - 3386, Volume 4, Issue 5, May 2017

[4] Pratiksha Jain, Neha Chopra, Vaishali Gupta, "Automatic License Plate Recognition using OpenCV," International Journal of Computer Applications Technology and Research Volume 3– Issue 12, 756 - 761, 2014, ISSN:-2319–8656

[5] Abhirup Khanna, Rishi Anand, "IOT Based Smart Parking System," 2016 International Conference on Internet of Things and Applications (IOTA), Maharashtra Institute of Technology, Pune, India 22 Jan - 24 Jan, 2016

[6]"Openalpr/openalpr."GitHub.
https://github.com/openalpr/openalpr

[7] Mr Basavaraju S R, "Automatic Smart Parking System using Internet of Things (IOT)," International Journal of Scientific and Research Publications, Volume 5, Issue 12, December 2015, ISSN 2250-3153