IoT Parking Apps with Car Plate Recognition for Smart City using Node Red

Smart City, IoT, Node Red

I. INTRODUCTION

Nowadays, with the growing number of vehicles on the road, it has resulted in an unexpected civic problem which solution, a car plate recognition system is developed and plays a major role in addressing these kinds of issues. The car plate recognition system gave us many applications from parking admission, monitoring urban traffic and to tracking automobile thefts, etc.

The proposed system aims to develop a gate system that applies the car plate recognition technologies. We will

Presnted by

Suryaprakash. S
Devaki. K
Adhilakshmi. S
Divya. G
Saranya. R

III. RESULTS AND DISCUSSIONS

A. Prototype Design

In the development of the project, the developer built the prototype to test a concept or features to meet the requirement of the system. Prototyping is a term that is used in a variety of contexts that includes the process of mechanical design, hardware design, and software design. A prototype, as shown in Fig. 2, is usually used to evaluate the new design in enhancing the system measurements by users and system analysts.

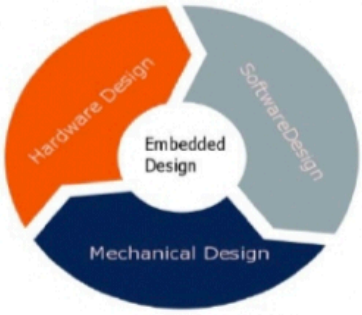

Fig. 2. Prototype Development Process.

a) Mechanical Design

In this project, the mechanical design that implemented by the developer is to use a wood board to place the system components. All the components that are used in the project have been glued to the board so that all the components remain intact with the exception of the raspberry pi camera that is connected to the raspberry pi 3 B+ as shown in Fig. 3.

The design will also use a passive infrared motion sensor placed near the servomotor to detect movement apart from an LCD and Bluetooth Module HC-06 that is placed behind the servo motor. Lastly, a Red LED is placed on the board and connected to the Raspberry Pi for the verification process, as shown in Fig. 4.

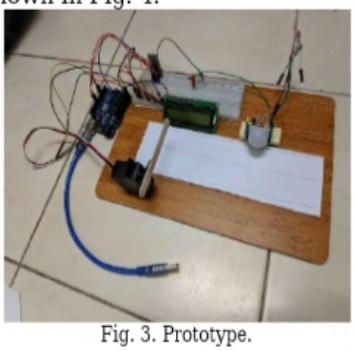

Fig. 3. Prototype.

Fig. 4. Camera Connected to Raspberry Pi 3 B+.

B. Hardware Design

a) Arduino Uno R3

In the hardware design of the prototype, a microcontroller called Arduino Uno R3 is used to connect Digital and Analog (I/O) pins on the board to components that are the servo motor, LCD and the Bluetooth Module as shown in Fig. 5. When all of these components are successfully connected to the microcontroller, the developer can begin to write the program for the connected components.

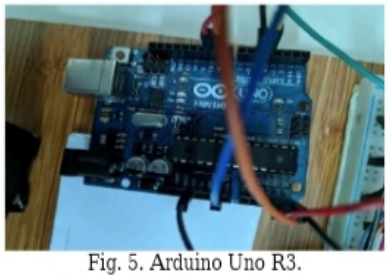

Fig. 5. Arduino Uno R3.

b) Raspberry Pi 3 B+

In this project, the developer uses Raspberry Pi 3 B+ to implement the recognition process of the car number plate. There are multiple components that will be connected to Raspberry Pi 3 B+. These components in Fig. 6, consist of a PIR motion sensor, an LED and the raspberry pi camera that is used to capture the front part of the car and the verification process.

Fig. 6. Raspberry Pi 3 B+.

c) Passive Infrared Motion Sensor at Entry Parking

The PIR motion sensor in Fig. 7. is a sensor that is able to detect motion of nearby objects between 5m and 12m without physical contact. The PIR motion sensor is used to detect the car and send a signal to the camera to capture the image of the front view of the car. A servo motor is used to represent the barrier gate of the system, and as the PIR motion sensor detects the presence of a car near the entry gate, it will activate the recognition process of the system. Once the car plate number is verified, an LED will be used to indicate the process.

Fig. 7. Passive Infrared Motion Sensor at Entry Parking.

d) Servomotor at Prototype

A servomotor function as a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. A servo motor can usually only turn 90° in either direction for a total of 180° movement. In this project, the servo motor is connected to the Arduino Uno R3 to represent as a barrier gate for entry parking. After the car number plate verification process, the barrier gate will be open manually by the admin using a Bluetooth connection through an app that can only be accessed by the admin. In Fig. 8. below, the servomotor is being used for the system gate.

Fig. 8. Servomotor represent as barrier gate.

e) Bluetooth Module HC-06 for controlling parking apps

The HC-06 is a class 2 slave Bluetooth module used in Fig. 9 is designed for transparent wireless serial communication. In this project, the Bluetooth module is connected to Arduino Uno R3 and be paired with an android device to send the user's account's balance fee commands from the parking app to the Servo Motor and LCD.

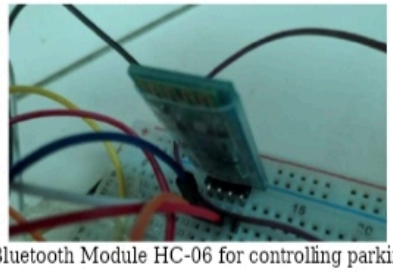

Fig. 9. Bluetooth Module HC-06 for controlling parking apps.

f) Liquid Crystal Display (LCD) for display user information

An LCD is an electronic display module in which it uses a liquid crystal to produce a visible set of characters. The 16x2 LCD display is a device that is commonly used in circuits in which it translates a display 16 characters per line in 2 such lines. Within this prototype, the LCD is placed on the board and connected to Arduino Uno R3 with a function to display customer parking fees and balance from the android app. Fig. 10. below shows the placement of the LCD on the prototype board.

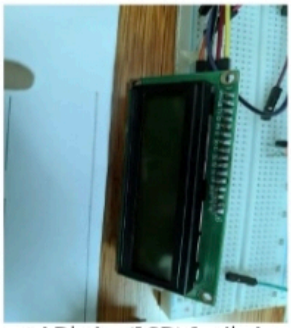

Fig. 10. Liquid Crystal Display (LCD) for display user information.

g) Raspberry Pi Camera Module v2

The Raspberry Pi Camera Module is a high-quality camera that features an 8 megapixel Sony IMX219 image sensor with a fixed focus lens. This camera is capable of 3280 x 2464 pixel static images with the support of a 1080p30 video stream. In the prototype, the raspberry pi camera module is used to capture the front view of approaching cars for the verification process of the plate numbers. The raspberry pi camera module in Fig. 11, is attached to the raspberry pi via a short ribbon cable to the small socket on the raspberry pi board which is designed specifically for interfacing with the camera.

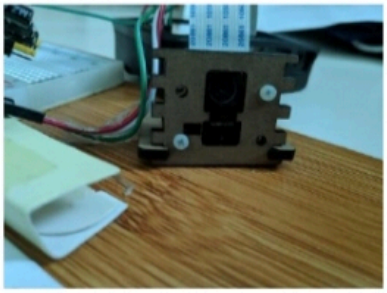

Fig. 11. Raspberry Pi Camera Module v2 for taking pictures of the car.

C. Software Design

a) Node-RED

Node-RED is low-code programming for event-driven applications tool that is used to wiring together hardware devices, APIs and online services in new and many interesting ways. Nowadays, Node-RED is basically an open-source visual editor that is used by many developers for the internet of things applications. The Node-RED system contains what we called "Nodes" which we can drag and drop on the visual editor and wire together. There are many varieties of Nodes that offer different functionality that ranges from a simple debug node through a Raspberry Pi node that allows the developers to read and write to the GPIO pins on the Raspberry Pi.

Fig. 12. Node-RED

In this project, the developer design a Node-RED flow used in identifying and to verify the car number plates. The Node-RED flow will be integrated with the Raspberry Pi GPIO pins where the verification process shall take place. For the car plate verification process, the Node-RED flow will control the devices, which are the Passive Infrared Motion Sensor, LED, and the Raspberry Pi Camera v2 that is connected to the Raspberry Pi. For this project, the developer has designed the flow that is used for the car plate number recognition process. In the designed flow, each of the nodes has different functions and plays a huge role in keeping the Node-RED flow working as intended. Fig. 13. below shows the complete working node-RED flow and all its nodes that were developed for this project.

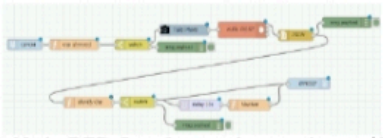

Fig. 13. Node-RED flow for car plate number verification

As shown in Fig. 14. below the "GPIO4" is used to control the PIR motion sensor that is connected Raspberry Pi GPIO pin number 4. This node will receive a digital input signal from the PIR sensor in the form of 0s and 1s input.

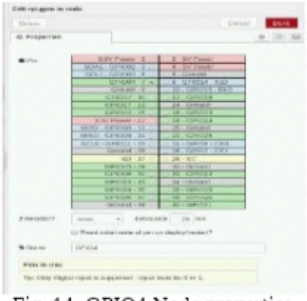

Fig. 14. GPIO4 Node properties

In the "Car detected" node shown on Fig. 15. below, it will receive the digital input signal from the GPIO4. If the input is 1, it will proceed onto the "Take Photo" node; else the flow will pass on to "msg.payload" node and "0" output" message will be displayed on the debug dashboard.

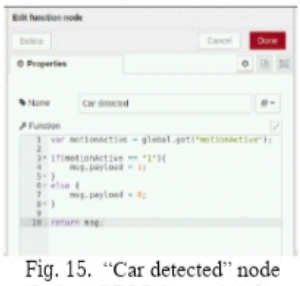

Fig. 15. "Car detected" node

If the input of the GPIO4 is 1, the flow signal will activate the "Take Photo" node. This node communicates with the Raspberry Pi Camera Module that is connected to the Raspberry Pi board. Fig. 16. below shows the properties of the "Take Photo" node such as the location path of the captured photos and the quality of the photos. For this project, the developer use 800x600 image resolution for the photos taken to keep save Raspberry Pi memory card spaces.

Fig. 16. "Take Photo" node properties

In the "cURL POST" node shown in Fig. 17., the main function of this node is to request a connection to the ALPR CloudAPI for the image recognition process. If the connection is successful, the flow will then proceed to the "Identify Car" node where in this node, the image that was taken earlier by the camera is analyze by the ALPR software and will produce the plate number results. These results are then will be compared to the registered number plate that is manually put in by the system admin, as shown in Fig. 18. below.

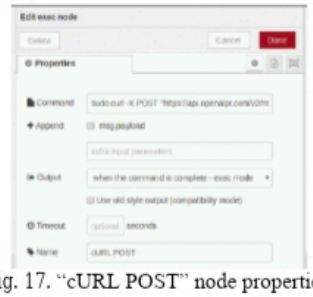

Fig. 17. "cURL POST" node properties.

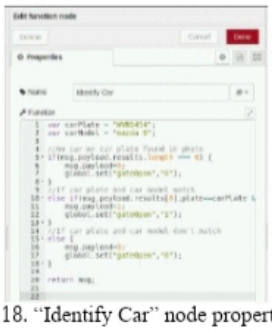

Fig. 18. "Identify Car" node properties.

In both Fig. 19. and Fig. 20. below, these two nodes are used to receive the input from the "Identify Car" node. The plate number results match with the one within the system, an input value '1' will be passed on the GPIO17 node wherein these nodes, and the LED will be active for the duration of 10s. If the plate recognition results came back as false, an input value '0' will be passed on the GPIO17 node and the LED will not be activated, and a message '0' will be displayed on the debug dashboard.

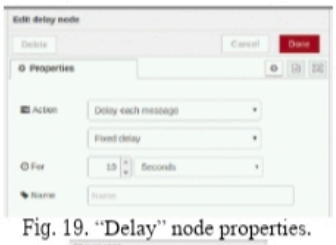

Fig. 19. "Delay" node properties.


Fig. 20. "GPIO17" node properties

b) Arduino IDE (Integrated Development Environment)

Arduino Uno R3 is considered to be an open-source platform that is used for building many electronics and the internet of things projects. Arduino Uno R3 is a microcontroller that consists of both physical programmable circuit board as well as a piece of software called Arduino IDE (Integrated Development Environment) that runs on a computer. This software is basically used to write and upload computer code to the Arduino Uno R3 physical board. Arduino IDE is open-source software that can be easily downloaded from the official Arduino Uno website. This software is quite convenient to use due to the fact that it is popular among developers that provide many reference codes on the internet. For this project, the developer is using the latest version of the software, which included in the figure below.


Fig. 21. Arduino IDE 1.8.10

The developer of this project will use the Arduino IDE to develop the app and divided the program into two parts. The first part of the program is used to control an app that can control the servomotor as the gate. The second part is to control the LCD that can display the balance fees of the customer's account. Lastly, this program also used the Bluetooth module HC-06 that sends the commands from the smartphone app to the devices.

1. Miit App Inventor

MIT App Inventor is the web application platform that integrated development environment originally provided by Google. This platform allows newcomer developers to computer programming to create application software or apps. MIT App Inventor is a free and open-source software that is released to the public and does not need to be downloaded. In this project, the developer will create two apps through the MIT App Inventor online website. The first app is to control the servo motor gate through the Bluetooth connection. While the second app is only to make payments.

Fig. 22. Mit App Inventor.

2. Servo Motor App to control gate

For this project, the developer has developed an app that can communicate with the servo motor. By using the Bluetooth Module, the app can be connected to the system to send commands to the Bluetooth Module. The module will then transfer the signals to the servo motor to open the gate. This app is only accessible to the admin; in which they are responsible to open the gate if the LED indicator connected to the Raspberry Pi implies that the car plate number is verified in the system.

Fig. 23. Servo Motor App

3. Parking Payment App

For the visitors, the developer has developed an online parking payment app that parking lot visitor is required to download and install from the app store. The visitor will need to register their account before start using the payment app. Once registered, the visitor will give an option to top up their account balanced to pay for the parking fees. The way this app function is that it is using the Bluetooth connection to connect to the Bluetooth Module on the system. The visitor needs to open their Bluetooth connection toggle on their smartphone in order to pay for the parking fees before entering the parking lot. If the Bluetooth module detects the Bluetooth connection on the phone, the charging fees will be automatically deducted, and the visitor account's balance will be display on the LCD.

Fig. 24. Parking Payment App login page

Fig. 25. Reload balance page.

D. Project Testing

During the prototype phase, the developer has combined the required components and connected all the circuits. In this prototype testing, the developer has run the system multiple times. During those runs, any error that occurred the developer will troubleshoot and come up with an idea on how to make the system work. As shown in figure 4.20 below, that is the picture of the car number plate that has been printed on a piece of A4 paper. The picture was taken outside of the developer's home area with the permission of the car owner. In this project, the paper will represent an actual car that will approach the gate. The number plate of the vehicle in the picture has been already put within the Node-RED database system.

Fig. 26. The car image above is representing the car plate number that will be used for the plate number verification testing.

When the PIR motion sensor detects motion within its range, it will send a signal to the Raspberry Pi Camera Module to automatically take photos of the object that triggers the sensor.

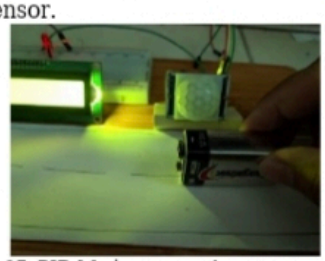

Fig. 27. PIR Motion sensor detects movement.

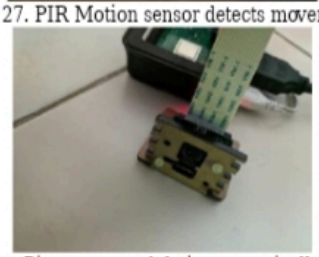

Fig. 28. Raspberry Pi camera module is automatically activated to take photos

Fig. 29. The photos of the license plate are being captured by the camera.

When the camera captures the picture, the system will start to do the plate number recognition process by using ALPR technology. It will compare the number plate results to the number plate that is registered in the system. If the number plate results are the same as in the system, it will activate the red LED for the duration of 10 seconds verifying the car plate number is the same as in the system as shown in Fig. 30. below.

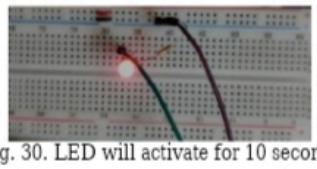

Fig. 30. LED will activate for 10 seconds.

The admin will then proceed to manually open the gate by pressing the button on the Bluetooth app to allow the visitors to pass through the gate, as shown in Fig. 31. below.

Fig. 31. The gate will be open to letting the car passes through.

During entering the parking, the Bluetooth module will detect the visitors parking payment apps and will automatically display the charging fees and remaining balance on LCD at the gate, as shown in Fig. 32. below.

Fig. 32. Parking fee charges and visitor account's balance

E. Project Results

The results for this project can be seen at the node-RED dashboard. The prototype testing was conducted a few times as there was an error that occurs in the node-RED flow, where the plate number verification process takes time to return a working result.

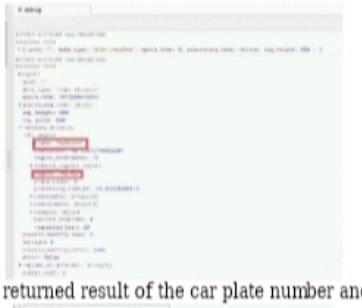

Fig. 33. The returned result of the car plate number and country.

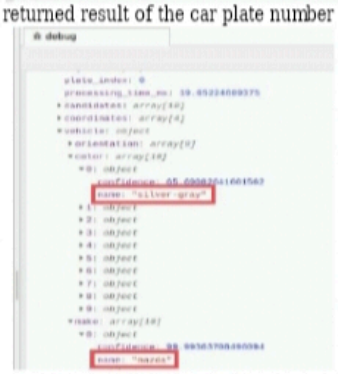

Fig. 34. The returned result of the colour and model of the car.

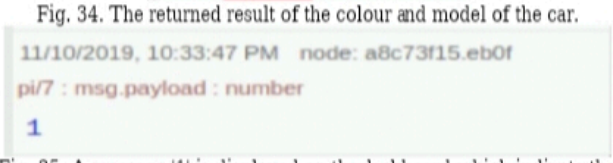

Fig. 35. A message '1' is displayed on the dashboard which indicate the car is verified.

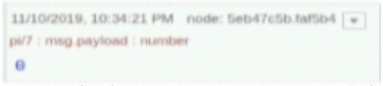

Fig. 36. A message '0' is displayed on the dashboard which indicate the car is not verified.

IV. CONCLUSIONS

The main objective of this project is to detect the car plate number using ALPR recognition technology and to provide an app for the admin to control the gate from afar away without having to sit right next to the gate itself. Other than that, the other main goal for the system is to create a platform for users to pay the parking lot fees through an online smartphone app. Raspberry Pi image processing technology can be applied to this project. One of the technology is facial recognition that can be used to provide more security into the system. With facial recognition, the system can keep track and record visitor faces and stored in the database.

[5] MAGNET Security & Automation Sdn. Bhd. 2013, Our Solutions - Long Range Parking Access System. Retrieved from https://www.magnet.com.my/our-solutions/long-range-parking-access-system.

[6] M-Tech Innovations Ltd. 2013, RFID Vehicle Access Control System. Retrieved from https://www.indiamart.com/proddetail/rfid-vehicle-access-control-systems-10638602230.html

[7] Puranic, Aniruddh & K, Deepak & V, Umadevi. (2016). "Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching". International Journal of Computer Applications.

[8] TimeTec Sdn. Bhd. 2019, i-Neighbour Smart Barrier GateTimeTec BLE-2 System. Retrieved from https://www.i-neighbour.com/TimeTecBarrier

Creating a smart parking IoT project using Node-RED involves several steps. Here's an outline along with some example code snippets:

1. Setting up Node-RED:

- Install Node-RED on your system or IoT device.
- Install necessary packages and dependencies.

2. Hardware Setup:

- Set up the necessary sensors (like ultrasonic sensors) in the parking spots.
- Connect these sensors to your IoT device (e.g., Raspberry Pi).

3. Writing the Flow:

In Node-RED, create a flow to handle the sensor data and control the parking status. Here's an example flow:

```json
[{"id":"dab5a248.2644a8","type":"ui_gauge","z":"dbed8b49.aef898","name":"Parking Status","group":"9c2a7697.dbb33","order":0,"width":0,"height":0,"gtype":"gage","title":"Parking Status","label":"","format":"{{value}}","min":0,"max":"10","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":600,"y":320,"wires":[]},{"id":"f17aa1f9.82d34","type":"function","z":"dbed8b49.aef898","name":"Generate Random Data","func":"msg.payload = Math.floor(Math.random() * 11);\nreturn msg;","outputs":1,"noerr":0,"x":380,"y":320,"wires":[["dab5a248.2644a8"]]},{"id":"d26236ac.35f7d","type":"inject","z":"dbed8b49.aef898","name":"","props":[{"p":"payload"},{"p":"topic","vt":"str"}],"repeat":"5","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":200,"y":320,"wires":[["f17aa1f9.82d34"]]},{"id":"9c2a7697.dbb33","type":"ui_group","name":"Parking","tab":"14bb5c32.f36d82","order":1,"disp":true,"width":"6","collapse":false},{"id":"14bb5c32.f36d82","type":"ui_tab","name":"Home","icon":"dashboard","order":1,"disabled":false,"hidden":false}]
```

Thank you