© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Smart Parking System Mobile Application using Ultrasonic Detector

Fatin Areena A Aziz[1], Siti Zarina Mohd. Muji[2*], Mohd Helmy Abd Wahab[2], Zarina Tukiran[2], C. Uttraphan[2], Nor'aisah Sudin[2]

[1]Axis Technology Resources (M) Sdn Bhd,
No. 27 Jalan Kempas 1, Taman Desaru Utama, Bandar Penawar, 81930, MALAYSIA

[2]Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2022.14.03.008
Received 16 June 2021; Accepted 04 June 2022; Available online 20 June 2022

Abstract: Finding an empty parking lot is a big challenge as it can take lots of time and resources. This paper describes the design and development of a prototype of a smart parking system for assisting the vehicle drivers who always experiencing problem in finding indoor parking lots. The objectives of the proposed system are to develop a IoT parking lot system utilizing ultrasonic sensor to detect the presence of a vehicle and to develop a mobile application to display the parking lot layout for assisting car drivers to locate an empty parking lot. For the controller unit, it consists of the Arduino UNO microcontroller which is programmed using JavaScript language. Arduino is used to control the sensor, build a connection with the firebase server and update any changes in the hardware to the firebase. This enables the ultrasonic transmitter and collects the sensing data to be passed to the data transmission and acquisition unit. Once the ultrasonic sensor detects a car, the signal is read and processed by the microcontroller before it is sent to the server to update the status of the parking availability. The accuracy of vehicle detection is 100% with a standard deviation for sensor A0 and sensor A1 are 0.5 and 0.32 respectively. The display result of the parking lot availability is automatically updated in the mobile phone application in real-time.

Keywords: Arduino, mobile phone application, parking system, ultrasonic, IoT, firebase

1. Introduction

The fourth industrial revolution 4.0 is leveraging digital technologies and real-time data to improve efficiency and cut costs. The trend is clear that one of most popular concept that utilizing most IR4.0 component is a smart city. There are many components in smart cities ranging from waste management to transportation including managing or monitoring parking system or smart parking system. Vehicle parking is an important place could be used by driver either in the building or outside building such as hospitals, residential government, shopping mall and stadium [1]. Car parking is known as a convenient parking spot for car owners to park without stopping traffic and will thus prevent interrupting other vehicles. Almost every building provides a parking lot so that the customer can easily park the car in the building.

Instead of providing a parking area in the building, the most important thing that will be offered in the city is the large amount of parking spaces according to the owner of the building and the easy place for them to park their car. For car drivers, they can find another alternate way to find a parking space where they can leave their vehicle, particularly when they're in a hurry. In order to save time, they often push a double parking lot and cause hassle to others.

Nowadays, several premises provide the facilities to assist drivers in locating an open parking spot where the driver

*Corresponding author szarina@uthm.edu.my

can check for an LED indicator board or a parking assistance panel that displays the amount of vacant parking spaces as seen in Figure 1. The board provides two information which are the number of empty parking spaces and the direction (whether left, right or forward) of the alleyway which to the empty parking spaces.

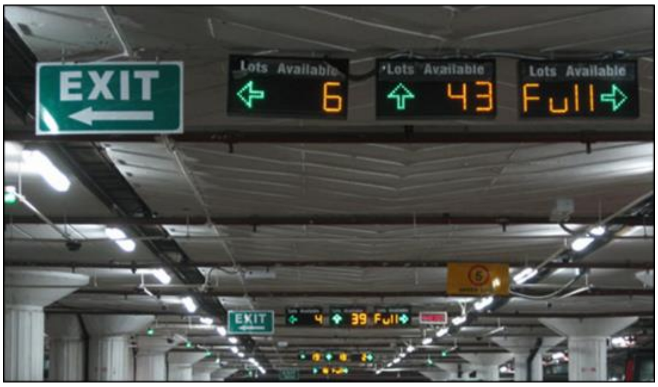


Fig. 1 - LED display board

This work aims to develop the Internet of Things (IOT) system parking using a mobile application that can detect the exact parking location. It uses firebase as a cloud system to update any changes in the hardware (sensor). The advantage of this application is that it will help drivers locate a vacant parking spot in a big building from a long distance, reducing quest time dramatically. The used of ultrasonic sensor has more advantage compare to other sensor such as infrared sensor. Infrared sensors have other drawbacks, such as their inability to use them in sunlight due to interference. This can hinder outdoor applications or applications in dark indoors. Ultrasonic sensors operate with sound waves, and as many variables do not impact the identification of obstacles. This make ultrasonic more reliable than infrared sensor.

2. Related Works

There are several benefits to IoT in applications such as monitoring actions, improved situational awareness, sensor-driven decision making and rapid control and reaction, where most sensors are linked to GPS and Wi-Fi [2]. The smart parking system is more focusing on less power consumption and more performance device so Raspberry pi is chosen in [2] as it has suitable hardware microprocessor that supports IoT for the implementation. However, the use of pi-camera make this project a little bit complicated as it has to handle a huge amount of image data.

The cloud-based IoT design for the smart parking infrastructure requires a cloud service that uses online storage to store details on accessible parking space in the parking area [3]. The centralized server is used to monitor details on the smart parking network that displays the amount of slots and the supply of free spaces. According to Abhirup Khanna and Rishi Anand, an IoT device is used as identity communication devices [3]. Devices may be managed, tracked or monitored using remote computers that are linked to the Internet. Cloud is a successful IoT collaborator as it serves as a portal where all sensor data can be viewed from distant locations. The IoT can be described as:

Physical object + controller, sensor and actuators + Internet = Internet of Things.

Abhirup Khanna and Rishi Anand have developed the smart parking system by using a mobile application that connected through the cloud. The device allows the customer to learn the location of a parking slot in real time, so that people can book a parking slot using the mobile application [3]. However, this project requires ESP8266 WiFi chip to make the ultrasonic wirelessly connected to the raspberry pi, which sometime could give error reading if the internet connection have a problem. In [4], the same technique applied by wirelessly connected the sensor with raspberry pi. Compared to our method, we are connected each of ultrasonic sensor to the host pc where its connection is more stable even inside the building.

Meanwhile, Chungsan et al. introduced the IoT as a smart parking sensor concept that uses middleware between users and terminal device [5]. They use Bluetooth and USIM (Universal Subscriber Identity Module) ID to locate the service of parking car. It also sends mode ID and USIM ID to server through Zigbee with gateway. They are then provided with the USIM ID of the user's smartphone through Bluetooth communication, so that these data are used for the parking area location service. Figure 2 displays the flow chart of the smart parking network where PKSensorMote controls the ultrasonic sensor and the magnetic sensor. The PKSensorMote transmits the collected data via the BaseStation module to the gateway. The radio communication collects and sends the data to the LotParking Control module on the host PC. The Lot Parking Monitor is used to record the database obtained and track the parking area in real time. The smartphone application communicates via Bluetooh with sensor motes. middlewareHowever the use of bluetooh communication permit the coverage of transmission and receiption of data compared to our system which use hos PC to connect to each sensor which more reliable and make the user can monitor the parking area from far of the parking location.

In [6], the system seems complicated as it use two microcontrollers to operate the parking system. In [7], the use of drone for getting the available parking system information is a new effort compare to other researcher. Nevertheless, both strategies may raise the operational costs of the parking system.

The propose method is by using a host PC that act as bridge between the ultrasonic sensor and firebase cloud, where the user can trace the parking lot available at the android smartphone.

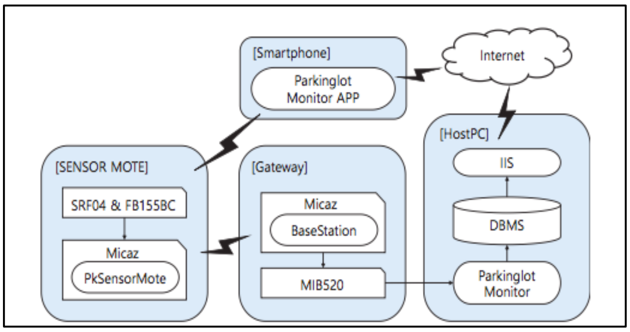


Fig. 2 - Smart parking system flow chart [4]

3. Objectives

System development includes any equipment that is used to build the IoT system which contain three main parts which is sensor, communication and cloud. For the sensor unit, it consists of ultrasonic sensor which purpose to sense the presence of car in each parking lot. It is controlled by Arduino Uno that become the controller unit. To send the sensor data from Arduino to the cloud, the host which have internet connection will act as the bridge between sensor and the cloud. The cloud used is firebase cloud.

3.1 Visual Studio Code

The visual studio code is a source code editor, developed by Microsoft for that supports Window, Linux and MacOS Operating Systems. This software supports various programming languages including JavaScript. It supports debugging, embedded Git control, syntax highlighting, intelligent code completion and code refactoring. The HC-SR04. If the red LED is switched on, the ultrasonic sensor detects the object and space is filled, but when the green LED is switched on, the parking space is still open.

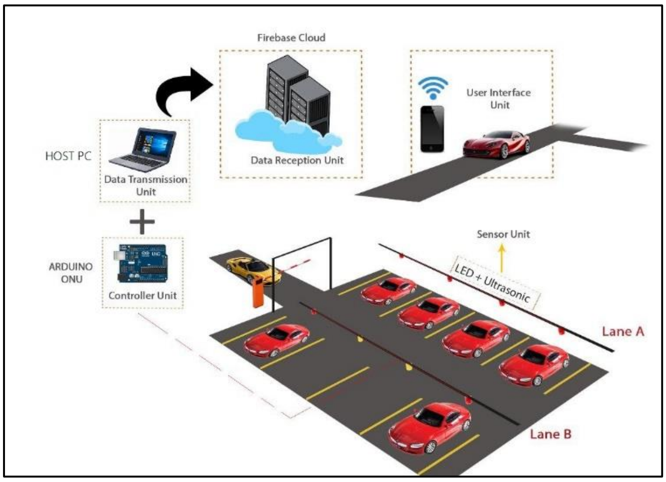


Fig. 3 - Parking lot system overview

For the controller unit, it consists of the Arduino UNO microcontroller that is used to control the sensor, build a connection with the firebase server and update any changes in the hardware to the firebase. This activates an ultrasonic sensor and collects the sensing data to be passed to the data transmission and retrieval unit. Figure 4 shows the flowchart of the communication between the Arduino and firebase server.

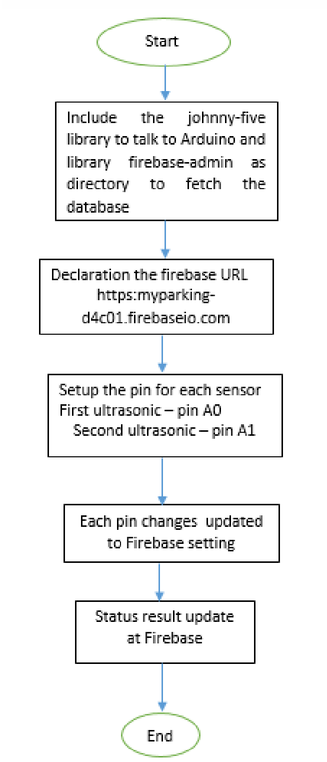


Fig. 4 - Flowchart between the Arduino and firebase server.

First, the required libraries, which are the Johnny-five and Firebase-admin must be applied by the microcontroller as shown at Figure 5.

```
JS server.js ×
const five = require('johnny-five'); // lib to talk to arduino
const admin = require('firebase-admin'); //lib database

const board = new five.Board();
const thresholdVal = 1
```

Fig. 5 - Library fetch for johnny-five and firebase-admin.

Johnny-five library is the library for Arduino interfacing while firebase-admin is used to access the database. Then the Firebase's url must also be declared as shown in Figure 5. The ultrasonic sensors are connected to the microcontroller's pin, A0 and A1. Any modifications to the ultrasonic status will be changed in the Firebase.

```
var serviceAccount = require('./myparking-d4c01-firebase-adminsdk-en51j-983dd86d1f.json');

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://myparking-d4c01.firebaseio.com"
});

```

Fig. 6 - Firebase URL

The ultrasonic model used is HC-SR04 where it comprises both transmitter and receiver on one side. The ultrasonic sensor measures the distance using the sonar; the ultrasonic pulse is transmitted from the unit and the distance to the target is determined by measuring the time required for the return of the echo. Output from the sensor is a variable-width pulse corresponding to the distance to the target.

The data transmission and acquisition unit transfer the data of accessible parking slot from the controller to the cloud system using Firebase cloud.

4. Mobile Application Unit

The mobile application is used to display the parking lot system in this work. The mobile application unit receives the data from the cloud. The mobile application is developed using Javascript. Both details on the position of the parking slot and the capacity of the parking slot will be shown on the cell phone.

5. Result and Analysis

This segment presents the finding of the hardware connection and the software development of the proposed Smart Parking System. The results include the otput from mobile phone application and hardware part especially red and green LEDs, as well as measurement of system efficiency. Figure 7 shows the system prototype that connected to the server and the location of the LEDs.

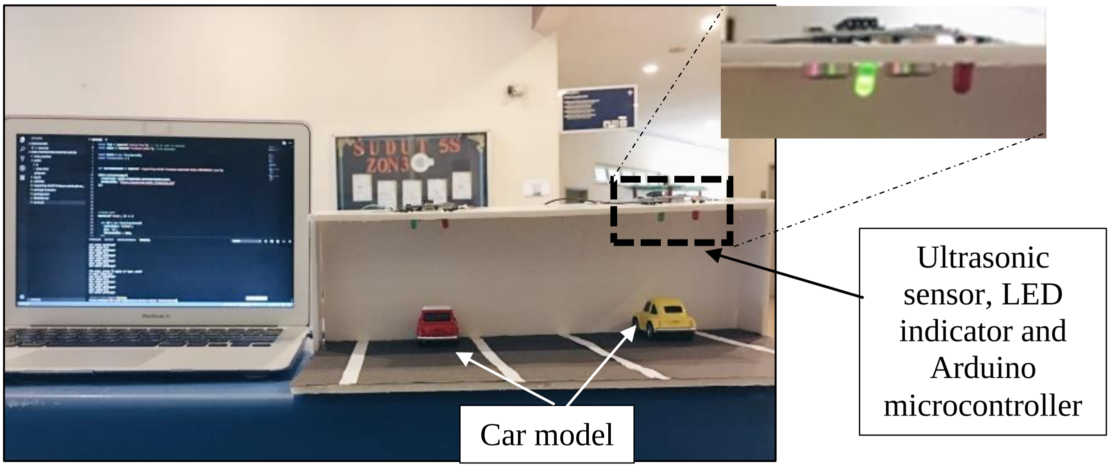
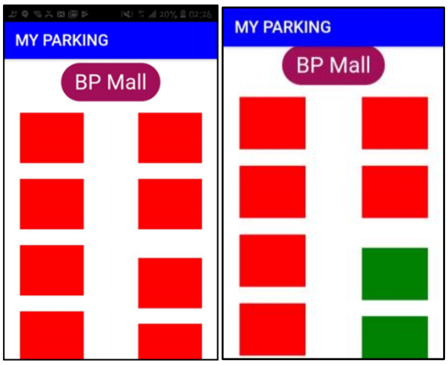


Fig. 7 - The prototype of the proposed system

In Figure 8(a), the design of mobile phone application has not connected to the server, hence, the colour of the indicators will be in red colour by default. Once the system is connected to the server, the indicator with green colour indicates the parking is available as shown in Figure 8(b).



(a) (b)

Fig. 8 - Design of the mobile phone application

Once the parking lot is occupied at lot A1, the green LED will turn off and the red LED will turn on as illustrated in Figure 10. At the same time, the mobile application is updated as shown in Figure 10.

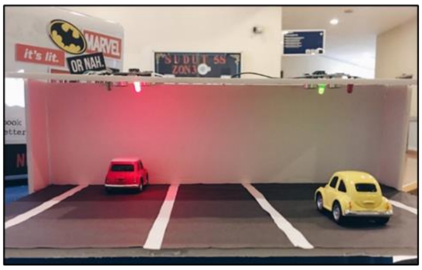


Fig. 9 - A vehicle occupied lot A1 and the LED turns into red

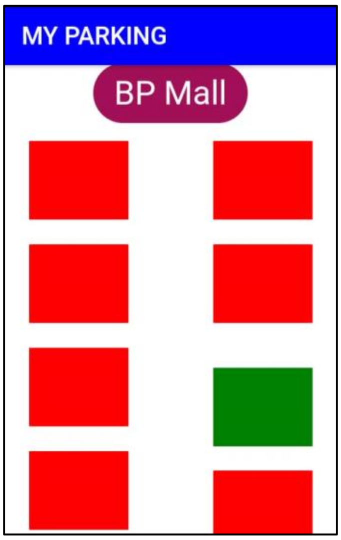


Fig. 10 - The mobile application shows the lot A1 is in green colour, indicates the lot is empty

The system performance is measured for 30 times for each parking lot to get the accuracy of the distance from ultrasonic sensor and car with adjusted below 10 cm distance. Figure 11 and 12 show the repetability of ultrasonic sensor measurement for 30 tests. The measurement indicates that there is a small variation between each measurement. It occurs when the sensor detects the different shape of the vehicle, and also when the car is parked in various locations. When there is no car park at the parking lot, the measured value is 12.1 cm.

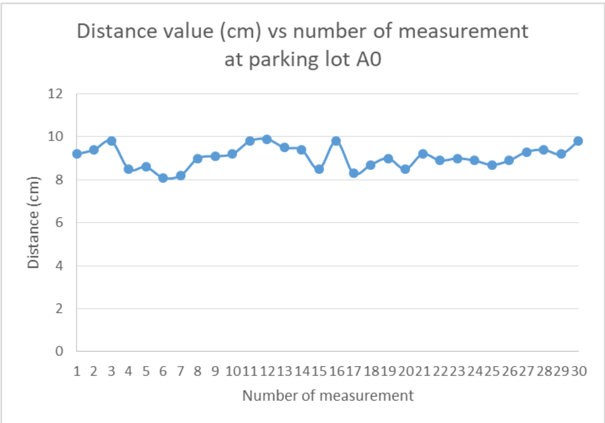


Fig. 11 - Repeatability of distance at parking lot A0 (cm)

Table 1 and 2 show the distance between the sensor and the car for parking at A1 and A0 are 9.03 cm and 9.06 cm respectively. The average value has been calculated from 30 different measurements. The results for A1 and A0 are plotted in the graph in Figure 13 and 14 respectively.

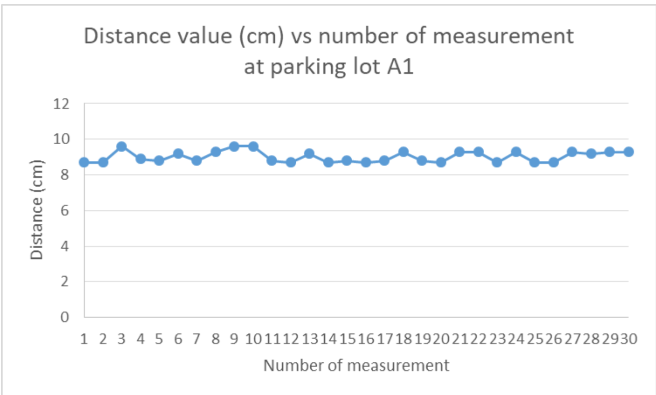


Fig. 12 - Repeatability of distance measurement at parking lot A1

Table 1 - The different between actual distance and measured by sensor at parking lot A1

Time car parked at lot A1	Distance measured by sensor(cm)	Actual Distance (cm)	The different between actual distance and measured by sensor
1	8.7	9.03	-0.33
2	8.7	9.03	-0.33
3	9.6	9.03	0.57
4	8.9	9.03	-0.13
5	8.8	9.03	-0.23
6	9.2	9.03	0.17
7	8.8	9.03	-0.23
8	9.3	9.03	0.27
9	9.6	9.03	0.57
10	9.6	9.03	0.57
11	8.8	9.03	-0.23
12	8.7	9.03	-0.33
13	9.2	9.03	0.17
14	8.7	9.03	-0.33
15	8.8	9.03	-0.23
16	8.7	9.03	-0.33
17	8.8	9.03	-0.23
18	9.3	9.03	0.27
19	8.8	9.03	-0.23
20	8.7	9.03	-0.33
21	9.3	9.03	0.27
22	9.3	9.03	0.27
23	8.7	9.03	-0.33
24	9.3	9.03	0.27
25	8.7	9.03	-0.33
26	8.7	9.03	-0.33
27	9.3	9.03	0.27
28	9.2	9.03	0.17
29	9.3	9.03	0.27
30	9.3	9.03	0.27

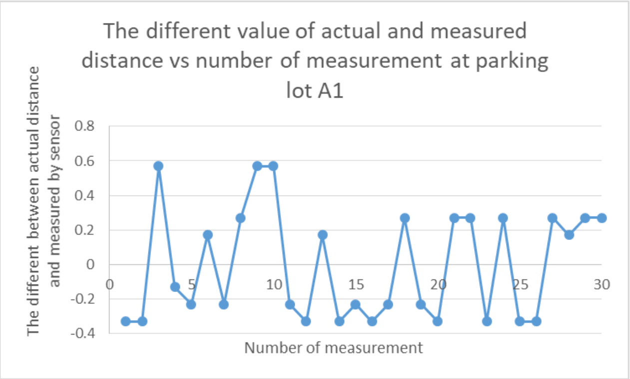


Fig. 13 - Plot of the different between actual and measured distance of parking lot A1

Table 2 - The different between actual distance and measured by sensor at parking lot A0

Time car parked at lot A1	Distance measured by sensor(cm)	Actual Distance (cm)	The different between actual distance and measured by sensor
1	9.2	9.06	0.14
2	9.4	9.06	0.34
3	9.8	9.06	0.74

4	8.5	9.06	-0.56
5	8.6	9.06	-0.46
6	8.1	9.06	-0.96
7	8.2	9.06	-0.86
8	9	9.06	-0.06
9	9.1	9.06	0.04
10	9.2	9.06	0.14
11	9.8	9.06	0.74
12	9.9	9.06	0.84
13	9.5	9.06	0.44
14	9.4	9.06	0.34
15	8.5	9.06	-0.56
16	9.8	9.06	0.74
17	8.3	9.06	-0.76
18	8.7	9.06	-0.36
19	9	9.06	-0.06
20	8.5	9.06	-0.56
21	9.2	9.06	0.14
22	8.9	9.06	-0.16
23	9	9.06	-0.06
24	8.9	9.06	-0.16
25	8.7	9.06	-0.36
26	8.9	9.06	-0.16
27	9.3	9.06	0.24
28	9.4	9.06	0.34
29	9.2	9.06	0.14

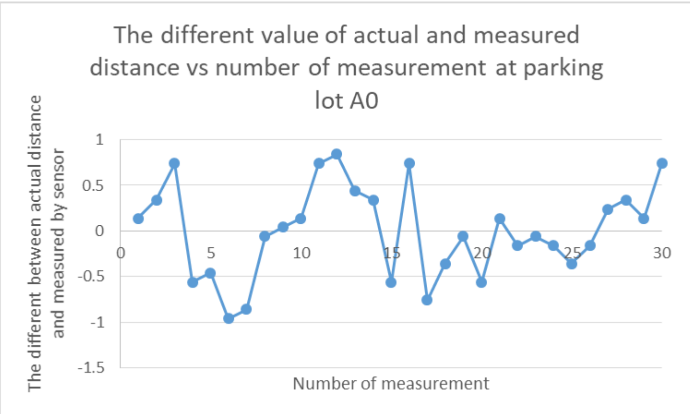


Fig. 14 - Plot of the different between actual and measured distance of parking lot A0

From Equation (1), the standard deviation for the measurement accuracy is 0.32 with average of 9.03 cm for lot A1 and 0.5 with average 9.06 cm for lot A0. It means that the response at A1 lot for this value scattered around mean in range 0.32 and between 8.5 to 9.5. The greater the standard deviation value, the more distributed the results, which means that the greater standard deviation value indicates that the data validity is not reliable. For A0, the data was dispersed between 9.0 and 10.0. Standard deviation represents the dispersion of a data set relative to its mean and is calculated as the square root of the variance. If the data points go farther from the mean, there is a higher deviation within the data set, which implies that the further the data stretches, the larger the standard variance.

Standard deviation is given by:

$$S_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \qquad (1)$$

n = the number of data points
x_i = each of the values of the data
$\bar{x}$ = the mean of x_i

Table 3 shows, the detection accuracy is 100% for 30 measurements. The detection latency is 0.01s, means the response time is considered fast. Meanwhile, the flag in the code for detecting the available parking lot will be updated in Firebase from “false” to “true” when the parking lot is still available.

Table 3 - The accuracy of hardware and software for car detection

Testing	Detection of car	
	Hardware	Software
1	√	√
2	√	√
3	√	√
4	√	√
5	√	√
6	√	√
7	√	√
8	√	√
9	√	√
10	√	√
11	√	√
12	√	√
13	√	√
14	√	√
15	√	√
16	√	√
17	√	√
18	√	√
19	√	√
20	√	√
21	√	√
22	√	√
23	√	√
24	√	√
25	√	√
26	√	√
27	√	√
28	√	√
29	√	√
30	√	√

6. Conclusion

As a conclusion, this work success to build IoT system for parking lot prototype. The code for Arduino that write using Visual Studio code software by include the desired library for Arduino and Firebase could help to reduce the use of multiple platform in this work. The prototype is working properly and can communicate with the Firebase smoothly. For accuracy, it shows 100% result of car detection with standard deviation for A1 and A0 sensors are 0.32 and 0.5 respectively. The proposed system can be enhanced with some other features for future development. Firstly, this system can include the GPS service in the user applications such that the user can know the distance to the parking lot in real-time. This service will demonstrate the way and lead the drivers to their chosen parking lot. This system can also be upgraded with the booking service feature where customer can reserve their preferred parking lot in advance.

Acknowledgment

This work is supported by the FRGS K074 and MDR H499 research grants.

References

[1] Nammoon Kim, Changqiang Jing, Biao Zhou and Youngok Kim. Smart Parking Information System Exploting Visible Light Communication, Vol 8, No 1, 2014.

[2] S. R., Basavaraju. Automatic Smart Parking System using Internet of Things(IOT), *International Journal of Scientific and Research Publications*, vol. 5, no. 12, December 2015.

[3] Abhirup Khanna and Rishi Anand. IOT based Smart Parking System, *International Conference on Internet of Things and Applications*, Jan 2016.

[4] Metilda Florence, S, Uma, M., Fancy, C., Saranya, G. A study of remotely booking slot for vehicle using Internet of Things, *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 5392-5399, 2020.

[5] L.Chungsan, Han Youngtak, Soobin Jeon, Dongmahn Seo and Inbum Jung. Smart Parking System for Internet of Things, *IEEE International Conference on Consumer Electronics*, 2016.

[6] N. Ramsari, and S. Utomo. Booking smart parking system using microcontroller, *IOP Conf. Series: Materials Science and Engineering* 830, 2020.

[7] V. Chauhan, M. Patel, S. Tanwar, S. Tyagi, N. Kumar. IoT Enabled real-Time urban transport management system, *Computers and Electrical EngineeringVolume* 86, 2020.

[8] Wu-Jeng Li, Chiaming Yen, You-Sheng Lin, Shu-Chu Tung, and ShihMiao Huang. Just IoT Internet of Things based on the Firebase Real-time Database,. Ministry of Science and Technology, Taiwan, Jun 2018.

[9] Deborah Lee Soltesz. What Does JavaScript Do? (2016). Retrieved from Techwalla website. https://www.techwalla.com/articles/what-does-javascript-do

[10] Jason. Benefits of Mobile Apps (2016). Retrieved from Vividus website. [Online]. Available: https://vividus.com.au/insights/benefits-of-mobile-apps/

[11] *The International Communication Union (2013). ICT Facts and Figures.* Retrieved from The International Communication Union Sites. [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013-e.pdf