The Development of Smart Parking System based on NodeMCU 1.0 using the Internet of Things

L Anjari and A H S Budi*

Department of Electrical Engineering Education, Universitas Pendidikan Indonesia, Jl. Dr. Setiabudhi 207, Bandung 40154, Indonesia

*agusheri@aol.com

Abstract. Recently the number of vehicle ownership has been increasing, hence it needs a large parking area. The shortage is how to look for the empty parking space in the large parking area and also the amount of illegal parking car that does not follow the rules of the parking space unit. In this paper, we proposed the smart parking system using the Internet of Things (IoT) that can be part of a solution for the parking problem. This system help in organizing the parking lot and helps the driver to reach their parking spots easily as they known which space is vacant. The parking space can be detected using an Infrared sensor that connects to the ESP12-E (NodeMCU) module that was programmed through Arduino IDE. Users can access parking space information using a smartphone via an application. Especially for users who have been registered before, they have a code for login the app as the requirement for security system and user parking convenience. The system can work with the purpose of the research appropriately.

1. Introduction

In recent years, the Internet of Things has been applied in many ways. The smart parking system is one part of the technology of the Internet of Things [1]. The concept of the Internet of Things starts from a device that can be traced, controlled, or monitored over the internet [2]. One of the systems of smart parking is to know the condition of parking lot via the internet. This is related to parking problems which one of them is the difficulty of knowing the condition of vacant space in the wide parking lot so that the driver spends his time just to find a parking place and tend to more difficult along with the increasing number of vehicle ownership [2-4]. Problems related to parking can be solved if the driver can be informed beforehand about the availability of parking space around the desired destination [2]. As the result, the concept of the Internet of Things applies to the smart parking system.

Various approaches and research have been done to overcome parking problems. Since the early 1970s, smart parking has been implemented throughout Europe, the UK, and Japan. The initial system is displayed in the driver's parking information such as availability status and/or the amount of space available [5]. More complex smart parking incorporates more advanced technology to serve customers with different needs [6]. A recent survey conducted by the International Parking Institute [6] reflects an increasing number of innovative ideas related to parking systems. Currently, there are certain parking systems that are able to provide real-time information about available parking spaces. Such systems require efficient sensors to be placed in parking lots to monitor parking spaces and rapid data processing units to gain practical insights of data collected from various sources [2].

According to these data, this study was to design a prototype of parking monitoring system as well as to develop parking system with reference to the pre-existing system [7]. In this system, we used NodeMCU as the microcontroller which was supported by Wi-Fi system. The proposed smart parking system was implemented using a mobile app that connects to the cloud. This system helped users to know the availability of parking space in real time.

2. Methods
The project of this smart parking system was built by some system such as hardware architecture, software architecture and system.

2.1. Hardware architecture
The main of the hardware architecture in this project was built by 3 component, as followed:

2.1.1. NodeMCU (ESP-12). The NodeMCU (Node MicroController Unit) in figure 1 is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains all crucial elements of the modern computer: CPU, RAM, networking (wifi), and even a modern operating system and SDK. NodeMCU in this prototype used a microcontroller module with a Wi-Fi system which was programmed using Arduino IDE. The version of NodeMCU was using NodeMCU v.1 board or commonly also called ESP-12E. Because the I / O is limited, then in this project used 2 NodeMCU to comply the goal of the project.
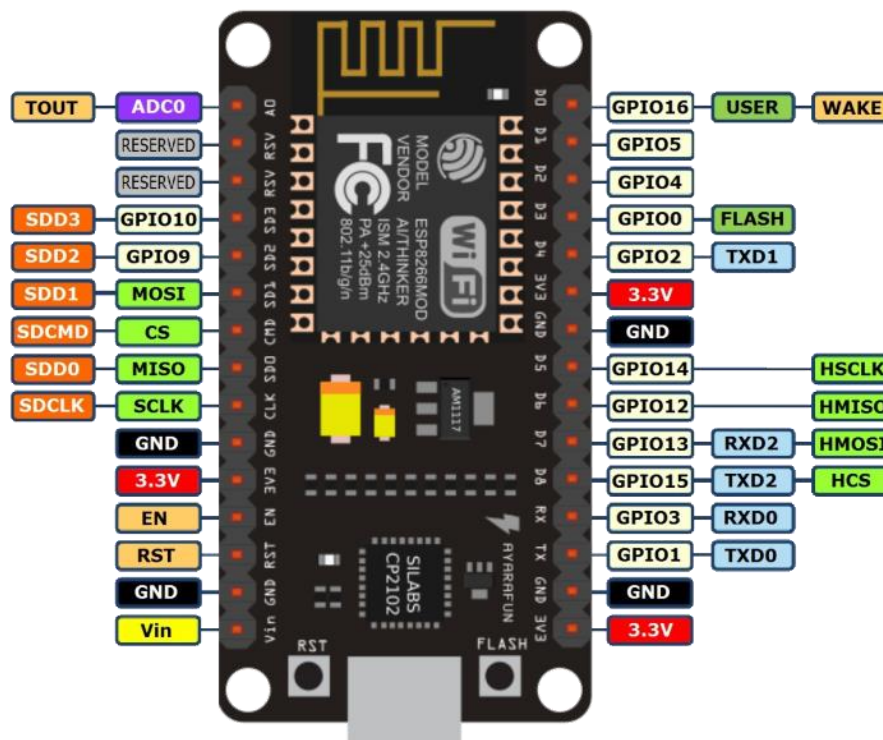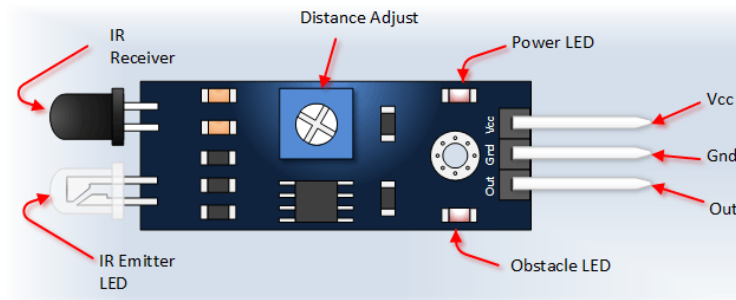

Figure 1. NodeMCU 1.0.

2.1.2. Infra-red obstacle. IR Obstacle is used for detecting the parking space, the way it works is when there is an obstacle that blocks the sensor, the LED will be active as a sign or information that the parking space is already filled by someone.

Figure 2. Infrared obstacle sensor.

In this prototype, 2 Sensors Infrared in figure 2 used to detect the car with the logical AND for showing on the app. The use of AND logic in sensor placement is intended to be more accurate parking space detection and also to avoid possible misuse of sensors in parking spaces.

2.1.3. Power supply adapter. The adapter is a circuit to convert AC voltage into DC. Power Supply Adapter is an adapter that can convert a large AC power voltage into a small DC voltage egg. From 220v AC voltage to 6v, 9v, or 12v DC voltage. The adapter used in this prototype is using 5V / 1.5A adapter in order to support the power consumption requirement for all components.
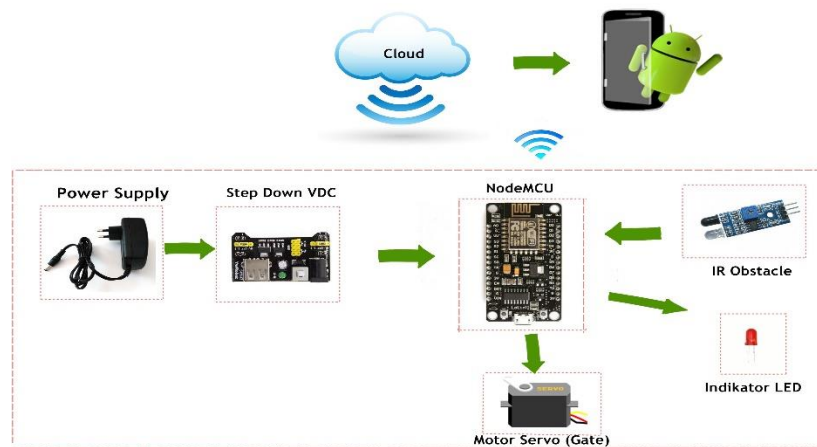
2.2. *Software architecture*

Software architecture is built by several systems, which is programmed by Arduino and sent into the cloud and then the app synchronizes with the real-time database on the cloud.

2.2.1. Arduino IDE. The Arduino Integrated Development Environment (IDE) is a software for write and upload the program into the NodeMCU for communicating between microcontroller and Cloud or a system.

2.2.2. Firebase. Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. Firebase provides a real-time database and backend as a service. The service provides application developers with an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The database is also accessible through a REST API. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Firebase in this project is used to store the data of prototype. Firebase here acts as a cloud. Cloud Firestore is a NoSQL document database that lets to easily store, sync, and query data for your mobile and web apps - at a global scale.
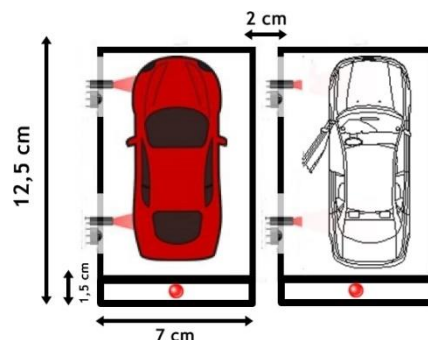
2.2.3. MIT app inventor. MIT App Inventor is an open source web application. It allows users to create an Android application. The app inventor replaces the complex language of text-based coding into a visual drag and drops building blocks. The app inventor can be used even by inexperienced users to create the basic and fully functional app. The MIT App Inventor was provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT) and the inventors are Mark Friedman and MIT Professor Hal Abelson. MIT app inventor in this project is used for display the real-time database on the firebase. The real-time database is the data of the sensor of smart parking project.

2.3. System architecture

Figure 3. Architecture of smart parking system.

The architecture of smart parking system as shown in figure 3 has depicted many components. The IR Obstacle sensors are connected to the microcontroller NodeMCU. NodeMCU chip comprises of a self-contained SOC (system on a chip) with integrated TCP/IP protocol stack that allows any microcontroller to access a Wi-Fi network. NodeMCU acts like an intermediate between the sensors and cloud. Cloud acts as a database to store all the records related to parking areas and users that have access to the system. The NodeMCU then transmits this data to the Firebase through Arduino IDE. The mobile application acts as an interface for the users to interact with the system on Cloud Firebase.
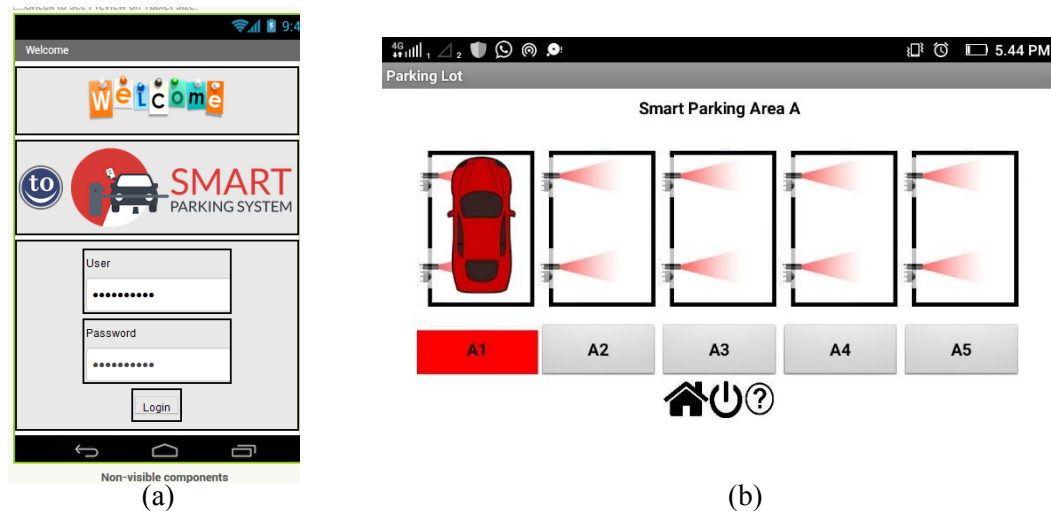

Figure 4. Architecture of prototype for parking space dimension.

A parking space as shown in figure 4 is a location that is designated for parking. Space many are delineated by road surface markings. The car fits inside the space, either by parallel parking, perpendicular parking or angled parking. The basic dimensions based on the Parking Planning and Operation Manual are dependent on door openings, vehicle type. The width of the door open will affect passenger comfort in and out of the vehicle. On the prototype of smart parking system is used perpendicular parking because it's commonly used in car parking lots.

3. Results and discussion
The result of this project, one of which is creating a smart parking app that is connected to the prototype that has been created. This app consists of several screens such as depicted in figure 5, and 6. The first screen in figure 5 is for the registered user login, the second screen in figure 6 for the menu, and the next screen in figure 7 is to display the contents of the menu one which is the parking lots menu of the prototype made.
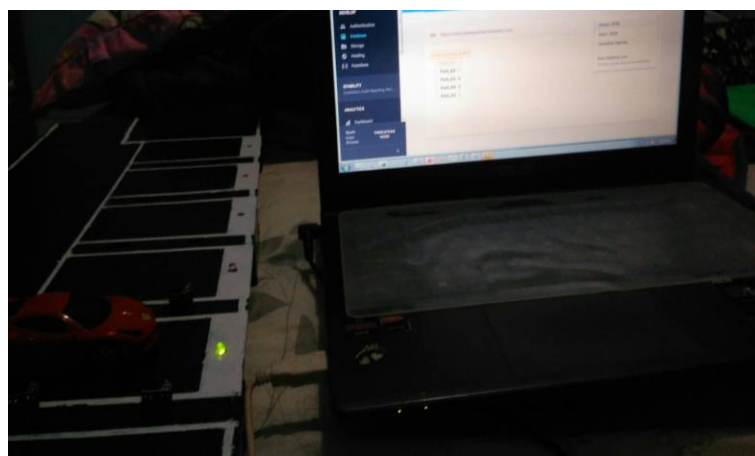
| (a) | (b) |

Figure 5. (a) The registered user login, (b) Display the contents of the menu.

Based on the prototype as shown in figure 10, the space parking A1-A5 will be detected filled or detected there is a car if the IR Obstacle sensor obstructed by the car. LED will be active as a sign or information that the parking space is already filled by someone. This system using a Wi-Fi connection to communicate between ESP8266 and Cloud Firebase. The data is sent from the sensor and synchronized with the data in the application. And then, the application will display the data parking lot in real-time.

The application is synchronized with a real-time database on firebase that acts as a Cloud as shown in figure 6. This is done to test the concept of the Internet of Things (IoT) that is when the data can be accessed by everyone anywhere and anytime via an internet connection. This project has several constraints that are the mismatch between the goals with the results achieved. These constraints include:

- In the planning of this project, the microcontroller that used is ESP32 because it has more I/O than NodeMCU. Because of the limitations of science that are understood and IoT platforms that support ESP32 is more complicated than ESP8266 there is a human error where the system created is not in accordance with the planning.

- NodeMCU has only 10 digital I/O, so the system is not built according to the planning that there are 2 sensors in each parking space.

Figure 6. Real-time database on firebase platform.

4. Conclusions

Smart Parking System on this project has been successfully made in accordance with the purpose of the smart parking system. One of the purposes is to know the condition of the parking lot through the internet that is connected with the application in real-time. This application can allow parking users to search for empty space in the parking lot whenever and wherever using applications based on Internet of Things (IoT).

References

[1]   Lee C, Han Y, Jeon S, Seo D and Jung I 2016 "Smart parking system for Internet of Things" *IEEE Int. Conf. Consum. Electron.* pp. 263–264.

[2]   Khanna A and Anand R 2016 "IoT based smart parking system" *Int. Conf. Internet Things Appl.* pp. 266–270.

[3]   Basavaraju S R 2015 "Automatic Smart Parking System using Internet of Things (IoT)" *Int. J. Sci. Res. Publ.* **5** (12) pp. 629–632.

[4]   Ibrahim F et al. 2016 "Smart Parking System Based on Embedded System and Sensor Network" *Int. J. Comput. Appl.* **140** (12) pp. 975–8887.

[5]   Chinrungrueng J, Sunantachaikul U and Triamlumlerd S 2007 "Smart Parking: An Application of Optical Wireless Sensor Network" *Int. Symp. Appl. Internet Work.* pp. 66–66.

[6]   Rodier C J, Shaheen S and Kemmerer C 2008 "Smart Parking Management Field Test: A Bay Area Rapid Transit (BART) District Parking Demonstration Final Report" *Transp. Res. Rec.* **2038** (1) pp. 62–68.

[7]   Septriyaningrum I A, Nugrahadi D T and Ridwan I 2016 "Perancangan Dan Pengembangan Prototype Sistem Parkir" *J. Ilmu Komput.* **3** (2) pp. 146–155.