An IoT Raspberry Pi-based Parking Management System For Smart Campus

ABSTRACT

Parking slots have become a widespread problem in urban development. In this context, the growth of vehicles inside the university's campus is rapidly outpacing the available parking spots for studentsand staff as well. This issue can be mitigated by the introduction of parking management for the smartcampus which targets to assist individually match drivers to vacant parking slots, saving time, enhance parking space utilization, decrease management costs, and alleviate traffic congestion. This paper develops an IoT Raspberry Pi-based parking management system (IoT-PiPMS) to help staff/studentsto easily find available parking spots with real-time vision and GPS coordinates, all by means of a smartphone application. Our system composes of Raspberry Pi 4 B+ (RPi) embedded computer, Pi camera module, GPS sensor, and ultrasonic sensors. In the IoT-PiPMS, RPi 4 B+ is used to gather and process data input from the sensors/camera, and the data is uploaded via Wi-Fi to the Blynk IoTserver. Ultrasonic sensors and LEDs are exploited to detect the occupancy of the parking spots withthe support of the Pi camera to ensure data accuracy. Besides, the GPS module is installed in the system to guide drivers to locate parking areas through the Blynk App. that discovers parking spaces availability over the Internet. The system prototype is fabricated and tested practically to prove its functionality and applicability. According to the results, the IoT-PiPMS can effectively monitor the occupancy of outdoor parking spaces in the smart campus environment, and its potency in terms of updating the data to the IoT server in real-time is also validated..

Introduction

The growing number of automobiles is rapidly outpacing the parking lots availability in urban areas; thus, parking has become a widespread challenge in metropolises development [1]. This problem can be observed clearly in cities, especially centers, shopping malls, open markets, government offices, hospitals, schools, and universities [2]. There are two main types of parking lots, indoor and outdoor. In conventional parking management systems, the information about parking spots cannot be efficiently collected and updated to the management platform [3, 4]. In such systems, many parking spaces exist as local data silos, but their parking spot information cannot be shared remotely with drivers [5]. Additionally, data of isolated parking cannot be transmitted to a unified platform for city parking management.

A smart parking concept has been introduced to mitigate the difficulties of parking spot management in a smart environment like the smart city [3, 6, 7]. Smart parking systems have been given considerable attention in the literature to assist individually match drivers to parking slots, enhance the utilization of parking spaces, reduce the cost of management, and alleviate congestion of traffic [8-10]. However, the existing smart parking systems still possess many constraints related to architecture, connectivity, and deployment. Existing systems that are deployed for smart parking use different types of wireless networks for communication between sensor nodes and a management platform server [11]. This requires additional gateways, thus increases system complexity and makes it hard to maintain and impedes the scalability of smart parking. This severely constrains the deployments of smart parking systems in most areas when necessary. The foliage effect on path loss, shadowing, and multipath dispersion.

Due to the increasing number of students/staff who drive to the university campus, the number of vehicles inside the campus also grows. Therefore, parking spaces are getting more occupied than ever. This problem poses a daily challenge to both students and staff alike, as they have to make many rounds just to find a vacant parking space. As a result, drivers would waste their precious time and energy where they might be late for their work or classes. The parking management system based on the Internet of Things (IoT) technology has great potential to overcome the stated issues faced by the conventional systems of parking management. Owing to the characteristics of the university campus with regard to size, the Internet infrastructure, and the existence of open spaces as outdoor parking lots, Raspberry Pi-based IoT system is a potential candidate to enable parking management for the smart campus. Recently, universities including our university (Universiti Malaysia Pahang) have devoted more focus to the smart campus paradigm in particular with the emergence of the Fourth Industrial Revolution (IR 4.0). Several initiatives have been started under the smart campus project, smart parking management is one of them. Thus, discovering parking spots around the university campus in real-time is essential for the convenience of staff, students, administrations, and even visitors. Furthermore, not only for convenience purposes, but such a smart system can help vehicles owner within the campus in time-saving, pollution reduction, and efficient resource utilization. In this context, the use of IoT networks will allow sharing of data about the outdoor parking slots of the campus over the Internet; thus, helps both drivers and security management.

Since parking monitoring needs detection of vehicle existence, multiple sensors should be deployed in parking spaces. For the sake of system accuracy, different sensor types should be combined depending on the desired data to be measured, such as distance, coordinates, and vision. Three sensors are exploited, which are ultrasonic, GPS, and Pi camera for video streaming as an add-on system function. One ultrasonic sensor can be placed per each parking spot, while one GPS sensor and one Pi camera can cover several parking slots, depending on components features and the measured parameters. Raspberry Pi as a single board computer would provide both higher efficiency and lower cost, especially for large-scale smart campus deployments. The Raspberry Pi with the embedded Wi-Fi interface can be used to easily connect to widespread hotspot internet coverage inside the campus and performs as the gateway of the IoT system. The collected data from various sensors will be uploaded to the Blynk IoT server and can be accessed via mobile apps.

In this work, an IoT Raspberry Pi-based parking management system (IoT-PiPMS) for smart campus or any similar environments is developed as a solution for monitoring and managing outdoor parking slots in real-time. We have also proposed and integrated three algorithms that gather the required measurements from the sensors to detect the occupancy of the parking spots. The developed IoT-PiPMSA system is attached to a parking space prototype that has been designed and fabricated for testing and validation purposes. The IoT-PiPMS is applicable for outdoor parking facilities including, but not limited to universities, schools, colleges, markets, hospitals, and shopping malls. The key benefits of the proposed system include detection accuracy enhancement of the system with an ultrasound sensor and Raspberry Pi Camera, simplicity of microcontroller system, can be operated in areas within Wi-Fi coverage, and the GPS support enables ease of locating parking facility. The number of available parking slots will be detected by placing an HCSR04 ultrasonic sensor on each parking slot and also using the Pi camera for visual detection. The available parking slot will be updated to the Blynk IoT server and can be accessed by both users and the management office. The major contribution of the current study includes four aspects:

i) We design an IoT-based parking management system using Raspberry Pi 4 B+ (RPi) to detect the occupancy of the outdoor parking space and update the information to an IoT platform for the smart campus use case.
ii) We fabricate the developed system as a prototype to emulate the real situation of outdoor parking lots for testing and validation purposes by implementing several sensing nodes with one Pi camera and a GPS sensor.
iii) We integrate three Python algorithms to gather data from the deployed sensors and send them to the Blynk IoT server in real-time. The status of parking slots will be displayed in a mobile-based IoT dashboard to be used by staff, students, visitors, and management as well.
iv) We report the results of the field trial experiments conducted using a real environment with end devices. The obtained results provide an insight into the system functionality and applicability in a smart campus.

Components Selection

The materials and components are chosen based on necessities and compatibility based on the conducted literature review. While conducting the review of related studies, we have surveyed multiple variations of designs and components used and also their respective function in smart parking systems. The hardware and software are decided during the component selection phase. The hardware components that are chosen are Raspberry Pi 4 Model B, Pi Camera v2.0, HSC-SR04 ultrasonic sensor, LED lights. The software tools that are chosen are Raspbian Operating System, Python, Fritzing, Siemens NX 10, Proteus Design Suite, Blynk IoT platform, and VNC. Once the component selection has been completed, the system architecture is figured out. In this subsection, an overview of the selected components is introduced.

Raspberry Pi 4 Model B

Raspberry Pi 4 Model B has been chosen for this project mainly due to its powerful processing capabilities as a single-board embedded computer and its ability to connect to the Internet either via Ethernet port or wirelessly using Wi-Fi or Bluetooth. Ease to connect to the Internet is one of the RPi advantages over Arduino, and it also supports a huge variety of programming languages such as Python, Java, C, C++, Perl, Ruby, BASIC; whereas Arduino, however, only accepts either Arduino or C/C++. Raspberry Pi 4 Model B is a direct upgrade over the Raspberry Pi 3 Model B+. It has Broadcom BCM2711 1.5 GHz quad-core 64-bit CPU with a Cortex-A72 processor, which is more efficient and much more powerful than Pi 3's 1.4 GHz processor. The GPU of Pi 4 is capable of running comfortably compare to Pi 3 due to Pi 4 improved clock speed: 500 MHz compared to Pi 3's 400 MHz. The RPi 4 Model B has a better CPU and GPU compared to Model 3 B, which is essential when we have decided on a visual-based detection method using the Pi camera. The Pi camera requires significant processing speed to be able to capture and process the images. Therefore, Raspberry Pi 4 is overall more suitable for our system and also able to give better performance in terms of video capturing and streaming.

Pi Camera

The pi camera is selected for the proposed system since it is a camera that is built specifically for Raspberry Pi. It is very suitable to use in a situation whose light is low because of its NoIR filter. Users could just plug in the camera onto the Pi board, run a few commands then the camera is active and ready to use. The Pi camera connects directly to the GPU, which is capable of 1080p30 HD video encoding. It is also capable of snapping pictures of 5MP resolution. Due to its attachment to the GPU, it does not draw any resources from the CPU, leaving it available for other processes. USB Webcams, on the other hand, drain resources from the CPU, lowering the performance of the entire system. As for the prices, this camera only costs around 25 USD, which is a great deal in terms of features and performances it provides.

HSC-SR04 Ultrasonic Sensor

The ultrasonic sensor is a sensor that transmits sound waves between 25 to 50kHz to the surrounding by detecting transmitted energy, which is reflected back to the sensor. The reflection of the ultrasonic wave, together with a signal processing module, the wave will be analyzed to detect the presence or absence of an object in the surrounding every 60 milliseconds. This sensor can be utilized by detecting vehicles and assessing the occupancy of the parking space. The ultrasonic sensor is easy to install without the need for facility closure. In the parking management system, a huge number of distance sensors are required (one sensor for each slot). The ultrasonic sensor is chosen due to its cost-effectiveness when considering the number of the required sensors. It only costs about 1 USD each, and it is still able to provide reliable results compared to other proximity sensors. In the deployed prototype, we have used four ultrasonic sensors. This sensor can utilize for detecting vehicles and assessing the occupancy of the parking space. The ultrasonic sensor is easy to install without the need for facility closure.

NEO-6M GPS Module

NEO-6M GPS Module is a GPS module capable of supporting microcontrollers. It is user-friendly and uses up only a little space on board. It is a very useful device for locating device location. This module is chosen to allow the positioning of the parking lot location.

Green LEDs

LEDs are very efficient in energy usage; it consumes only 10% of the power of what a normal incandescent bulb would normally consume. This, in turn, reduces the power costs due to low operating power. The LED is used in this project to indicate whether a parking spot is occupied or vacant by switching its light: ON green for vacant and OFF for occupied. One LED is used for each parking slot.

Raspbian

Raspbian is a free operating system that is used for modifying or program all models of Raspberry Pi. Raspbian is pre-installed with other applications for general use, programming, and educational purposes. It supports language for Python, Java, Scratch, and more. One would install Raspbian onto a PC or Micro SD and plug it into RPi, then connect a monitor to the Pi board, allowing the Pi board to run like a normal PC. Raspbian lets users install plenty of software from its open-source software repository for free. In this study, Python is used to develop the required coding for gathering data from various sensors to the RPi and upload it to the IoT cloud.

Blynk

Blynk is an IoT platform that enables the development and implementation of smart IoT devices with ease and speed. Tree major components that make Blynk a perfect candidate for our project: apps, server, and library. The apps offered by Blynk allows user to customize the widgets shown on the interface. Once the interface is done designed, the apps would then connect to an open-source Blynk server. The server acts as a centralized cloud service that would allow communications between the devices. Other than that, Blynk provides a variety of libraries that supports multiple hardware devices such as RPi, Arduino, and ESP8266. The server allows communication through Wi-Fi, Bluetooth, BLE, USB, and GSM.

Proteus Design Suite

Proteus is software that allows design engineers and technicians to create a sheet of electronic prints and schematics. Other than that, it also allows simulation of microcontrollers such as Atmel AVR, Arduino, or Microchip Technologies. Since there is no hardware required to use the simulation, it is convenient as a teaching tool or designing for mini-projects. We have simulated our system using Proteus to confirm the correctness of our circuit design before the real implementation.

Siemens NX 10

NX 10 is software designed for CAD/CAM/CAE. It is mainly used for designing prototypes by direct solid/surface modeling, perform engineering analysis like stress testing the prototype and also manufacture the finished design using included machining modules. The NX 10 is a direct competitor against Catia, SolidWorks, Autodesk Inventor, and Creo. The NX 10 is chosen because it has its own module for CAD, CAM, and CAE; users could transfer their project between the modules immediately with no data loss.

VNC Viewer

VNC Viewer is a client system that simply allows simultaneously access to many devices from one screen using the IP address. The VNC provides home computing environments access from everywhere on the public Web server. Thus, providing application sharing on the computer. It is usually used with RPi to access its desktop from other screens to increase the portability and mobility of Raspberry Pi. The VNC viewer is installed by default within Raspberry Pi allows users to see and operate the Raspberry Pi board desktop on their own laptop as if it is on the display monitor. Users just have to connect the Raspberry Pi and the laptop onto the same network and login to VNC on both devices.

System Architecture

We have decided about the layout of the system architecture as shown in Figure 1 based on the selected components. The signal flow and the interlinks between various components are suggested. In the developed system, Raspberry Pi is the main controller that collects data from the implemented sensing units and Pi camera. The GPS module provides the location for the parking area. Then the ultrasonic sensors, which sense the vehicle presence inside the parking spot where will trigger the data of the parking slot status to the Raspberry Pi, which uploads data to the IoT server with the embedded Wi-Fi module. Thus, drivers with Internetconnected devices such as smartphones and tablets can access the information about parking slots occupancy. The users can observe and monitor the parking area through the Pi-camera that provides visual monitoring via video streaming as an add-on function when required. Alternatively, the Pi camera is acting as a security surveillance monitor to the parking area to increase the security of the users. A real-time condition of the parking area is streaming to the cloud for monitoring, and it can be used by the management office for security purposes.

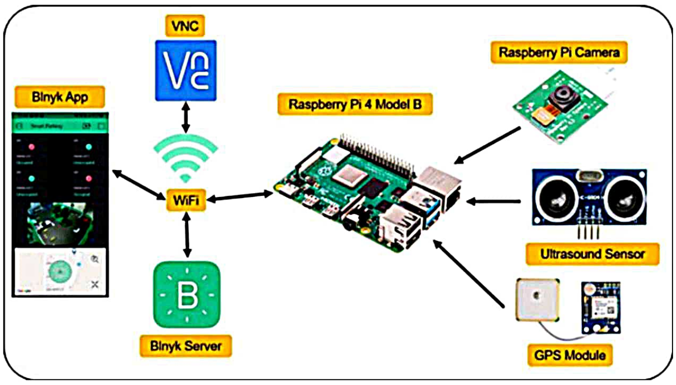

Figure 1. Raspberry Pi Based Smart Parking Management System architecture

IoT-PiPMS Design

Once the system architecture is confirmed, the electrical circuit diagram of the designed system is initiated using the Fritzing tool as shown in Figure 2. Raspberry Pi GPIO pins are connected to the input/output pins of the three sensors (ultrasonic sensor, GPS sensor, and Pi camera. In this design, only one ultrasonic is shown. However, in the testing prototype, more sensors are used. The HSC-SR04 is used to detect the presence of a car in the slot and accordingly, the RPi will change the LED status ON/OFF that indicates the parking slot is already occupied or not. The components are installed on the breadboard to ensure that the system works successfully before transferring the component onto the solder circuit panel. The Pi camera will trigger as the parking slot is occupied or not via visual streaming. The suggested design is then simulated using Proteus software to check the compatibility of the components. Figure 3 shows the stimulation of RPi with GPS module, Pi camera, and ultrasonic sensor in the Proteus software. After the electronics stimulation in Proteus simulation, all the electrical and electronic components were assembled and connected to the microcontroller and programmed using Python in the Raspbian OS. However, the Proteus software not fully utilized because some of the component libraries were not found in the software, as well as on the Internet. Therefore, the sensors and components had to be tested experimentally using RPi.

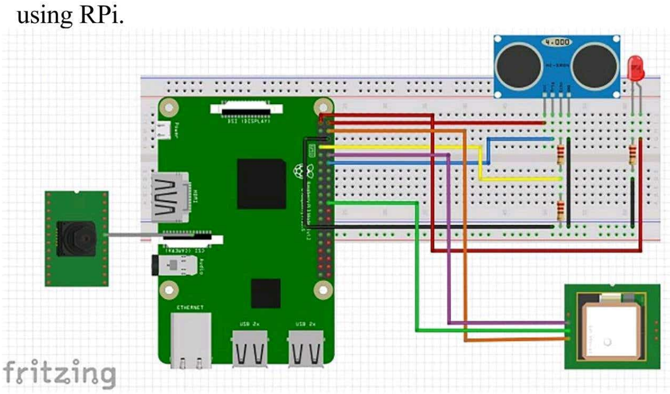

Figure 2. Component circuit diagram.

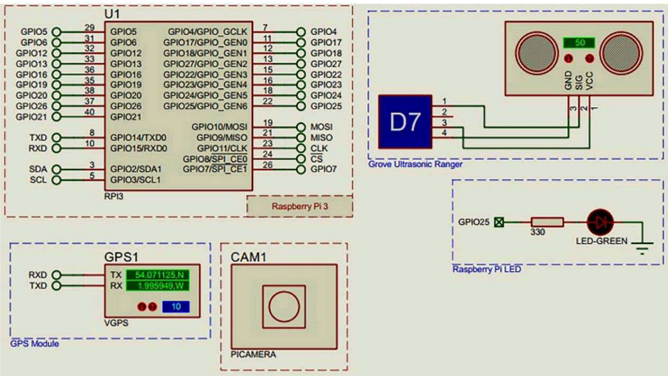

Figure 3. Proteus simulation of the control circuit.

Model Design

In order to emulate the functionality of the developed system, we need to design and fabricate a proper outdoor parking space model. Siemens NX 10 software is utilized to design the prototype with a measurement of 500×600 mm and a thickness of 15 mm, as shown in Figure 4. The reference of measurement is taken from the basic size of a square table. The purpose of this design is to visualize how the prototype will be created and where to put all the components. This model is designed with the exit and entry gate along with 8 parking slots to make it a clearer view on how the parking slot is. Among the 8-parking slots, there are four parallel parking slots and four perpendicular parking slots. As for the Pi camera and GPS Module, it will be placed at a higher place which is at the pole that is located at the corner of the parking space. Thus, one Pi camera can cover a wider space and streams video about the situation of the covered parking lot. Also, the GPS sensor will be placed on the same pole close to the microcontroller; thus, mitigating the required wiring. On the other hand, the ultrasonic sensors will be placed facing in front of the parking slot so it can immediately detect the car as the distance between the sensor and the car becomes closer.

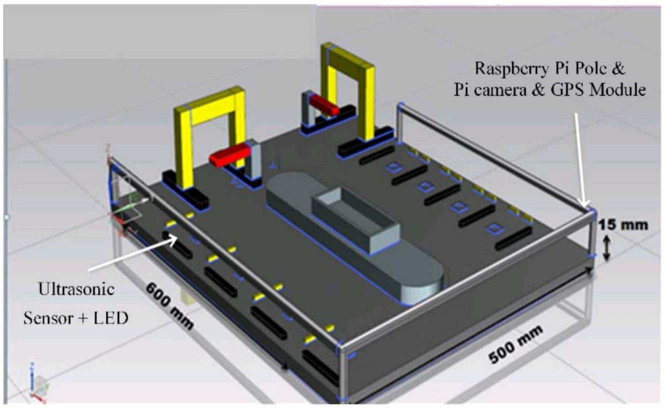

Figure 4. Design and dimensions of the prototype model

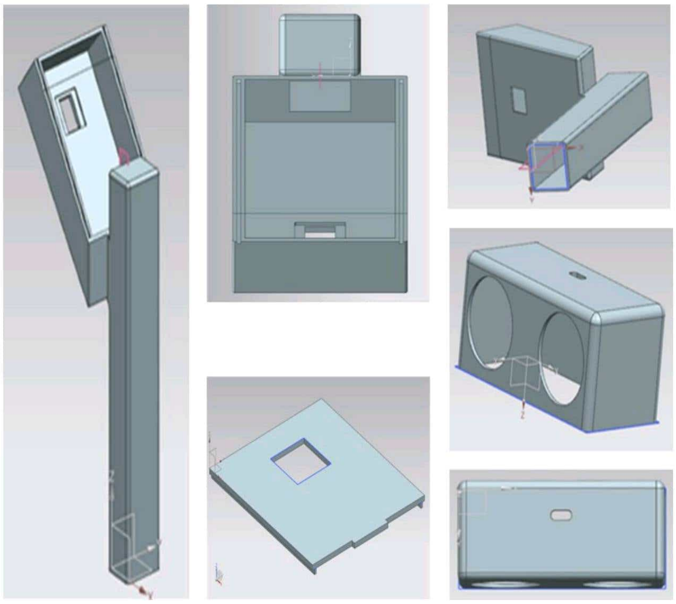

Figure 5. Design of components for system model

The RPi support pole is designed as shown in Figure 5. We measured every edge and corner of RPi to make the box fit to it. During the design process, we left some space around the designated box to provide space for some of the wiring parts. The designed RPi box also has a hole for the Pi camera. For the space for the wiring part, the pole was designed with the hole inside the RPi box and along with it. In order to place the GPS module on the same support, the RPi box cover is designed together with the GPS Module hole. The function of the GPS module installed there is to help drivers locate unoccupied parking slots. This cover box will close the upper side of the RPi box to protect the components that are put inside the box. Since we also use ultrasonic sensors to detect the presence of the car, the ultrasonic sensor box is also being designed with an LED hole. The ultrasonic sensor boxes have been specifically designed for the HC-SR04 sensor. The box has two holes at the front of the box for the transmitter and receiver of the sensor and one hole on top of the box for placing LED. The function of the LED light is to indicate the presence of the car.

Model Fabrication

Once the simulation is completed, the fabrication process is beginning. During the manufacturing process, the materials are assembling according to the simulated design. First of all, the process of measuring and cutting plywood takes place. Plywood is used as the base of the car parking model to strengthen the design. Next, the canvas board was used as the next base of the design. It was glued on the plywood. Both of them act as the base of the design. The function of the canvas board was to make it easier to put the hardware components in it without causing any scratch to the components. After both of the bases were glued, the other components were assembled. The next step is to create space for the car parking slots. To realize our model, the Styrofoam board and plastic board are used. To make the parking looks livelier and realistic, some fake trees and carpet grass were added. A hole was made at each parking slot for the wire to go through under the prototype. 3D printing is used to fabricate the part components of the smart parking model by using materials of PLA Filament. We successfully fabricate the model's components, including the Raspberry Pi pole, ultrasonic sensor box, RPi box cover, entry and exit parking signs using 3D printing. The fabricated model of the proposed system is shown in Figure 6.

Figure 6. The fabricated system model