PHASE :3 DEVELOPMENT PART 1

BUILDING THE IOT SENSOR SYSTEM AND RASPBERRY PI INTEGRATION.

Raspberry pi is the name of the “credit card-sized computer board” developed by the Raspberry pi foundation, based in the U.K. It gets plugged in a TV or monitor and provides a fully functional computer capability. It is aimed at imparting knowledge about computing to even younger students at the cheapest possible price. Although it is aimed at teaching computing to kids, but can be used by everyone willing to learn programming, the basics of computing, and building different projects by utilizing its versatility.

Raspberry Pi is developed by Raspberry Pi Foundation in the United Kingdom. The Raspberry Pi is a series of powerful, small single-board computers.

Raspberry Pi is launched in 2012 and there have been several iterations and variations released since then.

Various versions of Raspberry Pi have been out till date. All versions consist of a Broadcom system on a chip (SoC) with an integrated ARM-compatible CPU and on-chip graphics processing unit (GPU).

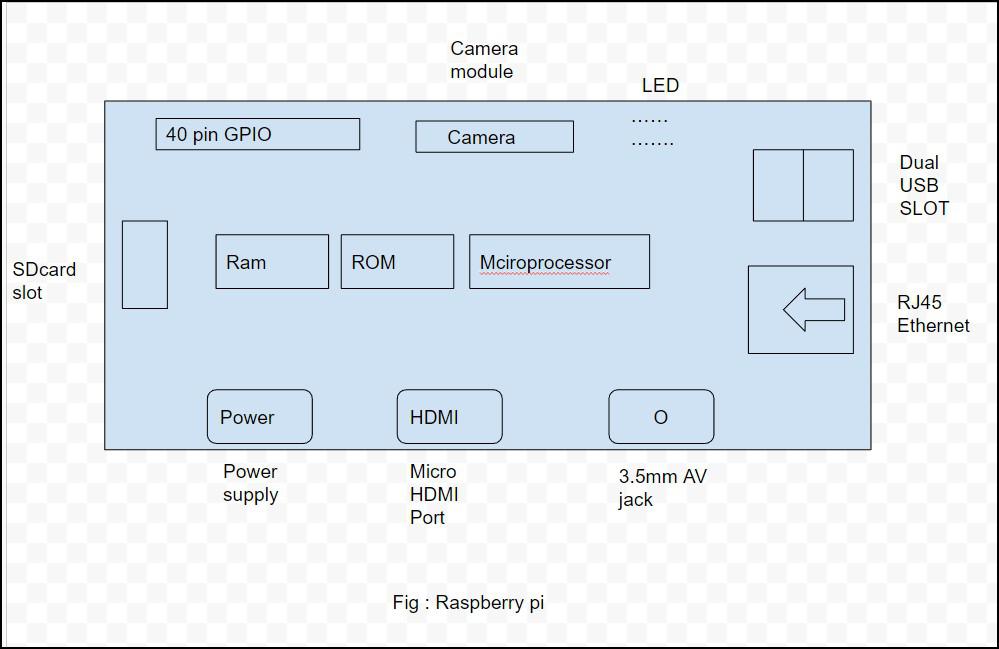
The original device had a single-core Processor speed of device ranges from 700 MHz to 1.2 GHz and a memory range from 256 MB to 1 GB RAM.

To store the operating system and program memory Secure Digital (SD) cards are used. Raspbian OS which is a Linux operating system is recommended OS by Raspberry Pi Foundation. Some other third party operating systems like RISC OS Pi. Diet Pi, Kali, Linux can also be run on Raspberry Pi.

**Used:**

It also provides a set of general purpose input/output pins allowing you to control electronic components for physical computing and explore the Internet of Things (IOT).

**Raspberry pi Diagram :**

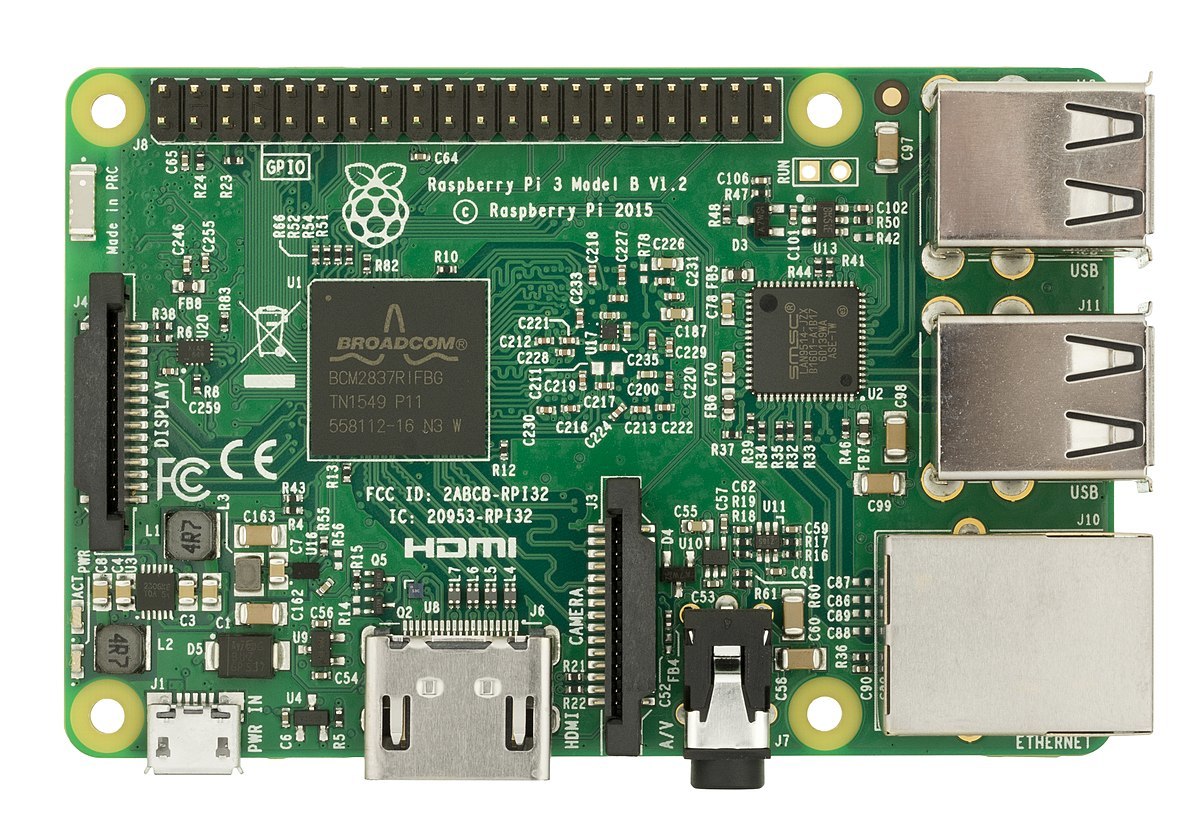


**Raspberry Pi model –**

There have been many generations of raspberry Pi from Pi 1 to Pi 4.  
There is generally a model A and model B.  
Model A is a less expensive variant and it trends to have reduce RAM and dual cores such as USB and Ethernet.

**List of Raspberry pi models and releases year:**

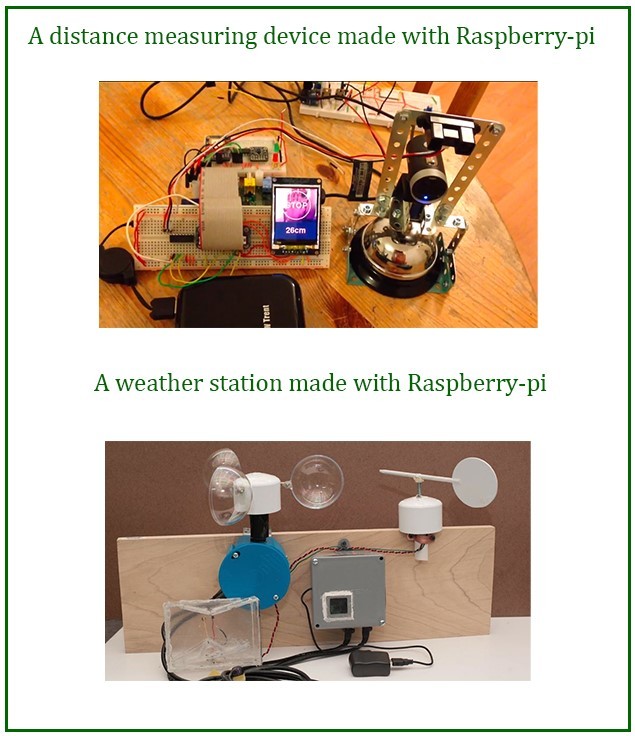
1. pi 1 model B – 2012
2. pi 1 model A – 2013
3. pi 1 model B+ -2014
4. pi 1 model A+ – 2014
5. Pi 2 Model B – 2015
6. Pi 3 Model B- 2016
7. Pi 3 Model B+ -2018
8. Pi 3 Model A+ -2019
9. Pi 4 Model A – 2019
10. Pi Model B – 2020
11. Pi 400 – 2021



**Specs of the Computer: –** The computer has a quad-core ARM processor that doesn’t support the same instruction as an X86 desktop CPU. It has 1GB of RAM, One HDMI port, four USB ports, one Ethernet connection, Micro SD slot for storage, one combined 3.5mm audio/video port, and a Bluetooth connection. It has got a series of input and output pins that are used for making projects like – home security cameras, Encrypted Door lock, etc.

**Versatility of Raspberry Pi: –** It is indeed a versatile computer and can be utilized by people from all age groups, it can be used for watching videos on YouTube, watching movies, and programming in languages like Python, Scratch, and many more. As mentioned above it has a series of I/O pins that give this board the ability to interact with its environment and hence can be utilized to build really cool and interactive projects.

**Examples of projects: –**It can be turned into a weather station by connecting some instruments to it for check the temperature, wind speed, humidity etc… It can be turned into a home surveillance system due to its small size; by adding some cameras to it the security network will be ready. If you love reading books it can also become a storage device for storing thousands of eBooks and also you can access them through the internet by using this device.



***Conclusion:****Concluding the article it is convenient to assert that it is a small and powerful computer at a dirt-cheap rate and can handle most of the task as a desktop computer.*

CONFIGURE IOT SENSOR TO DETECT PARKING SPACE OCCUPANCY.

1. Introduction:

Vehicles are important modes of transportation at present. However, the number of vehicles has increased greatly in recent years and the limited availability of vehicle parking spaces has caused several problems. Recently, parking management systems have been developed where electronic displays provide information about the available parking spaces as well as other types of parking information. However, this approach is not convenient because drivers must search for information about the available parking spaces via the electronic display and then search directly for the locations of empty parking spaces. Thus, despite the high installation costs for electronic displays, the current parking management systems are not satisfactory because they do not provide simple and useful information. In the era of the Internet of Things (IoT), smart “things” can be embedded with various types of sensor devices and communication devices, which may be deployed in different environments and the network can be configured without human interference, where their interactions with each other make new information available in the IoT ecosystem. In the future, it is expected that many application services well be developed for IoT environments because the sensors and microprocessors will have better performance, as well as being smaller in size with lower costs. Among the many expected IoT services, smart parking services are particularly important, and they may combine IoT technology with existing vehicle parking systems .

2. Related Work

2.1. Parking Management System RFID and camera devices are used to obtain information from vehicles in order to identify them in parking spaces. Thus, the parking management system uses an RFID device as a tag to identify a vehicle when it enters a car park as well as to guide users when they are searching for an empty parking space. In order to provide guidance to empty parking spaces, an RFID module must be attached to the vehicle and an additional display device is installed at the car park entrance (Wei et al., 2012). The user’s smartphone can be employed to guide them to an empty parking space and improve usability for users. However, parking management systems that use RFID devices have high maintenance costs because this approach requires special RFID modules for users and their vehicles.

2.2. Estimation Locations with Wireless Communication:

Time of arrival (TOA), time difference of arrival (TDOA), RSSI, and angle of arrival (AOA) methods may be employed to estimate the locations of users. These methods exploit various characteristics related to the frequency of wireless communication. The TOA method is also known as the time of flight. TOA uses the transmission time of a radio signal from the sender to the receiver to estimate the distance. The speed of a radio signal is equal to that of light. If this method is employed to facilitate location awareness among users, then accurate time synchronization between the sender and reviver is essential. However, time synchronization among smart things incurs high overhead costs. Thus, many studies have investigated accurate time synchronization in the IoT.

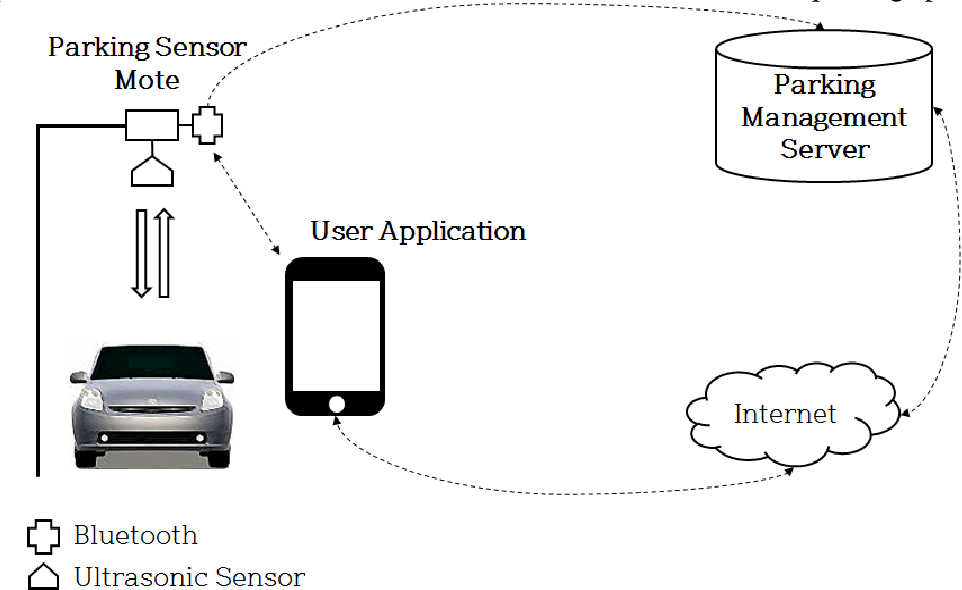
2.3. Route Finding Algorithm:

The shortest path algorithm seeks the path with the minimal cost between two nodes on a graph. Many algorithms have been developed for this purpose. In particular, the A\* algorithm is a best-first search algorithm, which performs a heuristic search of the nodes based on rankings. The implementation of this method is simple but it can find the shortest path in a small number of computations. In the A\* algorithm, a visited node is set as a closed node and an unvisited node is set as an open node. The A\* algorithm visits the open nodes continually until it reaches the goal node. In the node visiting procedure, this algorithm assigns each node with a fitness degree. The fitness (f) is derived using

Equation (1): f = g + h

1. Smart Parking System :

Figure 1 illustrates the smart parking system proposed in this study. The smart parking system comprises wireless parking sensor motes, a parking management server, and an application running on the user’s smartphone. The wireless parking sensor motes are installed on the ceiling above a parking space. Each mote comprises an ultrasonic sensor module, BLE communication module, and control module. The ultrasonic sensor in the sensor mote can determine whether the parking space is occupied. The sensor mote transmits the state of the parking space to the server and the server manages the states of all the parking spaces in the car park. The BLE module in the sensor mote communicates with the smart parking user application running on the user’s smartphone. This module receives the identification information for each parked vehicle. In addition, the BLE module operates as a beacon to facilitate location awareness for users of the parking space.



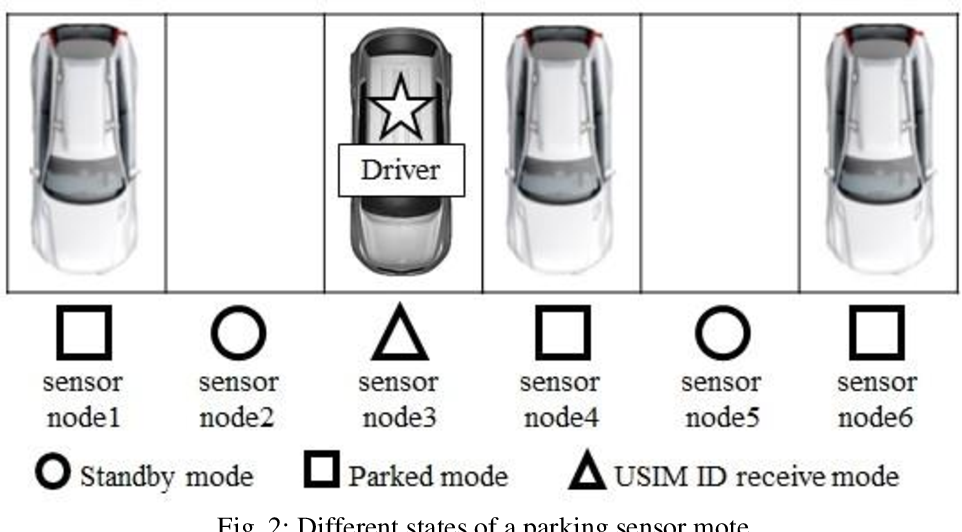
3.1. Vehicle Parking Decisions :  
Ultrasonic sensors have been used previously for recognizing the presence of vehicles in parking spaces, and they have also been deployed on roads to measure the traffic volume and the presence of vehicles (Jo et al. 2014; Jeon et al. 2014). In our method, we used a wireless parking sensor mote with an ultrasonic sensor to check the status of parking spaces. The steps involved in recognizing the presence of a vehicle in a parking space are as follows. First, an ultrasonic sensor is installed on the ceiling above a parking space and the distance from the ceiling to the ground is measured using a backward echo wave from the ultrasonic sensor. This distance is used as a basic value to determine the vehicle parking status. According to the enforcement regulations for indoor parking spaces, the height of parking spaces must be at least 2.3 m from the ceiling to the ground. In our method, we assume that the height is 2.5 m. If the parking space is empty, the ultrasonic sensor attached to the ceiling detects a distance of 2.5 m from the floor. Thus, a distance of 2.5 m is used as a basic value to determine the parking status for parking spaces. If a vehicle is parked in the space, the sensor would detect a distance between 0.5 m and 1 m depending on the height of the vehicle. Therefore, sensor data values less than 2.5 m indicate that a vehicle is occupying the parking space.

3.2. Location Awareness in Parking Space:

To provide a location awareness service related to parking spaces, a real-time awareness method is needed based on the movements of users. Recently, location awareness methods have been developed based on the inter-relationships between pervasive smart things deployed in the environment and this is an active research area (Mainetti et al. 2014; Lee et al. 2003; Zaruba et al. 2007). Our smart parking system provides a location awareness service to parking space users. After the vehicle has been parked, each user is distinguished according to their parking space because the parking application on the user’s smartphone automatically transfers identification information to the sensor mote.

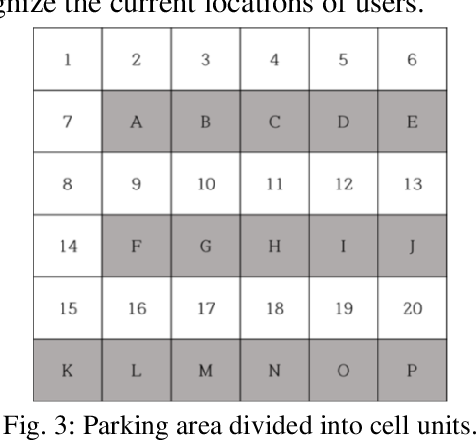
3.2.1 Location Awareness for Parked Vehicles:

In a smart parking space, the BLE modules of several sensor motes can communicate with the user’s smartphone simultaneously. The RSSI strength from a sensor mote becomes stronger as the distance between the sensor mote and the user smartphone decreases. Thus, the sensor mote with the strongest RSSI strength is selected to save the location of a parked vehicle. There is no need to calculate accurate distances to the connected sensor motes to find the sensor mote for the currently parked vehicle. However, a highly reliable selection of the correct sensor mote can be achieved when the relative values of the RSSI strengths are compared.



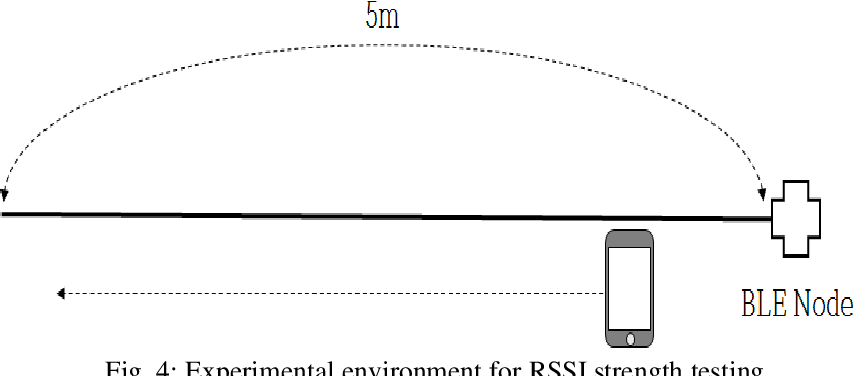
3.2.2 Location Awareness for Real-Time Location Guidance:

Our smart parking system provides a real-time location guide service to parking space users, where this service guides users to their parked vehicles or to parking spaces. This service requires a new real-time location awareness method because it must track the locations of users during the limited time when they are moving continuously in the car park. As mentioned in the previous section, the RSSI strength of BLE is used to facilitate location awareness guidance to parked vehicles and parking spaces. However, the BLE strength cannot be used when vehicles or pedestrians are moving because the sensor motes are not deployed in all areas. Thus, an accurate location awareness method and real-time processing are needed in order to guide the current user to the location of their vehicle. In this method, we divide the parking area into cell units where the cell size indicates the space where a vehicle can be parked. We use these cells to recognize the current locations of users.



3.2.3 Distance Measurement Based on the RSSI Signal Strength:

Based on the RSSI signal strength, we can calculate the distance between the sensor mote and the user’s smartphone. We used two devices in our experiments and Figure 4 shows the experimental environment employed for deriving the formula for the distance based on the RSSI signal strength. As shown in Figure 4, the BLE module of the sensor mote was fixed and the smartphone moved a distance of up to 5 m. We measured the RSSI strength at 10-cm intervals for 5 m. The experiment was performed 10 times in the same environment.



Implementation and Experiments:

4.1. Parking Sensor Mote As shown in Figure 7, we implemented the parking sensor mote proposed in this study, which comprised a Devan tech SRF04 ultrasonic sensor, Jinan Hua Mao HM10 BLE module, and Crossbow MicaZ control module. Table 1 shows the specifications of the MicaZ mote. The MicaZ is based on an At mega 128L processor with low power consumption, which has high scalability features with 51 extension pins. The MicaZ has been employed previously in implementations of wireless sensor networks in the IoT because of these features (Online). In this study, we used Tiny OS as middleware for the sensor mote, which is based on an eventbased operating system and it supports low power operation (Online).



4.2. User Parking Application :

The user parking application was developed in Android 4.1 version and installed on a smartphone. The basic screen displayed the current state of the parking space and it indicated the location where the user’s vehicle was parked. It also included functions for saving the location of the parked vehicle and the location guidance service to parking spaces was provided. Figure 8 shows the interface screen for the user parking application. The cells represent parking spaces or the pathways for vehicles and pedestrians. The gray cells indicate the available parking spaces and the black cells are the parking spaces occupied by vehicles. The white cells represent the pathways for vehicles and people.

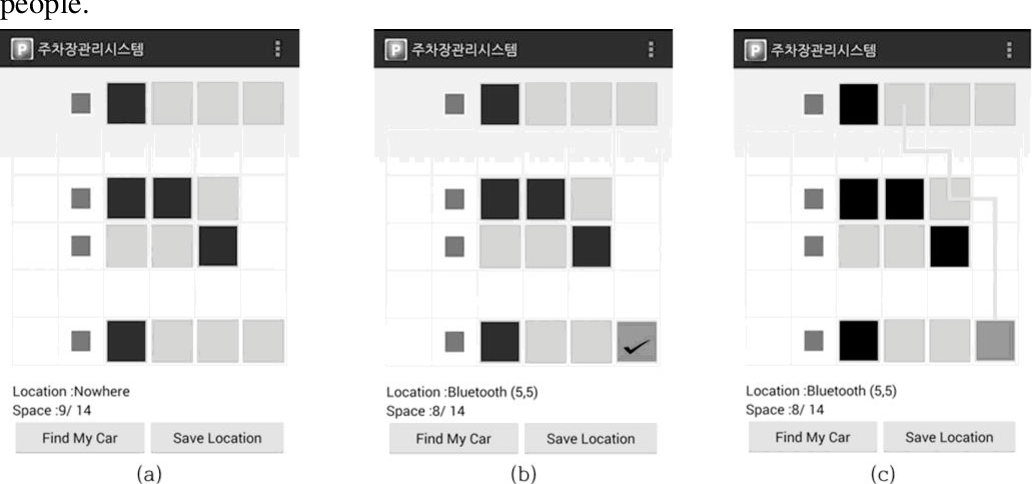


Fig. 8: User parking application: (a) saving the parking location, (b) current state of parking spaces, and (c) location guidance service.

4.3. RSSI Strength of Sensor Motes :

In this study, the BLE modules of the sensor motes were controlled at four signal strength levels, i.e., –23 dBm, –6 dBm, 0 dBm, and 6 dBm. The RSSI data indicated the different values according to the strength of each signal. In order to save the locations of parked vehicles in a reliable manner, we performed an experiment to determine the optimal signal strength within a range of 2–3 m. As mentioned in Section 3.1, this is the distance between the sensor mote on the ceiling and the floor of the parking space. Figure 9 shows the experimental environment used to measure the signal strengths from the BLE module, where the parking sensor mote was at a height of 2 m and we measured the signal strengths. The measurements were performed 300 times.

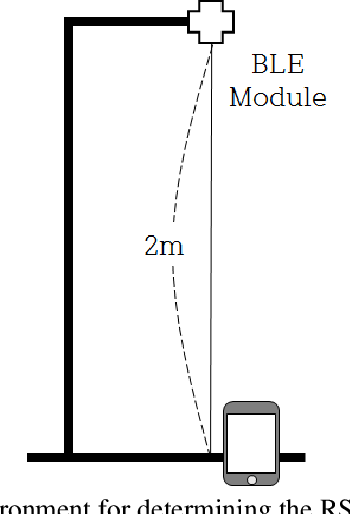
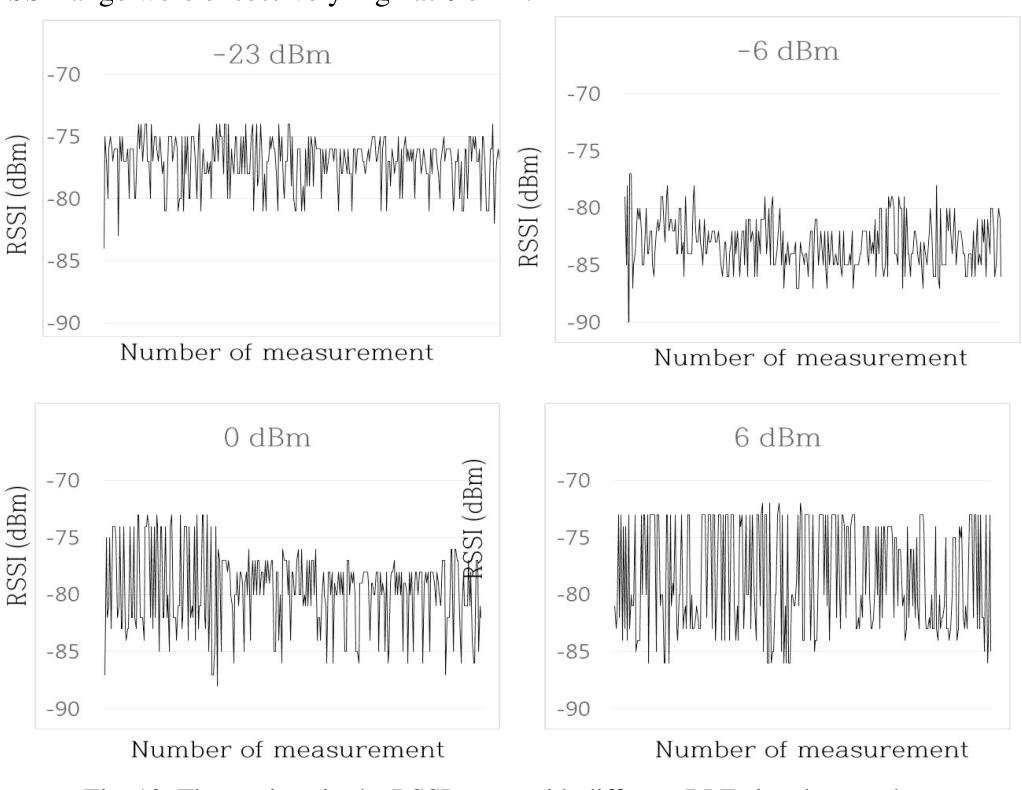
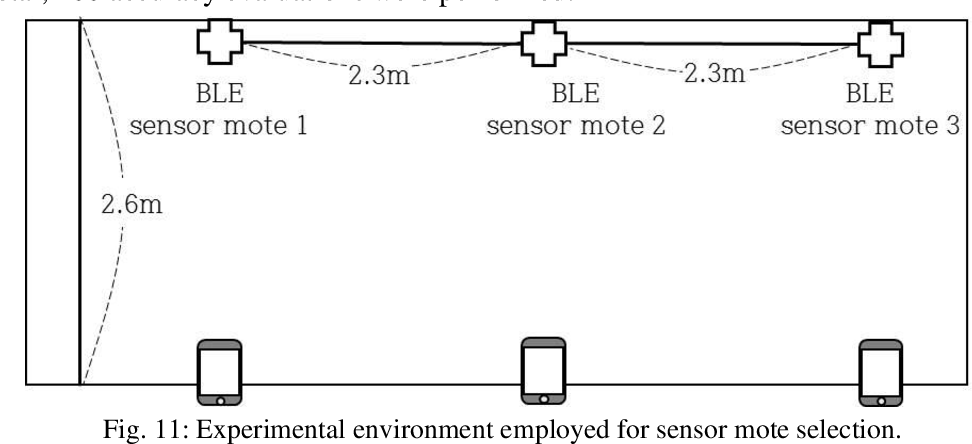


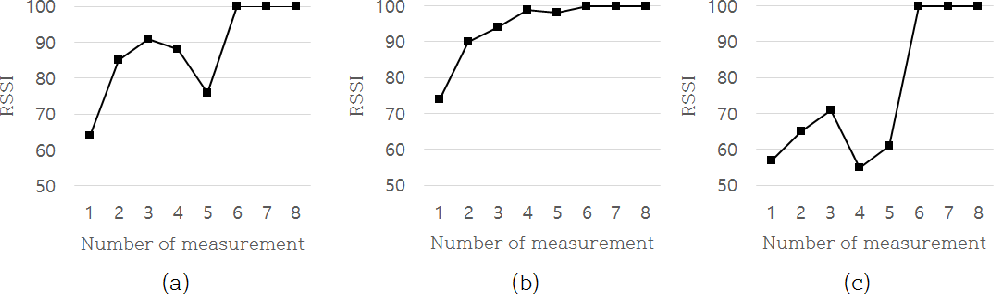
Figure 10 shows the RSSI values achieved with the BLE module at four signal strengths. At 2 m, the measured RSSI value was not proportional to the signal strength for the BLE module. According to the signal strength, the RSSI values exhibited different characteristics. Thus, at –23 dBm, the signal strength was strong and the fluctuations in the RSSI range were not large. At –6 dBm, the signal strength was low and the fluctuations in the RSSI range were similar to those at –23 DBM. However, at 0 dBm, the variation in strength was almost the same as that at – 6 dBm, but the fluctuations in the RSSI range were large. The fluctuations in the RSSI range were excessively high at 6 dBm.



4.4. Sensor Mote Selection for Parked Vehicles:

Several sensor motes may be connected with a user’s smartphone, but the sensor mote with the strongest RSSI should be selected before saving the location of the parked vehicle. Next, we constructed an experimental environment according to the specifications for a real parking space in order to evaluate the accuracy of the selection process. As shown in Figure 11, three sensor motes were deployed, where the distances between sensor motes were 2.3 m and that from the sensor mote to the smartphone was kept at 2.6 m. These distances were based on real parking spaces. Each space was occupied by a vehicle. We measured the RSSI for sensor motes located above each parking space. The accuracy evaluation confirmed a parked vehicle based on the sensor mote with the highest signal strength. In the experiments, we collected 1–8 RSSI data during each one minute interval for each parking space. The average values were calculated to increase accuracy. In total, 100 accuracy evaluations were performed



Accuracy of sensor mote selection: (a) sensor mote 1, (b) sensor mote 2, and (c) sensor mote 3. Figure 12 shows the accuracy obtained when the sensor mote with the highs.

5.Conclusion:

In this study, we implemented a smart parking system based on sensor motes with ultrasonic sensors and BLE communication devices in the IoT. The states of the parking spaces are checked in real-time by the smart parking system. In addition, based on user location awareness technology, the locations of parked vehicles are recognized and a location guidance service is provided to help users find their parking space. The ultrasonic sensor in the sensor mote is employed to determine the states of parking spaces. The ultrasonic sensor calculates the distance between the sensor mote and the floor of the parking space using a backward echo ultrasonic wave. If the distance between the sensor and the floor is below 100 cm, then the space is regarded as being occupied by a vehicle. We also implemented two location awareness methods based on the RSSI of the BLE modules in sensor motes. In order to obtain reliable RSSI data, we measured the fluctuations in the RSSI range according to the signal strength of the BLE module. We found that reliable RSSI data could be received when the signal strength was weak and the fluctuations in the RSSI were small. Therefore, we decided to use a signal strength of 6 dBm in our experiment.

WRITE PYTHON SCRIPT ON RASPBERRY PI TO COLLECT DATA FROM SENSORS AND SEND IT TO THE CLOUD OR MOBLIE APP SERVER.

Creating a Python script on a Raspberry Pi to collect data from sensors in a smart parking system and send it to the cloud or a mobile app service involves several components and may vary depending on the specific sensors and cloud service you are using. I'll provide a general outline for such a system:

1. **Set up Raspberry Pi:** Make sure you have a Raspberry Pi with an internet connection, and ensure it is configured with the necessary dependencies for your sensors.
2. **Install Necessary Libraries and Sensors:** Depending on the sensors used in your smart parking system, install the relevant libraries and connect the sensors to the Raspberry Pi.
3. **Collect Data from Sensors:** Write Python code to read data from your parking sensors. This could include ultrasonic sensors for detecting car presence in parking spots, magnetic sensors for detecting vehicle presence, or any other relevant sensors. Here's an example using ultrasonic sensors:

python  
import RPi. GPIO as GPIO

import time

# Set GPIO mode

GPIO. Set mode(GPIO.BCM)

# Define sensor pins

trigger\_ pin = 23

echo\_ pin = 24

# Setup sensor pins

GPIO. setup(trigger\_ pin, GPIO.OUT)

GPIO. setup(echo\_ pin, GPIO.IN)

# Function to measure distance

def measure\_ distance():

GPIO. output(trigger\_ pin, True)

time. sleep(0.00001)

GPIO. output(trigger\_ pin, False)

start\_ time = time. time()

stop\_ time = time. time()

while GPIO. input(echo\_ pin) == 0:

start\_ time = time. time()

while GPIO. input(echo\_ pin) == 1:

stop\_ time = time. time()

elapsed\_ time = stop\_ time - start\_ time

distance = (elapsed\_ time \* 34300) / 2

return distance

try:

while True:

distance = measure\_ distance()

print(f' Distance: {distance} cm')

time. sleep(1)

except Keyboard Interrupt:

GPIO. cleanup()

1. **Set Up Cloud Services:** Choose a cloud service for data storage and processing. Common options include AWS, Google Cloud, Azure, or IoT platforms like AWS IoT or Google Cloud IoT. Set up a project and credentials.
2. **Send Data to the Cloud:** Use the appropriate library or API for your chosen cloud service to send the parking sensor data. This may involve sending data over HTTP, MQTT, or another protocol. The code will vary based on the chosen service.
3. **Set Up a Mobile App or Web Service:** Create a mobile app or web service to visualize the parking data. You can use tools like Flask or a mobile app development framework like React Native.
4. **Retrieve Data in Mobile App/Service:** Implement code in your mobile app or web service to retrieve and display the parking data from the cloud service.
5. **Automate Data Collection:** Set up a loop or schedule the Python script for continuous data collection and transmission.
6. **Handle Errors and Security:** Ensure error handling in the script and apply appropriate security measures when dealing with sensitive data.

OUTPUT:

The provided Python script measures the distance using an ultrasonic sensor and prints the distance in Centi meters to the console. The output of the script will continuously display the distance as long as the script is running.

Here's an example of what the output might look like:

Distance: 45.0 cm

Distance: 50.2 cm

Distance: 48.5 cm

Distance: 52.1 cm

...

The actual distance values will vary depending on the position and detection by the ultrasonic sensor. If the sensor doesn't detect an object within its range, it may display a very large value (e.g., 4000 cm) or similar, indicating no object is within range.