Asian Journal of Convergence in Technology **Volume 5 Issue I**
ISSN No.:2350-1146, I.F-5.11

UNDERSTANDING THE ARCHITECTURE OF INTERNET OF THINGS USING A CASE STUDY OF SMART PARKING

Abhay Deshmukh[1], Radhika D. Joshi [2]
[1,2] Department of Electronics and Telecommunication Engineering
College of Engineering, Pune, India

Abstract— Nowadays, the Internet of Things (IoT) is the latest research trend all over the world. IoT is a multidisciplinary branch that includes electronics, civil, computer and mechanical. The things named after "smart" in the market are nothing but the products of IoT. To understand a complete picture of IoT, information of all the branches is necessary. In this paper, a case study of smart parking is explained in brief so that one should get familiar with all the aspects needed to implement any application of the IoT. It has importance in almost all sectors such as smart home, cities, environment, energy, retail, logistics, agriculture, industries, health, and lifestyle. The focus of IoT is on configuration, control, and networking via the internet of devices or "things" that are traditionally not associated with the internet. This paper discusses all the layers of the architecture of IoT. Using the knowledge offered by this paper, anyone interested in carrying out an end to end application of the IoT to implement any smart system will get a complete idea about how to proceed as well as about what sectors he/she should be aware of. In this paper, various methods to achieve a complete end to end smart parking are discussed and the same methods can be used to achieve various systems such as smart lighting, smart home, healthcare, smart agriculture, etc.

Keywords— ***The Architecture of the Internet of Things (IoT), Smart parking system, Protocols***

I. INTRODUCTION

Internet of Things can be defined as an interaction between physical and digital worlds. The digital world interacts with the physical world using several sensors and actuators. Another definition of IoT is a paradigm in which computing, and networking capabilities are embedded in any kind of conceivable object. These capabilities can be used to query the state of the object and to change its state, if possible. In common parlance, IoT refers to a new kind of world where almost all devices and appliances that are in use are connected to a network. Their use is made collaboratively to achieve complex tasks that require a high degree of intelligence. For this intellect and interconnection, IoT devices are armed with embedded sensors, actuators, processors, and transceivers. The storage and processing of data can be accomplished on the edge of the network itself as well as on a remote server. If any preprocessing of the data is possible, then it is typically done at either the sensor or some other proximate device. The data which is processed is sent to a remote server. Along with the challenges of data collection and handling, there are challenges in communication too. The communication among IoT devices is mainly wireless because they are generally installed at geographically dispersed locations.

The architecture of IoT has three layers namely perception, network, and application layers.

(i) A perception layer is a physical layer which has sensors for sensing and gathering information about the environment. (ii) A network layer is responsible for connecting to other smart and network devices with servers. Its main objectives are to transmit and process the sensor data. (iii) An application layer is for delivering application- specific services to the user.

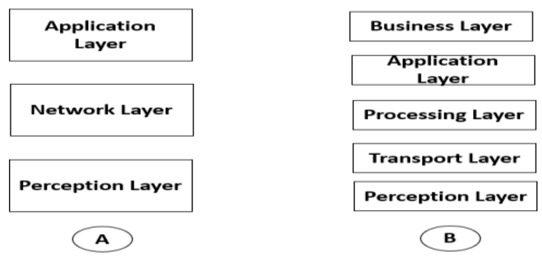


Fig. 1 Architecture of Internet of Things

The remaining three layers are Transport layer, Processing layer and Business layer. They have specific functions. A transport layer transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN, Bluetooth, RFID, and NFC. A processing layer is also known as a middleware layer. It stores, analyzes and processes a huge amount of data that comes from a transport layer. It can manage and provide a diverse set of services to lower layers. A Business layer manages the whole IoT system including applications, business and profit models, and users' privacy. The business layer has not been discussed further because its use is not within the scope of the present study.

Smart parking: For understanding the architecture of IoT, a case study of smart parking is implemented.

Fishing for a parking spot is a routine activity and it is estimated that nearly 30% of urban congestion is created by drivers finding for a parking spot, according to ITS Americas Market analysis. It results in fuel wastage, almost one million barrels of the world's oil every day. The reason behind the issue is an unavailability of a parking spot status. Thus, by using IoT, and by letting the devices get communicated with each other, it is possible to share the current parking status to the user prior to the start of a drive. A Smart Parking System mobile interface will be a great solution to the issue. Further, the system can be useful for revenue generation. A small change in the parking process with the help of IoT can bring happiness as discussed above, can reduce fuel wastage and parking slots can be effectively be optimized.

A prototype using Arduino and raspberry pi is incorporated. The project aims towards the way on how a real-parking status can be conveyed to the car driver so that the car driver can easily take a decision and without wasting time on searching for a vacant parking lot by just roaming. Parking and traffic on the road are the main sources of annoyance for drivers, merchants, employers as well as community officials in most cities. Smart parking services are on the top list of every research-oriented person and any businessman. In the era of IoT, it is being observed that smart parking solutions are considered innovative. If anyone wants to understand the importance in the innovation lying in opportunities with a smart parking system, then she/he must try to understand to whom the impact of parking problems is severe and think of the degree of the impact. For drivers, parking is usually seen as a compulsory trouble, and it frequently results into a disappointment. Thus, it may cause drivers to make use of alternate transportation choices to go to their destinations. Due to the frustration caused, they pass up going to those destinations and head to different destinations which will not have parking limitation. Parking control officers are typically perceived as bad guys for doing their jobs. However actually, as said in the study by Fybr, just 5% of parking rule breakers are literally cited. The parking management authorities should balance between the increasing social control usefulness against the view of rapid prosecution of parking rule breaker and raise issues from native businesses. For business, a proper place is crucial, if the parking is less or troublesome. The business might not pull towards you as many service users and workers as it should pull up and raise.

These parking issues may lead some consumers and labors to seek for other services. This is one of the important reasons why business does not progress and thus this may cause to relocate the business location. For the societies in crowded areas, parking is a vital problem observed daily. Limited parking causes people to search a new uncertain option for parking arrangements, which can be far from where they live or prefer to reside in another area where parking space is not an issue. It is obvious that the performance of all cities depends on tax returns and charges to produce and helping services for citizens and businesses within their borders. They must sense of balance parking revenues and citation enforcement, without alienate drivers, guests, and businesses.

II. RELATED WORK

In “Smart parking reservation system using short message services (SMS)” [1], Short message services (SMS) is used book parking spots. The microcontroller responds to SMS and provides a password and lot number. A secure car parking and reservation system using wireless technologies is explained in paper, “ZigBee and GSM based secure vehicle parking management and reservation system.” [2]. The system includes three modules, parking lot vacancy monitoring module, parking lot reservation module, and a security module. Parking lot monitoring module includes IR sensors and ZigBee modules which are interfaced with Microcontroller. Vacancy monitoring module detects the presence of a vehicle in the parking areas and provides the status to the users in real time. Reservation module includes a GSM modem which is interfaced with a coordinator system. Manjusha Patil, and Vasant N. Bhonge describes the work to modify the original WSN and use of RFID and Zigbee technology. The results are obtained when it is applied to parking garages [3].

In “New Smart Parking System Based on Resource Allocation and Reservations” [4], the system allocates and books an optimal parking slot based on the driver’s cost function that combines proximity to destination and parking cost which solves a mixed-integer linear programming (MILP) problem at each point. In “Automatic Parking Management System and Parking Fee Collection Based on Number Plate Recognition.”, author discusses an automatic parking system and electronic parking fee collection [5] based on a vehicle number plate recognition. The aim of this research study is to develop and implement an automatic parking system that will increase convenience and security of the public parking lot as well as collecting parking fee without hassles of using RFID. In addition to that, it has a parking guidance system that can show and guide user towards a parking space. The system uses image processing of recognizing number plates for operation of parking and billing system. M. Chester, A. Fraser, J. Matute, C. Flower, and R. Pendyala estimates how parking has grown in Los Angeles County (CA)[6] from 1900 to 2010 and how parking infrastructure evolves, affects urban form, and relates to changes in automobile travel using building and roadway growth models.

Here the authors of the paper in “Smart parking systems and sensors: A survey” [7], explore the concept of the smart parking system and their categories. The classifications of various existing systems are explained. The research of “Car park system: A review of smart parking system and it’s technology”, aims at developing an intelligent car parking system[8] that is more cost effective and user friendly than the already existing systems. The first phase aims at making cars detectable in the parking lot using sensors, while the second phase aims at communicating the collected data to the user remotely. The main aim of the author of, “Android based Smart Parking System” [9] is to propose a design of Android-based smart Parking System that regulates the number of cars to be parked on the designated parking area. This is done by automating the parking and un-parking of the car with the help of an Android application.

“A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies” paper introduces a novel algorithm that increases the efficiency of the current cloud-based smart-parking system [10] and develops a network architecture based on the IoT technology. This paper proposes a system that helps users automatically and a free parking space at the least cost. This cost is used to offer a solution of ending an available parking space upon a request by the user and a solution of suggesting a new car park if the current car park is full. Pampa Sadhukhan proposed a prototype of internet-of-thing based e-parking system [11] The proposed E-parking system uses an integrated component called parking meter to address issues related to parking as well as to provide smart parking management throughout the city.

III. METHODOLOGY

On the Perception layer, the parking system consists of parking nodes (slaves) and a cluster node (master). On the network layer, raspberry pi (coordinator) and a router are implemented.

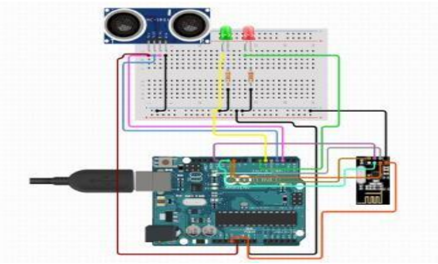


Fig. 2 Parking Node for case-study: Smart Parking

The parking node consists of an ultrasonic sensor which detects the presence of the car, two LEDs for the current status with two resistors for limiting a current through LEDs and a communication module named nRF24l01+. The nRF24l01+ module is wireless and thus makes complete system wireless and easy to deploy on the ground. Connections of a Parking node are as shown in Fig. 2.

The Cluster node consists of an ultrasonic sensor, two LEDs with two resistors for the same reason specified in the parking node. The Cluster node also consists of a RFID to detect the authenticated user at the entry of the Smart Parking System, Servo motor for allowing or not allowing the car (as a barrier boom). Connections of the Cluster node are shown in Fig. 3.

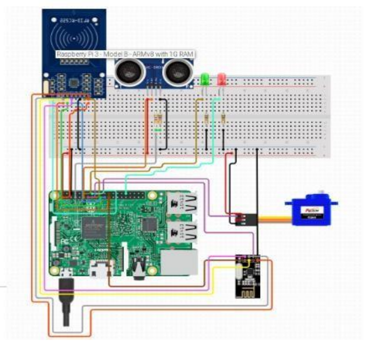


Fig. 3 Cluster node for Smart Parking

Figure 4 illustrates the steps in which the architecture can be implemented. The steps are as follows :- first it detects the car at a parking lot, after that it transfers the information gathered by the detector to the master, then it transfers the information gathered from the master to the coordinator. Further it creates a web application locally on the raspberry pi, and finally publishes the web application to the internet.

For detection of the car, an ultrasonic sensor is used at each parking lot. It is insensitive to the environmental changes and very precise than the rest available car detecting modules. The sensed information should be analyzed by some microcontroller which should take further decisions. For that Arduino slaves have been used. After that, the raspberry pi starts a web application.

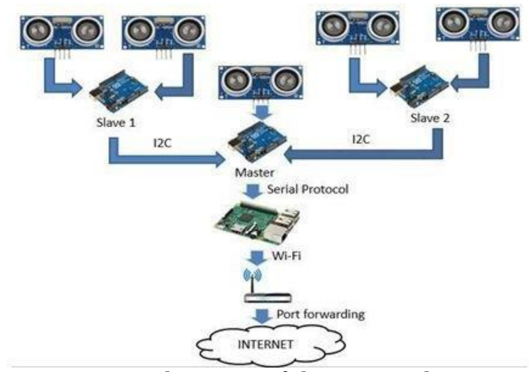


Fig. 4 Architecture of the proposed system

The raspberry pi is at the bottom of the stack, it acts as a hardware or physical layer as shown in Fig. 5. The Raspbian on raspberry pi acts as a software layer, which in general helps the hardware to work properly. The next on the web application stack is the application layer which consists of Uwsgi and Nginx. Uwsgi is a python script which will runs when the event is noticed on the user layer. Nginx is an interface between the user event and the python script which should be compiled according to the action taken by the user.

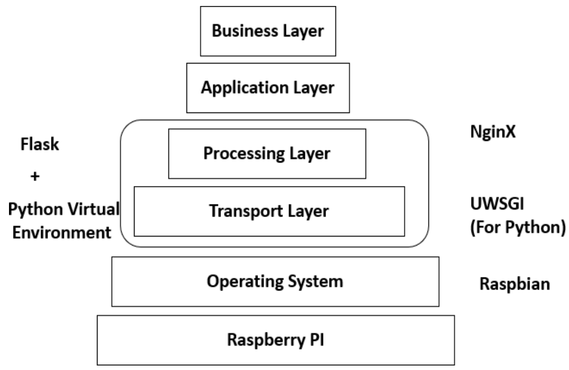


Fig. 5 Web Application Stack for Smart Parking

On the top of web-application stack, there is a user layer. It is nothing but what the user can see. The web interface lies at the user layer. The flow of parameters in the implemented system is shown in the Fig. 6.

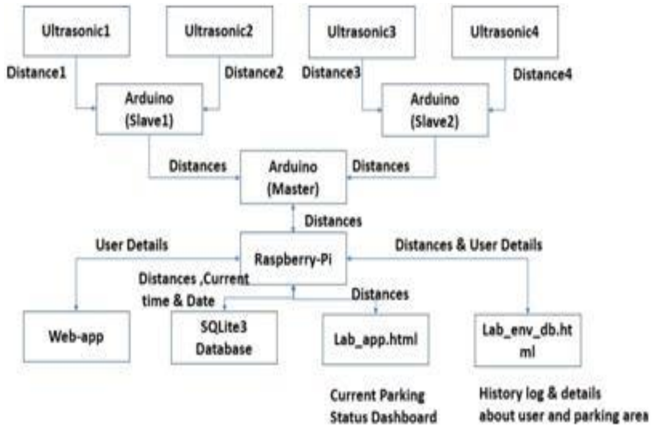


Fig.6 Parameter flow in proposed system

An ultrasonic sensor sends the distance between the car and itself and forwards it to the slave Arduino. Master Arduino collects the results from two slaves and forwards it to the raspberry-pi. Then the raspberry pi updates the database, that further updates the current parking status, and creates a car parking event log. After that, it is time for the system to push our data on cloud.

Fig.7 Identification of the Parking Node

Fig. 7 shows that the node with the device ID 'Ultrasonic Sensor' is connected to the cloud. The device type clears user the function of the device to the user at the cloud.

Event	Value	Format	Last Received
distance	{"Status":"24","distance":"24"}	json	a few seconds ago
distance	{"Status":"22","distance":"22"}	json	a few seconds ago
distance	{"Status":"23","distance":"23"}	json	a few seconds ago
distance	{"Status":"24","distance":"24"}	json	a few seconds ago
distance	{"Status":"33","distance":"33"}	json	a few seconds ago

Fig. 8 Reading of Node

As the device type is a "Sensor", it becomes very clear that the device behaves as a sensor. The rest is when the device is first time connected to the cloud. The IP of the Arduino is also known at cloud.

From Fig. 8, it is easy to understand that the device i.e., an Ultrasonic Sensor sends the real-time status to the cloud. The data is sent to the cloud because the data which is gathered from different parking nodes is very large and for the security aspect, it is needed to store the data somewhere in a specific format. The data which is stored on a cloud, is required for further analysis such as machine learning and integration of Artificial Intelligence and for better smart parking system.

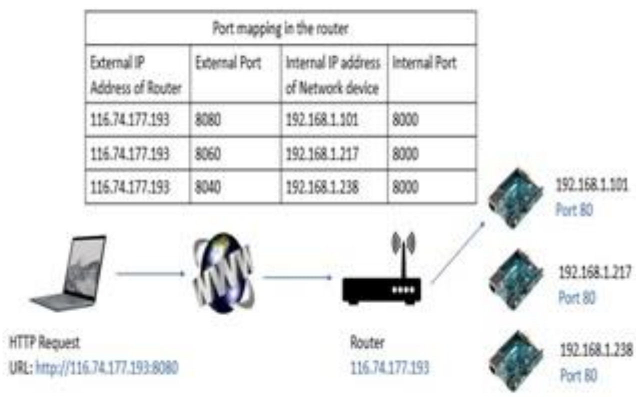
A list of the connection events reported for this device.

Message	Timestamp
Closed connection from 116.74.177.193. The connection was closed by the client.	Jul 1, 2018 4:29 AM
Token auth succeeded: ClientID='d:zvybpf:Sensors:UltrasonicSensor', ClientIP=116.74.177.193	Jul 1, 2018 4:28 AM

Fig. 9 Last Event Registered on IBM Bluemix

Fig. 9 helps to know the connection credentials used for the secured connection between a device and cloud. A unique client ID and a unique token is used at the cloud for the authenticity of the device at the cloud. Later for pushing our data from a local network to a global network, Port-Forwarding is needed.

It is known that each IP enabled device has two devices namely: - Private IP and Public IP. The Private IP is an IP for local network and the Public IP is an IP over the Internet. Assume, the system has 3 devices locally connected to the router with a gate IP of 116.74.177.193. The first device has IP 192.168.1.101:8000.



Port mapping in the router

External IP Address of Router	External Port	Internal IP address of Network device	Internal Port
116.74.177.193	8080	192.168.1.101	8000
116.74.177.193	8060	192.168.1.217	8000
116.74.177.193	8040	192.168.1.238	8000

Fig. 10 Port Forwarding

Another one is at the IP 192.168.1.217:8000 and the last one is at IP 192.168.1.238:8000. These IPs have been used locally. But the router has a function called NAT or Port-Forwarding which converts the local IP to a Public IP.

The table in Fig. 10 shows a conversion. The fact is that the ISPs in India do not allow port-forwarding and this is for the security of the user. The connection from local to public is possible but from public to local is difficult and sometimes not possible. This is one of the important reasons why pushing our local web-application over the internet becomes troublesome. Thus, it is important to have a solution to this problem. The solution found is NGROK tunneling. It creates a -random VPN and thus makes it possible to push a local website over the internet.

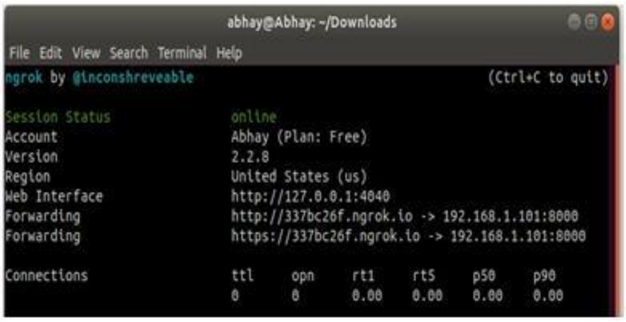


Fig. 11 NGROK: Local Tunneling

It is observed from Fig. 11 that the traffic visiting http://337bc26f.ngrok.io over the internet is sent towards the IP address 192.168.1.101:8000. Thus, the web-application is now globally accessible.

There are two scenarios of the Smart Parking system.

1) The user is not near the parking area

Part A:- In order to have a prior information, the driver logons to the web-application and checks the current parking status. As soon as the event is detected on the web application, the JavaScript talks to raspberry pi. The scenario being discussed is explained in Fig. 12.

Then the raspberry pi sends a query to the master for the latest status. The master (cluster node) creates a query to the slaves (parking nodes) and waits for the current time status. The slaves then senses the availability of the parking lots using the ultrasonic sensor. The ultrasonic sensor checks whether the constant obstacle is present or not. Then both the slaves send the status to the master. The master sends the record to the Raspberry Pi. The Raspberry Pi then stores the status in the Data logger and in the SQLite database. The raspberry pi further updates the status dashboard and the data logger dashboard. Based on that the driver can take a decision that the parking lots are available and thus, comes to park the car in that respective area. After that, he takes his car within the vicinity of the parking area.

Part B: - When the car is detected at the ultrasonic sensor of the master at the parking area entrances the master creates a query to the slaves and waits for the current time status. The slaves then sense the availability of the parking lots using the ultrasonic sensor. The ultrasonic sensor checks whether the constant obstacle is present or not. Then both slaves send the status to the master. The Master then checks the total current parking that is present. On that basis the master checks whether the car should be allowed to park or not. If the car is allowed, then the message is displayed on the LCD display which is near the entry of the parking area. The driver needs to authenticate himself using an RFID tag. When the car is detected in the parking lot, the respective slave send an update to the master. The master sends the record to the raspberry pi. The raspberry pi then stores the event and user identity in the Data logger and in the SQLite database. The raspberry pi further updates the status dashboard and the data logger dashboard.

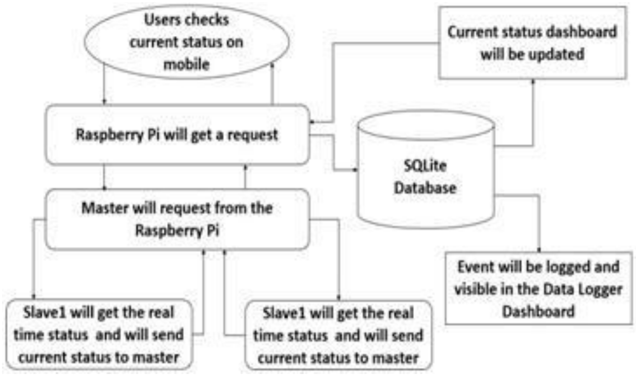


Fig. 12 User is not near the parking area

2) User is near the parking area

The situation is the same as in Part B. The same protocol is followed here which is explained in Fig. 13.

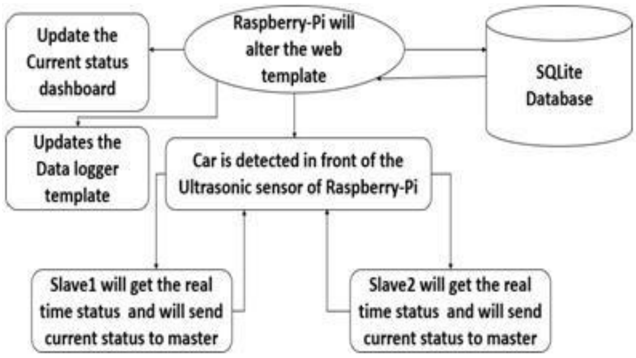


Fig. 13 User is near the parking area

In addition to the case study of the smart parking, concepts of indoor coverage and false alarms are also considered.

1) Indoor Coverage:

Every individual parking lot is accessible and traceable. The user also knows about the exact location where the user parked his/her car. Imagine that a car user goes for shopping near his/her home. Car user parked his/her car. As it is the best shopping mall in the city most of the people visited on the same day at the same timing. As the mall is popular, the parking area is very large. On the same day, the car user gets an emergency call from his/her boss for an urgent meeting. Thus, the user leaves his shopping beside and hurries to get to the car but then the user forgets the exact location where the car is parked. Then the consequence can also be imagined. Today's parking system generally uses a Ground Positioning System (GPS) to track the parking area, but inside the parking area, it cannot help to find where you parked your vehicle. The GPS has an indoor coverage issue. In this implementation, the problem has been an individual parking access to each parking lot. In our implementation it is possible because of the fixing of the micro-controller at each parking lot. Thus, when the car is parked at the parking lot it is possible to send the parking lot number and the exact position to the user, in which he had parked his car. Thus, in an emergency the indoor coverage is a great savior to the car driver and thus he is pleased by the parking service.

2) False Alarm (FA):

Submitting a wrong status to the database creates a wrong status on the web-app and the user is not able to get the exact and latest status from the web-app. The FA is taken care of by adjusting delays. If a status change is observed for a duration less that the threshold (4 sec) then, the SQLite database is not alerted. The purpose of the implementation is that the system should produce the real parking status, and, on that basis, the user of the system should be aware of the status and thus he can take further decision whether he should take his car for parking or not. Imagine even if the parking lot is empty, the current parking status dashboard shows the status to be "parked". On the other side though, the actual status is parked but dashboard is shows the status as "vacant". This situation can lead to more frustration for the user. This is the essential reason why this experimentation is worked on the solution. That the car driver should be happy to park, timing should be saved, fuel consumption along with the environmental pollution should reduce. Thus, in this case, taking care of such a false alarm is essential.

IV. EXPERIMENTATION AND RESULTS

In the research, both the implementation types are tried out. Some of them are as discussed below: -

1) Wired Implementation

In wired type of implementation, there are two main protocols.

a) Serial: - In the serial case, there are 4 USB ports present in Raspberry Pi. Thus, it is possible to attach 4 slaves to raspberry pi. Thus, only 8 Parking lots are possible in this case.

b) I2C: - In this case, attached 3 Arduino to each of the 4 Arduinos using I2C that are attached to the raspberry pi using serial. Thus, there are 24 total parking slots possible. But this is not the end, it is still possible to go for increasing the number of slaves to scale the system further.

2) Wireless Implementation

a) Using nRF24l01: To already connected 4 Arduinos, connect one nRF modules each which can further connect with the 6 Arduinos thus making 12 parking lots available. NOTE: For I2C just 3 ports of the Arduino which are GND, A4 and A5 are needed, further A4 and A5 are already not used. GND pin can be short circuited so that again it can be used for number of places. Whereas the nRF module needs 7 ports thus the complexity is more in the case of nRF24l01 than in I2C.

b) Using Wi-Fi: Raspberry Pi is an IP based microcontroller and has inbuilt Wi-Fi with total number of IP possible are 256. Among that, one will be Raspberry Pi itself and then the remaining 255 can be used for connecting the Arduinos using Wi-Fi, making 255 parking lots available.

Web-Interface provides the real-time parking status and gives a log of the car parking event for further security.

Fig. 14 Current status template for Smart Parking

The first case in Fig. 14 shows that all the four parking lots are vacant. Where green relates to vacant and red relates to occupied. Thus, in the second case two lots are vacant and two are occupied. Whereas in the third case it is observed that all four parking lots are occupied.

Fig. 15 Data Logger template for Smart Parking

The Data logger template is shown in Fig. 15. It gives an exact status and keeps logs for the events that occur in the smart parking system. It is important to have the details stored in the system so that if any crime occurs, the system owner should have the details of the event within the database. Thus, it is very important to have a data logger template. It is accessible for the system owner only.

V. CONCLUSION

All the layers of the architecture have been discussed and studied using the case study of Smart Parking. It is now possible to build any end to end IoT application. Assuming for 8 Parking lots, 9 ultrasonic sensors and 1 Raspberry Pi is used and 'n' is number of the slaves used. Comparison of wired and wireless prototype is illustrated in Table 1.

Table 1: Comparison between prototypes

Parameter	**Wired**		**Wireless**	
Protocols	Serial	I2C	nRf	Wi-Fi
Arduino	4	5	5	5
Scalability (s)	8	3n	2n	510
Delay (ms)	500	500	500	500

To conclude, if scalability is of concern, wireless type of prototype can be preferred and if the cost is concerned then wired prototype can be preferred.

VI. FUTURE SCOPE

The web-interface can be modified and improved. Some features like location tracking; payment option and the current parking status should reflect as it is. Also, the car detecting module should be improved such that no event should alter the status except the car parking event. As active RFID tags are costly, incorporation of the passive RFID tag can be done. Also, for tunneling purpose basic plan is used in this research. In the basic plan, domain which is used to display web application is randomly generated and is dynamic. But for business it is not preferred. Then it is needed to switch over the premium plan which offers custom domain and more parallel tunnel access.

References

[1] Noor Hazrin Hany, Mohamad Hanif, Mohd Hafiz Badiozaman, and Hanita Daud, "*Smart parking reservation system using short message services (SMS)"*, IEEE 2009

[2] Ashwin Sayeeraman, P.S.Ramesh, "*ZigBee and GSM based secure vehicle parking management and reservation system."*,Journal of Theoretical and Applied Information Technology, Vol. 37 No.2 , 2012.

[3] Manjusha Patil, Vasant N. Bhonge "*Wireless Sensor Network and RFID for Smart Parking System"* International Journal of Emerging Technology and Advanced Engineering ,Volume 3, Issue 4, 2013.

[4] Yanfeng Geng, and Christos G. Cassandras, "*New Smart Parking System Based on Resource Allocation and Reservations*", IEEE Transactions on intelligent transportation systems, VOL. 14, NO. 3, 2013.

[5] M. M. Rashid, A. Musa, M. Ataur Rahman, and N. Farahana, A. Farhana, "*Automatic Parking Management System and Parking Fee Collection Based on Number Plate Recognition.*", International Journal of Machine Learning and Computing, Vol. 2, No. 2, 2014.

[6] M. Chester, A. Fraser, J. Matute, C. Flower, and R. Pendyala, "Parking infrastructure: A constraint on or opportunity for urban redevelopment?A study of Los Angeles County parking supply and growth", J. Amer. Planning Assoc., vol. 81, no. 4, 2015.

[7] G. Revathi and V. R. S. Dhulipala, "*Smart parking systems and sensors: A survey*", International Conference on Computing, Communication and Applications, 2012.

[8] M. Y. I. Idris, Y. Y. Leng, E. M. Tamil, N. M. Noor, and Z. Razak, "*Car park system: A review of smart parking system and it technology*", Inf. Technology Journal vol. 8, no. 2, pp. 101–113, 2009.

[9] Pallavi Mane , Radha Deoghare, Samiksha Nagmote, Shubhangi Musle, Shraddha Sarwade, "*Android based Smart Parking System*" International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 5, 2015.

[10] Thanh Nam Pham, Ming-Fong Tsai, Duc Binh Nguyen, Chyi-Ren Dow, and Der-Jiunn Deng, "*A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies*", special section on emerging cloud-based wireless communications and networks, 2015.

[11] Pampa Sadhukhan, "*An IoT-based E-Parking System for Smart Cities*", International Conference on Advances in Computing, Communicatoins and Informatics, 2017.