IOT-BASED URBAN PARKING SYSTEM

Iana IVANCIUC[1*], Alexa ȚUGULEA[2], Viorel RÎNJA[1], Alexandru PRICOP[2]

[1]Technical University of Moldova, Faculty of Computers, Informatics and Microelectronics, Department of Software Engineering and Automatics, FAF-181 group, Chișinău, Republic of Moldova
[2]Technical University of Moldova, Faculty of Computers, Informatics and Microelectronics, Department of Software Engineering and Automatics, FAF-182 group, Chișinău, Republic of Moldova

*Corresponding author: Ivanciuc Iana, iana.ivanciuc@isa.utm.md

Abstract. *This article has the purpose to describe an Internet of Things project that will be implemented in order to solve the major problem of parking in urban areas in both developed and developing countries. It proposes a smart system for making the parking process easier and more secure. The main idea is the creation of smart parking using the Internet of Things and ultrasonic sensors, where available parking places could be displayed in a web/android application.*

Keywords*: IoT, parking, car, sensor, microcontroller, cloud, network, web/android application.*

Introduction

The world is in continuous progress that accelerates with astonishing speed. The population number increases repeatedly and the need for vehicles grows simultaneously. There is no doubt that car parking is a huge problem these days in both developed and developing countries. Many cities are suffering from lack of car parking areas with imbalance between parking supply and demand which can be considered the initial reason for metropolis parking problems. This imbalance is partially due to ineffective land use planning and miscalculations of space requirements during the first stages of planning [1]. Shortage of parking space, high parking tariffs, and traffic congestion due to visitors in search of a parking place are only a few examples of everyday parking problems. So, we decided to solve some of these problems by implementing an IoT project.

System Architecture

In many urban areas with high parking demand, when the government subsidizes its cost and freezes its apparent price at zero, there are many more people who want it than spots available. If a parking area is free, it does not stimulate turnover, people tend to take spots and hold on to them all day. As a result, drivers waste their time cruising, looking for scarce open space [2]. We propose to introduce a fee for this kind of parking areas. Our system will combine two main aspects. On the one hand, it will help drivers to find faster free parking slots, even to book a parking slot in advance, instead of waiting in the parking area or on the street. One the other hand, the government can set the price in such a way that people will pay as little as possible for parking without creating the cruising problem. The earned money can be used for urban infrastructure development.

The main data will be gathered by ultrasonic sensors which will be attached in every parking spot in order to indicate the availability. This data will be sent to the Arduino board, then to Raspberry Pi, which will make the connection with the Cloud via Network. Information such as spot availability, parking time of a car will be stored in a database and will be used in order to create User Interface - Web or Android application.

Figure 1 shows the block diagram of the proposed system.

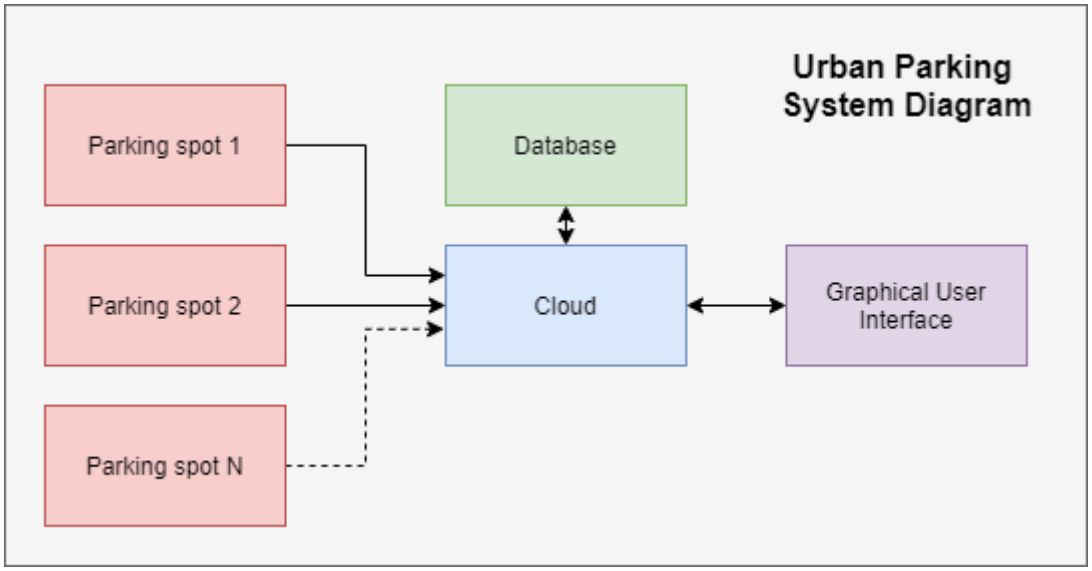

Figure 1. Block Diagram of Urban Parking System

Figure 2 represents the structure of Parking Spot Controller.

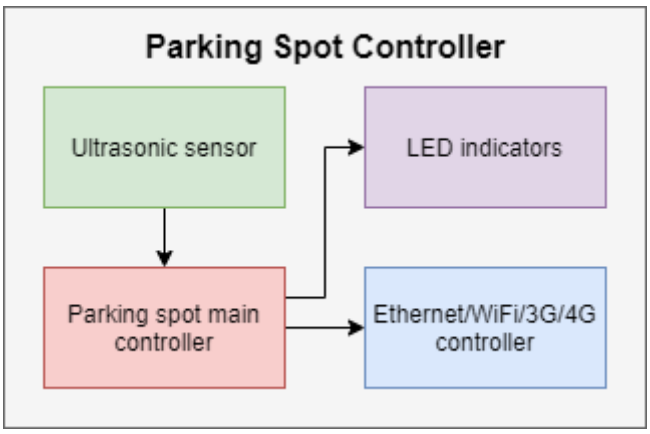

Figure 2. Block Diagram of Parking Spot Controller

Hardware

The system will need all kinds of hardware components, from sensors, LEDs and wires to MCUs. The needed items are listed below:

- *Ultrasonic Sensors* - work using sound waves, detecting obstacles and they are not affected by sunlight like Infrared Sensors (due to interference). For us, reliability is an important factor in our sensor selection, that is why we have chosen the Ultrasonic one [3].
- *LEDs* - there will be 3 types of LEDs: a. RED - the parking spot is currently in use; b. GREEN - the parking spot is available; c. YELLOW - the parking spot is booked (it will become green if the parking spot will not be taken by the booker in a certain period of time);
- *LCD Display* - used to show the number of available parking spots or to notify that there are not free spots (the drivers will know if there is it make sense to launch the application and to find an appropriate parking space);

- *Arduino Board* - it will collect data from ultrasonic sensors connected to parking places and send it to Raspberry Pi using a serial port [4].
- *Raspberry Pi Board* - the board will receive data from Arduino, make the network connection with the cloud and send all data there in order to use it in web/android application; [5]
- *Wires* - they will connect all hardware components and make data transmission between different layers possible;

Network

In order to connect the endpoint devices to the cloud, we need a way to provide a connection to the internet. We may assume that parking spots come in batches, so the endpoints will connect to a gateway through Wi-Fi modules, to a private WLAN (a Raspberry Pi), because it provides high connectivity for an area as wide as a common parking area.

Cloud

In order for system to operate, we will need a few entities to create the following structure:

- Database/Server which will manage requests and update the endpoints and GUI;
- Mobile application specific API primarily used for abstracted access to the parking spots sensors, as a location and availability struct and possibility to claim them as booked;
- Endpoint/Gateway specific API restricted to posting updates about various sensors states;
- Administrative application specific API, with more permissive CRUD requests;

To better describe the network scheme, let's analyze Figure 3.

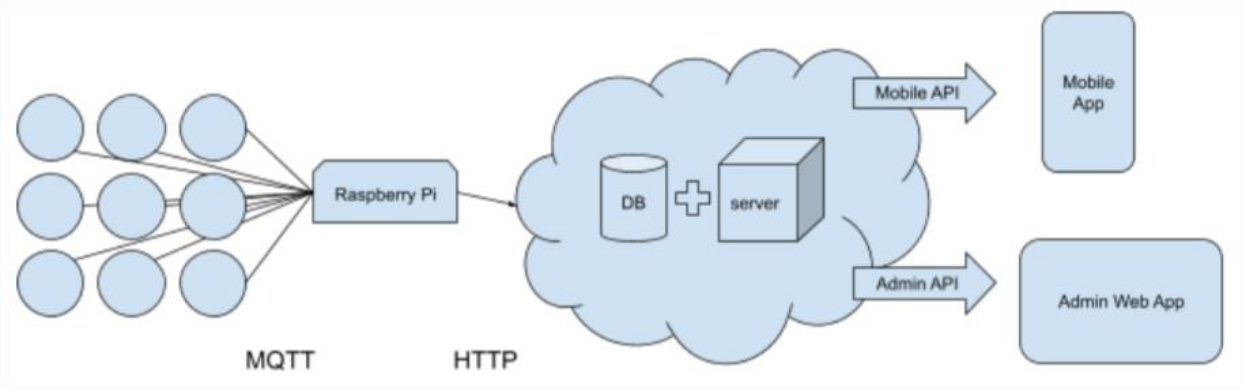

Figure 3. General Representation of the Network

It is worth noting that the Raspberry Pi is working as the gateway and uses MQTT [6] for efficient communicating with endpoints and http for RESTful requests [7] to the cloud.

Application

In our system, the mobile application is the interface that gives the user an overview about parking slot availability. This layer of the system is meant to consume the api provided by the cloud, to get data about various parking slots availability. Considering that car drivers are used to car specific applications (like taxi driver apps / city maps), to create a better user experience, the best way would be to comply with the driver's expectations. In such a case, the application should show a map, always centered at the current location and highlight near placed free parking spots. The user should be able to have the quickest access to a way to make a reservation to the nearest, automatically chosen, parking spot (one tap away, not to consume the driver's attention too much), or either book a specific spot. After choosing the spot, the user would be provided with a navigation

helper to drive to it, via conventional direction marks and guided voice assistance. After arrival, the user will be notified about 2SA and asked to confirm presence.

It becomes fairly straightforward that the application should be able to read and update data to the server about a spot availability, and combine it with data fetched from a map api, for custom spot pinning and directions.

Besides that, a separate application will be needed to easily administrate the new sensors connections and quick local fixes, which will provide a deeper access to the database and various running processes on the server, to the authorized users.

Conclusion

This paper contains the solution for a serious problem in urban areas – namely parking problem. In order to present the project from all perspectives, the paper includes the analysis of system architecture, system hardware, network connection and cloud integration. Also, it covers thorough description of User Interface (web/mobile application). The system is meant to solve traffic congestions and to ease drivers' life by helping them to find free parking spots during peak hours.

References

1. City Parking Solutions Throughout the Time. [online]. [accessed 04.03.2021]. Available: https://www.nwave.io/news/city-parking-solutions/.
2. Why Cities Should Implement Paid Parking? [online]. [accessed 04.03.2021]. Available: https://www.passportinc.com/blog/why-cities-should-implement-paid-parking/.
3. What is an Ultrasonic Sensor? [online]. [accessed 04.03.2021]. Available: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor.
4. Arduino, Overview. [online]. [accessed 04.03.2021]. Available: https://www.arduino.cc.
5. Raspberry Pi, Overview. [online]. [accessed 04.03.2021]. Available: https://www.raspberrypi.org.
6. MQTT, Overview. [online]. [accessed 04.03.2021]. Available: https://mqtt.org.
7. REST API, Overview. [online]. [accessed 04.03.2021]. Available: https://restfulapi.net.