International Journal of Engineering and Technology Innovation, vol. 9, no. 2, 2019, pp. 108-118

A New Parking-Space Detection System Using Prototyping Devices and Bluetooth Low Energy Communication

Karol Marso, Dominik Macko*

Faculty of Informatics and Information Technologies, Slovak University of Technology, Bratislava, Slovakia

Received 19 December 2017; received in revised form 28 June 2018; accepted 21 August 2018

Abstract

Parking in crowded city areas usually causes serious problems. Drivers spend an amount of time, effort, and fuel while finding someplace to park their vehicles. Although there are many solutions for free parking space indication; they are only suitable for parking lots inside the buildings, such as the shopping malls. However, there are not many solutions for exterior parking lots and for big cities as a whole. The purpose of this paper is to propose solutions to this problem, especially by considering issues like power consumption and communication approaches. A parking-space occupancy detection system is proposed, utilizing Bluetooth Low Energy communication and Arduino-based sensor devices. Raspberry Pi-based central devices redirect the collected data to the cloud-based database system, which makes them available for drivers to use mobile devices with Android systems. The selected components make the proposed system easily deployable and scalable.

Keywords: bluetooth, low energy, embedded system, internet of things, parking system

1. Introduction

The number of cars (or personal motor vehicles in general) is ever-increasing in the world, especially in developed countries and regions, where people can afford a car. It is not unusual that a family has more than one private car, and there are also many company cars available for their employees (e.g. for business trips or for personal use as well). The statistics provided by Eurostat indicate that there are about five hundred cars per thousand inhabitants [1].

Fig. 1 Red/green indicators inside parking garages [2]

This causes a huge problem not only for transportation infrastructure, but also for finding available parking spaces. People going anywhere by driving their own car even only for a short time, then they need to park the car. This problem gets even worse in cities, where it sometimes takes a huge effort to find a place to park a vehicle. It increases other traffic problems, such as traffic jams and air pollution in city centers. There are many existing solutions for parking lots in shopping malls or company

* Corresponding author. E-mail address: dominik.macko@stuba.sk
Tel.: +421-2-210-22-518; Fax: +421-2-654-20-587

buildings. For example, use red/green light-signs to indicate whether the parking space is currently used or is free for parking, as can be seen in Fig. 1. This solution can be used in the dark areas inside the buildings, where these lights can be spotted. Also, the lights can easily be provided with electricity from the existing infrastructure. However, this kind of solution faces serious challenges in exterior areas, where daylight causes the drivers not to see these lights. Therefore, the sensors are usually connected to navigation boards or even some smartphone apps, for indication of parking occupancy information. Another problem is that these systems are closed and don't communicate with each other, otherwise they can make finding a parking place much easier.

With the growth of the Internet of Things [3], finding a parking place is getting much easier. There are plenty of cheap devices with low power consumption that create small networks. These devices can communicate with each other in a citywide manner and provide useful information of free parking spaces or parking-capacity utilization. With such information, the system could provide various useful statistics, not only for drivers but also for city planners and government.

In this paper, a parking system architecture is proposed that can be scaled to the citywide area. The solution is utilizable for internal as well as external parking spaces. To demonstrate, a prototype using commonly known Arduino [4] and Raspberry Pi [5] devices is presented, and the communication is based on the widely used Bluetooth Low Energy [6] technology. In the next section, the related works and the existing solutions of the parking-space detection problems are summarized. The proposed system is described in Section 3, along with a prototype and simulation testing, accompanied by a discussion. Section 4 concludes the paper.

2. Related Works

Free parking spaces can be detected in many ways. One way is to detect them visually, using video cameras. There are existing works that are trying to use statistical models for predicting whether there is a vehicle in a parking space (e.g. [7-9]). The main problem of using cameras and image-processing to detect free parking spaces is that these devices are expensive, and the computations of the detection are too slow on cheap microcomputers. Therefore, such solution would not work with battery-powered low-power devices for the Internet of Things that communicate wirelessly.

Another way of detecting objects (including vehicles) is to use some less sophisticated sensors. A popular solution is to use ultrasonic sensors in parking buildings (e.g. [2]), which can also measure the distance to the objects. Even though an ultrasonic sensor can function precisely, it is sensitive to bad environmental conditions; snow or rain can cause the sensor not to work properly. Because of this, detecting parking spaces outside the buildings cannot be ultimately solved by using these sensors only; however, they can be combined with other sensors. For example, infrared sensors will not be influenced by the weather conditions mentioned above at all, as noted in [10]. The parking system for the Internet of Things [11] combines ultrasonic sensors with passive infrared sensors. Communication between these sensors is solved by Wi-Fi connection to a microcomputer Raspberry Pi. The parking system stores data in the cloud, thus it is a cheap and scalable approach. By using the smartphone application, the users can not only find free parking spaces but also pay for the parking. If a driver parks in a wrong place, the application would send a notification to the driver. In [12], infrared sensors are combined with magnetic sensors to provide an accurate solution. A combination of infrared, magnetic, and optical sensors is used in [13]. This solution is also unique by using solar energy to power the exterior sensor devices. Another solution [14] proposed a system with two modes – one for the interior (using an ultrasonic sensor) and another for the exterior (using a magnetic sensor). For communication, the system uses Bluetooth Low Energy. The system proposed in [15] uses RFID sensors for identification of vehicles and the Arduino device as a control unit, sending data to a cloud-server database.

Although there are many existing solutions, which are even being deployed into the commercial sphere, they are the most useful for small parking lots. On the other hand, there are only a few solutions for large cities. For example, [16] offer a community parking system that relies on sensors mounted in cars, which are able to detect free parking spaces. A passing car automatically sends information about a detected free parking space to the cloud, which then informs the nearby drivers about its location. It is a unique solution that does not require infrastructure; however, it requires the installation of sensor devices into the cars. The solution leads to many other problems. For example, if a car without the required sensor device is passing by a free parking space, it is undetected. Also, while the driver is moving to the detected free parking space, some cars without the sensor device can take the spot and the driver will not be notified.

There also exist reservation-based systems, such as [17-19], which do not rely on sensors to detect free parking spaces, but rather use some kind of a registration database. A driver reserves a parking space in a specific time by means of Android-based application, and the server-side of the system informs other users searching for a parking space about the unavailability of the reserved parking space during the period of the reserved time. The advantage of this system is that it can automatically bill the driver for parking, or even pay the fee using the smartphone. However, the system relies on the discipline of drivers to provide accurate data to the system, therefore it does not provide optimal utilization of parking spaces. For example, if the driver doesn't know how long he/she will park the car in advance, then the parking space may be reserved for a longer time than it is necessary. From an economic point of commercial parking lots, this is not a problem (the driver pays the entire time period); however, it is a huge problem, if other drivers cannot park the car because of this situation, the traffic congestion increases. Thus, such a system cannot be used for real-time occupancy information. Also, the system is closed, which means the parking spaces cannot be used by unregistered drivers (i.e. without the corresponding Android-based application) because it wouldn’t provide the correct information about occupancy.

Many different ways can be used for the communication between devices. A wired connection would be problematic in exterior areas, since it is required to connect all of the devices. Because of this, a wireless communication makes the solution more scalable, and thus it is more suitable. The most popular wireless technologies, such as Wi-Fi or Bluetooth, consume too much power; it is not suitable for battery-powered or energy-harvesting sensor devices with a limited amount of energy. However, the latest evolvement of Bluetooth standard is that there is a more power-efficient Bluetooth Low Energy technology [20].

The vision-based solutions for the detection of free parking spaces are not suitable for power-limited IoT applications because of their performance requirements. This is why the commercial solutions are mostly based on simpler sensors, such as infrared or ultrasonic and their combinations. However, the commercial solutions are closed and the information is not exchanged between parking lots using different parking systems. This is also a downside of usual Android-based parking systems, which rely on reservation databases to provide information about free parking spaces. In this paper, a more open solution is targeted that would be able to detect free parking spaces for drivers with a high precision; it could also provide useful data for cities (such as statistics of occupancy). One of the key contributions of the proposed system is that it is based on widely available and well-known devices using open software that do not charge license fees. It implies the high scalability because everyone can use this low-cost and open detection system to share parking-space occupancy information (e.g. personal parking spots, commercial parking lots, public parking spaces). Thus, it enables to share occupancy information in a citywide manner. The system can be used for interior and exterior parking lots, as well as for street-based public parking spaces. The system can even be combined with the existing parking systems to provide occupational information about administering parking spaces. The solution is proposed in the next section.

3. The Proposed System and Experimental Procedure

Based on the analysis of the existing solutions (summarized in Section 2) and available hardware components, the following requirements have been specified for the new system:

(1) Parking-space occupancy detection – the system should be able to track the occupancy of individual parking spots,

(2) Visualization of free parking spaces – the system should inform a driver where the car can be parked,

(3) Parking statistics – the system should provide historical and statistical data of the capacity utilization in time,

(4) Wireless communication – to maximize the scalability and reduce infrastructure costs and cabling problems,

(5) Low energy consumption – sensor devices can run on batteries for years,

(6) Easily available components – the system should be based on widely used well-known devices and open-source or free software without license fees.

The system proposed in this paper meets the specified requirements. The first requirement is achieved by selecting an appropriate combination of sensors, especially an infrared sensor for exterior areas and an ultrasonic sensor for interior usage. The second requirement is met by the development of an Android application, which presents detected information to drivers. The third requirement is fulfilled by providing local parking-lot statistics to its administrator; the global parking statistics to other users and municipalities. The fourth requirement is met by utilizing Bluetooth Low Energy as a communication technology that connects peripheral and central devices. The fifth requirement relates to the fourth one, as the low-power communication is used and the sensor devices are simple low-power end nodes. The selectors of system components adheres to the last requirement – well-known Arduino and Raspberry Pi devices have been selected, for which plenty of libraries, software, extensions, and user guides are freely available. The components and other aspects of the proposed system are described more thoroughly in the following subsections.

3.1. The system architecture and components

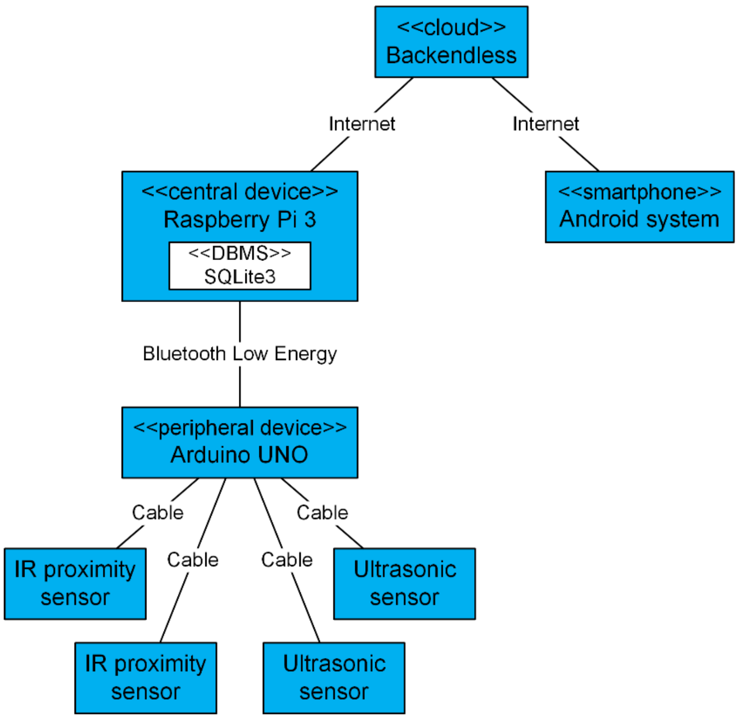

Fig. 2 The proposed system architecture

An overview of the proposed system architecture is illustrated in Fig. 2. There are two kinds of devices used:

(1) Peripheral devices - These devices collect data from sensors and are responsible for sending them to central devices. *Arduino UNO* has been selected as a device that would be suitable for the goal.

(2) Central devices - These devices need higher computing power and also some memory space to store all the data sent by peripheral devices. For these reasons, *Raspberry Pi 3* is used as a central device.

The sensors (an infrared proximity sensor for exterior usage combined with an ultrasonic sensor for interior usage) detect whether a parking space is occupied and are directly connected to the corresponding peripheral device, represented by Arduino. A single peripheral device can handle multiple parking spots (up to eight in the case of Arduino UNO).

Arduino [4] is a popular microcontroller, which has different versions. It is effective for collecting data from various sensors and has many extension shields that can connect it to other devices. For the prototype system, Arduino UNO has been used, which is cheap, easy to use, and has low power consumption. It can be easily extended using various shields, such as the Bluetooth Low Energy shield that was used in the proposed system. It enables to communicate wirelessly with a central device, represented by Raspberry Pi.

Raspberry Pi [5] is a microcomputer with many useful features. For example, the latest version (Raspberry Pi 3) can communicate by using Bluetooth Low Energy, which is built-in to this board. Raspberry Pi can be connected to the Internet using Ethernet or Wi-Fi. It can be used as a simple computer running a Linux-based operating system. Therefore, the *SQLite3* database management system (*DBMS*) can be installed; it enables to store higher amounts of data (such as important statistics) easily. Raspberry Pi has higher power consumption than Arduino UNO, but this is not really a problem since only peripheral sensor devices are powered by a limited energy source (e.g. a battery).

A central device sends data to the *Backendless* cloud server using the internet connection. A driver with an Android-based system can then use a *smartphone application*, which downloads the data from the cloud and displays demanded information about free parking spaces.

3.2. Collecting and storing the data

It is necessary to periodically collect the data about the parking-space occupation. Collecting and sending the data from sensors, using Arduino and the Bluetooth Low Energy shield, must be implemented in precisely selected intervals. Collecting the data too often would lead to a waste of power, and on the other hand, collecting them too rarely might cause the provided information is not the latest (i.e. it is inaccurate). Based on the power consumption of Bluetooth Low Energy [16], the ideal time would be around two seconds. Between the data collection time points, the sensor device is put into an energy-saving sleep state.

For clarification, Arduino reads the data from sensors every two seconds, writes these data into characteristics (i.e. a data structure used by Bluetooth Low Energy), and then switches to a sleep state. Raspberry Pi probes this data from all Arduino devices in reach, stores them into the SQLite3 database system, and every 25 seconds they are sent to the Backendless cloud server.

Advanced statistics for a parking-lot administrator are available in the local SQLite3 database. These statistics represent information about the day-wide capacity utilization of the parking lot as a whole, as well as information about the utilization of individual parking spots. The statistics available from cloud-based server represents citywide statistics about parking-spaces utilization. Such data can help to identify which parts of the city have an insufficient amount of parking spaces at the peak time, or which have enough capacity (or even the spare capacity). It can help with an urbanistic plan, or it can also be used by the parking-lots administrators to redirect drivers among each other.

3.3. The implemented prototype

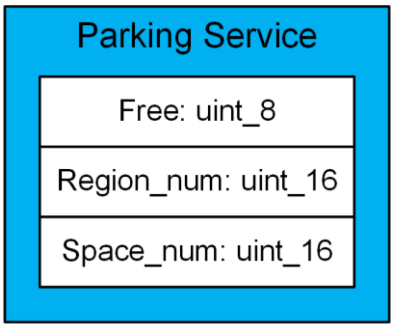

Fig. 3 Bluetooth low energy application services

Basic concepts of the proposed solution have been validated by implementing a prototype system, which operates as follows.

The communication between peripheral and central devices using Bluetooth Low Energy is accomplished using a custom profile, illustrated in Fig. 3. *Parking service* is an advertised service, which encapsulates three characteristics – Free, Region num, and Space num. The UUID (universally unique identifier) of this service is 5bab2ffaa-b355-4d8a-96ef-2963812dd0b8. *Free* is a characteristic that contains information about free parking spaces. *Region num* defines a region number, which can be used to identify different parking lots. *Space num* represents a unique number of the parking space in a certain region.

Such information is then filled into a *Spaces* structure, which is enriched by the identification number of the central device and timestamp of the information. Such enriched information is sent to the cloud, which also contains location coordinates of the registered parking regions. The region coordinates are used to locate the regions of the Google maps service.

To communicate with the Backendless cloud, the Rest API is used with the functions summarized in Table 1. Each Rest API message header includes these header parameters:

(1) application-id = A7B592CB-6F8B-B207-FFE8-3706A8A51100

(2) secret-id = AA29AB77-265B-016D-FF75-597FA2F18A00

(3) application-type = REST

Table 1 The used Rest API functions

Function	Description
GET /SPACES	Get all spaces
GET /SPACES?where= REGION_NUM%3DXX	Get spaces from the region
POST /SPACES	Insert new space
PUT /SPACES/:uuid	Update the parking space with the specified UUID

All of the requests were sent to URL http://api.backendless.com/v1/, which was further supplemented with the functions in Table 1. The *POST* and *PUT* requests have the same headers and, in addition, they have parameters obtained from sensors in their payloads.

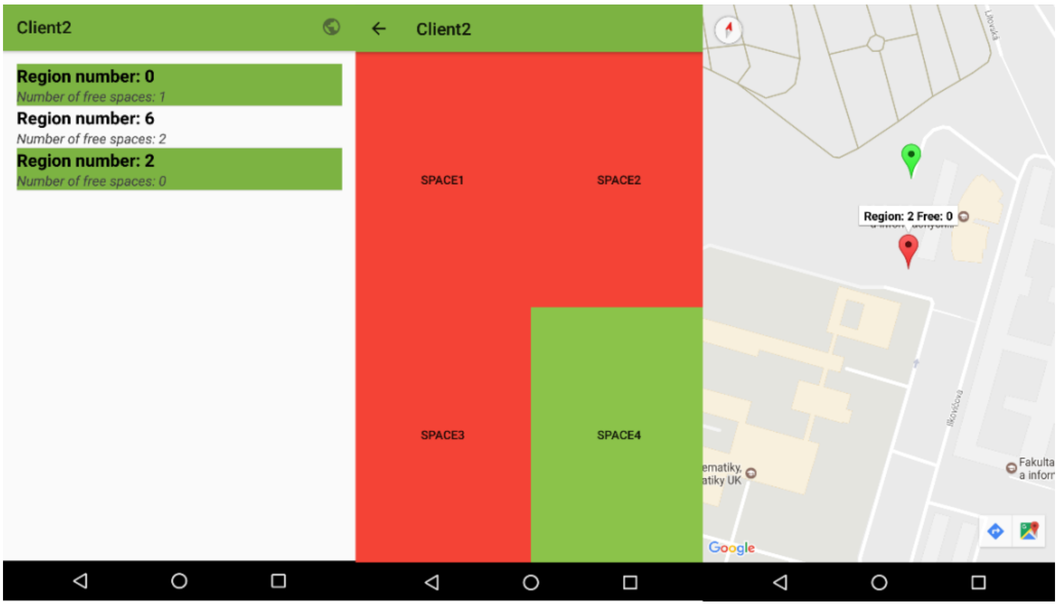

Fig. 4 Android application prototype windows

For a driver, a simple Android app has been developed, which displays region numbers along with a number of free parking spaces. An illustration of this app is provided in Fig. 4.

By selecting a region (the left part of Fig. 4), the app shows a detailed view that indicates which of the parking spots are not occupied (the green space in the middle part of Fig. 4). The app also utilizes a Google map-integrated localization function, which indicates where the parking lots are, and which one has the available capacity (using the typical red/green indication markers).

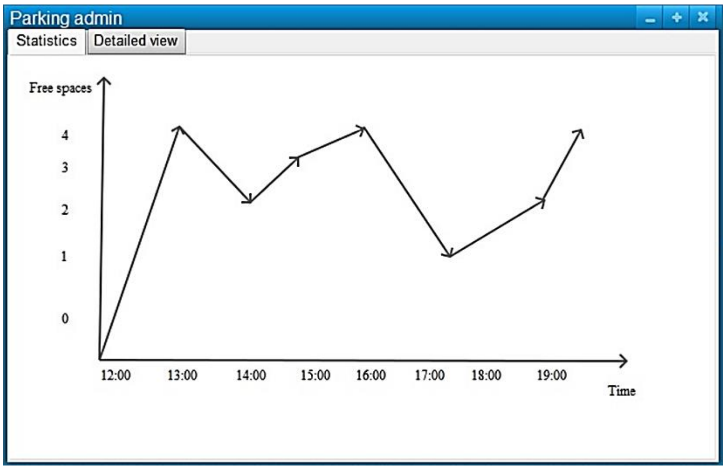

Fig. 5 The parking-lot administrator desktop application prototype

For a parking lot administrator, a desktop app with a graphical user interface was proposed, which can provide statistics, such as the number of free parking spaces during a day. A screenshot of the application prototype with the illustrated statistics is provided in Fig. 5.

3.4. Simulation-based testing

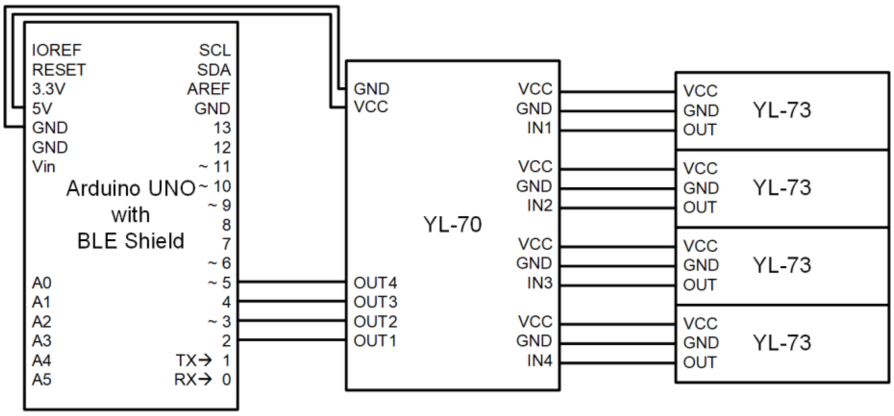

Fig. 6 Infrared proximity sensors connected to an Arduino device

The parking lots conditions have been simulated by using model vehicles. The test comprised of four parts – testing of vehicle detection, Bluetooth Low Energy communication, HTTP communication, and the Android application.

In order to connect infrared proximity sensors to Arduino, the Bluetooth Low Energy shield (RedBearLabs BLE Shield 2.1) has been firstly connected directly to the Arduino device (at least the pins 11, 12, 13 and 8, 9 should be connected to the same pins on Arduino). Then, supply ports and digital pins 2-5 have been connected to pins 1 – 4 on the YL-70 module, which supports four infrared proximity sensors (for each, VCC, GND, and INx pins were connected to VCC, GND, and OUT pins). A schematic view of the module interconnection is illustrated in Fig. 6.

After starting the Arduino, its Serial monitor was used to check whether the values are right in dependence on which sensor supposed to detect an obstacle (in the tested case it would be a vehicle). Apart from the infrared sensors, other sensors have been tried, such as ultrasonic sensors, vibration sensors, and photoresistors. However, only the ultrasonic and infrared sensors were able to provide enough information and successfully detect an obstacle.

After checking the sensors, the application on Raspberry Pi has been started. This application has successfully established the Bluetooth Low Energy connection with the Arduino communication module and read the available characteristics. It has been able to find these characteristics: free space, region number, and a number of spaces, which corresponds to the Parking service characteristics. After the obtained data were parsed, they have been checked whether they are correct when covering the sensor. From the result of the test, it has been concluded that Arduino has sent the correct data and Raspberry has received and processed them correctly.

When uploading the data to the Backendless server, Raspberry firstly downloads the data in order to get object IDs, which are used to update the information. An example of the obtained object ID during the test is provided below.

```
OBJECT:
0
3
1
FD28DEAE-4394-6630-FFBD-1F4B86058D00
```

After that, Raspberry either uploads (PUT) or inserts the data into the online database (POST). An example of such an operation during testing is provided below.

```
3
0
1
http://api.backendless.com/v1/data/SPACES/FD28DEAE-4394-6630-FFBD-1F4B86058D00
<Response [200]>
```

The response code 200 in the example above means that the HTTP request has been successful.

FREE INT	REGION_NUM INT	SPACE_NUM INT
0	0	2
0	0	1
0	0	0
1	0	3
1	6	0
0	6	1
1	6	2
0	6	3
0	2	0
0	2	1

Fig. 7 Comparison of the Backendless-stored data and the downloaded data in the Android application

The Android application loads the data from the Internet and displays them to drivers. After starting the application, it downloads the data from the Backendless cloud server. The data provided on the Backendless has been manually compared to the displayed information in the application, as illustrated in Fig. 7.

Based on the information in Fig. 7, the data are correctly interpreted (i.e. the numbers on the left correspond to the numbers on the right). When a region in the list has been selected by clicking, it has correctly displayed a screen with more detailed information. After clicking the back arrow, a test user has been navigated back to the list with regions. In the regions list, the user can click on a small icon of the globe. The user has consequently been navigated correctly to a map, where the regions have been visualized by markers. The marker was red when there was no free parking space available, and it was green when there was at least one free parking space. The locations for markers have been successfully downloaded from another database table in Backendless.

3.5. Discussion

The described work proposed a system that enables to collect important statistics about parking-space utilization in various regions of the city. Such data can be used by the municipality for better parking policies and urban planning. The offered solution enables to share parking-space occupancy information from the existing parking systems. It also offers to implement new parking-space occupancy detection sensor devices based on well-known Arduino Uno (i.e. low power, low cost) and shares the information using the proposed system. For parking lots administrators, the solution offers the Raspberry Pi-based central device, collecting the information from the sensor devices, sending them to the cloud storage, and providing statistics about the administered parking lot utilization.

Compared to the existing solutions, the proposed system is oriented to the citywide information about the availability of free parking spaces and statistics about their utilization in time (unlike the parking-lot oriented commercial systems). The occupancy detection is based on low-cost sensor devices that are able to run on batteries for years (unlike the vision-based solutions). Since the sensors are located in parking spots, they are independent of the vehicles (unlike the car-mounted sensors). In comparison to the reservation-based parking systems, the proposed solution provides real-time occupancy information and is available for everyone, also is independent of the drivers discipline to register and use the corresponding smartphone reservation applications.

However, the offered solution is not suitable as a replacement for the commercial parking systems. It does not enable to reserve a parking space or pay a fee for the parking-spot usage. It should be used in combination with such systems when it is needed, or it can be seen as an extension, providing occupancy information to the public and municipality.

There are other technological limitations of the proposed solution. A single Arduino Uno sensor device is able to monitor up to eight parking spaces. A single Raspberry Pi can collect information for up to 65536 parking spaces. The proposed cloud server can store the data from up to 65536 regions. These numbers could be scaled, but it would require a modification of the proposed data structures.

4. Conclusions

This paper is devoted to the problem of finding free parking spaces in big cities. More specifically, the proposed system is able to detect whether a parking space is occupied or not by using infrared and ultrasonic sensors. These sensors are controlled by a peripheral device, Arduino UNO in the tested prototype, which gathers data from up to eight sensors to process them and sends them to a central device (Raspberry Pi 3 in this case) using the Bluetooth Low Energy communication technology. The central device collects data from all peripheral devices in the range and stores them in the local SQLite3 database. These data are periodically updated to the Backendless cloud server via an internet connection, where the data are available for drivers using the proposed Android application for finding free parking spaces.

Besides this functionality, the proposed system gathers useful statistical information for parking-lot administrators (e.g. the occupation of individual parking spots in time), as well as global statistics for the municipality or government in a form of parking-capacity utilization per parking regions in time. It can be used to improve parking policies in cities (such as the limited-access areas in city centers or the setting of parking taxes) or for urban planning (e.g. how many new parking spaces must real-estate developers provide for new apartment buildings).

The advantage of the proposed system is that it is based on the widely available and well-known devices using the open software without any license fees. It can be used for interior and exterior parking lots, as well as for street-based public parking spaces. The system is scalable and each part can be easily replaced. The contribution is that the system enables anyone to build sensor devices (or to use the existing parking systems) and sent parking-space occupancy information to the cloud-based collector, where it will be available to the public (e.g. drivers, municipality). By implementing and testing the prototype, it has been proved that the system successfully detects free parking spaces and collects important parking statistics.

In the further work, there is a possibility to integrate navigation functionality into the system. It should be not very difficult, as the system already provides GPS locations of parking lots. Also, both applications (the Android app for drivers and the desktop app for administrators) need to be improved, since only prototypes have been developed for the testing purpose, which are not suitable for the industrial usage. Other Android-based solutions provide even the capability to reserve a parking space prior to the arrival and to pay the parking-fee using a smartphone. Although it is outside of the scope of the work, it also represents a possible extension in the future.

5. Acknowledgment

This paper was created with the support of the Ministry of Education, Science, Research and Sport of the Slovak Republic within the Research and Development Operational Programme for the project "University Science Park of STU Bratislava", ITMS 26240220084, co-funded by the European Regional Development Fund. This work was also supported by the Slovak Scientific Grant Agency (VEGA 1/0836/16), the Slovak Research and Development Agency (APVV-15-0789), and the Slovak Cultural and Educational Grant Agency (KEGA 011STU-4/2017).

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] "Passenger Cars in the EU," http://ec.europa.eu/eurostat/statistics-explained/index.php?title=Passenger_cars_in_the_EU#Overview, 2016.
[2] "Parking Lots Guiding System," http://www.tenetparking.com/parking.php?CateId=6, 2017.
[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645-1660, Jan. 2013.
[4] "Arduino," https://www.arduino.cc/, 2017.
[5] "Raspberry Pi: A Small and Affordable Computer that You Can Use to Learn Programming," https://www.raspberrypi.org/, 2017.
[6] nRF8001 Preliminary Product Specification-Bluetooth Low Energy, Nordic Semiconductor, 2011.
[7] Q. Wu and Y. Zhang, "Parking lots space detection," Machine Learning, Fall, 2006.
[8] R. Yusnita, F. Norbaya, and N. Basharuddin, "Intelligent parking space detection system based on image processing," International Journal of Innovation, Management and Technology, vol. 3, no. 3, pp. 232-235, June 2012.
[9] H. R. H. Al-Absi, J. D. D. Devaraj, P. Sebastian, and Yap Vooi Voon, "Vision-based automated parking system," 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), IEEE, October 2010, pp. 757-760.
[10] R. Grodi, D. B. Rawat, and F. Rios-Gutierrez, "Smart parking: Parking occupancy monitoring and visualization system for smart cities," in SoutheastCon, IEEE, July 2016, pp. 1-5.
[11] A. Khanna and R. Anand, "IoT based smart parking system," 2016 International Conference on Internet of Things and Applications (IOTA), IEEE, September 2016, pp. 266-270.
[12] SENSIT: Wireless Vehicle Detection Sensors, Nedap Identification Systems, 2008.
[13] "Solar Sensor: The World's Leading Parking Sensor," https://s3.amazonaws.com/frogparking/public/web/content/Downloads/Specification/Specification+Sheet+-+Solar+Sensor.pdf?uid=1
[14] C. Lee, Y. Han, S. Jeon, D. Seo, and I. Jung, "Smart parking system for Internet of Things," 2016 IEEE International Conference on Consumer Electronics (ICCE), IEEE, March 2016, pp. 263-264.
[15] T. N. Pham, M. F. Tsai, D. B. Nguyen, C. R. Dow, and D. J. Deng, "A cloud-based smart-parking system based on Internet-of-Things technologies," IEEE Access, vol. 3, pp. 1581-1591, September 2015.
[16] "Community-Based Parking: Helping One Another Find the Nearest Available Parking Space More Quickly," http://www.bosch-mobility-solutions.com/en/highlights/connected-mobility/community-based-parking/, 2017.
[17] S. Nagmote, P. Mane, S. Musle, and S. Sarwade, "A proposed automated car parking system based on Android," International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 4, no. 3, pp. 311-313, March 2015.
[18] Y. R. Dhumal, H. A. Waghmare, A. S. Tole, and S. R. Shilimkar, "Android based smart car parking system," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 5, no. 3, pp. 1371-1374, March 2016.

[19] Renuka R. and S. Dhanalakshmi, “Android based smart parking system using slot allocation & reservations,” ARPN Journal of Engineering and Applied Sciences, vol. 10, no. 7, pp. 3116-3120, April 2015.

[20] C. Gomez, J. Oller, and J. Paradells, “Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology,” Sensors, vol. 12, no. 9, pp. 11734-11753, August 2012.