IoT and XBee Based Central Car Parking Management System

Seda Ustun Ercan and Mohammed Sufyan Mohammed

***Abstract*— Parking in congested places frequently leads to major issues like air pollution, traffic jams, etc. This paper presents a prototype for an Internet of Things (IoT) based central car parking management system that attempts to minimize car parking problems. The proposed system is relatively scalable and consumes less power compared to other parking systems. This system uses Arduino microcontroller-based TCRT5000 infrared sensors to detect vehicles in the parking spaces. XBee peripheral modules are located in different parking lots at different locations to transmit the sensed data to XBee central module that is placed close to the parking lots. The central node then updates all data to the Blynk IoT platform via Wi-Fi wireless technology present in the NodeMCU board. The availability of parking lots can be viewed using a smartphone through Blynk application.**

Index Terms— **Blynk platform, central parking system, Internet of Things, NodeMCU, XBee.**

I. Introduction

PARKING OF vehicles in the parking area is becoming a hard task as the number of vehicles raises while the amount of parking spaces is limited. The outcome would be that the drivers would expend some time looking for parking places, slowing down vehicle flow and increasing traffic congestion. Because parking information is not readily available to drivers seeking for available parking spaces, there is a situation where traffic is backed up in parking lots and hunting for parking places. [1].

Different potential measures have been used to develop car parking management systems such as building new parking facilities which is rather expensive or designing automated parking system where mechanical or hydraulic car lifts regulate vehicles in the parking spaces [2]. An alternative to these very costly approaches is to augment existing parking facilities with smart systems that can allow more efficient use of parking facilities, as well as allowing the drivers to easily and quickly find available parking space. For example, use green/red light marks to indicate if parking space is currently occupied or free for parking [3]. This solution can be used in dark areas such as parking lots in shopping malls or company buildings, where these marks can be spotted. Such a solution faces some difficulties in exterior areas, like daylight causes the drivers not to recognize these marks light. Therefore, the detecting sensors are commonly connected to display screens or navigation boards, to indicate parking occupancy information [4]. However, this kind of systems are closed and do not communicate with each other, otherwise they can make finding parking spaces much easier. With the growth of Internet of Things (IoT) technology, finding available parking spaces is becoming much easier [5].

There are a lot of cheap and low-energy devices that create IoT networks. These devices have the ability to communicate with each other and provide helpful information on available parking spaces in a citywide manner. With such useful information, the IoT based parking systems could provide various necessary statistics, not only for the drivers and parking managers but also for municipalities or city planners. Several different methods can be used for communication between IoT devices. A wired connection could be an issue in external areas, as it is required to connect all the relevant devices. For this reason, a wireless connectivity makes the solution convenient and more scalable. The most common wireless communication technologies, such as Bluetooth or Wi-Fi, consume a lot of power and they are not appropriate for battery-powered devices. The recently launched zigbee technology employs the IEEE 802.15.4 standard and is a revolutionary, all-encompassing wireless communication technology [6]. When compared to other wireless networks like Wi-Fi and Bluetooth, Zigbee is one of the finest technologies to employ because it has many features like a very low power consumption, less expensive, simplicity to develop and deploy, support of large number of nodes, strong security, and high data dependability additionally, it has self-healing, self-routing, and fault tolerance capabilities [7]. It supports point to point and point to multipoint topology and also supports mesh network topology.

Different smart car parking management systems [8-13] have been suggested which utilize various modules like Arduino ethernet microcontroller or Wi-Fi based boards to transfer the

SEDA USTUN ERCAN, is with Department of Electrical-Electronics Engineering, Ondokuz Mayıs University, Samsun, Turkey, (e-mail: sedau@omu.edu.tr).

https://orcid.org/0000-0002-8688-5852

MOHAMMED SUFYAN MOHAMMED, is with Department of Electrical-Electronics Engineering, Ondokuz Mayıs University, Samsun, Turkey, (e-mail: mohammedsufyan1986@gmail.com).

https://orcid.org/0000-0002-6299-9003

Manuscript received August 11, 2022; accepted December 28, 2022.
DOI: 10.17694/bajece.1160300

sensed data into the cloud. These Arduino or Wi-Fi boards are placed in each parking lot. This certainly increases the cost due to the necessity to provide an internet connection for each Wi-Fi board. In addition to, it raises the power consumed in general. In the proposed parking system, instead of Wi-Fi module, we keep XBee module at each parking lot, and one as a central node that gathers all the sensed data and transfers it into the cloud using Wi-Fi wireless technology. Only the central node needs an internet connection to transmit the information to the cloud. So, the cost and power consumption are reduced. In another project, a real-time system that enables drivers to effectively discover and limit the vacant parking spaces remotely through website and Android App is designed and deployed [14].

The fundamental benefit of mesh networks is that other nodes may continue to communicate without interruption even if one of the links breaks. The hardware that transmits the Zigbee communication protocol is called XBee, and it is made by the digi international firm [15]. As seen in Figure 1, the XBee module. It often serves as both a transmitter and a receiver in wireless communications. XBee modules typically operate at 2.4 GHz with a 250 Kbps speed and a variety of antenna types that allow them to span a range of distances. Although the data rate of XBee module is lower than Wi-Fi, it has a wide application where the transmitted data is less and the power consumption is an issue of concern. The XBee modules contain sleep modes to increase battery life, allowing them to operate over several years on low-cost batteries. Connecting the XBee shield to the Arduino microcontroller is required.

Fig.1. XBee module

In this paper, we designed and implemented a prototype of a central car parking management system based on low power XBee wireless devices and IoT wireless technology. The proposed parking system assists drivers to find vacant parking spaces with the reservation facility. Unlike other studies, for the first time, the proposed parking system is usable in both external as well as internal parking facilities and thus the that system could solve all the previously mentioned problems to some extent.

II. SYSTEM OVERVIEW

Several requirements have been identified for the proposed parking system such as : (1) Parking lots occupancy detection – the proposed system must be able to detect the occupancy of individual parking lots. This requirement was achieved by choosing an appropriate set of sensors, specifically infrared sensors. (2) Visualization of available parking lots – the proposed system should notify the driver of the state of parking lots in a visible manner. This requirement was fulfilled by the development of a graphical user interface (GUI) on Blynk application (app), which presents information related to the parking lots for the drivers. (3) Wireless communication, (4) Low power consumption – to provide scalability and minimize cabling challenges and infrastructure costs. The third requirement was achieved by using Zigbee low power as a wireless communication technology that connects all system nodes. For the fourth requirement, it is largely related to the third requirement, where the low power zigbee technology was used and the selected devices are low power, simple and could run on batteries for a sufficient period of time. (5) Readily available software and components – the proposed parking system must be based on free or open-source software without license fees and well-known or widely used devices. The proposed system devices or components meet the fifth requirement as they have been selected, for which plenty of extensions, user guides, software and libraries are freely available. An overview of the proposed parking system architecture is illustrated in Figure 2. There are two kinds of nodes in the proposed parking system: Peripheral nodes and Central node.

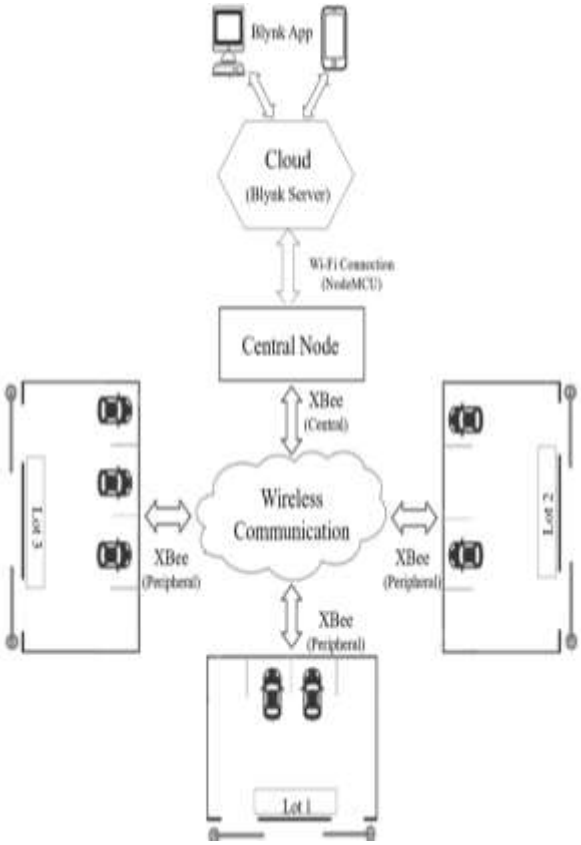


Fig.2. The proposed parking system architecture

Peripheral nodes are located at each parking lot to detect the presence of the vehicles in that area and to report the status to the central node constantly. At each peripheral node, there are Arduino Mega microcontroller, TCRT5000 infrared (IR) sensors, servo motors, liquid crystal display (LCD), matrix keypad, barrier gates and XBee module. Arduino [16] is a common microcontroller, which has many versions. It is

effective for gathering data from different sensors and has plenty of extension shields that can connect it to other devices.

For the proposed parking system, Arduino Mega 2560 development was used, which is cheap, easy to use, and has large storage capacity and low power consumption. It can be easily extended using various shields, such as XBee shield that was used in the proposed parking system. The Arduino microcontroller was programmed using the free and open-source Arduino software IDE, which makes it simple to create code and upload it to the board. The total number of parking spaces occupied in the parking lot was shown on an LCD. Check-in and check-out were controlled by barrier gates, which were driven by servo motors.. Matrix keypad to key in the passwords was used. IR sensors were used to detect the presence of the vehicles at the entrance and exit of the parking lot, and also were used to detect the state of each space within the parking lot. The IR sensors detect if the parking lot is available or not and are directly connected to the corresponding peripheral node, represented by the Arduino Mega microcontroller. A single peripheral node can handle multiple parking spaces at the same time. The IR sensors measure the distance of the existing vehicle. If the measured distance is less than the specified threshold value, then the parking space is identified as occupied. Flags were set for each parking space (flag=1 if occupied, flag=0 if available). To distinguish parking spaces information in each parking lot, the peripheral node sends the flag values along with the name of the parking lot (lot, lot2, etc) to the central node which transfers the sensed data to the cloud server. The central node acts as a central controller that records peripheral nodes data and controls the overall system. This node requires higher computing energy as well as some memory space. For these reasons, the Arduino Mega microcontroller was used as a central node as well.

XBee modules were used to provide communication and information exchange between the central node and peripheral nodes. XBee at the central side receives the data from multiple XBee modules at the peripheral side simultaneously. The peripheral XBee is installed in different parking lots to transfer the sensed information to the central XBee without the need to use internet or Wi-Fi connection at every parking lot. In other words, Wi-Fi protocol is used only at the central node to transfer the received data to the cloud. XBee modules could communicate with each other wirelessly and it could exchange the information over a wide distance up to 2 km range in this case. The central XBee should be at that range to receive the information from multiple peripherals XBee. For illustration, if there are three parking lots, instead of three Wi-Fi modules, we keep three XBee peripherals and one XBee central. Then we specify a point which is roughly within the specified range from the three parking lots and we install the central XBee at that point. XCTU software was used for the configuring of XBee modules. The central XBee feds the gathered data to the central node which transfers it to the back endless cloud based Blynk server using Wi-Fi connection through the internet. The cloud server acts as a database to store all the necessary records related to the parking lots and the drivers that have access to the parking system. NodeMCU module was used in the proposed parking system to connect the central node with Blynk server. NodeMCU is an open source, simple, programmable and Wi-Fi enabled module designed for IoT purposes. It includes firmware that runs on the ESP8266 Wi-Fi system on chip. The ESP8266 is a Wi-Fi chip with microcontroller capability and full TCP/IP protocol stack that allows any microcontroller to access the Wi-Fi network. There are many kinds of ESP8266 modules, but for this proposed parking system the ESP12-E NodeMCU module was used and was programmed through Arduino IDE software as well [17]. The implementation of the proposed central and peripheral nodes on a breadboard is illustrated in Figure 3 and Figure 4.

Fig. 3. Proposed central node hardware

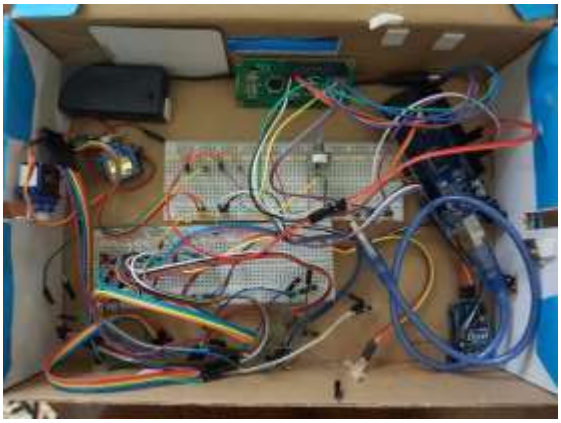


Fig. 4. Proposed peripheral node hardware

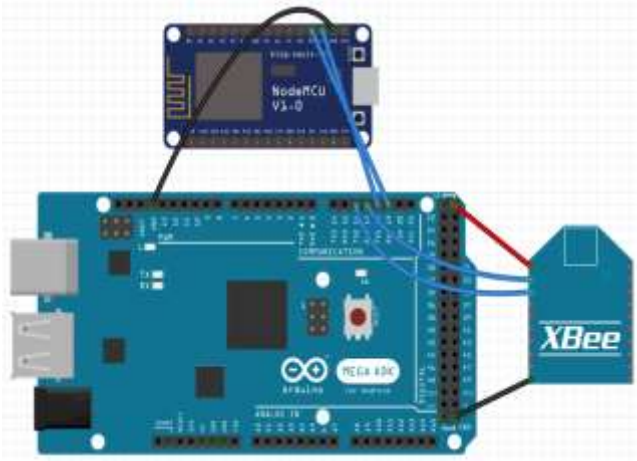


Fig. 5. The proposed central node diagram on fritzing

The interconnection between different electronic components of the proposed central and peripheral nodes are illustrated in Figure 5 and Figure 6. It was designed using fritzing environment. It is worth noting also that the system nodes are powered from separate power sources.

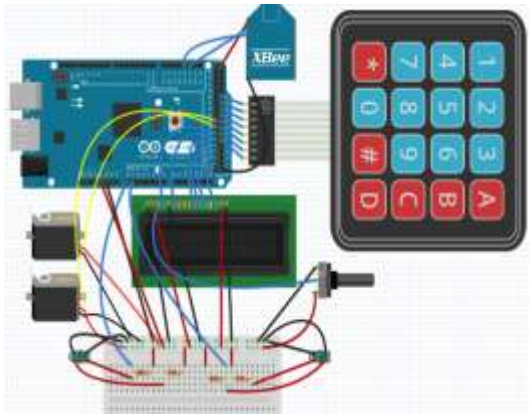


Fig. 6. The proposed peripheral node diagram on fritzing

III. BLYNK INTERNET OF THINGS PLATFORM

IoT technology refers to anything physical that is connected to the internet or exchanging data or information between physical devices and the internet. Blynk is a platform was designed for IoT purposes. It is compatible with both iOS and Android apps and can interact with various microcontrollers such as Arduino, Raspberry Pi, NodeMCU etc. It is able to remotely control devices over the internet and store, display and visualize data from different sensors for any desirable system. There are three main components in the Blynk IoT platform: (1) Blynk Application – It allows creation of graphical user interfaces for any project using different widgets that are provided. (2) Blynk Server – It is an open-source cloud based server, could easily handle thousands of devices and is responsible for all the communications between the smartphone (Blynk app) and the selected hardware. (3) Blynk Libraries – It enables communication, for all the popular hardware platforms, with the server and processes all the incoming and outcoming commands [18].

For the proposed parking system in this paper, a simple GUI was designed and developed using Blynk app, which displays parking lots names along with the total number of available parking spaces in each parking lot. Step-by-step procedures in designing the GUI Blynk app are illustrated in Figure 7. Firstly, we need to download and install Blynk app for android at Google plays store or iOS at app store. For the proposed parking system, two pieces of Value Display tool from Widget Box were used, each one to display the number of vacant parking spaces within each parking lots, as shown in Figure 7 (a). The Value Display tool displays incoming data from sensors or Virtual Pins. By clicking on Value Display tool for a one time, we can prepare it, as shown in Figure 7 (b). One of the Value display tools was labeled as Parking Lot A while the other one was labeled as Parking Lot B, as shown in Figure 7 (c). It can be observed that the two value display tools are related to the same SOURCE which is the Central Car Parking. Then from Input/PIN/Select pin we need to choose Virtual and from PIN we need to choose a virtual port number to provide the communication with the selected device. Virtual pins are a way to exchange any value or data between Blynk app and the devices. It could sent the values as strings and there are no practical limits on it. The process of sending any value or data from the selected device to Blynk app is done through a function called Blynk. VirtualWrite(Virtual Pin Number, Value) function, while any value or data from Blynk app can be sent to the selected device using Blynk_Write (Virtual Pin virtual pin's number as a parameter in the functions. Also, param.asInt () must be called inside the function, which returns an integer (It could be boolean or byte as well) value, that must be processed to execute other sentences in the main code. In the proposed parking system, V7 virtual port was used for parking lot A and V8 was used for parking lot B, as illustrated in Figure 7 (d) and Figure 7 (e) respectively. So, when XBee central receive data from XBee peripheral, it feds it to the Arduino Mega at the central node for processing. If the received data is related to the parking lot A, the central node transfers the processed data to the Blynk app through virtual port V7, and if the data is related to the parking lot B, the central node sends it through virtual port V8 to the Blynk app. The designed GUI on Blynk app dashboard with the Value Display tools is shown in Figure 7 (f). And with this the Value Display tools settings have been ended.

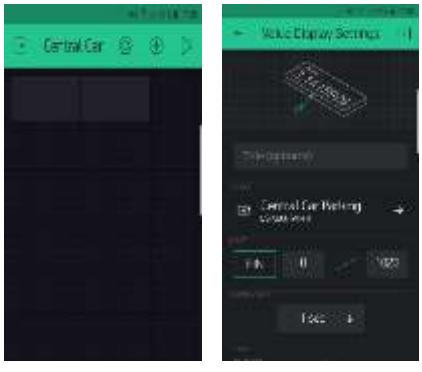
The designed GUI on Blynk app also includes parking space reservation option by clicking the reservation button in the designed GUI associated with the desired parking lot. For reservation process purpose the Styled button from Widget Box was used. The Styled button operates in switch or push modes. It allows sending any number value on button press and button release events. By default, the Styled button uses 1/0 (HIGH/LOW) values. The button sends 1 (HIGH) on press and sends 0 (LOW) on release. A title for ON/OFF LABEL was given and V9 virtual pin was used for the Styled button, as shown in Figure 7 (g). It can be seen that the Central Car Parking is the TARGET. After the reservation request is created, the boolean data (low or high) travels to the central node through Blynk server and NodeMCU module. The Arduino microcontroller then processes the received data and begins start to check whether there is any vacancy in the desired parking lot. In the case of the desired parking lot is not available means the central node will send a notification or a statement to the Blynk app dashboard telling that there is no available parking space. In the case of the desired parking lot is available means the central node will send a reservation acknowledgment message along with the entry password to the respective parking lot using XBee modules. The total number of occupied parking spaces at the desired parking lot will increase by one.

At the same time, the central node will update the desired parking lot information and the entry password to the Blynk app. A timer will be started automatically for that reservation. Before the timer expires the driver has to arrive at the desired parking lot and enter the entry password, otherwise the reservation process will get expire and in this case, the information of that desired parking lot will be updated and the updated information will be sent to the central node using XBee modules as well. The central node will be updated the information of the desired parking lot to the Blynk app again. If the driver arrives at the specified time means, the driver has to

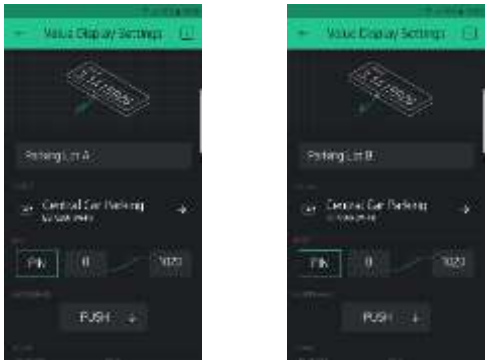
enter the corrected password in order to access the desired parking lot. If the entered password is correct, the barrier gate will get open and allow the driver to enter parking lot area. We have used the labeled Value tool from Widget Box for displaying the entry password and reservation statement. The Labeled Value tool displays incoming data from Virtual Pins or different sensors. It is a better version of the Value Display tool because it contains a formatting string. It was named as "Reservation" and V10 was used as a virtual pin, as shown in Figure 7 (h). It can be seen that the Central Car Parking is the SOURCE, and the virtual pins V7, V8 and V9 are already in use. The designed GUI on Blynk app dashboard with the tools used is shown in Figure 7 (i) where it can be seen all the tools with their names and their virtual ports.

There are a few changes that must be made to the main code or program before uploading. At first, to interface with Arduino software, Blynk library needs to be installed at the library folder of Arduino software IDE. Also, we need to use or to pre-install some other libraries like #include<ESP8266 Wi-Fi.h> library, #include <SPI.h> library, #include <SimpleTimer.h> etc, to programing using Arduino IDE. Node MCU 1.0 (ESP 12-E Module) module has to be selected as the board type. The COM port has to be chosen after connecting the selected device to the computer. The baud rate also must be set up to monitor the output of different devices in the serial monitor of Arduino IDE.

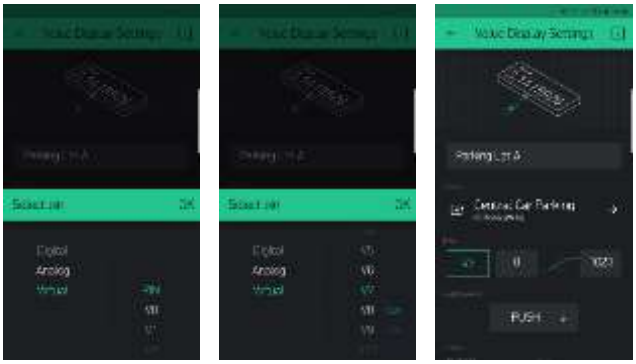
The Wi-Fi SSID and the Wi-Fi password should be changed by the actual Wi-Fi SSID and Wi-Fi password. The Auth Token also needs to be modified in the source code as it is given in the Blynk account. After checking all the connections, adding all the required library files to the Arduino IDE, setting the right ports of the computer and after completing all the steps, the main code has to be compiled and then uploaded to the selected module. After compiling and uploading the Arduino code, the peripheral node starts sending the data from IR sensors to the central node using XBee modules. The central node transfers the data to the Blynk cloud server and Blynk app through Wi-Fi connection. Some of the results of the designed parking system obtained on Blynk app are shown in Figure 8 (a) (b) (c). As an illustration, fig 8 (a) indicates that there are three parking spaces are available in the parking lot B while there are only one parking space is available in the parking lot A.



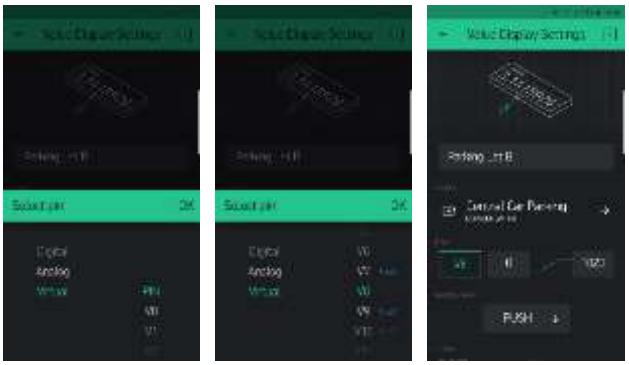
(a) (b)



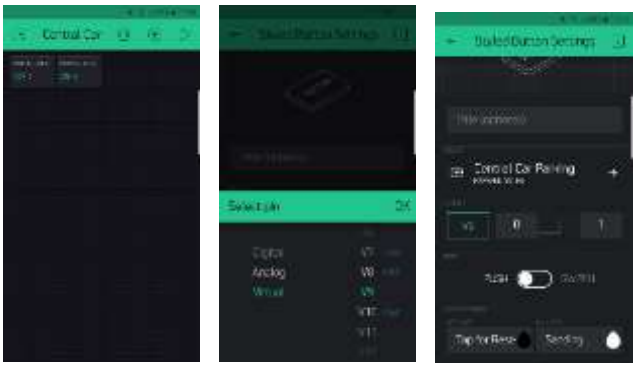
(c)



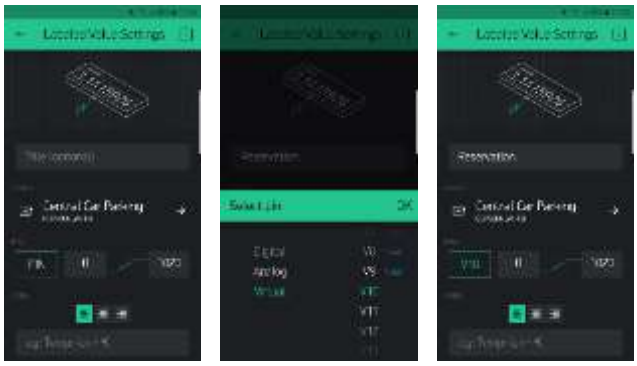
(d)



(e)



(f) (g)



(h)

(i)

Fig. 7. Illustrates the design stages of the GUI on Blynk app used in this paper

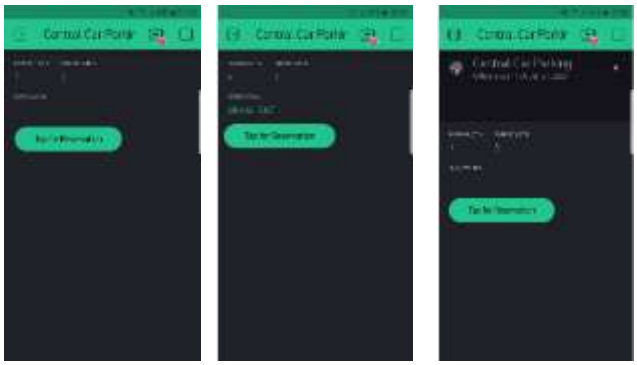


Fig 8. Blynk app showing availability status of parking lots

IV. CONCLUSION

The main purpose of this study is to suggest solutions to the problem of looking for available parking spaces in urban areas, particularly by considering issues such as communication approaches, power consumption and financial cost. More specifically, a prototype of a central car parking management system was designed and implemented based on IoT technology. The proposed parking system is capable of detecting available parking lots for the drivers with high accuracy. It can be applied for the street-based parking lots, as well as interior and exterior parking lots. Moreover, the proposed parking system can be combined with the present IoT based parking systems to provide information about managing parking lots in citywide.

One of the main contributions of the suggested parking system is that it is based on widely available devices utilizing well- known software and components that do not charge license fees. The selected devices make the proposed system scalable and easily deployable as each component can be replaced easily. In the tested prototype of the proposed parking system, Arduino Mega microcontroller was used which appeared to be cost effective and compatible with XBee modules with easy installation or programming and maintenance.

The prototype of the proposed parking system was implemented and tested for various scenarios and accurate results were obtained. The results of the intensive experiments proved that the proposed system successfully detects the presence of the vehicle at each parking lot, and updates the sensed data to the central node and Blynk app effectively. Based on that we conclude that the successful implementation of a central car parking management system can significantly reduce the problem of fuel, time and effort wastage, and pollution caused by blind searching and traffic congestion to a considerable amount.

However, the designed GUI on Blynk app needs to be improved as the prototypes were designed and developed for testing purposes only, and it is not suitable for commercial use. Other IoT based parking systems provide the ability to reserve parking spaces extensively and to pay the fee using a smartphone. In further work, we would develop an application for Andriod and iOS and test its usability in the real-time environment. Also, the integration of the navigation function into the proposed system is a possible extension in the future work.

REFERENCES

[1] L. Anjari and A. H. S. Budi, “The development of smart parking system based on NodeMCU 1.0 using the internet of things,” IOP Conference Series: Materials Science and Engineering, International Symposium on Materials and Electrical Engineering (ISMEE), Bandung, Indonesia, 16 November 2017.

[2] T. Bhanusri and K. P. Rao, “Advanced Car Parking System with GSM Supported Slot Messenger,” *Journal of Electronics and Communication Engineering (IOSR-JECE)*, vol. 10, issue 1, pp. 14-18, February 2015.

[3] A. Kianpisheh, N. Mustaffa, P. Limtrairut, and P. Keikhosrokiani, "Smart Parking System (SPS) Architecture Using Ultrasonic Detector," *International Journal of Software Engineering and Its Applications.* vol. 6, no. 3, pp. 55-58, July 2012.

[4] P. B. Natarajan and S. K. Ghosh, “Design and Implementation of Smart Car Parking System Using Lab VIEW,” *International Journal of Pure and Applied Mathematics*. vol. 120, no. 6, October 2018, pp. 329-338.

[5] A Khanna and R. Anand “IoT based smart parking system,” IEEE 2016 International Conference on Internet of Things and Applications (IOTA), Pune, India, 2016, pp. 266-270.

[6] H. C. Yee and Y. Rahayu, “Monitoring parking space availability via ZigBee technology,” *International Journal of Future Computer and Communication*. vol. 3, no. 6, pp. 377-380, December 2014.

[7] A. Asaduzzaman K. K. Chidella and M. F. Mridha, "A time and energy efficient parking system using Zigbee communication protocol," IEEE Southeast Con 2015, Fort Lauderdale, FL, USA, 2015, pp. 1-5.

[8] P. Sadhukhan, "An IoT-based E-parking system for smart cities," IEEE 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 1062-1066.

[9] A. Somani S. Periwal, K. Patel and P. Gaikwad, "Cross Platform Smart Reservation Based Parking System," IEEE 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, 2018, pp. 1-5.

[10] B. M. Mahendra, S. Sonoli, N. Bhat, Raju and T. Raghu, "IoT based sensor enabled smart car parking for advanced driver assistance system," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 2188-2193.
[11] A. Gupta, P. Rastogi and S. Jain, "Smart Parking System using Cloud based Computation and Raspberry Pi," 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) India, 2018, pp. 94-99.
[12] S. Mendiratta, D. Dey and D. Rani Sona, "Automatic car parking system with visual indicator along with IoT," 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, 2017, pp. 1-3.
[13] G. Aravindh and M. A. Kumar, "An Efficient Car Parking Management System using raspberry-pi," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 154-158, doi: 10.1109/ICISS49785.2020.9316049.
[14] A. Anand, A. Kumar, A. N. M. Rao, A. Ankesh and A. Raj, "Smart Parking System (S-Park) – A Novel Application to Provide Real-Time Parking Solution," *2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT)*, 2020, pp. 93-96, doi: 10.1109/MPCIT51588.2020.9350429.
[15] https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu
[16] https://docs.arduino.cc/static/de1277b5f061164f715cb610870edb1a/A000067-datasheet.pdf
[17] https://www.nodemcu.com/index_en.html#fr_54747661d775ef1a3600009e
[18] https://blynk.cloud/dashboard/register?_ga=2.141173108.732515720.1660131406-1285698961.1660131406

BIOGRAPHIES

SEDA USTUN ERCAN was born in Turkey. She received her B.S, M.Sc. and Ph.D. degrees in electrical and electronic engineering from Ondokuz Mayıs University (OMU), Samsun, Turkey, in 2005, 2009 and 2016, respectively. She is currently an Assistant Professor with the Department of Electrical-Electronics Engineering in Samsun, Turkey. Her research interests include communication and digital signal processing.

MOHAMMED SUFYAN MOHAMMED was born in in Iraq, in 1986. He received B.S and M.Sc. degrees in computer engineering from Mosul University, Iraq in 2010 and electrical and electronic engineering from Ondokuz Mayıs University (OMU), Samsun, Turkey, in 2020 respectively.