Digitalization of the Parking Lot at the Public University "Kadri Zeka" in Gjilan

https://doi.org/10.3991/ijes.v9i3.24219

Ragmi Mustafa[1], Basri Ahmedi[1], Kujtim Mustafa[2](✉)
[1]Universiteti Kadri Zeka Gjilan, Gjilan, Republic of Kosovo Serbia
[2]South East European University, Tetovo, North Macedonia
km26915@seeu.edu.mk

Abstract: The main purpose of this paper is to describe how the construction of the project was carried out, which controls the entries and exits in the parking lot of vehicles "SMART PARKING" at the Public University "Kadri Zeka" in Gjilan, in the Republic of Kosovo. In fact, this paper deals with the digitalization of vehicle parking lots. This project uses hardware and software components for its realization. Hardware components operate according to software instructions executed according to a working program code within a closed and endless programming cycle. This automatic form of operation of a digital car park not only regulates the traffic inside the car park but also satisfies the drivers of the vehicles with information given in a display or monitor. The information refers to vacancies in the parking lot and informs the driver of the vehicle about whether or not he can park his vehicle by allowing or not allowing the opening of the barrier. On the other hand the system becomes even more sophisticated when at the entrance of the smart parking lot is placed the key for searching for the opening of the barrier which may not open if there are no free parking spaces. All of this is controlled by software written in the C ++ programming language, of course within the Arduino UNO programming environment. Therefore, the electronic-hardware devices used in this paper/project are directed by the relevant software placed in the memory of the microcontroller in the system "Smart Parking". The reason for building this automatic vehicle parking system is to control and prevent the attempted parking of vehicles that would seriously impede the exit of previously parked vehicles in the parking lot of the Public University "Kadri Zeka" in the city of Gjilan, Republic of Kosovo.

Keywords—smart parking, arduino uno, memory, microcontroller, sketch, code, software, button, barrier

1 Introduction

Computer programming is important because nowadays, work is increasingly being done on the automated process of systems Computers and computer-controlled machines are able to do things efficiently and accurately. Such a solution is also offered in the automation of the parking lot in the university complex of the University

“Kadri Zeka” in Gjilan. In this way the easiest and safest access of the vehicle driver to use a smart parking is achieved. Of course, at the entrance-exit of the smart parking lot there is a barrier which opens or closes depending on the press of the respective button but at the same time controlled by the respective software loaded on the microcontroller of this system. This makes the car park at the public University “Kadri Zeka” in Gjilan “smart”. Each driver of the vehicle before entering the parking lot on the display-monitor sees the number of vacancies in the parking lot, and if there are vacancies, the switch is pressed to open the barrier. If the number of parking spaces is less than the maximum, by pressing the switch “PUSH BUTTON” the barrier opens which continues being so within specific time in the program code and then closes. At the same time in the display monitor decreases by one, the number of parking spaces. This informs the next vehicle driver who is looking for vacancies in “Smart Parking”. If there are no more vacancies, the display-monitor will again show information that there are no vacancies for parking. If the driver of the vehicle tries to open the barrier by pressing the opening key, it does not open and also an apology text is displayed by begging the driver to wait until a vehicle comes out from “SMART PARKING”. The exit from the parking lot will be realized in the same way, by pressing the switch on the inside of “SMART PARKING” for the opening of the barrier which in each case is opened and kept so for the time specified in the program-software code. The number of free seats when leaving the parking lot of a vehicle increases by one and so on. On the other hand, when the barrier is closed from the entrance to the car park, it is signaled with a red LED, and when it is open, it is signaled with a green LED. The solution of this problem for this “SMAR PARKING” is realized by using the following hardware equipment:

- An Arduino Uno board with ATmega328P microcontroller
- A USB cable for connecting the computer to the board for downloading software instructions,
- A prototype plate for electrical connection “Breadboard”,
- A 7-segment display with 16X2 pixels respectively 16 columns and two rows,
- Two resistors with suitable resistance of 230 [Ω],
- Two switches “PUSH BUTTON”,
- An electromotor for moving Barrier up and down,
- Micro-breakers,
- Barrier and other auxiliary mechanical materials
- Sufficient number of copper electrical conductors for connecting hardware components

The software part will contain the special program instructions for loading in the memory of the microcontroller of the Arduino Uno board.

2 Research methodology

Public University “Kadri Zeka” was established in the academic year 2014/15 with headquarters in Gjilan, Republic of Kosovo according to the decision of the Assembly of Kosovo [22]. For many decades on the campus where the University is located today has operated the Higher Pedagogical School for the preparation of educators for preschool children and primary school teachers since 1958 [21]. In the beginning, the University had four faculties of education, economics, law and computer science with a small teaching staff of exactly 52 of them and no more than 8 administration officials, for fulfilling University’s needs, which obviously was insufficient for respective faculties. On the other hand, knowing that all teachers and University officials travel by car a small parking area of 600 [m] is reserved, where no more than 50 cars can be parked, even though in the beginning it was sufficient. Initially, the management of the car park was by the employees of the faculty administration.

In the following years, the University “Kadri Zeka” begins to grow with the opening of two programs at the master level in the academic year 2017/2018 [18] [19] and a new program at the bachelor level in the academic year 2018 / 2019 [20]. In 2017/2018 period, the competition for a car park begins to increase, taking into account the total number of employees of the University “Kadri Zeka” in Gjilan. In Table 1 we will present the increase of University staff and the increasing rate of parking needs given that the number of parking spaces has not increased at all and it remains constant at 50 parking spaces.

Table 1. Increase in the number of employees at the University “Kadri Zeka” and the number of vehicles looking for a parking space in the period 2014 – 2021

Academic Year	Teaching Workers	Administration Workers	Total Workers	Competitors for a Car Park
2014/15	52	8	60	1.20
2017/18	61	12	73	1.46
2019/20	82	18	100	1.98
2020/21	104	30	134	2.68

From the processing of tabular data through the Microsoft Excel application [17] and the graphical presentation as in Figure 1, it is clear that in 2018 the need has increased to 1.46 requests for a parking space, while in 2020 the demand has passed to 1.98 which means that approximately 2 cars compete for one parking space. This increase of 2021 is explained by the fact that now the University is filled with administrative officials in the faculties but also in the university according to the respective needs. On the other hand, the number of teaching staff has increased because for the last five years the need for teachers has increased due to the increase of faculties number and study programs [21].

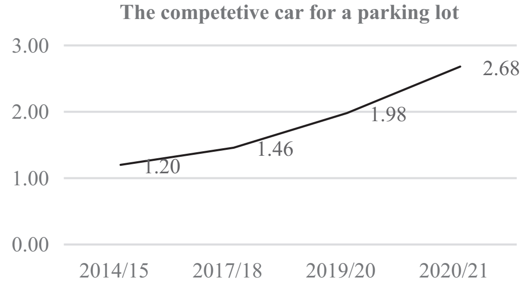

Fig. 1. Parking lot requirement based on academic year

A special feature is in June 2020 when from the quarterly isolation due to protection from the Covid 19, distance learning and the release of free movement from June 1 arises the need to discuss parking lots in the courtyard of the University “Kadri Zeka” in Gjilan. From the 1st to the 15th of June the lectures and exercises continued physically in the University in the classrooms, and the administrators began to perform services in person to the students and teachers every day, therefore this presented the problem of parking lots. Figure 2 shows the car park when the cars of the University workers enter and leave in a regular and without barriers.

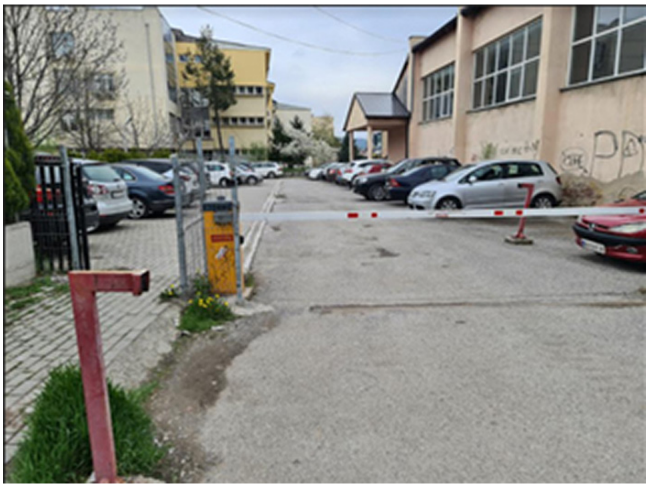

Fig. 2. Parking of vehicles without barriers entry-exit

It can be seen in Figure 3 the irregular parking of a car, let’s now even mention when more such are parked in the area between two parallel rows of car park.

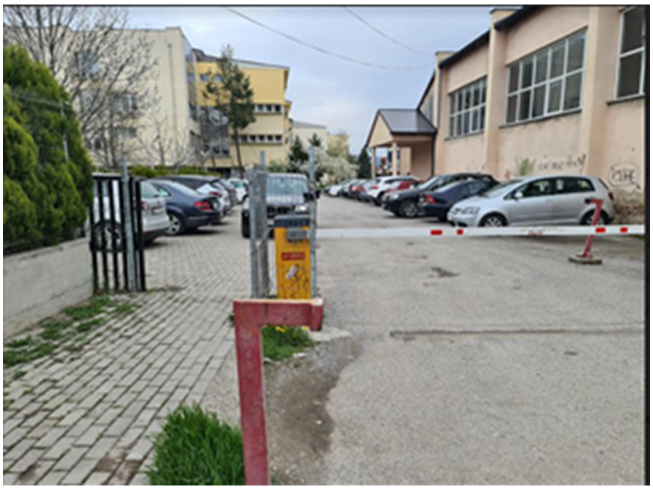

Fig. 3. Parking of vehicles with entry-exit barriers

In the case of parking as in Figure 3 serious problems are created because anyone who was careless in parking in the middle of the free zone, seriously obstructed the movement of cars for entry and exit. Meanwhile, this form of illegal parking takes time to find out whose car it is, by opening the doors of the classrooms, asking about whether is yours or his, wandering from hall to hall, office to office and thus serious obstacles were created during the winter semester from October to December 2020. One such problem occurred every day by two or more university workers. Therefore, it irritated not only the management staff of the University but also the teachers and especially the students when the teacher left the class and returned after half an hour or more, perhaps in many cases the teacher could not return at all. This irritating problem was discussed through the reactions of the student parliament of the University “Kadri Zeka” and the representative of the student parliament from the Faculty of Computer Science who asked the teachers of the faculty to provide a solution to this problem by proposing to find an electronic form in combination with construction. of a relevant software. Since computer science professors from the academic year 2014/15 together with students we have worked on many concrete projects such as Moodle for distance learning, e-learning [24], digitalization of classrooms [26], ukz-platforms [25] and other similar projects that have not been published on the Internet, we have found a way to solve this problem by digitizing the parking lot of the University “Kadri Zeka” in Gjilan. Therefore, two teachers in collaboration with an external staff, students and the support of the faculty council of the Faculty of Computer Science and the University management solved the problem easily by integrating in one system the necessary hardware, machinery and most importantly the relevant software.

3 Description of hardware equipment

3.1 Hardware arduino

Arduino is an Open-Source electronic platform based on easy-to-use hardware and software. Arduino Unos boards are able to read inputs - changes to a sensor, a finger pressed on a button, or a message in the app - and activate on an output - by activating a motor, turning on an LED, connecting contacts of a relay contact, contactor, activate an alarm or a similar ringtone. To achieve this, the C or C ++ programming language should be used within the Arduino development environment. All Arduino boards are fully open source, empowering users to build them independently and ultimately adapt them to their specific needs. [1]

The Arduino Uno board contains a microcontroller consisting of a memory microprocessor which is able to be programmed in order to understand, control, communicate with objects in the physical world. Due to its flexibility and low cost, the Arduino UNO has become a very popular choice for manufacturers and the choice of creators to create interactive hardware projects.

3.2 Types of arduino boards

Below are examples of different types of Arduino boards. In the boards named Arduino are official boards but there are also many clones in the market as in Figure 4. One of the best reasons to buy a clone is the fact that they are usually less costly than their official counterpart. For example, Adafruit and Sparkfun sell variations of Arduino boards that cost less, but still have the same quality as the originals.

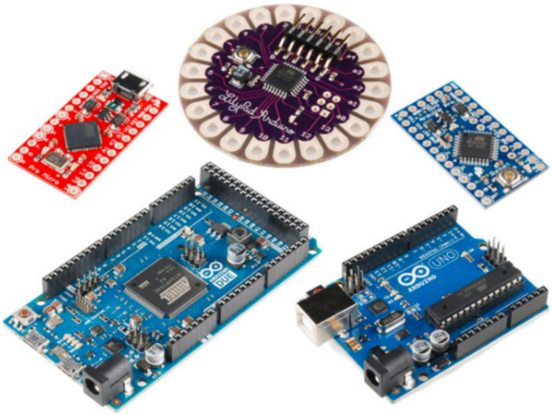

Fig. 4. Types of Arduino boards [2]

One of the most popular Arduino boards is the Arduino Uno. While it was not actually the first board to be launched, it remains to be the most actively used and documented on the market. Figure 5 shows Arduino Uno, which will be used for the realization of the project "Smart Parking".

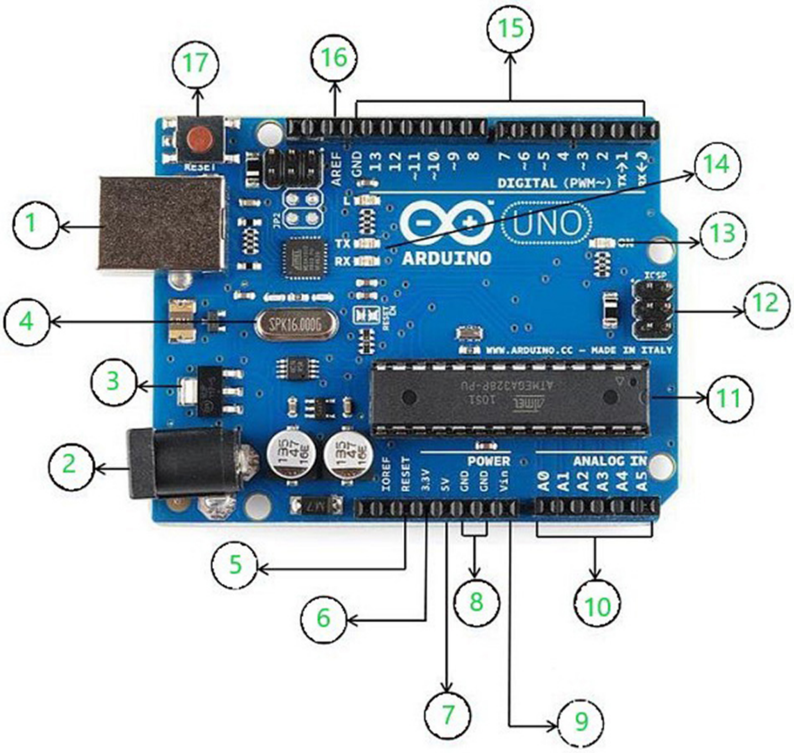

Fig. 5. Arduino plates Uno [2]

The composition and functions of the Arduino Uno parts are as follows:

- USB: can be used for both power supply and communication between computer and Arduino Uno board (IDE).
- Barrel Jack: used for power supply from a 9 [VDC] external battery
- Voltage regulator: regulates and stabilizes input and output voltages.
- Crystal Oscillator: keeps track of time and regulates processor frequency
- Reset Pin: can be used to reset Arduino Uno
- Pin 3.3V: can be used as 3.3V output [VDC]
- 5V pin: can be used as a 5V output [VDC]
- GND pin: can be used for electrical circuit earthing
- Vin pin: can be used to power the board
- Analog pin (A0-A5): can be used to read analog signals on (table) Arduino Uno inputs such as sensors, push buttons, microswitches or similar
- Microcontroller (ATMega328): the processing and logic unit of the board
- ICSP pin: a table programming head also called SPI
- Power indicator LED: indicates the power supply status of the Arduino Uno Board-Board (s)
- RX and TX LEDs: receive LED (RX) and transmit (TX) LEDs, flash when sending or receiving, serial data.
- Digital I / O pin: 14 pins capable of reading and outputting digital (xh) signals; 6 of these pins are also capable of PWM (Pulse Width Modulation)
- AREF Pins: can be used to set an external reference voltage as the upper limit for analog pins
- Reset button: can be used to reset the Arduino Uno board.

3.3 Arduino power supply

The Arduino Uno needs a power source in order to function and can be powered in a variety of ways. Connects the board directly to the computer as in Figure 6 we can even use a 9 [VDC] battery for supply [2].

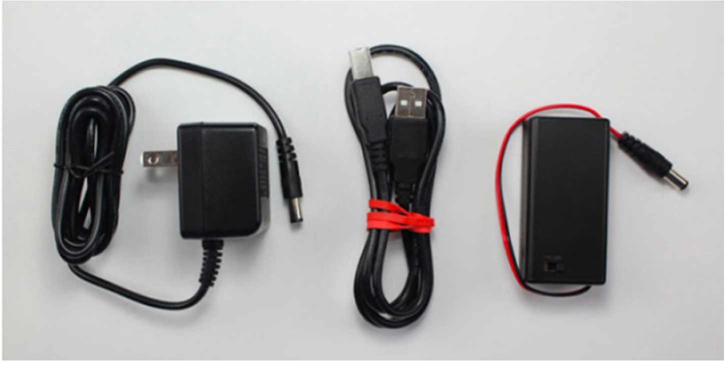

Fig. 6. USB Cables for supply of Arduino Uno [2]

3.4 Memory arduino uno

There are three sets of memory in the microcontroller used on AVR[1]-based Arduino UNO boards:

- Flash memory (program space), is where the Arduino sketch is stored.
- SRAM (static random-access memory) is where the sketch creates and uses variables when executed.
- EEPROM is the memory space that programmers can use to store long-term information. Flash memory (programming flash space), it is a place where is stored Arduino sketch.

Flash memory and EEPROM memory are not volatile (information continues to be stored after a power failure). SRAM is unstable and will be lost when the power supply is cut off.

The ATmega328P chip microcontroller on the Arduino Uno has the following amounts of memory: Flash 32k bytes (of which 5k is used for bootloader), SRAM 2k bytes, EEPROM 1k bytes [3].

3.5 Prototype breadboard of arduino uno

Another very important device when working with Arduino is breadboard. This device allows us to realize our Arduino project without having to permanently merge the circuit together. Using a breadboard allows us to create temporary prototypes and experiment with different circuit designs. Inside the holes (connection points), there are metal clamps which are connected to each other with strips of conductive material, which serve to conduct electricity.

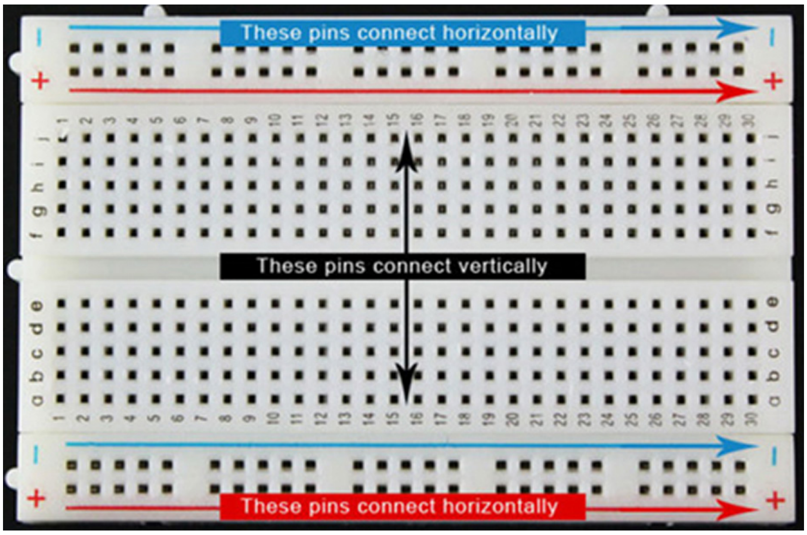
These pins connect horizontally
These pins connect vertically
These pins connect horizontally
+
-
a b c d e
f g h i j

Fig. 7. Arduino Breadboard [2]

[1] AVR is a microcontroller of the ATMEL family, used in the Arduino.

From Figure 7 we notice that the 2 rows with holes at the beginning and end are connected horizontally. And the other pins are connected in vertical shapes. Breadboard needs power supply provided by Arduino board using copper conductors.

3.6 "Jumper" wires

"Jumper" wires as shown in Figure 8 are simply wires that have connecting pins at each end, allowing them to be used to connect two points to each other without joining. Jumper wires are commonly used with breadboards to make it easy to change a circuit as needed. [4]

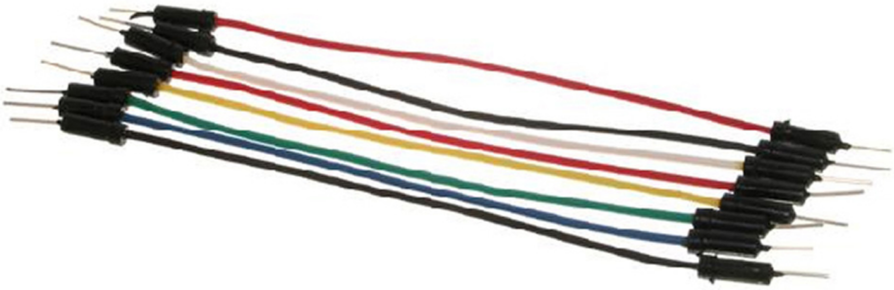

Fig. 8. "Jumper" wires [4]

3.7 Display LCD (Liquid Crystal Display) 16X2

Nowadays, we always use devices which are composed of LCD such as CD player, DVD player, digital clock, computers, and other technology. These devices are more subtle, and power consumption is extremely lower. A type of electronic display module used in a wide range of applications such as circuits and various devices such as cell phones, calculators, computers, televisions, etc. These screens are mainly preferred for light-emitting multi-segment diodes and in our case seven-segment LED diodes per pixel as it can be seen in Figure 9. The main benefits of using this module are cheap; simply programmable, animations and no restrictions on display of personalized characters, special animations [5].

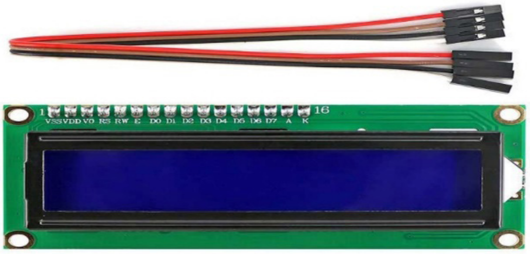

Fig. 9. LCD display [6]

Features of this LCD mainly include:

- Supply voltage 4.7[VDC]-5.3[VDC]
- It includes two lines where each one can produce 16 characters.
- Current consumption is 1 [mA] without light illumination
- Each character can be constructed with a matrix [5 × 8] pixel
- Alphanumeric alphabets and LCD numbers
- The screen is able to work in two modes like 4-bit and 8-bit
- These can be obtained in Blue and Green Lighting

In Figure 10 are presented the component parts of LCD 16X2

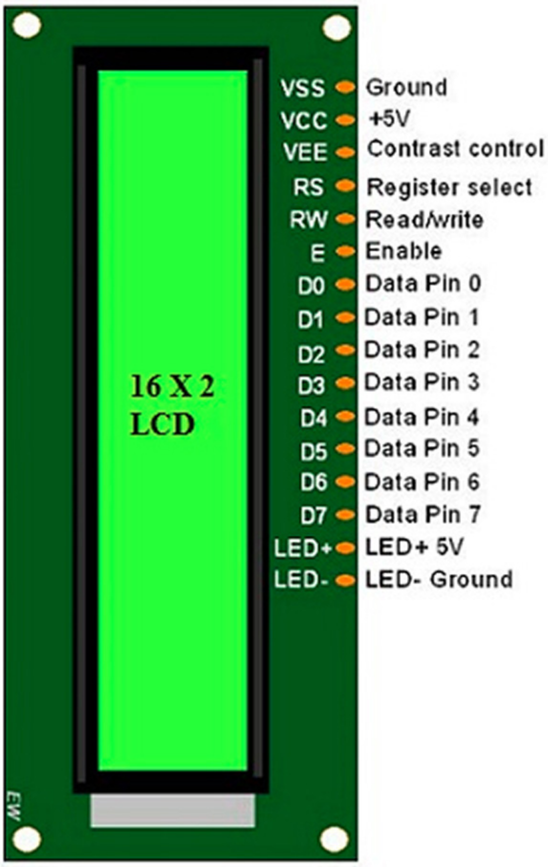

Fig. 10. Component parts of LCD 16X2 [6]

Pin1 (Ground / Pin Source): This is a GND display pin, used to connect the common GND terminal of the microcontroller unit or power source.

Pin2 (KQV / Pin for energy source. This is the display voltage supply pin, which is used to connect the power supply pin of the power supply.

Pin3 (V0 / VEE / Controller): This pin adjusts the display change, used to connect a variable POT (potentiometer) that can supply voltage from 0 to 5 [VDC]. This pin switches between the command or data register, which is used to connect a pin to the microcontroller unit, and gains either 0 or 1 (0 = data mode, and 1 = command mode). Pin4 (Registry Selection): This pin switches between the command or data register, which is used to connect a pin to the microcontroller unit and gains either 0 or 1 (0 = data mode, and 1 = command mode). This pin must be held up to execute the Read / Write process, and is connected to the microcontroller unit and held up continuously. Pins 7-14 (data pins). These pins are used to send data to the screen. These pins are connected in different ways 2, 4 and 8 wires. In the 4-wire mode, only four pins are connected to the microcontroller unit as 0 to 3, while in the 8-wire mode, the 8-pins are connected to the microcontroller unit as 0 to 7.

Pin15 (+ pin for LED): This pin is connected with + 5 [VDC] Pin 16 (- pin for): This pin is connected with GND [5].

3.8 "PUSH BUTTON" switch

A button is an electrical device that connects two points in a circuit when pressed. In this case, an LED lights up when the button is pressed. We connect three wires to the Arduino board. The first runs from one foot of the button through a 2.2 [kΩ] resistor to the 5 [VDC] supply. The second goes from the corresponding foot of the button to grounding. The third connects to a digital I / O pin (here pin 7) which reads the status of the button.

When the button is open there is no connection between the two pins of the button, so the pin is connected to 5 [VDC] and we read UP. When the button is closed, it makes a connection between his two pins, connecting the pin to the earthing system, so that we can read DOWN. (The pin is still connected to 5 volts, but the resistance between them means the pin is "closer" to the earthing system.) [6] Figure 11 shows how "PUSH BUTTON" is interrupted and continues another prototype for "Breadboard".

```
int ledPin = 13;  // pin selection for LED light
int inPin = 7;    // selecting pin for output/ exit
int val = 0;      // variables for reading pin status
void setup() {
  pinMode(ledPin, OUTPUT); // LED declaration as exit/
output
  pinMode(inPin, INPUT); // declaration of the button as
input
}
void loop(){
  val = digitalRead(inPin); // reading the input value
  if (val == HIGH) { // we see if the value is high 1)
  digitalWrite(ledPin, LOW); // Turning off the LED
light
   }  else {
  digitalWrite(ledPin, HIGH); // LED Up/above(1)
   }}
```

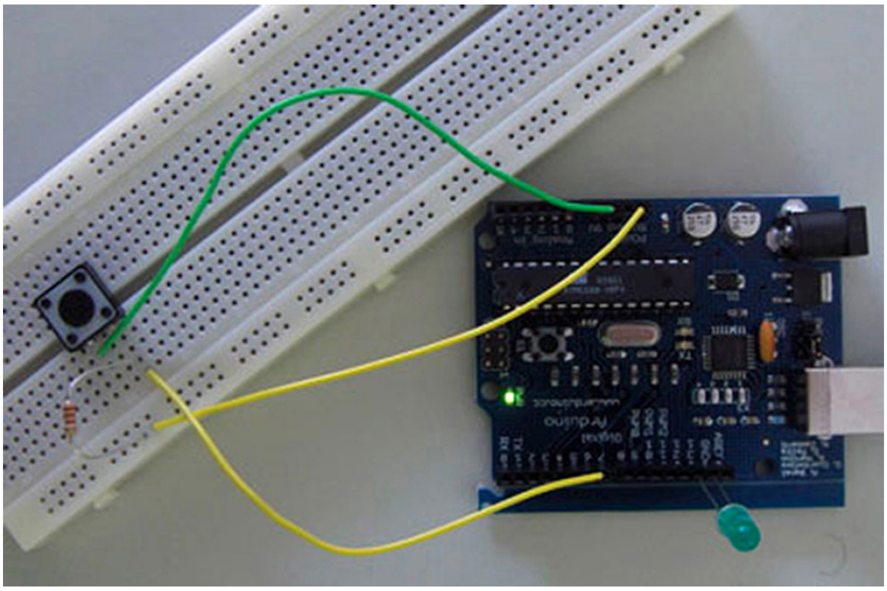

Fig. 11. "PUSH BUTTON" Switch

4 Programming code written in the C ++ programming language in the arduino uno development environment

In order to demonstrate the operation of this PUSH BUTTON which connects a LED for lightening, on the Arduino UNO board as in Figure 11, also is needed the program code written in the C ++ programming language under the Arduino Uno development environment which runs on the Atmega 328P microcontroller Arduino Uno.

4.1 Resistors

A resistor is a passive electrical component with two terminals that applies electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, regulate signal levels, split voltages, and terminate transmission lines, among other uses. The unit of resistance is called Ohm, which is usually written in the Greek letter [Ω] Omega. Since an Ohm is a low value of resistance (it does not resist much at all), resistance values in [kΩ] kiloohm (1000 [Ω]) and [MΩ] megaohm (1 000 000 [Ω]) are usually taken. Figure 12 shows the appearance of resistors.

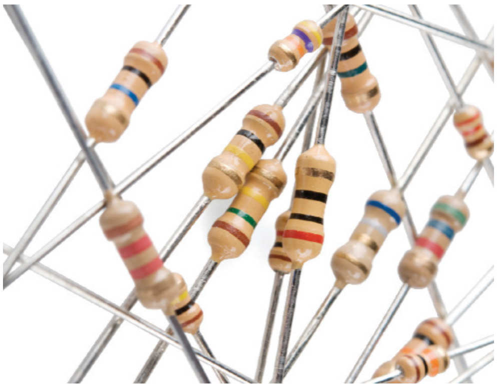

Fig. 12. Resistors [7]

4.2 Led diode

A light emitting semiconductor diode (LED) is a light source that emits light when current passes through it. Electrons in semiconductors recombine with electron holes, releasing energy in the form of photons. The color of light (corresponding to the energy of photons) is determined by the energy required for electrons to pass to the semiconductor band. White light is obtained using multiple semiconductors or a phosphor layer that emits light into the semiconductor device.

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity (IR) infrared light. Infrared LEDs are used in remote control circuits. The first LEDs with visible light were low intensity and limited to red. Modern LEDs are available along the visible wavelength, ultraviolet (UV) and infrared, with high light output. LEDs have many advantages over incandescent light sources, including lower power consumption, longer life, improved physical durability, smaller size, and faster switching.

LEDs are used in various applications such as aviation lighting, automotive lights, advertising, general lighting, traffic signals, camera lights, backlit lights, garden lights and medical devices, especially in industrial production machines. In this paper they are used to signal the opening and closing of the barrier. When the barrier is closed the red LED (it is signaled) illuminates while when the barrier is closed (it is signaled) then the green LED lights up. Figure 13 shows the appearance of semiconductor LED. [7]

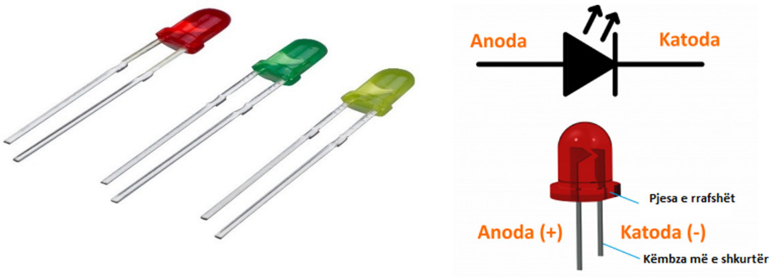

Fig. 13. LED diodes

When using an LED, it is always preferable to use a current-limiting resistor to protect the LED from the piercing voltage that will destroy-burn the Led Diode. If we connect the LED directly to 5 [VDC] without a resistor connected in series with the LED, the LED will be overloaded, will be too bright for a while, and then burn out. For this reason, we use resistors depending on how bright we want a led diode to be, in addition we will reduce the current which passes to the LED diode and not to exceed the maximum allowable current. Since both the resistor and the LEDs are connected to the output pins of the Arduino Uno board, at the same time with the protection of the LED diode, the Arduino Uno board is also protected.

Figure 14 shows the electrical circuit for the protection of an LED which is connected in series with a resistor. Since in this type of LED diode the current is limited to a maximum of 20 [mA] at a supply voltage of 5 [VDC] then we will calculate the resistance of the resistor that will be connected to the circuit with the LED diode in order to protect the LED diode from its destruction even when used in the ARDUINO UNO plate, it will be protected itself because currents are not allowed in an output pin more than 20 [mA]. [16]

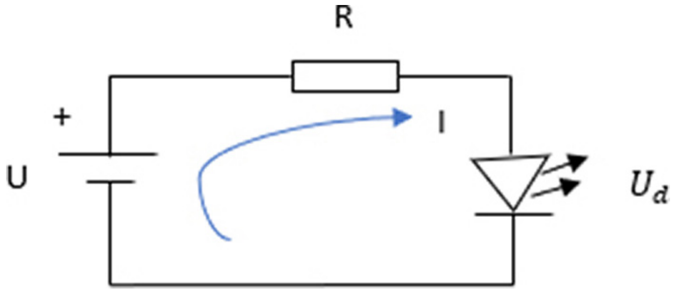

Fig. 14. Connecting the LED in series with the resistor for its protection

Starting from Kirhoff’s second law and the electrical circuit as in Figure 14, then the supply voltage U = 5 [VDC], the current I = 20 [mA] and the LED voltage U_d = 4.4 [VDC], we calculate resistance R as follows:

$$U - R \cdot I - U_d = 0 \tag{1}$$

$$R \cdot I = U - U_d \tag{2}$$

$$R = \frac{U - U_d}{I} = \frac{5[VDC] - 0.6[VDC]}{20[mA]} = \frac{4.4[VDC]}{20 \cdot 10^{-3}[A]} = 0.22 \cdot 10^3[\Omega] = 220[\Omega] \tag{3}$$

5 Arduino (IDE) software

Arduino Software (IDE) - features a text editor for writing code, a message area, a text keyboard, a toolbar with buttons for common functions, and a set of menus. Connects to the Arduino hardware to download programs and communicate with them. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and saved with the .ino file extension. The editor has features for cutting / pasting and searching / replacing text. The messaging area provides feedback during storage and export, additionally, it also displays errors. The console shows the output of text from the Arduino Software (IDE), including full error messages and other information. The lower right corner of the window shows the configured board and serial port. The toolbar buttons enable to verify and load programs, create, open and save sketches, and open the serial monitor.

Additional commands are found within five menus: File, Edit, Sketch, Tools, Help. The menus are sensitive toward the context, which means that only those items related to the work currently being performed, are available.

Before loading the sketch (program code), we need to select the correct items from the Tools> Board and Tools> Port menu. The boards are described below. In Windows, it is COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to detect, you are looking for USB serial devices in the ports of the Windows Device Manager. Once we have selected the correct port and serial board, press the load button on the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will be reset automatically and start loading. The Arduino software (IDE) will display a message when the upload is complete, or indicate an error. Libraries offer additional functionality for use in sketches, e.g., working with hardware or by manipulating data to use a library in a sketch, select it from the Sketch menu> Import Library. This will insert one or more statements at the top of the sketch and compile the library with your sketch. Since libraries are loaded onto the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code. There is a list of reference libraries. Some libraries are included with Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you can import a library from a zip file and use it in an open sketch. [8]

6 Program code in C ++ for digitalization of vehicle parking in the university complex of the university "Kadri Zeka" in Gjilan "Smart Parking"

Figure 15 provides a complete electrical scheme of necessary connections for the operation of "Smart Parking" through which is digitized the parking of vehicles of employees such as management, teachers and administrative officials and even official visitors from other institutions of the state of Kosovo in the public University. "Kadri Zeka" in Gjilan. We also wrote the pseudo code for this working smart parking and you can find in the appendix 1.

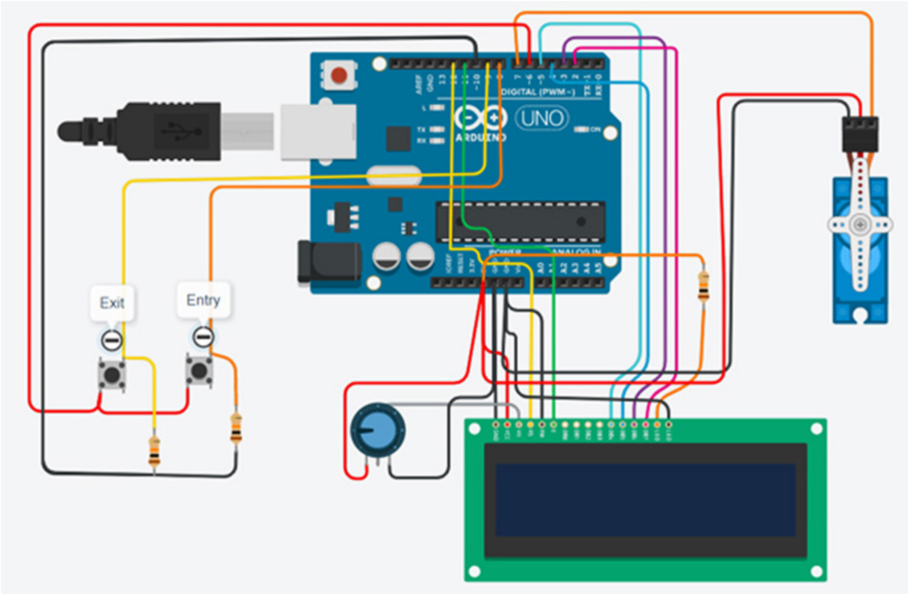

Fig. 15. Electrical scheme of digital parking at the Public University “Kadri Zeka” in Gjilan

7 Conclusions

The systems in the Republic of Kosovo are increasingly being digitalized more and more, although the economic and social transition is still in a long period. This, because it emerged from the terrible genocidal and destructive war of Albanian property and wealth by Serbia in the war of 1997 to 1999. Universities are now performing the role of state-building in Kosovo. Kosovo’s academic capacities or brains are largely being used for resource innovations leading to digitalization, in this case at the Public University “Kadri Zeka” in Gjilan. The digitalization of various systems is coming as a need of the time in state institutions, public administration, private businesses, shopping malls and even Universities. With the digitalization of various systems in public and private institutions, the elimination of bureaucracy, corruption, waiting (wasting time), daily stress and above all nepotism is being achieved.

The management of the digital car parking system at the University of Gjilan is really the first system of its kind digitized in the Universities and University Colleges in the Republic of Kosovo, perhaps even compared to other public and private institutions. The idea of this digital system arises from the creation of problems during improper parking by the negligence of teachers, administrators, foreign visitors, even senior officials from the Ministry of Education or government bodies. Furthermore, intentionally or not they seriously impede and block the exit of vehicles that have previously been parked. Therefore, with this system of digital parking in the public University “Kadri Zeka” of Gjilan is achieved:

1. Opening the barrier for entrance to the parking lot only when there are vacancies.
2. Counting exactly each day the number of cars entering and leaving the parking lot
3. That none of the University's management staff give the order to open the barrier by abusing official duty either for visitor or nepotism effect
4. That no careless car driver stresses one or more of his colleagues for reasons of parking the vehicle even in cases when there are no vacancies
5. The system will be self-managed without the presence of the human factor, in which case two jobs are eliminated for the opening and closing of the barrier in two shifts because the University traditionally works from 7:00 in the morning until 20:00 in the evening.
6. To finally eliminate the possibility of abusing the official duty of the human factor which since the beginning of 2015 to 2020 has been managed only in one shift by the human factor, not with extra care and not to mention a second shift when chaotic situations were created
7. To warn car drivers that in cases when there are no vacancies to seek a free place within the city in other parking lots even if for a fee
8. Time to be valued equally comparing how it is important today in the countries of the European Union and even the most developed countries of the world
9. Knowledge to be implemented in practice, both the united teachers and the students of the Faculty of Computer Science
10. To continue with similar innovative projects at the University "Kadri Zeka" in Gjilan and beyond, in the Republic of Kosovo

Within the period of time since the establishment of the Public University "Kadri Zeka" and also the Faculty of Computer Science within this University, other digital systems have been implemented and are still in operation today. These systems are as follows:

1. University Management System - SMU [27], which enables tracking, grading, graduation, reporting, online teaching of students by teachers as well as many student actions on this platform led mainly in digital form and remotely
2. Online learning system and placement of assignments, projects, comments, lectures, diploma topics and similar, on the e-learning platform [24] for both teachers and University students
3. Digital platform for following students in online learning, their presence in the classroom, cabinets, laboratories [25]. From this platform various reports can be downloaded and printed for the needs of teachers, management leaders of faculties, University management and even the service for budget and finance as well as that for quality assessment
4. Digital information system about what is happening in real time in a classroom, cabinet, laboratory or faculty space without opening the door of the space in question. This is achieved by placing the digital panel system with LEDs on entrance gate of classrooms or similar spaces [26]

All these digital systems are built by the very close and serious cooperation of the professors of the Faculty of Computer Science, assistants, external collaborators and students.

Therefore, the degree of cooperation in the Faculty of Computer Science will be increased even more in order to provide other digital systems either for the needs of the University or for other public and private institutions in the Republic of Kosovo.

8 References

[1] Team, T. A. (n.d.). What is Arduino? Retrieved from https://www.arduino.cc/en/Guide/Introduction

[2] Arduino Uno For Beginners - Projects, Programming and Parts (Tutorial). (2017, April 25). Retrieved from https://www.makerspaces.com/arduino-uno-tutorial-beginners/

[3] Team, T. A. (n.d.). Memory. Retrieved from https://www.arduino.cc/en/Tutorial/Foundations/Memory

[4] Hemmings, M. (n.d.). What is a Jumper Wire? Retrieved from http://blog.sparkfuneducation.com/what-is-jumper-wire#:~:text=Jumper wires are simply wires, change a circuit as needed

[5] N. (2019, August 26). LCD 16x2: Pin Configuration, Features and Its Working. Retrieved from https://www.elprocus.com/lcd-16x2-pin-configuration-and-its-working/#:~:text=LCD 16×2 Pin Diagram&text=Pin2 (VCC/Source Pin):,pin of the power source.&text=Pins 7-14 (Data Pins,mode and 8-wire mode.

[6] Last Minute Engineers. (2020, December 18). In-Depth: Interfacing an I2C LCD with Arduino. Retrieved from https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/

[7] Monk, S. (n.d.). Arduino Lesson 2. LEDs. Retrieved from https://learn.adafruit.com/adafruit-arduino-lesson-2-leds/resistors

[8] Environment. (n.d.). Retrieved from https://www.arduino.cc/en/guide/environment

[9] Md Ayan. (2020, September 13). *Automatic Car Parking Lot* [Video]. Md Ayan. https://www.youtube.com/watch?v=5xTh78LZGI0&ab_channel=MdAyan

[10] Tony Gaddis, Starting with Programming Logic & Design, 2007

[11] Tomas Floyd, Digital Fundamentals, Ninth Edition, 2007

[12] T. Gadis, L. Floyd, Introduction to Engineering Technology, 2007

[13] Frank D. Petruzella, Programmable Logic Controllers, 4th Edition, 2011

[14] Warwick Smith, C Programing with Arduino, 2018

[15] Bert Van Dam, Arduino Uno (45 projects for beginners and experts), 2018

[16] Prof. Ass. Dr. Ragmi Mustafa, Disa shembuj digjitalizimi në Arduino Uno, 2020

[17] John Walkenbach, Microsoft Excel 2000 Bible, 2000

[18] https://www.uni-gjilan.net/fakultetet/fakulteti-juridik-master-qeverisja-lokale-shoqeria-demokratike/rreth-fakultetit/

[19] https://www.uni-gjilan.net/fakultetet/fshk-fek-fju-master-e-qeversija/rreth-fakultetit/

[20] https://www.uni-gjilan.net/fakultetet/fakulteti-i-shkencave-aplikative/orari-i-provimeve/

[21] https://www.uni-gjilan.net/universiteti/historiku/

[22] https://www.uni-gjilan.net/wp-content/uploads/2020/02/Ratifikimi-i-vendimit-te-Themelimit-te-Universitetit-Kadri-Zeka-.pdf

[23] https://www.uni-gjilan.net/universiteti/stafi-akademik-2/

[24] http://e-learning.uni-gjilan.net

[25] http://www.ukz-platforma.net/

[26] https://publons.com/researcher/3866276/ragmi-mustafa/

[27] http://smu.uni-gjilan.net

9 Appendix 1: Pseudo Code

```
#include <Servo.h>       //Using library for Servo engine
#include <LiquidCrystal_I2C.h>       //Using library for
LCD engine
LiquidCrystal_I2C lcd(0x27,20,4);   // Putting address in
LCD
Servo myservo;  // Object creation named servo, for con-
trolling servo engine
#define ServoM 7   // defining constant for servo engine,
pin 7
#define Exit   9 // defining constant for exiting from
parking, pin 9
#define In     8 // defining constant for entering in par-
kin, pin 8
#define BarLow  90 // defining constant for monitoring the
moving angle of barrier 0-90˚
#define BarUp   177 // defining constant for monitoring the
moving angle of barrier
#define CAPACITY 3 // defining constant for free space at
parking
void setup() {
      myservo.attach(ServoM); // Initialization of Servo
      Engine
      lcd.init();// Initialize the LCD // Initialization
      of LCD display
      lcd.backlight();  // Light activation in LCD
      lcd.print("Space left for:"); // Printing text in
      LCD display
      pinMode(Exit, INPUT); // Configuration of pin 9
      (EXIT), declared as exit pin
      pinMode(In, INPUT); // Configuration of pin 8
      (EXIT), declared as entered pin

      myservo.write(BarLow); // Activation of servo engi-
      nee at 90 degree.
      digitalWrite(13,HIGH); // Activation of pin 13,
      with electricity.
      digitalWrite(12,LOW); // Activation of pin 12, for
      cutting electricity
}
     int Available=3; //Declaration of variable for stor-
     ing the free spaces in parking.
```

```
void loop(){   // Putting the program to execute endless,
on the loop.
      if(Available == 1) { // If there is a free parking
      space will execute the following code
      lcd.clear();  // Lcd display will clear the
      characters
      lcd.setCursor(1,0);  // Putting the cursor at row 1
      and column 0
      lcd.print(“Space left”); // Printing text at LCD
      display
      lcd.setCursor(0,1); // Putting the cursor at row 0
      and column 1
      lcd.print(Available);  // Printing text “Available”
      lcd.print(“ car”); // Printing text “car”
      }else{     //Else condition if the above condition
      is not executed
      if(Available >= 1) {     // If there is more than
      one free parking space
        lcd.clear();    // Lcd display will delete
        characters
        lcd.setCursor(1,0);    // Putting cursor at row 1
        and column 0
        lcd.print(“Space left  “);     // Printing of
        text “Space left”
        lcd.setCursor(0,1);  // Putting cursor at row 0
        and column 1
        lcd.print(Available);      // Printing of text
        “Available”
        lcd.print(“ cars”);                         //
        Printing of text “cars”
      }else {                                          //
      Else condition if two above conditions are not
      executed
        lcd.clear();                                    //
        Lcd display will delete characters
        lcd.setCursor(1,0);                 // Putting
        the cursor at row 1 and column 0
      lcd.print(“Sorry!”);    );         // Printing of
      text “Sorry”
      lcd.setCursor(0,1);              // Putting the
      cursor at row 0 and column 1
        lcd.print(“No place “);       // Printing of text
        “Sorry”
```

```
}
}
if(digitalRead(In) == 1)     // Reading of pin for
entering when the button is pressed
{
if(Available !=0) {    // Condition if there are
free parking space to decrease the value
Available--;
   digitalWrite(13,LOW);   // To pin 13, will not
   send the electricit
   digitalWrite(12,HIGH);    // To pin 12, we send
   electricity
   myservo.write(BarUp);      // Opening of servo
   engine
   delay(3000);                   // Delay of pro-
   gram for 3 seconds
   myservo.write(BarLow);    // Closing the servo
   engine
   digitalWrite(13,HIGH);       //  To pin 13, we
   open electricity
   digitalWrite(12,LOW);    // To pin 12 we cut
   electricity
}
   else {
     myservo.write(BarLow);   //  Closing the servo
     engine

     digitalWrite(13,HIGH);     // To pin 13, we
     open electricity


  }
  }


if(digitalRead(Exit)==1)        // Reading the pin
for exiting when the button is pressed
{
  if( Available != CAPACITY) {    // If there is
  any capacity of free parking spaces
  Available++;                        // Free park-
  ing spaces will increase with one
  digitalWrite(13,LOW);         //  To pin 13, we
  cut electricity
```

```
          digitalWrite(12,HIGH);       // To pin 12, we
          open electricity
           myservo.write(BarUp);       // Opening of servo
           engine
           delay(3000);                            // Paus-
           ing the program for 3 seconds
           myservo.write(BarLow);              // Closing
           the servo engine
           digitalWrite(13,HIGH);        //  To pin 13, we
           open electricity
           digitalWrite(12,LOW);            ////  To pin
           12, we cut electricity
          }
        }
        delay(200);    //  The program’s pause for 2
        seconds
}
```

10 Authors

Prof. Ass. Dr. Ragmi Mustafa, is a professor in Public University “Kadri Zeka” Gjilan, Republic of Kosovo. Has experience more than 25 years in teaching and managing projects. E-mail: ragmi.mustafa@uni-gjilan.net

Prof. Ass. Dr. Basri Ahmedi, is a professor in Public University “Kadri Zeka” Gjilan, Republic of Kosovo. Has experience more than 20 years in teaching. E-mail: basri.ahmedi@uni-gjilan.net

M.Sc. Kujtim Mustafa, PhD Cand., computer engineer, PhD Candidate at South East European University in Republic of Macedonia. Actually, working as software engineer and is specialized in Web API configurations. Has more than 10 years in developing and working as software engineer in Republic of Kosovo.

Article submitted 2021-06-04. Resubmitted 2021-07-22. Final acceptance 2021-07-29. Final version published as submitted by the authors.