Műszaki Tudományos Közlemények vol. 9. (2018) 251–254.
https://doi.org/10.33894/mtk-2018.09.58
Hungarian: https://doi.org/10.33895/mtk-2018.09.58,
http://eda.eme.ro/handle/10598/30343

ARDUINO AND RAPSBERRY PI BASED HYBRID RFID ACCESS CONTROL SYSTEM WITH MOBILE PLATFORM

Sándor USZKAI[1], Beatrix PAPP[2], Ádám NEMES[3], Timotei István ERDEI[4]

[1] *University of Debrecen, Faculty of Informatics, Debrecen, Hungary, uszkaisanyi@gmail.com*

[2] *School of Law and Social Sciences Criminology with Psychology United Kingdom, London, pappb@lsbu.ac.uk*

[3, 4] *University of Debrecen, Faculty of Engineering, Department of Mechatronics, Debrecen, Hungary*

[3] *nemes.adam96@gmail.com*

[4] *timoteierdei@gmail.com*

Abstract

Nowadays the ongoing trend of integrating everything into the cloud, connecting all sorts of gadgetry to the Internet (IoT) is omnipresent at this point, even in the industry, where small and medium sized companies alike can join this tendency. This is undoubtedly the result of the arrival of Industry 4.0, which caused an exponential growth in this field. This poses a new challenge for the IT field, since it needs in-depth knowledge of networking and routing, and formidable knowledge of the different devices and programming languages alike. Our project premises an RFID-based fully-fledged access control system, with a targeted market of small to large companies, and also individuals, as the result of ease of use, low price, and simplicity.

Keywords: *Android, Arduino, RFID, database, IoT, Industry 4.0.*

1. Introduction

The era of being afraid of the Internet is over, and anybody who deems the technology dangerous is now considered technophobic. The kindergarteners holding tablets as books, and in 1st and 2nd world countries it's becoming more and more common to educate students about the modern wonders of IT technologies. The term Internet of Things (IoT) involves basically any gadgetry, such as a fridge, a toaster or thermostat equipped with the required connection ports to be connected to the Internet, or into any other networks. These "things" obviously require careful security measures, and care, but can make our lives easier, and automate tedious tasks. During our work, we used tested and working solutions as the foundation for our IoT setup, involving the implementation of an RFID-based access control system, for the Electrical Engineering & Mechatronical Faculty of the University of Debrecen. Our system could potentially be utilized by any large or small corporation, or even the private sector, with some minimal changes to the basic setup.

2. Design points

There are several research laboratories in the B building of the University of Debrecen, that are equipped with high-value machines and robots. It is therefore clear that access to the rooms must be strictly regulated. In the era of the Internet, it is almost unimaginable to use the traditional key as a security solution, so RFID technology was chosen. As the system contains a number of hardware and software elements, the most important question of design was whether these tools and programming languages could be used together, and if so, how? In order to overcome compatibility issues, we chose physical devices that have public parameters. In the software components, platform-independent programming languages were preferred.

In order to operate the system safely, the logging database runs on a Linux based operating system, as these systems are equipped with enhanced security components and successful attacks are much less frequent against them.

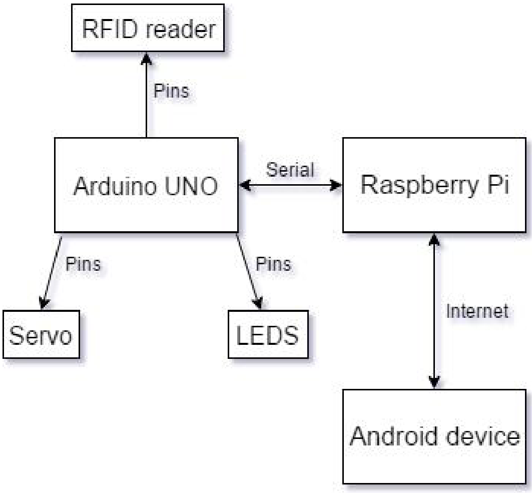


Figure 1. *System elements and interfaces*

During the development of the mobile app, Android was chosen, as it is the most widely used operating system today.

3. Physical structure of the system

The servomotor is an important physical element of the access control system. This performs the mechanical opening and closing of the door. An Adafruit TowerPro SG-5010 [2] servo was used. Since the 39g device has 5.5kg-cm torque, which is relatively high, it is suitable for moving the locks. Another criterion for selection was that the device should work on a relatively low voltage. Since the manufacturer has set 5V as optimal voltage, it is compatible with the Arduino UNO board.

UNO is an entry-level product of Arduino, ideal for novice developers, but also good for controlling more complex systems. The device is controlled by an ATmega328P 16 MHz microcontroller. It has 14 digital pins, 6 of which are capable of pulse width modulation (PWM). An additional 6 analogue pins are available for communication with analogue devices, in the test phase the potentiometer was connected to one of these pins to determine the required rotation angle of the servomotor. Further inputs and outputs of the panel are responsible for power supply and grounding.

The advantage of RFID technology is that those who are entitled to enter can conveniently identify themselves with a touch, in addition they can be operated with cards or so-called tags. Since each card or tag has a unique identifier, access is only available for cardholders in the database.

Table 1. *Points of Connection*

RFID–RC522	Arduino UNO
SDA	Digital Pin 10
SCK	Digital Pin 13
MOSI	Digital Pin 11
MISO	Digital Pin 12
RST	Digital Pin 3
3.3V	3.3V
GND	GND

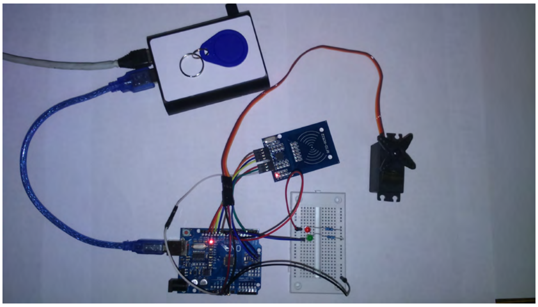


Figure 2. *The assembled physical system*

4. Arduino Sketch

When creating the sketch, primary goals were to create an easy-to-read code and to maximize speed. To accelerate coding, a pre-written RFID.h header was added to the source code, which contains the functions required for the RFID module.

The identifier of cards consists of 5 groups of digits, so it is necessary to create a five-element array for storing it. For the sake of transparency, special purpose functions and procedures have been declared separately in the source code, and they are implemented at the end of the source code.

To allow the reader to identify access cards, it must be checked whether the card ID is within the specified range. In a false case, the cycle starts again, if it’s true the RFID device reads the access card ID and sends it to the serial port where the Python script running on Raspberry Pi checks whether it is in the database. In order for the query to be executed by the time Arduino needs the result, a 1 second delay is set.

5. Configuring the Raspberry Pi

As the access control system can only handle one card at a time when selecting a physical database server, performance cost-effectiveness was more important than performance. For this, a Raspberry Pi 2 Model B was used, a low-priced, card-sized computer. Its advantage is that it can run multiple Linux-based operating systems. An 8GB Micro SD card is used as storage space. For speeding up the system, a class 10 card is used, which has one of the fastest reading and writing speed.

The advantages of Linux systems compared to Windows systems are that they are open source and most of them are available for free. This was an important aspect, as most of the servers in the industry are also Linux based.

A LAMP [4] server was configured on Raspberry Pi. LAMP is an acronym for Linux, Apache, MySQL, PHP, which indicates the used technologies. The latest version of Raspbian, Jessie, was installed on the device as an operating system.

Apache is the most widely used Web server for Linux systems. This Web server is able to serve Web sites that are requested by clients (typically browsers, but may be other applications). It supports HTTP (HyperText Transfer Protocol), HTTPS (Hypertext Transfer Protocol Secure), and FTP (File Transfer Protocol) protocols.

Apache provides 2 interfaces for the mobile app to communicate with the server, through these interfaces HTTP requests are made, which are essential in the communication of the Android application and the Python script which affects Arduino's operations.

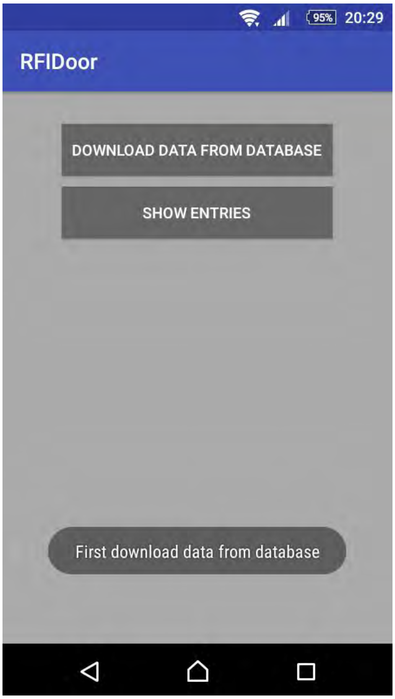
It was important in building the database to use a free, open source solution, so MySQL database was chosen. It has the advantage that the database manager is supported by many programming languages: PHP, C ++, Java, Delphi, Lisp, Perl, Python, Ruby, and so on. To run queries on the database of the completed access control system, PHP version 5 was chosen as this is commonly used with MySQL databases. phpMyAdmin was used to create and visualize the tables.

6. Android application

Because of the rapid development of mobile devices and the release of tablets, Android is one of the leading operating systems today. Tracking the access control system would be uncomfortable with a desktop computer by manually running SQL commands. Hence, the idea was to use a mobile device to control the system.

The application is named RFIDoor. It contains 2 activities: the first is MainActivity, the second is DisplayListView activity. Android applications have four main components: activity, service, content provider, and broadcast receiver. Activities are intended to interact with the user and to create objects, define classes, and so on.

The application supports Hungarian, as well as English. Depending on the localization of the device, with Hungarian localization setting texts are in Hungarian, otherwise in English.



First download data from database

Figure 3. *The app during run*

7. Testing & Conclusion

Since there is no perfect system, the project was subjected to several testing processes. Both hardware and software components have been tested by various methods to assure the safe operation of the system. First, I had to be convinced of the quality and integrity of the tools used.

Additionally, in the simulation environment of Autodesk Circuits [7] the LEDs and resistances were checked if they are suitable for combined use. The system was tested using an 80MBit network and at reduced bitrate, even with a weak network connection, it worked without delay. The database schema can be expanded, but in this project only the essential fields were included in order to maximize performance.

Our project was fruitful, the access control system has successfully passed through the testing process. By using open source Linux based network solutions, and software we have met the core principles of IoT and Industry 4.0 **[8]**. Our advantage over architectures that are similarly designed is that we, as developers, have full control over the system, allowing changes to be made easily, and even on-site.

Acknowledgements

The work/publication is supported by the EFOP-3.6.1-16-2016-00022 project. The project is co-financed by the European Union and the European Social Fund.

References

[1] Alaba F. A., Othman M., Hahsem I. A. T., Alotaibi F.: *Internet of Things Security: A Survey.* Journal of Network and Computer Applications 88. (2017) 10–28.
https://doi.org/10.1016/j.jnca.2017.04.002

[2] Adafruit: *Standard servo. TowerPro SG-5010* (2017, May 14).
https://www.adafruit.com/product/155

[3] Arduino: *Arduino UNO Rev 3.* (2017, May 14).
https://www.arduino.cc/en/main/arduinoBoardUno

[4] Sander van Vugt: *Setting Up a LAMP Server.* The Definitive Guide to SUSE Linux Enterprise Server 12, 309-329.
https://doi.org/10.1007/978-1-4302-6820-8_12

[5] Sanjib Sinha: *Python Environment* In: Beginning Ethical Hacking with Python. Apress, Berkeley, CA, 2017, 39-41.
https://doi.org/10.1007/978-1-4842-2541-7

[6] Murat Yener, Onur Dundar: *Android Application Development with Android Studio*. In: Expert Android® Studio, John Wiley & Sons Inc. 2016, 45-79.

[7] Autodesk Library.IO: Autodesk Circuits, (2017, May 14).
https://circuits.io/

[8] Trappey A. J. C., Trappey C., Govindarajan, U. H. Sun J. J.: *A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0.* Advanced Engineering Informatics 33. (2017) 208-229.
https://doi.org/10.1016/j.aei.2016.11.007