Smart watch access control application based on Raspberry Pi platform

Marina Antić*, Miloš Kosanović*, Slavimir Stošović*
*College of Applied Technical Sciences / Computer sciences, Niš, Serbia
marinamantic94@gmail.com, milos.kosanovic@vtsnis.edu.rs, slavimir.stosovic@vtsnis.edu.rs

***Abstract*—In this paper we described a system that allows access to the laboratory by unlocking the electromechanical lock with the smartwatch device, collecting the data about the working conditions (temperature and humidity) in the laboratory, keeping records of the present members and the preparation of the reports of the working time. The main hardware component of this system is the Raspberry Pi, which serves as the local server for the database of the registered users. Creating the user accounts and the registration of smartwatch devices for the control is performed in the PHP Web application. The software modules are realized by HTML, CSS, JavaScript, jQuery, Python, and PHP with the integration of the MySQL database.**

I.INTRODUCTION

The modern era has introduced the computers in all the areas of life which leads to complete automatization. The concept of connecting embedded devices within the existing Internet infrastructure is called the Internet of Things *(IoT)* [1]. It allows objects to be sensed and controlled remotely across the existing network infrastructure, thus creating opportunities that surpass the current communication of the two machines and cover various protocols, domains, and applications. Thereby, it refers to a wide range of devices such as air conditioner, washing machine, implants for cardiac monitoring, bio-chips installed in the domestic animals, vehicles that help in saving or acquisition of data in inaccessible terrain, such as VTS Explorer [2]–[3].

In order to enhance the learning and the understanding of computing, in the recent years many concepts of the cheap hardware have been developed that allow users to glimpse “under the hood“ without fear of making a damage to the expensive device. The two most popular concepts are known as Arduino [4] and Raspberry Pi [5].

Arduino is a physical computing platform based on open source panel with the simple input/output pins. Through digital input/output and analog input connectors, an Arduino microcontroller can receive signals from the environment or control the other electromechanical components, but still has more modest performances than the hardware called Raspberry Pi.

Raspberry Pi (RPi) is a computer of the small dimensions, with the aim of promoting computer science in schools [6] [7]. This credit card sized computer can perform many tasks just as a desktop computer, such as creating the spreadsheets, the word processing, playing the video games, etc. It also has the ability to play HD videos. Raspberry Pi can run several versions of Linux and it is used around the world in the various fields, from teaching programming to children, to building robots, to the home entertainment system.

Mobile phones are increasingly being used to control and manage the aforementioned systems. Nowadays, as the popularity of the wearable devices is also in constant increase, the smartwatch has become one of the commonly used devices in such systems. As for the smartwatch devices, nowadays, many companies have stood up for realization of such devices (LG, Samsung, Sony, Asus, Apple, etc.). In this project, we exclusively used the Samsung's wearable device, smartwatch model known as Samsung Gear S [8]. Applications for this device are being made for Tizen operating system, using the following technologies: HTML, CSS, JavaScript, and jQuery.

Within the system described in this paper, for the control of working time and the access to laboratory, Raspberry Pi is connected with the electromechanical assembly and is controlled by the application created for smartwatch and smartphones devices. PHP Web [9] application is also created and used to control the database of user accounts. Data recorded in MySQL [9] database can be used to create appropriate reports on working time.

In the second chapter, the basic hardware features of Raspberry Pi microcomputer used in this project are described. After the third chapter and a description of the system, in the fourth chapter there is a detailed scheme of the implemented electromechanical assembly. Chapter five gives a detailed description of the functionality of PHP Web application, while chapter six describes the specifics of the smartwatch application.

II.RASPBERRY PI

Ability to install several different operating systems (Debian GNU / Linux, Fedora, Arch Linux, Risc OS, and other), as well as programming in many languages (Scratch, C, C++, JAVA, Perl, Ruby, Python) with low price and small size, have made the RPi computer very popular.

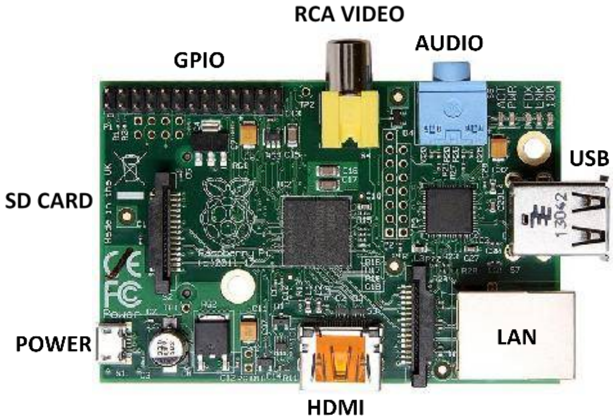

Figure 1. Raspberry Pi model B

The main component that allows small size of this computer, while keeping the powerfulness at the same time, is Broadcom BCM2835 system-on-chip containing RM1176JZFS processor core with floating point and clock speed of 700MHz (which is much better compared to Arduino's 16MHz), and VideoCore 4 graphics processing unit (GPU). Graphics processing unit makes it possible to use Open GL ES 2.0 and enables the decoding of 1080p30 H.264 signals and calculating general purpose at speeds of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPSs. This means that the Raspberry Pi can be connected to an HDTV and Blu-Ray high-quality video can be watched using the H.264 codec to 40Mbits/s.

Raspberry Pi Model B, used in this project (Figure 1), has 10/100 Ethernet port to allow surfing the Internet or using it as a web server, which is utilized in this project. Most of the Linux systems for RPi can be easily stored on SD card of 2GB, but larger cards are also supported. It also has a standard connector for Raspberry Pi camera. RPi model B has two integrated USB ports for connecting mouse and keyboard, but in order to connect multiple devices, USB HUB is used. It is recommended to use a hub with power supply in order not to overload the voltage regulator on the motherboard. Power supplying the Raspberry Pi is very simple; it is only needed to turn on any USB power in the micro USB port. The power button doesn't exist, so the RPi launches as soon as it is connected to the power supply, and it shuts down by simply removing the power supply.

What makes this computer exceptional is GPIO port of general purpose, with eight I/O pins through which additional sensors, relays or complete devices can be controlled. In this project, these pins are used for signal transmission in order to control electrical circuits, in this case an electric lock. The necessary power for Raspberry Pi (5V) is brought by this connector.

III. ELECTROMECHANICAL SYSTEM DESCRIPTION

The scheme of electromechanical assembly is shown in Figure 2. Communication of passive and active components with Raspberry Pi is achieved through a simple interface created on grid plate (Figure 3).

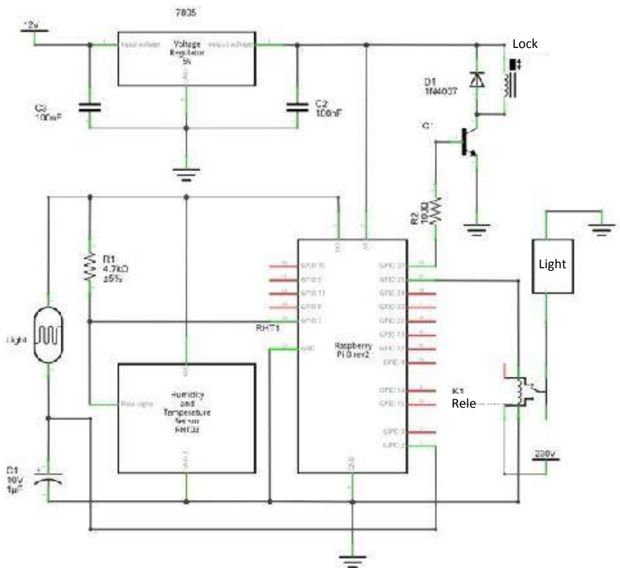

Figure 2. Scheme of electromechanical assembly

The entire circuit is powered with a 5V voltage regulator that provides L7805. At the entrance of this circuit we can bring maximum 40V, and then at the output we get stable 5V voltage. It is recommended to use a block capacitor at the input and output of the L7805 circuit, which ensures stable function and prevents regulators to self-oscillate.

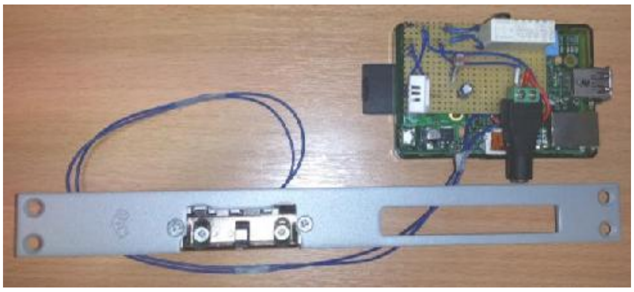

Figure 3. Electromechanical assembly Raspberry Pi with peripherals

Raspberry Pi and transistor switch are supplied by voltage of 5V, and their role is to control the power for activating the electromagnet in the lock. The switch is created using NPN transistor BDP947, marked on the chart as Q1, and its benefits are very small size and large collector current which is essential for activating electromagnet. 1N4007 diode is connected on electromagnet of the lock and its role is to protect the transistor from the reverse electromotive force that is present while switch is in the OFF mode. The impulse for “opening“ the lock is brought with pin 13 of Raspberry Pi through the resistor R2 (100 Ω) to the base of a switching transistor. At the moment, when impulse is on the base, current flows through the transistor and electromagnet of the lock, which leads to unlocking.

Sensors allow data acquisition in the laboratory, while Raspberry Pi camera’s role is to control video surveillance. Sensor that is used in this paper is temperature and humidity sensor DHT-22, and light-dependent resistor (LDR), which is also realized on RPi, thus giving the value of brightness in specific area.

IV. SYSTEM FUNCTIONALITY DESCRIPTION

Functionality of the entire system is illustrated in Figure 4. All system users must be registered in the database of laboratory members. Since the database is unique for working time and access control system and VTS Apps Team's Portal PHP application, this base is located on a web server and also on the Raspberry Pi.

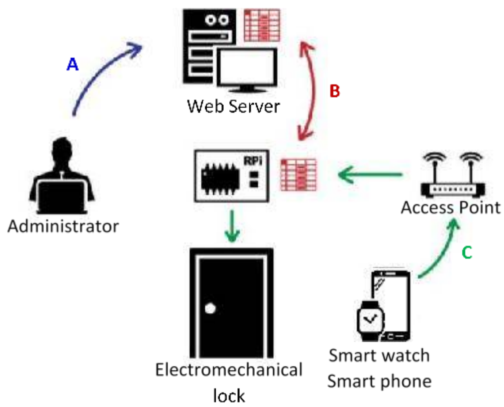

Figure 4. Scheme of realized system

So, the first step for administrator is to create user account (Figure 4 step A) by filling in Web form in PHP application that is shown in Figure 5. In this way, the user gets a universal username and password to access all the above mentioned applications of the Team.

While creating account, PHP application checks whether the entered username for a new user is already in use. At the same time, it checks form validation, in terms of checking the required fields and whether the data is in the correct format. If the created account meets all the conditions and safety checks, PHP stores account with detailed information of user in MySQL database and synchronizes the base with the one located on Raspberry Pi server (Figure 4 step B). As a consequence, the system can be used even without active internet connection.

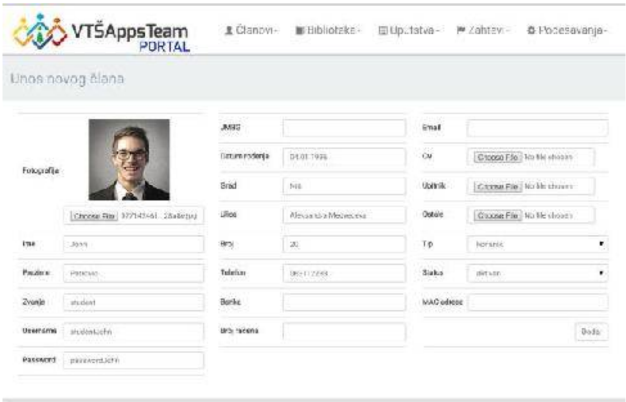

Figure 5. PHP Form for creating user account

After that, a user installs VTSAccessControl application on smartwatch device. The user can register only one device at the moment. Changing a registered smartwatch device for system control includes checking out of the previously registered device. When smartwatch application starts, Splash Screen appears first (Figure 6).

Figure 6. Splash Screen

Figure 7. Login Form

Appearance of login form that will appear after the Splash Screen depends on the results of the following checks:

1. Whether it is the first time application is started,
2. Whether the user was registered through smartwatch application,
3. Whether the user is approved by the administrator.

These three checks are performed by reading the file located in the internal memory of the device, which is encrypted by the decryption key, SHA-256 hash.

If the application has started for the first time, which means that user account has been created but the device has not been registered yet, the login form appears (Figure 7). User needs to fill in two required fields for username and password. By clicking on button “Sign up“, HTTP POST request is sent from smartwatch application to PHP application with the following parameters:

- token (the security key) encrypted by sha1 hash
- MAC address of the device
- username and
- password

PHP application gets HTTP request by smartwatch application and then processes it by examining all parameters that have been sent and searches for users in MYSQL database according to certain criteria (username and password), and then corresponds with specific *response* code.

Based on the received code, smartwatch application makes a decision which procedure should be realized. There are several codes that smartwatch application can receive from PHP application and these are:

- user with specific username and password doesn't exist in the database,
- token is not valid (security key),
- attempt of signing in with information of existing user but without the same registered device (attempt will be saved in the table of unauthorized users),
- user sent request for device registration,
- user is successfully registered.

If smartwatch application gets a response that user has successfully sent a request for device registration, new form will appear, called Wait Screen, with a message that the request for registration is successfully sent and that account will be activated as soon as administrator confirms it. At this point, PHP application sends a notification email with request for device activation. Each time the application is started, but after the device registration step, Splash Screen immediately redirects the user to the Wait Screen and checks whether the account is activated through HTTP POST request, after which the user is redirected to the Home Screen (Figure 8).

On the Home Screen the user has the information of the current time and access to three buttons for unlocking the door, check-in, and check-out. While Home Screen form is displayed, background checks whether there is an active Internet connection (WiFi, GPRS) and whether the user is connected to a particular VTSAppsTeam Access Point network device with predefined MAC address (Figure 4 step C). If there is an interruption of Internet connection, a dialog box appears with notification and shortcut for reactivating the Internet.

Only under the condition that the user's smartwatch device is connected to the proper Access Point, all the options on the Home Screen will be enabled. In that way safety is provided so the door cannot be unlocked unless the user is physically nearby the laboratory. In order to unlock the door, check-in or check-out, the smartwatch device in local network communicates with RPi through a socket whose server is written in the Python programming language. The procedure is as follows:

1. JSON object is created on smartwatch application with following parameters: username, password, MAC address, action (action that needs to be done), token (the security key),
2. JSON object is forwarded via socket,
3. RPi decodes the received parameters (decode them also in JSON) and checks the token and action that needs to be done,
4. When all checks are done, RPi sends an HTTP request to a remote server with PHP application and MySQL database,
5. PHP processes the request and inscribes data such as username, time of check-in/check-out, and responds with particular code message,
6. If the request does not pass the security check, the device is disconnected from socket.

Figure 8. Home Screen

V. SMARTWATCH APPLICATION DESCRIPTION

Lately, watches are not only devices for indicating time, but are also significant in providing brief information and have ability to serve as a kind of remote control that is always available at hand. Due to increasing popularity of application development for smartwatch, in this paper, a special emphasis is on this type of control.

Tizen is open and flexible operating system designed to enable support for different devices (mobile phones, watches, bracelets, cars, gaming consoles, televisions, etc.). Tizen is developed by a community of developers under an Open Source license and is open to all members who wish to participate. Tizen operating system comes in multiple profiles to meet all the requirements of industry. The current profiles are Tizen IVI, Tizen Mobile, Tizen TV, and Tizen for Wearable. It also offers complete flexibility with HTML5 support.

For Tizen application development, in this case smartwatch application, Tizen SDK is used because of the wearable development environment. Application in this project is developed using HTML5, CSS3, JavaScript, and JQuery technology. For example, to create a form for entering a username or password, it is necessary to include the following line of code:

```
<form>
    Sign up:<br>
    <input type="text" name="userN" value="Username"><br>
    <input type="password" name="psw" value="Password"><br>
</form>
```

Once the structure is generated, the layout and design of HTML elements are readjusted with help of CSS (Cascading Style Sheet). All the functionality is created using JavaScript language and its library, JQuery. JQuery simplifies and shortens the writing code.

Due to the small size of smartwatch screen (360x480) in pixels, a user interface creation represents a particular challenge. As mentioned earlier, there are three buttons for unlocking the door, check-in, and check-out that perform particular action depends on which button is pressed. These buttons are realized as *div* elements in HTML and CSS adds style to them through classes and IDs. HTML structure is done with following code:

```
<div id="unlock" class="container"></div>
<div id="checkIn" class="container-small"></div>
<div id="checkOut" class="container-small"><div>
```

Sliding finger from the top edge of the screen to the bottom presents leaving the application (exit).

VI. CONCLUSION

The system described in this paper was created out of the need to allow more users using the laboratory (one room) with a certain level of control. Smartwatch and smartphones devices or more precisely smart applications, are used as a “key“ to unlock the electromagnetic lock. Raspberry Pi microcomputer is the core of this system and has the task of keeping the data in user database. PHP Web application is also developed and enables managing user accounts, sending particular email messages and notifications. MySQL database is in charge of keeping records of arrival and departure times.

All described functionalities are confirmed in numerous experiments in real time, and another advantage of this system is its modularity, i.e. ability to add new functionalities.

ACKNOWLEDGMENT

The application was realized by members of the VTS Apps Team within the Samsung Apps Laboratory at College of Applied Technical Sciences in Nis. The authors are thankful to Samsung Electronics Adriatic, Belgrade Branch for their support in this work.

REFERENCES

[1] O. Vermesen, P. Fries, *Internet of Things - Converging Technologies for Smart Environments and Integrated Ecosystems*, River Publishers, 2013.

[2] VTŠ Explorer 1 promo film, http://www.youtube.com/watch?v=yKZJHq831Vg, january 2016.

[3] VTŠ Explorer presentation film, https://www.youtube.com/watch?v=Ph6pR9QKd88, january 2016.

[4] John Baichtal, Arduino for Beginners: Essential Skills Every Maker Needs, Amazon, 2013.

[5] http://www.raspberrypi.org/, january 2016.

[6] http://www.bbc.co.uk/blogs/legacy/thereporters/rorycellanjones/2011/05/a_15_computer_to_inspire_young.html, january 2016.

[7] Russel Barnes, *The Official Raspberry Pi Projects Book*, Raspberry Pi Trading Ltd, 1st edition, 2015.

[8] http://www.samsung.com/global/microsite/gears/gears_featurfe.html, january 2016.

[9] Hugh E. Williams and David Lane, Web Database Applications with PHP & MySQL, O'Reilly Media, may 2004.