Chapter

Embedded Systems Based on Open Source Platforms

Zlatko Bundalo and Dusanka Bundalo

Abstract

Ways and possibilities for design, implementation and application of microcomputer-based embedded systems using open source hardware and software platforms are considered, proposed and described. It is proposed to use open source hardware and software microcomputer-based technologies for design and implementation of embedded systems in many practical needs and applications. Main advantages and possibilities of application and implementation of such embedded systems are considered and described. Two practically designed and implemented systems performing needed data acquisition and control are presented and described. Used technologies for realization of the systems and embedded applications of the solutions are described. Open source microcomputer boards, appropriate sensors, actuators and additional electronics are used for implementation of the systems hardware. Open source tools and programs and LINUX operating system are used for implementation of the systems software. Modular approach is applied in the systems design and realization. Easy system expandability, simplifying maintenance and adaptation of the system to user requirements and needs are enabled with such approach. Balance between functionality and cost of the systems was also achieved. Optimization according to user requirements and needs, low consumption of electrical energy and low cost of the system are main advantages of such systems compared with standard embedded systems. These systems are optimized and specialized systems for specific needs and requirements of users.

Keywords: microcomputer-based embedded systems, open source hardware and open source software, LINUX operating system, modular system design, low cost of system, low consumption of electrical energy, system specialization and optimization

1. Introduction

Rapid development of electronic, computer and information technologies has enabled design and implementation of different types of so called embedded systems used for many different needs and purposes, for many practical applications. Such systems are designed and implemented to be maximally adapted to concrete application and to efficiently perform specific needed function. Also, such systems should be as simple as possible, with as much small dimensions as possible, as cheaper as possible and to consume as little energy as possible. Those systems are always specialized and optimized systems for some specific purpose and are used in almost all areas of human activities [1–4].

Different hardware and software solutions, different hardware and software platforms, can be used for development, design and implementation of such embedded systems. Those are mostly microprocessor-based solutions and platforms. What practical solution will be used primarily depends on specific requirements that such system should satisfy, in accordance with concrete need. Specially designed solutions and specially designed platforms are most often used for such purposes [1–4]. But, it can be shown that in many practical needs it is much more convenient and better to use so called open source hardware and software solutions, platforms of open source type [1–10].

Possible ways of design, implementation and application of embedded systems based on open source microcomputer hardware and software platforms are proposed and described here. Using open source hardware and software microcomputer-based technologies for design and implementation of embedded systems in many practical needs and applications is proposed. Main possibilities and advantages of such embedded systems are considered and described. Two such practically designed and implemented systems that perform appropriate data acquisition and control are presented and described. The systems hardware was implemented using open source microcomputer boards, appropriate sensors, actuators and additional electronics. The systems software was implemented using open source tools and programs and LINUX operating system.

The remainder of this chapter is organized in a following way. Section 2 considers basic concepts of embedded systems, main characteristics, way for design and motivation for practical application and implementation of such systems. Section 3 describes main concepts of proposed embedded systems design and implementation based on open source platforms. This section proposes way of design and implementation of open source hardware and software-based microprocessor embedded systems as specialized systems optimized for concrete user need. It also describes advantages of such design and implementation. Section 4 describes design of one practically implemented open source-based microprocessor embedded system for user identification in access to objects using mobile phone, as specialized systems optimized for user need. Section 5 shows and in more details describes design and implementation of one such based SCADA and RFID embedded system for industrial process control application, implemented using LINUX platform. Finally, a short summary and conclusions are given in Section 6.

2. Embedded systems

Embedded systems are electronic systems with very strong integration of hardware and software, designed to perform some specific functions [1–4]. Those are computer-based electronic systems built into another system, for what they provide better functionality and performance. Embedded systems are special purpose systems where the system is fully encapsulated by the device it controls. Such system performs one or more predefined tasks, usually with very specific requirements. Since system performs specific task, designer can optimize it in speed of operation, also reducing system dimensions and weight, energy consumption and cost. Embedded systems are often products of mass production and mass consumption. Such systems use microcomputer-based implementation [1–4].

Some of the basic characteristics of embedded systems are: performing specific tasks, supported by wide range of processors, real time limitations and should be cheap. If operating system is used it usually is Real Time Operating System (RTOS). In many cases such systems are battery powered and low-power systems, often must operate in extreme ambient conditions, have system resources necessary only for concrete use. Such systems keep all object code in ROM and require use of special means and methods for design [1–4].

There is wide range of application areas of embedded systems. Such systems are generally very inexpensive and are used in almost every electronic product. Some of main areas of embedded systems application are: computer peripheral devices, automotive electronics, aircraft electronics, trains, telecommunications, medical systems, military applications, authentication systems, consumer electronics, fabrication equipment, smart buildings and robotics [1–4].

Hardware design of embedded system includes selection of processor and interface logic for connection with environment. It also requires achieving balance in design and decision what will be realized by hardware and what by software. Decision how to divide design into hardware and software part is a key element in design of embedded system.

Characteristic of embedded systems is that cross development is usually used in software development. Generally, the embedded system software is developed on one platform and is executed on another platform. Also, it is required to obtain as efficient as possible programs in machine language. Since compilers do not generate efficient machine codes, programming in assembler language is often used. But, embedded applications are becoming more complicated and there is need to use programming in high level programming languages. Generally, there is no ideal programming language for programming of embedded systems. The most often used programming languages in embedded systems are: C, C^{++}, Java and DFL [1–4].

Processors used in practice can be divided into two groups: general purpose processors and special purpose processors. In the initial period embedded systems were using general purpose processors. Development of VLSI IC technologies enabled that embedded systems are mainly realized using special purpose processors (so called embedded processors). A key characteristic of applying of such processors is that designer should well know nature of concrete application, to be able to meet requirements and to select appropriate processor for a system. There are different classes of embedded processors: microcontrollers, RISC processors, Digital Signal Processors (DSP), multimedia processors, Application Specific Instruction Set Processors (ASIP). There are also classes of such processors for achieving high performances in intensive computing applications. Some of typical such processors are: network processors, digital signal processors (DSP), SoC (System on Chip) processors [1–4]. Advantages of using general purpose processors are: low cost for research, design, develop and test, high flexibility, low time to market. Advantages of using special purpose processors are: fast, low power consumption and small size solutions [1–4].

Depending on specific application and required characteristics, and considering the platform used for development and implementation, there are practically possible three ways of implementation of embedded systems:

- Embedded systems based on specially designed platform,
- Embedded systems based on open source platform,
- Embedded systems based on PLC (Programmable Logic Controller) platform.

Embedded systems based on specially designed platform use specially designed hardware platform, designed only for concrete application. The platform can be based on general purpose microprocessors or on special purpose processors. FPGA-based processor platforms or SoC platforms are very often used. In design of the platform everything is optimized and minimized, in order to satisfy all needs of concrete application. As far as software platform is concerned in such systems, the assembler language for programming is mainly used. Most often, operating system is not used and all programs are application programs. Also, all software is minimized and optimized, with the goal to meet all application requirements. Standard hardware and software development tools, or even specially developed tools, are used for development of such systems. Advantages of such design and implementation of embedded systems are: minimal dimensions and weight, minimal energy consumption, minimal price, maximal speed of operation, full optimization and minimization of all hardware and software resources. Disadvantages of this approach are: very long development time and great development costs, long time to market. This method of design and implementation is used when it is needed to realize and produce very large number of such systems, in products of mass consumption and mass production. In such situations this method has the best economic effects and such are obtained the systems with the lowest market price.

Embedded systems based on open source platform use some of on the market available open source hardware platforms. Such platforms are based on general purpose microprocessor or microcontroller or on DSP processor. For design and implementation of concrete system it is needed to select open source hardware platform, among all available. That platform should be the simplest and the cheapest and should meet all requirements of application. Concrete platform is chosen such that entire system be optimized and minimized, but with the goal that all requirements for application are met. As far as the software platform is concerned, high level programming languages are mainly used for programming in such systems. Rarely is used assembly language. Often, operating system is used in such systems, usually RTOS. Entire software is minimized and optimized with the aim to satisfy concrete purpose. Standard hardware and software development tools are used for development of the systems. Advantages of such method of design and implementation are: smaller development time and lower development costs, less time to market, optimization of all resources, and relatively high speed of operation. Disadvantages of this method are: greater dimensions and weight, higher energy consumption, higher price. This method should be used when it is needed to realize and produce relatively smaller number of such systems. In such situations this way of design and implementation has the best economic justification and gives the best economic effects.

Embedded systems based on PLC platform use some of on the market available PLC devices as hardware platforms. Such platforms are based on general purpose microprocessor or microcontroller or on DSP processor used in some available PLC device. For design and implementation of concrete system it is needed to select PLC platform, among all available. That platform should be the simplest and the cheapest, and should meet all needs of application. Concrete PLC platform is selected such that entire system can be optimized and minimized, but with the goal that all requirements for system application are met. As far as the software platform is concerned, in such systems mainly are used tools and languages that are used for PLC programming. Whole software is minimized and optimized according to application and with the aim to satisfy purpose of the system. Standard resources and tools for development of PLC-based systems are used for development of such systems. Advantages of such way of design and implementation are: high speed of operation, high reliability, short development time and low development costs, short time to market. Disadvantages of this way are: large dimensions and weight, high energy consumption, high price. This method is used when is needed to realize and produce small number of such systems. In such situations this way of design and implementation has the best economic justification and gives the best economic effects.

3. Design of embedded systems based on open source platform

High performances, high speed of operation, minimal energy consumption and large number of such solutions are not required in many practical needs and applications of embedded systems. In such applications the most important is that all needed system functionalities are met, with as short as possible system development and implementation time, and as little as possible system development and implementation costs [1–10]. It is also required as much as possible optimization and minimization of hardware and software, to achieve needed speed of operation. Also, it is needed as little as possible dimensions, weight and energy consumption, as high as possible reliability, as easy as possible maintenance, and as low as possible price of the system. For such purposes the best is to use embedded systems based on open source platforms. Such way of design and implementation enables simplification and acceleration of development, design and implementation process, reduction of time needed for all the activities, and reduction of costs of all the processes. It also enables modular approach, what facilitates, accelerates and reduces costs of system modification and maintenance. It is possible to optimize all hardware and software resources in accordance with needs of user, and to balance all necessary functionalities and price of the entire system.

Use of open source hardware platforms (suitable open source boards) drastically reduces and makes easier and cheaper process of development, design and implementation of hardware and entire system. It is only necessary to choose and select the most suitable open source platform for concrete application in accordance with all requirements of that purpose. It is also needed to select some additional hardware modules that should be added to the selected open source platform. Such is enabled modular hardware approach if it is necessary for concrete application. Other activities in design of hardware include defining ways and means for connecting system with environment, with sensors and actuators, and with user. It is very simple to perform since available hardware open source platforms offer large number of ways and circuits for interconnection and communication. In such design, it is constantly intended to minimize and optimize entire system hardware, and to satisfy concrete application of the system. With this way of design and realization of system hardware, it is practically very possible and feasible, with very low total cost.

Use of open source software platforms (suitable open source software development tools) reduces and makes easier and cheaper process of development, design and implementation of software and entire system. The development tools include open source programming languages and software tools, operating systems and debugging tools. It is only necessary to select the most suitable open source software tools for concrete application. For programming it is mainly used and it is intended to use as much as possible the high level programming languages. Such development of programs is simplified, accelerated and cheaper. Problem could only be in applications where very high speeds of operation are required, where programming in high level programming language can not meet needed speed. Programming in assembler language is used in such situations. But, it complicates, slows down and increases cost of development of the program. Therefore, it is tried to use programming in high level programming languages wherever it is possible. Programming in assembler language is used only in those parts of program where is needed as high as possible speed. It is also necessary to implement software modules that could be used in design of complete system software. Such is enabled modular software approach if it is necessary in concrete application. Also, in such systems it does not have to be used operating system. But some open source operating system can be used. As operating system slows down operation of the system and requires more hardware resources (larger memory), it is not used in applications where high speed of operation and small hardware resources are needed. In applications where it is not critical, it is preferable to use some open source operating system. Use of operating system very much facilitates and accelerates software design, implementation, modification and maintenance in entire system. It simplifies, accelerates and makes cheaper all activities in development and implementation of software and entire system. Further activities in development of software include programming of system communication with environment and with user. It is simple in such systems since available open source software platforms offer large number of different programs (drivers) for such communication for used hardware open source platforms. It is also constantly intended to optimize entire software of system and to satisfy concrete application. With this way it is practically very possible and feasible, with very low total costs. Manufacturers of hardware open source platforms also offer appropriate software open source platforms for their products that are mainly used in software development of such systems. This significantly simplifies, accelerates and makes cheaper development, design, implementation and maintenance of software of such systems.

Optimal integration of hardware and software of entire system is also very important in design of embedded systems. That enables and ensures achieving optimal characteristics of the system with satisfying all requirements, and with balance of hardware and software of system. In design of system, it is needed to decide what will be realized by hardware and what by software. It is always needed to minimize use of hardware and to maximize use of software. As less as possible, functions should be realized by hardware and as much as possible by software. It enables minimization of dimensions, weight, energy consumption and cost of the total system. Therefore, in development and implementation of such embedded systems, from the beginning should be parallel and simultaneously realized design of system hardware and software and their integration and verification. That all is also simplified in such embedded systems since are mainly used open source hardware and software platforms of the same manufacturer.

4. Embedded system for user identification in access to objects using mobile phone

Mobile phones and smartphones are standard communication devices that are used by many users. Mobile phone is microprocessor-based system with different peripheral communication elements and memories. SIM (subscriber identity module) card or smart card in mobile phone can memorize many data about user of mobile phone and different identification data (user name, access passwords and other identification data). It enables that mobile phone could be also used as identification device. The mobile phone could consolidate or replace many identification elements, different types of cards and other elements for identification. Such identification elements are for example: identity card, health card, passport, driving license, student index, remote device for access to a car, remote device for access to some object or area, for entering public transport vehicles, for electronic payments, etc. All that identification elements could be replaced by one mobile phone and identification activities could be performed much simpler. The embedded system that enables mobile phone to be not only communication, but also identification device is proposed and described here. Using such approach and such systems it can be avoided usage of other identification elements in many applications with user identification, for monitoring, tracking and control of access of users to different facilities, spaces and services using a standard mobile phone.

Developed and implemented system is based on application of open source Arduino microcontroller platform and smartphone with open source Android operation system. Bluetooth Low Energy (BLE) wireless technology was used for communication [5, 6]. The implemented solution is used for control of access of user to objects, for control of user access to the car parking place or garage. Only users with needed identification data in their smartphone and with appropriate International Mobile Equipment Identity (IMEI) number of the smartphone can access to the car parking place or garage. The system is connected with computer of PC type that is used as a central station. It gives possibility to realize monitoring of user presence in the object and memorizing and processing of such obtained information and data. It can be connected with other central stations or with main monitoring centre in the case of more complex system realization.

In order to increase reliability of control it is needed authentication of the user [5–8]. There are practically three authentication methods based on: something that user knows (password, key word, etc.), something that user owns (key, token, etc.), something that user is (voice, fingerprint, etc.) [11]. Implemented system uses and combines the first two methods: something that user knows (user name and password) and something that user owns (smartphone with appropriate IMEI number) [12].

4.1 Smartphone in identification of users

One of the most important characteristics of the mobile phone is portability. It allows mobile phone to be permanently close to the user and to be such an ideal element for user identification. Hardware architecture of smartphone is different from architecture of standard general purpose processor. There are significant differences in architecture of CPU used in smartphone [13]. Smartphone uses operating system as an integrated element [13]. The most used operating systems in smartphones are: Google Android, Apple iOS, Nokia Symbian, RIM BlackBerry, Samsung Bada, Microsoft Windows Phone. The application is a part of smartphone software providing additional features of smartphone. By using applications user adds needed functionalities to smartphone.

Bluetooth Low Energy (BLE) and Near Field Communication (NFC) are wireless technologies often used for data transfer between smartphone and other systems [14, 15]. Designers should to select one of the technologies that meet their needs. The BLE wireless technology was used for practically implemented and described embedded system. Reduced power consumption and low cost are the most important advantages of BLE wireless technology. These advantages are reasons for selecting BLE wireless technology for communication between smartphone and Arduino board-based system in the implemented solution.

Users carry many identification cards and devices every day. These identification elements are of different shape and production. That complicates all user activities that require identification. Many of user identification elements could be integrated into mobile phone. It is easier for users to carry one mobile phone with identification data for all identification systems they use than to carry many of different elements for identification. The risk of forgery and malversation also decreases in case of losing or stealing of identification element with some kind of intelligence as mobile phone is.

4.2 Design and implementation of user identification system using mobile phone

Design and implementation of user identification system using smartphone for entrance to the garage or to the car parking space is proposed and described here [5, 6]. **Figure 1** shows structure of the designed and implemented embedded system.

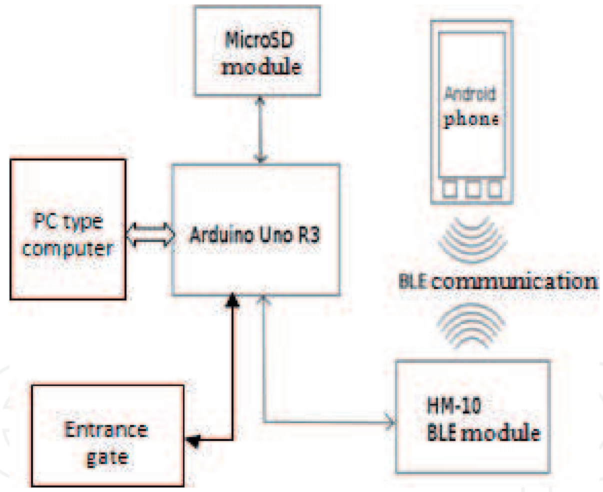


Figure 1.
Structure of implemented user identification system.

Smartphone with the Android operating system is used for entering user identification data at the entrance of the object (garage or car parking space). The microcomputer Arduino Uno R3 platform with ATmega328P microcontroller is used for checking and control of the identification process and for approval of user entry [15, 17]. The mechanism and elements for opening or closing entrance gate of object is connected to and controlled by the Arduino platform. Using Android application on the smartphone user enters username and password. That identification data is transferred via BLE wireless communication to the Arduino platform that is located in the object access point. Smartphone IMEI number is also transferred in the same time. User identification data including smartphone IMEI number is stored in the MicroSD card. Program on the Arduino platform checks out entered user data. User was successfully identified if sent user information is correct (matches with the user data stored on the MicroSD card). It is also obtained permission to access the garage or the car parking and entrance gate is opening. In the case that sent user information is not correct the access is not allowed. Arduino platform is connected with PC type computer that functions as system central station. Information about users, their permissions, accesses, presence and other data are memorized and kept on PC computer and can be analyzed on that computer. The PC type computer can be connected to other such computers (other central stations) and one main monitoring station if it is needed to realize more complex system for access control and monitoring more parking spaces or garages.

Valid user identification data is memorized on the MicroSD module. In that way was avoided entering identification data in Arduino program. It was also enabled to easily add, delete and update identification data using any device that can work with SD card (computer, mobile phone, etc.). Also, user identification would not be possible if the SD card is removed from SD module. It could be very simple mechanism to forbid further use of the system.

Communication and identification data exchange with Android-based smartphone is performed using HM-10 BLE wireless module. Two LEDs connected to Arduino board are used for indicating state of user identification and state of entry gate.

Open source Arduino IDE development environment was used for development of the embedded system [16, 17]. Arduino programming language very similar to C^{++} programming language was used for programming [16, 17]. Open source Android Studio was used for development of Android application [18].

Designed and implemented programs have two parts: program for the Arduino board and program for the Android smartphone. Program part for the Arduino board accepts commands from mobile phone and returns responses. All elements connected to the Arduino board are also controlled by that program. **Figure 2a** shows part of that program. Program part for the Android mobile phone processes answers obtained from the Arduino program and the Arduino board and interacts with the user. **Figure 2b** shows part of that program.

The user can use next command from Android mobile phone while the user is not identified and login:

- login—user identification process initialization (the LED 2 is turned on when user is identified).

The user can use the following commands after user identification:

- enabling access—command indicating that user was identified and enabling that access be enabled and open (open entry or access, this command turns on LED 1),
- disabling access—command indicating that user has left the space or service and that access be disabled and closed (closed entry or access, this command turns off LED 1),
- access status—command that returns information is entry enabled (open) or disabled (closed), that the LED 1 is on or off,
- logout—command by what the user interrupts and stops status of identified user and logout from the system (the LED 2 is turned off).

The Android application program on smartphone has three parts. The first part of the program scans whether active BLE devices exist. If exist active BLE devices near they will be shown in the list of devices to what the user can be connected. The second part of the application program is started when device from the list was selected and the connection process started by user pressing button "CONNECT". The user enters identification data (username and password) in this part of the

Figure 2.
Part of Arduino program (a) and part of android program (b).

program. Process of identification starts when user enters the data and presses the button "LOGIN". IMEI number of user smartphone is also transferred in the same time. User will receive message about wrong identification and will be forbidden system usage if the user enters wrong data or if IMEI number is wrong. The user is identified and goes to the third part of the application program if the transferred data is correct. The third part of the application program is the area where the user can stay or can leave the state of identified user and can return to the second part of the application program.

The application program on Android smartphone interacts with Arduino program part for identification. List of BLE devices to what users can be connected appears after starting Android application. It can be also restarted the scan process by pressing the button "REFRESH". It enables to find BLE devices that were later entered into the range of mobile phone or previously were not found for some reason (**Figure 3a**). By selecting one of the BLE devices from the list it is selected and marked that BLE device. Connection with the selected BLE device is performed by press on the button "CONNECT" (**Figure 3b**). It will be displayed the message "Please select device from list to connect" if the user was not selected any BLE device.

The application goes to login process and login form and starts identification of user after application connects to BLE device. **Figure 4a** shows that. User enters username and password in this form. By press on the button "LOGIN" it is performed sending of user identification data including user smartphone IMEI number to Arduino system (**Figure 4a**). User will receive an error message if sends wrong data (**Figure 4b**).

The user is login on the system and in Android application if the identification process was successful. LED diode (LED 2) connected to output of the Arduino board turns on (indication of user identification). User then goes on process and the form where can control entrance gate and LED diode (LED 1) connected to output of Arduino board (indication of access control). **Figure 4c** shows smartphone display form for the case when the user is identified and login.

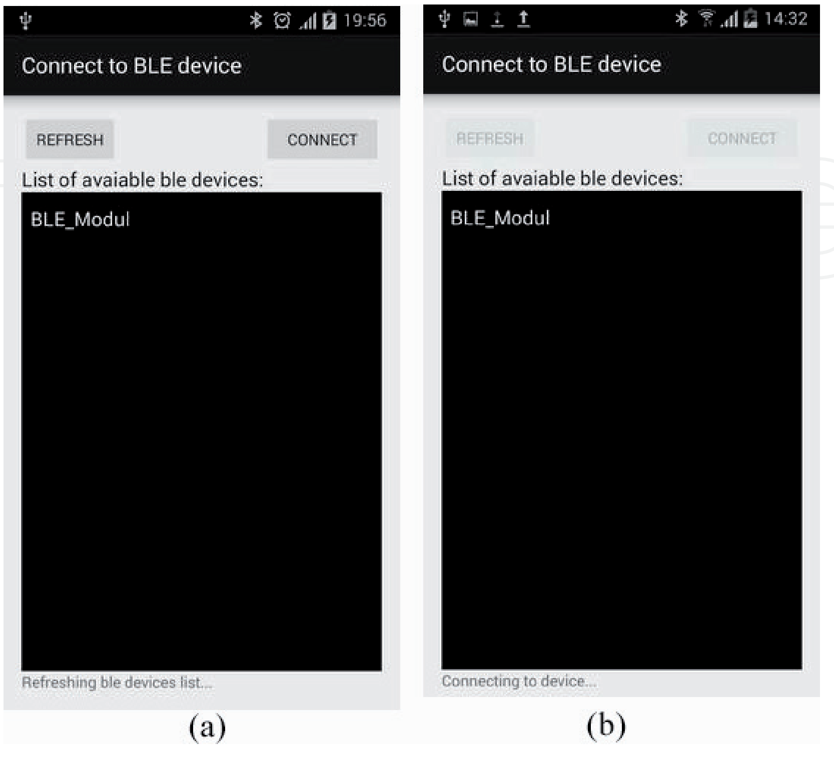
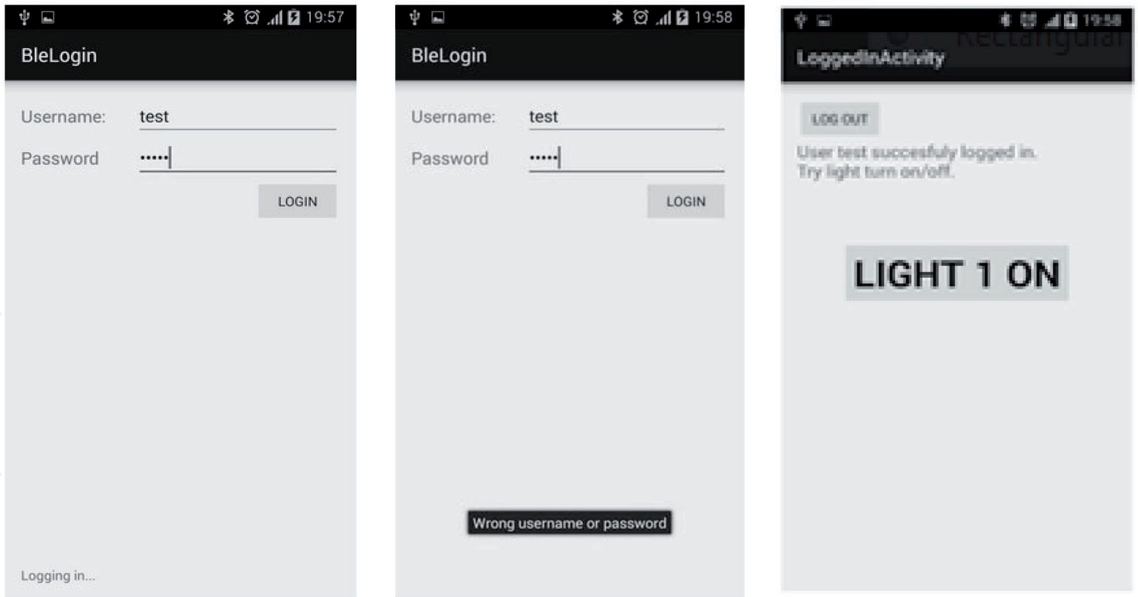


Figure 3.
Refresh of BLE devices list (a) and connection to selected BLE device (b).



(a) (b) c)

Figure 4.
User started with login process (a), user entered wrong user data (b) and login user form (c).

The implemented and described system shows that such way of use of embedded system, smartphone and wireless communication can very much simplify user daily and business activities and reduce costs. It can be simply enabled to access many places, objects and services and to perform complex operations and send and receive various data by using smartphone for identification. It is also used smartphone IMEI number to avoid misuse of the system by unknown and unwanted person.

5. Embedded SCADA and RFID system implemented on Linux platform

Microcomputer embedded systems are present in almost every part of human activities and environment. Application of such systems enables needed functionality, mobility and autonomy of many implemented systems with low cost. Some of many such systems applications are: acquisition, processing and storage of data from process; monitoring and control of processes; increasing security in the process. All this leads to so called Supervisory Control And Data Acquisition (SCADA) systems used for monitoring, control and data acquisition in industry systems [19, 20]. One of the most important requirements in industrial processes is security. So called Radio-Frequency Identification (RFID) is often used as security technology for identification of participants in the process [21, 22]. RFID is standard technology for identification objects and users in many cases because of contactless identification, relatively high speed, simplicity and low price.

Possibilities for design, implementation and application of embedded SCADA and RFID systems realized using open source technologies are considered, proposed and described here. One practically designed and implemented system for data acquisition and needed control in one industrial process is described [9, 10]. System hardware was implemented using open source microcomputer boards Pandaboard [23], Arduino/Genuino Uno and Mega [24], RFID sensors, temperature and air humidity sensors, WiFi router and additional electronics. Software was implemented using open source tools and programs on LINUX operating system [25]. Acquisition data of the system are: identification numbers of RFID sensor and RFID tag, temperature, air humidity, time of identification. Implemented system enables:

displaying information on LCD (Liquid Crystal Display), control using RFID, control and displaying of information using WUI (Web-based User Interface) and CLI (Command Line Interface), wireless and wired communication with user terminals.

5.1 Microcomputer based embedded SCADA and RFID systems

Standard SCADA systems are based on using PLC devices [19, 20]. PLC is a special type of controller based on microcomputer that works in real time and is used for control of processes and systems. PLC is very common in control systems because of its resistance to environmental conditions, reliability, functionality, ease of programming and maintenance. It was created possibility for development and implementation of embedded microcomputer-based SCADA systems for specific applications with progress and development of microcomputers. Main advantages of such systems are: optimization of system hardware, software, speed and storage needs; maximal adoption according to application, low cost. SCADA systems are consisted of: measurement equipment (sensors) and executive units (actuators), remote input/output modules, remote stations (terminal units), communication system and central station.

Radio-frequency identification is used in RFID systems [21, 22]. Process of identification is performed such that the RFID sensor detects the RFID tag brought in proximity to the sensor. It is performed identification of a person or of an object based on data in the tag. After successful identification it is enabled access of a user or an object to needed functions. RFID readers/writers use electromagnetic field for automatic detection and monitoring RFID tags. There are three types of RFID tags: passive RFID tags, active RFID tags and partly active RFID tags. Diversity of RFID tags enables selection of the most appropriate solution for a given application.

5.2 Design and implementation of SCADA and RFID system

The practically developed, designed and implemented system is specialized embedded SCADA and RFID system realized using open source hardware and software technologies [9, 10]. Implemented system enables:

- Control (turn on/turn off) of two high-power (energy) consumers using WUI, CLI and RFID sensors,
- Monitoring and changing description of the consumers,
- Acquisition and storage of information from RFID, temperature and air humidity sensors into data base,
- Two types of users (system administrator and ordinary users),
- Different authorizations for administrator and ordinary user,
- Authorization of user when accessing the system,
- Addition and change of users by administrator using WUI,
- Addition and change of RFID tags by administrator using WUI,
- Record of addition and change of users in the data base,

- Record of addition and change of RFID tags in the data base,
- Record of turning on and turning off of high-power consumers in the data base,
- Access to the system by SSH (secure shell) protocol using RSA cryptographic algorithm with key length of 2048 bits,
- Monitoring of the resources, network and processes of the system,
- Display of last 100 logs using WUI.

Process of design and implementation of the system was performed in next phases: hardware design, Bootloader design, Linux kernel and Device Tree design, ROOTFS and supporting applications, preparation of SD (Secure Digital) memory card, microcontroller programs design and programming. It was used one systematic approach for proper and optimized design of the system with the aim that the system performs all required functions and operations. Block scheme of designed and implemented system is shown in **Figure 5**.

The hardware in implemented system consists of open source elements: Pandaboard, Arduino/Genuino boards, LCD display, WiFi router, appropriate electronics and sensors.

The Pandaboard is open source development board based on dual-core 32-bit high-performance multimedia application microprocessor Open Multimedia Application Platform 4430 (OMAP 4430) and 1GB RAM and SoC (System on Chip) architecture [23]. Microprocessor OMAP 4430 is based on ARM (Advanced RISC Machine) Cortex-A9 architecture with very good performances and very small electrical energy consumption. The Pandaboard enables support for higher level operating systems and has performances that satisfy all needed requirements for central station in implemented SCADA and RFID system.

The Arduino/Genuino Uno and Arduino/Genuino Mega 2560 are open source development boards based on high performance, low power Atmel AVR 8-bit microcontrollers ATmega328P and ATmega2560, respectively, with supporting programs and libraries [24]. In the system design were used the Arduino/Genuino development boards to decrease time for performing needed operations and to increase possible distance between the system modules.

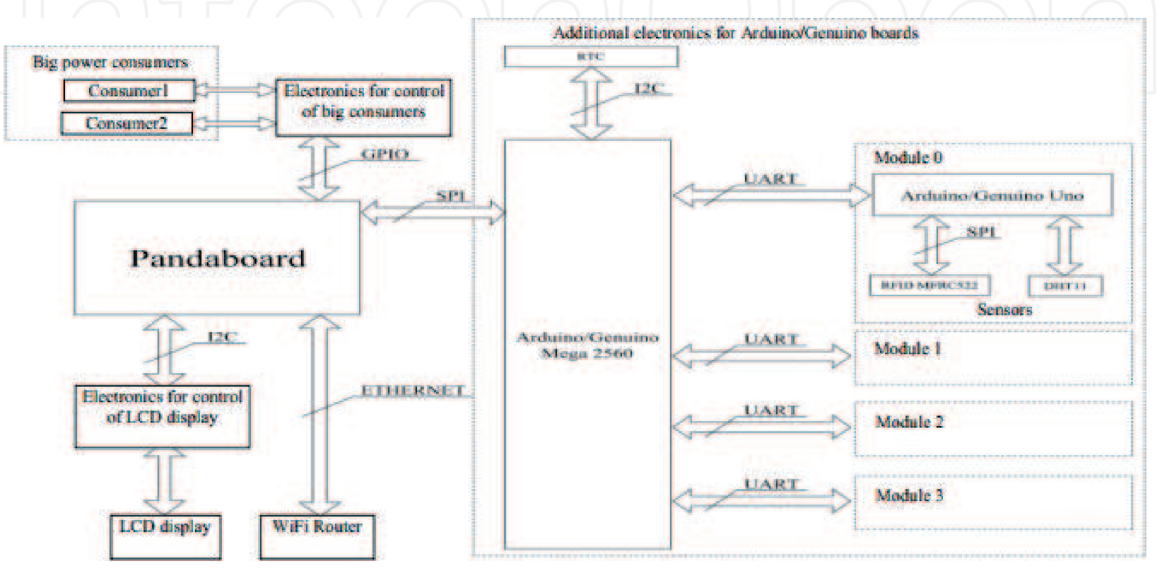


Figure 5.
Block scheme of implemented SCADA and RFID system.

Software of the system was implemented using open source software tools and programs. Appropriate open source tools and programs were used for development and implementation of software for Pandaboard and for Arduino/Genuino boards.

For implementation of hardware of the system were used: Pandaboard as central station, Arduino/Genuino boards as remote stations (system modules) and part for communication with the modules, additional electronics for the boards, electronics for control of high-power consumers, electronics for communication with LCD display, WiFi router TP-Link TL-WR740N. For implementation of additional electronics for Arduino/Genuino boards were used: Arduino/Genuino Mega 2560 board, I2C RTC (Real-Time Clock) based on DS3231 and AT24C32 circuits, modules.

Arduino/Genuino Mega 2560 board realizes communication of remote modules with the Pandaboard as central station. Such is enabled communication of the Pandaboard with four modules. Total number of remote stations in the system can be increased using multiplexors. The RTC circuit that provides information about the exact time even in a situation when the entire system is left without the main power supply source is connected on the Arduino/Genuino Mega 2560 board. Such operation is enabled by a battery integrated into the RTC circuit.

For implementation of the module (remote station) were used: Arduino/Genuino Uno board; temperature and air humidity sensor DHT11; RFID sensor MFRC522. The remote station enables system management and control using RFID tags from remote locations. It also sends needed information from the sensors to central station Pandaboard that has to perform appropriate actions on the complete system. This solution was selected to enable as much as possible system modularization. This way of design also enables easier service, maintenance and diagnostics of the errors in the system. It gives possibility for quick fault detection and replacement or repair of faulty module in event of a malfunction in the system. The system modularity results in higher reliability and greater configurability of the systems. So, it is faster and easier to customize and optimize the system in accordance with needs of user. Scheme of the module and scheme of additional electronics for Arduino/Genuino boards is shown in **Figure 6**.

Electronics for control of high-power consumers was implemented using: relays, Darlington transistor array circuit ULN2001A, resistors, LED diodes. There are

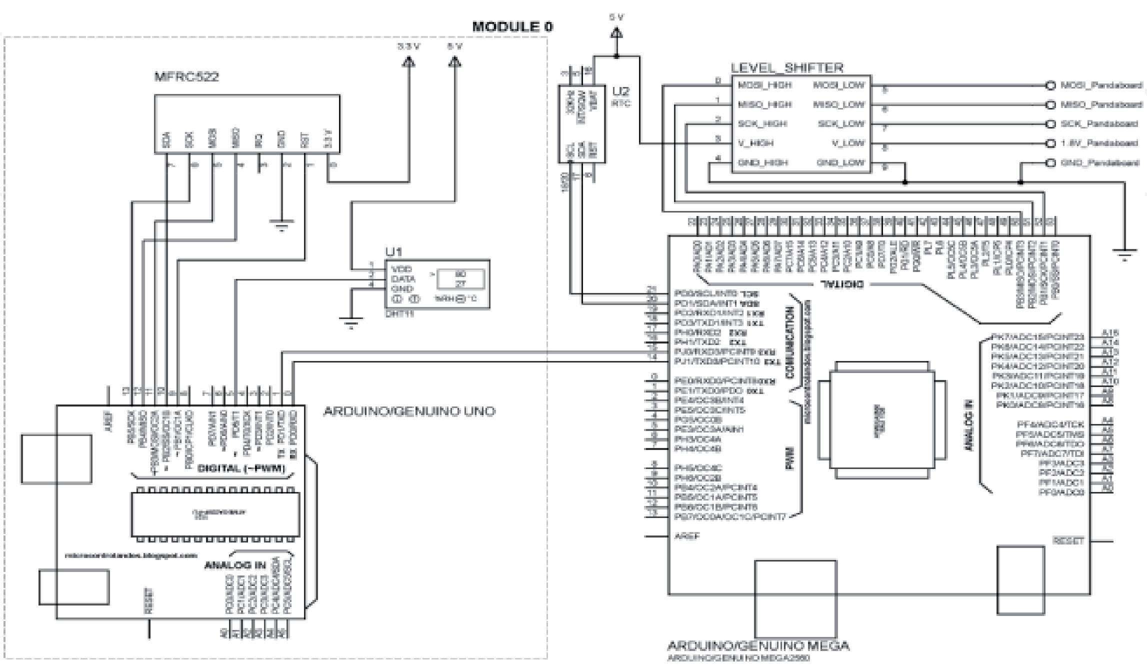


Figure 6.
Scheme of module and additional electronics for Arduino/Genuino boards.

two high-power consumers in the system. The LED diodes indicate state of the high-power consumers, what the high-power consumer is turned on or turned off.

Elements for turning on and turning off high-power consumers do not exist in the Pandaboard. The board also can not provide sufficient current and voltage for turning on and turning off the relays needed for control of high-power consumers. So, it was necessary to design additional electronics for control of the consumers. Integrated circuit ULN2001A with the Darlington coupling of transistors and with protection diodes was used in design of the scheme. The output current that the ULN2001A integrated circuit provides is 500 mA per channel. That is enough to ensure proper operation of the relays and indicator LED diodes. For protection of the system, it is needed that the power supply of electronics for control of high-power consumers be separated from the power supply of the Pandaboard. Such it is prevented that in a case of destruction of electronics for control of high-power consumers also be destroyed the Pandaboard, and vice versa. Designed electronics for control of high-power consumers enables the users to control the consumers with characteristics of AC (0–250) V and (0–10) A or DC (0–30) V and (0–10) A.

For implementation of electronics for control of LCD display were used: voltage level shifter, I2C communication module, LCD display and additional circuits. The voltage level shifter enables communication between Pandaboard and I2C communication module and LCD display that use different power supply voltage levels. The power supply voltage of the Pandaboard is 1.8 V and the power supply voltage for the voltage level shifter, I2C communication module and LCD display is 5 V. The maximal output current of the I2C pin of the Pandaboard is 3 mA. It is sufficient for correct operation of the voltage level shifter. Integrated circuit PCF8574 was used for I2C communication and LCD display control.

Development and design of needed electronics was performed using software Proteus 8 Professional on Microsoft Windows 10 operating system. After the design, it was practically implemented hardware part of the system that was tested. It was performed additional experimentation and testing of the system hardware part in order to decrease the system energy consumption.

The system module is remote station and operates as remote station. Communication of the module with the Arduino/Genuino Mega 2560 board is performed using UART interface. GPIO pins of the Pandaboard are used for communication of Pandaboard and electronics for control of high-power consumers. Additional electronics for Arduino/Genuino boards communicates with Pandaboard using SPI interface.

The Pandaboard is central unit and central station of the implemented system. It was used WiFi router to enable connection to the Pandaboard using mobile phone, tablet, laptop and other computers. Communication of the router and the Pandaboard is performed using Ethernet protocol. So, there are enabled WiFi and Ethernet connection to the implemented system. There is possibility to disable one of the two connections with the aim to increase communication security. WPA2 security protocol is used for protection of WiFi connection.

Development and cross compiling of the system software on the development computer was started after successful testing of the system hardware. The software cross compiling was performed using the software tool arm-linux-gnueabi-*.

Patch and source code of U-Boot bootloader v2015.10 downloaded from the Internet were used for design and implementation of Bootloader. The archived source code of the U-Boot and patch were downloaded from the Internet. In the design and implementation were performed: copying of patch into source code of U-Boot, setting parameters needed for cross compiling of U-Boot and patching of source code, deleting of not needed files and folders generated by previous compiling, cross compiling, copying of cross compiled U-Boot.

Version 4.1.14 of LINUX kernel downloaded from the Internet was used for design and implementation of the system. It was performed: programming of Device Tree file that enables correct mapping of devices during loading of LINUX kernel, setting parameters needed for cross compiling of LINUX kernel and Device Tree file, configuring of LINUX kernel, cross compiling of LINUX kernel and Device Tree file, setting of appropriate privileges to created files and folders.

During design and implementation of ROOTFS of the system first was cross compiled Busybox. It was used Busybox v1.24.1. and source code taken from the Internet. The archived source code was downloaded from the Internet and uploaded to the development computer. It was performed: setting variables needed for cross compiling of Busybox, configuration and cross compiling of Busybox, copying of needed libraries. In design and implementation of basic ROOTFS and configuring of the network were performed: creation of needed folders, creation of console devices, creation of ROOT users, setting of the network, copying of the files. During creation of ROOTFS it was used Dropbear version 2016.73 for SSH communication. It was realized: setting of variables needed for cross compiling of Dropbear, configuration and cross compiling of Dropbear, copying of needed files and folders, enabling automated starting of Dropbear when started operating system of Pandaboard.

SQLite program was used for management of data base. In the design and implementation were realized: setting variables needed for cross compiling of SQLite, configuration and cross compiling of SQLite, copying of needed files. **Figure 7** shows scheme of used data base. Information about users and RFID tags is stored and memorized in the data base. There are records about two types of users, the ordinary user and the administrator. Administrator can add new users and RFID tags, change activity state and update existing users and RFID tags. Every valid RFID identification is recorded in data base and information from sensors in that moment. For every change in data base is recorded time of change.

The Lighttpd, optimized open source HTTP server, was used for design and implementation of HTTP server. In the server design and implementation was realized: setting variables needed for cross compiling of Lighttpd, configuration and cross compiling of Lighttpd, creation of folder and copying of needed files, setting privileges to created folders and files. It was used Libxml2 library for parsing XML (Extensible Markup Language) and needed for cross compiling of PHP (Personal Home Page) used in design and implementation of web page. It was realized: setting variables needed for cross compiling of Libxml2, configuring and cross compiling of Libxml2. The PHP server-side script language was used for design and implementation of Internet web page. It was performed: setting variables needed for cross compiling of PHP, configuration of PHP, cross compiling of PHP, copying of files, creation of folder.

Application for control of high-power consumers was optimized and implemented as ash script to occupy minimum of space in implementation. Possibilities of the application are: turn on or turn off of consumer, monitoring state of consumer. Commands for control of consumers are: consumer turn on, consumer turn off, reading state of consumer.

Application for control of LCD display was also based on ash script. Command for control of LCD display gives possibility to show on the LED display needed string of text characters.

Application for communication with remote stations was written in C language. It enables acquisition of data from remote stations and recording important events into SQLite data base, and also time synchronization obtained from RTC.

It was designed and implemented graphical user interface in the form of Internet web page for monitoring and control of the system from remote locations.

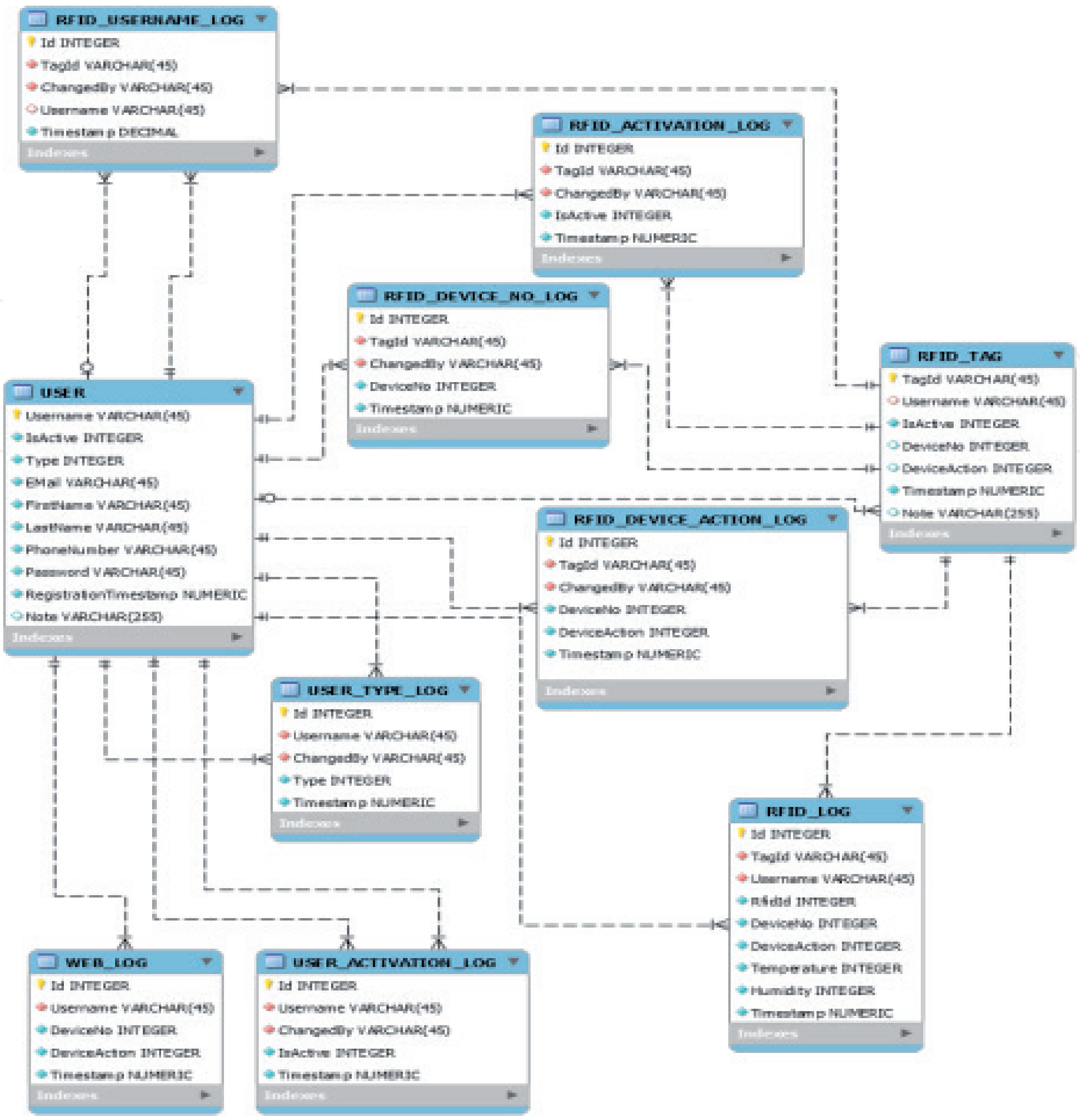


Figure 7.
Scheme of used data base.

For design and implementation were used next technologies: HTML (HyperText Markup Language), CSS (Cascading Style Sheets), PHP, JavaScript, AJAX (Asynchronous JavaScript And XML).

It was created and designed monitoring and control panel named PandaPower. It is needed to enter Username and Password on login page shown in **Figure 8a** during login to the system. It is not stored user Password in the data base but the hash of the user Password. In this way the administrator can not see the user Password and is prevented misuse of the data from the administrator side.

After authorization the PandaPower panel is available for the user. There are panels for ordinary user and for administrator. **Figure 8a** shows panel for ordinary user and **Figure 9a** shows panel for administrator. Ordinary user can use the panel for: Control of consumers and displaying load of SCADA system, Monitoring of consumers, network, processes and basic information of SCADA system, Displaying of the last 100 web logs. Administrator can use the panel for: Control of consumers and displaying load of SCADA system, Monitoring of consumers, network, processes and basic information of SCADA system, Displaying of the last 100 web and RFID logs, Control of RFID tags, Control of users. Every control of consumers, users and RFID tags is recorded in the data base. It is enabled to users to logout from the system after using of the PandaPower panel.

a) b)

Figure 8.
PandaPower login page (a) and panel for ordinary user (b).

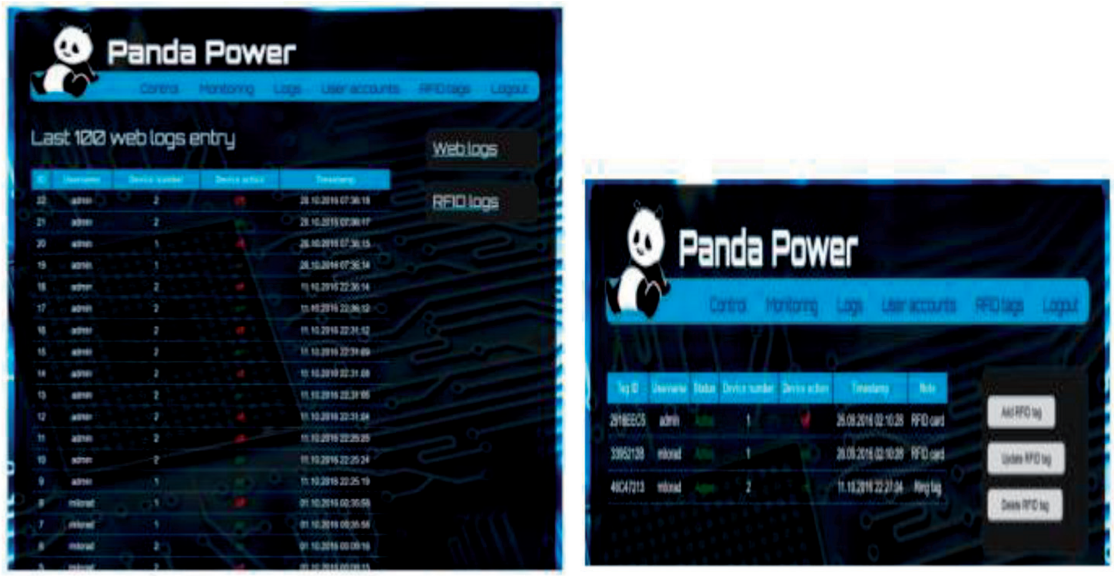
a) b)

Figure 9.
Panel for administrator (a) and administrator monitoring panel (b).

Administrator has possibilities to monitor and control users, consumers, network, process, RFID devices, basic information of the SCADA system. PandaPower administrator panel for monitoring consumers, network, process and basic information of the SCADA system is shown in **Figure 9b**. **Figure 10a** shows PandaPower administrator log panel that enables list of last 100 web and RFID system logs. **Figure 10b** shows administrator RFID panel for control of RFID tags.

Due to the support of Pandaboard, it was used SD memory card for storage of the data. SD memory card has appropriate performances and low cost. Formatting of SD card was performed on development computer using program fdisk on GNU/Linux Mint 17.2 operating system. Two partitions with names BOOT and ROOTFS were created on the SD card. Then were copied LINUX kernel, Device Tree, U-Boot and ROOTFS with all needed applications to the SD card.

It was also needed to program Atmel microcontrollers in the Arduino/Genuino boards. The Arduino IDE environment was used for the programming [24].



a) b)

Figure 10.
Administrator log panel (a) and administrator RFID panel (b).

Proposed and described embedded system is relatively complex but inexpensive. With such way of design the time needed for the system design and implementation, and system cost were significantly decreased.

6. Conclusions

Development of digital electronic technologies enables design and application of embedded systems for different needs and applications. Such systems are maximally adapted to concrete application and are specialized and optimized for a specific purpose. Different hardware and software platforms can be used for design and implementation of such systems. Specially designed platforms are the most often used for such purposes. But, in many practical needs it is much more convenient to use open source hardware and software platforms.

It is proposed here to use open source hardware and software microcomputer-based technologies for design and implementation of embedded systems for many practical applications. As illustration of such approach and its benefits, two such practically designed and implemented embedded systems were presented and described. Open source microcomputer boards, appropriate sensors, actuators and additional electronics were used for the systems hardware implementation. Open source software tools and programs and LINUX operating system were used for the systems software implementation. Modular approach that enabled easy expandability, simplifying maintenance and adaptation of the system to user needs was also used in the system design. It was also achieved balance between functionality and cost of the systems.

It was performed optimization in selection of all used components and tools, and optimization of all hardware and software designs, according to system needs, in the design and implementation of this embedded systems. Such it is achieved balance of functionalities and cost of the system. Main advantages of this embedded systems compared with other ways of design and realization are: lower cost, optimization according to application, smaller dimensions and weight, and lower electrical energy consumption. Some disadvantages are: lower reliability, lower certainty, constrained application on need for what the system was designed.

The Arduino hardware platform was chosen for the systems realization because that platform is easy available, very good documented, with many development tools, very cheap and can satisfy all needs of the systems concrete application with the small system cost. It is especially case with the system for user identification using smartphone that is relatively simple system. The Linux operating system was used in implementation of SCADA and RFID system since that system is more complex and not requiring high speed of operation. So, LINUX operating system was chosen because it enables faster and simpler design of system software, shorter system development time and smaller cost of the system.

Embedded systems based on open source platforms are not standard general purpose solutions or totally specially designed solutions. Such systems are less expensive, specialized and optimized solutions for some concrete user need. Such design and implementation should be used mainly for systems of smaller or medium complexity and where is needed smaller number of the systems. That approach can also be used for design and implementation of more complex and larger embedded systems. But, then it will be needed much more activities and time for system design and implementation and systems will be more expensive.

Author details

Zlatko Bundalo[1]* and Dusanka Bundalo[2]

1 Faculty of Electrical Engineering, University of Banja Luka, Banja Luka, Bosnia and Herzegovina

2 Faculty of Philosophy, University of Banja Luka, Banja Luka, Bosnia and Herzegovina

*Address all correspondence to: zlatbun2007@gmail.com

References

[1] Wolf W. Computers as Components: Principles of Embedded Computing Systems Design. San Francisco, USA: Morgan Kaufman; 2008

[2] Gajski D, Abdi S, Gerstlauer A, Schirner G. Embedded System Design: Modeling, Synthesis and Verification. Heidelberg, Germany: Springer; 2009

[3] Valvano J. Embedded microcomputer systems: Real-time interfacing. Boston; USA: Cengage Learning. 2011

[4] White E. Making Embedded Systems: Design Patterns for Great Software. Boston, USA: O'Reilly Media; 2011

[5] Bundalo Z, Ristić N, Bundalo D, Pašalić D, Cvijić B. Possibilities of using mobile phone for user identification. In: Proceedings of International Scientific Conference ERK2016; Portorose, Slovenia. 2016. pp. A:39-A:42

[6] Bundalo Z, Ristić N, Bundalo D, Pašalić D, Cvijić B. Possibilities of using mobile phone for user identification. ANNALS of Faculty Engineering Hunedoara—International Journal of Engineering. 2018;**16**(2):185-188

[7] Pašalić D, Cvijić B, Bundalo D, Bundalo Z, Kuzmić G. Embedded systems for user identification in access to objects and services using mobile phone. In: Proceedings of International Conference MECO2017; Bar, Montenegro. 2017. pp. 109-112

[8] Zulić M, Bundalo Z, Bundalo D. Microcontroller based embedded system for garage access control by mobile phone. In: Proceedings of International Scientific Conference ERK2017; Portorose, Slovenia. 2017. pp. 19-22

[9] Papić M, Bundalo Z, Bundalo D, Stojanović R, Kovačević Ž, Pašalić D, et al. Microcomputer based embedded SCADA and RFID systems implemented on LINUX platform. In: Proceedings of International Conference MECO2017; Bar, Montenegro. 2017. pp. 104-108

[10] Papić M, Bundalo Z, Bundalo D, Stojanović R, Kovačević Ž, Pašalić D, et al. Microcomputer based embedded SCADA and RFID systems implemented on LINUX platform. In: Microprocessors and Microsystems. Vol. 63. Amsterdam, Netherlands: Elsevier; 2018. pp. 116-127

[11] NIST Computer Security Handbook. 2008. Available from: http://www.sos.cs.ru.nl/applications/courses/security2008/nistiadraft.pdf

[12] Official Web site of GSMA. 2018. Available from: http://www.gsma.com/managedservices/mobile-equipment-identity/about-imei/

[13] Singh M, Jain M. Evolution of processor architecture in mobile phones. International Journal of Computer Applications. 2014;**90**(4):34-39

[14] Townsend K, Cufí C, Davidson R. Getting Started with Bluetooth Low Energy. Boston, USA: O'Reilly Media; 2014

[15] Igoe T, Coleman D, Jepson B. Beginning NFC: Near Field Communication with Arduino, Android, and PhoneGap. Boston, USA: O'Reilly Media; 2014

[16] Schmidt M. Arduino A Quick Start Guide. Dallas, USA: Pragmatic; 2011

[17] Blum J. Exploring Arduino: Tools and Techniques for Engineering Wizardry. Indianapolis, USA: John Wiley & Sons; 2013

[18] Annuzzi J, Darcey L, Conder S. Introduction to Android Application Development. Boston, USA: Addison-Wesley; 2013

[19] Bailey D, Wright E. Practical SCADA for Industry. Amsterdam, Netherlands: Newnes and Elsevier; 2003

[20] Knapp E. Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems. Amsterdam, Netherlands: Elsevier; 2011

[21] Finkenzeller K. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication. Indianapolis, USA: Wiley & Sons; 2010

[22] Prabhu N. Design and Construction of an RFID-Enabled Infrastructure—The Next Avatar of the Internet. Boca Raton, USA: CRC Press; 2014

[23] OMAP4460 Pandaboard ES System Reference Manual. Available from: https://www.cs.utexas.edu/~simon/378/resources/PandaBoardES.pdf

[24] Web site of Arduino. Available from: http://www.arduino.cc/

[25] Simmonds C. Mastering Embedded Linux Programming. Birmingham, United Kingdom: Packt Publishing; 2015