Technical University of Košice
Faculty of Electrical Engineering and Informatics

Design and implementation of a portable device for creating RFID tag lists

Bachelor's Thesis

2014 Peter Babič

Technical University of Košice
Faculty of Electrical Engineering and Informatics

Design and implementation of a portable device for creating RFID tag lists

Bachelor's Thesis

Study Programme:	Applied informatics
Field of study:	9.2.9 Applied Informatics
Department:	Department of Theoretical and Industrial Electrical Engineering (KTPE)
Supervisor:	Ing. Tibor Vince, PhD.
Consultant(s):	Ing. Norbert Ádám, PhD.

Košice 2014 **Peter Babič**

Errata

Design and implementation of a portable device for creating RFID tag lists

Peter Babič

Košice 2014

Page	Line	Wrong	Correct

Abstract

This thesis shows the process of building a portable hand-held device capable of storing RFID tags along with current date and time. The process goes from the idea on the paper, through the prototyping and testing stage up to the working device. The device is backed by a microcontroller and is able to read all Mifare compatible RFID cards and identificators, preferably student's ISIC cards, store them on the MicroSD card, in the CSV format, while providing an user interface for basic operation and a Serial connection over an USB cable for advanced operation.

Keywords

ISIC, microcontroller, MicroSD, Mifare, RFID, tags

Abstrakt

Táto práca ukazuje proces tvorby prenosného zariadenia, schopného ukladať RFID tagy spolu s aktuálnym časom a dátumom. Proces začína teoretickým rozborom, tiahne sa cez prototypovaciu a testovaciu časť až po fungujúce zariadenie. Zariadenie obsahuje mikrokontrolér a je kompatibilné so všetkými typmi Mifare RFID kárt a identifikátorov, avšak hlavným cieľom sú študentské ISIC karty. Dáta sú uložené na MicroSD kartu, v CSV formáte. K dispozícii je minimálne užívateľské rozhranie pre základne použitie a Sériové spojenie cez USB kábel pre rozšírené operácie.

Kľúčové slová

ISIC, mikrokontrolér, MicroSD, Mifare, RFID, tagy

TECHNICAL UNIVERSITY OF KOŠICE
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS
Department of Computers and Informatics

BACHELOR THESIS ASSIGNMENT

Field of study: **9.2.9 Applied Informatics**
Study programme: **Applied Informatics**

Thesis title:

Design and implementation of portable device for creating RFID tag lists
Návrh a realizácia prenosného zariadenia pre vytváranie zoznamu RFID tagov

Student: **Peter Babič**

Supervisor:	**Ing. Tibor Vince, PhD.**
Supervising department:	**Department of Theoretical and Industrial Electrical Engineering**
Consultant:	**Ing. Norbert Ádám, PhD.**
Consultant`s affiliation:	**Department of Computers and Informatics**

Thesis preparation instructions:

1. Describe general information about RFID technology and microcontrollers
2. Design and creation of PCB
3. Design and microcontroller program implementation according to supervisor instructions

Language of the thesis:	English
Thesis submission deadline:	30.05.2014
Assigned on:	31.10.2013

.... doc. Ing. Jaroslav Porubän, PhD.
Head of the Department

.... prof. Ing. Liberios Vokorokos, PhD.
Dean of the Faculty

I hereby declare that this thesis is my own work and effort. Where other sources of information have been used, they have been acknowledged.

Košice, May 30, 2014 . . . .

Signature

Acknowledgement

I would like to express my sincere thanks to my supervisor Ing. Tibor Vince, PhD., the main Supervisor, for his constant, and constructive guidance. Another mention should go Ing. Norbert Ádám, PhD. To all other who gave a hand, I say thank you very much.

Preface

This thesis is about constructing an electronic device from a scratch. It requires problem solving skills, analytical thinking and decision making. Among core technical skills used are electronics design, firmware programming and product documentation, which are all part of the product life-cycle. Practising and improving named skills was the main motivation to choose this topic.

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

μ micro, 10^{-6}

CPU Central Processing Unit

Hz Hertz, unit of frequency

I/O Input/output

ISIC International Student Identity Card

LCD Lyquid Crystal Display

MCU Microcontroller, sometimes uC or μC

microSD Secure Digital memory card

RAM Random-access Memory

RFID Radio Frequency Identification

ROM Read-only memory

RTC Real-time Clock

SI Système International

TI Texas Instruments

USB Universal Serial Bus

V Volt, basic unit in SI

List of Terms

File system is used in computing to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. Common file systems are: FAT, NTFS, EXT and many more.

Liquid-crystal Display is a flat display using light modulating properties of liquid crystals. They do not emit light themselves. LCD's are used in a wide range of applications, and are energy efficient.

Microcontroller is a compact microcomputer designed to govern the operation of embedded systems in motor vehicles, robots, office machines, complex medical devices, mobile radio transceivers, vending machines, home appliances, and various other devices. A typical microcontroller includes a processor, memory, and peripherals.

RFID radio frequency identification: a technology that uses electronic tags placed on objects, people, or animals to relay identifying information to an electronic reader by means of radio waves.

SD card is a non-volatile memory card format for use in portable devices. It is one of the most used memory card formats today. It divides to three categories by physical size: SD, MiniSD and MicroSD. There are also three categories, regarding it's capacity: SDSC, SDHC and SDXC.

Universal Serial Bus is an industry standard developed in the mid-1990s that defines the cables, connectors and communications protocols used in a bus for connection, communication, and power supply between computers and electronic devices.

Introduction

The idea is to construct a battery powered device that will be able to read the RFID tag from the ISIC cards students briefly place onto the reader part. The tags will then be stored on the detachable microSD card in structured format, alongside with time and date. The end result of this process is actually a attendance list of students. The device is to be used in the environment, where there are limited possibilities do proper attendance list, i.e. in the gym or when working in terrain.

Basic concept is to design and realize an electronic circuit, which will handle RFID communication with the student's ISIC card.

The project itself basically consists of this main parts:

- the microcontroller
- the RFID reader module
- the power source
- a display
- the storage memory for the lists
- a way to communicate with a PC
- a real-time tracking module
- some user interface

The two absolutely crucial areas that needs to described in deeper detail this project are microcontrollers and of course the RFID.

1 Microcontroller

A microcontroller is a self-contained system with peripherals, memory and a processor that can be used as an embedded system (it's brains). Most programmable microcontrollers that are used today are embedded in other consumer products or machinery including phones, peripherals, automobiles and household appliances for computer systems. Due to that, another name for a microcontroller is "embedded controller."

Microcontroller can only manipulate with limited amount of data at once. Most today microcontrollers are working with 8, 16 or 32 bits.

1.1 Brief history of microprocessors and microcontrollers

Since microcontrollers evolved out of microprocessors, some milestones in their history are worth noting[1][3]:

- Busicom - calculator manufacturer of Japan ordered design of some specific chips from Intel corporation (1969)
- M. E. Hoff came out with the idea of device, which operation will be governed by instructions stored in ROM memory
- F. Faggin built the actual device - the first microprocessor
- Intel bought rights to sell the device on their own from Busicom, Intel 4004 (4 bit) came to market (1971)
- Texas Instruments engineers G. Boone and M. Cohran packed CPU, RAM, ROM, I/O and peripherals to the one chip, the first microcontroller was born, but it was a commercial failure

- Intel improved the device, the Intel 8008 (8 bit) was born (1972)
- TI microcontroller was employed in calculators (1972)
- TI put their TMS1000 microcontroller on the market with a great success, for $2/pc, millions were sold (1974)
- Motorola entered the market with 68000 and Zilog with Z80, which caused the competition to grow exponentially

1.2 Microcontroller architectures and vendors

Microcontrollers are produced by roughly two dozens of companies, so the great diversity exists. Advanced integration is causing that more and more functionality is packed into these devices.

Some of the present MCU architectures and vendors are:

- 8051 introduced by Intel, today produced by most semiconductor manufacturers
- ARM
- MIPS
- AVR by Atmel
- PIC by Microchip
- DSP430, C2000 by Texas Instruments
- Freescale, ST Microelectronics, XMOS, ...

1.3 Structure of microcontroller

On the very basic level, the microcontroller consists of basic building blocks, illustrated on Figure 1 – 1:

- Microprocessor with Arithmetic-logic Unit (ALU), Control logic and Instruction decoder
- Memory
- Program counter
- Oscillator
- Input/Output circuitry
- Timers, Pulse-Width Modulation
- Analog-to-digital converter

Some additional blocks and peripherals are be included, depending on MCU model, such as:

- Brown-out detection
- Serial communication interfaces[13]
- USB controller
- Ethernet interface
- Watch-dog timer
- Interrupt routines

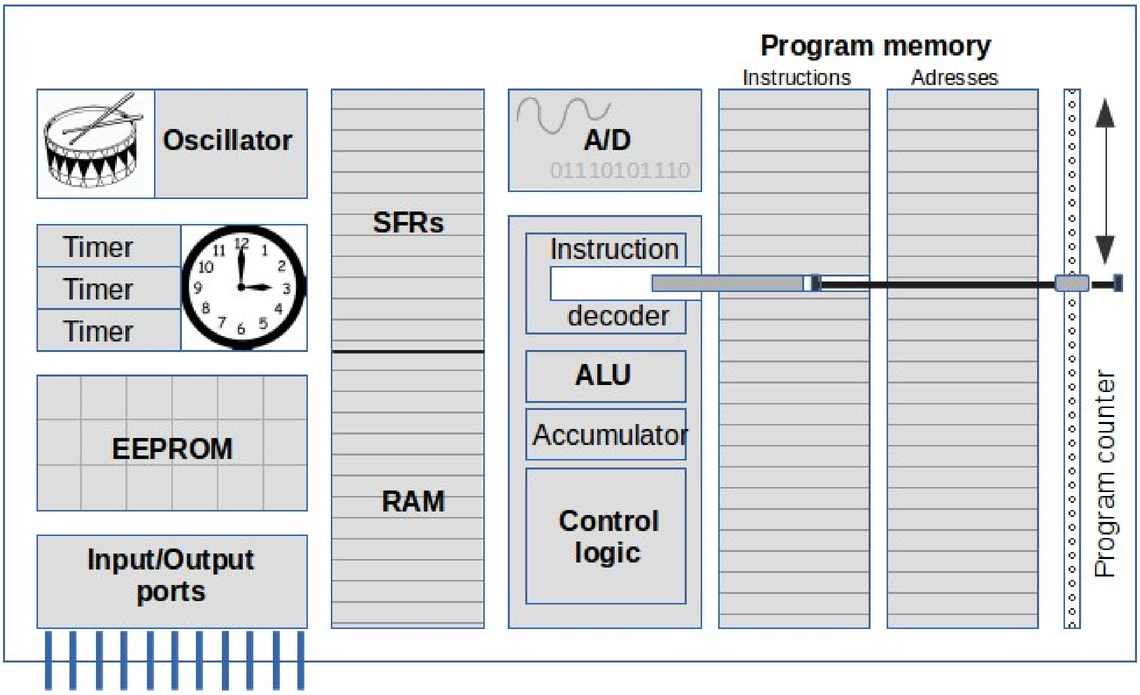

Figure 1 − 1 Naive representation of what is inside the microcontroller

Every block can get really complex to explain. For simplicity, only the blocks, that are particularly important for the understanding the device will be described in deer detail.

1.4 Microcontroller memories

Most microcontrollers do possess three types of memory:

- Flash stores instructions in order they will be executed
- RAM stores temporary data, needed for computations
- EEPROM stores data, that should be remembered, such as user configurations

Special-function Registers (SFRs) are part of RAM memory and changing them, will cause direct change in microcontroller behaviour.

1.5 Oscillator

Oscillator's job is to make sure, every block knows, when to do its job. It represents the function of commander, thus the drums illustration. There are two types of oscillators used:

- external
- internal

1.6 Serial communication interfaces

Communication interfaces are used for communication between MCU and other MCU or other devices. Most standard serial interfaces are UART, I2C and SPI.

UART is used between two devices. When more devices are communicating, I2C or SPI is used. SPI provides faster communication over I2C at the expense of used pins on microcontroller.

1.7 USB controller

A USB port[4] is a standard cable connection interface on personal computers and consumer electronics. USB ports allow stand-alone electronic devices to be connected via cables to a computer (or to each other).

USB is serial communication interface, derived off I2C. It can be implemented on microcontroller itself, or as standalone device/module.

1.8 Other useful information

Bluetooth is a wireless technology, for communication between devices and/or computers at short ranges. Display is the device showing visual information. Human interface provides a way for human to control the device. Non-volatile memory is preserving the data, even when power is lost.

2 RFID

RFID stands for Radio-Frequency IDentification [9]. The acronym refers to small electronic devices that consist of a small chip and an antenna. The chip typically is capable of carrying 2,000 bytes of data or less.

The RFID device serves the same purpose as a bar code or a magnetic strip on the back of a credit card or ATM card; it provides a unique identifier for that object. And, just as a bar code or magnetic strip must be scanned to get the information, the RFID device must be scanned to retrieve the identifying information.

From the energy point of view, RFID tags falls into one of the three categories:

- active
- battery-assisted passive
- passive

2.1 RFID frequencies

RFID is considered as a non specific short range device. It can use frequency bands without a license. Nevertheless, RFID has to be compliant with local regulations.

Frequencies used in RFID communication, illustrated on Figure 2 – 1:

- LF - low frequencies: 125 kHz - 134,2 kHz
- HF - high frequencies: 13.56 MHz
- UHF - ultra high frequencies: 860 MHz
- SHF - super high frequencies: 2.45 GHz

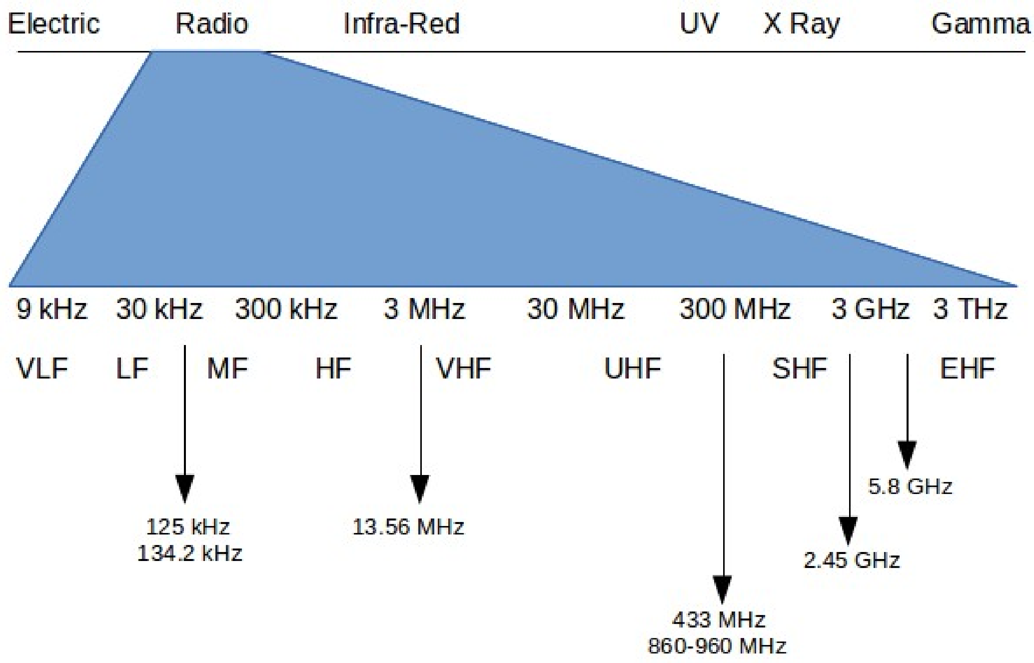

Figure 2 – 1 Graph of the electromagnetic spectrum with the frequency bands that the RFID systems can use

2.2 Mifare

Mifare is a 13.56 MHz contactless NFC[5] technology owned by NXP (founded by Philips). It falls into high frequency, passive RFID tag category. Mifare has proven popular as the card to use in ticketing, low value transactions and in student's ISIC cards. Mifare cards are powered by an internal antenna as the card moves into a magnetic field.

Cipurse is another such technology, although it is not owned by just one company, but rather alliance of companies. The next difference is that this standard is open. It's aim is to promote vendor neutrality, opposed to market Mifare dominance.

Some benefits of the Mifare are:

- multiple applications

- secure encryption prevents unauthorised access
- reduced maintenance, due to contactless technology
- alternatives to cards, such as fobs or watches
- only proximity is sufficient for read/write
- unique serial number (tag)

2.3 RFID reader

An RFID reader is basically a radio frequency transmitter and receiver controlled by a microprocessor. The reader, using an attached antenna, captures data from RFID enabled smart label tags, then passes the data to a microcontroller for processing.

3 Portable device requirements

The duties of the device are:

- show information on the display
- receive commands from user interface
- maintain real time and date
- obtain tag information from RFID reader module
- store tag information along with time and date in a memory
- process communication with PC

3.1 Displayed information

The information being shown to the user:

- user menu
- actual time/date
- actual set meeting
- status information

3.2 User menu entries

- set time/date
- create new meeting file with generic name

- delete meeting file
- set current meeting file to save into

3.3 PC communication data set

The device should be able to recognize and execute following commands on meeting files given from PC:

- show all
- delete
- create
- rename
- download

4 Hardware dependencies

Things that needs to be considered about components/modules, when designing this kind of device are:

- power consumption
- cost
- availability

4.1 Viable microcontroller

Microcontroller has to synchronize all the device duties, discussed earlier. These task are simple enough to be built around 8-bit microcontroller, which reduces cost and power consumption. Most viable options are:

- AVR ATmega168/ATmega328 has tons of available source codes
- AVR ATmega32u2 has built in USB controller

4.2 Display and user interface

Display has to have low power consumption. Apart from that, it should have at least two lines to be able to show enough information.

Touchscreen is option that combines a display and an user interface. If push-buttons are used as a human interface, the display is needed. Viable display types are LCD, oLED and e-ink.

4.3 Keeping track of time and date

Real-time can be tracked by either a MCU or using external RTC module. RTC module has own battery supply, the algorithms for time and date built in and also provides some additional nvRAM for storing data that should not get lost, but microcontroller usually has some EEPROM memory for this purpose, which will be much faster.

4.4 RFID reader module

Reader has to be able to decode Mifare tag information. The most obvious reason is to use MFRC522 Contactless reader IC controller from NXP Semiconductor, because it does all the hard job. It is quite expensive alone, but you can get it from the China already soldered it as a module with driving electronics, and what is more important, an inductor working as an antenna, for the price for the IC alone! Choosing this module solves a lot of problems.

4.5 Memory for storing lists

Lists of information should be stored in some form of electronically programmable non-volatile memory, in basic structured format. Common types are:

- Flash (internal or as a microSD card)
- EEPROM

MicroSD card needs its own special compartment, where it is inserted and makes electrical contact. User should be able to replace the microSD card within the device.

4.6 Communication with PC

Communication between the device and the PC has to be fast enough to provide quick way of downloading the lists stored in device memory. This can be accomplished with the use of USB. Using Bluetooth can be another or additional option for this purpose.

4.7 A power source

When considering a power source, a lot of options comes to play. First choice:

- replaceable power source
- non-replaceable power source

Replaceable source could be standard rechargeable or non-rechargeable cell batteries of two 1.5V batteries or one 9V battery. Device will need stand-alone battery compartment for battery maintenance.

For non-replaceable power source, combination of these could be implemented:

- solar panel
- Li-Po rechargeable battery
- supercapacitor
- wireless charging through RFID antenna

4.8 Firmware dependencies

For the firmware there are again two options: program it from scratch and use some platform. The first option is viable mainly to applications which have limitations such as RAM or Flash memory space, or where timing is crucial. Since these are not the case here, because we are limited mainly with the amount of power stored in battery, a framework/platform can be used to greatly aid in firmware development.

5 Contactless Attendance Logger

The Contactless Attendance Logger is a device constructed as a result of this work. The requirements for this device was detailed in sections 3 and 4. This section explains, how and why individual components, depicted in those sections were chosen. The fully assembled device, inside a case is shown at 5 − 1, with part labelled at 5 − 3.

Figure 5 − 1 Full assembled device - Contactless Attendance Logger

5.1 Firmware

For the firmware[14] side, the Arduino platform[11] will be used. The reason behind this decision is global community support and free software libraries for almost any possible component including most of the components suggested for this device. This will greatly reduce the software development time over plain C programming.

5.2 Power supplies and battery level

Four 1.5V AA type batteries in series will produce slightly more than 6V when fully charged. Together with low drop-out linear regulator LP2985-33[20] will provide steady 3.3V over full operable batteries voltage range. This setup will eliminate the need to incorporate a DC converter, which will make the design far simpler.

The device will be operable also without the main power supply, the batteries. Linear Technology LTC4412[21] chip will take care of switching between battery and USB supply, choosing the USB when provided. Two external diodes will provide further protection. To preserve battery voltage range, the low drop-out Schottky diode BAT42 is used on the batteries side.

The power supply part of the device is able to provide information about which power supply is active at given time, however it is not able to determine the current battery charge level. For this purpose small custom circuitry will be made to handle this task and to protect the MCU from over-voltage, since it cannot measure voltage level of the batteries at their full potential.

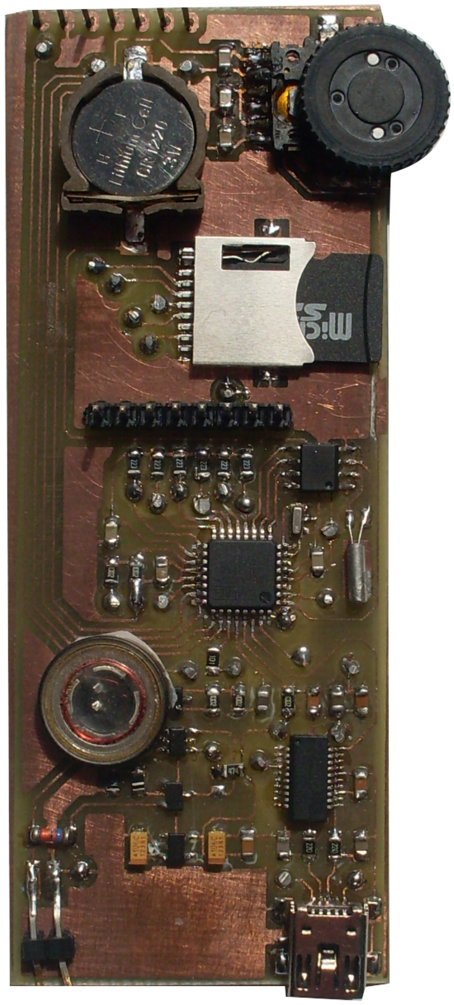

Figure 5 – 2 Disassembled device showing the main board, exposing components

5.3 Microcontroller

As a brain for the device is chosen the Atmel microcontroller, ATmega328P[17]. This MCU is fully compatible with the Arduino platform, so no further modifications are needed. It has 32kB of flash program memory, which gives us a reasonable amount of memory reserve, since estimations shows, that around 20kB of program memory space will be used.

5.4 Communication

The FTDI device FT321X[19] will be used for providing communication between microcontroller and the PC over an USB connection. It communicates with a MCU over an UART protocol, which needs precise clock source on both sides. This device has it's own precise clock already integrated. This chip is also cheaper, smaller and more reliable than a lot of other similar solutions on the market.

Since the device will use a fully compliant USB-to-UART bridge, the user will be able to reprogram the microcontroller over the USB without the need of any external device. This will help introducing further firmware updated to the end device. The USB side of the bridge will appear on host computer as Virtual COM Port[4].

5.5 Clock source

The microcontroller will use small compact 16MHz resonator with integrated run-up capacitors with 0.5% accuracy. Since the MCU cannot reliably operate clocked to 16MHz at 3.3V, clock prescaler, integrated inside the MCU, with a factor of 2 will be used to drop the clock frequency to 8MHz. This frequency is fully compatible with the Arduino platform, so no further code modifications will be needed. Lower frequency also help to preserve the batteries. The external clock source will reduce the errors over the UART communication.

5.6 Display

User will be able to read the information and menu from the chosen 24x48 dot matrix LCD with PCD8544 controller, known for presence in some popular Nokia mobile phones like 3310 or 5110 models. The reasons for this was really low power consumption, communication in SPI protocol, price and ability to easily display simple images, for example a battery level indication. Since the display has its own controller, if there is no actual need for updating the displayed data, microcontroller can go to sleep mode to preserve the battery energy. This is not possible in displays directly controlled by MCU, i.e. multiplexed 7-segment displays.

5.7 User interface

The minimalist design of this device was requiring at least 3 user inputs. For this purpose, re branded rotary encoder from old Sony Ericsson mobile phone is used. Now the device can be controlled with only one finger or thumb. The encoder will provide two rotation directions and also acts as a push-button. This will help scrolling the device menu.

Audio feedback for used will be handled by small speaker used commonly in hand-held devices to provide an audible feedback for user, indicating some events, i.e. successful card read operation.

5.8 Time and date tracking

For this task, the Maxim DS1302[18] chip is used. The precise clock source, watch crystal with 32.768kHz, that is fed directly into the chip. The chip is also backed by a small replaceable non-rechargeable 3V battery CR1220. This will make sure that the device will be fully operable without main battery, only with the USB connection active.

5.9 RFID reader module

The suggested RFID reader module from China manufacturers, with NXP's chip MFRC522 will be used for RFID communication. It handles all the Mifare protocol communication and also is able to communicate over SPI protocol. This plays nicely with the LCD display, since it is also communicating over SPI.

5.10 MicroSD card slot

MicroSD card will be pushed into standard MicroSD card slot. No module will be used, since they provide no real advantage and take precious space in hand-held devices. The card also communicates with MCU over the SPI protocol, which needs some surrounding electronics shared with other SPI related components.

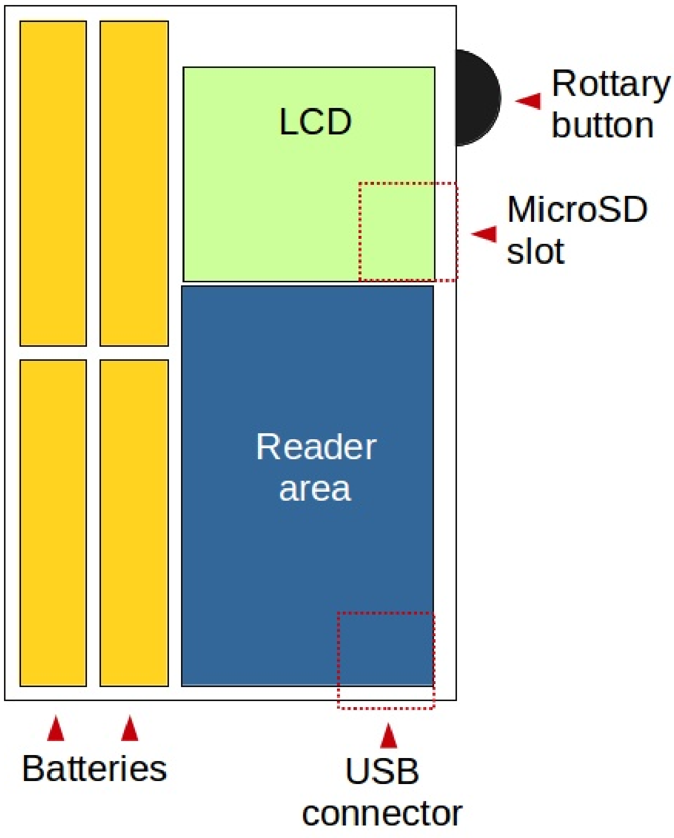

Figure 5 – 3 Illustration of the fully assembled device with labelled individual parts

6 User interface operation instructions

This section is devoted to instructions for proper operation of the device. The manual and all other information provided by the device is available only in English language.

6.1 MicroSD card preparation and usage

The device supports MicroSD cards up to 2 GB of storage space. The card itself has to be formatted with the FAT16 filesystem[12]. Consult your operating system help center for instructions. The device itself cannot do this task alone. Properly prepared card is then inserted with a gentle push into MicroSD slot near the top right corner of the device. This operation is possible only when the device is out of its case, since there is no dedicated hole in the case. Pull the card firmly to take it out of the device. When the card is inserted after the device was powered, or the card is broken or not supported, the CARD INIT ERROR6 − 1 error is displayed. When

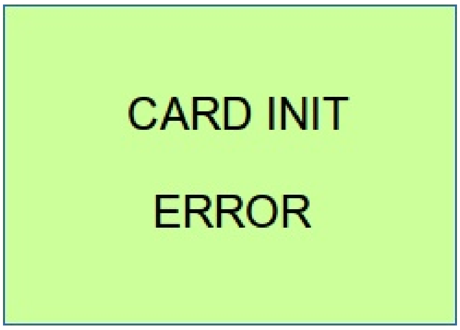

Figure 6 − 1 Illustration of the LCD display, when the device is unable to recognize the MicroSD card

this happens, please ensure that the card is fully inserted inside it's slot, power-down and power-up the device again, or try another card, if the error persists.

If the card is supported, but is formatted unproperly, the FAT16 INIT ERROR 6 − 2 is displayed. Without the working MicroSD card, the device will not be operable.

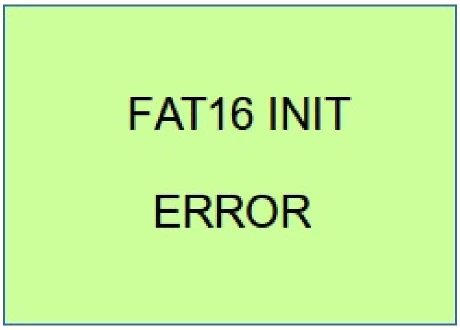

Figure 6 − 2 Illustration of the LCD display, when the MicroSD card is recognized, but is not formatted with FAT16 file system

6.2 Basic operation

The device starts itself as soon, as the power source is present. When inserted into the dedicated plastic case with no USB cable attached, the device is powered from the four contained AA batteries. This fact, along with the current battery level is indicated in the top left corner of the LCD display, in the power source indicator area. With battery mode active, the LCD display looks similar to illustration 6 – 3. Other information displayed on the LCD are current time in the right-top corner and actual storage file. The storage files are explained in advanced operation. The

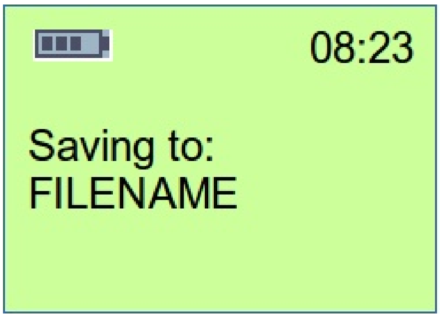

Figure 6 – 3 Illustration of the LCD display, when the device is in battery power supply mode

indicator has four stages, indicating battery level of circa 100, 75, 50, 25%. When the batteries re too too low for device to operate, it will not start, nor display anything on the LCD.

For the operation in the USB mode, connect the Mini-B USB connector to he bottom-right corner or the device. Connect the other side to some USB charger or PC. When connected to the PC, device acts as a Virtual COM Port, that is supported by most OS. You should be able to see it in your operating system device manager. Note that on Windows machines, it may take a while for device driver to install itself via Windows Update service. If this is not the case, you can use included CD to install the driver manually.

When the device is plugged into USB power source, it will immediately switch to the USB mode. This mode does not drain the battery and is indicated with a small USB standard icon in the top-left corner of the display, as illustrated on 6 – 4.

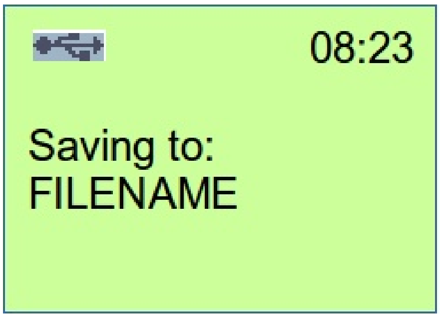

Figure 6 – 4 Illustration of the LCD display, when the device is in USB power supply mode

6.3 Menu and advanced operation

A main menu is accessed by push of the rotary button, which is the only manually operable element. The button is situated in the top-right corner of the device. When in menu, navigation is achieved by a rotation of the rotary button. The current selected menu entry is indicated by the black arrow on the left side of the display. The menu entry is selected by one more push of the button. The main menu itself is simple and contains only 4 entries 6 – 5: Storage, New file, Date/time and an Exit for return to the main screen, when the device is active. This means, that the device records RFID data and responds to Serial commands. Both this concepts will be explained later.

When for some reason, i.e. the MicroSD card is empty or the selected storage file was deleted, the "Please select a storage file" sign 6 – 6 will be sown on the display.

The purpose of the storage menu entry is to choose the file, where the data will be stored, the storage file. Navigation in this submenu is identical to the navigation of the of the main menu. When the file is selected, the device returns to the main screen.

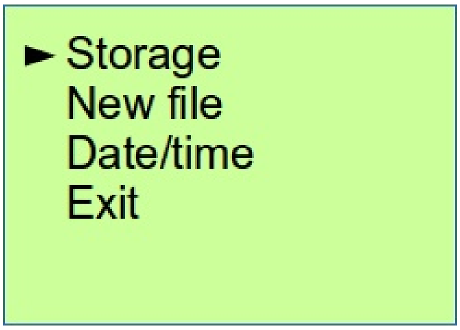

Figure 6 – 5 Illustration of the LCD display, the main menu is active

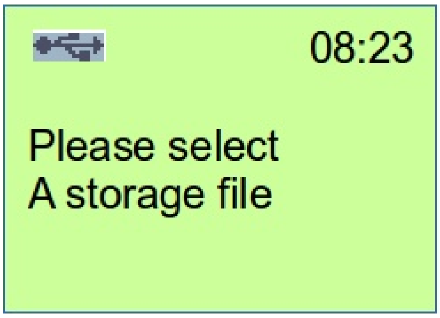

Figure 6 – 6 Illustration of the LCD display, the storage file is not selected

If again, the card is empty, the storage submenu will be empty too. This can be solved by accessing the second entry of the main menu - New file. The entry has no visible submenu and will return to the main screen when accessed. However, the empty storage file will be created. The file naming scheme is FILE1, FILE2, FILE3 and so on. The created file will then be shown on the Storage submenu, where can be selected and activated.

The third entry in the main menu is Date/time. When accessed, the current time and date is shown and can be changed. The actual changed value is indicated by continual blinking. The values are accessed by successive pushes of the button and can be changed by rotation of the rotary button. When the button is pushed on the last, year value, the menu is left to the main screen and the changes to date and time are saved.

6.4 Data recording and Serial commands

The device is active when there is no error or sign displayed and is out of menu, on the main screen - illustrated previously on $6-3$ and $6-4$. The screen contains the sign "Saving to:" and the storage file name beneath the sign. Now, when an RFID active ISIC card is resent in the blue device reader area, the actual date, time and the card's RFID tag is stored on inside the active storage file on the MicroSD card. Success is signalized by a short beep sound. Note that copy of the data that are stored are also sent over the USB cable, through the serial connection.

Serial connection in established by opening a serial terminal application i.e. HyperTerminal, PuTTy or similar one on the PC, to which the device is connected by USB. After choosing the correct port assigned to the device, the baud rate of 38400 is selected. Flow control should be turned off. The other parameters should be left in default state. This includes 8 bit mode, no parity bit and 1 stop bit. The notation for this configuration is 8N1. Consult the application manual for correct settings.

When the serial connection is active, the more advanced commands become available. The commands are typed into the terminal. No special characters are involved. If the command has one or more parameters, they are separated a by space character. When the inserted command is not recognized by the device, the brief help, containing list of possible commands is shown. For easier understanding, the command's arguments are CAPITALIZED. For convention, commands has similar structure as commands used in a Linux/UNIX environment. If the command has one or more arguments, they are all required. Note that file names must be 8 or less characters long. Longer file names may not work. The list of possible commands:

- `ls` - list all storage files
- `touch FILE` - create a storage file named FILE

- `rm FILE` - remove a storage file named FILE
- `rmall` - remove all storage files
- `mv FILE1 FILE2` - rename storage file FILE1 to FILE2
- `read FILE` - read the contents of the storage file FILE
- `readall` - read the contents of all storage files
- `date D M Y` - set current date to DD:MM:YYYY
- `time H I S` - set current time to HH:II:SS

6.5 Manual manipulation with MicroSD card contents

User is able to manipulate the files directly by inserting the MicroSD card root destination from the device into the dedicated device or a slot in a computer. This way, user is gaining tools provided by the operating system to manipulate the files. Such a functionality is useful in example, when only specific files needs to be deleted, or some special naming convention is used. However, few restrictions still apply, otherwise the device will not work as expected. The maximum of 8 characters long names rule has to be preserved. Another important factor is not to change the .CSV file extension of the existing files and also adding this extension to files created. It is not advised to create any sub-folders on the MicroSD card.

7 Technical details

This section covers some details of the inner workings of the device and also explains some design decisions. All the information contained here are meant to be read by the expert users.

7.1 Circuit explaination

The device circuit is shown on 7 − 1. It was created in the DipTrace electronics CAD software. The voltage regulator U5 is connected according to its datasheet[20]. So do the USB-to-UART bridge U4[19], the bus powered configuration is used. Apart from TX and RX lines, the only connection is the DTR line, which connects over a capacitor to the main microcontroller U1 reset line[16]. This allows firmware uploading over the serial port, so no external programmer is needed. Microcontroller gains this capability by running the Optiboot[2] bootloader[13]. The RTC chip U2[18] is also connected according to it's datasheet. Since the microcontroller does not have the hardware support for the Three line serial interface, communication is implemented purely in software on separate pins.

The PowerPath controller U3[21] has a special configuration. It's control CTL line is pulled-down by R10 and the GATE pin turns the P-channel MOSFET into On-state, so the current from from battery through Schottky diode D2. This diode is low drop-out, so together with a low drop-out voltage regulator, a minimum battery voltage for device operation can be as low as approximately 3.35V. This diode also stops current, that could flow from the USB bus through MOSFET substrate diode back to the batteries, when their voltage drops below the 5V. This could make batteries explode. The last function of this diode is to protect the whole circuit from situations, when a battery is inserted with wrong polarity.

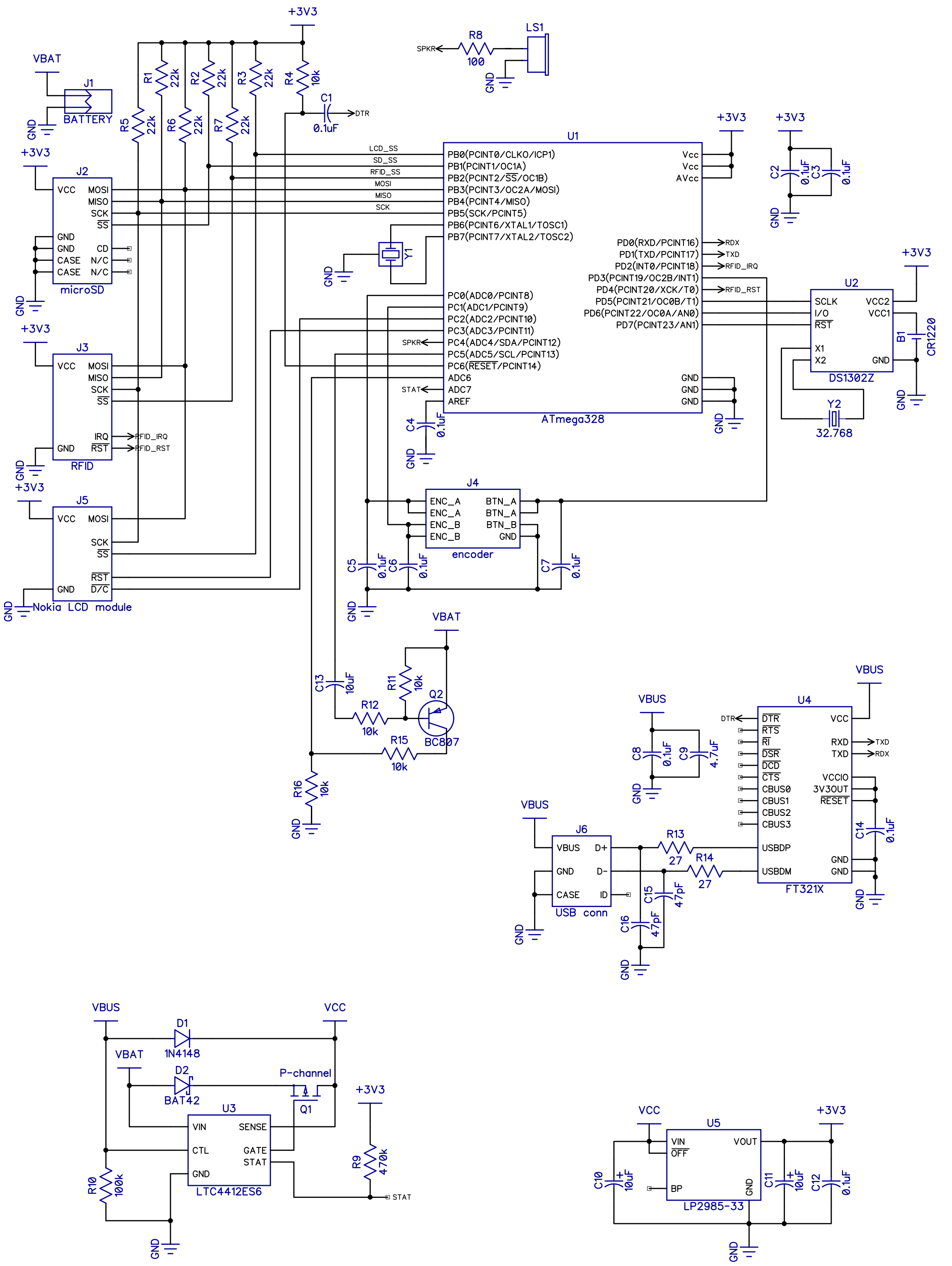

Figure 7 – 1 The device's schematic created in DipTrace

The U3 CTL line is pulled up by the USB bus. This shuts-down the MOSFET, so even when the battery voltage is higher than 5V, the current flows only from the USB bus through a normal junction diode D1. This diode also protects against wrong polarity, however this is unlikely in the USB bus, due to standardised connectors. It's main purpose is to stop the current flow from the batteries, when higher than 5V, to the USB bus. Also, the CTL line is protected, without the diode, it will be always pulled-up. Lastly, it leaves the U4 without power, when powered from batteries. This helps protects batteries, since there is no for the bridge without the USB connection.

Since the microcontroller provides two pins, that acts only as analog input, without the interrupt capability, we can utilize on of them to sense, which power source is active. This is achieved by connecting the U3 STAT line to ADC7, along with external pull-up resistor to 3.3V, since this analog inputs have no in-built ones. The other one is used for real analog sensing of the battery voltage. This is done by bringing the PC5 high. The transistor Q2 is then fully saturated, so the analog pin ADC6 can sense battery voltage through a voltage divider with the ratio of 1:1. This means that safe maximum battery voltage is 2 x 3.3 - around 6.6V. The capacitor C13 blocks DC current from battery through the transistor's base. After the measurement is done, the PC5 is brought down. This approach makes sure that no microcontroller pin connected to constant voltage higher than 3.3V and also that a battery isn't constantly drained by the voltage divider.

The only other microcontroller's inputs being the two rotary encoder lines, the button line and the RFID reader module IRQ, which is host interrupt request. All of this inputs needs a I/O pin that is capable of interrupt. Since all normal pins on the microcontroller are capable of pin change interrupt, they could be connected almost anywhere. However, the button and the IRQ are connected to PD2 and PD3, which are the external interrupt pins. Only this two pins can wake-up the MCU from sleep when the card is present or the button is pressed. Sleep helps to reduce

Table 7 – 1 Bill of materials used in the circuit

RefDes	Value	Pattern	Qty
B1	CR1220		1
C1 - C8, C12, C14	0.1uF	CAP_0805	10
C9	4.7uF	CAP_0805	1
C10, C11	10uF	TC_0805	2
C13	10uF	CAP_0805	1
C15, C16	47pF	CAP_0805	2
D1	1N4148	CAP_0805	1
D2	BAT42	MINIMELF	1
J1 – J6	connectors		1
LS1	Piezo		1
Q1	AP2309GN	SOT23	1
Q2	BC807	SOT23	1
R1, R2, R3, R5, R6, R7	22k	RES_0805	6
R4, R11, R12, R15, R16	10k	RES_0805	5
R8	100R	RES_0805	1
R9	470k	RES_0805	1
R10	100k	RES_0805	1
R13, R14	27R	RES_0805	2
U1	ATmega328	QFP-32/9x9x0.8	1
U2	DS1302Z	SOIC-8/150mil	1
U3	LTC4412ES6	SOT23-6	1
U4	FT321X	QSOP-20	1
U5	LP2985-33	SOT23-5	1
Y1	16.0	CSTCE16M0V53-R0	1
Y2	0.032768		1

the power consumption, thus preserve batteries. Whole action is invisible to the user.

Hardware encoder and button debouncing is done by the C5 - C7 along with MCU internal pull-up resistors activated on selected lines. R8 helps protect the pin driving the speaker, so it will not exceed it's rated 20mA maximum current source. Both the RTC watch crystal and the MCU resonator does not need a drive capacitors. They are built-in corresponding packages.

The last thing is SPI[13] communication. All lines has an external pull-up resistor. MCU built-in pull-up resistors should not be used for this purpose, since they are not active when the MCU in in reset - they are tri-stated. The communication lines pull-up resistors R1, R5 and R6 protects the slave SPI devices from any possible noise on these lines, when MCU is in the reset. They are also required by some MicroSD cards. The slave-select resistors R2, R3 and R7 are keeping all SPI slaves inactive, unless explicitly driven low by the MCU. This makes external MCU programming through the SPI possible.

7.2 Printed circuit board

The PCB schematic $7-2$ is dual-layer was created in the DipTrace software as well. The main design aim was to have a bottom side of PCB without the components to minimize the device height. It is also a good practice to have one layer of the PCB with as much ground plane, as possible[6]. The the PCB design had to be done in the way, that it is possible to create the PCB in amateur, home environment.

The process included application of dry photo-resist, which after exposed to UV light, covered by the actual PCB mask on transparent foil creates a etchant immune paths. For the etchant, the Hydrogen Peroxide with Hydrochlorid Acid was used. The drilling, vias and soldering was done manually.

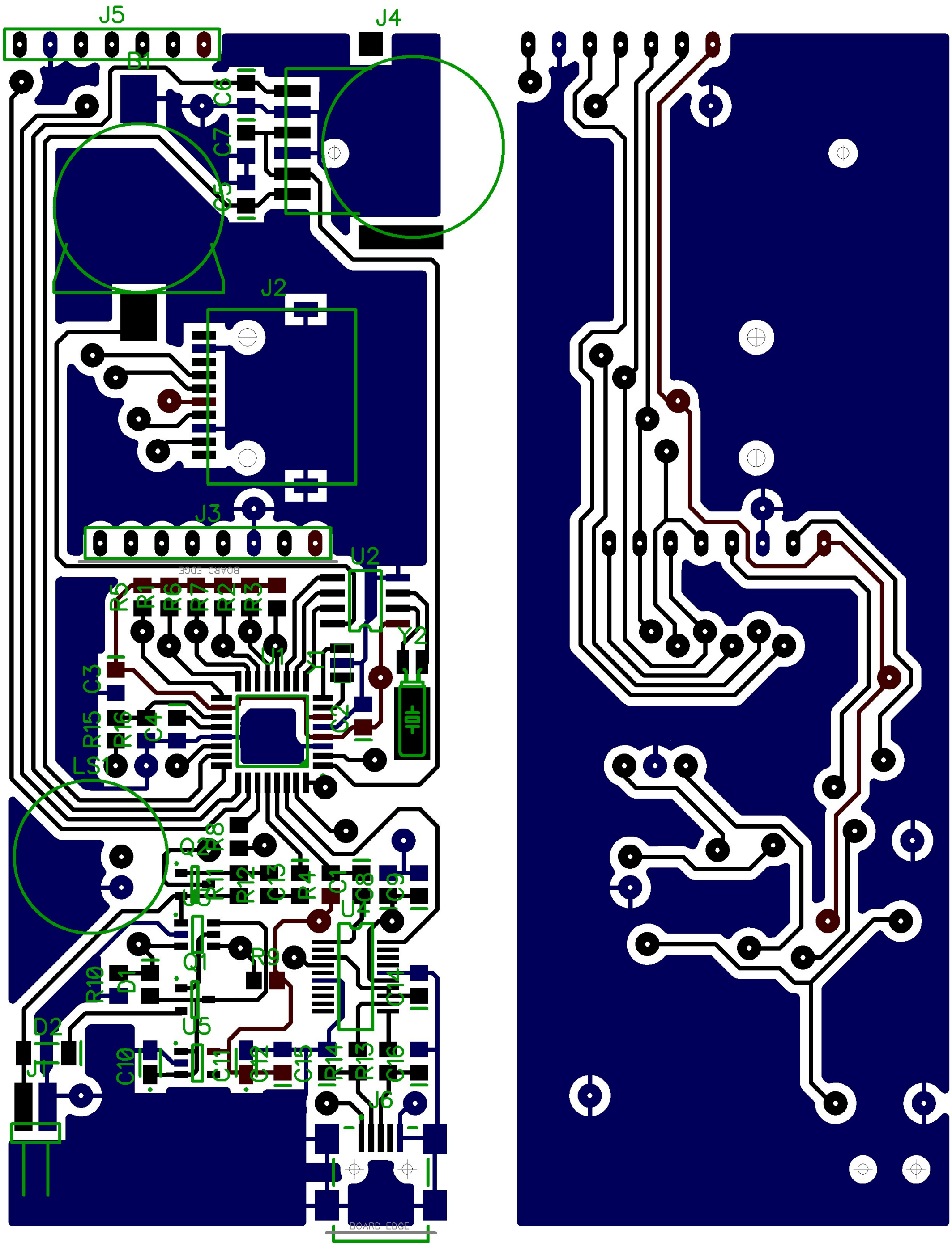

Figure 7 – 2 The device's PCB schematic created in DipTrace

7.3 Firmware algorithm

There should be a general idea of the device inner operation after the Circuit explanation section was read. The firmware consists of some separate Arduino files and libraries for operating the SPI, LCD, SD, RFID reader, RTC, clock prescaler and the rotary encoder. The strings for a serial communication and bitmaps for a LCD are stored in MCU flash memory, to preserve a precious RAM.

After the MCU is started, prescaler is activated, to make 8Mhz clock from 16MHz resonator for a safe 3.3V operation. Then the pull-ups are activated, the interrupts are initiated and the SPI communication is initialized. After that, LCD and RFID reader are initiated. In the end, the MicroSD card is tested. When everything is ready, the global interrupts are enabled and the program goes into the main loop, which goes over and over.

The main loop handles the received serial commands, refreshes the time,power indicator and storage file information, handles the received RFID data and shows menu, when the button was pressed. The menus and submenus are just endless loops again, that reacts on the button press or rotary encoder activity.

8 Conclusion

The main task was to construct a portable hand-held device, capable of reading the RFID tags contained in student's ISIC cards and store them onto the SD card. This task was fully accomplished and the device meets all the requirements. Additional features, like minimal user interface and a Serial connection over the USB with supportive commands were successfully included.

8.1 Possible future improvements

The biggest problem is absence of proper switch to turn device off. Present solution relies connectors making touch, when pushed. While it works, it is unintuitive and also unreliable. Also, the implementation of display's backlight would improve readability.

The second main concern are non-rechargeable batteries for main power and for RTC backup. The device could be improved by adding support for handling and charging the LI-PO rechargeable batteries. Used RTC chip has a charging capability, so the RTC backup battery should be changed to supercapacitor.

Also, the RFID reader module itself draws a lot of current, one order of magnitude more, compared to all other electronic. The passive LED diode contained on the module could be desoldered, to reduce the power consumption a bit.

There is also a lot of room to improve the efficiency of the firmware, regarding the power consumption. There was a little effort put in this, since the module itself draws far more current, so it is less important.

References

[1] Allan, R. *A History of the Personal Computer: The People and the Technology.* Allan Pub., 2001.

[2] Anderson, R., and Cervo, D. *Pro Arduino.* Technology in action series. Apress, 2013.

[3] Augarten, S. *State of the Art: A Photographic History of the Integrated Circuit.* Ticknor & Fields, 1983.

[4] Axelson, J. *Serial port complete [electronic resource]: COM ports, USB virtual COM ports, and ports for embedded systems, second edition.* Complete Guides Series. Lakeview Research, LLC, 2007.

[5] Coskun, V., Ok, K., and Ozdenizci, B. *Near Field Communication (NFC): From Theory to Practice.* Wiley, 2011.

[6] 'Hara, M. *EMC at Component and PCB Level.* Elsevier Science, 1998.

[7] Khandpur, R. *Printed Circuit Boards: Design, Fabrication, Assembly and Testing.* Tata McGraw-Hill, 2005.

[8] Kularatna, N. *Electronic Circuit Design: From Concept to Implementation.* Taylor & Francis, 2008.

[9] Lehpamer, H. *RFID Design Principles.* Artech House, 2012.

[10] MATHIVANAN, N. *PC-BASED INSTRUMENTATION: CONCEPTS AND PRACTICE.* PHI Learning, 2007.

[11] Monk, S. *Programming Arduino Next Steps: Going Further with Sketches.* McGraw-Hill Electronics. McGraw-Hill Education, 2013.

[12] Solomon, M., Rudolph, K., Tittel, E., Broom, N., and Barrett, D. *Computer Forensics JumpStart*. IT Pro. Wiley, 2011.

[13] Susnea, I., and Mitescu, M. *Microcontrollers in Practice*. Springer Series in Advanced Microelectronics. Springer, 2006.

[14] Toulson, R., and Wilmshurst, T. *Fast and Effective Embedded Systems Design: Applying the ARM mbed*. Fast and Effective Embedded Systems Design: Applying the ARM Mbed. Elsevier Science, 2012.

[15] Wikipedians, T. *An introduction to electronics*. The Art of Electronics. PediaPress GmbH, 2011.

[16] Arduino pro mini, rev.3 schematics. `http://arduino.cc/en/uploads/Main/Arduino-Pro-Mini-schematic.pdf`. (Visited on 05/27/2014).

[17] Atmega328p datasheet. `http://www.atmel.com/Images/doc8161.pdf`. (Visited on 05/27/2014).

[18] Ds1302 datasheet. `http://datasheets.maximintegrated.com/en/ds/DS1302.pdf`. (Visited on 05/27/2014).

[19] Ft231x datasheet. `http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT231X.pdf`. (Visited on 05/27/2014).

[20] Lp2985-33 datashet. `http://www.ti.com/lit/ds/symlink/lp2985-n.pdf`. (Visited on 05/27/2014).

[21] Ltc4412 datasheet. `http://cds.linear.com/docs/en/datasheet/4412fa.pdf`. (Visited on 05/27/2014).

Appendices

Appendix A CD with electronic form of thesis, schematics source files, firmware source code and all the software needed to open and compile the all the provided source files