Technical University of Košice
Faculty of Electrical Engineering and Informatics

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

System Guide

2014 Peter Babič

Contents

List of Figures

1 Program function

The main task of a program, running inside a microcontroller is to access the data obtained by a RFID reader module and store them on a storage medium, for which an MicroSD card was chosen. The hardest part of this task is to communicate with multiple components simultaneously and moving or modifying data they hold.

Program must also react on external events, that can come from the user interface, power management or PC communication circuitry and act accordingly, mostly by putting the response on a LCD display.

2 Program requirements

In present days, there exists multiple libraries for managing almost every action possible by this device. Since the Arduino platform was chosen, the libraries are often normalized to conform some compatibility standards. In most of them, object-oriented programming is embraced, which further simplifies implementation and code reusability.

2.1 Individual tasks

The program should handle these tasks:

- 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

2.2 Required libraries

Since the memories, contained within the microcontroller are small enough, that we could easily hit their limits, the used libraries should be small size, more importantly, they should consume as little RAM as possible. Potentially useful libraries:

- PCD8544 driver for dot-matrix LCD display
- MicroSD card library with FAT16 library
- MFRC522 Mifare communication chip
- commands over serial line library
- rotary encoder and push-button with interrupt support and aditional software debouncing
- clock prescaler library
- DS1302 RTC timekeeping chip library

We do not look explicitly for UART, SPI, ADC or EEPROM libraries, since they are natively part of the Arduino framework, and they work well.

3 Program description

The program consists of two main parts: the setup part and the loop part. The setup part does the things, that needs to be done once which are mostly initializations - clock prescaler, LCD, interrupts, SPI, pins and so on.

The loop part does the same thing all over again. This main cycle check, if there was a command received over a serial line, refreshes the time shown on the display, check if the button was pressed and then checks if some data are available from the RFID reader.

3.1 UML activity diagram

Activity diagram of the internal workings of the device's embedded system is shown in 3 – 1. As we can see from the activity diagram shown above, the device itself (red) is responsible for storing the actual time and date and keeping track of it, checking and reading the filesystem on the SD card for files and writing into them, providing the backend for the device human interface and servicing the communication with RFID capable card working on Mifare protocol.

Actions

- Set time and date is done by the interface consisting of LCD display and pushbuttons
- Load SD card with meeting names is done on any device capable of creating files on the FAT filesystem
- Choose meeting to record into again, done on the device pushbuttons interface

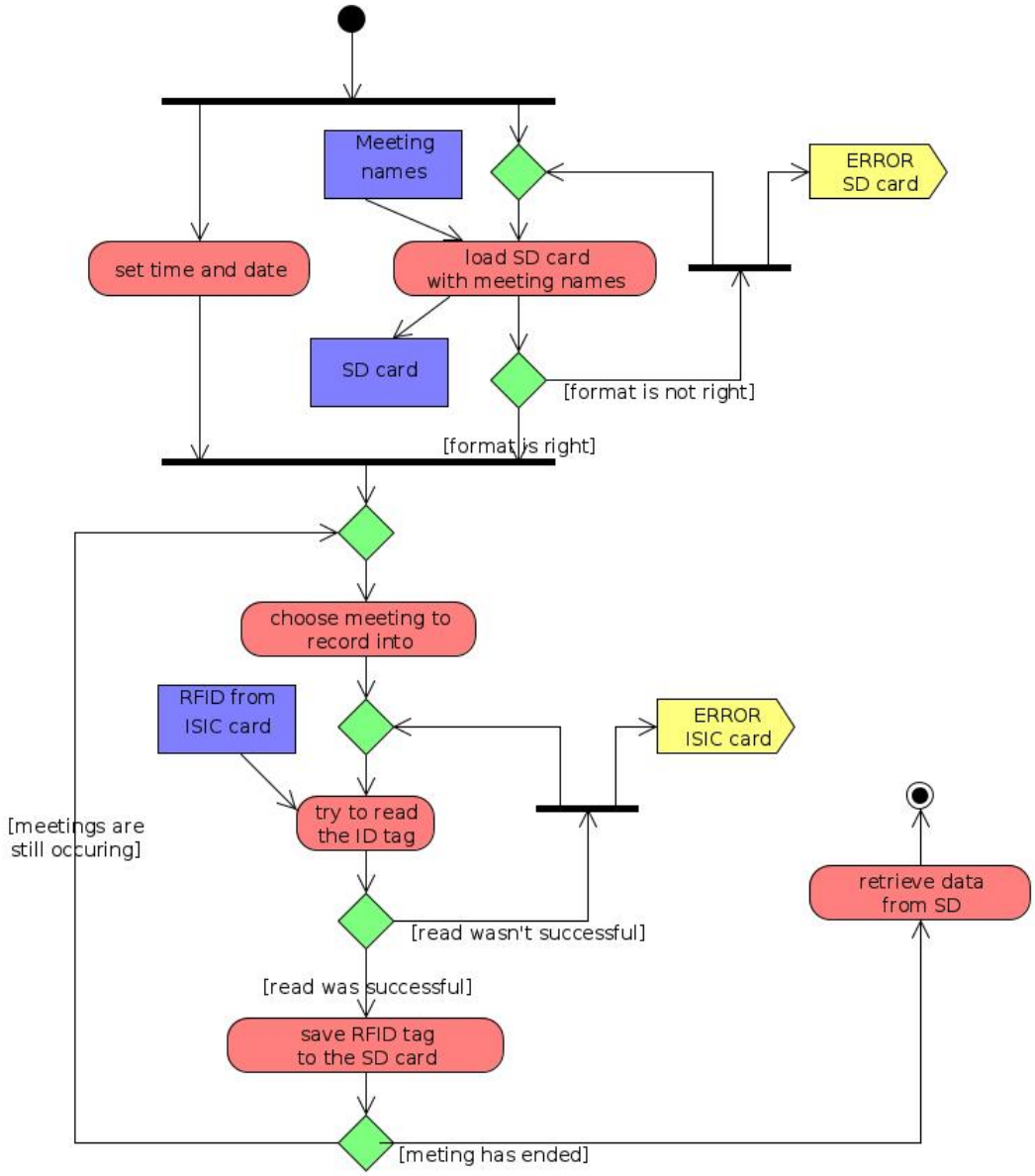

Figure 3 – 1 Activity diagram of the program in UML language

- Try to read the ID tag is done automatically when the ISIC card is present in the magnetic field of reader
- Save RFID tag to the SD card is done automatically when the reading of the RFID tag is successful; tag is stored with actual time and date

Data sources

- Meeting names of the text files without extension on the SD card
- SD card contains the files (meeting names) that contains the meetings data (time, date, RFID tag) on the FAT filesystem
- RFID tag unique tag read wirelessly from ISIC card

Signals

- SD card error is displayed on the LCD display
- ISIC card error is displayed on the LCD display

3.2 UML use-case diagram

Unlike th activity diagram, in the Use case diagram, shown in 3 – 2, the responsibilities of device users are depicted. From the diagram, it is clear that three systems are needed for whole concept. The main system (RFID reader) is the system embedded in the device. Two more subsystems are needed: one fro making sense of the data (statistics, analytics) and one for easy manipulation with SD card contents, since the human interface of the main system is very limited.

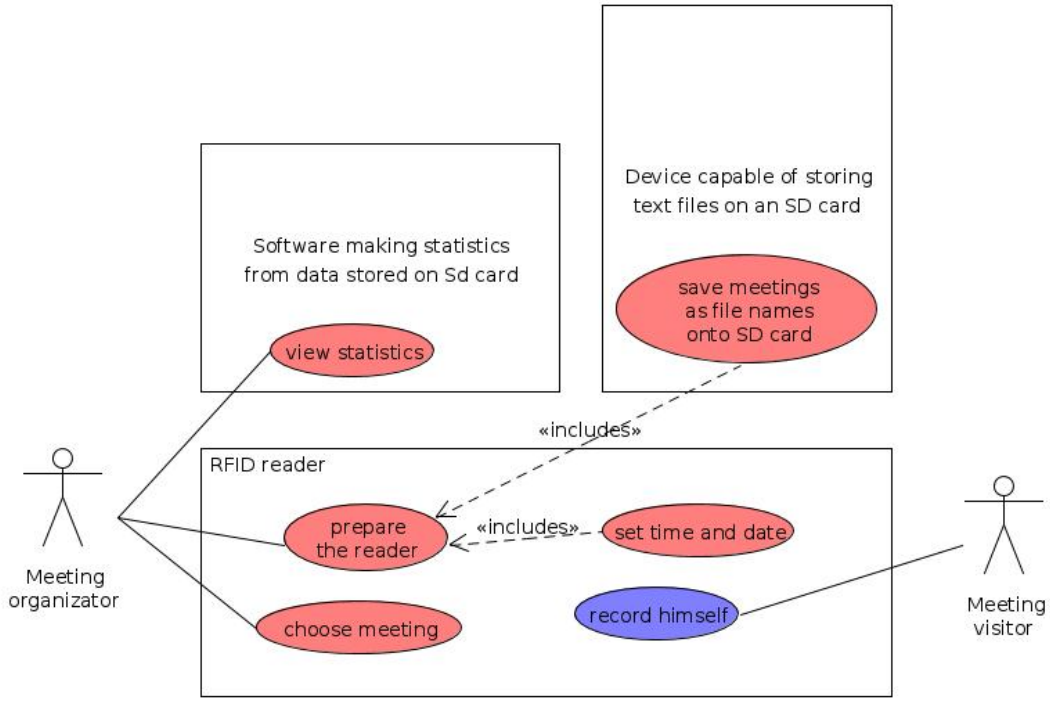

Figure 3 – 2 Use-case diagram of the program in UML language

Meeting orgnizator (red actions)

Organizator is responsible for preparing the reader by storing empty files, without extension on FAT filesystem formatted SD card on the device, which is capable of doing so. The file names are actually the meeting names. The data are stored in these files. He is also responsible for setting up time and date, after each battery replacement. Before meeting start he chooses the file which data will be saved into. Viewing the data statistics and analytics is subject to further expansion of the system, for now, they are only available in raw format.

Meting visitor (blue actions)

When the reader is fully prepared by the organizator. meeting visitor's only responsibility is to firmly put his ISIC card on the reader for brief period of time (until beep is sound is played). Data are wirelessly transmitted into the reader and processed there. When this happens, the visitor's presence on the meeting is recorded and stored on the SD card for further analysis.

4 Program compilation

The program can be compiled with the use of Eclipse IDE, when opened as a project right out of the box. Manual compilation is possible, but is really complicated and should not be used. Versions of compilation tools:

- `avrdude-5.11.1`
- `avr-libc-1.8.0`
- `binutils-avr_2.20.1-2`
- `gcc-avr_4.7.2-2`

References

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

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

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