Design and development of an Access Control System (Building Automation Systems)

Isidoros C. Vakondios[1]
Filippos G. Chalkiopoulos[2]
Prof. John Ellinas[3]

Abstract

In this thesis, we are implementing an Access Control System based on a topology, made with IP Readers and a Main Controller with IP network capabilities. Our main concern of this project, was to overcome the obstacles that most Access Control System have, the installation on a building and the ease of access from the administrator. The new elements that we gave on this system are the WiFi connectivity to IP Readers, that will lead as to a smaller structured network in the building and a Web Application that is served from the main controller to any device on the local network, from were this system is being controlled by the administrator.

1. Introduction

The main idea behind the term “access control”, describes any technique used to control the entrance in a secured area. Nowadays, with the term “Access Control System” we are describing a computer-based system with electronic control of the credentials of a person. That kind of system are capable of making decisions, for giving access to a person, in a particular area.[1] The access that is given to a person is based on the instruction, the administrator gave. There are multiple techniques to build an access control system. The main concern, for that kind of systems, is where in the system, the decision for access are taken. Over the years the designs that has been prevailed, made a transfer of the responsibility of access, from the main controller(s) of the system to readers. Readers were supplied with microcontrollers and were capable of making decisions. In this project we are building a system, with IP Readers and a Main Controller. IP Readers are capable of making decision about the access in the area that are installed. Main Controller is responsible for programming the IP Readers, and also delivers a web application for administration.

2. Overview

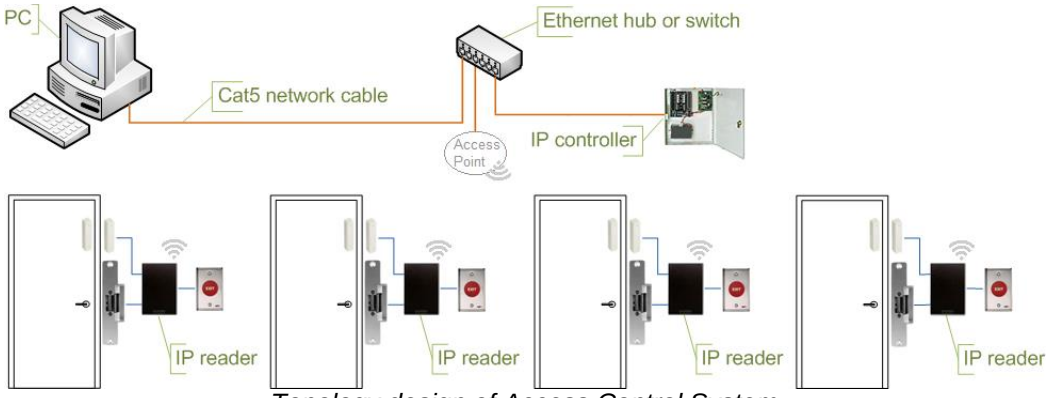
In this thesis, the Access Control System that we are implement, we have the “brain” of the system, which is the main controller. Around the main controller we are building the system with IP Readers that is being installed on any area that we need to control the access to. IP Readers make the check of the credentials, by reading from a RFID Reader the tags, that users are being supplied for access. The communication with the Main Controller, is performed with TCP/IP protocol through WiFi communication. The main controller keep the database of the system, with log and system changes. It also delivers a web application for the administrator to “talk”

[1] Isidoros C. Vakondios, Student, email: aris.vak2011@gmail.com

[2] Filippos G. Chalkiopoulos, Student, email: pcgchalkiopoulos@gmail.com

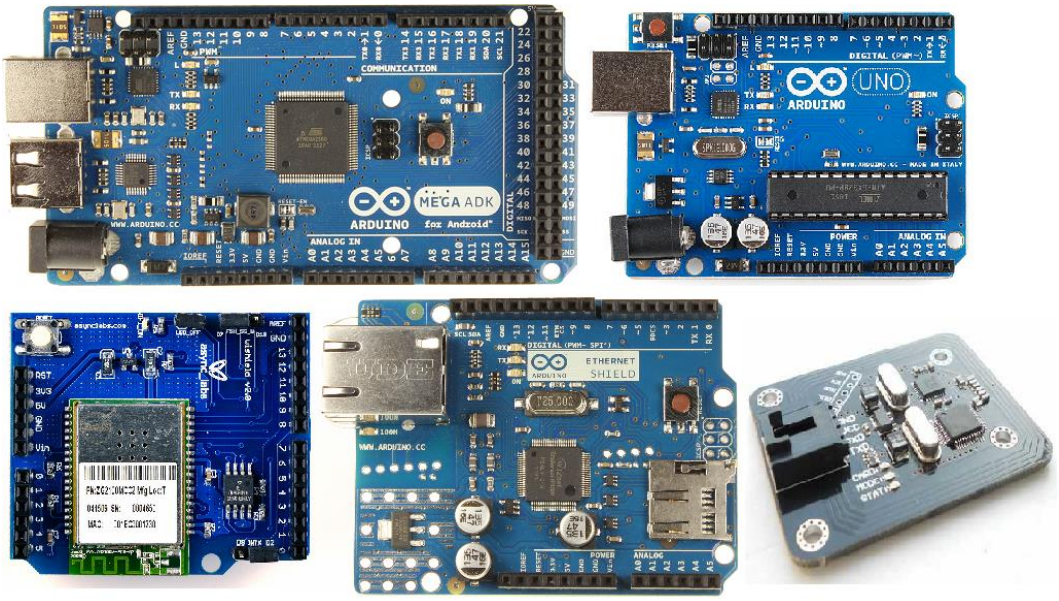
[3] John Ellinas, Professor, email: jellin@teipir.gr
Department of Electronic Computer Systems Engineering
Piraeus University of Applied Sciences

to the system, for making the necessary changes. Communication is also with TCP/IP protocol with Ethernet connectivity on network.



Topology design of Access Control System

For the development of this system, we have relied on the Arduino Platform. We are using two microcontrollers of the Arduino family, the Arduino Uno (based on atmega328), for the implementation of the IP Reader, and the Arduino Mega (based on atmega2560) for the implementation of the Main Controller. For the connectivity of this system we are using the WiShield 2.0 (based on Microchip' s MRFWB0MA) and the Ethernet Shield (with microSD slot) of the Arduino family. For the RFID reader, we are using a module based on the MRFC522 chip, with serial communication[2].



Project's Hardware

2.1. System Design

The IP Reader is being designed to control the entrance. The microcontroller is necessary to read the card code from the RFID reader, check it's memory and whether to give or not give access. After this process is being done, we have to send to Main Controller the action that was performed along with the card code. Alongside with the previous actions, we have to settle the IP Reader in a mode for getting data from Main Controller that are necessary for adding or deleting a user's card code from memory.

From the scope of Main Controller, we have to implement three different communication states. The first two will allow IP Readers to open a channel for sending data to a specific IP Port on the main controller and another one doing the opposite communication. The third state will implement a web server, that will allow web clients (through Port:80) to connect to server and serve them the web application through the web files that are stored in a microSD card. The database files are also being stored, on the microSD card

2.2. System Implementation

In an IP Reader the procedure that takes place, are the following actions:

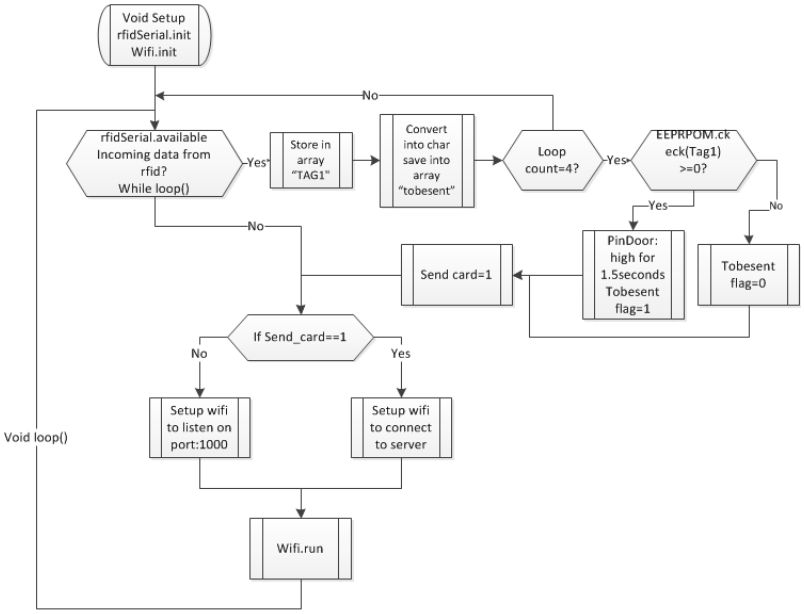
Card reading, through the RFID card reader. The communication between microcontroller and RFID is on a Serial bus of the microcontroller.

Card check. We check card code with the ones that microcontroller have in it's EEPROM memory and if a pair is found we give access.[3]

Sending action to Main Controller. Here we are sending data to main controller through WiFI with the use of WiShield 2.0. We are connecting WiShield on the microcontroller through the SPI interface.[4]

Listening to a TCP Port: We are listening the port:1000 for getting data from the Main Controller. These data will lead to the operation of adding or deleting a card from the EEPROM memory.

Here, we can see the flow chart of the Arduino sketch that was uploaded to implement the IP Reader.



IP Reader flow chart

For the construction of the Main Controller, we connected an Ethernet Shield to Arduino Mega. This shield also delivers a microSD slot. The actions that the Main Controller operates, are the following

Send data to IP Readers. Here we create an instance of EthernetClient class of the Ethernet Shield library. We are connecting the main controller as a client to the desired IP Reader, and through that channel we send the data for adding or deleting a user's card code.

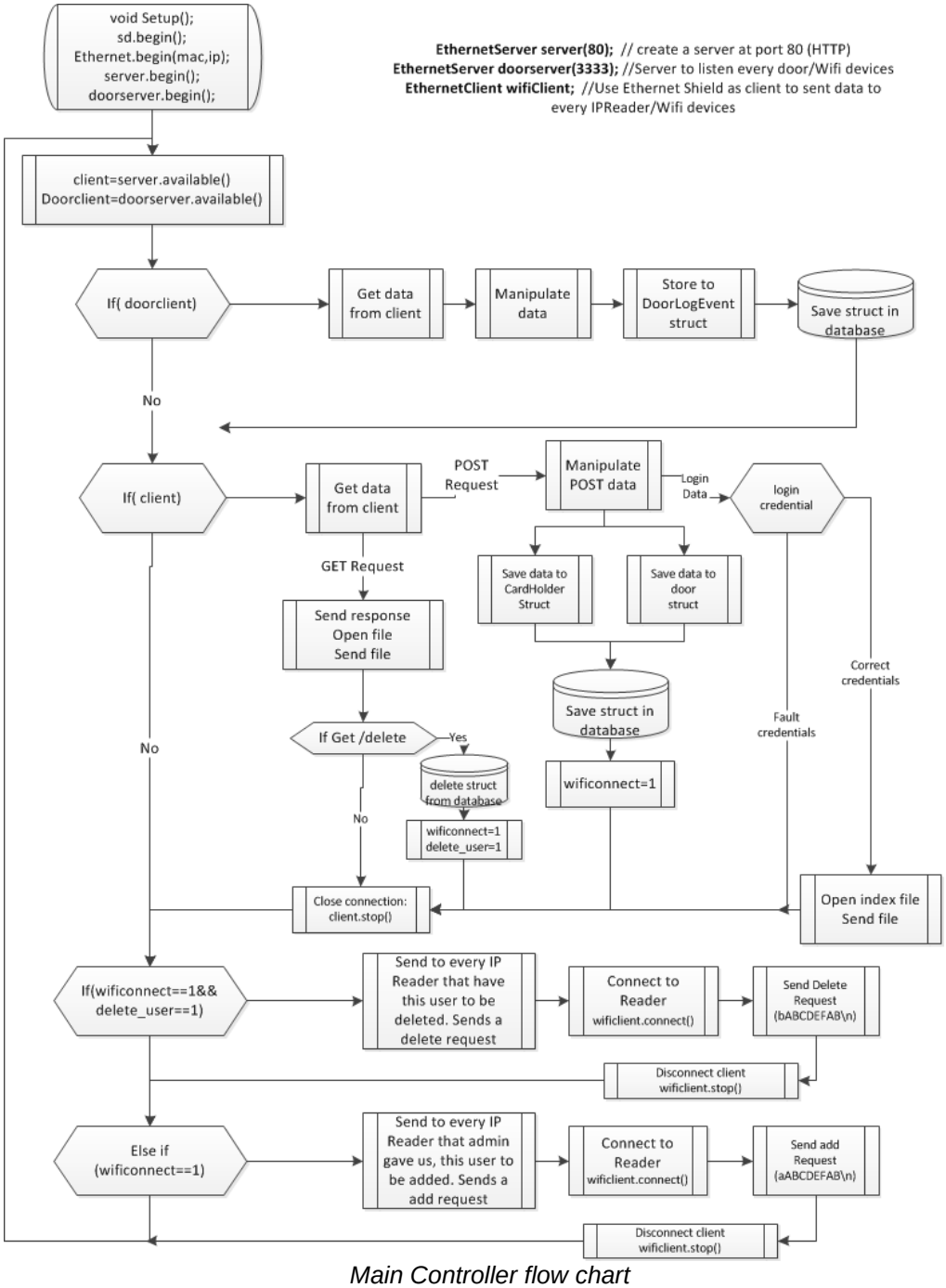
Receive data from IP Readers. We are using the EthernetServer class for creating an instance of a server with listen port:3333. If an IP Reader connects we read the data. The data are necessary for making the Log of the system.

Data exchange through microSD. With the use of the SDfat library we can create, modify, read or delete a file from the microSD memory card. Here we are storing the web page files, along with the database files.[5]

Manipulating database files (*.db). With the use of EDB library we can create a table on a database file and store them into microSD card. We can also store, read or delete a record from the table with the ease of saving and reading structs for every table record.[6]

Web page serving function. As we mentioned earlier we can use EthernetServer class for reading data from a particular port. Here we use port:80 that is responsible for HTTP messages thereby creating a Web Server. Depending on the message that was arrived, web server respond with the appropriate file or operation.

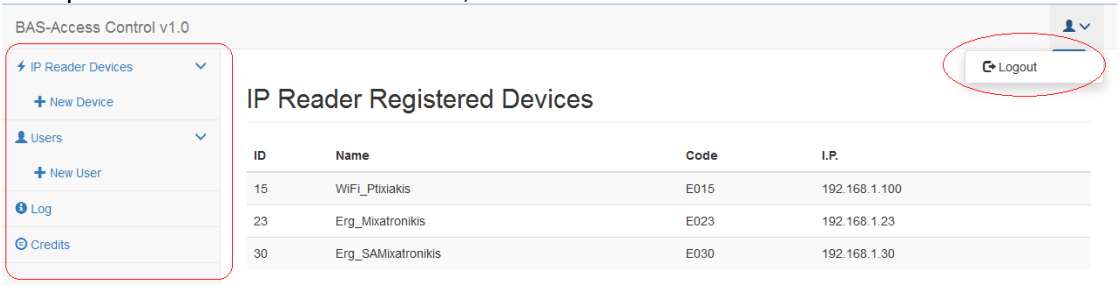
Here, we can see the flow chart of the Arduino sketch that was uploaded to implement the Main Controller's functions.



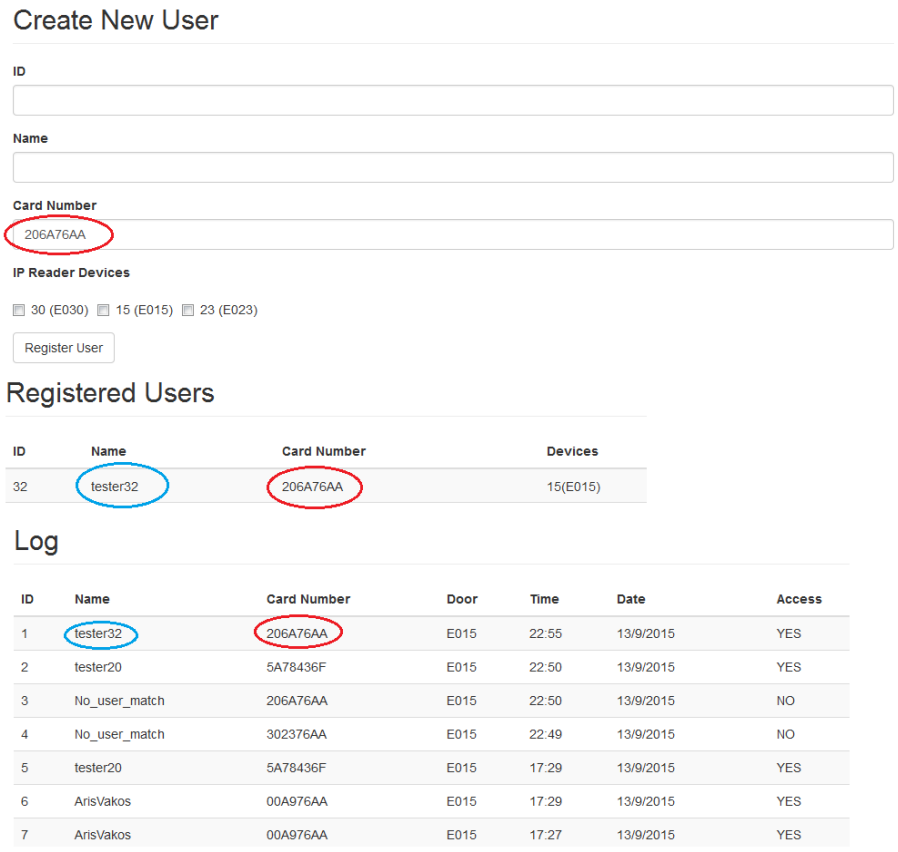
Main Controller flow chart

3. Results – System Administration

The above operations that was mentioned, were tested on real scenarios. Through the web application we can add, edit or delete a new IP Reader to the system giving the specific information that is need, such as a name and the IP address.



We can also add edit or delete a User. The fields that are required to store a new user on the system, are id, name, card code and the IP Readers that will have access to.
Finally we can see the Log button which will lead us to a screen that gives us the log information of the system.



3. Conclusions

The present thesis is dealing a new scope of creating an Access Control System. With our big concern to avoid network cables for IP Readers, we had to implement a WiFi communication between them and the Main Controller. Also with the Web Application, the administration of the system can be easy and can be accessed from any device on the network.

References

[1] Introduction to Access Control Systems, *by Silva Consultans*

http://www.silvaconsultants.com/introduction-to-access-control-systems.html

[2] 13.56MHZ RFID Reader/Writer Module
http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath=90_93&products_id=2156

[3] Arduino EEPROMex Library *by Thijs Elenbaas*
http://playground.arduino.cc/Code/EEPROMex

[4] WiShield library. Create by AsyncLab, modified by cmts
https://github.com/cmts/WiShield

[5] Arduino FAT16/FAT32 SdFat library https://github.com/greiman/SdFat

[6] Extended Database Library
http://playground.arduino.cc/Code/ExtendedDatabaseLibrary

You can find our Arduino sketches on the following links:
From www.codebender.cc
Create Database: https://codebender.cc/sketch:153127
IP Reader: https://codebender.cc/sketch:153128
Main Controller: https://codebender.cc/sketch:153129

From Github https://github.com/
https://github.com/PhilipChalkiopoulos/Thesis_Project.git