EE389 Electronic Design Lab II Project Report, EE Dept, IIT Bombay, November 2008

RFID Access Control

Group No: D8
Gauri Joshi, 05D10019, gaurijoshi@iitb.ac.in
Nistha Tandiya, 05D07036, nistha@iitb.ac.in
Harshad Maral, 05D07006, harshadm@ee.iitb.ac.in

Abstract

The problem statement that we started out with at the beginning of the semester was to design and implement an RFID access control system for a workplace. The aim was to come up with a cost-efficient RFID reader and lock that could be installed at the entrances of various offices of the workplace. These locks would also be the nodes of a local network with a central server that would facilitate two-way communication between the central server and each of the locks. We have been successful in achieving all the above-mentioned goals, the technical details of which we discuss in the next few sections.

1. Introduction

By the end of the first stage of EDL (EE318), we had built an RFID reader that could successfully sense the presence of RFID tags in its vicinity, read them and send out a valid/invalid decision based on the tags' information table stored in it. We had also been working on a stepper-motor controlled lock, but were unable to demonstrate successful working of the same. In this section, we discuss the details of the goals that we had laid down at the beginning of the project.

The first task was to fix the stepper-motor controlled lock and interface it with the RFID reader such that the lock opens the door automatically when a valid card is flashed near the reader and closes it after a specified amount of time. Needless to say, the lock should remain in the unlocked state in case an invalid card is flashed near the reader and even in the case of power failure.

The next task was to choose a networking standard and design a local network for RFID locks that would be suitable for workplaces like government offices, libraries or our very own EE department! The locks installed at the entrances of different rooms should regularly send the information of valid or invalid access sensed by them to a central server. The central server should be able to modify the tags' information table stored in the locks thus enabling the administrator to grant access to specific rooms to specific individuals only and also to update the table so as to add information of new employees.

Other tasks included designing a Graphical User Interface for the administrator's use at the central server and adding alarms like buzzer (that would beep if an invalid access is made) and LEDs (that would turn red if access is denied and green if access is granted) on the locks to make the project a complete RFID Access Control kit and as close to market products as possible.

2. Design approach

A. Hardware

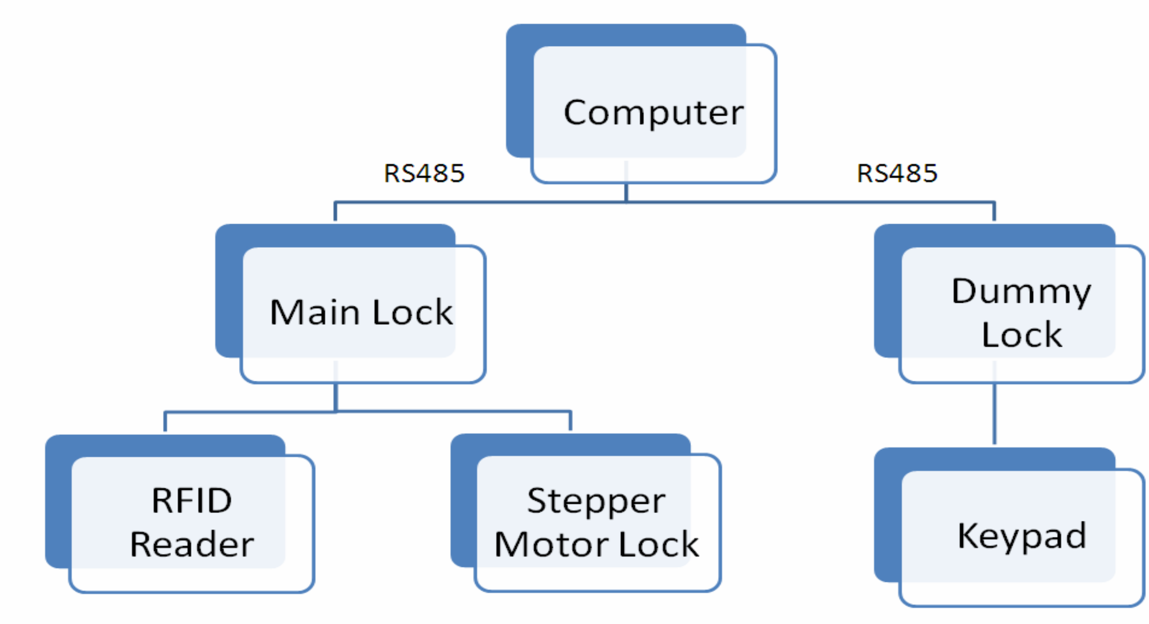
Ethernet seemed to provide all the functionalities of the network standard that we were looking for. However, due to reasons discussed in the next section we could not go ahead with it. The next best option was the RS485 standard that is popularly used in long-distance wired network-based communication systems.
We also decided to include a variety of display media like LCD, buzzer and LED as discussed above. We also included a stepper motor controlled door lock that will open and close when a valid tag is swapped. We chose a stepper motor over a magnetic lock because a stepper motor lock is more secure.

B. Software

In order to record and store the time at which an access was made at a particular lock, we had initially planned to use a real-time clock IC. However, we later decided to implement this functionality in software along with the graphical user interface.
Also, we optimized the code for RFID tag detection (from last semester) in terms of delay and other tag-detection algorithm issues.

3. Design of circuit

A. Block Diagram



B. Main Components
Our circuit has the following main components:

Microcontroller: We are using a 8051 series microcontrollers 89C51 and 2051. We chose to work with the 8051 series because our application does not require any high speed computation or memory capability in the microcontroller. The main role of the 89C51 microcontroller is to control the signaling to the RFID reader PCB. This consists of a clock signal, enable signal and 8 data lines. The 2051 microcontroller is used for the motor circuit and RS485 to RS232 converter.

RFID reader PCB: We have designed and fabricated the RFID reader which communicates with the RFID tags. The PCB consists of the RFID chip TRF7960 from Texas Instruments. This chip is used for RFID communication at the frequency 13.56 MHz. The other standard RF frequencies are 134 KHz (LF) and 868 MHz (Ultra-High-Frequency UHF). The advantage of 13.56 MHz frequency is that the RFID tags are cheapest for this frequency. The read range is sufficient for our application and penetration through obstructing materials os better than with UHF.

It is connected to the antenna through an impedance matching network as shown in the figure below. The reader has an antenna that sends a carrier signal to the RFID tag and energizes it. The tag then modulates the carrier with its response and sends it back to the RFID reader. The TRF7960 IC demodulates and decodes the data before sending it the microcontroller through the 8 data lines.

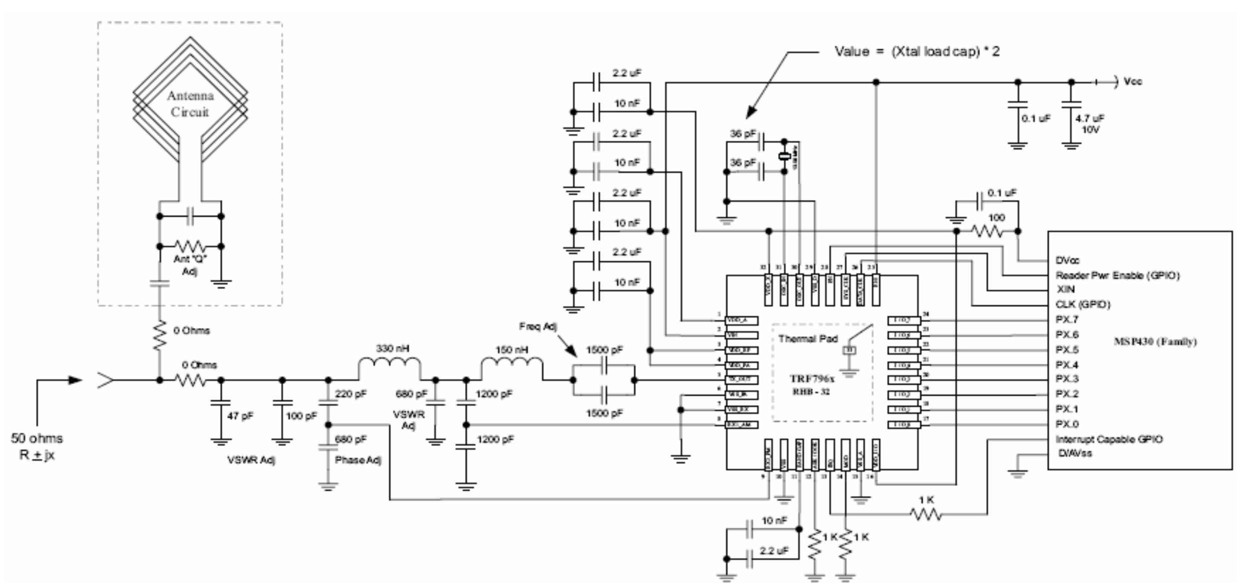


Fig1. RFID reader PCB circuit diagram

Antenna: We fabricated and tuned the antenna to the frequency 13.56 MHz using a network analyzer. Fig 2 shows the diagram for our antenna tuned at 13.56 MHz. Here is a description of the matching components and the tuning procedure. Across the parallel pads, are a 10k Ohm, 1W resistor to reduce the Q to 21 and 132 pF (100pF + 10 pF + 6 to 30 pF variable, 100 V) capacitance for the resonant frequency adjustment. In series is a 27 pF (10 pF + 5 to 20 pF variable, 100V) capacitance to match the antenna to 50 ohms.

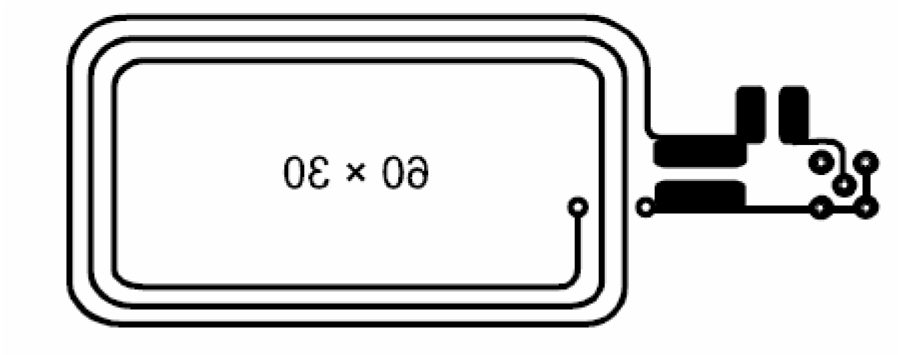


Fig 2. Antenna circuit diagram with dimensions

Motor circuit: After detection of a valid ID card, the main microcontroller directs the motor driver to open the lock. The lock is driven by a unipolar stepper motor. Instead of sending these signals from the central microcontroller we added a 2051 microcontroller to the motor PCB circuit. This reduced the number of external wires to be connected from the main PCB to the motor PCB.

The motor driver circuit consists of a 2051 microcontroller, ULN2003A which is an array of seven pair Darlington transistors, a free willing diode and some pull up resistors. Corresponding to each end of the coils of the motor 4 data lines are required from the microcontroller. ULN2003A takes these data lines at TTL logic from the microcontroller and converts them to a voltage level of 20V to feed the motor.

Network of Locks: To help multiple locks communicate with a central computer, we designed a network of locks. Here are the interfaces we tried.

a. **Ethernet interface:** We had initially planned to implement an Ethernet Interface using the IC ENC28J60 (Stand alone Ethernet controller with SPi interface). We even tested the SPI interface of the 89S8252 microcontroller with a temperature sensor. However, Ethernet requires the implementation of stack protocols. We decided that this would be difficult to complete in the given time period. Also, we needed a higher level microcontroller for implementing Ethernet and It would have been additional work to convert our RFID code to the new microcontroller syntax. Hence, we decided to use the serial interface.

b. **Token Ring RS232:** In general RS232 can only connect one microcontroller to the computer. However, after an interesting suggestion by Prof. D K Sharma we set up a token ring network using RS232. In this design we connected the microcontrollers in a ring with only one line TX/RX running between two uCs. But the drawback of this network is that if one uC in the ring fails, the whole network will stop working. So we replaced it with the RS485 network.

c. **RS485 network:** RS485 is a protocol similar to RS232 except that it allows us to connect multiple microcontrollers to the computer. We are using the RS485 transceivers SN65HVD05 from Texas instruments. Since RS485 signals are not directly recognized by the computer, we also designed and fabricated the PCB of a RS485 to Rs232 converter.

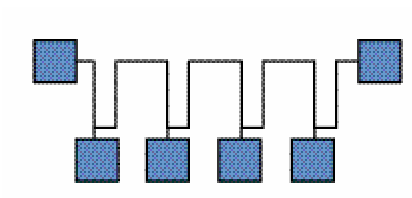


Fig 3. Daisy chain network (preferable)

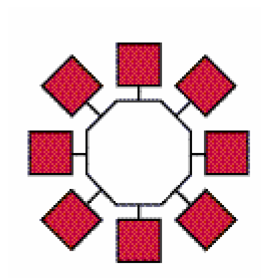


Fig 4. Token Ring (should be avoided)

Dummy Lock: To demonstrate the working of the RS485 network described above, we have also implemented a dummy lock. The RFID part is not present in the dummy lock. Instead, we use a 4x4 keypad to model the swapping of cards. The key number 0 to F is considered as the least significant bit of the card swapped.

System Time: We tested a circuit with the real time clock IC DS1307. We were planning to use this to find the time at which a card is swapped at any lock. However, since our GUI handled the job of getting system clock for this, we decided to use this facility in the computer instead. Taking the system clock form the computer helps us avoid unnecessarily sending real-time bytes over the network every time a card is swapped.

Graphic User Interface (GUI): We have successfully developed a GUI using MATLAB which records the lock number, Unique ID of the tag and the time of card swapping in an excel file.

4. Testing Procedure and Results

1. **RFID testing**: We had successfully implemented the card detection last semester. The range and penetration of RF was tested for our circuit. The range is found to be 8 to 10 cm which is sufficient for the application of RFID access control. It is also found that the detection is not affected by obstructing materials between the tag and the antenna. We tested this for plastic (5cm thick), cloth, cardboard and it was found that the system still functions properly.

2. **Stepper motor testing:** We tested the stepper motor lock for a range of voltages between 10 to 30V. It was found that the motor has sufficient torque to open the lock at driving voltage of 20V. Also, after testing, we found that the motor gets better torque in the half stepping mode (signals sent to turn on the motor coils are 1100, 0110, and so on) rather than the full stepping mode (1000,0100 etc signals Pull up resistors are introduced in the circuit since the microcontroller was not able to give proper voltage output. As the coils of the motor are frequently turned on and off, free willing diodes are introduced to prevent large back current to flow from the motor coils which may destroy other circuit elements.

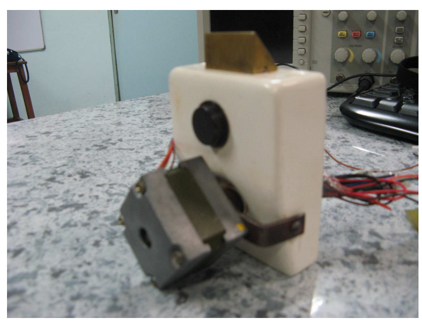


Fig 5. Stepper motor controlled Lock

3. **RS485 network testing:** We first tested the RS485 network for one way communication to the PC. We used this to create a system log of the list of UID's of swapped tags at a lock. This information was used in the Graphic User Interface and exported to an excel file.

   Then we tested the computer to uC interface with the RS485 converter. This interface was also successfully tested and used to validate and invalidate tags at each of the locks using the central computer. The central computer will send three bytes, the lock number, valid/invalid byte and UID to be validated or invalidated. (E.g. 1v3 (Lock1, UID 3 is set valid), 2i5 (Lock2, UID 5 is set invalid))

4. **PCB Testing:** We designed the main and dummy lock PCB and got it fabricated. However while testing the main circuit we found that the LCD part was not working properly. The RFID communication and RS485 part of the main PCB is working.

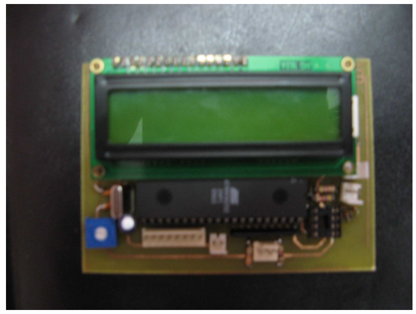


Fig 6. Final PCB main board.

5. Conclusion and further improvement

The above discussion shows that we have been able to implement and demonstrate a prototype of a network-based RFID Access Control. In this section we discuss the scope for improvement in individual parts of the project.

While RS485 is a good enough option, it still requires laying down cables every time a new lock is installed and the job might become tedious. A better option is Ethernet that we have explored in the previous sections. If one is able to make an Ethernet interface for these locks, the problem of cables will be solved and it will also ensure a faster communication speed between the locks and the central server.

We have also demonstrated the basic working of the stepper-motor controlled lock. For real-life scenarios though, we might require to make the opening and closing events faster. The torque provided by the motor should be improved upon in order to meet this requirement.

We have also shown the basic software that will run on the central sever and record information of the access made at different locks. The GUI needs to be improved upon and made more user-friendly and the feature of validating and invalidating cards from the server to the locks needs to be implemented.

6. References

[1] Datasheet TRF7960-61 Multi-Standard Fully Integrated 13.56-MHz Radio Frequency Identification (RFID) Analog Front End and Data Framing Reader System, Texas Instruments.

[2] Implementation of the ISO15693 Protocol in the TI TRF 796x, Aarthi Thiruvengadam, ShreHarsha Rao (Editor), November 2006

[3] Identification cards - Contactless integrated circuit(s) cards. Part 3: Anti-collision and transmission protocol, Texas Instruments, March 2000

[4] Datasheet SN65HVD05 RS485 transceivers, Texas instruments

[5] Data Sheet ENC28J60 Stand-Alone Ethernet Controller with SPI™ Interface, from Microchip Technology Inc., 2004
[6] Datasheet - DS1307/DS1308 64 X 8 Serial Real Time Clock, Dallas semiconductor
[7] Datasheet - ULN2003A, Texas Instruments
[8] RFID for dummies, Patrick J. Sweeney II, Wiley Publications, 2005
[9] microID® 13.56 MHz RFID System Design Guide, Microchip Technology Inc

7. Appendix A : User manual

There are five printed circuit baords in the complete project as follows:
1. RFID reader
2. Main lock
3. Dummy lock (only for the purpose of demonstration)
4. RS232-RS485 converter
5. Stepper motor controlled lock

Interconnection of these PCBs for proper functioning of the system is as follows. The RFID reader is connected to the main lock. The RS232-RS485 converter is installed at the central server; it is connected to the PC through the RS232 serial port and serves as a RS485 node on the network side. We have provided three connectors on this converter board that can be used as taps to connect three locks on the bus to the central server. Of course, the converter board can have as many connectors as there are locks in the network without a lot of hardware increment. The main lock should be installed on a wall near the door. The PCB corresponding to the stepper motor is purposefully kept separate from the lock PCB so that it can be installed on the door near the latch.