Mahdere Yoseph Ghebreyesus

RFID Based Coupon Management

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Electronics

Thesis

17 December 2015

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Mahdere Yoseph Ghebreyesus RFID Based Coupon Management 43 pages + 4 appendices 17 December 2015
Degree	BEng
Degree Programme	Bachelor of Engineering
Specialisation option	Electronics
Instructor(s)	Matti Fischer, Principal Lecturer (Project Manager)

The radio frequency identification project was carried out at ITSC plc in Ethiopia. The host who involves in oil and gas distribution use barcode for tracking activity. The drawbacks of barcode for instance lack of memory space, wearable nature, maintenance cost etc. was the initiation to the idea of this project. The aim of this project was to demonstrate the use of RFID over barcode for the purpose of auto-id and user data saving capability.

This paper illustrates the interfacing methodology used to communicate the RFID reader with teensy++ 2.0 embedded development unit and a C# application program to exchange data between the transponder. In addition, proteus simulation showed the functionality of the firmware used in the serial peripheral interface and the liquid crystal display. It also emulates the physical interconnection between teensy and the RFID reader.

Finally, the paper analyzed the result so that, the unique serial id of the transponder gave auto-id capability similar to a barcode. In addition, the read/write access to the transponder also provides the ability to store user defined data. Therefore, it concludes that the use of RFID indeed would satisfy the need of the company who hosted the project.

Keywords	RFID, Coupon Management, Teensy, C#, Barcode

Table of Contents

1 Introduction 1

2 History and Theoretical Background 2

2.1 History of RFID 3
2.2 Operating Principle 3
2.3 Modulation/Demodulation 5
2.3.1 Amplitude Shift Keying (ASK) 5
2.3.2 Subcarrier Load Modulation 6
2.4 Encoding 7
2.4.1 Manchester Encoding 7
2.4.2 Miller Encoding 8
2.5 Transponder 8
2.5.1 Passive Transponderr 9
2.5.2 Active Transponder 9
2.5.3 Read-only Transponder 9
2.5.4 Mid-Range Transponder 9
2.6 Reader 10
2.6.1 Control unit 10
2.6.2 RF Section 10
2.7 Antenna Module 11
2.7.1 Transponder Antenna 11
2.7.2 Reader Antenna 12
2.8 Serial Peripheral Interconnect (SPI) 12

3 Material 14

3.1 MIFARE RFID Reader 14
3.2 ISO/IEC 14443 Type A Card 15
3.3 Teensy 2.0++ Development board 17
3.4 Teensy Program Loader Software 18
3.5 ATMEL Studio 6 18
3.6 Visual C#.Net 18
3.7 LCD 18
3.8 Proteus ISIS 19
3.9 LM3940 19

4 Methodology 21

4.1 Software Simulation 21
4.2 Hardware Implementation 22
4.3 Application Program Interface 23

5 Testing and Result 25

5.1 LCD and SPI Simulation 25
5.2 Hardware Connection and Testing 27
5.3 Communication with C# Application 31

6 Discussion and Conclusion 35

References 37

Appendices
Appendix 1. C# Test Result
Appendix 2 Hardware Test Result
Appendix 3 Transponder Memory Access

Acronyms

ASK	Amplitude Shift keying
CAD	Computer-aided design
CRC	Cyclic redundancy check
C#	C sharp
EAS	Electronic article surveillance
EEPROM	Electronically Erasable Programmable Read Only Memory
ICT	Information and Communication Technology
IFF	Identification, friend, foe
ISP	In system programming
I2C	Inter-Integrated Circuit
LCD	Liquid crystal display
LED	light emitting diode
MISO	Master in slave out
MOSI	Master out slave in
NFC	Near-field communication
OOK	On-off Keying
OOP	Object oriented programming
RFID	Radio Frequency identification
RISC	Reduced Instruction Set Computer
SCLK	Serial clock
SPDR	Serial peripheral data register
SPI	Serial peripheral interface
SRAM	Static Random Access Memory
SS	Serial Select
UART	Universal asynchronous receive transmit

1 Introduction

Radio Frequency Identification (RFID) is a wireless technology used for automatic identification [1] of an object and/or person. Fundamentally, it consists of a transponder, a reader, and antenna. With several applications existing today, the growth and adaptation of the technology leap faster mostly in areas of access control and management applications [2; 3].

This project is part of coupon management system for access control application that has been carried out by ITSC technology support. The company operates in Ethiopia providing Information and communication technology (ICT) service and Support. The idea of this project was introduced by a customer of ITSC, which use coupons that incorporate barcode [4] for tracking purpose. Lack of storage capability, wearable nature of barcodes, reliability and maintenance etc. are few challenges which occur as a result of using coupons. Therefore, this project was initiated to study and show an alternative implementation using RFID to address the problems and provide a better solution.

Basically, the purpose of this paper is to demonstrate the use of RFID in resolving the drawbacks of the barcode. During the project, the communication and data exchange between the transponder and MIFARE RFID reader was illustrated. The teensy development board was used to control the communication protocol whereas the C# application program and a liquid crystal display (LCD) provided a user interface. The goal of this project is to show one possible implementation perspective using RFID technology to provide better management and versatility over the application of barcode based coupon system.

Thus theoretical insight and design methodologies, including hardware and software implementation will be presented. However, it is beyond the scope of this project to consider security and related aspects of the application regardless of its need.

2 History and Theoretical Background

2.1 History of the Company

The host company involves in oil and related product distribution. They use coupons for the purpose of tracking the route of their trucks when traveling across the country. The coupon basically consists of a barcode and name of the holder. Thus, every driver needs to scan their coupon at all stations of the company along the way to indicate the time of presence.

The barcode found at the heart of every coupon often fade out quickly regardless of the quality of printing used. This is because the coupons are usually kept in a pocket that exposed it to rubbing. Therefore, regular replacement is required at the expense of the company. On the other hand, delicate barcode scanners require continuous inspection and maintenance to keep them working.

In addition, the host introduced a new application for customers to collect interest amount. The intention was to attract customers through different bonuses awards like discount coupons, travel tickets etc. based on the amount collected. However, winners were expected to show a pile of receipts that required thorough inspection while claiming their prize. As a result, the company experienced several acts of cheating, theft and falsification although they have managed to attract customers in spite of the tedious manual extra workload.

Therefore, few months after the new application was terminated, the company conducts a proposal for implementing a new system able to integrate the above applications with systematic and better automation capability. ITSC came up with the idea of using smart card [5, 362] that would give all the benefit of automation and better security to satisfy their need. However, the proposal was declined due to budget and complexity reasons.

Later, the idea of using RFID became at the interest of the team and also grabbed the attention of the company. As a result, further studies had been conducted and demonstration was insisted by the host to show the use of RFID in comparison to the existing system that brought to the development of this project.

2.2 History of RFID

The publication in 1948 by Harry Stockman “Communication by Means of Reflected Power” [6] was considered one of the leading innovational concepts to the development and study of RFID. As years go by, several research and innovations were launched including radar that opens a new perspective to the technology. Although, direct use of RFID was not as such defined, related technologies were in progress that implicate its importance.

The first noticeable implementation also known as identification, friend, foe (IFF) [6] was introduced during World War II. This application was used by aircraft industries to identify enemy aircraft from ally counterpart by using unique identification code.

Furthermore, in 1960 different companies started to develop electronic article surveillance (EAS) devices for anti-theft control. The application uses wireless technology to sense the presence of an object. EAS is considered the first commercial application of RFID. Thereafter, the technology passed through different levels of development in providing new areas of application [6;2].

The performance and reliability of RFID invite more industries to incorporate the technology into their activity. For instance, transportation services [7], ticketing [8], logistic and supply chain management [9;10], healthcare [11], manufacturing [12] etc. While new applications are still emerging, like NFC [13] for RFID over mobile phones, the technology forecast to continue growing with additional features.

2.3 Operating Principle of RFID

The fundamental principle for data and power transfer between RFID reader and transponder follows the science of inductive coupling that uses electromagnetic field phenomena. A magnetic field is generated as a result of a moving charge around a conductor loop, in this case, antenna coil. This magnetic field, as it travels through the wire, gradually generates an electric field via the process of induction which further transforms into an electromagnetic field [5,40].

> "The area from the point where the electromagnetic field forms to the antenna are referred as near-field whereas the area from the antenna to the formation of electromagnetic wave that radiate off the antenna is referred as far-field" [5,112].

Inductive coupling is one of the basic operational principles used in the short distance near field RFID system which mostly operates at frequency 100 KHz – 30MHz. The methodology is that first the reader antenna coil generates the high-frequency electromagnetic field. When a transponder antenna become close to the reader, a portion of the field will penetrate the transponder antenna and induce voltage through induction.

When enough voltage is acquired from the reader, it will be rectified and used to power the microchip found on the transponder. This coupling method is mostly used for short distance RFID communication to power passive transponders. Figure 1 shows the basic presentation of an inductive coupling between a reader and transponder antennas.

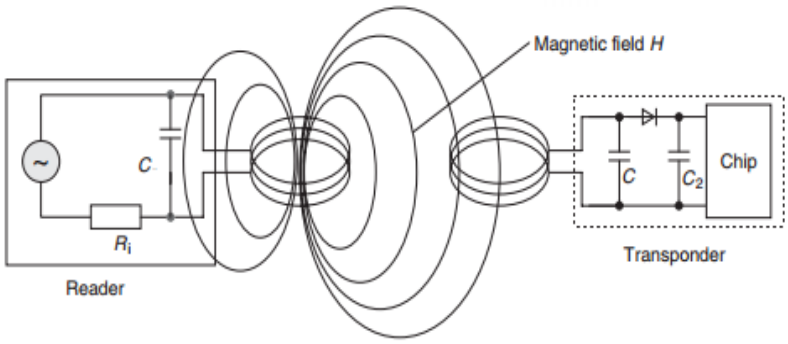


Figure1 Inductively coupled coils between a reader and transponder (Copied from Ilie-Zudor E, Kemeny Z. [5, 41])

As shown in figure 1, the capacitor C found at the reader must go along with the coil inductance of the reader antenna to form a parallel resonance circuit. The frequency at which the circuit resonates should correspond with the transmission frequency of the reader. With proper tuning and due to the characteristics of the parallel resonance circuit, a high current can be generated that further provide enough electromagnetic field strength to power the chip onboard at the transponder.

The same procedure holds by the transponder where the capacitor forms a parallel resonance circuit with the transponder coil inductance. This resonance has to be tuned in collaboration with the frequency used to attain maximum induced voltage at the tran-

sponder coil. Power transfer between the two antenna coils has to be optimum as well as the number of turn of winding that affect the coil inductance. The alignment, distance, and area of the coils must also be considered. Typically, for the frequency at 13.56MHz, 3-10 windings could be used [5, 41].

2.4 Modulation/Demodulation

This is the process of adding information onto a high-frequency carrier signal and vice versa. The use of High-frequency carrier is due to the size of the antenna which is one-quarter the wavelength of the propagating signal. Thus, the higher the frequency is the smaller the size and power use for trans-reception. Modulation can be done by altering the amplitude, frequency or phase of the carrier in accordance with the information signal.

2.4.1 Amplitude Shift Keying (ASK)

This is a digital modulation technique that uses analog sinusoid carrier signal modulated by a digital binary symbol. In this method, the amplitude of the carrier is changing in relation to the digital symbol. For instance, a bit value 1 can be assigned to one amplitude level of the carrier and bit 0 to another as shown in figure 2.

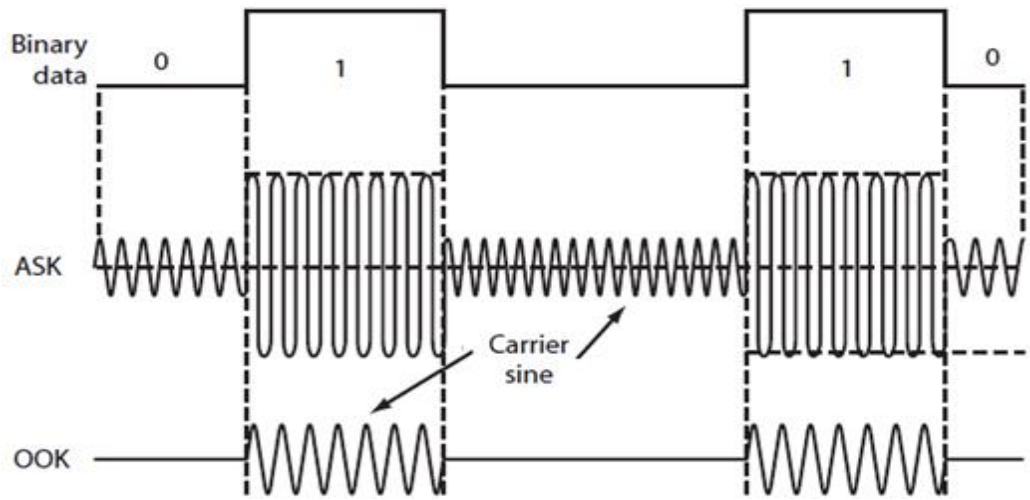


Figure 2 Amplitude Modulation and OOK modulation (Copied from electronic design Understanding Modern Digital Modulation Techniques [14])

A type of ASK modulation that fixes one of its amplitude level to zero and use the other to modulate the carrier signal is called 100% ASK or On-Off keying (OOK) modulation as in the second part of figure 2. It is the simplest ASK modulation used mostly in fiber optics. The drawback of ASK modulation is bandwidth inefficiency. Thus, it is mostly used for low cost and simple applications [5,182;15,126].

2.4.2 Subcarrier Load Modulation

This modulation technique is used mostly by the transponder to transfer data to the reader. First the transponders antenna needs to be inductively coupled with the reader then, the resonance frequency of the transponder has to be adjusted to the transmission frequency of the reader.

Once the above setup was completed, power would be acquired by the transponder due to the magnetic field effect. In return, the transponder leaves transformed impedance back on the reader. Switching a load resistor at the transponder would create a change of impedance that further generates a corresponding change of voltage at the reader. This change in voltage has the effect of amplitude modulation at the reader antenna. Therefore, synchronizing data to the switching time of the load resistor would, in fact, enable data transmission from the transponder back to the reader, and the process is called load modulation.

Demodulation is the process of extracting data out of the received signal. During this process, voltage rectification is required to power the chip on board. Since the variation of voltage is very small due to poor modulation, both detection, as well as retrieval of the information signal, is difficult at the reader even though voltage step-up is applied by the resonance circuit.

To overcome this problem, the transponder uses an additional resistor to switch at high frequency creating additional subcarrier less than the transmission frequency of the reader. Now, the transponder could apply ASK to modulate data onto the subcarrier frequency. This process is called load modulation with the subcarrier. Band pass is used to separate the sidebands which were created around the subcarrier frequency. There might also be more filters applied to smooth out ripples and unwanted frequency components. After that, the signal need to be amplified before and after demodulation to increase the strength [5, 41].

2.5 Encoding

This is a process of converting data bits into an actual message signal where it can be suitable for transmission through the physical medium. Encoding could be applied either to a single bit or group of bits called symbols. The rate at which the symbols are encoded to the signal level defines the baud. Decoding, on the other hand, is the inverse process for retrieving the data bits from message signal [15].

2.5.1 Manchester Encoding

This encoding method is applied mostly when data transmission is carried out from the transponder to the reader. It uses transition of the level at the middle of every period to represent a single data bit. For instance, a low to high transition can be assigned to data bit one and the reverse to data bit zero as shown in figure 3. Therefore, the given bit or symbol could be represented into the signal by switching the corresponding level at the middle of every clock period [16].

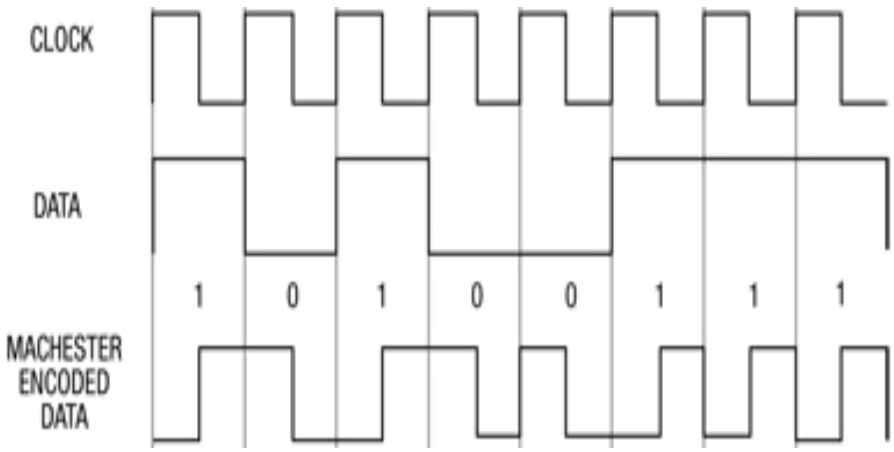


Figure 3 Manchester Encoding (Copied from digi-key, Considerations for Sending Data over a Wireless Link [17])

The transition at the middle of a period provides self-clocking in addition to the actual data transmission as shown in figure 3. It also provides better security and error detection. The main drawback of Manchester encoding is the use of high bandwidth as a result of the need for more transmission bits than the original data. [16].

2.5.2 Miller Encoding

This encoding method, on the other hand, is applied mostly when the communication is from the reader to the transponder. Similar to the method used in Manchester encoding, miller also applies transition at the middle of a period as shown in figure 4 [18, 16].

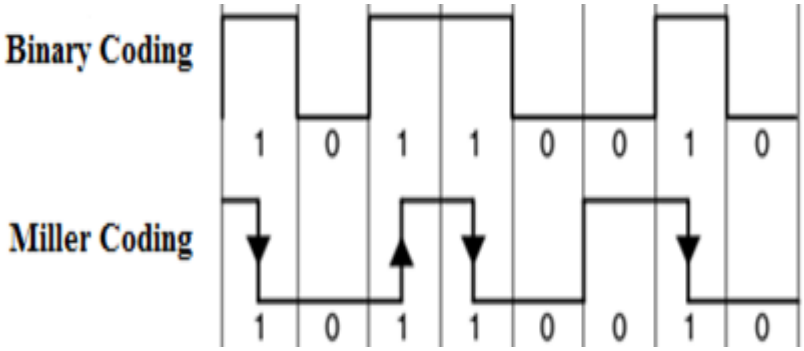


Figure 4 Miller Encoding (Copied from infosec institute, Introduction to RFID Security [19])

The transition at the middle occurs only when the value of the data bit is one as shown in figure 4. This phenomena used by miller encoding provides better tolerance to noise interference [18].

2.6 Transponder

These are primary elements of a RFID system that consists of an antenna and microchip. The purpose of the chip on the board is to provide memory space used for data storage and to hold the unique identification code. Manchester encoding used by the reader provides the transponder the clock signal necessary for the microchip to operate.

The RF section, on the other hand, provides modulation/demodulation and encoding/decoding capability during data communication [15; 5, 283]. It also performs load modulation to accomplish data transmission back to the reader. Different transponder types may incorporate additional logics like cryptology and other. This section will illustrate different types of transponders categorized according to the power source they use and memory capacity.

2.6.1 Passive Transponder

These transponder types do not have any power supply and thus, they are mostly used for short distance application. They acquire the energy to operate from the reader through coupling when they are at close proximity to the reader antenna.

Once enough energy is acquired to the power the chip, load modulation takes place (see 2.3) to transmit the data including the unique code back to the reader. This means the reader send data and energy to the transponder. In return, the transponder uses this energy to transmit data back to the reader.

2.6.2 Active Transponder

This transponder types, on the other hand, requires an external energy source like a battery to power the chip and thus, they are used for longer distance compared to passive transponders. Although, energy is not required from the reader, active transponders still lack generating a high-frequency signal. Therefore, they also apply load modulation technique to transmit data back to the reader without the need for an external source. The purpose of having an external source is to accommodate longer reading range relatively.

2.6.3 Read-only Transponder

As the name says for itself, this transponder types does not provide any data storage rather they hold only the unique serial code. These transponders can have longer reading range as a result of low power consumption of the chip. They are mostly used in applications similar to a barcode.

2.6.4 Mid-Range Transponder

This transponder type provides memory space up to 100kbyte used for data storage. They also provide anti-collision and authentication logics. Passive transponders use EEPROM to keep the content without the need for a continuous power source. However, active transponders use SRAM that requires continuous power to keep its content.

2.7 Reader

A reader is another primary element of RFID system that facilitates a communication interface used for data exchange between the transponder. Therefore, the firmware made for a reader is used as an intermediary interface that handles the work necessary for data transfer between the application software and the transponder. Different readers may include other logics in addition to the control units and RF section [15; 5].

2.7.1 Control unit

This section consists of a processor unit mostly a microcontroller responsible for handling digital systems and communication interfaces like SPI, UART, I2C etc. these interfaces provide a communication channel with external devices. A host processor mostly a computer holds the middleware that read/write data between the transponder via the reader.

The control unit in conjunction with its firmware is responsible for protocols decode/encode and signal processing for the RF section during communication. In addition, it can also provide services like authentication, anti-collision used during multiple accesses, encryption, and other security features [5, 323].

2.7.2 RF Section

This section, on the other hand, provides the transmission and reception of data through radio communication link between the reader and transponder. It consists of a transmitter, receiver, and coupling antenna sections. A crystal oscillator is used to generate a high-frequency signal used by the transmitter and receiver [5, 317].

2.7.2.1 Transmitter

Once the data signal is encoded at the control unit, ASK (see 2.3) takes place. It uses the encoded data signal and the high-frequency carrier signal generated by the oscillator. The resulting modulated signal will further be fed to the power amplifier to enhance the strength of the transmitted signal necessary during propagation.

The antenna needs to be tuned properly and matched for maximum power transfer. This is used to pass the transmitted signal that comes out of the amplifier output to be radiated by the antenna efficiently with no or minimum power loss [15,142]

2.7.2.2 Receiver

This consists of an amplifier and demodulator used for data reception. Once a signal is received by the antenna, bandpass filter eliminates the high-frequency components out of the received signal leaving only a band of frequency that corresponds to the data.

Obviously, the incoming signal is always weak. Therefore, an amplifier is used to increase the amplitude for demodulation. The data signal that was extracted would further pass through another filter to smooth out ripples and finally fed to the control unit for decoding and processing [15,143].

2.8 Antenna Module

This is another part of RFID system used to radiate/receive an electromagnetic wave. It is used by the transmitter and receiver section of the reader and transponder. Note that the transmitter and receiver use a single antenna properly tuned to minimize energy loss and maximize power transfer [15].

The operating frequency in HF RFID is 13.56MHz. Therefore, it is impractical to build an antenna to accommodate such wavelength. An alternative way of using a small closed loop antenna becomes a reality with relatively compromising the bandwidth, range etc.

2.8.1 Transponder Antenna

The purpose of transponder antenna is not only for data communication but also for collecting the energy necessary to power the chip on board. The implementation of small loop antenna becomes practical considering the available space.

Therefore, a coil of wire is placed in a circular structure having specific turns and radius. When current passes through the wire, magnetic field starts to build up around the

coil. The field intensity varies with the distance. The closer the field is the higher the intensity and that limits the reading distance of a transponder to span only a few centimeters long.

On the other hand, the voltage induced depends on the orientation of the coil. Therefore, careful optimization must be taken in addition to the inductance of the coil that will increase the quality factor of the antenna and the induced voltage [15,14].

Tuning is achieved by placing a capacitor making a parallel resonance circuit that gives maximum voltage at the port. Impedance matching is also accomplished by changing the alignment of the coil or by adding a reactive component used to match the impedance with the input of the chip.

2.8.2 Reader Antenna

This antenna is used for data communication and power transfer for the transponder. Since dimension is not as critical as it is for the transponder, a dipole type antenna can also be applied other than the loop antenna. The fundamental aspect here is the reading/writing rate and maximizing the distance. Reading distance could be optimized by increasing the antenna as well as the coil size, and reducing the coil radius which will create a strong field at the antenna. Having the requirement for more power for writing than reading, the inductance, polarization and presence of obstacle like metal would still affect the reading/writing distance including the range.

Although, there is relative flexibility to increase the antenna size to maximize the reading distance, but still it has to be kept to an acceptable range to avoid interference. The inductive nature of the loop antenna coil needs to be properly matched to the capacitive impedance to avoid reduction of power transfer [15, 85].

2.9 Serial Peripheral Interconnect (SPI)

This is a full-duplex synchronous serial communication type that uses separate lines for data and clock signals. There is Master and Slave mode of operation where master always initiates communication. The four lines used for SPI communication are illustrated as follows

- Master-In-Slave-Out (MISO): this line is connected as an input to master and an output to slave device used for unidirectional serial data transmission with MSB first.

- Master-Out-Slave-In (MOSI): this line is connected as an output to master and as an input to slave device and similarly used for unidirectional serial data transmission with MSB first.

- Serial Clock (SCLK): this line is used for synchronization of data transmission that exists between MISO and MOSI lines. It is configured as an input to the slave and the master always initiates the clock.

- Slave Select (SS): this is an input line that is used when multiple slave devices are connected and used for making a selection between the slaves.

3 Materials used in the Project

3.1 MIFARE RFID Reader

This is one basic RFID system from NXP that incorporate MFRC522 highly integrated chip as shown in figure 5. The chip handles the low-level operation needed for communication between the reader and transponder. The radio module and the transmitter/receiver section of the chip handle modulation/demodulation and encoding/decoding of signals providing reliable communication with transponders [20].

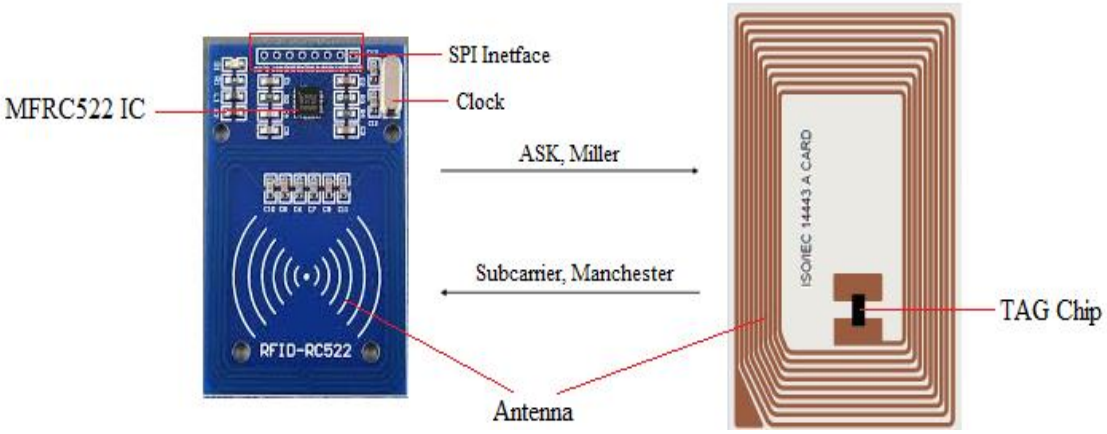


Figure 5 MIFARE RFID modules (modified from NXP MFRC522 Standard 3V MIFARE reader solution [20])

Input and output data stream buffering together with framing and error checking logics are also provided by the digital section of the chip in addition to SPI communication interfaces as shown in figure 5 above. The chip operates at 13.56MHz frequency and 2.5 - 3.3 V power source. The programmable pins enable the chip to be controlled through its internal registers [20].

The clock provides accurate timing including synchronized communication. When an external clock is used, clock jitter that is the distance of shifting of the pulse edge due to noise or other disturbance and duty cycle that is the percentage of occurrence of a positive level per one period must be considered carefully.

3.2 ISO/IEC 14443 Type A Card

MF1ICS50 a MIFARE transponder from NXP is used during the project. This is a passive type and mid-range transponder. It has 106Kbits/s data transfer speed. The transponder consists of 1Kbyte EEPROM organized in 4 blocks of 16 bytes each [21] as shown in figure 6.

		Byte Number within a Block																
Sector	**Block**	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	**Description**
15	3	Key A						Access Bits				Key B						Sector Trailer 15
	2																	Data
	1																	Data
	0																	Data
14	3	Key A						Access Bits				Key B						Sector Trailer 14
	2																	Data
	1																	Data
	0																	Data
:	:																	
:	:																	
0	3	Key A						Access Bits				Key B						Sector Trailer 0
	2																	Data
	1																	Data
	0																	Manufacturer Block

Figure 6 Memory Structure (Copied from NXP, MF1ICS50 Functional specifications [21])

A given transponder must be selected and authenticated before any memory operation is carried out. The manufacturer block found at the first sector of block 0 as shown in figure 6 are used to hold the serial id and other manufacturer data.

Since this block is read only, the data cannot be updated which makes the transponder unique. Sector Trailer found at the third block of every sector is grouped into access bit that defines the operation on the block and key A or B which defines the location. The sector trailer reserves 3 bits used for configuration of the data block in addition to the secret key used for the key type [21].

Data block includes block 0-2 except for sector 0 which are only block 1 and 2. these are used to store user defined data. The access bit configures the data block either as read/write or value block that consists of various operations like increment, decrement, restore etc. as shown in table 1.

Table 1 Transponder memory access operation (Copied from NXP, MF1ICS50 Functional specifications [21])

Operation	Description	Valid for Block Type
Read	reads one memory block	read/write, value and sector trailer
Write	writes one memory block	read/write, value and sector trailer
Increment	increments the contents of a block and stores the result in the internal data register	value
Decrement	decrements the contents of a block and stores the result in the internal data register	value
Transfer	writes the contents of the internal data register to a block	value
Restore	reads the contents of a block into the internal data register	value

The first three bits of access bit are used to control the operation of memory access using the secret keys A and B. This means there are 8 different combinations of using key A and/or B to perform the operations mentioned in table 1 in each block.

The first table shown in appendix 3 illustrates how the access bits allow or deny read/write access to key A and B. the second table shown in appendix 3, defines the access condition for the data block. Only reading and writing operation is allowed to the read/write block depending on the combination of the access bit. For instance, for access bit 100, writing is allowed via key B but reading could be done through key A or B. On the other hand, the value block allows additional operations listed in table 3 to be performed. For instance, access bit 110, only increment and writes operations are allowed via key B, but the rest could be done through key A or B [21].

The RF section takes care of modulation/demodulation, rectification of induced voltage used to power the chip, clock generator, reset and voltage regulation. Other sections like anti-collision, crypto for security, CRC and error check may also reside on the chip.

3.3 Teensy 2.0++ Development board

This is an embedded development board that consists of an AT90USB1286 AVR microcontroller unit. The board is organized to be used in a solderless breadboard. As shown in figure 7, the mini-B USB found onboard communicates at 12Mbit/s transfer rate. It is mainly used for In-System-Programming (ISP) the chip using the built in bootloader [22].

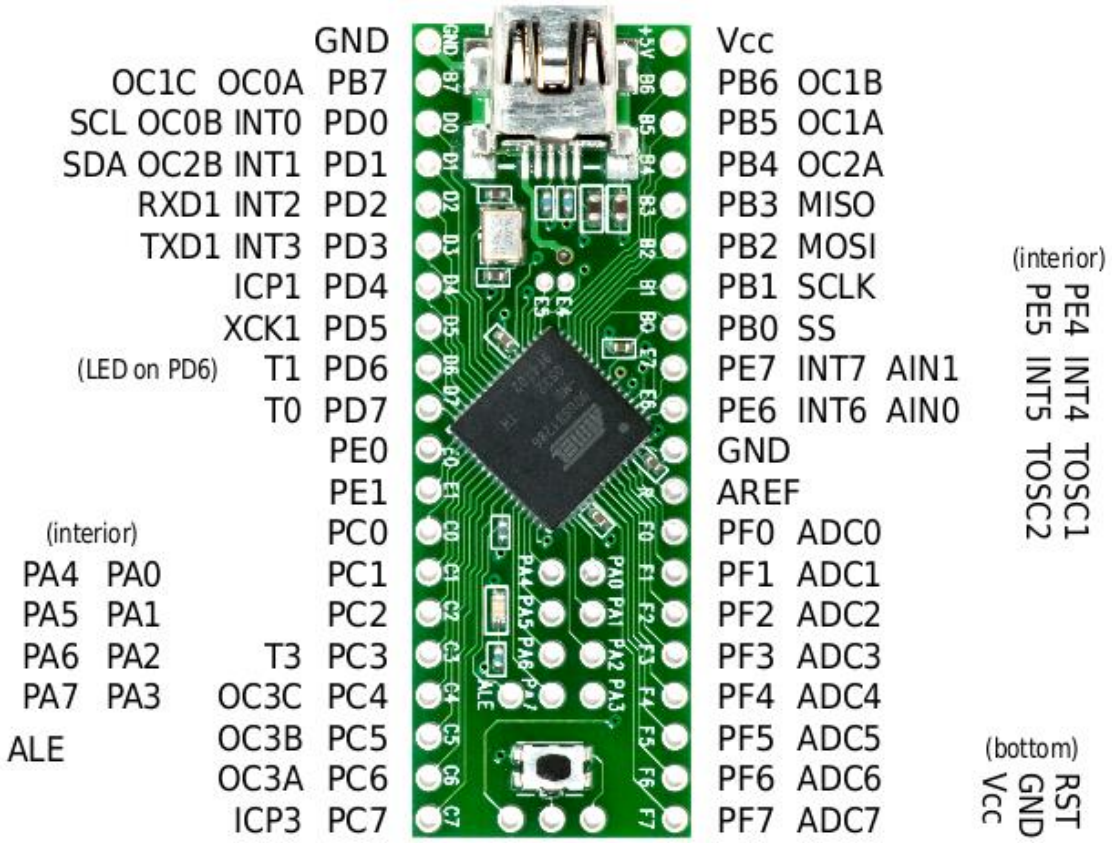


Figure 7 Pin layout of Teensy++ 2.0 development board (Copied from PJRC.com, [22])

The low power 8-bit AVR microcontroller shown in figure 7 operates at 16MHz frequency. It consists of reduced instruction set computing (RISC) that uses a single clock cycle per instructions. There are 135 instructions with 32 x 8 general purpose register. The 128Kbytes of flash memory can be used for application programs. Different systems including SPI, PWM, UART, ADC, Timer, Counter etc. are integrated into the chip [22]. In addition, there are 6 x 8-bit general purpose I/O ports used for interfacing LCD and other external devices.

3.4 Teensy Program Loader Software

This software is used to program teensy using the button on the board. When this button is pressed, the program counter will jump to address 0x1FC00 to locate the Halfkay bootloader. When the bootloader is executed, the chip will be set to programming mode and no data communication is possible. In the meantime, it activates the loader Software so that the chip would be programmed with a new source code. After the program is loaded, the software reboots the chip in order to release running the bootloader and start executing the new code.

Since there is no hard reset available on the board, it is always necessary to initiate the software whenever rebooting is required. After reboot, the usual serial communication starts to operate normally.

3.5 ATMEL Studio 6

This is a software program used for developing an application for AVR core microcontroller systems. It provides an integrated environment on a Microsoft.net platform to develop, debug and compile embedded application written either in C/C++ or assembly programming languages for AVR microcontroller units.

3.6 Visual C#.Net

This is a Microsoft software that incorporates an object-oriented programming (OOP) structure. OOP provide modular structures that perform an action through the interaction of objects where the logic behind the action is hidden to the used. This programming approach provides better security and management to the whole development cycle [23].

3.7 LCD

The 16x2 character type LCD module is used during this project. It incorporates the low power Hitachi HD44780 controller operates at +5v source. It also supports both 4-bit and 8-bit interfacing mode with 5x8 or 5x10 dot character font. As shown in table 2, the

LCD has at most 7 data and 3 control lines required in addition to the contrast and source pins.

Table 2 Pin description for 16x2 characters LCD

Interfacing Port	Description
D0-D7	Used for data communication as an 8-bit or 4-bit mode
Enable (E)	Used to transfer data
Read/Write (R/W)	Used to select between read or write
Register select (RS)	Used to select between a command or data
VEE	Used for LCD contrast adjustment

The 4-bit mode configuration requires only 4 data lines interfaced either to the most or least significant port pins of the processor. As shown in table 2, the EN control line is used to transfer whatever data is available at the data port to and from the LCD. The RW control line is used to read/write data to/from the data port. The RS control line, on the other hand, is used to choose between data and command. The contrast adjustment pin needs to be connected to a variable resistor in between the source and ground.

3.8 Proteus ISIS

This is a CAD software program used for virtual modeling and circuit simulation for microcontroller based design. The program is used to test a design prior to its practical implementation giving insight look to its application. It incorporates all the necessary modules including debugging tools.

3.9 LM3940

This is a voltage converter and regulator chip used to convert basically 5V down to 3.3V supply. Although, it can accommodate a range of input between 4.5 - 5.5V, the output voltage is always regulated to 3.3v [24].

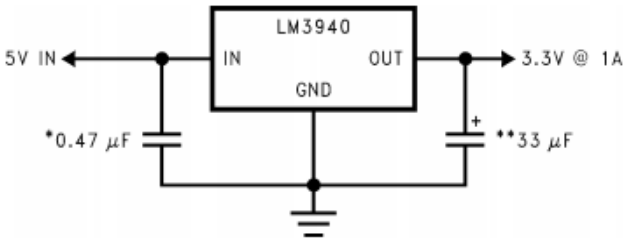


Figure 8 layout diagrams for LM3940 voltage regulator chip (Copied from Texas Instrument, LM3940 1-A Low-Dropout Regulator [24])

Having protection against excess temperature, it can hold its 3.3-V output in regulation to its input. As shown in figure 8, the output capacitor is used to maintain the stability that should be considered carefully in relation to the output current and temperature.

4 Methodology used during the Project

4.1 Software Simulation

In this section, a software simulation was carried out for the LCD and SPI modules using proteus ISIS program. The simulation layout was constructed in such a way that two AT90USB1286 processors were connected using the SPI master/slave protocol.

The slave processor indicates the connection between the LCD and Teensy board. Similarly, the master/slave connection shows the hardware communication between a teensy and RFID reader that incorporates SPI interface. The layout connection was made as shown in table 3,

Table 3 Pin configuration between the LCD, Teensy 2.0++ board, and RFID

LCD 4-Bit Mode	Teensy 2.0++	RFID Module
D7	C7	
D6	C6	
D5	C5	
D4	C4	
EN	D7	
RW	GND	
RS	D5	
	B3	MISO
	B2	MOSI
	B1	SCK
	B0	SDA

The LCD was configured in the 4-bit mode as shown in table 3. The uppermost pins of the data port were used in this case. To monitor the data exchange, an SPI analyzer was connected between the MISO and MOSI pins. The two lines are the data channels whereas the SCK was just a clock signal. The RW control pin of the LCD was tied to ground disabling reading capability.

After the interconnection was completed, the main program at the slave started initializing the LCD. This includes mainly configuring the communication in 4-bit mode, font setting etc. the LCD was turned on ready for operation after buffer was cleared and the function set was initialized. During communication, the function that writes into the LCD always needs to adjust the data to corresponding to the position of the data port.

In the meantime, the SPI initialization subroutine has configured each device as master and slave. The slave has to wait while the master generates the clock signal and initiates the communication. After the initialization was completed, data exchange started using the corresponding read/write subroutines found in each device.

4.2 Hardware Implementation

In this section, the hardware implementation between the RFID reader and teensy board was carried out. The interconnection was made as shown in table1. The teensy board was powered from USB port and provides a source to the LCD and RFID reader. Therefore, no external source was needed. The block diagram shown in figure 9 illustrates the structure of the whole system.

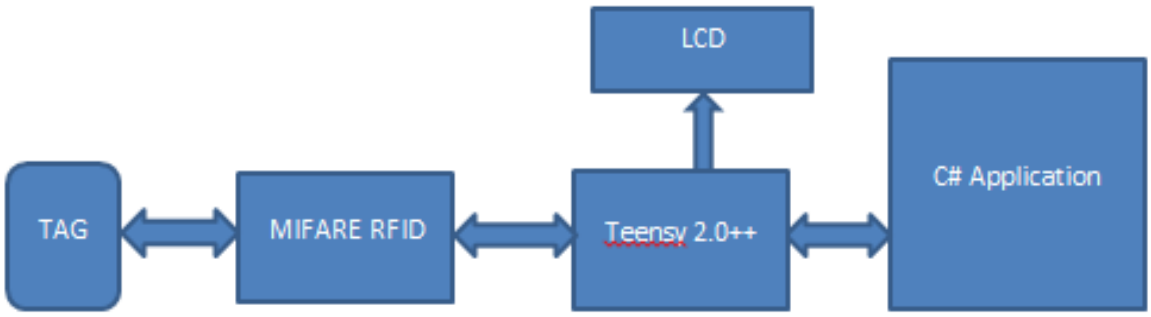


Figure 9 Block Diagram to show the communication between sub-systems

The teensy has bi-directional communication with all except the LCD that was configured only for writing although it was possible to read as well as shown in figure 9. Once the physical wiring was completed and all modules got the necessary power to operate, the next step was programming firmware into the microcontroller.

The firmware found on teensy initialized the LCD and SPI functions first. After that, a series of commands was executed to check the connection between RFID reader to teensy and USB communications to the computer. In addition, the application software was verified to initiate the communication with teensy.

After the above procedure was completed, the reader antenna was activated so that it starts to scan the interrupt vector of the reader. Whenever a transponder was found at the vicinity of the antenna, the interrupt would be triggered to detect the transponder. Once the transponder was verified, the chip would read the serial id. Every data should be encoded into a string array so that both the application program and LCD could be able to display the result.

The read operation was done at the transponder memory sector 4 block 2 using key A. The secret key was just 0x00. After the data was read and encoded, it was sent to the LCD in hexadecimal format. Writing data, on the other hand, requires organizing the outgoing data into an array. After that, writing the array into the transponder memory at sector 4 was carried out. The result was verified after reading the same memory space. Always necessary to check if the transponder stays intact to the reader antenna before attempting to read or write.

4.3 Application Program Interface

The USB interface found on teensy was initially used for programming the board. Therefore, data communication via the same port was not possible until a USB-Serial library file [18] was set up. This library file was configured into the firmware and initializes the USB port to be recognized by the computer as a standard COM port. This would further enable communication as a serial interface.

Once the communication was set up, the application software would recognize the COM port. After that, the serial port setting was configured and the connection was initiated. After the connection was established, the event handler used in the software scans the corresponding port for an incoming data. Whenever a transponder was detected, it would read the type of card and show the information into the label box. Following the detection and verification of the card, all other functions like reading the serial id, read/write user data could be done by sending the corresponding request to teensy.

Reading user data as well as encoding was done by the same function found on teensy. The only difference was that the output was sent to the USB in addition to the LCD. Similarly writing data onto the transponder required the outgoing data, in this case, the interest amount has to be saved into an integer array. Then, the array was sent to teensy via USB and writing to the same memory was carried out by the same function. Verification was also done by reading the same location and sending the result back to the application as well as to the LCD.

5 Testing and Result

5.1 LCD and SPI Simulation

Communication to the LCD required alignment of data or command to the position of the data port. This was because the LCD was configured to operate in 4-bit mode. As shown in listing 1, the data/command send out to the LCD, was shifted and saved into the data port before transferred to the pin.

```
for (i = 7; i >= 4; i--)
{
    LCD_Port &= ~(1 << i);
    LCD_Port |= (Tx_value & 1 << i);
}
LCD_CONTROL_PORT |= (1<<LCD_EN);
_delay_us(2);
LCD_CONTROL_POR
```

Listing 1 Save data into the port and transfer to LCD for C code

First the data was grouped into most significant bits (MSB) and least significant bits (LSB). After it was organized as MSB first structure, shifting was carried out using the code shown in listing 1. The shifting aligned the data to be sent via the data port. Once the data port was ready, toggling the EN control pin from state high to low would transfer the data to the LCD.

In the SPI interface used in the simulation, each processor was configured properly to be identified as master and slave during communication in addition to the actual pin connection. As shown in listing 2, the first line of code was used for master configuration and the second code for the slave.

```
SPI_DDR = (1<<SPI_MOSI)|(1<<SPI_SCK)|(1<<SPI_SS);
SPI_DDR = (1<<SPI_MISO);
```

Listing 2. SPI communication for master/slave setup for C code

As shown in listing 2, the master has generated the clock and also initiated the communication to the slave. Optionally the serial select pin could be used to select between multiple slaves. Once the above setup was completed, the slave adjusted its own clock and started to exchange data. Both reading and writing operations were done through the serial peripheral data register (SPDR).

The simulation window shown in figure 10 illustrates the interconnection between the master and slave processors as well as the LCD setup. The pinout (see 4.1) used in the simulation was to check for the SPI sub-routine which was used by teensy to communicate with the RFID reader. In the same way, it checked also the LCD communication for its proper operation.

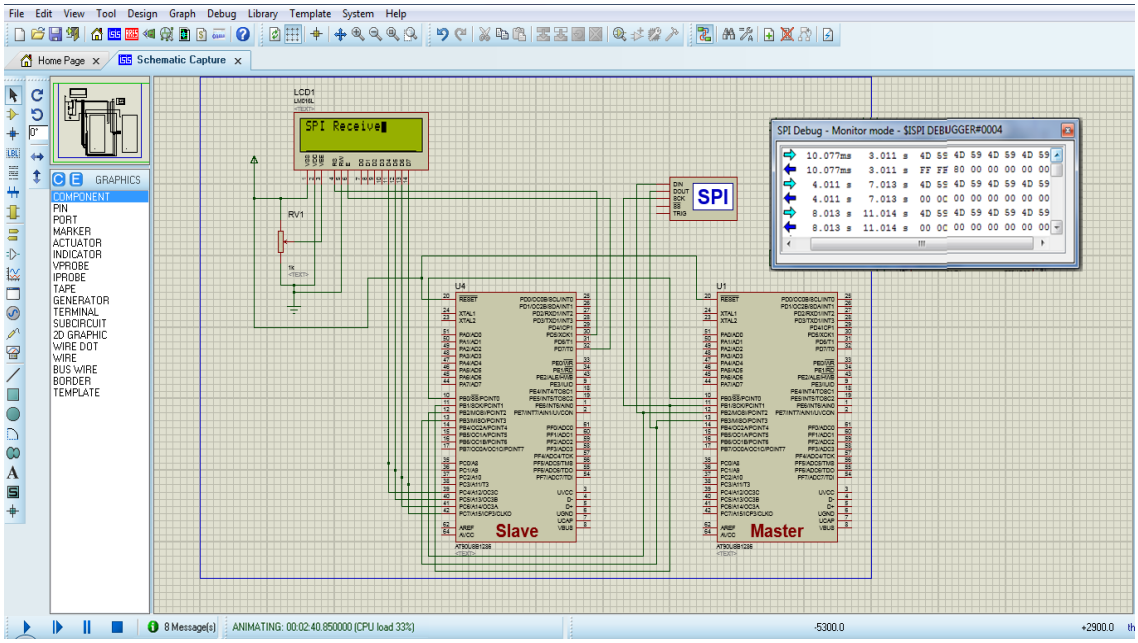


Figure 10 SPI Master/slave simulation test result

The SPI analyzer window shows, the master initiated the communication and continue sending the same data to the slave as it was set in the sample program code. The LCD, on the other hand, confirmed the received text as shown in figure 10.

Therefore, the simulation result clearly illustrates the functionality of the firmware used by the LCD and SPI. Also, It showed that the interconnection was valid for the actual hardware implementation.

5.2 Hardware Connection and Testing

The physical interconnection used during hardware implementation referred the simulation layout shown in the previous section. The master device, in this case, would be teensy which the LCD was connected to and the slave would be the RFID reader. Unlike the teensy and LCD module, the RFID reader operates on 3.3v. Therefore, LM3940 chip was applied to convert a 5v source from teensy and supplied the reader with 3.3v as shown in figure 11.

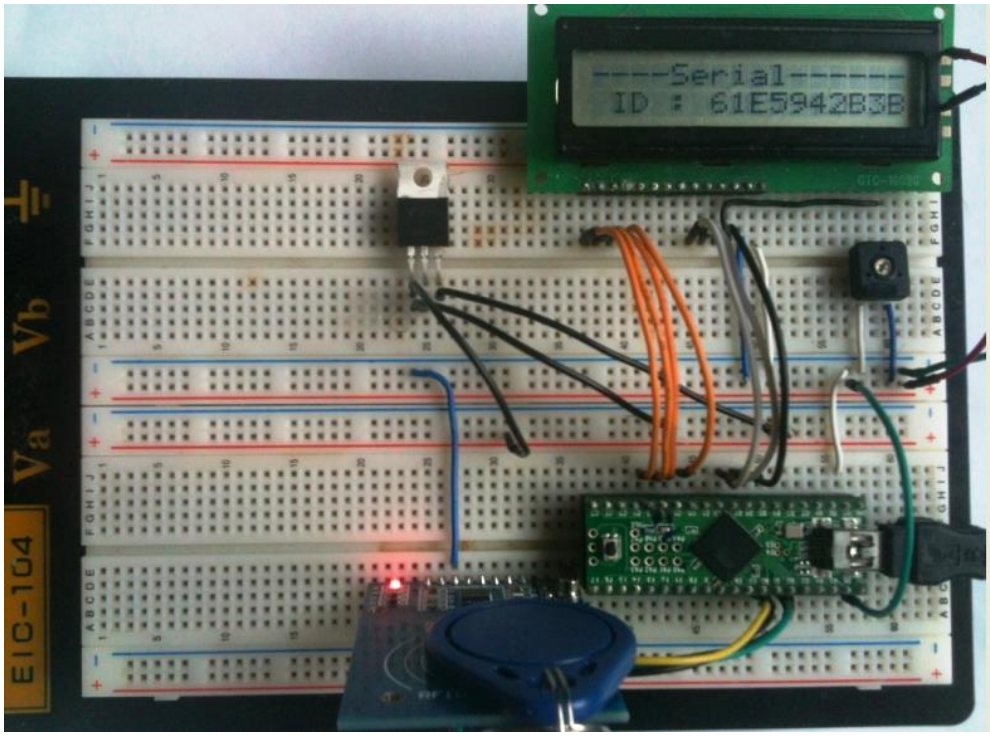


Figure 11 Hardware interconnection

The LCD showed the serial id that was read from the transponder found at the top of the reader as shown in figure 11. As illustrated in the previous section (see 4.2), the main firmware executed a series of checking and initialization subroutines. After the SPI (see 5.1) was initialized, the firmware continuously scans the interface until the existence of the reader was verified as shown in Appendix 2. The C code shown in listing 3 illustrates the operation as well as how the version and type of the reader were verified.

```
byte = mfrc522_read(VersionReg);
if(byte == 0x92) LCDWriteStringXY(4,1,"Detected");
else if(byte == 0x91 || byte==0x90)
LCDWriteStringXY(1,0,"MIFARE RC522v1");
Else LCDWriteStringXY(0,0,"No reader found");
```

Listing 3. Checking the availability of a RFID reader for C code

The internal register ‘`VersionReg`’ shown in listing 3 was continuously scanned until the reader was found. The return value verified not only the existence of the reader connected at the SPI port but also extracted the version and type of the reader as shown in appendix 2.

Once the reader was setup, the USB interface used to activate the communication to the computer was checked as shown in listing 4. This function could be disabled if no communication was needed with the computer. This means the system would work without using the application software. This gave the flexibility for the system to operate alone. The LCD was used to monitor the status.

```
if (!usb_configured())
{
      LCDWriteStringXY(5,0,"Error!")
      LCDWriteStringXY(1,1,"Connect USB");
      while (!usb_configured);
}
```

Listing 4 Checking USB connection between teensy and computer

The ‘`usb_configured`’ shown in listing 4, was set after the computer has recognized and configured the COM port where the teensy was connected to. Therefore, checking this value confirmed the communication channel to the computer was active as shown in appendix 2. Following the USB, the teensy firmware has synchronized itself to the application software found on the computer. The teensy would wait scanning the port for a connection request from the program. Unless a connection was established, no further operations could be carried out as shown in appendix 2.

At this point, the application software was responsible for setting up the necessary communication setting and initialized a connection. Listing 5 illustrates the C code used to check if a connection was established or not.

```
if (!(usb_serial_get_control() & USB_SERIAL_DTR))
{
      LCDWriteStringXY(5,0,"Error!");;
      LCDWriteStringXY(1,1,"START SOFTWARE");
      while (!(usb_serial_get_control() & USB_SERIAL_DTR));
}
```

Listing 5 Checking connection to the application program

The ‘`USB_SERIAL_DTR`’ shown in listing 5 was set when a connection was established by the application program. Therefore, continuously checking this register confirmed the firmware that the application program has started running and a connection was activated. This procedure gave synchronization between the two applications.

Once all connections and checking were verified, initializing the internal registers used by the RFID reader was carried out. This initialization required sending a series of configuration commands including, setting the mode 0x12, prescale register 0x2B etc. Using command 0x26 would wait for an interrupt that was initiated whenever a transponder was found at the reading range of the antenna. The serial id shown in figure 9 could be extracted from the transponder memory following the detection of the transponder using the code shown in listing 6.

```
serial_out[0] = 0x93;
serial_out[0] = 0x20;
status= mfrc522_to_card(0x0c,serial_out,2,serial_out,&unlen);
```

Listing 6. C code for reading the serial id (Copied from Eka Puji Widiyanto AVR GCC library rc522 [25])

The ‘`serial_out`’ array shown in listing 6 holds the serial id that was read from the transponder. The status variable was used to return an error if the operation fails. Once the serial id was verified, the LCD would display the result.

Read and write operations was performed to a specific block of memory space. In this project, section four was used to store data as illustrated in listing 7

```
recvData[0] = 0x30;
recvData[0] = 4;
mfrc522_calculateCRC(recvData,2,&recvData[2]);
status = mfrc522_to_card(0x0c,recvData,4,recvData,&unLen);
```

Listing 7 C code for reading user data (Copied from Eka Puji Widiyanto AVR GCC library rc522 [25])

The ‘recvData’ array carried the user data that was read from the transponder memory space as shown in listing 7. The status was used to return an error value if the operation fails. The error that occurred during reading was mostly if the transponder was removed or away from the reading distance of the antenna.

Writing user data was done on the fourth block of the transponder memory. First the data was saved into an array. Then using the code shown in listing 8, it was sent to the transponder. Here the status was used to check for the operation. Writing could also fail for the same reason as for reading.

```
for (i=0; i<16; i++)
{
   Buff[i] = *(writeData+i);
}
Mfrc522_calculateCRC(buff,16,&buff[16]);
Status = mfrc522_to_card(0x0c,buff,18,buff,&recvBits);
```

Listing 8 C code for writing data (Copied from Eka Puji Widiyanto AVR GCC library rc522 [25])

The whole read/write operation shown in appendix 2 illustrates that the existing value which was read from the transponder was 0009. After the data was updated to 0014, just simply adding 5 to the current value, it was Witten back to the same memory location. Reading back confirmed the data was indeed written as shown in appendix 2.

Therefore, fetching the serial id as well as reading/writing capability confirmed the communication and exchange of data to the transponder was possible.

5.3 Communication with C# Application

At this point, the hardware section of the project has accomplished the communication as well as the exchange of data between the transponder. Therefore, the next phase was to create the link between the computer and teensy board so that the application program could be able to exchange the same information that was used by the LCD.

The application software started with the authentication window, as shown in appendix 1. Commonly, authentication was done using password and username. After an authorized used was validated, the serial setting window configured the COM port (see 4.3) and other communication parameters as shown in figure 12.

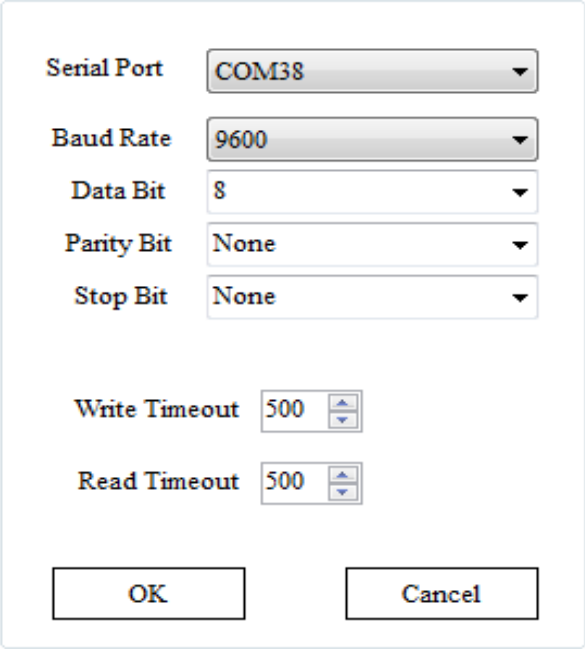


Figure 12 Serial port setting window

The baud rate shown in figure 12 determines the communication speed. It has to be configured to matches with the speed used by teeny. Otherwise, data losses could occur as a result of incompatibility. At least, the baud rate and data port have to be configured and the rest could be left to the default value as it appears.

After the setting was completed, the main window appeared as shown in appendix 1. This window provides all the menus and functionalities used by the application in graphical user interface (GUI). A serial object was created after the connection was established. By using the object, all functionalities related to the serial port including a request for reading the serial id or reading/writing user data would be accomplished as shown in listing 9.

First, the serial object opens the COM port and established a connection. If the connection fails, it was likely that some settings were wrong or missing. Therefore, it was possible to go back via the menu as shown in appendix 1 and re-adjust the setting. After the communication was successful, the main window showed the information regarding the parameters used in the current connection

```
Private void DataRecievedHandler(object sender, SerialData-
RecievedEventArgs e)
{
  SerialPort sp = (SerialPort)sender;
  Try
  {
     Indate += sp.ReadExisting();
     If (indata.Contains('#'))
     {
        String[] Read_data = indata.split('#');
        This.Invoke(this.myDelegate, new object[]{Read_data});
        Indata = null;
        Read_data = null;
     }
  }
  catch (Exception ex)
  {
    MessageBox.Show(ex.Message, "Error Reading Data");
  }
}
```

Listing 9 C# code used for Receive event handler

In the meantime following the connection, the application waits for the serial receive event handler shown in listing 9. This was used to trigger an even whenever an incoming data was available at the receive buffer of the serial port. Applying event handler prevents polling that would otherwise consume much of the resource while the processor continuously scans the port.

The delimiter character '#' was used by teensy to indicate the end of data transmission. Therefore, the application program used this character to group the incoming data received from the serial buffer as shown in listing 6. The '`Read_data`' array holds the request sent from the teensy and further examines the data so as to execute the corresponding subroutine. Writing data to the serial port was done using the serial object as shown in listing 10.

```
serial_obj1.Write(arg);
```

Listing 10 C code for sending request to teensy

The argument 'arg' shown in listing 10, was a character constant used to identify the specific request that was predefined between the C# program and teensy firmware. Since the values shown in table 4 are common to both applications, it was possible to reduce the synchronization error

Table 4. Pre-defined characters used between C# and teensy

Argument arg	Description
s	Read serial
r	Read used data
w	Write user data
x	Quit operation
i	Search and identity transponder

Each 'arg' shown in table 4, represents a specific subroutine at the teensy. Therefore, the application program was able to send a request to the teensy and in return, the event handler would receive the data. With this setup completed, the first request was the command 'I'. This request was sent to teensy to execute an internal register to de-

tect a transponder. In return, the application program displayed the result in addition with the type of the transponder as shown in appendix 1.

After the transponder was detected, the ‘s’ command was sent to teensy in a request for the serial id. In return, the serial id was read and displayed to the corresponding text box and also verified at the LCD as shown in appendix 2. This confirms the communication was functional and the data was also valid.

Reading user data from the transponder was accomplished by sending command ‘r’ to teensy. The message box was used to show the data that was read from the transponder shown appendix 1. In this case, it was 9. This data was also verified with the LCD that displayed the value 0009 as shown in appendix 2.

Writing user data to the transponder, on the other hand, required sending the data in addition to command ‘w’. First 2% interest for 500 was calculated as shown in appendix 1.Then adding the result which was 10 into the existing value has returned 19 as shown in appendix 1. This confirms the new data was successfully written to the transponder memory. Similarly, the operation was verified by the LCD as shown in appendix 2.

Therefore at this point, it was proved that the application can communicate with the teensy to access the transponder via the reader. This concludes, the communication to the transponder can be done either from a standalone microcontroller or further extend to the computer application through a serial communication channel.

6 Discussion and Conclusion

It was during the placement period while working at ITSC technology support, the idea of this project was introduced. The goal of this project was to demonstrate the use of RFID over barcode-based coupon used by the company for auto-id and user data saving capability in providing better performance and automation. Therefore, the choice of Mifare RFID module was to show the budget constraint developing the system in response to the previous proposal that was declined by the host. Otherwise, the choice of teensy was only because of my academic familiarity.

Referring the results, the serial id embedded into the chip of the transponder is unique that cannot be changed or modified. As a result, the ability to read this value which was achieved in this project could replace the barcodes used by the company for identification purpose.

On the other hand, the wireless communication between the RFID reader and transponder not only enhance the automation process but also reduce maintenance cost due to reduced human participation required during barcode operation. In addition, the ability to read/write user data onto the transponder memory was also achieved in this project. This provides versatility for the company to integrate the application of bonus system with the existing tracking application into one platform with minimum cost and deployment challenges.

I lined up to the idea of hybrid system stated in this research paper [26] that incorporate the use of both RFID and barcode which provided better performance. On contrary to the paper [26], in this project reliability has been achieved as a result of the durability of transponders relative to barcode and consistency of serial id found on the chip.

Overall, I have successfully met the goals during this project such that the application of RFID with all the benefits of durability and versatility not only resolve the challenges of the host in applying barcode but also implicate how its implementation could be done with minimum training requirements as a result of the similarity with the existing system.

Limitations and Recommendations of the study

Even though the study achieved its aim, there were unavoidable limitations. For instance, sending integer data from the application software to teensy sometimes generate errors. This was because of the conversion used on a string array. In addition, using delimiter character to synchronize the communication still generates bugs that require the teensy to restart. This could be resolved by applying interrupt at teensy that would avoid polling for the incoming data. The above challenges were not resolved mainly due to the limitation of time.

Therefore, I recommend further study to be carried out to address the drawbacks mentioned above. In addition, security is as vital to RFID as it is necessary to the company. Therefore, in reference to the continuation of the project, I recommend security issues to be addressed.

References

1. Schuster E. Auto-ID Technology: Creating an Intelligent Infrastructure for Business. Business Intelligence Report. 2005. [Online]
http://web.mit.edu/edmund_w/www/CutterAutoIDReportV2.pdf
[Accessed 2 September 2015]

2. Weis S a. RFID (Radio Frequency Identification): Principles and Applications. Emerging Technologies. 2010. [Online].
http://www.eecs.harvard.edu/cs199r/readings/rfid-article.pdf
[Accessed 15 September 2015]

3. Ilie-Zudor E, Kemeny Z. The RFID technology and its current applications [Online].
http://www.laxcen.com/pdf/1355486568RFID_technologyandapplications_PW45.pdf
[Accessed 16 September 2015]

4. AMERICAN BARCODE CONCEPTS. Bar Code Basics
Basic concepts for barcode and automatic identification beginners. [Online]
http://www.amerbar.com/zebrasupplies/specials/barcode_basics_net.pdf
[Accessed 5 October 2015]

5. Finkenzeller K. RFID Handbook.
Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification, and Near-Field Communication. E.3rd, 2010.
[Accessed 9 October 2015]

6. Landt J. The history of RFID. IEEE Potentials. 2005. [Online].
http://ieeexplore.ieee.org.ezproxy.metropolia.fi/stamp/stamp.jsp?tp=&arnumber=1549751
[Accessed 2 October 2015]

7. Malakar B, Roy BK. Survey of RFID applications in railway industry. 1st International Conference on Automation, Control, Energy, and Systems - 2014, [Online]
http://ieeexplore.ieee.org.ezproxy.metropolia.fi/xpls/icp.jsp?arnumber=6807999
[Accessed 17 October 2015]

8. Gnoni MG, Rollo A, Tundo P. A smart model for urban ticketing based on RFID applications. IEEM 2009 - IEEE International Conference on Industrial Engineering and Engineering Management. 2009. [Online]
http://ieeexplore.ieee.org.ezproxy.metropolia.fi/xpls/icp.jsp?arnumber=5373004
[Accessed 23 September 2015]

9. Guo Zhanglin. The Application of RFID Technology in the Logistics Supply Chain. Computer Science and Information Technology (ICCSIT), 3rd IEEE International Conference on (Volume:2) 2010. [Online]
http://ieeexplore.ieee.org.ezproxy.metropolia.fi/stamp/stamp.jsp?tp=&arnumber=5565170
[Accessed 12 November 2015]

10. Aiyun G, Debin D, Zhongwei H. RFID in the Livestock Supply Chain Management: Application and Development [Online].
2010 International Conference on E-Business and E-Government. 2010.
//www.scopus.com/inward/record.url?eid=2-s2.0-78649675633&partnerID=tZOtx3y1
[Accessed 3 November 2015]

11. Booth P, Frisch PH, Miodownik S. Application of RFID in an integrated healthcare environment. Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings. 2006 [Online]
http://ieeexplore.ieee.org.ezproxy.metropolia.fi/stamp/stamp.jsp?tp=&arnumber=4461698
[Accessed 21 November 2015]

12. Akbari A, Mirshahi S, Hashemipour M. Application of RFID System for the Process Control of Distributed Manufacturing System. 2015. [Online].
http://ieeexplore.ieee.org.ezproxy.metropolia.fi/stamp/stamp.jsp?tp=&arnumber=7129325
[Accessed 23 September 2015]

13. Hyvonen L, Pinto A, Troelsen J. Near field communication [Online].
http://www.google.com/patents?hl=id&lr=&vid=USPAT8212735&id=HG0fAgAAEBAJ&oi=fnd&dq=Near+Field+Communication&printsec=abstract
[Accessed 22 October 2015]

14. “electronic design” Understanding Modern Digital Modulation Techniques
http://electronicdesign.com/communications/understanding-modern-digital-modulation-techniques [Online]
[Accessed 14 October 2015]

15. Lozano-Nieto A. RFID Design Fundamentals and Applications.
CRC Press; 2011. [Online].
http://www.petronet.ir/documents/10180/2323246/RFID_Design_Fundamentals_and_Applications
[Accessed 23 October 2015]

16. V L. A Review of Manchester, Miller, and FM0 Encoding Techniques [Online].
The Smart Computing Review. 2014.
http://www.smartcr.org/view/download.php?filename=smartcr_vol4no6p6.pdf
[Accessed 11 November 2015]

17. “digi-key” Considerations for Sending Data over a Wireless Link [Online]
http://www.digikey.com/en/articles/techzone/2011/aug/considerations-for-sending-data-over-a-wireless-link
[Accessed 11 October 2015]

18. Benthien GW. Digital Encoding And Decoding. August 13,2007,
Revised March 30 2010; [Online].
http://gbenthien.net/encoding.pdf
[Accessed 8 November 2015]

19. “infosec institute” Introduction to RFID Security [Online]
http://resources.infosecinstitute.com/introduction-rfid-security/
[Accessed 15 October 2015]

20. “NXP” MFRC522 Standard 3V MIFARE reader solution
Rev. 3.8 — 17,September 2014. [Online]
http://www.nxp.com/documents/data_sheet/MFRC522.pdf
[Accessed 11 August 2015]

21. “NXP” MF1ICS50 Functional specifications, Product data sheet [Online]
http://www.nxp.com/documents/data_sheet/M001053_MF1ICS50_rev5_3.pdf
[Accessed 09 September 2015]

22. ”PJRC.com”. Teensy USB Development Board [Online]
https://www.pjrc.com/teensy/pinout.html
[Accessed 08 August 2015]

23. Schildt H. C# 3.0 The Complete Reference [Online].
Howard Michael, editor. McGraw-Hill; 2009.
www.HerbSchildt.com
[Accessed 23 October 2015]

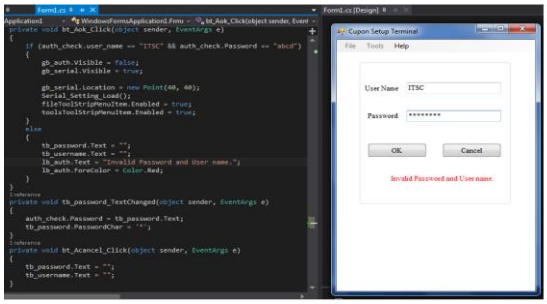
24. ”Texas Instrument” LM3940 1-A Low-Dropout Regulator for 5-V to 3.3-V Conversion [Online]
http://www.ti.com/lit/ds/symlink/lm3940.pdf
[Accessed 25 September 2015]

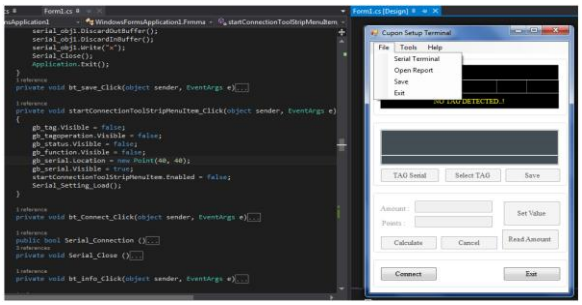
25 ”Eka Puji Widiyanto” AVR GCC library rc522 [Online]
https://github.com/ekapujiw2002/ex4-avr-gcc-library
[Accessed 25 September 2015]

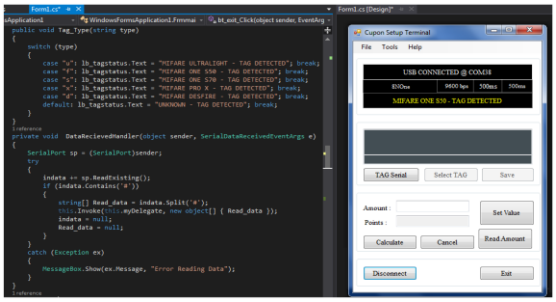
26 White G, Gardiner G, Prabhakar GP, Abd Razak A. A comparison of barcoding and RFID technologies in practice. J Information, Inf Technol Organ [Online]. 2007.
http://eprints.uwe.ac.uk/13460/
[Accessed 25 November 2015]

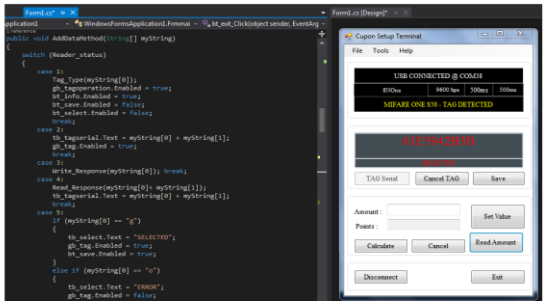
.

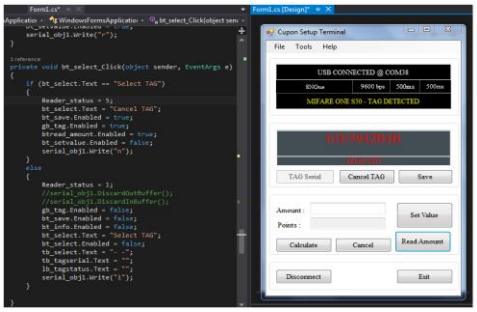
C# Test Results

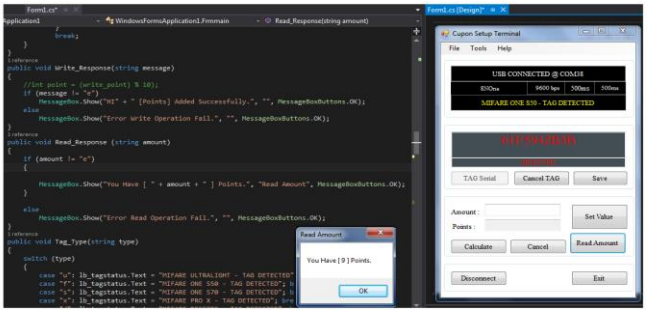


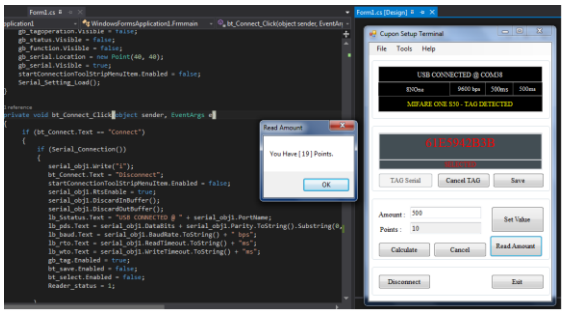


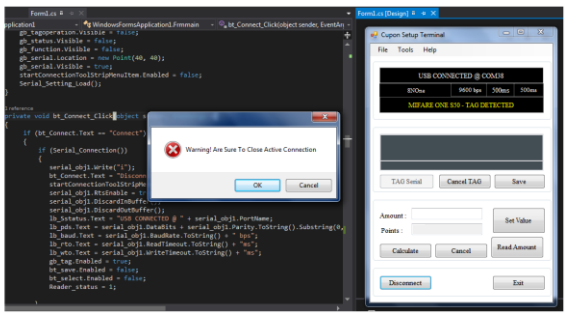




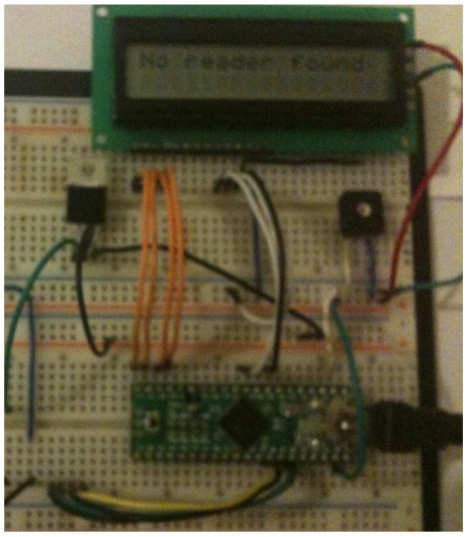


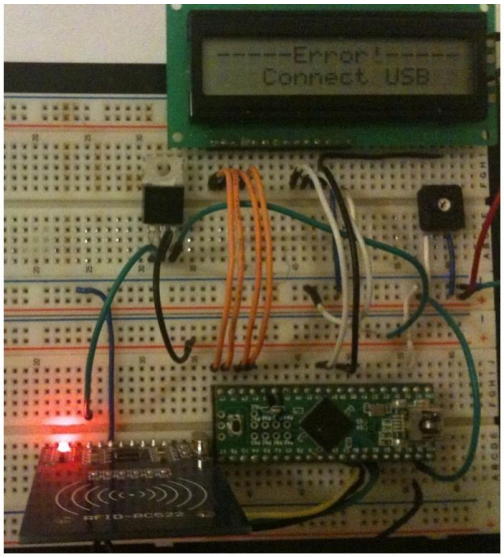


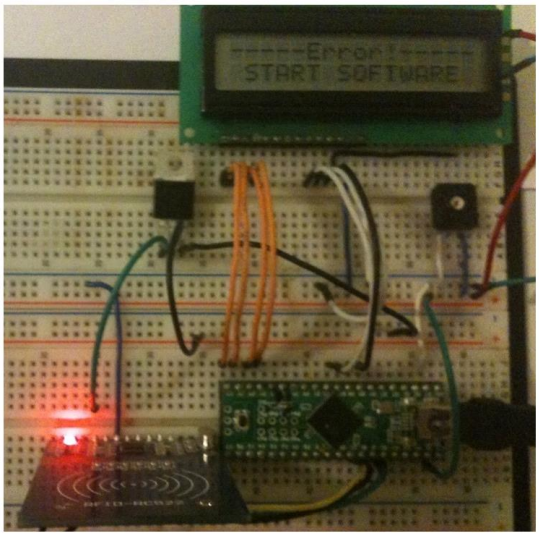


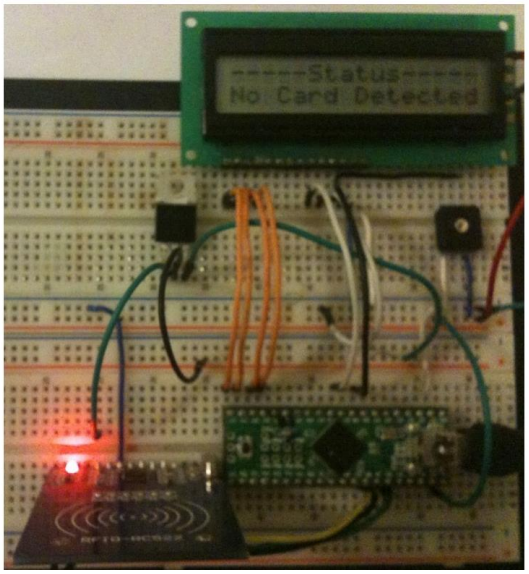


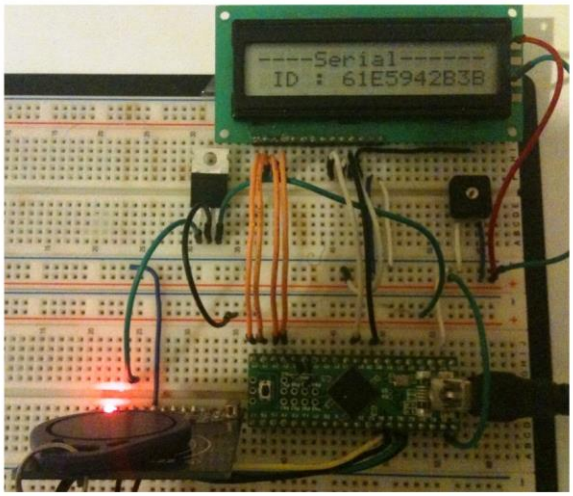
Hardware Test Result

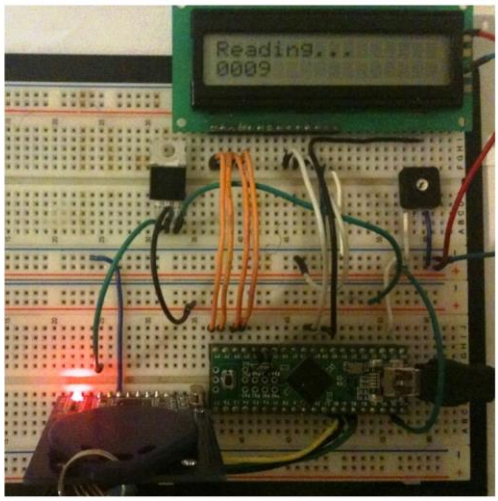


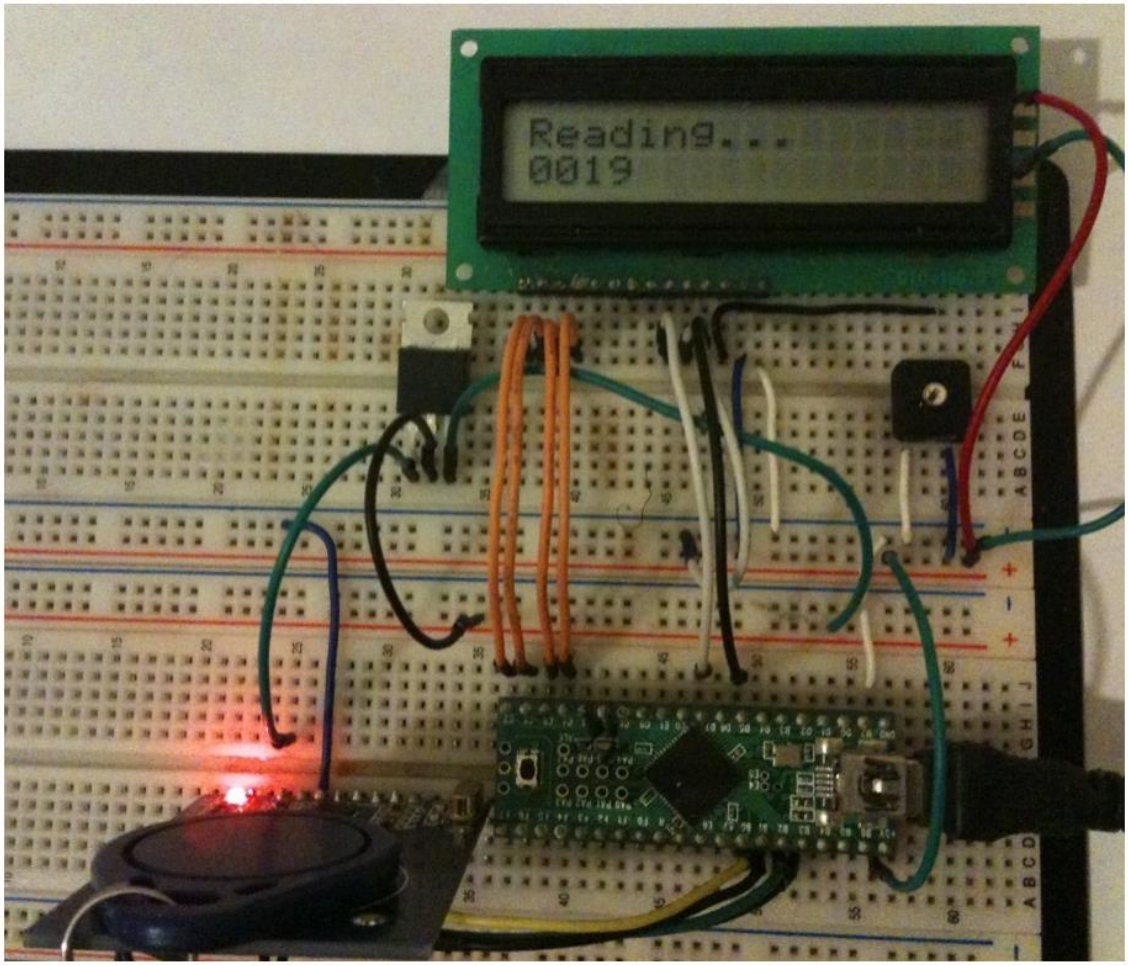







----Serial------
ID : 61E5942B3B


Reading...
0009


Reading...
0019

Transponder Memory access

Access bits			Access condition for						Remark
			KEYA		Access bits		KEYB		
C1	C2	C3	read	write	read	write	read	write	
0	0	0	never	key A	key A	never	key A	key A	Key B may be read
0	1	0	never	never	key A	never	key A	never	Key B may be read
1	0	0	never	key B	key A|B	never	never	key B	
1	1	0	never	never	key A|B	never	never	never	
0	0	1	never	key A	key A	key A	key A	key A	Key B may be read, transport configuration
0	1	1	never	key B	key A|B	key B	never	key B	
1	0	1	never	never	key A|B	key B	never	never	
1	1	1	never	never	key A|B	never	never	never	

Access bits			Access condition for				Application
C1	C2	C3	read	write	increment	decrement, transfer, restore	
0	0	0	key A|B[1]	key A|B[1]	key A|B[1]	key A|B[1]	transport configuration
0	1	0	key A|B[1]	never	never	never	read/write block
1	0	0	key A|B[1]	key B[1]	never	never	read/write block
1	1	0	key A|B[1]	key B[1]	key B[1]	key A|B[1]	value block
0	0	1	key A|B[1]	never	never	key A|B[1]	value block
0	1	1	key B[1]	key B[1]	never	never	read/write block
1	0	1	key B[1]	never	never	never	read/write block
1	1	1	never	never	never	never	read/write block