

# **Wired LAN Door Access Controller With Web-based Interface**

by

Ong Wen Sher

Dissertation Submitted In Partial Fulfilment Of

The Requirements For The

Bachelor Of Engineering (Hons)

(Electrical & Electronics Engineering)

DECEMBER 2010

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

## **CERTIFICATION OF APPROVAL**

**Wired Ethernet Interface Between Microcontroller-Based Access Controller And  
World Wide Web**

by

**Ong Wen Sher**

**A Project Dissertation Submitted To The  
Electrical & Electronics Engineering Programme**

**Universiti Teknologi PETRONAS**

**In Partial Fulfilment Of The Requirement For The**

**BACHELOR OF ENGINEERING (Hons)**

**(ELECTRICAL & ELECTRONICS ENGINEERING)**

Approved by,

---

(Mr. Patrick Sebastian)

Project Supervisor

**UNIVERSITI TEKNOLOGI PETRONAS**

**TRONOH, PERAK**

**DECEMBER 2010**

## **CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

---

ONG WEN SHER

## **ABSTRACT**

The advancement of technology has taken up a fast pace. Everything has to be done quickly and have high accessibility. Most of the door access security industries are competing to fulfil the above criteria in order to meet current demand from all users. This project's main objective is to develop prototype of a door access system which is able to communicate through Ethernet so that it allows the administrator to access and modify board settings through a remote PC connected to that network. The project uses Microchip development board as a platform and the firmware was modified and customized to fit the purpose of the project. This report will covers on the literature reviews that have been done, tools, equipments, procedures and also end result of the project.

## **ACKNOWLEDGEMENT**

First and foremost, I would like to thank Micro ID Sdn Bhd for their willingness to allow me to involve in the project collaboration with Universiti Teknologi PETRONAS. It was a great honor for me to assist them in completing this project as part of their marketable product through my final year project.

Apart from that, I would like to thank Mr. Patrick Sebastian, my supervisor for his endless supports and assistance during the hard time. Inevitably, I had gained a lot of knowledge from a year of final year project. Besides, Universiti Teknologi PETRONAS, especially the Electrical and Electronics department has contributed its help by providing me the necessary tools, equipments and facilities in order for me to complete the project smoothly.

Not to be forgotten, Microchip Inc. had offered their consultancy help when I faced difficulties in understanding their products. It had been a great help since most of the project deals with Microchip's product. Furthermore, I would like to express my gratitude to the lab technicians whom have contributed on this project by sharing their opinions and experience in solving technical and management problems. It had been a great opportunity to have work with them.

Last but not least, there are also appreciations to my fellow colleagues and family for giving me not only their technical support, but moral support as well. This project has indirectly thought deeply on teamwork and supports. Again, I would like to thank every individual that have directly and indirectly involved in this project.

Thank you.

## **TABLE OF CONTENTS**

CERTIFICATION OF APPROVAL .....	ii
CERTIFICATION OF ORIGINALITY .....	iii
ABSTRACT .....	iv
ACKNOWLEDGEMENT .....	v
LIST OF FIGURES .....	ix
LIST OF TABLES .....	x
LIST OF ABBREVIATION AND NOMENCLATURES .....	xi
CHAPTER 1 : INTRODUCTION .....	1
1.1 Background Of Study .....	1
1.2 Problem Statements .....	2
1.3 Objectives .....	4
1.4 Scope Of Study .....	4
CHAPTER 2 : LITERATURE REVIEWS .....	5
2.1 Transmission Control Protocol / Internet Protocol ....	5
2.2 Common Gateway Interface (CGI) .....	6
2.3 Hypertext Transfer Protocol (HTTP) .....	7
2.4 Address Resolution Protocol (ARP) .....	9
2.5 Dynamic Host Configuration Protocol (DCHP) .....	10

2.1.1 <i>Dynamic Allocation</i> .....	10
2.1.2 <i>Automatic Allocation</i> .....	11
2.1.3 <i>Static Allocation</i> .....	11
2.6 Media Access Control Address (MAC Address).....	12
2.6 Wiegand Protocol .....	13
2.7 Serial Peripheral Interface.....	16
CHAPTER 3 : METHODOLOGY.....	19
3.1 Procedure Identification.....	19
3.2 Tools And Equipment .....	22
CHAPTER 4 : RESULTS AND DISCUSSIONS .....	24
4.1 Results.....	24
CHAPTER 5 : CONCLUSION AND RECOMMENDATIONS .....	30
5.1 Conclusion .....	30
5.2 Recommendations .....	31
REFERENCES .....	32
APPENDICES .....	34
APPENDIX A : ELECTROMAGNETIC CARD READER.....	35
APPENDIX B : ELECTROMAGNETIC CARDS .....	36
APPENDIX C : EXPLORER16 DEVELOPMENT BOARD .....	37
APPENDIX D : ETHERNET MODULE (ENC28J60) .....	38
APPENDIX E : INPUTS AND OUTPUTS TO MCU .....	39
APPENDIX F : MICROCHIP FILE SYSTEM GENERATOR .....	40

APPENDIX G	: SCHEMATIC DIAGRAM FOR WIEGAND LINE .....	41
APPENDIX H	: INTERRUPT FOR CHANNEL D1 OF WIEGAND ....	42
APPENDIX I	: INTERRUPT FOR CHANNEL D0 OF WIEGAND ...	43
APPENDIX J	: SOURCE CODE FOR READING CARD.....	44
APPENDIX K	: SCHEMATIC DIAGRAM OF EEPROM (25LC512) ...	47
APPENDIX L	: SOURCE CODE FOR READ FROM EEPROM .....	48
APPENDIX M	: SOURCE CODE FOR WRITE TO EEPROM .....	49
APPENDIX N	: SOURCE CODE FOR ADD/DELETE USER .....	50
APPENDIX O	: HTML CODE TO ADD/DELETE USER .....	52

## LIST OF FIGURES

Figure 1 : Conventional layout of access controller system .....	1
Figure 2 : New system overview using router, wired and wireless LAN .....	3
Figure 3 : Flow of data to gateway programs [2].....	7
Figure 4 : Example GET request displayed on URL .....	8
Figure 5 : Cisco's Open System Interconnection (OSI) model .....	12
Figure 6 : Signal and pulse duration [8].....	14
Figure 7 : Example of a stream of bits with parity check .....	15
Figure 8 : Overview on Master and Slave block diagram.....	16
Figure 9 : Overall Project Work Flow .....	19
Figure 10 : Main page of the webpage.....	20
Figure 11 : Program Flowchart .....	24
Figure 12 : Layout changes in the webpage design .....	25
Figure 13 : HTML code to construct drop down box in Figure 12.....	26
Figure 14 : Add/Delete User Page .....	26
Figure 15 : HTML codes that builds up the box for administrator to add or delete user. ....	27
Figure 16 : Configuration page of a router, on Address Reservation .....	28
Figure 17 : Configuration page on DHCP Clients List, with the MAC Address..	29

## **LIST OF TABLES**

Table 1 : Structures of 26-bits with parity bits, facility code and card number....	14
Table 2 : Distribution of odd and even bits in a 26-bits structure.....	15
Table 3 : Summary table on parity bit checking condition .....	15
Table 4 : Description of each lines used in SPI communication.....	17
Table 5 : List of tools and their functions .....	22
Table 6 : List of equipments and their functions .....	23

## **LIST OF ABBREVIATION AND NOMENCLATURES**

HTML	HyperText Markup Language
EEPROM	Electrically Erasable Programmable Read-Only Memory
TCP/IP	Transmission Control Protocol / Internet Protocol
LAN	Local Area Network
CSS	Cascaded Style Sheet
CGI	Common Gateway Interface
IP	Internet Protocol
MPFS	Microchip File System
MCU	Microcontroller Unit
ARP	Address Resolution Protocol
SPI	Serial Peripheral Interface
MAC	Media Access Controller

# CHAPTER 1

## INTRODUCTION

### 1.1 Background Of Study

Wired Ethernet interface between microcontroller-based access controller and World Wide Web is a joint collaboration project with Micro ID Sdn. Bhd. An access controller is a device that ensure that only authorized users are allowed to access to a computer network [1].

Access controllers is a set of devices that receives input either from electromagnetic card, biometrics or pin number from user, verify them and grant access to the user accordingly. They are usually used to enhance security to protect confidentiality of resources, information and system.

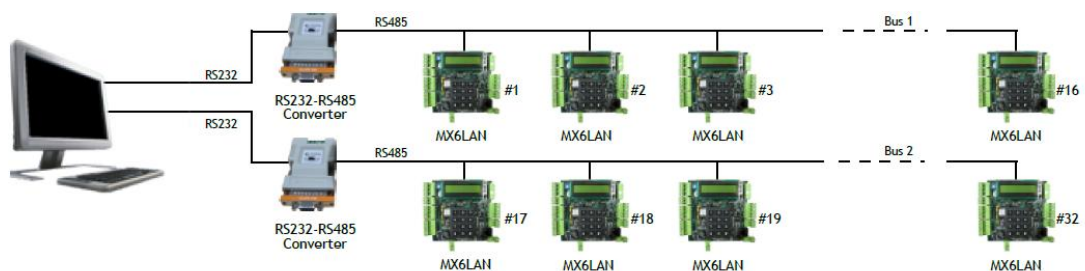


Figure 1 : Conventional layout of access controller system ([www.microidee.com](http://www.microidee.com))

Figure 1 shows the conventional layout of access controller system which has one dedicated host PC. Each of the doors has their access controller installed and connected to a host PC. The host PC will be the control centre that handles all the modification and settings of each access controller.

## **1.2 Problem Statements**

The conventional RS232 communication limits the distance of each connected devices. Apparently, the standard maximum length of RS232 cable is said to be about 15 metres [2]. The proposed solution that uses Ethernet cable can reach up to 100 metres [3] maximum. Hence, by using the proposed solution, the devices can be installed at further distance. It can be enhanced by using a router to amplify the signal.

Apart from that, there is only one RS232 port on a host PC. There can be only one access controller connected to a PC at a time. By using Ethernet, a cable can be connected to the router, enabling several devices can be connected at the same time from that point. RS232 does not provide unique identifications to each device that are connected to the host PC. However, using the proposed solution, the administrator will be able to distinguish the identification of each access controller by assigning internet protocol address (IP address) to each of them.

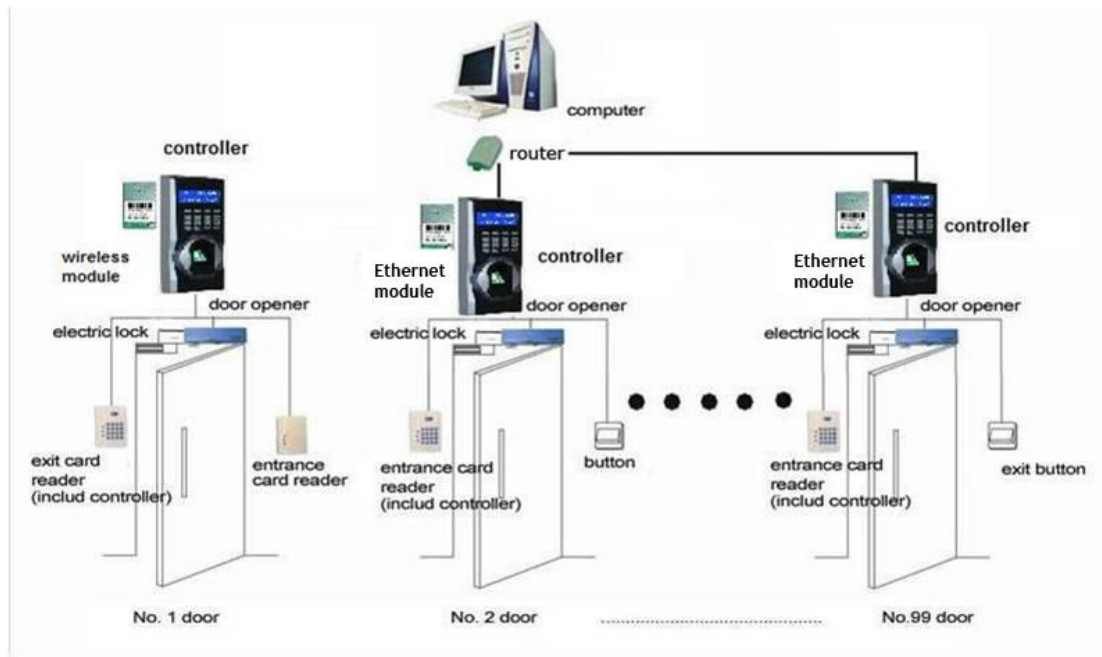


Figure 2 : New system overview using router, wired and wireless LAN  
([www.microidee.com](http://www.microidee.com))

Another one of the main concerns in access controller is the ease of accessibility. The existing access controller needs to be accessed directly through the controller in order to perform settings such as add user, delete user and other configurable settings. This has brought up the limitations of being immobile and limited area of coverage. By proposing a web-based solution, administrator of the access controller can change its setting without having to access directly on the device itself.

### 1.3 Objectives

The objective of this project is to solve the problem mentioned above. This includes:

- To design and develop an access controller board that communicates with the PC through Ethernet so that administrator can access and make changes to the system settings through a remote PC.
- To create user interface using HTML for data transfer, monitoring and maintaining purposes.
- To integrate more than one devices to form a system in order to demonstrate the unique identification of each devices.

### 1.4 Scope Of Study

In this project, there are a few scopes of study that are covered in the project. They are listed in the list below:

- Web language (*HTML, CSS*) – This project requires detailed study the web language in order to construct an interactive web page for the administrator to change their controller's setting. Besides, it is crucial to gain deep understanding on how the data transfer can be done between a web page and the controller.
- Programming Language (C language) – The whole project deals mainly on the programming code by using C30 Compiler from Microchip. This compiler utilizes the usage of C programming.
- TCP/IP stacks – This is one of the cores of the project. It determines the IP address assignment and also communication in between two medium.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Transmission Control Protocol / Internet Protocol (TCP/IP)**

TCP/IP is a type of protocol in communication apart from OSI. It consists of 5 layers in its architecture below.

- Physical Layer
- Network Access Layer
- Internet Layer
- Host-to-Host or Transport Layer
- Application Layer

##### **A. Physical Layer**

It is the layer where physical interface between two or more data transmission's medium are located.

##### **B. Network Access Layer**

This particular layer deals with the exchange of data between a system and the network attached to it. It is also covers the accessing and routing of data between two different systems.

### C. Internet Layer

This layer is where the Internet Protocol (IP) is used in this layer to provide routing services for multiple networks. However, IP is not only applicable to any end system, but also to a router. For further information, router is described as a processor that bridge two networks and mainly to deliver data from a network to the other. This is done through routing from source's end system to the destination's end system.

### D. Transport Layer

Transport layer usually functions to ensure the data reliability after being transferred through transmission media. It is crucial for the data to arrive at destination without any error just like how it was being sent from the source. In this case, Transmission Control Protocol (TCP) is commonly used to provide this accuracy.

### E. Application Layer

The highest layer, which is the application layer consists the logic to support several different user applications which includes FTP, HTTP, SMTP, TELNET, etc.

## **2.2 Common Gateway Interface (CGI)**

Common Gateway Interface (CGI) is a method for web servers to interact with gateway programs (or side servers) in sending data. It is suitable to use for dynamic files handling. CGI has been quite a popular interface in gateway programs of forms handling and databases to web server. A few examples of application that uses CGI the best is guestbook submission, feedback form, search engine and more. It will determine how those data are sent and received by the gateway program. For a CGI, there will be a need of a sub-directory called cgi-bin. If specified, then the server will understand that it needs to go into cgi-bin and find the file name (which is written after cgi-bin/).

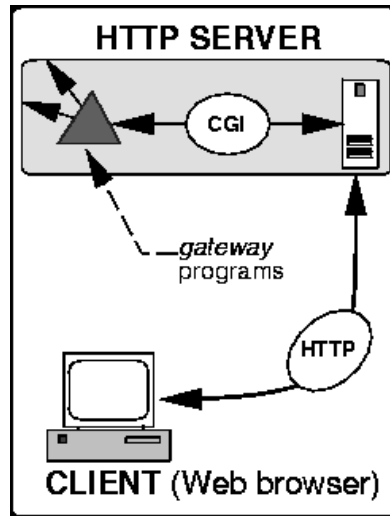


Figure 3 : Flow of data to gateway programs [4]

Figure 3 shows the diagram of how the data flow to gateway program. As seen in the figure, CGI acts as an intermediate to fetch gateway programs to the server. In that way, the server will be ready when the client requests for the information.

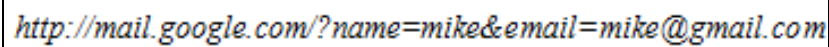
However, there is an important reminder that says whatever files that were accessed in cgi-bin subdirectory should be executed. They should not be read and sent out. CGI can be written in any language, but commonly in C. They do have limitations, but these limitations are caused by the remote computers' and the browsers that are used.

### 2.3 Hypertext Transfer Protocol (HTTP)

HTTP is an application layer in TCP/IP Protocols (can be on other protocols as well). Conventionally, HTTP utilizes port 80 in TCP port, but sometimes it can reside in port 8080 as well. In HTTP terms, web browsers are labelled as clients, while client that submits requests is called user agent. On the other hand, servers are programs or applications that host any web sites.

In the example of submitting forms, there are two different methods in sending the data which is through GET request or POST request[5].The method of submitting the data can be set by the users.

If assuming the user wants the data to be transferred through GET request, then the encoding would be much easier than the others. Considering the user tries to login to his email address with name and email address submitted, in result, the address would be like in Figure 4.



```
http://mail.google.com/?name=mike&email=mike@gmail.com
```

Figure 4 : Example GET request displayed on URL

However, GET request method is not recommendable for the storing of data which is long and also with the chain effect. Usually GET requests are only applicable to one time search, or data with no permanent state of change. One the example that can be given is the search box. In search box, users submit the data that they want to find then in return, will link to the pages that relate to the search keyword. The difference can be seen clearly after understanding the concept of POST request method.

In POST request method, it is seldom used unless the request will require permanent change of state on the server such as updating the database. In updating database, the data that has been submitted will be stored permanently inside the storage for future retrieval. Hence POST method is a more secured way in ensuring the accuracy. In POST, the input data can be very long compared to GET method. By just looking at this point, POST is more secure than GET because the latter can could snip off the last few important information (if it ever gets too long). The data that is passed through POST method do have their headers to ensure the consistency and accuracy of the data delivered.

## **2.4 Address Resolution Protocol (ARP)**

ARP is located in one of the layers of TCP/IP. It works in a way that it converts Internet Protocol (IP) address to a certain physical network address or preferably called MAC (Media Access Control) address. ARP mostly used on Ethernet networks, ATM, Token Ring and more. Among all the networks, we are going to focus only on Ethernet network which applies to this project.

In the problem statement, it is concerned that within a network, each controller can be connected all at once and there will not be clashing of data. This assurance is strong with the presence of ARP. For each Ethernet network card or adapters, they have their own MAC address assigned by manufacturers, which at best will ensure that the 48-bits (6 bytes) addresses are unique. With this uniqueness, several Ethernet adapters could be linked simultaneously to a central of a same network, but will receive different messages depending on which addresses it was forwarded to.

In short, during a data transmission from a device to another target source, the most important details that are needed will be the IP address and also MAC address. From these two addresses, ARP cache that was maintained on each device will have the IP-to-MAC address mappings. With this, the message or data can be easily sent to any target devices.

## **2.5 Dynamic Host Configuration Protocol (DHCP)**

DHCP is a standard protocol defined by RFC1541 that allows Internet Protocol address (IP address) distribution much easier and manageable regardless big or small network. Through DHCP, the server will automatically assign an IP address dynamically to the devices connected to it. These processes do not require any control from network administrator. This protocol will provide information on IP address, subnet mask, default gateway and Domain Name System (DNS) addresses

Inevitably, the usage of DHCP has a few pros and cons. For the pros, DHCP can ease the process of adding in new devices into the server. Instead of having network administrator to manually assign the devices, DHCP can immediately recognize new devices and assign an unused IP address. Besides, it can prevent the situation where more than one device had the same IP address. However, as much as DHCP is convenient, it faces some security concerns since it does not include any authentication mechanism. It will be more vulnerable to any attacks. There are solutions to this problem, which is to introduce DNS into the system. DNS will not be discussed in details in this report.

Technically, DHCP has three different methods that were practiced – dynamic, automatic and static allocation. Three of the aforementioned methods are further elaborated below.

### ***2.1.1 Dynamic Allocation***

For this type of allocation, initially, a range of IP addresses were allocated by the network administrator. Each device connected to the server will request IP address during initialization. However, this method has lease time, which means there are limitation of time duration before the server refreshes and renew the IP address of the devices connected to it. As an example, after a lease time of one hour, the server checks if there are any IP addresses that no longer used. These addresses will be opened for new IP addresses distribution. This method maximizes the usage

of IP addresses, but at the same time the devices might have different addresses each hour.

### *2.1.2 Automatic Allocation*

Automatic allocation is more or less similar to dynamic allocation method but in this method, there are no lease time used. The server will keep track of the previous IP address assigned to the device, and try to reassign the same address to it again.

### *2.1.3 Static Allocation*

This is the method that was utilized in the project, static allocation. Static allocation works by having Media Access Control Address (MAC Address) and Internet Protocol Address (IP Address) pair. The server will keep information on the MAC addresses of each device, and their designated IP addresses in its database. When a valid device joins the network, the server will check its MAC address and assigns the designated IP address. Using this method, a device will have the same IP address as long as the network administrator does not change the addresses pairs. Hence, prior setting the server to this mode, the network administrator needs to know the MAC address of all devices. If there is a new device needs to be added into the server, network administrator will have to redefine the addresses pair.

## 2.6 Media Access Control Address (MAC Address)

Media Access Control Address is a type of unique identification on all network interfaces for the purpose of common physical network segment. Each device has their own MAC address, thus giving them a one and only identity. MAC Address is located in the link layer of Open System Interconnection (OSI) Model.

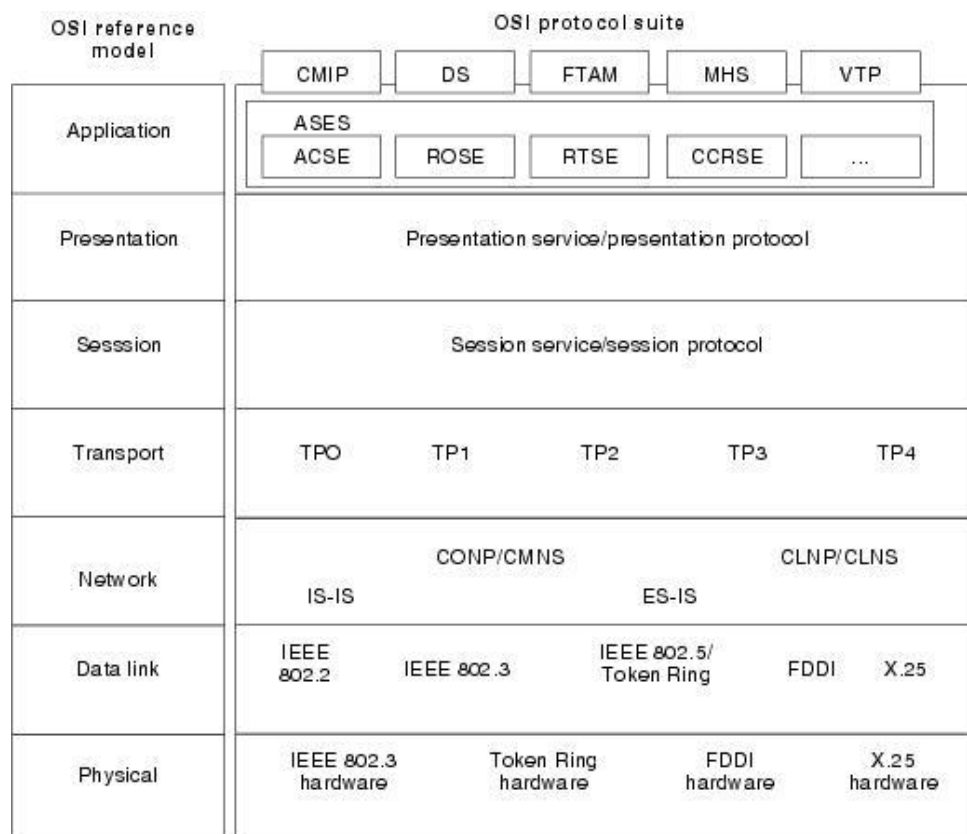


Figure 5 : Cisco's Open System Interconnection (OSI) model

In Figure 1, the 2<sup>nd</sup> layer from bottom is the data link layer, where the Media Access Control protocol was located. This protocol requires Media Access Control addresses. The history of MAC address originated from Xerox Ethernet addressing scheme. It has several alternative naming such as Ethernet hardware address (EHA), hardware address, adapter address and also physical address. These addresses were usually stored by the manufacturer itself especially the Network Interface Card (NIC). The information will be stored either in hardware or Read-Only Memory (ROM).

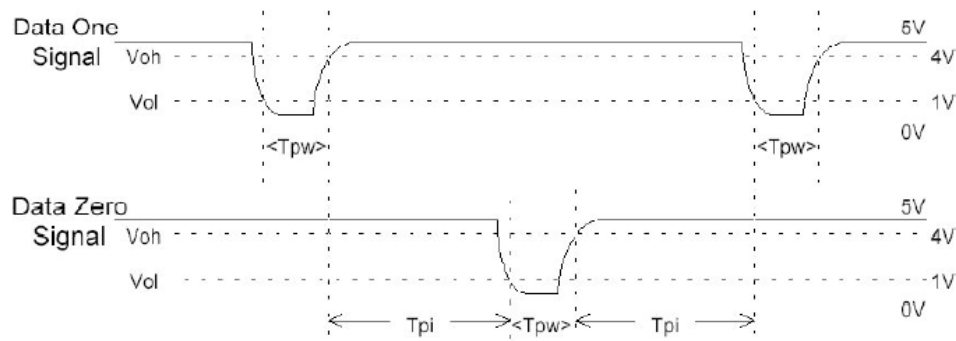
Format numbering of the addresses was created according to the name spaces, administered by Institute of Electrical and Electronics Engineers (IEEE). They are MAC-48, EUI-48 and EUI-64. However, Extended Unique Identification (EUI) is a registered trademark under IEEE. Conventional way of MAC addressing is usually 6 groups of 2 hexadecimal number, each separated by a hyphen (-) or colon (:). Example of MAC address that was usually seen is '04-34-76-ad-4e-f0' or '04:34:76:ad:4e:f0'.

In the application of Transmission Control Protocol/Internet Protocol (TCP/IP), there are methods to retrieve the MAC addresses by using the IP address itself. For Internet Protocol Version 4 (IPV4), Address Resolution Protocol (ARP) can be used, while Internet Protocol Version 6 (IPV6) utilizes Neighbour Discovery Protocol (NDP). Throughout the project, we will be focusing only on IPV4 instead of IPV6.

Even though MAC address has been set by the manufacturer prior to the distribution, permanent and globally unique, it is still possible for one to change the MAC address. This can be done by applying the method called MAC address spoofing. This project does not require the usage of address spoofing.

## **2.6 Wiegand Protocol**

Wiegand protocol is a well-known protocol used for the purpose of security, access control, time attendance and further related field. It is an interface that is used between readers and also the control panels of the respective device for the mentioned purposes. The unique thing about Wiegand protocol is that the data was sent through 2 different pins, which are D0 and D1. Pin D0 is responsible to trigger to the zero voltage for a short while to indicate logic '0'. Logic '1' is indicated by having D1 triggered to zero voltage.



Symbol	Description	Duration
Tpw	Time for pulse width	50 microsecond (us)
Tpi	Time for pulse interval	2 milisecond (ms)

Figure 6 : Signal and pulse duration [8]

Figure 6 visualizes that these short triggering happened for 50us after 2ms separation between each pulses. Referring to figure 6, “Data One Signal” represents D1 while “Data Zero Signal” represents D0.

There might be several types of Wiegand protocol, but in this project, Wiegand 26-bits are used. The structures of the 26-bits were broken down into 8-bits of facility code, 16-bits of card number and 2-bits for parity bits.

Table 1 : Structures of 26-bits with parity bits, facility code and card number

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
M	F	F	F	F	F	F	F	F	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	L
S	C	C	C	C	C	C	C	C																	S
B																									B

The Most Significant Bit parity (MSB) and Least Significant Bit parity (LSB) seen in table 1 acts as a counter to check whether or not the data received or transmitted is consistent.

Table 2 : Distribution of odd and even bits in a 26-bits structure

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
M	E	E	E	E	E	E	E	E	E	E	E	E	O	O	O	O	O	O	O	O	O	O	O	O	O	L
S																									S	
B																									B	

In 26-bits Wiegand, the first 12-bits excluding the MSB parity will be associated with the MSB parity while the last 12 bits excluding LSB parity are related to LSB parity. With reference to table 2, the first 12-bits which were labelled as ‘E’ will be checked for the number of logic ‘1’ available. If there is an even number of ‘1’s, MSB should give a ‘0’ value. If there is an odd number of logic ‘1’ in the yellow shaded area, LSB should give a value of ‘0’. A detailed table summarizing the parity bits checking system is shown as below, in table 3.

Table 3 : Summary table on parity bit checking condition

Number of ‘1’s in blue shaded area	Number of ‘1’s in yellow shaded area	Value of parity bits	
		MSB Parity	LSB Parity
5 (odd)	9 (odd)	1	0
8 (even)	10 (even)	0	1

If the parity bits do not adhere to the value in the table, then it will be known that the data has been corrupted or damaged.

Data stream 1:

1	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	0	0	1	1	0	0	0	0	1
M	F C								6				6				3				0				L
	E E E E E E E E E E E E										O O O O O O O O O O O O O O O O														

Figure 7 : Example of a stream of bits with parity check

In a valid example shown in Figure 7, it can be seen that for the first 12 bits of data shaded in blue box, there are odd number of ones. The MSB parity matches the condition shown in table 3, which is to show a value of '1'. As for the remaining last 12 bits shaded in maroon box, there is even number of ones. The LSB parity also matches the condition seen in table 3, thus proving the data is not damaged or corrupted.

## 2.7 Serial Peripheral Interface

Serial peripheral interface (SPI) is a type of communication which can be use to communicate with any peripheral devices or microcontroller especially when applying master and slave concept. Data of 8-bits are transferred between two or more devices within a speed by using just one line, which implies the serial transmission.

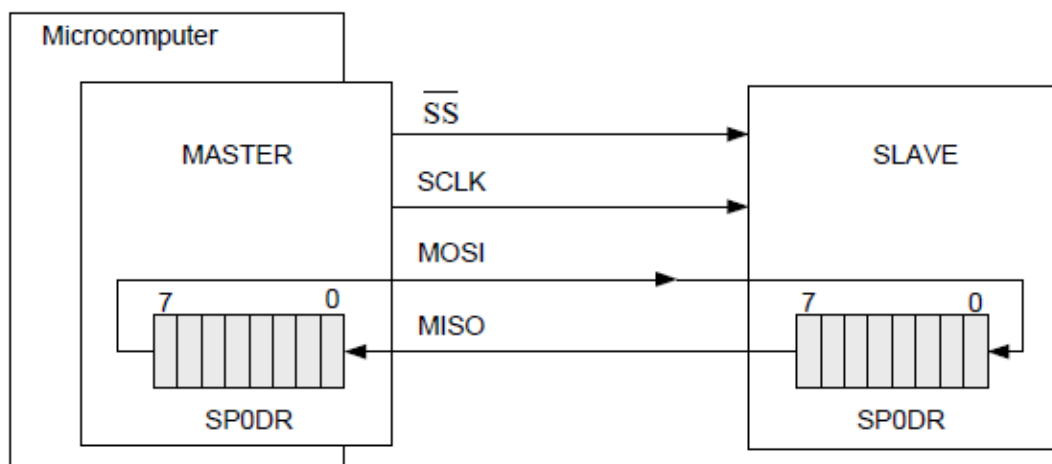


Figure 8 : Overview on Master and Slave block diagram

Figure 8 shows the connection of master and slave and how the data transmission occurs. There are four crucial lines that are involved in a serial communication for SPI, which include slave select (SS), serial clock (SCLK), master out slave in (MOSI) and master in slave out (MISO). The functions of each line are explained in the table shown below.

Table 4 : Description of each lines used in SPI communication

Port/Lines	Description
Serial Clock (SCK)	SPI is a synchronous protocol because they have clock input. This will enable the devices operating on SPI have variety on the clock rate. The data will not be interrupted when the clock rate varies. Instead, the data will varies accordingly to the clock rate. SPI works on Master-Slave mode. Master will have absolute control over the slave especially on the clock.
Master In Slave Out (MISO) and Master Out Slave In (MOSI)	SPI always communicates in 2 ways, which means, transferring mode. There is no one way communication for SPI because they have 2 data lines, serial data out (SDO) and serial data in (SDI) as written in microcontroller's datasheet. When two devices are in master and slave mode, they will use the term master in slave out (MISO) and master out slave in (MOSI).
Slave Select (SS)	This is also considered the most important line because it stands for Slave Select, or commonly called Chip Select. This is the line that will tell the master whether or not to transfer to the respective device.

A few of the advantages of using SPI include the full duplex communication whereby at one time, a serial input can be fed and a serial output can be push out simultaneously. SPI has higher throughput as compared to I<sup>2</sup>C and require less pin usage. They only need 4 basic pins in order to successfully communicate through SPI. SPI can reach up to 10Mbps/second while I<sup>2</sup>C has 100Kbits/second speed [9]. Therefore, SPI is very suitable for high speed communication as compared to other communication type.

However, there are some setbacks from using SPI communication. It is not suitable to use SPI if the master and slave devices are separated far away. SPI can only handle short distance and also limited number of slave it can support. This is one of the features that need to be sacrificed by SPI in satisfying the speed power. In addition, it cannot support too many slaves and operates only on a single master device. Here in this project, the application of SPI communication is not only limited to EEPROM application, but also Ethernet interface.

## CHAPTER 3

### METHODOLOGY

#### 3.1 Procedure Identification

In this report, the progress reported is the concluded from Final Year Project 1 and 2. Figure 9 below shows that the planned tasks were successfully completed.

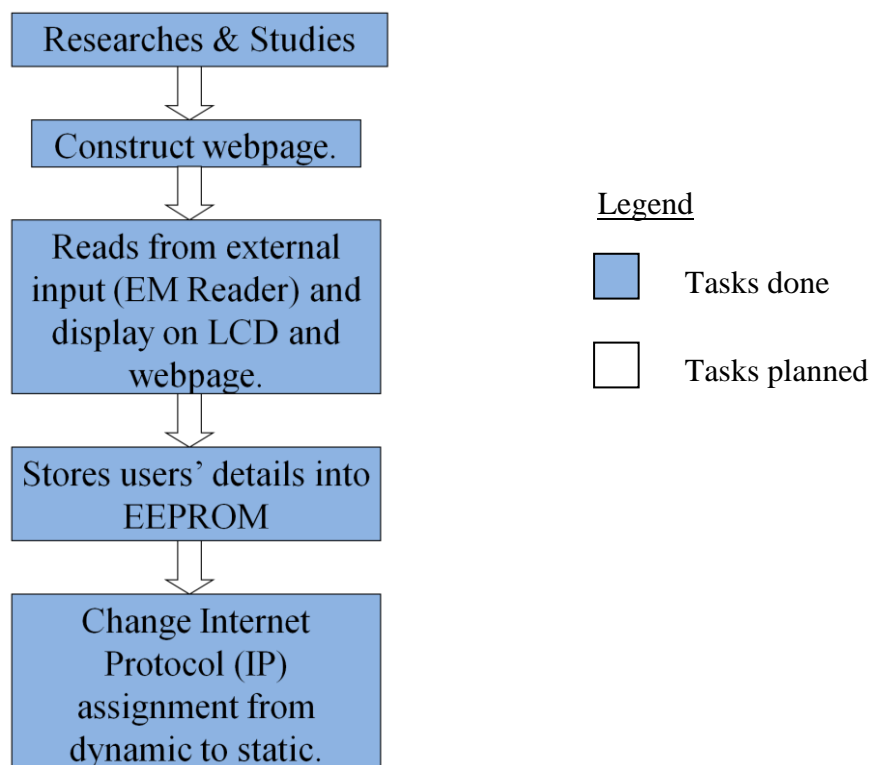


Figure 9 : Overall Project Work Flow

First three parts of the project flow were completed during Final Year Project 1. The remaining progresses were continued and completed during the Final Year Project 2.

The researches and studies refer to the literature reviews that were carried out along the projects. The webpage construction was completed during first phase of project, during FYP 1.

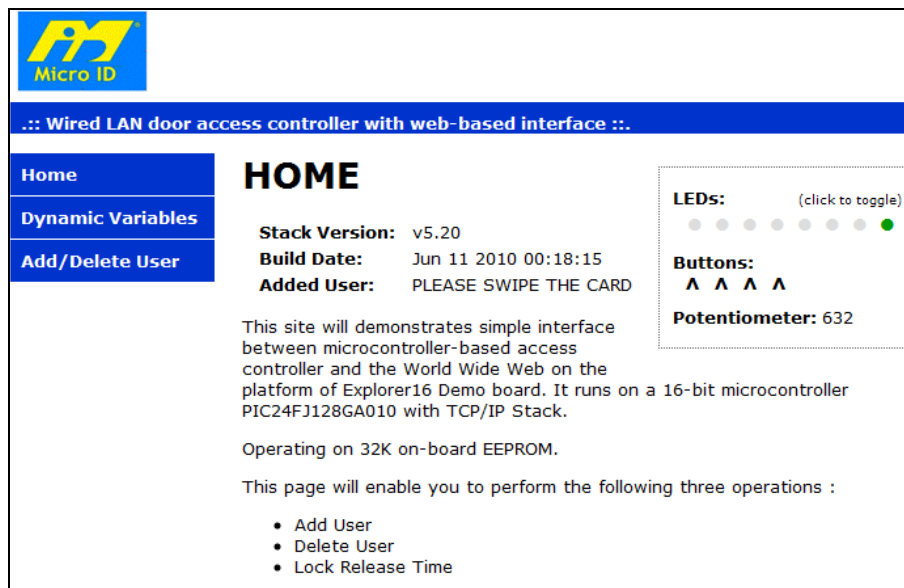


Figure 10 : Main page of the webpage

Figure 10 shows the main view of the modified webpage. There are LED indicators and as well as another sub-page that is dedicated to add or delete user. The enrollment of new user and deletion of existing user can only be done through the webpage.

The EM card reader that was installed externally will reads any information from the EM cards using Wiegand protocol. It will read the card number and compares it to the memory database of the users added before this. At the same time, the LCD will display the card number for a few instances before it returns to the default line.

As mentioned previously, the users' data will be added into the memory database for the verification part. Here, Serial Peripheral Interface (SPI) Electronically Erasable Programmable Read-Only Memory (EEPROM) was used to store such data. In fact, such information can only be entered by the administrator who has the access to the webpage.

Apart from that, the Internet Protocol (IP) address allocation has been modified from dynamic allocation to static allocation. This is crucial as we want to identify which IP address gives access to which controller mounted on the door. When integrated into a product, these controllers will be mounted on different doors located at different levels of a building. It is important for the administrator to keep track on the controller with full mobility and accessibility. Hence, static IP address allocation was chosen to be the one to be used in this project. It is not desirable that each devices get new IP address after the least time expired. Static IP allocation will ensure that for a fixed MAC address, a fixed IP address will be paired with it until the network administrator changes the addresses pair. Router was used as a device and centre to assign the IP address and also to let all devices communicate.

### 3.2 Tools and Equipment

Throughout this project work, several tools and equipments were utilized to carry out the work smoothly. The list of tools and its' purposes were listed in Table 5.

Table 5 : List of tools and their functions

Tools	Functions
Explorer 16 Development Board from Microchip	Serve as a platform for project development
Ethernet Module (ENC28J60)	Provides Ethernet interface to the development board
MPLAB IDE v8.33 (further upgraded to v8.50 in FYP 2)	Graphical User Interface (GUI) for compiler and programmer for the purpose of programming and debugging
MPLAB C30 v3.23 (C Compiler for PIC24)	Integrated into MPLAB IDE to serve to compile the C codes in PIC24 format.

Table 6 summarizes the list of equipments used throughout this project construction. There are brief descriptions on the functions accompanying the equipments.

Table 6 : List of equipments and their functions

Equipments	Functions
12V Power Adapter	An easy, small and user friendly power source to power up the development board
PicKit2 PIC Programmer	<p>A device that connects between a source (PC with compiled C codes) to target (microcontroller) in order to store the HEX code generated by the compiler into the target.</p> <p>Able to serve as a debugger device too.</p>
Electromagnetic (EM) Reader	A device to read the EM card and extract the information contained in the card.
Electromagnetic (EM) Card	Cards with unique card identification number for different users

## CHAPTER 4

### RESULTS AND DISCUSSIONS

#### 4.1 Results

Overall, the project will be able to receive input from an electromagnetic (EM) card reader and webpage form processing. The figure of electromagnetic (EM) card reader and the EM card can be referred at Appendix A.

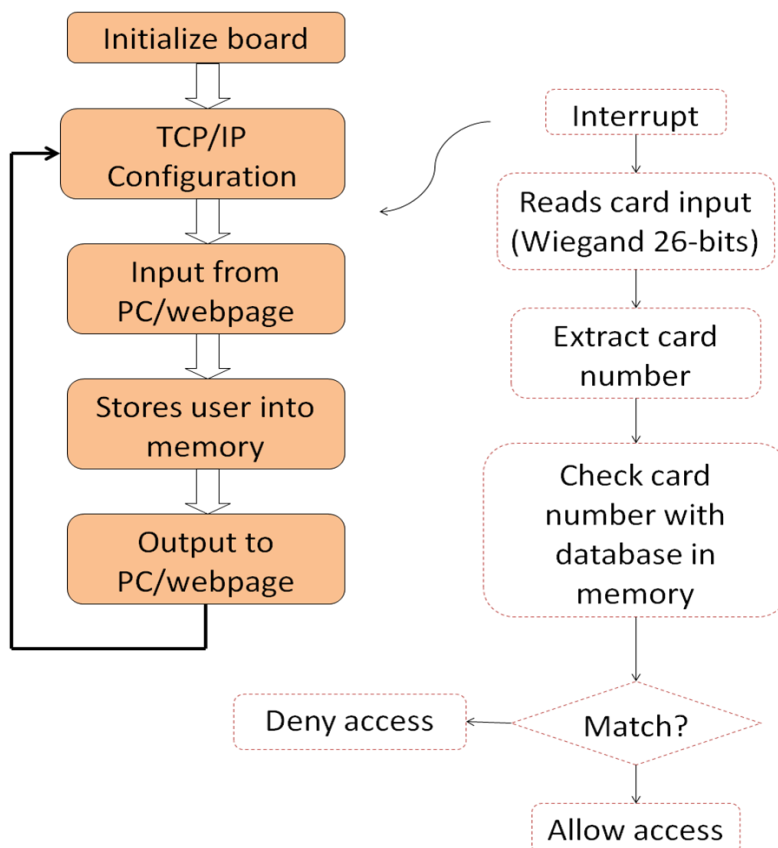


Figure 11 : Program Flowchart

From figure 11, the flowchart shows how the entire program flows. The program starts from board initialization which calibrates all the settings of the program. Then it moves to a permanent loop which consists of TCP/IP configuration, checking input from webpage, and outputs them. The EM card reader will act as an interrupt throughout the endless loop and feeding in 26-bits of information to the microcontroller. The 26-bits that are received are called the Wiegand protocol. Microcontroller will interpret the data and extract the important 16-bits card number and checks them with the existing memory database. If the data matches, then it will grant user the access to the door. Else, the door will not be unlocked, disallowing user to enter the door. The code fragments of reading cards and interrupts that feed in the card information can be seen in Appendix G, I and J respectively.

The following part of report will show the step-by-step of procedures accompanied by the snapshots from the webpage in detailed.



Figure 12 : Layout changes in the webpage design

```

<form method="get" action="index.htm">
<div class="examplebox">
<b>Lock Release Time Selection:</b> <select name="lock"><option value="0" ~lockDelay(0)~>1s</option><option
value="1" ~lockDelay(1)~>3s</option><option value="2" ~lockDelay(2)~>5s</option><option value="3" ~lockDelay
(3)~>10s</option></select>&nbsp;  <br /><input type="submit" value="Save" style="margin-top:5px;">
</div>
</form>

```

Figure 13 : HTML code to construct drop down box in Figure 12

In Figure 12, the first new layout was created by modifying the current hypertext markup language (html) and cascading style sheet (css). The example of the html code that builds up the drop down box for the lock release selection time can be seen in Figure 13. It is located in the main page of the webpage that displays the last modified date, status of the door too.

---



::: Wired LAN door access controller with web-based interface :::

[Home](#)  
[Dynamic Variables](#)  
[Add/Delete User](#)

## Add/Delete User

Please insert the card number to be added or deleted:

**ADD USER:**

**DEL USER:**

---

Figure 14 : Add/Delete User Page

On the left section of the webpage as seen in figure 14, there is a tab for “Add/Delete User” which allows the administrator to add and delete the users’ data from the memory. The data here refers to the 16-bits card numbers which are stored in a 512Kbits SPI EEPROM. The details of the HTML codes can be referred at figure 15.

```

~inc:header.inc~
<div id="content">

<h1>Add/Delete User</h1>

<p>Please insert the card number to be added or deleted:</p>
<form method="post" action="forms.htm">

<div class="examplebox">
<b>ADD USER:</b> <input type="text" name="lcdADD" maxLength="32"> &nbsp; <input type="submit" value="ADD">
</div>
</form>

<form method="post" action="forms.htm">

<div class="examplebox">
<b>DEL USER:</b> <input type="text" name="lcdDEL" maxLength="32"> &nbsp; <input type="submit" value="DEL">
</div>
</form>
</div>
<script type="text/javascript">
<!--
document.getElementById('hello').innerHTML = "~hellomsg~";
-->
</script>
~inc:footer.inc~

```

Figure 15 : HTML codes that builds up the box for administrator to add or delete user.

Figure 15 shows the HTML codes that build up “Add/Delete user” page. The lcdAdd and lcdDEL will be a label or variable name observed in the MPLAB C code. The associated C code is attached in Appendix H. From the input given by the administrator, it will be processed in the microcontroller and therefore convert it from string to hexadecimal. This hexadecimal value will then be stored in the stated SPI EEPROM as a database of users.

Next significant modification that has been successfully made would be the static allocation of IP address by utilizing the unique MAC address from each device. This can be set through the router page setting. In order to access to the router page setting, generally one would need to go to 192.168.1.1 on the address bar of a web browser. On the router configuration, the administrator will have the full authority to change the setting and thus, able to add in the MAC address and IP address pair.

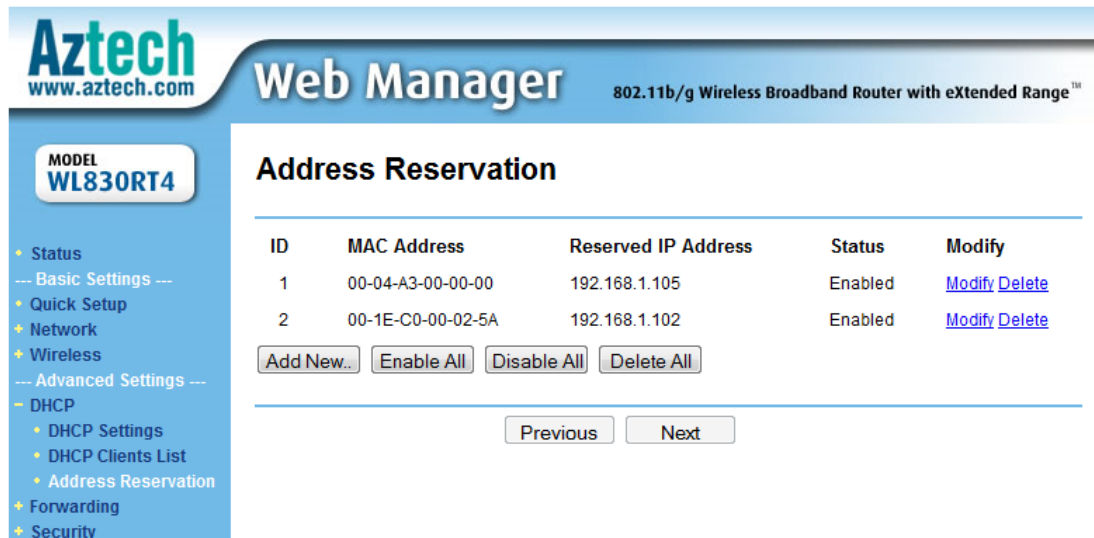


Figure 16 : Configuration page of a router, on Address Reservation

As seen in figure 16, that is the snapshot of the picture taken from the router configuration page. By accessing to DHCP settings on that page, administrator will be able to go to Address Reservation (which means static address allocation method) to pair up the MAC address and IP address. The MAC address can be obtained either by referring to the device label, or administrator can check it from the router configuration page.

**Aztech**  
www.aztech.com

**Web Manager**

802.11b/g Wireless Broadband Router with eXtended Range™

MODEL  
**WL830RT4**

- Status
- Basic Settings ---
- Quick Setup
- Network
- Wireless
- Advanced Settings ---
- DHCP
  - DHCP Settings
  - DHCP Clients List
  - Address Reservation
- Forwarding

### DHCP Clients List

ID	Client Name	MAC Address	Assigned IP	Lease Time
1	Wensher-UTP	00-22-69-B5-DA-46	192.168.1.100	01:52:30

Refresh

Figure 17 : Configuration page on DHCP Clients List, with the MAC Address.

Referring to figure 17, the snapshot shows the DHCP Clients List in a table. The list provides information of the clients that connects to the server. If there is more than a device, then the list will display all. From here, the MAC address of a connected device can be obtained. This info will be useful to be used in the address reservation as seen in figure 16.

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATIONS**

#### **5.1 Conclusion**

From this report, it can be concluded that the project is able to meet the objective which is to design and develop an access controller board which is able to connect through Ethernet and allows user to make monitor and modify their settings. The project is able to allow webpage monitoring and allowing each device to have their assigned IP address as their identification. Each board will have their own storage since they have external EEPROM installed on its own. This could distinguish the memory storage for different doors, if there is a need to give different access of location to different users.

Static IP address allocation is made possible by accessing through the router configuration page. It is a good solution for a device to ensure that it has the same IP throughout the operation time. This, in result, will help the administrator to govern and maintain the respective device. If IP addresses keeps changing dynamically, there are possibilities that the administrator will lose track of the devices on the server. It will be more time consuming to trackback the IP address if there are a huge number of devices connected together. The usage of a router is needed as a medium to connect all devices and communicates with the control centre. MAC address provides unique identification for each devices, thus lessen the problem of distinguishing two same devices.

## **5.2 Recommendations**

There can be more modifications that can be done to enhance the features of this project. There can be more type of input devices that can be attached to the board as long as they are able to verify the identification. Some of the examples are keypad, biometrics and more.

Looking in terms of the network, more security enhancement can be added to the project, such as user ID and password to enter the network. That can be explained as network key which is required before they allow entry to the settings page.

## REFERENCES

- [1] Microsoft.com (2010). DHCP (Dynamic Host Configuration Protocol) Basics. Retrieved from <http://support.microsoft.com/kb/169289>
- [2] RS232 Specifications and Standard. (2010). Retrieved from [http://www.lammertbiers.nl/comm/info/RS-232\\_specs.html](http://www.lammertbiers.nl/comm/info/RS-232_specs.html)
- [3] Dux Computer Digest. (17/6/2001) 100BASE-TX Ethernet Unshielded Twisted-Pair (UTP) Cable Maximum and Minimum Length Requirements. Retrieved from <http://www.duxcw.com/faq/network/cabling.htm>
- [4] Bradley Mitchell. (2010). DHCP - Dynamic Host Configuration Protocol. Retrieve from [http://compnetworking.about.com/cs/protocolsdhcp/g/bldef\\_dhcp.htm](http://compnetworking.about.com/cs/protocolsdhcp/g/bldef_dhcp.htm)
- [5] John Wobus. (10/26/1998).DHCP FAQ. Retrieved from [http://www.dhcp-handbook.com/dhcp\\_faq.html#widxx](http://www.dhcp-handbook.com/dhcp_faq.html#widxx)
- [6] Wikipedia. (21/9/2010). MAC Address. Retrieved from [http://en.wikipedia.org/wiki/MAC\\_address](http://en.wikipedia.org/wiki/MAC_address)
- [7] Cisco, Internetworking Technology Handbook. (2010) Open Systems Interconnection (OSI) Protocols. Retrieved from <http://www.cisco.com/en/US/docs/internetworking/technology/handbook/OSI-Protocols.html>
- [8] Hectrix LTD. (2005) Wiegand Application Note. Retrieved from <http://forums.parallax.com/attachment.php?attachmentid=74469&d=1287163230>

- [9] MicroController Pros Corporation. (16/12/2008) Microcontroller Interfaces, Part 1. Retrieved from <http://www.ucpros.com/work%20samples/Microcontroller%20Communication%20Interfaces%201.htm>
  
- [10] Microchip (2009) Overview and Use of the PICmicro Serial Peripheral Interface. Retrieved from [http://ww1.microchip.com/downloads/en/device doc/spi.pdf](http://ww1.microchip.com/downloads/en/device%20doc/spi.pdf)
  
- [11] Wikipedia. (2010) Wiegand Interface. Retrieved from [http://en.wikipedia.org/wiki/Wiegand\\_protocol](http://en.wikipedia.org/wiki/Wiegand_protocol)

## **APPENDICES**

## **APPENDIX A**

### **ELECTROMAGNETIC CARD READER**



## APPENDIX B

### ELECTROMAGNETIC CARDS



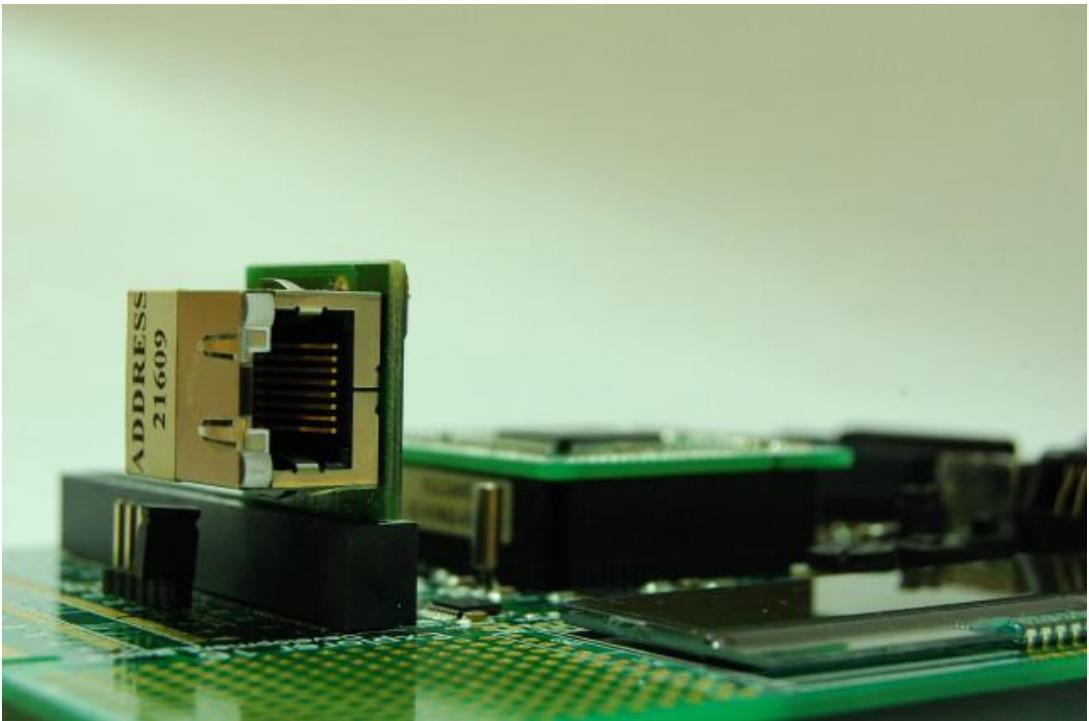
## APPENDIX C

### EXPLORER16 DEVELOPMENT BOARD



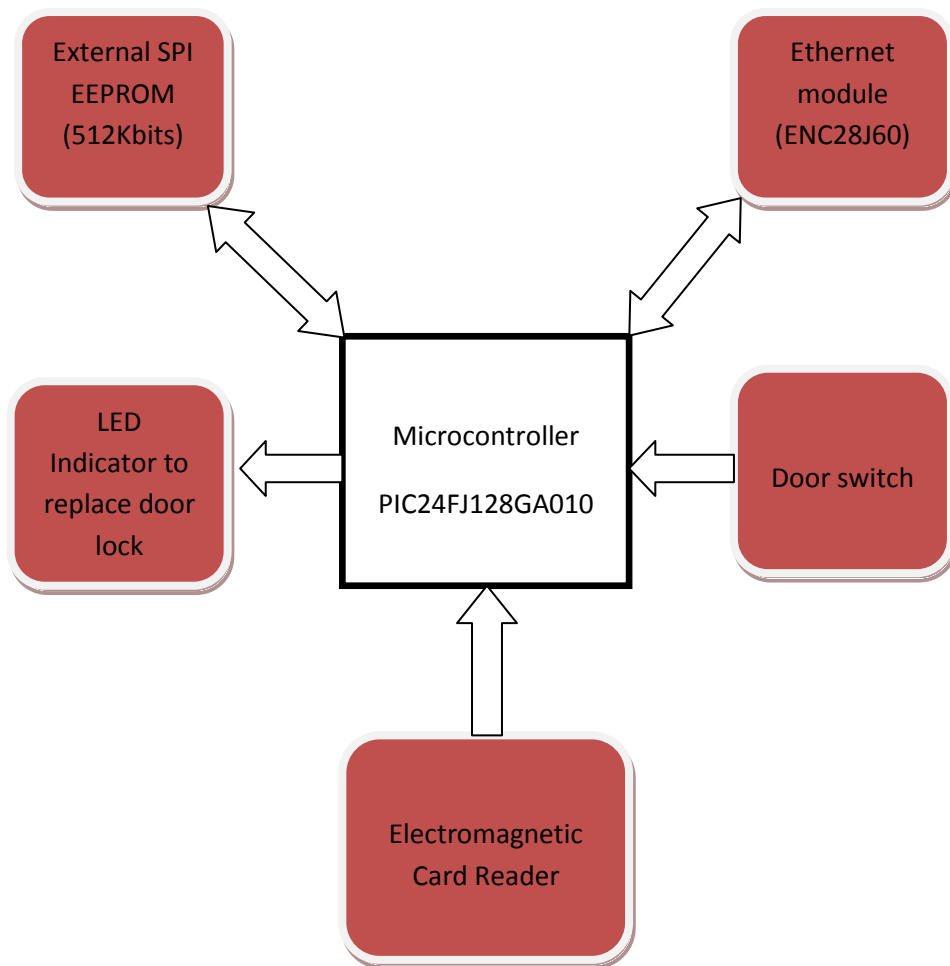
## APPENDIX D

### ETHERNET MODULE (ENC28J60)



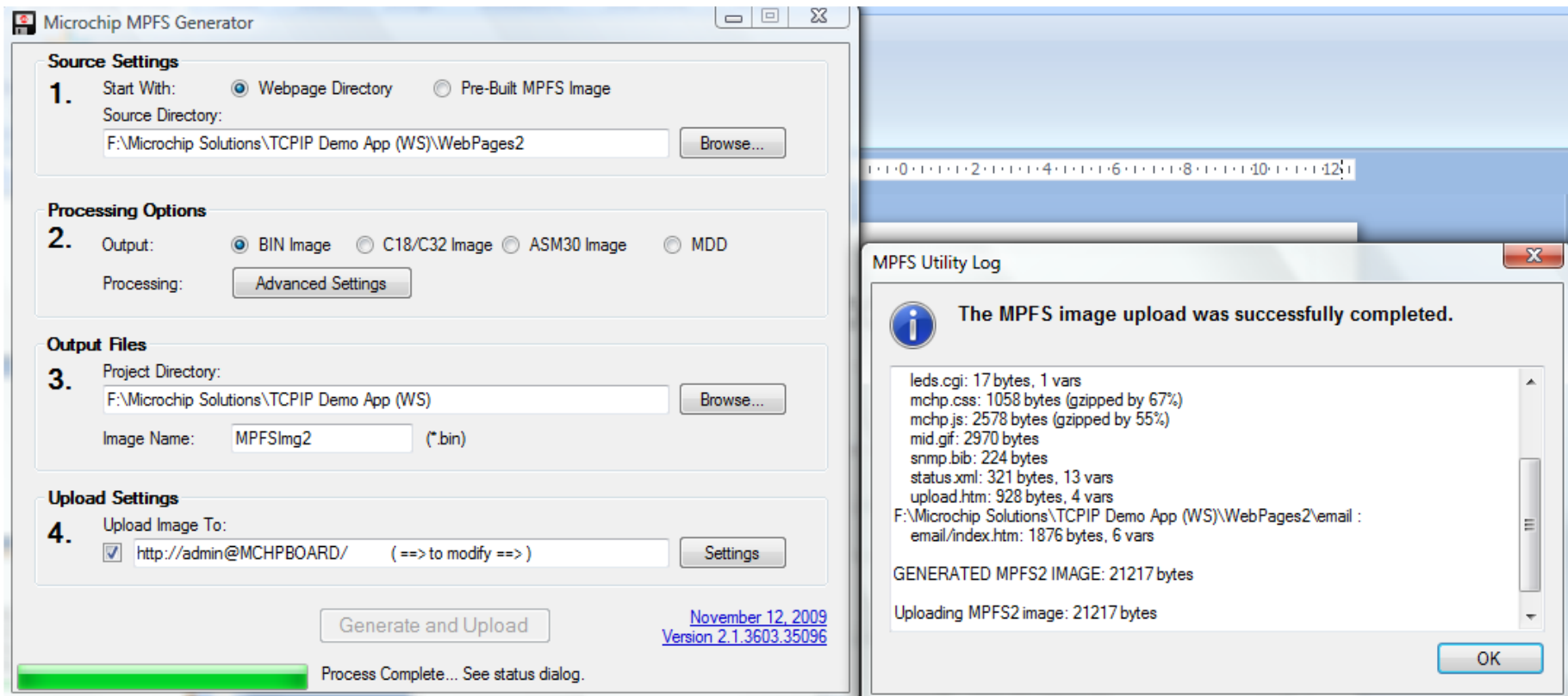
## APPENDIX E

### INPUTS AND OUTPUTS TO MCU



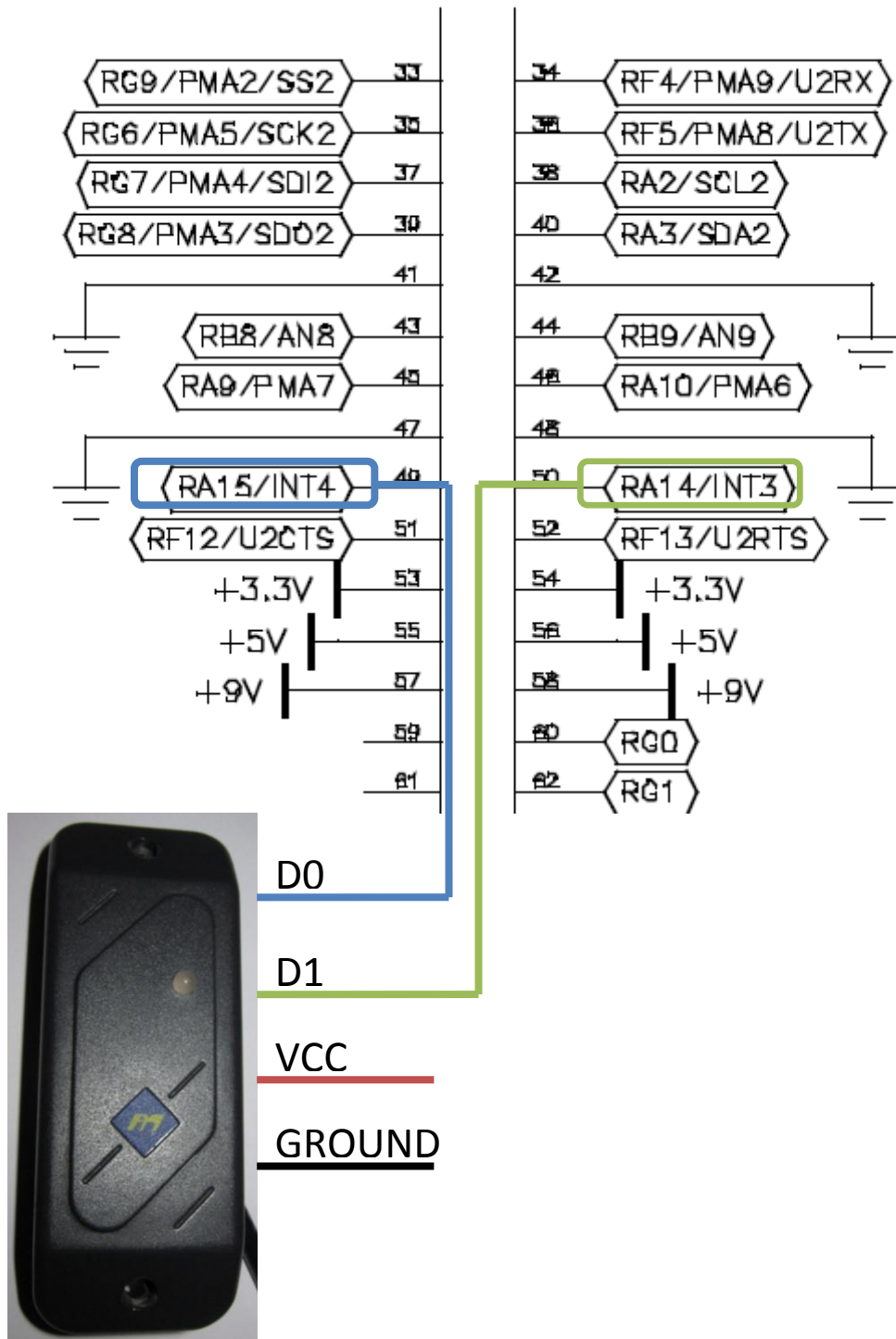
## APPENDIX F

### MICROCHIP FILE SYSTEM GENERATOR



## APPENDIX G

### SCHEMATIC DIAGRAM FOR WIEGAND LINE



## APPENDIX H

### INTERRUPT FOR CHANNEL D1 OF WIEGAND

```
void __attribute__((interrupt, auto_psv)) _INT3Interrupt(void)
{
    IFS3bits.INT3IF = 0;          //clear interrupt for INT3

    if(PORTAbits.RA14 == 0)      //D1
    {
        if(WiegandCounter==0)
        {
            MSBParity=1;
        }
        else if(WiegandCounter==25)
        {
            LSBParity=1;
        }
        else
        {
            NewWiegand = (NewWiegand<<1)|0x0001;
        }
        WiegandCounter++;
    }

    if(WiegandCounter==10)
    {
        NewWiegandH = NewWiegand;
        NewWiegand=0;
    }
    else if(WiegandCounter==26)
    {
        NewWiegandL = NewWiegand;
        NewWiegand=0;
        if((NewWiegandH&0x0001)==1)
        {
            NewWiegandL=NewWiegandL|0x8000;
        }
    }
}
```

## APPENDIX I

### INTERRUPT FOR CHANNEL D0 OF WIEGAND

```
void __attribute__((interrupt, auto_psv)) _INT4Interrupt(void)
{
    IFS3bits.INT4IF = 0;
    if(PORTAbits.RA15 == 0) //D0
    {
        if(WiegandCounter==0)
        {
            MSBParity=0;
        }
        else if(WiegandCounter==25)
        {
            LSBParity=0;
        }
        else
        {
            NewWiegand = NewWiegand<<1;
        }
        WiegandCounter++;
    }
    if(WiegandCounter==10)
    {
        NewWiegandH = NewWiegand;
        NewWiegand=0;
    }
    else if(WiegandCounter==26)
    {
        NewWiegandL = NewWiegand;
        NewWiegand=0;
        if((NewWiegandH&0x0001)==1)
        {
            NewWiegandL=NewWiegandL|0x8000;
        }
    }
}
```

## APPENDIX J

### C CODE FOR READING CARD IN MAIN FUNCTION

```
while(1)
{
    BYTE* cDest2;
    stop=0;

    if(WiegandCounter==26)
    {
        //Delay_Counter=DelayVal;
        WiegandCounter=0;

        ultoa(NewWiegandL, card);    //convert hex to string
        cDest2 = card;
        curr_lowerNib = NewWiegandL&0xff;
        LowerNibble=NewWiegandL;
        UpperNibble=NewWiegandL>>8;

        userAddress_read=0x0000;    //initial read address
        XEEBeginRead(userAddress_read);
        stop=0;

        while(stop==0)
        {

            readResult = XEERead();    //Reads first time for upper nibble
            //stop=0; //dummy instructions for debugging purposes.

            if(readResult==UpperNibble)//If UpperNibble is correct, check for LowerNibble
            {
                readResult = XEERead();    //Reads again for LowerNibble

                if(readResult==LowerNibble)    //break while loop if matches with database
                {
                    DOOR_OPEN_IO=1;    //open door if correct
                    LED1_IO=1;
                    stop=1;    //come out from loop
                }
            }
        }
    }
}
```

```

        // print the card number into LCD
        if(strlen((char*)cDest2) < 32u)
        {
            memset(LCDText, ' ', 32);
            strcpy((char*)LCDText, (char*)cDest2); //right now, cDest2 is holding the card info
            RecopyLCDData = (char*)&LCDText; //copy value of cDest into RecopyLCDData
        }
        else
        {
            memcpy(LCDText, (void *)cDest2, 32);
        }
        LCDUpdate();

        Delay_Counter=DelayVal;
    }

    else if(readResult!=LowerNibble)
    {
        userAddress_read+=2;
    }
    //upper nibble

    else if(readResult!=UpperNibble)
    {
        userAddress_read+=2;
    }

    if(userAddress_read==0x0150) //for sure the card is not in database (out of limit)
    {
        stop=1;
    }
    //end of while(stop==0) loop
    stop=0;
    XEEEndRead();

    //end of if(Wiegand==26)

//Display Delay
if(Delay_Counter>1)
{
    --Delay_Counter;
}

```

```

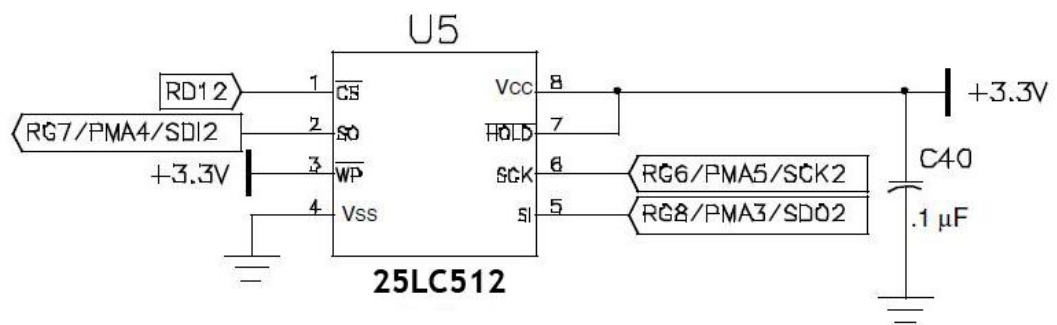
else if(Delay_Counter==1)
{
strcpypgm2ram((char*)LCDText, "TCPStack " VERSION " "
"          ");
LCDUpdate();
DisplayIPValue(AppConfig.MyIPAddr);
Delay_Counter=0;

DOOR_OPEN_IO=0;
LED1_IO=0;
}
.....(other codes)
} //end of main loop

```

## APPENDIX K

### SCHEMATIC DIAGRAM OF EEPROM (25LC512)



## APPENDIX L

### SOURCE CODE FOR READ FROM EEPROM

```

/*****
* Function:    XEE_RESULT XEEBeginRead(DWORD address)
* PreCondition:  None
* Input:       address - Address at which read is to be performed.
* Output:      XEE_SUCCESS
* Overview:    Sets internal address counter to given address.
*****/

XEE_RESULT XEEBeginRead(DWORD address)
{
    // Save the address and empty the contents of our local buffer
    EEPROMAddress = address;
    vBytesInBuffer = 0;
    return XEE_SUCCESS;
}

/*****
* Function:    BYTE XEERead(void)
* PreCondition:  XEEInit() && XEEBeginRead() are already called.
* Input:       None
* Output:      BYTE that was read
* Overview:    Reads next byte from EEPROM; internal address is incremented by one.
*****/

BYTE XEERead(void)
{
    // Check if no more bytes are left in our local buffer
    if(vBytesInBuffer == 0u)
    {
        // Get a new set of bytes
        XEEReadArray(EEPROMAddress, EEPROMBuffer, EEPROM_BUFFER_SIZE);
        EEPROMAddress += EEPROM_BUFFER_SIZE;
        vBytesInBuffer = EEPROM_BUFFER_SIZE;
    }
    // Return a byte from our local buffer
    return EEPROMBuffer[EEPROM_BUFFER_SIZE - vBytesInBuffer--];
}

```

## APPENDIX M

### SOURCE CODE FOR WRITE TO EEPROM

```

/*****
* Function:    XEE_RESULT XEEBeginWrite(DWORD address)
* PreCondition:  None
* Input:       address    - address to be set for writing
* Output:      XEE_SUCCESS
* Overview:    Modifies internal address counter of EEPROM.
* Note:       Unlike XEESetAddr() in xeprom.c for I2C EEPROM memories, this function is used only for
*             writing to the EEPROM. Reads must use XEEBeginRead(), XEERead(), and XEEEndRead().
*             This function does not use the SPI bus.
*****/

XEE_RESULT XEEBeginWrite(DWORD address)
{
    vBytesInBuffer = 0;
    EEPROMAddress = address;
    return XEE_SUCCESS;
}

/*****
* Function:    XEE_RESULT XEEWrite(BYTE val)
* PreCondition:  XEEInit() && XEEBeginWrite() are already called.
* Input:       val - Byte to be written
* Output:      XEE_SUCCESS
* Overview:    Writes a byte to the write cache, and if full, commits the write. Also, if a write boundary is
*             reached the write is committed. When finished writing, XEEEndWrite() must be called to
*             commit any unwritten bytes from the write cache.
*****/

XEE_RESULT XEEWrite(BYTE val)
{
    EEPROMBuffer[vBytesInBuffer++] = val;
    if(vBytesInBuffer >= sizeof(EEPROMBuffer))
        DoWrite();
    else if((((BYTE)EEPROMAddress + vBytesInBuffer) & (EEPROM_PAGE_SIZE-1)) == 0u)
        DoWrite();

    return XEE_SUCCESS;
}

```

## APPENDIX N

### C CODE FOR ADD/DELETE USER

```
// Copy up to 32 characters to the LCD
if(strlen((char*)cDest) < 32u)
{
    memset(LCDText, ' ', 32);
    strcpy((char*)LCDText, (char*)cDest);
    //The line that allow the text that entered to be displayed on webpage
    RecopyLCDDData = (char*)&LCDText;
}
else
{
    memcpy(LCDText, (void *)cDest, 32);
}
LCDUpdate();

//Added 21st Oct 2010 - To convert string to HEX and add/del user
String_Hex = atol(RecopyLCDDData);
LowerNibble = String_Hex;
UpperNibble = String_Hex>>8;

if(ADD_USER==1)          //add user
{
    //write card number to EEPROM
    XEEBeginWrite(userAddress_write);
    XEEWrite(UpperNibble);
    XEEWrite(LowerNibble);

    if(userAddress_write==0x0150)
    {
        userAddress_write=0x0000;
    }
    else
    {
        userAddress_write+=2;
    }
    LED7_IO=1;
    XEEEndWrite();
}
else if (DEL_USER==1)    //delete user. but must scan memory first only delete
{
```

```

stop=0;
userAddress_read=0x0000;
XEEBeginRead(userAddress_read);
//DOOR_OPEN_IO=1;

while(stop==0)
{
readResult = XEERead();    //Reads first time for upper nibble
stop=0;                    //dummy instructions for debugging purposes.

if(readResult==UpperNibble)//If UpperNibble is correct, check for LowerNibble
{
    readResult = XEERead();    //Reads again for LowerNibble
    if(readResult==LowerNibble)    //break while loop if matches with database
    {
        userAddress_read=userAddress_read-1;    //clear memory
        XEEBeginWrite(userAddress_read);    //clear memory
        XEEWrite(0x00);
        XEEWrite(0x00);
        XEEEndWrite();
        stop=1;
    }
    else if(readResult!=LowerNibble)
    {
        userAddress_read+=2;
    }
}
else if(readResult!=UpperNibble)
{
    userAddress_read+=2;
}

if(userAddress_read==0x0150)    //for sure the card is not in database (out of limit)
{
    stop=1;
}
}
stop=0;
XEEEndRead();
}
ADD_USER=0;
DEL_USER=0;

```

## APPENDIX O

### SOURCE CODE TO ADD/DEL USER FROM WEBPAGE

```
// If value is expected, read it to data buffer,
// otherwise ignore it (by reading to NULL)
if(!strcmppgm2ram((char*)curHTTP.data, (ROM char*)"lcdADD"))||(!strcmppgm2ram((char*)curHTTP.data,
(ROM char*)"lcdDEL"))
{
    cDest = curHTTP.data;

    if(!strcmppgm2ram((char*)curHTTP.data, (ROM char*)"lcdADD")) //if user is added
    {
        ADD_USER=1;
        DEL_USER=0;
    }
    else if(!strcmppgm2ram((char*)curHTTP.data, (ROM char*)"lcdDEL")) //if user is deleted
    {
        // String_Hex = atol((char*)curHTTP.data);
        DEL_USER=1;
        ADD_USER=0;
    }
}
else
    cDest = NULL;
```