SMART RECYCLE AND REWARD BIN

MELISSA LIM SIEW SEAN

A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor (Hons.) of Electronic Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2011

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ____________________________

Name : ____________________________

ID No. : ____________________________

Date : ____________________________

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"SMART RECYCLE AND REWARD BIN"** was prepared by **MELISSA LIM SIEW SEAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ____________________________

Supervisor : ____________________________

Date : ____________________________

Specially dedicated to

My beloved parents and sisters.

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Mr Ng Choon Boon for his invaluable advice and guidance throughout the development of this project.

In addition, I would also like to express my gratitude to my loving parents for their support and encouragement which gave me much strength and determination needed to complete this project. Also, special thanks to my two teammates, Ng Chun Hoe and Teh Khee Leong who had been helpful and constantly sharing their ideas and knowledge.

My project would not have been much success without the support and guidance from all of you. Thank you.

SMART RECYCLE AND REWARD BIN

ABSTRACT

The Smart Recycle and Reward Bin serves as the prototype for a Reverse Vending Machine (RVM). RVM is an innovative concept which has been introduced to western countries to help collect recycling materials and hence, to boost recycling activities. Our solar-powered prototype enables the user to recycle aluminium, glass and plastic beverage containers and gets reward points. The points for recycling is rewarded based on the beverage container material and are summed up before storing into a smart card. The electronic reward system is implemented using MIFARE technology contactless smart card platform which enables writing and reading of data into the card. Microsoft Visual Basic (VB) is used to develop a PC-based application for the reward system. Data exchange between the PC and smart card reader/writer is facilitated through a serial port using VB's MSComm control. The program incorporates a user-friendly Graphical User Interface (GUI) to serve the purpose of updating the smart card, tracking the reward points using a database and redemption of points. The database system is developed using Microsoft Access and the information stored is obtained from the Visual Basic program using ADO Data Control function. Also, the program allows sending of data from a microcontroller-based board in the recycle bin to the PC using UART. The approach used has proven to be successful where data exchange between microcontroller, PC and smart card reader/writer is accurate. C programming is used to program the PIC16F877A microcontroller for displaying messages and prompts on an LCD. Testing results show that the objectives of the project were achieved. Possible future works recommended is to eliminate the use of PC and to optimize the data storage in the smart card. Overall, the system was implemented successfully. The system as a whole provides a cost effective and simple solution for implementation of a Reverse Vending Machine.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

LCD	Liquid Crystal Display
PC	Personal Computer
RVM	Reverse Vending Machine
UART	Addressable Universal Asynchronous Receiver Transmitter
CPU	Central Processing Unit
RF	Radio Frequency
UHF	Ultra High Frequency
TTL	Transistor-Transistor Logic

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Reverse Vending Machine (RVM) is just like any vending machine available but because of its name with the additional ‘reverse’, the machine gives reward such as money and shopping coupons in exchange of certain materials recognized when given to the machine (Oxford Dictionary, 1997). Unlike normal vending machines where one has to insert money to get the product.

RVM is an innovative concept which has been introduced to western countries to help collect recycling materials and hence, to boost recycling activities. These machines, depending on types, have the ability to accept materials such as beverage cans, bottle glasses and plastics. RVM generally automates container recycling by accepting containers directly from the consumer, accounting for each container processed, and refunding the deposit to the consumer.

TOMRA is the world’s leading manufacturer of reverse vending machines for recycling beverage containers and is also a pioneer in its field. The company produces many types of RVM which have many abilities.

The benefits of recycling of the used beverage containers TOMRA collected in one hour proves to save the same amount of carbon dioxide emissions that would be created by driving a car 17 times around the world at the equator (Young, 2005).

Recycling of the used aluminium cans TOMRA collects in a day saves the energy that corresponds to what it takes to supply 400 US households with electricity for a year (Young, 2005). The recycling of only two percent of the containers collected by TOMRA's reverse vending machines offsets the entire direct carbon dioxide emissions created by the organization worldwide.

In order to identify all the different beverage containers by their shape, weight, material and/or barcode, the real-time image processing system is developed by TOMRA. The image processing is developed with a video camera which takes 25 pictures per second of the bottle as it is placed into the machine. This unique identification system guarantees that the customer always receives the proper refund for the returned package and ensures that the machine cannot be fooled.

Besides, additional functions of the RVM such as the ability of language selection, easier user interface where icon-based interface that is controlled through a large, colour touch-screen display and many flexible backroom systems that will help to optimize storage and compaction of the empty containers are also available.

Other companies such as the Reverse Vending Systems Company and ENVIPCO, uses other method when detecting the containers. These companies use some specific scanner named 3 UPC Scanners, where it has the ability to scan the beverage container's UPC code to recognize the recyclable materials.

1.2 Motivation

Recycling means converting wastes to useable materials (Oxford Dictionary, 1997). It is fully implemented in western countries with the support of its nations. However, in our country, littering is a norm. Local authorities encourage recycling programs but only to get lukewarm responses.

One of the programs executed is by placing recycling bins at public places. However, according to an article on Promoting Recycling Culture in The New Straits Times Press, the programme is a failure as those specially designed function bins as shown in Figure 1.1, are thrown with rubbish just like any other rubbish bins seen around (Leong, 1998). For solely this reason, we can see that the inconvenience and ineffectiveness of the recycling process demotivates people from practicing it.

Figure 1.1: Recycle Bins

According to the director of Treat Every Environment Special Sdn Bhd, or TrEEs, Christa Hashim stated the best system to boost recycling is by paying cash or shopping coupons to those who bring recyclable items to the stations located at shopping malls (Leong, 1998). However, the programme cannot be extended or fully implemented in our country as it requires a full team to manage it.

Hence, Reverse Vending Machine (RVM) is meant to encourage recycling habit by giving rewards to recyclers for every recycled item in terms of reward points. Realizing the advantages of RVM, many countries have implemented these machines. Even though Malaysia initiates ways to reduce waste, RVM is not used because of its high implementation cost and maintenance. The major motivation for this project is to enable the implementation of RVM in Malaysia by building a prototype which focuses on reduction in energy consumption and paper usage as well as cost saving.

1.3 Aims and Objectives

This project aim is to develop a solar powered prototype of Reverse Vending Machine (Smart Recycle and Reward Bin). This prototype enables the user to recycle aluminium, glass and plastic beverage containers and gets reward points. Through sensor systems, the container material is recognized and it is then dropped into respective bins through the aperture. In return, reward points are credited into stored value card according to items recycled.

The objectives of the project are:

i) To implement Mifare contactless smartcard as the electronic reward system.

ii) To implement a user friendly interface and display system.

iii) To create a database for authorized personnel to keep track of recycling activities, undertake remote problem solving and error checking.

1.4 Scope of Work

The scope of our project is divided into three main parts. Solar powered system is done by Teh Khee Leong and material recognition system is done by Ng Chun Hoe. The author implements the electronic reward system using smart card to store values earned by recycling instead of using coupons and receipts. The work was carried out simultaneously where Teh Khee Leong ensured that the solar system provides enough power for the whole prototype. Also, he developed the mechanical design of our prototype which accommodates the placement of sensors and separation of bins for different container materials. Ng Chun Hoe developed the sensor circuits used to identify the container materials, initiates the placement of sensors and implemented the relay for EM lock. Meanwhile, the author developed the electronic reward system using MIFARE technology and the display system. Programs are written using Microsoft Visual Basic and C programming.

CHAPTER 2

LITERATURE REVIEW

This section is to discuss the theoretical aspects leading to the implementation of the project. Review involves the recent implementation of the Reverse Vending Machine and the related technologies used. Materials such as journal, books and technical papers are referred.

2.1 Reverse Vending Machine

A Reverse Vending Machine is a device that accepts used beverage containers and returns money to the user (the reverse of the typical vending cycle). The machines are popular in places that have mandatory recycling laws or container deposit legislation.

The basic operations involve steps where the recycler places the empty bottle/can into the receiving aperture; the horizontal in-feed system allows the user to insert containers one at a time. The bottle/can is then automatically rotated; the bottle/can is then scanned by an Omnidirectional UPC Scanner, which scans the beverage container's UPC code to identify the manufacturer, product and the composition of container. To improve recognition, RVM also may include camera scanning system.

As for the reward system, RVM distributes valuable tokens, like coins or coupons, when beverage containers are recycled. The coupons are then used to redeem gifts at the counter. However, the awareness of environmental issues and the aim of reducing paper usage, the printing of coupons is not highly favoured.

Current application is by using electronic rewarding system. Personalized magnetic card or smart card system issues the users reward points. Electronic rewarding system also increases customer loyalty. Most supermarkets have issued loyalty cards to customer to store the reward points. The system may be integrated with other types of personal accounts, such as school ID or library card. Thus, this approach fully optimizes the use of card system and user convenience.

The user interface of RVM from Reverse Vending Corporation is a LCD screen to display video messages on the 17"/19" LCD monitor on the front panel of the unit. The RVM also has the capabilities to read the barcode of the product inserted and play a video which corresponds with this product.

Reverse Vending Machine has beverage container database to verify whether a bottle should be accepted or rejected. For the countries that do not have a national container database, the store can also alter and update the individual container and create database. The Reverse Vending Machines can also be accessed remotely allowing the trained technician, to undertake remote problem solving, error checking, updating of software, updating of barcode database and identification of material type and flow.

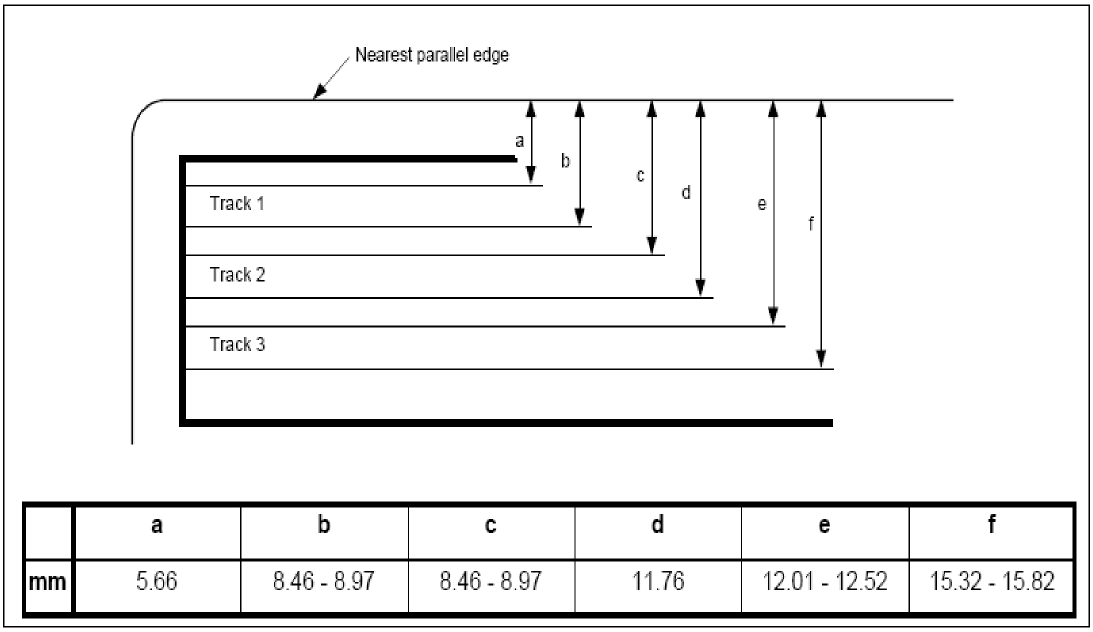
2.2 Types of Cards

As mentioned, a personalized stored-value card can be used to implement the reward system in RVM. The term stored-value card means the value data is physically stored on the card.

2.2.1 Magnetic Strip Cards

The data is stored in magnetic stripes on the card. The stripe can be encoded because the particles can be magnetized in either a north or south pole direction. By changing the direction of the encoding along the length of the stripe this allows information to be written on the stripe.

Data is laid out on a standard magnetic card in three tracks. A magnetic stripe card may have any of these tracks, or a combination of these tracks.



	a	b	c	d	e	f
mm	5.66	8.46 - 8.97	8.46 - 8.97	11.76	12.01 - 12.52	15.32 - 15.82

Figure 2.1: Layout of data tracks

The advantages of magnetic stripe include:

- Data can be modified or rewritten
- High data capacity in relation to bar codes
- Add security since it is not in human readable form
- Immune to contamination with dirt, water, oil, moisture, etc
- No moving components, physically robust

Some disadvantages of using magnetic stripe cards:

- It doesn't work in a distance, thus requiring close contact to the reader
- Data can be damaged by stray magnetic fields
- Since it's not in human readable form can be a disadvantage in some applications.

2.2.2 Smart Cards

A smart card, typically a type of chip card, is a plastic card that contains an embedded computer chip, either a memory or microprocessor type that stores and transacts data. This data is usually associated with either value, information, or both and is stored and processed within the card's chip. The card data is transacted via a reader that is part of a computing system (Wikipedia, 2010)

. Smart cards improve the convenience and security of any transaction. They provide tamper-proof storage of user and account identity. Smart card systems have proven to be more reliable than other machine-readable cards, like magnetic stripe and barcode, with many studies showing card read life and reader life improvements demonstrating much lower cost of system maintenance.

Smart cards also provide vital components of system security for the exchange of data throughout virtually any type of network. They protect against a full range of security threats, from careless storage of user passwords to sophisticated system hacks. Smart cards are defined according to how the card data is read/written and the type of chip implanted within the card and its capabilities (Rankl, 1998). The types of smart card is as shown in Figure 2.2.

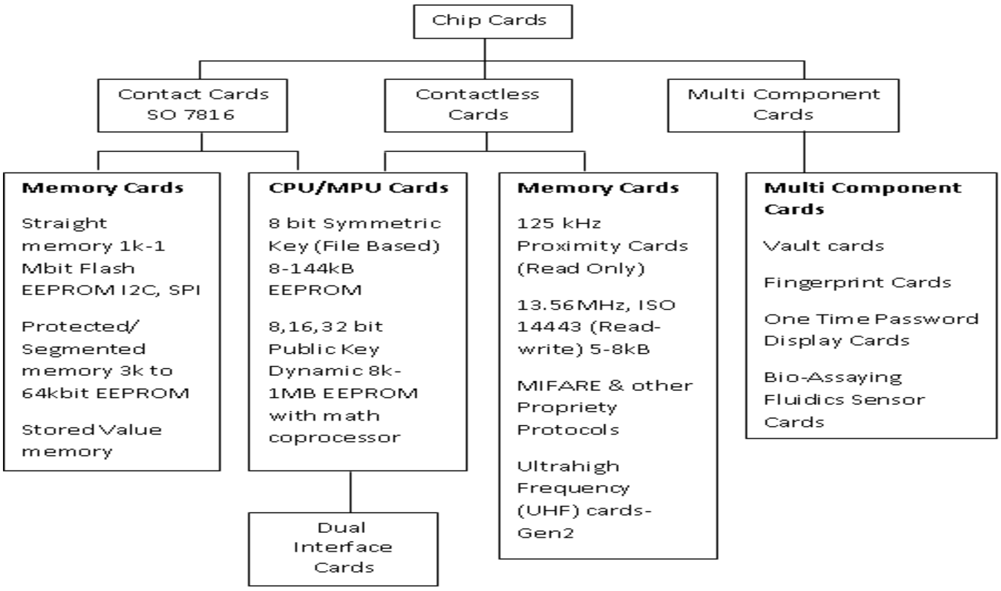

Chip Cards
Contact Cards SO 7816
Contactless Cards
Multi Component Cards
Memory Cards
Straight memory 1k-1 Mbit Flash EEPROM I2C, SPI
Protected/ Segmented memory 3k to 64kbit EEPROM
Stored Value memory
CPU/MPU Cards
8 bit Symmetric Key (File Based) 8-144kB EEPROM
8,16,32 bit Public Key Dynamic 8k-1MB EEPROM with math coprocessor
Dual Interface Cards
Memory Cards
125 kHz Proximity Cards (Read Only)
13.56MHz, ISO 14443 (Read-write) 5-8kB
MIFARE & other Propriety Protocols
Ultrahigh Frequency (UHF) cards-Gen2
Multi Component Cards
Vault cards
Fingerprint Cards
One Time Password Display Cards
Bio-Assaying Fluidics Sensor Cards

Figure 2.2: Types of smart card (Cardlogix, 2010).

As shown in Figure 2.2., smart cards currently come in two forms, contact and contactless.

- Contact cards require a reader to facilitate the bidirectional connection. The card must be inserted into a device that touches the contact points on the card, which facilitate communication with the card’s chip.

- Contactless cards use proximity couplers to get information to and from the card’s chip. As shown in Figure 2.3, an antenna is wound around the circumference of the card and activated when the card is radiated in a specific distance from the coupler. The configuration of the card’s antenna and the coupler facilitate connected states from a couple of centimetres to a couple of feet.

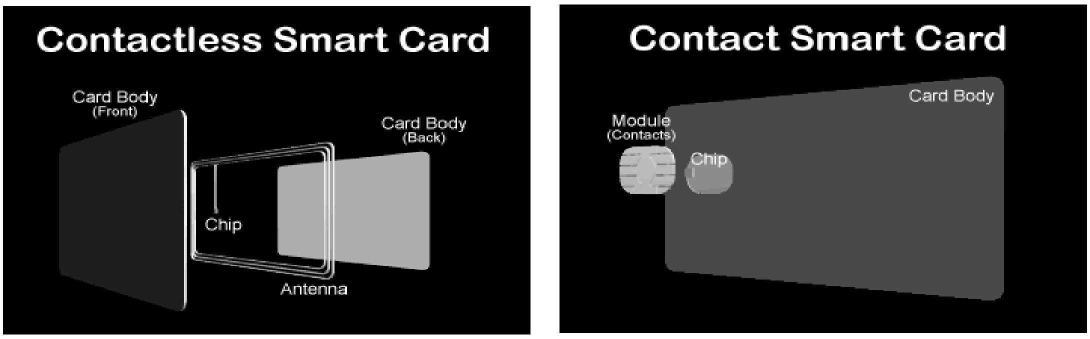


Figure 2.3: Two forms of contact for smart card.

Contact cards include memory cards which cannot manage files and have no processing power for data management. All memory cards communicate to readers through synchronous protocols. In all memory cards, read and write is done to a fixed address on the card. There are three primary types of memory cards: Straight, Protected, and Stored Value. Before designing in these cards into a proposed system the issuer should check to see if the readers and/or terminals support the communication protocols of the chip. Most contactless cards are variants on the protected memory/segmented memory card idiom.

Straight Memory Cards just store data and have no data processing capabilities. Often made with I2C or serial flash semiconductors, these cards were traditionally the lowest cost per bit for user memory. This has now changed with the larger quantities of processors being built for the GSM market. This has dramatically cut into the advantage of these types of devices. They should be regarded as floppy disks of varying sizes without the lock mechanism. These cards cannot identify themselves to the reader, so the host system has to know what type of card is being inserted into a reader. These cards are easily duplicated and cannot be tracked by on-card identifiers.

Protected/Segmented Memory Cards have built-in logic to control the access to the memory of the card. Sometimes referred to as Intelligent Memory cards, these devices can be set to write- protect some or the entire memory array. Some of these cards can be configured to restrict access to both reading and writing. This is usually done through a password or system key. Segmented memory cards can be divided into logical sections for planned multi-functionality. These cards are not easily duplicated but can possibly be impersonated by hackers. They typically can be tracked by an on-card identifier.

Stored Value Memory Cards are designed for the specific purpose of storing value or tokens. The cards are either disposable or rechargeable. Most cards of this type incorporate permanent security measures at the point of manufacture. These measures can include password keys and logic that are hard-coded into the chip by the manufacturer. The memory arrays on these devices are set-up as decrements or counters. There is little or no memory left for any other function.

Cpu/MPU Microprocessor Multifunction cards have on-card dynamic data processing capabilities. Multifunction smart cards allocate card memory into independent sections or files assigned to a specific function or application. Within the card is a microprocessor or microcontroller chip that manages this memory allocation and file access. Unlike other operating systems, this software controls access to the on-card user memory. This capability permits different and multiple functions and/or different applications to reside on the card, allowing businesses to issue and maintain a diversity of 'products' through the card.

For the card user, multifunction means greater convenience and security, and ultimately, consolidation of multiple cards down to a select few that serve many purposes.

As for contactless cards, these smart cards employ a radio frequency (RFID) between card and reader without physical insertion of the card. Instead, the card is passed along the exterior of the reader and read. Types include proximity cards which are implemented as a read-only technology for building access. These cards function with a very limited memory and communicate at 125 MHz. Another type of limited card is the Gen 2 UHF Card that operates at 860 MHz to 960 MHz.

True read and write contactless cards were first used in transportation applications for quick decrementing and reloading of fare values where their lower security was not an issue. They communicate at 13.56 MHz and conform to the ISO

14443 standard. These cards are often protected memory types. They are also gaining popularity in retail stored value since they can speed up transactions without lowering transaction processing revenues unlike traditional smart cards.

Variations of the ISO14443 specification specify chips from either specific or various manufacturers. The most popular are from NXP-(Philips) with their product MIFARE and Sony.

Multi-mode communication cards have multiple methods of communications, including ISO7816, ISO14443 and UHF gen 2. How the card is made determines if it is a Hybrid or dual interface card. Hybrid cards have multiple chips in the same card. These are typically attached to each interface separately, such as a MIFARE chip and antenna with a contact 7816 chip in the same card. Dual interface cards have one chip controlling the communication interfaces. The chip may be attached to the embedded antenna through a hard connection, inductive method or with a flexible bump mechanism (Cardlogix,2010).

Multi Component cards are for a specific market solution. For example, there are cards where the fingerprint sensor is built on the card. Or one company has built a card that generates a one-time password and displays the data for use with an online banking application. Vault cards have rewriteable magnetic stripes. Each of these technologies is specific to a particular vendor and is typically patented.

Some advantages that smart cards bring (Rankl, 1998):

- It has technical specification standards
- It gives security of information
- It has more organized information
- The process doesn't require paper
- One card can access multiple transactions
- It reduces fraud
- It has high memory capabilities

Smart cards also have some disadvantages, such as:

- Lack of technology to support users
- It is a potential area for computer hackers and computer viruses

2.3 MIFARE Smart Card

MIFARE is the NXP Semiconductors owned trademark of the reputedly most widely installed contactless smart card, or proximity card, technology in the world (Wikipedia, 2011).

MIFARE smart card is used in applications like public transport ticketing where major cities have adopted MIFARE as their e-ticketing solution of choice. The applications include public transportation, access control, event ticketing. These applications show that MIFARE smart card may be a good choice for storing and redemption of reward points.

Being the widely adopted choice of smart card, the advantages of MIFARE card is learnt. MIFARE guarantees anti-collision where an intelligent anti-collision function allows operating more than one card in the field simultaneously. The anti-collision algorithm selects each card individually and ensures that the execution of a transaction with a selected card is performed correctly without data corruption resulting from other cards in the field.

MIFARE smart card is also designed for simple integration and user convenience which could allow complete ticketing transactions to be handled in less than 100 ms (NXP,2008). Thus, the MF1ICS50 card user is not forced to stop at the reader leading to a high throughput at gates and reduced boarding times onto busses. The MIFARE card may also remain in the wallet during the transaction, even if there are coins in it.

Several security measures are also taken like mutual challenge and response authentication, data ciphering and message authentication checks support the protection of the system against various attack scenarios. The UID of the IC as a base of key diversification supports the security concept (NXP,2008). To provide security, mutual three pass authentication (ISO/IEC DIS 9798-2) is used. Individual set of two keys per sector (per application) are used to support multi-application with key hierarchy. Also, unique serial number is given to each device.

CHAPTER 3

METHODOLOGY

This section is to discuss the design methods to complete the project. Main tasks, difficulties and problems are listed. Explanation of the operating principles is given. Also, milestones to be achieved are set throughout the project.

3.1 Project Overview

The project is divided into three main parts which will be done simultaneously. As depicted by the title “Smart Recycle and Reward Bin”, the “Smart” characteristics of the bin is realised by using solar system, and the “Recycle” system is realised by using material recognition system. These two parts are done by my teammates. The author focus is on the “Reward” system which is realized by using MIFARE smart card platform, PC-based application developed using Visual Basic and PIC microcontroller, as shown in Figure 3.1.

We established three moments or “milestones” along the development of our project to ensure that our simultaneous development and teamwork can be achieved:

(i) at the beginning of the project, we made thorough research on our parts in the project and discussed ways to approach the ideas.

(ii) in the middle of the process, when problems in the initial approach can be detected, we discussed necessary corrections and future works.

(iii) a final assessment milestone is when the work is completed and our individual parts are integrated into a complete project.

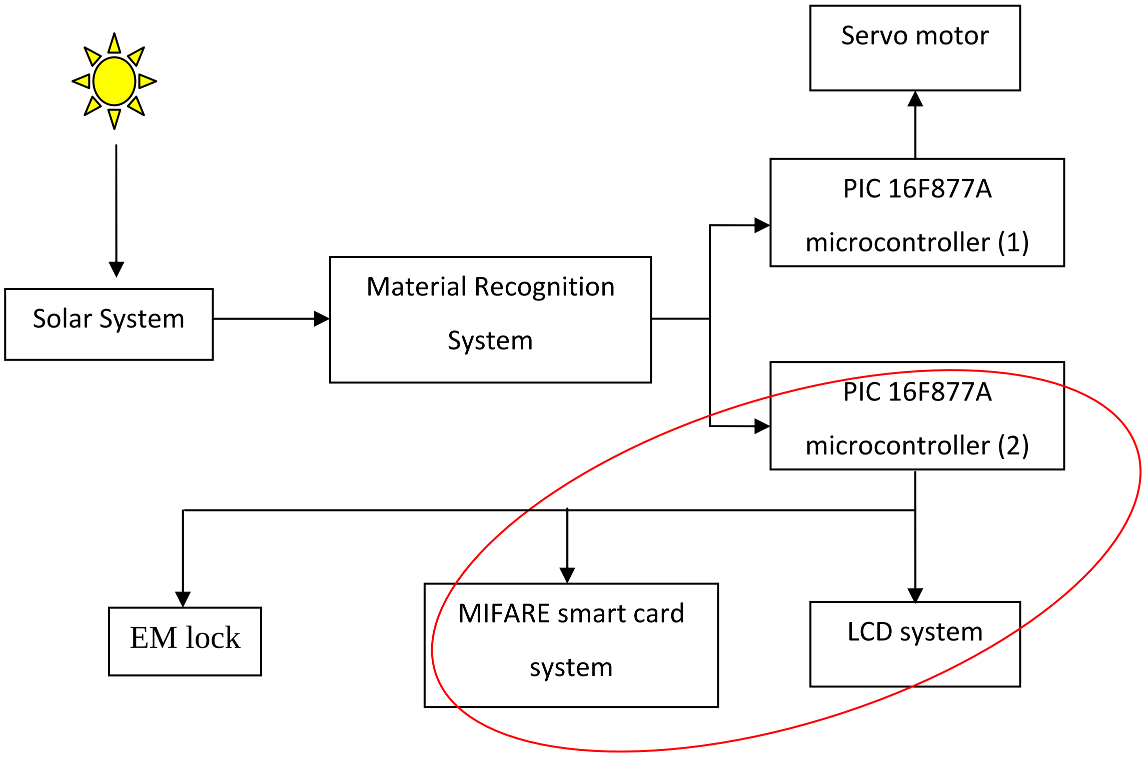


Figure 3.1: Overview of project.

3.2 Project Planning

Basically, the project is divided into a few stages to ensure smooth progress and success. The first stage focuses on literature review to get more information and idea about this project. Smart card technology is studied to understand the features, applications and how to program them. Also, studies on LCD, microcontrollers and the existing Reverse Vending Machine are done.

The second stage is to choose the suitable card type, LCD and microcontroller. In choosing a suitable card type, I must ensure that the card can be read/written and has a RS232 connection to enable communication. LCD with two lines of display and microcontroller PIC16F877A is used.

The third stage involves the software part of this project. Visual Basic is used to enable communication between PC and Mifare Reader/Writer, develop staff platform for redemption of points and creation of database. Microcontroller PIC16F877A is programmed using C programming for two purposes. Microcontroller sends command to PC by using UART signal about the functions to be done. Secondly, a program is also written for LCD to display message and prompt user input.

Fourth stage involves integration of our individual projects namely the electronic reward system and display system, auto-recognition of materials and solar energy system. The operation of these parts as a whole is tested.

Finally, analysis of Recycle and Reward Bin is done in terms of its user friendliness and accuracy. Limitation of the project and future enhancement is also analysed.

3.3 Electronic Reward System

The electronic reward system consists of a PC, card reader/writer module and PIC microcontroller board. There are a few steps in developing the electronic reward system for this project. The steps are:

1) Selection of card technology.
2) Selection of microcontroller.
3) Select software to program microcontroller and MIFARE card system.
4) Develop Visual Basic testing platform.
5) Implement hardware to connect MIFARE module and microcontroller.
6) Test communication between microcontroller, PC and MIFARE card reader/writer module.
7) Develop database using Microsoft Access.
8) Integrate systems to complete project.

3.3.1 Selection of Card Technology

In the project, electronic reward system requires the implementation of card technology where the recycle reward points can be stored into the recycler's card and points can be claimed from the card. Also, recycler is able to check the current points stored in card.

Based on the criteria needed, MIFARE card technology is chosen. There are considerations to be made in the selection of card technology:

1) Type of card used should allow memory operations such as read and write.
2) Allows serial interface using RS232.
3) Data stored are protected from fraud and hacks.
4) Passive cards without use of battery and user friendly.

3.3.2 Selection of Microcontroller

Microcontroller acts as the "brain" that commands the MIFARE module. MIFARE module is always awaiting command whether to read, write or increment points into the card. Hence, the primary concern is to choose the suitable microcontroller that can communicate with the MIFARE module.

Hence, PIC16F877A microcontroller is chosen due to its code efficiency, high performance and low cost applications. Moreover, the programming of microcontroller can be easily done using C programming.

3.3.3 Selection of Software

After determining the card technology and microcontroller to be used, the next step is to choose the software to program them. Microsoft Visual Basic is a good option for programming Mifare module because the following functions should be written:

- The MSComm control allows the transmission and reception of data through a serial port.
- Create a user friendly platform.
- Able to link to Microsoft Access database

Visual Basic enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects. A programmer can put together an application using the components provided with Visual Basic itself. The advantages and disadvantages are as follows [4]:

Advantages:

- The structure of the programming language is very simple.
- VB allows Rapid Application Development (RAD). Most of the works can be created by using components (Drag and Drop).
- VB is supported for other programming languages especially Java language and Flash Player.
- VB saves the file in .exe format

Disadvantages:

- The application of VB only supported by Window Operating Systems.

As for PIC16F877A, C programming can be used to program user prompts in LCD and to send commands to MIFARE card reader/writer module. The main reason is that there are a whole lot of tutorials that will help in learning the language. C is great for putting ideas into practice C is inherently easy to understand. The program should be able to achieve the following function:

- Accept inputs from sensor circuits and display relevant messages on LCD.
- Send command to MIFARE.
- Summation of points.

MPLAB which is a Window-based Integrated Development Environment (IDE) is used to write, debug, and optimize PICmicro applications for firmware product designs. MPLAB includes a text editor, simulator and project manager. The MPLAB IDE also provides feature-rich simulator and emulator environments to debug the logic of executables. MPLAB provides functions that allow users to:

- Create and Edit Source Files
- Group Files into Projects
- Debug Source Code
- Debug Executable Logic Using the Simulator or Emulator

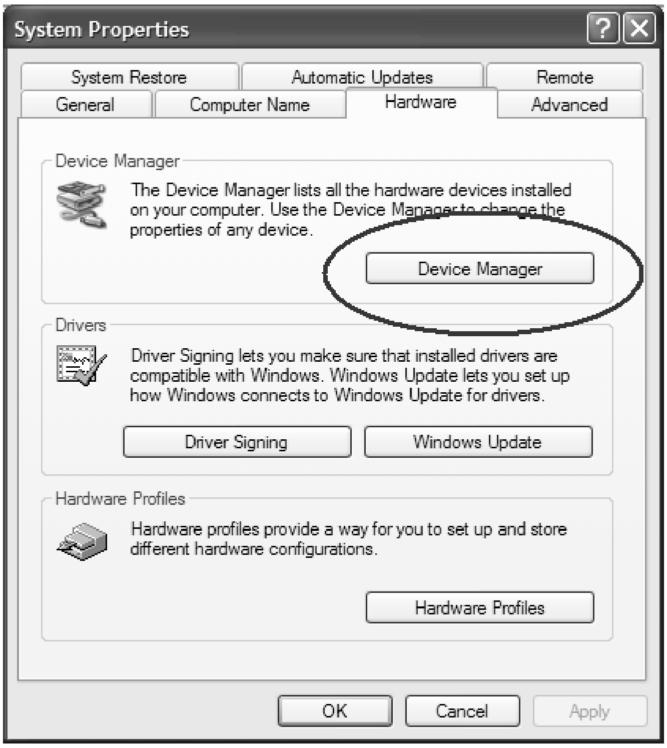
PICkit programmers are used to program and debug microcontrollers, as well as program EEPROM. PICkit 2 has a separate programmer/debugger unit which plugs into the board carrying the chip to be programmed. The PICkit 2 is open to the public, including its hardware schematic, firmware source code (in C language) and application programs (in C# language).

3.3.4 Develop Visual Basic testing platform

MIFARE technology is a rather advanced card technology with high security of data and requires detailed command format. Being a novice, it is important to learn basic functions and commands for MIFARE. Thus, the approach is to build a testing platform for MIFARE module using Visual Basic. The main testing to be done is on the Microsoft Comm. Control function, the connection for COM port used, read data and write data into card.

In order to run the testing program, the COM port is set and the following steps are done:

1. Determine the COM Port Number assigned by the system.

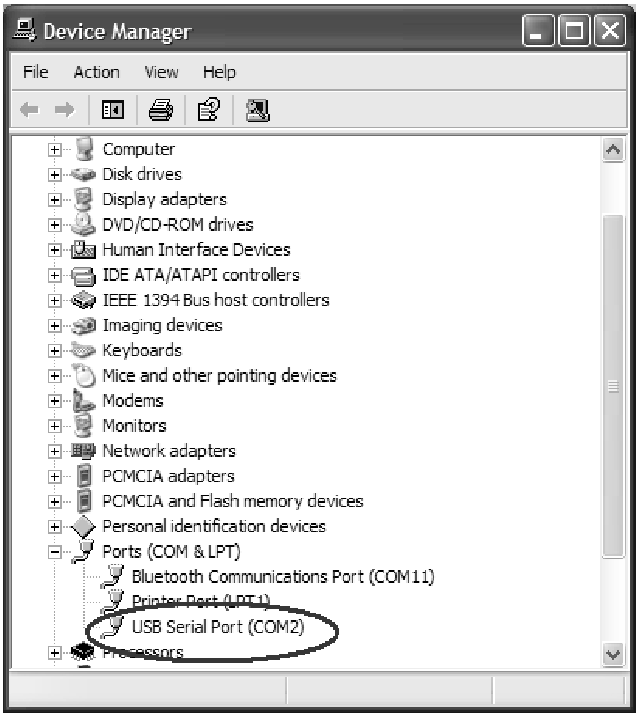


The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

Driver Signing lets you make sure that installed drivers are compatible with Windows. Windows Update lets you set up how Windows connects to Windows Update for drivers.

Hardware profiles provide a way for you to set up and store different hardware configurations.

Figure 3.2 : Device Manager is used to check the USB serial port.

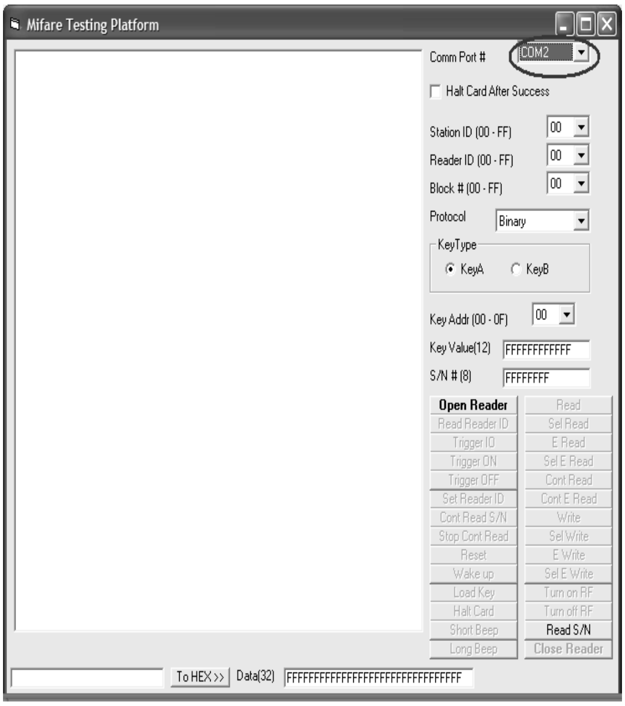


Device Manager

- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Imaging devices
- Keyboards
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- PCMCIA adapters
- PCMCIA and Flash memory devices
- Personal identification devices
- Ports (COM & LPT)
  - Bluetooth Communications Port (COM11)
  - Printer Port (LPT1)
  - USB Serial Port (COM2)
- Processors

Figure 3.3 : Serial port used is COM2.

2. Start the testing program



Mifare Testing Platform

Comm Port # COM2

□ Halt Card After Success

Station ID (00 - FF) 00

Reader ID (00 - FF) 00

Block # (00 - FF) 00

Protocol Binary

KeyType ● KeyA ○ KeyB

Key Addr (00 - 0F) 00

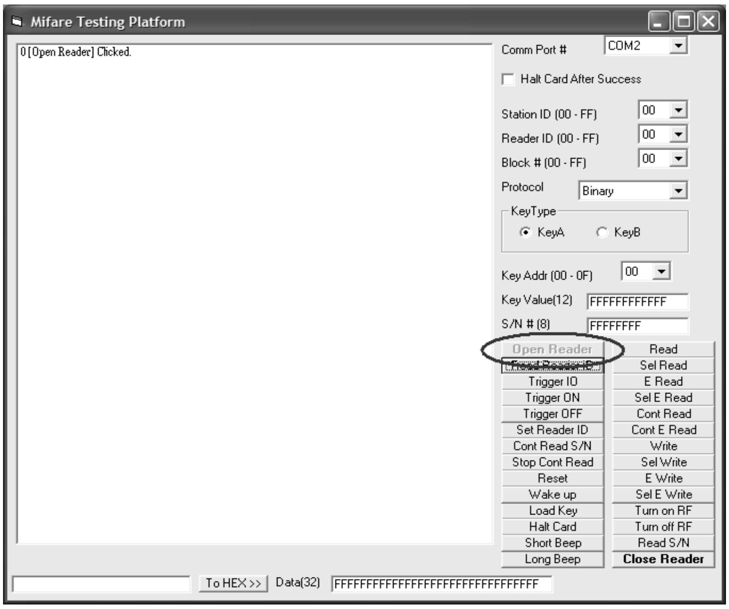
Key Value(12) FFFFFFFFFFFF

S/N # (8) FFFFFFFF

Data(32) FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Figure 3.4: Specify the COM number

3. Open the reader and click on long beep button. The beep will indicate that MIFARE module is successfully connected to PC.



Mifare Testing Platform

```
0 [Open Reader] Clicked.
```

Comm Port # COM2

□ Halt Card After Success

Station ID (00 - FF) 00

Reader ID (00 - FF) 00

Block # (00 - FF) 00

Protocol Binary

KeyType ● KeyA ○ KeyB

Key Addr (00 - 0F) 00

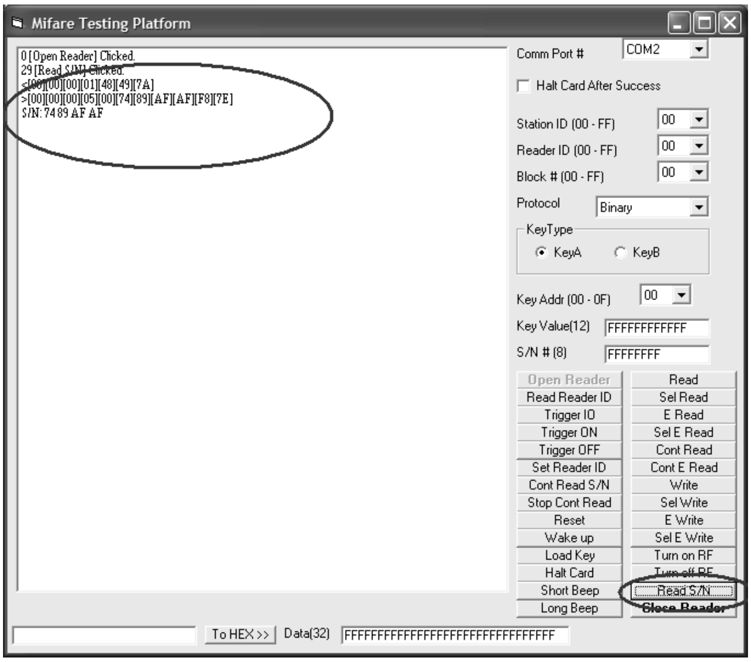
Key Value(12) FFFFFFFFFFFF

S/N # (8) FFFFFFFF

Data(32) FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Figure 3.5: Reader has been successfully connected.

4. With the card on the reader, click on Read S/N button and serial number will be shown in the list box:



Mifare Testing Platform

```
0 [Open Reader] Clicked.
29 [Read S/N] Clicked.
<[00][00][00][01][48][49][7A]
>[00][00][00][05][00][74][89][AF][AF][F8][7E]
S/N: 74 89 AF AF
```

Comm Port # COM2

□ Halt Card After Success

Station ID (00 - FF) 00

Reader ID (00 - FF) 00

Block # (00 - FF) 00

Protocol Binary

KeyType ● KeyA ○ KeyB

Key Addr (00 - 0F) 00

Key Value(12) FFFFFFFFFFFF

S/N # (8) FFFFFFFF

Data(32) FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Figure 3.6: Serial number is returned on the screen.

3.3.5 Implementation of Hardware

Basically, the implementation of hardware would require the connection between MIFARE module, microcontroller and PC as well as the LCD. Considerations made when building the hardware are:

1) Hardware should allow conversion of signal from RS232 serial port to signals suitable for TTL compatible circuit use.
2) The inputs from material sensor circuits to the microcontroller are simulated by replacing a wire connected to 5V for input=1 and connected to GND for input=0.

3.3.6 Test communication

After programming the microcontroller and MIFARE as well as implementing the hardware connections, the next step is to test the communication between them. A useful and easy method is by using Hyperterminal. In order to check if microcontroller is connected properly and PC can successfully receive data from it, a set of commands can be sent through HyperTerminal and view the results obtained on PC. The communication between microcontroller with LCD and MIFARE can be easily verified by observing the program flow and functions performed by the devices.

3.3.7 Develop database using Microsoft Access

Almost every Reverse Vending Machines have database be it database connected to online service center to keep the databases up to date or a separate database accessed remotely by trained personnel for countries that do not have national container database. Thus, Visual Basic enables the implementation of database by using ADO Data Control

3.3.8 Integrate systems to complete project

The final step would be integrating the MIFARE module system and LCD system with the rest of the project done by my teammates. The approach would be obtaining the microcontroller inputs from sensor circuits in the material recognition system done by Chun Hoe and ensure that the user prompts on the LCD follows the right flow. Then, the project is powered by solar energy done by my teammate, Khee Leong. Finally, the circuits and devices are placed onto the casing of the Smart Recycle and Reward Bin.

CHAPTER 4

HARDWARE IMPLEMENTATION

This section is to discuss the hardware and circuit diagrams used to give better view on the overall project. The architecture and characteristics of the components and devices is included in this chapter.

4.1 Electronic Reward System

Electronic Reward system consists of a PIC16F877A microcontroller, PC, MIFARE reader/writer module, RS232 level converter and LCD. Microcontroller PIC16F877A receives input from sensor circuits, performs summation of points and writes to the card. PC is needed to create database as well as to provide a staff platform for the redemption of points. MIFARE reader/writer module reads and writes points into the card, RS232 level converter converts signals from an RS-232 serial port to signals suitable for microcontroller, and LCD displays the messages. Overall block diagram of the project is as shown in Figure 4.1.

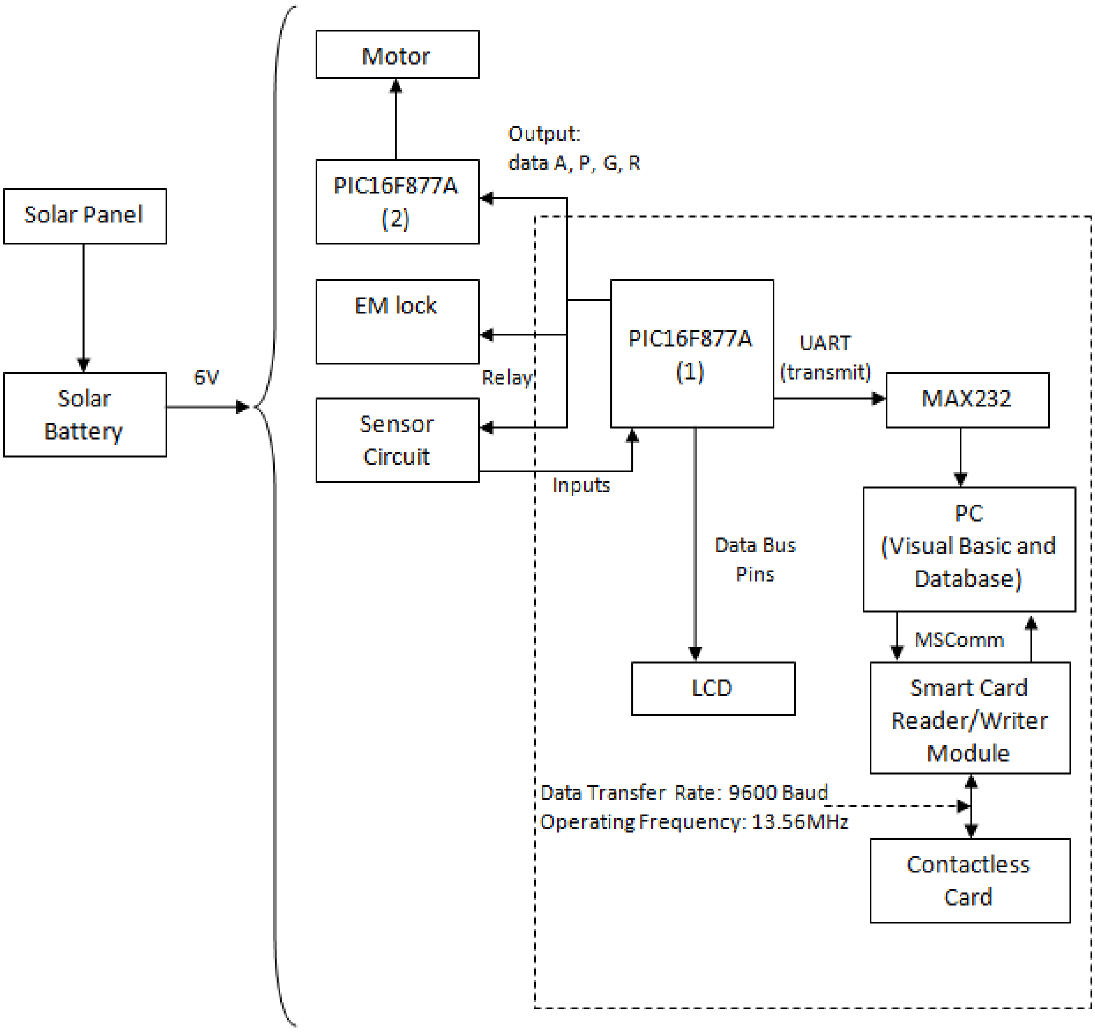


Figure 4.1: Overall block diagram of project.

4.1.1 Microcontroller PIC16F877A

A microcontroller can be programmed to make decision or to perform functions based on predetermined situations and selections. The ability of microcontrollers to perform mathematical and logic functions allows it to mimic sophisticated logic and electronic circuit (Iovine, 2004). Microcontrollers are responsible for the intelligence in most smart devices on the consumer market and most importantly it is inexpensive.

In this project, microcontroller has been used for both display system and reward system. In display system, microcontroller receives inputs from sensor circuits and pushbuttons. Then, it gives output to relay which controls EM lock and sensor circuits and to LCD for display of respective user prompts and messages. Also, it uses UART to send data to MIFARE Visual Basic program. For a clearer view, the pin connections are given in the following figure.

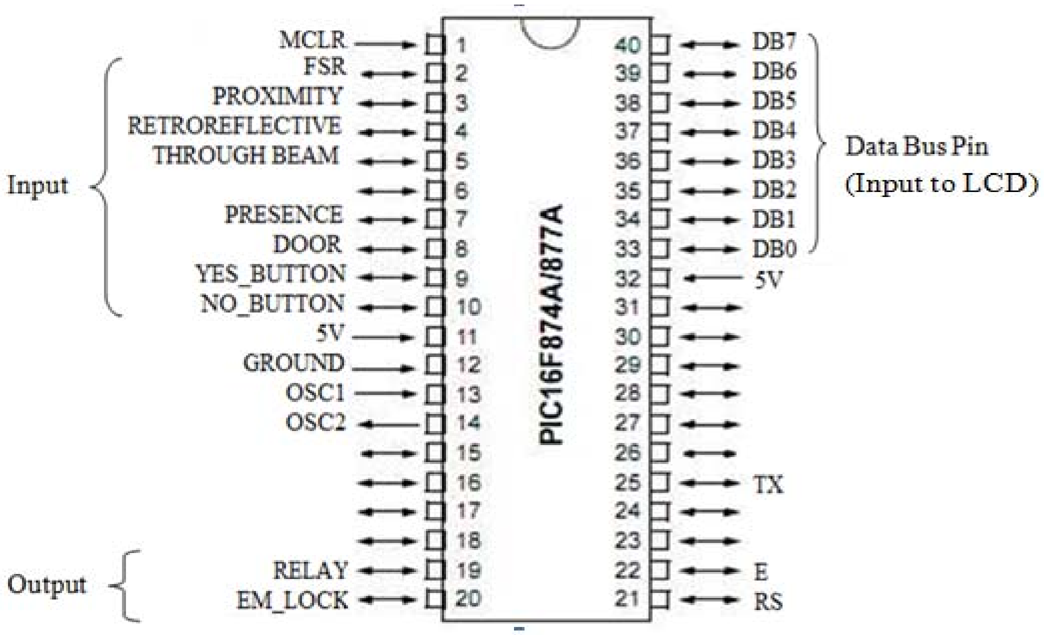


Figure 4.2: Pin connections for microcontroller.

4.1.2 Hitachi Chipset Based 2X16 LCD

This LCD may be interfaced in either 8 pins (8-bit interface) or 4 pins (4-bit interface), depend on the I/O pins available. For 16F877A which has 33 I/O pins, there should be no problem using the 8-bit interface which is much easier for software development. Pin 15 and 16 of LCD are for backlight. The 3rd pin (VEE) needs to be connected to a variable resistor to control the LCD contrast, while the R/W (read or write) pin is connected to the ground which indicates the LCD is in write mode. R/S and E are connected to RD2 and RD3 of PIC respectively. For data communication pins, DB0-DB7 (8-bit) is connected directly to RB0-RB7 of PIC. The pin connections are as shown in Figure 4.3.

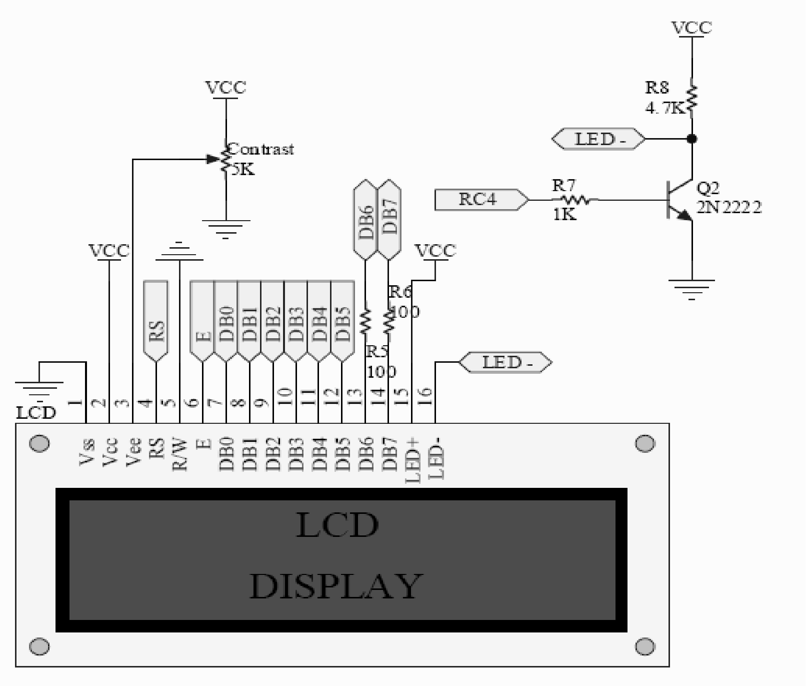


Figure 4.3: Schematic Circuit for LCD Display System

Table 4.2: Pin connections for LCD to microcontroller.

Pin	Name	Pin Function	Connection
1	VSS	Ground	GND
2	VCC	Positive Supply for LCD	5V
3	VEE	Contrast adjust	Connected to a preset to adjust contrast
4	RS	Select Register, select instruction or data register	RB4
5	R/W	Select read or write	GND(always write)
6	E	Start data read or write	RB5
7	DB0	Data bus pin	RD0
8	DB1	Data bus pin	RD1
9	DB2	Data bus pin	RD2
10	DB3	Data bus pin	RD3
11	DB4	Data bus pin	RD4
12	DB5	Data bus pin	RD5
13	DB6	Data bus pin	RD6
14	DB7	Data bus pin	RD7
15	LED+	Backlight positive input	5V
16	LED-	Backlight negative input	Connect to JP8

4.1.3 RS 232 Level Converter

The MAX232 is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The pinouts for MAX 232 is as shown in Figure 4.4. Serial RS-232 communication works with voltages -15V to +15V for high and low. On the other hand, TTL logic operates between 0V and +5V. Thus the RS-232 signal levels are far too high TTL electronics, and the negative RS-232 voltage for high cannot be handled at all by computer logic. To receive serial data from an RS-232 interface the voltage has to be reduced.

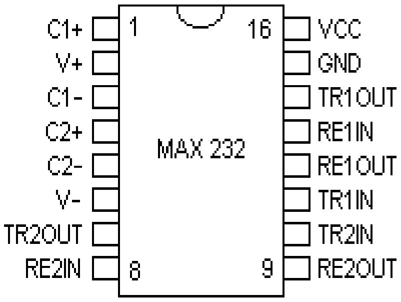


Figure 4.4: Pinouts for the MAX-232.

Also the low and high voltage level has to be inverted. This level converter uses a MAX232 and four capacitors as shown in Figure 4.5.

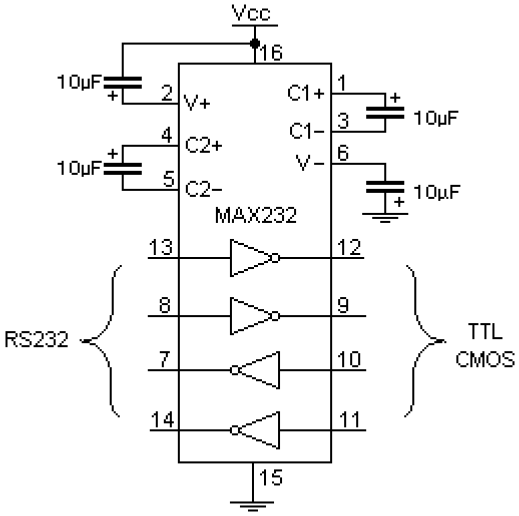


Figure 4.5: Typical MAX-232 Circuit.

The logic waveform as shown in Figure 4.6 illustrates the expected waveform from the UART when using the common 8N1 format. 8N1 signifies 8 Data bits, No Parity and 1 Stop Bit. A transmission starts with a start bit which is (Logic 0). Then each bit is sent down the line, one at a time. The logic waveform is only relevant for the signal immediately at the UART.

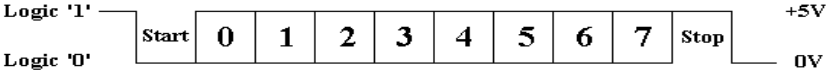


Figure 4.6: TTL/CMOS Serial Logic Waveform.

On the other hand, RS-232 logic levels uses +3 to +25 volts to signify a "Space" (Logic 0) and -3 to -25 volts for a "Mark" (logic 1). Any voltage in between these regions is undefined. Therefore this signal is put through a "RS-232 Level Converter". Signal present on the RS-232 Port of PC is as shown in figure 4.7.

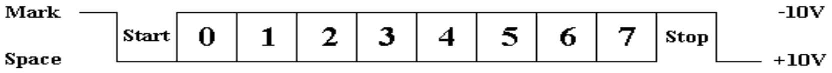


Figure 4.7: RS-232 Logic Waveform

When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15 V, and changes TTL Logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL.

Table 4.1: Voltage levels conversion

RS232 Line Type & Logic Level	**RS232 Voltage**	**TTL Voltage to/from MAX232**
Data Transmission (Rx/Tx) Logic 0	+3V to +15V	0V
Data Transmission (Rx/Tx) Logic 1	-3V to -15V	5V
Control Signals (RTS/CTS/DTR/DSR) Logic 0	-3V to -15V	5V
Control Signals (RTS/CTS/DTR/DSR) Logic 1	+3V to +15V	0V

RS232 DB-9 Connector is an analog 9-pin plug of the D-Subminiature connector family (D-Sub or Sub-D). It is mainly used for serial connections, allowing for the asynchronous transmission of data as provided for by standard RS-232.

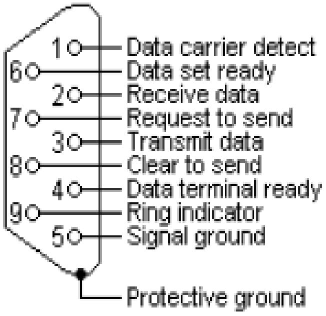


Figure 4.8: DB-9 Connector.

MAX232 chip has been implemented in the microcontroller board. It provides 2-channel RS232C port and requires external 1uF capacitors.

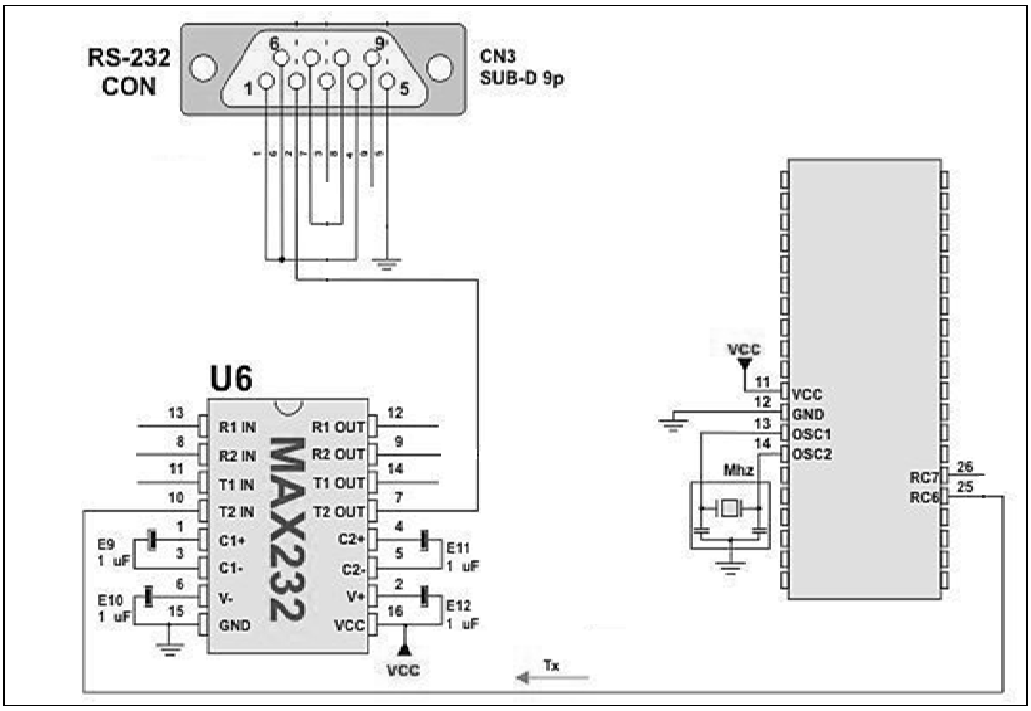


Figure 4.9: RS232 Level Converter

4.1.4 MFR1000SDK MIFARE Hardware

Hardware includes a MIFARE module and mifare transponder which is the MIFARE card. From the MIFARE module, connection has to be made to PC via RS232 serial cable. Connection also has to be made to VCC of 5V and to GND as shown in Figure 4.10. The communication between them uses RFID technology. Reader transmits and receives signal using four external antennas. Radio wave communication is made between tag and reader. Tag sends radio signal usually containing serial number and information to be decoded by reader.

This plug and play reader module is a compact device designed for simple integration into existing applications such as terminal, payment system, access control, handheld devices and etc. It includes an etched antenna on the back of the reader, and the reading range can go up to 80mm depending on the tag. In addition, user may tune the on-board trimmer to get better operating distance.

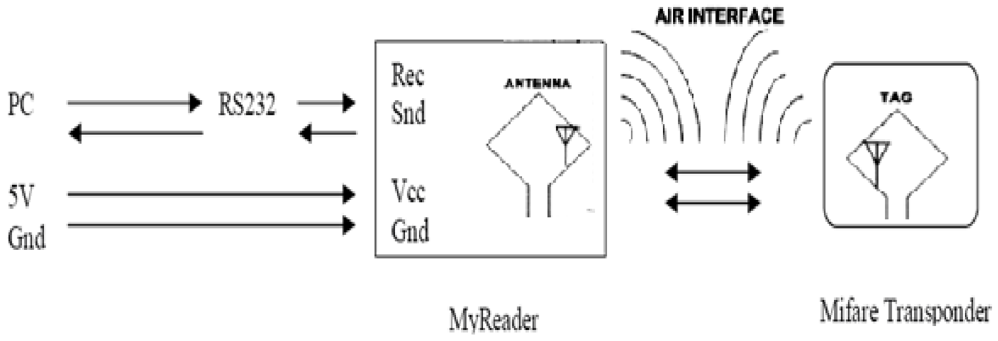


Figure 4.10: Connection block diagram.

It is also important to know the electrical specification listed below:

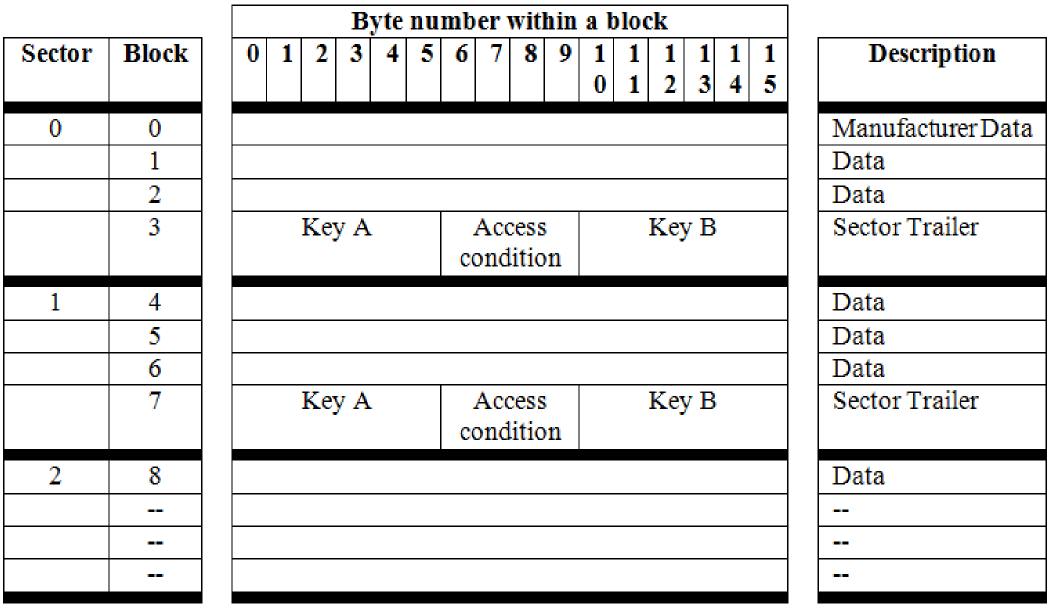
a. Operation frequency	: 13.56MHz
b. Baud rate	: 9600 Baud
c. Interface	: Serial RS232
d. Supply voltage	: Regulated 5V
e. Power consumption	: <160mA
f. Reading range	: Up to 80mm depending on TAG
g. Support ability	: Mifare Standard (1K), standard (4K)

4.1.5 MIFARE Card

The MIFARE name covers proprietary technologies based upon various level of the ISO/IEC 14443 Type A 13.56 MHz contactless smart card standard. The MIFARE Classic card is fundamentally just a memory storage device, where the memory is divided into segments and blocks with simple security mechanisms for access control.

The MIFARE Classic 1K offers 1024 bytes of data storage, split into 16 sectors; each sector is protected by two different keys, called A and B. They can be programmed for operations like reading, writing, and increasing value blocks. The simplicity of the basic cards means that they are inexpensive, which is largely the reason for their success in large-scale deployments, such as Oyster card which is a stored value card used to hold a variety of single tickets, period tickets and travel permits which must be added to the card prior to travel.

The 1K Byte Mifare Standard memory is organized in 16 sectors with 4 blocks each, whereas 4K Byte Mifare Standard memory is organized in 32 sectors with 4 blocks and in 8 sectors with 16 blocks. One block consists of 16 bytes. The last block of every sector is the sector trailer. Each sector trailer holds secret keys A and B, and the access condition for all blocks of that sector.



		Byte number within a block																
Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Description
0	0																	Manufacturer Data
	1																	Data
	2																	Data
	3	Key A						Access condition				Key B						Sector Trailer
1	4																	Data
	5																	Data
	6																	Data
	7	Key A						Access condition				Key B						Sector Trailer
2	8																	Data
	--																	--
	--																	--
	--																	--

Figure 4.11: Memory Organization.

CHAPTER 5

SOFTWARE IMPLEMENTATION

This section is to discuss the software used to achieve the project. Software is used to enable programming of microcontroller and communication for MIFARE to be done. Also, the program flow will be discussed in this chapter.

5.1 Display System

C programming is used to develop program for displaying user prompts and messages on the LCD screen. Source codes for the displaying of message use the same concept. String is sent to display the message. For example, when the machine is not in use, it will always display a welcome note on the LCD. The message will be displayed if the lid is closed and total points is zero. The source code for display of message is as shown below:

```
if(total_points==0 && door==1)                          //door close
        {
        lcd_clr();                                      //clear lcd
        lcd_goto(0);                                    //set the lcd cursor to location 0
        send_string("RECYCLE & REWARD");                //display welcome note
        lcd_goto(20);                                   //set the lcd cursor to location 20
        send_string("    WELCOME");                     //display welcome note
        __delay_ms(1000);                               //delay 1s
        }
```

From the flow chart, the messages to be displayed depend on conditions such as whether the lid is open or close and the presence of item in the machine. If the lid is left open, the message “Please Put Item and Close Door” is displayed. After the lid is closed, the sensor detects the presence of item. If item is present, the message “Please Wait. Processing…” is displayed.

The following display depends on input from sensor circuits. If the sensor circuits failed to identify the type of material, a message “Reject Item” is displayed. If the sensor circuits are able to detect the type of material, the respective message about material type and points rewarded are displayed. Take for example, plastic bottle is detected and the codes are used as shown. The same concept applies to glass bottles and aluminium cans. The source code is shown as follows:

```
if(fsr == 0 && proximity == 0 && retroreflective == 0 && through_beam == 1)
{
        rec_pts = 1;                                        //condition to enter loop
        uart_send(uart_data1[0]);                           //signal “p” sent to VB
        total_points+=1;                                    //total points increased by 1
        lcd_clr();                                          //clear lcd
        lcd_goto(0);                                        //set lcd cursor to location 0
        send_string("PLASTIC: 1 PTS");                      //display material type
        lcd_goto(20);                                       //cursor moves to next line
        send_string("TOTAL PTS: ");                         //Display string
        dis_num(total_points);                              //display total points
        send_string(" PTS: ");                              //display string
        __delay_ms(2500);                                   //delay 2.5s
        plastic = 1;                                        //send output  to PIC(2)
}
```

The prompts and messages are displayed sequentially based on the inputs and events that happen. The program flow is summarized as shown in Figure 5.1.

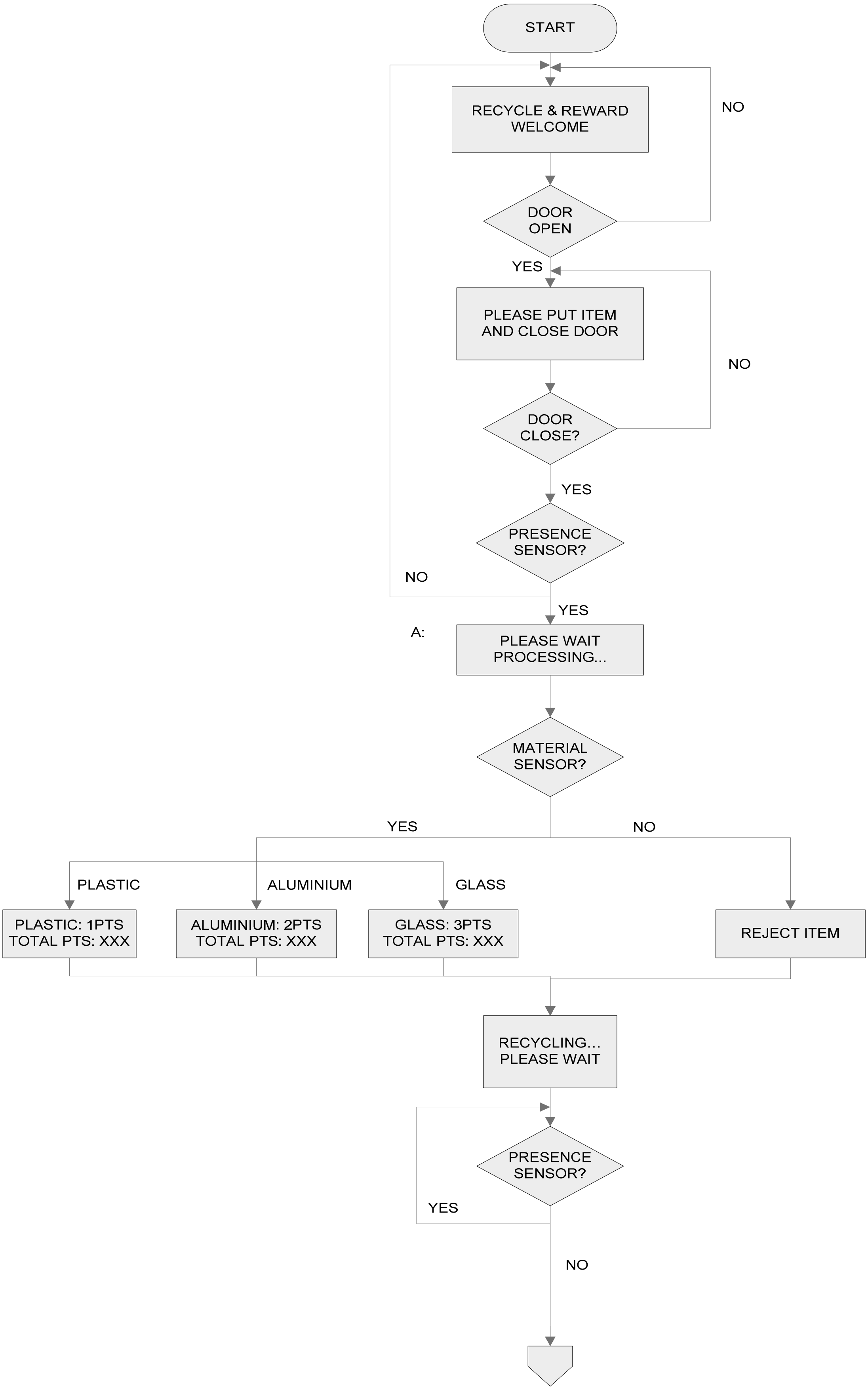


Figure 5.1: Program flow of display on LCD

After the container has been accepted by the machine, total points will be displayed and user is prompted to press the pushbutton “YES” to end recycling process. The source code used is as shown:

```
if (door==1)
{
        lcd_clr();                                  //clear lcd
        lcd_goto(0);                                //set the lcd cursor to location 0
        send_string("TOTAL PTS: ");                 // display string
        dis_num(total_points);                      //display the value of total points
        lcd_goto(20);                               //cursor moves to next line
        send_string("TO END,PRESS YES");            //display string
        while(yes==0 && door==1);                   // loop while “yes” not pressed and
                                                    //lid close
}
```

If user did not press “YES” button to end process and opens the lid to continue recycling, the process is repeated from the display of welcome message. If button is pressed to end recycling, total points are displayed and a prompt on whether to store points in card is posed to user. The source code is as shown below:

```
if(yes_pts==1)
{
        em_lock = 1;                                // lid is locked
        lcd_clr();                                  //clear lcd
        lcd_goto(0);                                //set the lcd cursor to location 0
        send_string("TOTAL PTS: ");                 // display string
        dis_num(total_points);                      //display the value of total points
        lcd_goto(20);                               //cursor moves to next line
        send_string("STORE POINT?");                //display string
        __delay_ms(1000);                           //delay 1s
        while(yes==1);                              //loop while “yes” not pressed
}
```

If user presses “NO” button, a message “Thank You” is displayed. Else, a message “Place card and Press Yes” is shown. After the user places card and press “YES” button, the message “Points Stored” is displayed.

```
else if(yes_pts==1 && no==1)                        // YES pressed to end, NO pressed
                                                    //to refuse to store points
{
        lcd_clr();                                  //clear lcd
        lcd_goto(0);                                //set the lcd cursor to location 0
        send_string(" THANK YOU :)");               // display string
        __delay_ms(2000);                           //delay 2s
```

```
          yes_pts=0;                                          //To clear yes_pts
          total_points=0;                                     //To clear total points
          goto start;
}
```

If user presses “YES” button to store points, data “s” is sent to Visual Basic application. Then, “PROCESSING…” is displayed on the LCD.

```
else if(yes_pts==3)
{
          uart_send(uart_data4[0]);
          lcd_clr();                                          //clear lcd
          lcd_goto(20);                                       //set the lcd cursor to location 0
          send_string("PROCESSING");

          for(int i = 0; i <=1; i++)                          // display sequential dots
          {
                    lcd_goto(30);
                    send_string(".");                         // display dot
                    __delay_ms(500);                          // delay 0.5s
                    send_string(".");                         // display second dot after delay
                    __delay_ms(500);                          // delay 0.5s
                    send_string(".");                         // display third dot after delay
                    __delay_ms(500);                          // delay 0.5s
                    lcd_goto(30);                             // clear
                    send_string("   ");
          }
```

After points are stored successfully, the LCD informs the user by displaying message “POINTS STORED” and “THANK YOU” as shown in the code below:

```
lcd_clr();                                                    //clear lcd
lcd_goto(0);                                                  //set the lcd cursor to location 0
send_string("POINTS STORED");                                 // display string
lcd_goto(20);                                                 //cursor moves to next line
send_string(" THANK YOU :)");                                 // display string
__delay_ms(2000);                                             // delay 2s
yes_pts=0;                                                    //To clear yes_pts
total_points=0;                                               //To clear total points
goto start;
```

The program flow for the described code is summarized as shown in Figure 5.2.

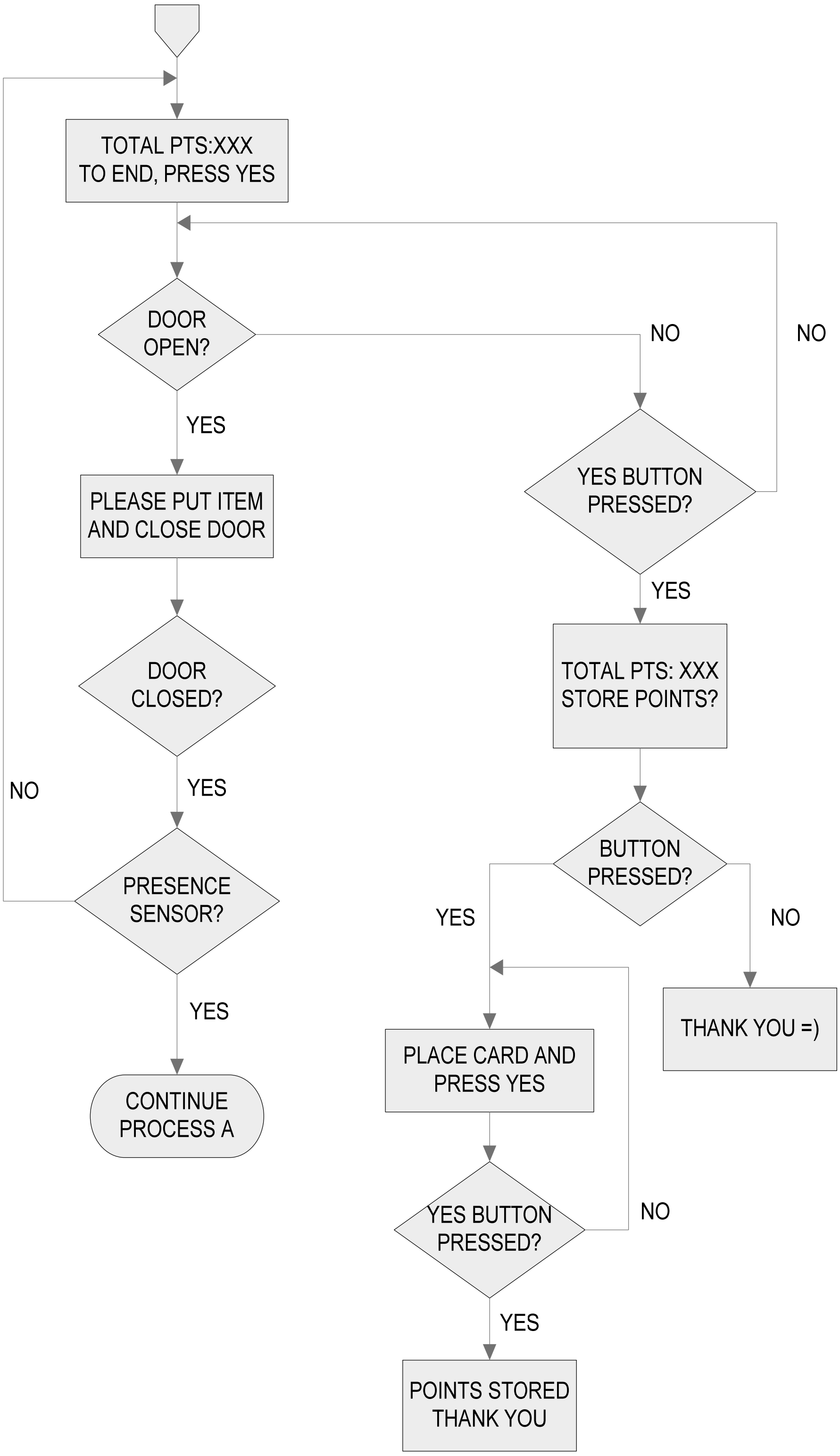


Figure 5.2: Continuation for program flow of display on LCD.

The program written also includes UART function and is summarized in Figure 5.3. Whenever the sensor circuit recognizes a material type, a data is sent to the PC. Take for instance, aluminium can is detected. The program will send the data “a” using UART function as shown below:

```
else if(fsr == 0 && proximity == 1 && retroreflective == 1 && through_beam == 0)
{           //partial code
            rec_pts = 1;
            uart_send(uart_data2[0]);                    // UART function to send “a”
            total_points+=2;
}
```

As for the UART function, the source code is as shown below:

```
void uart_send(unsigned char data)
{           while(TXIF==0);                                      //only send the new data after
            TXREG=data;                                          // previous data sent
}
```

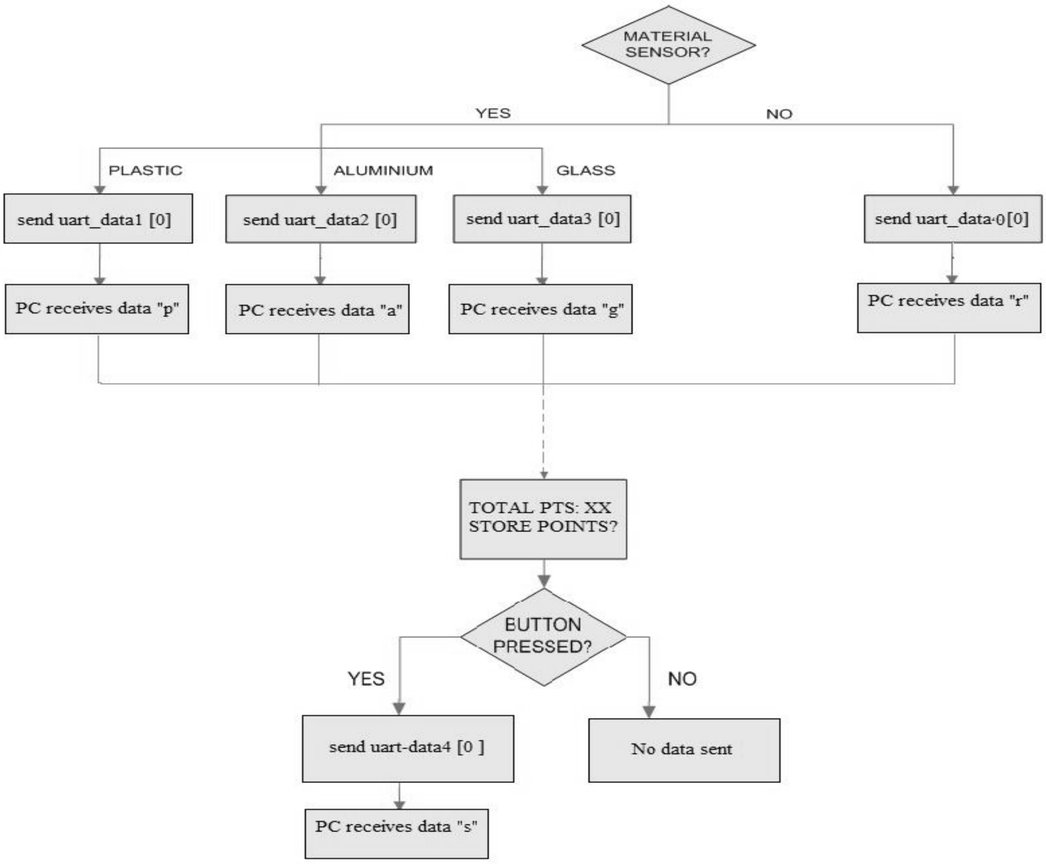


Figure 5.3: Program flow of UART function.

5.2 Reward System Using MIFARE Application

Throughout the development of the program, the instruction list is referred to send the correct hex value as data to represent the command desired. Take for instance, 50H represents the command to read reader ID number whereas 5eH represents the command to generate a short beep sound. The complete list of instructions is attached as appendix.

This data is then used in the binary protocol communication package. The binary protocol is defined as below:

Sending:

Host	sof	mid	sid	nodb	data	bcc	eot

Receiving:

Reader	sof	mid	sid	nodb	data	bcc	ack

Definitions:

- rts – request to send (70h). The rts should be sent more than once if no response was received from reader.
- ack – acknowledgement (7eh).
- sof – start of frame (00h).
- mid – This is the Master ID or station ID.
- sid – This is the reader ID number.
- nodb – The number of byte contains in the data block.
- data – This data block depends on the type of command.
- bcc – Block check character;
- eot – End of text (7a or 7b).

The byte setting includes transmission rate of 9600 Baud, Start Bit which is 1 bit, Stop Bit of 1 bit and Data Bits of 8 bits.

Hence, the communication protocol needs to be followed for every command sent. A few important examples are described as follows:

Pcd_areadonesnr (Type A) :

This instruction is used to read the type A card's serial number, sent format will be 00/mid/sid/01/48/bcc/7a where 7a means end of text.

Host

sof	mid	sid	nodb	data	bcc	eot
00	mid	sid	nodb	48h	bcc	eot

The returned format is 00/mid/sid/09/00/88/04/a7/50/00/00/00/00/bcc/7e which acknowledges the command sent. 00H means the instruction was executed successfully, or else an error code will be returned. If error code is returned, the nodb will be 01, since no data will be returned.

Reader

sof	mid	sid	nodb	data		bcc	ack
00	mid	sid	nodb	message	srn no	bcc	7eh

Pcd_write:

This instruction is used to write a block of data to a card. The sent format may be 00/mid/sid/19/41/05/00/01/02/03/04/05/06/55/55/55/55/aa/aa/aa/aa/55/55/55/55/05/fa/05/fa/bcc/7a (write to block 05, using key A, load key 01/02/03/04/05/06, write a value block with structure 4-byte value/4-byte value_b/4-byte value/block_no/block_no_b/block_no/block_no_b, no halt after success.)

Host

sof	mid	sid	nodb	data					bcc	eot
00	mid	sid	nodb	41h	Blk no	Keytype	Key	Data	bcc	eot

For the returned format may be 00/mid/sid/01/01/bcc/7e (error message returned, refer to the error list, the error is mi_notagerr which means no card detected.)

Reader

sof	mid	sid	nodb	data			bcc	ack
00	mid	sid	nodb	message	srn no	size	bcc	7eh

Pcd_beep

This instruction is used to generate a beep sound. The sent format would be 00/mid/sid/01/5d/bcc/7a.

Host

sof	mid	sid	nodb	data	bcc	eot
00	mid	sid	nodb	5dh	bcc	eot

There is no returned format required.

Pcd_read

This instruction is to read a block of data. In the sent format, the user has to load in the authentication key.

For example, 00/mid/sid/09/40/08/01/b0/b1/b2/b3/b4/b5/bcc/7b (reading block 8, using key B, load key b0/b1/b2/b3/b4/b5, halt the card after success.)

Host

sof	mid	sid	nodb	data				bcc	eot
00	mid	sid	nodb	40h	Blk no	keytype	key	bcc	eot

The definition for the sent format is:

- Blk no : Block/page number.
- Keytype : Select Key A or Key B for authentication. Enter 00 for Key A or 01 for Key B.
- Key addr : Eeprom key address (00h-0fh).
- Key : Six bytes key.
- Srn_no : Card serial number.
- Eot : 7a (no halt after success.) or 7b (halt after success.).

The returned format is:

Reader

sof	mid	sid	nodb	data				bcc	ack
00	mid	sid	nodb	message	srn no	size	data	bcc	7eh

The definitions for the returned format are:

- Message: 00H means the instruction was executed successfully, or else an error code will be returned. If error code is returned, the nodb will be 01, since no data will be returned.
- Srn no: This is the unique serial number of the card.
- Size: This one-byte block will tell the user the card type in the field.
- Data: This data block contains 16-byte data that returned from the card.

Pcd_value

This instruction is to perform arithmetic to the value of a card. In the sent format, the user has to load in the authentication key.

For example, 00/mid/sid/0f/42/01/00/a0/a1/a2/a3/a4/a5/02/c0/01/01/01/01/bcc/7a (decrease the value in block 01 by 01/01/01/01, using key A, load key a0/a1/a2/a3/a4/a5, transfer the result to block 02, no halt after success.)

Host	sof	mid	sid	nodb	data							bcc	eot
	00	mid	sid	nodb	42h	Blk no	Key type	Key	Transfer blk	Id mode	Value	bcc	eot

The definitions are:

- Blk no : Block/page number.
- Keytype : Select Key A or Key B for authentication. Enter 00 for Key A or 01 for Key B.
- Key addr : Eeprom key address (00h-0fh).
- Key : Six bytes key.
- Srn no: This is the unique serial number of the card.
- Transfer blk : The block address the user wish the result to be transferred. The transfer block should be in the same sector of block number.
- Idmode : Select decrement, increment or restore function to be performed.

no	**Function**	**idmode**
1	Decrement	C0H
2	Increment	C1H
3	Restore	C2H

- Value : This is the 4-byte value the user wish to decreased, or increased.

(When executing Restore function, please enter 00/00/00/00 as dummy.)

- Eot : 7a (no halt after success.) or 7b (halt after success.).

For ease of programming, Visual Basic program can be used to declare the command with a global constant hex value in the .bas file. For example, the command for Pcd_Value is 42H can be declared as follows:

```
Global Const  CMD_VALUE = &H42&
```

Thus, the author does not need to refer to the instruction list (Appendix A) when writing a command. The global constants declared are attached in (Appendix B). Also, each part of the format can be declared as long integers variable to assign values to them instead of writing the required information for the sent command byte by byte as shown below:

```
Case 26, 27:                                  ' long beep, short beep
    ReDim b(1 To 7)
    b(1) = SOF
    b(2) = GetStationID()                     ' GetStationID = Val(cbStationID.ListIndex)
    b(3) = GetReaderID()                      ' GetReaderID = Val(cbReaderID.ListIndex)
    b(4) = 1
    b(5) = IIf(Index = 26, &H5D, &H5E)        ' H5D long beep, &H5E short beep
    b(6) = BCC(b(), 5)
    b(7) = IIf(chkHaltCard.Value = 0, EOTA, EOTB)     ' 7a or 7b
    Call ProcMsg(b, 7, "<")
    Call SendPort(b)
```

Pcd_beep (Generate a beep sound.) (Send format.)

Host	sof b(1)	mid b(2)	sid b(3)	nodb b(4)	data b(5)	bcc b(6)	eot b(7)
	00	mid	sid	nodb	5dh	bcc	eot

When the format is declared as listed in Appendix B, short programs can be done more easily for the four important functions in the Visual Basic application developed as described in the following:

When button for checking previous points is activated, the point in the card is displayed on the screen. If the previous function is to increment the value, then the database is updated.

The program code is as shown below:

```
Case 2: 'check previous point, just do a CMD_READ
        m.nCmd = CMD_READ                   'm.nCmd = &H40 (Pcd_read command)
        m.nBlockNo = GetBlockNo()           ' GetBlockNo = Val(cbBlockNo.ListIndex)
        m.nKeyType = GetKeyType()           ' GetKeyType = CByte(cbKeyType.ListIndex)
        Call SetKeyValue(m)                 'default key value FFFFFFFFFFFF
        m.nHaltAfterCommand = IIf(chkHaltCard.Value = 0, 0, 1)
         res = SendMsg(m)
        showMsg "Value=" & m.nValue         'Display the values
        lbllast_pts.Caption = m.nValue       'shows the value of last total points rewarded
        MF_Button(3).Value = True            'Read S/N
        Adodc2.Refresh

        If Val(lblcount_check.Caption) > 0 Then
            Do Until Adodc2.Recordset.EOF          ' Check until end of database record
If Adodc2.Recordset.Fields("Card S/N") = lblserial_no.Caption Then  ' If card S/N match
                Adodc2.Recordset.Fields("Last Points") = lbllast_pts.Caption    ' gets the last point
                Adodc2.Recordset.Update                                       ' updates the database
                  Exit Do
              End If

            Adodc2.Recordset.MoveNext                              ' move to next row in database

             Loop
         End If
```

The reward points are added for the containers recycled. The mode of operation is increment. The database also has to be updated as shown in the code below:

```
Case 4: ' Increment

 m.nCmd = CMD_VALUE                     'm.nCmd = 42H refers to Pcd_value command
 m.nValueOpMode = INCREMENT                  ' mode of operation is increment
 m.nKeyAddr = GetKeyAddr()                  'GetKeyAddr = Val(cbKeyAddr.ListIndex)
m.nBlockNo = GetBlockNo()              ' GetBlockNo = Val(cbBlockNo.ListIndex)
m.nKeyType = GetKeyType()              ' GetKeyType = CByte(cbKeyType.ListIndex)
m.nValue = Val(lbltotal_pts.Caption)    'increase by value of total points
m.nTransferBlockNo = m.nBlockNo         'transfer to block chosen
m.nHaltAfterCommand = IIf(chkHaltCard.Value = 0, 0, 1) 'chose halt or don't halt
Call SetKeyValue(m)                     'default key value FFFFFFFFFFFF
res = SendMsg(m)

Adodc2.Refresh                          'updates database
MF_Button(3).Value = True 'Check S/N
With Adodc2.Recordset
Do Until .EOF            ' do until end of field
    If .Fields("Card S/N") = lblserial_no.Caption Then
      .Fields("Date      Modified") = lblDate.Caption & "   " & lblTime.Caption   ' database
       .Fields("Points Modified") = lbltotal_pts.Caption      ' total points is stored in database
     .Fields("Mode") = "Increment"                            ' type of event is increment
     .Update                                                  ' Adodc2 is updated
       Exit Do
    End If

           .MoveNext

           Loop

           End With
```

When customer wants to redeem points, the staff clicks on the button to write intended value into the card. The function will store new value into card and updates the data as shown in the code below:

```
Case 5: ' Write Value
    lblcount_check.Caption = Val(lblcount_check.Caption) + 1       ' to set the check last
                                                                   'point button is from
                                                                   'write mode
    MF_Button(2).Value = True                                 'Check previous value
    m.nCmd = CMD_WRITE_VAL                                    ' m.nCmd is 41F (Pcd_write)
    m.nKeyAddr = GetKeyAddr()                   'GetKeyAddr = Val(cbKeyAddr.ListIndex)
    m.nBlockNo = GetBlockNo()                   ' GetBlockNo = Val(cbBlockNo.ListIndex)
    m.nKeyType = GetKeyType()                   'GetKeyType = CByte(cbKeyType.ListIndex)
    If Val(lbltotal_pts.Caption) > 0 Then               ' if there is total points, means total
                                                        'points is to be added to last point
```

```
     m.nValue = Val(lbltotal_pts.Caption)              ' nValue is total point value.
End If
If Val(lbltotal_pts.Caption) = 0 Then                 ' if no total points, means the
                                                        ' purpose is to write new data into
                                                       ' card for redemption of points
     m.nValue = Val(txtwrite_pts.Text)                 ' nValue is now the points entered
                                                       ' by the staff
End If

m.nHaltAfterCommand = IIf(chkHaltCard.Value = 0, 0, 1) 'either half or don't
Call SetKeyValue(m)                                    'default key value FFFFFFFFFFFF
res = SendMsg(m)
lblcurrent_pts.Caption = m.nValue                      ' Current points displayed is the
                                                        'nValue.
MF_Button(3).Value = True                              'Check S/N
Adodc2.Refresh
With Adodc2.Recordset
Do Until .EOF

If lblserial_no.Caption = "" Then                    ' If no card, exit loop
   Exit Do
End If

If .Fields("Card S/N") = lblserial_no.Caption Then           ' If card S/N matches
       .Fields("Current Points") = lblcurrent_pts.Caption     ' update database
.Fields("Date Modified") = lblDate.Caption & "    " & lblTime.Caption
       .Fields("Points Modified") = lblcurrent_pts.Caption
       .Fields("Mode") = "Write"
       .Update
       Exit Do

End If

.MoveNext

If .EOF Then                                          ' if card is from new member
       .AddNew                                        ' add to new row in database
       .Fields("Date Created") = lblDate.Caption & "    " & lblTime.Caption    'update
       .Fields("Card S/N") = lblserial_no.Caption
       .Fields("Current Points") = lblcurrent_pts.Caption
       .Fields("Points Modified") = lblcurrent_pts.Caption
       .Fields("Mode") = "New"
       .Update
       Exit Do
End If

Loop

End With

txtwrite_pts.Text = ""     ' after value written to card, the value on screen becomes 0.
```

When button to store point is being pressed, it will trigger the event of incrementing the reward points and then write the total points into card. Also the database is being updated as shown in the codes below:

```
Case 7: ' Store point

    If Val(lblItem_no.Caption) > 0 Then        ' if there is recycled item

      lblcount_check.Caption = Val(lblcount_check.Caption) + 1      ' to set the check last
point button is from store point mode.

      MF_Button(2).Value = True              'Check Last Value in card
      MF_Button(3).Value = True              'Check S/N
      Adodc2.Refresh

      With Adodc2.Recordset
      Do Until .EOF
      If .Fields("Card S/N") = lblserial_no.Caption Then
        If Val(.Fields("Last Points")) > 0 Then
          MF_Button(4).Value = True          'Increment
          Exit Do
        End If

      End If

      .MoveNext

      If .EOF Then
        MF_Button(5).Value = True            'Write
      End If

      Loop
      End With

    MF_Button(8).Value = True                'Check Current Value in card


    Adodc1.Recordset.AddNew                   ' updates info on database
    With Adodc1.Recordset
      .Fields("Date Modified") = lblDate.Caption & "    " & lblTime.Caption
      .Fields("Card S/N") = lblserial_no.Caption
      .Fields("Reject Item Quantity") = lblcount_r.Caption
      .Fields("Plastic Quantity") = lblcount_p.Caption
      .Fields("Aluminium Quantity") = lblcount_a.Caption
      .Fields("Glass Quantity") = lblcount_g.Caption
      .Fields("Plastic Total Pts") = lblpts_p.Caption
      .Fields("Aluminium Total Pts") = lblpts_a.Caption
      .Fields("Glass Total Pts") = lblpts_g.Caption
      .Fields("Total Pts") = lbltotal_pts.Caption
```

```
       .Update
     End With

     lblItem_no.Caption = "0"                 ' set back to default
     lblmaterial.Caption = "-"
     lblpoints.Caption = "0"
     lblcount_r.Caption = "0"
     lblcount_p.Caption = "0"
     lblcount_a.Caption = "0"
     lblcount_g.Caption = "0"
     lblpts_p.Caption = "0"
     lblpts_a.Caption = "0"
     lblpts_g.Caption = "0"
     lbltotal_pts.Caption = "0"
     lblcount_check.Caption = "0"

     End If
```

MIFARE application developed is able to display information such as material type, points rewarded, number of items recycled, summation of total points. The program code is as shown below where aluminium can is inserted. The same concept applies to glass and plastic.

```
Private Sub CmdAluminium_Click()  ' if aluminium can is inserted, "aluminium" button value
                                  ' is true
  lblItem_no.Caption = Val(lblItem_no.Caption) + 1    ' Number of item inserted is
                                                      ' increased by 1
  lblmaterial.Caption = "Aluminium"                   ' Displays material type as aluminium
  lblpoints.Caption = "2"                             ' Points rewarded is displayed as 2
  lblcount_a.Caption = Val(lblcount_a.Caption) + 1 ' count number of aluminium can
                                                   ' recycled
  lblpts_a.Caption = Val(lblpts_a.Caption) + 2       ' total points rewarded for aluminium
                                                     ' cans
  lbltotal_pts.Caption = Val(lblpts_p.Caption) + Val(lblpts_a.Caption) + Val(lblpts_g.Caption)
'sum
```

MIFARE application is developed using Microsoft Visual Basic which offers several project templates that are designed to support development of different kinds of applications and components. The MSComm control provides serial communications for user application by allowing the transmission and reception of data through a serial port. MSComm is configured by writing the program as shown below:

```
 ' MSComm Activex Control supports Serial port programming
   MSComm2.RThreshold = 1                    ' enables firing of MSComm's oncomm event.
   MSComm2.InputLen = 10                       ' When Inputting Data, Input 10 Byte at a
                                               'time
   MSComm2.Settings = "9600,N,8,1"           ' 9600 Baud, No Parity, 8 Data Bits, 1 Stop Bit
MSComm2.DTREnable = False              ' Disable Data Terminal Ready to stop data transfer
   MSComm2.CommPort = 4                      ' Open COM port 4
End Sub
```

However, the communication protocol for Open Reader, Close Reader and DoMF is unknown because the setting has been readily given in a secured .dll file.

```
Declare Function OpenReader Lib "mfr.dll" (ByVal nPortNo As Long, ByVal nStationID As
Byte, ByVal nReaderID As Byte) As Long

Declare Function CloseReader Lib "mfr.dll" (ByVal handle As Long) As Long

Declare Function DoMF Lib "mfr.dll" (ByVal handle As Long, ByRef mfmsg As MF_MSG) As
Long
```

For example, **OpenReader()** opens the communication port numbered nPortNo for communication between the reader and user’s software application. It returns value 0 if there is error and returns communication non zero *handle* if successful to be used in future communication with reader via ***DoMF()***.

The format is as follows:

long OpenReader(long nPortNo, BYTE nStationID, BYTE ReaderID)

where the three parameters must be set properly to communicate with the reader; the comm. port number used by PC to connect with mifare module. the Station ID which is for PC and the Reader ID for the MIFARE module.

The program written is just to send the parameter to the function. The program code written is shown below:

```
Case 0: ' open reader
      hReader = OpenReader(cbPortNo.ListIndex + 1, GetStationID(), GetReaderID())
' gives parameter for OpenReader(long nPortNo, BYTE nStationID, BYTE nReaderID)
```

DoMF() allows the PC software communicates with the reader via the opened communication port handle, *handle*. The STRUCTURE MF_MSG, m, holds both the input and output parameters for ***DoMF()***.

The format is as follows:

long DoMF(long handle, MF_MSG *m)

CHAPTER 6

RESULTS AND DISCUSSIONS

This section is to discuss the testing of project and discuss its performance. Platforms and programs developed are tested to ensure that the functions desired are achieved. The main reason of testing is to ensure that data was sent correctly through the programs written so that MIFARE module and microcontroller can communicate well.

6.1 Display System

PIC16F877A is also programmed to display user prompts and messages on LCD. It is very important to ensure that the sequences of the display follows the events and user input. The testing of LCD system is done by representing sensor inputs with wires and outputs for EM lock and relay with LEDs.

During idle mode before the recycling takes place, there will be a Welcome message displayed on LCD. EM lock is deactivated to enable user to open the lid while Relay 1 is activated to operate sensor circuit as shown in Figure 6.1.

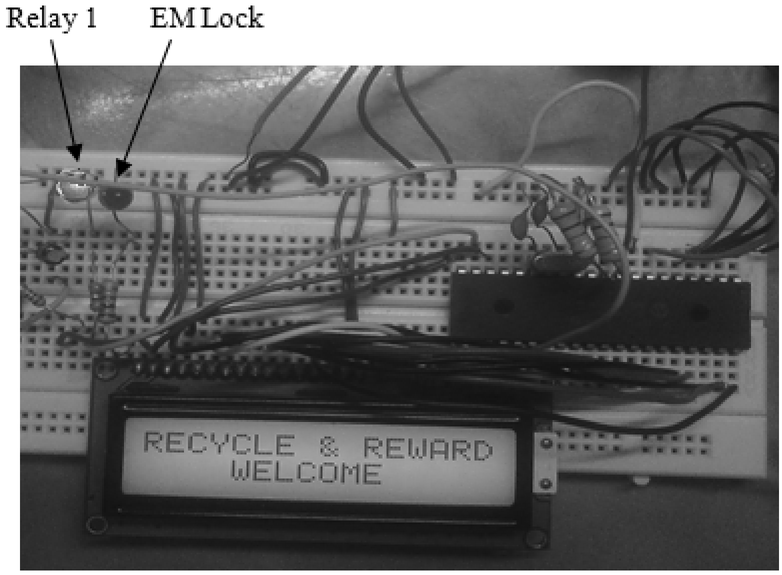


Figure 6.1: LCD displays “RECYCLE & REWARD” and “WELCOME”.

When recycler opens the lid of Smart Recycle and Reward Bin, the LCD displays an instruction for recycler on the following step. As shown in Figure 6.2, lid open is represented by wire connected to ground, and LED turned off indicates that EM lock is deactivated. Thus, the lid can be opened.

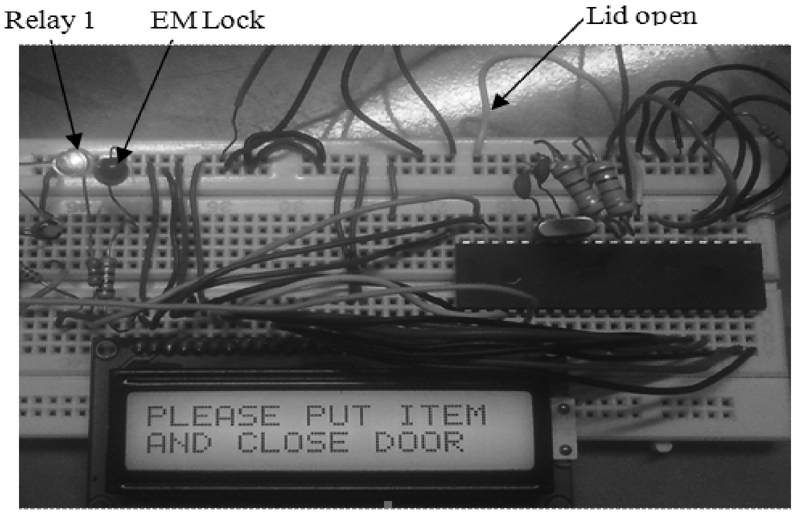


Figure 6.2: LCD displays “PLEASE PUT ITEM AND CLOSE DOOR”.

During the recycling process, the recycler inserts a beverage container and closes the lid. Input from presence sensor is HIGH, LCD will display message to ask recycler to wait for the processing as shown in Figure 6.3. LED for Relay 1 and EM lock is turned on. Lid cannot be opened for safety purpose and the sensor circuit starts to work.

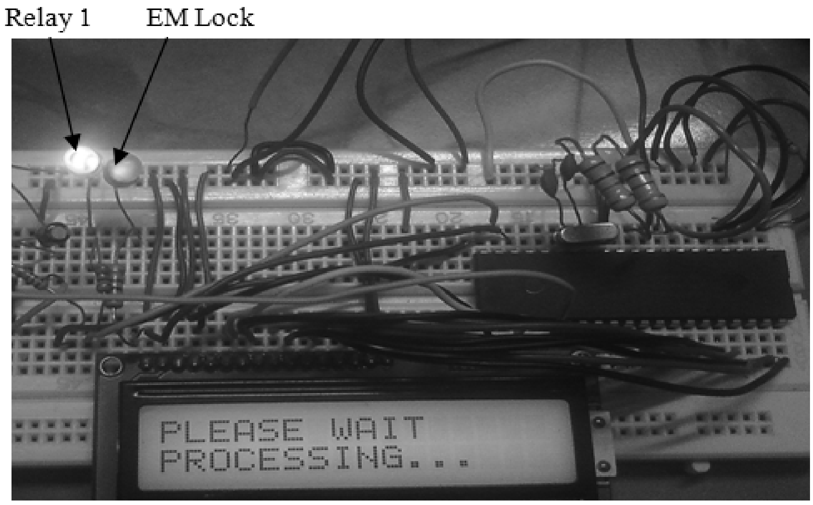
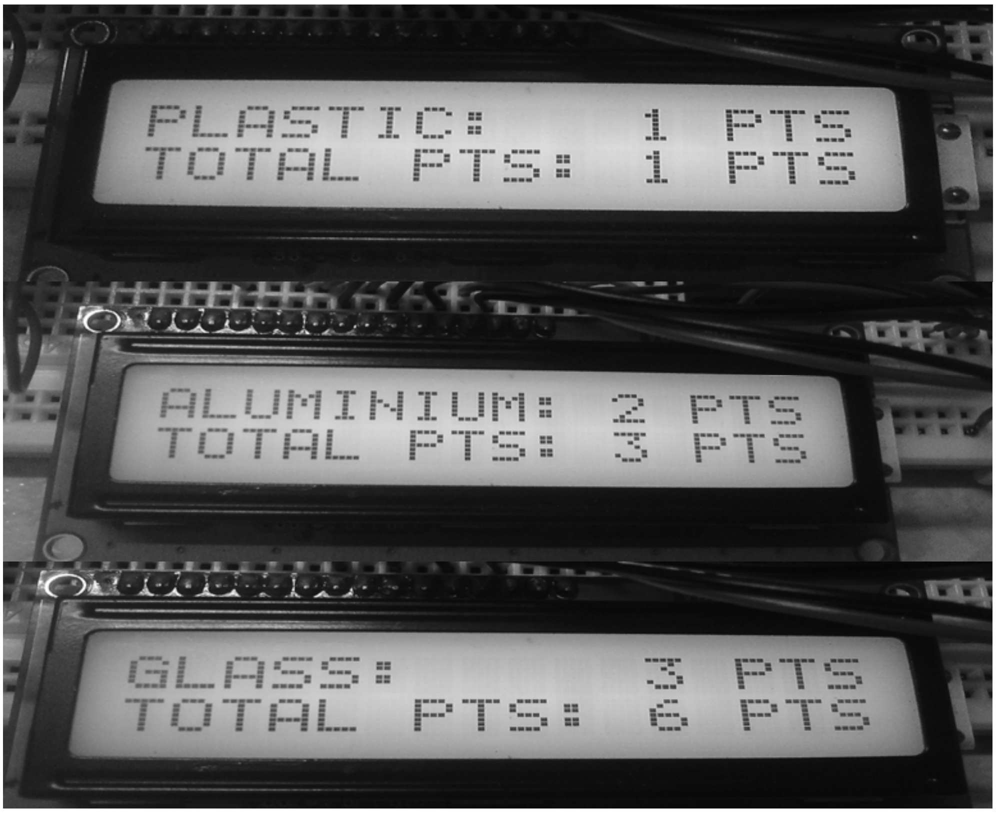


Figure 6.3: LCD displays “PLEASE WAIT” and “PROCESSING”

Then, sensor circuits will provide input on what type of material is detected. LCD proceeds to display the type of material and points awarded according to material type as shown in Figure 6.4. LCD also displays auto summation of points rewarded where the recycler starts by recycling a plastic bottle followed by aluminium can and glass bottle. Plastic bottle is rewarded 1 point, aluminium can is rewarded 2 points and glass bottle is rewarded 3 points.



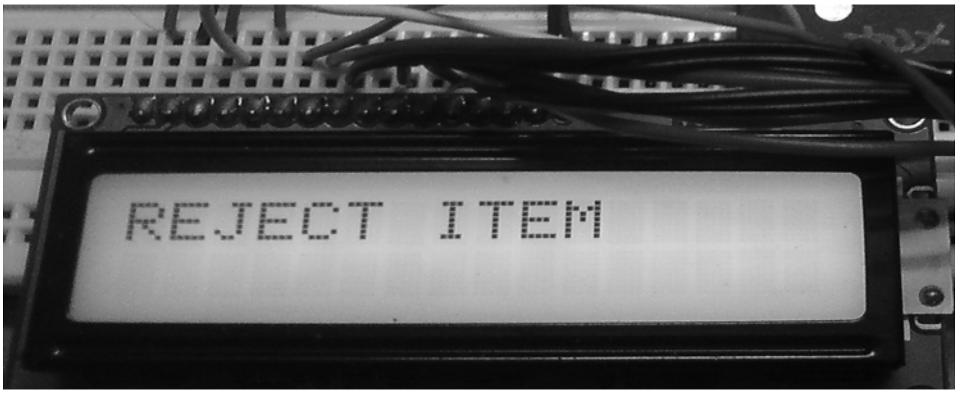
```
PLASTIC:     1 PTS
TOTAL PTS:   1 PTS
```

```
ALUMINIUM:  2 PTS
TOTAL PTS:  3 PTS
```

```
GLASS:        3 PTS
TOTAL PTS:    6 PTS
```

Figure 6.4: LCD displays type of material and summation of total points.

For items that are not recognized by sensors, input will be sent to enable LCD to inform user that it is a rejected item as shown in Figure and loops back to either displaying of total points received (Figure 6.6) or to the Welcome Message (Figure 6.1) if the first item entered is rejected.



```
REJECT ITEM
```

Figure 6.5: LCD displays “REJECT ITEM”.

After every container recycled, the sensor circuits including presence sensors give a LOW input. LCD displays the summation of points followed by user prompt to press “YES” button to end recycling as shown in Figure 6.6. LED turns off to indicate that EM lock is also deactivated to enable user continue recycling.

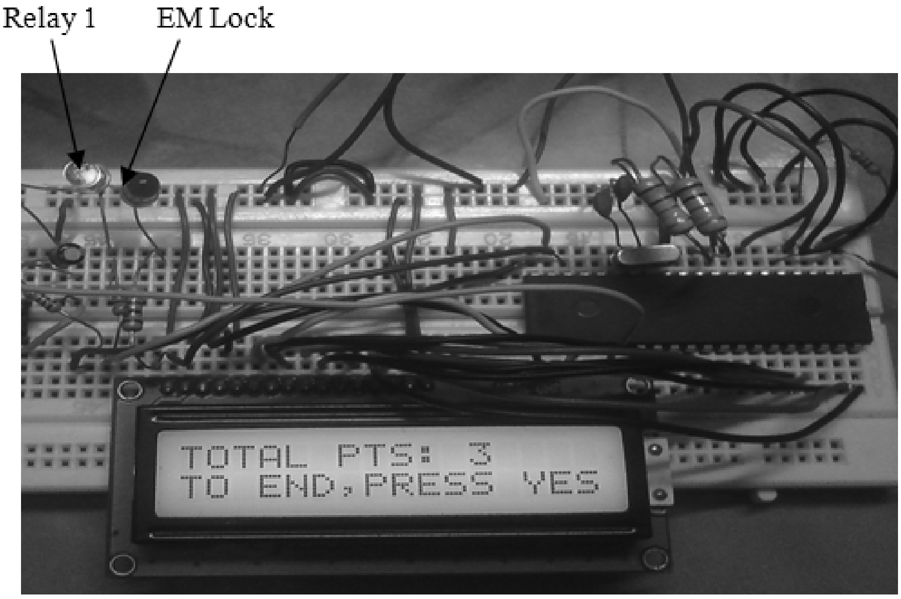


Figure 6.6: LCD displays total points and prompts user for input.

If user pressed “YES” button to end recycling, LCD prompts user whether to store points into card as shown in Figure 6.7.

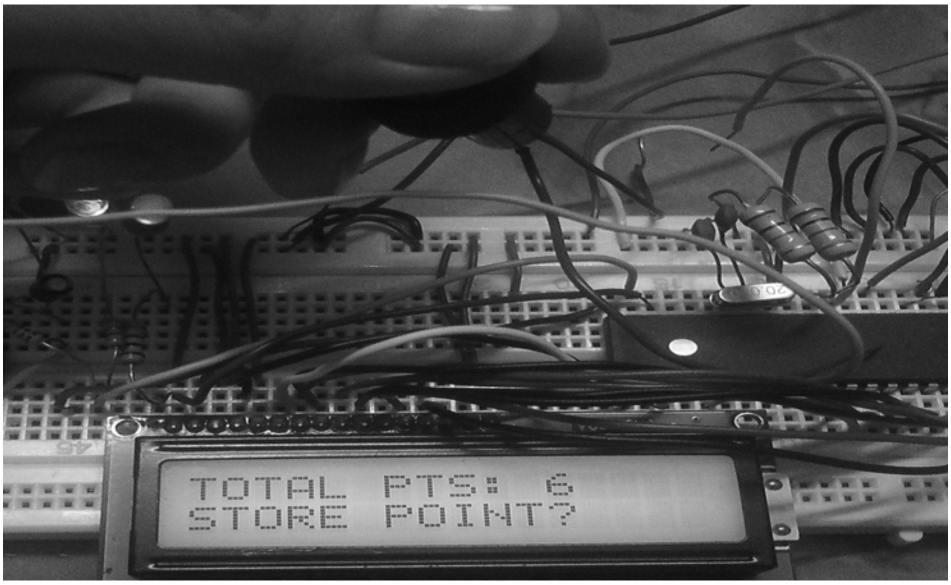


Figure 6.7: LCD prompts user to push button.

A non member would press “NO” button to the prompt “STORE POINTS?”. This would be the end of recycling process and LCD displays a “THANK YOU” message as shown in Figure 6.8.

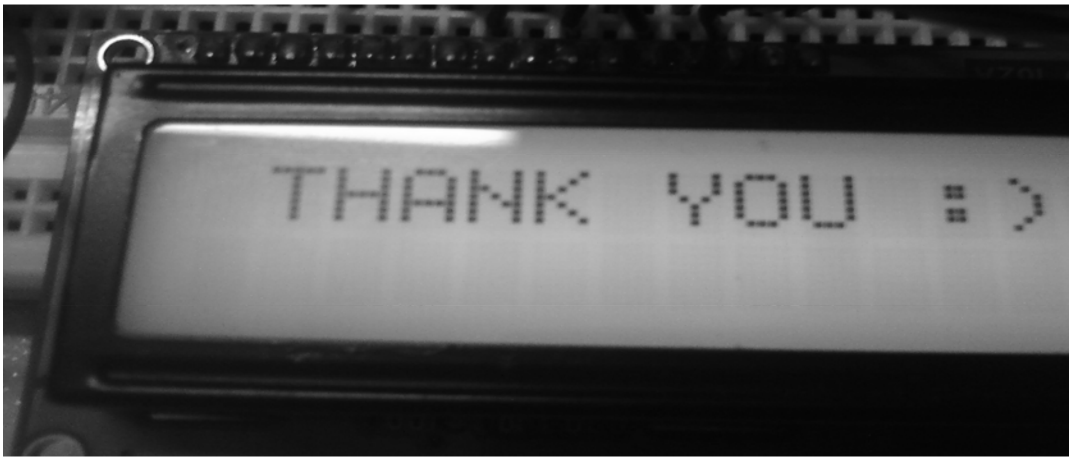


Figure 6.8: LCD displays “THANK YOU” message.

Recycler who wishes to store points into card will push “YES” button and further instruction will be displayed on the LCD as shown in Figure 6.9.

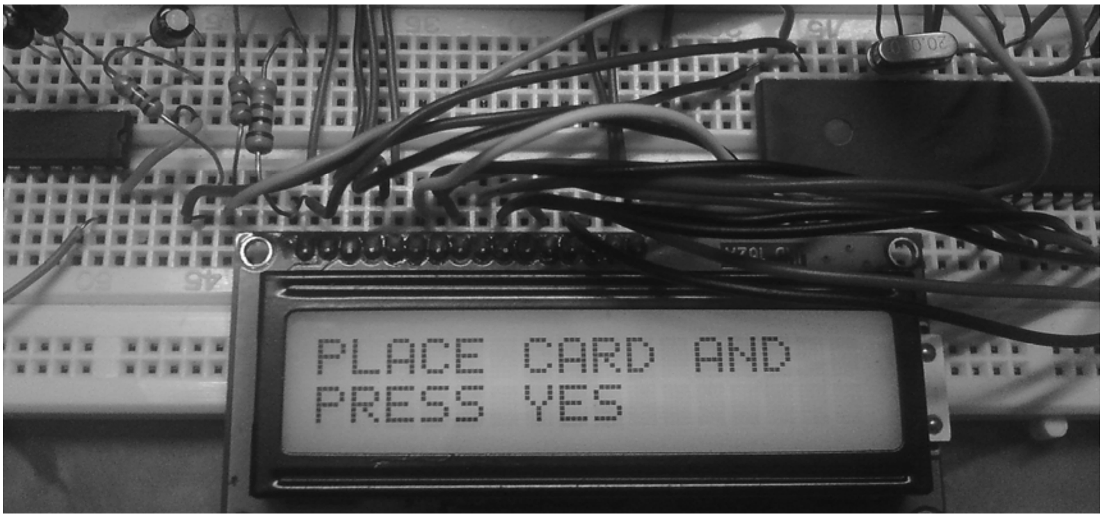


Figure 6.9: LCD displays instruction “PLACE CARD AND PRESS YES”

When the points are successfully stored, the user will be notified by message displayed on LCD as shown in Figure 6.10. After awhile, the LCD will return to the splash screen displaying a welcome message as shown in Figure 6.1.

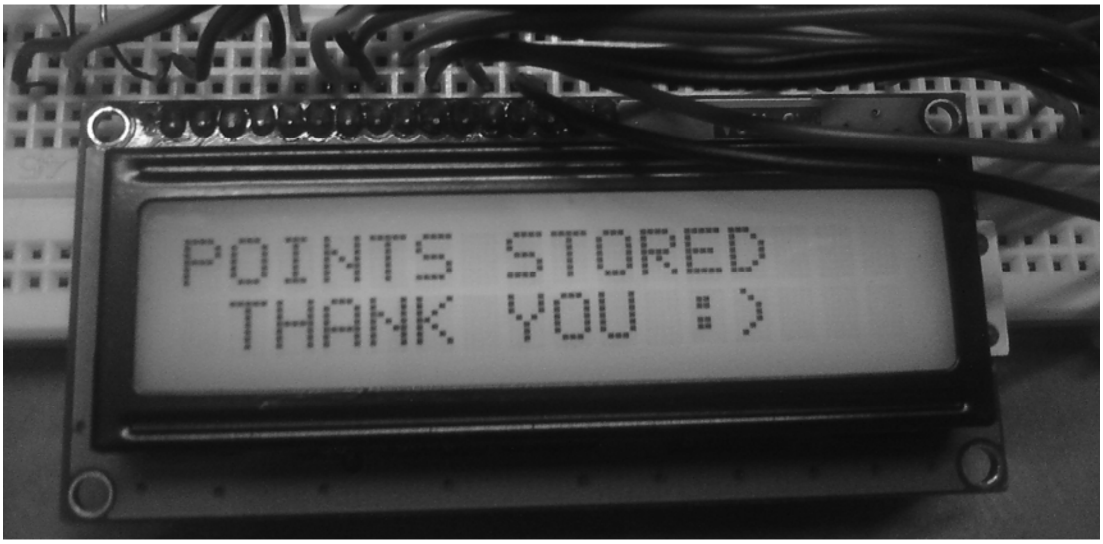


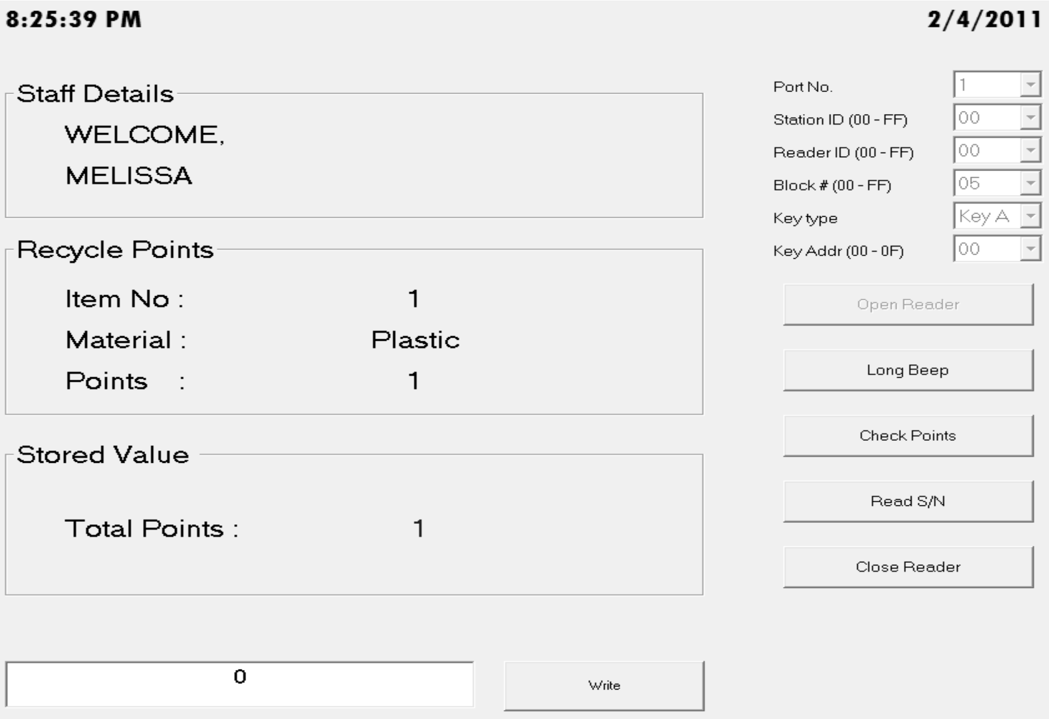
Figure 6.10: LCD displays message “POINTS STORED” and “THANK YOU”

6.2 MIFARE Platform Testing

Visual Basic is used to develop a user friendly application that enables the authorized personnel to make some changes to the data stored in the card such as write (redemption of points) and read (check the current points stored).

Also, the application shows the recycle points while recycling process is done. The actual reason is for ease of demonstration where it displays the type of beverage container material and the points rewarded.

The displaying of information depends on the type of container material recycled. The information includes number of items recycled, the type of material and total points rewarded. When plastic bottle is recycled, the application displays the points rewarded for plastic is 1 point as shown in Figure 6.11.


8:25:39 PM 2/4/2011

Staff Details
WELCOME,
MELISSA

Recycle Points
Item No : 1
Material : Plastic
Points : 1

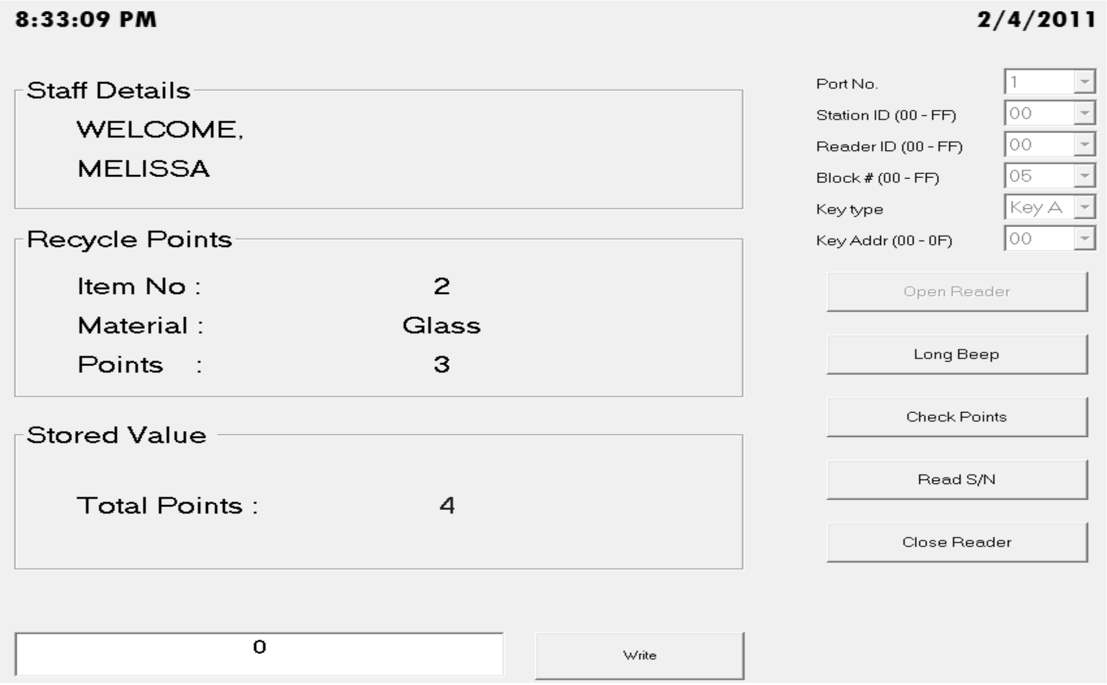
Stored Value
Total Points : 1

Port No. 1
Station ID (00 - FF) 00
Reader ID (00 - FF) 00
Block # (00 - FF) 05
Key type Key A
Key Addr (00 - 0F) 00

0

Figure 6.11: Display material type of plastic with rewarded point of 1.

When glass is inserted after the plastic bottle is recycled, the screen displays the points rewarded is 3 points as shown in Figure 6.12. Also, it shows the summation of total points earned.


8:33:09 PM 2/4/2011

Staff Details
WELCOME,
MELISSA

Recycle Points
Item No : 2
Material : Glass
Points : 3

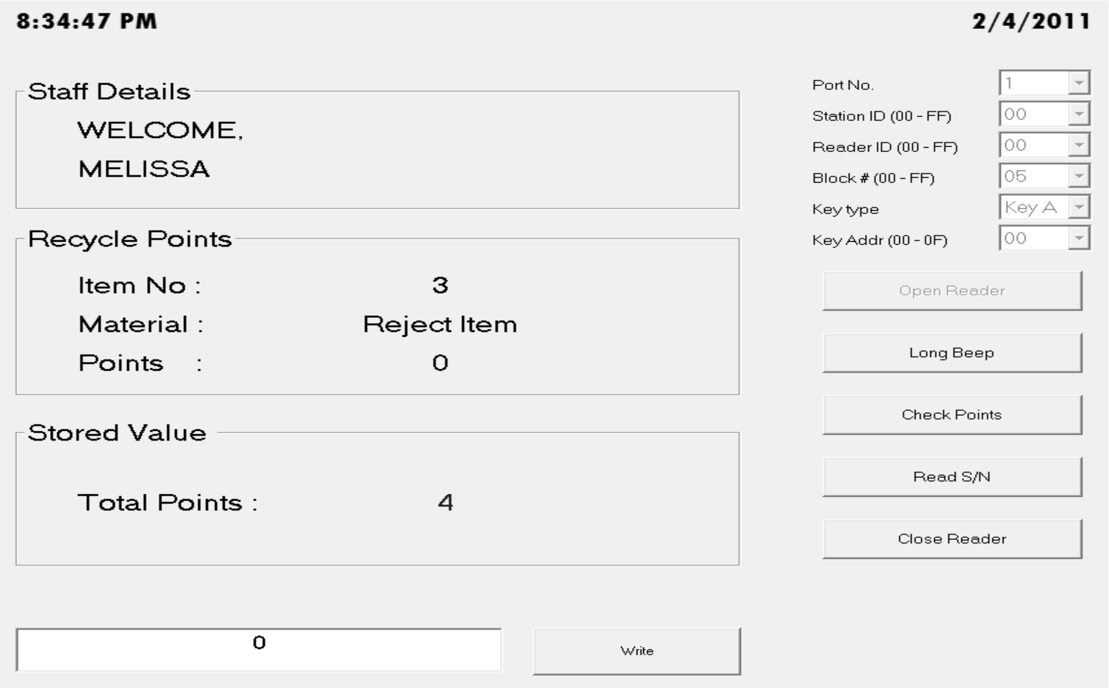
Stored Value
Total Points : 4

Port No. 1
Station ID (00 - FF) 00
Reader ID (00 - FF) 00
Block # (00 - FF) 05
Key type Key A
Key Addr (00 - 0F) 00

0

Figure 6.12: Display material type of glass with rewarded point of 3.

If the item inserted is rejected, no points are rewarded as shown in Figure 6.13. The total points remained unchanged.



8:34:47 PM 2/4/2011

Staff Details
WELCOME,
MELISSA

Recycle Points
Item No : 3
Material : Reject Item
Points : 0

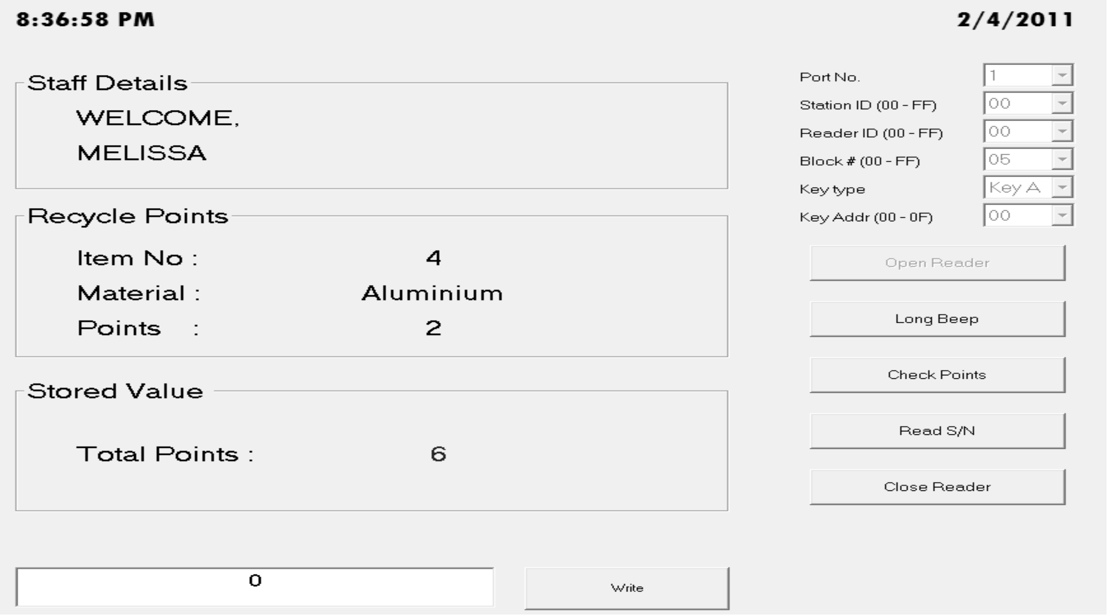
Stored Value
Total Points : 4

Port No. 1
Station ID (00 - FF) 00
Reader ID (00 - FF) 00
Block # (00 - FF) 05
Key type Key A
Key Addr (00 - 0F) 00

0

Figure 6.13: Rejected item detected and no points rewarded.

Aluminium cans will be rewarded 2 points as shown in Figure 6.14. Also, the total points have been increased to 6 points.



8:36:58 PM 2/4/2011

Staff Details
WELCOME,
MELISSA

Recycle Points
Item No : 4
Material : Aluminium
Points : 2

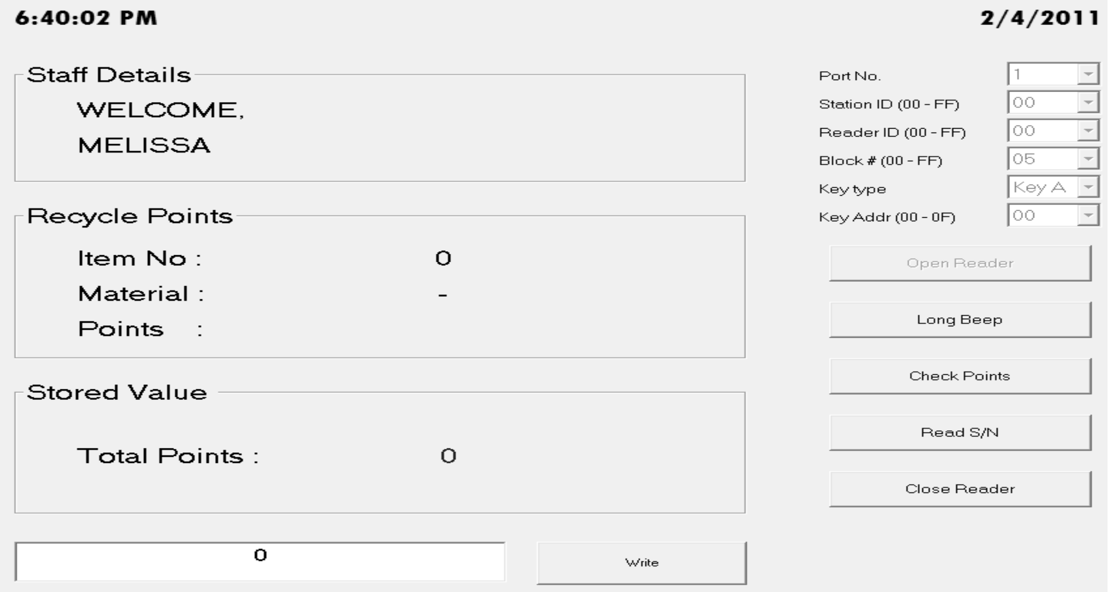
Stored Value
Total Points : 6

Port No. 1
Station ID (00 - FF) 00
Reader ID (00 - FF) 00
Block # (00 - FF) 05
Key type Key A
Key Addr (00 - 0F) 00

0

Figure 6.14: Display material type of aluminium with rewarded point of 2.

The total points are also showed where the recycle points are summed up at the end of the recycling process. When the user presses the button to indicate end of recycling process and flashes card to store points, the total points shown on the screen becomes zero as shown in Figure 6.15.



6:40:02 PM 2/4/2011

Staff Details
WELCOME,
MELISSA

Recycle Points
Item No : 0
Material : -
Points :

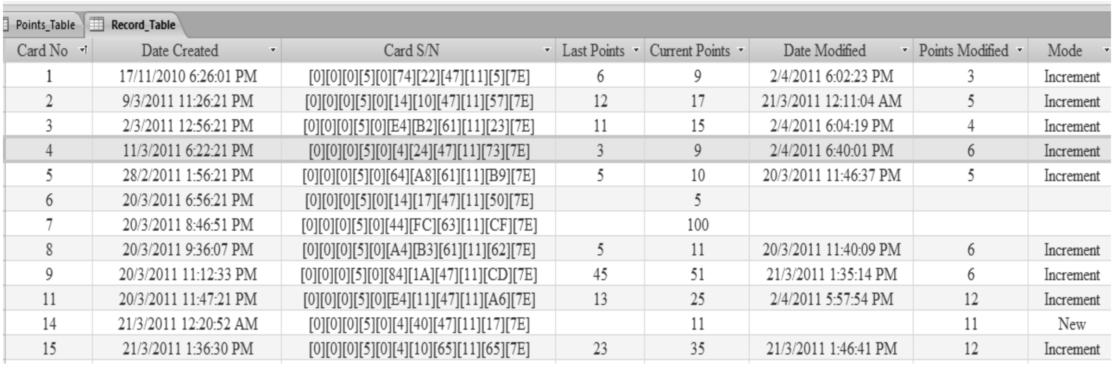
Stored Value
Total Points : 0

Port No. 1
Station ID (00 - FF) 00
Reader ID (00 - FF) 00
Block # (00 - FF) 05
Key type Key A
Key Addr (00 - 0F) 00

0

Figure 6.15: Reward points are stored into the card and the points on screen cleared.

As soon as the points are stored into the card, the database is updated. The activity is recorded in the database listing the date, card serial number, type of activity and the current points in the card as shown in Figure 6.16.

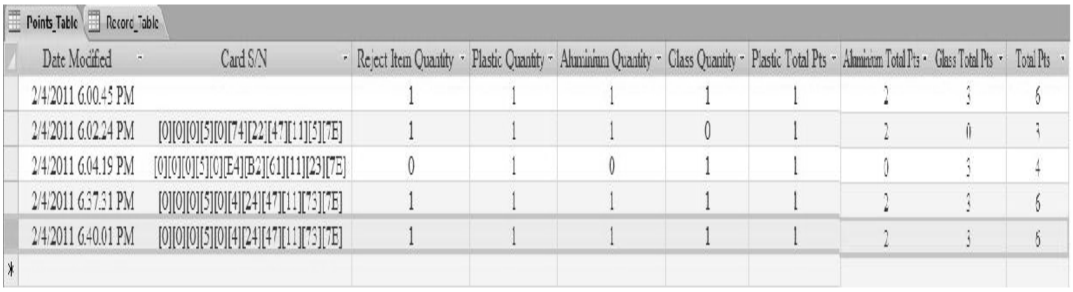


Points_Table Record_Table

Card No	Date Created	Card S/N	Last Points	Current Points	Date Modified	Points Modified	Mode
1	17/11/2010 6:26:01 PM	[0][0][0][5][0][74][22][47][11][5][7E]	6	9	2/4/2011 6:02:23 PM	3	Increment
2	9/3/2011 11:26:21 PM	[0][0][0][5][0][14][10][47][11][57][7E]	12	17	21/3/2011 12:11:04 AM	5	Increment
3	2/3/2011 12:56:21 PM	[0][0][0][5][0][E4][B2][61][11][23][7E]	11	15	2/4/2011 6:04:19 PM	4	Increment
4	11/3/2011 6:22:21 PM	[0][0][0][5][0][4][24][47][11][73][7E]	3	9	2/4/2011 6:40:01 PM	6	Increment
5	28/2/2011 1:56:21 PM	[0][0][0][5][0][64][A8][61][11][B9][7E]	5	10	20/3/2011 11:46:37 PM	5	Increment
6	20/3/2011 6:56:21 PM	[0][0][0][5][0][14][17][47][11][50][7E]		5			
7	20/3/2011 8:46:51 PM	[0][0][0][5][0][44][FC][63][11][CF][7E]		100			
8	20/3/2011 9:36:07 PM	[0][0][0][5][0][A4][B3][61][11][62][7E]	5	11	20/3/2011 11:40:09 PM	6	Increment
9	20/3/2011 11:12:33 PM	[0][0][0][5][0][84][1A][47][11][CD][7E]	45	51	21/3/2011 1:35:14 PM	6	Increment
11	20/3/2011 11:47:21 PM	[0][0][0][5][0][E4][11][47][11][A6][7E]	13	25	2/4/2011 5:57:54 PM	12	Increment
14	21/3/2011 12:20:52 AM	[0][0][0][5][0][4][40][47][11][17][7E]		11		11	New
15	21/3/2011 1:36:30 PM	[0][0][0][5][0][4][10][65][11][65][7E]	23	35	21/3/2011 1:46:41 PM	12	Increment

Figure 6.16: Record Table is the database that records the current points in card and the activity done.

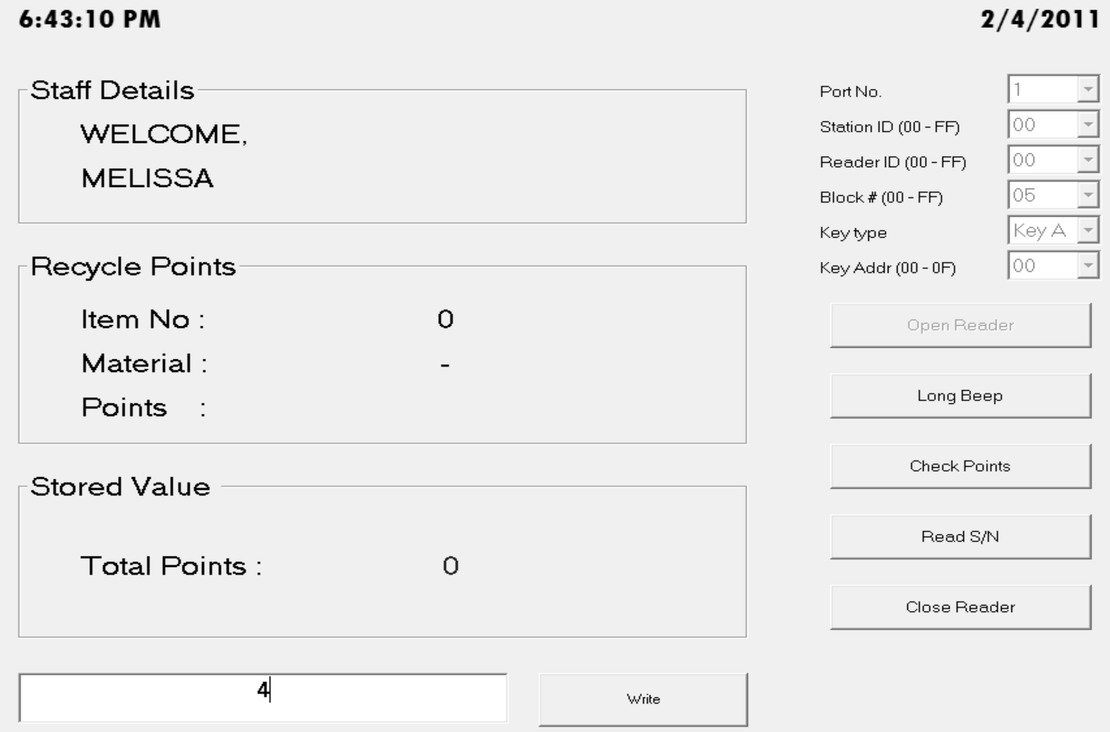
Points Table is created to record types of containers recycled and total points rewarded according to card Serial Number to keep track of member activities as shown in Figure 6.17.



Date Modified	Card S/N	Reject Item Quantity	Plastic Quantity	Aluminium Quantity	Glass Quantity	Plastic Total Pts	Aluminium Total Pts	Glass Total Pts	Total Pts
2/4/2011 6:00:45 PM		1	1	1	1	1	2	3	6
2/4/2011 6:02:24 PM	[0][0][0][5][0][74][22][47][11][5][7E]	1	1	1	0	1	2	0	3
2/4/2011 6:04:19 PM	[0][0][0][5][C][E4][B2][61][11][23][7E]	0	1	0	1	1	0	3	4
2/4/2011 6:37:31 PM	[0][0][0][5][0][4][24][47][11][73][7E]	1	1	1	1	1	2	3	6
2/4/2011 6:40:01 PM	[0][0][0][5][0][4][24][47][11][73][7E]	1	1	1	1	1	2	3	6
*									

Figure 6.17: Points Table is the database that records the quantity of beverage containers recycled and rewarded points.

This application also enables the redemption of points. Just like any loyalty card, the points will be redeemed at the counter with a staff updating the card points from the platform. The staff will check the points stored in the card and manually type the points to be redeemed and click the “write” button to deduct the redeemed points from the card as shown in Figure 6.18.



6:43:10 PM 2/4/2011

Staff Details
WELCOME,
MELISSA

Recycle Points
Item No : 0
Material : -
Points :

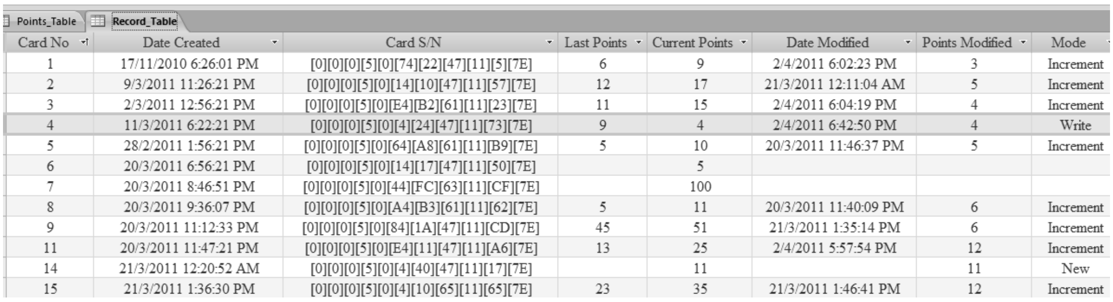
Stored Value
Total Points : 0

Port No. 1
Station ID (00 - FF) 00
Reader ID (00 - FF) 00
Block # (00 - FF) 05
Key type Key A
Key Addr (00 - 0F) 00

4

Figure 6.18: The card now holds 4 points instead of 9 points.

To verify that the points have been redeemed, the Record Table now shows that the current points stored in card is 4 as shown in Figure 6.19.



Card No	Date Created	Card S/N	Last Points	Current Points	Date Modified	Points Modified	Mode
1	17/11/2010 6:26:01 PM	[0][0][0][5][0][74][22][47][11][5][7E]	6	9	2/4/2011 6:02:23 PM	3	Increment
2	9/3/2011 11:26:21 PM	[0][0][0][5][0][14][10][47][11][57][7E]	12	17	21/3/2011 12:11:04 AM	5	Increment
3	2/3/2011 12:56:21 PM	[0][0][0][5][0][E4][B2][61][11][23][7E]	11	15	2/4/2011 6:04:19 PM	4	Increment
4	11/3/2011 6:22:21 PM	[0][0][0][5][0][4][24][47][11][73][7E]	9	4	2/4/2011 6:42:50 PM	4	Write
5	28/2/2011 1:56:21 PM	[0][0][0][5][0][64][A8][61][11][B9][7E]	5	10	20/3/2011 11:46:37 PM	5	Increment
6	20/3/2011 6:56:21 PM	[0][0][0][5][0][14][17][47][11][50][7E]		5			
7	20/3/2011 8:46:51 PM	[0][0][0][5][0][44][FC][63][11][CF][7E]		100			
8	20/3/2011 9:36:07 PM	[0][0][0][5][0][A4][B3][61][11][62][7E]	5	11	20/3/2011 11:40:09 PM	6	Increment
9	20/3/2011 11:12:33 PM	[0][0][0][5][0][84][1A][47][11][CD][7E]	45	51	21/3/2011 1:35:14 PM	6	Increment
11	20/3/2011 11:47:21 PM	[0][0][0][5][0][E4][11][47][11][A6][7E]	13	25	2/4/2011 5:57:54 PM	12	Increment
14	21/3/2011 12:20:52 AM	[0][0][0][5][0][4][40][47][11][17][7E]		11		11	New
15	21/3/2011 1:36:30 PM	[0][0][0][5][0][4][10][65][11][65][7E]	23	35	21/3/2011 1:46:41 PM	12	Increment

Figure 6.19: Record Table verifies the redemption of points.

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

The conclusion section closes the report by providing a summary to the content in the report. Also, the potential applications of the results and recommendations for future work are included.

7.1 Summary

The aim and objectives of this project has been achieved. The prototype of Reverse Vending Machine is successfully built and the whole system can function successfully.

The display of messages and user prompts for the display system on the Smart Reward and Recycle Bin was successful implemented by PIC16F877A microcontroller board to receive the data from sensor circuits and control the LCD. C programming was used to write the program for the microcontroller. The program written includes all messages and user prompts that follows the conditions of events. Also, the program enables the summation of points throughout the recycling process. UART is used to send data to the Visual Basic program on a PC.

The Visual Basic program for managing the reward system serves its function well by displaying staff details, details of recycled item and points rewarded as well as enable the redemption of points.

The electronic reward system using MIFARE card technology has achieved its intended function of storing reward points into the card and retrieving the data stored in card via a Mifare card reader/writer connected to the PC. By using MSComm function, it is also able to communicate with microcontroller which gives sensor inputs from the material recognition system. Also, ADO Data Control enables the storing of information into the database created using Microsoft Access. The database records the activities such as date of recycling process, the recycler's card details, the items recycled and points rewarded.

However, there are some limitations of the prototype. The system itself is still imperfect with certain shortcomings. Possible future works will be discussed later in this chapter. Overall, the system can be implemented successfully. The system as a whole provides a cost effective and simple solution for the implementation of Reverse Vending Machine in our country.

7.2 Recommendations

Being a prototype for the Reverse Vending Machine, the system is still at its infancy stage and further development should be made to enhance its reliability, increase its accuracy and functions.

From an overview of our project, there are many obstacles hindering the optimization of the system. Budget allocated in resource limits the purchase of high end devices, lack of research regarding Reverse Vending Machine in Malaysia, and absence of a standardized database for implementation of UPC scanner, not to mention its wide database collection if the prototype was to implement digital image processing approach.

7.2.1 Potential Application

Smart Recycle and Reward Bin is meant to encourage the public to develop the habit for recycling by rewarding them with points. Thus, the machine should be located at public places such as shopping complexes.

The private sectors can play a major role by sponsoring the machines. Take for instance, the supermarket sponsors the machine and encourages its shoppers to recycle. In return, the shoppers apply for loyalty cards where they are able to collect their reward points. The reward points can be used to redeem shopping vouchers or goodies from the supermarket.

7.2.2 Future Works

The main recommendation is to eliminate the use of PC in the system to reduce the cost. In this project, the use of PC is to allow the redemption of points through staff platform, tracking of points rewarded and card information through database and transferring of data between PIC and MIFARE module.

One recommended approach is to use Rabbit 2000 microcontroller development kit which is not implemented in this project due to limitation of budget.

Table 7.1: Comparison between Conventional controller and Rabbit Controller.

Conventional Controller	Rabbit Controller
Typical C	Dynamic C
Ordinary serial ports	Serial ports with library functions support
Smart card interface not ready	Smart card interface ready for transport layer programming
Microcontroller chip	Microcontroller module
Can’t compile directly. Need separate programmer	Compile direct into flash or RAM
Need separate emulator	Dynamic C debugger

A command layer handle and interprets command, transport layer handle low level command message which the rabbit controller communicates with smart card reader in this layer and physical layer handles data transmission. The device is connected together with serial port interface using a RS232 cable.

Another approach to enable communication between PIC16F877A and MIFARE module is by open the dll file to see the code written. However, it will be a difficult process and may have copyright issues. A DLL is a library file that contains functions for other programs to use. In order for these functions to be run their code must have been compiled. Compiling is the process of taking source code in a language like C++ or Visual Basic which can be read by humans, and converting it into machine code that can be read by a computer processor. As such it is very hard to 'read' DLL files as the actual functional code is machine code. In order to read the source code, future work involves attempting to reverse-engineer the machine code.

The next recommendation for future work is to further optimize memory of smart card. For this project, MIFARE module and card is used to solely store reward points because the application here is as loyalty cards. Further optimization of the memory can be done by storing different data at different blocks. The key authentication can be set to ensure safety since the access conditions will be required.

REFERENCES

Allen, J. Davis, & Soskin. (1993). Using coupon incentives in recycling aluminum: A Market Approach to Energy Conservation Policy. *The Journal of Consumer Affairs*, Volume 27, 300.

Bannatyne, R. and Viot, G. (1997). Introduction to Microcontrollers. *Journal of IEEE*, 0-7803-4303-4. pg 564-574

Leong,S.L, (1998, April 26). Promoting Recycling Culture. *The New Straits Times Press*.

Leuder, E. (2003). *Liquid Crystal Display*. John Wiley& Sons, Ltd.

Muhammad Tarmizi Lockman, Ali Selamat (2008). "Verification and Validation Communication Layer of Embedded Smart Card System." Universiti Teknologi Malaysia: Thesis Bachelor Of Degree.

Rankl, Wolfgang. (2010). *Smart Card Handbook* (4th ed.). John Wiley and Sons, Ltd

Return Recycling News, (2005). Retrieved March 29, 2011, from http://www.tomra.com

Benefits of Smart Cards. (2000). Retrieved March 22, 2011, from http://www.buzzle.com/articles/ benefits-of-smart-cards.html

Reverse Vending Machine [Brochure]. (2009). Retrieved from Reverse Vending Corporation website: http://www.reversevending.co.uk/

Smart Card Basics. (2010). Retrieved March 21, 2011, from CardLogix Corporation website: http://www.smartcardbasics.com/smart-card-overview.html

Microsoft Visual Basic. (2011, April 4). Retrieved March 29, 2011, from http://en.wikipedia.org/wiki/Microsoft_Visual_Basic

APPENDICES

APPENDIX A: Instruction List for MIFARE

MyReader Instruction		
Command		**Functions**
Pcd_eread	10H	Read a block of data from card. Load keys from eeprom.
Pcd_ewrite	11H	Write a block of data to the card. Load keys from eeprom.
Pcd_evalue	12H	Do arithmetic to the value of card. Functions include increment, decrement, and restore. Load keys from eeprom.
Pcd_evalue2	13H	Do arithmetic to the value of card. Decrement and transfer. For mifare light only. Load keys from eeprom.
Pcd_econtinuousread	14H	Continuously read a block of data from card. Load keys from eeprom.
Pcd_sread	20H	Read a block of data from a selected card. Load keys directly.
Pcd_swrite	21H	Write a block of data to a selected card. Load keys directly.
Pcd_svalue	22H	Do arithmetic to the value of a selected card. Load keys directly.
Pcd_svalue2	23H	Do arithmetic to the value of a selected card. For mifare light only. Load keys directly.
Pcd_shalt	24H	Turn off a selected card in field.
Pcd_seread	30H	Read a block of data from a selected card. Load keys from eeprom.
Pcd_sewrite	31H	Write a block of data to a selected card. Load keys from eeprom.
Pcd_sevalue	32H	Do arithmetic to the value of a selected card. Load keys from eeprom.
Pcd_sevalue2	33H	Do arithmetic to the value of a selected card. For mifare light only. Load keys from eeprom.
Pcd_read	40H	Read a block of data from card. Load keys directly.
Pcd_write	41H	Write a block of data to card. Load keys directly.

Pcd_value	42H	Do arithmetic to the value of card. Load keys directly.
Pcd_value2	43H	Do arithmetic to the value of card. For mifare light only. Load keys directly.
Pcd_areadsnr	44H	Continuously read the serial number of type A cards in field.
Pcd_stop	45H	Stop the continuously reading operation.
Pcd_loadkey2	46H	Load keys into eeprom.
Pcd_continuousread	47H	Continuously read a block of data from card. Load keys directly.
Pcd_areadonesnr	48H	Read type A card serial number.
Pcd_readid	50H	Read the reader id number.
Pcd_writeid	51H	Change the reader id number.
Pcd_readproductsnrno	52H	Read the unique reader IC number
Pcd_readproducttypeno	53H	Read the product type identification number
Pcd_rfoff	54H	Turn off the rf field.
Pcd_rfon	55H	Turn on the rf field.
Pcd_Trigger_control	58H	Enable or disable I/O pin.
Pcd_Trigger_IO	59H	Trigger I/O pin.
Pcd_resetrf	5aH	Reset the rf field.
Pcd_wake_up	5bH	Send a Mifare Request_All to reset the cards in the field.
Pcd_change_protocol	5cH	Switch to another protocol.
Pcd_beep	5dH	Generate a beep sound. (Provided buzzer is available.)
Pcd_short_beep	5eH	Generate a short beep sound. (Provided buzzer is available.)
Pcd_abreadsnr	5fH	Continuously read type A and B card UID.
Pcd_loadtypeareg	60H	Load type A registers into pcd.
Pcd_loadtypebreg	61H	Load type B registers into pcd.
Get_keypaddata	63H	Detect if any key button is being pressed. (Provided keypad is available.)
Trigger IO1	64H	Set IO 1 to ‘0’, or ‘1’.
Trigger IO2	65H	Set IO 2 to ‘0’, or ‘1’.
Trigger Relay	66H	Set relay to ‘0’, or ‘1’.
Get IO	67H	Get the IO1, IO2 and relay status.
Trigger_Buzzer_On	68H	Turn On Buzzer
Trigger_Buzzer_Off	69H	Turn Off Buzzer
Trigger_Success_LED	6AH	Turn On the Success / Working LED or Turn Off the PWR LED
Trigger_Power_LED	6BH	Turn Off the Success / Working LED or Turn On the PWR LED
Trigger_LED_OFF	6CH	Turn off the LEDs
Trigger_LED_ON	6DH	Turn on the LEDs
Buzzer_control	6EH	User defines number of beeps and duration of beep sound.

Buzzer_control_LED	6FH	User defines number of beeps and duration of beep sound, it also trigger LED.
Pcd_initiate	70H	Execute Initiate command to type B card.
Pcd_select	71H	Execute Select command to type B card.
Pcd_completion	72H	Execute Completion command to type B card.
Pcd_readblock	73H	Execute Read Block command to type B card.
Pcd_writeblock	74H	Execute Write Block command to type B card.
Pcd_protectblock	75H	Execute Protect Block command to type B card.
Pcd_getprotection	76H	Execute Get Protection command to type B card.
Pcd_breadsnr	79H	Continuously read type B card UID
Pcd_breadblock	7aH	Execute Initiate, Select, and Read Block commands to type B card.
Pcd_bwriteblock	7bH	Execute Initiate, Select, and Write Block commands to type B card.
Pcd_bprotectblock	7cH	Execute Initiate, Select, and Protect Block command to type B card.
Pcd_bgetprotection	7dH	Execute Initiate, Select, and Get Protection command to type B card.
---	---	Reserved
---	---	Reserved
Pcd_breadonesnr	81H	Read type B card ID.
Pcd_longledbuzzer	82H	Trigger success LED and beep sound for approximately 0.6 second.
Pcd_biledbuzzer	83H	Trigger success LED and beep sound two times.
LCD_line0	FAH	Display a string of characters on the LCD line 0 (Provided LCD is available.)
LCD_line1	FBH	Display a string of characters on the LCD line 1 (Provided LCD is available.)
LCD_line2	FCH	Display a string of characters on the LCD line 2 (Provided LCD is available.)
LCD_lone3	FDH	Display a string of characters on the LCD line 3 (Provided LCD is available.)

APPENDIX B: MFR API Library Ver. 2.10

```
Option Explicit

Global Const CMD_OK = 0
Global Const CMD_FAIL = &H1&
Global Const CMD_SEND_FAIL = &H2&
Global Const CMD_CANCEL = &H4&
Global Const CMD_RESETDATA = &H9020&
Global Const CMD_READ = &H40&
Global Const CMD_CHK_CARD = &H9040&
Global Const CMD_WRITE = &H41&
Global Const CMD_WRITE_VAL = &H41FF&
Global Const CMD_EREAD = &H10&
Global Const CMD_EWRITE = &H11&
Global Const CMD_EWRITE_VAL = &H4011&
Global Const CMD_SEL_READ = &H20&
Global Const CMD_SEL_WRITE = &H21&
Global Const CMD_SEL_WRITE_VAL = &H4021&
Global Const CMD_SEL_HALT = &H24&
Global Const CMD_SEL_EREAD = &H30&
Global Const CMD_SEL_EWRITE = &H31&
Global Const CMD_SEL_EWRITE_VAL = &H4031&
Global Const CMD_RESET = &H5A&
Global Const CMD_STOP = &H45&
Global Const CMD_LOADKEY2 = &H46&
Global Const CMD_READ_SERIAL_NO = &H48&
Global Const CMD_READ_ID = &H50&
Global Const CMD_WRITE_ID = &H51&
Global Const CMD_SWITCH_READER = &H5010&
Global Const CMD_RF_OFF = &H54&
Global Const CMD_RF_ON = &H55&
Global Const CMD_TRIGGER_CONTROL = &H58&
Global Const CMD_TRIGGER_IO = &H59&
Global Const CMD_WAKE_UP = &H5B&
Global Const CMD_VALUE = &H42&
Global Const CMD_EVALUE = &H12&
Global Const CMD_SEL_VALUE = &H22&
Global Const CMD_SEL_EVALUE = &H32&

Global Const CMD_LONG_BUZZER = &H5D&
Global Const CMD_SHORT_BUZZER = &H5E&

Global Const CMD_DISP_LCD_LINE1 = &H1&
Global Const CMD_DISP_LCD_LINE2 = &H2&
Global Const CMD_DISP_LCD_LINE3 = &H3&
Global Const CMD_DISP_LCD_LINE4 = &H4&
Global Const CMD_CLEAR_LCD = &H4004&

Global Const CMD_DETECT_KEY_PRESSED = &H5E&
Global Const CMD_DETECT_KEY_PRESSED1 = &H915E&
Global Const CMD_INPUT_STR = &H905E&
Global Const CMD_BUZZER = &H5F&
Global Const CMD_CURSOR_OFF = &H60&
Global Const CMD_CURSOR_ON = &H61&
Global Const CMD_DISP_CHAR = &H62&
Global Const CMD_DISP_STR = &H9062&
Global Const CMD_SHORT_BEEP = &H63&
Global Const CMD_GET_MSG = &H8000&
Global Const CMD_GET_LIB_VER = &H8100&


'For card arithmetic operations
Global Const DECREMENT = &HC0&
Global Const INCREMENT = &HC1&
Global Const RESTORE = &HC2&

'KEY VALUE TYPE
Global Const KEY_TYPE_A = 0
Global Const KEY_TYPE_B = 1

'ERR DEFINITIONS
Global Const ERR_UNKNOWN_CMD = &H9999&
Global Const ERR_READER_NOT_OPENED = &H9000&
Global Const ERR_FAIL_TO_CLOSE_READER = &H9010&
Global Const ERR_INCOMPLETE_DATA = &H9020&
Global Const ERR_WRONG_BCC = &H9030&
Global Const ERR_TIMEOUT = &H9040&
Global Const ERR_UNKNOWN = &H9050&
Global Const ERR_NO_EOT = &H9060&
Global Const ERR_NO_ACK = &H9070&
Global Const ERR_INVALID_BUFFER = &H9080&
Global Const ERR_NO_DATA = &H9090&
Global Const ERR_BUFFER_OVERFLOW = &H90A0&
Global Const ERR_BAD_LCD_POSITION = &H90B0&
Global Const ERR_INVALID_CARD = &H90FF&
Global Const ERR_MI_UNDEFINED = &H8000&
Global Const ERR_MI_NOTAG = &H8001&
Global Const ERR_MI_CRC = &H8002&
Global Const ERR_MI_EMPTY = &H8003&
Global Const ERR_MI_AUTH = &H8004&
Global Const ERR_MI_PARITY = &H8005&
Global Const ERR_MI_CODE = &H8006&
Global Const ERR_MI_SERN = &H8008&
Global Const ERR_MI_KEY = &H8009&
Global Const ERR_MI_NOTAUTHERR = &H800A&
Global Const ERR_MI_BITCOUNTER = &H800B&
Global Const ERR_MI_BYTECOUNTER = &H800C&
Global Const ERR_MI_IDLE = &H800D&
Global Const ERR_MI_TRANS = &H800E&
Global Const ERR_MI_WRITE = &H800F&
Global Const ERR_MI_INCR = &H8010&
Global Const ERR_MI_DECR = &H8011&
Global Const ERR_MI_READ = &H8012&
Global Const ERR_MI_OVERFLOW = &H8013&
Global Const ERR_MI_POLLING = &H8014&
Global Const ERR_MI_FRAMING = &H8015&
Global Const ERR_MI_ACCESS = &H8016&
Global Const ERR_MI_INIT = &H801A&
Global Const ERR_MI_INTERFACE = &H801B&
```

```
Type MF_MSG
   nCmd                       As Long
   nBlockNo                   As Long

   nKeyType                   As Long
   nKeyAddr                   As Long

   SerialNo(0 To 15)          As Byte
   nSerialNoLen               As Long

   KeyValue(0 To 15)          As Byte
   nKeyValueLen               As Long

   data(0 To 1023)            As Byte
   nDataLen                   As Long

   nCardType                  As Long
   nEcho                      As Long
   nBeep                      As Long
   nPassword                  As Long
   nValue                     As Long
   nTransferBlockNo           As Long
   nValueOpMode               As Long
   nTimeout                   As Long
   nLength                    As Long
   nHaltAfterCommand          As Long
End Type



' MyReader APIs
' OpenReader() returns the non-zero handle which is needed in CloseReader() and DoMF
'
Declare Function OpenReader Lib "mfr.dll" (ByVal nPortNo As Long, ByVal nStationID As Byte, ByVal
nReaderID As Byte) As Long
Declare Function CloseReader Lib "mfr.dll" (ByVal handle As Long) As Long
Declare Function DoMF Lib "mfr.dll" (ByVal handle As Long, ByRef mfmsg As MF_MSG) As Long
Declare Function SendMF Lib "mfr.dll" (ByVal handle As Long, ByVal msgHandle As Long) As Long
Declare Sub GetMifareKey Lib "mfr.dll" (sn As Byte, ByVal SectorNo As Byte, key As Byte)

' System APIs
Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
```