UITM SMART ATTENDANCE ACCESS CONTROL SYSTEM

Thesis presented in partial fulfillment for the award of the
Bachelor in Electrical Engineering (Honors)
Of
UNIVERSITI TEKNOLOGI MARA MALAYSIA

ZAHARAH BINTI HAIRUZI
FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITI TEKNOLOGI MARA
40450 SHAH ALAM, MALAYSIA
JANUARY – APRIL 2010

APPROVAL SHEET

SUBMISSION OF PROJECT REPORT (THESIS) FOR
FINAL YEAR PROJECT 2
(EEE690)
B.ENG (HONS) ELECTRICAL ENGINEERING

This is to certify that the necessary amendment has been made to the project report (thesis) according to the recommendation.

This Thesis is approval by:

...…………………………………………………………………

MISS RAFIDAH BINTI ROSMAN (PROJECT SUPERVISOR)

Date: ………………………

DECLARATION

It is hereby declared that all the materials in this thesis are the results of my own work has been clearly acknowledge in the thesis

ACKNOWLEDGEMENT

In the name of Allah the most Beneficent and Merciful. A deep sense of thankfulness to Allah who has given me the strength, ability and patience to complete this project and thesis as it is today.

Firstly, I would like to take this opportunity to put into words my deepest gratitude and appreciation to the project supervisor, Miss Rafidah binti Rosman and Mr. Ahmad Ihsan bin M. Yassin as my guider, for their support, guidance, patience, encouragement and abundance of ideas during the completion of this project.

Secondly, special thanks to honourable panels, Dr. Azilah and Mr. Syed Abdul Mutalib for their comments, invaluable suggestions and outstanding deliberations to improve the project during the project presentation.

I would also like to express my extraordinary appreciation to my family and friends for their invaluable support along the duration of my studies until the completion of this thesis.

Finally yet importantly, thanks to all the persons who are directly or indirectly contributed because their perspective and guidance helped greatly to point me in the right direction until the completion of this thesis.

"THERE'S LIGHT AT THE END OF THE TUNNEL".

ABSTRACT

Traditional attendance recording systems are still being used in most educational institutions. The traditional attendance recording method involves the distribution of attendance sheets, which are signed by students. This method of recording attendance are wastes time, as well as disrupting the student's attention in class. Furthermore, the lecturers need to check the record and calculate the attendance manually. Consequently, this influences the quality of teaching system. UiTM Smart Attendance Access Control System is proposed as a solution to this problem. This system uses Radio Frequency Identification (RFID) to access the recording attendance system which is the easier, faster and in an efficient way. The attendance will be captured immediately once the students place their ID card on the reader and the system is automated calculation of the attendance. Minimum requirements for UiTM students to qualify to seat in examination is 80%, hence the system will view the status of the students for their eligibility. The system was developed using the PIC16F877A microcontroller by Microchip and integrated with the computer through RS232 or Universal Serial Bus (USB) port. Attendance is stored in a Microsoft Access database, using integration with Visual Basic 6.0 software. A prototype of the system has been successfully developed.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE	TITLE	PAGE

LIST OF TABLES

LIST OF ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
CCS	Custom Computer Services
CCD	Charge-Coupled Device
DBMS	Database Management System
DB9	D-sub Connectors 9-pins
DIP	Dual Inline Package
EEPROM	Electrically Erasable Programmable Read-Only Memory
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
ICSP	In-Circuit Serial Programming
ID	Identification
IDE	Integrated Development Environment
I/O	Input/ Output
LCD	Liquid Crystal Display
MPLAB	Micro Processor LAB
MS	Microsoft
MCU	Microcontroller Unit
PC	Personal Computer
PCB	Printed Circuit Board
PIC	Peripheral Interface Controller
RAM	Random Access Memory
RF	Radio Frequency
RFID	Radio Frequency Identification
RS232	Recommended Standard 232
ROM	Read Only Memory
SMS	Short Message Service
TTL	Transistor-Transistor Logic
UiTM	Univerisiti Teknologi Mara
USART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VB6	Visual Basic 6.0

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Most academic institutions implement class attendance systems to collect the attendance record of students in classes. The purpose of this system is to ensure that the students attend class on a regular and consistent basis during the whole of the semester. Disciplinary actions are usually taken on students who do not attend class regularly, including barring from examinations, etc.

Conventional method for attendance collection is distribution of attendance sheets during class. The attendance sheets are passed between students in the classroom, and the students take turn in signing them. The lecturer must then collect and verify the attendance of the students manually, and decide whether a student meets the minimum attendance requirements to sit for examination.

Because validation and insertion of data are done manually, this procedure of taking attendance during class wastes a lot of lesson time. Moreover, the lecturer doesn't specifically know whether the students are actually in the classroom, as there is a large number of students are involved. For lecturers who are concerned with the attendance, they may call upon students one by one, which causes more of time to be wasted. Furthermore, if the attendance sheet is missing, the attendance needs to be re-signed, but theoretically, the data is not acceptable.

Because of this, the UiTM Smart Attendance Access Control System is proposed as a system to record student class attendance automatically. The system is based on Radio Frequency Identification (RFID) technology. It records the attendance of students using a RFID reader. To sign in, the students need to place their ID card on the reader. The system automatically identifies and records the student attendance in the class when the ID card is tapped. The system automatically provides analysis for the attendance of the students, and indicates whether they are eligible to sit for examinations (if the accumulated attendance meets the minimum 80% requirement for UiTM). The system will help and ease the lecturers' job in recording and calculating the students' attendance with minimum error. This system has computerized the tradition way of taking attendance. Another purpose for developing this software is to generate the report automatically at the end of the session or in the between of the session.

1.2 OBJECTIVES OF THE PROJECT

The main objective of the project is to help and ease lecturers to record and calculate the percentage of attendance by providing a low-cost electronic attendance recording system. To achieve this, the project objectives are broken down as follows:

1. To develop a RFID receiver circuit to record student attendance based on their matrix card.

2. To develop a Graphical User Interface (GUI) for the attendance system, and interface it with the RFID circuit. The GUI must provide the interface for the students to record their attendance, as well as provide analysis for the attendance records to determine whether the student is eligible to sit for the examination.

3. To develop the database for storage of the data.

1.3 SCOPES OF THE PROJECT

The scope of the project is as follows:

1. The RFID receiver circuit will interface with the student card based on the Wiegand protocol.
2. The GUI will be developed using Visual Basic 6.0.
3. The RFID receiver circuit will be interfaced with the GUI using the RS232 serial protocol.
4. The database will be created using the Microsoft Access DBMS.
5. The system will be divided into two parts: attendance recording (accessible to students) and analysis (accessible to lecturer only).

1.4 ORGANIZATION OF THE THESIS

The structure of the thesis was carefully planned to give a clear explanation on the overall of the project. The thesis is divided into six chapters.

Chapter 1: This chapter starts with the introduction or outline of the system which gives background information about the subject area. It also consists of the problem statement, objectives and the scope of this work.

Chapter 2: Describes previous works that have been done in this research area, as well as some literature background of several protocols, hardware and software used in this system.

Chapter 3: Describes an overall method that is applied in this project. This also includes elaboration on both the hardware and software design with the aid of schematic diagrams and flow charts.

Chapter 4: This part explains the project's results and discussion.

Chapter 5: Conclusion and future development of this system.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter describes previous works done in this area, as well as a review of the development tools and the protocols used in the system. Section 2.2 and Section 2.3 presents literature on RFID technology, as well as several automatic attendance system found in literature. Section 2.4 presents some literature on the PIC microcontroller, while the Wiegand and RS232 protocols used in this project are presented in Section 2.5.

2.2 RADIO FREQUENCY IDENTIFICATION (RFID)

Radio Frequency Identification (RFID) is a technology that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency (RF) portion of the electromagnetic spectrum to uniquely identify an object, animal, or person [1].

A RFID system consists of three components: an antenna and transceiver (often combined into one reader) and a transponder (the tag). During operation, the transceiver scans for nearby RFID transponders. If a transponder is detected, the transceiver then interrogates it for the data it contains. The data can then be passed on to other hardware/software to be processed.

2.3 REVIEWS ON EXISTING ATTENDANCE SYSTEMS

2.3.1 RFID – BASED ATTENDANCE SYSTEM

In [1], a portable RFID based attendance system for education was presented. This system allowed update of microcontroller by integrating the In-Circuit Serial Programming™ (ICSP™) pins and serial programmer from time to time. The real time clock is included in the system whereby if the system is powered off, the time is still running. The system used HyperTerminal software to view the recorded attendance. The performance of the RFID based attendance system was evaluated on a different tag positions and distance. The test proved that the reader has about 5cm detection range if scan from top position.

2.3.2 ISAMS: TRACKING STUDENT ATTENDANCE USING INTERACTIVE STUDENT ATTENDANCE MANAGEMENT SYSTEM

A framework for the future implementation of a student attendance system was presented in [2]. The system uses a barcode scanner to record the students' attendance automatically, and counts the number of attendance. The Charge-Coupled Device (CCD) scanner was able to read barcodes quickly and easily, but had limitations in terms of distance and dimensions of the barcode.

2.3.3 AUTOMATED RECORDING OF TIME AND ATTENDANCE USING RFID

In [3], a RFID based attendance system for university environment was presented. The system was developed using the bar code. The students' attendance weight was also calculated in order to confirm their eligibility to sit in examinations

[3]. However, the system did not generate the student attendance percentage for analysis.

2.4 THE PIC MICROCONTROLLER

A microcontroller is a highly integrated chip that contains all components comprising a controller. Microcontroller is used in automatically controlled products and devices. PIC microcontrollers are popular processors developed by Microchip Technology with built-in RAM, memory, internal bus, and peripherals that can be used for many applications. PICs are popular due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and re-programming with flash memory) capability.

The Harvard architecture is used in PIC which instructions and data come from conveniently separate sources. It simplifies timing and microcircuit design greatly, and this pays benefits in areas like clock speed, price, and power consumption. Figure 2.1 shows the block diagram of Harvard architecture of the microcontroller. Harvard architecture has separate data and instruction busses, allowing transfers to be performed simultaneously on both busses. It is possible to have two separate memory systems for Harvard architecture. As long as data and instructions can be fed in at the same time, then it doesn't matter whether it comes from a cache or memory. At higher clock speeds, caches are useful as the memory speed is proportionally slower. Harvard architectures tend to be targeted at higher performance systems, and so caches are nearly always used in such systems.

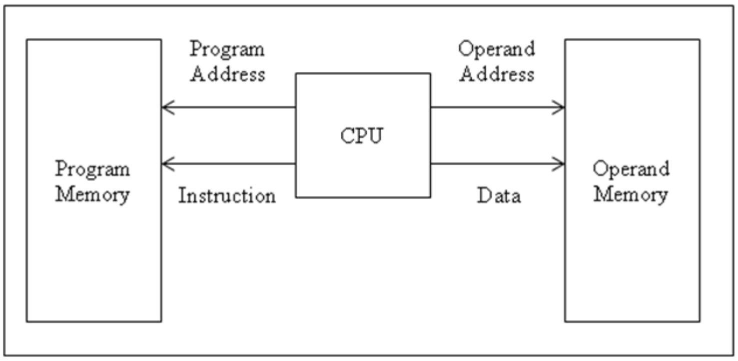


Figure 2.1: Harvard Architecture

Each microcontroller family has its own set of instructions, which carry out essentially the same set of operations, but using different syntax.

2.5 INTERFACING PROTOCOLS

2.5.1 WIEGAND PROTOCOL

The Wiegand interface is the *de facto* industry standard protocol used for security data encoding and interfacing for card readers (smart card, proximity cards) to control panels [12]. The industry-standard Wiegand protocol is encoded into 26-bit binary format with a specific arrangement of binary data [12]. There are other data formats like the 34-bit & 37-bit format which use the same signalling standard as that of Wiegand but have different data formatting standard.

As shown in Figure 2.2, the Wiegand format contains 8 bits for the facility code field and 16 bits for the ID number field. Mathematically these 8 facility code bits allow for a total of just 256 (0 to 255) facility codes, while the 16 ID number bits allow for a total of only 65,536 (0 to 65,535) individual ID’s within each facility code [12]. Due to the mathematical limitations of the 26-bit Wiegand format, code duplication might occur. Table 2.1 provides a summary the 26-bit Wiegand format.

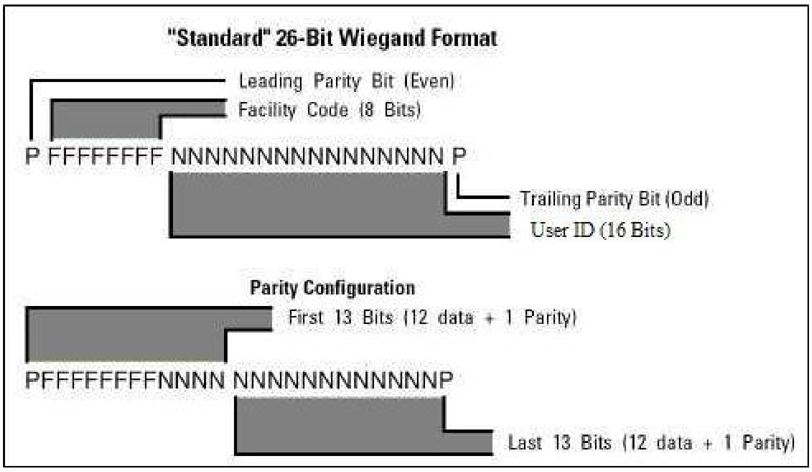


Figure 2.2: 26-bits Wiegand Format

Table 2.1: Summary of the 26-bits Wiegand Format

Bit Number	Purpose
Bit 1	Even parity over bits 2 to 13
Bits 2 to 9	Facility code (0 to 255); Bit 2 is MSB
Bits 10 to 25	ID Number (0 to 65,535); Bit 10 is MSB
Bit 26	Odd parity over bits 14 to 25

2.5.2 RS232 SERIAL PROTOCOL

RS232 stands for "recommended standard 232", a traditional method for transferring information between a computer and its peripherals (Figure 2.9). This cable is most commonly used in order to successfully transfer data from one technological device to another. The RS232 standard has been largely replaced by other connections, like USB cables. These new cables are faster, more reliable and use less energy than the traditional standard, but they are digital and require the data to be sent in packets and then unpacked. In this project, the SK40B has a built in RS232 protocol through a DB9 connector as shown in Figure 2.9.

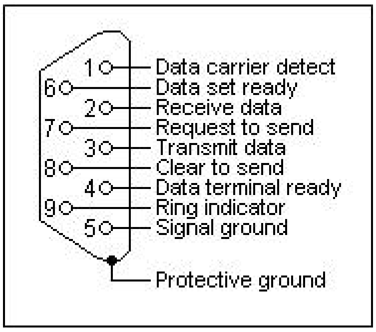


Figure 2.3: RS232 DB9 Pinout

Table 2.2: Serial (RS232) Port Interface Pinout and Signals

Pin Number	Acronym	Full Name	Direction	Mean
1	DCD	Data Carrier Detect	←	Used by the Data Set to indicate to the Data Terminal that the Data Set has detected a carrier
2	RXD	Receive Data	←	The data sent from the Data Set and received by the Data Terminal
3	TXD	Transmit Data	→	The data sent from the Data Terminal and received by the Data Set
4	DTR	Data Terminal Ready	→	Used by the Data Terminal to signal to the Data Set that it is ready for operation, active high
5	SG	Signal Ground		Common return for all signals on the interface
6	DSR	Data Set Ready	←	Used by the Data Set to signal to the Data Terminal that it is ready for operation and ready to receive data, active high
7	RTS	Request To Send	→	Used by the Data Terminal to signal the Data Set that it may begin sending data. The Data Set will not send out data without this signal, active high
8	CTS	Clear To Send	←	Used by the Data Set to signal the Data Terminal that it may begin sending data. The Data Terminal will not send out data without this signal, active high

9	RI	Ring Indicator	←	Used by the Data Set to indicate to the Data Terminal that a ringing condition has been detected

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter describes the methodology in designing the UiTM Smart Attendance Access Control System. This chapter first describes the development tools used (Section 3.2), followed by important aspects of hardware design (Section 3.3) and software design (Section 3.4).

3.2 DEVELOPMENT TOOLS

This section describes the tools / software / components used to develop the system. Several software were used to develop the system, namely Visual Basic 6.0 (for Graphical User Interface design) (Section 3.2.1), the Microsoft Access DBMS (for database design) (Section 3.2.2). On the hardware part, the PIC16F877A microcontroller, SK40B PIC Programmer Board, UIC00A USB ICSP PIC Programmer and RFID Reader and Tag are described in Section 3.2.3 to Section 3.2.7.

3.2.1 VISUAL BASIC 6.0

Visual Basic 6.0 (VB) is an event-driven programming language and associated development environment from Microsoft. A VB6 program is made up of many subprograms, each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another depending on the objective. It also provides the easier programming and user friendly function because this program can visualize the arrangement of components or controls on a form, specifying attributes and actions of those components. That can be used to enhance the project. Furthermore, it enables development of graphical user interface (GUI) applications that can access to databases by using Microsoft Access Database Management System (DBMS).

3.2.2 MICROSOFT ACCESS

Microsoft Office Access is a DBMS product by Microsoft to create simple database solutions for simple data entry and viewing. Access tables support a variety of standard field types. The database can be divided into a front end database that contains the application forms and is linked to tables stored in a back end shared database containing the data. This application is Visual Basic compatible, and Visual Basic has built-in components to access databases created using Microsoft Access.

3.2.3 THE PIC16F877A MICROCONTROLLER

The PIC16F877A microcontroller, as shown in Figure 3.1, is the largest chip of the 16F87x family, which is a small piece of semiconductor integrated circuits. A package type of these integrated circuits is Dual Inline Package (DIP) stand for Dual Inline Package for semiconductor IC. This package is very easy to be soldered onto the strip board. However using a DIP socket is much easier so that this chip can be plugged and removed from the development board.

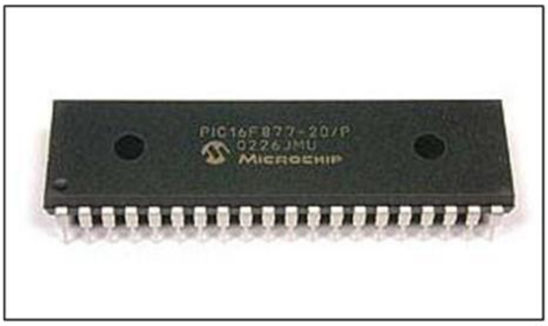


Figure 3.1: PIC16F877A Microcontroller

This microcontroller is a 40-pins device. It features 819214 flash program memory, 368 bytes of RAM, 256 bytes of EEPROM data memory, 200 ns instruction execution, 33 I/O pins, 8 channels of 10-bit Analogue-to-Digital (A/D) converter, 2 capture/compare/PWM functions, three timers, an analogue capture and comparator circuit, Universal Asynchronous Receiver Transmitter (USART), and internal and external interrupt facilities [11]. This IC can be reprogrammed and erased up to 10,000 times.

In this project, the source code used is in assembly language. The program is downloaded into the PIC using the UIC00A programmer via the MPLAB IDE software. Figure 3.2 shows the pin diagram of the PIC16F877A.

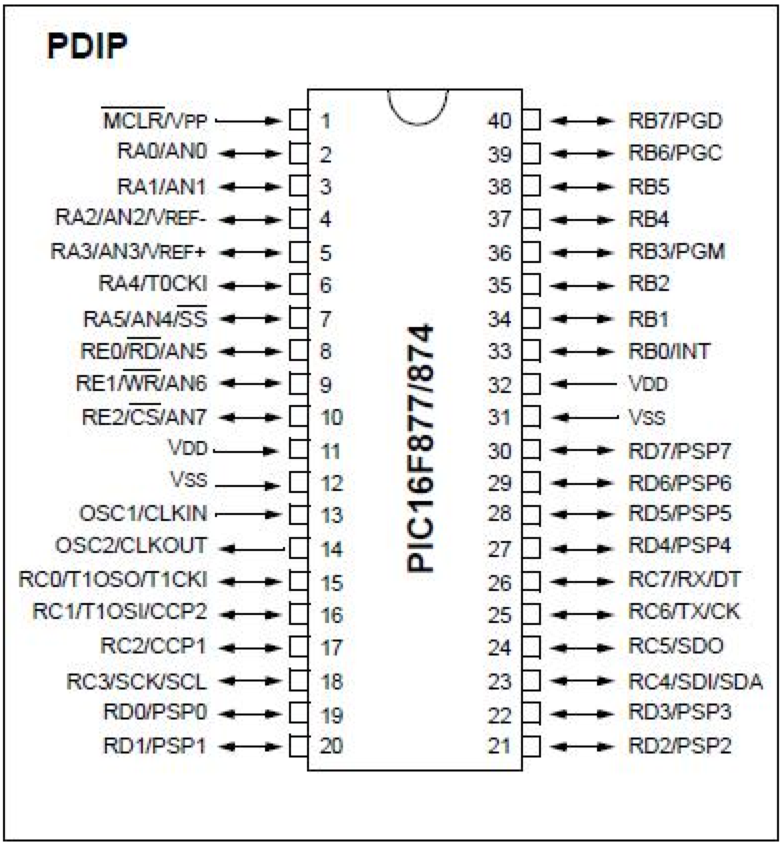


Figure 3.2: PIC16F877A Pin Diagram

3.2.4 THE SK40B PIC PROGRAMMER BOARD

The SK40B PIC programmer board is designed to offer a start up platform for development. However, all interface and program has to be developed by the user. It offer plug and use features. It is perfectly fit for 40 pins 16F and PIC18F PIC.

SK40B is designed to offer an industrial grade PCB. It is compact, powerful, flexible and robust for a start-up platform which is suitable for starters, hobbyists and also experts. The board saves a lot of development and soldering time at the hardware developer side. It does not require extra components for the PIC to function and all 33 I/O pins are nicely labelled to avoid miss-connection by users.

It also comes with a connector for UIC00A which is low cost USB ICSP PIC Programmer and an RS232 (Serial) hardware port on board allowing alternative method to load program via boot loader. With UIC00A, a program can be loaded in less than 5 seconds.

Users are able to utilize the function of PIC by directly plugging in the I/O components in whatever way that is convenient to user. This kit comes without the PIC microcontroller to provide the freedom for user to choose PIC type. As a result, PIC16F877A is chosen as the microcontroller for this project. Figure 3.3 shows the board layout of the SK40B PIC programmer board while Table 3.1 shows the labelled board layout of the board.

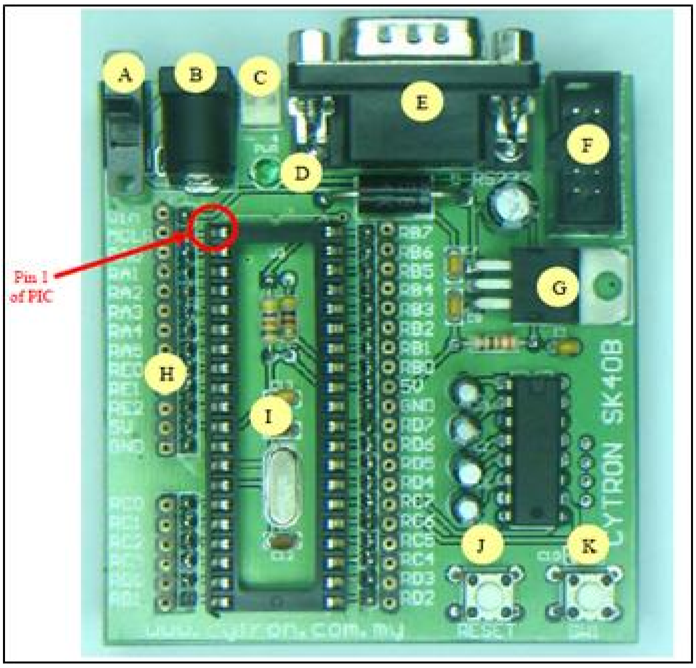


Figure 3.3: Components on SK40B

Table 3.1: Labelled Board Layout of the SK40B PIC Programmer Board.

Label	Function	Label	Function
A	Slide switch for main power supply	G	5V regulator
B	DC power adaptor socket	H	header pin and turn pin
C	Battery connector	I	40 pins IC socket for PIC MCU
D	Power indicator LED	J	Reset button
E	Serial cable adaptor (female DB9)	K	Programmable push button
F	Connector for UIC 00A Programmer		

3.2.5 THE UIC00A USB ICSP PIC PROGRAMMER

The UIC00A offers low cost yet reliable and user friendly PIC USB programmer solution designed to program popular Flash PIC MCU PIC16F family [10]. It can also program 16bit PIC MCU. On board ICSP^{TM} (In-Circuit Serial Programming) connector offers flexible method to load program. It supports on board programming which eliminate the frustration of plug-in and plug-out of PIC MCU.

This is also allows to quickly program and debug the source code while the target PIC is on the development board.

UIC00A is designed to be plugged and played with USB connection since USB port is commonly available and widely used on computer. This programmer obtained it power directly from USB connection making it a truly portable programmer. It is ideal for field and general usage. UIC00A offers reliable, high speed programming and free windows interface software. Figure 3.4 shows the UIC00A USB ICSP PIC Programmer.

Figure 3.4: UIC00A USB ICSP PIC Programmer

It is designed with capabilities and features of:

i. Industrial grade PCB with surface mount component to offer small size yet reliable as a quality product.
ii. USB Plug and Play function.
iii. An IDC cable is included for external on-board programming.
iv. Auto load program capability.
v. Optional external power to target PIC should be +5V.

vi. No external power required for UIC00A to function.

vii. Optional socket (UIC-S) to program 18 pins, 28 pins and 40 pins microcontroller.

3.2.6 RFID READER

An RFID reader, as show in Figure 3.5, is a device that is used to interrogate an RFID tag. The reader has an antenna that emits radio waves; the tag responds by sending back its data.

Figure 3.5: RFID Reader

A number of factors can affect the distance at which a tag can be read (the read range). The frequency used for identification, the antenna gain, the orientation and polarization of the reader antenna and the transponder antenna, as well as the placement of the tag on the object to be identified will all have an impact on the RFID system's read range.

3.2.7 RFID TAG

An RFID tag, as shown in Figure 3.6, is a microchip combined with an antenna in a compact package; the packaging is structured to allow the RFID tag to be attached to an object to be tracked. The tag's antenna picks up signals from an RFID reader and then returns the signal, usually with some additional data (like a unique serial number or other customized information).

Figure 3.6: RFID Tag

3.3 HARDWARE DEVELOPMENT

3.3.1 BLOCK DIAGRAM

Figure 3.7 shows the block diagram of this project. There are three main parts which are inputs, processor and display. The PIC16F877A microcontroller receives, reads and interprets the information or input from RFID reader and uses it shows the output via graphical user interface and database by using RS232 serial connector. MAX232 amplifies the logic level of the circuit from 5 volts to 12 volts. The function of LCD is to display instruction and present operation condition in the form of messages that displays ASCII characters.

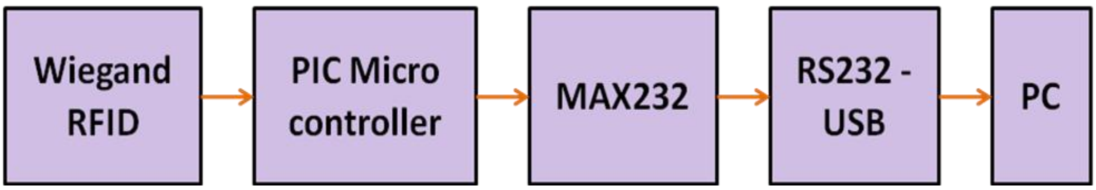


Figure 3.7: Block Diagram of System

As a passive tag and passive reader, the tag can only transmit the ID as it is in close proximity to the reader (about 1cm to 3cm). Overlapping may cause current to be shared by the overlapping cards. Hence retransmission and delay will cause the Wiegand protocol to be at fault.

As the reader receives the retransmitted signal by the tag, it will receive 26-bits that carried together its ID somewhere in between. Out of the 26-bits that is received and accepted by the PIC, only 6-bits will be compared via the pre-programmed data in the PIC. The PIC will then display the names of the holder of the tag on the LCD.

The same data will be sent from the PIC pins to MAX232 chip on the SK40B to amplify the 5 volts from the PIC to 12 volts as receives by the serial RS232 protocol to be sent to the computer. This data will then be received via the communication port to be interpreted by the Visual Basic 6.0.

The system accepts from two different levels of accesses which are Lecturer and Student. Student is allowed for class registration access only while lecturer may access to view the attendance record as well as the students' profile and the percentage of the attendance. The lecturer need to key in the matrix number of a student before proceeding to the viewing session where the students' information is obtained from the database. The number of percentage will be coloured in black if the accumulated attendance meets the 80% minimum requirements. Otherwise, the percentage will be in red colour showing that the student has been barred for examination.

3.3.2 SCHEMATIC DIAGRAM

Figure 3.8 shows the schematic diagram of the system. The hardware used is the Wiegand protocol RFID reader to communicate with PIC16F877A to identify the user's identity from the RFID tag and the PIC16F877A microcontroller for the signal processing. The MAX232 chip used to pull-up to 12 volts and pull-down to -12 volts to provide a serial communication to the PC (Personal Computer) using the RS232 protocol from the microcontroller pins.

The data received from the MAX232 will then be analyzed by the Visual Basic GUI to register and store the student's attendance and calculate the student's attendance percentage.

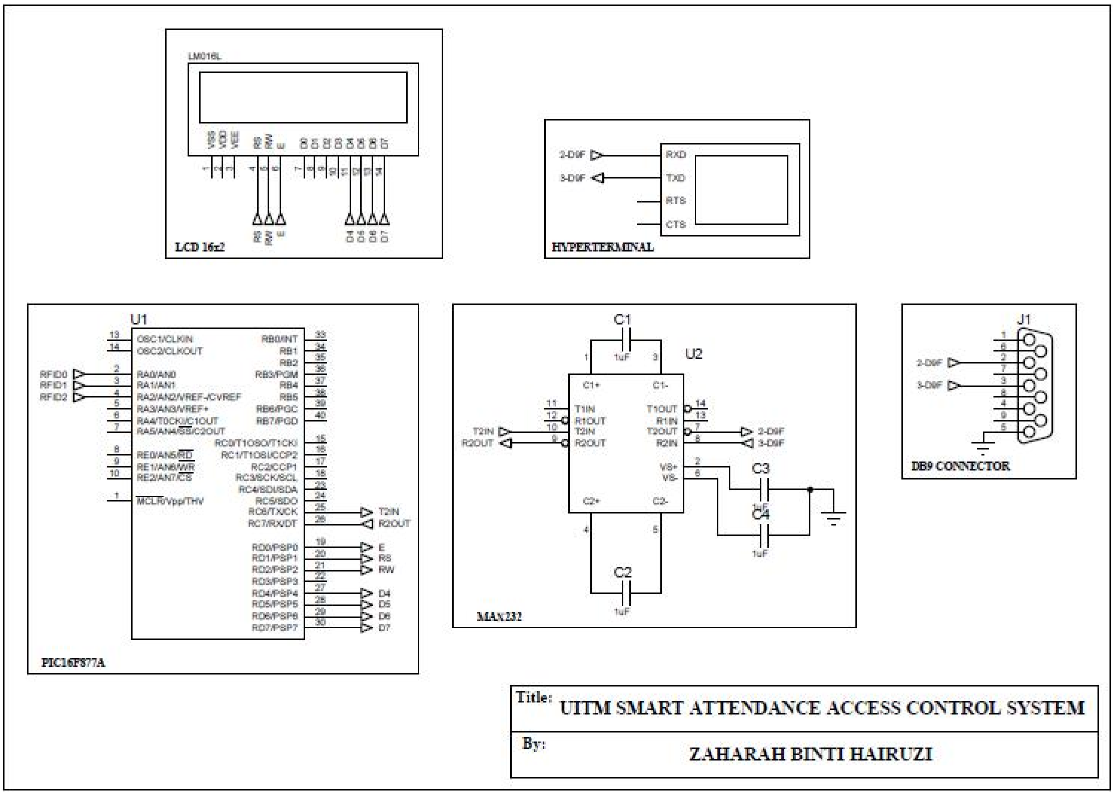


Figure 3.8: Schematic Diagram in PROTEUS ISIS

For communication with the microcontroller the Wiegand interface uses two wires for carrying the card data to the controller these wires are called as DATA0 and DATA1. Normally both these lines are high i.e. when no data is being sent. A '0' is sent by making DATA0 line LOW & DATA1 line HIGH. Whereas a '1' is sent by making DATA1 line LOW & keeping DATA0 line HIGH. This signal is at TTL level and not an open collector signal so can be directly connected to the Microcontroller. A typical pulse width is 50 μs with an inter-spacing of 1ms but the actual timing values and output circuitry (open collector/TTL) are determined by the card reader manufacturer.

3.4 SOFTWARE DEVELOPMENT

This section begins with a description of the program flowchart, followed by the design steps to develop the software and the database. There were several steps involved in software development stage:

i. Coding microcontroller in CCS C compiler
ii. Compiling microcontroller in MPLAB IDE
iii. Simulation microcontroller in PROTEUS ISIS
iv. Designing interface in VB6.
v. Database design.

These steps are described in detail in Section 3.4.2 to Section 3.4.6.

3.4.1 SOFTWARE FLOWCHART

Figure 3.9 shows the flowchart of PIC16F877A that has been programmed in C programming. According to the flowchart, when a student tags his/her card on the RFID reader, the microcontroller verifies the tag number by comparing it with predefined tag ID numbers already stored in the microcontroller ROM. The reader emits a sound when the card is successfully read. After the card has been successfully read, the RFID reader establishes a communication with the RS232 cable in order to send the student card identification to the database system.

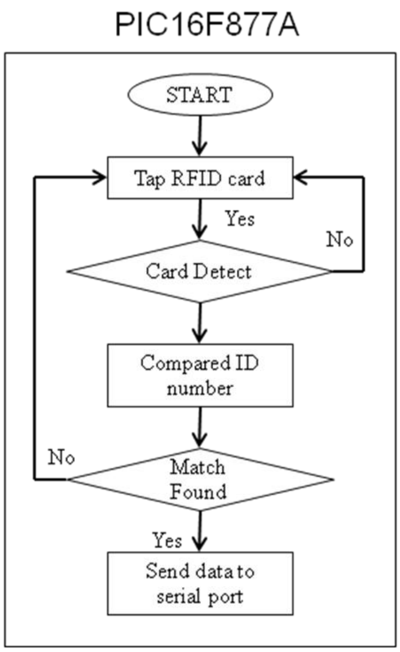


Figure 3.9: Flowchart used as a Guideline in C Programming

3.4.2 CODING IN CCS C COMPILER

CCS (Customer Computer Services) provides a complete, integrated tool suite for developing and debugging embedded applications running on Microchip PIC. The coding was performed in the CCS C compiler. The programming being done in the MPLAB IDE (Micro Processor LAB Integrated Development Environment) simultaneously since the CCS C compiler was embedded earlier in the MPLAB.

3.4.3 COMPILING IN MPLAB IDE

MPLAB IDE is a Windows based Integrated Development Environment (IDE) for the Microchip Technology Incorporated PICmicro microcontroller (MCU) families. MPLAB IDE allows writing, debugging, and optimizing the PICmicro MUC applications for firmware product designs. MPLAB IDE includes a text editor, simulator, and project manager. MPLAB IDE is also other Microchip or third party development system tools. MPLAB IDE runs as a 32-bit application on MS Window, is easy to use and includes a host of free software components for fast application development and super-charged debugging.

After the programming was done, the code was compiled and converted into hex files. The hex file was then burned and executed by the PIC16F877A program memory either physically or by simulation. As shown in Figure 3.2, the space of memory and disassembly files can be observe for a progress check up during the programming in MPLAB.

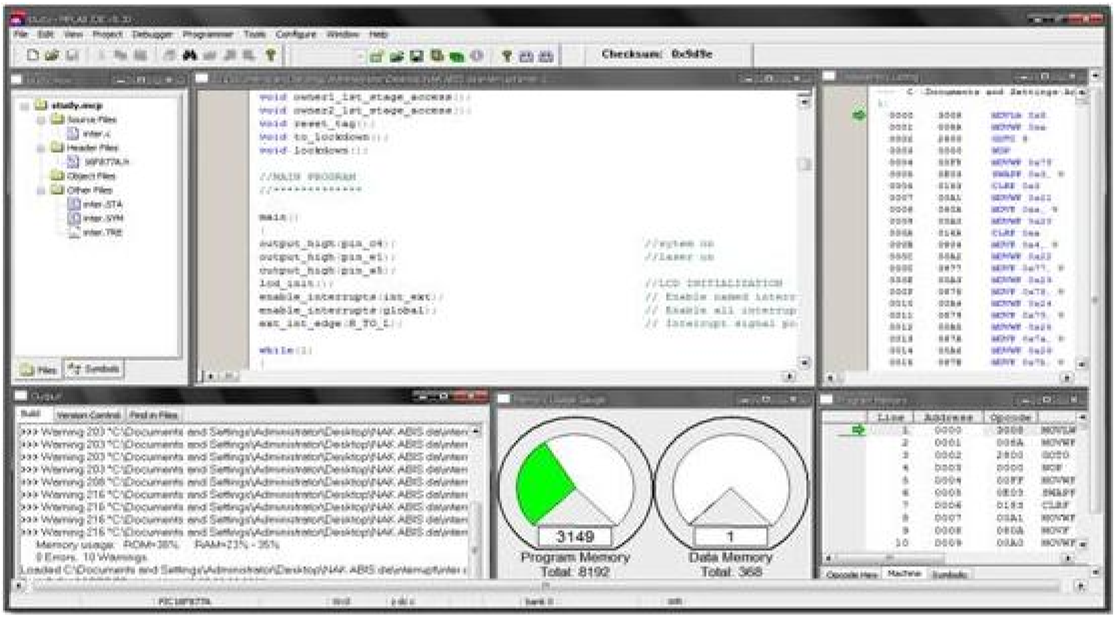


Figure 3.10: The C Programming in the MPLAB IDE Software

3.4.4 SIMULATION IN PROTEUS ISIS

The system is designed in PROTEUS ISIS after compiled the programmed C language. Various inputs can be tested and output can be observed. The mechanism of simulation is done and tested as in real time as shown in Figure 3.11. The purpose of the simulation is only for testing of the C programming before the actual hardware's development. This compiler takes a source text file and converts it to machine code.

The disassembled C language will then be placed in the simulation of the PIC16F877A.

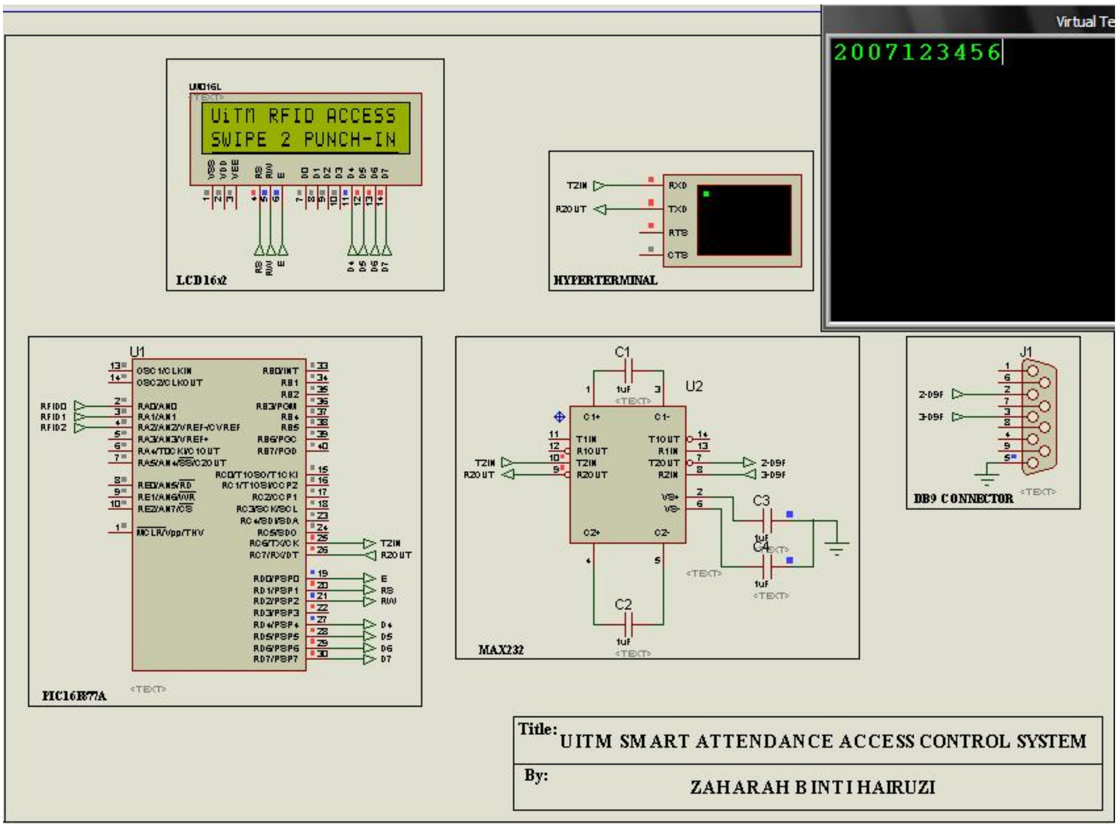


Figure 3.11: Hardware and Software Simulation in PROTEUS ISIS

3.4.5 DESIGN INTERFACE IN VISUAL BASIC 6.0

Figure 3.12 shows the flowchart of the GUI that designed in Visual Basic 6.0. There are six basic forms for this project.

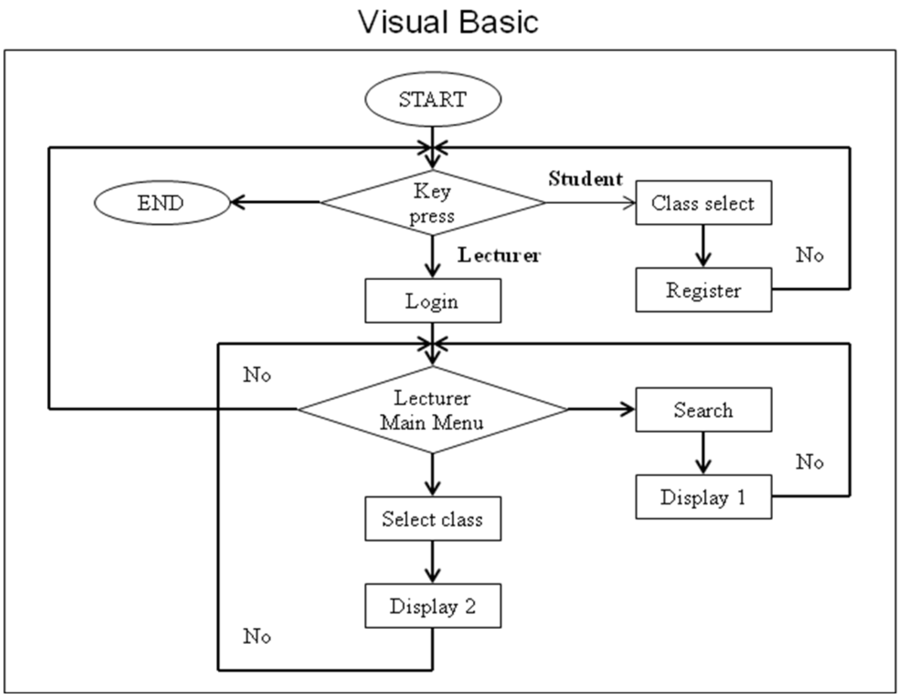


Figure 3.12: Flowchart of GUI Design

Figure 3.13 shows the home page of this system. It consists of three buttons namely the class registration button, student attendance record button and the exit button. The class registration button will access to class attendance page which is shown in Figure 3.14. Students can access when they click this button. The attendance records button can only be accessed by the lecturers where the lecturer’s username and password is needed which is shown in Figure 3.15. The “EXIT” button will exit the user from the whole system.

Figure 3.13: Snapshot of Home page

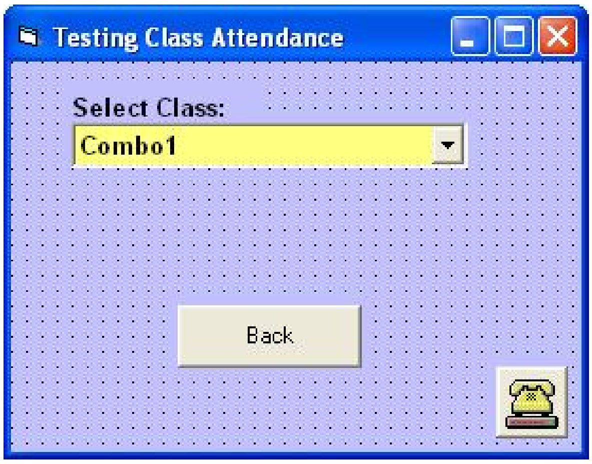


Figure 3.14: Snapshot of Class Attendance page

According to Figure 3.14, the students need to select a class that they are attending before tapping their ID card. They need to click the "Back" button after a message box displayed that they are registered to allow other students register for the class.

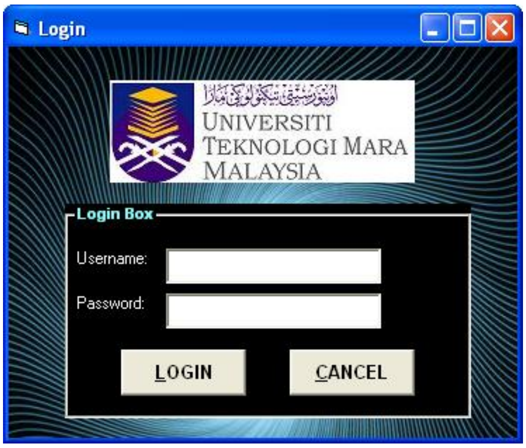


Figure 3.15: Snapshot of Login page

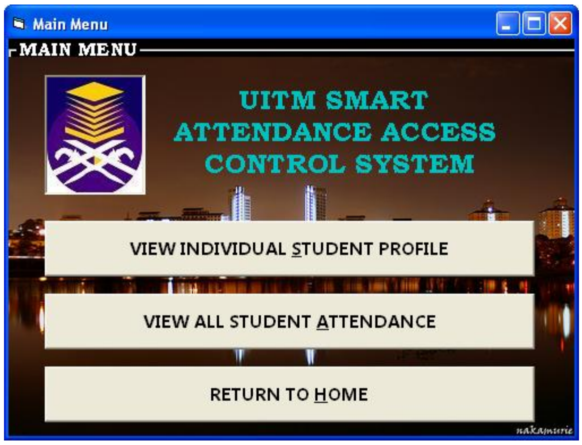


Figure 3.16: Snapshot of Main Menu page

Figure 3.15 shows the snapshot of Login page. This is where lecturers need to login before entering to the Main Menu page to access to view their students attendance record as shown in Figure 3.16. The lecturers need to key in their username and password in the empty boxes and clicked "LOGIN". The "CANCEL" button will clear the username and the password in the text box.

Figure 3.16 shows the snapshot of Main Menu page. There are three main buttons where the "STUDENT PROFILE" button will access to view students profile page as shown in Figure 3.17, "STUDENT ATTENDANCE" button will access to view the students attendance records as shown in Figure 3.18 and the "RETURN TO HOME" button will return to Home page as shown in Figure 3.13.

Figure 3.17 is the snapshot of the Student Profile page. The system will show all the information about students including the percentage of the attendance and the status of the students. In this page, the "SEARCH" button is for searching the students profile by entering the students' matrix number and the "UPDATE" button purpose is for updating the database if there is any information needs to be updated. The "BACK" button will return the user to the Main Menu page.

Figure 3.18 shows the Student Attendance List page. The select class will show the date and time of classes. The "VIEW" button will list down the students' name in the student attendance list. The "BACK" button will exit from this page.

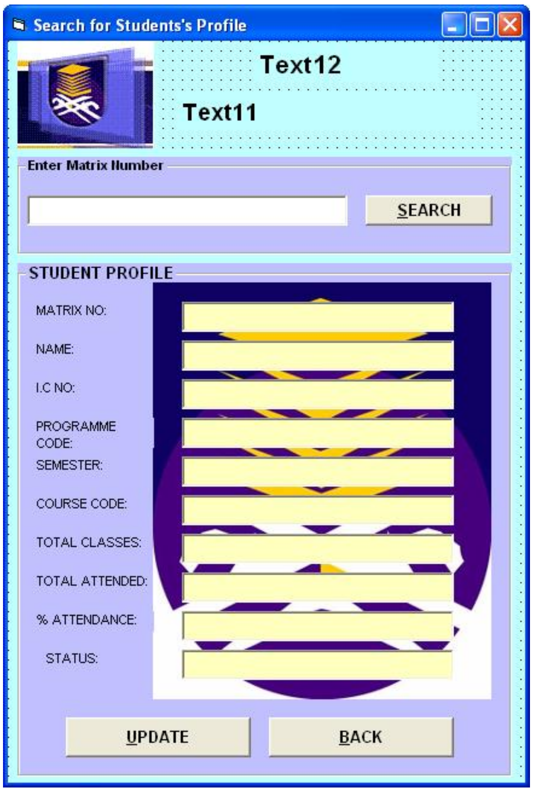


Figure 3.17: Snapshot of Student Profile Page

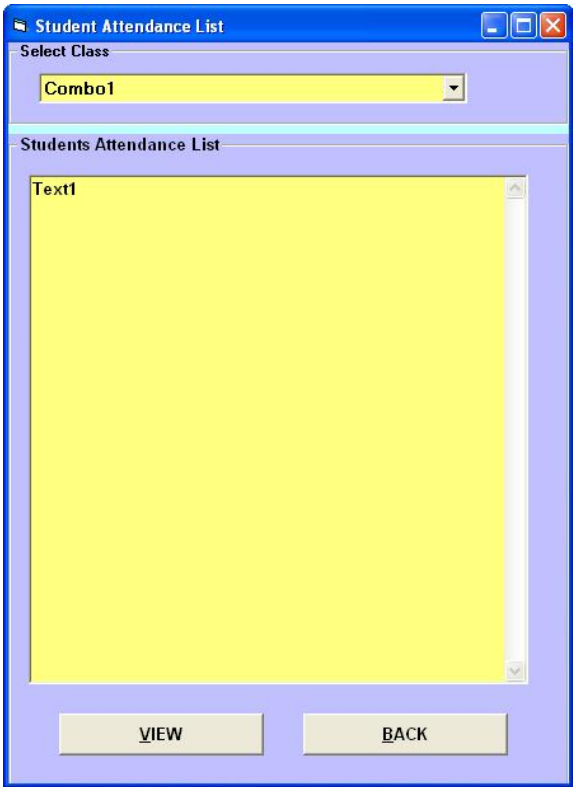


Figure 3.18: Snapshot of Students Attendance List Page

3.4.6 DATABASE DESIGN

The Entity Relationship Diagram (ERD) was used to design the database. Two entities were defined, which were STUDENT and CLASS. The STUDENT entity stores the following attributes:

1. STUD_ID: Primary key defined for student table.
2. STUD_MATRIX: Matrix card number. Unique identifier for student.
3. STUD_NAME: Student name.
4. STUD_IC: Student identity card number. Unique identifier for student.
5. STUD_PCODE: Program code.
6. STUD_SEMESTER: Semester enrolled.
7. STUD_CCODE: Course code.
8. STUD_ATTENDED: How many classes attended by the student.

9. STUD_SHOULDATTEND: How many classes should have been attended by the students.

The CLASS entity stores the following attributes:

1. CLASS_ID: Primary key for table CLASS.
2. CLASS_DAY: Date when the class is held.
3. CLASS_CODE: Subject code.

Since the relationship between the CLASS and STUDENT is many-to-many (M: N), the relationship needed to be decomposed into two one-to-many relationships (1: M). This resulted in the table MASUK, which links records in STUDENT and CLASS using primary keys obtained from both entities. The table MASUK has the attributes:

1. ID: Primary key for table MASUK.
2. CLASS_ID: Class ID number.
3. STUD_MATRIX: Student matrix card.

The resulting Entity-Relationship Diagram (ERD) is shown in Figure 3.19. Based on the ERD, three tables were defined, with the abovementioned attributes.

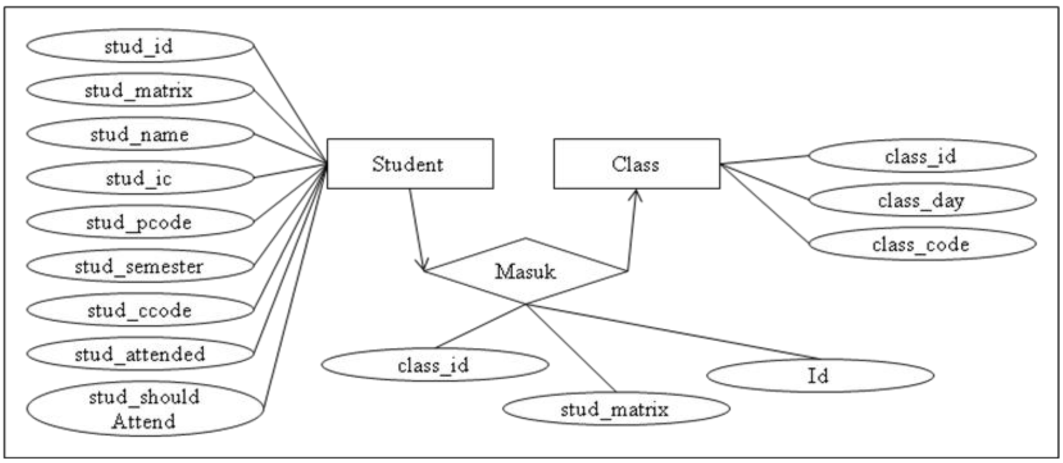


Figure 3.19: Entity Relationship Diagram of Student Attendance System

CHAPTER 4

RESULTS AND DSCUSSIONS

4.1 INTRODUCTION

This chapter describes the results and discussion on the UiTM Smart Attendance Access Control System. This chapter first describes the hardware construction (Section 4.2) and the software system (Section 4.3), followed by database system (Section 4.4) and limitations of the system (Section 4.5).

4.2 HARDWARE CONSTRUCTION

Figure 4.1 shows the hardware that has been constructed involving RFID reader, PIC16F877A, SK40B Programmer board, RS232 serial port and LCD.

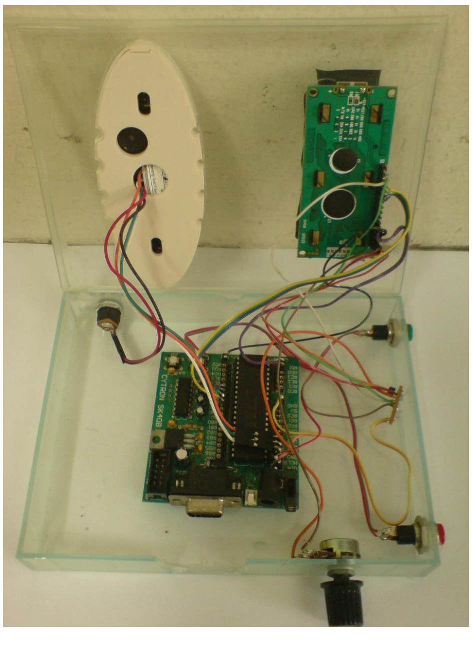


Figure 4.1: Hardware Construction

As the system starts, the LCD will display "UiTM RFID ACCESS SWIPE 2 PUNCH-IN" (as shown in Figure 4.2), this will tell that the system is stable and ready to run. The software design for the RFID reader was made such that it waits for the user to tap/swipe their card unto the RFID reader. Hence no action will be done at this moment.

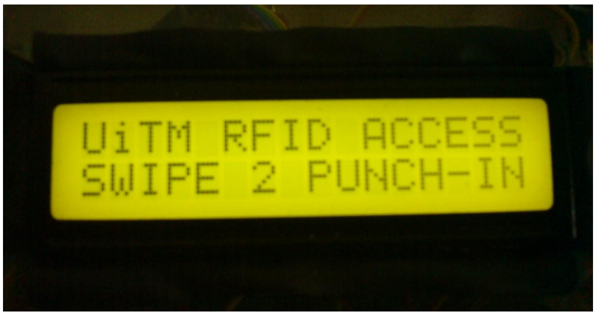


Figure 4.2: Initial LCD Display

As the user swipe the RFID tag (as shown in Figure 4.3) the LCD will display the name of the owner to indicate that the user has already registered the name to the system.

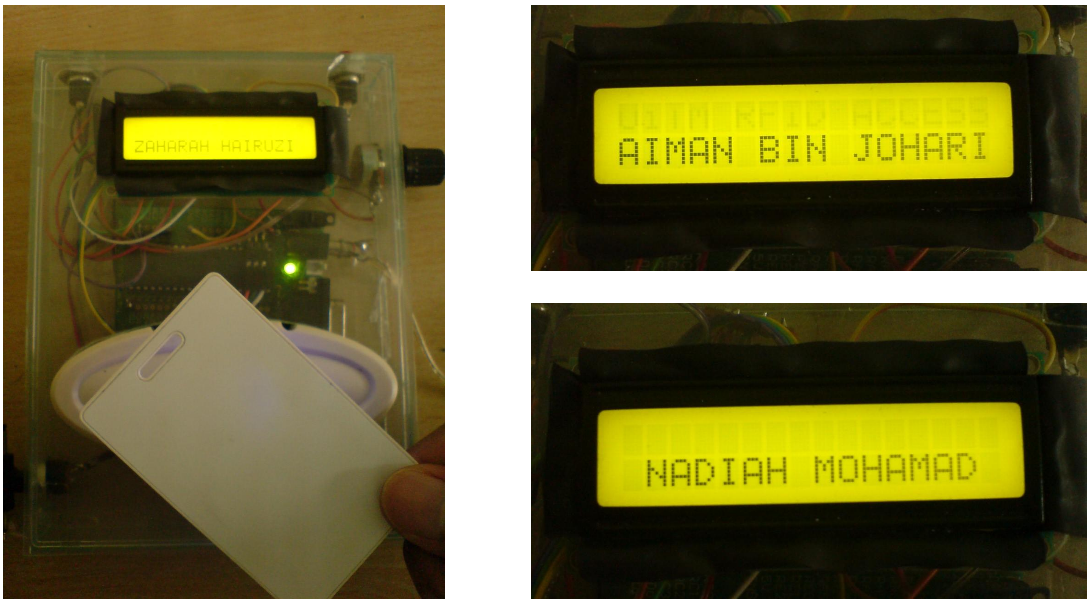


Figure 4.3: LCD displaying students' name when the RFID is tapped

Figure 4.4 shows the display that the students already swipe their ID cards.

Figure 4.4: Display in between Swipes

After the swipe, the hardware (PIC16F877A) will analyze (through software) for the user identity that will later on passed onto the serial communication between the hardware and the PC (Personal Computer). The data will be received by the visual basic GUI and registered into the attendance table to be analyzed by the 60% attendance algorithm programmed in Visual Basic.

The RFID system is using the 26 bit Wiegand protocol to be analyzed by the PIC. The PIC will only pick 8-bits as per the software design to be compared with the registered students' identity.

4.3 SOFTWARE SYSTEM

The purpose of the project is to record the students' attendance based on the IDs that they have been tapped and automatically calculates the percentage of accumulated attendance for UiTM students. This system accepts from two different levels of accesses which are Lecturer and Student.

Student is allowed for class registration access only while lecturer may access to view the attendance record as shown in Figure 4.5. The system was designed such that it requires lectures' username and password to avoid deception from student.

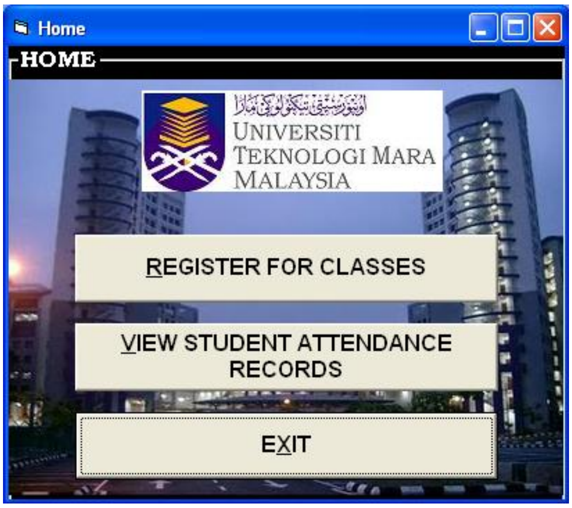


Figure 4.5: Snapshot of accessing for Lecturer and Student page

Figure 4.6 and Figure 4.7 shows the Class Attendance page. From the Figure 4.6, it shows the date and time for classes of specific subjects. Students need to select one of the classes as shown in Figure 4.7 before tapping their ID cards. The "BACK" button will return them to the previous page after the system shows the message box that they are registered and to allow other students to register for the class.

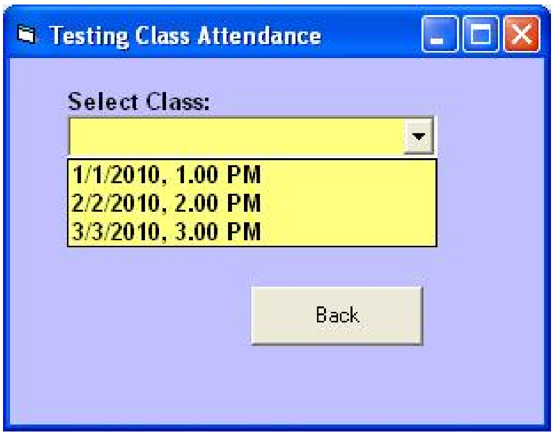


Figure 4.6: Snapshot of Class Selection

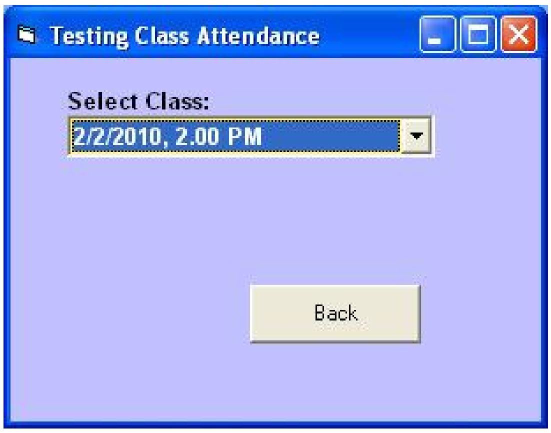
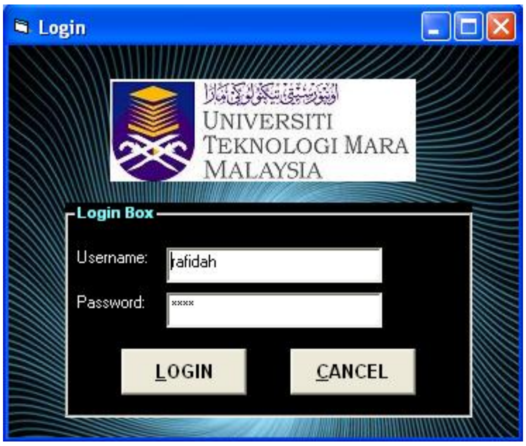


Figure 4.7: Snapshot of Selected Class

Figure 4.8 shows the username and password being keyed in by a lecturer in the Login page before proceeding to the next page as shown in Figure 4.9.



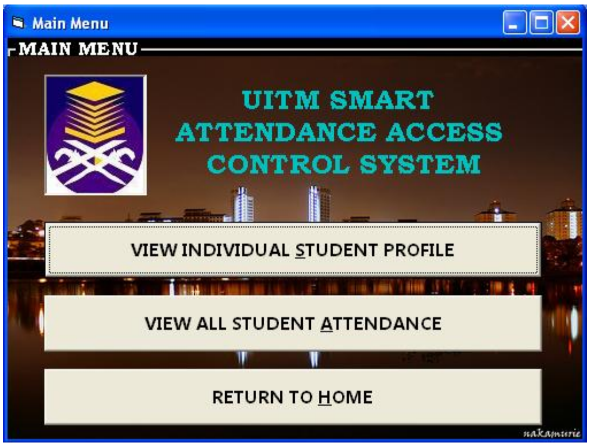


Figure 4.8: Snapshot of Running Login page

Figure 4.9: Snapshot of Main Menu page

Figure 4.9 shows Main Menu after lecturers logged in their usernames and passwords. The "Students Profile" button is linked to the page as shown in Figure 4.10 and Figure 4.11 while "Student Attendance" button will be linked to the form that displays list of students who attend to class on specific days as shown in Figure 4.12.

Figure 4.10 and Figure 4.11 shows the page where lecturers can view the student profiles after the lecturer key in the matrix number of a student. The system shows the percentage of the attendance for the student. The calculation of 80% attendance is what makes the project stands out from the previous research being done on attendance system.

The number of percentage will be coloured in black if the accumulated attendance meets the 80% minimum requirements. Otherwise, the percentage will be in red colour showing that the student has been barred for examination. In this form, the lecturers can edit the information and it is automatically updated in the database as the update button is clicked.

Figure 4.10 shows the status of this student is eligible to seat for examination of a subject where the system shows the percentage of the attendance is 80%. The student's name is coloured in black. Figure 4.11 shows the opposite status where this student is bar from seating in examination. The system shows the percentage of attendance is 40% and the name is coloured in red.

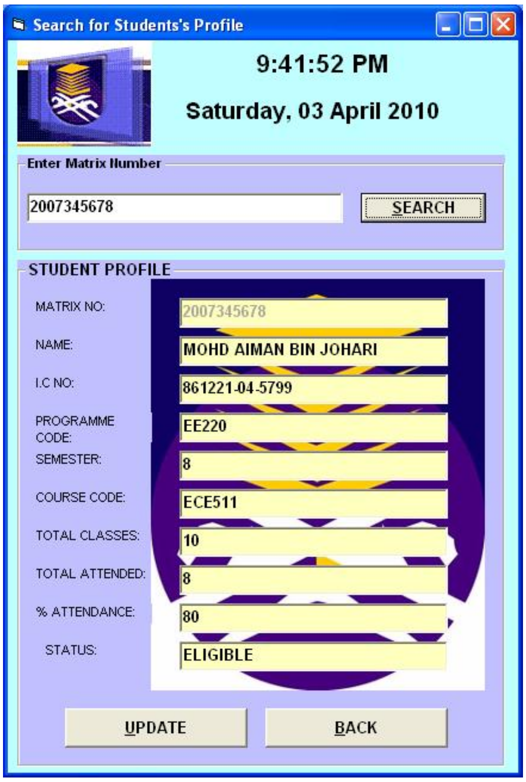


Figure 4.10: Snapshot of Eligible Status

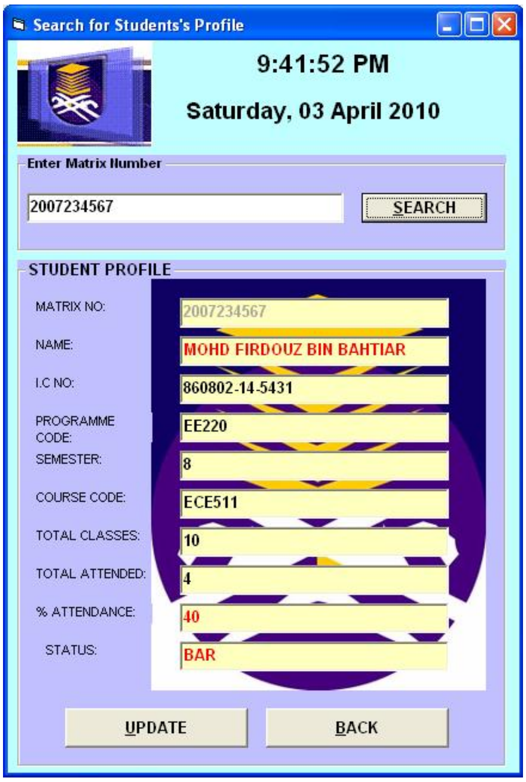


Figure 4.11: Snapshot of Bar Status

Figure 4.12 shows the Student Attendance List page. The list of students attended the classes can be viewed through the selection of the list of classes per subject in the combo box at the top of the form.

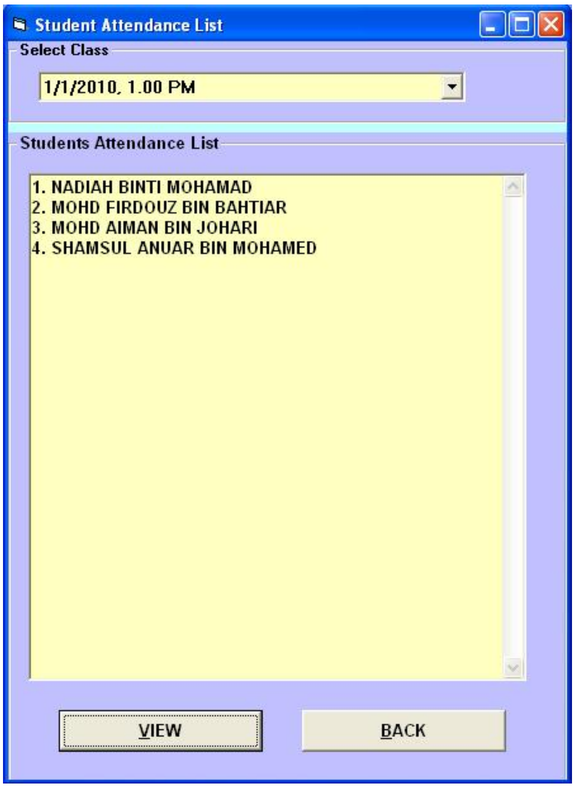


Figure 4.12: Snapshot of Student Attendance List page

4.4 DATABASE SYSTEM

There are three main tables for this system which are “Student” as shown in Figure 4.13, “Class” as shown in Figure 4.14 and “Masuk” as shown in Figure 4.15. Student’s table contains of students’ information where the information will be displayed via Student Profile page using Visual Basic. Lecturers will have to fill in this table except for STUD_ATTENDED. This is because the system will automate the attendance when the students tapped their ID card.

STUDENT TABLE

class | masuk | student

stud_id	stud_matrix	stud_name	stud_shouldAttend	stud_attendec	stud_ic	stud_pcode	stud_semester	stud_ccode
1	2007123456	NADIAH BINTI MOHAMAD	10	8	860909-14-5628	EE220	8	ECE511
2	2007234567	MOHD FIRDOUZ BIN BAHTIAR	10	3	860802-14-5431	EE220	8	ECE511
3	2007345678	MOHD AIMAN BIN JOHARI	10	8	861221-04-5799	EE220	8	ECE511
4	2007456789	SHAMSUL ANUAR BIN MOHAMED	10	4	860809-01-5743	EE220	8	ECE511

Figure 4.13: Snapshot of Student Table

Figure 4.14 shows the “Class” table where lecturers need to fill in this table for the subject, date and time of the class and course code of the subject. The CLASS_DAY depends on the class session for a semester.

CLASS TABLE

class

class_id	class_name	class_day	class_code
1	Digital System & Microprocessor	1/1/2010, 1.00 PM	ECE511
2	Digital System & Microprocessor	2/2/2010, 2.00 PM	ECE511
3	Digital System & Microprocessor	3/3/2010, 3.00 PM	ECE511

Figure 4.14: Snapshot of Class Table

Figure 4.15 is the “Masuk” table where the table is updated when students tapped their ID cards. The table shows the matrix number of students with the class ID.

MASUK TABLE

class | masuk

ID	stud_matrix	class_id
21	2007123456	1
22	2007234567	1
23	2007345678	1
24	2007456789	1

Figure 4.15: Snapshot of Masuk Table

4.5 LIMITATIONS OF THE SYSTEM

Every crucial part of the system is simulated and tested for its reliability. Therefore, it was discovered that the components has their limitations and weaknesses.

During the development of the system, each of the components was put under test for its reliability and limitations. A few experiments had been conducted to ensure that the access part of the system (namely the RFID reader) is able to read the RFID smartcard in considerably quick time and effectively putting into considerations of the distance between the smartcard and the reader.

Due to the nature of passive RFID reader, low cost and manufacturing settings, the distance test conducted unto the RFID reader was found ineffective to read the smartcard after being put 5cm away from the reader.

The data delay of 0.25 seconds is inevitable due to the reading method of the Wiegand protocol. Some other delays include the LCD display and the serial communication between the hardware and the PC. Hence the accumulated delay will cause the user of a 3 second usage to register itself to the attendance system.

The RFID reader has an only one card tap/swipe per reading to reduce bottleneck. Hence a more than one card tap/swipe will cause the reader to halt/stop. The RFID works only at a close proximity of 0cm to 5cm (max) reading distance. If user fails to get in the reading range, the system will fail to acknowledge the student's attendance.

Due to usage of serial to USB conversion and vice versa, the delays might cause the overlapping data to be read by the system. This too has to be put into consideration. Although the provision of parity can be done, it was concluded that by not including the parity may reduce the complexity of the receiving end of the system.

CHAPTER 5

CONCLUSION AND FUTURE DEVELOPMENT

5.1 CONCLUSION

The beauty of the system provides automation to the system for cost reduction and by achieving the objectives to provide automation to the students' attendance record system integrated with RFID.

This project successfully achieved the objectives. The system can interface RFID with PC. The system is equipped with other modules that ease the process of recording attendance for surveillance purposes. The interconnection between forms bring the system achieves its goal where it can calculates the percentage of attendance and shows the status of 80% attendance.

5.2 FUTURE DEVELOPMENT

The future development of the system can be done by improving both hardware and software aspects.

Through the hardware, the system delay can be reduced to quicken the read time by the RFID. The hardware can be modified to read multiple cards at once. Through the software, the program can be modified and the system can be applied to both students and staff. The access to the system can be developed to be accessed through the internet too. The system can be hooked to a server so that the attendance can be viewed by lecturers and students through the internet. The system can be able to notify lecturers, students or parent through SMS or email.

REFERENCES

[1] T.S.Lim, S.C. Sim and M.M. Mansor, *"RFID Based Attendance System"*, Faculty of Engineering and Technology, Multimedia University.

[2] Jonathan Sidi, Syahruk N. Junaini and Lau S. Ling, *"ISAMS: Tracking Student Attendance using interactive Student Attendance Management System"*, Department of Computing and Software Engineering, Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak.

[3] Aysha Qaiser and Shoah A Khan, *"Automation of Time and Attendance using RFID Systems"*, Faculty of Software Engineering, Fatima Jinnah Women University.

[4] Nigel Gardner, *"PICmicro MCU C – An introduction to programming The Microchip PIC in CCS C"*, Bluebird Electronics 2002.

[5] Finkenzeller, Klaus. *"RFID handbook: Fundamentals and applications in contactless smart cards and identification translated by Rachel Waddington"*, 2nd edition. John Wiley & Sons, Ltd. 2003.

[6] Martin P Bates, *"Programming 8-bit PIC microcontrollers in C with interactive hardware simulation"*, Newness 2004.

[7] Lou Tylee, *"Microsoft Visual Basic"*, KIDware, 1998.

[8] Francesco Balena, *"Programming Microsoft Visual Basic 6.0"*, Microsoft Press, 1999.

[9] Assa Abloy, "*Understanding Card Data Formats – Wiegand Format*", HID Corporation, 2005.

[10] Cytron Technologies, *"UIC00A USB ICSP PIC Programmer"*, Cytron Technologies Sdn. Bhd., 2008.

[11] Microchip, *"PIC16F877A Data Sheet – 28/40/44-Pins Enhanced Flash Microcontrollers"*, Microchip Technologies Inc., 2003.

[12] Robotshop, *"Wiegand Protocol Format"*, RobotShop Distribution Inc., 2004

APPENDICES

Appendix A: Schematic Diagram of the System

Appendix B: SK40B Manual

Appendix C: PIC16F877A Datasheet

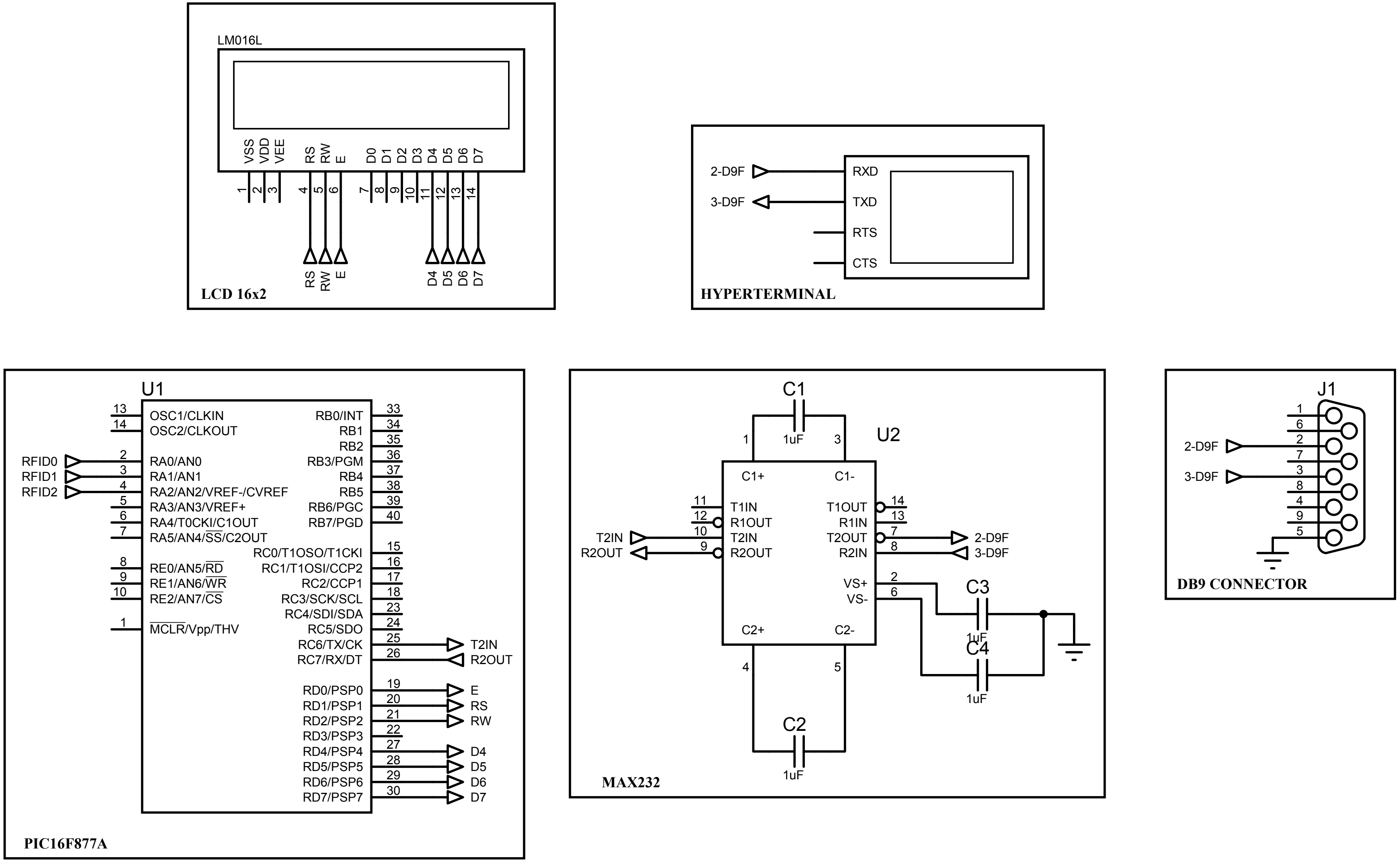
Appendix D: JHD 162A Datasheet

Appendix E: Wiegand RFID Reader Datasheet

Appendix F: Program

APPENDIX A

SCHEMATIC DIAGRAM



Title: UITM SMART ATTENDANCE ACCESS CONTROL SYSTEM

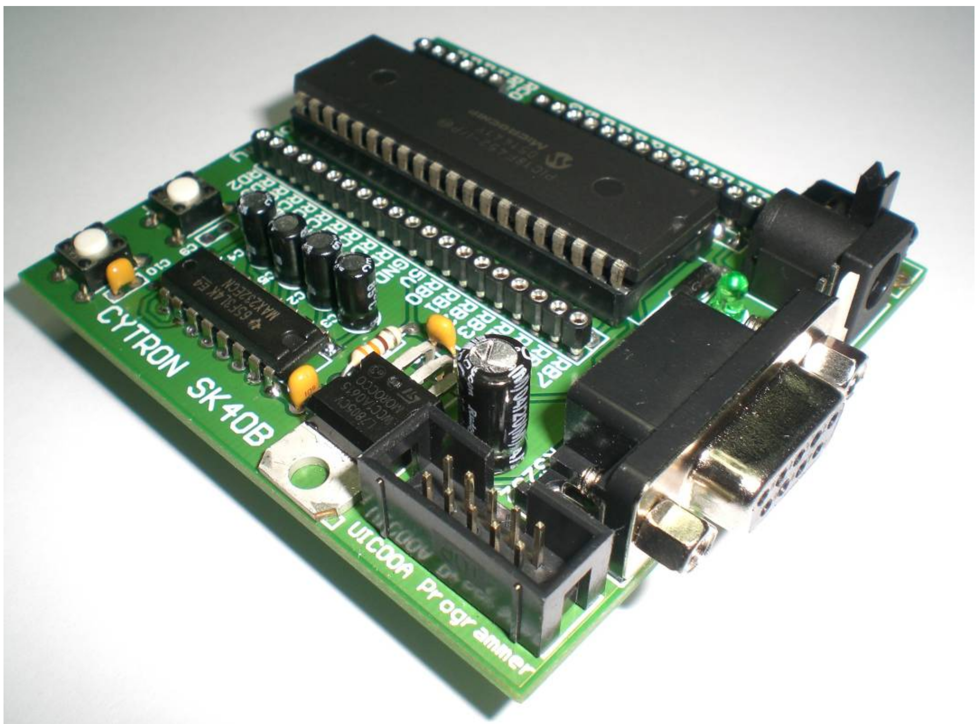
By: ZAHARAH BINTI HAIRUZI

APPENDIX B

SK40B MANUAL

SK40B

PIC MICROCONTROLLER START-UP KIT



User's Manual

V1.1

Dec 2007

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Index

1. INTRODUCTION AND OVERVIEW

SK40B is enhanced version of SK40A. It is designed to offer an easy to start board for PIC MCU user. However, all interface and program should be developed by user. This board comes with basic element for user to begin project development. It offer plug and use features. This kit is designed to offer:

- Industrial grade PCB
- Every board is being fully tested before shipped to customer
- Compact, powerful, flexible and robust start-up platform
- Suitable for hobbyists and experts
- Save development and soldering time
- No extra components required for the PIC to function
- **All 33 I/O pins** are nicely labeled to avoid miss-connection by users
- **Connector for UIC00A** (low cost USB ICSP PIC Programmer) - simple and fast method to load program
- RS232 (Serial) hardware on board allowing alternative method to load program easily via bootloader
- Fully compatible with SK40A
- No more frustrated work plugging PIC out and back for re-programming
- Perfectly fit for 40 pins **16F and PIC18F PIC**
- No more confusion between programmer and bootloader
- With UIC00A, program can be loaded in less than 5 seconds

SK40A come with:

- 5V voltage regulator (1A max)
- 20MHz crystal oscillator
- Reset button
- 1 x programmable push button
- RS232 hardware for serial communication to PC
- Connector for UIC00A
- On/Off switch for main power
- DC adaptor socket as power input
- And all the necessities to eliminate users difficulty in using PIC

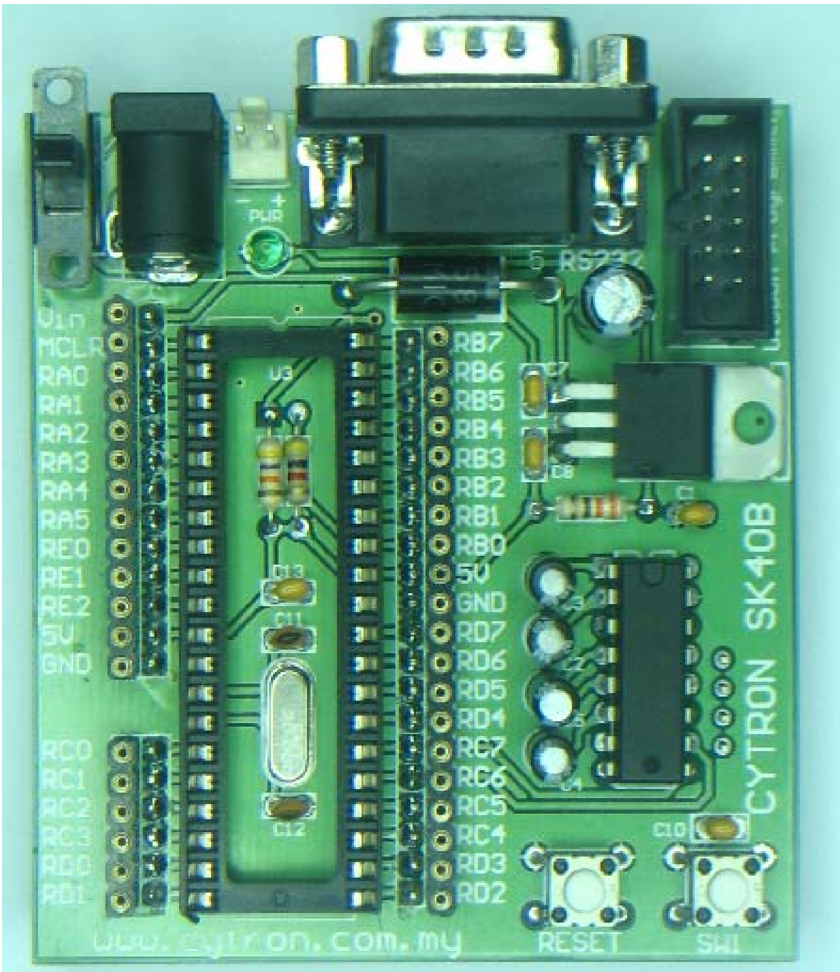
Users are able to utilize the function of PIC by directly plugging in the I/O components in whatever way that is convenient to user. With UIC00A connector on board, user can start developing projects and have fun with this kit right away. Of course, bootloader can still be applied in loading program. This kit comes **WITHOUT** PIC microcontroller to provide the freedom for user to choose PIC type.

This document explains the method to use SK40B.

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

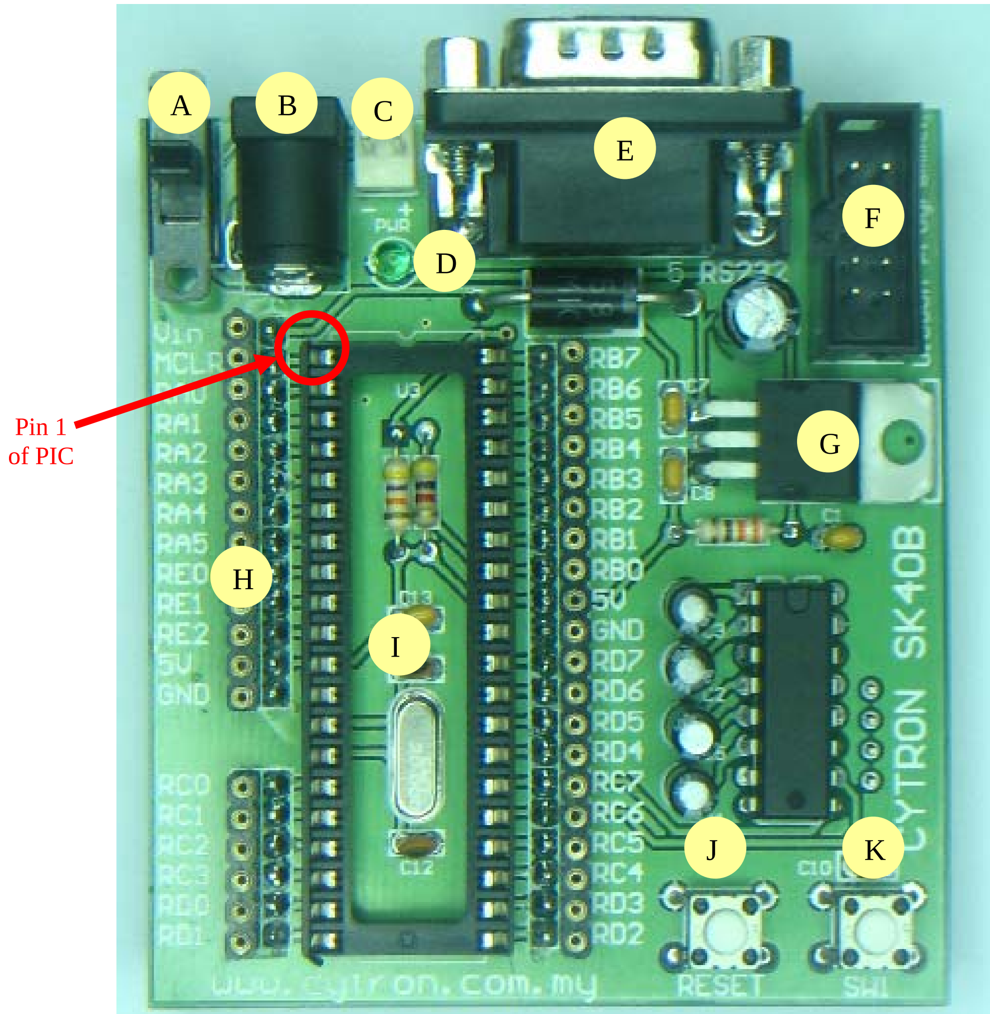
2. PACKAGING LIST

Please check the parts and components according to the packing list. If there are any parts missing, please contact us at sales@cytron.com.my immediately.



1. 1 x SK40B board with all components shown soldered
2. PIC MCU – Not included, please purchase separately from Cytron website
3. Serial Cable – Not included, please purchase separately from Cytron website
4. UIC00A – Not included, please purchase separately from Cytron website
5. User Manual – Not included, please download from Cytron website

3. BOARD LAYOUT



Label	Function	Label	Function
A	Slide Switch for main power supply	**G**	5V regulator
B	DC power adaptor socket	**H**	Header pin and turn pin
C	Battery connector	**I**	40 pin IC socket for PIC MCU
D	Power indicator LED	**J**	Reset button
E	Serial cable adaptor (female DB9)	**K**	Programmable push button
F	Connector for UIC00A Programmer		

A – is a slide switch to On/Off the power supply from DC adaptor or Battery connector. Pushing the switch down will ON SK40B.

B – is a DC power adaptor socket for user to plug in DC adaptor. The input voltage should be ranged from 7 to 15V.

C – is a 2510 2 way connector for battery input. The battery voltage should be between 7 to 15V. Please ensure the polarity of voltage is correctly plugged before power up SK40B. The "+" and "--" is have been labeled at the side of connector.

D – is an indicator LED for on board 5V regulator. It will light ON as long as the input power is correctly connected and the slide switch is ON.

E – is a serial cable connector, female DB9. It should be connected to male DB9 serial cable.

F – is a 2x5 box header for UIC00A, USB ICSP PIC Programmer.

G – is a 5V voltage regulator

H – Consist of several line of header pin and turn pin. Header pin provide connector for user to solder SK40B to prototype board and use the I/O of PIC MCU. It is fully compatible between SK40A. Turn pin offer simple way to check voltage with multi-meter probe. 40 pins of PIC MCU except OSC (connected to crystal) are extended out to these pin. There is an extra pin on top of MCLR which is labeled as Vin, is connected to the input power.

I – 40 pin IC socket for user to plug in any 40 pin PIC MCU (8 bit). It can either be 16F or 18F PIC. Of course the IC package should be PDIP. **Please ensure the first pin is at the top side.** Inside IC socket, there some electronics components, it include a 20MHz Crystal.

J – is a push button with the function of Reset for PIC MCU.

K – is a push button connected to RB0 of PIC MCU. This is an extra input button for user. It can be programmed as an input switch.

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

4. PRODUCT SPECIFICATION

SK40B is designed to offer starting up platform for development, the specification of PIC MCU used should be referred.

However, there is a limitation that user should beware of. The input supply should be **from 7 to 15V**. Battery or DC adaptor can be used.

Note: Only 1 power supply should be provided to SK40B.

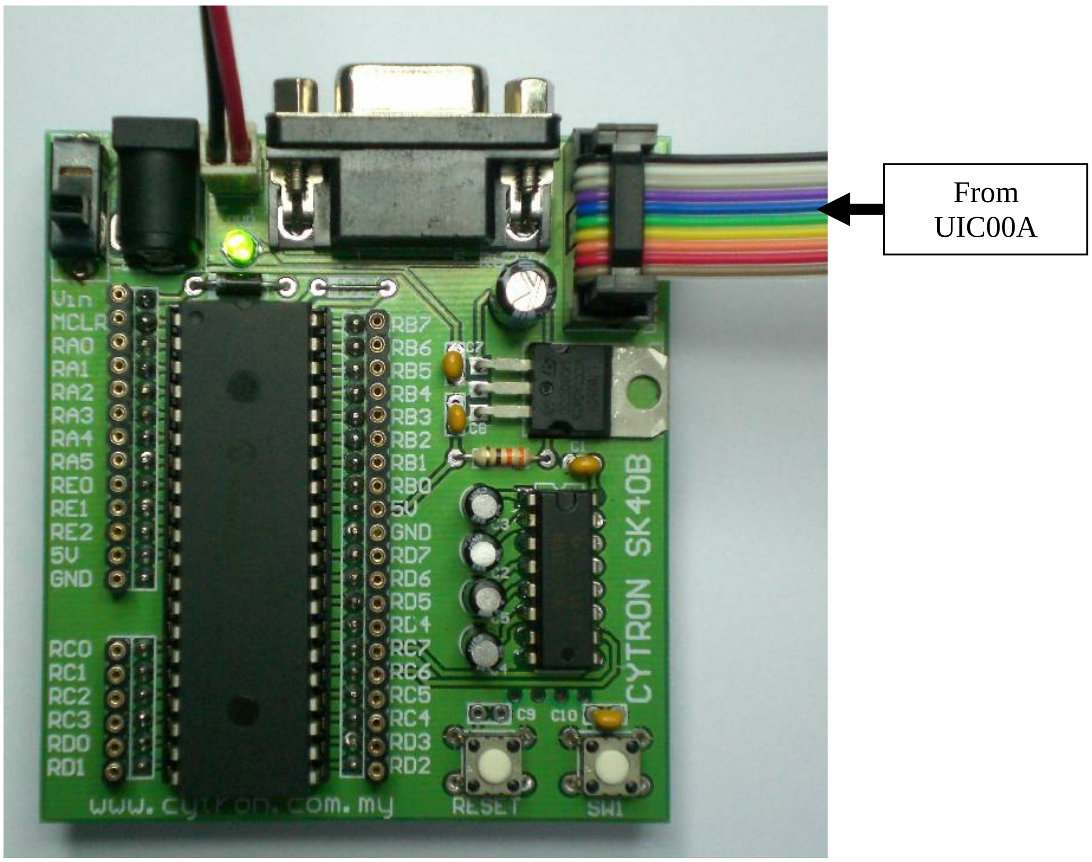
Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

5. INSTALLATION (HARDWARE)

SK40B come with UIC00A ICSP USB programmer connector and bootloader capability to offer simple way for downloading program. However, from user feedback and experience, using UIC00A ICSP programmer is very easy and save plenty of development time.

5.1 Loading Program Using UIC00A Programmer

Connect SK40B as shown in following figure.



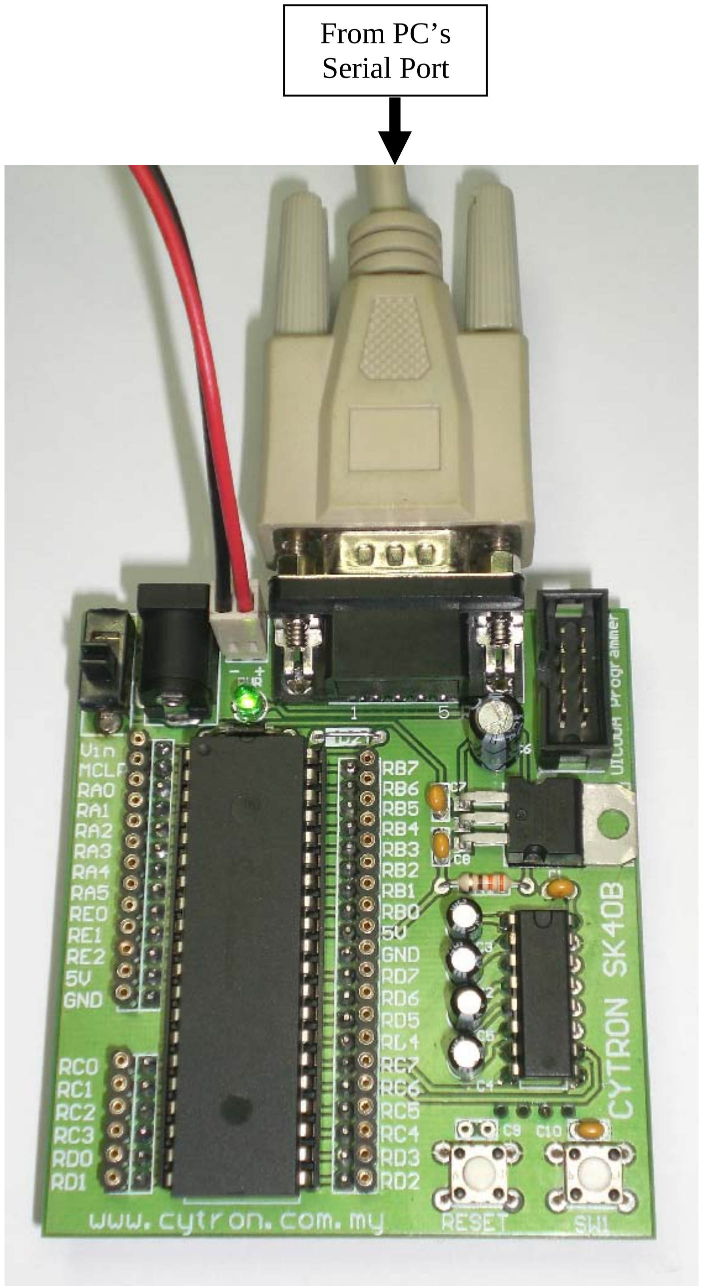
After plug in 40 pin PIC MCU (make sure the orientation is correct), SK40B should be powered **either from battery or DC adaptor**. Now, the hex code is ready to be loaded to SK40B. For the usage of UIC00A, please refer to UIC00A User's Manual.

RB7 and RB6 have been connected to UIC00A connector; both these pins are used for ICSP. User is advice not to use these pin as input. Even when using as output, RB7 or RB6 pin are recommended to be used in controlling non critical device such as LED, LCD, 7 segments or buzzer. It is recommended to isolated ICSP signals from application circuit by using series resistor (range 220 ohm and above). Furthermore, **NO** capacitance component (capacitor) should be connected to these 2 pins. Please refer to UIC00A User's Manual for further details.

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

5.2 Loading Program Using Bootloader

Connect SK40B as shown in following figure.



After plug in 40 pin PIC MCU with bootloader firmware (make sure the orientation is correct), SK40B should be powered either from battery or DC adaptor. Now, the hex code is ready to be loaded to SK40B. For the method of using bootloader, please refer to SK40A User's Manual.

Since SK40B have bootloader hardware, 2 pins have been used for bootloader function. Pins that involved are:

- RC7, used as RxD and connected to MAX232

- RC6, used as TxD and connected to MAX232

Please refer to Appendix A for the detail of schematic. Do consider carefully these pre-connect interface if development require those pins.

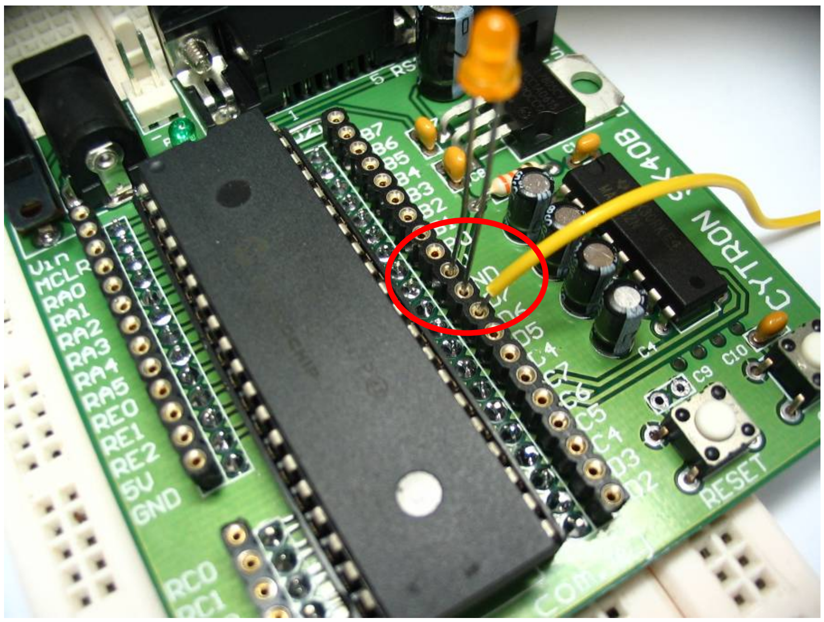
Note: New PIC will not work with bootloader, bootloader firmware is required.

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

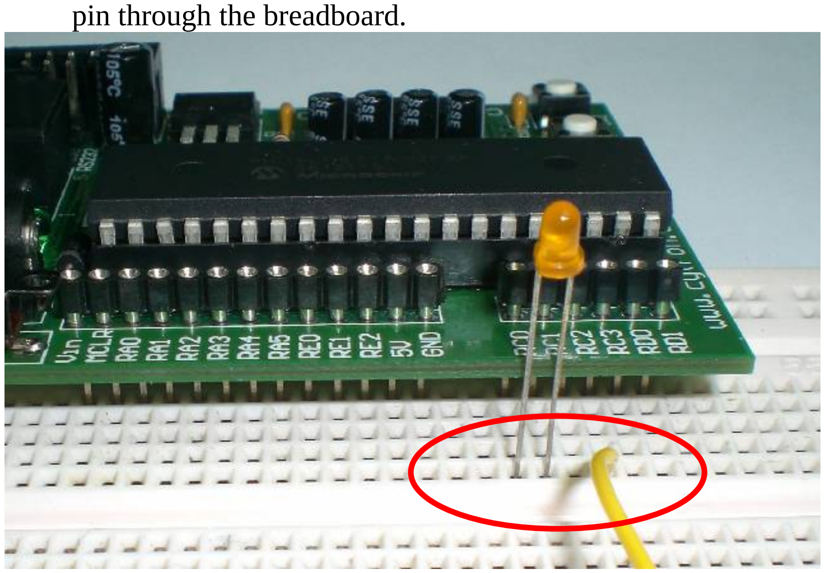
6. GETTING STARTED

SK40B is ready to be plug and use, no extra driver or software is necessary. It is a hardware platform, for those that use bootloader or UIC00A, please install the necessary driver or configure the correct setting in window. SK40B is ready be used to start the electronics interface. The I/O of the microcontroller can be access through few methods:

1. I/O port (Top)
    - Connect the components that needed onto the I/O port.
    - Extend the I/O port to another board using jumper wire

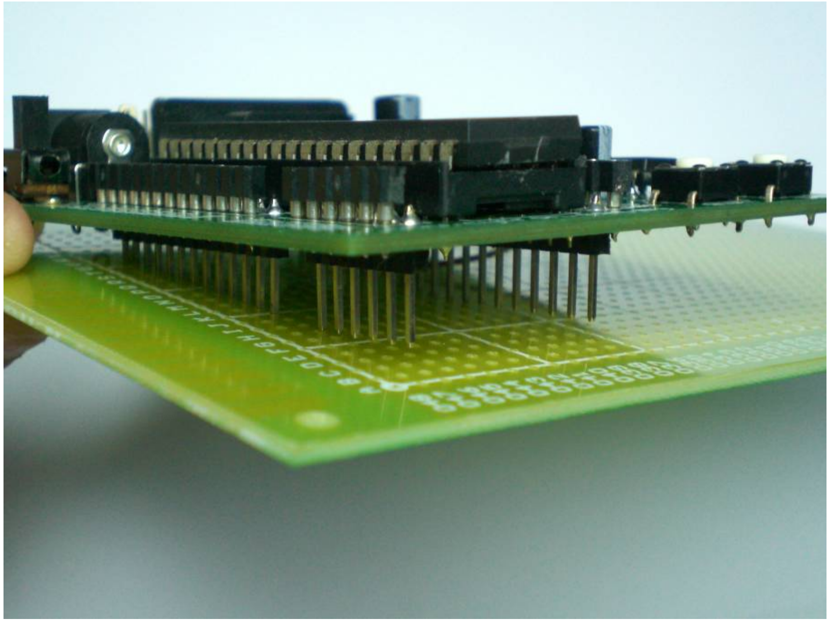


2. I/O pins (Bottom)
    - Plug the I/O pins of the Start-up Kit onto a breadboard. Then, access I/O pin through the breadboard.



Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

- Plug the I/O pins of the Start-up Kit onto a donut board. Solder the pins onto the board to access the I/O.



7. WARRANTY

- Product warranty is valid for 6 months.
- Warranty only applies to manufacturing defect.
- Damage caused by mis-use is not covered under warranty.
- Warranty does not cover freight cost for both ways.

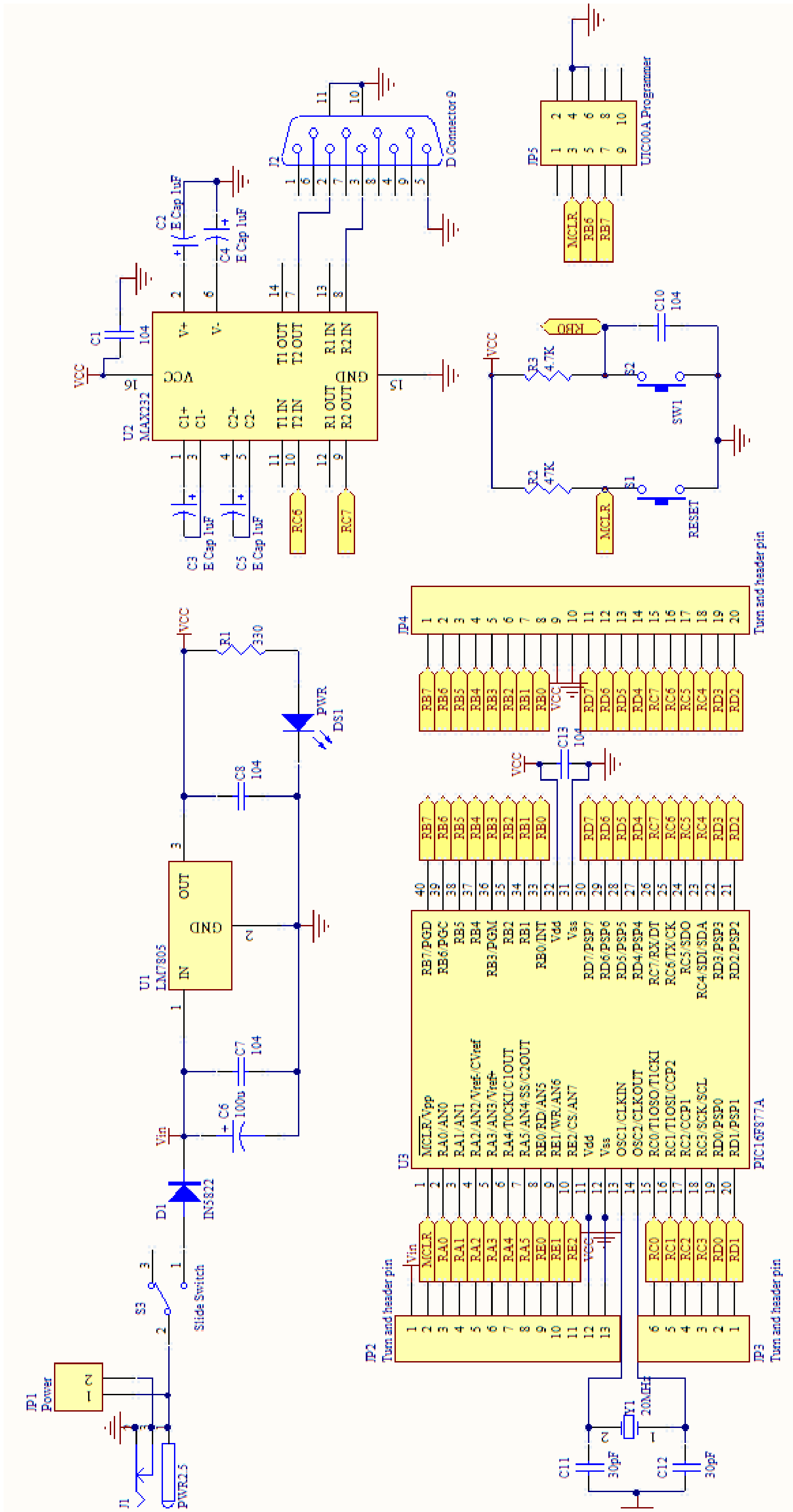
Prepared by
Cytron Technologies Sdn. Bhd.
19, Jalan Kebudayaan 1A,
Taman Universiti,
81300 Skudai,
Johor, Malaysia.

Tel: +607-521 3178
Fax: +607-521 1861

URL: www.cytron.com.my
Email: support@cytron.com.my
sales@cytron.com.my

Created by Cytron Technologies Sdn. Bhd. – All Rights Reserved

Appendix A


MAX232
D Connector 9
UIC00A Programmer
RESET
SW1
LM7805
PWR
Slide Switch
Power
PWR2.5
Turn and header pin
20MHz
30pF
PIC16F877A

APPENDIX C

PIC16F877A DATASHEET

APPENDIX D

JHD 162A DATASHEET

JHD162A SERIES

■CHARACTERISTICS：

DISPLAY CONTENT：16 CHAR x 2ROW

CHAR. DOTS：5 x 8

DRIVING MODE：1/16D

AVAILABLE TYPES：

TN，STN(YELLOW GREEN、GREY、B/W)

REFLECTIVE、WITH EL OR LED BACKLIGHT

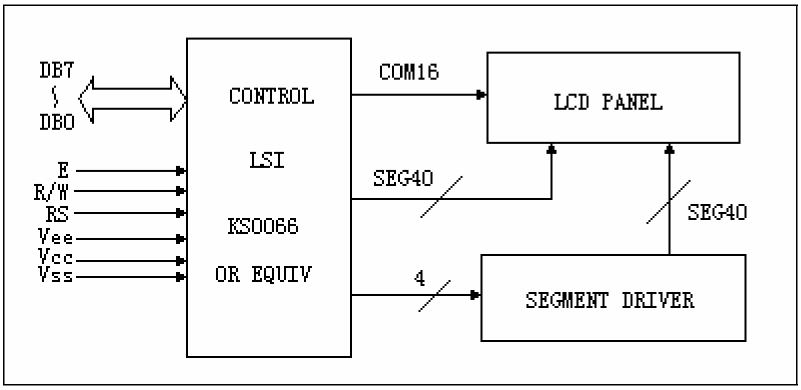
EL/100VAC，400HZ

LED/4.2VDC

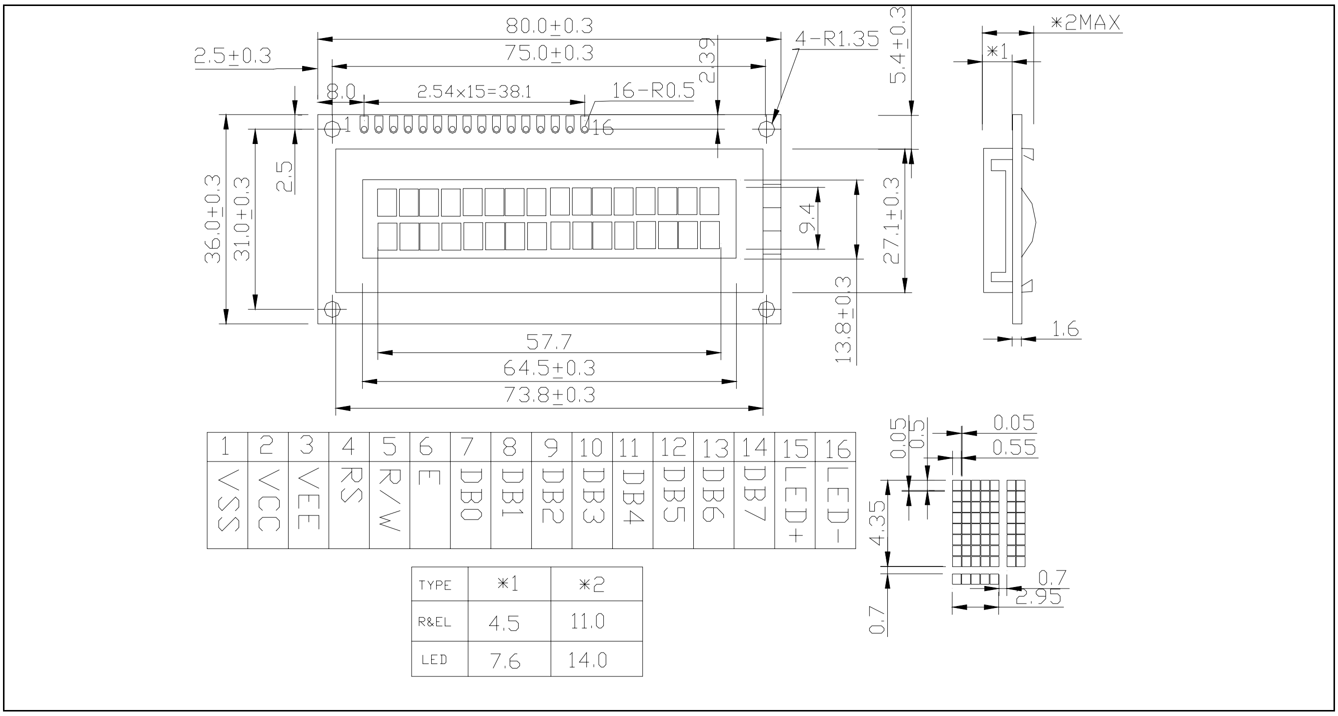
■PARAMETER (V_{DD}=5.0V±10%,V_{SS}=0V,T_a=25°C)

Parameter	Symbol	**Testing Criteria**	**Standard Values**			Unit
			Min.	Typ.	Max	
Supply voltage	V_{DD}-V_{SS}	-	4.5	5.0	5.5	V
Input high voltage	V_{IH}	-	2.2	-	V_{DD}	V
Input low voltage	V_{IL}	-	-0.3	-	0.6	V
Output high voltage	V_{OH}	-I_{OH}=02mA	2.4	-	-	V
Output low voltage	V_{OL}	I_{OL}=1.2mA	-	-	0.4	V
Operating voltage	I_{DD}	V_{DD}=5.0V	-	1.5	3.0	mA

■APPLICATION CIRCUIT



■DIMENSIONS/DISPLAY CONTENT



PIN CONFIGURATION

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VSS	VCC	VEE	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	LED+	LED-

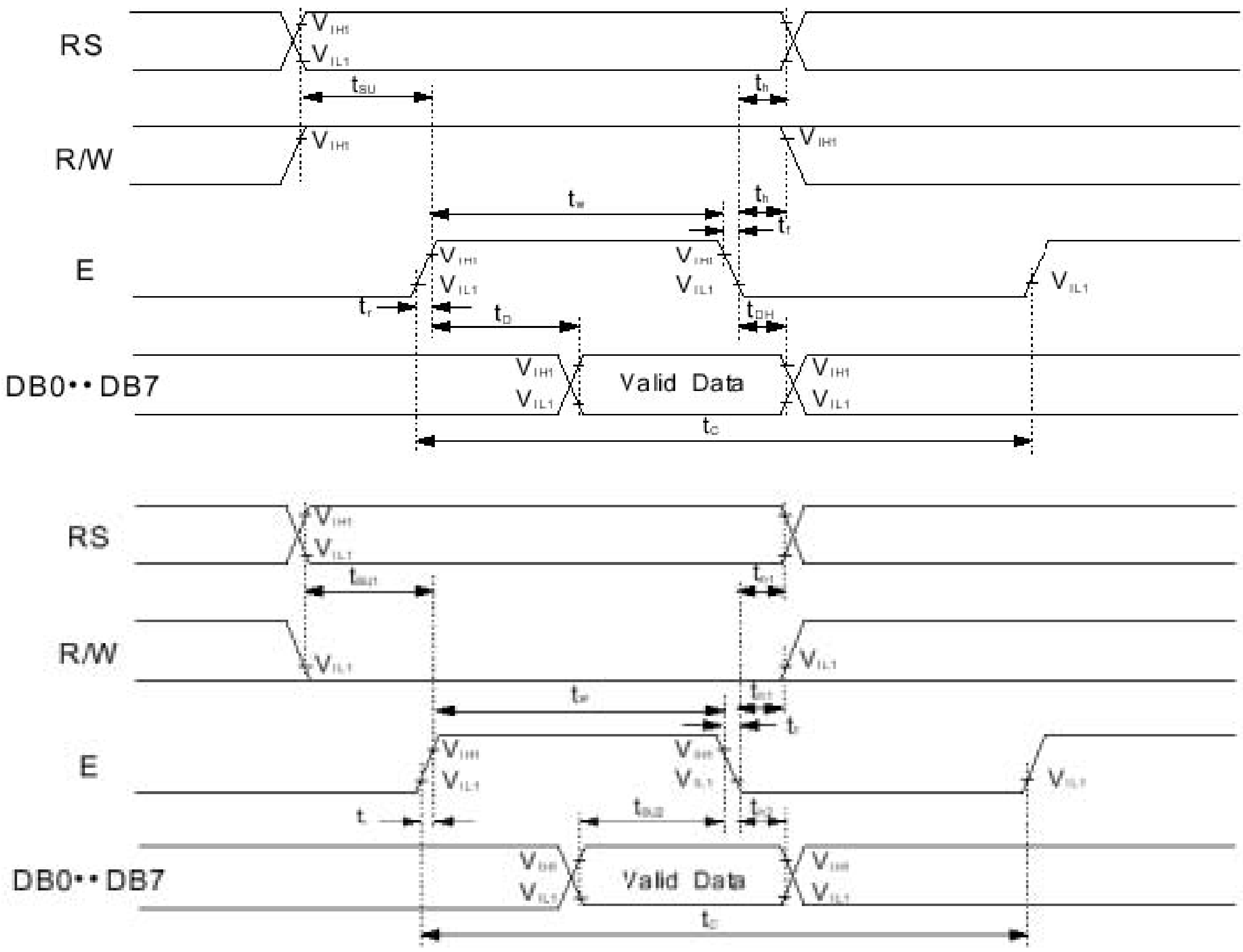
AC Characteristics Read Mode Timing Diagram

Table 12. AC Characteristics (V_{DD} = 4.5V ~ 5.5V, Ta = -30 ~ +85°C)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode (Refer to Fig-6)	E Cycle Time	tc	500	-	-	ns
	E Rise / Fall Time	t_R,t_F	-	-	20	
	E Pulse Width (High, Low)	tw	230	-	-	
	R/W and RS Setup Time	tsu1	40	-	-	
	R/W and RS Hold Time	t_{H1}	10	-	-	
	Data Setup Time	tsu2	80	-	-	
	Data Hold Time	t_{H2}	10	-	-	
Read Mode (Refer to Fig-7)	E Cycle Time	tc	500	-	-	ns
	E Rise / Fall Time	t_R,t_F	-	-	20	
	E Pulse Width (High, Low)	tw	230	-	-	
	R/W and RS Setup Time	tsu	40	-	-	
	R/W and RS Hold Time	t_H	10	-	-	
	Data Output Delay Time	t_D	-	-	120	
	Data Hold Time	t_{DH}	5	-	-	

Table 13. AC Characteristics (V_{DD} =2.7V ~ 4.5V, Ta = -30 ~ +85°C)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode (Refer to Fig-6)	E Cycle Time	tc	1000	-	-	ns
	E Rise / Fall Time	t_Rt_F	-	-	25	
	E Pulse Width (High, Low)	tw	450	-	-	
	R/W and RS Setup Time	tsu1	60	-	-	
	R/W and RS Hold Time	t_{H1}	20	-	-	
	Data Setup Time	tsu2	195	-	-	
	Data Hold Time	t_{H2}	10	-	-	
Read Mode (Refer to Fig-7)	E Cycle Time	tc	1000	-	-	ns
	E Rise / Fall Time	t_R,t_F	-	-	25	
	E Pulse Width (High, Low)	tw	450	-	-	
	R/W and RS Setup Time	tsu	60	-	-	
	R/W and RS Hold Time	t_H	20	-	-	
	Data Output Delay Time	t_D	-	-	360	
	Data Hold Time	t_{DH}	5	-	-	

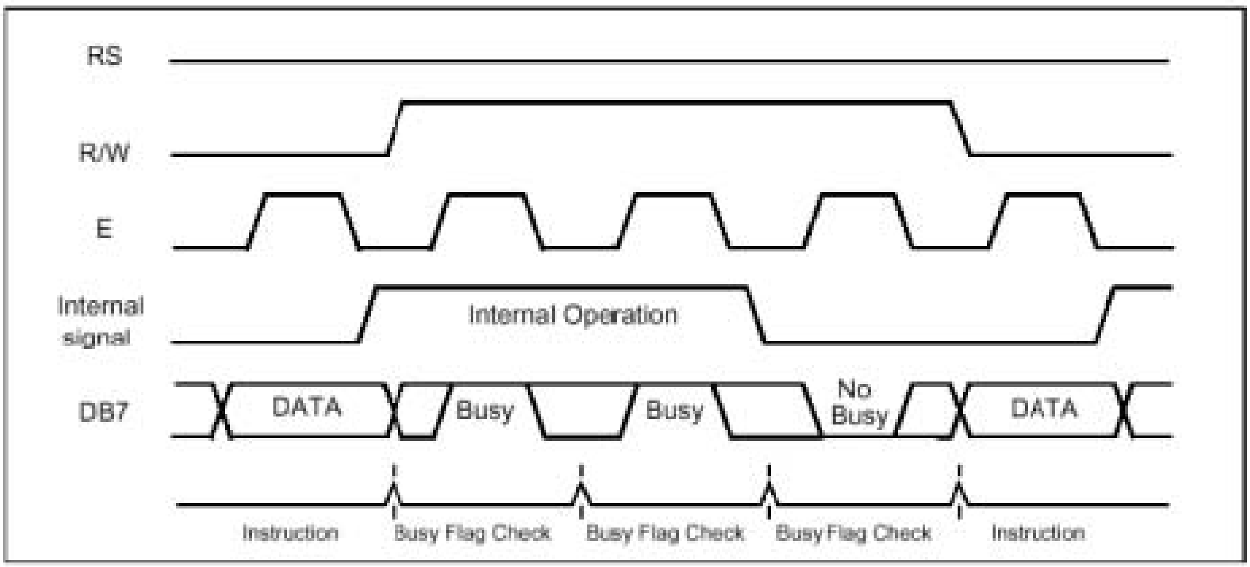


Write Mode Timing Diagram

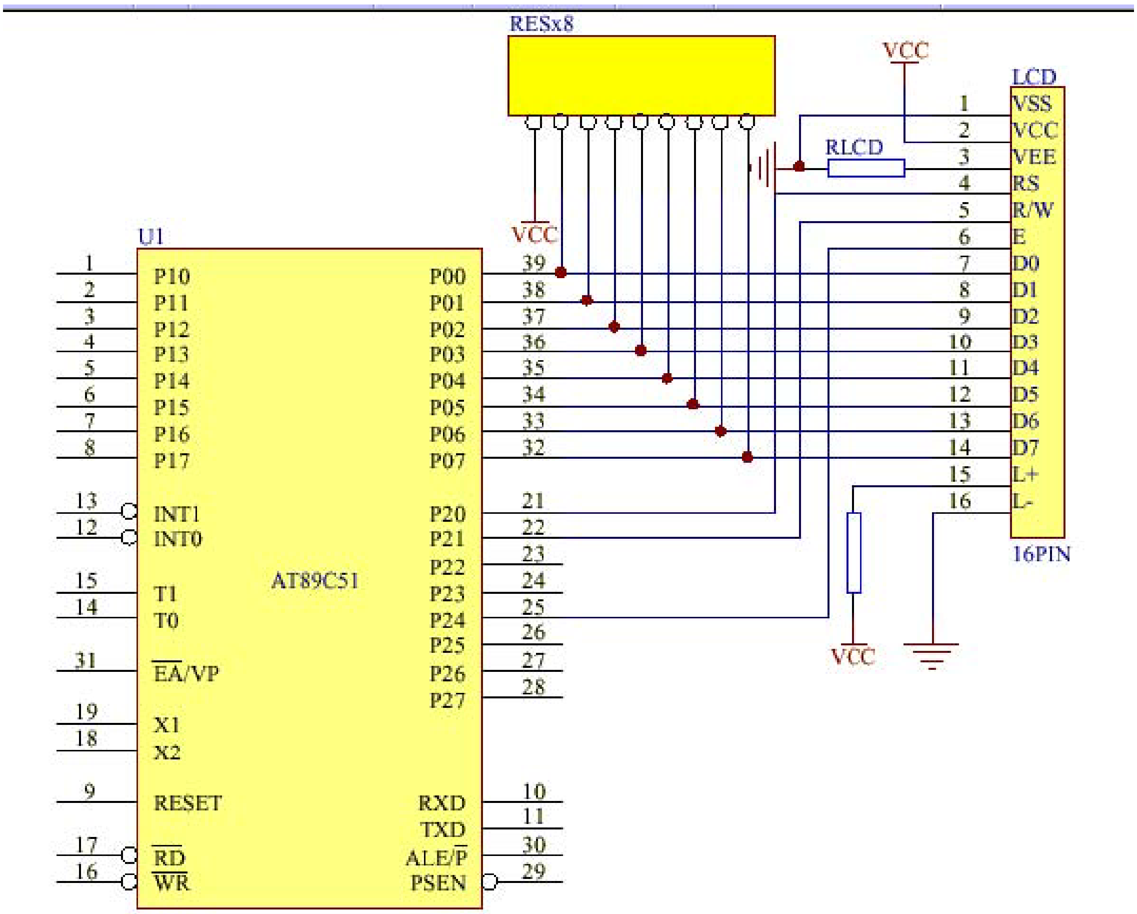
Timing

1) Interface with 8-bit MPU

When interfacing data length are 8-bit, transfer is performed at a time through 8 ports, from DB0 to DB7. Example of timing sequence is shown below.



Connection



CGROM

Table 5. Relationship between Character Code (DDRAM) and Character Pattern (CGRAM)

Character Code (DDRAM data)								CGRAM Address						CGRAM Data								Pattern number
D7	D6	D5	D4	D3	D2	D1	D0	A5	A4	A3	A2	A1	A0	P7	P6	P5	P4	P3	P2	P1	P0	
0	0	0	0	×	0	0	0	0	0	0	0	0	0	×	×	×	0	1	1	1	0	pattern 1
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	0	0	1	
											0	1	1				1	1	1	1	1	
											1	0	0				1	0	0	0	1	
											1	0	1				1	0	0	0	1	
											1	1	0				1	0	0	0	1	
											1	1	1				0	0	0	0	0	
				⋮							⋮								⋮			⋮
0	0	0	0	×	1	1	1	0	0	0	0	0	0	×	×	×	1	0	0	0	1	pattern 8
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	0	0	1	
											0	1	1				1	1	1	1	1	
											1	0	0				1	0	0	0	1	
											1	0	1				1	0	0	0	1	
											1	1	0				1	0	0	0	1	
											1	1	1				0	0	0	0	0	

Example

```
#include <reg51.h>
#include <intrins.h>
sbit dc=0xa0;      /*P2.0 LCD 的 RS 21*/
sbit rw=0xa1;      /*P2.1 LCD 的R/W 22*/
sbit cs=0xa4;      /*P2.4 LCD 的 E 25*/
sfr  lcdbus=0x80;  /*p0LCD 数据 D0=P0.0*/
unsigned int sys10mscounter;
unsigned char syslimitcounter;
char path1[8]={0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f};/*自定义符号 横1*/
char path2[8]={0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00};/*自定义符号 横2*/
char pats1[8]={0x15,0x15,0x15,0x15,0x15,0x15,0x15,0x15};/*自定义符号 竖1*/
char pats2[8]={0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a};/*自定义符号 竖2*/
void soft_nop(){}
void soft_10ms()/***********12MHZ 提供10MS 软件延时************/
{    register int i;
for(i=0;i<711;i++);
```

```
}
void soft_20ms()/***********12MHZ 提供20MS 软件延时************/
{	soft_10ms();
soft_10ms();
}
void hard_10ms(unsigned int delaytime) /*基于10MS 的硬件延时*/
{	sys10mscounter=delaytime;
while(sys10mscounter);
}
unsigned char data lcdcounter;
bit lcdusing1,lcdusing2;
bit lcd_checkbusy()/*检查LCD 忙*/
{	register lcdstate;
	dc=0;			/*dc=1为数据,=0 为命令.*/
	rw=1;			/*rw=1为读,=0 为写.*/
	cs=1;			/*cs=1选通.*/
soft_nop();
lcdstate=lcdbus;
cs=0;
return((bit)(lcdstate&0x80));
}
void lcd_wrcmd(unsigned char lcdcmd) /*写LCD 命令*/
{	lcdusing1=1;
while(lcd_checkbusy());
lcdbus=lcdcmd;
	dc=0;		/*dc=1为数据,=0 为命令.*/
	rw=0;		/*rw=1为读,=0 为写.*/
	cs=1;		/*cs=1选通.*/
soft_nop();

	cs=0;
	lcdbus=0xff;
	lcdusing1=0;

}
		void lcd_moveto(char position) /*移动光标到指定位.0-79*/
		{ register cmd=0x80;

	lcdcounter=position;
		if (position > 59)
		position += 0x18;
		else
```

```
        {    if (position > 39)position -= 0x14;
              else
             {                   if (position > 19)position += 0x2c;
             }
        }

    cmd=cmd|position;
    lcd_wrcmd(cmd); } void lcd_wrdata(char lcddata) /*在当前显示位置显示数据*/ { char i;
    lcdusing2=1;
    while(lcd_checkbusy());
    if(lcdcounter==20){

        lcd_moveto(20);
        while(lcd_checkbusy());
        }

    if(lcdcounter==40){
        lcd_moveto(40);
        while(lcd_checkbusy());
        }

    if(lcdcounter==60){
        lcd_moveto(60);
        while(lcd_checkbusy());
        }

    if(lcdcounter==80){
        lcd_moveto(0);
        while(lcd_checkbusy());
        lcdcounter=0;
        } /*为通用而如此*/

    lcdcounter++;
    lcdbus=lcddata;
    dc=1; /*dc=1为数据,=0  为命令.*/
    rw=0; /*rw=1为读,=0  为写.*/
    cs=1; /*cs=1选通.*/
    soft_nop();
    cs=0;

    lcdbus=0xff;
    lcdusing2=0; } void lcd_string(char *strpoint) /*在当前显示位置显示LCD  字符串*/
{ register i=0;
    while(strpoint[i]!=0){
```

```
            lcd_wrdata(strpoint[i]);
            i++;
            }

} void lcd_init()/*初始化*/

{   lcd_wrcmd(0x38);        /*设置8  位格式,2  行,5*7*/
    lcd_wrcmd(0x0c);        /*整体显示,关光标,不闪烁*/
    lcd_wrcmd(0x06);        /*设定输入方式,增量不移位*/
    lcd_wrcmd(0x01);        /*清除显示*/
    lcdcounter=0;
}

void lcd_cls()/*清除显示*/ { lcd_wrcmd(0x01);
      lcdcounter=0; } void timer0(void) interrupt 1 /*T0  中断*/ { TH0=0xd8; /*12M,10ms*/
      TL0=0xf6;
      TR0=1;
      if(sys10mscounter!=0)sys10mscounter--; /*定时器10ms*/
      if(syslimitcounter!=0)syslimitcounter--; /*定时器10ms*/

}
               main()
               {
               unsigned char j;
               IE=0;P0=0xff;P1=0xff;P2=0xff;P3=0xff; /*初始化T*/
               lcd_init();soft_20ms();
               TMOD=0x51;
               TH0=0xd8; /*12M,10ms*/
               TL0=0xf6;
               TR0=1;ET0=1;EA=1;

      while(1)
      {
      /*全黑横一横二竖一竖二U Q ABCD... */
      lcd_init(); /*全黑*/
      for(j=0;j<80;j++){lcd_wrdata(0xff);}
      hard_10ms(50);
      lcd_init(); /*横一可参考自行设计符号*/
      lcd_wrcmd(0x40);
      for(j=0;j<8;j++)lcd_wrdata(path1[j]);

      for(j=0;j<100;j++)lcd_wrdata(0);
      hard_10ms(50);
      lcd_init(); /*横二*/
```

```
        lcd_wrcmd(0x40);
        for(j=0;j<8;j++)lcd_wrdata(path2[j]);
        for(j=0;j<100;j++)lcd_wrdata(0);
        hard_10ms(50);
        lcd_init(); /*竖一*/
        lcd_wrcmd(0x40);
        for(j=0;j<8;j++)lcd_wrdata(pats1[j]);
        for(j=0;j<100;j++)lcd_wrdata(0);
        hard_10ms(50);
        lcd_init(); /*竖二*/
        lcd_wrcmd(0x40);
        for(j=0;j<8;j++)lcd_wrdata(pats2[j]);
        for(j=0;j<100;j++)lcd_wrdata(0);
        hard_10ms(50);
        lcd_init();
        lcd_string("UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
        UUUUUUUUUUUUUUUUUUUUUUUU

UUUUU"); hard_10ms(50); lcd_init();
        lcd_string("QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ
        QQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQ"); hard_10ms(50); lcd_init();
        lcd_string("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwx
        yz0123456789+-!
#$%&?"); hard_10ms(50); }
}
```

APPENDIX E

WIEGAND RFID READER DATASHEET

Understanding Card Data Formats

Wiegand™ Format

The term Wiegand is applied to several characteristics related to access control readers and cards. Unfortunately, the word is used carelessly and can lead to unnecessary confusion. Here are the basics.
Wiegand is:

1. A specific reader-to-card interface
2. A specific binary reader-to-controller interface
3. An electronic signal carrying data
4. The standard 26-bit binary card data format
5. An electromagnetic effect
6. A card technology

For the purposes of this white paper, we will address items 2 and 4.
(NOTE: There are additional card/reader attributes that are also described by the term, Wiegand.)

When HID customers say, "Wiegand format", they typically refer to the general concept of security card data encoding. But be aware that the term, Wiegand format, is also often understood to mean the standard 26-bit format, which is a very specific arrangement of binary card data. Some basic facts:

- A format describes what a number means, or how a number is used. The format is not the number itself,
- The number of bits does not indicate the format except for standard 26-bit. For example, there are over 100 different 34-bit formats alone.
- Within a given bit length (34-bit, 37-bit, etc.), the size and location of each data element may change. For example:
  - o One 34-bit format may have an 8-bit Facility Code starting with bit #2.
  - o Another 34-bit Facility Code may be 12 bits starting with bit # 21.
- The capability of the access control panel will dictate what formats will and will not work.

If I see a string of numbers, 19495981699 it may mean nothing. If you describe it as a phone number in the United States, then it is immediately understood that 949 is the area code, etc. Knowledge of the format allows you to decode the data. It always appears in the format, (xxx) yyy-zzzz, because telephone company switching equipment specifies it exist in this format.

The telephone company has maintained this format for many years and migrated to it slowly over the years adding numbers in groups. Security equipment has similar format demands however the security industry does not want the format known and they often change the formats to keep the changes confidential.

All specific card formats are identical in both 125 kHz Prox and 13.56 MHz iCLASS® cards. This ensures that any controller capable of understanding data from 125 kHz cards and readers will also seamlessly work with 13.56 MHz cards and readers.

The Standard 26-Bit Format

The format in which a card is programmed is determined by the data pattern that will be compatible with the access control panel. All HID credentials (card, fobs, tags, etc.) can be programmed with the standard 26-bit card data format.

The Standard 26-bit Format is an Open Format.
An Open Format means that anyone can buy HID cards in a specific format and that specific format description is publicly available. The 26-bit format is a widely used industry standard and is available to all HID customers. Almost all access control systems accept the standard 26-bit format. 26-bit originated with true Wiegand swipe card technology.

www.hidcorp.com
© 2005 HID Corporation. All rights reserved.
An ASSA ABLOY Group company
ASSA ABLOY

The HID ordering code number for the Standard 26-bit format is H10301.

H10301 has 255 possible facility codes from one to 255. There can be up to 65,535 card ID numbers, from one to 65,535, per facility code. The total number of cards that can use the entire range without duplication is 16,711,425. There are no restrictions on the use of this format. It is not documented by HID and HID does not restrict duplication of card numbers.

HID produces and manages over 1,000 other card data formats, some of them share the same fundamental concepts as the 26-bit format. Other card manufacturers also have unique, proprietary formats.

H10301 describes binary encoded data. The format is represented in the next figure:

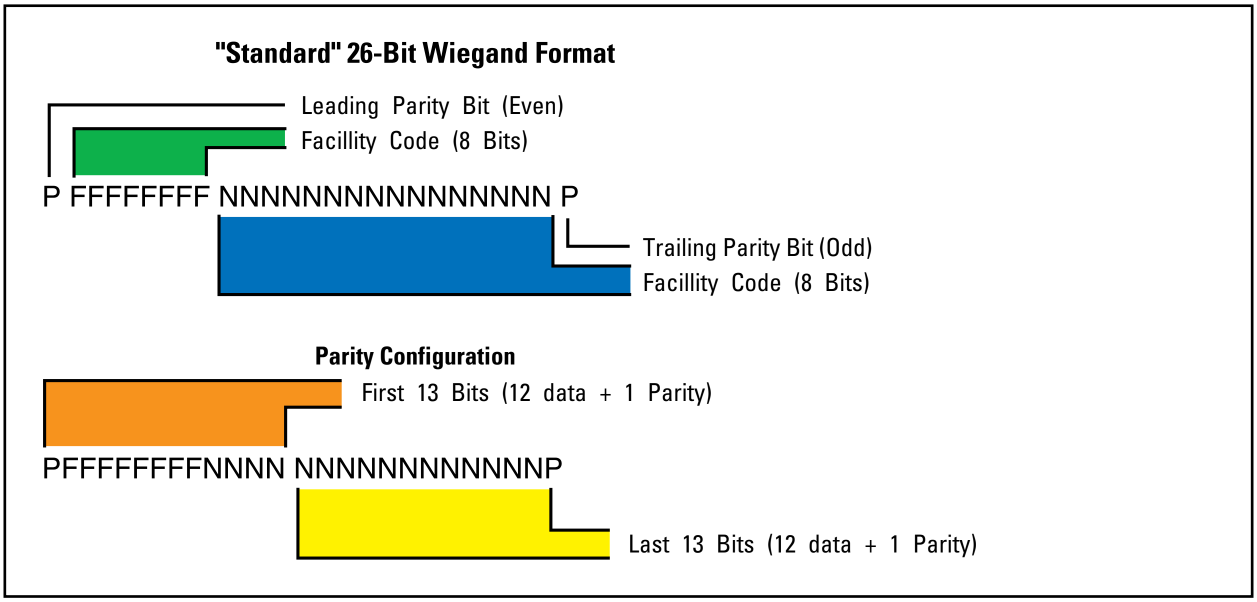


Figure 1: 26-Bit Wiegand Public Format

- The maximum Facility Code is 255 because if all eight Facility Code bits are set to ones, they equal 255 decimal.
- The maximum Card Number is 65,535 because when all sixteen Card Number field bits are ones, it equals decimal 65,535.

A NOTE ON PARITY: A parity bit is used as a very simple quality check for the accuracy of the transmitted binary data. The designer of the format program will decide if each parity bit should be even or odd. A selected group of data bits will be united with one parity bit, and the total number of bits should result in either an even or odd number.

In the example above, the leading parity bit (even) is linked to the first 12 data bits. If the 12 data bits result in an odd number, the parity bit is set to one to make the 13-bit total come out even. The final 13 bits are similarly set to an odd total.

Other Hypothetical Formats

To further clarify how formats may be organized, we present two additional hypothetical examples.

NOTE: Since actual formats do require a varying degree of security, we will only present hypothetical examples with the exception of standard 26-bit.

© 2005 HID Corporation. All rights reserved.

In the standard 26-bit format, H10301, or the programmable field is specified as the Facility Code. The incrementing field is called the Card Number. These data groupings can have many different names depending upon which format is under discussion. The same name usually means something different from format to format. Therefore, another hypothetical format could look like this:

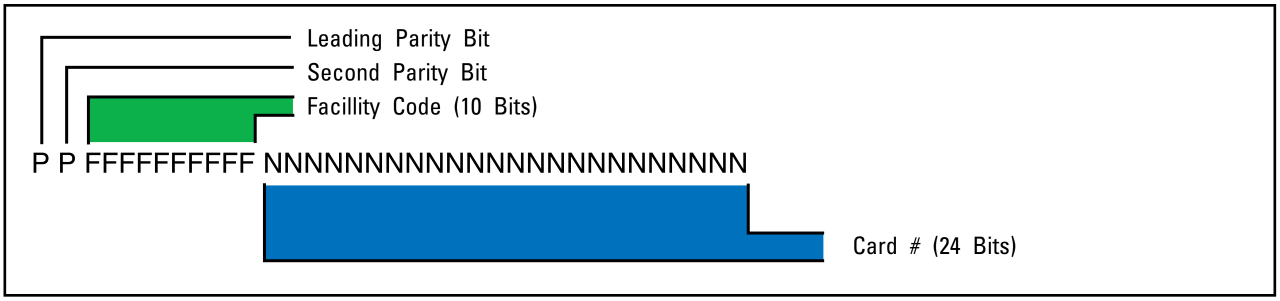


Figure 2: Hypothetical 36-Bit Card Data Format

The Leading Parity bit could relate to one subset of the data string and the Second Parity bit relate to an entirely different subset. This format also has fields named Facility Code and Card Number, but if you compare it to H10301, its format is very different, and would probably not work on a customer's system that was setup for H10301.

The person who creates the format's unique field names has the ability to assign the names as well. Review the following hypothetical format:

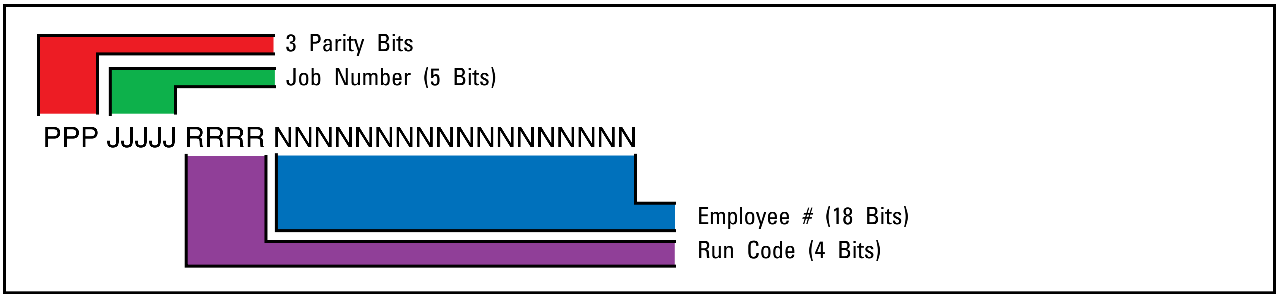


Figure 3: Hypothetical 30-bit Card Data Format

This format has three parity bits, a five-bit programmable field called Job Number, another four-bit programmable field called Run Code, and an 18-bit incrementing field called Employee Number.

When taking information from a customer about their format, it is important to obtain the exact values they want in the programmable fields. The customer, not HID, supplies this information.

Please note that customers often confuse the terms, Facility Code and Site Code. Some formats have a field called Facility Code and others have a Site Code, while others may have neither – or both. You must be certain to utilize the correct terms when ordering your cards.

To avoid duplicating cards that are already in use on a site, customers must know the existing card numbers.

System installers will also need to know the format name, and specific field information in order to setup their security panels and enroll cards. In fact, it is almost impossible to batch enroll cards without these specifics. HID always includes a Cross Reference List with every card order that has the format, and all specific card data listed. HID sales orders and card box labels also include this same information.

© 2005 HID Corporation. All rights reserved. An ASSA ABLOY Group company ASSA ABLOY HID

Corporate 1000

HID offers a unique card data format program named Corporate 1000, where the end user owns the unique format and HID guarantees that the customer's format will not be duplicated. Furthermore, the customer must provide written authorization to HID for an integrator or distributor, etc. to be able to purchase the specified Corporate 1000 cards from HID. This gives the customer absolute control over the manufacture, distribution and delivery of their specific cards. Additional Corporate 1000 facts:

- The customer's access control panels must be capable of decoding the Corporate 1000 format.
- All Corporate 1000 formats are 35 bits long but the data will be arranged in different increments along that 35-bit string. A card number, for example, may be broken into three or more parts and randomly spaced along the string. This ensures uniqueness in every Corporate 1000 format.
- HID currently manages several hundred unique Corporate 1000 formats and many more are still readily available.
- As with all other formats, Corporate 1000 formats are identical in 125 kHz Prox and 13.56 MHz iCLASS contactless smart card technology.

The Wiegand Reader-to-Controller Interface

An interface defines how two devices communicate with one another. Various HID readers can communicate with access control panels using a variety of well-established, industry-standard interfaces including:

- Wiegand
- Serial (RS232, RS422, RS485)
- Clock-and-Data (Magnetic Stripe Track/2) – Also known as ABA format.

We will concentrate on the Wiegand interface because it is the most prominent industry interface for card access control.

The Wiegand interface consists of three conductors (wires) called Data Zero (usually green), Data One (usually white), and Data Return (usually black). When installers obtain an HID reader, they expect to see these three names on the connection points (terminals) for both the reader and the access control panel. All current standard HID reader types are available with a Wiegand interface. The three wires carry Wiegand data, also called the Wiegand signal.

Since the card data is binary, the reader simply receives the radio frequency (RF) data from the card, translates it from RF to Wiegand protocol and sends the complete binary string to the controller. Zeros travel on the green wire, ones on the white wire and the controller combines the two strings of characters into the original set of binary data.

NOTE: The reader performs no processing or quality checking of the data. It simply receives the (RF) data from the card and converts it to Wiegand protocol for immediate transmission to the controller.

Panel Format Settings

Access control panels are built to reject card data that does not conform to a specific pre-defined format. Almost all panels can use the 26-bit standard format (possibly in addition to the manufacturer's own proprietary formats). Simple panels might use only one or two formats, but more sophisticated panels are software configurable and accept virtually all of the different formats. Some can even create customized formats. Once a format is determined, the panel is then configured. Proximity cards must be ordered for use with that configured format and the card must be programmed to conform to the panel's format in order to function.

This one-way flow of setup information from panel to card is intentional. It makes it difficult for an unauthorized person who finds a proximity card to know where and how to use it. Even a person with the technical knowledge and equipment cannot positively identify the card format. The reason is due to the format information residing in the panel and not on the card. A format is not a number — it is a way of looking at a number. The card data merely conforms to the panel format.

© 2005 HID Corporation. All rights reserved.
An ASSA ABLOY Group company ASSA ABLOY HID

There are several essential pieces of information for placing HID card orders.

- Format Name (Example: H10301.cdf) - HID programs cards for our customers using hundreds of existing formats. There is no default format. The customer must always specify to HID which format they want.
- Programmable field information - Looking again at H10301.cdf, it has a Leading one-bit parity field, followed by an eight-bit programmable field, a 16-bit incrementing field and a Trailing Parity bit.

Math Translations – Decimal, Binary and Hexadecimal

While our everyday math is based on the decimal system, computers always use binary math. In binary, each column can only contain a one (1) or zero (0). Binary data is often combined together into convenient four-bit hexadecimal, or hex units called Nybbles. Hex values are displayed as 0 – F.

1	00000001	1
2	00000010	2
3	00000011	3
4	00000100	4
5	00000101	5
6	00000110	6
7	00000111	7
8	00001000	8
9	00001001	9
10	00001010	A
11	00001011	B
12	00001100	C
13	00001101	D
14	00001110	E
15	00001111	F
16	00010000	10
32	00100000	20
64	01000000	40
100	01100100	64

With hexadecimal, 24 bits can be represented with only six characters.
(1111) (1111) (1111) (1111) (1111) (1111) Groups of four bits called Nybbles.
F F F F F F

- A Facility Code of 255 looks like FF in hex (15 x 16) + 15 = 255
- An identification number of 65535 looks like FFFF in hex
  (15 x 4096) + (15 x 256) + (15 x 16) + 15 = 65535

Many panels use hexadecimal math because it is compact, and directly represents binary. HID ProxPro® and MaxiProx® readers can be configured to output data to the panel in hex via RS232 or RS422.

© 2005 HID Corporation. All rights reserved.
An ASSA ABLOY Group company
ASSA ABLOY
HID

APPENDIX F

UIC00A USB ICSP PIC PROGRAMMER

APPENDIX G

PROGRAM

HOME PAGE

```
Private Sub Command1_Click()

Form8.Visible = True
Form6.Visible = False

End Sub
```

```
Private Sub Command2_Click()

Form3.Visible = True
Form6.Visible = False

End Sub
```

```
Private Sub Command3_Click()

End

End Sub
```

CLASS ATTENDANCE

```
Option Explicit
Dim db As Database
Dim rs As Recordset
Dim sql As String
```

```
Private Sub Command1_Click()

Set rs = db.OpenRecordset("masuk", dbOpenDynaset)

rs.AddNew
rs!stud_matrix = Combo2.ItemData(Combo2.ListIndex)
rs!class_id = Combo1.ItemData(Combo1.ListIndex)
rs.Update

Dim temp As Integer
temp = MsgBox("Student has been registered.", vbOKOnly)
```

```
End Sub
```

```
Private Sub Command2_Click()

Form3.Visible = False
Form6.Visible = True

End Sub
```

```
Private Sub Form_Load()

'Initialize the Comm control
MSComm1.CommPort = 4
MSComm1.Settings = "9600,N,8,1"

'Comm event will be triggered when
'a single character is received
MSComm1.RThreshold = 6

'Open the port
MSComm1.PortOpen = True

Set db = OpenDatabase("C:\Documents and Settings\aaa\Desktop\latest project\reg_system.mdb")

Dim i As Integer
Dim sql As String

sql = "select * from class"
Set rs = db.OpenRecordset(sql, dbOpenSnapshot)
i = 1

Combo1.Clear

If rs.RecordCount > 0 Then

   Do Until rs.EOF = True

      Combo1.AddItem rs!class_day
      Combo1.ItemData(Combo1.NewIndex) = rs!class_id
      rs.MoveNext
```

```
        i = i + 1

    Loop

End If

rs.Close

sql = "select * from student"
Set rs = db.OpenRecordset(sql, dbOpenSnapshot)
i = 1
Combo2.Clear

If rs.RecordCount > 0 Then

    Do Until rs.EOF = True

        Combo2.AddItem rs!stud_name
        Combo2.ItemData(Combo2.NewIndex) = rs!stud_matrix
        rs.MoveNext
        i = i + 1

    Loop

End If

rs.Close

End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub
```

```
Private Sub MSComm1_OnComm()

'When a comm event occurs
'Was it a "receive" event? If so, add the received character
'to the text Box and set the insertion point at the end of
'the text. other events are ignored
Dim string1 As String
```

```
'Dim string2 As String
Dim stud_matrix As Long
Dim updateFlag As Boolean

Select Case MSComm1.CommEvent

   Case comEvReceive

   updateFlag = False

   'Text1.Text = MSComm1.Input
   string1 = MSComm1.Input        'new
   ' string1 = Left(string2, 6)
   If (string1 = "1081e10") Then

      stud_matrix = 2007123456
      updateFlag = True

   ElseIf (string1 = "7d71e10") Then

      stud_matrix = 2007234567
      updateFlag = True

   ElseIf (string1 = "8d71e10") Then

      stud_matrix = 2007345678
      updateFlag = True

   Else

      updateFlag = False

   End If

   Dim temp As Integer
   Dim class_id As Integer

   If updateFlag = True Then

      class_id = Combo1.ItemData(Combo1.ListIndex)
```

```
        sql = "select * from masuk where stud_matrix = " & stud_matrix & " and class_id =
" & class_id
        Set rs = db.OpenRecordset(sql, dbOpenSnapshot)

        Dim noRekod As Integer
        noRekod = rs.RecordCount

        If noRekod = 0 Then

            Set rs = db.OpenRecordset("masuk", dbOpenDynaset)

            rs.AddNew
            rs!stud_matrix = stud_matrix
            rs!class_id = Combo1.ItemData(Combo1.ListIndex)
            rs.Update

            sql = "select * from student where stud_matrix = " & stud_matrix
            Set rs = db.OpenRecordset(sql, dbOpenDynaset)
            Dim bilKehadiran As Integer
            bilKehadiran = rs!stud_attended
            bilKehadiran = bilKehadiran + 1

            rs.Edit
            rs!stud_attended = bilKehadiran
            rs.Update

            temp = MsgBox("Student has been registered.", vbOKOnly)

        End If

    Else

        temp = MsgBox("Student does not exist in database.", vbOKOnly)

    End If

End Select

End Sub
```

LOGIN PAGE

```
Private Sub Command1_Click()

If Text1.Text = "rafidah" And Text2.Text = "1234" Then

    Form8.Visible = False
    Form2.Visible = True

End If

End Sub
```

```
Private Sub Command2_Click()

    Form6.Visible = True
    Form2.Visible = False

End Sub
```

MAIN MENU PAGE

```
Private Sub Command1_Click()

Form2.Visible = False
Form9.Visible = True

End Sub
```

```
Private Sub Command2_Click()

Form2.Visible = False
Form4.Visible = True

End Sub
```

```
Private Sub Command3_Click()

Form6.Visible = True
Form2.Visible = False
```

```
End Sub
```

STUDENT PROFILE

```
Option Explicit
Dim db As Database
Dim rs As Recordset
Dim sql As String
Dim temp As Variant
Dim percentDatang As Variant
```

```
Private Sub Command1_Click()

' create sql query to search
sql = "select * from student where stud_matrix = " & Text1.Text
Set rs = db.OpenRecordset(sql, dbOpenSnapshot)

Text8.Text = rs!stud_matrix
Text2.Text = rs!stud_name
Text3.Text = rs!stud_ic
Text4.Text = rs!stud_pcode
Text5.Text = rs!stud_semester
Text6.Text = rs!stud_shouldAttend
Text7.Text = rs!stud_attended
Text10.Text = rs!stud_ccode
rs.Close

If Val(Text6.Text) <> 0 Then

   percentDatang = (Val(Text7.Text) / Val(Text6.Text)) * 100
   Text9.Text = percentDatang

   If percentDatang >= 80 Then

      Text9.ForeColor = vbBlack
      Text13.Text = "ELIGIBLE"
      Text13.ForeColor = vbBlack
      Text2.ForeColor = vbBlack

   Else
```

```
        Text9.ForeColor = vbRed
        Text13.Text = "BAR"
        Text13.ForeColor = vbRed
        Text2.ForeColor = vbRed

    End If

Else

    Text9.Text = ""

End If

End Sub
```

```
Private Sub Command2_Click()

Set rs = db.OpenRecordset(sql, dbOpenDynaset)

rs.Edit

rs!stud_name = Text2.Text
rs!stud_ic = Text3.Text
rs!stud_pcode = Text4.Text
rs!stud_semester = Text5.Text
rs!stud_ccode = Text10.Text
rs!stud_shouldAttend = Val(Text6.Text)
rs!stud_attended = Val(Text7.Text)

rs.Update

rs.Close

Text8.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text10.Text = ""
```

```
Text1.Text = ""
Text1.SetFocus

temp = MsgBox("Update Successful.", vbOKOnly)

End Sub
```

```
Private Sub Command3_Click()

Form9.Visible = False
Form2.Visible = True

End Sub
```

```
Private Sub Form_Load()

Set db = OpenDatabase("C:\Documents and Settings\aaa\Desktop\latest project\reg_system.mdb")

'Text1.SetFocus
Text12.Text = Time
Text11.Text = Format(Date, " dddd, dd mmmm yyyy")

End Sub
```

STUDENT ATTENDANCE LIST

```
Option Explicit

Dim db As Database
Dim rs As Recordset
Dim rs2 As Recordset
Dim sql As String
Dim sql2 As String
```

```
Private Sub Command1_Click()

Dim name2 As String
Dim temp As String
Dim temp1 As String
```

```
Dim temp2 As String
Dim count As Integer

name2 = ""
temp = CStr(Combo1.ItemData(Combo1.ListIndex))

sql = "select stud_matrix from masuk where class_id = " & temp
Set rs = db.OpenRecordset(sql, dbOpenSnapshot)

If rs.RecordCount > 0 Then

   count = 1

   Do Until rs.EOF = True

      temp2 = rs!stud_matrix
      sql2 = "select stud_name from student where stud_matrix = " & temp2
      Set rs2 = db.OpenRecordset(sql2, dbOpenSnapshot)
      temp2 = rs2!stud_name
      name2 = name2 & count & ". " & temp2 & vbCrLf
      rs.MoveNext
      count = count + 1

   Loop

End If
rs.Close
rs2.Close

Text1.Text = name2

End Sub
```

```
Private Sub Command2_Click()

Form4.Visible = False
Form2.Visible = True

End Sub
```

```
Private Sub Form_Load()

Set db = OpenDatabase("C:\Documents and Settings\aaa\Desktop\latest project\reg_system.mdb")

Dim i As Integer

Set rs = db.OpenRecordset("class", dbOpenSnapshot)

i = 1

Combo1.Clear

If rs.RecordCount > 0 Then

   Do Until rs.EOF = True

      Combo1.AddItem rs!class_day
      Combo1.ItemData(Combo1.NewIndex) = rs!class_id
      rs.MoveNext

   Loop

End If

rs.Close
End Sub
```

PIC16F877A SOURCE CODE

```
////////////////////////////////////////////////////////////////////////////////////////////
////                                                                                  ////
////                 The following have special meaning:                              ////
////                   \f  Clear display                                              ////
////                   \n  Go to start of second line                                 ////
////                   \b  Move back one position                                     ////
////                                                                                  ////
////  lcd_gotoxy(x,y) Set write position on LCD (upper left is 1,1)                   ////
////                                                                                  ////
////  lcd_getc(x,y)   Returns character at position x,y on LCD                        ////
////                                                                                  ////
////      JHD 162A LCD                                                                ////
////                                                                                  ////
////    D0  enable                                                                    ////
////    D1  rs                                                                        ////
////    D2  rw                                                                        ////
////    D4  D4                                                                        ////
////    D5  D5                                                                        ////
////    D6  D6                                                                        ////
////    D7  D7                                                                        ////
////                                                                                  ////
////      RFID wiegand reader                                                         ////
////                                                                                  ////
////      pin a0 - d0 (white)                                                         ////
////      pin a1-  d1       (green)                                                   ////
////      pin a2 - buzz (orange)                                                      ////
////                                                                                  ////
////                                                                                  ////
////////////////////////////////////////////////////////////////////////////////////////////

#include <16F877A.h>
#use delay(clock = 20000000)
#use RS232 (baud=9600, xmit=pin_c6, rcv=pin_c7, parity=N)
#include <lcd.c>
#int_ext
#fuses hs, noprotect, nowdt, nolvp

int buffer[3],i,n,x=0,rfidt;
int akey;
```

```
char access[8],temp[8];

void lcd_welcome_disp();
void scankey();
        // Read keypad
void scankey_act();
void RFID_setup();
void RFID_read();
void RFID_switch();

//*****************************************
//main
//*****************************************

void main()
{

        lcd_init();
                //LCD INITIALIZATION
        RFID_setup();

        // display lcd
        lcd_welcome_disp();

        while (1)
        {

                // get switch value
                scankey();

                // wait for swipe
                RFID_read();

                // if swipe, then log
                RFID_switch();

                //delay for 250 miliseconds
                delay_ms(250);
        }
}
```

```
//RFID READER SET-UP
//*******************

	void RFID_setup()
	{
	lcd_init();
	set_tris_a(0b00001011);
	output_low(pin_a2);
	//on buzzer & disable reader
	output_high(pin_a2);						//off
buzzer & enable reader
	}

//LCD Welcoming Display
//*******************
	void lcd_welcome_disp()
	{
	lcd_putc("\fUiTM RFID ACCESS");
	lcd_putc("\n SWIPE 2 PUNCH-IN");
	delay_ms(300);
	}

//key scanning
//*******************
void scankey()
{

	// scan the buttons and update akey
	if(input(PIN_C1))
	{ akey = 1; delay_ms(500); /*while(input(PIN_C1)) { } ;*/ }
	if(input(PIN_C2))
	{ akey = 2; delay_ms(500); /*while(input(PIN_C2)) { } ;*/ }
	if(input(PIN_C3))
	{ akey = 3; delay_ms(500); /*while(input(PIN_C3)) { } ;*/ }

}

void RFID_read()
{
```

```
buffer[0]=0;
//clear buffer address
buffer[1]=0;
buffer[2]=0;

x=0;
        for(i=0; i<26; i++)
//wiegand 26 binary format
                {
                 while(input(pin_a0)==1 && input(pin_a1)==1);

                        if(input(pin_a0)==0)
                        {
                                if(i !=0 && i !=25)
//not start n end bit
                                {
                                  shift_left(buffer,3,0);
                                }
                        }
                if(input(pin_a1)==0)
                {
                        if(i !=0 && i !=25)
                        {
                          shift_left(buffer,3,1);
                        }
                }
                delay_us(400);
//time for rfid reading delay
                }

sprintf(access,"%x%x%x",buffer[2],buffer[1],buffer[0]);//conversion from hex to
character

for(n=0;n<7;n++){temp[n]=access[n];}
}

// RFID SWITCHING
// ******************************************
void RFID_switch()
        {
if(access[5]=='8'&& access[4]=='d'&& access[3]=='7'
```

```
                    && access[2]=='1'&&access[1]=='e'&&access[0]=='1')
                              {

          printf("%c%c%c%c%c%c%d",access[5],access[4],access[3],access[2],access[1],a
ccess[0],akey);
                    lcd_putc("\fNadiah Mohamad"); akey = 0;
                              }

if(access[5]=='7'&& access[4]=='d'&& access[3]=='7'
                    && access[2]=='1'&&access[1]=='e'&&access[0]=='1')
                              {

          printf("%c%c%c%c%c%c%d",access[5],access[4],access[3],access[2],access[1],a
ccess[0],akey);
                    lcd_putc("\fMohd  Firdouz"); akey = 0;
                              }

if(access[5]=='1'&& access[4]=='0'&& access[3]=='8'
                    && access[2]=='1'&&access[1]=='e'&&access[0]=='1')
                              {

          printf("%c%c%c%c%c%c%d",access[5],access[4],access[3],access[2],access[1],a
ccess[0],akey);
                    lcd_putc("\fAiman Johari "); akey = 0;
                              }
          }
```