# MULTIPLE RFID DOORS CONTROL SYSTEM

**SYED MOHD SYUKRIZAL BIN AHMAD KORI**

**UNIVERSITI MALAYSIA PAHANG**

# MULTIPLE RFID DOORS CONTROL SYSTEM

SYED MOHD SYUKRIZAL BIN AHMAD KORI

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

APRIL, 2006

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : _____

Name : NURUL HAZLINA BINTI NOORDIN

Date : 8 MAY 2008

Signature : _____

Author : SYED MOHD SYUKRIZAL BIN AHMAD KORI

Date : 5 MAY 2008

*To my beloved mother and father*

## ACKNOWLEDGEMENT

First of all, I would like to thank my project supervisor Mrs. Nurul Hazlina Binti Noordin, who has given me much strong logistic support while implementing the project given. She has always assisted me when I handling my project. Besides, I would like to express my sincere appreciation for her valuable advises, guidance and encouragement. This has aspired me more confident in trying new things.

Secondly, I would like to thank my family members for giving me their loves and supports throughout my five years of studies in UMP Gambang, Pahang.

Special thanks to staff FKEE, who have given me a great help in accomplishing this project.

At last but not least, I would like to say millions of thanks to all my course mates and those who has lending me their helping hand.

Thank you.

## ABSTRACT

These days the security system is an important part for a house or premise to avoid such incident like robbery. A lot of security systems types have been apply to make sure the safety of the premise. One of it is automatic identification system which really helpful human especially to identify the user ID (identification) of the premise. Radio Frequency Identification (RFID) is a component that suitable for this project. This project was designed and built the system using RFID for access the doors of the premise by indentify the RFID tag user. It then output their description such as ID numbers and user names to an attaches LCD screen. The system is powered by 9 V batteries or using AC TO DC adapter. The main idea in this project is to create a system which that can provide security for the safety of the premise. At the heart of project is a PIC 16F876A microcontroller will be used as a brain of the system that will control the system through a source code (C code) build using MPLab software. In this project, passive tag will be use as an identifier of the user. In the tag will be include RFID tag ID numbers of the user. Every RFID tag is unique because the ID of the tag is different for all the tag. So, it is really safe to use the RFID system for a premise security. Additionally, the PIC will process the data which receive from RFID tools and responsible for sending the output to the LCD screen which will be show the character.

# ABSTRAK

Pada masa kini, sistem keselamatan adalah bahagian yang amat penting bagi sesebuah rumah atau premis untuk mengelak sebarang kejadian yang tidak diingini seperti rompakan. Pelbagai jenis sistem keselamatan telah aplikasikan untuk memastikan keselamatan sesuatu premis. Salah satu daripadanya ialah sistem pengenalpasti automatik yang amat membantu manusia untuk mengenal pasti ID (pengenalan) pengguna sesebuah premis. Pengenalan Frekuensi Radio (RFID) ialah satu komponen yang sangat sesuai untuk projek ini. Projek ini direka dan dibina sistemnya dengan menggunakan RFID melalui pintu-pintu premis dengan mengenal pasti teg RFID pengguna. Ia kemudiannya mempamirkan penerangan seperti nombor ID dan nama pengguna pada skrin LCD. Sistem ini menggunakan 2 biji bateri yang membekalkan kuasa sebanyak 9 V dan disertai 2 suis yang dapat menghidupkan dan mematikan litar dan sistem RFID. Idea utama projek ini adalah untuk mencipta satu sistem yang boleh member keselamatan kepada sesebuah premis. Pengawal mikro digunakan sabagai otak di dalam sistem ini yang akan mengawal sistem melalui satu kod C (kod sumber) yang dibina menggunakan pekakas lembut MPLab. Di dalam projek ini teg pasif akan digunakan sebagai pengenalan pengguna. Di dalam setiap teg mengadungi nombor ID pengguna. Setiap RFID teg adalah unik kerana di dalam setiap teg mempunyai nombor ID yang tersendiri yang berbeza bagi setiap teg. Jadi, ianya amat selamat menggunakan sistem RFID untuk keselamatan sesebuah premis. Tambahan pula, PIC akan memproses data yang diterima dari peralatan-peralatan RFID dan bertanggungjawab menghantar keluaran kepada skrin LCD yang akan ditunjukkan dalam bentuk karektor.

# TABLE OF CONTENTS

## LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| **RFID** | = | Radio Frequency Identification |
| **LCD** | = | Liquid Crystal Display |
| **MHz** | = | Megahertz |
| **GHz** | = | Gigahertz |
| **PIC** | = | Programmable Intelligent Computer |
| **V** | = | Volts |

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF APPENDICES

# CHAPTER 1

## INTRODUCTION

This chapter explains the background of Radio Frequency Identification (RFID) technology Multiple RFID Door Control System. The development of this project entirely depends with the assistance of RFID technology, microcontrollers, DC motors and programming language. The problem statement of this project, the objective of the project and project scopes are also elaborated in this chapter.

Radio Frequency Identification (RFID) technology is expanding rapidly with its applications in a wide range of area. RFID technology consist of RFID reader and RFID tags. There are generally two types of RFID tags: active RFID tags, which contain a battery and thus can transmit its signal autonomously, and passive RFID tags, which have no battery and require an external source to initiate signal transmission. In this project, passive RFID tag is used. This technology has become an intermediate in a wide variety of applications such as in industries as an instrument for identification, in automobile manufacturing and homeland security applications. The primary goal of RFID technology is to automatically identify data that are contained in electromagnetic fields. RFID tags do not require any physical contact with the reader for identification

process. Most RFID tags are inexpensive and small where it derives its power from the signal produced by the RFID reader.

There are various types of microcontroller available nowadays, from various manufacturers, with different functionalities. PIC, MC68HC11 from Motorola, and Basic Stamp are among of the commonly used microcontroller in electronic control devices today. Some applications of microcontroller are controlling the rotation of DC motor including forward and reverse. For simple application of DC motor control the PIC is suitable to used because of it simples and unique features.

## 1.1 Design Objectives

The objective of this project is to design and develop the RFID technology for multiple doors security system where the accesses to all doors are controlled by one main control system.

## 1.2 Project Scope

a. The multiple RFID doors security system is the doors accesses can only be access by the IDs that has been preprogram into the PIC16F876A. For the doors access, the received RFID tag ID will be compared with preprogram ID is using PIC16F876A.

b. For matching ID, it will allow the user to access which door that has been preprogramming for the ID. If the ID (non-user) is not match they will not gain access for the doors system.

# CHAPTER 2

# LITERATURE REVIEW

This chapter discusses elaborately the system designs that have been implemented in the final system.

## 2.1 Radio Frequency Identification (RFID)

RFID is a dedicated short range communication (DSRC) technology. The term RFID is used to describe various technologies that use radio waves to automatically identify people or objects [1]. RFID technology is similar to the bar code identification systems we see in retail stores everyday; however one big difference between RFID and bar code technology is that RFID does not rely on the line-of-sight reading that bar code scanning requires to work [1].

### 2.1.1 RFID History

It's generally said that the roots of radio frequency identification technology can be traced back to World War II [2]. The Germans, Japanese, Americans and British were all using radar—which had been discovered in 1935 by Scottish physicist Sir Robert Alexander Watson-Watt—to warn of approaching planes while they were still miles away [2]. The problem was there was no way to identify which planes belonged to the enemy and which were a country's own pilots returning from a mission [2].

The Germans discovered that if pilots rolled their planes as they returned to base, it would change the radio signal reflected back [2]. This crude method alerted the radar crew on the ground that these were German planes and not Allied aircraft (this is, essentially, the first passive RFID system) [2].

Under Watson-Watt, who headed a secret project, the British developed the first active identify friend or foe (IFF) system [2]. They put a transmitter on each British plane. When it received signals from radar stations on the ground, it began broadcasting a signal back that identified the aircraft as friendly [2]. RFID works on this same basic concept. A signal is sent to a transponder, which wakes up and either reflects back a signal (passive system) or broadcasts a signal (active system) [2].

Advances in radar and RF communications systems continued through the 1950s and 1960s [2]. Scientists and academics in the United States, Europe and Japan did research and presented papers explaining how RF energy could be used to identify objects remotely [2]. Companies began commercializing anti-theft systems that used radio waves to determine whether an item had been paid for or not [2]. Electronic article surveillance tags, which are still used in packaging today, have a 1-bit tag [2].

The bit is either on or off [2]. If someone pays for the item, the bit is turned off, and a person can leave the store [2]. But if the person doesn't pay and tries to walk out of the store, readers at the door detect the tag and sound an alarm [2].

### 2.1.2 The Technology Behind RFID

With RFID, the electromagnetic or electrostatic coupling in the RF (radio frequency) portion of the electromagnetic spectrum is used to transmit signals [1]. An RFID system consists of an antenna and a transceiver, which read the radio frequency and transfer the information to a processing device (reader) and a transponder, or RF tag, which contains the RF circuitry and information to be transmitted [1]. The antenna provides the means for the integrated circuit to transmit its information to the reader that converts the radio waves reflected back from the RFID tag into digital information that can then be passed on to computers that can analyze the data [1].

### 2.1.3 RFID Reader

An RFID reader 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 [3].

A number of factors can affect the distance at which a tag can be read (the read range) [3]. 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].

### 2.1.4 Types Of RFID Tags

In RFID systems, the tags that hold the data are broken down into two different types [1]. Passive tags use the radio frequency from the reader to transmit their signal. Passive tags will generally have their data permanently burned into the tag when it is made, although some can be rewritten [1].

Active tags are much more sophisticated and have on-board battery for power to transmit their data signal over a greater distance and power random access memory (RAM) giving them the ability to store up to 32,000 bytes of data [1].

## 2.2 System Block Diagram

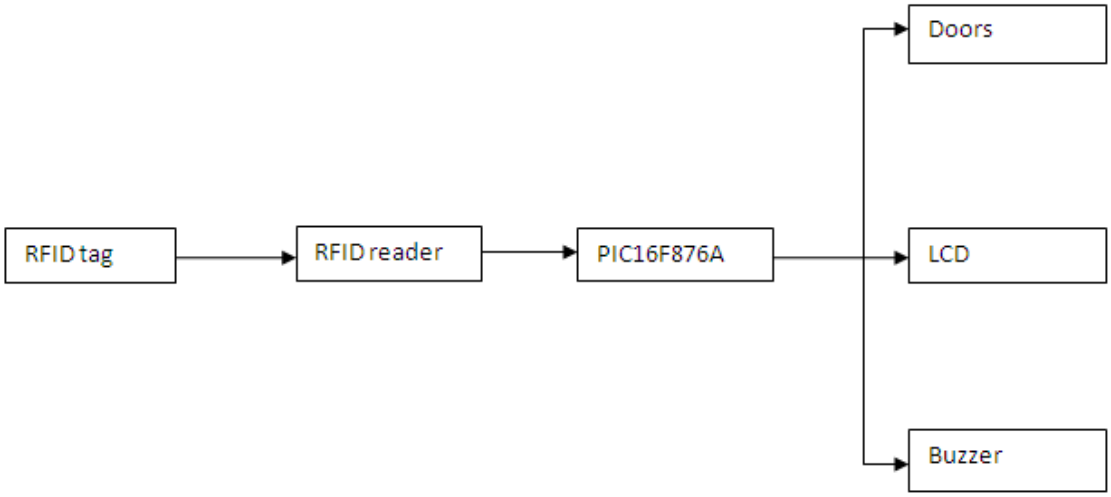



**Figure 2.1:** Block Diagram for Multiple RFID Doors Control System

Figure 2.1 is a simple block diagram of the system. Basically, from the block diagram, this module can be divided into 3 parts. They are RFID transmitting and receiving, microcontroller main circuit, and the output including doors, LCD display and buzzer.

For the RFID transmitting and receiving part, the RFID reader will receive the data (incoming signal) from the passive tag. Then the data will be transmit to the microcontroller main circuit for the comparing the data with the preprogram data.

For the microcontroller main circuit microchip PIC16F876A is being used. This main circuit is the heart of the whole systems where all input and output will be control from this circuit. C programming language is been use for programming design to

control the system. In this part the data that has being receive from the RFID reader will be compare with the preprogram data to control the output.

For the output part, the data that has been compare with the preprogrammed will give access to which doors the user can access. The L293D motor driver and two DC motors will make up for the doors. The buzzer will be the indicator for user or non-user and the LCD display is for displaying the name and ID of the RFID tag user.

# CHAPTER 3

# HARDWARE IMPLEMENTATION

## 3.1 Multiple RFID Doors System

The multiple RFID door system require a microprocessor to process data from RFID reader and to operate the output that has been preprogrammed for each data from the RFID reader. For this reason, microcontroller is the best solution for this task. This is because microcontroller has transmitter and receiver, in this case receiver is need to read the transmitting data from RFID reader. The transmitting and receiving data also happen at the RFID part. The data from RFID tag is transmit to the RFID reader and the reader receive it before transmit it to the microcontroller.

### 3.2 PIC16F876A

Figure 3.1 show the pin diagram of PIC16F876A that will be use as the heart of this project.

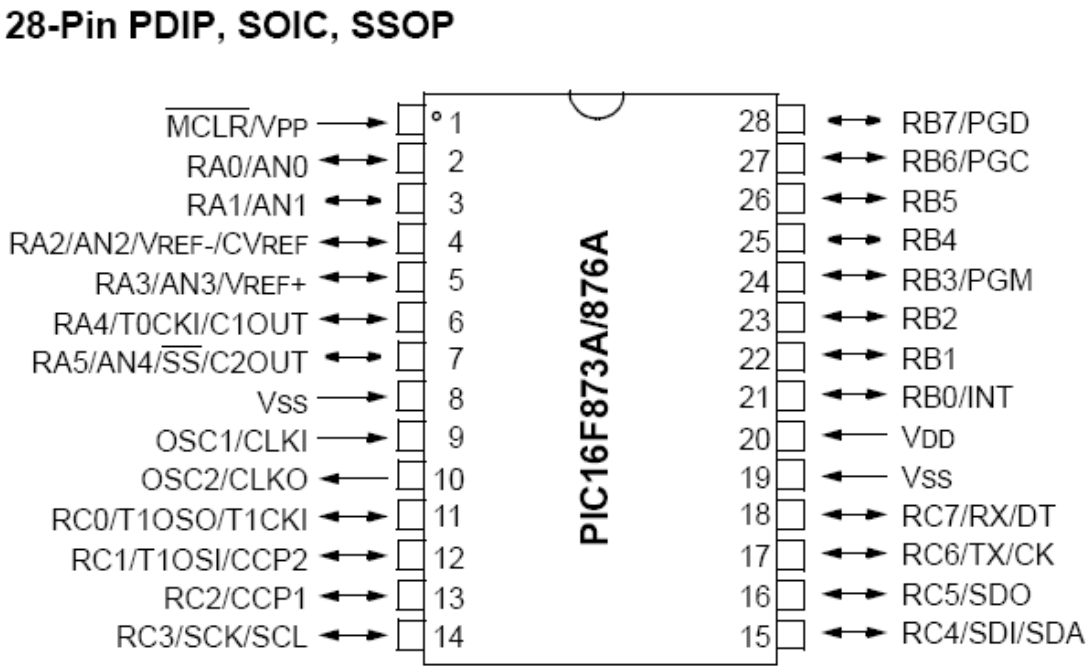



**Figure 3.1:** PIC16F876A Pin Diagram

A PIC16F87XA would be able to do various task. For PIC16F876A version, it has 256 bytes of EEPROM and 368 bytes SRAM while the ROM is disabled. This is a 16-bit microcontroller that have 3 ports that are Port A, Port B and Port C. [4] Several of features are built-in in the PIC16F876A that contain:

- High-Performance RISC CPU
  - ⇨ Only 35 single-word instructions to learn
  - ⇨ All single-cycle instructions except for program branches, which are two-cycle
  - ⇨ Operating speed: DC – 20 MHz clock input
    DC – 200 ns instruction cycle
  - ⇨ Up to 8K x 14 words of Flash Program Memory,
    Up to 368 x 8 bytes of Data Memory (RAM),
    Up to 256 x 8 bytes of EEPROM Data Memory
  - ⇨ Pinout compatible to other 28-pin

- Peripheral Features
  - ⇨ Timer0: 8-bit timer/counter with 8-bit prescaler
  - ⇨ Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
  - ⇨ Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
  - ⇨ Two Capture, Compare, PWM modules
    - Capture is 16-bit, max. resolution is 12.5 ns
    - Compare is 16-bit, max. resolution is 200 ns
    - PWM max. resolution is 10-bit
  - ⇨ Synchronous Serial Port (SSP) with SPI™ (Master mode) and I2C™ (Master/Slave)

- ⇨ Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- ⇨ Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- ⇨ Brown-out detection circuitry for Brown-out Reset (BOR)

- Analog Features
  - ⇨ 10-bit,8-channel Analog-to-Digital
  - ⇨ Converter (A/D)
  - ⇨ Brown-out Reset (BOR)
  - ⇨ Analog Comparator module with:
    - Two analog comparators
    - Programmable on-chip voltage reference (VREF) module
    - Programmable input multiplexing from device inputs and internal voltage reference
    - Comparator outputs are externally accessible

- Special Microcontroller Features
  - ⇨ 100,000 erase/write cycle Enhanced Flash program memory typical
  - ⇨ 1,000,000 erase/write cycle Data EEPROM memory typical
  - ⇨ Data EEPROM Retention > 40 years
  - ⇨ Self-reprogrammable under software control
  - ⇨ In-Circuit Serial Programming™ (ICSP™) via two pins
  - ⇨ Single-supply 5V In-Circuit Serial Programming
  - ⇨ Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
  - ⇨ Programmable code protection
  - ⇨ Power saving Sleep mode
  - ⇨ Selectable oscillator options

- ⇨ In-Circuit Debug (ICD) via two pins

- CMOS Technology:
  - ⇨ Low-power, high-speed Flash/EEPROM technology
  - ⇨ Fully static design
  - ⇨ Wide operating voltage range (2.0V to 5.5V)
  - ⇨ Commercial and Industrial temperature ranges
  - ⇨ Low-power consumption

Figure 3.2 illustrates the architecture of the PIC16F876A port functionality.

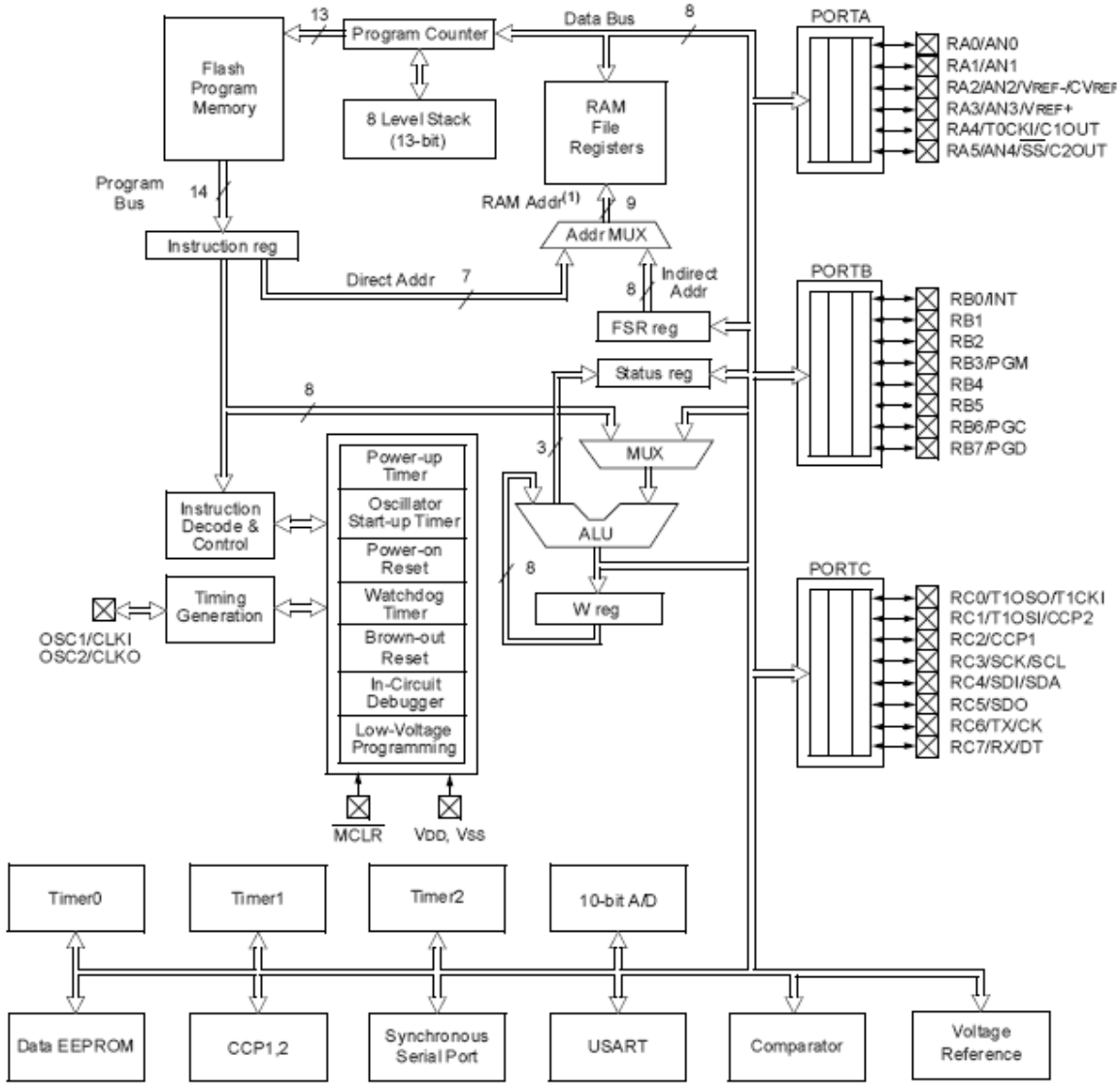



**Figure 3.2:** PIC16F876A Block Diagram

### 3.2.1 PIC16F876A Basic Circuit

Figure 3.3 is the basic circuitry for operating the PIC16F876A with the connection with UIC00A Programmer Connector for the serial communication. UIC00A Programmer Connector is used for the RS-232 line to connect between PC and the PIC16F876A so that the programs from PC can be loaded into the PIC16F876A EEPROM. For the operating speed, 20 MHz input clock crystal is used for this set-up.

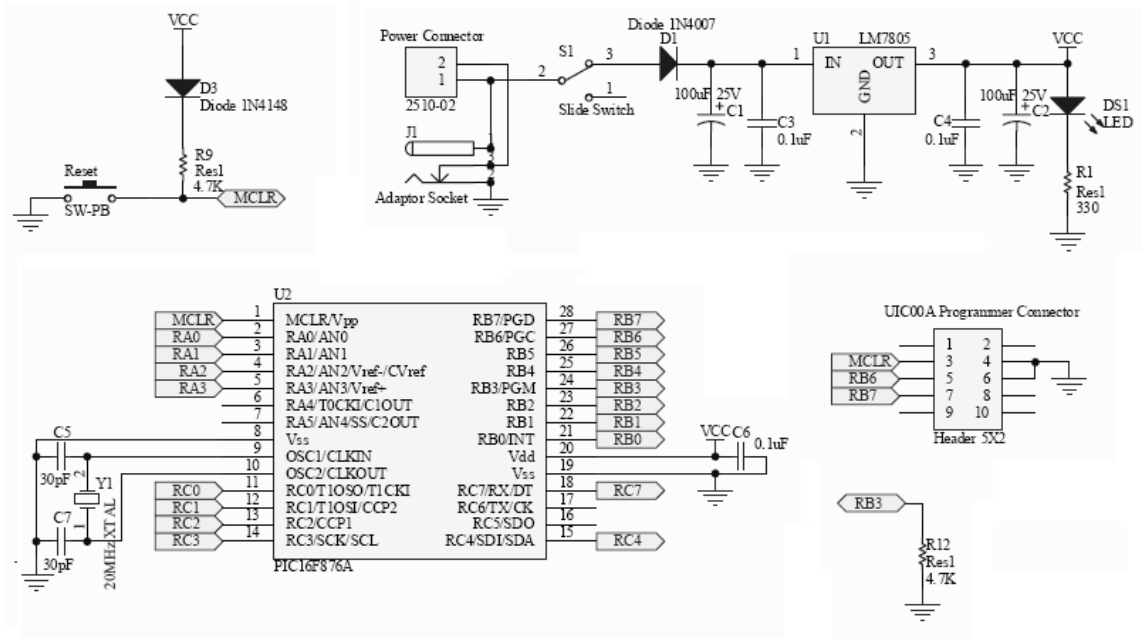



**Figure 3.3:** Basic Circuitry of The Operating PIC16F876A Schematic

## 3.3 Hardware

The hardware portion of the whole system involves various types of modules in order for the system to function or operate. They are the UIC00A Programmer circuit module, the power supply circuit module, the reset circuit module, the outputs circuit module, the DC motor driver circuit module, the LCD display circuit module, the RFID circuit module and the PIC's circuit module itself. All these modules have to be combined and connected into one single circuit to operate the system. The complete circuit of the host system is included in the Appendix A for reference.

### 3.3.1 UIC00A Programmer Circuit Module

Figure 3.4 show the schematic of UIC00A programmer circuit module for loading the program (hex file) into the PIC16F876A.

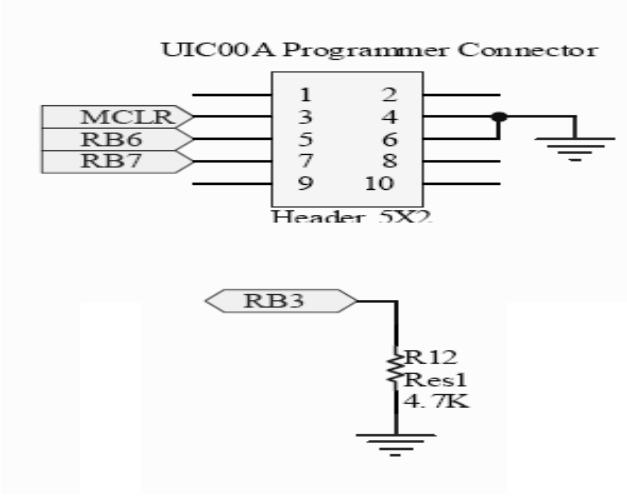



**Figure 3.4:** UIC00A Programmer Circuit Module Schematic

ICSP for programming PIC microcontroller is as shown as in the figure 3.3. MCLR, RB6 and RB7 pin need to be connected to the USB In Circuit Programmer (UIC00A) to program the PIC16F876A. At the same time, RB3 pin need to be pull low to 0V to disable low voltage programming, because the programmer is using high voltage programming. MCLR pin is applied for the PIC programmer goes into programming mode when the programmer are interface with the PIC16F876A.

### 3.3.2 Power Supply Circuit Module

Figure 3.5 shows the schematic of power supply for the circuit.

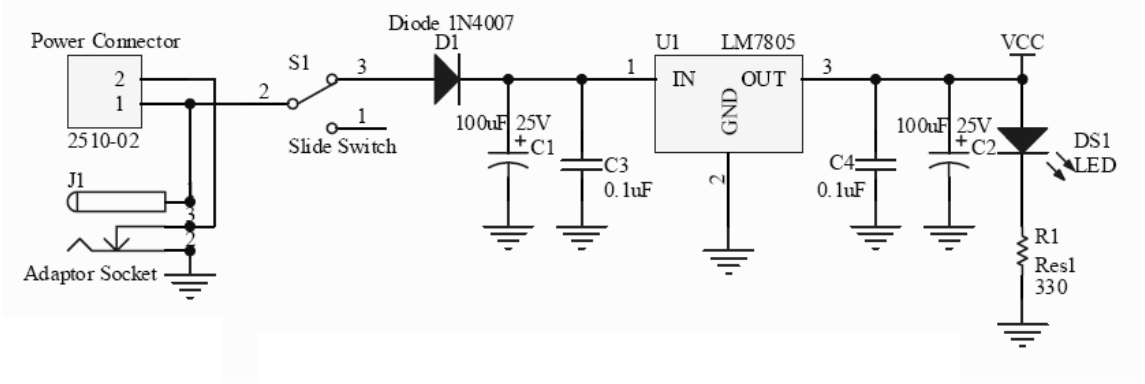



**Figure 3.5:** Power Supply Circuit Module Schematic

This module is use to power up the circuit. The power module supplies a regulated +5V to power up the system. AC to DC or 9V-12V battery can be used to power up the circuit. Higher input voltage will produce more heat at LM7805 voltage regulator. Typical voltage is 12V. Anyhow, LM7805 will still generate some heat at

12V. There are two types of power connector for the circuit, DC plug (J1) and 2510-02 (Power Connector). Normally AC to DC adaptor can be plugged to J1 type connector.

As Figure 3.4 shown, the D2 is use to protect the circuit from wrong polarity supply. C7 and C11 is use to stabilize the voltage at the input side of the LM7805 voltage regulator, while the C8 and C12 is use to stabilize the voltage at the output side of the LM7805 voltage supply. The LED is a green LED to indicate the power status of the circuit. R13 is resistor to protect LED from over current that will burn the LED.

### 3.3.3 Reset Circuit Module

Figure 3.6 show the schematic of reset circuit module. In this part, when reset button is push MCLR pin will active low and clear all the operation of the microcontroller program and back to it initial state. If malfunctions occurs the reset module is important to reset the programming to it initial state. The D3 is use to protect the circuit from wrong polarity supply

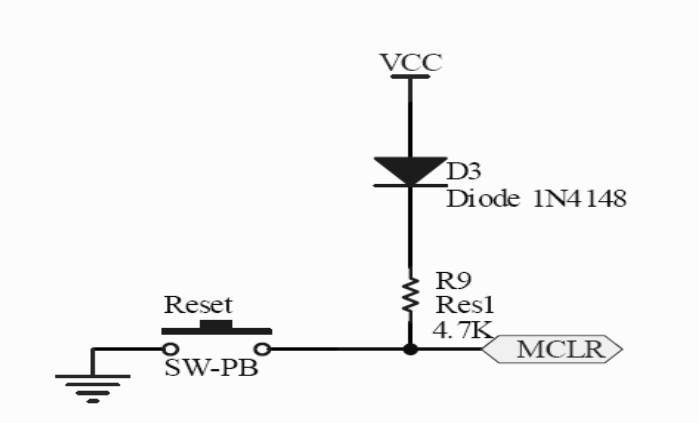



**Figure 3.6:** Reset Circuit Module Schematic

### 3.3.4 Outputs Circuit Module

Figure 3.7 show the schematic of outputs circuit module. DS2, DS3 and LS1 are connected to I/O pin of PIC16F876A as outputs. DS2, DS3 and LS1 are respectively connected to RA2, RA3 and RC0 pin. The function of R10 and R11 are to protect from over-current that will burn the LED. When the output is in logic 1, the LED will ON, while when the output is in logic 0, the LED will OFF. Also for buzzer, when the output is in logic 1, the buzzer will activate (beep), while when the output is in logic 0, the buzzer will deactivate.

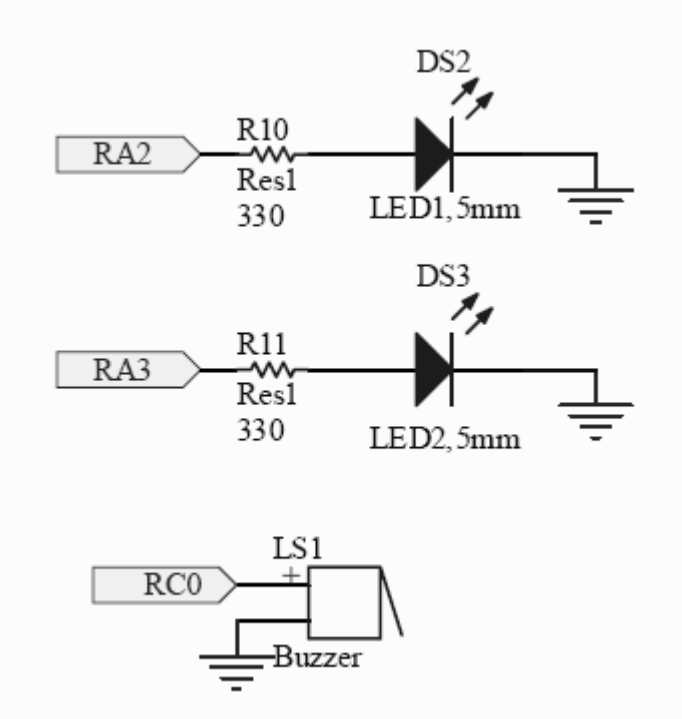



**Figure 3.7:** Outputs Circuit Module Schematic

### 3.3.5 DC Motor Driver Circuit Module

Figure 3.8 show the schematic of DC motor driver circuit module. L293D is used for this circuit. To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included as shown in the block diagram in figure 3.9. This device is suitable for use in switching applications at frequencies up to 5 kHz. The I/O pin of PIC16F876A is assigned as input for the driver motor to move the motors for operation forward and reverse of the DC motors. RA2 is used to enable the motor driver

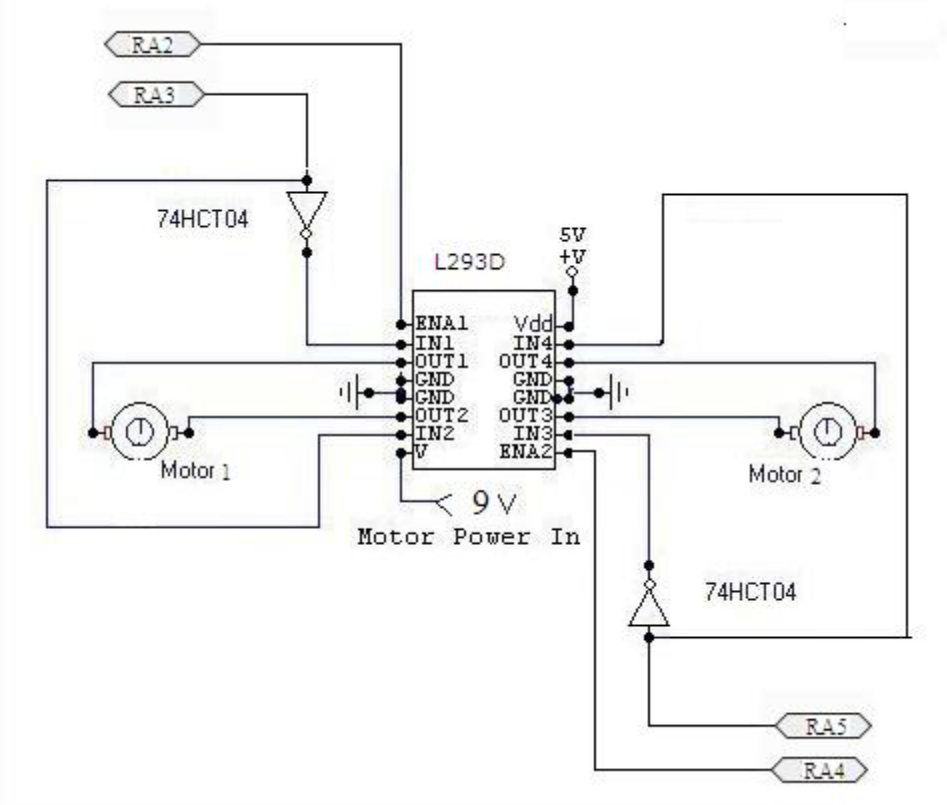



**Figure 3.8:** DC Motor Driver Circuit Schematic

The I/O pin of PIC16F876A is assigned as input for the driver motor to move the motors for operation forward and reverse of the DC motors. RA2 and RA4 are used to enable the motor driver and also to move the DC motor forward. The usage of 74HCT04 is to invert the logic from the I/O pin of PIC16F876A. The logic symbol of the 74HCT04 is as shown in Figure 3.10. When the RA2 and RA4 are in logic 1 the motor driver will enable and the motor will rotate forward mode if the RA3 and RA5 are in logic 0. And motor will rotate reverse mode if the logic is of RA3 and RA5 is 1. When the logic give to enable pin L293D the DC motor will stop.

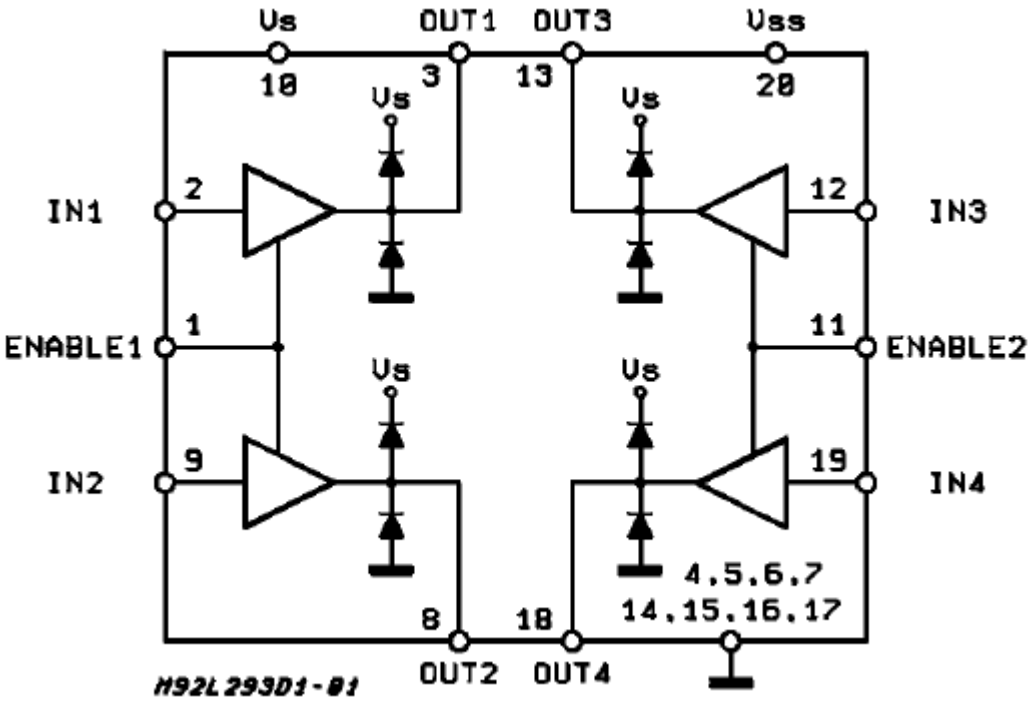



**Figure 3.9:** L293D Block Diagram

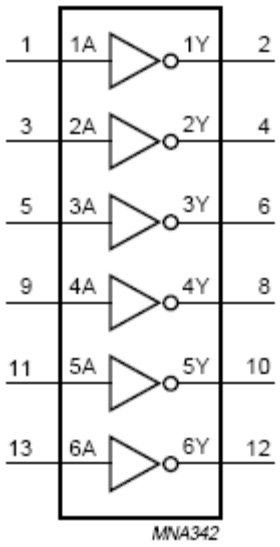



**Figure 3.10:** L293D Logic Symbol

### 3.3.6 LCD Display Circuit Module

Figure 3.11 show the schematic of LCD display module circuit.

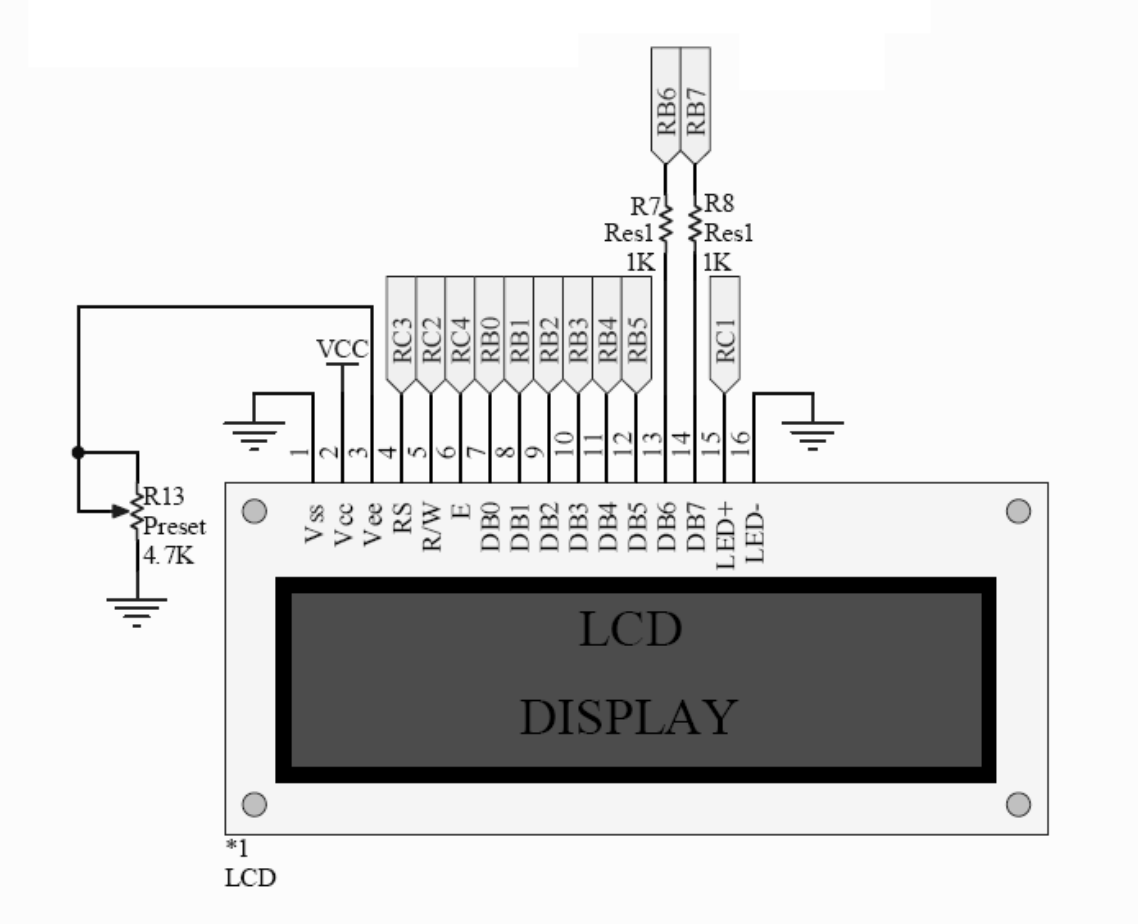



**Figure 3.11:** LCD Display Circuit Module schematic

The LCD used for this project is a 2x16 character LCD. Vss and $V_{EE}$ pins of the LCD are connected to ground while pin $V_{DD}$ is connected to power supply of +5V. Pins RC2, RC3 and RC4 of microcontroller are connected to the select read or write (RW), select register (RS) and enable (E) pins of LCD respectively. As operate the LCD in 8-bits mode, pins RD0, RD1, RD2, RD3, RD4, RD5, RD6 and RD7 of LCD are connected to microcontroller pins RB0, RB1, RB2, RB3, RB4, RB5, RB6 and RB7 respectively.

Pins 15 and 16 of LCD represents for backlight positive input and backlight negative input. Thus, pin 15 is connected to RC1 and pin 16 is grounded.

### 3.3.7 RFID Circuit Module

Figure 3.12 show the schematic of RFID circuit module. In this module RFID reader is interfacing with the PIC16F876A. The RFID reader comes with a serial port (DB9) for data communication and a PS2 connector to supply 5V for the reader. There are 5 output wire of RFID reader, only 4 are necessary. Table 3.1 shows the connection of the output wire.

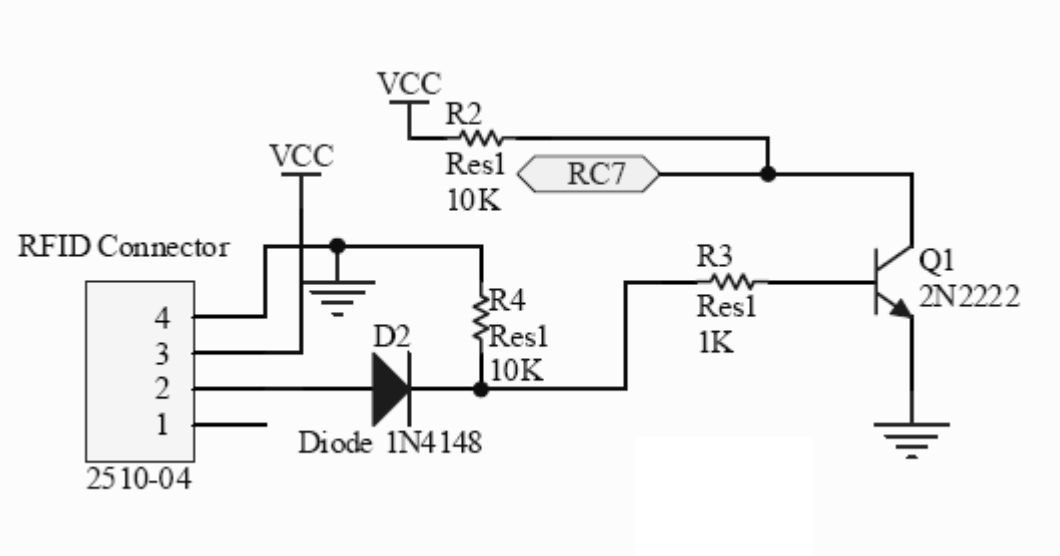



**Figure 3.12:** RFID Circuit Module Schematic

The output of the RFID reader is serial UART in logic +10V/-10V, and the baud rate is 9600bps. The Figure 3.12 shown is used to convert +10V/-10V logic to +5V/0V logic for PIC microcontroller.

**Table 3.1:** RFID Output Wire Connection

| **Colour** | **Pin function** | **Connection** |
|---|---|---|
| Yellow | RX | Not connected |
| Green | TX | RC7 |
| Red | VCC (5V) | 5V |
| Brown | Ground | GND |
| Shielding wire | Not connected | Not connected |

### 3.3.8 PIC16F876A Circuit Module

Figure 3.13 show the schematic of PIC16F876A module circuit.

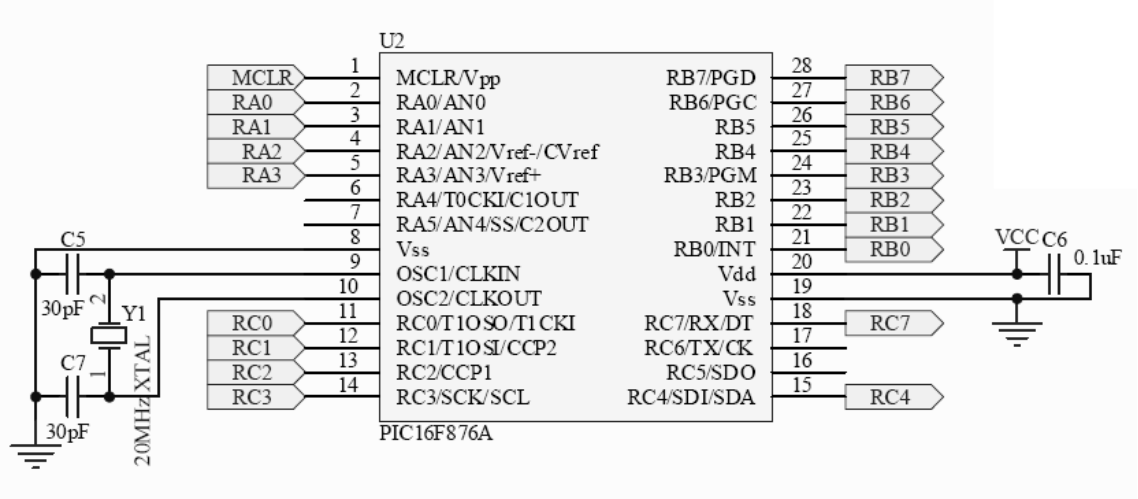



**Figure 3.13:** PIC16F876A Circuit Module Schematic

All pins of the PIC microcontroller is connected to its corresponding modules. The clock section governs all system timing and thus is really responsible for the proper operation of all system hardware. The clock section usually consists of a crystal oscillator and clock circuitry set up to operate the processor at its specified clock rate. Using a high frequency crystal oscillator and dividing it down to a lower frequency provides for greater stability. For the clock section a 20 MHz crystal is used which will be divided by microcontroller to operate at 5MHz frequency room temperature.

# CHAPTER 4

# SOFTWARE IMPLEMENTATION

## 4.1 Software Overview And Design

In designing the functionality of the project, a programming has been made to be load to the PIC16F876A in C programming language using In Circuit Serial Programmer, (ICSP). The programming is about comparing data that was received from the RFID system with the preprogram data in the PIC16F876A. When the RFID tag is place on the reader, the reader will read the data and send in to PIC16876A to compare the data with the pre program data. This the places where the data will be verify. If the data is match, the tag user can access which doors has been assign for the data and the ID code and user name will display on the LCD display. Meanwhile if the data is not matched, the tag user cannot access any doors and it will display “user not found” on the LCD display. Figure 4.1 shows the software flow chart for this system.

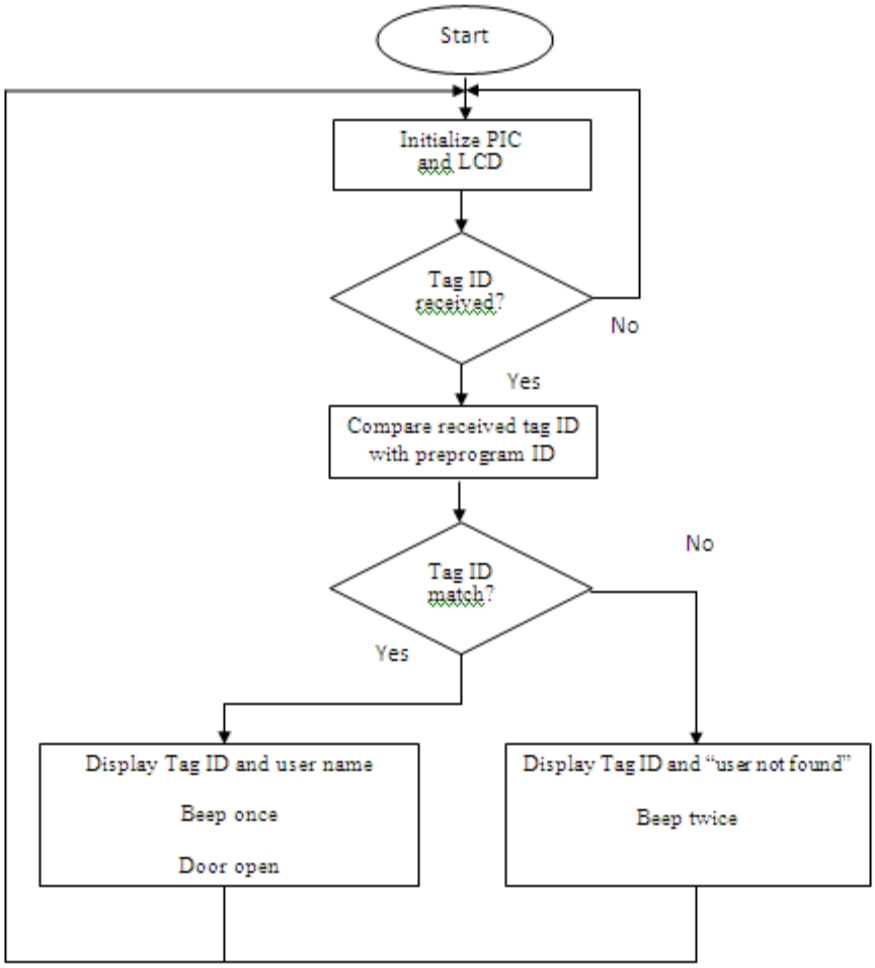



**Figure 4.1**: Software Flow Chart of Multiple RFID Door Control System

## 4.2 C Programming Language

C programming language is a computer programming language that creates lists of instructions for a computer or a microcontroller to follow. C is one of thousands of programming languages currently in use. C has been around for several decades and has won widespread acceptance because it gives programmers maximum control and efficiency.

### 4.2.1 C Programming Language Compilation

To uploading the programming, these source codes need to be compiling first. For this project the source code is made using MPLAB software to compile it HEX file and other component. The HEX file is the one that will load into the PIC16F876A.

### 4.2.2 Source Code To Microcontroller

UIC00A is a PIC-USB programmer. It is designed to program the microcontrollers. On board ICSP (In Circuit Serial Programming) connector offers flexible methods to load program. It supports on board programming which eliminate the frustration of plug-in and plug-out programming of PICs. This allows the microcontroller to be quickly programmed and debugged the source code while the target PIC on the board itself. Since USB port have become a popular and widely used

on laptops and desktop PCs, UIC00A is available for plug and play with USB connection. This programmer obtain its power directly from USB connection, thus no external power supply is required.

To load the program a software call PICkit 2 Programmer is used. The program must be in HEX file. By launching the PICkit 2 Programmer, the HEX file can be writing into the PIC16F876A. The following programming interface appears as shown in Figure 4.2 and notifies that the PICkit 2 programmer the target device is found and connected.

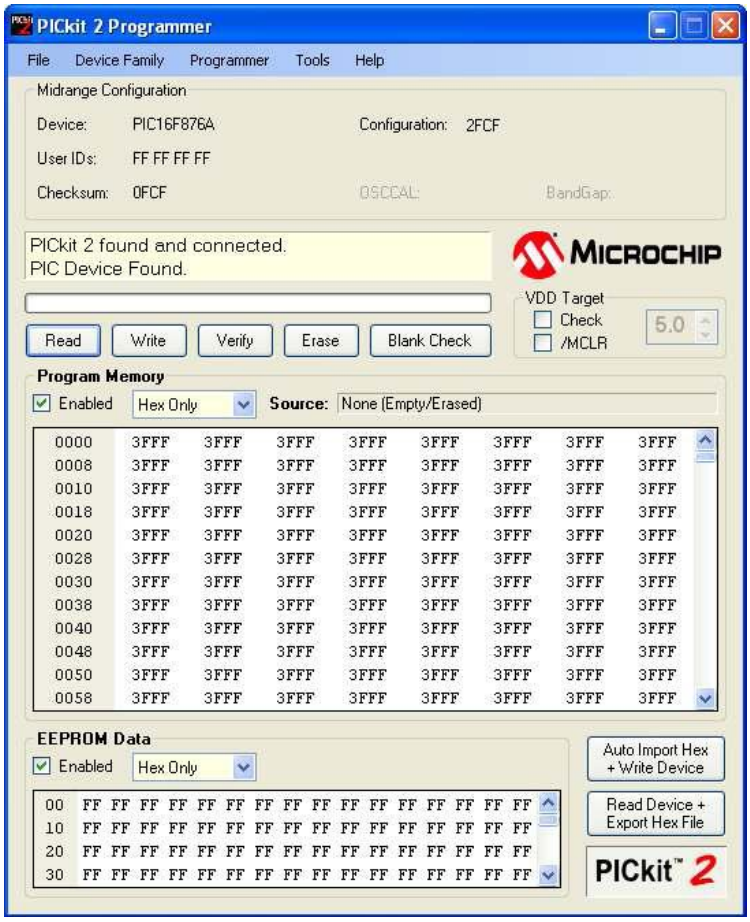



**Figure 4.2:** PICkit 2 Programmer Interface

First of all, erasing data in microcontroller is the first thing to do and the blank check is executed. And then for load the HEX file, the HEX file is import from PC and then executes write operation of the PICkit 2 programmer. The operation status will be displayed and the status bar will turn green if writing is successful as shown in Figure 4.3.

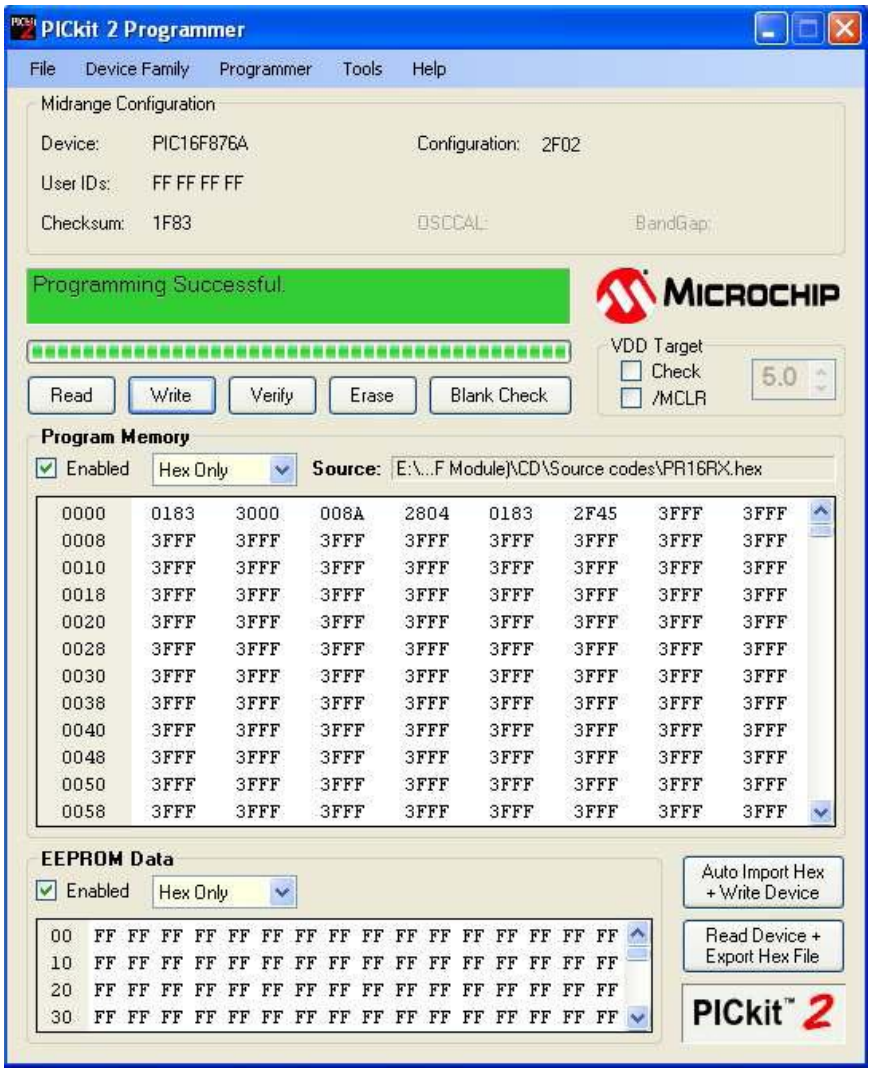



**Figure 4.3:** PICkit 2 Programmer Interface Indication

## 4.3 C Programming Language Design

The C programming language consists of output on LCD, receiving data from RFID reader and comparing the data. The full source code for the project is presented in Appendix B.

### 4.3.1 Output On LCD Display

The first output of the system is an LCD display that will display the information about the system. The LCD display that is used for this project is 16 bit LCD display mean the output will display in two lines on the LCD display. The LCD display will display respectively for the two lines and phrases such as “RFID door’’ for the first line and “security” for the second line, “Place Your ID” the first line and “on the Reader” the second line and “processing…..” the first line. In order for the LCD display to display the stated phrases, the PIC should be programmed with appropriate C programming language. The source code is shown in Figure 4.4.

```
.
.
.
lcd_goto(0);
send_string("   RFID door");
lcd_goto(20);
send_string(" security");
.
.
.
lcd_goto(0);
send_string("Place your ID");
lcd_goto(20);
send_string("on the reader.");
.
.
.
lcd_goto(20);
send_string("Processing.....");
```

**Figure 4.4:** C Code of LCD's Output

### 4.3.2 Receiving Data From RFID Reader

The main operation of this project is to receive data from RFID reader and compare the data with the preprogram data in the PIC16F876A. Therefore, the incoming data from RFID reader is very important as this will assist to proceed to other operations such as data comparison. If data cannot be received from RFID reader, the whole system will not function. Thus, for effectively receive data from the RFID reader; source code shown in Figure 4.5 is used.

```
for(i=0;i<10;i+=1)data[i]=uart_rec();
```

**Figure 4.5:** C Code to Receive RFID Data

The source code shown above in Figure 4.5 explains that "for(i=0;i<10,i+=1)" ten characters should be received from the reader where the ten characters represent a code. The "i" are characters and the command "data[i]=uart_rec()" represents the incoming data from RFID reader.

### 4.3.3 Comparing The Data

After data have been received from the RFID reader, the data should be compared with the preprogram data in the PIC16F876A to verify whether the ID represents a user or non user. If a user is identified, the user's name will be displayed in the LCD display and if a non-user is identified as "user not found" will be displayed. Figure 4.6 shows the source code for unsigned character ID and user name of multiple users for the RFID tag that will be compare with the data receive from the RFID reader.

```
unsigned char data[10];

unsigned char id_1[10]={"0008349791"};
unsigned char id_2[10]={"0008353275"};

unsigned char user_1[10]={"SYUKRIZAL"};
unsigned char user_2[10]={"HASLAN    "};
```

**Figure 4.6:** Source Code of Unsigned Character ID and User Name

Figure 4.6 shows the unsigned character ID and user name of to user for the Multiple RFID Door Control System. The 1st user is owner of the RFID tag with ID 0008349791and name of SYUKRIZAL and 2nd RFID tag owner with ID 0008353275 and name HASLAN.

Figure 4.6.1 show the source code of data comparison for 1st RFID tag owner. If the data match with the unsigned character ID for the 1st owner of the RFID tag as shown in figure 4.6, the ID number and the user name will display on the LCD display. The command "send_string("ID: ") is display "ID: " and "(for(i=0;i<10;i+=1)send_char(id_1[i])" is display the tag ID on the first line of the LCD display. For example "ID: 0008349791". Meanwhile for the command "send_string("user: ") is display "user: " and for "(for(i=0;i<10;i+=1)send_char(user_1[i])" is display the user name on the second line of LCD display. For example "user: SYUKRIZAL". This process is also the same for the 2nd owner comparison part as shown in figure 4.6.2.

```
{
case 1:
                    .
                    .
                    .
                    lcd_goto(0);
                    send_string("ID: ");

                    for(i=0;i<10;i+=1)send_char(id_1[i]);
                    lcd_goto(20);
                    send_string("user: ");

                    for(i=0;i<10;i+=1)send_char(user_1[i]);
                    .
                    .
                    .
                    break;
```

**Figure 4.6.1:** Source Code of Data Comparison for 1st RFID Tag Owner

```
case 2:
                .
                .
                .
                lcd_goto(0);
                send_string("ID: ");

                for(i=0;i<10;i+=1)send_char(id_2[i]);
                lcd_goto(20);
                send_string("user: ");

                for(i=0;i<10;i+=1)send_char(user_2[i]);
                .
                .
                .
                break;
```

**Figure 4.6.2:** Source Code of Data Comparison for 2nd RFID Tag Owner

Figure 4.6.3 show the Source code of data comparison for non-user. The command "send_string("ID: ") is display "ID: " and "send_string("user not found"). If the data that received from RFID reader is not match it will display "ID: user not found".

```
default:
                lcd_goto(0);
                send_string("ID: ");

                for(i=0;i<10;i+=1)send_char(data[i]);
                lcd_goto(20);
                send_string("user not found");
                .
                .
                .
                break;
```

**Figure 4.6.3:** Source Code of Data Comparison for non-user

# CHAPTER 5

# RESULTS

## 5.1 Introduction

Extensive testing has been performed part by part on the developed system and the approach taken in executing the various tests are discussed in this chapter.

In multiple RFID door control system, various tests are conducted. The tests include PIC16F876A module test, RFID module test, DC motor driver circuit module test and LCD display circuit module test. These tests conducted to make sure all the systems part function well due to most of the part system are interact each other. Each module is integrated into a single system to produce a fully functioning temperature control system.

## 5.2 PIC16F876A Circuit Module Test

As been mentioned before, this project is using PIC16F876A. The connection for this module is as shown by Figure 5.1.

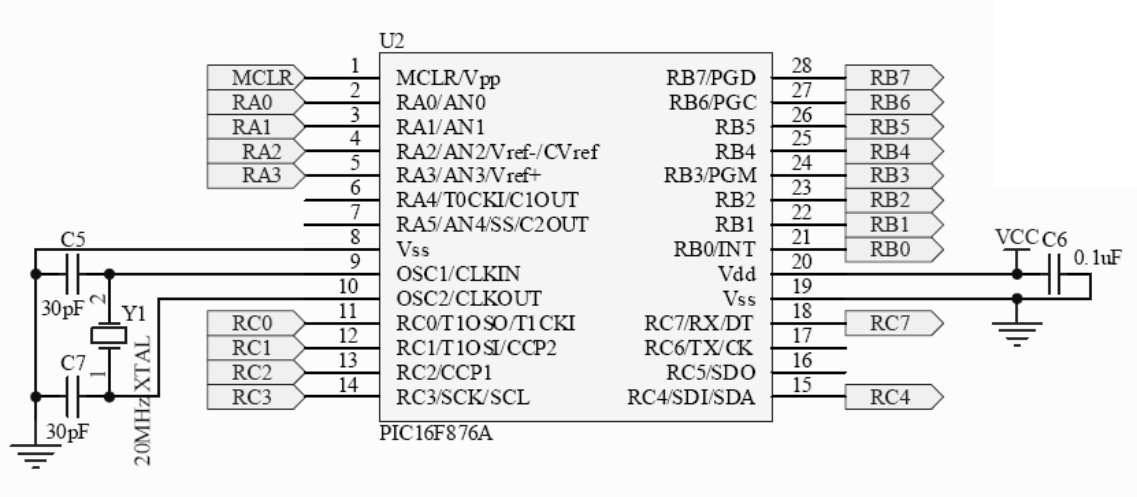



**Figure 5.1:** PIC16F876A Circuit Module Schematic

The power supply for this schematic is 5V that provided by the 7805 voltage regulator which its input voltage is supplied from the battery. The reset pin of the PIC is normally set to logic high because it is active-low. When the reset button is pressed, reset pin will be set to logic low and it will be activated. The crystal used in this project is 20MHz. The clock for the PIC is quarter of the frequency of the crystal that being used, that is for this project, the clock of PIC is 5MHz. Thus the instruction cycle for each is 200 ns. UIC00A Programmer Connector is used for the purpose of programming the built-in EEPROM. The pin VDD is connected to 5V power supply and the pin Vss is connected to ground to in order to use the ADC. Thus, the result of this module reveals that the controller is operating properly.

## 5.3 RFID Circuit Module Test

A simple manual test is performed for test the functionality of the RFID tag and RFID reader. Figure 5.2 show the connection of the PIC16F876A circuit module. 5V power supply is supply to pin 3 of the RFID connector. Pin 4 of the connector is grounded. Pin 2 is for transmit the data from reader to PIC. For testing the functionality of the RFID, when the power supply is on the LED on the RFID reader is on with green colour. When RFID tag is put near or on the RFID reader the LED turn red. This mean there is transmitting data process from RFID tag to the RFID reader. So, as the result, the RFID circuit module is working properly.

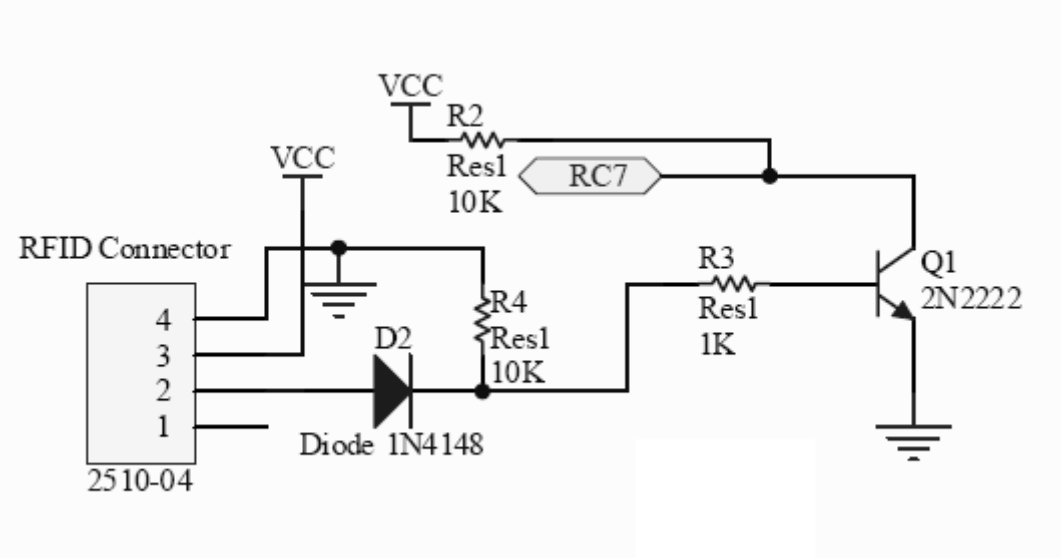



**Figure 5.2:** PIC16F876A Circuit Module Schematic

## 5.4 DC motor driver circuit module test

Figure 5.3 show the connection of the DC motor driver circuit for the project. A simple programming has been made to test the driver circuit. When the pin RA2 and RA3 is given by logic "1" the driver is activate and both motor are rotating forward. This is because RA3 and RA5 at this time is at logic "0". So the inverter will turn pin IN1 and IN3 to logic "1" and IN2 and IN4 to logic "0". Other wise the if the enable input pin RA2 and RA3 is given by logic "1" and the inverter input RA3 and RA5 is also set to logic "1" the motor will rotate in reverse mode. And if enable input is set to logic "0" the driver will deactivate and all motor operation is stop immediately.

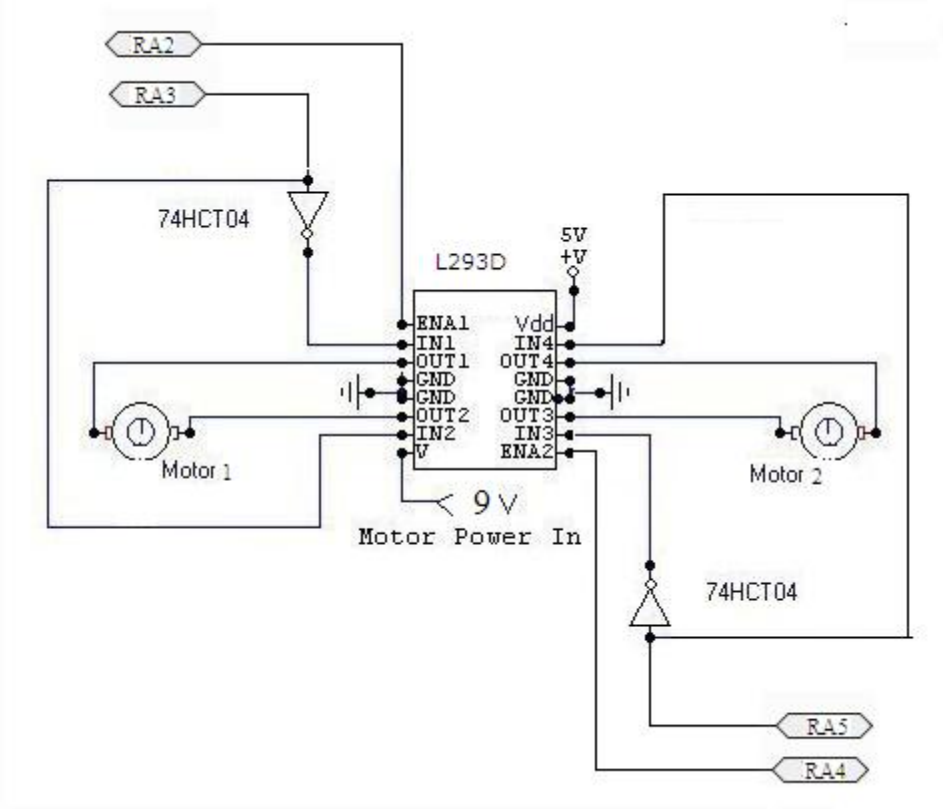



**Figure 5.3:** DC Motor Driver Circuit Schematic

## 5.5 LCD display circuit module test

In order to test the LCD functionality, the circuit as shown in Figure 5.4 is connected. Then, a simple program to display your name is made and load to the PIC16F876A. When the program is executed, the LCD will display your name. Therefore, it is concluded that LCD module testing is successfully completed and can be used for display purposes.

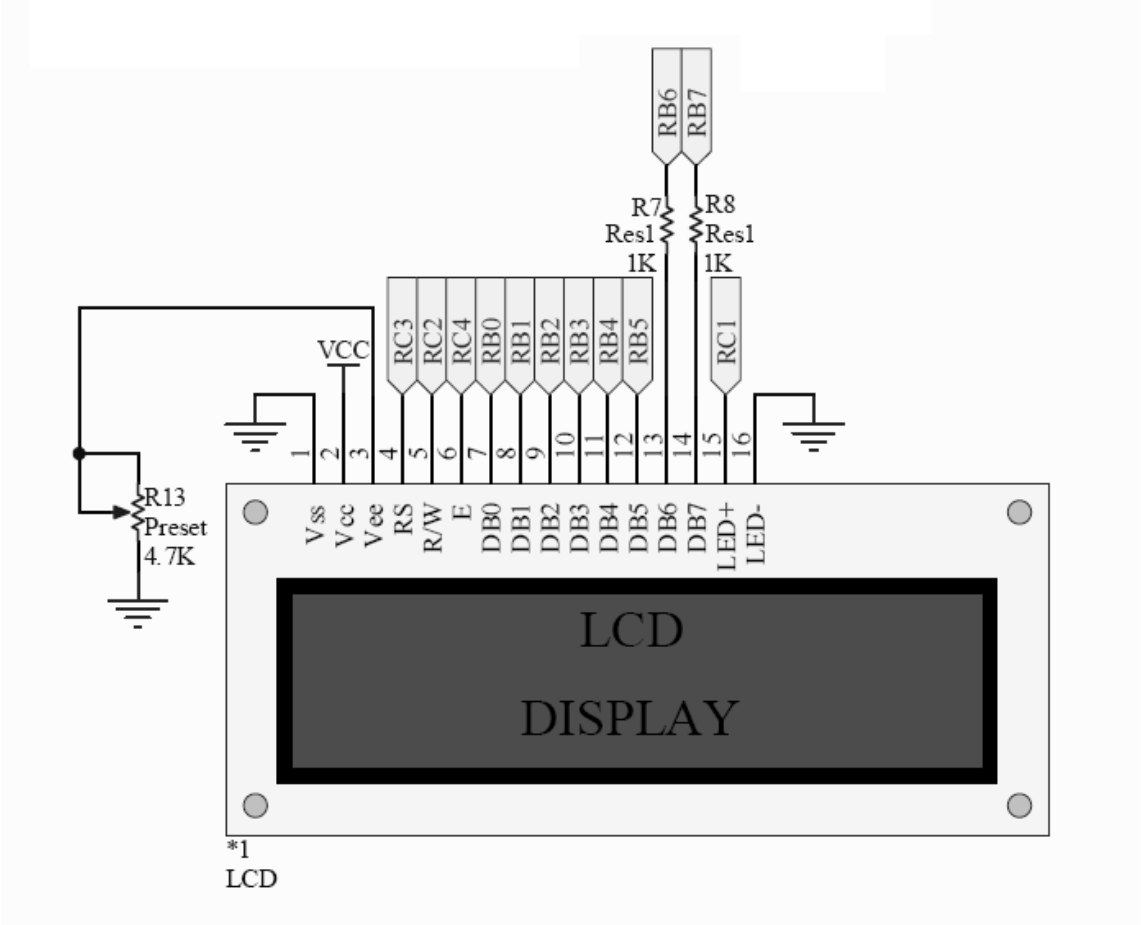



**Figure 5.4:** LCD Display Circuit Module schematic

## 5.6 Hardware Result

When the system is on with 5V power supply, he LCD display displays the welcoming note like “RFID security door”. Then, “Place Your ID on the Reader” will be displayed as shown in Figure 5.5.


**Figure 5.5:** Welcoming Note Output on LCD display

When the identity card is placed near the RFID reader, the card's ID number the user name is read and displayed on LCD display as shown in Figure 5.5.1, Figure 5.5.2 and Figure 5.5.3

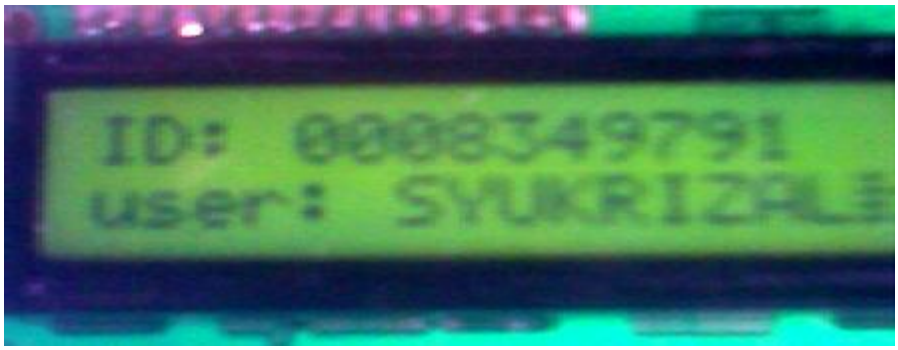



**Figure 5.5.1:** Identification of The 1st Owner tag

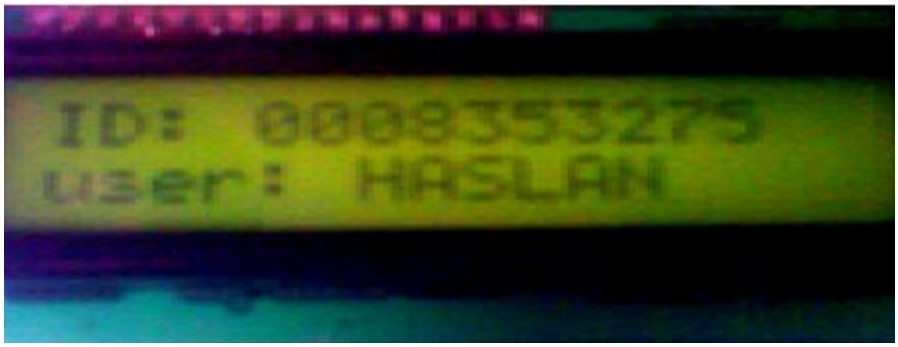



**Figure 5.5.2:** Identification of The 1st Owner tag

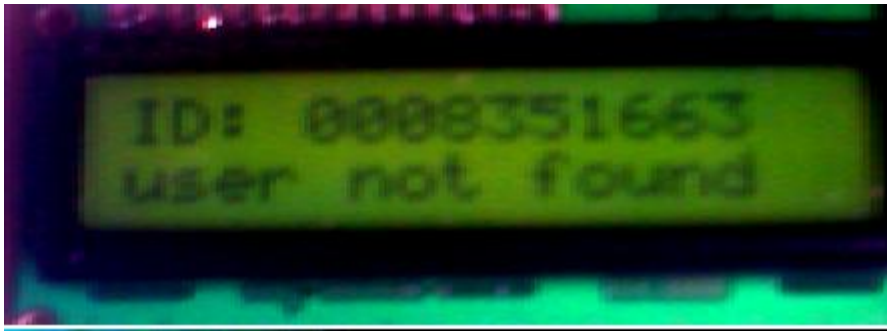



**Figure 5.5.3:** Identification of The Non-user

# CHAPTER 6

# CONCLUSION AND FUTURE RECOMMENDATIONS

## 6.1 Conclusion

This project is fully completed and but the objectives are not accomplished. The output of the project is LED, LCD display and DC motors. For the LED part, it was set on the 1st DC motor. For forward mode, the LED1 is on as indicator of enable pin of the DC motor driver. For reverse mode, the LED1 is on to active the DC motor driver and LED2 is on to show the process of reverse mode. For the LCD display, the result is successful accomplish. The output is display properly the data that has been program to the PIC16F876A either for welcome note or the data receive from the RFID reader. There is two DC motor where used but only 1 is working properly. This is due to the unwanted noise occurs to the DC driver part when using only one power supply for the whole system.

RFID system is suitable for security system because of RFID tag unique function where each tag has it own ID number. Only the ID that has been program can be use for this project to open doors. So these mean other tags that have not been program cannot open any doors. It is really safe security method that should been apply to our house or premises. Through RFID technology, appropriate architecture design, PIC16F876A and C Language Programming, the new world of automatic identification technology may help us increased security and reducing human intervention in the security field.

## 6.2 Future Recommendations

This project has successfully demonstrated a RFID system communication for doors control system. Future work on this project may include:

- Interface with personal computer to record the ID and time so at the same time it can be used as punch card.

- Set clock on the project to show the time the user access the door and record it automatically to personal computer.

- Installed with voice recorder to verify the user or non-user.

- Interface the security door system with the CCTV system and record each user and time of the access to personal computer.

## 6.2 Costing and Commercialization

The overall cost of the multiple RFID doors control system is around RM 309.36 per unit. The approximate costing is as shown in Table 6.1. However, if it is produced in a large amount, the cost will be reduced greatly. As this system can be used in large security scope of area, thus it has the potential to be commercialized especially for premises, houses or other building that needs high security level.

Furthermore, this project can be add on more future such as interface the security door system with the CCTV system and record each user and time of the access to personal computer for more higher security level or interface with personal computer that will record all the accessing of the premises or building.

**Table.6.1:** Approximate Cost for Multiple RFID Doors Control System

| Component | Specification | Quantity | Estimated cost |
|---|---|---|---|
| PIC16F876A | 28 PINS | 1 | RM25.00 |
| 2X16 LCD DISPLAY | | 1 | RM 30.00 |
| DC Motor | | 2 | RM 18.00 |
| ADAPTER SOCKET(DC) | | 1 | RM 0.50 |
| BUZZER | 4.7KΩ | 1 | RM 2.00 |
| DIODE | IN4007 | 1 | RM 0.10 |
| DIODE | IN4148 | 2 | RM 0.20 |
| LED | GREEN | 1 | RM 0.13 |
| LED | RED | 2 | RM 0.26 |
| CRYSTAL | 20 MHz | 2 | RM 1.00 |
| CAPACITOR | 22pF | 6 | RM 0.42 |
| CAPACITOR | 0.1uF | 2 | RM 0.36 |
| CAPACITOR | 100uF | 3 | RM 0.20 |
| VOLTAGE REGULATOR | LM7805 | 2 | RM 2.00 |
| PUSH BUTTON | | 2 | RM 1.00 |
| RESISTOR | 330Ω | 5 | RM 0.30 |
| RESISTOR | 4.7kΩ | 4 | RM 0.24 |
| RESISTOR | 1kΩ | 3 | RM 0.18 |
| RESISTOR | 10kΩ | 2 | RM 0.12 |
| IC BASE | 28 PINS | 1 | RM 0.35 |
| SCREWS & NUTS | 3MM | 8 | - |
| RFID READER | | 1 | RM 220.00 |
| RFID TAGS | | 3 | RM 9.00 |
| **TOTAL AMOUNT** | | | **RM 309.36** |

## REFERENCES


[1] Webopedia. "All About RFID"
Available at: *http://www.webopedia.com/DidYouKnow/Computer_Science/2005/rfid.asp*

[2] RFID Journal. "RFID GET START"
Available at: *http://www.rfidjournal.com/article/view/1338/1/129*

[3] Technology.com™ – where science meet fiction. "RFID READER"
Available at: *http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=54*

[4] Cytron Technologies. "RFID READER"
Available at: *www.cytron.com.my/listProductCategory*

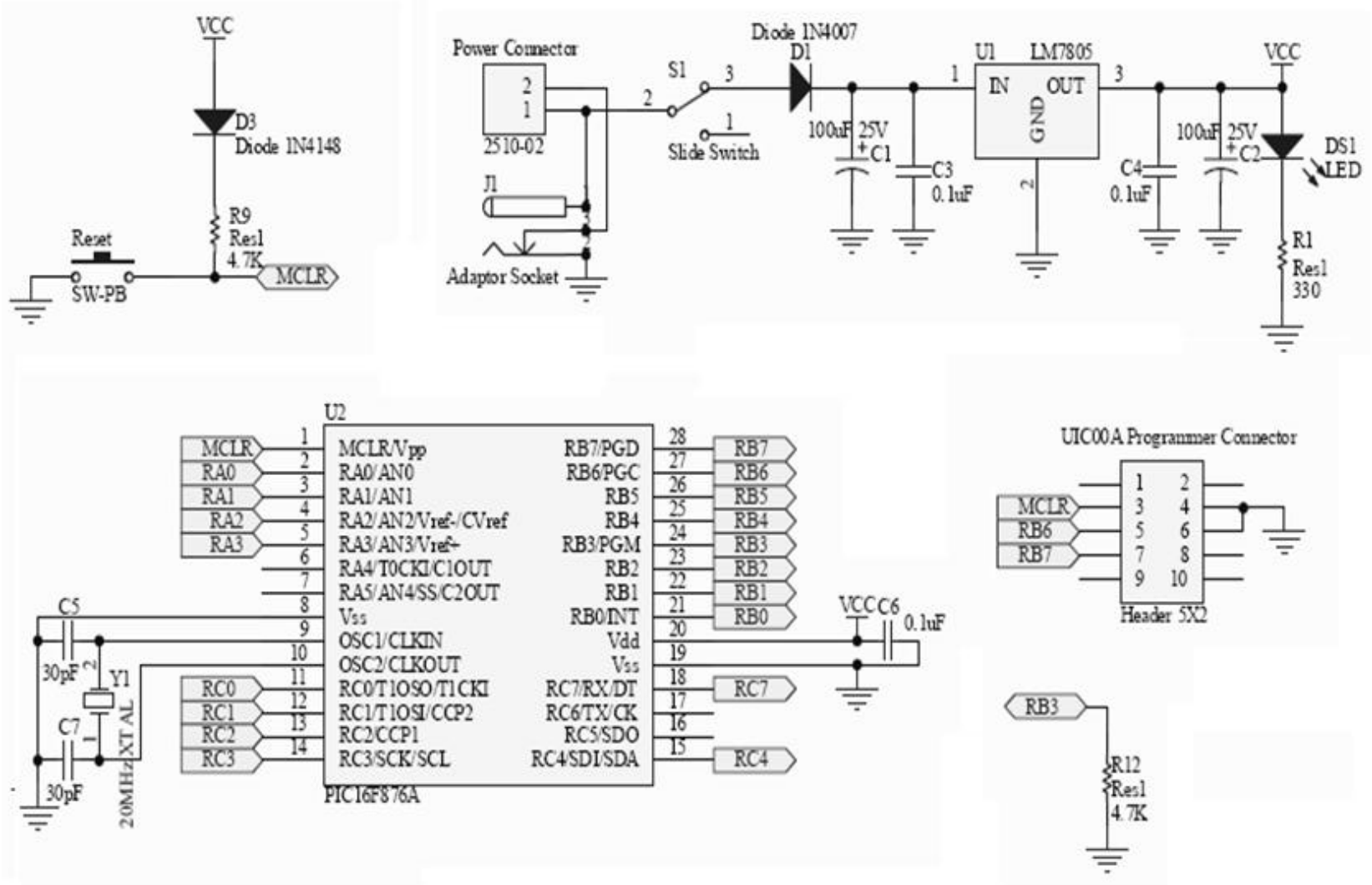
[5] ALL DATA SHEET
Available at: *http://www.alldatasheet.com*

[6] PIC16F87XA Data Sheet
Available at:*http://www.microchip.com/downloads/en/DeviceDoc/39582b.pdf*

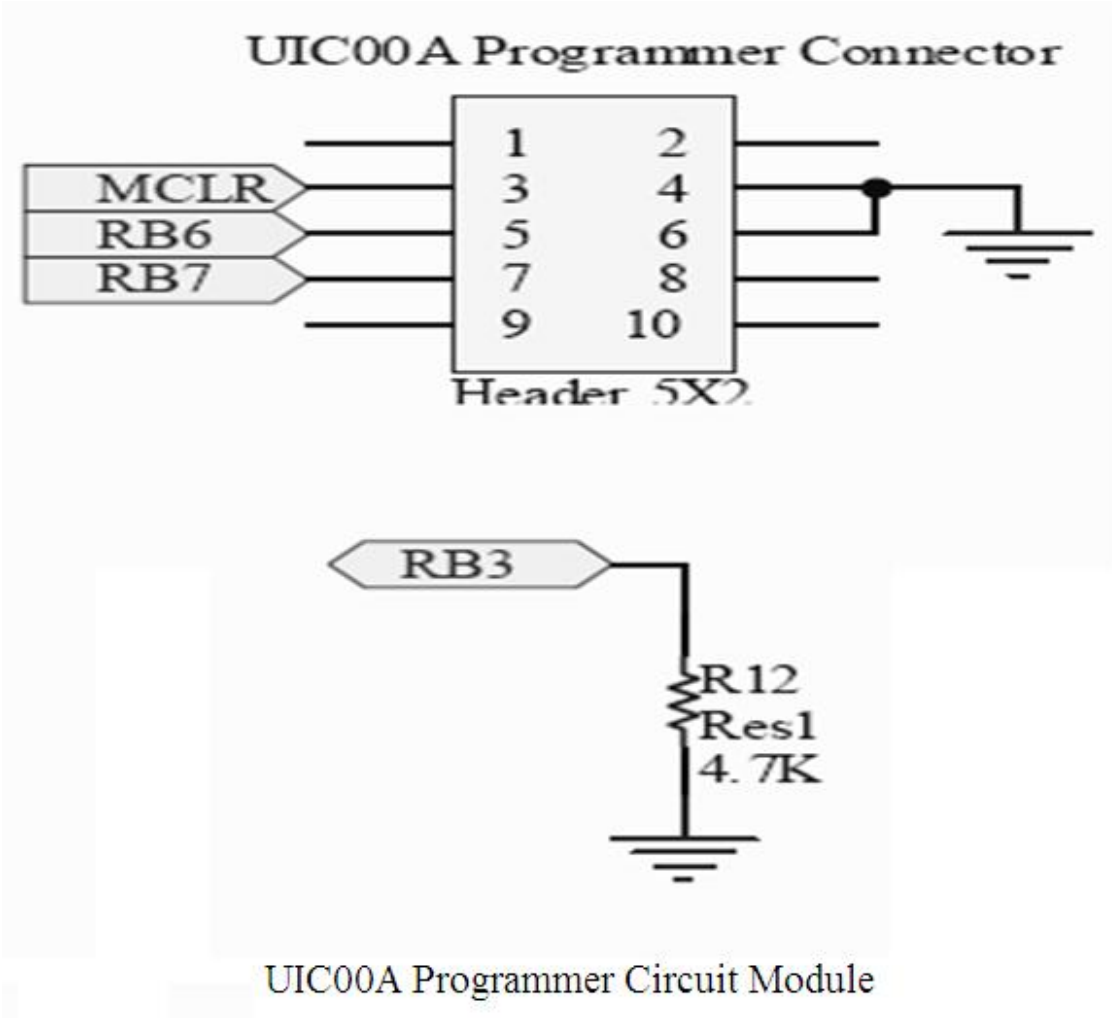
[7] PIC16F87XA Data Sheet
Available at: *http://www.microchip.com/stellent/idcplg?IdcService*

# APPENDIX A

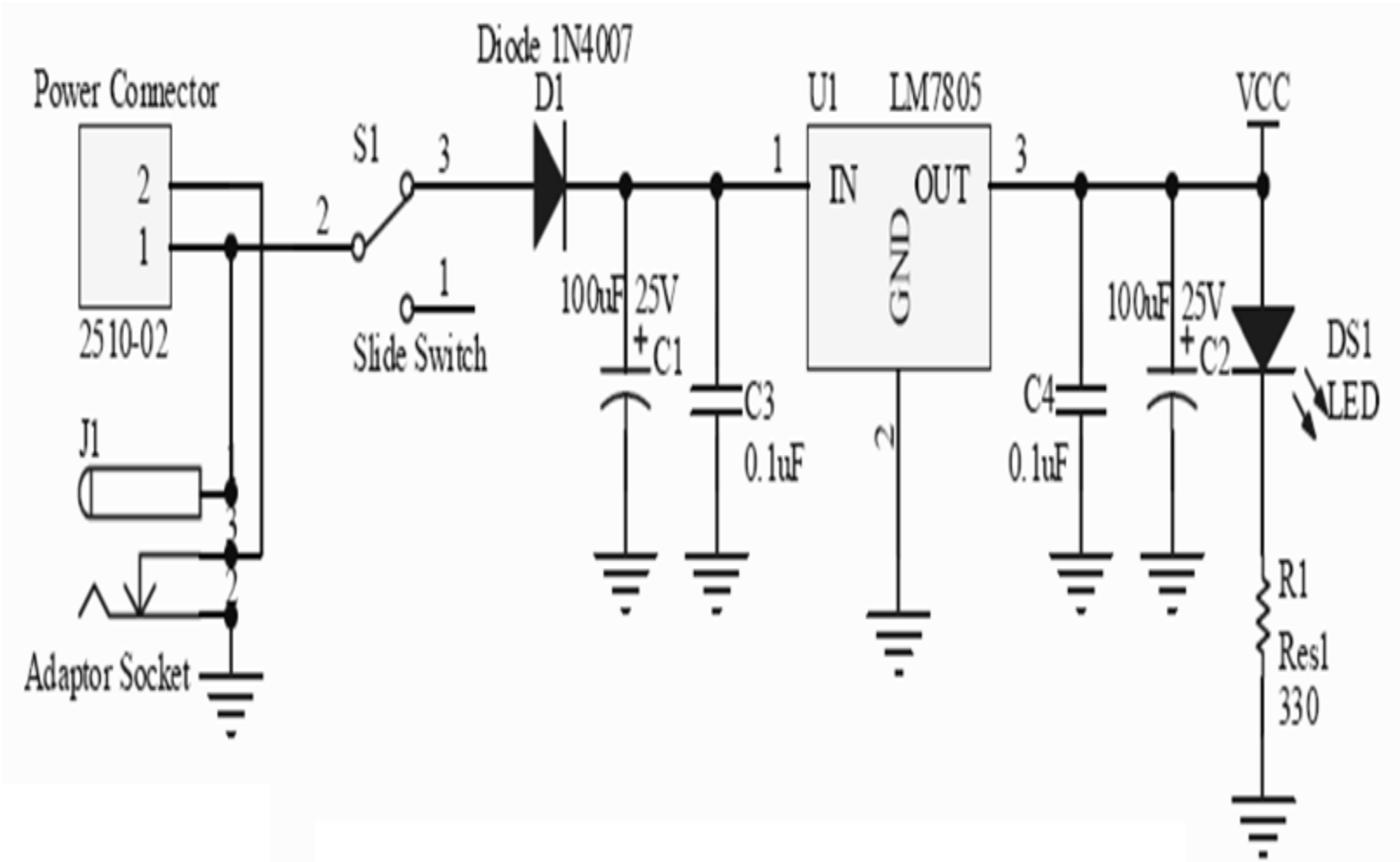
## Project Schematic



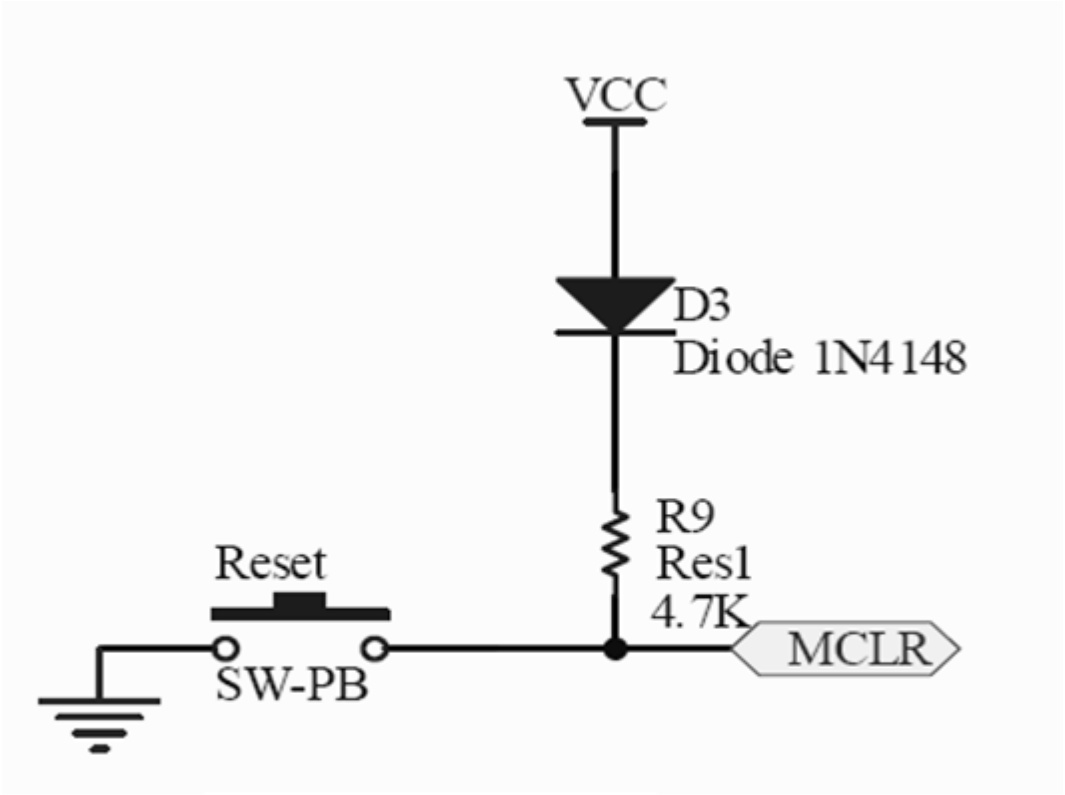

MICROCONTROLLER MAIN BOARD CIRCUIT



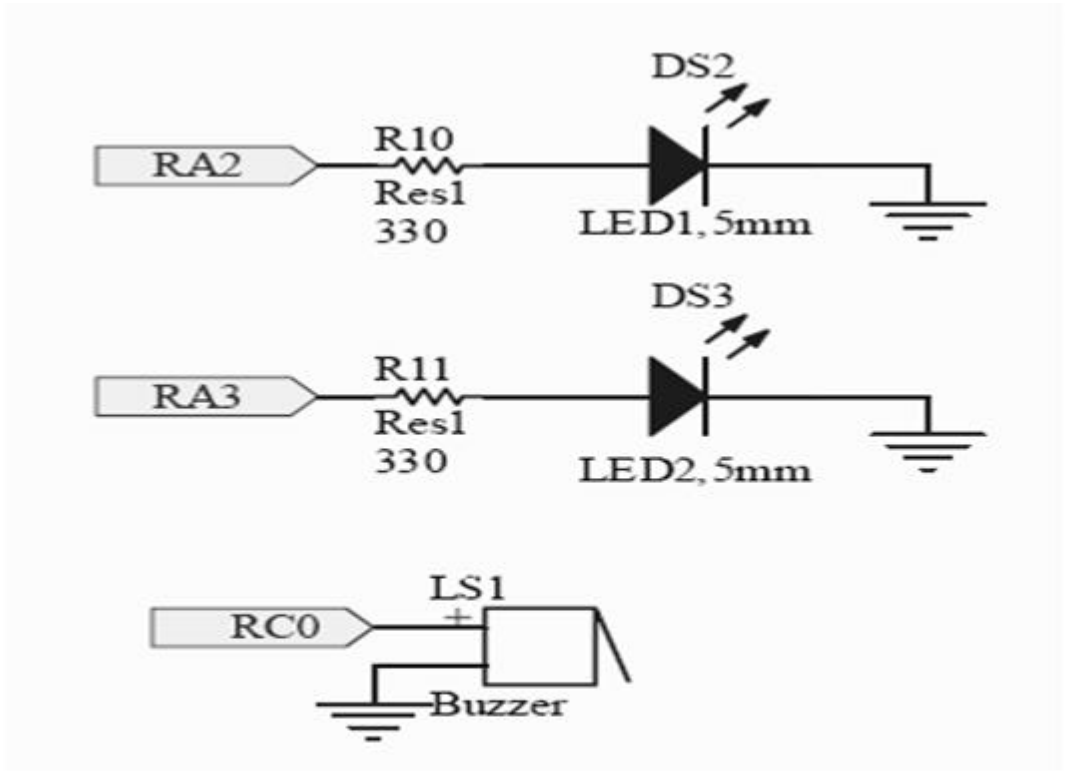

UIC00A Programmer Circuit Module



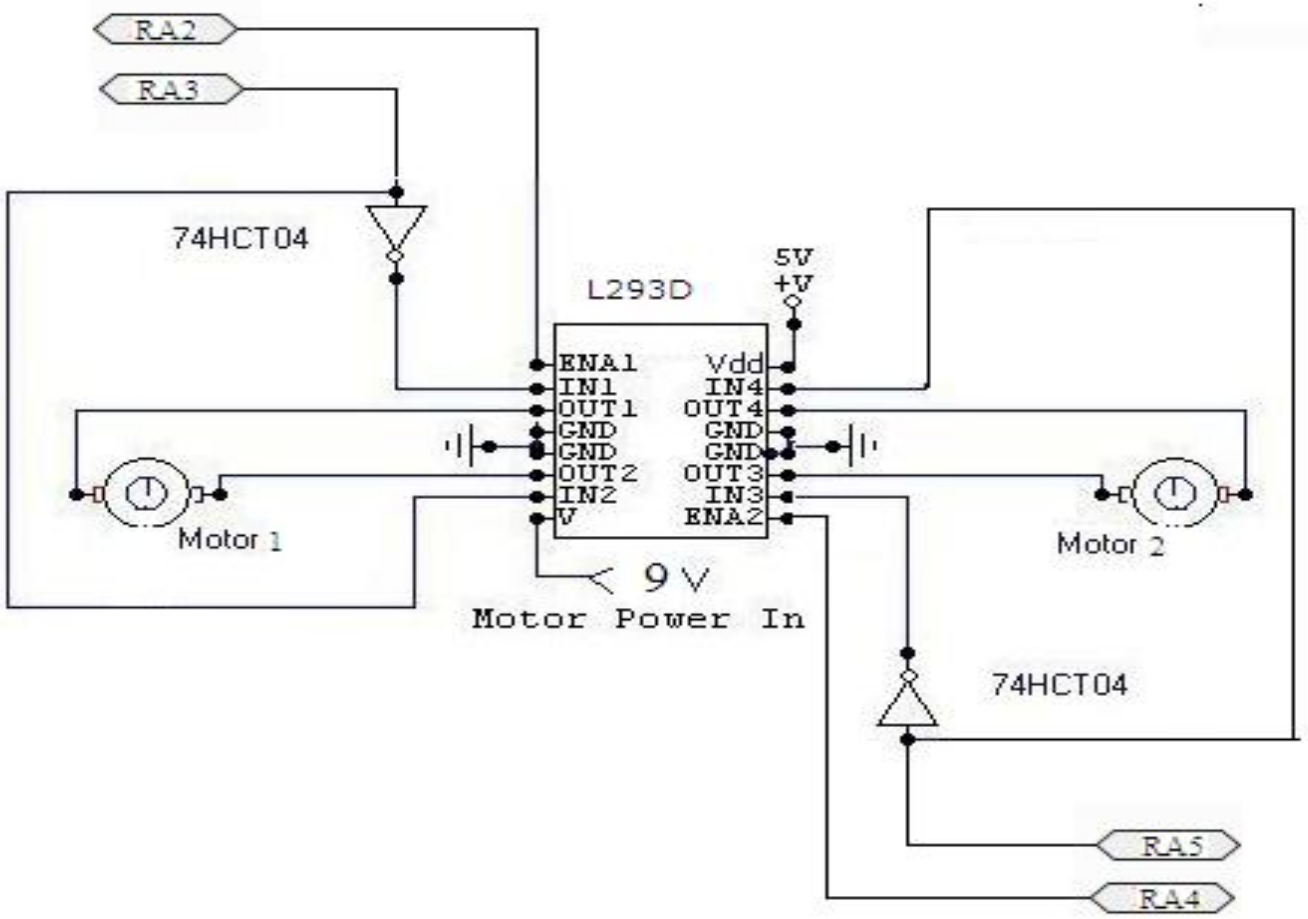

Power Supply Circuit Module



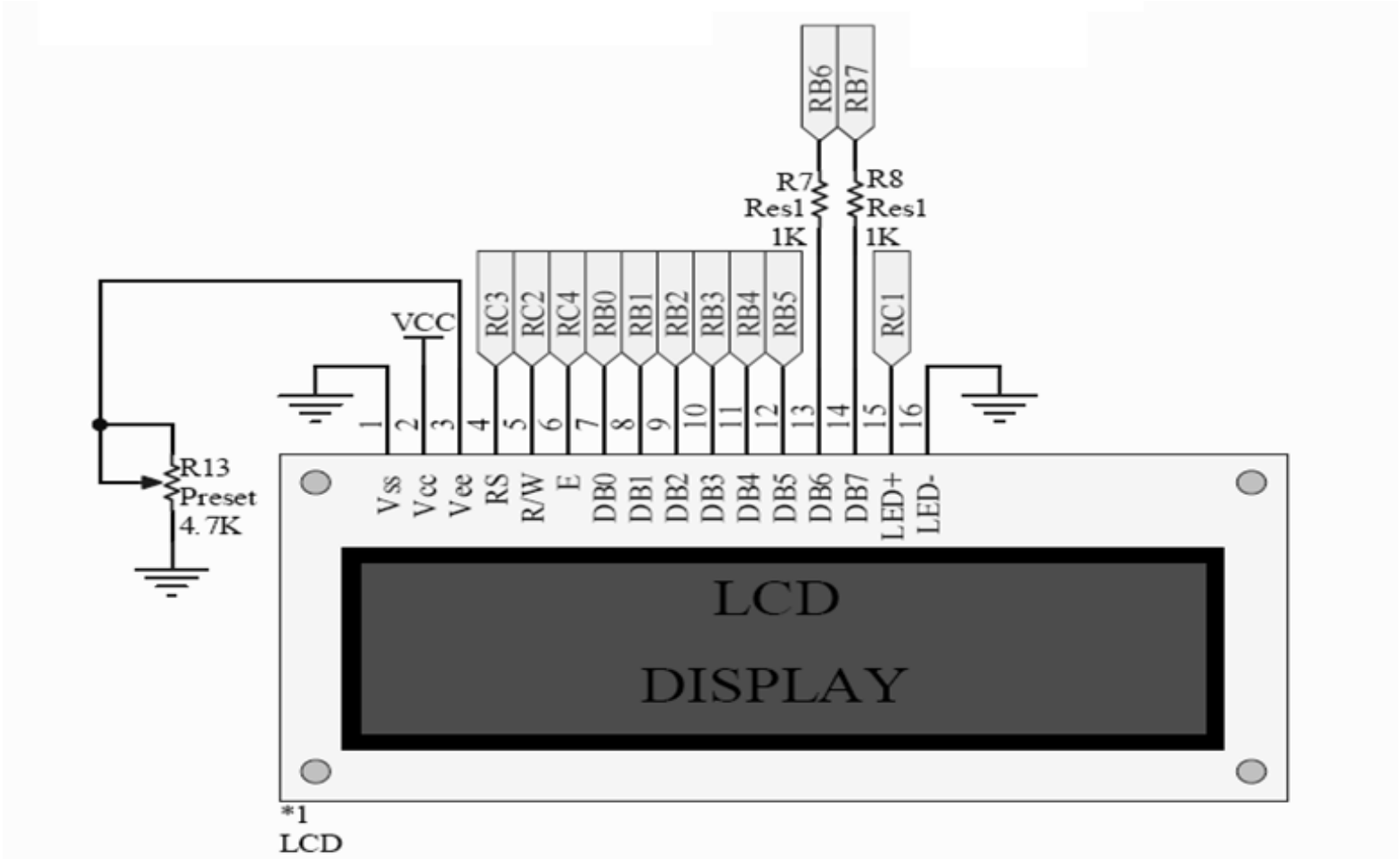

Reset Circuit Module



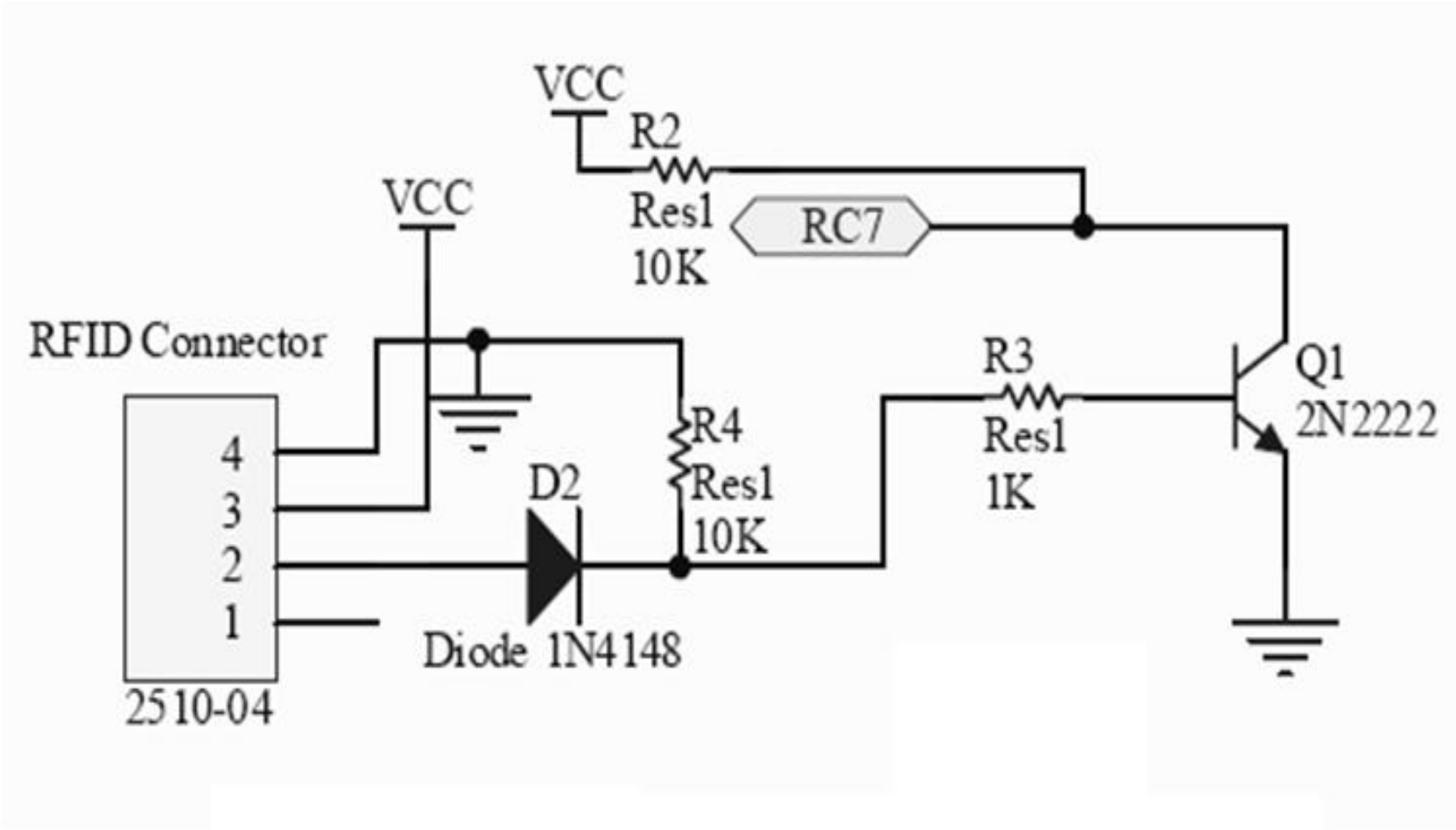

Outputs Circuit Module



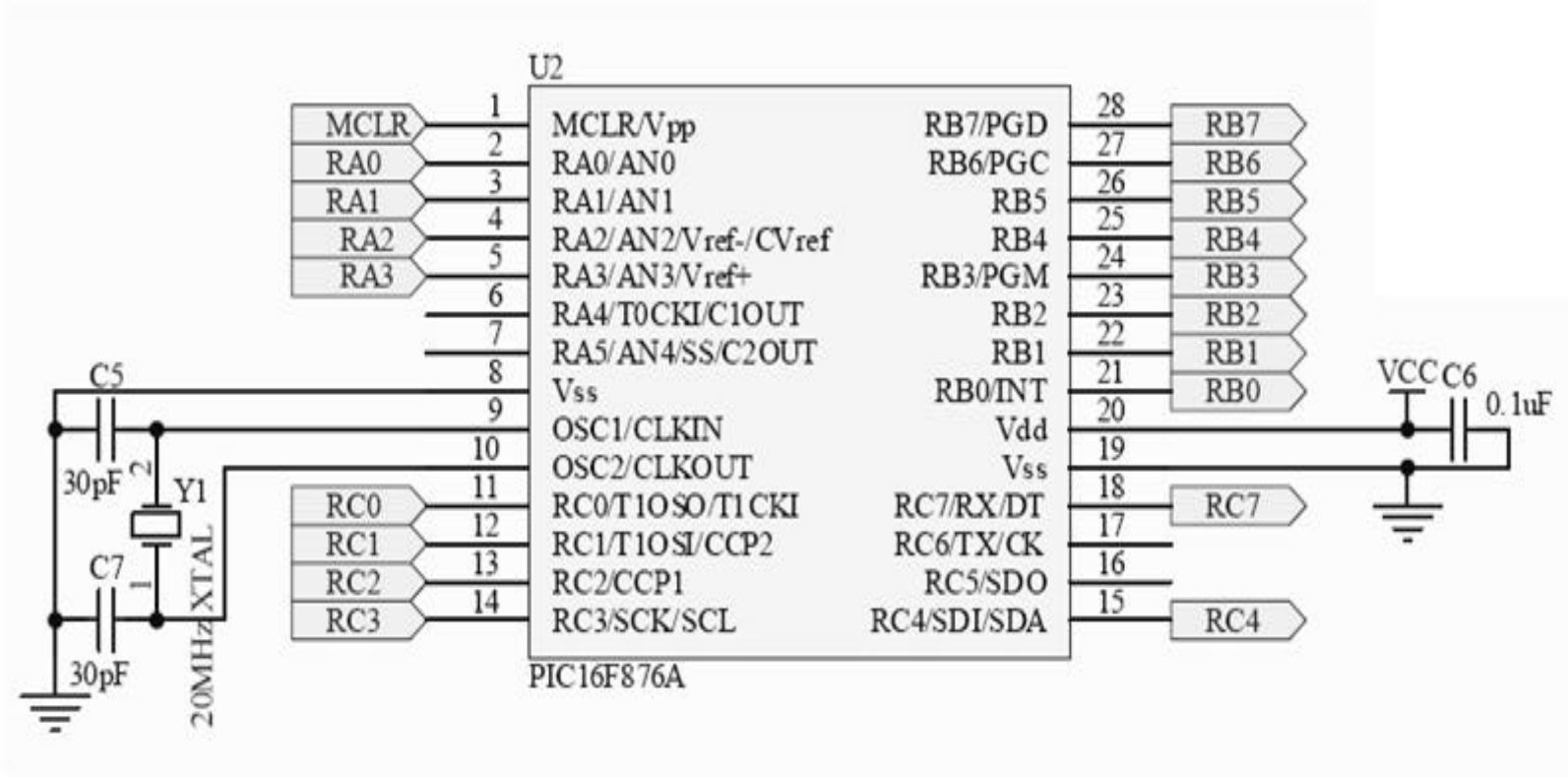

DC Motor Driver Circuit Module




LCD Display Circuit Module




RFID Circuit Module




PIC16F876A Circuit Module

# **APPENDIX** B

## Project Code Listing

```
#include <pic.h>

__CONFIG ( 0x3F32 );                //configuration for the  microcontroller


#define     rs          RC3     //RS pin of the LCD display

#define     rw          RC2     //R/W pin of the LCD display

#define     e           RC4     //E pin of the LCD display

#define     b_light             RC1     //backlight of the LCD display (1 to on backlight)

#define     buzzer      RC0     //buzzer (1 to on buzzer)

#define     button1             RA0     //button (active low)

#define     button2             RA1     //button (active low)

#define     lcd_data    PORTB               //LCD 8-bit data PORT

#define     led1        RA2     //led 1/enable1/door 1 open (active high)

#define     led2        RA3     //led 2/door close (active high)

#define     out1        RA4     //led 3enable2/door 2 open (active high)

#define     out2        RA5     //led 4/door close (active high)


void delay(unsigned long data);

void send_config(unsigned char data);

void send_char(unsigned char data);
```

```
void lcd_goto(unsigned char data);

void lcd_clr(void);

void send_string(const char *s);

unsigned char uart_rec(void);

void beep(void);


//=============================================================

void main(void)

{

        //assign variable

        unsigned char i,temp,database;

        unsigned char data[10];


        unsigned char id_1[10]={"0008349791"};  //define the Tag ID here

        unsigned char id_2[10]={"0008353275"};  //change this ID to the tag ID that
        // user want to read


        unsigned char user_1[10]={"SYUKRIZAL"};      //define the Tag user here

        unsigned char user_2[10]={"HASLAN    "};        //change this user name to
        //the tag ID owner name
```

```
//set I/O input output
TRISB = 0b00000000;         //configure PORTB I/O direction
TRISC = 0b10000000;         //configure PORTC I/O direction
TRISA = 0b11000011;         //configure PORTA I/O direction

//setup USART
SPBRG = 0x81;         //set baud rate to 9600 for 20Mhz
BRGH = 1;             //baud rate high speed option
TXEN = 1;             //enable transmission
CREN = 1;             //enable reception
SPEN = 1;             //enable serial port

//setup ADC
ADCON1 = 0b00000110;     //set ADx pin digital I/O

//configure lcd
send_config(0b00000001);   //clear display at lcd
send_config(0b00000010);   //lcd return to home
send_config(0b00000110);   //entry mode-cursor increase 1
send_config(0b00001100);   //display on, cursor off and cursor blink off
send_config(0b00111000);   //function set
```

```
//set initial condition
buzzer=0;               //off buzzer
b_light=1;              //on backlight
led1=0;                 //off led 1
led2=0;                 //off led 2

lcd_clr();                          //clear lcd
lcd_goto(0);                        //set the lcd cursor to location 0
send_string("   RFID door");        //display welcome note
lcd_goto(20);                       //set the lcd cursor to location 20
send_string(" security");           //display welcome note

beep();                             //initial beep
delay(200000);                /     /delay for display the welcome note

//infinity loop
while(1)
{
        lcd_clr();                          //clear lcd
        lcd_goto(0);                        //set lcd cursor to location 0
```

```
send_string("Place your ID");       //display note
lcd_goto(20);                       //set lcd cursor to location 20
send_string("on the reader.");      //display note

for(i=0;i<10;i+=1)data[i]=uart_rec();       //wait for 10 character
data from RFID reader


lcd_clr();                          //clear lcd
lcd_goto(20);                       //set lcd cursor to location 20
send_string("Processing.....");     //display "Processing...."
delay(40000);                       //delay

database=0;             //clear the value of database and start scanning

//comparing with the 1st id
temp=0;                 //comparing the receive id with the saved id
for(i=0;i<10;i+=1)              //comparing digit by digit
{
        if((data[i])!=(id_1[i]))temp=1;     //if the id is different from
//the id define above,
}                                   //then set temp=1;
```

```
            if(temp==0) database=1;             //if temp=0, mean the id match,
//set the database=1


            //comparing with the 2nd id
            temp=0;             //comparing the receive id with the saved id
            for(i=0;i<10;i+=1)          //comparing digit by digit
            {
                  if((data[i])!=(id_2[i]))temp=1;       //if the id is different from
//the id define above,
            }                                   //then set temp=1;
            if(temp==0) database=2;             //if temp=0, mean the id match,
//set the database=1


            lcd_clr();                          //clear lcd
            switch(database)
            {
                  case 1:                       //id 1 match
                        led1=1;
                        lcd_goto(0);            //set lcd cursor to location 0
                        send_string("ID: ");        //display "ID: "
                        for(i=0;i<10;i+=1)send_char(id_1[i]);//display tag ID
                        lcd_goto(20);           //set lcd cursor to location 20
```

```
            send_string("user: ");        //display "user: "
            for(i=0;i<10;i+=1)send_char(user_1[i]);
            //display user name
            beep();                                  //beep once
            delay(100000);
            led1=0;
            delay(300000);
            led2=1;
            led1=1;
            break;

        case 2:                          //id_2 match
            led3=1;
                                         //on led 3
            lcd_goto(0);             //set lcd cursor to location 0
            send_string("ID: ");                  //display "ID: "
            for(i=0;i<10;i+=1)send_char(id_2[i]);//display tag ID
            lcd_goto(20);            //set lcd cursor to location 20
            send_string("user: ");        //display "user: "
            for(i=0;i<10;i+=1)send_char(user_2[i]);
            //display user name
```

```
                beep();                         //beep once
                delay(100000);
                led4=0;
                delay(300000);
                led4=1;
                led3=1;
                break;

        default:                //id doesnt match
                lcd_goto(0);            //set lcd cursor to location 0
                send_string("ID: ");            //display "ID: "
                for(i=0;i<10;i+=1)send_char(data[i]);
                //display tag ID
                lcd_goto(20);   //set lcd cursor to location 20
                send_string("user not found");
                //display "user not found"
                beep();                         //beep twice
                beep();
                break;
}
delay(100000);                  //delay
```

```
                led1=0;                                 //off led after the process
complete
                led2=0;




        }


}
//============================================================
void delay(unsigned long data)      //delay function, the delay time
{                                   //depend on the given value
        for( ;data>0;data-=1);
}

void send_config(unsigned char data)      //send lcd configuration
{
        rw=0;                   //set lcd to write mode
        rs=0;                   //set lcd to configuration mode
        lcd_data=data;                  //lcd data port = data
        e=1;                    //pulse e to confirm the data
        delay(50);
```

```
        e=0;
        delay(50);
}

void send_char(unsigned char data)          //send lcd character
{
        rw=0;                   //set lcd to write mode
        rs=1;                   //set lcd to display mode
        lcd_data=data;                  //lcd data port = data
        e=1;                    //pulse e to confirm the data
        delay(10);
        e=0;
        delay(10);
}

void lcd_goto(unsigned char data)  //set the location of the lcd cursor
{                                       //if the given value is (0-15) the
        if(data<16)                     //cursor will be at the upper line
        {                               //if the given value is (20-35) the
                send_config(0x80+data);      //cursor will be at the lower line
}                               //location of the lcd cursor(2X16):
```

```
else                        // ---------------------------------------------------
{                           // | |00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15| |
data=data-20;               // | |20|21|22|23|24|25|26|27|28|29|30|31|32|33|34|35| |
send_config(0xc0+data);     // ---------------------------------------------------
        }
}


void lcd_clr(void)                  //clear the lcd
{
        send_config(0x01);
        delay(600);
}


void send_string(const char *s)     //send a string to display in the lcd
{
        unsigned char i=0;
        while (s && *s)send_char (*s++);


}


unsigned char uart_rec(void)                //receive uart value
```

```
{
        unsigned char rec_data;
        while(RCIF==0);             //wait for data
        rec_data = RCREG;
        return rec_data;            //return the data received
}


void beep(void)                 //short beep function
{
        buzzer=1;       //on buzzer
        delay(10000); //short delay
        buzzer=0;       //off buzzer
        delay(10000); //short delay
}
```

# APPENDIX C

## PIC16F876A Data Sheet

# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

### High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
  DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and $I^2C$™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference ($V_{REF}$) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

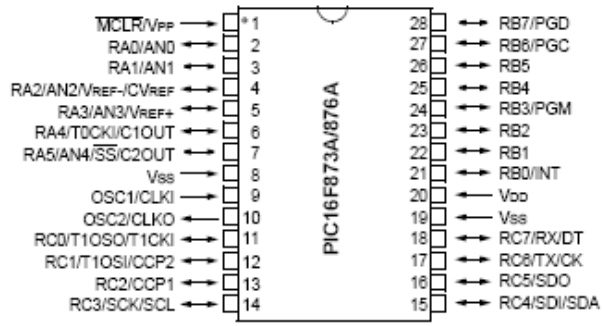
### CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

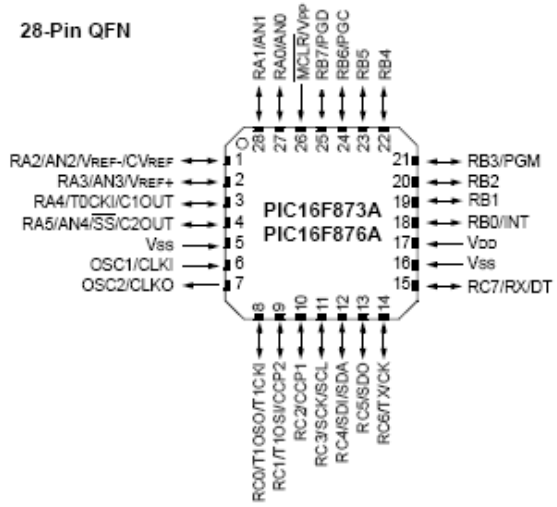
| Device | Program Memory | | Data SRAM (Bytes) | EEPROM (Bytes) | I/O | 10-bit A/D (ch) | CCP (PWM) | MSSP | | USART | Timers 8/16-bit | Comparators |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bytes | # Single Word Instructions | | | | | | SPI | Master $I^2C$ | | | |
| PIC16F873A | 7.2K | 4096 | 192 | 128 | 22 | 5 | 2 | Yes | Yes | Yes | 2/1 | 2 |
| PIC16F874A | 7.2K | 4096 | 192 | 128 | 33 | 8 | 2 | Yes | Yes | Yes | 2/1 | 2 |
| PIC16F876A | 14.3K | 8192 | 368 | 256 | 22 | 5 | 2 | Yes | Yes | Yes | 2/1 | 2 |
| PIC16F877A | 14.3K | 8192 | 368 | 256 | 33 | 8 | 2 | Yes | Yes | Yes | 2/1 | 2 |

## Pin Diagrams

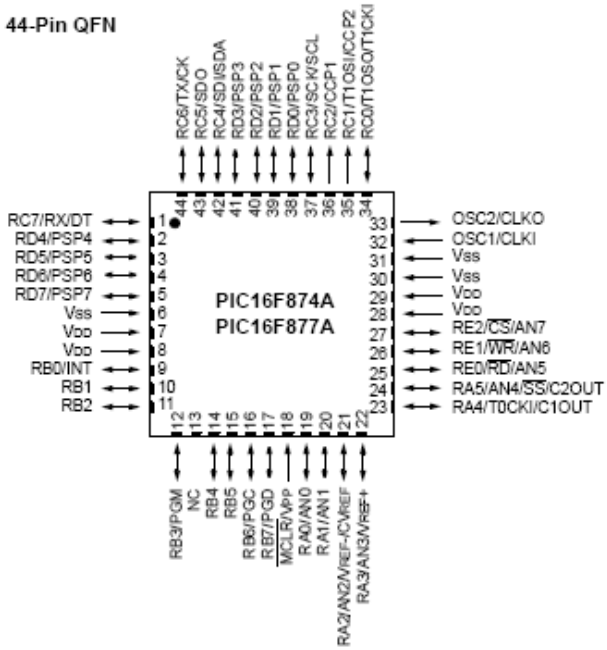
28-Pin PDIP, SOIC, SSOP




28-Pin QFN




44-Pin QFN



## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

**TABLE 1-1: PIC16F87XA DEVICE FEATURES**

| Key Features | PIC16F873A | PIC16F874A | PIC16F876A | PIC16F877A |
|---|---|---|---|---|
| Operating Frequency | DC – 20 MHz | DC – 20 MHz | DC – 20 MHz | DC – 20 MHz |
| Resets (and Delays) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) |
| Flash Program Memory (14-bit words) | 4K | 4K | 8K | 8K |
| Data Memory (bytes) | 192 | 192 | 368 | 368 |
| EEPROM Data Memory (bytes) | 128 | 128 | 256 | 256 |
| Interrupts | 14 | 15 | 14 | 15 |
| I/O Ports | Ports A, B, C | Ports A, B, C, D, E | Ports A, B, C | Ports A, B, C, D, E |
| Timers | 3 | 3 | 3 | 3 |
| Capture/Compare/PWM modules | 2 | 2 | 2 | 2 |
| Serial Communications | MSSP, USART | MSSP, USART | MSSP, USART | MSSP, USART |
| Parallel Communications | — | PSP | — | PSP |
| 10-bit Analog-to-Digital Module | 5 input channels | 8 input channels | 5 input channels | 8 input channels |
| Analog Comparators | 2 | 2 | 2 | 2 |
| Instruction Set | 35 Instructions | 35 Instructions | 35 Instructions | 35 Instructions |
| Packages | 28-pin PDIP<br>28-pin SOIC<br>28-pin SSOP<br>28-pin QFN | 40-pin PDIP<br>44-pin PLCC<br>44-pin TQFP<br>44-pin QFN | 28-pin PDIP<br>28-pin SOIC<br>28-pin SSOP<br>28-pin QFN | 40-pin PDIP<br>44-pin PLCC<br>44-pin TQFP<br>44-pin QFN |

## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

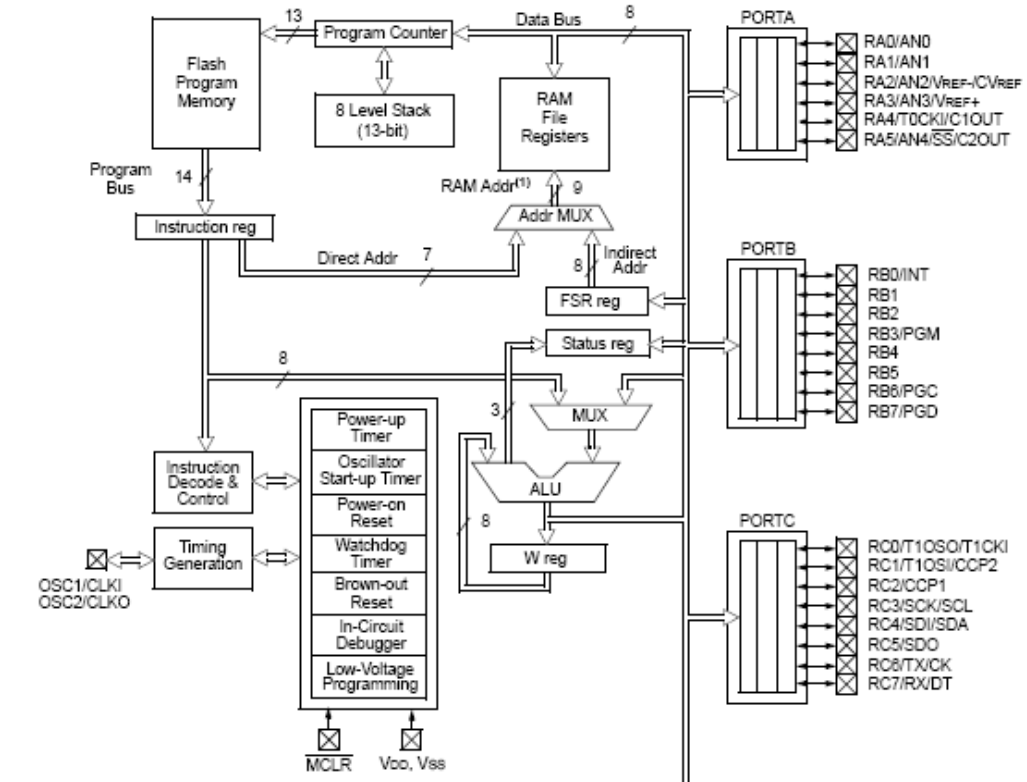
The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

**TABLE 1-1: PIC16F87XA DEVICE FEATURES**

| Key Features | PIC16F873A | PIC16F874A | PIC16F876A | PIC16F877A |
|---|---|---|---|---|
| Operating Frequency | DC – 20 MHz | DC – 20 MHz | DC – 20 MHz | DC – 20 MHz |
| Resets (and Delays) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) |
| Flash Program Memory (14-bit words) | 4K | 4K | 8K | 8K |
| Data Memory (bytes) | 192 | 192 | 368 | 368 |
| EEPROM Data Memory (bytes) | 128 | 128 | 256 | 256 |
| Interrupts | 14 | 15 | 14 | 15 |
| I/O Ports | Ports A, B, C | Ports A, B, C, D, E | Ports A, B, C | Ports A, B, C, D, E |
| Timers | 3 | 3 | 3 | 3 |
| Capture/Compare/PWM modules | 2 | 2 | 2 | 2 |
| Serial Communications | MSSP, USART | MSSP, USART | MSSP, USART | MSSP, USART |
| Parallel Communications | — | PSP | — | PSP |
| 10-bit Analog-to-Digital Module | 5 input channels | 8 input channels | 5 input channels | 8 input channels |
| Analog Comparators | 2 | 2 | 2 | 2 |
| Instruction Set | 35 Instructions | 35 Instructions | 35 Instructions | 35 Instructions |
| Packages | 28-pin PDIP<br>28-pin SOIC<br>28-pin SSOP<br>28-pin QFN | 40-pin PDIP<br>44-pin PLCC<br>44-pin TQFP<br>44-pin QFN | 28-pin PDIP<br>28-pin SOIC<br>28-pin SSOP<br>28-pin QFN | 40-pin PDIP<br>44-pin PLCC<br>44-pin TQFP<br>44-pin QFN |

FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM




| Device | Program Flash | Data Memory | Data EEPROM |
|---|---|---|---|
| PIC16F873A | 4K words | 192 Bytes | 128 Bytes |
| PIC16F876A | 8K words | 368 Bytes | 256 Bytes |

**Note 1:** Higher order bits are from the Status register.

**TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION**

| Pin Name | PDIP, SOIC, SSOP Pin# | QFN Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|
| OSC1/CLKI | 9 | 6 | | ST/CMOS(3) | Oscillator crystal or external clock input. |
| OSC1 | | | I | | Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. |
| CLKI | | | I | | External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins). |
| OSC2/CLKO | 10 | 7 | | — | Oscillator crystal or clock output. |
| OSC2 | | | O | | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. |
| CLKO | | | O | | In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| $\overline{MCLR}$/VPP | 1 | 26 | | ST | Master Clear (input) or programming voltage (output). |
| $\overline{MCLR}$ | | | I | | Master Clear (Reset) input. This pin is an active low Reset to the device. |
| VPP | | | P | | Programming voltage input. |
| | | | | | PORTA is a bidirectional I/O port. |
| RA0/AN0 | 2 | 27 | | TTL | |
| RA0 | | | I/O | | Digital I/O. |
| AN0 | | | I | | Analog input 0. |
| RA1/AN1 | 3 | 28 | | TTL | |
| RA1 | | | I/O | | Digital I/O. |
| AN1 | | | I | | Analog input 1. |
| RA2/AN2/VREF-/ CVREF | 4 | 1 | | TTL | |
| | | | I/O | | Digital I/O. |
| RA2 | | | I | | Analog input 2. |
| AN2 | | | I | | A/D reference voltage (Low) input. |
| VREF- | | | O | | Comparator VREF output. |
| CVREF | | | | | |
| RA3/AN3/VREF+ | 5 | 2 | | TTL | |
| RA3 | | | I/O | | Digital I/O. |
| AN3 | | | I | | Analog input 3. |
| VREF+ | | | I | | A/D reference voltage (High) input. |
| RA4/T0CKI/C1OUT | 6 | 3 | | ST | |
| RA4 | | | I/O | | Digital I/O – Open-drain when configured as output. |
| T0CKI | | | I | | Timer0 external clock input. |
| C1OUT | | | O | | Comparator 1 output. |
| RA5/AN4/$\overline{SS}$/C2OUT | 7 | 4 | | TTL | |
| RA5 | | | I/O | | Digital I/O. |
| AN4 | | | I | | Analog input 4. |
| $\overline{SS}$ | | | I | | SPI slave select input. |
| C2OUT | | | O | | Comparator 2 output. |

**Legend:** I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.
**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.
**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION (CONTINUED)

| Pin Name | PDIP, SOIC, SSOP Pin# | QFN Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|
| | | | | | PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. |
| RB0/INT | 21 | 18 | | TTL/ST(1) | |
| RB0 | | | I/O | | Digital I/O. |
| INT | | | I | | External interrupt. |
| RB1 | 22 | 19 | I/O | TTL | Digital I/O. |
| RB2 | 23 | 20 | I/O | TTL | Digital I/O. |
| RB3/PGM | 24 | 21 | | TTL | |
| RB3 | | | I/O | | Digital I/O. |
| PGM | | | I | | Low-voltage (single-supply) ICSP programming enable pin. |
| RB4 | 25 | 22 | I/O | TTL | Digital I/O. |
| RB5 | 26 | 23 | I/O | TTL | Digital I/O. |
| RB6/PGC | 27 | 24 | | TTL/ST(2) | |
| RB6 | | | I/O | | Digital I/O. |
| PGC | | | I | | In-circuit debugger and ICSP programming clock. |
| RB7/PGD | 28 | 25 | | TTL/ST(2) | |
| RB7 | | | I/O | | Digital I/O. |
| PGD | | | I/O | | In-circuit debugger and ICSP programming data. |
| | | | | | PORTC is a bidirectional I/O port. |
| RC0/T1OSO/T1CKI | 11 | 8 | | ST | |
| RC0 | | | I/O | | Digital I/O. |
| T1OSO | | | O | | Timer1 oscillator output. |
| T1CKI | | | I | | Timer1 external clock input. |
| RC1/T1OSI/CCP2 | 12 | 9 | | ST | |
| RC1 | | | I/O | | Digital I/O. |
| T1OSI | | | I | | Timer1 oscillator input. |
| CCP2 | | | I/O | | Capture2 input, Compare2 output, PWM2 output. |
| RC2/CCP1 | 13 | 10 | | ST | |
| RC2 | | | I/O | | Digital I/O. |
| CCP1 | | | I/O | | Capture1 input, Compare1 output, PWM1 output. |
| RC3/SCK/SCL | 14 | 11 | | ST | |
| RC3 | | | I/O | | Digital I/O. |
| SCK | | | I/O | | Synchronous serial clock input/output for SPI mode. |
| SCL | | | I/O | | Synchronous serial clock input/output for $I^2C$ mode. |
| RC4/SDI/SDA | 15 | 12 | | ST | |
| RC4 | | | I/O | | Digital I/O. |
| SDI | | | I | | SPI data in. |
| SDA | | | I/O | | $I^2C$ data I/O. |
| RC5/SDO | 16 | 13 | | ST | |
| RC5 | | | I/O | | Digital I/O. |
| SDO | | | O | | SPI data out. |
| RC6/TX/CK | 17 | 14 | | ST | |
| RC6 | | | I/O | | Digital I/O. |
| TX | | | O | | USART asynchronous transmit. |
| CK | | | I/O | | USART1 synchronous clock. |
| RC7/RX/DT | 18 | 15 | | ST | |
| RC7 | | | I/O | | Digital I/O. |
| RX | | | I | | USART asynchronous receive. |
| DT | | | I/O | | USART synchronous data. |
| VSS | 8, 19 | 5, 6 | P | — | Ground reference for logic and I/O pins. |
| VDD | 20 | 17 | P | — | Positive supply for logic and I/O pins. |

**Legend:** I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.
**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.
**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

# APPENDIX D

## LCD 162A Data Sheet

## ● GENERAL SPECIFICATION

Interface with 4-bit or 8-bit MPU(directly connected M6800 serial MPU)

### Display Specification

Display color-Display background color: ①STN: Yellow-Green,Blue-Gray, Black-White

②TN: Position,Negative

Polarizer mode: Positive,Negative;Reflective ,Transflective, Transmissive

Viewing angle: 6:00 OR 12:00

Display duty: 1/16 Driving bias: 1/5

Character Generator ROM (CGROM):10080 bits(208 characterX5X8 dots)&(32 character X5X11 dots)

Character Generator RAM (CGRAM): 64 X 8 bits (8 charactersX5X8 dots)

Display Data RAM (DDRAM): 80X8 bits ( 80 characters max)

### Mechanical characteristics (Unit:mm)

Extenal dimension: 84.0X44.0X10.0 (15.0 for LED Backlight)

View area: 61.0X15.8 Character font: 5X7dots+cursor

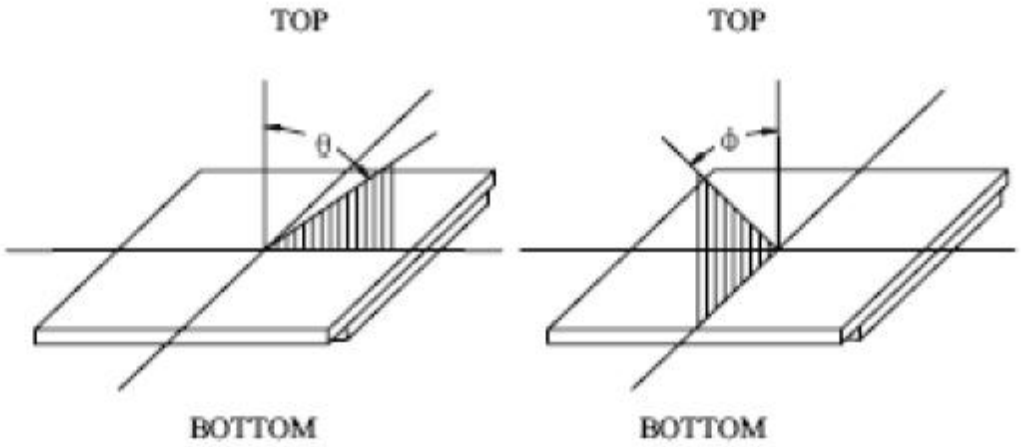
Character size: 2.96X5.56 Dots size: 0.528X0.625

Character pitch: 3.55X6.15

### POWER: +5V power

## ● Optical Characteristics

(1) Definition of viewing Angle



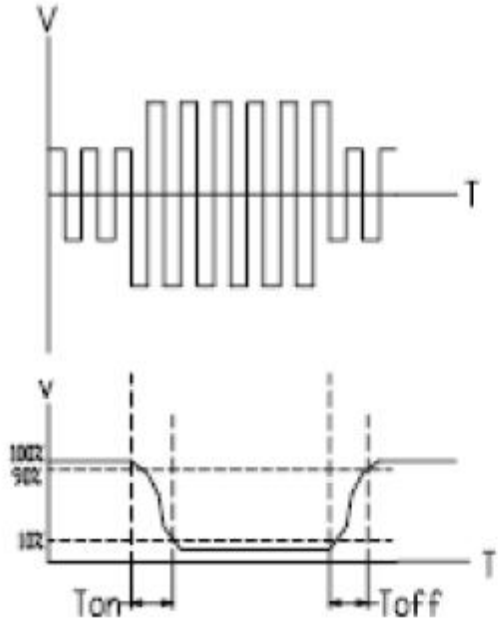

(2) Definition of Contrast Ratio:

$$\text{Contrast Ratio} = \frac{\text{Reflectance value of non-selected state brightness}}{\text{Reflectance value of selected state brightness}}$$

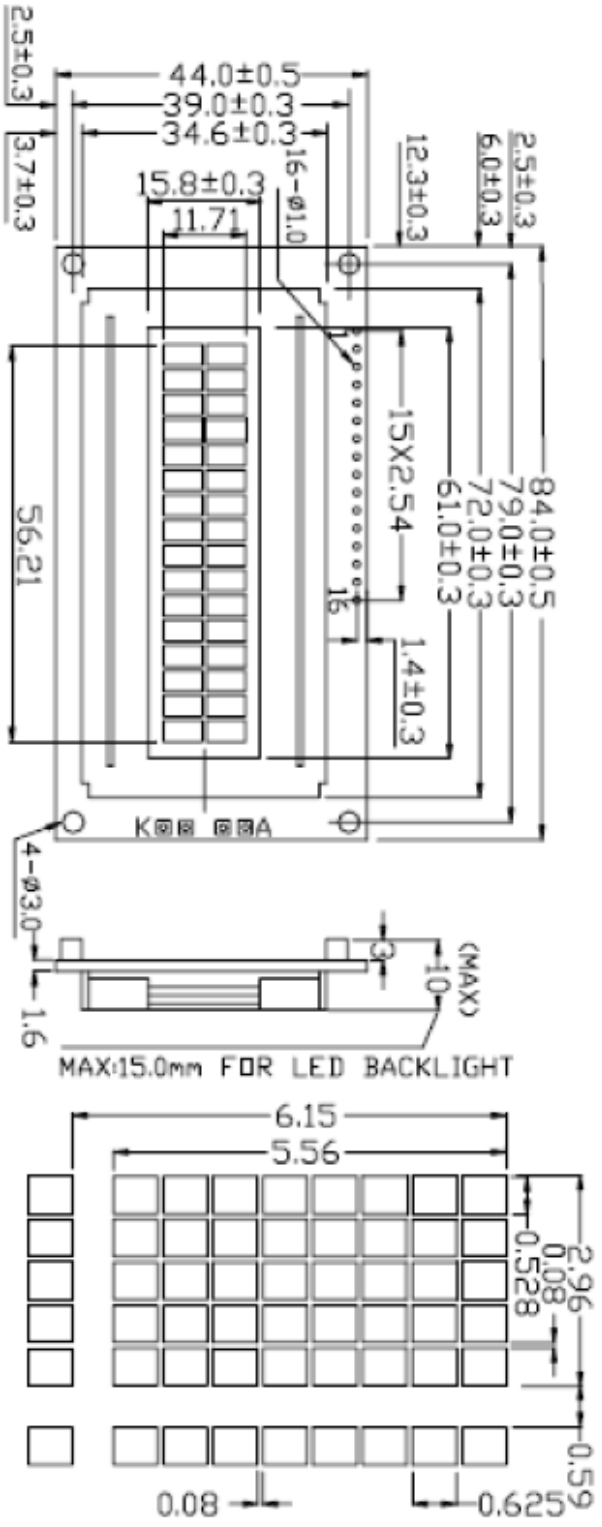
Test condition : standard A light source

(3) Response Time

Response time is measured as the shortest period of time possible between the change in state of an LCD segment as demonstrated below



- **External Dimension**



## ● Absolute Maximum Ratings

| Item | Symbol | Condition | Standard Value | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| Supply Voltage for logic | Vdd | Ta=25℃ | -0.3 | 7.0 | V |
| Supply Voltage for LCD | V5 | | Vdd-13.5 | 0 | V |
| Input Voltage | Vi | | -0.3 | Vdd+0.3 | V |
| Operating Temperature | Top | - | 0 | 50 | ℃ |
| Storage Temperature | Tstg | - | -20 | 70 | ℃ |

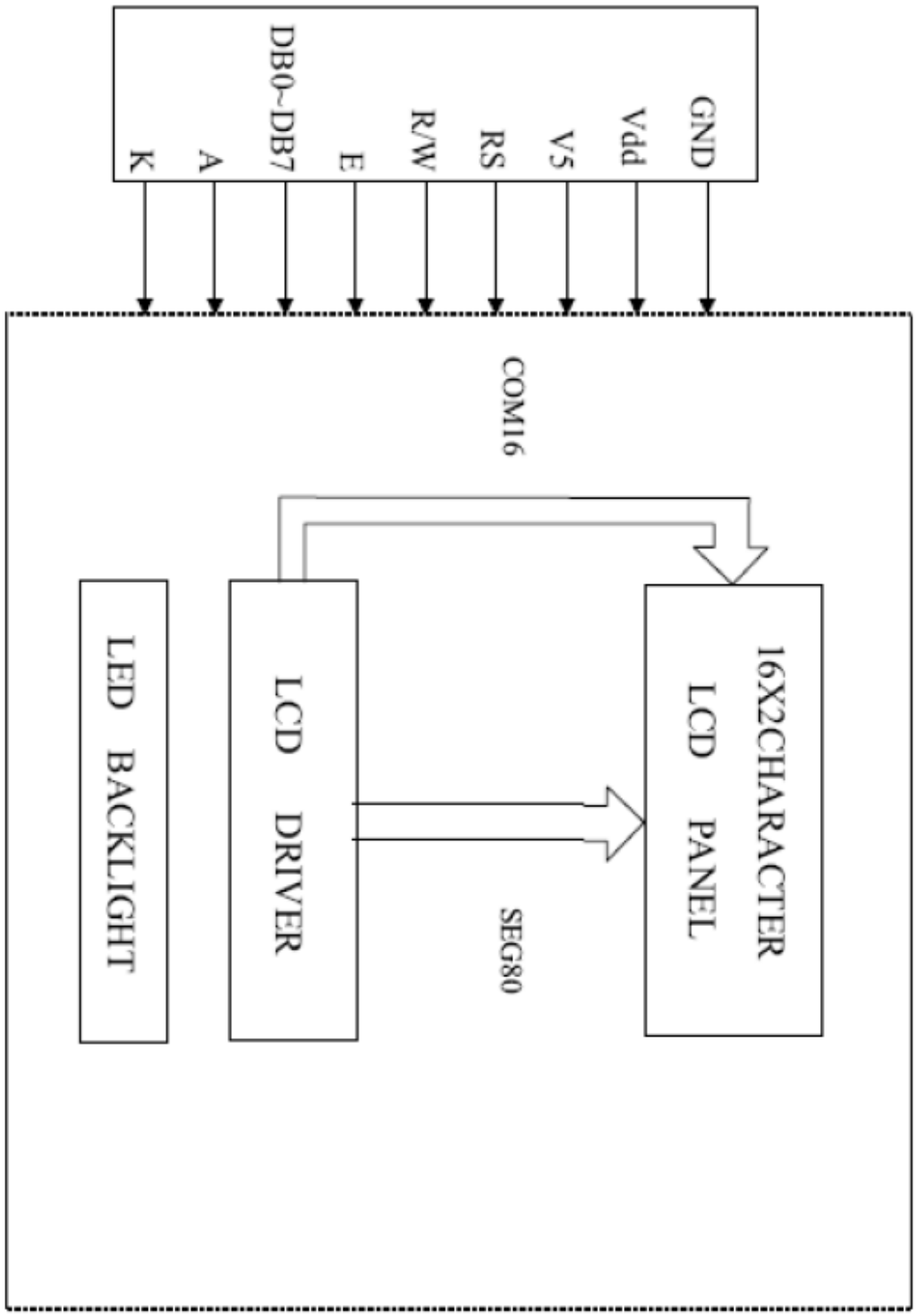
## ● Electrical Characteristics (Ta=25℃, Vdd= 5.0V)

| Item | Symbol | Condition | Standard Value | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Type | Max | |
| Supply Voltage for logic | Vdd-GND | - | 4.5 | 5.0 | 5.5 | V |
| Supply Current for logic | Idd | Vdd=5V | - | 1.5 | 3.0 | mA |
| Driving Current for LCD | Iee | | - | 0.4 | 1.0 | mA |
| Driving Voltage for LCD | Vdd-V5 | | 3.8 | 4.5 | 4.9 | V |
| Input Voltage H level | Vih | | 2.2 | - | Vdd | V |
| Input Voltage L level | Vil | | -0.3 | - | 0.6 | V |
| Output Voltage "H" | Voh | Ioh=-0.205mA | 2.4 | - | - | V |
| Output Voltage "L" | Vol | Iol=1.2mA | - | - | 0.4 | V |

## ● Absolute Maximum Ratings For LED Backlight

| PARAMETER | SYMBOL | CONDITION | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| Supply Voltage | $V_{LED}$ | If=200mA | - | 4.2 | - | V |
| LED Forward Consumption Current | If | Vf=4.2V | - | 83 | - | mA |
| LED Allowable Dissipation | Pd | - | - | 350 | - | mW |

- **Block Diagram**



## ● AC Characteristics (Vdd=4.5V~5.5V,Ta=-30~+85℃)

| Mode | Characteristic | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Write Mode | E Cycle Time | $t_C$ | 500 | - | - | ns |
| | E Rise/Fall Time | $t_R, t_F$ | - | - | 20 | |
| | E Pulse Width (High,Low) | $t_W$ | 230 | - | - | |
| | R/W and RS Setup Time | $t_{SU1}$ | 40 | - | - | |
| | R/W and RS Hold Time | $t_{H1}$ | 10 | - | - | |
| | Data Setup Time | $t_{SU2}$ | 80 | - | - | |
| | Data Hold Time | $t_{H2}$ | 10 | - | - | |
| Read Mode | E Cycle Time | $t_C$ | 500 | - | - | ns |
| | E Rise/Fall Time | $t_R, t_F$ | - | - | 20 | |
| | E Pulse Width (High,Low) | $t_W$ | 230 | - | - | |
| | R/W and RS Setup Time | $t_{SU}$ | 40 | - | - | |
| | R/W and RS Hold Time | $t_H$ | 10 | - | - | |
| | Data Output Delay Time | $t_D$ | - | - | 120 | |
| | Data Hold Time | $t_{DH}$ | 5 | - | - | |

# APPENDIX E

## LM7805 Data Sheet

FAIRCHILD
SEMICONDUCTOR®

www.fairchildsemi.com

# MC78XX/LM78XX/MC78XXA

## 3-Terminal 1A Positive Voltage Regulator

### Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

### Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

TO-220

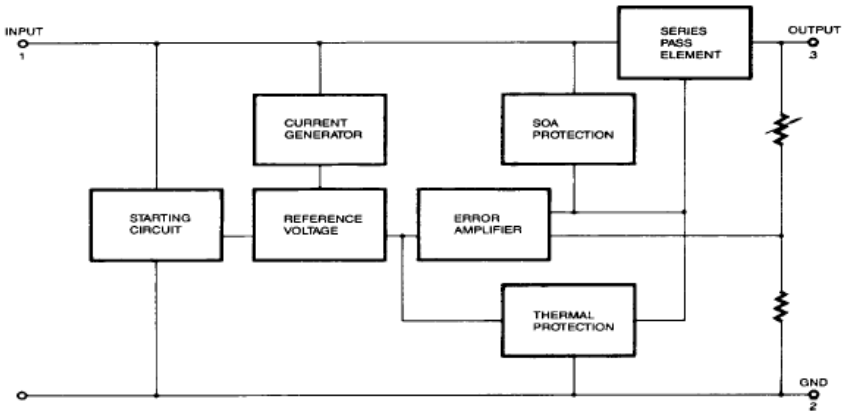
1

D-PAK

1

1. Input 2. GND 3. Output

### Internal Block Digram




Rev. 1.0.1

## Absolute Maximum Ratings

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Input Voltage (for $V_O$ = 5V to 18V)<br>(for $V_O$ = 24V) | $V_I$<br>$V_I$ | 35<br>40 | V<br>V |
| Thermal Resistance Junction-Cases (TO-220) | $R_{\theta JC}$ | 5 | °C/W |
| Thermal Resistance Junction-Air (TO-220) | $R_{\theta JA}$ | 65 | °C/W |
| Operating Temperature Range | $T_{OPR}$ | 0 ~ +125 | °C |
| Storage Temperature Range | $T_{STG}$ | -65 ~ +150 | °C |

## Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit ,0°C < $T_J$ < 125°C, $I_O$ = 500mA, $V_I$ = 10V, $C_I$= 0.33µF, $C_O$= 0.1µF, unless otherwise specified)

| Parameter | Symbol | Conditions | | MC7805/LM7805 | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| Output Voltage | $V_O$ | $T_J$ =+25 °C | | 4.8 | 5.0 | 5.2 | V |
| | | 5.0mA ≤ Io ≤ 1.0A, $P_O$ ≤ 15W<br>$V_I$ = 7V to 20V | | 4.75 | 5.0 | 5.25 | |
| Line Regulation (Note1) | Regline | $T_J$=+25 °C | $V_O$ = 7V to 25V | - | 4.0 | 100 | mV |
| | | | $V_I$ = 8V to 12V | - | 1.6 | 50 | |
| Load Regulation (Note1) | Regload | $T_J$=+25 °C | $I_O$ = 5.0mA to1.5A | - | 9 | 100 | mV |
| | | | $I_O$ =250mA to 750mA | - | 4 | 50 | |
| Quiescent Current | $I_Q$ | $T_J$ =+25 °C | | - | 5.0 | 8.0 | mA |
| Quiescent Current Change | $\Delta I_Q$ | $I_O$ = 5mA to 1.0A | | - | 0.03 | 0.5 | mA |
| | | $V_I$= 7V to 25V | | - | 0.3 | 1.3 | |
| Output Voltage Drift | $\Delta V_O/\Delta T$ | $I_O$= 5mA | | - | -0.8 | - | mV/ °C |
| Output Noise Voltage | $V_N$ | f = 10Hz to 100KHz, $T_A$=+25 °C | | - | 42 | - | µV/Vo |
| Ripple Rejection | RR | f = 120Hz<br>$V_O$ = 8V to 18V | | 62 | 73 | - | dB |
| Dropout Voltage | $V_{Drop}$ | $I_O$ = 1A, $T_J$ =+25 °C | | - | 2 | - | V |
| Output Resistance | $r_O$ | f = 1KHz | | - | 15 | - | mΩ |
| Short Circuit Current | $I_{SC}$ | $V_I$ = 35V, $T_A$ =+25 °C | | - | 230 | - | mA |
| Peak Current | $I_{PK}$ | $T_J$ =+25 °C | | - | 2.2 | - | A |

**Note:**

1. Load and line regulation are specified at constant junction temperature. Changes in $V_O$ due to heating effects must be taken into account separately. Pulse testing with low duty is used.

# APPENDIX F

## LM7805 Data Sheet

## FEATURES

- Complies with JEDEC standard no. 8-1A
- ESD protection:
  HBM EIA/JESD22-A114-A exceeds 2000 V
  MM EIA/JESD22-A115-A exceeds 200 V.
- Specified from –40 to +85 °C and –40 to +125 °C.

## DESCRIPTION

The 74HC/HCT04 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A. The 74HC/HCT04 provide six inverting buffers.

## QUICK REFERENCE DATA

GND = 0 V; $T_{amb}$ = 25 °C; $t_r = t_f \leq 6.0$ ns.

| SYMBOL | PARAMETER | CONDITIONS | TYPICAL | | UNIT |
|---|---|---|---|---|---|
| | | | HC04 | HCT04 | |
| $t_{PHL}/t_{PLH}$ | propagation delay nA to nY | $C_L$ = 15 pF; $V_{CC}$ = 5 V | 7 | 8 | ns |
| $C_I$ | input capacitance | | 3.5 | 3.5 | pF |
| $C_{PD}$ | power dissipation capacitance per gate | notes 1 and 2 | 21 | 24 | pF |

**Notes**

1. $C_{PD}$ is used to determine the dynamic power dissipation ($P_D$ in μW).
   $P_D = C_{PD} \times V_{CC}^2 \times f_i \times N + \Sigma(C_L \times V_{CC}^2 \times f_o)$ where:
   $f_i$ = input frequency in MHz;
   $f_o$ = output frequency in MHz;
   $C_L$ = output load capacitance in pF;
   $V_{CC}$ = supply voltage in Volts;
   N = total load switching outputs;
   $\Sigma(C_L \times V_{CC}^2 \times f_o)$ = sum of the outputs.
2. For 74HC04: the condition is $V_I$ = GND to $V_{CC}$.
   For 74HCT04: the condition is $V_I$ = GND to $V_{CC}$ – 1.5 V.

## FUNCTION TABLE

See note 1.

| INPUT | OUTPUT |
|---|---|
| nA | nY |
| L | H |
| H | L |

**Note**

1. H = HIGH voltage level;
   L = LOW voltage level.

## ORDERING INFORMATION

| TYPE NUMBER | PACKAGE | | | | |
|---|---|---|---|---|---|
| | TEMPERATURE RANGE | PINS | PACKAGE | MATERIAL | CODE |
| 74HC04N | −40 to +125 °C | 14 | DIP14 | plastic | SOT27-1 |
| 74HCT04N | −40 to +125 °C | 14 | DIP14 | plastic | SOT27-1 |
| 74HC04D | −40 to +125 °C | 14 | SO14 | plastic | SOT108-1 |
| 74HCT04D | −40 to +125 °C | 14 | SO14 | plastic | SOT108-1 |
| 74HC04DB | −40 to +125 °C | 14 | SSOP14 | plastic | SOT337-1 |
| 74HCT04DB | −40 to +125 °C | 14 | SSOP14 | plastic | SOT337-1 |
| 74HC04PW | −40 to +125 °C | 14 | TSSOP14 | plastic | SOT402-1 |
| 74HCT04PW | −40 to +125 °C | 14 | TSSOP14 | plastic | SOT402-1 |
| 74HC04BQ | −40 to +125 °C | 14 | DHVQFN14 | plastic | SOT762-1 |
| 74HCT04BQ | −40 to +125 °C | 14 | DHVQFN14 | plastic | SOT762-1 |

## PINNING

| PIN | SYMBOL | DESCRIPTION |
|---|---|---|
| 1 | 1A | data input |
| 2 | 1Y | data output |
| 3 | 2A | data input |
| 4 | 2Y | data output |
| 5 | 3A | data input |
| 6 | 3Y | data output |
| 7 | GND | ground (0 V) |
| 8 | 4Y | data output |
| 9 | 4A | data input |
| 10 | 5Y | data output |
| 11 | 5A | data input |
| 12 | 6Y | data output |
| 13 | 6A | data input |
| 14 | $V_{CC}$ | supply voltage |

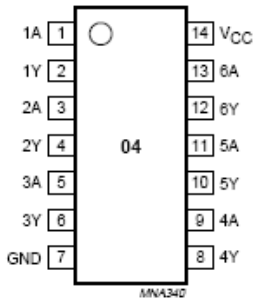
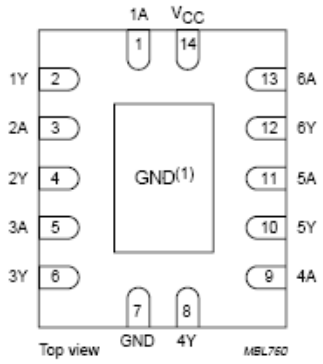



Fig.1 Pin configuration DIP14, SO14 and (T)SSOP14.




(1) The die substrate is attached to this pad using conductive die attach material. It can not be used as a supply pin or input.

Fig.2 Pin configuration DHVQFN14.

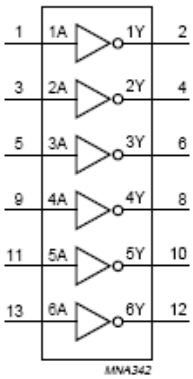



Fig.3 Logic symbol.

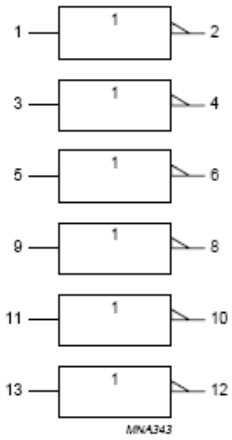



Fig.4 IEC logic symbol.

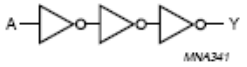


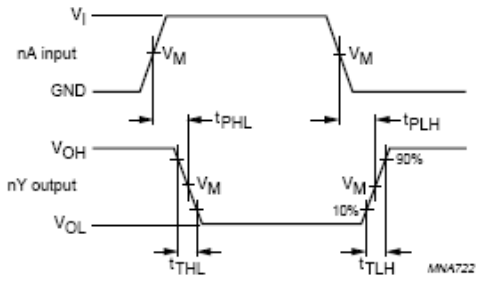

Fig.5 Logic diagram (one inverter).

**Family 74HCT04**

GND = 0 V; $t_r = t_f \leq 6.0$ ns; $C_L = 50$ pF.

| SYMBOL | PARAMETER | TEST CONDITIONS | | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|---|
| | | WAVEFORMS | $V_{CC}$ (V) | | | | |
| $T_{amb}$ = 25 °C | | | | | | | |
| $t_{PHL}/t_{PLH}$ | propagation delay nA to nY | see Figs 6 and 7 | 4.5 | – | 10 | 19 | ns |
| $t_{THL}/t_{TLH}$ | output transition time | see Figs 6 and 7 | 4.5 | – | 7 | 15 | ns |
| $T_{amb}$ = –40 to +85 °C | | | | | | | |
| $t_{PHL}/t_{PLH}$ | propagation delay nA to nY | see Figs 6 and 7 | 4.5 | – | – | 24 | ns |
| $t_{THL}/t_{TLH}$ | output transition time | see Figs 6 and 7 | 4.5 | – | – | 19 | ns |
| $T_{amb}$ = –40 to +125 °C | | | | | | | |
| $t_{PHL}/t_{PLH}$ | propagation delay nA to nY | see Figs 6 and 7 | 4.5 | – | – | 29 | ns |
| $t_{THL}/t_{TLH}$ | output transition time | see Figs 6 and 7 | 4.5 | – | – | 22 | ns |

**AC WAVEFORMS**




For 74HC04: $V_M$ = 50%; $V_I$ = GND to $V_{CC}$.
For 74HCT04: $V_M$ = 1.3 V; $V_I$ = GND to 3.0 V.

Fig.6 Waveforms showing the data input (nA) to data output (nY) propagation delays and the output transition times.