Chapter 1: MCD-DEMO-II Development Board System Introduction

Chapter 2: MPLAB IDE Integrated Development Environment

Chapter 3: Use of On-line Debugger MCD2

Chapter 4: DEMO-II System Functional Modules Details

Chapter 5: Practice of DEMO-II Development Board

MCD-DEMO-II Development Board System Introduction

1.1 Product Overview

MCD-DEMO PIC DEVELOPMENT BOARD II (hereinafter referred to DEMO-II) is a multifunctional PIC microcontroller development platform which has been carefully designed and developed by Qianglongsheng Electronic Technology (Shenzhen) Co., Ltd. based on many years' development experience and the original DEMO-I. It integrated the common external resources and simulation interfaces. Associating with the data and a large number of examples of programs that our company provides, you will be able to fully master microcontroller programming technology in a shortest time. Particularly it is suitable for self learning for microcontroller beginners as well as electronic lovers.

DEMO-II development board together with online debugger PIC16-MCD2 produced by our company, or together with online debugger ICD2 produced by Microchip, can bring you a multiplier benefit.

The following points may illustrate how your choice was wise.

- Rich board resources
- Optimized modular design
- Superior production technology
- Rich supporting examples
- Low selling prices
- Comprehensive technical guidance
- Perfect after service
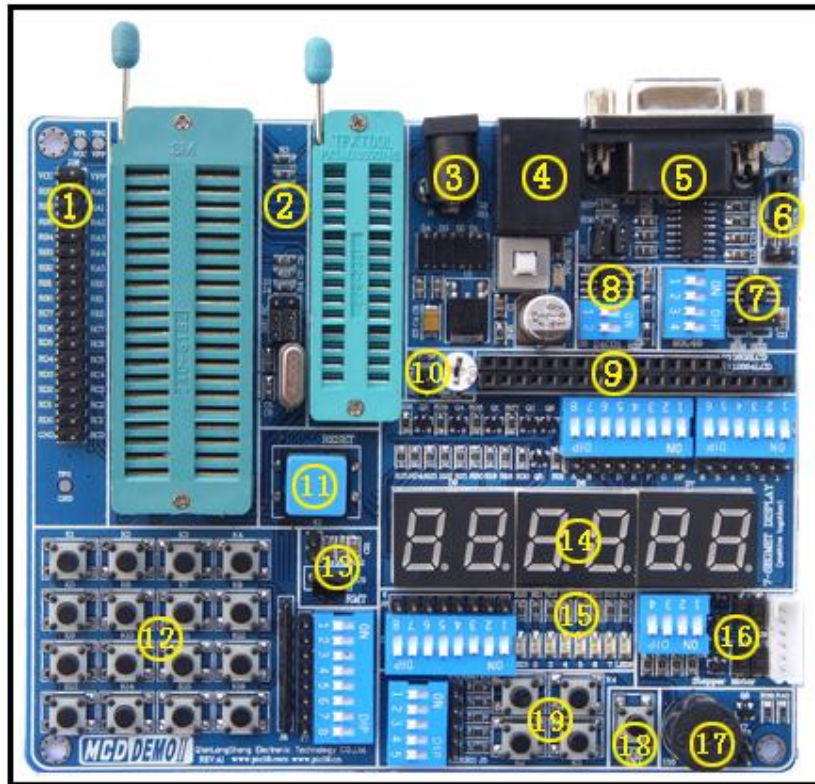
1.2 Board Resource Introduction

1) I/O external output

2) 40P/28P chip socket

3) Power module

4) Simulation interface

5) RS232 communication module

6) DS18B20 module

7) SPI communication module

8) IIC communication module

9) LCD12864 and 1602 LCD module

10) A/D converter module

11) Reset button

12) 4*4 matrix keyboard module

13) Remote control receiver & decoder module

14) six-digital display module

15) 8 on-in-turn light module

16) Stepper motor module

17) Beeper module

18) External interrupt input module

19) Independent button module

1.3 Products Schematic

LCD1602

12864LCD

DS18B20

REMOTE CONTROL

A/D INPUT

RS232 COMMUNICATION

24CXX I2C EEPROM

93CXX SPI EEPROM

STEPPER MOTOR

7-SEGMENT DISPLAY

BUZZER

Title	MCD-DEMO-II BOARD(1)		
Size A3	Number	Revision	A0
Date: 10-Apr-2008		Sheet of	
File: E:\cxq\mcd-demoii_sl\demo.DDB		Drawn By: Zhongруntian	

POWER SUPPLY

PUSH BUTTONS

8*LED

RESET&ICSP

40PIN

28PIN

DIRECT PORT ACCESS

4X4 KEY MATRIX

Title	MCD-DEMO-II BOARD(2)		
Size B	Number	Revision	A0
Date: 10-Apr-2008		Sheet 2 of 2	
File: E:\cxq\mcd-demoii_sl\demo.DDB		Drawn By: Zhongruntian	

Page 4 of 33

Chapter 2 MPLAB IDE Integrated Development Environment

MPLAB IDE (hereinafter referred as MPLAB) is the powerful software integrated development environment provided by Microchip for its PIC microcontroller. It allows users to create, record, edit and comply programs of microcontrollers of PIC series on their own computer systems, and it even can achieves dynamic simulation and debugging and run like virtual exercises.

2.1 The installation of MPLAB

MPLAB is completely free software offered by Microchip. You can obtain the latest installation files through the following two ways.

1), Visit our website: www.pic16.com

2), Visit Microchip's website: www.microchip.com

After downloading the files, you only need to use compression/decompression software tools such as WINZIP to depress and release the files in your computer, and then run SETUP.EXE (or Install.exe) program, and follow a step-by-step installation guide (You may also do not need changes any of the settings, just click "Next") until completion of the installation.

2.2 MPLAB Simple Application

2.2.1 Create a Simple Project

Edit source codes

Click the MPLAB icon at your WINDOWS desktop, or choose Start à All application à Microchip à MPLAB IDE V.xx à MPLAB (Vx.x for MPLAB version) to start running MPLAB integrated environment. Shown as Figure 2-1.
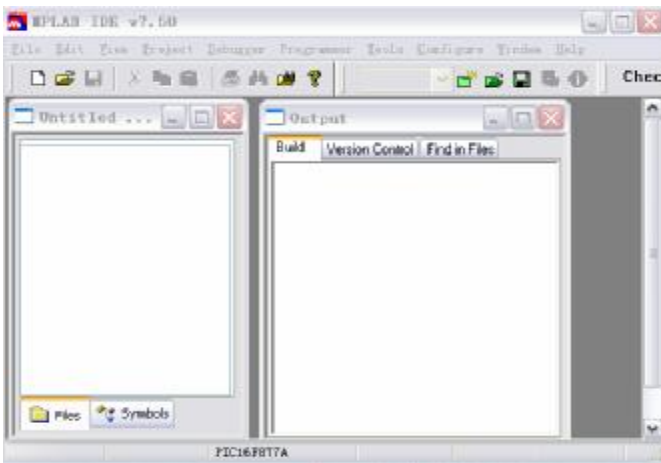
Figure 2-1 MPLAB main window   **Figure 2-2** text editor window

Choose menu command File à new, MPLAB will automatically call MPLAB Editor (source editor), and the work area will have a text editor window, and you can complete input of source code. As shown in Figure 2-2.

Edit the source codes in "Text editor window", then select the menu command File à Save to save to the source file to the specified directory, as shown in Figure 2-3.

Figure 2-3 Save source codes   **Figure 2-4** Project wizard welcome interface

Use the wizard to create project files

step 1, Chose menu command Project→Project Wizard to come to the Welcome interface shown as Figure 2-4.

Step 2, directly click "Next", and select chip model, as shown in Figure 2-5.

Step 3, click "Next", select the appropriate compiler tools according to the source language and chip to be used, as shown in Figure 2-6.

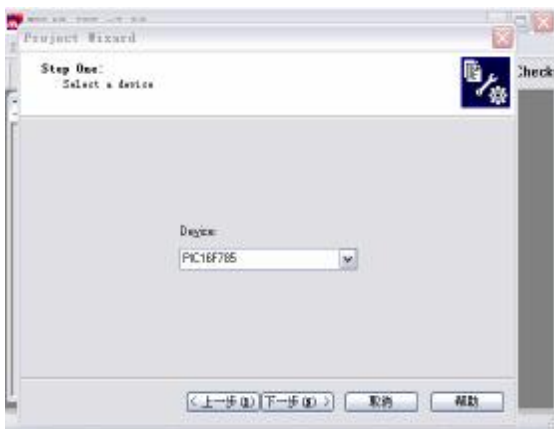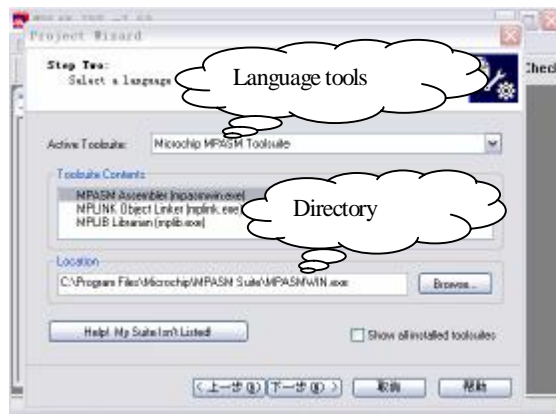**Figure 2-5** Choose chip model   **Figure 2-6** Choose compiler tools

Step 4, click "Next", choose the directory where the project is saved and complete the project name, as shown in Figure 2-7.

Step 5 click "Next", the add source codes to the project, as shown in Figure 2-8.

Figure 2-7 Choose directory to save project   **Figure 2-8** source selection

Step 6, click "Next", as shown in Figure 2-9 to come to tips interface.

Figure 2-9 Tip interface

Step 7, directly click "Done", and exit the wizard.

So far, we have completed establishing a project the source. For more details, please refer to MPLAB Operation Manual.

2.2.2 Debugging the program

Based on the source code edited and project created in above, this section will show you a brief introduction about complying and debugging a program.

Compile

Implement menu command Project à Build All and MPLAB will automatically call the tools mentioned in above Step 3 of establishment of project for you to compile this source code. When completing compile, the interface will be shown as Figure 2-10.
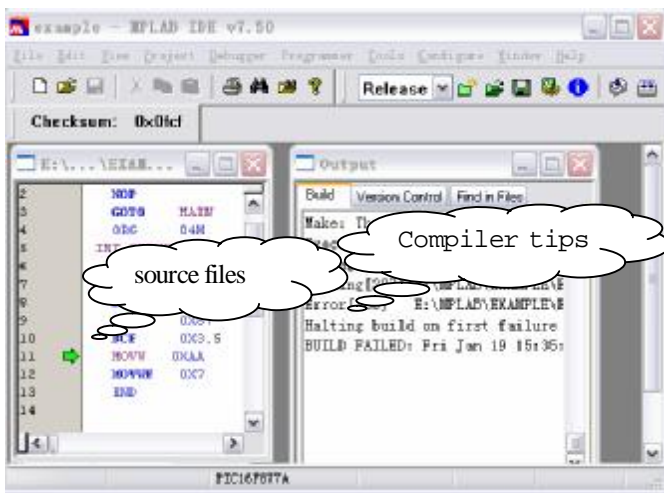

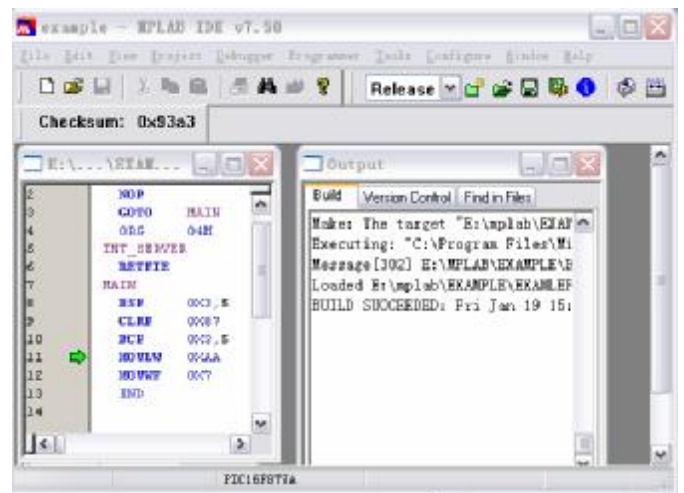**Figure 2-10** Source compiled results

Figure 2-11 Compiling succeeds

From the output window of Figure 2-11, we can see the results that the program compiling failed because of a wrong source, double-click the message, and the cursor will automatically stay at the line where there's an error, and at the margin of most left of this line there's a "Green Arrow". It is obvious that the type of "MOVLW" has been wronged as "MOVW". Now correct the typo and compile it again, the results are shown in Figure 2-11.

Message [302] information tips draws your attention to BANK selection, even if BANK is set up correctly, the messages will still appear, but it does not affect the results of the implementation of program.

If there is an error in the program, it will not generate the target HEX file, to get HEX file you need to correct all errors in the source codes.

Page 7 of 33

Debugging

Debugging program is to test whether the program you deigned is operational, whether it produces correct result as wanted, whether there's any   defects in your design, whether the algorithm design is reasonable, and whether it can accurately control the various hardware resources, and whether it can obtain desired results.

Choose debugging tools

Select menu command Debugger à Select Tool, to select the simulator connected to the PC as the debugging tool, or you may select the software debugger which comes with the MPLAB software as the debugger for the target program. After choice, it will open the corresponding toolbar.

Observe debugging results

The internal storage area of PIC microcontroller can be divided into several sections: program memory, hardware stack, file registers, special function registers and EEPROM data memory. In the course of the operation of program, it will repeatedly read, write or modify the contents in the storage area. Therefore, we can observe the changes of content in storage area corresponding to the operation of program so as to understand the operation of program, and achieve the purpose of debugging. To open storage area we can choose the View menu commands, as shown in Figure 2-12.

Apart from the use of these storage area to observe the debugging process, we can also add the concerned specific modules to the observation window to monitor the results. Implementation the menu command View à Watch and the observation window will open, as shown in Figure 2-13.
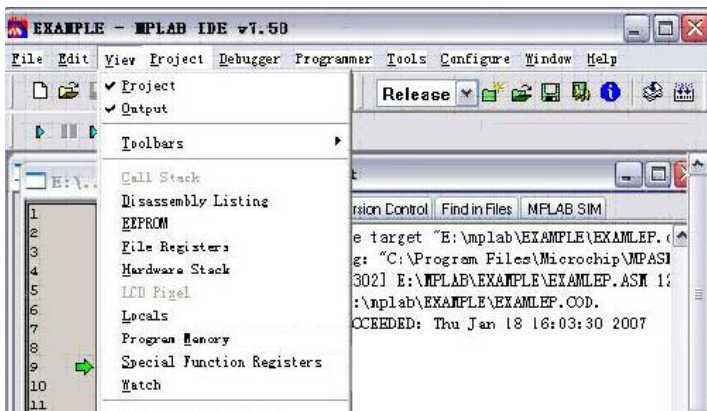

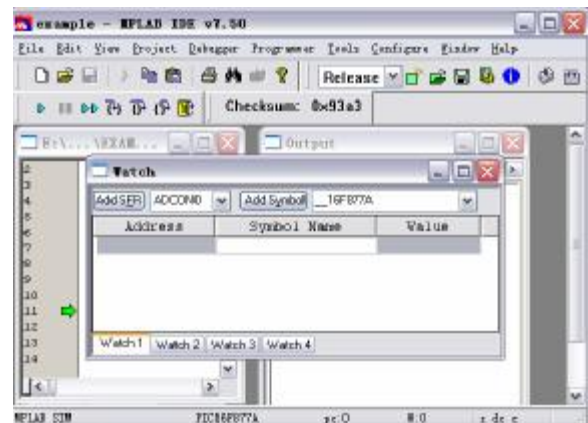**Figure 2-12** Menu command to open storage area   **Figure 2-13** Observation window

Thus, we have introduced a simple use of MPLAB, and through the learning in this chapter, we should be able to complete the source code edit, compile and debug.

Note: For more information on the MPLAB please visit the website of MICROCHIP and download "MPLAB User Manual".

Chapter 3: Use of On-line Debugger MCD2

This chapter briefly introduces the installation of MCD2 , the connection of MCD2 to PC as well as the DEMO-II, and the common troubleshooting. For more detailed information, please refer to MCD2 User Manual.

3.1 Get to Know MCD2

MCD2 is designed and produced by our company. It is fully compatible with MICROCHIP product MPLAB-ICD2. It is a cost saving substitute of online simulator ICE. MPLAB ICD2 can support most of the Flash Technology chips, which can be used not only as debugger but also a chip programming device used in production. It is the only model of chip programming device as well as simulator in China which supports PIC24FJXX/ DSPIC30FXX/ DSPIC33FJXX. And it is the only one to have over 80% of the PIC engineering customers. It is also the most convenient, fastest and highest performance-price ratio for factories during the production of PIC18FXX/ PIC24FJXXDSPIC30FXX and DSPIC33FJXX, as shown in Figure 3-1. (Changes of product appearance or interface alignment will be subject to no prior notice, please follow the actual product you have purchased).

Figure 3-1 MCD2

1 --- RS232 (serial) communication port, through the serial cable provide along with the product you may connect the MCD2 to a PC serial communication port.

2 --- USB reset button to solve the problem that when it fails for MCD2 to get connection to the USB there will be a message ICD0019, ICD0021 requesting you to unplug and then plug the USB again, now with the rest button, you may only press it and wait for 8 seconds to start re-connection and establish communications. (Please do not press reset button when USB communication is ongoing).

3 --- USB communication port, through a USB cable to achieve the high-speed USB and PC communication.

Note: Please do not use MCD2 USB and serial to connect to a PC at the same time.

4 --- ERROR, ERROR indicator.

5 --- BUSY, run / busy indication.

6 --- POWER, power indicator.

7 --- ICSP, download or debugging output ports.

8 --- MCD2, external power port, for AC/DC7.5V-12V, more than 200MA current external power input.

3.2 MCD2 Connections

MCD2 PC and the user target board must be properly connected before it can work properly.

MCD2 and DEMO-II connection

The connection between MCD2 and DEMO-II is shown in Figure 3-2.

input external power supply

Connect MCD2 to DEMO-II
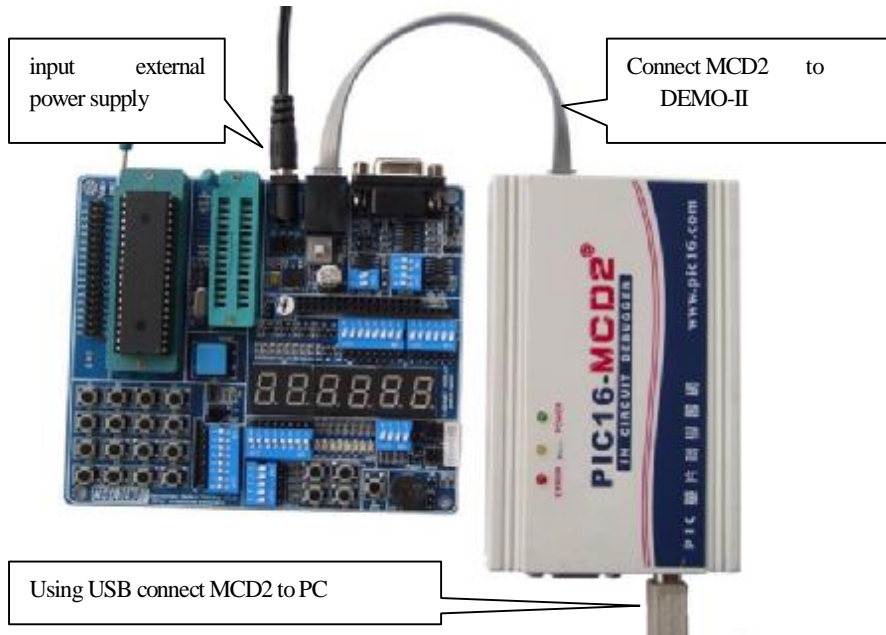
Using USB connect MCD2 to PC

Figure 3-2 MCD2 and DEMO-II connection

Step 1: via USB or serial cable connect MCD2 to a PC.

Step 2: connect an external 9V power supply to DEMO-II.

Step 3: connect the MCD2 and DEMO-II using a 20cm six core flat cable.

MCD2 and PC connectivity

Connect MCD2 to PC serial port through RS232 communication port

Serial communication is relatively slow, and it is generally not recommended to use this method, but in case the computer does not have a USB port or USB port can not be used, this would be the only option.

To use this method you must first set the properties of PC communication port correctly. Specific operations are: Right-click "My computer" à Click "Properties", à Open "System Properties" à Open "Hardware" Option à Click "Device Management" and open the device management window à Show the "Communication Port" and its sub items à Right click the port that connects to the MCD2 and edit the properties of it.

After parameter setting according to the figure shown below, click the "Advanced" button, and get rid of the enable option "use FIFO buffer (requires 16550 compatible UART)".

Save settings, and restart your computer for settings to take effect.

Connect MCD2 to PC with USB port

The USB cable provided along with the product is used to connect MCD2 to PC through USB cable, you may not connect it to the user board at first. Then the system will give out a message that it has found a new hardware, and requires the installation of drivers, now you only need to install the driving program following a step by step guide. Please refer to "PIC16-MCD2 Manual" for specific details, as we are not going to describe it in detail here.

USB driver is located under subdirectory: X: / Program Files / MPLAB IDE/Drivers2000/ICD2_USB (and if the OS is windows98/ME, it will be icd2w98.inf under the subdirectories X: / Program Files / under MPLAB

IDE/Drivers98/ICD2_USB. If the OS is Windows XP, this INF file locates in X: / Program Files / MPLAB IDE / Drivers XP/ICD2_USB under) (here X refers to the IDE C or D where the MPLAB is installed) Note: Please do not connect the MCD2 and PC through USB before installation of MPLAB IDE 6.XX.

Install and configure MPLAB according to MCD2

Generally please follow the below procedures to use MCD2:

1) Connect MCD2, PC and user objective board according to Figure 3-2.

2) Click on the Desktop icon MPLAB to start MPLAB software.

3) Choose the model of microcontroller that is used on the target board.

4) Configure the target chip with bit information.

5) Choose MPLAB ICD2 as a debugging tool, or programming tool.

6) Choose the correct communication port according to the actual connection between MCD2 and PC as well as the target board power supply mode.

7) Complete the debugging and programming of program using MCD2

Full procedures of using MCD2

Step 1, in accordance with the methods described in the chapter "MCD2 and DEMO-II Connection" to set up MCD2 and DEMO-II connection.

 Step 2, plug the MCU chip into the corresponding socket on the board.

Step 3, activate MPLAB software, and open the project file that needs debugging.

Step 4, implement the menu command Configure à Select Device to select the model of target MCU chip.

Step 5, implement the menu command Configure à Configuration Bits to configure related Bits.

Step 6, implement the menu command Debugger à Select Tools to select the tool as MCD2 debugger.

Step 7, implement the menu command Debugger à Settings to select communication port settings and DEMO-II power supply method.

Step 8, implement the menu command Debugger à Connect to establish the communication between MCD2 and DEMO-II.

Step 9, implement the menu command Debugger à Program to burn the target MCU chip with debugged program.

Step 10, implement the menu command Debugger à Run   or other operation mode to debug the target program.

Step 11, complete debugging and disconnect MCD2 and DEMO-II.

3.3 MCD2 FAQ

1, ICDWarn0020: Invalid target device id……?

In this issue, please confirm the following:

1) the chip board is correct.

2) whether the communication between the chip and MCD2 is good or not (whether there is a 5-pin multiplexing).

3) whether there are components such as capacitor and diode on VPP pin.

4) whether the power settings are correct or not (whether the voltage on the target board is normal)

2, ICDWarn0019: Debug mode is not enabled……? (Unable to enter debug mode)

In this issue, please confirm the following:

1) whether the communication between the chip and MCD2 is good or not

2) whether the background debugging options in the configuration word of the chip is activated or not.

3) whether the oscillation or reset functions of the chip work normal or not.

4) whether the target board power supply is in good condition.

5) whether the codes are write-protected.

3, ICD0019: Communication: Failed to open port…….? (Can not open communications ports)

In this issue, please confirm the following:

1) Make sure that the hardware works properly and the connectivity works properly. If you use USB interface, please make sure the USB driver can function well (you may check the ICD2 connection conditions in the "Hardware Device Management", and you may disconnect them for a few seconds and reconnect them.

2) ICD2 linking order: connect the hardware -> start MPLAB IDE software

3) in this situation you may try to re-download operating system (debugger-Download Icd2 Operating System)

Note:

For more information please refer to the "PIC-MCD2" Manual

Chapter 4: DEMO-II System Functional Module Details

This chapter will describe in more detail on DEMO-II development board and the functional modules by illustrating the schematics of the modules and the points that needs to pay attention to during usage. And in the CD-ROM that is provided along with the product, we have included a number of examples prepared by using such functional modules, and these examples cover the layout of the hardware and software, and the source code with detailed annotation in Chinese as well as the executed results, which aids to the reference especially for beginners. As the experimental board is modular in design, the configuration of your project may diverse in large amount, and the output ports are all available for external resources. The hardware and program can be at reader's own will, so readers may draw inferences about other cases from one instance.

4.1 All I/O External Output Module

All I/O resources on DEMO-II development board are designed for external output. Users can build their own circuit by taking advantage of the existing resources of DEMO-II. As shown in Figure 4-1.

This module consists of the following main components:

1) 40 pin chip all I/O (PORT A/B/C/D/E)

2) serial programming voltage VPP.

3) power supply VCC and GND.

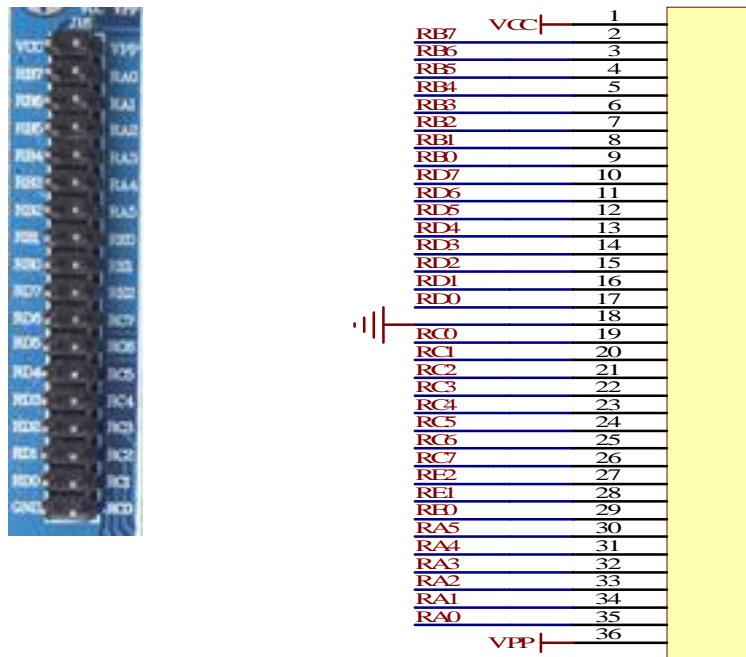**Figure 4-1** all I/O and external output schematic

4.2 Chip Socket and Clock Selection

The module consists of the following major components:

1) 40pin chip socket.

2) 28pin chip socket.

3) The system clock selection (crystal or external RC).

   DEMO-II can support all 40pin and 20pin PIC16FXXX and PIC18FXXX chips whose pins are compatible with PIC16F87X, as shown in Figure 4-2.

   Short jump to the lower terminal of J3, J4 (close to the crystal OSC), then you have selected the OSC as the system clock, while if you short jump the other two terminals you will have selected an external OSC as your system clock. The crystal OSC will be using plug in and out method so that the users may change OSC with different frequency by their own choice. See Figure 4-3.
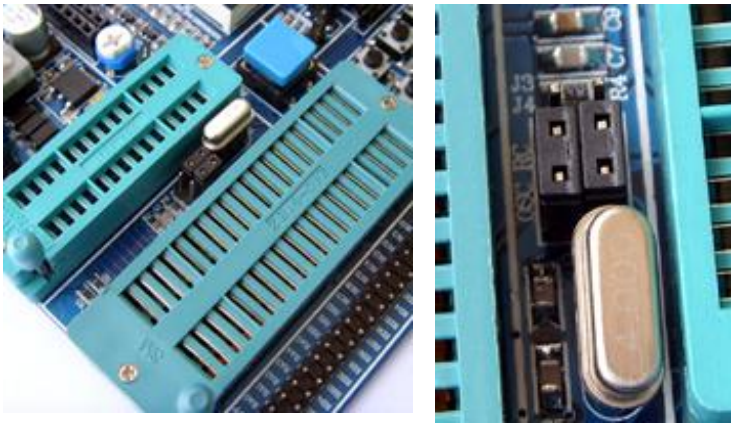
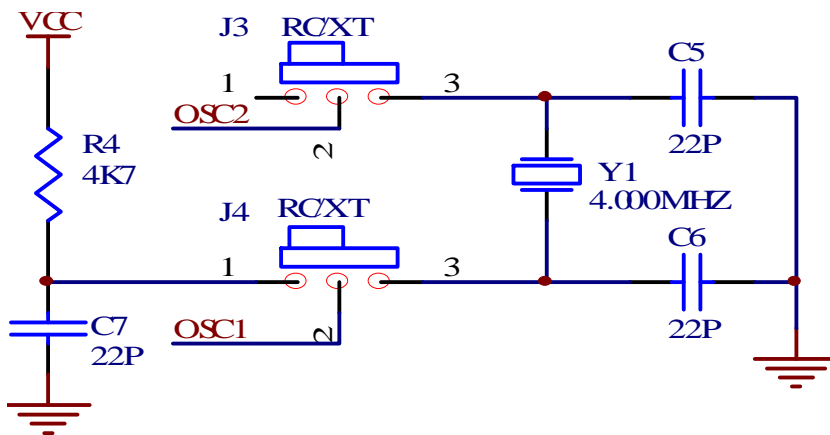**Figure 4-2** 40pin/28pin chip and system clock selections

Figure 4-3 system clock selection schematic

4.3 Power Module

The Power Development Board adopted the bridge rectifier circuits, we need not consider the polarity of external power, the output voltage range is 7.5-9 V and the DC or AC power with current more than 200 mA can be used directly. The module is shown in Figure 4-4.

Press the power switch and the whole board power supply is connected, while otherwise the power supply is disconnected.

The module consists of the following major components:

1), power input.

2), rectifier, filter, regulator.

3), the power switch.

4), power indicator.

Figure 4-4 power modules
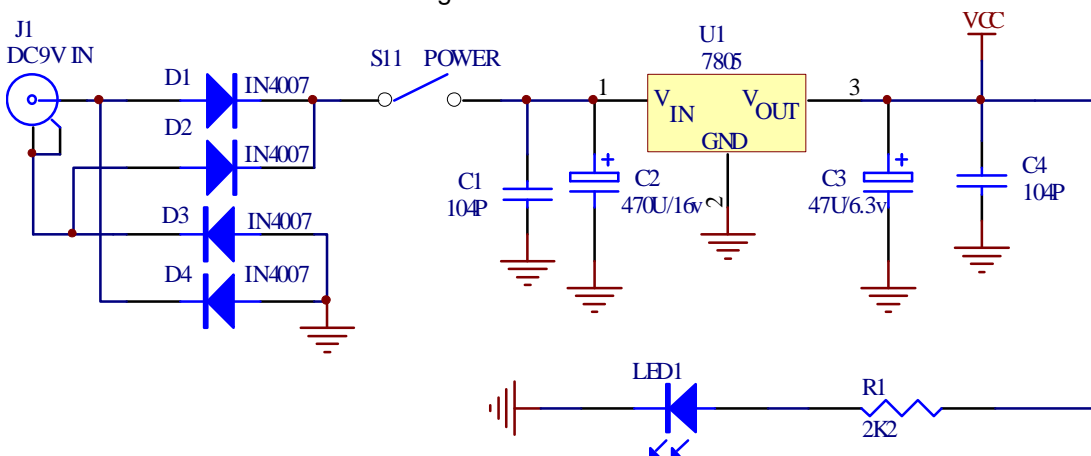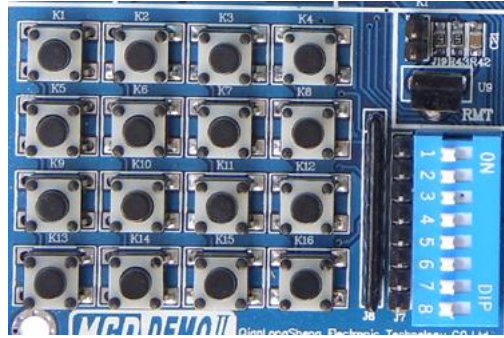
Power module schematic shown in Figure 4-5.

Figure 4-5 schematic power modules

4.4 4*4 Matrix Keyboard

The module has the following main components:

1) 16 keys.

2) coding switching.

3) interface socket.

4) up-pull resistance.

4x4 matrix keyboard

Descriptions on this module are as below:

1) 16 keys connected to the 8 pins in accordance with the mode 4x4 PORTC.

2) the module is controlled by the coding switch. When it is not used, it is recommended to disconnect the switches so as not to affect other modules.

3) you can practice the application by using the keyboards from other matrix through interface socket (coding switch must be at a disconnected status).

4) there's a 10K Omiga up-pull resistor to ensure that the level of voltage keeps stable.

5) in the CD-ROM provided along with product, there're simple examples about "4x4 keyboard matrix".
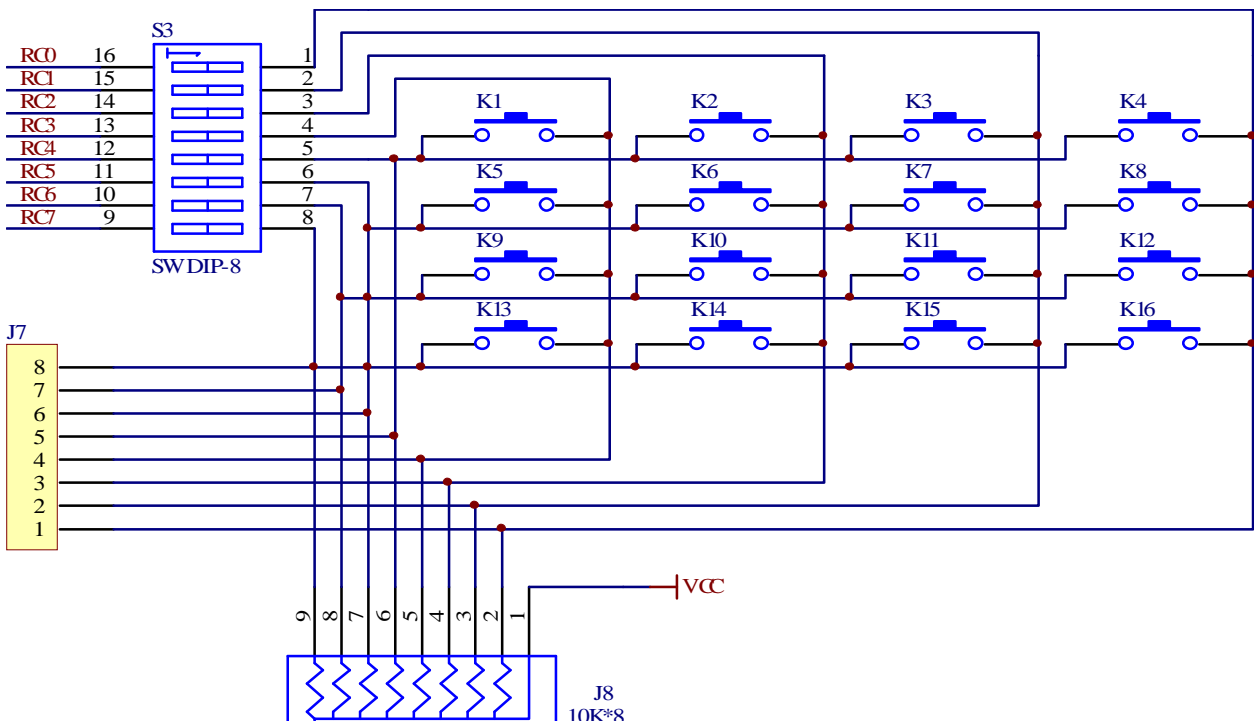
4*4 matrix keyboard schematic

4.5 Simulation Interface and Reset Button

Simulation interface is reserved for user to connect MCD2 to a simulator. The interface completely follows MCD2/ICD2 interface requirement, user may need to do no changes when using it. It is shown in Figure 4-6. Reset button is used to reset the MCU, as shown in Figure 4-7.
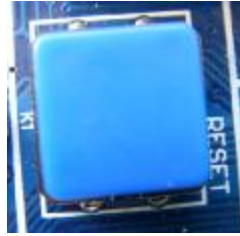

Figure 4-6 Simulation interface   **Figure 4-7** Reset button

Simulation Interface and reset button schematics are shown in Figure 4-8.
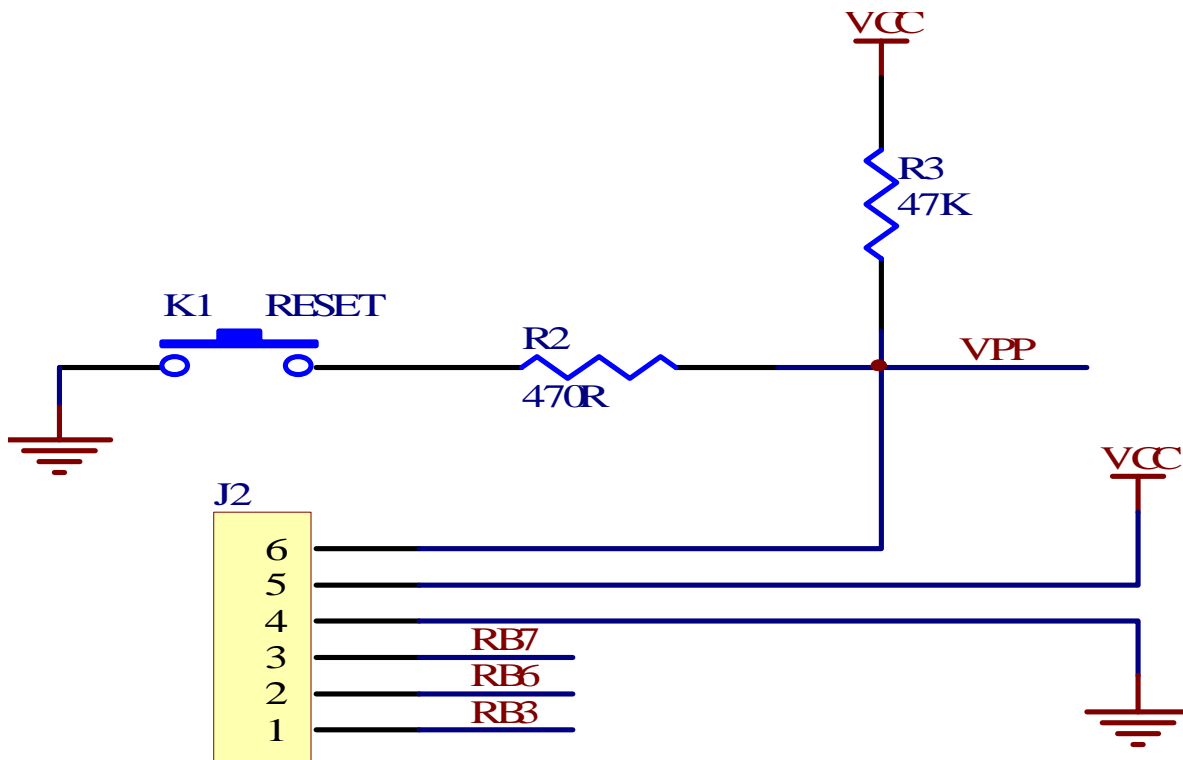

Figure 4-8 Simulation interface and reset button schematics

4.6 RS232 Communication Module

The modules mainly shows how for the MCU to communicate with external devices such as PC by using USART module , as shown in Figure 4-9.

The module consists of the following main components:

1) RS232 voltage level converter chip

2) RS232 communication port (9-pin serial port)

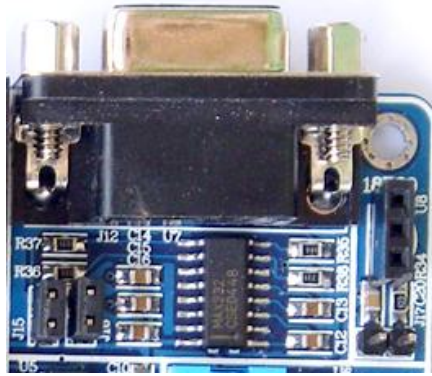3) two jumper wires J15 and J16

Figure 4-9 RS232 communication

This module is described as below:

1) the serial module connected through two jumper wires to the USART module of MCU at the interface of RC6 and RC7 ports.

2) two jumper wires control the connection of serial module and MCU. When this module is in use, we must ensure that access jumper wire is in the connected state, when not in use, we must ensure that the jumper wire is in disconnected status.

3) CD-ROM provided along includes the examples about this module for reference.
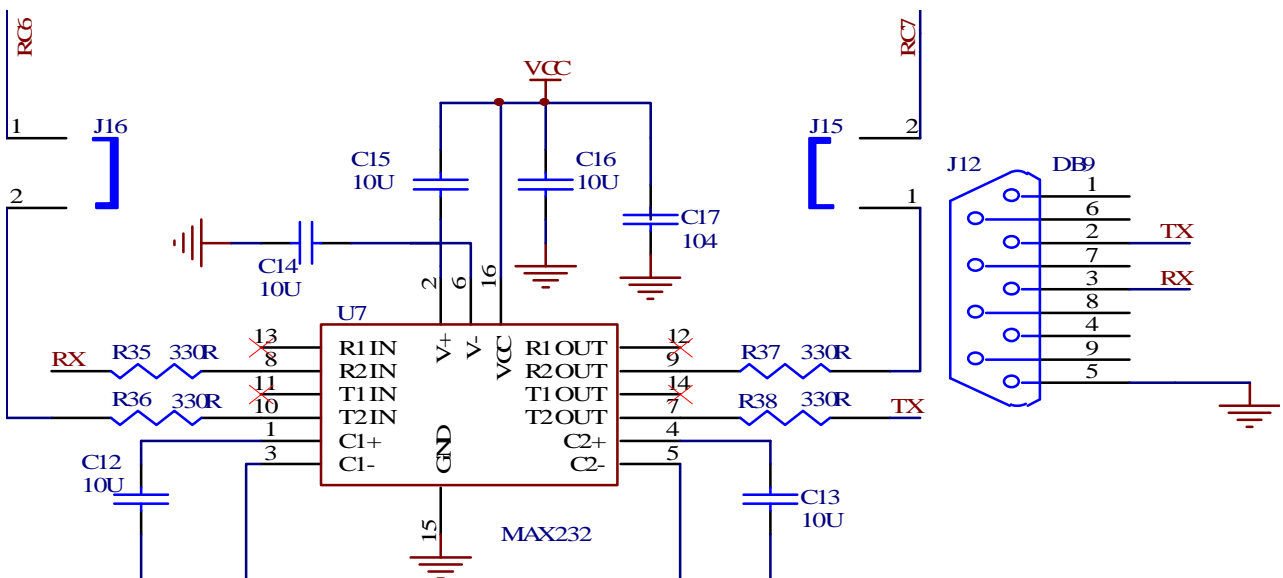
The schematic of this module is shown in Figure 4-10.

Figure 4-10 RS232 communication module schematic

4.7 DS18B20 Temperature Module

This module mainly shows the use of temperature sensors DS18B20, as shown in Figure 4-11.

The module consists of the following major components:

1) Temperature Sensor DS18B20 socket.

2) Jumper wire J17 (to be used as a switch).

Descriptions on this module:

1) DS18B20 is an optional product to be purchased.

2) DS18B20 information included in the CD-ROM provided along with product.

3) DS18B20 output by connecting jumper wire J17 to the RA2 pin of the MCU.

4) When this module is in use, we must ensure that access jumper wire is in the connected state, when not in use, we must ensure that the jumper wire is in disconnected status.

Figure 4-11 DS18B20

5) examples about DS18B20 were included in the CD-ROM for reference.

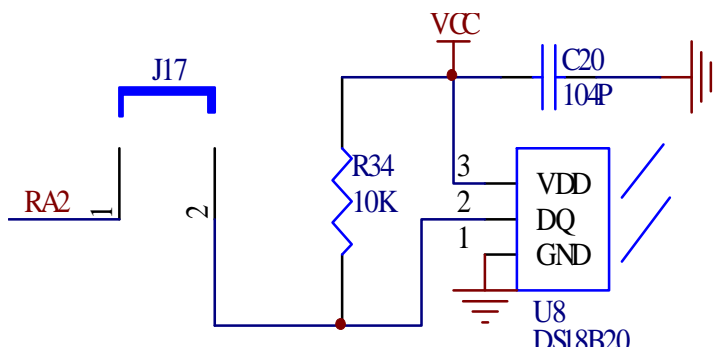The schematic of this module is shown in Figure 4-12.

Figure 4-12 schematic of DS18B20

4.8 SPI Communication Module

This module mainly shows the SPI communication protocol through access to the external device EEPROM 93LCXXX, shown as Figure 4-13.

Figure 4-13 93LCXXX EEPROM

The module consists of the following components:

1) 93LCXXX EEPROM.

2) coding switch.

3) ROM WORD SIZE selection jumper wire.

Descriptions on this module as below:

1) 93LCXXX EEPROM data are included in the CD-ROM.

2) the SPI communication port of 93LCXXX: SDI, SDO and SCL are connected to the RC5, RC4 and RC3 of the communication port of MCU through the coding switch, and the chip-select signal connects to the RC2 pin of the MCU, and therefore, it can be controlled by hardware.

3) When this module is in use, we must ensure that the coding switch is in the connected state, when not in use, we must ensure that it is in disconnected status so as not to affect the normal operation of other modules.

4) When using 93LCXXA (ROM SIZE is 8 bits), or 93LCXXB (ROM SIZE is 16 bits), the WORD SIZE jumper wire is non-functional. When using 93CXXC, jumper wire decides to choose WORD SIZE for 8 or 16 bits.

5) 93LC46B EEPROM related program examples were included in the CD-ROM for reference.
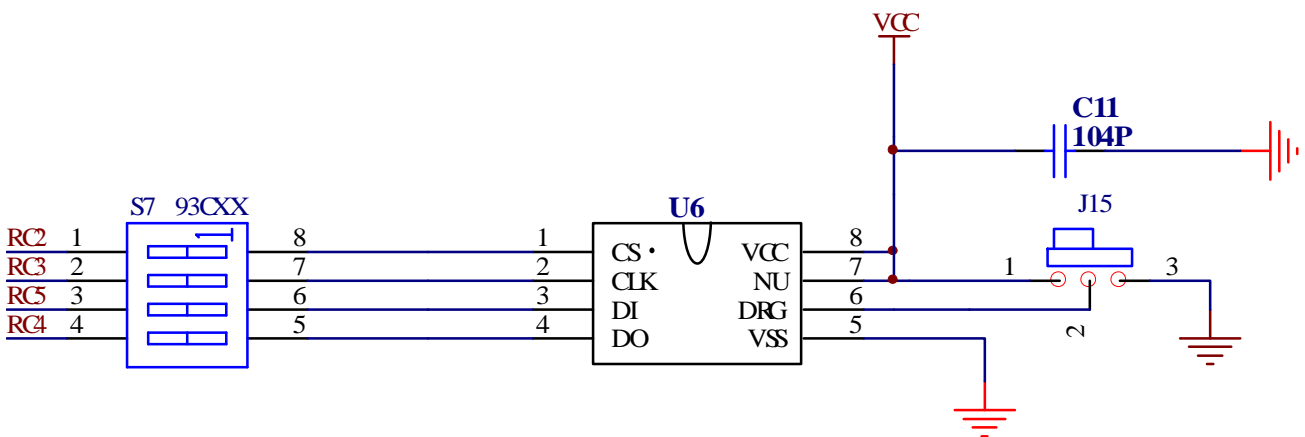
The module of the schematic shown in Figure 4-14.

Figure 4-14 schematic of SPI Communication

4.9 IIC Communication Module

This modules mainly shows the ICC protocol through access to the external device EEPROM 24CXX, as shown in Figure 4-15:

The module has the following main components:
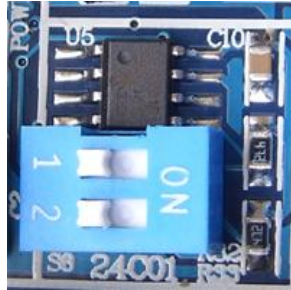
1) 24CXX EEPROM

2) double bit coding switch

Figure 4-15 24CXX EEPROM

Descriptions on this module are as below:

1) information about 24CXX EEPROM was included in the CD-ROM.

2) the IIC communication port SDA of 24XX is connected to the IIC communication port RC4 and RC3 of MCU CK to allocate yards and switch connected to the MCU so it can be controlled by hardware.

3) When this module is in use, we must ensure that the coding switch is in the connected state, when not in use, we must ensure that it is in disconnected status so as not to affect the normal operation of other modules.

4) Programs about read and write of 24 C01B EEPROM is included in the CD-ROM for reference.
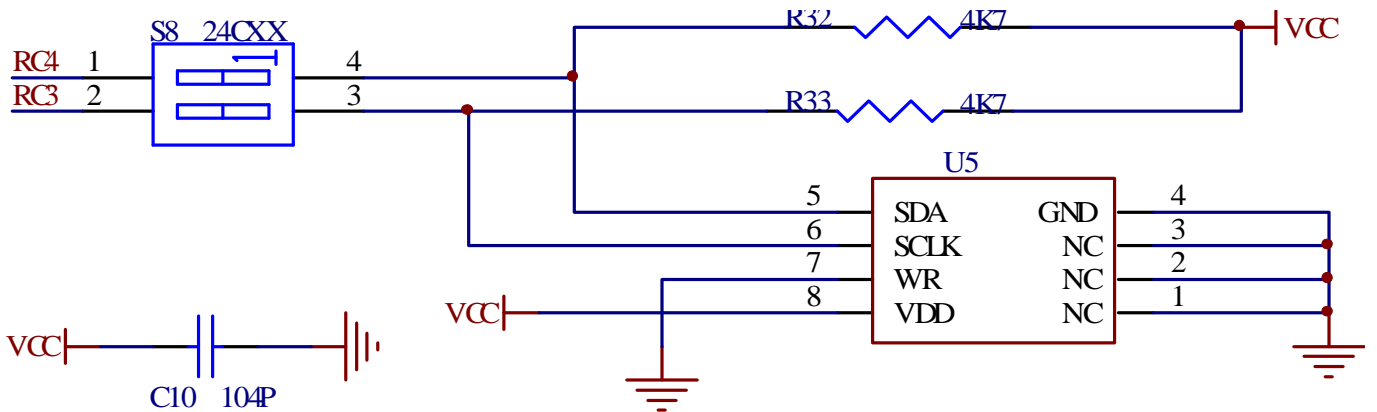
The schematic of this module is shown in Figure 4-16.

Figure 4-16 IIC communication schematic

4.10 LCD12864 and 1602 LCD Module

This modules majorly includes LCD12864 socket and 1602 LCD socket, as shown in Figure 4-17.

Figure 4-17 LCD12864 and 1602 LCD sockets

Descriptions of this module are as following:

    1) LCD 12864LCD and 1602 are using PORTA as the control bit, PORTD as data bit. Only the MCU which has PORTD can execute LCD display.

    2) All the pins of this module are connected to the MCU directly, and there's no coding switch to control it. When the LCD is not in use, it is suggested that the LCD is taken off from the socket, while if it is in use, we should shut off other modules, or else the LCD will not display.

    3) The development board can not adjust the brightness of 1602 LCD or the contrast of LCD12864.

    4) All LCD used in this development board are products of "Jinghanda Electronics Companies", if you use other products, please confirm whether or not it is compatible.

5), Data about 1602LCD and 12864LCD are included in the CD-ROM.

6) Application examples about 1602LCD and 12864LCD are included in the CD-ROM.
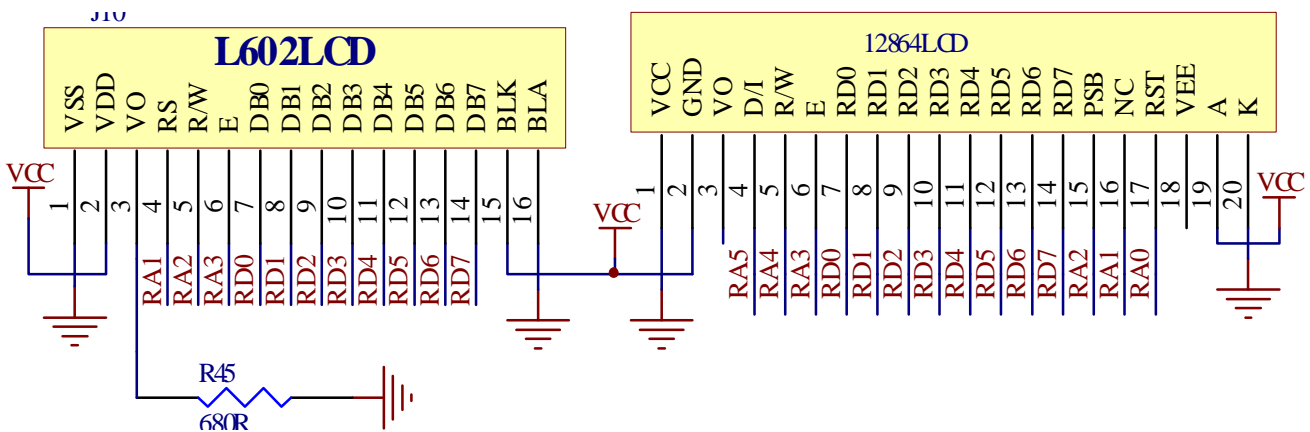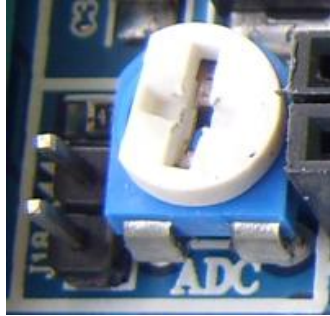

Schematics of these two module are shown in Figure 4-18.

Figure 4-18 schematics of 12864LCD and LCD1602

4.11 A/D Converter Module

This module main executes the conversion from analog signal to digital signal.

A/D Converter Module

This module mainly consists of the following components:

1) One 1K adjustable potentiometer.

2) Jumper wire J18.

Descriptions of this module are as the following:

1) The potentiometer is connected to the RA0 port through jumper wire J18.

2) When the A/D converter is in use, please make sure that the jumper wire is on while if it is not in use please sure it should be cut off so as not to affect Other modules.

3) CD-ROM provided along with the product includes examples of A/D converter for reference.
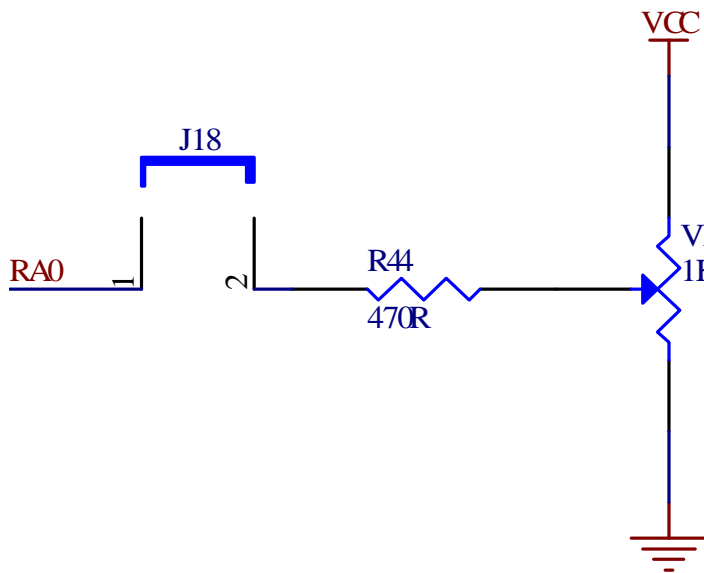
Schematic of this module is shown in Figure 4-19.

Figure 4-19 A/D converter schematic

4.12 Remote Control Receiver and Decoder Module

This module main executes the receiving and decoding regarding infrared remote-control, as shown in Figure 4-20.

Figure 4-20 Remote decoding module

This module consists of the following major components:

1) Infrared remote control receiver

2) Interface Socket

Descriptions on this module are as the following:

1) Information about infrared remote encoding chips 6121 is included in the CD-ROM.

2) The output of the remote control receiver is connected

to the RA1 port of MCU through a jumper wire.

3) When this module is in use, please make sure that the jumper wire is on while if it is not in use please make sure the jumper wire is disconnected.

4) The CD-ROM contains examples about remote decoding for reference.
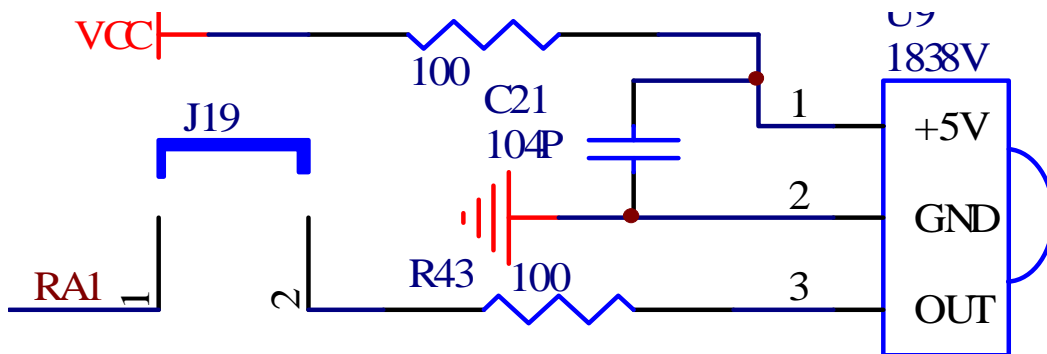
The schematic of this module is shown in Figure 4-21

Figure 4-21 Remote decoder schematic

4.13 six digital control module

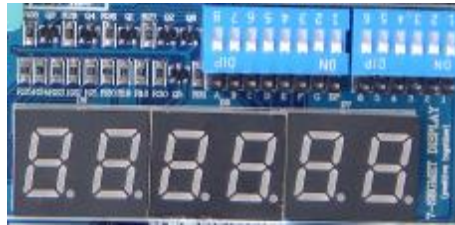This module main introduces the use of multi-digital LED display, as shown in Figure 4-22.

Figure 4-22 digital LED control module

The module consists of the following major components:

1) 6 digital LED.

2) Bit control and segment control coding switch.

3) Driving circuit.

4) Interface socket.

Descriptions of this module are as the following:

1) The segment control of the digital LED is connected to the PORTD of MCU through coding switch.

2) The bit control of the digital LED is connected to the PORTA of MCU through coding switch.

3) When this module is in use, we must ensure that the coding switch is in the connected state, when not in use, we must ensure that it is in disconnected status so as not to affect the normal operation of other modules.

4) You may also achieve the display of digital LED by use of interface socket. (And at this point you must ensure that the coding switch is disconnected.)

5) The six digital LED are using a common anode.

6) The CD-ROM contains examples for reference. (One static display and one dynamic display examples are include respectively).
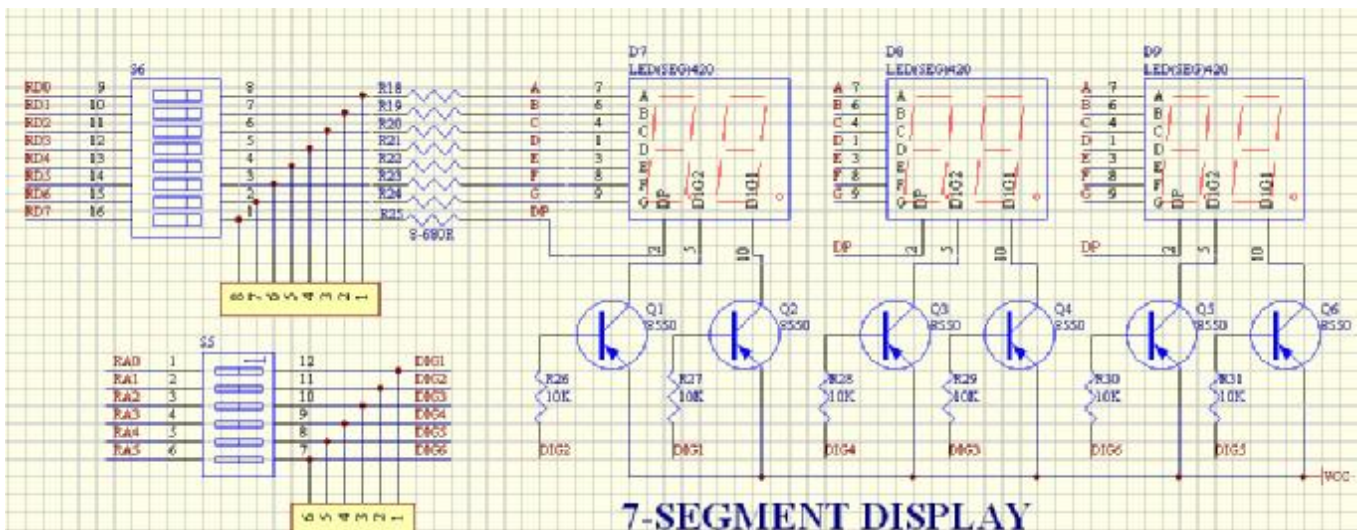
The schematic of this module is shown in Figure 4-23.

Figure 4-23 Digital LED schematic

4.14 8 Play-in-turn Light Module

This module is shown in Figure 4-24. It consists of the following main components:

1) 8 LED.

2) 8 coding switches.

3) Interface socket.

Figure 4-24 8 Play-in-turn Light Module

Descriptions of this module are as the following:

1) 8 LED are on when the I/O outputs HIGH voltage.

2) Each LED is controlled by a separate coding switch. When this module is not in use it is suggested that the

corresponding bit be disconnected so as to avoid causing unnecessary disruption.

3) You may test the I/O ports of other MCU by using the interface socket.

4) The CD-ROM includes some simple examples of this module, which cover "lit each LED" a "Simple play-in-turn", just

for users' reference.

The schematic of this module is shown in Figure 4-25.
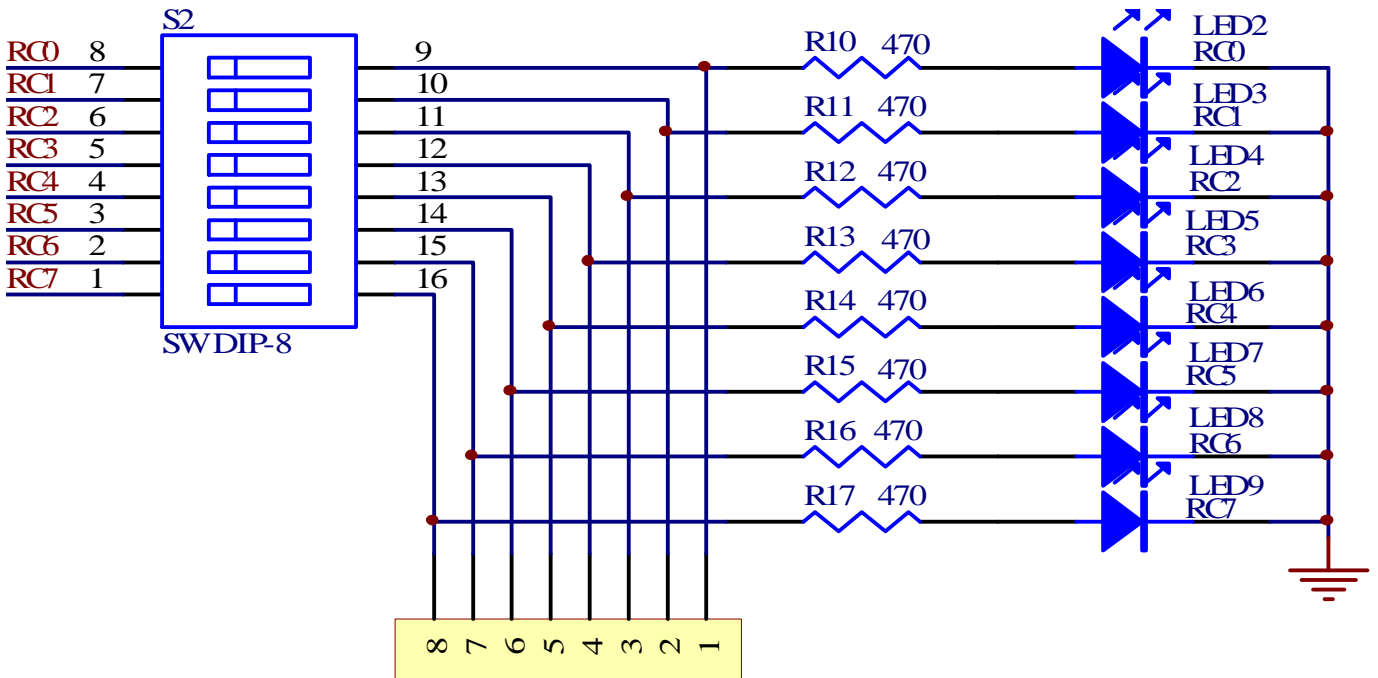

Figure 4-25 Schematic of play-in-turn

4.15 Stepper Motor Module

This module mainly shows the operation of stepper motor. Through the study of this section, we should understand the relevant knowledge about stepper motor. This module is shown in Figure 4-26.

The module consists of the following components:

1) Stepper motor interface.

2) Drive circuit.

3) Coding switch.

4) Current-limiting fuses.

Figure 4-26 Stepper motor module

Descriptions on this module are as the following:

1) The stepper motors used in this development board are products 35BY48S053 and the compatible products of "Changzhou City Fengyuanweite Motor Co., Ltd.". If you use of other motor, please check the number of phases and the electrodes order.

2) The driving circuit uses a simple FET to drive.

3) As the current of stepper motor is quite big, it is suggested that the coding switch is disconnected when this module is not in use so as to avoid the impact on other modules.

4) The CD-ROM includes examples of the use of steppers for reference. It is also including speed control, direction control and stepping distance control.

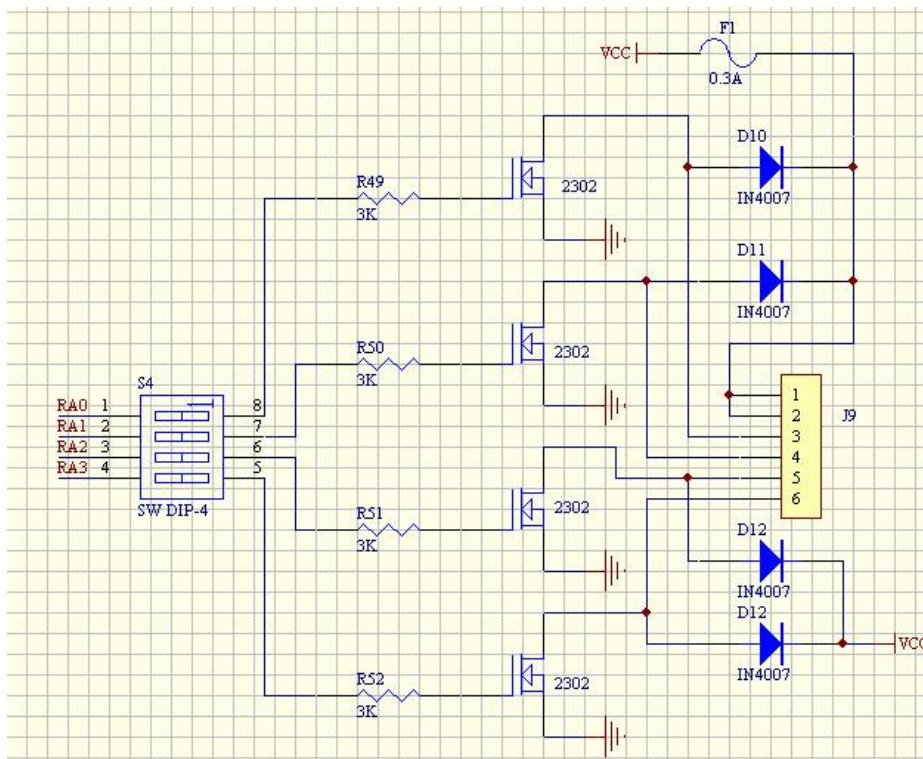The schematic of this module is shown in Figure 4-27.

Figure 4-27 Stepper motor module schematic

4.16 Beeper Module

This module mainly shows how to use MCU to control the beeper, as shown in Figure 4-28.

The main module of the following components:

1) Beeper

2) Jumper wire J17

Figure 4-28 Beeper module

Descriptions about this module are as the following:

1) The beeper is connected to the RC2 pin of MCU through a jumper wire.

2) When this module is in use please make sure the jumper wire is connected while when it is not in use please make sure that the jumper wire is disconnected.

3) When the jumper wire J17 is on connected status, we may hear a weak beeping even if there's no chip on the board, which is a normal phenomenon.

4) The CD-ROM includes programs of beeping for reference.
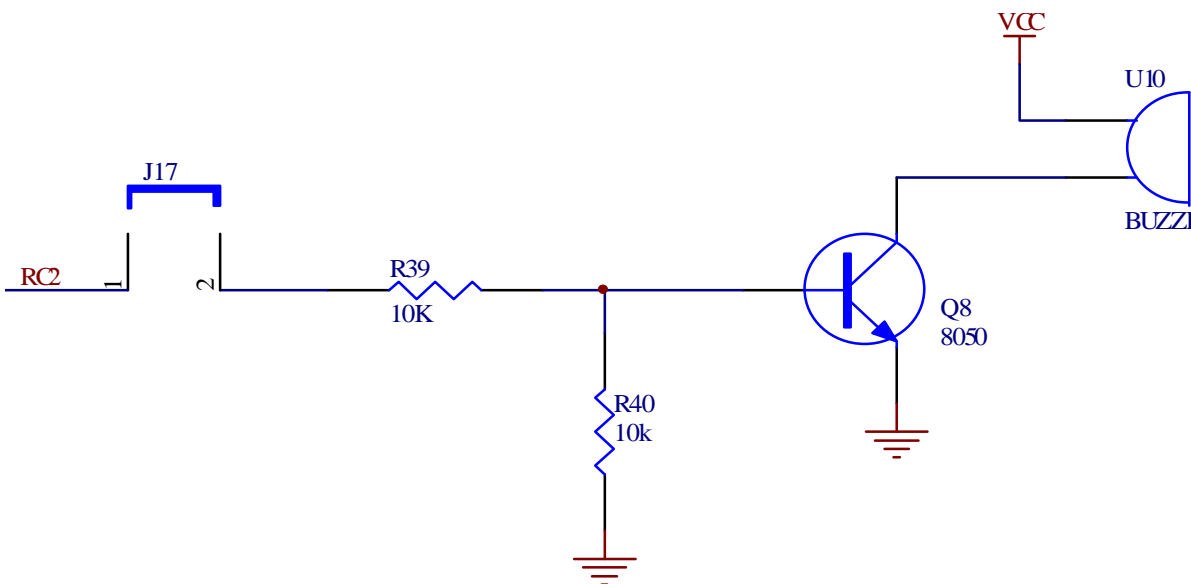
This module schematic shown in Figure 4-29.

Figure 4-29 Schematic of beeper

4.17 Independent Key and External Interrupt Module

This module is independent key and external interrupt module, as shown in Figure 4-30.

**Figure 4-30** Independent Key and External Interrupt Module

This module consists of the following main components:

1) Five independent buttons.

2) One five-bit coding switch.

3) Interface socket.

Descriptions of this module are as the following:

1) All keys are at LOW voltage when they are pressed down, and are at high-impedance when they are released (when they are in use, we should open the internal up-pull resistance of the MCU, so that when the keys are released they are at HIGH voltage.)

2) K1 not only can be used as ordinary key, but it can also be used to trigger external interrupt.

3) All the keys are connected to the Port B of the MCU, when they are in use please make sure that the coding switches are connected, and when they are not in use please make sure that the switches are not connected.

4) When using Port B as keys, it is suggested that the weak up-pull function of Port B be enabled, while other ports are used it is also suggested that up-pull resistors are added to the board.

5) You may also use the keys of other MCU by taking advantage of the interface socket.

6) The CD-ROM contains examples about the use of this module.

The schematic of this module is shown in Figure 4-31.
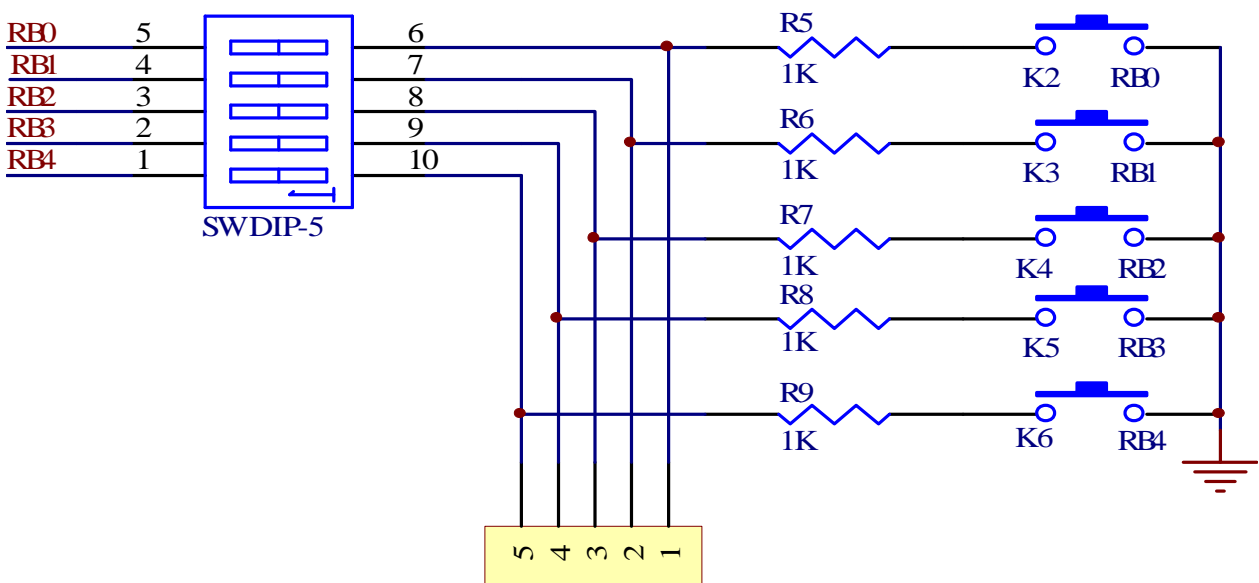
Figure 4-31 Keys and external interrupt module

Chapter 5 Practice of DEMO-II Development Board

Actual practice purposes:

This chapter takes the "4 * 4 matrix keyboard module" and the "six digital control module" as the basis, and have systemic introduction about the use of DEMO-II development board, including the use of MPLAB edit, compiling source and MCD2 debugging results.

Hardware layout:

1) 4x4 matrix keyboard controlling the coding switches S3 which are all on.

2) The 1st and 2nd position of 6-bit digital LED coding switches are both on, while other bits are off (we use only the 5th and 6th digital LED to display.)

3) All the coding switches that controls the six LED are ON.

Software planning:

1) In this software we ignore the jitter of keys, when the software detects LOW voltage, it would consider that there's a key press.

2) In this software, we do not consider the situations when several keys are pressed down simultaneously. When there are several keys being pressed, we consider that the key which has the smallest number is pressed, for example, when K1 and K2 are pressed at the same time, we only consider that K1 is pressed.

3) When there is no button being pressed, the two digital LED display as "FF", when a key is pressed, the LED will show the number of key that is pressed, for example, when K1 is pressed, the LED's display as "01", while K16 is pressed, LED's display as "16."

Prepare and compile the source codes:

Double-click MPLAB icon on the Desktop to run MPLAB programming environment.

1) In accordance with the method introduced in 2.2.1, edit a new source code, and save it as "KEY4x4.asm".

2) In accordance with the method introduced in 2.2.1, establish a new engineering project, and complete the settings as below: select PIC16F877A in the second step as the target chip; select "Microchip MPASM Toosuilte" as the compiling tool in the third step; select "KEY4x4" as the new project name (the suffix can be omitted), and the directory for saving the project is the same as that of the source. After editing the source and establishing the project, MPLAB interface will be shown as Figure 5-1.
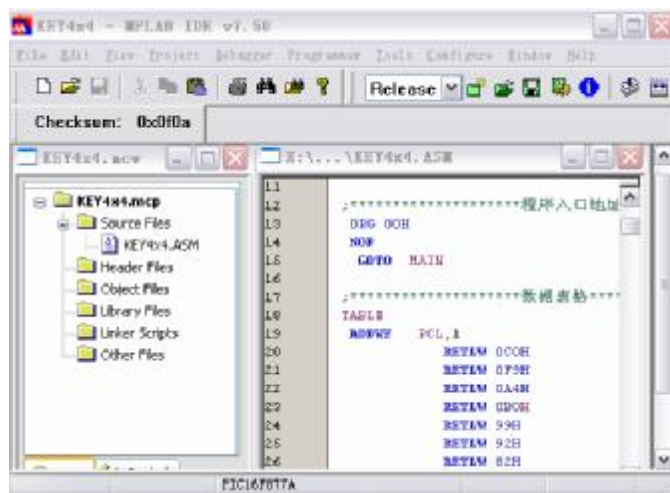

Figure 5-1 MPLAB interface after editing source code and establishing the project

3) Implement the menu command Project à Build ALL or the icon [icon] in the tool bar to compile the source code. The compiled results will be shown as Figure 5-2. We can see from the figure as "BUILD SUCCEEDED", which indicates that the compiling is successful and it has created a KEY4x4.HEX file under the directory of the project (only compiling succeeds can it create such a file).
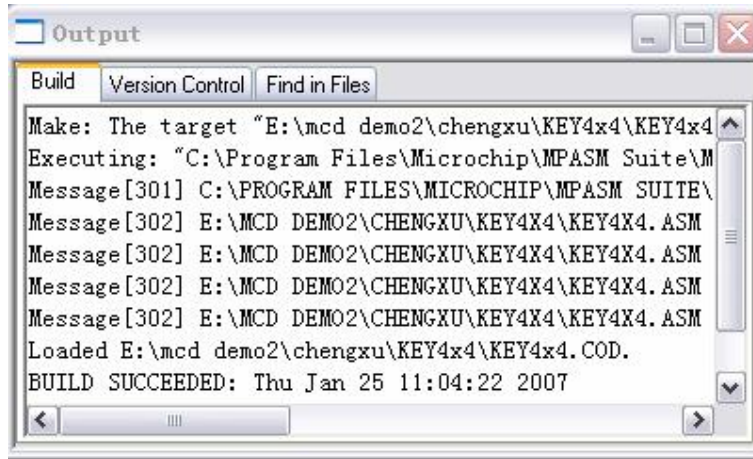

Figure 5-2 Source compiling results

It can be seen from the figure that there are several messages, suggesting that the registers used are not in the correct BANK, please carefully check the program to ensure that all the registers are in the right BANK (even if entirely correct, the information will still emerge, but it does not affect the implementation of results).

If there are ERROR[num] or WRNING[num] in the compiling results (num means the error or warning number), you may locate the cursor to the position that cause such errors or warnings by double clicking the message immerged, and correct the source code and re-compile it   repeatedly until it shows the result as in Figure 6-2.

Using MCD2 to debug the source

1) In accordance with the method shown in Figure 3-2, connect the MCD2 to PC and DEMO-II development board.

2) Implement the menu command Debugger à Select Tool to select MPLAB ICD 2 (same as MCD2) as a debugging tool.

3) Implement the menu command Debugger à Settings to select the USB as the communication port, and the board will use its own power supply (i.e. disable "MCD2 power from the target board". It is shown in Figure 5-3.
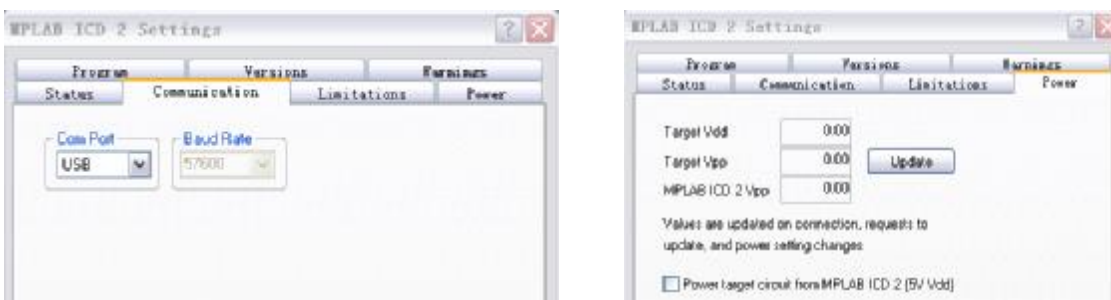

Figure 5-3 Set up communication ports and power supply for target board

4) Implement menu command Debugger à Connect or icon in the tool bar to connect MCD2 and DEMO-II development board, and when it is successfully connected it will shown as in Figure 5-4 with a message.

5) Implement menu command Debugger à Program or the icon in the tool bar to program/burn the objective codes to the target MCU chip (Note: now the MCU cannot run   on offline mode but has to be run under debugging mode), after programming it will show a message as in Figure 5-5.

6) Implement menu command Debugger à Run or toolbar icon, we can see the display of "FF" in the DEMO-II

development board, shown as Figure 5-6; Now if any key is pressed the LED will display a the number of that key, for example if K06 is pressed, the LED will display "06", shown as Figure 5-7.
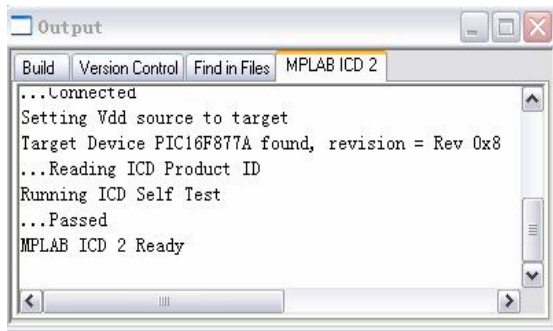

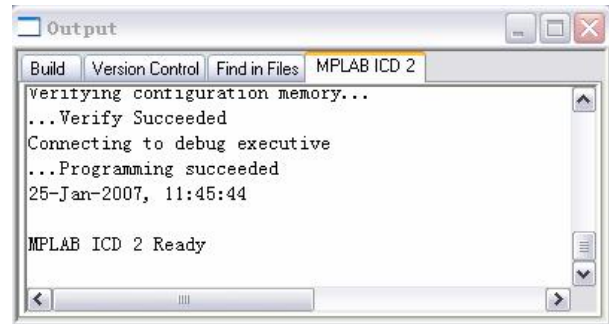**Figure 5-4** Connecting information   **Figure 5-5** Programming Information

7) After several debug, we have basically achieved the desired results, and the debug completed. Disconnect the connections.
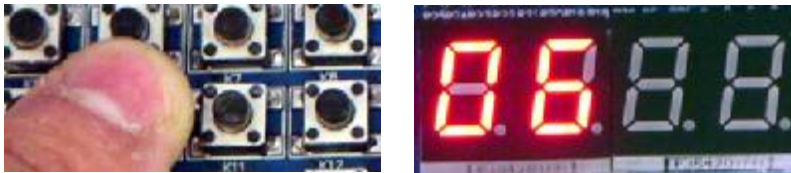

Figure 5-6 Result when no key is pressed

Figure 5-7 Result when K06 is pressed

Programming the target MCU using MCD2

The source codes prepared as described above has passed the debug by using MCD2, and we have achieved our desired goal, now we may use MCD2 to burn/program the KEY4x4.HEX file into the target MCU chip in order that the MCU can run offline.

1) Connect the MCD2 and the DEMO-II development board in accordance with the method shown in Figure 3-2.

2) Implement the menu command Programmerà Select Too to select MPLAB ICD2 (same as MCD2) as a debugging tool.

3) Set up the configured bits and other parameters according to the specific conditions.

4) Implement the menu command Programmer à Program, to program/burn the KEY4x4.HEX file into the target MCU chip.

5) Disconnect DEMO-II and MCD2 to see the results of MCU execution.

Appendix 1: Packing List and Contact

When you receive the products, please check the box to see if all accessories are complete. This product should include the following components:

- DEMO-II development board, 1 pc;
- PIC16F877A MCU, 1 pc;
- 1602LCD, 1 pc;
- Power supply, 1 pc;
- Serial cable, 1 pc and telephone cable, 1 pc;
- Manual and CD-ROM, 1 copy respectively;

Qianlongsheng Electronics Technology (Shenzhen) Co., Ltd.

WEB: http://www.pic16.com http://www.pic16.cn

Contact: 13-G, Huining Building, Minzhiqianlong Garden, Longhua Town, Shenzhen, Guangdong,P. R. China

Zip Code: 518131

Tel: 0755-28187975

Fax: 0755-28187976