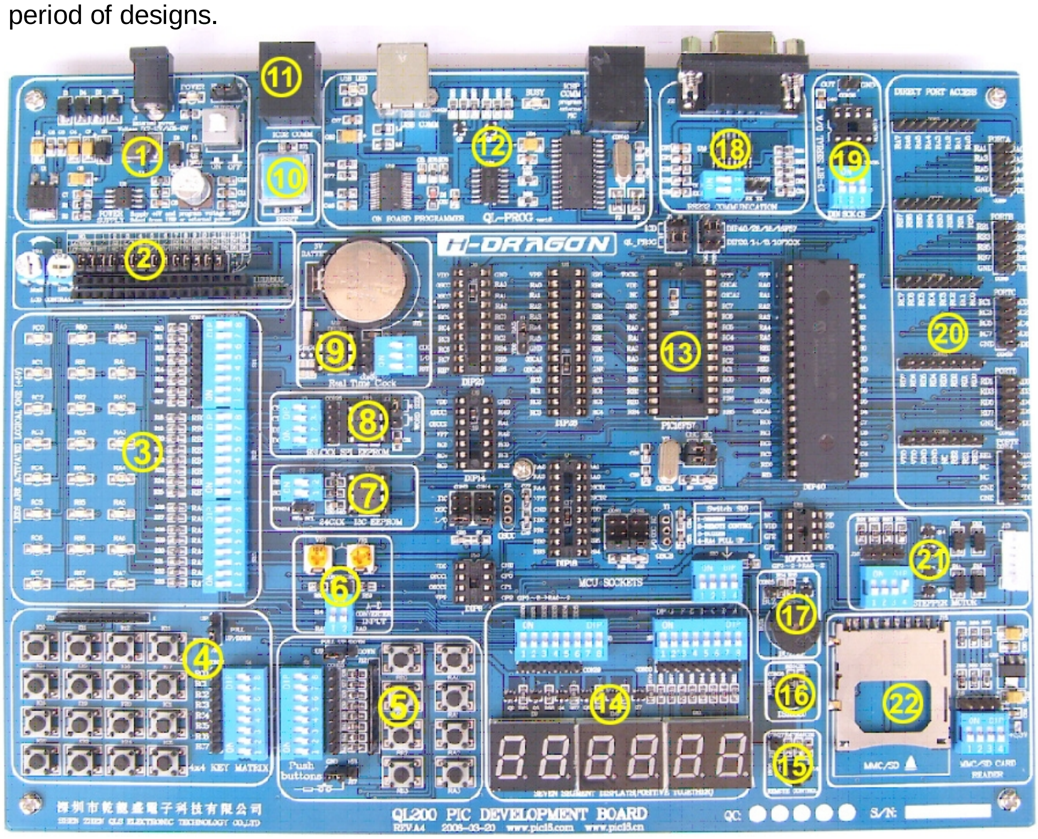
Chapter 1 Overview of QL200 DEVELOPMENT BOARD

QL200 PIC DEVELOPMENT BOARD (hereinafter referred to QL200) is a multifunctional PIC microcontroller development platform designed and developed by Shenzhen QSL Electronics Co., Ltd. based on our experience of many years' development and our efforts of months' design. It integrated the common external resources and simulation interfaces. Particularly it is suitable for self learning for microcontroller beginners as well as electronic lovers. It has rich and flexible hardware resources and the expansion feature as well as the free resource allocation, which makes it a convenient debugging tool for PIC engineers to use especially in their beginning period of designs.



1) Power Module
2) 12864LCD and 1602 LCD module
3) LED I/O module
4) 4x4 matrix keyboard module
5) Button module
6) A/D converter module
7) IIC communication module
8) SPI communication module
9) DS1302 module
10) MCU reset module
11) ICD / MCD online debugging interface
12) On-board programming module (USB Communication)
13) Chip socket and system clock selection
14) Digital LED display with 6 bit and 7 sections
15) Remote control receiver and decoder module
16) DS18B20 digital temperature measurement module
17) Beeper experimental module
18) USART serial communication module
19) 10-bit D/A converter module
20) Port output module
21) Stepper Motor Module
22) SD/MMC card Read/Write Module

Note: On-board programming module please see Chapter 4, and details of other modules please refer to Chapter 5.

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

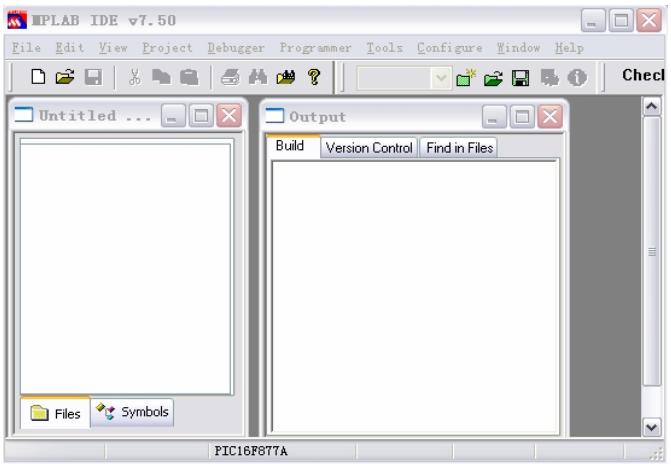
After downloading the files, you only need to use compression/decompression software tools such as WINZIP to depress and release the files in your computer, and then run SETUP.EXE (or Install.exe) program, and follow a step-by-step installation guide (You may also do not need changes any of the settings,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



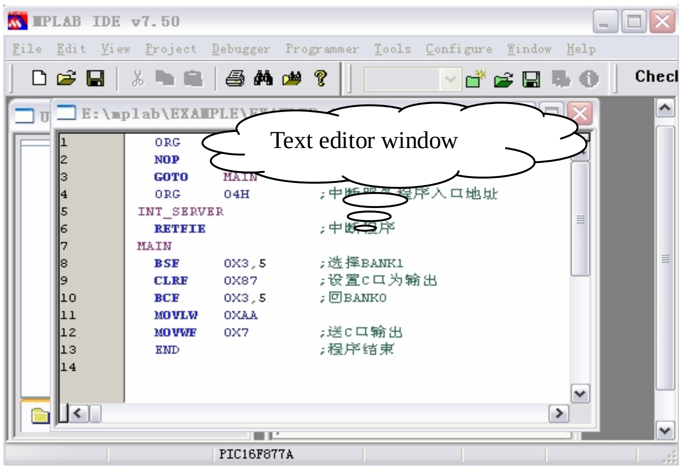


Figure 2-1 MPLAB main window **Figure 2-2** text editor window

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 <u>Project à Project Wizard</u> 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.

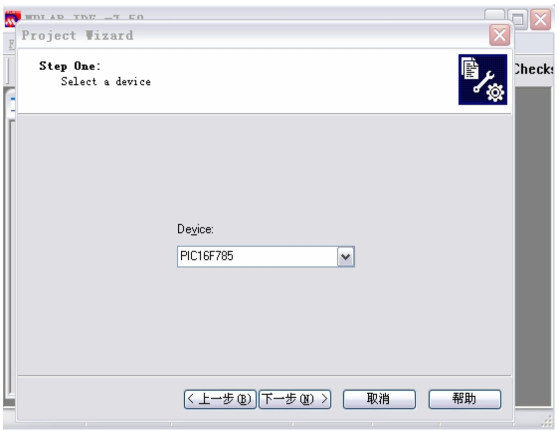


Figure 2-5 Choose chip model

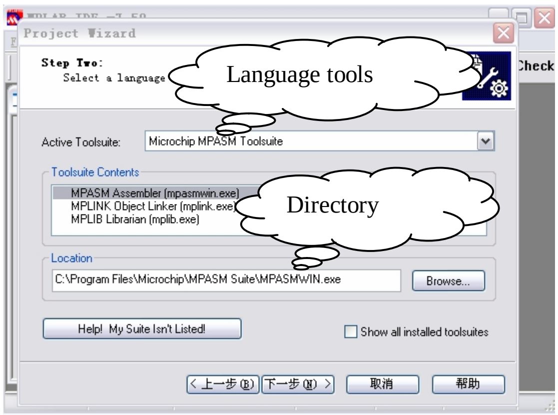


Figure 2-6 Choose compiler tools

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.

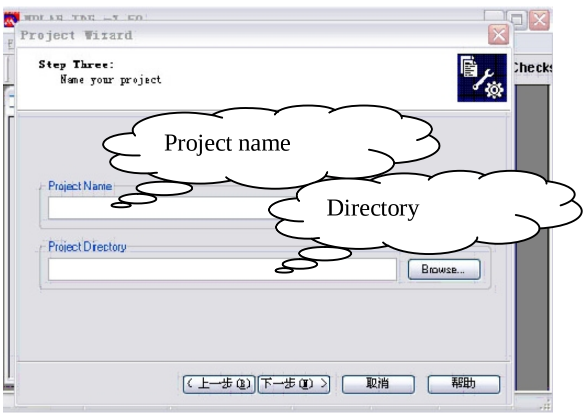


Figure 2-7 Choose directory to save project

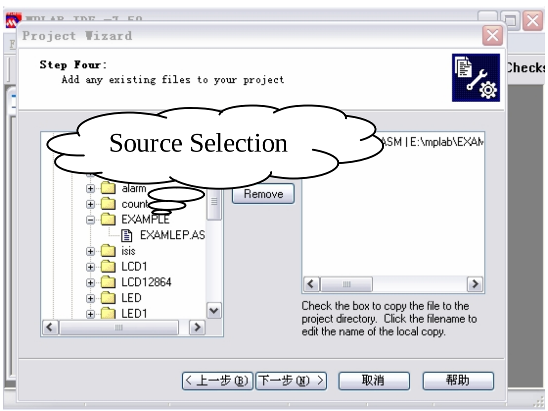


Figure 2-8 source selection

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

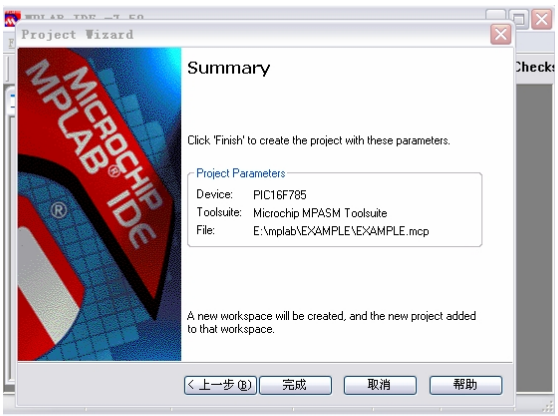


Figure 2-9 Tip interface

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

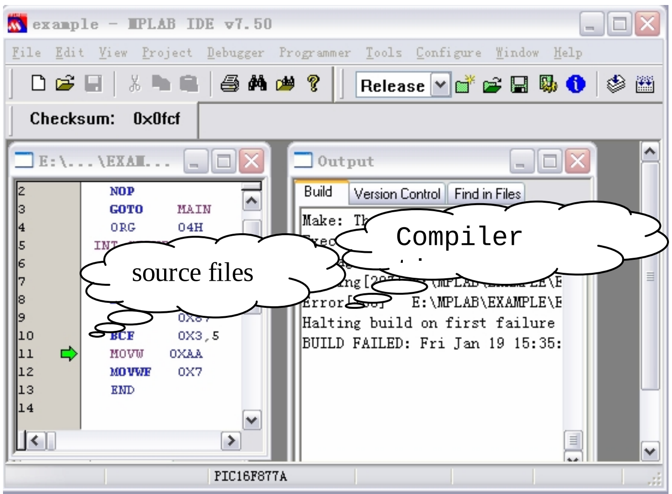
So far, we have completed establishing a project the source. For more details, please refer to MPLAB Operation Manual.

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.



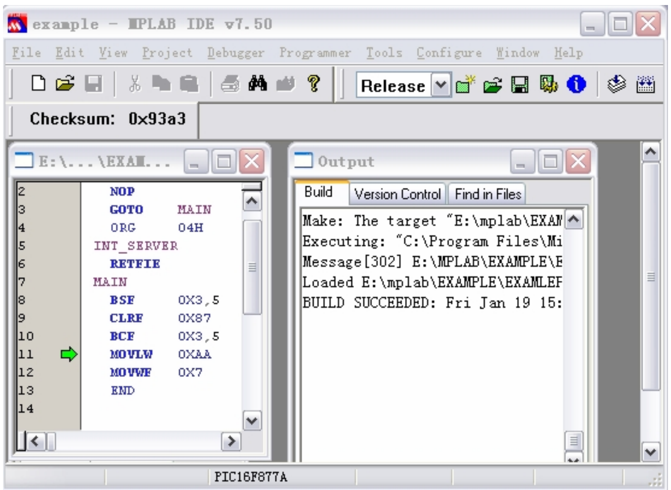


Figure 2-10 Source compiled results **Figure 2-11** Compiling succeeds

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.

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.

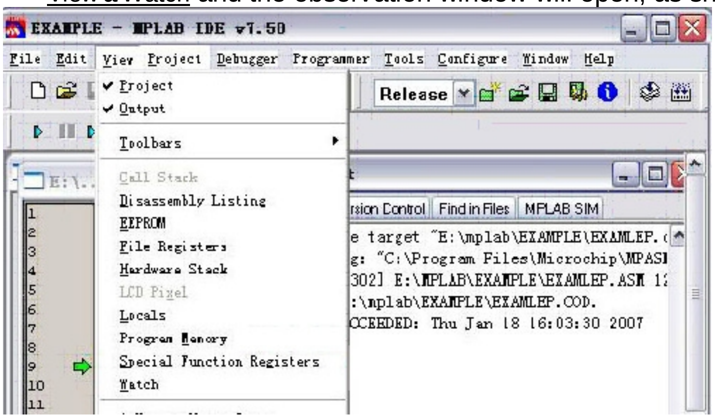
Choose debugging tools

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.



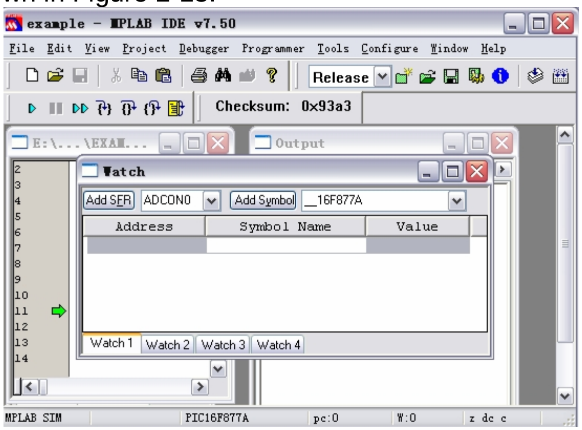


Figure 2-12 Menu command to open storage area **Figure 2-13** Observation window

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 can't work normal until it connects with PC and users target board.

MCD2 connection with the QL200

MCD2 connection with the QL200 shown in Figure 3-2 (we let MCD2 plate QL200 provides for power users).

Figure 3-2 MCD2 connection with the QL200

Step 1: USB or serial cable connect to PC to MCD2
Step 2: MCD2 external 9 V power supply
Step 3: use ICSP cables connecting MCD2 and QL200.

After the connection between MCD2 and QL200, set the jumpers of J5,J6,J7,J8 according to the chips.As shown in Figure 3-3.

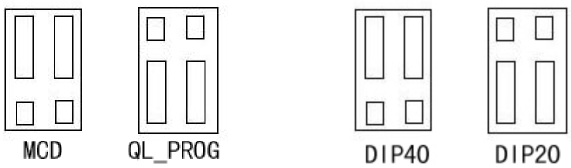


Figure 3-3 set jumpers

MCD2 and PC connectivity

Connect through RS232 between MCD2 and PC

Because serial communication rate is relatively slow, it is generally not recommended to use this method, but not in the computer USB port or USB ports can not use the circumstances, this is the only choice.

if use of serial connections, Complete the following steps to change the Flow Control and FIFO settings for a serial communications port on a PC running the Windows operating system.

Windows 98/ME:

On your PC, select Start>Settings>Control Panel.

In the Control Panel, double-click the System Icon.

In the Systems Properties dialog, click the Device Manager tab.

If necessary, expand the Ports selection by clicking the "+" sign next to it.

Double-click the I/O port to which the MPLAB ICD 2 is connected.

In the Flow Control field, select Hardware.

Click the Advanced button, deselect the "Use FIFO" box, and click OK.

Reboot the PC to implement the change.

Windows NT:

Note: You may need administrator privileges on your computer to change these settings.

On your PC, select Start>Settings>Control Panel.

In the Control Panel, double-click the Ports Icon.

Double-click the I/O port to which the MPLAB ICD 2 is connected.

Select the "Port Settings" tab.

In the Flow Control field, select Hardware.

Click the Advanced button, deselect the "Use FIFO" box, and click OK.

Reboot the PC to implement the change.

Windows 2000/XP

Note: You may need administrator privileges on your computer to change these settings.

On your PC, select Start>Settings>Control Panel.

In the Control Panel, double-click the System Icon.

In the Systems Properties dialog, click the Hardware tab and click the Device Manager button.

Double-click "Ports (COM & LPT)" to expand the Ports selection.

Double-click the I/O port to which the MPLAB ICD 2 is connected.

Select the "Port Settings" tab.

In the Flow Control field, select Hardware.

Click the Advanced button, deselect the "Use FIFO" box, and click OK.

Reboot the PC to implement the change.

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 QL200 Connection" to set up MCD2 and QL200 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 QL200 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 QL200.

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 ICD2 Manual

Chapter 4 QL200 programming module QL-PROG

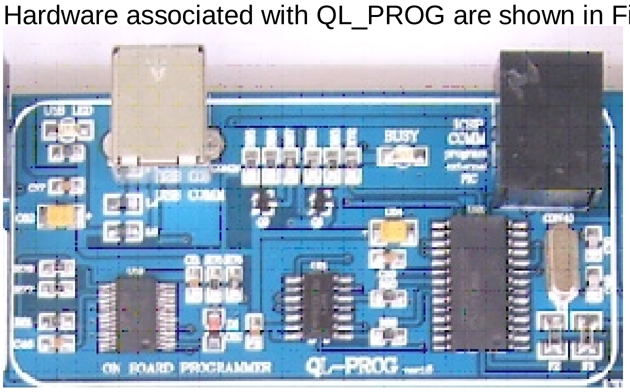
QL200 programming module QL_PROG (hereinafter referred to QL_PROG) is an important component of QL200. The user can easily download the objective program to the target MCU through simple setup so as to carry out various experiments conveniently.

QL_PROG has the following features:

1) Hardware connections are simple while software is easy to use.
2) Applicable for programming of most microcontrollers of PIC 10/12/ 16/18 series except 16C5X.
3) Hardware and software can be upgraded, which is quite simple.
4) Comprehensive information hints to keep you understand the working status clearly.
5) Built-in ICSP interface enables you to directly used ICSP interface for online programming on other board.
6) When the software was activated, it automatically searches hardware, and you may need to do no settings before you use the software.
7) Programming success rate is high with sufficient programming depth.
8) During programming, you can protect and change the "clock calibration value."

4.1 Hardware Introduction of QL200 Programming Module

Hardware associated with QL_PROG are shown in Figure 4-1 and Figure 4-2.



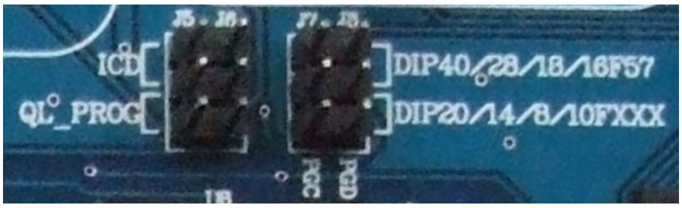


Figure 4-1 QL200 programming module **Figure 4-2** Jumper wire selection

The QL-PROG and PC are connected through "USB COMM" interface for communication.

The indicator BUSY keeps on during the whole process of programming, and after the completion of programming it automatically become off.

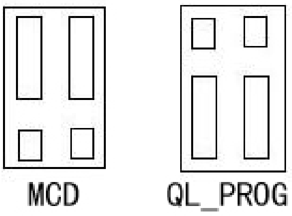
Through ICSP interface, users can use QL-PROG to program other target microcontrollers in other boards.

The programming module gets power supply from USB.

During online programming the ICSP may not use external power, but the jumper wires of power module should be set to the USB position.

Through jumper wires J5 and J6 to choose whether use QL_PROG to program target MCU or use MCD2 to debug MCU (with modules 11) as shown in Figure 4-3.

Through jumper wires J7 and J8 to choose programming/debugging DIP40/28/18/16F57 chips or DIP20/14/8/10FXXX chips (because the PGD and PGC of DIP40 are RB7 and RB6 ports, while the PGD and PGC of DIP20 are RA0 and RA1 ports). As shown in Figure 4-4.



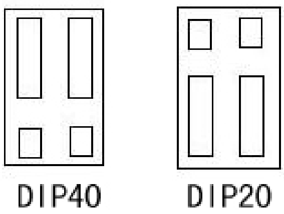


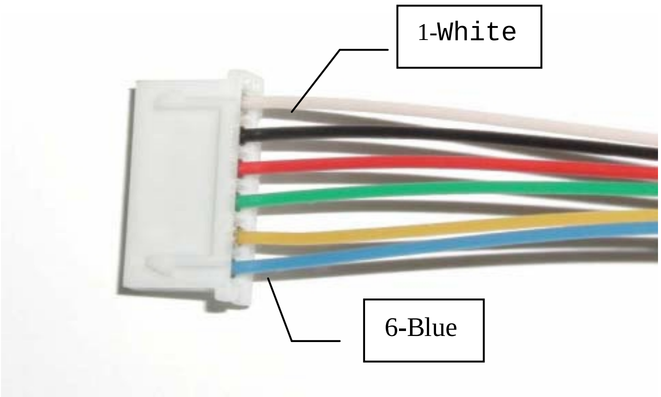
Figure 4-3 Choose MCD2 or QL-PROG **Figure 4-4** Choose DIP40 or DIP20

Programming other target boards by using ICSP online programming interface

At the right side of the front of programming module, there is a 6 PIN telephone socket, which is online programming interface. We can use the downloading cable which was provided along with this product for direct chip on-board (online) programming.
During online programming the ICSP may not use external power, but the jumper wires of power module should be set to the USB position.

Sequence of ICSP download cable is exactly same with MCD2/ICD2 as shown below:

1—MCLR/VPP
2—VDD
3—GND
4—PGD
5—PGC
6—LVP-ENABLE



When using it, we only need to connect the output cable of ICSP with corresponding pin of chip you desire to download. For 12XXX series chip, the connection shall be subject to PGC-GP1 and PGD-GP0.
In case of in-circuit download, relative external circuits connecting to ICSP output cable on user board shall be shut down. Otherwise, the effects of other external circuits may cause failure of programming. Furthermore, if current consumption of 5V voltage on user board is more than 100MA, the user board must be self-supplied with power of 5V voltage. In addition, VPP pin in download circuit shall be connected with 10K resistor pulling up to VCC.
In technical manual of MICROCHIP, there are typical circuits about in-circuit programming for references.

4.2 Software Introduction of QL200 Programming Module

QL200 programming software QL_PROG is what our company specifically developed for programming module QL_PROG of QL200.
The software can be procured from the CD-ROM coming along with our product, or you may also download the latest version from our website.

1. Installation of Software

Find file QL-PROGvXX.EXE under "soft" of CD, double click this file to access installation interface of application software, and operate according to prompts to finish the installation.

2. Installation of USB Driver

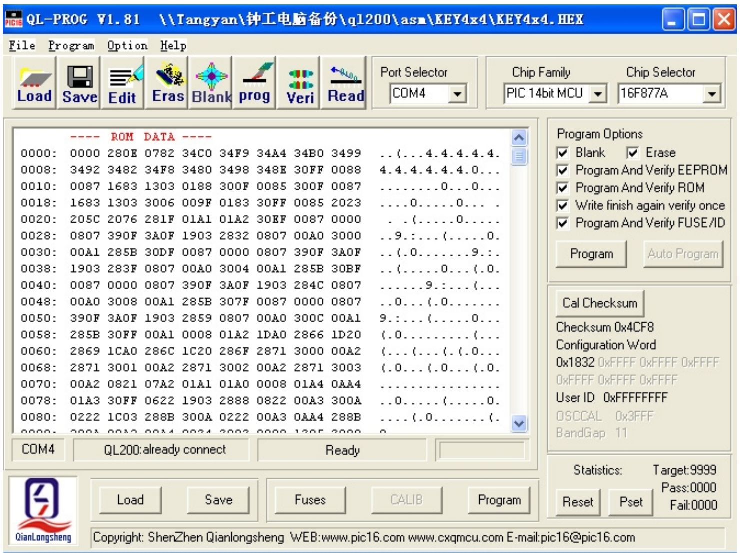
When you first use the USB of this programmer for communication, please connect PC and programmer hardware through USB cable. Then, PC system will indicate to find new hardware and request installation of driver for this hardware.
You can operate in regular way and designate path to install driver, which is under directory “QL200 usbdriver” of attached CD.
After installation, a serial port will be added to your computer. Now the programmer is connected to the serial port converted from USB.
When there is no system message to find new hardware after programmer hardware is connected, please check:

- Any bad or broken connection for USB cable?
- Does USB of computer works smoothly?
- Maybe there is similar USB driver already installed in your computer. You may connect programmer hardware, right click “My Computer”, select “Property” in pop-up list, switch to “Hardware” from “System Property” window, and then click “Device Manager” to check “Port (COM and LPT)”. If you can find “USB

Serial Port (COMX)", the driver of this programmer has been installed in your computer (this operation requests proper connection of hardware and PC).

3. Running of Software

After proper installation, shortcut of this software will appear on desktop. We can double click it to run the software. (Note: Run the software after proper connection of hardware so as to identify the hardware easily.)



4. Use of Software

1） Select Communication Port/Select Programmer Hardware

Connect hardware and power up, run the software which will automatically search programmer hardware. If search fails and indicates "Have not find board…", you can set programmer hardware manually. We offer two methods for user to select communication port and programmer type.

Method 1:Directly run【Auto Find Borad】under【Option】. Software will scan the port connecting hardware automatically and check the type of programmer. After scanning, software will switch to correct connecting port and programmer type and indicate to find the programmer. If software fails to find programmer, please check if the connection to programmer hardware and power supply.

Method 2:Directly select the port from the list of "Port Selector" at top right of the software. (Note: Only existing ports of computer are shown in the list. If only COM1 is installed at your computer, there is only COM1 for your choice. If you have connected USB but can't find it in the list, the reason may be that the software is started too fast. You can shut down the software and restart it.)

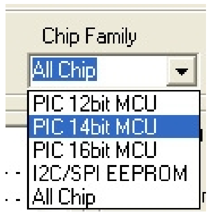
2) Select Chip Family and Chip model

A. Select Chip Type

The control for selection of chip type is at top right of the software as shown in right diagram:

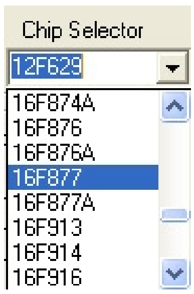
You may select the type to shorten the chip list for fast selection of proper chip.

If you don't know exactly the device, please select "All Chip" in the list of "Chip Family"



B. Select Chip

The control for selection of exact chip is at top right of the software as shown in right diagram:



If the desired chip type is not listed, please change the chip Family or select "All Chips" and try again.

3. Load File to Program

Run【File】-【Load File】, or press "Load" button on programming software panel to load machine code file of the chip you desire to program.

This programmer software supports load of BIN files and HEX files.

4. Edit Buffer

You may edit current buffer by running【Option】-【Edit】or pressing "Edit" button on software panel. After running the command software, pop-up window to edit buffer will appear as shown in below diagram:

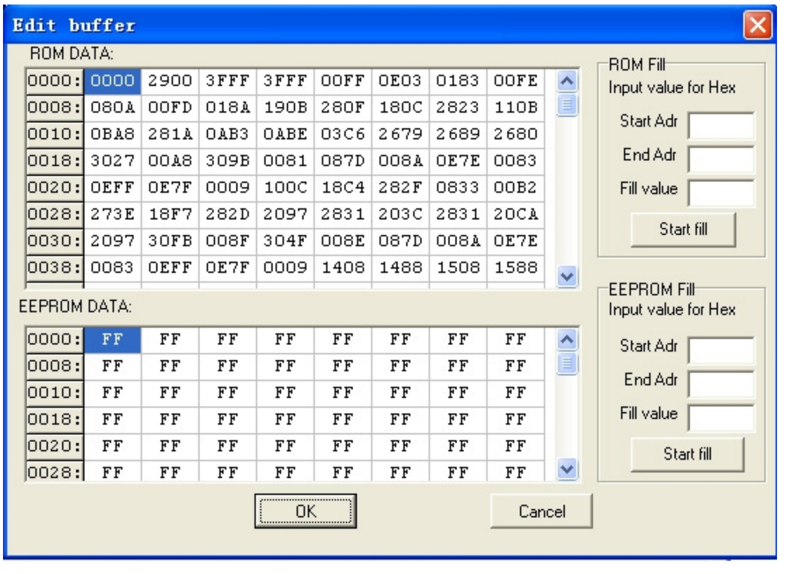


Figure 4-8 Edit Buffer

Change the content of buffer: first put mouse cursor on the address to edit, double left click to select address content, delete original content with "Delete" key on keyboard, and then input the new content.

5. Set Configuration Bit

You may set or change the configuration bit of chip by running【Option】-【View/edit Fuses And ID】or pressing "Fuses" button on software panel. After running the command software, pop-up window to set configuration bit will appear as shown in below diagram. Set the options according to your requirement and click "OK" button.

User's ID can also be input in set window of configuration bit.

If configuration bit information has been written in your program, direct programming will be available without reset of configuration bit.

The following are listed for your reference:

Setup of external crystal oscillator: select LP for below 200K; select XT for 200K-4m; select HS for 2M-20M. For 4M crystal oscillator at critical state of XT and HS, if chip doesn't work when you select XT, please try HS.

For the beginner who doesn't know how to set configuration bit, to shut down or disable the functions which are not required if possible (WDT,PWRTE,BODEN,Debug,LVP... all these functions can OFF or disabled).

If the chip is programmed for production, please carry out the setup correctly according to the requirements of your hardware and softwar

> Note: Setup of configuration is very important, which directly affects the availability of your program. Consequently, before programming, we must confirm the setup of configuration bit is entirely correct. Different chips require different configuration bit options. The detailed setup information can be obtained by checking the DATASHEET of chip (DATASHEET of chip can be downloaded from official website of MICROCHIP at www.microchip.com).

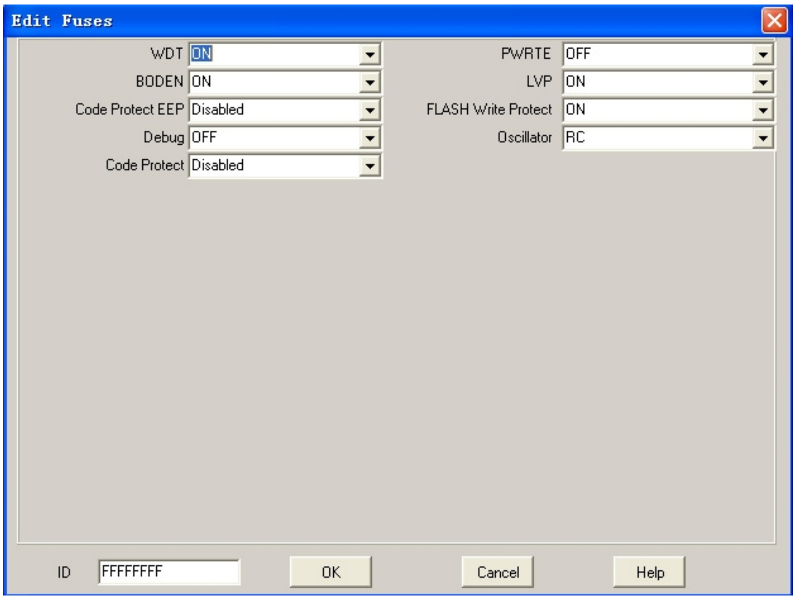


Figure 4-9 Chip's Fuses

6. Setup and Protect of Parameter Calibration of Internal Clock

For most of the chips with internal clocks, there is calibration value of internal clock stored in the last address of ROM when leaving the factory. This value can be used to calibrate the accuracy of internal clock frequency in the program when internal oscillation is applied for the chip.

There are two options in our software under【Option】-【Cal Program Options】, that is 【Preserve OSCCAL Calibration Value】and【Use File】.

If we select【Preserve OSCCAL Calibration Value】, our software will always protect this value in chip unchanged regardless of value in the last address of current buffer and regardless whether you run "CALIB" on software panel to change this value or not. In this case, "CALIB" button is invalid now.

If we select【Use File】, the value in current buffer will be used to change the original calibration value of clock when programming. Now, we may press "CALIB" button on software panel to input the calibration value of clock we desire.

For new chip or the chip whose clock calibration value is never changed, we recommend【Preserve OSCCAL Calibration Value】to protect this value of chip unchanged.

For the chip whose clock calibration value has been changed, we may select【Use File】and input the value we desire by pressing "CALIB" button on software panel (this value may be read from other chips with the same lot number). Then, we can start the programming. When all chips have their own clock calibration values, we had better reset【Cal Program Options】to【Preserve OSCCAL Calibration Value】.

How to use the internal RC oscillator of PIC as the accurate system clock?

Starting from PIC12C508, gradually, a lot of models of PIC have their internal RC oscillator which can be used as the system clock. With this feature, the cost of the overall system becomes further lower, and because the internal RC oscillation has been calibrated, it is more accurate clock than the external RC oscillators.

However, if you want to use internal RC oscillation and at the same time you want to get an accurate clock, there is still a good method. Besides that you need to use the internal RC oscillator when you are programming IC, you also need to add some commands in the original codes. And the uses of commands are different according to different types of PIC.

As for the PIC with 12 - BIT structure, such as PIC12C508, Microchip has added a command "movlw 0x .." to the last line of the IC program memory, and when the chips is reset, IC will first execute this command and then skip to Address 0 to continue running. Therefore, if you have not modified the properties of the W register, then w register will always keep the original value "..". And the value ".." is the calibrated value what Microchip measured out. If you fill it to the OSCCAL register, then subsequently the RC oscillator will be "quite accurate" 4 MHz. So usually a simple approach is to fill a "movwf OSCCAL" command to the first line of the codes.

As for the PIC with 12 - BIT structure, such as PIC12F629, Microchip has added a "retlw 0x .." command at the last line of IC program memory. Therefore, at any location of program you can add "call LAST-ADDRESS" (for PIC12F629, LAST-ADDRESS = 0x3ff) and "movwf OSCCAL" these two commands to calibrate the internal RC oscillation.

7. Save File

You may run【File】-【Save】or press “Save” button on software panel to save the current file as hexadecimal system file. At the same time, configuration information of chip is saved. For the next time this file is loaded, configuration information of chip will be automatically set as the configuration word saved.

(1) Click icon on tool bar or select【File】-【Save】, dialog to save the file will pop up.
(2) Select the file to save in list of “Save Type (T)”. For example, to save file of hexadecimal system, please select “Hex File (*.Hex)”.
(3) Input file name to save in “File Name” box.
(4) Click “Save” button in dialog.

8. Complete Procedure of Chip Programming

Step 1: Connect QL200 and PC.
Step 2: Start QL_PROG programming software.
Step 3: If the tips show not successfully connected, then please manually select Programmer Model, and select hardware ports to enable the establishment of communications between hardware and software (you may also execute the menu command [Option] à [Auto Find Board] to automatically set programmer model and ports).
Step 4: Select Chip Family and Chip Model.
Step 5: Place and insert the target chip into the chip socket. And set a jumper wires J5 and J6, J7, J8.
Step 6: Load the target file.
Step 7: Confirm and set configuration bits (fuses) of the chip.
Step 8: If the chip has internal clock and in the program the clock calibration values is used, then please set [Cal Program Options] under menu [Option] or calibrate the clock (generally you may skip this step).
Step 9: Set Program Options as necessary.
Step 10: Press "Program" button to begin programming. After completion, there will be messages of "PASS" or "ERROR".

4.3 List of Programmable Chips that QL_PROG Supports

QL_PROG currently supports the devices listed in the table below, of and course, after the software is upgraded, it will be able to support more devices (about upgrading information you may refer to our website www.pic16.com)

10 series:						
PIC10F200	PIC10F202	PIC10F204	PIC10F206	PIC10F220	PIC10F222	
12C series:						
PIC12C508	PIC12C508A	PIC12C509	PIC12C509A	PIC12C671	PIC12C672	PIC12CE518
PIC12CE519	PIC12CE673	PIC12CE674				
12F series:						
PIC12F508	PIC12F509	PIC12F629	PIC12F635	PIC12F675	PIC12F683	PIC12F615
PIC12HV615						
16C series:						
PIC16C505	PIC16C554	PIC16C558	PIC16C61	PIC16C62	PIC16C62A	PIC16C62B
PIC16C63	PIC16C63A	PIC16C64	PIC16C64A	PIC16C65	PIC16C65A	PIC16C65B
PIC16C66	PIC16C66A	PIC16C67	PIC16C620	PIC16C620A	PIC16C621	PIC16C621A
PIC16C622	PIC16C622A	PIC16C71	PIC16C71A	PIC16C72	PIC16C72A	PIC16C73
PIC16C73A	PIC16C73B	PIC16C74	PIC16C74A	PIC16C74B	PIC16C76	PIC16C77
PIC16C710	PIC16C711	PIC16C712	PIC16C716	PIC16C745	PIC16C765	PIC16C773
PIC16C774	PIC16C83	PIC16C84				
16F series:						
PIC16F505	PIC16F506	PIC16F54	PIC16F57	PIC16F59	PIC16F627	PIC16F616
PIC16HV616	PIC16LF627A	PIC16F627A	PIC16F628	PIC16LF628A	PIC16F628A	PIC16F630
PIC16F636	PIC16F639	PIC16F648A	PIC16F676	PIC16F683	PIC16F684	PIC16F685
PIC16F687	PIC16F688	PIC16F689	PIC16F690	PIC16F716	PIC16F72	PIC16F73
PIC16F74	PIC16F76	PIC16F77	PIC16F737	PIC16F747	PIC16F767	PIC16F777
PIC16F785	PIC16F83	PIC16F84	PIC16F84A	PIC16F87	PIC16F88	PIC16F818
PIC16F819	PIC16F870	PIC16F871	PIC16F872	PIC16F873	PIC16F873A	PIC16F874
PIC16F874A	PIC16F876	PIC16F876A	PIC16F877	PIC16F877A	PIC16F913	PIC16F914
PIC16F916	PIC16F917					
18 series:						
PIC18F242	PIC18F248	PIC18F252	PIC18F258	PIC18F442	PIC18F448	PIC18F452
PIC18F458	PIC18F1220	PIC18F1320	PIC18F2220	PIC18F2221	PIC18F2320	PIC18F2321
PIC18F2331	PIC18F2410	PIC18F2420	PIC18F2431	PIC18F2450	PIC18F2455	PIC18F2480
PIC18F2510	PIC18F2515	PIC18F2520	PIC18F2525	PIC18F2550	PIC18F2580	PIC18F2585
PIC18F2610	PIC18F2620	PIC18F2680	PIC18F4220	PIC18F4221	PIC18F4320	PIC18F4321
PIC18F4331	PIC18F2680	PIC18F4220	PIC18F4221	PIC18F4320	PIC18F4321	PIC18F4331
PIC18F4510	PIC18F4515	PIC18F4520	PIC18F4525	PIC18F4550	PIC18F4580	PIC18F4410
PIC18F4420	PIC18F4431	PIC18F4450	PIC18F4455	PIC18F4480	PIC18F4585	PIC18F4610
PIC18F4620	PIC18F4680					

Note: After upgrade of the software, it can support more devices, and upgrading information, please pay close attention to our website.

Chapter 5 Details of the Functional Modules of QL200

This chapter describes all the functional modules of QL200, and the schematics and photos of these modules are illustrated in this chapter. Examples using these modules are collected in the CD-ROM that comes along with this product. As this experimental equipment is of modular design, the configurations may vary from case to case, besides, the output ports are all opened to the external uses. Since the hardware configuration and software arrangement are according to user’s will, user may draw inferences about other cases from one instance.

5.1 MCU Port Resources and System Clock Selection

This module is the core portion of QL200 development board, as shown in Figure 5-1.
This module consists of the following main components:

1) 40-pin chip socket
2) PIC16F57 chip socket
3) 28-pin chip socket
4) 20-pin chip socket
5) 18-pin chip socket
6) 14-pin chip socket
7) 8-pin chip socket
8) PIC10FXXX chip socket
9) Clock source option OSCA (this clock source is used for DIP40/28 and PIC16F57).
10) Clock source option OSCB (this clock source is used for DIP18).
11) Clock source option OSCC (this clock source is used for DIP20/14/8).

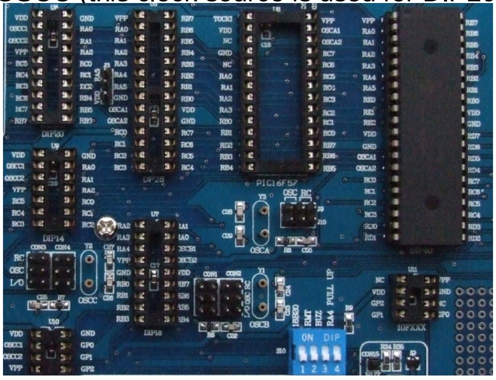


Figure 5-1 MCU socket and clock sources

Chip-pin Resources:

About the details of pin resources of each chip please refer to its respective manual, and here we just list some specific pins used in this development board which needs the user to pay attention to.

1) Pin 8 of PIC10FXXX is used for VPP and MCLR, and cannot be used as I/O GP.
2) Pin 4 of DIP8 is used for VPP and MCLR, and cannot be used as I/O GP3.
3) Pin 4 of DIP14 is used for VPP and MCLR, and cannot be used as I/O RA3.
4) Pin 4 of DIP18 is used for VPP and MCLR, and cannot be used as I/O RA5.
4) Pin 4 of DIP20 is used for VPP and MCLR, and cannot be used as I/O RA3.
6) TOCKI PIC16F57 must connect to the power supply.
7) Pin 7 of DIP28 may connect to RA5 or VDD through selection of jumpers (Pin 7 of 18FXX31 is VDD, and Pin 7 of other 28-pin PIC is RA5).
8) Pin 2 and 3 of DIP8/14/20 can not only be used as clock input and output, but they can also be used as I/O port (in the condition that the internal clock mode is chosen at the point of clock selection.)
9) GPIO port of 8-pin chip are connected to other chips.
10) Pin 15 and 16 of DIP20 section can not only be used as clock input, but they can also be used as I/O port (in the condition that the internal clock mode is chosen at the point of clock selection.)
11) There’s a 10K up-pull resistor on RA4 (which is controlled by the coding switch S10 next to the LED), for some chips the up-pull resistor must be activated so that the RA4 can out put high level voltage.
(In the experiment of 12864 LCD and 1602 LCD RA4, the up-pull resistor of RA4 must be activated, which is set S10_4 to be ON).

System Clock Selections:

1) PIC10FXXX clock selection. PIC10FXXX can only use the Internal RC oscillation.
2) DIP8/14/20 clock selection. As shown in Figure 5-2.

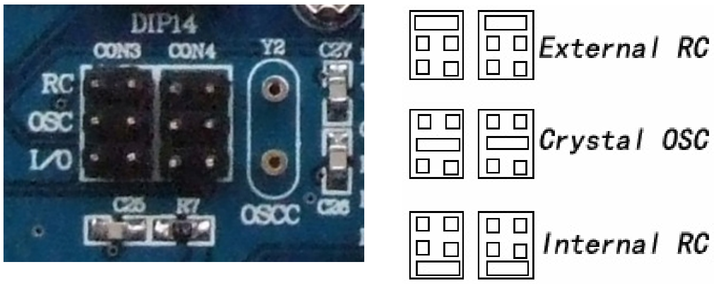


Figure 5-2 DIP8/14/20 Clock

3) DIP20 clock selection. As shown in Figure 5-3

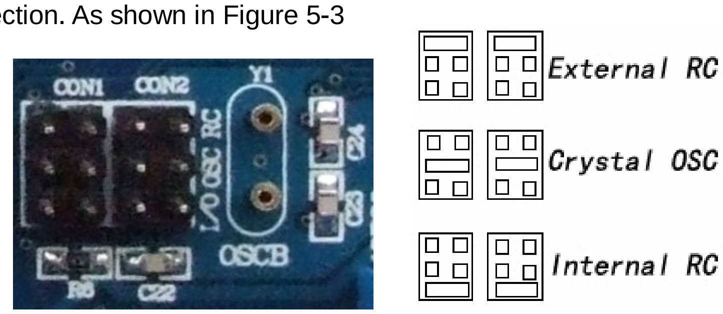


Figure 5-3 DIP20 clock selection

4) DIP40/28/PIC16F57 clock selection. As shown in Figure 5-4

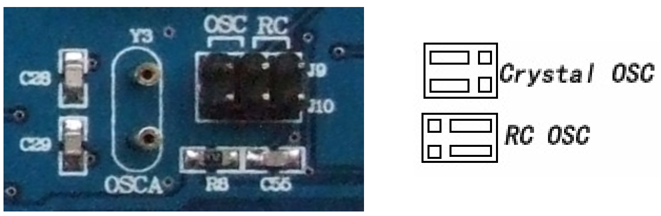


Figure 5-4 DIP40/28/PIC16F57 clock selection

5.2 Power Module

Power modules supplies power for the entire development board QL200, it provides 5V DC output for all functional modules to work (except that the QL_PROG get power from USB)and it also provides 13V DC high voltage output for programming. The module is shown in Figure 5-5.
The module consists of the following major components:
1) External power input
2) Power indicator
3) Power selection jumper wire (external power supply or USB)
4) Power switch

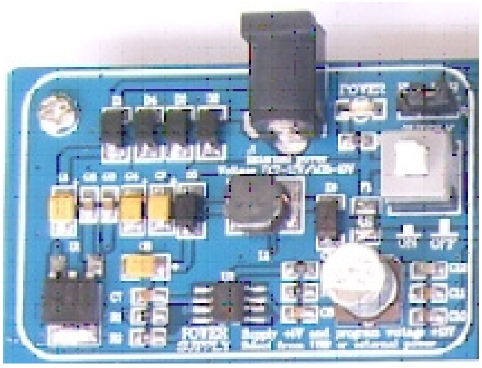


Figure 5-5 Power module

Descriptions of the power module are as follows:
1) Use bridge rectifier for external power input, and you do not have to consider the positive and negative power supply, as long as the voltage range is 7 V ~ 9 V and the current is over 200 MA.
2) You may choose to use external power supply or the USB power supply for the system.
3) Self-locking switch controls the power of the entire development board, which is used to avoid unplugging the power or USB connection. (the programming module get power supply directly from the USB without switching control, but the power of the target board is under the control of this power module when you are programming the target board, so please switch on this power module when you are programming the target board.)
12) When the power supply is connected, the power indicator will be on, which can be one of the evidence whether the system is powered on or not.

Power module schematic shown in Figure 5-6.

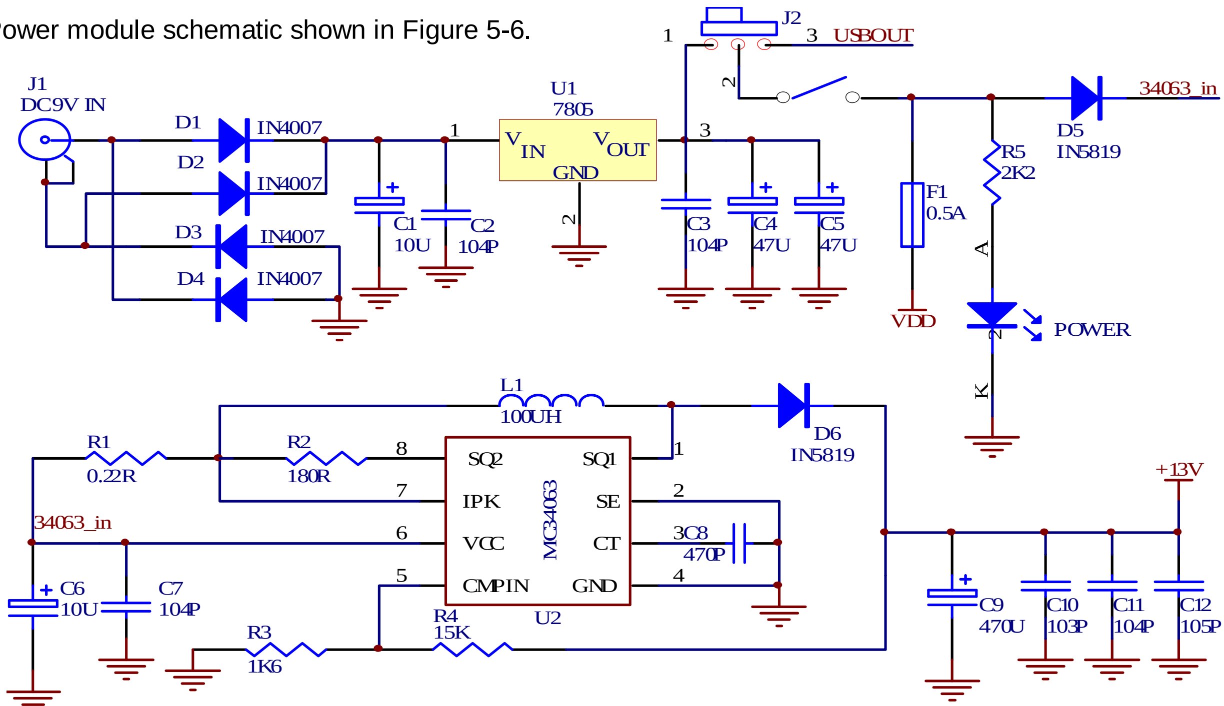


Figure 5-6 Power modules schematic

5.3 12864LCD and 1602LCD Modules

12864LCD and 1602LCD modules are shown in Figure 5-7. The modules consist of the following components:
1) Character LCD1602 and contrast adjustment potentiometers.
2) Graphics array LCD12864, and contrast adjustment potentiometers.

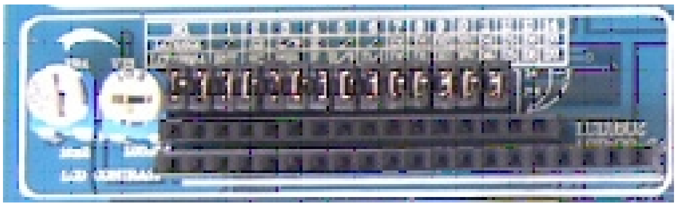


Figure 5-7 12864LCD and 1602LCD Modules

Descriptions of these modules are as the follows:
1) 12864LCD and 1602LCD both use the PORTA as the control bit and the PORTD as data bit.
2) All the pins of this module are jump connected to the pins of MCU without coding switches to control it. It is recommended that the LCD is plugged off the socket.
3) In the experiment of LCD1602 or LCD12864, the up-pull resistor of RA4 required to be activate, i.e., set the S10_4 ON.
4) Datasheets of 12864LCD and 1602LCD are included in the Datasheet directory of CD-ROM that comes along with product.
5) Examples of 12864LCD and 1602LCD are included in the Example directory of CD-ROM that comes along with product.

Schematics of these modules are shown in Figure 5-8.

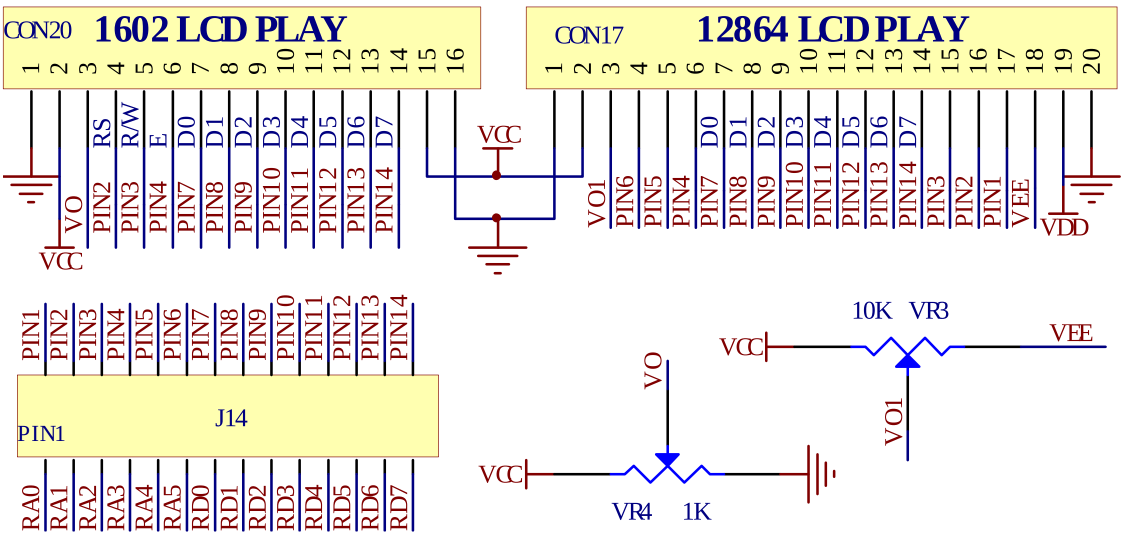


Figure 5-8 Schematics of 12864LCD and 1602LCD

5.4 PORTA, I PORTB and the LED module PORTC

The module shown in Figure 5-9 consists of the following components:

1) 8 LED lights on PORTA, and PORTA of PIC16F628 etc has 8 bits.
2) 8 LED lights on PORTB.
3) 8 LED lights on PORTC.

Descriptions about this module are as the follows:

1) All the three ports have respective 8 bits output
2) All the three ports use separate coding switch to control them. It is recommended that if this module is not in use, the corresponding bits should be shut off to avoid interference with other functions.
3) Through the LED socket of PORTC, you can conduct the experiment of other ports (such as PORTD and PORTE of 40-bit MCU). However, when this is in use, please make sure that the coding switch must be switched off.
4) All LED are grounding with negative pole, and if you want to lit the LED, just output a high level voltage at the corresponding pin of the port.
5)In the CD-ROM examples of "single LED light" and "play-in-turn" are included for reference.

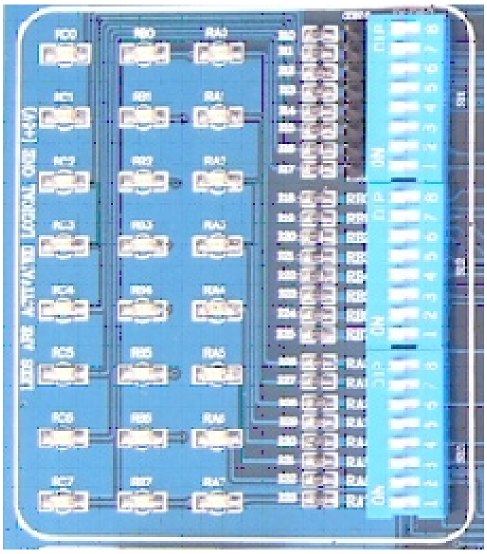


Figure 5-9 LED Module of Ports A/B/C

The schematic of this module is shown in Figure 5-10 below:

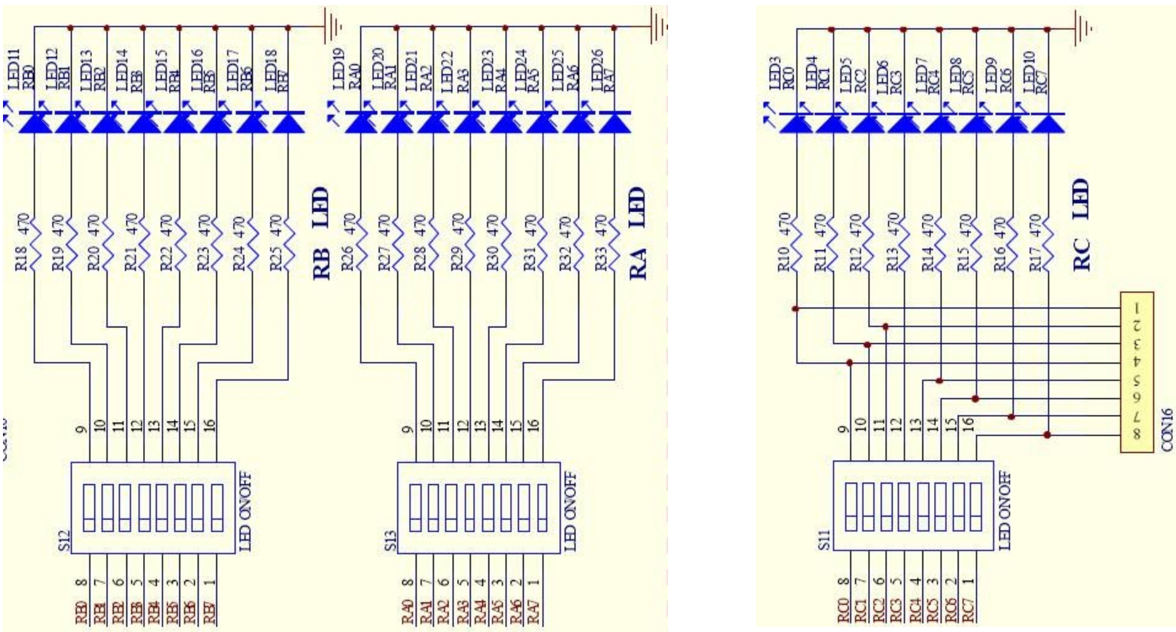


Figure 5-10 Ports A/B/C LED schematic

5.5 4x4 matrix keyboard

This module is shown in Figure 5-11. It consists of the following main components:

1) 16 keys.
2) Coding switching.
2) Interface socket.
2) Up-pull resistors.

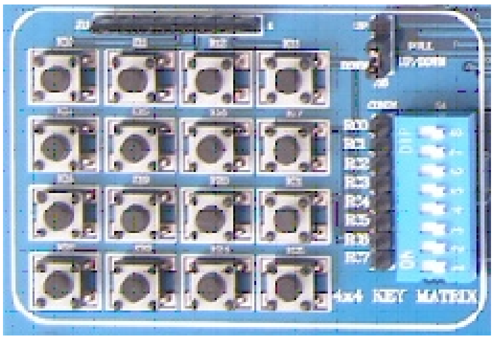


Figure 5-11 4x4 matrix keyboard

Descriptions about this module are as the following:

eys are connected to the 8 bits of PORTC in 4 x4 array

2) This module uses coding switch to control it. It is recommended that if this module is not in use, the corresponding bits should be shut off to avoid interference with other modules.

3) Through the interface socket you may practice the other matrix keyboards (you must make sure the coding switch is off).

4) There's a 10K up-pull or down-pull resistors to ensure the stability of voltage level of data online.

5) Examples about "4 x4 keyboard matrix" are included in the CD-ROM.

The schematic of this module is shown in Figure 5-12.

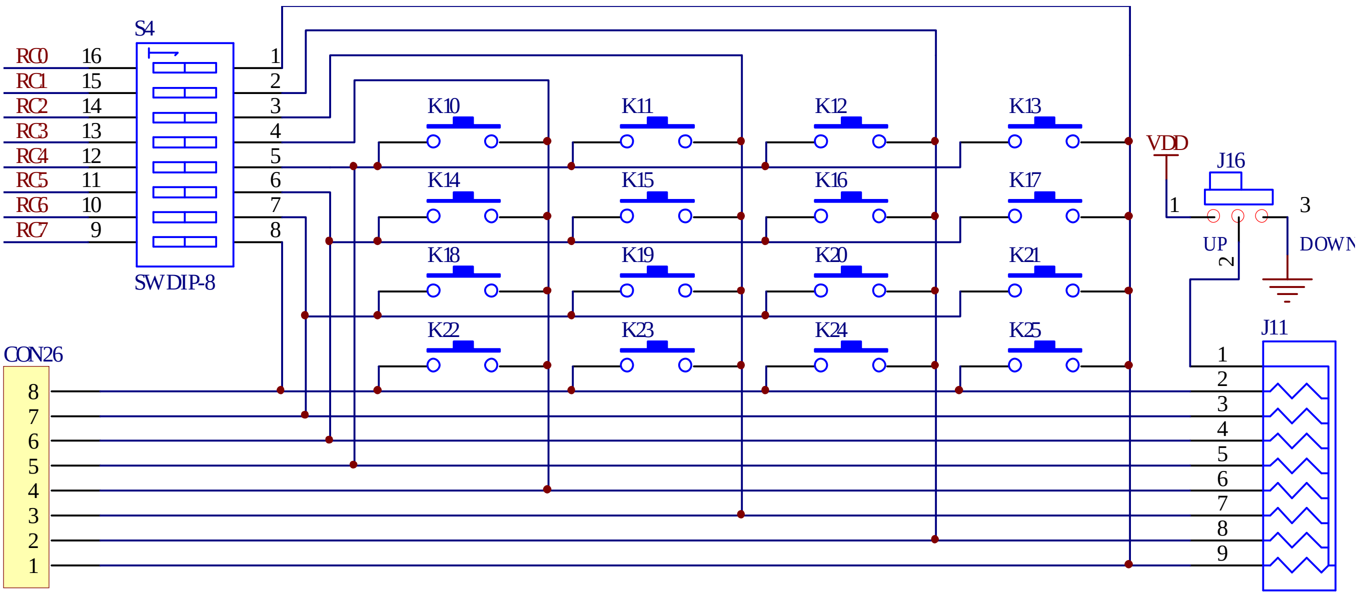


Figure 5-12 4x4 matrix keyboard schematic

5.6 PORTA0~3 and PORTB0~3 Key Module

This module is about single key functions, as shown in Figure 5-13.

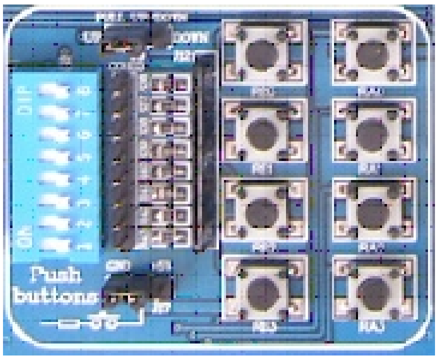


Figure 5-13 Keys

This module consists of the following components:

1) 4 keys on PORTA0 ~ 3;
2) 4 keys on PORTB0 ~ 3;
3) 10K up-pull or down-pull resistors on 8 pins, and jumper select.
4) Coding switch
5) Interface socket

Descriptions of this module are as the following:

1) 8 keys are connected to the PORTA0 ~ 3 and PORTB0 ~ 3.
2) When the keys are pressed, the output level of voltage is LOW or HIGH(jumper J17 select)
3) The 8 keys are controlled by the coding switches to decide whether they are connected to the pins of MCU, and when this module is not in use please make sure the coding switches are not connected to avoid the affect on other modules.
4) Through the interface socket, you can do other output experiment (with the coding switches disconnected).
5) In case the keys on PORTB0 ~ 3 are used, if the internal up-pull resistors are activated, you may use the jumper wire to switch them off to disconnect them from the internal up-pull resistors. But for the keys on PORTA0~3, in order to keep the stable voltage level, the jumper wires must always be ON.
6) Examples about use of the key module are contained in the CD-ROM.

The schematic of this module is shown in Figure 5-14.

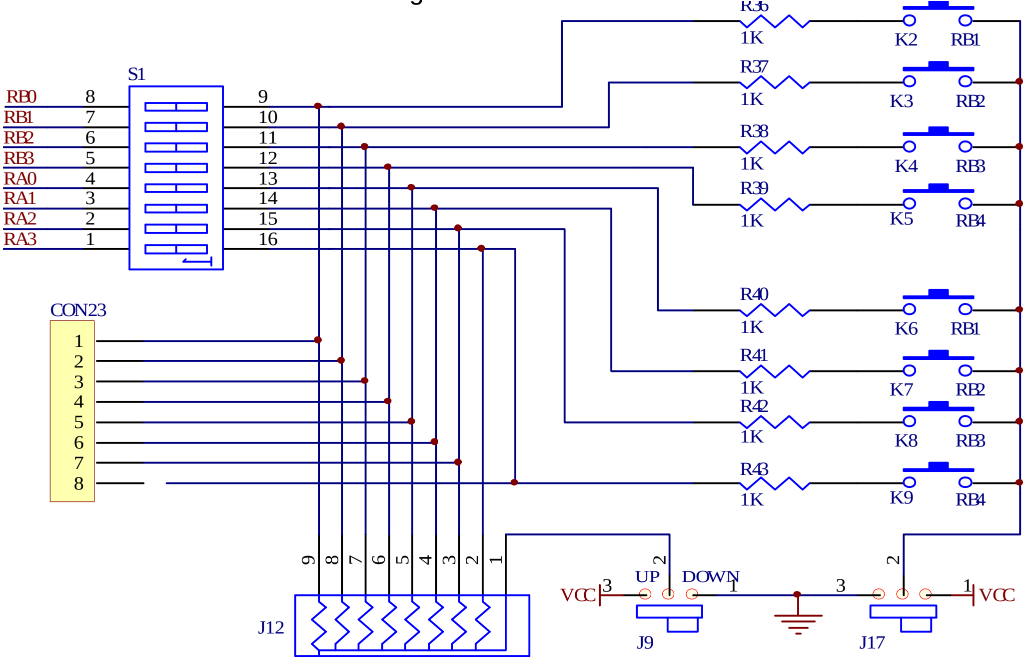


Figure 5-14 Schematic of the keys

5.7 A/D Converter Module

This module main executes the conversion from analog signal to digital signal, as shown in Figure 5-15.

Figure 5-15,

This module has the following components:

1) Two 1K potentiometers
2) One coding switch with 2 bits
3) Interface socket

Descriptions of this module are as the following:

1) 2 potentiometers are connected to RA0 and RA1 ports respectively.
2) When the A/D converting function of RA0 and RA1 is in use, please make sure that the coding switch is on, and while it is not in use, please make sure the coding switch is off in order to avoid the normal function of RA0 and RA1 for other modules.

3) By using the interface socket you can experiment with A/D converter functions of other MCU (you must ensure that the coding switch is off.)
4) Examples about A/D conversion are included in the CD-ROM.

The schematic of this module is shown in Figure 5-16.

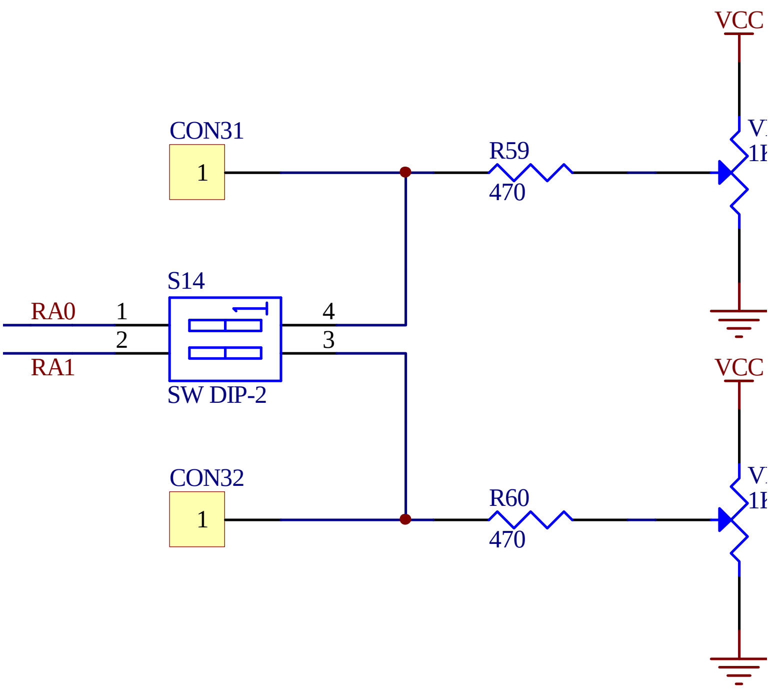


Figure 5-16 Schematic of A/D converter module

5.8 IIC protocol and 24CXX EEPROM Module

This module mainly experiments the external access device EEPROM 24CXX, and the access of MCU to 24CXX follows the IIC protocol. See Figure 5-17

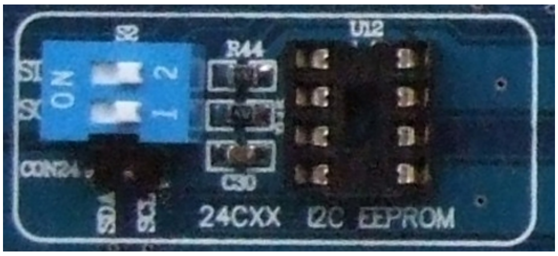


Figure 5-17 24CXX EEPROM

The module has the following main components:

1) 24CXX EEPROM
2) Two bits coding switch
3) Interface Socket

Descriptions to this module are as follows:

1) 24CXX EEPROM datasheet is included in CD-ROM.
2) The IIC communication ports of SDA and SCK are connected to the IIC communication ports RC4 and RC3 through coding switch. So you can use hardware control.
3) When this module is in use, please make sure that the coding switches are on, and while it is not in use please make sure that the coding switches are off to avoid affecting the normal operation of other modules.
4) By using the interface socket, it is also possible to achieve the communication between24 CXX and other MCU chips. (please make sure that the coding switches are off).
5) Examples about 24C01B EEPROM are included in the CD-ROM for reference.

The schematic of this module is shown in the Figure 5-18.

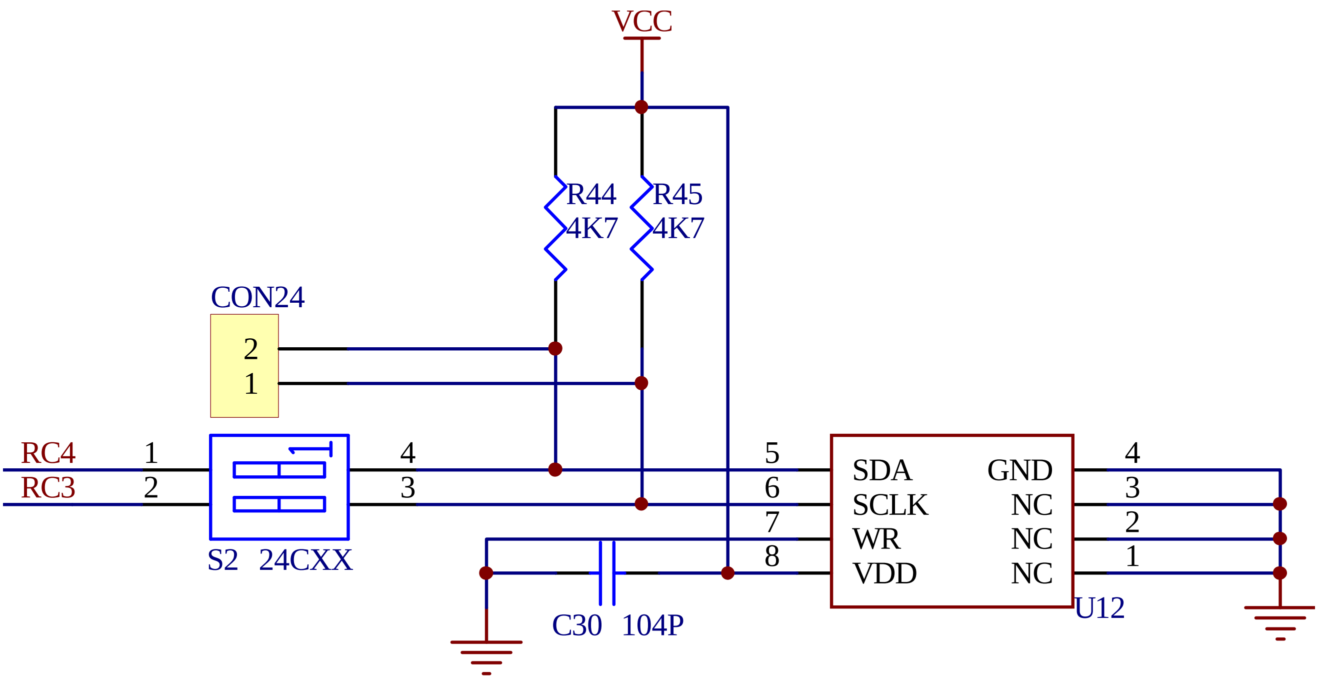


Figure 5-18 Schematic of access of MCU to 24CXX EEPROM

5.9 SIP Protocol and 93LCXXX EEPROM Module

This module mainly experiments the external access device EEPROM 93LCXXX, and the access of MCU to 93LCXXX follows the SIP protocol. See Figure 5-19

Figure 5-19 93LCXXX EEPROM

The module consists of the following components:
1) 93LCXXX EEPROM
2) Coding switch
3) Selection Jumper for ROM WORD SIZE
4) Interface socket

Descriptions to this module are as follows:
1) 93LCXXX EEPROM datasheet are contained in the CD-ROM.
2) The SPI communication ports of SDI, SDO and SCL are connected to the SPI communication ports RC5, RC4 and RC3 through coding switch. So you can use hardware control.
3) When this module is in use, please make sure that the coding switches are on, and while it is not in use please make sure that the coding switches are off to avoid affecting the normal operation of other modules.
4) By using the interface socket, it is also possible to achieve the communication between 93LCXXX and other MCU chips. (please make sure that the coding switches are off).
5) When the 93LCXXA (ROM SIZE is 8 bits) or 93LCXXB (ROM SIZE is 16 bits) are in use, the WORD SIZE jumper has no effect. When the 93CXX is in use, the jumper decides the selection of 8 bits or 16 bits of WORD SIZE.
6) Examples about 24C01B EEPROM are included in the CD-ROM for reference.

The schematic of this module is shown in the Figure 5-20.

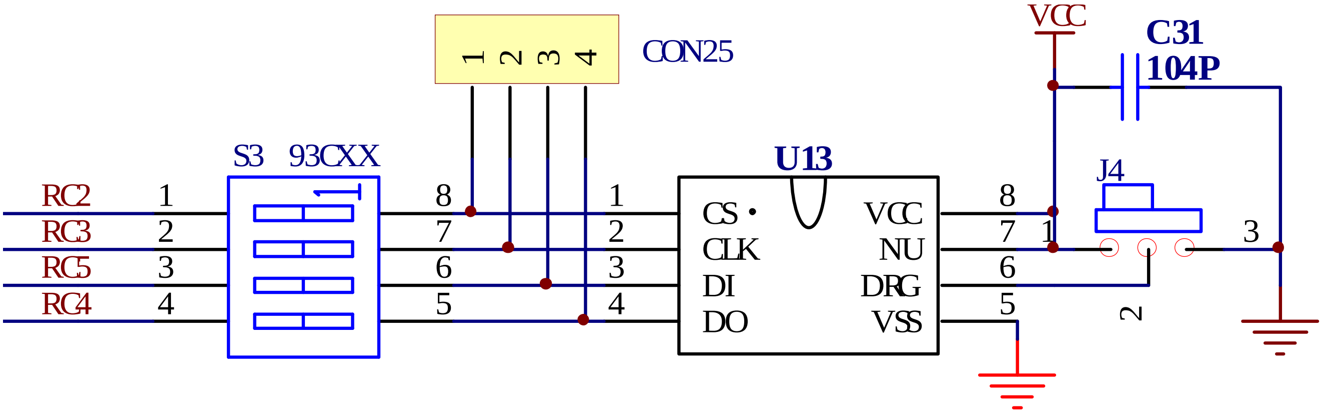


Figure 5-20 93LCXXX EEPROM schematic

5.10 DS1302 Clock Module

This module mainly experiments the use of real-time clock chip DS1302 use, as shown in Figure 5-21.

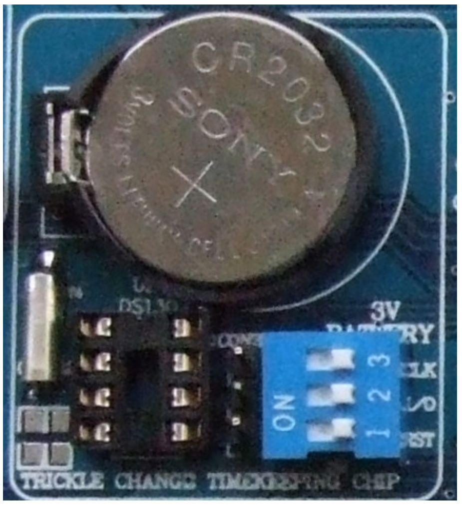


Figure 5-21 DS1302 clock module

This module consists of the following major components:
1) Clock chip DS1302
2) 3V button battery
3) Coding switch
4) 32.768K crystal oscillator
5) Interface socket

Descriptions of this module are as follows:
1) Datasheet of the chip DS1302 is included in the CD-ROM.
2) The communication ports of CLK, I/O and RST of DS1302 are connected to the MCU at RB0, RB4 and RB5 through coding switch
3) When this module is in use, please make sure that the coding switches are on, and while it is not in use please make sure that the coding switches are off to avoid affecting the normal operation of other modules.
4) By using the interface socket, it is also possible to achieve the communication between DS1320 and other MCU chips. (Please make sure that the coding switches are off.)
5) Even the 32.768K crystal start-up capacitor is not welding and will not affect the start-up of the crystal oscillation.
6) Examples about access to DS1302 are included in the CD-ROM for reference.

The schematic of this module is shown in Figure 5-22.

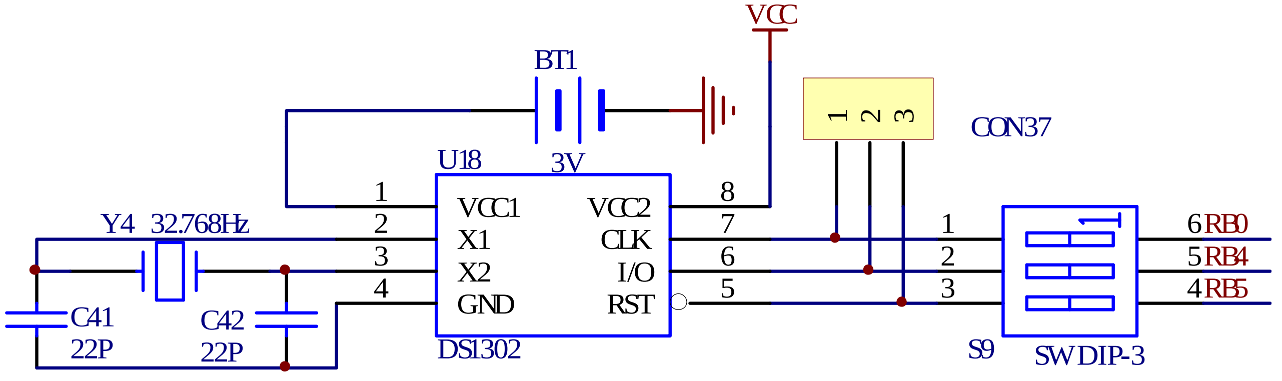


Figure 5-22 Schematic of DS1302 clock module

5.11 Six Digital LED Module

This module introduces the use of a multi digital LED, as shown in Figure 5-23.

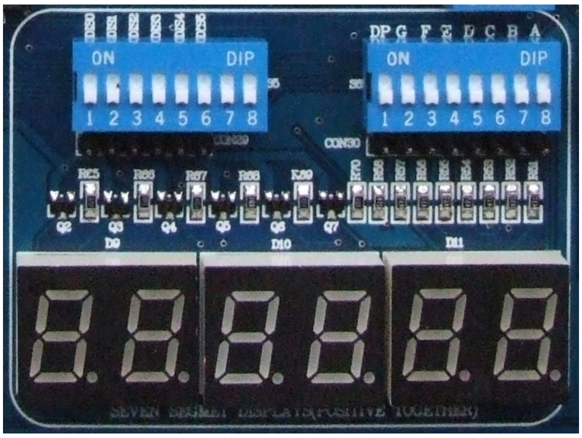


Figure 5-23 Six Digital LED Module

The module consists of the following main components:

1) 6 digital LED
2) Coding switches for bit control and section control
3) Driving circuit
4) Interface socket

Descriptions of this module are as the following:
1) The section control of LED is connected to PORTD of MCU through coding switch.
2) The bit control of LED is connected to PORTA of MCU through coding switch. When this module is in use please make sure that the coding switch are on, and when this module is not in use please make sure the coding switches are off to avoid affect on other modules.
4) By using the interface socket, it is also possible to achieve the display of LED with other MCU chips. (Please make sure that the coding switches are off.)
5) The six digital are connected to a common anode pole.
6) Examples about this module are contained in the CD-ROM for reference.

The schematic of this module is shown in Figure 5-24.

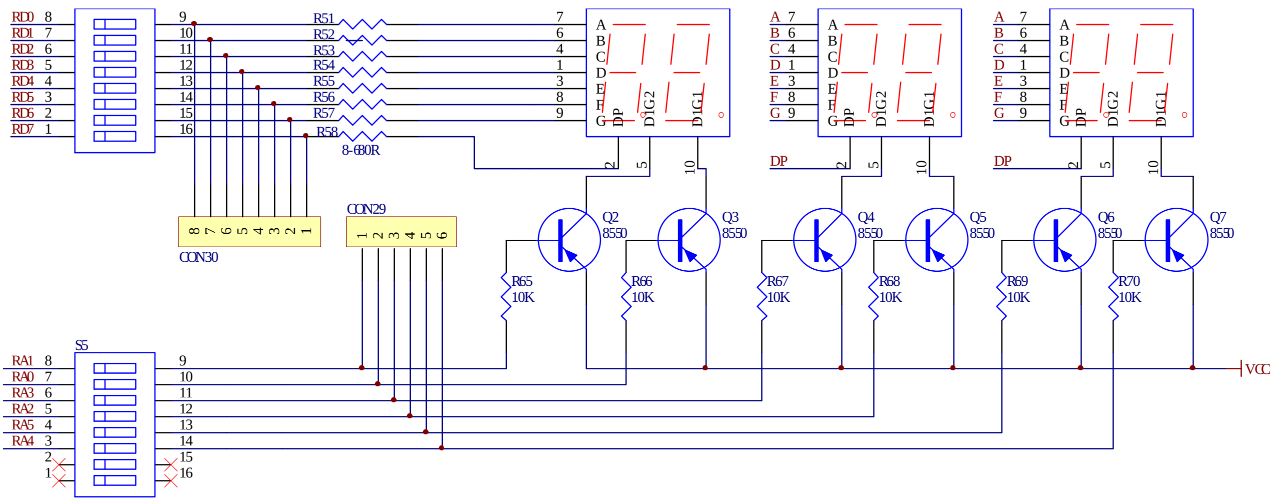


Figure 5-24 Six digital LED schematic

5.12 Remote Control Receiver Module

This module mainly experiments the receiving and decoding of infrared remote control, as shown in Figure 5-25.

Figure 5-25 Remote Control Receiver Module

This module consists of the following main components:
1) Infrared remote control receiver
2) Coding switch
3) Interface socket

Descriptions about this module are shown as following:
1) The datasheet of infrared remote control coding chip 6121 is contained in the CD-ROM.
2) The output of remote control receiver is connected to the RA1 of MCU through coding switch.
3) This module uses the second bit of 4-bit coding switch S10, while the rest 3 are shared with other modules. When this module is in use please make sure the coding switch is connected, while it is not in use please make sure the coding switch is off.
4) By using the interface socket, you may experiment the remote control decoding with other pins. (The second bit of coding switch must be off.)
5) Examples of this module are contained in the CD-ROM.

The schematic of this module is shown in Figure 5-26.

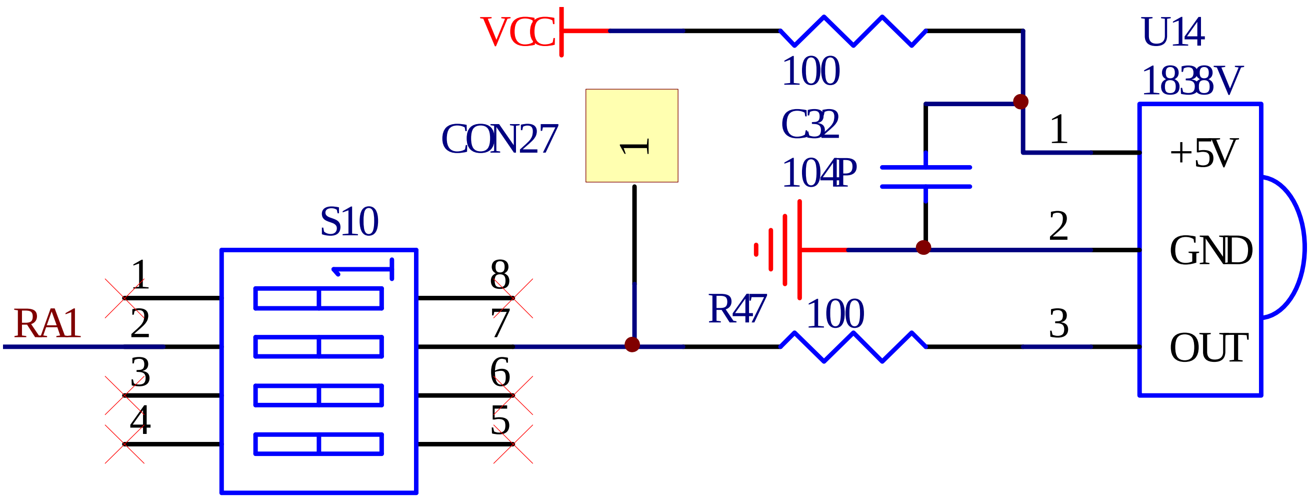


Figure 5-26 Remote Control Receiver and Decoder Module

5.13 DS18B20 Module

This module mainly experiments the use of temperature sensors DS18B20. As shown in Figure 5-27.

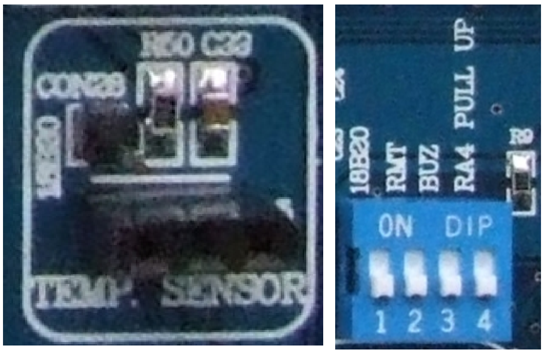


Figure 5-27 DS18B20 module

This module consists of the following major components:
1) Temperature Sensor DS18B20
2) Coding switch
3) Interface socket

Description of this module are as follows:
1) Datasheet of DS18B20 is included in the CD-ROM.
2) DS18B20 output is connected to the RA0 port of MCU through coding switch.
3) This module uses the first bit of 4-bit coding switch S10, while the rest 3 are shared with other modules. When this module is in use please make sure the coding switch is connected, while it is not in use please make sure the coding switch is off.
4) By using the interface socket, you may experiment the communication with DS18B20 through other pins. (The first bit of coding switch must be off.)
5) Examples of this module are contained in the CD-ROM.

The schematic of this module is shown in Figure 5-28.

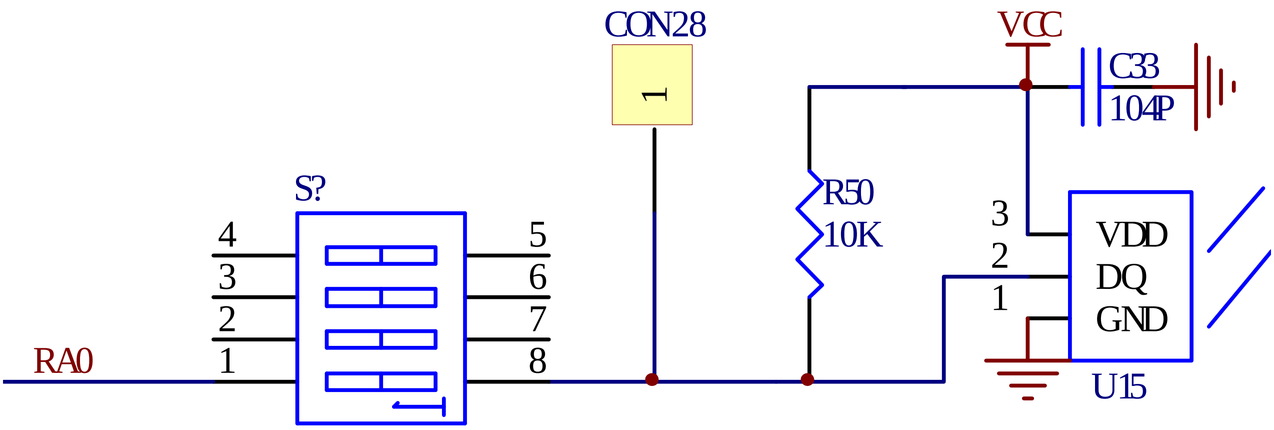


Figure 5-28 DS18B20 module

5.14 Beeper Module

This module mainly shows how to control the beeper with MCU. Please see Figure 5-29.

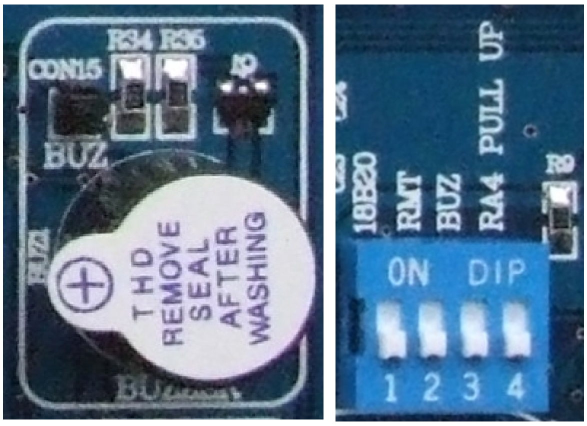


Figure 5-29 Beeper module

This module consists of the following main components:
1) Beeper
2) Coding switch
3) Interface Socket

Descriptions of this module are shown in the following:
1) The beeper is connected to the RC2 pin of MCU through coding switch.
2) This module uses the third bit of 4-bit coding switch S10, while the rest 3 are shared with other modules. When this module is in use please make sure the coding switch is connected, while it is not in use please make sure the coding switch is off.
3) By using the interface socket, you may experiment the communication with beeper through other pins. (The third bit of coding switch must be off.)
5) Examples of this module are contained in the CD-ROM.

The schematic of this module is shown in Figure 5-30.

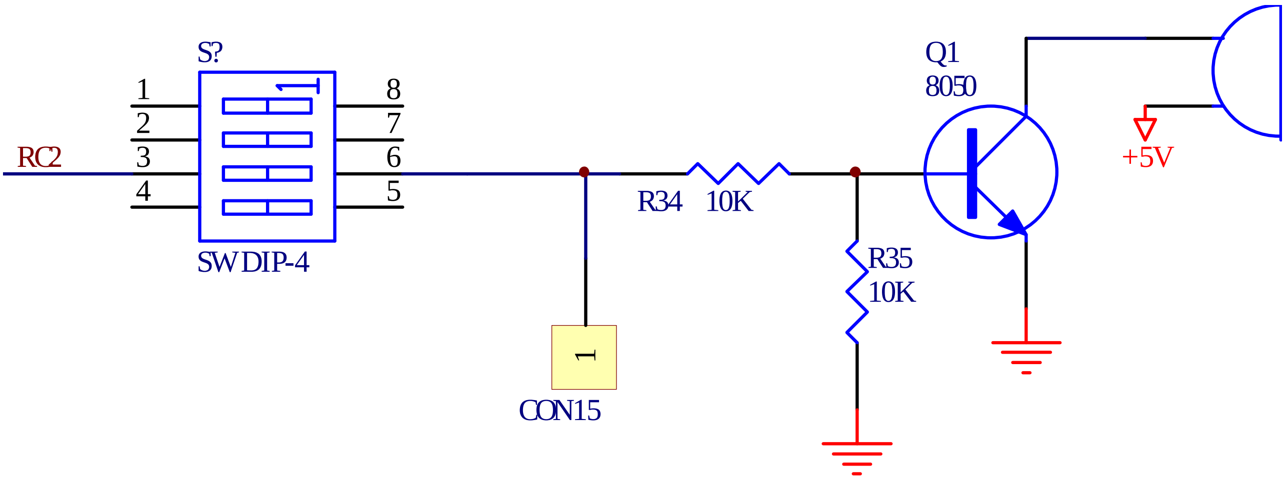


Figure 5-30 Beeper module

5.15 RS232 Communication Module

This module mainly experiments how to use the USART functional module to communicate with other external device (such as PC). See the Figure 5-31.

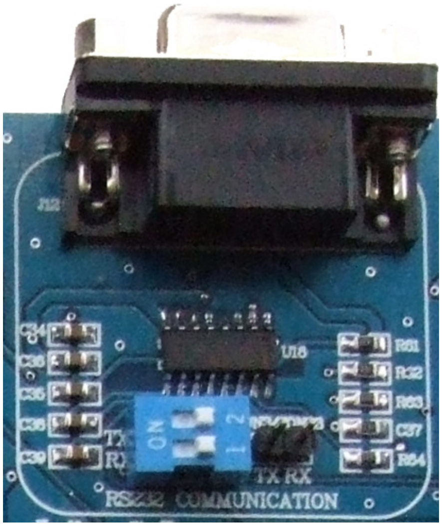


Figure 5-31 RS232 communication

This module mainly consists of the following components:
1) RS232 voltage level converter chip
2) RS232 communication port (9-pin serial port)
3) Coding switch
4) Interface socket

Descriptions of this module are as following:
1) The serial module is connected to the USART module interface RC6 and RC7 through coding switch.
2) The coding switch controls the connection between the serial module and the MCU. When this module is in use, please make sure the coding switch is on, and while it is not in use, please make sure the coding switch is off.
3) By using the interface socket, you may experiment the communication between the other pins of MCU and serial port. (use software to simulate the USART)
4) Examples about this module are contained in the CD-ROM.
The schematic of this module is shown in Figure 5-32.

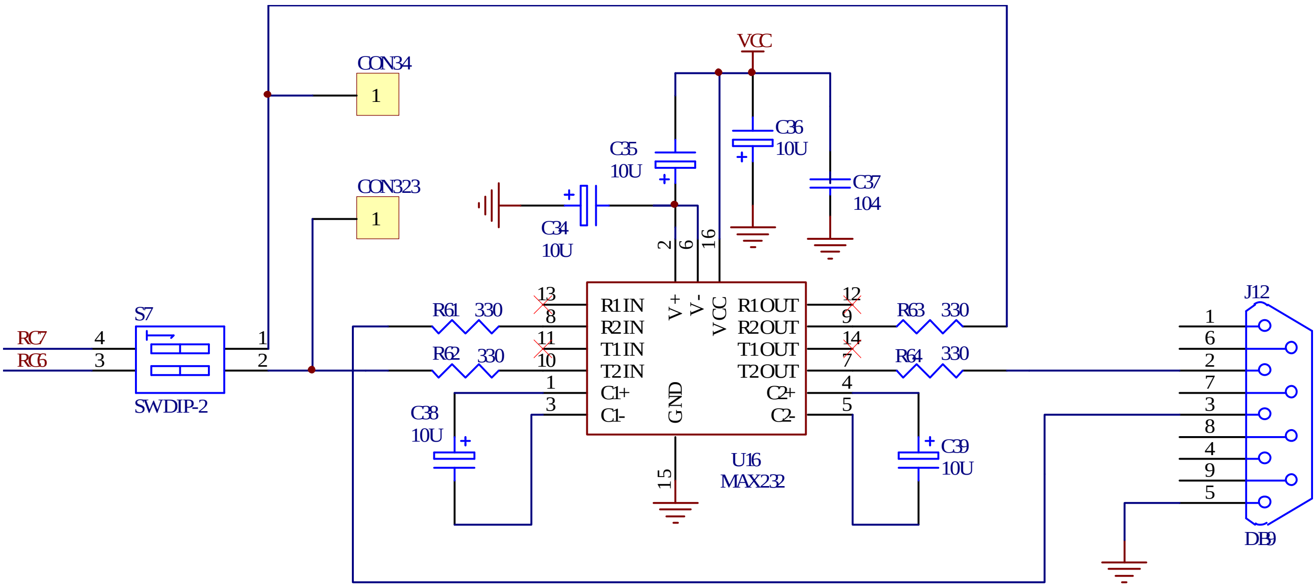


Figure 5-32 RS232 communication schematic

5.16 10-bit D/A Conversion Module

This module mainly experiments how to convert the digital signals into analog signals. As shown in Figure 5-33.

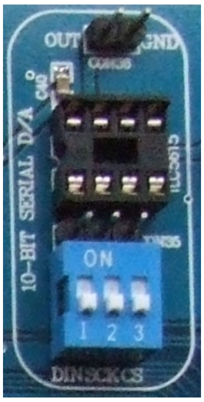


Figure 5-33 TLC5651 module

This module has the following components:
1) 10-bit D/A converter chip TLC5615
2) Result of the conversion output interface
3) Coding switch
4) Interface socket

Descriptions about this module are as the following:
1) TLC5615 information is included in the CD-ROM.
2) TLC5615 external communications follows the SPI protocol.
3) The communication ports DIN, SCK and CS of TLC5615 connected to the RC5, RC3 and RC2 of the MCU through coding switch.
2) The coding switch controls the connection between the serial module and the MCU. When this module is in use, please make sure the coding switch is on, and while it is not in use, please make sure the coding switch is off.
5) By using the interface socket, you can use other MCU ports to communicate with TLC5615 on the condition that the coding switch is off.
6) The result output socket outputs the conversion result.
7) Examples about this module are contained in the CD-ROM.

The schematic of this module is shown in Figure 5-34.

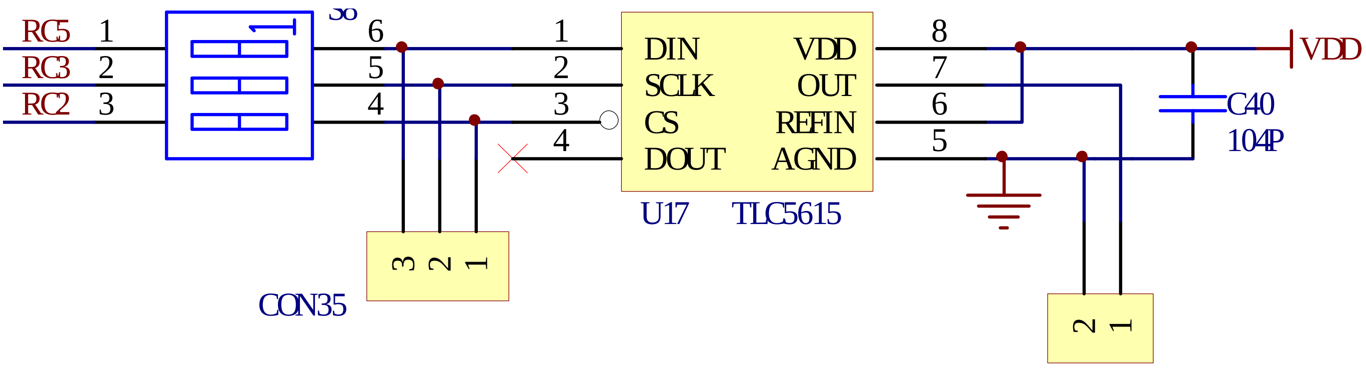


Figure 5-34 TLC5615 D/A converter schematic

5.17 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 5-35.

The module consists of the following components:

1) Stepper motor interface.

2) Drive circuit.

3) Coding switch.

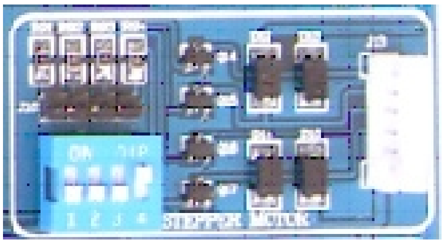


Figure 5-35 Stepper motor module

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 5-36.

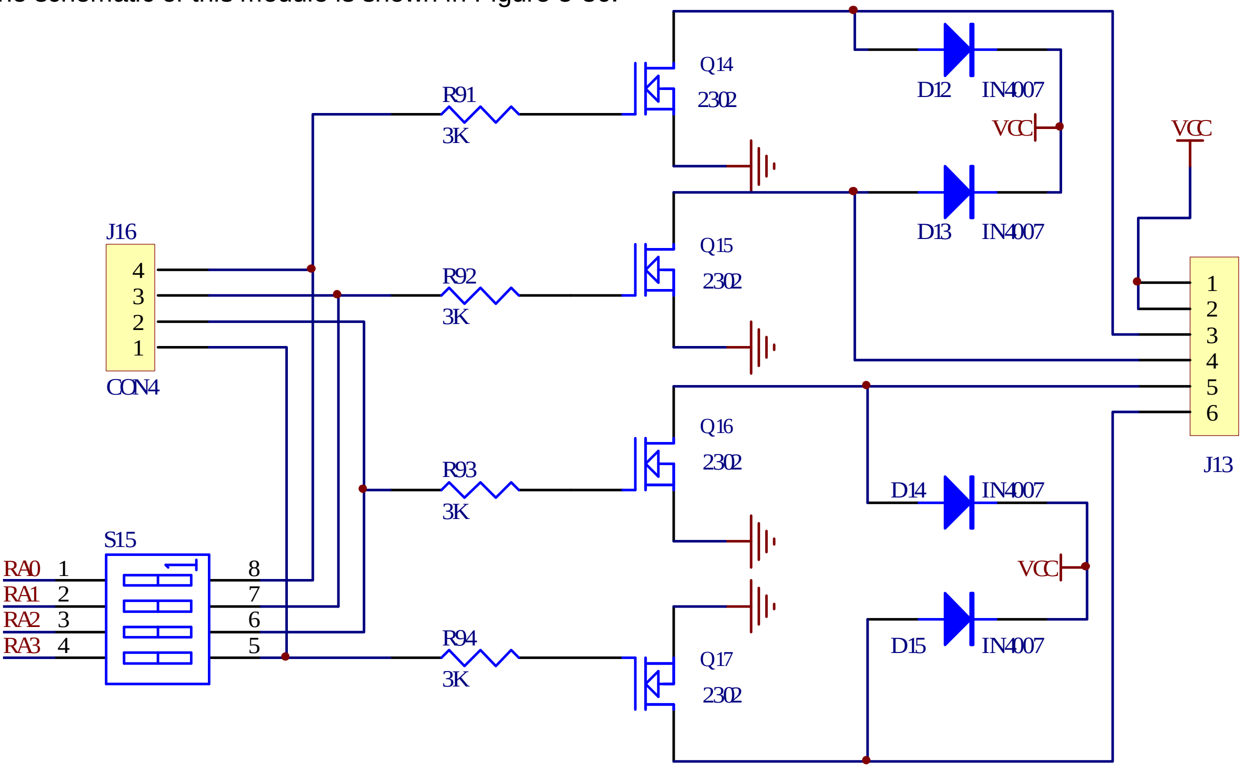


Figure 5-36 Stepper motor module schematic

5.18 SD/MMC card Read/Write Module

This module mainly shows the SD/MMC card Read/Write. Through the study of this section, we should understand the relevant knowledge about SD/MMC card Read/Write. This module is shown in Figure 4-26.

MMC card is used as storage media for a portable device,in a form that can easily be removed for access by a PC. For example,a digital camera would use an MMC card for storing image files.With an MMC reader(typically small box that connects via USB or some other serial connection).communicates with Multi Media Card via SPI communication.

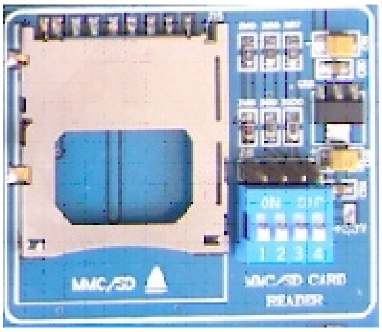


Figure 5-37 Stepper motor module

The schematic of this module is shown in Figure 5-38.

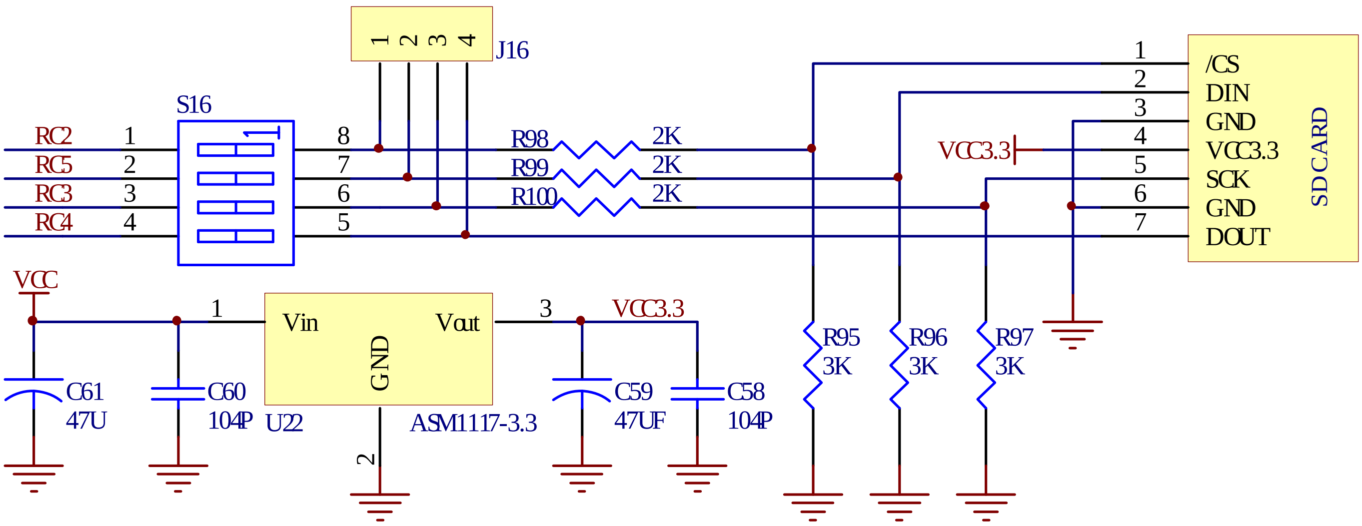


Figure 5-38 SD/MMC card Read/Write Module schematic

5.19 Example of board resource arbitrary allocation of I/O

This subsection shows how to use the existing resource to conduct LED experiment with PORTD. As shown in Figure 5-39.

After learning this section, you may be able to deduce to other cases regarding the allocation of board resources.

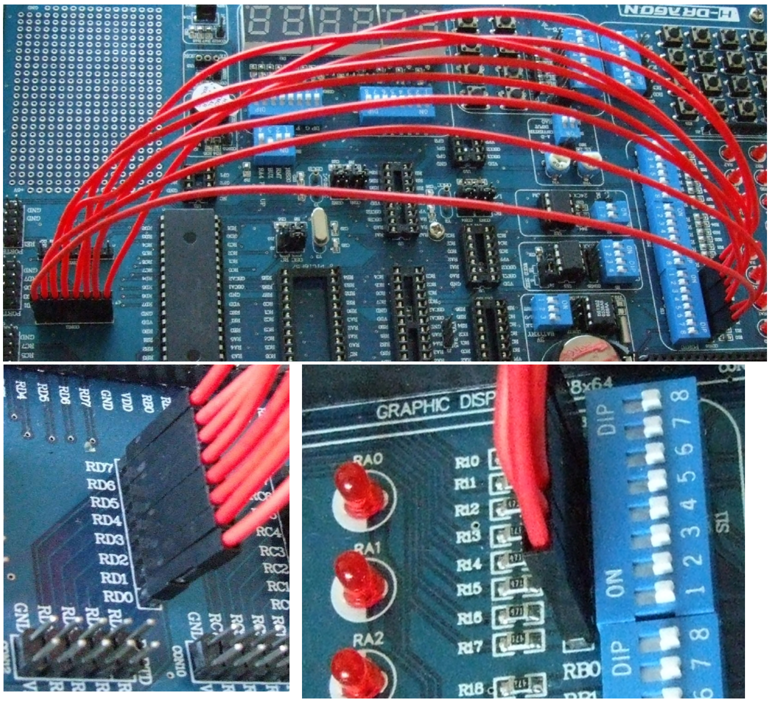


Figure 5-39 LED of PORTD expansion

Expansion method:

1) One end of the jumper connected to coding switch S11, and the other end connected to the PORTD socket of port output module.

2) All bits of coding switch S11 must be off.

3) The wiring must be at correct sequence (RD0-RC0 …… RD7 - RC7)

4) The expansion method for other modules are similar to the abovementioned procedure.

Chapter 6 QL200 System Practice

Practice Purposes:

Based on "4x4 matrix keyboard module" and the "six digital LED module", this chapter systematically describes the use of QL200 experiment board, including using MPLAB to edit and compile the source codes, using the MCD2 to debug and use QL_PROG to program the chips.

Hardware Planning:

1) 1) 4x4 matrix keyboard controlling the coding switches S4 which are all on.;

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

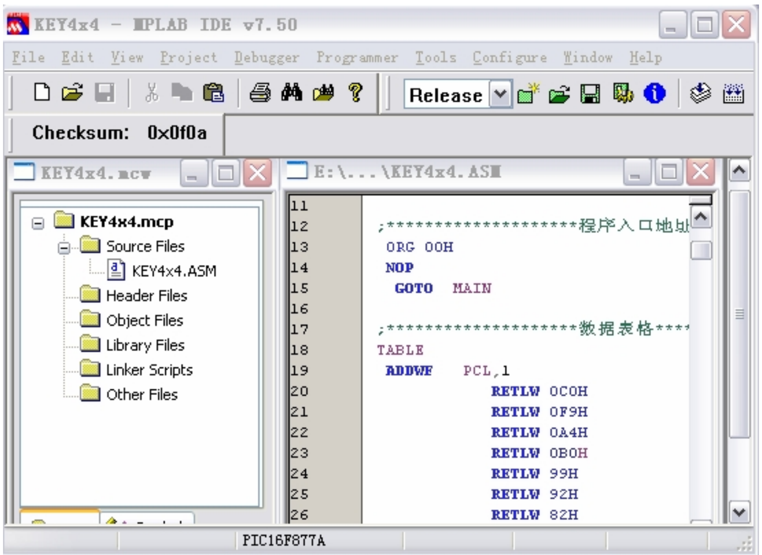


Figure 6-1 MPLAB interface after editing source code and establishing the project

3) Implement the menu command Projectà Build ALL or the 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).

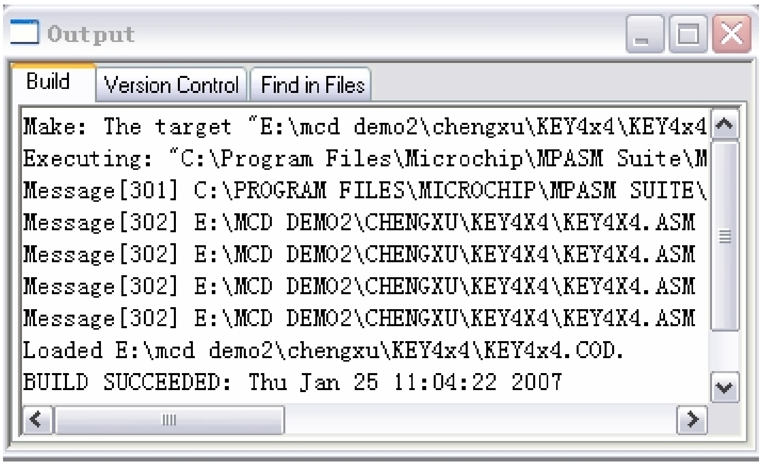


Figure 6-2 Source compiling results

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) Using MPLAB ICD2 installation wizard to set MCD2
Executive Debugger> MPLAB ICD2 Setup Wizard to run MPLAB ICD2 Setup Wizard, follow the "Next" hint to set up for all options (see below set method:)

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 6-4 with a message.
5) Set the configuration bit as Oscillator: HS, and all other options as OFF or Disable.
6) 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 in offline mode but has to be run under debugging mode), after programming it will show a message as in Figure 6-5.
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 6-6; Now if any key is pressed the LED will display a the number of that key, for example if K10 is pressed, the LED will display “10”, shown as Figure 6-7.

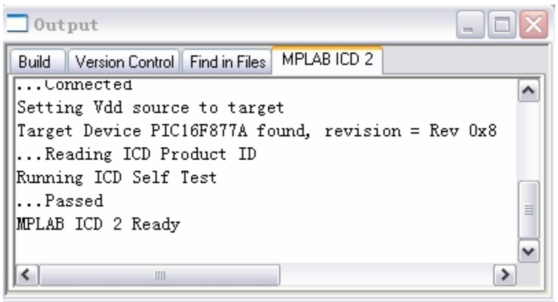


Figure 6-4 connection information

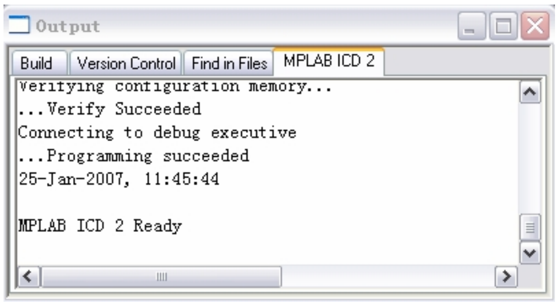


Figure 6-5 Programming Information

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

Figure 6-6 Result when no key is pressed

Figure 6-7 Result when K10 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.

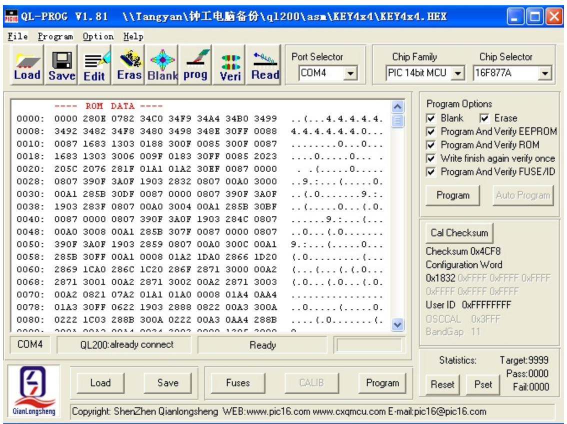
1) Disconnect the MCD2 and QL200 development board and connect the QL_PROG with target MCU in accordance with the method shown in Chapter 4.1.

2) Double click the QL-PROG icon to run QL_PROG software and wait for the software scanning of hardware.

3) Choose MCU PIC16F877 as the target MCU.

4) Execute menu command Option à View/Edit fuses AND ID or “Fuses” button to set the parameters. After that the interface will be shown as Figure 6-9.Click OK to save the settings.

5). Execute menu command Program à Program or “Program” button to program the target HEX file into the MCU chip.



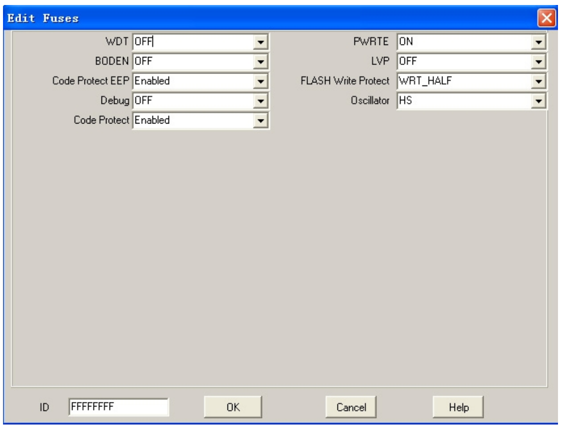


Figure 6-8 QL_PROG Buffer

Figure 6-9 Config Information

7) After programming successfully, you can see the result as shown in Figure 6-6 and 6-7.

Notes: Should you have anything unclear, please feel free to contact us by email.