MPLAB®-ICD USER'S GUIDE

MPLAB®-ICD User's Guide

Table of Contents

Chapter 3. Tutorial – Tut877

Chapter 4. Getting Started with MPLAB-ICD

Chapter 5. MPLAB-ICD Basic Functions

Table of Contents

MPLAB®-ICD User's Guide

General Information

Introduction

This first chapter contains general information that will be useful to know before running MPLAB-ICD.

Highlights

Topics covered in this chapter:

- About this Guide
- Warranty Registration
- Recommended Reading
- Troubleshooting
- The Microchip Internet Web Site
- Development Systems Customer Notification Service
- Customer Support

About This Guide

Document Layout

This document describes how to use MPLAB-ICD as a development tool to debug firmware on a target board. The manual layout is as follows:

- **Chapter 1: MPLAB-ICD Preview** – Describes what MPLAB-ICD is and how it works.
- **Chapter 2: MPLAB-ICD Installation** – Describes how to install MPLAB-ICD hardware and software and establish communications.
- **Chapter 3: Tutorial – Tut877** – Shows you how to develop and debug an application using MPLAB-IDE Projects and MPLAB-ICD.
- **Chapter 4: Getting Started with MPLAB-ICD** – Tells you how to get MPLAB-ICD up and running.
- **Chapter 5: MPLAB-ICD Basic Functions** – Describes the basic functions of MPLAB-ICD.
- **Chapter 6: MPLAB-ICD Menu Options** – Describes the menu options of MPLAB-ICD.
- **Chapter 7: Troubleshooting** – Provides information on solving common problems.
- **Appendix A: MPLAB-ICD Hardware** – Provides a technical descrip-

tion of the header, module, and demo board hardware.

- **Index** – Provides a cross-reference listing of terms, features, and sections of this document.
- **Worldwide Sales and Service** – Lists Microchip sales and service locations and telephone numbers worldwide.

Conventions Used in this Guide

This manual uses the following documentation conventions:

Table: Documentation Conventions

Description	Represents	Examples
Code (Courier font):		
Plain characters	Sample code Filenames and paths	`#define START` `c:\autoexec.bat`
Angle brackets: < >	Variables	`<label>, <exp>`
Square brackets [ ]	Optional arguments	`MPASMWIN [main.asm]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments An OR selection	`errorlevel {0|1}`
Lower case characters in quotes	Type of data	`"filename"`
Ellipses...	Used to imply (but not show) additional text that is not relevant to the example	`list ["list_option..., "list_option"]`
0xnnn	A hexadecimal number where n is a hexadecimal digit	`0xFFFF, 0x007A`
Italic characters	A variable argument; it can be either a type of data (in lower case characters) or a specific example (in uppercase characters).	`char isascii (char, ch);`
Interface (Helvetica font):		
Underlined, italic text with right arrow	A menu selection from the menu bar	*File > Save*
Bold characters	A window or dialog button to click	**OK**, **Cancel**
Characters in angle brackets < >	A key on the keyboard	<Tab>, <Ctrl-C>
Documents (Helvetica font):		
Italic characters	Referenced books	*MPLAB IDE User's Guide*

Updates

All documentation becomes dated, and this user's guide is no exception. Since MPLAB IDE, MPLAB-ICD, and other Microchip tools are constantly evolving to meet customer needs, some MPLAB dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site at http://www.microchip.com to obtain the latest documentation available.

Warranty Registration

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in your Warranty Registration Card entitles you to receive new product updates. Interim software releases are available at the Microchip web site.

Recommended Reading

This user's guide describes how to use MPLAB-ICD. The user may also find the data sheets for specific microcontroller devices informative in developing firmware.

README.ICD

For the latest information on using MPLAB-ICD, read the README.ICD file (ASCII text file) included with the MPLAB-ICD software. The README.ICD file contains update information that may not be included in this document.

README.XXX

For the latest information on other Microchip tools (MPLAB, MPASM, etc.), read the associated README files (ASCII text file) included with the MPLAB software.

MPLAB IDE User's Guide (DS51025)

Comprehensive guide that describes installation and features of Microchip's MPLAB Integrated Development Environment, including the editor and simulator functions in the MPLAB environment.

MPASM User's Guide with MPLINK and MPLIB (DS33014)

This user's guide describes how to use the Microchip PICmicro assembler (MPASM), the linker (MPLINK), and the librarian (MPLIB).

Technical Library CD-ROM (DS00161)

This CD-ROM contains comprehensive data sheets for Microchip PICmicro® devices available at the time of print. To obtain this disk, contact the nearest Microchip Sales and Service location (see back page) or download individual data sheet files from the Microchip website (http://www.microchip.com).

Embedded Control Handbook Vol.1 & 2 (DS00092 & DS00167)

These handbooks contain a wealth of information about microcontroller applications. To obtain these documents, contact the nearest Microchip Sales and Service location (see back page).

The application notes described in these manuals are also obtainable from Microchip Sales and Service locations or from the Microchip web site (http://www.microchip.com).

PICmicro Mid-Range MCU Family Reference Manual (DS33023)

This document explains the operation of the PIC16CXXX MCU family architecture and peripheral modules. To obtain this document, contact the nearest Microchip Sales and Service location (see back page).

Microsoft Windows® Manuals

This manual assumes that you are familiar with the Microsoft Windows operating system. Many excellent references exist for this software program, and should be consulted for general operation of Windows.

Troubleshooting

See Chapter 7 for information on common problems.

The Microchip Internet Web Site

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® Navigator or Microsoft Internet Explorer®. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip website is available by using your favorite Internet browser to attach to:

http://www.microchip.com

The file transfer site is available by using an FTP program/client to connect to:

ftp://ftp.microchip.com

The website and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles, and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors, and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

Development Systems Customer Notification Service

Microchip started the customer notification service to help our customers keep current on Microchip products with the least amount of effort. Once you subscribe to one of our list servers, you will receive email notification whenever we change, update, revise or have errata related to that product family or development tool. See the Microchip Internet Web Site for other Microchip list servers.

The Development Systems list names are:

- Compilers
- Emulators
- Programmers
- MPLAB
- Otools (other tools)

Once you have determined the names of the lists that you are interested in, you can subscribe by sending a message to:

```
listserv@mail.microchip.com
```

with the following as the body:

```
subscribe <listname> yourname
```

Here is an example:

```
subscribe mplab John Doe
```

To UNSUBSCRIBE from these lists, send a message to:

```
listserv@mail.microchip.com
```

with the following as the body:

```
unsubscribe <listname> yourname
```

Here is an example:

```
unsubscribe mplab John Doe
```

The following sections provide descriptions of the available Development Systems lists.

Compilers

The latest information on Microchip C compilers, Linkers, and Assemblers. These include MPLAB-C17, MPLAB-C18, MPLINK, MPASM as well as the Librarian, MPLIB for MPLINK.

To SUBSCRIBE to this list, send a message to:

```
listserv@mail.microchip.com
```

with the following as the body:

```
subscribe compilers yourname
```

Emulators

The latest information on Microchip In-Circuit Emulators. These include MPLAB-ICE and PICMASTER.

To SUBSCRIBE to this list, send a message to:

```
listserv@mail.microchip.com
```

with the following as the body:

```
subscribe emulators yourname
```

Programmers

The latest information on Microchip PICmicro device programmers. These include PRO MATE II and PICSTART Plus.

To SUBSCRIBE to this list, send a message to:

```
listserv@mail.microchip.com
```

with the following as the body:

```
subscribe programmers yourname
```

MPLAB

The latest information on Microchip MPLAB, the Windows Integrated Development Environment for development systems tools. This list is focused on MPLAB, MPLAB-SIM, MPLAB's Project Manager, and general editing and debugging features. For specific information on MPLAB compilers, linkers, and assemblers, subscribe to the COMPILERS list. For specific information on MPLAB emulators, subscribe to the EMULATORS list. For specific information on MPLAB device programmers, please subscribe to the PROGRAMMERS list.

To SUBSCRIBE to this list, send a message to:

```
listserv@mail.microchip.com
```

with the following as the body:

```
subscribe mplab yourname
```

Otools

The latest information on other development system tools provided by Microchip. For specific information on MPLAB and its integrated tools refer to the other mail lists.

To SUBSCRIBE to this list, send a message to:

```
listserv@mail.microchip.com
```

with the following as the body:

```
subscribe otools yourname
```

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Corporate Applications Engineer (CAE)
- Hotline

Customers should call their distributor, representative, or field application engineer (FAE) for support. Local sales offices are also available to help customers. See the back cover for a listing of sales offices and locations.

Corporate applications engineers (CAEs) may be contacted at (480) 786-7627.

In addition, there is a Systems Information and Upgrade Line. This line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.

The Hotline Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-786-7302 for the rest of the world.

Chapter 1. MPLAB-ICD Preview

1.1 Introduction

This section gives you a preview of MPLAB-ICD (In-Circuit Debugger) features and functions, as well as its hardware and software elements.

1.2 Highlights

Topics covered in this chapter:

- What is MPLAB-ICD
- How MPLAB-ICD Helps You
- Resources Used By MPLAB-ICD
- MPLAB-ICD Components
- MPLAB Integrated Development Environment (IDE)
- MPLAB Development Tools

1.3 What is MPLAB-ICD

MPLAB-ICD is a low cost evaluation kit for the PIC16F87X series microcontrollers. Utilizing the In-Circuit Debugging capability of the PIC16F87X and Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, the MPLAB-ICD is a programmer as well as an in-circuit debugger. It operates under the MPLAB IDE, connects to an application, and runs like the PIC16F87X microcontroller in the design.

MPLAB-ICD is intended to be used as an evaluation and debugging aid in a laboratory environment.

The MPLAB-ICD offers these features:

- Real-time and single-step code execution
- Breakpoints
- In-circuit debugging
- Built-in programming
- 3.0V to 5.5V operating range
- Operation from voltage (V_{DD}) supplied by the target application
- Operating frequencies from 32 kHz to 20 MHz
- Source level and symbolic debugging
- MPLAB IDE user interface
- Compatibility with Microsoft Windows® 3.X, Windows 95/98, Windows NT®, and Windows 2000®
- RS-232 Interface

1.4 How MPLAB-ICD Helps You

The MPLAB-ICD allows you to:

- Debug your source code in your own application
- Debug your hardware in real-time
- Program a target PIC16F87X using Microchip's ICSP protocol

1.5 Resources Used By MPLAB-ICD

Due to the built-in in-circuit debugging capability of the PIC16F87X and ICSP function offered by the debugger, the MPLAB-ICD will use the following on-chip resources:

- $\overline{\text{MCLR}}/V_{PP}$ shared for programming
- Low-voltage ICSP programming disabled
- RB6 and RB7 reserved for programming and in-circuit debugging
- Six general purpose file registers reserved for debug monitor (see below)
- First program memory location (address 0x0000) must be a NOP instruction
- The last 256 or 288 words of program memory area are reserved for Debug Code (depending on device, see below)
- One stack level not available

Data and Program Memory Used by MPLAB-ICD

The MPLAB-ICD uses the following file register and program memory locations in the PIC16F87X target device:

Processor	File Registers Used	Program Memory Used
PIC16F870/871/872	0x70, 0x0BB-0x0BF	0x06E0-0x07FF
PIC16F873/874	0x70, 0x0EB-0x0F0	0x0EE0-0x0FFF
PIC16F876/877	0x70, 0x1EB-0x1EF	0x1F00-0x1FFF

1.6 MPLAB-ICD Components

The MPLAB-ICD consists of four basic components:

1. MPLAB-ICD module
2. MPLAB-ICD header
3. MPLAB-ICD demo board
4. MPLAB IDE software (on your PC)

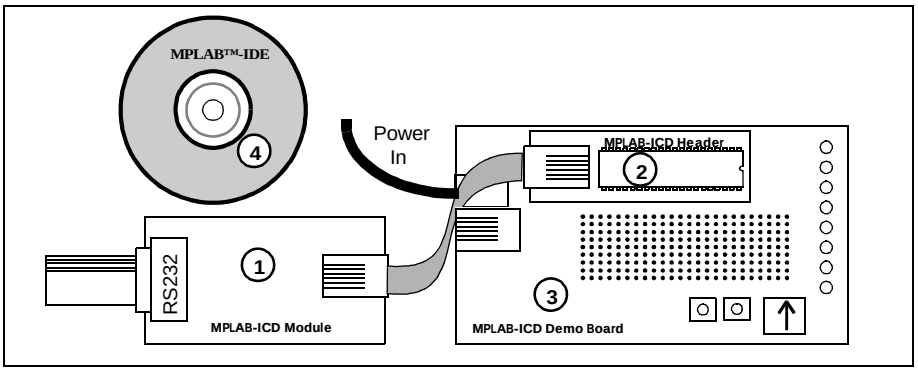


Figure 1.1: MPLAB-ICD Components

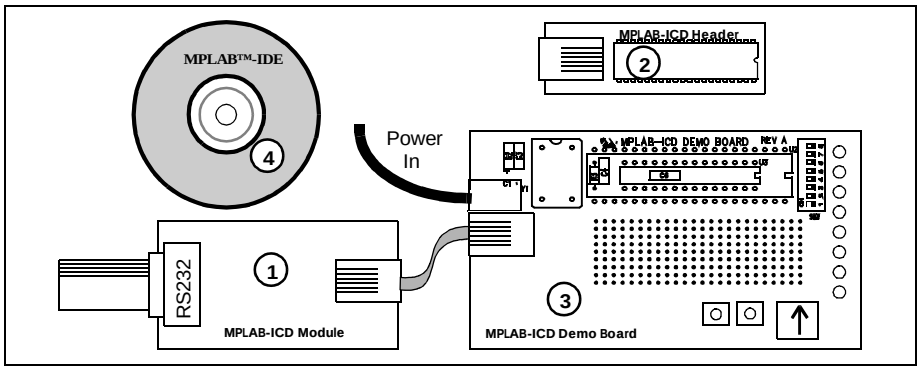


Figure 1.2: MPLAB-ICD Module Connected Directly to Demo Board

1.6.1 MPLAB-ICD Module

The MPLAB-ICD module contains all debugging, programming, and control logic. It is connected to the PC's serial port via a 9-pin serial cable and to the MPLAB-ICD target or demo board using a 6-wire modular cable. If the application board does not include a 6-wire modular connection (shown as item 12 in Section 1.6.3), the module can be connected to the MPLAB-ICD header as a means of providing this connection.

The module contains the firmware to provide serial communications to the PC, to drive the MPLAB-ICD communications to the target board, demo board (or to the header), and to program a target PIC16F87X using ICSP, all from the MPLAB IDE. The module is powered from the target application (or demo board) and requires 70 mA (max) in addition to what the target consumes.

The modular cable can be plugged into a modular connector on **either the MPLAB-ICD header or** the application circuit with the appropriate connections to the PIC16F87X to allow in-circuit emulation of QFP and DIE parts **as described in Section 2.5.2**.

1.6.2 MPLAB-ICD Header

As a convenience, the MPLAB-ICD header provides a means of connecting the MPLAB-ICD module to a target board that does not incorporate a 6-wire modular connection into its design. The MPLAB-ICD module connects to the MPLAB-ICD header via a 9-inch modular cable. For in-circuit emulation, a **40-pin** PIC16F87X needs to be plugged into the header which then plugs into a 28-pin or 40-pin PIC16F87X DIP socket on an application or the MPLAB-ICD demo board **as described in Section 2.5.1**.

The MPLAB-ICD header is powered by the target application, from a 3.0V to 5.5V source.

1.6.3 MPLAB-ICD Demo Board

The demo board is provided for demonstration and evaluation of the PIC16F87X without a target application board. It can be connected to the MPLAB-ICD module directly or via the MPLAB-ICD header. The PIC16F877 can be unplugged from the header and plugged directly into the demo board for stand-alone operation.

The demo board (Figure 1.3) has the following hardware features:

1. 40- and 28-pin sockets. For information on using stand-offs to select the desired pin count socket, see Section A.3.1
2. Eight DIP switches to connect and disconnect each of the eight LEDs to and from its respective PORTC pin.
3. Eight red LEDs connected to PORTC for displaying 8-bit binary values.
4. Two push-button switches, one for RESET and one for external stimulus on RB0.
5. A potentiometer for analog input on RA0.

6. A small prototyping area.
7. A connector area to access the I/O pins of the PIC16F87X for expansion prototyping.
8. A jumper to select the RC oscillator (approximately 2 MHz) or an external crystal.
9. Socket for external crystal.
10. A connector for a 9V, 0.75A power supply, similar to the PICSTART® Plus programmer power supply.
11. Provisions for a MAX232 and associated hardware that may be populated to add RS-232 capability.
12. A modular cable connection that can be used to connect the demo board directly to the MPLAB-ICD module.

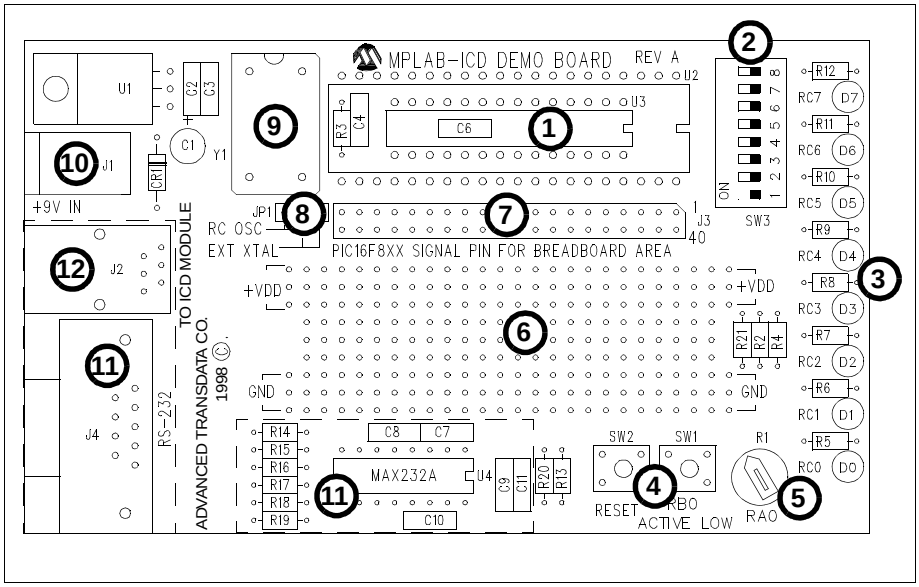


Figure 1.3: MPLAB-ICD Demo Board

1.6.4 MPLAB IDE Software

The MPLAB IDE software runs in the Windows 3.1, Windows 95/98, Windows NT, or Windows 2000 environment. It provides full display, modification, and control of the target application under emulation.

1.7 MPLAB Integrated Development Environment

The MPLAB desktop provides an environment for developing and debugging your application. MPLAB-ICD is integrated into the MPLAB IDE.

This document covers the basic setup and operation of the MPLAB-ICD, but it does not cover all functions of the MPLAB IDE. Read the *MPLAB IDE User's Guide* (DS51025) to get a full understanding of the features and debug capabilities of the MPLAB IDE.

1.8 MPLAB Development Tools

The MPLAB IDE integrates several tools to provide a complete development environment.

- **MPLAB Project Manager**

  Use the Project Manager to create a project and work with the specific files related to the project. When using a project, source code is rebuilt and downloaded to the simulator or emulator with a single mouse click.

- **MPLAB Editor**

  Use the MPLAB Editor to create and edit text files such as source files, code, and linker script files.

- **MPLAB-SIM Simulator**

  The software simulator models the instruction execution and I/O of the PICmicro Microcontrollers (MCUs).

- **MPLAB-ICE Emulator**

  The MPLAB-ICE emulator uses hardware to emulate PICmicros in real time, either with or without a target system.

- **MPASM Universal Assembler/MPLINK Relocatable Linker/ MPLIB Librarian**

  The MPASM assembler allows source code to be assembled without leaving MPLAB. MPLINK creates the final application by linking relocatable modules from MPASM and MPLAB-C17 (PIC17CXXX C compiler). MPLIB manages custom libraries for maximum code reuse.

- **PRO MATE® II and PICSTART® Plus Programmers**

  Develop code with the simulator or an emulator, assemble or compile it, and then use one of these tools to program devices. This can all be accomplished with MPLAB. Although PRO MATE II does not require MPLAB to operate, programming is easier using MPLAB.

- **Third Party Tools**

  Many other companies have development tools for Microchip products that work with MPLAB. Consult the *Microchip Third Party Guide* (DS00104).

Chapter 2. MPLAB-ICD Installation

2.1 Introduction

This chapter describes how to install the MPLAB-ICD hardware and software and establish communications between the MPLAB-ICD demo board and PC.

2.2 Highlights

Topics covered in this chapter:

- MPLAB-ICD Kit Components
- Host Computer System Requirements
- Installing the Hardware
- Installing the Software

2.3 MPLAB-ICD Kit Components

The MPLAB-ICD system kit includes the following items (Figure 2.1):

1. The MPLAB-ICD header
2. The MPLAB-ICD module
3. The MPLAB-ICD demo board
4. RS-232 cable
5. 40-pin DIP and 28-pin SDIP connection sockets
6. 9-inch 6-conductor modular cable
7. CD with complete MPLAB software and documentation
8. Manuals:
   - *MPLAB-ICD User's Guide* (this document)
   - *MPLAB IDE User's Guide* (not shown)
   - *MPASM with MPLINK and MPLIB User's Guide* (not shown)
9. A Warranty Registration card (not shown)

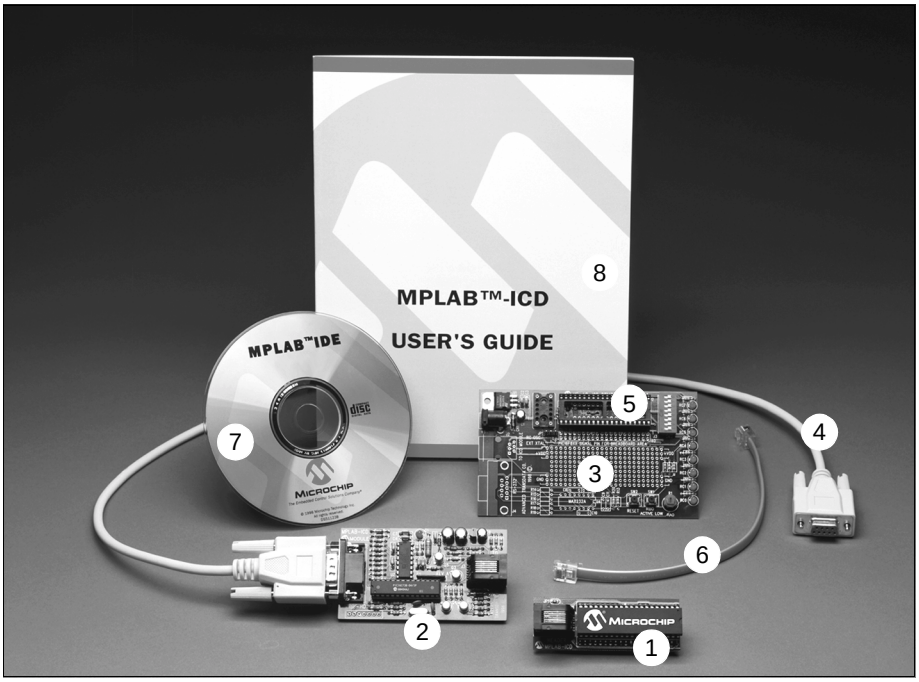


Figure 2.1: MPLAB-ICD Kit Components

2.4 Host Computer System Requirements

To take advantage of the debugger system features, you must install the MPLAB software on a host computer having the following minimum configuration:

- Pentium-class PC-compatible machine
- Microsoft Windows 3.X, Windows 95/98, WIndows NT, or Windows 2000
- 16 MB RAM, 32 MB recommended
- 45 MB available hard disk space
- One available serial port
- 9V, 0.75A Power Supply (PICSTART Plus or equivalent)

2.5 Installing the Hardware

2.5.1 Connecting the Module, Header, and Demo/Target Board

Install the MPLAB-ICD system hardware by following these steps:

1. Plug the 40-pin PIC16F87X device into the 40-pin DIP socket in the MPLAB-ICD header.

To debug this part on the target or demo board	Use this part on the header
PIC16F870	PIC16F871
PIC16F871	PIC16F871
PIC16F873	PIC16F874
PIC16F874	PIC16F874
PIC16F876	PIC16F877
PIC16F877	PIC16F877

Note: Devices without a 40-pin equivalent cannot be used with the MPLAB-ICD header. To debug such devices, you must connect the module directly to the demo/target board as described in Section 2.5.2.

2. Connect the 9-inch modular cable between the MPLAB-ICD module and the MPLAB-ICD header.
3. Plug the 40-pin connection socket into the 40-pin DIP socket in the MPLAB-ICD demo board or target application. If using the demo board, make sure to insert the 40-pin stand-off first (see Section A.3.1).
4. Plug the MPLAB-ICD header into the connection socket in the demo board or target application (Figure 2.2). If you are debugging a 28-pin part, connect the MPLAB-ICD header to a 28-pin SDIP socket in the demo board/target via the 28-pin SDIP connection socket.
5. Connect the RS-232 cable between the serial port of the host computer and the MPLAB-ICD module.
6. Turn on the power to the host computer.
7. Turn on the power to the MPLAB-ICD demo board/target application which also powers the MPLAB-ICD module. (For the demo board, you can use the 9 VDC center-positive power adapter supplied with the PICSTART Plus Programmer.)

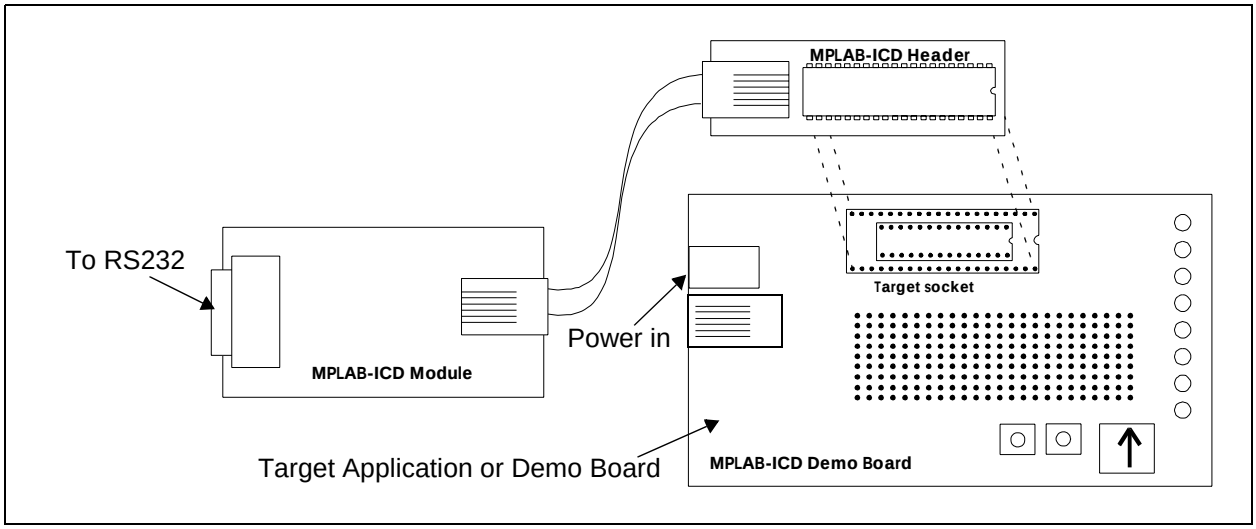


Figure 2.2: MPLAB-ICD Hardware Connection

2.5.2 Connecting the Module and Demo/Target Board

You can connect the module directly to your target board or to the demo board if your target board incorporates the modular cable (RJ-6) connection described in the following table.

J2 Pin	Signal
6	RB3
5	RB6
4	RB7
3	Ground
2	+V_{DD}
1	V_{PP}

This configuration also provides the full MPLAB-ICD functionality.

1. Plug a PIC16F87X into the 28-pin or 40-pin DIP socket in the MPLAB-ICD demo board or target board.
2. Connect the 9-inch modular cable between the MPLAB-ICD module and the MPLAB-ICD demo board or your target board.
3. Connect the RS-232 cable between the serial port of the host computer and the MPLAB-ICD module.
4. Turn on the power to the host computer.
5. Turn on the power to the MPLAB-ICD demo board/target application which also powers the MPLAB-ICD module. (For the demo board, you can use the 9 VDC center-positive power adapter supplied with the PICSTART Plus Programmer.)

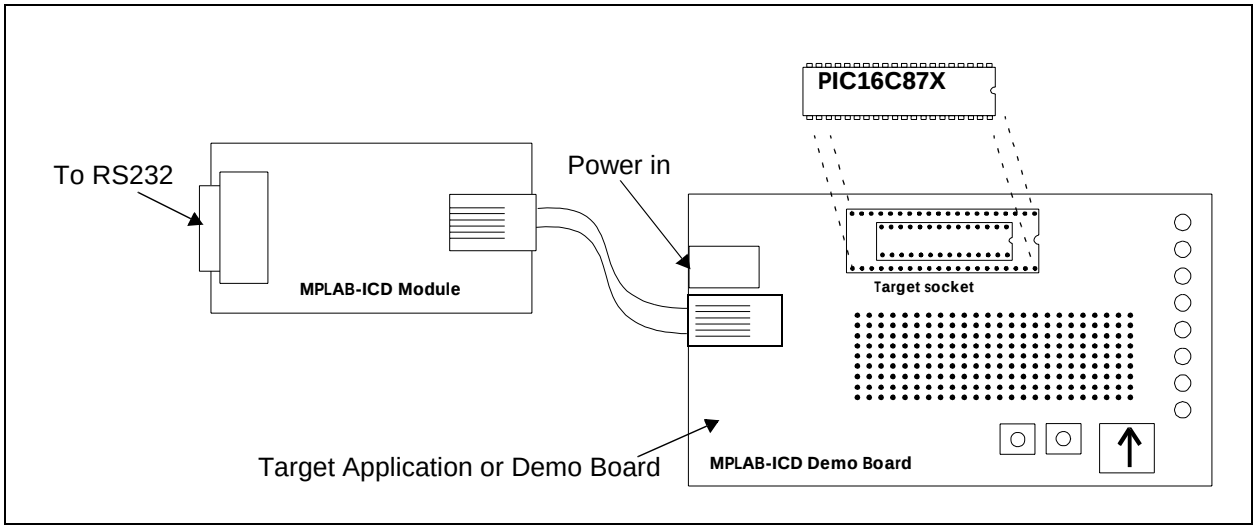


Figure 2.3: MPLAB-ICD Module and Board Connection (Without Header)

2.6 Installing the Software

The MPLAB-ICD is an add-on tool for the MPLAB IDE. To install MPLAB IDE, follow these steps:

1. Enter Microsoft Windows and insert the MPLAB IDE CD-ROM into the CD-ROM drive.
2. Execute the setup program:
   a) Windows 3.1: From the File Manager, or from the *Program Manager>Run* option, run **X:\MPvvvvv.exe**, where **X** is the drive designation of the CD-ROM drive, and **vvvvv** is the version number of MPLAB.
   b) Windows 95/98: Click the **Start** button and select Run. Enter **X:\MPvvvvv.exe**, where **X** is the drive designation of the CD-ROM drive, and **vvvvv** is the version number of MPLAB. Then click **OK**.
3. Follow the instructions to install the MPLAB IDE with MPLAB-ICD support. Be sure to check mark the MPLAB-ICD component on the component selection dialog that appears immediately after the Welcome dialog.

Chapter 3. Tutorial – Tut877

3.1 Introduction

The MPLAB-ICD is a programmer for the PIC16F87X family as well as an in-circuit debugger. This tutorial will help you start using the MPLAB-ICD hardware and software to program your part and debug source code.

This tutorial uses a sample project, Tut877. The `tut877.asm` program is a simple implementation of the PIC16F877's analog-to-digital (A/D) converter using the MPLAB-ICD demo board. This program configures the A/D module to convert on A/D channel 0 (connected to the potentiometer on the demo board) and display the results on the LEDs on PORTC.

3.2 Highlights

Topics covered in this chapter:

- Creating a Hex File for Debugging
- Setting up the MPLAB-ICD and MPLAB IDE
- Setting ICD Programming and Debugging Options
- Programming the PIC16F877
- Setting up the Demo Board
- Running Tut877
- Debugging Tut877
- Tut877 Functionality

3.3 Before You Begin

This project requires the MPLAB-ICD debugger to be connected to your PC as described in Section 2.5. MPLAB IDE must also be installed and running.

3.4 Creating a Hex File for Debugging

You will need to create a new project to include the source file `tut877.asm` and build the hex file `tut877.hex`. See the next sections for instructions on how to create a new project, `tut877.pjt`.

3.4.1 New Project Directory

In File Manager (Windows 3.1) or Windows Explorer, create a subdirectory for the new project, `\MPLAB\tut877`. Move the `tut877.asm` file from `\MPLAB` to this subdirectory.

3.4.2 New Project

Select *Project > New Project*, select the directory for the new project, and then enter `tut877.pjt` in the File Name box. Click **OK**.

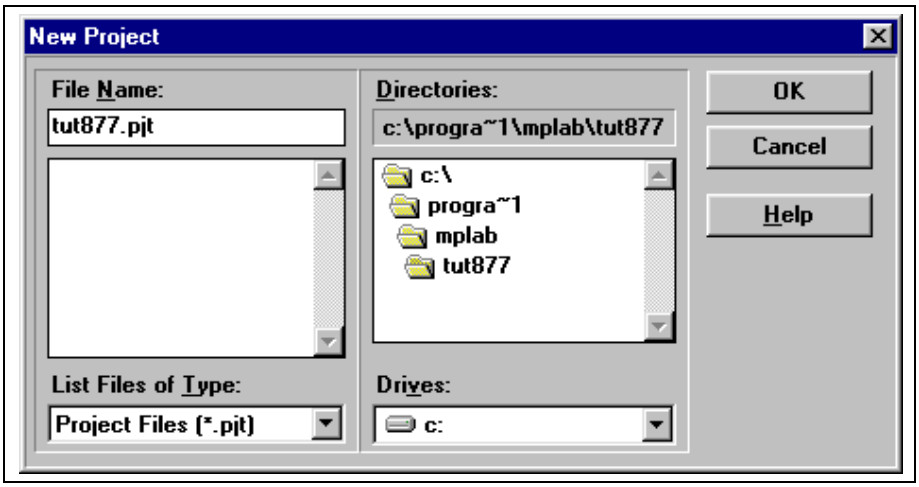


Figure 3.1: New Project - sample.pjt

3.4.3 Project Dialog

The Edit Project dialog should open.

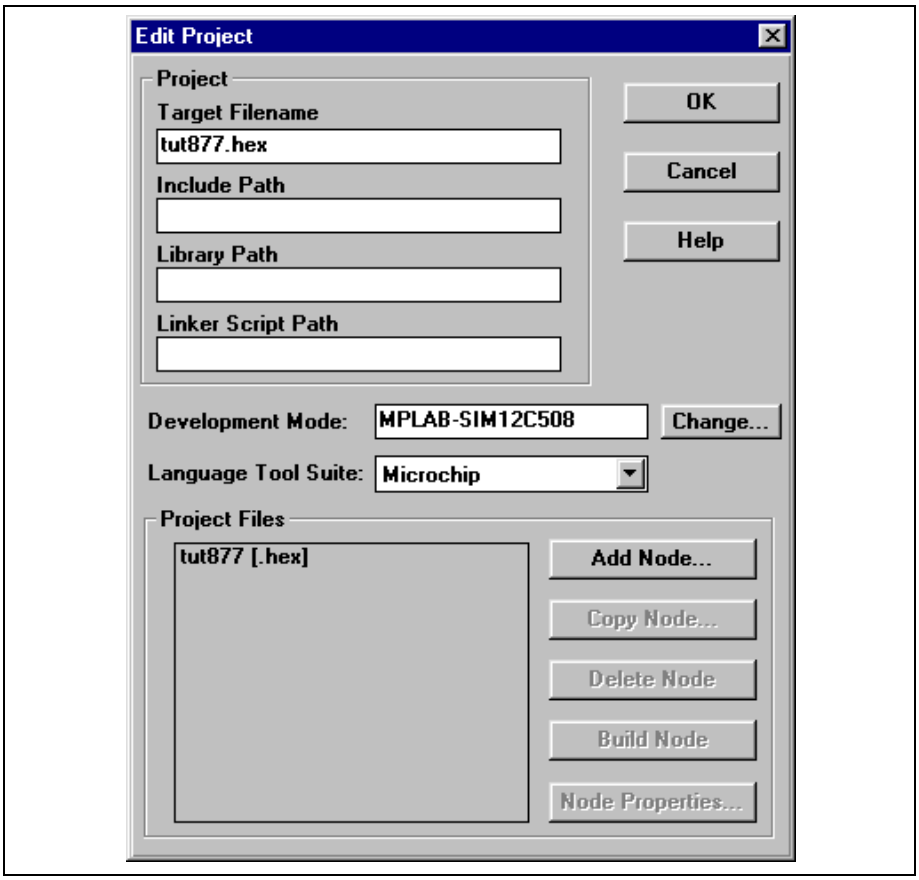


Figure 3.2: Edit Project Dialog Before Setting Development Mode

Notice the development mode. It indicates we are working with the MPLAB-SIM simulator and a PIC12C508 processor. Your display will indicate whatever development mode and processor you were working with previously in MPLAB IDE. We will need to change these settings now.

Click **Change**.

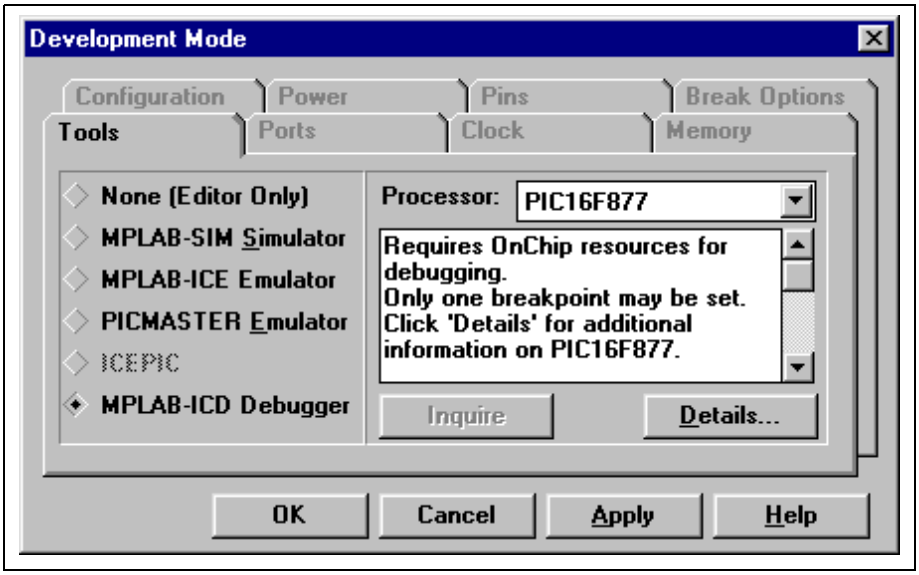


Figure 3.3: Setting the Development Mode

Select MPLAB-ICD Debugger under Tools. Make sure you select the PIC16F877 processor. Click **OK**.

MPLAB IDE will now establish communications with the MPLAB-ICD. The MPLAB-ICD dialog will momentarily appear during this process. If you receive an error message, double-check the connections for power supply, socket seating, and cable seating. For more detailed troubleshooting information, refer to Chapter 7.

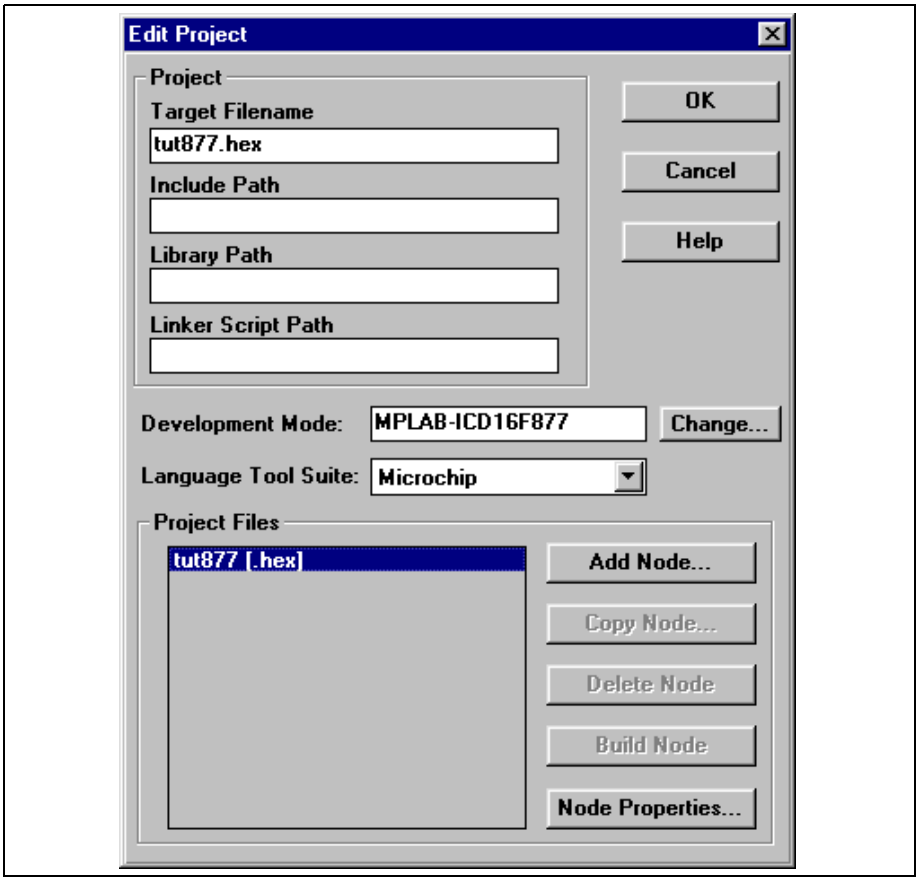


Figure 3.4: Edit Project Dialog

Notice that the correct development mode and processor are shown in the Edit Project dialog.

Click on the line `tut877 [.hex]` in the Project Files area of the Edit Project dialog, and then click the **Node Properties** button.

3.4.4 Set Node Properties

The Node Properties dialog shows the command line switches for the tool, in this case MPASM. When you first open this dialog, the checked boxes represent the default values for the tool. For this tutorial, these settings do not need to be changed from their defaults.

Click **OK** to return to the Edit Project dialog.

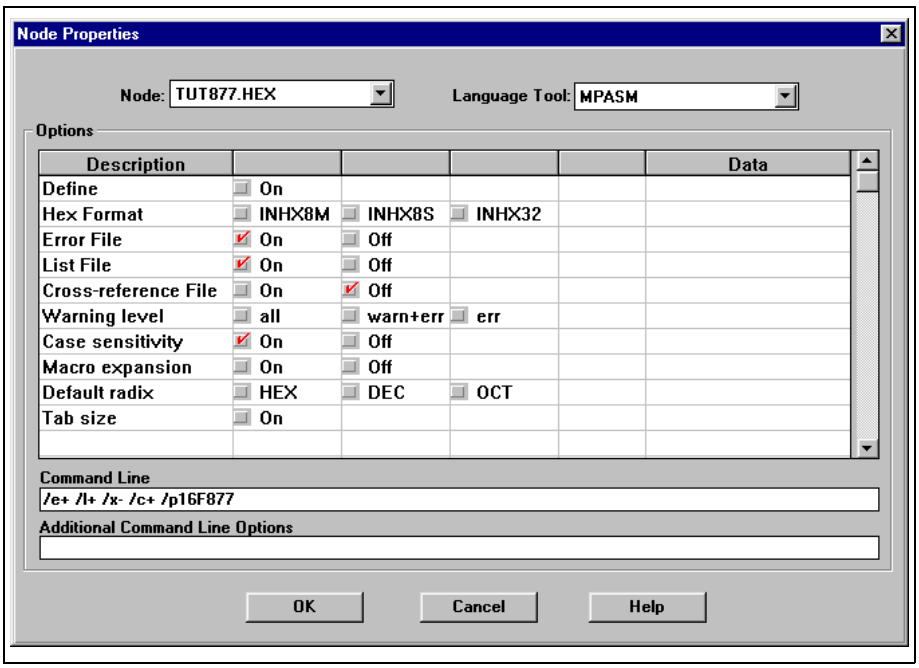


Figure 3.5: Node Properties Dialog

3.4.5 Add Node

Click **Add Node** from the Edit Project dialog to open the Add Node dialog. Select `tut877.asm` for this tutorial, and then click **OK**.

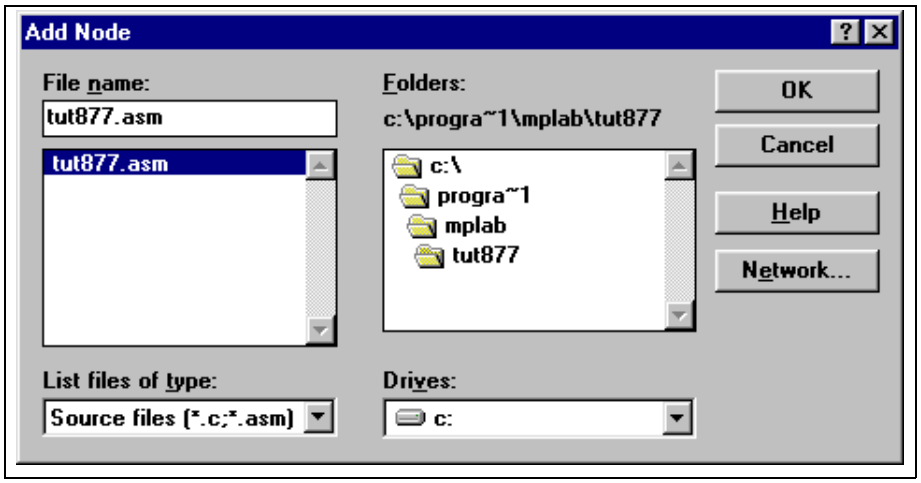


Figure 3.6: Add Node

3.4.6 Complete Project Setup

In this simple example, no entries were made in the Path boxes. As your application becomes more complex, you may need to enter the directories of your Include Files in the appropriate boxes.

MPASM always makes a `.hex` file with the same name as the source `.asm` file. The Project Manager will create a `tut877.hex` file when the project is built.

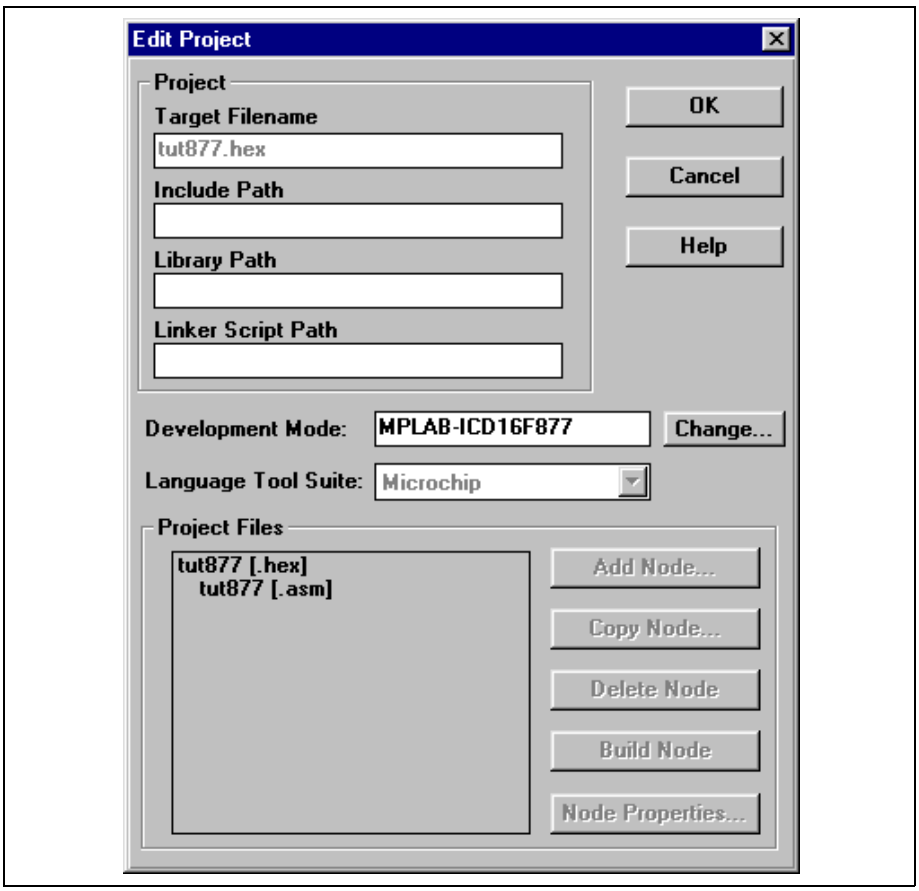


Figure 3.7: Edit Project Dialog with Node

Click **OK** to close the Edit Project Dialog.

Now select *Project > Save Project* from the MPLAB IDE menu to save your project.

3.4.7 Make Project

Select *Project > Make Project* from the menu to compile the application using MPASM. A Build Results window shows the command line sent to the assembler. It should look like this:

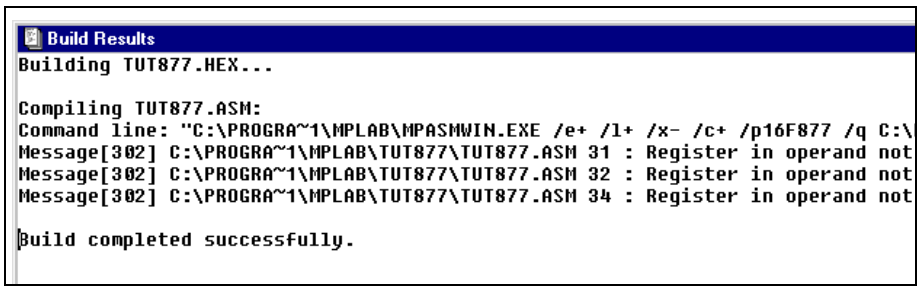


Figure 3.8: Build Results Window

Note: Message [302] is simply letting you know that, at the designated line number (i.e., 31, 32 and 34) you are specifying a file register that is not in bank zero. This is not an error and your code will compile correctly.

Click the X in the upper right corner of the Build Results window to close it. Or, click the Restore icon next to the X to resize the window. Subsequent builds will use this smaller Build Results window.

3.5 Setting Up MPLAB-ICD and MPLAB

At this point, the MPLAB-ICD dialog should be on your desktop. Make the selections described in this section to set up MPLAB IDE for use with the MPLAB-ICD hardware.

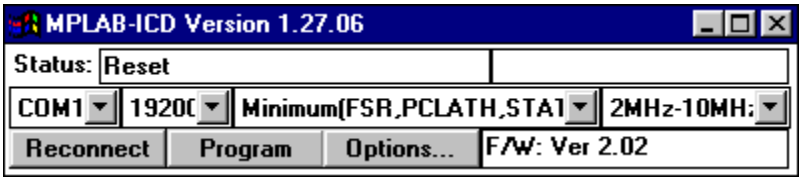


Figure 3.9: MPLAB-ICD Dialog

Table 3.1: MPLAB-ICD Dialog

Item	Options
Status	The Status bar displays the executed MPLAB-ICD function and the status. When you program a device, you can watch the progress in this area. When the operation is complete, the Status box displays the message "Waiting for user command."
COM Port and Baud Rate Pull-Down Menus	Make sure these values match the settings of your operating system.
Upload Options Pull-Down Menu	Select Minimum for now. Later in the tutorial we will change this for debugging.
Operating Frequency Range Pull-Down Menu	Select the operation frequency range of 2 MHz – 10 MHz.

Click **Options** to continue with the tutorial.

3.6 Setting ICD Programming and Debugging Options

To program the target PIC16F877 device, the ICD Options dialog must first be set up for programming. The small MPLAB-ICD dialog was opened when you entered the MPLAB-ICD development mode. Click **Options** in the MPLAB-ICD dialog to open the ICD Options dialog.

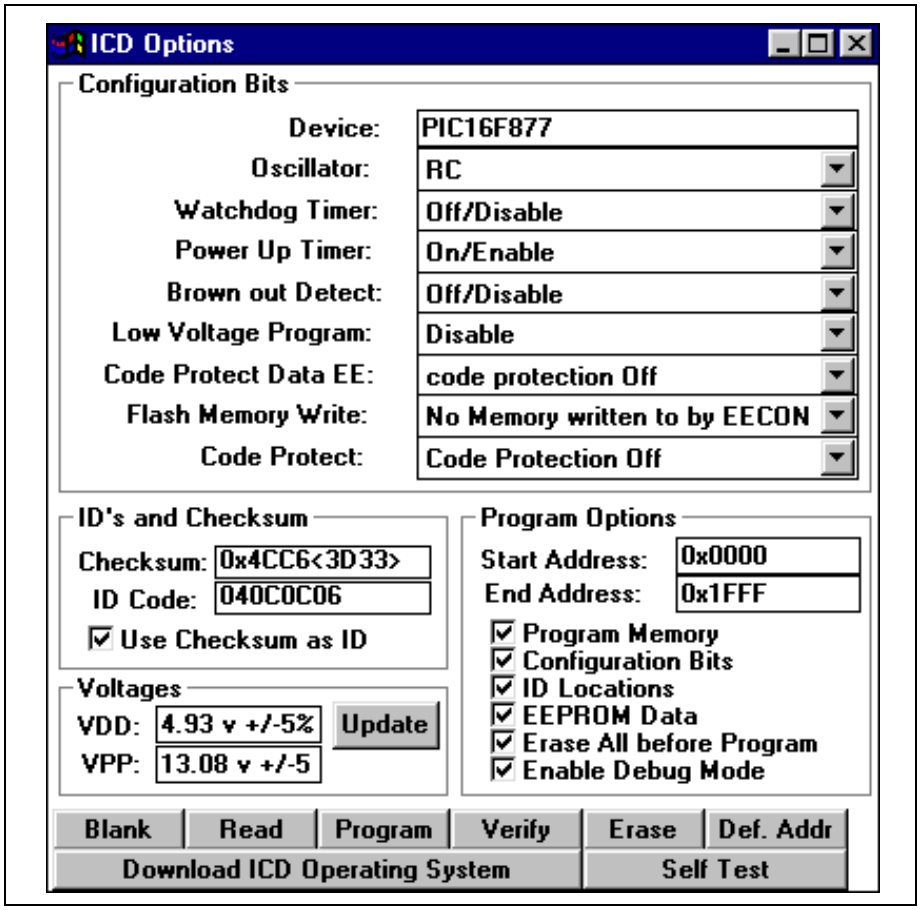


Figure 3.10: MPLAB-ICD Options Dialog

3.6.1 Configuration Bits and Device Selection

This section of the ICD Options dialog allows you to set the various configuration bits on the PIC16F87X processors. Click the arrow and select from the list.

Table 3.2: Configuration Bits and Device Selection

Item	Options
Device	The PIC16F877 device should be shown, as selected in the Development Mode dialog. If not, use the Project menu items to save your project and close the project. Select *Options > Development Mode*, select the correct device, and click **OK**. Then re-open the Tut877 project.
Oscillator	RC is used in this tutorial. Check the demo board to make sure the oscillator jumper JP1 is correctly placed for the RC OSC option (see item 8 in Figure 1.3). The RC frequency for the demo board is approximately 2MHz.
Watchdog Timer	For this tutorial, the Watchdog Timer (WDT) should be off/disabled.
Power Up Timer	For this tutorial, the Power Up Timer (PWRT) should be on/enabled.
Brown Out Detect	For this tutorial, the Brown Out Detect (BOD) should be off/disabled.
Low Voltage Program	Low voltage ICSP programming should be disabled when using the MPLAB-ICD. This means that you may use RB3 as digital I/O and you must use HV on MCLR for programming.
Code Protect Data EE	Turn off code protection for this tutorial.
Flash Memory Write	No memory will be written to EECON for this tutorial.
Code Protect	Turn off code protection for this tutorial.

3.6.2 IDs and Checksum

The checksum for the data and the ID code are displayed here. For this tutorial, Use Checksum as ID by selecting the checkbox.

3.6.3 Voltages

Allows you to check the V_{DD} and V_{PP} voltages on the target application by clicking the **Update** button.

MPLAB-ICD develops its needed $V_{PP} \approx 13V$ from the target board's V_{DD} through use of a switching boost converter.

3.6.4 Program Options

The program address range (Start Address and End Address) is the range of program or data memory that will be read, programmed, or verified. The default program address range is set to the maximum program memory available based on the device you selected. This tutorial will use the default.

Make sure that all checkboxes under Program Options are checked. This means that all memory, ID, and configuration bits will be programmed. Also, all memory will be erased before programming and Debug Mode will be enabled.

3.6.5 Function Buttons

During the tutorial you will click these buttons to perform the assigned function on the PIC16F87X in the MPLAB-ICD header or target/demo board.

3.7 Programming the PIC16F877

Click the **Program** button to program `tut877.hex` and debug code into the PIC16F87X in the MPLAB-ICD header or demo board. Programming may take a couple of minutes. During programming, the Status box shows the current phase of the operation. When programming is complete, the Status box displays the message “Waiting for user command.”

Note:	The debug code is special code at 1F00h-1FFFh in the PIC16F877 that must be present to use the in-circuit debugging capabilities of the MPLAB-ICD.

You can minimize or move the MPLAB-ICD dialog, but do not close it. Closing the MPLAB-ICD dialog will disable the ICD. To reenable the ICD, you will have to select *Options > Development Mode*. Select Editor Only and click **Apply**. Then select MPLAB-ICD and click **OK** to reenable the ICD.

3.8 Setting Up the Demo Board

Before we begin our debugging, let's make sure the demo board, is set up:

- the RC OSC option has been selected using jumper JP1.
- the DIP switch (SW3) has all switches in the ON position, to connect all LEDs to their respective PORTC pin.

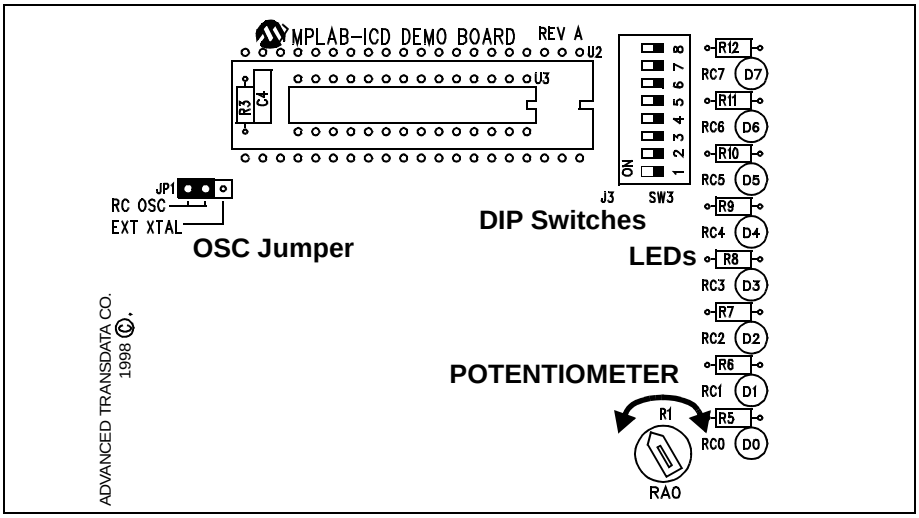


Figure 3.11: Setting Up the Demo Board

3.9 Running Tut877

The MPLAB-ICD executes in real-time mode or in step mode.

- Real-time execution occurs when the PIC16F87X in the MPLAB-ICD header is put in MPLAB IDE's Run mode.
- Step mode execution can be accessed after the processor is halted.

Begin in real-time mode:

- Open the `tut877.asm` file for viewing (*Open > File*)
- Click the Run toolbar button or issue the *Debug > Run > Run* command

The Status bar at the bottom of the MPLAB IDE desktop should turn yellow. To change the colors that signify a program run, select *Options > Environment Setup* and click the **Colors** tab.

On the demo board, turn the arrow on the potentiometer (RA0) till it points to the DIP switch (SW3). Observe the LEDs. If the program were working correctly, you would see a binary representation of the voltage value across the potentiometer. However, an insidious bug has been placed in the Tut877 program! Of course, this will give us the opportunity to debug the code.

Click the Halt toolbar button or issue the *Debug > Run > Halt* command to stop the program execution. Reset the program by selecting *Debug > Run > Reset*.

3.10 Debugging Tut877

Any of the following could be preventing the Tut877 program from working:

- The A/D converter value is not being properly written to PORTC (LEDs)
- The A/D converter is not on or has not been set to convert
- A typo in the source code is causing the program to function improperly

To explore the first possibility, you will set a break point at the line of the file that writes the value of A/D result to PORTC. Highlight or place the cursor on the following line of code from `tut877.asm`:

```
movf ADRESH,W ;Write A/D result to PORTC
```

Click on the right mouse button to access a shortcut menu. Select *Break Point(s)* from the shortcut menu as shown in Figure 3.12. This line will now be marked as a break point. To change the colors that mark a break point, select *Options > Environment Setup* and click the **Colors** tab.

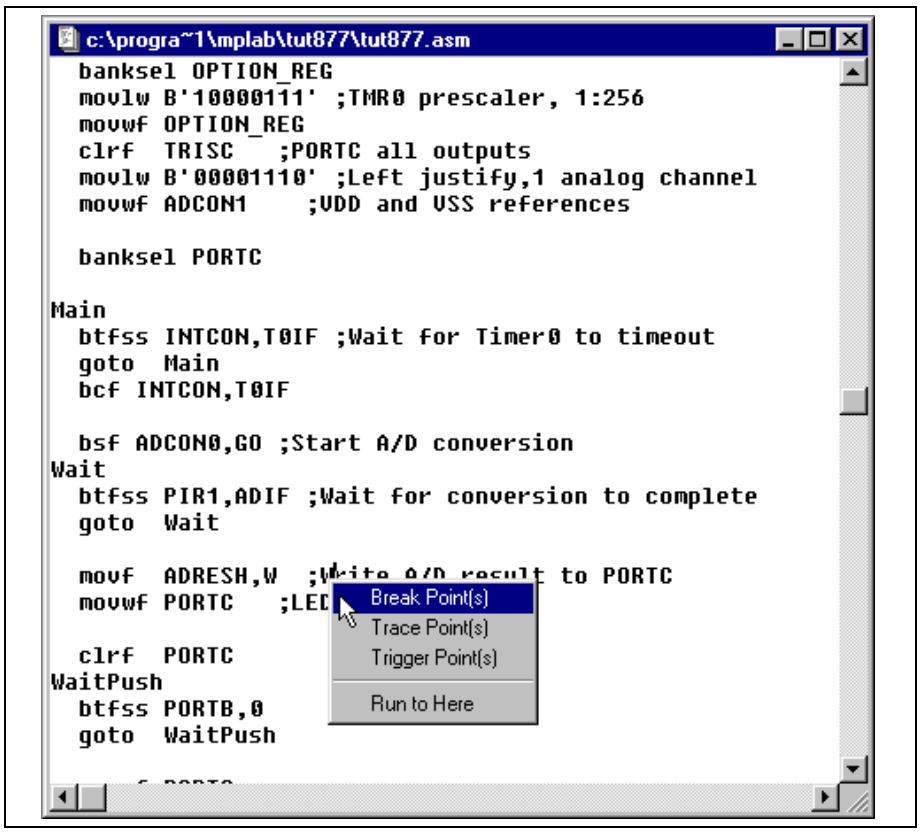


Figure 3.12: Set Breakpoint

Click the Run toolbar button or issue the *Debug > Run > Run* command to run the program once again in real-time mode.

A break point will stop a program's execution when the program executes the line marked as a break point. However, our sample program is not halting. Therefore, halt it yourself now by clicking the Halt toolbar button or issuing the *Debug > Run > Halt* command.

Look at the source code (`tut877.asm`) window and notice where the program halted. Our sample program halted on one of the two lines in the Wait routine as shown in Figure 3.13. Based on the halt location and the fact that the program never reaches the break point, we conclude that the problem is in the A/D conversion—the A/D flag for conversion complete is not being set.

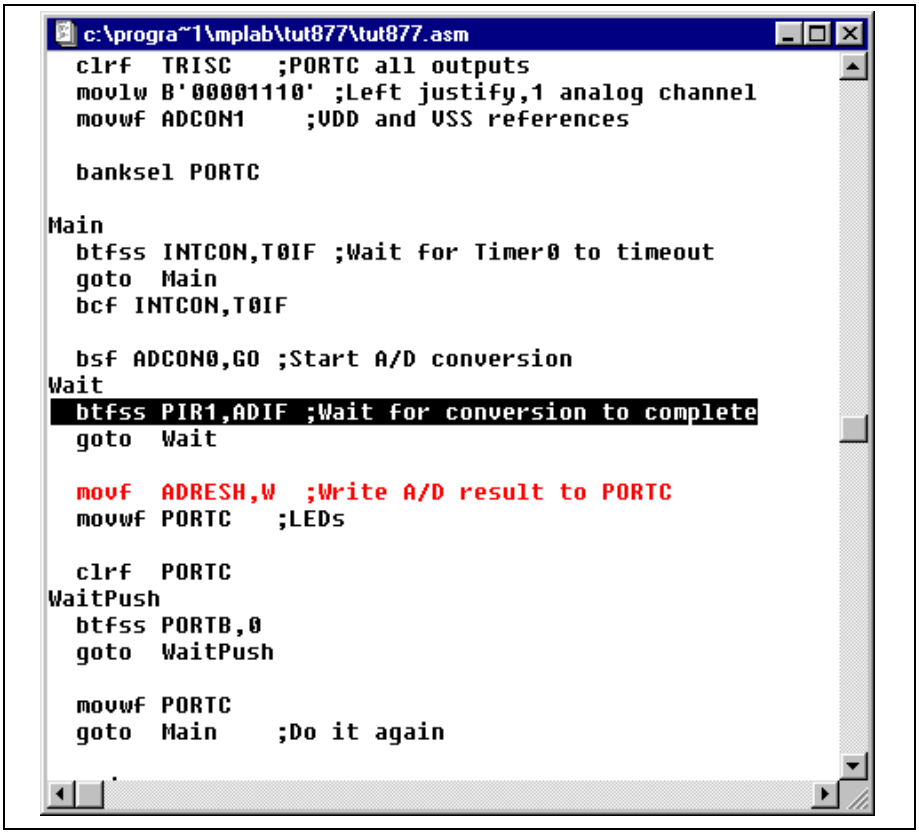


Figure 3.13: Program Halted

A/D conversion initialization and setup occurs at the beginning of the program. To check out this code, first reset the program by selecting *Debug > Run > Reset*. The first instruction after Start should be highlighted.

Open a new watch window so you can watch the A/D register values change as the program executes. Select *Window > Watch Windows > New Watch Window*. The Add Watch Symbol dialog will open, with the Watch_1 new watch window behind it. Add the symbols ADCON0 and ADCON1 as shown in Figure 3.14.

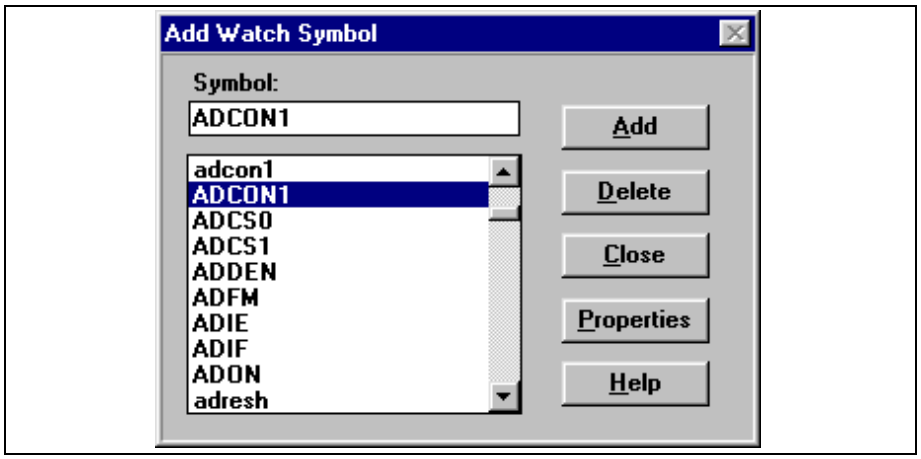


Figure 3.14: Adding Watch Symbols

Click **Close** when finished. The selected symbols should now be visible in the watch window as shown in Figure 3.15.

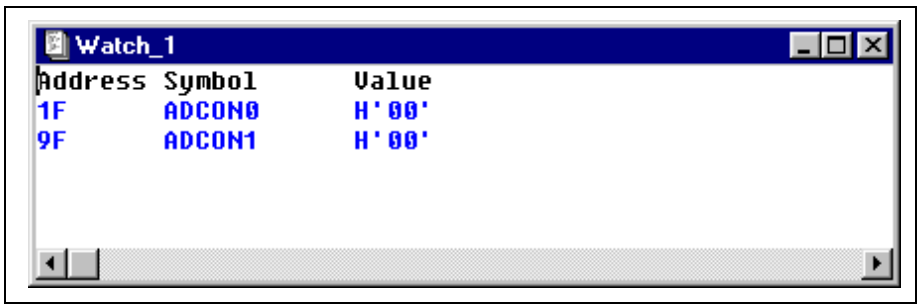


Figure 3.15: Watch Window

In the MPLAB-ICD dialog, set the upload options to Minimum & Watch Windows. The program may run slower now, but this setting allows us to see the Watch windows for debugging. Close or minimize the MPLAB-ICD Options dialog.

In the tut877.asm source code, set a break point at the second instruction after Start. Again, highlight or place the cursor on the following line of code from `tut877.asm`:

```
clrf PORTC ;Clear PORTC
```

Click on the right mouse button to access a shortcut menu. Select *Break Point(s)* from the shortcut menu. This line will now be marked as a breakpoint.

Finally, click the Run toolbar button or issue the *Debug > Run > Run* command to run the program in real-time mode.

This time the program will stop after it executes the breakpoint line of code, and the instruction after the breakpoint will be highlighted as shown in Figure 3.16.

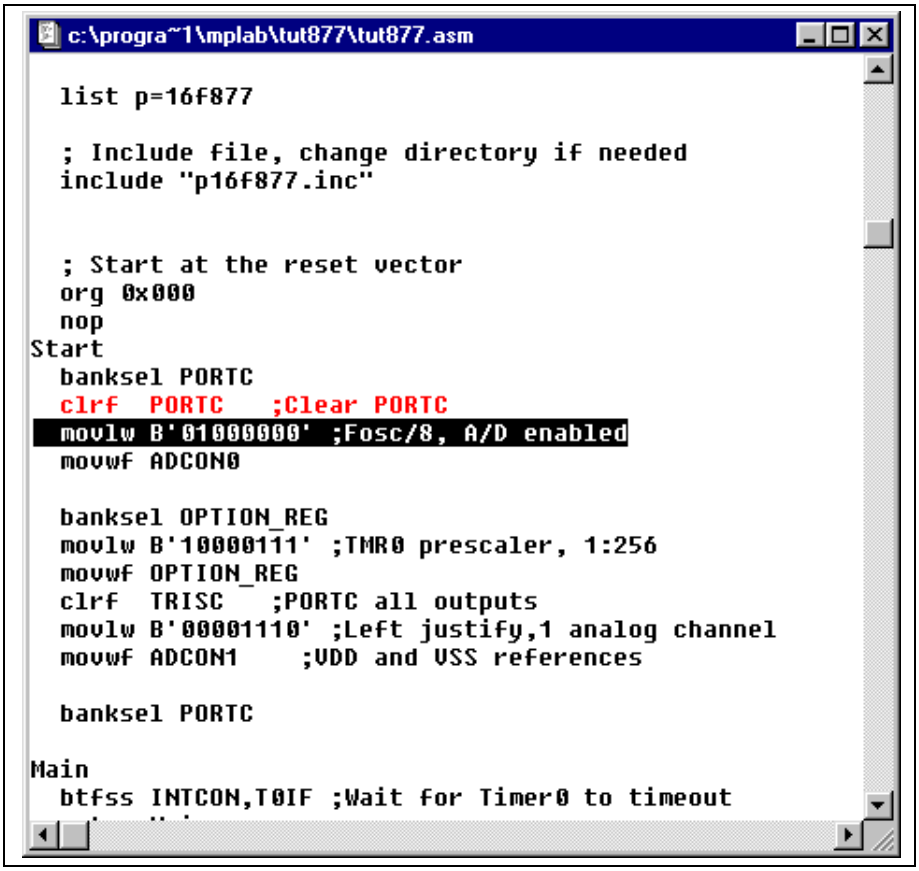


Figure 3.16: Program Halted After Break

Now single step by clicking on the Step toolbar button or issuing the *Debug > Run > Step* command. Single step twice and then examine the values of the registers ADCON0 and ADCON1 in the watch window. You should notice that ADCON0 has a value of 40 hex as shown in Figure 3.17.

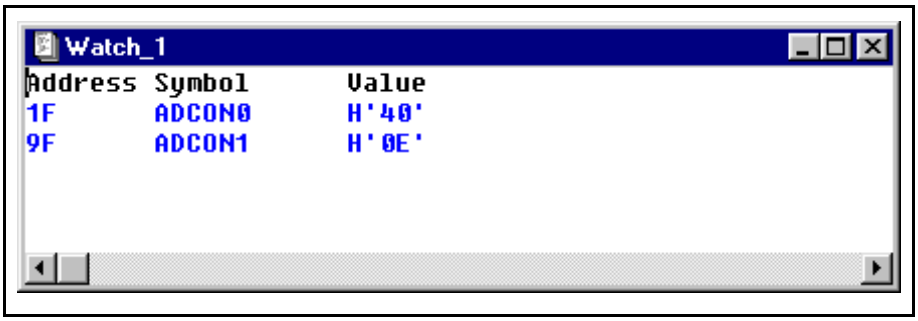


Figure 3.17: Updated Watch Window

This corresponds to the binary value designated in the program, but is this value correct? On examining the *PIC16F87X Data Sheet* (DS30292) section on A/D, you will see that the last bit should be a one, not a zero, to turn the A/D module on.

To fix this bug, change:

```
movlw B'01000000' ;Fosc/8, A/D enabled
```

to

```
movlw B'01000001' ;Fosc/8, A/D enabled
```

Select *File > Save* to save your changes and select *Project > Make Project* to rebuild the project.

A message will tell you the program has been rebuilt and you must reprogram the ICD in order for your changes to take effect. Click the **Program** button in the MPLAB-ICD dialog to reprogram the ICD to reflect your change.

When the MPLAB-ICD dialog's Status box indicates that it is waiting for your next command, you are ready to run your program again.

Click the **Run** toolbar button or issue the *Debug > Run > Run* command to run the program in real-time mode. Some of the LEDs should now be lit. Turn the potentiometer (RA0) to change the value displayed on the LEDs.

The source code in this tutorial contained only one bug. However, real code may have more. Using the MPLAB-ICD and MPLAB IDE debugging functions, you can successfully find and fix the bugs in your code.

3.11 Tut877 Main Routine

The main routine of `tut877.asm` (Figure 3.18) begins by configuring PORTC, the A/D module and Timer0. It then waits for a Timer0 overflow to start the A/D conversion of the value from the potentiometer. When the conversion is complete, the value is displayed on the LEDs, and the program loops back to wait for another Timer0 overflow to start another A/D conversion.

For more information on A/D module operation, please refer to the *PICmicro Midrange Microcontroller Family Reference Manual* (DS33023) for an operational description and a list of related application notes.

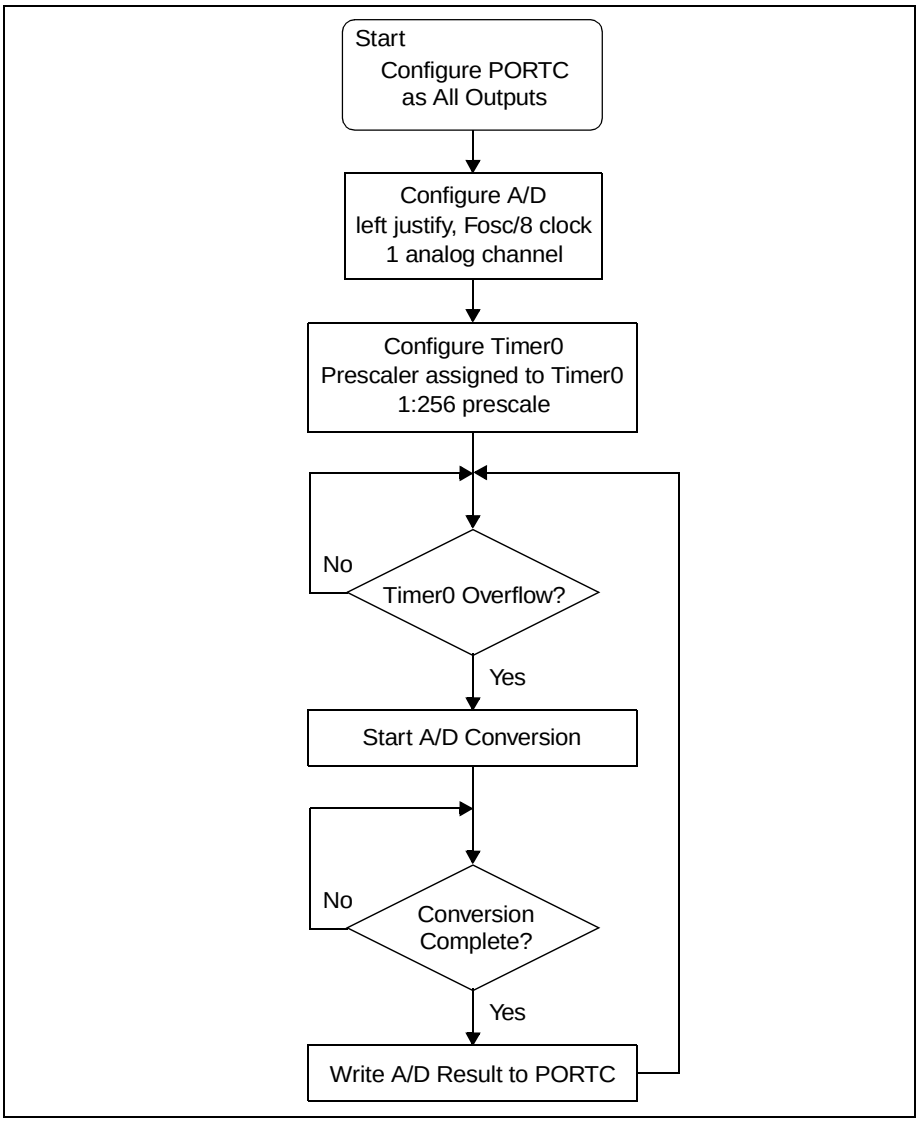


Figure 3.18: Program Flow Chart

3.12 Tut877 Corrected Source Code

This is a functional version of `tut877.asm`.

```
;*********************************************************
;*  TUT877.ASM
;*********************************************************
;*  Microchip Technology Incorporated
;*  16 December 1998
;*  Assembled with MPASM V2.20
;*********************************************************
;*  This program configures the A/D Module to convert on
;*  A/D channel 0 (the potentiometer) and display the
;*  results on the LEDS on PORTC.  Make sure that the DIP
;*  switch SW3 has all switches in the ON position.
;*********************************************************

     list p=16f877

     ; Include file, change directory if needed
     include "p16f877.inc"


     ; Start at the reset vector
     org     0x000
     nop
Start
     banksel   PORTC
     clrf      PORTC          ;Clear PORTC
     movlw     B'01000001'    ;Fosc/8, A/D enabled
     movwf     ADCON0

     banksel   OPTION_REG
     movlw     B'10000111'    ;TMR0 prescaler, 1:256
     movwf     OPTION_REG
     clrf      TRISC          ;PORTC all outputs
     movlw     B'00001110'    ;Left justify,1 analog channel
     movwf     ADCON1         ;VDD and VSS references

     banksel   PORTC

Main
     btfss     INTCON,T0IF    ;Wait for Timer0 to timeout
     goto      Main
     bcf       INTCON,T0IF

     bsf       ADCON0,GO      ;Start A/D conversion
Wait
```

```
     btfss     PIR1,ADIF      ;Wait for conversion to complete
     goto      Wait

     movf      ADRESH,W       ;Write A/D result to PORTC
     movwf     PORTC          ;LEDs

     clrf      PORTC
WaitPush
     btfss     PORTB,0
     goto      WaitPush

     movwf     PORTC
     goto      Main           ;Do it again

     end
```

Chapter 4. Getting Started with MPLAB-ICD

4.1 Introduction

This chapter describes how to get the hardware and software for MPLAB-ICD up and working.

4.2 Highlights

Topics covered in this chapter:

- Communicating with MPLAB-ICD
- MPLAB-ICD Operations
- Getting the Most from MPLAB IDE – Using Projects

4.3 Communicating with MPLAB-ICD

Follow the steps listed below to set up MPLAB IDE for use with the MPLAB-ICD hardware.

1. Make sure the MPLAB-ICD is connected to the host PC via the RS-232 cable as described in Section 2.5.
2. Make sure power is provided to the target application from which the MPLAB-ICD module will be drawing power.
3. From the Microchip MPLAB IDE program group, run MPLAB IDE.
4. Select *Options > Development Mode* and click the **Tools** tab to open the Development Mode dialog.
5. Select the MPLAB-ICD development mode. Select the PIC16F87X processor you are going to debug.

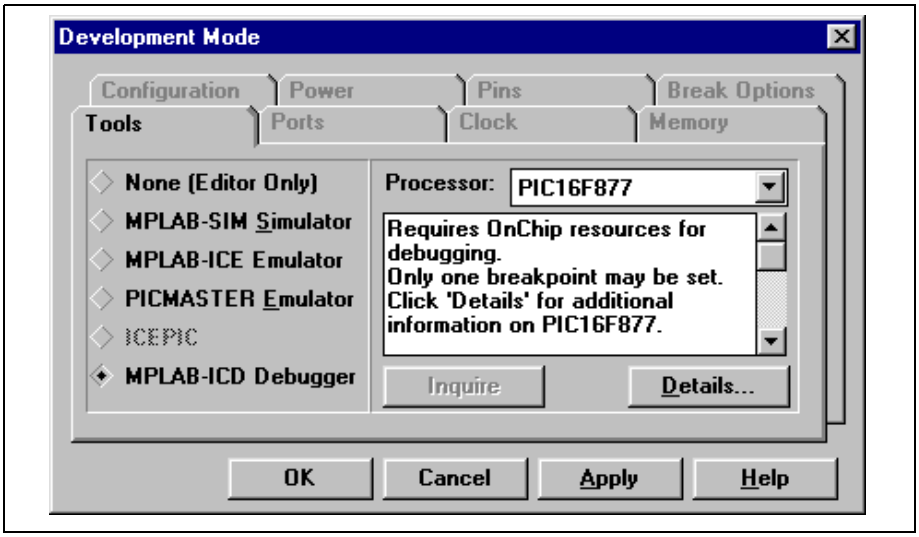


Figure 4.1: Development Mode Dialog Box

6. Click **OK**.

The Development Mode dialog will close and the MPLAB-ICD dialog will open.

Note: If you close the MPLAB-ICD dialog, select *Options > Development Mode* and click the **Tools** tab to open the Development Mode dialog. Select MPLAB-ICD Debugger and click **OK**. The MPLAB-ICD dialog will re-open.

If MPLAB IDE was previously in ICD mode, MPLAB IDE will either find the MPLAB-ICD module and start, or it will display a message that it cannot find the MPLAB-ICD. If MPLAB IDE cannot find the MPLAB-ICD, select the Editor Only option from the dialog and click **Apply**. Then, repeat steps 4 – 6.

See Chapter 5 for more information on the MPLAB-ICD dialogs.

If MPLAB IDE still cannot find MPLAB-ICD, see Chapter 7.

4.4 MPLAB-ICD Operations

The MPLAB-ICD is a programmer for the PIC16F87X family as well as an in-circuit debugger. It programs hex files into the PIC16F87X and offers basic debugging features like real-time code execution, stepping, and breakpoints. Its debug feature is built inside the PIC16F87X and activated by programming the Debug Code into the target processor. It has limited functions when compared to a full-featured in-circuit emulator but provides cost-effective functions to debug and program applications for a PIC16F87X.

To enable in-circuit debugging, the Debug Code residing in the microcontroller in the MPLAB-ICD module is programmed into the target PIC16F87X. The code is an MPASM module that will be programmed into the PIC16F87X on the MPLAB-ICD header automatically by MPLAB. This code will reside at the end of program memory. For example, it will reside in 0x1F00 to 0x1FFF of the PIC16F877 processor (see Section 1.5).

4.5 Getting the Most from MPLAB IDE – Using Projects

MPLAB IDE is the host software for the MPLAB-ICD and the MPLAB-SIM simulator. It functions as a sophisticated debugging tool, providing access to RAM, ROM, EEPROM, and a variety of other debug functions.

Note: If you do not put your source files into a project, MPLAB IDE cannot debug properly.

4.5.1 MPLAB IDE Project Features

Developing and debugging code in the MPLAB IDE environment is based on projects. Although emulation can be performed without having a project open, projects have the following advantages:

- Single or multiple source files can be easily built and maintained.
- Symbolic debugging is available.
- The debugging environment can be saved for later use.

Some of the information that is retained with a project is:

- Development mode and processor
- Source files associated with the project
- Name of the final PICmicro executable file
- Open windows and their sizes and positions
- Named break settings
- Configuration bit settings

4.5.2 Creating a Project

Select *Project > New Project* to open the Edit Project dialog and create a project. The project will contain information about your source, object, and other files as well as a variety of important project settings.

4.5.3 Saving a Project

To save your current project to retain the values for later use or to use as a backup or default as you continue with your debugging, click **OK** in the Edit Project dialog. Then select *Project > Save Project* to save your project.

Development Mode

The selected development mode is retained with the project information. To change the development mode for a project, follow these steps:

- Open the project. The previously used development mode will be selected.
- Change the development mode by selecting *Options > Development Mode* to access the Development Mode dialog. Select the development mode and click **OK**.
- Select *Project > Save Project* to save the project.

For more information on creating and using projects, refer to the *MPLAB IDE User's Guide* (DS51025).

Chapter 5. MPLAB-ICD Basic Functions

5.1 Introduction

This chapter discusses the basic operations of MPLAB-ICD and MPLAB IDE debugging functions of the In-Circuit Debugger. For more information on general debugging features, refer to the *MPLAB IDE User's Guide*.

5.2 Highlights

Topics covered in this chapter:

- The MPLAB-ICD Dialogs
- Program Execution
- Breakpoints
- How to use MPLAB-ICD

5.3 The MPLAB-ICD Dialogs

Once you have set up MPLAB IDE to work with the MPLAB-ICD, you can set up the MPLAB-ICD for programming and debugging.

5.3.1 MPLAB-ICD Dialog

The MPLAB-ICD dialog is always open when the ICD is enabled. You can minimize or move the MPLAB-ICD dialog, but do not close it. Closing this dialog disables the debugger. You must then select *Options > Development Mode*, select MPLAB-ICD, and click **OK** to reenable the debugger.

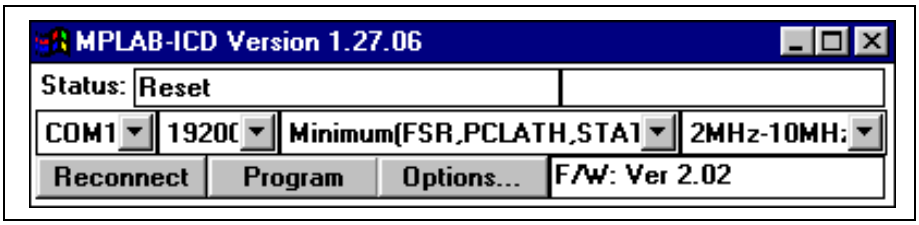


Figure 5.1: MPLAB-ICD Dialog

The MPLAB-ICD dialog has the following settings.

Table 5.1: MPLAB-ICD Dialog

Item	Options
Status	The Status bar displays the executed MPLAB-ICD function and the status.
COM Port	Select the COM port (COM1, COM2, COM3, or COM4) for MPLAB-ICD communications.
Baud Rate	Select the COM port baud rate for MPLAB-ICD communications.
Upload Options	The options for the amount of data uploaded are: • Minimum (FSR, W, Status, PCLATH) (Very Fast) • SFRs only (Fast) • Minimum and Watch windows (Pretty Fast) • All Registers (Slow) If you are using breakpoints and single step, you can upload selected data to improve speed. The first selection has less than a second delay, where as the last selection will have a delay of approximately 2 seconds. Data will be uploaded on a single step, a breakpoint, or a halt.

Note: To save these settings, be sure to click the "X" in the upper right corner of the MPLAB-ICD dialog before you exit the debugger or close your project.

Table 5.1: MPLAB-ICD Dialog (Continued)

Item	Options
Operating Frequency Range	Select the operation frequency range of the MPLAB-ICD. Options are: • 32 kHz – 500 kHz • 500 kHz – 2 MHz • 2 MHz – 10 MHz • 10 MHz – 20 MHz
Reconnect	If you had to change the COM port or baud rate, click Reconnect to reestablish communications using the new values.

Click **Options** in the MPLAB-ICD dialog to open the ICD Options dialog.

5.3.2 ICD Options Dialog

The ICD Options dialog is accessed by clicking **Options** in the MPLAB-ICD dialog. The ICD Options dialog can be closed or minimized without disabling the ICD.

The ICD Options dialog (Figure 5.2) has all the programming and setup functions for the MPLAB-ICD. These functions include basic programmer functions like blank check, program, read, verify, erase, and debug mode.

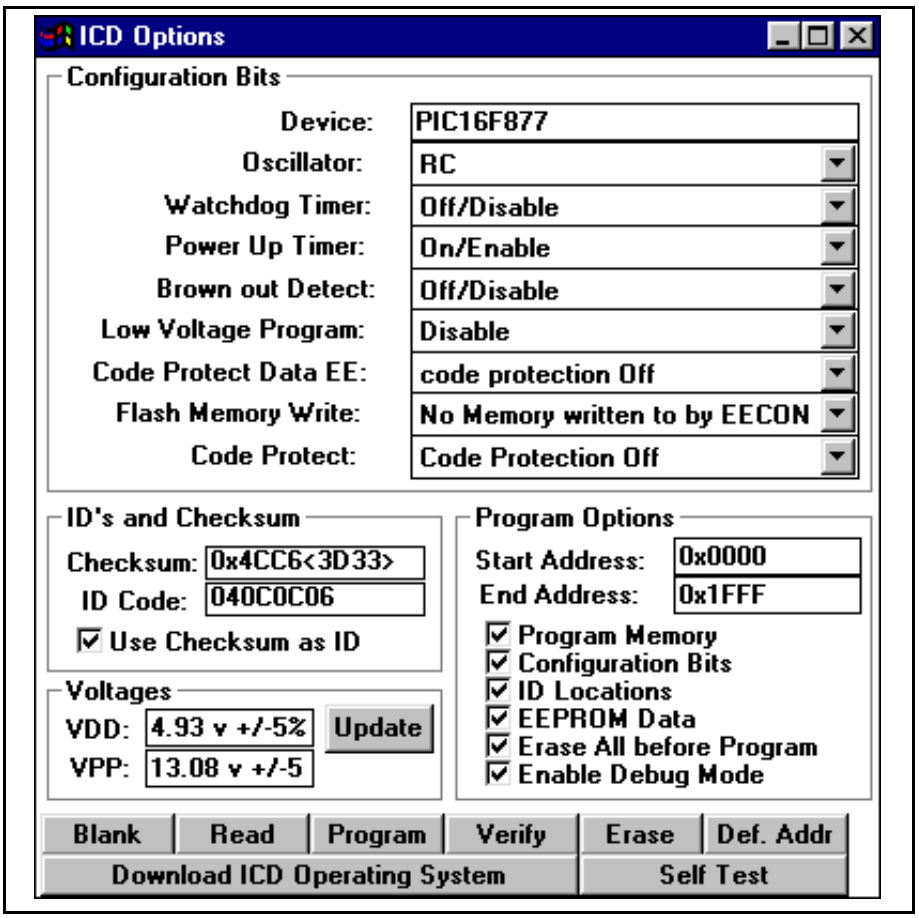


Figure 5.2: MPLAB-ICD Options Dialog

The ICD Options dialog has the following settings.

5.3.3 Configuration Bits and Device Selection

This section of the dialog allows you to set the various configuration bits on the PIC16F87X processors. To select, click on the arrow to see the list of choices and then select the correct item.

Table 5.2: Configuration Bits and Device Selection Options

Item	Options
Device	This box shows the processor you selected in the Development Mode dialog. If you change the device here, any open MPLAB IDE projects will close (except in Editor Only mode) and program memory, configuration bits, and IDs will be cleared.
Oscillator	RC, LP, XT or HP
Watchdog Timer	Select On/Enable or Off/Disable. Usually should be disabled for debugging.
Power Up Timer	Select On/Enable or Off/Disable. Usually should disabled for debugging.
Brown Out Detect	Select On/Enable or Off/Disable. Must be disabled for debugging.
Low Voltage Program	Select low voltage programming function or use RB3 as digital I/O and use HV on MCLR for programming. Low voltage ICSP programming must be disabled for debugging.
Code Protect Data EE	Select On/Enable or Off/Disable. Must be disabled for debugging.
Flash Memory Write	Select whether to allow unprotected program memory to be written by EECON control. Must be disabled for debugging.
Code Protect	Select the desired range. Must be disabled for debugging.

5.3.4 IDs and Checksum

Table 5.3: IDs and Checksum Selection Options

Item	Options
Checksum	Displays the checksum for the data.
ID Code	Displays the ID code.
Use Checksum as ID	To use the checksum as the ID, select the checkbox.

5.3.5 Voltages

Table 5.4: Voltage Selection Options

Item	Options
VDD	Displays the current value of the VDD voltage. MPLAB-ICD develops its needed VPP ≈ 12V from the target board's VDD through use of a charge pump.
VPP	Displays the current value of the VPP voltage.
Update	Allows you to check the current value of the VDD and VPP voltages on the target application.

5.3.6 Program Options

Select or clear the checkbox next to the item to select or deselect the memory areas for the programmer functions. For example, if you want to read program memory only, select Program Memory and clear the checkboxes for the other memory types.

Table 5.5: Program Selection Options

Item	Options
Start Address, End Address	The starting and ending address range in program memory for programming, reading, or verification. **Note:** The address range does not apply to the Erase function. The Erase function will erase all data on the PIC16F87X. The default program address range is set to the maximum program memory available based on the device you selected. To identify file register and program memory locations that ICD uses in the target device, see Section 1.5.
Configuration Bits	Program the configuration bits. You can set the configuration bits in the top portion of the ICD Options dialog.
ID Locations	Program the ID locations. You can set the ID locations in the IDs and Checksum area of the ICD Options dialog.
EEPROM Data	For devices with data EEPROM, program the data memory from the data in the EEPROM Memory window.
Erase All before Program	Select this option to erase all program memory before you click **Program**.

Table 5.5: Program Selection Options (Continued)

Item	Options
Enable Debug Mode	Select this option to program debug code into a device. The debugging operations will be enabled each time you click **Program**. If you are only using MPLAB-ICD to program a part, clear the Enable Debug Mode check box. The debug code will not be downloaded to program memory and MPLAB-ICD will not be enabled for debug operations.

5.3.7 Function Buttons

Click the function buttons to perform the assigned function on the PIC16F87X in the MPLAB-ICD header. If you have specified an address range in the Program Options section, the assigned function will be performed on the specified memory range and type only (except for the erase function).

Table 5.6: Function Button Selection Options

Item	Options
Blank	Checks to see if device is blank.
Read	Reads memory areas specified under Program Options: program memory, configuration bits, ID locations, and/or EEPROM data.
Program	Programs memory areas specified under Program Options: program memory, configuration bits, ID locations, and/or EEPROM data. Also downloads debug code in program memory if Enabled Debug Mode is checked.
Verify	Verifies programming of memory areas specified under Program Options: program memory, configuration bits, ID locations, and/or EEPROM data.
Erase	Erases all data on the PIC16F87X including memory, ID, and configuration bits.
Download Operating System	Downloads an updated version of the MPLAB-ICD firmware (operating system) to the MPLAB-ICD's FLASH processor.
Self Test	Performs a self-test on the MPLAB-ICD hardware.
Def.Addr	Retores the Start and End addresses to the defaults fo the device. End addresses with Debug Mode enabled: • PIC16F873/874:0xEDF • PIC16F876/877:0x1EFF End addreses with Debug Mode disabled: • PIC16F873:0xFFF • PIC16F876/877:0x1FFF

5.4 Program Execution

The MPLAB-ICD executes in real-time mode or in step mode.

- Real-time execution occurs when the PIC16F87X in the MPLAB-ICD header or demo/application board is put in MPLAB's Run mode.
- Step mode execution can be accessed after the processor is halted.

5.4.1 Real-Time Execution

When the MPLAB-ICD is run in real time, instructions execute just as the processor would without the debugger. The PIC16F87X executes in real time until a breakpoint halts the debugger or until the HALT function is manually executed.

To execute in real time, click the Run toolbar button or issue the *Debug > Run > Run* command. The Debug toolbar provides Run, Halt, and Step buttons for controlling the debugger. While in the run mode, register displays on the screen will not update.

5.4.2 Step Mode Execution

Step Mode Execution occurs when you single step the processor or execute *Debug > Run > Step*. Step mode execution allows you to step through the code one instruction at a time, watch the program flow, and see the register contents at each instruction (as set in the dialog box).

Debug > Run > Animate automatically single steps the processor until you halt it. To view the changing registers in the Special Function Register window or the Watch windows, use Animate. Animate runs slower than the Run function.

5.5 Breakpoints

A breakpoint is a state where the processor halts after a certain condition is met. The MPLAB-ICD provides the followings ways to set a breakpoint:

- Break on Address Match
- Break on User Halt

5.5.1 Break on Address Match

The Debug function of the PIC16F87X allows one breakpoint to be set. This breakpoint can be at any program memory address location. The processor breaks after the instruction is executed. For example, if a breakpoint is set at address 005Ah, the processor breaks after executing the instruction at address 005Ah.

To set a breakpoint, select *Debug > Break Settings*. You can also set a breakpoint by selecting the source code address in Program Memory or selecting the line of code in the source code window, clicking the right mouse button to access a shortcut menu, and selecting *Break Point(s)* from the shortcut menu.

5.5.2 Break on User Halt

The PIC16F87X executes until you click the Halt button or select *Debug > Run > Halt.*

5.6 How to use MPLAB-ICD

The MPLAB-ICD is a programmer as well as an in-circuit debugger. The typical functions that would be used to debug and test an application are discussed in the following sections.

5.6.1 Load the Hex File for Debugging

Load the hex code to MPLAB IDE using the *File > Import > Import to Memory* command. To identify file register and program memory locations that ICD uses in the target device, see Section 1.5. Your code should be limited accordingly.

Note:	If you need to specify values for the EEPROM data memory in your source code, use a starting address of 0x2100.

Once the hex file is imported, you can use the Windows menu to open the following windows:

- Program Memory
- Special File Registers
- File Registers
- EEPROM Memory

5.6.2 Program the Target Processor

In the ICD Options dialog, check that you have chosen the correct device, oscillator, and configuration bit settings. Next, make sure the Program Options are correct for your application. If Enabled Debug Mode is selected, the debug code will be downloaded and the PIC16F87X will be enabled for debug operations.

Click the **Program** button to program the PIC16F87X in the MPLAB-ICD header or target/demo board. Programming will take a couple of minutes, longer if the target application is running at a low voltage.

Note:	The debug code is a special code at the end of program memory (see Section 1.5). It must be present in order for you to use the in-circuit debugging capabilities of the MPLAB-ICD.

After this, you can close the ICD Options dialog, but do not close the MPLAB-ICD dialog. Closing the MPLAB-ICD dialog with disable the debugger.

5.6.3 Debug Functions

You can use the following functions with MPLAB-ICD. You can only set one breakpoint at a time.

- **Run** – Real time program execution.
- **Reset** – Reset the PIC16F87X ($\overline{\text{MCLR}}$ Reset).
- **Break Settings** – Set breakpoints.
- **Halt** – Halt program execution.
- **Step** – Single step through program execution. After the program is halted, each selection of Step will execute one line of the program.
- **Animate** – Single step through program execution automatically. After the program is halted, select Animate to execute one line of the program at a time.
- **Change PC** – Change program counter.

5.6.4 Modify Functions

If you need to modify the program being debugged, simply assemble a new hex file and download it. You can also use the Modify dialog to edit program memory before downloading it into the 16F87x with the **Program** button. If you do this, the object code generated by your source code will not match the code in the PIC16F87X.

You can use *Window > Modify* to change a register, words of program memory, or data in the EEPROM area. Select *Window > File Registers* and *Window > Special Function Registers* to view internal data registers.

Chapter 6. MPLAB-ICD Menu Options

6.1 Introduction

This chapter gives detailed information on using the MPLAB IDE menu options directly applicable to the ICD Debugger.

6.2 Highlights

Topics covered in this chapter:

- File Menu
- Debug Menu
- Options Menu

6.3 File Menu

Instead of using MPLAB IDE projects, you can use the *File > Import > Import to Memory* command to load the hex file to MPLAB-ICD. Once loaded into memory, you can select items from the Windows menu to view program memory, special file registers, file registers, and EEPROM.

6.4 Debug Menu

The following debug functions are available to control the MPLAB-ICD.

6.4.1 Run

Run options allow you to run your program in real-time or single-step mode. See Section 5.6.3 for a brief discussion of each run option.

6.4.2 Breakpoint Settings

The MPLAB-ICD debug function supports a single breakpoint at a time. Use the *Debug > Breakpoint Settings* command or press <F2> to open the Breakpoint Settings dialog. Enter the address location where the breakpoint is to be set. You can also select the location or line of source code where the breakpoint is to be set and use the right mouse button to set the breakpoint (see Section 5.5).

6.5 Options Menu

Select *Options > Development Mode* to select MPLAB-ICD and specify the processor to be debugged.

NOTES:

Chapter 7. Troubleshooting

7.1 Introduction

This section describes some common problems associated with running MPLAB-ICD and steps to follow to resolve those problems.

7.2 Highlights

Topic covered in this chapter:

- Common Problems

7.3 Common Problems

Communications cannot be established with MPLAB-ICD.

If you cannot establish communications with MPLAB-ICD, follow these steps:

1. Make sure there is power to the demo board/target application. MPLAB-ICD is powered by the demo board/target application. Also, if you are using the demo board, make sure the power supply has the correct rating (9.0V, 0.75A).
2. Check that the MPLAB-ICD header is plugged in to the demo board/target application correctly; e.g., all header pins are plugged into the socket and the header is correctly oriented. Also, make sure you are using the correct stand-off if you are using the demo board.
3. Check that the PIC16F87X is plugged in to the MPLAB-ICD header correctly; e.g., all pins are plugged into the socket and the PIC16F87X is correctly oriented.
4. Check that the connection between the MPLAB-ICD and the host computer, via the RS-232 cable, is secure.
5. Check that the connection between the MPLAB-ICD header and MPLAB-ICD module, via the 9-inch modular cable, is secure.
6. Check the settings in the MPLAB-ICD dialog. Make sure you selected the correct PICmicro, COM port, and baud rate for your application.

MPLAB IDE attempts to establish communication with the MPLAB-ICD upon enabling the debugger. If communication cannot be established, no programming or debugging can occur. An error message appears if the attempt to establish communication fails. If a communication attempt fails, try again after correcting the problem, or cancel.

SOLUTIONS:

- Make sure the power supply is connected, and the LED on the MPLAB-ICD module is on and not blinking. If the LED on the MPLAB-ICD module is blinking, reset the module by cycling the power to the application on and off. Then reselect the COM port for the MPLAB-ICD in the

MPLAB-ICD dialog box. The LED should stop blinking.

- Try connecting the MPLAB-ICD module to a different serial port. If your PC has a 25-pin serial port, you will need a 25 to 9 serial port adapter.
- Make sure that a COM port is properly set up exclusively for use by the debugger. Check the resources to ensure they are operating properly and that there are no conflicts with other devices. This commonly happens when you have a modem or other serial device that is improperly configured. Consult your Windows manual or other reference literature. You can try removing, reconfiguring, or disabling the conflicting device, but do so only if you are familiar with those procedures. See the steps below for Windows 3.1 or Windows 95.
- Some system errors are caused by driver and hardware incompatibility. See the steps below for Windows 95.
- If you have a COM port but MPLAB IDE will not let you select it (the option is grayed out) you may be able to assign the port manually by editing the MPLAB.ini file. Typically, this occurs if you have a gap in your COM port list (i.e., you have a COM1, COM2, and a COM4, but no COM3). In this case you may be able to fix it by opening MPLAB.ini (use FIND to locate this file) and editing the section called [MPLAB-ICD] so that the setting CommPort=1 is set to the port you want selected. This is just a work-around to a deeper problem in which Windows is incorrectly reporting port availability through the 16-bit driver.
- You must use the Microsoft Windows communications driver that is native to the version of Windows that you use. If you use Windows 3.10, look for the file COMM.DRV in your \WINDOWS\SYSTEM directory. That file MUST have a time of 3:10a. The time denotes the version. If you are using Windows for Workgroups, look for the same file as above, but the time stamp on that file should be 3:11a. If these files differ, you may need to re-install windows or install that file from another source. This problem is not likely to occur in Windows 95.
- Make sure you are not using a third party communications driver. Open your SYSTEM.INI file and look for the line in the [OPTIONS] section that reads

  COMM.DRV=COMM.DRV

  If this line reads differently you are using a different communications driver.

Windows 3.1:

A serial mouse will use a COM port, as will an external modem. An internal modem has its own COM port, so if you have a second COM port on your PC, set it so it won't conflict with either the mouse or the modem.

Windows 95:

Windows 95 requires special attention to setting up COM ports. If you suspect a driver - hardware incompatibility, try changing Flow Control to Hardware and/or turning off the FIFO for the serial port. In Windows 95 this is done in the Control Panel. Click the **System** Icon. Next, click the **Device Manager** tab, and click **Port Settings**. If necessary, expand the Ports selection by clicking the "+" sign next to it. Double-click the I/O port that MPLAB-ICD is connected to. This is where you can set flow control to Hardware. To turn off FIFO, click the **Advanced** button, deselect the Use FIFO box, and click **OK**.

If communications still cannot be established, contact Microchip Customer Support as described in *General Information*.

The MPLAB-ICD module returns a communications error on Reset, Single Step, Halt or Run.

When using the MPLAB-ICD Demo Board, make sure that the oscillator select jumper is correctly set to RC or oscillator and that the part is correctly programmed for this selection. If using an oscillator, make sure that there is one in the socket. This problem usually results in the part being programmed OK and debug mode set OK but when a reset, single step, halt or run command is executed, the ICD module returns a communications error.

Debug mode doesn't seem to be working.

1. Closing the MPLAB-ICD dialog will disable the ICD. If you have closed this dialog, you will need to re-enable MPLAB-ICD. Select *Options > Development Mode*, select MPLAB-ICD, and click **OK**.
2. Debug code must be programmed into the device before debug mode will work. Select **Enable Debug Mode** from the ICD Options dialog to program the device with debug code.

When single stepping, the program runs too slowly.

Check the upload options in the MPLAB-ICD dialog. If you are uploading a large amount of data; e.g., all memory, you can expect about a 5 second delay between steps. To reduce the delay, select an option that uploads less data.

When single stepping, the program runs too quickly OR some registers are not updated.

Check the upload options in the MPLAB-ICD dialog. If you are uploading a small amount of data, the time between steps will be less than if you were uploading all memory. If you are not getting all of the data uploaded that you expect, check that you have selected the correct type(s) of registers for upload.

When halting, single stepping, or stopping on a breakpoint, MPLAB IDE seems to lock up.

Check the upload options in the MPLAB-ICD dialog. If you are uploading a large amount of data; e.g., all memory, you can expect about a 5 second delay between steps. To reduce the delay, select an option that uploads less data.

The following I/O pins are not functioning correctly: RB3, RB6 or RB7.

These pins are reserved for programming/debugging. Refer to Section 1.5 for more information.

Appendix A. MPLAB-ICD Hardware

A.1 Introduction

The MPLAB-ICD hardware consists of the MPLAB-ICD module, header, and demo board.

A.2 MPLAB-ICD Module and Header

The MPLAB-ICD module and header are be used to perform the ICD functions on the demo board or in the target application.

A.2.1 Voltage and Current Specifications

The MPLAB-ICD header is powered by the target application, from a 3.0V to 5.5V source. The current ratings for the ICD in different operating modes and voltages are as follows:

Operating Mode	5V	3V
Debug	35 mA (max)	20 mA (max)
Run	40 mA	25 mA
Verify	60 mA	50 mA
Program	60 mA	50 mA

A.2.2 Silkscreens and Schematics

This section contains the silkscreen and schematic diagrams for the MPLAB-ICD module and header.

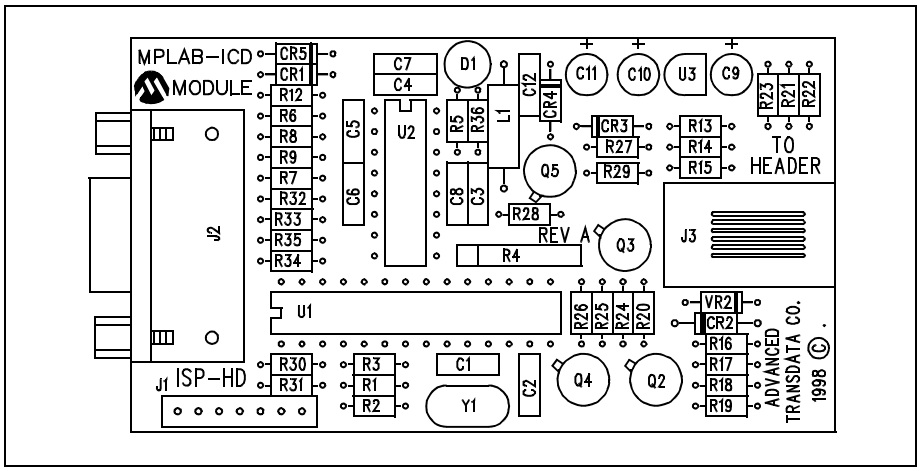


Figure A.1: MPLAB-ICD Module Silkscreen

Figure A.2: MPLAB-ICD Module Schematic, Part 1

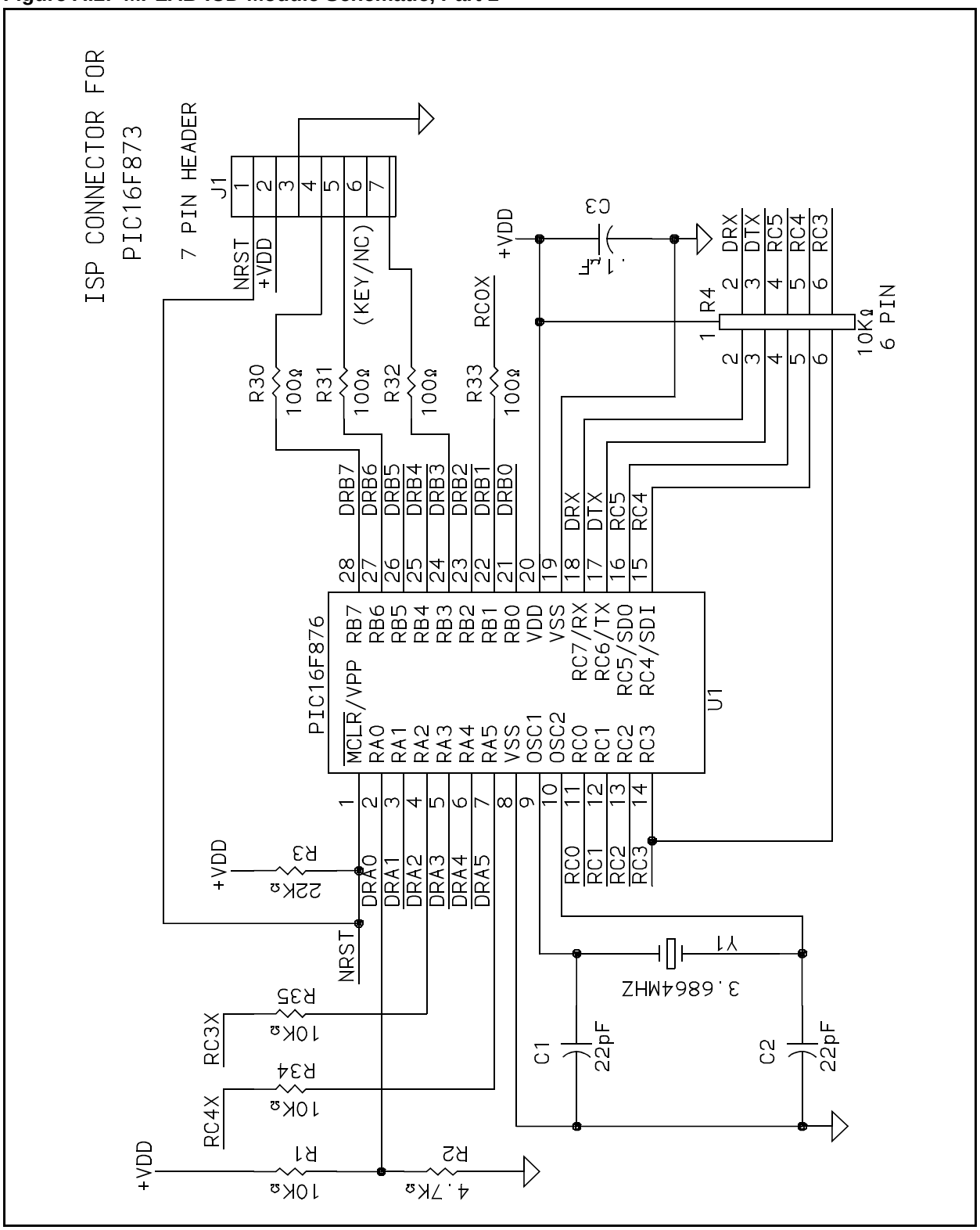


Figure A.3: MPLAB-ICD Module Schematic, Part 2

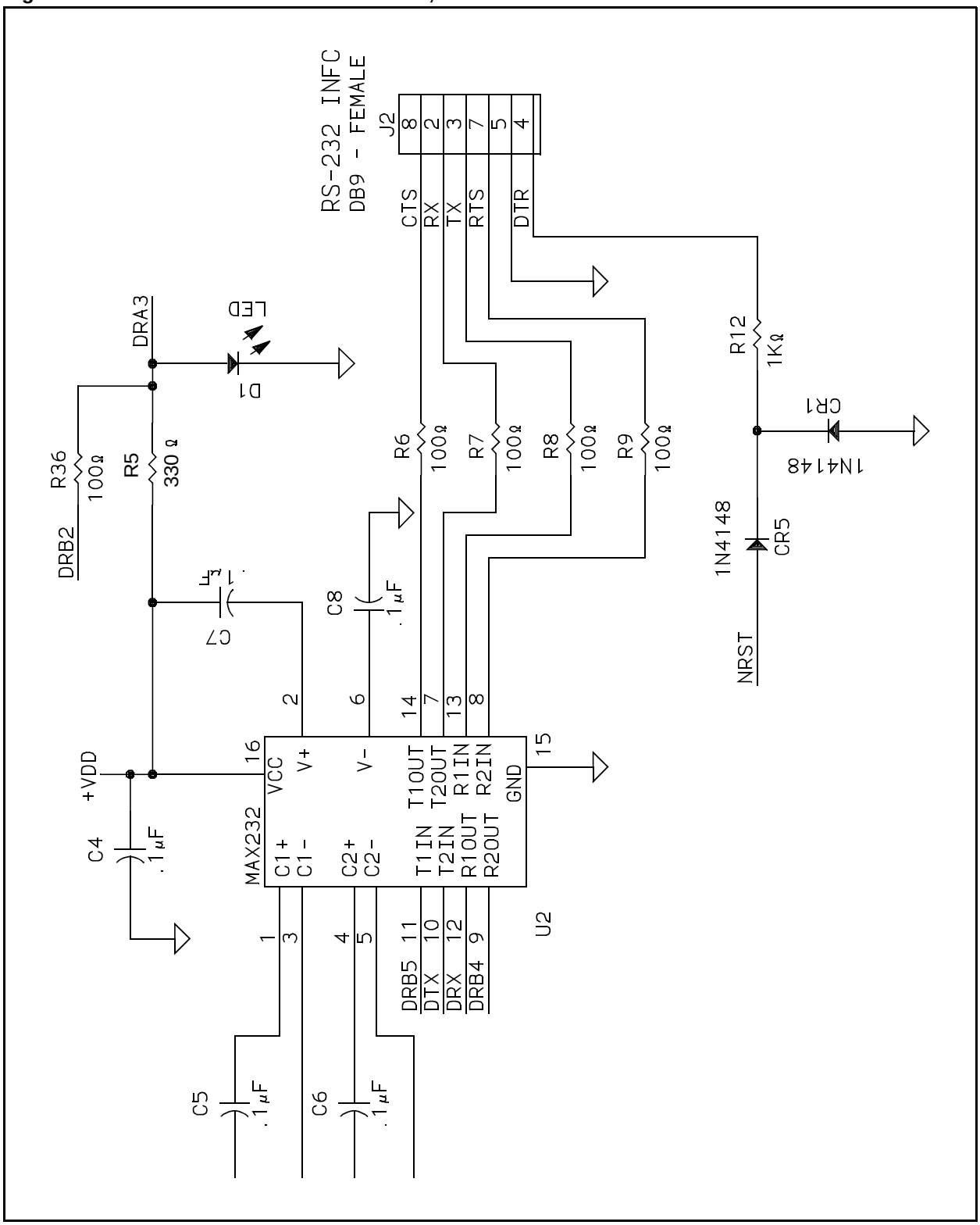


Figure A.4: MPLAB-ICD Module Schematic, Part 3

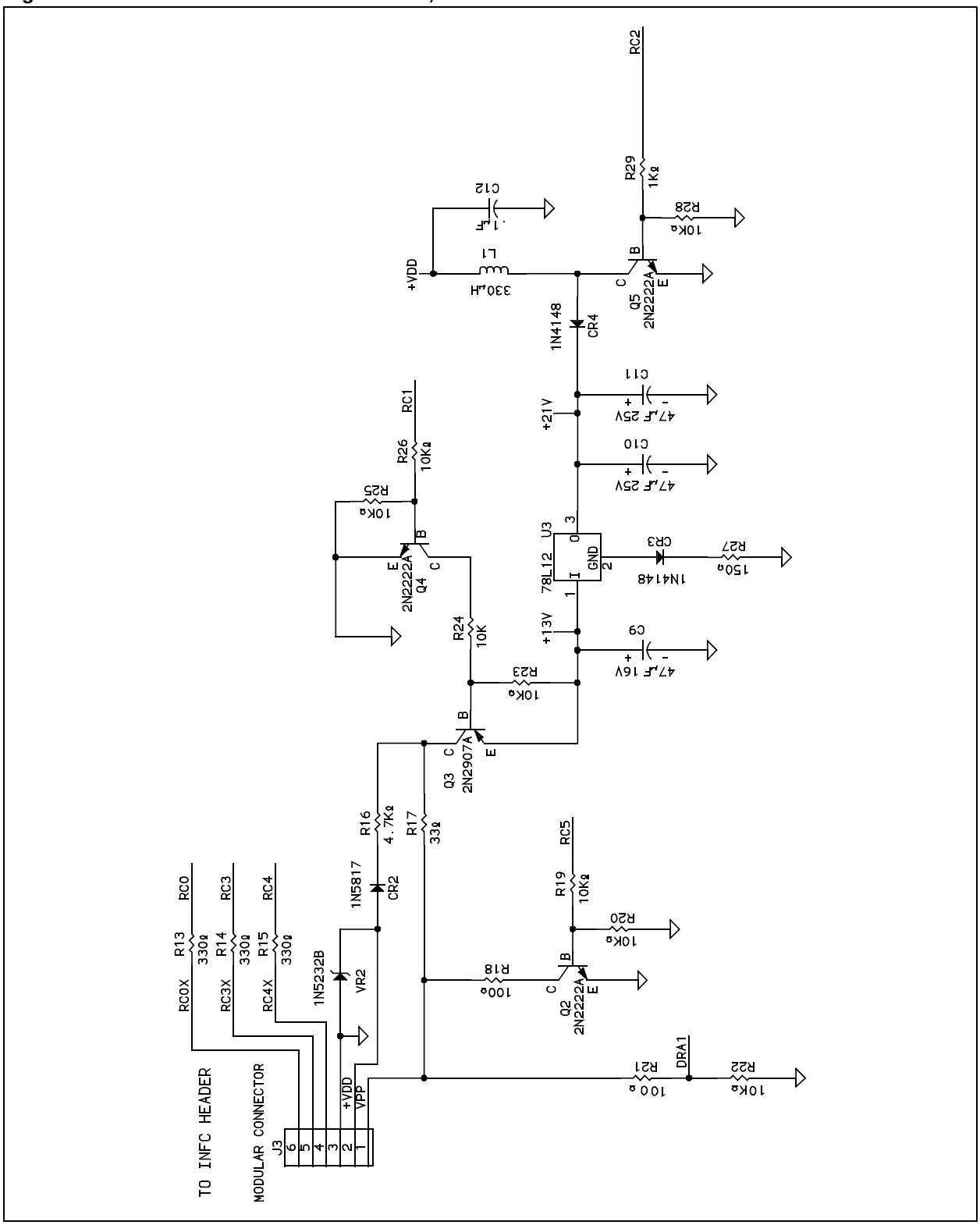


Figure A.5: MPLAB-ICD Header Silkscreen

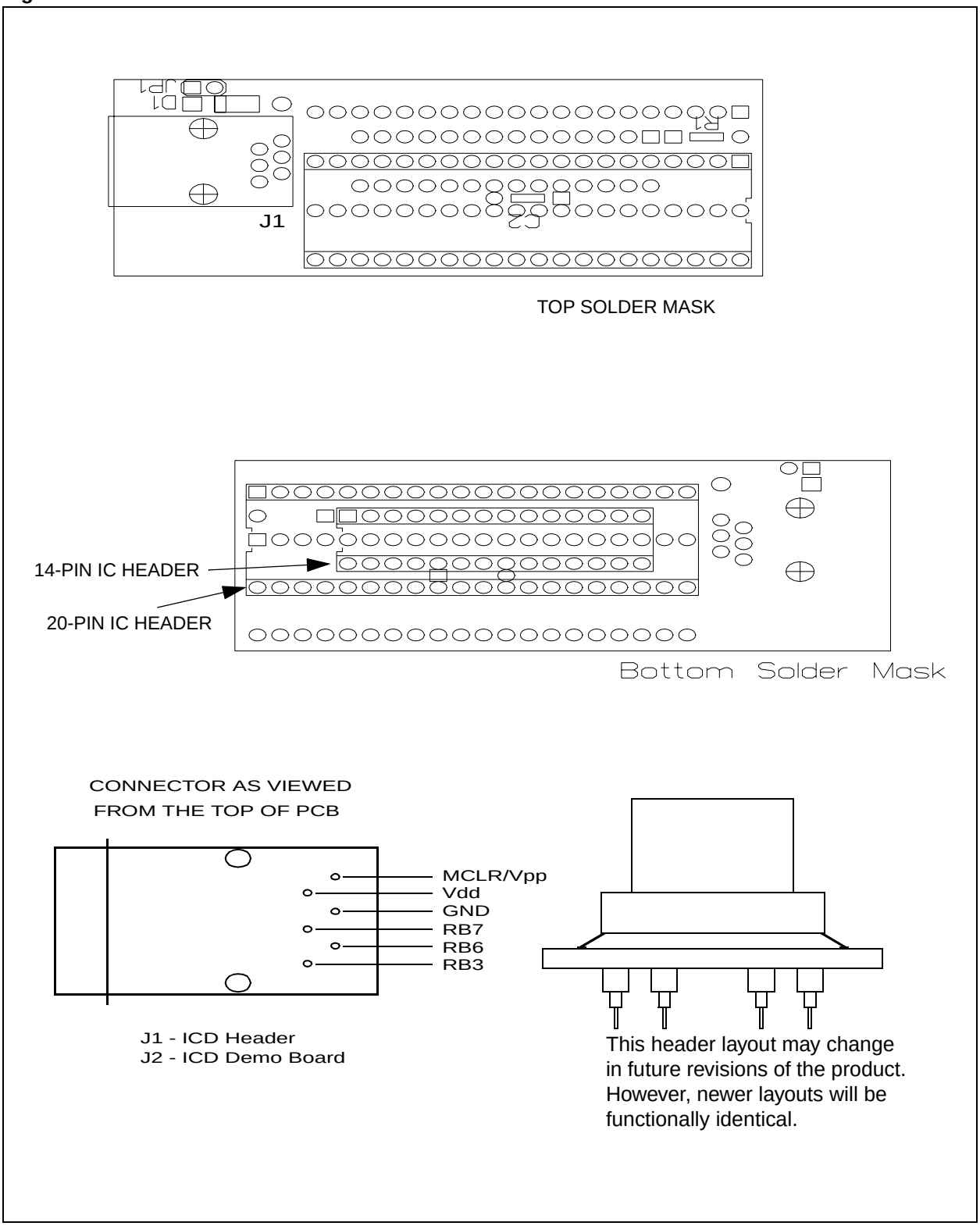
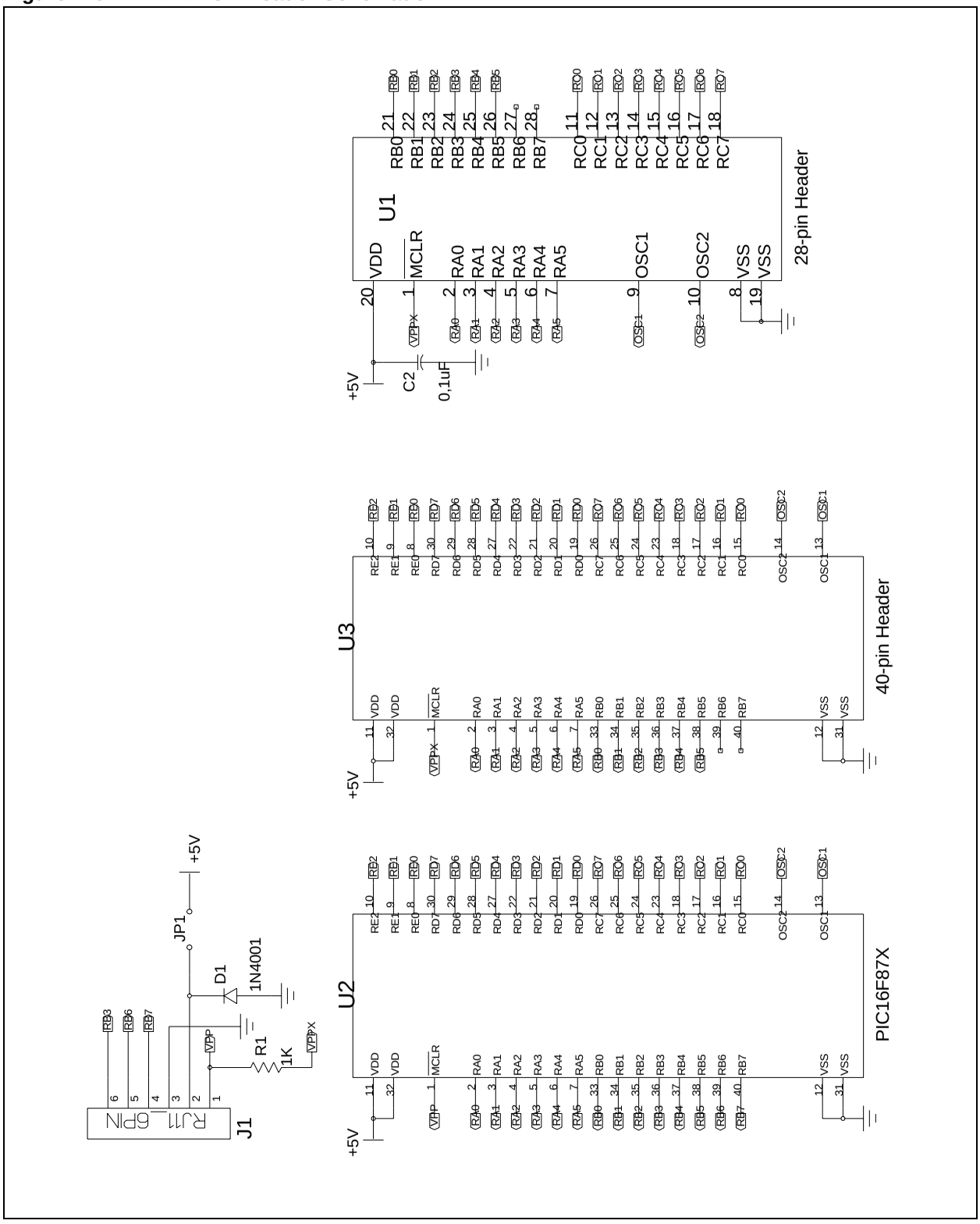


Figure A.6: MPLAB-ICD Header Schematic



A.3 MPLAB-ICD Demo Board

The MPLAB-ICD demo board is a simple demonstration board that supports PIC16F87X microcontrollers. The board can be used stand-alone with a pre-programmed part, or with the MPLAB-ICD module and header. A sample program is provided to demonstrate the unique features of the devices. Although a 40-pin and a 28-pin socket are provided to accommodate the different packages, only one processor can be run at a time.

A.3.1 Processor Sockets

Available sockets are:

- 40-pin socket for PIC16F871/874/877
- 28-pin socket for PIC16F870/872/873/876

Additional 40- and 28-pin sockets have been provided as stand-offs. Simply insert the desired socket onto the demo board and then insert the MPLAB-ICD header. Without the stand-off, the header would plug into both sockets, and not function properly. Or simply insert the device directly onto the demo board.

A.3.2 Display

Eight red LEDs are connected to PORTC of each processor type. The PORTC pins are set high to light the LEDs.

A.3.3 DIP Switches

Eight DIP Switches are provided in a package as SW3. When all switches are in the ON position, each of eight red LEDs is connected to a pin of PORTC.

A.3.4 Power Supply

The MPLAB-ICD module is powered by the target application's V_{DD} with a voltage range of 3.0V to 5.5V (and current value of 0.75A or larger). If you own a PICSTART Plus Programmer, you can use its power adapter to power the demo board.

A.3.5 ICD Connection

A modular cable connection next to the power supply can be used to connect the demo board directly to the MPLAB-ICD module.

A.3.6 RS-232 Serial Port

An RS-232 level shifting IC has been provided to support connection to an RS-232 host through the DB9 connector. The port is configured as DCE, and can be connected to a PC using a straight through cable, as opposed to a null modem cable. The circuitry must be populated by the user:

- U4 – Analog Devices MAX233A or equivalent
- R14-R19 – 330Ω, 1/8W resistor
- C7-C11 – 0.1 μF capacitors

A.3.7 Push-button Switches

Two push buttons provide the following functions:

- MCLR to reset the processor
- Active low switch connected to RB0

A.3.8 Oscillator Options

You can use the on-board RC oscillator circuit or plug an oscillator in the 4-pin socket. Make sure to set the jumper (JP1) to the proper selection.

- Socket provided for clock oscillator – use an oscillator from 32 kHz to 20 Mhz
- RC circuit – the frequency generated by the 4.7K resistor and 20 pF capacitor ranges from about 3.5 Mhz to 6 Mhz depending on the operating voltage and ambient temperature.

A.3.9 Analog Input

A 1K ohm potentiometer is connected through a series 470 ohm resistor (to protect the part should the pin be configured as an output) to RA0/AN0. The port can be adjusted from V_{DD} to GND to provide an analog input to the PIC16F87X parts.

A.3.10 Silkscreens and Schematics

This section contains the silkscreen and schematic diagrams for the MPLAB-ICD demo board.

Figure A.7: MPLAB-ICD Demo Board Silkscreen

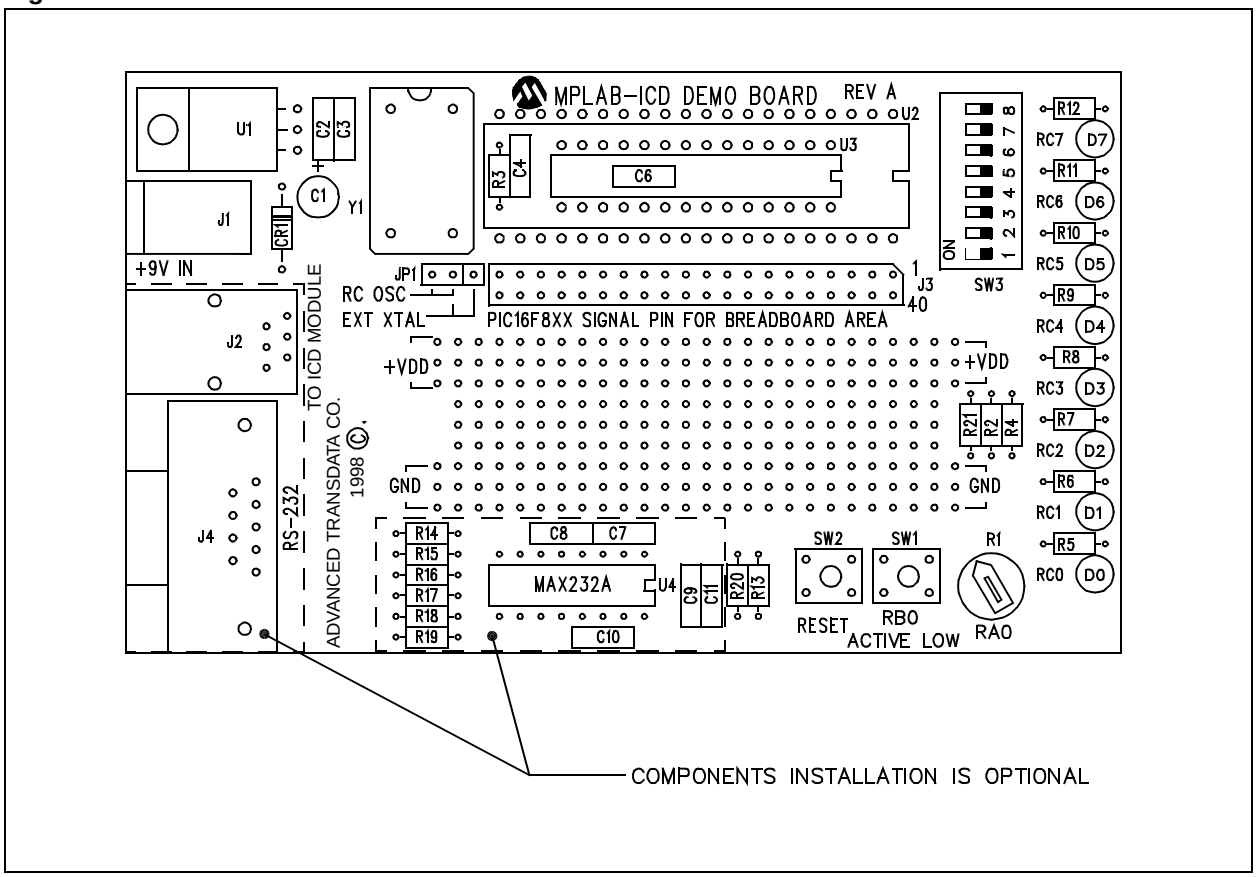


Figure A.8: MPLAB-ICD Demo Board Schematic, Part 1

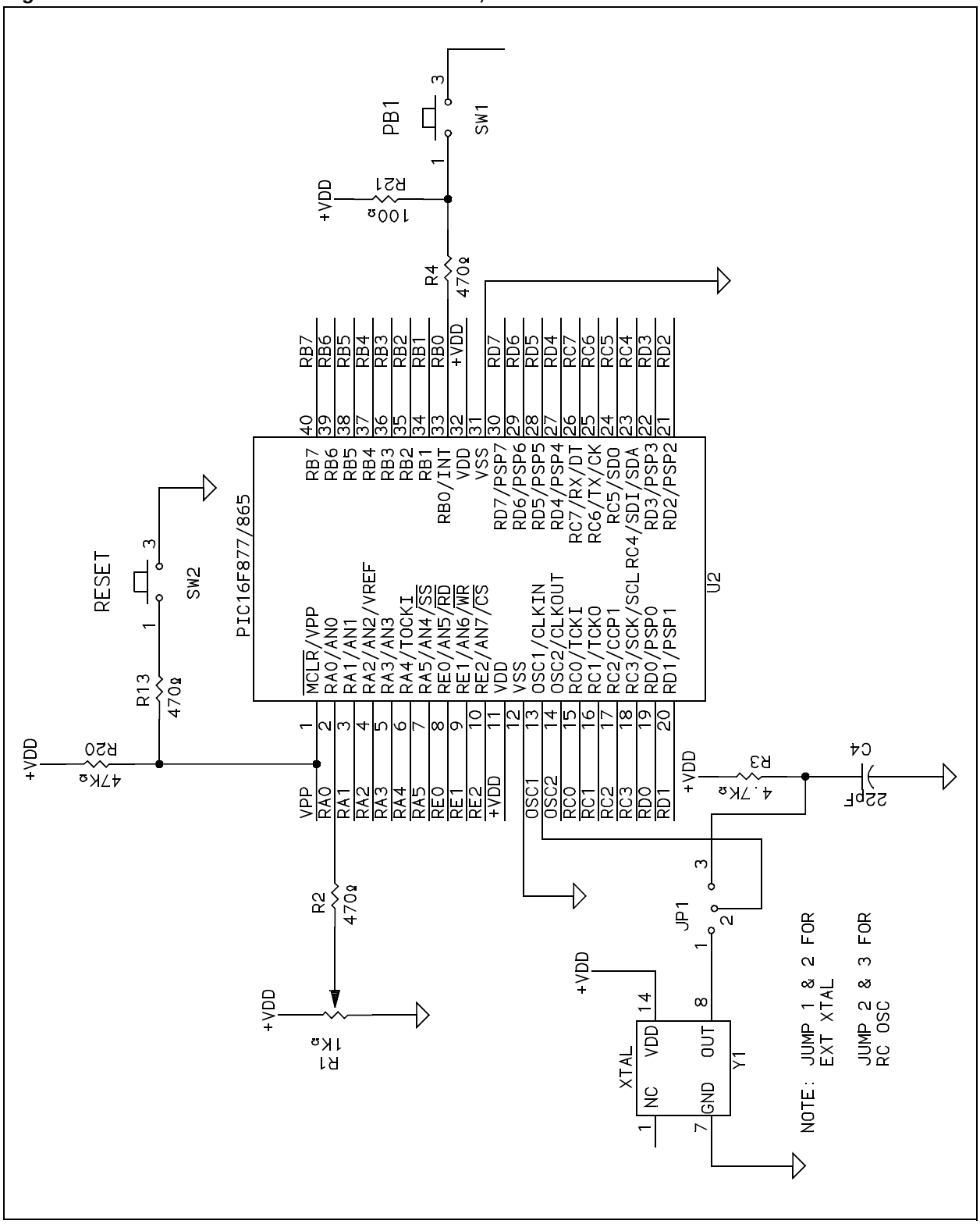


Figure A.9: MPLAB-ICD Demo Board Schematic, Part 2

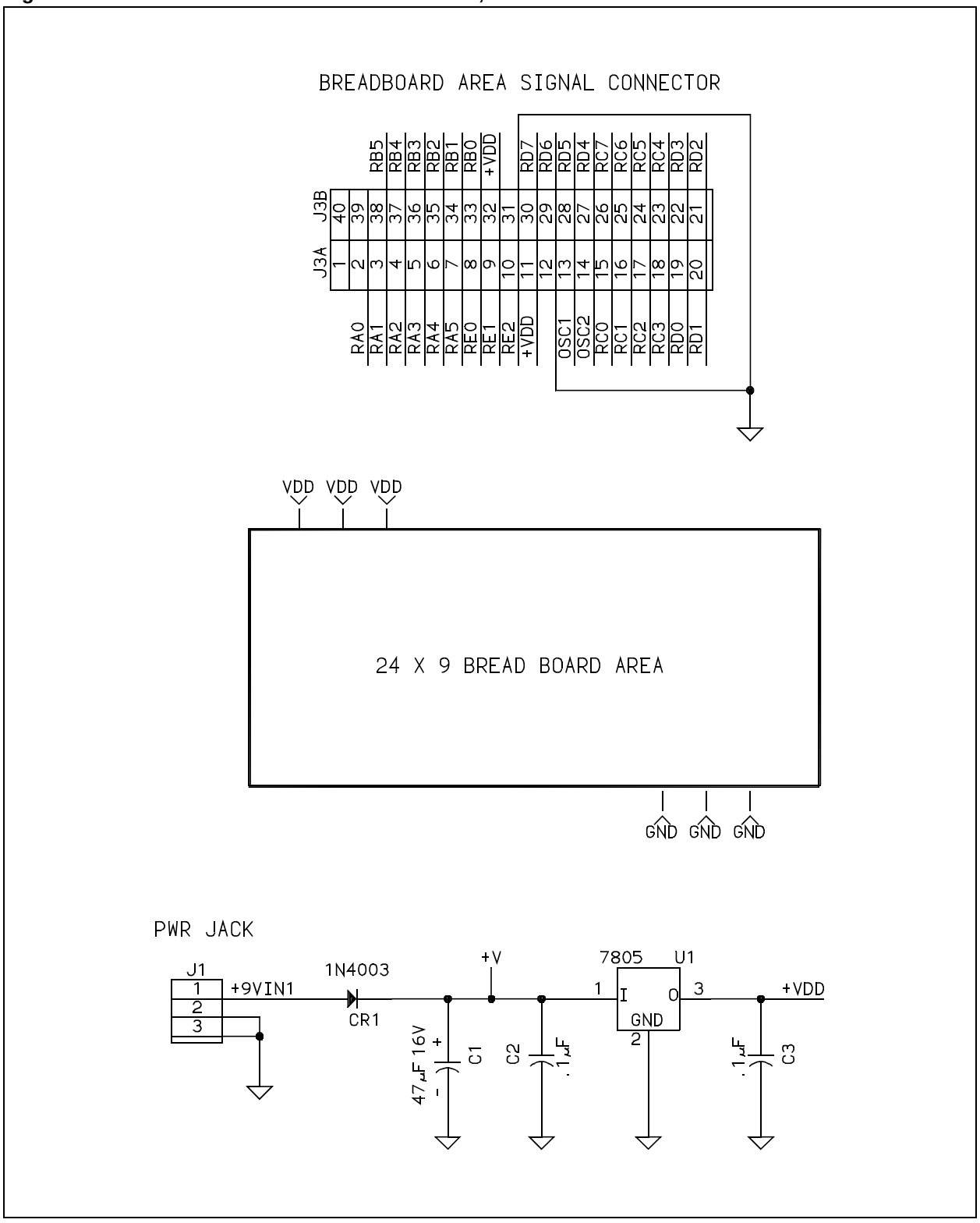


Figure A.10: MPLAB-ICD Demo Board Schematic, Part 3

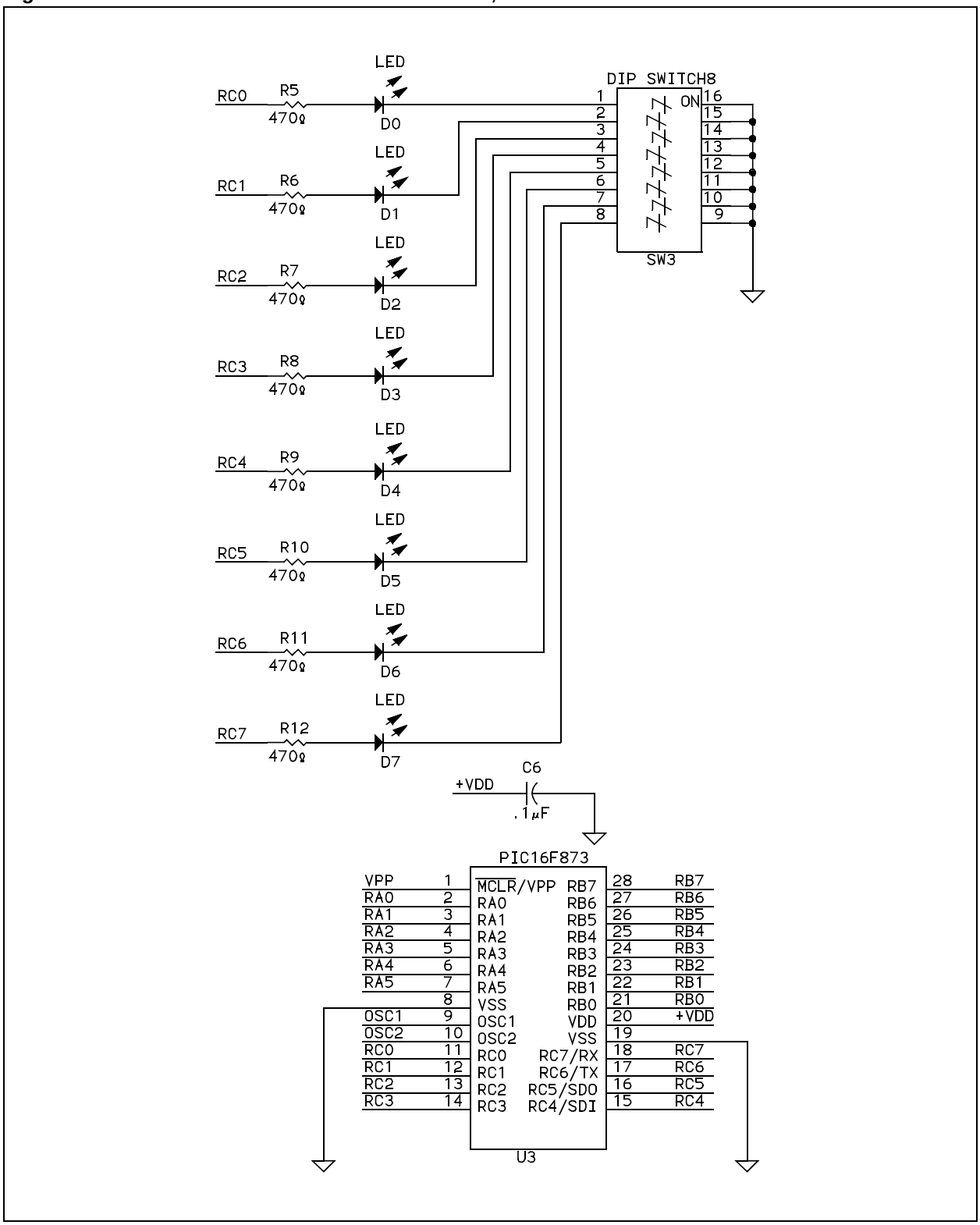
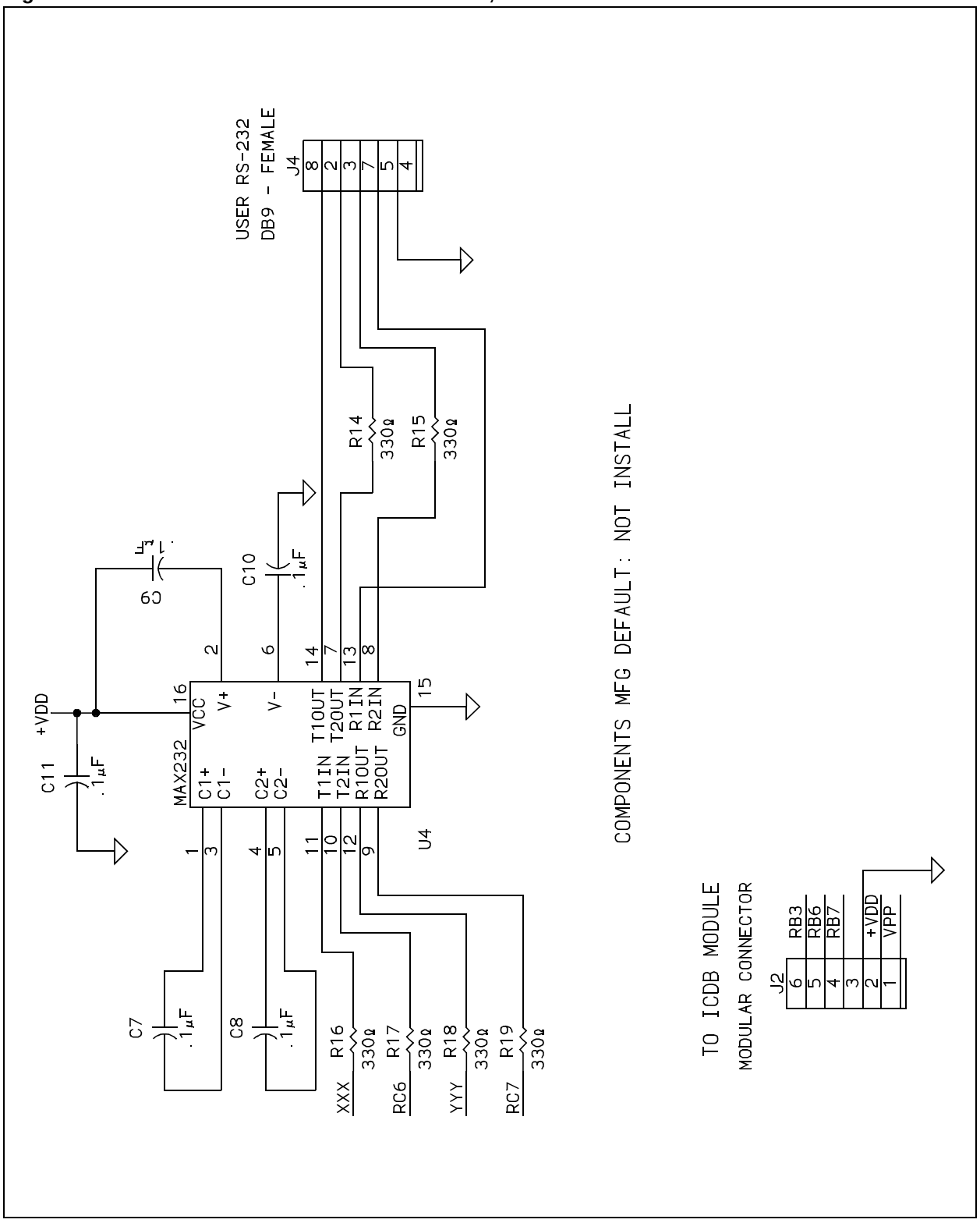


Figure A.11: MPLAB-ICD Demo Board Schematic, Part 4



NOTES:

Index

NOTES:

NOTES:

NOTES:

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-786-7200 Fax: 480-786-7277
Technical Support: 480-786-7627
Web Address: http://www.microchip.com

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508-480-9990 Fax: 508-480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
Microchip Technology Inc.
4570 Westgrove Drive, Suite 160
Addison, TX 75248
Tel: 972-818-7423 Fax: 972-818-2924

Dayton
Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Detroit
Microchip Technology Inc.
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

AMERICAS (continued)

Toronto
Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

ASIA/PACIFIC

Beijing
Microchip Technology, Beijing
Unit 915, 6 Chaoyangmen Bei Dajie
Dong Erhuan Road, Dongcheng District
New China Hong Kong Manhattan Building
Beijing 100027 PRC
Tel: 86-10-85282100 Fax: 86-10-85282104

Hong Kong
Microchip Asia Pacific
Unit 2101, Tower 2
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India
Microchip Technology Inc.
India Liaison Office
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-0061 Fax: 91-80-229-0062

Japan
Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa 222-0033 Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Shanghai
Microchip Technology
Unit B701, Far East International Plaza,
No. 317, Xianxia Road
Shanghai, 200051 P.R.C
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

ASIA/PACIFIC (continued)

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan, R.O.C
Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Denmark ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Arizona Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 München, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5858 Fax: 44-118 921-5835

01/21/00

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV — DNV MSC The Netherlands Accredited by the RvA — ISO 9001 / QS-9000 REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.