

ICD Module

(P/N 905501)

Instruction Manual



283 Indian River Road

Orange, CT 06477 USA

Tel 203-799-7875 Fax 203-799-7892

www.diversifiedengineering.net



Table of Contents

Section	Description	Page
1	Overview	3
2	Intended Audience	3
3	Equipment for the ICD	3
4	Getting Started	4
4.1	Design Considerations	4
4.2	Connections	4
4.3	System Set-Up	6
5	Debugging	6
5.1	ICD Set-Up	6
5.2	Running	7
5.3	Setting Breakpoints	8
5.4	Stopping	8
5.5	Stepping	8
5.6	Resetting	8
5.7	Watch Windows	9
6	Programming	9
6.1	Processor Selection	9
6.2	Environment Set-Up	9
7	Updating the ICD Operating System	9
8	Operating Below 4.5V	10
9	Tips	10
10	Troubleshooting	11
11	Specifications	11
	Limited Warranty	15
	Customer Service	15



1 Overview

An ICD (In-Circuit Debugger) Module is a development tool used for debugging embedded micro-controllers in a circuit. It can be used to debug and program embedded micro-controllers with flash memory from the Microchip family of controllers. The ICD is a cost-effective tool for testing software in real time and is compatible with the Diversified Engineering Educational Products.

2 Intended Audience

This user's manual assumes the reader has basic knowledge writing software in MPLAB. Students who want to expand their horizons and explore the possibilities developing projects using embedded micro-controllers can benefit greatly with the help of an ICD Module. Engineers and experimenters who can use a compact method for developing products in-house or out in the field will also appreciate the power of an ICD. Programs can be updated after production simply by erasing and re-programming the micro-controller while it is in the circuit using the ICD. Some common changes, for example, would be to a critical algorithm or constants for a specific part that may have special characteristics for custom designed products.

3 Equipment for the ICD

Microsoft Windows 3.X/95/98/2000
Processor Speed 486 or greater
4MB RAM (minimum)
8MB Hard Drive Space (minimum)
MPLAB-IDE
Available Serial Port
Serial Cable
ICD Module
Target Board
Connector



4 Getting Started

4.1 Design Considerations

When developing software for a product, a few changes and considerations need to be made in order to utilize the ICD. The first item is to add a NOP instruction immediately after program origin 00h. Next, 70h and 1EBh to 1EFh are general-purpose registers reserved specifically for the ICD only. If these addresses are being used, the software needs to be altered to allow the ICD to access them. Program memory 0x1F00 to 0x1FFF is reserved for the debug code and pins RB6 and RB7 on the target chip are used by the ICD for in-circuit debugging and programming.

4.2 Connections

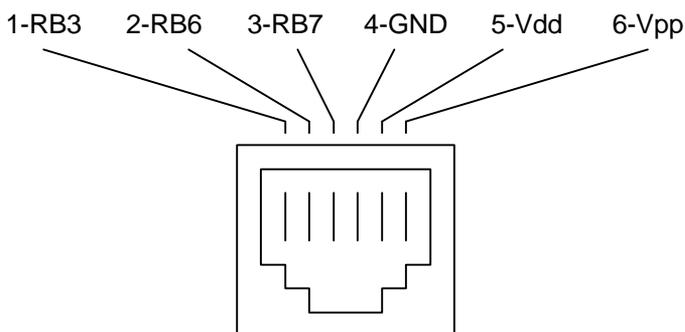


Figure 1 (View of ICD RJ11 Female Connector Looking In).

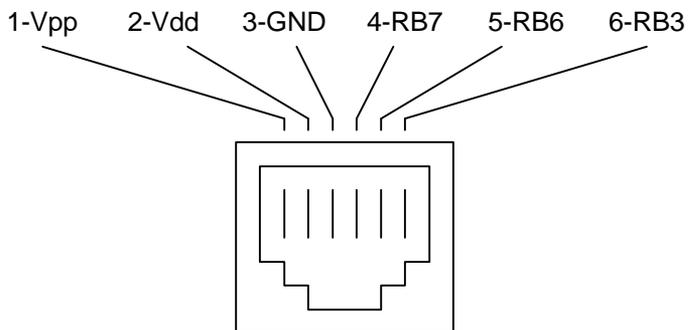


Figure 2 (View of Target Board RJ11 Female Connector Looking In).

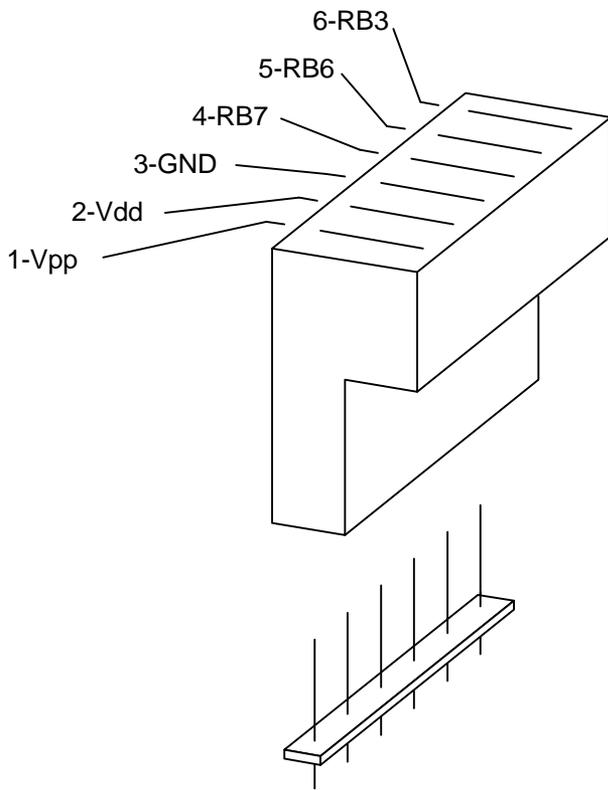


Figure 3 (View of MTA Connector and Target Board Header).



4.3 System Set-Up

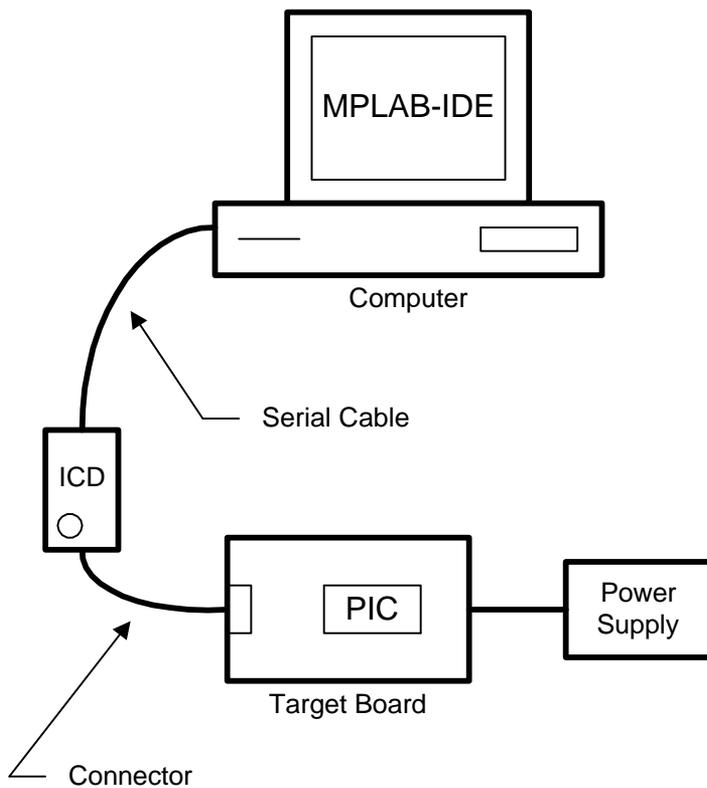


Figure 4 (ICD System Set-Up).

5 Debugging

5.1 ICD Set-Up

Open MPLAB-IDE and select the following:

File>Options>Development Mode



1. Under the “Tools Tab” select MPLAB ICD Debugger and choose the processor using the pull down menu. Under the “Break Options Tab” check the debugging preferences for breakpoints.
2. Press the “Apply” button then “OK”
3. If the ICD was unable to establish communications at this point, check the connections and try to reconnect.

At this point, open up the project that requires debugging and go through the following checklist:

1. Make sure the Status field indicates a connection has been made.
2. Set the communication port, baud rate, register update and target board frequency.
3. Verify the firmware version handles the processor in the target board.
4. Press the Options button on the MPLAB ICD Version dialog box.
5. Make sure the device matches the one actually being used.
6. Set the configuration bits to compliment the design.
7. Set the program options as desired.
8. Make sure the Enable Debug Mode is checked.
9. Use the Update button to check the voltages.
10. Press the Blank button to check if the target processor needs to be erased.
11. If the device is blank then skip the next step.
12. To blank the device, the Erase All Before Program box can be checked or it can be erased manually simply by pressing the Erase button.
13. Press the Program button and wait until programming is complete.
14. The Verify button can be pressed to compare the program memory in MPLAB to the program memory in the device.
15. The Read button can be pressed to upload the program memory of the device to the program memory of MPLAB.

The tool bar can be customized to optimize the users debugging time. This manual will use the menu bar to explain the debugging.

5.2 Running



Debug>Run>Run

This will initiate the program and allow the target board to operate with the software.

5.3 Setting Breakpoints

Breakpoints can be set in either the List file (*.lst) or the Main file (*.asm) file. The List file can be opened by executing Window>Absolute Listing. A breakpoint can only be set on an MPLAB instruction from the instruction set. Macros can be handled by using the List file to view the expanded macro and break on the first MPLAB instruction. To set a breakpoint, maneuver the mouse cursor over the line of code to be monitored and click the right mouse button. Then use the left mouse button to choose the breakpoint setting. Only one breakpoint can be set at a time and it will only stop if the program tries to execute that line of code while running. If stopped, the registers will be updated depending on the register update selection. Use caution when setting a breakpoint in an actual system because when the program stops, the device on the target board stays in the last state.

5.4 Stopping

Debug>Run>Halt

This will stop the program and indicate what line the program left off along with updating the registers depending on the register update selection. Use caution when stopping in an actual system because when the program stops, the device on the target board stays in the last state.

5.5 Stepping

Debug>Run>Step

This will execute one line of code at a time while updating the registers if chosen to do so.

5.6 Resetting

Debug>Run>Reset



This will reset the device to program origin 00h.

5.7 Watch Windows

When using watch windows, be sure that they are selected properly in the ICD Version dialog box so that they are updated or not updated as expected.

Window>Special Function Registers

All special function registers can be viewed.

Window>Watch Window

All general-purpose registers can be viewed by selecting each one individually and choosing how they are to be displayed.

6 Programming

6.1 Processor Selection

Only Microchip Flash devices can be programmed using the ICD module.

6.2 Environment Set-Up

Use the debugging steps with the exception of leaving the Enable Debug Mode box blank. When the program is complete, unplug the connector to the target board and the program should then run on its own.

7 Updating the ICD Operating System

This feature is supported starting with Firmware V2.02 and MPLAB IDE V5.00. The ICD is programmed with a version that is current at the time of production. As newer versions of MPLAB become available or new products are being introduced, the ICD should be updated accordingly. With the ICD and the target board connected,



press the “Download ICD Operating System” button in the Options dialog box and wait until the download is complete.

8 Operating Below 4.5V

When deciding to operate below 4.5V, bulk erase should not be used. Doing so will cause the processor to be code protected which may be undesirable. Make sure the “Erase All Before Programming” box is unchecked and do not use the “Erase” button. Instead, specify the program memory range depending on debug or program mode and this will cause the ICD to erase before programming on a word by word basis. In order to work properly, the device has to be initially blank or erased and operate at the voltage intended. It can be erased by bringing the voltage above 4.5V and pressing the “Erase” button, using the PIC Start Plus or using the Pro Mate II.

9 Tips

- Only one Options Dialog Box should be open at a time to prevent confusion.
- Before opening up MPLAB, make sure the ICD is connected properly and the target board is on.
- Always close MPLAB before removing power from the target board.
- Be careful when stopping MPLAB while testing target board. For instance, when driving a DC motor with an H-Bridge configuration using a PWM signal, it is possible for one line to be high and one line to be low. This will make the motor run at full speed when the program stops. Resetting the program will tri-state all of the I/O pins.
- If stepping and stopping is taking too long while debugging, choose to update fewer registers.
- Breakpoints can only be set on actual PIC instructions, not on macros. The list file breaks up the macros into individual instructions so the breakpoint may be placed there.
- Only one breakpoint can be set at any given time.
- The processor voltage and the ICD voltage should be the same.
- Program a hex file by importing it to memory (open ICD first).



10 Troubleshooting

- ICD is not connecting, check the serial cable and make sure the target board is on.
- Incorrect end address when trying to program, use default end address.
- Error requested operation when trying to run or step while the LED is solid, check Vpp, RB6 and RB7 connections.
- Error requested operation when trying to run or step while the LED is solid, check if the firmware version number handles the processor being used. Check the file named (readme.icd) in the MPLAB directory for latest versions to handle new processors.

11 Specifications

- 3V to 5V operation
- Maximum current required from target board is:

	5V	3V
Standby	30mA	20mA
Program	50mA	40mA

- Overall dimensions are approximately:

2 $\frac{3}{4}$ inches long
1 $\frac{3}{4}$ inches wide
1 inch high



NOTES:



NOTES:



NOTES:



Limited Warranty

Diversified Engineering and Manufacturing Inc. (DEM) warrants that the Product(s) will be free from defects in material and workmanship for a period of one (1) year from the date of purchase. Diversified Engineering & Manufacturing Inc. will, at its discretion, repair or replace any part(s) found to be defective in the Product(s) resulting from defective workmanship, material or both. All costs for packaging and transporting *to and from* Orange CT are the responsibility of the customer. All units returned for repair should have prepaid shipping charges, include a proof of purchase receipt and a detail of the problem/symptom.

There is no other Warranty express or implied. This warranty does not apply to any defect, failure or damage caused by improper use or storage of the product. DEM will not be obliged to provide warranty service for units that:

- 1) were damaged from improper use or interconnect to external equipment;
- 2) that have been modified or tampered with or
- 3) improperly stored or exposed to elements.

Customer Service

Technical support and service is available from:

9:00 AM to 4:30 PM , M-F, EST.

TEL (203) 799-7875 FAX (203) 799-7892

All materials should be addressed to:

Diversified Engineering and Manufacturing Inc.

283 Indian River Road

Orange, CT 06477

- This publication and the information provided are subject to change, errors and omissions.
- MPLAB-IDE, MPLAB, PIC Start Plus, Pro Mate II, PICmicro and PIC are tradenames of Microchip Technology Inc..
- Microsoft Windows is a tradename of Microsoft

Copyright ©2001 Diversified Engineering and Manufacturing Inc.

All rights reserved.