

PIC-ICD PIC16F87x In Circuit Debugger

PIC-ICD is a low-cost development system for Microchip's flash based PIC16F87x microcontrollers. Utilizing the chips' In-Circuit Debugging (ICD) capability and Microchip's In-Circuit Programming (ISP) protocol, the PIC-ICD allows users to debug source code in the application, debug hardware in real-time and program a target PIC16F87X device. It provides a cost-effective way to allow users to "emulate" the PIC16F87X MCU without the need of an in-circuit emulator. With special caution, the PIC-ICD can also be used to support most PIC16C6X/7X parts.

The PIC-ICD system runs under its own IDE or under MPLAB™ from Microchip. The software provides the front end for all the programming and emulation controls like running, stepping, setting breakpoint, etc.

PIC-ICD comes with the debugger module and a Demo Board. The ICD Module contains all debugging, programming and control logic. It is connected to the PC's parallel port which provides fast communication enabling all data registers to be updated instantaneously during stepping.

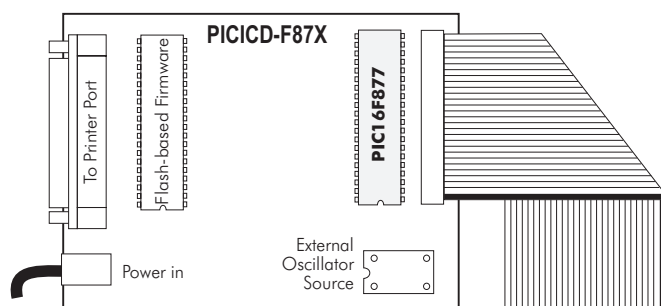
The debugger cable ends in a 40-pin DIP plug for in-system test. A 40-pin to 28-pin SDIP adapter is also included to debug the smaller 28-pin PIC16F87x parts. Optional PLCC, QFP and

SOIC surface mount adapters are also available.

The ICD Demo Board is provided for demonstration and/or evaluation of the PIC16F87x in the absence of a target application board. The demo board provides LEDs, DIP switch, push buttons, and a potentiometer for demonstrating the chip's features. A small prototyping area and RS232 circuitry is also available for experimentation. The PIC16F877 can be unplugged from the PIC-ICD module and plugged directly into the demo board for stand-alone operation.

Features

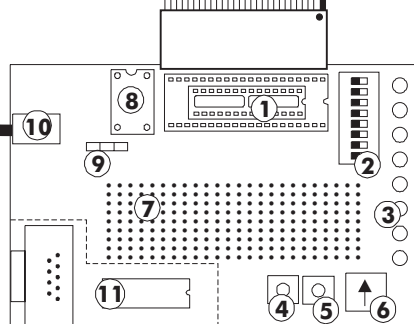
- Real-time code execution
- High speed parallel port interface
- In-circuit debugger and Programmer
- 2.5V to 6.0V operating range
- One level Real-time Breakpoint
- Two External Break Inputs
- Conditional Animation Break
- Operating frequencies from 32Khz to 20Mhz
- Runs under PICICD IDE (Win 95/98/NT) or MPLAB™ (Win 95/98 only)
- Source level and symbolic debugging
- Software animation trace captures 3 user-defined variables in addition to opcode, W, STATUS, FSR



PIC-ICD Resources Used

Due to the built-in ICD and ISP functions, the PIC-ICD will use up the following on-chip resources:

- MCLR/Vpp shared for programming
- RB6 and RB7 reserved for programming and debugging (can be released for use when not in debugging mode via jumpers)
- 6 general purpose registers (70h, 1EBh-1EFh) reserved for debug monitor
- program memory (0x1F00-0x1FFF reserved for debug code; one stack level not available).

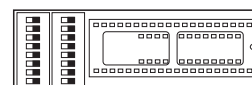


PIC-ICD Operation

The PIC16F877 must be plugged in and programmed with the test program and debug codes for debugging. To run codes without debugging capability, reprogram the codes and fuse bits (without debug codes) and set the jumpers on the PIC-ICD to "Normal IC Mode" to free RB6 and RB7 for test.

40-pin to 28-pin Header

Connect this header to the end of the 40-pin emulation cable to convert it to a 28-pin SDIP footprint. There are two 8-position DIP switches on the header which allow users to use PORTD of the 16F877 to emulate PORTB of the smaller 16F873/876 and compensate for the RB6 and RB7 lost during debugging.



ICD Demo Board

- 1 40-pin/28-pin DIP sockets for in-system test connection
- 2 DIP switch - set to on for Port C high
- 3 Port C Indicator LEDs
- 4 Hardware Reset button
- 5 RBO (active low) button
- 6 Potentiometer for RAO
- 7 Prototype Area
- 8 Socket for external onboard crystal
- 9 Oscillator setting
- 10 9VDC power in (not required when hosted by PIC-ICD)
- 11 RS232 circuitry

CHF 285.- excl. MWST., without Demo Board CHF 235.00

PIC-ICD PIC16F877 In-Circuit Debugger

The PIC-ICD debugger comes with an Integrated Development Environment (IDE) software which runs under Windows 95/98/NT. The IDE user interface is so intuitive that there is no learning curve to the system and users can be up and running within minutes. All information regarding the processor and the application is available right on the screen. It provides separate windows to examine source codes, data memory, special file registers and watch variables. For users who are more familiar with and prefer to use the MPLAB™ IDE from Microchip, a special DLL driver is included which allows the PIC-ICD to run off the parallel port under Windows 95/98 (Win NT not supported with MPLAB.).

Special Function Tool Buttons

Load File - Loads COD, binary and hex files to the IDE for debug Opens Device Configuration Window - Selects device type and fuse bit settings Compile - Compiles the source file using supported compilers Toggle Breakpoint - Quickly enables or disables current program counter as breakpoint Show Next Statement - Quickly returns to the next instruction to be executed after strolling through source codes Add Watch Variables - Isolates special variables to be inspected in a separate window Run to Cursor - Executes codes in real-time from current program counter to the one highlighted by the cursor; can be used to step over or exit function calls Animation Trace - Enables saving the non-real-time program execution to a memory buffer Update Registers - Updates all registers in the current program states

The Animation Trace allows users to specify 3 registers to be saved to the non-real-time animation trace in addition to the W, STATUS and FSR registers, opcode and source instructions.

Conditional Animation Break between 2 operands and select from a list of break

There are 2 external break input sources from the target circuit: BRK1 is an edge trigger break and BRK2 is an edge count break which halts the processors after a specified number of edge changes.

Feature Comparison between MPLAB™ -ICD and PIC-ICD

	MPLAB™-ICD	PIC-ICD
Interface	Serial Port	Parallel Port
Operating Environment	MPLAB™ IDE - Win 95/98/NT	PICICD - Win 95/98/NT, MPLAB™ IDE - Win 95/98
Emulation Vdd Voltage	3 to 5.5 volt	2.5 to 6 volt
External Break	None	Edge trigger break input and edge count break input
Bits RB6, RB7	Occupied at all times	Free when not in debug mode via jumper setting
Target Frequency	Cannot read back target frequency	Free when not in debug mode via jumper setting
Firmware Upgrade	Reprogram a new firmware chip	Can read back target frequency in debug mode
Function Timing for 8K words:	Depends on target frequency and voltage	Re-download codes to flash-based firmware chip
• Read and Verify	• 24 seconds	Independent of target frequency and voltage
• Complete Program Cycle	• 2 min 45 seconds	• 2.5 seconds
• Program Debug Module	• 5.5 seconds	• Less than 1 min 45 seconds
• Reset	• 2.6 seconds	• 4 seconds
• Update all Registers	• 1.35 to 60 seconds	• 1.2 seconds
		• Less than 1 second