Fondamenti su strumenti di sviluppo per microcontrollori PIC

MPSIM

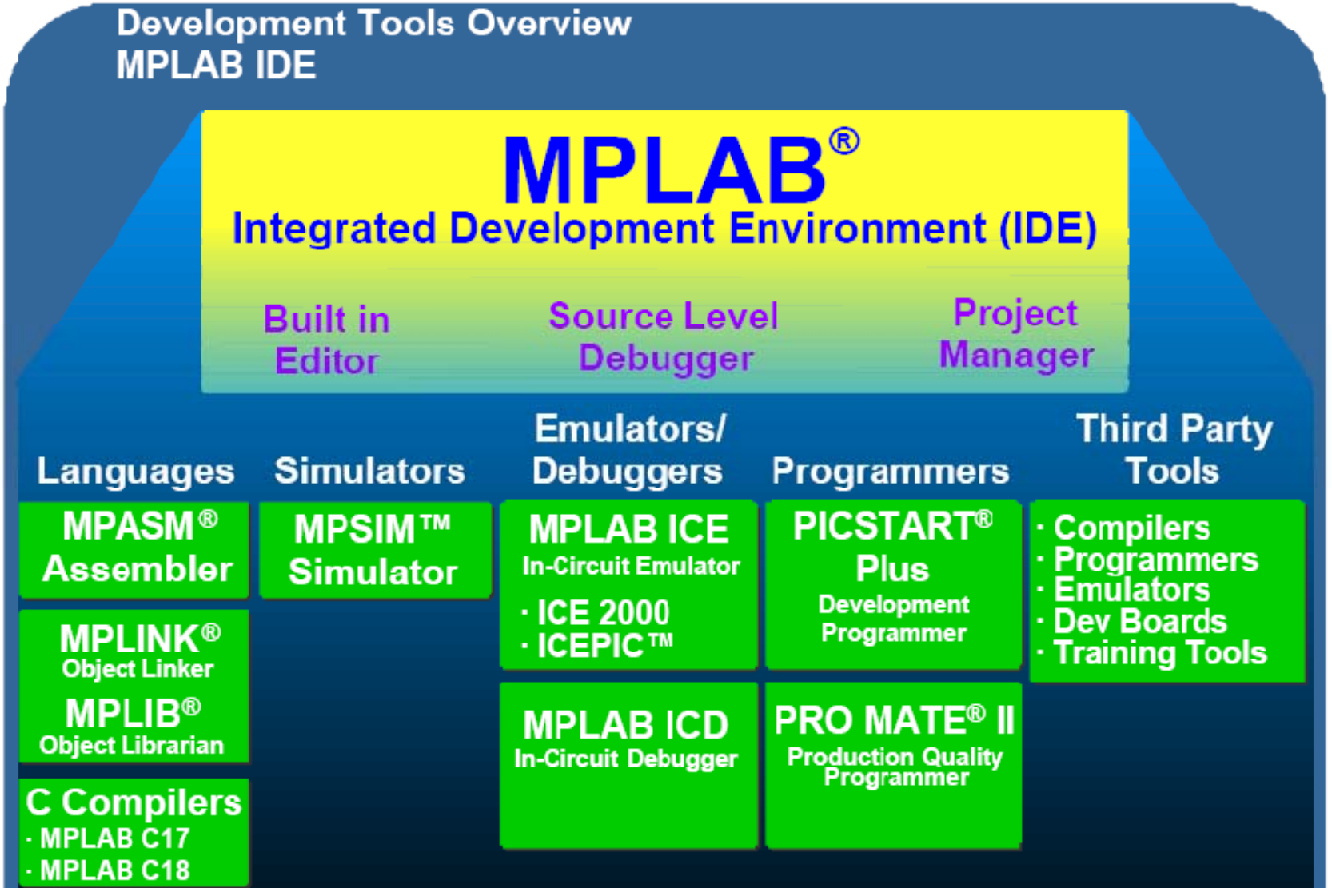
ICE 2000

ICD 2

REAL ICE

PICSTART

Ad uso interno del corso Elettronica e Telecomunicazioni


Development Tools Overview
MPLAB IDE
MPLAB®
Integrated Development Environment (IDE)
Built in Editor
Source Level Debugger
Project Manager
Languages
MPASM® Assembler
MPLINK® Object Linker
MPLIB® Object Librarian
C Compilers
· MPLAB C17
· MPLAB C18
Simulators
MPSIM™ Simulator
Emulators/ Debuggers
MPLAB ICE In-Circuit Emulator
· ICE 2000
· ICEPIC™
MPLAB ICD In-Circuit Debugger
Programmers
PICSTART® Plus Development Programmer
PRO MATE® II Production Quality Programmer
Third Party Tools
· Compilers
· Programmers
· Emulators
· Dev Boards
· Training Tools

MPLAB SIM /1

MPLAB SIM is a discrete-event simulator for:

- PICmicro microcontroller (MCU) families
- dsPIC digital signal controller (DSC) families

MPLAB SIM allows you to:

- Modify object code and immediately re-execute it
- Inject external stimuli to the simulated processor
- Set pin and register values at prespecified intervals
- Acquire real-time time stamping so you can see how long your code takes to execute
- Trace the execution of the object code (MCU's only)
- Graphically view digital pin signals over a defined time period (logic analyzer)

MPLAB SIM /2

When MPLAB SIM is simulating running in real-time, instructions are executing as quickly as the PC's CPU will allow. This is usually slower than the actual device would run at its rated clock speed.

The speed at which the simulator runs depends on the speed of your computer and how many other tasks you have running in the background. The software simulator must update all of the simulated registers and RAM, as well as monitor I/O, set and clear flags, check for break and trace points in software and simulate the instruction with instructions being executed on your computer's CPU.

The execution speed of a discrete-event software simulator is orders of magnitude less than a hardware oriented solution.

Often loops will be used in your code to generate timing delays. When using the simulator, you might wish to decrease these time delays

MPLAB SIM /3

MPLAB SIM only simulates to the register level, not the pin level, e.g., RB0 represents the value in bit0 of the PORTB register, not the value on the pin named RB0. This makes sense as the simulator is a software model, and not actual device hardware.

ICE 2000 /1

MPLAB ICE 2000 is an In-Circuit Emulator (ICE) designed to emulate most PICmicro microcontroller (MCU) devices.

MPLAB ICE 2000 performs basic functions such as start and stopping emulation (with step-by-step feature), viewing processor memory and files registers, using software and hardware breakpoints, using trigger in/out settings, monitoring emulator states and operations, plus advanced features such as external memory usage, instruction data trace, complex triggering

ICE 2000 /2

MPLAB ICE 2000 allows you to:

- Debug your application on your own hardware in real time.
- Debug with both hardware and software breakpoints.
- Measure timing between events using the stopwatch or complex trigger.
- Set breakpoints based on internal and/or external signals.
- Monitor internal file registers.
- Emulate full speed up to 48 MHz (depending on the device).
- Select the oscillator source in software.
- Program the application clock speed.
- Trace data bus activity and time stamp events.
- Set complex triggers based on program and data bus events, and external inputs.

ICE 2000 /3

The components of the MPLAB ICE 2000 emulator kit are listed below.

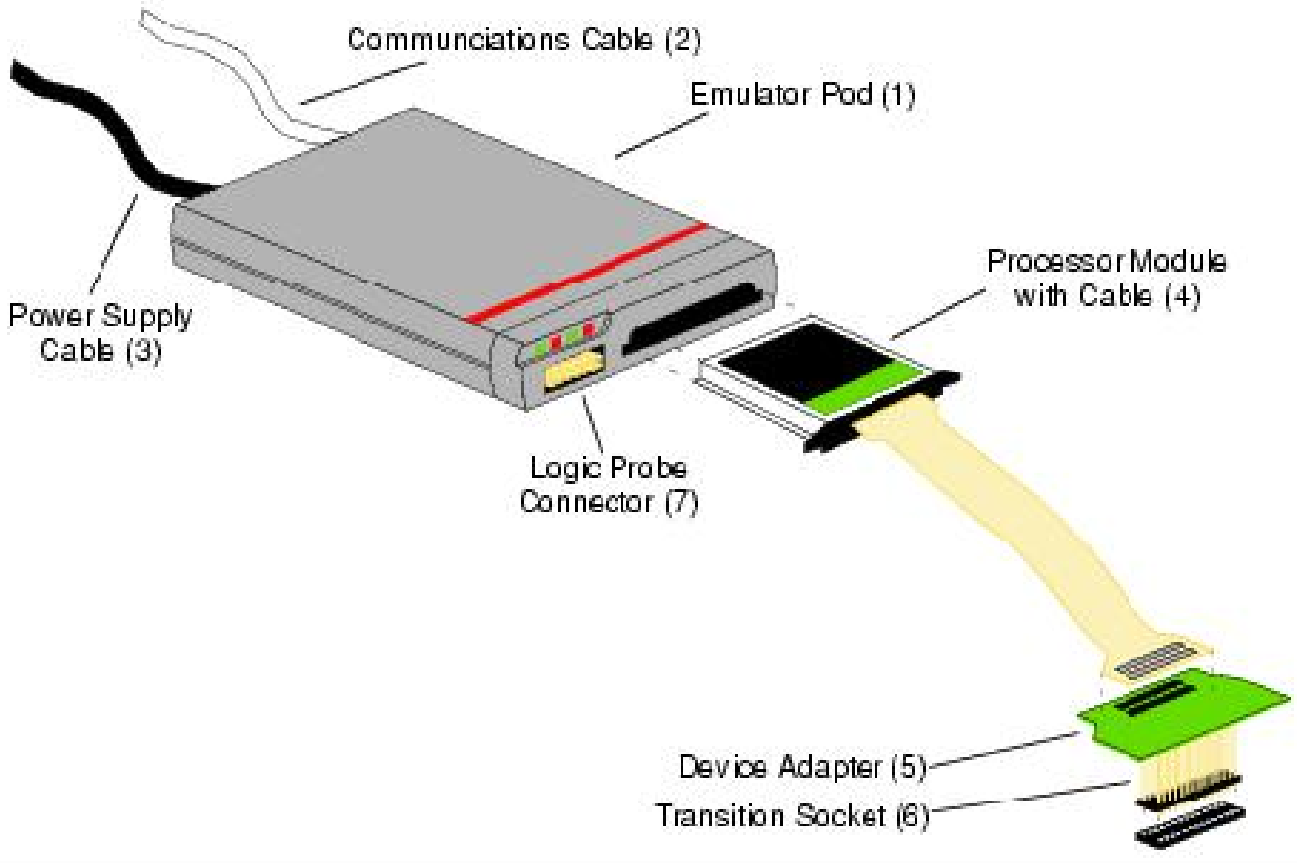
1) MPLAB IDE Quick Start
2) CD-ROM with MPLAB IDE software and on-line documentation
3) Parallel communications cable to connect the emulator pod to a PC
4) Emulator pod
5) Power supply and cable
6) Emulator stand
7) Logic probes

Additional hardware that may be ordered separately:

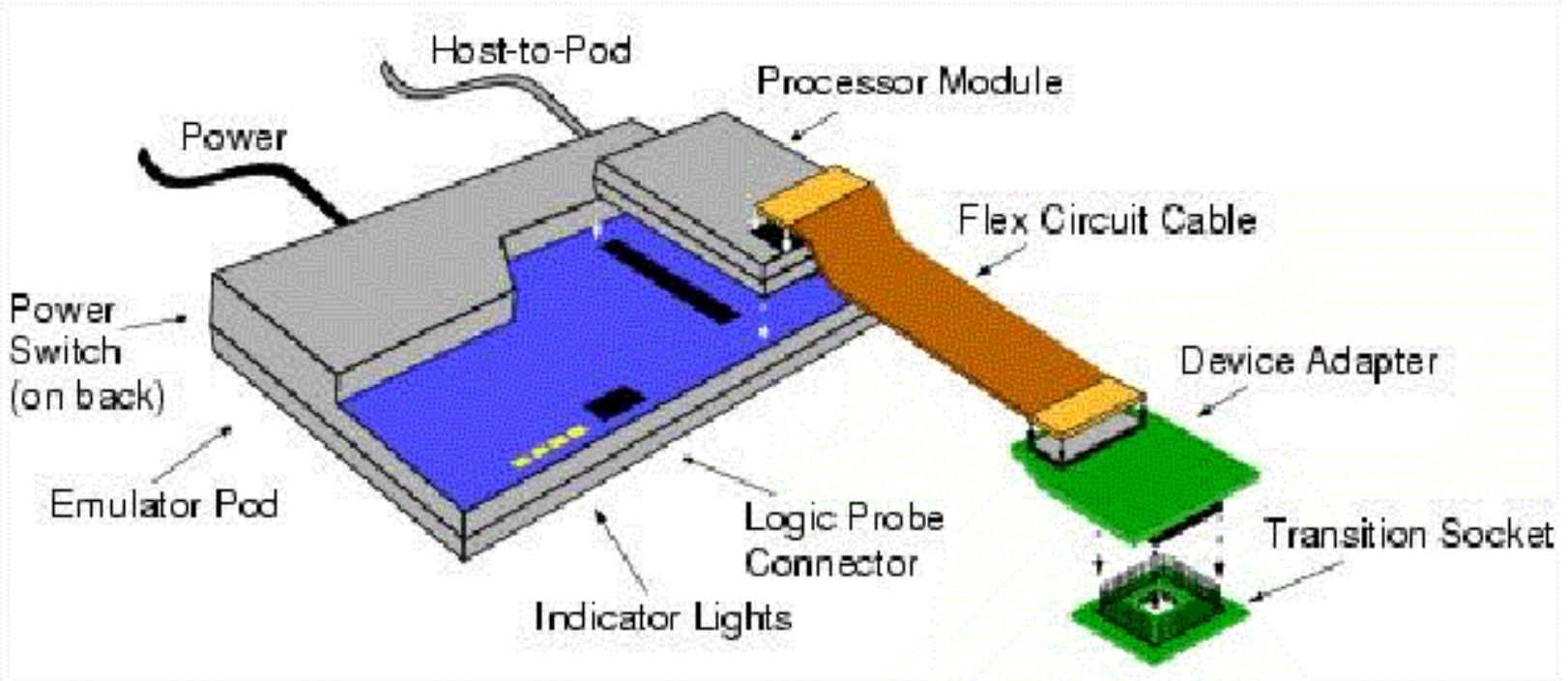
8) Processor module (with flex circuit cable)
9) Device adapter to connect the processor module to the transition socket
10) Transition socket to connect the device adapter to the target system
11) Parallel-to-USB converter (not shown)

ICE 2000 /4

ICE 2000 /5


Communciations Cable (2)
Emulator Pod (1)
Processor Module with Cable (4)
Power Supply Cable (3)
Logic Probe Connector (7)
Device Adapter (5)
Transition Socket (6)

ICE 2000 /6


Host-to-Pod
Processor Module
Power
Flex Circuit Cable
Power
Switch
(on back)
Device Adapter
Emulator Pod
Logic Probe
Connector
Transition Socket
Indicator Lights

ICE 2000 /7

The emulator pod connects to the PC through either a parallel port or a USB port. The pod contains the hardware necessary to perform the common emulator functions, such as trace, break and emulate. It also permits the communications between PC and processor module

The processor module inserts into a slot in the front of the emulator pod. It contains the hardware necessary to emulate a specific device or family of devices

The device adapter is connected to the processor module by the flex circuit cable. Device adapters are interchangeable assemblies that allow the emulator to interface to a target application system. Device adapters also have control logic that allows the target application to provide a clock source and power to the processor module

ICE 2000 /8

The transition socket is connected to the device adapter. Transition sockets are available in various styles to allow a common device adapter to be connected to one of the supported surface mount package styles.

The logic probes may be connected into the logic probe connector on the emulator pod.

ICD 2 /1

The MPLAB ICD 2 is a low-cost in-circuit debugger (ICD) and in-circuit serial programmer (ICSP).

The MPLAB ICD 2 offers these features:

- Real-time and single-step code execution
- Breakpoints, Register and Variable Watch/Modify
- In-circuit debugging
- Target Vdd monitor
- Diagnostic LEDs
- MPLAB IDE user interface
- RS-232 serial or USB interface to a host PC

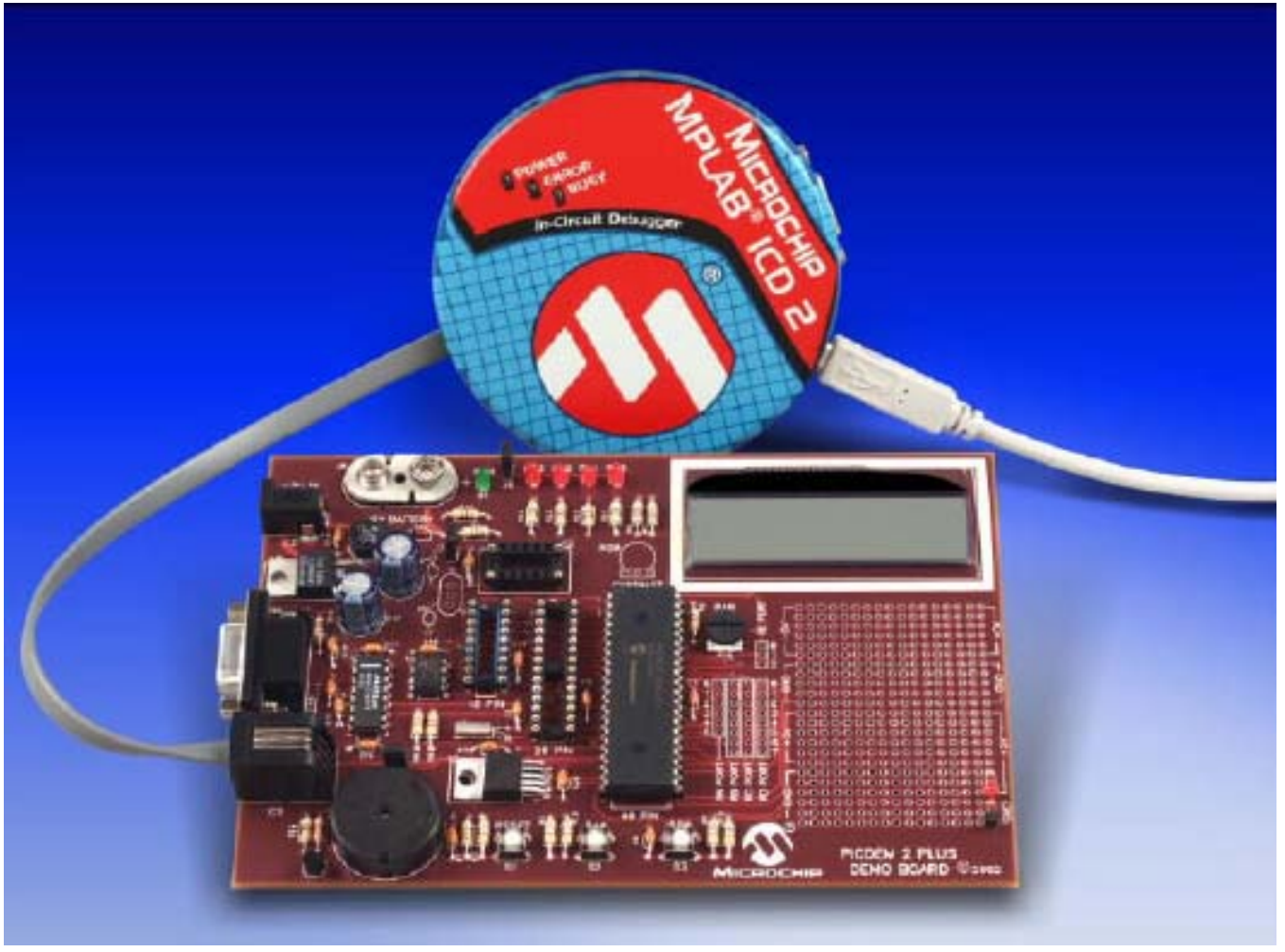
ICD 2 /2

The MPLAB ICD 2 allows you to:

- –Debug your source code in your own application
- –Debug your hardware in real-time
- –Program a supported device using Microchip's ICSP protocol

The MPLAB ICD 2 utilizes, if present, the in-circuit debugging capability built into the FLASH devices. In absence, it’s impossible to proceed with debugging

ICD 2 /3


MICROCHIP
MPLAB® ICD 2
POWER
ERROR
BUSY
In-Circuit Debugger
MICROCHIP
PICDEM 2 PLUS
DEMO BOARD

ICD 2 /4

ICD vs. ICE

The in-circuit debugger (ICD) is a cost-efficient alternative to an in-circuit emulator (ICE). It can do many things that were previously done only with more expensive hardware, but the cost benefits come with a trade-off of some of the conveniences of an in-circuit emulator.

As opposed to an ICE, some of the requirements of the in-circuit debugger are:

- –The in-circuit debugger requires exclusive use of some hardware and software resources of the target.
- –The target PICmicro MCU must have a functioning clock and be running.
- –The ICD can debug only when all the links in the system are fully functional.

ICD 2 /5

ICD vs. ICE

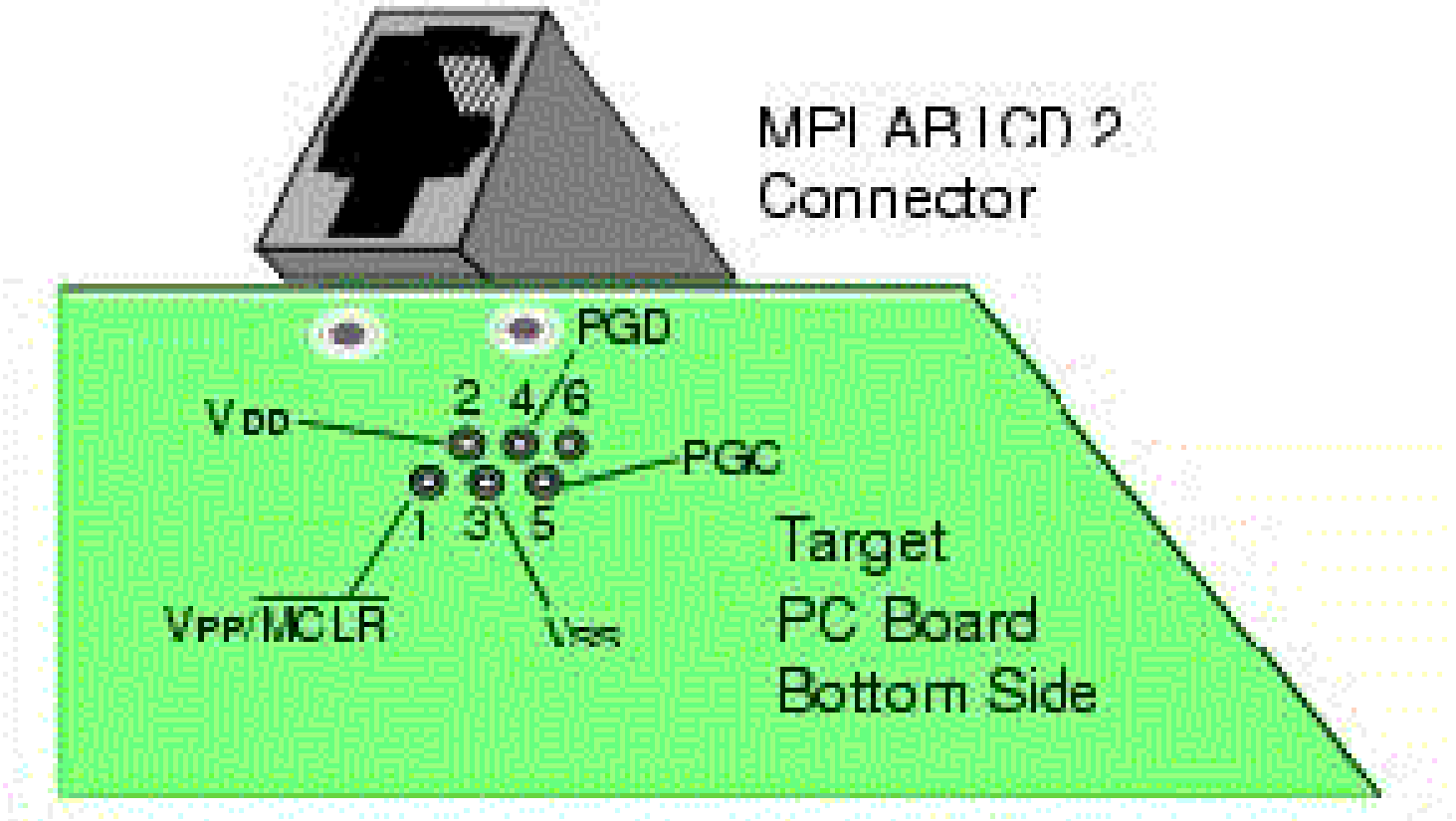
An emulator can run code even without being connected to the target application board whereas an ICD may not be able to debug at all if the application does not run. On the other hand, an in-circuit debug connector can be placed on the application board and connected to an ICD even after the system is in production, allowing easy testing, debugging and reprogramming of the application. In this situation the ICD has some distinct advantages:

a connection to the application after the production cycle does not require extraction of the microcontroller in order to insert an ICE probe;

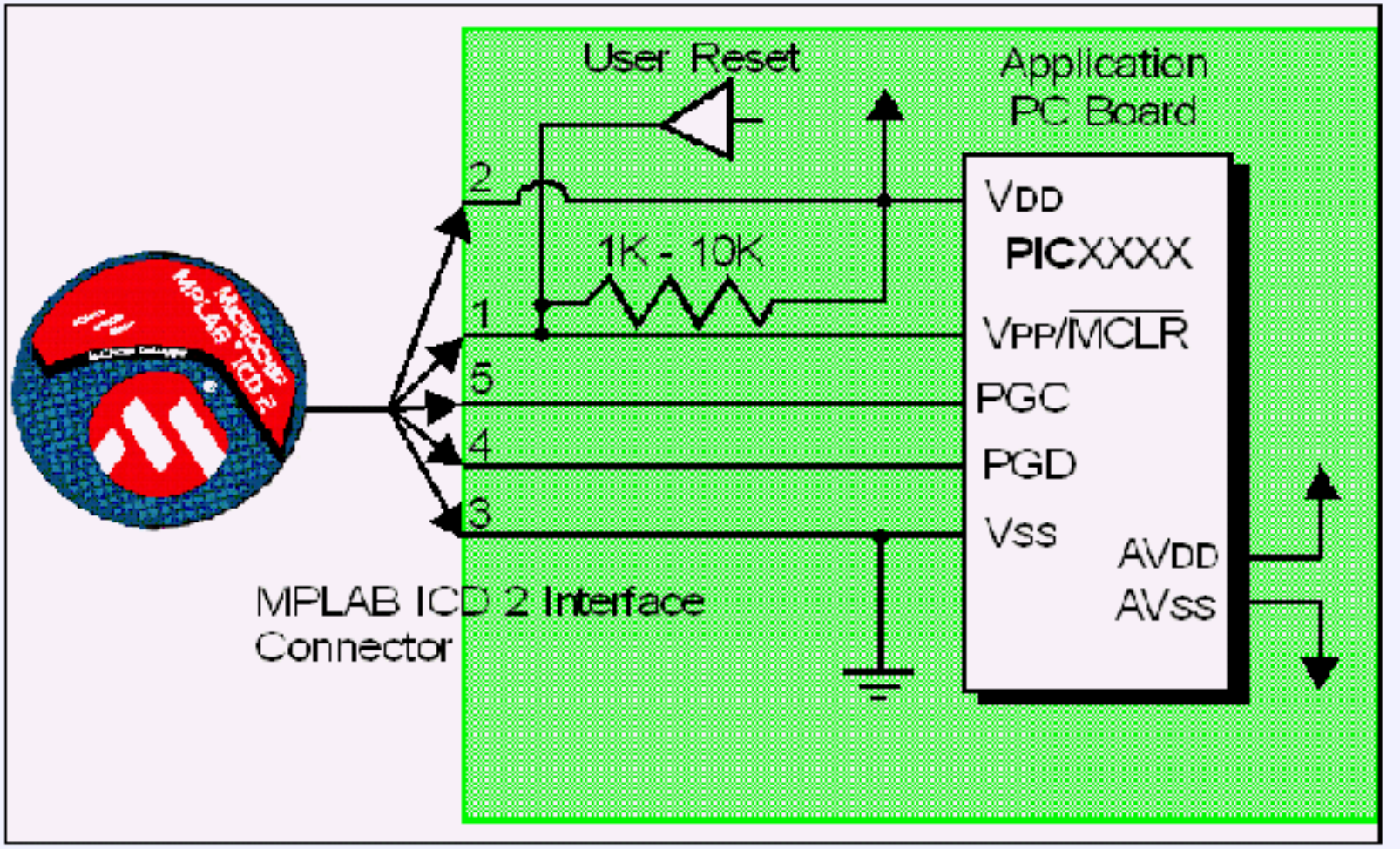
the ICD can re-program the firmware in the target application without any other connections or equipment.

ICD 2 /6

In-circuit debug connector



ICD 2 /7


User Reset
Application
PC Board
2
1
5
4
3
1K - 10K
VDD
PICXXXX
VPP/MCLR
PGC
PGD
Vss
AVDD
AVss
MPLAB ICD 2 Interface
Connector

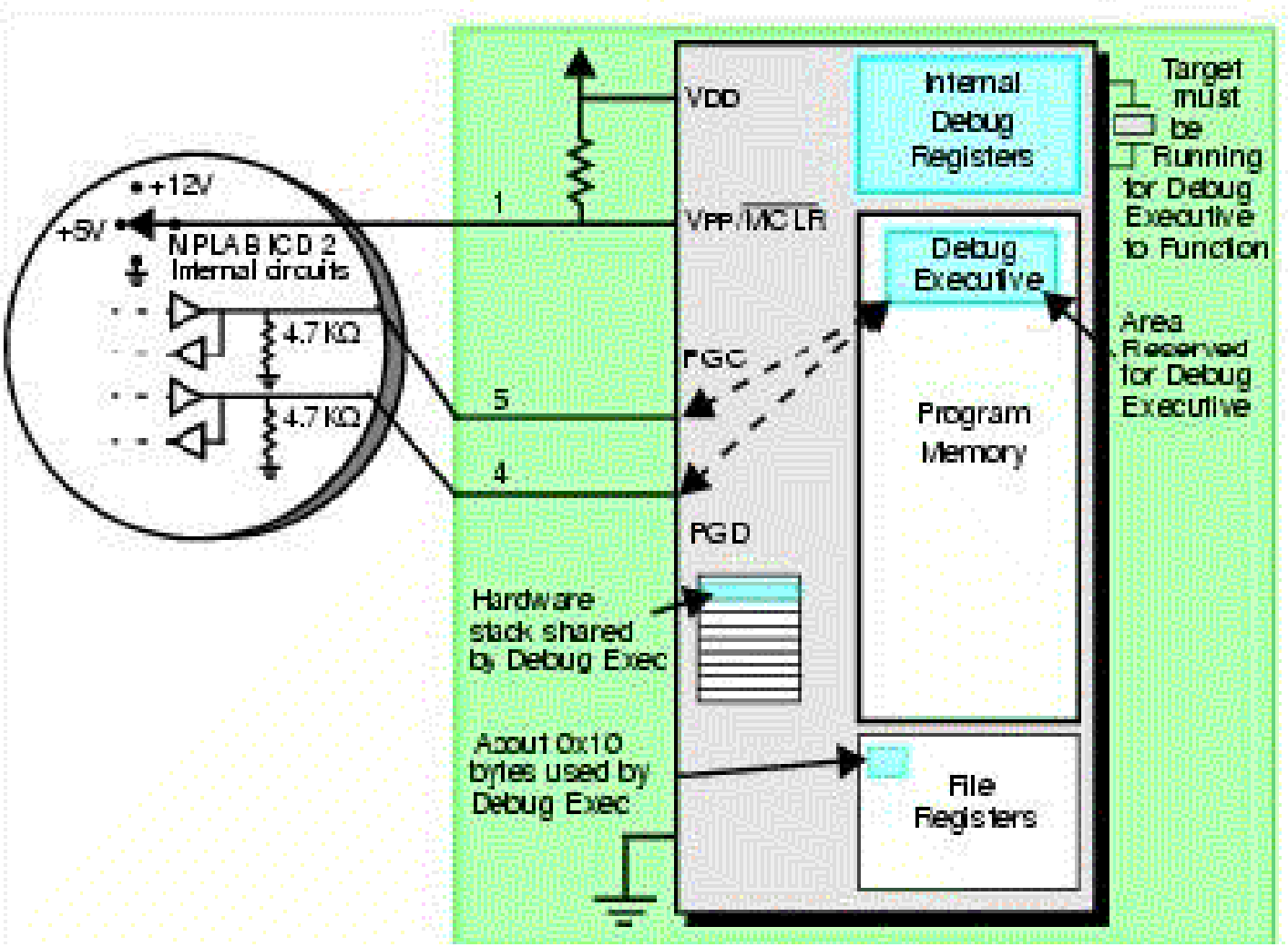
ICD 2 /8

Debug Mode

- There are two steps to using MPLAB ICD 2. The first requires that an application be programmed into the target PICmicro MCU. The second uses the internal in-circuit debug hardware of the target PICmicro MCU to run and test the application program. These two steps are directly related to the MPLAB IDE operations:
  1. Programming the code into the target.
  2. Using the debugger to set breakpoints and run.

- If the target PICmicro MCU cannot be programmed correctly, MPLAB ICD 2 will not be able to debug.

ICD 2 /9


+12V
+5V
MPLAB ICD 2
Internal circuits
4.7 KΩ
4.7 KΩ
1
5
4
VDD
VPP/MCLR
PGC
PGD
Hardware stack shared by Debug Exec
About 0x10 bytes used by Debug Exec
Internal Debug Registers
Debug Executive
Program Memory
File Registers
Target must be Running for Debug Executive to Function
Area Reserved for Debug Executive

ICD 2 /10

Requirements For Debug Mode

To debug (set breakpoints, see registers, etc.) with the MPLAB ICD 2 there are critical elements that must be working correctly:

- MPLAB ICD 2 must be connected to a PC. It must be powered by an external power supply or the PC via the USB cable, and it must be communicating with MPLAB IDE software via the RS-232 or USB cable.
- The MPLAB ICD 2 must be connected as shown to the Vpp, PGC and PGD pins of the target PIC MCU with the modular interface cable (or equivalent). Vss and Vdd are also required to be connected between the MPLAB ICD 2 and target PIC MCU.
- The target PIC MCU must have power and a functional, running oscillator. If the target PIC MCU does not run - for whatever reason - MPLAB ICD 2 cannot debug.
- The target PIC MCU must have its configuration words programmed correctly:
    - the oscillator configuration bits should correspond to RC, XT, etc., depending upon the target design
    - the target PIC MCU must not have the Watchdog Timer enabled.
    - the target must not have code protection enabled
    - the target must not have table read protection enabled.

ICD 2 /11

Resources Used for PIC16F Devices

MCLR pin reserved for debugging, i.e., you cannot use this pin as digital I/O while debugging.

MCLR/VPP shared for programming.

Low voltage ICSP programming disabled for devices that support this type of programming.

RB6 and RB7 reserved for programming and in-circuit debugging. Therefore, other functions multiplexed on these pins will not be available during debug.

One stack level not available.

REAL ICE /1

REAL ICE is an in-circuit emulator that is controlled by a PC running MPLAB IDE software on a Windows platform.

Supports hardware and software development for selected Microchip PIC microcontrollers (MCUs) and dsPIC Digital Signal Controllers (DSCs) that are based on In-Circuit Serial Programming (ICSP) programming capability and Standard DUT Programming (STDP) 2-wire serial interfaces.

The emulator system will execute code like an actual device because it uses a device with built-in emulation circuitry, instead of a special emulator chip, for emulation.

All available features of a given device are accessible interactively, and can be set and modified by the MPLAB IDE interface.

In addition to emulator functions, the REAL ICE in-circuit emulator system also may be used as a development programmer.

REAL ICE /2

The MPLAB REAL ICE in-circuit emulator system allows you to:

debug your application on your own hardware in real time
debug with hardware breakpoints
debug with software breakpoints (future)
set breakpoints based on internal and/or external signals
monitor internal file registers
emulate full speed
program your device
trace lines of code or log variable/expression values

REAL ICE /3

REAL ICE vs. ICE 2000/4000 Emulators

The REAL ICE in-circuit emulator system is a next generation In-Circuit Emulator (ICE) system.

It differs from classical in-circuit emulator systems (e.g., MPLAB ICE 2000) in a single way: the production device and emulation device are the same.

This means that the actual device/emulated device differences are all but eliminated.

For example, speed bottlenecks caused by bringing internal busses off-chip and using external memories on classical emulator systems are eliminated by using the actual device for emulation.

Another significant benefit is that there is no time lag from when the device is released to when an emulator module to support the device can be released.

REAL ICE /4

REAL ICE vs. ICD 2

The REAL ICE in-circuit emulator system is similar to the MPLAB ICD 2 in-circuit debugger system, but surpasses it in speed and functionality.

Even with standard communication, the REAL ICE in-circuit emulator is faster than the MPLAB ICD 2; with the high-speed communication option, it is much faster.

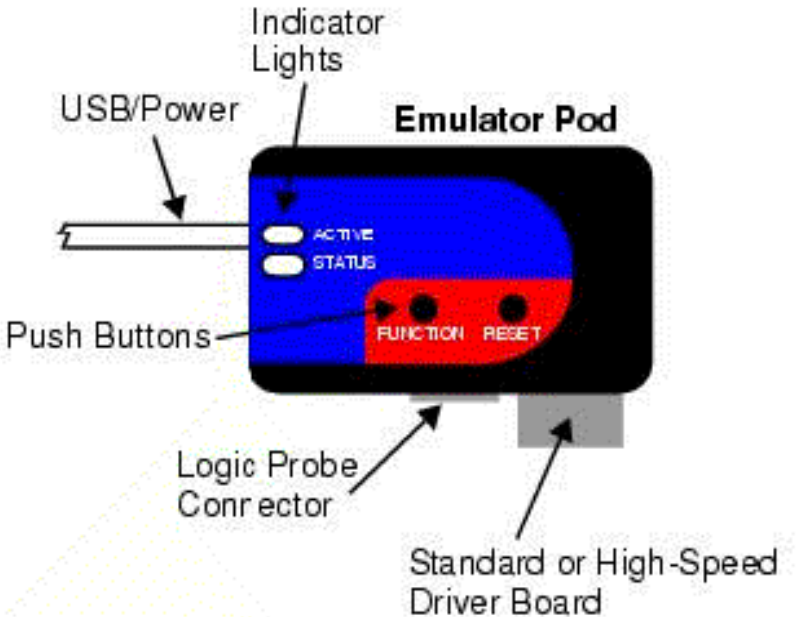
Also, in addition to basic debug functions, the MPLAB REAL ICE in-circuit emulator incorporates "emulation" functions, such as trace.

REAL ICE /5

System Configurations

The MPLAB REAL ICE in-circuit emulator system consists of these basic items:

- Emulator pod with indicator lights, push buttons and a logic probe connector
- USB cable to connect a PC to the emulator pod and power the pod
- Driver board and modular cable(s) to connect the emulator pod to an ICE header or target board



REAL ICE /6

Hardware specification: Emulator POD

This component has the interface processor (dsPIC DSC), the USB 2.0 interface capable of USB speeds of 480 Mb/sec, a Field Programmable Gate Array (FPGA) for general system control and increased communication throughput, an SRAM (1Mx8), for holding the program code image for programming into the emulation device on-board Flash, the external trigger logic, user interface push buttons and LED indicators.

The MPLAB REAL ICE in-circuit emulator system supports two types of interfaces to the target processor. They consist of the standard driver board and an optional high-speed driver board. These boards are inserted into the emulator pod via a card guide.

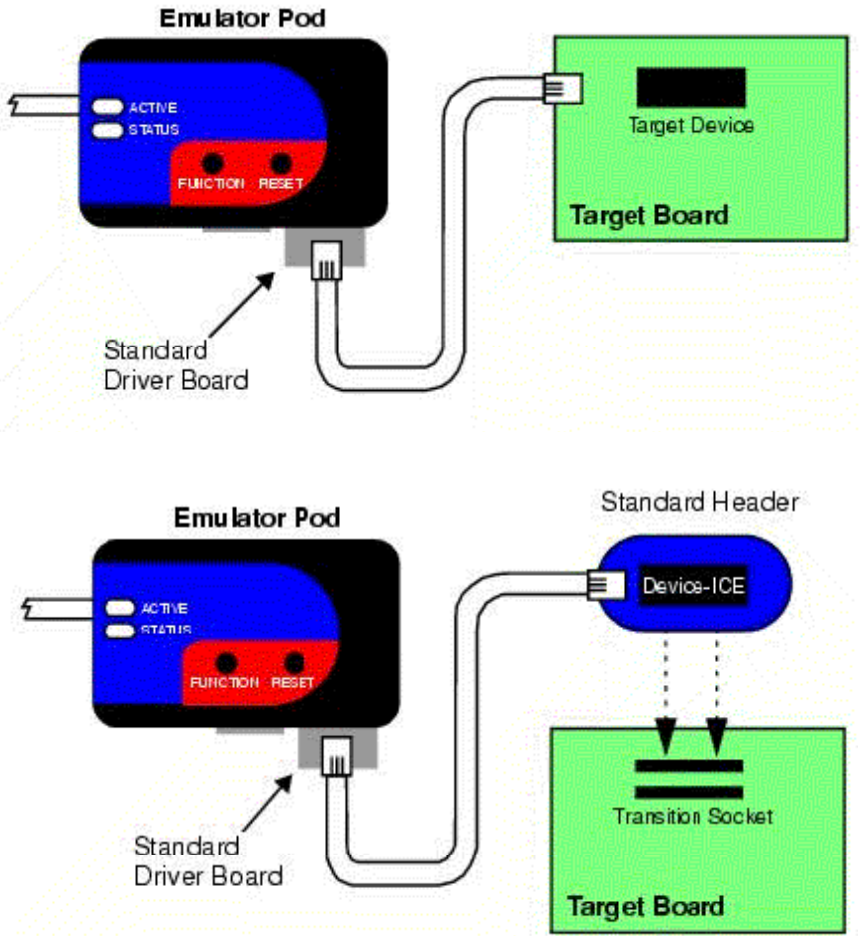
Durability/insertion life cycle of the card guide: 10,000 cycles

REAL ICE /7

Standard communications

The emulator system can be configured to use standard communication for both programming and debugging functions. This 6-pin connection is the same one used by the ICD 2, and provides the same amount of functionality.

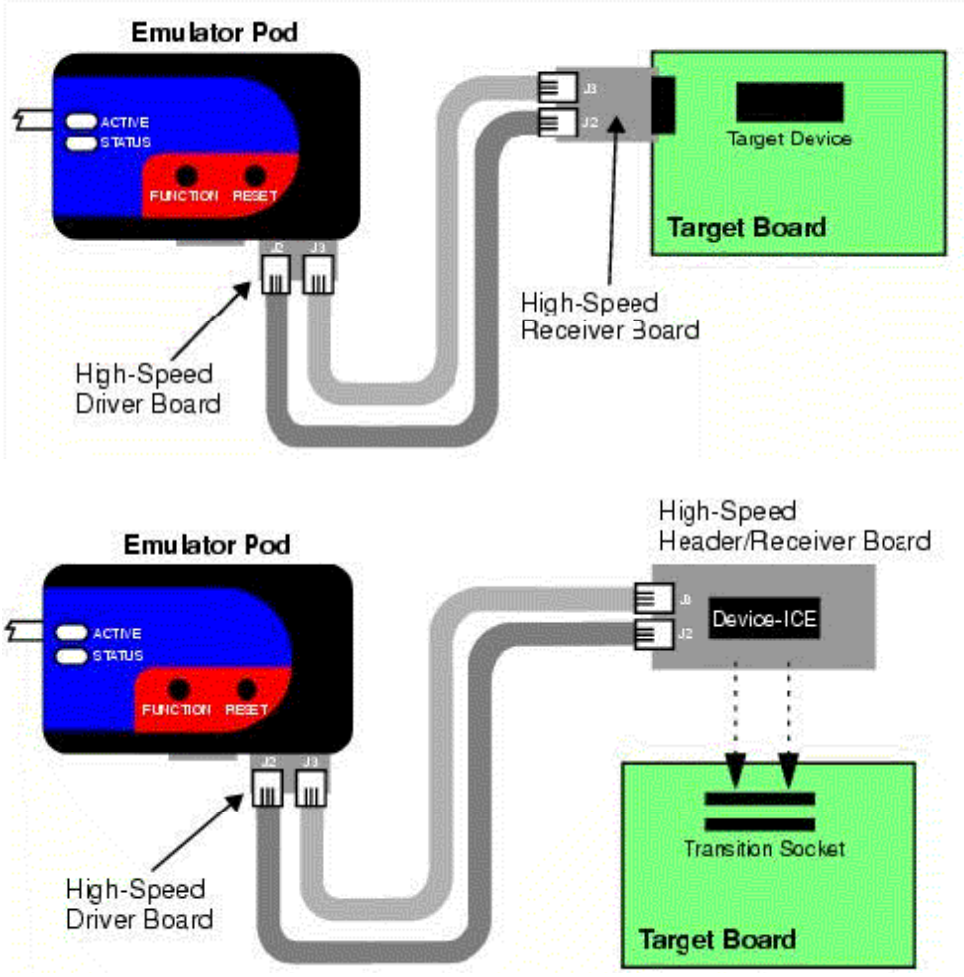
The standard driver board is plugged into the emulator pod to configure the system for this type of communication with the target. The modular cable can be either inserted into a matching socket at the target, where the target device is on the target board, or inserted into a header board, which is then plugged into the target board.



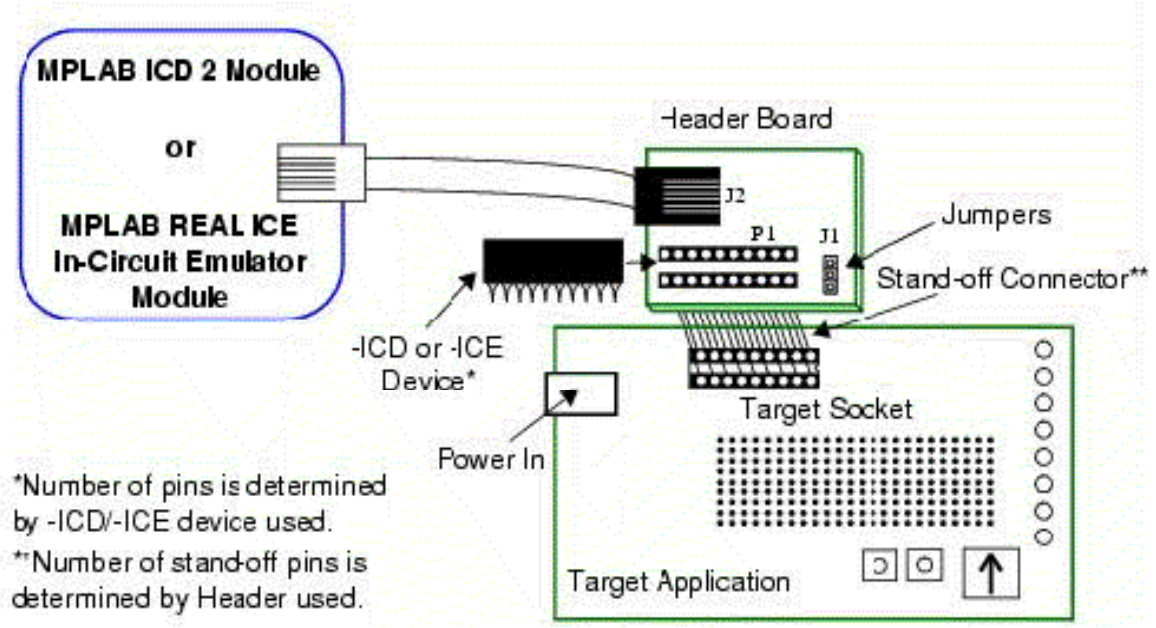
REAL ICE /8

High speed communications

The emulator system can be configured to use high-speed communication for both programming and debugging functions. With a high-speed driver/receiver board, this connection allows, for high-speed operations, a longer distance between the emulator and target and additional tracing functionality over a standard connection.



HEADER



A special ICD or ICE device is connected to a header board to be used with the ICD 2 in-circuit debugger or REAL ICE in-circuit emulator. This device is mounted on the top of a header and its signals are routed to the debugger or emulator connector. On the bottom of the header is a socket that is used to connect to the target board

PICSTART /1


Development Tools Overview
PICSTART Plus Programmer
- Supports all devices except 16C432/433
- Interfaces to MPLAB
- Adapter sockets for PLCC packages
- Connects to PC via RS-232 port
- Firmware upgrades available from www.microchip.com
PICSTART PLUS
DEVELOPMENT PROGRAMMER
8P
18P
28P
40P
MICROCHIP

PICSTART /2

Once you have code up and running through either the emulator or simulator, you will need to program your devices.

The PICSTART Plus is a very convenient universal desktop development programmer that supports all devices and also runs under the MPLAB IDE.

It has a 40-pin ZIF socket which supports DIP packages up to 40 pins directly; Microchip also offers 84 and 68 pin PLCC adapters for those larger pin count surface mount devices.

The PICSTART Plus connects to a PC through a standard RS-232 serial cable.

Firmware upgrades are available www.microchip.

The PICSTART Plus firmware is upgraded by first of all, obtaining a PIC17C44 device, plugging it into the ZIF socket and then programming the latest firmware into that device. Simply replacing the existing PIC17C44 with the newly programmed device enables you to do future firmware upgrades for free at the cost of your first PIC17C44 device.