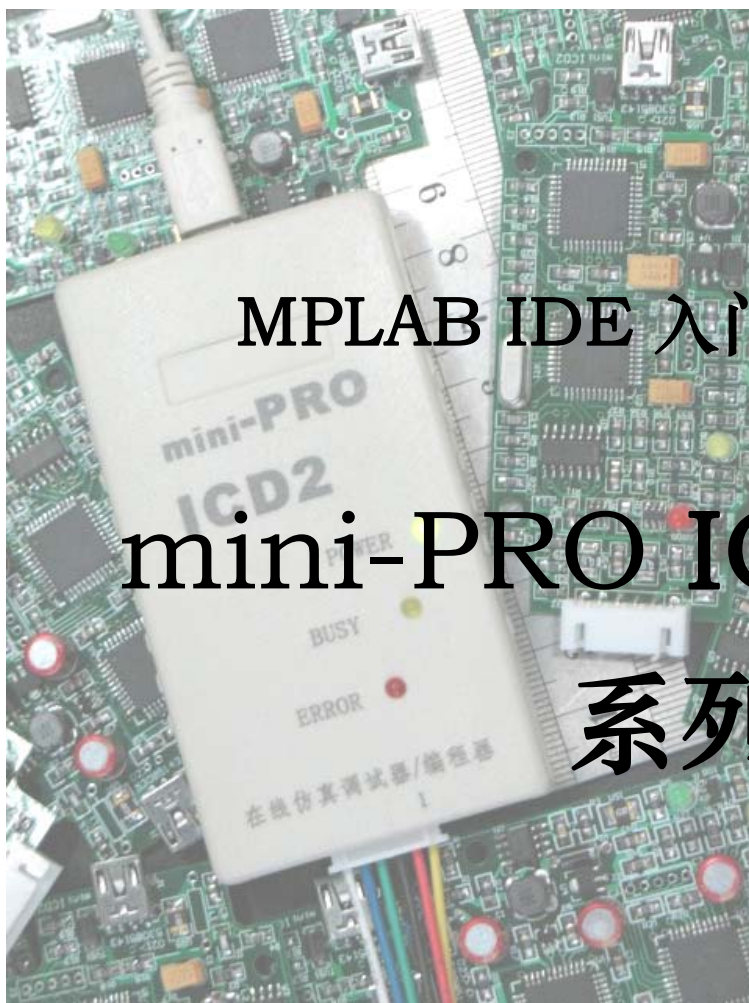


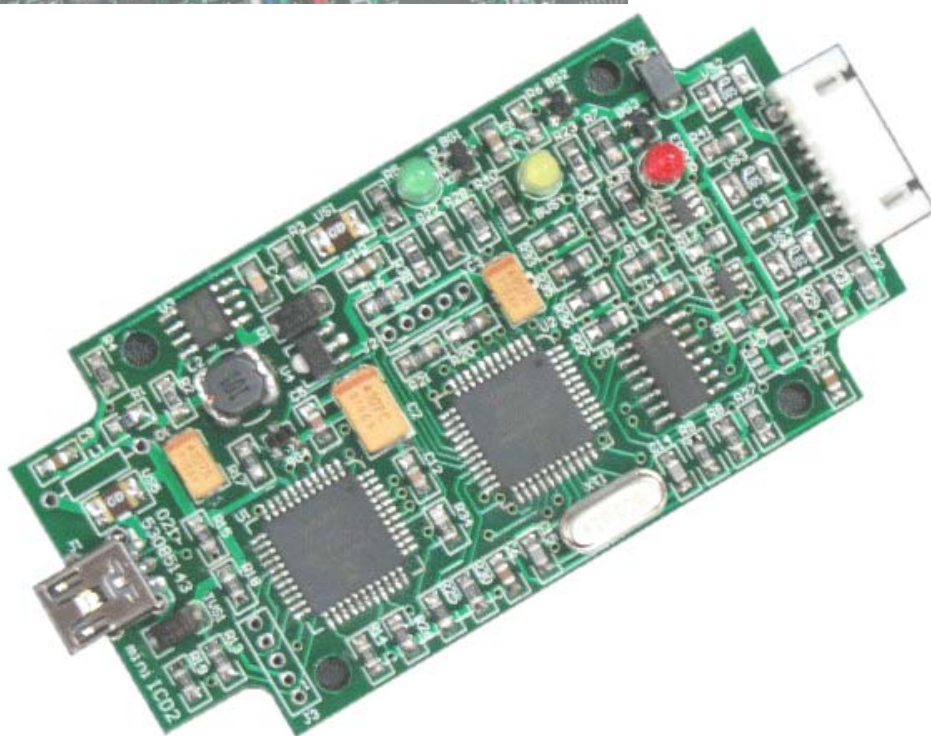


# MPLAB IDE 入门指导

# mini-PRO ICD2

# 系列说明书

编著 尤思胜



# 目 录

|                                  |    |
|----------------------------------|----|
| mini-PRO ICD2 在线仿真调试器/编程器简介..... | 2  |
| MPLAB ICD2 简介.....               | 3  |
| 在线调试器和在线仿真器比较.....               | 3  |
| 仿真调试功能的限制.....                   | 4  |
| 调试序保留的资源.....                    | 4  |
| 使用的软件.....                       | 4  |
| ICD2 支持的芯片型号.....                | 5  |
| ICSP 接口与芯片的连接.....               | 5  |
| 6 芯排线连接说明.....                   | 6  |
| ICD2 USB 驱动安装.....               | 6  |
| ICD2 串口设置.....                   | 7  |
| 选择芯片型号.....                      | 7  |
| 设置芯片 CONFIG(配置)字.....            | 7  |
| 使用 ICD2 作为调试器.....               | 8  |
| 1. 硬件连接.....                     | 8  |
| 2. 设置 ICD2 选项.....               | 8  |
| 2.1 选择工具类型.....                  | 8  |
| 2.2 设置通信方式.....                  | 8  |
| 2.3 设置编程选项.....                  | 8  |
| 2.4 设置电源.....                    | 9  |
| 2.5 工作状态及其它信息.....               | 9  |
| 3. 建立工程项目.....                   | 9  |
| 3.1 建立项目.....                    | 9  |
| 3.2 添加(删除)文件.....                | 10 |
| 4. 使用 ICD2 进行调试.....             | 10 |
| 4.1 打开源程序和变量观察窗口.....            | 10 |
| 4.2 软件模拟器的秒表功能.....              | 12 |
| 4.3 编译项目.....                    | 12 |
| 4.4 编程(烧写)目标芯片.....              | 12 |
| 4.5 运行及运行方式.....                 | 12 |
| 4.6 断点设置.....                    | 13 |
| MPLAB IDE 的主菜单.....              | 13 |
| 使用 ICD2 作为编程器.....               | 14 |
| 1. 硬件连接.....                     | 14 |
| 2. 选择工具类型.....                   | 14 |
| 3. ICD2 的编程设置.....               | 15 |
| 4. 导入导出烧写代码文件.....               | 15 |
| 5. 读写芯片操作.....                   | 15 |
| 项目实例.....                        | 16 |
| 常见问题解释.....                      | 17 |

## mini-PRO ICD2 在线仿真调试器/编程器

mini-PRO ICD2 是在 Microchip 最新版本 MPLAB-ICD2 框架上研发的在线仿真调试器/编程器,工作在 Microchip 免费提供的 MPLAB-IDE 集成开发环境,实现 ICD2 的全部功能,其主要功能特性有:



- ◆ 源程序编辑
- ◆ 直接在源程序界面调试
- ◆ 可设置一个断点
- ◆ 寄存器和变量观察
- ◆ 程序代码区观察
- ◆ 寄存器修改
- ◆ 单步调试
- ◆ 停止冻结芯片的运行(不包括定时器)
- ◆ 实时在线调试
- ◆ 可做编程器使用

根据多年生产和使用 ICD2 遇到的问题,mini-PRO ICD2 相对 MPLAB-ICD2,作以下几点改进:

- 去除 232 串口连接,改用单一高速 USB2.0 接口(PIC18F4550 高达 12Mbit/s)。
- 将 RJ45-6 接口改成 XH-6 接口,增强电源输出能力和连接可靠性。
- 直接由 USB 给主机和目标板(可控制)供电,无需外接电源,实现一线联机。
- 模拟开关 DG411 改为强驱动能力分立组合,解决以此引起的返修问题。
- 将 74HC126 输出驱动改为新型双电源驱动电路,增强通信驱动能力,真正支持目标器件电压范围 1.8 V - 5.5V,方便低压器件的烧写与调试。
- 增加完善的 PTC(自恢复)过电流输出保护和 TVS 过电压保护电路。
- 小巧轻便,外型尺寸为: 90x50x16mm,净重: 40g,适合外出携带。

除此之外,合理的结构布局和优化的 PCB 抗干扰设计,结合采用表面贴片工艺生产,使稳定性和可靠性大大提高,解决了 MPLAB-ICD2 使用过程中可能遇到的绝大多数问题。适合 PIC 单片机初学者作为学习工具使用,更是专业开发人员必备的 PIC 单片机开发利器。

上海湘渐电子科技有限公司

[www.youmcu.com](http://www.youmcu.com)

2007-09-01



本说明书仅针对 PIC ICD2、mini-PRO ICD2 的硬件部分及 MPLAB IDE 软件入门指导，配合使用的软件为 MPLAB IDE V7.xx 及以上版本。软件附在随机光盘中，也可到网站上下载最新版本。如需软件的更详细使用请参阅 MPLAB IDE 的中文使用说明书、MPASM 用户指南等。

## ◆ MPLAB ICD2 简介

MPLAB ICD2 简称 ICD2，是由美国 Microchip 公司设计的低价位 PIC 开发工具，本公司在此框架上通过改进设计，目前有 PIC ICD2 和 mini-PRO ICD2 二种产品，工作在 Microchip 免费提供的 MPLAB IDE 集成开发环境，实现 ICD2 的全部功能。其中 mini-PRO ICD2 去除了 232 串口连接，改用单一高速 USB2.0 接口，增加了完善的 PTC(自恢复)过电流输出保护和 TVS 过电压保护电路。并增强了内部电源输出供电能力。二种产品的外型如下：



## ◆ 在线调试器和在线仿真器比较

MPLAB ICD2 在线调试器是 ICE（在线仿真器，如 Picmate 系列和 ICE2000 系列等）的一个廉价替代品。这款低价位的 PIC 开发工具，利用 Flash 工艺芯片的程序区自读写功能，把监控程序连同调试目标代码一同写入待调试的芯片中，利用监控程序来实现仿真调试功能。当然 ICD2 也具备编程烧写功能，可作为**开发型编程器**使用。

ICD2 能完成以前只有在昂贵的 ICE 上才能实现的调试工作，但这也是以牺牲在线仿真器的一些功能和使用便利性为代价的。

与 ICE 在线仿真器相比，在线调试器在运行时有下列限制和要求：

- 仿真功能和调试手段受限制（详见下述）。
- 需要占用被仿真芯片的一些软硬件资源（详见下述）。
- MCLR/VPP 引脚为编程所共用。
- PGC 和 PGD 引脚被用于在线调试和编程。
- 这些被占用的芯片引脚连接有一定的限制。
- 目标芯片必须有一个能正常运行的时钟。
- 只能设置一个断点，每次修改程序编译后都必须对目标芯片重新烧写。

仿真手段的限制主要有以下几点：

- 单步运行不能响应中断。由于硬件的限制，ICD2 执行单步时不能跳转到中断入口处。
- 可以访问堆栈，但是不能观察其变化。
- MPLAB IDE 状态栏中显示的时钟频率与实际情况无关。实际时钟是由目标板上的振荡电路决定的，IDE 不能正确的体现它。
- 每次修改程序之后或通讯异常都需要重新对芯片进行编程。

调试功能的限制主要有以下几点:

- 调试时不能使用 SLEEP 指令, 即无法仿真睡眠功能。
- 调试时不能使用看门狗 WDT。
- 调试时不能开启芯片的代码保护 (不能烧写保密位)。
- 调试时不能开启低电压编程模式。
- 有**内部复位内部振荡**的芯片如 16F676 等, 不能同时使用内部复位, 内部振荡。一旦使用过一次, 下次 ICD2 再联接时有可能无法识别到目标芯片, 此时需要用 PIC 专用的编程器如 PICSTART-PLUS、PICPRO 等将其擦除后才可再次使用。

尽管 ICD2 与 ICE 相比有一些不足, 但它也有一些突出的**优点**:

- 在产品开发时, 只要保留 5 芯插针, 即可直接与 ICD2 相连, 而不需要取下单片机来插入 ICE 仿真头。在产品量产后, 可用该 5 芯插针对板上芯片进行编程。尤其对贴片封装的芯片特别方便, 在代码修改后 ICD2 可以在目标应用中对产品进行现场升级。
- 设备成本低, 调试环境 100%真实对应芯片的实际工作环境, 即使用了实际的振荡电路, 工作于实际的工作电压, 在调试时可随时将代码写入芯片进入实际工作状态。

**调试须保留的资源**

当使用 MPLAB ICD2 进行开发时, 必须要知道给器件调试用保留的资源。可通过帮助文件查看保留资源的详细说明。也可以在 ICD2 调试模式下, 打开数据 (File Registers) 寄存器、程序 (Program Memory) 存储器的观察窗口查看, **其中标注为 RR 的部分为保留单元不可用**。一般在线调试时需使用如下的片内资源:

- 复位入口处第一条指令不可用 (一般可写为 NOP 指令)。
- 一或两个堆栈单元被 MPLAB ICD2 调试器使用。
- 几个通用数据寄存器被保留用于在线调试。
- 程序存储器的高地址部分保留给调试器监控执行代码。
- PIC18FXXX 系列的影子堆栈在调试时不可用。目标程序可使用影子堆栈, 但不能调试快速中断返回或快速子程序调用返回, 因为 ICD 在获得一个断点时会覆盖影子堆栈。
- 当使用 MPLAB ICD2 调试 dsPIC 系列芯片时 0x800-0x84F RAM 空间无法使用。

而 ICE 则完全没有上述限制, 可提供强大的调试能力。如断点个数不限, 可支持多种条件断点, 有不同深度的各种跟踪方式, 具有多种事件的复杂触发功能等等。

在实际开发和调试时, ICE 仿真器内可提供存储器和时钟, 即使没有与目标应用板相连也可运行代码。而 ICD2 要求在实际应用调试时必须要有样片, 并需连接成具体应用电路, 在实际应用中如硬件运行不正常时将无法进行调试。这两点要求 (样片和硬件正常) 对具体应用来说有点不方便。但在学习阶段, 可以通过选用相应的实验板来解决。而对于项目开发来说也可以首先直接设计硬件电路。毕竟硬件设计是项目开发的基础, 也是关键的一步。

## ◆ 使用的软件

MPLAB ICD2 使用的软件平台是 Microchip 的 MPLAB IDE v7. xx 及以上 (建议使用最新版本) 集成开发环境, 兼容 Win98、WinNT、Win2000 和 Win XP 等操作系统。它包括源程序编辑器, 汇编语言 (可选 C 语言) 编译和链接器, 并可支持多种第三方程序语言编译链接器等。

因为 MPLAB IDE 集成环境是 Microchip 为其所有的开发工具研发的, 故使用 ICD2 前需要选择工具。又由于 ICD2 可以充当调试器和编程器, 所以在 debugger 和 Programmer 菜单中都可以进行选择。但必需**注意区分** ICD2 在这两种状态下的 Program 菜单和  按键所执行后编程结果是不同的。在 debugger 模式下, Program 是把目标代码和监控程序一起写入待调试的芯片中, 这个代码在离开 ICD2 后是不能单独运行的。而在 Programmer 模式下, 用 Program 写入目标代码的芯片可运行于实际应用中。而 ICD2 调试时所保留的资源, 可以让芯片全部正常使用。

## ◆ ICD2 支持的芯片型号

MPLAB ICD2 支持大部分 F 系列 PIC 芯片。具体支持的芯片型号，请参考帮助文件，或点击 MPLAB IDE 主菜单 Configure→Select Device 查看。目前 ICD2 不支持 FLASH 工艺（调试模式）的 PIC 芯片有：PIC16F7x、PIC16F84A、PIC16F54/57 等，但编程模式却是支持的。

此外，对于管脚数量较少（8~20pin）的芯片，尽管 ICD2 可以支持调试，但由于芯片的可用 I/O 本来就不多，ICD2 通信又占用了管脚，在使用中将不能有效地调试这些器件。这时 ICD2 需要配合仿真头使用，而不能直接通过 6 芯线进行调试。下表列出了常见仿真头型号：

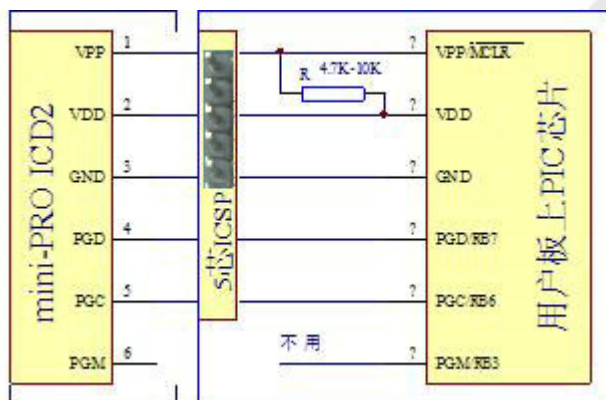
| 仿真头型号     | 可仿真的芯片型号                                | 备 注                                                   |
|-----------|-----------------------------------------|-------------------------------------------------------|
| AC162050* | PIC12F629/675                           | 仿真头上为专用芯片，不能用普通芯片替代；专用芯片其实为普通芯片+调试管脚，从而减少对 MCU 管腿的占用。 |
| AC162052* | PIC16F630/676                           |                                                       |
| AC162053* | PIC16F627A/628A/648A                    |                                                       |
| AC162054  | PIC16F716                               |                                                       |
| AC162055  | PIC16F684                               |                                                       |
| AC162056  | PIC16F688                               |                                                       |
| AC162057  | PIC12F635/PIC16F636                     |                                                       |
| AC162058  | PIC12F683                               |                                                       |
| AC162059  | PIC10F200/2/4/6、PIC12F508/509、PIC16F505 |                                                       |
| AC162066  | PIC16F639                               |                                                       |
| AC162060  | PIC16F785/HV785                         |                                                       |
| AC162061  | PIC16F685/687/689/690、PIC16F631/677     |                                                       |
| AC162070  | PIC10F220/222、PIC12F510、PIC16F506       |                                                       |

带\*的仿真头，必须使用内部振荡或外部有源振荡；MCLR 脚必须是高电平；当 GP1/RA1 为高电平时将无法进行调试；不能在对 GPIO 操作的指令上设置断点。

调试 PIC18FXXX 系列时：

- 设置的程序区必须 8 个 word 对齐。例如起始地址可以设置为 0x10，而不能是 0x12；结束地址可以是 0x1F 而不能是 0x14。。
- 对于 PICXX20，AVdd 和 AVss 引脚必须接上。

## ◆ ICSP 接口与芯片的连接



建议用 PIC 单片机做项目设计时尽量保留 ICSP 接口，特别是贴片封装的 Flash 型芯片，更有保留的必要。ICSP 接口在开发时可用于仿真调试，量产后可用此接口对芯片编程及产品现场升级。在硬件电路设计定型后，生产和软件开发完善可同步进行。ICSP-5 芯插针接口与芯片具体连接如图所示，插针可用 SIP2.54-5 封装，如需更小空间，也可用 SIP2.0-5 封装。具体应用设计时，必须注意以下几点：

- 复位引脚不可直接接入大电容，以免影响 Vpp 电压的快速建立。如确实需要在应用中需接入复位电路等，应在电路中串接隔离电阻。



- 为不影响串行时钟和数据的高速正确传输,PGC 和 PGD 引脚不应接有电容、电感等。
- PGC 和 PGD 引脚工作时需双向通信,不要在电路中串接二极管等影响通信的器件。
- 因 ICD2 内部对 PGC 和 PGD 引脚上已接有下拉电阻,所以不可在这二个引脚接上拉电阻,以免引起电平判断错误。

## ◆ 6 芯排线连接说明

6芯排线可以直接把ICD2连接到仿真头进行调试,如目标板上已预留有5芯插针(ICSP)接口,也可以直接把ICD2连接到目标板进行调试。水晶头RJ-45在使用中经常可出现接触不良等现象,为使连接更可靠,现将其改为XH-6和TJC8-6接口并将引线加粗,体积小却可大电流工作,具体外型及对应引脚说明如下:



1-黄=VPP 2-红=VCC 3-黑=GND 4-绿=PGD (一般为RB7) 5-兰=PGC (一般为RB6) 6-白=一般不用

## ◆ ICD2 USB驱动安装

Windows 经常会尝试安装默认的 USB 驱动,ICD2 不能使用默认的驱动。如果安装了错误的驱动,ICD2 工作将会出现问题。此时可查看设备管理器, Microchip 的 USB 驱动列在 USB 设备下。有时未知的 USB 设备会出现在“other devices (其它设备)”列表下。如果发现这样的设备,应将其删除,或参考 C:\Program Files\Microchip\MPLAB IDE\ICD2\Drivers\clnicd2.htm 根据其说明步骤进行更新或移除操作。然后再按以下步骤安装(如无发现问题或第一次使用直接按以下步骤安装)。安装步骤如下:

1、首先安装好 MPLAB IDE 软件。软件在光盘中(或下载最新版本),可按提示进行安装。



2、当一个新的 USB 设备第一次插入 Windows XP 或 98/ME 系统时,会出现一个对话框提示系统找到了新硬件设备。也可能出现对话框告知设备的数据库正在建立或更新。点击“下一步”继续安装。

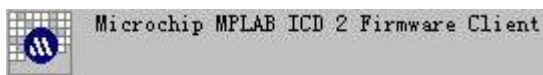
3、选择“搜索设备的最新驱动程序”,点击“下一步”。

4、如图所示,在“指定位置”前打勾,再点击“浏览”按钮来选择复制文件来源。在常规安装下,选择(取决 IDE 安装路径) C:\Program Files

\Microchip\MPLAB IDE\ICD2\Drivers 路径,这个路径包括以下文件: ICD2W98.INF、ICD2W2K.INF、ICD2W98.SYS、ICD2W2K.SYS、ICD2W98L.SYS、ICD2W2KL.SYS 等。

4、选择完成后,点击“下一步”继续。

5、此时可能会再次出现提示的系列对话框。系统会自动加载与设备通讯所需的 MPLAB ICD2 的 USB 客户端驱动。安装到最后一个对话框将提示“Microchip MPLAB ICD 2 Firmware Client”“Windows 已经安装了新硬件设备所需的软件”。点击“完成”来结束安装。



6、此时查看设备管理器,可看到如图所示设备,至此 MPLAB ICD2 USB 就可以正常使用了。

## ◆ ICD2 串口设置

MPLAB IDE 的默认连接方式是 USB, 如果不做处理, 每次启动软件, 都默认为 USB 方式连接。要更改默认连接为串口方式, 请按以下步骤操作:

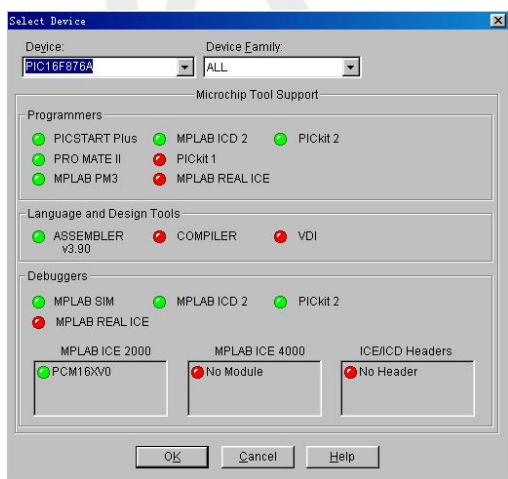
打开 MPLAB IDE 调试器或者编程器选项, 例: 用调试器设定时用 Debugger→Select Tool 选择 MPLAB ICD2。然后通过 Debugger→MPLAB ICD2 Setup Wizard 按提示进行设置, 设置一次后即可将默认通讯方式改为串口。

在设定采用串口方式连接时, 还应做如下设置, 以保证连接的可靠性。

打开“设备管理器”(可以通过右击**我的电脑**→**属性**→**硬件**打开设备管理器), 在“端口 (COM 和 LPT)”一栏中选择相应的 COM 口, 打开其属性对话框 (双击或右键选择), 在“端口设置”页面。进行如下设置。

- 流控制: 选择为“硬件”方式。
- 点击“高级”按钮, 把“使用 FIFO 缓冲区 (...)”前面的那个勾去掉, 即不使用 FIFO。设置完点击“确定”保存设置, 为保证设置可靠有效, 最好重新启动电脑。一旦设置好, 下次使用则无需再设置。

## ◆ 选择芯片型号



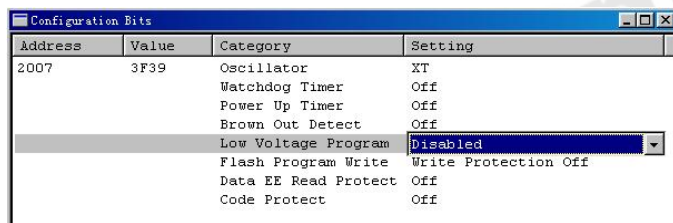
MPLAB IDE 软件运行时, 首先需要选择调试芯片的型号 (不同系列的芯片其调试支持工具是不一样的), 点击主菜单的 Configure→Select Device, 打开如图所示窗口, 在 Device 的下拉菜单里选择需要调试的芯片型号。

选择了相应的芯片型号之后, 可在下面的窗口“Microchip Tool Support”栏, “Programmer”框里和“Debugger”框里看到各种 Microchip 编程工具和调试工具对该芯片的支持情况, 其中:

- 1) 红点表示该工具目前还不支持当前芯片;
- 2) 绿点表示该工具可以支持当前芯片;
- 3) 黄点表示该工具目前对当前芯片还处在测试

状态; 在选择完芯片型号之后点击“OK”确认。

## ◆ 设置芯片 CONFIG(配置)字



要保证芯片能正确地工作还需要设置相应的 CONFIG 字, 点击主菜单的 Configure → Configuration Bits, 弹出一个如图所示的“Configuration Bits”窗口。可在这个窗口里进行相应的配置字设置。

点击“Setting”列下面的相应选项之后将会出现一个下拉菜单, 可以在这个菜单里面选择相应的设置。请注意在使用 ICD2 作为调试器的时候, 需要关闭“Low Voltage Program”(低电压编程)选项, 以及关闭“Code Protect”(代码保护)选项。

PIC 单片机的配置字极为关键, 必须保证芯片配置字的正确设定, 否则 ICD2 可能根本无法正常运行。在调试模式可出现错误提示, 编程模式将代码写入芯片后则不能正常工作。

请务必养成良好习惯, **直接把配置字信息写在源程序当中**, 这样经编译生成的 hex 烧写文



件，在导入时（ICD2 调试、编程模式、及其它编程器载入）就包含有配置字信息，不必每次在上图中进行设定，更可避免不必要的错误。

## ◆ 使用 ICD2 作为调试器

### 1. 硬件连接

硬件的正确连接步骤是：

- 1) 用 USB 线或 232 串口线连接 PC 和 ICD2 主机，
- 2) 用 6 芯线连接 ICD2 主机和仿真头，
- 3) 将仿真头插到目标板上，
- 4) 确认 ICD2 的电源设置后接上 ICD2 电源或给目标板上电，

**注：**如不使用仿真头，而是通过目标板预留的 ICSP 接口进行调试，那么步骤 2) 和 3) 可合并为：用 6 芯线连接 ICD2 主机和目标板的调试接口。

### 2. 设置 ICD2 选项

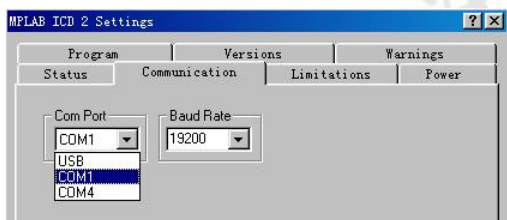
#### 2.1 选择工具类型

如果要选择 ICD2 作为调试工具，那么需要在 MPLAB IDE 中进行工具选择设置。点击主菜单的 Debugger→Select Tool，选择为 MPLAB ICD2。

这时系统将会提示 “ICDWarn0030: ICD2 is about to download a new operating system. ....Downloading Operating System”。这是系统在下载配合所选择调试芯片使用的硬件操作系统，需等待一段时间以便完成此操作。

当硬件操作系统下载完成之后，系统将会检查 ICD2 的连接情况。如果提示 “Target Device \*\*\*\* found” 那么一切正常；如果提示 “ICDWarn0020:Invalid target device id ....” 那么请检查电源设置以及 6 芯电缆的连接情况。

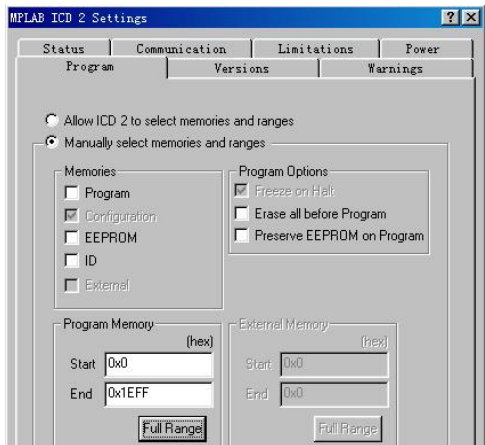
#### 2.2 设置通信方式



如果连接了 ICD2 的 USB 线，并安装了相应的 USB 驱动程序，那么 MPLAB IDE 将默认 ICD2 的通信方式为 USB 接口连接方式。需要更改通信方式，就要进行重新设置。

点击主菜单的 Debugger→Settings，系统将弹出一个 MPLAB ICD2 Settings 窗口（如图所示）。选择 “Communication” 页，在 “Com Port” 栏里选择相应的 COM 口，然后在 “Baud Rate” 栏里选择相应的通讯波特率。设置完成后点击 “应用” 按钮保存设置。

#### 2.3 设置编程选项



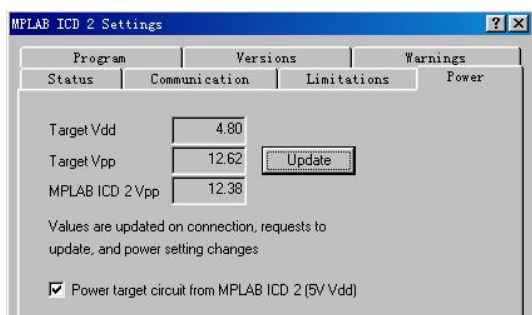
在 MPLAB ICD2 Settings 窗口选择 “Program” 页设置相应的编程选项。一般情况下（默认）可选择 Allow ICD 2 to select memories and ranges 项，即在此选项前打勾，系统会 **自动计算选择编程地址范围**。

如需手动设置编程选项应选择 Manually select memories and ranges，在 “Memories” 栏目里相应的选项之前打上勾，比如在 “ID” 前打勾，在编程时将会写入 ID 码。

也可以设置编程时程序的起始和终止地址，在 “Program Memory”（程序存储器地址）栏目里设置相应的选项。在设置时请注意开始字节不能超过 0x10，而结束地址则需要符合 ICD2 的编程要求为监控程序模块预留 255 个语句的空间。合理设置起始

和结束地址可以增加编程的速度。设置完成后点击“应用”按钮保存设置。

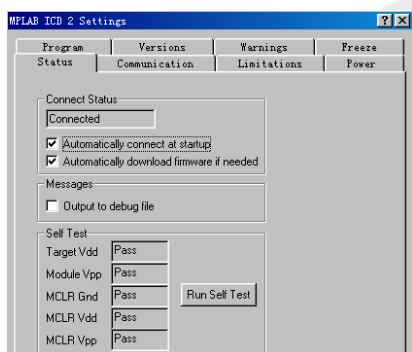
## 2.4 设置电源



如果目标板上没有电源，则需要选择由 ICD2 提供电源的方式。在 MPLAB ICD2 Settings 窗口的“Power”中设置，如图所示。Pro-ICD2 设计有较大的驱动能力，**需注意的是 USB 口及延长线等的负载能力**。如台式机不要将延长线插在面板前的 USB 口上，而应插入主板即机箱后的 USB 口。

如果目标板上已经有电源了，不需要由 ICD2 主机来提供电源，那么把“Power target circuit from MPLAB ICD2”（从 ICD2 供电）前面的勾去掉即可。点按“Update”时可以刷新当前目标板上的 Vdd、Vpp 以及主机提供的编程电压情况。设置完成后点击“应用”按钮保存设置。

## 2.5 工作状况



在“Status”栏目中可显示当前状态。在此栏目中，勾选“Automatically connect at startup”项，在启动 ICD2 时**自动与目标系统联机**。

勾选“Automatically download firmware if needed”项，**自动下载更新 ICD2 内部操作系统**。针对不同 PIC 芯片 ICD2 内部操作系统的不同，选用此功能后，在 ICD2 与目标系统联机时，如发现不同的操作系统，就会在相关安装目录下查找正确的驱动文件，并下载到 ICD2 进行自动更新。

点击“Run Self Test”可执行自检。自检正常时相应项显示 Pass，否则应根据出错项检查 ICD2 硬件状况。设置完成后点击“应用”按钮保存设置。

## 2.6 其它信息

还可以在 MPLAB ICD2 Settings 窗口面了解到调试器其它相关的信息：

- “Warning”栏目：显示警告信息，在这个栏目里决定是否打开相关的警告信息；
- “Limitation”栏目：从这里可以了解到 ICD2 的一些使用限制信息；
- “Versions”栏目：版本信息；

当 MPLAB ICD2 Settings 窗口的所有项目设置完成后，如果点击“确定”退出设置，则系统将会保存所有当前的设置；如果点击 MPLAB ICD2 Settings 窗口右上方的“×”来退出设置，那么所有未“应用”保存的设置将不被保存。

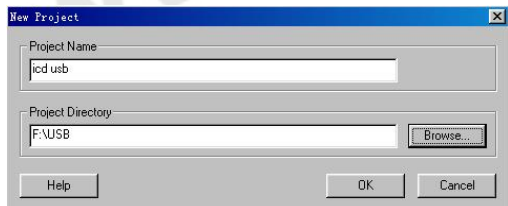
## 3. 建立工程项目

### 3.1 建立项目

在对源程序进行调试之前首先需要建立一个项目，这样 MPLAB IDE 就能把相关的调试信息包含进去。**特别提示：项目存放路径、项目名称、源程序名，全部不支持中文 !!!**

建议将所有**调试所需的文件放在同一个目录里**，以便于调试管理。建立项目前创建源程序，点击主菜单的 Files→New，弹出一个空白页写上程序代码后保存为“\*.asm”即可。

由于 MPLAB IDE 自带的文本编辑器使用存在许多不便，也可使用记事本或其它所熟悉的文本编辑器对源程序进行新建、编辑、修改，确认后可保存为“\*.asm”或“\*.c”文件。



点击主菜单的 Project→New，来建立新项目。在弹出的“New Project”（新项目）窗口里面的“Project Name”（项目名称）栏里输入相应的项目名，然后使用“Browse”（浏览）按钮来选择源程序保存的路径。如图所示，点击“OK”退出，这时将

在所选择的项目路径里生成一个 (\*.mcw) 的文件。

点击主菜单的 View, 在 Project 和 Output 二项前打勾, 打开 “\*.mcw” 和 “Output” 窗口。

### 3.2 添加 (删除) 文件



把调试所需的文件加入到项目里。在主界面中 “\*.mcw” 的窗口里可看到, “\*.mcw” 的文件夹树下有六个分支, 分别是 “Source Files” 源文件、“Header Files” 头文件、“Object Files” 目标文件及 “Library Files” 库文件、“Linker Script” 链接描述、“Other

Files” 其它文件, 如图所示。其中源文件夹里是必须添加源程序文件的, 以下几个文件夹可以不添加文件。

可以点击相应分支文件夹并通过右键菜单 “Add Files” 来添加文件, 这些文件通常被称之为节点文件, 是在调试过程需要调用的。添加源程序后的结果如左图所示。

同样, 也可以在选定的文件上使用右键菜单来删除文件。在删除文件之前请确认此文件已经不再被该项目使用。

## 4. 使用 ICD2 进行调试

### 4.1 打开源程序和变量观察窗口

双击已添加的源程序可打开源程序 (\*.asm) 编辑窗口, 在这里可对源程序进行编辑调试。IDE 还提供了多种变量观察方式, 可使用主菜单 View 选项打开或关闭这些窗口。可使用的观察变量窗口有: 数据寄存器 “File Registers” 观察窗口、特殊功能寄存器 “Special Function Registers” 观察窗口、自选变量 “Watch” 观察窗口、以及程序存储器 “Program Memory”、硬件堆栈 “Hardware Stack” 和 EEPROM 等窗口。

数据寄存器 “File Registers” 观察窗口, 是一个全面的除 W 外所有片内寄存器内容的观察窗口, 但由于数据量太大, 观察时反而显得不直观, 且明显影响调试速度, 所以可不开启。

特殊功能寄存器 “Special Function Registers” 观察窗口, 芯片内所有特殊功能寄存器内容观察窗口, 如关心的寄存器内容不多, 也可关闭此窗口, 单独使用下面窗口。

在调试运行过程中, 可能只需要观察一些特定少量的寄存器值。为了使用方便, 可以把一些



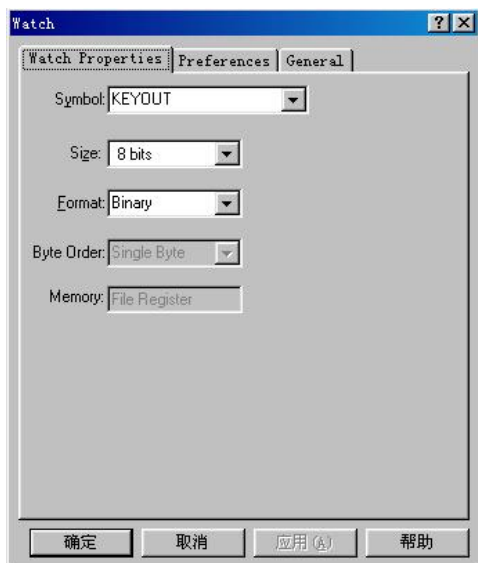
经常要查看的寄存器添加到 **自选变量观察窗口**。使用主菜单命令 View→Watch, 系统将会弹出一个 “Watch” (自选变量) 窗口, 此窗口可将特殊功能寄存器、源程序中自定义的变量 (全部数据寄存器) 随意添加后进行观察, 当然也可随意将变量删除。如图所示, 其中的红色行表示程序运行后变量内容相对已经改变, 以提醒引起注意。

在窗口的 “Add SFR” 按钮后面的下拉菜单中显示的是特殊功能寄存器的名称, 可从下拉菜单中选取后并使用 “Add SFR” 按钮, 将需要观察的特殊功能寄存器添加到自选变量列表中。同样的, 也可以在 “Add Symbol” 按钮后面的下拉菜单中选取自定义的变量, 并使用该按钮将所选变量添加到自选变量列表中。

为了观察方便, 还可以利用自选变量窗口中的 “Watch1” ~ “Watch4” 4 个子窗口分类添加自选变量组合, 以适应不同程序段调试时的观察。在自选变量窗口点击右键, 选择相应菜单项可随意增加、删除、改名子窗口等操作。



使用自选变量窗口的最大特点是：可随时改变观察窗内各观察项的**数据显示格式**，这是个非常实用的功能，如在调试过程中，可将延时寄存器、加、减、乘、除等计算结果变量设为十进制显示，位操作结果变量及 I/O 口设为二进制显示，通信数据可设为十六进制显示等。



在自选变量窗口中点击右键，选择“Properties”打开“Watch”窗口，再选择“Watch Properties”页可进行数据显示格式设定，如图所示。

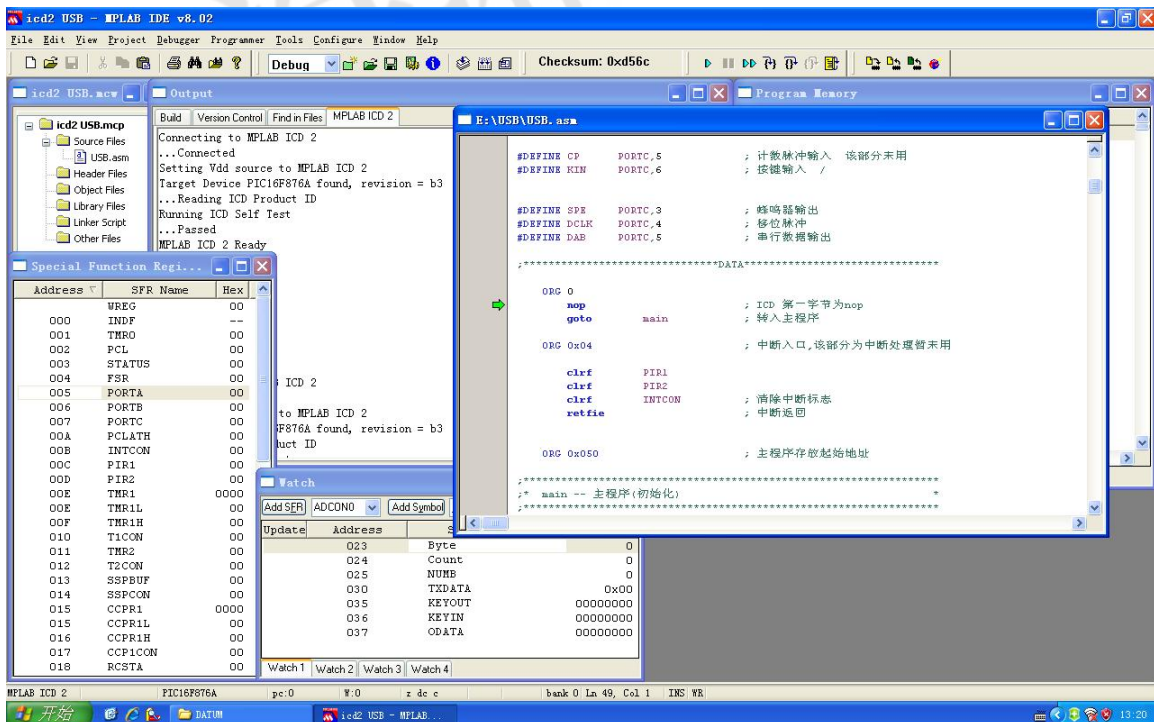
首先在“Symbol”下拉项内选择需设定的观察变量。在“Size”下拉项内选择字长（8 位=1 字节、16 位=2 字节、24 位=3 字节、32 位=4 字节），对于多字节构成的变量，还可在“Byte Order”下拉项内设定字节的高低显示顺序。再在“Format”下拉项内选择显示格式，Hex 十六进制，Binary 二进制，Decimal 十进制，ASCII 码格式，MCHP Float 特定浮点数格式，IEEE Float 标准浮点数格式，Single Bit 寄存器单个位显示格式。选择后点击“确定”保存。

所有观察窗口内的显示数据不仅可供观察，也可通过鼠标**左键双击其中某个显示数据**，即可对其显示内容直接进行修改。

以上的所有设置都被保存在项目文件里，以后如要用 ICD2 对原项目进行调试，可直接**点击主菜单 Project→Open** 命令，选择项目后打开即可进入如下图所示的调试界面。

任何时候各种原因引起的通信中断，可用主菜单的 Debugger→Connect 命令，或点击按钮使 ICD2 重新联接。

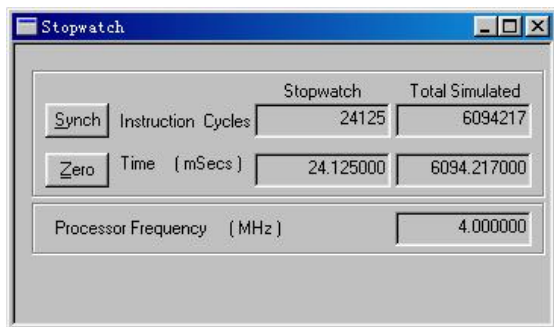
选择了几个常用观察窗口后的实际工作界面如下图（供参考）所示。在退出 MPLAB IDE 时如在跳出的保存提示窗口点“是”，工作界面的观察窗口等内容将被保存。



主界面及变量观察窗口

## 4.2 软件模拟器的秒表功能

秒表“Stopwatch”功能是软件模拟器的特色，也是其它硬件调试工具所不具备的，它可以精确地显示程序实际运行的指令周期，并根据确定的时钟频率转换成相应的运行时间同步显示。尽管 ICD2 调试器此项功能不可用，但可以作为辅助调试手段配合 ICD2 使用，对需要有精确时间要求的程序调试带来极大方便。



要在 ICD2 调试程序时使用秒表功能，点击主菜单的 Debugger→Select Tool，选择 MPLAB SIM 可从 ICD2 调试模式转为软件模拟器，再用主菜单的 Debugger→StopWatch 打开秒表窗口，如图所示。图中“Instruction Cycles”行显示指令周期，“Time”行显示根据时钟频率转换成的相应时间数。“Synch”为同步按钮，“Zero”

为清零按钮。“Total Simulated”列显示的是累计值，也即软件模拟器启动运行至当前的全部指令值，按同步按钮可将该值复制到秒表值里。“Stopwatch”列显示的是秒表值，按清零按钮后可使该值变为零，程序继续运行时秒表值从新开始计时。

秒表可以精确地测定程序运行的时间，如要测定一个延时子程序的运行时间，可在“CALL”指令上设置断点或用“Run To Cursor”命令让程序运行到该行时停下，按清零按钮使秒表值为零。然后用  单步跨过运行或用“Run To Cursor”命令让程序运行到下一行时停下，此时秒表所显示的值就是该延时子程序的运行时间（包括 CALL 和 RETURN 指令执行时间）。

“Processor Frequency”行显示的是时钟频率，点击主菜单的 Debugger→Settings 系统可弹出一个“Simulator Settings”窗口，在“Osc/Trace”页可对秒表的时钟进行修改，以对应于调试芯片的时钟频率。

点击主菜单的 Debugger→Select Tool，选择 MPLAB ICD2 可回到 ICD2 调试模式，此时秒表窗口将自行关闭，再次使用软件模拟器时秒表窗口也会自动开启。

## 4.3 编译项目

在调试前和对源程序修改后都要执行项目编译操作，以便生成相应的调试文件。点击主菜单的 Project→Build All，或者使用工具栏上的  快捷图标来编译整个项目。

编译的结果在“Output”窗口的“Build”页显示。如果编译发现了错误，则不能生成调试所需的 (\*.hex) 文件，这将不能继续进行下一步。要根据错误提示（在错误提示行 [双击鼠标可直接使光标跳到对应的错误源程序行](#)）修改程序后，重新编译，直至提示编译成功通过。

## 4.4 编程（烧写）目标芯片

调试前一定要将 [最新编译的代码下载到目标芯片](#)。当“Output”窗口的“Build”页中提示编译成功之后，编译过程生成了相应的 (\*.hex) 文件。此时可使用工具栏上的  快捷图标对目标芯片进行编程操作，在编程时需要注意芯片的 CONFIG 字设置，以及编程区域设置。

如果编程过程中出现“校验失败”等提示，请检查芯片的 MCLR、PGD、PGC 引脚连接电路。

## 4.5 运行及运行方式

在编程成功之后就可以进行调试了。ICD2 提供了多种运行及调试的手段。综合运用这些运行调试手段可以比较方便的验证程序的功能，并发现其中存在的缺陷和错误。

按工具栏上的快捷图标的顺序由左至右，ICD2 运行的方式有：

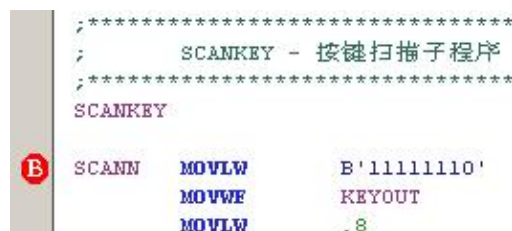
-  “Run”：全速运行，如果运行过程中遇到了断点将会停止。使用这个运行方式可以直观的观察程序的运行效果，并整体的验证其功能。
-  “Halt”：停止运行。请注意，由于 ICD2 是通过使用 Flash 工艺芯片的自编程功能来实现调试功能的，在执行此功能时，并不代表芯片已经实际停止了运行，而只是不再执行当前程序的功能了。此时如果发生了掉电或者干扰导致芯片与 ICD2 主机之间发

生了通信错误，则需要重新执行“联接→编程”的操作。

-  “Animate”：单步连续运行。每执行完一条指令，更新相关观察窗口的数据内容后，再继续自动下一条指令的运行。
-  “Step Into”：单步运行。使用单步的功能可以观察到当前语句执行的效果。利用单步功能可以实现多种调试目的，例如，可以使用该功能来观察程序分支的跳转方向，或者用来观察数据的转移、运算等。
-  “Step Over”：单步跨过运行。使用这个功能时系统将跳过子程序的调用过程，直接执行到调用的下一语句，这样在遇到“CALL”语句时就可以利用这个功能直接看到调用的结果了。假如该语句调用的是没有必要进行分析的延时子程序，那么使用此功能就可以避免单步运行时烦琐的循环操作过程。
-  “Step Out”：单步跨出运行。指令运行停止于子程序内部时，用跨出运行方式可以使子程序剩余指令一次执行完成，停在 CALL 后下一条指令处。**ICD2 该命令不可用。**
-  “Reset”：复位运行。使芯片复位，对于汇编源程序来说，执行后程序 PC 将指向芯片的复位地址；对于 C 语言源程序来说，执行此功能后程序 PC 则是指向 main() 函数首地址。

此外工具栏上还有读写操作的快捷图标 ，具体使用参看下面使用 ICD2 作为编程器部分（5. 读写芯片操作）介绍。

#### 4.6 断点设置



断点功能是在调试中经常使用的一个非常重要的调试手段，这项功能使芯片在运行到相应的程序语句时停止下来，可以根据目标板的运行情况或者相关 RAM 变量的值来分析运行的情况。**ICD2 只能提供一个断点。**设置时只需在设置断点的源程序行双击即可，如上图所示，其中红底的 B 就是断点位置。再双击该行可取消断点的设置。

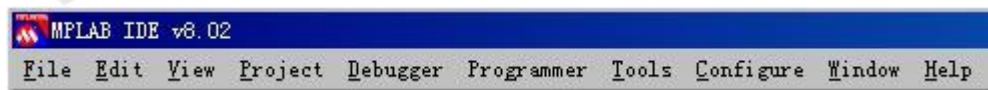
此外在源程序 (\*.asm) 调试窗口点击右键，可弹出批量命令菜单，如左图所示，在这里可实现禁止、使能、删除所有断点的操作。对已设有太多断点而搞不清具体位置时，可选“Breakpoints”下面的“Remove All...”命令删除所有断点。

左图所示中还有二个常用而方便的命令。“Run To Cursor”命令是让程序全速运行，直到当前光标所在的指令行停下，该命令可省去频繁设定和取消断点带来的麻烦，非常适用需要偶尔停一下看看运行结果的地方。

“Set PC at Cursor”命令是直接让程序指针强行跳转到当前光标所在的指令行，注意不是通过运行而使光标停在此指令行，该命令对局部指令代码调试非常适用。

此外点击“Properties”命令，可弹出“Editor Options”窗口，在该窗口可对文本编辑内容进行各种设定，如文本的字体、源程序语法颜色的区分、鼠标双击设定断点...等。

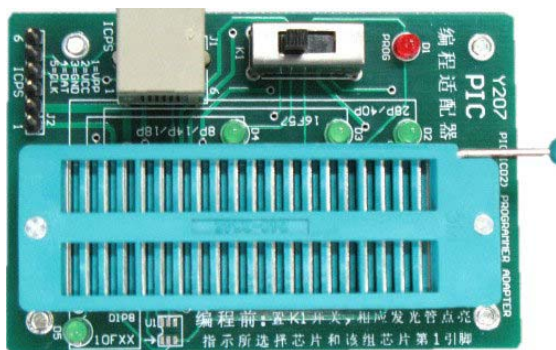
#### ◆ MPLAB IDE 的主菜单



上图是主菜单，在以上介绍的（如 Project→New 等...）表明需在下拉菜单中打开。



## ◆ 使用 ICD2 作为编程器



ICD2 编程“Programmer”模式支持的器件，比调试“debugger”模式支持的器件更多，所以非常适合用 ICD2 作为**开发型的编程器**使用。对于已预留有 5 芯插针（ICSP）接口的实际应用产品和实验目标板等，可直接与 5 芯插针接口连接烧写，如果需要对空的芯片（如 DIP 封装等）编程，还必须用 **Y207-PIC 编程适配器**（对应微芯公司 AC162049 通用编程模块）配合使用，适配器外型如图所示。

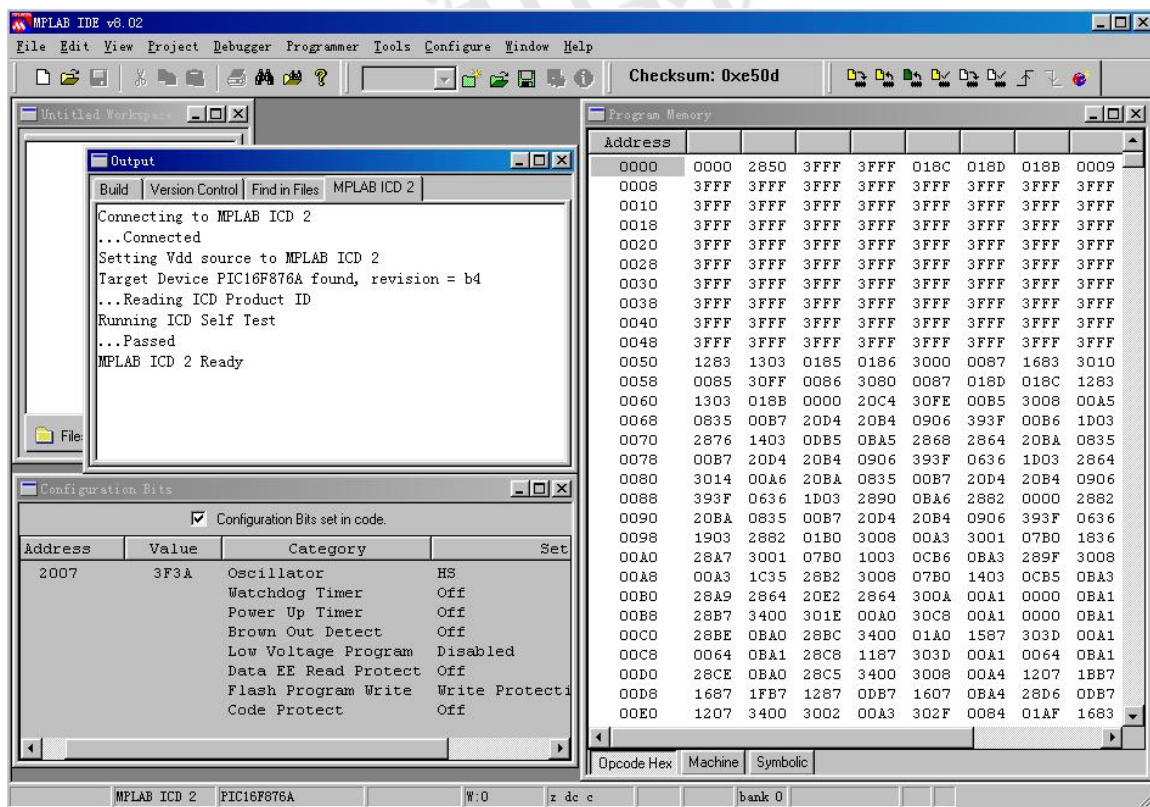
Y207-PIC 编程适配器是 ICD2 的最佳配套工具，可与 ICD2/kit2/you-K182 等使用 ICSP 接口的编程器配合，直接烧写 40P 以下的 PIC 系列直插芯片（包括 10Fxxx 以及 16F57 等）。通过切换开关选择芯片并由 LED 指示所选芯片第 1 引脚，非常方便直观。

### 1. 硬件连接

用 USB 线或 232 串口线连接 PC 和 ICD2 主机，用 6 芯线连接 ICD2 主机和 Y207-PIC 编程适配器，根据所选择的芯片型号，拨动编程适配器上的开关，相应的发光管点亮，指示所选择芯片和该组芯片的第 1 引脚。

### 2. 选择工具类型

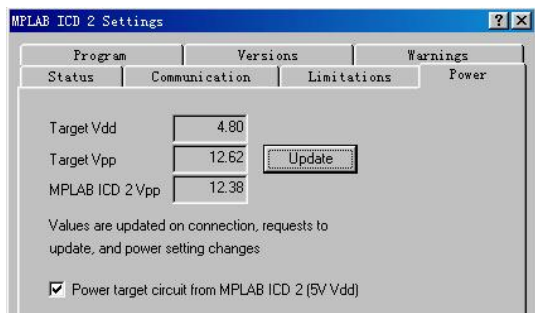
如果要选择 ICD2 作为编程器使用，那么需要在 MPLAB IDE 中进行工具选择设置。点击主菜单的 Programmer→Select Programmer，选择为 MPLAB ICD2。使用主菜单 View 选项打开程序存储器“Program Memory”窗口，点击主菜单 Configure → Configuration Bits 打开配置字窗口，实际工作界面如下图（供参考）所示。



编程器工作界面

### 3. ICD2 的编程设置

点击主菜单 Programmer→Settings 可打开 MPLAB ICD2 Settings 窗口, 在该窗口中, 可完成对 ICD2 编程模式的具体设置, 先选择“Power”来设置电源选项, 在作为编程器使用时必须



把“Power target circuit from MPLAB ICD2”前面的勾选上, 即选择由 ICD2 提供电源的方式。如图所示。

其它选项设置, 如芯片选择、通信方式、编程选项、(CONFIG)配置字等设置, 可参考使用 ICD2 作为调试器部分的说明, 在调试“debugger”模式下的所有设置, 在这里全部有效。设置完成后点击“确定”或“应用”按钮保存设置。

除非有特殊需要, 最好将“Low Voltage Program”(低电压编程)选项关闭。如果选择开启了这个选项, 那么芯片将会在特定条件下启动低电压烧写模式(请参看芯片的烧写资料)。

### 4. 导入导出烧写代码文件

在对芯片烧写之前, 首先将需要烧写的代码文件(\*.hex)导入到内存中, 导入后在程序存储器“Program Memory”窗口中能看到文件内容。

使用主菜单 File→Import 命令来导入需要烧写的(\*.hex)文件。如果希望从芯片中读入代码文件, 那么可以使用主菜单 Programmer→Read 或按 快捷图标将代码读入到内存中, 读入之后根据实际应用情况设置 CONFIG 字, 即可对芯片进行烧写。

将内存中的数据导出保存。使用主菜单 File→Export 命令, 在弹出的“Export Hex File”窗口中选择好相应项(勾选的项将被同时保存), 按“确定”按钮, 接下来选择文件保存的路径和文件名后按“保存”按钮, 即可对内存中的数据保存为(\*.hex)文件。

### 5. 读写芯片操作

可以使用主菜单的“Programmer”子菜单下的烧写相关命令, 也可以使用工具栏的快捷图标来进行相关操作。工具栏的快捷图标从左到右依次对应的主菜单操作是:

- “Program”: 烧写芯片。内存中数据写入芯片。
- “Read”: 读取芯片内容。**注意:** 如果芯片已加密, 读出的则是不正确无效代码。
- “Reads EEPROM”: 读取芯片内 EEPROM 内容。
- “Verify”: 程序区校验。如果芯片已加密, 将提示校验失败。
- “Erase”: 擦除目标芯片。执行此操作成功后, 目标芯片将成为空白芯片。
- “Blank Check”: 检验目标芯片是否为空白芯片。
- 上升沿这个按钮让复位脚电平由低变高, 使 ICD2 释放对目标板的控制, 这样目标板就能进入运行状态, 开始实际工作。
- 下降沿这个按钮让复位脚电平由高变低, 使目标板上芯片进入复位状态停止运行。

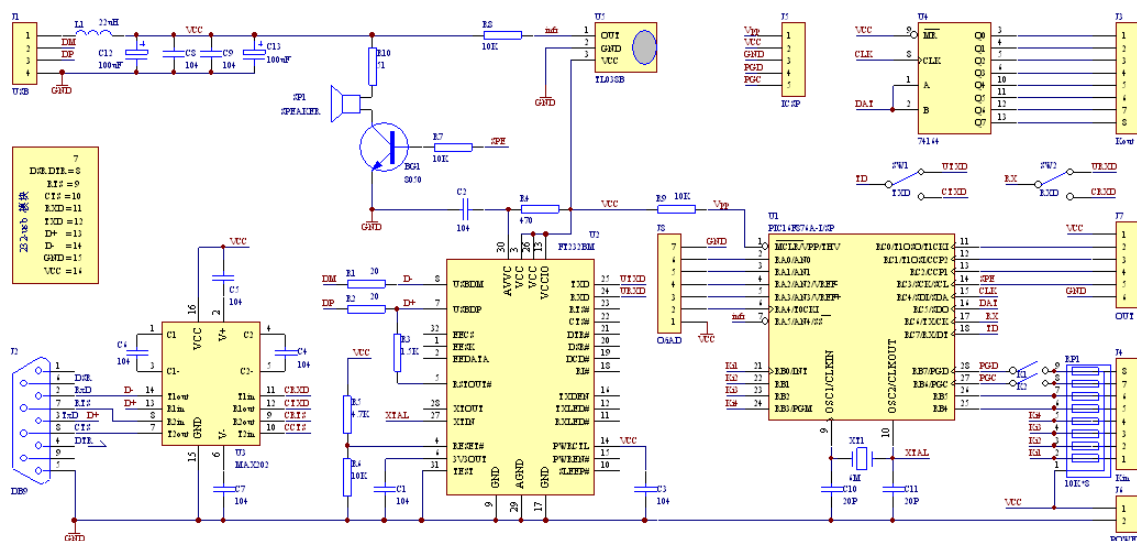
按钮 的功能是烧写程序后, **不用把 ICD2 与目标板断开就能运行**。在调试过程中若要验证程序的实际运行效果, 可先让目标芯片脱离 ICD2 调试模式, 再选择 ICD2 为编程模式, 并使用工具栏上的 快捷图标对目标芯片进行编程操作, 注意芯片的 CONFIG 字设置。烧写成功后按 快捷图标目标板就能进入运行状态, 开始实际工作。按 快捷图标使目标板上的芯片进入复位状态, 停止运行。此时可以结束编程模式再把 ICD2 设回调试模式, 重新进行调试。

- 如发生通信中断情况, 可用这个按钮使 ICD2 重新联接。

以上介绍的为入门应用, 详细的使用可参考 MPLAB IDE 使用说明书, [应用技术的提高](#) 可通过项目实践或参考其他应用技术资料。再通过从简单到复杂的多次反复应用和实践。才会得到较大的收获和积累。最终走向成功!



## ◆ 项目实例





## ◆ 常见问题解释

### 1、ICDWarn0020: Invalid target device id (expected=0x70, read=0x0)

出现这项提示时，是 ICD2 主机与目标芯片之间通信不正常，可检查：

- 目标板上的芯片是否正确，电源“Power”设置是否正确。
- 芯片的调试引脚线路连接是否错误，引脚上是否有电容或电感等元件。
- 如果读回的 ID 不是 00，但与期望的 ID 不符合，则为软件选择的芯片型号与实际的目标芯片型号不符合。
- 如果读回的 ID 为 00，则为 ICD2 主机与芯片之间通信有问题，检查六芯线与芯片的连接是否可靠。由于水晶头与卡座之间经常需要插拔比较容易松动而导致接触不良，大部分的 0020 警告都是由于这个原因产生的。
- 另外电源地线未连接好，包括目标板电源供电不足，也可导致此提示。常见为只连接了一组的电源地到六芯线，请把所有电源地均连接好，包括 AVDD，AVSS。

### 2、ICD0019: Communications: Failed to open port: (Windows::GetLastError() = 0x2, '系统找不到指定的文件。')

ICD0021: Unable to connect with MPLAB ICD 2

出现这二项提示时，是 PC 与 ICD2 之间通信不正常，可检查：

- ICD2 是否已连接到 PC，用 232 串口连接的，检查供电和“Power”设置是否正确。
- 驱动是否正确安装，或者由于 PC 的 USB 口供电不足导致驱动无法正确加载。
- 如果热插拔 ICD2 的 USB 线，很容易出现以上提示，关掉 MPLAB IDE，过一会再重新启动 IDE 软件并连接即可。

### 3、ICD0161: Verify failed (MemType = Program, Address = 0x40, Expected Val = 0x0, Val Read = 0x3FFF)

出现该提示时，是由于目标板芯片烧写不正确导致检验失败，可检查：

- 目标芯片的 MCLR/VPP 引脚上没有除 ICD2 接口外的其他电路，常见复位电路请采用跳线进行隔离。
- ICD2 与芯片的接口线路中有没有连接电感或电容特性的电路。
- 如果选择由 ICD2 向目标板供电，则有可能由于 ICD2 不能向目标板提供足够的工作电流，导致烧写不稳定出现烧写失败。此时应该采取目标板独立供电。并在软件上不要选择从 ICD2 向目标板供电。
- 注意：某些特殊的芯片（例如 PIC16F627A），在同时使用了内部振荡和内部复位的情况下，烧写成功后也将会提示 CONFIG 区校验失败的信息。这是正常的表现，由于同时使用了内部振荡和内部复位，一旦烧写成功，芯片将直接运行目标程序，此时由于无法进入烧写模式，校验必然是不正确的。

### 4、ICD0083: Debug: Unable to enter debug mode. Please double click this message for more information.

无法进入调试模式，出现该错误时，一般为振荡不正常。确认 MPLAB IDE 软件上设置的振荡类型与实际的目标板是否符合，或者目标板振荡是否起振。

- 4MHz 以上晶振推荐使用 HS 模式，如果是外部 32K 低频晶振，重新“connect”一下可执行调试指令。
- 内部或外部晶振使能了 PLL 功能而进不了 Debug 模式的，关掉 PLL 即可。
- 配置位设置不当引起的，查看 Configure→Configuration Bits，确保看门狗被禁止，代码保护、掉电检测等被关闭。
- 复位引脚和芯片的接口线路中有没有连接电感或电容特性的电路。

5、**ICDWarn0015:** Program memory has changed since last program operation? Continue with Debug operation?

调试时出现该提示，是程序修改之后没有再重新编译、烧写到芯片里。

6、... Failed Self Test. See ICD2 Settings (status tab) for details.

**自检不通过提示。**ICD2 自检是对 Target Vdd、Target Vpp、MCLR Gnd、MCLR Vdd、MCLR Vpp 五个管脚电压进行检测，任何的一项不正常都会通不过。出错具体情况可在 ICD2 的 status 栏进行查看，找出出错的项，再检查该项的相关硬件连接。若是 PIC18J、PIC24、dsPIC30、dsPIC33 系列的器件，是无法从 ICD2 供电给它们的，要求目标板外接电源，且要注意它们的电压范围。

可能有些芯片会提示 Vpp High 从而导致自检通不过，这是 ICD2 的固件中检测 Vpp 电压时所设的上限阈值偏小所致，通常可忽略，ICD2 还是能正常编程和调试的。

7、**ICDWarn0044:** Target has an invalid calibration memory value (Value). Continue?

用 ICD2 调试或烧写 PIC16F630/676.. 等带内部 RC 的器件时，这些器件的程序存储器最后一个单元上的内部 RC 校准值已被修改，可用 PICKIT 1 重新生成一个值，也可用 MPLAB IDE 的菜单命令 Programmer→settings，在打开的窗口里，选“calibration memory”页，在“Allow ICD2 to program calibration memory”前打勾，再填一个新的值，烧写后再次连接就没问题了。

8、**当在程序的 XX 位置设置一个断点时，为什么它停止在地址 XX+1 处，而不是地址 XX 处？**

ICD2 在断点后的指令上暂停。这意味着设置了断点地址 XX 处的指令会被执行，接着当它发现断点时，程序计数器会指向地址 XX+1。如果需要在某代码的 XX 指令处暂停，必须在地址 XX 处插入一条 NOP 指令。

9、**单步运行时，定时器为什么运行不正常？**

这是使用在线调试器的缺点之一。由于代码实际上是在调试执行程序中运行，在调试执行程序运行期间，即使用户的应用程序被中止，定时器也会继续运行。

10、**为什么 ICD2 在调试的时候中断不正常？**

这是因为 ICD2 在调试的时候占用了一些内部资源，导致不能调试芯片的定时器中断和 PIC18 系列器件的高优先级中断。

11、**在内存中看见 70h 为 RR，且不能手动修改，这是为什么？**

说明这个地址的寄存器已被 ICD2 占用，ICD2 在线调试的时候会占用芯片的部分资源，具体的可以参考下面文档：Microchip\MPLAB IDE\ICD2\hlpMPLABICD2.chm\Operation\MPLAB ICD2 Overview\Resources Used By MPLAB ICD2 帮助文件中查看，在内存中查看标为 RR 时不可用。

12、**通过 RETFIE 指令来使用高优先级中断时，为什么 W、STATUS 和 BSR 寄存器的值会改变？**

用于高优先级中断和 CALL FAST 的影子寄存器被 ICD2 使用了。这些是保留给 ICD2 操作的资源。如果断点设置在 CALL FAST 子程序内，或者在使用 RETURN FAST 或 RETFIE 指令时使用了影子寄存器的高优先级中断服务程序，将会出现问题。

13、**为什么有时软件上无法选择由 ICD2 向目标板供电？**

对于 PIC18FXXJXX、PIC24、dsPIC30F、dsPIC33F 系列的器件，MCHP 为了保护开发工具，禁止了该项操作，调试或烧写的时候目标器件需要独立供电。

14、**在用 PIC12F629/675/16F630/676/627A/628A/648A 仿真头时，需要注意哪些地方？**

必须使用内部振荡或外部有源振荡；MCLR 引脚必须为高电平。当 GP1/RA1 为高电平时将无法进行调试，不能在对 GPIO 操作的指令上设置断点。

15、**在调试 PIC12F629/675/PIC16F627A/628A/PIC16F630/676/PIC16F648A 时，为什么会有 Invalid target device id 警告？**

在使用 ICD2 仿真头调试这些器件的过程中，器件的 GP1/RA1 引脚不能被拉高，否则会有警告 Invalid target device id。参见 MPLAB ICD2 (DS51292) 文档，获得更多关于对这些器件

使用 ICD2 的信息。

**16、有些芯片使用时用到了内部晶振和内部复位，如 PIC16F630/690 等，用 ICD2 编程时一旦使能内部晶振和内部复位会有警告信息：**

ICDWarn0033: You have selected Internal MCLR and Internal Oscillator in your configuration settings. If your code makes use of port pins that correspond to Clock and Data pins in programming mode, you may not be able to reprogram your device. See on-line help for this warning for more information. (OK/Cancel)

如忽略该信息，初次 ICD2 可以成功编程该芯片。但如果芯片中已有配置字是内部晶振和内部复位的程序，且程序中有对 ICSPCLK 和 ICSPDAT 管脚的操作，再次编程就会有校验失败的错误提示。这是因为当使用内部晶振和内部复位时，ICD2 一旦和目标板相连，目标芯片即已工作，ICD2 对 ICSPCLK 和 ICSPDAT 的控制被目标芯片对这两脚的操作打乱，故而进不了编程模式。这时该考虑其他编程工具，或者在程序开头加 2ms 以上延时，保证芯片在真正执行程序前进入编程模式。

**17、连接时出现 ICD0152: Failed to read all of requested bytes (0x0 of 0x8)**

如果 ICD2 是串口连接的，需修改串口属性，具体步骤如下：

右击我的电脑→属性→硬件→设备管理器，在端口（COM）→COM1（如果用其他 COM 口，则选择相应的 COMx）→属性→端口设置，在这里要做两个设置：

1 流控制方式：选择为硬件

2 点“高级”按钮，在“使用 FIFO 缓冲区...”把前面的勾去掉。设置完后重启电脑。

**18、为什么需要 ICD 转接仿真头？**

对于 ICD2 支持的低引脚器件，如果在线调试引脚被保留，将不能有效地使用这些器件。试想一下，对于 8 个引脚的器件，6 个 I/O 引脚失去其中 3 个的情形。为此，生产了特殊的转接头来仿真这些低引脚器件，转接仿真头是单片机加上与 ICD2 的在线通讯接口引脚，这使得能在目标应用中使用所有的引脚。

这样做的优点是可以用 ICD2 开发低引脚器件。缺点是对于这些器件，在进行在线调试时，不能在目标应用中采用 ICD2 的 ICSP 连接口，而要使用仿真头来开发使用。

**19、dsPIC 使用 ICD2 在线调试的接口选择。**

在某些情况下，dsPIC 默认的编程和调试引脚功能 PGC/EMUC 和 PGD/EMUD，可能与其他外设（如 IIC、SPI 或 UART 模块等）引脚复用。在这种情况下，应用中可将这些引脚用于编程，但不能用于在线调试。此时在线调试应该使用备用调试通道，EMUC/EMUD1、EMUC2/EMUD2 或 EMUC3/EMUD3，具体的电路设计请参考 152816.pdf 文档。

**20、为什么在配置锁相环（PLL）振荡器时 ICD2 被挂起？**

这是 PICmicro 单片机所要求的。在对 PLL 振荡器的配置位编程后，电源需断开后再加到目标板上。如果不这样做，目标 PICmicro 单片机将没有时钟，调试模式将不能工作。同样，如果在切换到 PLL 模式时，电源没有被断开再连接，器件可能会运行，但没有使用 PLL。

**21、ICD2 调试时 ERROR 灯突然点亮，但调试其它 MCU 又正常。**

这可能是 ICD2 占用了芯片内部资源，程序到一定规模如果发生 RAM 或 ROM 资源冲突，就会引起错误，ERROR 灯就亮。

**22、在 PICMATE2004/2002 下编译通过的程序，通过 MPLAB IDE 编译失败。**

这种情况大多是由于符号的大小写敏感选项未正确设置造成的。按如下步骤进行设置：建立一个项目，添加源文件，到 Project→Build Option→Project 打开窗口，在 MPASM Assembler 页的 Disable case sensitivity 选项，默认为不打勾，区分大小写，打勾则不区分大小写。

**23、在第三方设计的仿真工具上调试仿真正常的代码文件，写入芯片后不能运行。**

这种情况是第三方工具软件编译保存的格式不同，或是不包含 CONFIG 配置字信息所致。



使用 File→Import 导入第三方工具软件编译的 hex 文件, 检查芯片设置和 CONFIG 配置字正确后, 然后再烧写。

或使用 MPLAB IDE 编译器, 对在第三方仿真器下调试通过的源代码再次编译。这时候需要建立必要的项目, 而且必须对芯片类型, CONFIG 配置字进行相应的设置。此时产生的 hex 文件是包含配置字信息的 hex 文件。可用 File→Export 命令导出保存, 用于对芯片的烧写。

**24、安装 USB 接口时, 选择的是从指定位置安装, 第一次提供的是 ICD2W2K1. INF, 安装完后又重复安装, 提供的是 ICD2W2KL. INF, 这是怎么回事?**

这是正常的, 它需要安装两次。

**25、ICD2 的 USB 驱动程序在哪里?**

在 MPLAB IDE 的安装目录下: Program Files\Microchip\MPLAB IDE\ICD2\Drivers\icdw2k.inf(icdw2kl.inf)。

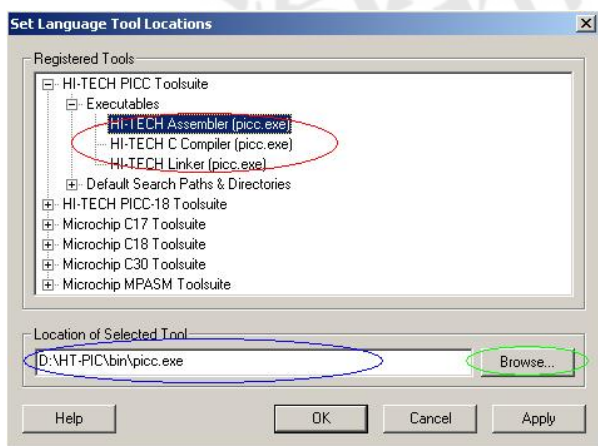
**26、当安装时, 为什么找不到驱动, 即使能在驱动文件夹中看到它们, 并且在驱动向导中指向了正确的文件夹也不行?**

这个问题可以这样解决: 退出驱动安装向导, 再到控制面板选择“Add New Hardware (添加新硬件)”。在系统搜索新硬件后, 选择“No, the device isn't in the list.”, 然后选择“No, I want to select the hardware from a list.”, 再选择“Universal Serial Bus Controller.”。当出现“Have Disk...”按钮时, 进入驱动文件夹, 并选择正确的驱动。

**27、网上有人说无论是自搭电路或是使用适配器, 只要接入相应的晶体即可同时构成烧写和仿真双功能, Vpp 接脚无须任何处理, 会自动在二者间切换(此功能世上独此一家), 不像其他的 ICD2 需要更改 Vpp 电路(通常是在仿真时将芯片的 MCLR 端和 ICD 的 Vpp 连接断开, 再通过一个 4.7k 电阻接到 VDD)。这是怎么回事?**

误导初学者。用户通过 MPLAB IDE 的下拉菜单, 均可方便地让 ICD2 在烧写和仿真功能之间任意切换, 根本无需更改 Vpp 电路!!!

## 设置语言工具



如果需要使用 C 语言来编程, 那么还需要设置语言工具的路径。

点击主菜单的 Project→Set Language Tool Locations 来设置语言工具路径。在系统弹出的“Set Language Tool Locations”(设置语言工具位置)对话框里选择相应的语言工具, 本例中选择的是 HI-TECH PICC 编译器。在相应的语言选项的子树“Executables”(可执行文件)树下点击相应的选项, 使用“Browse”(浏览)来设置对应的可执行文件。如图所示。

在本例中, 将“HI-TECH Assembler”、“HI-TECH Compiler”和“HI-TECH Linker”都设置指向为 D:\HT-PIC\bin\picc.exe, 这是 PC 上的 HI-TECH PICC 编译器路径, 应该根据实际安装情况来设置。

如果只使用汇编语言进行编程, 那么他们可以不设置这项, 直接使用默认设置。