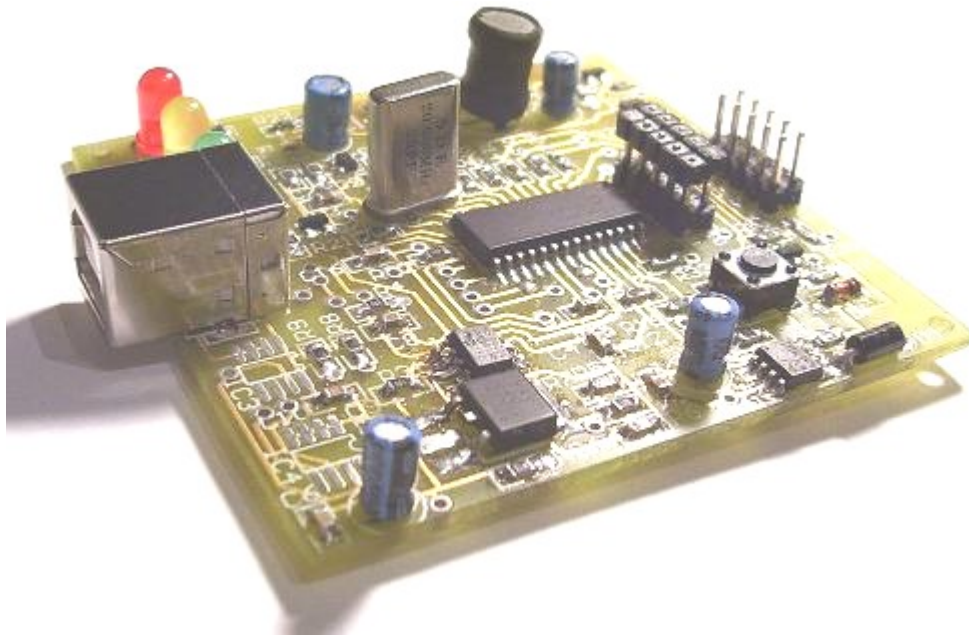


“PICkit2” Manual del Usuario – Versión 1.0
Diseñado por: Ing. Roberto Simone

www.edudevices.com.ar
e-mail: soporte@edudevices.com.ar

“PICkit2”



Manual del Usuario

Versión 1.0

***Herramienta de programación y
Debug para dispositivos Microchip***



www.edudevices.com.ar

“PICkit2” Manual del Usuario – Versión 1.0
Diseñado por: Ing. Roberto Simone

www.edudevices.com.ar
e-mail: soporte@edudevices.com.ar

Control de versiones:

1.0 Edición ORIGINAL OCTUBRE del 2011.

- Contenido

- 1.0 Introducción.**
- 2.0 Características generales.**
- 3.0 Contenido del kit.**
- 4.0 Instalación del software.**
- 5.0 Instalación del Hardware del Sistema.**
- 6.0 Conector ICSP para la conexión a la plaqueta de aplicación.**
- 7.0 Poniendo en marcha el sistema.**
- 8.0 El modo Debug.**
- 9.0 Familias de MCU y el “PICkit 2”**
- 10.0 Preguntas mas frecuentes.**
- 11.0 Layout.**

Diseñado por:

Ing. Roberto Simone

Ingeniero de aplicaciones Microchip.

EduDevices.com.ar

E-Mail Privado: robertosimone@arnet.com.ar

Comentarios & sugerencias:

EduDevices.com.ar

República Argentina

Tel: (5411) 4584-3142

info@edudevices.com.ar

www.edudevices.com.ar

1.0 – Introducción:

El propósito de este manual es describir el funcionamiento de la herramienta *PICkit 2* y dar instrucciones para su uso en las distintas familias de microcontroladores Microchip. La herramienta *PICkit 2* es una herramienta abierta de bajo costo, de programación y debugging en tiempo real para una amplia variedad de microcontroladores de Microchip. Entre ellas podemos citar a las familias **PIC10F / PIC12F / PIC16F / PIC18F / PIC24F / PIC24H / dsPIC30F / dsPIC33F**.

El kit consiste de una **Placa Principal** que se conecta a una computadora por USB. Junto con ella, una **Placa de Zócalos** se conecta a la principal, permitiendo la programación de dispositivos con encapsulado DIP. Por último una **Placa Demo**, permite usarla como plaqueta de aplicación de prueba.

Las funciones de debugging están disponibles si se establece un vínculo, mediante el cable correspondiente, entre el *PICkit 2* y la aplicación a depurar. A lo largo del manual se explican los requisitos para lograr que un **microcontrolador admita el funcionamiento en modo debug**.

De ningún modo el *PICkit 2* pretende reemplazar en funcionalidad a herramientas de mayor prestación como **MPLAB ICD2 / MPLAB ICD3 / MPLAB ICE**. Se trata de una alternativa que ofrece alguna de las funcionalidades de las herramientas citadas pero a un costo mas accesible.

A título informativo se brinda una lista parcial de MCU soportados por el *PICkit 2*:

- PIC12F629 (solo programación)
- PIC12F675 (solo programación)
- PIC16F627A (solo programación)
- PIC16F883
- PIC16F884
- PIC18F252
- PIC18F452
- dsPIC33FJ128GP802
- Ver otros dispositivos en foros Microchip.

2.0 Características generales

El *PICkit 2* puede tanto **Programar y Emular en Tiempo Real** distintos MCU de **Microchip** de las familias **PIC12F / PIC16F / PIC18F / dsPIC30F / dsPIC33F**. Para una y otra tarea emplea una interface de 5 hilos, incluyendo GND. Esta interface de **5 hilos** es útil para todos los dispositivos que están al alcance del *PICkit 2* sin mayor hardware externo ni alimentación externa.

En cuanto al funcionamiento de un MCU en **modo Debug** todo depende del dispositivo empleado. Por lo general **los dispositivos de pocos pines** poseen un hardware limitado y por ello para **emular en tiempo real se requiere de una placa adicional llamada Header Board**, comercializada por Microchip. Los dispositivos de 28 ó mas pines pueden ser puesto en modo Debug simplemente con la interface de 5 hilos.

El modo Debug permite intercalar un **Breakpoint** en algún punto del programa y correr en tiempo real, tal como lo haría normalmente, hasta alcanzar el breakpoint donde la ejecución se detiene, y el *PICkit 2* toma el control del MCU. A partir de allí nos es posible volcar el contenido de la RAM del MCU en la computadora para evaluar el comportamiento del programa que se está ejecutando.

2.1 Diferencia entre un debugger y un ICE

Un **debugger** es una alternativa menos costosa respecto a un **ICE (In-Circuit Emulator)**. Existen pros y contras a la hora de utilizar una herramienta de bajo costo como el *PICkit 2*. Citamos algunas de sus características:

- Ciertos pines del MCU deben dedicarse a la comunicación con el *PICkit 2* que de otro modo estarían para su uso en una aplicación.
- El MCU a debuggear (depurar) debe tener una **fuentes de reloj** en funcionamiento.
- El *PICkit 2* permite trabajar con el dispositivo (MCU) que usará realmente la aplicación.
- **No hay que remover el MCU** de la aplicación, tanto para efectuar la Programación de la memoria Flash como para la Emulación en Tiempo Real del programa del usuario.

Características del PICKit2:

- Programador integrado al **MPLAB**.
- Alimentación desde el puerto USB.
- Puede proveer alimentación a la placa bajo depuración.
- Reemplazo económico del **MPLAB ICD3** o del **PICKit 3**.
- Comunicación a 2 hilos.
- **VDD** de los MCU **desde +1,8V a +5V**.
- Detección de cortocircuitos en VDD y VPP del circuito bajo ensayo.
- Contempla las familias de microcontroladores **PIC10, PIC12, PIC16, PIC18, PIC24, dsPIC30** y **dsPIC33**.
- Puede funcionar como programador **STAND ALONE (sin PC)**.
- Soporta **2 breakpoints** durante las sesiones de Debug.
- Emulación para frecuencias del MCU desde **32KHz**.
- Es un dispositivo **USB HID**. Los drivers se cargan automáticamente.

Características del kit (placas):

- Placa principal **100% compatible** con **PICKit 2** original.
- **Placa de Zócalos** para la programación de dispositivos **DIP de 8 / 14 / 18 / 28 / 40 pines**.
- Placa **“Demo 16F”**. Posee un **PIC16F883** que permite usar aplicaciones de ejemplo incluídas en el CD de cortesía.

3.0 Contenido del kit

El kit *PICKit 2* viene provisto de los siguientes elementos:

- **Placa principal *PICKit 2*.**
- **Placa de Zócalos** para programación de dispositivos de encapsulado DIP.
- **Placa de evaluación “*Demo 16F*”** con un microcontrolador **PIC16F883**.
- Arnés de cable plano de **10 hilos IDC-10 Hembra / Hembra** para la conexión **ICSP**.
- Cable serie Tipo A – B USB 2.0.
- CD ROM de instalación (Manual del usuario, Bibliografía, etc.)

4.0 Instalación del software

4.1 Instalación del MPLAB IDE

Al momento de la publicación de este manual existen dos tipos de entornos de desarrollo **MPLAB IDE de Microchip**.

- **MPLAB IDE tradicional versión 8.76.**
- **MPLAB X versión de evaluación (Beta).**

El **MPLAB X** es una versión renovada del **MPLAB IDE** tradicional pero actualmente se encuentra en versión de prueba, por lo que algunas de sus posibilidades se hallan limitadas. La **versión Beta** del **MPLAB X** no permite el empleo del *PICkit 2*. Con la futura aparición de la versión definitiva del **MPLAB X** se podrá emplear el *PICkit 2* con este entorno. Por el momento recomendamos el **MPLAB IDE** tradicional.

En el CD de cortesía se encuentra el archivo de instalación del **MPLAB IDE** tradicional. Este entorno permite:

- Escribir y editar programas escritos en **ensamblador** para todas las familias de microcontroladores de Microchip.
- **Simulación por software** de los programas compilados.
- Programación de todos los microcontroladores de Microchip con las herramientas oficiales de acuerdo a las limitaciones y características de cada familia de procesadores, y de acuerdo a cada dispositivo en particular.

Tengase en cuenta que el **MPLAB IDE no soporta** en forma nativa la compilación de archivos fuentes escritos en **lenguaje C**. Para hacerlo se necesita **instalar compiladores especiales**. Ellos **no son gratuitos** pero existen **versiones de evaluación** que si lo son, y pueden **descargarse de la página Web de Microchip**. Las versiones de evaluación tienen un tiempo de uso limitado y la compilación no optimiza los recursos de memoria del código obtenido.

Requerimientos mínimos del sistema

Hardware:

- Procesador → Pentium o equivalente.
- Memoria → 128MB (se recomienda 512 MB o superior).
- Puerto USB 2.0 (@ 500mA).
- Lectora de CD ROM.

Sistema Operativo:

MPLAB IDE ha sido probado en los siguientes sistemas operativos:

- Windows® 2000 SP4.
- **Windows XP SP2.**
- Windows Vista™ .
- **Windows 7 OS.**
- Windows XP 64.
- Windows Vista 64.
- Windows 7 64 OS.

(Sistemas Operativos Recomendados)

Espacio en Disco:

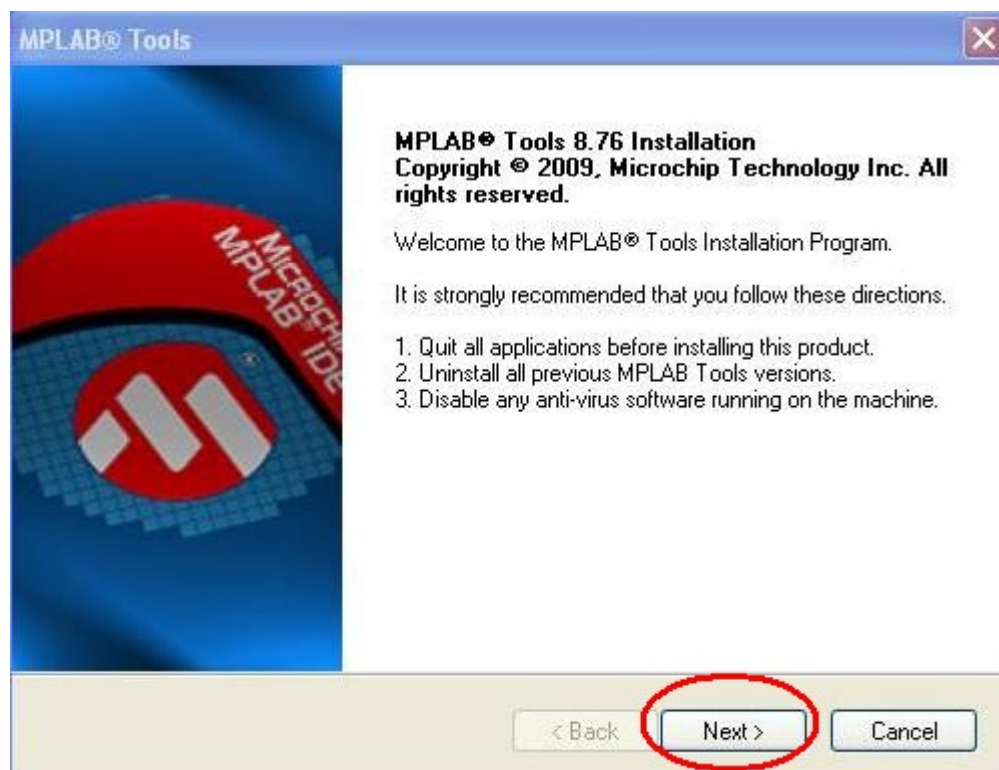
400MB de espacio libre en disco.

Pasos a seguir:

1) Introduzca el CD ROM con la etiqueta electrónica **“PICkit 2”** en el lector de la PC o Notebook a utilizar y seleccione la siguiente ruta:

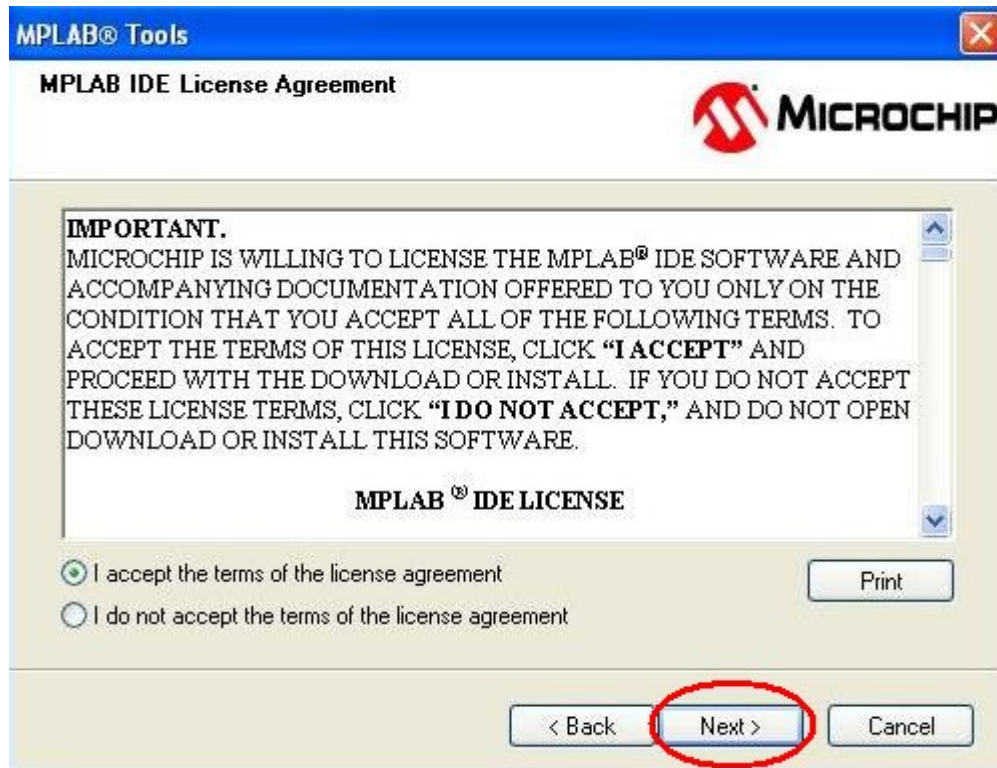
D:\PICkit_2\MPLAB IDE

2) Ejecutar el archivo “setup” de dicha carpeta y nos aparecerá la siguiente ventana:



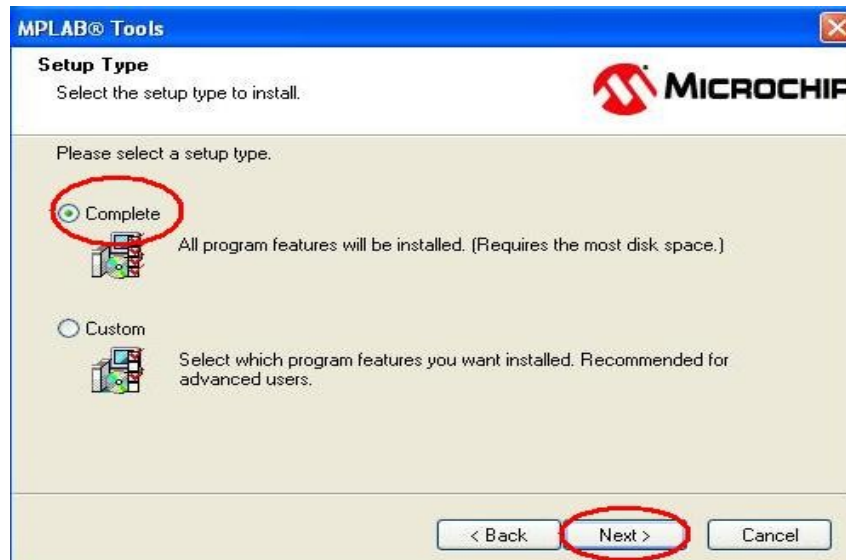
3) Hacemos cliq en el botón “Next” para comenzar la instalación.

4) Aparecerá una pantalla de Bienvenida y se deberá seguir atentamente las indicaciones de las sucesivas pantallas a lo largo de la instalación que nos irán guiando paso a paso en la misma.



Al hacer cliq en el botón Next damos comienzo a la instalación.

6) Sugerimos elegir la instalación completa:

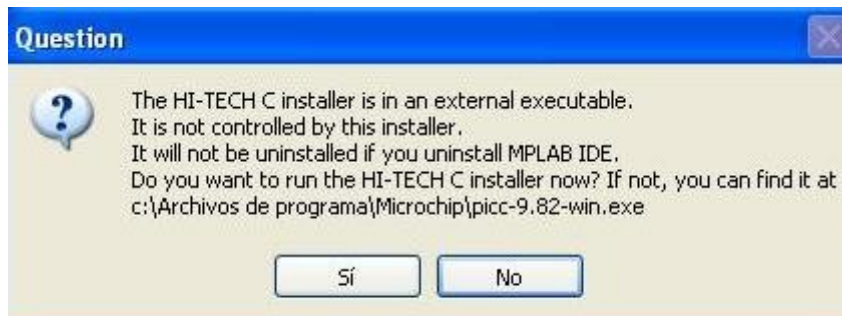


y no cambiar la carpeta de instalación:



En las próximas ventanas **debemos aceptar los términos de la licencia** del software si queremos que se instale el MPLAB IDE.

7) En determinado momento, el instalador nos ofrecerá la posibilidad de copiar un compilador de lenguaje C, llamado **“HI-TECH C”**. Se trata de un compilador cuya instalación es independiente del **MPLAB IDE**. El usuario puede decidir libremente si desea instalar en este punto a este compilador.



Como último paso el instalador nos solicita que reiniciemos la computadora. **Permitimos que lo haga** y cuando haya finalizado ya tendremos instalado el **MPLAB IDE**.

4.1 – Instalación del software PICkit 2 Programmer

Una aplicación muy útil provista junto con el **MPLAB IDE** es el **PICki2 Programmer**. Se trata de un software para programar las memorias FLASH, EEPROM y los fusibles de configuración de los microcontroladores Microchip, mediante el **PICkit 2**, por supuesto.

Este software es particularmente útil en la etapa de producción ya que el único archivo que nos permite abrir es el **código objeto (.hex)**. A diferencia del **MPLAB IDE**, en el cual debemos abrir el archivo fuente, o incluso el proyecto, compilar, y recién entonces tendremos la chance de poder programar las memorias del microcontrolador, con este software abrimos directamente el código previamente compilado y ya podemos programar a los dispositivos.

Pasos a seguir:

1) Introduzca el CD ROM con la etiqueta electrónica **“PICkit 2”** en el lector de la PC o Notebook a utilizar y seleccione la siguiente ruta:

D:\PICkit_2\Software PICkit 2

2) Ejecutar el archivo “setup” de dicha carpeta y nos aparecerá la siguiente ventana:



El resto de la instalación es muy sencilla, el asistente nos irá guiando para tener instalado el software complementario al MPLAB IDE.

4.2 – Instalación de los drivers

El *PICkit 2* es un **dispositivo de interfaz humana o HID (Human Interface Device)** por lo cual Windows provee automáticamente los drivers necesario. En consecuencia **no es necesaria la instalación de ningún driver desde el CD ROM.**

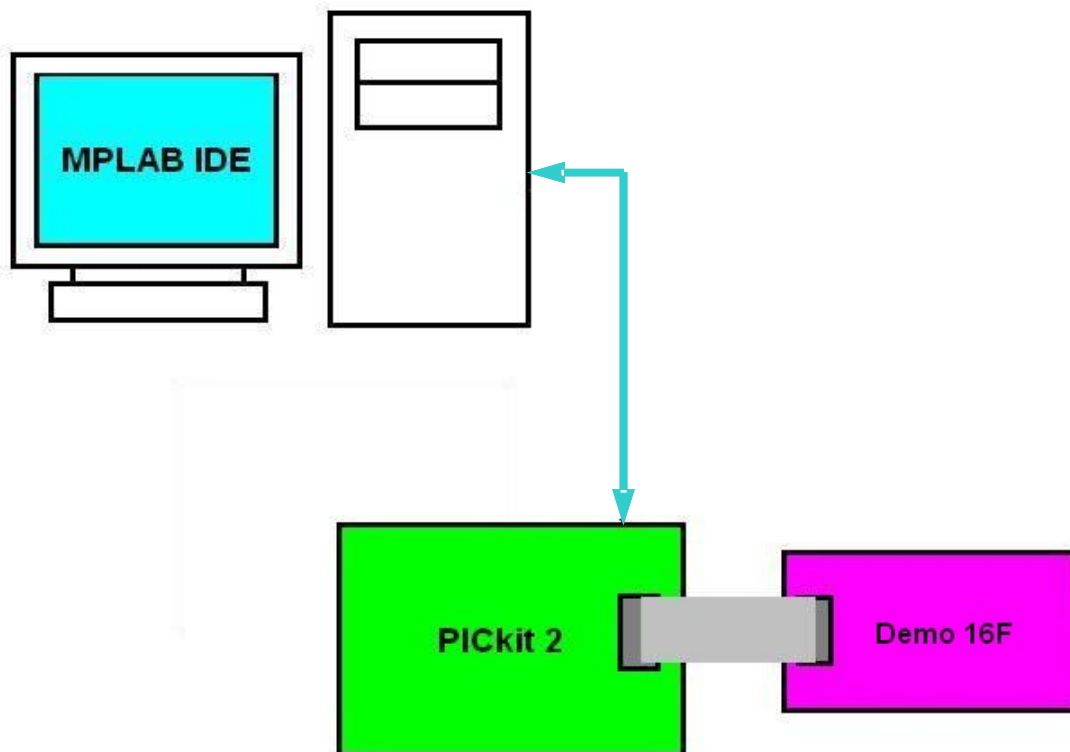
5.0 - Instalación del hardware PICkit 2.

Tras haber instalado el **MPLAB IDE** y el **PICkit 2 Programmer** es momento de presentar los 3 esquemas de trabajo brindados por esta herramienta. Para ello recordemos que el kit contiene 3 placas:

- **Placa principal “PICkit 2”**-- Es la herramienta propiamente dicha.
- **Placa “Demo 16F”** -- Placa de demostración y práctica con el MCU **PIC16F883**.
- **Placa “Zocalos PICkit 2”** -- Placa con zócalos para la programación de dispositivos con encapsulados DIP.

5.1 – Esquema 1. PICkit 2 + Demo16F. Alimentación provista por el PICkit 2

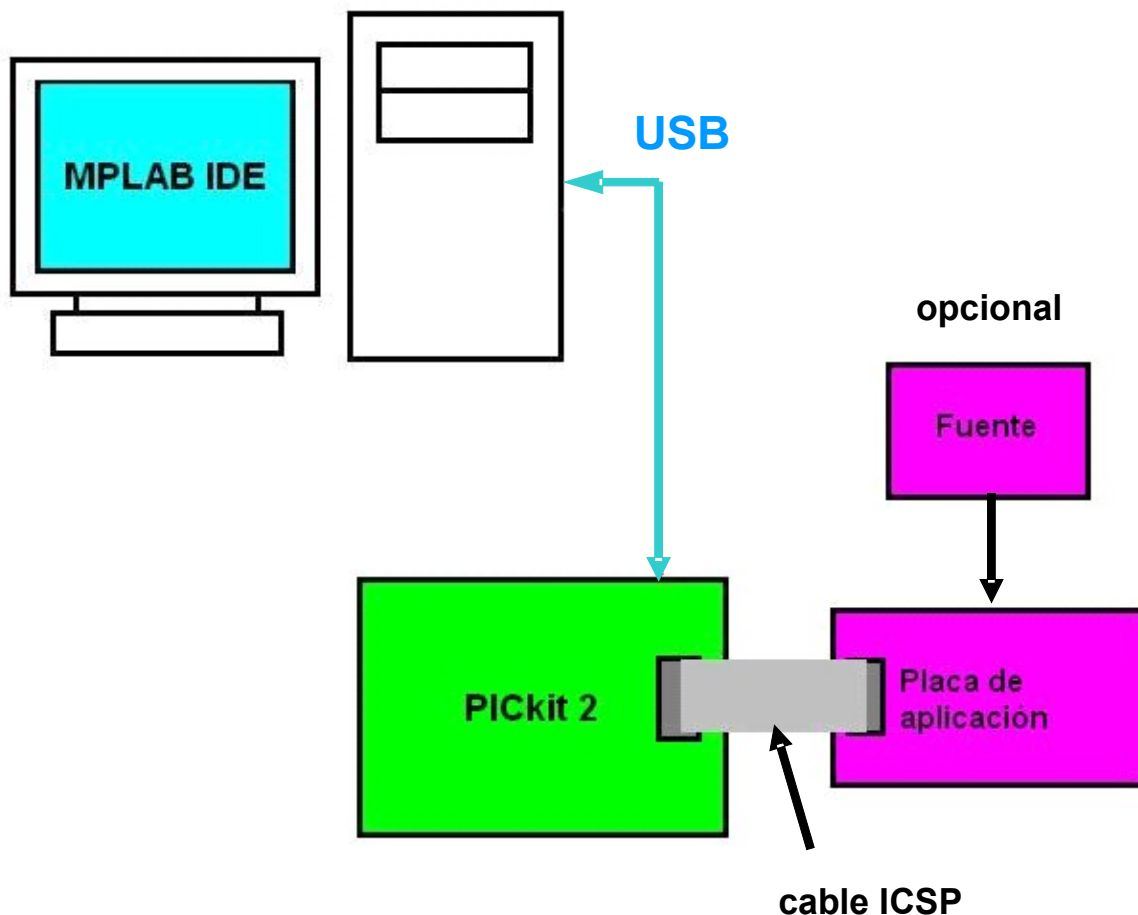
El primer esquema nos muestra como conectar el **PICkit 2** a la plaqueta **“Demo16F”**:



Este es el esquema mas conveniente cuando abrimos por primera vez el kit, pues nos permite comprobar su correcto funcionamiento utilizando 2 plaquetas que ya han sido probadas.

5.2 – Esquema 2. PICkit 2 + placa de aplicación.

Durante el desarrollo y depuración de aplicaciones diseñadas por el usuario el esquema de uso del “PICkit 2” será el siguiente:

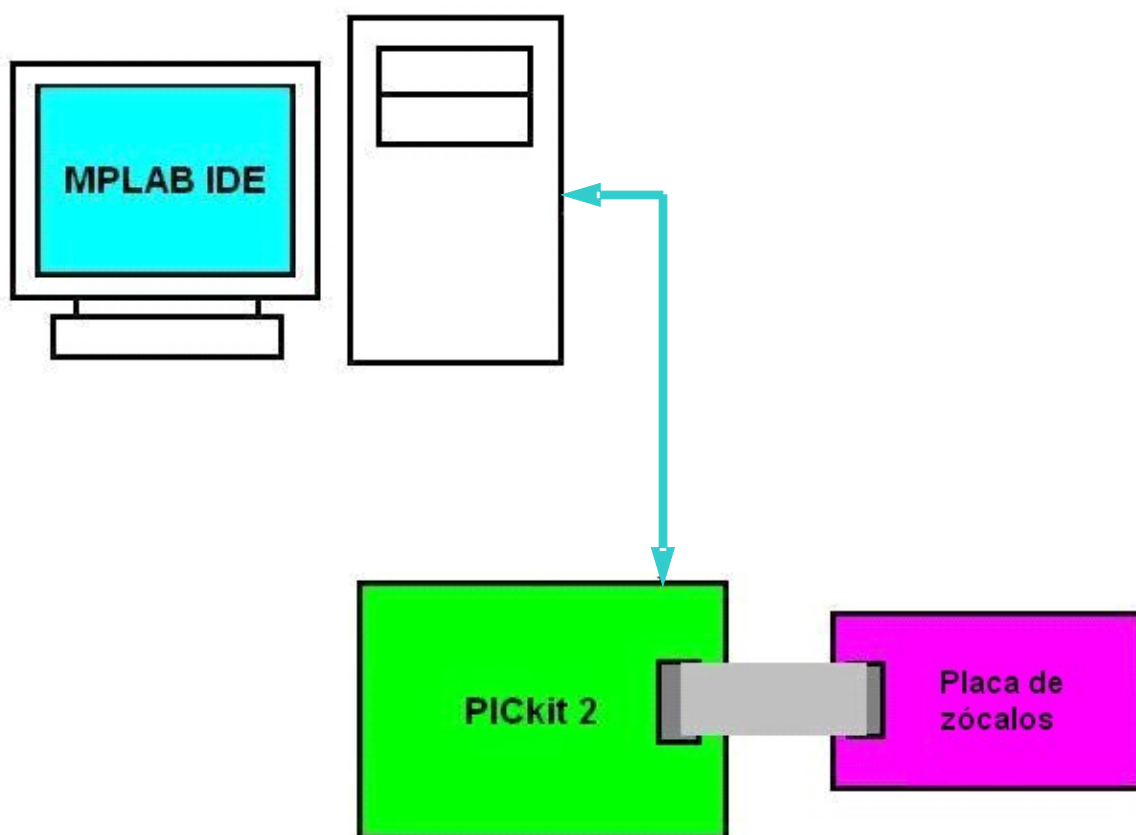


Es muy similar al primero, pero la placa “*Demo 16F*” fue **reemplazada por la placa de aplicación del usuario**, la que eventualmente puede tener su propia fuente de alimentación. La otra posibilidad es que el propio *PICkit 2* suministre energía a la aplicación.

5.3 – Esquema 3. PICkit 2 + placa de zócalos. Alimentación provista por PICkit 2.

En los 2 esquemas anteriores podemos programar los microcontroladores de la placa de aplicación. Y también ponerlos en modo **Debug** para agilizar el proceso de diseño. Sin embargo, en algunos casos, como en la producción masiva, no es de interés colocar en modo **Debug** a los MCU, sino simplemente cargar el firmware en la memoria de programa.

Cuando empleamos MCU con encapsulado DIP el tercer esquema de conexión de responde mejor a las necesidades de producción:



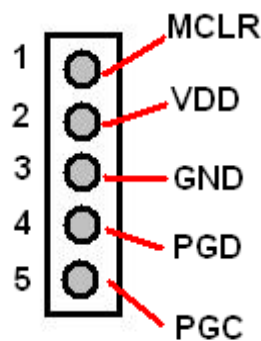
Los zócalos permiten **programar MCU DIP de 8 / 14 / 18 / 28 / 40 pines**. En particular, los **dispositivos de 18 / 28 / 40 pines requieren la conexión a 0V del pin PGM del MCU** para una correcta detección y programación del dispositivo. **Una resistencia de pull-down ha sido incluida en la placa de zócalos.**

Los MCU de montaje superficial se ajustan obligatoriamente al esquema 2.

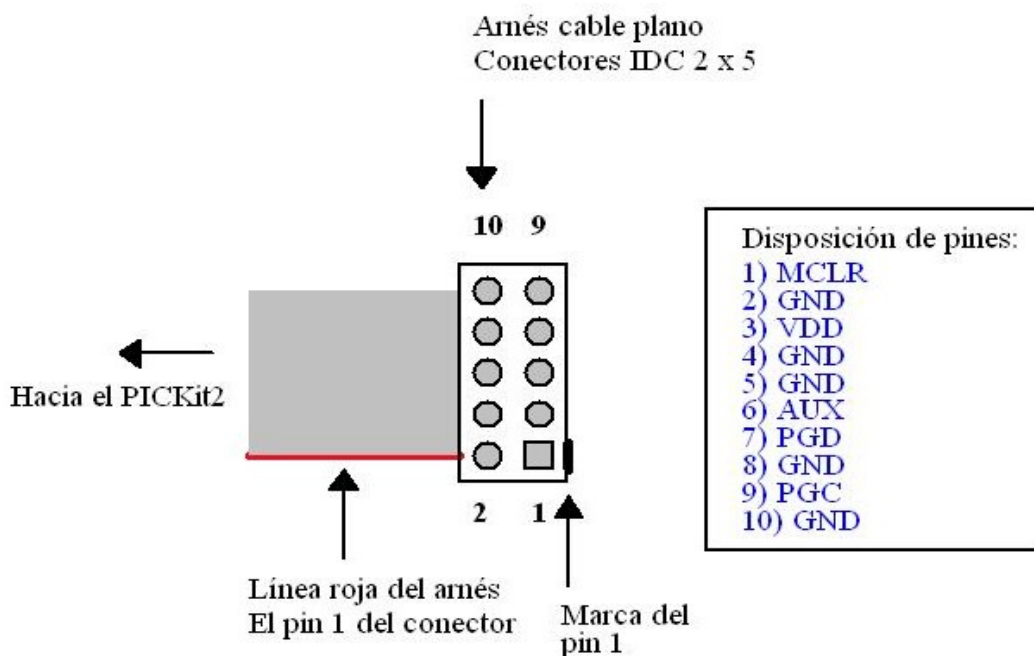
6.0 – Conector ICSP para la conexión con la placa de aplicación

El método de comunicación utilizado por el *PICkit 2*, y otras herramientas, para programar los distintos dispositivos Microchip, es una comunicación sincrónica a 2 hilos, conocida por el acrónimo **ICSP (In Circuit Serial Programming)**.

La distribución de las señales en el conector de programación de Microchip es la siguiente:



Esta disposición de pines es respetada en el conector **ICSP** del *PICkit 2* de **Edudevices**, solo que **agrega una segunda hilera de pines, conectados mayormente a GND**, con la intención de permitir mayor longitudes del cable plano del arnés de programación.



Arnés + conector ICSP visto desde arriba

6.1 – Descripción de pines.

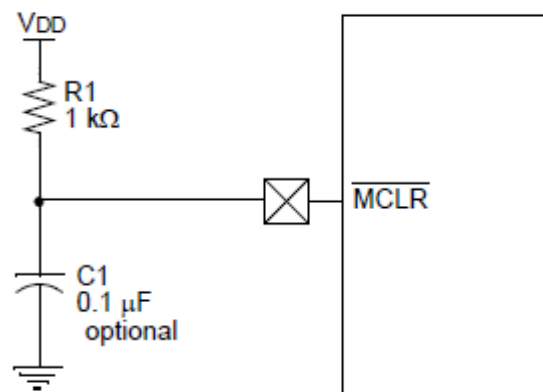
MCLR “Master Clear/VPP”

(pin 1 conector Microchip – pin 1 conector EDUDEVICES)

Cuando un PIC funciona en **modo Debug o durante la programación**, el **“PICkit 2”** provee una tensión de **+13,8V** a este pin para **colocarlo en modalidad de programación o de Debug**. Esta tensión no es la tensión elevada aplicada a la FLASH del MCU para borrarla. Dicha tensión es generada internamente por una bomba de carga, que es habilitada por estos +13,8V. Por este motivo el **pin MCLR consume muy poca corriente**.

Durante el funcionamiento normal del MCU, algunos dispositivos permiten emplear **este pin como puerto de entrada**. Una tercer función que puede cumplir este pin es la de **RESET externo**.

Es muy recomendable respetar el circuito de RESET externo sugerido por Microchip:

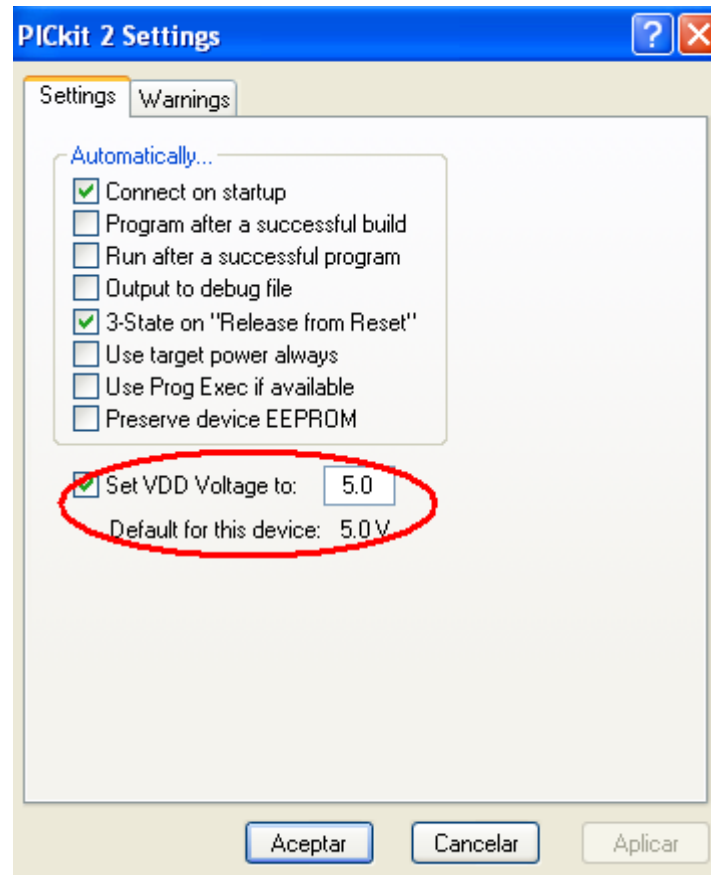


En algunos dispositivos, como el **PIC16F877**, se sugiere incorporar una resistencia adicional al circuito. **Siempre debe consultarse la hoja de datos de cada MCU en particular** para mayores precisiones.

VDD “Alimentación VDD del MCU de la aplicación”

(Pin 2 conector Microchip – pin 3 conector EDUDEVICES)

Si la alimentación del MCU es provista por el “PICKit 2”, debe configurarse el MPLAB IDE a tal efecto:



El nivel de tensión elegido dependerá tanto del dispositivo elegido como de la placa de aplicación con la que se está trabajando. Si la placa de aplicación consume una corriente excesiva o directamente existe un cortocircuito en la alimentación, el “PICKit 2” lo detectará e indicará el error en pantalla y mediante un LED.

GND “Masa”

(Pin 3 conector Microchip – pin 5 conector EDUDEVICES)

PGD “Pin de datos de la comunicación sincrónica ICSP/Debug”

(Pin 4 conector Microchip – pin 7 conector EDUDEVICES)

Ya hemos mencionado que la comunicación con los MCU se realiza por un par sincrónico. Esta señal es la línea de datos de esa comunicación.

PGC “Pin de reloj de la comunicación sincrónica ICSP/Debug”

(Pin 5 conector Microchip – pin 9 conector EDUDEVICES)

Ya hemos mencionado que la comunicación con los MCU se realiza por un par sincrónico. Esta señal es la línea de datos de esa comunicación.

6 – 2 Precauciones en la conexión ICSP

Cuando el **“PICKit 2”** es conectado a una placa de aplicación se deben tomar ciertas precauciones para evitar el daño de la herramienta o que impidan que la herramienta comunique con el MCU.

Primero citamos aquellas de carácter destructivo:

- **No inyectar al “PICKit 2” tensiones mayores a la VDD del MCU en los pines MCLR y VDD.**
- **Evitar cortocircuitos o consumos excesivos en los pines MCLR y VDD.**
Si bien el **“PICKit 2”** es capaz de detectar estas situaciones y desconectar estas señales de la placa de aplicación, poner a la herramienta a salvo de estas situaciones la protege de daños.

Además de estas situaciones que pueden dañar la herramienta, existe otras, **que de no evitarlas impedirán la comunicación** entre el **“PICKit 2”** y el MCU. Mencionamos alguna de ellas:

- **No deben colocarse resistencias de pull-up en los pines PGC y PGD.** Forman divisores de tensión con los pull-down del **“PICKit 2”**, disminuyendo el nivel lógico que recibe el MCU.
- **No deben colocarse capacitores en los pines PGC y PGD. La comunicación sincrónica con el MCU es de alta velocidad.** Los capacitores no hacen sino impedir rápidas transiciones en estas señales.
- **No deben colocarse capacitores en la entrada MCLR.** Es suficiente con una resistencia de pull-up a VDD.

No deben intercalarse diodos en las líneas de PGC y PGD. Impiden la comunicación bidireccional entre el **“PICKit 2”** y el MCU.

7.0 – Poniendo en marcha el sistema

Para aprender a usar el **“PICkit 2”** nada mejor que “poner mano” en una aplicación de ejemplo. Para ello el kit cuenta con la plaqueta **“Demo 16F”** que dispone de un microcontrolador **PIC16F883** de 28 pines en encapsulado SMD (SOIC). Esta plaqueta de aplicación tiene:

- 8 LEDs de montaje superficial.
- 1 tact switch.
- 1 preset (apto para prácticas que utilicen el conversor A/D).

La mayoría de los pines de la placa **“Demo 16F”** dispone de pads donde pueden soldarse terminales o cable para conectarla a otro hardware. La plaqueta puede alimentarse desde el propio **“PICkit 2”**, pero si el usuario lo desea puede alimentarla externamente, situación propia para su uso independiente del **“PICkit 2”**.

8.0 El modo Debug

El **modo Debug** es aquel que nos permite **Emular en Tiempo Real** en los MCU Microchip. Por Debug se entiende que la ejecución de un firmware cargado por un MCU es controlada por la herramienta **“PICkit 2”**. Ella nos da la posibilidad de insertar un **breakpoint (punto de parada)**. **Si damos a la herramienta la orden de ejecutar un programa, este correrá a la misma velocidad y con las mismas características con que lo haría el programa cuando no se encuentra en modo Debug (modo Release).**

Al alcanzarse la línea de programa en la que se insertó el breakpoint, el control del MCU lo toma el **“PICkit 2”**, y por lo tanto la PC. A partir de allí podemos volcar el contenido de la RAM del MCU en la computadora y evaluar el desempeño del programa.

Para entrar en **modo Debug** se deben cumplir 2 pasos:

- **Cargar el código dentro del MCU.**
- **Usar el debugger para colocar breakpoints y correr el programa.**

La secuencia de operaciones que deben realizarse desde el **MPLAB IDE** para satisfacer estos 2 pasos es explicada en la sección 7, **“Poniendo en marcha el sistema”**.

8.1 Requisitos para poner un MCU en modo Debug.

Poner con éxito a un MCU en **modo Debug** requiere que se cumplan con ciertas premisas:

- Se debe tener conectado un **“PICkit 2”** a una PC.
- El **“PICkit 2”** debe proveer alimentación a la aplicación o bien esta debe tener su propia alimentación.
- El **“PICkit 2”** debe estar conectado al MCU con la **conexión de 5 hilos** explicada en la sección 6. La **VDD** y **VSS** del MCU **deben conectarse a VDD y GND** del conector ICSP **aunque el “PICkit 2” no provea alimentación.**
- El MCU de la aplicación debe estar **energizado y tener un oscilador funcionando.** Si el MCU no está corriendo el **“PICkit 2”** no puede debuggear.
- El MCU debe tener configurado los **fusibles de configuración** correctamente:
 - **No tiene que estar habilitado el Watchdog timer.**
 - **No debe protegerse la memoria de programa** (No CODE PROTECT ENABLED – debe estar en **DISABLED**).
 - **No debe protegerse la lectura de tablas** (No TABLE READ PROTECTED – debe estar en **NOT PROTECTED**).

8.2 Detalles del modo Debug

Cuando en el **MPLAB IDE** establecemos un **breakpoint**, la dirección del breakpoint es guardada en unos **registros internos de debug especiales** dentro del MCU. Existen comandos enviados por las líneas PGD y PGC, desde el **MPLAB IDE**, para comunicarse con estos registros.

Al dar la orden de correr el programa (**Run**) desde el **MPLAB IDE**, el **“PICkit 2”** elevará la línea MCLR, para que corra el programa en el MCU hasta que alcance la dirección de programa guardada en los registros internos, que contienen la dirección del breakpoint.

Al alcanzar esta dirección se dispara en el MCU un mecanismo similar a una interrupción que ejecuta una **rutina llamada “Debug Executive”**. La aplicación del usuario queda detenida. El **“PICkit 2”** se comunica con esta Debug Executive y le envía la información al **MPLAB IDE**. Este último, a través de una serie de consultas con el **“PICkit 2”**, recupera información de los registros en RAM y del STATUS.

El Debug Executive al ser un rutina que corre en memoria de programa, consume algunas posiciones de FLASH, algunas posiciones de la pila (1 ó 2) y cerca de 14 registros de RAM. Al ser una rutina de FLASH, se explica porque no se puede debuggear un MCU cuyo oscilador no funciona o que no tiene alimentación.

8.3 Diferencias entre el modo Debug y el modo Release

Al trabajar en modo Debug también debemos grabar en el MCU el firmware de aplicación. Sin embargo, si energizamos la placa de aplicación sin conectarle el “PICkit 2”, el programa **no correrá**. Si deseamos ejecutar el firmware independientemente del “PICkit 2” es necesario grabar el firmware en **modo Release**, que es la forma en que correrán las aplicaciones en última instancia.

8.4 Limitaciones del modo Debug

- El “PICkit 2” utiliza varios recursos del MCU mientras está en **modo Debug**.
- **Al modificarse el código fuente de un programa, el MCU debe ser reprogramado con las actualizaciones.**
- Al enviarse un comando de **RESET** al MCU emulado, el **Program Counter** vuelve al valor **0x0000** y la ejecución se detiene allí. El programa **no se re-ejecuta automáticamente**.
- La cantidad de **breakpoints simultáneos** es limitada.
- Puede ocurrir un fenómeno conocido como **breakpoint skidding**.
- **No puede usarse** la instrucción **SLEEP** durante el Debug.
- **No puede usarse** el **watchdog timer** durante el Debug.
- La **pila no es visible** aunque pueda accederse a ella.
- **No pueden correrse programas a una frecuencia inferior a 32KHz.**

9.0 Familias de MCU y el “PICkit 2”

A continuación mencionamos características comunes a cada familia de MCU de Microchip. Si bien podemos encontrar excepciones, esta información es útil a la hora de elegir un PIC para realizar un diseño y también como una guía a la hora en que surgen los problemas.

9.1 Familia PIC10FXXX

Esta familia de microcontroladores puede ser programada con el “PICkit 2” sin problemas empleando la conexión a **5 hilos**. Si se desea debuggear un firmware para estos microcontroladores es necesario emplear una **Header Board** adicional. Leer el documento correspondiente desde el CD-ROM de cortesía.

9.2 Familia PIC12FXXX

Esta familia de microcontroladores puede ser programada con el “PICkit 2” sin problemas empleando la conexión a **5 hilos**. Si se desea debuggear un firmware para estos microcontroladores es necesario emplear una **Header Board** adicional. Leer el documento correspondiente desde el CD-ROM de cortesía.

9.3 Familia PIC16FXXX

Esta familia de microcontroladores puede ser programada con el “PICkit 2” empleando la conexión a 5 hilos. Si se desea debuggear un firmware para estos microcontroladores es necesario emplear una **Header Board** adicional. Leer el documento correspondiente desde el CD-ROM de cortesía.

9.4 Familia PIC18FXXX

Esta familia de microcontroladores puede ser programada con el “PICkit 2” empleando la conexión a **5 hilos**. Con la misma conexión es posible debuggear un firmware cargado en el microcontrolador. En algunos dispositivos es posible que el convertor A/D tenga su propia alimentación (AVDD y AVSS). Para una correcta detección, programación y debug del mismo es necesario que todas las alimentaciones, analógicas y digitales, estén debidamente conectadas.

10.0 – Preguntas mas frecuentes FAQ.

¿Porqué cuando intento hacer entrar al modo Debug a un PIC y obtengo el siguiente mensaje?:

PK2Error0028: Unable to enter debug mode

NOTE: This device requires an ICD Header for debug. See "Header Specification" DS51292.

Ello se debe a que el PIC utilizado está catalogado como un **microcontrolador de pocos pines**. Si desea hacerlo entrar en modo Debug necesita de una plaqueta adaptadora llamada **Header Board**. Para mas detalles lea el documento **Header Board Specification** que se encuentra en el CD-ROM de cortesía en la ruta:

D:\PICkit 2

¿Porqué el “*PICkit 2*” no detecta el MCU cuando quiero programarlo?

Existen múltiples razones:

- El MCU no recibe alimentación ni del “*PICkit 2*” ni externa.
- No se han respetado las indicaciones dada en el **punto 6.2**.
- No se ha llevado a **nivel bajo el pin PGM** (en los MCU que cuentan con este pin).
- Se ha seleccionado el MCU equivocado desde el **MPLAB IDE** (el **MPLAB IDE** indica el código de MCU detectado. Si este es **0x0000** o **0xFFFF**, entonces **no se lo ha detectado** correctamente. Si el código es **otro**, lo mas probable que **haya detectado un dispositivo diferente** al seleccionado).

¿Cuál es la razón por la que el MCU es detectado, programado o entra en modo Debug solo alguna veces?

Habitualmente este problema se debe a que no se ha llevado a nivel bajo la **señal de PGM**.

¿A qué se debe que el **MPLAB IDE** me informe un cortocircuito en **VDD** si todas las conexiones entre el “*PICkit 2*” y el MCU es correcta?

Habitualmente se debe a que construimos una fuente de alimentación en la plaqueta de alimentación pero luego intentamos alimentar el MCU desde el “*PICkit 2*”. Para este caso se debería deshabilitar la alimentación del “*PICkit 2*”.

Grabé la FLASH del “PICkit 2” desde el modo Debug. Remuevo el “PICkit 2”, alimento la plaqueta de aplicación pero esta no funciona ¿A qué se debe?

Para correr una aplicación sin el “PICkit 2” como intermediario debe grabarse el firmware en **modo Release**.

¿Cómo sé si el “PICkit 2” puede programar al PIC que estoy utilizando?

Desde el **MPLAB IDE**, desde la barra de menú se selecciona **Configure→Select Device**. Se busca en la lista de dispositivos al MCU en cuestión. En el grupo “**Programmers**” veremos al “**PICkit 2**” junto con un círculo que puede ser rojo, amarillo ó verde. Si es verde el “**PICkit 2**” permite programar al MCU elegido.

¿Cómo sé si el “PICkit 2” puede debugear al PIC que estoy utilizando?

Desde el **MPLAB IDE**, desde la barra de menú se selecciona **Configure→Select Device**. Se busca en la lista de dispositivos al MCU en cuestión. En el grupo “**Debuggers**” veremos al “**PICkit 2**” junto con un círculo que puede ser rojo, amarillo ó verde. Si es verde el “**PICkit 2**” permite debugear al MCU elegido.

11.0 Layout

Placa “PICkit 2”

