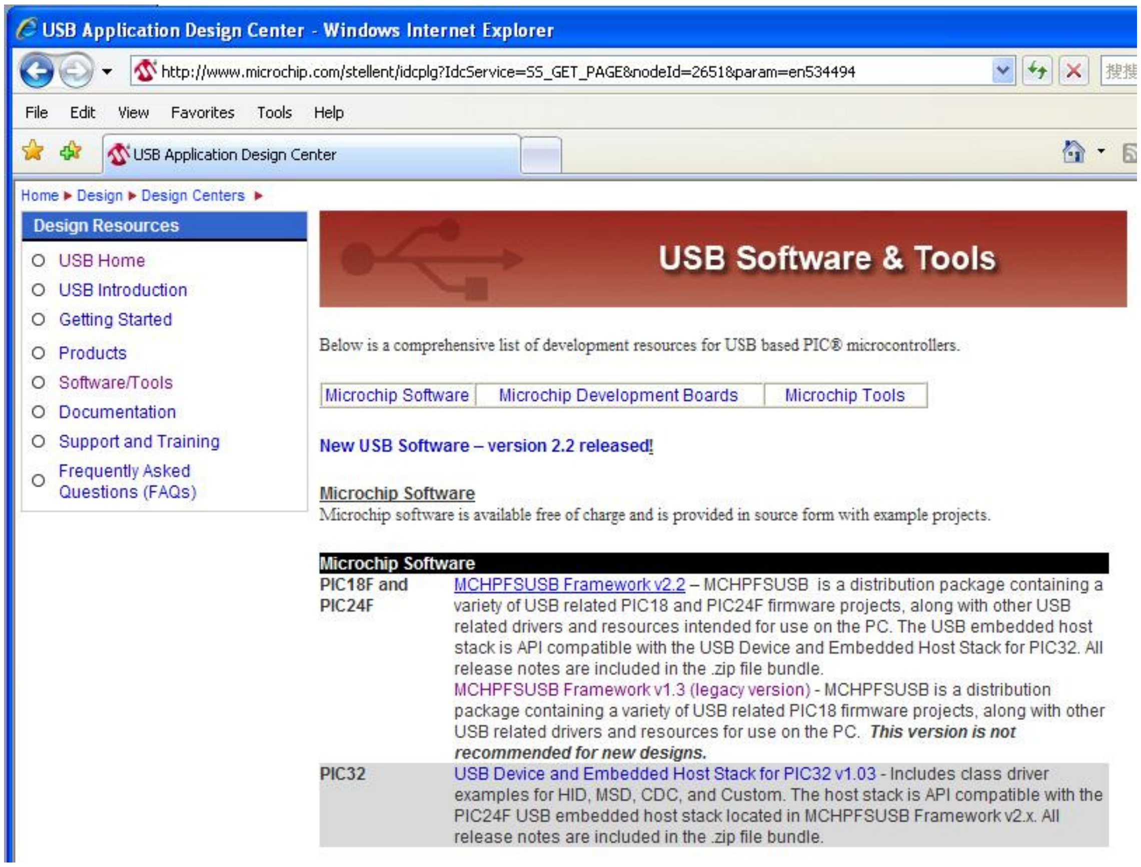
User Guide for USB Applications

Development board for PIC18LF4550

PROCEDURE

MCHPFSUSB Framework v2.2 is a distribution package containing a variety of USB related PIC18 and PIC24 firmware projects. Its web site is located at

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2651&param=en534494


USB Application Design Center - Windows Internet Explorer
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2651&param=en534494
File Edit View Favorites Tools Help
USB Application Design Center
Home ▸ Design ▸ Design Centers ▸
Design Resources
USB Home
USB Introduction
Getting Started
Products
Software/Tools
Documentation
Support and Training
Frequently Asked Questions (FAQs)
USB Software & Tools
Microchip Software
Microchip Development Boards
Microchip Tools

Below is a comprehensive list of development resources for USB based PIC® microcontrollers.

New USB Software – version 2.2 released!

Microchip Software

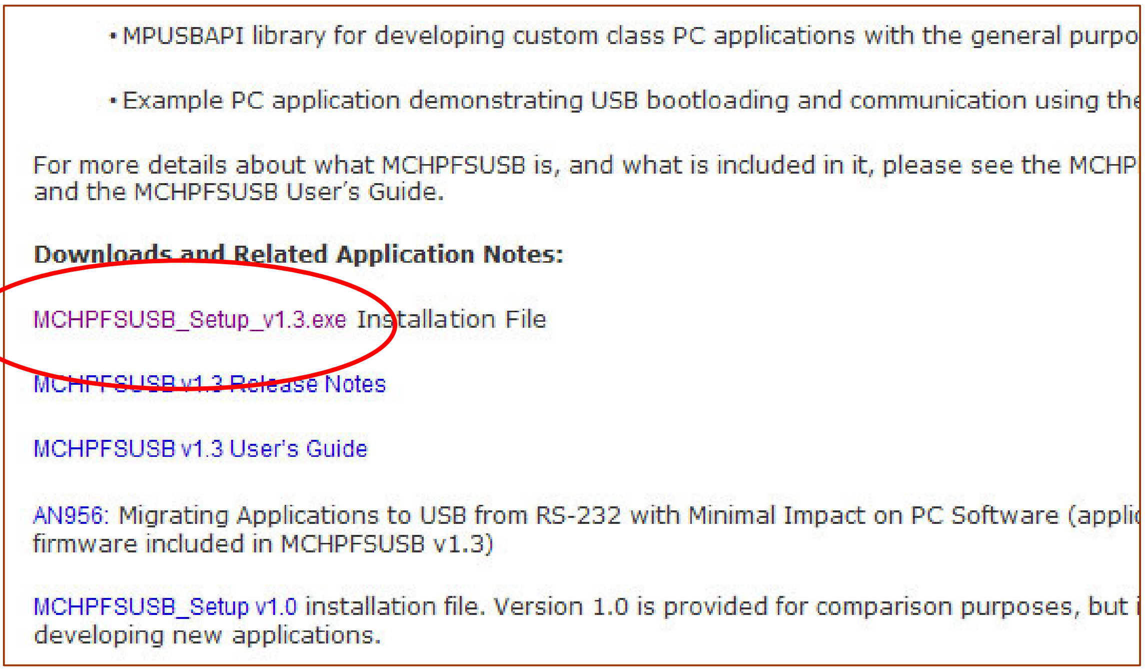
Microchip software is available free of charge and is provided in source form with example projects.

Microchip Software	
PIC18F and PIC24F	MCHPFSUSB Framework v2.2 – MCHPFSUSB is a distribution package containing a variety of USB related PIC18 and PIC24F firmware projects, along with other USB related drivers and resources intended for use on the PC. The USB embedded host stack is API compatible with the USB Device and Embedded Host Stack for PIC32. All release notes are included in the .zip file bundle. MCHPFSUSB Framework v1.3 (legacy version) - MCHPFSUSB is a distribution package containing a variety of USB related PIC18 firmware projects, along with other USB related drivers and resources for use on the PC. ***This version is not recommended for new designs.***
PIC32	USB Device and Embedded Host Stack for PIC32 v1.03 - Includes class driver examples for HID, MSD, CDC, and Custom. The host stack is API compatible with the PIC24F USB embedded host stack located in MCHPFSUSB Framework v2.x. All release notes are included in the .zip file bundle.

Download and extract MCHPFSUSB Framework v2.2 to your local hard drive. If you accept the default installation path, projects for PIC18 and PIC24 will be installed at C:\Microchip Solutions with USB xxx as the prefix.

Afterwards, download the USB driver for Microchip devices from this link. At time of writing, the latest version of MCHPFSUSB is version 1.3.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2124&param=en532204&page=wwwFullSpeedUSB


• MPUSBAPI library for developing custom class PC applications with the general purpo

• Example PC application demonstrating USB bootloading and communication using the

For more details about what MCHPFSUSB is, and what is included in it, please see the MCHP and the MCHPFSUSB User's Guide.

Downloads and Related Application Notes:

MCHPFSUSB_Setup_v1.3.exe Installation File

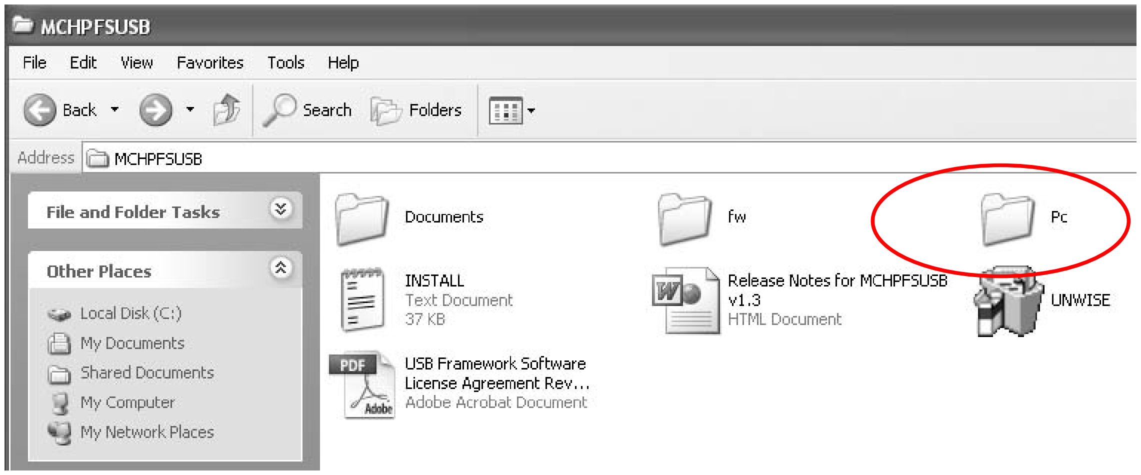
MCHPFSUSB v1.3 Release Notes

MCHPFSUSB v1.3 User's Guide

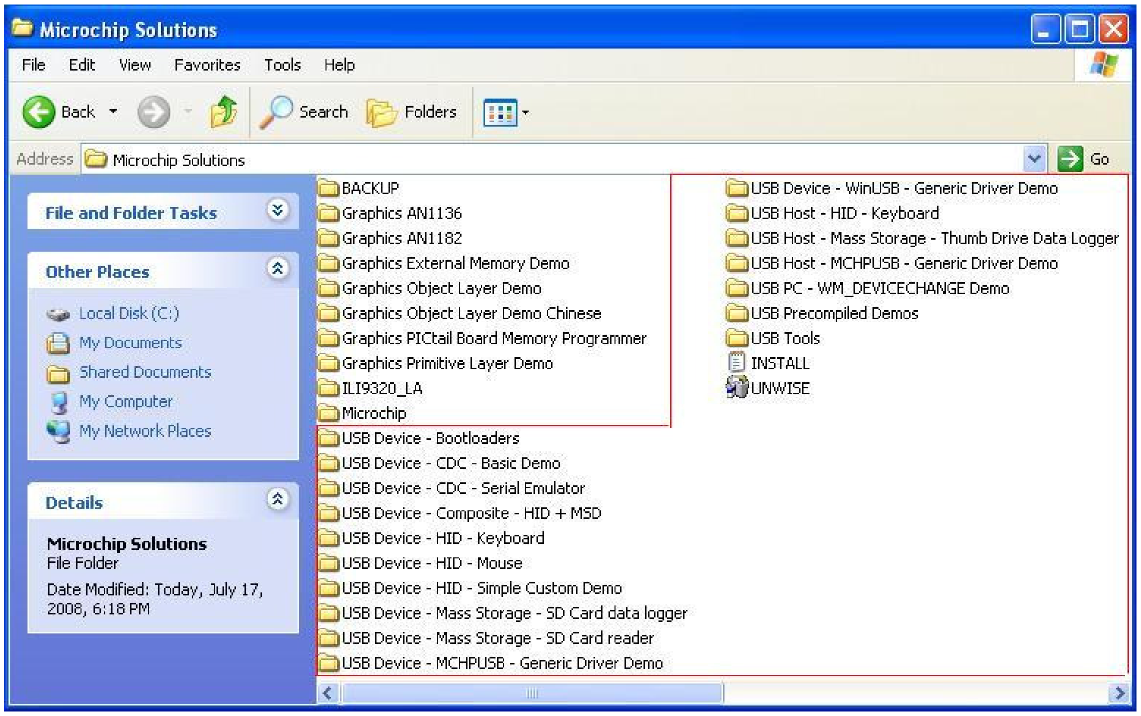
AN956: Migrating Applications to USB from RS-232 with Minimal Impact on PC Software (applic firmware included in MCHPFSUSB v1.3)

MCHPFSUSB_Setup v1.0 installation file. Version 1.0 is provided for comparison purposes, but i developing new applications.

This Setup program will install MCHPFSUSB which is the Microchip device drivers for the PC side plus few demonstration programs for hid, cdc device etc. The default installation path is C:\MCHPFUSB. If it happens that you have installed an old version of it, the installation wizard will prompt you to remove the old version prior to an update. What matters is the Pc folder under C:\MCHPFUSB which contains the USB driver. We will need this folder later.



Launch MPLAB IDE v8.02 or above to open these projects. Screenshot below highlights all folders relevant to USB application.


Microchip Solutions
File Edit View Favorites Tools Help
Back
Search
Folders
Address Microchip Solutions
Go
File and Folder Tasks
Other Places
Local Disk (C:)
My Documents
Shared Documents
My Computer
My Network Places
Details
Microchip Solutions
File Folder
Date Modified: Today, July 17, 2008, 6:18 PM
BACKUP
Graphics AN1136
Graphics AN1182
Graphics External Memory Demo
Graphics Object Layer Demo
Graphics Object Layer Demo Chinese
Graphics PICtail Board Memory Programmer
Graphics Primitive Layer Demo
ILI9320_LA
Microchip
USB Device - Bootloaders
USB Device - CDC - Basic Demo
USB Device - CDC - Serial Emulator
USB Device - Composite - HID + MSD
USB Device - HID - Keyboard
USB Device - HID - Mouse
USB Device - HID - Simple Custom Demo
USB Device - Mass Storage - SD Card data logger
USB Device - Mass Storage - SD Card reader
USB Device - MCHPUSB - Generic Driver Demo
USB Device - WinUSB - Generic Driver Demo
USB Host - HID - Keyboard
USB Host - Mass Storage - Thumb Drive Data Logger
USB Host - MCHPUSB - Generic Driver Demo
USB PC - WM_DEVICECHANGE Demo
USB Precompiled Demos
USB Tools
INSTALL
UNWISE

Let's start with the firmware project USB Device – Bootloaders. Launch MPLAB, open project from
Project →
HID – Bootloader →
HID Bootloader – Firmware for PIC18 Non-J Devices.

Right from the source file BootPIC18NonJ.c we can see that this project has been built for PIC18 series as follows (extract from the source code).

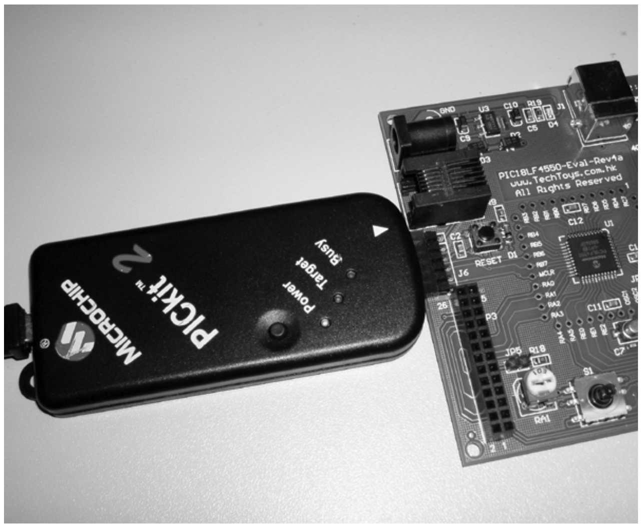
```
*       Microchip USB HID Bootloader for PIC18F and PIC18LF versions of:
*       PIC18F4553/4458/2553/2458
*       PIC18F4550/4455/2550/2455
*       PIC18F4450/2450
*
*********************************************************************
* FileName:         BootPIC18NonJ.c
* Dependencies:     See INCLUDES section below
* Processor:        PIC18
* Compiler:         C18 3.20+
* Company:          Microchip Technology, Inc.
```

First, check the compilation environment to see if it suits your installation path of C18. The author uses D: drive for C18 installation which most probably will be different from yours.

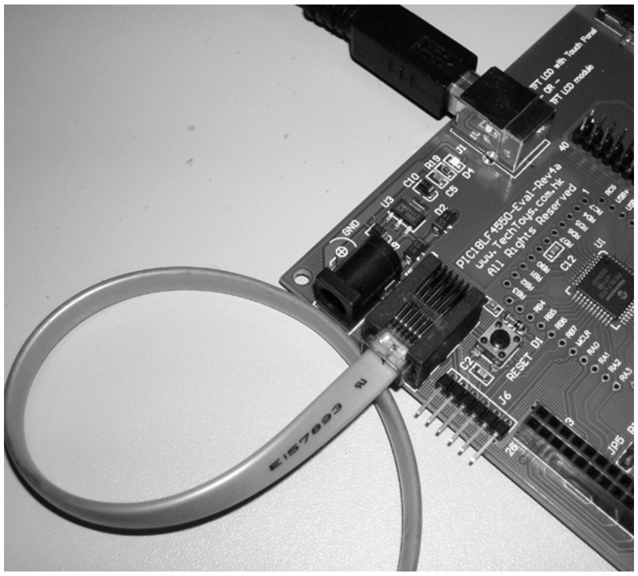
Under the main menu, click on ***Project*** → ***Build Options*** → ***Project*** to bring up the Build Options Window. Click on the ***Directories*** tab. There is a ***Show directories for:*** pull-down menu containing three important options being ***Include Search Path*** (define the search path for *.h header files), ***Library Search path***, and ***Linker-Script Search path***.

If you have accepted the default installation path for C18, the relevant directories will be C:\MCC18\h, C:\MCC18\lib, and C:\MCC18\lkr. Because the author uses D: drive for C18, the paths D:MCC18\h, D:MCC18\lib, and D:MCC18\lkr have been used.

From ***Project*** main menu, click on ***Build All (Ctrl+F10)*** to build the program. The hex code generated is located under the project folder with file name
HID Bootloader PIC18 Non J.hex. One may program the microcontroller with this hex code by an ICD2 debugger or PICKit2 Programmer. Details of these devices could be found under www.Microchip.com. Powering options are USB cable or a 5V DC power jack. A USB cable is preferred for simplicity.

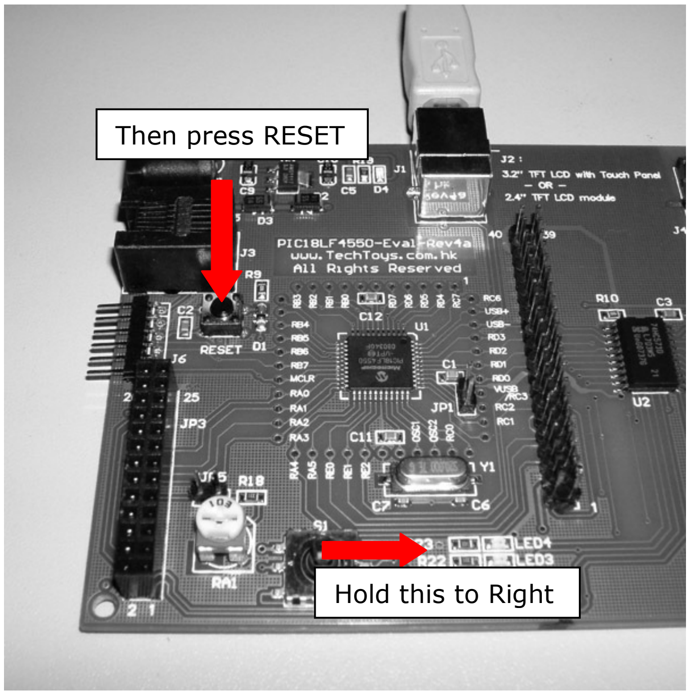


PICKit 2 connection

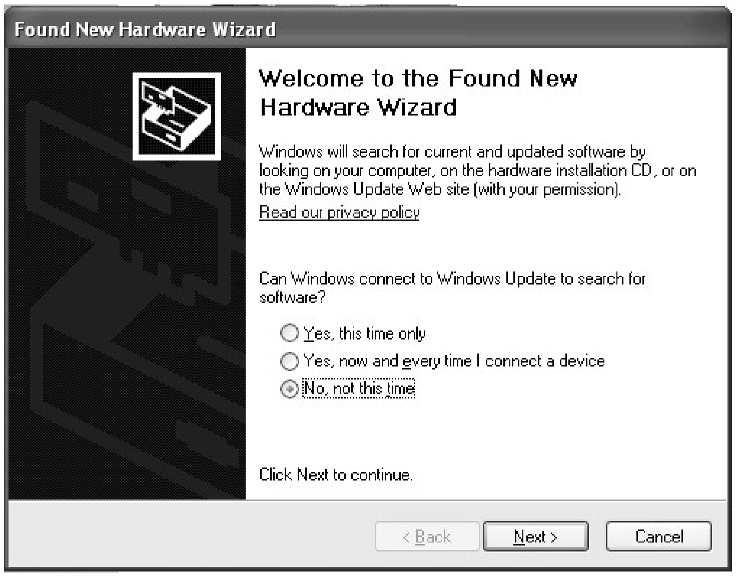


ICD2 connection

Having downloaded the bootloader code, we need to wake up the microcontroller with the bootloader code running (it is not a simple reset!). Hold the 5-way navigator joystick right and press reset.



The microcontroller will be reset to bootloader mode. If it is the first time you are running this application, your workstation will be prompt to ***New Hardware Found Wizard*** with Windows Driver Update screen as below. Click on ***No, not this time*** and click ***Next*** to proceed. Select ***Install from a list or specific location (Advanced)*** option in the next page.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

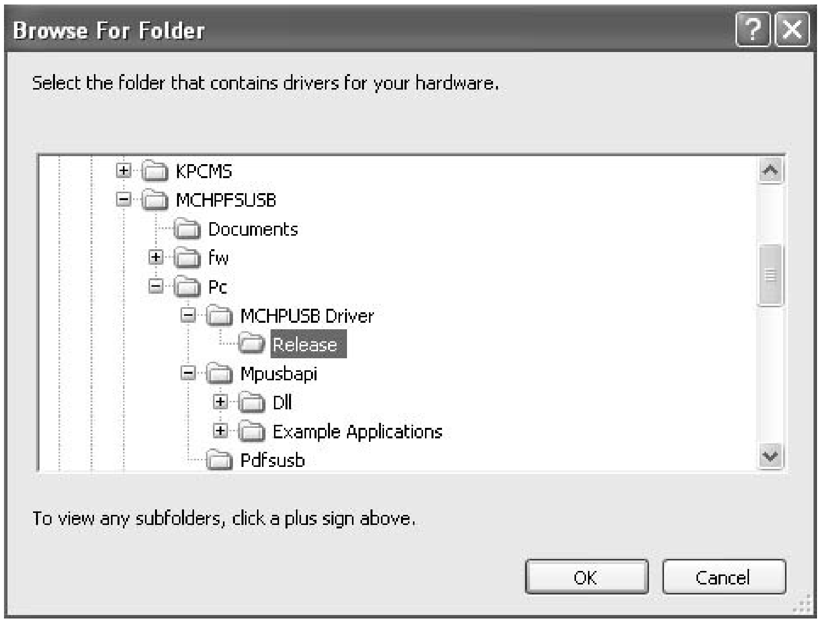
Read our privacy policy

Can Windows connect to Windows Update to search for software?

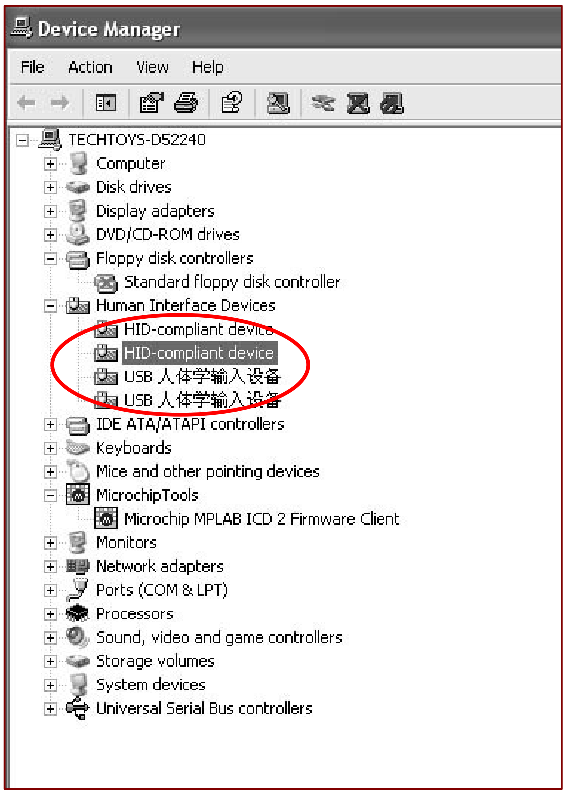
- Yes, this time only
- Yes, now and every time I connect a device
- [✓ No, not this time]

Click Next to continue.

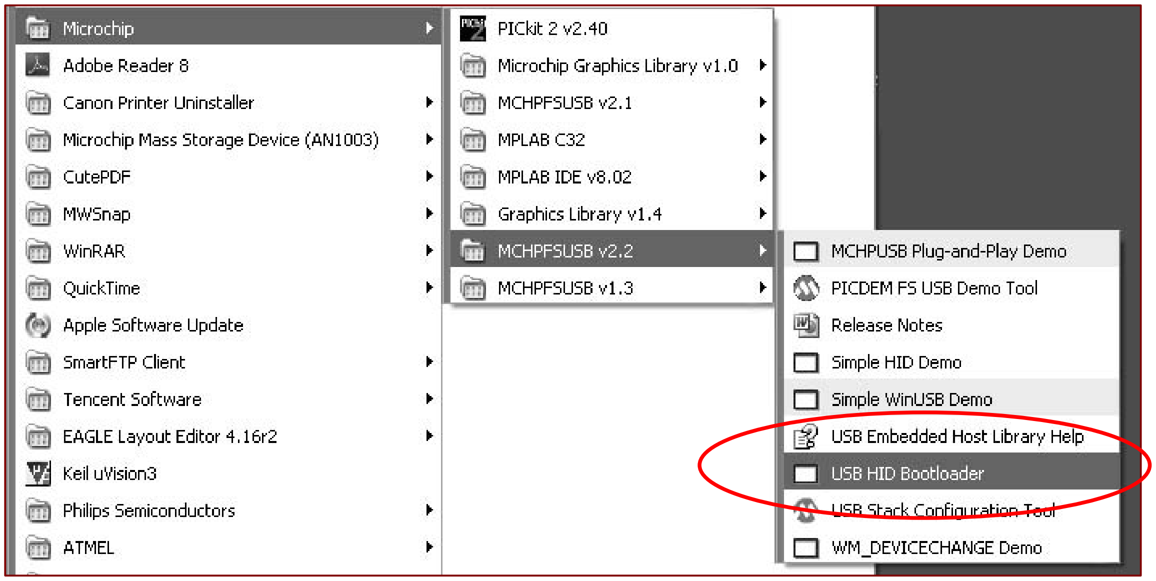
Browse to the folder ***C:\MCHPFSUSB\Pc\MCHPUSB Driver\Release*** and click OK. Click ***Next*** and continue installation by ignoring all warnings. This procedure will happen one more time for HID-compliant device. Just go ahead to install them.



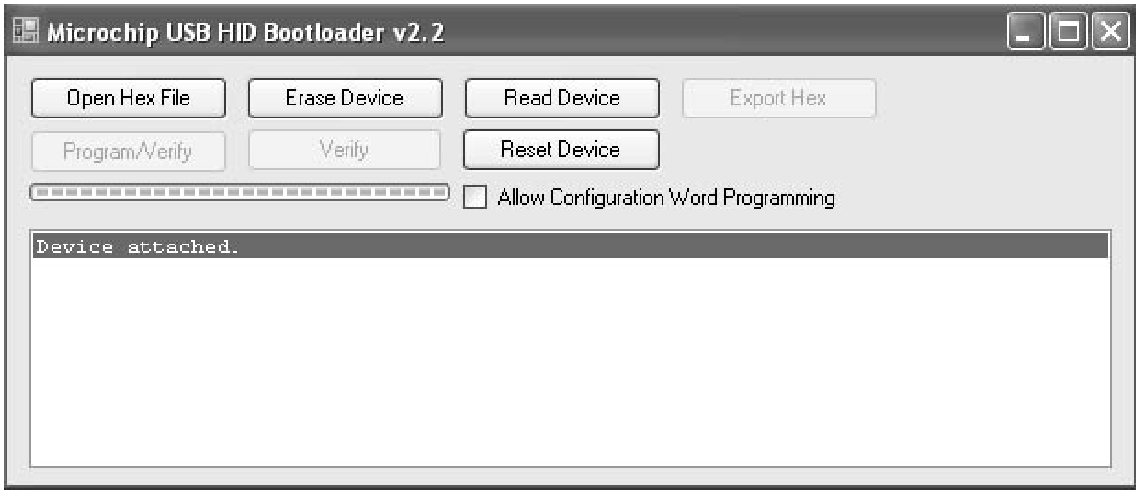
If everything goes all right, you will see a new set of HID-complaint device and a USB Human Input Device (sorry about the Chinese) in the Device Manager.



This is something new with this version. From Windows Start, browse to Microchip → MCHPFUSB v2.2 → USB HID Bootloader and launch this application.



This is a new application released by Microchip for hex code download. From this application you may perform Erase Device, Hex code download etc.



```
Device attached.
```

Click Open Hex File, browse to C:\Microchip Solutions\USB Device – HID – Mouse\ and select this very long file name

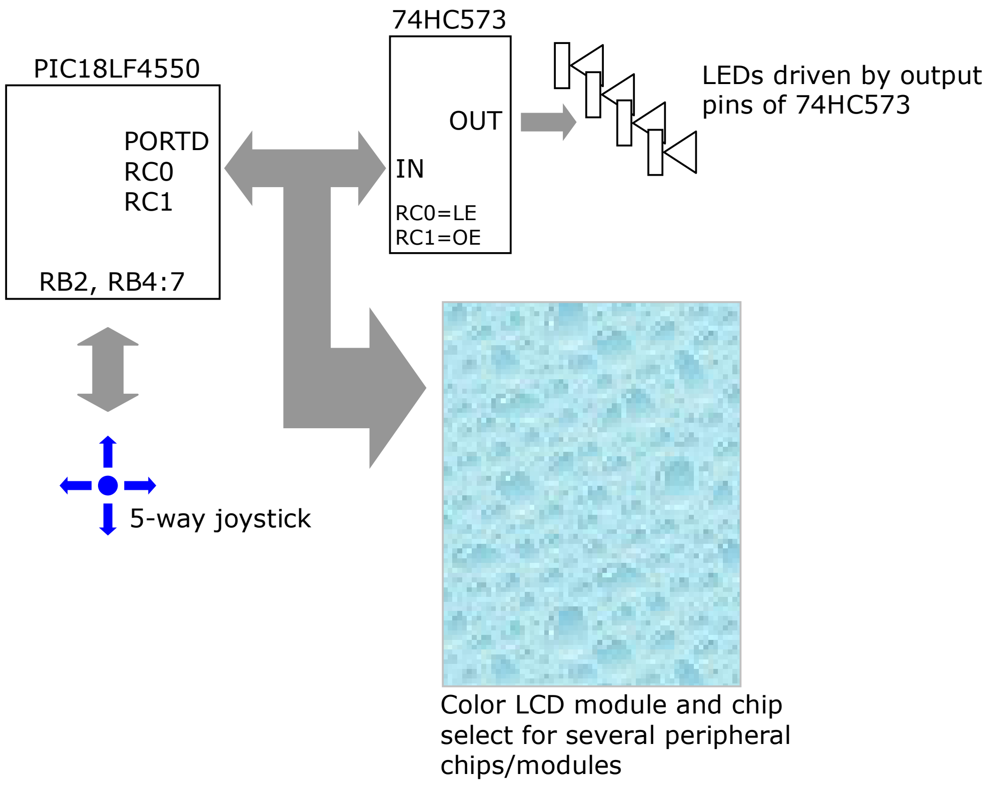
USB Device - HID - Mouse - C18 - PICDEM FSUSB - HID Bootload.hex.

Click on ***Program/Verify***. Watch the message window for "Erase/Program/Verify Completed Successfully" and then press RESET key onboard to start this HID Mouse application. Immediately you will see your mouse cursor is circling like crazy! This is the HID example for an USB mouse out-of-the-box! If you want to quit, simply hold the joystick Right, and press RESET again to bring up the bootloader program for the next demo. NO NEED TO UNPLUG USB cable.

WHAT HAPPEN TO THE LEDs?

From last section you may have noticed that nothing was happening with all those LEDs (LED1-LED4) and there is no response from the 5-way joystick either. This scenario is a bit different from the result obtained by the original PICDEM™ FS USB Demo Board. It is because the schematic of PIC18LF4550-Eval-Rev 4A is different. LEDs are now driven by four output pins of the latch device 74HC573 as an I/O expander, plus there is no pull-up resistor for input RB4 and RB5. Furthermore, pins RB6, RB7, and RB2 have been used for more input options. They are connected to a 5-way navigator joystick as an input control. A block diagram below illustrates the idea. For further details please refer to schematic under Doc 01 at

http://www.techtoys.com.hk/PIC_boards/PIC18LF4550-Eval-Rev4A/PIC18LF4550-Rev4A.htm.


74HC573
PIC18LF4550
PORTD
RC0
RC1
RB2, RB4:7
OUT
IN
RC0=LE
RC1=OE
LEDs driven by output pins of 74HC573
5-way joystick
Color LCD module and chip select for several peripheral chips/modules

There was no latch device in previous versions. However, including a low-cost 74HC573 latch will double PORTD to 16 pins with just two controlling pins RC0, and RC1 for LE and OE extra. The down side is that it is not possible to directly use all demo code for PICDEM™ FS USB Demo Board without slight modification.

Use the same HID USB Mouse example in last section. Open the workspace from directory C:\Microchip Solutions\USB Device - HID - Mouse\HID - Mouse – Firmware\ USB Device - HID - Mouse - C18 - PICDEM FSUSB.mcp.

Navigate to the project panel at the left and open the header file HardwareProfile.h. This is the only file required. Relevant code with comment is listed on next page.

```
#ifndef HARDWARE_PROFILE_H
#define HARDWARE_PROFILE_H

….

#if defined(PICDEM_FS_USB)
….
//#define mInitAllLEDs()              LATD &= 0xF0; TRISD &= 0xF0;                              (1)
//Set RC0, RC1 output high and low for 74HC573D latch for PIC18LF4550-Eval-Rev4A board
#define mInitAllLEDs()                LATD |= 0x0F; TRISD &= 0xF0; LATCbits.LATC1 = 0; \        (2)
                                      LATCbits.LATC0 = 1; TRISC &= 0xFC

#define mLED_1           LATDbits.LATD0                                                         (3)
#define mLED_2           LATDbits.LATD1
#define mLED_3           LATDbits.LATD2
#define mLED_4           LATDbits.LATD3

#define mLED_1_On()        mLED_1 = 0;                                                          (4)
#define mLED_2_On()        mLED_2 = 0;
#define mLED_3_On()        mLED_3 = 0;
#define mLED_4_On()        mLED_4 = 0;

#define mLED_1_Off()       mLED_1 = 1;                                                          (5)
#define mLED_2_Off()       mLED_2 = 1;
#define mLED_3_Off()       mLED_3 = 1;
#define mLED_4_Off()       mLED_4 = 1;
…

//#define mInitAllSwitches()  TRISBbits.TRISB4=1;TRISBbits.TRISB5=1;                            (6)
//enable weak pull-up for PORTB with INTCON2bits.RBPU = 0 for PIC18LF4550-Eval-Rev4A board

#define mInitAllSwitches()            TRISBbits.TRISB4=1;TRISBbits.TRISB5=1; \                  (7)
                                      INTCON2bits.RBPU = 0;
#define mInitSwitch2()                TRISBbits.TRISB4=1;
#define mInitSwitch3()                TRISBbits.TRISB5=1;
#define sw2                           PORTBbits.RB4
#define sw3                           PORTBbits.RB5
….
```

An extract of HardwareProfile.h to show the relevant code to be replaced

Line (1)	First, comment this original definition for LATD[3:0]
Line (2)	Add this line under line (1). We need to set LATD[3:0] high to turn off all LEDs. Then we need to set RC0 (LE) high and RC1 (OE) low for 74HC573. Setting OE low will enable 74HC573. Setting LE high will release the latch therefore OUT[7:0] just follow RD[7:0].
Line (3)	Starting from this line until line (4) define mLED_X for LATDbits.LATDX. Nothing has been changed.
Line (4)	Starting from this line define LED ON macros. Because we are using current sink method, an output low turns on LEDx.
Line (5)	Similar to LED ON macros, we need to set mLED_X = 1 to turn them off.
Line (6)	Comment this line.
Line (7)	Because there is no external pull-up resistors, we need to enable the weak pull-up option for PORTB. Starting from the next line defines simple switches sw2 & sw3. If one wants to use RB2, RB6, and RB7 for input, just follow the definition e.g. #define sw4 PORTBbits.RB5, etc.

Again, by compile and download this program you will see LED1 and LED2 are blinking, and repeat clicking the joystick down will start/stop USB mouse circling.