Tutorial 2: In Circuit Serial Programming (ICSP) and Bootloaders

I. INTRODUCTION

Programming PIC devices is a time consuming procedure. Most of the time, the codes you develop do not work as desired and you need to program your device over and over correcting the mistakes and bugs in the firmware implementation. In the old-fashioned programming style, you unplug your microcontroller from your application circuit, plug it into a JDM programmer (search the internet) and look for a PC which has a serial COM port with RS232 interface. You not only lose time but also your motivation during that procedure.

The objective of this tutorial is to introduce the concept of in circuit programming and some of the tools in which you don't need to remove your device from the application circuit to program it. Instead, you can program it serially with a slight modification on the application circuit. You do not even need a COM port (may be once). By implementing your own programmer once (or you may prefer buying it), you can program almost any PIC with little effort. Furthermore, programming PICs with these tools will be much faster and more reliable than a simple JDM programmer.

There are mainly two firmware programming tools covered in this tutorial, namely the PICKit2 and MCHP framework bootloaders, both of which are the free products of Microchip®. These will be briefly explained in the following sections. You are also encouraged to build up your own PICKit2 programmers which will cost you much less than when you buy it from the market.

Prerequisites:

No prerequisites needed.

II. PICKit 2

PICKit2 is a free PIC development tool produced by Microchip®. It is composed of a programming hardware and a software application running on PC. The software is also supported by MPLAB as PIC programmer tool and once you install it, you can program your devices from MPLAB using the PICKit2 interface.

To begin with, download the package "Pickit2 hardware_firmware.rar" from the tutorials page of the course. This is a complete clone of the actual PICKit2 device and you can also follow up the explanations in http://www.mcuhobby.com/articles.php?article_id=7 to build this device.

Print the board in the file "PicKit2-1-mod.pdf" with a **laser printer** (preferably a high quality one). Drill the holes and etch the board as explained in the tutorial on "Simple PCB making". Assemble the components as shown in the files "pk21-mod-brd2.png" and "pk21-mod-sch.png". At the end, you should have a PICKit2 clone similar to "pk21_complete.png". You may change the USB connector type if you like but in that case

you should also change the USB cable or you should make the connections by yourself. **Therefore, be careful when assembling the USB connector part**.

Now, in order to run this device, you should once program its firmware at first. In order to do that you will need another programmer called JDM (a programmer to program a programmer). It is a commonly used PIC programmer and you can find it everywhere. So you can either buy it or build it by yourself (You will need the JDM only once, so it is better to borrow it from someone else or build it together, **you can find the circuit schematics of JDM programmer in the course webpage**). Assuming that you have the JDM hardware, download the Winpigpgm (a PIC programming software) from http://members.aon.at/electronics/pic/picpgm/download.html

Install the program on a **desktop PC** which has a serial COM port with RS232 interface. Plug the device PIC18F2550 on the JDM programmer. You should locate the PIC18F2550 device to the middle layer of the JDM programmer such that the first pin of the device (top left pin, numbered as 1) corresponds to the first pin of the JDM programmer.

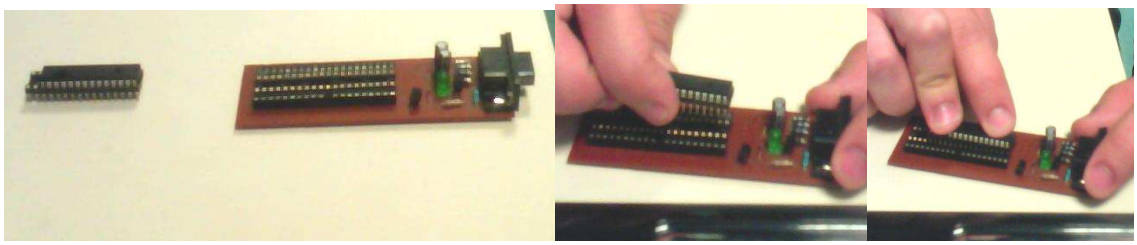

Figure 1: Programming the PICKit2 firmware to 18F2553 device with the JDM Programmer. Proper placement of the device onto the programmer.

Plug the JDM programmer to one of the COM ports and run "PICPgm Programmer" application.

Figure 2: WinPicPgm working with the JDM Programmer

The software will automatically detect your device (if not then your device or JDM programmer may not be working properly). Once the JDM programmer is identified by the PICPgm software, browse for the PICKit2 firmware "PK2V023200.hex" which is provided in the "Pickit2 hardware_firmware.rar" package. Program the device (whether the device is

18F2550 or not is not so important. You can use a PIC device in the same family as 18F2550. Here, the PICKit2 firmware is programmed into 18F2553) and wait for the verification. If the verification fails, then try until you see that "programming successful".

Plug your device to the PICKit2 hardware. Now you have the PICKit2 firmware programmed into your programming device. For the PICKit2 software, download the PICKit2 installation package (final release is v2.61) from the web site (in the downloads section):
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023805&redirects=pickit2

If your system does not have the .NET framework previously installed, download "PICKit2 v2.61 Install with .NET framework". From the package, run "PICKit2Setup.exe" and install it to your system with the default settings.

You can see the user interface of the software in Figure 3.

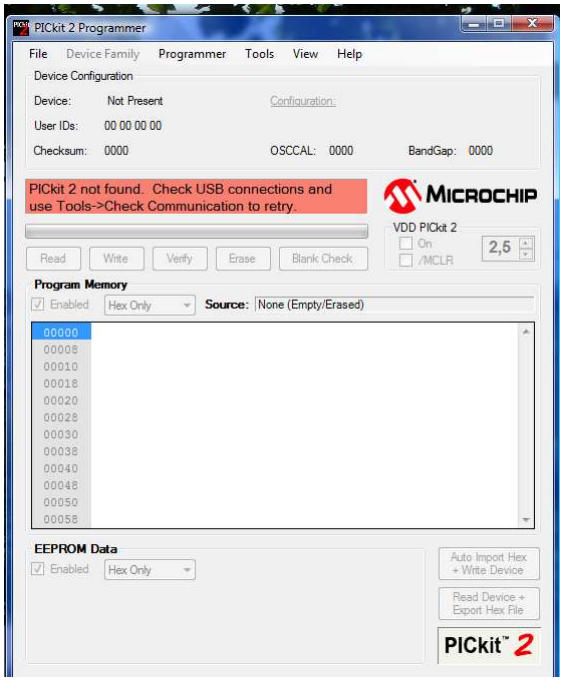

Figure 3: PICKit2 – User Interface

Here, the program waits for the PICKit2 hardware to be connected to your system. Assuming that you already have the PICKit2 hardware, you can connect it to one of the USB ports of your PC and your system will recognize the device as a PICKit2 device. In the software, click on "Tools" from the menu bar and select "Check Communication".

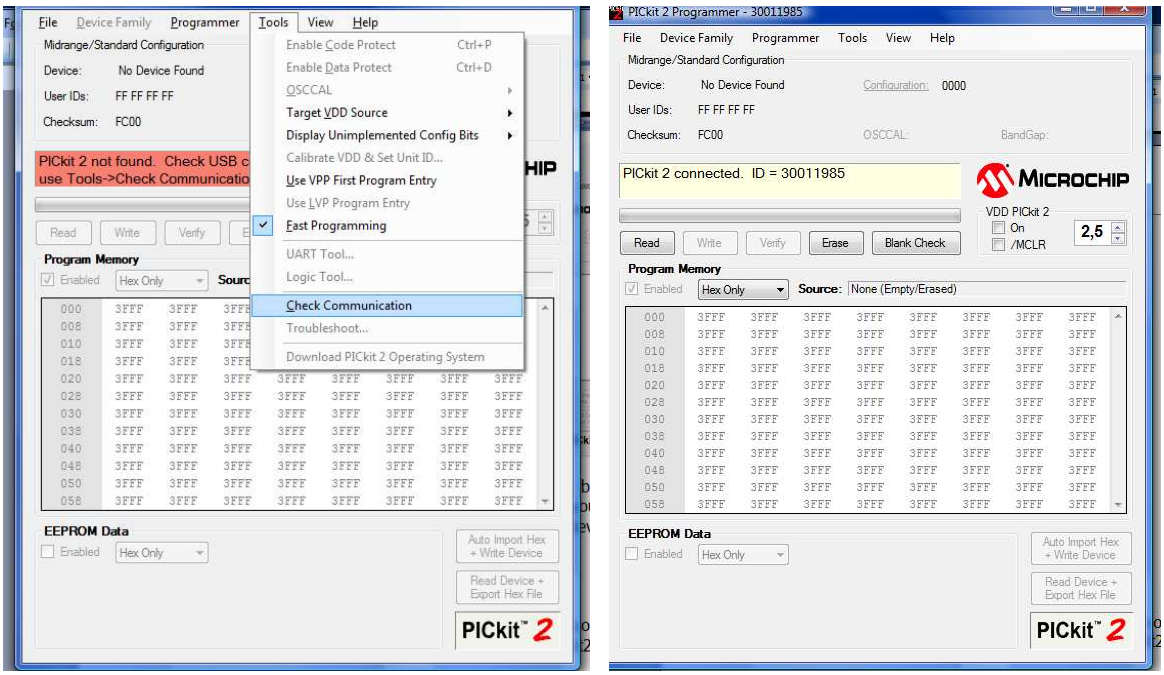

Figure 4: PICKit2 – Setting up the hardware communication

Once you see "PICKit2 connected" on the software, you will have nothing to do with the JDM programmer and you can uninstall the winpicpgm from your system if you like.

You can program almost every PIC device with PICKit2 not only from USB (which is available in every computers of today), but also without removing your PIC from the application circuit. There are also other powerful features of PICKit2 like **debugging** which is the topic of the tutorial on "Firmware Debugging with PICKit2". To see the list of supported devices see the ReadMe file (Click on Help and select ReadMe). You can also access the guiding documents about using PICKit2 from the "Help" in the menu bar.

As an example let's program some firmware to a device. To link this part to the rest of this document, let's program a bootloader firmware to PIC18F4553. You will see what a bootloader does in the related section.

In MPLAB, open the project MCHPUSB.mcp located in C:\Microchip Solutions\USB Device - Bootloaders\Vendor Class - MCHPUSB Bootloader\Bootloader - Firmware for PIC18F4550 Family Devices

Build the project and wait until you see that "BUILD SUCCEEDED". If you encounter an error in the compilation (and by the time if you did not modify the code) then that means your C18 evaluation period has expired or you are using the academic (lite) version of the MPLAB C18 compiler. In that case, the optimizations on memory management are disabled during the compilation (your code can not fit to the memory specifications on the linker file) and you should try building the bootloader project in another computer whose evaluation period still continues.

Now construct the following circuit to program the bootloader.

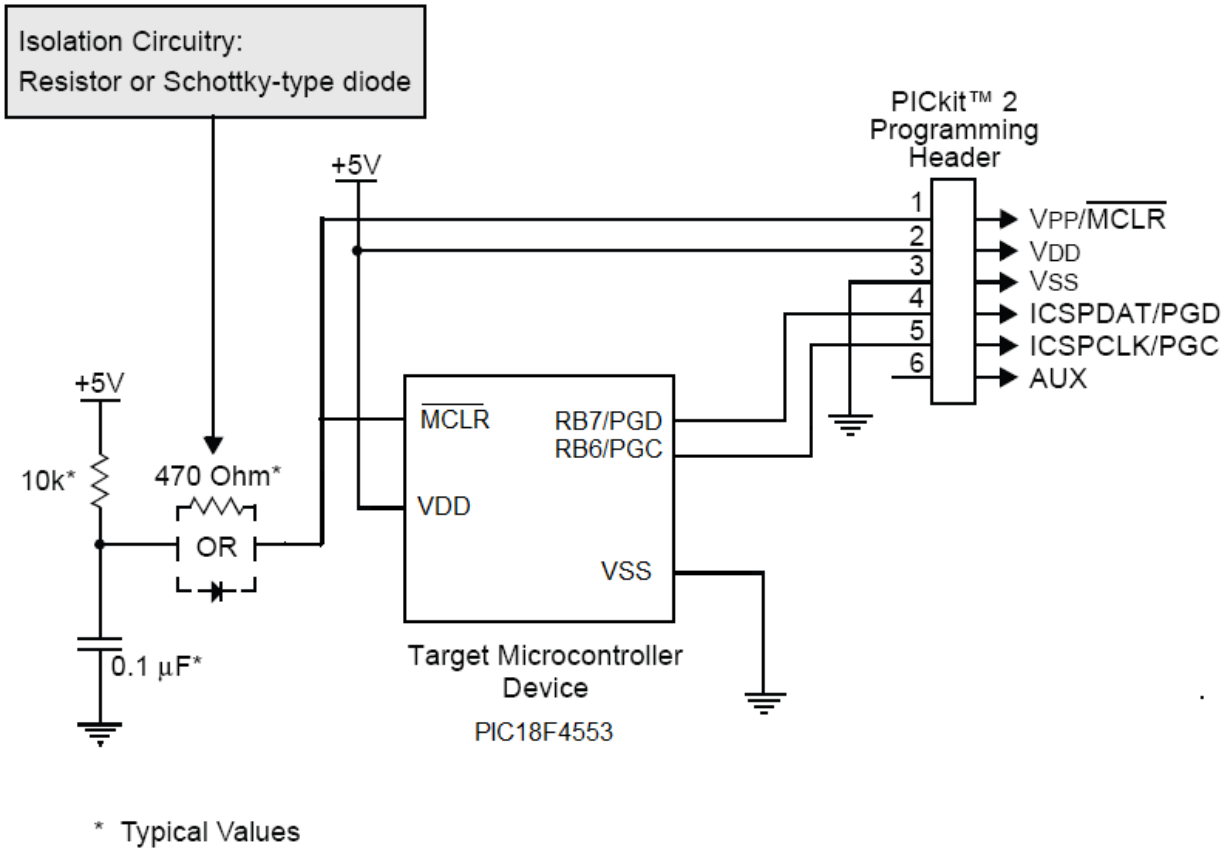

Figure 5: Connecting the PICKit2 hardware with the application circuit (modified from the User Guide of PICKit2)

where the target microcontroller is the PIC18F4553 in our case. Restart PICKit2 software, the software should automatically detect your device (18F4553). Load the file "MCHPUSB.hex" from File->Import Hex

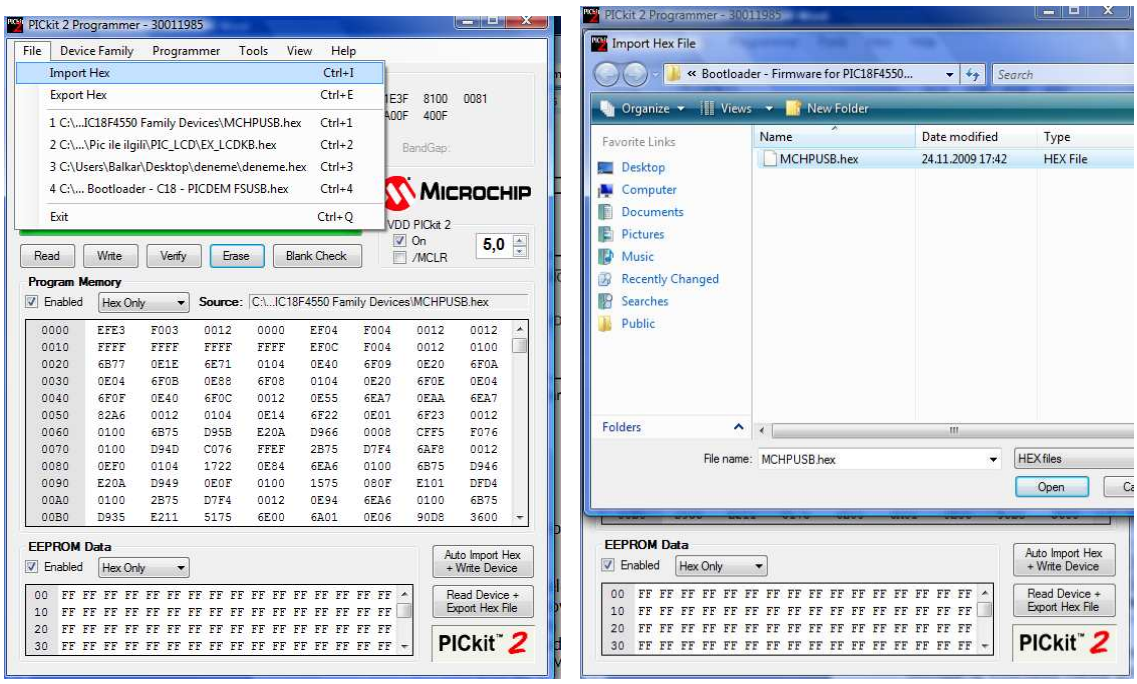

Figure 6: Loading the firmware to be programmed with PICKit2

If you can't see the message "Hex file imported successfully" then check the wiring between PICKit2 and the 18F4553 device.

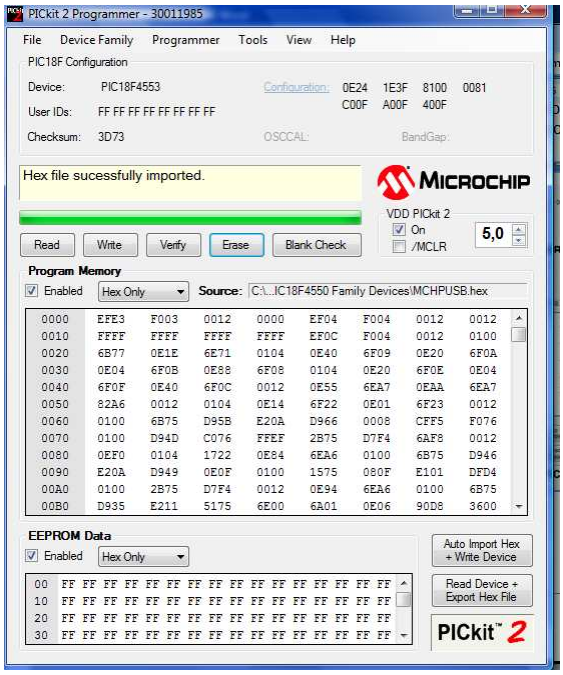

Figure 7: Successful import of the firmware

After that press "Write" and wait for the programming until you see "Programming Successful".

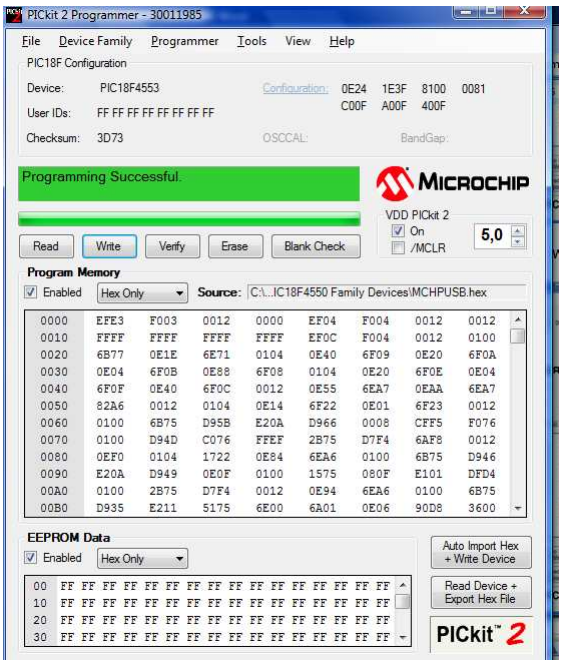

Figure 8: PICKit2 - Writing the firmware to device is complete

Now you have the bootloader firmware programmed to your 18F4553 device. Note that we have programmed a HEX file to 18F4553 device which is compiled for 18F4550 device. This does not affect the application of the firmware very much as the family of these two devices is the same (like the 18F2553 case).

You can also check other utilities of PICKit2 from "Tools". There are the Logic and Uart tools, the explanations of which you can find from "PICKit2 User Guide.pdf" (under Help).

III. Bootloading

Bootloading is another way of programming your PIC device without removing it from your application circuit. It is a kind of booting system installed on the PIC device which allows two main operations: running the actual application firmware or updating/reprogramming the application firmware. You can consider it as a simple operating system for PICs. To implement a bootlading mechanism, only a slight modification on the application circuit is required. You add a switch (and a pull up resistor R6) to one of the pins of your microcontroller as in Figure 9.

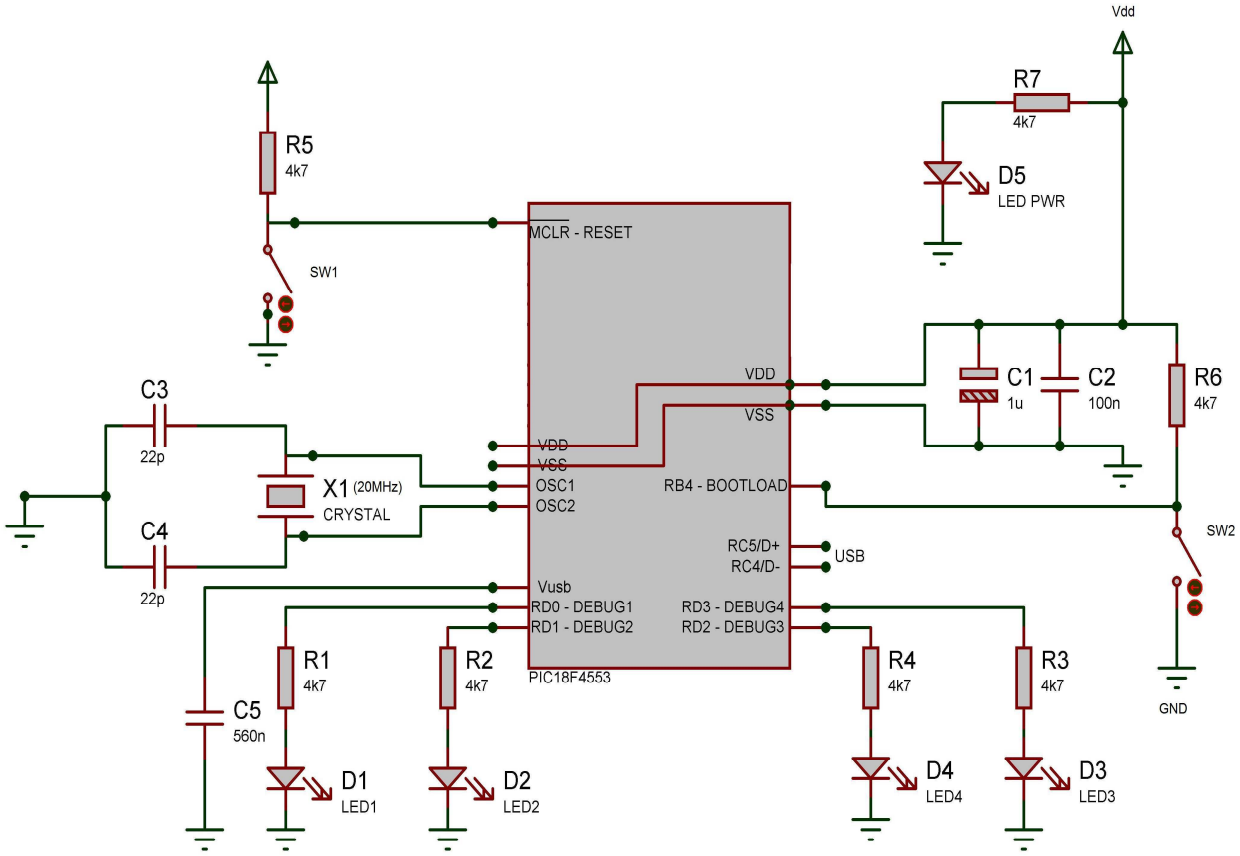

Figure 9: A uC application circuit with the bootloader switch (SW2).

Now, assume that a bootloader firmware is once programmed into your PIC18F device (via a PICKit2 or a JDM Programmer) and you set up the circuit in Figure 9. Here, the bootloader operation in is controlled by both SW1 and SW2. If you want to program your application firmware via the bootloader, you should start the device with the SW2 pressed. That is, to enter the "bootload mode", press and hold SW2 and while holding SW2, push and release SW1 (SW1 is used to restart the device firmware); you can release SW2 after that. To run the application firmware (programmed by the bootloader) or to enter the "device mode" from bootload mode, you only press SW1.

Entering the bootload mode triggers the bootloader firmware to load a new application firmware to the device. This is performed by establishing a communication between the device and a PC software. The microchip MCHP USB framework provides two bootloading mechanisms called the MCHP Vendor Bootloader and the HID bootloader. Both have similar structures in terms of circuit configuration and the PC software. Therefore, you can prefer any of them, as far as you are aware of the slight differences in the application firmware.

A. Microchip - MCHP USB Vendor Bootloader

You can see the PC software of MCHP Vendor Bootloader in Figure 10. The software is located in "C:\Microchip Solutions\USB Tools\Pdfsusb"

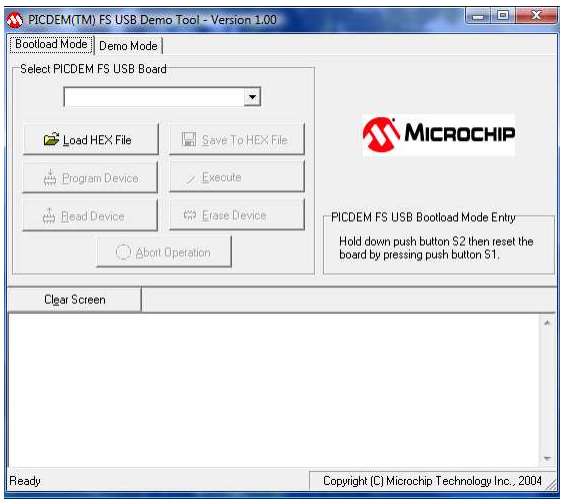

Figure 10: MCHP framework - PICDEM FS USB DEMO Bootloading Software (Vendor)

If you are going to use this PC software for bootloading, your PIC18F device should be initially programmed with the Vendor bootloader firmware. You can find the project at C:\Microchip Solutions\USB Device - Bootloaders\Vendor Class - MCHPUSB Bootloader\Bootloader - Firmware for PIC18F4550 Family Devices.

Here, you should open the project and build it with the default settings; it will finally create the file "MCHPUSB.hex" which you will program with a programmer (PICKit2 or JDM). Note that, this firmware can only be used for 18F4550 family devices; if you have a PIC18F device from another family you should use another bootloader (may be the HID one).

When you set up your circuit like in Figure 9 and enter the bootload mode, the MCHP Vendor Bootloader will identify your device and the options in the user interface will be enabled as in Figure 11.

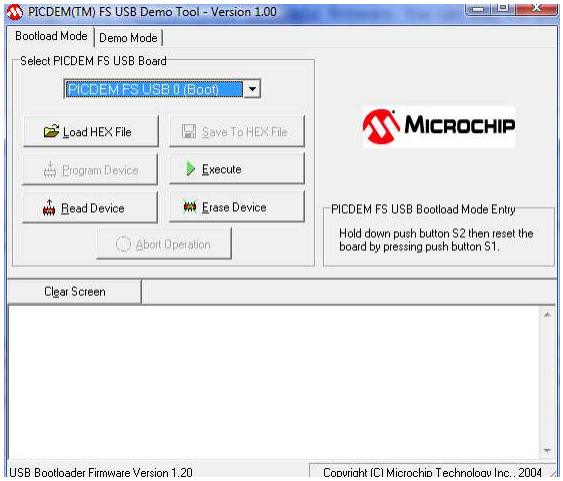

Figure 11: Vendor MCHP bootloader identifying the PIC18F4550 family device programmed with the Vendor bootlader firmware.

With the vendor USB bootloader, you can load your application firmware **if they are also configured to run with the vendor bootloader. If the configurations do not match, you should modify your application code accordingly**. For instance, the

CDC example discussed in tutorial 1 was default configured to run with the **HID bootloader** (actually this is valid for all the examples) and we modified the codes to run it with the **MCHP vendor bootloader**. To remember what is done to make the application firmware work with the MCHP bootloader, we removed the linker file "rm18F4550 – HID Bootload.lkr" and added "rm18F4550 – MCHPUSB Bootload.lkr" instead (further modifications should be done for a different device, see tutorial 1). Moreover, we commented the line #define PROGRAMMABLE_WITH_USB_HID_BOOTLOADER in "HardwareProfile - PICDEM FSUSB.h" (you can access it from "HardwareProfile.h" by right click on the line and Open File etc.) and uncommented the line #define PROGRAMMABLE_WITH_USB_MCHPUSB_BOOTLOADER. Further modifications might be required in "HardwareProfile.h" to include your device if it is different than 18F4550 (see tutorial 1).

The advantage of using MCHP USB bootloader is that the bootloader firmware covers less memory space in the device than the HID counterpart. Therefore, if you are using 18F4550 family device it will be wiser to use the MCHPUSB vendor bootloader.

As a last comment, you can change the pin for your booting switch (SW2) from the main() function in main.c in MCHPUSB project. Furthermore, if your C18 compiler (standard version, not the student version) evaluation period expires, you will get error when you want to compile the bootloader project. Therefore, it is highly recommended that you compile that project for several times with different sw2 pin locations and save the HEX files before the expiration date (or you can use another computer as well☺)

B. HID Bootloader

HID bootloader differs from the vendor counterpart in terms of the number of supported devices. With the HID bootloader, you can program not only 18F4550 family devices but also other USB PIC family devices (see the list in the project folder) as well (which is an advantage of the HID bootloader over the vendor one. You can see the PC software of HID Bootloader in Figure 12. The software and precomiled hex files are located in "C:\Microchip Solutions\USB Device - Bootloaders\HID - Bootloader"

Figure 12: User Interface of the HID Bootloader provided in MCHPUSB framework

Once you burn the HID bootloader firmware into the device (only once) with the PICKit2 or JDM programmer, the software will recognize the bootloader and will respond by enabling its visual components (Erase, Read, Program/Verify the device etc.). The bootloading procedure (sw2 & sw1 combination) discussed above is also valid for the HID bootloader.

Final Remarks

When programming the application firmware with HID bootlader, you should take the linker script and Hardware Profile configurations in the project. **Otherwise, the memory for the bootloader and the application firmware will overlap!** (This statement is valid also for the vendor bootloader) In that case, you will need to reprogram the bootloader firmware into the microcontroller.

Furthermore, the same problem in compiling the bootloader firmware (in the vendor case) would appear if the due date of the C18 compiler expires.