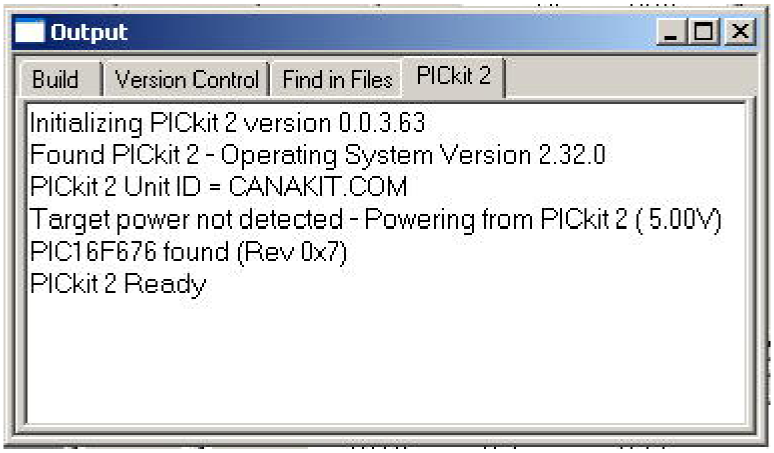
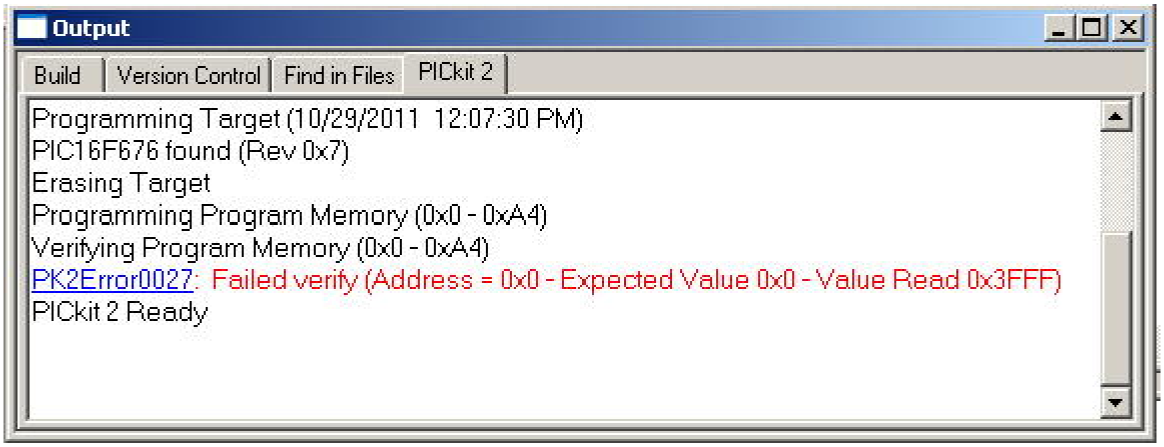
CanaKit Programmer Error 27 Workaround
Mark Spencer, WA8SME
mspencer@arrl.org
860-381-5335

Introduction. Many readers of the *PIC Programming for Beginners* book that are using the CanaKit programmer supplied in the kit of parts purchased from the ARRL are having difficulty with an ERROR 27 when they attempt to program the PICs. Diligently following the directions in the book, they will connect the programmer and the programmer will be recognized.



```
Initializing PICkit 2 version 0.0.3.63
Found PICkit 2 - Operating System Version 2.32.0
PICkit 2 Unit ID = CANAKIT.COM
Target power not detected - Powering from PICkit 2 ( 5.00V)
PIC16F676 found (Rev 0x7)
PICkit 2 Ready
```

When the program is complied, built, and sent to the programmer, the return is an ERROR 27 message and the programming stops.



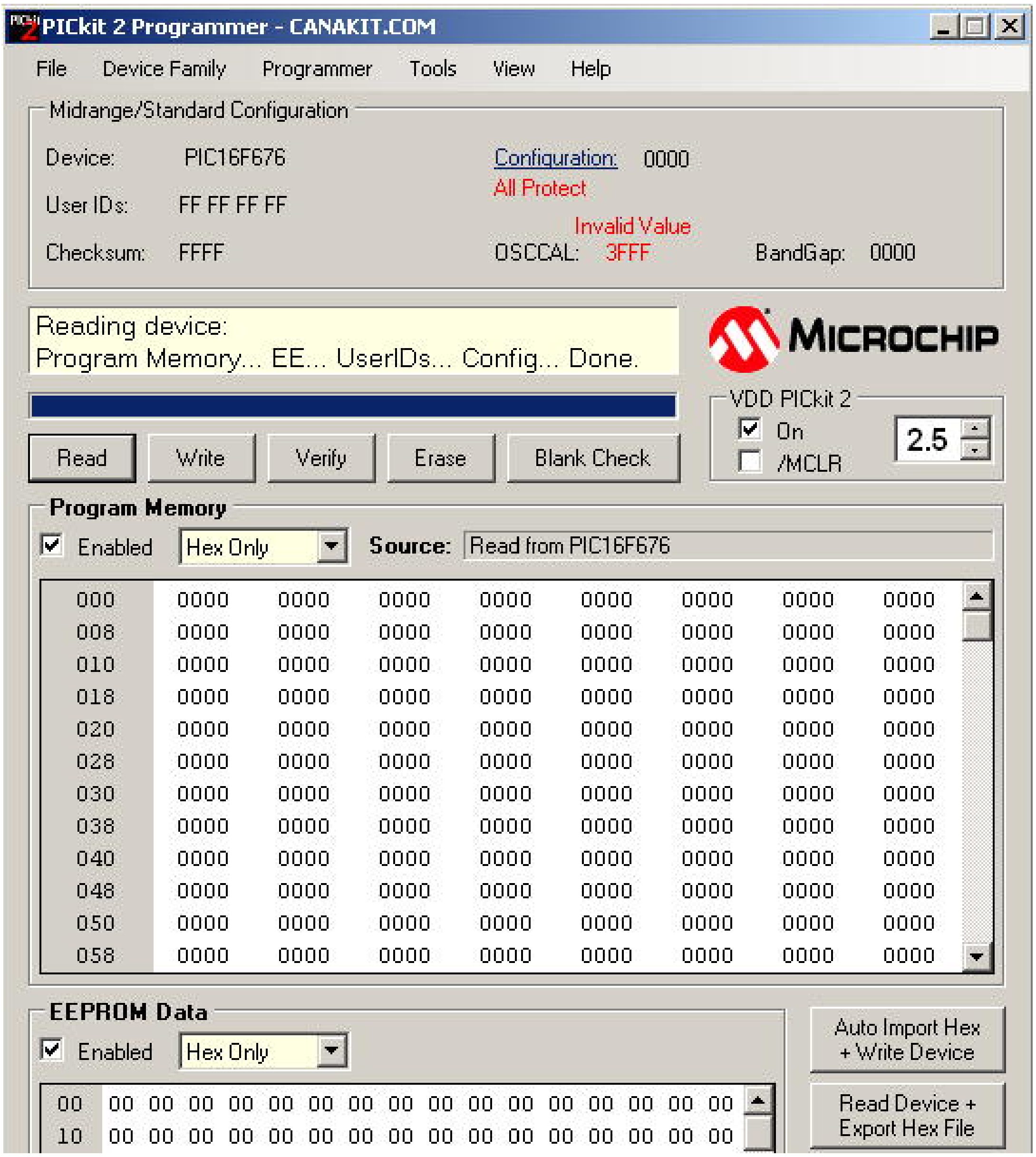
```
Programming Target (10/29/2011  12:07:30 PM)
PIC16F676 found (Rev 0x7)
Erasing Target
Programming Program Memory (0x0 - 0xA4)
Verifying Program Memory (0x0 - 0xA4)
PK2Error0027:  Failed verify (Address = 0x0 - Expected Value 0x0 - Value Read 0x3FFF)
PICkit 2 Ready
```

The reason why this is happening is unknown. There appears to be a conflict between the CanaKit PICKit2 clone programmer and MPLab IDE. When you hit the programming icon in MPLab, a number of steps are taken by the software to accomplish the actual programming. Early in the programming sequence, the contents of a critical register called the OSCCAL value is retrieved from the PIC and is supposed to be stored temporarily while the rest of the program is sent to the PIC. After the main program is delivered to the PIC, the OSCCAL value is re-installed back onto the PIC. For some yet undetermined reason, this is not happening with the CanaKit programmer. The OSCCAL value is lost and the programming is stopped with the ERROR 27 message returned. The OSCCAL value is actually gone and needs to be recovered before the PIC device can be used again as mentioned in the book.

The CanaKit works fine with another programming software package called PICkit 2 (not to be confused with the programmer called PICKit 2). The following steps outline a workaround to allow you to continue your studies of PIC programming while using the CanaKit programmer with PICkit 2 software.

Using PICKit 2 Software to Program the PIC. The work around takes advantage of the fact that the CanaKit appears to work fine with the PICKit 2 software package. This PICKit 2 software is a pretty functional programming package. While it does not facilitate the writing of programs as you are learning to do in the book (MPLab is still required for program development), the PICKit 2 software utilities allow you to recover a lost or corrupted OSCCAL value and also to program the PIC from a HEX file.

Step 1. Install the PICkit 2 software package. Go to the Microchip web page at this URL: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023805. On the right side of the page you will see this link: **PICkit 2 v2.61.** Download and install the PICkit 2 software accepting the defaults. When you run the software with your CanaKit programmer attached, you will see the software displayed like this:


PICkit 2 Programmer - CANAKIT.COM
File
Device Family
Programmer
Tools
View
Help
MICROCHIP
Read
Write
Verify
Erase
Blank Check
Auto Import Hex + Write Device
Read Device + Export Hex File

Midrange/Standard Configuration

Device: PIC16F676

User IDs: FF FF FF FF

Checksum: FFFF

Configuration: 0000

All Protect

Invalid Value

OSCCAL: 3FFF

BandGap: 0000

```
Reading device:
Program Memory... EE... UserIDs... Config... Done.
```

VDD PICkit 2

☑ On

☐ /MCLR

2.5

Program Memory

☑ Enabled Hex Only **Source:** Read from PIC16F676

```
000   0000  0000  0000  0000  0000  0000  0000  0000
008   0000  0000  0000  0000  0000  0000  0000  0000
010   0000  0000  0000  0000  0000  0000  0000  0000
018   0000  0000  0000  0000  0000  0000  0000  0000
020   0000  0000  0000  0000  0000  0000  0000  0000
028   0000  0000  0000  0000  0000  0000  0000  0000
030   0000  0000  0000  0000  0000  0000  0000  0000
038   0000  0000  0000  0000  0000  0000  0000  0000
040   0000  0000  0000  0000  0000  0000  0000  0000
048   0000  0000  0000  0000  0000  0000  0000  0000
050   0000  0000  0000  0000  0000  0000  0000  0000
058   0000  0000  0000  0000  0000  0000  0000  0000
```

EEPROM Data

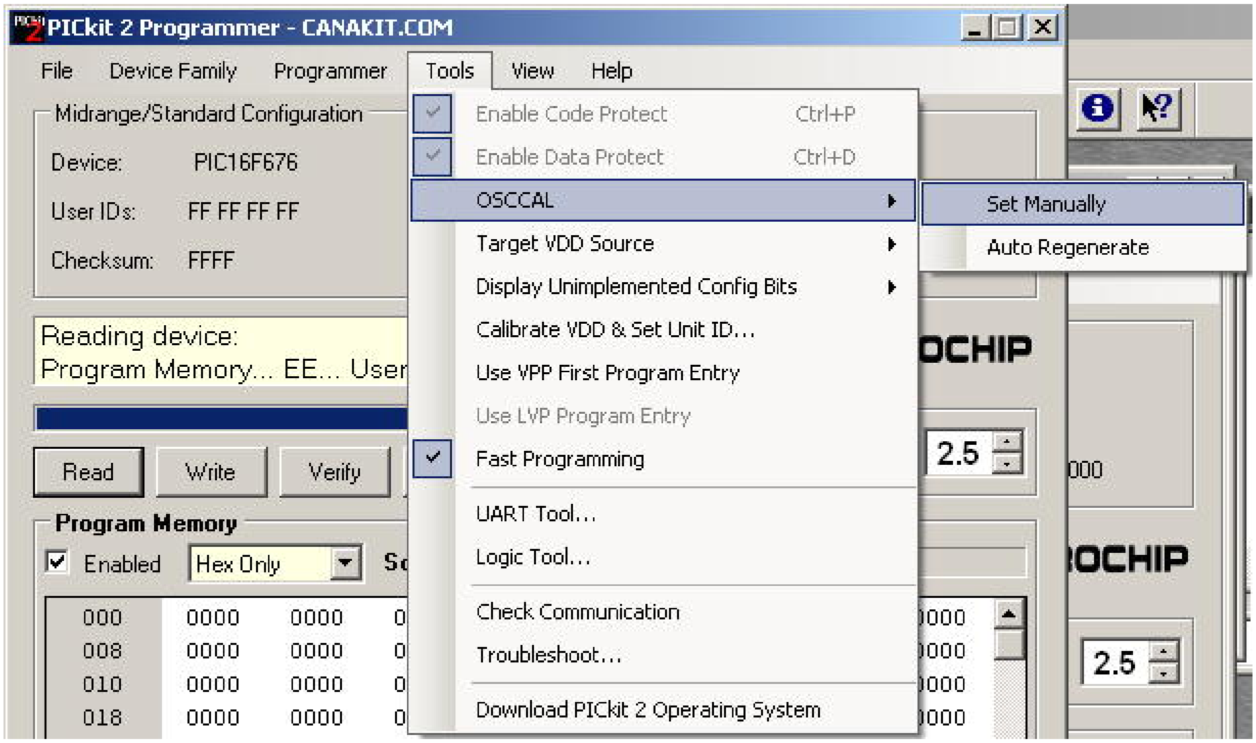
☑ Enabled Hex Only

```
00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Notice that the OSCCAL value is highlighted in red and indicates that there is an invalid value of 3FFF (if your PIC has been previously corrupted, if not, the value here will probably be valid with the value of 34XX…the XX are hex numbers that reflect the PICs OSCCAL value.

Step 2. Recovering the OSCCAL value. This step can take two directions. One if you were diligent in recording the OSCCAL value as suggested in the book before you do any programming of the device. The other direction is if you forgot to do that, or couldn't for some reason, and allows you to recover the unknown value to make the PIC usable again.

Direction one. Click on Tools/OSCCAL/Manual Set.



Midrange/Standard Configuration

Device: PIC16F676

User IDs: FF FF FF FF

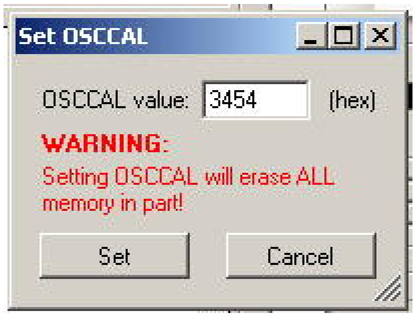
Checksum: FFFF

Reading device:
Program Memory... EE... User

☑ Enabled

```
000    0000    0000    0
008    0000    0000    0
010    0000    0000    0
018    0000    0000    0
```

This popup window will be displayed allowing you to manually enter the OSCCAL value, in this case 3454.

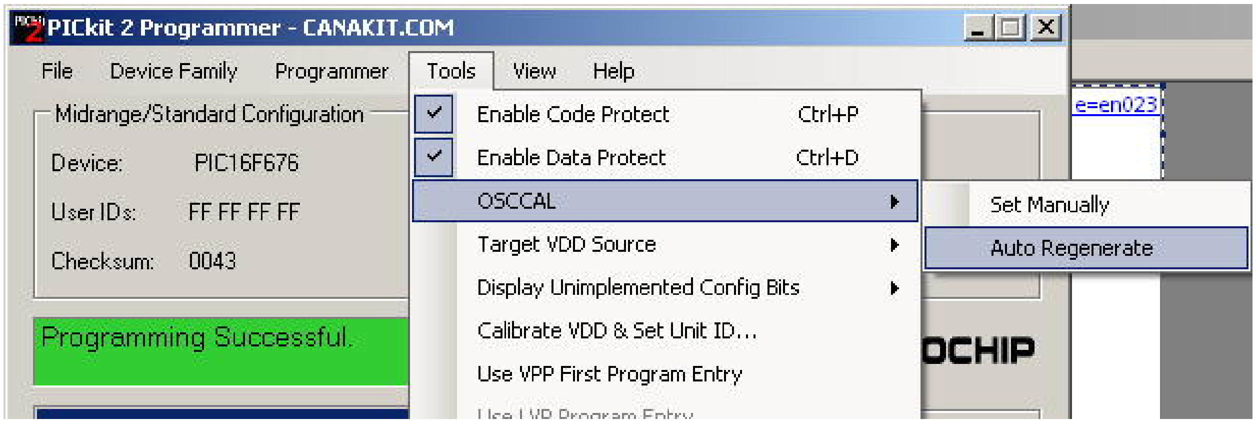


OSCCAL value: 3454 (hex)

WARNING:
Setting OSCCAL will erase ALL memory in part!

All the OSCCAL values will begin with 34, flowed by two alpha-numeric characters that are the actual HEX values of the OSCCAL value. Don't worry about the WARNING, that ship has already sailed. Click on Set. Once the software has reinstalled the OSCCAL value, you will be returned to the main software window and the correct OSCCAL value will be displayed and you're ready to move on.

Direction two. If you have no idea what the OSCCAL value is for a particular device, you can allow the software to experimentally determine the value and install it. This system apparently is not perfect, because fequently you will find that the software generated OSCCAL value and the factory determined OSCCAL value (that you read off of a fresh device) are not always the same. There must be some range of acceptable values because I have used the software generated OSCCAL values for many applications with no problem. To have the software determine and load the OSCCAL value, click on Tools/OSCCAL/Auto Regenerate.



PICkit 2 Programmer - CANAKIT.COM

Midrange/Standard Configuration

Device: PIC16F676

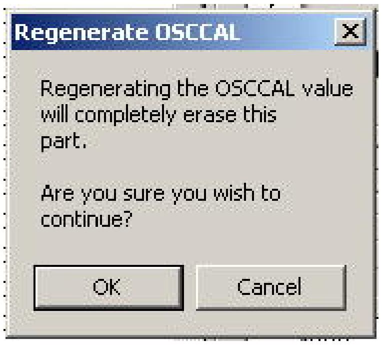
User IDs: FF FF FF FF

Checksum: 0043

Programming Successful.

File
Device Family
Programmer
Tools
View
Help
Enable Code Protect Ctrl+P
Enable Data Protect Ctrl+D
OSCCAL
Target VDD Source
Display Unimplemented Config Bits
Calibrate VDD & Set Unit ID...
Use VPP First Program Entry
Use LVP Program Entry
Set Manually
Auto Regenerate
e=en023
OCHIP

Then follow the prompts to allow the software to regenerate and install the OSCCAL value. Click on OK.

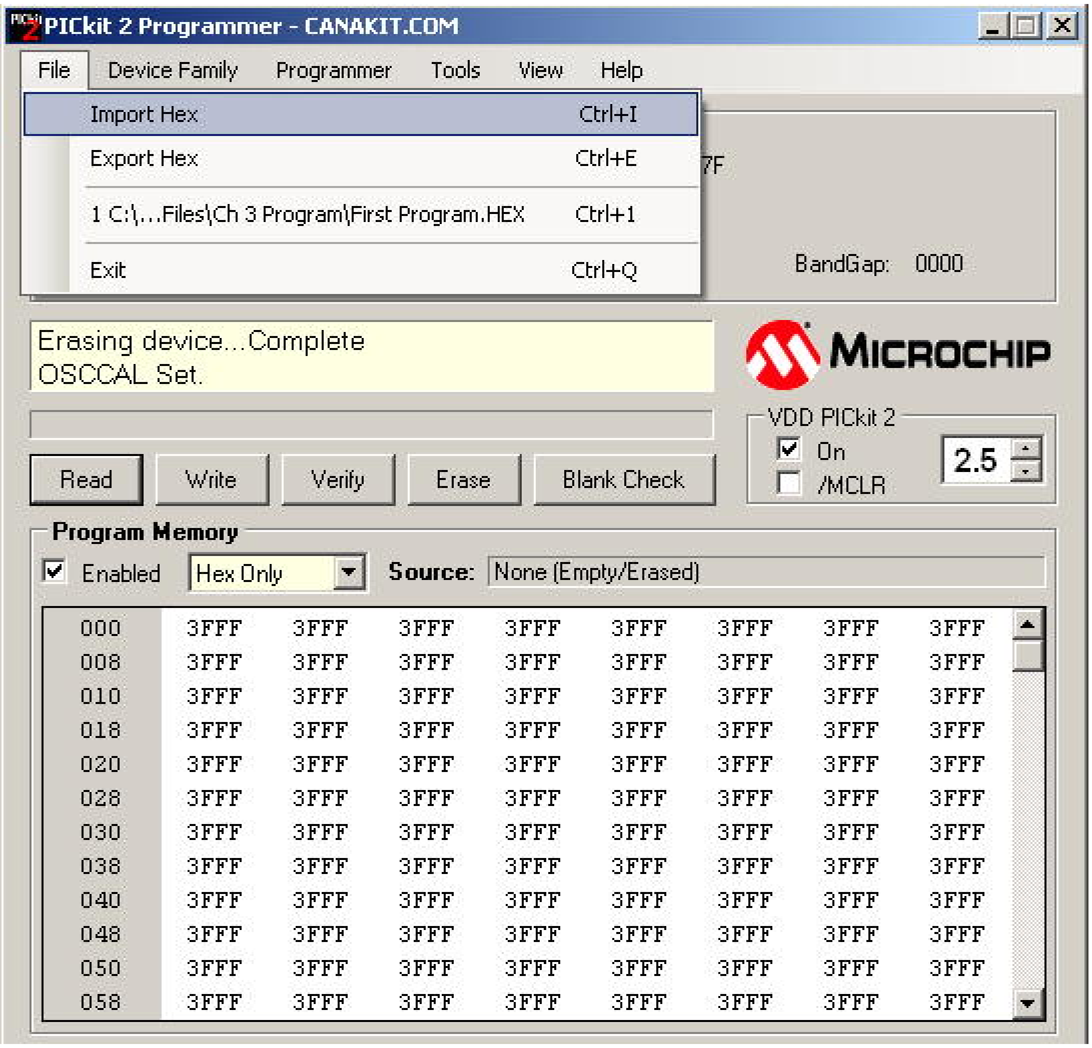


Regenerate OSCCAL

Regenerating the OSCCAL value will completely erase this part.

Are you sure you wish to continue?

Step 3. Using PICkit 2 software to download a HEX file into the PIC. When you use MPLab IDE to author and build a program for the PIC, the software actually creates a number of files without your direct intervention. The programs that you write in the MPLab editor window are not the actual files that are downloaded into the PIC, but the "human" language interpretation of the program. The MPLab compiler actually converts your writing into machine language that the PIC understands, and the program in machine language is actually what is installed onto the PIC. That machine language version of your program is in the HEX file (HEX is short for hexadecimal…base 16 numbering system) and has the extension .HEX. If you look in the directory where your program files are stored, you will find the associated HEX file for the program you are working on. It is this HEX file that you will install using PICkit 2 software. After you have created your programs as outlined in the book in MPLab, instead of using MPLab to install the program on the PIC, launch the PICkit 2 software.



BandGap: 0000

```
Erasing device...Complete
OSCCAL Set.
```

VDD PICkit 2

☑ On

☐ /MCLR

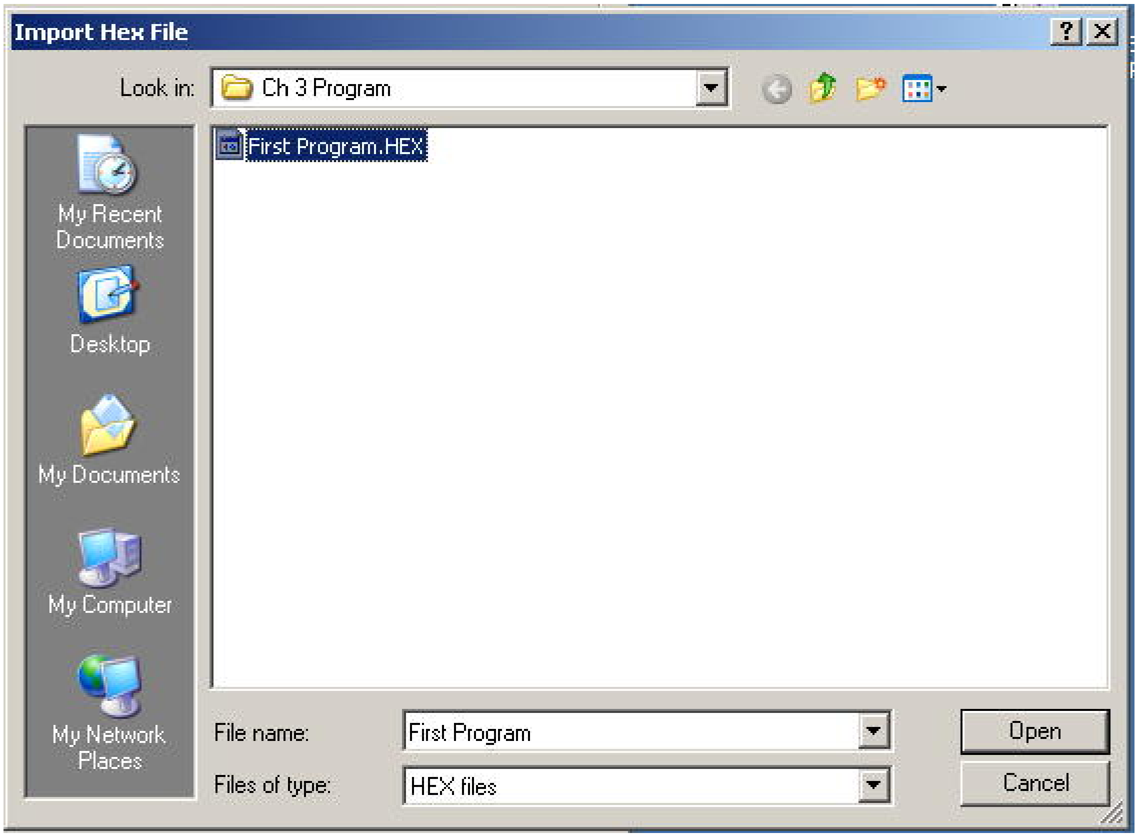
2.5

Program Memory

☑ Enabled Hex Only **Source:** None (Empty/Erased)

```
000   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
008   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
010   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
018   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
020   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
028   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
030   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
038   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
040   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
048   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
050   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
058   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
```

Click on File/Import Hex and navigate to where your programs are being filed on your hard disk.


Import Hex File

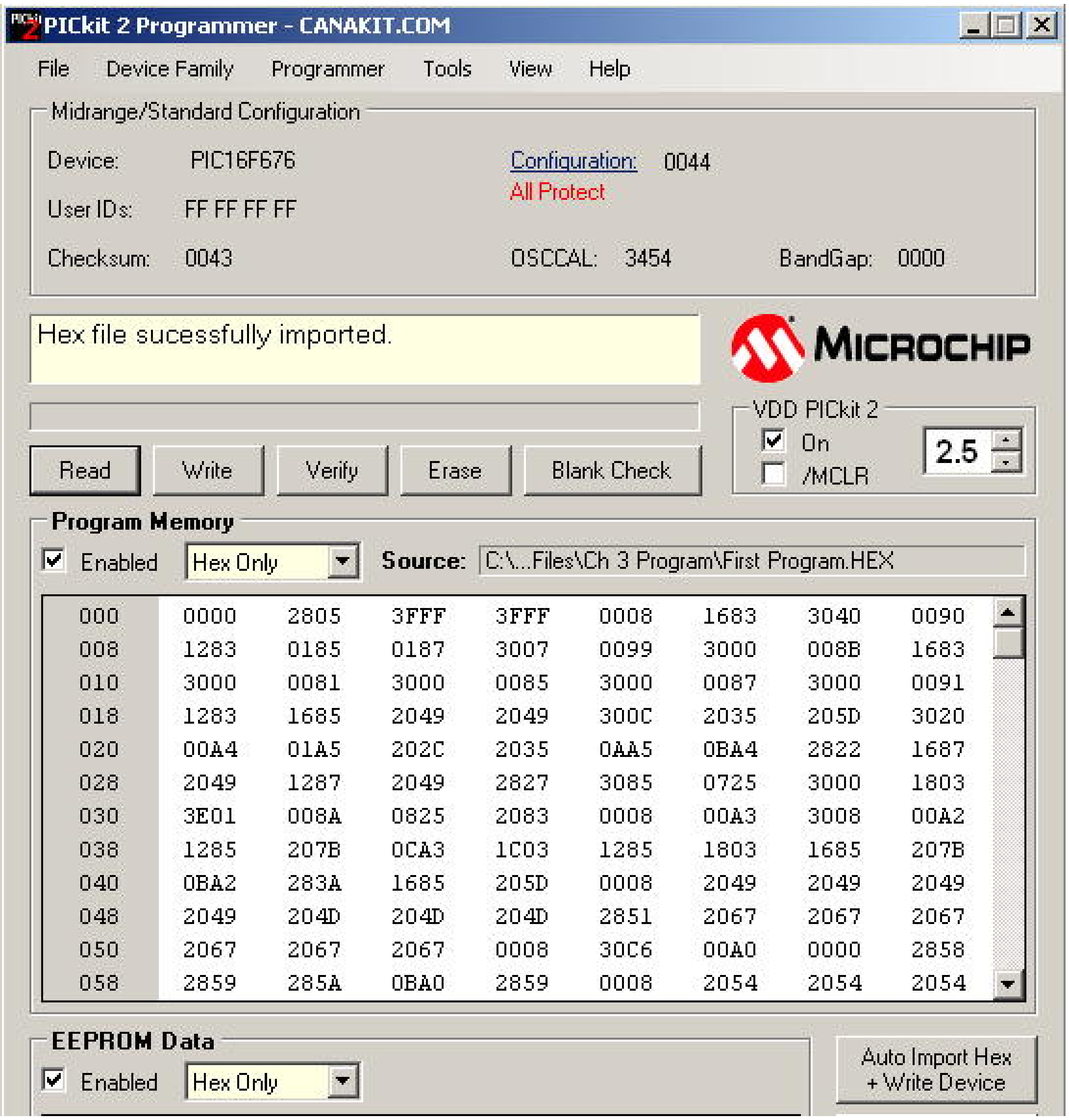
Look in: Ch 3 Program

First Program.HEX

File name: First Program

Files of type: HEX files

If you use the default folders for the program files in the book, you will find the HEX file for the program as shown here for the first program in chapter 3. Click on Open and the HEX version of your program will be imported into PICkt 2 and is ready for installation into the PIC memory.



Midrange/Standard Configuration

Device: PIC16F676

Configuration: 0044

All Protect

User IDs: FF FF FF FF

Checksum: 0043

OSCCAL: 3454

BandGap: 0000

Hex file sucessfully imported.

VDD PICkit 2

☑ On

☐ /MCLR

2.5

Program Memory

☑ Enabled Hex Only

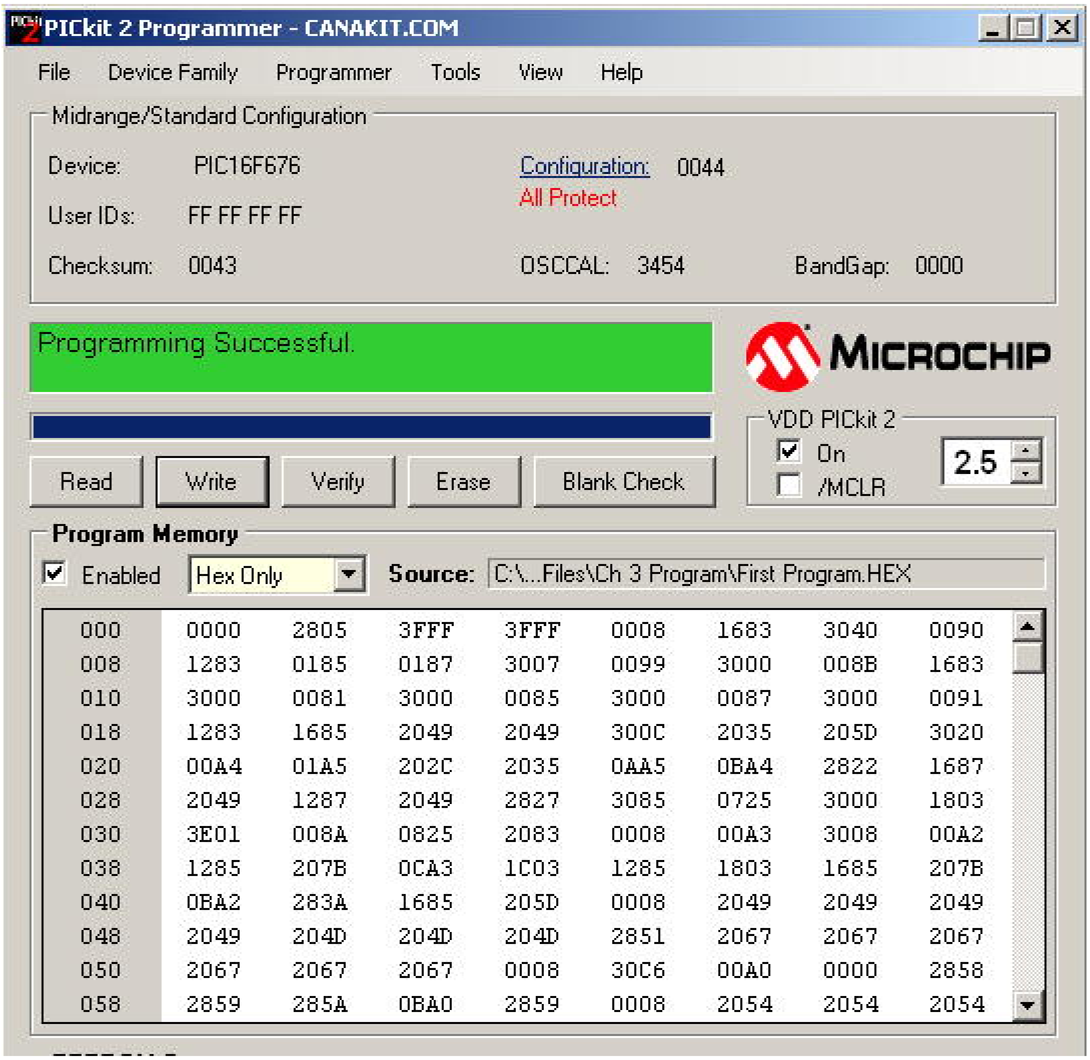
Source: C:\...Files\Ch 3 Program\First Program.HEX

```
000   0000  2805  3FFF  3FFF  0008  1683  3040  0090
008   1283  0185  0187  3007  0099  3000  008B  1683
010   3000  0081  3000  0085  3000  0087  3000  0091
018   1283  1685  2049  2049  300C  2035  205D  3020
020   00A4  01A5  202C  2035  0AA5  0BA4  2822  1687
028   2049  1287  2049  2827  3085  0725  3000  1803
030   3E01  008A  0825  2083  0008  00A3  3008  00A2
038   1285  207B  0CA3  1C03  1285  1803  1685  207B
040   0BA2  283A  1685  205D  0008  2049  2049  2049
048   2049  204D  204D  204D  2851  2067  2067  2067
050   2067  2067  2067  0008  30C6  00A0  0000  2858
058   2859  285A  0BA0  2859  0008  2054  2054  2054
```

EEPROM Data

☑ Enabled Hex Only

You will notice that the HEX values in the Program Memory area of PICkit 2 have changed from all 0000's to HEX numbers that reflect the program values. Finally, click on Write and the PICkit 2 software will do its thing and install the program onto the PIC.



Midrange/Standard Configuration

Device: PIC16F676

User IDs: FF FF FF FF

Checksum: 0043

Configuration: 0044

All Protect

OSCCAL: 3454

BandGap: 0000

Programming Successful.

VDD PICkit 2

☑ On

☐ /MCLR

2.5

Program Memory

☑ Enabled Hex Only

Source: C:\...Files\Ch 3 Program\First Program.HEX

```
000   0000  2805  3FFF  3FFF  0008  1683  3040  0090
008   1283  0185  0187  3007  0099  3000  008B  1683
010   3000  0081  3000  0085  3000  0087  3000  0091
018   1283  1685  2049  2049  300C  2035  205D  3020
020   00A4  01A5  202C  2035  0AA5  0BA4  2822  1687
028   2049  1287  2049  2827  3085  0725  3000  1803
030   3E01  008A  0825  2083  0008  00A3  3008  00A2
038   1285  207B  0CA3  1C03  1285  1803  1685  207B
040   0BA2  283A  1685  205D  0008  2049  2049  2049
048   2049  204D  204D  204D  2851  2067  2067  2067
050   2067  2067  2067  0008  30C6  00A0  0000  2858
058   2859  285A  0BA0  2859  0008  2054  2054  2054
```

Now that the program is in the PIC, it will start running on the PIC. Here is where you remove the PIC from the CanaKit programmer and install it in your circuit for testing.

Conclusion.

If you are getting ERROR 27 messages when you are using the CanaKit programmer with MPLab IDE, there is an unidentified conflict that is being worked. There is a workaround that will allow you to continue with your studies of PIC programming while using the CanaKit.

1. Install the PICkit 2 software package from the Microchip web site. The CanaKit programmer appears to work will with the PICkit 2 software package. PICkit 2 software allows you to install program HEX files onto the PIC.
2. Using PICkit 2, recover the OSCCAL value that was lost or corrupted during the attempt to program using the CanaKit with MPLab. The OSCCAL value can be restored using the known factory determined OSCCAL value or it can be auto regenerated using the PICkit 2 utility. (The

loss of the OSCCAL value appears to only be happening when using the CanaKit programmer with MPLab, it does not happen when using the PICkt 2 software.)

3. After you have developed the program using MPLab, build the program, but stop short of using MPLab to install the program onto the PIC.
4. Launch the PICkit 2 software. Click on File/Import and navigate to the folder where your program files are stored. Click on the appropriate file with the .HEX extension to import the hex version of your program into PICkit 2.
5. Click on Write to install the program. The OSCCAL value should not be lost using this procedure so set 2 above will not be needed further.
6. Remove the PIC form the CanaKit programmer and install it into your circuit for testing.