

# PIC Programming using ICSP

This page shows you how to use ICSP for PIC programming giving some ICSP connection diagrams.

Note Part of this page shows you how to use ICPROG which is a parallel port/serial port based software that requires use of an older computer having either a serial port or a parallel port - there are lots around. These days those interfaces have been replaced with USB ports so you can not use ICPROG if you only have USB interface ports.

For USB ports, buy a USB programmer such as pickit2 or pickit3 (programmable using the Free MPLAB X software from Microchip - see screen shots on programming a pre-bult hex file below).

Jump to [ICSP Overview](#).

Jump to [ICSP Connections](#) (for typical PICS).

Jump to [MPLAB X](#) Instructions for programming a hex file.

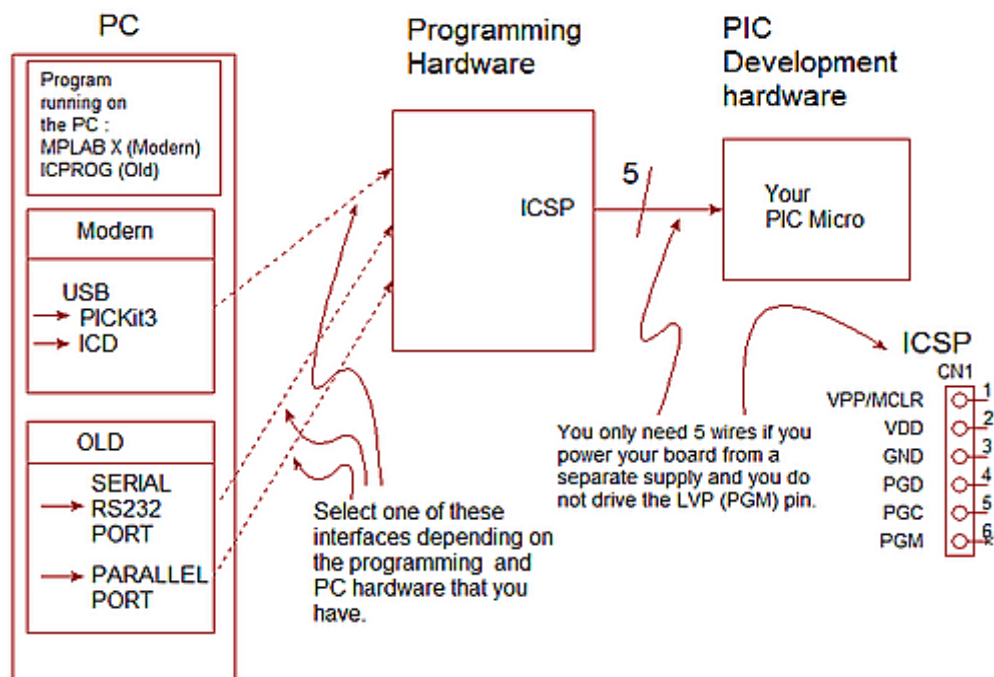
Jump to [ICPROG](#) Instructions for programming a hex file.

## ICSP

[ICSP](#) (In Circuit Serial Programming) is the serial interface used by the PIC micro to download a program into the PIC micro's program memory.

*Note: You can also program the internal EEPROM as well.*

You can learn more about ICSP [here](#).



## MPLAB X : Programming a Hex file

MPLAB X is a big program and it is sometimes difficult to see exactly how to do the simplest task. One of those tasks is programming a simple hex file without doing anything else!

The instructions below will guide you to effortlessly complete this fundamental task:

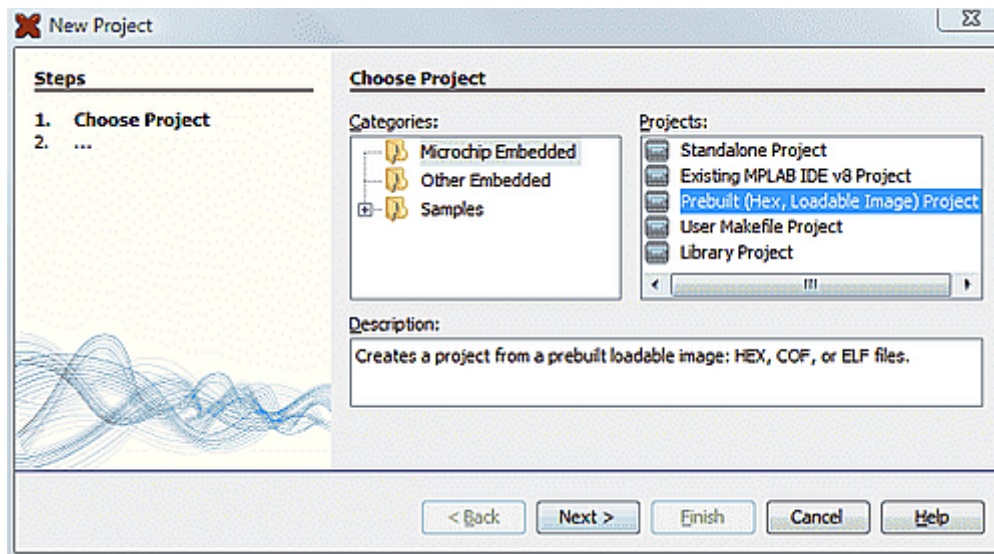
Download MPLAB X (Currently Version 2.2)

## Creating a Prebuilt-Project in MPLAB X

Start MPLAB X...wait...wait...wait...

...Click Menu: File-->New Project...

...and select "Prebuilt (Hex, Loadable Image) Project",



Click Next.

Choose your programmer, in this case PICKit3.

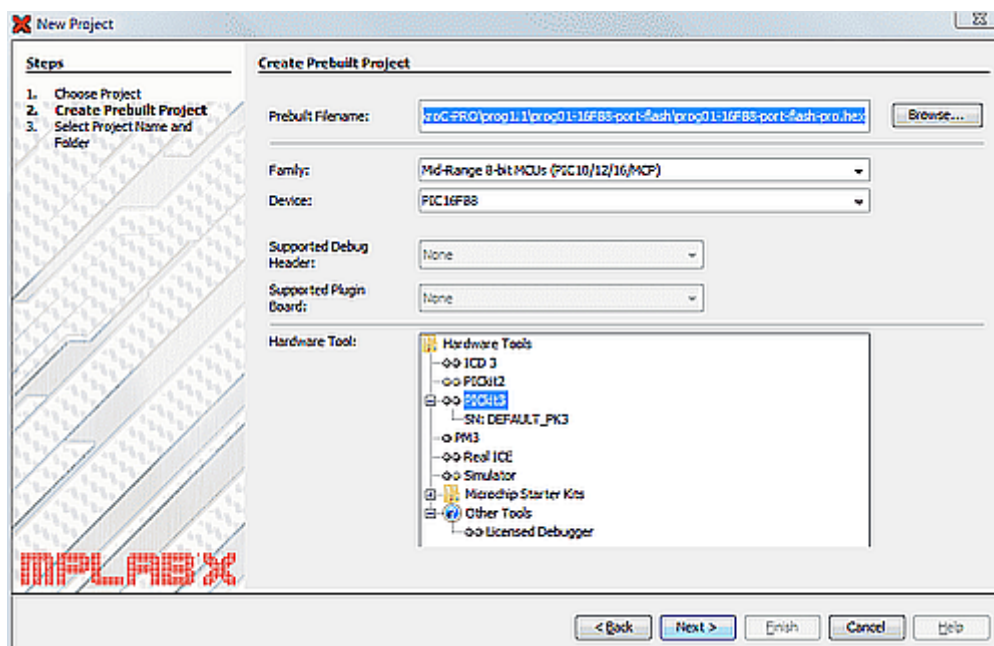
Click the Browse button and navigate to your hex file e.g.:

C:\PIC-C-Course-MikroC-PRO\prog1.1\prog01-16F88-port-flash\prog01-16F88-port-flash-pro.hex

In this case program 1 in the PIC C Course.

Choose the Family: Mid-Range 8bit MCUs,

Choose the device PIC16F88



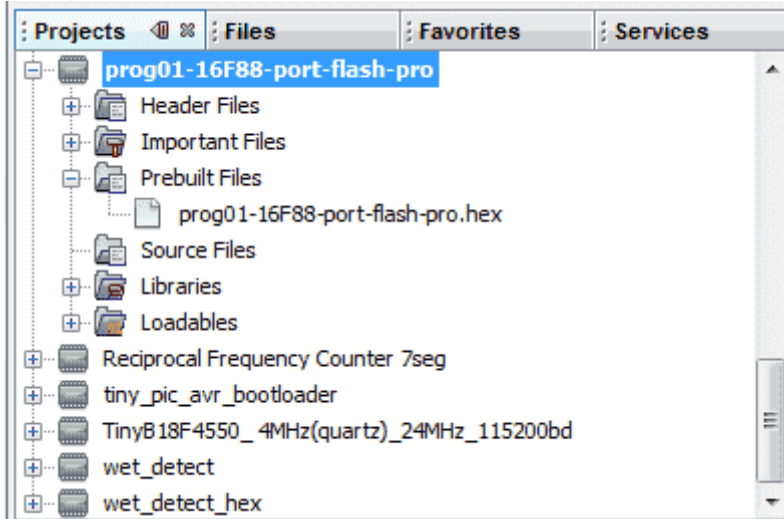
Click Next,

In the new pop-up window (not shown here), leave everything else at default values,  
Click Finish, and a project will be created at directory:

prog01-16F88-port-flash-pro.X

Click Finish and the main project pane will be updated with the new project named:

prog01-16F88-port-flash-pro



## Programming the Hex file

The above screen-shot shows the hex file that will be used.

To program in the new hex file you can hit the "green down arrow-to-chip" icon.



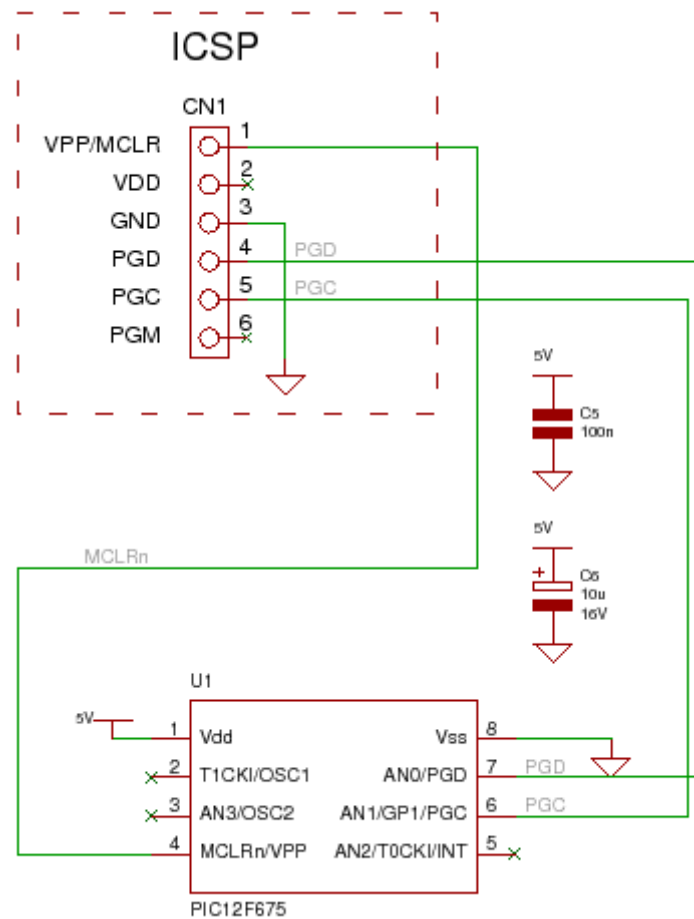
You can also program the file using the menu (in this case no build is performed since it is pre-built):

Run-->Run Main Project

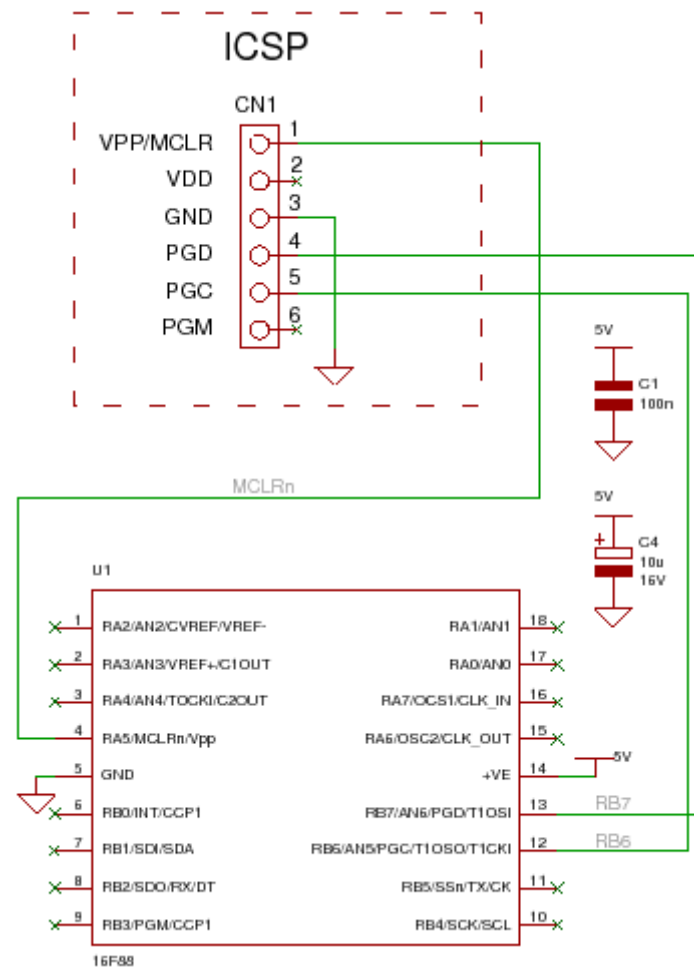
## PIC ICSP connections

This section shows connection pins for PIC programming.

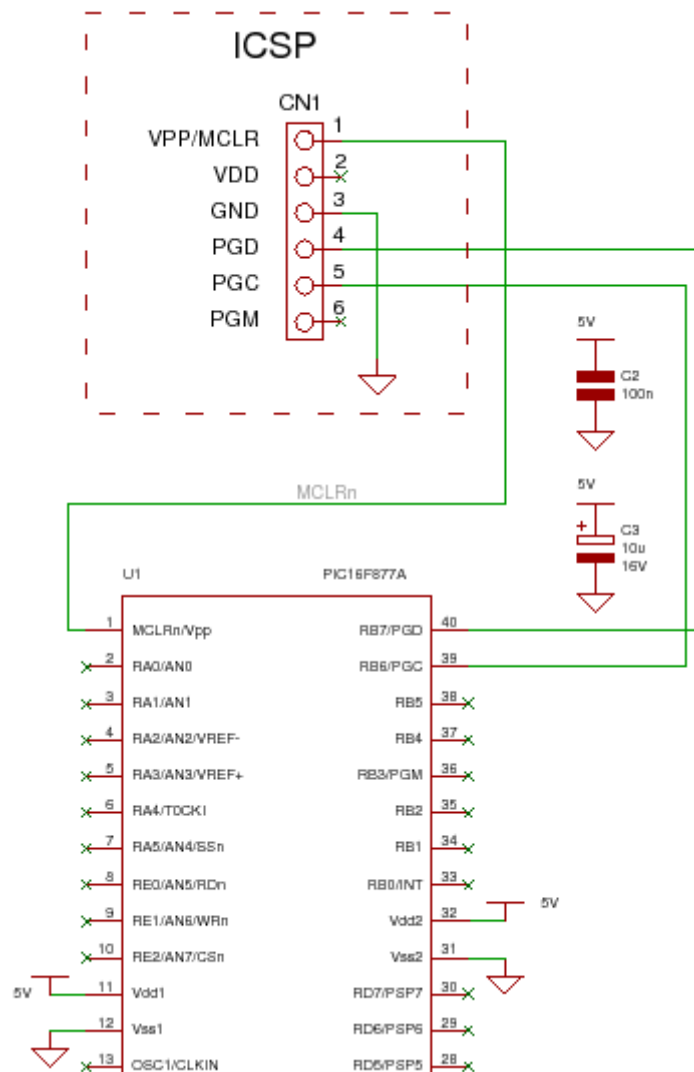
### 12F675 : ICSP PIC Programming connections:



### 16F88 : ICSP PIC Programming connections:



### 16F877A : ICSP PIC Programming connections:



## Legacy information for ICPROG

to use ICPROG requires a parallel or serial port.

The rest of this page is left for reference for if you are using ICPROG and it is also useful to see the ICSP connections, that remain the same whichever programming method you use.

**Note:** The following discussion on ICPROG is left here for reference. Most computers only have USB port so ICPROG is not useful to you. Therefore you should use the standard programming interface either Pickit2 or Pickit3 and use MPLABX as the programming IDE.

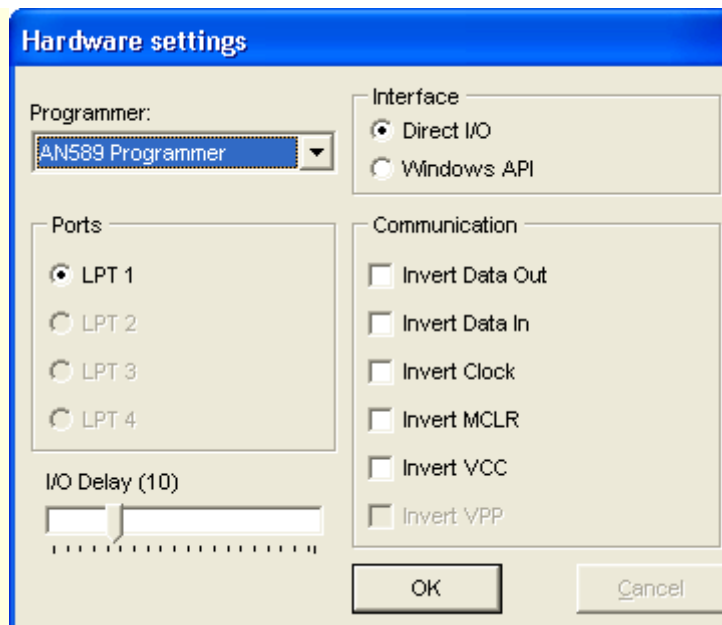
### ICPROG for PICMicro

ICPROG is a free windows program that you can use for PIC Programming. It interfaces using either the serial or parallel port on a PC, via programming hardware, to the ICSP pins on the PIC micro.

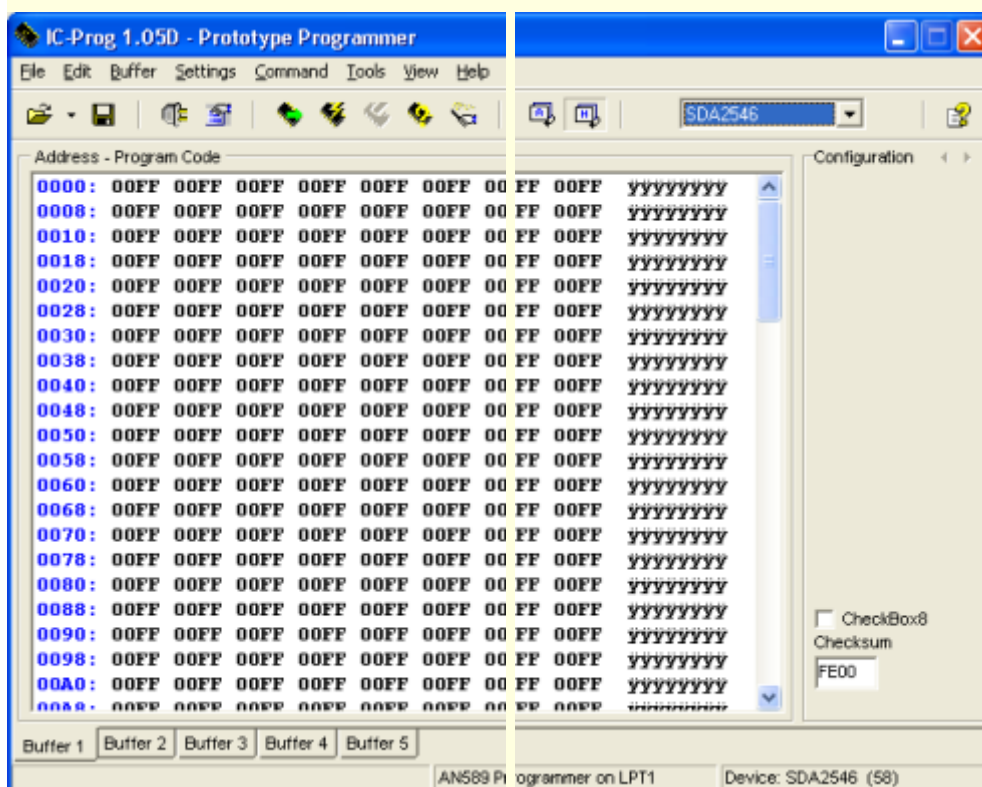
You can download ICPROG [here \[opens a new window\]](#).

ICPROG uses the hex file generated either from an assembler such as MPASM or a compiler such as MikroC.

When you first start ICPROG this is the screen you see for selecting the hardware for the PIC programmer; I have selected [AN589 programmer](#) - a parallel port programmer.



After this the main program screen is displayed. Don't be put off by all the address program code area. The important controls are the Open button and the buttons with ICs underneath.



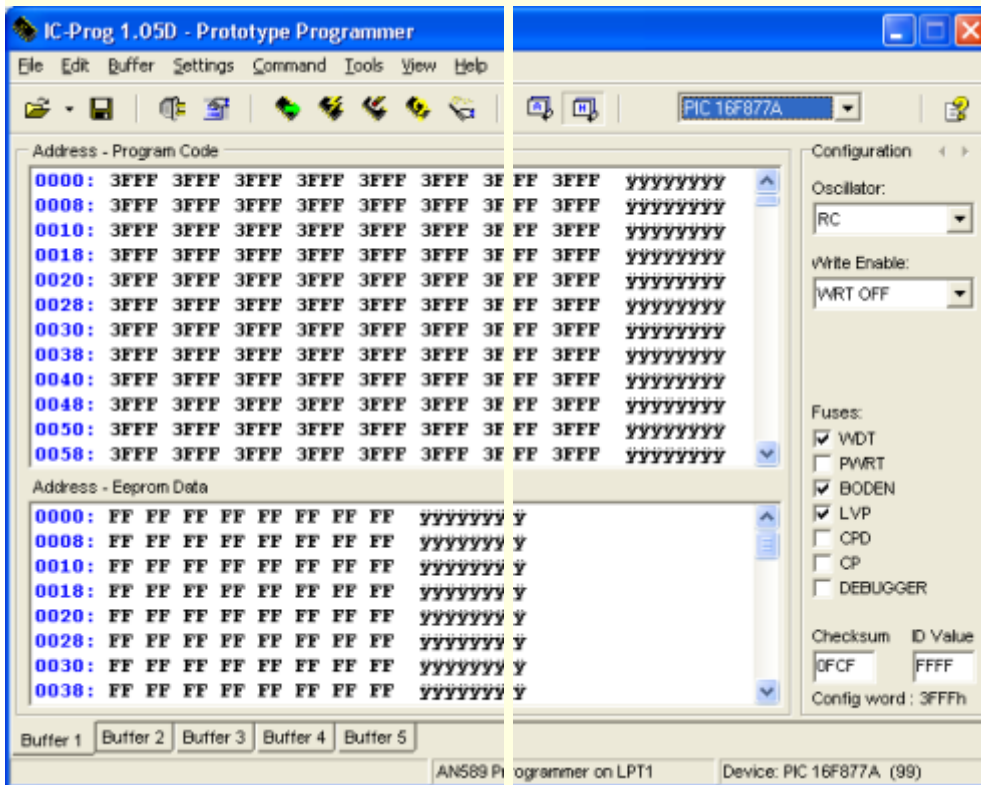
## Select Device

To use it for PIC programming select the device from the menu Settings--> Device--> Microchip PIC.

Here I have chosen a PIC16F877A device.

The Device menu also shows a lot of other devices that you can use with ICPROG.

Note: The screen changes to accommodate the chip features (here EEPROM oscillator, program 'Write enable' and chip configuration bits).



## Hardware check

You can check to see if ICPROG is communicating with your hardware by using the menu option Settings-->Hardware check. This lets you toggle the state of individual ICSP signals and the VCC power.

Note you'll need an oscilloscope or multimeter to check the signal state.

## Load Hex file

The next thing to do with the PIC software is programming the PIC by loading up the hex file (unless you have a 12F675 that has an oscillator calibration value that you will want to keep -in this case do a read first).

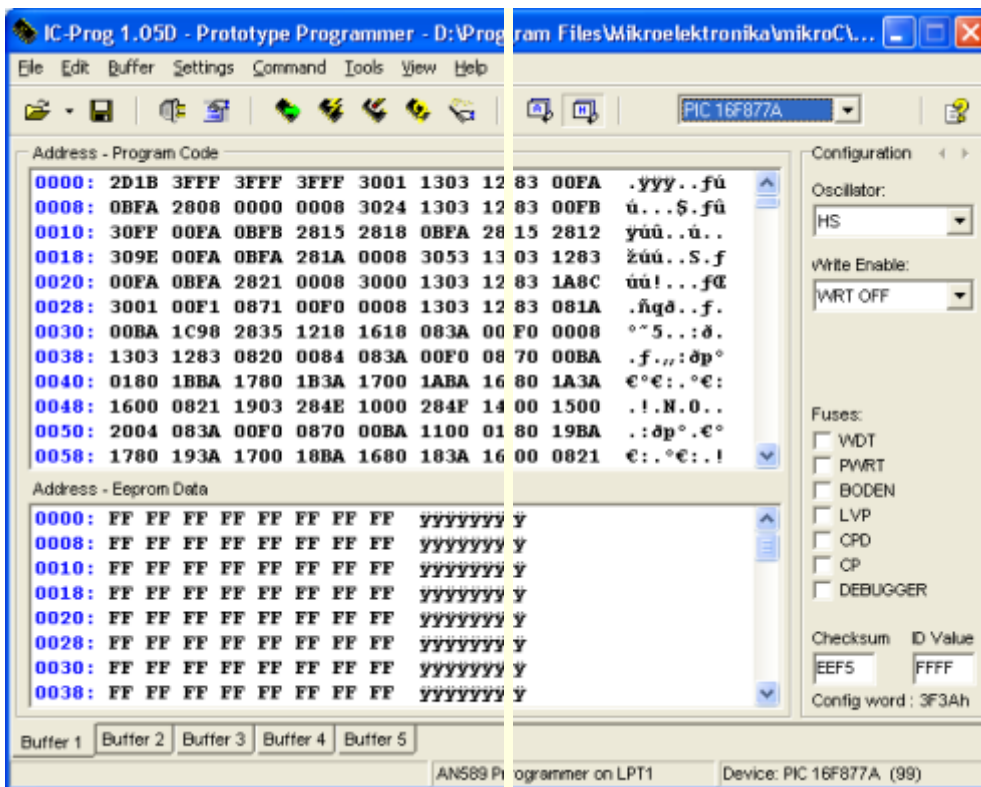
Note: You have to select the device manually it is not defined in the hex file. Although ICPROG stores the current device in use between sessions you have to select a new device manually when you start a new hardware project

Note: Fuses and oscillator settings are controlled from the hex file so all you need to do is check that they match your settings in the compiler or assembler.

Hit the open button to load your hex file.

Note how the oscillator, write enable and fuses have changed (controlled from the hex file):





## Erase device



The next PIC programming action is to erase the device by hitting the erase device button. It sends a command to the PIC which erases the whole device including protection bits (in newer devices). Old devices used to be un-usable after you had set the protection bits!

So this command quickly erases the chip.

Note: You don't have to use the erase button as the program button will overwrite the chip contents anyway. It is sometimes useful if you want to convince yourself that the program is actually programmed into the device correctly.

## Program device



The program device button does just that it programs the contents loaded from the hex file (in ICPROG memory) into the program memory of the PIC micro. If there is any EEPROM in the chip then it programs this as well. Finally it programs the configuration word.

## Verify device



The verify device button reads back the entire contents of the chip ensuring that it matches the hex file (loaded into ICPROG).

Note: Depending on settings in menu :

Settings--> Options--> Programming

verification may automatically happen at the end.

Once you get used to ICPROG and your hardware works reliably you may only want to do a verify if something does not work otherwise it takes more time up.

## Read device





The read device button reads back the entire contents of the chip into the currently selected buffer (shown at the bottom of the ICPROG window). So you can save the hex file or compare it to the contents of another buffer.

These are the only controls you need for successful PIC programming using ICPROG.

Note: The above discussion is left in for reference, use PicKit2 or PickKit3 and MPLABX for USB operation.