

# WinPicProg 1.91

## Free programming software for various of the MicroChip PIC series of micro-controllers by Nigel Goodwin.

Following the name change to WinPicProg I've now bought the domain name winpicprog.co.uk so all future updates will be under this address, with links from my [old pages](#). I'm happy to try and help with any problems people have, either via [email](#) or (preferably) via the new PHPBB [support forum](#), where other users will be able to take advantage of the answers. I've added an FAQ page to the support forum, and the first stop with any queries should be to look there!, all users are required to register, and will be emailed when any updates take place.

I've had a lot of requests for a circuit diagram for a programmer, so you can download the diagram for the [P16PRO40](#) programmer here, this works nicely with my software, and is one of the programmers I use. You can order it as a complete kit from [Dontronics](#). I've been sent PCB layouts for a couple of modified versions of the P16PRO40, you can download details on how to build them, [Ozipic](#) by Martin Sniedze and [P16PRO](#) by Carlier Patrick, Carlier has offered to make small quantities of boards available, you can contact him on [metacar@pi.be](mailto:metacar@pi.be).

At one time there were plenty of P16PRO40 (and other) kits available, but these seem to have all but disappeared. Happily, I received an email from a gentleman called "Gajjar Dinesh" in India, who's now selling a parallel port ICSP programmer, and he kindly sent me a sample to evaluate. He dispatched the kit on 25th August, and I received it on 30th August, which I thought was pretty good considering it spanned a weekend and a bank holiday Monday!. The kit includes a very high quality PCB (which you can order separately), and all the required components, and only took about half an hour to put together.

You can order the kit from his website at <http://products.foxdelta.com/icp.htm>, it's the **FD-ICP**, and he also sent me the optional "Programming Adaptor Board" which allows you to use it as a normal (non-ICSP) programmer - I would suggest you order that as well.

To use the **FD-ICP** you need to set WinPicProg to 'Custom Hardware' mode and set it like this, I also had to set the Vpp Delay to about 5mS, this is because of the reset pulse generated on the programmer board - you might need to experiment with this setting. The reset pulse is a vital part of ICSP, and is a major reason why non-ICSP programmers don't work with ICSP.

	Vdd	Vpp	Vpp40	MCLR	Output	Clock	Input
Pin	D2	D3	D4	N/A	D0	D1	Ack
Invert	No	Yes	Yes	N/A	Yes	Yes	Yes

[Download Program Site 1 \(400kB\)](#)

[Download Program Site 2 \(400kB\)](#)

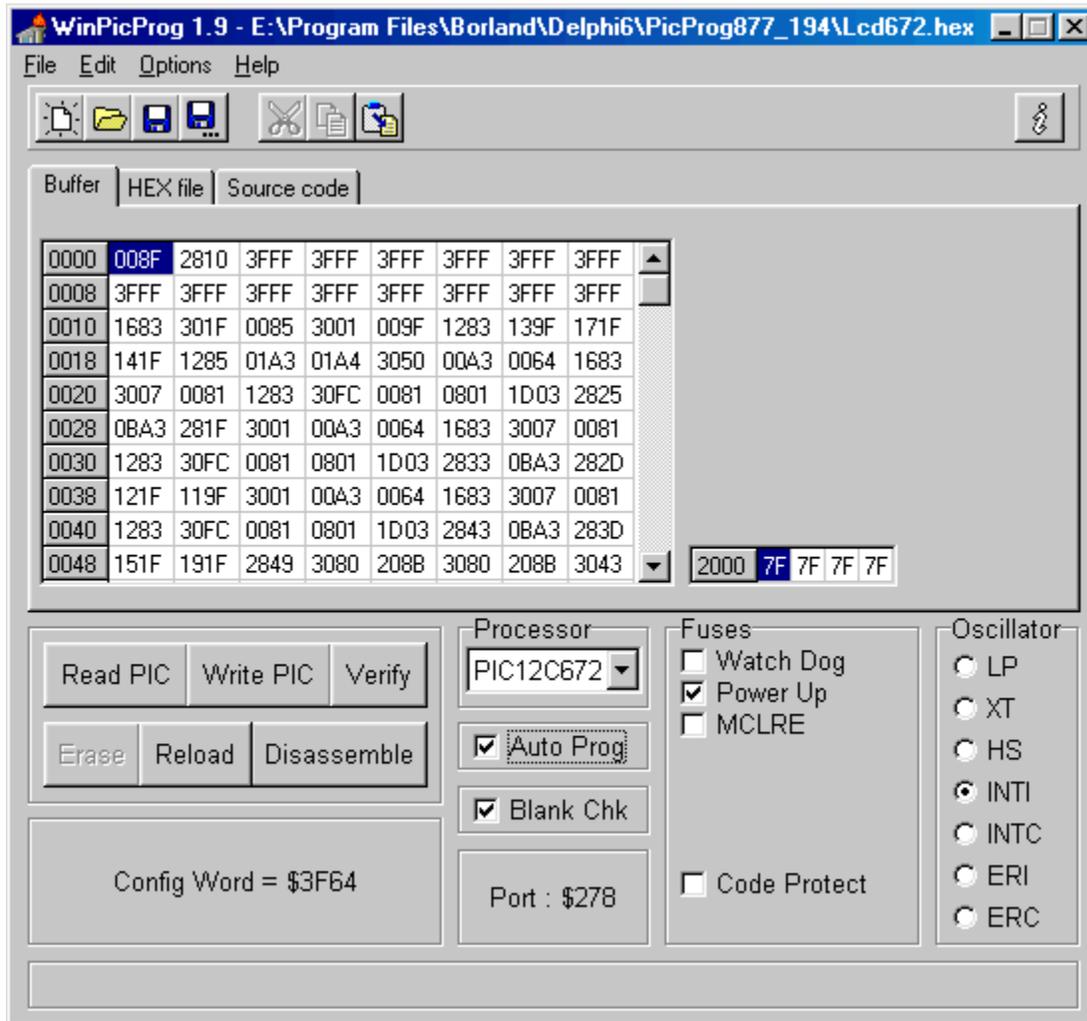
[Download Driver Site 1 \(1.5mB\)](#)

[Download Driver Site 2 \(1.5mB\)](#)

[Supported Chips](#)

[Support Forum!.](#)

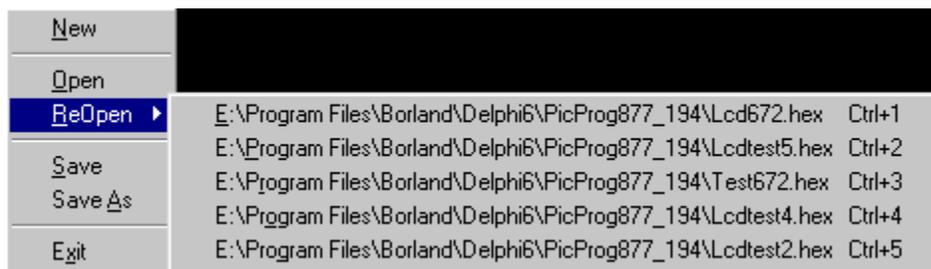
Screen Shot:



This is the latest revision of my PicProg series, it has undergone fairly drastic changes, and I've now added support for a limited range of EPROM based PIC's, in particular the 12C508/9, the 12C671/2, and the 16C505. Because of the increased number of chips supported, I've changed the Processor selection to a drop-down box. These supported EPROM PIC's are the only ones I currently have JW (UV erasable) versions of, if anyone wants a particular chip adding I'm willing to look into adding it, but would appreciate an erasable chip sending for testing (as the 16C505JW was - thank you Gene!). I haven't had time to update the help file for a fair while now, so it's no longer included with the program downloads. You can download the last version of the help file from here [Download HelpFile](#).

This picture shows the main screen, set to write to a PIC12C672, with a file already loaded and the filename displayed at the top. As the 12C672 doesn't have any on-board EEPROM data area, the buffer area to the right hand side is blank. Also, as this is an EPROM chip, the 'Blank Chk' box is visible, as this is checked (the default setting) the chip will be checked for being blank before programming - any problems with stored Oscillator Calibration values will result in a prompt, asking for a Yes/No reply before continuing. If you only want to alter a few bytes (obviously only ones to zeros), you can do so by un-checking the 'Blank Chk' box, an already programmed chip can then be written with no checks performed.

With the addition of the 12 bit EPROM chips I've had to extend the dis-assembler, it now copes with both 12 and 14 bit code, it also now is able to display larger source files, and I've added a much greater range of register and flag labels, it appears to be working extremely well - as usual, I'll be pleased to hear of any problems or suggestions. One 'problem' I already know off (and don't really see a satisfactory resolution for!) is dis-assembling of Org instructions in 12 bit code, with 14 bit code I do this by checking for a blank byte (\$3FFF), but with the 12 bit code a blank byte (\$FFF) is a valid instruction 'xorlw 0xff', which amazingly was used in the first code I tried it on!, as things stand it will be dis-assembled as an Org line - but will still assemble and run as normal.



Following previous suggestions by users I've added a couple of new facilities, firstly I've added a 'ReOpen' option to the File Menu, this displays the last 5 files you loaded, allowing you to select them with a single mouse click - or by holding the Ctrl key down and pressing the numbers 1 to 5 - this is shown clearly in the picture on the left. Secondly, I've added an 'Auto-Update' option to the 'Options' menu, when turned on this checks for the loaded hex file changing - if it does change it prompts you to reload it. However! - if you also have 'Auto-Prog' selected it will automatically reload the changed file and blow the chip - without any user intervention (obviously you need a chip in the

programmer for this to work!). I've been using this a lot recently - it makes life very easy - edit the source file in your usual editor and save it to disk, run MPASM on the file to produce a .HEX file, and then load the file into the programmer and blow the chip (so far, nothing new!). Make sure 'Auto-Update' is checked, and 'Auto Prog' is checked, you can then minimize WinPicProg. Having decided the software needs revising, insert the chip back into the programmer (erasing a UV chip first), edit the source code with your editor, and save to disk, run MPASM, make sure the edited file is selected, and press 'Assemble' - assuming MPASM doesn't find any errors the .HEX file will be updated, WinPicProg will detect the change, reload the file and blow the PIC.

Following previous suggestions by users I've added a couple of new facilities, firstly I've added a 'ReOpen' option to the File Menu, this displays the last 5 files you loaded, allowing you to select them with a single mouse click - or by holding the Ctrl key down and pressing the numbers 1 to 5 - this is shown clearly in the picture on the left. Secondly, I've added an 'Auto-Update' option to the 'Options' menu, when turned on this checks for the loaded hex file changing - if it does change it prompts you to reload it. However! - if you also have 'Auto-Prog' selected it will automatically reload the changed file and blow the chip - without any user intervention (obviously you need a chip in the programmer for this to work!). I've been using this a lot recently - it makes life very easy - edit the source file in your usual editor and save it to disk, run MPASM on the file to produce a .HEX file, and then load the file into the programmer and blow the chip (so far, nothing new!). Make sure 'Auto-Update' is checked, and 'Auto Prog' is checked, you can then minimize WinPicProg. Having decided the software needs revising, insert the chip back into the programmer (erasing a UV chip first), edit the source code with your editor, and save to disk, run MPASM, make sure the edited file is selected, and press 'Assemble' - assuming MPASM doesn't find any errors the .HEX file will be updated, WinPicProg will detect the change, reload the file and blow the PIC.

**Windows 2000** - there's been a few problems with the program under Windows 2000, it has a problem loading files, and fails with an error displaying a '\$'. I don't know why this happens under 2000, but I'm looking into it as and when time allows, in the meantime if you select the 'HEX File' tab before you load a file, it works OK.

**Update: Nov 2002** - In version 1.91 I've now sorted the problems under Windows XP, which should also cure the same problem under NT and 2000. I've also added support for a couple more chips (the 12F629 and 12F675), and added a menu option to disable the automatic hardware detection.

I retain copyright to all programs, but they may be freely downloaded and distributed, as long as no charge is made. This means they may NOT be distributed via BBS's using premium rate (RIP-OFF!) phone numbers.

You can reach me by email at: [nigelg@winpicprog.co.uk](mailto:nigelg@winpicprog.co.uk) or [nigelg@lpilsley.co.uk](mailto:nigelg@lpilsley.co.uk)

**Visit my main website [lpilsley.co.uk](http://lpilsley.co.uk)**