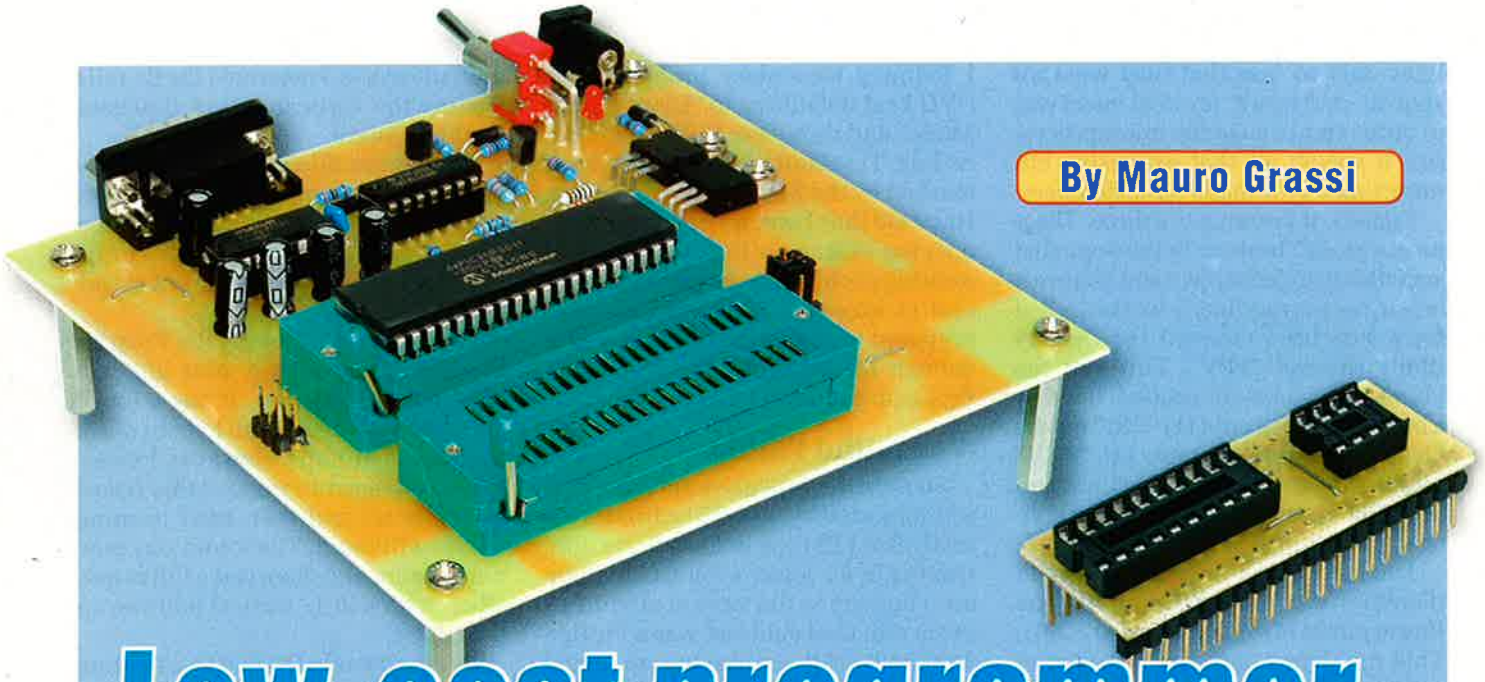


By Mauro Grassi

Low-cost programmer for dsPICs & PICs

This low-cost unit can program all dsPIC30F series microcontrollers in the DIP package, along with most PIC microcontrollers. It's easy to build and uses standard parts.

PICs ARE NOW ONE of the most widely used microcontrollers. Like all micros, they greatly simplify many electronic designs, are reconfigurable in the field and allow projects that would be unwieldy or overly complex without them. In addition, extra features can often be added retrospectively to the firmware.

Although the PIC family of microcontrollers is well known (we have published many projects that employ PICs), Microchip also manufactures the lesser-known dsPIC30F series of microcontrollers.

These are microcontrollers with similar peripherals to those found on standard PICs but which have an enhanced instruction set augmented with DSP (digital signal processing) type operations. They are 16-bit microcontrollers and are surprisingly powerful, running at speeds in the tens of MIPs (millions of instructions per second).

Dedicated single-cycle DSP operations like MAC (multiply and accumulate) allow them to perform real-time signal processing using multiple 40-bit accumulators. They also incorporate hardware multiplication and division and have surprisingly fast ADC acquisition modes. These features make them well-suited to many digital signal processing applications.

One such device, the dsPIC30F4011, will feature in a new digital Musicolour lightshow project to be published soon in SILICON CHIP. This particular device can perform a real-time FFT (Fast Fourier Transform) on audio-band signals with ADC acquisition modes that can operate at up to 1MS/s (1 million samples per second). It runs at close to 30MIPs and has 48kB of program memory.

Programming them

The dsPIC30F series of microcontrollers are extremely useful but most older PIC programmers cannot program them. This is due to incompatibilities with the pin-outs of the dsPIC family.

As a result, we have designed this simple, low-cost dsPIC and PIC programmer. It can program all the dsPIC30F family of microcontrollers that are available in a DIP package, as well as almost all regular PICs. It uses freely-available software (for the PC) and is easy to build.

By the way, if you have ever wanted to experiment with DSPs (digital signal processors), the dsPIC30F series is a good starting point. Microchip offers a lot of documentation and source code for free on their website **www.microchip.com**.

Programming procedure

Our new programmer is based on the original COM84 style programmer – so named because it was designed to program 16F84 microprocessors from

a serial port. There are really three lines which are necessary to program most PICs and microcontrollers in the dsPIC30F family: CLOCK (PGC), DATA (PGD) and V_{PP} (programming voltage).

Incidentally, the dsPIC30F family has two programming modes – enhanced and standard. The enhanced mode is faster and requires a programming executive or "bootloader" to be programmed in first. However, this programmer uses only the slower ICSP mode that is standard across the PIC family (ICSP = In-Circuit Serial Programming).

If you are interested in the details of the ICSP protocol, refer to the Microchip website at **www.microchip.com** (look for the "memory programming specifications").

Programming mode is entered by raising V_{PP} up to around 13V. Data is then programmed into the microcontroller by serially shifting commands and data using the PGC and PGD lines.

The PGC line synchronises the exchange of serial bits, while the PGD line contains the data. The PGD line is bidirectional, allowing reading and writing of the microcontroller.

For example, there is a command code for "Erase" which will erase the flash memory of the microcontroller. There are also commands for "Writing" and "Reading" pages, etc. As soon as the microcontroller enters programming mode, it starts listening for commands.

Circuit details

To successfully program a PIC or dsPIC series microcontroller, we must be able to control the PGC, PGD and VPP lines in the correct fashion. The SILICON CHIP dsPIC/PIC Programmer achieves this by giving control of these lines to the software running on a PC. This software program is called "WinPIC" and it makes sure that the correct procedure is followed for a particular device.

Fig.1 shows the circuit details. As can be seen, the dsPIC/PIC Programmer has two distinct supply rails (+5V & +13.6V) and these are derived from the DC supply rail using two 3-terminal regulators (REG1 & REG2). S1 is the power on/off switch, LED1 provides power indication and diode D1 provides reverse polarity protection.

REG2 is an LM317T variable voltage regulator. Its output is determined by the bias applied to its ADJ terminal, as determined by the voltage divider formed by the 120Ω resistor and the series 1.1kΩ & 82Ω resistors.

If R1 is the resistance between the OUT and ADJ terminals (120Ω in our case) and R2 is the resistance between ADJ and GND (1182Ω), then the LM317T will regulate its output voltage to: $V = 1.25 \times (1 + R2/R1)$. Note, however, that slight manufacturing variations mean that the 1.25 factor can be anywhere between 1.2 and 1.3 in actual practice.

In this case, R1 & R2 have been selected so that REG2 regulates its output to 13.6V in typical conditions. This provides the MCLR-bar/Vpp voltage for the microcontroller which should

Main Features & Devices Supported

Features

(1) Will program all dsPIC30F series microcontrollers in the DIP package
(2) Will program most PICs in DIP package
(3) Uses PC freeware WinPIC for Windows
(4) Connects to the serial (RS232) port of a PC
(5) Very low cost

Minimum Supported Devices (others may also work)

10F series
10F200/202/204/206 (E) (*)

12F series
12F508/509 (E)
12F609/615 (E)
12F629/675 (E) (*)
12F635/636/639 (E)
12F683 (E)

16F series
16F610/616 (E)
16F627/627A/628/628A (*)
16F630/631/636/639/676/677/684/685/687/688/689 (E)
16F648/648A
16F716
16F73/737/74/76/77
16F818/819
16F84/84A/87/88 (*)
16F870/871/872
16F873/873A/874/874A/876/876A/877/877A (*)
16F913/914/916/917

18F series
18F2220/2320/4220/4320
18F2331/2431/4331/4431
18F2420/2520/4420/4520
18F2450/4450
18F2455/2550/4455/4550 (*)
18F2480/2580/4480/4580
18F2525/26204525/4620
18F2439/2539/4439/4539
18F242/252/442/452/
18F2585/4585/2680/4680
18F248/258/448/458
18F2682/2685/4682/4685

dsPIC30F series
dsPIC30F2010 (*)
dsPIC30F2011/3012 (*)
dsPIC30F2012/3013 (*)
dsPIC30F3010 (*)
dsPIC30F3011 (*)
dsPIC30F3014/4013 (*)
dsPIC30F4011 (*)
dsPIC30F4012 (*)

(*) = tested & passed. (E) = requires external connection or adaptor socket.

Parts List

1 PC board, code 07105081, 122 x 120mm
1 adaptor PC board, code 07105083, 52 x 19mm
1 16V 400mA DC plugpack
1 SPDT right-angle PC-mount toggle switch (S1)
1 PC-mount 2.5mm DC socket (CON1)
1 DB9 female right-angle socket (CON2)
1 DIP14 IC socket
1 DIP16 IC socket
2 DIP40 ZIF sockets
2 jumper shunts
1 8-pin DIL header with 2.54mm spacing
1 6-pin DIL header with 2.54mm spacing
1 500mm length of 0.7mm tinned copper wire
4 M3 x 6mm screws
2 M3 nuts
2 M3 x 10mm screws
4 9mm long M3 tapped spacers

Semiconductors

1 MAX232A RS232 line driver receiver (IC1)
1 74LS04 hex inverter (IC2)
1 BC337 NPN transistor (Q1)
1 BC327 PNP transistor (Q2)
1 7805 5V regulator (REG1)
1 LM317T regulator (REG2)
3 1N4004 diodes (D1-D3)
1 red 3mm LED (LED1)

Capacitors

1 10µF 16V electrolytic
7 1µF 16V electrolytic
2 100nF monolithic (code 100n or 104)
2 22pF ceramic

Resistors (0.25W, 1%)

6 2.2kΩ
1 1.1kΩ
1 120Ω
1 82Ω
3 39Ω

ideally be between 12.8V and 13.1V. However, anything from 13.4V to 13.8V is actually OK at REG2's output, since this is fed through transistor switch Q2 and series diode D2 before being applied to the MCLR-bar/V_{PP} (master clear/programming voltage) pin of the microcontroller to be programmed.

In operation, the regulated 13.6V rail from REG2 is switched on and off by PNP transistor Q2 which in turn is switched on and off by NPN transistor Q1. When pin 3 (Tx) of the serial port is high, it will switch Q1 on, in turn switching Q2 on and applying around 13V to the MCLR-bar/V_{PP} pin on the microcontroller to be programmed.

Conversely, when pin 3 of the serial port is low, Q1 will be off and therefore Q2 will also be off. In this case, the 2.2kΩ resistor on D2's cathode will pull the MCLR-bar/V_{PP} pin low.

Basically, on a PIC or dsPIC microcontroller, the MCLR-bar/V_{PP} pin acts either as a Reset (0V) or a programming voltage pin (around 13V for PICs or between 9V and 13V for a dsPIC30F series microcontroller). When MCLR-bar/V_{PP} is low, the microcontroller is in the Reset state (meaning that all its configurable pins are high impedance inputs). When it is high (around V_{DD} = +5V), the microcontroller runs in program mode and if it is at Vpp the microcontroller will enter programming mode.

It was a deliberate design decision to switch the MCLR-bar/V_{PP} line between 0V and V_{PP} rather than between V_{DD} and V_{PP}. This was done to avoid possible damage to the microcontroller being programmed.

To explain, if the MCLR-bar/V_{PP} line were switched between V_{DD} and V_{PP}, the program would run on the microcontroller when programming finishes. If that program were to drive the output pins (as digital outputs or as peripheral outputs), it could cause excessive currents to flow and damage the output stages of those pins.

That's because the ZIF sockets have many power connections to accommodate different PICs and dsPICs (+5V and GND). As a result, some of the microcontroller's output pins could be shorted to +5V or to ground if the program were to run.

For this reason, the V_{PP} pin is switched from 0V to 13V so that the microcontroller is never in the running mode.

Of course, if you were to incorporate this programmer onto a PC board that catered for ICSP (in-circuit-serial-programming) then you would have this line switch from V_{DD} (+5V) to 13V and the reset would occur on any transition from 13V down to 5V. Refer to the section entitled "External Programming Using CON3") for more details.

Note that some PIC microcontrollers can be configured to disable the Reset function of the MCLR-bar/V_{PP} pin, allowing it to be used for an alternative (multiplexed) function. This should be avoided when using this programmer with a dsPIC or PIC plugged into a ZIF socket, for the reasons outlined above (this does not apply when using CON3 to program an external device).

Regulator REG1 is used to derive the +5V rail and this is used to power IC1, IC2 and the microcontroller being programmed. This +5V rail is bypassed using 10µF, 1µF and 100nF capacitors, while a 1µF capacitor also bypasses REG1's input.

Control lines

The relevant lines used in the RS-232 serial interface to control the dsPIC/PIC Programmer are derived from pins 3, 4, 5, 7 & 8.

Pin 5 is the ground connection while pins 3, 4 & 7 (respectively Tx, DTR and RTS) are outputs from the serial port. In particular, pins 4 & 7 are digital outputs, while pin 3 is usually the Transmit line of the serial port. These are controlled by the WinPIC software on the PC as appropriate.

Finally, pin 8 (CTS) is an input pin and this is used to read data from the microcontroller, as required to verify or read the state of the memory.

IC1 is a MAX232 RS-232 line driver receiver. Its job is to translate between the RS-232 voltage levels (ie, ±10V) at the serial port and the TTL levels (0-5V) used by the microcontroller. As mentioned, pins 4 & 7 of the serial port are standard digital outputs and these are connected directly to IC1.

In operation, the MAX232 actually inverts the levels and so its outputs at pins 9 & 12 are fed to inverter IC2a & IC2f (part of a 74LS04 hex inverter) to invert them back again.

Pin 7 of the serial port controls the PGC (CLOCK) line and is applied to the microcontroller via IC1, IC2a and a 39Ω resistor (to limit the current). In addition, a 22pF ceramic capacitor is used to filter any high-frequency noise on this line.

Pin 4 controls the PGD line (DATA) output. When it goes low, so does the pin 12 output of inverter IC2f. Diode D3 allows a low level from IC2f to drive the PGD line but blocks high-level signals from IC2f. A 2.2kΩ pull-up resistor is used instead to pull this line high. This allows the WinPIC software to read the PGD line from the microcontroller via pin 8 of the serial

port (ie, after sending pin 4 of the serial port high).

So the PGD line is actually "bi-directional" and is used as an output when writing to the microcontroller and as an input when reading from the microcontroller.

Note that, as with the PGC line, the PGD line is fed via a 39Ω resistor and is filtered using a 22pF ceramic capacitor to reduce spurious noise.

Two ZIF (zero insertion force) sockets are used to accept the microcontroller to be programmed. ZIF SKT1 is used for dsPIC30F series microcontrollers and they should always be aligned with their pin 1 going to pin 1 of the ZIF socket.

Alternatively, ZIF SKT2 should be used for programming standard PICs like the 16F88. As before, pin 1 of the microcontroller goes to pin 1 of the ZIF socket.

Note, however, that the 10F and 12F series of PICs are not compatible with the onboard ZIF socket. These must be programmed via an external adaptor board, as described later, or by using CON3 and a breadboard.

External programming

CON3 is a 6-pin header and its pinout is arranged as shown in Table 1. It can be used to access the five relevant lines required to program both PICs and dsPICs externally (see the section entitled "External Programming Using CON3" below).

For example, if your PIC is not actually compatible with the pinning of ZIF SKT2 (eg, if you have a PIC10F202), then you may use this connector to access the relevant lines. These lines can be connected to, say, a breadboard, to program your PIC off the PC board. Of course, you can also use this connector to program microcontrollers in circuit as well.

Jumper settings

Finally, there is an 8-pin header which accepts jumper shunts JP1-JP4. **However, only two of the four positions should ever be shorted at any one time.** Table 2 shows the jumper functions.

In practice, you must set these according to the microcontroller being programmed. Either JP1 or JP2 (but not both) must be shorted according to the type of dsPIC being programmed in ZIF SKT1, while JP3 or JP4 (but not both) must be shorted according

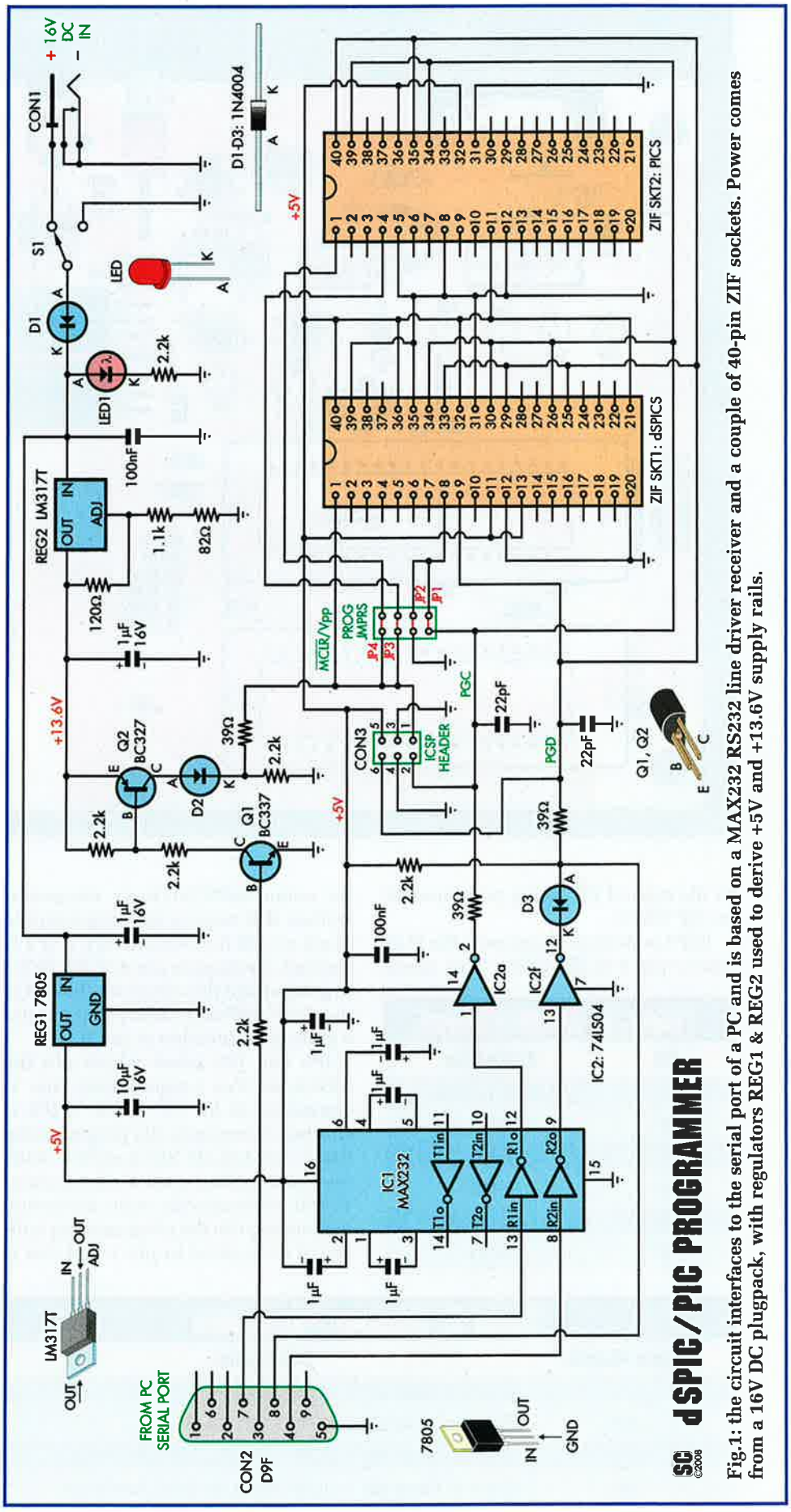


Fig.1: the circuit interfaces to the serial port of a PC and is based on a MAX232 RS232 line driver receiver and a couple of 40-pin ZIF sockets. Power comes from a 16V DC plugpack, with regulators REG1 & REG2 used to derive +5V and +13.6V supply rails.

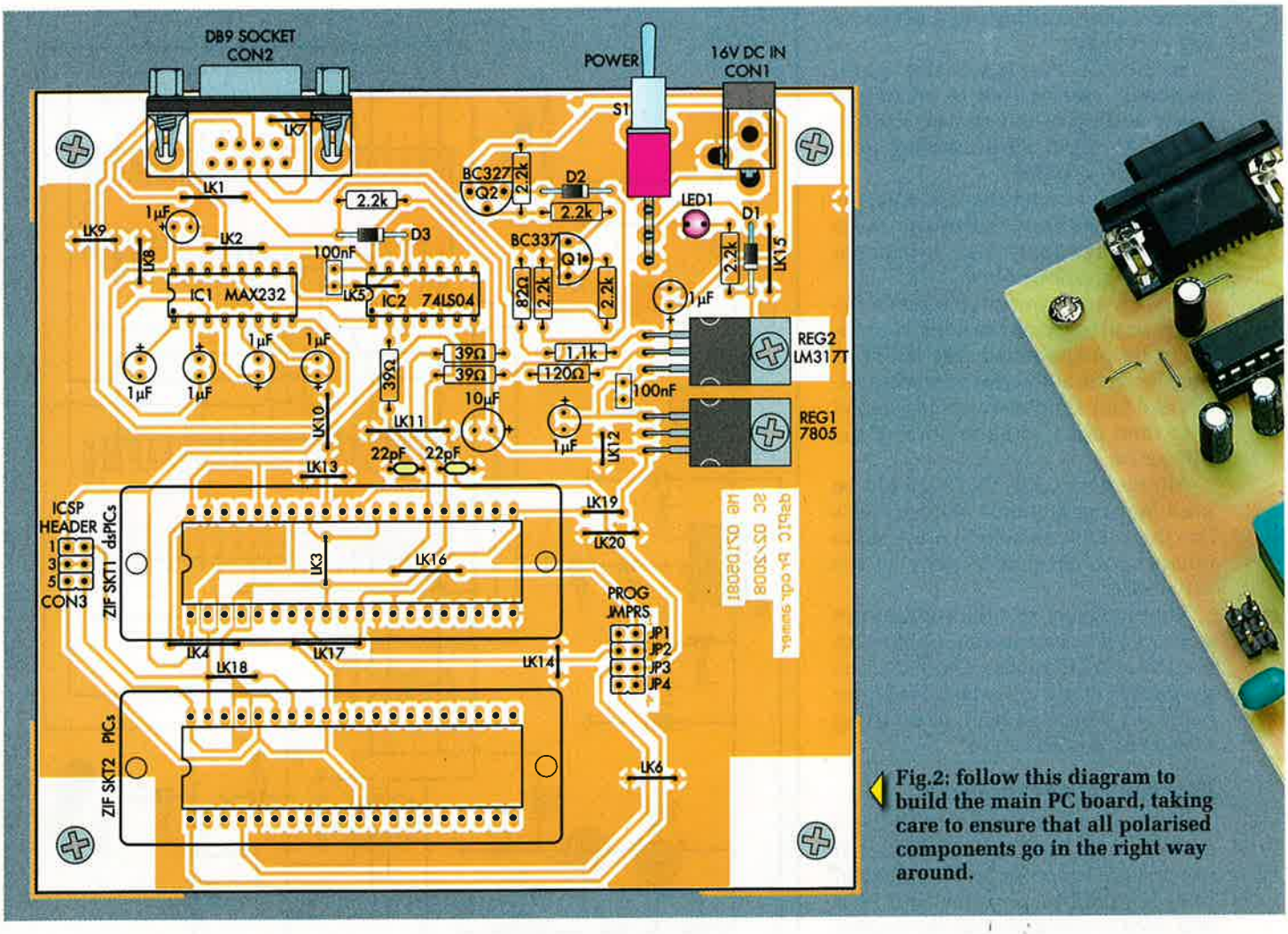


Fig.2: follow this diagram to build the main PC board, taking care to ensure that all polarised components go in the right way around.

to the type of PIC being programmed in ZIF SKT2.

If JP1 is shorted, it connects the PGC line to pin 8 of ZIF SKT1. This caters for some dsPIC30Fxxxx microcontrollers that require the programming clock on pin 8. Alternatively, if JP2 is shorted, it connects pin 8 of ZIF SKT1 to ground and this caters for the rest of the dsPIC30Fxxxx family that require a ground connection at pin 8.

JP3 and JP4 select which pin the MCLR-bar/V_{PP} programming line is connected to on ZIF SKT2. If JP3 is shorted, it connects the programming line to pin 4 of ZIF SKT2 and this suits microcontrollers such as the popular 16F88. Alternatively, some microcontrollers require the programming voltage to be applied to pin 1 and this is selected by installing JP4 instead.

Table 1: CON3 Pinout

Pin	Description
1	MCLR-bar/V_{PP}
2	PGC
3	GND
4	GND
5	+5V rail (V_{DD})
6	PGD

Table 2: Jumper Functions

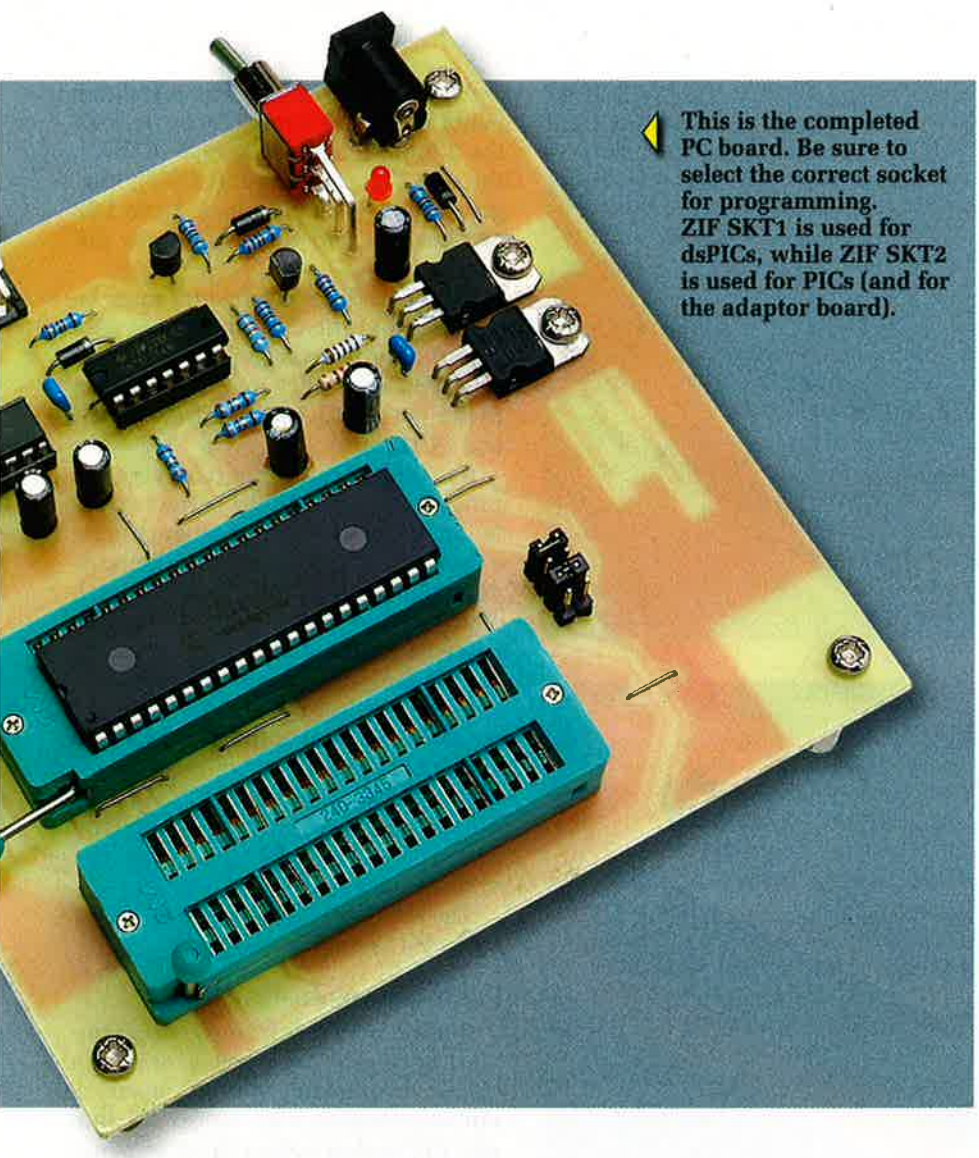
Jumper Number	Description
JP1	Short to make pin 8 of ZIF SKT1 the PGC pin
JP2	Short to make pin 8 of ZIF SKT1 GND
JP3	Short to make pin 4 of ZIF SKT2 the /MCLR/VPP pin
JP4	Short to make pin 1 of ZIF SKT2 the /MCLR/VPP pin

Warning: it is quite possible to damage a microcontroller installed in either ZIF socket by incorrectly setting jumpers JP1-JP4, so check Tables 2 & 4 carefully before inserting a microcontroller into its socket and applying power. However, a more likely outcome is that you will not damage the microcontroller (as they usually have protection diodes) but the programming will not be successful.

In summary, you must install either JP1 or JP2 (but NOT both) when programming a dsPIC and either JP3 or JP4 (but NOT both) when programming a PIC.

External Programming Using CON3

The 6-pin header CON3 can be used to program a PIC or dsPIC that's either mounted in-circuit on a separate board or installed on a breadboard. For example, this is one way of programming a PIC microcontroller that doesn't have



This is the completed PC board. Be sure to select the correct socket for programming. ZIF SKT1 is used for dsPICs, while ZIF SKT2 is used for PICs (and for the adaptor board).

a compatible pin-out with the ZIF sockets – see Table 4.

Devices that fall into that category include the 10Fxxxx and 12Fxxxx series of PICs, as well as some of the 16Fxxxx series.

The pin-outs for CON3 are shown in Table 1 and include the GND, +5V, MCLR-bar/V_{PP}, PGC and PGD lines. These are the only lines you will ever need to program your microcontroller.

If the microcontroller is on a powered board, you can ignore the +5V line (pin 5) and simply connect CON3's GND (pin 3 or 4) to the ground of your board. It's then simply a matter of connecting the MCLR-bar/V_{PP}, PGC and PGD lines to the appropriate pins on your PIC or dsPIC.

Optional Adaptor Board for 10F & 12F series PICs

We have also designed an optional adaptor board for 10F and 12F series PICs – see Fig.3. This adaptor plugs directly into ZIF SKT2 on the dsPIC/PIC Programmer and the position of the jumper on JP3 or JP4 is irrelevant when using the adaptor.

As shown in Fig.3, the adaptor has 20-pin and 8-pin IC sockets. The 8-pin socket is for 10F series PICs and the 20-pin socket is for 12F series PICs. As usual, the microcontroller to be programmed should be oriented so that its pin 1 is connected to the socket's pin 1. In addition, pin 1 of the adaptor board goes to pin 1 of ZIF SKT2.

You will need to refer to the microcontroller's datasheet and ensure that the pin-out is compatible with the ZIF socket by referring to the schematic diagram.

Construction

The dsPIC/PIC Programmer is built on a PC board coded 07105081 and measuring 122 x 120mm. The companion adaptor board is coded 07105082 and measures 52 x 19mm. Fig.2 shows the main board layout, while Fig.3 shows where the parts go on the adaptor board.

As usual, begin by checking the PC boards for defects, such as breaks in the tracks or shorts between adjacent tracks. It's rare to find any problems these days but it's still a good idea to check, as defects can be difficult to spot after the parts are installed.

Once these checks are completed, start the main board assembly by installing the 20 wire links. Use tinned copper wire for these links and make sure that they are nice and straight. You can straighten the link wire by clamping one end in a vice and these stretching the wire slightly by pulling on the other end with a pair of pliers.

Note that link LK7 goes under the RS-232 socket (CON2), while LK3 & LK6 are under ZIF SKT1.

Follow these with the 12 resistors. Table 1 shows the resistor colour codes to help you select the values but it is prudent to check each one using a DMM before it is soldered in place,

Table 3: Resistor Colour Codes

❑	No.	Value	4-Band Code (1%)	5-Band Code (1%)
❑	6	2.2kΩ	red red red brown	red red black brown brown
❑	1	1.1kΩ	brown brown red brown	brown brown black brown brown
❑	1	120Ω	brown red brown brown	brown red black black brown
❑	1	82Ω	grey red black brown	grey red black gold brown
❑	3	39Ω	orange white black brown	orange white black gold brown

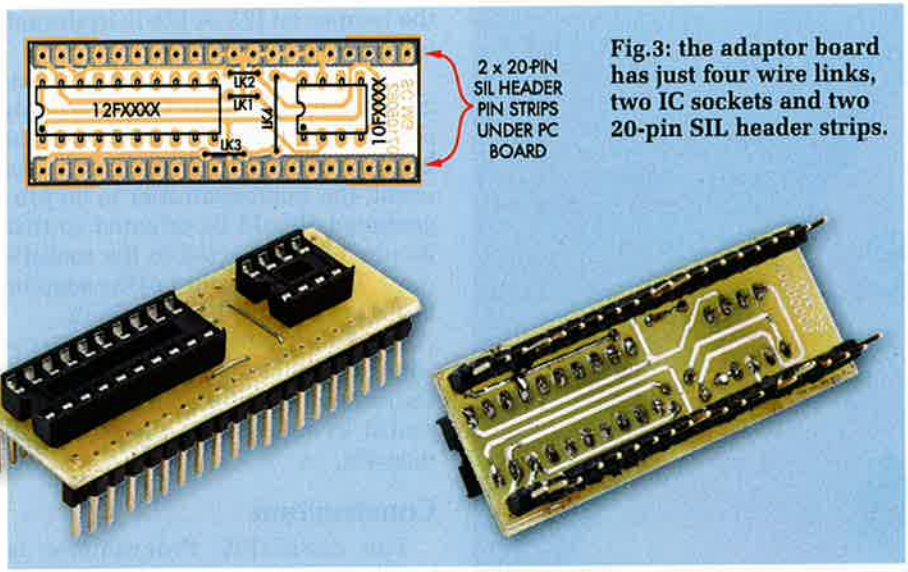
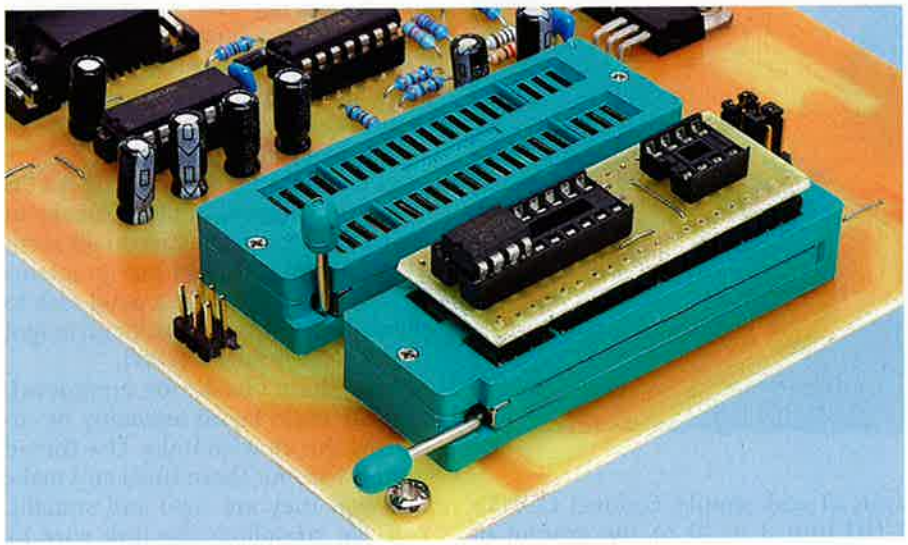


Fig.3: the adaptor board has just four wire links, two IC sockets and two 20-pin SIL header strips.



The adaptor board is used for programming 10F & 12F series PICs. As shown here, it plugs into ZIF SKT2 on the dsPIC/PIC Programmer board.

as some colours can be difficult to decipher.

The three diodes are next on the list. Be sure to install them with the correct polarity, as indicated on the parts layout diagram (Fig.2). Once they're in, install the two transistors, again making sure that they are correctly oriented.

Don't get the transistors mixed up. Q1 is a BC337 NPN transistor, while Q2 is a BC327 PNP type. Check that each is installed in its correct location.

Now for the capacitors: the ceramic and monolithic types are not polarised and can go in either way around. However, the electrolytic capacitors are polarised, so be sure to install them correctly.

The next step is to install IC sockets for IC1 & IC2. Again, make sure that these parts go in the right way around – ie, notched ends to the right. Note, however, that these sockets are optional. **Do not install the ICs at this stage – that step comes later, after the power supply has been checked out.**

Regulators REG1 & REG2 can now be mounted. These are both installed with their metal tabs flat against the PC board. To do this, first bend their leads down by 90° about 6mm from their bodies. That done, fasten each regulator to the PC board using M3 x 10mm screws and nuts, then solder their leads.

Do NOT solder the leads before bolting the devices down, as this could crack the soldered joints and damage the PC board as the nuts are tightened. Make sure also that each device is installed in its correct location.

All that remains now is to install the major hardware items. These include the 2.5mm DC power socket (CON1), the RS-232 connector (CON2), toggle switch S1, the 6-pin & 8-pin DIL pin headers and the two 40-pin ZIF sockets.

Note that the 8-pin header must be installed but the 6-pin header is necessary only if want to program a PIC or dsPIC externally and need access to the +5V, GND, MCLR-bar/V_{PP}, PGC and PGD lines!

Be sure to install the two large 40-pin ZIF sockets with the correct orientation. If you will only be programming a few microcontrollers occasionally, you can replace these with much cheaper IC sockets but the ZIF sockets make life much easier (and are worth the extra money in our opinion).

Finally, secure four M3 x 9mm spacers to the corner positions of the board using M3 x 6mm machine screws. These are used to support the board off the bench top during use. If you like, you can also fit four rubber feet to these spacers.

The dsPIC/PIC Programmer is now ready for testing.

Preliminary testing

Before using this new programmer, it should be given a thorough check. **Important: do not insert a microcontroller (PIC or dsPIC) into any ZIF socket before these tests are completed.**

A 16V DC plugpack should be used to power the dsPIC/PIC Programmer, although you can also probably use a 15V DC plugpack (just). Apply power and you should see the red indicator LED light. If it doesn't, check the supply polarity and if that's OK, check the polarity of the LED.

Assuming that the LED lights, the next step is to check the voltages at the outputs of the two regulators. You should measure +5V at the output of REG1 (anything from 4.8-5.1V is normal), while REG2's output should be close to 13.6V (13.4-13.8V is OK).

If REG's output is lower than 13.4V, increase the value of the 82Ω resistor (eg, to 120Ω) to bring it into the 13.4-13.8V range. Conversely, if the output is higher than 13.8V, decrease the value of the 82Ω resistor.

Alternatively, if REG2's output is outside the designated range, check the voltage between REG2's OUT & ADJ terminals. This value can then be

used to calculate a new value for R2 from the formula given in the circuit description.

If the supply rails are correct, switch off and fit IC1 & IC2 to their respective sockets. That done, connect a serial cable between the programmer and your PC.

Adaptor board assembly

Fig.3 shows the parts layout for the adaptor board. It's a snap to assemble – just install the four wire links, the two IC sockets (watch their orientation) and the two 20-pin SIL pin headers.

Note that the pin headers are mounted on the copper side of the board. To install them, push their longer pins through until they sit flush with the top of the PC board, then initially solder just a pin at either end. The remaining pins can then be soldered, after which the plastic strips are slid down the pins until they rest against the soldered joints.

You are now ready to install the WinPIC software on your PC.

Software installation

As mentioned above, the software to use with this programmer is WinPIC, available from either **http://freenet-homepage.de/dl4yhf/winpicpr.html** or from the SILICON CHIP website at **www.siliconchip.com.au**. Once it has been downloaded, it's installed by running the executable file ***winpicsetup.exe***.

By the way, do not confuse WinPIC with other software that's available, such as WinPIC800. The latter is a completely different program and it will NOT work with this programmer.

Setting up WinPIC

After installing WinPIC, you should make sure that it is correctly set up to work with the programmer. Here's how to configure WinPIC:

(1) Start WinPIC and click on the "Interface" tab (see Fig.4);

(2) Ensure "COM84 programmer for serial port" is selected from the drop down menu;

(3) Ensure that the correct COM port is set;

(4) Check that both ZIF sockets are empty and that the programmer is connected to the PC via a serial cable;

(5) Apply power to the programmer and click on "Initialize!";

(6) In the "Options" tab, select either PortTalk or SMPORT (both are faster than using the Windows API). By contrast, if you wish to use a USB-RS232 converter cable, you are probably safer selecting the "no direct access at all, only use win API" option. This will be slower but will ensure that WinPIC accesses the correct windows drivers installed for your USB-RS232 converter. Refer to the section "Using USB-RS232 Converters" in the accompanying panel for more information.

Using A USB-RS232 Converter Cable

This dsPIC/PIC Programmer is designed to work with native RS-232 serial ports. However, many computers today, especially notebooks, do not have a serial port, as it has been superseded by USB.

Although USB-to-RS232 converter cables are available, not all will work correctly with this programmer. And for those that do work, programming may be considerably slower compared to working direct from a serial port.

The reason some converters don't work has to do with the low-level interface and the implementation of the USB-to-RS232 converter. In particular, the problem arises because some USB-to-RS232 converters are imperfect emulations of the serial port.

In normal use, pin 3 (Tx) of the RS232 serial port is the transmit line, used to send data at the selected baud rate. Most USB-to-RS232 converters will correctly emulate this, as it is necessary for full duplex data transmission.

However, COM84 style programmers like this one use pin 3 (Tx) of the serial port for the programming voltage and hence as a simple digital output. This is an unconventional use of the Tx line. It is accomplished in the WinPIC software by setting the "break" flag in the line control register (bit 6). However, some USB-to-RS232 converters (and their supplied software driver) do not emulate the break flag functionality and therefore will not work with this programmer.

USB-to-RS232 converters based on the newer FTDI chips, especially the FT232R, could possibly work, given that the specifications claim that the FT232R has inbuilt support for line break. It is, of course, up to the manufacturer of the USB-to-RS232 converter as to whether the full features of the interface ICs are supported through the supplied software driver.

If you would like to try a USB-to-RS232 converter with this programmer, you should make sure that it supports line break and that the "no direct access at all, only use Win API" option is selected in the "Options" tab of WinPIC. This means that WinPIC will not access the serial ports directly but only through the Windows API.

This ensures that WinPIC talks to the windows driver for your USB-to-RS232 converter, rather than trying to access ports that are not implemented. As indicated above, this may result in substantially slower operation than with a native serial port.

In our case, we tested the Prolific GUC-AD9 USB-RS232 converter on Windows XP and it worked. The only drawback was that it was slow – up to 10 times slower than when running the programmer direct from a serial port.

This is related to latencies in the windows API and the windows driver for the converter. A small delay (in the order of milliseconds) occurs when switching any control line and these small delays all add up to a considerable delay due to the huge number of switching requests made by WinPIC.

Note: the Prolific GUC-AD9 USB-RS232 converter is available from Jaycar (Cat. XC4834).

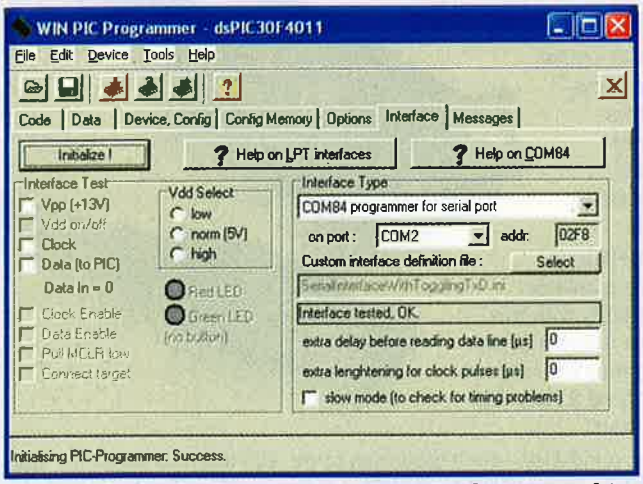


Fig.4: clicking the Interface tab in WinPic brings up this window. Ensure "COM84 programmer for serial port" is selected for the Interface Type and be sure to choose the correct COM port.

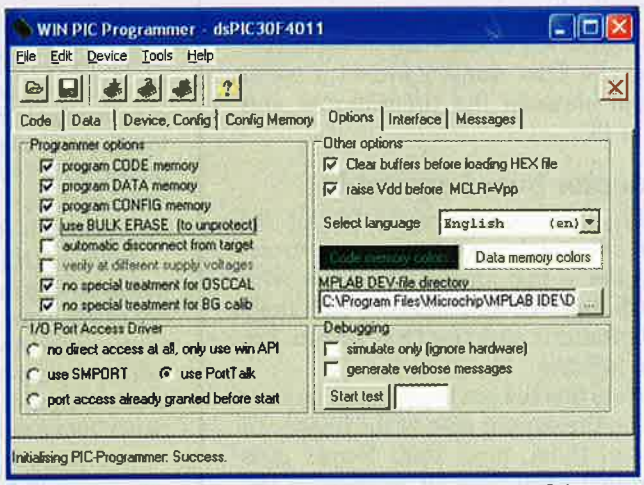


Fig.5: after selecting the device to be programmed (see text) go to the Options tab and select the options shown here. The dsPIC or PIC can then be programmed as outlined in step 4.

Once the programmer has been initialised correctly by WinPIC, you are ready to program some PICs. Here's the procedure, step-by-step:

(1) Check that the power is off, then insert the PIC or dsPIC you wish to program into its corresponding ZIF socket (according to Table 4).

(2) Set the jumpers as indicated in Table 4. Note that either JP1 or JP2 (but NOT both) must be installed for dsPICs. Similarly, either JP3 or JP4 (but NOT both) must be installed for PIC microcontrollers, as set out in the table.

If these jumpers are incorrect, programming will almost certainly fail.

(3) Once the jumpers have been set, apply power, start WinPIC, go to ***Device –> Select*** and select the PIC or dsPIC you wish to program from the drop down menu. That done, go to the "Options" tab and select the options as shown in Fig.5.

(4) To program the dsPIC or PIC, go to ***File –> Load –> Program Device*** and select the hex file to be programmed. Note that the fuse bits should be within the hex file and they will be programmed as well.

WinPIC should now start to program your device and then verify its contents. You can use the "Code" tab to see the program memory.

If programming is successful, you should see the message "Programming finished, no errors" at the bottom lefthand corner of the window.

You can also erase, read and verify a microcontroller using WinPIC, although you should keep in mind that reading a code protected device will result in zero readings for the program memory bytes. For more detailed information on how to use WinPIC, refer to its help menu.

Finally, note that WinPIC accesses the serial port on your PC and requires real-time control of the programming signals. It is therefore possible that it will lock up while programming is in progress and fail to respond to mouse or keyboard commands.

To prevent this, avoid having other Windows programs running in the background while WinPIC is programming a device. If the WinPIC window stops responding when programming a device, simply wait for it to finish.

If everything is working correctly, you should see the message "Initialising PIC-Programmer: Success" at the bottom of the WinPIC window, as shown in Fig.5.

Troubleshooting

If you receive the message "WARNING: Could not initialize programmer!" instead, you can test the interface manually to narrow down the list of possible problems. Here's what to do:

(1) Clicking the "V_{PP}(+13V)" box should toggle pin 1 of CON3 (the external programming header) from 0V (box un-ticked) to around +12.5-13V (box ticked). If this doesn't happen, check that transistors Q1 & Q2 are the correct types. If they are, trace the signal from pin 3 of the serial port to pin 1 of CON3, checking at each stage that the signal toggles as this box is "ticked" and "un-ticked" in WinPIC.

(2) Clicking on the "Clock" box should toggle pin 2 of CON3 from 0V (un-ticked) to around +4-5V (ticked).

If that doesn't happen, check the MAX232 and its surrounding capacitors. That done, check the signal at pin 7 of the serial port, then at pins 13 & 12 of IC1, pin 1 of IC2, pin 2 of IC2 and finally pin 2 of CON3.

Note that the MAX232 (IC1) should level translate the signal level at pin 13 to about +5V at pin 12.

(3) Clicking on the "Data (to PIC)" box should toggle pin 6 of CON3 from 0V to around +3.5-5V and you should see the "Data In=" field change from 0 to 1. The latter should be 0 with the box un-ticked and 1 otherwise.

If this is not the case, check the signal at various points on the circuit from pin 4 of the serial port to pin 6 of CON3. Check also that pin 8 of the serial port is receiving the correct level (read by WinPIC and displayed in the "Data In=" field).

Read the FAQ

Finally, if the programmer is still not working, there could be issues with WinPIC. Refer to the online FAQ

Table 4: Setting Jumpers JP1,JP2 & JP3,JP4

Device	ZIF socket	JP1	JP2	JP3	JP4
10F200/202/204/206	Ext	N/A	N/A	N/A	N/A
12F508/509	Ext	N/A	N/A	N/A	N/A
12F609/615	Ext	N/A	N/A	N/A	N/A
12F629/675	Ext	N/A	N/A	N/A	N/A
12F635/636/639	Ext	N/A	N/A	N/A	N/A
12F675	Ext	N/A	N/A	N/A	N/A
12F683	Ext	N/A	N/A	N/A	N/A
16F610/616	Ext	N/A	N/A	N/A	N/A
16F627/627A/628/628A	ZIF SKT2	N/A	N/A	Short	Open
16F630/631/636/639/676/677/684/685/687/688/689	Ext	N/A	N/A	N/A	N/A
16F648/648A	ZIF SKT2	N/A	N/A	Short	Open
16F716	ZIF SKT2	N/A	N/A	Short	Open
16F73/737/74/76/77	ZIF SKT2	N/A	N/A	Open	Short
16F818/819	ZIF SKT2	N/A	N/A	Short	Open
16F84/84A/87/88	ZIF SKT2	N/A	N/A	Short	Open
16F870/871/872	ZIF SKT2	N/A	N/A	Open	Short
16F873/873A/874/874A/876/876A/877/877A	ZIF SKT2	N/A	N/A	Open	Short
16F913/914/916/917	ZIF SKT2	N/A	N/A	Open	Short
18F2220/2320/4220/4320	ZIF SKT2	N/A	N/A	Open	Short
18F2331/2431/4331/4431	ZIF SKT2	N/A	N/A	Open	Short
18F2420/2520/4420/4520	ZIF SKT2	N/A	N/A	Open	Short
18F2450/4450	ZIF SKT2	N/A	N/A	Open	Short
18F2455/2550/4455/4550	ZIF SKT2	N/A	N/A	Open	Short
18F2480/2580/4480/4580	ZIF SKT2	N/A	N/A	Open	Short
18F2525/26204525/4620	ZIF SKT2	N/A	N/A	Open	Short
18F2439/2539/4439/4539	ZIF SKT2	N/A	N/A	Open	Short
18F242/252/442/452/	ZIF SKT2	N/A	N/A	Open	Short
18F2585/4585/2680/4680	ZIF SKT2	N/A	N/A	Open	Short
18F248/258/448/458	ZIF SKT2	N/A	N/A	Open	Short
18F2682/2685/4682/4685	ZIF SKT2	N/A	N/A	Open	Short
dsPIC30F2010	ZIF SKT1	Open	Short	N/A	N/A
dsPIC30F2011/3012	ZIF SKT1	Short	Open	N/A	N/A
dsPIC30F2012/3013	ZIF SKT1	Open	Short	N/A	N/A
dsPIC30F3010	ZIF SKT1	Open	Short	N/A	N/A
dsPIC30F3011	ZIF SKT1	Short	Open	N/A	N/A
dsPIC30F3014/4013	ZIF SKT1	Short	Open	N/A	N/A
dsPIC30F4011	ZIF SKT1	Short	Open	N/A	N/A
dsPIC30F4012	ZIF SKT1	Open	Short	N/A	N/A

Ext = use an external programming header or the adaptor board.

at **http://freenet-homepage.de/dl4yhf/winpic/winpic_faq.htm** as a first resort if you are experiencing problems.

Because WinPIC tries to switch the programming lines in real time and because Windows is a multi-tasking operating system, timing problems could arise. For this reason, it is prudent to use the "slow mode" option in the "Interface" tab if you suspect there may be timing problems. **SC**

WinPIC software for the Low-Cost dsPIC & PIC Programmer

Serial PIC programming software used for the Low-Cost dsPIC & PIC Programmer.

We did not develop this software but are making it available for the convenience of readers.

Note that this software is no longer being developed. It seems to work on Windows 7 but it's possible that some features don't work on newer versions of Windows.

Users who have no success with WinPIC may want to try WxPIC (http://wxpic.free.fr/).

Cat No SC1895. Price: Free

Files:

- WinPicSetup.exe (WinPIC installer; 786.8KB)
- readme.txt (WinPIC "Read Me" file; 33.6KB)

Related to: Project: Low-Cost dsPIC/PIC Programmer (May 2008 [print issue] [online issue])

Other item relevant to this project:

- dsPIC/PIC Programmer PCB patterns (PDF download) [07105081/2] (Free) Add to trolley