A D.I.Y. PIC PROGRAMMER

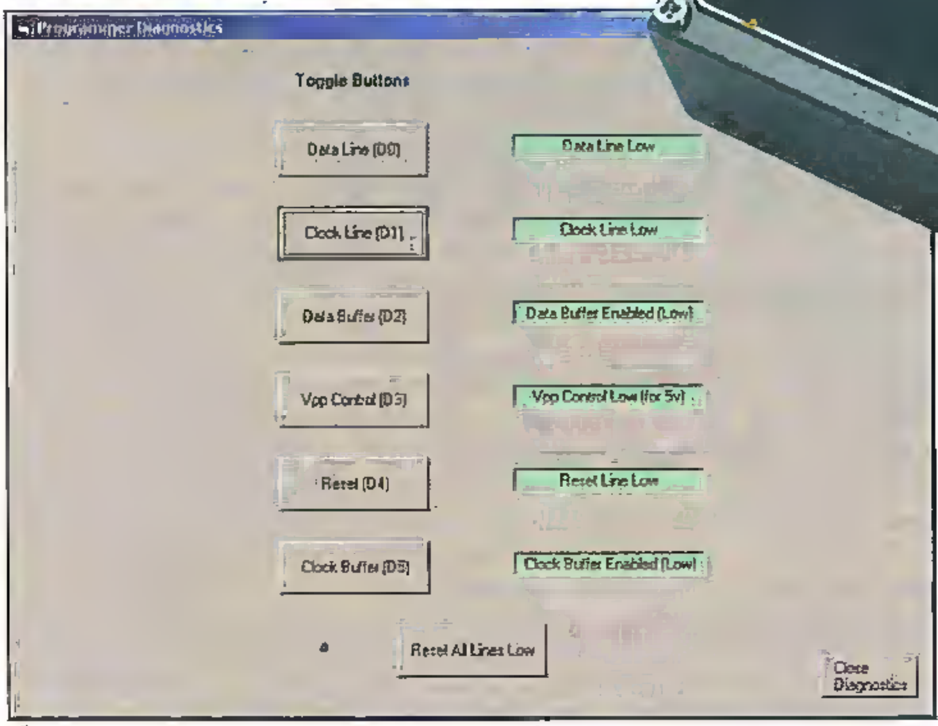
PART 1

by Malcolm Peill

Have you ever had the idea of a project that could automate some task around the home or your hobby that could easily be performed with some control logic. However, the thought of designing a circuit that would involve many logic gates, analogue input operational amplifiers and all their associated components was just too involved. Even having accomplished this, the project would be dedicated to one particular task and be very inflexible in trying to add any design improvements once these were found necessary. Of course your desktop computer could do the job, but leaving your 1Ghz Pentium pc powered on 24 hours a day just to turn on the sprinkler system when the plants need watering seems a bit of an overkill.

The solution is to make use of a very useful yet under-appreciated device, the MicroController. These single chip devices are in effect a small self-contained computer system, and are used extensively in many commercial products requiring a small amount of processing power in a limited space.

These devices have come a long way since the days of the first microprocessors. Microchip are a major manufacturer of these MicroController's, and offer an extensive range of devices with part numbers prefixed with the letters 'PIC'. They do have those same functions as microprocessors, but also include flash memory to store the program, some RAM to manipulate variables, digital input and output ports, timers, A/D converters, serial I/O and others depending on the particular device selected.

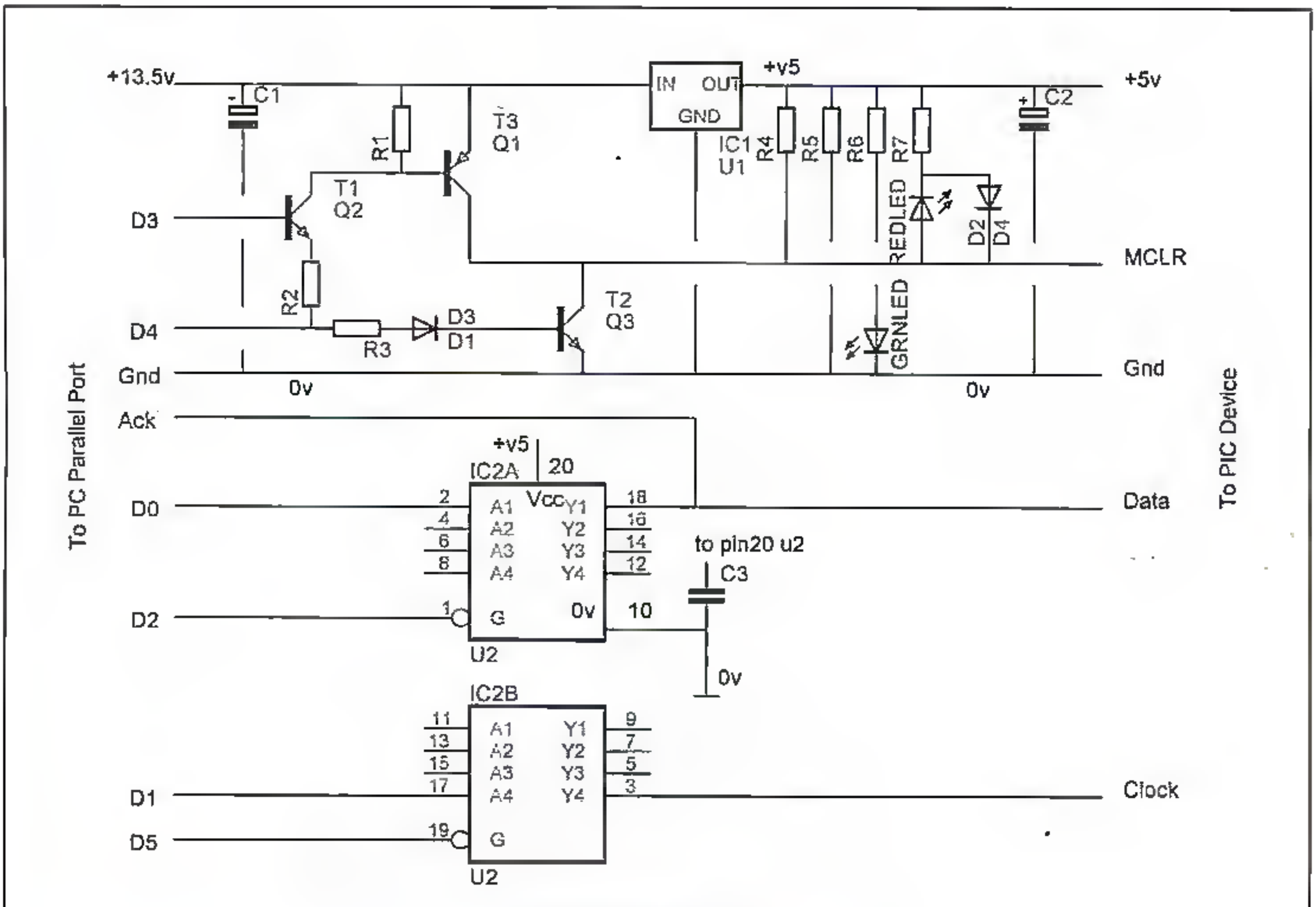


The drawback with these devices being used by the electronics enthusiast has been the relatively expensive equipment necessary to program them, and hence the reason for this project. Initially, we will go through the basics of what is needed to be able to program these PIC's, along with a design based on that shown in Microchip's application notes freely available from their web site www.microchip.com.

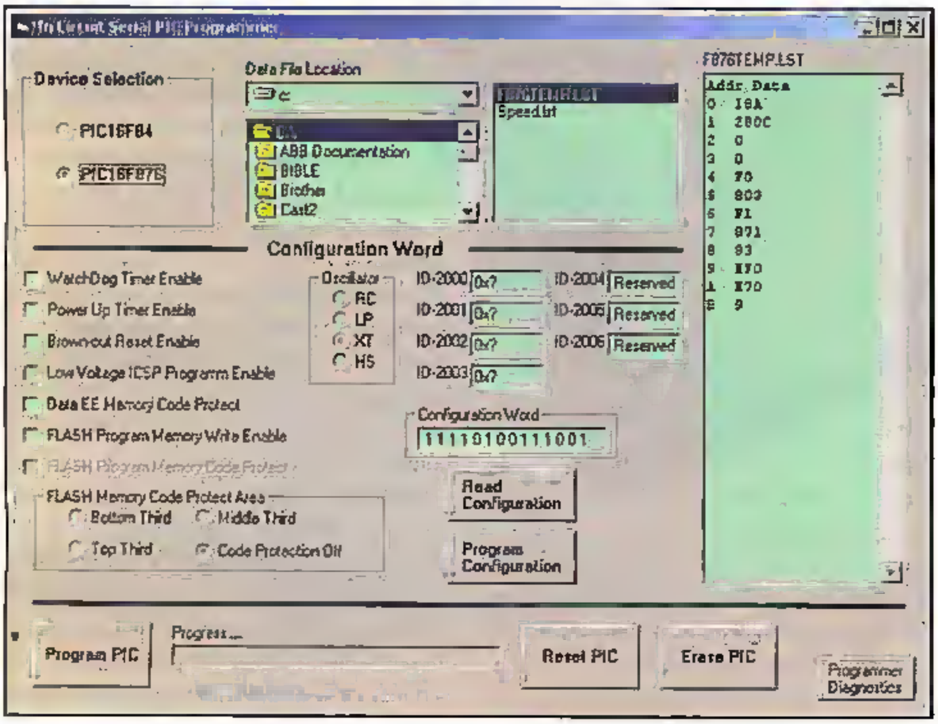
PIC Programming Requirements

There are a large number of variants in the PIC family, and this project has concentrated on two particular devices, the PIC16F84 and the PIC16F876. The '84 is relatively new, and includes digital input and output ports along with an on-chip timer, plus other useful functions. For those more adventurous who need additional I/O, the '876 also includes serial I/O and A/D converters, along with a higher number of timers and digital input /output lines.

Both of the PIC's can be programmed using the same ICSP (in-circuit serial programmer). This allows a project to be designed and built, and the PIC then reprogrammed as many times as necessary to



PIC Programmer Circuit



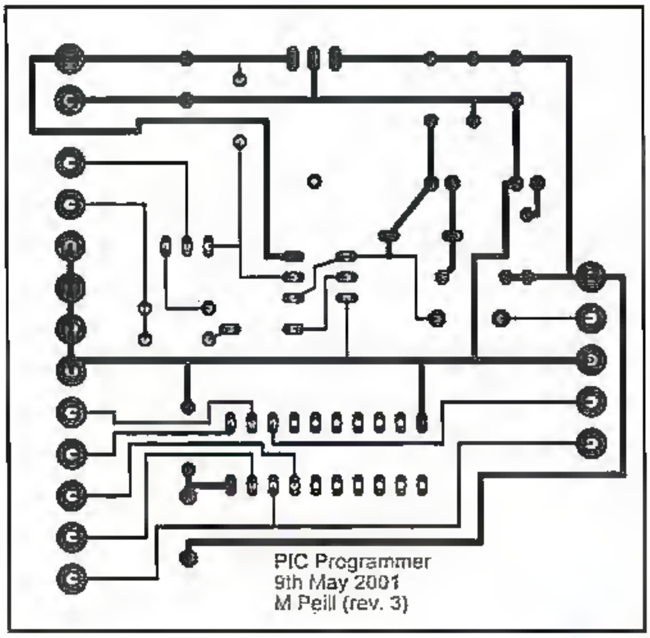
fix bugs and add enhancements to the software without having to dismantle or disturb the hardware.

There are several steps necessary to get an application running in your final project.

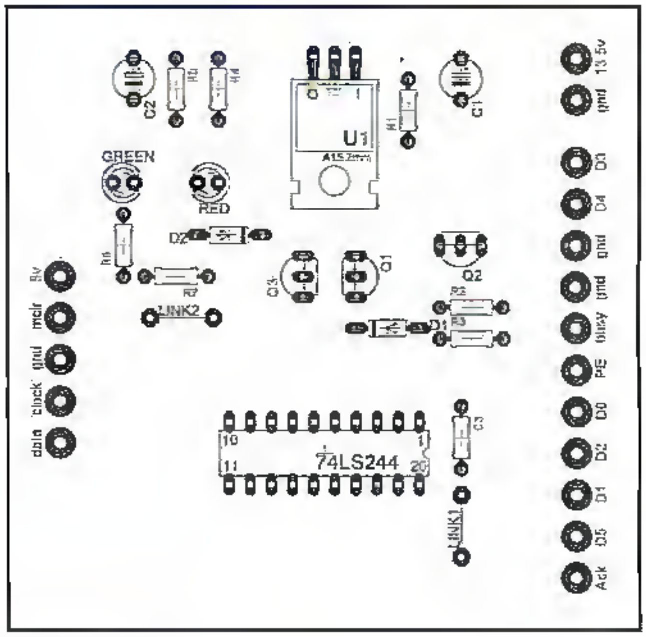
- First, write the assembler level program that will control your project (and remember it can be modified later if mistakes are made, so no devices have to be replaced).
- Secondly, the program currently in assembler mnemonics has to be converted into machine code.
- Thirdly that machine code has to be transferred to the PIC.

Thankfully, Microchip has made the first two steps relatively painless in providing tools to do the job. Again on their web site they have, freely available, the software application into which you write your code in mnemonics. This file is then saved and imported into a second application that assembles the program into machine level code. This produces a number of files, one being the listing file. Once this is done, this final code from the listing file is transferred to the PIC using the programmer detailed in this article that is based on the circuit given in Microchip's application note, using software written by the author available on CD (see details in the Parts List section).

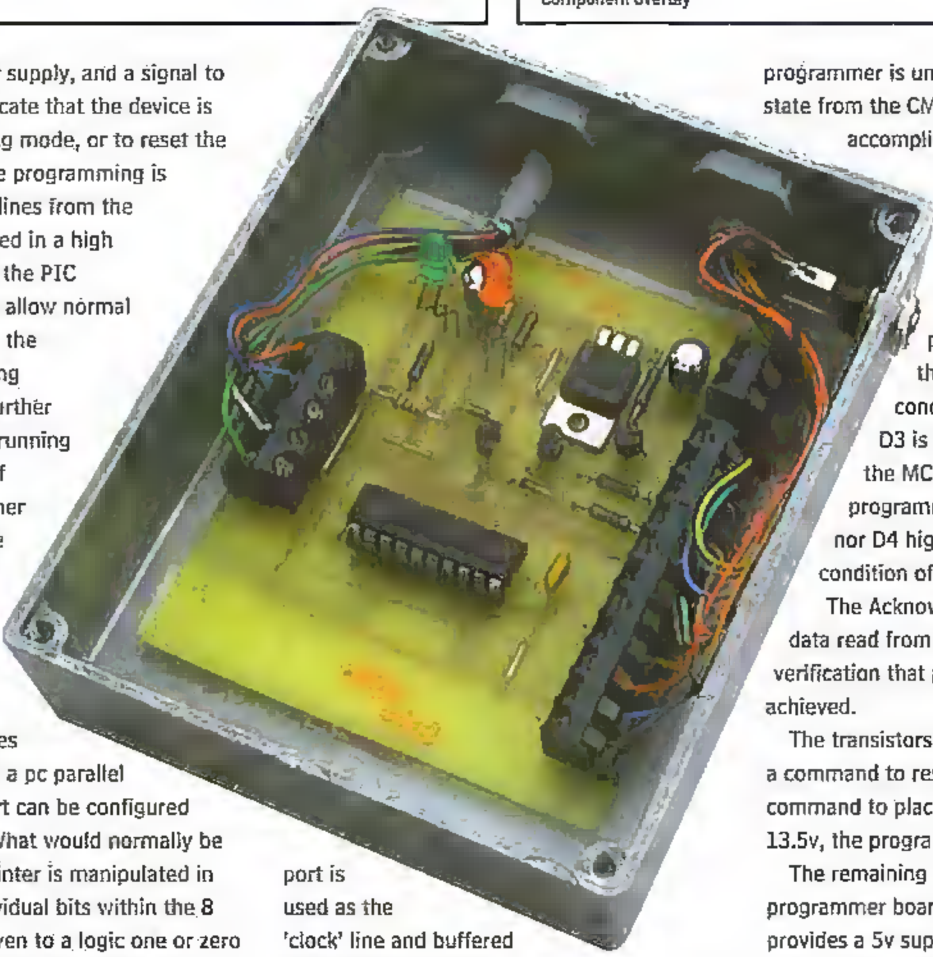
The PIC itself is programmed by a serial data stream into one of its digital I/O port lines, D7. Associated with the data is a serial clock used to gate the data into the device, and this is fed into a second digital I/O port line, D6. The only other connection necessary



PCB Track



Component overlay



to the PIC is a power supply, and a signal to the MCLR line to indicate that the device is entering programming mode, or to reset the PIC as required. Once programming is complete, the driver lines from the programmer are placed in a high impedance state and the PIC momentarily reset to allow normal operation of the PIC, the programmer remaining connected to allow further modifications to the running code as necessary. Of course the programmer can be removed once all changes are complete.

Theory of Operation

The programmer takes its drive signals from a pc parallel port. The parallel port can be configured as a standard port. What would normally be data directed to a printer is manipulated in such a way that individual bits within the 8 bit wide byte are driven to a logic one or zero as required in order to output data to the PIC. Data from the PIC via the programmer is fed into what would normally be a status line from the printer, but again can be interrogated to read back the appropriate information.

The circuit is quite straightforward. Data bit D0 from the parallel port is used as the 'data' line and is simply buffered through a logic gate before being sent to the PIC 'programming data' input line on its RB7 pin. In the same way, bit D1 from the parallel port is used as the 'clock' line and buffered on its way to the PIC 'programming clock' input line on its RB6 pin. Data bits D2 and D5 are toggled by software running within the pc to switch off the output from the data and address buffers respectively, placing them in a high impedance state. This allows the same pins on the PIC to operate as standard digital input or output lines as required by your project. Of course if these lines are used for other purposes, care must be taken not to load the pins in such a way that the programmer is unable to drive their logic state from the CMOS buffer. This can be accomplished with simple pluggable links on the project board, or rather more thought given to the design of any interface circuitry used.

Data bit D4 is used to place the MCLR line at 0v, the master clear or reset condition of the PIC. Data bit D3 is used to impose 13.5v onto the MCLR line, placing the PIC in programming mode. With neither D3 nor D4 high, the normal operating condition of MCLR is set at 5v.

The Acknowledge bit is used to feed data read from the PIC back to the p.c. for verification that programming has been achieved.

The transistors on board are used to ensure a command to reset the PIC overrides a command to place MCLR at either 5v or 13.5v, the programming mode.

The remaining integrated circuit on the programmer board is a 5v regulator chip. This provides a 5v supply to the tri-state buffer on board, and also power to the project circuit that you are attempting to program. There are two L.E.D.s on board, one indicates that power is applied, and the second indicates that the unit is in programming mode. An external 13.5v power source is required that is used to feed the voltage regulator and also provides the programming voltage necessary to program the 16F84 PIC.

The second type of PIC, the 16F876 mentioned earlier in the article, actually has the option of being programmed at low voltage without the use of the 13.5v line. This provides a means of simplifying the programmer to some extent and eliminates the need for the external supply. However, in order that both types of PIC could be programmed, the supply remains necessary.

Construction

Building the programmer is relatively straightforward. The cable used to interface with the p.c. parallel port only uses 11 of the I/O lines. This can either be built from component parts, or as an easy option you could utilise a spare printer cable, cutting off the Centronics connector and stripping back the appropriate wires as required.

25way D Type Parallel Port Pin No.	Signal Name
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
10	Acknowledge
11	Busy
12	PE
18	Ground
25	Ground

Table 1.

On the prototype built and shown in the photos, terminal strips were used to wire cables to the circuit board. To save a little cost these could be eliminated and the cables wired direct to the p.c.b. Again, a d.c. power connector was used to feed the 13.5v external supply to the unit. This could be dispensed with and flying leads used to connect to the p.s.u.

For the output cable that connects to the PIC project board, a terminal strip was used that can plug directly onto terminal posts. These posts take up little room on the PIC board and are very useful as a means of reconnecting the programmer at a later date should you want to reprogram the PIC without disturbing your project in any way. The connector blocks are available from RS Components, part number 426-159, with mating terminal posts 426-165.

It has been left to the constructor to select a suitable box to house the unit, only requiring sufficient holes to feed the input and output cables and power supply, together with 2 holes for the indicator LED's.

The application note available from Microchip goes into great detail on how to manipulate the lines to program a PIC. It includes flow charts that would allow you to write appropriate software to use with their programmer if you are that way inclined. For those looking for an easier life, the author has written a software application that can be used directly with this circuit to allow you be program your PIC project with the minimum of effort (see Parts List).

Testing the Unit

If you have elected to use the application software available here, it has the facility not only to program the PIC, but to enter a diagnostic mode where the operation of the programmer can be checked prior to connecting to your first PIC project.

Firstly install the software and plug the programmer into the parallel port of the p.c. Once you run the software it has the ability to manipulate individual lines from the p.c. parallel port and check that output lines switch state as expected. To enter this mode, click on the 'programmer diagnostic' button. You can then click on appropriate buttons to toggle each line as required and observe the result using a scope or digital multimeter.

The result of switching each line is given in table 2.

Where to Next

If all is well, you now have a functional PIC Programmer, and you may be happy to go away and start developing your complex control systems. With a little surfing around the Microchip web site (www.microchip.com) you will find the tools and documents necessary to create source code and program the PIC as required. On the other hand, you may just be wondering what on earth you need to do next. In the second part of this series we will go through the basics of what a PIC actually needs in terms of hardware and software in order to get the device to do something useful. We will discuss the configuration requirements and the basic instruction set common to most of the PIC family of devices, and then go through the actions necessary to write your very first PIC program.

Start with all lines set to logic 0, or low.

Action	Result
D0 low, D2 low	Data low
D0 high, D2 low	Data high
D0 low, D2 high	Data high impedance
D0 high, D2 high	Data high impedance
D1 low, D5 low	Clock low
D1 high, D5 low	Clock high
D1 low, D5 high	Clock high impedance
D1 high, D5 high	Clock high impedance
D3 low, D4 low	MCLR high (+5v)
D3 low, D4 high	MCLR low
D3 high, D4 low	MCLR programming (+13.5v)
D3 high, D4 high	MCLR low

Table 2.

Parts List

Resistors all resistors 5%, ? watt

R1, R2, R3, R4	2k
R5, R6	750R
R7	1k2
Capacitors	
C1	22uf, 35v electrolytic
C2	68uf, 10v electrolytic
C3	0.047uf, 100v ceramic
Semiconductors	
Q1	2N3906
Q2, Q3	2N3904
U1	78M05CT
U2	74LS244
Miscellaneous	
LED	red
LED	green
D1, D2	1N4148

Terminal Blocks and cable as required.

The printed circuit board is available from the author at a cost of £10.00.

The programmer software is available on CD at a cost of £15.00.

Post and packing for either or both of the above is £1.00. Allow 28 days delivery.

Please send cheques (payable to Mr M Peill) to:

Mr M Peill, 65 High Rigg, Brigham, Cockermouth, Cumbria, CA13 0TA.
e-mail: malcolm@peill.org.uk