

Copyright © 2008, Wimborne Publishing Ltd
(Sequoia House, 398a Ringwood Road, Ferndown, Dorset BH22 9AU, UK)
and TechBites Interactive Inc.,
(PO Box 857, Madison, Alabama 35758, USA)

All rights reserved.

The materials and works contained within EPE Online — which are made available by Wimborne Publishing Ltd and TechBites Interactive Inc — are copyrighted.

TechBites Interactive Inc and Wimborne Publishing Ltd have used their best efforts in preparing these materials and works. However, TechBites Interactive Inc and Wimborne Publishing Ltd make no warranties of any kind, expressed or implied, with regard to the documentation or data contained herein, and specifically disclaim, without limitation, any implied warranties of merchantability and fitness for a particular purpose.

Because of possible variances in the quality and condition of materials and workmanship used by readers, EPE Online, its publishers and agents disclaim any responsibility for the safe and proper functioning of reader-constructed projects based on or from information published in these materials and works.

In no event shall TechBites Interactive Inc or Wimborne Publishing Ltd be responsible or liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or any other damages in connection with or arising out of furnishing, performance, or use of these materials and works.

READERS' TECHNICAL ENQUIRIES

We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We regret that we cannot provide data or answer queries on articles or projects that are more than five years' old. We are not able to answer technical queries on the phone.

PROJECTS AND CIRCUITS

All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it. A number of projects and circuits published in EPE employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD adaptor.

COMPONENT SUPPLIES

We do not supply electronic components or kits for building the projects featured; these can be supplied by advertisers in our publication Practical Everyday Electronics. Our web site is located at www.epemag.com

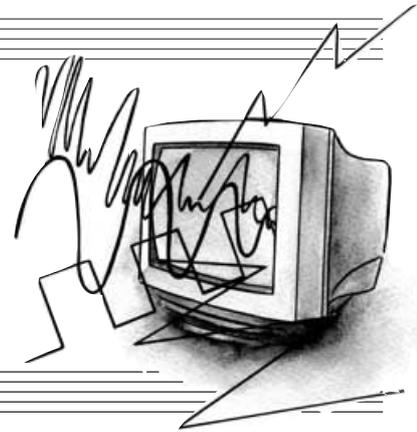
We advise readers to check that all parts are still available before commencing any project.



To order your copy for only \$18.95 for 12 issues go to www.epemag.com

PIC DUAL-CHANNEL VIRTUAL SCOPE

JOHN BECKER



A virtual oscilloscope for monitoring audio frequency waveforms via your computer.

MID-RANGE is perhaps a reasonable opening description of this design. At one end of the scale of *EPE* virtual oscilloscope designs is the simple Analogue Input Waveform Display computer interface used with the *Teach-In 2000* series. At the other end is the sophisticated and versatile *EPE Virtual Scope* of Jan/Feb '98.

With the *TI* interface, analogue waveforms from a single source are digitised under elementary computer control and displayed on screen. The sampling rate depends heavily on the rate at which the computer operates. A maximum signal frequency of perhaps 1kHz can just be displayed using a 120MHz computer.

The *TI* unit's simplicity of construction and use make it an ideal "eye-opener" for electronic novices, enabling them to gain an insight into what happens with electronic circuits and waveforms.

The *V-Scope*, on the other hand, uses a very complex set of electronic circuits to convert two analogue channels simultaneously for computer display at frequency rates up to several megahertz. It uses a mixture of QuickBASIC routines which access high-speed machine code assemblies. Operation is principally under mouse control. Its circuit complexity, however, makes it unsuitable for building by inexperienced constructors.

PIC V-SCOPE

Sitting between these two designs is the *PIC V-Scope* described here. It provides waveform display of two signals simultaneously at rates much higher than the *TI* design offers, although lower than *V-Scope* can handle. It is considerably easier to construct than *V-Scope*. With care, even less-experienced hobbyists should stand an excellent chance of constructing it successfully.

In order to use the design, however, you need a PC-compatible computer capable of running QBASIC or QuickBASIC and for it to "read" mouse controls via those programming dialects. As discussed later, the controlling program allows you to check on both points before purchasing any components. The author has run the prototype under Windows 3.1, 95 and 98.

Many of the functions offered by this design are closely similar to those offered by *V-Scope*. Indeed, the controlling software is a cut-down version of that design.

CIRCUIT DIAGRAM

The complete circuit diagram for the *PIC V-Scope* is shown in Fig.1. Both input channels are identical and are formed around op.amps IC1a and IC1b.

Taking just Channel 1, the signal is brought into socket SK1 and to switch S1, at which a choice of gain is offered, $\times 1$

(unity) or $\times 10$, as set by resistors R1 and R2 in relation to the value of R3. Other gains may be preset by changing the values of the resistors during assembly.

From R1/R2, the signal is routed to a.c. coupling capacitor C1, which can be bypassed via switch S2 for d.c. operation. The signal is then fed to the inverting input of IC1a, from where it is output to the PIC microcontroller, IC2, a PIC16F877 device.

Mid-rail bias (2.5V) for both op.amps is set by the potential divider formed across R9 and R10, with smoothing provided by capacitor C3. This bias must be taken into account when monitoring d.c. signals.

The PIC16F877 microcontroller, as discussed in several previous *EPE* issues, has eight inputs which can be used for analogue-to-digital conversion. In this design, only inputs RA0 and RA1 are used. All other active PIC port pins are used in conventional digital input or output mode.

The basic role of the PIC is to perform the A-to-D conversion and output the result to the computer via socket SK4, either immediately on conversion, or after storing it temporarily in memory chip IC3.

PORTD is used for the memory data routing, while PORTC outputs the data to the computer. As is frequently the case with *EPE* designs, the computer's parallel printer port handshake lines are used for the data output. Read/write control of IC3 is via the RA4/OE and RA5/WE connection pairs.

Communication from the computer to the PIC is via computer data lines DA0 to DA3, also connected through socket SK4.

SPECIFICATIONS . . .

- PIC microcontroller performs analogue-to-digital conversion, buffer memory storage/recall and output to PC-compatible computer for display
- Computer control of all PIC program modes
- Selectable dual channel or either channel individually
- Two signal display modes, analogue and digital
- Waveform gain switchable for $\times 1$ or $\times 10$
- Input impedance 10k Ω and 100k Ω
- Maximum input signal level before display clipping 5V pk-pk, or 50V if $\times 10$ scope probe used
- Input switchable for a.c. or d.c.
- Waveform synchronisation (sync) selectable on/off by channel, with controllable trigger levels
- Waveform shift vertically for each channel
- Frequency range: nominally audio, but extending well below 1Hz and above 10kHz
- Frequency counting and waveform amplitude monitoring for each channel, selectable on/off
- Frequency count accuracy presettable from screen for fine tuning, with automatic recall of settings on start-up
- Three sampling modes: via 2Kbyte or 32Kbyte buffer memory, or immediate
- Screen grid on/off
- Waveform display hold on/off
- Waveform data output to disk, on demand or automatic, date and time stamped
- Waveform data input from disk files
- Disk files selectable by mouse from on-screen directory
- Operation via parallel printer port
- Port address selectable on-screen, with automatic recall on start-up
- Screen dump to printer, date and time stamped
- All functions mouse-selectable
- Additional keyboard control for some options
- Power supply, d.c., nominally 9V, but 7V to 15V acceptable, approx 6mA average, on-board 5V regulator

Resistors R12 to R15 bias the PIC's respective input pins to 0V when the computer is not connected.

The PIC is operated at 5MHz, as set by crystal X1, and use of the 20MHz version (PIC16F877-20) is recommended. Whilst the author has successfully "over-run" a standard PIC16F877-4 (nominally 4MHz max.) in this circuit, satisfactory results cannot be guaranteed for other assemblies.

The PIC may be programmed on-board by those who have suitable programmers (e.g. PIC Toolkit Mk2 of May/June '99). Terminal strip TB1 provides the access connections.

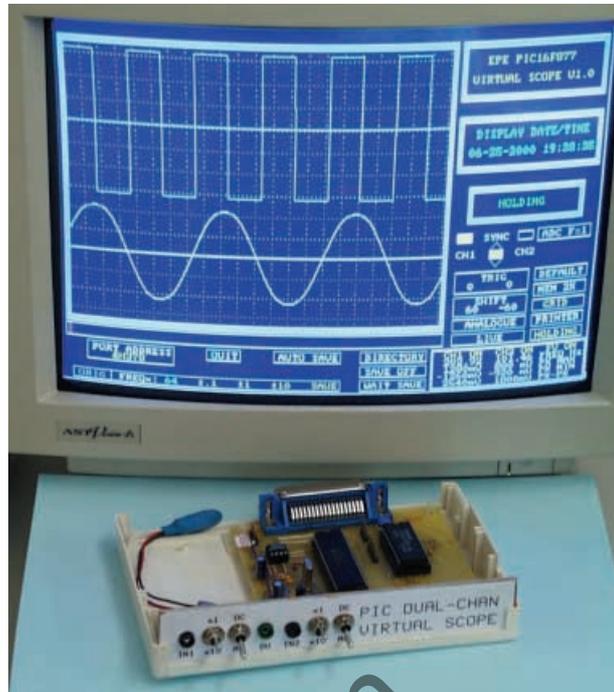
The prototype is powered at 9V d.c., with connection via a PP3 battery clip. A power supply socket could be used instead. The circuit may be powered at between 7V and 15V d.c., with an average current consumption of about 6mA.

Full software, including source code, is available on 3.5-inch disk (for which a nominal handling charge applies) or free via the EPE web site. Pre-programmed PICs are also available from an external supplier. For more details see this month's *Shoptalk* page.

PROGRAM OPERATION

Whether the PIC is instructed by the computer to sample with or without using the buffer memory, the basic process is the same. It samples the analogue input signals as fast as it can, converts them to digital format which it sooner or later outputs to the computer.

A series of handshake commands are exchanged between the PIC and computer in order to maintain the correct sequence of events. The 8-bit data is output as two 4-bit nibbles, which are re-assembled by the computer software to a single byte and plotted on screen according to the value of the signal voltage. The entire sequence of data input and plotting is performed by a set of machine code routines.



This simple but powerful tool allows two signal waveforms to be viewed on a PIC-compatible computer screen.

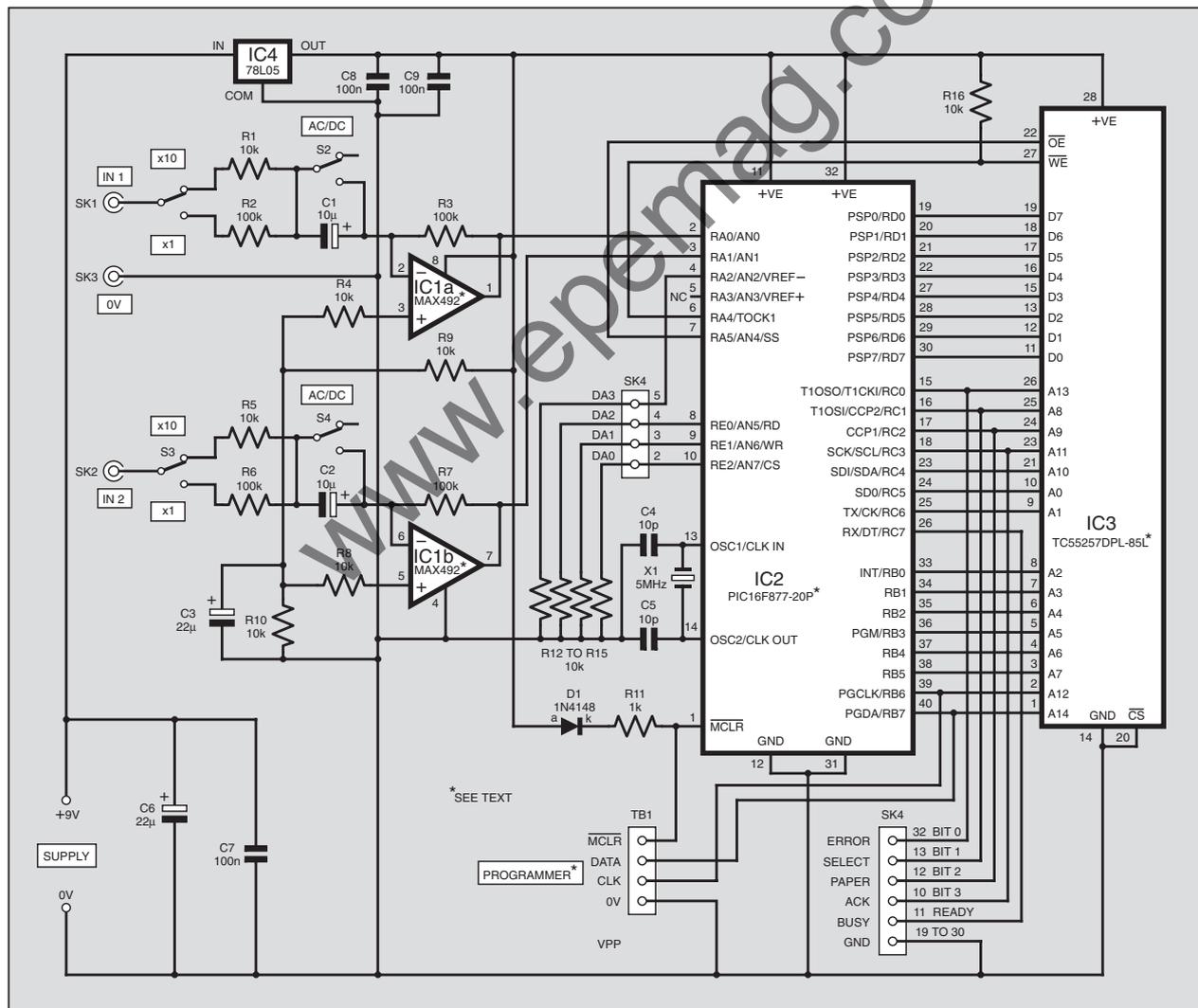


Fig.1. Complete circuit diagram for the PIC Dual-Chan Virtual Scope.

During this process, the computer software assesses the data for display synchronisation, in which repeating signal traces commence on screen at the same relative waveform amplitudes. The sync trigger thresholds can be set under mouse control, both for amplitude levels, and whether positive or negative-going triggering is required. Sync control may be turned on or off via the mouse.

Data is also analysed for signal frequency and amplitude, with the results output to the screen at the end of each waveform traversing the display area. Frequency is quoted in Hertz (Hz). Amplitude is quoted in three values, maximum and minimum peaks, and the absolute difference between the two (peak-to-peak), in volts. This analysis sequence may also be turned on and off via the mouse.

The PIC microcontroller can be instructed to sample either two channels in parallel (the second channel sampled a few microseconds after the first), or either channel individually.

MEMORY

There are two basic sampling modes, immediate or buffered. In the immediate mode the PIC samples and converts the analogue data to digital and immediately outputs it to the computer. The process, however, is slowed by the fact that handshaking and sample taking are interlinked, the PIC and the computer waiting for each other's response before the next sample is taken.

This method is more suited to sampling lower frequency waveforms, of a few tens of hertz or less.

The fastest method is achieved by using the memory buffer, IC3. Samples are taken as in the immediate mode but immediately stored in IC3. Handshaking is not required and the process is entirely under PIC control.

Each sample is output to IC3 via PORTD and with IC3's \overline{OE} (output enable) pin held high, the \overline{WE} (write enable) pin is toggled low then high again, an action which causes the data to be stored.

The memory's data address is controlled by PORTB and PORTC, providing a 15-bit (32768 bytes) address range. After each sample is written to IC3, the address count is updated, ready for the next sample.

TWIN BUFFERS

There are two memory buffer modes, 32Kbyte and 2Kbyte, respectively allowing 32768 and 2048 samples to be stored. In dual-channel mode, alternate addresses are used for each channel, with a maximum sample quantity of 16K and 1K per channel. In single channel mode the full allocated block is used by that channel.

When the memory count reaches 2K or 32K, as appropriate, the PIC signals to the computer that the data is now ready to be transferred.

A slightly cut-down version of the sampling routine for one channel is shown in Listing 1. In the actual program, another routine is also called, to determine whether or not the mode needs to be changed.

To read data back from the memory in the same order, the counters are reset, IC3 \overline{WE} is set high and \overline{OE} set low with the commands:

```
MOVLW %00100000
MOVWF PORTA
```

Data is then recalled from each address via PORTD, with the command **MOVF PORTD,W**. The data is output to the computer, after which the counter is updated and the next sample recalled. The data transfer uses the same handshake protocol as for the immediate mode. The routines are too complex to be listed here.

This buffered technique allows samples to be taken far more rapidly than the immediate mode. There is, though, a brief delay between each sample batch being displayed on screen, but this is usually almost unnoticeable.

The choice of sampling mode is mouse-controlled, the computer sending the mode commands to the PIC via data lines DA0 to DA3.

A summary of the sampling modes is as follows:

LISTING 1. Single channel data sampling and storage

DX0:	BSF ADCON0,GO	; start conversion for 1st sample
	PAGE1	
	CLRF TRISD	; PORTD as outputs
	PAGE0	
	MOVLW %00110000	; set WE hi (bit 4), OE hi (bit 5)
WAITAD0:	MOVWF PORTA	
	CLRF PORTB	; reset PORTB counter
	MOVWF LIMIT,W	; set limit value into PORTC
	MOVWF PORTC	
	BTFSC ADCON0,GO	; wait until conversion flag is set
WAITAD0:	GOTO WAITAD0	
	MOVF ADRESH,W	; get ADC val
	BSF ADCON0,GO	; start conversion for next sample
	MOVWF PORTD	; put it out to mem
	BCF PORTA,4	; toggle mem WE down
	BSF PORTA,4	; toggle mem WE up
	INCFSZ PORTB,F	; inc PORTB counter, is it 0?
	GOTO WAITAD0	; no, get next sample
	DECFSZ PORTC,F	; yes, dec PORTC counter, is it 0?
	GOTO WAITAD0	; no, get next sample
(end of routine)	; yes, so end of batch storage	

- Single channel (A or B): Immediate, continuous, no batch limit
- Single channel (A or B): Buffered, 2048 or 32768 samples per batch
- Dual channel (A + B): Immediate, continuous, no batch limit
- Dual channel (A + B): Buffered, 1024 or 16384 samples per channel per batch

MODE RATES

To simplify programming and, more particularly, to speed data acquisition, separate routines are used in the PIC program for each mode.

There is also a choice of the rate at which the PIC actually does the analogue-to-digital conversion. As detailed in Table 11.1 of the PIC16F87x data book, the maximum rate (T_{osc}) at which the PIC can perform its A/D conversion is dependent upon the frequency at which the PIC is operated.

COMPONENTS

Resistors

R1, R4, R5
R8 to R10
R12 to R16 10k (11 off)
R2, R3,
R6, R7 100k (4 off)
R11 1k
All 0.25W 5% carbon film or better.

See
SHOP
TALK
page

Capacitors

C1, C2 10 μ radial elect, 16V (2 off)
C3, C6 22 μ radial elect, 16V (2 off)
C4, C5 10p ceramic disc, 5mm (2 off)
C7 to C9 100n ceramic disc, 5mm (3 off)

Semiconductors

D1 1N4148 signal diode
IC1 MAX492 dual op.amp, rail-to-rail (see text)
IC2 PIC16F877-20P microcontroller, pre-programmed (see text)
IC3 TC55257DPL-85L 32Kbyte SRAM (see text)
IC4 78L05 +5V 100mA voltage regulator

Miscellaneous

S1 to S4 min.s.p.d.t. toggle switch (4 off)
SK1 to SK3 2mm single socket (3 off) (see text)
SK4 36-way Centronics socket, right-angle, p.c.b. mounting
TB1 4-way 1mm pin-header strip (see text)
X1 5MHz crystal

Printed circuit board, available from the EPE PCB Service, code 275; plastic case, 180mm x 120mm x 40mm; 8-pin d.i.l. socket; 28-pin d.i.l. socket; 40-pin d.i.l. socket; PP3 battery clip (see text); p.c.b. supports, self-adhesive, low profile (4 off); connecting wire; solder, etc.

Approx. Cost
Guidance Only

£33
excluding case

For a 5MHz clock, as used in this design, register **ADCON0** has a maximum recommended **ADCS1:ADCS0** (bits 7 and 6) value of binary 01 (**8Tosc**). However, on experimentation, it was found that the **2Tosc** rate (binary 00) worked perfectly satisfactorily, providing a four-times increase in A/D conversion speed.

Because this higher operating rate cannot be guaranteed for all devices, the PIC software has been given the option for either rate to be selected from the computer (see later). **ADCON0** bit 7 is held low and bit 6 is toggled between high and low to alternate between **8Tosc** and **2Tosc** (respectively).

DELAY CALLS

In the author's *PIC16F87x Mini Tutorial* (Oct '99), an example of A/D conversion was illustrated in which a delay was imposed during multiple sampling routines, in order to allow time for the PIC to fully acquire each analogue value before the conversion is performed.

The PIC16F87x data book discusses this timing subject, but presents a complicated formula for establishing the optimum value. In the *Tutorial* example, delays of 256 clock cycles were given.

Further experience with the PIC16F87x in several other applications, however, shows that this length of delay can be considerably reduced. In the *PIC V-Scope* routine for single channel sampling, the delay caused simply by the number of commands in the routine is all that is required.

For dual channel sampling, though, an additional delay routine is called (**DELAYB**) which, with the **CALL** command itself, introduces a delay of nine clock cycles in addition to that caused by the sampling routine itself.

Without this additional delay, it was found that small amounts of each channel's sampling were superimposed on the other channel's sampling. The reason being that the channels are sampled alternately throughout the full cycle and the PIC uses the same internal conversion circuit for all sampling. The effect was most noticeable when one channel was sampling a 5V logic waveform while the other sampled a lower amplitude analogue waveform.

The sample source inputs are alternated between **RA0** and **RA1** by toggling **ADCON0** bit 3. The **DELAYB** routine is called immediately following the change of this bit, and eliminates the effect. In fact a delay of only six clock cycles also cured it, but another three were included to allow for tolerance variations between devices.

BEFORE YOU BUILD

Through feedback from readers it is apparent that not all PC-compatible computer systems and their software are capable of running a QuickBASIC or QBasic program with the type of machine code routines that are an integral part of this design.

It is strongly recommended that before you purchase any components for this circuit, you obtain and run its program as a test of your computer's suitability.

First though, check that your computer has either QuickBASIC or QBasic already

installed. If it has not, the chances are that you have QBasic available on your Windows 95/98 CD ROM. Recent correspondence in *Readout* says that it is to be found in one of the following directories:

Win95: \other\oldmsdos
Win98: \tools\oldmsdos

There are two files, **QBASIC.EXE** and **QBASIC.HLP**, being the program and its Help file. Make a directory (folder) called **QBASIC** on your hard drive and copy both files into it. To run the program, just type **QBASIC**.

Your *PIC V-Scope* files must all be copied into the same directory that holds QBasic or QuickBASIC. From either version of Basic, then load and run program **PVSCOPE.BAS** (which may also be called via the **PVSCOPE.BAT** file).

It is important to note that *QuickBASIC* must be loaded and run with the command **QB/L**, which additionally loads a QuickBASIC library routine that allows machine code to be run with a Basic program. The *PIC V-Scope* program will crash if the **QB** library routine is not loaded. *QBasic* has this routine automatically included and does not require (or accept) the **QB/L** command.

On entry, the *PIC V-Scope* screen should appear similar to that in the photographs, except that it will say that it is waiting for a response from the circuit, and will be without signal trace lines. Ignore this statement and check that the mouse cursor arrow is visible and that you can move it around the screen. Additionally check that when you left-click it on the values in the **TRIG** and **SYNC** boxes they increment, and decrement if you right-click.

Note that throughout this text the terms *left-click* and *right-click* mean pressing the left-hand and right-hand mouse buttons, respectively. If *click* is used on its own, either button will perform the required action.

Also check that when either mouse button is held down and any keyboard key (except **Q**) is pressed, the **TRIG** or **SYNC** values continue to change.

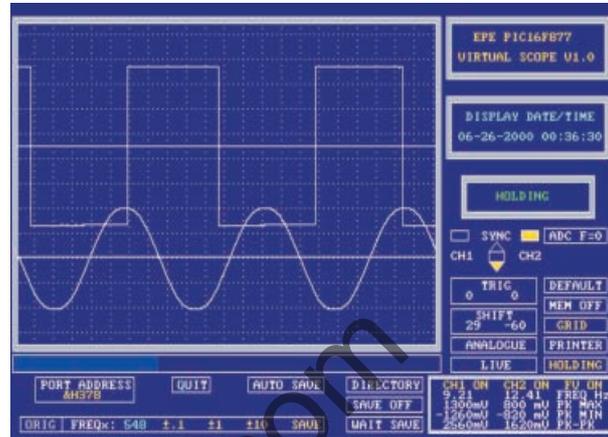
If the mouse performs these actions, your computer should be capable of running the rest of the program and controlling the *PIC V-Scope* circuit. Sadly, if it does not do these things, your computer is incapable of controlling the circuit.

QUITTING THE PROGRAM

Pressing **Q** causes a Quit (exit) from running the program. Left-clicking on the box labelled **QUIT** causes the computer to "bleep" and a screen statement appears asking if you are sure you want to quit.

Left-click if you wish to quit, or right-click if you wish to continue running the program.

If the program has initially been run from the **PVSCOPE.BAT** command, quitting will return you to the screen from which the command was originally given. If the program has been loaded and run from the Basic menu, the program will stop, telling you to "press any key", and then show the program listing within the Basic editing environment. To exit from Basic in this instance, press in turn **ALT, F, X** (the usual exit command keys).



Twin low-frequency waveforms sampled in immediate mode, with negative-going sync applied to Channel 2.

CONSTRUCTION

First a note on the op.amp (IC1) and memory (IC3). A MAX492 dual op.amp having rail-to-rail outputs was used in the prototype. It is likely that other dual rail-to-rail op.amps could be used if this Maxim device is hard to locate (see *Shoptalk* page). Alternatively, a more universal op.amp such as the LM358 or TL072 could be used, although their outputs do not swing fully between the power line voltages.

Whilst the TC55257DPL-85L 32Kbyte SRAM (static random access memory) device used for IC3 is readily available, the 85ns access time (as indicated by the 85L suffix) is faster than actually needed and the 100ns version would be acceptable.

Printed circuit board component layout and tracking details are shown in Fig.2. This board is available from the *EPE PCB Service*, code 275.

Assemble the board in order of component size, starting with the link wires, and noting that two go under the IC3 position. Sockets must be used for IC1, IC2 and IC3. Do not insert these dual-in-line (d.i.l.) devices until the correctness of the power supply regulator circuit around IC4 has been proved. The d.i.l. devices are CMOS and require the usual anti-static handling precautions, e.g. touching an earthed (grounded) item before handling them.

Note that some components are mounted directly on the switches. Sockets SK1 to SK3 are 2mm single types in the test model but other types may be substituted if preferred (e.g. BNC sockets for use with proper scope probes).

Use 1mm terminal pins for the off-board connections to the battery clip and front panel components. A 4-pin 1mm pitch pin-header was used for TB1

allowing the author to plug in *PIC Toolkit Mk2* via an existing connector (for on-board PIC programming – see later).

In the photograph of the p.c.b., two additional pin-header strips plus a preset pot are visible. These were purely for the author's use during program development (an l.c.d. was used to monitor various aspects) and are not required for the published version. Ignore the unused holes seen in the p.c.b. component layout.

The plastic case used has detachable front and rear panels and measures 180mm × 120mm × 40mm. A source part number is quoted on this month's *Shoptalk* page. In the prototype, the front panel components are positioned 12.5mm (0.5in) apart. The rear panel was omitted, allowing easy access to the computer connection socket, although a suitable slot could be cut if preferred.

If the unit is to be used with an external power supply rather than a 9V battery, a socket could be added to the front or rear panel.

FIRST CHECKS

Having thoroughly checked the complete assembly for satisfactory solder joints and component positioning, apply power and check that +5V is available at the output of regulator IC4, and at other principal points shown in the circuit diagram.

If all is well, the PIC can be programmed on-board, via the TB1 connector. The *PIC Toolkit Mk2* programmer is ideal for this. The configuration settings required by the PIC before loading it with the program itself are those shown in Table 1. Insert IC1 and IC3 after the PIC has been programmed.

Table 1. PIC16F877 configuration settings

CP1	CP0	DBG	NIL	WRT	CPD	LVP
1	1	1	1	1	1	0
BOR	CP1	CP0	POR	WDT	OS1	OS0
0	1	1	0	0	1	0

PORT REGISTER

Any PC might have any one of three possible parallel printer port register pairs set as the active input/output address. In hex, the addresses are 378/379, 278/279, 3BC/3BD. The *PIC V-Scope* program must be set to use the same address pairs as set within the computer's own system configuration.

When the *PIC V-Scope* Basic program is run, the "waiting response from circuit" message will initially be shown, but should disappear when the unit is plugged in and switched on, providing the correct port address applies. The initial port address default value is for 378/379.

If the message does not disappear, click on the PORT ADDRESS box at the bottom left of the screen. The value will change from showing &H378 to &H278 (&H being the prefix recognised by QB as meaning hex). If the message still does not disappear, click again to show &H3BC. A third click returns to &H378. The program automatically deduces the secondary address (e.g. &H379) from the primary.

If none of these addresses cause the computer to recognise that *PIC V-Scope* is

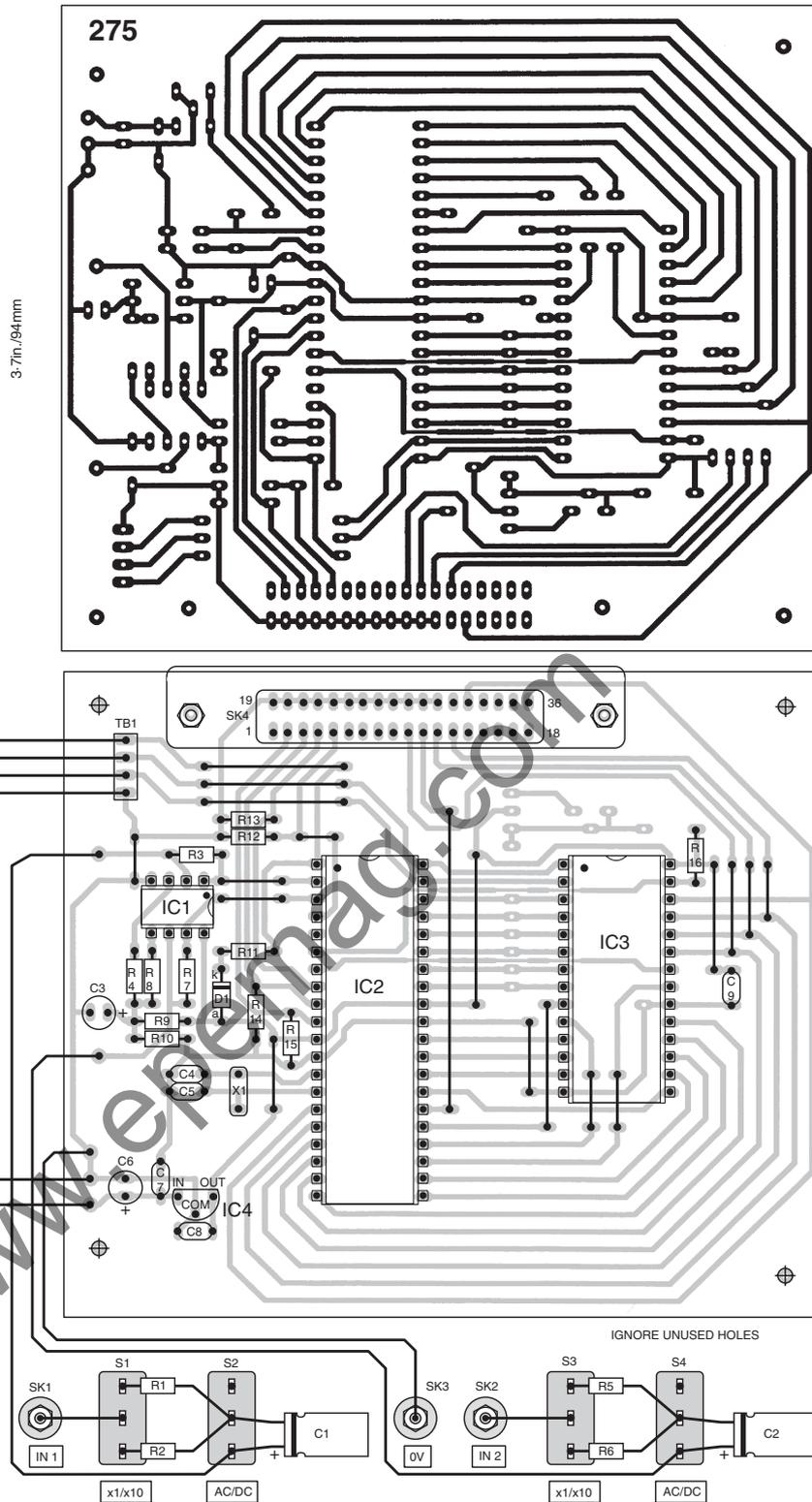


Fig.2. Printed circuit board full size copper foil master pattern and component layout. Ignore unused holes (see text).

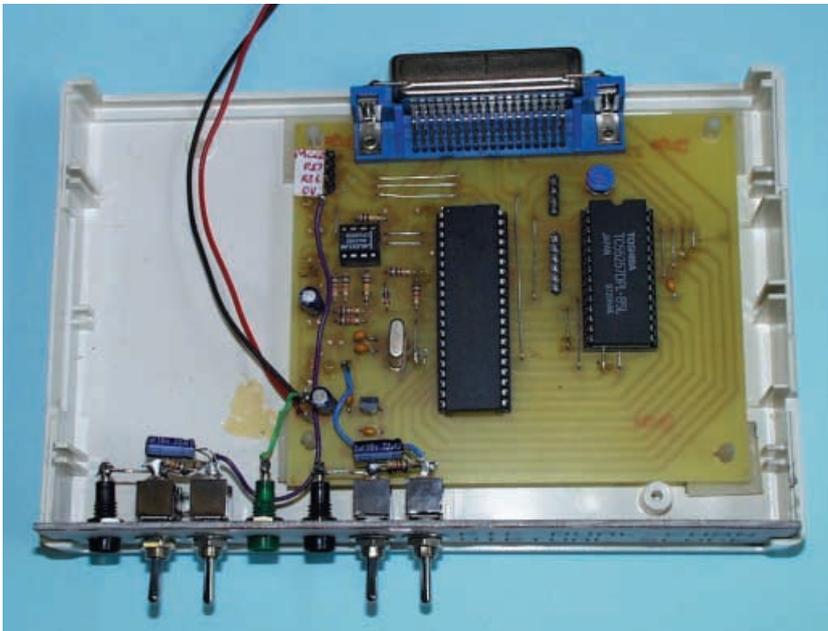
connected and powered, recheck your assembly and its connections.

When the message disappears, the display area fills with a grid and two horizontal lines. The program and unit are now successfully operational.

The selected port register address value is automatically stored on disk (file **PSCOPATH.TXT**) and is recalled when the program is next loaded and run.

USING THE SCOPE

With the *PIC V-Scope* connected and powered, on entry to the program screen the dual-channel mode is operational. Without signals being input to the circuit, two continuous horizontal lines will be seen across the display area. They are the signal trace lines for Channel 1 (upper) and Channel 2 (lower).



Details of the assembled prototype. Ignore the preset pot and extra pin-header strip seen on the p.c.b. (see text).

There are in fact two continuous lines in each position, the second indicating the 2.5V midway reference for that channel. Due to component tolerances in the circuit, there may be a slight displacement between the trace and reference lines. The latter is generated by the computer as the reference level, the other depends on the actual midway voltage as seen by the PIC's ADC.

You may also just be able to make out a third line, which is dotted. This is the synchronisation trigger line, showing the level at which sync and frequency values are referenced.

Move the mouse pointer above the left-hand TRIG value (0 at present) and repeatedly left-click on this value. The dotted trigger line will be redrawn slightly higher up the screen on each click, immediately recommencing from the left even if the trace has not reached the right of the display. Right-clicking will lower the line position. The TRIG value in the box will increment or decrement accordingly on each click.

Clicking on the right-hand TRIG value similarly alters Channel 2.

As said earlier, pressing any keyboard key (except Q) while a mouse button is held pressed, progressively changes the TRIG value. The redrawing of the screen data, however, waits until the key has been released.

Clicking on the SHIFT values changes the relative positions of all three lines for the selected channel.

Connect a signal generator to input 1, with an output frequency of about 1kHz, and with any reasonably uniform waveform shape. Set the input switches for x1 and AC.

Note that if *PIC V-Scope* is battery powered, there must be a connection between its common (0V) socket SK3 and the common (ground) line of the signal generator or its power supply.

Increasing the amplitude of the input waveform from zero, observe the waveform similarly increasing on the screen display. Experiment with shifting the display up and down, and with changing the input frequency.

Clicking in the column headed CH2 ON at the bottom right of the display turns off Channel 2, with the

heading changing to CH2 OFF. Channel 1 is now displayed on its own, more centrally on the screen. As the PIC is now only sampling one channel, the process is quicker and the waveform on screen is seen to be expanded horizontally – fewer cycles across its width.

Clicking on CH2 OFF restores Channel 2, with Channel 1 returning to its previous position and cycles per width.

Column CH1 ON behaves in the reverse fashion in respect of Channel 1. The program prevents both channels being switched off at once.

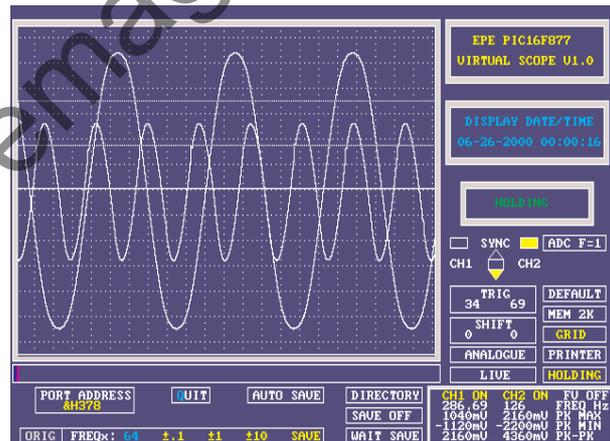
MEMORY BOX

The PIC is currently sampling with the buffer memory set for 2K bytes. A right-hand screen box confirms this, stating MEM 2K. Left-clicking on this box sets the buffer memory for 32K bytes, confirmed in the box as MEM 32K.

The waveform is now plotted at the end of this much longer sample batch, each screen-full consisting of consecutive sections of the memory. There is a brief pause when the memory has been down-loaded and the next batch is sampled.

A blue bargraph below the display area shows the progress of the sampling and display.

Left-clicking on the MEM 32K box puts the program into immediate mode, in which the buffer memory is not used, confirmed as MEM OFF. Waveforms are now much more closely spaced since the sampling and output process is slower, as stated earlier.



Twin waveforms sampled in MEM 2K mode. Note the dotted sync trigger lines, with the larger waveform negatively triggering just after its peak.

Left-clicking on MEM OFF returns the mode to MEM 2K. The MEM box can be right-clicked to select the options in the reverse order. The process is fully cyclic.

SYNC

So far you will have found that each waveform screen-full has started at different points in the waveform. Using SYNC stabilises the display so that repeating waveforms start at similar amplitude points.

Centre-right of the screen are the sync control boxes. The box to the left of the word SYNC is shown as yellow, indicating that sync is switched for Channel 1. However, the 3-box "lozenge" has its centre section in yellow, indicating that sync is turned off.



Clicking in the upper or lower triangles turns them yellow instead, indicating that sync is now active, triggering on the upwards or downwards edges, respectively. Clicking on the central box again turns sync off. Clicking in the box to the right of SYNC allocates sync to Channel 2.

The position in the waveform at which sync is triggered is changeable via the TRIG box, as described earlier. Note that if the value is set outside the amplitude range of the signal, the display will freeze because the trigger amplitude is not being found. There is a narrow "window" of values through which the waveform has to pass for the sync level to be recognised.

Sync may be turned off while the display is frozen. There might be a brief pause before the mouse button is responded to due to a time-out routine while a sync level is being sought from the input signal.

DISPLAY HOLD

Clicking on the RUNNING box (lower right) causes the display to hold once it has reached the right-most point, upon which the word HOLDING is shown. Pressing any key (except Q) steps the display onto the next batch of samples. Clicking on HOLDING restores the program to RUNNING mode.

SCREEN GRID

Clicking on GRID (lower right) toggles the display grid on and off. Each vertical grid square represents approximately 400mV. The full screen height represents about 5V. The program does not allow signal amplitudes to be "tuned" precisely. This would have required extra processing commands (and thus time) both for the PIC and the data acquisition program, slowing down the display.

For a similar reason, the horizontal grid squares are not time/frequency related. Both vertically and horizontally, the grid should be used only as a guide to the relative positions of waveform points.

WAVEFORM SAVING

Waveforms can be saved to disk (in the same current QB directory/folder on the hard drive), under automatic or manual control. For the manual saving option, click on SAVE OFF (lower centre) to reveal SAVE ON. When a full batch of samples has been received, the WAIT SAVE box changes to SAVE NOW.

Left-clicking on the SAVE NOW box causes the raw waveform data (numerical values) to be saved with a unique coded time and date file name, which is displayed at the top of the screen until the next batch is waiting to be saved.

Right-clicking on WAIT SAVE aborts the save option for the current batch.

Clicking on SAVE ON turns off the saving mode.

Clicking on AUTO SAVE causes each batch of data to be saved upon completion, again with timed and dated file names. This mode automatically sets the buffer memory

for 32K if it is currently in 2K mode. In MEM OFF mode, 32K samples are always saved. All saved files hold 32K samples.

SAMPLE RECALL

Saved sample files can be recalled via the DIRECTORY or LIVE boxes. Left-clicking on either clears the display screen to show a directory of the sample files on disk. If there are no files yet saved, the screen tells you so, reverting to the display screen once the mouse button is released.

Files are listed with their coded names and can be selected via the mouse. Moving the mouse causes a highlight bar to shift over the file names. Left-clicking on a name reveals a lower screen line which asks if you wish to load that file, showing its code name and the time and date it was saved. To accept the file either left-click or press "Y". Doing so loads it and returns to the display screen, showing the loaded waveforms.

Right-clicking or pressing "N" at this time allows you to choose another file.

Before a file name is selected, the directory can be exited by right-clicking. The highlight can move across most of the screen, but only a proper file name can be accepted, except any shown in the first line.

The first line shows the file name of the current replay waveform (if loaded). It also shows, at top left, the most recent file name selected by the mouse. This name cannot be selected from this position, the location is used by the program when reading the screen data. (Reading from the screen for directory purposes is quite complex, as a study of the program's source code will reveal.)

During replay mode, the LIVE box changes to show REPLAY. The SYNC, TRIG and SHIFT values become those at

File name = 07913713.Y00

Imagine it split as 07 9 1 37 13 .Y00

Digits 1 and 2 = day of month (7)

Digit 3 = month number in hex from 1 to C (9 = September)

Digit 4 = hour in hex-fashion - 0 to 9 followed by A to N (0 to 23), in this case 1 = 1 a.m. (post-midnight oil and toil in the workshop yet again!)

Digits 5 and 6 = minutes in decimal (37)

Digits 7 and 8 = seconds in decimal (13)

.Y00 = year 2000 (who cares about Millennium Compliancy in this instance!)

The example file name thus decodes as 7 September 2000 at 1:37.13 a.m. and would be shown as 07SEP00 1:37.13 if selected.

Saved files can only be deleted from DOS or Windows.

FREQUENCY COUNTING

While waveform batches are being input, they are analysed for signal frequency. There must be at least three waveform cycles displayed for frequency to be calculated. The total number of amplitude changes above and below the trigger threshold is counted and at the end of the batch input this value is related to the number of sample bytes and then to a conversion value depending on the channel and buffer memory modes in use.

As explained earlier, the data acquisition rate changes depending on these modes and in relation to the speed at which the PIC and the computer process data.

Since these factors will be different for individual users, an on-screen correction facility has been provided.

The long oblong box at the bottom left of the screen is the area from which the values can be changed. The word **FREQx:** is followed by the value (in blue) by which the basic frequency count result is to be multiplied. Initially, this is the default value used by the author with the prototype. To the right of this value are three increment/decrement options. Clicking on these will change the multiplying value by the same amount, left-click upwards, right-click downwards.

Frequency factor correction needs to be done for each channel mode and using different settings of the frequency generator (which should be calibrated or have a frequency count display). Table 2 shows the modes and their suggested alignment frequencies for ADC F = 0 and ADC F = 1 (see later).

For each mode, change the screen's frequency multiplying value until the frequency box at the bottom right shows close to the same frequency value(s) being input. Exact matching of the values should not be expected due to the nature of the sampling and analysis process.

During frequency alignment, a "Save when done" message is shown above the box. When you have finished alignment, click on SAVE in the box. The new factors will be stored in file **PSCOPATH.TXT** and will automatically be recalled when the program is next loaded and run.



Displaying a single square waveform via Channel 2, with Channel 1 turned off.

the time of recording. These and some other boxes become coloured red. The memory mode and save options are inoperative during replay.

To revert from replay mode to live sampling, right-click on DIRECTORY or REPLAY.

FILE NAME CODING

Saved waveform file names are coded in exactly the same fashion as used with the original *V-Scope*. The following is an example:

Table 2. Suggested channel alignment modes and frequencies.

CHAN 1	CHAN 2	MEM	FREQ (F=0)	FREQ (F=1)
On	Off	Off	200Hz	200Hz
Off	On	Off	200Hz	200Hz
On	On	Off	200Hz	200Hz
On	Off	2K	10kHz	4kHz
Off	On	2K	10kHz	4kHz
On	On	2K	2kHz	1kHz
On	Off	32K	10kHz	4kHz
Off	On	32K	10kHz	4kHz
On	On	32K	2kHz	1kHz

The word ORIG is at the left of the box. Clicking on this recalls the author's default values, which you then have the option of saving in place of your own if you wish.

Correction of the frequency values can be done on any occasion you want.

DIGITAL DISPLAY

So far the sampled data has been displayed as analogue waveforms. There are two other modes available, Digital and Lissajous.

Left-click on the ANALOGUE box (lower right). It changes to DIGITAL and a whole mass of waveforms appear on screen, up to 16. They are in two groups of eight, representing the eight bits of each channel's samples. The bits are in ascending order on the screen, bit 0 (LSB) lowest in each block, bit 7 (MSB) highest.

The digital display option was principally used in the original *V-Scope* to display 8-bit digital signals via separate input lines, but it seemed worthwhile leaving the mode in *PIC V-Scope* even though the input channels are analogue.

Frequency and voltage analysis are turned off in this mode, as is their display box at bottom right. Channels may still be turned on or off in this mode.

LISSAJOUS

Left-clicking on DIGITAL moves the display on to LISSAJOUS mode. Lissajous is that mode in which one channel provides vertical deflection while the other controls the horizontal. In a real scope it can be useful in visual comparison of frequency and phase between two signals.

It was written for the original *V-Scope* in which the sampling repetition rates are fast. Frankly, though, it has no realistic use in *PIC V-Scope*, other than as a visual curiosity. However, it seemed a pity to delete it and it has been retained purely for that reason.

Left-clicking on the display mode box (now showing as LISSAJOUS), returns the display to analogue mode. Right-clicking reverses the mode display order of selection.

PRINTER OUTPUT

The use of the PRINTER box (lower right) is the same as with the original *V-Scope*, and requires the printer port cable to be temporarily removed from the unit.

When the screen is HOLDING, its display can be output to some types of

printer by clicking on PRINTER.

The machine-code routine was originally written for an Epson ESC/P2 compatible 24-pin dot-matrix printer but has been found to also work with an Epson stylus Photo 750 ink-jet printer.

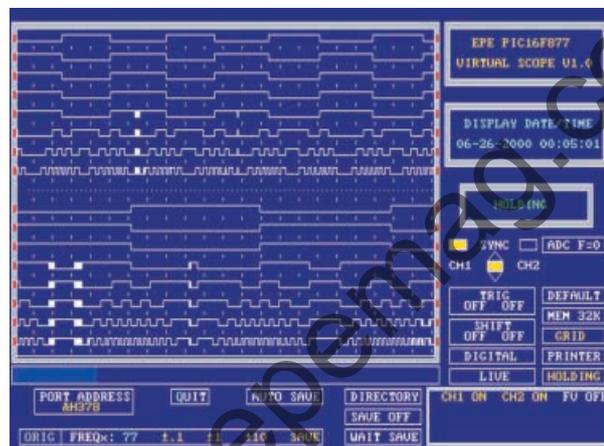
It seems probable that other similar Epson printers can also be used. Nothing

is known about laser printers in this context – reader feedback to the Editorial office will be appreciated, for possible inclusion in *Readout*.

The entire screen display is printed (in monochrome), including the boxes as well as the waveforms.

Once the print is complete, reconnect the cable to the unit and click on DEFAULT, as discussed in a moment.

Do not click on PRINTER if the printer is not ready to receive data. Doing so could cause a **Device Fault** error from the computer and a program crash.



Digital display screen showing the 8-bit logic levels for two analogue input waveforms.

You may be able to recover from the crash by ensuring the printer is ready and then pressing the <ENTER> key on the screen error display word OK. This will reveal the program listing at the point at which the error occurred. Now press function key <F5>.

If the computer finds that all is well, it will resume the program at the point where it crashed and send the screen data to the printer. Failing that, you will need to fully restart the program by using <shift-F5>.

If you are running the *PIC V-Scope* program through Windows, you can produce screen dumps which can be printed in colour (if you have a colour printer!). When ready, with the program Holding, press the keyboard's <Print Screen> key. The screen will momentarily flicker as data is copied into the Windows clipboard. Now exit the program.

From the Windows desktop screen left-click the Windows Start icon to reveal the Start menu. Select Programs, select Accessories and then select Paint. Within Paint click on Edit and then on Paste. The image stored in the clipboard will be

placed on screen. It can be saved under whatever name you choose, as well printed out to paper.

If the program is being run under MS-DOS, the <Print Screen> key works differently. The screen's text data is output directly to the printer, but not the graphics data (unless you have a DOS Graphics routine installed and active – consult your MS-DOS handbook).

OTHER OPTIONS

FREQ OFF

The frequency and amplitude analysis routine may be turned off or on by clicking in the FV ON column at bottom right. This does not affect the rate at which the PIC samples data, but does increase the rate at which each sample batch is displayed.

DEFAULT

If *PIC V-Scope's* power is interrupted while the display program is running, the PIC's mode setting data will be lost and it will revert to 2K 2-channel mode when reconnected. The same thing could happen if the printer port lead is disconnected (as is necessary when screen-dumping to a printer).

In this situation, once power or the printer lead are reinstated, it is necessary to click on the DEFAULT box to restore the PIC's mode values to those being used by the display program.

ADC BOX

The ADC box, middle right, can be clicked to alternate between F = 1 and F = 0. This toggles the **Tosc** sampling rate value discussed earlier.

DATE/TIME BOX

Towards screen top right is the Display Date/Time box. This is updated on each completion of a waveform plotted across the screen. It should not be regarded as a true real-time clock since it is not updated while sampling is taking place, nor when the display is being Held, nor when the printer routine is in use.

INFO BOXES

The top-right box shows the program identity and version number. The current version is V1.0. This number will be updated should any program modifications be introduced (which will be reported in *EPE* if they are).

The third box down on the right was originally introduced for the author's development information. Regard it as you wish – it has no significant user value!

OTHER READING

Discussing how to get the best out of any type of scope is beyond the reach of this article. However, the *EPE* feature article *More Scope for Good Measurements* (June/July –96) is a useful text to read. See the *Back Issues* pages.

May you too find *more scope* for improving your understanding and successful construction of electronic circuits through using *PIC V-Scope* (and continuing to read *EPE*)! □