

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

Constructional Project

MICRO PICSCOPE by JOHN BECKER

Visual signal monitoring with frills!

It is astonishing what opportunities are continuing to be revealed for the recently introduced PIC16F87x series of microcontrollers. This *Micro-PICscope* is a prime example of a design idea whose implementation was greatly simplified by using one of these devices.

The *Micro-PICscope* is a handy little item of test gear and of benefit to anyone's workshop. Using an alphanumeric liquid crystal display (LCD), it is basically a signal tracer, but one with the great advantage that it shows a representation of the signal waveform that is being traced. This is shown across eight of the LCD character cells and is a real-time trace of the monitored waveform.

Not only that, the display also shows the frequency of the signal being monitored, and its peak-to-peak voltage. The frequency range covered is basically for

audio, but frequencies well to either side of this range can be traced.

Several ranges of control are offered by switch selection, covering the sampling rate, and synchronization on/off for the 'scope display. The signal input is switchable to provide different maximum peak voltage monitoring ranges. Selection of AC or DC input is provided.

The entire design requires only two ICs, a PIC micro and an opamp, plus a 2-line by 16-character LCD. An optional third IC provides power regulation if required.

A typical example of the LCD screen display is shown below.

DESIGN HISTORY

Some 12 or more years ago, when the author first became familiar with "intelligent" alphanumeric LCDs, it became

apparent that by using the internal programmable character generator, their screens might be capable of displaying a simple representation of a signal being monitored. *(There is a GREAT article on these displays in the EPE Online Library at*

www.epemag.com, Ed.)

He had already designed and published a similar purpose unit based on multiplexed LED (light emitting diode) displays. In that unit (*PE* June '92), an LM3914 bargraph IC was used in conjunction with shift registers and digital multiplexers to portray a waveform across four 5 x 7 bit matrixed LEDs. It was very effective, although somewhat power hungry.

At the time though, microcontrollers were an unknown to the author and a method by which an LCD screen could be similarly used eluded him. Whereas LED matrices require only logic control, alphanumeric displays require a variety of data commands to be provided under program control.

For some simple operations LCD commands can be generated using codes pre-programmed into an EPROM (electrically programmable read only memory). This technique had already been successfully used by the author in a real-time *Morse Code Decoder* (*EE* Jan '87), but did not lend itself to circuit board signal tracing and monitoring.

The introduction of the versatile PIC16x84 microcontroller re-sparked interest in the idea, and it could have been done using that PIC with a separate analogue-to-digital converter (ADC). Then along comes the PIC16F87x family – with built-in ADC. Bingo, the idea was now as good as



Example of the screen display obtainable on the LCD module.

constructed to full workshop working order as a single chip design – apart from writing the command program, of course!

CIRCUIT DETAILS

In fact, as you will see from the complete circuit diagram for the *Micro-PICscope* in Fig.1, the practical implementation of the idea has been expanded a bit to use more than just a microcontroller (IC2). A buffering and gain setting opamp (IC1a) has been included as well. So too has a 5V regulator (IC3), allowing the unit to be powered from 9V or 12V supplies. If you have an existing well-stabilized 5V supply, IC3 may be omitted.

The signal being monitored is brought into opamp IC1a. As set by resistors R1 to R3, the

gain can be selected by switch S1 to be x1 (unity – via R2) or x10 (via R1). Other gain-setting values could be chosen instead. For example, a 10kΩ resistor could be used for R3 instead of a 100kΩ. This would provide switchable gains of unity and

1/10.

Switch S2 provides selection of AC or DC signal coupling, switching capacitor C1 in and out of circuit. The output from IC1 to the microcontroller is DC coupled. You will spot that the opamp is



Completed *Micro-PICscope* showing general layout of display and controls.

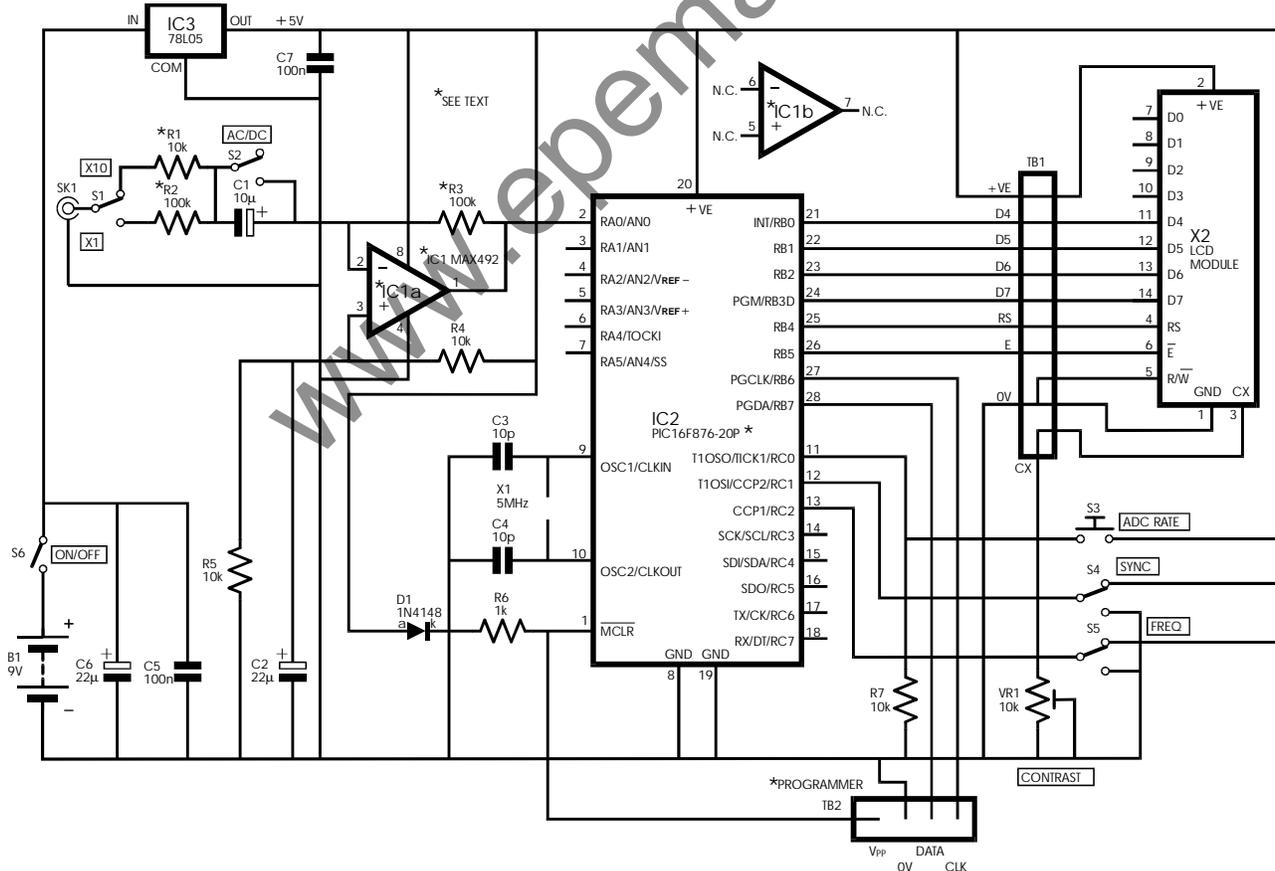


Fig.1. Complete circuit diagram for the *Micro-PICscope*. The voltage regulator IC3 is optional – see text.

used in inversion configuration. Software ensures that the signal is seen the "right way up"!

A MAX492 dual opamp is used for IC1, with the second half ignored. This device is part of the author's stock "library" and has proved itself for its nearly rail-to-rail output swing. The circuit has also been used with a TL082 device, which provides good frequency range, although does not offer full rail-to-rail output (typically more like 4V swing for a 5V split supply as used here – split by resistors R4 and R5).

MICROCONTROLLER

Microcontroller IC2 is a PIC16F876 device, operated at a clock rate of 5MHz, as set by crystal X1. Because of this clock rate, the 20MHz version of the PIC should be used. The "standard" version has a maximum guaranteed clock rate limit of 4MHz. However, you may find it interesting to note that the author has successfully used the 4MHz version at rates well in excess of 5MHz.

The PIC16F87x family has been discussed at length in previous issues of *EPE* and the '876 will not be described in detail here. Suffice to say that it has three input/output (I/O) ports, one of which, PORTA, can be used for analog-to-digital conversion via five of its pins (RA0 to RA3, plus RA5). In this design, only RA0 (pin 2) is used, its input being taken directly from the output of opamp IC1a at pin 1.

Internally, the PIC is programmed by the software so that the voltage reference for the ADC is taken as 0V to 5V (the power line voltage range). Consequently, an A-D conversion value of 255 results

when the input to RA0 is at the positive supply line level of 5V. A result of zero occurs when the RA0 input is at 0V.

Output to the LCD (X2) is via PORTB, using lines RB0 to RB5 to control the display in conventional 4-bit mode. Connections to the LCD are via the terminal pin block TB1. The order of the pins, both physically on the printed circuit board (PCB) and in terms of program control, is identical to that used by the author in many of his recently published designs. Display contrast of the LCD screen is set by preset VR1.

EXTERNAL CONTROL

External control of the PIC's monitoring and timing functions is actioned via PORTC, through pins RC0 to RC2. The functions controlled are the ADC sampling rate (via S3), waveform synchronization on/off (S4), and frequency counter display on/off (S5). These will be discussed later.

Pins RB6 and RB7, whilst not actively used by the design itself, can be used to program the PIC via a suitable programmer, such as *PIC Toolkit Mk2* (May-June '99). The MCLR pin (master reset) is normally powered at 4.3V (5V – 0.7V) via diode D1 and buffering resistor R6. This allows programming voltage control without disturbing the normal 5V supply rail from voltage regulator IC3.

Terminal pin block TB2 provides access to MCLR, RB6, RB7 and the 0V common line. The pin order on the PCB is the also same as that used by the author in previous designs. This will be welcomed by those who have established a plug-in link between *Toolkit Mk2* and such designs! (The author intends for

all his future PIC designs to use this same pin configuration for LCD and programming connections.)

CONSTRUCTION

Details of the PCB component and track layouts are shown in Fig.2. This board is available from the *EPE Online Store* (code 7000259) at www.epemag.com

Regular readers will know this author's preferred constructional order: wire links, resistors, diodes, small capacitors, IC sockets and then on upwards in order of component size.

Dual-in-line sockets should be used for IC1 and IC2. Note that microcontroller IC2 is the narrow version (0.3in width between pin rows, as opposed to 0.6in).

Details of the switch and signal input connections are also shown in Fig.2. Socket SK1 may be a different type to that shown if preferred.

The LCD module might have one of two possible pin connection arrangements. They are shown in Fig.3.

As always, do a thorough check of the component positioning, orientation and solder joint integrity before applying power. Do not insert IC1, IC2 or the LCD until the output from regulator IC3 has been validated, exactly 5V (within a few millivolts) for a supply between 7V and 12V.

An output voltage from IC3 other than 5V will usually indicate a fault in construction – too high and IC3 is wrongly inserted, too low and there may be a short circuit somewhere on the board.

FIRST RUN

When happy about the power supply, test the circuit with IC1, IC2 and the LCD plugged in (correctly!). Set switch S4 (Sync) off and S5 (Frequency) on. When power is applied, the PIC first goes into an LCD initialization routine, in which it sets the LCD for 2-line 4-bit mode.

Following this, text messages similar to those in the photographs should appear. The signal trace display in the top left LCD character cells should show as a straight line about half way up the screen. Adjust preset VR1 to set the screen contrast (you may see nothing at all until you have adjusted it).

Having read the sections all about the control program, you can then feed in an audio signal, play with the switches, and see the results on screen. The input signal amplitude should be selected so that the majority of the LCD vertical pixel range is used.

ENCLOSURE

A small plastic box was used to house the prototype. The PCB has been designed so that the LCD can be mounted above it using stand-off pillars, although the prototype did not use this option.

The rectangular viewing slot was cut by first drilling small perimeter holes and using a file to smooth the edges to shape and size. Holes must also be drilled to suit the switches and input socket. The prototype used a 3.5mm jack socket for the power input, but other techniques, such as a battery connector, can be used.

A-TO-D CONVERSION

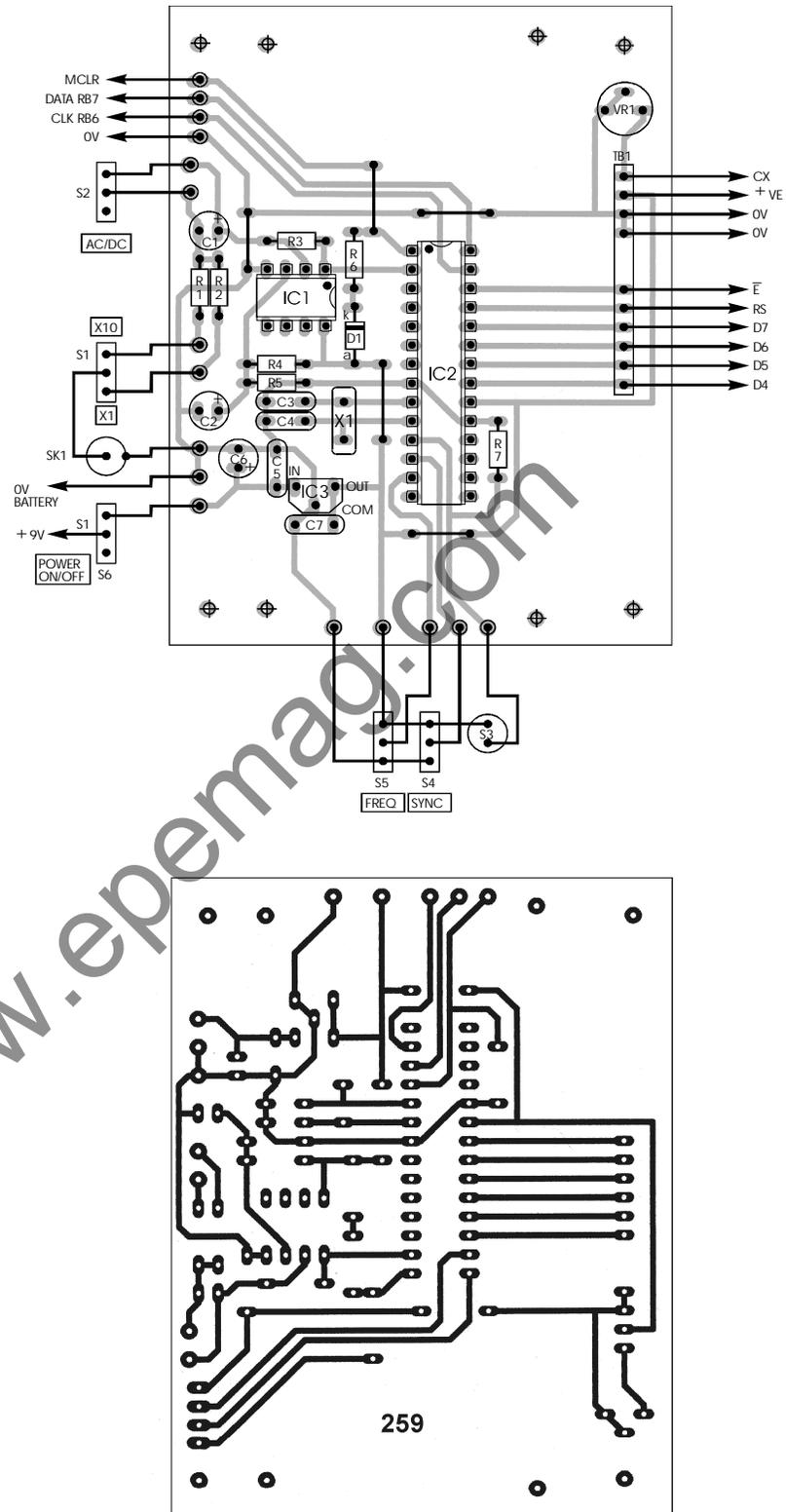


Fig.2. Printed circuit board component layout, wiring to the off-board components and (approximately) full-size copper foil master for the Micro-PICscope.

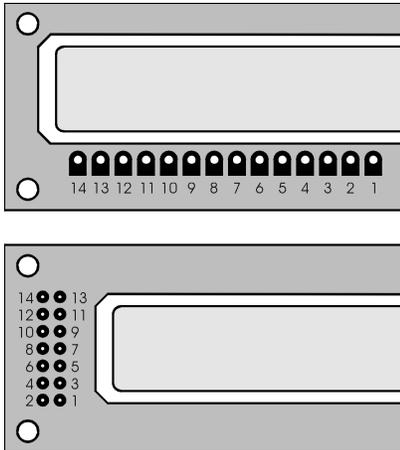


Fig.3. Pinout arrangement of the two basic LCD formats.

Basic A-to-D conversion using a PIC was discussed in the *Mini PIC16F87x Tutorial* of Oct '99. A simplified version of the routine described there is used here:

In the START routine while in PAGE1, register ADCON1 is set with the binary value of %00000101, which tells the PIC that the 2-byte ADC register is to be justified left, with RA0 as an analog input referenced to +VE and 0V.

Then, back in PAGE0, by loading register ADCON0 with a binary value of %010000001, ADC conversion is activated at an oscillator rate of one-eighth of the clock rate ($F_{osc}/8$).

(Readers who write their own PIC software should note that in order for PAGE commands to be used with the *Toolkit* programmer and the PIC16F87x family, bit RP1 of the PIC's STATUS register must be set to 0 – as it is in the START routine of the *Micro-PICscope* source code.)

A single ADC sample is taken when the command **BSF ADCON,GO** is issued, where the quaint GO term (Microchip's description!) refers to bit 2. Sampling and conversion are

not quite instantaneous and the program repeatedly polls the GO bit until it goes low, signifying an end to the conversion process. There are several subroutines which are used in the program to perform this task, one of them being:

```

WAITS1: BTFSC
ADCON0,GO
GOTO WAITS1
MOVF ADRESH,W
    
```

When the GO bit is clear, the command **MOVF ADRESH,W** retrieves the high byte of the 2-byte conversion result. Because the display is only eight pixels high, the low byte is not needed (see the *Mini Tutorial*, or the PIC16F87x data book, for details of the conversion result format options).

The value held in ADRESH (and now also in W) is that which represents the voltage level of the signal being sampled. It cannot yet, however, be put out to the LCD screen, there's a great deal of work to be done first! For the moment, this value is simply stored in one of a set of temporary memory locations. The conversion and storage is performed 128 times before further action is needed.

DISPLAY PRINCIPLE

Before that "further action" is described, it is first necessary to understand the concept behind the way in which a waveform can be displayed on the LCD by making use of its character generator.

You already know that the LCD used here has two display lines each having 16 character cells. Each of these 32 cells consists of a matrix of LCD pixels, arranged as five across by eight down (see Fig.4a).

Normally, the LCD module places pre-programmed (as part of the module's control chip manufacturing process) alphanumeric data into these cells according to command codes from an external source, a PIC in this case.

However, the module has the facility to allow eight characters to be "designed" by the user and called as well as the standard alphanumeric set. These characters are stored at module address locations 0 to 7.

At first sight, when examining the LCD data sheet, it might appear that address locations 8 to 15 can also be used to hold custom characters as well. Regrettably for an application such as this PICscope, addresses 8 to 15 only hold repeats of the data at addresses 0 to 7. Thus only eight addresses can be used for alternative character data, hence the PICscope only having eight cells for waveform display.

Eight cells each having five pixels horizontally allows 40 waveform samples to have their values plotted at eight vertical pixel levels.

The reason that 128 samples are taken even though only 40 will be displayed from each block is to allow for frequency and amplitude values to be more readily established.

CHARACTER GENERATOR

The way in which data for a character cell is evaluated is illustrated in Fig.4b. Each of the seven rows making up the cell display are treated individually. The five pixels of each row are numbered from 4 to 0, allowing a 5-bit binary number to be compiled. Logic 1 in a bit

position turns on the equivalent pixel, while logic 0 turns it off.

Having established which bits are to be active, the 5-bit binary number for each row (expanded to 8-bit with zero in bits 7 to 5) is sent to the required character generator address, of between 0 and 7. The same procedure can be used for the other seven possible addresses, each of them storing different data, as appropriate.

When character data is being generated in this way, the LCD is first told that the data about to arrive is destined for the character generator rather than for the screen display. In the program the initial command is given by:

```
MOVLW %01000000
CALL LCDLIN
BSF RSLINE,4
```

which sets the character generator to address 0 from which address onwards the

Resistors

R1, R4, R5, R7 10k (4 off)
R2, R3 100k (2 off)
R6 1k

All 0.25W 5% carbon film

Potentiometer

VR1 10k miniature round preset

Capacitors

C1 radial electrolytic, 16V
C2, C6 22u radial electrolytic, 25V (2 off)
C3, C4 10p ceramic, 5mm pin spacing (2 off)
C5, C7 100n ceramic, 5mm pin spacing (2 off)

Semiconductors

D1 1N4148 signal diode
IC1 MAX492 dual opamp
IC2 PIC16F876-20P microcontroller (20MHz version, 0.3-inch width) preprogrammed
IC3 78L05 +5V 100mA voltage regulator (see text)

COMPONENTS

Miscellaneous

S1, S2, S4, S5 miniature s.p.d.t. toggle switches (4 off)
S3 miniature push-to-make switch
S6 miniature s.p.s.t. (or s.p.d.t.) toggle switch
SK1 BNC socket (see text)
X1 5MHz crystal
X2 2 line x 16 characters per line alphanumeric crystal display

Printed circuit board available from the *EPE Online Store* (code 7000259) www.epemag.com; power supply connector (see text); plastic case, 150mm x 80mm x 50mm; 8-pin DIL socket; 28-pin DIL socket; 1mm terminal pins (or strips) for TB1 and TB2; PCB and LCD supports (8 off); connecting wire; cable ties; solder, etc.

**See also the
SHOP TALK Page!**

*Approx. Cost
Guidance Only*

\$31

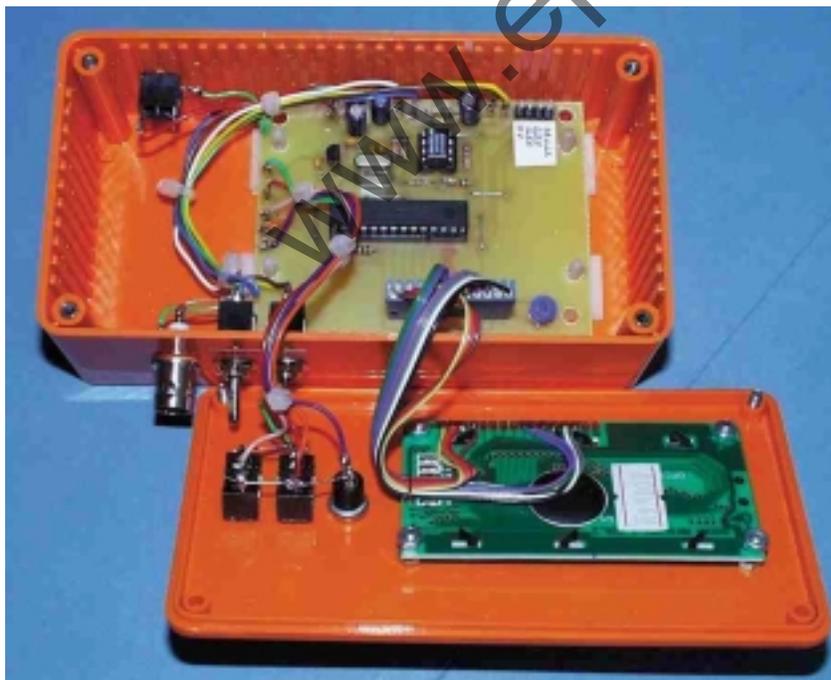
“designed” data is to be stored, the address incrementing each time a data byte is written to the

LCD. The LCDLIN call is to one of several standard routines, which the author wrote some years ago to send various commands and data to an LCD module.

For all eight character cells to be fully programmed, $8 \times 8 = 64$ bytes of data are written to the character generator.

Once the character generator has been programmed, the data held at each of the eight address blocks can be called to the screen by simply accessing that address in the same way that “normal” alphanumeric data is accessed.

For example, to display letter “A” on screen you might use either **MOVLW 'A', CALL LCDOUT**, or **MOVLW 65, CALL LCDOUT**, the value of 65 being the ASCII value for capital letter ‘A’. In both cases the character held at character generator address decimal 65 would be displayed on screen, which,



Completed unit showing the LCD module mounted on the lid of the case and wiring to the PCB.

Constructional Project



fortunately for us all is indeed the letter "A".

Similarly, to show the character newly programmed into address 3, the commands would be **MOVLW 3, CALL LCDOUT**.

The data held at character generator addresses 0 to 7 can be changed as often as required. In this design it is typically changed about twice per second (faster with S4 and S5 off). All data at these addresses is lost when power is switched off.

WAVEFORM CHARACTERS

When a full block of samples has been converted and stored, the data is then analyzed for amplitude and compiled into 64 bytes for sending to the character generator. Row 8 is the top row and (naturally) represents the highest voltage range that can be displayed. The display is, of course, compressed to one-eighth of the conversion value received.

The analysis procedure is far more complex than can be described here. It is not just a matter of ascertaining which row a value should be allocated to. The result also has to be "doctored" so that the active pixels are seen to be as close to

a continuous line as possible.

For instance, suppose the waveform is alternating rapidly between high and low levels at a rate faster than the sampling can keep pace with. Without corrective action, you might only see pixels on the upper and lower lines, those between remaining blank.

The corrective action fills in those blank pixels so that they



Close-up of typical screen display. There are three rates, the maximum counting frequency is about 17kHz.

appear as though they naturally follow on from each other. This was an extremely difficult process to write the program for! Even experienced programmers might have difficulty analyzing the way in which the source code has been written – be warned!

However, you don't need to understand the program in order to use it. Just load it into your

PIC (or buy a ready-programmed PIC – see later).

Although there is a great deal of processing being carried out, each batch of sampled data is displayed in rapid succession and really does give a "real-time" display of what is happening in a monitored circuit.

FREQUENCY AND AMPLITUDE

Each batch of data is also analyzed for waveform frequency and peak-to-peak amplitude values.

Amplitude is easily determined by simply looking for the maximum and minimum conversion values and then relating them to the maximum possible sample level of 255. The latter, as said earlier, represents the positive line level, which has been assumed to be exactly 5V. If you need greater accuracy for signal level voltages use your multimeter to read them! The PICscope is only intended for providing an approximate value (but it's still pretty accurate).

Note that the PIC does not monitor which gain setting has been selected. It simply reports the voltage it finds on its RA0 pin. You must mentally adjust the value shown if the gain is

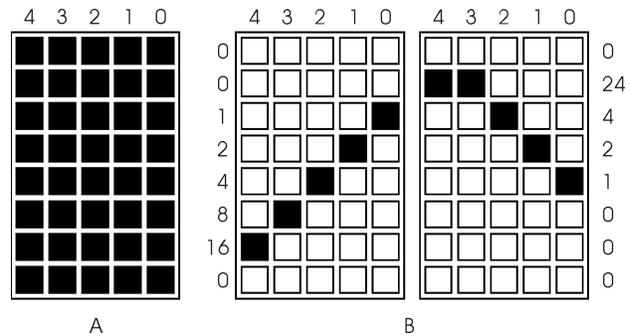


Fig.3(a). LCD character cell matrix with all pixels on, (b) example of waveform representation across two character cells.

other than unity.

Frequency is assessed by counting the number of times the voltage level crosses a preset threshold value. The result is then divided by two to obtain the equivalent number of cycles per batch, the rate of data acquisition being pre-determined by the sampling rate, which in turn is relative to the master clock rate.

Much effort went into writing the software so that relative timings were maintained consistently, irrespective of conditional branch timings in the sampling routines.

There are three rates at which the ADC can be set to sample waveforms, selectable by pressing switch S3. The rates cycle as a repeating group of three, numbered from 0 to 2. The number of the selected rate is shown at the top right of the screen. It is not a representation of the frequency range covered.

The rates are set according to the value by which the master clock oscillator is divided via the ADCON0 prescaler. This value is set by bits 6 and 7 in the ADCON0 register. Rate 0 sets $F_{osc}/2$, rate 1 sets $F_{osc}/8$, rate 2 sets $F_{osc}/32$. The routine which reads S3 and sets the bits commences at label **TESTIT**, following on into **GETRATE**, near the end of the source code listing.

FREQUENCY CALCULATION

Relating the ADCON0 sample rate to the actual frequency of the signal has to take into account the time taken for all the commands in the sampling routine to be performed. As experienced programmers will acknowledge, such matters are not always



Completed circuit board mounted on self-adhesive plastic stand-off pillars.

readily calculated (friend and *EPE* author Andy Flind has researched heavily into this – we hope he'll one day share it with us all!).

The solution here was to count the swings above and below the threshold level and then divide the answer by a conversion factor, with a separate factor for each of the three sampling rates.

Using a subtractive technique, the conversion involves fractions, which are fixed in the software as two-byte binary numbers.

For example, for Rate 0, the MSB is set at decimal 85 and the LSB at 70, which has an equivalent decimal value of 21830 ($256 \times 85 + 70$). From this value is repeatedly subtracted the count value determined when counting the swing changes, each successful subtraction incrementing a counter. Thus the cycle count is divided into the conversion factor, the secondary counter providing the answer. The result is remarkably accurate. (MSB and LSB, incidentally, mean

Most and Least Significant Bytes, respectively.)

During prototype testing, the unit was fed with a frequency of 4000Hz and the "fraction" values repeatedly adjusted by trial and error until the LCD also showed a value of 4000Hz.

Having established the values for the three ranges, the input frequency was raised to see how far accuracy was maintained, the results were:

Rate	Generator	Display
0	17007Hz	16984Hz
1	17007Hz	16998Hz
2	5827Hz	5812Hz

Beyond those frequencies, the unit began to display harmonic frequency values because the generator rate exceeded the rate at which the waveform could be sampled.

The values programmed into the unit depend, of course, on the exact frequency at which the crystal controlled oscillator functions. However, crystal controlled oscillators, while not being perfectly tuned to a given frequency, do stick closely to it.

Consequently, other units should achieve results that are not too different from the author's.

Those who wish to experiment are referred to the sub-routines at **GETFREQ0**, **GETFREQ1** and **GETFREQ2**, where the preset values can be changed. The program will naturally need to be re-assembled and reprogrammed into the PIC.

Analysis of the peak-to-peak and frequency values can be switched off using S5. This speeds up the rate at which the screen is fed with a fresh waveform display.

SYNCHRONISATION

Switch S4 turns the waveform synchronization facility on and off. When synchronization is on, before each sampling batch is started the software waits until the waveform voltage has twice passed through a preset trigger level. Only then does it start sampling the rest of that batch.

This facility allows the waveform display to start about half-way up the screen, providing a degree of stability to

a repetitive waveform. Inevitably, the process increases the wait period before each new display is shown. There is a time-out counter, which prevents the system from "locking-up" should the waveform not cross the sync thresholds.

It is best to start off sampling any new signal with sync off, only turning it on once adequate signal levels are being received. The source routines, which control sync start at label **WAITS1**. The full batch sampling routine commences at **WAITAD0**.

IN THE PIPELINE

That, in a nutshell, is really all there is to tell about this astonishingly simple signal monitor (simple in hardware terms – but certainly not regarding software writing!). Designing it has fulfilled one of the author's ambitions. Software is in TASM.

Another yet to be fulfilled is to design a more advanced LCD based scope, which will give far greater resolution to the waveform shapes displayed. Such a design, though will have to wait until the exorbitant cost

of LCD graphics displays comes down greatly!

What is in the pipeline, however, is the *Virtual PICscope* in which one of the PIC16F87x family is used to sample two waveforms simultaneously and output the data to a PC computer for display on its screen.

Finally, if you have any ideas for PIC-based workshop designs (or any other type of design, of any sort), please let us know.

Go to next section