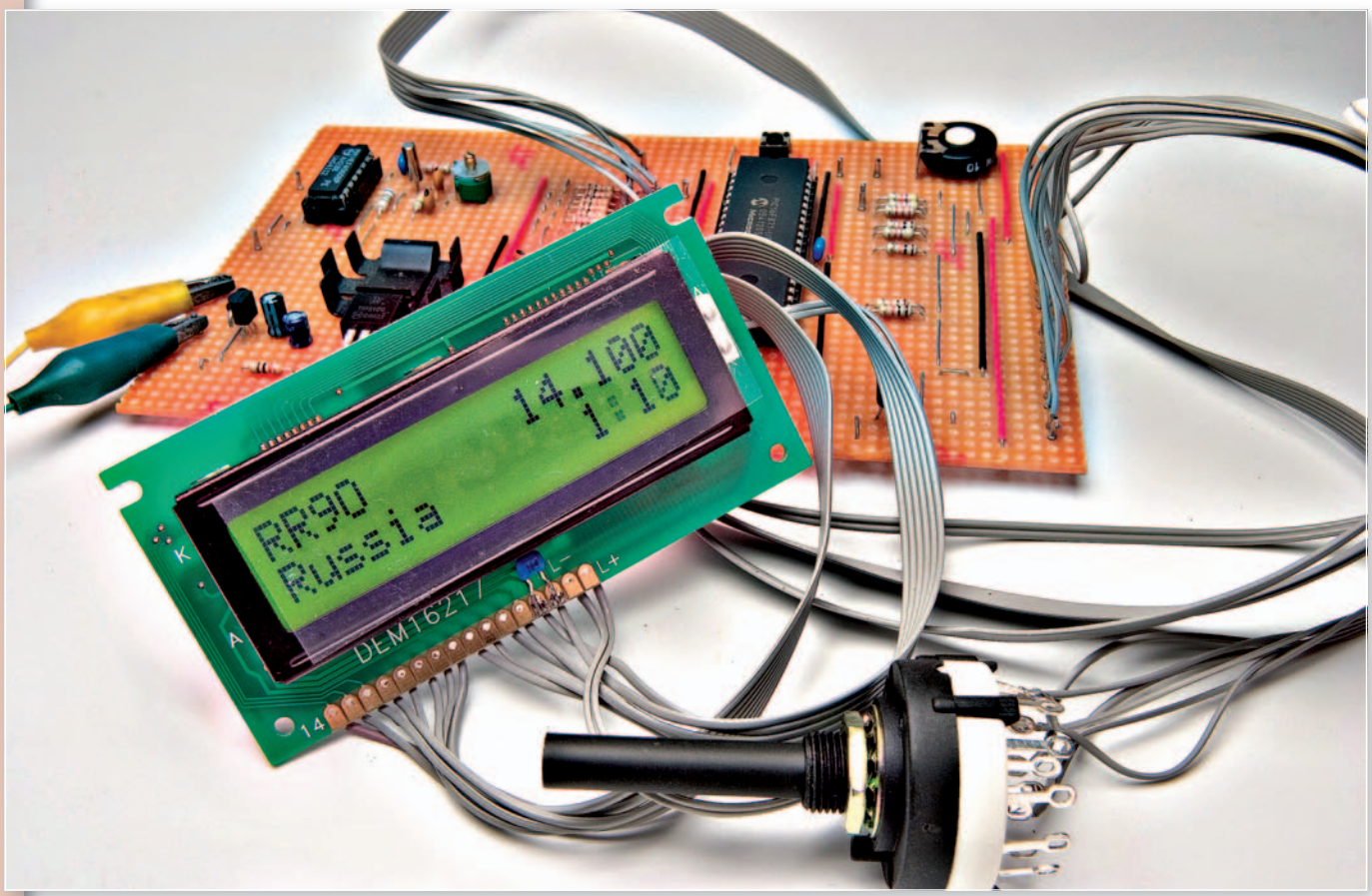
The Practical Wireless IBP Beacon Clock (PIC Version) part 2

Providing an LCD Readout



This month's project provides a visual display of the IBP beacon transmitter's callsign as they appear in the 18-beacon, three minutes cycle.

In the May 2007 issue of *PW*, I described a PIC-based International Beacon Project (IBP) clock, which used light emitting diodes (l.e.d.s) to indicate, in real time, which of the 18 IBP beacons should be transmitting in its allocated time slot on 14.100, 18.110, 21.150, 24.930 or 28.200MHz. The design was effectively a five band microcontroller version of the CD4000 series logic design featured in the December 2001 and January 2002 issues of *PW*.

Following on from Part 1 of the PIC-based International Beacon Project (IBP) 'clock', Phil Cadman G4JCP describes an extremely effective additional unit – a liquid crystal display unit that will provide visual identification of the 18 individual beacon time slots.

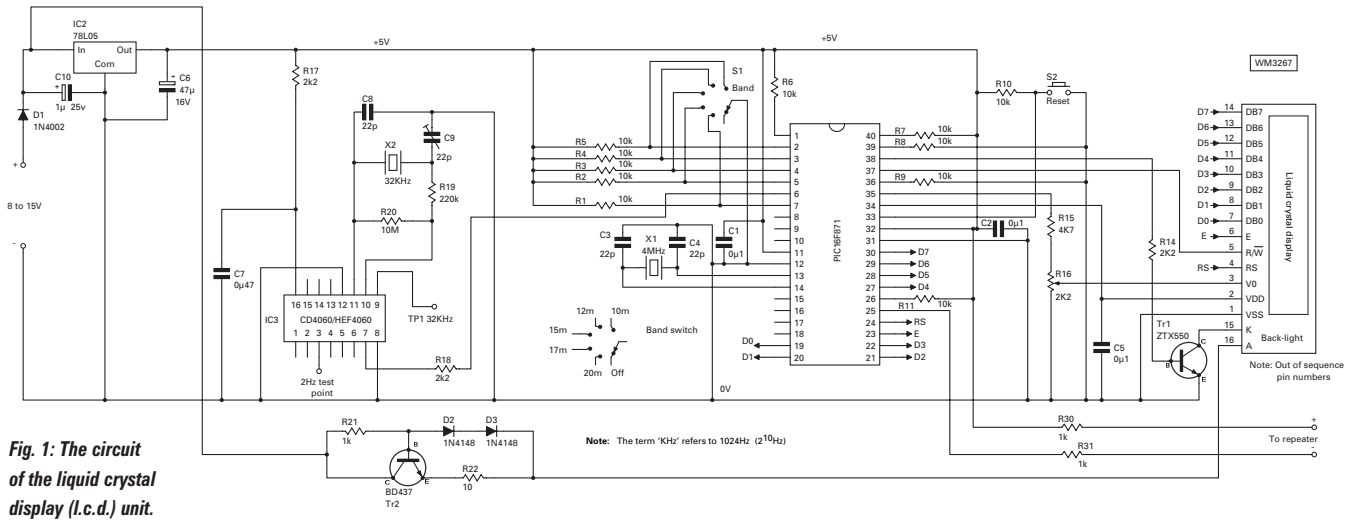


Fig. 1: The circuit of the liquid crystal display (l.c.d.) unit.

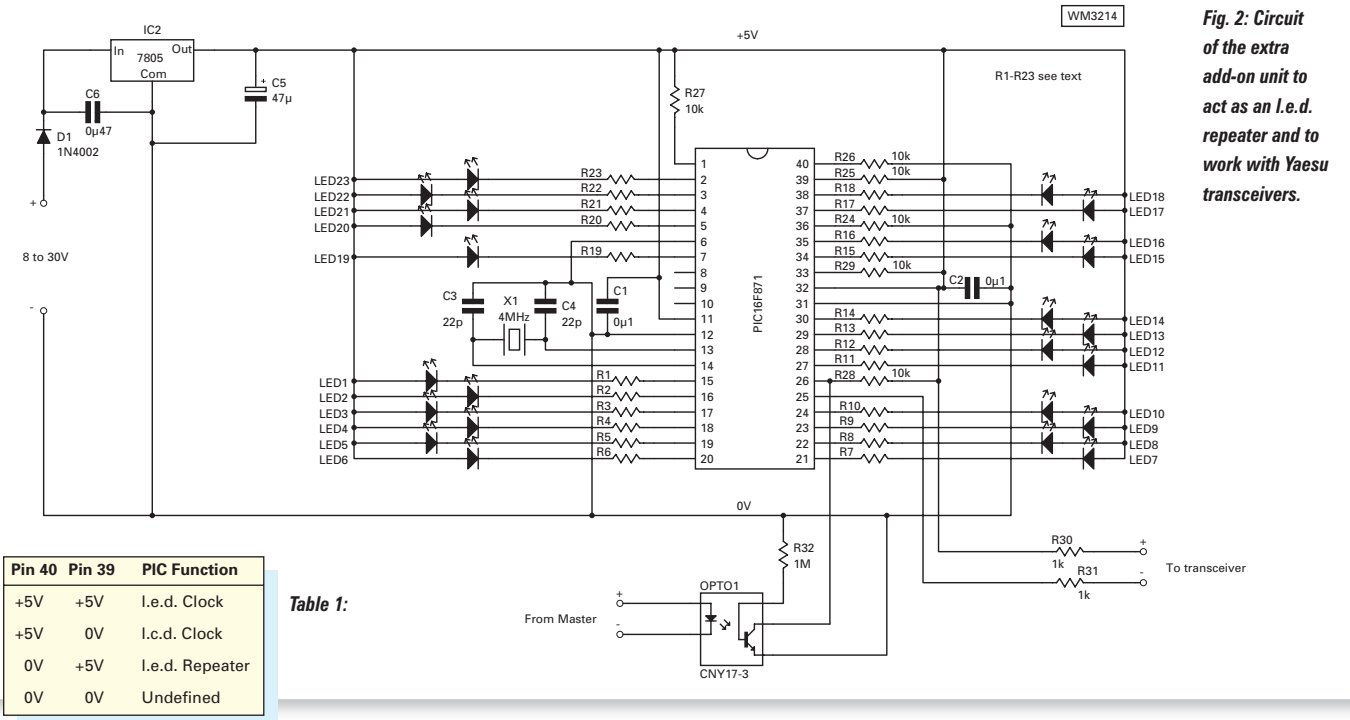


Fig. 2: Circuit of the extra add-on unit to act as an l.e.d. repeater and to work with Yaesu transceivers.

Pin 40	Pin 39	PIC Function
+5V	+5V	l.e.d. Clock
+5V	0V	l.c.d. Clock
0V	+5V	l.e.d. Repeater
0V	0V	Undefined

Table 1:

Using a microcontroller allowed me to trade hardware for software, making the circuit comparatively simple but adding the complication of having to write some software! This exchange is almost always worth making, even in the case of the l.e.d. clock.

Once we have a microcontroller in the design, other, more complex options become possible, including the use of a liquid crystal display (l.c.d.). Physically replacing the l.e.d.s used last time with an l.c.d. is quite straightforward and the circuit - shown in **Fig. 1** – is no more complex than the l.e.d. design.

The Display Circuit

Once again the timebase for the clock is provided by a 32,768Hz miniature watch crystal connected to a CD/HEF4060 oscillator/divider integrated circuit (i.c.). An output at 2,048Hz is taken from pin 7 of the CD/HEF4060 to drive the PIC's timer0 function. Trimmer capacitor C9 should be adjusted so that X2 resonates at exactly 32,768Hz.

The frequency can be checked either by measuring the buffered 32,768Hz output on pin 9 or by measuring the period of the 2Hz output on pin 3. The values of C8 and C9 are suitable for a crystal requiring a load capacitance of 12pF (a common value for these watch crystals). If C9 can't quite pull the crystal exactly on frequency, connect a 5 or 10pF ceramic capacitor in parallel with the trimmer.

The maximum current drawn at 5V is small (less than 10mA), so a 78L05 regulator is perfectly adequate. The band select switch, S1, selects the required band by connecting the appropriate band input (normally pulled high) to 0V. The sixth position is used for turning the l.c.d. off (more about this later). Switch S2 is the reset switch, which is used to set the clock to the beginning of the three minute beacon cycle. The three minute cycle starts on the hour and repeats at three minutes past, six minutes past, nine minutes and so on up until 57 minutes past the hour before starting again at the next hour. There are 20 complete three minute beacon cycles per hour.

The l.c.d. clock can drive an l.e.d. repeater - see **Fig. 2** - communicating at 4800 baud using the PIC's USART. I've chosen 4,800 baud because that's the default baud rate used by some Yaesu transceivers - specifically the FT-817/857/897 series - and because it's not too fast for the opto-couplers. (Again, more about this later).

I described the function of many of the individual pins on the

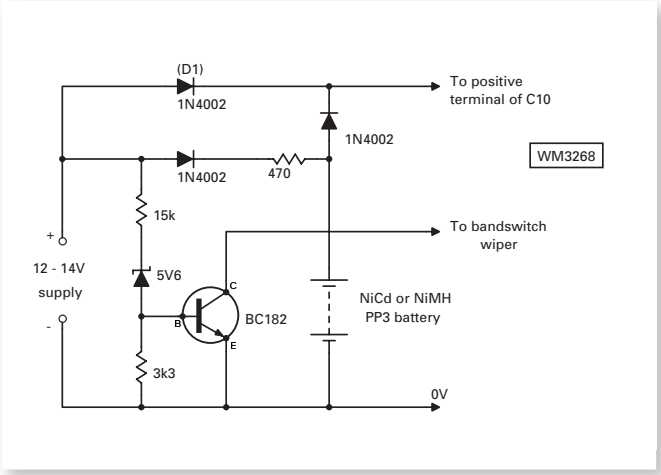


Fig. 3: Power supply and charging circuit.

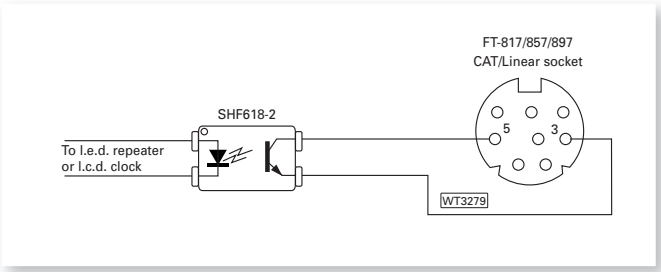


Fig. 4: Opto-coupler circuit for use with Yaesu equipment. The pinout is as viewed onto the socket or from behind the plug.

PIC last time, so I won't repeat the information here, save for the function inputs, pins 39 and 40. At switch on, the software checks the voltages on pins 39 and 40, and these voltages are then used to determine the function of the PIC. (See **Table 1**).

Unused pins on the PIC – the lower four pins of Port C and the three pins of Port E – are left unconnected. During initialisation, the software configures them as outputs and sets them low. They're free to be used as desired by anyone modifying the software.

Of course, the big difference between the l.c.d. clock and the l.e.d. clock is the liquid crystal display itself. Driving an l.c.d. is not the same as driving a light emitting diode! For anyone unfamiliar with these very useful devices, the following information may be of interest.

The LCD Module

The l.c.d. module I've chosen is a two line by 16-character display type **DEM 16217 SYH**. It's an inexpensive 'supertwist' display with an l.e.d. back light. Like the vast majority of character based l.c.d. modules, it uses an **Hitachi HD44780** (or equivalent) controller chip. Full data on the HD44780, which can handle up to 80 character displays – is available on the Internet. (As is a wealth of example software to drive it!)

The HD44780 has been the industry standard for many years, so readers should feel free to try any surplus two line by 16 character l.c.d. module they may have! One word of caution though: the required liquid crystal driving voltage - Vo - can vary somewhat between displays. This voltage is referenced to Vdd (the +5V supply) and is measured in the negative direction. It's made variable because it's effectively the contrast adjustment and it can (and usually does) change with temperature.

The DEM 16217 SYH will work with a Vo of around 4.5V. Remember that's negative with respect to the +5V rail, so relative to ground it's about 0.5V. Potentiometer R16 provides any necessary adjustment. Other l.c.d. modules may need a Vo greater than 5V, hence they will need a **negative** supply rail.

Character based displays usually have standard pin outs. Starting at pin 1, there is Vss (0V) and then Vdd (+5V). Pin 3 connects to Vo as mentioned above. Data is transferred to and from the l.c.d. module through eight data lines (DB0 to DB7), which are managed by three control signals – RS, R/W and E. The register select (RS) signal (pin 4) determines whether the byte sent to the display is interpreted as a command or as display data (low for a command, high for data).

When the read/write signal (R/W) on pin 5 is low, data is transferred from the host processor to the display. When high, data can be read out of the display's internal memory by the host.

In many designs, no reads from the display are needed so R/W is often tied directly to Vss (0V). The enable signal (E) on pin 6 strobes the command or data into (or out of) the display.

Finally, pins 7 to 14 carry (in sequence) the data bus DB0 to DB7. I'm sure that readers who remember the early days of computing and the Motorola MC6800 microprocessor (and the Mos Technology 6502), will recognise these signals. Here, software in the PIC is used to create what is in effect a microprocessor data bus and control signals.

Backlights & Variations

Not all displays have a backlight and the type of backlight can vary. The DEM 16217 SYH uses an array of light emitting diodes. Rather than power the backlight from the +5V rail, I've chosen to use a constant current source fed from the incoming supply. The backlight current is set by R22 (10Ω) giving about 75mA (0.75V divided by 10Ω). You can change R22 to adjust the brightness, **but please do not** let the backlight current exceed 120mA!

If you always plan to run the l.c.d. clock from a fixed supply,

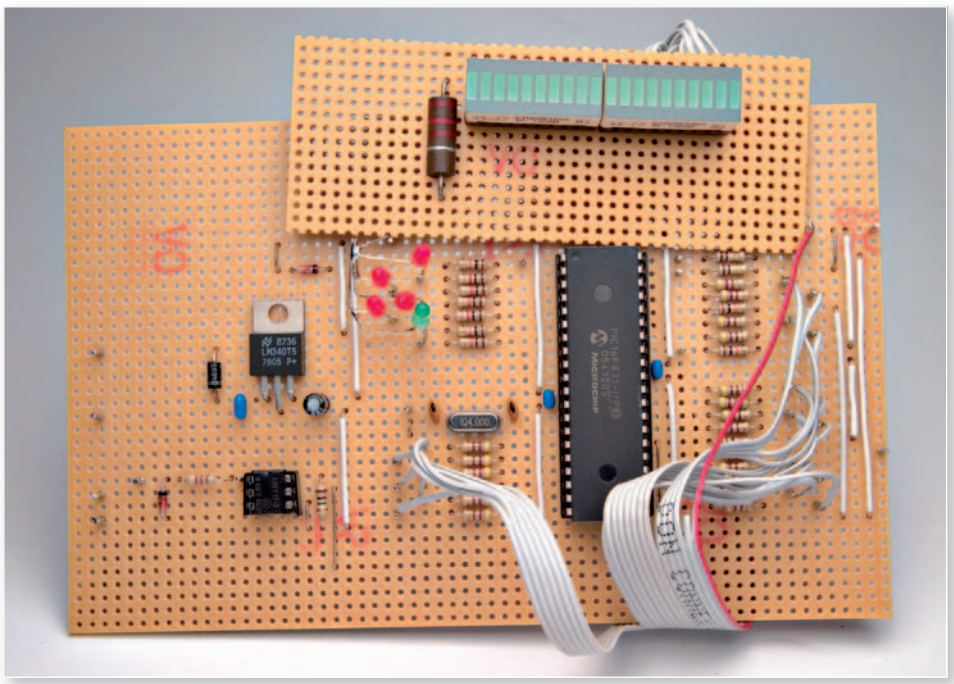


Fig. 5: The l.e.d clock repeater (front).

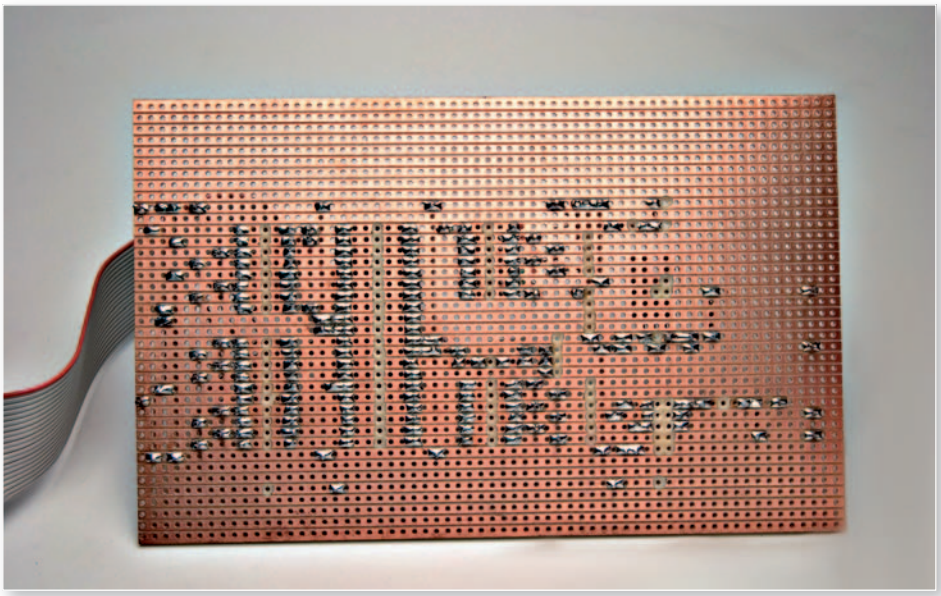


Fig. 6: Rear view (copper track side) of Veroboard layout of the l.e.d. clock repeater

Powering the display via the PIC introduces some undesirable impedance in the supply, so C5 provides local decoupling. It's best to mount this component on the l.c.d. module itself.

To save power when running the clock from batteries, simply turn off the backlight. Either wire an on/off switch in series with the backlight, or use a single pole change-over switch to switch the base of Tr1 from its connection with R14 to the 0V rail. The latter option saves 2mA of base current. Every little action helps reduce current consumption!

then Tr2 and its associated components can be replaced by a resistor. Try a 100Ω 2W component when using a 13.8V supply. By the way, backlight connections **do vary** and they can cause confusion, so always check the display's data sheet. You'll notice that this l.c.d. has its backlight pins - 15 and 16 - adjacent to pin 1, so be aware!

By the way, you can use other than the specified transistors for Tr1 and 2 as long as they are broadly equivalent. The critical parameter is the gain: Tr1 should have a minimum gain of 100 at a collector current of 100mA and Tr2 a minimum gain of 40 at 100mA.

Clock Software

Although the IBP clock software doesn't read from the display, the R/W line is nevertheless connected to a PIC port pin. After all, the R/W line has to be tied somewhere and using a port pin retains the option of reading the display data should the need ever arise.

A somewhat unusual feature is powering the display via two of the PIC's port pins. This allows the PIC to completely switch off the display when it's not needed. Similarly, transistor Tr1 allows the PIC to control the backlight too.

When the band switch is in the off position, only the PIC itself and the 32KHz time base remain powered. In this condition, the 78L05's quiescent current is often greater than the current taken by the PIC, so to reduce the 'display off' current to an absolute minimum, I suggest replacing the 78L05 with a TS2950CT-5.0 ultra low dropout regulator (Maplin code N69CA). When fitted with this alternate regulator, the prototype l.c.d. clock consumed just 1.2mA with its display off.

The diagram, **Fig. 3**, shows one method of providing a rechargeable battery back up for when the clock is normally powered from a 12 to 14V supply. Removal of the external supply automatically switches the band switch into the 'off' position by disconnecting the wiper from 0V. **Note:** D1 in Fig. 3 is the same component as D1 in Fig. 1.

Other Display Points

Next, I think it's a good idea to mention some other points concerning the display. To start, The HD44780 can communicate with the host processor using just four of its eight data lines (so called 4 bit mode).

Moving the l.c.d. data bus to the unused lower four pins of Port C will free up the whole of Port D and, as Port E is already unused, it's then feasible to replace the 40-pin 16F871 with the 28-pin 'skinny DIP' 16F870. (The necessary software changes would be minimal).

Map Mounted Attraction

While a liquid crystal display is very nice, l.e.d.s mounted on a map are both attractive and informative when illuminated, particularly if they're mounted on a great circle map. At a glance, you'll be able to see both the beacon heading and relative distance. So, rather than forgo such a display (or have to make both an l.c.d. clock and an l.e.d. clock and keep them synchronised!), I thought an l.e.d. repeater would be useful.

Repeater Function

The diagram, Fig. 2, shows the circuit of the l.e.d. repeater, which can be driven from both the l.c.d. clock and the l.e.d. clock. The circuit is very similar to that of the l.e.d. clock; the 32KHz timebase is not needed and the band switch is replaced by five l.e.d.s which indicate the band selected.

As in the l.e.d. clock, I suggest that constructors choose the values of those resistors (R1-23) in series with the l.e.d.s to give the desired brightness **but please don't** exceed an l.e.d. current of 15mA. Also, pin 36 on the PIC controls the l.e.d. drive, either active low - as in Fig. 2 - or active high. For active high outputs, simply tie pin 36 high by connecting the 10k resistor on pin 36 to +5V instead of 0V.

Communication between the l.c.d./l.e.d. clock and the repeater is via the PIC's USART, at 4800 baud. I've used an optocoupler to prevent ground loops and to give a measure of r.f. immunity. The serial transmission format is one start bit, eight data bits, no parity and one stop bit. Each time the l.c.d. or l.e.d. clock updates its display, a byte is sent to the repeater. It has the following format:

<D7 D6 D5> <D4 D3 D2 D1 D0>

<Band no.> < Beacon no.>

<D4 D3 D2 D1 D0> = Beacon number, one to eighteen. A beacon number of zero will switch all the beacon l.e.d.s off, and a beacon number of 31 will turn all the beacon l.e.d.s on (lamp test). Beacon numbers of 19 to 30 inclusive are ignored.

<D7 D6 D5> = Band number, one to five. Band 1 is 14.100MHz and band 5 is 28.200MHz. Again, a band number of zero will switch all the band l.e.d.s off and a band number of 7 will turn all the band l.e.d.s on. A band number of six is ignored.

It's possible to drive the repeater using an RS-232 serial interface. Just wire a 1N4148 diode in series with a 4.7kΩ resistor and connect them in series with the l.e.d. inside the optocoupler. Then connect all three components between signal ground (SG) and transmit data (TD) on the serial interface (cathode of the l.e.d. to SG). Naturally, you'll need to write a suitable program to drive the repeater.

Use With Yaesu Rigs

Not wishing to waste the USART's transmitter, I've given the repeater the ability to drive the CAT(tm) interface on Yaesu's FT-817/857/897 series of transceivers using the circuit shown in **Fig. 4**.

Whenever the repeater receives a different band number, it sends out a 'Set Frequency' command, thus automatically tuning the transceiver to the correct frequency. However, the transceiver must be set to c.w. on the five IBP bands of 14.1, 18.110, 21.150, 24.930 and 28.2MHz (on the chosen v.f.o.) for this to work properly. Once again, the optocoupler prevents ground loops. Oh, you can use a CNY17-3 optocoupler instead of the SFH618-2. However, the advantage of the SFH618-2 is its small size, which allows it to fit inside a mini din plug.

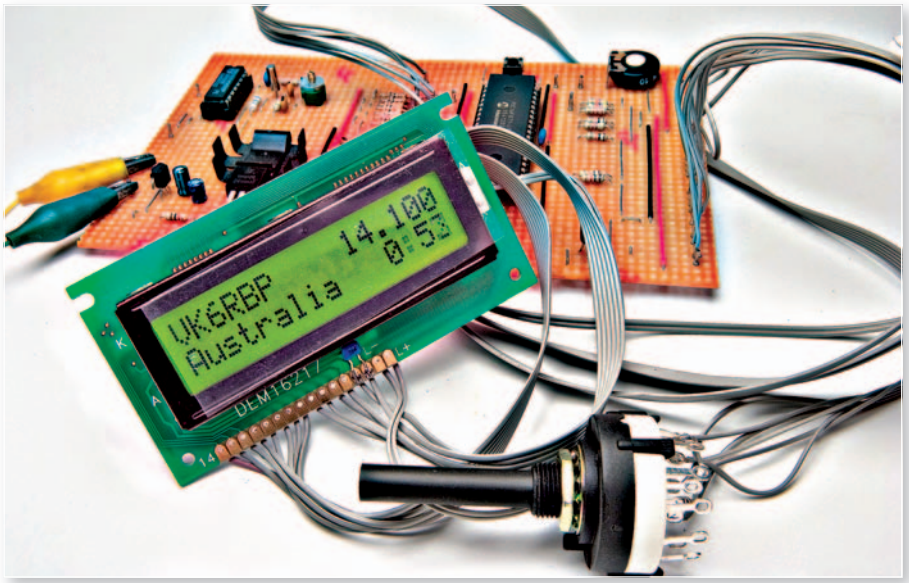


Fig. 7: The completed and working l.c.d. display unit. The beacon shown is the VK6RBP Australian beacon, (Number 6 in the three minute cycle).

More Clock Information

Returning to the l.c.d. clock, the l.e.d. clock's active high/active low control pin (pin 36) clearly has no function when an l.c.d. is involved. Consequently, I decided to make pin 36 control the type of data emanating from the l.c.d. clock's USART.

When tied low – as in Fig. 1 – the l.c.d. clock outputs single bytes for use by the l.e.d. repeater. However, if pin 36 is tied high (+5V), then the USART outputs 'Set Frequency' commands, just like the l.e.d. repeater. (Yes, I know this is confusing!) Use the circuit of Fig. 4 exactly as if you were connecting it to the l.e.d. repeater.

A word of warning! Although the optocoupler makes it extremely unlikely that any physical harm will come to any of the FT-817/857/897 series of transceivers, it's theoretically possible to interfere with their correct operation if things go wrong. Please, always make sure that the CAT baud rate is set to 4800 (the default) before connecting any transceiver to either the l.e.d. repeater or the l.c.d. clock. And also check that pin 36 is tied high when using the l.c.d. clock.

Conditions Improving!

Although we're pretty much at the bottom of the current sunspot cycle at the moment, conditions are going to improve very soon. Even now, day-to-day band conditions can vary greatly and it's really worth checking!

Monitoring the IBP beacons is an excellent way of keeping track of propagation, and maybe working that elusive DX. This is particularly true of QRP operation, where realising a band is opening before the 'big guns' find out, gives low power stations a brief window of opportunity.

Within their individual 10 second time slots the beacons send their callsigns and the first dash at the 100W level, then switch down to 10W, then 1W and finally transmit at the 100mW level. When you can hear the last dash at the 100mW level you will know the propagation conditions between you and that particular beacon are very favourable. I hope these late PIC IBP clocks prove useful. Good DX! ●

Fig. 8: A suitable CAT programming interface. The resistors do not interfere with the process but have leads suitable to fit into the rig's socket.

Note on In-circuit programming

Phil G4JCP writes: While developing the software for this project, I found a problem relating to the in-circuit programming of PIC microcontrollers. The problem occurs when using the Velleman PIC Programmer and Experimentation Board K8048, and I'm sure it must occur with other in-circuit programmers too.

It seems that PICs are very susceptible to glitches on their serial programming clock line - PGC - pin 39. Capacitive coupling between the data and clock lines can cause glitches on the clock line when the voltage on the data line changes very fast. Long leads – especially if twisted together – (okay, I should have known better!) can make the problem worse.

There are two simple remedies that ensure trouble free programming. First, keep the in-circuit programming leads as short as possible (and don't twist them together like I did!). Second, connect a 47pF (or 100pF) disc ceramic capacitor from PGC (pin 39) to the nearest 0V point. I suppose a similar capacitor connected between PGD (pin 40) and 0V wouldn't hurt either. Since connecting such a capacitor, I have had no problems at all, even with relatively long (200mm) twisted wires.

Programs and PICs

Should you feel confident to program your own PIC for this project, then the source code for this project, and the HEX file which is used by the programming software, is available from http://www.g4jcp.freeserve.co.uk/

Alternatively, if you'd prefer to have ready-programmed 16F871 PICs, they're available from the **Kit Radio Company** - see the Advertiser list for details.