PICRONOS L.E.D. WALL CLOCK

JOHN BECKER

Part Two

Ancient and modern techniques display timely brilliance on a grand scale!

LAST month we described the circuit and construction of the *PICronos L.E.D. Wall Clock*. This month we conclude by describing its software and setting-up.

FIRST CHECKS

Having thoroughly checked the board, apply power and immediately check that the output of regulator IC2 is at 5V, within a few per cent, and that the output of regulator IC3 can be varied by VR1 from about 1·5V to 10·6V (assuming a 12V power supply input voltage).

If all is well switch off the power and insert the pre-programmed PIC. Whilst the board has connections (TB1) which allow the PIC to be programmed in situ, brand new PIC16F877s should be programmed for the first time in a PIC programmer, with **no external components connected to other PIC pins**.

This restriction is due to the configuration set for the PIC16F87x family of devices during manufacture. (The same also applies to PICs in the 'F62x family.)

Once the PIC has been programmed, further programming can then be performed in situ on the PICronos board. For instance, you might want to modify some of the author's code to suit your own preferences. Re-programming on-board can then offer considerable advantages.

RING TIME

With the programmed PIC in place and the power on, the two rings of l.e.d.s should be "active". The outer ring should have one l.e.d. turned on, indicating the hours time that the programming currently thinks exists. The timing and correction values used at this time are those last used by the author as data statements in the source code (applies only to PICs that have been programmed with the embedded data statements in the HEX file), and could have any value.

The inner ring should show two active l.e.d.s., indicating the seconds and minutes counts respectively. Apart from the central "colon" no other l.e.d.s should be on.

Check that all l.e.d.s in the inner ring come on at some point during a one minute cycle. If any do not, check that they are inserted the correct way round, and that both their leads are soldered! The same check should be made with the other l.e.d.s as appropriate once the clock is fully running.

Ignore the switches for the moment.

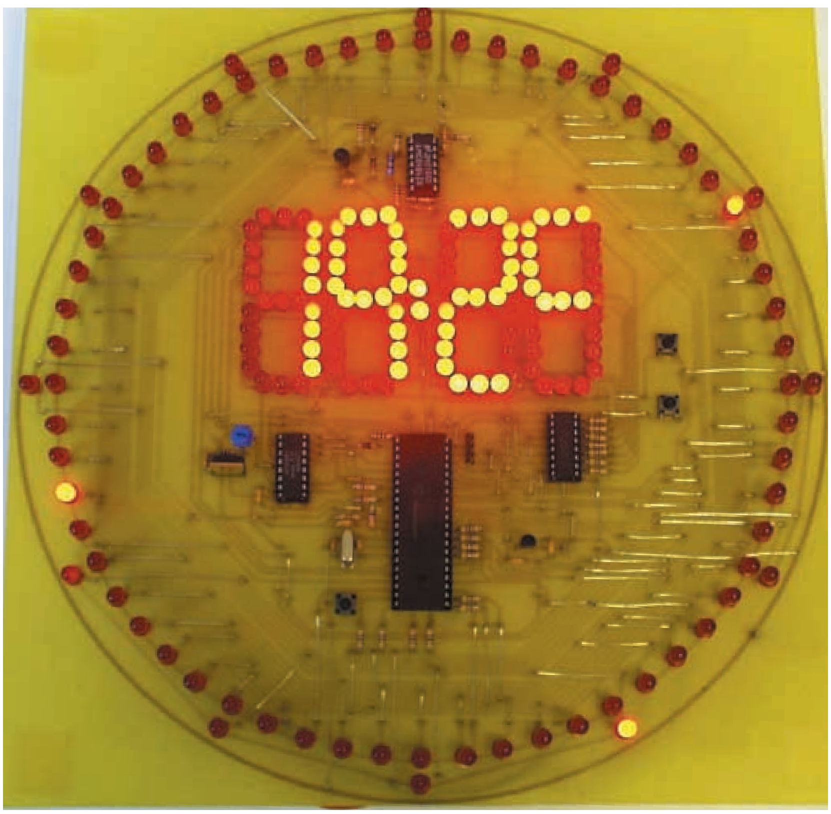
TEMPERATURE

Before checking the digits display, insert temperature sensor IC6 and op.amp IC7. Carefully measure the voltage between the sensor's output and negative (–) pins on a digital multimeter, set to a range suited to monitoring up to +5V d.c.

Note the voltage shown. The sensor has an output that changes by 0·1V per degree Celsius in relation to the voltage at its negative pin. Zero volts between the output and negative pins indicates a temperature of 0°C. Consequently, the ambient temperature experienced by the sensor will be displayed by the reading on your meter in millivolts. Thus, if your meter shows 205mV, the ambient temperature is 20·5°C.

Warm the sensor between your fingers and observe how the meter reading changes, probably upwards.

Now check the voltage at the output of the op.amp at pin 14. It should change in sympathy with the voltage across the two points just measured, but at an amplified value of ×10 per °C. There will, however, be a bias voltage at the op.amp output, typically in the region of about 2·5V. This is due to the sensor's negative lead being raised above 0V by the two diodes, D176 and D177, at about 0·6V per diode.



A mild day, 19·2°C, at 5.10 p.m., 42 seconds and counting!

This displacement is taken care of later through the software, as will be discussed. For the moment, note the output voltage at IC7 pin 14.

DIGITS CHECK

You are now in a position to check the four display digits. With the power still on, first adjust preset VR1 for a the maximum possible output voltage at regulator IC3, and note the value.

Then switch off and insert IC4 and IC5, the interface devices that power the "digit" l.e.d.s.

Applying power again, check that the outputs of both regulators are still at 5V for IC2 and the noted value for IC3.

Observe the digits, checking in particular that all l.e.d.s in each segment come on at some stage. Correct the cause if any do not (with the power off, of course). Check that their brilliance can be changed using potentiometer VR1.

Next observe the specific behaviour of the digits in terms of what they display. A cycle of three values should be seen repeating, changing every five seconds.

The cycle should show hours and minutes in the form hh:mm, date in the form dd:mm, and temperature in the form 20:6^c (the 'c' – for Celsius – occupying segments *a, f, g*, and the colon representing a decimal point).

All four digits will be always be active whenever time is displayed. With the calendar data, any leading zeros of the "tens" will appear blank. The "tens" digit of the temperature display will also be blanked for values less than 10.

While watching the digits, also note that the ring displays indicate the time values, appropriate to the numerical values of the digits when hours and minutes are shown.

The temperature display is likely to show an inaccurate value at this time, although it should still change if the sensor's temperature is changed.

ALIGNMENT

Two paths are provided for aligning the clock values to those that actually exist. The main path is concerned with values that you occasionally have to change with any calendar clock – current time, month and day of month.

The second path allows correction of less likely factors to be made, the accuracy of the clock's time keeping (is it gaining or losing?), the range of the temperature display (is it showing too hot or too cold?), and of the year (although this is never displayed except in correction mode).

The main path is more appropriate to use first. There are several modes within this path and they are selected in order by switch S2. The values displayed within each mode are changed by switch S1. The value only increments (counts upwards), rolling over to an appropriate start value once an upper value has been reached.

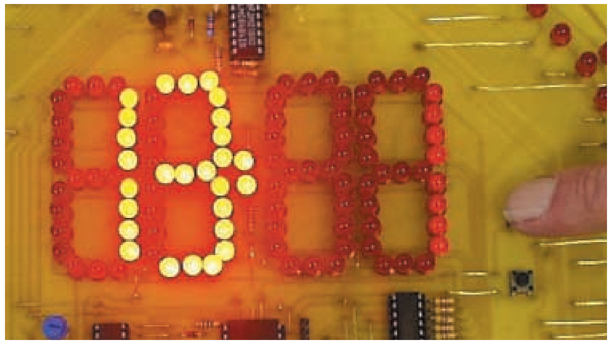
The modes will now be discussed in turn, and in the order that occur for each press of switch S2.

1. MONTH VALUE

When S2 is pressed while the clock is still in the normal time-keeping display, the first correction mode entered is that for changing the month value. This is in numerical form, from 1 to 12. It is not possible to display the months in an alphabetical form with 7-segment displays.

It is only the month value displayed (in the two right-hand digits). The days digits are blanked. The l.e.d. rings are also blanked.

Momentarily pressing switch S1 increments the displayed value by one unit. Holding it pressed causes it to repeatedly increment at a rate of about twice per second. After passing 12 it will roll over to 1. A pause of up to half a second may be experienced between the switch being pressed and a response seen.



Setting the day of the month.

2. DAY OF THE MONTH

When in the previous mode, pressing switch S2 causes the next mode to be selected – days of the month. Again only those digits required for correction are displayed, on the first two digits this time, with the months digits blanked. The l.e.d. rings remain blanked.

The maximum days count for any month is that appropriate to the month in question, including February in relation to leap years (see later).

Pressing switch S1 increments the days count, rolling over to 1 following the maximum for the selected month.

3. HOURS CHANGING

Pressing switch S2 now calls up the hours correction mode. The hours are displayed in the first two digits, while the other two digits (the minutes) are blanked. In this mode the hours l.e.d. ring is made active.

Pressing switch S1 increments the hours digits in a 24-hour cycle, recommencing from 0 when 23 has been passed. The l.e.d. hours ring will also been seen to increment, although it functions on a 12-hour basis, as with a standard analogue clock.

4. MINUTES CHANGING

The next press of switch S2 enters the minutes changing mode. Minutes are displayed in the right-hand digits, with the left-hand digits blanked. The hours l.e.d. ring is blanked, but the minutes ring is activated.

Switch S1 causes the minutes to increment between 0 and 59, rolling over to zero following 59. The count is displayed on the digits and on the l.e.d. ring.

The next press of switch S2 ends the main correction cycle. At this point, the set date and time values are stored to the PIC's data EEPROM, where they remain even if the power supply should cease. They are automatically recalled when power is restored.

Immediately following the data storage, the inner ring of l.e.d.s cycles rapidly through all 60 positions, and continues to do so until switch S2 is released. At that moment the seconds count is set to zero and real-time clock display is resumed, using the newly set values.

Note that in any of the above correction modes, if a value does not need changing, just press switch S2 again to bypass it.

SECONDARY ALIGNMENT

The second alignment path is concerned with three values – the accuracy of the clock's time keeping, the correctness of the temperature display, and the setting of the year value.

This path is activated by pressing switch S3. Within the path's modes, switch S1 continues to increment values when pressed, but switch S2 now causes them to decrement, in other words, a plus and minus count facility now applies.

TIMING ACCURACY

Even though the clock is crystal controlled, its timing accuracy can still drift with time. This is due to low-cost crystals such as used with microcontrollers being subject to a manufacturing tolerance. This factor is not always quoted in manufacturing data sheets, nor is it of relevance in many microcontrolled designs.

It becomes important, though, in a real-time clock situation and so such a clock requires some form of timing adjustment to be included. In some applications a small-value trimmer capacitor may be included with the crystal circuit. It is, though, comparatively easy to implement correction through software in conjunction with plus and minus switches.

The technique used in this design is similar to that used in the author's *Canute Tide Predictor* of June '00. Three counting registers are allocated in relation to the pre-seconds count, and three registers which hold a changeable timing factor. Effectively, it is a fractional value having many decimal places.

The default value for the correction registers is in the binary form of:

10000000 00000000 00000000

that is, three bytes of which the most significant byte (MSB – left-hand byte) holds binary 10000000 (128 decimal),

The PIC's internal Timer 0 (TMR0) is set in software for its pre-count register to divide the clock cycle rate by 1:4 (ADCON0 is set with a binary value of 00000010 during the PIC's initialisation routine).

The PIC automatically divides its controlling crystal rate by four. Each of these subdivisions is regarded as one PIC clock cycle. The pre-count divider is decremented by each clock cycle. Each time the divider rolls over to zero, an interrupt flag is set. The divider is then reloaded with its preset division value (four in this case). The effective rate is 1/400 of a second (400Hz).

In this application, the PIC's interrupt procedure is not activated and the status of the flag is simply read by the software, Each time the flag becomes set, the software resets the flag to zero and decrements a counter which has initially been set to 200. The pre-count cycle then starts again and the process is repeated.

When the "200" counter has reached zero, the three bytes of the above timing value are added to the three bytes of the registers associated with the fractions of a second count.

It will be seen that if these three fractional counters start at zero, then adding the timing value twice will cause bit 7 of the MSB to rollover to 0, i.e. 128 + 128 = 256, which equals 0 for an 8-bit byte (you need to understand binary to follow this argument!). Because MSB rollover has occurred during an adding procedure, the PIC's Carry flag becomes set. If MSB rollover has not occurred, the Carry flag remains unset.

Each time the Carry flag is found to be set, a true seconds counter is incremented. When it rolls over beyond 59, it is reset to zero and a minutes counter is incremented. When it rolls over beyond 59 it too is reset and the hours are incremented. Similarly, the clock and calendar values are incremented and reset as a ripple effect when appropriate through the chain of values.

It will be seen that if the MSB of the timing value is actually set to binary 10000001 (129 decimal), then the additive cycles will cause the Carry flag to become set more quickly than in the previous case, resulting in the clock's timing running faster. Equally, it will be seen that an MSB binary value of 01111111 (127 decimal) will cause the clock to run more slowly.

At the far end of the correction scale, it should be obvious that by changing the value in the least significant byte (LSB – right-hand byte) even smaller fractional corrections can be made. Have a think about what a difference of just one unit between LSB values can achieve in terms of the clock's accuracy – not quite infinitesimally small, but close!

ADJUSTMENT CONSIDERATIONS

Timing accuracy adjustment is the first option offered when switch S3 is pressed. In this mode both digits and seconds l.e.d.s are activated. Although the discussion in the previous section implied that a single bit of the LSB could be affected to change the timing, in practice there is a physical constraint.

The constraint is due to the best rate of response of the plus/minus switches being about half a second. Any faster response could adversely test the response time of the user – resulting in overshoot beyond the value required.

However, a response time of half a second becomes a problem when the value of the *second* byte needs changing by just one unit. The user could well have to sit with a finger on the button for possibly 256/2 = 128 seconds while byte one, the LSB, increments from zero and rolls over to cause the increment in byte two. Tedious to say the least!

This is especially true in the early days of the clock being put into service, when changes to any of the bytes may be required to bring the timing accuracy into range. Ideally, another switch is required for this aspect, but it would have been extremely complicated to implement a fourth switch since all PIC pins are otherwise in use.

Consequently, the value changes in the LSB have been set at eight units change per switch press, resulting in a maximum of 32 changes being required to affect byte two. It is a compromise, but the end result is still a remarkable degree of timing accuracy that can be achieved.

ADJUSTING FRACTIONS

The value of the LSB is displayed via the seconds l.e.d. ring. Prior to activating the required l.e.d. the LSB value is divided by eight and the result is displayed on the appropriate l.e.d., between positions (seconds) 0 and 31.

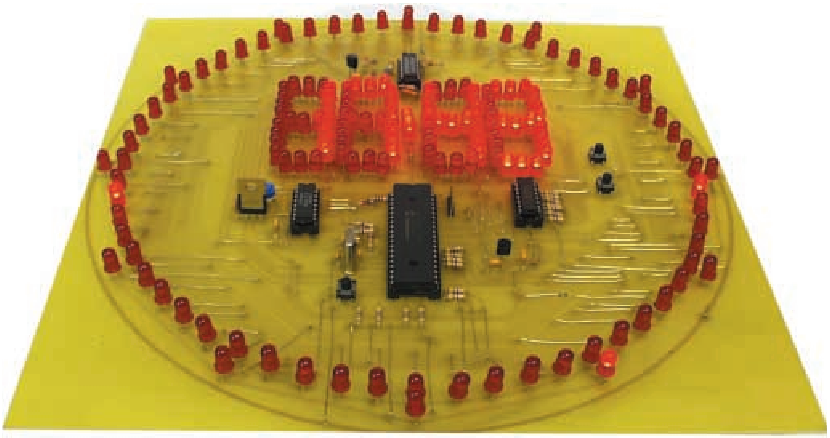
Any rollover of the LSB, upwards or downwards, accordingly changes byte two by one unit.

Putting the effect of this into context, the author's clock when first tested using the MSB set to binary 10000000 (decimal 128) and with the other bytes at zero (i.e. no correction), gained 10 seconds in about 20 hours.

Lowering the MSB value by one unit (to binary 01111111, decimal 127) to slow down the clock caused it to lose about 23 seconds in a similar period.

Progressively adjusting the values in the correction bytes between the upper and lower extremes just stated, the correct setting for the three bytes was established at decimal 127, 127, 250. During this procedure, the clock was typically left to run for about 12 to 20 hours between settings, with much extended periods the nearer the required final value became. Simple maths were also used to estimate the next setting that should be tried in relation to the previous two.

A similar procedure needs to be performed by everyone building this clock, although you may be fortunate enough to be supplied with a crystal whose exact value is nearer to the ideal of 3276800·00Hz.



TIMING ADJUSTMENT

Once the timing accuracy adjustment mode has been entered, the current value of the LSB (divided by eight) will be displayed on the seconds l.e.d. ring as described. In a sense this display can be regarded as the "fraction" of the change in units of the other bytes, whose value is displayed on the digits l.e.d.s.

These values are displayed in relation to a plus and minus value, not the actual byte values just discussed. Zero correction (when the MSB is set at decimal 128, and the other bytes at 0) is just displayed as 0. Each change of the value in the second byte is then represented as a change of units from the so-called zero value represented by bytes one and two, preceded by a minus sign on the left-hand digit for negative values.

The three other digits are used for the numerical value, allowing a correction range of –999 to 0 to 999 to be displayed. The fractional values in byte three are shown on the l.e.d. ring, lying between 0 and 31 as discussed. The author's above correction byte values are thus displayed as "– : 6" on the digits (leading zeros blanked). Ignore the colon's presence in this mode – it does not represent a decimal point.

Switches S2 and S1 increment or decrement the count as required.

TEMPERATURE ADJUSTMENT

Pressing switch S3 while in the timing adjustment mode causes the temperature adjustment mode to be entered. As in the main display, the four digits show the temperature in the form of 20·6^{c}.

Because the amplifier around IC7 provides a gain of ×10 within the tolerance of the resistors (one per cent), it is only the range of the display value that needs adjusting, either upwards or downwards (but note reference to VR2 and R30A in Part One).

Again switches S1 and S2 provide plus/minus adjustment. Calibration of the display can be made in one of two ways. With both of them the unit should have been left switched on for several minutes for its operating temperature to stabilise before making changes.

The first method is to place an accurate (mercury filled) thermometer alongside the sensor. Wait for its temperature reading to stabilise in that position. Using the switches, then adjust the display value until it corresponds with the thermometer reading.

The second method is to monitor the voltage across the sensor's output and minus pins (as done earlier), and then to adjust the display value accordingly.

YEAR ADJUSTMENT

The clock software has been made "millennium compliant", up to the year 9999! To set it for the correct year, press switch S3. On releasing the switch the display will show 2003. It can be incremented (or even decremented) by using S1 and S2 as before.

The years value is never shown when the clock is running "in real time", but the value is used to assess whether February has 28 or 29 days. If you happen to be setting up the clock for the first time on 29th February of a leap year, year setting should be done prior to setting the month and its days. Otherwise it can wait until later.

When satisfied with the years display, press switch S3 again, causing the new correction values to be stored into the EEPROM memory, as before. Again the l.e.d. ring is fully activated and when the switch is released, the seconds count is set to zero and the clock resumes its normal role.

In either of these three correction modes, if a value does not need changing, just press switch S3 again to bypass it.

MODE FREEZING

A final facility has been included with the clock – to "freeze" the cyclic display mode changing on any function, clock, date or temperature. Press switch S1 when the required display is shown, wait about a second, release it and the cyclic change stops. It resumes again next time S1 is pressed.

TIMELY DEPARTURE

It just remains for the author to thank Fernando for inspiring him to design this clock. It has been a challenge in several ways, but principally because of the necessity to maintain high speed multiplexing at all stages without visual flicker.

Readers familiar with PIC programs might care to see how many repeated subroutine calls associated with multiplex maintenance are included, especially in the adjustment and calibration routines. If any of you think you can simplify it all – well, maybe you can, if you "have time on your hands", and good luck if you do try it.

Despite the complexities of achieving this design, it has been time well spent and the end result was worth the effort!

Now, where can this design be hung on the overcrowded walls of the workroom? ...

☐

PLEASE TAKE NOTE

Part 1, June '03, Fig.5, IC5 pins 1 and 9 should be connected to the +5V line, not 0V. The p.c.b. is correct.