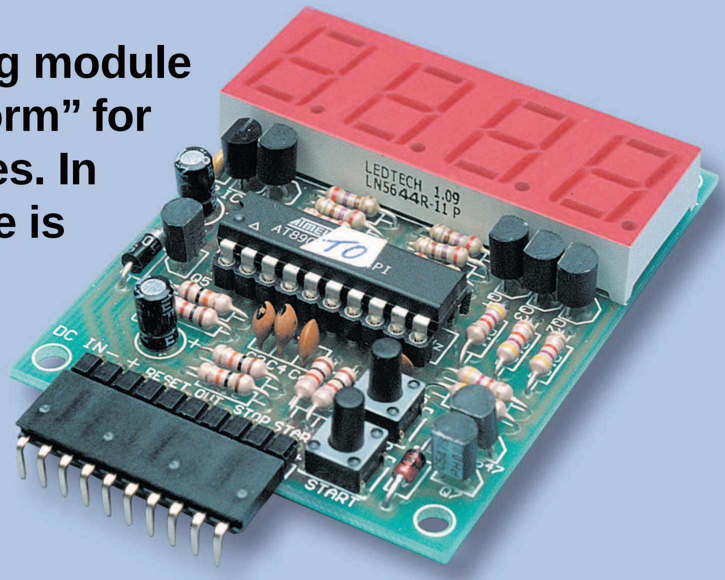
This compact 4-digit timing module forms the hardware "platform" for six different timing modules. In each case, the only change is the firmware programmed into the microcontroller that controls it. Need another type of timer? No problem; just change the microcontroller chip.




# 4-Digit Crystal-Controlled Timing Module

***Just change the chip to build a Stopwatch, a Photographic Timer, a Frequency Meter or a Programmable Down Timer***

THIS SIMPLE LITTLE module measures just 61 x 67mm and is basically a start/stop timer. It's crystal-controlled to ensure accuracy, features an open-collector NPN output and sports a 4-digit LED display.

Currently, there are six timer firmware ICs available. You simply specify which one you want to build. The choices available to you are as follows:

(1) **A Simple Photographic Timer** (K-148T1);
(2) **A Stopwatch with Pause** function (K148T2);
(3) **A 40kHz Auto-Ranging Frequency Meter** (K148T3);
(4) **A Programmable Down Timer** which counts down in minutes from a maximum of 10,000 minutes (K148-T4);
(5) **A Programmable Down Timer** which counts down in hours from a maximum of 10,000 hours (K148T5); or
(6) **A Programmable Down Timer** which counts down in seconds from a maximum of 10,000 seconds (K148-T0).

As supplied, the kit comes with option (6). If you want one of the other functions, the firmware (in the form of a different microcontroller IC) must be purchased separately. The documentation supplied with each option describes how it works.

Please note that, for this design, all source code is copyright and is not released with the firmware.

## Main features

As already stated, the design features a 4-digit 7-segment LED display (with decimal points) plus an open-collector output. Depending on your application, this output can be used to operate a relay or sound a buzzer at the end of the timing period.

In addition, there are three inputs to the circuit: Reset, Start & Stop. The Reset input is a hardware reset to the microcontroller, while the Start & Stop input functions vary according to the firmware used.

All inputs are normally pulled high and may be pulled low by switches or relays, or by an open collector output (ie, when the transistor turns on).

Two on-board pushbutton switches are also connected across the Start & Stop inputs. These enable you to test the basic operation of the timer module without hooking up external hardware (apart from a power supply). Basically, they are there to help you

By FRANK CRIVELLI & PETER CROWCROFT

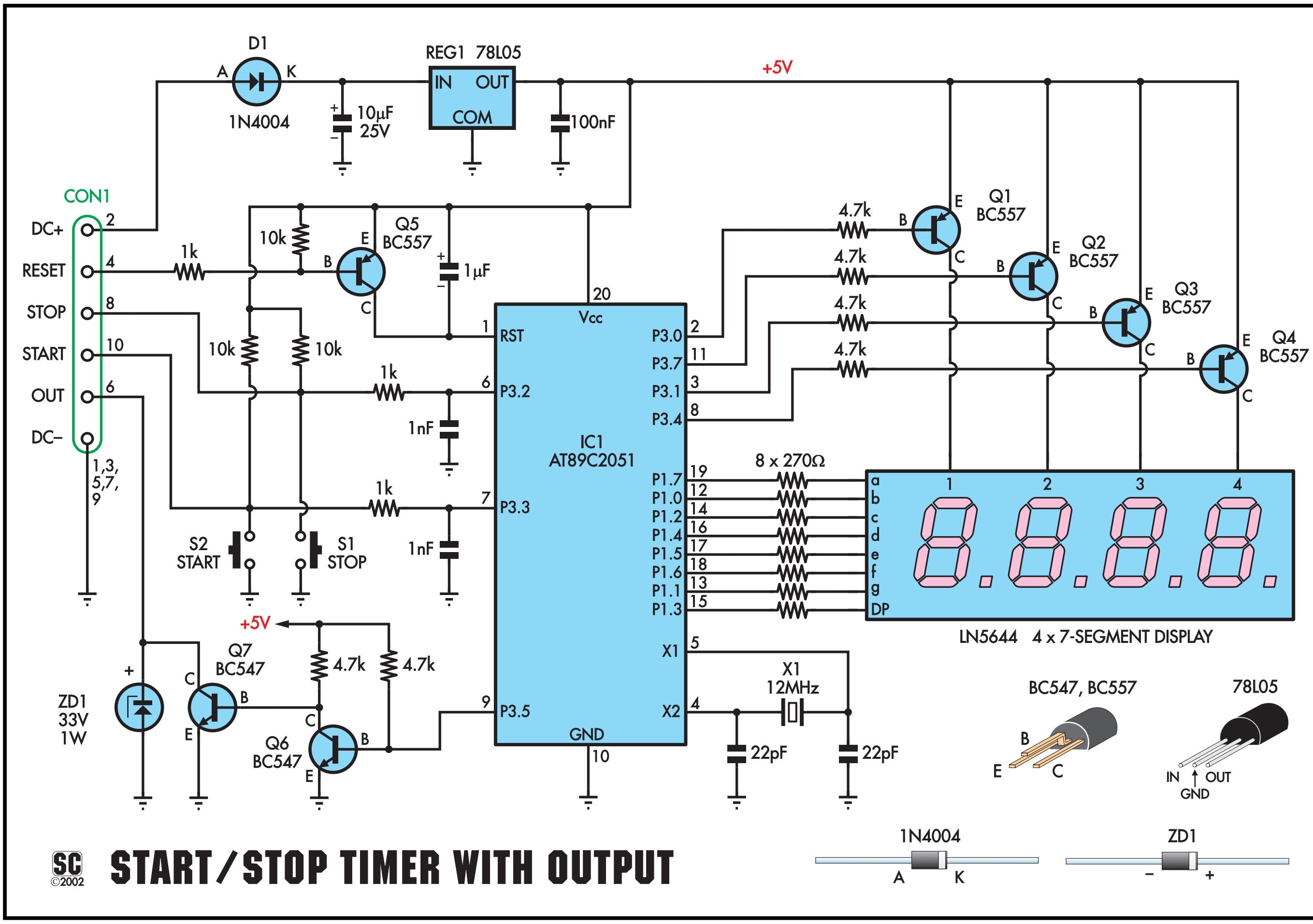



**Fig.1: the circuit uses a single Atmel microcontroller (IC1) to drive a 4-digit LED display in multiplex fashion. Crystal X1 provides the timing, while Q6 and Q7 switch the output line. The circuit function can be altered by changing IC1.**

get the unit "up and running".

To make the module easy to use, all the inputs and outputs are brought out to a single 10-way header pin. What's more, each input or output "pair" includes its own ground pin (see Fig.2).

Note that when using the output to switch a load, this load must be connected between the output pin on the PC board and a positive DC voltage. For example, to switch a 12V relay, connect the relay between the output pin and +12V.

## Circuit details

Fig.1 shows the circuit details of the timer. It uses just one IC – an Atmel AT89C2051 microcontroller. This micro has 2KB of flash programmable and erasable memory and is compatible with the industry standard MCS-51 instruction set. A data sheet can be downloaded from Atmel's website at **www.atmel.com**

The microcontroller IC is preprogrammed to provide each specific timer function. This not only reduces the component count but also allows us to provide more features than are possible using dedicated logic ICs. And the overall cost is much lower.

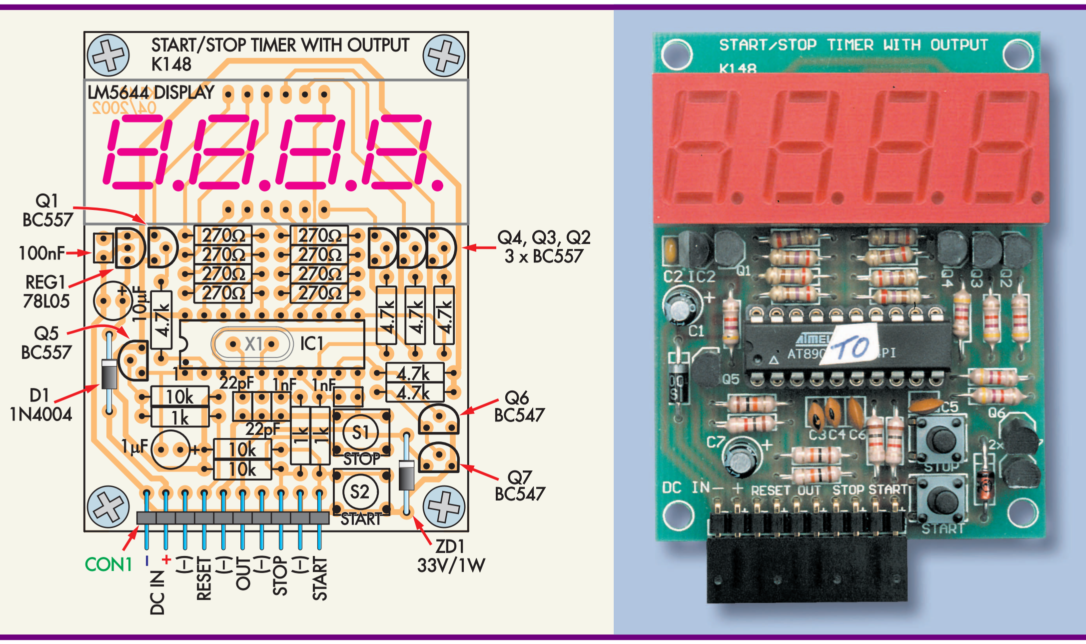
A 12MHz crystal (X1) on pins 4 & 5 provides a stable clock signal. This particular value was chosen because the microcontroller divides the crystal frequency by 12 to produce its own internal clock signal. This gives us an accurate 1μs timebase for elapsed time measurement.

The display is a 4-digit, common anode, multiplexed, 7-segment display (LN5644). This means that all the LEDs in a single digit share a common anode (positive) connection. The cathodes (negative) of each segment (a-g) are connected across the four digits, forming a matrix. This minimises the number of pins needed to drive the display but requires a more

## SPECIFICATIONS

| | |
|---|---|
| **Timing Range (Down Timer)** | 0-10,000 seconds; or 0-10,000 minutes; or 0-10,000 hours |
| **Timing Ranges (Photographic Timer)** | 60, 90, 120, 300, 600 & 900 seconds |
| **Frequency Ranges (Frequency Meter)** | 0-10kHz & 10-40kHz (TTL logic) |
| **Inputs** | Start, Stop and Reset (active low) |
| **Output** | open collector NPN transistor, 100mA @ 30V |
| **Power Supply** | 9-12V DC @ 50mA |
| **Display** | 4-digit 7-segment LED with decimal point |
| **Dimensions** | 51 x 66mm |
| **External Connector** | 10-way right-angle SIL header (male or female) |

Fig.2: install the parts on the PC board as shown here but don't install IC1 until after you've completed the initial voltage checks (see text). Take care to ensure that all polarised parts are correctly oriented.




complex method (ie, multiplexing) to do it.

Multiplexing is a technique whereby each digit is turned on in sequence and then only for a short period of time. What's more, only one digit is on at any given time. In this design, each digit is turned on for 1ms in every 8ms. There is also a 1ms gap between one digit turning off and the next turning on. However, this is all much faster than the human eye can distinguish so it looks like all the displays are constantly on.

This effect is called "persistence of vision".

As shown in Fig.1, pins 13-19 of IC1 drive the display segments (and the decimal point) via eight 270Ω resistors. These resistors limit the maximum current that can flow through each segment. In addition, pins 2, 11, 3 & 8 (P3.0-P3.4) drive PNP transistors Q1-Q4. Each transistor switches the power to its corresponding display digit in response to a low-going signal from IC1.

## Start & stop inputs

The Start and Stop inputs are connected to pins 6 & 7 of IC1 via low-pass filters consisting of 1kΩ resistors and 1nF capacitors. These inputs are normally pulled high via 10kΩ resistors and these resistors, along with the low-pass filters, reduce the chances of false triggering.

Note that the filter time constants are 1µs – input pulses shorter than this don't make it to the microcontroller.

Power-on reset is provided via the 1µF capacitor on pin 1. In addition, the microcontroller can be reset by pulling the Reset line at pin 4 of the header low. This "low" is inverted by PNP transistor Q5 to provide the required high-going reset signal to pin 1 of the microcontroller.

Note that Q5 is normally held off by the 10kΩ resistor connected between its base and the +5V rail.

NPN transistors Q6 & Q7 are used as simple switches to provide an active low, open-collector output. They work like this: normally, pin 9 of IC1 is high and so Q6 is on and Q7 is off. Subsequently, at the end of the timing period, pin 9 goes low and so transistor Q6 turns off.

As a result, Q7's base is pulled high via a 4.7kΩ resistor and so Q7 turns on and pulls pin 6 of the header socket (OUT) to ground.

Note that Q7 is protected by zener diode ZD1 which breaks down and conducts if the voltage across Q5 exceeds 33V. In addition, ZD1 immediately conducts and protects Q7 if any negative voltages are applied to its collector – eg, the back EMF generated when relay coils switch off.

## Why use two transistors?

At first glance you may wonder why two transistors are used to switch the output. Why not eliminate one of the transistors and simply use an active

## Parts List

1 PC board (K148)
2 miniature pushbutton switches
1 12MHz crystal (X1)
2 10-pin IC socket strips
1 10-pin male header
1 10-pin female

**Semiconductors**
1 AT89C2051-24PC Atmel microcontroller, T0 firmware, IC1 (see text for other microcontroller options)
1 LN5644 4-digit, common-anode LED display
5 BC557 PNP transistors (Q1-Q5)
2 BC547 NPN transistors (Q6,Q7)
1 78L05 5V regulator (REG1)
1 33V 1W zener diode (ZD1)
1 1N4004 silicon diode (D1)

**Capacitors**
1 10µF 25V electrolytic
1 1µF 16V electrolytic
1 100nF monolithic
2 1nF ceramic
2 22pF ceramic

**Resistors** (0.25W, 5%)
8 270Ω (red, purple, brown, gold)
3 1kΩ (brown, black, red, gold)
6 4.7kΩ (yellow, purple, red, gold)
3 10kΩ (brown, black, orange, gold)

## Programmable Down Timer (K148T0): How It Works

The microcontroller supplied with the kit is marked "T0" and contains the program for a 4-Digit Programmable Down Timer with output and reset. The timing is in seconds, with a maximum programmable time of 10,000 seconds (0000) – equivalent to 2 hours, 46 minutes and 40 seconds.

The unit has four operating modes that control the output function when the timer reaches zero. We'll look at these shortly.

### Programming

The two buttons marked Start and Stop are used to program the starting time and select the operating mode.

When power in initially applied, the display shows 0000. If you press the Start button at this point, the timer will start to count down from 10,000s so do not do that. If you did, reconnect the power and start again at 0000.

Programming the start value is done one digit at a time, starting with the leftmost digit. The decimal points are used to indicate which digit is being set at any given time. This is always the digit immediately to the left of the decimal point displayed.

Here's the step-by-step programming procedure:

(1) Press the Stop button once to enter programming mode. The leftmost decimal point will come and the display will show 0.000.

(2) Use the Start button to set the value required in the leftmost digit; ie, from 0-9. When you have programmed in this value (eg, 5), press the Stop button again to move the decimal point to the right (50.00).

(3) Use the Start button to program in the value for next digit (eg, 4), then press Stop again to move to the third digit (540.0)

(4) Repeat the above procedure to program the last two digits

(5) Press Stop after setting the units digit. The display will now switch functions to allow the operating mode to be set. Initially, the current operating mode (probably 1) will be displayed.

(6) Use the Start button to set which of the four operating modes you want (see below for a description of each), then press the Stop button. The display will blank momentarily to indicate that programming mode has ended and then indicate the programmed start value (ie, the value it has been set to count down from).

The timer is now programmed and ready to go.

**Starting the timer:** to start the timer, either press the Start button or pull the Start input to ground. The timer will then start counting down towards zero. Note: the Stop button has no affect while the timer is counting down.

**Stopping the timer:** the only way to stop the timer once it has started counting is via the Reset input; ie short the Reset pin to ground. The timer will then reset to its programmed value (the operating mode is not affected).

Note that if the timer loses power, it will restart in Mode 1 with a preset value of 0000 (10,000 seconds).

### Operating modes

There are four operating modes that control the timer and the output (see below). Note that the Reset input does not affect the operating mode.

**Mode 1 – Timer Stop, Output Hold (default):** this is the default mode at power up. The timer stops when it reaches zero and the Output pin goes low and stays low. You then have to press Reset (ie, short the Reset pin to ground) to continue.

**Mode 2 – Timer Overrun, Output Hold:** this is the same as Mode 1 except that the timer continues counting down past zero, wraps around to 9999 and starts counting down from there. The Output pin goes low at a count of zero and stays low. Short the Reset pin to ground to return to the preset timer value.

**Mode 3 – Auto Reset, Pulse Output:** when the timer reaches zero, the Output pulses low for 20ms and the timer resets itself to the programmed value and stays there. You can count down again from the preset value by pressing Start.

**Mode 4 – Timer Overrun, Pulse Output:** same as Mode 2 except that the output pulses low for 20ms instead of staying low. Counting wraps to 9999 and starts counting down. Short the Reset pin to ground to return to the preset timer value.

Once the counter has stopped counting down, you can reset the timer value by pressing Stop and then programming in the time and the mode as described previously.

The hours and minutes Programmable Down Timers (kits K148T4 & K148T5) work in a similar fashion.

high signal from IC1 to switch the output transistor? It's all to do with what happens on reset.

What happens on reset is that the microcontroller's I/O ports are configured as inputs (via internal hardware) and "float" high. If the I/O pin was connected directly to the output transistor, then the output would be "on" during reset. It would then switch "off" after reset as the onboard firmware took over.

In other words, the output would momentarily "flick" on during the reset period – which is not what we want. Using the extra transistor means that we can use a low signal to turn the output on and a high to turn it off, which eliminates any glitches during reset.

The 4.7kΩ resistor on pin 9 of IC1 ensures a "solid" high level signal to

## Simple Photographic Timer (K148T1)

This version of the kit (K148T1) is a simple countdown timer with six preset times: 60, 90, 120, 300, 600 and 900 seconds.

At power up, the default count time is 60s but pressing the Stop button cycles through the other preset time delays.

At the end of the count, the output goes low for 2s and the timer then resets back to the selected time period, ready to start again.

## Auto-Ranging Frequency Meter (K148T3)

The 40kHz Auto-Ranging Frequency Meter (K148T3) measures frequency up to 40kHz over two ranges: 0-10kHz and 10-40kHz. Range-switching is automatic and the gating period is 1s on the low range and 0.1s on the high range.

Basically, a frequency cycle is measured by a high-to-low transition at the Start input of the timer module. For the Atmel microcontroller, a high is defined as 1.2-5V DC while a low is 0-0.9V DC (ie, TTL signal levels).

The display reading is always in kHz, with the decimal point position indicating the range. The maximum reading is 9.999kHz on the low range (1Hz resolution) and 99.99kHz on the high range (10Hz resolution). Note, however, that the maximum frequency that the unit can measure is 40kHz.

The open collector output is "active" when the counter switches to the high range. This output could be used to drive a LED or some other device to indicate that the input frequency is greater than 9.999kHz.

As it stands, the circuit works fine with 5V logic circuits. However, a preamplifier stage (to condition the input signal) will be necessary if you want to measure the frequency of low-level signals; eg, audio signals.

A simple broadband preamplifier that will do the job is shown on the Kitsrus website. It uses just two transistors and a handful of other parts and can easily be built on a piece of stripboard.

turn the output off (ie, Q6 on and Q7 off).

### Power supply

The circuit is powered from an 8-9V DC supply (eg, a plugpack). This is fed to REG1, a 78L05 3-terminal regulator, to derive a +5V supply rail for the remainder of the circuit. Diode D1 provides reverse polarity protection, while supply line filtering is provided by 10μF and 100nF capacitors.

### Timing accuracy

The crystals supplied have a tolerance of ±30ppm, so the actual crystal frequency could vary by as much as 360Hz either side of 12MHz – an uncertainty of ±003%. Over a 1-hour timing period, this amounts to a maximum error of ±0.108 seconds.

However, prototype testing showed that the actual error was more like -1.25 seconds/hour (-0.035%). The factors affecting this include not only the design of the oscillator circuit itself (in this case, a Pierce configuration) but also such variables as temperature and component layout.

It all boils down to this: the assembled unit should be accurate to within ±0.05%, or 1.8 seconds/hour. If possible, do an accurate test over 24 hours (1440 minutes) using the telephone company's time service to determine the number of seconds gained or lost per hour. For critical applications, you can vary the two load capacitors on the crystal to reduce timing errors (say between 10pF and 56pF).

### Construction

This is the easy part, although you do need to have good soldering skills. That's because the PC board pads are quite small and are fairly close to each other. It is recommended that you use a fine-tipped soldering iron and thin solder when installing the parts. Also, don't use too much solder, as this increases the risk of solder bridges between adjacent pads.

Fig.2 shows the assembly details. Begin by installing the resistors (see the parts list for the colour codes), then install the diodes (D1 & ZD1). Make sure that the cathode (striped) end of each diode matches the striped end on the PC board overlay.

Crystal X1 goes in next and this can be installed either way around. Note that it is located between the IC socket pin rows. Make sure that it is sitting flush against the PC board surface before soldering it into place.

Now comes the IC socket. It consists of two 10-pin machine socket strips. This technique was necessary because the crystal would not fit inside a normal IC socket. Solder just one pin first, then check that the strip is sitting correctly in the holes before soldering the remaining pins (the socket strip must be vertical and flush down on the PC board).

The capacitors can now be installed, taking care to ensure that the two electrolytics (10μF and 1μF) are correctly oriented. That's easy – just align each capacitor's positive lead with the "+" sign on the component overlay diagram.

Next, install the transistors and REG1. Don't get these confused – transistors Q1-Q5 are BC557s (PNP types), while transistors Q6 and Q7 are BC547s (NPN types). REG1 is the 78L05 3-terminal regulator. The outline on the PC board shows its orientation (ditto for the transistors).

Push the transistors down as far as possible (without applying excessive force) before soldering their leads. Note that they should all sit lower than the top surface of the display when it is installed – you can temporarily insert the display to check this. This will help later on if you decide to mount the PC board in a case.

Double check that you don't have any solder bridges across the transistor pins, as they are close together.

Finally, install the two pushbutton switches, the 10-way 90° pin header strip (for the inputs and output) and the LED display. Take care with the display orientation – the decimal

## WHERE TO BUY A KIT

Kits and microcontroller ICs for the "K148 Start/Stop Timer" are available from two companies:

(1) Ozitronics – phone (03) 9434 3806 (**www.ozitronics.com**);
(2) Oatley Electronics – phone (02) 9584 3563 (**www.oatleyelectronics.com**).

If you have any technical problems or questions, or if you want slightly altered firmware for a particular application, you can contact the kit developer at **frank@ozitronics.com.** Information on other kits in the range (eg, the Atmel 89Cxxx Programmer, K123) is available from **www.kitsrus.com**

Note: copyright of the PC board and the source code for the Atmel microcontroller is retained by the author.

points go towards the microcontroller.

Note that two 90° pin header strips are supplied in the kit – a male header and a female header. It's up to you as to which one you mount on the PC board for the external connections.

## Testing

Do not install the microcontroller into its IC socket yet – that step comes later, after you have made a few basic voltage checks.

To test the unit, apply power and, using your multimeter, measure the voltage between pins 20 & 10 of the IC socket. You should get a reading of 5V (within a few millivolts).

If this checks out, switch off and carefully insert the microcontroller into its socket (noting its polarity). Check that all the IC's leads go into the socket and that none are bent outwards or under the body of the IC.

Finally, reapply power and check that the display lights. The digits displayed will depend on the specific microcontroller used. In most cases, it will show all zeros.

## Troubleshooting

Poor soldering ("dry joints") is the most common reason for the circuit not working. If you strike problems, the first thing to do is to check all soldered joints carefully under a good light and resolder any that look suspicious. Make sure that there are no solder bridges or "splashes" shorting out adjacent points on the PC board.

You should also carefully check that the parts are in their correct positions and that all parts are correctly oriented. Check that none of the pins have been bent under the body of the IC.

What about the transistors? Q6 and Q7 are NPN types (BC547) while all the others are PNP types (BC557). Did you get them mixed up? Did you confuse the 78L05 regulator with one of the transistors?

Finally, check that REG1's output is at 5V. If there is no voltage at the output of this regulator, check the voltage at its input – it should be at least 8V DC. Anything less and the regulator will not operate correctly.

If there's no voltage here, then it's possible that D1 has been installed the wrong way around – either that or you've inadvertently reversed the supply leads. **SC**