Christmas Project

# TWINKLING STAR

BART TREPAK

*Add a "sparkle" to your festive decorations.*

IF you are tired with the old star which usually decorates your Christmas tree and would prefer something more eye catching, then this Twinkling Star circuit could be just what you need!

The star itself can be bought or made from tinsel stuck onto cardboard or a sheet of polystyrene sprayed gold or silver, or indeed simply your old star if it is suitable. The details of this are left to you, this article concerns itself only with the electronics!

## TWINKLE TWINKLE

The easiest way to add some interest to the star is to simply light it up and for this light emitting diodes (l.e.d.s) are eminently suitable. They are available in a wide variety of colours and a festive display can be made without resorting to lamps and coloured lenses.

A static display is not very interesting and a flashing display is far more attractive. Rather than simply connecting all the l.e.d.s to an oscillator to make them all flash together, a twinkling effect is more effective and desirable. Twinkling, by its nature of course, suggests some sort of random flashing and this could be achieved by building a number of oscillators each driving its own l.e.d. or group of l.e.d.s.

With a 5-pointed star for example, five separate oscillators would be required each running at a slightly different rate. This would produce an overall random effect but it would be obvious that each l.e.d. was in fact switching on and off quite regularly.

With digital circuits, where the output is a logical function determined by the current inputs (and perhaps even previous states), it is impossible to produce a truly random output where the occurrence of the next state cannot be predicted with any certainty. In these circumstances, a pseudo random code generator (PRCG) based on shift registers is normally used. As the name suggests, this is a circuit which produces a seemingly random series of pulses, although on closer inspection the series is found to repeat at regular intervals.

How long the sequence takes to repeat depends on the number of shift register stages used. By having a suitably large number, the sequence can be made random enough for the application. Such circuits are used extensively in the generation of audio white noise and input signals for logic systems for test purposes, as well as in computer games to introduce the element of chance. They are even used in secure remote controls, which send a different code each time they are used, thus preventing the infra-red transmission being copied and used later by an unauthorised person.

## PSEUDO RANDOM

To get an idea of how a PRCG works, it is best to consider the simple arrangement of the 4-stage shift register with XNOR (exclusive-NOR) feedback shown in Fig.1.

Each shift register stage has an output which can be either 0 or 1. With each successive clock pulse, the output of each stage assumes the state of the preceding stage.

The output of an XNOR gate is high when the inputs have equal logic values. If the values are unequal, however, the output will go low. In Fig.1, the inputs of the gate are connected to stage outputs Q3 and Q4. The gate's output feeds into the data input (D) of the first stage. Consequently, input D is continually changing its logic state in response to the logic on outputs Q3 and Q4.

Assuming that initially all the stage outputs are each set at 0, the sequence for the shift register is as shown in Table 1.

**Table 1. Logic sequence for Fig.1.**

| STEP | Q1 | Q2 | Q3 | Q4 | D |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 2 | 1 | 1 | 0 | 0 | 1 |
| 3 | 1 | 1 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 1 | 1 |
| 5 | 1 | 0 | 1 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 | 0 |
| 7 | 0 | 1 | 1 | 0 | 0 |
| 8 | 0 | 0 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 0 | 1 | 0 | 0 | 1 |
| 11 | 1 | 0 | 1 | 0 | 0 |
| 12 | 0 | 1 | 0 | 1 | 0 |
| 13 | 0 | 0 | 1 | 0 | 0 |
| 14 | 0 | 0 | 0 | 1 | 0 |
| 15 | 0 | 0 | 0 | 0 | 1 |

It will be seen that after 15 clock pulses, the outputs again contain all 0s and the sequence will then repeat. Thus, if l.e.d.s are connected to each of the Q outputs, they will be on for the periods marked 1, and off for those marked 0. To a casual observer each l.e.d. will appear to be switching randomly, although with a short

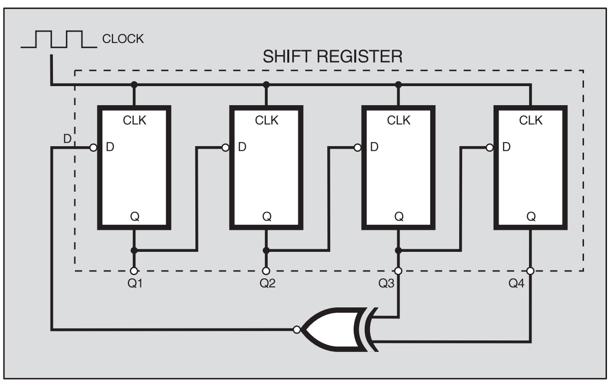
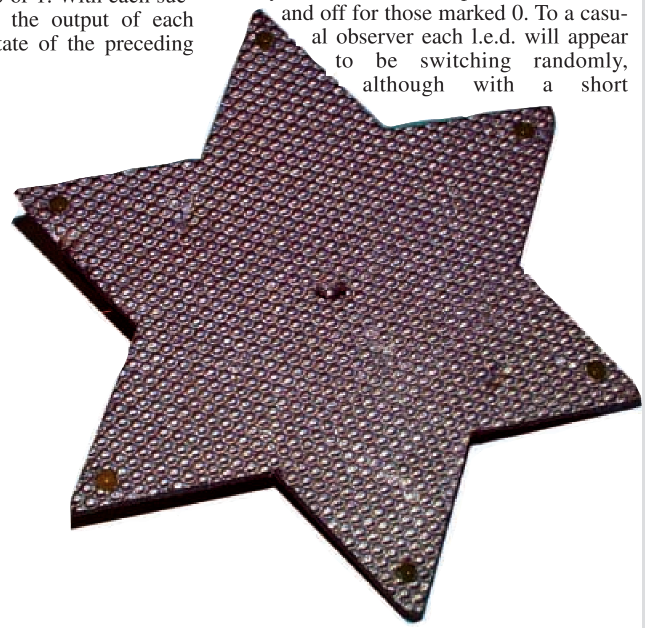



*Fig.1. A four-stage pseudo random code generator based on shift registers and an exclusive-NOR gate.*



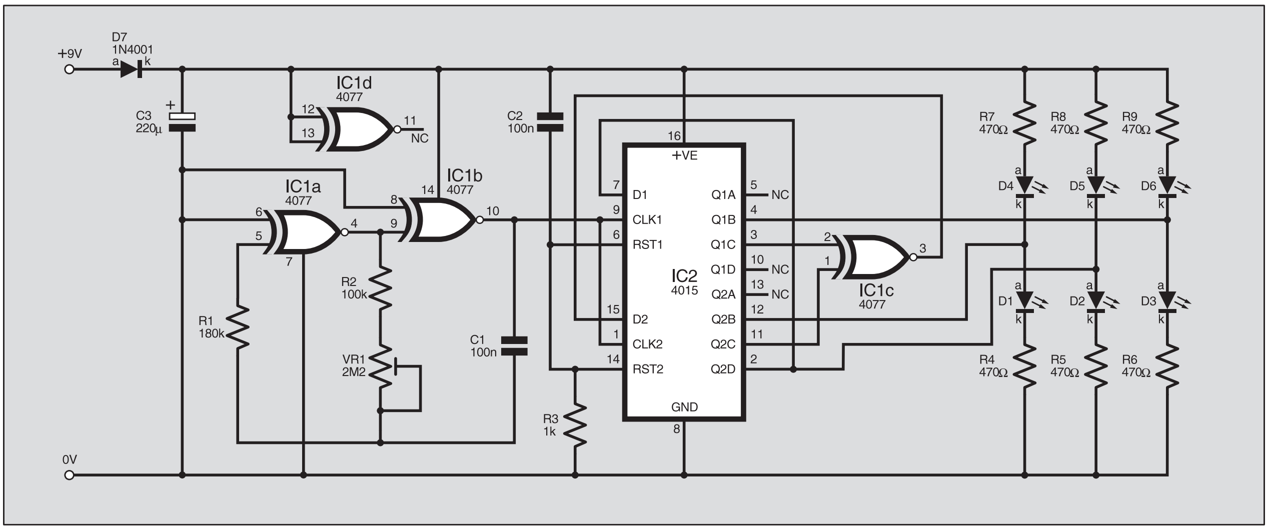



*Fig.2. Complete circuit diagram for theTwinkling Star.*

sequence like this, a pattern may soon be spotted.

The theory of pseudo random generators is quite complex but a few simple rules can be discerned. It will be noticed, for example, that the switching pattern for each output is the same, except that it is delayed by a certain number of clock periods. Thus the second stage is the same as the first but delayed by one clock period, while the fourth stage is the same but delayed by three periods.

In general, the maximum number of states will be given by $2^n - 1$, where $n$ is the number of stages. In the above example where $n = 4$, the number of different states is $2^4 - 1 = 16 - 1 = 15$.

To increase the maximum number of states, the additional stages must be connected "within" the XNOR feedback loop. Adding them on the end of the shift register "outside" the feedback loop would, of course, only repeat the same patterns again.

The above sequence is known as a "maximum length sequence" and to obtain this, the feedback connections must be chosen carefully. If the outputs had been taken from the second and fourth stages, for example, the sequence would repeat after only six clock pulses. However, if the first and fourth stages were used, the output pattern would also happen to be a maximum length of 15.

You can prove this for yourself if you can run the short QBasic program in Listing 1. The length of the sequence can be extended by adding more zeros to the length of A$. The XNOR gate inputs A and B can be "connected" to any Q output by changing the number allocated to them in variables INPUTA and INPUTB. The sequence is stepped through pressing any key.

**Listing 1.**

```
'TWINKLE TEST
CLS : A$ = "0000": B = LEN(A$) – 1
INPUTA = 3: INPUTB = 4
LOOPIT:
IF MID$(A$, INPUTA, 1) = MID$
  (A$, INPUTB, 1) THEN D$ = "1"
ELSE D$ = "0"
IF VAL(A$) = 0 THEN PRINT
  "START SEQUENCE"
PRINT A, A$, D$: A$ = D$ + LEFT$
  (A$, B): A = A + 1
HOLD: IF INKEY$ = "" THEN
  GOTO HOLD
GOTO LOOPIT
```

As the program proves, if more stages are used, the feedback will need to be taken from other outputs to obtain a maximum length sequence. Sometimes even more than two outputs are needed (try modifying the program for this), which then requires the use of a network of XNOR gates.

Thus an 8-stage shift-register, for example, happens to require more than two outputs to be used while a seven stage shift-register requires only two outputs to be connected (1 and 7, or 3 and 7) and produces a sequence of 127 states before repeating.

The mathematics for determining the feedback connections are beyond the scope of this article (or of the simple program) but for those interested, the feedback connections for various length shift registers required to obtain maximal length sequences are shown in Table 2.

Note that only those requiring two input feedback are shown.

**Table 2: Maximum length sequences**

| *Number of Stages* | *Feedback Taps* | *Sequence Length* |
|---|---|---|
| 3 | 1 and 3 | 7 |
| 4 | 1 and 4 | 15 |
| 5 | 3 and 5 | 31 |
| 6 | 1 and 6 | 63 |
| 7 | 1 and 7<br>or 3 and 7 | 127 |
| 9 | 4 and 9 | 511 |
| 10 | 3 and 10 | 1,023 |
| 11 | 2 and 11 | 2,047 |
| 15 | 1 and 15<br>or 4 and 15<br>or 7 and 15 | 32,767 |
| 17 | 3 and 17<br>or 5 and 17<br>or 6 and 17 | 131,071 |
| 18 | 7 and 18 | 262,143 |
| 20 | 3 and 20 | 1,048,575 |
| 21 | 2 and 21 | 2,097,151 |
| 22 | 1 and 22 | 4,194,303 |
| 23 | 5 and 23<br>or 9 and 23 | 8,388,607 |
| 25 | 3 and 25<br>or 7 and 25 | 32,554,431 |
| 39 | 4 and 39 | $5{\cdot}5 \times 10^{11}$ |

## FORBIDDEN STATE

In the previous example, the initial shift-register state was assumed to be all zeros. This was done not only because this was an easily recognised state, enabling the repeating pattern to be more easily seen, but also for a much more important reason. The "all-ones" state would also give an easily recognised condition but this is a "forbidden state" because it would simply cause 1s to be continually shifted along the register and provide no output at all (try it with the program).

In practical circuits this state must therefore be prevented from occurring as it is quite possible that it might occur at switch on.

In this Twinkling Star circuit, however, the problem is side-stepped by using a simple power-on reset circuit.

## CIRCUIT DIAGRAM

The complete circuit diagram for the Twinkling Star is shown in Fig.2.

Two XNOR gates, IC1a and IC1b form an oscillator whose frequency is determined by the values of capacitor C1 and the total resistance of resistor R2 in series with preset potentiometer VR1. The latter is used to alter the frequency.

IC2 is a dual 4-bit shift-register. The two registers are connected in series to produce an 8-bit register. XNOR gate IC1c provides the feedback and is connected to the third and seventh stages of the register to give a total of 127 different output states.

The outputs chosen for driving the l.e.d.s are taken from the second, fourth and sixth stages. These provide exactly the same states but separated in time by two clock cycles so that all the l.e.d.s will appear to be switching randomly.

To ensure that there are no periods when all the l.e.d.s are off, each output drives two l.e.d.s., which require opposite logic states to turn them on and off.

Any colour can be chosen for the l.e.d.s, although it may be necessary to change the values of some of the ballast resistors if a uniform brightness between different colours is required.

As the current consumption is fairly high, the circuit should be powered by a small mains adaptor delivering about 9V d.c. Also see later.

# COMPONENTS

**Resistors**

| | | |
|---|---|---|
| R1 | 180k | |
| R2 | 100k | |
| R3 | 1k | |
| R4 to R9 | 470Ω (6 off) | |

See SHOP TALK page

**Potentiometer**

| | |
|---|---|
| VR1 | 2M2 min. vertical preset |

**Capacitors**

| | |
|---|---|
| C1, C2 | 100n ceramic, 5mm spacing (2 off) |
| C3 | 220µ radial elect. 16V |

**Semiconductors**

| | |
|---|---|
| D1 to D6 | l.e.d., colour of choice (6 off) |
| D7 | 1N4001 rectifier diode |
| IC1 | 4077 quad XNOR gate |
| IC2 | 4015 dual 4-bit shift register |

**Miscellaneous**

Printed circuit board, available from the *EPE PCB Service*, code 276; d.i.l. 14-pin socket; d.i.l. 16-pin socket; 2-pin screw-terminal block, p.c.b. mounting; "star" material (see text); connecting wire; solder, etc.

*Approx. Cost Guidance Only* **£14** *excluding case*

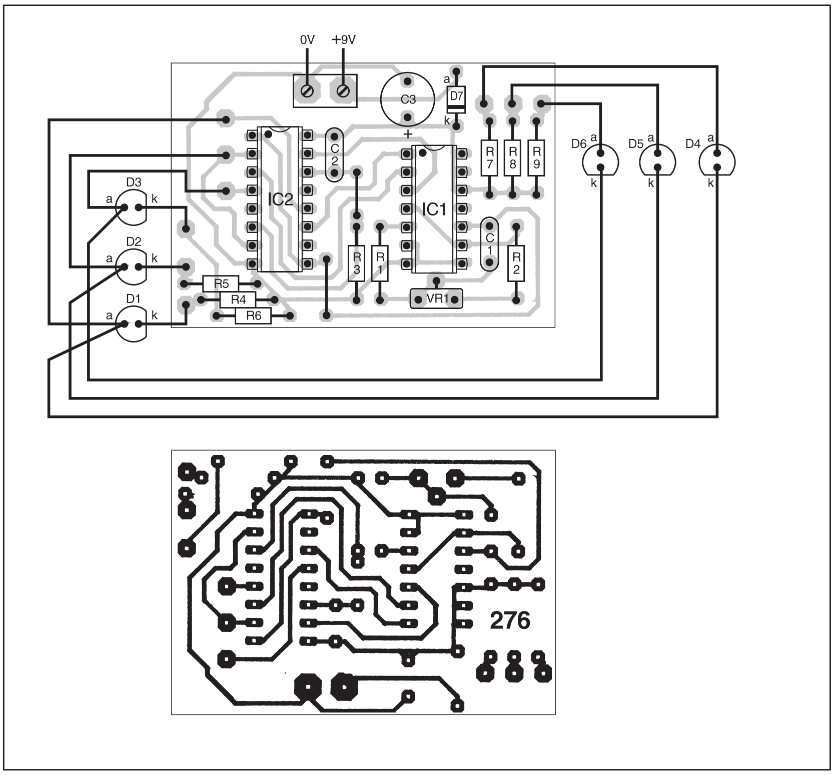



*Fig.3. Printed circuit board component layout, interwiring to off-board l.e.d.s and full-size copper foil master.*

## CONSTRUCTION

The circuit is constructed on a printed circuit board whose details are shown in Fig.3. This board is available from the *EPE PCB Service*, code 276.

Construction should begin with the lowest profile components (i.e. resistors and diode), and then proceed to the capacitors and i.c.s. The latter should be fitted with sockets as they are CMOS devices and prone to damage if mishandled. Touch a grounded (earthed) item before handling them.

In use, the p.c.b. is mounted on the back of the star. This could be cut out from card, polystyrene or some other sheet material on which the l.e.d.s can be mounted. To obtain the maximum flexibility in positioning the l.e.d.s and the size of the star, these are mounted off the p.c.b. and connected to it with suitable lengths of insulated wire.

It is probably best to construct the star first and then mount the l.e.d.s and p.c.b. The wiring details are shown in Fig.3. Care should be taken to ensure that the l.e.d.s are connected the right way around as reversal could cause them not to light.

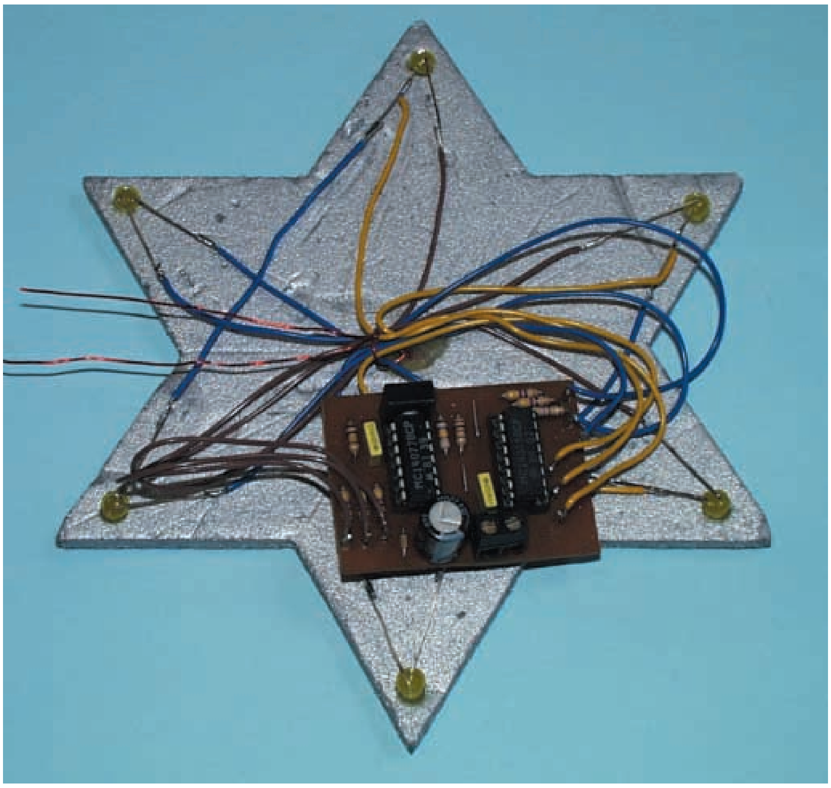
If you choose to use a 12V d.c. supply, two or three l.e.d.s could be wired in *series* in place of each l.e.d. shown. This would enable a larger number of l.e.d.s to be mounted on the star, although all l.e.d.s in the same group would light simultaneously, of course. The values of the series resistors would then also have to be adjusted to maintain a suitable level of brightness, and without overloading the output capability of IC2. The l.e.d.s should ***not*** be connected in parallel.

When complete, there should be a total of nine wires coming from the star for connection to the p.c.b. These should be colour coded so that there is no confusion about which should be connected to the positive and negative (0V) lines.

Provided the circuit has been wired correctly, it should work without any adjustments. Resistors R4 to R9, however, may need selecting depending on the relative brightness of the l.e.d.s used. The speed of the flashing can be adjusted according to taste by means of preset VR1.

Finally, there is no reason why this circuit should not be used with some other kind of tree top ornament, a twinkling fairy, snowman or Father Christmas. □

*Completed prototype "star" with p.c.b. mounted on the back.*



Christmas Project

# CHRISTMAS BUBBLE

OWEN BISHOP

*Keep the party balloons intact, watch light bubbles burst instead!*

THIS project gets its name because it looks like a large bubble, repeatedly swelling up and bursting. The effect is produced by arrays of l.e.d.s arranged in three concentric circles with a jumbo l.e.d. in the centre. First the central l.e.d. lights, then the rings around it light up in order, expanding outward until the bubble finally "bursts'' then swells again.

This is a flexible project that can be "programmed'' for a wide range of effects. There is more than one way to burst a bubble and there are more things to do with any of the l.e.d. arrays. By varying the connections and the layout of the l.e.d.s, all manner of repeating displays may be realised.

Effects can include spinning lights and travelling lights. There is also a spare area of the board on which, for little extra cost, you can set up a more elaborate display or install another complete display driven by the same counter circuit. This simply entails adding two more sets of components repeating those from IC3 onwards.

## HOW IT WORKS

The circuit is quite a simple one, for which the schematic diagram is shown in Fig.1. An oscillator, IC1, generates pulses at about 5Hz. This rate is suitable for producing a lively display. The pulses go to a decade counter, IC2. Its ten outputs repeatedly go high one at a time, in order from 0 to 9.

The next stage consists of four 4-input NOR gates, IC3a/b and IC4a/b, which are connected variously to the counter outputs. If any one or more inputs of a particular NOR gate is high, the output of the gate is low. The output from each gate goes to one of four *pnp* transistors, TR1 to TR4. Pulling low the base (b) of the respective transistor turns it on and current flows to the l.e.d. group, causing the group to light.

For example, as connected in Fig.1, l.e.d. D1 comes on for counts 0 to 4 and goes out for counts 5 to 9. The l.e.d.s in the second group, D2 to D5, are wired in parallel. They come on for counts 1 to 4 and are out for the other counts.

The full sequence for all four groups is shown in Table 1. To make the action clear, it only indicates when l.e.d.s are switched on. They are switched off if not marked.

Other combinations of inputs could alternatively be connected instead to produce other effects, which is why a wide range of lighting sequences is possible with this project.

The breadboard layout in Fig.5 shows the connections required to correspond with Table 1 and the circuit in Fig.1. You may choose different connections if you prefer.

Since different numbers of l.e.d.s are switched, the values of resistors R3 to R10 are chosen accordingly. Table 2 shows the details and selection criteria in relation to a 6V d.c. power supply. The values must be changed if other quantities of l.e.d.s are used should you choose to modify the circuit, or if a different power supply voltage is used.

The groups of l.e.d.s are arranged as shown in Fig.2. All the l.e.d.s in a ring are connected together in parallel. The bubble spreads during counts 0 to 4. The central l.e.d. goes out at count 5. It "bursts'' at counts 6 and 7, briefly reappears in outline at count 8 and is gone again at count 9. The sequence then repeats indefinitely, taking about two seconds for each repeat.

**Table 1: L.E.D. Switching Sequence.**

| *Count* | *Centre* | *Inner ring* | *Middle ring* | *Outer ring* |
|---|---|---|---|---|
| 0 | ON | | | |
| 1 | ON | ON | | |
| 2 | ON | ON | ON | |
| 3 | ON | ON | ON | ON |
| 4 | | ON | ON | ON |
| 5 | | ON | ON | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | ON |
| 9 | | | | |

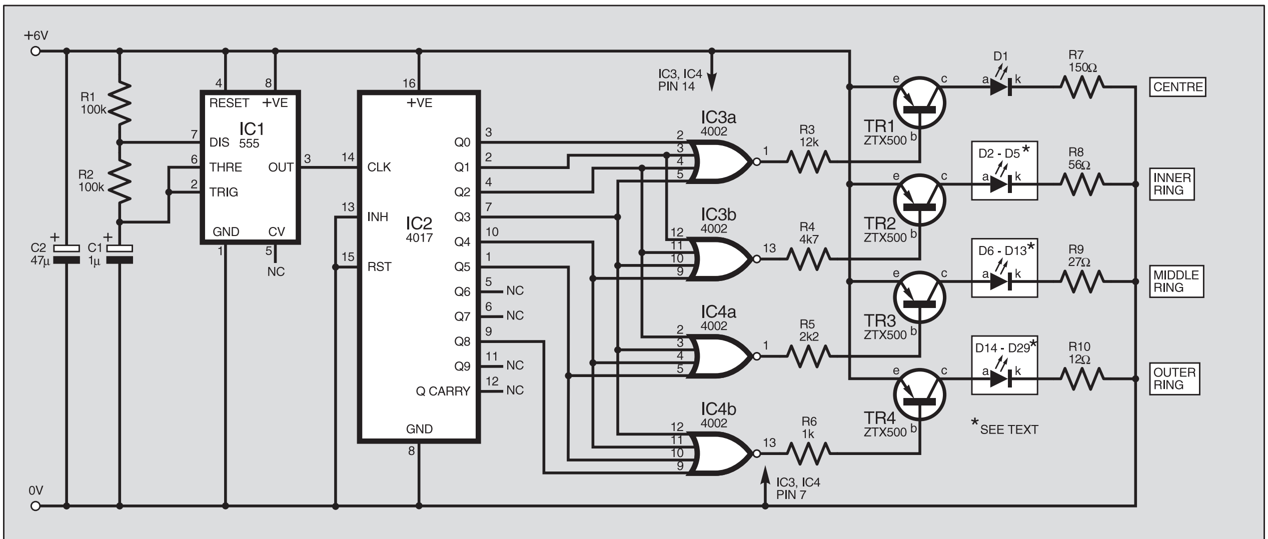



*Fig.1. Full circuit diagram for the Christmas Bubble.*

As the project is a decoration for Christmas and other festivities, it is likely to be run for several hours at a time. It is powered either by a mains-powered 6V d.c. battery eliminator, which should be capable of supplying at least 500mA. Alternatively, use four D-type cells.

## DISPLAY CONSTRUCTION

The display is assembled on a piece of thin card about 120mm square. To make it more decorative, use coloured card or stick coloured paper or cutouts on it.

The l.e.d.s are mounted by pushing their leads through holes in the card (see Fig.3) as far as the lugs will allow. Mark out the circles on the rear of the card and use a stout pin to make pairs of holes about 2·5mm apart, as in Fig.4.

Working on one ring at a time, push the leads of the l.e.d.s through the holes from the front. Make sure that the cathodes (k) of all the l.e.d.s are toward the centre of the circle. Cut each lead to about half-length, then bend it back to touch the rear of the card. Take a length of solid-strand connecting wire and strip off the insulation. Run the wire around the circle leaving a gap as shown.

Tuck the wire under the bent cathode wires and then use fine pliers to kink the cathode wire more sharply to grip the circular wire. Solder each joint, working as quickly as possible to avoid damaging the l.e.d.s. Repeat this operation with another wire to join the anodes (a). Repeat the whole operation for each ring and for the central l.e.d.

## CONTROL BOARD ASSEMBLY

The component requirements depend on the numbers of l.e.d.s used and the patterns in which they are switched. The components list caters for the Bursting Bubble as illustrated in Fig.1, Fig.2, and Table 1.

The control board shown in Fig.5 carries everything except the battery and the l.e.d.s. Assemble the circuit around IC1, for which a socket should be used. Check your connections and then apply power. Confirm that the output from pin 3 is a series of pulses at about 5Hz.

Assemble the circuit for IC2, again using a socket and checking your soldering. When power is applied, each output gives a 0·2s pulse every two seconds.

Next make the connections between IC2 and IC3/IC4, again using sockets for the latter. Link the like-notation points together, e.g. link Q1 at IC2 to Q1 at both IC3 and IC4. Although only one connection per "Q" is shown at IC2, the horizontally adjacent holes may be used as well for the same notation.

**Table 2: Resistor selection detail for a 6V d.c. supply.**

| *Group* | *Number of L.E.D.s* | *Total current (mA)* | *Base resistor* | *Series resistor* |
|---|---|---|---|---|
| Centre | 1 jumbo | 20 | R3 12k | R7 150Ω |
| Inner ring | 4 | 60 | R4 4k7 | R8 56Ω |
| Middle ring | 8 | 120 | R5 2k2 | R9 27Ω |
| Outer ring | 16 | 240 | R6 1k | R10 12Ω |

## COMPONENTS

**Resistors**

| | |
|---|---|
| R1, R2 | 100k (2 off) |
| R3 | 12k |
| R4 | 4k7 |
| R5 | 2k2 |
| R6 | 1k |
| R7 | 150Ω |
| R8 | 56Ω |
| R9 | 27Ω |
| R10 | 12Ω |

All 0·25W, 5% carbon film or better

See SHOP TALK page

**Capacitors**

| | |
|---|---|
| C1 | 1µ radial elect. 10V |
| C2 | 47µ axial elect. 10V |

**Semiconductors**

| | |
|---|---|
| D1 | l.e.d., 10mm, any colour |
| D2 to D29 | l.e.d., 5mm, any colour, any shape (28 off) |
| IC1 | 555 timer |
| IC2 | 4017 decade counter |
| IC3, IC4 | 4002 4-input NOR gate (2 off) |
| TR1 to TR4 | ZTX500 *pnp* transistor (4 off) |

**Miscellaneous**

Stripboard 29 strips x 39 holes; 1mm terminal pins (6 off), 8-pin d.i.l. socket; 14-pin d.i.l. socket (2 off), 16-pin d.i.l. socket; min. crocodile clip; battery holder for four D-type cells; connecting wire (single-strand and multistrand); solder; card and materials for decoration.

*Approx. Cost Guidance Only* **£10** *excluding batts*

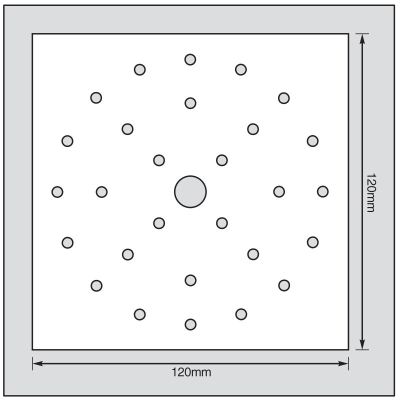



*Fig.2 (above left). Front panel display layout and measurements.*

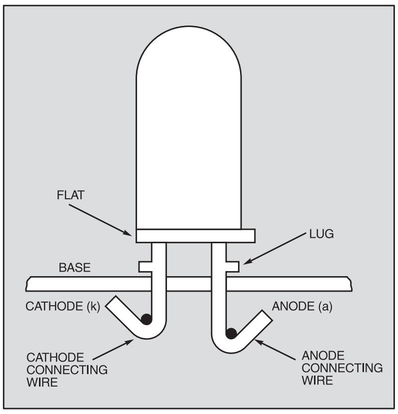



*Fig.3 (above right). suggested method of mounting l.e.d.s on the display wire rings.*

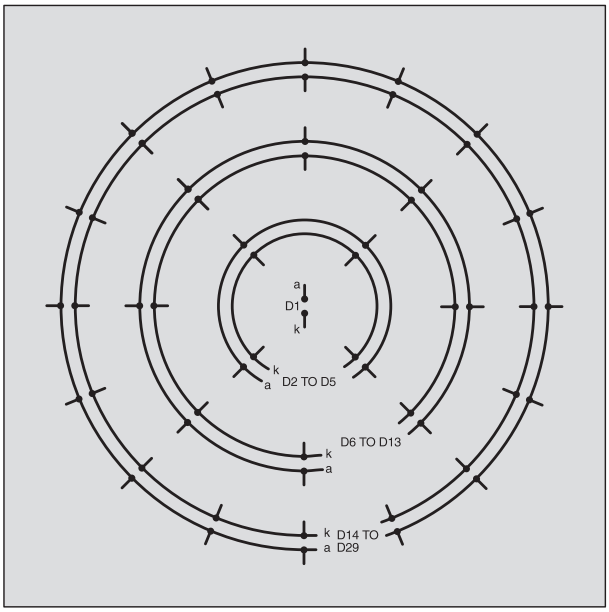
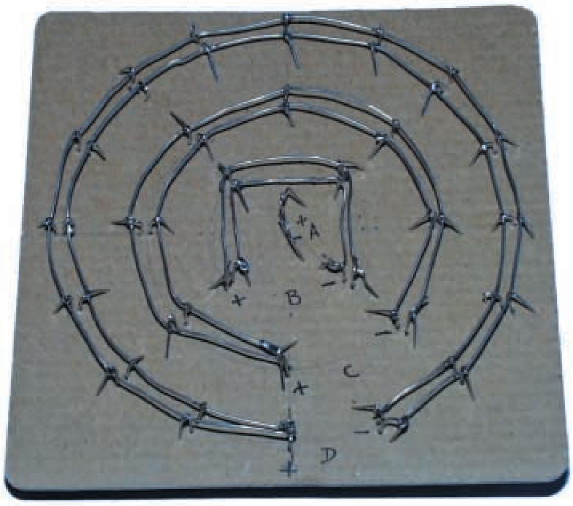
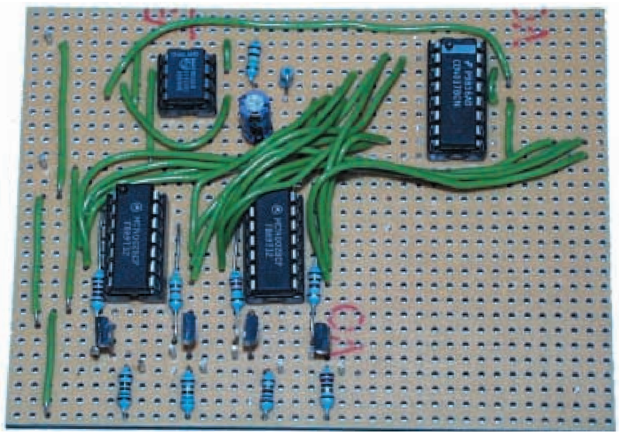



*Fig.4 (right). Connecting up the l.e.d.s in groups.*



The l.e.d.s soldered to the display wire rings.



Completed prototype stripboard component layout.

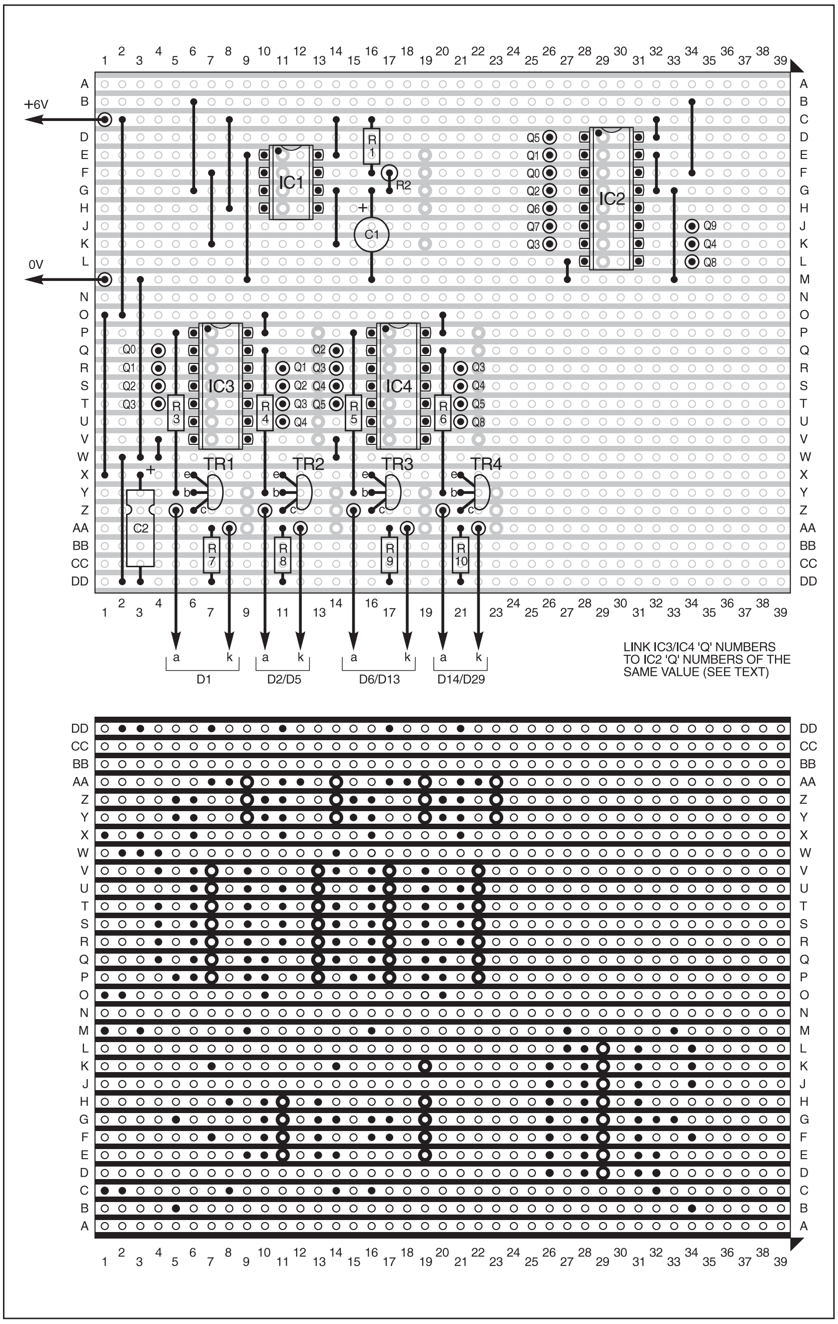



Fig.5. Stripboard component layout and underside copper break details.

Using thin flexible wire, join the ends of the l.e.d. rings to the notated points on the control board. The cathode (k) rings connect to the dropper resistors (pins on row AA) and the anode rings to the collectors (c) of the transistors (pins on row Z), as marked.

You are now ready to view the effect. If any of the l.e.d.s fail to light, check the soldering. Also, check that the l.e.d.s are mounted the right way round, with the “flat” on the rim indicating the cathode pin. If the rows light up out of sequence or at the wrong times, carefully check the connections between IC2 and the inputs of the gates.

A power switch is not really essential for this project. The 0V power line is soldered directly to the negative tag of the battery holder. Make the lead long enough so that the battery can be sited in some inconspicuous place away from the control board and display. Solder a crocodile clip to the positive supply lead. This just clips on to the positive tag of the battery box when power is required.

## WRAPPING UP

The final step is to make the project more presentable as an object to hang on the Christmas tree, over the mantleshelf, or outside the front door. The battery box may be camouflaged by wrapping it with Christmas wrapping paper, to that it looks like a present on the tree. Lodge the “parcel” in the angle of a stout branch.

The control box is hidden behind the display panel and may be attached to it by large blobs of Blu-Tack or a double-sided adhesive pad. Beware of creating short-circuits. To improve the appearance further, make a shallow tray of thin cardboard to mount on the back of the display panel to cover the control board.

There are several different sequences that can be devised with the rings of l.e.d.s as in Fig.4. It is also possible to connect them in other ways. For instance the l.e.d.s of the outer two rings can be re-connected in eight sectors of three l.e.d.s each, with opposite sectors connected. Suitable “programming” would give the effect of a rotating fan.

As said earlier, different connections between the counter and the NOR gates could also be made. Make sure that the soldering of any relocated wires is satisfactory before re-applying power.

A Happy and Scintillating Christmas to all readers! □

Christmas Project

# FESTIVE FADER

STEVE DELLOW

*Relax your Noelistic senses with smoothly changing lighting effects.*

EVERY electronics hobbyist fantasizes about the festive season – it's the greatest opportunity of the year to show off your talents and prove that you can actually build something useful! The reason? The "Christmas Tree Lights Controller", of course! It's your annual chance to dazzle your friends and family with your skills . . . maybe.

Unfortunately, most of the options seem to have been explored by the mass manufacturers. Lights that flash in every conceivable type of sequence and pattern, playing medleys of the most obscure carols ever composed. Not exactly guaranteed to sustain the Christmas cheer.

## DESIGN BASIS

The design concept of the circuit described here was to cater for the other end of the market – something a bit more subtle (if that's possible).

Consider the atmosphere late on Boxing Day when the trifle's just been round for the umpteenth time, and the Best Value multipack of beer is proving a little difficult to dispose of.


As you're finally dozing off, the mother-in-law breezes in and crows "Why aren't the Tree Lights on?". And before you can get a cushion over your head, *BattleStar Gallactica* erupts in the corner!

Bad news! What you'd prefer is something a bit gentler on the senses – a display that gently fades up and down over a period of time to assist in generating a more relaxed atmosphere.

Specifically, the circuit uses automatic phase-angle control to produce the slow cycling of a mains load backwards and forwards between the extremes of fully off and fully on.

In our application here, we have chosen to connect a set of Christmas Tree Lights as the load, but the concept can be extended to a wide variety of applications, providing an interesting insight into how low-voltage electronics can be used to safely control substantial amounts of power.

## MAINS WAVEFORM

To understand the circuit operation, we must first look at the mains waveform which is available to power the load. The mains electricity supply delivers an a.c. signal to our houses, generally in the shape of a sine wave. as illustrated in Fig.1.

The size of the potential difference at any moment in time is directly proportional to how much power we can extract from the supply. As you can see, this varies between the peak when we can get a lot of power, and the zero-crossing points when no power is delivered.

When we switch the lights under normal circumstances we apply the full waveform cycle to our load, and they run at full power. However, in this case, we want to have control of how much power is applied to the load, so we can dim the lights up and down. So in effect we need to have some control over the amount of the mains waveform that is applied.

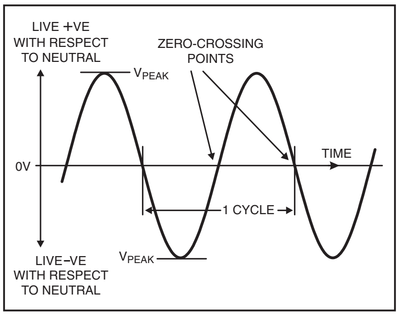



*Fig.1. Mains a.c. waveform.*

The two main ways of achieving this are *burst-fire control* and *phase-angle switching*. Both are employed extensively in industrial and domestic equipment to achieve power control. Burst firing is used mainly for loads with a large "inertia" such as heaters, and lets through the a.c. waveform in bursts of complete cycles. In other words, it applies a measured number of full cycles depending on how much power is to be delivered, see Fig.2.

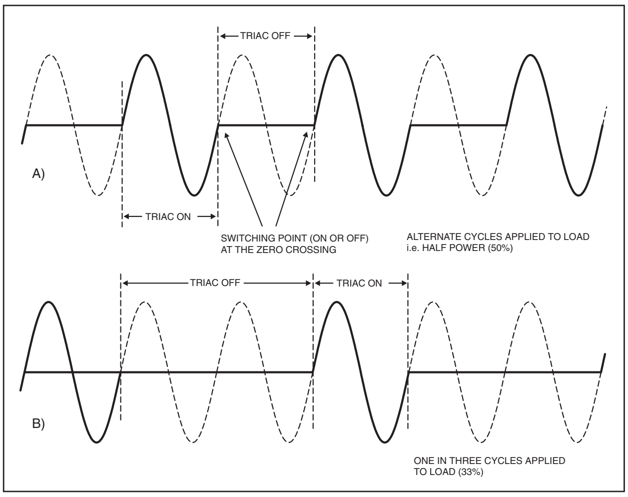



*Fig.2. Burst fire control.*


## COMPONENTS

**Resistors**

| | |
|---|---|
| R1, R6, R13, R15, R16 | 10k (5 off) |
| R2 to R4, R10, R11 | 47k (5 off) |
| R5 | 1k |
| R7 | 510Ω |
| R8 | 51Ω |
| R9 | 330Ω |
| R12 | 220k |
| R14 | 100k |

See SHOP TALK page

**Potentiometer**

| | |
|---|---|
| VR1 | 10k multiturn preset |

**Capacitors**

| | |
|---|---|
| C1, C2 | 100$\mu$ radial elec. 16V (2 off) |
| C3, C6 | 1$\mu$ ceramic |
| C4, C5, C10, C11 | 100n ceramic |
| C7, C8 | 47$\mu$ radial elec. 16V (2 off) |
| C9 | 220$\mu$ radial elect. 16V |

**Semiconductors**

| | |
|---|---|
| D1, D2 | 1N4148 signal diode (2 off) |
| TR1, TR2 | BC109 *npn* general purpose transistor (2 off) |
| CSR1 | TIC206D 600V 1A triac |
| IC1 | 78L05 +5V voltage regulator, 100mA |
| IC2 | 79L05 –5V voltage regulator, 100mA |
| IC3, IC4, IC7 | ICL7621 dual op.amp (3 off) |
| IC5 | LM317L variable voltage regulator |
| IC6 | MOC3020 optically isolated triac |

**Miscellaneous**

| | |
|---|---|
| REC1 | 50V 1A rectifier, d.i.l. |
| T1 | mains transformer, twin 6V secondaries, 3VA |
| SK1 | Euro style fused mains connector |
| SK2 | 13A mains socket, panel mounting |

Printed circuit board. available from the *EPE PCB Service*, code 277; plastic or metal case 190mm x 110mm x 55mm (see text); p.c.b. mounting supports (4 off); mains cable to suit; connecting wire; solder etc.

***Approx. Cost Guidance Only*** **£20** ***excluding case***

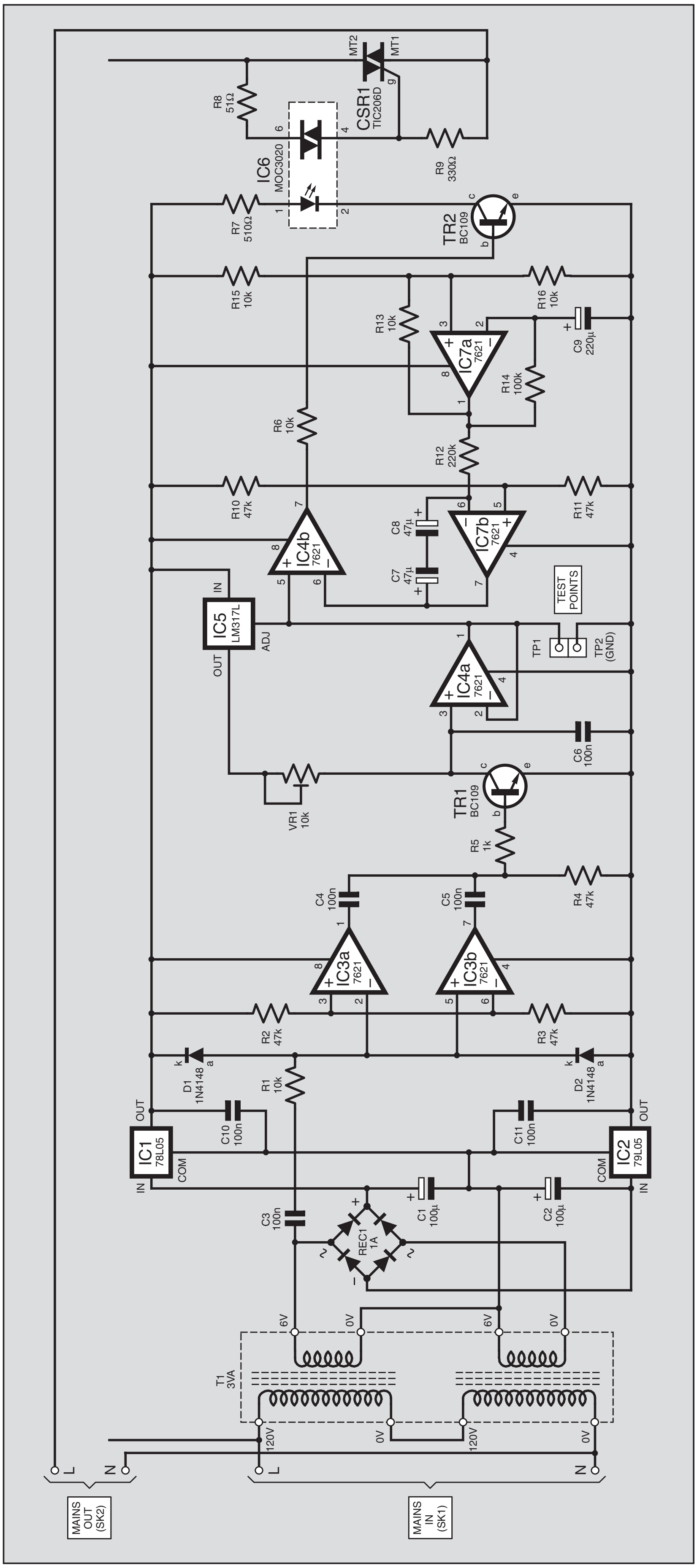
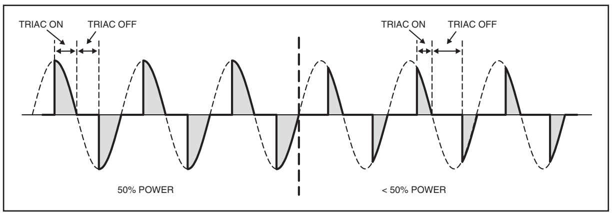


*Fig.3. Complete circuit diagram for the Festive Fader.*



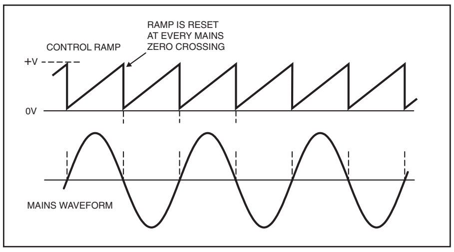




*Fig.4 (above). Phase angle control.*

*Fig.5 (left). Synchronised control ramp alongside mains waveform.*

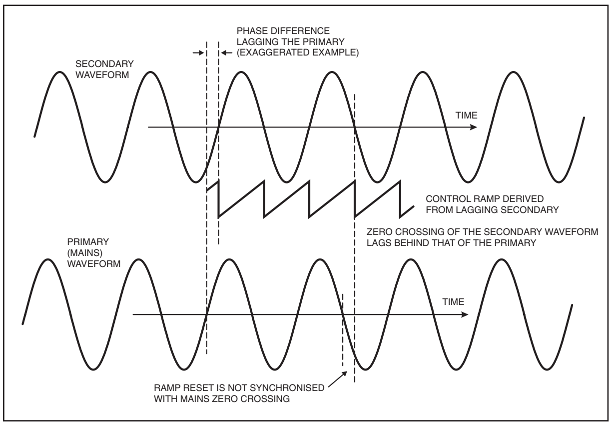



*Fig.6. Transformer phase shift problems.*

The cycle "blocks'' always begin and end at zero-crossing points to reduce switching interference to a minimum. This is fine for large loads but not so for lights, which tend to flicker annoyingly when controlled in such a way – not what we really want.

The alternative is phase-angle control which "chops'' up the a.c. waveform, as illustrated in Fig.4. By choosing the point where we apply our "chop'' we can accurately control how much energy is passed through into the load, and so we have a means of controlling the brightness of our lights.

There is a risk of interference with this style of switching, but the lights are generally quite low power systems, so it's not normally a problem.

## CIRCUIT DESCRIPTION

Referring to the Festive Fader circuit diagram in Fig.3, power for the system is taken via a small mains transformer, T1, a 3VA type being sufficient to supply the circuit's needs. The secondaries are joined together in series and fed through a full-wave bridge rectifier, REC1, to convert the a.c. signal to d.c. for the internal power supply.

This raw d.c. is smoothed by capacitors C1 and C2 to give about 9V across each component. The voltage regulator configuration that follows the smoothing capacitors may be a little confusing at first – creating a +10V d.c. supply, but using two regulators to do it! Why not use just a single one and save money?

The reason is that it is necessary to maintain everything symmetrical about the centre tapping of the transformer secondary, so that our zero voltage detection system works.

Since we are using phase angle switching, we need to have a reference that tells us exactly where we are in the mains signal to permit accurate switching of the mains-controlling triac (CSR1).

The chosen method here is to create a rising voltage ramp (or sawtooth) that starts from 0V at the beginning of each half cycle of the mains. This ramp rises linearly until it is reset at the end of the period – see Fig.5.

This will give us the means to control the lamps from fully off to fully on – if we switch the triac on at the far left hand end of the ramp, the lamps will get most of the energy from the associated mains cycle i.e. be very bright. However, if we switch them when the ramp is a lot "higher'', then less energy is delivered and the lamps are proportionally dimmer. It's so easy!

## ZERO CROSSING

To start (and reset) this synchronising ramp we need to find the zero crossing points of the mains waveform. There are a number of ways of doing this, and a large proportion involve making direct connections to the primary of the transformer. This results in there being a number of "live'' components on the circuit board – which can be quite hair-raising when you're trying to do fault finding!

The argument in favour is that the mains signal in the secondary windings on most cheap transformers cannot be guaranteed to be in-phase with that of the primary. This means that the ramp would be "shifted'' in time with respect to the actual mains signal, and we wouldn't be switching things at the right moment, see Fig.6.

To understand the information in Fig.6, read it in the following manner:

1. Look at the primary waveform
2. Compare it with the secondary waveform (note that it lags behind the primary)
3. The control ramp is derived from the secondary waveform and its zero crossings
4. Compare the ramp reset points with the zero crossings of the primary waveform
5. Start again.

## SAFETY RAMP

A "Safety First'' approach has been taken with this design, and the ramp waveform has been derived from the transformer secondary. Consequently money is saved on mains-rated components, not to mention rubber gloves! If there does appear to be a phase shift problem there is a simple modification to the circuit to cure this.

A sample of the a.c. signal from the secondary is coupled through C3 and R1, and clipped by diodes D1 and D2 to protect the inputs of op.amps IC3a and IC3b (in Fig.1).

If the signal at pin 2 of IC3a is displayed on an oscilloscope (with reference to 0V), you will see that the waveform is sitting nice and symmetrical between the supply rails. This has come about as a result of the split regulator arrangement, which has made sure that the zero crossing points occur as close as possible to the half supply voltage (5V).

We now make use of the two op.amps IC3a and IC3b to create a zero-crossing detector. The op.amps are both configured as simple voltage comparators. When the non-inverting (+) input is higher than the inverting (–) input, the output is near the supply rail (10V), and when the conditions are reversed, the output switches to ground (0V).

By setting the switching threshold at half the supply voltage, we can use these comparators for zero-voltage crossing

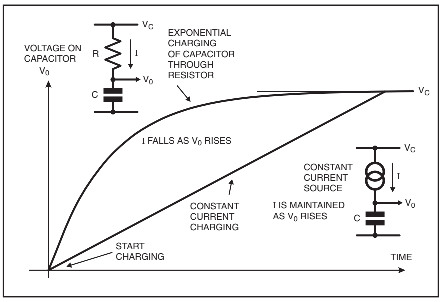



*Fig.7. Constant current charging of a capacitor.*

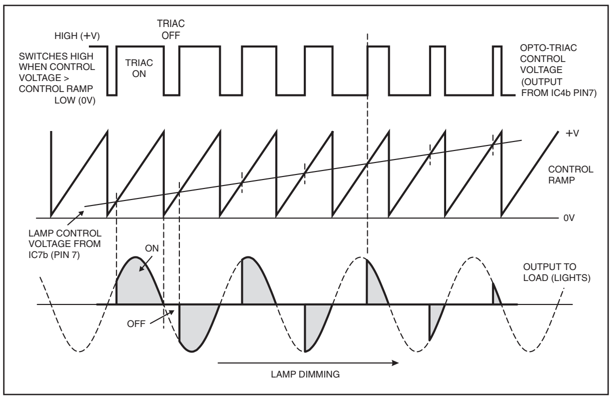



*Fig.8. Opto-triac control voltage.*

point detection. Their outputs switch between "high'' and "low'' every time the input signal crosses the threshold (which is effectively the zero-crossing point) and produce a square wave signal at the outputs.

If you look carefully at the connections to the inputs, you will see that this results in the two generated square waves being in anti-phase. When the signal voltage rises through the threshold, IC3a output switches high, but IC3b output goes low, and vice versa when the signal input falls back through the threshold.

So, having created a zero-crossing detector circuit we need to consider how it can be used to synchronise the timing ramp.

## RAMP SYNCHRONISATION

The ramp is created by charging up a capacitor under controlled conditions. Generally, we tend to charge up these devices through a resistor connected to a fixed voltage source. This produces the classic exponential curve, because the charging voltage across the resistor falls as the voltage on the capacitor rises. In other words, as time progresses, there's less "push'' to force more charge onto the capacitor. See Fig.7.

However, if we were able to maintain a steady push then the voltage would rise, not in a curve but in a straight line. This steady push is known as a constant current, and is created by adding circuitry that maintains a fixed voltage across the charging resistor. In this circuit, op.amps IC4a and IC4b plus variable-voltage regulator IC5 and preset VR1 are used for this task.

They combine to produce a constant current to charge capacitor C6, and the result is a rising voltage ramp, the height of which is ultimately limited by the supply rail. Op.amp IC4a is wired simply as a unity gain buffer to make sure that the charge (and therefore voltage) on C6 is undisturbed by the operation of the regulator.

The process is relatively simple. Regulator IC5 is used to create a constant voltage across VR1. Because there is a constant voltage across this resistance, there must be a constant current through it. This current pushes charge into C6 and as it does so, the voltage on the capacitor rises.

The regulator responds to this and maintains the voltage across VR1 so that the current is kept flowing at the same rate. The resistance is made variable to allow the ramp to be optimised, and a test point is included to allow it to be displayed on an oscilloscope.

We have created a ramp, then, but we need to reset it to 0V at the start of each mains half-cycle. This is where we bring in the zero-crossing signal created earlier. A transistor is connected across the ramp capacitor, C6, and by switching this on very briefly at the zero-crossing point, the charge on the top of the capacitor is removed, thus returning the ramp to 0V.

The transistor switching signal is created by differentiating the waveforms from the outputs of IC3a/b. Nice, short switching pulses are produced by the action of capacitors C4 and C5, which allow only the a.c. component of the signal through, i.e. when the voltage changes. The transistor responds only to the positive going pulses and the result is a good, clean reset.

## FADE CONTROL

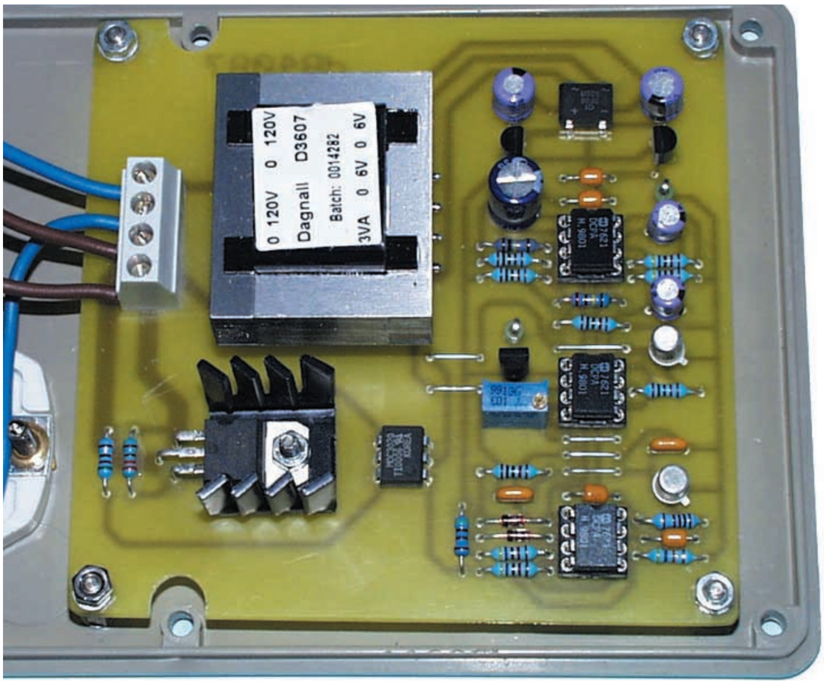
Now that we have a synchronised ramp, the next step is to create some sort of control signal that will command the triac to fade the lights up and down. The simplest method has got to be voltage control – when the voltage is at one extreme, say 0V, we want the lamps to be fully on, and at the other, fully off.

This works in well with the ramp voltage, which is also moving between the supply rails. Lighting experts may at this juncture remind us that the brightness of an incandescent lamp is not linearly related to the point at which you switch it on during the mains cycle. It's not even a simple matter of doing a mathematical integration, as there are other physical effects that come into play.

For this application it was decided to keep things simple, and the "error'' is not really noticeable unless you're sitting right under the Christmas tree with a light meter!

Op.amps IC7a and IC7b are used to produce a slow linear ramp which moves backwards and forwards between 0V and the supply rail – it's really a triangle wave. The rate is set so slow that the signal actually spends time clamped to supply or ground at the top and the bottom before it starts to move back again. These are the points where the lights would be fully on or fully off.

The first op.amp, IC7a, is set up as a slow square-wave oscillator. You can see the output switching simply by using a voltmeter. The rate is set by resistor R14




*Finished circuit board. It is essential to use an insulating kit to mount the triac heatsink on the p.c.b.*

and capacitor C9. Their values could be modified to speed things up a bit, or slow them down even further!

The second stage (op.amp IC7b) operates as an active integrator, turning the square wave input into a ramp. Here, the total value of capacitors C7 and C8 in relation to the value of resistor R12 control the gradient of the ramp, i.e. how fast the lights fade.

## TRIAC CONTROL

The final step is to apply a control signal to the triac, and we return to the comparator function of an op.amp to help us here. By applying the zero-crossing ramp and the control voltage to the inputs of op.amp IC4b, we produce a switching waveform, as shown in Fig.8.

Whenever the half-cycle ramp is greater than the control voltage, the op.amp's output switches high and transistor TR2 turns on. This turns on the light emitting diode (l.e.d.) inside the opto-triac package of IC6, which triggers the integral triac into conduction.

The current capability of IC6's triac is limited, so an external device with greater capacity is added to cope with a real load such as our lights. The specified triac will cope with loads up to about 1kW.

So, if the triac is conducting, this means that mains current will flow – and the lights will be on; but only for that half-cycle! When the mains goes back through zero, the triacs stop conducting – so we have to trigger them *every* half cycle. This is why we keep resetting the ramp – we have to keep retriggering the triacs to keep the lights working! Consequently, the circuit is kept very busy while we relax to the gentle fading of the lights . . .

## CONSTRUCTION

Since there are mains connections involved in this design it is strongly recommended that the published printed circuit board (p.c.b.) is used rather than stripboard (which isn't voltage rated for such work anyway). The p.c.b. assembly and tracking details are shown in Fig.9. The board is available from the *EPE PCB Service*, code 277.

**This circuit should only be constructed by those who are adequately familiar with constructing mains operated circuits.**

Before beginning any construction work, get yourself well prepared! Make sure the soldering iron tip is nice and clean, all your tools are close at hand, then find a really quiet spot away from any likely disturbances. Close reference to the circuit layout is essential – time spent here will save much grief later on.

Start by soldering in the wire links – there are six of them in all – plus the two test points. Then move on through the following sequence – resistors, capacitors, bridge rectifiers, and diodes. Then onto the transistors and regulators taking care to get their orientation absolutely right.

If it's all looking good, fit the transformer to the board – which may need a firm push to get its pins snugly down through the holes.

## FIRST TEST

Test that the power supply is up and running before fitting the more expensive bits! First, make a careful check of all your soldering – look out for solder splashes, dry joints, solder bridges between tracks, etc. Once again, time spent here can save money . . . use a good magnifying glass to be absolutely sure.

When you've finished, carefully wire up a fused mains supply (one that's Residual Current Circuit Breaker (RCCB) protected is best) to the inputs marked L and N (Live and Neutral), then make sure that the board is firmly supported on an insulating surface.

Keeping well clear of the live parts, switch on. The check here is to confirm that the 10V d.c. supply is correctly running – there are a number of points in the circuit where this can be measured using a multimeter. Be sure that you know where you're putting your probes before you try

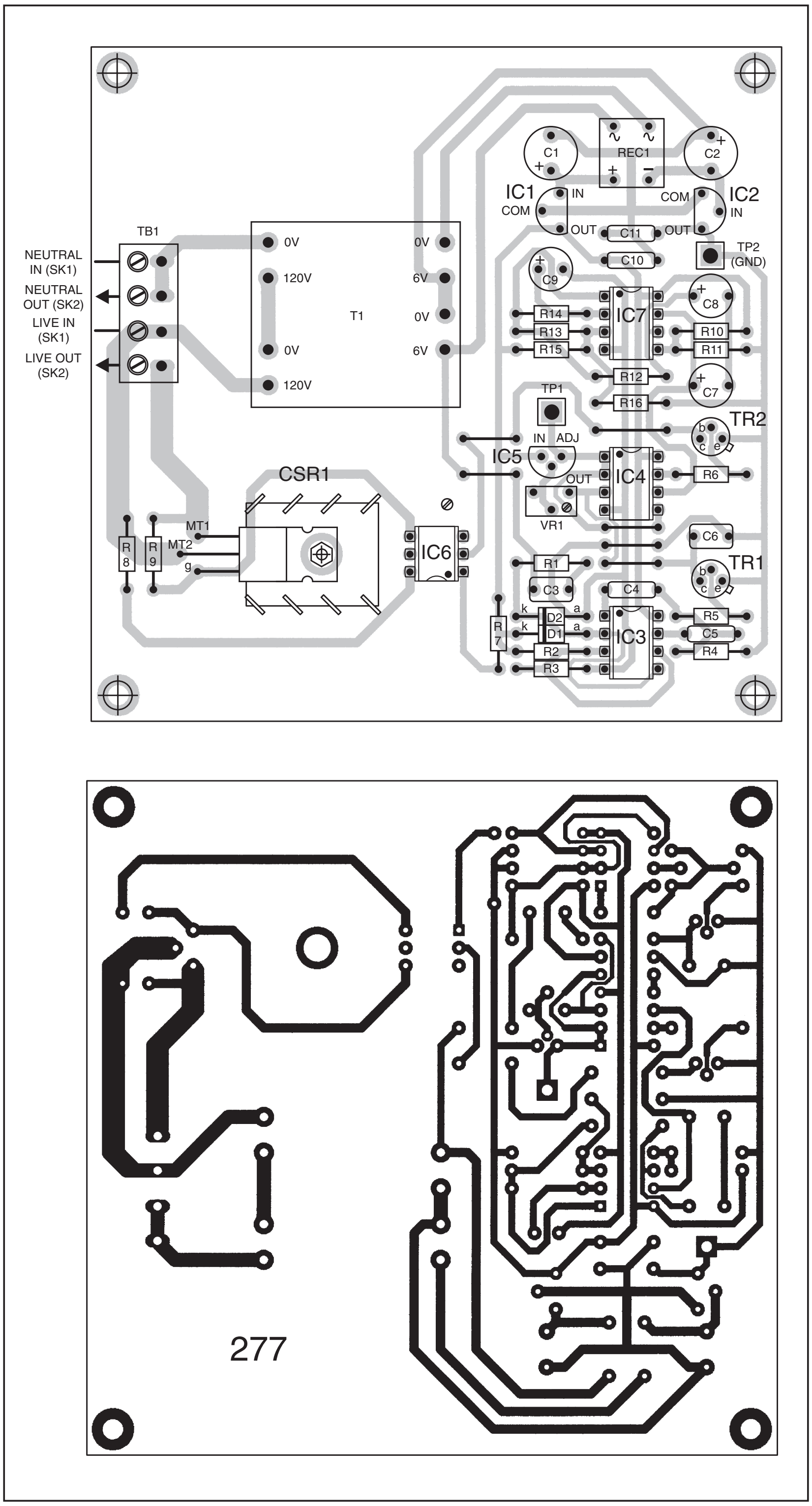


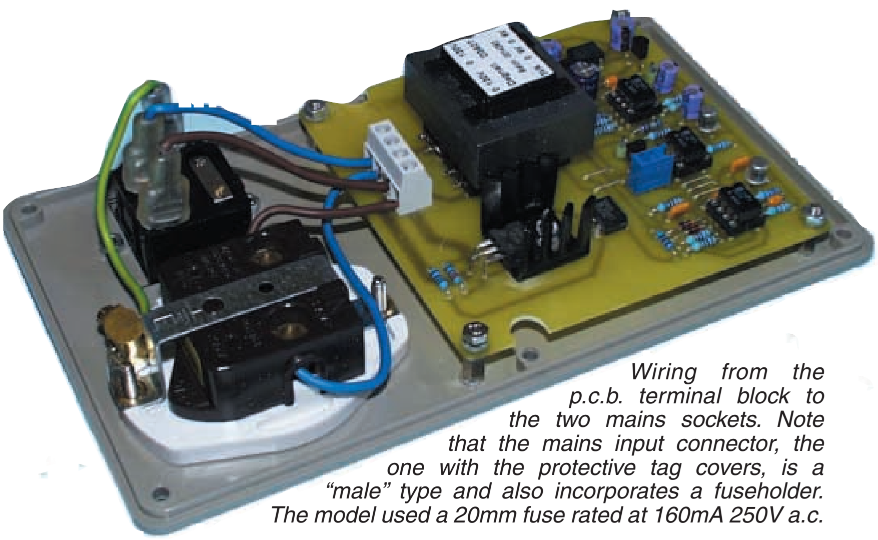

*Fig.9. Festive Fader printed circuit board component layout and full-size underside copper foil master pattern. Beware some tracks carry the mains supply.*

and make a measurement. Always keep your other hand in your pocket to avoid short-circuits through your body.

## TRIAC MOUNTING

Once you're confident that everything's getting power, switch off and disconnect the mains connections completely. The integrated circuits (IC3 to IC7) and triac CSR1 can now be soldered into the board. As usual, check carefully that you're putting them in the right way round.

With mains loads up to 60W, the triac can be mounted vertically into the board, but if heavier loads are planned, there's space for it to be bolted down with a small TO220 heatsink. In this case, it is essential that you use an insulating kit, otherwise the heatsink body will be at mains potential!



*Wiring from the p.c.b. terminal block to the two mains sockets. Note that the mains input connector, the one with the protective tag covers, is a "male" type and also incorporates a fuseholder. The model used a 20mm fuse rated at 160mA 250V a.c.*

## ALIGNMENT

With everything now in place, it's time for some real action! Another visual check of the installation is definitely worthwhile, then wire up to the mains once again, this time connecting a small mains lamp (60W is fine) to the *Live out* and *Neutral out* terminals.

It would be sensible at this stage to secure the board to some firm surface using mounting pillars in conjunction with the holes provided in the p.c.b. This will keep everything under control while setting up the circuit.

It helps if an oscilloscope is available, but it isn't essential. Adjust preset VR1 to mid-travel, check your mains connections again, then switch on. After about ten seconds, everything should have settled, and the lamp will be fading slowly up and down. The scope is useful to check what shape the ramp is, and this can be done by using the test points.

Ground the 'scope to the 0V point at the edge of the board, and then hook onto test point TP1 for the signal. The ramp should start from 0V, rise to about 9V or 10V then reset to 0V again, repeating every 10ms, so set your scope's gain and timebase accordingly, and check whether this is the case.

Adjust preset VR1 and see what happens to the shape of the ramp – leave it at a setting where the lamp seems to fade at a consistent rate between its two extremes, and spends about the same time at those limits.

If you're feeling adventurous, look at the output waveform on IC4b (pin 7), and you'll see that when this is permanently high the lamp is fully on, but in the low state the lamp is off. When the lamp is fading, you'll see two abrupt waveform edges appear. The falling edge coincides with the reset of the ramp, but the rising edge defines the point in the mains half cycle at which the triac is turned on.

Set the 'scope to trigger on the falling edge, watch the sequence, and see how it relates to what state the lamp is in. It's not the end of the world if you haven't got a 'scope – the ramp can be set just as effectively by watching how the lamp responds – it shouldn't take long.

## PHASE CORRECTION

It was mentioned earlier that there is a remote chance that the transformer secondary may be sufficiently out of phase with the primary to cause control problems. This could be a lead or a lag error – one will cause the lamp to suddenly go to full on at the end of its fade down, and the other will result in the opposite effect.

The simple fix for this is to make some adjustments to the values of resistors R2 and R3. If you're sure that you've got a problem, remove these resistors and substitute a 100kΩ potentiometer, connect its outer terminals to the respective d.c. supply lines and its wiper to the board at the R2/R3 junction point.

Adjust the potentiometer to find a suitable voltage that takes account of the lead or lag error. Then measure its resistances to either side of the wiper to find the new values and fit resistors of about the same value (within the usual E24 series range). Basically, by changing the reference voltage, you're moving the zero-detection point backwards and forwards in time to take account of the error.

## ENCLOSURE

Once you're happy with the operation of the circuit, it's time to start putting everything safely in a box. As mentioned earlier, mounting holes are provided to allow the board to be fixed down and pillars are recommended to allow it to stand away from the surface.

For all mains connections, make sure that the cable is overrated for the job – 6A type is probably best. If you decide to use a direct cable connection for the mains input to the terminal connector, make sure that some good strain relief is included inside the box. A single cable tie round the cable is NOT good practice – a P-clip installation is far more professional (and lasts longer)!

The mains earth must be connected to both mains input and output connections. If a metal enclosure is used this *must* also be earthed.


*Completed unit showing the two mains sockets. The input connector is the "male" type.*

For the output side of things a commercial mains socket would appear to be the order of the day, and a chassis mounted type is recommended so as to keep everything compact. As shown in the photographs, connect the earth from the input mains cable directly across to the output socket. If a metal case is being used, make sure that this gets a good earth connection too.

## IN USE

This Christmas Tree Lights Controller has proved to be a useful and interesting addition to the festive display. Note that it is intended for use with light sets that connect *direct* to the mains, and *not* via a transformer, so check before you connect up.

As mentioned earlier, there's no need to restrict it to seasonal use, and it probably has a wide variety of other applications, maybe not even using lights as the load. Multiple units would produce a subtle but interesting light-show, and it would be quite easy to extend the basic circuit to a multi-channel system by adding extra control stages.

The other option is to create a totally manual controller by using a potentiometer that gives 0V to +10V into pin 6 of IC4b. Whatever circuit you decide to go with, remember that there's mains power around, so be absolutely sure before you change things! □

# PICTOGRAM

ANDY FLIND

*Become a novelty flasher at the Mad Hatter's (or other's) Xmas party! (Andy caps it nicely!)*

THE idea for this project germinated around the time the author's son announced forthcoming nuptials. As the groom's father, yours truly was expected to take a fairly prominent part in the proceedings, and those in key roles were requested by the happy couple to wear full traditional dress including, for the men, top hats.

It wasn't long before the notion of a ring of flashing light emitting diodes (l.e.d.s) around this spectacular piece of headgear was conceived. Well, some of us never quite grow up, as our wives frequently tell us.

The resulting display could, of course, equally well be applied to a belt or headband, or to almost any festive decoration.

## PIC-TURE THIS!

The task seemed ideally suited to a design using a PIC microcontroller. It's easy enough to make a bunch of l.e.d.s flash with some CMOS logic, but with a PIC the circuit becomes even simpler, and seemingly endless flashing and chasing patterns can be generated with very simple software. Before the advent of these devices these sequences would have required whole boardfuls of components, and would have been almost impossible to implement for this application.

The prototype was practically "thrown together" on stripboard. After all, there are many other matters to be attended to before a wedding and time for such frivolous activities as electronics was in chronically short supply.

## HAT TRICK

As the hat was obtained from a dress hire service a way of attaching it without causing damage had to be found, which will be described later.

After the event it was realised that the circuit had almost unlimited possibilities for amusing and eye-catching displays and decorations so it was re-designed onto a small printed circuit board and provided with an optional built-in display. The software was improved and tidied up, and the final result is now suitable for general release to other constructors.

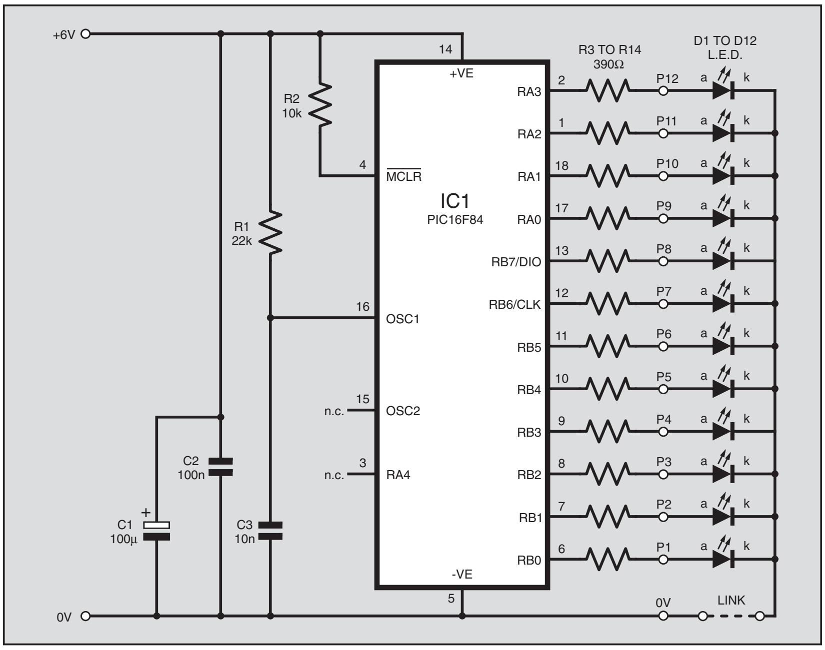



*Fig.1. Complete circuit diagram for the PICtogram.*

The full circuit for the PICtogram novelty project appears in Fig.1. For the greatest simplicity and lowest cost the PIC's RC (resistor-capacitor) clock option is used with a frequency set by resistor R1 and capacitor C3 to about 4kHz. Since the PIC divides the oscillator frequency by a factor of four, this results in processing taking place at about 1ms per step, which may help to make program timing simple to calculate.

In practice, the effects of the patterns are very subjective, it's easier to just "fiddle" with the timing factors until it "looks right". The MCLR connection is pulled high by resistor R2, which has a higher than usual value to allow it to be programmed "in circuit" by John Becker's excellent *PIC Toolkit Mk2* programmer. The high value is needed to allow the 12V programming voltage to pull this pin high enough to obtain programming mode.

Twelve outputs are used, consisting of the whole of Port B and the first four of Port A, to drive 12 l.e.d.s. Each has a current limiting resistor, R3 to R14, to control the l.e.d. current. The l.e.d.s used are rated at 2mA, but they are slightly over-driven in this circuit, which results in their being bright enough for most indoor situations.

It was found necessary to reduce the loading on pins RB6 and RB7 for in-circuit programming. An easy, single-connection way to achieve this is by disconnecting the common-cathode connection to all the l.e.d.s via the link shown in Fig.1.

For extended programming sessions a switch can be connected across this point so that it can be easily disconnected whilst programming actually takes place. The link also allows the built-in l.e.d. display to be easily disabled if the unit is programmed with the help of this and then connected to an external display.

Capacitors C1 and C2 are the usual supply decouplers used in battery operated circuits of this type.

There is a choice of methods for mounting the l.e.d.s. The printed circuit board holds the PIC and its components, plus facilities for mounting 12 l.e.d.s in a small circle.

As discussed later, the l.e.d.s could alternatively be mounted on a separate plastic strip and connected back to the p.c.b. by a ribbon cable. In this case the ring of l.e.d.s on the p.c.b. would be omitted.

## CONSTRUCTION

The PICtogram project is built up on a small printed circuit board (p.c.b.), which also accommodates a ring of programming/display l.e.d.s. This board is available from the *EPE PCB Service*, code 279.

The p.c.b.'s assembly and layout details are shown in Fig.2. It has the built-in ring of l.e.d.s with the common-cathode disconnection link at their centre. A line of connections for external output to the optional l.e.d. assembly is included (terminals P1 to P12, 0V).

Construction of this board is carried out by fitting the link wire, resistors, ceramic capacitors, and then the electrolytic capacitor C1. This capacitor is positioned horizontally on the board to achieve a low profile.

Ideally, a socket should be used for the PIC IC1, but this makes it the tallest component on the board! For the lowest possible profile IC1 could, perhaps, be simply soldered in place if it is to be programmed in situ. However, note that soldering a commercially pre-programmed PIC would probably negate its guarantee.

The l.e.d.s for this board are fitted with their cathodes (k) all facing towards the centre of the circle.

## PROGRAMMING TIME

Following construction, it's programming time! A ready-programmed PIC is available for this project with a total of eighteen flashing, rotating and special patterns which are invoked in a sequence which it is hoped will be found both interesting and pleasant. This PIC can be simply fitted into place and the unit powered up, when it should operate with no problems.

## COMPONENTS

**Resistors**
- R1 22k
- R2 10k
- R3 to R14 390Ω (12 off)

See SHOP TALK page

**Capacitors**
- C1 100µ radial elect. 10V
- C2 100n ceramic
- C3 10n resin-dipped ceramic

**Semiconductors**
- D1 to D12 red l.e.d., 2mA type (12 off)
- IC1 PIC16F84 microcontroller preprogrammed (see text)

**Miscellaneous**

Printed circuit board, available from the *EPE PCB Service*, code 279; optional min. s.p.s.t. switch (see text); solder pins; connecting wire; solder etc.

*Approx. Cost Guidance Only* **£15**

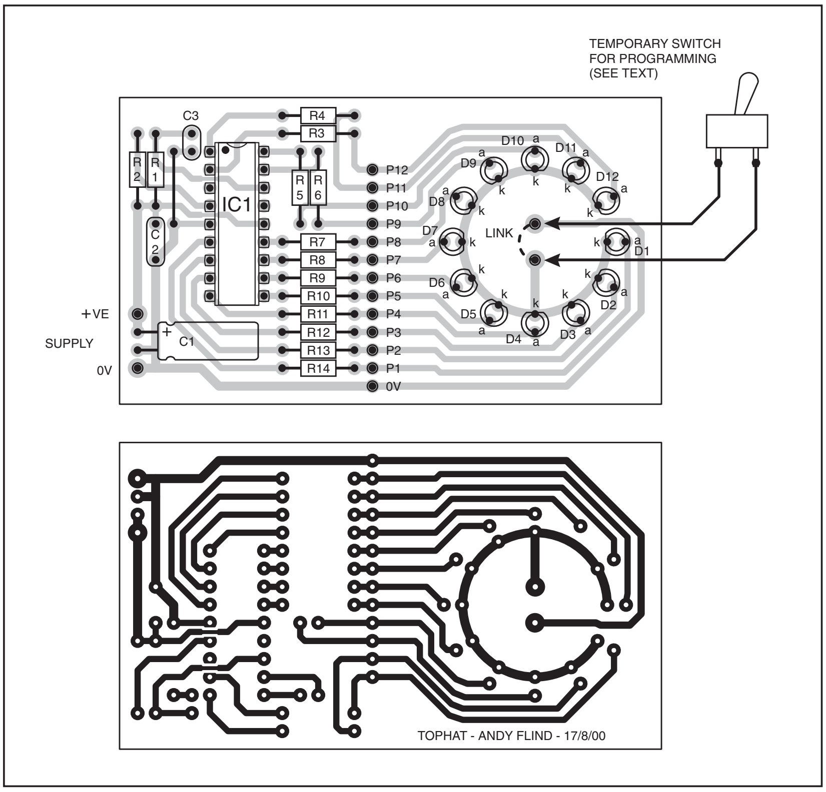
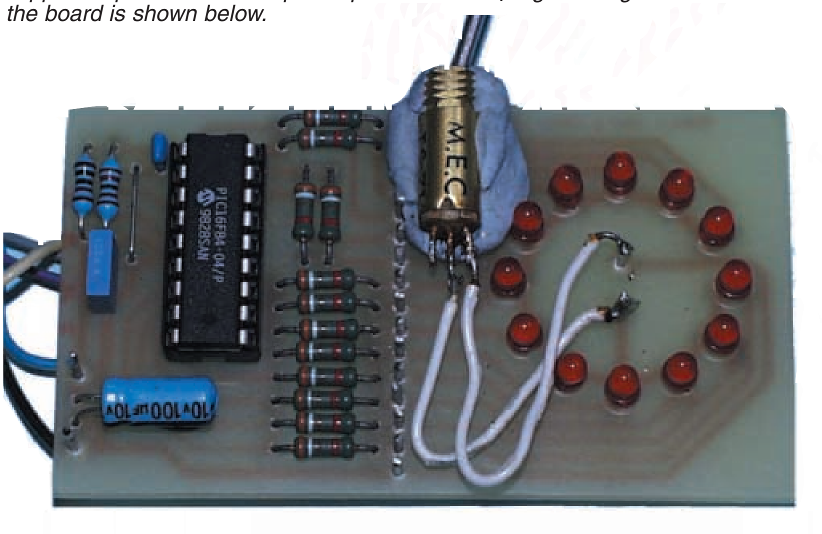



*Fig.2. PICtogram printed circuit board (p.c.b.) component layout and full-size copper foil pattern. The completed p.c.b. with the "programming" switch attached to the board is shown below.*



*Producing an illuminated plaque or wheel by placing the l.e.d.s around the perimeter of a cardboard disc.*

However, the real fun is to be had in writing the programs for a project of this kind and hopefully this is what many readers will choose to do. As mentioned earlier the author used John Becker's *Toolkit Mk2* for programming, though it has to be admitted that the one used had been modified to include, amongst other things, a beefier 5V regulator.

If a "standard" *Toolkit Mk2* is used, it would be advisable to keep an eye on the temperature of the regulator when the l.e.d.s are being driven, or to use an external supply for this project.

The connections for in-circuit programming are shown in Fig.3. *Toolkit Mk2* makes programming a very simple operation since the 12V programming voltage and reset pulses to $\overline{\text{MCLR}}$, plus most other housekeeping jobs, are carried out automatically by the associated software.

It will be necessary to disconnect the link to the l.e.d. cathodes whilst programming is taking place though, and the temporary connection of a switch for this is shown in Fig.2.

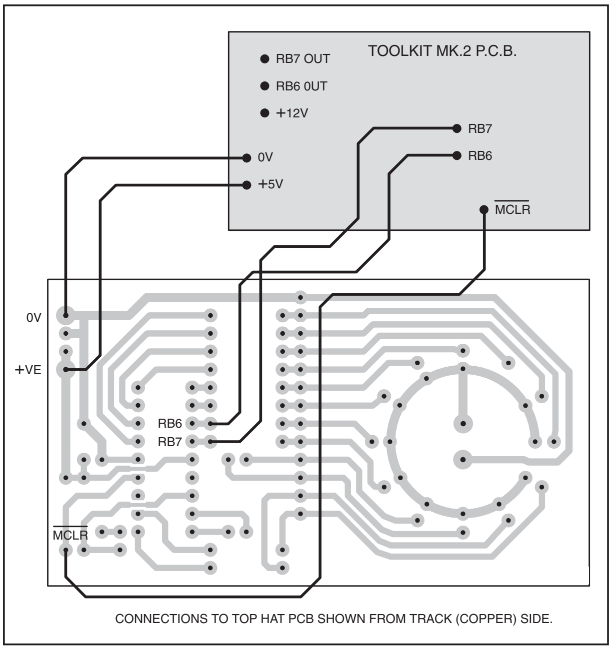



*Fig.3. In-circuit programming set-up using the Toolkit Mk2 programmer.*

Most constructors will probably start by obtaining an unprogrammed PIC and a copy of the software source code, most of which should be fairly simple to follow.

As this is an ideal project to hone initial programming skills, some notes on the methods used may be helpful. The original was written as TASM assembly source code but it shouldn't be too difficult for MPASM users to follow and modify as desired (*Toolkit Mk2* can translate between TASM and MPASM).

## ROUTINE EVENTS

The program uses three basic types of routine. One consists of loading a pattern onto the output, implementing a delay, then loading another pattern. Or, perhaps, just turning all the l.e.d.s off and doing another delay, at a set speed for a set number of times, using a loop. This is the simple flashing routine, and there are eight of these in the original program.

Another routine consists of loading a pattern and then causing it to rotate clockwise or anti-clockwise with a simple procedure to rotate the bits in the output files which can be used in a repeating loop. There are three of these patterns, in both clockwise and anti-clockwise versions, a total of six in all.

Finally, there are the more complicated patterns, where bits are turned on and off individually, which give the most pleasing results of all but produce a lot more code. There are four of these.

All these routines are written as self-contained subroutines so that a "program" can be built up simply using a string of "calls". Timing is carried out by a simple loop called "**dly**" placed right at the end of the program. This uses a variable called "**rate**" which is loaded before "**dly**" is called. Another variable called appropriately "**reps**" controls the number of times each pattern will repeat.

The effect generated by the flashing l.e.d. patterns is highly subjective so there really is no real alternative to testing each one as it is written to see what it looks like, and then adjusting the timing and repeats to get the preferred effect.

A glance at the full source code will show that the routines, when assembled into a full program, take up a lot of space. This creates problems simply of scrolling through them all in the edit screen during the trial-and-error creation process, and for those of us who do this sort of work with ancient computers that have been relegated to the workshop, whilst keeping the gigahertz Pentium mega-machine elsewhere for web surfing, it can lead to slow processing and programming.

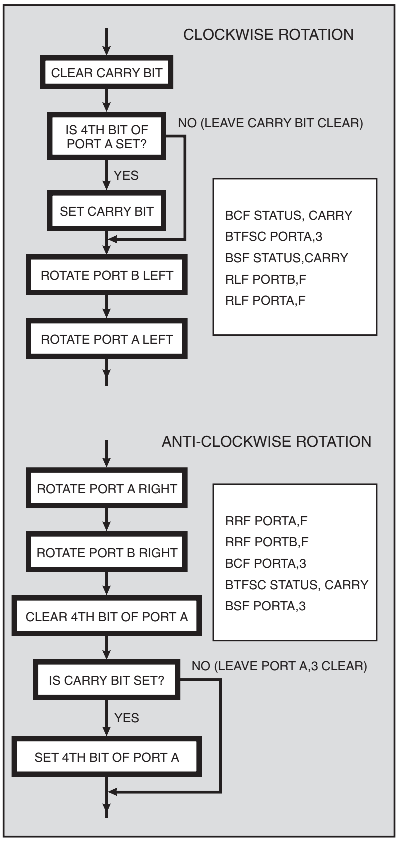



*Fig.4. Flow diagram for rotating a "file-and-a-half" left and right.*

John Becker has recently released a new version of *Toolkit Mk2* software (discussed in *EPE* Nov '00) which provides the incredibly useful "include" directive to overcome this difficulty.

Let's say a piece of code, perhaps a complete flash routine, has been tested and is working perfectly. It can be saved on its own under a filename, perhaps "**flash_1.asm**", and replaced in the main program with the statement "**.include flash_1.asm**" (or "**$include flash_1.asm**)". At assembly time, the assembler will simply add the code into the main program and convert it into object code as if it were there in full.

The advantages of this are twofold. First, there will be far less to scroll through when entering the text editor and navigating to the section currently under development. Secondly, an "**include**" can be excluded from the assembly process with a single "**;**" (semi-colon) placed before it, making it much easier to turn whole chunks of program on and off and, in the case of older computers, speeding assembly and programming times.

MPASM will have similar commands and directives, and programmers not familiar with them should investigate as they are incredibly helpful. Their inclusion (sorry!) in the latest *Toolkit* software is most welcome.

Rotating bits around a file and a half, Port B and the first four bits of Port A, may appear difficult to beginners in programming. In fact, the procedure is quite simple and involves only a few lines of code. The trick is to read and set the status carry bit at the appropriate points so that the desired value is read in on the next rotation command. Flow diagrams and some code for this are shown in Fig.4.

Hopefully these notes will encourage would-be programmers to dip their toes in the water, as this is an ideal project to experiment with. It helps to have a picture of the l.e.d. layout labelled with their port and bit assignments whilst designing the

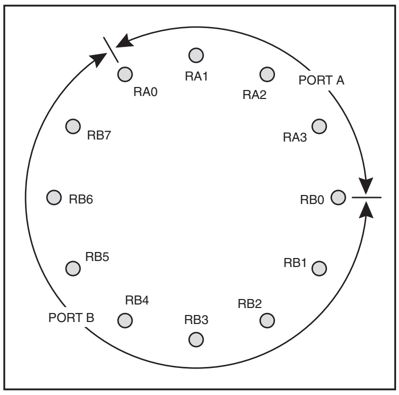



*Fig.5. Circuit board l.e.d. layout, with their port and bit assignments.*

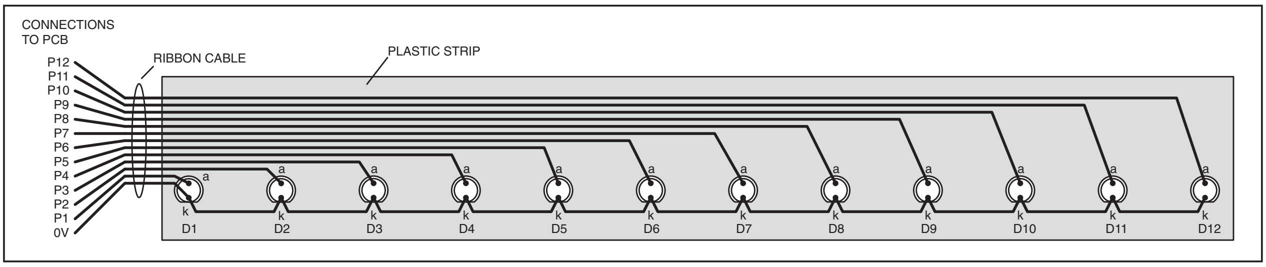
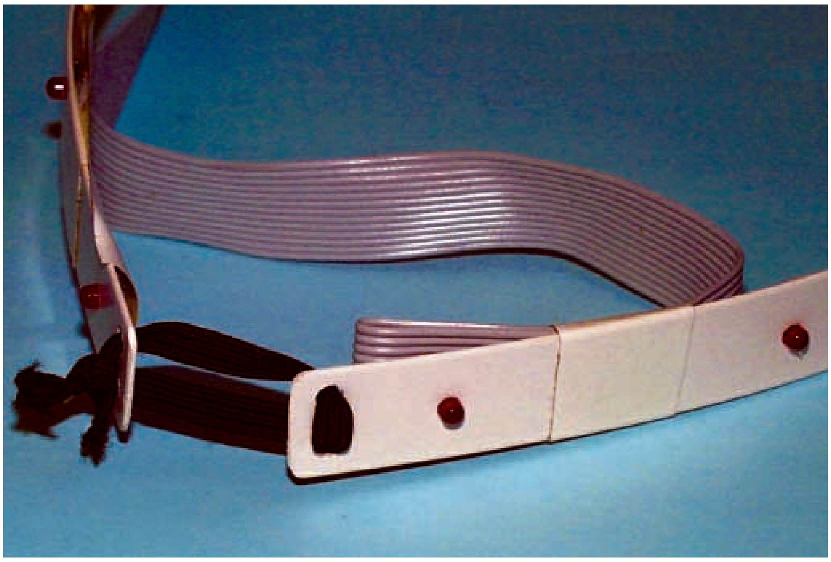



*Fig.6. Interwiring between the l.e.d.s in the hat or head band.*

more complex patterns, so this is shown in Fig.5.

The number of patterns and sequences that can be developed is practically endless; perhaps we should hold a competition for the most spectacular one produced! (*Well, perhaps not. But the code might be worthy of a mention in* Readout*!. Ed.*)

Incidentally, John Becker's *PICtutor* is probably still the best introduction around for the complete newcomer to the PIC and would easily enable anyone to learn how to program this project.



*The two ends of the l.e.d. carrying plastic strip are secured with elastic to form a headband, hatband, belt etc: Take your* PIC!

## ON DISPLAY

The original top hat display was constructed using plastic shelf-edging strip obtained from a DIY store. A piece of this was cut just long enough to go around the hat and be secured in place with elastic. This strip was drilled with a line of holes which were a tight fit for the l.e.d.s, which were pressed in and connected to the board with ribbon cable as shown in Fig.6.

The board was taped into some bubble-wrap packaging with the battery connector hanging out of one side, and this was pressed onto the connector of a holder containing four AAA cells; there was no on-off switch. The battery and board sat in the rim of the hat and, like the l.e.d. band, were kept in place with elastic. This was, after all, a serious rush job!

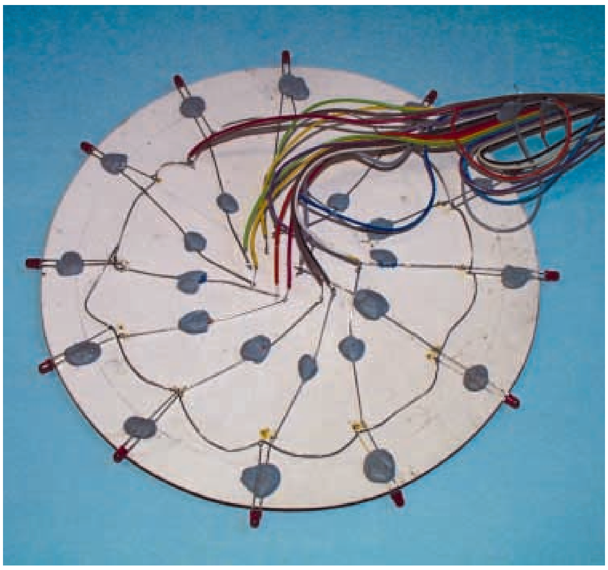
Readers possessing their own top hat could install the board and l.e.d.s directly into the hat with some kind of secret switch, where it would be almost invisible until switched on, for a far better effect. Those attending a lot of weddings might like to consider this option! (But do be aware that these days proper top hats are quite valuable.)

Other possibilities for this project are limited only by the constructor's imagination, though as a suggestion for a very simple method the board could be installed in a small case of some kind using its built in l.e.d. display to make a very unusual brooch, badge, or decoration. The l.e.d.s don't have to be all red, by the way, any other colour could be used instead.

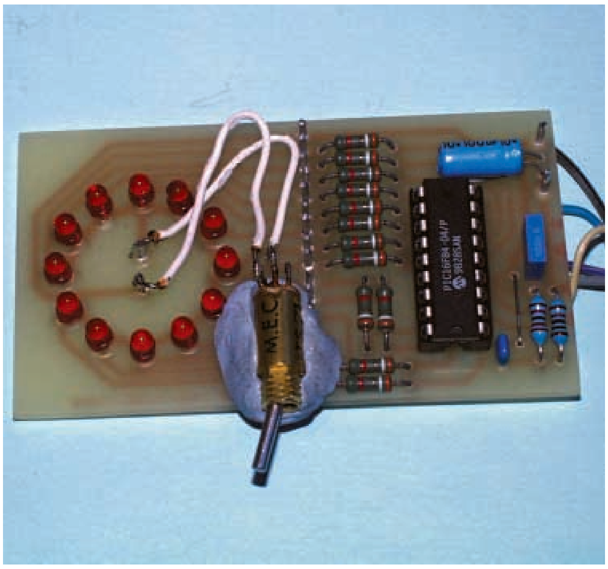
If you are programming your own PIC, note that it needs to be *initialised* for RC oscillator, Watchdog off, Start-up Timer on (Power on Reset).

Finally, for readers with the question on their lips, the answer is *No*, this project was not actually worn when walking up the aisle in the church!

The author has been asked this, several times, so the answer has to be provided! It was used late in the evening when the disco was in full swing, the lights were low, and most of the guests were sufficiently well lubricated to appreciate the novelty. □



*Off-board l.e.d.s arranged around a cardboard disc.*



*Programming switch soldered to the "link" pins on the p.c.b.*