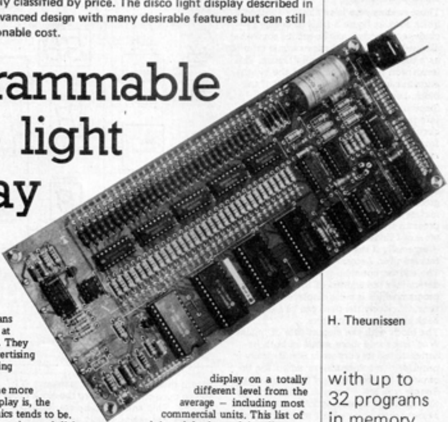
The essential ingredients for any self-respecting disco are well known, plenty of the right type of music, an abundance of coloured lights and gently top up the remaining space with people.
Any hi-fi will provide the music but the light display is rather more specialized. There are of course many variations on the theme ranging from the mediocre to 'way-out' and invariably classified by price. The disco light display described in this article is a very advanced design with many desirable features but can still be built at a very reasonable cost.

programmable disco light display



H. Theunissen

with up to 32 programs in memory

Light displays are very popular for many applications other than discos. They are excellent in the home, for instance, as a means of providing 'atmosphere' at parties or social gatherings. They are also very useful for advertising purposes for the enterprising businessman.
It cannot be denied that the more interesting a disco light display is, the more complex the electronics tends to be. This is mainly due to the fact that each light source, in most cases a mains powered lamp, must be controlled seperately, resulting in a 'channel' usually consisting of some form of logic decoding, a mains interface, and a firing circuit. This channel must then be duplicated for however many lamps are involved. Regretfully, we have not been able to do away with this problem. Ironically, however, it may be seen as an advantage simply because it allows for the easy expansion of the overall system – especially if the control electronics are designed with this in mind! It will become apparent that the circuit in this article can be as large as your imagination or your wallet allows!
A major disadvantage of the average disco light display is that the available light patterns are an integral part of the control circuit, possibly the contents of a memory IC, which must be purchased. This means that the pattern cannot be changed very easily, if at all. At this point we can start to sing the praises of the circuit here because this disco display is fully programmable. Furthermore, program changes can be made at any time by simply operating switches (no IC changes). The circuit also contains its own memory allowing up to 32 different programs to be stored.
There are numerous other highly desirable features of the circuit that put this disco display on a totally different level from the average – including most commercial units. This list of do's and don'ts explains all . . .

- Entirely user-programmable at any time.
- Up to 30 channels can be accomodated.
- Program selection can be run fully automatically or manually.
- 8 switched program run times available.
- Internal memory divided into: 16 (2 'banks' of 8) programs of 128 steps, or 32 (4 'banks' of 8) programs of 64 steps.
- Overall size of memory optional.
- Battery back-up for memory.
- Programs, banks and current memory address indicated by LED displays.
- Opto isolation from mains.
- All lamps switched at zero-crossing point of mains to reduce interference.
- Personal choice of display configuration (a matrix configuration makes possible a display with 225 lamps!)

So much for what the circuit can do, now how about what it doesn't do!

- It doesn't cost an arm and a leg.
- It does not require any programming skill.
- It does not require a great deal of practical ability to build it.
- It doesn't play the Hokey Cokey (although some may not consider this to be a major disadvantage) yet!

To sum up then, the circuit contains all of the desirable features (that we could think of at least) and yet it can be operated without 'computer' experience. The complete display can be as small or as large as desired; it may even be expanded at a later date.

Basic principles

Those readers who have already sneaked a quick look at figure 3 (this of course includes everybody) may be getting somewhat alarmed at what most articles would refer to as 'a slightly complex' circuit diagram. This impression is just a figment of the imagination as can be proved with the aid of the block diagram of figure 1.

Since it is the memory that holds all the information, this forms the heart of the circuit and all other 'blocks' either feed to or from it. The structure of the contents of the memory is illustrated in figure 2. It will be seen that it is divided into 'banks' (two or four depending on desired memory size) each of which in turn is divided into 8 programs. This simple method allows the total memory to be divided into reasonable program lengths and provides an excellent means of finding any given program quickly – especially if the program and bank counters are given 7-segment display read-outs!

The address counter, as its name suggests, determines the address of that part of the program which is being displayed at any one time. Obviously the same can be said of the bank and program counters.

The block with the elegant title of 'mains sync' is a shade more subtle in both its activities and its purpose in life. Basically it provides a synchronization signal for the circuit at the frequency of the mains supply. Simple, you say – but wait. It also ensures that the clock signal is synchronized to the zero-crossing point of the mains frequency and, by so doing, it eliminates the need for all those zero-crossing detectors that usually accompany each lamp-switching triac in the mains interface of the display. The answer to the next question is that, since the clock is synced to the zero-crossing point of the mains, all data changes at the output of the memory will always occur at the same point. The lamps will therefore always switch on (or off) at the zero-crossing point!

One further point before we leave the block diagram. The printed-circuit board designs for the display drive circuits do not appear in this article but they should grace the next issue.

The circuit diagram

The mains zero-crossing point detector is formed by IC1 (gates N1 . . . N3) in the circuit diagram of figure 3. The mains supply is present between the X and Z terminals and is applied to N1 via a voltage divider consisting of resistors R1 . . . R3. The inputs of N1 contain two diodes which chop the waveform of the mains supply to provide a square wave with an amplitude that is equal to the supply voltage of IC1.

The output of IC1 is differentiated by means of C1/R5 and C2/R6 and fed to the two inputs of N3. The resulting output of N3 is a pulse of about 200 μs at every zero-crossing point of the mains frequency. This pulse train is then fed, via a driver transistor, T1, and an opto-coupler, IC2, to the clock input of FF1. This ensures complete isolation between the mains supply in the zero-crossing detector stage and the rest of the circuit. It is for this reason also that the power supply – connected between X and Y – for the detector stage is derived from the triac control board.

The memory address counter is IC7 which will increment the address data by one at every clock pulse received at its clock input at pin 10. The clock signal for this purpose is generated by means of the variable fre-

1

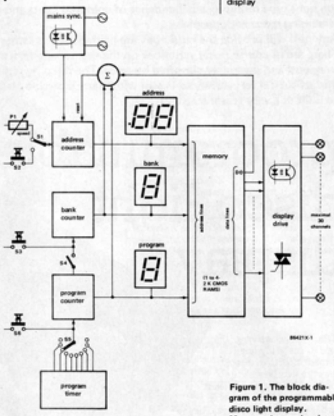


Figure 1. The block diagram of the programmable disco light display. Memory size can be chosen by the user.

2

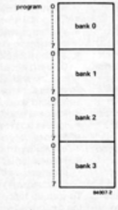


Figure 2. This illustrates the manner in which the memory is structured to provide easy access to any program.

Function table

- S1: A – RUN MODE
  B – PROGRAM & STEP MODE
- S2: STEP (increment address counter)
- S3: BANK increment (+1)
- S4: BANK automatic increment ON/OFF
- S5: PROGRAM RUN TIMES in minutes
- S6: MANUAL PROGRAM increment (+1)
- S7: DATA WRITE
- S8: WRITE PROTECT (key switch)
- S9: Mains switch
- S10: RESET switch
- S11 . . . S40: DATA switches
- P1: RUN speed control

3

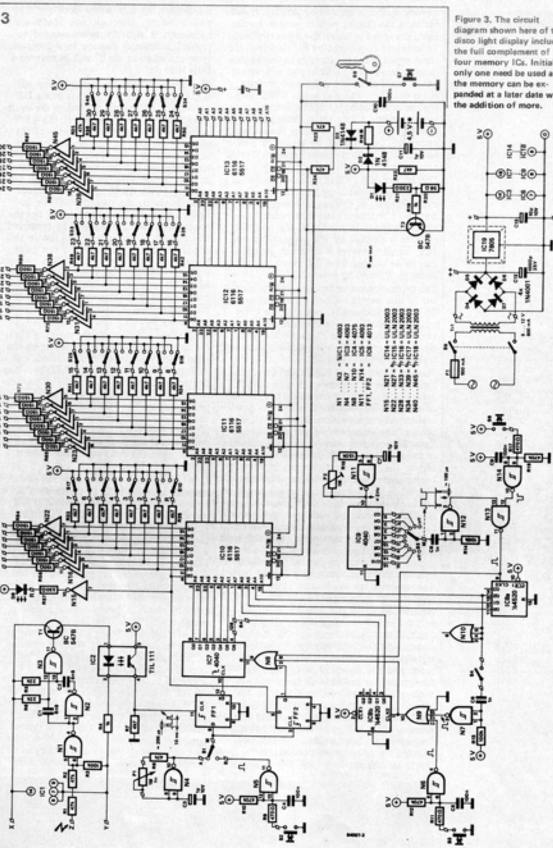


Figure 3. The circuit diagram shown here of the disco light display includes the full complement of four memory ICs. Initially only one need be used and the memory can be expanded at a later date with the addition of more.

quency oscillator formed by gate N4. If, for instance, the display pattern were a running light, the speed at which the lights run could be increased or reduced by P1. However, the clock signal is not fed directly to the address counter but via FF1 which, you will remember, is itself clocked by the zero-crossing point detector. The end result is that any change in the address counter is directly synced to the mains zero-crossing point. Switch S1 is included to allow the address counter to be 'stepped' by means of pushbutton S2. This is of course necessary during programming.

One half of IC8 (IC8a) forms the program counter which has a continuous count-up cycle from 0 . . . 7. That is, it counts up 8 steps (8 programs) and then resets to 0 only to begin the cycle again. The program counter is clocked by the program timer, IC9, which provides 8 different program run times ranging from 7.5 seconds to 16 minutes, selected by switch S5.

The program counter can also be incremented by one step at a time by means of switch S6 which, incidentally, overides the timer output. It should be realized that if S5 is switched to one of its off positions, any given program will run indefinitely until it is changed manually by S6.

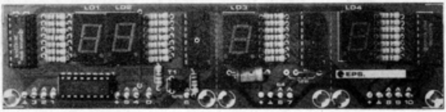
The remaining half of IC8 (IC8b) forms the bank counter which, depending on the program size, continually counts up in either 2 or 4 steps. This counter can also be stepped manually by means of pushbutton S3. To obtain fully automatic operation, that is, a continuous cycle through all the programs in the memory, switch S4 can be closed and at the highest program count, the bank counter will be incremented by one. It will be seen that the counters for program and bank are interconnected via an OR gate, N8. This ensures that each time either the bank or program counters are updated the address counter is reset to zero; after all, it is only reasonable that a new program should begin at the beginning! For those who are wondering what that strange little thing perched on the line to S4 is, it is simply a redundant gate.

We now come to the memory itself, of which the full complement of four 2K-CMOS-RAMS are shown in the circuit diagram (IC10 . . . IC13). In normal operation these are of course in the 'READ' mode and the contents of the address, determined by the address, program, and bank counters, are used to switch on (or off as appropriate) the output to the display itself. Normally then, the R/$\overline{W}$ pins of each memory are held high by resistor R23. This line must therefore be taken low whenever a program is to be entered or modified and this is carried out by switch S7 via switch S8. Switch S8 is a safety 'lock-out' key switch which, although not absolutely necessary, is strongly recommended to prevent accidental damage to a program. How and when to use S7 will be covered a little later on.

The power supply for the memory ICs is taken from the 5 V line via diode D2. Should the 5 V supply fail (when the equipment is not in use for instance), the 4.5 V battery will preserve the contents of the memory via D3. At the same time, the absence of the 5 V supply will switch off transistor T2 and inhibit the memory outputs by causing the $\overline{CE}$ inputs of the memory ICs to go to logic 1 via resistor R24. In short, the memory will be inoperative (in the low power mode) but the contents of the memory will be preserved. The current consumption is so low in this condition that the battery will last quite literally for years but should still be changed every 12 months or so. A NiCd (three cells of 1.2 V) can also be used in which case resistor R18 (270 Ω) is required to provide a charging current. This resistor is not needed with a dry battery.

Each of the data lines of the memory ICs is fed to the LED in an opto-coupler on the triac board via a driver, N15 . . . N45, and an indicator LED. The indicator LED provides a direct read-out of the data at that particular address. This is of course essential during programming. The data lines are also fed to the programming switches S10 . . . S40 via resistors. When S7 is pressed, and S8 is switched on, the data set by these switches is written in the memory at the address indicated.

One final detail before we leave the circuit diagram of figure 3. The D0 output of IC10, switch S10, and the associated LED (via driver N15) all have a particular significance. It will be seen in 'programming' below that the length of a program (or sequence) can be a maximum of 128 or 64 steps. However, this may be more than required and therefore some means of programming the end of a sequence and returning to the beginning must be provided. This is where data line D0 of IC10 comes in. In the normal course of programming D0 will be logic low until the end of a sequence when a 1 will be entered at this location (by S10). When the display is up and running, a 1 appearing at D0 will be synced with the address oscillator by FF2 and used to reset the address counter to zero via N8. The display sequence will then start from the beginning again. LED D8 serves to indicate this 'reset' pulse when it occurs. The reset bit (D0 of IC10) is not synced with the zero-crossing-point pulses. However, as the reset only occurs at the end of a program, this will cause negligible inter-



4

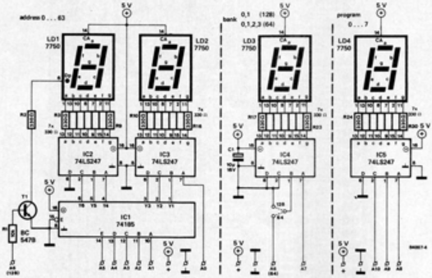


Figure 4. The address, program and bank indicator circuit diagram is shown here. The address lines are connected to those on the main circuit of figure 3.

ference.
The circuit diagram for the four LED displays, LD1 . . . LD4, is shown in figure 4. The printed-circuit board layout for this circuit is shown in figure 6. The address reference A0 . . . A10 refer to those at the right of the main circuit diagram of figure 3. An appropriate link must be made to set the program step-length at the input to the decoder for LD3. If the program step-length is 128 steps, transistor T1 switches on the decimal point of LD1 for address counts above 63.

Construction

If the printed-circuit boards illustrated in figures 5 and 6 are used, construction of the electronics section of the disco display should prove no problem. However, before assembly can begin, the final design format must be decided upon. This refers in particular to the LED read-out display board which, it will be noticed, can be divided into three separate sections. This has been done to allow the maximum flexibility of the design as it was considered that many readers may wish to build the display controller into an existing piece of equipment. The complete printed-circuit board as shown in figure 6 will match the suggested front panel design illustrated in figure 7.
After assembly has been completed, not forgetting the two links (64 or 128 step program length), the two boards should be interconnected. This can be carried out by short lengths of wire or, if preferred, ribbon cable can be used. All the address lines as marked on the two boards, with the exception of A6, must be connected. For a 64-step program this is taken to the point marked A6 on the board containing display LD3. If a 128-step program has been chosen, it must be connected to A6 on the board containing displays LD1 and LD2.
There are three + terminals and three ⊥ terminals on the display board. These are separately interconnected: one + and one ⊥ is connected to the + and 0 terminals respectively on the main board near C12. If the display board is separate, each + and ⊥ should be connected with the + and 0 on the main board. The common point for the anodes of the indicator LEDs should be taken to the + terminal near C12. The cathodes are connected to channel outputs 1 . . . 30. Another set of + and ⊥ terminals will be found on the main board: these are for the switches. The switch connections should preferably be commoned after the switches have been mounted on the front panel as this requires only two wires to be returned to the main board.
Normally, the channel indicator LEDs are connected in series with the LEDs in the opto-couplers on the triac board. To enable the circuit to be tested at this stage, some form of current limiting must therefore be included as a temporary measure. Two diodes type 1N4001 are therefore connected in series with the 5 V supply and the common anode connection of the indicator LEDs. The LEDs should have a forward voltage of about 1.6 V. If the indicator LEDs are dispensed with and only the opto-coupler LED is used, the display pattern is, of course, shown by the display itself: resistors R58 . . . R87 should then be 330 Ω.
It will be remembered that the supply for the mains zero-crossing point detector is derived from the triac control board: this

Max. no. of channels	ICs required
7	10-14-15
15	10-11-14-15-16
23	10-11-12-13-14-15-16-17
30	10-11-12-13-14-15-16-17-18

5

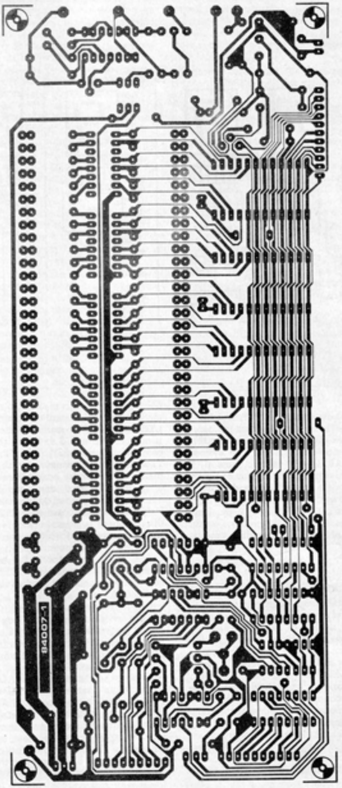
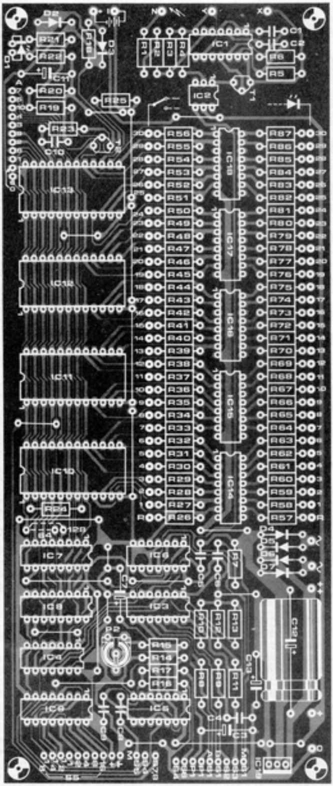


Figure 5. The component layout and track pattern of the printed-circuit board for the circuit diagram of figure 3.

will be clarified in the next issue when the triac unit as a whole will be described. To enable the main board to be tested at this stage, the zero-crossing point detector can be supplied from the main board. DO NOT FORGET to remove these connections when the triac board is connected. The supply connections are: X to + and Y to 0 (near C12) and Z to one of the secondary a.c. terminals of the mains transformer (that is, one of the ~ terminals on the main board).



A final construction point: IC19 must be fitted on a small heat sink.

Programming

When the circuit is first switched on (prior to any programming), the memory ICs will contain garbage, but you knew that, of course!

Display sequences will obviously depend on the contents of the memory and the chosen program format (64 or 128 steps). Further-

Parts list

Resistors:
R1,R2,R8,
R23 . . . R25 = 47 k
R3,R13,R14 = 100 k
R4,R19 = 1 k
R5,R6 = 22 k
R7,R22 = 4k7
R9,R11,R17 = 470 Ω
R10,R12,R16 = 470 k
R15 = 150 k
R18 = see text
R20 = 56 Ω
R21 = 220 Ω
R26 . . . R56 = 4k7
R57 = 330 Ω
R58 . . . R87 = 150 Ω

Capacitors:
C1,C2 = 6n8
C3,C7,C11 = 1 μ/10 V
C4,C5,C9,C10 = 100 n
C6,C8 = 1 n
C12 = 1000 μ/25 V
C13 = 10 μ/10 V

Semiconductors:
T1,T2 = BC 547B
D1 = red LED
D2,D3 = 1N4148
D4 . . . D7 = 1N4001
D8 = LED
Optional: 30 LEDs for channel indication
IC1,IC3,IC5 = 4093
IC2 = TIL 111
IC4 = 4075
IC6 = 4013
IC7,IC9 = 4040
IC8 = 4520
IC10 . . . IC13 = 6116 or 5517
IC14 . . . IC18 = ULN 2003
IC19 = 7805

Miscellaneous:
P1 = 1 M lin. potentiometer
P2 = 1 M preset
S1 = single pole two way toggle switch
S2,S3,S6,S7 = push-to-make switch
S4 = single pole toggle switch
S5 = 12 way wafer switch
S8 = single pole key-lock switch
S9 = double pole mains toggle switch
S10 . . . S40 = single pole two way toggle switch
Tr1 = mains transformer 9 . . . 12 V, 800 mA secondary
4.5 V battery
F1 = fuse 500 mA slow
Heatsink for IC19 (SK13, KL105)
Printed-circuit board 84007-1

NOTE: remember that some components are not required and others in smaller numbers if all channels are not used.

6

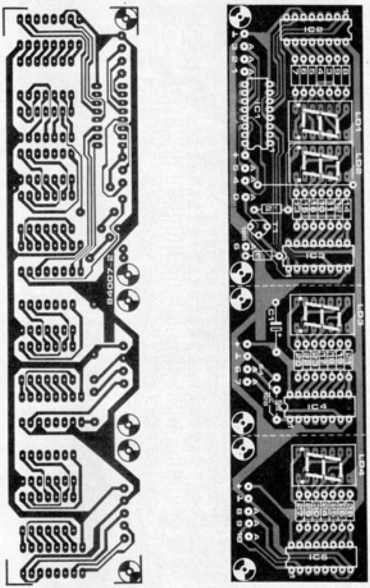


Figure 6. The printed-circuit board for the address, program and bank counters can be divided into three parts if preferred.

Parts list.

Resistors:
R1 = 10 k/1/8 W
R2 . . . R30 = 330 Ω/ 1/8 W

Capacitors:
C1 = 10 μ/16 V

Semiconductors:
T1 = BC 547B
IC1 = 74185
IC2 . . . IC5 = 74LS247
LD1 . . . LD4 = 7750
Printed-circuit board 84007-2

more, a full memory will allow a total of 30 channels (lamps) and these can be arranged in any number of pleasing designs including a dot-matrix for alpha-numerics (that means letters and numbers, sir!). Having decided upon the display format and the sort of programs that are to be used, the links at the A6 address line must be fitted as described under 'construction'. Before starting the programming, it is advisable to commit the desired sequences or patterns to paper as even 64 steps can get decidedly confusing.

To adjust preset P2, set switch S5 in position ½ (minute) and adjust preset P2 so that the program display jumps on one every 30 seconds.

Off we go then. Set switch S1 to position B (step mode), S4 to OFF, and S5 to off to prevent the program from jumping on during loading. Switch on the switch-key S8 and press S6 and S3 to get the right program and bank. The address display should read 00; if not, press S3 or S6 until the right program and bank are indicated on the display. The program data is set by switches S10 . . . S40 (or whatever number you have decided upon). Any one of these switches set to 5 V denotes a logic high and causes the appropriate lamp to light. A switch set the other way gives a logic 0 and the corresponding lamp does not light. Are you still with us?

Set the program data and press switch S7. The data lines are now inputs and the memory ICs will be fed with a write pulse and accept the data set with the switches. When S7 is released, the data lines revert to being outputs and the set pattern will be indicated by the channel LEDs. Now press S2 once (to increment the address by one), set the data switches, and again press S7. If an error was made during the entering of

7

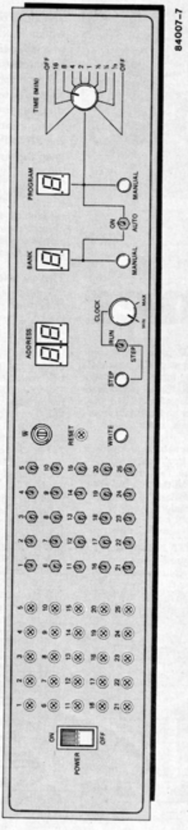


Figure 7. A suggested design for a 19 in (483 mm) front panel for the programmable disco lights. This panel matches the complete printed-circuit board of figure 6.

the data, simply set the correct data and press S7 again. This works, however, only before S2 has been pressed. If S2 has been operated, press S6 until the same program is indicated on the display. Then go to the faulty address by means of S2, alter the data, press S7, and proceed to the next address by pressing S2.

As mentioned previously, data line D0 will remain logic low until the end of a pattern. On the address following the last line of the sequence, set S10 to 5 V (logic high); this can also be done on the last line of the sequence itself (together with the program data) if preferred. With some display patterns (especially running light patterns) it improves the display continuity, but it really is a matter of choice. Try some simple patterns to see the effect. And that's all there is to it apart from a few pointers.

At the end of the programming, do not forget to switch off the key-switch otherwise (in the case of disco's) you might find yourself arriving at a booking with this terrific new display you have been raving about only to find that you have a completely garbled memory. Not good for the old ego, chaps!

Don't be too worried about making a false entry during programming as mistakes can be easily rectified. You do not have to reprogram the entire memory, just the line containing the error. Unless of course, you have a major disaster on your hands. In this case, turn the telly off and lock the door before starting!

It is possible to include delays and acceleration in your program by repeating the same data in several addresses. This makes a very effective display when properly done but it does require careful planning with due regard to program length (64 or 128 steps). Remember not to be caught out by the address counter read-out. This just indicates from 0 to 63. If a program length of 128 steps is opted for, the decimal point of LD1 signifies the 'upper' 64 step range.

Set S1 to position A when the program should run; adjust the run time with P1. It may happen that when the run time is increased (that is, smaller resistance of P1), the pattern on the display does not run smoothly (stutters) or even stops altogether. This is caused by the frequency of N4 being too high in relation to that of the zero-crossing pulses. Because the trigger levels of different makes of 4093 show wide variations, this erratic running may or may not occur. The adjustment range of P1 should be set by means of R8 and/or C3 so that stuttering or stopping of the pattern just does not occur.

A point worth noting! The disco display is completed, programmed, and ready to go to work . . . However, when it is switched on, nothing happens: no lights, no LEDs, just panic! Fear not, gentle DJ, all will be as it should be if you just press the program step switch, S6, and a program will start from the beginning.

To end, we are sure you don't need reminding that the mains supply is a little 'conspicuous' in this circuit. Please take care, as we have no desire to reduce our circulation by stopping yours! ◼