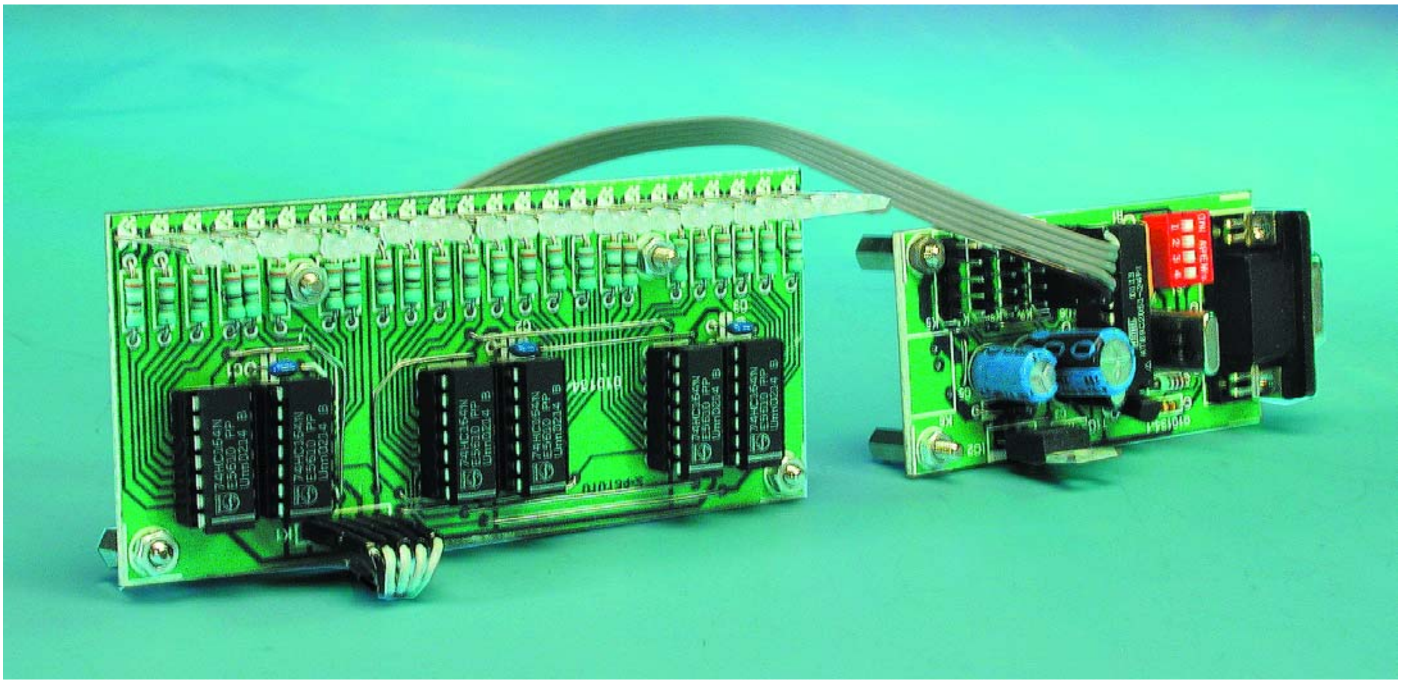
# Two-Colour Running Light

## with microprocessor control

Design by A. Köhler

**Lighting effects are always popular and we frequently publish such projects in the pages of *Elektor Electronics*. Here is a circuit that uses a microcontroller to drive up to 96 bicolour LEDs in a wide range of patterns. A particular feature of the design is the way it exploits the persistence of human vision.**



Instead of `regular' LEDs this running light uses so-called bicolour LEDs which can emit two different colours, usually red and green. These are available in versions with three connections (a common cathode and separate anodes) or with two connections. We use the second type in this circuit, where the two LEDs are connected in antiparallel so that one LED or the other lights according to the direction of current flow.

A special circuit is required to drive the bicolour LEDs. Since the LEDs generally only require a current of between 5 mA and 20 mA to light with sufficient brightness, the circuit does not have to switch a particularly large current. HCT logic devices, which can sink or source up to 8 mA, can therefore safely be used. The bicolour LEDs are connected between two such outputs. If the two outputs are at the same logic level, both LEDs will remain dark; if the levels are different either the red or the green LED will light, according to the direction of current flow.

Since every LED will require two outputs, they cannot be driven directly from the microcontroller's outputs. A serial drive circuit is therefore used that requires just two signals: with just a data signal and a clock signal we can control a large number of LEDs. The serial-to-parallel conversions are done by HCT shift registers. In the first phase of operation the data bits are shifted rapidly through the shift registers: so rapidly, in fact, that this operation in not visible to the naked eye. In the second phase of operation the data remains at the outputs of the shift registers for a longer period and thus appears stable. After a brief pause a new pattern is loaded into the shift registers. With a suitable physical arrangement of the LEDs the effect can be improved still further.

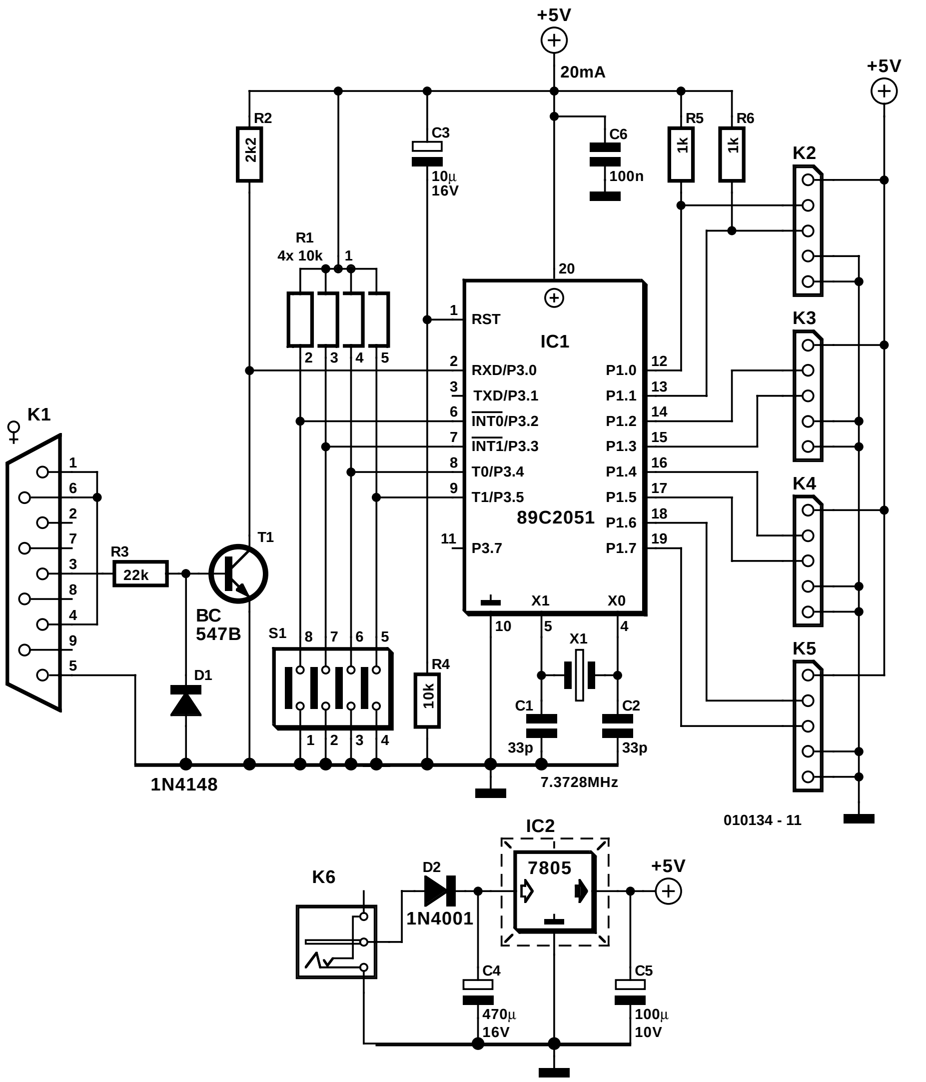



Figure 1. This minimal microcontroller circuit can control up to 96 bicolour LEDs.

## The microcontroller...

The circuit consists of a central microcontroller unit and up to four bicolour LED modules. The arrangement shown in **Figure 1** is a straightforward microcontroller circuit employing an Atmel 89C2051. The data sheet for this device can be found at www.atmel.com/atmel/acrobat/doc1045.pdf. Components C3 and R4 form a power-on reset network. A primitive RS232 interface is provided via pin 2 for receiving data only; the transmit function via pin 3 is not implemented. Transistor T1 performs the voltage level conversion. D1 limits negative input voltages, while R3 limits the base current of T1 as well as the current through the diode. This circuit removes the need for the charge pump circuit that would otherwise be required.

The oscillator connected to pins 4 and 5 uses a 7.3728 MHz crystal from the junk box. This oscillator frequency restricts the choice of available RS232 baud rates, but simplifies the programming of longer delay loops. Any other crystal could of course be used, although a few delay constants in the software would then have to be adjusted to suit.

Pins 6 to 9 are connected to a DIP switch with pull-up resistors. Depending on the software in the microcontroller, these can be used, for example, to select between light patterns, or to set the baud rate of the RS232 interface. Neither of these options is yet implemented in the software.

The LED modules are controlled via port 1. Each has a data signal (P1.0, P1.2, P1.4 and P1.6) and a clock signal (P1.1, P1.3, P1.5 and P1.7) to control a group of LEDs via their shift registers. Since the connections are short no protection circuitry is necessary. A point to note is the provision of pull-up resistors on port pins P1.0 and P1.1. These pins serve as non-inverting and inverting inputs (respectively) to an internal comparator. Without pull-up resistors of less than 2 kΩ the LEDs would not be driven correctly: the other pins of port 1 have internal pull-ups. The current drawn remains under the 15 able carrying data and clock as well as power.

This is derived from a mains power supply which must have an output voltage of at least 9 V and be able to deliver a current of 200 mA per LED module plus 20 mA for the controller board. That can add up to a sizeable total current, and so a heatsink must be fitted to voltage regulator IC2.

## ... and the four LED modules

As can be seen from **Figure 2** each LED module consists of 24 bicolour LEDs and six 74HCT164 shift registers. The data bits are presented to the data inputs (pins 1 and 2) of IC1, the first shift register, and then taken

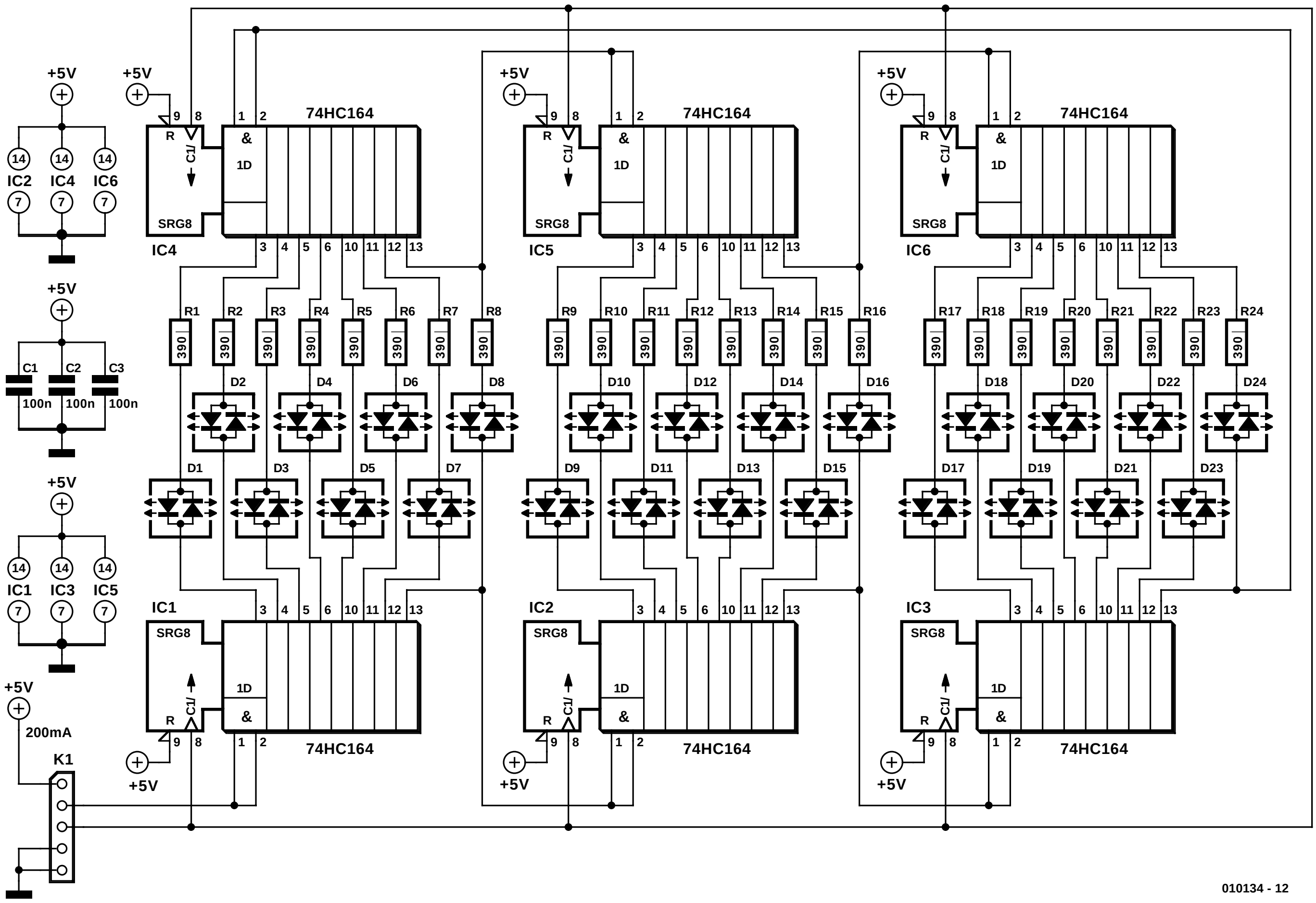


Figure 2. Two shift register outputs are used per bicolour LED.

from the final output (pin 13) to the inputs of the second shift register. In this way a chain is formed of the six shift registers, which are all clocked together (by pin 8).

The LEDs are connected to the outputs of the shift registers with series current-limiting resistors. A value of 390 Ω gives a current in the red LED of 9 mA, in the green of 7 mA. This is at the limit of what the logic ICs can supply, and does indeed give adequate brightness.

Populating the circuit boards should not present any difficulties, as long as the `smallest components first' principle is adhered to. Watch out for the three wire links. The microcontroller on the control circuit board (**Figure 3**) should be fitted in a socket. If connection to a PC will not be required, the sub-D connector and level shifting circuit can be dispensed with. Also, if the DIP switch will not be used, it too can be dispensed with, along with its pull-up resistors. Unused inputs should however be tied to a fixed voltage level, RxD to +5 V.

The single-sided printed circuit board for the LED module could not be routed without wire links, as clearly shown in **Figure 4**. Insulated wire should be used for the links, since in places they run close alongside one another, have bends or pass under IC sockets. Apart from that, there is little else to say about the circuit board.

## Software variants

Now that we have seen that the hardware does not contain any big surprises, the key to the circuit must lie in the software. The software can be downloaded free of charge from the *Elektor Electronics* website (under this month's Free Downloads: look for number **010134-11**). The software is available not just as `ready to burn' hex code, but also as assembler source code.

The basic version of the software does not use the DIP switch. It could, for example, be used to select between a number of different patterns; this could also be done via the serial port.

The software is built around the following routines.

**TAKT 1 to TAKT 4** (clock 1 to clock 4)
These routines generate the clock signals for the shift registers. In order to allow the modules to be updated at different times, each module is provided with its own routine.

**MUAU 1 to MUAU 4**
These routines emit the patterns, and are designed for LED modules with six shift registers, the number of shift registers being given in R2. R1 is used as a bit counter. A byte is fetched from the pattern table and split into individual bits via the carry flag. When a complete byte has been transmitted, the next byte is fetched from the table.

**ZEIKO** (time correction)
To allow all modules to use the same pattern, the pattern table pointer must be adjusted after each module has been processed.

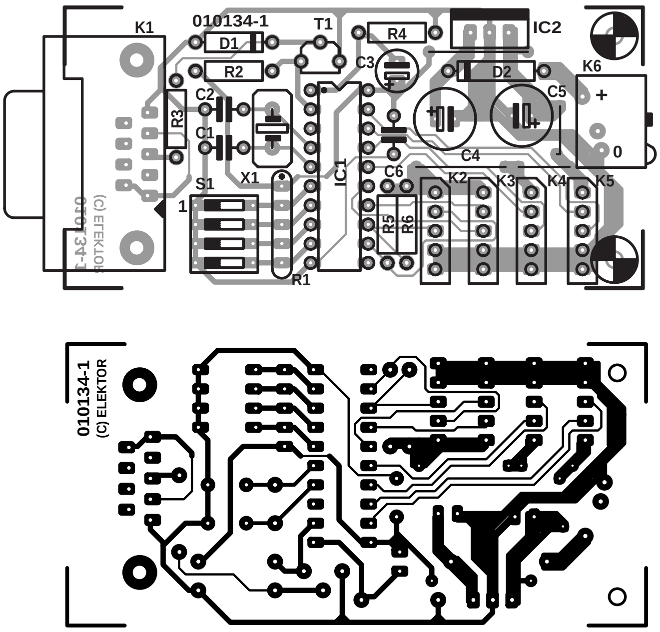



Figure 3. The microcontroller printed circuit board.

**LOE1, LOE2** (clear 1, clear 2)
These routines shifts zeros into all the shift registers of one module, thus extinguishing all the LEDs.

## COMPONENTS LIST
### Controller board

**Resistors:**
R1 = 4-way10kΩ SIL array
R2 = 2kΩ2
R3 = 22kΩ
R4 = 10kΩ
R5,R6 = 1kΩ

**Capacitors:**
C1,C2 = 33pF
C3 = 10µF 16V radial
C4 = 470µF 16V radial
C5 = 100µF 16V radial
C6 = 100nF

**Semiconductors:**
D1 = 1N4148
T1 = BC547B
IC1 = AT89C2051-12PC, programmed, order code **010134-41**
IC2 = 7805

**Miscellaneous:**
K1 = 9-way sub-D socket, angled pins, PCB mount
K2-K5 = 5-way SIL pinheader
K6 = mains adaptor socket, PCB mount
PCB, order code **010134-1**
Disk, order code **010134-11**

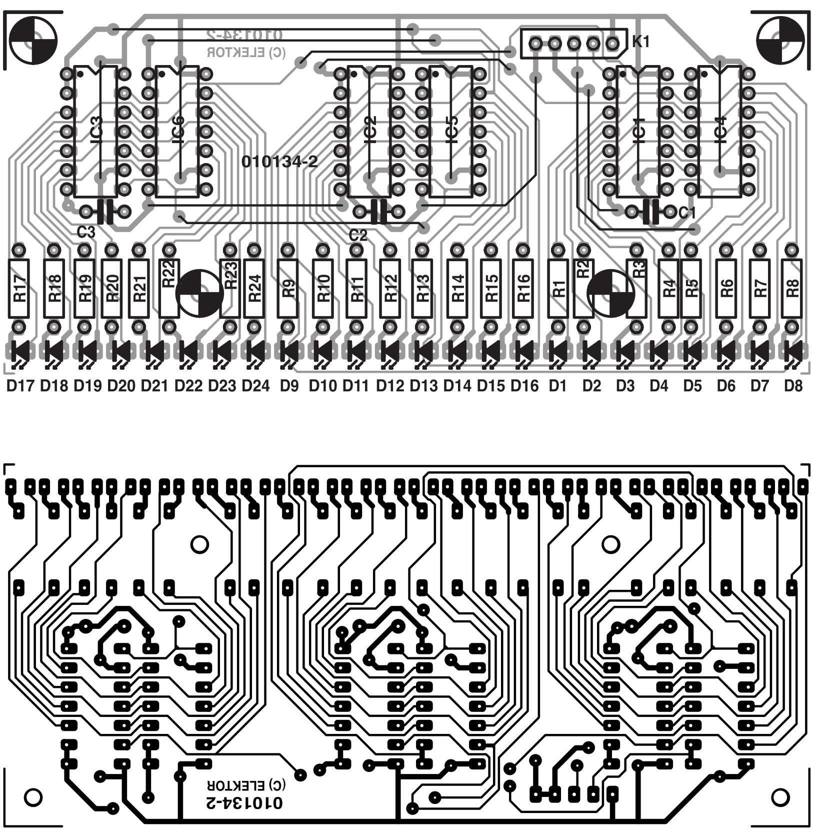



Figure 4. Beware! There are many wire links on the LED module printed circuit board.

**ZEIT** (time)
This is a small delay loop that slows the display down to a reasonable speed. This speed can easily be adjusted.

The largest part of the program consists of the nine pattern tables. With the help of the documentation it should be possible to replace the contents of the tables with your own patterns.

(010134-1)

## COMPONENTS LIST
### LED board

**Resistors:**
R1-R24 = 390Ω

**Capacitors:**
C1,C2,C3 = 100nF

**Semiconductors:**
D1-D24 = bicolour LED, 2 pins, 3mm dia.
IC1-IC6 = 74HC164

**Miscellaneous:**
K1 = 5-way SIL pinheader
PCB, order code **010134-2**