Build this tiny microcontroller driven lightshow

Microcontroller driven LED displays are popping up everywhere nowadays, from shop windows, to tiny belly-button raver lights. However, most of these have a few problems. Firstly, the cheap ones are usually just that—so cheap that they are basically a throw-away item. They often have poor-quality LEDs, or the LEDs are overdriven and die an early death.

Secondly, most of these lights are not programmable, so you are stuck with the basic program and sequence they are built with.

This kit allows you to build a high-quality, fully programmable light display capable of producing any colour you desire. What's more, it is a mere 32mm x 24mm in size, and can run from three 1.5 volt button cells for several hours. It also has a tiny pushbutton switch, so you can control the sequence when you desire. With careful programming, you can even program it with more than one sequence, and use the switch to flip between them!

The brain behind the colours

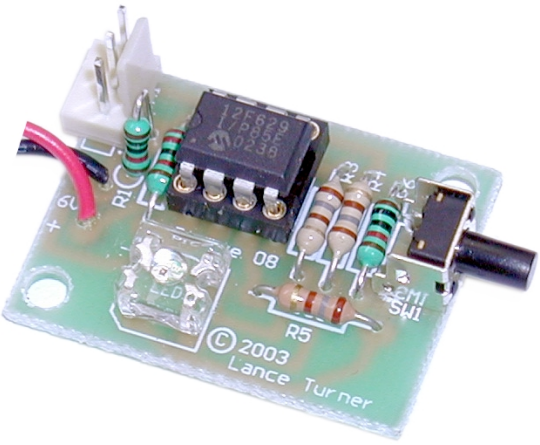
The brain of the circuit is the Picaxe08M pre-programmed microcontroller from Revolution Education in the UK. This company provides a range of microcontroller and micro-based boards and products, and one of those is the Picaxe series.

These consist of various Pic microcontrollers that have been programmed with a bootstrap program that allows them to be connected directly to a serial port on your PC without the need for special programmer boards. Just two resistors are required for the three-wire interface. This makes it possible to produce projects where the micro can be reprogrammed in-situ in seconds.

If you look at the circuit board, you will notice the three-pin connector and the resistors R1 and R2—they are all that is needed to connect the microcontroller to the serial port of a PC or other device. No other microcontroller is this simple to use. This reason, and the availability of a free basic interpreter for this range of micros, are why these micros were chosen for this and similar kits. They are simply easier to use and far cheaper than anything else on the market. However, they do have programming space and speed limitations, but in most cases, unless you are trying to sample or switch at radio frequencies or whatever, this generally doesn't cause problems

The circuit

As already mentioned, all of the hard work is done by the microcontroller, and there is not much else to the rest of the circuit. The micro directly drives an RGB 5mm LED using three of its output pins, via current limiting resistors R3 to R5, while a fourth pin is used to monitor the state of the pushbutton switch. The LED has four pins, one pin is a common cathode and the other three are connected to the anodes of the three

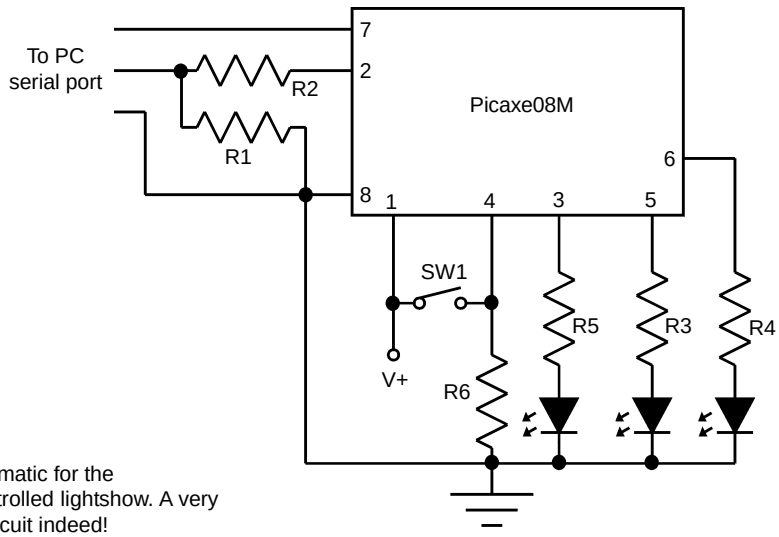


LEDs, allowing them to be driven separately.

Note: on the overlay of the circuit board supplied, resistors R3 to R5 have no value labels. This is because the value depends on how bright you want each LED chip in the RGB LED to be, and can be varied, especially if you find that the blue and green LED chips are less bright than the red. The red LED chip has a lower voltage drop than the other two, so receives more current. Calculations limit the resistors to the following values when running from a 4.8 volt supply:

R3: 75 ohms
R4: 100 ohms
R5: 120 ohms.

The 10k resistor, R6, is used to tie



The schematic for the microcontrolled lightshow. A very simple circuit indeed!

the switch input to ground when the switch is not being pressed, rather than letting it float. And that's it!

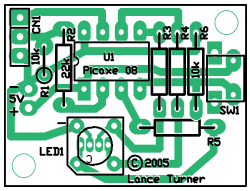
Assembly

Well, being so simple, there isn't much to say here, other than make sure you get the LED the correct way around. Also, be sure to use the supplied IC socket for the microcontroller.

Once everything is in place, apply a **regulated** (ie, from a three or four cell battery pack or regulated plugpack) DC voltage of up to around 5 volts and press the button. The LED should start going through an animated sequence (the microcontroller is supplied with a sequencing program already programmed in). If nothing happens, check your work, making sure there are no shorts between tracks or pads anywhere, and the micro is the correct way around. Also check that the pushbutton input pin on the micro changes from 0 to 5 volts when you press the button. Another test is to measure the voltages on the output pins. If they are changing, but the LED is not glowing, then the LED is most likely in the wrong way around.

Something to be aware of is that the Picaxe08M can behave strangely or even be erased, losing its bootstrap program and rendering it a plain Pic microcontroller rather than a Picaxe, when driven from a full 6 volts. If you suspect this is the problem, then use three batteries instead of four, or use rechargeable cells instead—this will give you a 4.8 volt supply.

The layout of the circuit board, reproduced approximately full size. Its tiny size allows it to be used in all manner of places—you could even wear it.



Programming

If you get bored with the pre-programmed sequence, you can program your micro with any sequence that will fit into the 256 byte program space. A number of programs are available for download from my website at www.ledsales.com.au/kit_examples/. Programming cables are available from my website also. However, you can make your own by using an old 9-pin serial cable (an old mouse is ideal for this) and a 3-pin header socket. Instructions for making the cable, as well as the programming software, are available on the Revolution Education website at www.rev-ed.co.uk

Parts list

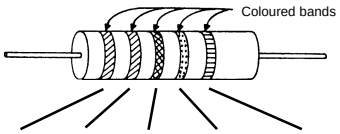
R1, R6 ..... 10k
R2 ..... 22k
R3 ..... 75 ohm
R4 ..... 100 ohm
R5 ..... 120 ohm
U1 ..... Picaxe08M
SW1 ..... Pushbutton
CN1 ..... 3 way header
LED1 ..... RGB 5mm or superflux LED

Identifying the parts of your Picaxe controller kit

The following guide should help you to identify the parts and assemble it successfully.

Resistors

The coloured bands represent numbers and multipliers as shown in the table. Some resistors have four bands (two digits, a multiplier band and a tolerance band) while some resistors have five bands. Five-band resistors are read in the same manner as four-band resistors, except that the first three bands are digits, the fourth a multiplier and the fifth the tolerance band.



Colour	Hundreds	Tens	Units	Multiplier	Tolerance
Black	0	0	0	1	20%
Brown	1	1	1	10	1%
Red	2	2	2	100	2%
Orange	3	3	3	1000	
Yellow	4	4	4	10,000	
Green	5	5	5	100,000	
Blue	6	6	6	1,000,000	
Violet	7	7	7		
Grey	8	8	8		
White	9	9	9		
Gold				0.1	5%
Silver				0.01	10%

RGB LED

The LED has four pins—one cathode (the negative pin) and three anodes, one for each colour. It is important that the cathode is placed in the negative (-ve) hole on the circuit board. The other three pins are bent to fit the other three LED holes in whatever way fits best, as which LED is lit is determined by the software that runs the picaxe so they don't have to go in specific holes.

The Picaxe08M

This is the 8 pin device. Note that it must be installed the correct way around. Pin 1 is marked with a small dot, and/or the pin 1 end of the chip will have a semicircular notch or mark. The IC socket also has this notch, so make sure it is installed the correct way around also, as this notch is used to indicate device polarity.