Controlling High Powered LEDs

by Ben Rowland, September2009

Abstract

LED's have always been a staple component in programming microcontrollers, they are most commonly used as indicators that elements of a program are functioning correctly. However, more people are realising that high powered LED's have a multitude of uses beyond the simple applications of their lower powered brethren. In this article Ben shows us how to design your circuit to utilise these functional devices.

Requirements

Software:

- Any licence type of any Flowcode version for any variant type.

Hardware:

- LED E-block

A very simple way of determining if a microcontroller is behaving correctly is through the use of LEDs. This could take the form of a simple "on light" to let you know there is power going to the device through to displaying binary numbers to indicate traffic on a communications bus. High power LEDs are now becoming more and more popular in embedded systems with prices falling as a result.

Here are a number of simple applications that could be created simply by using high power LEDs combined with a bit of ingenuity.

Multicolour lamp
Hi-power spotlight
Portable camping light
Propeller clock
PC, Car illuminations
Hydroponics grow systems

So to get started with using these hi power LEDs you first have to work out exactly what the LEDs need to be able to do.

1. Do the LEDs need to be a single colour or multicolour?
2. Do the LEDs all need to be the same brightness or should they be individually controllable?
3. Do the LEDs need to dim or are they constantly on?
4. What power source am I going to use to drive the LEDs?

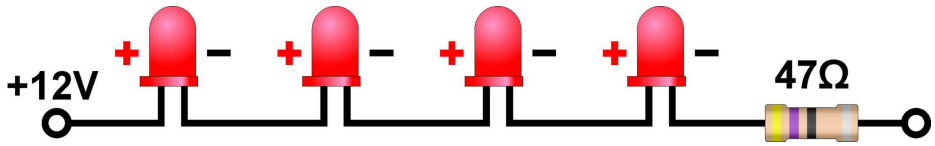
Firstly I am going to show you how to design a circuit to allow you to control the LED up to its full potential brightness without risking damage. Consider this device from rapid electronics, It is a Red LED capable of generating 20,000mCD with product code 55-2476. Looking at the device datasheet we can tell that the absolute maximum values for forward current are 50mA and forward voltage are 2.6V. It is best not to push the LED too far so maybe drop one of the values slightly to allow the LED a bit of tolerance. For this particular LED I will drop the forward voltage to 2.5V.

We can now use an LED wizard to design our circuit. Please see the references for the link to the wizard.

For this test circuit I am going to make a portable LED light. For this I am going to use a 12V lead acid battery to allow me to produce a long lasting bright light. So I put the following details into the calculator.

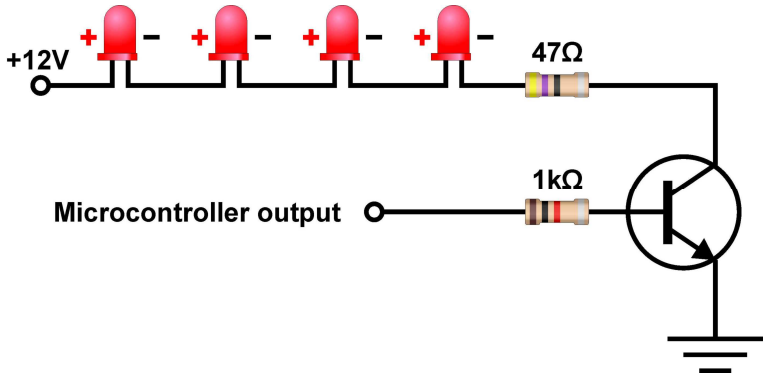
Source voltage - 12
Diode Forward Voltage - 2.5
Diode forward current - 50
Number of LEDs - 4

This generates a basic schematic to allow me to drive the LEDs at full brightness. Note that in this configuration all of the LEDs are tied together so only one control line is required. In other configurations it is best to remember that the resistor will always make your circuit less efficient so the more LEDs you can combine into a single channel the more efficient your circuit will be.

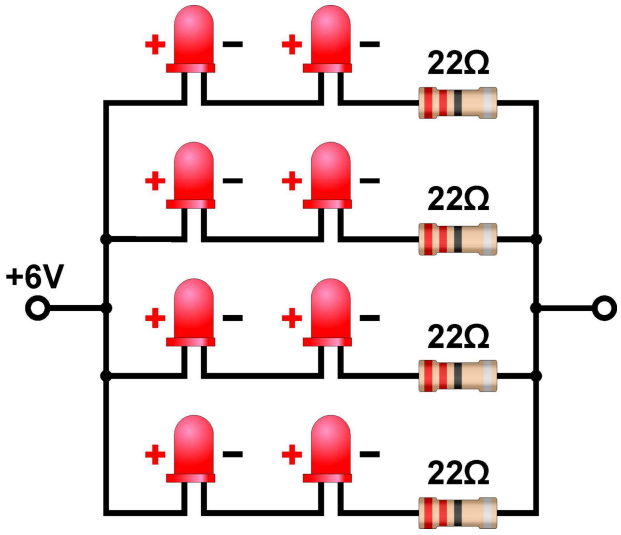


The LED wizard also tells me the current that the circuit is going to draw so I can calculate a rough estimate of how long my battery will last when powering the circuit. My battery is a 7.5Ah battery and my LED circuit consumes 50mA making for a maximum life expectancy getting on for 150 hours (7.5 / 0.05 = 150). Not bad for a luminous intensity of approx 80,000mCD.

To control the LEDs you can either use an onboard PWM peripheral or for more channels use a timer interrupt software driven PWM method.

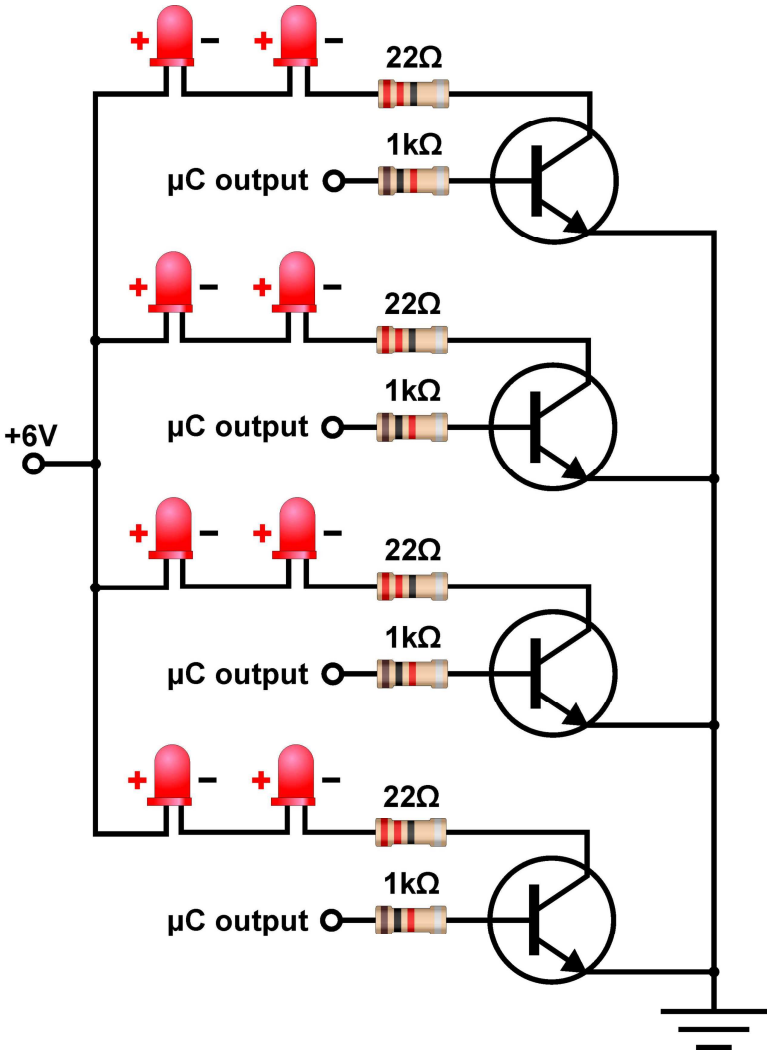


Here is another example using the same LEDs but in a single configuration. Notice that the supply voltage has been changed to better suit the configuration of the LEDs. We are now pulling 200mA through the circuit due to the increase in the number of resistors however we also have much more control over the LEDs as we can control each pair of LEDs separately.



7.5Ah divided by 200mA results in a life expectancy of 37.5 hours, still fairly impressive for a luminous intensity of up to 160,000mCD.

Which would translate into a circuit like this.



As this circuit uses four channels of output the PWM peripheral onboard the chip is not going to be much use. Therefore we need to use the software approach of creating the brightness control signal. To allow the software approach to work at low speeds such as 4MHz without flicker I have reduced the scale of the PWM from 0-255 to 0-63. This allows for a good flexibility of control at low speed without flicker. At a faster clock speed a larger resolution of control can be achieved without flicker.

To calculate the refresh period of the LED you would use the following equation.

((Clock speed / 4) / Timer_Period) / Resolution = Refresh rate (Hz).

Which works out as follows.

((4MHz / 4) / 256) / 64 = 61Hz refresh rate.

For any further questions on creating your own high powered LED circuits please visit the Matrix Multimedia user forums and join in the discussions there.

Further reading

Below are some links to other resources and articles on related subjects, and technical documentation relating to the hardware used for this project...

Flowcode:	http://www.matrixmultimedia.com/flowcode.php
ECIO:	http://www.matrixmultimedia.com/ecio.php
LED Wizard:	http://led.linear1.org/led.wiz
Learning Centre:	http://www.matrixmultimedia.com/lc_index.php
User Forums:	http://www.matrixmultimedia.com/mmforums
Product Support:	http://www.matrixmultimedia.com/sup_menu.php