

F2010 Abstract

"MagicLamp" an easy introduction into the Motorola HC08 family

Motorola made a small sized micro in a package with only 8 pins. But small sized does not mean low performance or limited development tools of the micro. In fact, Motorola made a bundle consisting of an intelligent micro, high performance development tools with an excellent cost effectiveness. Using the "68HC908QT4 Demonstration Kit" kit let us try out, how powerful and intelligent a micro project can be. The project should be used for demonstration and introduction into micro programming, so that absolute beginners can check it, too. On both sides, hardware and software, the project was as easy as possible designed.

The project needs the starter kit itself, a few external components (some transistors, resistors and three LEDs, red, green and blue) and a little piece of fantasy. We want to control the light intensity of the LED by software, so that we get a combination of fascinating light effects. Let us call the project "Magic Lamp". The project was inspired by a romantic love story, in which a nice and beautiful girl bewitches an absolute realistic police officer with her esoterically fragrant candle lights (at the end of the story, boy gets girl of course). That sounds a little bit strange, but let us figure it out, if we can combine science and fantasy into a micro-controller project.

Magic Lamp



The forward voltage of the LED depends on the colour. For example, red light emitting diodes have a forward voltage of about 2.5 V, green 3.9 V and blue 3.8 V (SuperFlux LEDs by LUMILEDS).

The main task of the "lamp controller" is controlling the light intensity of the chromatic LEDs. An easy way is the use of some resistors with different values for every LED. The controller switches one or more of the resistors for different light intensities (look on left side of figure 1).

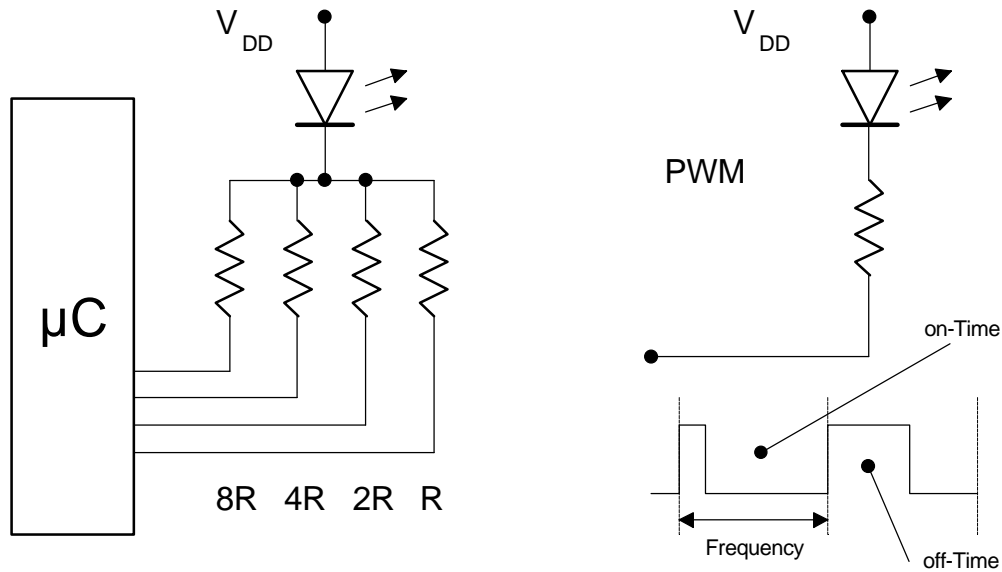


Figure 1 different methods for controlling light intensity

This method is very easy, and changing the light intensity is possible by changing the bit pattern on the micro port lines. Nevertheless, in the example the LED can be controlled in 16 steps. More steps need more port lines. For three LEDs with 256 steps per LED a number of 24 port lines is necessary (the MC68HC908QT4 in the 8 pin package has only 6 I/O lines).

Another way for light intensity control is the use of the pulse-width-modulation (PWM). The light intensity depends on the relation between on and off-time of the pulse signal. The frequency of the signal has to be high enough, so that the eye cannot detect any flickers.

Generating an PWM signal requires a little bit of software overhead. With the help of a software loop or a timer the signal has to be switched on and off cyclically.

3 Hardware

For the project we need the starter kit, three transistors, a small sized prototype PCB, some resistors and three LEDs. Please look at the schematic and see the simple hardware.

For demonstration use only, the transistors are not required. The I/O port line of the MC68HC908QT4 has a sink/source capacity of about 20 mA, so that the LED can be connected directly.

For better view it is recommended to spend more power on the LED. Let us use high efficiency LEDs by LUMILED. With a forward current of 35 mA you get an impressive light intensity by the diodes.

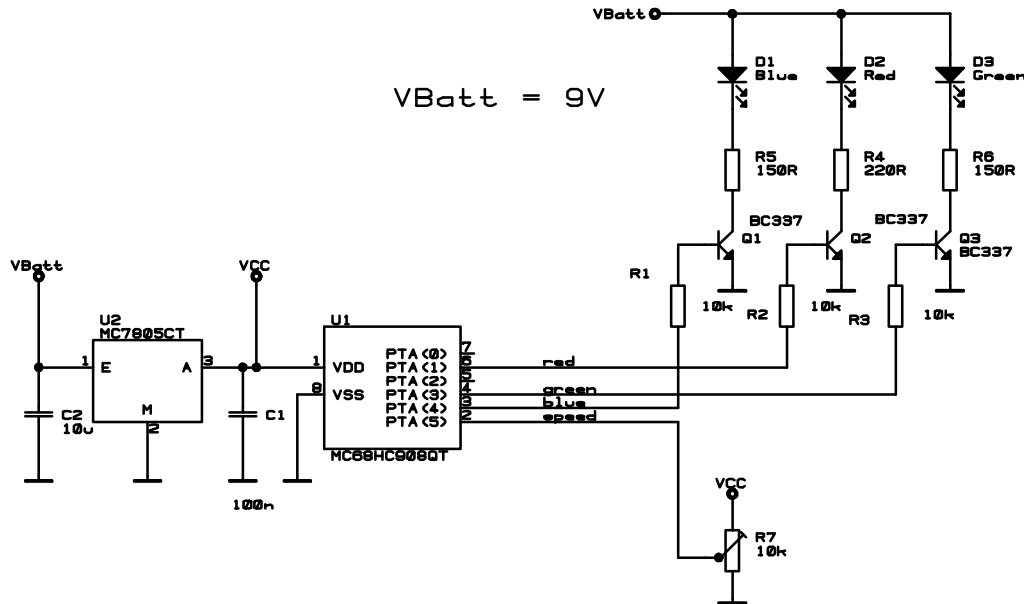


Figure 2 Schematic of the "Magic Lamp"

Figure 2 shows the complete hardware of the "Magic Lamp". If you use the starter kit as controller unit, you have to connect the specified I/O lines with the dedicated chromatic LEDs.

4 Software

Because of the simple hardware the software also should be as easy as possible, so that the complete project can be built by (absolute) beginners over the weekend

At first the Metrowerks-IDE has to be installed on the PC. Please look at the installing note on your starter kit CD. At least you have to ask about a license key for the Metrowerks C compiler (look at Metrowerks web site).

The Metrowerks IDE is a very powerful design tool with a huge amount of features. The large number of features sometimes makes the use of the tool a little bit confusing (especially for beginners). The best way for creating an own project, is to copy an existing one into a new folder.

Before we start programming, some notes about using C for microcontroller. Experienced PC users use a lot of pre defined functions or libraries when they program on a PC based platform. Simple I/O operations are done by *printf()*, *sprintf()*, ... etc, for example. Even mathematical or graphic functions are often used.

On micros most of the resources, like RAM and ROM are strictly limited. If you imagine that a floating point value needs at least 8 byte RAM you know that the uncritical use of resources