

Dokumentation RGB Led Dimmer

Ziel:

Ich wollte eine RGB Led ansteuern, und zwar nicht nur den Farbton, sondern auch die Helligkeit. Es gibt im Web zwar einige RGB-Kontroller Projekte, doch habe ich kein einziges gefunden, welches auch die Helligkeit dimmen kann.

Generelle Beschreibung:

Die Software ist in C geschrieben und steuert eine RGB Led an.

Momentan dimmt das Programm 10 Farbtöne (36° Abstände) automatisch...

Natürlich kann die Auflösung drastisch erhöht werden. Ich habe es aber so gewählt damit das Video nicht allzu lange wird.

Farbton und Helligkeit werden im HSV-Farbraum angegeben:

$$H \in [0, 360], S, V, R, G, B \in [0, 1]$$

Die Routine HSV_to_RGB() rechnet die HSV Werte schlussendlich von [0,1] in 8-Bit RGB Werte um. Im Timer0 Overflow Interrupthandler wird dann die Led über PWM angesteuert. Mit H kann also der Farbton gewählt werden. S ist die Sättigung und sollte man meiner Meinung immer auf 100%=1 lassen. V ist die Helligkeit und kann beliebig verändert werden.

Das ganze befindet sich noch im Entwurfsstadium, da aber jemand danach geschrien hat, veröffentliche ich mal den aktuellen Stand.

Hardware:

Atmega 16 uC, interner 8MHz Takt.

Port A.0-2 steuern über 4.7 k Ohm Widerstände npn Transistoren an.

Kollektor je an Vcc, Emitter über entsprechender Widerstand an die entsprechende Anode der Led. Gemeinsame Kathode der Led an GND.

Emitterwiderstände:

Für R 100 Ohm .

Für G 750 Ohm.

Für B 560 Ohm.

Meines Erachtens ist die Wahl der Emitterwiderstände von grosser Bedeutung, da RGB der Led unterschiedliche Leuchtstärken haben.

Da ich kein richtiges Datenblatt hatte bin ich folgendermassen vorgegangen:

R hat am wenigsten Leuchtkraft. Minimaler Widerstand gewählt → 100 Ohm.

Programm mit H=30, S=1, V=1 gestartet. Nun sollte die Led  so leuchten.
G Widerstand so lange verändert bis ich diese Farbe hatte.

Programm mit H=60, S=1, V=1 gestartet. Nun sollte die Led  so leuchten.
G Widerstand so lange fein verändert bis ich diese Farbe hatte.

Nun ist G auf R abgestimmt. Jetzt Fehlt nur noch die Abstimmung von R auf B.

Programm mit H=330, S=1, V=1 gestartet. Nun sollte die Led  so leuchten.
B Widerstand so lange verändert bis ich diese Farbe hatte.

Programm mit H=300, S=1, V=1 gestartet. Nun sollte die Led  so leuchten.

B Widerstand so lange fein verändert bis ich diese Farbe hatte.

Nun ist auch B auf R abgestimmt. Jetzt kommt der Test;
Programm mit H=0, S=1, V=1 gestartet. Nun sollte die Led möglichst weiss leuchten.

Farbwahrnehmung ist subjektiv, doch ich finde es ist mir ziemlich gut gelungen und ich bin zufrieden mit dem Resultat. ☺

HSV-RGB Realtime Vorschau Konverter auf
<http://www.csgnetwork.com/csgcolorsel4.html>

To do Software:

Leider ist mir die Softwarekonvertierung nur mit Integer, also ohne float nicht richtig gelungen, nach x Stunden Fehlersuche habe ich es aufgeben und mit float gerechnet. Dadurch bläht sich die Codegrösse zwar enorm auf, aber egal, der uC hat ja eh nichts Besseres zu tun. Ich bin ein blutiger C-Anfänger, was man bestimmt im Programm erkennt, wer es besser kann, darf gerne versuchen woran ich gescheitert bin.

Eine log. Umrechnung der V-Werte wäre auch nicht schlecht, somit wir die Helligkeitswahrnehmung für das Auge linear.

Die Software könnte man noch auf mehrere unabhängige Leds erweitern.
Ich habe noch 10 dieser Leds, mal schauen was ich damit anstellen werde. Zum Glück kann man dimmen, sie sind nämlich saumässig hell ☺

Mir schwebt ein RGB Led Cube vor, so auf Basis von $4 \times 4 \times 4 = 64$ Leds, in einem Plexiglas Gehäuse, jo das würde wohl nett Aussehen. Ausserdem ein Paar Taster / Potis um den Farbton und die Helligkeit zu wählen.

Copyright:

Da im Wörlid Weid Wäb sowieso Anarchie herrscht (Was auch ich gut finde), darf das Programm und diese Doku beliebig geklaut und modifiziert werden.
Ich würde mich allerdings auf Eure RGB Controller freuen. Ausserdem sind konstruktive Vorschläge und Kritik im Forum willkommen.

Dankeschön:

An Karl Heinz Buchegger, ohne ihn wäre ich gar nie auf die Idee gekommen, dass das Dimmen mittels HSV und umrechnen nach RGB ans Ziel führt.

Quellen:

Wie man konvertiert und was genau der HSV-Farbraum ist wird hier gut erklärt:
http://en.wikipedia.org/wiki/HSV_color_space