

8-fach 12-Bit Soft-PWM LED-Dimmer für mega48

Der Dimmer ist für LED's gedacht, deren Helligkeit logarithmisch (also der menschlichen Wahrnehmung folgend) gedimmt werden sollen.

Dazu werden die gewünschten Helligkeitswerte als 1-Byte-Wert je LED an den Dimmer geschickt, der sucht sich aus einer Tabelle den zugehörigen Compare-Wert aus dem 12-Bit Raum (0..4095) und füttert damit den Timer.

Die Daten werden alternativ via TWI oder RS232 entgegengenommen.

Es können bis zu 8 LEDs bedient werden (wenn die serielle Schnittstelle verwendet werden soll, dann entfällt leider eine LED am Pin RxD).

Der Controller läuft mit 20MHz, bei einem TOP-Compare-Wert von 4096 und einem Prescale von 64 ergibt sich eine Wiederholungsrate von 76.3 Hz ($20.000.000 / (4096 * 64)$). Als Controller kann ein mega48 oder mega8 verwendet werden, der Speicherbedarf liegt bei unter 3000 Byte.

Ein kleinere Prescale-Wert (das wäre 8) ist nicht möglich, da zwischen zwei Compare-Werten nur 8 CPU-Takte verbleiben, in dieser Zeit können aber die notwendigen Änderungen der Register nicht einmal in Assembler durchgeführt werden.

Das Dimmen erfolgt mit Hilfe des Timers1 und des COMPA-IRQ.

Damit das für 8 PortPins funktioniert, muss einiges an Vorarbeit getan werden:

die Schaltvorgänge sind in zeitlicher Abfolge (=Helligkeit) zu sortieren und der jeweils erforderliche LED-Portstatus ist zu ermitteln.

Die Interrupt-Routine muss knapp gehalten werden, da sie innerhalb von 64 Takten einen neuen Compare-Wert und die Portpins schreiben muss, ansonsten kann der nächste Schaltvorgang verloren gehen.

Zusätzlich zum 'normalen' Schalten der LEDs ist ein Fading eingebaut:

zwischen den zuletzt verwendeten Helligkeitswerten und den neu angegebenen kann ein linearer Übergang erzeugt werden.

Die Anzahl der Zwischenschritte kann von 3 bis 255 frei gewählt werden.

Es werden 76 Übergänge pro Sekunde erzeugt, bei einer Schrittzahl von 255 dauert der Übergang also über 3 Sekunden.

Soll der Übergang noch länger dauern, dann kann noch eine Verzögerungen zwischen den Werteänderungen eingebaut werden (es wird dann nur nach x.ten PWM-Neustart ein neuer Helligkeitswert gerechnet).

Um Werte an den Dimmer zu senden, ist folgende Sequenz zu übergeben (unter der Annahme, dass 7 LEDs definiert sind);

TWI: TWI_ADR, Parameter, LED0, LED1, LED2, LED3, LED4, LED5, LED6
 2, 0, 255, 255, 192, 0, 0, 255, 155

Seriell: Parameter, LED0, LED1, LED2, LED3, LED4, LED5, LED6
 0, 255, 255, 192, 0, 0, 255, 155

Der Parameter hat folgende Bedeutung:

- 0 die Helligkeitsdaten werden sofort übernommen
- 1 das nachfolgende Byte legt eine Verzögerung beim Dimmen fest, es wird nur 1 weiteres Byte ausgewertet und in eine Variable geschrieben! (der Wert 1 verdoppelt, der Wert 2 verdreifacht
- 2 Der Dimmer verfällt in den Zufallsmodus. Das Abschicken eines anderen Befehls beendet diesen Modus, es wird aber der gerade laufende Übergang erst beendet (das kann einige Sekunden dauern).
- 3-255 die Schrittzahl für das Überblenden zum angegebenen Helligkeitswert

Hinweis:

Es müssen immer genausoviele Werte an den Dimmer geschickt werden wie die Summe der (Leds + 1) ergibt (die Empfangsroutine erkennt den Eingang eines neuen Kommandos an der Zahl der eingegangenen Bytes).

Zum Thema Soft-PWM und LED-Fading gibt es in der Artikelsammlung einige ausführliche Beiträge.

Die Berechnung der COMPA-Werte-Tabelle wurde nach den Angaben aus dem LED-Fading Beitrag durchgeführt.

Die zugehörige Excel-Tabelle habe ich beigefügt.

Wenn etwa ein Dimmen mit 100Hz für nötig gehalten wird, dann kann man die geänderten Rahmenbedingungen in der Tabelle ändern und sich die neuen Wert in die Headerdatei exportieren

(Anmerkung: um 100Hz zu erreichen, müsste man den TOP-Wert reduzieren auf:
 $20.000.000 / (100 * 64) = 3125$).

Achja - zur Schaltung.

Alle LEDs werden am PortD angeschlossen und sind alternativ gegen VCC oder gnd geschaltet (zum Umschalten gibt es ein #define HIGHSIDE, das man bei Bedarf auskommentieren kann).

Bei den kleinen 2mA-LEDs lässt sich der Dimm-Effekt nur erahnen.

Besser sind die üblichen 1W-LEDs - aber die benötigen natürlich noch jeweils einen FET (wie etwa BUZ11) zur Ansteuerung.

Wenn der Dimmer über RS232 angesprochen werden soll, dann will PD0 die Daten annehmen und steht nicht mehr für eine LED zur Verfügung.

Über TWI erfolgt der Anschluss an den PC4/PC5 (Pullups nicht vergessen).

Ein 20MHz Quarz ist Pflicht, weil bei den 8Mhz des eingebauten Oszillators die LEDs fürchterlich flackern.

Und für den Empfang via RS232 ist ein Quarz ebenfalls notwendig.

Im beigefügten Programmcode sind 3 LED's an PD5, PD6 und PD7 voreingestellt, um den Dimmer anzusprechen müssen also 4 Byte (ggf. vorher die TWI-Adresse) abgeschickt werden.

Michael S.