

# Mikrokontroller Beschaltung

Datum: 2023-12-27  
Author: [engelbert.gruber@htlinn.ac.at](mailto:engelbert.gruber@htlinn.ac.at)

## Inhaltsverzeichnis

<b>1 arduino IDE installieren</b>	<b>1</b>
1.1 Installation	1
1.2 Ausprobieren : Blinken	1
1.3 Blinken mit Helligkeitsregelung	2
1.4 PWM mit analogWrite	3
1.5 mehr LEDs	3
1.6 Stromquelle	3
1.7 geregelte Stromquelle	4
<b>2 Geregelte Spannungsquelle</b>	<b>5</b>
<b>3 TODOs nächste Schritte</b>	<b>6</b>
<b>A Schaltplanzeichen</b>	<b>7</b>
A.1 ASCIIART	7
A.2 circuitkz	7
<b>R Revisionslog</b>	<b>8</b>

## 1 arduino IDE installieren

### 1.1 Installation

Download: für Windows gibt es App, MSI or Zip.

- Zeige: Menü - File - Examples
- Dateibenennung mit Datum im Namen
- Tools - Autoformat (Strg-T) ... Format erleichtert das Lesen.

### 1.2 Ausprobieren : Blinken

Das Beispiel: File - Examples - Basic - Blink laden

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

### 1.3 Blinken mit Helligkeitsregelung

Wir blinken schneller, so schnell, dass man das Blinken nicht sieht, 99 fps.

Pseudocode

```
hell    = HIGH;
dunkel = HIGH; delay(1);  LOW; delay(10);
```

arduino c++

```
const int Bright = 9;
const int Dimmed = 10;
void setup() {
  pinMode(Bright, OUTPUT);
  pinMode(Dimmed, OUTPUT);
  digitalWrite(Bright, HIGH); // immer ein
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(Dimmed, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1);                  // wait for a second
  digitalWrite(Dimmed, LOW); // turn the LED off by making the voltage LOW
  delay(10);                 // wait for a second
}
```

mit einem Oszilloskop anschauen.

Arbeitsablauf

- Titel: Helligkeitssteuerung mit dem  $\mu\text{C}$
- Beschreibung / Erklärung:
  - Helligkeit = Lichtenergie die auf das Auge trifft
  - Energie = Leistung x Zeit
  - Leistung  $P = U \times I$
  - $U$  = die Diodendurchgangsspannung = ziemlich konstant
  - $I = (5\text{V} - \mu\text{C-Ausgang} - U_{\text{diode}}) / R$  ... alles konstant
- bleibt zur Regelung "t" ... Pulsweite
- Schaltplan : zwei LEDs mit Vorwiderstand (berechnen) am  $\mu\text{C}$
- Plan für Aufbau am Steckbrett ...
- Aufbauen
- Programm : eine LED permanent ein, die andere dimmen/schnell blinken
- Messaufbau mit Oszilloskop.
  - Plan Zeichnen
  - aufbauen
  - messen
  - Ergebnis dokumentieren.

## 1.4 PWM mit analogWrite

$\mu$ Cs unterstützen PWM in Hardware, weil das die PWM

- genauer (beim Servo wichtig) und
- vom Programmablauf unabhängig macht,  
einmal eingestellt läuft die PWM-Ausgabe vom Timer gesteuert ohne die CPU zu belasten.

Den selben Arbeitsablauf wie in 3.

## 1.5 mehr LEDs

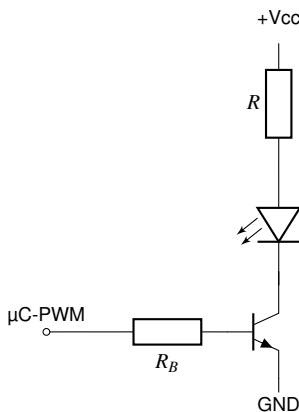
Der  $\mu$ C kann nur 40mA pro Anschluss (100mA an 8 Anschlüssen).

Wir müssen den Strom verstärken ... verstärken = Transistor (vielleicht)

Der bipolare Transistor

- hat 3 Anschlüsse und
- eine Stromverstärkung  $\beta$  ... das was wir brauchen.

Der Strom der in die Basis fließt erlaubt  $\beta$ -mal so viel durch den Kollektor . (aufzeichnen!)



- R begrenzt nach wie vor den Strom ...
- aber wir können mehr von +Vcc beziehen ... und deshalb mehrerer LEDs parallel schalten
- ACHTUNG: Belastung des Widerstands berechnen.

Wir können aber auch +Vcc grösser als 5V machen

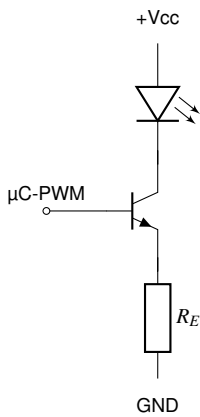
- zwei Versorgungsspannungen : 5V für den  $\mu$ C, 12V für die LED
- Diese Schaltung nennt man open collector, da der Kollektor des Transistors nicht im  $\mu$ C-Teil ist, offen.  
dann können wir mehrere LEDs in Serie schalten ... und brauchen nur einen Vorwiderstand für die LEDs
- ACHTUNG: Belastung des Widerstands berechnen.
- ACHTUNG: Der Transistor hat auch Grenzen: Leistung und Spannung.
- ACHTUNG: es braucht einen Basisvorwiderstand,  $R_B$ .  
Weil sonst 5V an der Basis des Transistors anliegen ... der eine Diodenstrecke ist, mit 0,7V maximal Spannung.

## 1.6 Stromquelle

Der Basisvorwiderstand ist in *Serie* zur Basisdiode ...

Bei einer Serienschaltung ist die Reihenfolge egal.

Was passiert, wenn wir  $R_B$  nachschalten ... nach dem Emitter? ... er wird zum  $R_E$  *UND*



- der Basisstrom fließt immer noch durch, es ist eine Serienschaltung.
- aber es fließt auch der Kollektorstrom durch ( $I_E = I_B + I_C$ )

$$I_C = \beta \cdot I_B \dots I_E = (1 + \beta) \cdot I_B$$

das bedeutet  $R_E$  sollte nur ein  $\beta$ -tstel von  $R_B$  sein um die selbe Wirkung auf  $I_B$  zu haben wie  $R_B$ .

$$U_B = 5V \text{ (}\mu\text{C-Ausgangsspannung)}$$

$$U_{BE} = 0,7V \text{ (Diodenstrecke)}$$

$$U_E = U_B - U_{BE} = 5V - 0,7V = 4,3V$$

$$I_E = 4,3V / R_E = I_C - I_B$$

$$I_C = I_{LED}$$

#### Regelwirkung:

- Wenn der Strom  $I_E$  steigt,
  - erhöht sich die Spannung am  $R_E$  und
  - damit wird  $U_{BE}$  kleiner,
  - damit regelt der Transistor den Strom hinunter,
- und umgekehrt.

Das ganze ist eine Stromquelle:

- $R_E$  bestimmt den maximalen Kollektorstrom ...
- ist unser Vorwiderstand für die LED *und* die Basis.

Wir können im Kollektorkreis die  $V_{CC}$  erhöhen und mehr LEDs einbauen, der  $I_{LED}$  bleibt der gleiche (ausser  $V_{CC}$  ist zu niedrig).

- ACHTUNG: Wir müssen mehr als 5V für die LED-Versorgung.
  - Weil  $U_{BE}$  vom  $\mu\text{C}$  5V ist,
  - $U_{BE}$  kleiner als 0,7V ist
  - die Spannung am Emitter,  $U_E$  ungefähr 5V-0,7V ist
  - $U_{CE}$  beim durchgeschaltene Transistor ungefähr 0,2V ist

ist der Kollektor der Stromquelle nicht unter 4,5V.

- Man kann die die  $U_{BE}$  niedriger machen (Spannungsteiler), dann muss  $V_{CC}$ -LED nicht so hoch sein.
- ABER: Wenn die  $U_E$  kleiner 1V, ergeben die Schwankungen der  $U_{BE}$  (vielleicht nicht mehr vernachlässigbare) Schwankungen im Konstantstrom.

## 1.7 geregelte Stromquelle

Wenn man die PWM durch einen Tiefpass glättet bekommt man Gleichspannung, so können wir eine geregelte Gleichstromquelle bauen ...

**Bemerkung:** die Nichtlinearität der Basis-Emitter-Diode macht das ganze interessanter, kann aber in der Software kompensiert werden.

Die Berechnung des Tiefpasses ist ein Abwägen zwischen Restwelligkeit der Gleichspannung und Trägheit des Amplitudenwechsels.

Je tiefer die Eckfrequenz desto weniger bleibt von den Pulsen der PWM sichtbar, aber desto langsamer reagiert das System auf Änderungen der Pulsweite.

## 2 Geregelte Spannungsquelle

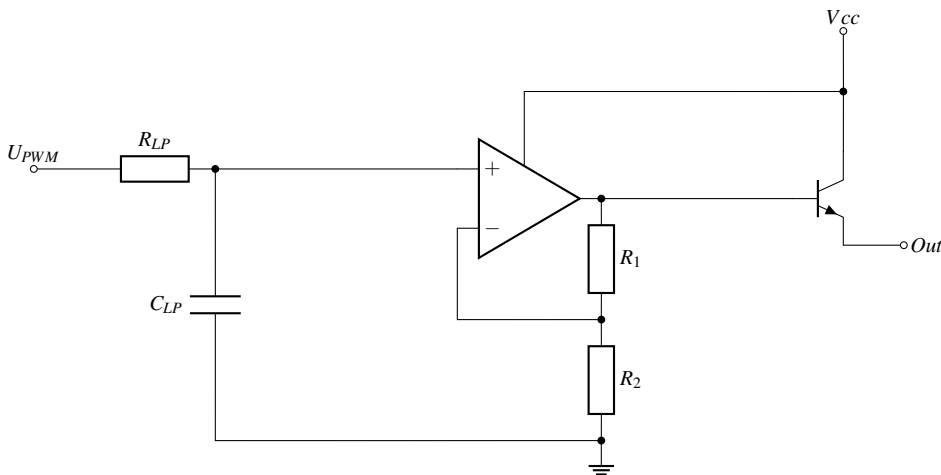
Wie müssen wir die Schaltung von der Stromquelle ändern wenn es eine Spannungsquelle werden soll?

Wir können nur den Transistor umdrehen beziehungsweise den Ausgang an den Emitter verlegen. Da  $U_{BE}$  ziemlich konstant ist, ist  $U_{Out}$  dann  $U_B - U_{BE}$  und wir können über  $U_B$  regeln.

Um mehr als 4,3V erhalten zu können müssen wir  $V_{cc}$  erhöhen **UND**  $U_B$  ...

- entweder mit einem Transistor verstärken
- oder einen OPV verwenden ...

der OPV als nichtinvertierender Verstärker ist ... einfacher.

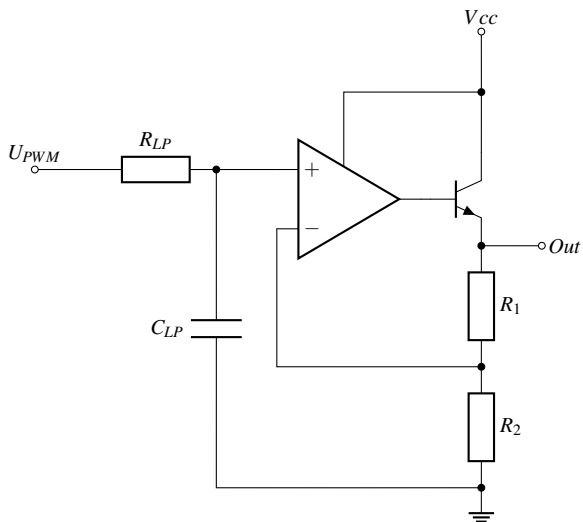


- an  $R_2$  haben wir die gleiche Spannung wie am nicht invertierenden Eingang ... 0 bis 5V,
- wenn wir bis 20V Ausgangsspannung haben wollen müssen wir 20,7V an der Basis können:  $U_{R1} + U_{R2} = 20,7V$  bei  
 $U_{NI} = 5V$   
 $U_{R2} = 5V$   
 $U_{R1} = 20,7V - 5V$   
 $R_1 = R_2 \times (20,7 - 5) / 5$

**ACHTUNG:** Der Transistor muss die Leistung und die Spannung vertragen ... Datenblatt konsultieren.

**ABER:** Eigentlich wollen wir ja die  $U_E$  ( $U_{Out}$  einstellen nicht die  $U_E + U_{BE}$ )

Wir können  $R_1$  an den Emitter anhängen, dann fällt der Einfluss von  $U_{BE}$  weg.



Dann regelt der Operationsverstärker die Emitterspannung ( $U_{Out}$ ).

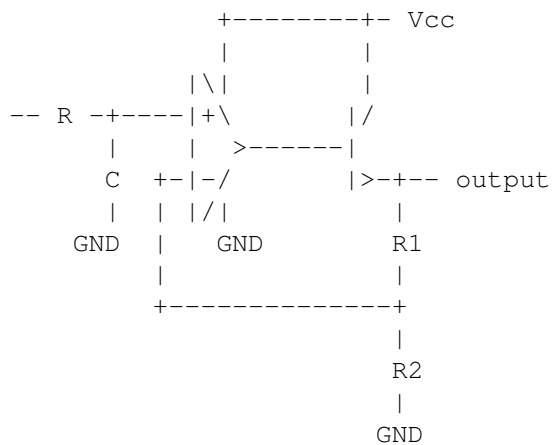
### 3 TODOs nächste Schritte

- besserer Titel ... viel Elektronik für Mikrocontroller
- Oszilloskop Bilder
- Bauteildimensionierungsbeispiele.
- ...

## A Schaltplanzeichen

### A.1 ASCIIART

Die erste Version



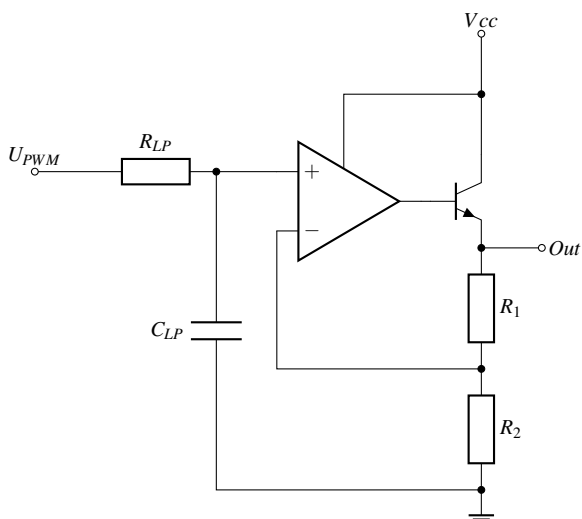
### A.2 circuitikz

Die schönere Version mit circuitikz

```

\begin{circuitikz}[scale=0.8, transform shape]
\draw (0,0)
node[above]{$U_{PWM}$} to[short, o-] ++(1,0)
to[R,l=$R_{LP}$] ++(2,0) coordinate(LP)
-- ++(1,0) node[op amp, noinv input up, anchor=+] (OA){}
(OA.out) ++(1,0) node[npn] (Q){}
(OA.out) to[short, -] (Q.B)
(Q.E) to[R,l=$R_1$] ++(0,-2) coordinate(FB)
to[R,l=$R_2$] ++(0,-2) coordinate(GND) node[ground]{}
to[short, *-] (GND -| LP)
to[C,-*,l=$C_{LP}$] (LP)
(FB) to[short, *-] (FB -| OA.-) -- (OA.-)
(Q.C) to[short, -o] ++(0,2) node[above]{$V_{cc}$}
++(0,-1) coordinate(VCC) to[short, *-] (VCC -| OA.up) -- (OA.up)
(Q.E) to[short, *-o] ++(1,0) node[right]{$Out$}
;
\end{circuitikz}
  
```

ergibt diese Zeichnung



## R Revisionslog

2023-12-27 Anhang mit Schaltplan und Revisionslog.