DAC - Digital to Analog Converter

Datum: 2025-03-05

Inhaltsverzeichnis

1 analogWrite im arduino **1**

2 dac am esp32 **1**

3 Eigenbau **2**
- 3.1 R2R-Netzwerk . . . 2
  - 3.1.1 Wie geht das leichter zu rechnen . . . 4
- 3.2 3Bit DAC diskret . . . 5
  - 3.2.1 Statisch testen . . . 5
  - 3.2.2 Test mit arduino . . . 6
  - 3.2.3 Glidges . . . 7
- 3.3 Aufbau mit Widerstands-IC . . . 10
- 3.4 Layout , Routing . . . 12

4 Anhang **13**
- 4.1 Gegenkopplung beim OPV . . . 13
- 4.2 Kirchhoff . . . 13
- 4.3 Helmholtz . . . 13

Um am Oszilloskop XY-Bilder anzeigen zu können, brauchen wir Analogspannungen.

1 analogWrite im arduino

analogWrite verwendet Pulsweitenmodulation: 500Hz 8bit

Das bedeutet die Auflösung ist maximal 5V/256, cirka 20mV.

Frequenzen über 10Hz sind problematisch.

Aufgabe 1: genaue Messung der Möglichkeiten:

- maximale Frequenz, Rise-/Falltime
- Auflösung

Aufgabe 2: kann man höhere Bitraten oder Frequenzen benutzen.

2 dac am esp32

der esp32 hat zwei DAC in Hardware eingebaut: 8bit Auflösung.

Aufgabe 3: Welche Frequenzen kann man noch ausgeben?

3 Eigenbau

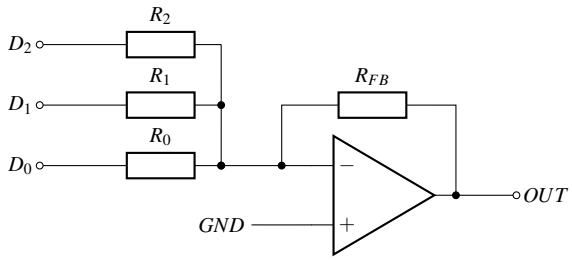
Wir haben n-digitale Eingänge und einen analogen Ausgang.

Das addieren von Spannungen ist ... schwierig, wenn sie den selben Bezugspunkt haben, weil man sie dann nicht in Serie schalten kann.

Wir haben: $U = R * I$

- R ist Hardware, wir können Widerstände stapeln ... das hilft nur bedingt.
- Ströme kann man gut addieren, das passiert an jedem Knotenpunkt.

OPV-Addierer



Da die digitalen Signale D0, D1, D2 alle die selbe Spannung haben, 0 oder 5 Volt, wandelt man diese mit R0, R1, R2 in Stöme der gewünschten Größe.

I2 ist doppelt so groß wie I1 und viermal I0.

Der Knotenpunkt am invertierenden Eingang des OPV wird von diesem auf GND gehalten (siehe Gegenkopplung beim OPV) deshalb beinflußt die Spannung an D2 den Strom I0 nicht.

Die Ströme I0, I1 und I2 können nur über R weiter fließen (laut Kirchhoff müssen sie das) deshalb ist

$$OUT = R * (I0 + I1 + I2).$$

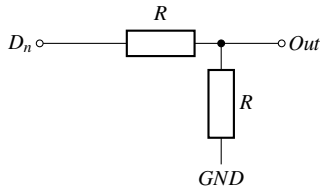
Für niedrige Auflösungen ist die Schaltung gut verwendbar. Wenn Rn aber in den Toleranzbereich von R0 kommt wird es schwieriger.

3.1 R2R-Netzwerk

Binär heißt: das höchstwertige Bit ist der halbe Zahlenwert

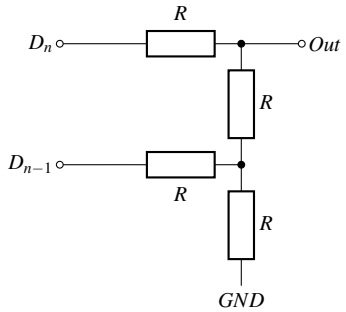
Zum Beispiel ein Byte kann 256 Werte darstellen, Bit7 hat den wert 128.

Wir beginnen:



Fertig: Wenn Dn High ist, ist am analogen Ausgang die halbe Spannung.

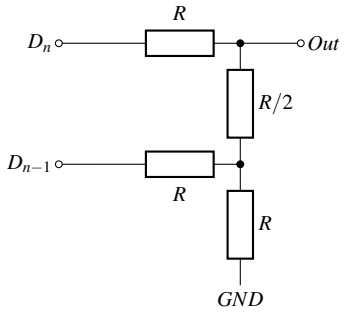
weiter (wir sind HTL-Schüler und probieren eine Wiederholung):



Blöd: Der Spannungsteiler für Dn ist jetzt $R-(R+R/2)$, das heißt OUT ist nicht mehr die Hälfte.

Wir nehmen an dass Dn-1 Low, also auf GND ist, dann sind die beiden R parallel (Helmholtz).

Quickfix:



Fertig: Wenn Dn High ist, ist am analogen Ausgang die halbe Spannung.

Helmholtz: Wir setzen Dn auf GND und Dn-1 auf HIGH, rechnen den Strom durch ... R beim Dn aus und daraus OUT.

- Ux, die Spannung am Knoten nach Dn-1:

$$Ux = HIGH * (R||(R+R/2))/(R+(R||(R+R/2)))$$

- Die Spannung OUT:

$$OUT = Ux * R/(R/2+R)$$

- Für HIGH = 1V, R = 1 Ohm :

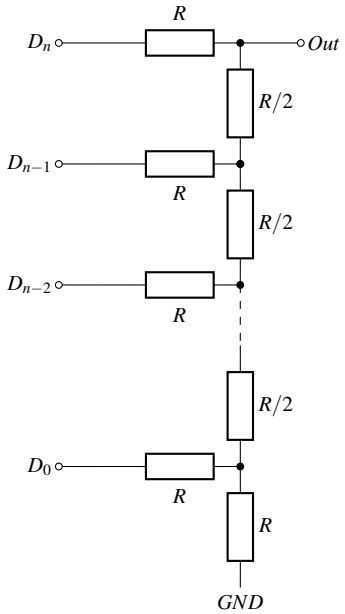
$$\begin{aligned} R||(R+R/2) &= (R*(R+R/2))/(R+(R+R/2)) \\ &= (1*1.5)/(1+1.5) \\ &= 1.5/2.5 = 0.6 \end{aligned}$$

$$Ux = 0.6/1.6$$

$$OUT = Ux*(1/1.5) = (0.6/1.6)/1.5 = 0.25$$

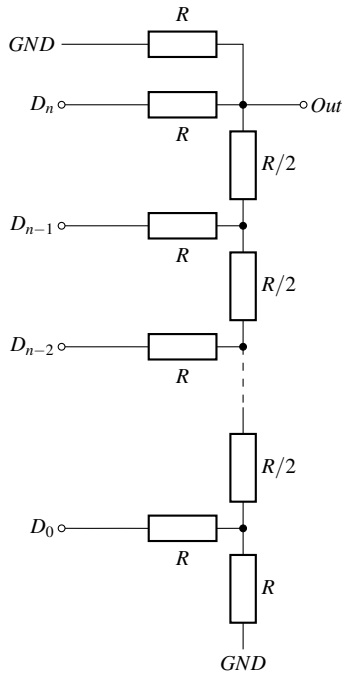
Bei Dn HIGH ist OUT 0.5, das ist doppelt so viel.

BINGO: !



3.1.1 Wie geht das leichter zu rechnen

Wir bauen noch einen Widerstand ein:



und denken um von Spannungen auf Strom, weil das mit Helmholtz günstiger ist.

Wenn Dn HIGH ist und Dn-1 LOW:

- am Knoten OUT nach oben R zu GND

  nach unten $R/2+R||R=R/2+R/2=R$

  das bedeutet der Strom teilt sich hier.

  OUT ist $R*I/2$

Wenn Dn LOW ist und Dn-1 HIGH:

- am Knoten nach Dn-1 nach unten R zu GND

  nach oben $R/2+R||R=R/2+R/2=R$

  das bedeutet der Strom teilt sich hier,

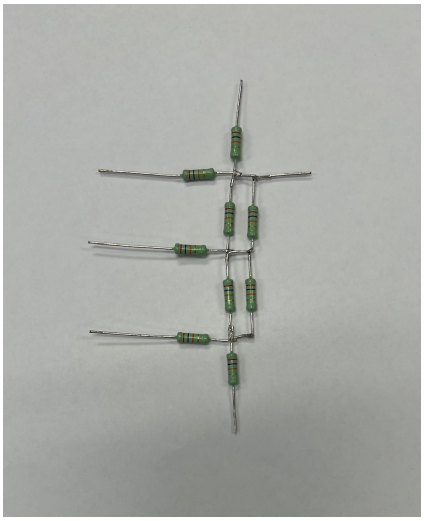
  kommt zum Knoten OUT hat nach links zu Dn ein R zu GND nach oben ein R zu GND ... der Strom teilt sich noch einmal

  OUT ist $R*I/2/2$

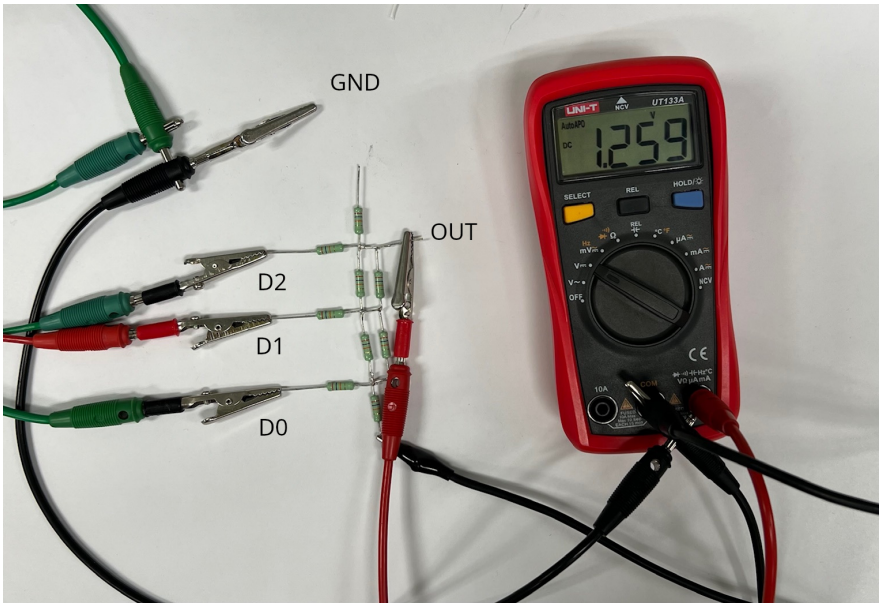
und das passiert bei jedem weitern Eingang.

Das oberste R zu GND kann man wie oben nach gerechnet weglassen.

3.2 3Bit DAC diskret



3.2.1 Statisch testen



Schaltung Variante B. Der obere Widerstand ist nicht auf GND.

- D0 und D2 (grün) sind auf GND
- D1 (rot) ist auf 5V

Alle Eingänge auf 0V, je einen Eingang auf 5V und OUT messen

5V Pin	D2/1/0	U OUT	U OUT B
D0	0 0 1	0.419	0.629
D1	0 1 0	0.839	1.259
D2	1 0 0	1.680	2.528

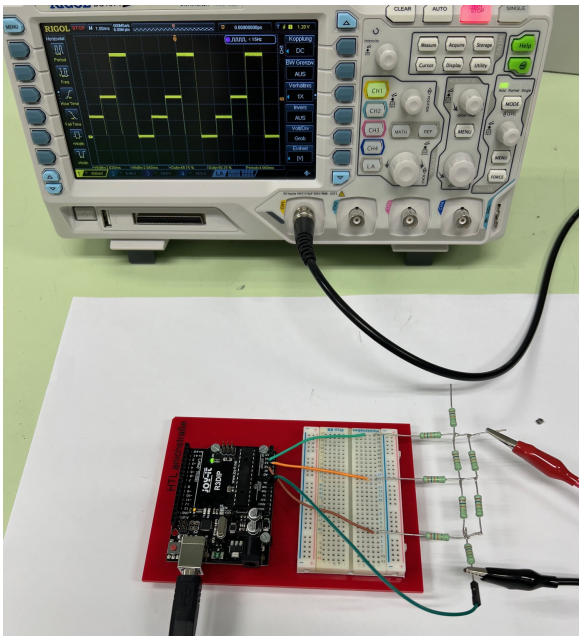
Die Spannung an OUT verdoppelt sich jeweils.

Bei Variante B, ohne der oberste Widerstand sind die Werte höher, aber auch jeweils das doppelte. Den Widerstand haben wir zum leichteren Erklären hinzugefügt haben, aber auch nachgerechnet, dass er für die Funktion nicht notwendig ist.

3.2.2 Test mit arduino

```
void setup() {
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);
}

void loop() {
  digitalWrite(A0, 0);
  digitalWrite(A1, 0);
  digitalWrite(A2, 0);
  delay(1);
  digitalWrite(A0, 1);
  digitalWrite(A1, 0);
  digitalWrite(A2, 0);
  delay(1);
  digitalWrite(A0, 0);
  digitalWrite(A1, 1);
  digitalWrite(A2, 0);
  delay(1);
  digitalWrite(A0, 0);
  digitalWrite(A1, 0);
  digitalWrite(A2, 1);
  delay(1);
}
```



Eine Rampe ausgeben in dem wir von 0 bis 7 zählen.

```
void loop() {
  static int v = 0;
  digitalWrite(A0, v & 1);
  digitalWrite(A1, (v >> 1) & 1);
  digitalWrite(A2, (v >> 2) & 1);
  delay(1);
  v = v + 1;
}
```

- `static int v = 0;` unsere Laufvariable, enthält den Zählerstand.
- `v & 1` v wird mit dem Wert 1 binär verundet, das bedeutet nur der Inhalt des Bit0 bleibt bestehen alle anderen

werden auf 0 gesetzt.

- `(v >> 1)` v wird um 1 Bit nach links verschoben, Bit 1 wird zu Bit 0 und so weiter.

3.2.3 Glitches

Im vorigen Oszilloskop-Bild sieht man

- Beim Wechsel von 1 auf 2 geht das Signal kurz auf 0, ebenso beim Wechsel von 3 auf 4.
- Beim Wechsel von 5 auf 6 geht das SIgnal noch einmal auf 4 zurück.

Das passiert, weil beim Wechsel von 1 auf zwei

- zuerst Bit 0 auf 0 gesetzt wird, dann ist das Muster D0/1/2 0,0,0 und damit die Spannung OUT 0
- dann das Bit 2 gesetzt wird, dann ist das Muster 0,1,0 und die Spannung 2.

Das selbe passiert beim Wechsel von 3 auf 4.

D2/1/0	an OUT
011	3
010	2
000	0
100	4

Der Wechsel von 7 auf 0 passiert in Schritten: 7,6,4,0

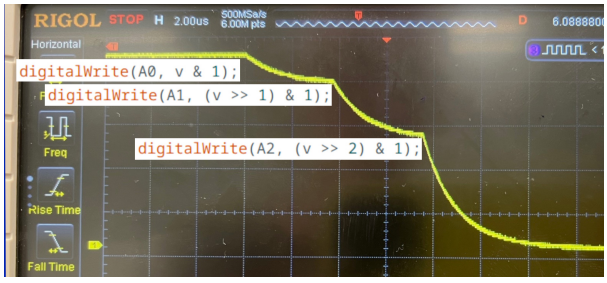
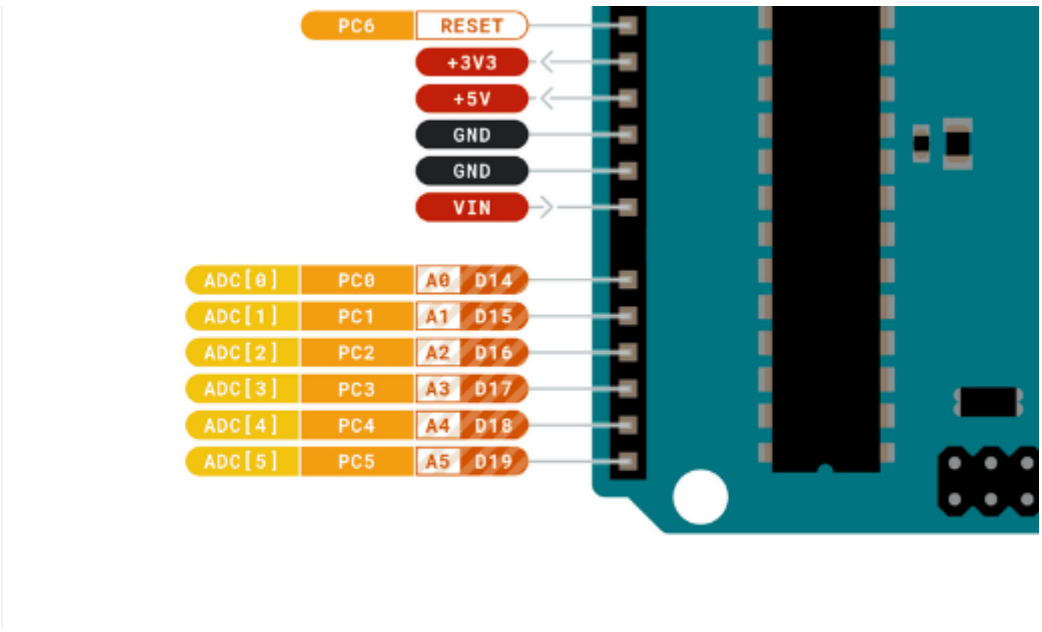


Abbildung 1: Horizontal 2µs/Div

Der Befehl `digitalWrite` benötigt ungefähr 3.8µs

Man kann das umgehen, wenn man alle 3Bits auf einmal setzt, wenn das möglich ist. Es ist weil A0,A1 und A2 alle am PORTC des Mikrokontrollers ATMega328 sind.



Man kann alle 3 Bit des Ports mit einer Zuweisung setzen.

```
void loop() {
  static int v = 0;
  PORTC = v & 0x07;
  delay(1);
  v = v + 1;
}
```

Die Flanke beim Wechsel von 7 auf 0 ist nicht mehr sakkadiert ... aber irgendwo ist ein Kondensator der entladen werden muss.

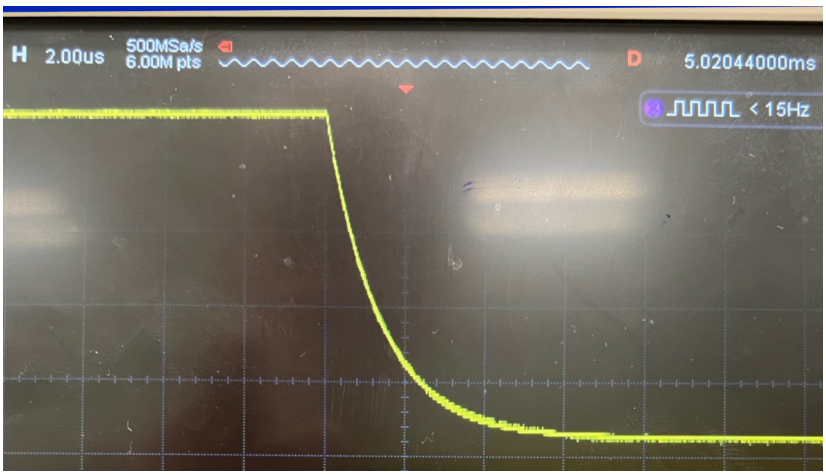


Abbildung 2: Horizontal 2µs/Div

Zeitkonstante, Abfall um 63% cirka 3V in 1.2ms.

Wenn man das Signal mit einer 10:1 Prüfspitze abnimmt wird die Entladekurve steiler.

Abbildung 3: Horizontal 200ns/Div

Das bedeutet die Eingangskapazität des Oszilloskops von 13pF muss über den DAC entladen werden. Ich habe in diesem Model 15kOhm Widerstände.

Wenn wir das Programm etwas beschleunigen, weniger bremsen, indem wir andstatt 1ms nur noch 1µs warten

```
void loop() {
  static int v = 0;
  PORTC = v & 0x07;
  delayMicroseconds(1);
  v = v + 1;
}
```

Abbildung 4: gelb - ohne Prüfspitze, hellblau - mit 10:1 Prüfspitze

3.3 Aufbau mit Widerstands-IC

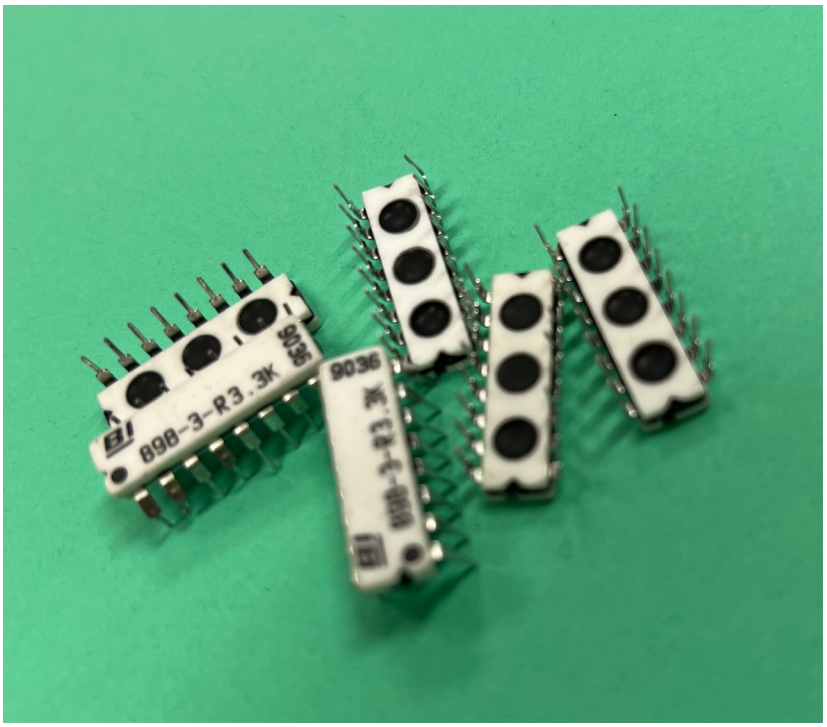


Abbildung 5: Modell 898 Resistor Networks

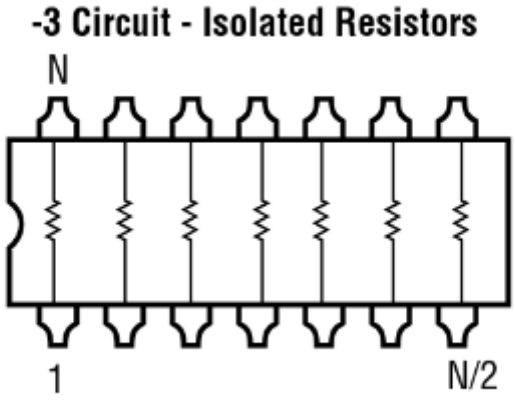
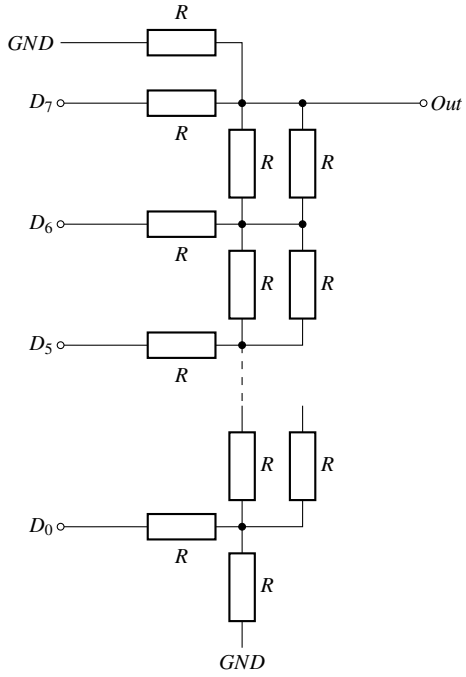


Abbildung 6: Der IC enthält in Variante "-3" 8 Widerstände.

Die R2R Variante mit den parallel geschaltenen R kommt mit weniger Widerständen aus.

Aufgabe 4: Wieviele Widerstände braucht es pro Bit-Auflösung ?



3.4 Layout , Routing

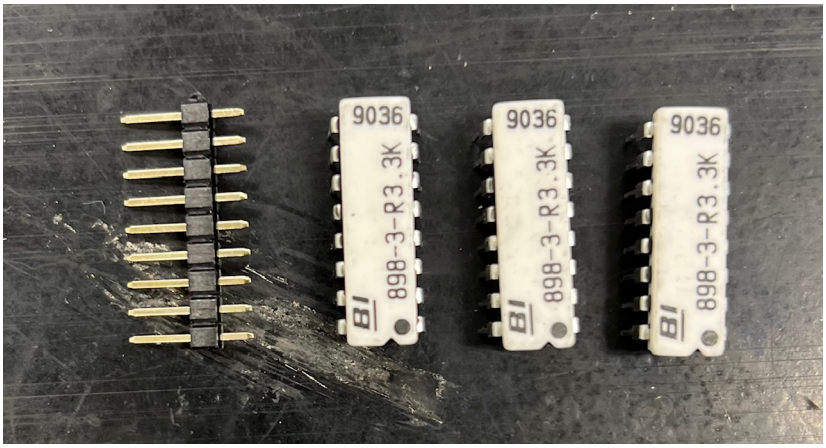


Abbildung 7: Die Bauteile und das ungefähre gewünschte Layout

TODO: Layouten beschreiben.

Aufgabe 5: mit drei ICs einen mindestens 6-Bit DAC bauen.

Statisch testen:

Alle Eingänge auf 0V, je einen Eingang auf 5V und OUT messen

5V Pin	U OUT
0	
1	
2	
3	
4	
5	
6	
7	

Aufgabe 6: mit arduino ein Signal ausgeben und am Oszillosop anschauen.

```
void setup() {
  // alle pinMode OUTPUT
}

// rampe ausgeben
void loop() {
  // eine Variable von 0 bis 255 zaehlen
  // die Bitwerte auf die pins schreiben
}
```

Linearität kontrollieren

Any glitches, digitalWrite braucht wie lange ?

Direkt zum Port schreiben ... nur 6 bit am arduino.

Aufgabe 7: Hardware DAC in Betriebnehmen

Latches im 12 bit Hardware DAC anschauen

Aufgabe 8: mit sechs ICs einen 16-Bit DAC

Messen

Aufgabe 9: mit neun ICs einen 24-Bit DAC

Stückpreis bei mouser 50EUR.

Messen

4 Anhang

vielleicht in externe Dokumente.

4.1 Gegenkopplung beim OPV

TODO

4.2 Kirchhoff

Die Summe der Ströme in einem Knoten ist Null.

Anders gesagt, alles was hineingeht muss irgendow heraus, wenn es nicht drinnen sitzen bleibt.

4.3 Helmholtz

Überlagerungsprinzip nach Helmholtz.

Man ersetzt alle bis auf eine Spannungs- und Stromquelle durch deren Innenwiderstand.

Berechnet den Strom durch einen günstig liegenden Widerstand.

Wiederholt das für jede Quelle.

Summiert die Ströme und rechnet die Spannung aus.