

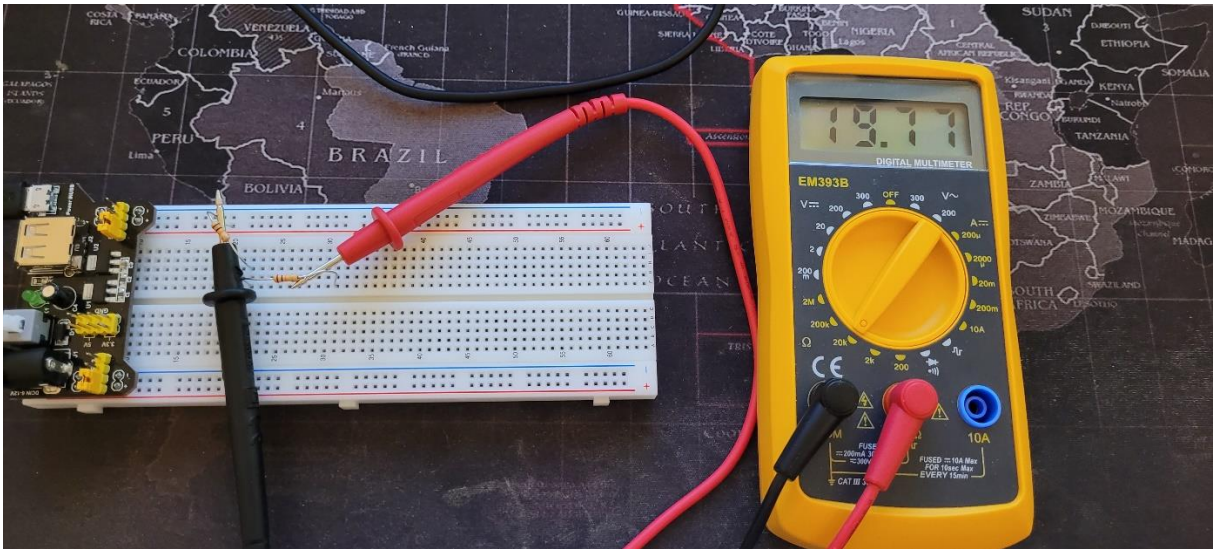
Übung 2

1.

1.1. Der Gesamtwiderstand in Reihe geschalteter Widerstände ist die Summe der Teilwiderstände. Hier: $R_{Total} = \sum_{i=1}^n R_i = R_1 + R_2 = 10k\Omega + 10k\Omega = 20k\Omega$. Somit ist der Gesamtwiderstand $20k\Omega$

Messung: $19.77k\Omega$

Da die Widerstände eine Toleranz von 5% (goldener Ring) hat, ist der Wert den Erwartungen entsprechend:

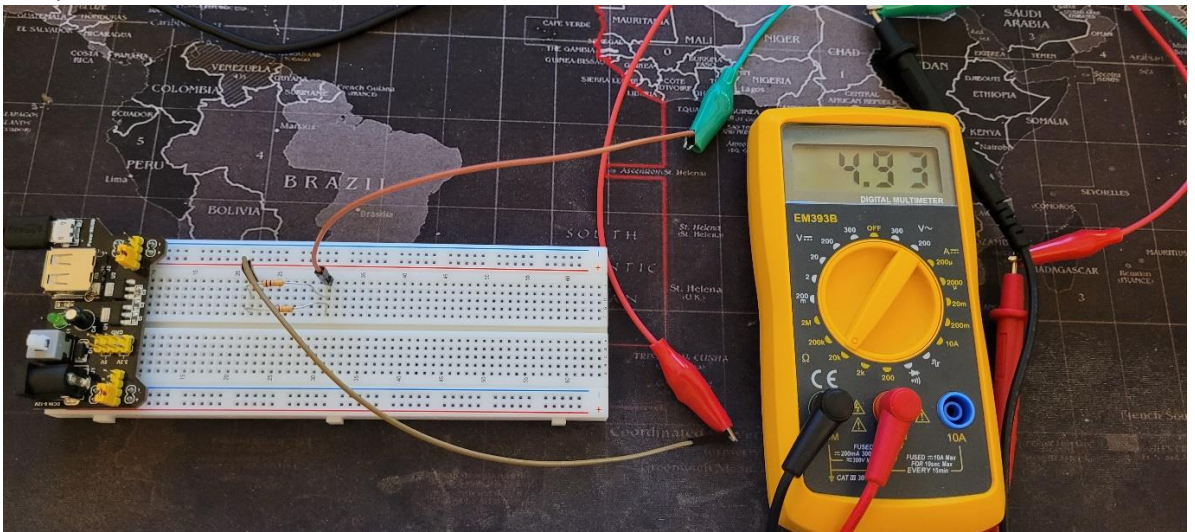


1.2. Der Kehrwert des Gesamtwiderstands parallel geschalteter Widerstände ist die Summe der Kehrwerte der Teilwiderstände. Hier: $R_{Total} = \sum_{i=1}^n \frac{1}{R_i} = \frac{1}{R_1} + \frac{1}{R_2} = \frac{1}{10k\Omega} + \frac{1}{10k\Omega} = \frac{1}{5k\Omega}$.

Somit ist der Gesamtwiderstand $5k\Omega$.

Messung: $4.93k\Omega$

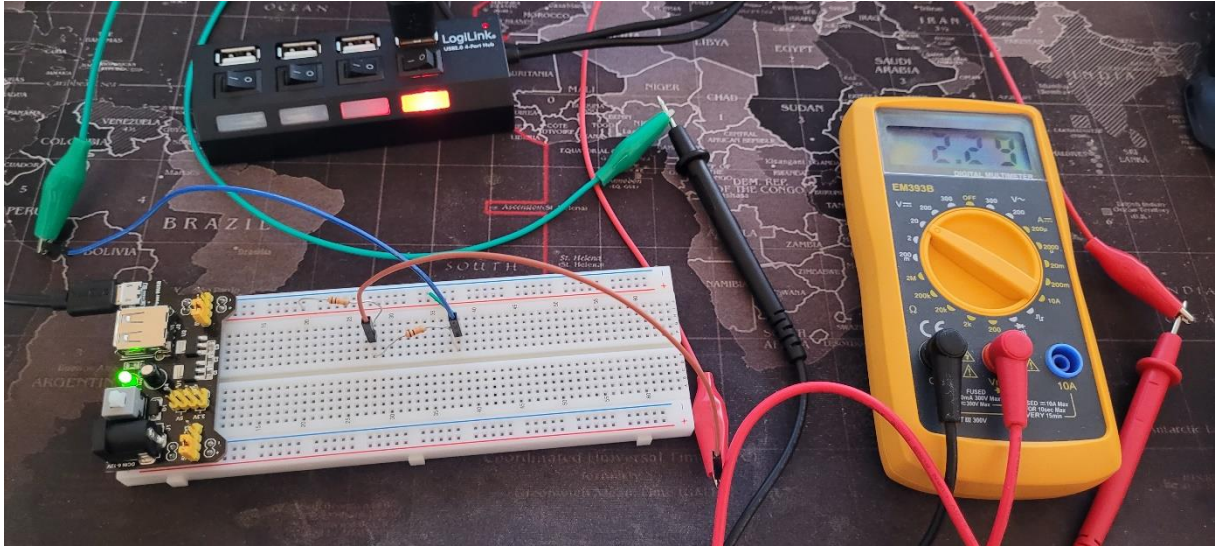
Da die Widerstände eine Toleranz von 5% (goldener Ring) hat, ist der Wert den Erwartungen entsprechend:



1.3. Die Berechnung des Spannungsteilers funktioniert analog zur Reihenschaltung: $\frac{U_i}{R_i} = \frac{U_{Total}}{R_{Total}}$.

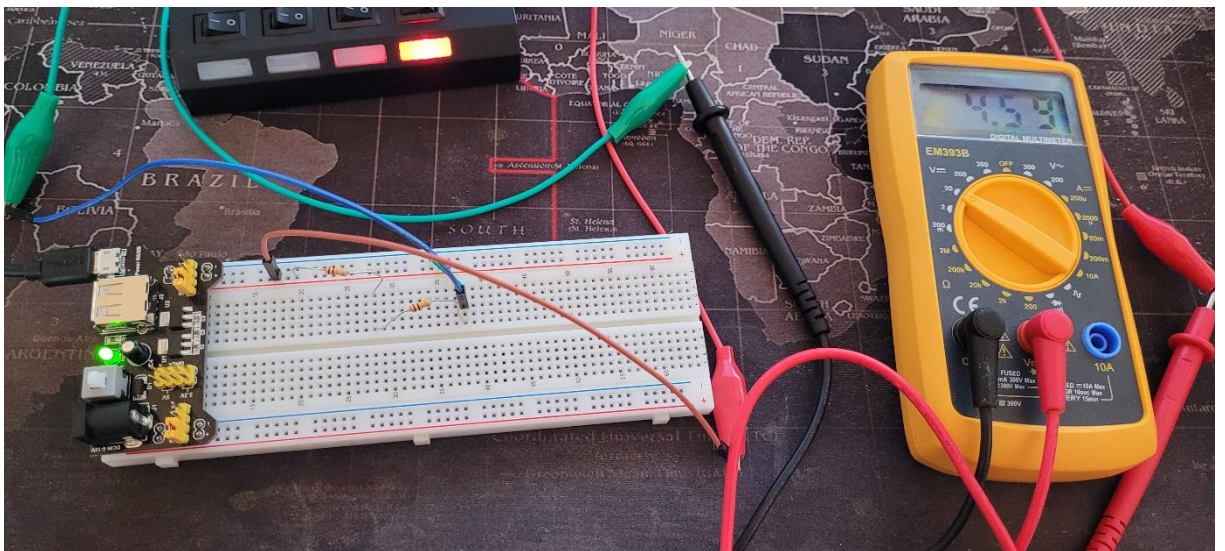
Für U1 gilt daher: $\frac{U_1}{10k\Omega} = \frac{5V}{20k\Omega} = 2,5V$

Messung: 2.29V



Für U2 gilt daher: $\frac{U_2}{20k\Omega} = \frac{5V}{20k\Omega} = 5V$

Messung: 4.59V



1.4.

1.4.1. Ich baue weiteren Widerstand ein. Damit sinkt die Menge des Stroms, der bei der LED ankommt. Am besten wäre es wahrscheinlich, den Widerstand in Form eines Potentiometers einzufügen, dann kann die Helligkeit flexibel eingestellt werden, ohne weitere Änderungen vornehmen zu müssen.

1.4.2. Ich schalte insgesamt 6 Batterien mit 1.5V in Reihe. Dann summiert sich die Spannung der Batterien auf 9V und der Motor kann betrieben werden. Die Kapazität entspricht dann allerdings der einer einzigen Batterie.

1.4.3. Schaltet man Batterien parallel, bleibt die Spannung gleich, die Kapazität erhöht sich aber entsprechend. Um die Kapazität für den Motor adäquat zu erhöhen, müsste man allerdings 6 weitere, in Reihe geschaltete Batterien parallelschalten. Würde man z.B. nur 5 weitere parallelschalten, würde nach dem Aufbrauchen des ersten Packs nur noch 5 weitere zur Verfügung stehen, also nur eine Spannung von 7.5V, was wiederum nicht zum korrekten Betrieb des Motors reicht.

1.4.4. Man könnte die LED's in Reihe schalten. Vor die erste LED kommt wieder ein Widerstand. Anschließend werden die LED's in Reihe dahinter gepackt. Recherche und ein Test auf dem Breadboard haben ergeben, dass der Strom mit jeder LED ein bisschen weniger wird (1. LED 2.03V, 2. LED 1.97V) und die LED's immer ein wenig dunkler werden.

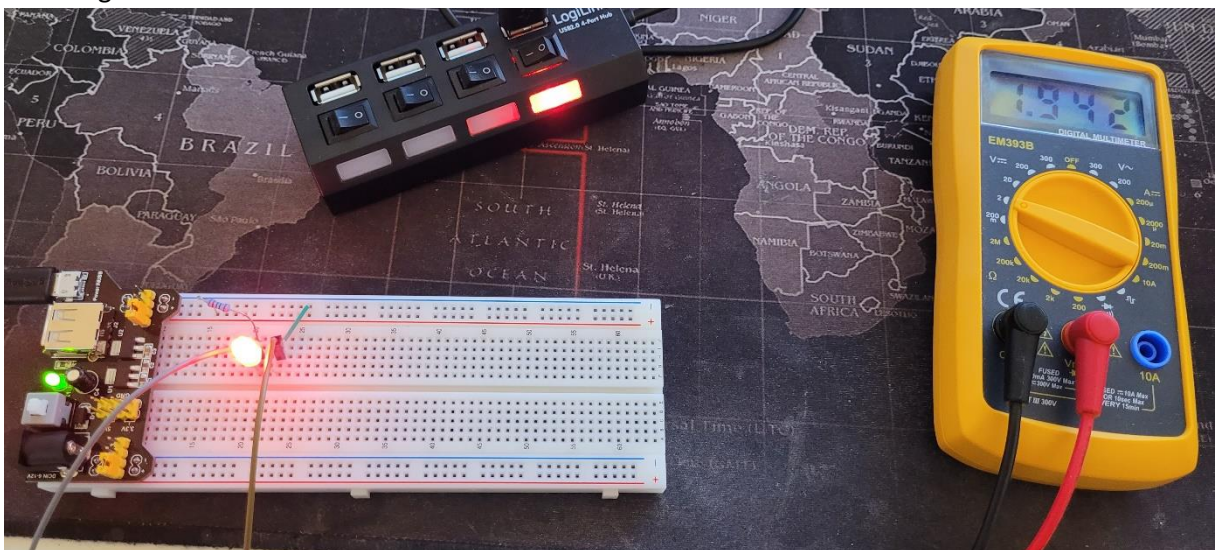
2. Zieht man den Kondensator ab und hält ihn an die Beine einer LED, entlädt sich der Kondensator schlagartig und die LED leuchtet kurz (halbe Sekunde) hell auf, bevor die Helligkeit rapide abnimmt, bis sie wenig später erloschen und der Kondensator vollständig entladen ist.

3.

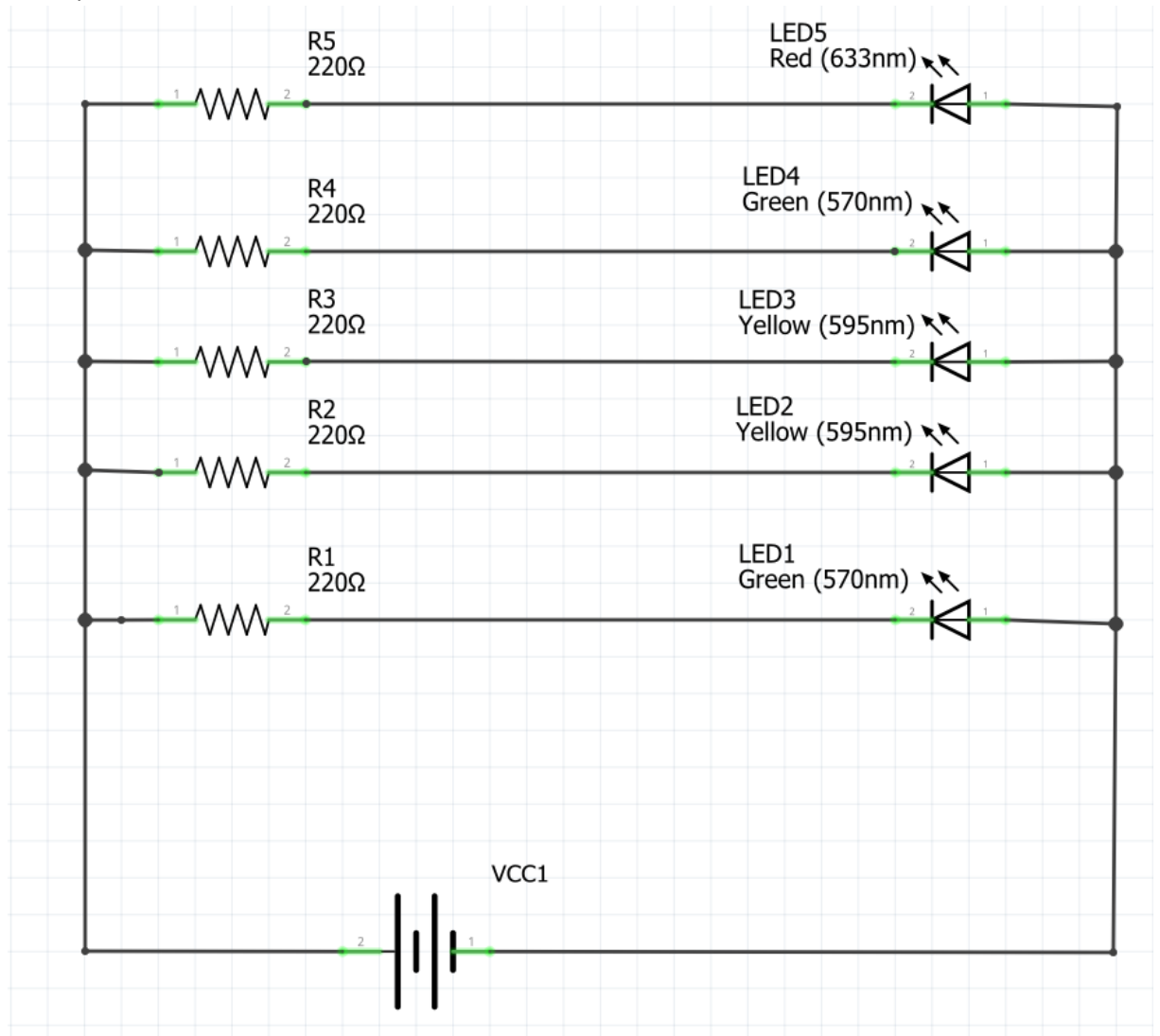
$$3.1. R_V = \frac{U_R}{I} = \frac{1.9V}{0.02A} = 95\Omega$$

$$3.2. U_R = R_V * I = 220\Omega * 0.02A = 4.4V$$

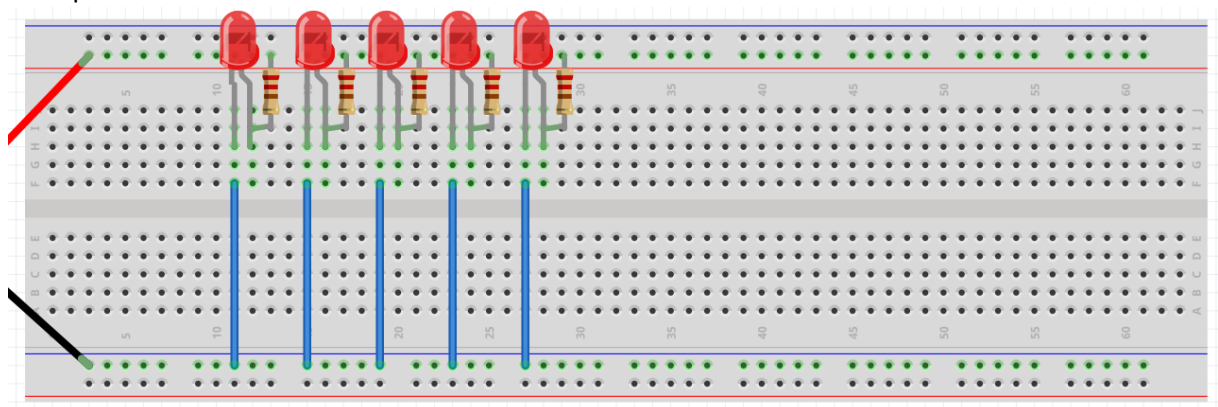
Messung: 1.942



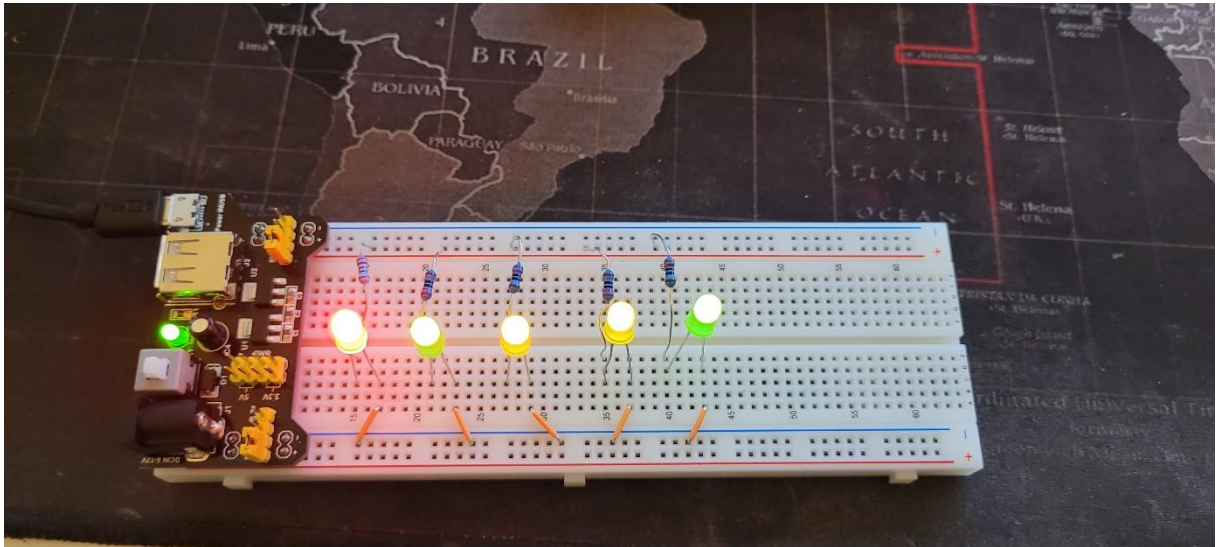
3.3. Schaltplan:



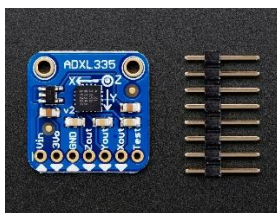
Steckplan:



Umsetzung:



4. "adafruit! 3-axis accelerometer":



Datasheet: <https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>

Ein Accelerometer („Beschleunigungsmeter“) ist Sensor, mit dem die Beschleunigung bzw. die Beschleunigungskräfte gemessen werden können. Das ADXL335 kann statische Beschleunigung der Schwerkraft in Neigungssensiblen Anwendungen sowie dynamische Beschleunigung als Folge von Bewegung, Schock oder Vibration wahrnehmen. Das ADXL335 kann bis zu $\pm 3g$ wahrnehmen.

„WEMOS LOLIN D1 Mini“:



Datasheet: https://www.wemos.cc/en/latest/_static/files/sch_d1_mini_v3.0.0.pdf

Der D1 Mini ist ein Mini-Wifi-Board mit 4MB Flash-Speicher. Es wurde für den Arduino-Mikrocontroller konzipiert.

„Ultrasonic Ranging Modoule“:

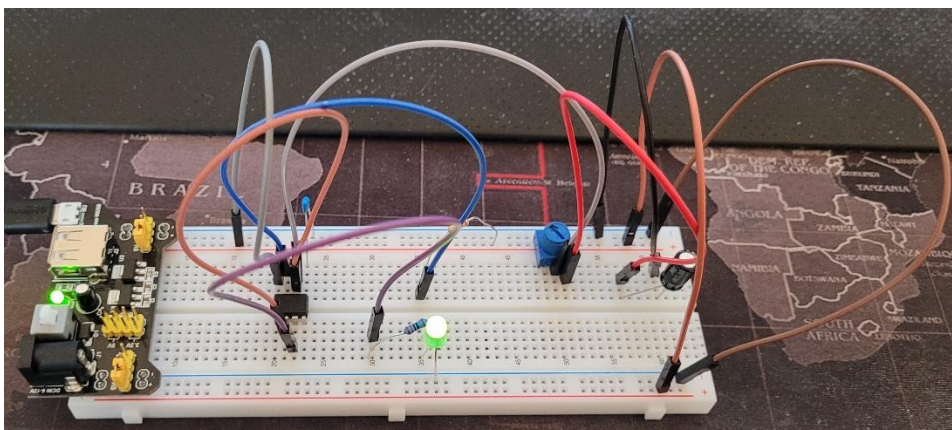


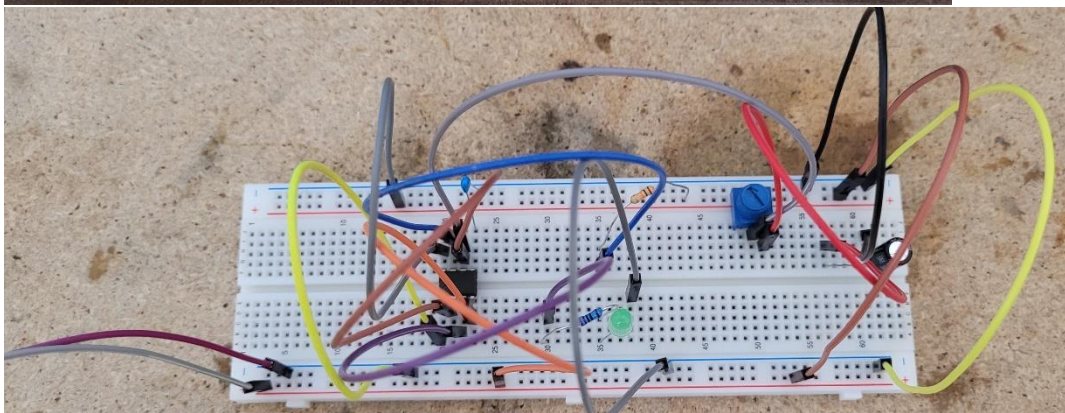
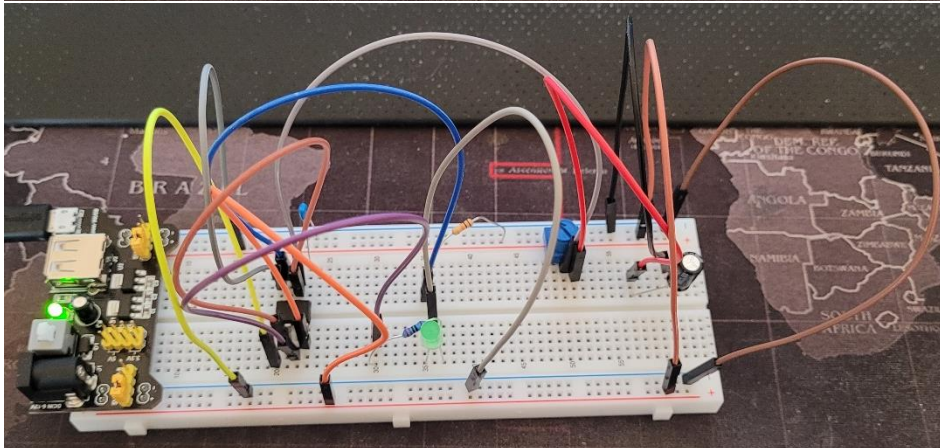
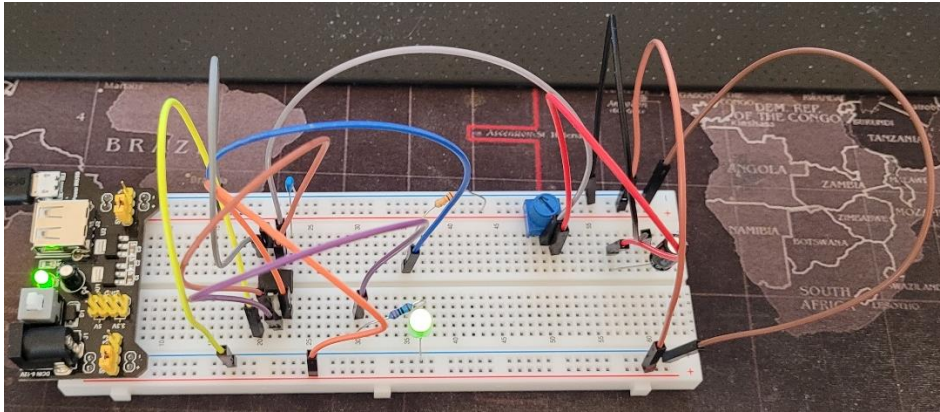
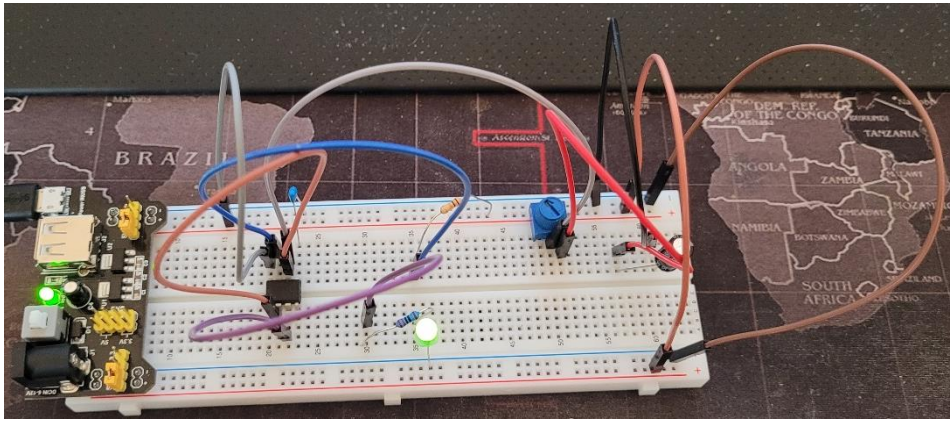
Datasheet: https://www.wemos.cc/en/latest/_static/files/sch_d1_mini_v3.0.0.pdf

“Ultrasonic-Module” sind Ultraschall-Sensoren, die die Entfernung zu anderen Objekten messen können. Dazu werden ultraschall-wellen ausgesendet und wiederaufgenommen. Zur Entfernungsmessung wird dann die gemessene Zeit zwischen aussenden und wieder empfangen gemessen. Das HC-SR04 wurde ebenfalls für den Einsatz mit einem Arduino konzipiert.

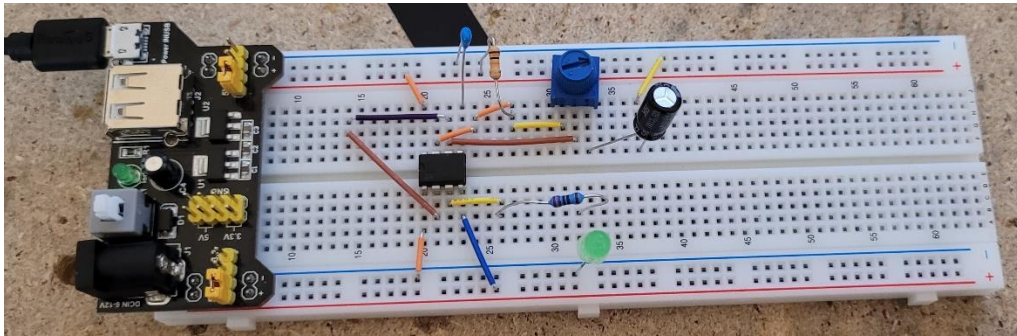
5.

5.1. Ähnlich wie beim Programmieren habe ich mir gedacht: Erstmal frei drauf los bauen und gucken was bei rumkommt. Das hat auch einigermaßen gut funktioniert, wobei ich mit den langen Verbindungen schnell mal den Überblick verloren hab. Ich habe es dann auch relativ schnell geschafft die LED zum Leuchten zu bringen. Allerdings hat sie nur dauerhaft geleuchtet und nicht geblinkt. Nach den ersten Versuchen ist mir relativ schnell aufgefallen, dass ich einige Verbindungen, insbesondere die Grounds, vergessen habe. Wahrscheinlich weil ich wieder mal den Überblick verloren habe! Zwischenzeitlich hat die LED dann nicht mehr geleuchtet. Einen Grund dafür habe ich nicht wirklich erkennen können. Viele Tutorials bzw. Hilfestellungen, habe immer nur einseitige Stromversorgungen und eine Bridge zwischen den rechten und linken Polen. Dementsprechend bin ich dann ebenfalls darauf umgestiegen und habe die LED wieder zum Leuchten gebracht. Ein Blinken war aber immer noch nicht zu erkennen. Stunden später (keine Ahnung wie lange wirklich), ist mir dann aufgefallen, dass Pin6 über den Flip-Flop (quasi den Timer-Teil) entscheidet und habe dann feststellen können, dass über das Poti die Frequenz einstellbar ist. Da mein Poti auf maximalem Widerstand war, war die Frequenz so hoch, dass es so ausgesehen hat, als ob die LED dauerhaft leuchtet, obwohl sie nur so schnell geblinkt hat. So konnte ich das die funktionierende Schaltung bauen.

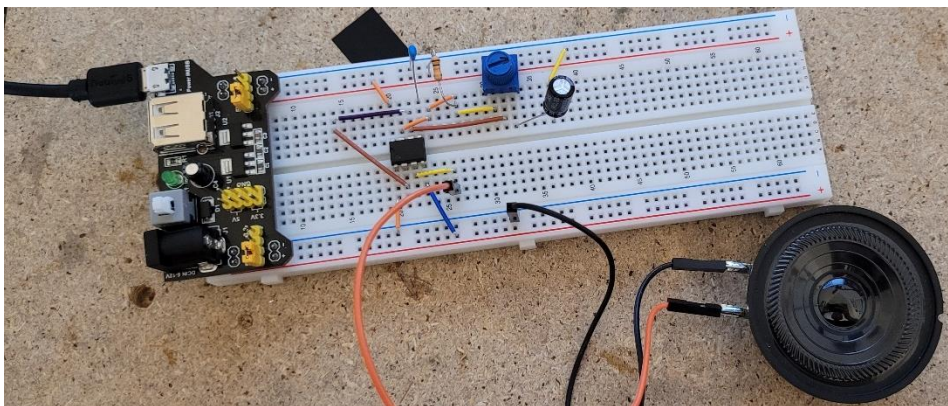




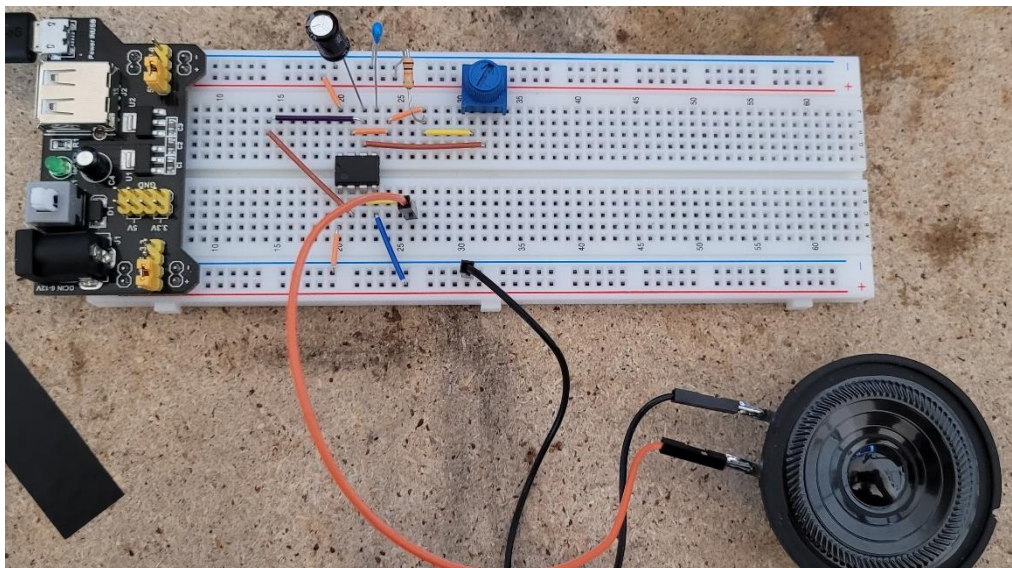
- 5.2. Beim Umstecken habe ich mich dann gefragt, warum ich die Pol-Brücken überhaupt brauche, wenn meine PSU doch ohnehin an beide angeschlossen ist. Dementsprechend habe ich dann umgebaut wie abgebildet umgebaut.



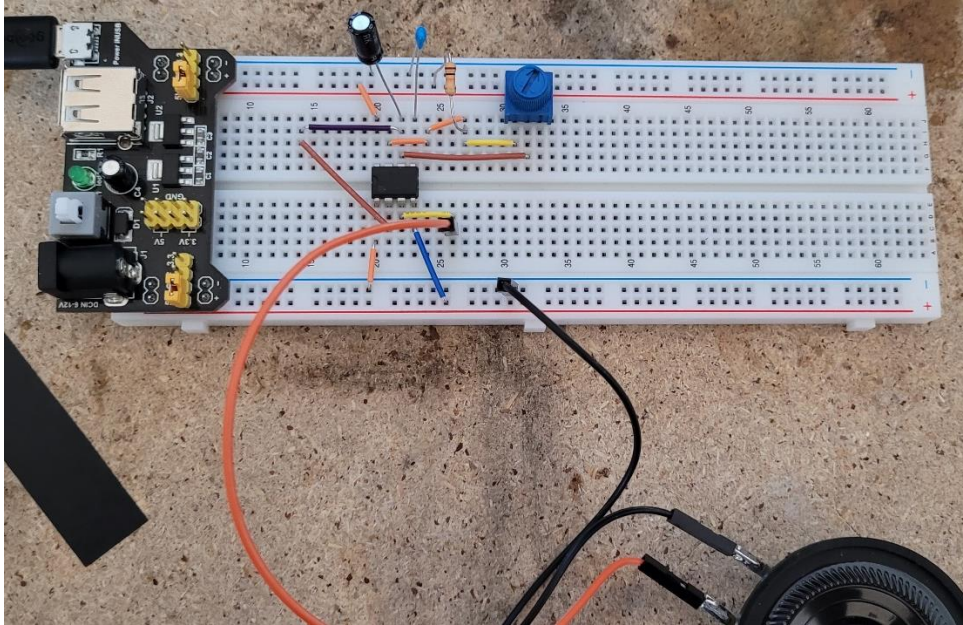
- 5.3. Meine ursprüngliche Schaltung hat funktioniert (mit LED), hat aber eine Anomalie beim Lautsprecher aufgewiesen: Die Zyklen wurden immer schneller und das anfängliche knacken wurde nach einigen Durchläufen zu einem dauerhaften Ton. Meine erste Schaltung sah wie folgt aus:



Nach einiger Ratlosigkeit und anschließender Hilfe per PN, ist aufgefallen, dass der Kondensator (hier an Poti-Pin3 ganz rechts) direkt mit dem Pin6 verbunden werden soll und nicht hinter dem Poti hängt. Dementsprechend habe ich meine Schaltung umgebaut:



Für den dauerhaften Ton müsste man entsprechend wieder die Frequenz anpassen. Pin6 ist für den Status des Flip-Flop, also wie oft gewechselt wird, verantwortlich. Verändert man die Vergleichs-Amplitude an Pin6, z.B. durch einen anderen Kondensator (da dieser als Vergleichsobjekt genommen wird), ändert sich dementsprechend auch der Flip-Flop, also wie oft der Timer geschaltet wird. Da wir nur einen kleineren Kondensator ($0,47\mu\text{F}$ statt $100\mu\text{F}$) haben, habe ich zuerst diese Variante probiert, was auch zum Erfolg (dauerhafter Ton) geführt hat:



- 5.4. Mit zwei Tasten hätte ich selber einen XOR bauen müssen. Praktischerweise haben wir einen Kippschalter, der das für mich über nimmt in der Box. Diesen habe ich dann an die Position der beiden Kondensatoren gebaut, sodass mit Schalter der Ton ausgewählt werden kann. Außerdem habe ich noch die Steckverbindungen ein wenig vereinfacht:

