

1 Aufgabenstellung

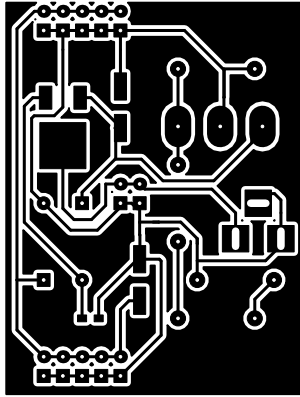
Es soll ein Doppelpendel gebaut werden, welches sich wenig bis vernachlässigbar chaotisch verhält. Dazu muss die Masse des unteren Pendels vielfach größer als die des oberen Pendels sein. Das obere Pendel besitzt keine weitere Masse und ist 50cm bis 1m lang. Es dient grundsätzlich nur der Aufhängung des unteren Pendels, welches aus einem 50cm langem Stab bestehen soll. Über einen Magneten am Ende des Pendels und einer Spule an bzw. nahe der Ruheposition des Pendels soll mittels eines Mikrocontrollers (ATmega16) sowohl die Schwingungsdauer gemessen als auch auf die Position des Pendels in Echtzeit geschlossen werden. Je nach Auslenkung sollen auf einem 50cm langem LED-Streifen verschiedene Farben des Regenbogens angezeigt werden. Beim Erreichen einer Amplitude soll ein Regenbogen entlang des LED-Streifens angezeigt werden, der sich bis zum Erreichen der Ruheposition zu einer einheitlichen Mittelwertfarbe der Extrema wandelt und bis zum Erreichen der anderen Amplitude wieder zu einem Regenbogen, jedoch in umgekehrter Reihenfolge, wandelt. Bei den LEDs soll es sich um WS2812B oder protokoll-gleichen Alternativen handeln. Das Protokoll der *Daisy-Chain* soll implementiert werden. Ebenfalls soll der Mikrocontroller mittels der zur Messung verwendeten Spule das Pendel antreiben, sodass es bei gegebener Energiezufuhr ewig schwingen könnte. Zur Spannungsversorgung wird ein Netzteil verwendet, welches 12 Volt Gleichstrom mit 6 Watt und hoher Restwelligkeit liefert, die auszugleichen ist und auf konstante 5 Volt Gleichstrom geregelt werden soll.

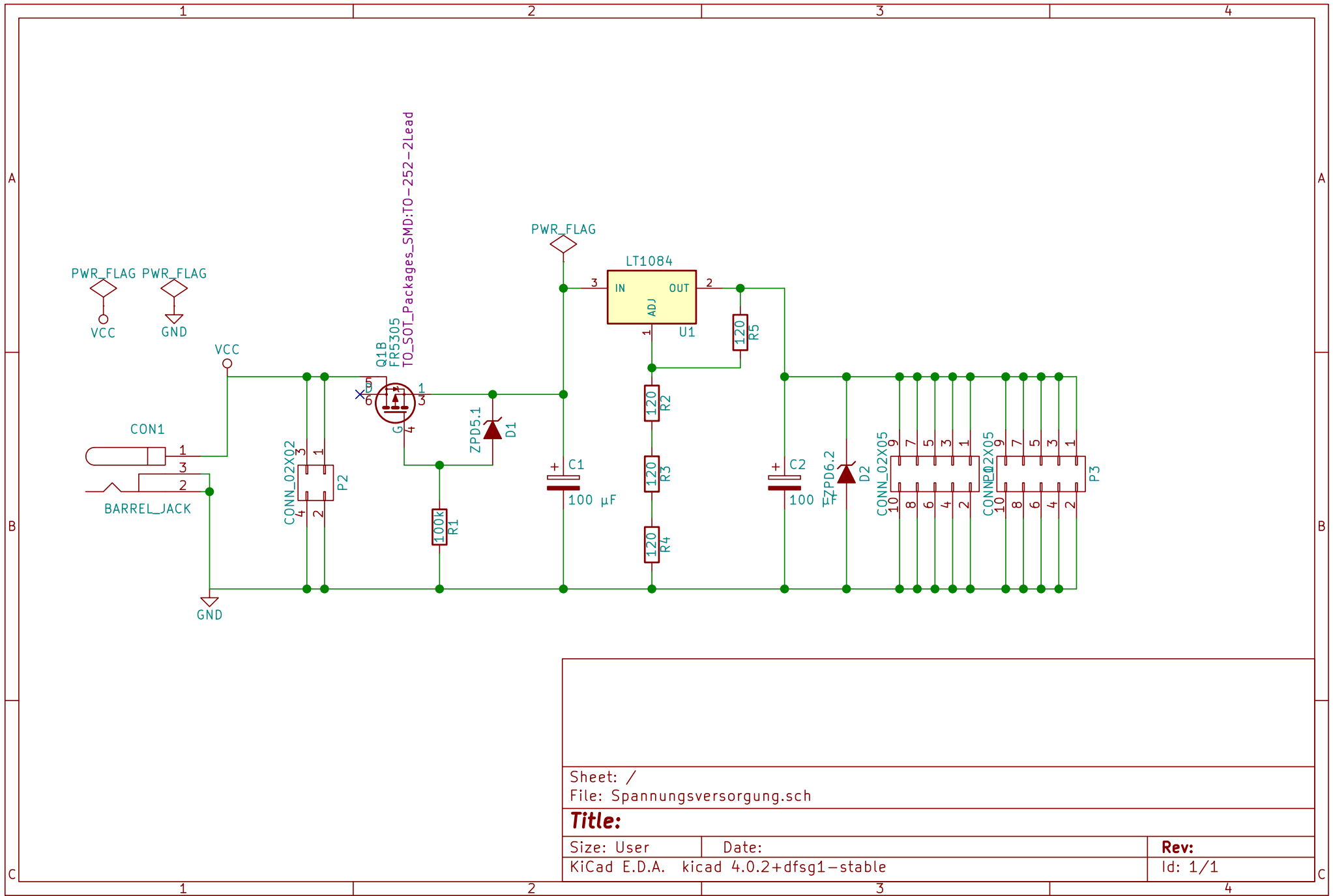
2 Spannungsversorgung

Zu aller erst wurde die Versorgungsspannung in Angriff genommen. Ein Netzteil mit den durch die Aufgabenstellung gegebenen Anforderungen soll über eine Buchse (*Barrel-Jack*) angeschlossen werden. Dessen Signal wird über einen Linearregler (LT1084) mit einer Widerstandsrückkopplung von 1 : 3 auf 5 Volt reguliert. Kondensatoren vor und nach dem Linearregler dienen dem Ausgleich leichter Schwankungen. Ein Verpolungsschutz wurde zusätzlich vor und eine Zenerdiode, die bei 6.2 Volt in Sperrichtung leitet, nach dem Linearregler eingebaut.

Daraus entstand eine eigene Platine, die eigens geplant, entworfen, bedruckt, geätzt, gebohrt und gelötet wurde. Dabei wurde auf Wiederverwendbarkeit geachtet und Pins angebracht, um die Platine auf einem handelsüblichen Steckbrett als 5-Volt-Spannungsversorgung nutzen zu können. Bei einigen Test unter Vollast wurde der Linearregler sehr heiß, Abhilfe schaffte die Montage eines Kühlkörpers und eines kleinen Lüfters (12 Volt, 0.5 Watt).

Nun folgen der Footprint und der Schaltplan der Platine:





Sheet: /
 File: Spannungsversorgung.sch

Title:

Size: User Date:
 KiCad E.D.A. kicad 4.0.2+dfsg1-stable

Rev:
 Id: 1/1

3 Mikrocontroller und Programmer

Als Mikrocontroller wurde ein ATmega328P verwendet, da nur ein solcher zur Verfügung stand. Beim Programmieren wurde stets darauf geachtet, dass jegliche Programme ebenso auf einem ATmega16 ausgeführt werden könnten. Als Programmer wurde ein Arduino mit modifizierten *ArduinoISP*-Sketch als *In System Programmer* eingesetzt. Somit wurde dann über die Verbindung mit SCK, MISO, MOSI und RESET der eigentliche Mikrocontroller geflasht. Der Einfachheit halber wurde sowohl für den Programmer als auch für den Mikrocontroller ein Breakoutboard auf einer Lochrasterplatine hergestellt. Die physische Verbindung wurde mit Jumperkabeln geschaffen. Zum Kompilieren und Flashen wurde ein Makefile verwendet.

4 LEDs: WS2812B

Das Daisy-Chain-Protokoll wurde gemäß Datenblatt implementiert (siehe *pendulum*).

4.1 Timing

Es wurde ein Oszillator mit 20 MHz als externer Taktgeber für den Mikrocontroller verwendet. Dies führte zu 50 ns Ausführungszeit pro Befehl. Um das WS2812B-Protokoll zu implementieren, mussten mit Assembler Routinen implementiert werden, die genau 6, 8, 14 oder 15 Takte andauerten. Die genauen Zeiten werden als Kommentar in *pendulum.c* aufgeschlüsselt. Ein Problem hierbei war die Optimierung des Compilers, da dieser selbst auf niedrigster Optimierungsstufe mehrere NOP (*no operation*) wegoptimierte. Abhilfe schaffte zuletzt die Verwendung eines bzw. mehrerer *relativer Sprung + 0*.

4.2 Colors

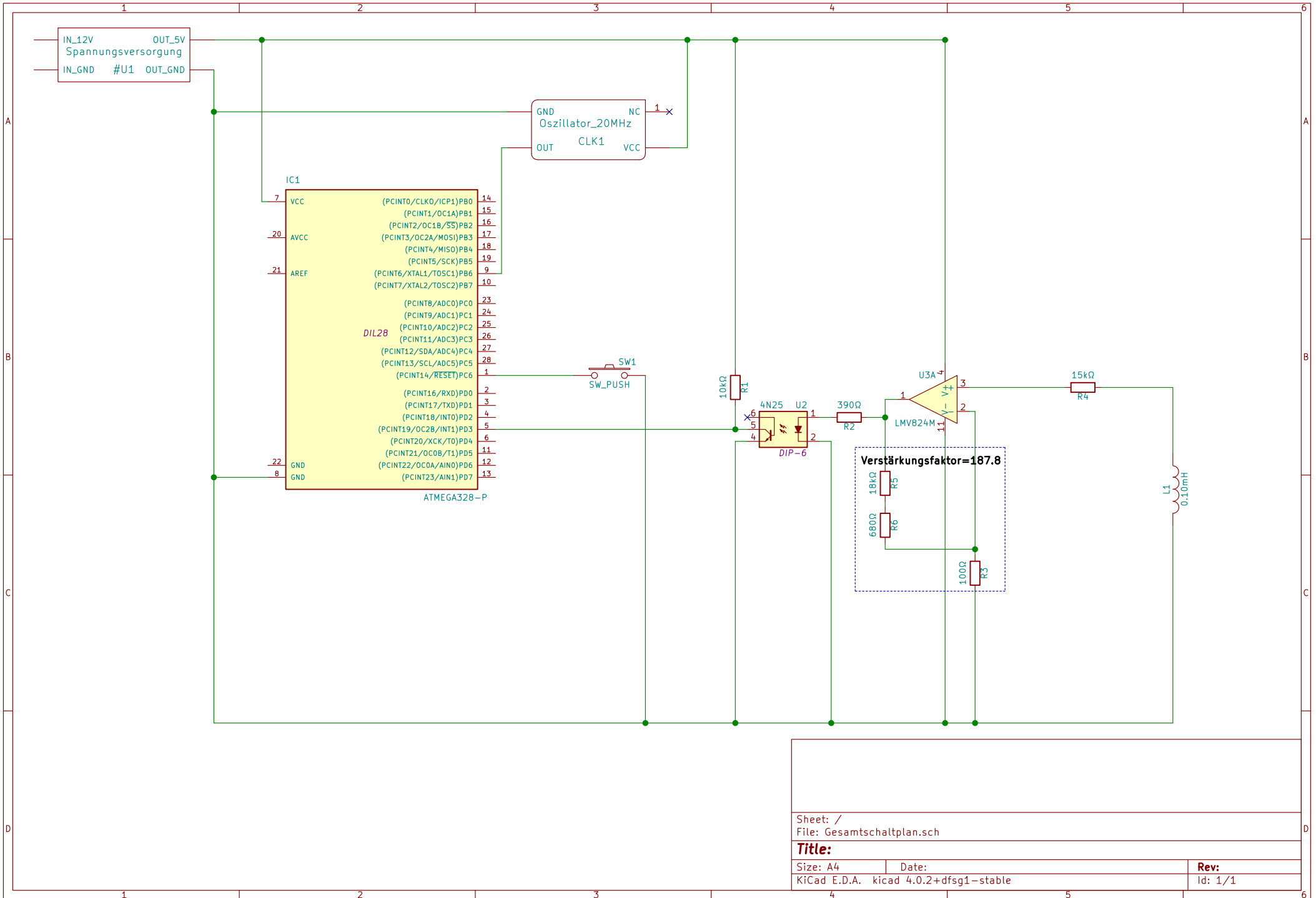
Um die Farben gemäß der Aufgabenstellung regenbogenähnlich darzustellen, wurde die HSV-Farbskala verwendet (siehe *color_hsv.c*). Hier wurde die Minimum- und die Maximum-Farbe gezielt in festen Schritten in Grad zu- bzw. voneinander verschoben. Anschließend wurde eine lineare Interpolation zwischen diesen beiden Werten durchgeführt, um die einzelnen Farben der LEDs zu ermitteln. Zuletzt wurden die HSV-Farben und RGB-Farben umgerechnet. Eine Demonstration einer 1-Grad-Verschiebung wurde in *led-driver* implementiert.

5 Pendelaufbau

Als Pendel wurde zu Beginn ein einfaches Fadenpendel mit einem Gewicht und einem Magneten aufgebaut und für jeglich Test verwendet. Schnell wurde klar, dass die Anforderung des Anstoßens aufgrund der hohen Masse vernachlässigt werden muss. Die spätere Umsetzung mittels eines Holzstabs als unteres Pendel führt leider zu keiner wesentlichen Verbesserung des Problems. Das Pendel muss somit per Hand angestoßen werden und schwingt nicht länger als eine Minute. Das Gegenmagnetfeld der Messspule bremst das Pendel zusätzlich bei jedem Passieren der Ruheposition. Im finalen Aufbau wurde auf das elektromagnetische Anstoßen in Gänze verzichtet.

Verbesserungsvorschläge Das Pendel sollte leichter gebaut werden. Eine *kräftigere* Spule müsste eingesetzt werden.

6 Gesamtschaltplan



Sheet: /
 File: Gesamtschaltplan.sch

Title:

Size: A4 Date:
 KiCad E.D.A. kicad 4.0.2+dfsg1-stable

Rev:
 Id: 1/1

6.1 Signale

Beim schwingenden Pendel mit Magnet wird eine Spannung in der Spule induziert, die an bzw. nahe der Ruheposition platziert wurde. Diese Spannung beträgt 5 bis 20 mV und muss für die Nutzung in Digitalschaltungen verstärkt werden. Dazu wurde ein Operationsverstärker (LMV824M) verwendet und per Widerstandsverhältnis auf einen Verstärkungsfaktor von rund 190 eingestellt. Dies reichte aus um das verstärkte Rauschen unter 1 Volt zu halten und bei Erregungen durch das vorbeischwingende Pendel auf über 1 Volt zu kommen. Dies wurde dann in einen Optokoppler (4N35) an dessen Anode eingespeist. Hinter einem Pullup-Widerstand wurde der Kollektor des Optokopplers an den externen Interrupt-Pins des Mikrocontrollers angeschlossen, welcher auf fallende Flanken reagieren soll. Dies hat zur Folge, dass Pendel diese fallende Flanke genau in der Ruheposition auslöst.

Diese Kombination aus Messspule, Operationsverstärker und Optokoppler resultierte in folgenden Signalen:

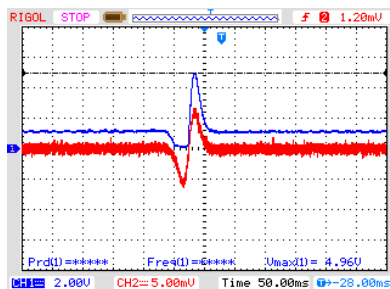


Abbildung 1: CH1: Verstärkte Spannung am Ausgang des Operationsverstärkers; CH2: Induzierte Spannung der Spule am Eingang des Operationsverstärkers

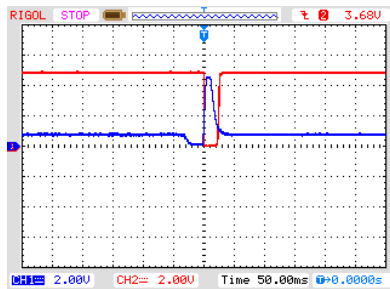


Abbildung 2: CH1: Verstärkte Spannung am Ausgang des Operationsverstärkers bei noch *großer Auslenkung* des Pendels; CH2: Ausgang des Optokopplers/Signal für externen Interrupt

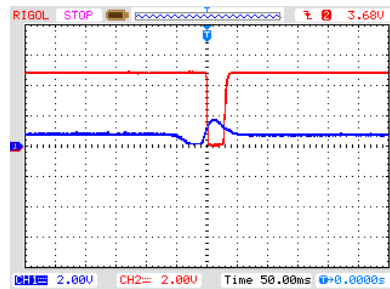
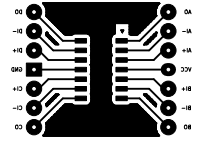


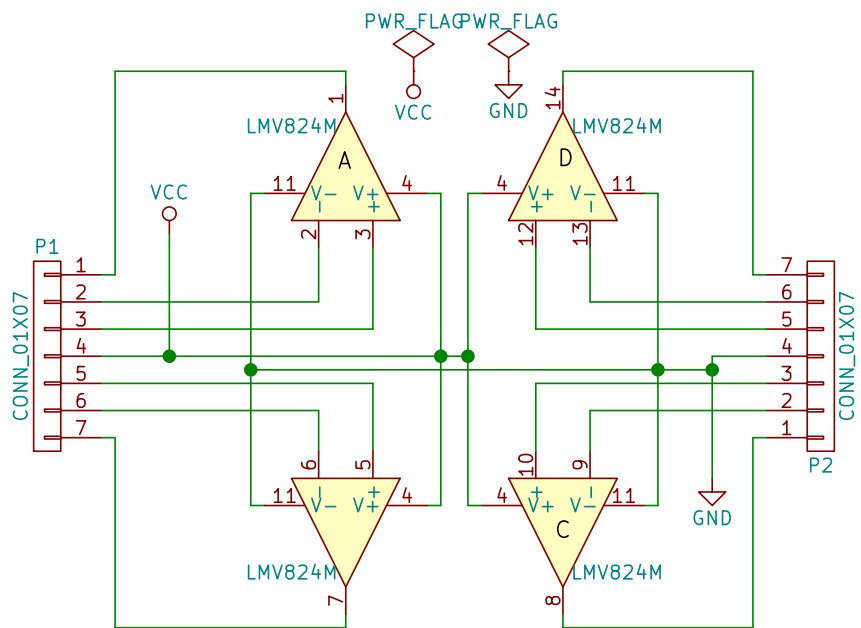
Abbildung 3: CH1: Verstärkte Spannung am Ausgang des Operationsverstärkers bei nur noch *kleiner Auslenkung* des Pendels; CH2: Ausgang des Optokopplers/Signal für externen Interrupt

6.2 Timer Interrupts

Interne Timer 0 und 1 wurden in timer.c so konfiguriert, dass die in 32-bit bis über 200s und in 64-bit rund 30000 Jahre genau Zählen können. Somit kann die Periodendauer auf 50ns genau berechnet werden. Außerdem wurde eine Echtzeitberechnung der Pendelposition und der zugehörigen Farbwerte ermöglicht.

Da der Operationsverstärker nur als SMD-Bauteil zu Verfügung stand, wurde eine weitere Breakout-Platine erstellt. Nun folgend der Footprint und der Schaltplan der Platine des Operationsverstärkers:





Sheet: /
File: OPV.sch

Title:

Size: User

Date:

Rev:

KiCad E.D.A. kicad 4.0.2+dfsg1-stable

Id: 1/1

7 Fazit

Es wurde eine für das gegebene Netzteil und zu der Gesamtschaltung passende Spannungsversorgung geschaffen. Das Daisy-Chain-Protokoll der WS2812B-LEDs wurde erfolgreich implementiert, sowie eine HSV-zu-RGB-Umrechnung samt einer linearen Interpolation der Regenbogenfarben. Induzierte Spannungen an der Messspule wurden geglättet, verstärkt und schließlich genutzt, um einen externen Interrupt am Mikrocontroller auszulösen. Der Mikrocontroller selbst wurde über einen Arduino als In-System-Programmer mit Assembler-/C-Programmen geflasht. Es wurde ein 20 MHz Oszillator als externer Taktgeber verwendet und u.a. mittels Timer-Interrupts eine Echtzeitberechnung der Farben und deren Ausgabe realisiert. Zusätzlich sind einige Platinen/Breakoutboards entworfen und gefertigt wurden.

Auf das Aufbringen des LED-Streifens auf den Stab des 2. Pendels wurde verzichtet, ebenso wie auf das elektromagnetische Anstoßen aufgrund des Masseproblems.

Die Anforderungen der Aufgabenstellung wurden somit mehrheitlich umgesetzt.

Hinweis 1 Datenblätter aller verwendeten Bauteile, sowie Quellcodedateien, Schaltpläne, Messungen, etc. sind ebenfalls in einem Repository¹ zu finden.

Hinweis 2 Beim Planen und Umsetzen des Projekts wurde stets versucht, vorhandene Bauteile zu verwenden oder zu recyceln. Dies reduzierte zwar die Materialkosten, minderte jedoch auch drastisch die Auswahl der zu verwendenden Bauteile. Ein weiteres Ziel war es die Umsetzung sehr modular zu gestalten, sodass eine Wiederverwendbarkeit der Bauteile und Platinen ermöglicht wird. So können diese auch in zukünftigen Projekten zum Einsatz kommen.

¹<https://git.tiband.de/ht/mca-pendel>