

Auftraggeber: gibb

Betreuer: Burri Matthias

Autor: David Jäggli, Nils Jäggi

Datum: 11.11.2020

Der Auftrag dieses Projektes ist es einen Fidget-Spinner zu realisieren, der mit dem PoV-Prinzip (Persistance of Vision) einen Text/Symbole anzeigen kann.

BüP

Hot-Board

Probelauf IPA

Inhalt

[1 Einleitung 2](#_Toc528075819)

[1.1 Zusammenfassung 2](#_Toc528075820)

[2 Hauptteil 3](#_Toc528075821)

[2.1 Einleitung 3](#_Toc528075822)

[2.2 Realisierung der Hardware 4](#_Toc528075823)

[2.2.1 PIC-Mikrocontroller 4](#_Toc528075824)

[2.2.2 Hall-Sensor 4](#_Toc528075825)

[2.2.3 Speisung 5](#_Toc528075826)

[2.2.4 LED Ausgabe 5](#_Toc528075827)

[2.2.5 ICSP- Schnittstelle und Taster 6](#_Toc528075828)

[2.2.6 Mechanische Architektur 7](#_Toc528075829)

[2.3 Software 8](#_Toc528075830)

[2.3.1 Hauptsoftware 8](#_Toc528075831)

[3 Schlussteil 9](#_Toc528075832)

[3.1 Fazit 9](#_Toc528075833)

[3.2 Probleme 9](#_Toc528075834)

[3.3 Vergleich mit Pflichtenheft 9](#_Toc528075835)

[3.4 Quellenverzeichnis 10](#_Toc528075836)

Anhang

[A Anhang 11](#_Toc528075874)

[A.1 Gesamtschema 11](#_Toc528075875)

[A.2 Blockschaltbild 12](#_Toc528075876)

[A.3 Pflichtenheft 13](#_Toc528075877)

[A.4 Bestückungsplan und Stückliste 14](#_Toc528075878)

[A.5 Bedienungsanleitung 16](#_Toc528075879)

[A.6 Zustandsdiagramm Hauptsoftware 17](#_Toc528075881)

[A.7 Zeitplan 18](#_Toc528075882)

# Einleitung

## Zusammenfassung

Die Lernenden wurden in 2er Gruppen eingeteilt. Jede Gruppe muss eigenständig ihr zugeteiltes Projekt realisieren. Das Projekt soll in einer Zeitspanne von 39 Tagen ausgearbeitet werden.

Der Auftrag dieses Projektes ist es einen Fidget-Spinner zu realisieren, der mit dem PoV-Prinzip (Persistence of Vision) einen Text/Symbole anzeigen kann. Mit der Trägheit des Auges «sieht» der Mensch einen Text. Die einzige hardwaremässige Einschränkung war es einen PIC-Mikrocontroller zu verwenden. Alle Lernenden müssen eine vollständige Dokumentation zu ihrem Projekt schreiben und ihr Produkt präsentieren.

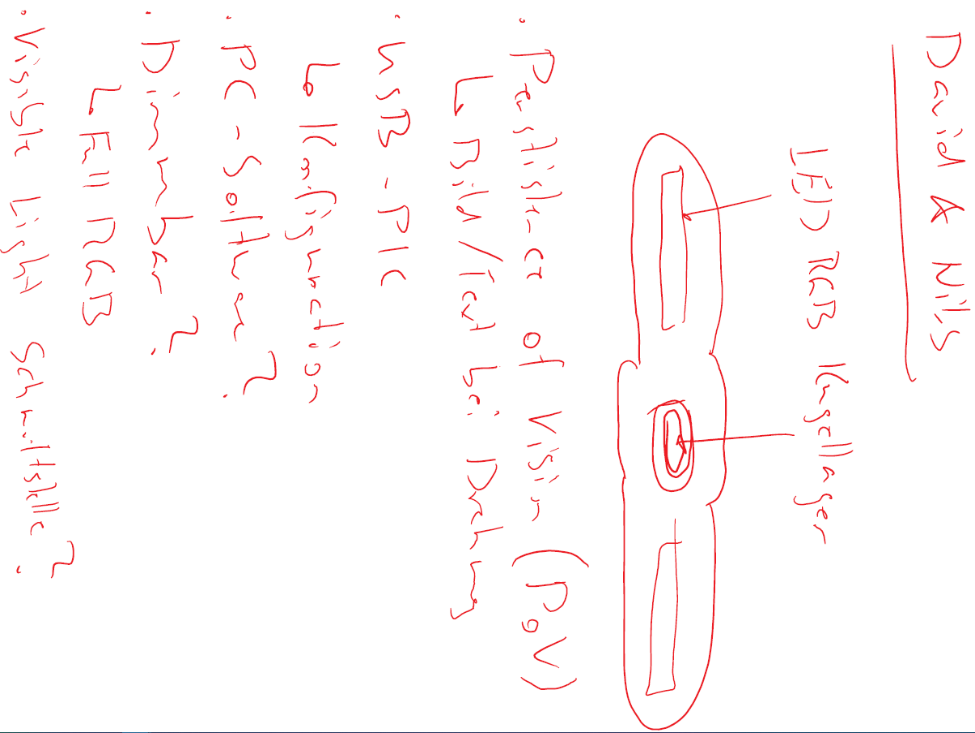
Das Resultat ist ein Leiterplatte mit einem Kugellager in der Mitte, die wie ein 2-teiliger Fidget Spinner zugeschnitten ist. Auf der einen Seite sind 7 LEDs in einer Reihe angeordnet um den Text anzuzeigen sobald der Fidget Spinner dreht. Die Umdrehungen werden mit einem Hall-Sensor gemessen, der durch 2 Magnete im Fidget Spinner Cap ausgelöst wird. Der anzuzeigende Text wird beim Programmieren festgelegt und kann nur durch ein erneutes Herunterladen des kompletten Programmes geändert werden.

Nachdem Grundinformationen gesammelt wurden konnte damit begonnen werden die einzelnen Schaltungen zu entwerfen. Mit dem fertigen Schema kann das Layout gezeichnet und anschliessend die Leiterplatte gefertigt werden. Sobald die Hardware vollständig funktioniert, wird die Hauptsoftware geschrieben. Gegen Ende der Projektzeit stellen alle Gruppen Ihr Projekt den anderen Lernenden vor.

Das Projekt konnte fast komplett umgesetzt werden, ausser dass die Serielle Schnittstelle nicht funktioniert. Um die Farbe und den Text zu ändern wird ein Taster verwendet. Wenn noch Zeit zur Verfügung stände, könnte die USB- Schnittstelle noch realisiert werden.

# Hauptteil

## Einleitung

Im 2. Lehrjahr realisieren die Elektroniker des Basislehrjahrs selbstständig ein Projekt. Das Projekt wird in 2er Gruppen realisiert und ist von den Ausbildnern vorgesetzt.

Der Auftrag ist es das Gerät zu realisieren und am Ende zuerst zu Präsentieren und dann die Dokumentation inklusive Layout, Software etc. abzugeben.

Ziel dieses Projekts ist es einen Fidget Spinner zu realisieren, der mittels PoV(Persistance of View) einen Text anzeigen kann. Zusätzlich soll dies über eine Serielle Schnittstelle verändert werden können. Weitere Informationen im Pflichtenheft.

Zu Beginn wurde einmal zusammengetragen, was alles benötigt wird und wie es umgesetzt werden soll. Danach wurde geschaut welche Bauteile verwendet werden sollen, um die einzelnen Schaltungen zu realisieren. Die Schaltungen mussten nur noch ins Altium übertragen und das Layout gezeichnet. Sobald die Hardware funktionstüchtig war wurde die Software geschrieben.

## Realisierung der Hardware

Die Hardware beinhaltet die 12 Taster und einen Mikrocontroller, welcher die Tastenanschläge in einen von der Software interpretierbaren Command umwandelt. Dafür wurde ein STM32L412KB verwendet. Dieser bringt bereits ein Protokoll zur Kommunikation über Virtual-COM-Port mit sich.

### PIC-Mikrocontroller

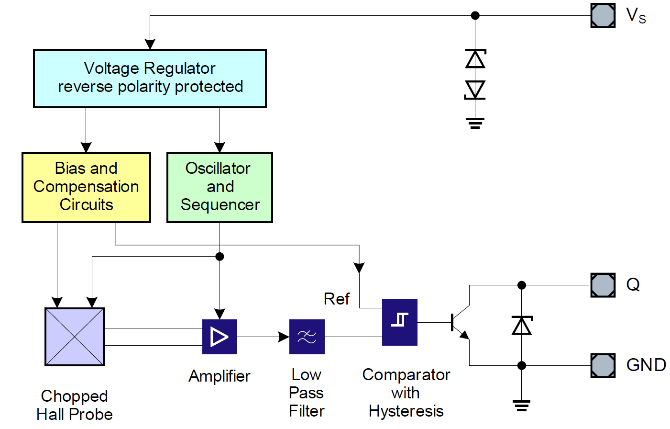


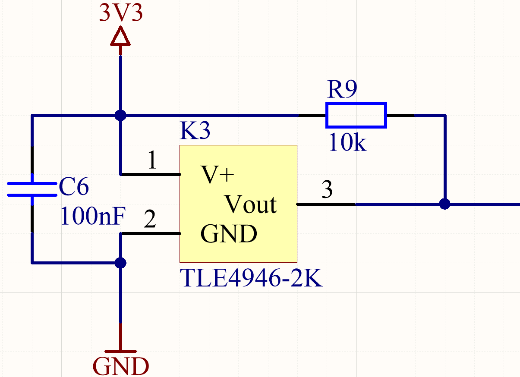
Die zentrale Komponente des Fidget-Spinners ist der Mikrocontroller. Er wertet die Signale des Hall-Sensors aus und steuert die LEDs an. Die Vorgabe ist es einen PIC von Microchip zu verwenden. Da unser Gerät über eine serielle Schnittstelle kommunizieren können soll, wird ein Controller mit USB benötigt. Diese haben zwei Pins an denen direkt D+ und D- des USB Ports angeschlossen werden kann. Der einzige der über diese Funktion verfügt und genug Pins zur Verfügung stellt ist der PIC18F45K50.

**Spezifikationen:**

* Gehäuse: TQFP44
* Programmspeicher: 32 KB
* SRAM: 2 KB
* EEPROM: 256 B
* Schnittstellen: UART, SPI, I2C, USB

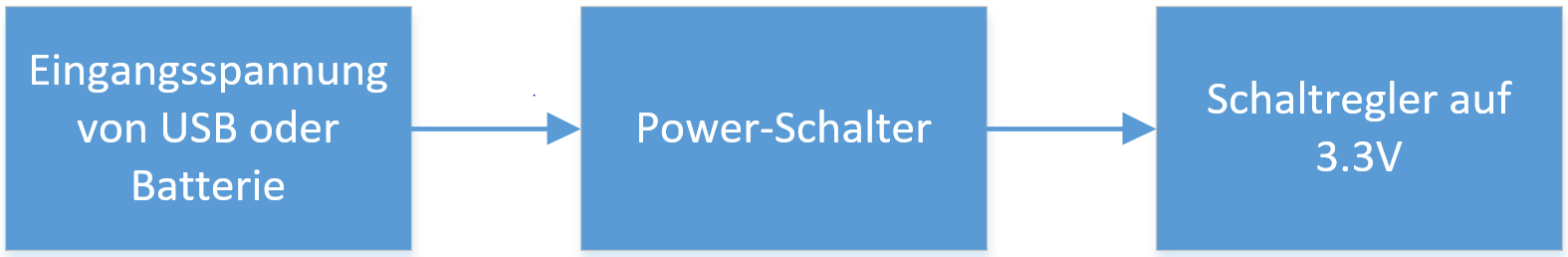
### Hall-Sensor

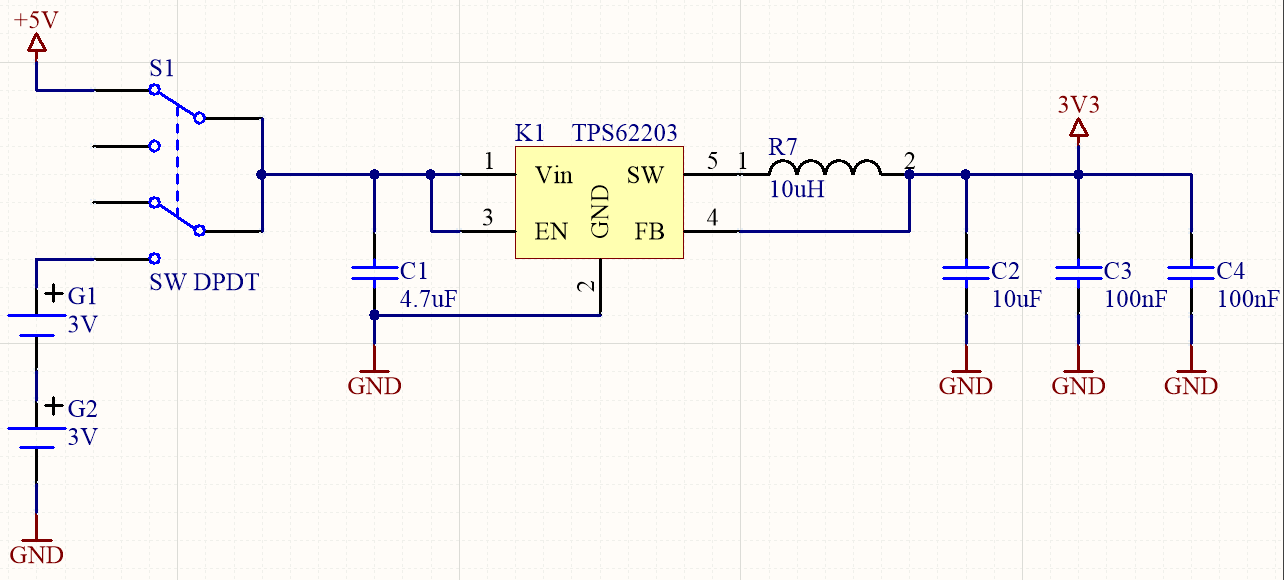
Um die LEDs im richtigen Moment anzusteuern werden die Umdrehungen des Fidget-Spinners fortlaufend mit einem Hall-Sensor gemessen. Der in diesem Projekt verwendete Hall-Sensor ist ein einfaches Latch. Bei einem vorhandenen Magnetfeld ist die Q-GND-Strecke miteinander verbunden. Die Schaltung benötigt nur noch einen externen Pull-Up-Widerstand um das Signal zu definieren, falls kein erkennbares Magnetfeld vorhanden ist. Ebenfalls braucht es noch einen 100nF Abblockkondensator.

Unter der Abdeckung des Kugellagers sind 2 Magnete verbaut die genug stark sind um den Sensor auszulösen, wenn er vorbeikommt. Wir verwenden 2 Stück aus dem Grund, da es so genauer wird, weil der Fidget-Spinner kontinuierlich langsamer wird. Die Spezialabdeckungen wurden in 3D gezeichnet und danach bei Parts-Printing in Grün ausgedruckt lassen.

Mit den verwendeten Magneten sind beim Sensor ungefähr 175 Gauss vorhanden. Der absolute Maximalwert um zu schalten liegt bei 2 mT = 20 Gauss. Das bedeutet, dass das Magnetfeld der Magnete genug stark ist um vom Hall-Sensor erkannt zu werden.

### Speisung



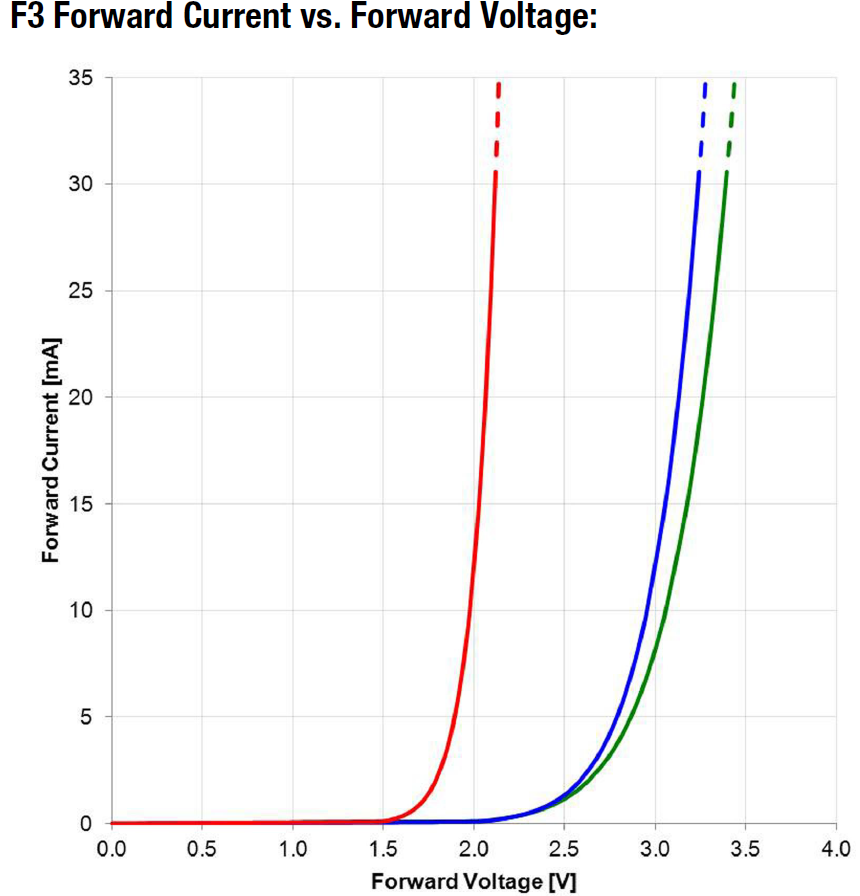
Damit der Spinner gut drehen kann, braucht es eine optimale Gewichtsverteilung. Deshalb wird das Gerät mit 2 Knopfbatterien betrieben, die dann hauptsächlich für die Gewichtsverteilung zuständig sind. Mit einem Spannungsregler werden dann die 6V der beiden Batterien in Serie auf 3.3V geregelt. Anstelle von einem linearen Spannungsregler, wird ein Buck-Converter verwendet. Der grosse Vorteil von einem Buck-Converter ist, dass er die Differenzspannung nicht in Wärme umwandelt. Beim Fidget-Spinner wären das 45% der Energie. Stattdessen wird die Spannung nur teilweise eingeschaltet ähnlich wie bei einem PWM Signal. Auf diese Weise werden nur 5% der Energie in Wärme umgewandelt.

Für die Speisung wird der TPS62203 verwendet. Dieser Regelt fix von max. 6V auf 3.3V und ist in einem SOT23-5 Gehäuse. Um richtig zu Funktionieren benötigt er 2 Kondensatoren und eine Spule. Die Grössen dieser Bauteile sind fix vom Datenblatt gegeben. Bei der Kommunikation über die UART-Schnittstelle wird das Gerät über USB gespiesen. Es kann dann über einen on-on Schalter zwischen Batterie- und USB Betrieb gewechselt werden. Sobald die USB Schnittstelle gekappt ist funktioniert er auch als Power-Schalter.

### LED Ausgabe

Um den Text ausgeben zu können werden 7 RGB LEDs benützt. Es wurde sich gegen 8 LEDs entschieden, denn bei 7 LEDs ist es einfacher später die Buchstaben auszugeben, da es eine Mitte hat und so die Zeichen symmetrisch dargestellt werden können. Die LED werden alle einzeln an einen Pin des Mikrocontrollers angeschlossen. So können die LEDs in 7 verschiedenen Farben leuchten.

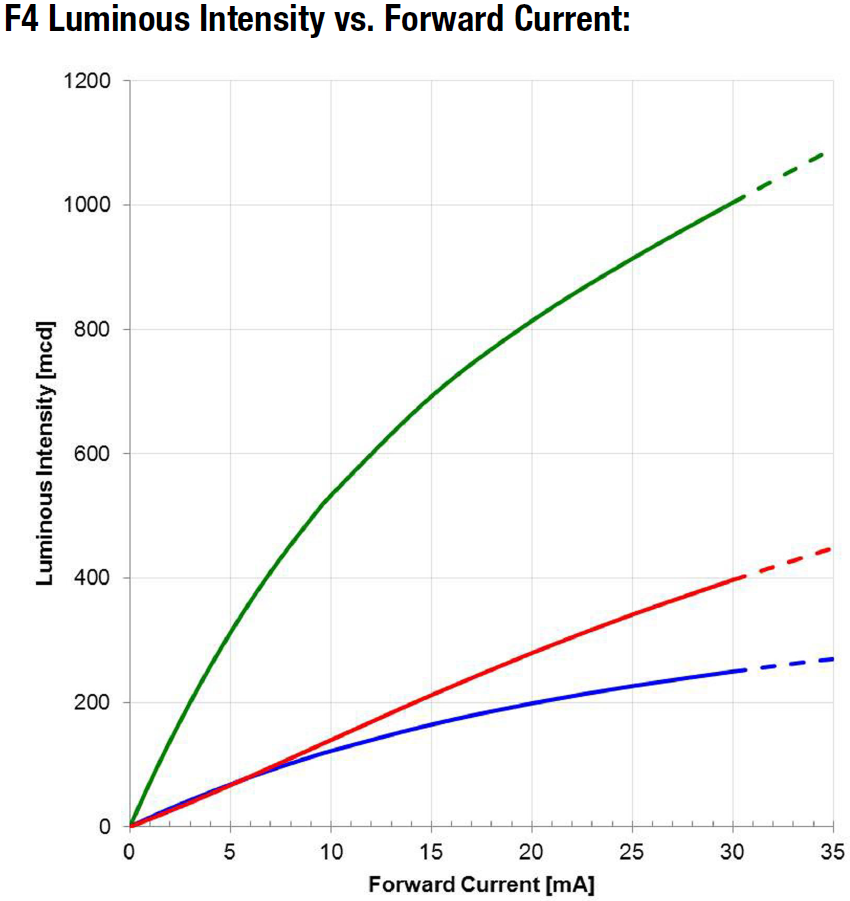
**Vorwiderstände:**



Die LEDs sollen alle ca. mit 100mcd (milli-Lumen) leuchten.

**Grün:**

**Blau:**

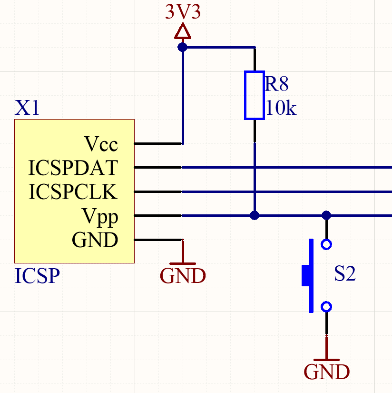


**Rot:**

Die LED Helligkeit wurde auf dem Breadboard getestet und ergab, dass der Widerstand der Blauen und Grünen LED zu klein ist

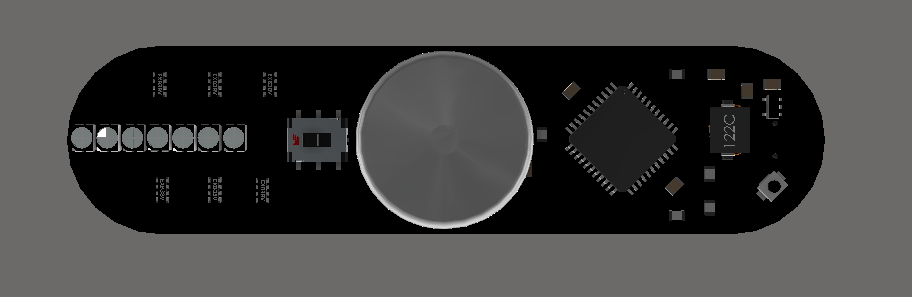
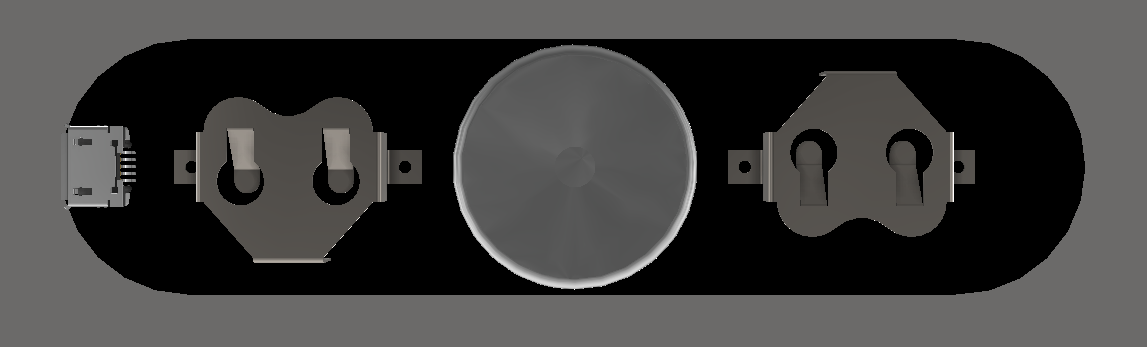
Ob alle LEDs gleich hell sind konnte dadurch erkannt werden, dass ein schönes Weiss entsteht sobald alle LEDs leuchten

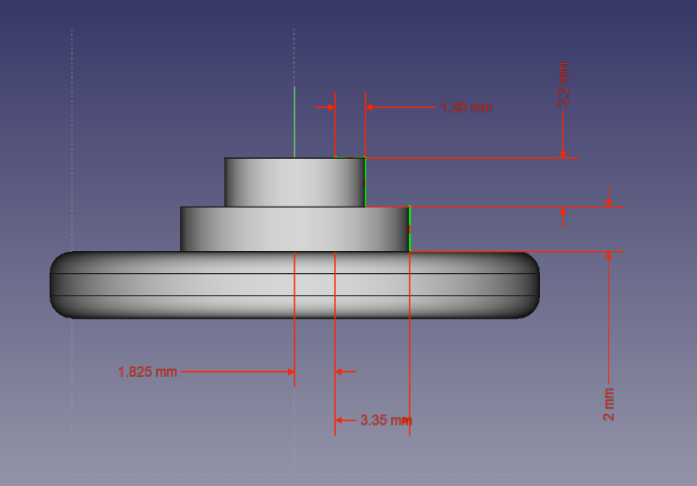
### ICSP- Schnittstelle und Taster

Um den PIC zu programmieren wurden unter der Kugellager-Abdeckung die ICSP-Schnittstelle platziert. Dort ist sie wenig sichtbar. Um zu Programmieren wird eine Stiftleiste angelötet die danach wieder entfernt wird. Um den PIC zu Programmieren muss ebenfalls ein Draht von PIN 33 () zu MCLR der ICSP-Schnittstelle gezogen werden. Der Grund dafür ist, dass die beiden Programmierleitungen der ICSP-Schnittstelle auf die PINs des Special-Programming-Modes verbunden sind. Und für dieses Special-Programming wird der PIN als Reset benötigt.

Es wurde ebenfalls ein kleiner Taster verbaut. Dieser ist als Reset gedacht kann aber nach Belieben auch als Eingabe-Taster verwendet werden.

### Mechanische Architektur



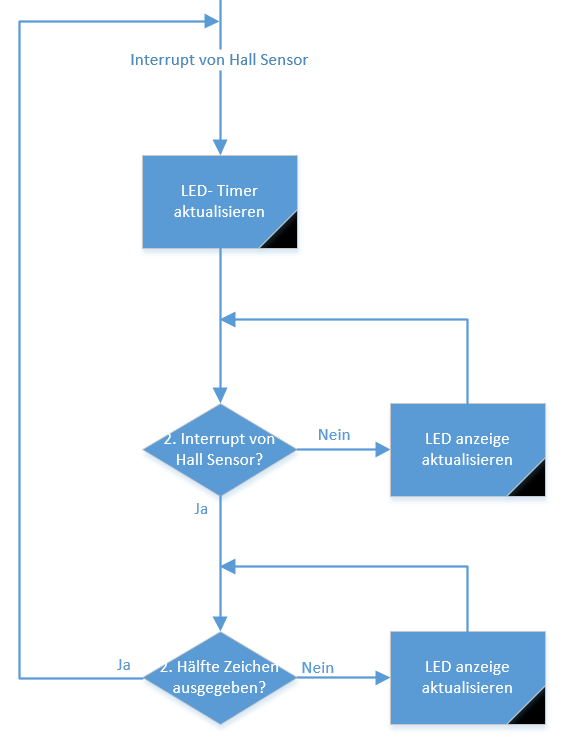
Es wurde sich dazu entschieden kein Gehäuse zu verwenden, da zu wenig Platz vorhanden ist und vor allem danach der Spinner sehr klobig wäre. In die Leiterplatte wird ein Loch gebohrt, wo dann das Kugellager eingeklebt wird. Für besseren Halt werden Finger-Caps verwendet. Diese wurden selber mit FreeCAD gezeichnet und anschliessend 3D-gedruckt.

## Software

### Hauptsoftware

Die Buchstaben sind alle in einem 2-dimensionalem Array abgespeichert und sind nach der gleichen Reihenfolge wie in der ASCII-Tabelle geordnet was das Aufrufen mit Schleifen sehr einfach macht. Sie haben eine Auflösung von 5x7 Pixel und wurden als 5 8Bit Werte abgespeichert.

Der ganze Vorgang zum Berechnen wie lange jedes LED leuchten muss wird durch ein Interrupt vom Hall-Sensor ausgelöst. Sobald das genannte Interrupt ausgelöst wird fängt ein Timer an zu Zählen bis zum nächsten Interrupt, was eine halbe Umdrehung des Fidget Spinners entspricht. Danach kann mit der gemessenen Zeit ausgerechnet werden wie lange ein einzelner Abschnitt eines Buchstabens angezeigt werden muss. Ein Abschnitt im Kreis ist gleich eine Reihe LEDs.



# Schlussteil

## Fazit

Es war ein sehr spannendes Projekt mit vielen neuen Problemen denen wir gegenübergestellt wurden. Jedoch ist es schade, dass wir das Thema nicht selbst auslesen durften, da wir unseres als sehr schwierig empfanden und nicht alle unsere Ziele erfüllen konnten.

## Probleme

Kurz nachdem der Fidget Spinner zum ersten Mal vom PC erfolgreich eingerichtet und erkannt wurde ist die USB-Buchse abgerissen. Das Ganze war schon so instabil da es nicht viele Pads gab wo sie hätte befestigt werden können. Wir haben dann eine neue Buchse bestellt und diese dann verdrahtet. Zur besseren Befestigung wurde sie dann auch noch angeleimt.

Die USB-Kommunikation konnte nicht realisiert werden, da die Datenkommunikation nicht aufgebaut werden konnte. Der Fidget Spinner wurde zwar vom PC erkannt, aber es konnten keine Daten ausgetauscht werden. Als Ersatz zur USB-Schnittstelle wurde der Reset-Taster als normaler Eingabe- Taster umfunktioniert.

## Vergleich mit Pflichtenheft

Es wurden fast alle Obligatorischen Ziele erreicht. Es können 16 Zeichen mithilfe RGB- LEDs angezeigt werden. Die Farbe kann zwischen 7 Farben ausgewählt werden. Zusätzlich ist der Fidget Spinner über Knopfbatterien gespiesen und die Gewichtsverteilung ist auch gut.

Es wurde sich dagegen entschieden den Text an gleicher Stelle anzeigen zu lassen, aufgrund dessen, dass sich der Text relativ unleserlich unter dem Finger befinden könnte je nach Haltung der Finger-Caps.

Ein Ziel war es auch den Text mithilfe TeraTerm verändern zu können. Dies haben wir nicht erreicht, da die Serielle Schnittstelle nicht funktioniert. Jedoch kann man aus 3 verschiedenen, vordefinierten Texten mithilfe des Tasters wechseln.

Somit haben wir ausser einem obligatorischen Ziel alles erreicht was geplant war.

## Quellenverzeichnis

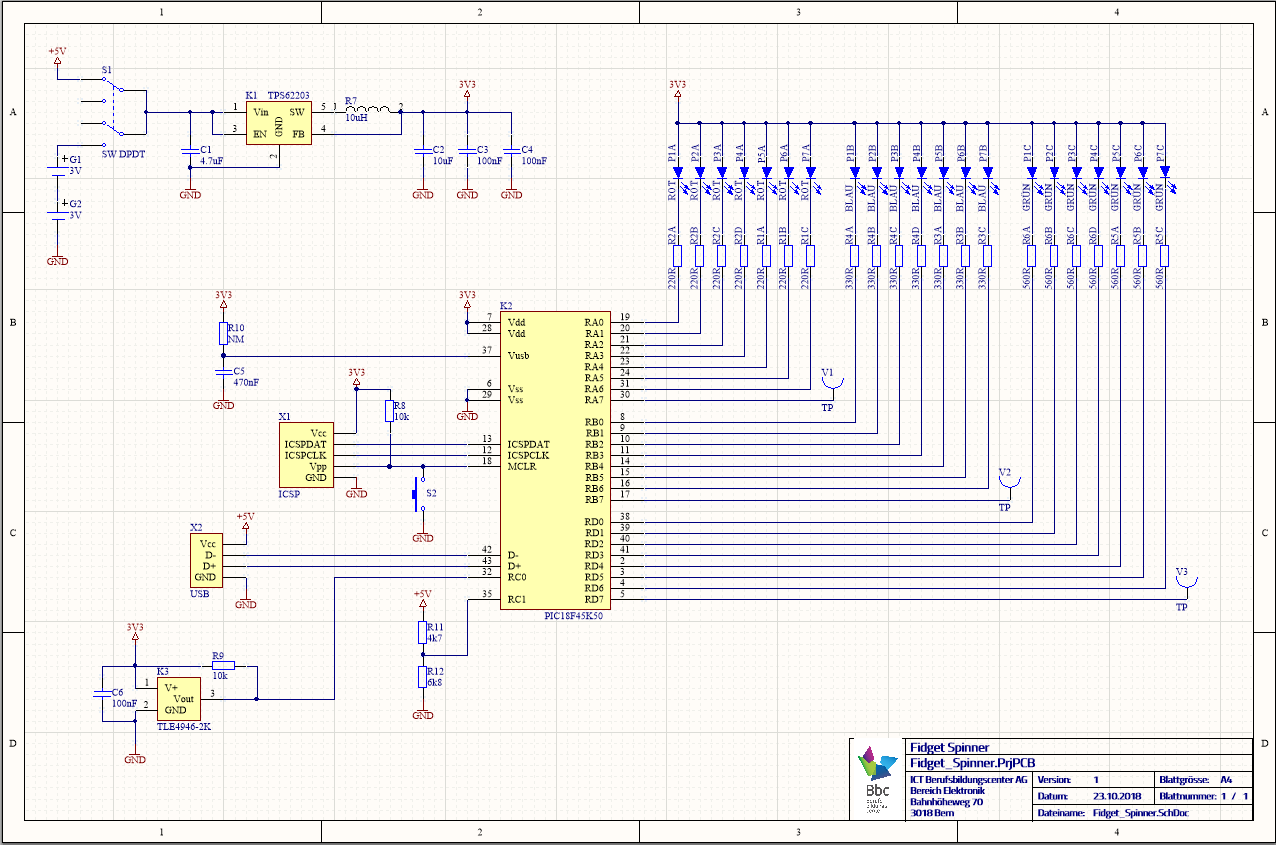
Microchip: PIC18(L)F2X/45K50, Microchip, 27.07.2018, <http://ww1.microchip.com/downloads/en/DeviceDoc/PIC18F2X_45K50-30000684B.pdf>

Würth Elektronik: WL-SFTW SMD TOP LED full-color waterclear, Würth Elektronik eiSos GmbH & Co. KG, 27.7.2018,<http://www.farnell.com/datasheets/1911491.pdf?_ga=2.146832631.1915304690.1540276737-1213338847.1534140400>

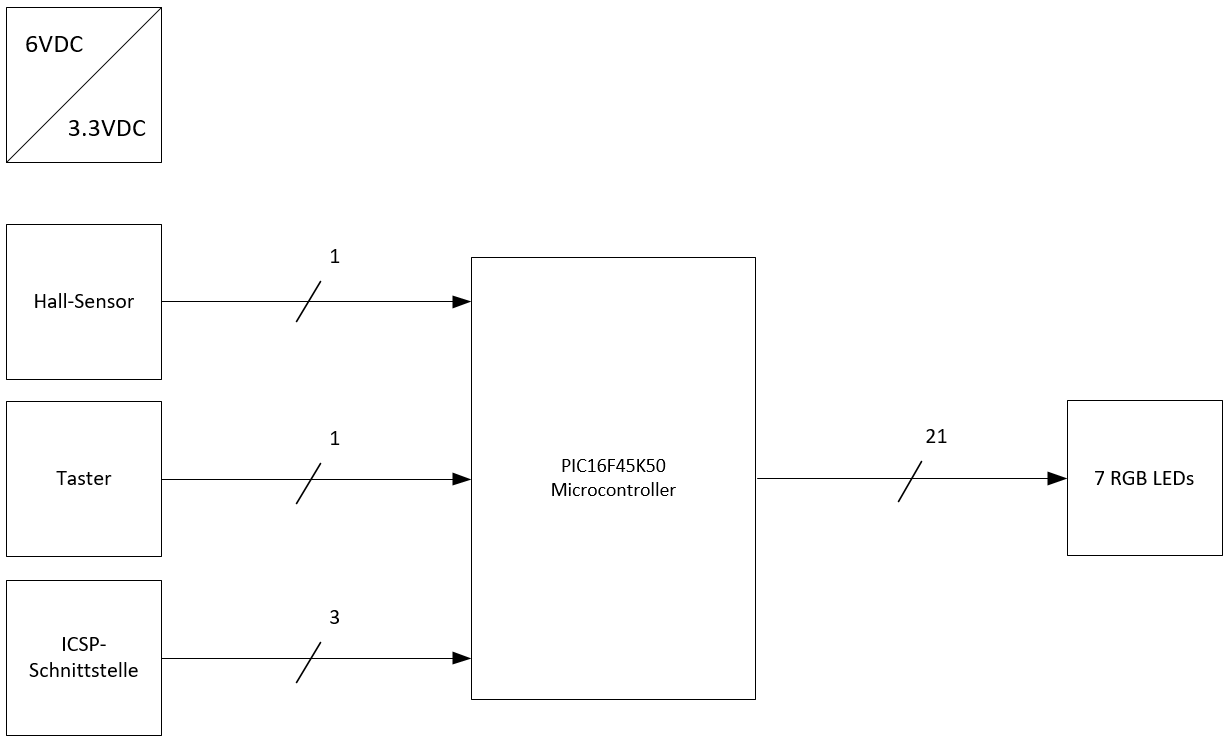
Infineon: TLE4946-2K, infenion, 10.10.2018, <http://www.farnell.com/datasheets/1932173.pdf?_ga=2.190443627.1915304690.1540276737-1213338847.1534140400>

Texas Instruments: TPS6220x High-Efficiency, Texas Instruments, 27.10.2018, <http://www.ti.com/lit/ds/symlink/tps62203.pdf>

1. Anhang
   1. Gesamtschema



* 1. Blockschaltbild



* 1. Pflichtenheft

**Beschreibung**

Ziel des Projekts ist es einen Fidget-Spinner zu realisieren, der mit dem PoV (Persistance of Vision) - Prinzip einen Text anzeigen kann. Dazu ist es nötig LEDs auf einem selbst konstruierten Fidget-Spinner zu installieren und diese im richtigen Moment anzusteuern. Mit der Trägheit des Auges «sehen» wir dann einen Text. Der angezeigte Text und die Farbe sollen jederzeit über die Serielle, Virtual COM-Port Schnittstelle verändert werden können. Dazu soll, wenn möglich, eine eigene Windows Applikation geschrieben werden. Ansonsten wird der Fidget Spinner mit dem Programm TeraTerm konfiguriert.

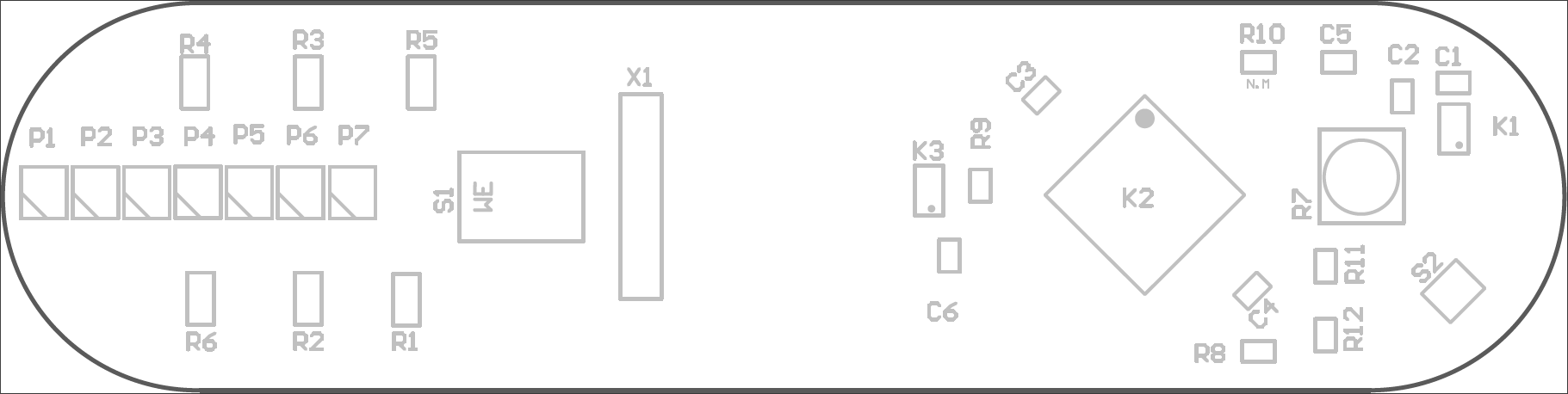
**Obligatorische Ziele**

* Einen Text anzeigen lassen (min. 15 Zeichen)
* Den Text verändern können (Tera Term)
* RGB LEDs für die Textanzeige verwenden
* Farben einstellbar (7 Farben)
* PIC-Mikrocontroller mit USB verwenden
* Gute Gewichtsverteilung auf den beiden Flügeln des Fidget-Spinners
* Speisung über Batterien
* Schrift bleibt kontinuierlich an derselben Stelle

**Optionale Ziele**

* Farben dimmbar (volles Farbspektrum)
* Eigene Windows-Software (C#) zum definieren des Textes bzw. der Farbe schreiben
  1. Bestückungsplan und Stückliste

**Oberseite**



**Stückliste Oberseite**

|  |  |  |  |
| --- | --- | --- | --- |
| Menge | Position | Bezeichnung | Händler |
| 1 | C1 | Cap, Cer, 4,7uF, 0805 | Distrelec |
| 1 | C2 | Cap, Cer, 10uF, 0805 | Distrelec |
| 3 | C3, C4, C6 | Cap, Cer, 100nF, 0805 | Distrelec |
| 1 | C5 | Cap, Cer, 470nF, 0805 | Distrelec |
| 1 | K1 | Buck-Converter, TPS62203, SOT23-5 | Distrelec |
| 1 | K2 | Microcontroller, PIC18F45K50, TQFP44 | Farnell |
| 1 | K3 | Hall-Effekt-Sensor, TLE4946-2K, SOT23-3 | Farnell |
| 7 | P1 – P7 | RGB-LED, 3528 | Farnell |
| 2 | R1, R2 | Widerstandsnetzwerk, 220R, 0612 | Distrelec |
| 2 | R3, R4 | Widerstandsnetzwerk, 330R, 0612 | Distrelec |
| 2 | R5, R6 | Widerstandsnetzwerk, 560R, 0612 | Farnell |
| 1 | R7 | Spule, 10uH | Farnell |
| 2 | R8, R9 | Res, 10k, 0805 | Distrelec |
| 1 | R11 | Res, 4,7k, 0805 | Distrelec |
| 1 | R12 | Res, 6,8k, 0805 | Distrelec |
| 1 | S1 | Switch on-on, SW DPDT | Distrelec |
| 1 | S2 | Taster, B3U | Farnell |
| 1 | X1 | ICSP-Programmierschnittstelle, 5-Pin Stiftleiste | Distrelec |
| 1 | X2 | Micro-USB Buchse, USB AB | Distrelec |

**Unterseite**



**Stückliste Unterseite**

|  |  |  |  |
| --- | --- | --- | --- |
| Menge | Position | Bezeichnung | Händler |
| 2 | G1, G2 | 16mm Knopfbatterie 3V | Farnell |
| 1 | X2 | Micro-USB Buchse, USB AB | Distrelec |

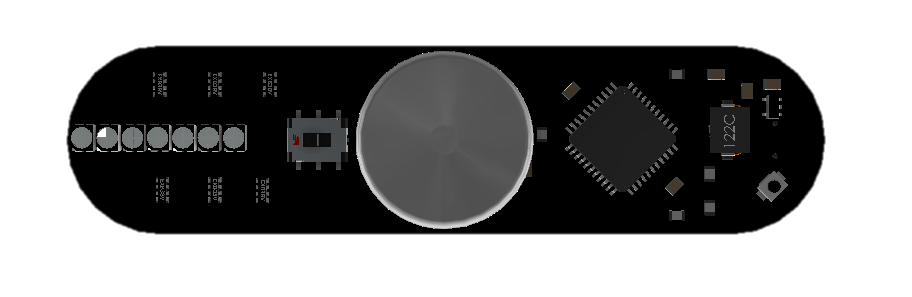
* 1. Bedienungsanleitung

**Text anzeigen**

Um einen Text anzeigen zu können muss nur der Schalter in Richtung Mitte gestossen werden und der Spinner gedreht werden.

**Farbe wechseln**

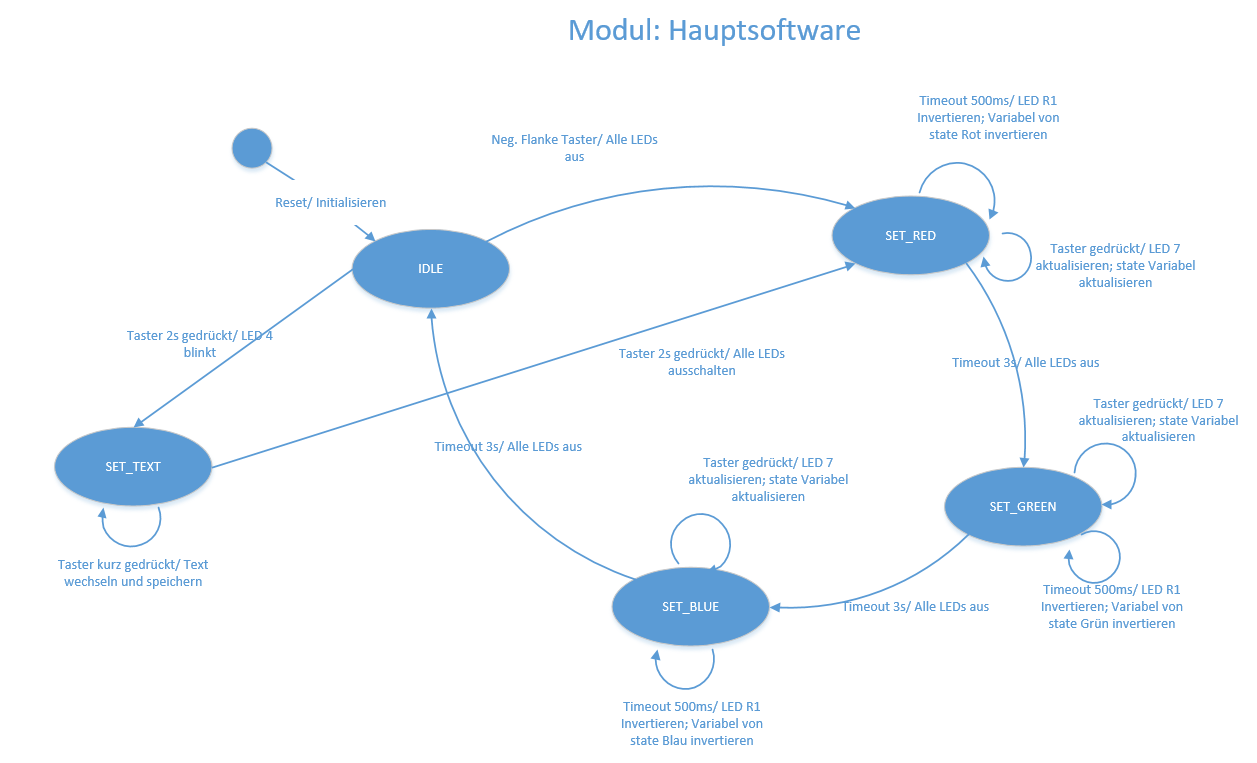
Um eine Farbe zu wechseln muss zuerst der Taster kurz gedrückt werden. Anschliessend blinkt die 1. LED in den Farben Rot, Grün und Blau. Bei jeder Farbe kann mit einem kurzen Tastendruck entschieden werden ob die Farbe leuchten soll oder nicht (Farbmischung ist möglich). Standartmässig leuchten die LEDs in Grün.



Taster

Power-Schalter OFF/ON

* 1. Zustandsdiagramm Hauptsoftware



* 1. Zeitplan

