

Maturitätsarbeit an der Kantonsschule Zürich Nord

# Regelungstechnik

## PID-Parameter anhand einem Quadrocopter erforschen

Charpoan Kong

M6d

Kantonsschule Zürich Nord

12. Oktober 2019

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Was ist ein Quadrocopter?</b>	<b>2</b>
<b>3</b>	<b>PID-Regler und Filter</b>	<b>3</b>
3.1	Was ist ein Regler? . . . . .	3
3.1.1	P-Regler . . . . .	3
3.1.2	I-Regler . . . . .	4
3.1.3	D-Glied . . . . .	4
3.1.4	PID-Regler . . . . .	5
3.2	Digitaler Low-Pass Filter . . . . .	5
<b>4</b>	<b>Äusserer Aufbau</b>	<b>6</b>
4.1	Motoren . . . . .	6
4.2	Akku . . . . .	6
<b>5</b>	<b>Der Flightcontroller</b>	<b>7</b>
5.1	Komponenten . . . . .	7
5.1.1	Microcontroller . . . . .	7
5.1.2	Sensoren . . . . .	7
5.1.3	Empfänger . . . . .	8
5.1.4	Schnittstellen . . . . .	9
5.2	Leiterplatte . . . . .	9
5.2.1	Design . . . . .	9
5.2.2	Bau . . . . .	10
5.3	Programmierung . . . . .	11
5.3.1	Motoren und Timing . . . . .	11
5.3.2	PID Implementierung . . . . .	11
<b>6</b>	<b>Fernbedienung</b>	<b>11</b>
<b>7</b>	<b>Versuche</b>	<b>11</b>
7.1	PID absoluter Winkel oder Winkelgeschwindigkeit . . . . .	11
7.1.1	Absoluter Winkel . . . . .	11
7.1.2	Winkelgeschwindigkeit . . . . .	11
7.1.3	Ergebnisse . . . . .	11
7.2	Filter . . . . .	11
7.2.1	Ohne Filter . . . . .	11
7.2.2	Filter 1 . . . . .	11
7.2.3	Filter 2 . . . . .	11
7.2.4	Ergebnisse . . . . .	11

<b>8</b>	<b>Schlussfolgerung</b>	<b>11</b>
<b>9</b>	<b>Glossar</b>	<b>12</b>
<b>10</b>	<b>Quellenverzeichnis</b>	<b>13</b>
10.1	Literaturverzeichnis . . . . .	13
10.2	Abbildungsverzeichnis . . . . .	13
10.3	Personenverzeichnis . . . . .	13

# 1 Einleitung

Ein Quadrocopter ist ein Flugobjekt mit 4 Propellern. In der Vergangenheit wurden schon Versuche mit Flugobjekten mit 4 Propellern gemacht z.B. wie der Luftfahrtpionier Étienne Oehmichen. der 1920 den Oehmichen No. 2 gebaut hatte. Damals waren die Propeller elastisch und man konnte mit Seilzügen den Anstellwinkel der Propeller einstellen. Er konnte damit einige Rekorde aufstellen.[5]

In der heutigen Zeit finden Quadrocopter viele Anwendungen, wie z.B. bei Suchaktionen, im Militär, oder auch als Spielzeug. Mit der schnellen entwicklung von Mikroprozessoren wurden Quadrocopter immer kleiner und einfacher zu realisieren. Auch können damit teurere Helikopterflüge für Kartographie oder Luftaufnahmen billiger realisiert werden.

Mit dieser Arbeit will ich mit dem Bau und dessen Regelung befassen. Der Quadrocopter wird selbst gebaut mit dem Hauptziel den Flightcontroller selbst zu designen, bauen und programmieren. So sollte die Regelung selbst implementiert werden. Dannach sollten Versuche gemacht werden, die bestimmen sollten, wie der PID-Regler stabiler gemacht werden kann und wie ungleichheiten, wie sie durch Vibration entstehen, auszugleichen.

## 2 Was ist ein Quadrocopter?

Ein Quadrocopter ist wie in der Einleitung beschrieben ein Flugobjekt. Man könnte meinen, dass man einen stabilen Flug erreichen kann, wenn man den vier Propellern den gleichen Schub gibt. Aber dies ist leider nicht möglich, da viele Umwelteinflüsse einen stabilen Zustand verhindern wie z.B. Wind, unterschiede in den Rotoren oder Motoren, assymetrie des Rahmens etc. So muss ein Computer, der Flightcontroller(FC), das Gleichgewicht erhalten, indem dieser den Schub der vier Motoren so steuert das der Quadrocopter im Gleichgewicht ist oder dieser einen anderen bestimmten Winkel hält.

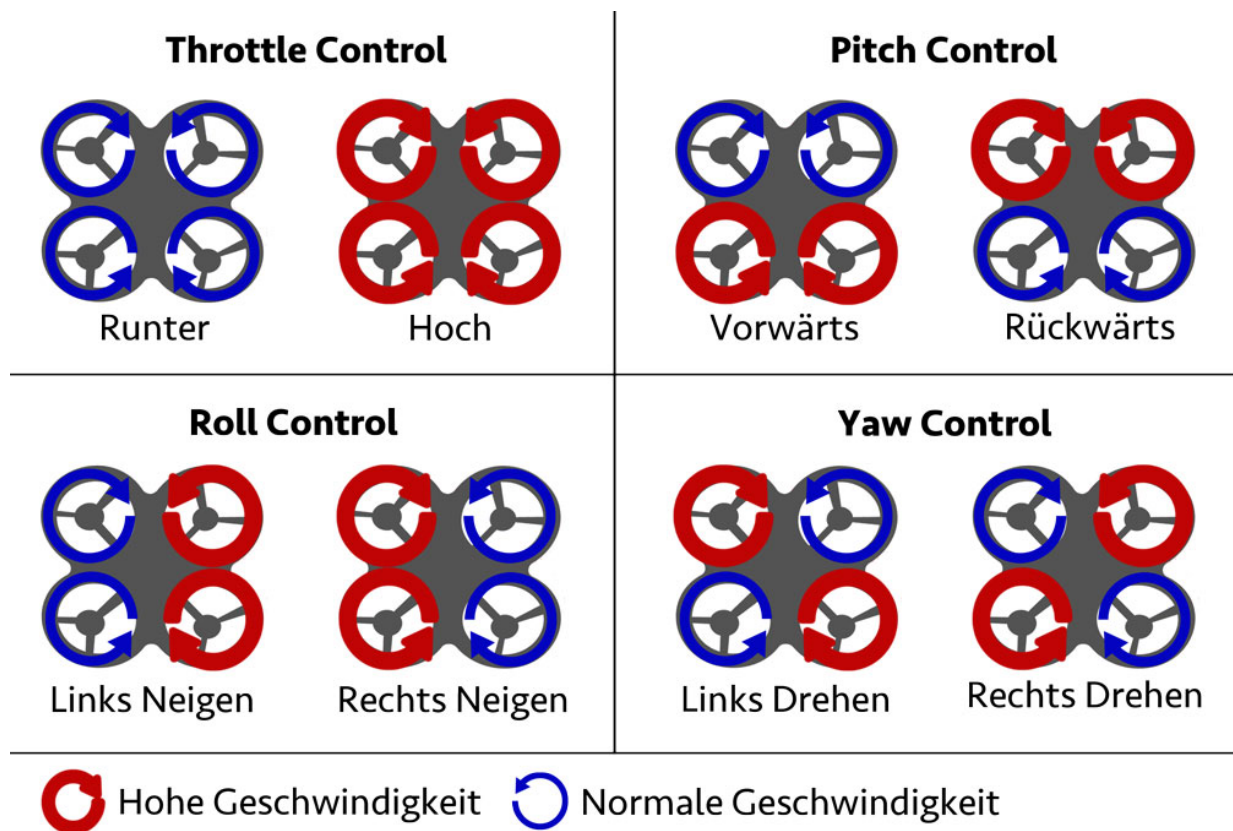


Abbildung 1: Drehrichtung und Schub bei verschiedenen Steuereingaben

Man sieht in dieser Abbildung welche Rotationsrichtung die Motoren haben. Die benachbarten Motoren müssen eine gegensätzliche rotation haben, da sonst der Quadrocopter zum rotieren kommt. So wird das Drehmoment des anderen Motors ausgeglichen. Auch wird gezeigt, dass man den Quadrocopter mit erhöhen bzw. erniedrigen der Motorleistung der richtigen Motoren, diesen in die gewünschte Richtung steuern kann. Der FC hält dieses Gleichgewicht mit einem Proportional-,Integral- und Differenzial-Regler(PID-Regler).

## 3 PID-Regler und Filter

### 3.1 Was ist ein Regler?

Eine Regelung ist eine Steuerung mit Rückkoppelung. Es wird ein Wert z.B. die Drehzahl eines Motors überwacht und je nach gewünschter Drehzahl, das Drehmoment des Motors geregelt, dass er auch diesen halten kann, auch wenn eine Last am Motor hängt.[3]

Der Regelkreis misst die Regelgröße z.B. mit einem Sensor. Dann wird dieser mit dem Soll-Wert verglichen um die Regelabweichung zu bestimmen. So kann ein z.B. Motor gestellt werden damit dieser sich dem Soll-Wert nähert.

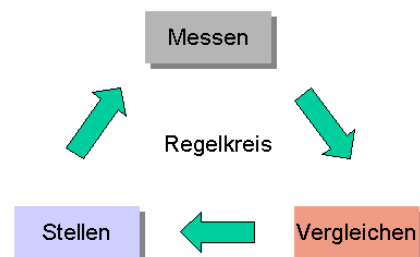


Abbildung 2: Der Regelkreis

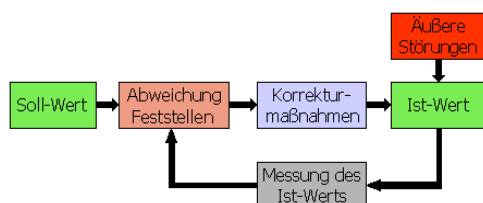
Die Regelabweichung kann dabei einfach mit der Differenz des Ist-Wertes  $x$  mit dem Soll-Wertes  $w$  bestimmt werden.

$$e(t) = w - x \quad (1)$$

In der Abbildung 3 sieht man, wie vorher beschrieben, wie ein solcher Regelkreis funktioniert. In der Regelungstechnik wird versucht einen solchen Regelkreis mathematisch zu modellieren. In dem Kapitel wird nur der PID-Regler und seine Komponenten besprochen, da dieser häufig bei Quadrocoptern benutzt wird.

Der PID-Regler besteht aus mehreren Reglern und dem D-Glied. Unten wird beschrieben wie die einzelnen Komponenten wirken.

#### 3.1.1 P-Regler



Der P-Regler wirkt linear. Dieser Regler gibt die Regelabweichung verstärkt und unverzögert mit dem Faktor  $K_p$  weiter. Das Problem dabei ist, dass diese Abweichung bleibend ist und somit den Soll-Wert über- bzw. unterschiesst. Dieser Regler wirkt mittelschnell.[3]

Abbildung 3: Wirkungsweise des Regelkreises

$$y(t) = Kp * e(t) \quad (2)$$

```

1 void P_Regler(float Kp, float error, float &P){
2     P = kp * error;
3 }

```

Listing 1: P-Regler C++ Pseudocode

### 3.1.2 I-Regler

Beim Integralregler wird die Regelabweichung über die Zeit summiert und mit dem Faktor  $Ki$  verstärkt. Dabei werden Abweichungen vollständig eliminiert, da dieser Regelwert immer anwächst solange die Regelabweichung nicht Null ist. Dieser Regler wirkt langsam.[3]

$$y(t) = Ki \int_0^t e(t) dt \quad (3)$$

```

1 void I_Regler(float Ki, float error, float &I ){
2     I += I+error*Ki;
3 }

```

Listing 2: I-Regler C++ Pseudocode

### 3.1.3 D-Glied

Das Differenzialglied schaut auf die Differenz der Regelabweichung zur vorherigen Regelabweichung und wird mit dem Faktor  $Kd$  verstärkt. Deshalb reagiert dieser sehr schnell und gibt den beiden anderen Vorhaltezeit. Differenzialglied ist kein Regler da es alleine nichts regeln kann sondern nur auf Veränderungen in der Regelabweichung reagiert.[3]

$$y(t) = Kd * e(t) \frac{d}{dt} \quad (4)$$

```

1 void D_Glied(float Kd, float error, float &previous_error, float dt,
2     float &D){
3     D = Kd * (error - previous_error);
4     previous_error = error;
5 }

```

Listing 3: D-Regler C++ Pseudocode

### 3.1.4 PID-Regler

Beim PID-Regler werden die Eigenschaft der einzelnen Regler und dem D-Glied vereint.

$$y(t) = Kp * e(t) + Ki * \int e(t) dt + Kd * e(t) \frac{d}{dt} \quad (5)$$

```
1 void PID_Regler(float P, float I, float D, float &PID){  
2     PID = P + I + D;  
3 }
```

Listing 4: D-Regler C++ Pseudocode

## 3.2 Digitaler Low-Pass Filter

Ein Low-Pass Filter filtert hohe Frequenzen aus. Man kennt diesen Filter meist aus der Elektronik oder der Tontechnik. Beim Quadrocopter wird ein solcher Filter für das unterdrücken des Rauschens der IMU verwendet. Dieses Rauschen wird durch die Vibrationen der Motoren erzeugt und liegt ca. bei 133Hz.

Beim Digitalen Low-Pass Filter(DLPF) wird ein analoger RC-Low-Pass Filter emuliert. Herleitung?

Vibrationen verursachen beim D-Glied starke ausschläge, wenn diese hohe Amplituden haben. Auch wird dieser dann unbrauchbar, da die Regelabweichung durch die Vibrationen verfälscht wird.



## 4 Äusserer Aufbau

### 4.1 Motoren

Die verwendeten Motoren sind sogenannte bürstenlose Motoren. Diese werden mit einem Electronic Speed Controller(ESC) angesteuert, da diese wie Drehstrommotoren funktionieren. Da ihre RPM von Spannung abhängig ist gibt man noch einen  $Kv$ -Wert an. So kann man mit der Formel

$$RPM = Kv * U \quad (6)$$

die Rotationen pro Minute berechnen.

Beim meinem Quadrocopter handelt es sich um ReadyToSky RS2205 Motoren mit einem  $Kv$  von 2300. Ich habe diese Motoren ausgesucht, da sie für genug Leistung bringen für einen Quadrocopter mit einer Rahmen länge von 250mm.(Bild 1)

### 4.2 Akku

Der Akku versorgt alle Komponenten, die Strom brauchen mit Strom. Im Quadrocopter wird ein LiPo Akku benutzt, der drei Zellen in Serie geschaltet hat mit einer Kapazität von 2.4Ah. Die minimale Spannung beträgt 11.1V. Es gibt noch das Entladerate, die als  $C$ -Rate angegeben wird. Mit der Kapazität multipliziert gibt es den maximalen Entladestrom an.[1]

$$A_{max} = Q * C \quad (7)$$

(Bild r)

Man benutzt in Quadrocoptern meist LiPo Akkus, da sie eine hohe Energiedichte besitzen und viel Leistung abgeben können. Das aber hat auch zur Folge, dass man diese Akkus sorgfältig behandeln muss, da sie auch im Extremfall, Feuer fangen oder explodieren können. So darf man die einzelnen Zellen nicht unter einer Spannung von 2.7V bringen. Deshalb wird der Akku auch an einem Akkuüberwacher angeschlossen, der piepst, wenn der Akku verbraucht ist. Auch braucht man ein Balancerladegerät, das kontrolliert, dass alle Zellen gleichmässig geladen werden. Eine hohe Kapazität garantiert nicht immer eine längere Flugzeit, da mehr Kapazität auch mehr Gewicht bedeutet.

## 5 Der Flightcontroller

Der Flightcontroller ist das Herz des Quadrocopter er liest alle Sensoren und gibt den ESCs die Steuerimpulse. In dem Kapitel wird zuerst in die wichtigsten Komponenten eingegangen, dann wird der Design-, Bau und Programmierprozess beschrieben.

### 5.1 Komponenten

#### 5.1.1 Microcontroller

Der Microcontroller der auf dem FC benutzt wird ist ein ARM-Cortex-M7 STM32F722RET6 Microcontroller von STMicroelectronics. Dieser besitzt alle Schnittstellen um die einzelnen Sensoren und andere Komponenten auszulesen und anzusteuern. Da dieser nicht genug Pins hat, wird noch ein zweiter Microcontroller benutzt und zwar der ATmega328P, der auch auf dem Arduino/Genuino UNO zu finden ist. Dieser liest den Drucksensor und einige ADCs aus und kommuniziert mit dem I2C-Protokoll mit dem STM32.

Microcontroller können anders als Mikroprozessoren ohne Zusatz Pehpetrie benutzt werden, da sie schon einen Flash, RAM etc. besitzen. Aber dies kommt mit einer tieferen Taktrate daher. Dies stört aber den Quadrocopter ist da diese völlig ausreichend ist. Der Hauptcontroller läuft mit einer Taktfrequenz von 200MHz.

#### 5.1.2 Sensoren

Die wichtigsten Sensoren des Quadrocopter sind der Gyrosensor und der Beschleunigungssensor. In dem FC wurde eine ICM-20689 von TDK InvSense benutzt. Dieser hat beide Sensoren in einem Chip integriert und wird per SPI-Protokoll ausgelesen. Als Backup wird ein MPU-6050 benutzt, auch vom gleichen Hersteller, dieser wird mit dem langsameren I2C-Protokoll ausgelesen.

Beide sind sogenannte MEMS-Sensoren, diese werden Direkt auf den Siliziumwaver der Chips realisiert. (Bild) Man ätzt die Sensorstrukturen und deren auslese Schaltung auf einen Chip. MEMS bedeutet microelectromechanical systems. Das bedeutet, dass diese Sensoren auf einer mechanischen Basis funktionieren.

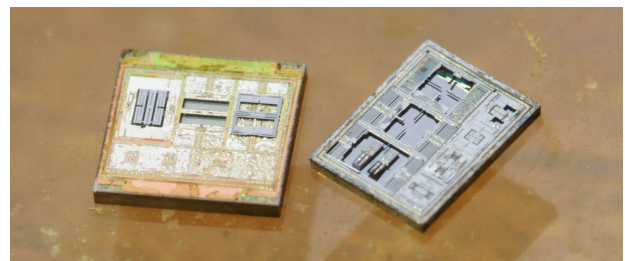


Abbildung 4: Dice des MPU6050

Da der Gyrosensor nur Winkelgeschwindigkeit, bei den Sensoren  $\frac{^\circ}{s}$  und nicht  $\frac{rad}{s}$  misst muss diese noch mit der Loop-Zeit verrechnet werden um den Winkel zu bekommen.

$$Grad = \omega * dt \quad (8)$$

Wenn man den Absoluten Winkel alleine mit dem Gyrosensor misst hat dieser in der Zeit einen Drift. Zum einen wird das durch das Rauschen im Sensor veruracht und zu anderen auch von einem nicht genauen Loop-Zeit. Auch kann nicht der Anfangswinkel bestimmt werden und man müsste den Quadrocopter immer vom Boden aus starten. Deswegen braucht man da den Beschleunigungssensor. Dieser misst die Beschleunigung auf allen Achsen relativ zur Erdbeschleunigung. Beim Stillstand misst er also  $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \vec{g}$ . Man kann einfach die Norm des Vektors nehmen und mit der Beschleunigung an der gewünschten Achse verrechnen um dann den Winkel zu bekommen.

$$Grad = \left\| \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \right\| \vec{g} * \frac{180}{\pi} \quad (9)$$

So hat man den absoluten Winkel mit zwei Sensoren gemessen und muss diese nur noch Kombinieren. Dies geschieht mit der Formel des Komplementärfilters. Man muss auch beachten das man nur den Absoluten Winkel nur für die Nick-und Rollachse berechnen kann da, für die Gierachse noch ein Magnetometer nötig wäre für eine Null-Referenz.

$$Grad_{komp} = 0.96 * Grad_{Gyro} + 0.04 * Grad_{Accl} \quad (10)$$

Man gibt dem Winkel aus dem Beschleunigungssensor einen tiefen Faktor da dieser Winkel nicht mehr genau stimmen wird, wenn der Quadrocopter in eine Richtung beschleunigen würde, dies wird erst bei starken Beschleunigungen der Fall sein, sonst stimmt dieser Wert ungefähr. Aber so kann man den Anfangswinkel des Quadrocopter bestimmen, ihn auch aus der Hand starten lassen und bekommt auch noch stabilere Werte. (Barometer ?)

### 5.1.3 Empfänger

Als Empfänger wird ein NRF24L01 Breakoutboard mit PA und LNA benutzt. Der ist ein 2.4GHz Sender und Empfänger von der Firma Nordic Semiconductor. Die Kommunikation erfolgt digital und der Chip besitzt auch eine Zyklische Redundanzprüfung(CRC). Damit wird kontrolliert, ob ein Datenpaket fehlerhafte Bits besitzt. Allenfalls wird es repariert oder verworfen. PA heisst Power Amplifying und bedeutet, dass das Signal verstärkt wird und so grössere Reichweiten erreicht werden können und LNA bedeutet Low Noise Amplifying und bedeutet, dass schwache Empfangssignale verstärkt werden.[2] Vom Hersteller wird eine Reichweite von bis zu einem Kilometer versprochen aber in einer Wohnsiedlung kommt man mit Sichtkontakt bis zu 300m, was ausreichend genug ist.

### 5.1.4 Schnittstellen

Damit der Mikrocontroller mit den Sensoren und der restlichen Pehpetrie kommunizieren kann braucht es Schnittstellen und Protokolle die diese Komponenten verbindet. Zuerst wird

## 5.2 Leiterplatte

Auf Leiterplatte oder auch Printed Circuit Board(PCB) wird die Schaltung des FC realisiert. Dieses besteht aus Kunststoff mit den aufgeätzten Kupferbahnen und den verzinnten Lötstellen.

### 5.2.1 Design

Zuerst habe ich mir überlegt was für einen Microcontroller ich wählen sollte. Bei meinen ersten Versuchen mit dem Raspberry gingen schief, da der Regler nicht schnell genug reagierte und eine nicht konstantes Regelverhalten gezeigt hat. Einerseits lief ein Betriebssystem parallel zum Flightcontroller, andererseits war die Regelung nicht Echtzeit genug. Dessenwegen habe ich im Internet nachgeschaut was für Microcontroller auf den meisten Quadroopter verbaut sind. Dort werden meist MCU aus der STM32-Familie verwendet. So kam ich auf die Webseite von Oscar Liang<sup>1</sup> bei der die einzelnen MCU-Generationen aufgelistet waren mit den Vor- und Nachteilen. Ich entschied mich für die neuste Generation aus dieser MCU-Familie, einen STM32F7xx MCU zu benutzen. Nachher habe ich vorhandene Boards angeschaut und gesehen, dass die meisten den STM32F722RET6 benutzen aus dieser MCU-Generation. Als nächstes überlegte ich mir was für einen Gyro- und Beschleunigungssensor ich benutzen wollte. Ich habe schon eine MPU-6050 IMU aber ich wollte noch einen mit SPI-Protokoll, da dies schneller ist, und der andere als Backup, so recherchierte ich was für IMU's die FC's benutzen<sup>2</sup> und entschied mich für den ICM-20689. Als Empfänger habe ich einen nRF24L01 PA + LNA gewählt. Als Backup wird noch ein PPM Eingang eingebaut für die standard Funkübertragung über eine normale Fernbedienung. Da für die Versuche Daten gebraucht werden, wird eine microSD-Karten Schnittstelle eingebaut, um die Daten aufzuzeichnen. Damit der Flightcontroller mit dem PC kommunizieren kann um zu Debuggen muss ein FTDI-Chip(F232RL) eingebaut werden damit wird das UART-Protokoll, in ein für den PC verständliches Protokoll übersetzt. Um die Höhe zu bestimmen habe ich mich für den BMP280 Drucksensor ausgewählt um die barometrische Höhe zu bestimmen. Da dieser für das Auslesen eine lange Zeit braucht habe ich überlegt noch eine NebenMCU einzubauen. Deshalb habe ich den ATmega328 gewählt da er auch auf den Arduino UNO drauf ist und ich mich damit schon auskenne. Über diese MCU werden auch die Ultraschallsensoren ausgelesen, die eine genauere Höhenmessung ermöglicht, aber nur bis zu sechs Metern. Da die ESC's einen 5V Versorgungsausgang haben muss noch ein Span-

---

<sup>1</sup><https://oscarliang.com/f1-f3-f4-flight-controller/> aufgerufen am 31.03.2019

<sup>2</sup><https://blog.dronetrest.com/inertial-sensor-comparison-mpu6000-vs-mpu6050-vs-mpu6500-vs-icm20602/> aufgerufen am 31.03.2019

nungsregler eingebaut werden, da die meisten IC's 3.3V brauchen, der Schulmechaniker Herr Thurnherr hat mir den LM3940 empfohlen.

Die Leiterplatte wird mit dem Program Eagle von Autodesk designt. In dem Programm kann man die einzelnen Bauteile zusammensetzen und verbinden(Siehe Abbildung 5).Ich musste nach dem Datenblatt der einzelnen IC's, Kondensatoren und Widerstände einbauen. Die Schaltbilder für die einzelnen IC's, die nicht schon in der Standardbibliothek vorhanden sind, habe ich aus dem Internet heruntergeladen. Herr Thurnherr hat mich dabei unterstützt und alle Fragen beantwortet. Nach mehreren Kontrollen wurde die Leiterplatte bei JLCP-CB und die einzelnen Teile bei LCSC, Arrow, RS und Conrad bestellt. Dazu wurde noch eine Schablone für das SMD-löten bestellt.

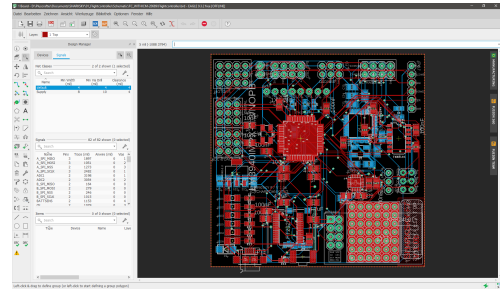


Abbildung 5: Autodesk Eagle

### 5.2.2 Bau

Beim Zusammensetzen des FC wird ein spezielles Lötverfahren gebraucht und zwar das SMD reflowing. Damit man die Komponenten nicht per Hand löten muss wird mit einer Schablone Lötpaste, die Flussmittel und das Lötzinn erhalten, aufgetragen. Dann wird die Schablone entfernt und die Teile werden platziert. Dies geschieht im industriellen Rahmen mit sogenannten "Pick and Place" Maschinen. Später wird die Leiterplatte in den Ofen getan und nach einer bestimmten Temperaturkurve erhitzt, bis die Komponenten verlötet sind[4].

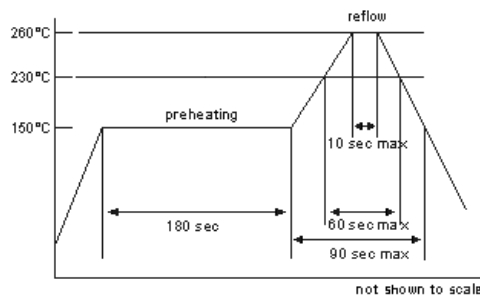


Abbildung 6: Temperaturkurve für Reflowlöten

Da die Leiterplatte doppelseitig ist kann man hier nicht auf beiden Seiten das Reflow-Verfahren benutzen. Deshalb wurde Zuerst auf der Rückseite Lötpaste für den ATmega,

dessen Quarzoszillator und dem SD-Kartenslot aufgetragen und die Komponenten wurden platziert. Dann wurden diese mit dem Heissluftföhn verlötet. Um das Herausfallen der Teile zu verhindern werden sie mit hitzebeständigen Klebeband fixiert. Dann wurde auf einer Holzplatte Löcher an den Positionen der Verlöteten Komponenten gebohrt. Damit wird erreicht, dass man die Vorderseite, Plan auf dieser Holzplatte gelegt werden kann. Mit den restlichen PCBs, man kann nur fünf auf einmal bestellen, wurde ein Rahmen erstellt um so das teilweise gelötete Board zu fixieren. Dann wurde die Schablone ausgerichtet und festgemacht. So konnte man mit Lötpaste und einem Spachtel die Löcher und somit auch die Lötstellen mit Lötpaste füllen. Dann wurde die Schablone entfernt. Mithilfe des Mikroskops wurden die Teile dann auf der Leiterplatte platziert. Im Ofen wurden sie dann ungefähr nach der Temperaturkurve gebacken und so war die Vorderseite gelötet. Die Hinterseite wurden dann per Hand fertig gelötet worden. Dann wurden nur noch die Konnektoren angelötet und der Flightcontroller ist fertig zum programmieren.

## **5.3 Programmierung**

### **5.3.1 Motoren und Timing**

### **5.3.2 PID Implementierung**

## **6 Fernbedienung**

## **7 Versuche**

### **7.1 PID absoluter Winkel oder Winkelgeschwindigkeit**

#### **7.1.1 Absoluter Winkel**

#### **7.1.2 Winkelgeschwindigkeit**

#### **7.1.3 Ergebnisse**

### **7.2 Filter**

#### **7.2.1 Ohne Filter**

#### **7.2.2 Filter 1**

#### **7.2.3 Filter 2**

#### **7.2.4 Ergebnisse**

## **8 Schlussfolgerung**

## 9 Glossar

## 10 Quellenverzeichnis

### 10.1 Literaturverzeichnis

#### Literatur

- [1] fpvracing.ch. *Multicopter Komponenten*. Webseite. Oktober 2019. URL: <https://fpvracing.ch/de/content/8-multicopter-komponenten>.
- [2] Sparky256 (Fiktiver Name). *What is a PA/LNA?* Webseite. Oktober 2019. URL: <https://electronics.stackexchange.com/questions/237267/what-is-a-pa-lna/237278>.
- [3] Waste(Fiktiver Name). *Regelungstechnik*. Webseite. Oktober 2019. URL: <https://de.wikipedia.org/wiki/Quadrocopter#Entwicklung>.
- [4] sauter-elektronik.de. *What is a PA/LNA?* Webseite. Oktober 2019. URL: <https://www.sauter-elektronik.de/wissenswertes/reflow-loeten>.
- [5] Wikipedia. *Quadrocopter*. Webseite. Oktober 2019. URL: <https://de.wikipedia.org/wiki/Quadrocopter#Entwicklung>.

### 10.2 Abbildungsverzeichnis

#### Abbildungsverzeichnis

1	<a href="https://fpvracing.ch/de/content/7-grundsatzliche-funktion-quadrocopter-multicopter">https://fpvracing.ch/de/content/7-grundsatzliche-funktion-quadrocopter-multicopter</a>	2
2	<a href="https://rn-wissen.de/wiki/images/2/25/Regelkreis1.png">https://rn-wissen.de/wiki/images/2/25/Regelkreis1.png</a>	3
3	<a href="https://rn-wissen.de/wiki/images/5/5d/Regelkreis2.png">https://rn-wissen.de/wiki/images/5/5d/Regelkreis2.png</a>	3
4	<a href="https://de.wikipedia.org/wiki/TDK">https://de.wikipedia.org/wiki/TDK</a>	7
5	Screenshot Eagle FC brd file	10
6	<a href="http://www.comtec-crystals.com/service-5.php">http://www.comtec-crystals.com/service-5.php</a>	10

### 10.3 Personenverzeichnis