

Maturitätsarbeit an der Kantonsschule Zürich Nord

Regelungstechnik PID-Parameter anhand einem Quadrocopter erforschen

Charpoan Kong

M5d

Kantonsschule Zürich Nord

23. September 2019

Inhaltsverzeichnis

1	Einleitung	1
2	Bau und programmierung des Quadrocopters	2
2.1	Der Flightcontroller	2
2.1.1	Genereller Aufbau	2
2.1.2	Überlegungen beim Bau	2
2.1.3	Leiterplatten Design und Bau	3
2.1.4	Programmieren	4
2.2	Die Fernbedienung	4
2.2.1	Konstruktion	4
2.2.2	Programmieren	4
3	Glossar	5
4	Quellenverzeichnis	6
4.1	Literaturverzeichnis	6
4.2	Abbildungsverzeichnis	6
4.3	Personenverzeichnis	6

1 Einleitung

2 Bau und programmierung des Quadrocopters

2.1 Der Flightcontroller

Der Flightcontroller ist sozusagen das Herz des Quadrocopters. Er steuert die Motoren an, liest und verarbeitet die Daten der Sensoren, die für die Regelung wichtig sind.

2.1.1 Genereller Aufbau

Der Flightcontroller besteht aus mehreren IC's die alle eine eigene Funktion haben. Der Hauptmicrocontroller ist ein STM32F722RET6. Dieser ist ein 32-Bit ARM-Cortex-M7 controller mit einer maximalen Taktfrequenz von 216Mhz. Dieser liest die zwei IMU's aus, regelt und steuert die Motoren. Auch liest und schreibt dieser auf die microSD-Karte. Durch zwei UART-Schnittstellen kann er seriell mit einem Computer und einen GPS-Empfänger kommunizieren. Der Controller liest auch den Empfänger 2.4 Ghz aus. Als Nebemicrocontroller habe ich einen ATmega328 benutzt, der auch auf dem Arduino Uno benutzt wird. Dieser liest den Drucksensor, die Ultraschallsensoren und zwei ADC Eingänge aus. Er kommuniziert per I2C mit dem Hauptmicrocontroller.

2.1.2 Überlegungen beim Bau

Zuerst habe ich mir überlegt was für einen Microcontroller ich wählen sollte. Bei meinen ersten Versuchen mit dem Raspberry hat dies nicht geklappt. Einerseits lief ein Betriebssystem parallel zum Flightcontroller, andererseits war die Regelung nicht Echtzeit genug. Dessenwegen habe ich im Internet nachgeschaut was für Microcontroller die meisten Quadrocopter haben. Offensichtlich werden meist MCU aus der STM32F-Familie verwendet. So kam ich auf die Webseite von Oscar Liang¹ bei der die einzelnen MCU-Generationen aufgelistet waren mit den Vor- und Nachteilen. Ich entschied mich für die neueste Generation, eine 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 ansteuerung, der schneller ist, und auch als Backup dienen sollte, so recherchierte ich was für IMU's die

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

FC's benutzen² und entschied mich für den MPU6000(vielleicht ICM-20689), da diese toleranter gegen Vibrationen sind. Als Empfänger habe ich einen nRF24L01 PA + LNA genommen, da dieser eine grosse Reichweite hat und in einem Stadtgebiet auf eine Reichweite von ca. 250-300m mit einigen Bäumen gekommen ist bei meinen früheren Messungen. Dieser übermittelt die Daten digital und hat einen CRC-Redundanzprüfung drauf, welche Fehlerhafte Übertragungen filtert bzw. korrigiert. Auch kann man in der ACK-Message Daten zurück zur Fernbedienung senden. Aber als Backup wird noch ein PPM Eingang eingebaut für die Standart Funkübertragung über eine kommerzielle 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. Das UART-Protokoll wird auch für die Kommunikation für den GPS-Empfänger und OSD(für HUD Darstellung falls eine Kamera eingebaut wird) benutzt. Um die Höhe zu bestimmen habe ich den BMP280 Drucksensor ausgewählt. Da dieser für das auslesen eine lange Zeit braucht habe ich überlegt noch eine NebenMCU einzubauen. Ich habe 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 Spannungsregler eingebaut werden, da die meisten IC's 3.3V brauchen, der Schulmechaniker Herr Thurnherr hat mir den LM3940 empfohlen. Vielleicht muss noch ein Buck-converter eingebaut werden, falls die ESC's nicht genug Leistung liefern können.

2.1.3 Leiterplatten Design und Bau

Die Leiterplatte wird mit dem Program Eagle von Autodesk designt. In dem Programm kann man die einzelnen Bauteile zusammensetzen und verbinden. (Bild) Ich musste nach dem Datenblatt der einzelnen IC's, Konsensatoren und Widerstände einbauen. Die Library's für die IC's habe ich aus dem Internet Heruntergeladen. Herr Thurnherr hat mich dabei unterstützt und alle Fragen beantwortet.

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

2.1.4 Programmieren

2.2 Die Fernbedienung

2.2.1 Konstruktion

2.2.2 Programmieren

3 Glossar

4 Quellenverzeichnis

4.1 Literaturverzeichnis

4.2 Abbildungsverzeichnis

4.3 Personenverzeichnis