

Hochschule Bremen  
Fakultät für Elektrotechnik und Informatik  
Internationaler Studiengang Mikrosystemtechnik

## Diplomarbeit

---

Entwicklung und Erprobung eines Autopiloten mit  
inertialer Messeinheit

---

Eingereicht am: 08. April 2009

Verfasst von: Andreas Dei

1. Prüfer: Prof. Dr.-Ing. Heinrich Warmers

2. Prüfer: Prof. Dr. phil. Sören Peik

## Eidesstattliche Erklärung

Hiermit erkläre ich unter Eides statt, dass ich die vorliegende Diplomarbeit zum Thema „Entwicklung und Erprobung eines Autopiloten mit inertialer Messeinheit“

selbstständig verfasst und keine weiteren als die aufgeführten Hilfsmittel verwendet habe, dass Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche gekennzeichnet sind und dass diese Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Bremen, den 08. April 2009

---

Andreas Dei

## Danksagung

An dieser Stelle möchte ich mich bei den Personen bedanken, die mich bei der Fertigstellung dieser Arbeit unterstützt haben.

Herrn Prof. Dr.-Ing. Heinrich Warmers für Aufgabenstellung, die Betreuung und die Unterstützung während der Arbeit.

Herrn Prof. Dr. phil. Sören Peik für die Übernahme der Funktion des Zweitprüfers.

Herrn Oliver Riesener für die Unterstützung während der Arbeit in Hard- und Softwarefragen.

Herrn Christoph Niemann für die Unterstützung bei Softwarefragen.

Herrn Henning Sauerland für das Korrekturlesen.

Meinen Eltern für die immerwährende Unterstützung während des Studiums und der vorherigen Schulzeit.

Da wir in unser Projektarbeit ein autonomes Fluggerät mit dem Paparazzi System realisieren konnten, welches die Urheber unter die GPL und so jedem zugänglich und frei nutzbar gemacht haben, möchte ich den Gedanken an freie Soft- und Hardware fördern und stelle diese Arbeit und die zugehörigen Programme und Hardwarebeschreibungen ebenfalls unter die GPL.

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>9</b>
<b>2</b>	<b>Aufgabenstellung</b> .....	<b>10</b>
<b>3</b>	<b>Funktionsfestlegung</b> .....	<b>11</b>
3.1	<i>Sensorplatine</i> .....	12
3.2	<i>Rechnerplatine</i> .....	13
3.2.1	Rechnerplatine 1 (LPC2148) .....	14
3.2.2	Rechnerplatine 2 (LPC2214 / LPC2294).....	15
3.2.3	Rechnerplatine 3 (TMS320F28335) .....	17
3.2.4	Auswahl Rechnerplatine .....	19
3.3	<i>Netzteil</i> .....	19
3.4	<i>Anordnung der Platinen</i> .....	19
<b>4</b>	<b>Sensorauswahl</b> .....	<b>20</b>
4.1	<i>Gyroskope</i> .....	20
4.2	<i>Beschleunigungssensoren</i> .....	21
4.3	<i>Luftdruck (statisch)</i> .....	22
4.4	<i>Luftdruck (Differenz)</i> .....	23
4.5	<i>Magnetometer</i> .....	24
4.6	<i>Temperatur</i> .....	25
<b>5</b>	<b>BauteilAuswahl</b> .....	<b>25</b>
5.1	<i>Analog-Digital-Umsetzer</i> .....	25
5.2	<i>Spannungsversorgung</i> .....	26
<b>6</b>	<b>Blockschaltbild</b> .....	<b>29</b>
<b>7</b>	<b>Schaltungsentwurf</b> .....	<b>31</b>
7.1	<i>Mikrocontroller</i> .....	31
7.2	<i>Gyroskope</i> .....	31
7.3	<i>Beschleunigungssensoren</i> .....	34
7.4	<i>Kompass</i> .....	36

---

7.5	<i>Luftdruck</i> .....	36
7.6	<i>Staudruck</i> .....	37
7.7	<i>Temperatur</i> .....	37
7.8	<i>Analog – Digital – Umsetzer</i> .....	38
7.9	<i>Spannungsversorgung</i> .....	38
7.9.1	LT1761.....	38
7.9.2	LM1117 .....	38
7.9.3	PTH08080.....	38
7.10	<i>Chipselect Decoder</i> .....	39
7.11	<i>Leistungsschalter / Kameraschalter</i> .....	39
7.12	<i>I<sup>2</sup>C Levelshifter</i> .....	39
7.13	<i>Servo/Zähler</i> .....	40
7.14	<i>Steckleisten</i> .....	40
7.14.1	UART0 X109 (Funk).....	40
7.14.2	UART1 X110 (GPS).....	41
7.14.3	I <sup>2</sup> C0 X111 .....	41
7.14.4	I <sup>2</sup> C1 X112 .....	41
7.14.5	SPI1-1 X117 .....	42
7.14.6	Analog X116.....	43
7.14.7	IR0 X107.....	44
7.14.8	RC PPM IN X113 .....	44
7.14.9	CAM X106 .....	45
7.14.10	USB X108.....	45
<b>8</b>	<b>Bauteilpositionierung</b> .....	<b>46</b>
8.1	<i>IMU</i> .....	46
8.2	<i>MCU</i> .....	48
<b>9</b>	<b>Das Routen</b> .....	<b>48</b>
<b>10</b>	<b>Prozess zum Löten von Bauteilen in BGA und QFN Gehäusen</b> .....	<b>50</b>
10.1	<i>Bestehendes</i> .....	50
10.2	<i>Parameterermittlung</i> .....	51
10.3	<i>Probelötungen von BGA Bausteinen</i> .....	54
10.4	<i>Endgültiger Lötprozess</i> .....	58
<b>11</b>	<b>Die Bestückung</b> .....	<b>62</b>

---

<b>12</b>	<b>Zusatzmodule .....</b>	<b>67</b>
12.1	<i>Motorsteller .....</i>	67
12.2	<i>Das GPS-Modul.....</i>	67
12.3	<i>Funkmodul .....</i>	68
12.4	<i>Programmieradapter .....</i>	68
<b>13</b>	<b>Inbetriebnahme.....</b>	<b>69</b>
13.1	<i>Kontrolle Versorgungsspannungen .....</i>	69
13.2	<i>Analoge Signale der Sensoren .....</i>	70
<b>14</b>	<b>Kabelpläne.....</b>	<b>72</b>
14.1	<i>K-01 MCU – IMU .....</i>	72
14.2	<i>K-02 MCU – Programmieradapter .....</i>	72
14.3	<i>K-03 MCU – Motorsteller.....</i>	73
14.4	<i>K-04 MCU – USB-Buchse .....</i>	73
14.5	<i>K-05 MCU – GPS .....</i>	73
14.6	<i>K-06 MCU – RS Empfänger .....</i>	74
14.7	<i>K-07 MCU – MCU – Radiotronix.....</i>	74
<b>15</b>	<b>Softwareanpassung .....</b>	<b>74</b>
15.1	<i>Fernsteuerung.....</i>	75
15.2	<i>Interne Analog-Digital-Converter .....</i>	76
15.3	<i>SPI Treiber.....</i>	77
15.3.1	<i>Die Funktion „spi_init“ .....</i>	77
15.3.2	<i>Die Funktion „spi_start“ .....</i>	80
15.3.3	<i>Die Funktion „spi_int“ .....</i>	80
15.3.4	<i>Die Funktion „sm_ctrl“ .....</i>	84
15.4	<i>Magnetometer .....</i>	90
15.5	<i>Umrechnungsfunktionen .....</i>	92
<b>16</b>	<b>Kalibrierung.....</b>	<b>93</b>
16.1	<i>Drehratensensoren.....</i>	93
16.2	<i>Beschleunigungssensoren .....</i>	96

---

<b>17</b>	<b>Aufbau des Quadropters.....</b>	<b>99</b>
17.1	<i>Rahmenbau .....</i>	99
<b>18</b>	<b>Erfahrungen bei der Inbetriebnahme und Integration.....</b>	<b>103</b>
18.1	<i>Layoutfehler.....</i>	103
18.1.1	Fehlerhafte Bauteile.....	103
18.1.2	Leiterbahnfehler.....	103
18.2	<i>Schaltungsfehler.....</i>	103
18.2.1	Verstärkerstufe Staudrucksensor .....	103
18.3	<i>Sonstige Fehler .....</i>	105
18.3.1	Schwingen mehrere Versorgungsspannungen .....	105
18.3.2	Einbrüche in der digitalen 5V Versorgung .....	108
18.3.3	Störungen auf dem I <sup>2</sup> C Bus.....	108
18.3.4	Motorstörungen.....	108
<b>19</b>	<b>Flugbetrieb .....</b>	<b>111</b>
19.1	<i>Inbetriebnahme für den Flugbetrieb.....</i>	111
19.2	<i>Flugbetrieb.....</i>	112
<b>20</b>	<b>Zusammenfassung .....</b>	<b>113</b>
<b>21</b>	<b>Anhang.....</b>	<b>114</b>
21.1	<i>Referenzliste.....</i>	114
21.1.1	Literaturverzeichnis .....	114
21.1.2	Abkürzungsverzeichnis.....	115
21.1.3	Abbildungsverzeichnis.....	116
21.1.4	Tabellenverzeichnis .....	119
21.2	<i>Sensorauswahl .....</i>	122
21.2.1	Gyroskope.....	122
21.2.2	Beschleunigungssensoren .....	123
21.3	<i>Schaltpläne.....</i>	125
21.3.1	MCU .....	125
21.3.2	IMU.....	127
21.3.3	Sensorplatine ADXRS610 .....	129
21.3.4	Sensorplatine LISY300A1.....	132
21.3.5	Sensorplatine MLX90609-R2.....	135
21.3.6	Servoadapter .....	138
21.3.7	Programmieradapter.....	139

---

21.3.8	GPS .....	140
21.4	<i>Layouts</i> .....	141
21.4.1	MCU .....	141
21.4.2	IMU.....	144
21.4.3	Sensorplatine ADXRS610 .....	147
21.4.4	Sensorplatine LISY300AL .....	148
21.4.5	Sensorplatine MLX90609-R2.....	149
21.4.6	Servoadapter .....	150
21.4.7	Programmieradapter.....	150
21.4.8	GPS .....	150
21.5	<i>Programmablaufpläne</i> .....	151
21.5.1	PAP für „spi_int“ .....	151
21.5.2	PAP für „sm_ctrl“ .....	152
21.6	<i>Kalibrierung</i> .....	157
21.6.1	Kalibrierkurzanleitung .....	157
21.6.1.1	Gyroskope .....	157
21.6.1.2	Beschleunigungssensoren .....	161
21.6.2	Kalibriertabellen .....	162
21.6.2.1	Gyroskope .....	162
21.6.2.2	Beschleunigungssensoren .....	162
21.6.3	Grafisch Darstellung der Kalibrierwerte.....	164
21.6.3.1	Gyroskope .....	164
21.6.3.2	Beschleunigungssensoren .....	172

# 1 Einleitung

Diese Arbeit beschreibt die Entwicklung, Aufbau und Inbetriebnahme einer miniaturisierten inertialen Messeinheit, kurz IMU (Inertial Measurement Unit), in Verbindung mit einem Autopiloten für den Einsatz auf Miniatur Schweb- und Flächenfluggeräten.

Die Aufgabe einer IMU ist es, jederzeit ihre Lage und damit auch die Lage des Fluggeräts im Raum bestimmen zu können. Um diese Aufgabe erfüllen zu können sind mehrere verschiedene Sensorsysteme erforderlich, welche eine Reihe von Informationen liefern.

Zu den für die Lageerkennung unerlässlichen Informationen gehören die Drehgeschwindigkeit um alle drei Achsen des Fluggeräts, eine Erfassung der wirkenden Beschleunigungskräfte, die unter anderem zur Berechnung der Neigungswinkel des Fluggeräts verwendet werden. Die Erfassung der Drehgeschwindigkeiten erfolgt über Gyroskope, welche inzwischen als MEMS (Mikro-Elektrisch-Mechanisches System) erhältlich sind, was den Aufbau einer miniaturisierten IMU erst möglich macht. So besitzen herkömmlich mechanische Gyroskope durchaus Abmessungen von 20x20x20cm, ein MEMS Gyroskop dagegen kommt mit 7x7x4mm aus. Auch zur Messung von Beschleunigungen sind inzwischen MEMS Sensoren erhältlich, die die Messung im miniaturisierten Maßstab ermöglichen. Eine alternative für Gyroskope und Beschleunigungssensoren stellen mehrere Infrarotsensoren dar, welche die Umgebungstemperatur rund um das Fluggerät messen. So stellt sich der Himmel als sehr kalt, der Erdboden als warm dar. Aus diesen Temperaturdifferenzen zwischen Himmel und Erdboden lässt sich die Lage berechnen. Die Lageerfassung über Infrarotsensoren birgt jedoch eine Anzahl von Nachteilen, so funktionieren sie nur außerhalb von Gebäuden und mit Abstand zu großen Flächen welche eine Wärmestrahlung aufweisen. Ebenso verfälschen diesige Luft und auch nahe Waldgebiete das Messergebnis. Dazu ist diese Art der Sensorik sehr langsam und daher nur für Flächenfluggeräte geeignet, jedoch nicht für dynamisch fliegende Schwebefluggeräte.

Ein elektronischer Kompass ermöglicht bei Schwebefluggeräten einen Ausgleich von Rotationen um die Z-Achse, die sich durch Driftfehler der Gyroskope ergeben. In der Z-Achse gibt es im Gegensatz zu der waagerechten Position, wo aus der Erdbeschleunigung berechnete Winkel als Referenz dienen, keine Referenz. Diese nötige Referenz bildet der Kompass.

Elektronische Kompass bestehen aus hochempfindlichen, auf magnetische Felder empfindliche, Sensoren, welche in allen drei Achsen das sie umgebende Magnetfeld messen. Zwar würde für die Kompassfunktion ein zweiachsiges Sensorsystem

ausreichen, jedoch wäre dieses dann neigungsempfindlich, was bei Fluggeräten zu vermeiden ist. Es erfolgt daher die Messung aller drei Achsen.

Für eine Positionsbestimmung ist ein GPS System die erste Wahl, lassen sich doch an jedem Punkt der Erde innerhalb kurzer Zeit die Koordinaten ermitteln an denen man sich befindet. Für Mikrofluggeräte ist jedoch die Genauigkeit ein Problem, denn ein GPS System hat einen relativ großen Fehler von typisch 5m bei freier Fläche und bis zu 30m in Stadtgebieten, hinzu kommt ein Auswandern der Position. Ein Schweben an einem festen Punkt ist daher schwierig. Abhilfe schafft hier ein Differenz-GPS, wo mit Hilfe einen zweiten, ortsfesten GPS Empfängers die Genauigkeit und Stabilität um ein vielfaches verbessert werden kann.

Das GPS System bietet auch die Möglichkeit Informationen über die aktuelle Höhe zu erhalten, jedoch sind diese Informationen noch ungenauer als die Positionsdaten. Der typische Fehler beträgt etwa 15m, so dass eine zuverlässige Höhenregelung, welche für autonome Flüge, zwingend erforderlich ist, kaum möglich ist. Eine genauere und schnellere Methode ist es hier den Luftdruck zu messen. Dieser sinkt mit zunehmender Höhe ab. Heutige MEMS Luftdrucksensoren besitzen eine so hohe Auflösung, dass in Verbindung mit hochauflösenden Analog-Digital-Umsetzern Höhenunterschiede von wenigen Zentimetern erfasst werden können.

Der Autopilot ist die eigentliche Intelligenz des Systems, er beinhaltet den Mikrocontroller, der die von der IMU gelieferten Daten erhält, sie verrechnet und die Motoren bzw. Steuerelemente ansteuert.

## 2 Aufgabenstellung

Aufgabenstellung Diplomarbeit Herrn Andreas Dei Matr. Nummer 146792

### **Entwicklung und Erprobung eines Autopiloten mit inertialer Messeinheit**

An der Hochschule Bremen wurden in den letzten Jahren Autopiloten aufgebaut und erprobt um Mikrofluggeräte autonom zu betreiben, die mit einem Infrarot-Lagemesssystem ausgestattet waren. Das leichteste Fluggerät hatte eine Abflugmasse von 195g, eine Spannweite von 40cm und erreichte auf einem internationalen Wettbewerb anlässlich der EMAV08 eine zweite Platzierung. Bei der Erprobung stellte sich heraus, dass es in Norddeutschland oft Wetterlagen gibt, bei denen das Lagemesssystem über Temperaturdifferenzmessungen nicht zuverlässig funktioniert. Weiter wurden Prototypen von Autopiloten für Schwebefluggeräte entwickelt, die durch Einsatz von mikroelektronischen Drehraten- und Beschleunigungssensoren eine Lagebestimmung durchführen. Diese Prototypen sind für den Einsatz in

Mikrofluggeräten zu groß und zu schwer. Herr Dei erhält im Rahmen seiner Diplomarbeit die Aufgabe einen universellen Autopiloten zu entwickeln und zu erproben, der mit MEMS- Sensoren für die Lage- und Höherfassung bestückt ist. Dieser soll sowohl für Mikroflächenflugzeuge als auch für Schwebefluggeräte geeignet sein.

Im Einzelnen sind folgenden Aufgaben zu leisten:

- Marktanalyse und technische Bewertung verfügbarer MEMS-Sensoren
- Entwicklung einer Leiterplatte für Sensoren zur Realisierung einer universellen inertialen Messeinheit mit SPI-Schnittstelle
- Erarbeitung und Dokumentation einer Technologie, um im Laborbetrieb die verwendeten Sensoren zu verlöten
- Neuentwicklung einer universellen Rechnerleiterplatte mit 32-Bit ARM7 Mikrocontroller zum Betrieb mit der neuentwickelten Sensorenleiterplatte als auch mit den bisher eingesetzten Infrarotsensoren
- Anpassung und Test von Treiberprogrammen
- Kalibrierung der Messeinheit
- Aufbau und Erprobung eines Quadropter, Integration der neuentwickelten Elektronik

Für die Durchführung dieser Arbeiten stehen Herrn Dei die folgenden Hilfsmittel zu Verfügung:

- Regelungs-, Filter- und Steuerungsprogramme eines neuentwickelten Quadropters
- Hardware und Software des Paprazzi-Projektes des WS2006 in dessen Verlauf Herr Dei einen autonome Flug mit einem Flächenflugzeug durchführte.

Die Ergebnisse der Arbeit (Programme und Dokumentation) sind unter die GPL zu stellen.

### **3 Funktionsfestlegung**

Zu Beginn der Diplomarbeit wurden mit Prof. Dr. Warmers die Anforderungen und Realisierungsmöglichkeiten festgelegt.

Hierbei wurde schnell deutlich, dass es sinnvoll ist die Sensoren mit den zugehörigen Analog-Digital Umsetzern auf eine vom Mikrokontroller getrennte Platine zu setzen. Die Sensorplatine, im folgenden nur IMU genannt, sollte mit einer Schnittstelle ausgestattet werden, die es ermöglicht die IMU auch an bereits bestehende Mikrocontrollerboards, hierbei insbesondere das Paparazzi Tiny, anzuschließen. Während sich für die Auslegung der IMU schnell eine gute Lösung fand, gab es für das Mikrocontrollerboard, im folgenden nur MCU genannt, mehrere Möglichkeiten.

### 3.1 Sensorplatine

Für den Entwurf der Sensorplatine standen eine Reihe verschiedener Sensoren und Bauteile zur Auswahl, die zu Beginn des Projekts gesammelt und nachfolgend aufgelistet werden. Aus diesen Möglichkeiten wurden später die tatsächlich eingesetzten ausgewählt.

- ADC für Drucksensoren
  - 24-Bit 4-Kanal MAX11040, zu neu?
  - 24-Bit 1-Kanal LTC2400, zu langsam?
  - 24-Bit 1-Kanal LTC2440, Funktionsprinzip geeignet?
- ADC für Gyroskope und Beschleunigungssensoren
  - MAX1168, 16Bit, 2Stück?
- Gyroskope
  - ADXRS610, geeignet?, BGA löfbar?
  - Andere Gyroskope einsetzbar?
- Beschleunigungssensoren
  - Zwei verschiedene Sensoren mit unterschiedlichen Messbereichen?
- Schnittstellen
  - SPI, analoge Signale auf Stecker, freie analoge Eingänge auf Stecker
- Versorgungsspannung
  - Für fünf Lithium Polymer Zellen (im Extremfall = 21,5V), 2 Stromversorgungsstecker vorhanden
- Temperatur
  - Wenn vorhanden über internen Sensor des Gyroskop, ansonsten externer Sensor.
- Druckmessung
  - Absolutdruck- und Staudrucksensor

- Kompass
  - Auf Sensorplatine integriert oder extern?
  - Wenn extern mit eigener Intelligenz?
- Kompatibilität
  - Sensorplatine muss an Paparazzi Tiny 13 [13] anschließbar sein.
- Platinenbreite 35mm, max 40mm, vier 3mm Befestigungsbohrungen in quadratischer Anordnung mit 30mm Abstand

Als Schnittstelle zwischen der IMU und der MCU wurde die SPI Schnittstelle gewählt. Sie ist in der Lage die geforderten Datenmengen zu transportieren. Ebenso sind die im ersten Gespräch favorisierten Analog-Digital-Umsetzern MAX1168 [D14] ebenfalls mit einer SPI Schnittstelle ausgestattet, so dass es bei Verwendung der MAX1168 die Verwendung der SPI obligatorisch wäre. Zusätzlich zu den eigentlichen Signalen der SPI Schnittstelle müssen weiterhin Leitungen für die Chip-Select-Signale vorgesehen werden. Am SPI Bus benötigt jeder angesteuerte Baustein zusätzlich zu den Datenleitungen noch ein Chip-Select-Signal um den Baustein auszuwählen. Jedes bislang verwendete Controllerboard verfügt über eine herausgeführte SPI Schnittstelle sowie über genügend Datenleitungen, so dass die geforderte Kompatibilität gegeben ist. Zusätzlich bestand der Wunsch, dass die analogen Signale der Sensoren auf Steckleisten geführt werden, so dass von einem externen Controllerboard auch die analogen Signale unabhängig von der SPI Schnittstelle ausgewertet werden könnten. Außerdem ist so ein leichter Test der analogen Signale leichter möglich.

Im Fall der Beschleunigungssensoren und der Gyroskope war es erwünschenswert, unterschiedliche Messbereiche auf allen Achsen und wenn möglich, auch auf den verschiedenen Achsen verschiedenen Messbereiche zu ermöglichen. Im Fall der Beschleunigungssensoren ist daher vermutlich eine Lösung mit 2 Sensoren vorzusehen.

Die Auswahl der Sensoren bedurfte noch einem genauen Vergleich der Eigenschaften der auf dem Markt befindlichen Sensoren was in Kapitel 4 Sensorauswahl näher erläutert wird.

## 3.2 Rechnerplatine

Die Auswahl des geeigneten Mikrocontrollers für die Rechnerplatine stellte einige Probleme dar. In ersten Überlegungen wurden drei Mikrocontrollertypen favorisiert. In die nähere Auswahl kamen dabei der bereits eingesetzte und bekannte LPC2148 von NXP, die größeren Varianten des LPC2148, LPC2214 bzw. LPC2294 ebenfalls von NXP, sowie der TMS320F28335 von Texas Instruments. Tabelle 3.1 zeigt in Kurzform die Vor- und Nachteile, eine genauere Beschreibung erfolgt in Absatz 3.2.1 - 3.2.3.

Tabelle 3.1: Vor- und Nachteile verwendbarer Mikrocontroller

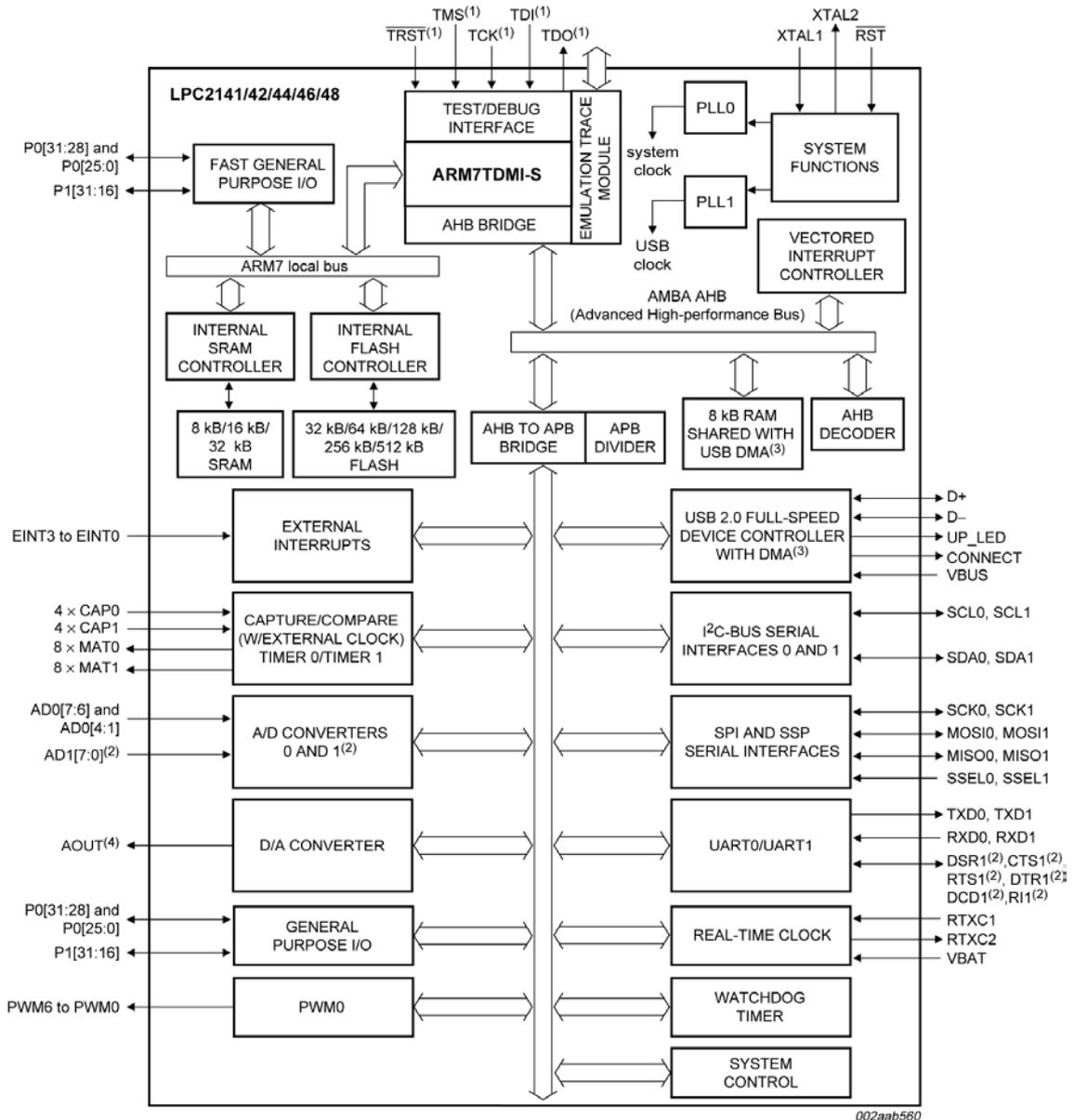
<b>Controllertyp</b>	<b>Vorteile</b>	<b>Nachteile</b>
LPC2148	Bekannte Prozessorstrukturen, lauffähige Software und viele Treiber bereits vorhanden, weit erprobt	Wenig I/O Leitungen, mäßige Rechenleistung
LPC2214 oder LPC 2294	Durch externen RAM und Flash genügend Speicher, viele I/O Leitungen	Keine Steigerung der Rechenleistung gegenüber dem LPC2148, Umstellung auf neuen Controllertyp
TMS320F28335	Sehr viel Rechenleistung, genügend I/O Leitungen	Teurer Compiler, neuer Prozessortyp, daher Einarbeitung in neue Umgebung erforderlich → Zeitaufwändig, aufwändige Leiterplatte

Für alle Rechnerplatinen bestand die Forderung nach bestimmten Komponenten, welche folgend aufgeführt sind.

- 10 Servoausgänge
- RC Eingang
- Freie ADC auf Steckerleisten geführt
- 2x UART
- 2x PC
- 2x SPI
- Möglichkeit zum Anschluss eines SD-Kartenlesers
- Einige GPIO auf Stecker
- Eventuell USB Master
- Platinenbreite 35mm, max. 40mm

### 3.2.1 Rechnerplatine 1 (LPC2148)

Der bereits bekannte und oft eingesetzte LPC2148 stellt die konservative Lösung da. Über ihn gibt es ausreichend Erfahrungen und bestehende Software. Das größte Problem was gegen ihn sprach waren die zu wenigen I/O Leitungen, die schon in vergangenen Projekten zu Problemen führten. Außerdem reicht die Rechenleistung noch für aktuelle Anwendungen aus, jedoch besteht kaum Reserve für zukünftige, rechenintensivere Erweiterungen. Bild 3.1 zeigt das Blockschaltbild des LPC2148.



(1) Pins shared with GPIO.

(2) LPC2144/46/48 only.

(3) USB DMA controller with 8 kB of RAM accessible as general purpose RAM and/or DMA is available in LPC2146/48 only.

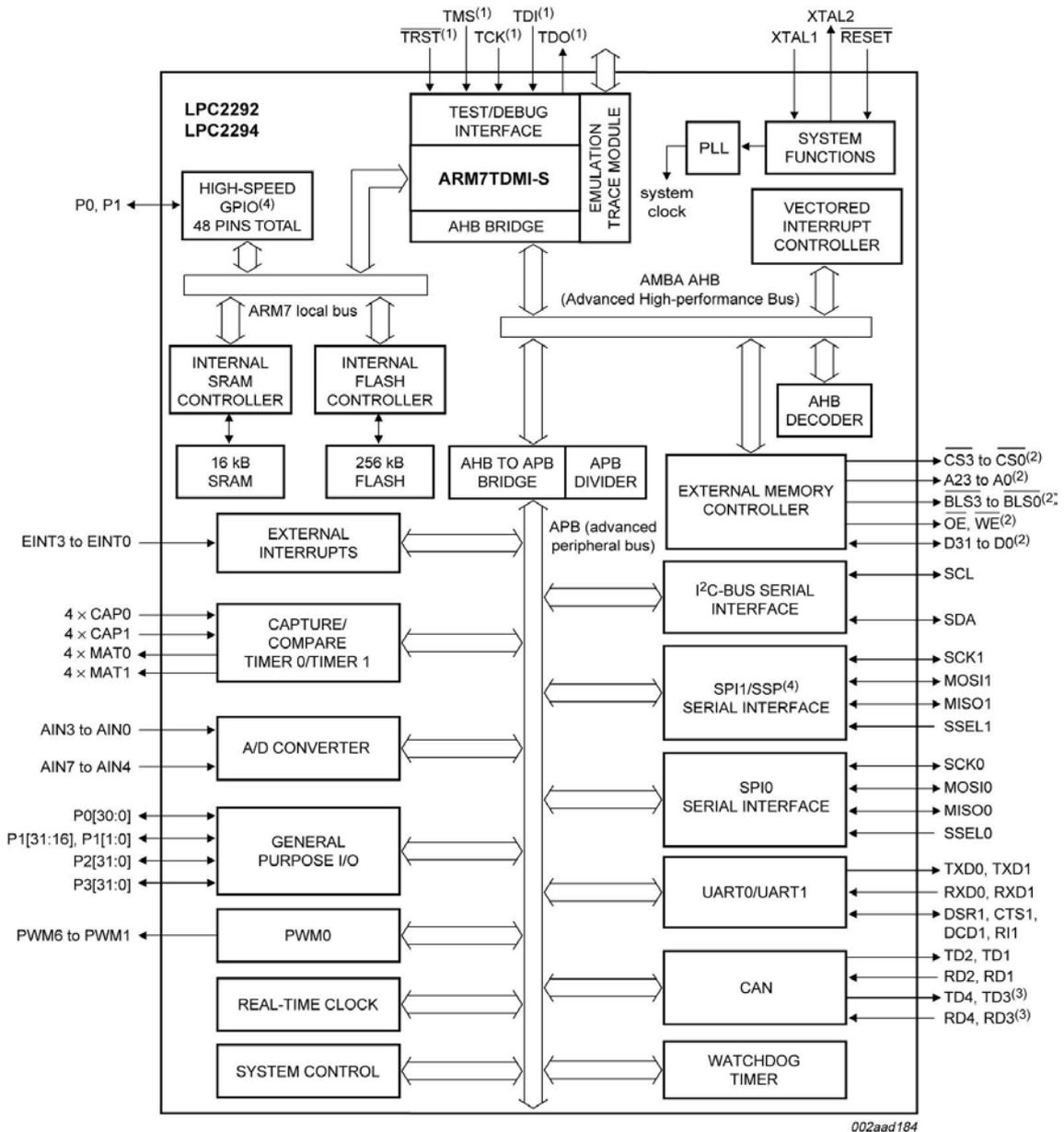
(4) LPC2142/44/46/48 only.

Bild 3.1: Blockschaltbild LPC2148 [D7]

### 3.2.2 Rechnerplatine 2 (LPC2214 / LPC2294)

Der LPC2214/LPC2294 ist ein dem LPC2148 ähnlicher Prozessor, er besitzt jedoch keinen internen RAM und Flash Speicher. RAM und Flash muss in externen Bausteinen angegliedert werden. Somit ergibt sich der Vorteil von mehr als ausreichendem Speicherplatz. Ebenso stehen durch mehr Pins des LPC2214 mehr I/O Leitungen zur Verfügung, was wiederum dem Mangel an I/O Leitungen entgegen wirkt. Der LPC2214 und LPC2294 sind zueinander sehr ähnlich, als Hauptunterschied besitzt der LPC2294

einen integrierten CAN Controller. Problematisch ist hier jedoch auch der Umstieg auf einen neuen, wenn auch ähnlichen Prozessortyp, der Probleme in der Zeitplanung der Arbeit verursachen könnte. So bestehen keinerlei Erfahrungen über externe Speicher. Der Umstieg auf diesen Prozessor wäre somit auch mit einem Risiko der Nichtfunktion behaftet. Bild 3.2 zeigt das Blockschaltbild des LPC2294



- (1) When test/debug interface is used, GPIO/other functions sharing these pins are not available.
- (2) Pins shared with GPIO.
- (3) Available in LPC2294 only.
- (4) SSP interface and high-speed GPIO are available on LPC2292/2294/01 only.

Bild 3.2: Blockschaltdiagramm LPC2294 [D9]

### 3.2.3 Rechnerplatine 3 (TMS320F28335)

Der TMS320F28335, ein Digital Signal Controller mit Gleitkommaeinheit, würde mehr als genug Rechenleistung auch für zukünftige Projekte bieten. Ebenso verfügt er über eine Vielzahl von bereits integrierten Schnittstellen und ausreichend I/O Pins, was das bisher beim LPC2148 bestehende Problem von zu wenigen I/O Leitungen lösen würde. Als kritisch wurde jedoch angesehen, dass es sich um einen völlig unbekanntem Prozessortyp handelte bei dem keinerlei Erfahrung vorhanden war. Zudem gibt es keine freie Entwicklungsumgebung, der Compiler muss gekauft werden. Der Punkt, dass es sich um einen neuen, unbekanntem Prozessor handelt, schien es als unmöglich zu erscheinen, im Zeitrahmen einer Diplomarbeit das Projekt erfolgreich abzuschließen. Bild 3.3 zeigt das Blockschaltbild des TMS320F28335.

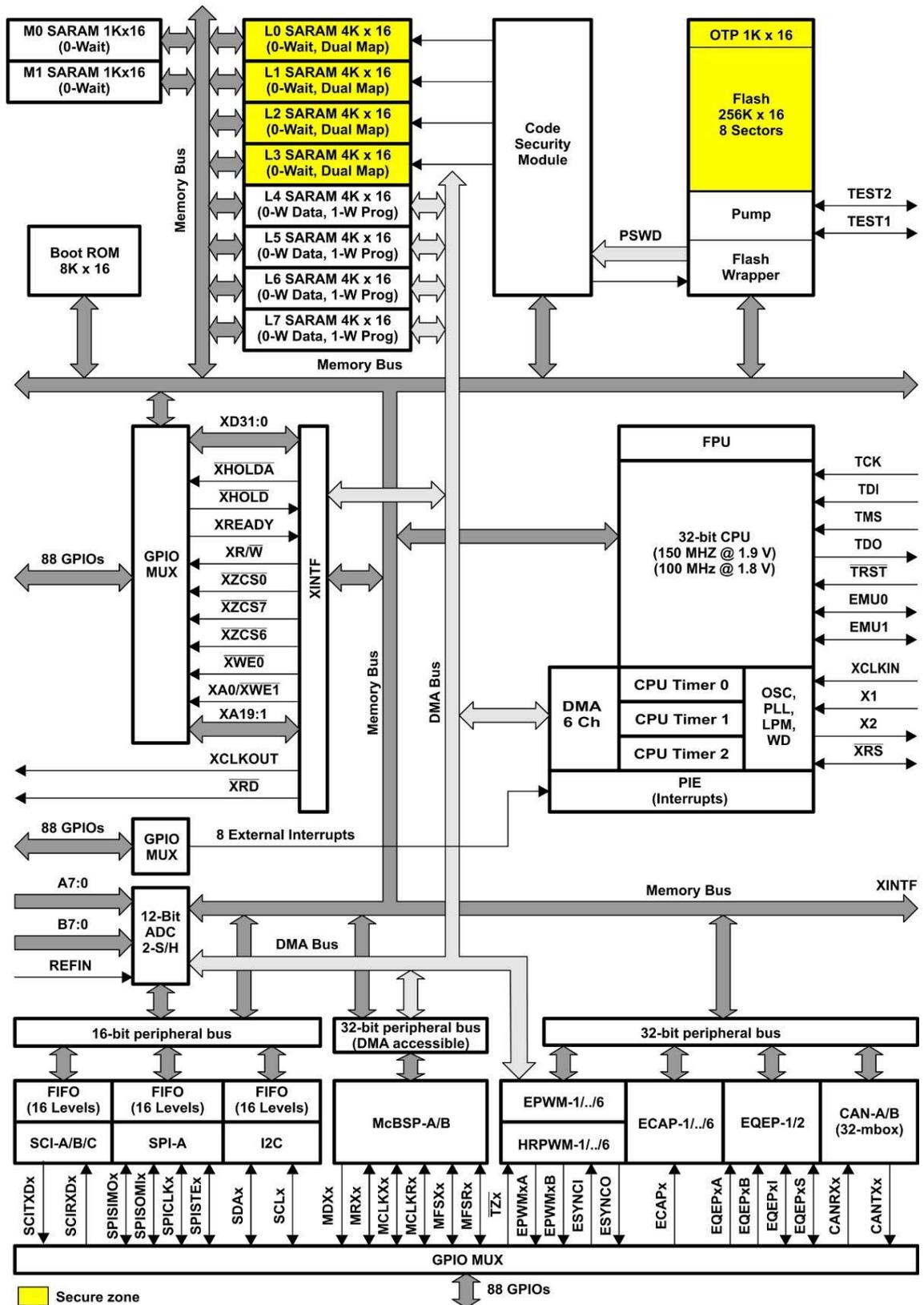


Bild 3.3: Blockschaltbild TMS320F28335 [D10]

### 3.2.4 Auswahl Rechnerplatine

Die Wahl fiel schließlich auf die konservative Variante mit dem LPC2148, da hier die größten Chancen gesehen wurden, in der gegebenen Zeit ein flugfähiges System zu konstruieren. Das bestehende Problem der zu wenigen I/O-Leitungen wurde zuvor durch eine Abschätzung der benötigten und vorhandenen Leitungen überprüft um sicherzustellen, dass genügend I/O Leitungen zur Verfügung stehen um das Projekt zu realisieren. Das Ergebnis der Abschätzung zeigte, dass die Zahl ausreichen sollte. Die fehlende Rechenleistung sollte durch die Möglichkeit ausgeglichen werden über die zweite SPI Schnittstelle ein fertiges Rechnermodul, z.B. ein ARM9 Board, anschließen zu können, das rechenintensive Aufgaben übernimmt und mit dem LPC2148 kommunizieren kann.

### 3.3 Netzteil

In der Funktionsfestlegung wurde ebenso festgelegt, dass das 5V Netzteil, welches mit dem Schaltregler PTH08080 [D15] aufgebaut sein sollte, zusammen mit dem Zählerbaustein zu Servosignaldekodierung so auf der MCU Platine anzuordnen, das es auch von dieser getrennt betrieben werden kann, falls sich durch den Schaltregler Störungen in der Sensorik ergeben sollten. Im Falle von Störungen kann somit der Schaltregler weiter entfernt von der Sensorik montiert werden. Die Möglichkeit zu Trennung sollte so konzipiert werden, dass beide Leiterplatten zusammen als eine hergestellt werden können und auch eine elektrische Verbindung zwischen ihnen besteht. Beide Leiterplatten sollen sich durch einen einfach Schnitt trennen lassen, die elektrischen Verbindungen sollen dann über zwei Stecker per Kabel herstellbar sein

### 3.4 Anordnung der Platinen

Durch die Verwendung von zwei Leitplatten für IMU und MCU stellte sich die Frage nach einer sinnvollen Anordnung dieser Leiterplatten. Nach ersten Überlegungen beide Leiterplatten hintereinander zu reihen, was zu einer langen, schmalen Leiterplatte geführt hätte, wurde die Variante, beide Leiterplatten zu stapeln, bevorzugt, da diese Variante kompakter ist.

Erste Überlegungen, die MCU unten und die IMU oben anzuordnen, wurden schnell verworfen, da sich eine ungünstige Kabelführung ergeben würde. Alle Kabel müssten seitlich an die MCU angeschlossen werden, was den praktischen Platzbedarf in der Breite erheblich vergrößert hätte. Die MCU oben zu platzieren würde den Vorteil bringen, dass alle Kabel auf der Oberseite von oben gesteckt werden könnten. Die IMU besitzt keine für den normalen Flugbetrieb nötigen Steckverbinder, somit ist es nicht nachteilig sie unten zu positionieren. Außerdem kann der zwangsläufig bei einem Stapel durch die Steckverbinder entstehende Zwischenraum durch die ebenfalls hohen senkrechten Platinen genutzt werden, so dass der Raum optimal ausgenutzt werden kann und sich keine großen Zwischenräume bilden.

## 4 Sensorauswahl

### 4.1 Gyroskope

Für die Auswahl der Gyroskope wurden zunächst per Internetrecherche auf dem Markt erhältliche MEMS Gyroskope erfasst.

Folgende Gyroskope sind derzeit auf dem Markt erhältlich (Typ, Hersteller):

- IDG-300 [D16], InvenSens
- ENC-03J [D17], Murata
- ADXRS150/300 [D18/19], Analog Devices
- ADXRS61X Series [D4], Analog Devices
- LISY300AL [D20], ST Microelectronics
- MLX90609-R2 [D21], Melexis

Um die Eigenschaften miteinander vergleichen zu können, wurden die wichtigsten Kenndaten in Tabelle 21.1 (siehe Anhang) eingetragen. Besonders zu beachten sind hierbei der Temperaturdrift, die Nichtlinearität, sowie die Rauschdichte, da diese bei zu schlechten Werten die Funktion der IMU gegebenenfalls beeinträchtigen oder verhindern.

Die Auswertung der Tabelle 21.1 kommt zu folgenden Ergebnissen für die einzelnen Gyroskope:

- IDG-300, InvenSens
  - Vorteile: Günstig beschaffbar; 2-Achsig, zusammen mit einem Z-Gyro kann eine IMU ohne senkrecht aufgestellte Platinen realisiert werden; geringes Rauschen.
  - Nachteile: Sehr hoher Temperaturdrift, große Nichtlinearität.
- ENC-03J, Murata
  - Vorteile : leicht beschaffbar, viele Erfahrungen in der Verwendung für IMU's vorhanden
  - Nachteile: große Bauform, ungünstige Achsenanordnung, schlagempfindlich, sehr hoher Temperaturdrift
- ADXRS150/300, Analog Devices
  - Vorteile : Sehr gute Werte
  - Nachteile: hoher Preis; schwer zu beschaffen; ADXRS300 ist bereits abgekündigt; BGA Gehäuse

- ADXRS61X Series, Analog Devices
  - Vorteile : Sehr gute Werte, geringster Temperaturdrift und geringste Nichtlinearität aller untersuchten Sensoren; RatioMetric Eingang, ermöglicht Anpassung der Ausgangsspannung an die Referenzspannung des ADC; in gleicher Bauart mit unterschiedlichen Messbereichen erhältlich
  - Nachteile: hoher Preis; schwer zu beschaffen; BGA Gehäuse
- LISY300AL, ST Microelectronics
  - Vorteile: günstig und gut beschaffbar
  - Nachteile: geringe Oszillatorfrequenz, die in anderen Projekten schon zu Problemen durch Beeinflussen durch Motorvibrationen geführt hat [4].
- MLX90609-R2, Melexis
  - Vorteile: Preiswert, von Hand verlötbar
  - Nachteile: sehr große Bauform

Die Wahl viel schließlich trotz des hohen Preises und der schlechten Beschaffbarkeit auf die ADXRS61X Serie. Begründet liegt dies in den sehr guten Werten, sowie in guten Erfahrungen, die von anderen Projektgruppen mitgeteilt wurden [2, 5]. Für den Prototypen konnte das Beschaffungsproblem dadurch verhindert werden, dass gesamt acht Sample des ADXRS610 von Analog Devices erhalten werden konnten. Aufgrund des BGA Gehäuses ist jedoch ein Prozess zu entwickeln, der es ermöglicht diese Sensoren mit Hochschulmitteln zu verarbeiten.

Jedoch wurde während der Auswahl der Gyroskope beschlossen, eine Möglichkeit zu schaffen, verschiedene Typen nutzen zu können. Dadurch, dass zwei der Sensoren auf senkrecht gestellte Platinen montiert werden müssen, kann hier ein Austausch durch den Tausch der gesamten senkrechten Platinen erfolgen. Für den Z-Gyro, der normalerweise direkt auf die Hauptplatine bestückt werden würde, wurde in diesem Zug festgelegt, ihn auf eine kleinen Platine zu montieren, die wiederum auf die Hauptplatine bestückt wird. Als Ersatz sollen daher ebenso die LISY300AL und die MLX90609-R2 verwendbar sein.

## 4.2 Beschleunigungssensoren

Die Auswahl der Beschleunigungssensoren erfolgte auf gleiche Art und Weise wie die bei den Gyroskopen.

Von bekannten Herstellern wurden Sensoren herausgesucht und die wichtigsten Daten in Tabelle 21.2 und Tabelle 21.3 (siehe Anhang) zusammengefasst. Besonderes Augenmerk bei dem Vergleich lag hier bei dem Temperaturdrift, Nullpunktdrift,

Nichtlinearität und Achsenübersprechen. Zur besseren Übersicht und Einschätzung der Werte wurde eine farbliche Kennzeichnung vorgenommen, die einen Vergleich unter allen Sensoren darstellt. So wurden die besten Werte grün, die schlechtesten Werte rot eingefärbt. Werte im Zwischenbereich wurden gelb und orange eingefärbt.

Die Auswertung ergab, dass die Sensoren von Analog Devices den besten Gesamteindruck vermittelten. Die Sensoren von ST Microelectronics vermittelten insgesamt auch kein schlechtes Bild und hätten bei einigen Bausteinen eine einfache Umschaltung des Messbereiches von 2 oder 6g ermöglicht. Insgesamt überzeugten die Analog Devices Sensoren jedoch mehr. Hinzu kam noch das in einem Probeaufbau einer IMU innerhalb des Labors der verwendete ST Sensor einen massiven Empfindlichkeitsfehler zeigte.

Die genauere Auswahl fiel schließlich auf eine Kombination des ADXL330 [D22], ADXL320 [D23] und ADXL323 [D24]. Je nach Kombination können die Achsen somit auch in unterschiedlichen Messbereichen vermessen werden. Die Art der Kombination wurde im Design ermittelt.

### 4.3 Luftdruck (statisch)

Die Vorgaben für den statischen Luftdrucksensor waren durch einen Messbereich von 0 – 115kPa gegeben. Zusätzlich sollte aus Platzspargründen das kleinste erhältliche Gehäuse zum Einsatz kommen.

Zur Auswahl standen hier einige Sensoren der Firma Freescale Semiconductors.

- MPX4100 Series
- MPX4115 Series
- MPXA / MPXH 6115 Series

Die elektrischen Daten der Sensoren unterscheiden sich untereinander nicht oder nur minimal, so dass hier keine Unterscheidung nötig ist. Einzig verfügen die Sensoren der 4xxx Serie über einen Spannungsausgang, während die Sensoren der 6xxx Serie über einen Stromausgang verfügen, aus dem erst mittels Widerstand eine Spannung erzeugt werden muss.

Bei den Sensoren der 4115 Serie handelt es sich ausschließlich um konventionelle bedrahtete Gehäuse mit großen mechanischen Abmessungen, daher scheidet diese Serie aus.

Die 4100 Serie ist in bedrahteten wie auch in SMD Gehäusen der Bauform SOP (Small outline Package) erhältlich.

Sensoren der 6115 Serie mit dem Buchstabenkürzel MPXA sind in einem SOP Gehäuse untergebracht während Sensoren mit dem Kürzel MPXH in einem SSOP (Super Small Outline Package) untergebracht sind. Zusätzlich sind diese Sensoren in einer Version

für erhöhte Luftfeuchtigkeitsanforderungen erhältlich, die durch den Zusatz Z an der Buchstabenkombination gekennzeichnet sind.

Da die MPXH6115 [D25] Sensoren im kleinsten Gehäuse lieferbar sind wurden diese ausgewählt. Für besondere klimatische Verhältnisse kann auch der MPXHZ6115 eingesetzt werden. Zusätzlich sind zwei unterschiedliche Gehäuseformen erhältlich, die sich durch einen Stutzen zum Schlauchanschluss (MPXH6115AC6U/C6T1) und planer Ausführung ohne Stutzen MPXH6115A6U/6T1 unterscheiden.

#### 4.4 Luftdruck (Differenz)

Zur Fahrtmessung an Flächenfluggeräten sollte ein Differenzdrucksensor zur Auswertung eines Staudruckrohrs vorgesehen werden, der so ausgelegt sein sollte um Geschwindigkeiten bis ca. 20m/s messen zu können.

Hierzu war zuerst zu ermitteln, welche Drücke bei entsprechender Geschwindigkeit auftreten um den Messbereich entsprechend auszuwählen. Über umgewandelte Form der Bernoulli-Gleichung (4.1) [6] ist dies möglich.

$$p = \frac{1}{2} * \rho * v^2 \quad (4.1)$$

Die Dichte von Luft wird mit  $1,27\text{kg/m}^3$  angenommen, somit ergibt sich bei einer maximalen Geschwindigkeit von  $20\text{m/s}$  eingesetzt nach (4.2) ein maximaler Druck von  $2,58\text{mbar}$

$$P = \frac{1}{2} * 1,27 \frac{\text{kg}}{\text{m}^3} * 20^2 \frac{\text{m}^2}{\text{s}^2} = 258\text{Pa} = 2,58\text{mbar} \quad (4.2)$$

Eine Recherche des Angebots der Differenzdrucksensoren von Freescale ergab, dass der MPXV7002 [D26] mit 2kPa entsprechend 20mbar, den kleinsten Messbereich der angebotenen Sensoren besitzt. Dies beträgt noch immer das 10-fache des zu erwartenden Anwendungsmaximums. Eine Alternative bestand im Sensor HCLA02X5EU von Sensortronics mit einem Messbereich von 2,5mbar. Dieser Sensor hatte jedoch einen sehr hohen Preis, so dass gegen ihn entschieden wurde und der Messbereich des MPXV7002 durch nachfolgende Elektronik angepasst werden sollte.

In der Projektarbeit [7] wurde bereits die Fahrtmessung per Staudruckrohr und Differenzdrucksensor untersucht. Dort wurde als Sensor ein MPX2010 mit 10kPa Messbereich verwendet. Hier wurde ebenfalls auf den in diesem Fall noch

ungünstigeren Messbereich hingewiesen. Durch die nachfolgende Verstärkerschaltung konnte das gewünschte Signal aber einwandfrei erzeugt werden. In diesem Projekt wurden ebenfalls Versuche zum erwartenden Druck durchgeführt, die im Bereich der zur Dimensionierung verwendeten Berechnungen liegen.

## 4.5 Magnetometer

Als Magnetsensoren standen zum einen der MicroMag3 als fertiges Modul zur Verfügung wie auch eine Schaltungsvariante mit drei KMZ51 Hall-Sensoren [D11], entwickelt von Holger Buss [8], basierend auf einem Applikationsbereich von NXP [D12]

Der MicroMag3 ist ein fertiges Modul [D13], welches über die SPI Schnittstelle angesprochen wird. Hier besteht jedoch der Nachteil, dass er ein sehr langsamer Baustein ist, der in seiner Messzeit den SPI Bus nicht frei gibt, so dass keine sichere Datenerfassung von den flugwichtigen Sensoren mehr möglich ist. Hier muss daher überlegt werden den MicroMag3 an einen anderen SPI BUS als die Sensoren anzuschließen oder den MicroMag3 mit einem eigenen kleinen Mikrokontroller zu versehen, der den MicroMag3 über den SPI ansteuert und die Daten per I<sup>2</sup>C an die MCU weitergibt. Die Montage des MicroMag3 direkt auf der IMU Platine scheidet aufgrund seines großen Bauvolumens aus, so dass er auf jeden Fall als externes Modul vorzusehen ist.

Die Variante mit den KMZ51 Sensoren ließe sich direkt mit auf die IMU integrieren. Jedoch besteht hier die Gefahr, dass Magnetfelder oder magnetische Komponenten die Sensoren beeinflussen. Um dies zu überprüfen wurden im Vorfeld Untersuchungen mit einem fertigen Modul nach [8] durchgeführt. Hierzu wurden die analogen Ausgangssignale der Sensoren gemessen und die Auswirkungen der verschiedenen Bauteile untersucht. Als kritische Bauteile wurden vor allem die Luftdrucksensoren (wegen einer Edelstahlplatte im Deckel) und der Schaltregler angesehen. Festgestellt werden konnte, dass keiner von beiden Luftdrucksensoren eine Auswirkung auf die Magnetsensoren hatte, der Schaltregler war jedoch in der Lage bei großer Stromentnahme in unmittelbarer Nähe die Sensoren geringfügig zu beeinflussen. Da jedoch der Regler ein vielfaches von der Entfernung, in der er Störungen verursacht, entfernt ist, ist eine Verfälschung der Messergebnisse unwahrscheinlich.

Aufgrund der höheren Kompaktheit mit weniger externen Modulen und entsprechendem Kabelaufwand erschien die Variante mit den KMZ51 direkt auf der IMU als sinnvoller. Falls es jedoch wider erwarten zu Störungen kommen sollte, die eine Auswertung der Magnetfeldlinien verhindern, sollte es möglich sein, sowohl den MM3 als auch ein Modul mit den KMZ51 extern anzuschließen. Hierzu muss somit die I<sup>2</sup>C und die zweite SPI Schnittstelle von außen nutzbar sein, sowie ein I/O Pin für Steuerzwecke zur Verfügung stehen. Dieses wurde der Planung der Steckerleisten für die Schnittstellen mitbedacht.

## 4.6 Temperatur

Für den Temperatursensor war ein Mindestmessbereich von  $-30^{\circ}\text{C}$  -  $+60^{\circ}\text{C}$  bei einer Genauigkeit von  $< 1\%$  bei  $25^{\circ}\text{C}$  vorgegeben. Da der Temperatursensor eine späte Erweiterung darstellte, die erst nach Beendigung des Platinenlayouts geplant wurde, kam nur ein analoger Sensor in einem möglichst kleinen Gehäuse in Frage, da kaum Platz mehr zur Verfügung stand und ebenso der Platz für einen digitalen Bus nicht gegeben war.

Die Auswahl eines geeigneten Temperatursensors erfolgte über die Digikey [9] Homepage und deren Produktkatalog nach folgender Strategie:

Produktkatalog → Sensoren, Transducer → Temperatur → Filter: Analog, Artikel auf Lager.

Eine weitere Selektion erfolgte über das Gehäuse. Als Gehäusebauformen, die klein genug waren um doch auf der Platine positioniert werden zu können, zeigten sich 4-MicroSMD, SOT23-5, SOT23-6 sowie SC70-5.

Im 4-MicroSMD war der LM20 erhältlich, welcher aufgrund seiner nicht ausreichenden Genauigkeit von 1,5% ausschied, zudem ist das 4-MicroSMD Gehäuse schwer zu verarbeiten.

Der im SOT23-5 erhältliche MAX6608 schied durch die nicht ausreichende Genauigkeit von 2% aus.

Der im SOT23-6 erhältliche MAX6611 ist nur in großen Stückzahlen lieferbar.

Von den unter SC70-5 aufgelisteten Bauteile erreichte schließlich nur der MAX6605 [D6] die geforderte Genauigkeit von unter 1%. Erste Wahl wäre somit der MAX6605, da dieser jedoch recht teuer ist, wurde versucht eine preisgünstige Alternative zu ermitteln. Hier fiel dann die Wahl auf den LM20 [D27], der bei bestimmtem Layout, als Pinkompatibel angesehen werden kann.

## 5 BauteilAuswahl

### 5.1 Analog-Digital-Umsetzer

Als Analog-Digital-Umsetzer für die Signale der Beschleunigungs- und Drehratensensoren, sowie die Signale der anderen Sensoren werden hochauflösende Typen mit einer möglichst hohen Kanalanzahl benötigt. Die Wahl fiel hierbei auf den MAX1168 [D14] von Maxim. Dieser ADC besitzt eine Auflösung von 16-Bit, 8-Kanäle und ist durch sein Umsetzverfahren der sukzessiven Approximation schnell, das Messergebnis steht sofort im Anschluss der Messanfrage zur Verfügung. Er wurde

bereits für vorausgegangene Projekte erfolgreich verwendet [1], ebenso existieren für ihn bereits die nötigen Softwaretreiber.

Zur Auswertung des Absolutluftdrucks wird eine höhere Auflösung als 16-Bit benötigt, um auch Höhenänderungen im cm-Bereich erfassen zu können. Der in früheren Projekten verwendete einkanalige 24Bit Umsetzer LTC2400 [D2] von Linear Technology erwies sich hierbei als sehr langsam. Als schnellerer 24Bit Umsetzer ist ebenfalls bei Linear Technology der LTC2440 [D1] erhältlich. Im Gegensatz zum LTC2400, der mit einer festen Genauigkeit und damit festen Konvertierungsgeschwindigkeit arbeitet, kann der LTC2440 durch ein per SPI gesendetes Kommandowort auf verschiedene Genauigkeiten und damit Konvertierungsgeschwindigkeiten umgestellt werden. Die erreichbare Konvertierungsgeschwindigkeit des LTC2440 ist abhängig von der Genauigkeit mit der die Konvertierung erfolgen soll. Je weniger Zeit für einen Konvertierungsvorgang zur Verfügung steht, je geringer ist die Genauigkeit. Die Geschwindigkeit ist jedoch im Vergleich mit dem LTC2400 bei gleicher Genauigkeit von 24Bit erheblich höher (~37ms LTC2440 [D1, S.14], ~133,3ms LTC2400 [D2. S.4]).

## 5.2 Spannungsversorgung

Als Vorgabe für die Spannungsversorgung war festgelegt, dass beide Platinen mit einer einzigen Spannung versorgbar sind. Dabei sollte die Eingangsspannung möglichst hoch sein, im Idealfall geeignet für fünf Lithium Polymer Zellen. Die Nennspannung von Lithium-Polymer-Zellen beträgt 3,7V, kann jedoch bis 4,3V ansteigen. Auf 5 Zellen ergibt dies eine Nennspannung von 18,5V bzw. 21,5V [10]. Es sollte somit eine maximale Eingangsspannung von 23V möglich sein.

Beide Platinen benötigen zum Betrieb eine analoge und digitale Spannungsversorgung mit je 3,3V und 5V.

Zur Abschätzung des ungefähren Strombedarfs wurden die Verbraucher der IMU in Tabelle 5.1 gelistet.

Tabelle 5.1: Stromaufnahme der IMU Komponenten

Verbraucher	Digital 3,3V	Digital 5V	Analog 3,3V	Analog 5V
ADXRS610	-	3x 5mA	-	3x 3mA
ADXL320	-	-	0,48mA	-
ADXL330	-	-	0,32mA	-
MPXH6115	-	-	-	10mA
MPXV7002	-	-	-	10mA
TLC2262	-	-	-	0,5mA
LTC2440	-	11mA	-	1mA
MAX1168	1,5mA	-	-	4mA
TLC2264	-	-	-	0,5mA
KMZ51	-	ca. 10mA	-	ca. 10mA
74HC138	-	0,08mA	-	-
MAX6605	-	-	0,01mA	-
	1,5mA	36,08mA	0,81mA	45mA

Als Spannungsregler für die analogen Spannungen sollten spezielle rauscharme Spannungsregler zum Einsatz kommen. Die Suche nach geeigneten Reglern stellte sich insofern als schwierig heraus, da alle für solche Anwendungen in Frage kommenden Spannungsregler nicht den geforderten Eingangsspannungsbereich besitzen. In der Regel sind diese Regler nur bis 16V, in seltenen Fällen bis 20V, konzipiert. Ausführliche Recherche bei diversen Anbietern ergab, dass der LT1761 [D3] von Linear Technology der Spannungsregler mit dem größten Eingangsspannungsbereich ist. Dieser Regler ist für maximal 20V konzipiert und damit aber auch unter der geforderten Spannung. Jedoch sind keine anderen Regler auf dem Markt. Aus diesem Grund muss die zur Versorgung verwendete LiPo Zellenzahl von fünf auf vier gesenkt werden. Sollen mehr als vier Zellen verwendet werden ist ein Vorsatz vorzuschalten, der die Spannung reduziert. Die maximale Stromentnahme des LT1761 beträgt 100mA was für die Anwendung ausreichend ist. Kritisch ist jedoch die auftretende Verlustleistung bei der höchsten anzunehmenden Eingangsspannung, sie beträgt nach 553,5mW.

$$I_{out(max)} * (V_{in(max)} - V_{out}) + I_{GND} * V_{in(max)} \quad (5.3)$$

IGND wird aus einem Diagramm [D3, S.9] bestimmt, dieses ist jedoch nur bis Eingangsspannungen von 10V gültig, so dass IGND für 17,3V auf etwa 1mA abgeschätzt werden musste. Somit ergibt sich den mit 45mA am stärksten belasteten Analog 5V Regler nach (5.4) eine Verlustleistung von 566,2mW.

$$P_v = 45mA * (17,2V - 5V) + 1mA * 17,2V = 566,2mW \quad (5.4)$$

Das Datenblatt gibt eine maximale Sperrschichttemperatur von 125°C an, die laut Datenblatt nach (5.5)(5.6) näherungsweise berechnet werden kann.

$$T_J = P_v * R_{th} \quad (5.5)$$

$$T_J = 566,2mW * 150 \frac{^{\circ}C}{W} = 84,9^{\circ}C \quad (5.6)$$

Die maximale Sperrschichttemperatur berechnet sich aus der durch die Verlustleistung erzeugten Temperatur + Umgebungstemperatur.

$$T_{J_{max}} = T_v + T_{Umgebung} \quad (5.7)$$

Somit ergibt sich nach (5.8) eine maximale Umgebungstemperatur von 40,1°C, über der die Imu nicht mit der maximalen Eingangsspannung betrieben werden darf.

$$T_{Umgebung} = T_{J_{max}} - T_v = 125^{\circ}C - 84,9^{\circ}C = 40,1^{\circ}C \quad (5.8)$$

Als Regler für die digitalen Spannungen sind keine besonders rauscharmen Regler notwendig, jedoch ergab sich hier wiederum das Problem, dass die im Handel verfügbaren Regler entsprechender Baugröße keine ausreichende Spannungsfestigkeit besaßen bzw. teurer waren als die LT1761. Daher wurden für die digitalen Versorgungen ebenfalls die gleichen Regler verwendet.

Die Stromaufnahme der MCU wird hauptsächlich durch die angeschlossene Peripherie in Form von Servos, Kamera, Funkmodul und ähnlichem bestimmt. Die Stromaufnahme der Servos wurde mit etwa 2A angenommen, somit wurde für den digitalen 5V Zweig der MCU das fertige Schaltreglermodul PTH08080 [D15] verwendet, welches einen maximalen Ausgangsstrom von 2A liefern kann. Mit einer maximalen

Eingangsspannung in Höhe von 18V ist er noch mit vier Lithium-Polymer-Zellen verwendbar. Die digitale 3,3V Versorgung kann zur Reduzierung der Verlustleistung per LDO aus den 5V gewonnen werden. Die nötige Belastbarkeit wird hauptsächlich von Mikrocontroller sowie angeschlossener Peripherie bestimmt. Um genügend Reserven für Peripheriebaugruppen zu besitzen wurde die 3,3V Versorgung großzügig dimensioniert und mit einem LM1117-3,3 [D28] ausgestattet, der in der Lage ist bis zu 800mA zu liefern. Um das Funkmodul vor Störungen andere Baugruppen zu schützen wurde es mit einem eigenen Spannungsregler, ebenfalls vom Typ LM1117-3,3 ausgestattet. Durch die großzügige Dimensionierung können die anderen Bauteile in der Auslegung unberücksichtigt bleiben, da sie nur einen geringen Strombedarf besitzen. Die beiden analogen Versorgungsspannungen dienen nur zu Versorgung der Mikrocontroller internen ADC sowie eventueller Peripherie, woraus nur ein geringer Strombedarf in Höhe einiger mA resultiert. Es wurden daher die gleichen Regler wie auf der IMU eingesetzt, da diese mit 100mA Belastbarkeit bereits die kleinsten lieferbaren Regler waren.

## 6 Blockschaltbild

Bild 6.1 zeigt das Blockschaltbild des Autopiloten mit seinen wichtigsten Komponenten. Auf der Seite links der grauen Strichpunktlinie sind die Komponenten der IMU, wie Sensoren und Analog-Digital-Umsetzer dargestellt.

Rechts der grauen Strichpunktlinie befinden sich die Komponenten der MCU. Den Kern bildet hier der Mikroprozessor, umgeben von den Peripheriebausteinen.

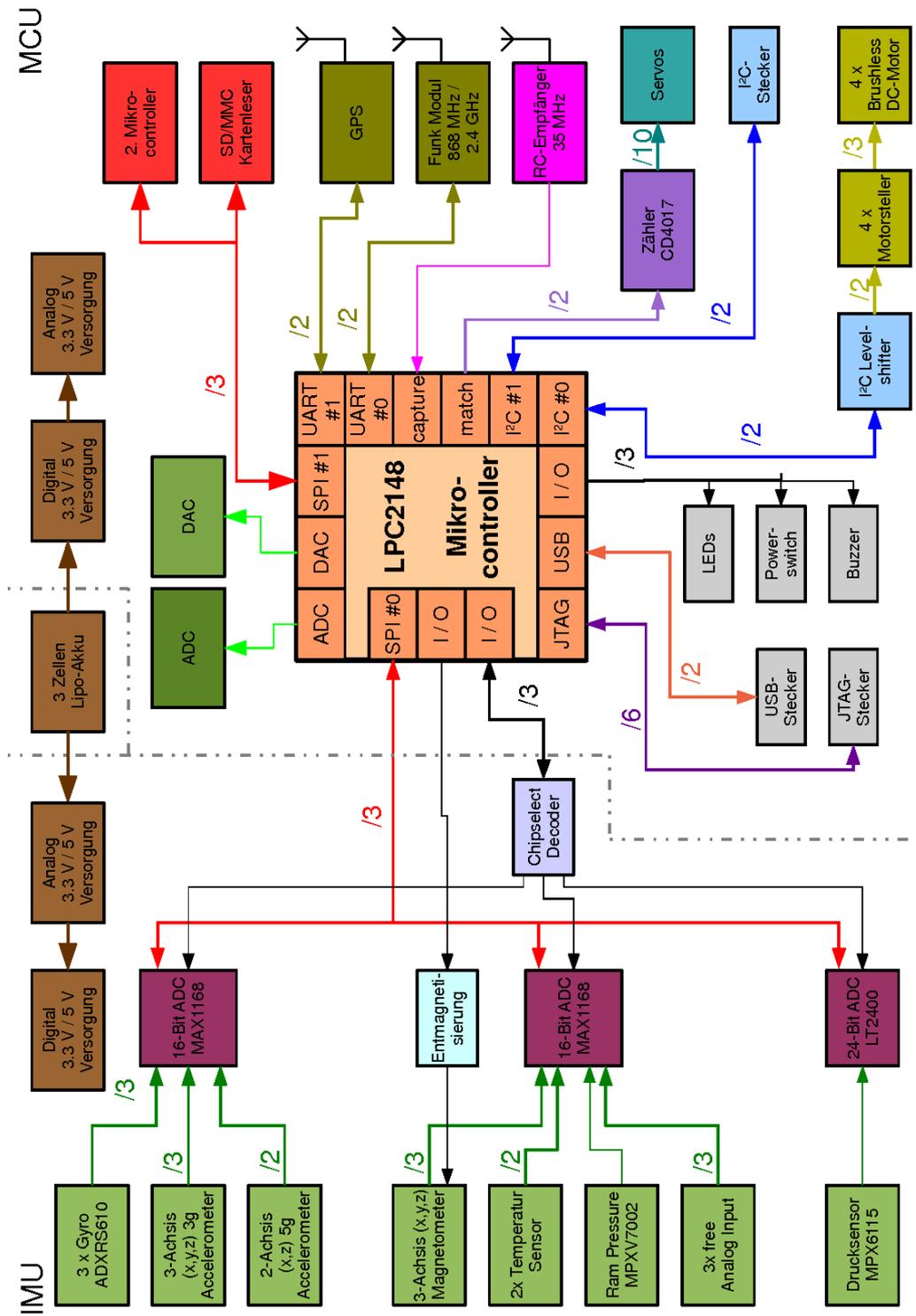


Bild 6.1: Blockschaltbild des Autopiloten

## 7 Schaltungsentwurf

Das Kapitel Schaltungsentwurf befasst sich mit der äußeren Beschaltung der Bausteine, der Dimensionierung der Peripheriebausteine, sowie der Verschaltung der Bauteile untereinander.

Die Schaltpläne sind im Anhang im Kapitel 21.3 aufgeführt.

### 7.1 Mikrocontroller

Der Mikrocontroller LPC2148 [D7] befindet sich in einer Standardschaltung. Alle Versorgungsanschlüsse sind mit 100nF gegen Masse abgeblockt. Die Takterzeugung erfolgt mittels einem 12MHz Quarz und den beiden zugehörigen 22pF Kondensatoren. Die Erzeugung des Resetimpuls beim Einschalten wird über ein RC Glied bestehend aus 10k $\Omega$  gegen 3,3V und 100nF gegen Masse erzeugt. Der Anschluss des USB Ports erfolgte nach Angabe im Datenblatt (Seite 32 Fig. 10) mit den üblichen 33 $\Omega$  Serienwiderständen in den Signalleitungen sowie 18pF Kondensatoren von den Signalleitungen gegen Masse. Die Connectschaltung wurde mit einem Transistor durchgeführt dessen Typ aus den Papparazzi Entwürfen übernommen wurde. Die 5V des USB Bus werden nicht direkt an den VBUS Pin des Controllers gelegt sondern zunächst über einen Spannungsteiler auf 3,3V begrenzt.

### 7.2 Gyroskope

Die drei Gyroskope des Typs ADXRS610 sind nach den Herstellerangaben im Datenblatt [D4] beschaltet.

Die Bauteile der die Gyroskope sind jeweils gleich nummeriert, jedoch unterscheidet sich die erste Ziffer zur Unterscheidung der Achsen. Bauteile beginnend mit 5 bezeichnen die X-Achse, beginnend mit 6 die Y-Achse und beginnend mit 7 die Z-Achse. Beschrieben wird hier nur die X-Achse.

Die analoge 5V und digitale 5V Spannungsversorgung für die Gyroskope sind getrennt ausgeführt und am Baustein mit je 100nF / 10V abgeblockt (C505 und C506). Um die für die Piezoresonatoren erforderliche Spannung von ca. 15V zu erzeugen, besitzen die Gyroskope interne Ladungspumpen, die aus der 5V Versorgungsspannung ca. 15V erzeugen. Die Ladungspumpen benötigen hierzu externe Kondensatoren, die im Datenblatt mit 22nF angegeben sind (C502 und C503). Da die Kondensatoren mit den erzeugten 15V belastet werden wurden hier Kondensatoren der Bauform 0603 verwendet, die eine Spannungsfestigkeit von 50V aufweisen. In Baugröße 0402 wären maximal 16V möglich, und somit keinerlei Sicherheitsreserve, was auf Dauer zum Ausfall der Kondensatoren führen könnte. Gleiches gilt für den Filterkondensator der Ladungspumpen C501, er ist im Datenblatt mit 100nF angegeben und in Bauform 0603 bei 50V Spannungsfestigkeit vorgesehen.

Aus R501 und R502 ist ein Spannungsteiler aufgebaut, um eventuell die Ausgangsspannung der Gyroskope an den Analog-Digital-Umsetzer anzupassen. Im Normalbetrieb werden die Gyroskope vom ADC mit der  $V_{\text{ratio}}$  Spannung in Form der Referenzspannung des ADC versorgt, so dass eine optimale Anpassung zwischen Gyroskopen und ADC erfolgt. In diesem Fall ist R501 mit  $0\Omega$  zu bestücken und R502 ist unbestückt zu lassen. Sollte die  $V_{\text{ratio}}$  Spannung jedoch nicht durch den ADC, sondern durch die Versorgungsspannung geliefert werden, ist eine Pegelanpassung durch den Spannungsteiler notwendig.

R503 dient zum Abtrennen der  $V_{\text{ratio}}$  Spannung. Dies ist nötig wenn Sensoren des Typs ADXRS150 bzw. ADXRS300 bestückt werden sollten (s.u.).

Über C507 kann die maximale Bandbreite der Gyroskope eingestellt werden. Neben C507 wird sie durch einen internen Widerstand mit  $180\text{k}\Omega$  bestimmt. Die Berechnung der Bandbreite erfolgt durch Formel (7.1), die auch im Datenblatt [D4, S.9] aufgeführt ist. Als Standardwert sind für C507  $2,2\text{nF}$  vorgesehen, was eine Bandbreite von ca.  $400\text{Hz}$  ergibt. Mit diesem Wert wurden bisher gute Erfahrungen gemacht. Sollten jedoch zukünftig Probleme auftreten oder Änderungen gewünscht sein, zeigt Tabelle 7.1 geläufige Bandbreitenwerte für C507.

$$f_{\text{out}} = \frac{1}{2 * \pi * R_{\text{out}} * C507} \quad (7.1)$$

Tabelle 7.1: Geläufige Werte für C507

Bandbreite (Hz)	Wert C507 (nF)	Wert C507 nach Reihe (nF)	Resultierende Bandbreite (Hz)
50	17,684	18	49,122
100	8,842	8,2	107,829
200	4,421	4,7	188,127
400	2,210	2,2	401,907
800	1,105	1,0	884,195

Durch die Lötjumper SJ501 und SJ502 können die Gyroskope in den Selbsttestmodus versetzt werden, indem diese geschlossen werden. Die Konfiguration und die zu erwartenden Ausgangsspannungswerte (in Ruhelage) sind in Tabelle 7.2 dargestellt. Die Widerstände R504 und R505 sind Pulldownwiderstände um bei geöffneten Lötjumpfern einen Nullpegel zu garantieren.

Tabelle 7.2: Selbsttestmodi

	<b>SJ501</b>	<b>SJ502</b>	<b>RateOut (Ruhelage)</b>
Normalbetrieb	geöffnet	geöffnet	0V
Test System 1	geöffnet	geschlossen	-0,5V
Test System 2	geschlossen	geöffnet	+0,5V
Test System 1&2	geschlossen	geschlossen	0V

Das Design der Schaltung ist auf Gyroskope der ADXRS61X Serie optimiert. Diese Serie beinhaltet baugleiche und pinkompatible Sensoren mit unterschiedlicher Auflösung. Die verschiedenen erhältlichen Typen mit den Drehraten zeigt Tabelle 7.3.

Tabelle 7.3: Erhältliche Gyroskope

<b>Gyroskop</b>	<b>Maximale Drehrate</b>
ADXRS610	±300°/s
ADXRS612	±250°/s
ADXRS613	±150°/s
ADXRS614	±50°/s

Es ist jedoch auch möglich die älteren Sensoren vom Typ ADXRS150 [D18] bzw. ADXRS300 [D19] zu bestücken. Hierbei sind jedoch kleine Anpassungen im Vergleich zur Originalbestückung durchzuführen.

- C501 ist anstatt mit 100nF mit 47nF zu bestücken
- R503 darf nicht bestückt werden, da der ADXRS150 bzw. ADXRS300 eine eigenen Referenzspannung erzeugt, die nicht in die Schaltung eingespeist werden darf. Eine Bestückung von R503 kann Schäden am Gyroskop oder am ADC zur Folge haben.
- Die Ausgangsspannung ist über den Spannungsteiler an den ADC anzupassen.

Im Vorlauf des Schaltungsdesigns um die ADXRS610 stellte sich die Frage ob die Schaltkreise an dem  $V_{ratio}$  Pin eine Spannung erwarten oder eine bereit stellen. Die Erklärung zum  $V_{ratio}$  Prinzip ist im Datenblatt recht kurz gehalten. Aus dem Text muss man die Annahme ziehen, dass es sich um einen Eingang handelt, an dem der Schaltkreis eine Spannung erwartet, während in einem Übersichtsschaltplan der  $V_{ratio}$  Pin als Ausgang gekennzeichnet ist.

Um kein Risiko einer nicht funktionstüchtigen Leiterplatte einzugehen wurde daher ein Testaufbau hergestellt (Bild 7.1). Der ADXRS610 wurde dazu umgekehrt auf eine Lochrasterplatine geklebt und die Anschlüsse über dünnen Kupferdraht auf die Lochrasterplatine geführt, wo die Montage der restlichen Bauteile erfolgte.

Die erste Inbetriebnahme erfolgte ohne eine Spannungsquelle an  $V_{\text{ratio}}$ . Eine Messung der Spannung an  $V_{\text{ratio}}$  ergab 0V, ebenso war das Drehraten- und das Temperatursignal bei 0V. Daher wurde nun eine Spannung an  $V_{\text{ratio}}$  angelegt. Hiernach war an dem Drehratensignal die Hälfte der Betriebsspannung zu messen, wobei sich das Signal bei Dreheinwirkung veränderte. Ebenso lag ein Temperatursignal vor, welches sich bei Erwärmen des Sensors auch änderte.

Es konnte somit einwandfrei nachgewiesen werden, dass an dem  $V_{\text{ratio}}$  Pin die Referenzspannung des ADC anzulegen ist.

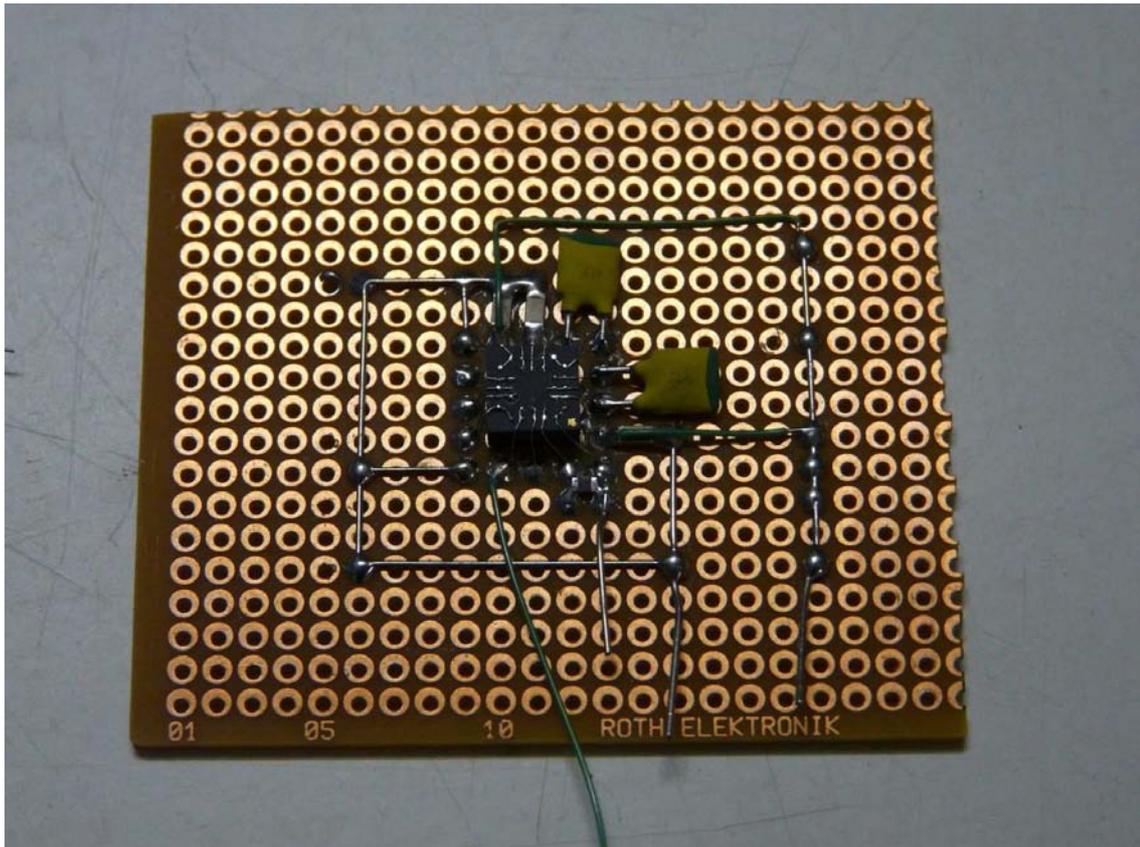


Bild 7.1: Versuchsaufbau für Gyroskop ADXR610

### 7.3 Beschleunigungssensoren

Die IMU besitzt 2 Beschleunigungssensoren, die auf unterschiedliche Art und Weise genutzt werden können.

In der Vollausstattung befindet sich auf der Hauptplatine ein 3-achsiger ADXL330 [D22] mit  $\pm 3g$  Messbereich und auf der senkrechten Y-Platine ein 2-Achsiger ADXL320 [D23] mit  $\pm 5g$  Messbereich. Als weiterer verwendbarer Sensor ist der ADXL323 [D24] mit 2 Achsen und einem Messbereich von  $\pm 5g$  erhältlich.

Je nach Anwenderwunsch können die Beschleunigungssensoren unterschiedlich bestückt werden um unterschiedliche Messbereiche zu erhalten. Als einfachste Variante

ist es möglich den ADXL330 alleine zu betreiben, diese ist jedoch nur auf der Hauptplatine bestückbar. Der Platz auf der Y-Platine bleibt dann unbestückt.

Des weiteren ist es möglich die IMU mit zwei 2-achsigen Sensoren vom Typ ADXL320 oder 323 zu betreiben. Wird einer der 2-achsigen Sensoren anstelle des ADXL330 bestückt so liegt anstelle der Z-Achse ein NC Pin am ADC. Der ADC Port 5 liefert daher kein Signal und darf nicht ausgewertet werden. Die 2-achsigen Sensoren können untereinander je nach Wunsch beliebig ausgetauscht werden, der Sensor auf der Hauptplatine misst dabei die X- und Y-Achse, der Sensor auf der Y-Platine mit die X- und Z-Achse.

Beschaltet sind die Beschleunigungssensoren nach Datenblattangaben. Die Beschaltung unterscheidet sich dabei zwischen den beiden 2-achsigen Sensoren nicht und zwischen den 2-achsigen und dem 3-achsigen Sensor nur durch die zusätzliche dritte Achse.

Die Versorgung der Sensoren erfolgt durch die analoge 3,3V Spannungsversorgung, die direkt am Baustein durch 100nF abgeblockt ist.

Jeder analoge Ausgang besitzt einen Kondensator zu Bandbreitenbegrenzung. Das Datenblatt gibt hierbei in einer Tabelle Wert für verschiedene Bandbreiten an.

Tabelle 7.4: Filterkondensatorauswahl

<b>Bandbreite in Hz</b>	<b>Kondensator in <math>\mu F</math></b>
1	4,7
10	0,47
50	0,1
100	0,05
200	0,027
500	0,01

Für individuelle Bandbreiten können die Kondensatoren nach Formel (7.2) berechnet werden

$$C_{(x,y,z)} = \frac{1}{2 * \pi * 32k\Omega * F_{-3dB}} \quad (7.2)$$

Aufgrund von guten Erfahrungen wurde eine Bandbreite von 100Hz gewählt welche Werten für C343-C345 und C622-C623 von 50nF entsprechen würden. Da diese Wert außerhalb der Standardreihen liegt wurde der nächstliegende Wert von 47nF gewählt, die Bandbreite beträgt daher ca. 106Hz

## 7.4 Kompass

Die Schaltung für den Kompass ist einem Vorschlag des Applikationsberichtes AN00022 [D12] von Philips Semiconductors angelehnt. Die Dimensionierung der analogen Auswertelektronik wurde von dem MK3Mag Modul [8] übernommen, ebenso wie das Hinzufügen der Kondensatorkette zur Erzeugung des Flip Impulses.

## 7.5 Luftdruck

Die Dimensionierung der Abblockkondensatoren des MPXH6115 ist aus dem Datenblatt [D25] entnommen. Der Ausgang des Sensors ist mit einer Kombination von 2 Widerständen und einem Kondensator versehen, die es ermöglichen sowohl Sensoren mit Strom- als auch mit Spannungsausgang zu nutzen. Für den Fall der Nutzung von Sensoren mit Spannungsausgang ist R301 unbestückt zu lassen, R302 und C303 werden nach Schaltplan bestückt. Für die Nutzung von Sensoren mit Stromausgang wird R301 nach Schaltplan bestückt, R302 mit  $0\Omega$  und C302 mit  $47\text{pF}$ , dies entspricht der Angabe im Datenblatt.

Der nachfolgende Impedanzwandler ist auf Empfehlung der AN1646 [D5] zu Rauschreduzierung nachgeschaltet. Entgegen der Angabe im Datenblatt wird hier kein MC33502, der nicht lieferbar ist, verwendet, sondern ein vergleichbarer Rail-to-Rail Operationsverstärker TLC2262. Der Ausgang des Impedanzwandlers ist über einen Spannungsteiler zur Pegelanpassung an den Eingang des 24-Bit ADC gelegt. Der Spannungsteiler kann je nach Bedürfnis noch nachträglich angepasst werden. In der vorgesehenen Version ist er mit je  $3,3\text{k}\Omega$  bestückt, was zur Folge hat, dass die Ausgangsspannung des Impedanzwandlers halbiert wird, begründet dadurch, dass der ADC einen Eingangsspannungsbereich von  $V_{REF}/2$  besitzt. Bei einer Referenzspannung von  $5\text{V}$  bedeutet dies, dass Spannungen im Bereich von  $-2,5\text{V} - +2,5\text{V}$  verarbeitet werden können. Spannungen über  $2,5\text{V}$  liefern einen Maximalausschlag. Die Ausgangsspannung des Luftdrucksensors beträgt maximal ca.  $4,8\text{V}$ , die vom Impedanzwandler nicht verändert werden. Durch den Spannungsteiler ergibt sich somit eine Spannung von etwa  $2,4\text{V}$  am Eingang, was noch eine gewisse Reserve zum maximalen Messbereich darstellt um so Überschreitungen des Messbereichs durch etwaige Toleranzen zu vermeiden.

Die Beschaltung der Versorgungsspannung des LTC2440 mit Kondensatoren erfolgt nach Datenblattempfehlung. Die Referenzspannung wird durch die analoge  $5\text{V}$  Versorgungsspannung gebildet, ist jedoch durch eine zusätzliche Induktivität von  $10\mu\text{H}$  und einen  $100\text{nF}$ , zur Vermeidung von Störungen, entkoppelt.

In den sendenden I/O Leitungen des SPI Bus des LTC sind Spannungsteiler integriert um den  $5\text{V}$  Pegel des LTC2440 auf die  $3,3\text{V}$  Pegel des LPC2148 herabzusetzen. Zwar wird beim LPC2148 angegeben, dass seine Eingänge  $5\text{V}$  tolerant sind, jedoch hat es dabei in der Vergangenheit bereits einige Probleme mit Zerstörung von I/O Pins gegeben. Die Spannungsteiler sind daher als Sicherheitsmassnahme integriert worden.

Umgekehrt erkennt jedoch der LTC2440 die 3,3V Pegel des LPC2148 eindeutig als High Pegel, so dass hier keine weiteren Maßnahmen erforderlich sind.

## 7.6 Staudruck

Der Differenzdrucksensor MPXV7002 [D26] verfügt über einen Messbereich von  $\pm 20$ mbar bei einer Aussteuerung von 0,5V – 4,5V. Ohne einen Druckunterschied beträgt die Ausgangsspannung 2,5V. Für die Staudruckmessung wird nur der positive Messbereich, also von 2,5V – 4V verwendet. Durch den zu großen Messbereich des Sensors werden bei einem erwarteten maximalen Druckunterschied von 2,58mbar nur effektiv ca. 12,9% des Messbereichs genutzt, was einer Ausgangsspannung von 258mV entspricht. Um den Bereich des ADC besser ausnutzen zu können muss die geringe Ausgangsspannung verstärkt werden, wobei sich das Problem ergibt, dass der Nullpunkt bei 2,5V liegt und so diese 2,5V immer mitverstärkt wurden. Aus diesem Grund wurde die virtuelle Masse des nachfolgenden nicht invertierenden Verstärkers, in Form des TLC2262, durch R309 und R311 auf 2,5V angehoben. Der Verstärker selbst sollte die 258mV auf ca. 3,8V verstärken. Die maximale Aussteuerung des ADC wird zwar erst bei 4,096V erreicht, jedoch sind diverse Toleranzen zu berücksichtigen, so dass eine Reserve eingeplant wurde um ein Übersteuern zu vermeiden. Für eine Verstärkung von 0,2V auf 3,8V ist eine Verstärkung von 14,7 notwendig woraus sich aus der umgestellten Gleichung (7.3) zur Verstärkungsberechnung für nicht invertierende Operationsverstärker ein Widerstandswert für R312 von 301,4k $\Omega$  ergibt. Der nächstliegende Wert einer Reihe beträgt 300k $\Omega$ . Der Wert für R311 wurde bereit im Zuge der Massehochlegung auf 22k $\Omega$  festgelegt.

$$R312 = (v - 1) * R311 = (14,7 - 1) * 22k\Omega = 301,4k\Omega \quad (7.3)$$

Der sich am Ausgang befindliche Spannungsteiler ist für individuelle Anpassung vorgesehen, wird in der Standardversion jedoch nicht gebraucht. Daher ist R313 mit 0 $\Omega$  zu bestücken und R314 ist unbestückt zu lassen.

## 7.7 Temperatur

Die Beschaltung des Temperatursensors MAX6605 (IC351) entspricht der im Datenblatt [D6] empfohlenen.

Durch R351 ist es möglich den Temperatursensor vom ADC zu trennen, falls der Sensor nicht benötigt oder der ADC Kanal anderweitig benötigt wird.

## 7.8 Analog – Digital – Umsetzer

Die IMU verwendet zwei Analog Digital Umsetzer MAX1168 [D14] von Maxim zur Digitalisierung der Sensordaten. Die Aufteilung der Sensoren auf beide ADC erfolgte so, dass der erste ADC alle schnellen Signale, also die der Gyroskope und der Beschleunigungssensoren verarbeitet und der zweite die langsameren Daten vom Magnetsensor, Staudruck sowie Temperatur verarbeitet.

Die ADC lassen sich in zwei verschiedenen Moden betreiben, wofür einer speziell für den Einsatz zusammen mit einem DSP konzipiert ist, während der andere den normalen Betrieb per SPI Schnittstelle darstellt. Durch die Widerstände R401 und R402 bzw. R411 und R412 sind die ADC für den Betrieb mit der SPI Schnittstelle konfiguriert.

Die Beschaltung mit Kondensatoren erfolgte nach Datenblattempfehlungen [D14].

Bei ADC0 besteht die Besonderheit, dass seine interne Referenzspannung über den  $0\Omega$  Widerstand R403 zu den Gyroskopen geleitet wird, welche wiederum diese als  $V_{\text{ratio}}$  Spannung nutzen um ihren Messbereich auf diese Spannung anzugleichen.

## 7.9 Spannungsversorgung

### 7.9.1 LT1761

Alle in dem Projekt verwendeten LT1761 sind nach den Datenblattangaben [D3] beschaltet. Der  $2,2\mu\text{F}$  Eingangstantalkondensator ist aufgrund der hohen möglichen Eingangsspannung als  $25\text{V}$  Typ festgelegt. Der  $10\mu\text{F}$  Ausgangstantalkondensator ist bei der  $3,3\text{V}$  Ausgangsspannungen als  $6,3\text{V}$  Typ ausgelegt, bei den  $5\text{V}$  Spannungen als  $10\text{V}$  Typ. Die analogen Regler sind am Eingang zusätzlich mit einer  $0,47\mu\text{H}$  Induktivität versehen.

### 7.9.2 LM1117

Beide sich im Projekt befindlichen LM1117 wurden in gleicher Art nach Datenblattangaben [D28] beschaltet, dies sieht einen  $10\mu\text{F}/10\text{V}$  Tantal-Kondensator am Eingang vor, wenn sich der Schaltkreis in großer Entfernung zum Ladeelektrolytkondensator befindet. Da kein Ladeelektrolytkondensator vorhanden ist, ist dieser Kondensator somit vorzusehen. Weiter empfiehlt das Datenblatt einen  $22\mu\text{F}$  Tantal-Kondensator am Ausgang. Zusätzlich wurde dem Ausgangskondensator noch ein keramischer  $100\text{nF}$  Kondensator parallel geschaltet um hochfrequente Störspitzen zu unterdrücken.

### 7.9.3 PTH08080

Die Beschaltung des PTH08080 erfolgte nach Datenblattempfehlungen [D15]. Dieses sieht für den Eingang einen  $100\mu\text{F}$  Kondensator vor, welcher einen niedrigen ESR aufweist. Dies wurde in der Auswahl des Kondensator berücksichtigt. Die

Spannungsfestigkeit wurde mit 25V hoch gewählt, was jedoch aufgrund der hohen möglichen Eingangsspannung mit zusätzlicher Reserve als sinnvoll erschien. Als Typ wurde hier ein Aluminiumelektrolytkondensator verwendet, da laut Datenblatt bei Tantalkondensatoren die Spannungsfestigkeit mindestens das Doppelte der zu erwartenden Gleichspannungsbelastung betragen muss, was zu einem erheblich mehr an Platzbedarf (mehrere Kondensatoren nötig) und zu deutlich höheren Kosten geführt hätte. Bei Aluminiumelektrolytkondensatoren trifft dieser Fall nicht zu.

Als Ausgangskondensator wurde ein Tantalkondensator mit 150 $\mu$ F verwendet. Zwar benötigt der PTH08080 nicht zwingend einen Ausgangskondensator, jedoch lässt sich hiermit die durch den Schaltregler verursachte Restwelligkeit mindern.

## 7.10 Chipselect Decoder

Die Chipselectschaltung besteht aus zwei 74HC138, wovon jeweils einer auf der IMU und einer auf der MCU bestückt ist. Die Versorgungsspannung ist nah am Schaltkreis durch 100nF abgeblockt. Die Versorgung erfolgt mit 3,3V, somit besitzen alle Chipselectsignale ebenso 3,3V Pegel. Die Eingangssignale vom Controller sind mit 4,7k $\Omega$  versehen um sofort nach dem Einschalten einen High Pegel zu erhalten falls der Controller zu diesem Zeitpunkt noch undefinierte Signale ausgibt.

Auf der IMU selektieren die ersten 3 Ausgänge die 3 ADC. Auf der MCU befindet sich kein SPI Gerät, daher sind alle Chip Select Signale über SPI1-A (X117) herausgeführt. Die bereits auf der IMU vergebenen Signale wurden auf Lötkontakte herausgeführt.

## 7.11 Leistungsschalter / Kameraschalter

Der Leistungsschalter TPS2015 für die Kamera ist nach Datenblattangaben [D29] beschaltet, dieses sieht einen 100nF Kondensator am Eingang wie auch am Ausgang vor. Zusätzlich ist am Ausgang ein 22 $\mu$ F Kondensator nahe der Last vorzusehen.

Der Überstromsignalausgang bleibt unausgewertet, die Steuerung erfolgt durch den Mikrocontroller über den Enable-Anschluss.

## 7.12 I<sup>2</sup>C Levelshifter

Der I<sup>2</sup>C Levelshifter PCA9306 wurde nach Datenblattangaben [D30] beschaltet. Jedoch wurden nicht die im Datenblatt empfohlenen Widerstandswerte für die Pullup-Widerstände der Busleitungen verwendet, da es mit diesen Werten zu Fehlfunktionen auf dem Bus kam. Die Pullup-Widerstände wurden daher mit 3,3k $\Omega$  bestückt, mit welchen eine Versuchsschaltung, bei der die Widerstände aufgrund eines Listenfehlers eigentlich falsch bestückt waren, bereits einwandfrei funktionierte.

### 7.13 Servo/Zähler

Der CMOS Zähler CD4017 benötigt als externe Beschaltung lediglich einen 100nF Kondensator zum Abblocken der Versorgungsspannung. Die Versorgung erfolgt durch die digitale 5V Versorgung.

Die Ausgangssignale werden auf Steckleisten geführt, wovon Kanal 1-6 auf 2,5mm Stiftleisten geführt werden und Kanal 7-10 auf 3polige Molexstecker gelegt wurden.

### 7.14 Steckleisten

Die Anordnung der Signale auf den Steckerleisten erfolgt nach einem bestimmten Prinzip, so ist Pin 1 immer Masse gefolgt von der analogen Masse. Weiterhin die digitalen Spannungen mit ihrer Höhe abfallend gefolgt von den analogen Spannungen. Zum Ende des Steckers folgen die Signalleitungen. Nicht auf jedem Stecker sind alle Spannungen vorgesehen, in dem Fall erfolgt die Anordnung nacheinander nach oben genannter Reihenfolge.

Eine Ausnahme bildet der Anschluss des USB Signals (X108), dieser ist nach dem allgemeinen Standard des USB Steckers angeschlossen, d.h. auf Pin 1 5V und auf dem letzten Pin GND.

#### 7.14.1 UART0 X109 (Funk)

UART0 ist ein 8-poliger Molexstecker der die UART0 Schnittstelle nach außen zur Verfügung stellt. Primär ist sie zum Anschluss des Funkmoduls gedacht. Zusätzlich beherbergt sie einen I/O Pin des Controllers um eventuelle Zusatzsignale des angeschlossenen Gerät bedienen zu können. Da UART0 ebenfalls zur Programmierung des Controllers benötigt wird ist zusätzlich das Resetsignal P0.14, der zum aktivieren des Programmiermodus benötigt wird, herausgeführt.

Tabelle 7.5: Belegung X109 UART0

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse
2	+5V_MCU	5V digital
3	+3V3_MCU	3,3V digital
4	TXD0	Transmit Data
5	RXD0	Receive Data
6	FUNK_IO	I/O Pin (P1.17)
7	/RESET	Reset
8	SDA_P0.14	Programmierung

### 7.14.2 UART1 X110 (GPS)

Analog zu UART0 ist UART1 ebenfalls ein 8-poliger Molexstecker mit gleichen Signalen um Geräte ohne Steckerwechsel oder Adapter an beide seriellen Schnittstellen anschließen zu können. Pin8 ist unbeschaltet da über UART1 keine Programmierung möglich ist.

Tabelle 7.6: Pinbelegung X110 UART1

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse
2	+5V_MCU	5V digital
3	+3V3_MCU	3,3V digital
4	TXD0	Transmit Data
5	RXD0	Receive Data
6	GPS_TIMEPULS	I/O Pin (P1.16)
7	/RESET	Reset
8	-	-

### 7.14.3 I<sup>2</sup>C0 X111

I<sup>2</sup>C0 ist auf einem 3-poligen Molexstecker herausgeführt und ist primär zum Anschluss der Motorsteller gedacht. Die Schnittstelle hat 5V Pegel und ist durch einen Levelshifter, der die Signalpegel zwischen 3,3V und 5V anpasst, an den Mikrocontroller angepasst.

Tabelle 7.7: Pinbelegung X111 I<sup>2</sup>C0

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse
2	SDA0	I <sup>2</sup> C Daten (5V)
3	SCL0	I <sup>2</sup> C Takt (5V)

### 7.14.4 I<sup>2</sup>C1 X112

I<sup>2</sup>C1 ist auf einem 6poligen Molexstecker herausgeführt und direkt mit dem Mikrocontroller verbunden. Zum Schutz der Controllerpins sind in den Signalleitungen 100Ω Serienwiderstände (R192 & R193) eingefügt. An den Signalleitungen sind zwei Gruppen von Pullup-Widerständen vorgesehen, die es ermöglichen den Bus entweder mit 3,3V oder mit 5V Pegel zu betreiben. Es ist jedoch zu beachten, dass bei Verwendung von 5V ein erhöhtes Risiko auf Beschädigung der Controllerpins besteht.

Für 3,3V Buspegel sind R188 & R189 zu bestücken während R190 und R191 unbestückt bleiben müssen.

Für 5V Buspegel sind R190 & R191 zu bestücken während R188 und R189 unbestückt bleiben müssen.

Auf SDA1 liegt ebenso ein Programmiersignal, daher sollte während der Programmierung nichts an I<sup>2</sup>C1 angeschlossen sein.

Da an I<sup>2</sup>C1 ebenfalls ein Magnetsensor angeschlossen werden kann, dies wurde vorgesehen falls der auf der IMU integrierte Sensor durch Störfelder beeinflusst wird, ist zusätzlich auf Pin 6 der für den Magnetsensor nötige I/O Pin herausgeführt.

Tabelle 7.8: Pinbelegung X112 I<sup>2</sup>C1

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse
2	+5V_MCU	5V digital
3	+3V3_MCU	3,3V digital
4	SDA_P0.14	I <sup>2</sup> C Daten
5	SCL1	I <sup>2</sup> C Takt
6	MAG_IO	I/O Pin für Mag-Sens

#### 7.14.5 SPI1-1 X117

Die SPI1 Schnittstelle ist über zwei Molexstecker herausgeführt. Während X117 alle Signale besitzt sind auf X114 nur einige Signale herausgeführt. Die SPI1 ist zum Anschluss eines weiteren Rechnerboards, einer SD-Karte oder eines Magnetsensors.

Auf X117 sind zusätzlich zu den 3 SPI Signalen und SSEL die beiden Betriebsspannungen sowie alle noch freien Chip-Select-Leitungen des Chip-Select-Decoders herausgeführt. Die freien Chip-Select-Leitungen reagieren dabei auf die binären Bitkombinationen 3-6.

Auf X114 liegen nur 3,3V sowie nur die erste freie Chip-Select-Leitung.

Tabelle 7.9: Pinbelegung X117 SPI1-1

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse
2	+5V_MCU	5V digital
3	+3V3_MCU	3,3V digital
4	SSEL1	SSEL1
5	MOSI1	MOSI1
6	MISO1	MISO1
7	SCK1	SCK1
8	CS_RES_1	Freier CS (CS#3)
9	CS_RES_2	Freier CS (CS#4)
10	CS_RES_3	Freier CS (CS#5)
11	CS_RES_4	Freier CS (CS#6)

Tabelle 7.10: Pinbelegung X114 SPI1-2

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse
2	+3V3_MCU	3,3V digital
3	SSEL1	SSEL1
4	MOSI1	MOSI1
5	MISO1	MISO1
6	SCK1	SCK1
7	CS_RES_1	Freier CS (CS#4)

### 7.14.6 Analog X116

Der Analogsteckverbinder X116 ist ein 8-poliger Molexstecker auf dem analoge Eingänge des LPC2148 nach außen geführt sind. Darunter auch der DAC des LPC2148, der entweder als DAC oder ADC1 verwendet werden kann. ADC5 ist zusätzlich zu einem Interruptpin umkonfigurierbar. Um auch höhere äußere Spannungen verarbeiten zu können ist ADC5 mit einem Spannungsteiler ausgestattet, der je nach Bedarf bestückt werden kann. In der Standardversion ist der Pin der Steckerleiste direkt mit dem LPC2148 verbunden, dies bedeutet, dass R195 mit  $0\Omega$  bestückt ist während R196 unbestückt ist.

Zu beachten ist weiterhin, dass auf ADC4 und ADC5 über  $0\Omega$  Widerstände die Statussignale von zwei ADC der IMU gelegt werden können. Ist dies der Fall dürfen ADC4 und ADC5 nicht extern beschaltet werden.

Tabelle 7.11: Pinbelegung X116 Analog

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GNDA	Masse analog
2	+5V_A_MCU	5V analog
3	+3V3_A_MCU	3,3V analog
4	DAC_ADC1	Analogausgang (AUOT) oder Analogport (AD0.4)
5	ADC2	Analogport (AD1.2)
6	ADC3	Analogport (AD1.3)
7	ADC4	Analogport (AD1.4)
8	ADC5_INT	Analogport (AD1.5) o. Interrupt (EINT2)

### 7.14.7 IR0 X107

IR0 ist ein 6-poliger Molexstecker, welcher drei analoge Eingänge des LPC2148 nach außen führt und speziell zum Anschluss der Infrarotsensoren zur Lageerkennung dient. Zusätzlich werden die analogen 3,3V sowie 5V mit der analogen Masse herausgeführt.

Tabelle 7.12: Pinbelegung X107 IR0

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GNDA	Masse analog
2	+5V_A_MCU	5V analog
3	+3V3_A_MCU	3,3V analog
4	IR1	Analogport für IR Sensor (AD0.1)
5	IR2	Analogport für IR Sensor (AD0.2)
6	IR3	Analogport für IR Sensor (AD0.3)

### 7.14.8 RC PPM IN X113

RC\_PPM ist ein 6-poliger Molexstecker zum Anschluss des Fernsteuerempfängers. Das Summensignal wird zum Schutz des Controllpins über 100Ω an den Controller geleitet. Zusätzlich sind auf dem Stecker noch die Signale SERVO\_CLK und SERVO\_RST untergebracht, welche es ermöglichen hier auch extern einen Zählerbaustein für die Servos anzuschließen, bzw. einen Empfänger soweit umzubauen um dessen internen Zähler zu nutzen.

Die beiden zur Verfügung stehenden Betriebsspannungen sind zur Vermeidung von Störungen mit je 100nF gegen Masse abgeblockt.

Tabelle 7.13: Pinbelegung X113 RC PPM IN

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse digital
2	+5V_MCU	5V digital
3	+3V3_MCU	3,3V digital
4	RC_PPM	Eingang Fernsteuer- summensignal
5	SERVO_RST	Zähler Reset
6	SERVO_CLK	Zähler Takt

### 7.14.9 CAM X106

CAM ist ein 7-poliger Molexstecker und dient zum Anschluss einer Kamera. Er stellt durch ein L-R Glied gefiltert, die Batteriespannung, 5V und Masse des digitalen Zweig zur Verfügung. Zusätzlich werden die Servokanäle S6-S8 herausgeführt, um Servos zur Kamerapositionierung betreiben zu können. Ebenso wird eine vom Controller schaltbare 5V/1A Versorgung bereitgestellt.

Tabelle 7.14: Pinbelegung X106 CAM

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	GND	Masse
2	V+	Batteriespannung
3	+5V_MCU	5V digital
4	S6	Servosignal 6
5	S7	Servosignal 7
6	S8	Servosignal 8
7	PWR_SW	5V geschaltet

### 7.14.10 USB X108

USB ist als 4-poliger Molexstecker vorgesehen und dient zum Anschluss des USB Signals an den Controller. Zu beachten ist, dass der Stecker im Gegensatz zu allen anderen Steckverbindungen nicht mit Masse auf Pin1 beginnt, sondern nach allgemeinem USB Standard beschaltet ist, die bedeutet, dass auf Pin1 5V des USB Bus liegen auf Pin 2 D-, auf Pin 3 D+ und auf Pin 4 GND.

Tabelle 7.15: Pinbelegung X108 USB

<b>Pin</b>	<b>Signal</b>	<b>Beschreibung</b>
1	V+_USB	5V des USB Bus
2	USB-	USB Daten -
3	USB+	USB Daten -
4	GND	Masse

## 8 Bauteilpositionierung

### 8.1 IMU

Als mechanische Vorgaben für die IMU waren vier 3mm Befestigungsbohren in quadratischer Anordnung mit je 30mm Lochabstand vorgeschrieben. Die Abmessungen der Platine waren so klein wie möglich zu halten. Die Schlauchanschlüsse des Differenzdrucksensors durften nur nach vorne oder hinten zeigen und nicht über den Platinenrand hinaus zeigen.

Beginnend wurde die Anordnung der senkrecht gestellten Sensorplatinen ermittelt. Hierzu war maßgebend, dass der 2-achsige Beschleunigungssensor senkrecht zur Platinenoberfläche und parallel zu Flugrichtung stehen muss um die X und Z Achse messen zu können.

Von den Gyroskopen muss jeweils eins quer zur Flugrichtung und eins parallel zu Flugrichtung aufgestellt werden. Zur Vermeidung von Winkelfehlern ist es sinnvoll beide aufgestellten Platinen zu einem Winkel zu verbinden, so das sich ein Winkel in L-Form ergibt, wobei der längere Steg parallel zur Flugrichtung liegt.

Der Winkel der Leiterplatten wurde nun als Eagle Symbol erstellt und konnte so probeweise in dem zur Verfügung stehenden Platz positioniert werden. Hierbei stellte sich schnell heraus, dass der Winkel in einem Quadrat von 40x40mm Grundfläche, welches sich bei einem Bohrabstand von 30x30mm ergibt, nicht unterbringen lies. Der Winkel hätte in diesem Fall mitten auf der Platine gelegen und so keinerlei Platz für den Differenzdrucksensor und die Z-Gyroskopplatine ließ. Die Länge der Platine wurde daher auf 45mm erhöht, so dass der Winkel um die obere linke Befestigungsbohrung herum bis an den linken Rand der Trägerplatine gelegt werden konnte. Somit blieb rechts der Winkelplatine genügend Platz für den Differenzdrucksensor. Der Differenzdrucksensor muss zwingend auf der gleichen Seite wie die aufgestellten Platinen montiert werden, da er sonst die Bauhöhe des gesamten Moduls unnötig vergrößert hätte. Der Sensor hätte nun mit den Schlauchanschlüssen nach oben oder unten zeigend montiert werden können. Es zeigte sich, dass es durch den ADC, der auf der Unterseite unter dem Winkel positioniert wurde um kurze Leitungslängen zu

ermöglichen, sinnvoll war, den Differenzdrucksensor noch unten zeigend zu positionieren, da somit oben genügend Platz für die Steckleiste des ADC blieb. Zu positionieren war noch der Magnetsensor für die Z-Achse, welcher ebenfalls senkrecht aufgestellt werden musste. Ob dieser Sensor nun parallel oder quer zur Flugrichtung vorgesehen wird, war jedoch aufgrund seiner Achsausrichtung unerheblich. Da jedoch die quer zur Flugrichtung ausgerichtet Platine nicht vergrößert werden konnte, da sie ansonsten über den Rand der Trägerplatine hinausgeragt hätte bzw. mit einer Steckerleiste kollidiert wäre, blieb nur die Verlängerung der parallel zur Flugrichtung positionierten Platine. Dies war auch in der Hinsicht vorteilhaft, da an dieser Position noch genügend Platz für den X- und Y-Magnetsensor war, so dass alle Magnetsensoren räumlich im gleichen Gebiet untergebracht werden konnten. Da jedoch der Platz wegen des Differenzdrucksensors begrenzt war, mussten die letzten 3 Kontaktflächen der Sensorplatine so ausgelegt werden, dass sie nur von einer, anstatt von beiden Seiten, zu verlöten waren um genügend Platz für den Differenzdrucksensor zu lassen.

Die restlichen Bauteile wurden in die noch vorhandenen Lücken so positioniert, das sich möglichst kurze Leitungswege ergaben. Der als extra Platine ausgeführte Z-Gyroskop wurde hierbei in zweiter Ebene über anderen Bauteilen positioniert.

Bild 8.1 zeigt die Anordnung der Achsen auf der Leiterplatte.

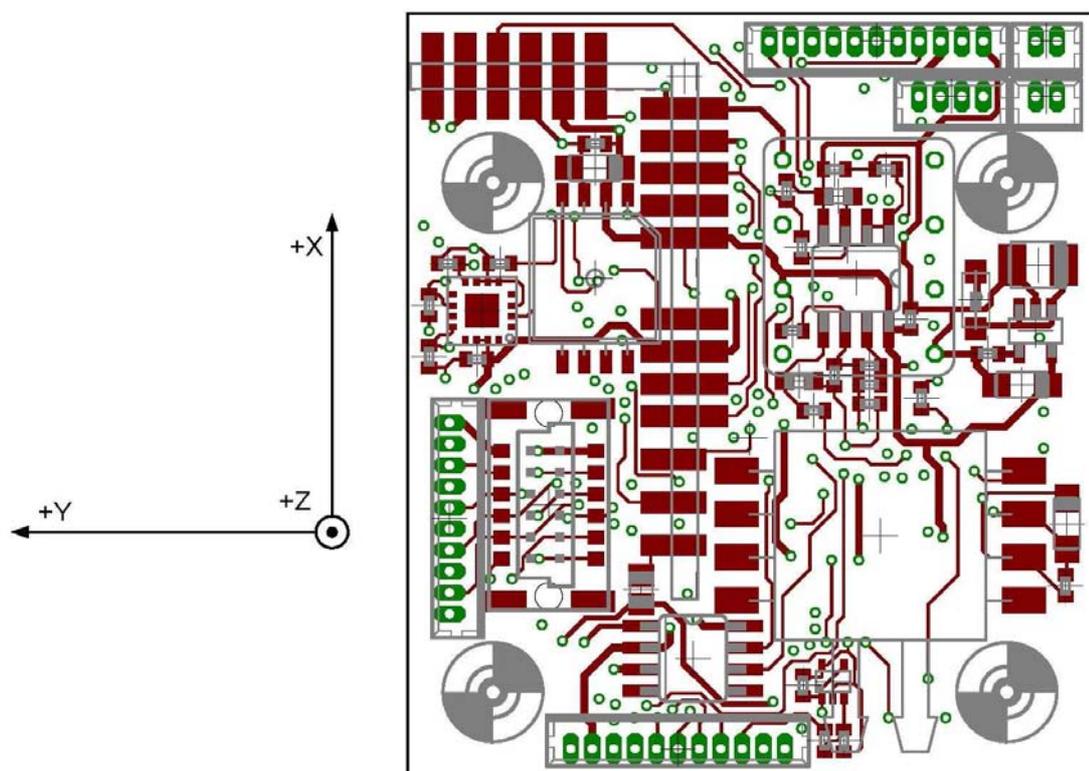


Bild 8.1: Achsenanordnung der Leiterplatte

## 8.2 MCU

Dadurch, dass die Steckverbinder, die IMU und die MCU miteinander verbinden, bereits auf der IMU eine feste Position erhalten haben, war die Position auf der MCU auch bereits festgelegt, da die Verbinder untereinander liegen müssen.

Bei der MCU stellte sich als größtes Problem heraus die Steckleisten auf der Platine unterzubringen. Es war auf der gleichen Grundfläche wie der IMU zuwenig Platz. Steckleisten in zwei Reihen hintereinander anzuordnen scheiterte, da dafür ebenfalls die Grundfläche nicht ausreichte, sowie Leiterbahnen von Steckern der hinteren Reihe nicht an der vorderen Reihe vorbei geführt werden konnten. Die Verwendung von SMD Molexsteckern schied ebenso aus, da diese zwar durch die SMD Montage nicht die untere und mittleren Lagen beeinflussen, jedoch einen sehr großen Flächenbedarf besitzen, so dass die komplette Oberseite mit Steckern blockiert gewesen wäre und kein Platz für andere Bauteile mehr bestanden hätte. Ein Versuch die Leiterplatte im gleichen Maß wie die IMU zu routen scheiterte. Es war daher nötig die MCU Platine nach unten um weitere 7mm zu vergrößern. Durch den entstanden Platz konnten mehrere Steckleisten nebeneinander unter den Befestigungsbohrungen platziert werden, die vorher zwischen den Bohrungen in mehreren Reihen hintereinander platziert werden mussten. Durch die Vergrößerung der Grundfläche bestand nun ausreichend Platz, so dass die restlichen Bauteile problemlos, unter Beachtung möglichst direkter Leitungswege, positioniert werden konnten. Auf der Unterseite war jedoch zu beachten, dass keine hohen Bauteile an Positionen platziert wurden an denen sich hohe Bauteile auf der IMU Platine befinden, um ein Zusammenstoßen zu verhindern. Dies bedeutet, dass an der Position, wo der Platinenwinkel mit den Sensoren sowie an der Stelle des Differenzdrucksensors, keine hohen Bauteile platziert werden durften.

## 9 Das Routen

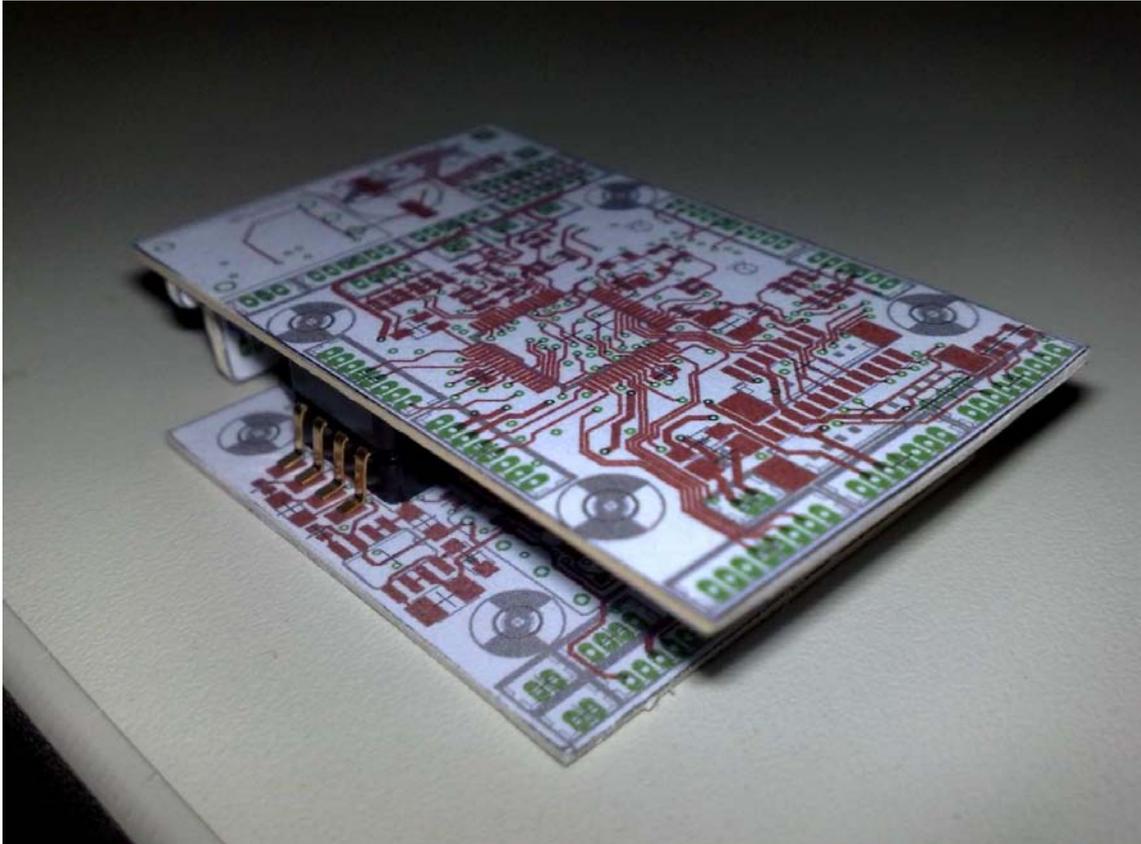
Das Routen erfolgte unter Beachtung üblicher Regeln. Allgemein wurden der obere und untere Layer für kurze Wegstrecken, also die Verbindung von benachbarten Bauteilen, genutzt. Über den oberen der beiden Mittellayer wurden Signale geführt welche lange Wegstrecken auf der Platine zurücklegen. Der untere der Mittellayer war einer Massefläche vorbehalten. Leitungen welche stark störende Signale führen (SPI) wurden in Abstand und möglichst nicht parallel zu empfindlichen, analogen Leitungen verlegt.

Signalleitungen wurden mit einer Leiterbahnbreite von 6mil verlegt. Stromversorgungsleitungen wurden mit dynamischer Breite verlegt. Endstücke, die nur noch ein Bauteil versorgen, wurden ebenfalls mit 6mil verlegt. Bereiche mit höheren Strömen besitzen breiten bis zu 16mil.

Das Routen erfolgte ausschließlich manuell per Hand um einen optimalen Leiterbahnverlauf zu erreichen und die Packungsdichte möglichst hoch zu gestalten.

Spätere Versuche mit Autoroutern, um eventuell bessere Ergebnisse zu erreichen, scheiterten. Weder der Eagle interne Autorouter noch der Autorouter Free Route waren in der Lage, alle Leitungen zu verlegen, was per Hand bereits geschafft wurde.

Nach vollendeter Bauteilausrichtung und Routingvorgang wurde ein Pappmodell des Autopiloten erstellt um zu kontrollieren ob die Baugruppen mechanisch zusammen passten oder ob sich Bauteile gegeneinander behinderten. Mit Hilfe des Modells konnten mechanische Konflikte ausgeschlossen werden und zugleich konnte ein Eindruck gewonnen werden wie der neue Autopilot aussehen wird.



*Bild 9.1: Pappmodell mit beiden übereinander gelegten Teilen*

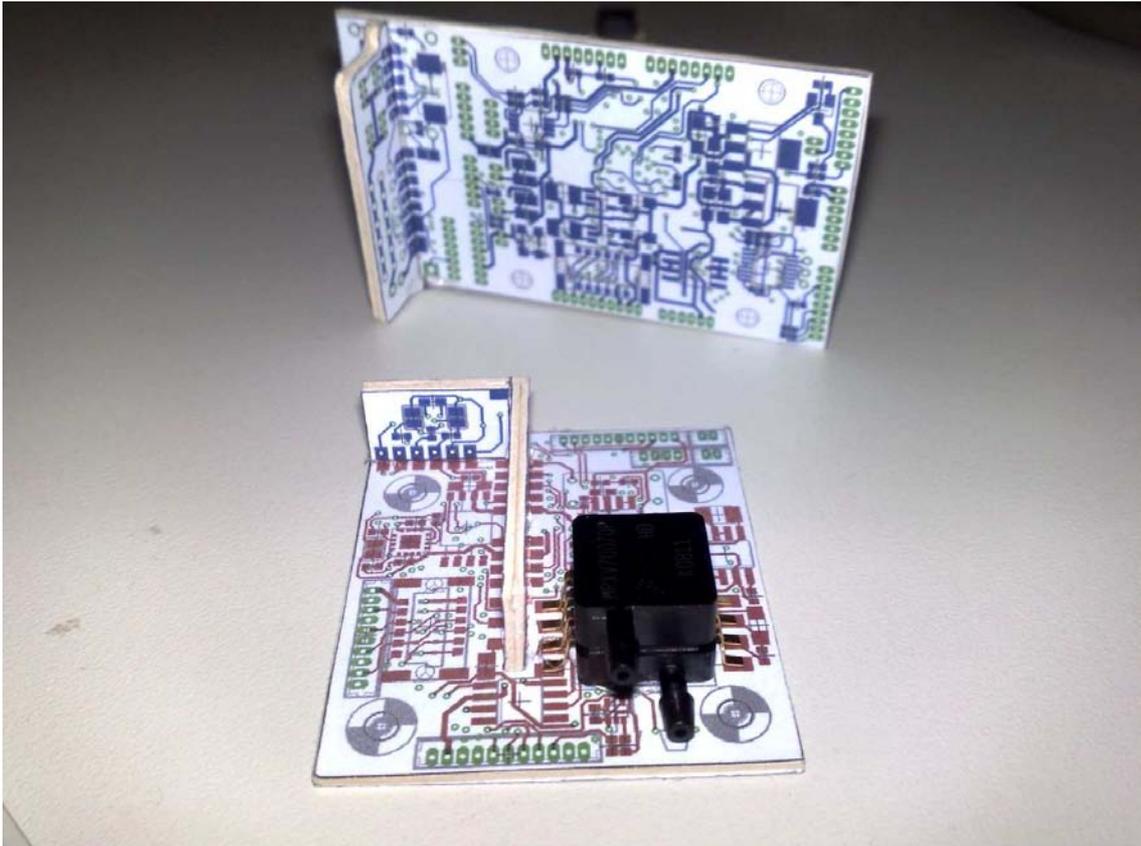


Bild 9.2: Pappmodell mit beiden einzelnen Teilen

## 10 Prozess zum Löten von Bauteilen in BGA und QFN Gehäusen

### 10.1 Bestehendes

Zum Löten der BGA und QFN Gehäuse ist es erforderlich einen Reflowlötprozess zu benutzen. Hierzu stand aus einem früheren Projekt ein Pizzaofen mit externer Steuerung durch eine Schaltung der Zeitschrift Elektor [11] zur Verfügung.

Der Ofen wurde von mir im Rahmen einer Aushilfstätigkeit nach dem Entwurf der Elektor aufgebaut und in Betrieb genommen. Eine genaue Einstellung sowie Ermittlung eines optimalen Temperaturverlaufs erfolgte jedoch noch nicht.

Ein Reflowprozess beinhaltet einen bestimmten Temperaturverlauf, der von der Reflowanlage möglichst gut realisiert werden sollte. Als Anhaltswerte sind in manchen Datenblättern von Bauteilen Reflowtemperaturverläufe vorgegeben.

Ein Reflowtemperaturverlauf gliedert sich in folgende Phasen:

- Phase 1: Vorheizen
- Phase 2: Halten
- Phase 3: Anheizen auf Reflowtemperatur
- Phase 4: Reflowphase
- Phase 5: Abkühlen

Nach Aufbau des Ofens wurden bereits einige Tests durchgeführt. Hierbei stellte sich bei Verwendung der von Elektor in der Software angegebenen Standardparameter heraus, dass diese viel zu hoch eingestellt waren, was zu einem Verbrennen der Platine führte. In weiteren Versuchen wurden die Temperaturen weiter gesenkt, es traten jedoch immer mal wieder leichte Bräunungen, vor allem an faserigen Schnittkanten, auf. Da bereits die Temperaturen soweit gesenkt wurden, dass zu bezweifeln war ob noch eine saubere Verlötung erzielt werden konnte, war die Überlegung ob direkte Infrarotstrahlung die Platinenoberfläche mehr erhitzte als die Luft um den Temperatursensor. Um diesen Effekt auszuschließen wurde unter den oberen Heizstab eine Abdeckung aus Aluminiumfolie angebracht, der direkte Infrarotstrahlung abschirmen sollte. Ein nennenswerten Erfolg brachte dies jedoch nicht. Dies war der Stand zu Beginn der Diplomarbeit.

## 10.2 Parameterermittlung

Zur Ermittlung von Parametern, die eine erfolgreiche Verlötung ohne Beschädigung der Platine ermöglichen wurden basierend auf den Erfahrungen weitere Tests durchgeführt. Alle Testverläufe erfolgten mit Platinenreststücken und teilweise darauf angebrachten SMD Widerständen zur Beurteilung der Lötstellen.

Zu Beginn wurden zunächst einige Testläufe mit den letzten Parametern (Tabelle 10.1) der damaligen Inbetriebnahme durchgeführt.

Um die Vermutung, dass die Platine durch Infrarotstrahlung stärker erhitzt wird als die Luft um den Temperatursensor, weiter zu untersuchen, wurde der Temperatursensor zusätzlich nicht frei im Luftraum platziert, sondern im Inneren eines Stück Platinenmaterials. Dazu wurde seitlich in ein Platinenstück ein Loch von 1,3mm Durchmesser gebohrt und der Sensor dort hinein platziert. Bei einer Gesamtstärke des Platinenmaterials von 1,5mm ergibt sich beidseitig eine Restdicke von 0,1mm, so dass nahezu die Oberflächentemperatur gemessen werden kann.

Die Steuerung des Ofens verfügt über die Möglichkeit die Temperaturdaten im Sekundentakt über eine serielle Schnittstelle auszugeben bzw. nach Aktivierung der Log-Funktion die Daten im internen EEPROM zu speichern und später abzurufen. Für den Testbetrieb ist es jedoch einfach die Daten direkt aufzuzeichnen, da die Log-Funktion nach jedem Reflowvorgang neu aktiviert werden muss.

Tabelle 10.1: Parameter 1. Durchlauf

<b>Prozessphase</b>	<b>Wert</b>
Preheatslope	2°C/s
Preheattemp	140°C
Soaktime	2:00min
Endtemp	155°C
Reflowtime	25sek
Reflowtemp	210°C

Die Testdurchläufe mit Parametern aus Tabelle 10.1 zeigten dabei folgendes Ergebnis:

- Leiterplattenoberfläche unverändert
- An faserigen Kanten leichte Braunfärbung
- Lötpaste einwandfrei geschmolzen

Um die thermischen Belastungen zu reduzieren wurde nun die Parameter nach Tabelle 10.2 abgeändert. Um bei der reduzierten Reflowtemperatur noch eine ausreichendes Schmelzergebnis zu erreichen, wurde die Reflowzeit etwas erhöht und um eine gleichmäßigere Durchwärmung zu erhalten, die Haltezeit erhöht. Im Ergebnis zeigte sich:

- Leiterplattenoberfläche unverändert
- An faserigen Kanten leichte Braunfärbung
- Lötpaste einwandfrei aber erst spät geschmolzen

Die Tatsache, dass sich auch noch bei derart niedriger Reflowtemperatur, die bereits nahe am Schmelzpunkt der Lötpaste liegt, noch immer Verfärbungen ergeben, ließ das gesamte Verfahren bereits als ungeeignet erscheinen. Für weitere Versuche wurden bei gleichen Parametern erneute Durchläufe durchgeführt, bei denen der Sensor einmal in einem Leiterplattenstück angebracht wurde und einmal bei dem der obere Heizstab durch ein Schild aus Aluminiumfolie abgeschirmt wurde. Der untere Heizstab wird durch eine Schale des Ofens standardmäßig immer abgeschirmt.

Die Ergebnisse zeigten keine Unterschiede zu Ergebnissen mit einem frei liegenden Sensor. Allerdings nimmt die maximale Aufheizgeschwindigkeit stark ab, so dass der Ofen eine deutlich längere Zeit benötigt um die Temperaturen zu erreichen, was anstatt einer geringeren Temperaturbelastung mehr zu einer höheren Belastung führt.

Tabelle 10.2: Parameter 2. Durchlauf

<b>Prozessphase</b>	<b>Wert</b>
Preheatslope	2°C/s
Preheattemp	120°C
Soaktime	2:30min
Endtemp	155°C
Reflowtime	30sek
Reflowtemp	190°C

Die Tatsache, dass die Lötpaste erst recht spät und träge schmilzt zeigt, dass die Reflowtemperatur noch zu niedrig ist und für eine erfolgreiche Lötverbindung erhöht werden muss. Als Vorlage für ein Profil wurde das empfohlene Profil für bleihaltige Bauelemente im Datenblatt des ADXL330 verwendet. Tabelle 10.3 zeigt die Parameter im Überblick.

Aus den empfohlenen Werten wurde nun das neue Profil in Tabelle 10.4 erstellt. Da weiterhin davon ausgegangen werden musste, dass der Ofen heißer wird als angezeigt, wurden die Werte etwas geringer gewählt.

Tabelle 10.3: Parametervorschlag aus Datenblatt des ADXL330

<b>Prozessphase</b>	<b>Wert</b>
Preheatslope	3°C/s
Preheattemp	100°C
Soaktime	60 – 120sec
Endtemp	150°C
Reflowtime	10 – 30sek
Reflowtemp	240°C

Tabelle 10.4: Erstellte Parameter aus Parametern des ADXL330 Datenblatts

<b>Prozessphase</b>	<b>Wert</b>
Preheatslope	2°C/s
Preheattemp	120°C
Soaktime	100sec
Endtemp	140°C
Reflowtime	15sek
Reflowtemp	235°C

Mit diesen Parametern erstellte Leiterplatten zeigten überraschenderweise keine größeren Verfärbungen als Leiterplatten aus Versuchen mit Parametern aus Tabelle 10.2. Versuchsweise wurde auch mit diesen Parametern noch mal der Schirm

verwendet. Hier zeigte sich noch deutlicher die zu geringe Temperaturerhöhung und den damit in die Länge gezogenen Lötvorgang.

Als letzte Anpassung wurde die Reflowtemperatur von 235°C auf 225°C gesenkt, dafür aber die Reflowzeit von 15sek auf 20sek erhöht, da die Lötpaste deutlich unter 235°C schmilzt und sich so ein ausgeglichener Temperaturverlauf ergibt.

Mit den in Tabelle 10.5 aufgeführten Parametern wurden nun erste Testverlötungen an BGA Bausteinen durchgeführt.

*Tabelle 10.5: Endgültige Parameter*

<b>Prozessphase</b>	<b>Wert</b>
Preheatslope	2°C/s
Preheattemp	120°C
Soaktime	100sec
Endtemp	140°C
Reflowtime	20sek
Reflowtemp	225°C

### 10.3 Probelötungen von BGA Bausteinen

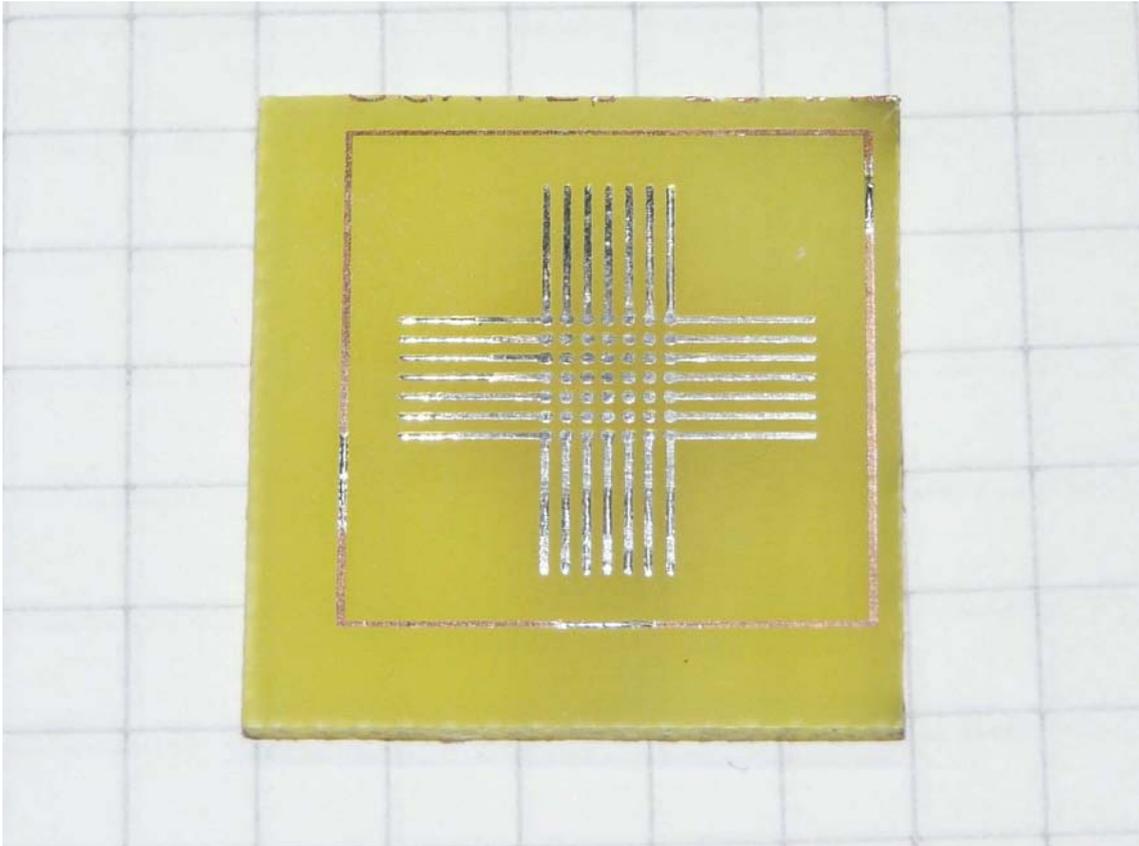
Um das Positionieren und Verlöten von BGA Bausteinen zu üben, war es sinnvoll Testbausteine zu beschaffen, um die teuren und schwer beschaffbaren Drehratsensoren zu schonen.

Der Markt bietet hierfür Dummybauteile an, die äußerlich normalen Bausteinen gleichen, jedoch keine Funktion besitzen. Für Testzwecke sind intern jedoch immer 2 Pins miteinander verbunden, so dass getestet werden kann ob alle Kontakte einwandfrei verlötet wurden und ob Kurzschlüsse zwischen einzelnen Pins vorliegen.

Hierzu wurde bei zwei Distributoren nach Bauteilen im 49-BGA (entspricht von der Größe den ADXRS Sensoren) angefragt. Jedoch scheiterte die Beschaffung an einer Mindestabnahme im Bereich von 400-500 Stück und dem damit verbundenen hohen Preis sowie an einer langen Lieferzeit von mehreren Wochen.

Als alternative wurde nun bei Maxim nach Bauteilen im 49-BGA Gehäuse gesucht, die als Muster lieferbar sind. Hierbei fiel die Wahl auf den DS2148, einen Line-Transceiver. Die eigentliche Funktion ist für den Test jedoch völlig unerheblich da sie nur mit großem Aufwand realisierbar gewesen wäre.

Als Testleiterplatte wurde eine einfache, einseitige Leiterplatte mit dem BGA Footprint erstellt, bei dem jeweils die außenliegenden Pins zur Kontrolle von Kurzschlüssen nach außen geführt wurden. Die innenliegenden Pins wurden nicht kontaktiert. Die Testplatine wurde selbst hergestellt, verzinkt und zerschnitten (Bild 10.1).



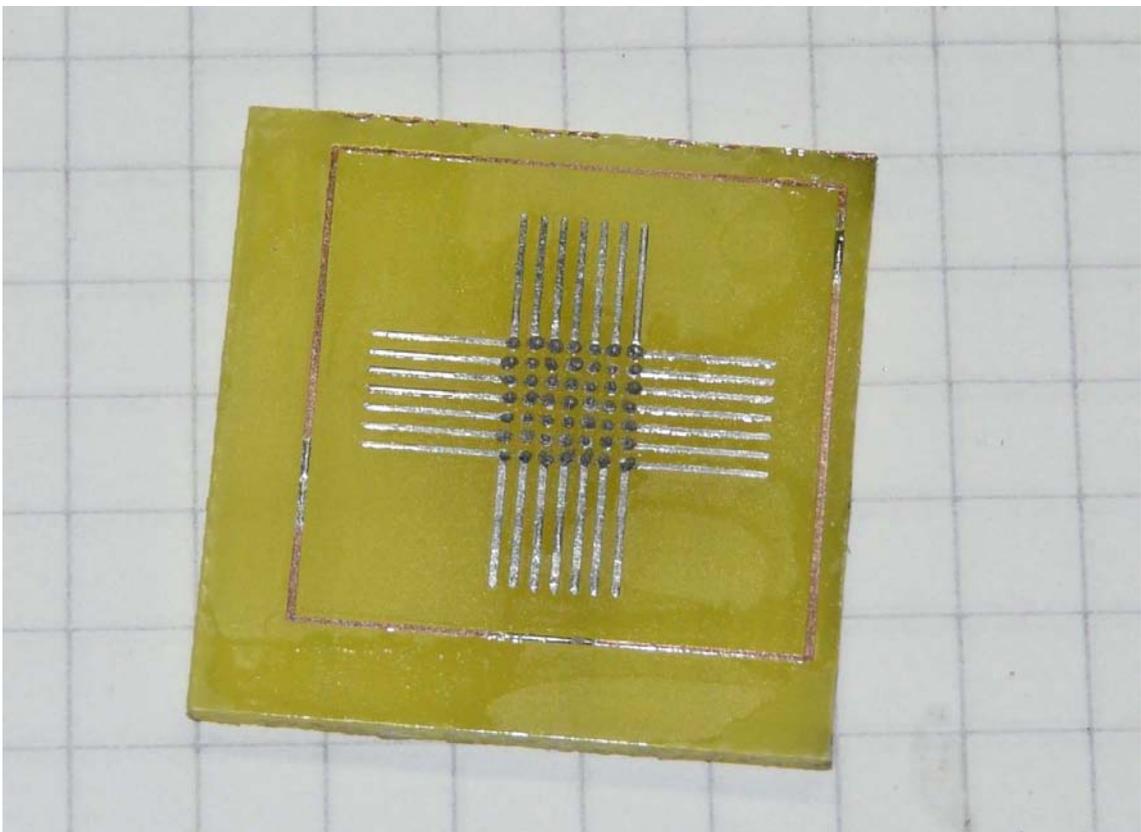
*Bild 10.1: Testplatine für BGA Lötversuche*

Bei der verwendeten Lötpaste handelt es sich um Bleihaltige Lötpaste des Typs CR-44. Bleihaltige Lötpaste hat den Vorteil eines geringeren Schmelzpunkts, sowie eine gewisse Selbstpositionierwirkung durch die Oberflächenspannung des flüssigen Lots. Sie ist daher bei nicht optimalen Bedingungen bleifreier Lötpaste vorzuziehen, soweit sie zulässig ist. Für Forschungszwecke ist der Einsatz z.B. zulässig.

Lötpaste ist recht empfindlich gegenüber zu hohen Lagertemperaturen und Überlagerung, da sie dabei ihre Konsistenz verliert und sich Flussmittel und Lotkugeln entmischen. Das Problem der zu hohen Konsistenz und der Entmischung trat auch bei der für dieses Projekt verfügbaren Lötpaste auf. Sie musste daher vor Verwendung erst wieder aufbereitet werden. Die Aufbereitung erfolgte indem etwas Lötpaste in einen alten Teelöffel gegeben wurde und mit einer Pipette ein Tropfen Isopropanol hinzugefügt wurde. Durch vorsichtiges Verrühren ließ sich so eine geeignete Konsistenz einstellen. Die Konsistenz muss so gewählt sein, dass sich auf den BGA Pads winzige Tropfen bilden lassen ohne das die Lötpaste zerfließt (zu flüssig) oder nicht am Pad haftet (zu zäh) (Bild 10.2). Aufgetragen wurde die Lötpaste mit einem 0,6mm dicken Silberdraht, der mit der Spitze in die Lötpaste getaucht wurde und anschließend auf das Pad getippt wurden (Bild 10.3). Da durch die Entmischung und Verdünnung der Lötpaste eventuell nicht mehr genug Flussmittel enthalten ist, sollte die Platine vorher mit einem dünnflüssigem Flussmittel benetzt werden (Schlieren auf der Platine in Bild 10.3), welches auch bei Wellenlötanlagen Verwendung findet.



*Bild 10.2: Aufbereitete Lötpaste*



*Bild 10.3: Testplatine mit auftragener Lötpaste*

Die Positionierung der BGA Bausteine kann nur nach Augenmaß erfolgen, wobei man auf möglichst gleichen Seitenabstand achten sollte. In dieser Arbeit wurden die Bauteile mit einer Pinzette dicht über der Platine in Position gebracht und fallen gelassen. Leichte Korrekturen sind aber später auch möglich.

Nach erfolgreichem Auftragen der Lötpaste und Positionieren der Bausteine wurde die Platine im Ofen mit den ermittelten Parametern verlötet. Die Verlötung war augenscheinlich gelungen, jedoch hatte sich die Platine deutlich verfärbt, die Ursache hierfür ist nicht klar, da sich eine Platine aus der gleichen Mutterplatine unter gleichen Parametern nicht verfärbte. Eventuell war eine zu dicke Flussmittelschicht verantwortlich, die beim Lötprozess verbrannt ist. Zur genaueren Untersuchung der Testplatine wurde diese in einem Isopropanolbad von Flussmittelresten gereinigt und die nebeneinander liegenden Pins mit einem Durchgangsprüfer auf Kurzschlüsse untersucht. Hierbei konnten jedoch keine Kurzschlüsse festgestellt werden. Eine optische Prüfung ergab, dass man gegen das Licht gesehen aus jeder Richtung zwischen den Pinreihen durchsehen konnte, so dass auch an den inneren Pins ein massiver Kurzschluss ausgeschlossen werden konnte. Eine feine Verbindung konnte so jedoch nicht ausgeschlossen werden.

Nach gleichem Prinzip wurde nun der Prozess mit einer weiteren Testplatine und einem weiteren Test-IC wiederholt. Das Flussmittel wurde nun etwas dünner aufgetragen und nach dem verlöten zeigten sich keinerlei Verfärbungen an der Platine. Auch an dieser Platine zeigte der Durchgangsprüfer und die optische Kontrolle keine Kurzschlüsse. Für eine genauere Untersuchung der Lötverbindung war es sinnvoll auch die inneren Lötverbindungen untersuchen zu können. Um diese zu untersuchen wurde versucht, das IC mechanisch von der Platine zu brechen. Hierbei brach idealerweise die Verbindung zwischen den Balls und dem Gehäuse, so dass nun eine Platine mit den verlöteten Balls und ein völlig ebenes IC vorlagen. Eine genaue Untersuchung der Platine mit den verlöteten Balls ergab an allen Pins eine einwandfreie Verlötung ohne Kurzschlüsse (Bild 10.4).

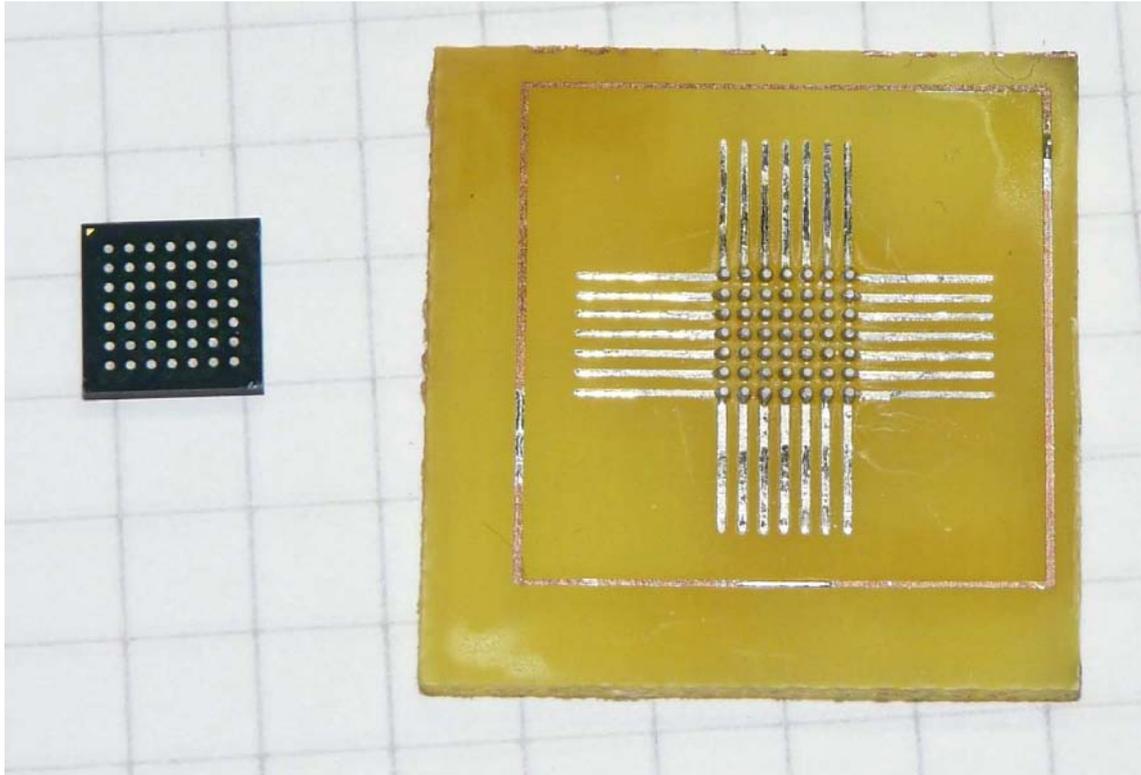


Bild 10.4: Nach verlöten wieder entferntes BGA IC

#### 10.4 Endgültiger Lötprozess

Die erste Verlotung erfolgte mit einem LISY300AL auf der Z-Gyro Platine, da der LISY300AL noch relativ einfach und günstig beschaffbar ist und die Gyro-Z Platine in viel größere Stückzahl geliefert wurde als eigentlich benötigt. Sollten der Reflowprozess eventuell doch Bauteile oder Platinen schädigen wäre der Schaden nicht so gravierend.

Die Verarbeitung erfolgte, wie bereits beschrieben, beginnend mit dem Auftragen einer dünnen Schicht Flussmittel und anschließend mit dem Auftragen der Lötpaste. Da die Pads für den LISY300AL recht groß sind, bereitete dies ebenso wie die Positionierung kaum Schwierigkeiten (Bild 10.5 oben). Nach Durchlauf durch den Ofen war der Gyro augenscheinlich fest sauber verlötet (Bild 10.5 unten). Eine Durchgangsprüfung ergab keine Kurzschlüsse. Nach Bestücken der restlichen Bauteile per Hand erfolgte ein Anlegen der Betriebsspannung unter Strombegrenzung. Eine Messung des Ausgangssignal zeigte einwandfreie Funktion.

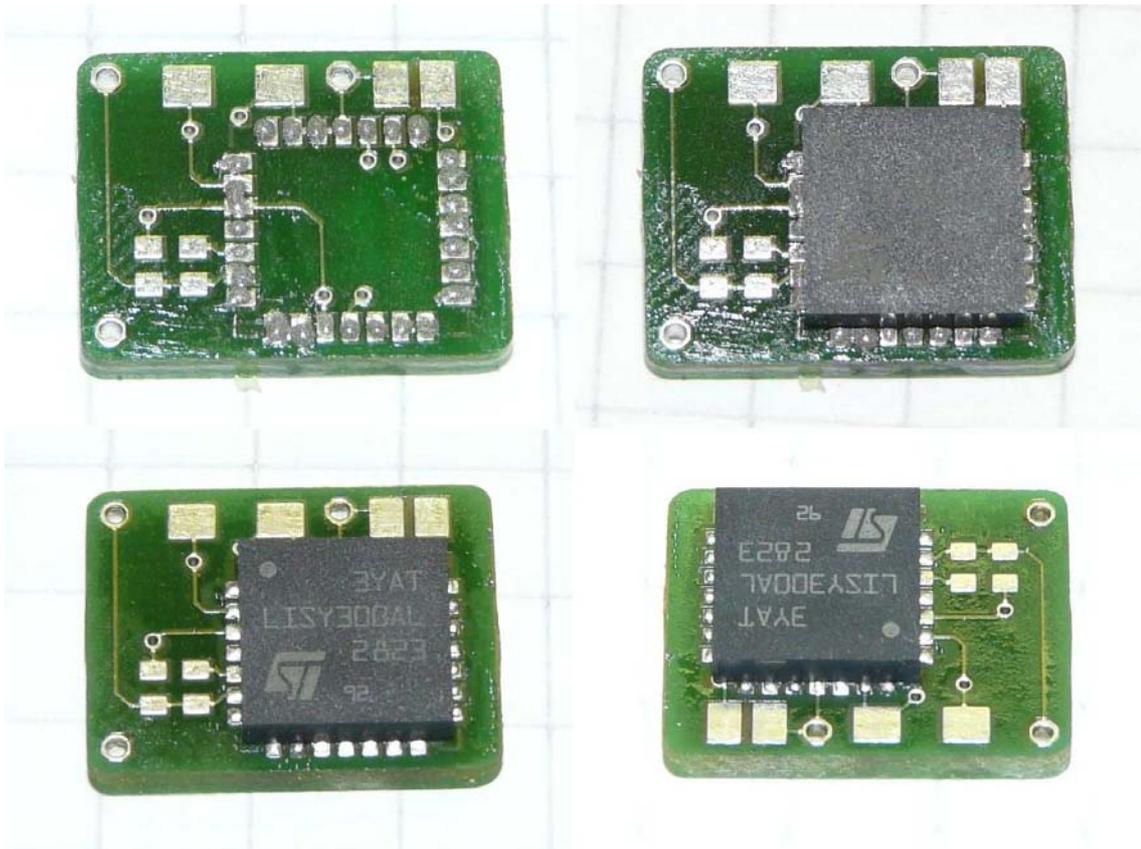


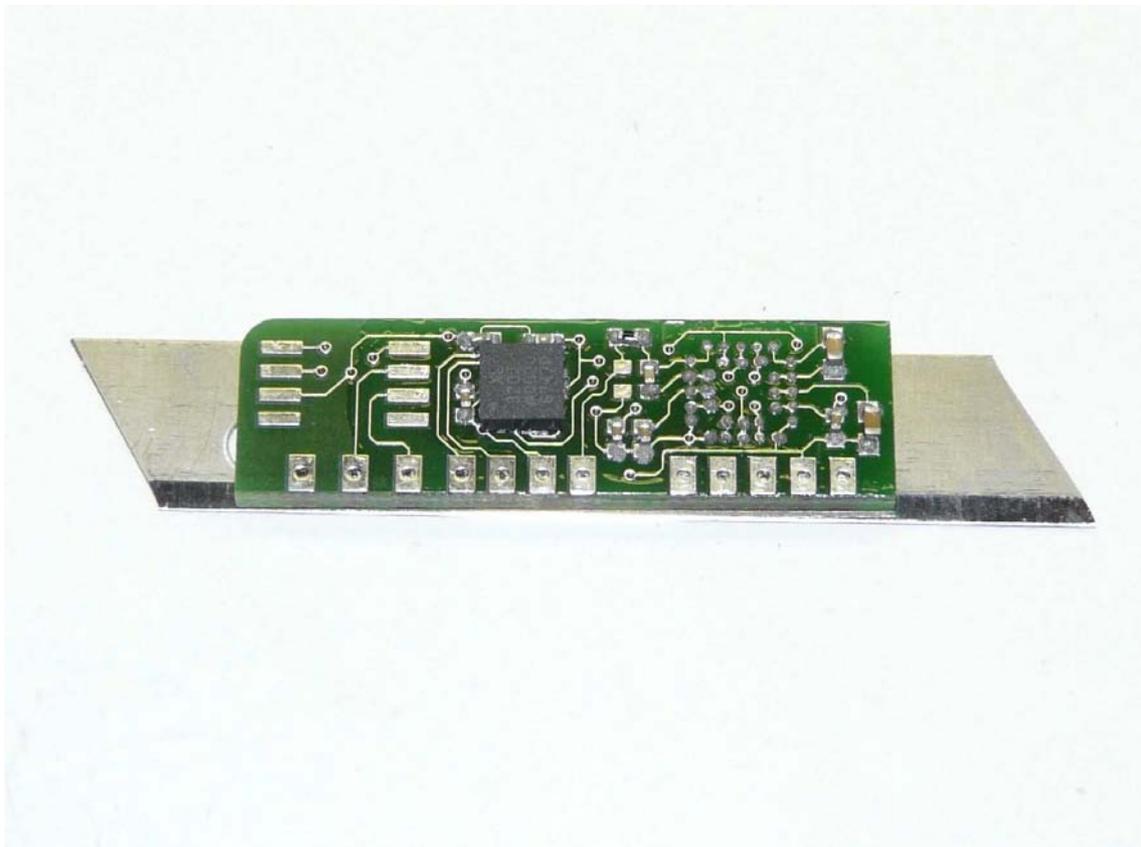
Bild 10.5: LISY300A1, oben links: Paste aufgetragen, oben rechts: Platziert, unten: Verlötet

Im nächsten Schritt wurde ein ADXRS610 mit gleicher Vorgehensweise ebenso auf einer Z-Platine bestückt. Das Auftragen der Lötpaste war hier aufgrund der geringen Padgröße etwas schwieriger und erforderte etwas Übung. Nach Durchlauf durch den Ofen, anschließender Bestückung der restlichen Bauteile per Hand und Inbetriebnahme zeigte sich jedoch keinerlei Reaktion auf Drehbewegungen. Das Ausgangssignal befand sich exakt auf  $VDD/2$ . Eine Kontrolle des Temperatursignales zeigte jedoch eine Reaktion auf eine Temperaturänderung, so dass die Grundfunktionalität des Gyros gegeben sein musste. Eine Kontrolle der Signale mittels Oszilloskop an den Kondensatoren der Ladungspumpen ergab jedoch nur ein Massepegel. Ein Vergleich mit den entsprechenden Signalen auf der Testplatine ergab, dass hier eigentlich Rechtecksignale anliegen müssten. Da ein Bestückungsfehler durch die Funktion des Temperatursignales ausgeschlossen werden konnte, war eine fehlende Verbindung eines BGA Pins die wahrscheinlichste Ursache. Hierbei spielte es eventuell eine Rolle, dass die Gyroskope nur in bleifreier Ausführung vorlagen und durch das bleifreie Lot die Temperatur eventuell nicht für eine saubere Verbindung aller Pins ausgereicht hat. Zur Reparatur wurde daher ein zweiter Reflowdurchgang bei um  $5^{\circ}\text{C}$  erhöhter Reflowtemperatur vorgenommen. Um eine Verbindung zu erleichtern wurde neues Flussmittel aufgetragen. Nach erneutem Durchlauf zeigte das Gyroskop schließlich einwandfreie Funktion. Hieraus schließt sich, dass bleihaltigen Bauteile bei  $225^{\circ}\text{C}$  verlötet werden können, bleifreie Bauteile jedoch bei  $230^{\circ}\text{C}$  verlötet werden sollten.

Folgend wurde nun die X-Platine bestückt, bei der auch die Bauteile der Oberseite bereits im Reflowverfahren mit verlötet werden sollten. Hierzu wurde auf die Pads ein wenig Lötpaste aufgetragen und die Bauteile in die Lötpaste gedrückt. Nach dem Verlöten und Handbestücken der Bauteile auf der Unterseite zeigte sich bei der Inbetriebnahme einwandfreie Funktion.

Bei der Y-Platine war zusätzlich noch der Beschleunigungssensor ADXL320 mitzubestücken, der ebenfalls nach für die Gyroskope bekannten Prinzip bestückt wurde. Die Bauteile der Oberseite wurden ebenfalls im Reflowprozess mitverlötet. Auf Bild 10.6 und Bild 10.7 ist die Y-Sensorplatine mit bestückten Bauteilen und noch unverlötet, sowie verlötet dargestellt. Auch diese Platine zeigte inklusive des Beschleunigungssensors einwandfreie Funktion.

Bild 10.7 zeigt einen fertigen Satz an Sensorplatinen mit dem ADXRS610, sowie eine Z-Platine mit dem LiSY300AL.



*Bild 10.6: Y-Sensorplatine, bestückt noch nicht verlötet*

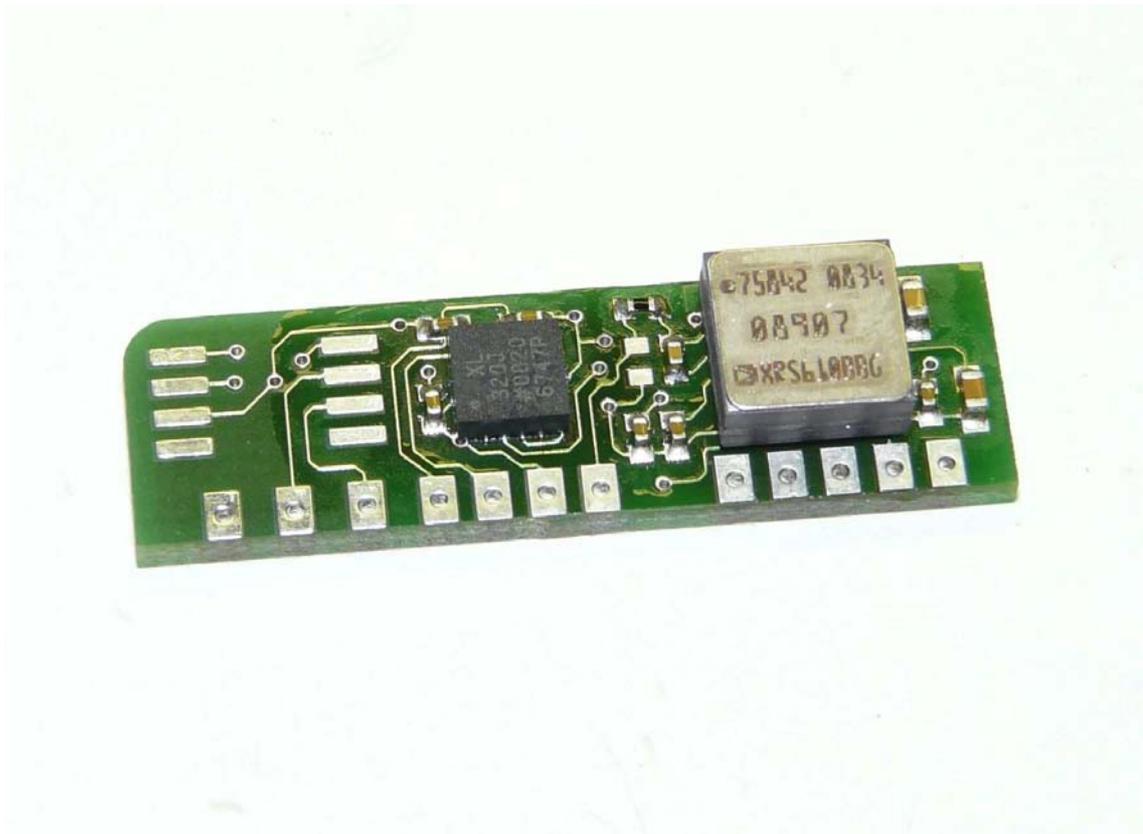


Bild 10.7: Y-Sensorplatine, verlötet



Bild 10.8: Fertig bestückter Sensorplatinensatz

## 11 Die Bestückung

Das Bestücken ist, abgesehen von den Bauteilen, die zwingend in einem Reflowprozess gelötet werden müssen und der geringen Baugröße vieler Bauteile, relativ leicht durchzuführen. Alle Bauteile, die auf der gleichen Seite liegen wie die Bauteile, die mit Reflow verlötet werden müssen, können ebenso per Reflow verlötet werden. Dazu wird eine geringe Menge Lötpaste auf die Pads aufgetragen, und die Bauteile in die frische Lötpaste eingedrückt (Bild 11.1). Es ist nicht sinnvoll erst alle Pads mit Lötpaste zu versehen und anschließend alle Bauteile einzudrücken, da die Lötpaste sehr schnell austrocknet und die Bauteile dann nicht mehr durch ihre Klebewirkung festhält. Unter Umständen kann es auch zu einem Abplatzen der Lötpaste kommen. Vor allem passive Bauteile und auch die SMD Molexstecker lassen sich so schnell und sauber verlöten. Auch integrierte Schaltkreise mit größerem Pinabstand sind kein Problem. Als Problem stellte sich aber der LPC2148 mit einem Pitch von 0,5mm heraus, da es nicht möglich war die Lötpaste vernünftig auf den sehr schmalen aber langen Pads aufzutragen. Mit der „Drahtauftupfmethode“ kann die Paste lediglich rundlich aufgetragen werden, was bei den geringen Pinabständen immer wieder zu Kontakt zwischen zwei Pins führte. Während des Lötprozess bildeten sich so einige Lötbrücken (Bild 11.2). Einige davon waren gut sichtbar und konnten entfernt werden. Andere bildeten sich jedoch auf der Rückseite der Pins und waren so optisch von der Vorderseite kaum zu erkennen und konnten nur durch Hinzufügen von frischem Lötzinn und anschließendem Absaugen entfernt werden.

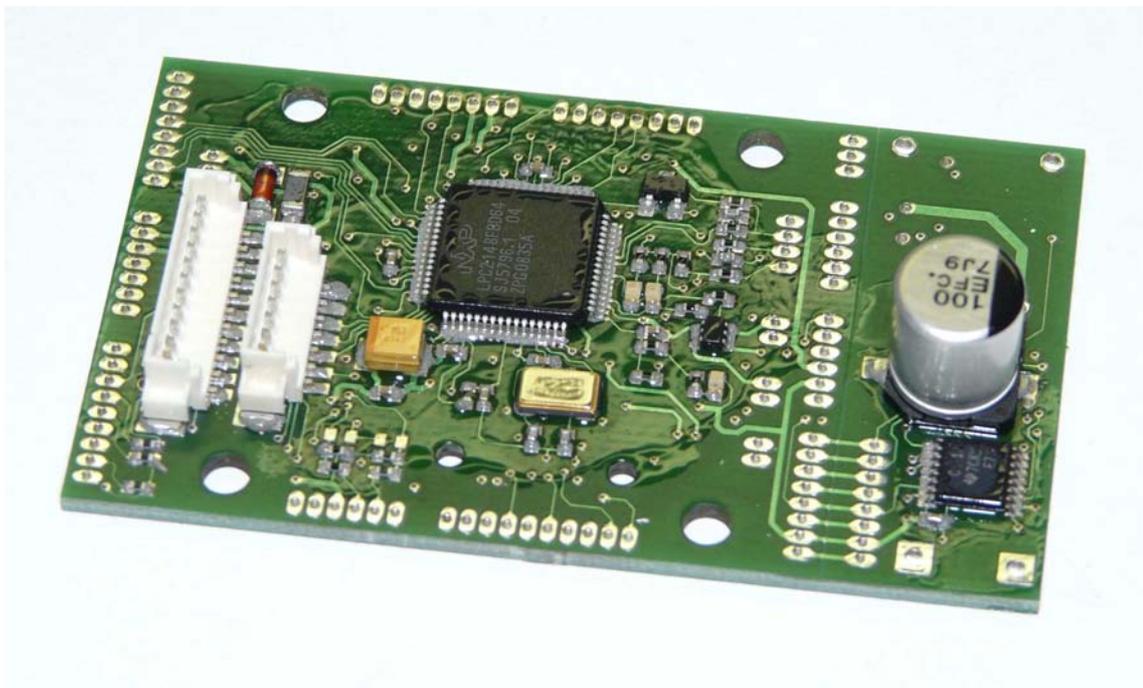
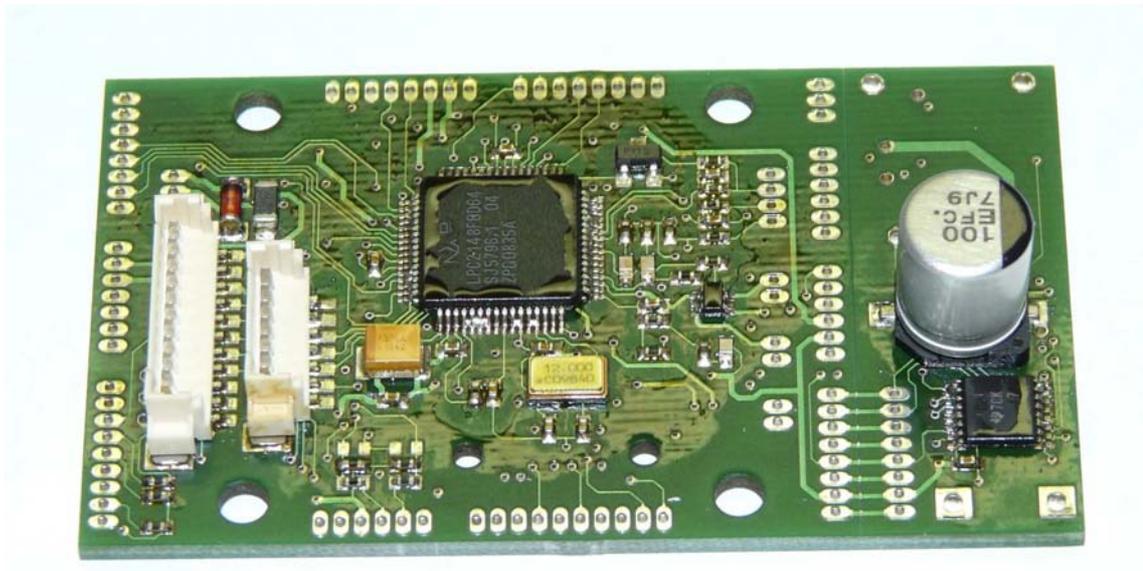


Bild 11.1: MCU Platine vor dem Reflowprozess



*Bild 11.2: MCU Platine direkt nach dem Reflowprozess*

Alle Bauteile auf der Unterseite müssen per hand bestückt werden. Für die kleinen 0402 Bauteile ist teilweise durch die Heißverzinnung des Leiterplattenherstellers genügend Zinn auf der Leiterplatte vorhanden, um die Bauteile, die mit Hilfe einer Pinzette positioniert werden, einseitig zu fixieren. Die unverlötete Seite kann nun unter zur Hilfenahme von sehr feinem Lötzinn (0,32mm) verlötet werden. Anschließend sollte dies auch mit der vorher nur fixierten Seite passieren. Die Verwendung von Flussmittel kann das Löten vereinfachen und die Qualität der Lötstellen erheblich erhöhen. Nach Beendigung der Bestückung ist die Leiterplatte in wasserfreiem Isopropanol zu waschen. Dabei ist zu beachten, dass nicht alle Bauteile gewaschen werden dürfen (Luftdrucksensoren), diese Bauteile sind nach dem Waschprozess zu bestücken. Auf Bild 11.3 - Bild 11.8 sind die fertig bestückten Platinen mit Ober und Unterseite, sowie das zusammengesteckte Modul zu sehen.

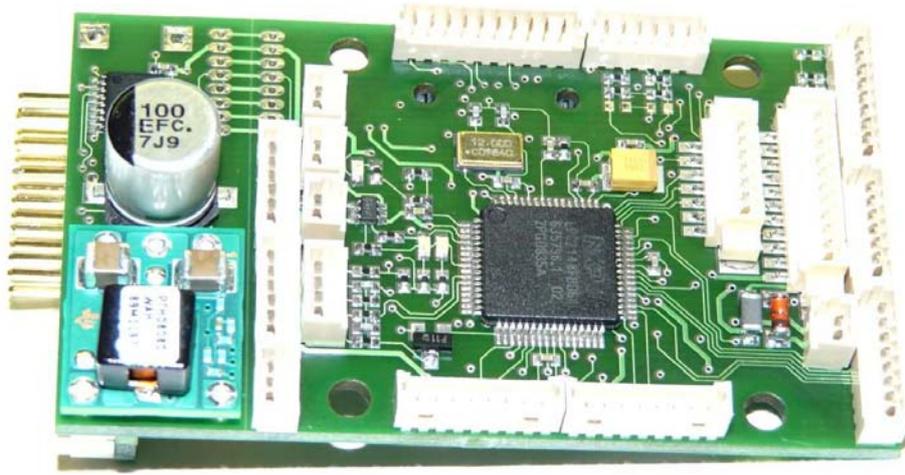


Bild 11.3: Oberseite MCU

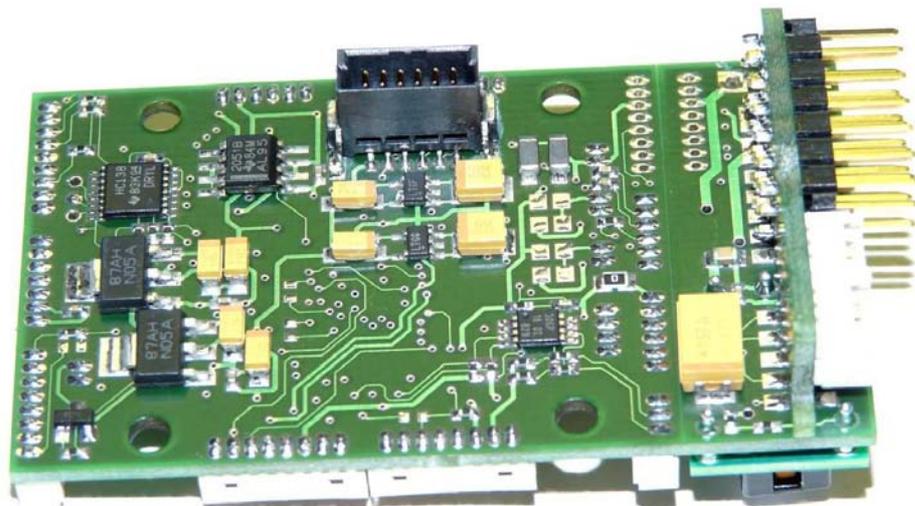


Bild 11.4: Unterseite MCU



Bild 11.5: Oberseite IMU

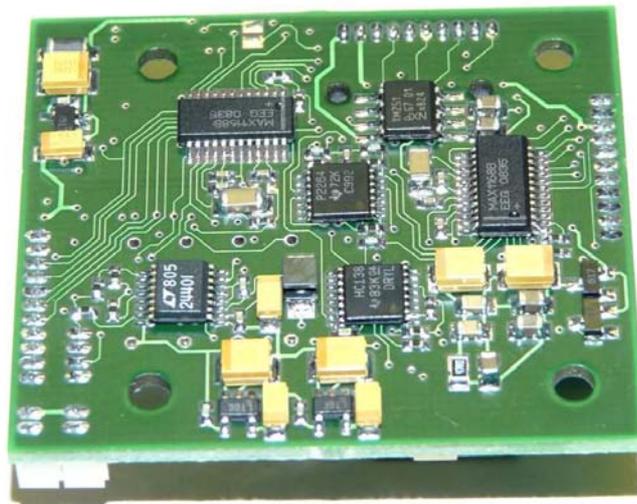
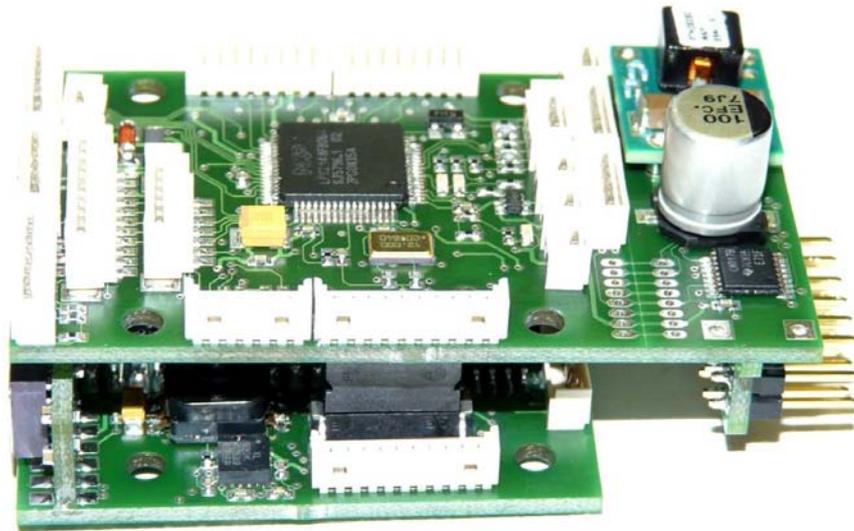


Bild 11.6: Unterseite IMU



*Bild 11.7: MCU und IMU gesteckt*



*Bild 11.8: MCU und IMU gesteckt*

## 12 Zusatzmodule

### 12.1 Motorsteller

Bei den zu verwendeten Motorstellern handelt es sich um die Brushlesscontroller von Holger Buss und Ingo Busker in der Version 1.2 [12].

Die Bestückung erfolgte nach Anleitung des Herstellers ohne besondere Vorkommnisse.

Nach Beendigung der Bestückung wurden die Motorsteller nach Anleitung des Herstellers mit der Motortestsoftware programmiert und anschließend in Betrieb genommen. Es zeigte sich hier bei allen vier Stellern einwandfreie Funktion, so dass im Anschluss die Betriebsfirmware in der Version 0.41 einprogrammiert wurde.

Nach Anlöten der Kabel für Betriebsspannung und Motorphasen wurden die Platinen in Schrumpfschlauch eingeschrumpft und montiert.



Bild 12.1: Bereits eingeschrumpfter und am Quadrokofter montierter Motorsteller

### 12.2 Das GPS-Modul

Für den aufzubauenden Quadrokofter sollte ein uBlox GPS Module vom Typ SAM-LS verwendet werden. Bei dem Modul handelt es sich um eine Ausführung, die normal für den Anschluss einen Folienkabels vorgesehen ist. Anstatt über Folienkabel ist jedoch auch ein Anschluss über Adapterplatinen auf eine Trägerplatinen möglich (Bild 12.3).

Dieses Modul wurde bereits in einem früheren Projekt verwendet, so dass die in diesem Projekt verwendete Adapterplatine lediglich auf neue Erkenntnisse abgewandelt wurde. Hierzu gehört der Austausch der Lithiumbatterie, die nur eine geringen Lebensdauer aufwies, gegen einen Gold Cap Kondensator mit zugehöriger Ladeschaltung (Bild 3.1).

Der Aufbau erfolgte als 3,3V Variante, so dass ein Großteil der Bauteile, unter anderem der Spannungsregler, nicht zu bestücken waren.



Bild 12.2: GPS-Modul



Bild 12.3: Verbindung GPS-Modul mit Trägerplatine

### 12.3 Funkmodul

Die Realisierung der Funkstrecke zur Telemetrieübertragung erfolgt mittels Radiotronic Funkmodulen, die bereits fertig vorhanden waren. Hier erfolgte nur die Anfertigung eines Adapterkabels zum Anschluss an den Autopilot sowie die Einstellung auf einen eigenen Funkkanal. Als Kanal wurde 34 eingestellt.

### 12.4 Programmieradapter

Um den Mikrocontroller auf dem fertigen Autopiloten auch ohne Bootloader zu programmieren bzw. um den Bootloader einspielen zu können, ist ein Programmieradapter notwendig, der den Mikrocontroller über UART0 mit dem PC verbindet. Des Weiteren muss zum Einleiten des Programmiermodus der Pin P0.14 auf

Masse gezogen werden und hiernach ein Reset durchgeführt werden. Der Programmieradapter bietet hierzu die Möglichkeit den Pin P0.14 per Schalter auf Masse zu ziehen, sowie über einen Taster einen Reset durchzuführen.

Beide seriellen Steckverbinder ermöglichen mit dem selben Kabel den Anschluss des Programmieradapters, so dass es auch im Normalbetrieb möglich ist, Daten über die seriellen Schnittstellen an einen PC zu übertragen. Ein Reset ist dabei an beiden seriellen Steckern durchführbar, ein Programmieren jedoch nur über UART0. Die Möglichkeit einen Reset über den Programmieradapter ist auch während des Flashens des Mikrocontroller über USB hilfreich, da hier vor jedem Upload ein Reset ausgeführt werden muss.

Aufgebaut ist der Programmieradapter mit dem bekannten Pegelwandlerbaustein MAX232 und der bekannten, im Datenblatt angegebenen, Beschaltung, sowie dem Schalter für den Programmierpin, dem Taster für den Resetimpuls, einer D-SUB Buchse für den seriellen Anschluss und einem Molexstecker zur Verbindung mit dem Autopiloten.



Bild 12.4: Programmieradapter

## 13 Inbetriebnahme

### 13.1 Kontrolle Versorgungsspannungen

Um im Falle von Designfehlern einen Schaden teurer Bauteile zu vermeiden erfolgte als erstes die Bestückung der Spannungsregler und ihren Peripheriebauteilen.

Nach Bestückung erfolgte das Anlegen der Betriebsspannung unter Strombegrenzung und Kontrolle der ausgangsseitigen Versorgungsspannungen, sowie aller Pads von Schaltkreisen, die Versorgungsspannung führen mittels Multimeters.

Hierbei wurde kein Fehler festgestellt, alle Versorgungsspannungen lagen im erlaubten Toleranzbereich.

*Tabelle 13.1: Kontrolle der Versorgungsspannungen*

<b>Versorgungsspannung</b>	<b>Soll in V</b>	<b>Ist in V</b>
MCU 3,3V analog	3,3	3,286
MCU 3,3V digital	3,3	3,3
MCU 5,0V analog	5,0	5,03
MCU 5,0V digital	5,0	5,04
IMU 3,3V analog	3,3	3,312
IMU 3,3V digital	3,3	3,323
IMU 5,0V analog	5,0	4,98
IMU 5,0V digital	5,0	5,02

## 13.2 Analoge Signale der Sensoren

Da die Beschleunigungs-, sowie Drehratensensoren in löstechnisch kritischen Gehäusen untergebracht sind, erfolgte nach dem Bestücken dieser Sensoren umgehend ein Funktionstest durch Ausmessen der analogen Ausgänge.

Hierzu wurden die für den Sensor benötigten Betriebsspannungen angelegt und bei den Beschleunigungssensoren die Reaktion auf die Erdanziehung überprüft, sowie ob gemessene Wert für  $1g$  in etwa mit den Datenblattangaben übereinstimmte. Die Drehratensensoren wurden für den Test per Hand einer Drehbewegung ausgesetzt und überprüft ob hieraus eine Reaktion der Ausgangsspannung erfolgte. Da sämtliche Drehratensensoren auf Zusatzplatinen montiert sind erfolgte diese Tests bereit mit den Einzelplatinen und nicht erst im fertigen Modul.

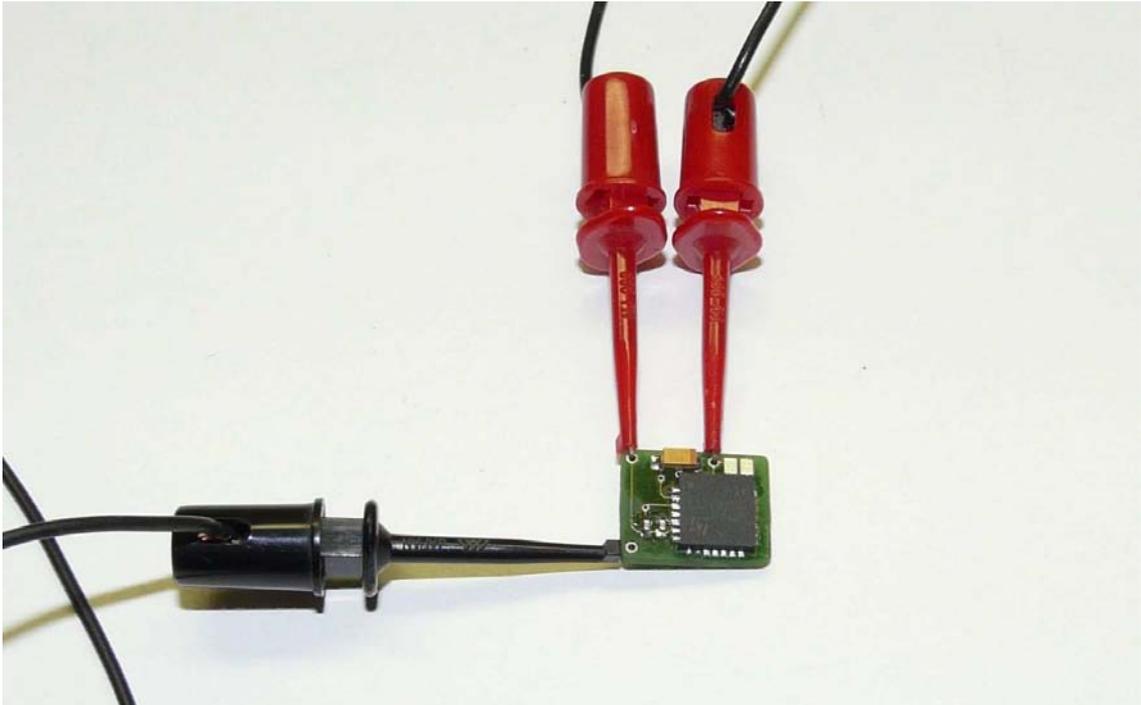


Bild 13.1: LISY300AL im Funktionstest

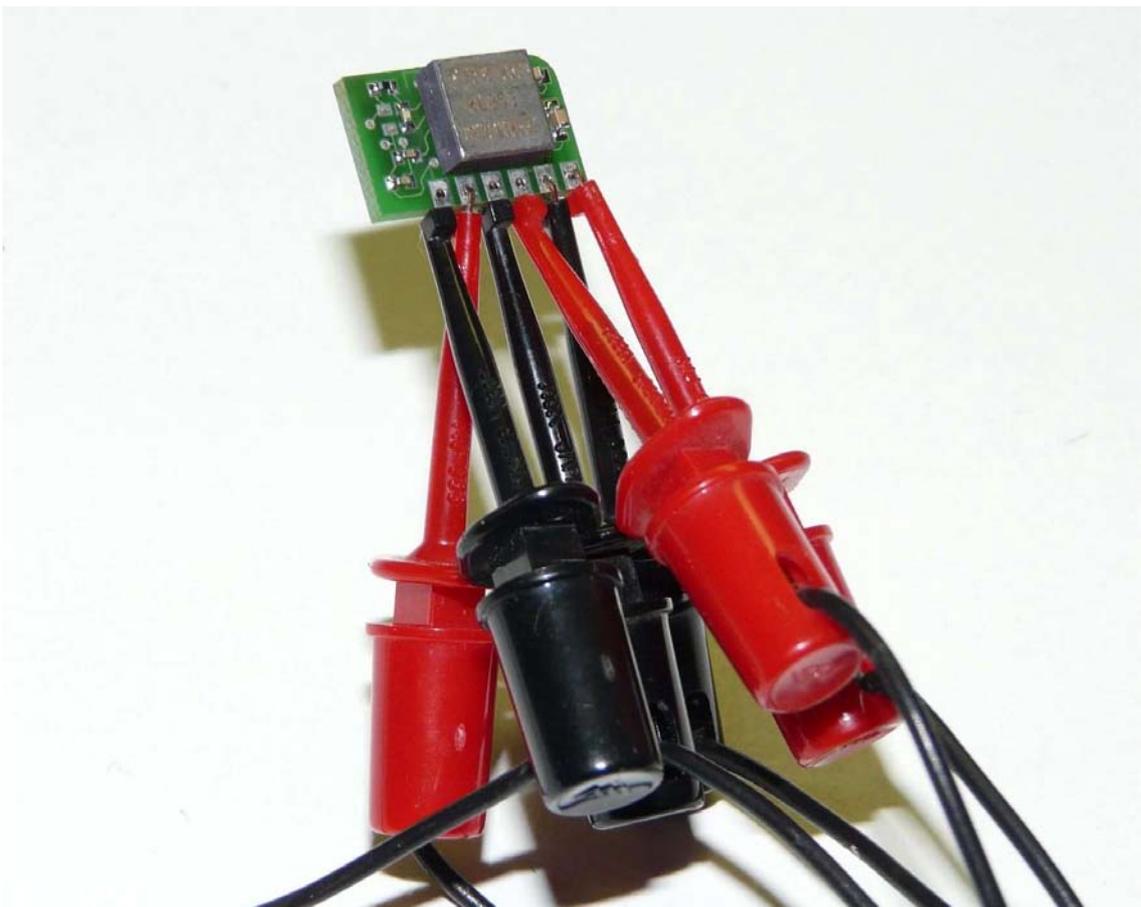


Bild 13.2: ADXRS610 im Funktionstest

## 14 Kabelpläne

### 14.1 K-01 MCU – IMU

Tabelle 14.1: Kabelplan K-01

<b>Pin X105</b>	<b>Signal</b>	<b>Farbe</b>	<b>Pin X404</b>
1	GND	br/ws	1
2	GNDA	bl/ws	2
3	V+	or/ws	3
4	MOSI0	ws/bl	4
5	MISO0	ws/gr	5
6	SCK0	ws/vi	6
7	CS_0	ws/ge	7
8	CS_1	ws/or	8
9	CS_2	ws/rs	9
10	MAG_IO	vi/ws	10

### 14.2 K-02 MCU – Programmieradapter

Tabelle 14.2: Kabelplan K-02

<b>Pin X109</b>	<b>Signal</b>	<b>Farbe</b>	<b>Pin Programmer</b>
1	GND	br/ws	1
2	+5V	ge/ws	2
3	+3,3V	rs/ws	3
4	TXD	ws/ge	4
5	RXD	ws/or	5
6	-	-	6
7	Reset	ws/bl	7
8	P0.14	ws/gn	8

### 14.3 K-03 MCU – Motorsteller

Tabelle 14.3: Kabelplan K-03

<b>Pin X111</b>	<b>Signal</b>	<b>Farbe</b>	<b>Pin Motorsteller</b>
1	GND	ws/br	1
2	SDA0	vi/ws	2
3	SCL0	ws/vi	3

### 14.4 K-04 MCU – USB-Buchse

Tabelle 14.4: Kabelplan K-04

<b>Pin X108</b>	<b>Signal</b>	<b>Farbe</b>	<b>Pin USB Buchse</b>
1	V+_USB	gn/ws	1
2	D-	hbr/ws	2
3	D+	ws/hbr	3
4	GND	br/ws	5

### 14.5 K-05 MCU – GPS

Tabelle 14.5: Kabelplan K-05

<b>Pin X109</b>	<b>Signal</b>	<b>Farbe</b>	<b>Pin GPS</b>
1	GND	br/ws	1
2	+5V		
3	+3,3V	ge/ws	2
4	TXD	ws/ge	4
5	RXD	ws/or	3
6	IO		
7	Reset		
8	P0.14		

## 14.6 K-06 MCU – RS Empfänger

Tabelle 14.6: Kabelplan K-06

<b>Pin X113</b>	<b>Signal</b>	<b>Farbe</b>	<b>Pin RC_Empfänger</b>
1	GND	br/ws	1
2	+5V	ge/ws	2
3	+3,3V		
4	RC_PPM_IN	hbr/ws	3
5	Servo_RST		
6	Servo_CLK		

## 14.7 K-07 MCU – MCU – Radiotronix

Tabelle 14.7: Kabelplan K-07

<b>Pin X109</b>	<b>Signal</b>	<b>Farbe</b>	<b>Pin GPS</b>
1	GND	br/ws	2
2	+5V		
3	+3,3V	rs/ws	1
4	TXD	ws/ge	4
5	RXD	ws/or	3
6	IO	ws/bl	6
7	Reset		
8	P0.14		

# 15 Softwareanpassung

Die Steueralgorithmen für den Flugbetrieb eines Quadropters wurden als Hilfsmittel für diese Arbeit bereits fertig zur Verfügung gestellt. Aufgabe war es jedoch noch die Software an die Gegebenheiten der neuen Hardware anzupassen.

Die zur Verfügung gestellte Software wurde auf einer Testhardware, basierend auf dem Paparazzi Tiny 13 erstellt. Zwar wurde beim Entwurf der neuen Hardware versucht, möglichst viele Pins mit der gleichen Funktion zu belegen wie beim Tiny 13, jedoch war dies teilweise nicht möglich, da Pins durch Schnittstellen belegt waren, die das Tiny 13 nicht verwendet, die aber bei der neuen Hardware gefordert waren.

Als Hauptaufgabe galt das Schreiben der SPI Treiber um die Sensordaten im Unterschied von der Tiny-Variante nicht über die Internen ADC's zu lesen sondern über externe ADC's einzulesen, welche über den SPI Bus angesprochen werden.

Alle Softwareänderungen sind so durchzuführen, dass die Software durch Schalter zwischen der Tiny (Tiny13) und der neuen Hardwarevariante (HBC) umstellbar ist und so auf beiden Systemen gleichermaßen funktionstüchtig ist - vergleiche Programmtext auf folgender Seite. So ist es möglich durch Setzen von *#define TINY13* bzw. *HBC* zwischen beiden Varianten umzuschalten ohne das weitere Programmtextänderungen notwendig sind. Zukünftige Erweiterungen und Änderungen sollte diesem Schema folgen.

## 15.1 Fernsteuerung

Für die Anpassung der Fernsteuerung an die neue Hardware musste lediglich eine Pinconfiguration geändert werden, da es möglich war, den Fernsteuereingang ebenfalls an einen *CAP0.2* Pin zu legen. Der LPC2148 verfügt über gesamt 3 *CAP0.2* Pins, die in den *PINSEL* Registern ausgewählt werden können. Es darf immer nur einer der drei Pins als *CAP0.2* Pin konfiguriert sein. Sind mehre Pins als *CAP0.2* konfiguriert, tritt eine Priorisierung in Kraft und nur der am höchsten priorisierte Pin ist aktiv.

Das Tiny verwendete den Pin *P0.6* als *CAP0.2* Pin, die neue Hardware nutzt *P0.16* als *CAP0.2* Port. Für die Anpassung war es daher nur notwendig, in der *fc.c* die Konfiguration von *P0.6* zu entfernen und die von *P0.16* hinzuzufügen.

---

```
#ifndef TINY13

    // P0.6 as capture 0.2 for timer0
    PINSEL0 |= (1<<13);
    PINSEL0 &= ~(1<<12);
#endif //TINY13

#ifndef HBC

    // P0.16 as capture 0.2 for timer0
    PINSEL1 |= (1<<0)|(1<<1);

#endif //HBC
```

---

## 15.2 Interne Analog-Digital-Converter

Die internen Analog-Digital-Umsetzter wurden ähnlich zum Tiny verwendet und belegt, um möglichst einheitliche Systeme zu erhalten. Lediglich 2 Ports konnten nicht identisch belegt werden, so dass hier eine Anpassung der Steuersoftware erforderlich war.

Alle Ports des *ADC0* sind zwischen Tiny und neuem Autopiloten identisch, so dass hier keine Anpassung erforderlich war. An *ADC1* kam Port *AD1.2* hinzu während Port *AD1.7* wegfiel. Dies erforderte die zusätzliche Konfiguration von Pin *P0.10* im *PINSEL0* Register als ADC. Die Konfiguration von Pin *P0.22* als ADC im *PINSEL1* Register fiel dagegen weg.

Konfiguration der *PINSEL* Register für ADC1 Tiny:

---

```
// switch pins for ADC1: ad1.2, ad1.3,ad1.4,ad1.5
PINSEL0 |= (3<<20)|(3<<24)|(3<<26)|(3<<30);

// switch pins for ADC1 an Pin: ad1.6;
PINSEL1 |= (1<<11);
PINSEL1 &= ~(1<<10);
```

---

Konfiguration der *PINSEL* Register für ADC1 HBC:

---

```
// switch pins for ADC1: ad1.3,ad1.4,ad1.5
PINSEL0 |= (3<<24)|(3<<26)|(3<<30);

// switch pins for ADC1 an Pin: ad1.6,ad1.7;
PINSEL1 |= (1<<11)|(1<<12);
PINSEL1 &= ~((1<<10)|(1<<13));
```

---

Zusätzlich war eine Änderung im Register *AD1CR* notwendig, welches unter anderem bestimmt, welche Kanäle einen Interrupt auslösen können. Die hier angegebene Maske war so zu verändern, dass Kanal 7 zu entfernen war und Kanal 2 hinzu kam.

*AD1CR* neu:

---

```
AD1CR |= 0x7C; // channel 7 (6 5 4) (3 2) 1 0
```

---

*AD1CR* alt:

---

```
AD1CR |= 0xF8; // channel (7 6 5 4) (3) 2 1 0
```

---

Zu beachten ist, dass die Werte der internen ADC noch in die ursprünglichen Arrays geschrieben werden, welche für das Tiny gebildet wurden. Verwendet werden diese Werte jedoch im jetzigen Zustand nicht. Für eine Verwendung der internen ADC ist es daher angebracht, die Arrays auf die neuen Gegebenheiten anzupassen.

## 15.3 SPI Treiber

Alle Sensoren sind durch insgesamt drei Analog-Digital-Umsetzer über den SPI Bus mit dem Mikrocontroller verbunden. Die ADC's müssen entsprechend ihrer Anforderungen vom Mikrocontroller angesprochen und ausgelesen werden.

Um Rechenzeit einzusparen erfolgt die Steuerung des SPI Bus über einen Hardwareinterrupt. Die standardmäßigen Routinen zum Senden und Empfangen von Daten über den SPI Bus warten mittels einer *while-Schleife* bis die Datenübertragung auf dem Bus abgeschlossen ist. Die Zeit, in der auf den Abschluss der Datenübertragung gewartet wird, steht dabei nicht für andere Aufgaben zur Verfügung. Da Rechenzeit nicht im Überfluss vorhanden ist und das Auslesen der ADC's über den SPI Bus zeitlich sehr oft geschieht, ist die Interrupt-Steuerung zwingend erforderlich.

Der die Kommunikation des SPI Bus durchführende Algorithmus setzt sich aus zwei Zustandsautomaten zusammen, wobei der erste als Interruptservice-Routine aufgerufen wird und das Lesen und Schreiben auf dem Bus kontrolliert. Der zweite Zustandsautomat wird vom ersten aufgerufen und teilt dem ersten mit was gelesen und geschrieben werden soll.

### 15.3.1 Die Funktion „spi\_init“

Die Funktion *spi\_init* übernimmt die Initialisierung der SPI Schnittstelle und setzt alle Einstellungen. Sie wird in main.c im Initialisierungsbereich aufgerufen

Der Reihe nach werden in dieser Funktion die Pin-Konfigurationen, die Einstellungen der SPI Schnittstelle, sowie die Einstellungen für den Interrupt vorgenommen.

Konfiguration der ChipSelect Pins als Ausgänge [D8, S.84 Tab.68]

---

```
IO1DIR |= (1<<21)|(1<<22)|(1<<23); //CS pins as output
```

---

Konfiguration des Ausgang zur Ansteuerung der Entmagnetisierspulen als GPIO und als Ausgang [D8, S.76 Tab.67].

---

```
//PIN for MAG_FLIP as GPIO / Direction = Output
PINSEL0 &= ~( (1<<14)|(1<<15) );
```

---

---

```
IOODIR |= (1<<7);
```

---

Konfiguration der Pins für SPI0 als SPI [D8, S.76 Tab.60].

---

```
//Pins for SPI0
PINSEL0 |= (1<<8)|(1<<10)|(1<<12);
PINSEL0 &= ~((1<<9)|(1<<11)|(1<<13));
```

---

Konfiguration der SPI0 Taktgeschwindigkeit im *S0SPCR* Register. Begrenzend wirkt sich hierbei die Spezifikation des MAX1168, der eine maximale SPI Frequenz von 4,8MHz erlaubt. Die Berechnung erfolgt durch einen festzulegenden Teiler aus dem Kerntakt des Mikrocontrollers, der in diesem Fall auf 60MHz eingestellt ist. Für die maximal erlaubte SPI Frequenz von 4,8MHz wäre ein Teiler von 12,5 erforderlich. Nach [D8, S.178 Kap.12.4.4] muss der Teiler gleich oder größer 8, sowie eine ganze und gerade Zahl sein. Der nächste mögliche Teiler ist somit 14, was einer SPI Frequenz von 4,28MHz entspricht.

Der LTC2440 könnte mit einer maximalen Frequenz von 20MHz ausgelesen werden, jedoch würde dies keinen Geschwindigkeitsvorteil bringen, das der LTC2440 im Vergleich zu den beiden MAX1168 nur selten gelesen wird und eine individuelle Umkonfiguration des SPI Bus mehr Zeit kosten würde als es bringen würde. Die Einstellung des SPI Takts ist daher für alle drei Bausteine einheitlich eingestellt.

---

```
S0SPCCR = 0x000E; //SPI SCK divider (60MHz/14=4,28MHz)
```

---

Das *S0SPCR* Register enthält die grundlegende Konfiguration der SPI Schnittstelle [D8, S.176f Tab.156]. Bit 0 und 1 sind reserviert und werden nicht behandelt.

Bit 2 legt die Anzahl der pro senden/empfangen übertragenen Bits fest. Eine 0 legt die Anzahl fest auf 8 Bit, eine 1 ermöglicht die Einstellung der Anzahl über die Bits 11:8. Da der MAX1168 seine Antwort in 16Bit schickt und der LTC2440 24 32Bit benötigt soll die Übertragung in 16Bit Blöcken erfolgen um nicht zu viele kleine Datenblöcke übertragen zu müssen. Somit ist Bit 2 auf 1 zu setzen und die Größe von 16Bit über die Bits 11:8 einstellen.

Bit 3 und 4 legen die Phase und die Polarität des SPI Bus fest, wobei die Phase beschreibt, ob die Daten bei der ersten Flanke (fallenden) (CPHA=0) oder bei der zweiten (steigend) (CPHA=1) übernommen werden sollen. Die Polarität legt fest, ob der SPI Takt im Ruhezustand low (CPOL=0) oder high (CPOL=1) sein soll. Die Einstellung von CPHA und CPOL richtet sich nach dem angesprochenen Baustein und kann durchaus von Baustein zu Baustein variieren. Der MAX1168 erfordert nach P26 CH „Serial Interfaces“ die Einstellung für CPHA=CPOL=0. Die nötige Konfiguration

für den LTC 2440 ist aus seinem Datenblatt [D2, S.11 Fig.3] zu entnehmen. Aus dem Timingdiagramm ist zu erkennen, dass der LTC2440 ebenso die Einstellung CPHA=CPOL=0 benötigt. Somit kann die Einstellung von CPHA und CPOL in der init Routine erfolgen und muss nicht individuell für jeden Baustein eingestellt werden.

Bit 5 legt fest ob der SPI0 Port im Slave- (=0) oder im Mastermode (=1) betrieben werden soll. Für den benötigten Mastermode ist somit Bit 5 auf 1 zu setzen.

Bit 6 legt fest ob zuerst das MSB (=0) oder das LSB (=1) übertragen werden soll. Bei dem MAX1168 ist in [D14, S.18 Abb.11] und bei dem LTC2440 [D2, S.11 Fig.3] zu erkennen, dass das MSB zuerst übertragen werden muss. Bit 5 ist somit 0 zu setzen.

Bit 7 aktiviert (=1) oder inaktiviert (=0) einen Interrupt bei Ende der Datenübertragung. Da ein Interrupt gewünscht ist, ist Bit 7 auf 1 zu setzen.

Bit 8-11 legen die Anzahl der zu übertragenden Bits fest. Aktivierung erfolgte über Bit 2. Für den gewünschten 16Bit Transfer sind Bit 8-11 auf 0 zu setzen.

Zur Erhöhung der Geschwindigkeit werden die Bits zunächst in eine Variable geschrieben und am Ende in das Register übertragen.

---

```

unsigned short int s0spcr_var = S0SPCR;

//          DataSizeSelBit, MasterMode. IRQ On
s0spcr_var |= (1<<2)|(1<<5)|(1<<7);

//          CPHA=0, CPOL=0, MSB
s0spcr_var &= ~(1<<3)|(1<<4)|(1<<6)|
              // 16BitDataSize
              ((1<<8)|(1<<9)|(1<<10)|(1<<11)) );

S0SPCR = s0spcr_var;

```

---

Für die Konfiguration des Interupts ist es nötig die Funktion anzugeben, welche im Falle des Interrupts aufgerufen werden soll (*VIC\_ADDR\_SPI0*).

Des weiteren ist es nötig in *VIC\_CTRL\_SPI0* den Interruptvektor zu aktivieren und den gewünschten Slot zuzuweisen. Der Slot ist Hardwaretechnisch festgelegt und ist in diesem Fall 10. Über ein Define ist ihm die Bezeichnung *VIC\_Slot\_SPI0* zugewiesen.

Über *VICIntEnable* wird der Interrupt selbst aktiviert.

---

```

//set interrupt vector
VIC_ADDR_SPI0 = (unsigned long) spi_int;

// use it for SPI0 interrupt
VIC_CTRL_SPI0 = VICVectCntl_ENABLE | VIC_Slot_SPI0;

```

---

---

```
//enable SPI0 interrupt
VICIntEnable = (1<<VIC_Slot_SPI0);
```

---

Beim ersten Start ist es nötig den SPI anzustoßen, also einen ersten Interrupt auszulösen um so die ISR aufzurufen. Dies erfolgt durch den Aufruf der Funktion `spi_start`.

---

```
// start spi interrupt
spi_start();
```

---

### 15.3.2 Die Funktion „spi\_start“

Die Funktion `spi_start` wird innerhalb von `spi_init` einmal aufgerufen um die SPI Routinen anzustoßen. Hierzu werden zunächst alle Bausteine deselektiert und dann ein Dummybyte gesendet. Da kein Baustein selektiert wird, hat dieses keinen Einfluss. Es löst jedoch einen Interrupt aus, der die ISR startet und so die SPI Kommunikation startet.

---

```
// unselect all spi devices
CSSEL(UNSEL);

// send dummy bytes to start spi interrupt
SpiSend(0x0000);
```

---

### 15.3.3 Die Funktion „spi\_int“

`spi_int` ist die Interrupt-Service-Routine, die bei jedem Interrupt aufgerufen wird und die die eigentlichen Schreib- und Lesevorgänge auf dem SPI Bus koordiniert. Die Steuerung der Schreib- und Lesevorgänge erfolgt über eine 3-stufige Zustandsmaschine.

Im Zustand 0 wird zunächst die zweite, steuernde Zustandsmaschine aufgerufen. Diese liefert über die Variable `chip_select` die Information, welcher der drei Bausteine ausgewählt werden muss. Über `cmd_word` wird mitgeteilt, welches Kommandowort an den Baustein gesendet werden muss. Die Variable `stor_lo` beinhaltet die Position an welcher das gelesene Ergebnis im Array, welches die Werte abspeichert, gespeichert werden muss. Über die Variable `mode` wird mitgeteilt, ob im 16Bit (MAX1168) oder im 32 Bit (LTC2440) Mode gelesen werden muss. Hierüber erfolgt somit die Unterscheidung zwischen MAX1168 und LTC2440.

Der entscheidende Unterschied zwischen der Ansteuerung des LTC2440 und den beiden MAX1168 liegt darin, dass der LTC2440 zuerst zweimal mit je 16Bit gelesen werden muss. Hiernach wird das neue Kommandowort geschickt. Es wird somit immer das alte Ergebnis gelesen und danach das neue Kommandowort gesendet. Die

MAX1168 werden dagegen erst mit dem Kommandowort versorgt und danach kann sofort das Ergebnis eingelesen werden.

Im Zustand 0 erfolgt eine Unterscheidung, ob im Mode 16 oder 32 zu arbeiten ist. Für den Mode 32 wird ein Dummybyte übertragen, um die erste Hälfte des Ergebnisses des LTC2440 übertragen zu können. Im 16 Bit Mode wird das Kommandowort an den MAX1168 gesendet.

Anschließend wird *state* auf 1 gesetzt und durch das *break* die Zustandsmaschine verlassen. Nach Rücksetzen das SPI0 InterruptBit und der Vector Interruptadresse wird die InterruptServiceRoutine verlassen.

---

```

case 0:

    sm_ctrl (&chip_select, &cmd_word, &stor_lo, &mode);

    CSSEL(chip_select);

    if (mode == 32) {

        //Send Dummy Word

        SpiSend (0x0000)

    }

    else {

        //Send CMD Word

        SpiSend (cmd_word);

    }

    h_state = 1;

    break;

```

---

Sobald die Datenübertragung, die in Zustand 0 ausgelöst wurde, beendet ist, wird ein Interrupt ausgelöst und die ISR aufgerufen und in der Zustandsmaschine der Zustand 1 ausgeführt.

In diesem wird im 32 Bit Mode der im SPI Datenregister vorhandene Wert, welcher die obere Hälfte des Ergebnis des LTC2440 darstellt, um 16 Stellen nach links verschoben in eine Variable geschrieben. Anschließend wird erneut ein Dummyword gesendet um den zweiten Teils des Ergebnis zu übertragen.

Im 16 Bit Modus wird nur ein Dummyword gesendet um das Ergebnis zu übertragen.

Nach Setzen von *state* auf 2 wird die ISR wie unter Zustand 1 verlassen.

---

```

case 1:

    if (mode == 32) {

```

---

---

```
        // mode 32 read first data
        ltc_tmp = S0SPDR<<16;
        //send 2nd dummy word
        SpiSend (0x0000);
    }
    else {
        //mode 16 send dummy word
        SpiSend (0x0000);
    }
    h_state = 2;
    break;
```

---

In Zustand 2 wird im 32 Bit Mode der zweite Teil des Konvertierungsergebnis gelesen und in die unteren 16Bit der temporären Variable geschrieben.

Im folgenden wird überprüft ob das Ergebnis gültig ist. Hierzu wird getestet, ob Bit 31 0 oder 1 ist. Bei dem Bit 31 handelt es sich um das EOC Bit des ADC, dieses ist 1, wenn die Konvertierung noch nicht abgeschlossen und das Ergebnis somit ungültig ist. Nur wenn Bit 31 0 ist wurde zu einem Zeitpunkt gelesen, wo die Konvertierung abgeschlossen war und das Ergebnis somit gültig ist. Ist das Ergebnis ungültig, werden die gelesenen Werte verworfen und auch 0 gesetzt. Sind die eingelesenen Werte gültig, so erfolgt eine Aufbereitung des Ergebnisses. Dieses ist nötig, da sich das eigentliche gelesene Ergebnis als 24 Bit Wert in der Mitte der gelesenen 32 Bit befindet. Bit 28 stellt das MSB und Bit 5 das LSB dar. Durch eine Maskierung werden Bit 29-31 entfernt. Durch ein Schieben um 5 Stellen nach rechts werden die 5 unteren Bits verworfen, so dass das Ergebnis nun rechtsbündig in der Variablen steht. Das Bit 29 ist als SignBit beschrieben. Dies zeigt jedoch nur an, ob die Eingangsspannung am Differenzeingang des ADC positiv oder negativ ist. Es ist kein echtes SignBit für das Ergebnis. Da durch das Schaltungsdesign nur positive Spannungen am ADC Eingang möglich sind, kann auf dieses Bit verzichtet werden.

Im 16 Bit Mode wird lediglich das eingelesene Ergebnis aus dem SPI Datenregister in eine Variable geschrieben.

---

```
case 2:
    if (mode == 32) {
        // mode 32 read second data
        ltc_tmp |= S0SPDR;
```

---

---

```

        if ( (ltc_tmp & (1<<31)) == 1 ) {
            // set error
            imu_value = 0;
        } else {
            //ignore first 4 Bits, !!! sign Bit canceled !!!
            // shift 5 pos. right make Bit 5 to LSB
            imu_value = ((ltc_tmp & 0xFFFFFFFF) >> 5);
        }
    }
else {
    //mode 16 read data
    imu_value = S0SPDR;
}

```

---

Ebenfalls in Zustand 2 erfolgt die Berechnung von Mittelwerten. Hierzu werden eine bestimmte Zahl von Messwerten addiert und später durch die Anzahl geteilt. Es ergeben sich stabilere Messwerte im Vergleich zu einzelnen Messwerten. Die Anzahl ist in der Variabelendeklaration von *imu\_average\_nb[]* für jeden Sensor individuell einstellbar.

Nachdem nach der Mittelwertbildung der Zustand wieder auf 0 gesetzt wird, erfolgt noch die Besonderheit, dass im 32 Bit Mode das neue Kommandowort an den LTC2440 gesendet wird. Dies kann erst nach der Mittelwertbildung geschehen, da die Mittelwertbildung zeitlich länger dauert als eine Datenübertragung auf dem SPI Bus, es würde somit einen Konflikt geben, welcher zur Blockade der SPI Kommunikation führen würde. Nach Übertragung des Dummywords wird die Zustandmaschine und ISR regulär verlassen.

Im Falle des 16Bit Mode erfolgt jedoch keine neue SPI Kommunikation die einen Interrupt auslösen würde. Aus diesem Grund muss ohne ein *break* direkt bei dem Zustand 0 fortgesetzt werden. Dies ist auch der Grund, das der Zustand 0 unter Zustand 2 angeordnet ist und nicht an oberster Stelle.

---

```

// calculate average values
imu_value_sum[stor_lo] += imu_value - imu_offset[stor_lo];
if((imu_count[stor_lo]++) > imu_average_nb[stor_lo]) {
    // export value

```

---

---

```
imu_average[stor_lo]=imu_value_sum[stor_lo]/imu_count[stor_lo];

// cleanup

imu_count[stor_lo] = 0;

imu_value_sum[stor_lo] = 0;

analogconversion();

}

h_state = 0;

if (mode == 32) {

//mode 32 send cmd word

SpiSend (cmd_word);

break;

}

/* if mode = 16 no break; */
```

---

#### 15.3.4 Die Funktion „sm\_ctrl“

Die Funktion *sm\_ctrl* stellt die Informationen zur Verfügung, die die Zustandmaschine, welche die SPI Schreib- und Lesevorgänge kontrolliert (*spi\_int*), benötigt. Dies beinhaltet über die Variable *cs* welcher Baustein über das Chip Select Signal aktiviert werden soll, über *cmd\_imu* welches Kommandowort gesendet werden muss, über *stor\_loc* die Position an welcher Position des speichernden Arrays gespeichert werden soll, sowie über *mode\_p* in welchem Modus (16 oder 32 Bit ) gearbeitet werden muss.

Herzstück der Funktion ist eine Zustandsmaschine mit 20 Zuständen. Jeweils ein Zustand entspricht einem Sensorwert. Besonderheiten sind jedoch bei den drei Magnetsensoren und dem Absolutdrucksensor gegeben.

Die Magnetsensoren haben die Eigenschaft sich zu magnetisieren und müssen daher durch eine integrierte Entmagnetisierspule entmagnetisiert werden. Die Spulen werden abwechselnd mit einem positiven und negativen Impuls belegt und danach jeweils der Sensorwert gelesen. Es ergeben sich somit pro Sensor zwei Werte, der erste nach einem positiven Impulse, ein weiterer nach einem negativen Impuls. Durch eine spätere Verrechnung dieser beiden Werte lassen sich Offsetfehler beheben. Für alle drei Achsen ergeben sich somit nicht nur drei Werte, sondern sechs Werte und somit auch Zustände. Zusätzlich sind Wartezeiten zwischen dem Auslesen einzuhalten, so darf nur alle 2ms gelesen werden. Das Gleiche trifft auch auf den Absolutdrucksensor zu, der je nach Einstellung eine bestimmte Zeit zur Konvertierung benötigt und nicht vor Beendigung der Konvertierung gelesen werden darf.

Zu Beginn erfolgt zunächst die Erstellung der Kommandowörter für die drei Bausteine

---

```

        // ext clk | int. Ref. allways on | no scan
unsigned short cmd_MAXA = (0<<8)|(0<<9)|(0<<10)|(0<<11)|(0<<12);

// ext clk | int. Ref. allways on | no scan
unsigned short cmd_MAXB = (0<<8)|(0<<9)|(0<<10)|(0<<11)|(0<<12);

unsigned short cmd_LTC= 0x4000; //27,5Hz = 36,36ms conversion
time, 375nV noise, 24Bit (DS p14)

```

---

Die ersten 5 Zustände sind identisch und daher zusammengefasst. Sie lesen die ersten 5 Kanäle des ersten MAX1168 ein, entsprechend den drei Gyro Achsen, sowie der x und y Achse des 3g Beschleunigungssensor ADXL330.

Das Kommandowort wird aus dem am Anfang der Funktion gebildeten Rohform sowie der Variablen *channel* zusammengesetzt. *Channel* gibt an welcher Kanal des MAX1168 gelesen werden soll. Diese Variable wird nach jedem Durchlauf um 1 erhöht, ihr Wertebereich erstreckt sich von 0-7. Für die Bildung des Kommandoworts wird *channel* um 13 Stellen nach links verschoben, um bei dem 16Bit Kommandowort die 3 höchstwertigen Bits zu belegen.

Die Variable *sl* bestimmt die Speicherposition und wird in jedem Zustand auf *stor\_loc* geschrieben und anschließend inkrementiert.

Die Funktion der *do-while* Schleife sowie der Variablen *redo* wird später erläutert.

---

```

do {
    switch (state) {
    case 0:    //Gyro X,Y,Z, ACC X,Y
                redo = 0;
                /* kein break; */
    case 1:
    case 2:
    case 3:
    case 4:

```

---

---

```
*cmd_imu = cmd_MAXA | (channel<<13);  
*cs = CS_MAX1168A;  
*mode_p=16;  
*stor_loc=sl;  
channel++;  
sl++;  
state++;  
break;
```

---

Die Zustände 5-7 sind identisch zu den Zuständen 0-4 aufgebaut, jedoch als Einzelblöcke ausgeführt. Dies liegt darin begründet, dass je nach den verschiedenen Möglichkeiten zur Bestückung verschiedener Beschleunigungssensoren diese Zustände teilweise nicht benötigt werden und der Quellcode so leichter angepasst werden kann.

Im Zustand 7 ist aber noch zu beachten, dass die Variable *channel* nicht weiter erhöht wird, sondern auf 0 zurückgesetzt wird, da alle Kanäle des ersten MAX1168 gelesen wurden und nun folgend bei Kanal 0 des zweiten MAX1168 fortgesetzt wird.

---

```
case 5: //ACC Z  
*cmd_imu = cmd_MAXA | (channel<<13);  
*cs = CS_MAX1168A;  
*mode_p=16;  
*stor_loc=sl;  
channel++;  
sl++;  
state++;  
break;
```

---

---

```
case 7: //ACCZ2  
*cmd_imu = cmd_MAXA | (channel<<13);  
*cs = CS_MAX1168A;  
*mode_p=16;  
*stor_loc=sl;  
channel=0;
```

---

---

```

sl++;

state++;

break;

```

---

Die Zustände 8,9 und 10 lesen die Daten der drei Magnetsensoren nach einem positiven Entmagnetisierimpuls (Flip) ein. Das Lesen der Magnetsensoren darf erst 2ms nach dem Flipimpuls geschehen, somit erfolgt der Aufruf zeitgesteuert über einen 1ms timer, sowie über die Variable *flip\_h\_l*. *flip\_h\_l* dient zur Unterscheidung ob gerade positiv oder negativ entmagnetisiert wurde.

Für den Fall, dass eine Bedingung nicht zutrifft wird *channel* sicherheitshalber wieder auf 0 gesetzt, sowie *state* und *sl* auf 11 um zum Zustand 11 zu springen. Da Zustand 9 und 10 nur mit Zustand 8 zusammen gelesen werden, werden sie somit mit übersprungen. Die Variable *redo*, welche ebenso 1 gesetzt wird, bewirkt, dass die *do-while* Schleife und somit die Zustandsmaschine erneut durchlaufen wird und so sofort zu Zustand 11 übergegangen wird ohne den Funktionsaufruf von *sm\_ctrl* zu beenden. Würde der Funktionsaufruf direkt beendet werden, würde die *spi\_int* Funktion falsche Werte geliefert bekommen.

Die *do-while* Schleife führt somit die Zustandsmaschine solange immer wieder aus bis *redo* wieder 0 gesetzt wurde. Das Rücksetzen von *redo* erfolgt in Zustand 11, 14 und 0.

Die Zustände 9 und 10 sind wieder nach bekanntem Prinzip aufgebaut, jedoch befindet sich in Zustand 10 noch das Auslösen des negativen Entmagnetierimpulses mit *FLIP\_L* und das Setzen von *flip\_h\_l* auf 0 zur Kenntlichmachung, dass als nächstes die negativen Werte eingelesen werden müssen.

Die Zustände 11-13 sind identisch zu den Zuständen 8-10 aufgebaut, lesen jedoch die Sensoren nach einen negativen Entmagnetisierimpuls ein. Entsprechend wird in Zustand 13 nicht ein negativer sondern ein positiver Entmagnetisierimpuls ausgelöst und *flip\_h\_l* 1 gesetzt.

---

```

case 8: //MAG X,Y,Z

    if ( (timer1_cnt > magnet_cnt) && (flip_h_l == 1) ) {

        magnet_cnt = timer1_cnt+1;

        *cmd_imu = cmd_MAXB | (channel<<13);

        *cs = CS_MAX1168B;

        *mode_p=16;

        *stor_loc=sl;

        channel++;

        sl++;
    }

```

---

---

```
        state++;
    } else {
        channel=0;
        state = 11;
        sl=11;
        redo = 1;
    }
    break;

case 9:
    *cmd_imu = cmd_MAXB | (channel<<13);
    *cs = CS_MAX1168B;
    *mode_p=16;
    *stor_loc=sl;
    channel++;
    sl++;
    state++;
    break;

case 10:
    *cmd_imu = cmd_MAXB | (channel<<13);
    *cs = CS_MAX1168B;
    *mode_p=16;
    *stor_loc=sl;
    channel=0;
    sl++;
    state++;
    flip_h_l=0;
    FLIP_L;
    break;
```

---

Die Zustände 14-18 sind wieder nach bekannter Weise aufgebaut und beinhalten keine Besonderheiten. Lediglich in Zustand 14 befindet sich ein Setzen von *redo* auf 0 um nach eventuell versuchtem Ausführen von Zustand 8 oder 11 keinen weiteren Durchlauf der *do-while* Schleife zu erhalten.

In Zustand 19 erfolgt die Werteübergabe für das Einlesen des Luftdrucksensors. Die Werteübergabe erfolgt nur alle 50ms, da eine gewisse Konvertierungszeit eingehalten werden muss. Die derzeitige Einstellung erfordert eine Konvertierungszeit von ca. 37ms, es ist als noch eine Sicherheitsreserve von 13ms vorhanden, die bei Bedarf noch verkürzt werden kann. Im Gegensatz zu allen vorherigen Zuständen wird nun *mode\_p* auf 32 gesetzt, und da es sich um den letzten Zustand handelt, werden *state* und *sl* wieder auf ihren Startwert von 0 gesetzt.

Im Falle, dass die 50ms noch nicht vorüber sind, tritt der *else*-Fall in Kraft, wo wiederum *state* und *sl* auf 0 gesetzt werden, *redo* jedoch 1. Durch die *do-while* Schleife wird somit ein erneuter Durchlauf der Zustandsmaschine erreicht. Das Rücksetzen von *redo* erfolgt unmittelbar in Zustand 0.

---

```
case 19: //AP
    if (timer1_cnt > ltc_cnt) {
        ltc_cnt = timer1_cnt+49; //only every 50ms
        *cmd_imu = cmd_LTC;
        *cs = CS_LTC2440;
        *mode_p=32;
        *stor_loc=sl;
        sl=0;
        state=0;
        break;
    } else {
        state = 0; //wieder besser gleich bei 0 weitermachen
        redo = 1;
        sl=0;
    }
    break;
```

---

## 15.4 Magnetometer

Die Einbindung des Magnetometers erfolgte in die Datei *mm3.c*. Hierbei handelt es sich um eine vorhandene Routine, welche speziell für den MicroMag3 geschrieben wurde. Diese Routine musste nun soweit angepasst werden, dass sie mit dem auf den KMZ51 basierenden Magnetometer ebenso ein die Nordposition berechnet (Heading). Als Ergebnis dieser Berechnung liegt die Ausrichtung als Winkel von 0-359° vor, wobei sich bei 0° die Nordposition befindet

Die Routine arbeitet ursprünglich so, dass sie den MicoMag3 ausliest und dessen Rohdaten in ein Struct schreibt. Die Änderung besteht nun darin, nicht die Daten des MicroMag3 zu verarbeiten, sondern die der KMZ51 Sensoren. Die Daten der KMZ51 liegen im *imu\_average* Array bereit, jedoch existieren pro Sensor zwei Werte. Ein Wert nach einem positiven Flipimpuls, sowie ein Wert nach einem negativen Flip Impuls. Durch Subtraktion dieser beiden Werte lassen sich Offsetfehler kompensieren.

Erste Versuche haben jedoch ergeben, dass mit den direkten Werten der KMZ51 kein korrektes Heading errechnet werden konnte. Eine Analyse der Verarbeitungsroutine ergab, dass diese nur in bestimmten Wertebereichen arbeitet und daher auf Rohdaten in einem bestimmten Bereich angewiesen ist. Dieser Bereich ist auf die Rohdaten des MicroMag3 abgestimmt. Zum Anpassen der Werte der KMZ51 an die des MicroMag3 wurde ein Faktor berechnet mit dem die Werte der KMZ51 geteilt werden. Der Wertebereich des MicroMag3 liegt etwa bei 40-80, der der KMZ51 bei etwa 14000-30000. Es hat sich gezeigt, dass der Algorithmus zur Berechnung des Headings mit einem Faktor von 350 gute Ergebnisse liefert.

Bild 15.1 zeigt das berechnete Heading bei einer hin und zurück geführten 360° Rotation. Bild 15.2 zeigt die Rohdaten der drei Magnetsensoren aus welchen das Heading berechnet wird.

---

```
#ifndef HBC

    // copy values to struct

    MM3.x_axis=(imu_average[IMU_MAGXL]-imu_average[IMU_MAGXH])/350;

    MM3.y_axis=(-imu_average[IMU_MAGYL])-(-imu_average[IMU_MAGYH])/350;

    MM3.z_axis=(imu_average[IMU_MAGZL]-imu_average[IMU_MAGZH])/350;

#endif //HBC
```

---

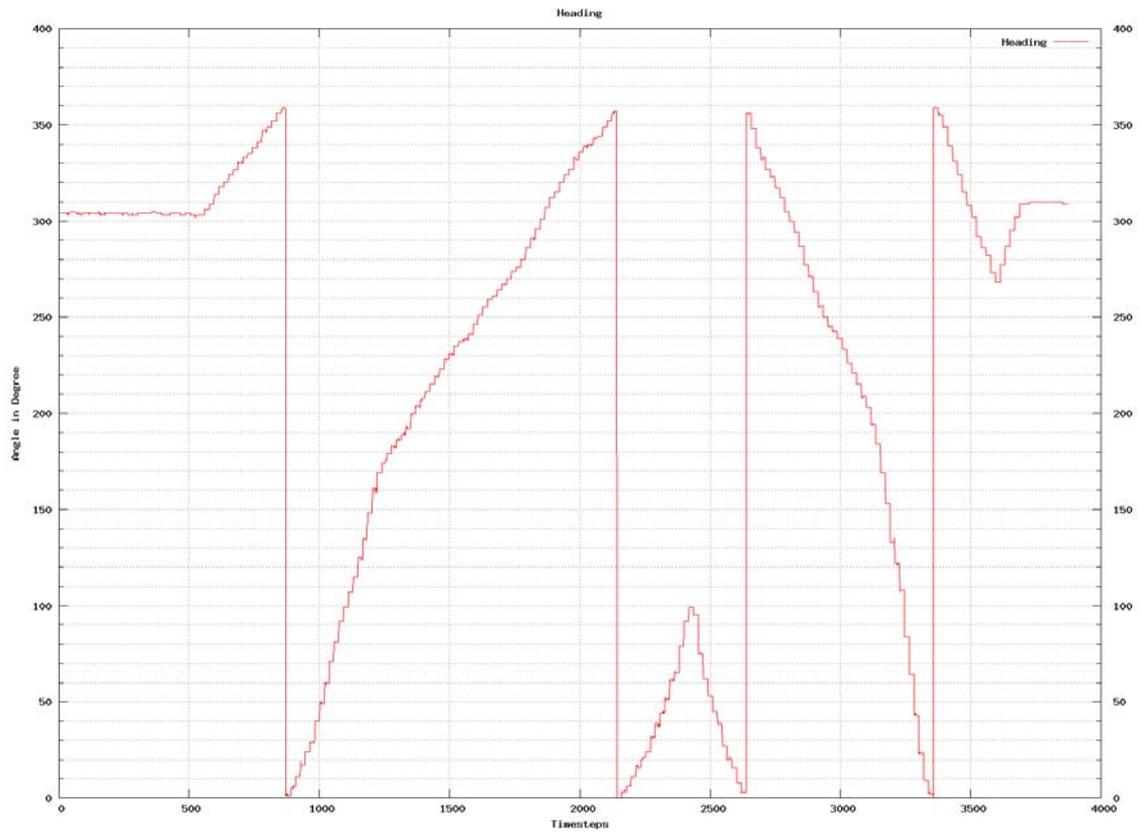


Bild 15.1: Berechnetes Heading

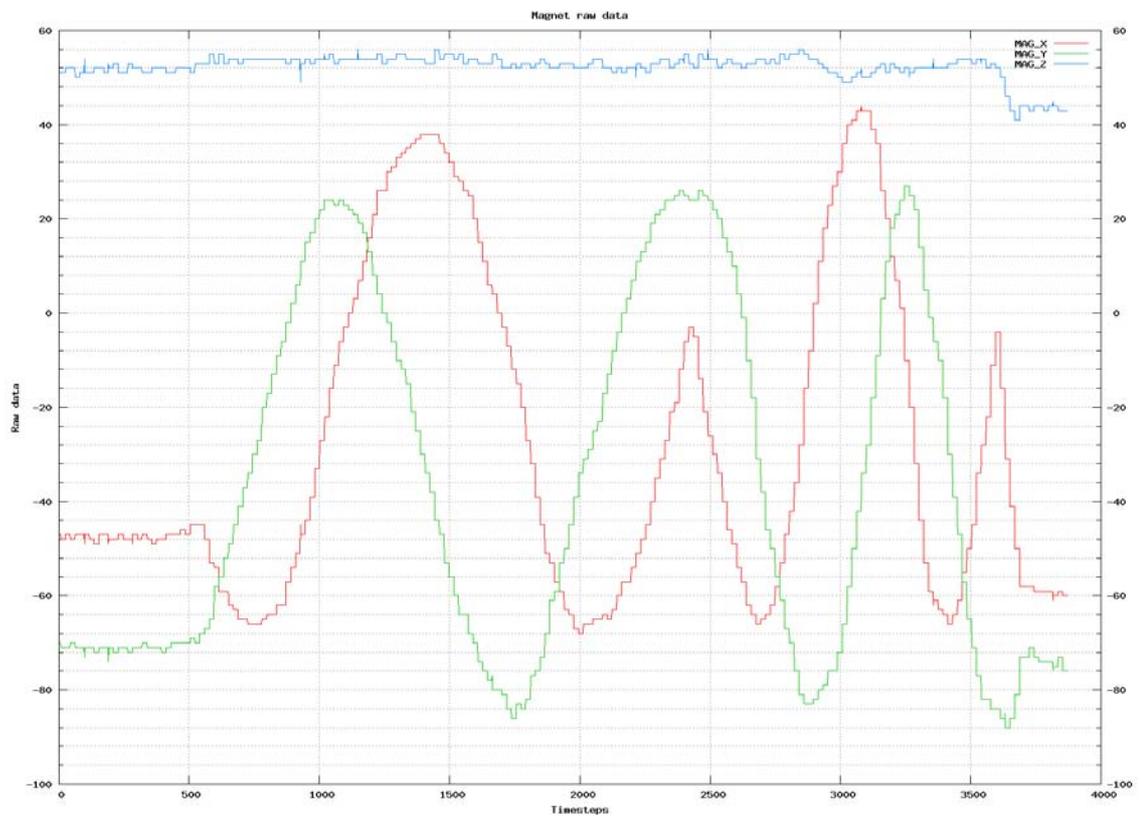


Bild 15.2: Rohdaten der Sensoren

## 15.5 Umrechnungsfunktionen

In der Datei *convert.c* werden die Umrechnungen zwischen den Bitwerten der Analog-Digital-Umsetzer und den physikalischen Einheiten vorgenommen. Hierzu wurden in der gegebenen Datei die benötigten Codeblöcke dupliziert, in *defines* eingeschlossen und die aus der Kalibrierung erhaltenen Werte eingetragen.

Ebenso erfolgt hier die Anpassung der Wirkrichtung.

Für zukünftige weitere Autopiloten wird es aufgrund der jeweils unterschiedlichen Kalibrierwerte erforderlich sein weitere Unterscheidungen der Baugruppen zu tätigen.

---

```
#ifndef HBC

void accel2ms2( void ) {

    accel[ACC_X] = (float)(imu_average[IMU_ACCX])/5235*9.81;
    accel[ACC_Y] = (float)-(imu_average[IMU_ACCY])/5228*9.81;
    accel[ACC_Z] = (float)(imu_average[IMU_ACCZ])/5441*9.81;
}

#endif //HBC
```

---

Auf gleiche Weise erfolgt die Anpassung bei den Gyroskopdaten. Hier fiel jedoch auf, dass in der Berechnungsfunktion ein Fehler vorliegt.

Korrekt müsste die Berechnung

$$Konwert = \frac{Bitwert}{\pi * Faktor} \quad (15.1)$$

Lauten und nicht wie in der Funktion angegeben.

$$Konwert = Bitwert * \frac{\pi}{Faktor} \quad (15.2)$$

Da bei einer Korrektur des Fehlers in der *convert.c* alle nachfolgenden Regelalgorithmen ebenso angepasst werden müssten, wurde der Fehler übernommen. Bei Fortsetzung des Projekts sollte dieser Fehler umgehend behoben werden. Die Ausgegebenen Werte befinden sich somit nicht in der gewollten Einheit rad/s.

---

```
#ifndef HBC
void gyro2rads( void ) {
    gyro[G_ROLL] = (float)-(imu_average[IMU_ROLL]);
    gyro[G_ROLL] *= PI / 1381 ;
    gyro[G_PITCH] = (float)-(imu_average[IMU_PITCH]);
    gyro[G_PITCH] *= PI / 1388 ;
    gyro[G_YAW] = (float)(imu_average[IMU_YAW]);
    gyro[G_YAW] *= PI / 1388 ;
}
#endif //HBC
```

---

## 16 Kalibrierung

Für einen späteren völlig autonomen Flugbetrieb eines Quadropters ist eine Kalibrierung der Sensoren erforderlich.

Es werden hierzu den Sensoren definierte Bedingungen auferlegt und die Antworten der Sensoren aufgezeichnet. Als Antworten werden die von den ADC's gelieferten Rohdaten ohne einen Offsetausgleich verwendet. Diese werden über die Debug-Routine *dbg\_imu\_full* über die serielle Schnittstelle ausgegeben und per PC gelesen und in Dateien geschrieben. Hierbei hat es sich bewährt für jede neue Positionierung der IMU eine eigene Datei anzulegen. Wichtig ist auch noch einer Umpositionierung die Daten erst mitzuschreiben, wenn sich die Lage der IMU stabilisiert hat. Die Daten werden etwa 1 Minute mitgeschrieben, was ca. 1000 Messwerten entspricht. Aus diesen 1000 Messwerten wird der arithmetische Mittelwert gebildet, der wiederum zur Berechnung der Faktoren verwendet wird, welche in die Konvertierungsroutine eingetragen werden.

### 16.1 Drehratensensoren

Für die Kalibrierung der Drehratensensoren muss eine konstante Drehbewegung mit bekannter Drehgeschwindigkeit zur Verfügung gestellt werden.

Als einfache Methode bot sich hier ein Plattenspieler an. Das bereitgestellte Gerät verfügt über 2 Geschwindigkeitsstufen von  $33 \frac{1}{3}$  U/min und 45 U/min, angetrieben durch einen Gleichstrommotor. Als Problem stellte sich die Genauigkeit der Drehzahl

heraus. Zwar konnte über Feinregler die Drehzahl korrigiert, jedoch nicht auf einen definierten Wert eingestellt werden.

Es war somit von Nöten eine Drehzahlmessung zu konstruieren. Die einfachste Möglichkeit bestand hierbei in der Verwendung einer einfachen Lichtschranke. Die Lichtschranke sollte bei jeder Umdrehung des Plattentellers einmal unterbrochen werden und so zählbare Impulse erzeugen. Zur Auswertung der Impulse diente der im Labor vorhandene Frequenzzähler, der eine hohe Genauigkeit und Auflösung bei geringem Aufwand bot.

Die Lichtschranke wurde aus der vorhandenen Sendediode TSSS2600 und dem zugehörigen Fototransistor TEST2600 aufgebaut. Die Sendediode wird im Gegensatz zur üblichen Beschaltung nicht mit einem Rechteck impulsförmig gespeist, sondern mit einem konstanten Gleichstrom von ca. 50mA, um so Beeinflussungen durch die Impulse bei der späteren Signalauswertung zu verhindern. Aufgrund der geringen Entfernung und der somit geringen erforderlichen Sendeleistung ist dies problemlos möglich.

Der maximal bei Beleuchtung durch den Fototransistor fließende Strom wird über einen Widerstand von  $4,7k\Omega$  auf ca. 1mA begrenzt. Das Signal zwischen Fototransistor und Widerstand ist somit bei unbestrahltem Transistor bei etwa 0V, bei Bestrahlung ergaben Messungen etwa 1V. Das Signal wird von einem Operationsverstärker  $\mu A741$  18-Fach verstärkt. Der eigentlich unnötig hohe Verstärkungsfaktor bewirkt hier eine schnelle und sichere Vollaussteuerung des OP um ein möglichst sauberes Signal zu erhalten. Anschließend wurde das Signal einem Schmitt-Triger 74HC14 zugeführt um einen Logikpegel zu erhalten. Dies war nötig, da der  $\mu A741$  bei gewollten 0V etwa 1,2V und bei gewollten 5V nur etwa 4V lieferte. Abhilfe hätte hier ein Rail-to-Rail OP oder eine symmetrische Spannungsversorgung geschaffen. Ein Rail-to-Rail OP stand jedoch nicht zur Verfügung und eine symmetrische Spannungsversorgung wäre zu aufwändig gewesen. Somit erfolgte der Umweg über den 74HC14 an dessen Ausgang nun ein sauberes Logiksignal zur Verfügung stand, welches problemlos vom Frequenzzähler erkannt wurde.

Über die Feinregler des Plattenspielers kann nun die Drehzahl abgeglichen werden. Eine Drehzahl von  $33 \frac{1}{3}$  U/min entspricht einer Frequenz von 555,556 mHz und bei 45 U/min einer Frequenz von 750 mHz. Bei den Versuchen hat sich ergeben, dass sich ab der vierten Nachkommastelle Laufunruhen des Plattenspielers auswirken, diese erreichbare Genauigkeit ist jedoch ausreichend. Zu beachten ist noch, dass der Plattenspieler vor dem Abgleich etwa 10-15min zum Warmlaufen in Betrieb zu nehmen ist. Tests haben gezeigt, dass die Drehgeschwindigkeit zu Beginn noch zunimmt, bis sie sich nach genannter Zeit stabilisiert.

Zur Ausrichtung der IMU auf dem Plattenteller wurde eine Halterung konstruiert, mit der es möglich ist die Gyroskope auf einfache Weise rechtwinklig und auf der Rotationsachse auszurichten. Die Halterung besteht aus mehreren Pertinaxplatinen, welche rechtwinklig verlötet sind. Zur einfachen und genauen Positionierung sind in Position der Gyroskopachsen Bohrungen eingebracht, in welche die Plattenspielerachse

gesteckt werden kann. Für jede Achse befindet sich je Drehrichtung eine Bohrung. Es ist somit möglich durch Drehung bzw. umgekehrter Montage der IMU alle drei Achsen sowohl in positiver als auch in negativer Drehrichtung zu kalibrieren.

Der genaue Kalibriervorgang ist in Kapitel 21.6.1 beschrieben.

Für die Umrechnung der Sensorwerte in der Konvertierungsroutine von Bit zu rad/s ist ein Faktor erforderlich, welcher aus den aufgenommenen Messdaten errechnet werden muss. Hierzu wird die Drehzahl des Plattenspielers, gemessen durch die Impulse der Lichtschranke und in mHz dargestellt, in rad/s umgerechnet. Von den aufgezeichneten Bitwerten für die verschiedenen Drehgeschwindigkeiten wird die Differenz zum Neutralwert gebildet. Der Neutralwert wurde während der Kalibrierung der Beschleunigungssensoren für jede Achse mitgemessen, so dass gesamt 6 Werte zur Verfügung stehen, aus denen der arithmetische Mittelwert zur Berechnung verwendet wird (Tabelle 16.4). Der benötigte Faktor ergibt sich aus der Division von der Betragsdifferenz der Bitwerte mit der Drehgeschwindigkeit in rad/s. Die zur Verfügung gestellte Konvertierungsroutine benötigt jedoch den Faktor geteilt durch  $\pi$ . Der letztendlich in der Konvertierungsroutine eingesetzte Wert für den Faktor errechnet sich aus den Mittelwerten aller vier Faktoren welche aus den vier verschiedenen Geschwindigkeiten errechnet wurden. Die Berechnungen für die drei Gyroskope sind in Tabelle 16.1 - Tabelle 16.3 aufgeführt.

Die für die Gyroskopkalibrierung aufgenommenen Werte sind aufgrund Ihrer Vielzahl nicht in dieser Arbeit aufgelistet, sondern nur als Datei im SVN abgelegt. Jedoch erfolgt im Anhang in Kapitel 0 eine grafische Darstellung der Messwerte mit einem zugehörigen Histogramm zur Darstellung der Häufigkeitsverteilung der Messwerte (Bild 21.46 - Bild 21.60). Da es sich bei der Variation der Messwerte um einen Zufälligen Prozess handelt, muss in dem Histogramm eine Glockenkurve erkennbar sein. Ist dies nicht der Fall, ist die Integrität der Messwerte anzuzweifeln und diese zu überprüfen.

Tabelle 16.1: Berechnung Umrechnungsfaktor X-Gyroskop

<b>X-Achse</b>	<b>Frequenz Plattenspieler in mHz</b>	<b>Dreh- geschwin- digkeit in %s</b>	<b>Dreh- geschwin- digkeit in rad/s</b>	<b>Messwert in bit</b>	<b>Betrags- differenz</b>	<b>Faktor * PI</b>	<b>Faktor</b>
33	555,50	199,980	3,4903	47333	15040	4309	1372
45	749,65	269,874	4,7102	52657	20364	4323	1376
-33	555,30	199,908	3,4891	17059	15234	4366	1390
-45	750,12	270,043	4,7131	11745	20548	4360	1388
						Mittelwert:	1381

Tabelle 16.2: Berechnung Umrechnungsfaktor Y-Gyroskop

<b>Y-Achse</b>							
<b>Drehzahl soll in U/min</b>	<b>Frequenz Platten- spieler in mHz</b>	<b>Dreh- geschwin- digkeit in %s</b>	<b>Dreh- geschwin- digkeit in rad/s</b>	<b>Messwert in bit</b>	<b>Betrags- differenz</b>	<b>Faktor * PI</b>	<b>Faktor</b>
33	555,50	199,980	3,4903	47518	15225	4362	1388
45	750,90	270,324	4,7180	52931	20638	4374	1392
-33	555,50	199,980	3,4903	16642	15651	4484	1427
-45	750,20	270,072	4,7136	11211	21082	4473	1424
						Mittelwert:	1408

Tabelle 16.3: Berechnung Umrechnungsfaktor Z-Gyroskop

<b>X-Achse</b>							
<b>Drehzahl soll in U/min</b>	<b>Frequenz Platten- spieler in mHz</b>	<b>Dreh- geschwin- digkeit in %s</b>	<b>Dreh- geschwin- digkeit in rad/s</b>	<b>Messwert in bit</b>	<b>Betrags- differenz</b>	<b>Faktor * PI</b>	<b>Faktor</b>
33	555,00	199,800	3,4872	47390	15097	4329	1378
45	750,20	270,072	4,7136	52675	20382	4324	1376
-33	555,20	199,872	3,4884	17192	15101	4329	1378
-45	750,30	270,108	4,7143	11891	20402	4328	1378
						Mittelwert:	1377

Tabelle 16.4: Berechnung der Gyroskopneutralmittelwerte

<b>Achse</b>	<b>X- Gyroskop Ruhe in bit</b>	<b>Y- Gyroskop Ruhe in bit</b>	<b>Z- Gyroskop Ruhe in bit</b>
X1	32293	32209	32092
X2	32295	32221	32093
Y1	32292	32206	32087
Y2	32294	32216	32090
Z1	32298	32210	32088
Z2	32287	32208	32088
	32293	32212	32090

## 16.2 Beschleunigungssensoren

Die Kalibrierung der Beschleunigungssensoren gestaltet sich im Vergleich mit den Gyroskopen als einfach, da als Referenz die Erdanziehungskraft genutzt werden kann. Zur Kalibrierung wird die IMU wieder auf die Halterung montiert und für jede Achse die positiven und negativen Messwerte sowie die Neutralposition aufgezeichnet.

Die genaue Durchführung der Kalibrierung ist dem Kapitel 21.6.1.2 zu entnehmen.

Die aus den bei der Kalibrierung aufgenommenen Daten errechneten Mittelwerte sind im Anhang in den Tabellen Tabelle 21.7 - Tabelle 21.12 aufgeführt. Die eigentlichen Messwerte sind aufgrund ihrer hohen Anzahl nicht einzeln aufgeführt, diese sind im SVN [14] abrufbar. Wie auch bei den Gyroskopen befindet sich im Anhang auch für die Beschleunigungssensoren für jeden Sensor jeder Achse eine grafische Darstellung der Messwerte sowie ein Histogramm zur Darstellung der Häufigkeitsverteilung der Messwerte (Bild 21.65 - Bild 21.73) um die Integrität der Messwerte zu überprüfen.

In diesen Diagrammen sind bei einigen Sensoren Ausreißer zu erkennen, welche bereits deutlich negative Auswirkungen auf die Form der Glockenkurve haben (Bild 21.63). Einzelne geringe Ausreißer sind jedoch möglich und können toleriert werden.

Aus diesen Tabellen wurden für jede Achse von jedem Sensor bei +1g, -1g sowie 0g die Daten in einer je Tabelle zusammengefasst (Tabelle 16.5 - Tabelle 16.9). Der Wert für 0g ist wiederum ein Mittelwert aus den vier Werten, welche bei der Messung der anderen Achsen für  $X=0g$  gemessen wurden. Für die Berechnung des Korrekturfaktors, welcher in der Konvertierungsroutine einzutragen ist, wurde jeweils zwischen +1g und 0g sowie zwischen -1g und 0g die Betragsdifferenz gebildet. Beide Werte sollten nahe beieinander liegen und sich nur innerhalb des üblichen Sensorrauschens ( $\pm 150\text{Bit}$ ) bewegen. Zur Bildung des Korrekturfaktors wird noch einmal der Mittelwert von beiden Betragsdifferenzen gebildet.

Betrachtet man die aus den in der Kalibrierung gemessenen Werten errechneten Differenzen so zeigen sich wirklich nur geringere Abweichungen. Dies bedeutet, dass die Sensoren sowohl im positiven als auch in negativen Messbereich nahezu gleiche Beträge liefern.

Der in der Planung erhoffte Vorteil gegenüber den LIS Sensoren von ST hat sich somit bestätigt. Im Gegensatz zu den LIS Sensoren, welche in Versuchen unterschiedliche Ergebnisse im positiven und negativen Messbereich lieferten (+1g/-0,8g), zeigen die Analog Devices Sensoren sehr gute Ergebnisse.

Tabelle 16.5: Kalibrierwerte X-Achse

<b>ACC X</b>	<b>Messwert in Bit</b>
+1g	32210
0g	26995
-1g	21740
+1g – 0g	5215
-1g – 0g	5255
Korrekturfaktor	5235

Tabelle 16.6: Kalibrierwerte Y-Achse

<b>ACC Y</b>	<b>Messwert in Bit</b>
+1g	30115
0g	24775
-1g	19660
+1g – 0g	5340
-1g – 0g	5115
Korrekturfaktor	5227,5

Tabelle 16.7: Kalibrierwerte Z-Achse

<b>ACC Z</b>	<b>Messwert in Bit</b>
+1g	33657
0g	28175
-1g	22775
+1g – 0g	5482
-1g – 0g	5400
Korrekturfaktor	5441

Tabelle 16.8: Kalibrierwerte X2-Achse

<b>ACC X2</b>	<b>Messwert in Bit</b>
+1g	29260
0g	26259
-1g	23058
+1g – 0g	3001
-1g – 0g	3201
Korrekturfaktor	3101

Tabelle 16.9: Kalibrierwerte Z2-Achse

ACC Z2	Messwert in Bit
+1g	29630
0g	26487
-1g	23371
$ +1g - 0g $	3143
$ -1g - 0g $	3116
Korrekturfaktor	3129,5

## 17 Aufbau des Quadropters

Als Nachweis der Funktionstüchtigkeit der neuen Hardware sollte diese in einem Quadropten eingesetzt werden. Ein Quadropter ist eine Schwebefluggerät mit 2 rechts bzw. 2 linksdrehenden Luftschrauben der ausschließlich durch Drehzahlveränderung der elektrischen Antriebsmotoren gesteuert wird. Quadropten sind von Hand nicht fernzusteuern und benötigen drei Drehratensensoren um einen stabilen Flug zu ermöglichen. Es handelt sich um ein inhärent instabiles Flugsystem welches durch die Basisregelung stabilisiert werden muss. Beschreibungen der Algorithmen des Steuerverfahrens können den Arbeiten [2], [3] und [5] entnommen werden.

### 17.1 Rahmenbau

Der Aufbau des Rahmen erfolgte aus Aluminiumprofilen mit quadratischem Querschnitt und 10mm Kantenlänge. Der Abstand der Motorachsen wurde mit 400mm gewählt. Zum Erreichen einer möglichst hohen Stabilität wurden die Profile jeweils einseitig ausgeklinkt. Durch die so gebildete Verschachtelung ergibt sich in Verbindung mit zwei Zentralen, jeweils unten und oben angeordneten Platten die größtmögliche Stabilität. Bei den verwendeten zentralen Platten handelt es sich um 1,5mm starkes FR4 Platinenmaterial ohne Kupferrauflage. Die Platten besitzen eine Kantenlänge von 75mm. Überflüssiges Material in Zwischenräumen wurde zur Gewichtsreduzierung entfernt. Zur Montage des Autopiloten sind zusätzliche Stege mit Befestigungsbohrungen mit 30mm Abstand parallel zu den Flugachsen vorgesehen worden, in denen die Elektronik mit Abstandsbolzen montiert werden kann. Der Entwurf der Platten erfolgte mit Eagle. Die Strukturen wurden anschließend wie bei einer normalen Leiterplattenherstellung fotolithographisch auf das Kupfer übertragen, geätzt und die Platten anhand der Kupferstrukturen ausgesägt. Dies Verfahren erwies sich als schneller und einfacher als

die Platten herkömmlich anzuzeichnen, vor allem können so die Platten leicht reproduziert werden.

Die Aluminiumprofile und Platten wurden mit Edelstahlschrauben V2A M3x16, Unterlegscheiben und Muttern verschraubt. Da Edelstahl nicht magnetisch ist, ist bei Verwendung von Edelstahlschrauben keine Beeinflussung der Magnetsensoren durch parasitäre Magnetfelder zu befürchten. Eine zuerst erdachte zusätzliche Verklebung der Profile und Platten erwies sich nach der ersten Probemontage als unnötig, da bereits nur durch die Verschraubung eine mehr als ausreichende Stabilität erreicht wurde.

Während der ersten Flugerprobungen stellte sich heraus, dass Schwingungen der Motoren auf die Sensoren einkoppelten, was zu Aufschwingen und unruhigem Flugverhalten führte. Die Montage auf Abstandsbolzen direkt auf dem Rahmen wurde daher wieder verworfen und die Elektronik mit Abstandsbolzen auf einer Sperrholzplatte montiert, welche wiederum flächig auf Dämpfungsschaumstoff geklebt wurde. Die Schaumstoffplatte kann nun wiederum auf den Rahmen aufgeklebt werden, eventuell störende Schrauben sind vorher in dem Schaumstoff auszusparen. Durch diese Maßnahme konnte das Einkoppeln der Schwingungen beseitigt werden.

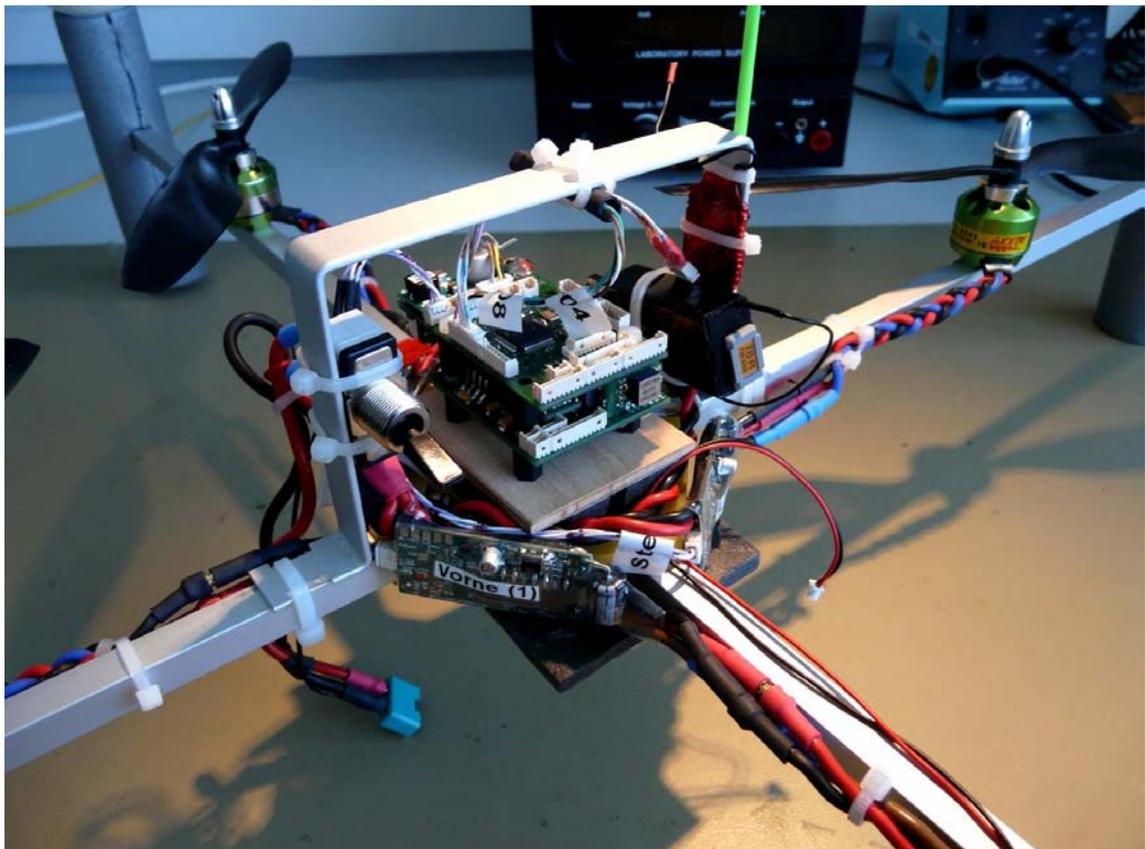
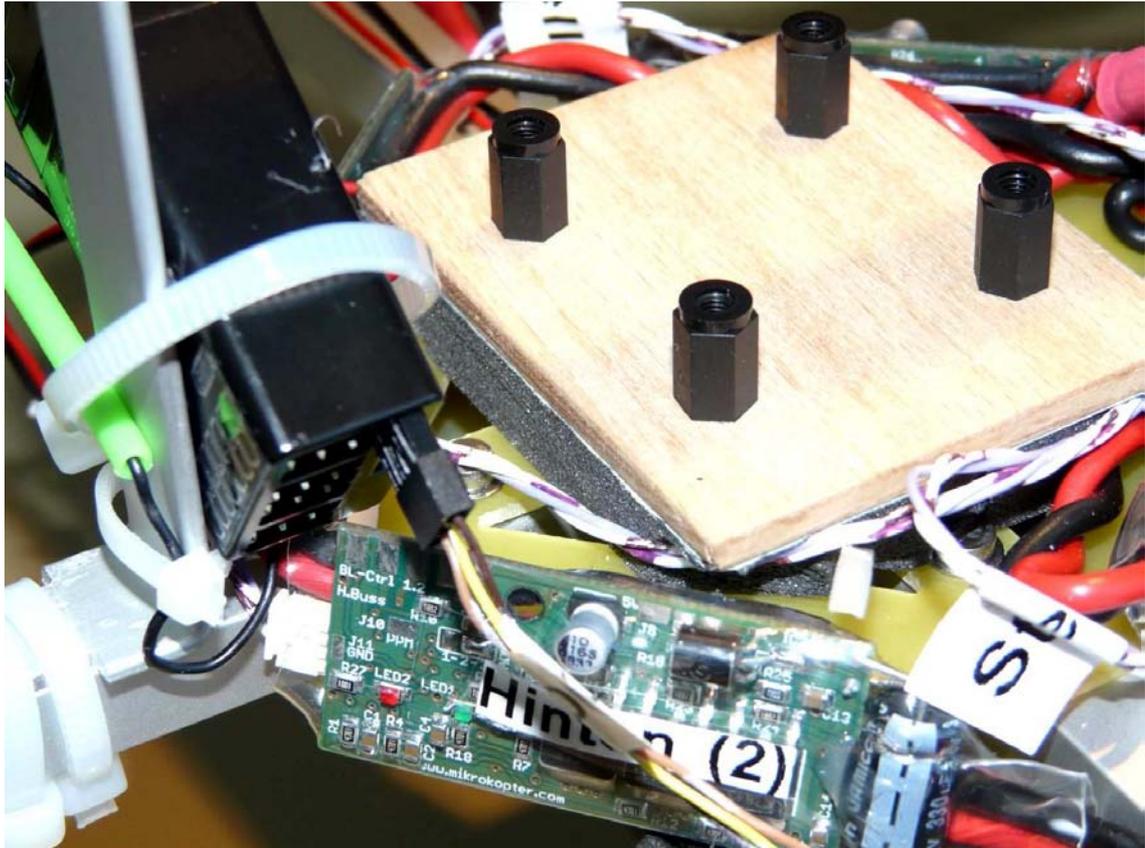
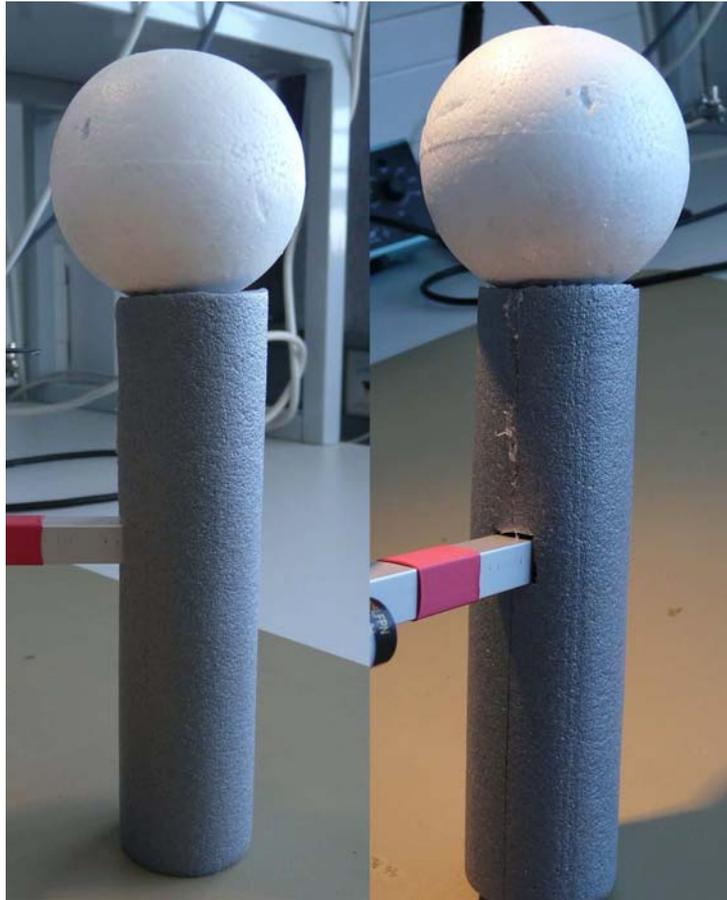


Bild 17.1: Auf Quadropterrahmen Montierter Autopilot



*Bild 17.2: Auf Schaumstoff montierter Träger für die Elektronik*

Das Landegestell, sowie der Überschlageschutz bestehen aus Isolationsschaumstoff für Wasserleitungen. Montiert wurden die Schaumstoffröhren an 8mm starke Rundhölzer, welche durch Bohrungen an den Auslegerenden gesteckt und durch Klebstoff gegen verrutschen gesichert wurden. Während nach unten der Schaumstoff über das Holz überragt, stehen am oberen Ende die Holzstäbe über, auf deren Enden 50mm Styroporkugeln aufgesteckt sind.



*Bild 17.3: Landegestell und Überschlageschutz*



*Bild 17.4: Fertig aufgebauter Quadropters*

## **18 Erfahrungen bei der Inbetriebnahme und Integration**

### **18.1 Layoutfehler**

#### **18.1.1 Fehlerhafte Bauteile**

Beim Erstellen der Bauteile für den Platinenverbinder sowie für die Beschleunigungssensoren wurde vergessen, die Kupferflächen, welche bei den Platinenverbindern zur mechanischen Stabilisierung, sowie die Kupferfläche, welche sich mittig unter den Sensoren befindet, von Lötstoplacklack auszusparen. Bei den gelieferten Platinen war es somit von Nöten den Lack an den betreffenden Stellen vorsichtig mechanisch mit einem Messer zu entfernen und die freigelegte Kupferfläche anschließend zu verzinnen.

Der Fehler wurde in der Bauteilebibliothek behoben und ist somit auch in der alten Platinenversion behoben.

#### **18.1.2 Leiterbahnfehler**

Die Leiterbahn V+ von X404 nach X405 ist zu dünn geroutet worden. Anstatt der eigentlich notwendigen Breite von 16mil wurden nur 6mil verwendet. Da es sich hier nur um ein kurzes Stück handelt und die Strombelastung nicht allzu groß ist sind keine Maßnahmen erforderlich. Der Fehler ist als Schönheitsfehler anzusehen.

Der Fehler ist in Platinenversion 1.1 behoben.

### **18.2 Schaltungsfehler**

#### **18.2.1 Verstärkerstufe Staudrucksensor**

Bei der Inbetriebnahme des Differenzdrucksensors zeigte sich eine Vollaussteuerung des ADC. Eine Kontrolle des analogen Eingangssignals ebenfalls einen zu hohen Wert und keine Veränderung bei Druckänderungen. Schon ohne Druckänderung wurde die anliegende Eingangsspannung soweit erstärkt, dass diese den maximalen Eingangspegel des ADC überschritt.

Bei Kontrolle der Schaltung wurde festgestellt, dass ein Widerstand des Differenzverstärkers vergessen wurde und so die Verstärkung so sehr hohe Werte annahm. Der fehlende Widerstand konnte nachbestückt werden, indem er „huckepack“ auf C309 aufgelötet wurde. In der Platinenversion 1.1 ist der Zusätzliche Widerstand als R310a eingefügt worden.

Die Überprüfung und Korrektur der Widerstandswerte erfolgte durch eine Pspice Simulation. Ziel war ein Ausgangsspannungsverlauf von möglichst 0V – 4V bei einer Eingangsspannungsänderung von 2,5V – 2,75V, was einem Druckbereich von 0mbar – 2,5mbar entspricht.

Die Simulation zeigte zu Beginn einen guten jedoch nicht idealen Verlauf. Durch eine Anpassung der Widerstandswerte konnte dieser auf ca. 0,8V – 3,9V eingestellt werden, was als gut anzusehen ist. Die ermittelten Widerstände wurden bestückt und eine Funktionsprüfung zeigte eine gute Reaktion auf Druckänderungen. Idealerweise sollte noch ein Abgleich per Staudruckrohr im Windkanal oder in einer Druckkammer erfolgen. Hierbei sollte auch auf eventuelle Nichtlinearitäten in der Verstärkung geachtet werden. Aufgrund der Komplexität dieses Prozess wurde dies in dieser Arbeit jedoch nicht weiter behandelt.

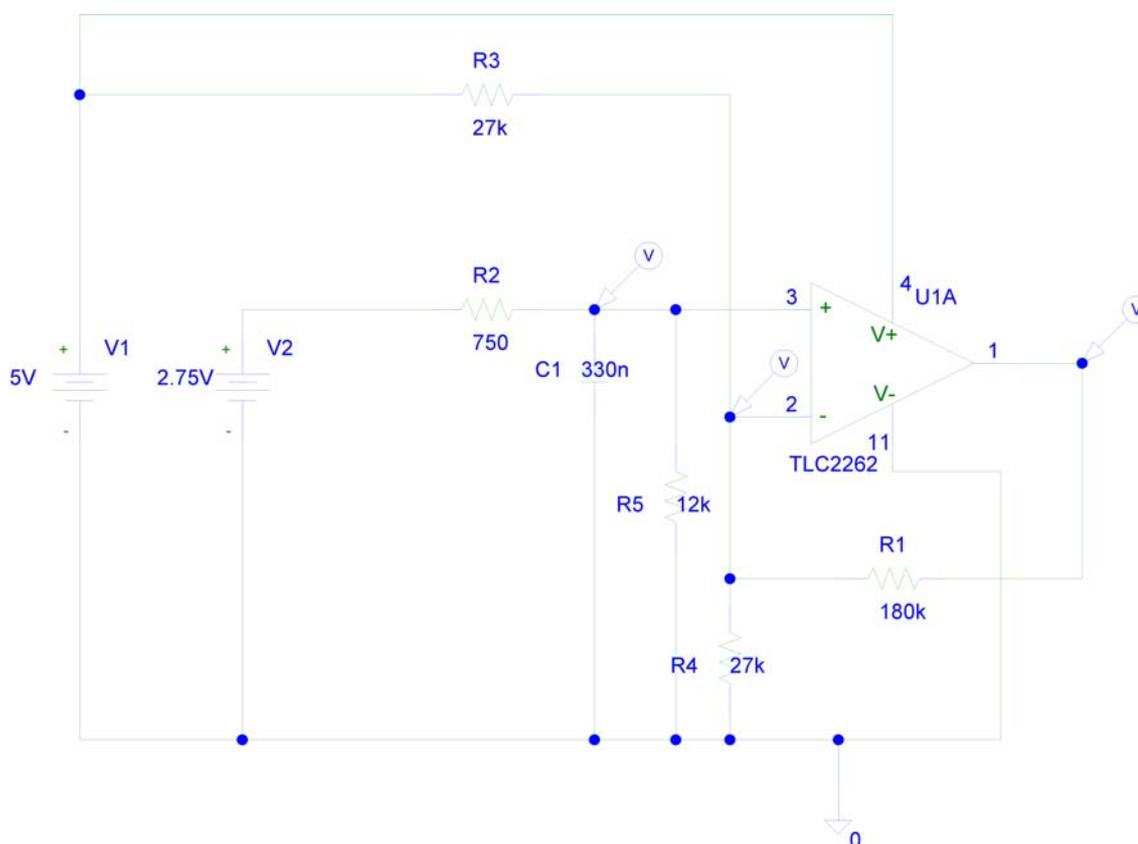


Bild 18.1: Simulationsschaltung

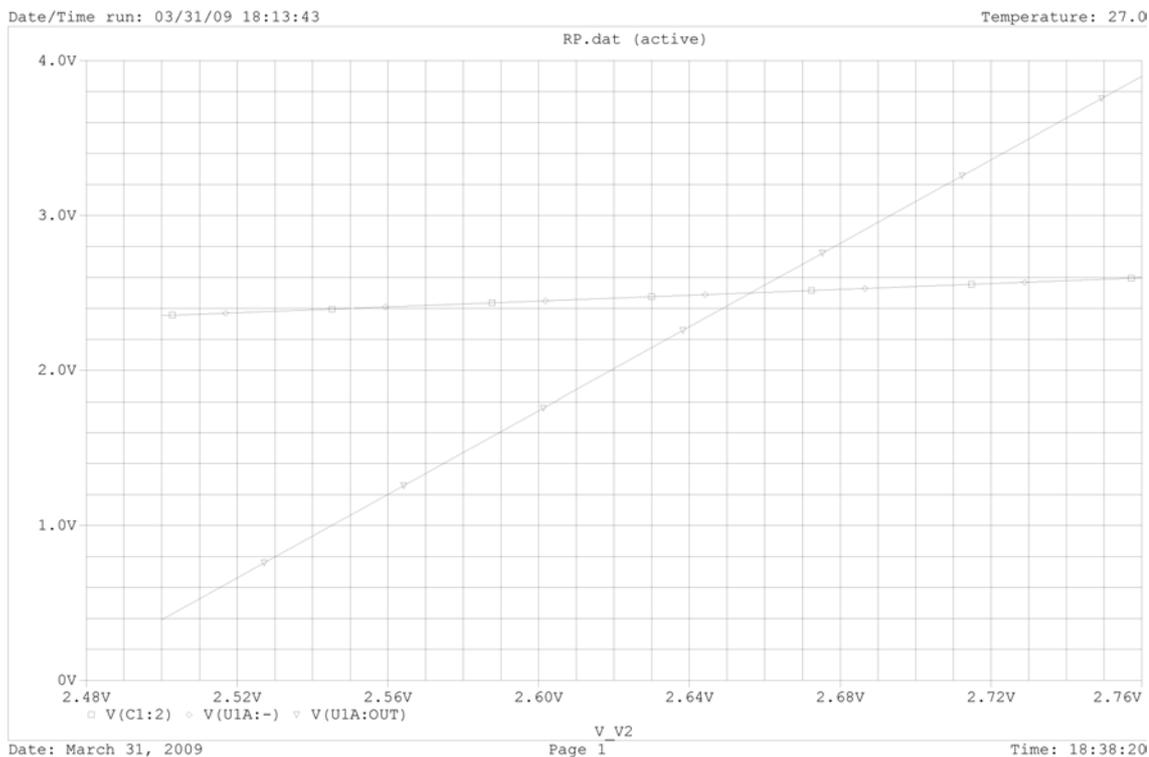


Bild 18.2: Ausgangsspannung zu Eingangsspannung

## 18.3 Sonstige Fehler

### 18.3.1 Schwingen mehrere Versorgungsspannungen

Bei den eingelesenen Werten des Differenzdrucksensors fiel eine starke Schwankung der Werte zwischen annähernd 0 und annähernd Maximum auf, die erst einmal nicht zu erklären waren. Eine Überprüfung des Eingangssignals am ADC per Oszilloskop ergab einen rechteckförmigen Signalverlauf, welcher die Schwankung erklärte (Bild 18.3). Die Werte die der ADC lieferte, waren somit scheinbar korrekt und ein Softwarefehler unwahrscheinlich. Die Kontrolle des Eingangssignal am Operationsverstärkereingang und des Ausgangssignals am Differenzdrucksensor zeigten einen sägezahnförmigen Verlauf mit einer Frequenz von ca. 420Hz. Eine weitere Kontrolle der analogen 5V Betriebsspannung zeigte einen überlagerten Sägezahn von 1,8V (Bild 18.4). Es lag nun nahe, dass der Spannungsregler ein Schwingen verursachte. Eine Kontrolle der Schaltung, des Layouts und des Datenblatts ergab jedoch keinen Fehler. Nach Datenblattangaben [D3] war die Beschaltung so gewählt, dass sich der Regler in einem absolut stabilen Bereich befindet. Auch die verwendeten Kondensatoren waren für diesen Anwendungszweck geeignet. Linear Technology schreibt für die Ausgangskondensator einen maximalen ESR von  $3\Omega$  vor. Die verwendeten Kondensatoren haben einen ESR von  $1,8\Omega$  (10V) bzw.  $1,5\Omega$  (6,3V), und sind somit geeignet.

Da der stabile Bereich maßgeblich durch den Bypasskondensator bestimmt wird, welcher im Normalfall das Restrauschen noch einmal signifikant verringern soll, und mit größerem Wert die Instabilität laut Datenblatt zunimmt, wurde der Bypasskondensator zunächst vollständig entfernt. Das Entfernen des Kondensator behob das Schwingen, jedoch ist auch das extrem geringe Restrauschen, weshalb der Regler ursprünglich eingesetzt wurde, so nicht zu erreichen.

Die Kontrolle der anderen fünf Regler des gleichen Typ ergab auch bei weiteren Reglern ein deutliches Schwingen (Tabelle 18.1) in Sägezahnform, jedoch geringer als bei den analogen 5V, während 3 Regler unterhalb der Rauschgrenze des Oszilloskop blieben. Bei schwingenden und nicht schwingenden Reglern handelt es sich teilweise um benachbarte Regler mit gleichem Layout, so dass ein grundlegendes Problem im Layout für die Schwingungsursache unwahrscheinlich ist.

Die Ursache ist noch weiter zu erforschen. Sollte das Schwingen nicht unter Kontrolle gebracht werden können, ist ein Austausch der Regler vom Typ LT1761 gegen Pinkompatible Regler des Typs LP2992 zu erwägen.

*Tabelle 18.1: Größe überlagerter Schwingung*

<b>Versorgungsspannung</b>	<b>Schwingung in mV</b>
MCU 3,3V analog	50
MCU 5,0V analog	300
IMU 3,3V analog	200
IMU 3,3V digital	50
IMU 5,0V analog	1850
IMU 5,0V digital	50

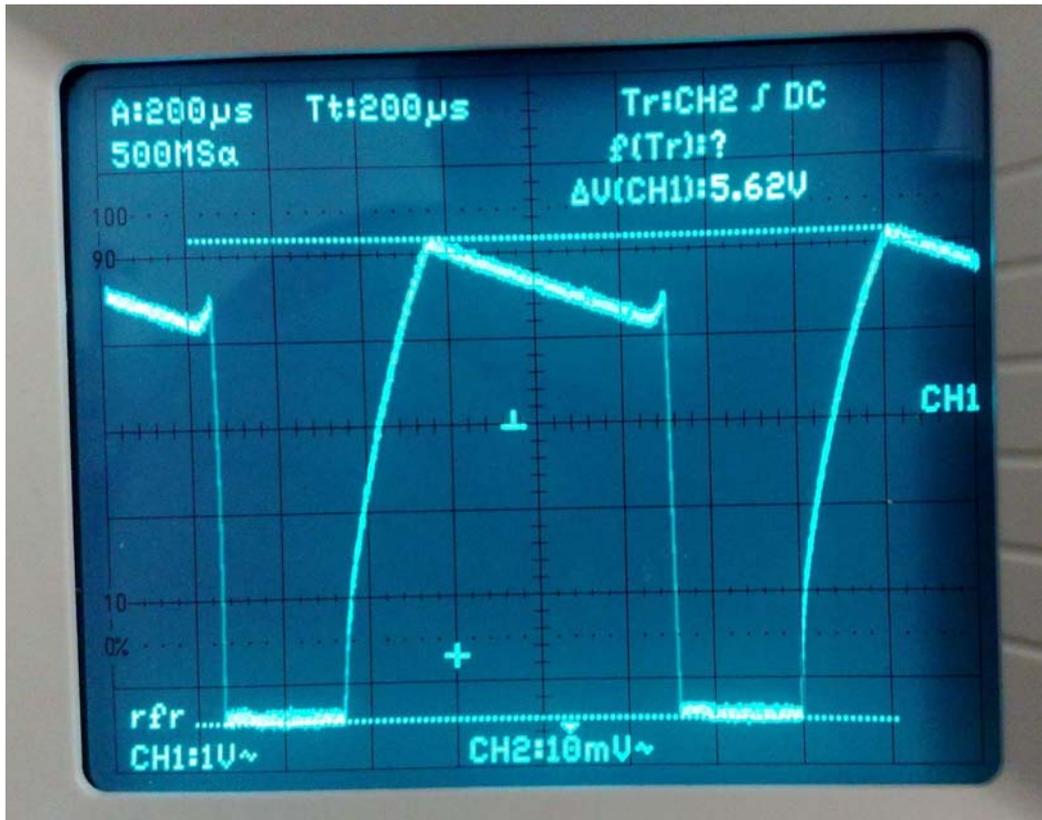


Bild 18.3: Rechteckförmiges Signal am Differenzdrucksensor

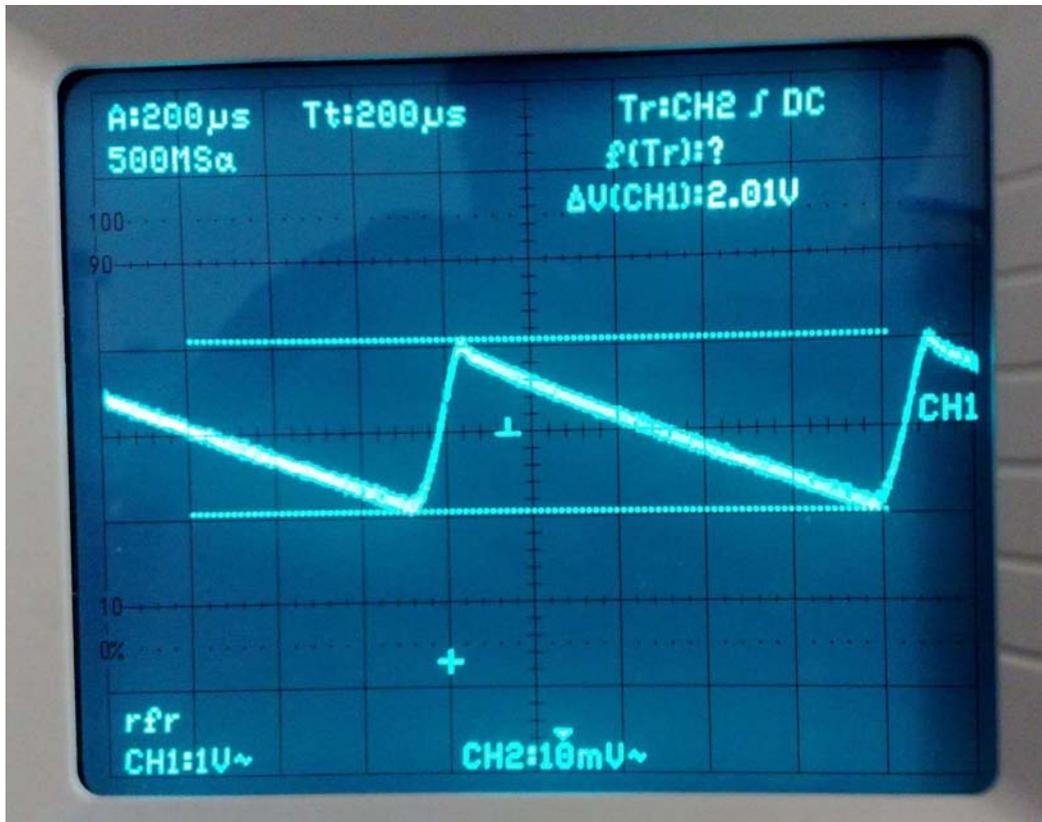


Bild 18.4: Sägezahnförmige Schwingungen auf der 5V Versorgungsspannung

### 18.3.2 Einbrüche in der digitalen 5V Versorgung

Bei den Untersuchungen der Schwingungen auf den Versorgungsspannungen fiel auf, dass auf der digitalen 5V Versorgung der IMU, nadelförmige Einbrüche der Spannung auf ca. 4V vorhanden waren. Diese Einbrüche traten Zeitgleich mit dem Flip-Impuls der Magnetsensoren auf. Die Spannungsversorgung war nicht in der Lage den Ladestrom der Kondensatoren nach einem Flip-Impuls auszugleichen. Daher wurde die Versorgung mit einem 150µF Tantalelektrolytkondensator gepuffert, was das Problem nahezu behob.

### 18.3.3 Störungen auf dem I<sup>2</sup>C Bus

Während der ersten Inbetriebnahme und Tests der Motoransteuerung fiel ein Stottern der Motoren auf. Während der Tests der Motorsteller war jedoch ein gleichmäßiger Lauf der Motoren festgestellt worden.

Die Überprüfung des Signalverlaufs am I<sup>2</sup>C Bus ergab, dass der Signalpegel nicht immer 0V erreichte sondern nur etwa 1V. Der Verdacht fiel hierbei auf die ungewöhnlich niedrigen Pullup-Widerstände, welche im Datenblatt für den PCA9306 [31] vorgeschlagen wurden. Es war durchaus möglich, dass die auf den Motorstellern verbauten ATMEGA8 Mikrocontroller nicht in der Lage waren, den Signalpegel vollständig auf 0V zu ziehen.

Da bereits ein Prototyp des Pegelwandler mit eigentlich falsch bestückten PullUp Widerständen einwandfrei funktionierte, wurde beschlossen, die funktionierenden Werte von 3,3kΩ zu übernehmen.

Nach Austausch der Widerstände waren keine Störungen mehr messbar, die Funktion somit einwandfrei.

### 18.3.4 Motorstörungen

Während der Inbetriebnahme für den Flugbetrieb stellten sich an den Motoren zeitweise merkwürdige Störungen an den Motoren ein, welche sich darin äußerten, dass die Motoren stotterten. Die Motoren liefen schlagartig kurzzeitig auf eine höhere Drehzahl um danach sofort wieder in der Solldrehzahl zu arbeiten. Zwar hatten diese Störungen keine oder nur geringe Auswirkungen auf die Flugeigenschaften, jedoch ist eine Beseitigung obligatorisch. Problematisch stellte sich heraus, dass der Fehler nicht dauerhaft vorhanden war, was die Fehlersuche erschwerte.

Für die Fehlersuche musste zunächst ermittelt werden, ob die Störungen durch Sensorwerte oder durch die Fernbedienung oder durch die Hardware allgemein verursacht wurden. Hierzu wurde zunächst die Einbindung der Sensordaten der Beschleunigungssensoren und Gyroskope unterbunden, so dass nur noch Signale der Fernsteuerung Einfluss nahmen. Im Ergebnis waren danach keine Störungen mehr feststellbar. Auch nach Einbindung der Beschleunigungssensoren zeigte eine einwandfreie Funktion. Sobald jedoch wieder die Gyroskope mit eingebunden wurden

trat der Fehler wieder auf. Eine selektive Einkopplung der einzelnen Gyroskope ergab, dass lediglich das X- und Y-Gyroskop die Motorstörungen verursachten, das Z-Gyroskop hatte keinen negativen Einfluss.

Es folgte eine grafische Untersuchung der vom ADC ausgehenden Rohdaten, ob in diesen bereits Ausreißer zu erkennen sind, welche die Störungen verursachen. Bild 18.5 zeigt die aufgenommenen Rohdaten. Der Quadrokopter wurde zur Erzeugung von Ausschlägen nacheinander auf jeder Achse gedreht. Zu erkennen sind lediglich sehr wenige geringe Ausreißer, welche nicht die Motorstörungen, die in erheblich größerer Anzahl auftraten, erklären.

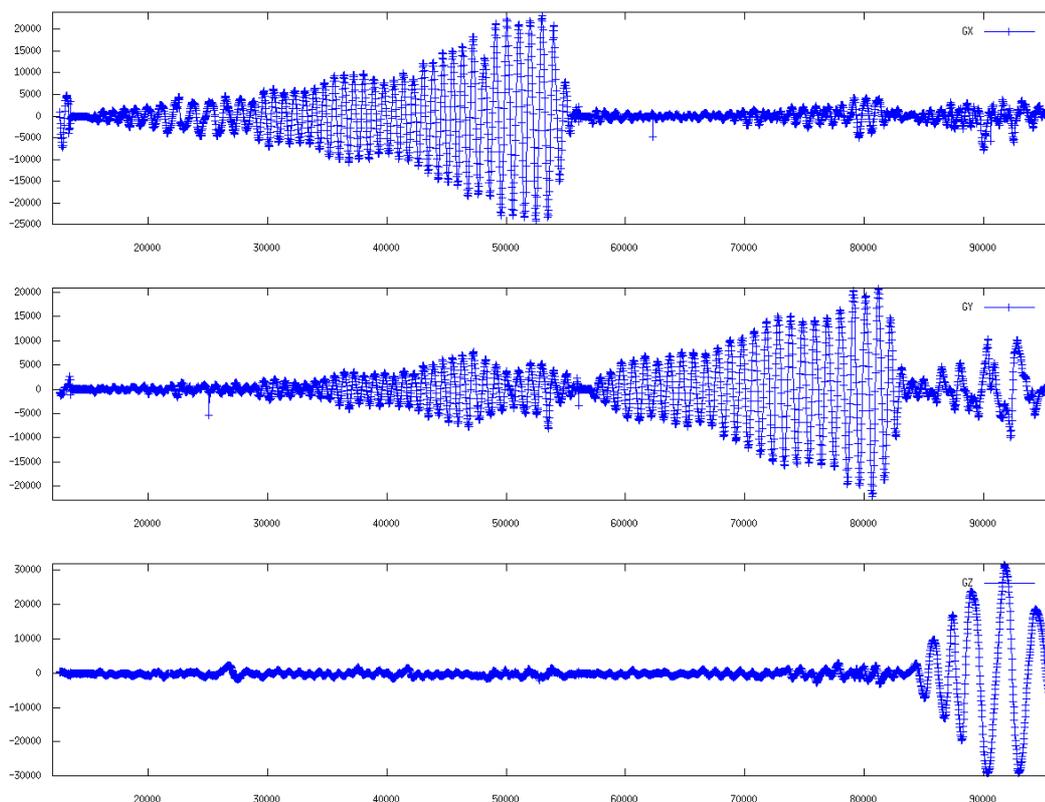


Bild 18.5: Gyroskoprohdaten

Zum Ausschließen von Fehlern in den Umrechnungsfunktionen wurden nun die Ergebnisse der Umrechnungsfunktionen grafisch dargestellt (Bild 18.6). Auch hier sind keine Ausreißer zu beobachten. Zu beachten ist, dass Bild 18.5 und Bild 18.6 während verschiedener Aufzeichnungen entstanden und daher nicht direkt miteinander vergleichbar sind.

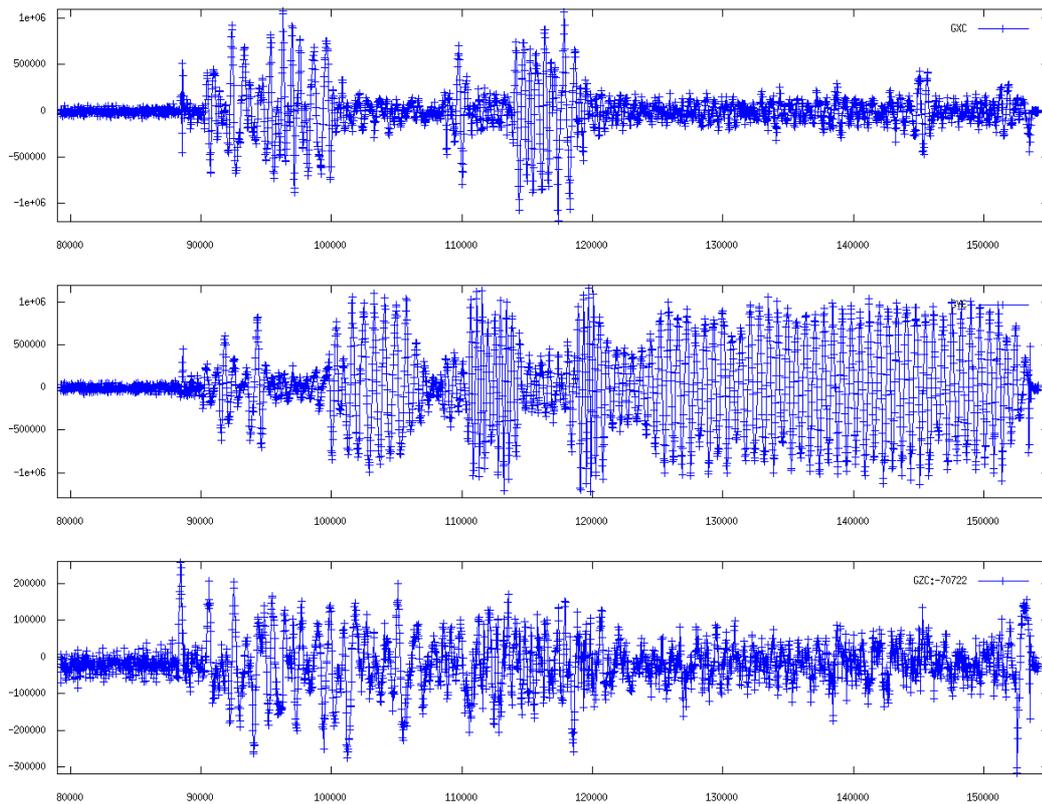


Bild 18.6: Ergebnisse der Umrechnungsfunktionen

Bild 18.7 zeigt die Werte, welche an die Motorsteller gegeben werden. Erst hier sind deutlich die Ausreißer zu sehen, welche für die Störungen der Motoren verantwortlich sind.

Trotz ausgiebiger Fehlersuche konnte bis zum Ende der Arbeit keine eindeutige Fehlerursache gefunden werden. Eine Möglichkeit besteht darin, dass es in den Algorithmen des Mixers, welcher alle Information mischt, die an die Motoren geführt werden, zu Fehlern kommt. Des weiteren ist ein Hardwarefehler, insbesondere eine schlechte Lötstelle nicht auszuschließen, sind doch komplexe Bauteile mit anspruchsvollen Gehäusen verbaut worden. Aufgrund des komplexen Aufbaus der IMU wurde entschieden, diese für eine eventuelle Fehlersuche nicht zu zerlegen, sondern vorerst auf die Fertigstellung der zweiten IMU zu warten, bei der zur Zeit noch einige Sensoren fehlten, und vorerst zu untersuchen, ob derartige Fehler auch bei ihr auftreten.

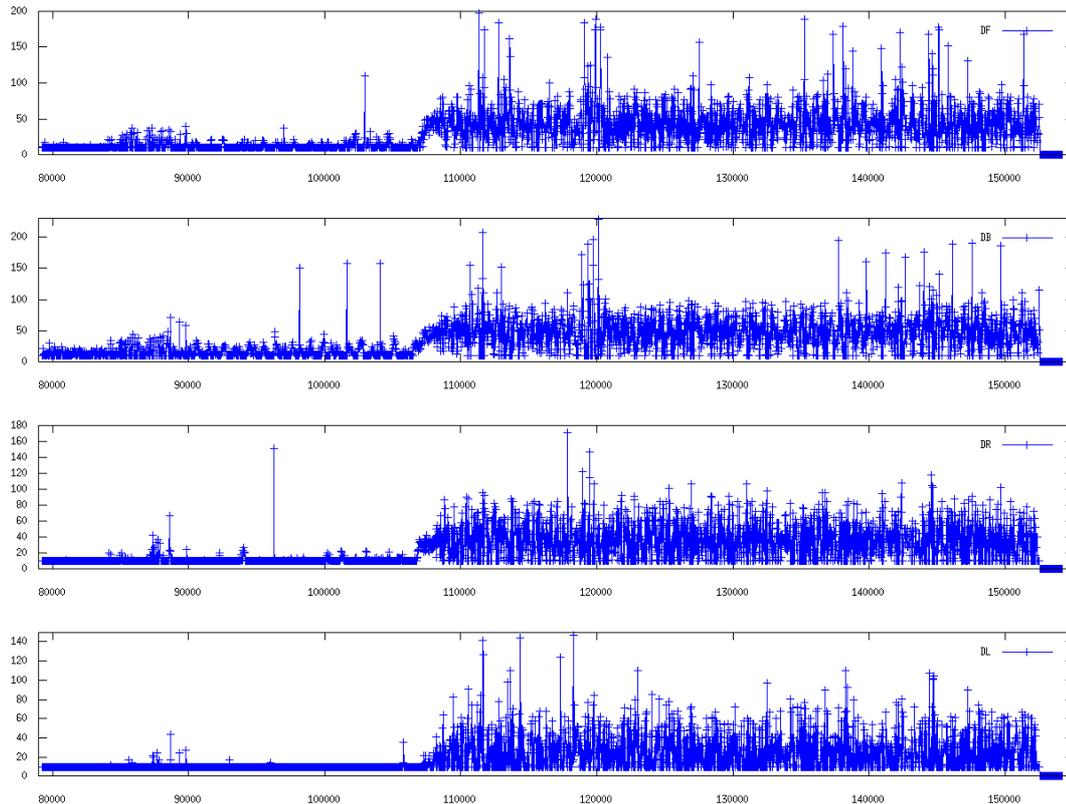


Bild 18.7: Motorstellerwerte mit Ausreißern

## 19 Flugbetrieb

### 19.1 Inbetriebnahme für den Flugbetrieb

Nach der elektrischen Inbetriebnahme erfolgte, wie beschrieben, das Anpassen der Software und Schreiben der SPI Treiber zum Auswerten der Sensoren. Nachdem alle Sensoren erfolgreich gelesen werden konnten, erfolgte die Integration des Offsetabgleich sowie das Integrieren der Konvertierungsfunktion, um die als Bit-Wert vorliegenden Sensordaten in physikalische Größen zu konvertieren, da die Regelalgorithmen und der Kalmanfilter diese definierten, für jede Hardware identischen, benötigen.

Vor dem ersten Testflug erfolgte die Kontrolle der Wirkrichtungen. Ist eine Wirkrichtung nicht korrekt, so führt z.B. ein Signal von einem Gyroskop nicht zum Gegensteuern gegen die Lageänderung sondern unterstützt noch die Lageänderung, was sofort zu einem Absturz führen würde. Die einfachste Methode die Wirkrichtungen zu testen ist hierbei den Quadrocopter bei geringen Propellerdrehzahlen, fest in der Hand gehalten, Lageänderungen auszusetzen. Steuert er diesen gegen und ist er kaum aus

seiner Lage zu bringen so ist die Wirkung korrekt. Kippt er dagegen sofort weg ist die Wirkrichtung falsch. Die Gyroskope lassen sich hierbei durch schnelle Lageänderungen testen, während die Beschleunigungssensoren auf langsame Bewegungen reagieren und durch die Winkelbestimmung versuchen den Quadrokopter wieder aufzurichten. In einigen Flugversuchen wurden die Reglerparameter noch optimiert.

## 19.2 Flugbetrieb

Zum Abschluss der Arbeit war es möglich, stabil und kontrolliert per Fernbedienung zu fliegen. Die praktische Eignung der innerhalb dieser Arbeit entstanden Hard- und Software ist somit nachgewiesen. Bild 19.1 zeigt den Quadrokopter im Schwebeflug im Flur des Elektrotechnikhochhauses der Hochschule.

Ein Video des Erstflugs ist unter <http://vimeo.com/3781642> abrufbar.



*Bild 19.1: Quadrokopter im Flug*

## 20 Zusammenfassung

Innerhalb dieser Arbeit konnte ein für alle Arten von Mikrofluggeräten geeigneter Autopilot konzipiert, entwickelt und anhand eines neu aufgebauten Quadropters praktisch erprobt werden.

Um die Entwicklungsziele zu erreichen wurde eine Marktanalyse von Drehraten-, Beschleunigungs-, Luftdruck- Differenzdruck-, Magnet- und Temperatursensoren durchgeführt und besonders geeignete Sensoren ausgewählt. Bei den Beschleunigungssensoren können unterschiedliche Sensoren bezüglich Achsenzahl und Messbereich bestückt werden, um verschiedenen Anforderungen zu erfüllen.

Bei der Anordnung der Sensoren und des Mikrocontrollers wurde eine Realisierung auf zwei getrennten Leiterplatten gewählt um die Funktionen Messen und Verarbeiten räumlich zu trennen und durch Nutzung des SPI-Busses nur wenigen Verbindungen zu benötigen. Durch die Verwendung von drei Analog-Digital-Umsetzern mit insgesamt 17 Eingängen konnten Anschlüsse des Mikrocontrollers LPC2148 für weitere Aufgaben genutzt werden. So sind gegenüber der bisherigen Lösung zusätzliche JTAG-, I2C- und SPI-Schnittstellen nutzbar.

Es wurden Treiberprogramme für die neue Hardware entwickelt und getestet, sowie die bestehenden Steuer- und Regelprogramme für den ferngesteuerten Flug des aufgebauten Quadropters angepasst.

Die neu entwickelte IMU lässt sich universell auch für andere Mikrocontroller und Rechnersysteme anwenden. So ist geplant, sie für das Forschungsvorhaben an der Hochschule Bremen „Autonome Flächenflugzeuge für die Klimaforschung“ in Verbindung mit einem LINUX-Rechner einzusetzen.

Hervorzuheben ist, dass der neue Autopilot mit MEMS-Sensoren, gegenüber der bisher verwendeten Lösung mit thermooptischen Lagemessung, allwetterfähig ist und so die Einsatzmöglichkeiten deutlich erweitert sind.

## 21 Anhang

### 21.1 Referenzliste

#### 21.1.1 Literaturerzeichnis

- [1] Studienarbeit, Henning Sauerland, Andreas Dei, SS2007, Hochschule Bremen
- [2] Diplomarbeit, Andreas Kleinert, SS2008, Hochschule Bremen
- [3] Bachelorthesis, Christoph Niemann, WS2008/2009, Hochschule Bremen
- [4] Forum <http://www.mikrokoetter.de>, <http://forum.mikrokoetter.de/topic-post67563.html>, Post 29ff, 19.03.09
- [5] Waypoint Navigation, Stabilization and Sensor Drift Compensation of VTOL MAV, Pfr. Dr. Heinrich Warmers, Andreas Kleinert, Juli 2008
- [6] Wikipedia, Strömung nach Bernoulli und Venturi, [http://de.wikipedia.org/wiki/Strömung\\_nach\\_Bernoulli\\_und\\_Venturi](http://de.wikipedia.org/wiki/Strömung_nach_Bernoulli_und_Venturi), 03.12.2008
- [7] Semesterprojekt 8, WS2006/2007, <http://www.isms.hs-bremen.de/dokuwiki/doku.php?id=elem:projekte:projekt8:sensorgruppe:sensoren>
- [8] <http://www.mikrokoetter.de/ucwiki/MK3Mag>, 15.12.2008
- [9] Digikey, <http://www.digikey.de>, 08.12.2008
- [10] Wikipedia, <http://de.wikipedia.org/wiki/Lithium-Polymer-Akkumulator>, 04.12.2008
- [11] Elektor 1/2006, S. 32ff, SMD-Lötöfen
- [12] <http://www.mikrokoetter.de/ucwiki/BrushlessCtrl>, 28.02.2009
- [13] <http://paparazzi.enac.fr/wiki/Tiny>, 27.11.2008
- [14] SVN, Akaflieg, Labor ELEM, Hochschule Bremen, <https://www.akaflieg.hs-bremen.de/trac/akaflieg/browser>
  
- [D1] Datenblatt, Linear Technology, LTC2440, Rev. C
- [D2] Datenblatt, Linear Technology, LTC2400, Rev. A
- [D3] Datenblatt, Linear Technology, LT1761, Rev. C
- [D4] Datenblatt, Analog Devices, ADXRS610, Rev. 0
- [D5] Applikationsbericht, Freescale Semiconductor, AN1646, Rev. 2
- [D6] Datenblatt, Maxim, MAX6605, Rev. 2
- [D7] Datenblatt, NXP, LPC2148, Rev. 01

- [D8] Benutzerhandbuch, NXP, LPC2148, Rev. 02
- [D9] Datenblatt, NXP, LPC2294, Rev. 7
- [D10] Datenblatt, Texas Instruments, TMS320F28335, SPRS439E
- [D11] Datenblatt, Philips Semiconductors, KMZ51, 13.06.2000
- [D12] Applikationsbericht, Philips Semiconductors, AN00022, 30.03.2000
- [D13] Datenblatt, PNI, MicroMag3, 04.2005
- [D14] Datenblatt, Maxim, MAX1168, Rev. 0
- [D15] Datenblatt, Texas Instruments, PTH08080, SLTS235C
- [D16] Datenblatt, InvenSense, IDG-300, 14.08.07
- [D17] Datenblatt, MuRata, ENC-03, 29.03.99
- [D18] Datenblatt, Analog Devices, ADXRS150, Rev. B
- [D19] Datenblatt, Analog Devices, ADXRS300, Rev. B
- [D20] Datenblatt, STMicroelectronics, LISY300AL, Rev. 1
- [D21] Datenblatt, Melexis, MLX90609-R2, Rev. 002
- [D22] Datenblatt, Analog Devices, ADXL330, Rev. A
- [D23] Datenblatt, Analog Devices, ADXL320, Rev. 0
- [D24] Datenblatt, Analog Devices, ADXL323, Rev. 0
- [D25] Datenblatt, Freescale Semiconductor, MPXH6115, Rev. 3
- [D26] Datenblatt, Freescale Semiconductor, MPXV7002, Rev. 0
- [D27] Datenblatt, National Semiconductor, LM20, 30.10.2008
- [D28] Datenblatt, National Semiconductor, LM1117, 04.2006
- [D29] Datenblatt, Texas Instruments, TPS2015, SLVS159B
- [D30] Datenblatt, Texas Instruments, PCA9306, SCPS113I

### 21.1.2 Abkürzungsverzeichnis

ADC	Analog-Digital-Converter
DSP	Digitaler Signal Prozessor
GPS	Global Positioning System
I/O	Input/Output
IC	Inter-Integrated Circuit
IMU	Inertial Measurement Unit

ISR	Interrupt Service Routine
MEMS	Micro-Electro-Mechanical System
MM3	MicroMag3
SPI	Serial Peripheral Interface
USB	Universal Serial Bus

### 21.1.3 Abbildungsverzeichnis

<i>Bild 3.1:</i>	<i>Blockschaltbild LPC2148 [D7]</i> .....	15
<i>Bild 3.2:</i>	<i>Blockschaldbild LPC2294 [D9]</i> .....	16
<i>Bild 3.3:</i>	<i>Blockschaltbild TMS320F28335 [D10]</i> .....	18
<i>Bild 6.1:</i>	<i>Blockschaltbild des Autopiloten</i> .....	30
<i>Bild 7.1:</i>	<i>Versuchsaufbau für Gyroskop ADXRS610</i> .....	34
<i>Bild 8.1:</i>	<i>Achsenanordnung der Leiterplatte</i> .....	47
<i>Bild 9.1:</i>	<i>Pappmodell mit beiden übereinander gelegten Teilen</i> .....	49
<i>Bild 9.2:</i>	<i>Pappmodell mit beiden einzelnen Teilen</i> .....	50
<i>Bild 10.1:</i>	<i>Testplatine für BGA Lötversuche</i> .....	55
<i>Bild 10.2:</i>	<i>Aufbereitete Lötpaste</i> .....	56
<i>Bild 10.3:</i>	<i>Testplatine mit aufgetragener Lötpaste</i> .....	56
<i>Bild 10.4:</i>	<i>Nach verlöten wieder entferntes BGA IC</i> .....	58
<i>Bild 10.5:</i>	<i>LISY300Al, oben links: Paste aufgetragen, oben rechts: Platziert, unten: Verlötet</i>	59
<i>Bild 10.6:</i>	<i>Y-Sensorplatine, bestückt noch nicht verlötet</i> .....	60
<i>Bild 10.7:</i>	<i>Y-Sensorplatine, verlötet</i> .....	61
<i>Bild 10.8:</i>	<i>Fertig bestückter Sensorplatinensatz</i> .....	61
<i>Bild 11.1:</i>	<i>MCU Platine vor dem Reflowprozess</i> .....	62
<i>Bild 11.2:</i>	<i>MCU Platine direkt nach dem Reflowprozess</i> .....	63
<i>Bild 11.3:</i>	<i>Oberseite MCU</i> .....	64
<i>Bild 11.4:</i>	<i>Unterseite MCU</i> .....	64
<i>Bild 11.5:</i>	<i>Oberseite IMU</i> .....	65
<i>Bild 11.6:</i>	<i>Unterseite IMU</i> .....	65
<i>Bild 11.7:</i>	<i>MCU und IMU gesteckt</i> .....	66
<i>Bild 11.8:</i>	<i>MCU und IMU gesteckt</i> .....	66
<i>Bild 12.1:</i>	<i>Bereits eingeschrumpfter und am Quadrocopter montierter Motorsteller</i>	67
<i>Bild 12.2:</i>	<i>GPS-Modul</i> .....	68
<i>Bild 12.3:</i>	<i>Verbindung GPS-Modul mit Trägerplatine</i> .....	68
<i>Bild 12.4:</i>	<i>Programmieradapter</i> .....	69
<i>Bild 13.1:</i>	<i>LISY300AL im Funktionstest</i> .....	71
<i>Bild 13.2:</i>	<i>ADXRS610 im Funktionstest</i> .....	71
<i>Bild 15.1:</i>	<i>Berechnetes Heading</i> .....	91
<i>Bild 15.2:</i>	<i>Rohdaten der Sensoren</i> .....	91

<i>Bild 17.1:</i>	<i>Auf Quadrokopterrahmen Montierter Autopilot.....</i>	<i>100</i>
<i>Bild 17.2:</i>	<i>Auf Schaumstoff montierter Träger für die Elektronik.....</i>	<i>101</i>
<i>Bild 17.3:</i>	<i>Landegestell und Überschlagsschutz.....</i>	<i>102</i>
<i>Bild 17.4:</i>	<i>Fertig aufgebauter Quadrokopter .....</i>	<i>102</i>
<i>Bild 18.1:</i>	<i>Simulationsschaltung .....</i>	<i>104</i>
<i>Bild 18.2:</i>	<i>Ausgangsspannung zu Eingangsspannung .....</i>	<i>105</i>
<i>Bild 18.3:</i>	<i>Rechteckförmiges Signal am Differenzdrucksensor .....</i>	<i>107</i>
<i>Bild 18.4:</i>	<i>Sägezahnförmige Schwingungen auf der 5V Versorgungsspannung ...</i>	<i>107</i>
<i>Bild 18.5:</i>	<i>Gyroskoprohdaten .....</i>	<i>109</i>
<i>Bild 18.6:</i>	<i>Ergebnisse der Umrechnungsfunktionen.....</i>	<i>110</i>
<i>Bild 18.7:</i>	<i>Motorstellerwerte mit Ausreißern.....</i>	<i>111</i>
<i>Bild 19.1:</i>	<i>Quadrokopter im Flug .....</i>	<i>112</i>
<i>Bild 21.1:</i>	<i>Schaltplan MCU Teil 1 .....</i>	<i>125</i>
<i>Bild 21.2:</i>	<i>Schaltplan MCU Teil 2 .....</i>	<i>126</i>
<i>Bild 21.3:</i>	<i>Schaltplan IMU Teil 1.....</i>	<i>127</i>
<i>Bild 21.4:</i>	<i>Schaltplan IMU Teil 2.....</i>	<i>128</i>
<i>Bild 21.5:</i>	<i>ADXRS610 X-Sensorplatine .....</i>	<i>129</i>
<i>Bild 21.6:</i>	<i>ADXRS610 Y-Sensorplatine.....</i>	<i>130</i>
<i>Bild 21.7:</i>	<i>ADXRS610 Z-Sensorplatine.....</i>	<i>131</i>
<i>Bild 21.8:</i>	<i>LISY300AL X-Sensorplatine .....</i>	<i>132</i>
<i>Bild 21.9:</i>	<i>LISY300AL Y-Sensorplatine .....</i>	<i>133</i>
<i>Bild 21.10:</i>	<i>LISY300AL Z-Sensorplatine .....</i>	<i>134</i>
<i>Bild 21.11:</i>	<i>MLX90609-R2 X-Sensorplatine.....</i>	<i>135</i>
<i>Bild 21.12:</i>	<i>MLX90609-R2 Y-Sensorplatine .....</i>	<i>136</i>
<i>Bild 21.13:</i>	<i>MLX90609-R2 Z-Sensorplatine .....</i>	<i>137</i>
<i>Bild 21.14:</i>	<i>Servoadapter.....</i>	<i>138</i>
<i>Bild 21.15:</i>	<i>Programmieradapter .....</i>	<i>139</i>
<i>Bild 21.16:</i>	<i>GPS Modul.....</i>	<i>140</i>
<i>Bild 21.17:</i>	<i>MCU Platine, Layer 1.....</i>	<i>141</i>
<i>Bild 21.18:</i>	<i>MCU Platine, Layer 2.....</i>	<i>142</i>
<i>Bild 21.19:</i>	<i>MCU Platine, Layer 16.....</i>	<i>143</i>
<i>Bild 21.20:</i>	<i>IMU Platine, Layer 1 .....</i>	<i>144</i>
<i>Bild 21.21:</i>	<i>IMU Platine, Layer 2 .....</i>	<i>145</i>
<i>Bild 21.22:</i>	<i>IMU Platine, Layer 16.....</i>	<i>146</i>
<i>Bild 21.23:</i>	<i>ADXRS610 X-Sensorplatine .....</i>	<i>147</i>
<i>Bild 21.24:</i>	<i>ADXRS610 Y-Sensorplatine.....</i>	<i>147</i>
<i>Bild 21.25:</i>	<i>ADXRS610 Z-Sensorplatine.....</i>	<i>147</i>
<i>Bild 21.26:</i>	<i>LISY300AL X-Sensorplatine .....</i>	<i>148</i>
<i>Bild 21.27:</i>	<i>LISY300AL Y-Sensorplatine .....</i>	<i>148</i>
<i>Bild 21.28:</i>	<i>LISY300AL Z-Sensorplatine .....</i>	<i>148</i>
<i>Bild 21.29:</i>	<i>MLX90609-R2 X-Sensorplatine.....</i>	<i>149</i>
<i>Bild 21.30:</i>	<i>LISY300AL Y-Sensorplatine .....</i>	<i>149</i>
<i>Bild 21.31:</i>	<i>LISY300AL Z-Sensorplatine .....</i>	<i>149</i>

<i>Bild 21.32:</i>	<i>Servoadapter</i> .....	<i>150</i>
<i>Bild 21.33:</i>	<i>Programmieradapter</i> .....	<i>150</i>
<i>Bild 21.34:</i>	<i>GPS-Modul</i> .....	<i>150</i>
<i>Bild 21.35:</i>	<i>Programablaufplan "spi_int"</i> .....	<i>151</i>
<i>Bild 21.36:</i>	<i>PAP "sm_ctrl" Seite 1</i> .....	<i>152</i>
<i>Bild 21.37:</i>	<i>PAP "sm_ctrl" Seite 2</i> .....	<i>153</i>
<i>Bild 21.38:</i>	<i>PAP "sm_ctrl" Seite 3</i> .....	<i>154</i>
<i>Bild 21.39:</i>	<i>PAP "sm_ctrl" Seite 4</i> .....	<i>155</i>
<i>Bild 21.40:</i>	<i>PAP "sm_ctrl" Seite 5</i> .....	<i>156</i>
<i>Bild 21.41:</i>	<i>Anschlüsse Lichtschrankelektronik</i> .....	<i>157</i>
<i>Bild 21.42:</i>	<i>Position Holzunterbrecherscheibe</i> .....	<i>158</i>
<i>Bild 21.43:</i>	<i>Montage der IMU auf Kalibrierhalterung (X-, Y-, Z-Gyroskop)</i> .....	<i>159</i>
<i>Bild 21.44:</i>	<i>Montage der IMU auf Kalibrierhalterung für entgegengesetzte Drehrichtung (X-, Y-Gyroskop)</i> .....	<i>160</i>
<i>Bild 21.45:</i>	<i>Montage der IMU auf Kalibrierhalterung für entgegengesetzte Drehrichtung (Z-Gyroskop)</i> .....	<i>161</i>
<i>Bild 21.46:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse 0U/min</i> .....	<i>164</i>
<i>Bild 21.47:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse 33U/min</i> .....	<i>165</i>
<i>Bild 21.48:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse 45U/min</i> .....	<i>165</i>
<i>Bild 21.49:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse - 33U/min</i> .....	<i>166</i>
<i>Bild 21.50:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse - 45U/min</i> .....	<i>166</i>
<i>Bild 21.51:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 0U/min</i> .....	<i>167</i>
<i>Bild 21.52:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 33U/min</i> .....	<i>167</i>
<i>Bild 21.53:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 45U/min</i> .....	<i>168</i>
<i>Bild 21.54:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse - 33U/min</i> .....	<i>168</i>
<i>Bild 21.55:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse - 45U/min</i> .....	<i>169</i>
<i>Bild 21.56:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 0U/min</i> .....	<i>169</i>
<i>Bild 21.57:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 33U/min</i> .....	<i>170</i>
<i>Bild 21.58:</i>	<i>Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 45U/min</i> .....	<i>170</i>

<i>Bild 21.59: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse - 33U/min .....</i>	<i>171</i>
<i>Bild 21.60: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse - 45U/min .....</i>	<i>171</i>
<i>Bild 21.61: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse 0g .....</i>	<i>172</i>
<i>Bild 21.62: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse +1g.....</i>	<i>172</i>
<i>Bild 21.63: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse - 1g .....</i>	<i>173</i>
<i>Bild 21.64: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 0g .....</i>	<i>173</i>
<i>Bild 21.65: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse +1g.....</i>	<i>174</i>
<i>Bild 21.66: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse - 1g .....</i>	<i>174</i>
<i>Bild 21.67: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 0g .....</i>	<i>175</i>
<i>Bild 21.68: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse +1g.....</i>	<i>175</i>
<i>Bild 21.69: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse - 1g .....</i>	<i>176</i>
<i>Bild 21.70: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X2-Achse 0g .....</i>	<i>176</i>
<i>Bild 21.71: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X2-Achse +1g.....</i>	<i>177</i>
<i>Bild 21.72: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X2-Achse - 1g .....</i>	<i>177</i>
<i>Bild 21.73: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z2-Achse 0g .....</i>	<i>178</i>
<i>Bild 21.74: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z2-Achse +1g.....</i>	<i>178</i>
<i>Bild 21.75: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z2-Achse - 1g .....</i>	<i>179</i>

#### **21.1.4 Tabellenverzeichnis**

<i>Tabelle 3.1: Vor- und Nachteile verwendbarer Mikrocontroller.....</i>	<i>14</i>
<i>Tabelle 5.1: Stromaufnahme der IMU Komponenten .....</i>	<i>27</i>
<i>Tabelle 7.1: Geläufige Werte für C507.....</i>	<i>32</i>
<i>Tabelle 7.2: Selbsttestmodi .....</i>	<i>33</i>
<i>Tabelle 7.3: Erhältliche Gyroskope .....</i>	<i>33</i>

Tabelle 7.4:	Filterkondensatorauswahl .....	35
Tabelle 7.5:	Belegung X109 UART0.....	40
Tabelle 7.6:	Pinbelegung X110 UART1.....	41
Tabelle 7.7:	Pinbelegung X111 I <sup>2</sup> C0.....	41
Tabelle 7.8:	Pinbelegung X112 I <sup>2</sup> C1.....	42
Tabelle 7.9:	Pinbelegung X117 SPI1-1.....	43
Tabelle 7.10:	Pinbelegung X114 SPI1-2.....	43
Tabelle 7.11:	Pinbelegung X116 Analog .....	44
Tabelle 7.12:	Pinbelegung X107 IR0.....	44
Tabelle 7.13:	Pinbelegung X113 RC PPM IN .....	45
Tabelle 7.14:	Pinbelegung X106 CAM .....	45
Tabelle 7.15:	Pinbelegung X108 USB .....	46
Tabelle 10.1:	Parameter 1. Durchlauf.....	52
Tabelle 10.2:	Parameter 2. Durchlauf.....	53
Tabelle 10.3:	Parametervorschlag aus Datenblatt des ADXL330.....	53
Tabelle 10.4:	Erstellte Parameter aus Parametern des ADXL330 Datenblatts .....	53
Tabelle 10.5:	Endgültige Parameter.....	54
Tabelle 13.1:	Kontrolle der Versorgungsspannungen.....	70
Tabelle 14.1:	Kabelplan K-01.....	72
Tabelle 14.2:	Kabelplan K-02.....	72
Tabelle 14.3:	Kabelplan K-03.....	73
Tabelle 14.4:	Kabelplan K-04.....	73
Tabelle 14.5:	Kabelplan K-05.....	73
Tabelle 14.6:	Kabelplan K-06.....	74
Tabelle 14.7:	Kabelplan K-07.....	74
Tabelle 16.1:	Berechnung Umrechnungsfaktor X-Gyroskop.....	95
Tabelle 16.2:	Berechnung Umrechnungsfaktor Y-Gyroskop.....	96
Tabelle 16.3:	Berechnung Umrechnungsfaktor Z-Gyroskop.....	96
Tabelle 16.4:	Berechnung der Gyroskopneutralmittelwerte.....	96
Tabelle 16.5:	Kalibrierwerte X-Achse .....	98
Tabelle 16.6:	Kalibrierwerte Y-Achse.....	98
Tabelle 16.7:	Kalibrierwerte Z-Achse.....	98
Tabelle 16.8:	Kalibrierwerte X2-Achse .....	98
Tabelle 16.9:	Kalibrierwerte Z2-Achse.....	99
Tabelle 18.1:	Größe überlagerter Schwingung .....	106
Tabelle 21.1:	Gyroskope im Vergleich.....	122
Tabelle 21.2:	Beschleunigungssensoren im Vergleich 1.....	123
Tabelle 21.3:	Beschleunigungssensoren im Vergleich 2.....	124
Tabelle 21.4:	Mittelwerte für Kalibrierung X-Achse .....	162
Tabelle 21.5:	Mittelwerte für Kalibrierung Y-Achse .....	162
Tabelle 21.6:	Mittelwerte für Kalibrierung Z-Achse .....	162
Tabelle 21.7:	Mittelwerte für Kalibrierung X-Achse +1g .....	162
Tabelle 21.8:	Mittelwerte für Kalibrierung X-Achse -1g.....	163

---

<i>Tabelle 21.9:</i>	<i>Mittelwerte für Kalibrierung Y-Achse +1g.....</i>	<i>163</i>
<i>Tabelle 21.10:</i>	<i>Mittelwerte für Kalibrierung Y-Achse -1g.....</i>	<i>163</i>
<i>Tabelle 21.11:</i>	<i>Mittelwerte für Kalibrierung Z-Achse +1g.....</i>	<i>163</i>
<i>Tabelle 21.12:</i>	<i>Mittelwerte für Kalibrierung Z-Achse -1g.....</i>	<i>164</i>

## 21.2 Sensorauswahl

### 21.2.1 Gyroskope

Tabelle 21.1: Gyroskope im Vergleich

Sensor	IDG-300	Enc-03J	ADXRS610	ADXRS612	ADXRS613	ADXRS614	LISY300AL	MLX90609-R2
Hersteller	InvenSense	Murata	AD	AD	AD	AD	STM	Melexis
Measurement Range +/- in °/sec	500	300	300	250	150	75	300	300
Sensitivity/Temperature Drift in %	10	20	2	2	3	3	4	5
Nonlinearity in % of FS	1	5	0,1	0,1	0,1	0,1	0,8	0,5
Zero Level in V	1,23	1,35	2,5	2,5	2,5	2,5	1,65	2,5
Rate Noise Density in °/sec/sqrtHz	0,014	not specified	0,05	0,06	0,04	0,04	0,1	2
Sensor Resonate Frequ in kHz	12--15	not specified	14,5	14,5	14,5	14,5	4,5	8,2

### 21.2.2 Beschleunigungssensoren

Tabelle 21.2: Beschleunigungssensoren im Vergleich

Sensor	LIS202DL	LIS302DL	LIS331DL	LIS3LV02DL	ADX1330
Hersteller	STM	STM	STM	STM	AD
Analog / Digital	Digital	Digital	Digital	Digital	Analog
Messbereich +/- in g	2 & 8	2 & 8	2 & 8	2 & 6	3
Anzahl der Achsen	2	3	3	3	3
Gehäuse	LGA14	LGA14	LGA16	LGA 16	LFCSP_LQ-16
Versorgungsspannung min. in V	2,16	2,16	2,16	2,16	1,8
Versorgungsspannung max. in V	3,6	3,6	3,6	3,6	3,6
Stromaufnahme in mA	0,3	0,3	0,3	0,65	0,32
Sensitivity Temperature Drift in %	-	-	-	-	-
Sensitivity Temperature Drift in %/°C	0,01	0,01	0,01	0,025	0,015
Nonlinearity typ in % of FS	-	-	0,5	x,y=2, z=3	0,3
Nonlinearity max in % of FS	-	-	-	-	-
Zero g level in V	-	-	-	-	1,5
Zero-g Level Change Vs Temperature in mg/°C	-	0,5	0,5	0,5	1
Rate Noise Density in µg/sqrtHz	-	-	-	-	-
Rate Noise Density in µg/sqrtHz rms	-	-	-	-	x,y=280; z=350
Cross Axis in %	-	-	-	3,5	1

Sensor	ADXL204	ADXL323	ADXL320	SCAS000	MMA7260QT
Hersteller	AD	AD	AD	VTI	Freescale
Analog / Digital	Analog	Analog	Analog	Digital	
Messbereich +/- in g	1,7	3	5	3	1,5, 2, 4, 6
Anzahl der Achsen	2	2	2	3	3
Gehäuse	LCC-8	LFCSP_LQ-16	LFCSP_LQ-16	kA	QFN1622-02
Versorgungsspannung min. in V	3,3	1,8	2,4	2,36	2,2
Versorgungsspannung max. in V	3,3	5,26	5,25	3,6	3,6
Stromaufnahme in mA	0,5	0,32	0,48	0,15	0,5
Sensitivity Temperature Drift in %	0,3	-	-	-	-
Sensitivity Temperature Drift in %/°C	-	0,015	0,01	0,01	0,02
Nonlinearity typ in % of FS	0,2	0,3	0,2	1	1
Nonlinearity max in % of FS	1,25	-	-	-	-
Zero g level in V	1,65	1,5	1,5	-	1,65
Zero-g Level Change Vs Temperature in mg/°C	0,15	0,6	0,6	0,8	x=0,6; y=5,8; z=0,8
Rate Noise Density in µg/sqrtHz	-	-	-	-	-
Rate Noise Density in µg/sqrtHz rms	170	280	250	-	350
Cross Axis in %	1,5	1	2	3	5

Tabelle 21.3: Beschleunigungssensoren im Vergleich 2

Sensor	LIS202DL	LIS302DL	LIS331DL	LIS3LV02DL	ADXL330
Hersteller	STM	STM	STM	STM	AD
Analog / Digital	Digital	Digital	Digital	Digital	Analog
Messbereich +/- in g	2 & 8	2 & 8	2 & 8	2 & 6	3
Anzahl der Achsen	2	3	3	3	3
Gehäuse	LGA14	LGA14	LGA16	LGA 16	LFCSP_LQ-16
Versorgungsspannung min. in V	2,16	2,16	2,16	2,16	1,8
Versorgungsspannung max. in V	3,6	3,6	3,6	3,6	3,6
Stromaufnahme in mA	0,3	0,3	0,3	0,65	0,32
Sensitivity Temperature Drift in %	-	-	-	-	-
Sensitivity Temperature Drift in %/°C	0,01	0,01	0,01	0,025	0,015
Nonlinearity typ in % of FS	-	-	0,5	x,y=2, z=3	0,3
Nonlinearity max in % of FS	-	-	-	-	-
Zero g level in V	-	-	-	-	1,5
Zero-g Level Change Vs Temperature in mg/°C	0,5	0,5	0,5	0,2	1
Rate Noise Density in µg/sqrtHz	-	-	-	-	-
Rate Noise Density in µg/sqrtHz rms	-	-	-	-	x,y=280, z=350
Cross Axis in %	-	-	-	3,5	1

Sensor	ADXL204	ADXL323	ADXL320	SCA3000	MMA7260QT
Hersteller	AD	AD	AD	VTI	Freescale
Analog / Digital	Analog	Analog	Analog	Digital	-
Messbereich +/- in g	1,7	3	5	3	1,5, 2, 4, 6
Anzahl der Achsen	2	2	2	3	3
Gehäuse	LCC-8	LFCSP_LQ-16	LFCSP_LQ-16	kA	QFN1622-02
Versorgungsspannung min. in V	3,3	1,8	2,4	2,35	2,2
Versorgungsspannung max. in V	3,3	5,25	5,25	3,6	3,6
Stromaufnahme in mA	0,5	0,32	0,48	0,15	0,5
Sensitivity Temperature Drift in %	0,3	-	-	-	-
Sensitivity Temperature Drift in %/°C	-	0,015	0,01	0,01	0,02
Nonlinearity typ in % of FS	0,2	0,3	0,2	1	1
Nonlinearity max in % of FS	1,25	-	-	-	-
Zero g level in V	1,65	1,5	1,5	-	1,65
Zero-g Level Change Vs Temperature in mg/°C	0,15	0,6	0,6	0,8	x=0,6, y=5,8, z=0,8
Rate Noise Density in µg/sqrtHz	-	-	-	-	-
Rate Noise Density in µg/sqrtHz rms	170	280	250	-	350
Cross Axis in %	1,5	1	2	3	5

# 21.3 Schaltpläne

## 21.3.1 MCU

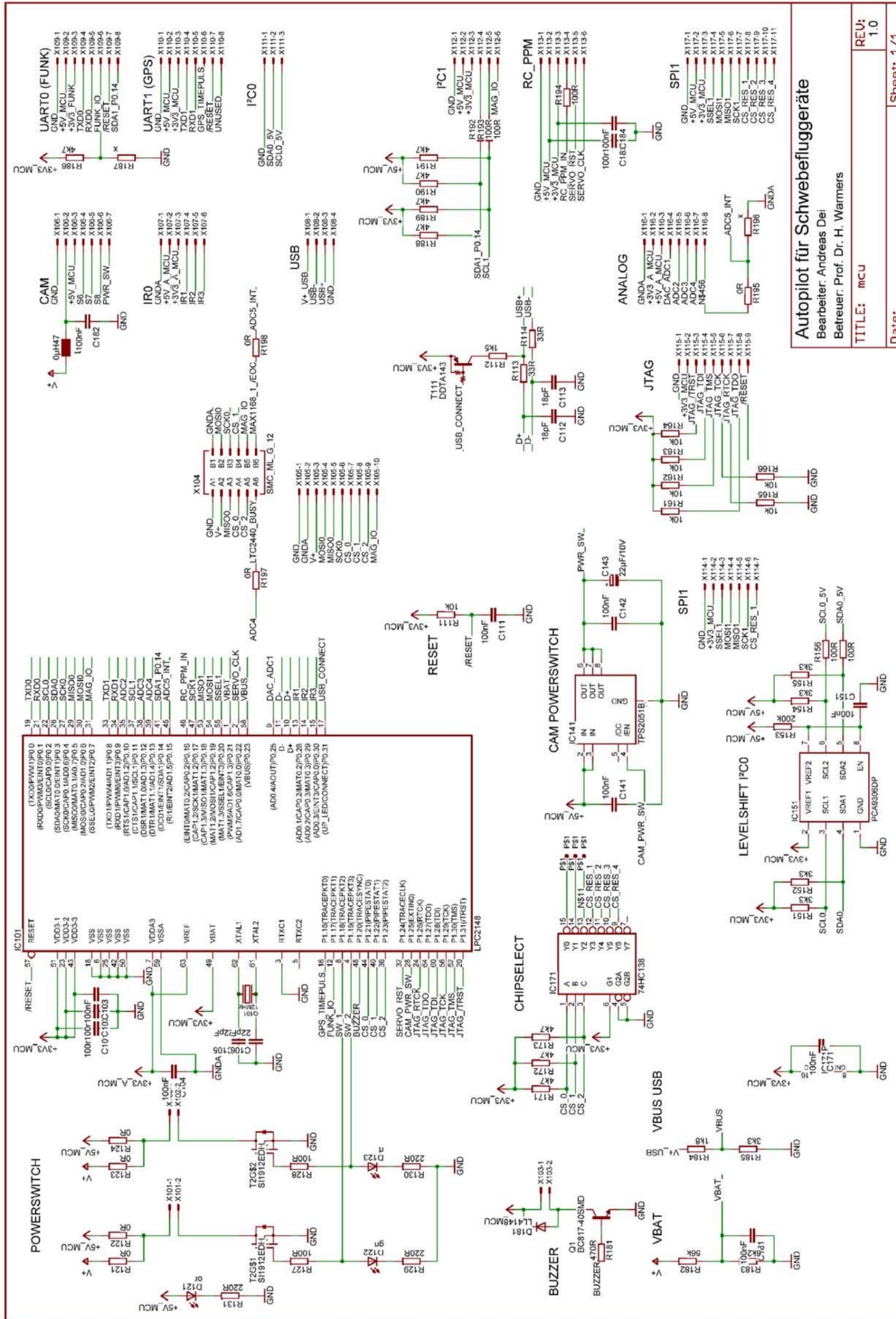


Bild 21.1: Schaltplan MCU Teil 1

**Autopilot für Schwebefluggeräte**  
 Bearbeiter: Andreas Dei  
 Betreuer: Prof. Dr. H. Warmers  
 TITLE: mcu  
 REU: 1.0  
 Date: Sheet: 1/1

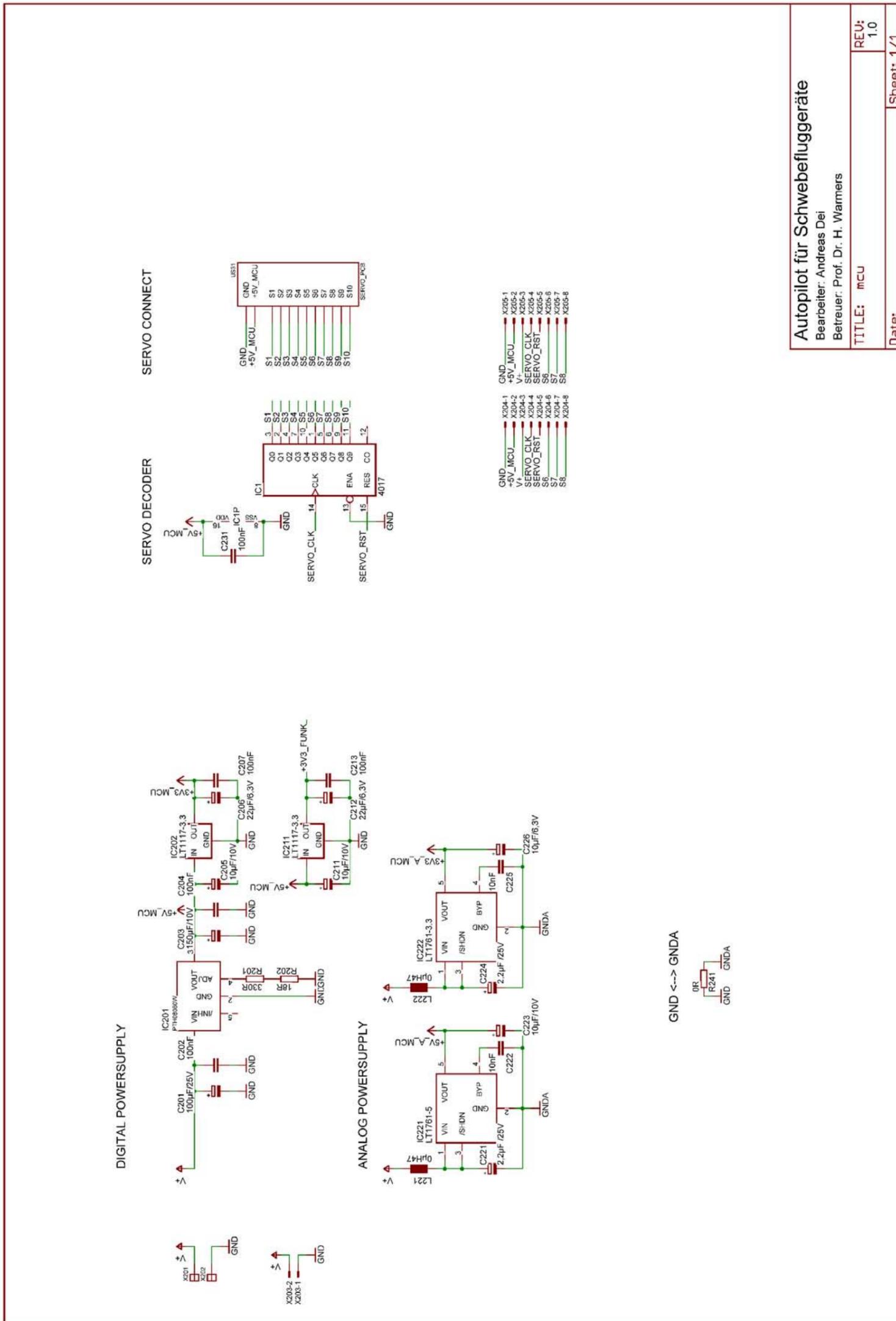


Bild 21.2: Schaltplan MCU Teil 2

Autopilot für Schwebefluggeräte

Bearbeiter: Andreas Dei

Betreuer: Prof. Dr. H. Wamers

TITLE: mcu

REU: 1.0

Date:

Sheet: 1/1

21.3.2 IMU

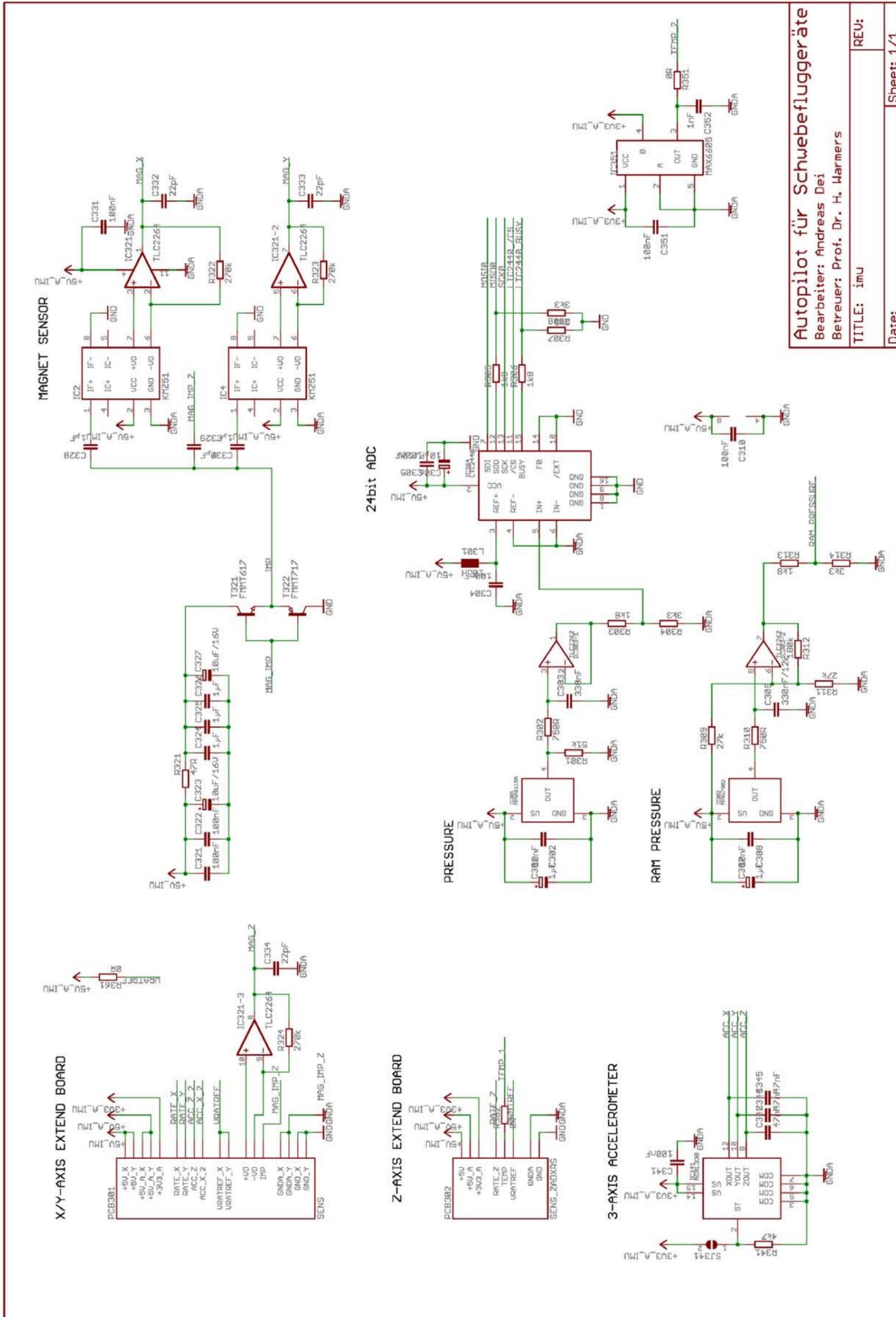
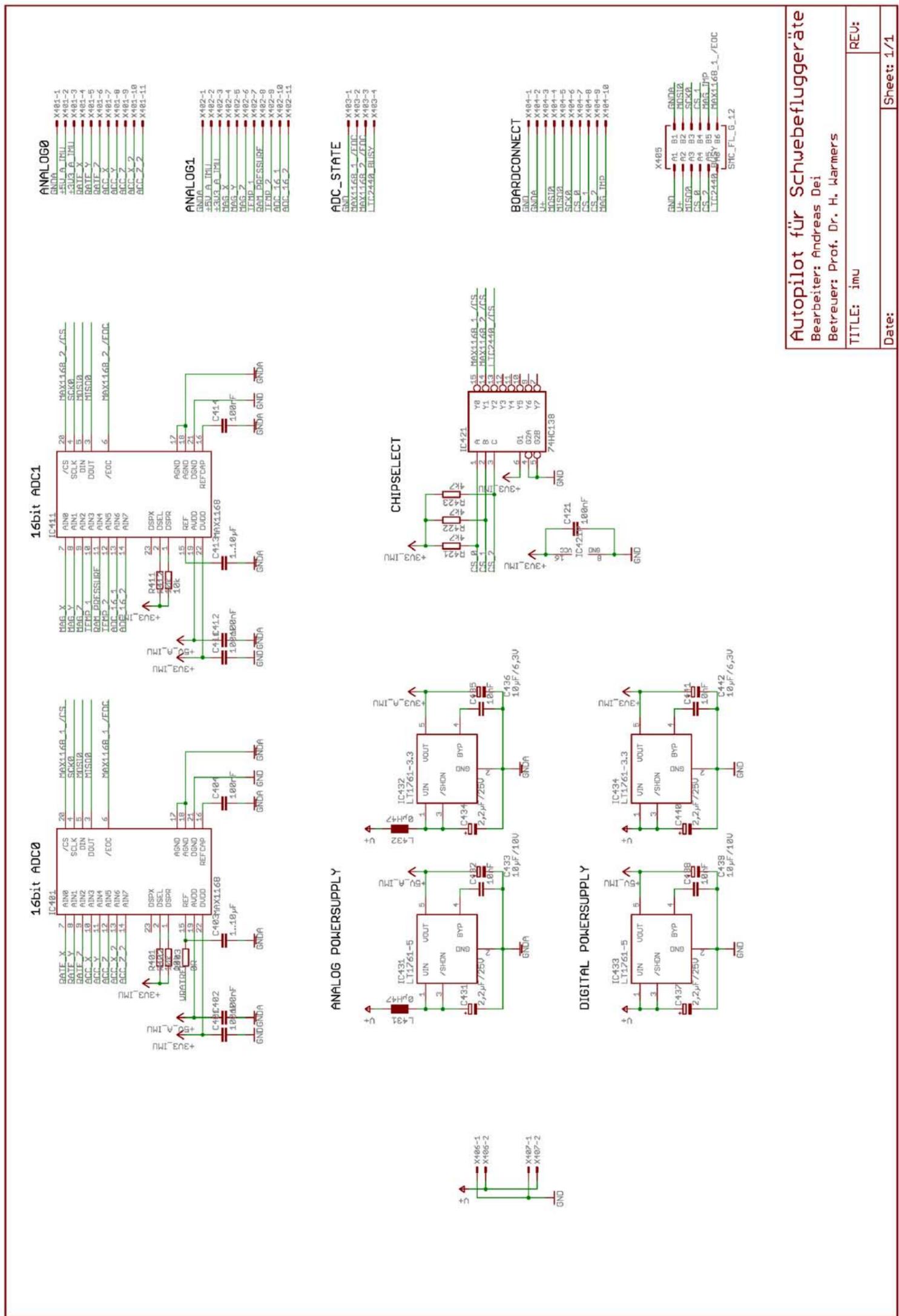


Bild 21.3: Schaltplan IMU Teil 1

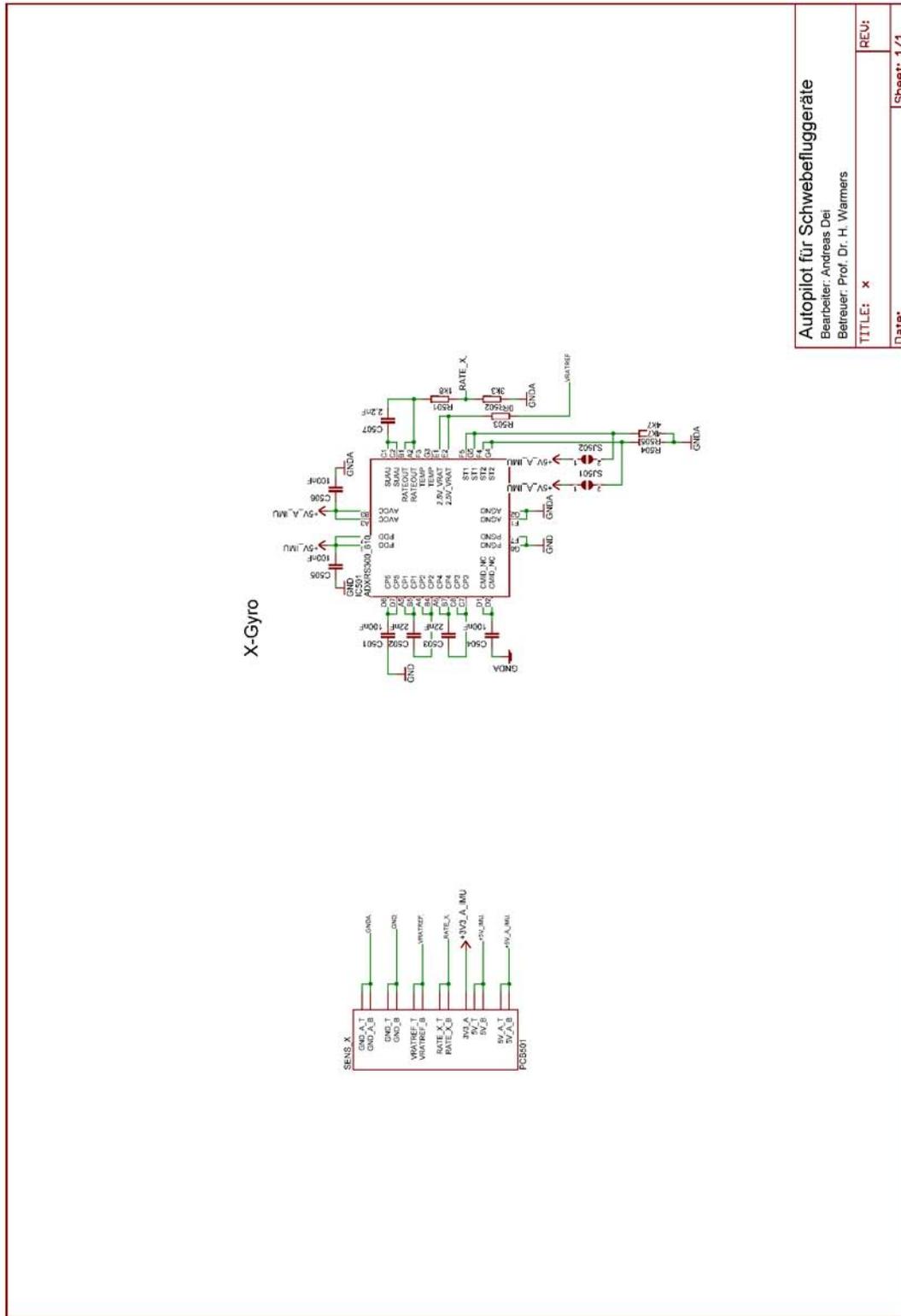
Autopilot für Schwebefluggeräte  
Bearbeiter: Andreas Dei  
Betreuer: Prof. Dr. H. Wärmers  
TITLE: imu  
REV:  
Date:  
Sheet: 1/1



**Autopilot für Schwebefluggeräte**  
 Bearbeiter: Andreas Dei  
 Betreuer: Prof. Dr. H. Warmers  
 TITLE: imu  
 REV:  
 Date: Sheet: 1/1

Bild 21.4: Schaltplan IMU Teil 2

### 21.3.3 Sensorplatine ADXRS610



**Autopilot für Schwebefluggeräte**

Bearbeiter: Andreas Dei

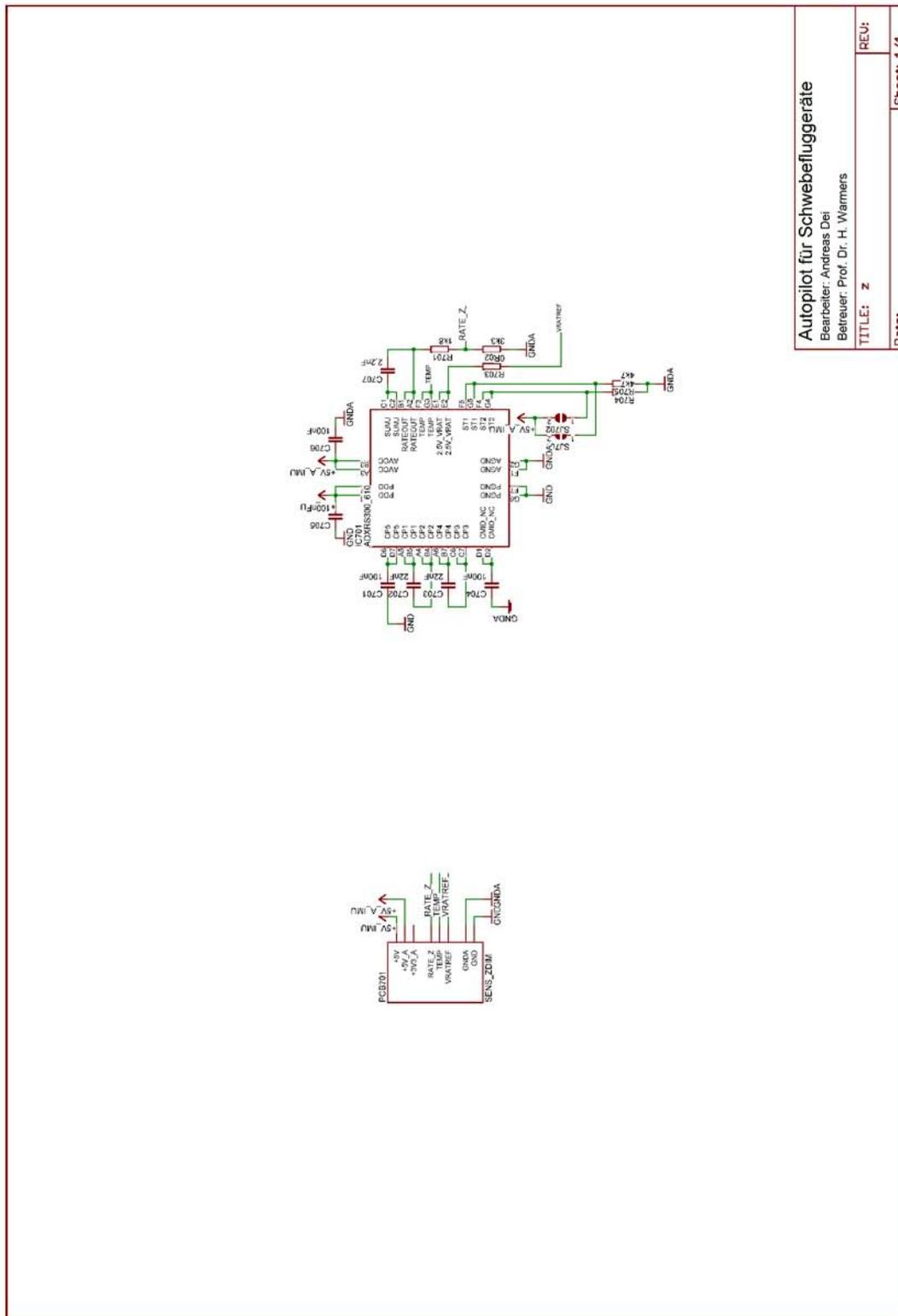
Betreuer: Prof. Dr. H. Warmers

TITLE: **x** REV:

Date: Sheet: 1/1

Bild 21.5: ADXRS610 X-Sensorplatine





**Autopilot für Schweflugaräte**

Bearbeiter: Andreas Del

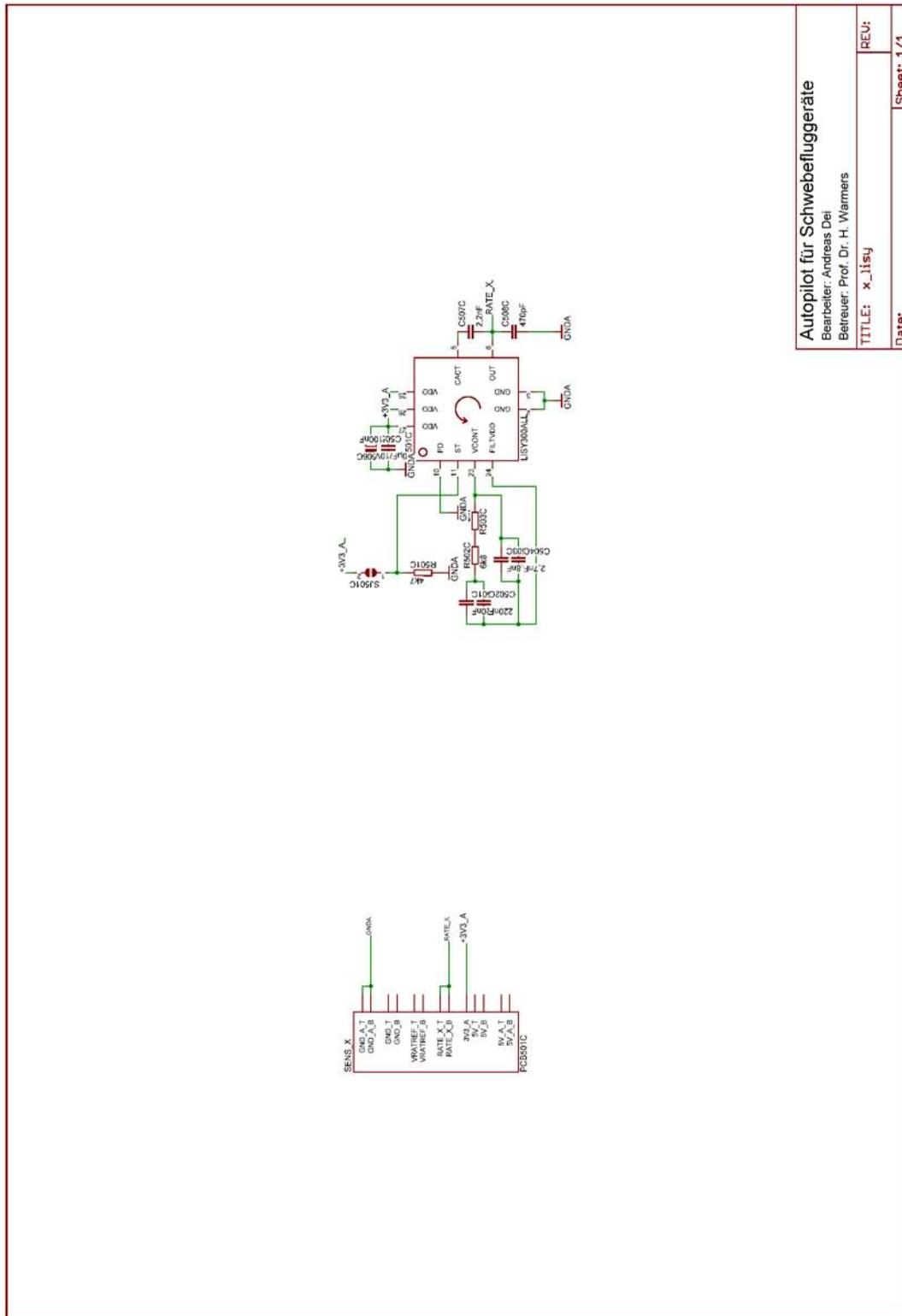
Betreuer: Prof. Dr. H. Wärmers

TITLE: **z**      REV:

Date:      Sheet: 1/1

Bild 21.7: ADXRS610 Z-Sensorplatine

### 21.3.4 Sensorplatine LISY300AI



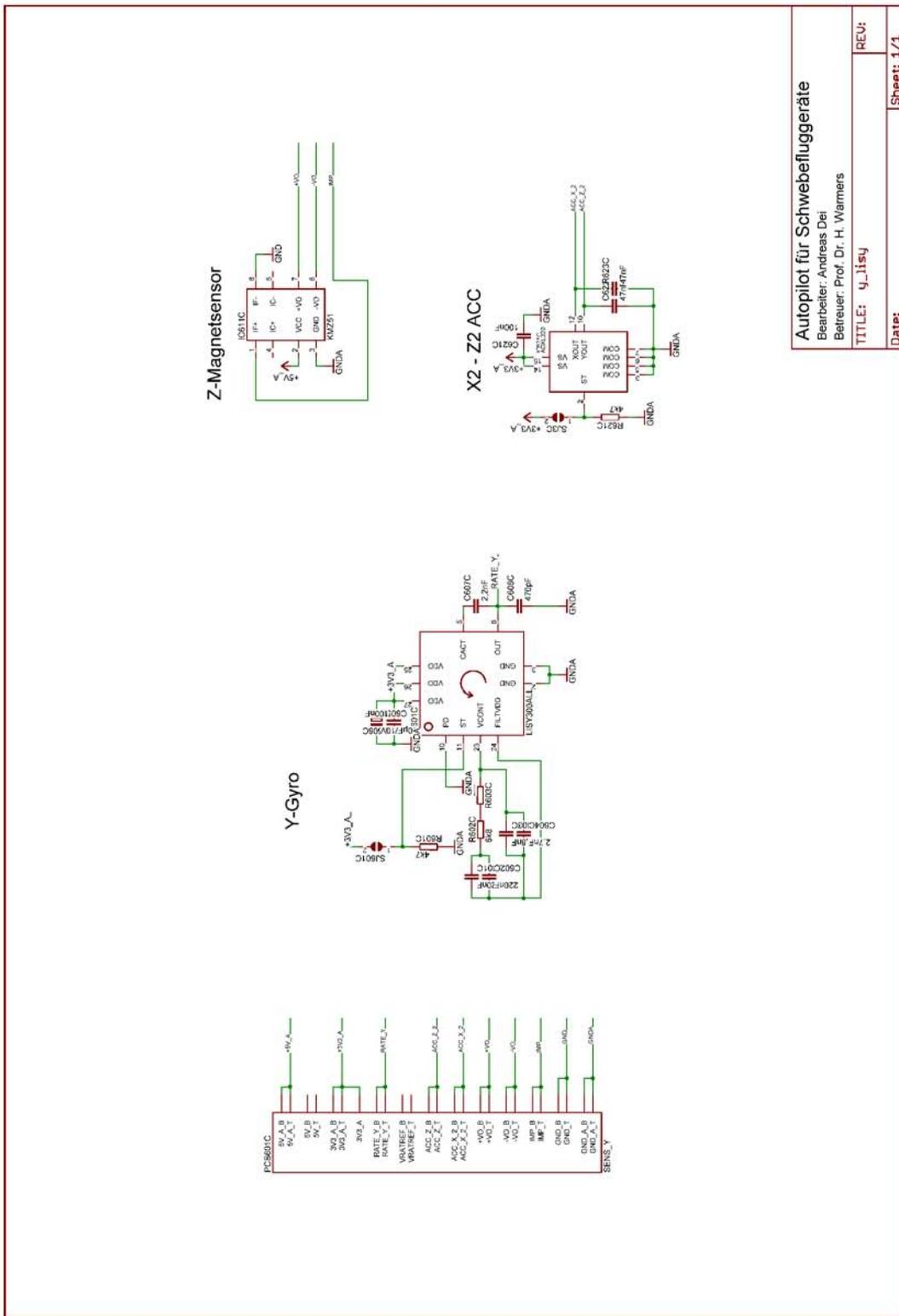
**Autopilot für Schwebefluggeräte**

Bearbeiter: Andreas Dei  
 Betreuer: Prof. Dr. H. Warmers

TITLE: x\_lisy

REU:  
 Date: Sheet: 1/1

Bild 21.8: LISY300AL X-Sensorplatine

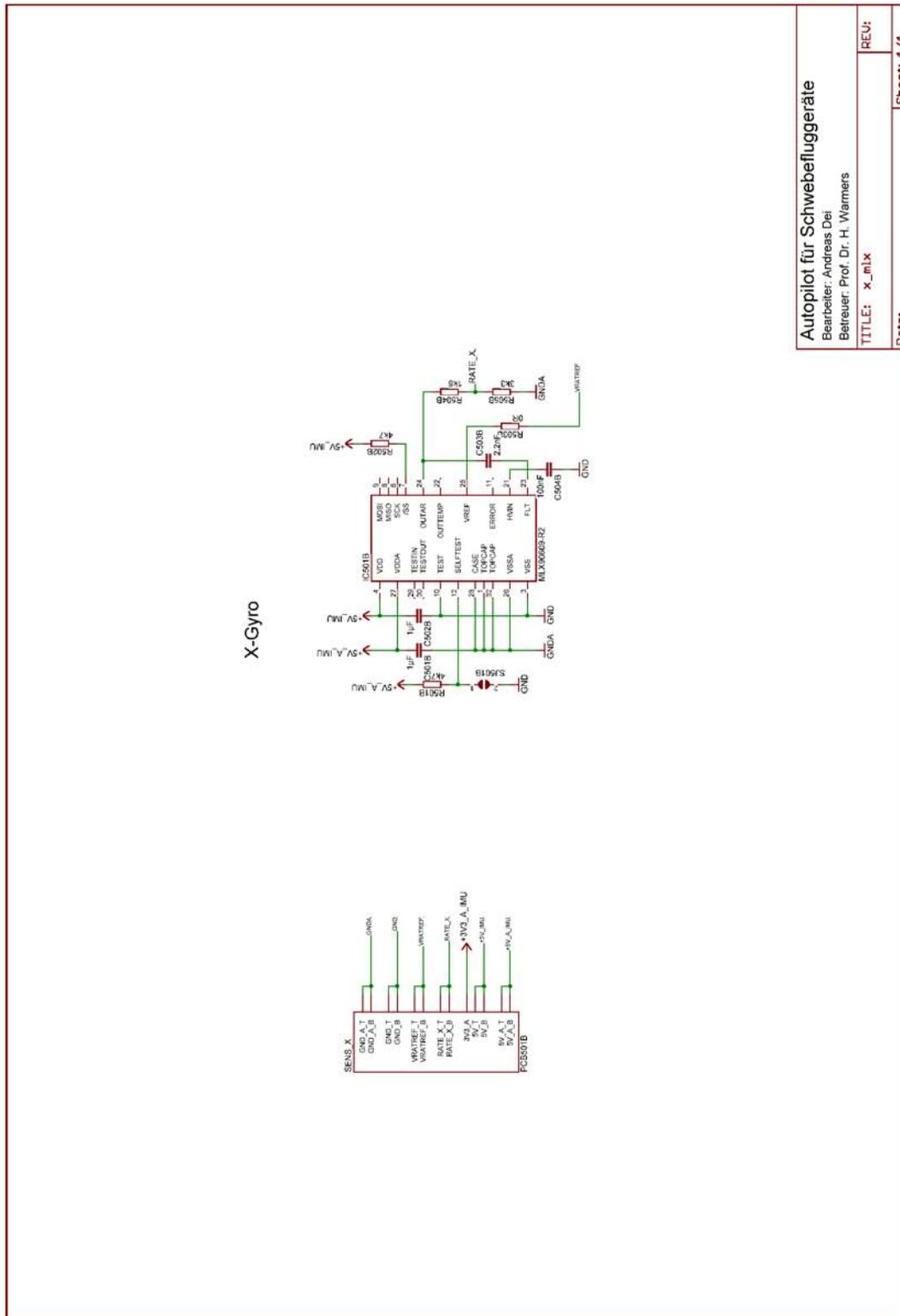


**Autopilot für Schweflugaräte**  
 Bearbeiter: Andreas Dei  
 Betreuer: Prof. Dr. H. Wärmers  
 TITLE: u\_lisy  
 Date: Sheet: 1/1

Bild 21.9: LISY300AL Y-Sensorplatte



### 21.3.5 Sensorplatine MLX90609-R2



**Autopilot für Schwebefluggeräte**  
 Bearbeiter: Andreas Dei  
 Betreuer: Prof. Dr. H. Warmers  
 TITLE: x\_mlx  
 REVU:  
 Date: Sheet: 1/1

Bild 21.11: MLX90609-R2 X-Sensorplatine

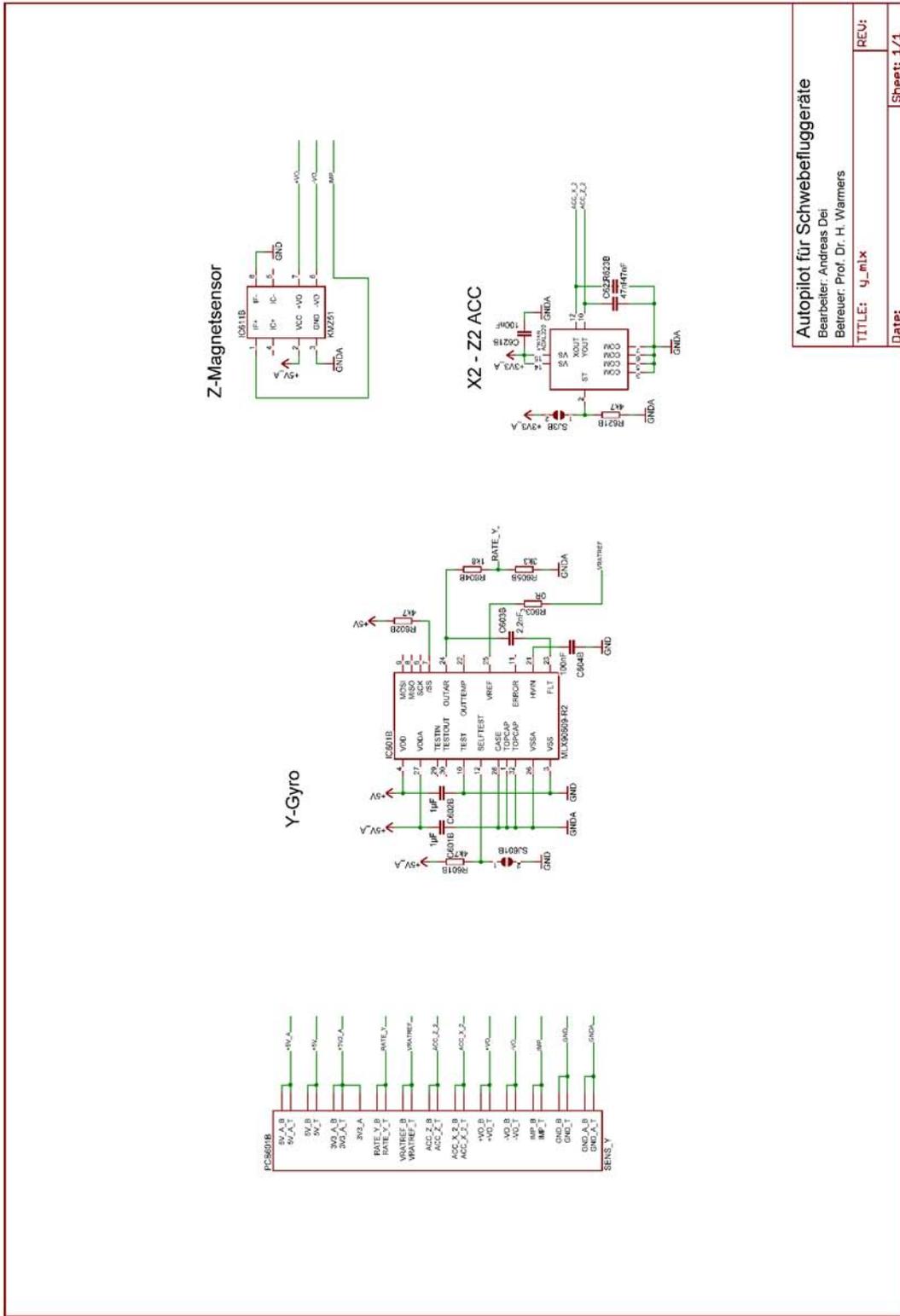
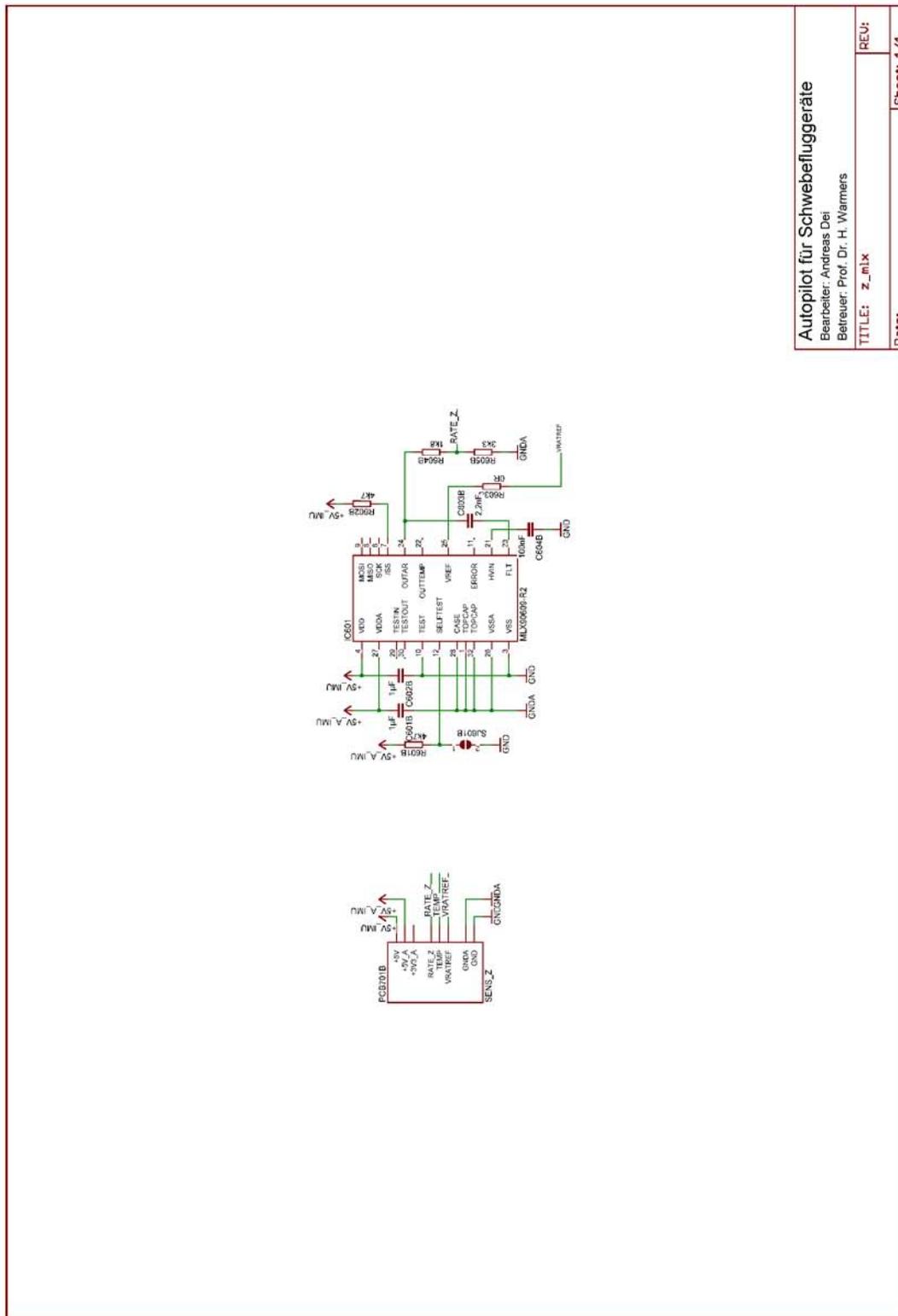


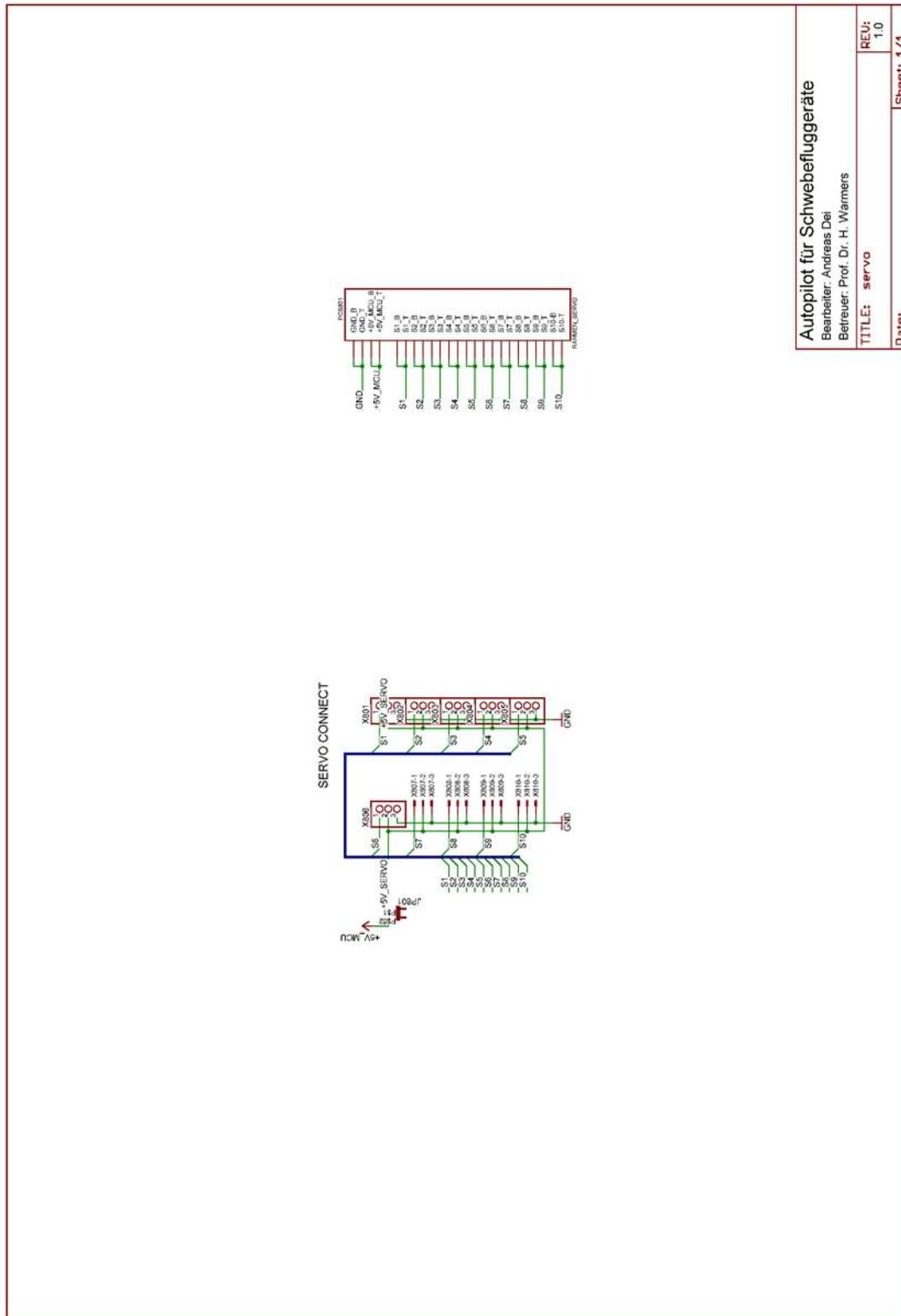
Bild 21.12: MLX90609-R2 Y-Sensorplatte



<b>Autopilot für Schweflugaräte</b>	
Bearbeiter: Andreas Dei	
Betreuer: Prof. Dr. H. Wärmers	
TITLE: z_mlx	REV:
Date:	Sheet: 1/1

Bild 21.13: MLX90609-R2 Z-Sensorplatine

### 21.3.6 Servoadapter



**Autopilot für Schwebefluggeräte**

Bearbeiter: Andreas Dei

Betreiber: Prof. Dr. H. Warmers

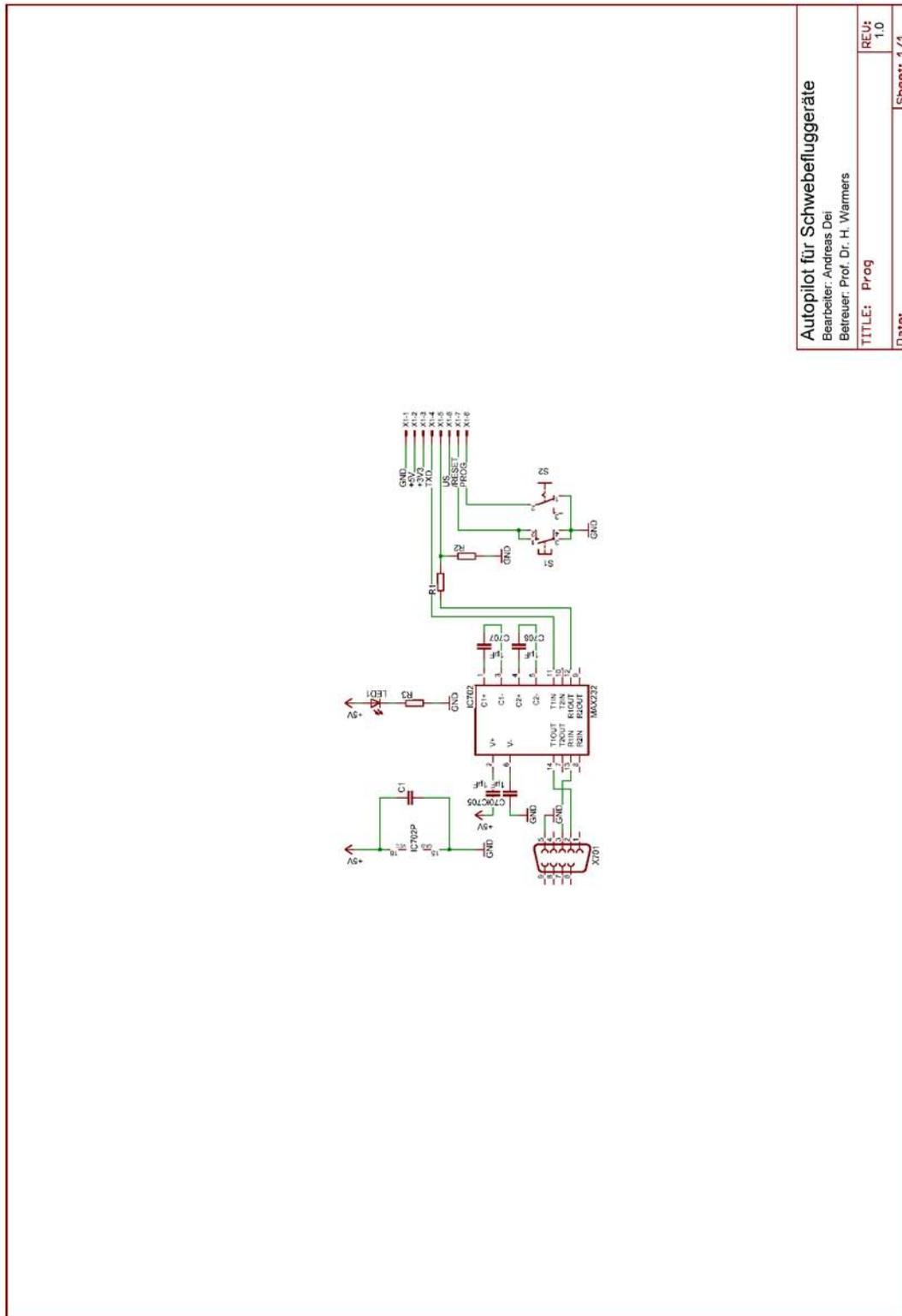
TITLE: servo

REV: 1.0

Date: Sheet: 1/1

Bild 21.14: Servoadapter

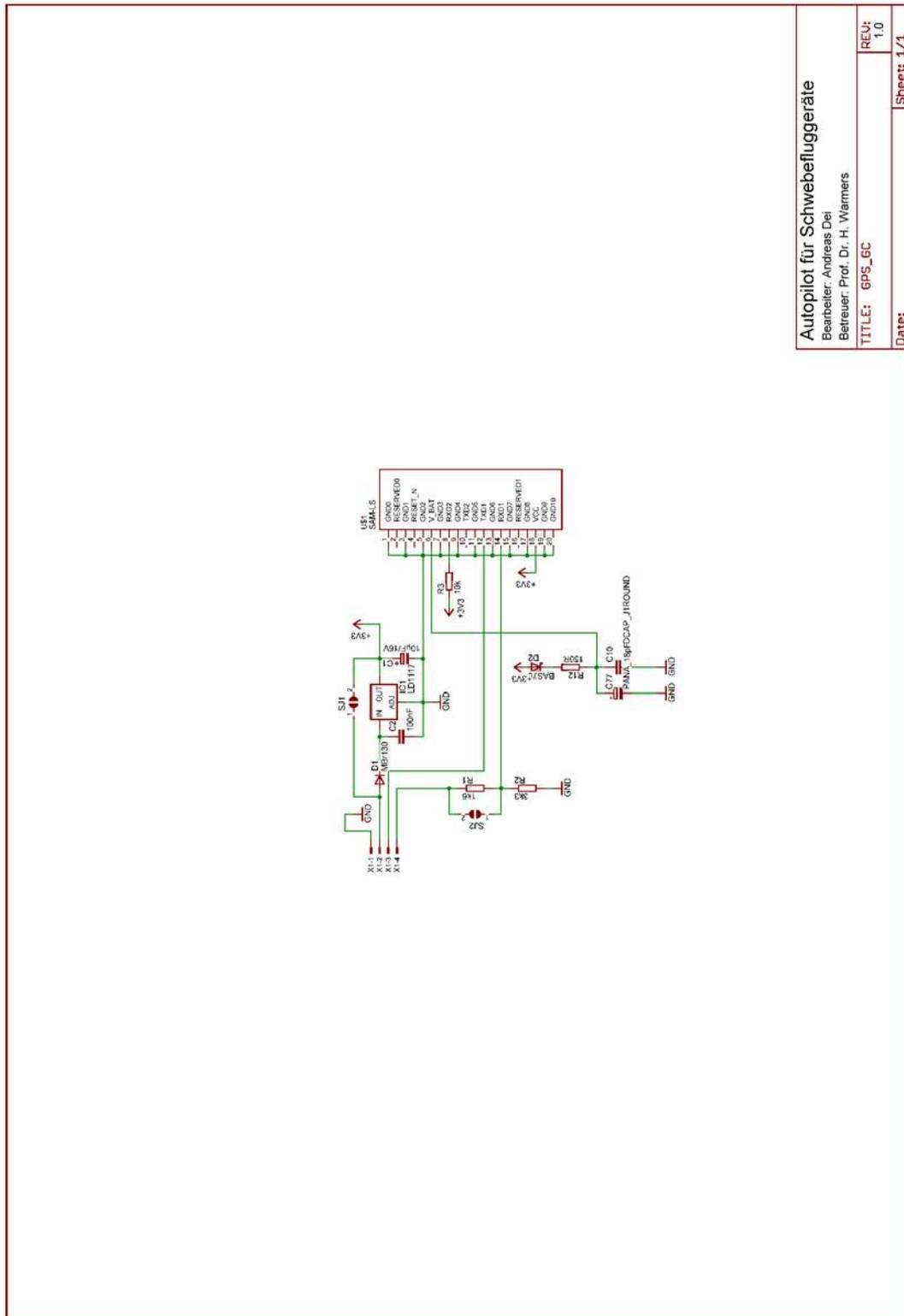
### 21.3.7 Programmieradapter



**Autopilot für Schwebefluggeräte**  
 Bearbeiter: Andreas Dei  
 Betreuer: Prof. Dr. H. Warmers  
 TITLE: Prog  
 REV: 1.0  
 Date: Sheet: 1/1

Bild 21.15: Programmieradapter

21.3.8 GPS



Autopilot für Schwefluggeräte

Bearbeiter: Andreas Dei  
 Betreuer: Prof. Dr. H. Warmers

TITLE: GPS\_GC  
 REV: 1.0

Date: Sheet: 1/1

Bild 21.16: GPS Modul

## 21.4 Layouts

### 21.4.1 MCU

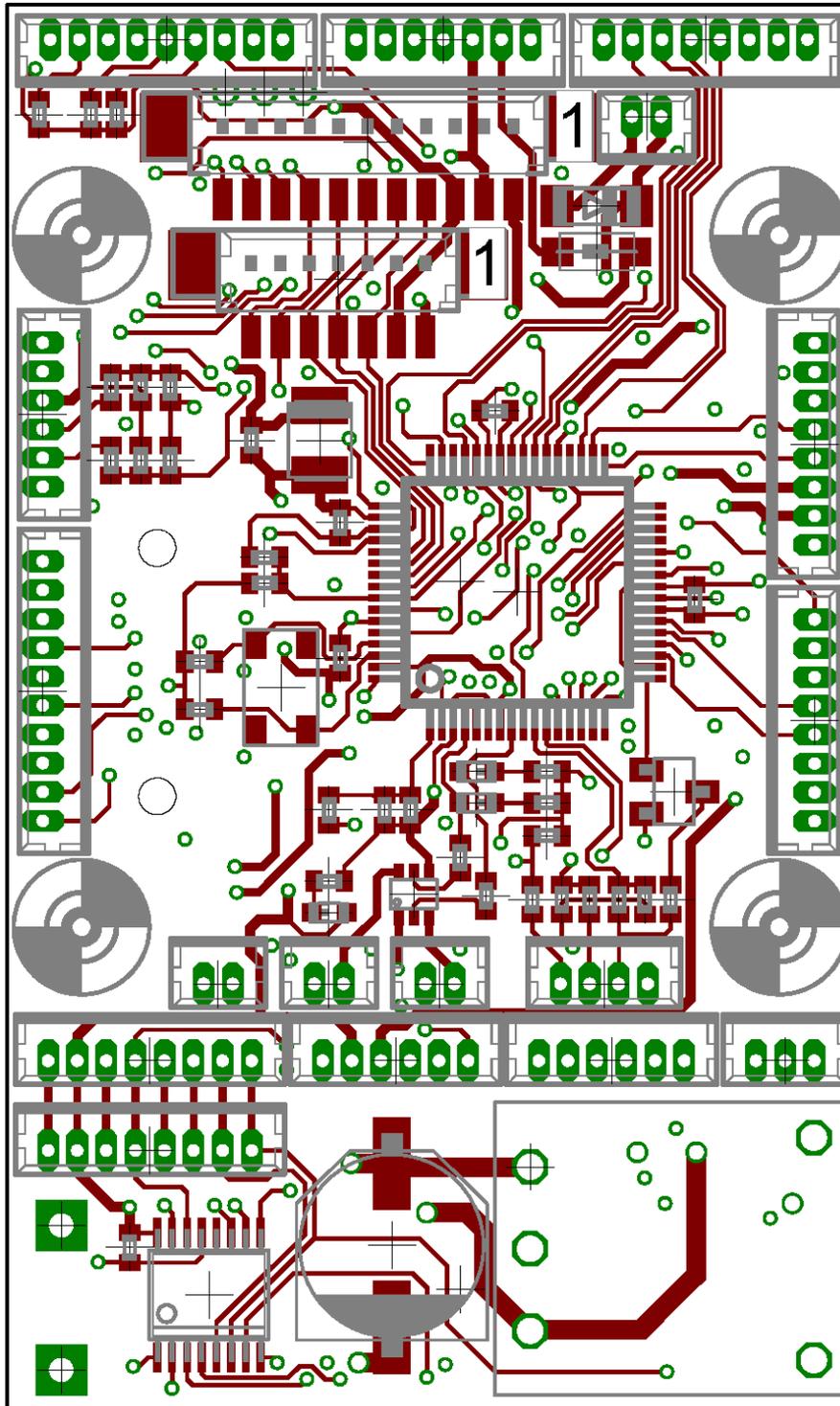


Bild 21.17: MCU Platine, Layer 1

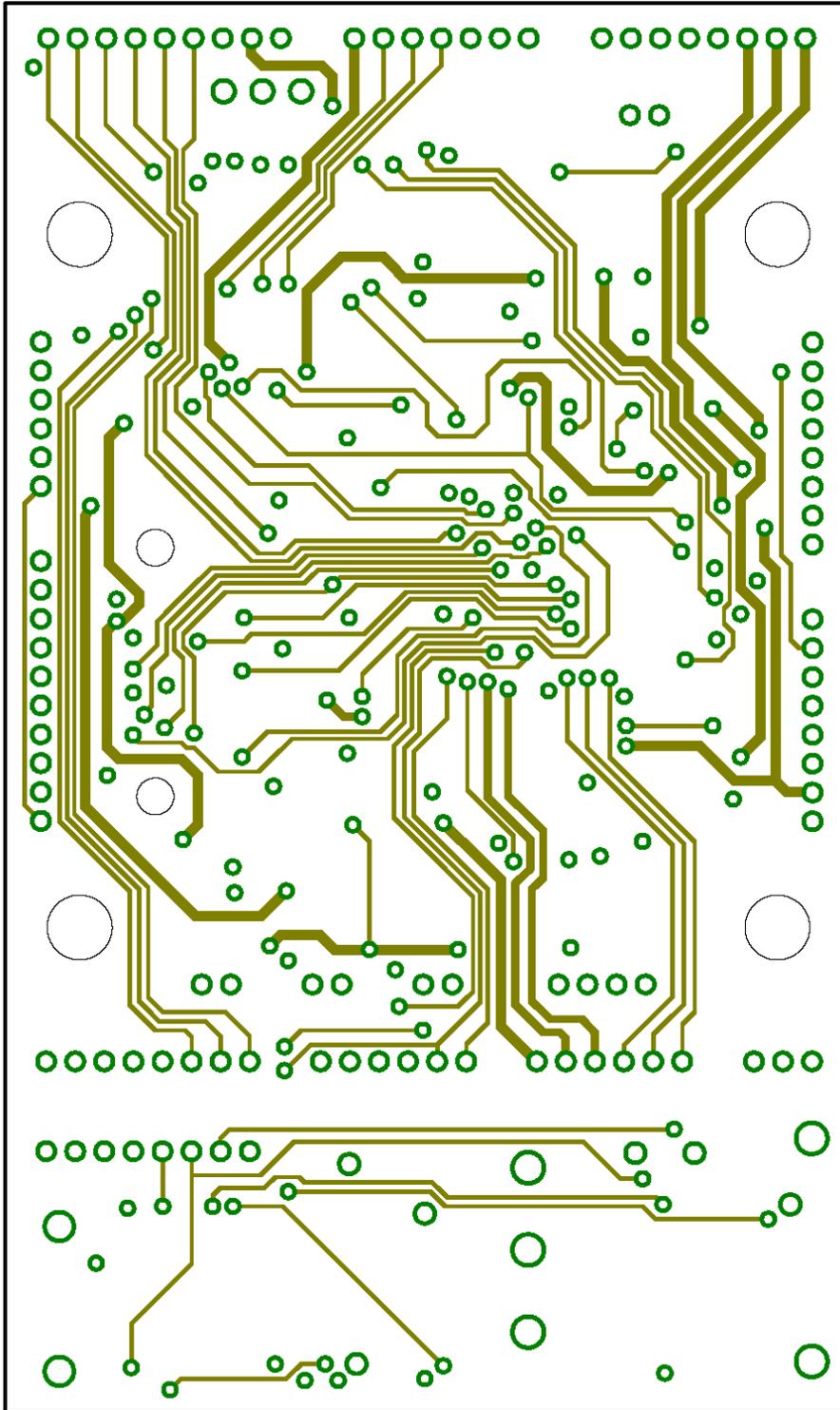


Bild 21.18: MCU Platine, Layer 2

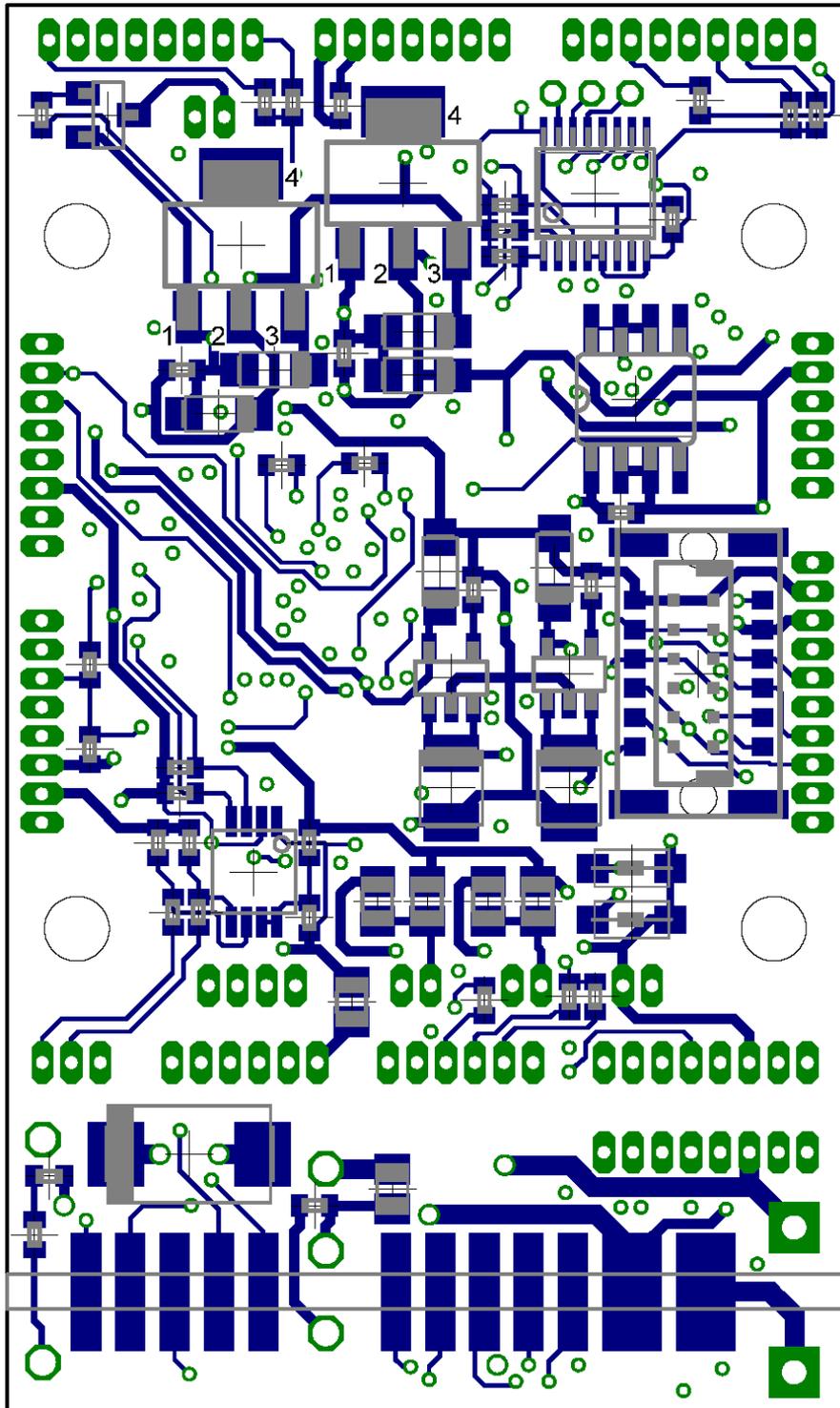


Bild 21.19: MCU Platine, Layer 16

## 21.4.2 IMU

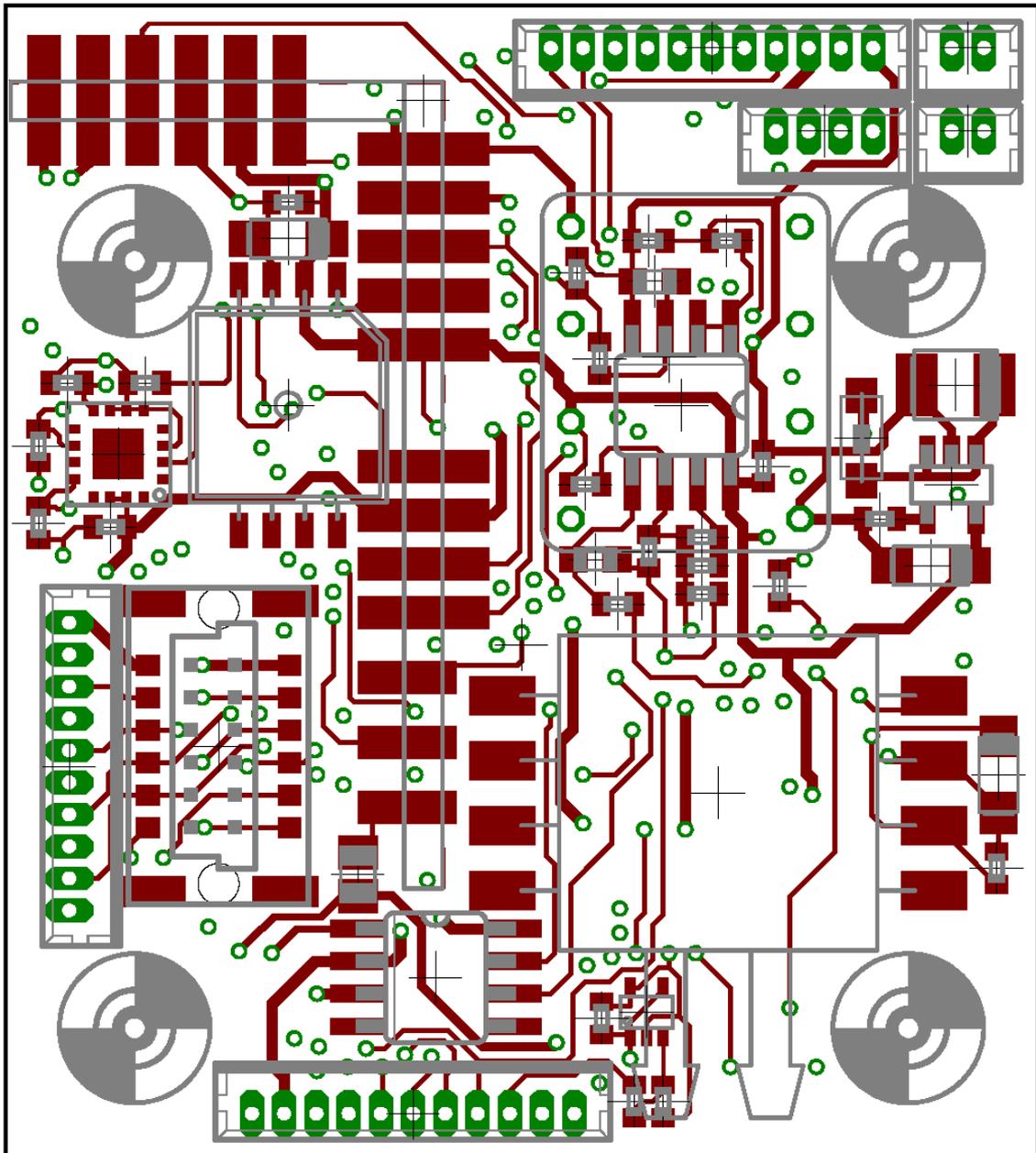


Bild 21.20: IMU Platine, Layer 1

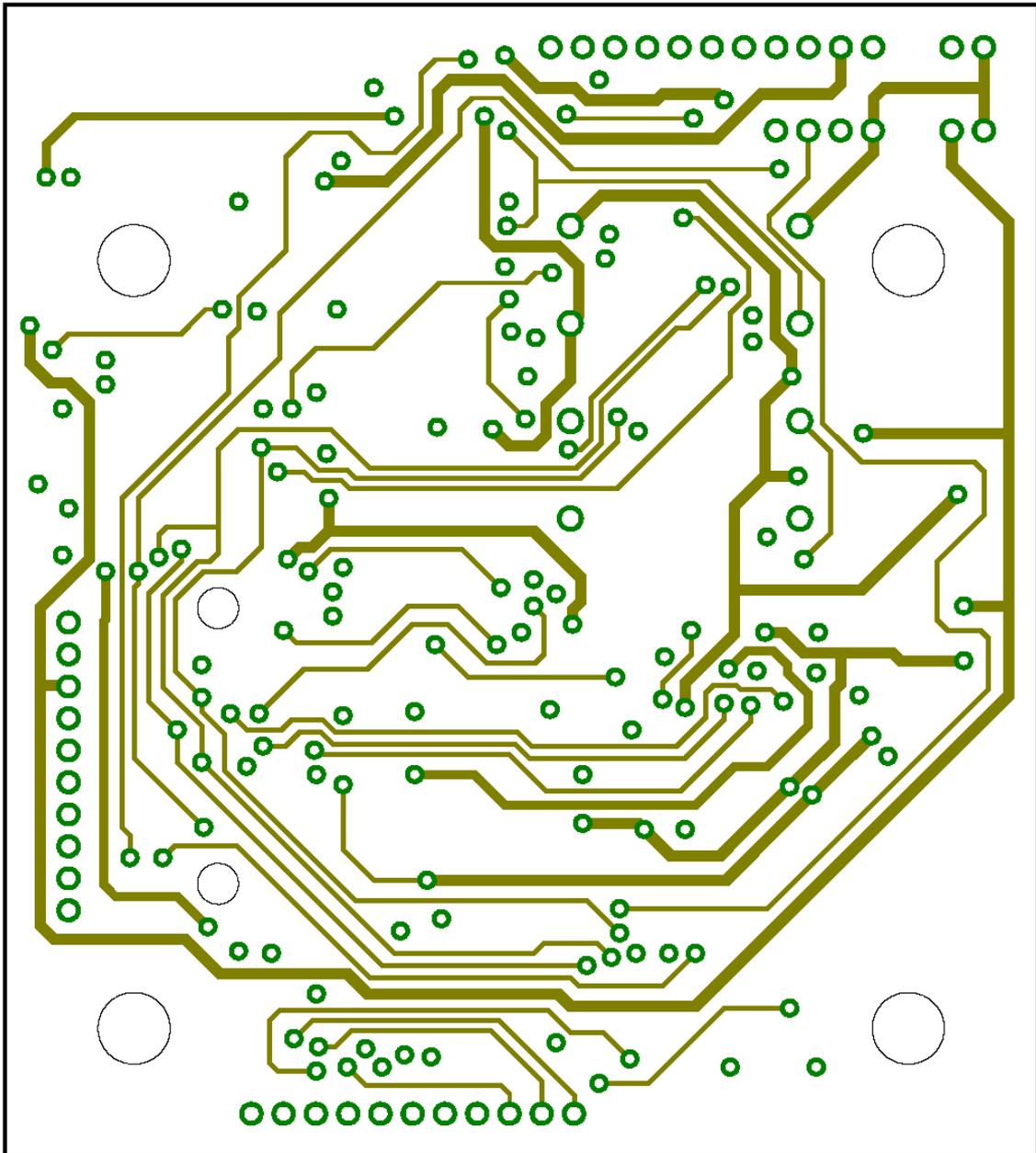


Bild 21.21: IMU Platine, Layer 2

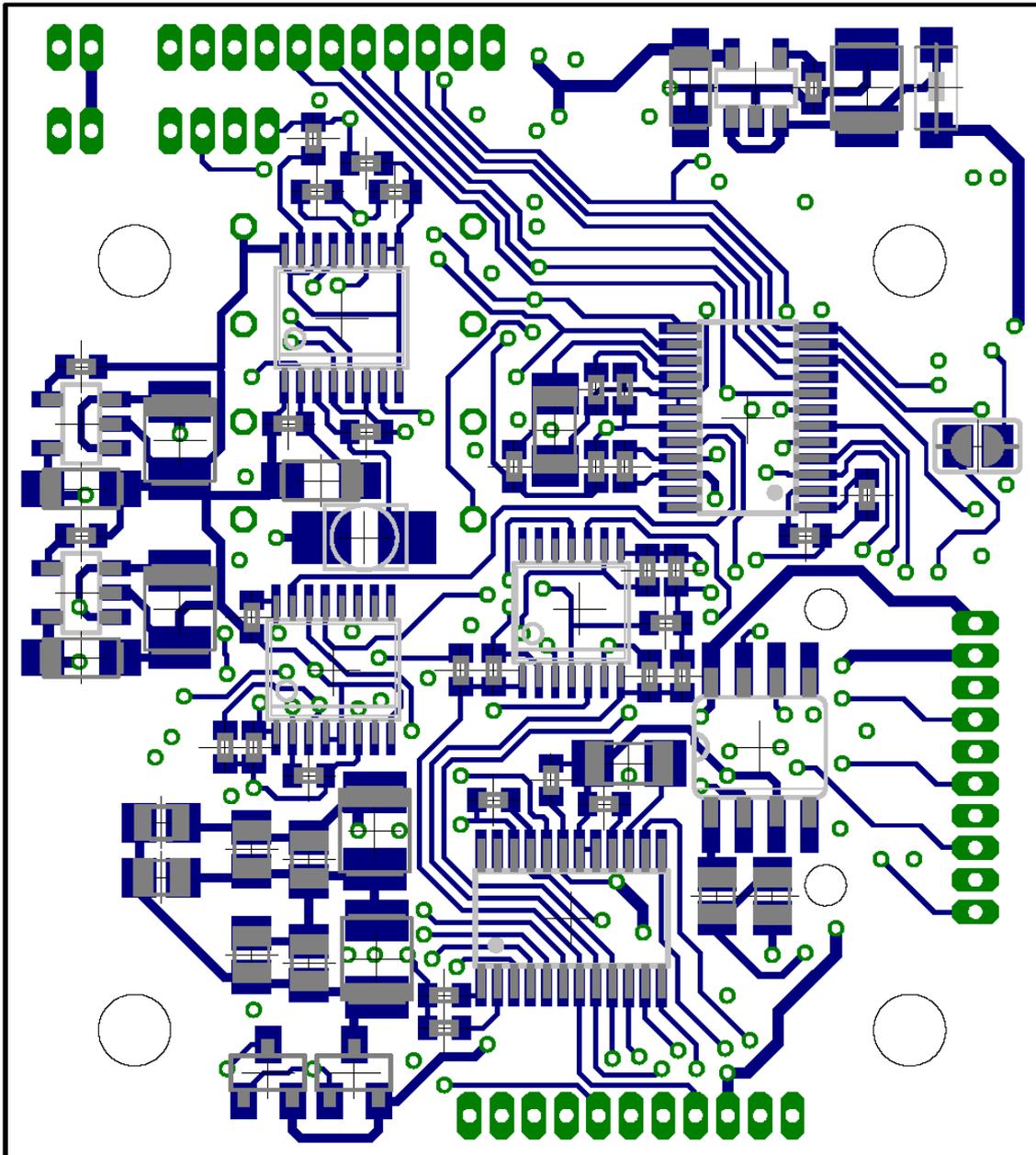


Bild 21.22: IMU Platine, Layer 16

### 21.4.3 Sensorplatine ADXRS610

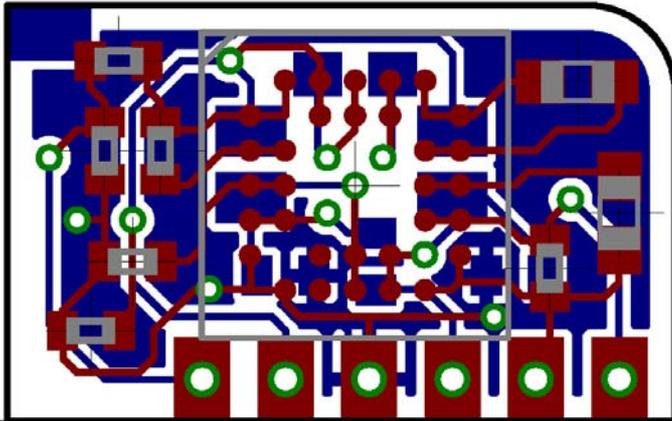


Bild 21.23: ADXRS610 X-Sensorplatine

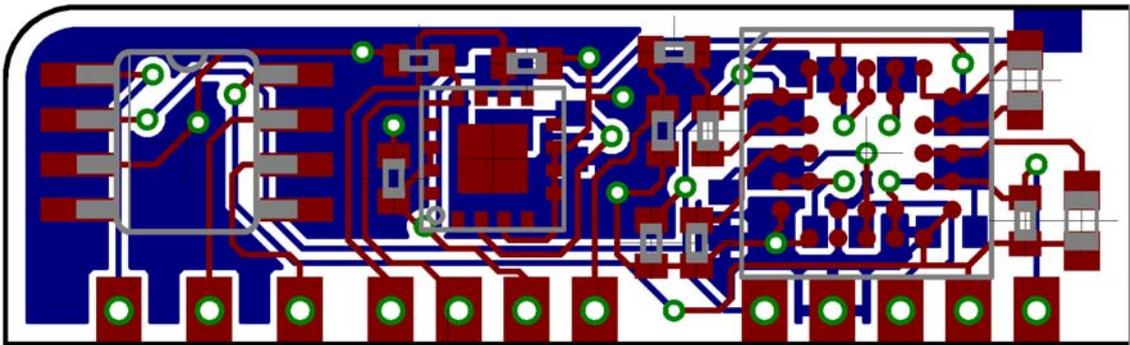


Bild 21.24: ADXRS610 Y-Sensorplatine

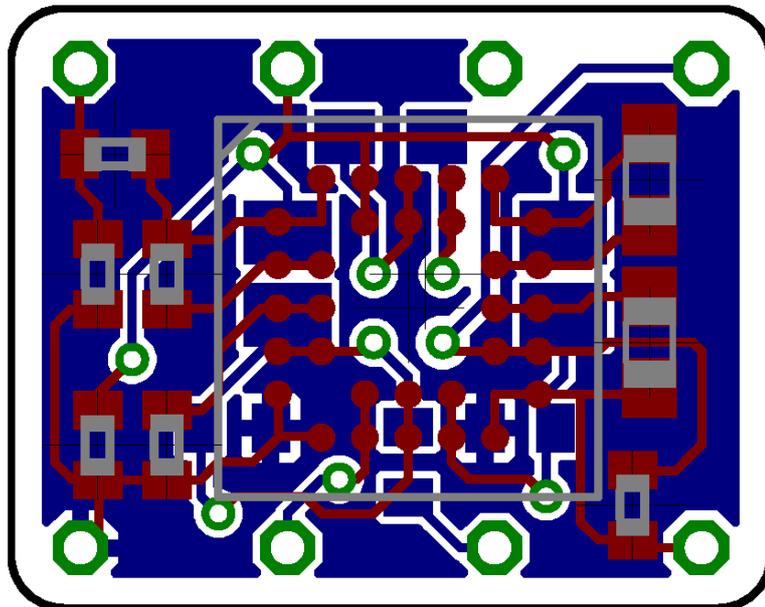


Bild 21.25: ADXRS610 Z-Sensorplatine

#### 21.4.4 Sensorplatine LISY300AL

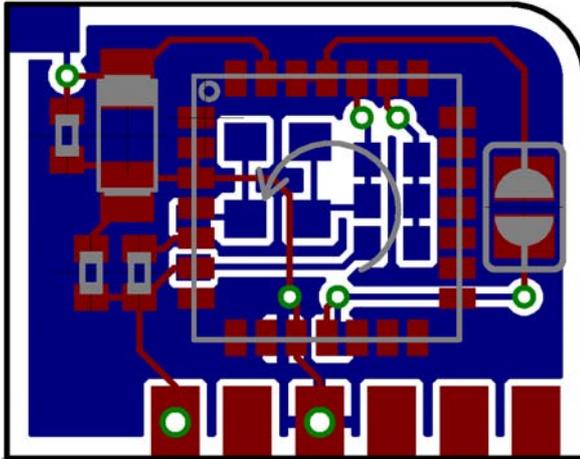


Bild 21.26: LISY300AL X-Sensorplatine

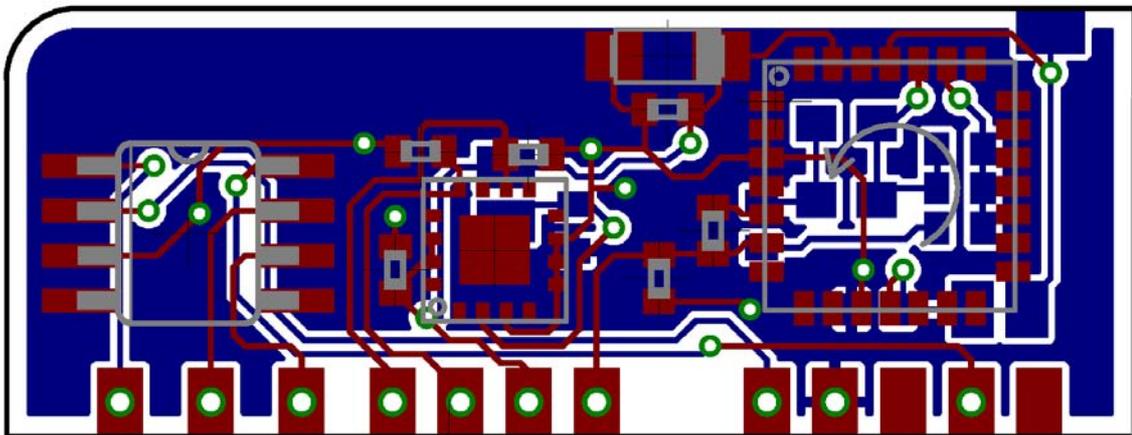


Bild 21.27: LISY300AL Y-Sensorplatine

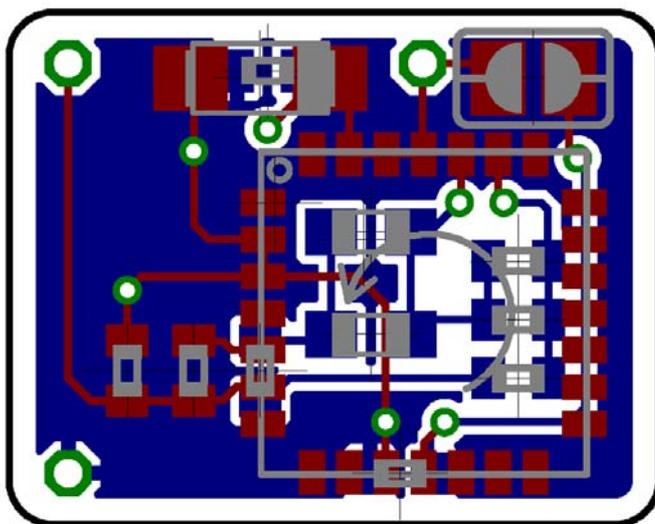


Bild 21.28: LISY300AL Z-Sensorplatine

### 21.4.5 Sensorplatine MLX90609-R2

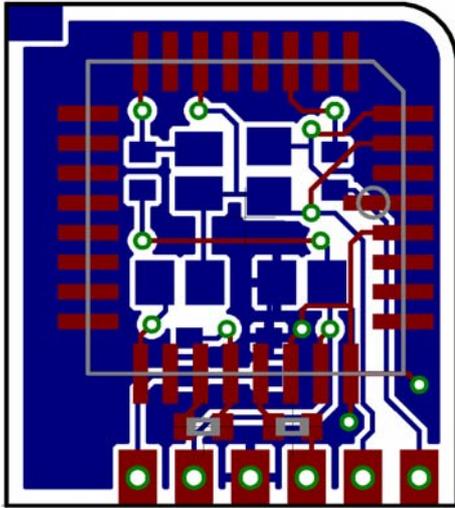


Bild 21.29: MLX90609-R2 X-Sensorplatine

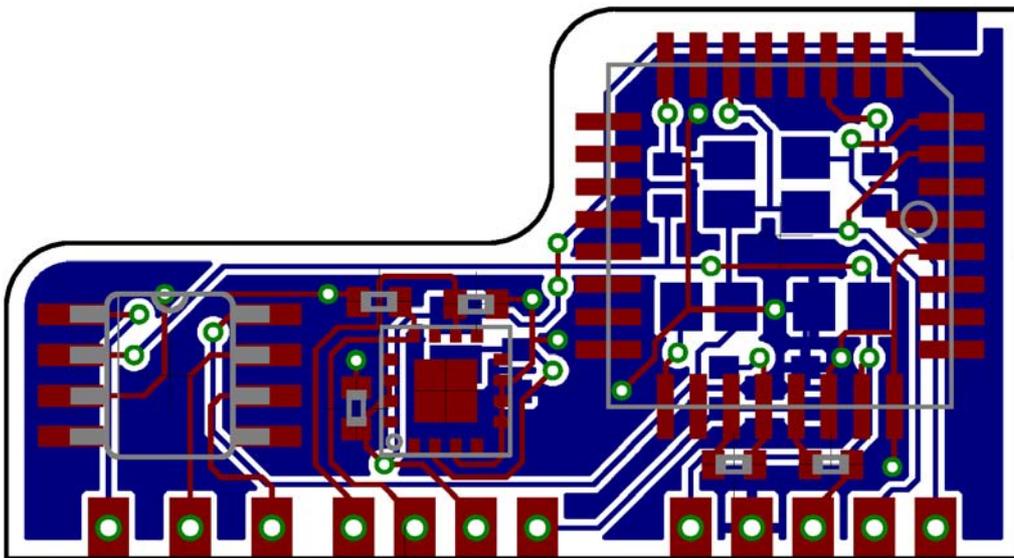


Bild 21.30: LISY300AL Y-Sensorplatine

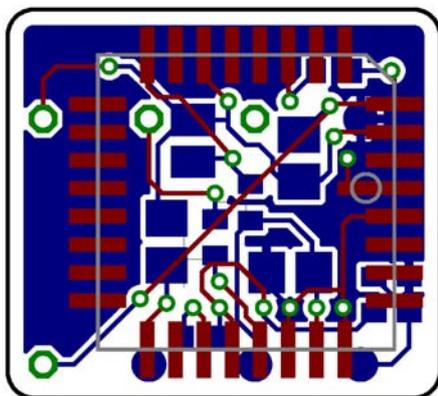


Bild 21.31: LISY300AL Z-Sensorplatine

### 21.4.6 Servoadapter

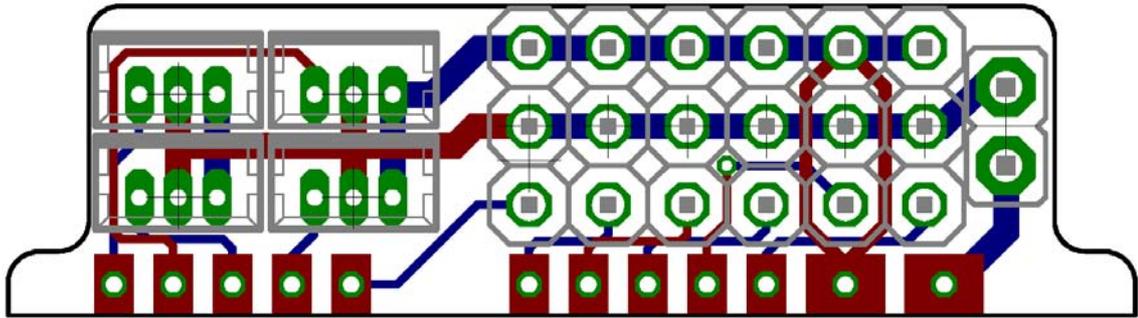


Bild 21.32: Servoadapter

### 21.4.7 Programmieradapter

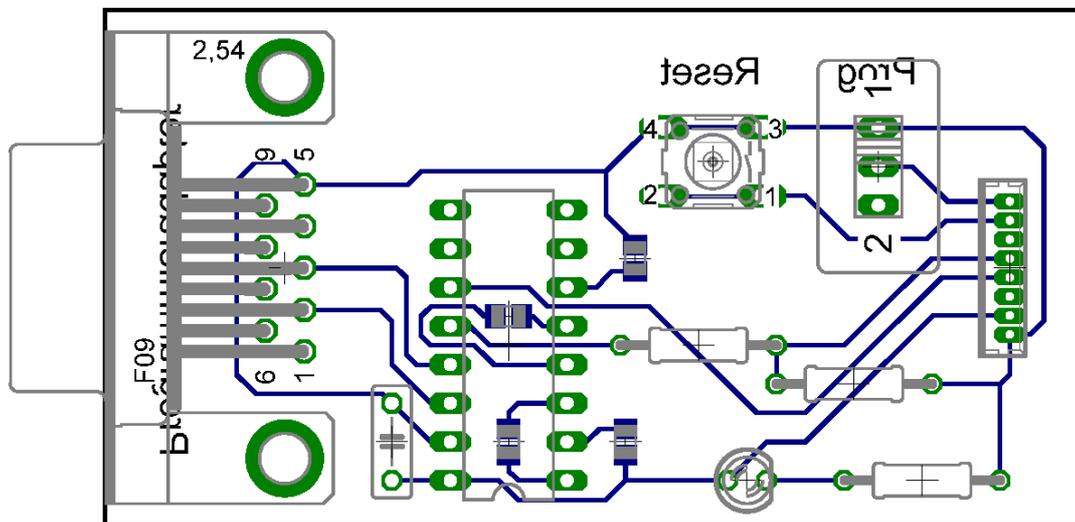


Bild 21.33: Programmieradapter

### 21.4.8 GPS

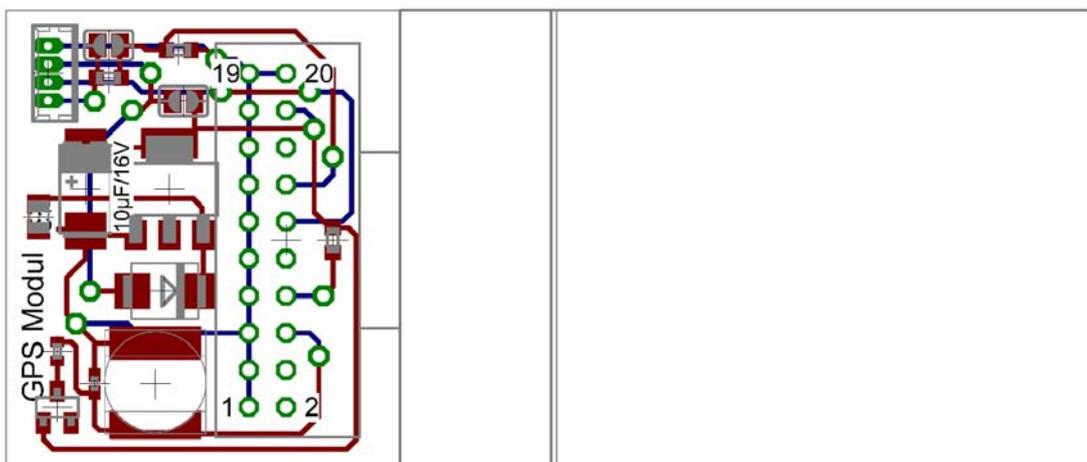


Bild 21.34: GPS-Modul

## 21.5 Programmablaufpläne

### 21.5.1 PAP für „spi\_int“

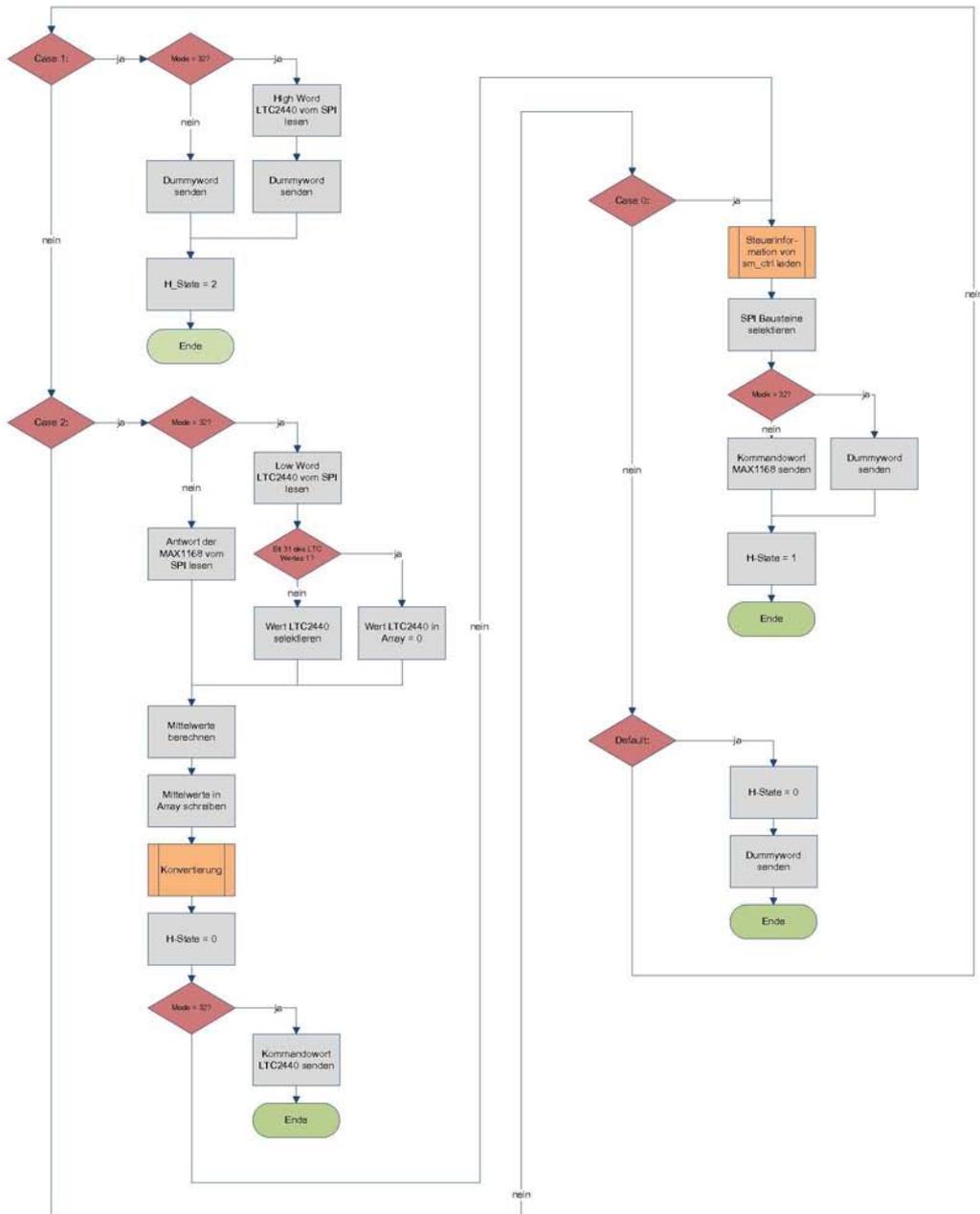


Bild 21.35: Programablaufplan "spi\_int"

## 21.5.2 PAP für „sm\_ctrl“

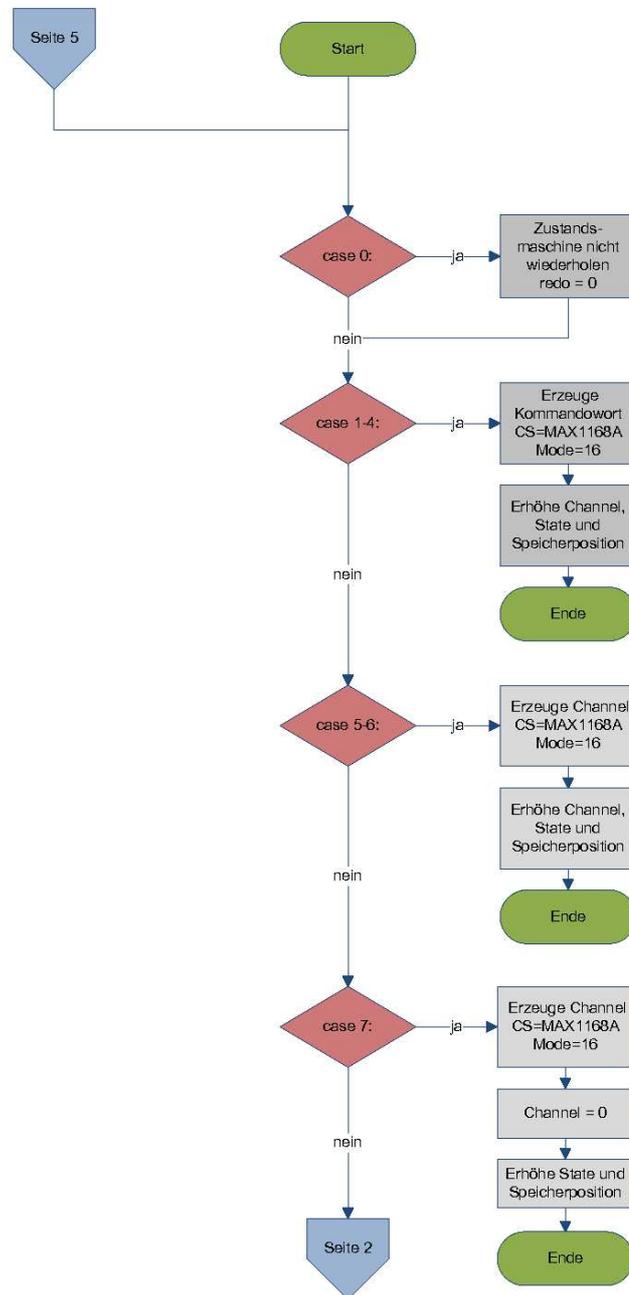


Bild 21.36: PAP "sm\_ctrl" Seite 1

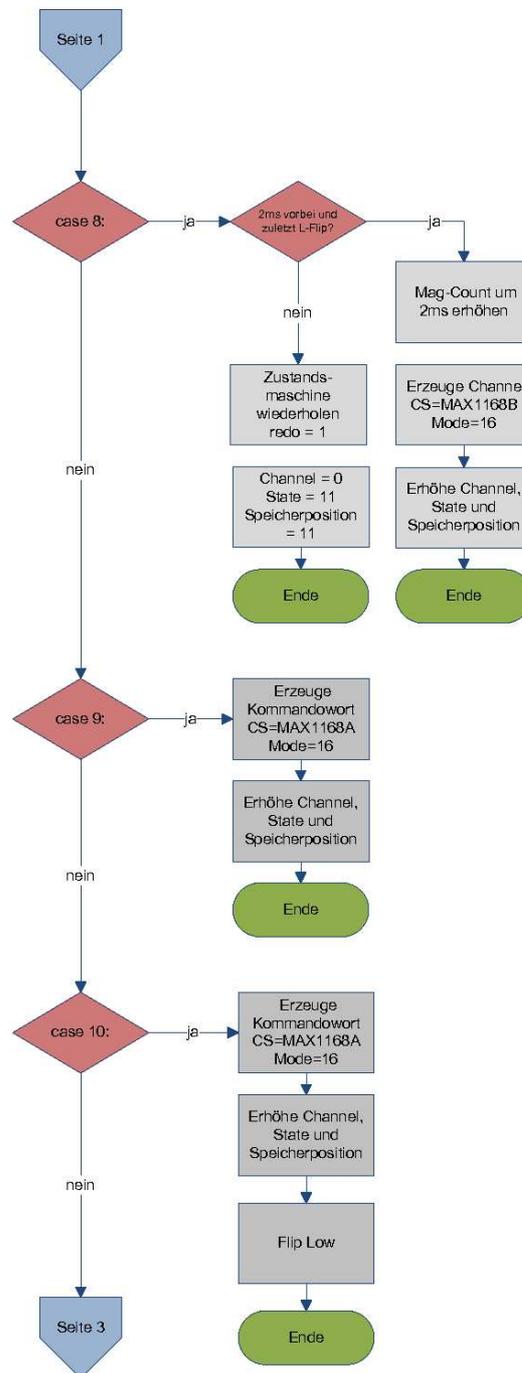


Bild 21.37: PAP "sm\_ctrl" Seite 2

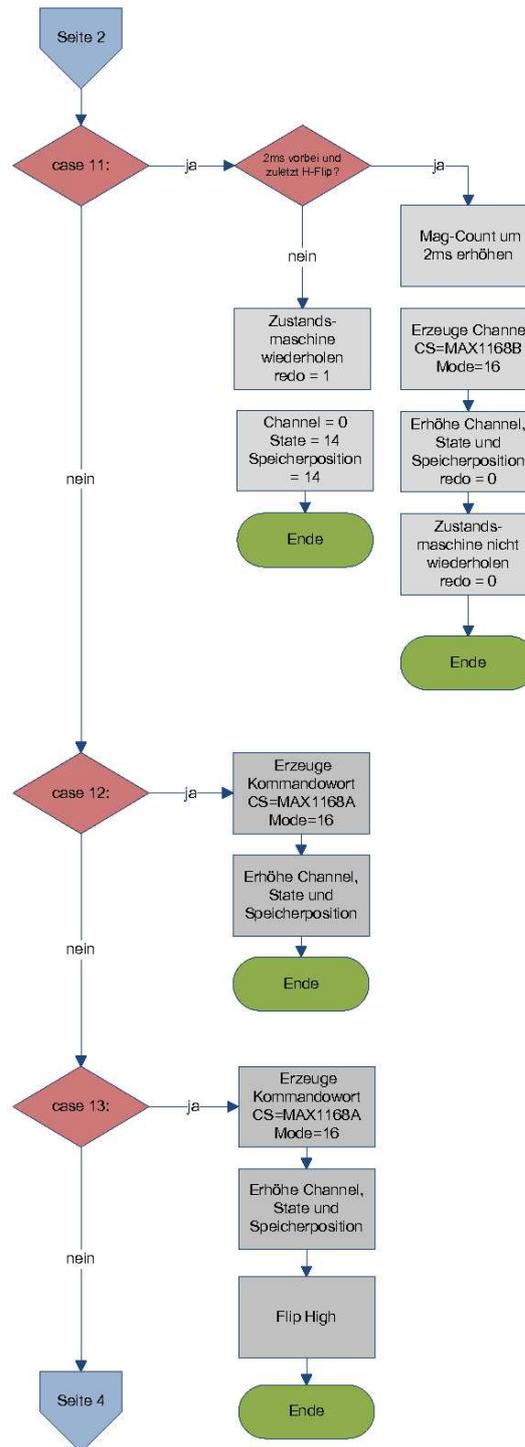


Bild 21.38: PAP "sm\_ctrl" Seite 3

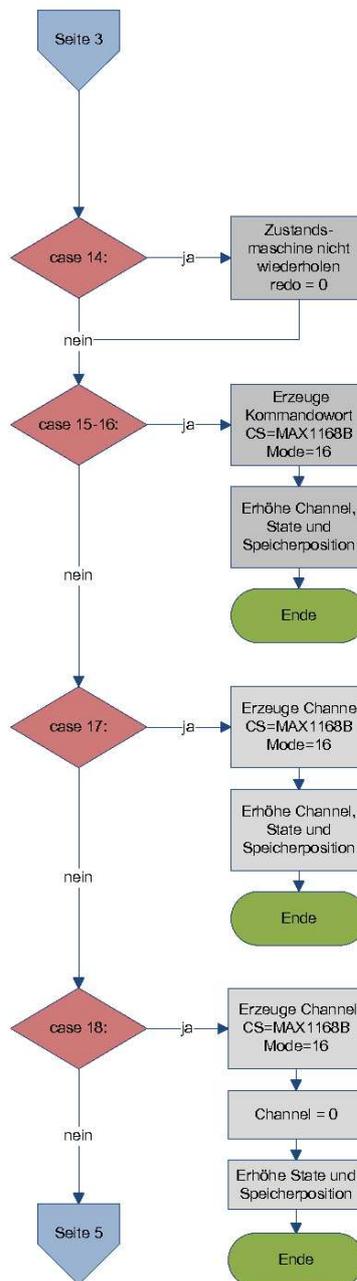


Bild 21.39: PAP "sm\_ctrl" Seite 4

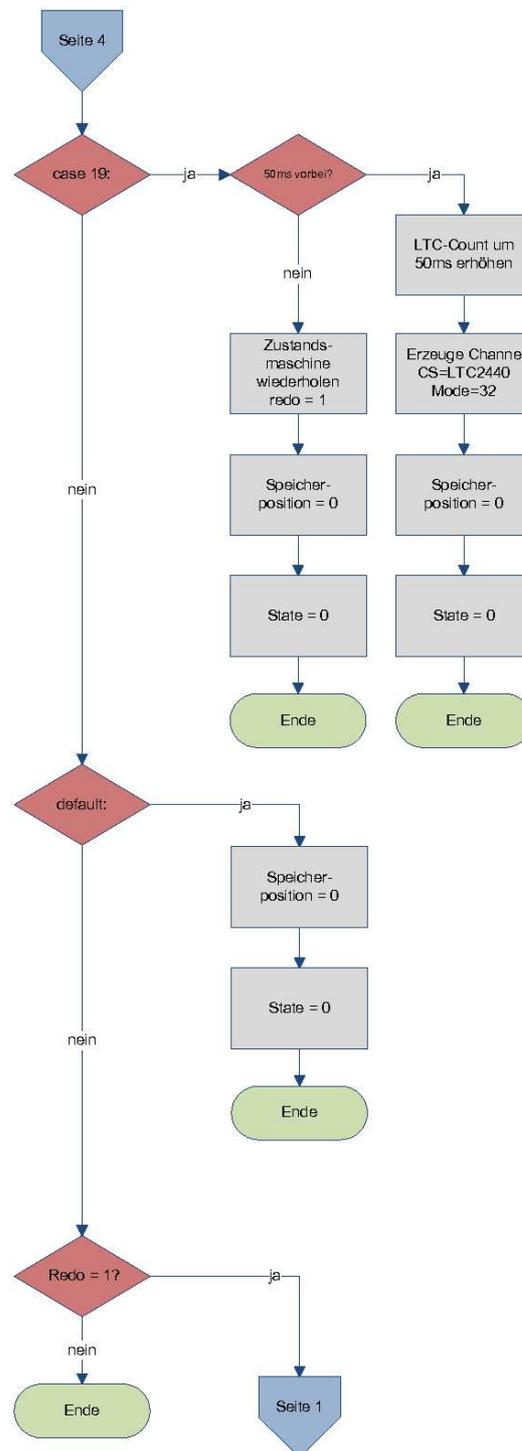


Bild 21.40: PAP "sm\_ctrl" Seite 5

## 21.6 Kalibrierung

### 21.6.1 Kalibrierkurzanleitung

#### 21.6.1.1 Gyroskope

- Aufbau Plattenspieler und Stromversorgung des Plattenspielers herstellen.
- Lichtschrankenelektronik mit Spannung versorgen (8-15V) und Verbindung zum Frequenzzähler herstellen (Tastkopf von Oszilloskop)

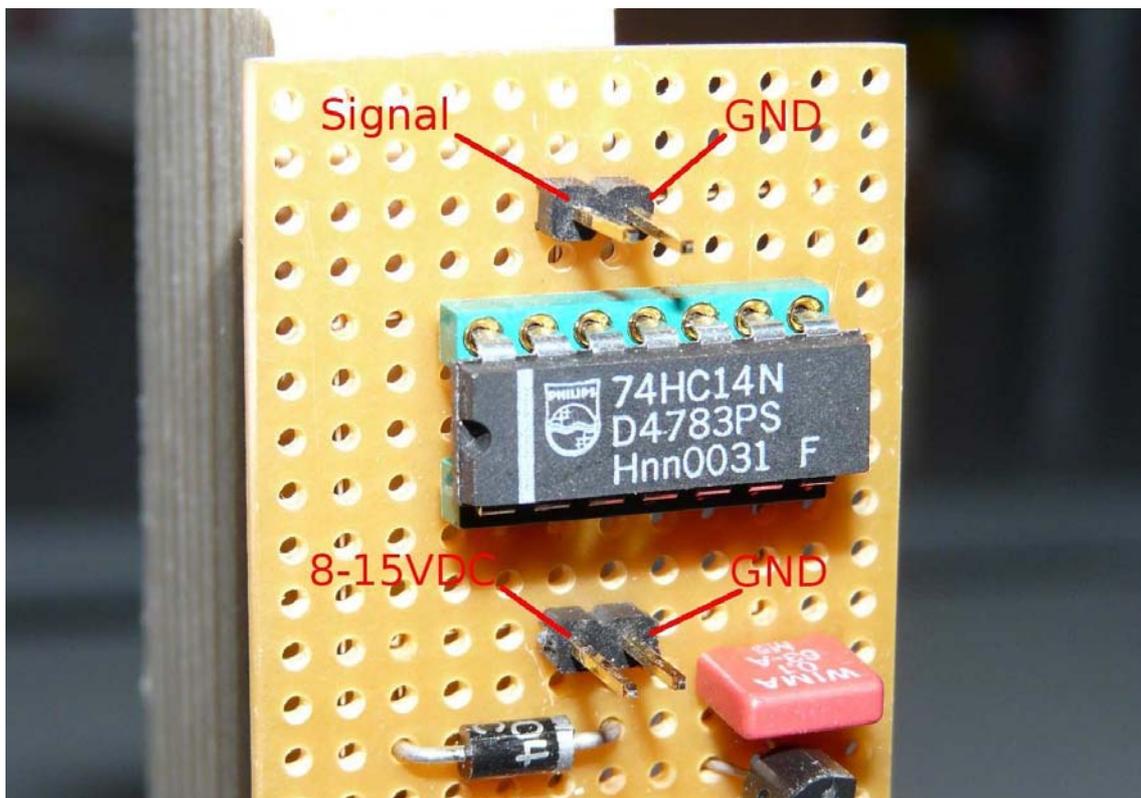


Bild 21.41: Anschlüsse Lichtschrankenelektronik

- Holzunterbrecherscheibe so positionieren, dass der Strahl der Lichtschranke unterbrochen wird, sie aber nirgends anstößt.

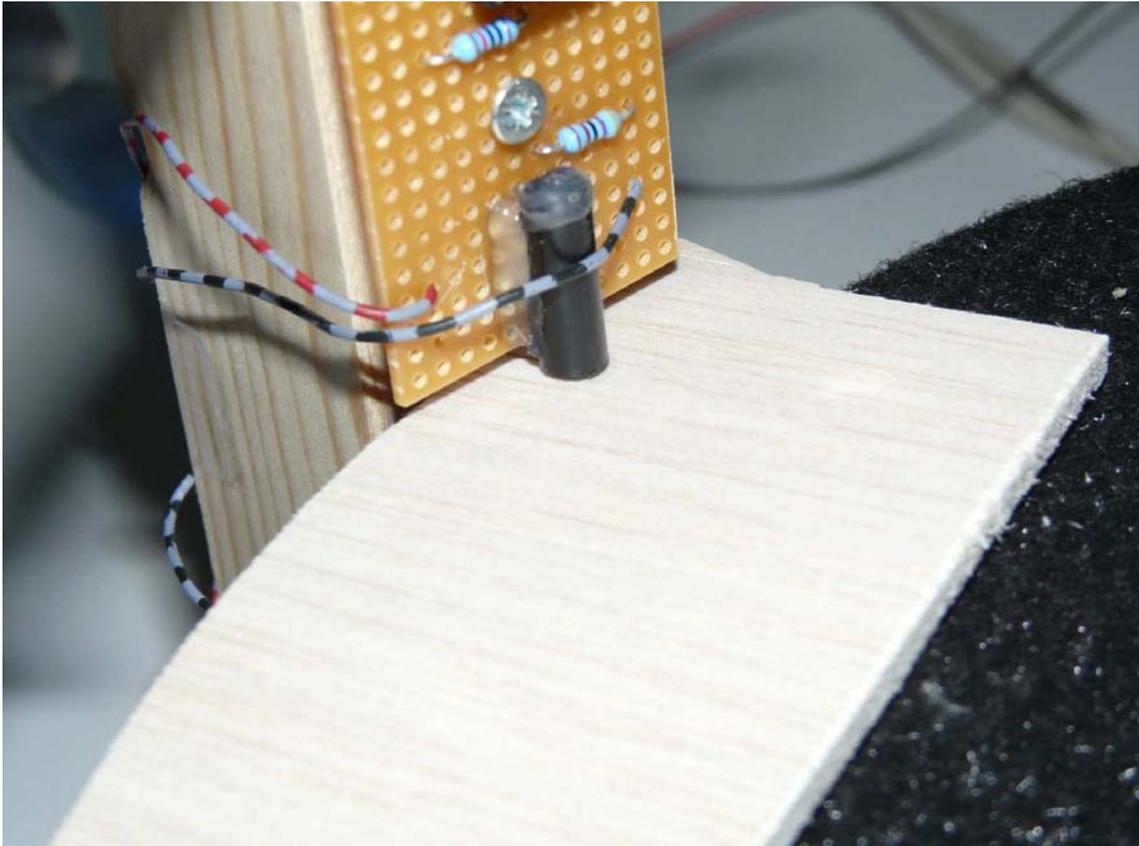


Bild 21.42: Position Holzunterbrecherscheibe

- Plattenspieler auf 45U/min einstellen und warm laufen lassen (ca. 15min) bis sich die angezeigten Werte stabilisieren.
- Plattenspieler mit Feinjustierungen abgleichen. 45U/min ergeben eine Frequenz von 750mHz, 33U/min ergeben 555,556mHz.
- Offsetbildung in main.c abschalten (`imu_offset_set`)
- Debugroutine `imu_dbg_full` in main.c aktivieren.
- Funkstrecke betriebsbereit machen
- IMU auf Halter aufschrauben (X Gyroskop muss zur Seite mit den Löchern zeigen (Bild 21.43)).
- Das Aufzeichnen der Werte geschieht am Laborrechner über minicom. Die Aufzeichnung wird mit `strg a + l` gestartet und beendet.

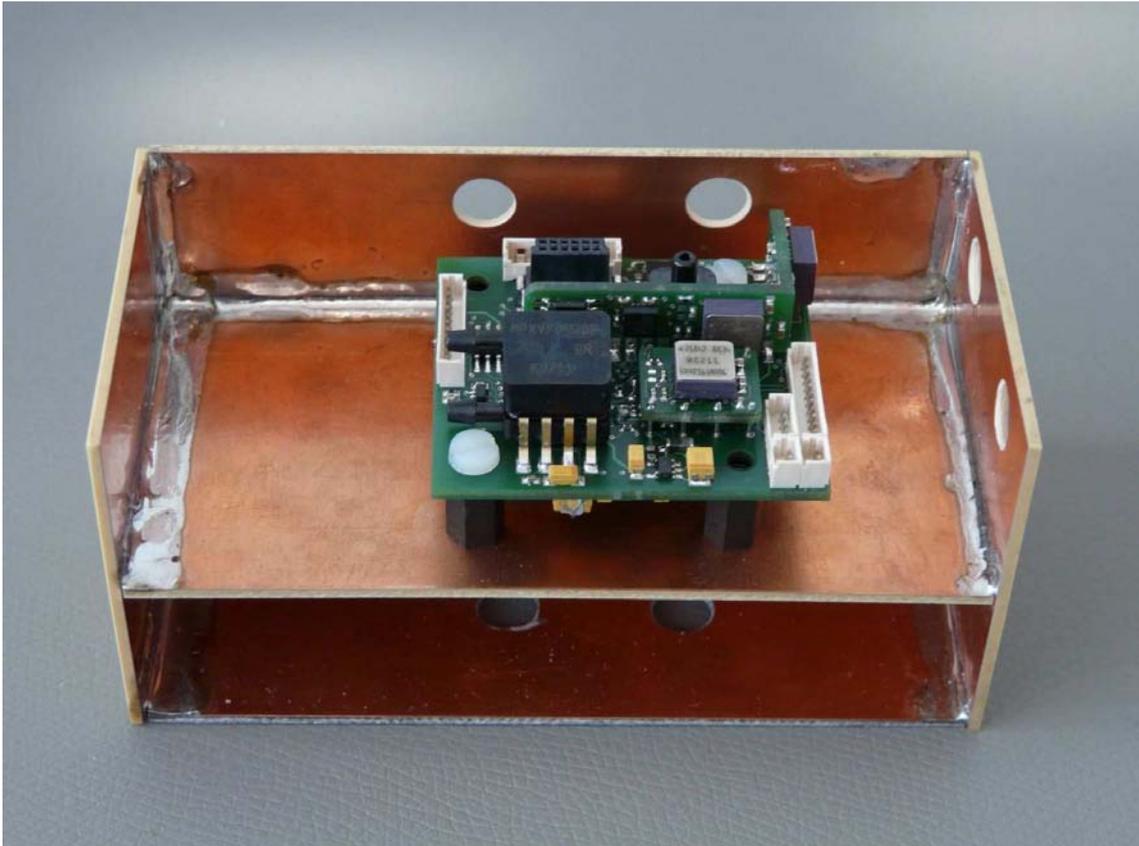


Bild 21.43: Montage der IMU auf Kalibrierhalterung (X-, Y-, Z-Gyroskop)

- Zum Kalibrieren des X-Gyroskop Halter hochkant aufstellen, so dass die Achse des Plattenspieler in dem Loch steckt welches sich direkt unter dem X Gyroskop befindet.
- Plattenspieler auf 45U/min einschalten und warten bis die Drehzahl stabil ist. Anschließend die Werte 1 Minute aufzeichnen.
- Plattenspieler auf 33U/min einstellen und wiederum warten bis die Drehzahl stabil ist und die Werte 1 Minute aufzeichnen.
- Zum Kalibrieren des Y- und Z-Gyroskop den Halter mithilfe der Bohrungen so positionieren, dass sich der Y- bzw. Z-Gyroskop in der Achse des Plattenspieler befindet und anschließend die Werte wie beschrieben aufzeichnen.
- Zum Erzeugen einer entgegengesetzten Drehung ist die IMU um 180° verdreht auf dem Kalibrierhalter zu montieren (X-Gyroskop zeigt zur Seite ohne Löcher) (Bild 21.44), da der Plattenspieler nicht rückwärtsdrehen betrieben werden kann. X- und Y-Gyroskop können so wie beschrieben ausgemessen werden.

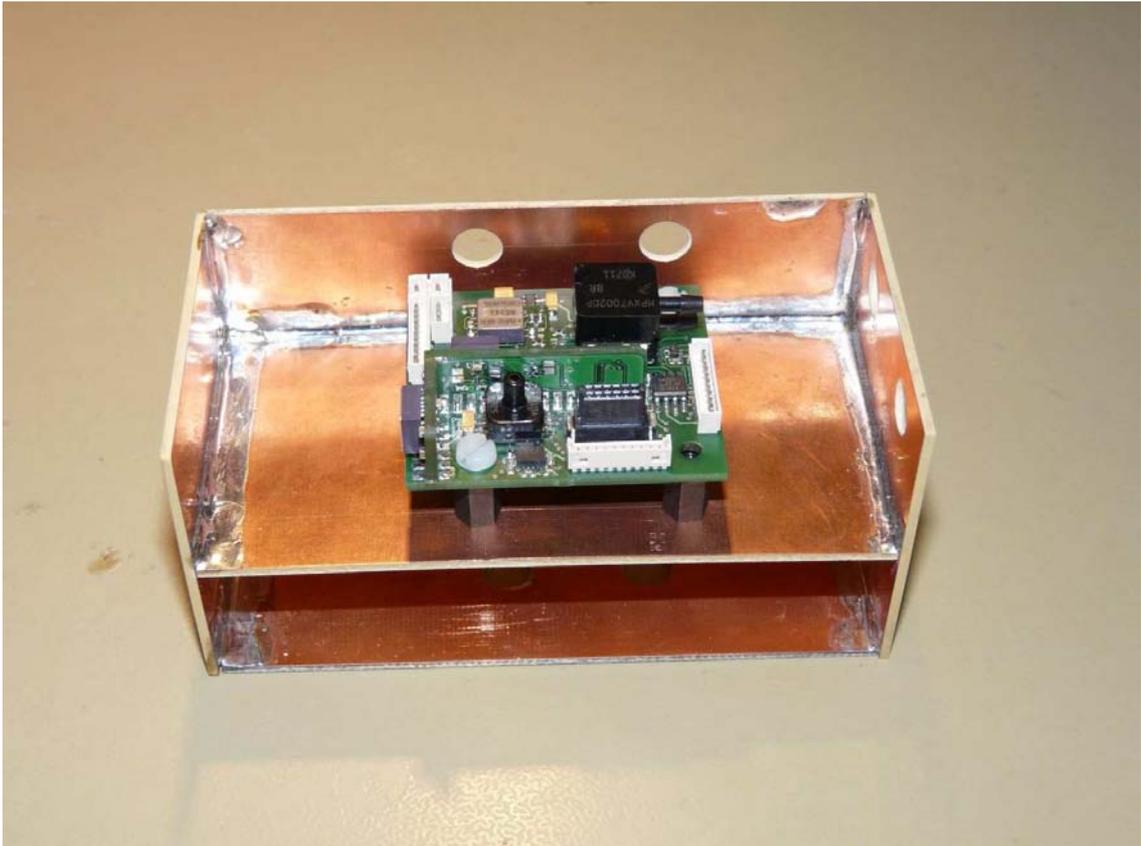


Bild 21.44: Montage der IMU auf Kalibrierhalterung für entgegengesetzte Drehrichtung (X-, Y-Gyroskop)

- Für die Kalibrierung des Z-Gyroskop in entgegengesetzter Drehrichtung ist die Platine über Kopf zu montieren (Bild 21.45). Hierzu sind unter Umständen zusätzliche Abstandsbolzen nötig, da die IMU sonst auf der Halterung aufliegt. Die MCU ist nun nicht mehr direkt aufsteckbar, so dass beide Platinen mit Kabel K-01 zu verbinden sind.

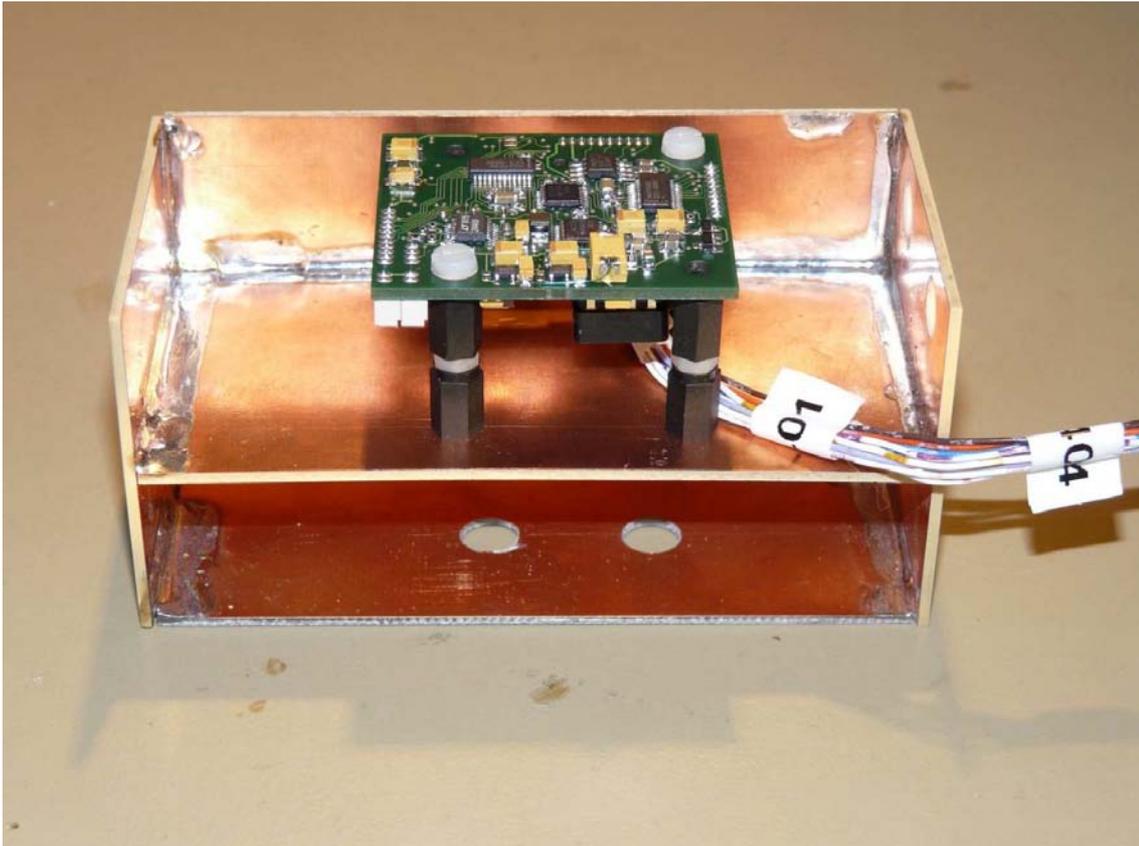


Bild 21.45: Montage der IMU auf Kalibrierhalterung für entgegengesetzte Drehrichtung (Z-Gyroskop)

#### 21.6.1.2 Beschleunigungssensoren

- IMU auf Halter aufschrauben (X Gyroskop muss zur Seite mit den Löchern zeigen (Bild 21.43)).
- Werte in allen möglichen sechs Positionen 1 Minute aufzeichnen. Dabei für jede Position eine neue Datei erstellen und erst aufzeichnen, wenn die IMU in neuer Position steht. Nicht alle Positionen in eine Datei schreiben, da ein späteres Filtern der einzelnen Positionen aufwändiger ist als separate Dateien zu erstellen.

## 21.6.2 Kalibriertabellen

### 21.6.2.1 Gyroskope

Tabelle 21.4: Mittelwerte für Kalibrierung X-Achse

<b>Drehgeschwindigkeit soll in U/min</b>	<b>Drehgeschwindigkeit ist in %s</b>	<b>Messwert in Bit</b>
33	199,980	47333
45	269,874	52657
-33	199,908	17059
-45	270,043	11745

Tabelle 21.5: Mittelwerte für Kalibrierung Y-Achse

<b>Drehgeschwindigkeit soll in U/min</b>	<b>Drehgeschwindigkeit ist in %s</b>	<b>Messwert in Bit</b>
33	199,980	47518
45	270,324	52931
-33	199,980	16642
-45	270,072	11211

Tabelle 21.6: Mittelwerte für Kalibrierung Z-Achse

<b>Drehgeschwindigkeit soll in U/min</b>	<b>Drehgeschwindigkeit ist in %s</b>	<b>Messwert in Bit</b>
33	199,800	47390
45	270,072	52675
-33	199,872	17192
-45	270,108	11891

### 21.6.2.2 Beschleunigungssensoren

Tabelle 21.7: Mittelwerte für Kalibrierung X-Achse +1g

<b>Sensor</b>	<b>Beschleunigung in g</b>	<b>Messwert in Bit</b>
ACC X	1	32210
ACC Y	0	24809
ACC Z	0	28131
ACC X2	1	29260
ACC Z2	0	26534

Tabelle 21.8: Mittelwerte für Kalibrierung X-Achse -1g

<b>Sensor</b>	<b>Beschleunigung in g</b>	<b>Messwert in Bit</b>
ACC X	-1	21740
ACC Y	0	24951
ACC Z	0	28218
ACC X2	-1	23058
ACC Z2	0	26440

Tabelle 21.9: Mittelwerte für Kalibrierung Y-Achse +1g

<b>Sensor</b>	<b>Beschleunigung in g</b>	<b>Messwert in Bit</b>
ACC X	0	27080
ACC Y	1	30115
ACC Z	0	28265
ACC X2	0	26223
ACC Z2	0	26516

Tabelle 21.10: Mittelwerte für Kalibrierung Y-Achse -1g

<b>Sensor</b>	<b>Beschleunigung in g</b>	<b>Messwert in Bit</b>
ACC X	0	26883
ACC Y	-1	19660
ACC Z	0	28048
ACC X2	0	26101
ACC Z2	0	26456

Tabelle 21.11: Mittelwerte für Kalibrierung Z-Achse +1g

<b>Sensor</b>	<b>Beschleunigung in g</b>	<b>Messwert in Bit</b>
ACC X	0	27012
ACC Y	0	24481
ACC Z	1	33657
ACC X2	0	26142
ACC Z2	1	29630

Tabelle 21.12: Mittelwerte für Kalibrierung Z-Achse -1g

Sensor	Beschleunigung in g	Messwert in Bit
ACC X	0	27006
ACC Y	0	24859
ACC Z	-1	22775
ACC X2	0	26216
ACC Z2	-1	23371

### 21.6.3 Grafisch Darstellung der Kalibrierwerte

#### 21.6.3.1 Gyroskope

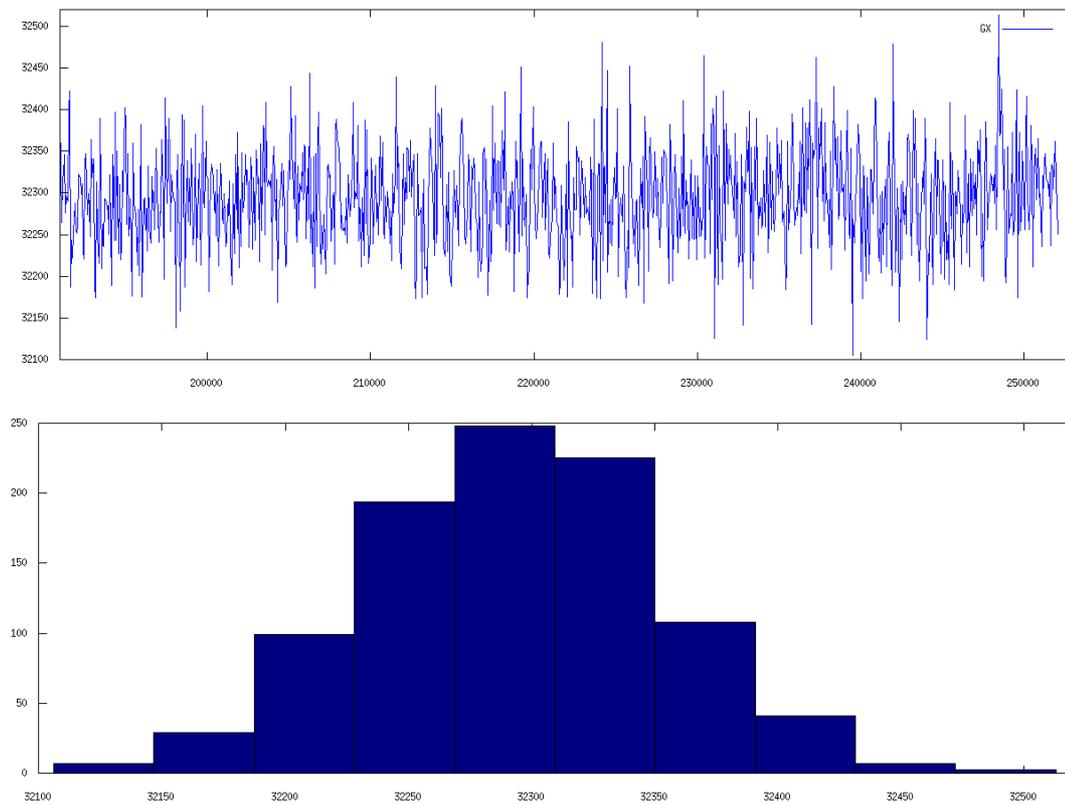


Bild 21.46: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse 0U/min

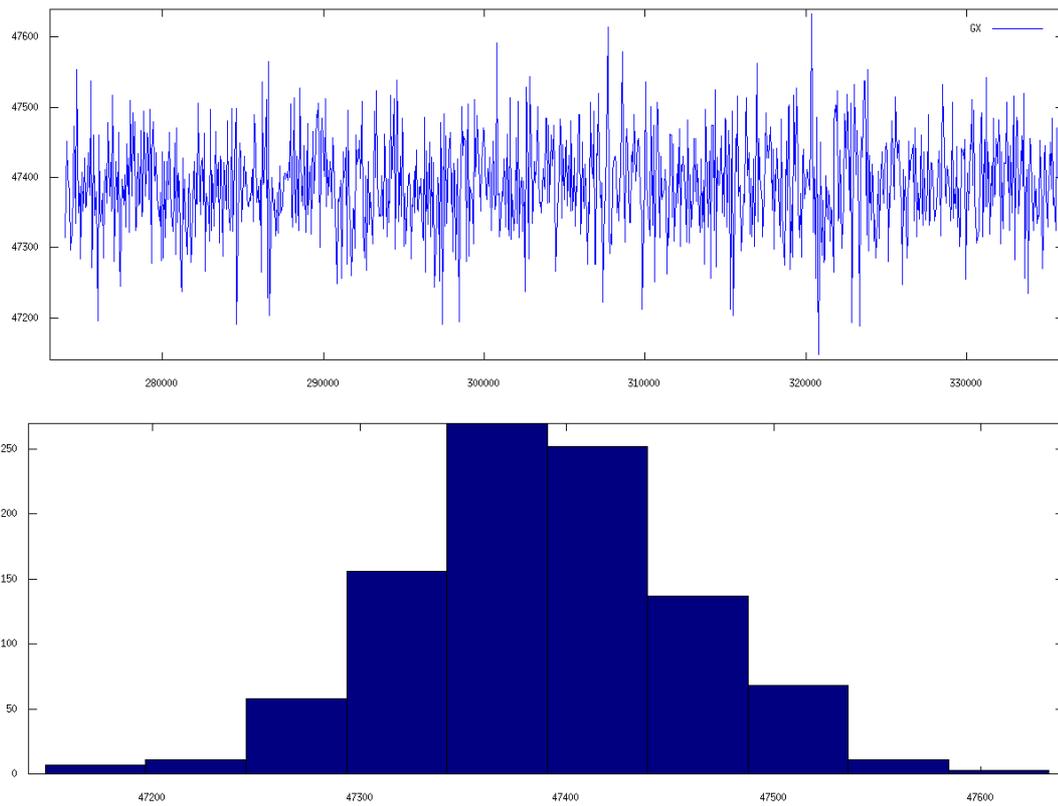


Bild 21.47: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse 33U/min

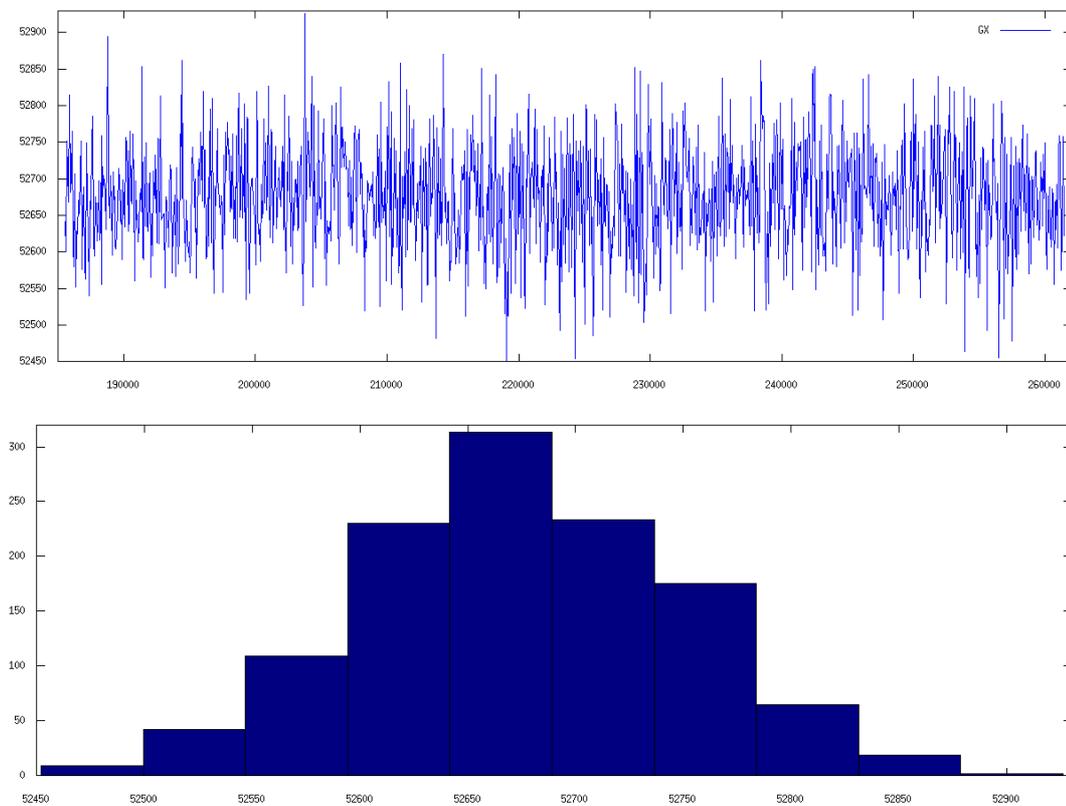


Bild 21.48: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse 45U/min

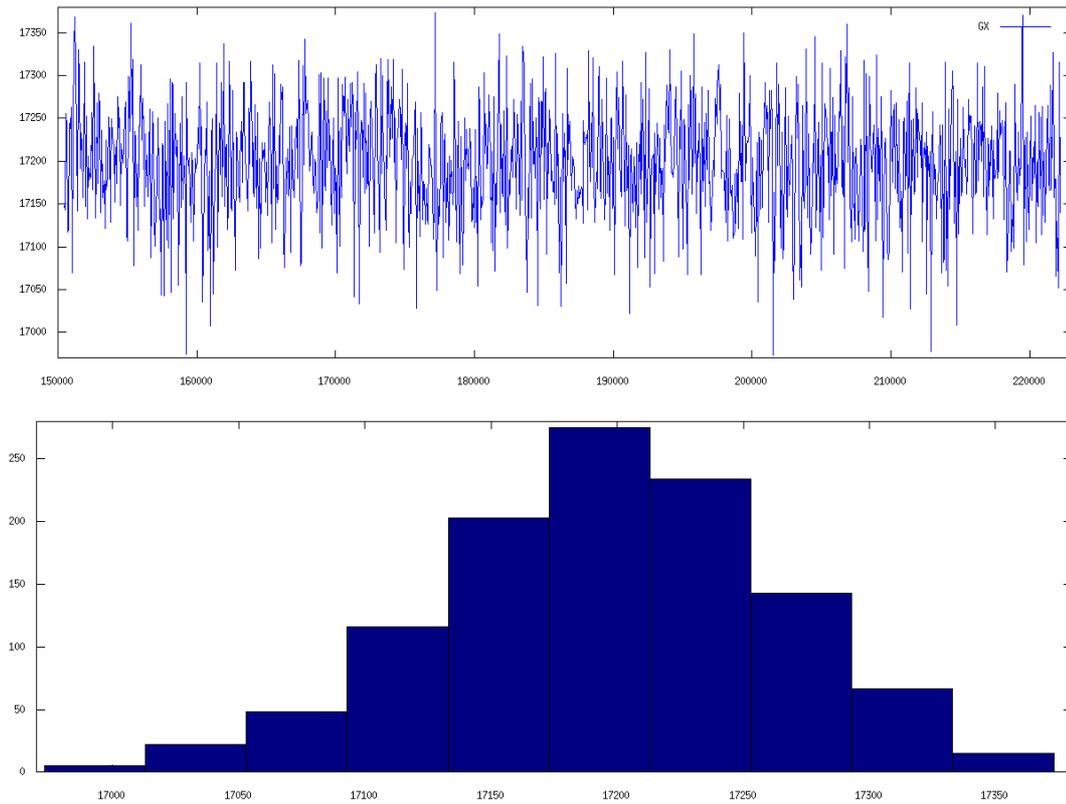


Bild 21.49: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse -33U/min

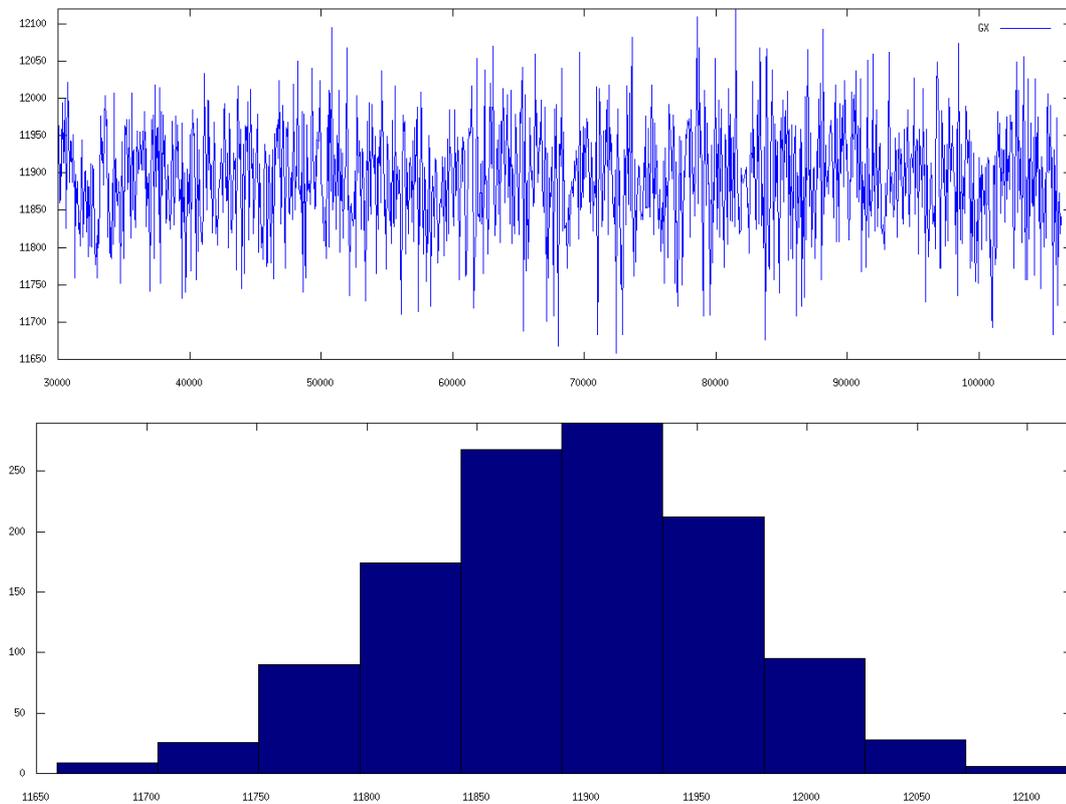


Bild 21.50: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse -45U/min

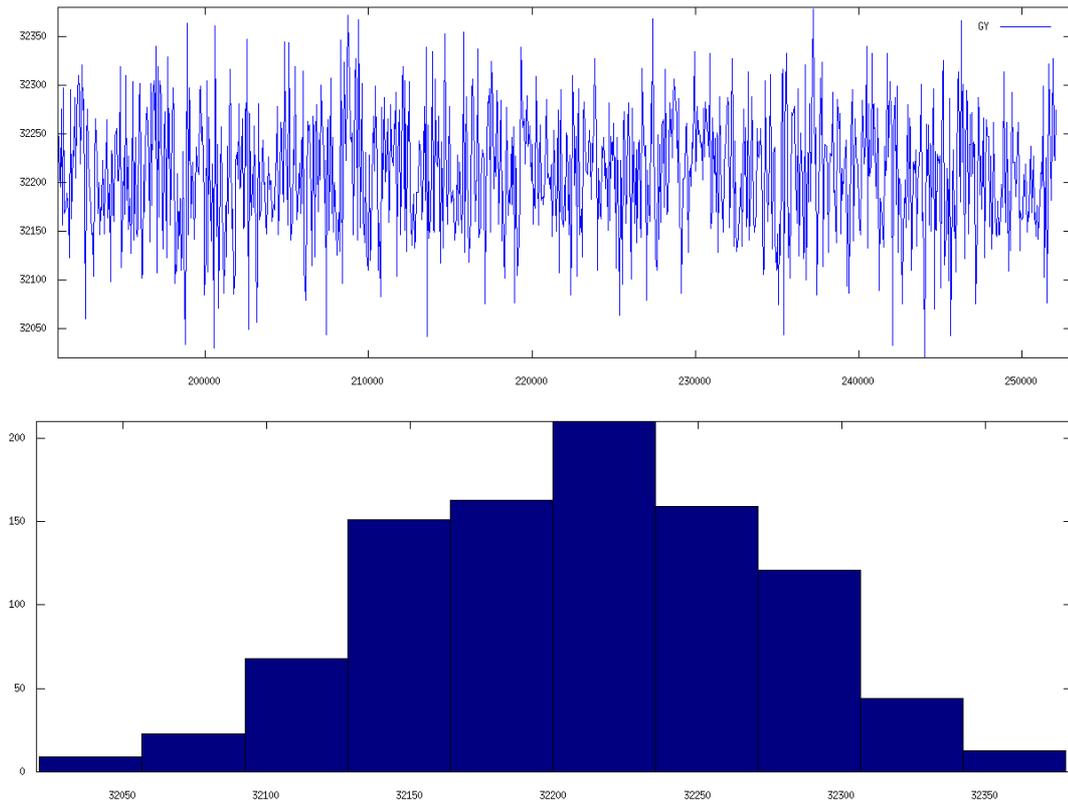


Bild 21.51: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 0U/min

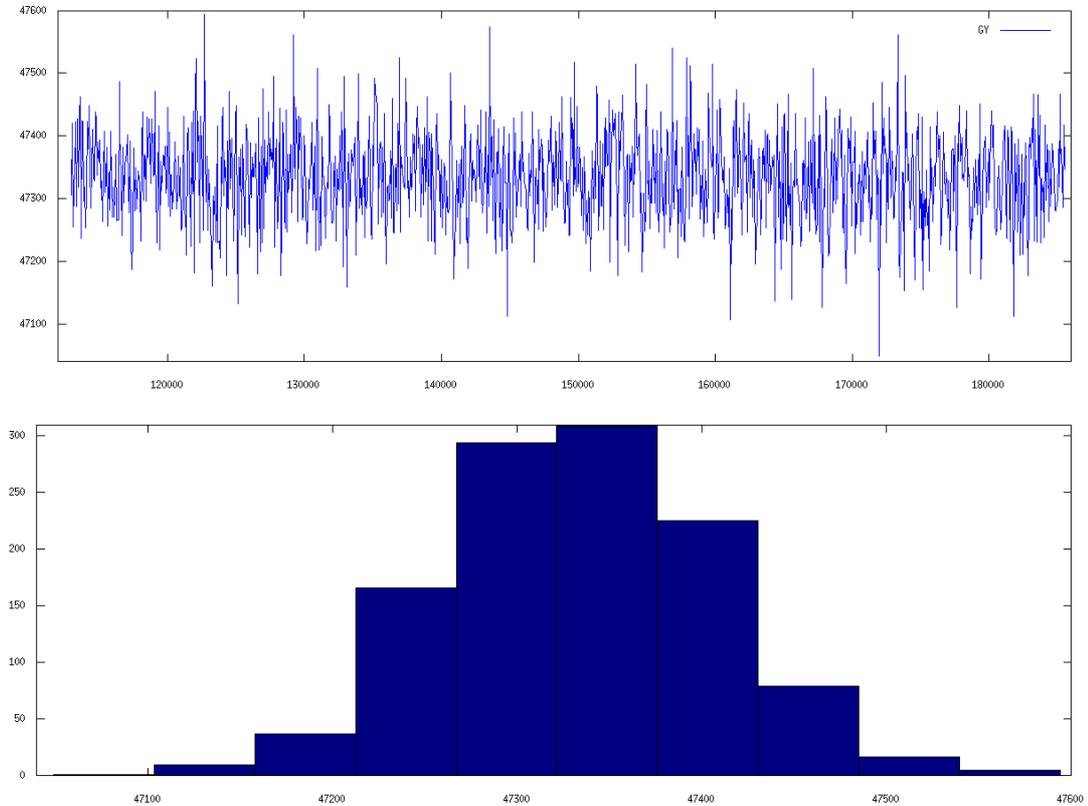


Bild 21.52: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 33U/min

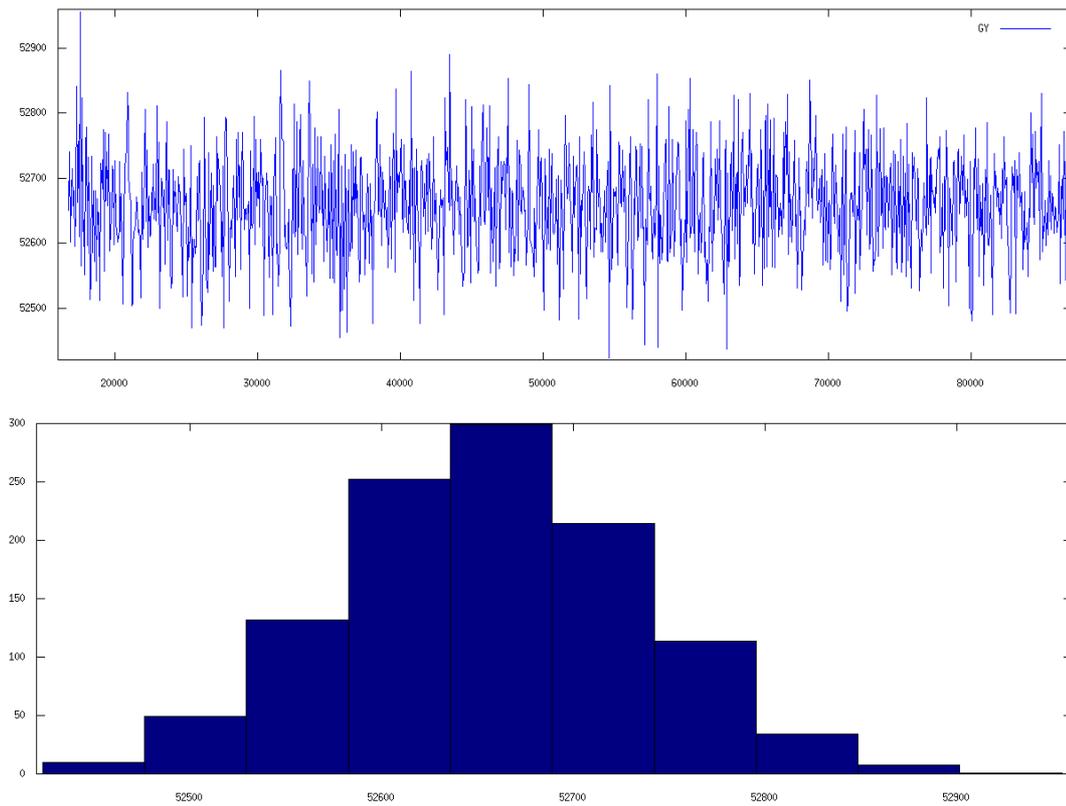


Bild 21.53: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 45U/min

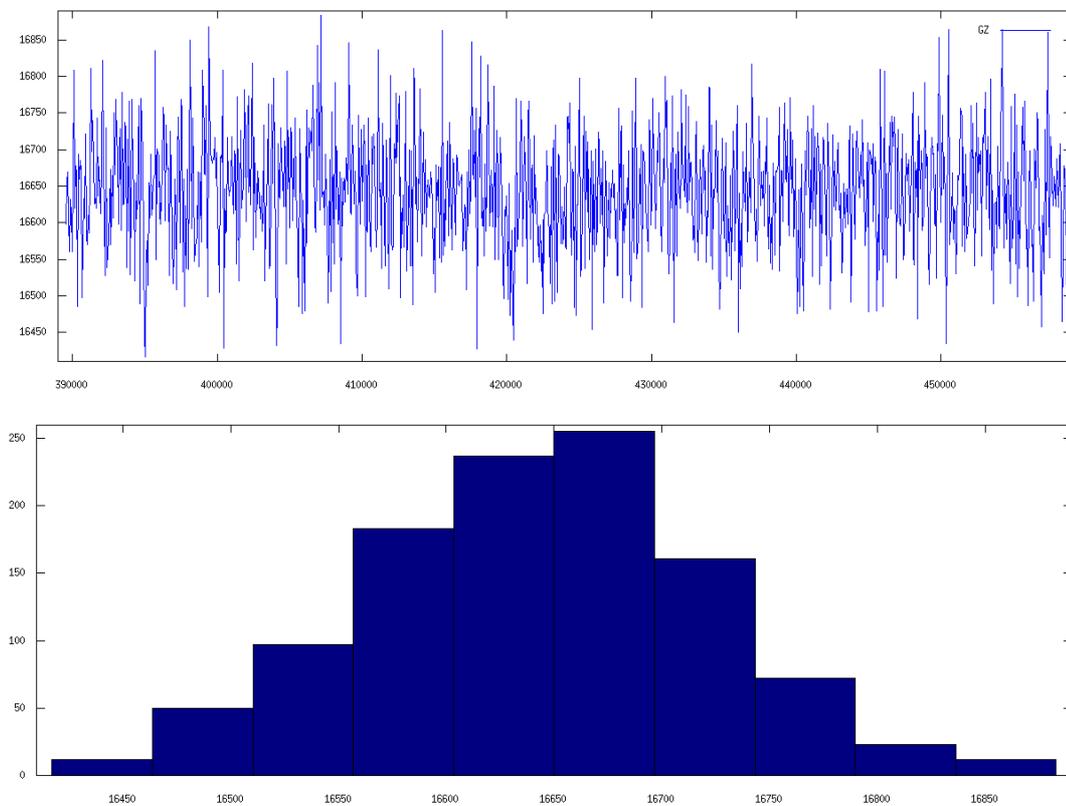


Bild 21.54: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse -33U/min

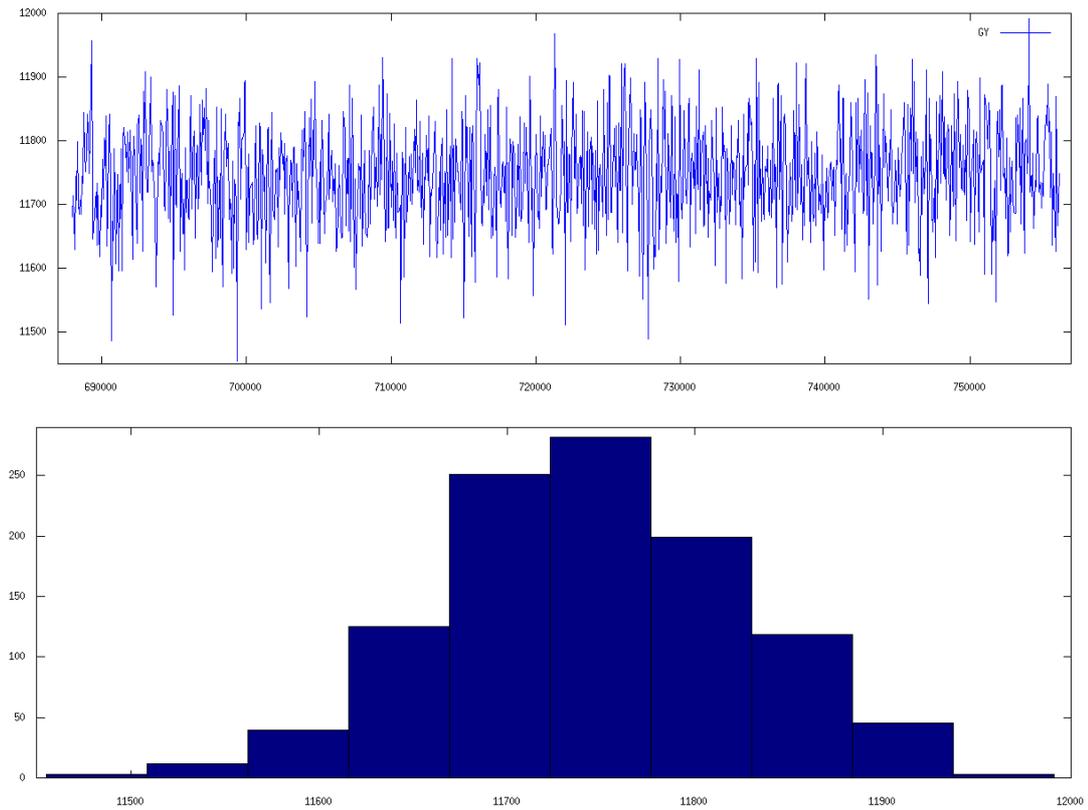


Bild 21.55: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse -45U/min

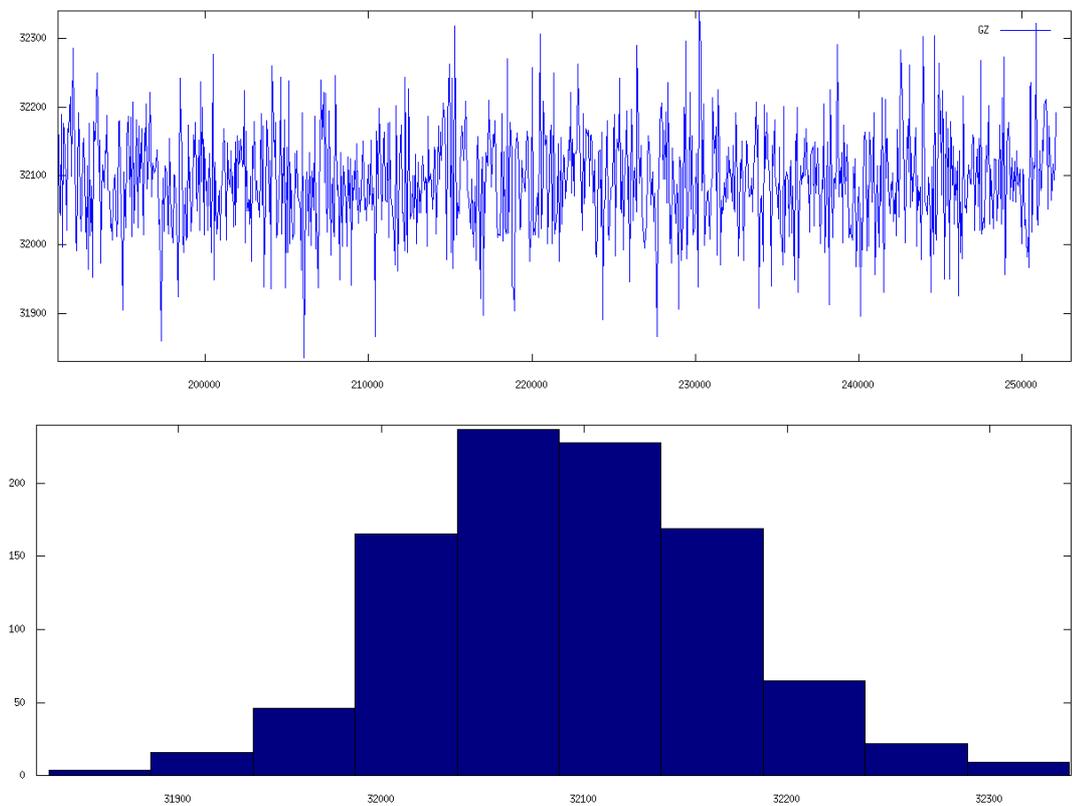


Bild 21.56: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 0U/min

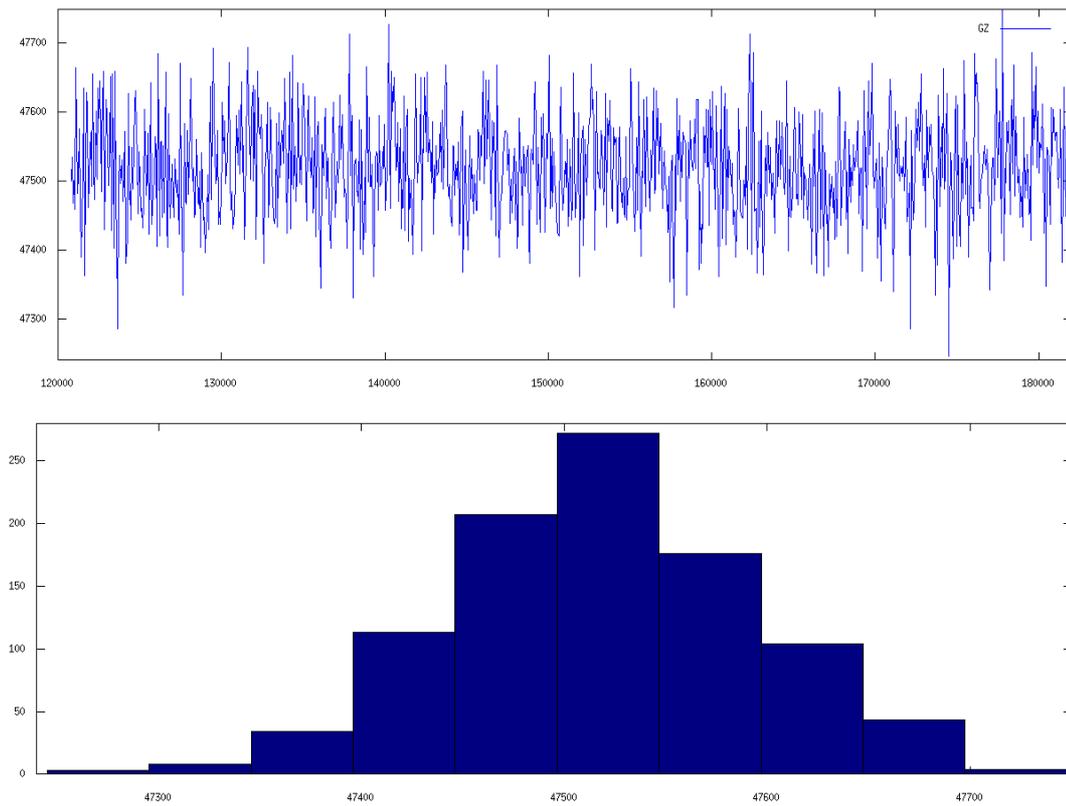


Bild 21.57: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 33U/min

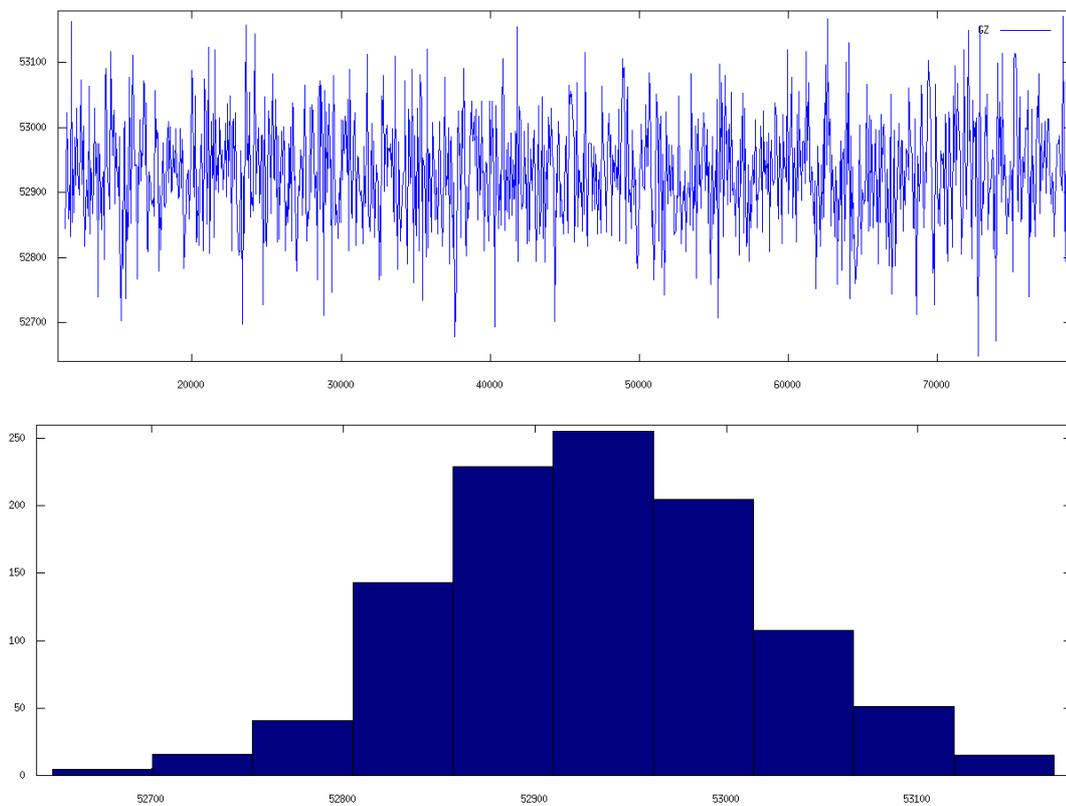


Bild 21.58: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 45U/min

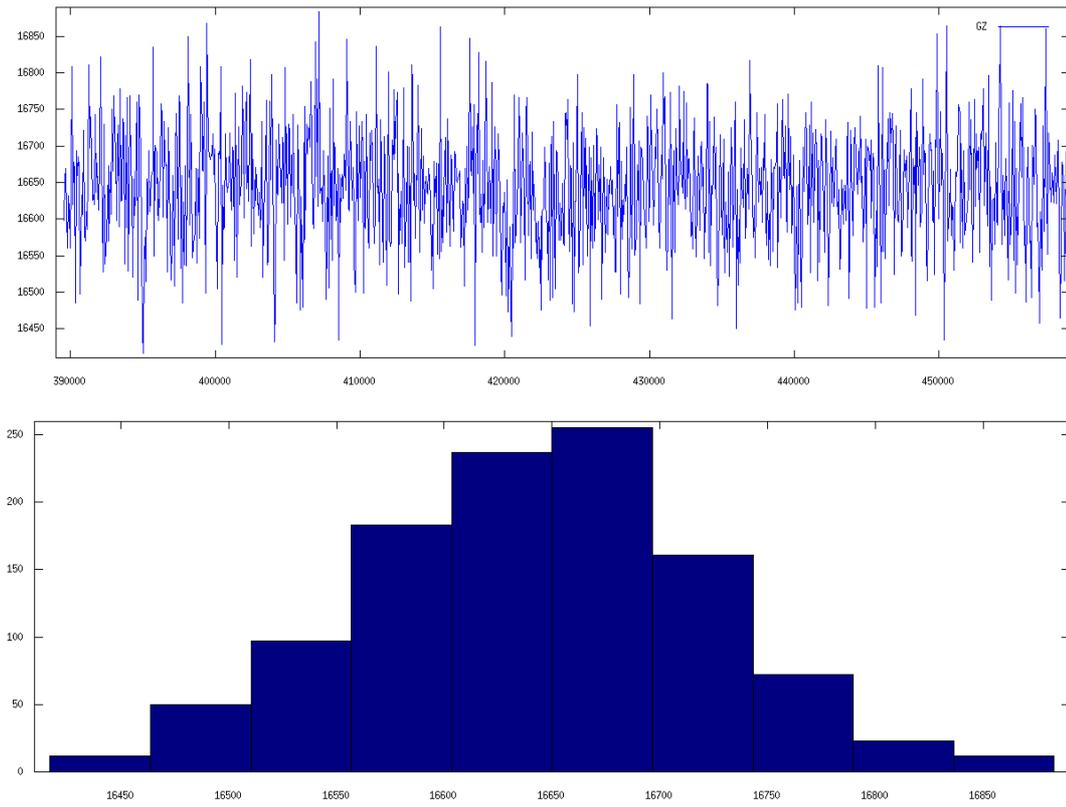


Bild 21.59: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse -33U/min

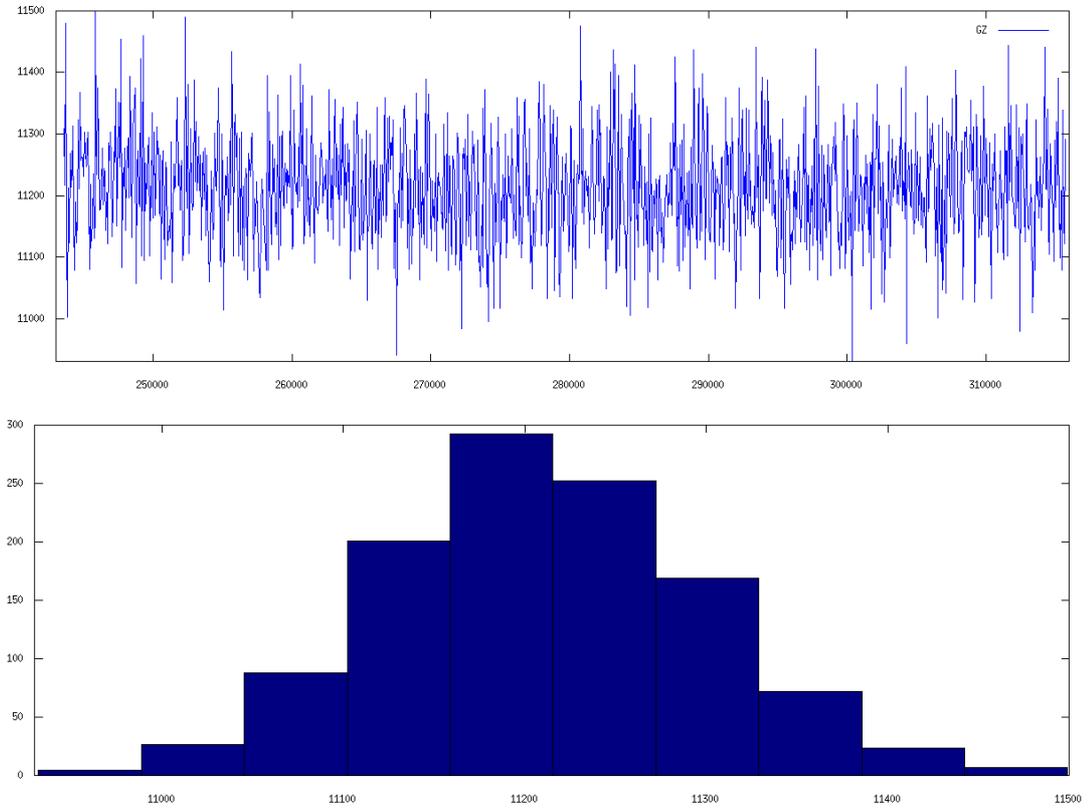


Bild 21.60: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse -45U/min

## 21.6.3.2 Beschleunigungssensoren

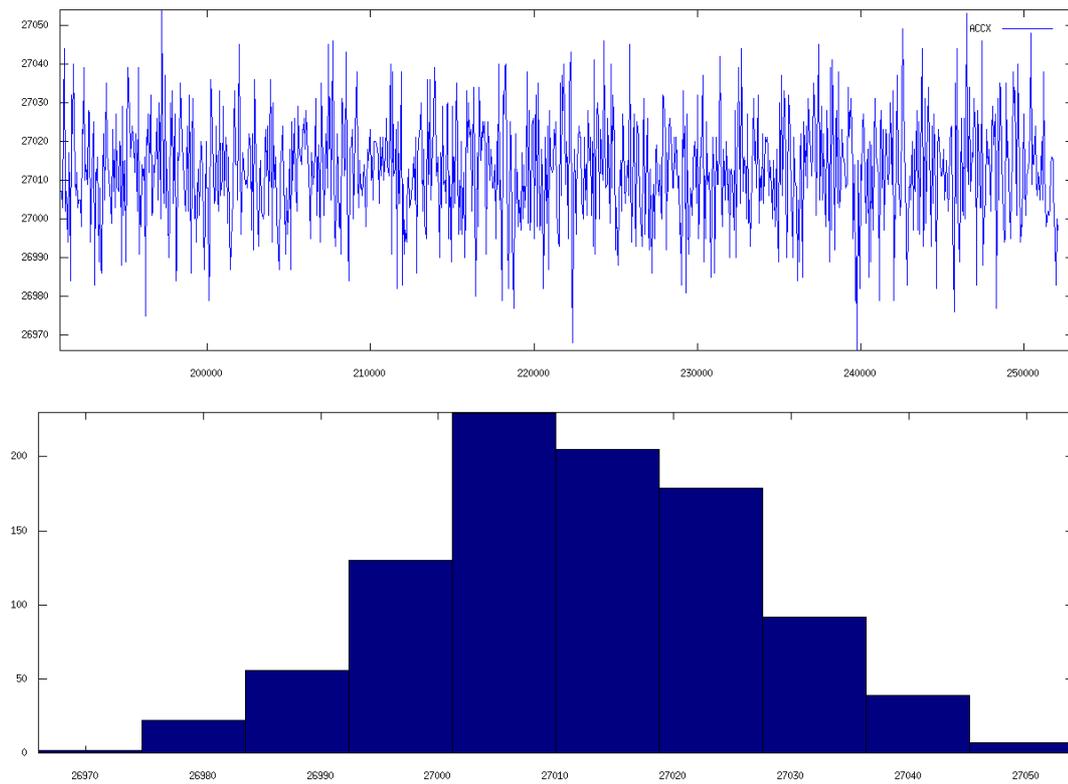


Bild 21.61: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse  $Og$

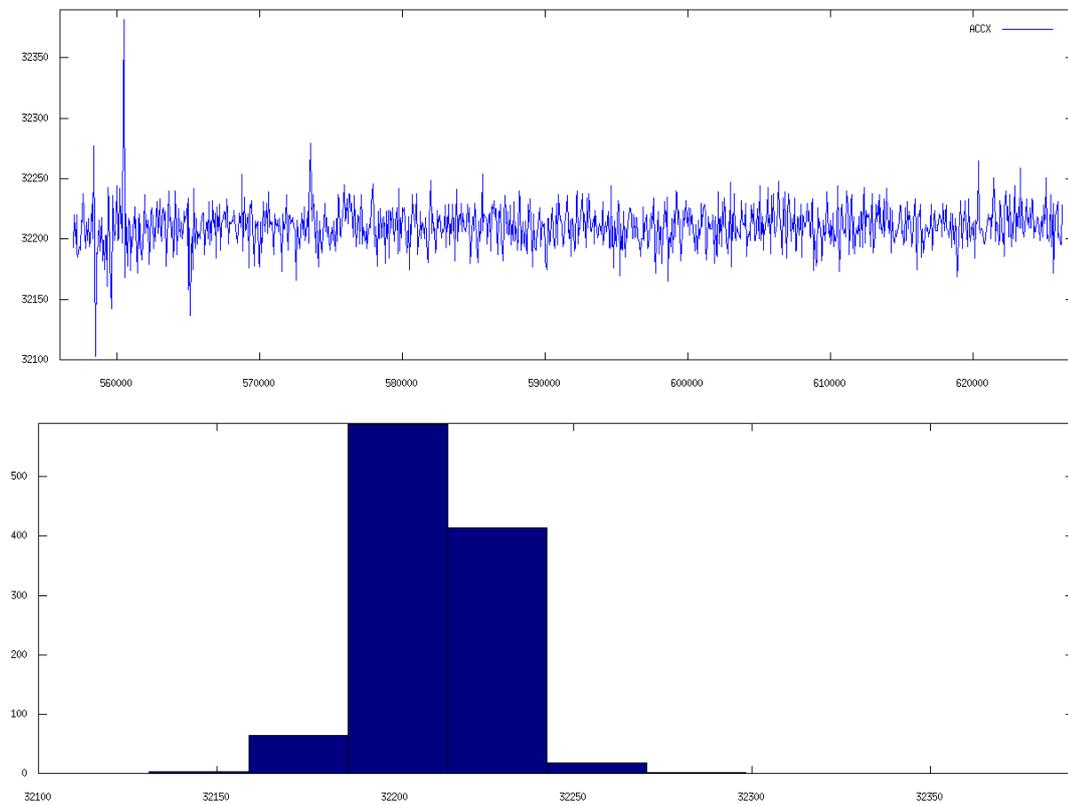


Bild 21.62: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse  $+1g$

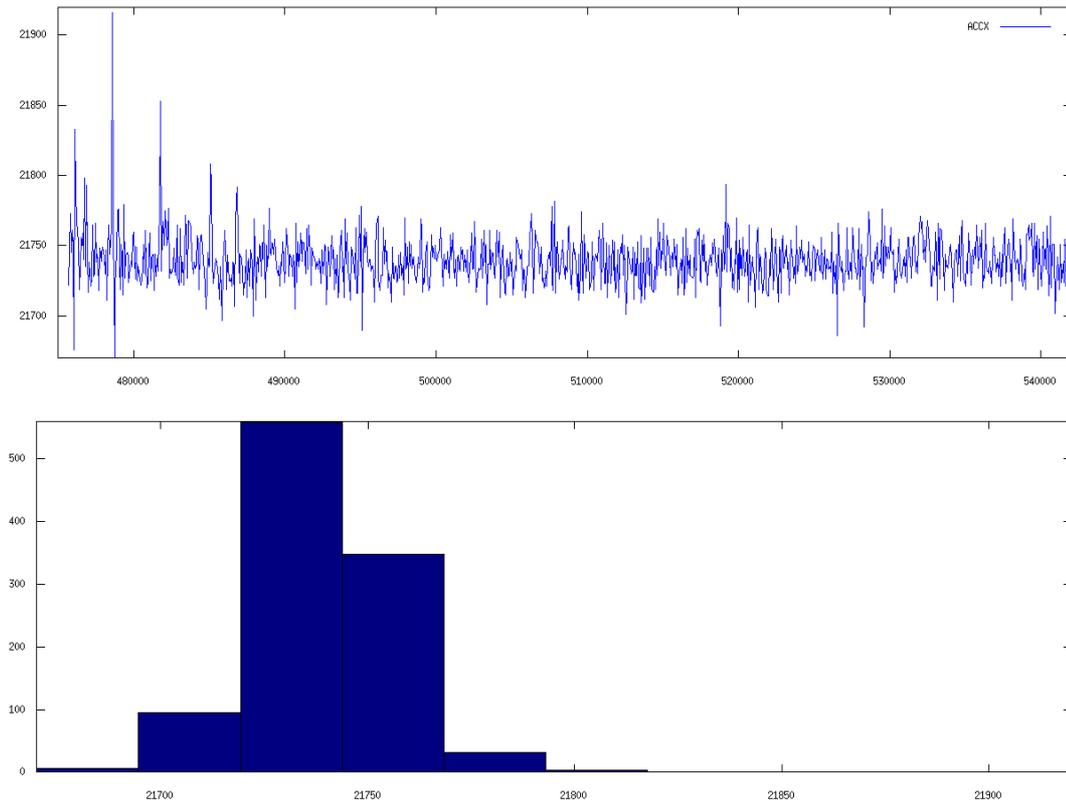


Bild 21.63: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X-Achse -1g

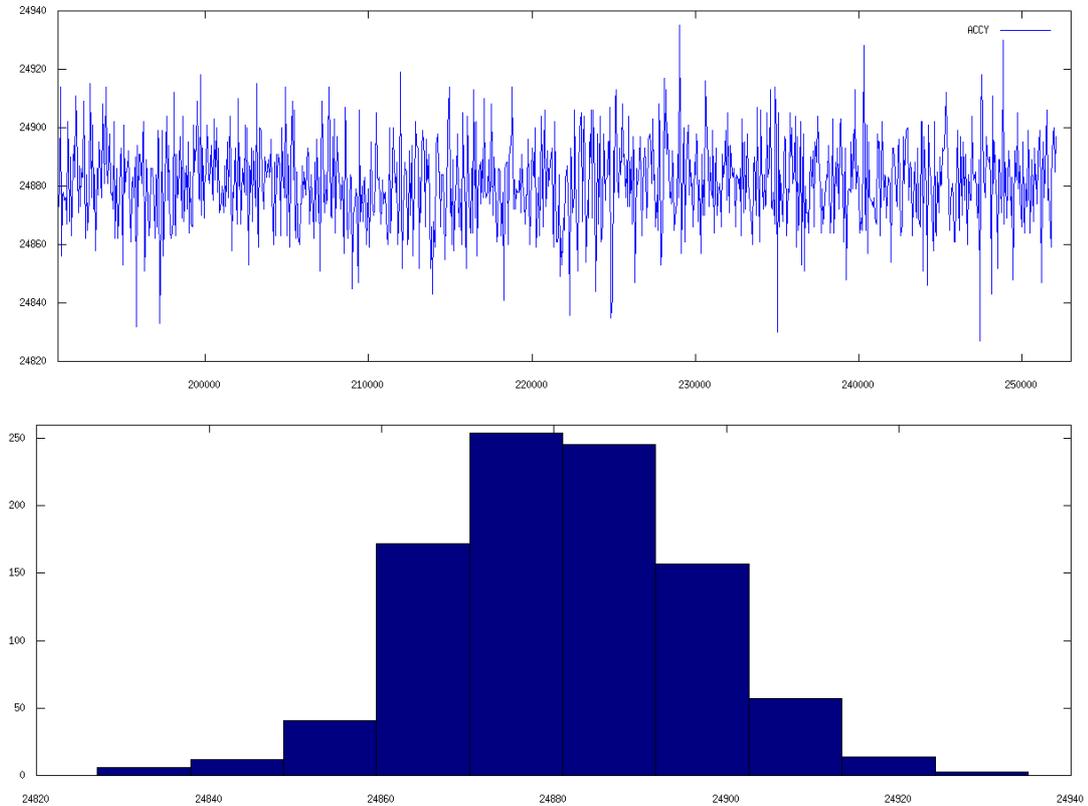


Bild 21.64: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse 0g

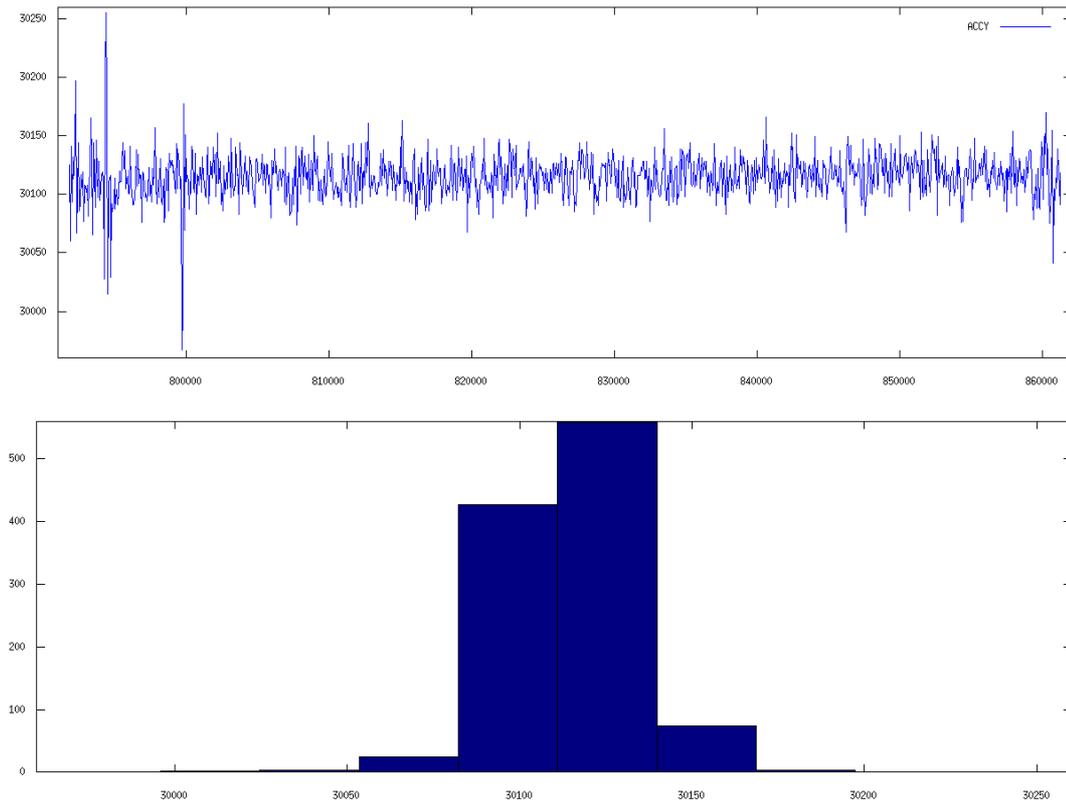


Bild 21.65: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse +1g

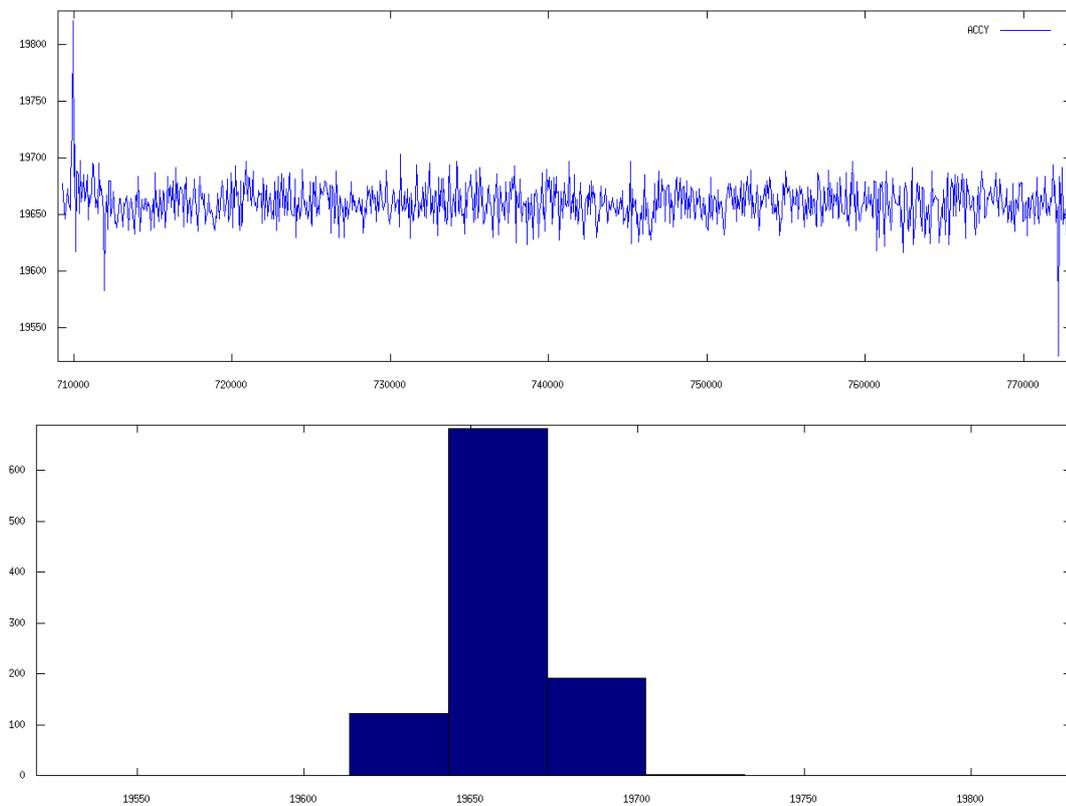


Bild 21.66: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Y-Achse -1g

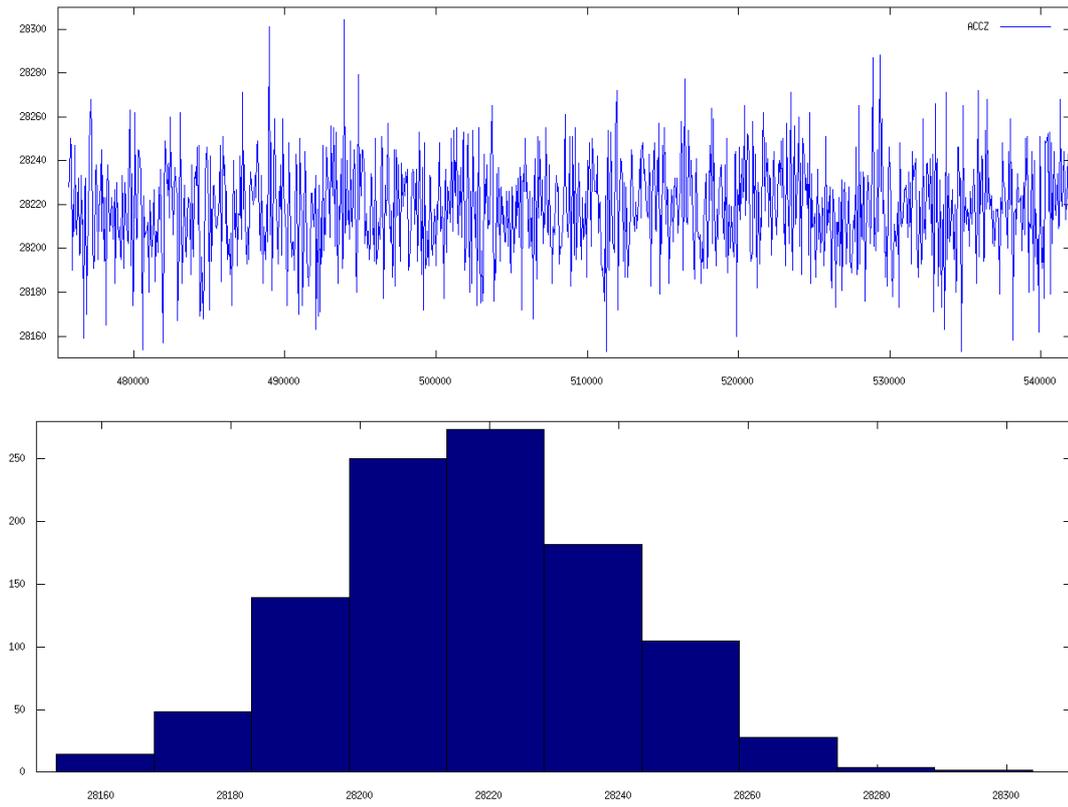


Bild 21.67: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse 0g

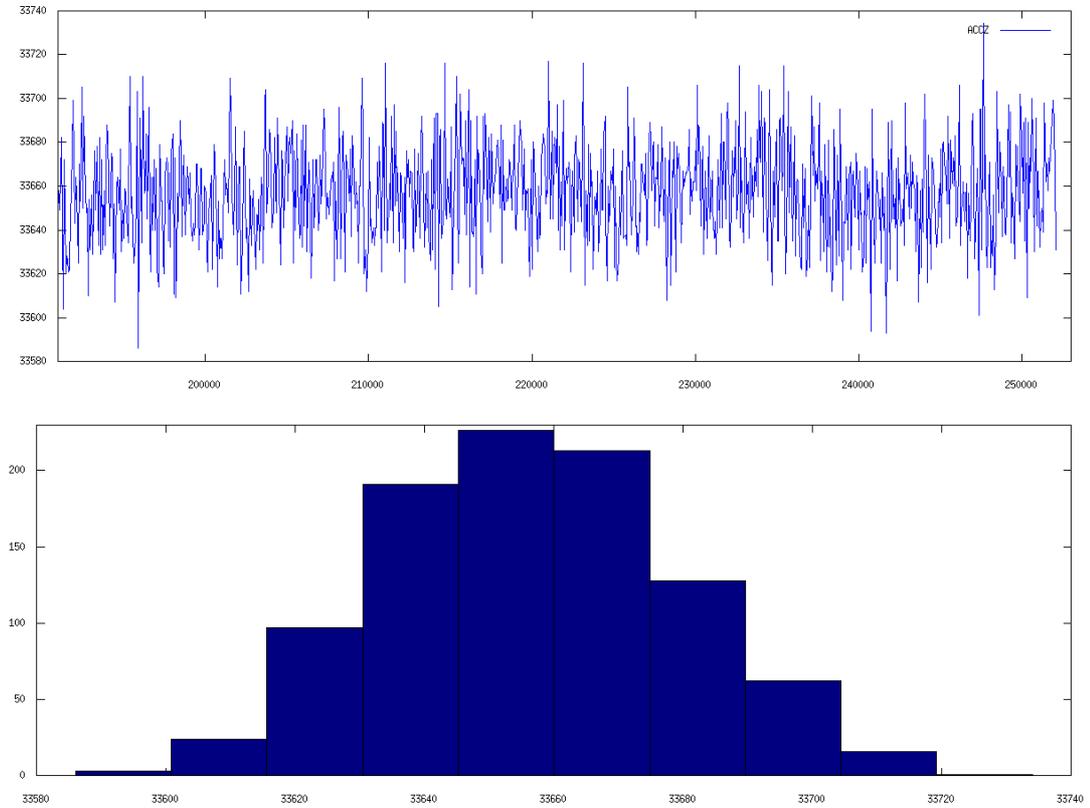


Bild 21.68: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse +1g

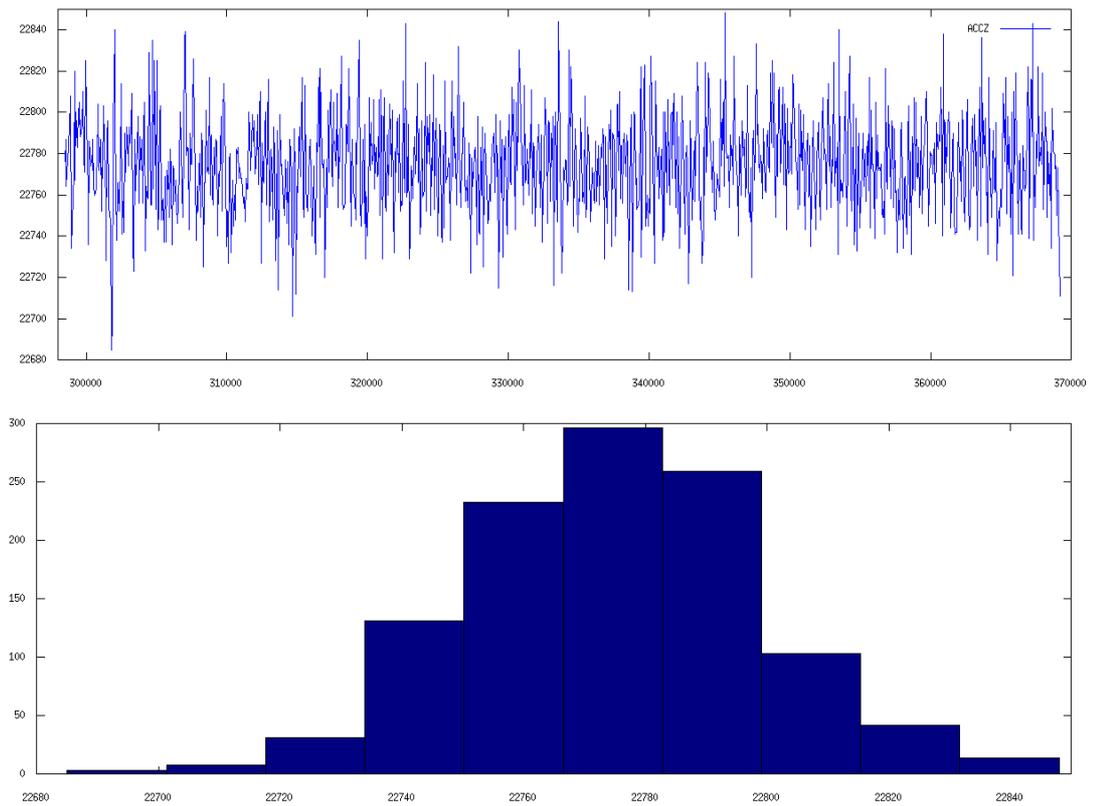


Bild 21.69: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z-Achse -1g

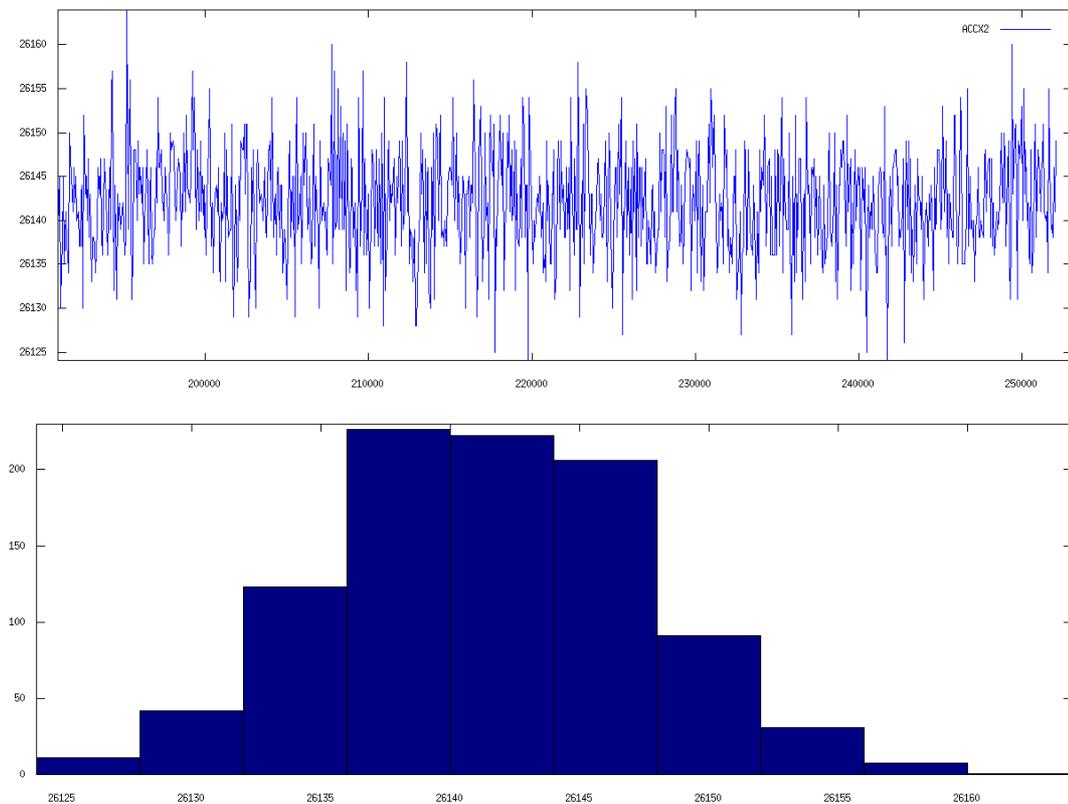


Bild 21.70: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X2-Achse 0g

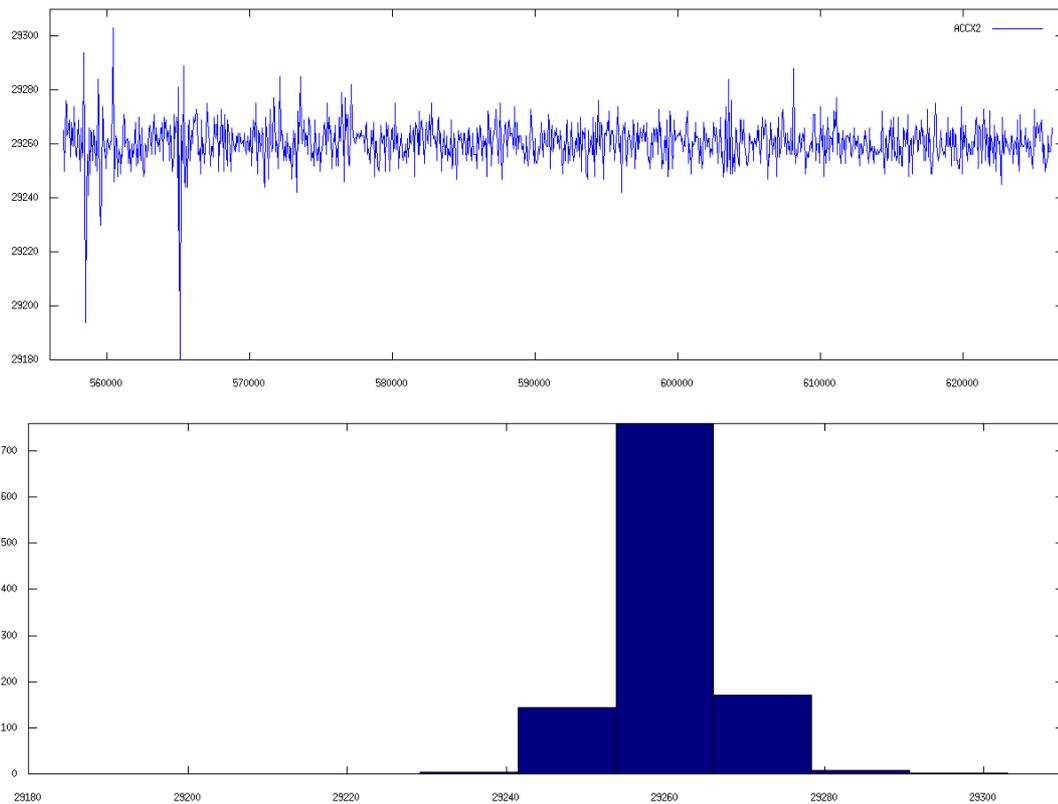


Bild 21.71: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X2-Achse +1g

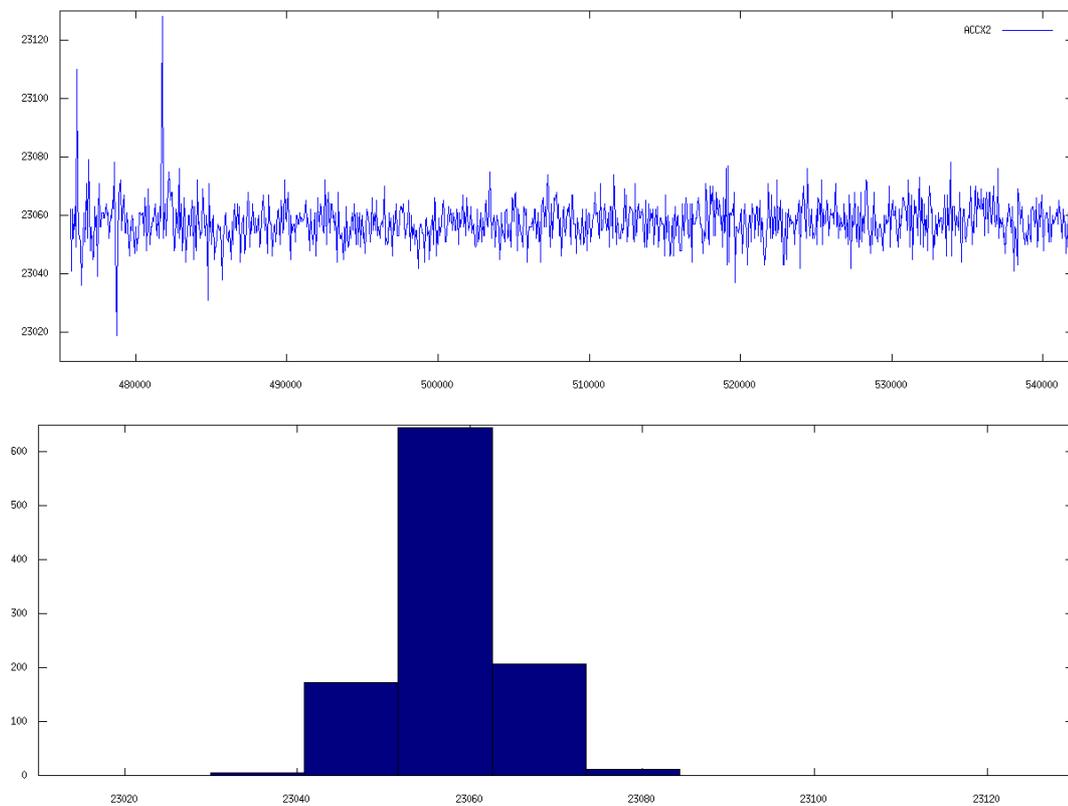


Bild 21.72: Grafische Darstellung der Kalibriermesswerte mit Histogramm, X2-Achse -1g

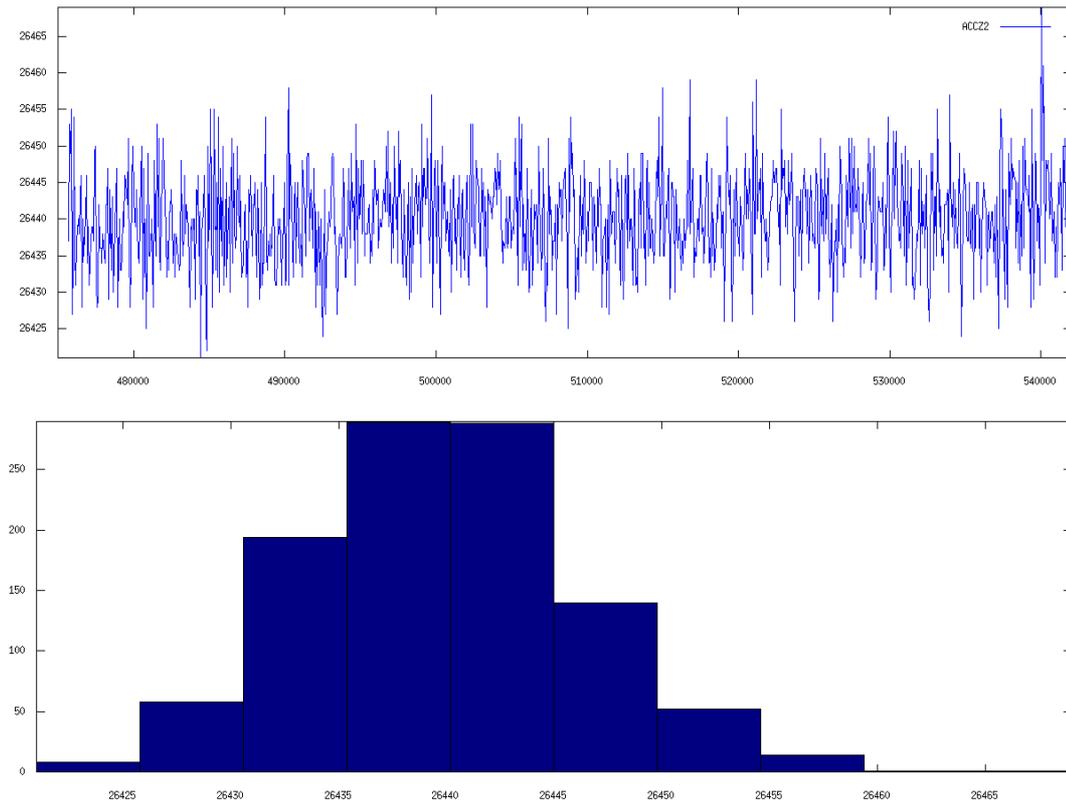


Bild 21.73: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z2-Achse 0g

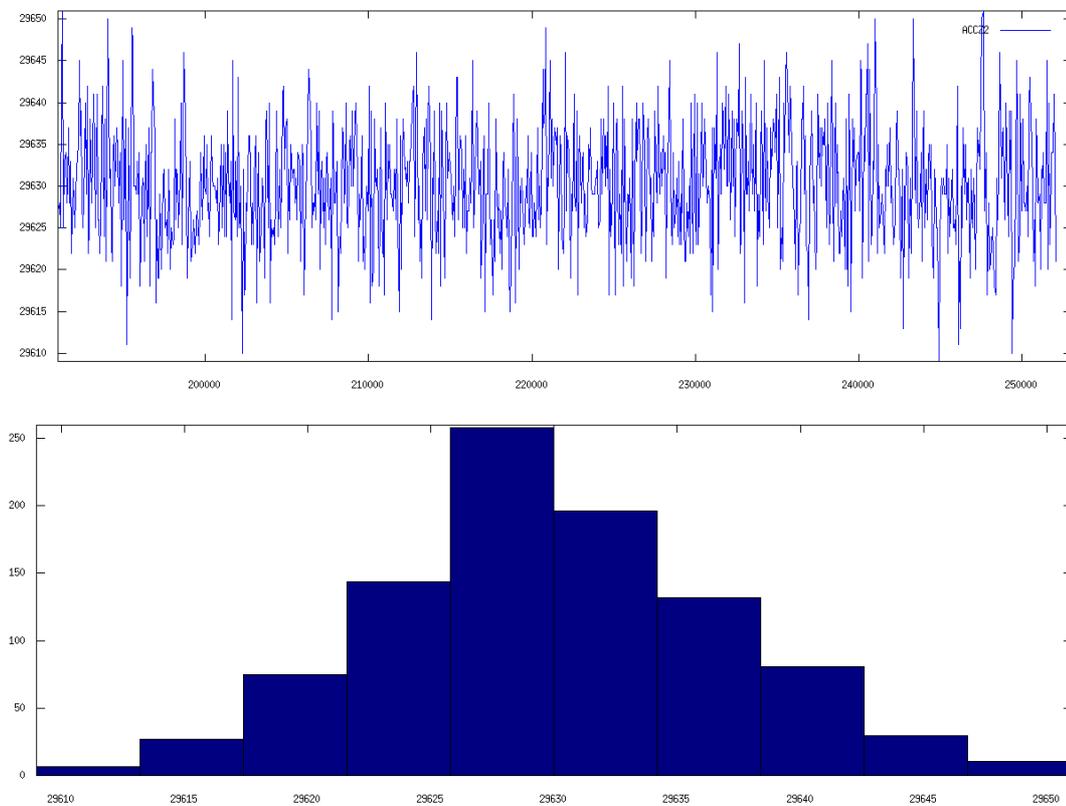


Bild 21.74: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z2-Achse +1g

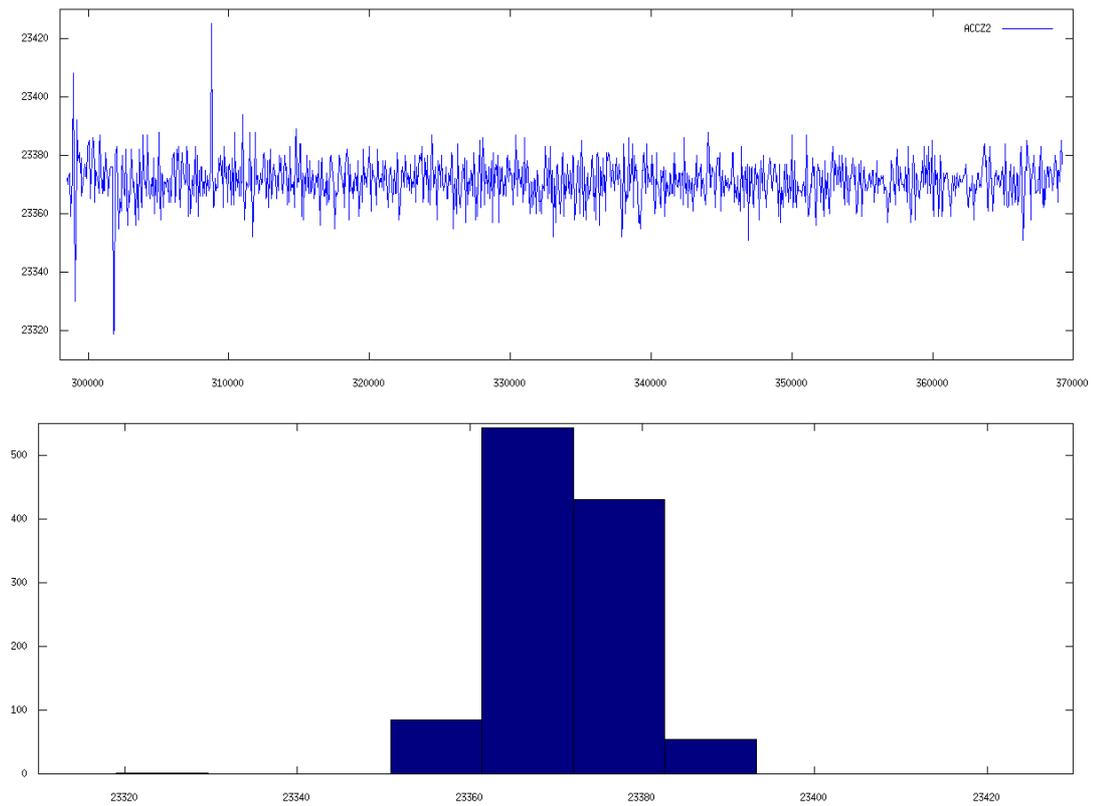


Bild 21.75: Grafische Darstellung der Kalibriermesswerte mit Histogramm, Z2-Achse -1g