Höhere Technische Bundeslehranstalt Wien 3, Rennweg
IT & Mechatronik

HTL Rennweg :: Rennweg 89b
A-1030 Wien :: Tel +43 1 24215-10 :: Fax DW 18

Diplomarbeit

Hovering Steward

ausgeführt an der
Höheren Abteilung für Informationstechnologie/Ausbildungsschwerpunkt
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2015/2016

durch

Christina Bornberg
Katharina Joksch
Markus Kaiser
Alexander Punz
Lucas Ullrich

unter der Anleitung von

Mag. Andreas Fink
DI Herbert Fleck

Wien, 11. Jänner 2016

1 Projektidee

Die Idee des Projektes, ist die Entwicklung eines innovativen Logistiksystems in der Gastronomiebranche, das durch Verwendung von Multicoptern umgesetzt wird. Optional wird ein Event namens 'Fluorescent Bakery' durchgeführt, bei dem das System zum Einsatz kommt.

Im Restaurant setzt sich jeder Gast auf seinen Platz. Anschließend können die Gäste auf Tablets, welche sich auf jedem Tisch befinden, Speisen und Getränke bestellen. Die Bestellungen werden an einen Server geschickt. Der Server schickt die Daten an ein, für Personal zugängliches Tablet und zeigt die Bestellungen und die dazugehörige Tischinformation an.

Der weitere Ablauf besteht darin, dass ein Kellner die bestellte Speise in die vorgefertigte Befestigung auf dem Multicopter stellt. Danach wählt er in einem Programm aus, um wessen Bestellung es sich handelt. Sobald der Kellner auf den „Bestellung ausliefern"-Button drückt, sendet der Server die Tischinformation an den Multicopter.

Der Multicopter fliegt autonom zum ausgewählten Tisch und landet auf einer Plattform, die er mithilfe eines geeigneten Positionierungssystems findet. Er fliegt dabei nur auf den vorgesehenen, für Besucher als gesperrt gekennzeichneten Wegen, was das Überfliegen von Menschen verhindert.

An der Plattform angekommen, legt der Multicopter die Speise ab und fliegt zurück zu seinem Stützpunkt (Base).

1.1 Ablauf

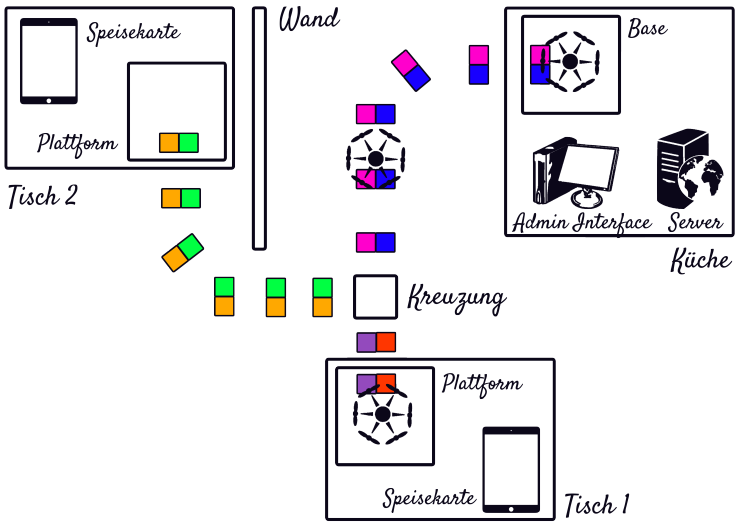


Abbildung 1.1: Der Weg einer Bestellung

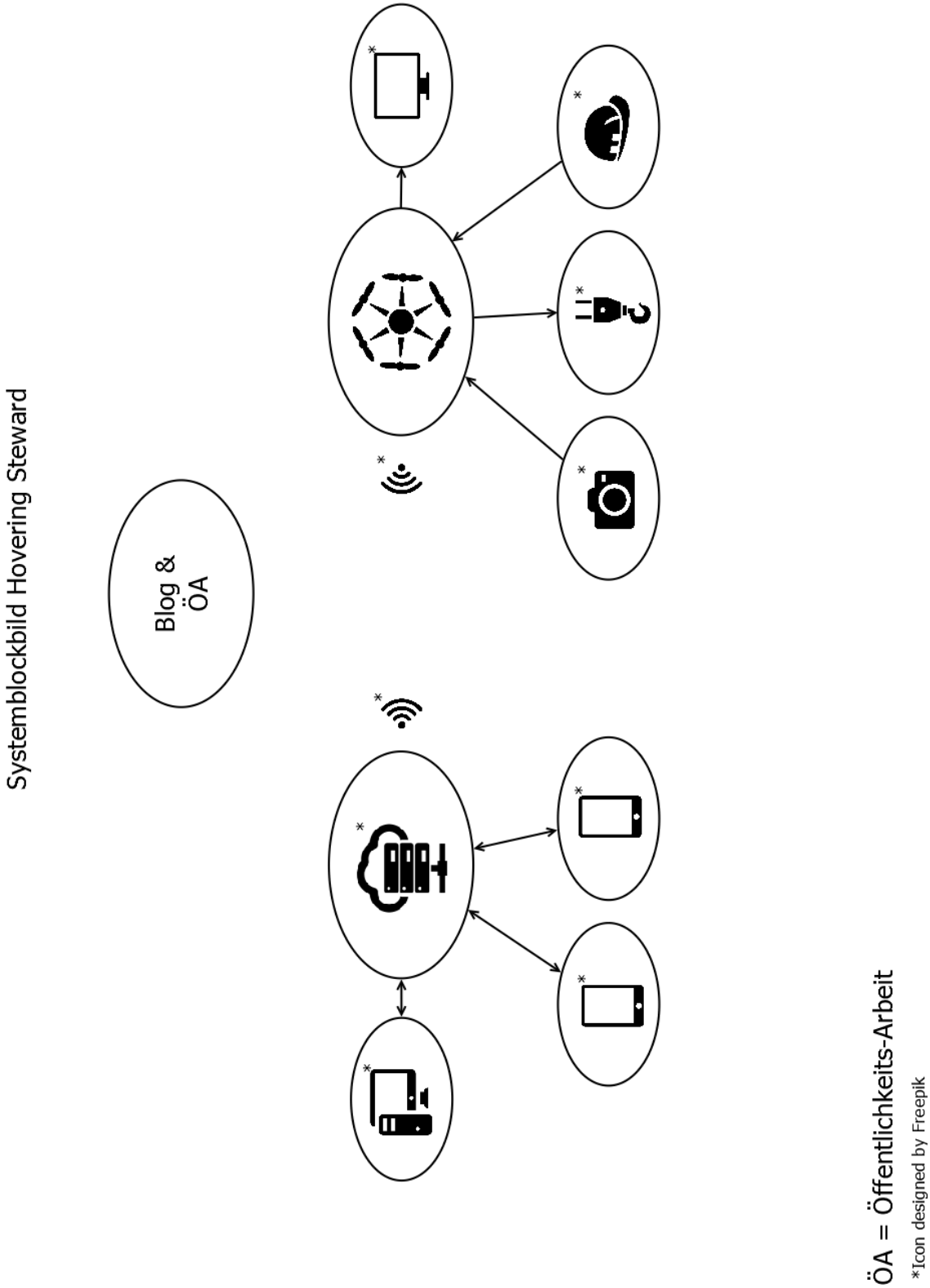


Abbildung 1.2: Das Zusammenwirken der Komponenten

2 Elektronik

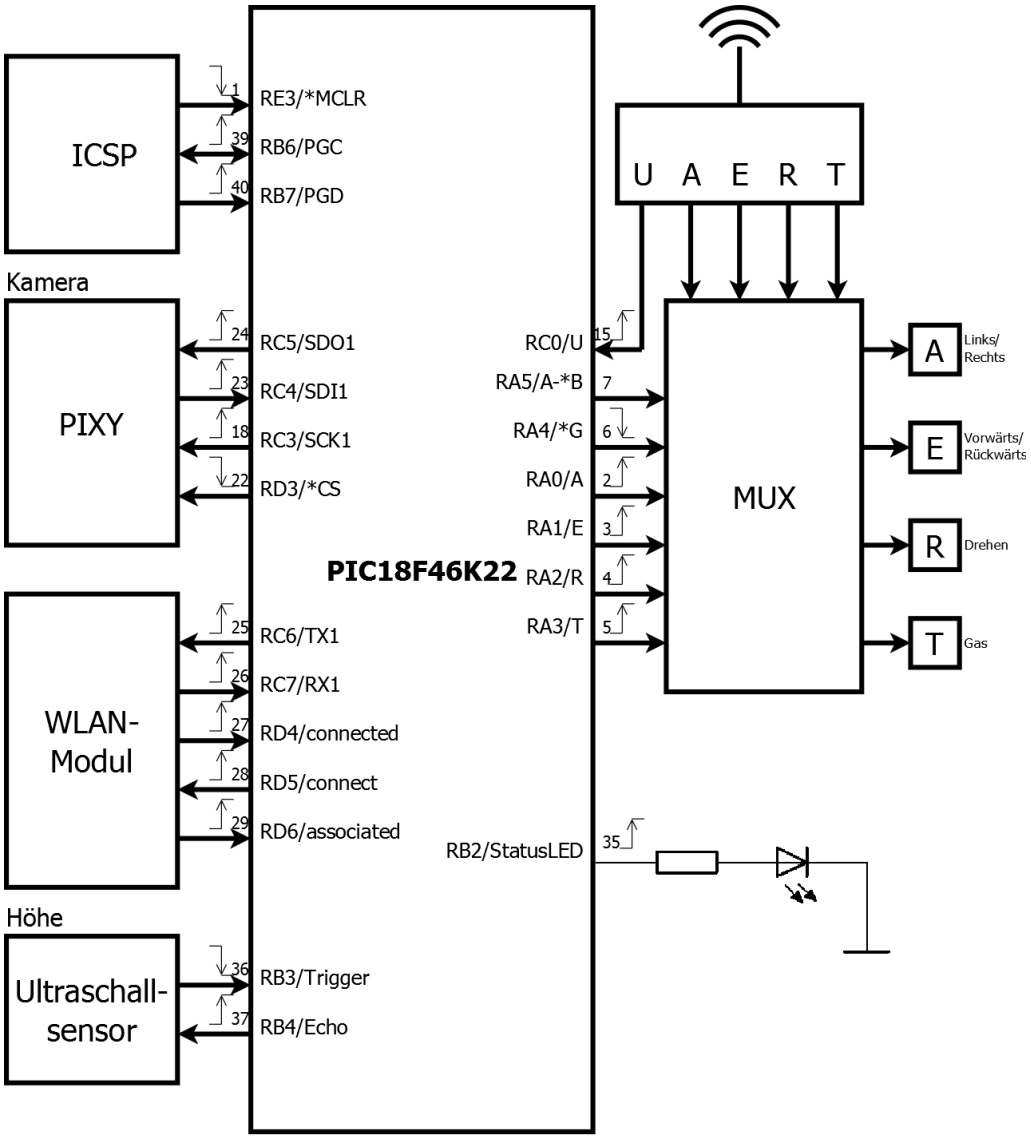


Abbildung 2.1: Blockschaltbilder der Hauptplatine

2.1 Sensoren

Um einen autonomen Flug zu realisieren sind unterschiedliche Sensoren notwendig. Durch die Sensoren muss es möglich sein die aktuelle Flugposition im Raum zu bestimmen und so den richtigen Weg zu finden.

2.1.1 Kamera

Aufgrund der Zuverlässigkeit einer Kamera im Vergleich zu Systemen die auf Distanzmessungen basieren wurde für die generelle Wegfindung das Kameramodul PIXY cmucam5 gewählt. Dieses gibt die notwendigen Informationen bereits fertig ausgewertet aus. Es wird ein Objektcode, dessen Position auf dem Bild sowie die Objektgröße wiedergegeben. Anhand dieser Informationen und einer vorher zugesendeten Route kann der Weg zum Ziel bestimmt werden.

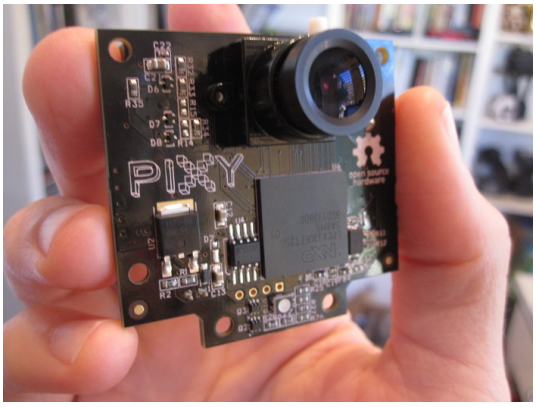


Abbildung 2.2: Kamera PIXY cmucam5

2.1.2 Ultraschallsensor

Da das Bestimmen der Flughöhe durch die Kamera zwar möglich ist, sich aber sehr aufwändig gestaltet, es müssen diverse Objektparameter bekannt sein und es besteht ein hoher Rechenaufwand, wird für die Messung der FLughöhe ein Ultraschallsensor verwendet. Dieser bietet die Möglichkeit die Flughöhe auf eine Distanz von bis zu 3 m zu bestimmen.

3 Firmware

3.1 Kamera

Das Kameramodul besitzt viele unterschiedliche serielle Schnitstellen. Aufgrund der schnellen Datenübertragung wurde das Serial Peripheral Interface, kurz SPI, gewählt.
Mit dieser Schnitstelle ist es möglich mehrere Teilnehmer unabhängig von einander anzusprechen, das eignet sich vorallem für Erweiterungen der Hardware.
Die PIXY cmucam5 verlangt bei der Übertragung der Daten über SPI Synchonistationsbytes. Durch diese ist es der Kamera möglich zu überprüfen ob eine korrekte Datenübertragung stattfindet. Um ein word, also 16 Bit, an Daten zu übertragen müssen insgesamt drei Bytes unterschieden werden:

- 0x5a, um Daten von der Pixy zu lesen, das nachfolgende Byte hat keine Bedeutung
- 0x5b, um Daten zu senden und zu lesen, das nachfolgende Byte enthält eine Bedeutung
- 0x0, wird nachfolgend zu 0x versendet, kein weiterer Nutzen

Die Auswertung der Bilder gestaltet sich aufgrund der vergleichsweise großen Datenmenge etwas aufwändiger. Von der Kamera werden 50 Bilder pro Sekunde erfasst und können somit auch übertragen werden. Jedes einzelne Bild besteht aus einem word (0xaa55) um das neue Bild zu indizieren und 8 weiteren words, diese sind:

- Farbcode oder Objekt
- Checksum
- Objektnummer
- X-Position
- Y-Position
- Breite
- Höhe
- Drehwinkel (Nur bei Farbcodes, sonst 0)

Für jedes Word muss hierbei ein neues Synchonisationsbyte gesendet werden.
Möchte man die Farbe der LED ändern, die mit der PIXY verbindbaren Servos ansteuern oder die Helligkeit der Kamera ändern so muss man 0x5b als Synchonisationsbyte senden, ansonsten 0x5a.

Abbildung 3.1: Ein einfacher Farbcode

3.1.1 Erkennen eines Bildes

Um ein Bild zu erkennen muss so lange nach der Startbedingung gesucht werden bis sie gefunden ist.

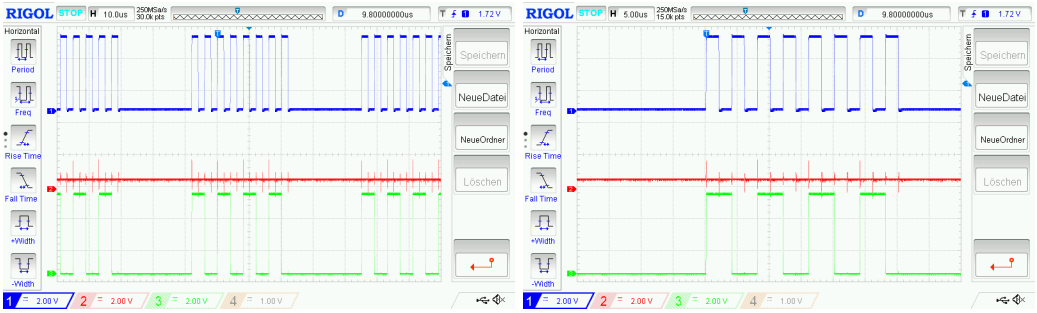
```
while(frame == 0) {
    w = ExchangeSpi2char(PIXY_SYNC, DUMMY);
    if(lw == PIXY_FRAME_OBJ && w == PIXY_FRAME_OBJ) {
        frame = 1;
        obj_type = 0;
    } else if(lw == PIXY_FRAME_OBJ && w == PIXY_COLORCODE) {
        frame = 1;
        obj_type = 1;
    } else if(w == 0 && lw == 0){
        frame = 0;
    }
    lw = w;
    c++;
    if(c > 254) {
        return 0;       // Kommentar
    }
}
```

Dieser Programmablauf versucht 254 mal die Startbedingung eines neuen Bildes zu erfassen. Diese wird zusammen mit dem nächsten word genutzt um festzulegen ob ein Farbcode oder ein normales Objekt erkannt wurde. Sobald ein Bild und Farbcode/Objekt erkannt wurde wird die Schleife verlassen und das abspeichern der übrigen Daten beginnt. Wird nach 254 Veruchen nichts erkannt wird die gesamte zuvor aufgerufene Funktion verlassen, das verhindert ein ewiges festhängen in der Funktion.

3.1.2 Überprüfen der SPI-Schnittstelle

Um sicher gehen zu können, dass die Schnittstelle auch korrekt funktioniert musste diese zuerst überprüft werden. Hierzu wird ein am Pin möglichst variierender Wert genommen (z. B. 01010101) und mit einem Oszilloskop der tatsächliche Ausgang auf der entsprechenden Leitung überprüft.

Anhand dieser Messungen zeigt sich, dass die Kamera eindeutig nicht das zurückliefert was zuvor gesendet wurde sondern eigene Daten. Zusätzlich sind eindeutige Störungen im Signal erkennbar.

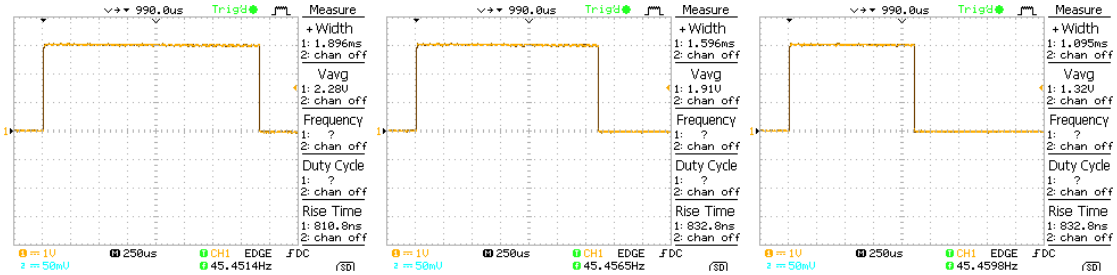


(a) Großer Zeitbereich (b) Kleiner Zeitbereich

Abbildung 3.2: Ausgang der SPI-Schnittstelle

3.1.3 Bestimmen des Flugmodus

Um im Notfall manuell in das Geschehen eingreifen zu können soll der Hexacopter wahlweise Manuell oder automatisch gesteuert werden. Der Flugmodus wird mit einem Schalter an der Fernsteuerung bestimmt. Dieser veranlasst den Empfänger dazu einen Puls mit einer Länge zwischen 1095 µs und 1896 µs auszugeben. Das Signal ist alle 20 ms periodisch wiederkehrend und besitzt 3 Signalwerte:



(a) Mode0 (b) Mode1 (c) Mode2

Abbildung 3.3: Signale des U-Pins

Den unterschiedlichen Signalen werden unterschiedliche Flugmodi zugeordnet. So entsprechen die zwei kürzeren Impulse (1095 µs und 1596 µs) einem manuellen Flug und der lange Impuls (1896 µs) einem autonomen Flug. Der Impuls wird am Micro Controller durch einen 16-Bit Timer und dessen Gate-Anschluss ausgewertet. Die Signale um den Hexacopter zu steuern werden durch einen Multiplexer zu dem Flightcontroller geleitet, abhängig vom Flugmodus werden die Steuerbefehle des 2,4 GHz Empfängers oder jene des Micro Controllers genutzt.

```
unsigned int Fkt_CalcTime(void) {
    unsigned int time_pulse = TMR3H;
    time_pulse <<= 8;
    time_pulse |= TMR3L;
    TMR3H = 0;
    TMR3L = 0;
    NOP();
    return time_pulse;
```

```
9  }

11 bit Fkt_ModeCheck(void) {
       unsigned int time = Fkt_CalcTime();
13     if(time < GEAR_TIME){
           LED = 0;
15         return 0;
       }else if(time >= GEAR_TIME) {
17         LED = 1;
           return 1;
19     }
       NOP();
21 }
```

4 Mechanik

4.1 Rotorschutz

In dem gewählten Bausatz des Hexacopters ist kein Rotorschutz beinhaltet, eines der Zeile ist es den Rotor möglichst gut zu schützen und das maximale Abfluggewicht nicht zu überschreiten. Da der Hexacopter mit einer maximalen Last von 1,9 kg belastet werden kann, können Materialen wie Stahl oder Aluminium für den Rotorschutz nicht verwendet werden. Da die Möglichkeit besteht, Teile in einer vernünftigen Qualität vor Ort drucken zu können, wurde die Fertigung mittels des 3D-Druckers gewählt. Mit diesem ist es möglich den erforderlichen Rotorschutz sehr genau und leicht drucken zu können. Die folgende Abbildung zeigt den fertigen Rotorschutz.

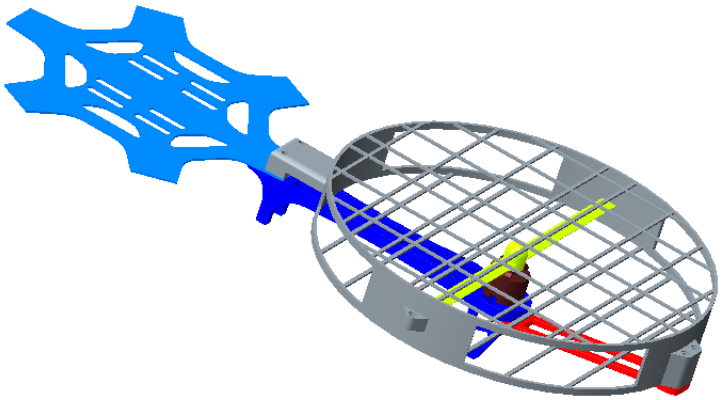


Abbildung 4.1: Rotorschutz systematisch dargestellt

Der Rotorschutz umrandet den Propeller, damit dieser im Falle eines Absturzes nicht zerstört wird. Die Lamellen ober- und unterhalb des Rotors sollen Personen gegen Verletzungen schützen. Der Ring wird durch die Strebe (rot) gestützt, damit er nicht nach unten abbrechen kann. Durch die großen Abmessungen bzw. der Form, kann der Ring nicht als ein Teil gedruckt werden, er wird dadurch in verschiedene Sektoren unterteilt. Der Ring wird in eine untere und obere Hälfte unterteilt bzw. beide Hälften wieder in zwei Teile. Die einzelnen Glieder werden dann mit Schrauben und Zweikomponentenkleber zusammengefügt.

4.2 Platinen

Um das Projekt umsetzen zu können, ist es erforderlich einerseits den Mikroprozessor mit den Sensoren zu verbinden und andererseits die Kommunikation zwischen der Speisekarten-

App und dem Hexacopter zu können. Damit diese Funktionen erfüllt werden können, werden zwei Platinen verwendet.

Das wichtigsten Bauteilgruppen der Kommunikationsplatine sind das WLAN-Modul, dieses ermöglicht die Kommunikation zwischen der App und dem Hexacopter und der Spannungsregler. Der Spannungsregler sorgt dafür, dass das WLAN-Modul mit 3.3 V versorgt wird und der Rest der Platine mit 5 V. Die Hauptplatine beinhaltet ebenso mehrere Bereiche: Den Multiplexer, den Mikroprozessor und die Steckverbindungen für die einzelnen Sensoren. Der Multiplexer wechslet zwischen der Steuerung der Flugmodi durch die Fernbedienung oder dem Mikroprozessor.

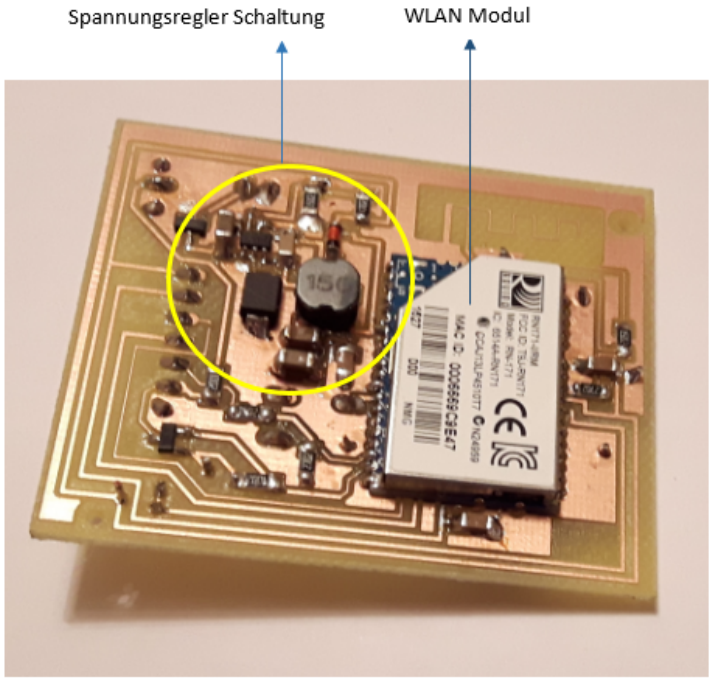


Abbildung 4.2: Kommunikationsplatine Unterseite

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed

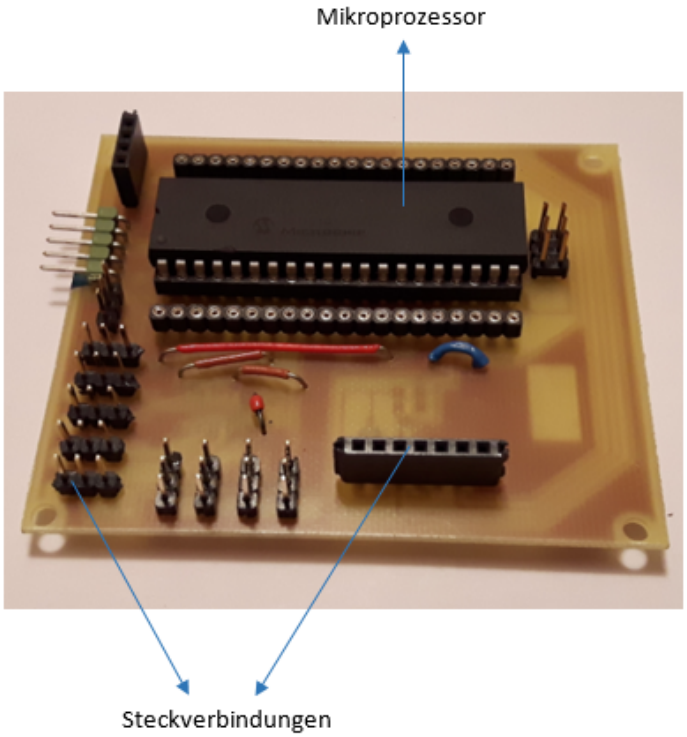


Abbildung 4.3: Sensorplatine Oberseite

elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

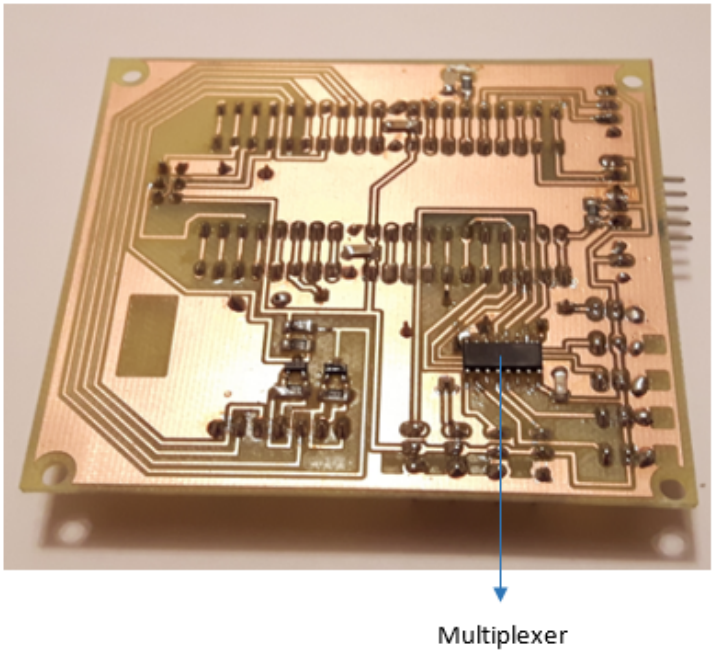


Abbildung 4.4: Sensorplatine Unterseite