



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# **Bachelorarbeit**

**Mehdi Kaf**

**Zuverlässige Personenerkennung durch Energieeffiziente und verteilte Sensorik**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Mehdi Kaf

**Zuverlässige Personenerkennung durch Energieeffiziente und verteilte Sensorik**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stephan Pareigis  
Zweitgutachter: Prof. Dr. Wolfgang Fohl

Eingereicht am: 17. September 2016

**Mehdi Kaf**

**Thema der Arbeit**

Zuverlässige Personenerkennung durch Energieeffiziente und verteilte Sensorik

**Stichworte**

Radar, Personenerkennung, ARM Mikrocontroller, digitale Signalverarbeitung, FFT, Platinenlayout

**Kurzzusammenfassung**

Gegenstand dieser Arbeit ist die Entwicklung eines Stromsparsames Intelligentes Sensorsystems zur Umgebungsüberwachung

**Mehdi Kaf**

**Title of the paper**

Reliable Human Detection through energy-efficient and distributed sensor technology

**Keywords**

Radar transceiver, Motiondetection, ARM microcontroller, digital signal processing, FFT, Board design

**Abstract**

This document is about developing an intelligent and energy-efficient Sensor system for surveillance

# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Einführung</b>   | <b>1</b>  |
| 1.1      | Ziel dieser Arbeit . . . . .                                    | 1         |
| 1.2      | Gliederung . . . . .  | 1         |
| <b>2</b> | <b>Technische Grundlagen</b>                                    | <b>2</b>  |
| 2.1      | Sensorik für Überwachung . . . . .                              | 2         |
| 2.1.1    | Passiv Infrarot Sensor . . . . .                                | 2         |
| 2.1.2    | Ultraschall . . . . .   | 2         |
| 2.1.3    | Radar . . . . .   | 3         |
| 2.1.4    | Pulsradar . . . . .   | 4         |
| 2.1.5    | Dauerstrich Radar . . . . .                                     | 5         |
| 2.1.6    | Frequency modulated continuous wave radar . . . . .             | 7         |
| 2.2      | Sensorenvergleich . . . . .                                     | 7         |
| 2.3      | Zusammenfassung und Auswahl des benötigten Radartypen . . . . . | 10        |
| <b>3</b> | <b>Technische Anforderungen und Analyse</b>                     | <b>11</b> |
| 3.1      | Anforderung an das Sensorsystem . . . . .                       | 11        |
| 3.2      | Signalverarbeitungskette . . . . .                              | 12        |
| 3.3      | Mikrokontroller . . . . .                                       | 12        |
| 3.4      | Abtastung . . . . .   | 13        |
| 3.5      | Verstärkung . . . . .   | 13        |
| 3.6      | Auswahl der Radarsensoren . . . . .                             | 16        |
| 3.6.1    | IPS165 CW Doppler-Sensor . . . . .                              | 16        |
| 3.6.2    | K-MC1_LP low Power Radarsensor . . . . .                        | 17        |
| 3.7      | Bewegungserkennung . . . . .                                    | 19        |
| 3.7.1    | Schmitt Trigger . . . . .                                       | 19        |
| 3.7.2    | Software Basierte Erkennung . . . . .                           | 20        |
| <b>4</b> | <b>Design</b>   | <b>22</b> |
| 4.1      | Übersicht . . . . .   | 22        |
| 4.2      | Hardware Komponenten . . . . .                                  | 22        |
| 4.2.1    | MCU . . . . .   | 22        |
| 4.2.2    | Verstärker . . . . .  | 23        |
| 4.2.3    | Ozillator . . . . .   | 23        |
| 4.2.4    | Stromversorgung . . . . .                                       | 23        |
| 4.2.5    | Sensor Energie-Management . . . . .                             | 24        |

|          |   |           |
|----------|---|-----------|
| 4.2.6    | Reichweitenerhöhung . . . . .                 | 24        |
| 4.2.7    | Schnittstellen . . . . .                      | 25        |
| 4.2.8    | Platinenlayout . . . . .                      | 28        |
| 4.3      | Entwicklungsumgebung . . . . .                | 32        |
| 4.3.1    | PC Software . . . . .                         | 32        |
| 4.3.2    | Firmware . . . . .                            | 32        |
| 4.3.3    | Telemetrie . . . . .                          | 34        |
| <b>5</b> | <b>Realisierung</b>                           | <b>35</b> |
| 5.1      | Platine . . . . .                             | 35        |
| 5.2      | Firmware . . . . .                            | 36        |
| 5.3      | PC Verbindung . . . . .                       | 38        |
| 5.4      | Fast Fourier Transformation . . . . .         | 38        |
| 5.5      | Fenster Funktion . . . . .                    | 38        |
| 5.6      | Bewegungserkennung . . . . .                  | 39        |
| 5.7      | Python Telemetrie . . . . .                   | 41        |
| <b>6</b> | <b>Tests</b>                                  | <b>44</b> |
| 6.1      | MCU Auslastung . . . . .                      | 44        |
| 6.2      | Stromverbrauch . . . . .                      | 44        |
| 6.3      | Reichweitentest . . . . .                     | 47        |
| <b>7</b> | <b>Zusammenfassung</b>                        | <b>52</b> |
| 7.1      | Gesammelte Erfahrungen . . . . .              | 52        |
| 7.2      | Evaluierung des entwicklten Systems . . . . . | 52        |

# Tabellenverzeichnis

|     |  |    |
|-----|--|----|
| 2.1 | Sensorenvergleich Quelle: <b>Weidmann</b> . . . . .  | 9  |
| 3.1 | Auszug aus dem Datenblatt des Sensors . . . . .      | 18 |
| 4.1 | Übersicht der Eigenschaften des STM32F07C8 . . . . . | 23 |
| 4.2 | Stromverbrauch der Hardwarekomponenten . . . . .     | 23 |
| 4.3 | Felder eines CAN Frames . . . . .                    | 27 |
| 6.1 | Auslastung des ARMs . . . . .                        | 44 |
| 6.2 | Reichweitentest . . . . .                            | 49 |

# Abbildungsverzeichnis

|      |   |    |
|------|---|----|
| 2.1  | Pirsensor Quelle: ST . . . . .  | 3  |
| 2.2  | Ultraschall . . . . .   | 3  |
| 2.3  | Monostatic-Radar . . . . .  | 4  |
| 2.4  | Doppler: Quelle Wikipedia . . . . .   | 6  |
| 2.5  | Die Sägezahnmodulation Quelle: Weidmann . . . . .   | 8  |
| 2.6  | Dreiecksmodulation Quelle: Weidmann . . . . .   | 8  |
| 2.7  | Zwei Frequenz Modulation Quelle: Weidmann . . . . .   | 9  |
| 3.1  | übersicht des Gesamtsystems . . . . .   | 11 |
| 3.2  | Signalverarbeitungskette . . . . .  | 12 |
| 3.3  | Der Pegel des unverstärkten Signals des IPM165 Sensors in Rot liegt bei 80mV,<br>das verstärkte Signal in Blau bei 1.8 V . . . . .                            | 14 |
| 3.4  | Vorgeschlagene Verstärkerschaltung der Firma Rfbeam . . . . .   | 14 |
| 3.5  | Bandpass: bestehend aus einem Hochpass (R1, C1) und Tiefpass (R2, C2) . . . .   | 15 |
| 3.6  | Spannungsteiler: Durch die zwei gleichen Widerstände R5 und R6 wird die<br>Versorgungsspannung halbiert . . . . .   | 16 |
| 3.7  | IPM 165 . . . . .   | 17 |
| 3.8  | K_MC1_LP . . . . .  | 18 |
| 3.9  | Ein invertierender Schmitt Trigger . . . . .  | 19 |
| 3.10 | Schmitt Trigger: in Rot ist das Oszilloskop Bild des IPM 165 Sensors zu sehen,<br>in Blau die Ausgabe des Schmitt Triggers . . . . .                          | 20 |
| 4.1  | Übersicht des Gesamtsystems . . . . .   | 22 |
| 4.2  | Transistorschaltung zum Pulsen der Stromversorgung des Radarsensors . . . .   | 24 |
| 4.3  | Oben: Aufbau eines Can Frames, Unten: Die Zustände des CAN bus . . . . .  | 26 |
| 4.4  | Die eagle Schaltung zeigt die Verbindung zwischen den CAN_Rx und CAN_Rx<br>pins des $\mu C$ 's dem CAN Transceiver (mitte) und dem Verbindungstecker (Rechts) | 27 |
| 4.5  | Uart frame . . . . .  | 28 |
| 4.6  | Die Eagle Teilschaltung zeigt die Verbindung zwischen dem Mikrocontroller<br>und dem SWD Programmierstecker . . . . .   | 29 |
| 4.7  | Schaltplan Der Mikrocontrollerplatine . . . . .   | 30 |
| 4.8  | verstärkerschaltung . . . . .   | 31 |
| 4.9  | Platinenlayout . . . . .  | 32 |
| 4.10 | DualInterleaved . . . . .   | 33 |
| 4.11 | ADC injected Mode . . . . .   | 34 |
| 5.1  | Unbestückte Platine . . . . .   | 36 |

|     |  |    |
|-----|--|----|
| 5.2 | Flussdiagramm . . . . .  | 37 |
| 5.3 | Hanningwindow Quelle: Wikipedia . . . . .  | 39 |
| 5.4 | Cfar: Die Guard-Zellen werden bei der Mittelwertbildung nicht einbezogen . .   | 40 |
| 5.5 | Telemetrie Anwendung mit angeschlossenem Stereo Radar Sensor . . . . .   | 41 |
| 5.6 | Flussdiagramm der Telemetrie Anwendung . . . . .   | 42 |
| 6.1 | Verbrauch des IPM 165 . . . . .  | 45 |
| 6.2 | Verbrauch des KCM1_LP . . . . .  | 46 |
| 6.3 | Testgelände . . . . .  | 47 |
| 6.4 | testhardware: Links der Radarsensor, in der Mitte befindet sich die enwickelte<br>Platine, Rechts unten Die AdHoc Platine und oben der Segger J-Link . . . . . | 48 |
| 6.5 | Radarsignal in einem 15 m Abstand . . . . .  | 49 |
| 6.6 | Fourier Transform und CFAR schwelle bei 15 m . . . . .   | 50 |
| 6.7 | Radarsignal in einem 30 m Abstand . . . . .  | 50 |
| 6.8 | Fourier Transform und CFAR Schwelle bei 30 m . . . . .   | 51 |

# Listings

|     |  |    |
|-----|--|----|
| 5.1 | Kommandos der Telemetrie Anwendung . . . . .                   | 38 |
| 5.2 | Dequevlist . . . . .   | 42 |
| 5.3 | Struktur für die Einstellung der Parameter des ADC's . . . . . | 43 |

# 1 Einführung

Die folgende Bachelorarbeit entstand bei der Firma AdHoc Networks GmbH. Ein aktuelles Projekt beschäftigt sich mit der Entwicklung und Herstellung eines netzwerkbasierten Überwachungssystems. Das System besteht aus mehreren miteinander vernetzten Sensorknoten, die zusammen die Überwachung eines Bereiches ermöglichen.

## 1.1 Ziel dieser Arbeit

Im Laufe dieses Projektes wird ein Sensorsystem entwickelt, das ein Betreten des vom Sensornetzwerk überwachten Bereiches zuverlässig erkennt. Zusätzlich soll, im Falle einer Bewegungserkennung, das bestehende Überwachungssystem über ein Kommunikationsinterface benachrichtigt werden. Dabei sollen geeignete Sensoren und Auswertungsmethoden eingesetzt werden, die eine möglichst hohe Reichweite und einen niedrigen Stromverbrauch erzielen.

## 1.2 Gliederung

Diese Arbeit teilt sich in sieben Kapitel. Nach einer kurzen Einführung und Darstellung der formalen Anforderungen in Kapitel 1, werden in Kapitel 2 [Technische Grundlagen] die gängigen Technologien, die in der Überwachungssensorik zum Einsatz kommen verglichen. Hierbei wird ein tieferer Einstieg in die Radartechnologie vermittelt, da diese für den weiteren Verlauf dieser Arbeit benötigt wird. In Kapitel 3 [Technische Anforderungen und Analyse] wird eine Analyse der technischen Anforderungen durchgeführt, die für den Entwurf und die Entwicklung des Sensorsystems benötigt werden. Darauf folgend befasst sich das Kapitel 4 mit dem Design der Hardware und Softwarekomponenten. In Kapitel 5 wird die konkrete Realisierung der Hardware und Softwarekomponenten durchgeführt. Abschließend werden in Kapitel 6 und 7 Tests durchgeführt und analysiert, die gesammelten Erfahrungen wiedergegeben sowie ein Ausblick in die Zukunft gegeben.

## 2 Technische Grundlagen

### 2.1 Sensorik für Überwachung

#### 2.1.1 Passiv Infrarot Sensor

Ein Passiv Infrarot Sensor (PIR) detektiert Licht mit einer bestimmten Wellenlänge im Infrarotspektrum. Die PIR Sensortechnik macht sich die physikalische Eigenschaft zu Nutze, dass jedes Objekt Wärme als Infrarot Energie ausstrahlt. Ein PIR Sensor besteht aus zwei für Infrarot empfindliche pyroelektrische Sensoren, die zwei verschiedene Detektionsbereiche bilden, siehe Abbildung 2.1. Damit eine Bewegung erkannt wird, müssen zwei unterschiedliche Lichtintensitäten an den Detektionsbereichen gemessen werden. Beim Eintreten einer Person in den Detektionsbereich des Sensors ändert sich die vom Sensor aufgenommene Temperatur. Bei einer Bewegungserkennung kann dann am Ausgang des Sensors eine kleine AC Spannung gemessen werden. Die meisten handelsübliche PIR Sensoren werden vom Hersteller mit einer Auswertungslogik ausgestattet, um Information aus diesen Spannungen zu gewinnen. Falsche Detektion können bei einer Änderung der Lichtverhältnisse, wie an heißen Tagen oder bei einer Platzierung in der Nähe einer Wärmequelle auftreten. Die Reichweite liegt in der Regel in einem Bereich von 3 bis 10 m. Aus diesen Gründen lässt sich schließen, dass PIR Sensoren für die Umgebungsüberwachung nur bedingt für dieses Projekt einsetzbar.

#### 2.1.2 Ultraschall

Ein Ultraschall Sensor besteht aus einem Sende- und einem Empfangsmodul. Das Sendemodul strahlt Ultraschallwellen aus, die am Zielobjekt reflektiert werden. Die Reflektion wird vom Empfangsmodul empfangen. Anhand der Latenzzeit zwischen dem Sende- und Empfangszeitpunkt, kann die Entfernung zum Zielobjekt bestimmt werden. Diese Sensoren sind preisgünstig und zeichnen sich durch eine hohe Empfindlichkeit aus. Ultraschallsensoren werden aufgrund ihrer niedrigen Reichweite von 0.2 bis 3 Metern (vgl. [MURATA](#)) zumeist als Näherungssensoren eingesetzt.

Ultraschallwellen werden von Materialenoberflächen wie z.B. Schaumstoff oder auch Textil absorbiert. Zudem können bestimmte Geräuschequellen in der Nähe des Sensors zu falschen

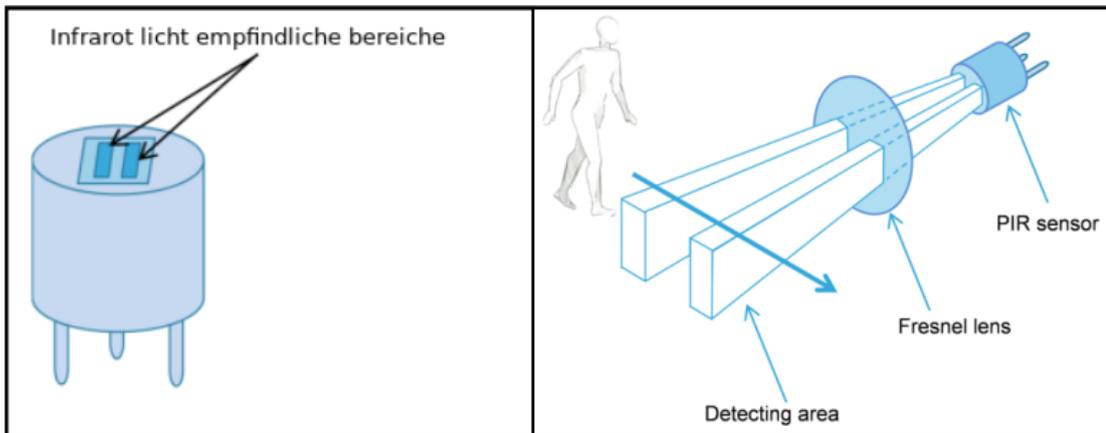


Abbildung 2.1: Pirsensor Quelle: ST

Ergebnissen führen. Zusammenfassend lässt sich sagen, dass Ultraschallsensoren aufgrund der geringen Reichweite und der Störanfälligkeit für die Umgebungsüberwachung ungeeignet sind.

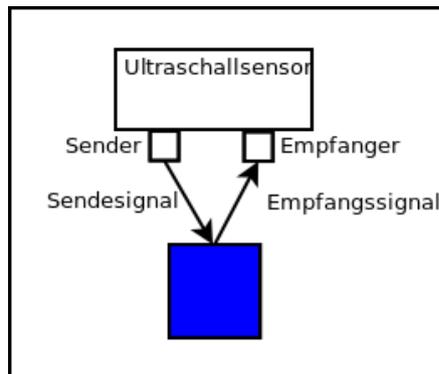


Abbildung 2.2: Ultraschall

### 2.1.3 Radar

Die Radar Technologie basiert auf der, vom deutschen Forscher Heinrich Hertz bewiesenen Eigenschaft, dass sich elektromagnetische Wellen an metallischen und dielektrischen Körpern reflektieren. Das Prinzip nachdem ein Radar funktioniert, kann wie folgt erklärt werden. Von einer Antenne aus werden elektromagnetische Wellen ausgestrahlt, befindet sich ein Objekt in deren Strahlungsrichtung, wird ein Teil der ausgesendeten Leistung reflektiert. Dieses

Echosignal wird von der Empfangsantenne erfasst. Durch die Messung der Laufzeit des Signals und die Kenntnis der Ausbreitungsgeschwindigkeit der Radarwellen kann die Entfernung des Ziels berechnet werden. Mit Hilfe der Radartechnologie können Entfernung- und die Geschwindigkeitsinformationen eines Ziels bestimmt werden.

Im folgenden Abschnitt werden Bauarten und Unterarten des Radars vorgestellt.

### Monostatisches und Bistatisches Radar

Bei einem Monostatischen Radar links auf Bild 2.3 zu sehen, befinden sich die Sende- und Empfangsantennen am gleichen Ort (meist eine Antenne für das Senden und Empfangen). Bei einem Bistatischen Radar rechts im Bild 2.3 sind Sende- und Empfangsantenne an verschiedenen Stellen positioniert. In diesem Fall wird eine Kommunikationsverbindung zwischen den Sende- und Empfangsstationen erforderlich, um die gewonnenen Daten auszutauschen und die Sende- und Empfangssignale zu synchronisieren. (siehe. Goebel (2003) (S. 22)) Die meisten im Handel erhältlichen Radarsensoren sind monostatisch (siehe Bild 2.3).

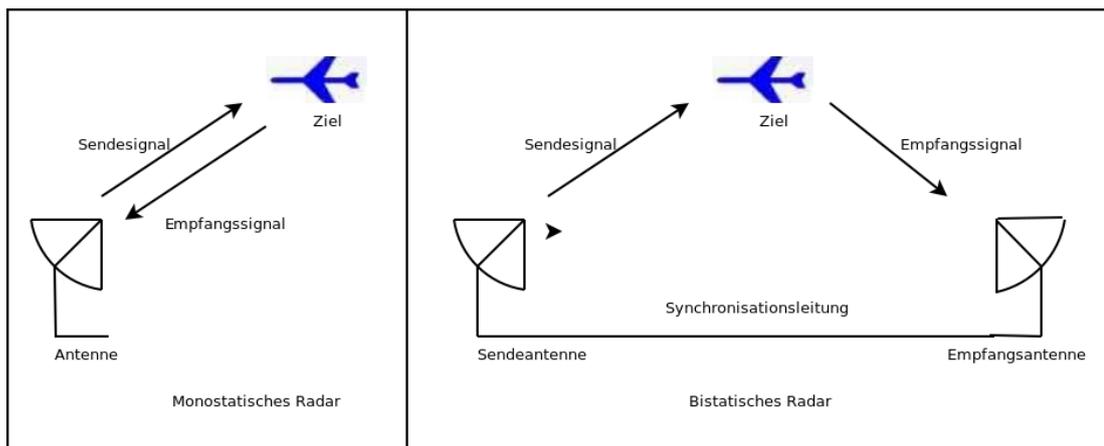


Abbildung 2.3: Monostatic-Radar

### 2.1.4 Pulsradar

Bei einem Pulsradar wird ein Signal bestehend aus einer Folge hochfrequenter Pulse gesendet. Ähnlich wie bei einem Sonar, kann aus dem zeitverzögerten Empfangssignal auf die Entfernung des Zieles geschlossen werden (vgl. Goebel (2003) (S. 115)). Die Laufzeit  $\tau$  ist wie folgt definiert:

$$\tau = \frac{2R}{c_o} \quad (2.1)$$

Die Konstante  $c_0$  stellt die Geschwindigkeit der elektromagnetischen Wellen dar, und  $R$  die Distanz zwischen dem Radar und seinem Ziel. Durch eine Umformung der Gleichung kann die Entfernung  $R$  ermittelt werden:

$$R = \frac{\tau * c_0}{2} \quad (2.2)$$

### 2.1.5 Dauerstrich Radar

Bei dem Dauerstrichradar, auch CW Radar genannt, wird ein kontinuierliches Signal konstanter Frequenz und Amplitude gesendet. Das Sendesignal mit der Amplitude  $A_s$ , der Kreisfrequenz des Sendesignals  $\omega_s$  und der Phasenverschiebung  $\varphi_s$  lässt sich wie folgt beschreiben:

$$S_s(t) = A_s \sin(\omega_s t + \varphi_s) \quad (2.3)$$

Das Empfangssignal beinhaltet zudem die Laufzeit  $\tau$ :

$$S_e(t) = A_s \sin(\omega_s(t - \tau) + \varphi_s) \quad (2.4)$$

Aus der Messung der Phasendifferenz  $\varphi$  kann die Entfernung  $R$  berechnet werden:

$$R = \frac{c\tau}{2} = \frac{c \cdot \Delta\varphi}{2\omega_s} \quad (2.5)$$

Da die Phasendifferenz  $\Delta\varphi$  eine periodische Funktion mit einer Periode von  $2\pi$  ist, kann sie nur in einem Bereich von  $-\pi \leq \Delta\varphi \leq \pi$  bzw.  $0 \leq \Delta\varphi \leq 2\pi$  eindeutige Entfernungsinformation  $R_{inst}$  liefern. Die maximal messbare, eindeutige Entfernung kann durch das Setzen der maximal möglichen Phasendifferenz von  $2 * \pi$  in die obige Formel berechnet werden:

$$R_{inst} = \frac{\pi c}{\omega_s}$$

Als Beispiel kann bei einer Frequenz von 1 GHz nur ein eindeutiger Entfernungsbereich von 15 cm gemessen werden (vgl. [Goebel \(2003\)](#) (S. 167-170)).

Um eine Variante des CW Radars den CW Doppler-Radar vorzustellen, muss zunächst der Doppler-Effekt erklärt werden.

### Doppler-Effekt

Bei einem Radar tritt der Doppler-Effekt ein, wenn sich Sender und/oder Empfänger relativ zueinander bewegen. So ist eine Frequenzverschiebung zwischen dem Sende- und Empfangssi-

gnal zu beobachten. Dieser physikalische Effekt wurde erstmals 1842 von dem Österreicher Christian Doppler beschrieben. Die Empfangsfrequenz ist wie folgt definiert:

$$f_e = f_s \frac{\sqrt{1 - \beta^2}}{1 - \beta \cos \alpha} \quad (2.6)$$

$$f_d = f_e - f_s = 2f_s \frac{v}{c} \cos \alpha \quad (2.7)$$

Mit  $\beta = \frac{v}{c}$ ,  $f_s$  die Sendefrequenz,  $v$ , relative Geschwindigkeit und  $c$  die Lichtgeschwindigkeit. Die Doppler-Frequenz  $f_d$  ist die Differenz zwischen der Sende- und Empfangsfrequenz. Die Formel für die Doppler-Frequenz wird aus der Formel 2.6 hergeleitet. Dieser Effekt wird in der Radartechnik genutzt, um die relative Geschwindigkeit zwischen Sender und Empfänger zu ermitteln. Bewegt sich der Empfänger auf den Sender zu ist die Doppler-Frequenz positiv. Entfernt sich dieser vom Sender ist sie negativ. Dieses Phänomen ist auch in der Akustik zu beobachten. Ein Beispiel dafür ist die Sirene eines Krankenwagens. Nähert sich der Wagen einer, am Strassenrand stehenden Person, hört diese einen höheren Ton. Die Frequenz des Martinshorn bleibt zwar konstant, allerdings verringert sich die Entfernung zum Empfänger und dadurch die Laufzeit, was dazu führt, dass die Schallwellen immer schneller beim Empfänger ankommen.

Die Abbildung 2.4 veranschaulicht dieses Phänomen. Bei einem CW Doppler Radar wird anstatt

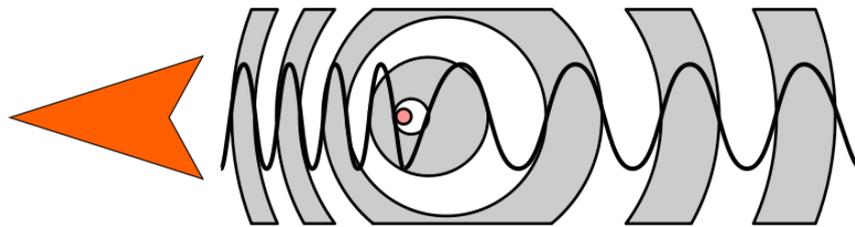


Abbildung 2.4: Doppler: Quelle Wikipedia

der Phasenverschiebung die Doppler Frequenz ausgewertet, dadurch wird die Messung der Geschwindigkeit  $v$  des Zieles mit Hilfe der Formel 2.8 ermöglicht.

$$v = \frac{f_D \cdot c}{2f_s \cos \alpha} \quad (2.8)$$

Mit  $f_D$  die Doppler-Frequenz,  $f_s$  die Sendefrequenz und  $\alpha$  der Winkel zwischen der Bewegungsrichtung des Zieles und die Aufstellrichtung des Sensors.

Die Doppler-Frequenz kann an einem fiktiven Beispiel mit der Formel 2.8 berechnet werden: bewegt sich eine Person mit 5 km/h direkt auf einen CW Doppler Radar Sensor zu, ergibt sich eine Doppler-Frequenz von 220 Hz bei einer Sendefrequenz von 24.125 GHz. Die meisten Dopplersensoren für die Verkehrsüberwachung senden mit einer Frequenz von 24 bis 24,250 GHz.

$$f_D = 2 * \frac{f_s \cdot v \cdot \cos(0)}{c} = \frac{2 * 24,125 \cdot 10^6 \text{ Hz} \cdot 5 \text{ km/h}}{1079252848,8 \text{ km/h}} = 223.5 \text{ Hz} \quad (2.9)$$

### 2.1.6 Frequency modulated continuous wave radar

Das Frequency modulated continuous wave radar (FMCW) erweitert das CW-Radar um einen Frequenzmodulationseingang. So kann die Sendefrequenz geändert werden. Durch die Erhöhung der Modulationdauer der Sendefrequenz kann eine höhere Eindeutigkeit bei der Entfernungsmessung erreicht werden. Ein höherer Frequenzhub der Sendefrequenz  $\Delta f_s$  führt zu einer genaueren Entfernungsmessung. Wobei  $\Delta f = f_{smax} - f_{smin}$  und  $f_s$  die Sendefrequenz sind. Die drei wichtigsten Arten der Frequenzmodulation sind die lineare Sägezahnmodulation in Bild 2.5, die Dreiecksmodulation in Bild 2.6 und das Zweifrequenz-Verfahren in Bild 2.7. Die Sägezahnmodulation ermöglicht die reine Entfernungsmessung, während mit der Zweifrequenzmodulation und der Dreiecksmodulation, eine gleichzeitige Entfernungs- und Geschwindigkeitsmessung möglich ist. Die Entfernung bei einer Dreiecksmodulation, unter der Annahme, dass das Ziel sich in einem Winkel von 0 Grad zu dem Empfänger bewegt, lässt sich mit der nachfolgenden Formel berechnen: (siehe. Weidmann (S. 167))

$$R = \frac{c_0 \cdot f_b \cdot T_m}{4f_M} \quad (2.10)$$

## 2.2 Sensorenvergleich

Vergleicht man die in der Tabelle aufgelisteten Sensoren, PIR, Ultraschall und Radar, lässt sich feststellen, dass PIR und Ultraschall zwar preisgünstig in der Anschaffung sind, aber eine begrenzte Reichweite aufweisen und stör anfälliger sind. Der Radar hingegen zeichnet sich durch detaillierte Geschwindigkeits- und Entfernungsinformationen aus, ist allwettertauglich und durchstrahlt sogar Kunststoffe. Trotz der mittleren Anschaffungskosten ist der Radar durch seine Vorteile der geeigneter Sensor im Vergleich.

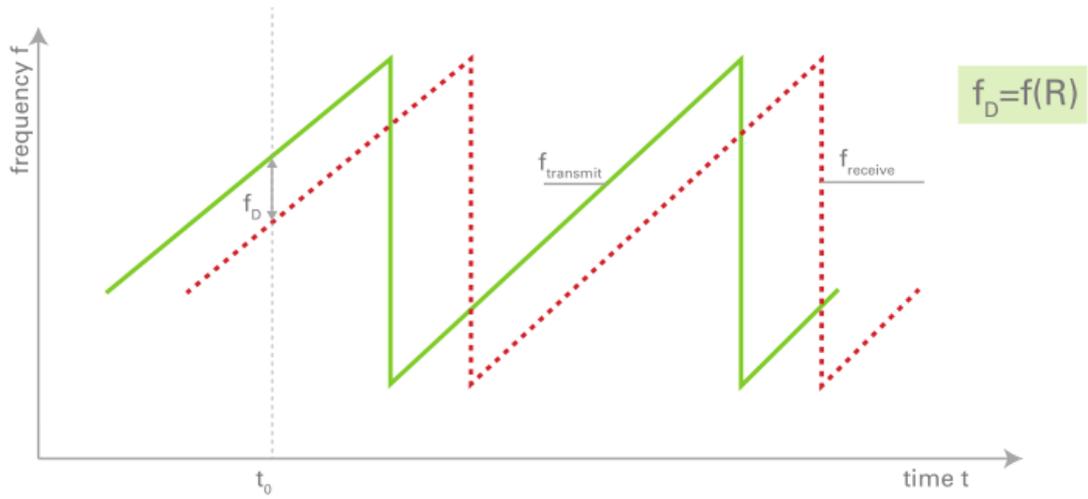


Abbildung 2.5: Die Sägezahnmodulation Quelle: Weidmann

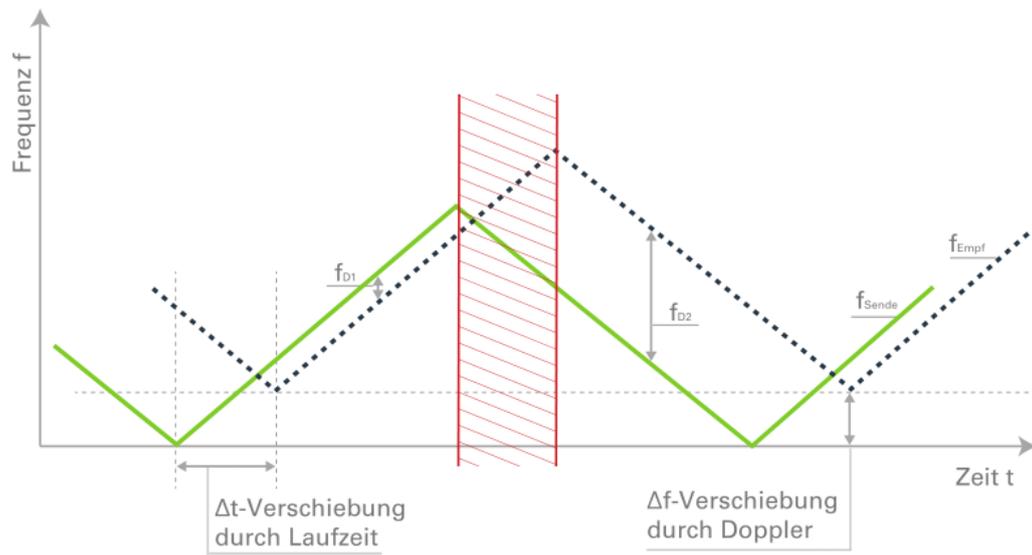


Abbildung 2.6: Dreiecksmodulation Quelle: Weidmann

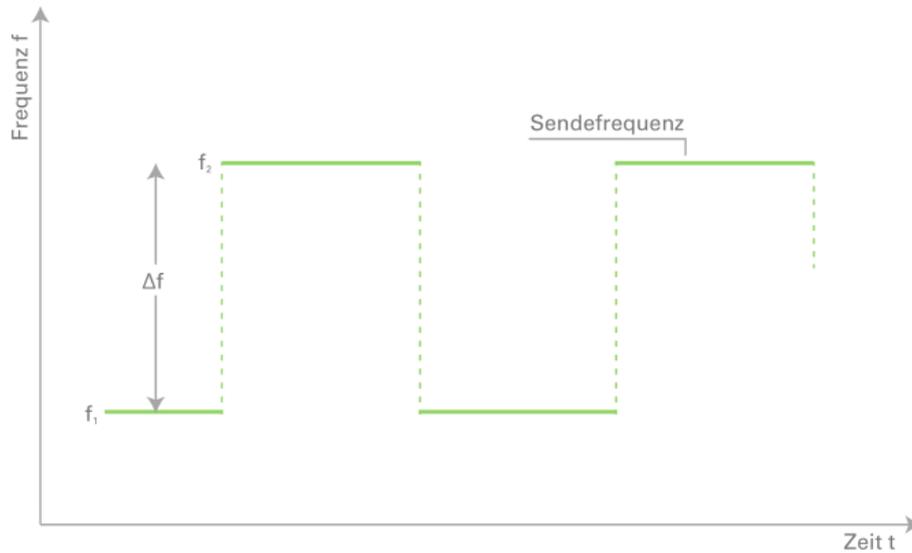


Abbildung 2.7: Zwei Frequenz Modulation Quelle: Weidmann

Tabelle 2.1: Sensorenvergleich Quelle: Weidmann

| Sensor          | Vorteile  | Nachteile   |
|-----------------|---|---|
| Passiv Infrarot | <ul style="list-style-type: none"> <li>• Geringer Stromverbrauch</li> <li>• Preisgünstig</li> </ul>   | <ul style="list-style-type: none"> <li>• Begrenzte Reichweite</li> <li>• Störanfällig</li> <li>• Kompliziertes Linsensystem</li> <li>• Reiner Präsenz Detektor</li> </ul> |
| Ultraschall     | <ul style="list-style-type: none"> <li>• Preisgünstig</li> <li>• Entfernungsinformation</li> </ul>  | <ul style="list-style-type: none"> <li>• Geringe Reichweite</li> <li>• Störanfällig</li> <li>• Keine geschwindigkeitsinformation</li> </ul>                               |
| Radar           | <ul style="list-style-type: none"> <li>• Geschwindigkeitinformation</li> <li>• Entfernungsinformation</li> <li>• Allwettertauglich</li> <li>• Durchstrahlt Kunststoffe</li> </ul> | <ul style="list-style-type: none"> <li>• Mittlere Kosten</li> <li>• Hohe Bandbreite für hohe Auflösung</li> </ul>   |

## 2.3 Zusammenfassung und Auswahl des benötigten Radartypen

Pulsradare sind für die Bewegungserkennung unpraktisch, da aufgrund der kleinen Signallaufzeit bei kleinen Entfernungen, die Berechnung aufwändig ist. Des Weiteren müssen Festziele ausgeblendet werden. Beispielsweise beträgt die Laufzeit der Signale bei einem Ziel in zehn Metern Entfernung gerade mal 67 ns. Das erschwert die Auswertung mittels eines Mikrocontrollers.

$$\tau = \frac{2 * R}{c} = \frac{2 * 10m}{300.000.000 \frac{m}{s}} = 67ns \quad (2.11)$$

Mit R die Distanz und c die Lichtgeschwindigkeit.

Aufgrund des mittleren Preises und der Verfügbarkeit, sind Dauerstrich Doppler-Radare im Bereich der Bewegungserkennung gut geeignet. Da die Doppler-Frequenz erst bei einer Bewegung entsteht, müssen feste Ziele nicht extra ausgefiltert werden. Eine Auswertung der Signale kann zudem mit einfachen Schaltungen realisiert werden.

## 3 Technische Anforderungen und Analyse

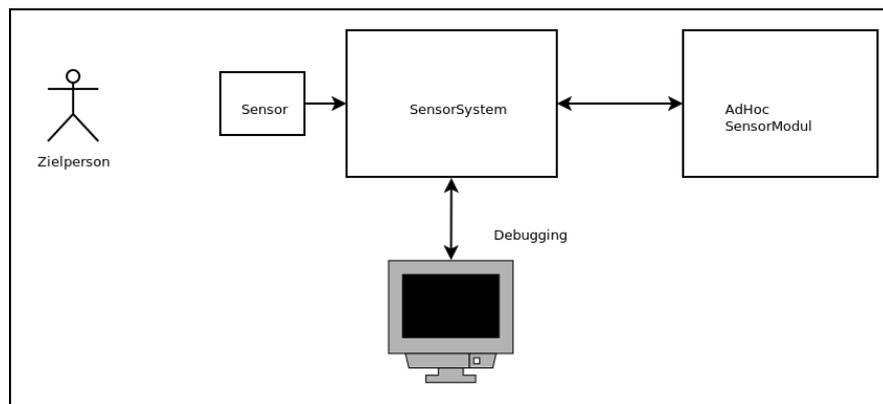


Abbildung 3.1: übersicht des Gesamtsystems

In diesem Kapitel wird ein Überblick über den Gesamtaufbau (Abbildung 3.1) des Sensorsystems verschafft. Es werden Bestandteile des Sensorsystems und ihre Funktionsweise präsentiert und genauer auf die Anforderungen an die Hardware, wie den Sensor und den Mikrocontroller eingegangen. Abschließend werden verschiedene Ansätze für die Bewegungserkennung vorgestellt.

### 3.1 Anforderung an das Sensorsystem

Das zu entwickelnde System soll die bestehenden mobilen Überwachungseinheiten von AdHoc Networks erweitern und die Auswertung der Radarsignale übernehmen, um Bewegungen in einem überwachten Bereich zu entdecken. Weiterhin sollen die Ergebnisse der Bewegungserkennung über eine Kommunikationsschnittstelle an die Überwachungseinheiten weitergeleitet werden. Das System soll stromsparsam und zuverlässig arbeiten. Die eingesetzten Sensoren sollten Reichweiten von bis zu 50 Metern ermöglichen.

## 3.2 Signalverarbeitungskette

Die Signalverarbeitungskette Abbildung 3.2 beschreibt die Arbeitsweise des Gesamtsystems. Der CW Radarsensor erzeugt ein Dopplersignal. Dieses Signal wird zunächst verstärkt, um von einem AD-Wandler erfasst werden zu können. Anschliessend werden die digitalen Werte von einem Mikrocontroller weiterverarbeitet, um eine Bewegung festzustellen. Das Ergebnis der Verarbeitung wird über eine Kommunikationsschnittstelle an die Steuerplatine weitergeleitet.

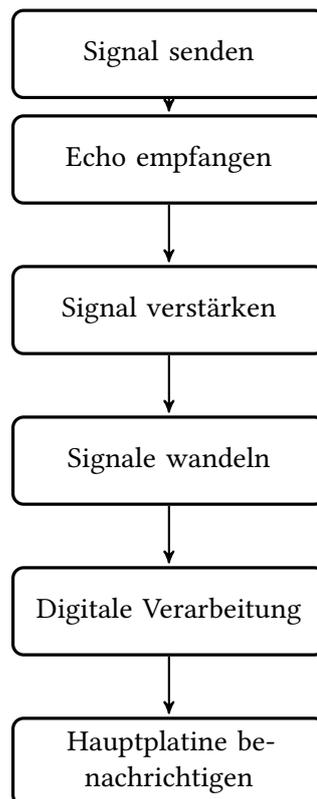


Abbildung 3.2: Signalverarbeitungskette

## 3.3 Mikrocontroller

Der Mikrocontroller muss in der Lage sein die Ausgangssignale des Radarsensors in Echtzeit zu digitalisieren, zu verarbeiten und eine Detektion an die Steuerplatine zu melden. Das setzt voraus, dass die Recheneinheit über einen AD-Wandler und genügend Rechenleistung verfügt, um die eindigitalisierten Daten zu verarbeiten. Zudem wird zu Debugzwecken, und zur Steuerung

des Mikrocontrollers eine Schnittstelle benötigt, worüber die Verbindung zum PC realisiert wird. Eine weitere Schnittstelle dient zur Kommunikation mit den mobilen Überwachungseinheiten. Darüber hinaus soll der Mikrocontroller mit der bereits vorhandenen Laborausrüstung von AdHoc Networks kompatibel sein und keine weitere Hard- oder Software benötigen.

## 3.4 Abtastung

Laufgeschwindigkeiten von 4 cm/s bis 4 m/s erzeugen bei einem Dauerstrich Doppler Radarsensor Frequenzen im Bereich von 6 bis 600 Hz. Der Merksatz des Abtasttheorems nach CE. Shanon besagt, dass ein kontinuierliches Tiefpass-Signal mit einer Frequenz abgetastet wird, die mehr als doppelt so groß ist, wie die höchste im Signal vorkommende Frequenz. (siehe Meyer, (S. 158-161)) Unter Beachtung des Abtasttheorems muss die Sampling Frequenz des AD Wandlers größer sein als 1.2 Khz. Eine 12 Bit Auflösung für die ADC Werte kann die Quantisierungsfehler gering halten. Somit kann ein analoges Signal von 0 bis 3.3 V auf einem Bereich von 0 bis 4096 abgebildet werden. Im Vergleich dazu stände, bei einer 8 bit Auflösung nur ein Bereich von 0 bis 256 zu Verfügung.

## 3.5 Verstärkung

Die meisten Dopplersensoren werden aus Kostengründen ohne Verstärker hergestellt. Sie liefern an ihrem Ausgang Spannungen im Bereich von 100 nV bis 100 mV. Dies erschwert die Auswertung der Signale am ADC. Um eine höhere Signalqualität am ADC zu erreichen, ist eine Verstärkerschaltung notwendig. Auf der Abbildung 3.3 ist ein Vergleich zwischen einem unverstärkten Sensorsignal (Rot) und dessen Verstärkung (Blau) abgebildet.

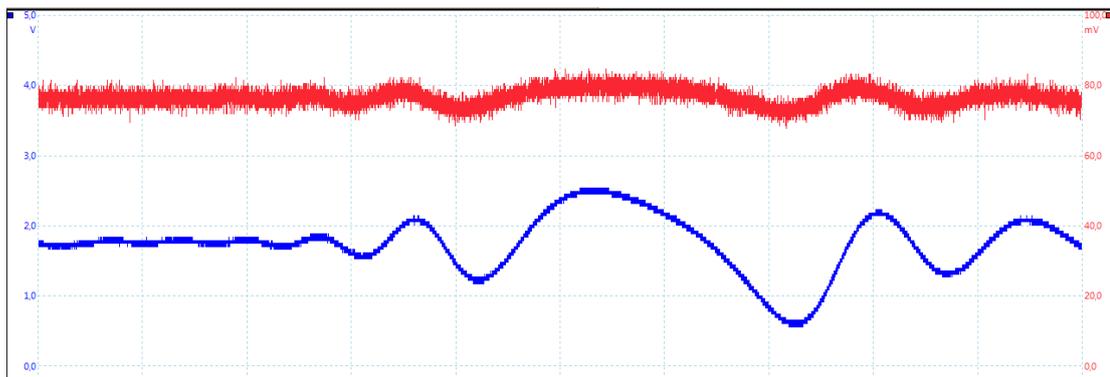


Abbildung 3.3: Der Pegel des unverstärkten Signals des IPM165 Sensors in Rot liegt bei 80mV, das verstärkte Signal in Blau bei 1.8 V

### Verstärker für CW Doppler-Radare

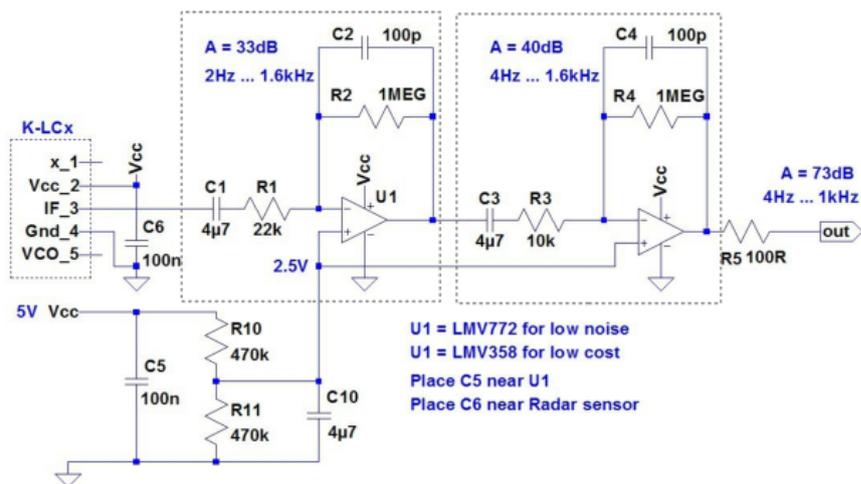


Abbildung 3.4: Vorgeschlagene Verstärkerschaltung der Firma Rfbeam

In den Applikationsschriften des Sensorherstellers **GmbH (a)**, wird die Verstärkerschaltung, die in 3.4 abgebildet ist, vorgeschlagen. Die Gesamtverstärkung beträgt 73 dB. Diese Schaltung besteht aus zwei Verstärkungsstufen und einen Spannungsteiler. Jede Verstärkungsstufe bildet einen Bandpass (Abbildung. 3.5). Dadurch kann die Bandbreite der Frequenzen des Eingangssignals begrenzt werden. Die Verstärkung wird durch das Verhältnis der Widerstände R1 und R2 angegeben. Die Gesamtverstärkung ist die Summe der Verstärkungen der einzelnen Stufen.

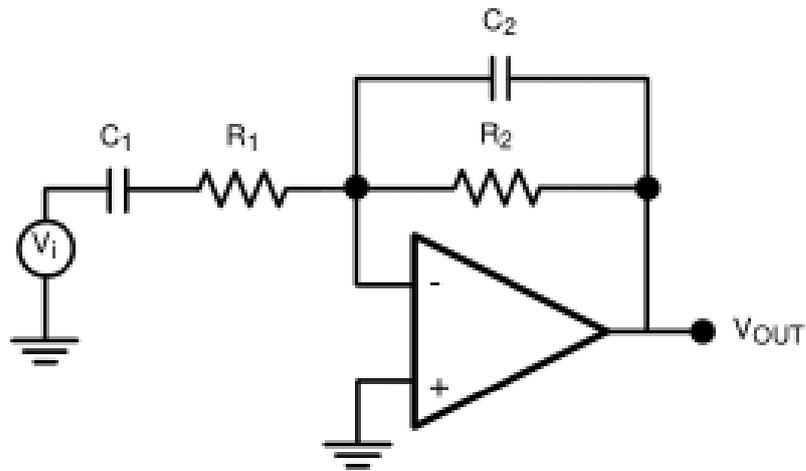


Abbildung 3.5: Bandpass: bestehend aus einem Hochpass (R1, C1) und Tiefpass (R2, C2)

$$v = 20 \cdot \log\left(\frac{R_2}{R_1}\right) \text{db} \quad (3.1)$$

Die Grenzfrequenz des Hochpasses kann mit der folgenden Formel berechnet werden:

$$f_c = \frac{1}{2\pi \cdot R_1 \cdot C_1} \quad (3.2)$$

Die Grenzfrequenz des Tiefpasses:

$$f_c = \frac{1}{2\pi \cdot R_2 \cdot C_2} \quad (3.3)$$

### Spannungsteiler

Da negative Spannungen über den ADC nicht meßbar sind, wird die Leerlaufspannung um die Hälfte des Pegels der Versorgungsspannung angehoben. Dazu wird ein einfacher Spannungsteiler (Abbildung 3.6) mit zwei gleichen Widerständen verwendet. Die Kapazitäten C11, C12 und C18 gleichen den Spannungsabfall der Stromquelle aus.

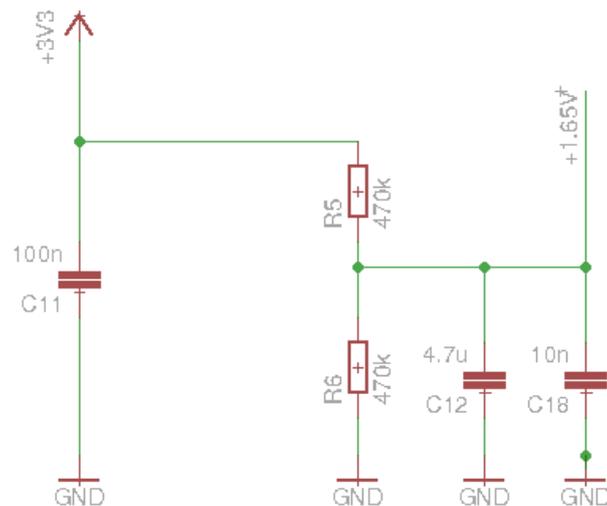


Abbildung 3.6: Spannungsteiler: Durch die zwei gleichen Widerstände R5 und R6 wird die Versorgungsspannung halbiert

## 3.6 Auswahl der Radarsensoren

Unter den zur Verfügung stehenden Sensoren befinden sich ein Mono CW Doppler-Sensor und ein Stereo Low Power CW Doppler-Sensor. Diese unterscheiden sich durch ihre Reichweite, Öffnungswinkel und ihre Betriebsspannung.

### 3.6.1 IPS165 CW Doppler-Sensor

Der IPM-165 (Abbildung 3.7) ist ein kostengünstiger Mono CW Radarsensor, der nur Präsenz- und Geschwindigkeitsinformationen liefern kann. Dieses Modul ist hauptsächlich für die Bewegungserkennung konzipiert und besitzt laut Hersteller eine Reichweite von ca 10 bis 15m. (Quelle [InnoSenT](#)). Dieser Sensor wird aufgrund seiner Reichweite, Verfügbarkeit und den geringen Preis für die Test der Platine und des Verstärkers genutzt.

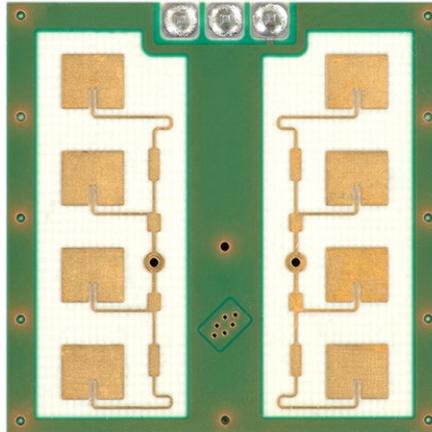


Abbildung 3.7: IPM 165

| Parameter                     | Symbol       | Min    | Typ    | Max  | Units  |
|-------------------------------|--------------|--------|--------|------|--------|
| transmit frequency            | F24          | 24.000 | 25.250 |      | GHz    |
| temperature drift (frequency) | $\Delta f$   | -1     | -      |      | MHz/°C |
| output power (EIRP)           | P_out        | 16     | -      |      | dBm    |
| IF-output                     | offset       | -300   | -      | 300  | mV     |
| IF-amplifier                  | no amplifier |        | -      |      |        |
| full beam width @ -3dB        | horizontal   |        | 80°    |      | °      |
|                               | vertical     |        | 35°    |      | °      |
| side-lobe suppression         | horizontal   | 12     |        |      | dB     |
|                               | vertical     | 13     |        |      | dB     |
| supply voltage                | V_CC         | 4.75   | 5.0    | 5.25 | V      |
| supply current                | I_CC         | 30     | 40     |      | mA     |
| operating temperature         | T_OP         | -20    | 60     |      | °C     |

### 3.6.2 K-MC1\_LP low Power Radarsensor

Mit dem K-MC1\_LP low power Stereo Radar-Sensor (Abbildung 3.8) kann die Präsenz, die Geschwindigkeit und die Laufrichtung ermittelt werden. Die Entfernung kann nicht bestimmt werden, da ein Frequenzmodulationseingang nicht vorhanden ist.

Die Bewegungsrichtung kann durch die Phasenverschiebung der beiden Ausgangssignale ermittelt werden. Durch den geringen Stromverbrauch und die große Reichweite von bis zu 50 Metern ist dieser Sensor ideal für die Umgebungsüberwachung geeignet. Ferner besitzt dieser

### 3 Technische Anforderungen und Analyse

---

Sensor eine schmale Strahlungskeule mit 12 Grad in die horizontale Richtung und 25 Grad in die vertikale Richtung.

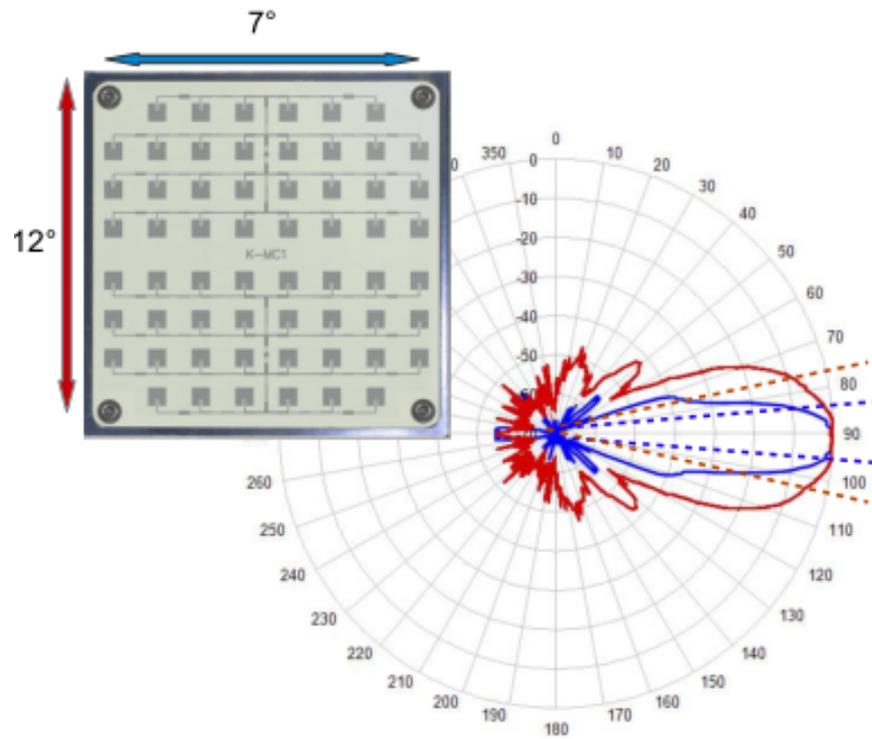


Abbildung 3.8: K\_MC1\_LP

Tabelle 3.1: Auszug aus dem Datenblatt des Sensors

|                            |              |
|----------------------------|--------------|
| Current consumption (typ.) | 7.5mA        |
| Supply Voltage             | 3.15V bis 6V |
| IF output DC offset (typ.) | 1.5V         |

## 3.7 Bewegungserkennung

### 3.7.1 Schmitt Trigger

Bei einem Schmitt Trigger handelt es sich um eine Komparatorschaltung mit Hysterese. Die Schaltung (Abbildung 3.9) besteht aus einem Operationsverstärker, einem Spannungsteiler und einem mitgekoppelten Widerstand. Mit einem Schmitt Trigger ist es möglich zwei Schwellenwerte  $U_H$  und  $U_L$  zu bestimmen, bei deren Überschreitung oder Unterschreitung ein Ausgangssignal ausgegeben wird. Diesen Bereich nennt man die Hysteresebreite. Der Ausgangspegel wird aufrechterhalten, solange sich der Eingangspegel zwischen  $U_H$  und  $U_L$  hält. Der Schmitt Trigger stellt eine einfache Methode dar, analoge Signale zu digitalisieren.

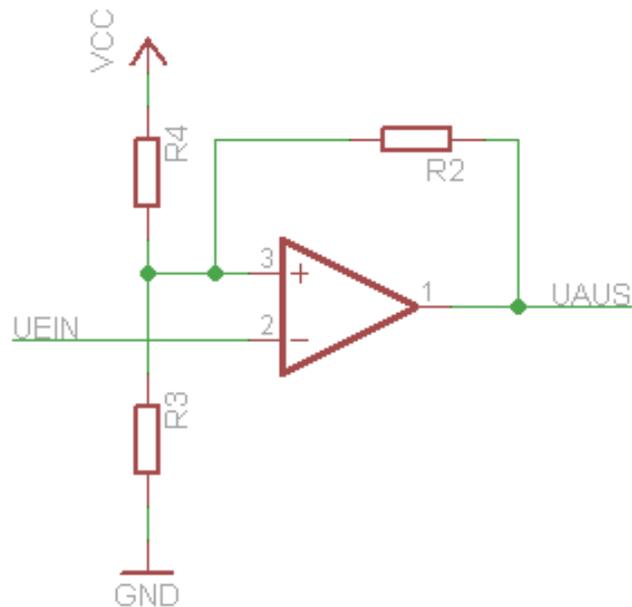


Abbildung 3.9: Ein invertierender Schmitt Trigger

- Spannungsteiler
- Hysterese

$$U_H = U_{ref} + (U_{HA} - U_{ref}) \frac{R_1}{R_1 + R_2} \quad (3.4)$$

$$U_L = U_{ref} - (U_{ref} - U_{LA}) \frac{R_1}{R_2 + R_1} \quad (3.5)$$

- Hysteresebreite

$$B_H = U_H - U_L \quad (3.6)$$

Der Mitkopplungswiderstand  $R_2$  lässt sich wie folgt dimensionieren:

$$R_2 = R_1 \frac{U_H - U_H A - U_L + U_L A}{U_L - U_H} \quad (3.7)$$

Die Hysteresebreite von verhältnis  $R_1/R_2$  ab. Wählt man  $R_2$  so dass,  $R_2$  größer als  $R_1$  ist, erhält man einen sensitiven Schmitt Trigger mit einer geringen Hysteresebreite. Dies erlaubt Signale mit einer kleinen Amplitude zu erfassen. In der Abbildung 3.9 ist die Ausgabe eines invertierenden Schmitt Triggers zu sehen, der an einem Radarsensor angeschlossen ist (vgl. [mikrocontroller.net](http://mikrocontroller.net))

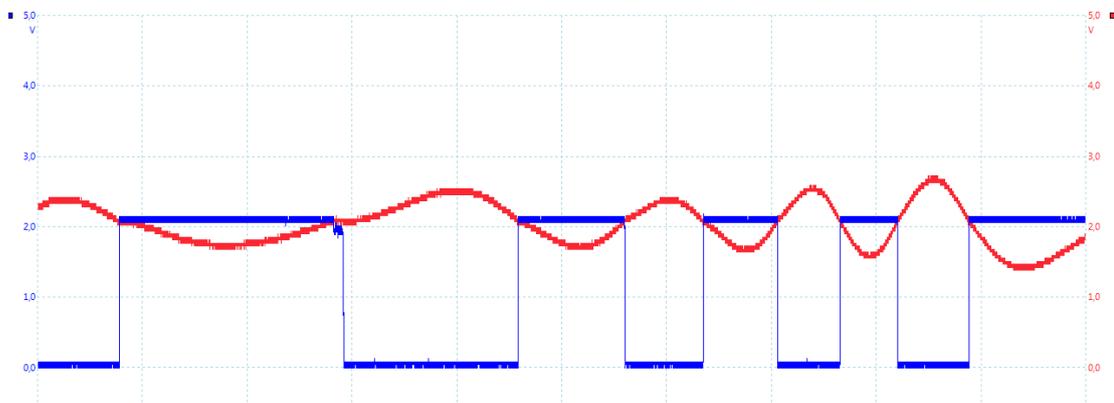


Abbildung 3.10: Schmitt Trigger: in Rot ist das Oscilloskop Bild des IPM 165 Sensors zu sehen, in Blau die Ausgabe des Schmitt Triggers

### 3.7.2 Software Basierte Erkennung

Aus den Datenblättern des Herstellers RfBeam (siehe . [RFbeam](#)) geht hervor, dass durch den Einsatz der Fast Fourier Transformation für die Signalverarbeitung eine Verdopplung der Reichweite gegenüber einer reinen Hardware basierten Erkennung möglich ist.

#### Fourier Transform

Die Basis für die Fourier Transformation ist die Fourier Reihe. Mit ihrer Hilfe kann eine periodische Funktion  $f(t)$  (Formel 3.8) in ihre Sinus und Cosinus Anteile zerlegt werden. Dabei bestimmen die Cosinus Glieder die geraden und die Sinus Glieder die ungeraden Anteile von

$f(t)$ , wobei die Koeffizienten  $a_k$  und  $b_k$  (Formel 3.9 und 3.10) die Amplituden der unterschiedlichen Frequenzen darstellen. Dabei ist  $\omega_k$  die Kreisfrequenz,  $a_0/2$  der Gleichstromanteil und  $T_p$  die Grundperiode.

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cdot \cos(k\omega_0 t) + b_k \cdot \sin(k\omega_0 t)) \quad (3.8)$$

$$a_k = \frac{2}{T_p} \int_0^{T_p} x(t) \cdot \cos(k\omega_0 t) dt \quad (3.9)$$

$$b_k = \frac{2}{T_p} \int_0^{T_p} x(t) \cdot \sin(k\omega_0 t) dt \quad (3.10)$$

Mit der Fourier Transformation kann eine Spektralanalyse eines abgetasteten Signals durchgeführt werden. So erhält man nach der Transformation von  $N$  Abtastwerten im Zeitbereich,  $N$  Abtastwerte im Frequenzraum (Vgl. Ludloff (S. 7-67)). Die dadurch entstandenen Spektrallinien stellen die Vielfachen der Abtastfrequenz dar, wobei die nullte Stelle den Gleichstromanteil anzeigt und die letzte Stelle das  $N-1$  Vielfache der Abtastfrequenz. Die dominierenden Frequenzen haben höhere Amplitudenwerte, die sich als Peaks im Frequenzbereich herausstellen. Um eine Bewegung festzustellen wird ein Schwellenwert errechnet bei dessen Überschreitung eine Bewegung gemeldet wird.

## 4 Design

In diesem Kapitel werden konkrete Bauelemente für den Aufbau der Microcontrollerplatine ausgesucht, und Schnittstellen für die Kommunikation festgelegt. Darüber hinaus wird die dazugehörige Schaltung sowie das Layout entworfen und die Funktionweise der benötigten Software erklärt.

### 4.1 Übersicht

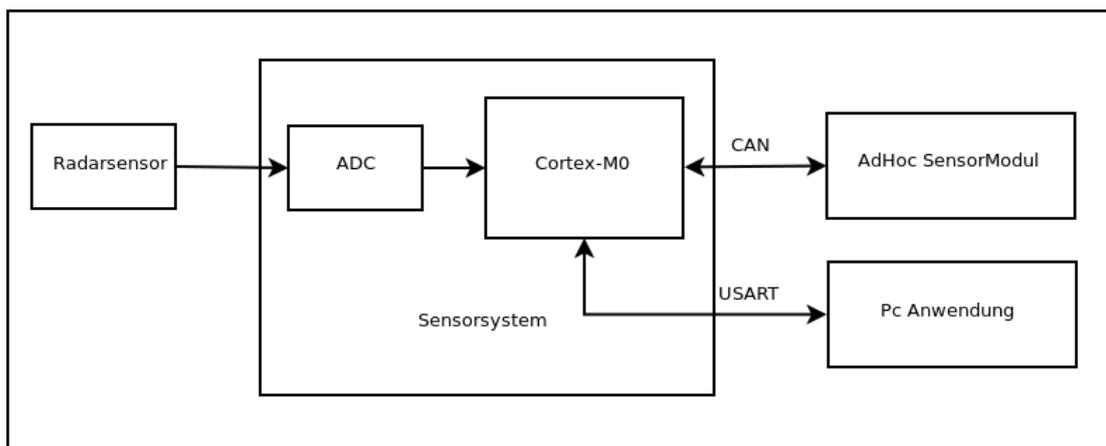


Abbildung 4.1: Übersicht des Gesamtsystems

### 4.2 Hardware Komponenten

#### 4.2.1 MCU

Als Microcontroller kommt der 32 bit ARM Cortex-M0 mit der Bezeichnung STM32F07C8 der Firma STMICROELECTRONICS zum Einsatz. Dieser bietet mit seinen 48 Mhz Takt genügend Rechenleistung, um die Signalverarbeitung in Echtzeit durchzuführen. Weitere Eigenschaften des ARMs werden in der Tabelle 4.1 aufgelistet.

Tabelle 4.1: Übersicht der Eigenschaften des STM32F07C8

|            |                                |
|------------|--------------------------------|
| Core       | ARM 32-bit Cortex-M0           |
| Memory     | 64 to 128 Kbytes               |
| SRAM       | 16 Kbytes                      |
| IO supply  | 2.0 bis 3.6 V                  |
| Pins       | 48                             |
| ADC        | 12 bit ADC (16 channels)       |
| DAC        | 12 bit DAC ( 2 channels)       |
| Timers     | 16                             |
| Interfaces | Usart, CAN, SPI, I2C, USB, SWD |

#### 4.2.2 Verstärker

Der LMV774 ist ein low noise Operationsverstärker. Durch seine kleinen Dimensionen und den niedrigen Stromverbrauch eignet sich dieser gut für die anstehende Aufgabe. (TI (b))

#### 4.2.3 Oszillator

Ein Oszillator dient als Taktquelle für Mikroprozessoren. Externe Quartz Oszillatoren haben eine höhere Genauigkeit, als der im Cortex-M0 integrierte RC Oszillator. Aus diesem Grund wird der 8 MHz Quartz benutzt, der bereits im Adhoc Sensormodul verwendet wird.

#### 4.2.4 Stromversorgung

Tabelle 4.2: Stromverbrauch der Hardwarekomponenten

| Part      | Verbrauch in micro A | Verbrauch bei 48 Mhz in mA |
|-----------|----------------------|----------------------------|
| Cortex-M0 | 1                    | 24.1                       |
| ADC       | 4.1                  | 0.197                      |
| DAC       | 4.7                  | 0.225                      |
| CAN       | 12.4                 | 0.595                      |
| USART3    | 5.4                  | 0.259                      |
| LMV774    |                      | $(0.55 * 4) = 2.1$         |

Der Mikrocontroller wird durch die Batterie des AdHoc Sensormoduls mit einer Spannung von 3.3V versorgt. Ein Spannungsregler ist auf der Hauptplatine vorgeschaltet und dient zur Stabilisierung der Betriebsspannung des Akkus. Anhand der Tabelle 4.2 kann der Strombedarf des System im Betrieb eingeschätzt werden. Es ergibt sich ein Stromverbrauch von 27.476 mA.

#### 4.2.5 Sensor Energie-Management

Laut Weidmann (S. 140) verbrauchen die meisten Doppler Dauerstrich Radarsensoren nicht weniger als 30 mA. Um den Verbrauch im Akkubetrieb zu senken, wird die Spannungsversorgung des Sensors getaktet. Dies wird mit Hilfe einer einfachen Mosfet Schaltung erreicht (Abbildung 4.2). Sie besteht aus einem Widerstand und einen Mosfet Transistor. Der Transistor fungiert als Schalter für die Stromversorgung. Unter der Voraussetzung, dass der Oscillator im Sensor schnell genug anschwingt, kann mit Hilfe eines Timers der Vcc pin des Sensors gepulst werden.

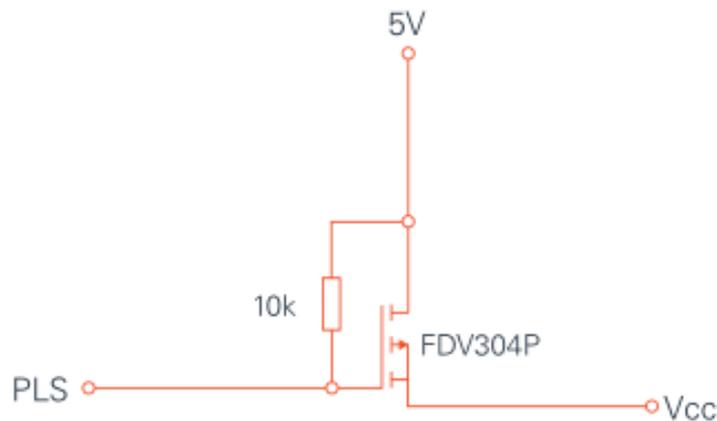


Abbildung 4.2: Transistorschaltung zum Pulsen der Stromversorgung des Radarsensors

#### 4.2.6 Reichweitenerhöhung

Da die Eigenschaften der Sensoren durch den Herstellungsprozess fest vorgegeben sind, ist eine Reichweitenerhöhung nur durch die Erhöhung der Sendeleistung möglich. In Goebel (2003) (S. 70) wird aus der Radargleichung die Formel 4.2 für die Reichweite hergeleitet. Eine Verdoppelung der Reichweite wäre mit einem Verzehnfachen der Sendeleistung verbunden. Daraus resultiert, dass die maximal erlaubte Sendeleistung von 100 mW bei 24 GHz Sensoren

überschritten wird (Gleichungen 4.3 und 4.4). Um dies nachvollziehen zu können, muss die Radargleichung vorgestellt werden. Die Gleichung für den Radar 4.1 stellt eine Beziehung zwischen den technischen Daten eines Radarsystems und seiner Erfassungsreichweite her (vgl. Weidmann (S. 135)).

$$\frac{P_r}{P_t} = \frac{g_t \cdot g_r \cdot \sigma^2}{r^4} \quad (4.1)$$

Die Kenngrößen der Radargleichung sind:

- $P_r$  : Die Empfangsleistung des Radars
- $P_t$  : Die Sendeleistung des Radars
- $g_t$  : Der Gewinn der Sendeantenne
- $g_r$  : Der Gewinn der Empfangsantenne
- $\sigma$  : Die Rückstreulfläche oder Radarquerschnitt. In der Praxis bedeutet dies die effektiv wirksame Reflektionsfläche des Zieles

$$R_{max} = \sqrt[4]{\frac{P_s G_e G_s \sigma}{P_e}} \quad (4.2)$$

$$2 * R_{max} = \sqrt[4]{16} * \sqrt[4]{\frac{P_s G_e G_s \sigma}{P_e}} \quad (4.3)$$

$$2 * R_{max} = \sqrt[4]{\frac{16 * P_s G_e G_s \sigma}{P_e}} \quad (4.4)$$

Die Empfindlichkeit des Sensors kann durch die Anwendung der Fourier Transformation erhöht und dadurch die Reichweite des Sensors gesteigert werden (siehe. GmbH (b)(S. 8)). Zur Festlegung der CAN IDs wird ein Dipswitch mit 4 Schaltern verwendet, womit 16 mögliche Adressen einstellbar sind.

Für die Anbringung der Sensorplatine ist eine Fläche von 4.5x4 cm vorgesehen, weshalb die Dimension, bei der Auswahl der Komponenten, eine wichtige Rolle spielt. Aus diesem Grund, werden nur SMD Komponenten verbaut.

### 4.2.7 Schnittstellen

#### Kommunikation über CAN

Falls eine Bewegungserkennung stattgefunden hat, wird eine CAN Nachricht erstellt, und eine GPIO Leitung, die mit dem AdHoc Steuermodul verbunden ist, auf HIGH gesetzt, dadurch

wird der Microcontroller der Steuerplatine aufgeweckt, falls er sich im Schlafmodus befindet. Anschliessend wird die CAN Empfangsroutine aufgerufen und die Nachricht empfangen.

## CAN

Als Kommunikationsinterface wird CAN gewählt, da beide Mikrocontroller über diese Schnittstelle verfügen und die meisten anderen Schnittstellen in Benutzung sind.

Die CAN Schnittstelle (Controlled Area Network) ist eine vom ISO definierte, serielle Schnittstelle, die ursprünglich von der Automobilindustrie entwickelt wurde, um Kabelbäume und Gewicht in Fahrzeugen zu reduzieren. CAN ist wenig anfällig für elektrische Interferenzen und kann zusätzlich Selbstdiagnose führen und Übertragungsfehler korrigieren.

Für die Datenübertragung braucht CAN zwei Leitungen. Eine  $C_H$  (HIGH) und eine  $C_L$  Leitung (LOW). Eine logische 0 wird durch das Setzen der  $C_H$  Leitung auf high und der  $C_L$  Leitung auf low signalisiert. Bei einer logischen 1 werden beide Leitung auf die Hälfte des Pegels der Versorgungsspannung gehalten (untere Hälfte in Abbildung 4.3) Quelle: Datenblatt des SN65HD23x (vgl. TI (a)).

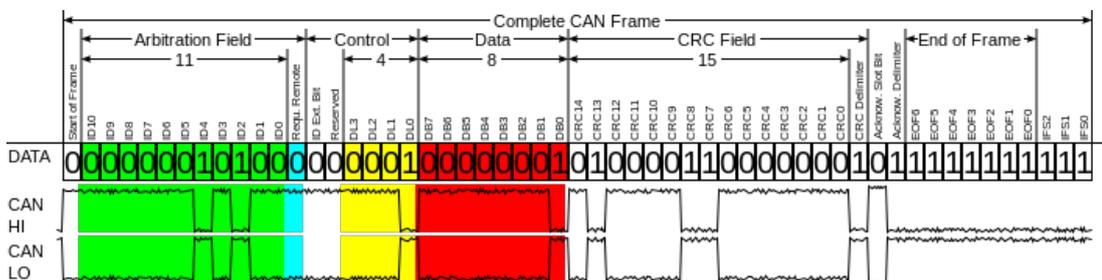


Abbildung 4.3: Oben: Aufbau eines Can Frames, Unten: Die Zustände des CAN bus

## CAN Transceiver

Der SN65HVD231 wird genutzt, da er bereits in anderen Projekten der Firma AdHoc Networks verwendung findet. Er erreicht eine Datenübertragungsrate von bis zu 1 Megabit pro Sekunde (Mbps) und enthält mehrere Schutzmaßnahmen, um das CAN Netzwerk robuster zu machen. Er kann direkt über die Versorgungsspannung von Microcontroller betrieben werden (vgl. TI (a)). Der Gesamtverbrauch des Transceivers wird als  $550 \mu A$  pro Verstärker angegeben, dadurch ergibt sich ein Gesamtverbrauch von 2.1 mA für alle vier Stufen.

Tabelle 4.3: Felder eines CAN Frames

|           |  |
|-----------|--|
| SOF       | Start of Frame bit                                 |
| 11 bit ID | Standard CAN 11 bit Kennung für Priorität          |
| RTR       | Übertragungsanforderung broadcast                  |
| IDE       | single Identifier Extension                        |
| r0        | reserviert for weitere standard                    |
| k         | 4 bit daten gröse                                  |
| DATA      | bis zu 64 bit Daten                                |
| CRC       | 16 bit redundancy check (checksum)                 |
| ACK       | zur Signalisierung einer erfolgreichen Übertragung |
| EOF       | 7 bit End of Frame                                 |
| IFS       | 7 bit interframe space                             |

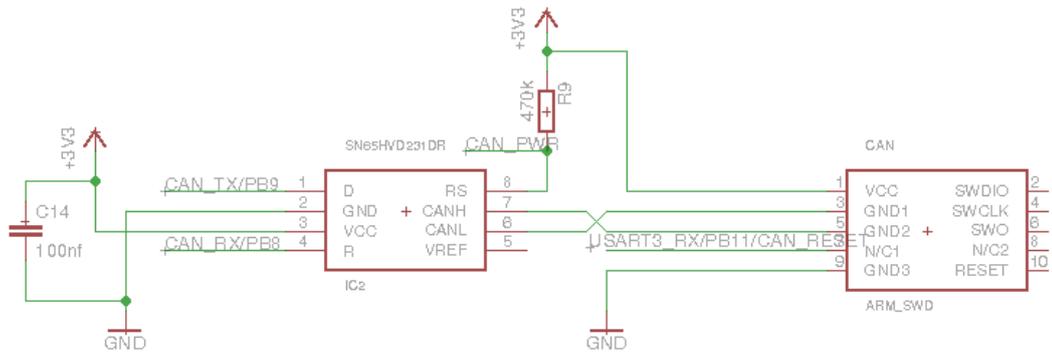


Abbildung 4.4: Die eagle Schaltung zeigt die Verbindung zwischen den CAN\_Rx und CAN\_Tx pins des  $\mu C$ 's dem CAN Transceiver (mitte) und dem Verbindungstecker (Rechts)

## USART

Für die Kommunikation mit dem PC wird eine serielle Schnittstelle benutzt. Die vom ADC eingelesenen Werte werden über die USART Schnittstelle des Microcontrollers an die Python Telemetrie Anwendung weitergeleitet. Dadurch ist es möglich Daten mit einer Geschwindigkeit von bis zu 1 Mbit/s zu übertragen. Die Datenübertragung erfolgt über zwei Schieberegister für das Senden und Empfangen.

Ein Usart Frame besteht aus 5 bis 9 DATA bits, einen Start bit und zwei Stop bits. (Abbildung 4.5). Das LSB wird als Erstes gesendet. Die Baudrate ist von 9600 bis zu 921600 bps einstellbar.

| Bit number | 1         | 2             | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     | 11          | 12 |
|------------|-----------|---------------|--------|--------|--------|--------|--------|--------|--------|--------|-------------|----|
|            | Start bit | 5-9 data bits |        |        |        |        |        |        |        |        | Stop bit(s) |    |
|            | Start     | Data 0        | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 | Data 8 | Stop        |    |

Abbildung 4.5: Uart frame

## SWD

Serial Wire Debug (SWD) ist eine von ARM entwickelte Schnittstelle, die den Zugang zu den Debug Registern einer ARM MCU ermöglicht. Anders als beim JTAG kommt SWD mit nur 2 Leitungen aus, einer Clock Leitung und einer bidirektionellen Datenleitung. Leitungen für die Stromversorgung sind nicht nötig. Mit dieser Schnittstelle sind Bandbreiten, bei einer Frequenz von 50 Mhz, von bis zu 4 MB/s zu erreichen. Zusätzlich ist eine zuverlässige Fehlererkennung integriert. Über den SWD Port lässt sich der Mikrocontroller sowohl programmieren, als auch debuggen.

### 4.2.8 Platinenlayout

Der Entwurf der Schaltung wird mit Hilfe der CAD Software Eagle von CADsoft erstellt. Für das Schaltungsdesign, bietet Eagle Bauteilbibliotheken mit verschiedenen Komponenten an. Zusätzlich ist es möglich diesen Katalog mit Hilfe des Symbol Editors, oder durch Herunterladen von fertigen Komponentenbibliotheken zu erweitern (z.B von Adafruit).

Der Eagle Device Editor bietet die Möglichkeit an neue, nicht in Eagle vorhandene Komponenten hinzuzufügen. Mit Hilfe der, in den Datenblätter beschriebenen Größen und Abständen

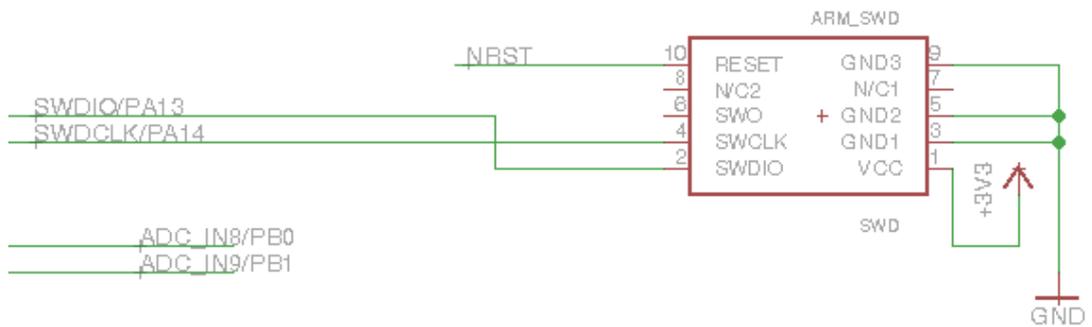


Abbildung 4.6: Die Eagle Teilschaltung zeigt die Verbindung zwischen dem Mikrocontroller und dem SWD Programmierstecker

können neue Footprints für die neuen Komponenten erstellt werden.

Die Mikrocontrollerplatine wurde nach den Vorschlägen des Herstellers ST angefertigt (siehe [Microelectronic](#)).

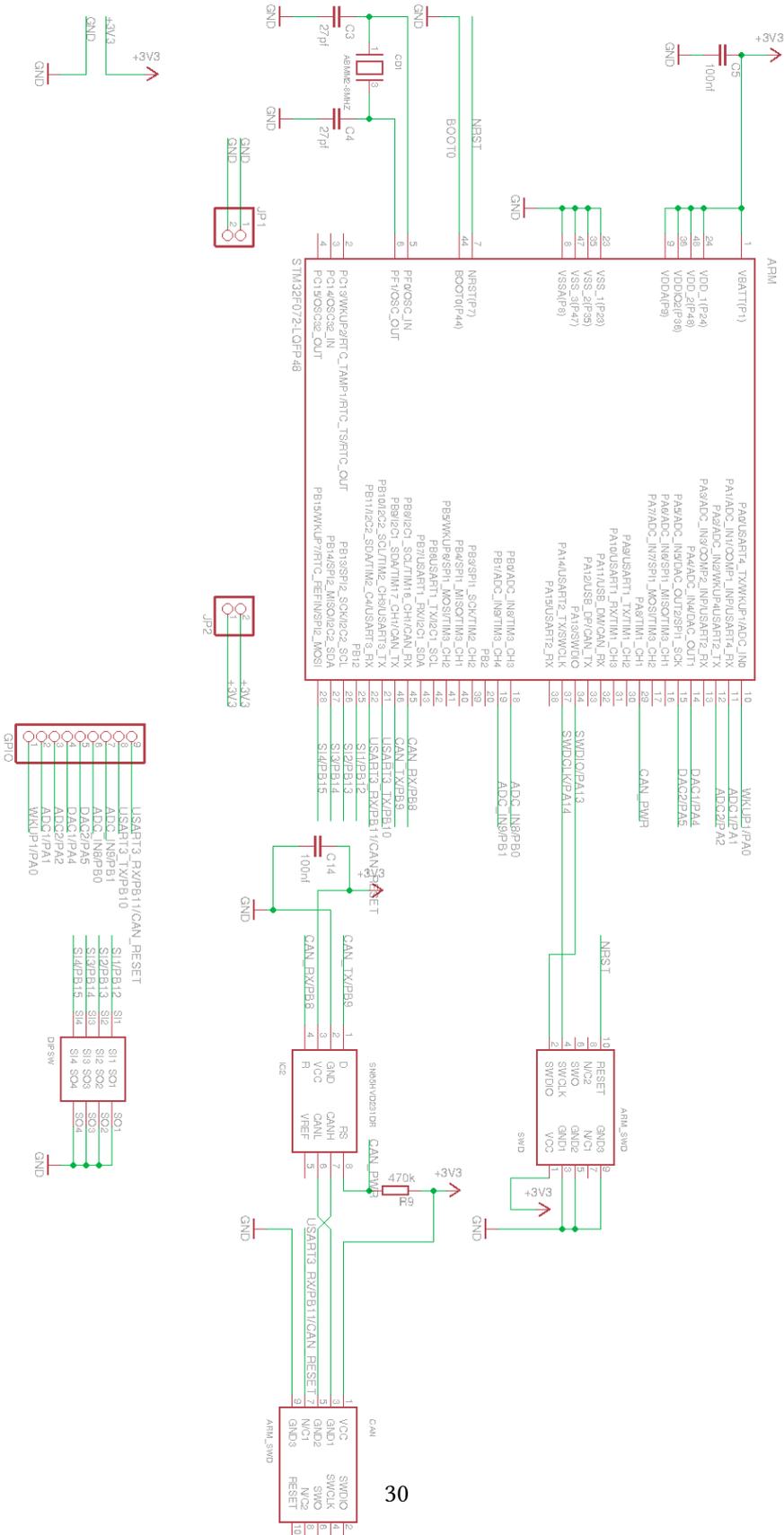
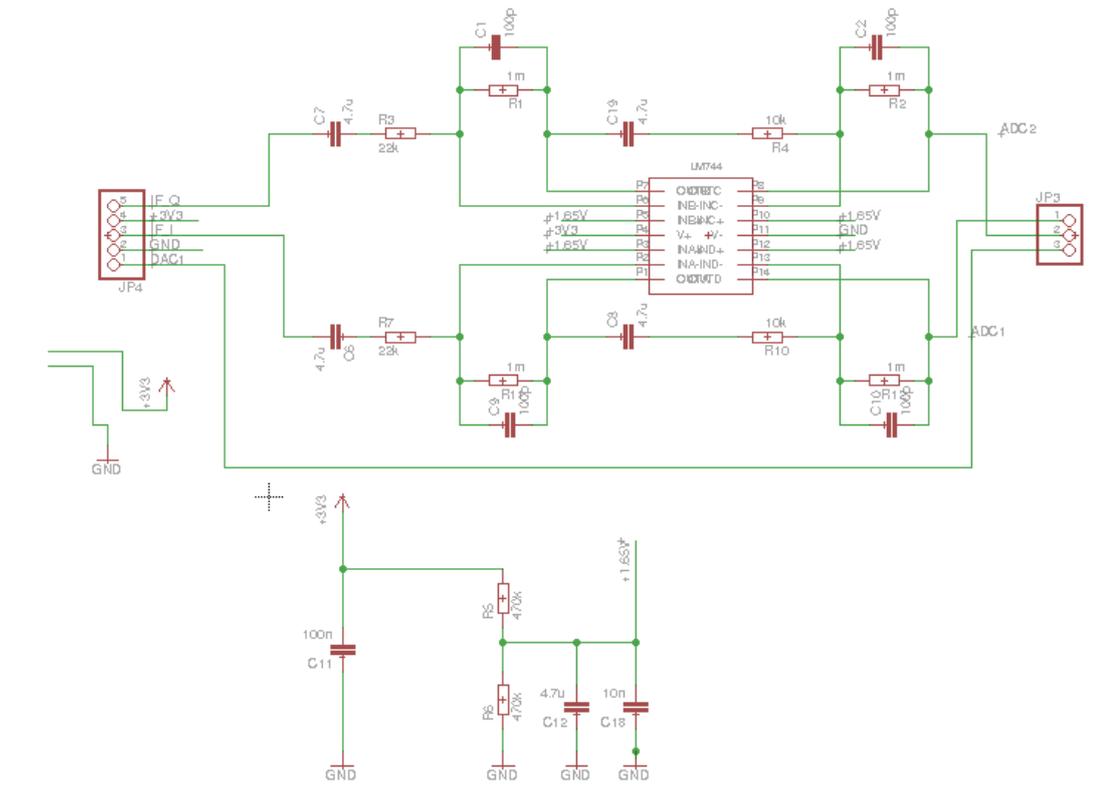


Abbildung 4.7: Schaltplan Der Mikrocontrollerplatine

### Analoge Verstärkerschaltung

Die analoge Schaltung für den Verstärker entspricht der vom Hersteller empfohlenen Schaltung. Zuerst wurde diese auf einem Breabboard realisiert und getestet, bevor sie in Eagle erstellt wurde.



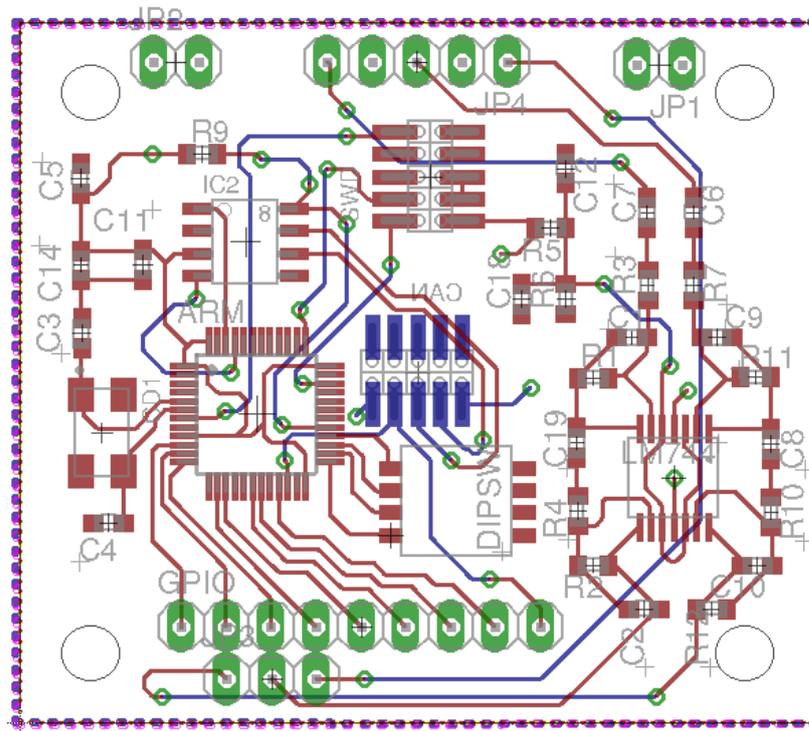


Abbildung 4.9: Platinenlayout

## 4.3 Entwicklungsumgebung

### 4.3.1 PC Software

Die Telmetrie Anwendung wird in Python unter Linux in einem einfachen Editor entwickelt.

### 4.3.2 Firmware

Die Firmware des Cortex-M0 wird in der Eclipse IDE entwickelt. Zum Flaschen und Debuggen des Microcontollers kommt der vorhandene Jlink Programmer der Firma Segger zum Einsatz. Die Microcontoller Firmware wird in die C Programmierspache implementiert. Das CAN-Modul für die Hauptplatine wurde in C++ programmiert. Zusätzlich wird die nachfolgende Software bei der Entwklung benutzt.

- Gnu Compiler Collection (gcc) (**GNU**)
- Gnu Debugger (gdb)

- JLinkGDBServer (gdb server auf der MCU Seite)

### AD Wandler

Über den, in dem Cortex-M0 integrierten AD-Wandler können analoge Werte mit einer 12 bit Auflösung, bei einer Frequenz von bis zu 1 Mhz digitalisiert werden. Als Referenzspannung wird die Versorgungsspannung von 3.3V übernommen. Der AD Wandler im STM32 unterstützt verschiedene Capture Modi. Im Single conversion Mode werden die Signale eines einzelnen Kanals gelesen. Im Dual/Triple interleaved Mode können bis zu 3 ADCs gebündelt werden um analoge Signale eines einzelnen Kanals zu lesen und so die Samplingrate zu verdreifachen. Um mehrere Signale gleichzeitig auszuwerten, ist der Multichannel mode auszuwählen.

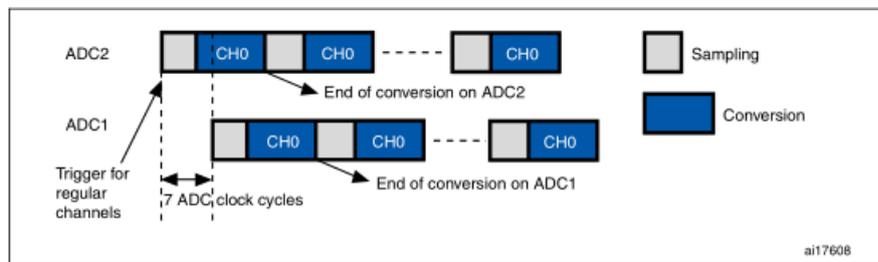


Abbildung 4.10: DualInterleaved

- Polling Mode: Im Polling Modus wird die Umwandlung durch den Aufruf der Funktion `ADC_PollForConversion()` manuell getriggert.
- Time triggered interrupt mode: Im Interrupt Modus wird der ADC durch einen Timer getriggert, so steht nach Ablauf einer Periode der eingestellten Timer-Frequenz ein neuer Wert im ADC Register.
- Injected Mode: Der Injected Mode ermöglicht die Priorisierung bestimmter Kanäle. Die Abbildung 4.11 zeigt, dass die gerade laufende Konvertierung von Kanal eins durch den höher priorisierten Kanal zehn unterbrochen und nach Ablauf dessen bearbeitung wieder aufgenommen wird.

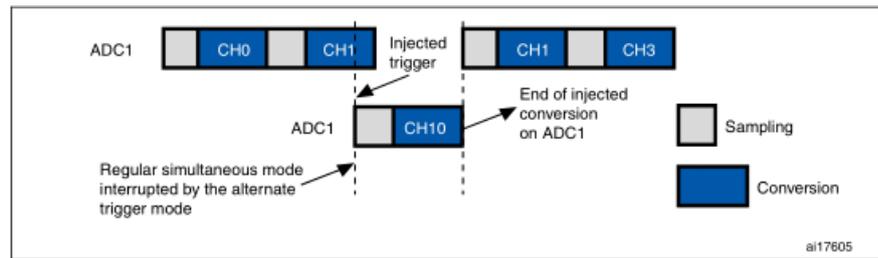


Abbildung 4.11: ADC injected Mode

- Multichannel Mode: Im multichannel Modus werden mehrere Kanäle sequentiell konvertiert und die Werte im ADC Datenregister abgelegt.
- DMA: Der DMA Modus macht es möglich, die CPU zu entlasten. Ein Speicherbereich wird dem dma Controller zugeordnet. Nachdem dieser voll oder halbvoll mit Werten aus den ADC beschrieben wurde, wird der Benutzer über einen Interrupt benachrichtigt.

Um die Abtastfrequenz zu bestimmen wurde der Time triggered Modus ausgewählt.

### 4.3.3 Telemetrie

Um die korrekte Arbeitsweise der Hardware zu überprüfen, wird eine Telemetrie Anwendung in Python entwickelt. Als Widget Toolkit kommt die Opensource Pyqtgraph ([Pyqtgraph](#)) Bibliothek zum Einsatz. Sie bietet eine Reihe von plot widgets an, die das Visualisieren der Sensorsignale ermöglichen. Sie basiert auf dem multiplatform Opensource Gui Toolkit Qt. Für die serielle Kommunikation mit dem Microcontroller kommt die Bibliothek Pyserial zum Einsatz.

## 5 Realisierung

Um die Platine bei einem Hersteller anfertigen zu lassen, müssen die Designregeln des Herstellers eingehalten werden. Diese Regeln bestimmen unter anderem die minimale Leitungsbreite oder den minimalen Abstand zwischen zwei benachbarten Pins. Die meisten Hersteller stellen Dateien für Designregeln, für die meistverbreiteten CAD Programme zur Verfügung und können mit wenig Aufwand geladen werden. Das ist schneller, als die manuelle Eingabe und weniger anfällig für Tippfehler.

Nach Abschluss des Schaltungsdesigns und des Platinenlayouts, können nun die Gerber Dateien exportiert und auf der Internetseite des ausgewählten Platinenherstellers hochgeladen werden. Dieser übernimmt dann die Herstellung der Platine.

### 5.1 Platine

Die fertige unbestückte Mikrocontrollerplatine (Abbildung 5.1) wird nach drei Wochen vom Hersteller geliefert. Bevor die Platine bestückt wird, muss auf die Pads der Komponenten ein Film mit Lötpaste aufgetragen werden. Dabei ist darauf zu achten, dass die Paste nicht zu fest aber auch nicht zu flüssig ist. Anschließend werden die Komponenten mit Hilfe einer Pinzette auf die Pads gelegt und in einem Reflow Ofen bei einer Temperatur von 250 Grad gebacken.

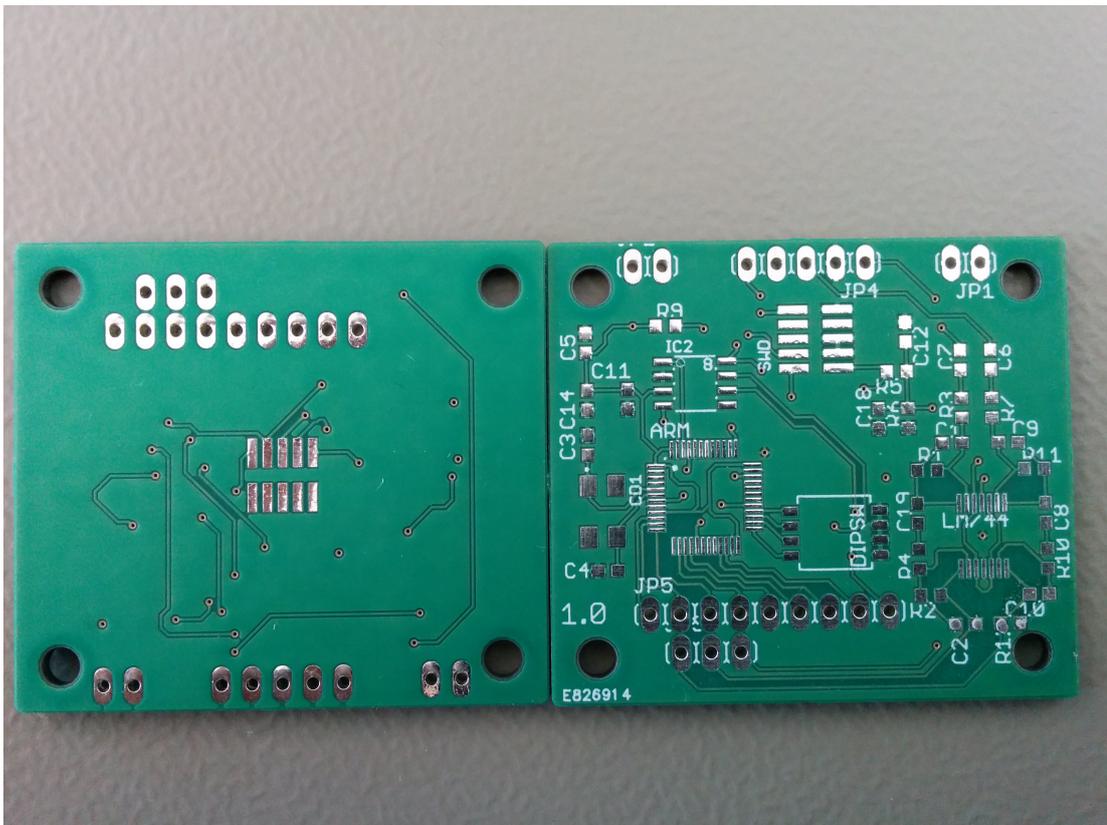


Abbildung 5.1: Unbestückte Platine

## 5.2 Firmware

Nach dem Start des Mikrocontrollers werden die verwendete Peripherie und die Schnittstellen initialisiert. Anschliessend werden, nach der Herstellung der seriellen Verbindung, die Laufzeitparameter wie zB. Frequenz des ADCs, Baudrate, und Packetgröße eingestellt. Auf der Abbildung 5.2 ist ein Flussdiagramm der den Ablauf der Software darstellt.

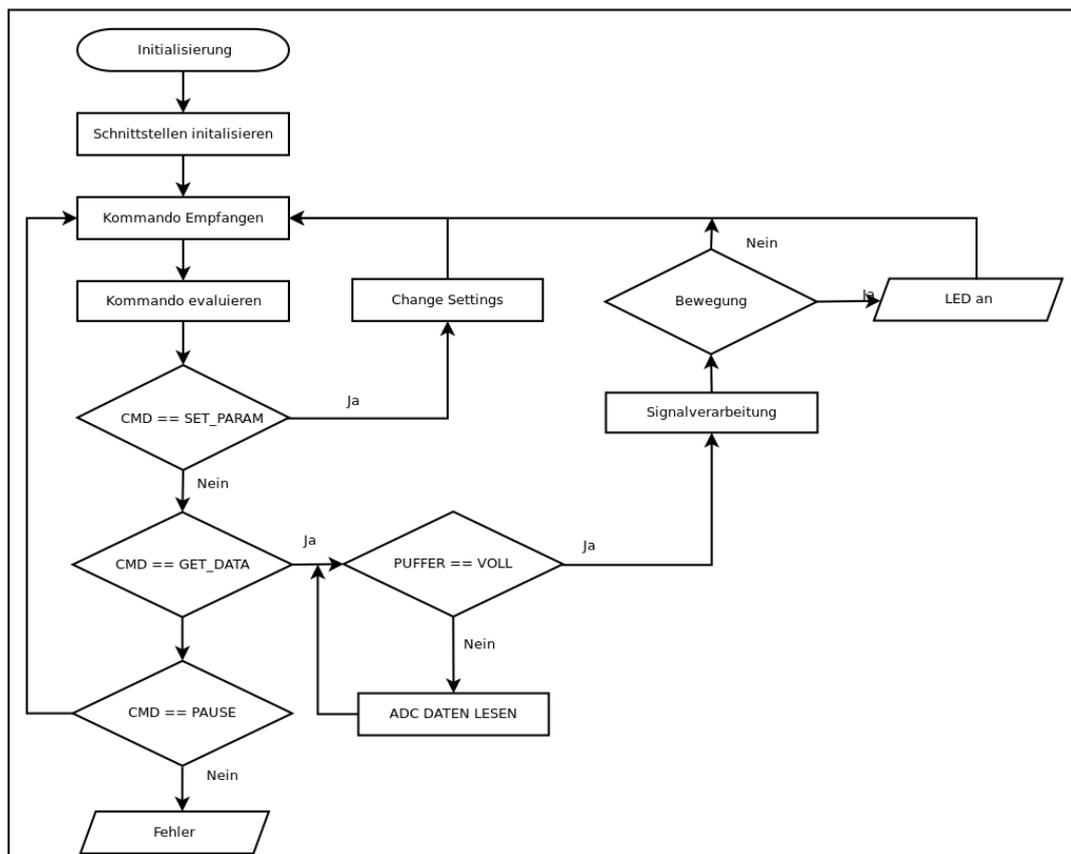


Abbildung 5.2: Flussdiagramm

```

1
2     typedef enum {SET_FREQ = 0xFB,
3                   SET_NUM_CHANNELS = 0xFC,
4                   START = 0xFA,
5                   PAUSE = 0xF0,
6                   GET_SETTINGS = 0xFE
7                   } commands;

```

Listing 5.1: Kommandos der Telemetrie Anwendung

### 5.3 PC Verbindung

Die Verbindung zum PC wird über die serielle Schnittstelle realisiert. Dazu werden die Rx und Tx Pins des Mikrocontrollers mit Hilfe eines FTDI Kabels an den USB Anschluss des PCs angeschlossen. Unter Linux erscheint ein neues Peripheriegerät unter dem Pfad /dev/ttyUSBX.

```

1     serport = serial.Serial("/dev/ttyUSB0", baudrate)
2     serdata = serport.read(size)

```

### 5.4 Fast Fourier Transformation

Die Fast Fourier Transformation ist eine schnelle Implementierung der Diskrete Fourier Transformation. Sie wird durch die Arm Bibliothek CMSIS-DSP zur Verfügung gestellt. Da die MCU über keine Floating Point Unit verfügt, ist eine Berechnung mit Fließkommazahlen in Echtzeit nicht realisierbar. Die Float Werte werden in das Festkommazahlenformat Q31 umgewandelt, danach wird die Radix4 FFT durchgeführt. Anschließend werden die komplexen Beträge errechnet, der Index der Frequenz mit dem höchsten Peak gesucht und zurückgegeben.

### 5.5 Fenster Funktion

Da die Radarsignale nicht periodisch sind, stimmen Anfang- und Endwert eines abgetasteten Intervalls nicht überein und können zu einer Verfälschung des FFT Ergebnisses führen. Aus diesem Grund werden verschiedene Fensterfunktionen wie Hamming, Hanning und Blackman genutzt. Die Hanning-Fensterfunktion wird folgenderweise beschrieben:

$$w(n) = \frac{1}{2} \left[ 1 - \cos \left( \frac{2 \cdot \pi n}{N} \right) \right] \quad (5.1)$$

$N$  bestimmt dabei die Anzahl der abgetasteten Werte.

Die FFT Eingabewerte werden zuerst mit einer Fensterfunktion multipliziert bevor der Fourier Transform durchgeführt wird. Dadurch werden die Anfangs- und Endwerte weniger gewichtet und somit die Verfälschung des Signalspektrums minimiert. (siehe [Ludloff](#) (S. 7-68)).

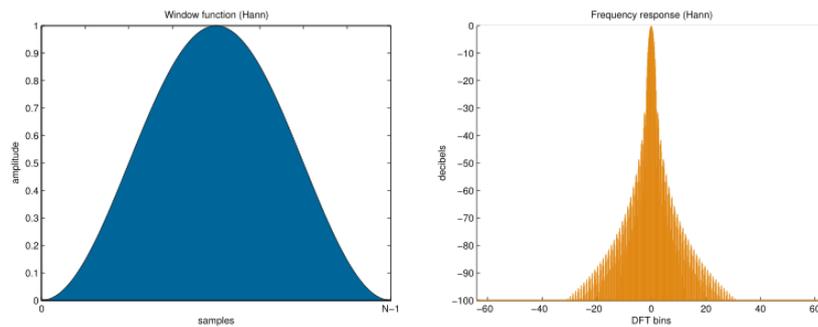


Abbildung 5.3: Hanningwindow Quelle: Wikipedia

## 5.6 Bewegungserkennung

Durch die Verwendung der Fourier Transformation kann die Empfindlichkeit des Systems erhöht werden. Die Radarsensorwerte werden ausgelesen und durch den Fourier Transform vom Zeitbereich in den Frequenzbereich geführt. Aus den daraus errechneten Werten wird, mit Hilfe eines einfachen Cell-averaging CFAR (Constant False Alarm Rate) Algorithmus (Abbildung 5.4), eine adaptive Schwelle gebildet. Dadurch soll die Falschmeldewahrscheinlichkeit bei schlechten Wetterverhältnissen gering gehalten werden. Eine Detektion findet statt, wenn dieser Schwellenwert überschritten wird (vgl. [Ludloff](#) (S. 9-16)).

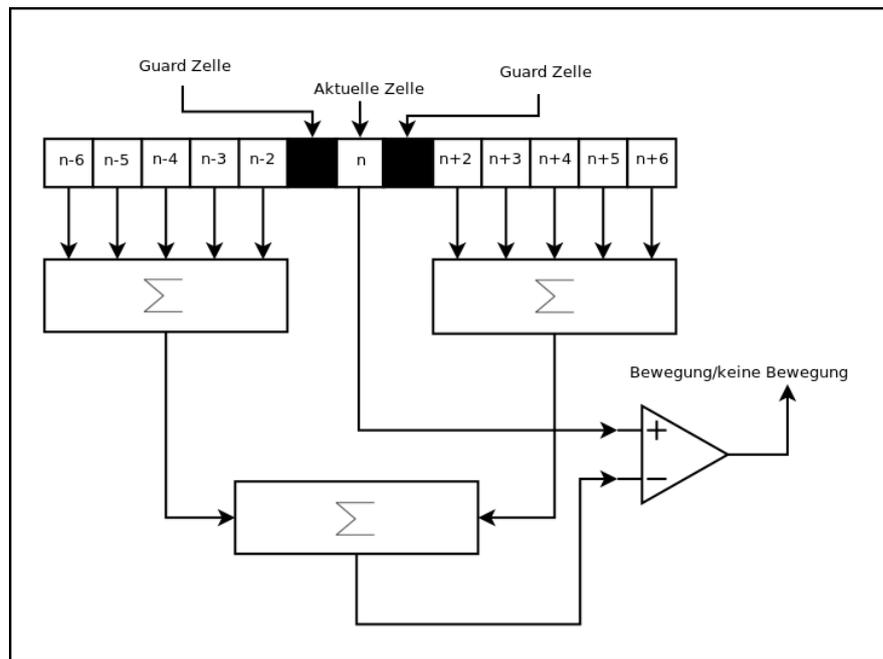


Abbildung 5.4: Cfar: Die Guard-Zellen werden bei der Mittelwertbildung nicht einbezogen

## 5.7 Python Telemetrie

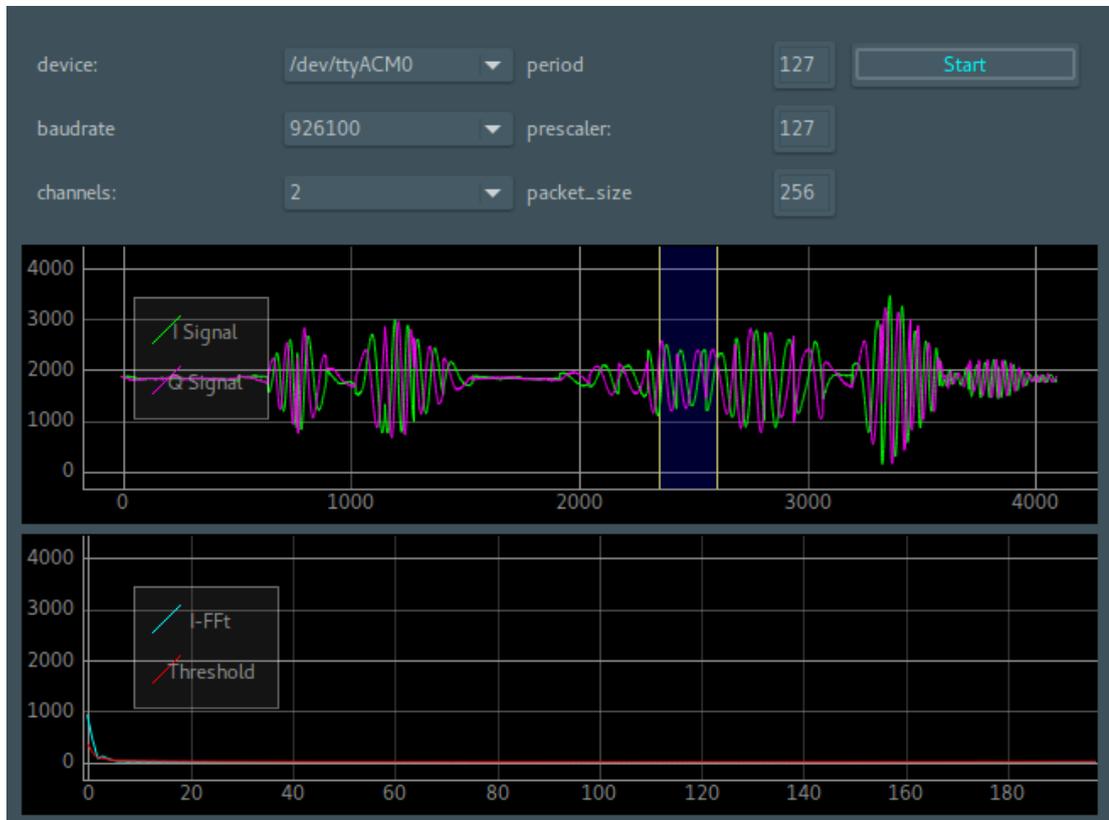


Abbildung 5.5: Telemetrie Anwendung mit angeschlossenem Stereo Radar Sensor

Die Python Anwendung läuft in zwei Threads. Für die Thread Synchronisation kommt das Python Threading Module zum Einsatz. Ein Erzeuger Thread übernimmt das Auslesen der Daten aus der seriellen Schnittstelle und legt diese in eine Queue der Klasse Deque ab. Die Deque Klasse ist wie die Tests im listing 5.2 belegen, wesentlich schneller, als die native Python List. Ein Verbraucher Gui Thread liest dann die Daten aus und zeigt sie in einem Plot Widget an.

Außerdem ist es damit auch möglich die Frequenz des ADCs, die Größe der Datenpakete und die Anzahl der ADC Kanäle während des Betriebs einzustellen. Die Kommunikation mit dem Microcontroller basiert auf einem einfachen binären Protokoll.

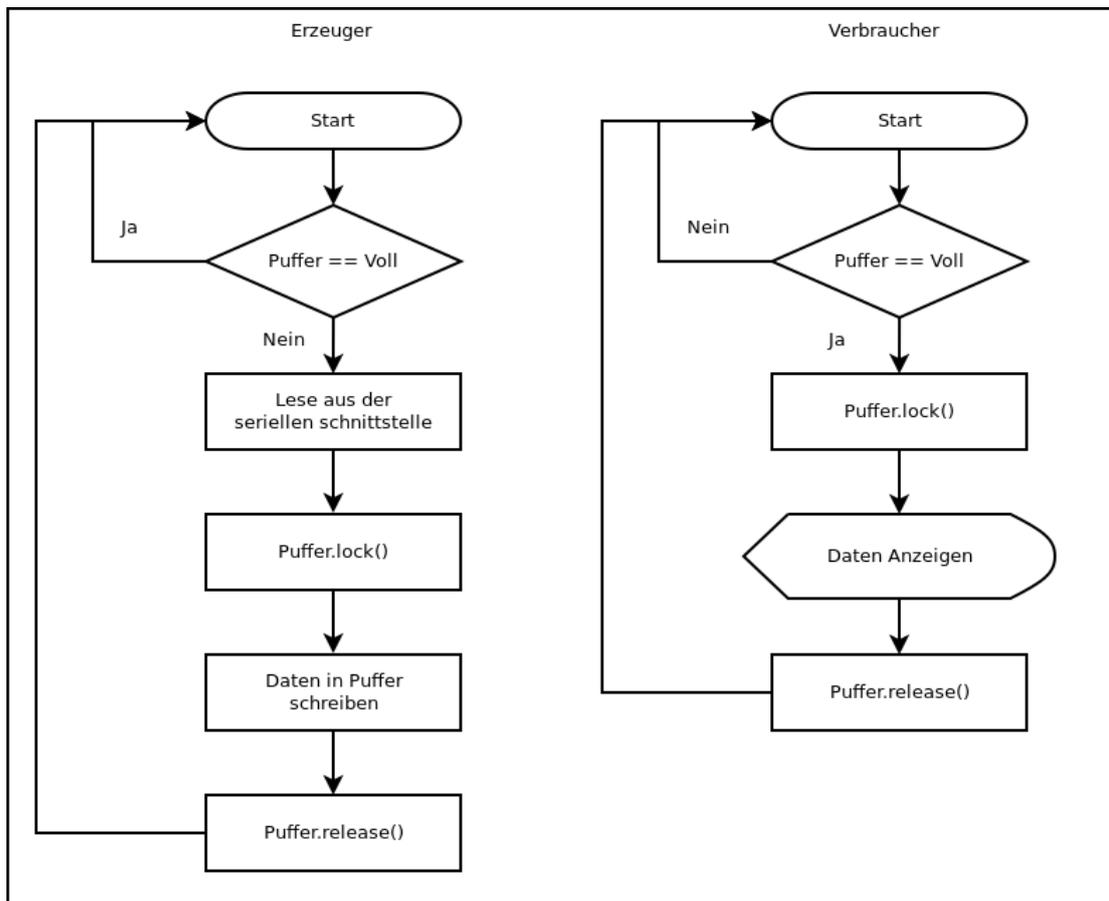


Abbildung 5.6: Flussdiagramm der Telemtrie Anwendung

Python Datentypen sind dynamisch, was zu einigen Problemen bei der Datenübertragung über die seriellen Schnittstelle führen kann. Integer Typen benötigen acht Bytes bei einem 64 Bit Prozessor. Um auch uint8\_t und uint16\_t Datentypen übertragen zu können, werden die Python Integer mit Hilfe der to\_bytes Funktion konvertiert. Diese Funktion nimmt die Größe in Bytes und die Endianess als Parameter. Um die konvertierten Daten nicht einzeln übertragen zu müssen, werden sie mit der Funktion Struct.pack() des Struct Moduls in eine C Struktur verpackt und anschließend versendet.

```

1 python -mtimeit -s
2 'import collections' -s 'c = collections.deque()' 'c.appendleft(1)'
3 10000000 loops, best of 3: 0.116 usec per loop
4
  
```

```
5 python -mtimeit -s 'c = []' 'c.insert(0, 1)'  
6 100000 loops, best of 3: 36.4 usec per loop
```

Listing 5.2: Dequevslist

```
1 struct __attribute__((__packed__)) cmd_setting {  
2     uint8_t cmd;  
3     uint8_t num_channels;  
4     uint16_t size;  
5     uint16_t period;  
6     uint16_t prescaler;  
7     uint32_t baudrate;  
8 } cmd_settings;
```

Listing 5.3: Struktur für die Einstellung der Parameter des ADC's

## 6 Tests

### 6.1 MCU Auslastung

Um die Laufzeit der einzelnen Schritte des Microcontollers zu ermitteln, wurde ein Logic Analyzer zur Hilfe genommen. Die Software Pulseview, die als Frontend für die Sigrok Bibliothek dient, erlaubt die Messung der Laufzeiten. Dazu wird ein GPIO Pin vor dem Aufruf der jeweiligen Routine auf high gesetzt und unmittelbar danach wieder auf low runter gesetzt. Durch das Einfügen einer langen Verzögerung nach jedem Schleifendurchlauf können die Einzelnen Schleifendurchläufe unterschieden werden.

Tabelle 6.1: Auslastung des ARMs

|     |         |     |     |
|-----|---------|-----|-----|
| FFT | 22.6 ms | 200 | 200 |
| ADC | 43,6 ms | 200 | 200 |

### 6.2 Stromverbrauch

Der Stromverbrauch wurde mit dem Monsoon PowerMonitor gemessen. Diese Kombination aus Hard- und Software kann als Stromquelle dienen, und gleichzeitig den Verbrauch messen. Mit der zugehörigen Software PowerTool kann der Energieverbrauch graphisch dargestellt werden.

#### **Stromverbrauch des Sensorsystems mit IPM165**

Aus der Abbildung 6.1 ist zu entnehmen dass der Stromverbrauch des Sensorsystems mit angeschlossenem IPM165 bei 41 mA liegt.

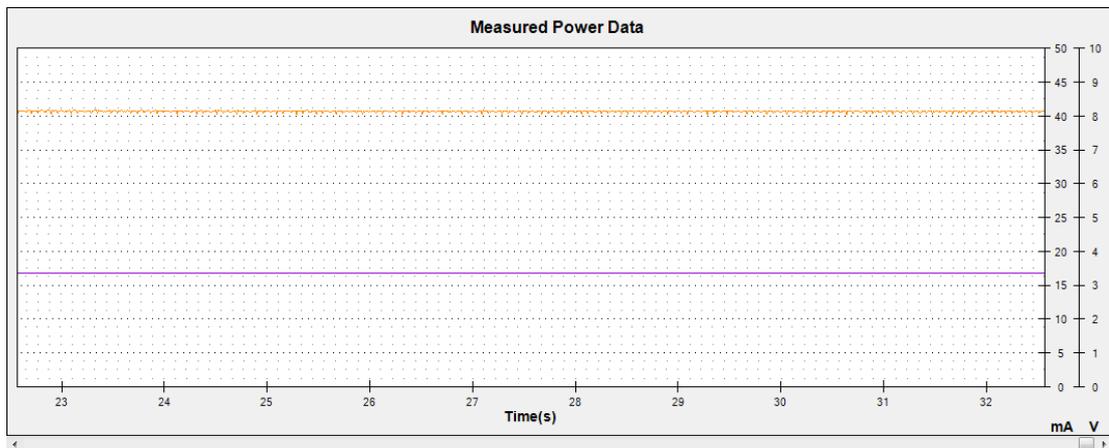


Abbildung 6.1: Verbrauch des IPM 165

### **Stromverbrauch des Sensorsystems mit KCM1\_LP**

Bei dem KMC1\_LP ist der Verbrauch bei 26 mA. Dieser Sensor verbraucht 15 mA weniger als der IPM165. Das ist damit zu begründen, dass der KMC1\_LP intern getaktet wird.

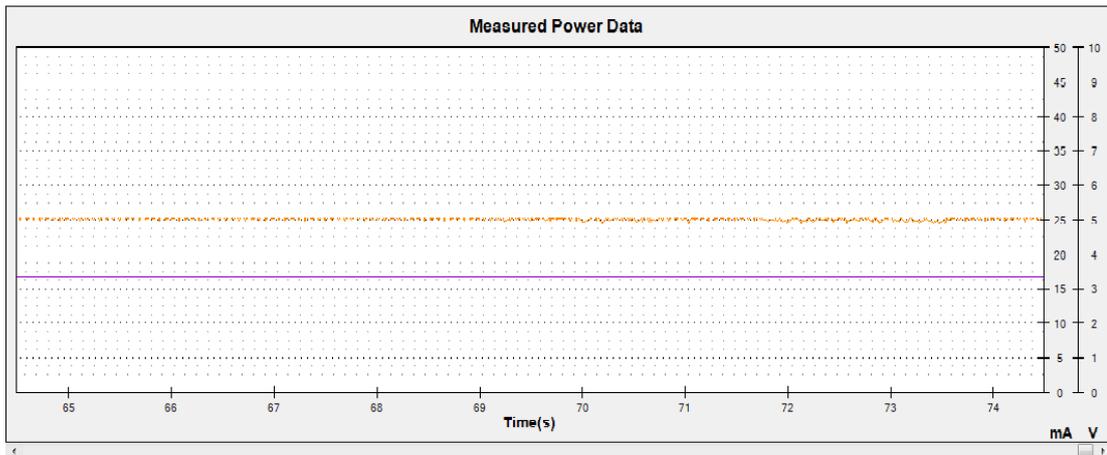


Abbildung 6.2: Verbrauch des KCM1\_LP

## 6.3 Reichweitentest

### Testbedingungen

Der Test für die Bewegungserkennung findet im freien Raum auf einem privaten Gelände statt (Abbildung 6.3). Für die Markierung der Abstände wird ein 30 Meter Maßband, Verkehrshütchen und Kreide benutzt. Bei der Durchführung dieses Tests werden zwei Personen benötigt. Eine Person bewegt sich zu den Markierungen und die Andere hält die Ergebnisse fest. Die zu detektierende Person ist in diesem Fall ca 1.70 m groß von nomaler Statur.

### Testaufbau

Zunächst wird das Feld markiert. Die Verkehrs-hütchen werden in 10 Meter Abstände aufgestellt und in 5 Meter Abständen Kreidemarkierungen gesetzt. Der KCM1\_LP Sensor wird an die entwickelte Platine angeschlossen und in Richtung der Hütchen ausgerichtet. An einem der Platinen GPIO Ausgänge wird eine LED angeschlossen, die eine Bewegung anzeigt. Zusätzlich wird über die Telemtrie Anwendung der Signalverlauf des ADC's verfolgt.



Abbildung 6.3: Testgelände

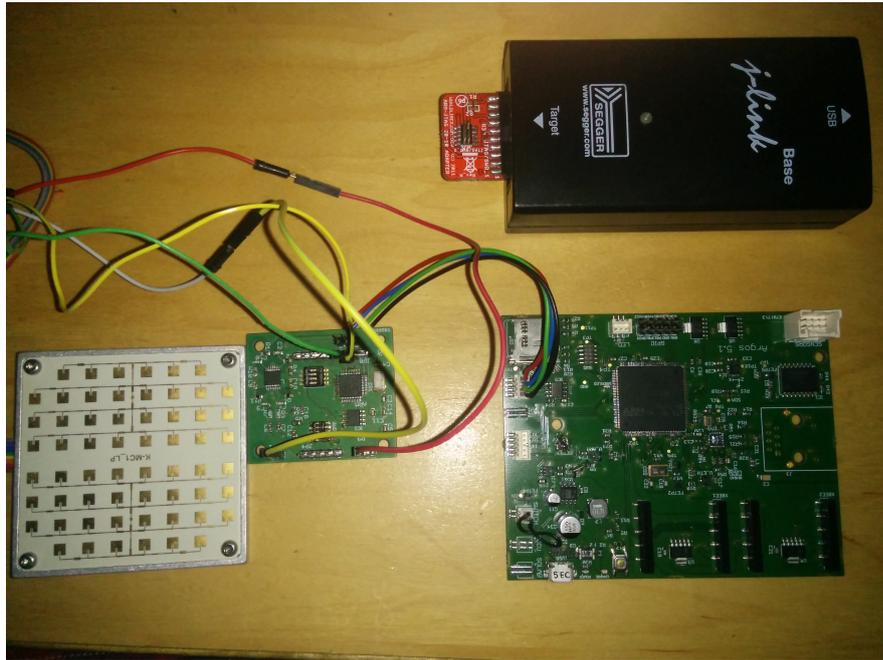


Abbildung 6.4: testhardware: Links der Radarsensor, in der Mitte befindet sich die entwickelte Platine, Rechts unten Die AdHoc Platine und oben der Segger J-Link

### Testdurchführung

Nachdem das System bereit ist, entfernt sich die Zielperson entlang der Markierungen vom Sensor. Nach dem Erreichen einer Markierung wird die Zielperson gebeten sich in einem Abstand von 1 bis 2 Metern um die Markierung herum zu bewegen. Ist ein Leuchten der LED festzustellen wird ein Screenshot des Signalverlaufs gemacht und ein Ja in einer Detektionstabelle geschrieben.

## Testergebnisse

Tabelle 6.2: Reichweitentest

| Abstand in m | Test 1 | Test 2 | Test 3 | Test 4 |
|--------------|--------|--------|--------|--------|
| 5            | Ja     | Ja     | Ja     | Ja     |
| 10           | Ja     | Ja     | Ja     | Ja     |
| 15           | Ja     | Ja     | Ja     | Ja     |
| 20           | Ja     | Ja     | Ja     | Ja     |
| 25           | Ja     | Ja     | Ja     | Ja     |
| 30           | Ja     | Ja     | Ja     | Ja     |
| 35           | Nein   | Ja     | Nein   | Nein   |

Aus der Tabelle 6.2 ist zu erkennen, dass ab einer Entfernung von 30 Metern keine zuverlässige Erkennung mehr erfolgt. Dies ist auch auf der Abbildung 6.7 zu sehen da keine Ausschläge mehr zu erkennen sind. Im Vergleich sind bei einem 15 Meter Abstand in Abbildung 6.5 immer noch kleine Ausschläge zu sehen. Die Abbildungen 6.6 und 6.8 zeigen die zugehörigen Fourier Transform mit CFAR threshold.

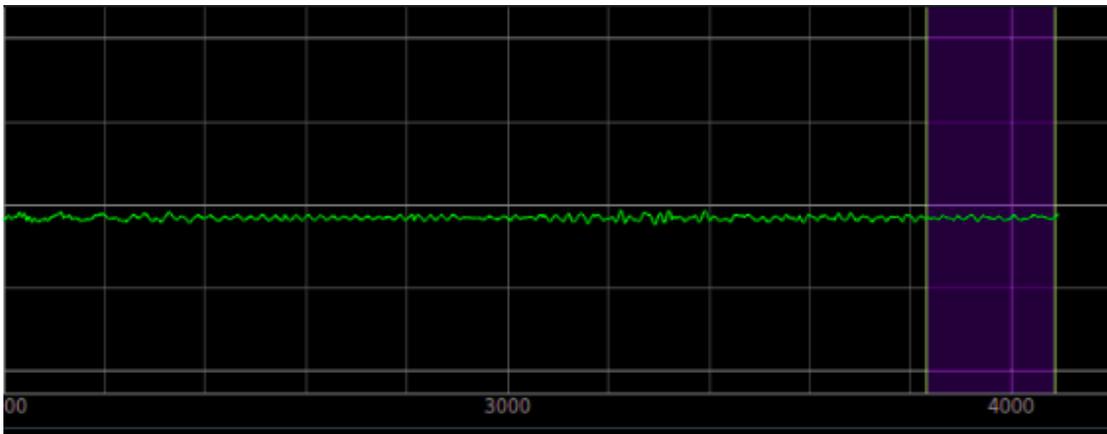


Abbildung 6.5: Radarsignal in einem 15 m Abstand

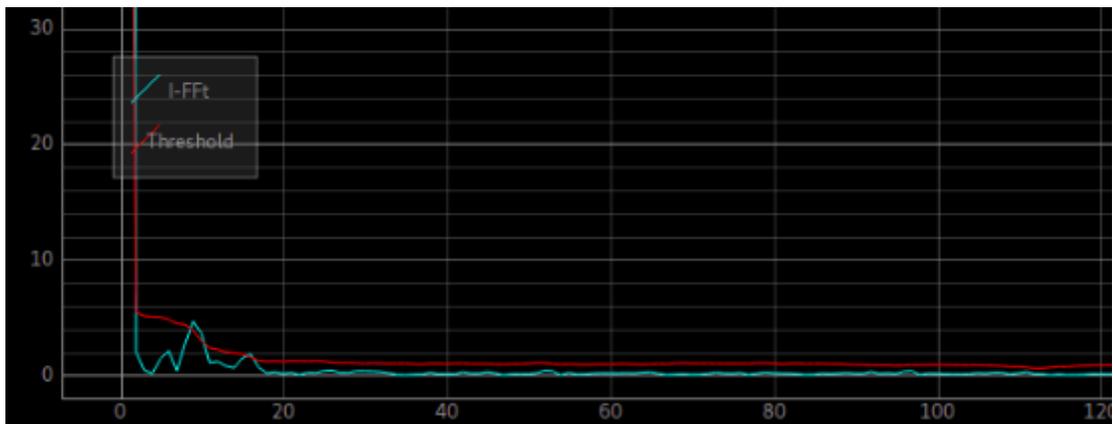


Abbildung 6.6: Fourier Transform und CFAR schwelle bei 15 m

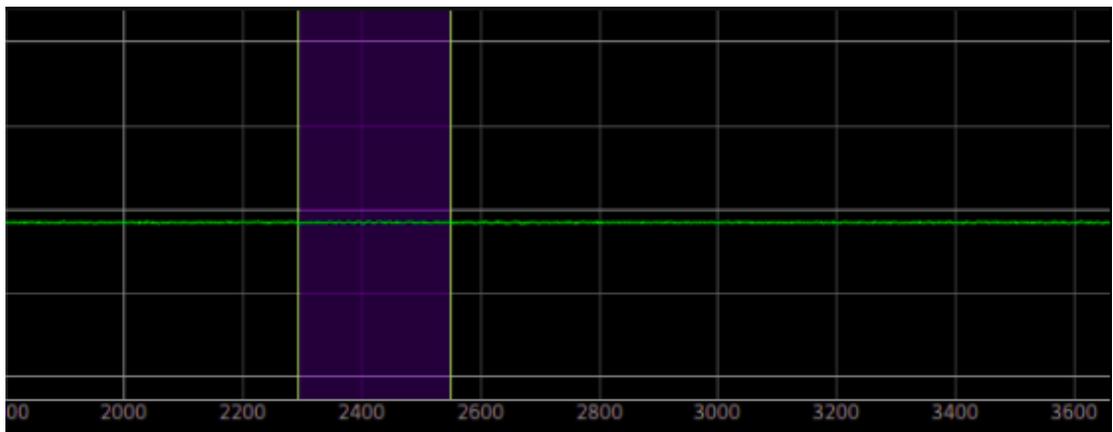


Abbildung 6.7: Radarsignal in einem 30 m Abstand

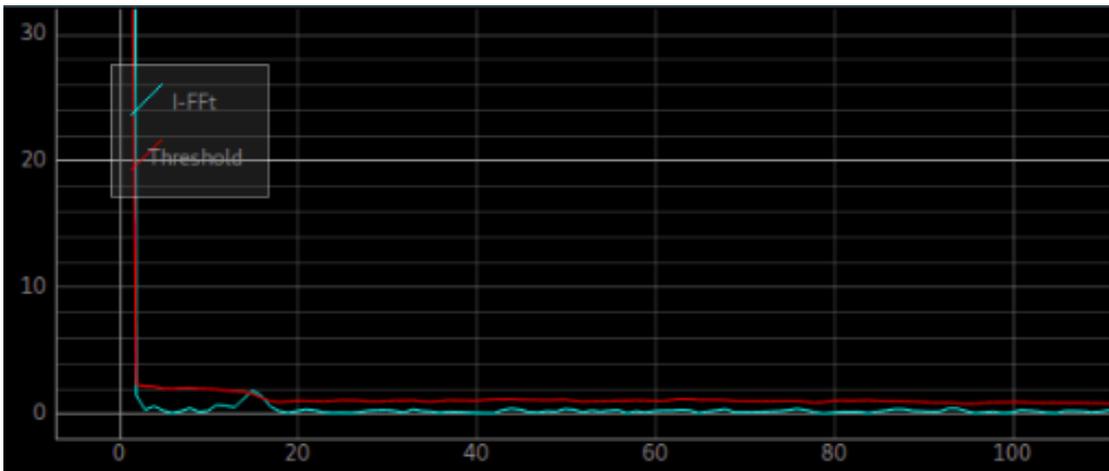


Abbildung 6.8: Fourier Transform und CFAR Schwelle bei 30 m

# 7 Zusammenfassung

## 7.1 Gesammelte Erfahrungen

Während der gesamten Entwicklungszeit wurden neue Erkenntnisse im Bereich der Radartechnik und Signalverarbeitung gewonnen. Des Weiteren konnten Erfahrungen mit der Schaltungsdesign-Software Eagle gesammelt. Fertigkeiten bei der Störungssuche und deren Fehlerbehebung im Zusammenhang mit der Inbetriebnahme der fertigbestückten Platinen konnten angeeignet werden. Zusätzlich haben sich die gelernten Grundlagen der Elektro- und Meßtechnik verfestigt und konnten praktisch umgesetzt werden.

## 7.2 Evaluierung des entwickelten Systems

Im Rahmen dieser Arbeit wurde eine stromsparsame, mikrocontrollergesteuerte Platine entwickelt, die die Signale eines Radarsensors auswertet und eine, über ein Interface angeschlossene Überwachungseinheit benachrichtigt, falls eine Bewegung detektiert wird. Zu Beginn der Arbeit wurden zwei wichtige Anforderungen an das Gesamtsystem gestellt: Zum einen die Energieeffizienz und zum anderen die Erlangung einer großen Reichweite bei der Bewegungserkennung. Die Tests zeigten, dass durch die ausgewählten energieeffizienten Komponenten und den stromsparsamen K-MC1\_LP Sensor, das Sensorsystem im Betrieb 26 mA verbraucht. Dies entspricht einer Gesamtleistung von 85.8 mW. Die Reichweitentests ergaben eine zuverlässige Bewegungserkennung in einem Bereich von 0 bis 30 Metern. Dies ist wesentlich unter den angestrebten 50 Metern, die im Datenblatt des Sensors angegeben sind. Ein Grund dafür könnte die 3,3 V Versorgungsspannung sein, da die Sensoren für Spannungen von bis zu 6 V ausgelegt sind. Das Sensorsystem wird zukünftig bei AdHoc Networks weiterentwickelt und Teile des AdHoc Sensormoduls auf das entwickelte System verlagert.

# Literaturverzeichnis

- [GmbH a] GMBH, RFbeam M.: Application Note AN-04. . – URL [www.rfbeam.ch](http://www.rfbeam.ch)
- [GmbH b] GMBH, RFbeam M.: RFbeam. . – URL [www.rfbeam.ch](http://www.rfbeam.ch)
- [GNU ] GNU: Gnu Compiler Collecrction. . – URL <http://gcc.gnu.org>
- [Goebel 2003] GOEBEL: Radartechnik Grundlagen und Anwendungen. (2003). ISBN 978-3-8007-3141-1
- [InnoSenT ] INNOSENT: Datenblatt IPM165. . – URL [http://www.innosent.de/fileadmin/media/dokumente/DATASHEETS\\_2016/Datenblatt\\_IPM-165\\_V8.5.pdf](http://www.innosent.de/fileadmin/media/dokumente/DATASHEETS_2016/Datenblatt_IPM-165_V8.5.pdf)
- [Ludloff ] LUDLOFF, Albrecht: Praxis Wissen Radar und Radarsignalverarbeitung. ISBN 978-3-8348-0597-3
- [Meyer ] MEYER, Martin: Signalverarbeitung. ISBN 978-3-8348-0597-3
- [Microelectronic ] MICROELECTRONIC, ST: Getting started with stm32f0xx Hardware development. . – URL [http://www.st.com/content/ccc/resource/technical/document/application\\_note/c9/19/d7/b8/6b/0e/4c/d3/DM00051986.pdf/files/DM00051986.pdf/jcr:content/translations/en.DM00051986.pdf](http://www.st.com/content/ccc/resource/technical/document/application_note/c9/19/d7/b8/6b/0e/4c/d3/DM00051986.pdf/files/DM00051986.pdf/jcr:content/translations/en.DM00051986.pdf)
- [mikrocontroller.net ] MIKROCONTROLLER.NET: Schmitt Trigger. . – URL <https://www.mikrocontroller.net/articles/Schmitt-Trigger>
- [MURATA ] MURATA, Ultraschall S.: MURATA Ultraschall Sensor. . – URL [http://www.produktinfo.conrad.com/datenblaetter/500000-524999/506195-da-01-en-ULTRASCHALL\\_SENSOR\\_MURATA\\_MA40S4S.pdf](http://www.produktinfo.conrad.com/datenblaetter/500000-524999/506195-da-01-en-ULTRASCHALL_SENSOR_MURATA_MA40S4S.pdf)
- [Pyqtgraph ] PYQTGRAPH: Pyqtgraph. . – URL <http://www.pyqtgraph.org>

[RFbeam ] RFBEAM: Product Overview. . - URL [http://www.rfbeam.ch/fileadmin/downloads/appnotes/Product%20Overview\\_2014-11.pdf](http://www.rfbeam.ch/fileadmin/downloads/appnotes/Product%20Overview_2014-11.pdf)

[TI a] TI: CAN Tranceiver. . - URL <http://www.ti.com/lit/ds/symlink/sn65hvd233.pdf>

[TI b] TI: Operationsverstaerker. . - URL <http://www.ti.com/lit/gpn/lmv774>

[Weidmann ] WEIDMANN: Radarsensorik. ISBN 978-3-89754-411-6

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 17. September 2016

---

Mehdi Kaf