

# ULTRASCHALL ZUR MESSUNG VON VOLUMENSTRÖMEN

---

## Überprüfung der Nutzbarkeit, Genauigkeit und Kosten einer Ultraschall gestützten Messmethode

---

Bach David

# Inhaltsverzeichnis

<b>1 Messprinzip</b>	<b>1</b>
1.1 Physik . . . . .	1
1.2 Messverfahren . . . . .	2
1.2.1 Ultraschallwandler . . . . .	2
1.2.2 Signalverarbeitung . . . . .	3
1.2.3 Mikrocontroller . . . . .	3
1.2.4 Aufbau . . . . .	3
<b>2 Signalverarbeitung</b>	<b>4</b>
2.1 Analog . . . . .	4
2.2 Digital . . . . .	6
<b>3 Zeitmessung</b>	<b>9</b>
3.1 Bestimmung von Nulldurchgängen . . . . .	9
3.2 Bestimmung des Beginns . . . . .	10
3.3 Statistik . . . . .	10
<b>4 Mikrocontroller</b>	<b>12</b>
4.1 Einführung Mikrocontroller . . . . .	12
4.2 Pinout . . . . .	13
4.3 Methoden . . . . .	13
<b>5 Analyse der Daten</b>	<b>17</b>
5.1 Prüfling . . . . .	17
5.2 Messungen . . . . .	18
5.3 Problematiken . . . . .	18
5.3.1 Übertragung Rohr . . . . .	19
5.3.2 Signalstärke . . . . .	20
<b>6 Theoretische Abhandlung</b>	<b>22</b>
<b>7 Kostenschätzung</b>	<b>24</b>
<b>8 Ergebnis</b>	<b>25</b>
<b>9 Ausblick</b>	<b>26</b>
<b>10 Anhang</b>	<b>27</b>
<b>11 Quellen</b>	<b>28</b>

# Abbildungsverzeichnis

1.1	Veranschaulichung des Messprinzips . . . . .	1
1.2	Die verwendeten Ultraschall-Wandler . . . . .	3
1.3	Aufbau der Versuche . . . . .	4
1.4	Sensorik . . . . .	4
2.1	Schaubild der Platine . . . . .	4
2.2	Schaltplan der analogen Signalverarbeitung . . . . .	5
2.3	Rohes Signal, aufgenommen durch A/D-Wandler . . . . .	6
2.4	Signal nach digitalem matched Filter . . . . .	7
2.5	Berechnung der Hüllkurve . . . . .	7
3.1	Berechnung des Eingangs des Signals . . . . .	9
3.2	Interpolation . . . . .	10
4.1	Messung Aussen Koppelung . . . . .	13
4.2	Messung Aussen Entkoppelt . . . . .	13
5.1	Prüfling . . . . .	17
5.2	Sensoren montiert . . . . .	17
5.3	Messwerte . . . . .	18
5.4	Histogramm . . . . .	18
5.5	Prüfling Aussen . . . . .	19
5.6	Messung Aussen Koppelung . . . . .	19
5.7	Messung Aussen Entkoppelt . . . . .	19
5.8	Signal in 3 cm Distanz . . . . .	20
5.9	Signal in 16 cm Distanz . . . . .	20
5.10	Signal in 35 cm Distanz . . . . .	20
5.11	Hüllkurven in 16 cm Abstand . . . . .	21
6.1	Laminare Strömung durch ein Rohr . . . . .	22
10.1	Veranschaulichung des alternativen Messprinzips . . . . .	27

## Nomenclature

*A/D* Analog Digital

*MC* Mikrocontroller

*Periode* Wert für die Obergrenze des Timers

*Prescaler* Divident des Taktes zur Erzeugung der Timer Frequenz

*Pulsbreite* Anteil der Timer Periode an der der Ausgang eingeschaltet ist

*PWM* Pulse Width Modulation

*SNR* Signal to Noise Ratio

*ToF* Time of Flight

*US* Ultraschall

*VPP* Voltage Peak to Peak

## Zusammenfassung

Ziel dieser Arbeit ist es, die bisherigen Ergebnisse zur Untersuchung der Nutzbarkeit von Ultraschall Messmethodiken zur Bestimmung von Volumenströmen zu evaluieren. Dabei wird in Tiefe auf alle relevanten Details die den momentanen Stand auszeichnen eingegangen. Die Kosten für die Sensorik und Elektronik hat einen Punkt erreicht, bei welchem die Untersuchung dieser Technologie interessant geworden ist. Das Auftreten von Lösungen von Mitbewerbern, zum Beispiel [lindab], verstärkt diesen Eindruck. Dabei gilt es vor allem folgende Fragen zu beantworten:

- Mit welchen Kosten kann man für eine solche Lösung rechnen?;
- Welche Genauigkeit und Präzision lässt sich erreichen?;
- Wie robust ist die Bestimmung gegenüber Turbulenzen und Wirbeln?;

Um dies zu tun, wird im folgenden auf die fundamentalen physikalischen Prinzipien, die Signalverarbeitung, die Programmierung und Logik des Controllers sowie einer theoretischen Abhandlung über das Messprinzip eingegangen. Unterstützt wird dies mit Messungen und Versuchsergebnissen der ersten Prototypen.

Abschliessend wird zur Fortführung des Projekten ein Dokument zur Verfügung gestellt, welches das Einrichten der Entwicklungsumgebung und Starten von Versuchsreihen erklärt.

# 1 Messprinzip

## 1.1 Physik

Das hier verwendete Prinzip zur Bestimmung des Luftstroms basiert auf einer ToF (TimeofFlight) Messung zwischen Senden eines US-Signals und Empfangen desselben. Ist die Strecke zwischen den Sensoren bekannt, beziehungsweise wird die Methodik kalibriert, so kann aufgrund von Abweichungen von der Ausgangszeit ein Wert für die Geschwindigkeit der sich bewegenden Luft bestimmt werden. Folgende Abbildung zeigt das Konzept:

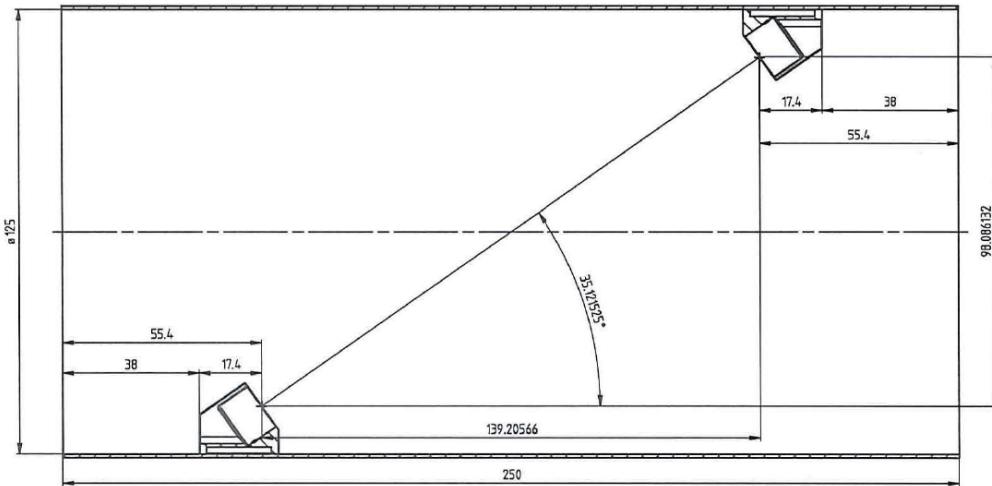


Abbildung 1.1: Veranschaulichung des Messprinzips

Über folgende Formeln kann nun ein Zusammenhang zwischen der gemessenen Zeit und der Windgeschwindigkeit errechnet werden:

$$T = \frac{S}{|\vec{v}_{Gesamt}|} \quad (1.1)$$

wobei  $|\vec{v}_{Gesamt}| = |\vec{v}_s| + \cos(\alpha) \cdot |\vec{v}_w|$

Mittels folgenden Formeln kann aus dem Zeitverhältnis zwischen ruhender Luftmasse und Luftstrom die mittlere Geschwindigkeit bestimmt werden:

$$|\vec{v}_w| = \frac{\left(\frac{T_0}{T} - 1\right) \cdot |\vec{v}_s|}{\cos(\alpha)} \quad (1.2)$$

Mit:

$v_s$  = Schallgeschwindigkeit

$v_w$  = Windgeschwindigkeit

$T$  = gemessene Zeit

$T_0$  = ruhe Zeit

$\alpha$  = Neigungswinkel

$S = v_s \cdot T_0$  = Strecke zwischen den Sensoren

Die bestimmte Geschwindigkeit  $v_w$  ist dabei ein Mittelwert des Vektorfeldes der realen Verteilung über die Strecke S. Zur Behandlung der Genauigkeit und Korrektheit im Bezug zum realen Strom dieser Messmethode wird im Kapitel [Theoretische Abhandlungen](#) eingegangen.

Aus dieser Grösse kann nun einfach der mittlere Volumenstrom durch das Rohr bestimmt werden:

$$\Phi = A \cdot v_w \quad (1.3)$$

$$A = \text{Querschnittsflaeche} \quad (1.4)$$

Eine Alternative, hier leider nicht realisiert, Messmethode zeigt Abbildung [10.1](#)

Der Vorteil dieses Verfahrens, beruht auf Hin- und Rückmessung und dadurch Elimination der Temperaturabhängigkeit der Schallgeschwindigkeit. Im Anhang sind die Berechnungen dafür durchgeführt.

Weiterhin sei zu erwähnen, dass bei der singulären Richtungsmessung die Referenzzeit der korrekten Zeit entsprechen muss, welche der Schallimpuls benötigt um in Ruhe die Strecke S zurückzulegen. Ansonsten erhält man falsche Werte für die Geschwindigkeit. Das liegt daran, dass die relative Abweichung von Referenzzeit und gemessener Zeit ein Mass für die Geschwindigkeit ist. Bei einer grossen Distanz entspricht eine Differenzzeit von  $1 \mu\text{s}$  einer anderen Geschwindigkeit als bei einer sehr kurzen Distanz. Sobald man aber Hin- und Rückrichtung bestimmt ist die absolute Zeit irrelevant.

## 1.2 Messverfahren

Das zentrale Element der Messung beruht auf einer sehr genauen Messungen der Zeit zwischen Senden und Empfangen des US-Signales.

### 1.2.1 Ultraschallwandler

Die verwendeten Piezowandler strahlen mit einer Frequenz  $f = 40 \text{ kHz}$ . Das Anlegen einer Wechselspannung mit exakt dieser Frequenz sorgt dafür das der US-Wandler, abhängig von der VPP der Spannung, Schallwellen kegelförmig abstrahlt. Wird ein weitere Wandler durch eine solche Schallwelle stimuliert, so liegt an seinem Signalausgang eine schwache Wechselspannung an. Die Generation der  $40 \text{ kHz}$  wird von zwei Pins des Mikrocontrollers übernommen. Im Kapitel [4](#) wird das Verfahren erläutert. Hier wichtig zu erwähnen ist nur, dass die Sensoren mit 6 VPP betrieben werden. Die Sensoren selber stammen aus der Autoindustrie und werden hauptsächlich zur Abstandsmessung genutzt. Dort müssen Distanzen im Bereich von 10 - 60 cm vermessen werden.

#### Wichtig:

Die Abstrahlwinkel der Sensoren sind nicht symmetrisch! Dies ist dem Datenblatt der Sensoren zu entnehmen. Dass heisst, dass die Ausrichtung der Sensoren einen Einfluss auf die Messung hat!



Abbildung 1.2: Die verwendeten Ultraschall-Wandler

### 1.2.2 Signalverarbeitung

Das schwache Signal muss zur weiteren Verwertung verstärkt und gefiltert werden. Das Kapitel **Signalverarbeitung** geht auf die verwendete analoge sowie digitale Signalverarbeitung ein.

### 1.2.3 Mikrocontroller

Die Logik wird durch einen Mikrocontroller der Marke ST durchgeführt dieser kümmert sich um folgende Schritte:

- Ansteuern des Senders mittels eines PWMs zur Erzeugung von Schallwellen
- Aufnahme der Wechselspannung des Empfängers mittels eines A/D-Wandlers
- Durchführen der digitalen Signalverarbeitung
- Berechnung der Zeitdauer und/oder Geschwindigkeit des Luftstromes
- Ausgabe der Daten per UART über einen COM Port

Im Kapitel **Mikrocontroller** wird genauer auf das verwendetet Modell sowie die verwendeten Methoden, Funktionen und Bibliotheken eingegangen. Bis dato werden die Daten zur Analyse benutzt, da es sich noch um nur einen Prototypen handelt.

### 1.2.4 Aufbau

Folgende Bilder zeigen den Aufbau zur Messung und die Verbindung der Platinen:



Abbildung 1.3: Aufbau der Versuche

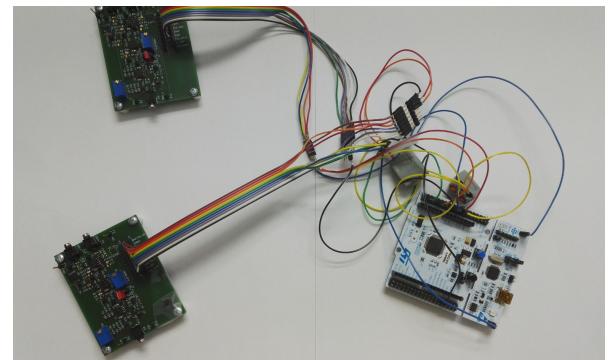


Abbildung 1.4: Sensorik

## 2 Signalverarbeitung

### 2.1 Analog

Die verwendete Platine hat folgendes Erscheinungsbild:



Abbildung 2.1: Schaubild der Platine

Dazu gehört folgender Schaltplan:

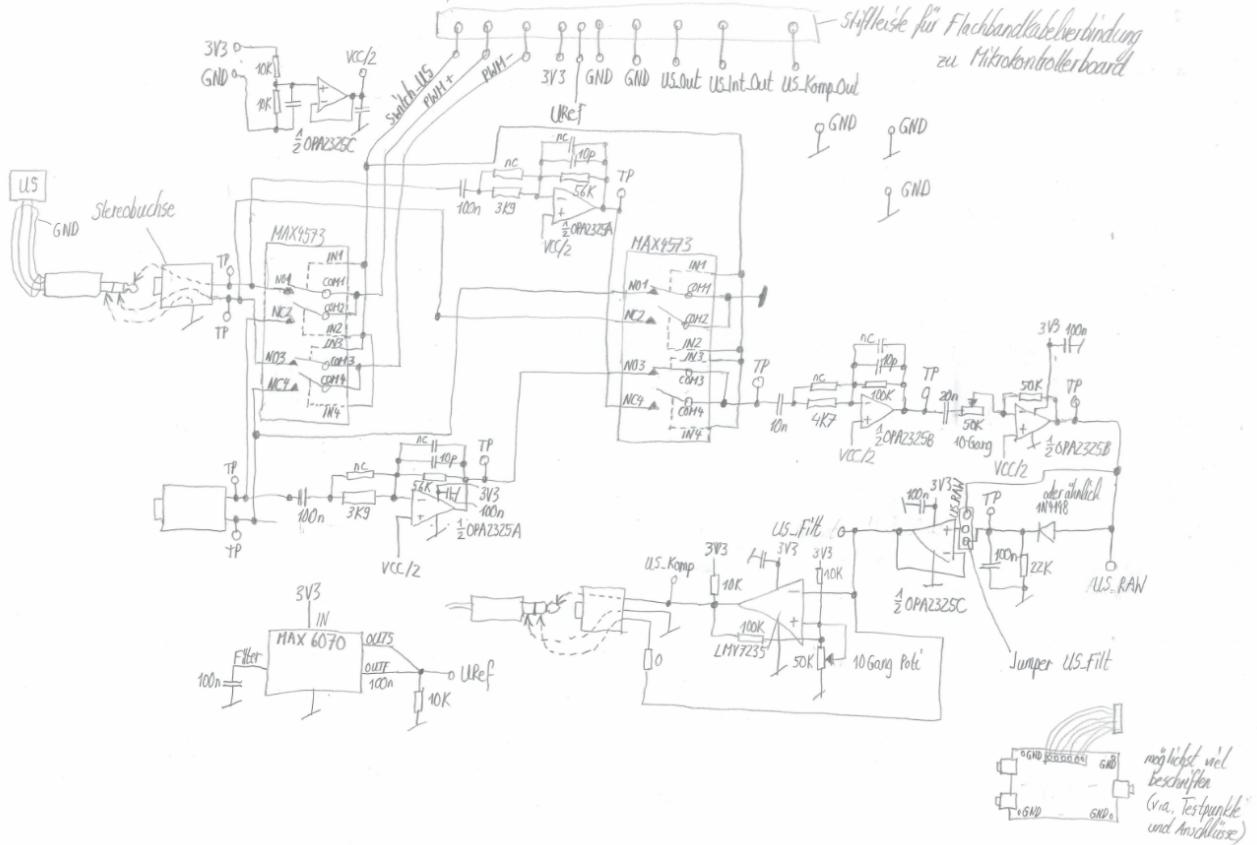


Abbildung 2.2: Schaltplan der analogen Signalverarbeitung

Das Signal wird dabei zuerst per Bandpass auf 40 kHz gefiltert sowie verstrkt, um den Faktor 14. Danach durchluft es einen weiteren identischen Ablauf und wird nun um den Faktor 21 verstrkt. Als nchstes liegt ein Variabler Verstrker an. Zum Ende hin wird das Signal gleichgerichtete und ber einen Komparator geleitet, dessen Sensitivitt variabel einstellbar ist. Uber ein Flachbandkabel knnen an verschiedenen Stellen die Signale abgenommen werden. Fr den hier beschriebene Methodik ist nur der Ausgang *US\_RAW* von Relevanz. Dieser stellt das Analog gefilterte Signal vor der Gleichrichtung dar. Dieses wird auf den A/D-Wandler des Mikrocontrollers gefhrt. Ein vom A/D-Wandler aufgezeichnetes Signal sieht dabei folgendermassen aus:

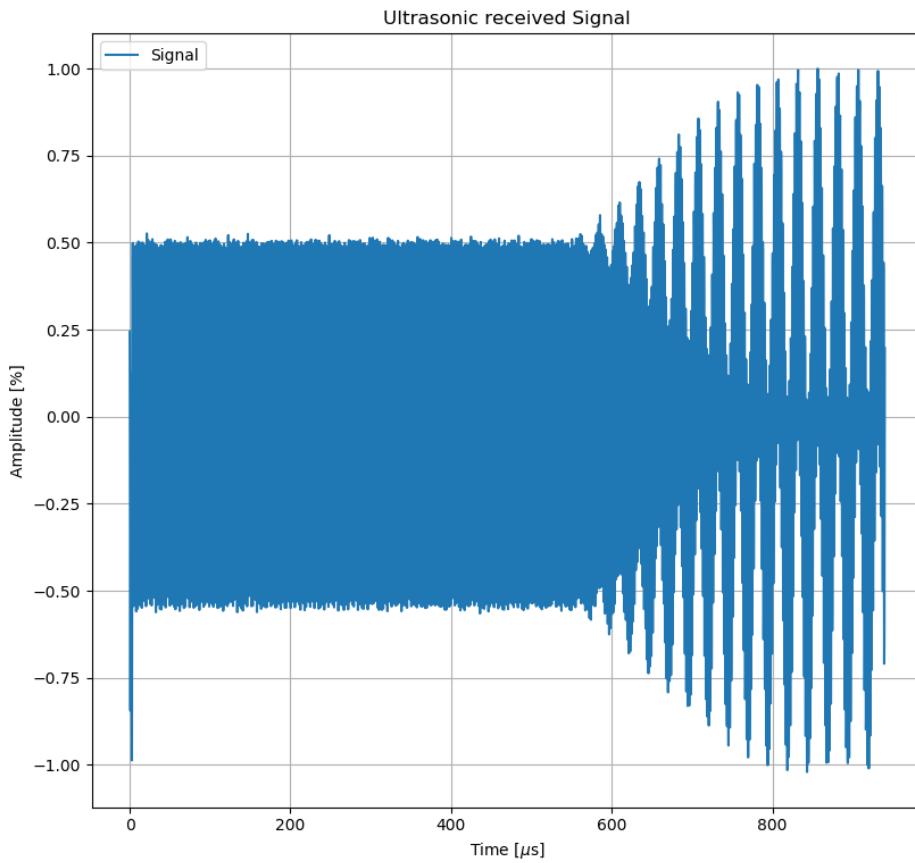


Abbildung 2.3: Rohes Signal, aufgenommen durch A/D-Wandler

Das enorme Rauschen entsteht hauptsächlich aufgrund der gewählten Buchsen zur Übertragung der Spannung der Ultraschall Sensoren zur analogen Verarbeitung vor. Dies kann in Zukunft geändert werden.

## 2.2 Digital

Wie man erkennen kann liegen einige hochfrequente Schwingungen innerhalb der 40 kHz Perioden vor. Um das Signal zu filtern und das Verhältnis von Rauschen zu Signal signifikant zu verbessern werden folgende Methoden benutzt:

### Normalisieren

Zu erst wird das Signal, welches als Spannung in mV vorliegt, per Bildung eines Mittelwertes und Normalisierung auf den maximalen Spannungswert kalibriert.

**Matched Filtern** Nun wird auf dieses Signal ein sogenannter Matched Filter angewendet. Die Funktionsweise sei im folgenden erläutert:

Die Frequenz des Signals ist bekannt und damit wird eine Sinus Periode gebildet. Das Signal sowie diese Sinus-Kurve werden dann korreliert. In einfachen Worten ausgedrückt wird die 'Gleichheit' zwischen Signal und Sinuskurve für alle möglichen Kombinationen der Phase durchgeführt. Das Signal welches dieses maximiert ist das gefilterte Signal.

Mathematisch wird dieses über eine Lineare Faltung realisiert:

$$(f * g)(n) = \sum_{k \in D} f(k)g(n - k) \quad (2.1)$$

Die DSP (Digital Signal Processing) Bibliotheken des verwendeten Mikrocontrollers bieten dafür geeignete Funktionen an, sodass einfach eine Korrelation berechnet werden kann.

Das folgende Signal zeigt die Arbeit eines solchen Filters:

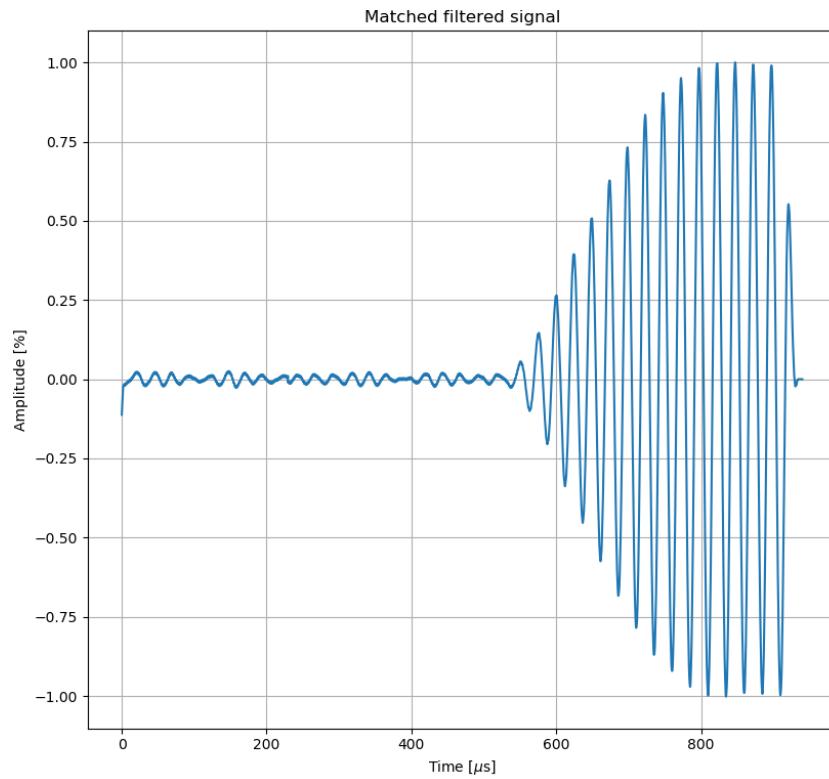


Abbildung 2.4: Signal nach digitalem matched Filter

Zur späteren Bestimmung des Zeitpunktes wenn das Signal den Empfänger erreicht hat, wird eine Hüllkurve des Empfangssignals benötigt. Folgende Abbildung zeigt das Verfahren:

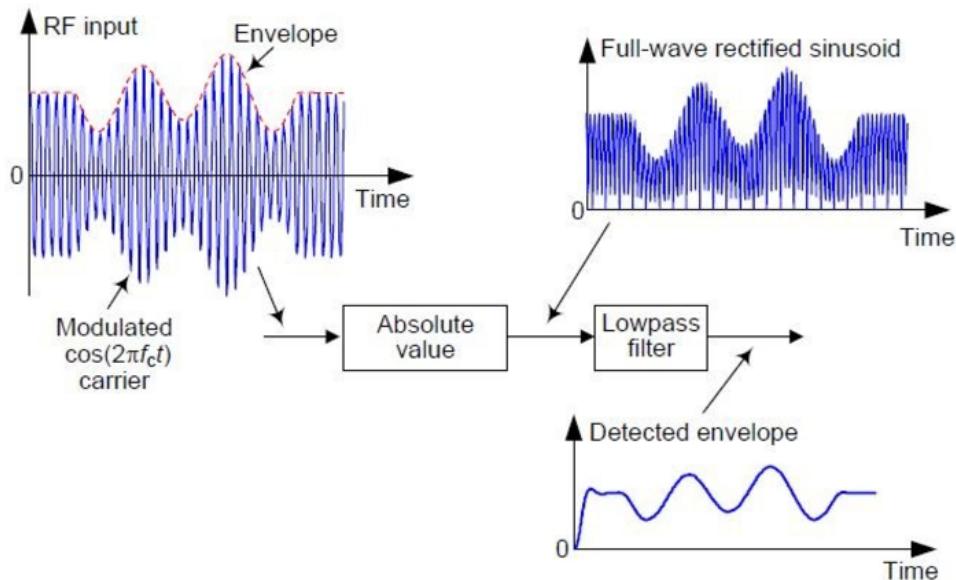


Abbildung 2.5: Berechnung der Hüllkurve

Auch hier bieten die DSP Bibliotheken genügend Auswahl. Das einzige, was getan werden muss ist, die Filterparameter für den Lowpass Filter zu bestimmen. Dazu wird eine Python Methode

verwendet.

Um die Filterparameter zu bestimmen müssen diverse Eigenschaften gesetzt werden:

```
1 sample_rate = 1./(0.188 * pow(10,-6))
2 nsamples = 5000
3 t = np.array(range(len(waves[0]))) * 0.188 * pow(10, -6)
4
5 nyq_rate = sample_rate / 2.0
6 width = nyq_rate/53000000
7 ripple_db = 175
8 N, beta = signal.kaiserord(ripple_db, width)
9 cutoff_hz = 5000
10
11 taps = signal.firwin(N, cutoff_hz/nyq_rate, window=('kaiser', beta))
```

Hier zu sehen ist der Code welcher zur Bestimmung der Filterparameter verwendet wird. Die Parameter `sample_rate` und `nsamples` sind durch die Auflösung des A/D-Wandlers vorgegeben. `nyq_rate` ist die Nyquist Rate des Filter Fensters, `width` stellt die Breite des Übergangs des Filters dar und wurde so bestimmt, dass die Hüllkurve optimal liegt, `ripple_db` ist der Parameter der die Glattheit der Kurve darstellt und auch dieser wurde entsprechend angepasst. Die restlichen Werte berechnen das Filter Fenster und die `taps`. Letzteres dient als Input Parameter für die DSP Filter Funktion des Mikrocontrollers.

### 3 Zeitmessung

Dieses Kapitel beschäftigt sich mit der Bestimmung der verstrichenen Zeitdauer zwischen Senden und Empfangen eines Signals. Es liegt eine analog sowie digital bearbeitet Signalkurve vor. Zur Verdeutlichung zeigt folgende Abbildung eine beispielhafte Analyse:

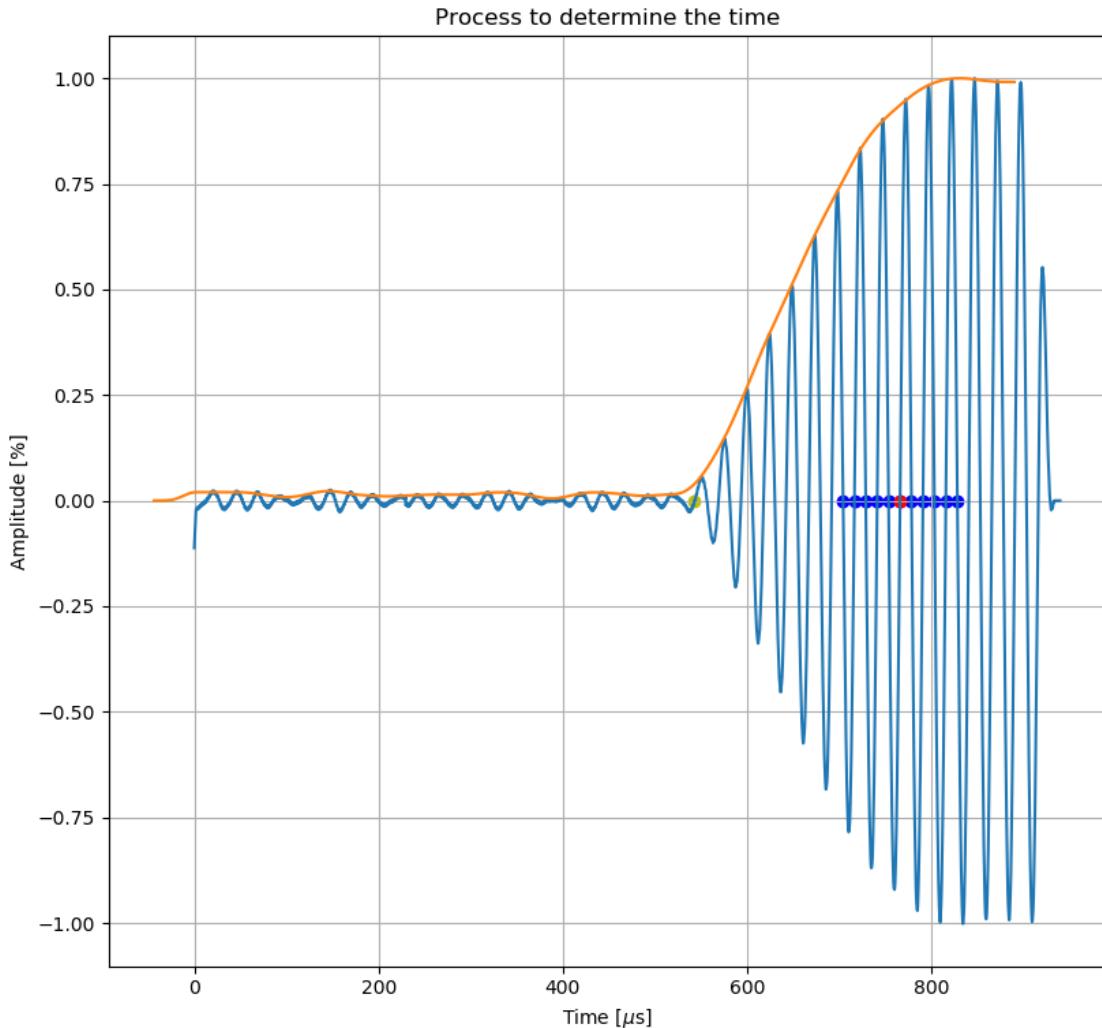


Abbildung 3.1: Berechnung des Eingangs des Signals

Grundsätzlich läuft die Bestimmung folgendermassen ab:

#### 3.1 Bestimmung von Nulldurchgängen

Ab einer festen Amplitude werden die nächsten 11 Nulldurchgänge bestimmt. Dabei wird, um die Limitation der Auflösung des A/D-Wandlers zu umgehen, der exakte Nulldurchgang interpoliert: In der folgenden Darstellung entspricht  $t_{diskret}$  dem Wert bevor ein vorzeichen Wechsel im Signal  $f$  vorliegt und  $t_{diskret+1}$  dem danach liegenden Wert.

$$t_{Nulldurchgang} = t_{diskret} \cdot \frac{|f(t_{diskret})|}{|f(t_{diskret})| + |f(t_{diskret+1})|} \quad (3.1)$$

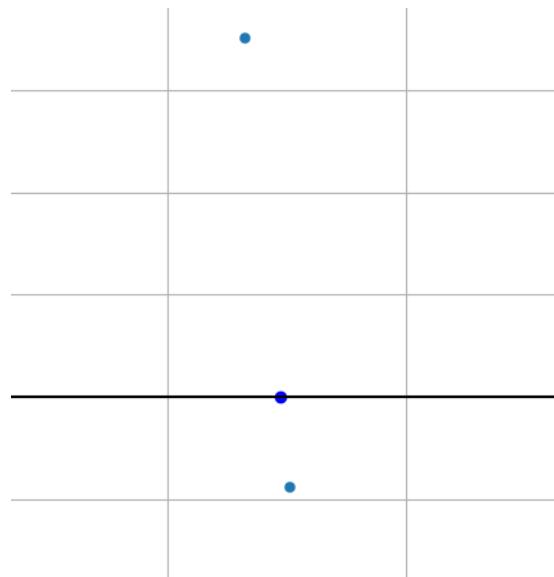


Abbildung 3.2: Interpolation

Abbildung 3.2 verdeutlicht dieses noch einmal: Diese Werte werden dann gemittelt.

## 3.2 Bestimmung des Beginns

Nun gilt es diesem Nulldurchgang die korrekte Ordnung zuzuweisen. Dazu muss der Anfang des Signals bestimmt werden. Nun kommt die Hüllkurve zum Einsatz. Auch in diesem Fall wird ein absoluter Referenzwert gesetzt und der Zeitpunkt bestimmt, wenn die Hüllkurve diesen Wert erreicht hat.

Nun wird die Differenz zwischen Mittelwert der Nulldurchgänge und diesem Beginn gebildet:

$$\Delta_t = \overline{t_{\text{Nulldurchgang}}} - t_{\text{Hüllkurve}} \quad (3.2)$$

Dieser Wert wird durch  $12.5 \mu$  geteilt, dabei handelt es sich um die Breite von einer halben Amplitude einer 40 kHz Welle, um die Ordnung des Nulldurchgangs zu erhalten:

$$N = \frac{\Delta_t}{12.5\mu} \quad (3.3)$$

N wird gerundet zum nächsten Integer und damit ist die Ordnung des Nulldurchgangs bekannt. Aus dem Wert des Nulldurchgangs,  $\overline{t_{\text{Nulldurchgang}}}$ , kann nun der Beginn des Signals errechnet werden:

$$T = \overline{t_{\text{Nulldurchgang}}} - N \cdot 12.5\mu \quad (3.4)$$

Diese Methode ist einem einfachen Amplituden Filter vorzuziehen, da die Signal Kurve bei Luftbewegungen einigen Fluktuationen ausgesetzt ist und dadurch nur suboptimal arbeiten würde. Der Vorteil des hier vorgestellten Verfahrens liegt in seiner Robustheit gegenüber solchen Störungen.

## 3.3 Statistik

Um diese Methode weiterhin robuster gegenüber Störungen zu machen werden mehrere solcher Zeiten kurz hintereinander bestimmt. Diese Liste an Werten wird durch einen Medianfilter geführt welcher die 20% grössten und kleinsten Werte entfernt. Die zurückbleibende Liste wird

gemittelt. Praktisch wird die Liste sortiert und ein Mittelwert der mittleren Werte durchgeführt.

$$\bar{T} = \frac{\sum_{n=2}^5 \begin{bmatrix} T_1 & T_2 & T_3 \\ T_4 & T_5 & T_6 \end{bmatrix}}{\sum_{n=2}^5 n} \quad (3.5)$$

In Kapitel I wurde gezeigt wie aus der Zeit die Stromgeschwindigkeit und dann sukzessive der Volumenstrom berechnet werden kann.

Dabei ist hier anzumerken, dass der Zeitpunkt 0, als der erste vom AD-Wandler digitalisiert Wert definiert wird. Natürlich liegt zwischen dem Starten des Wandlers und dem Starten der PWMs eine gewisse Zeit, abhängig von den Zyklen des Mikrocontrollers. Das bedeutet, dass die gemessene Zeit nicht korrekt ist. Da aber nicht die absolute Zeit, sondern immer Differenzen zwischen Zeiten von Interesse sind, und man annehmen kann, dass der Mikrocontroller immer die gleiche Zeit braucht zwischen dem Starten des Wandlers und dem Starten der PWMs, hat dies keinen Einfluss auf die Genauigkeit und Präzision der Messung.

## 4 Mikrocontroller

Dieses Kapitel wird zuerst kurz auf das Pinout des Controllers eingehen und sich dann umfassend mit der Programmierung selbigen beschäftigen.

### 4.1 Einführung Mikrocontroller

Der hier verwendete Mikrocontroller ist das Modell STM32L476, welcher ausreichend Prozessorleistung durch den M4Cortex liefert und eine umfassende Peripherie besitzt. In diesem Dokument wird nicht auf die Einrichtung einer Entwicklungsumgebung eingegangen. Dafür dient das Getting Started Dokument welches auch im Projekt Ordner zu finden ist. Hier sind Links zu den relevanten Datensätzen:<sup>1</sup> Der Controller wird mittels C++ programmiert und man benötigt eine Entwicklungsumgebung um die Programme zu flashen. Die grundsätzliche Idee ist, dass in der main()-Methodik des Programms eine Endlosschleife läuft welche durchgängig die Messzeit evaluiert.

Hier zum Einsatz kommt folgende Peripherie:

- Timer
- PWM
- A/D-Wandler
- UART

Im folgenden wird kurz auf die einzelnen Elemente eingegangen:

#### Timer:

Timer sind die Allrounder beim Mikrocontroller. Sie können zur schlichten Zeitmessung verwendet werden, zum Timen von Geschehnissen, als Trigger und Interrupt, zur Generation von verschiedenen Signalen (PWM, Output Compare, ...) und so weiter. In diesem Fall werden zwei Timer verwendet. Einer zur Zeitmessung und einer zur Generation eines PWM-Signales [timer].

#### PWM

PWM steht für Pulse Width Modulation und ist ein gängiger Begriff wenn es zu Mikrocontrollern kommt. Grundsätzlich funktioniert eine PWM so: Man generiert per Timer eine gewisse Frequenz, dies geschieht dadurch, dass man den Timer mit dem Prozessor Takt, oder jedem beliebigen kleineren, zu einer gewissen Grenze zählen lässt. Der Mikrocontroller stellt nun Methoden zur Verfügung welche eintreten falls zum Beispiel besagte Grenze erreicht ist. Um nun aus diesem ein PWM zu machen muss man in der Methode (welche hier `PeriodElapsedCallback()` heisst) nur definieren, dass ein bestimmter Pin getoggelt wird. Nun wird der Pin mit besagter Frequenz ein und aus geschaltet. Dies muss in diesem Fall nicht selber gemacht werden, sondern der Timer hat eine eigene Funktion um genau das zu tun. In diesem Fall wird eine leichte Abwandlung dieser Funktion verwendet. Es werden zwei PWMs generiert welche aber invers sind. Dazu muss man vorher eine Pulsbreite definieren. Diese, in Prozent, sagt aus wie lange der Pin getoggelt werden soll im Bezug zur Frequenz. Dass heisst, der erste schaltet bei 0% an und stoppt bei 50%, der andere schaltet einen **anderen** Pin bei 50% an und bei 100% aus. Da beide exakt simultan laufen und beide Pins mit dem Sender verbunden sind, erscheint für ihn ein Wechselspannungssignal mit doppelter Spannung des Pins, in diesem Fall 6 V, und einer Frequenz wie die einzelnen PWMs definiert haben. [pwm]

#### A/D-Wandler

---

<sup>1</sup>Alle gesammelten Datensätze liegen im Ordner 05\_Entwicklung/STM32 - Sheets

Der A/D-Wandler wandelt die analoge Eingangsspannung an einem Pin in einen digitalen Wert um. Es gibt verschiedene Modi wie man diesen Wandler betreiben kann: Im Single-Conversion oder Continous-Conversion Modus. Hier wird zweiteres benutzt. Wichtig zu beachten ist, dass an einen A/D-Wandler nur positive Spannungen in einem gewissen Bereich angelegt werden können. Die maximale Spannung ist durch den Referenzwert des Mikrocontrollers vorgegeben. Die Auflösung des Wandler ist durch den Hersteller vorgegeben. Hier wird ein 12-Bit Wandler verwendet, dies bedeutet er teilt seinen Messbereich in 4096 diskrete Schritte auf. Die Sample Rate, als die Dauer, welche für die Wandlung von einem Wert vorliegt, wird mit einem als **fast** deklarierten Timer mit  $0.188 \mu\text{s}$  angegeben. Dies begrenzt die maximale zeitliche Auflösung. Die aufgenommenen Werte werden per DMA auf den internen RAM geschrieben. Die Grösse dieses Buffers ist grundsätzlich freiwählbar, da aber der RAM begrenzt ist, ist die momentatne Grösse auf 5000 Werte begrenzt was einer Zeitmessung von  $940\mu\text{s}$  entspricht. [ad]

### UART

Das UART ist ein Kommunikationsprotokoll zum Übertragen von Daten mittels einer seriellen Schnittstelle. Diese wird verwendet um die gemessenen und verarbeiteten Signale zu Übertragen. Es kann zum Teil passieren, dass bei der seriellen Übertragung ein Semicolon nicht übertragen wird oder die erstellte Datei anderweitig defekt ist. Dies erkennt man in der Darstellung wenn extrem hohe Werte vorliegen. Diese können manuel entfernt werden.

## 4.2 Pinout

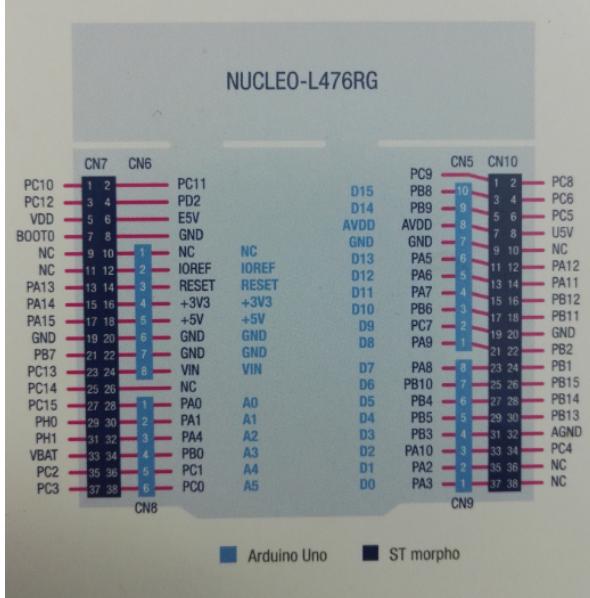


Abbildung 4.1: Messung Aussen Koppelung

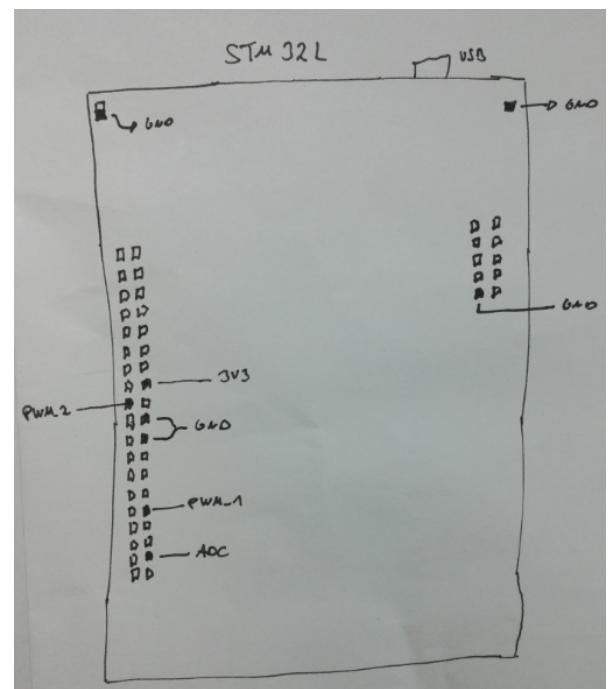


Abbildung 4.2: Messung Aussen Entkoppelt

Abbildung ?? zeigt das Pinout des Controllers. Diese sind intern mehrfach besetzt und können diversen Zwecken dienen.

## 4.3 Methoden

Hier wird nun die `main.cc` - Methode des Controllers in ihrer Vollständigkeit vorgestellt.

```

1 int main(void)
2 {
3     HAL_Init();
4     SystemClock_Config();
5
6     MX_GPIO_Init();
7     MX_DMA_Init();
8     MX_TIM2_Init();
9     MX_USART2_UART_Init();
10    MX_TIM5_Init();
11    MX_ADC1_Init();
12
13 //Hello_World();
14
15 while (1)
16 {
17     for (int i=0; i<5;i++)
18     {
19         time_array[ i ] = 0;
20     }
21
22     for(int i=0; i<5;i++){
23         HAL_Init();
24
25         MX_GPIO_Init();
26         MX_DMA_Init();
27         MX_TIM2_Init();
28         MX_USART2_UART_Init();
29         MX_TIM5_Init();
30         MX_ADC1_Init();
31
32         count = 0;
33         time_increment = 0;
34         timer_value = 0;
35
36         HAL_TIM_Base_Start_IT(&htim5);
37         HAL_ADC_Start_DMA(&hadc1, g_ADCBuffer, ADC_BUFFER_LENGTH);
38
39         HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
40         HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
41
42
43         HAL_Delay(100);
44         HAL_TIM_PWM_Stop(&htim2, TIM_CHANNEL_1);
45         HAL_TIM_PWM_Stop(&htim2, TIM_CHANNEL_2);
46
47         Measure_Time();
48         time_array[ i ] = time;
49         HAL_Delay(250);
50     }
51
52     Insertion_Sort(time_array);
53     arm_mean_f32(&time_array[1], 3, &time);
54 //f1ow();
55
56
57 //Value_Out(v_w, 1);
58 //Value_Out(time, 0);
59     Signal_Out(&matched_signal[ADC_BUFFER_LENGTH], ADC_BUFFER_LENGTH-2);
60
61 //Signal_Out(signal, ADC_BUFFER_LENGTH);
62 }
63 }
```

Es handelt sich hierbei um die `main`-Methodik der `main.cc`-Datei.

Zeilen 3 - 11 kümmern sich um die korrekte Initialisierung der Treiber. In diesem Fall wurde entschieden die Treiber der HAL Bibliothek zu verwenden, welche eine Abstraktion über den low-level Zugriff normaler Mikrocontroller darstellt. Der Vorteil ist, dass man sich nicht direkt sorgen über Register und Bits der einzelnen Peripherien machen muss. Nachteil ist dadurch natürlich auch ein eventuell geringeres Verständnis. **Wichtig:** Hier sind alle Kommentare, zur besseren Lesbarkeit, der eigentliche Datei entfernt. Im schon vorher erwähnt Getting Started

Guide wird darauf eingegangen, warum diese Kommentare dort sind und welchen Zweck sie erfüllen. In diesem Kapitel wird ausschliesslich auf die selbst entwickelten Methoden und Funktionen eingegangen. Ein grossteil der Konfiguration wird durch das Hilfsprogramm STMCubeMX übernommen. In diesem kann per graphischer Oberfläche Code, Pinout und Konfiguration des Controllers vorgenommen werden und der entsprechend Code generiert werden.

Ab Zeile 16 beginnt die Endlosschleife. Der erste `for`-loop initialisiert das Array welches zur Speicherung der einzelnen Messzeiten, vor Medianfilter und Mittlung, dient. Die ab Zeile 23 folgende `for`-Schleife kümmert sich um die Bestimmung der besagten einzelnen Messzeiten, wie es auch im Kapitel 2 beschrieben wird. Alle Treiber bis auf die `SystemClock_Config()` werden nochmals initialisiert, damit für jede Messung die gleichen Bedingungen vorliegen. Dies ist leider notwendig, da sonst das Rauschen und die Genauigkeit der nacheinander gemessenen Zeiten sehr hoch ist.

Die Variablen in Zeilen 33 - 35 dienen zur Messung der Zeit, welche mittels Timer `htim5` in Zeile 37 gestartet wird. Eine detaillierte Erklärung wie Timer in der HAL-Bibliothek funktionieren ist hier zu finden: <https://visualgdb.com/tutorials/arm/stm32/timers/hal/>

Zeile 38 startet den A/D-Wandler. Er wird im sogenannten DMA Modus gestartet. Das DMA ist eine Art RAM Speicher eines Mikrocontrollers, aber er hat die Eigenschaft, dass man simultan in ihn schreiben kann während der Prozessor mit anderen Dingen beschäftigt ist. Das bedeutet in diesem Fall, dass der A/D-Wandler kontinuierlich Spannung die am entsprechenden Pin anliegen wandelt und in das DMA schreibt. **Wichtig:** Anfang und Länge sind die zweiten Argumente dieser Methode, beide sind im Kopf der `main.cc` definiert, und sobald der Controller signalisiert, dass dieses Array voll ist, stoppt die Messung. Das heisst die maximale Zeitdauer die ein Singal aufgezeichnet werden kann ist durch diese Länge begrenzt und die Länge ist durch die RAM Grösse limitiert. Momentan liegt diese Grösse bei 5000, da der RAM recht ausgelastet ist. Wie schon im oberen Kapitel erwähnt erhält man mit der Sample Rate des Wandlers damit eine maximale Messzeit von:  $5000 \cdot 0.188\mu s = 940\mu s$ . Hier ist es auch noch relevant zu erwähnen, dass je nachdem wo man den A/D-Wandler startet man unterschiedliche Zeiten erhält, da der Beginn des Wandelns als Zeitpunkt 0 angesehen wird. Da aber nur Zeitdifferenzen von Relevanz sind ist es nebensächlich, ob die absolut gemessene Zeit korrekt ist oder nicht. Zeilen 40 und 41 starten die PWMs zur Erzeugung einer 40 kHz Schwingung welche auf den Sender geführt wird. Auf `TIM_CHANNEL_2` läuft dabei die inverse Modulation.

Die folgende Zeile ruft die Funktion `Measure_Time()` auf welche die gesamte digitale Signalverarbeitung aus Kapitel 2 durchführt und den Inhalt der Zeitmessung in die Variable `time` schreibt. Im folgenden wird der entsprechende Index im Array mit diesem Wert besetzt. Die beiden letzten Befehle dienen nur der Ausgabe der Signale per UART an einen definierten COM Port.

Ist diese Schleife beendet, so wird das Array sortiert und es wird der Medianfilter und die Mittelung wie in Kapitel 3 erläutert ein finaler Zeitwert bestimmt.

Nun wird kurz darauf eingegangen, wie die Messung gestoppt wird. Der folgende Code-Block zeigt die Funktion welche aufgerufen wird sobald der gesamte DMA mit Werten beschrieben ist.

```

1 void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* AdcHandle)
2 {
3     HAL_ADC_Stop_DMA(&hadc1);
4     HAL_TIM_Base_Stop_IT(&htim5);
5     timer_value = __HAL_TIM_GET_COUNTER(&htim5);
}
```

Sobald diese Funktion aufgerufen wird, ist die Messung offensichtlich vorbei, aus dem Grund wird der Wandler gestoppt. Weiterhin wird der Timer 5 welcher zu reinen Zeitmessung benutzt wird gestoppt und sein momentaner Wert eingelesen. Diese werden später in `Measure_Time()` dazu verwendet um die Zeit der Messung zu bestimmen.

Grundsätzlich sollte der Code der Methode `Measure_Time()` selbsterklärend sein. Er stellt alle

digitalen Prozesse aus Kapitel 2 dar. Hier werden nun nur ausgewählte Aspekte erläutert:

```

1 //Determine the time increment
2 time_increment = ((float)count * 2.5 + (float)timer_value/100 * 2.5)/ADC_BUFFER_LENGTH;
3
4 // Calculate the correlation
5 arm_correlate_f32(signal, ADC_BUFFER_LENGTH, sin_array, sin_length, matched_signal);

```

Die Werte zur Berechnung des `time_increment` basieren auf der Wahl des Prescalers und der Periode des Timers. `count` stellt die Anzahl dar, mit welcher der Timer zu seiner Periode gezählt hat. Aufgrund der Frequenz mit welcher er getaktet ist, entspricht jedes Erreichen der Periode  $2.2 \mu s$ . Der zweite Teil stellt den Wert der aktuellen Zählung dar. Geteilt wird durch die Breite des AD-Wandler Buffers um so die Zeit Skala zu erhalten. Grundsätzlich ist bekannt mit welcher maximalen Geschwindigkeit der Wandler Werte digitalisiert, alle  $0.188 \mu s$ . Aber hier wird dieses noch einmal überprüft. Es sei zu bemerken, dass das Berechnen der Korrelation in Zeile 5 ein Output-Array hat, welches doppelt so gross sein muss, wie das Eingangssignal. Das Array `matched_signal` wird deshalb in der Kopfzeile mit doppelter Größe erzeugt. Der aber nur der zweite Teil dieses Arrays das gefilterte Eingangssignal enthält, wird erstes als Speicher für die Hüllkurve verwendet. Aus diesem Grund, wird immer in der Mitte von `matched_signal` zugegriffen.

## 5 Analyse der Daten

Nach reichlich theoretischen Diskussion über den Ablauf und die Verarbeitung wird in diesem Kapitel der Prototyp vorgestellt und die ersten Ergebnisse von diversen Messungen präsentiert.

### 5.1 Prüfling



Abbildung 5.1: Prüfling



Abbildung 5.2: Sensoren montiert

Abbildung ?? zeigt eine Photo des Rohres.  
Dies entspricht dem in Kapitel I gezeigten Prinzip.

## 5.2 Messungen

Die ersten Evaluationen und Proof-of-Concept fanden nur mittels der Sensorik, ohne einen eigentlichen Prüfling, statt. Und die ersten im folgenden gezeigten Ergebnisse sind in Luft, ohne das Rohr erstellt worden. Sie dienen dazu die Genauigkeit der Sensoren sowie des Messprinzips und der Signalverarbeitung darzustellen. Die folgende Abbildung zeigt eine Verteilung von Mehrfachmessungen der Zeit in ruhender Luft und daneben ein Histogramm der Abweichungen vom Mittelwert: Diese liefern folgende Ergebnisse:

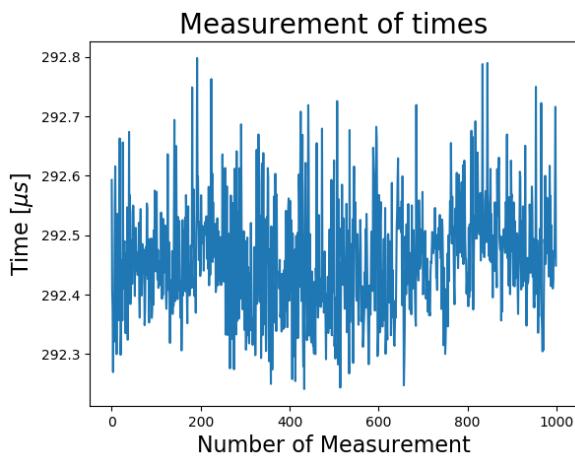


Abbildung 5.3: Messwerte

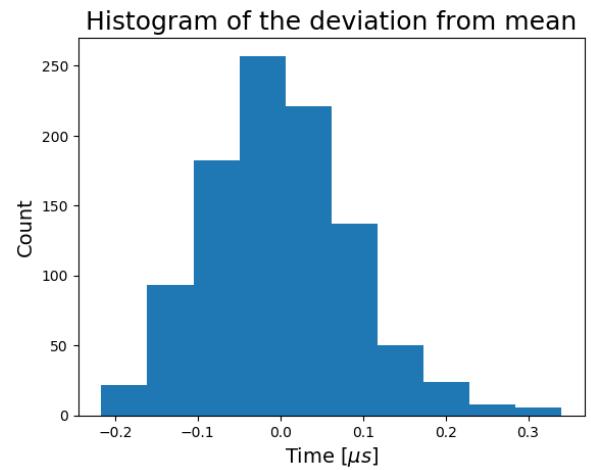


Abbildung 5.4: Histogram

$$\bar{T} = 292.46 \mu s \pm \sigma_T , \quad \sigma_T = 0.09 \mu s \quad (5.1)$$

Dies ist die maximal zu erreichende Genauigkeit in der reinen Zeitauflösung. Damit ein ungefähres Gefühl für diese Genauigkeit entsteht hier ein kurzes Rechenbeispiel des Versuchsaufbaus aus Kapitel 1.

Es soll nun bestimmt werden mit welcher Genauigkeit die Geschwindigkeit des Luftstromes bestimmt werden kann:

Aus Gleichung wird  $T = T_0 + \sigma_T$  gesetzt, da dies die minimale Auflösung ist. Für  $\alpha$  werden  $35^\circ$  verwendet und für  $v_s$   $343 \frac{m}{s}$ , für S wird  $S = 0.15 \text{ m}$  verwendet. Dies sind Beispielhafte Annahmen:

$$\Rightarrow v_w^{min} = \frac{\left(\frac{T_0}{T} - 1\right) \cdot v_s}{\cos(\alpha)} = -0.086 \left[\frac{m}{s}\right] \quad (5.2)$$

Der Wert ist negativ aufgrund der Tatsache, dass die Zeit  $T$  als grösser als  $T_0$  gewählt wurde, das Signal also länger braucht und somit der Volumenstrom in negative Richtung fliessen muss. Unter optimalen Bedingungen kann man also mit einer ungefähren Auflösung von  $0.1 \left[\frac{m}{s}\right]$  rechnen.

Nach 1.3 ist dadurch natürlich die Genauigkeit für den Volumenstrom vorgegeben. Dies liegt in einer anzustrebenden Genauigkeit für die Geschwindigkeit der Luftmassen. Verbessern kann man diesen Wert durch die Nutzung von besseren und genaueren Prozessoren sowie wahrscheinlich einer präziseren analogen Verarbeitung. Aber auch hier sind Grenzen gesetzt.

## 5.3 Problematiken

Werden die beiden Sensoren, wie in Abbildung ?? gezeigt, montiert und es wird obige Messung wiederholt, so zeigen sich einige Problematiken auf die hier, auch mit entsprechenden Lösungsansatz, eingegangen wird.

### 5.3.1 Übertragung Rohr

Durch die Magnet Vorrichtung sitzen die Sensoren fest auf dem Rohr. Dadurch kommt es zur Übertragung der Schwingung des Senders auf das Rohr und dadurch auch auf den Empfänger. Beispielhaft sind nun die Orientierung der Sender auf dem Rohr gezeigt und die dazu korrespondierende Messdaten:



Abbildung 5.5: Prüfling Aussen

Es ist klar, dass in diesem Fall das Rohr die Schwingungen überträgt. Denn wenn man weiterhin den Sender auf das äussere Rohrstück fixiert erhält man bei identischer Orientierung folgendes Ergebnis: Ein Lösungsansatz ist, O-Ringe oder vergleichbare Materialien unter die Sensoren zu

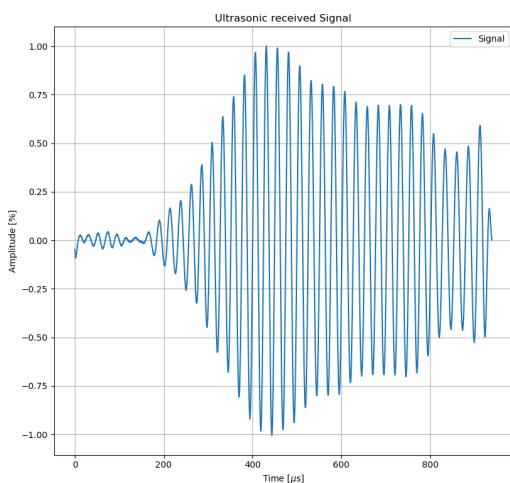


Abbildung 5.6: Messung Aussen Koppelung

setzen um diese zu entkoppeln.

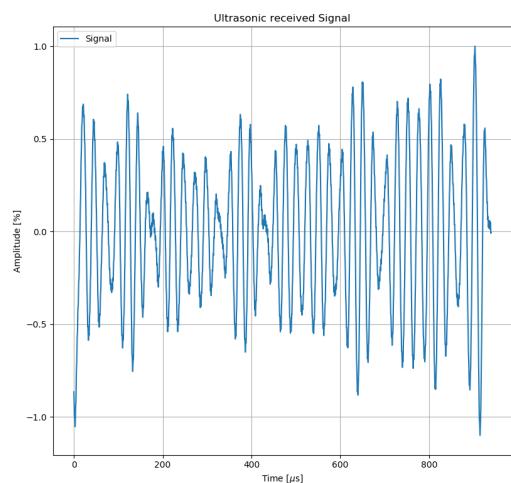


Abbildung 5.7: Messung Aussen Entkoppelt

### 5.3.2 Signalstärke

Misst man in unterschiedlichen Distanzen so fällt die Intensität radial ab, da die Sensoren Kugelwellen emittieren. Im momentanen Aufbau werden die Sensoren mit 6 VPP betrieben, da jeder GPIO-Pin des MC 3 Volt liefert. Die schwächere Intensität hat folgen auf die SNR und auf die Genauigkeit der Zeitmessung. Folgende Abbildungen zeigen drei verschiedene Distanzen: Man beachte die unterschiedlichen relativen Rauschanteile zu Beginn jeder Messung, die Mes-

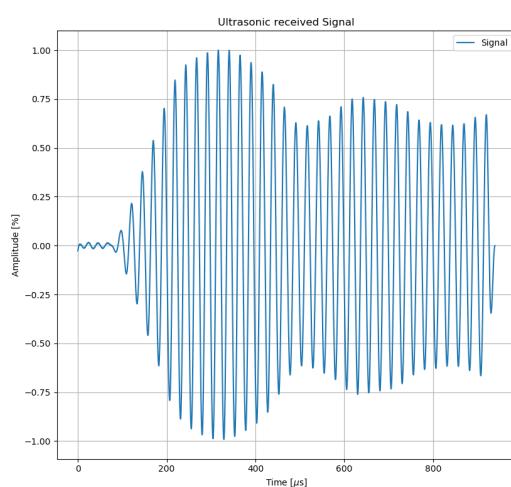


Abbildung 5.8: Signal in 3 cm Distanz

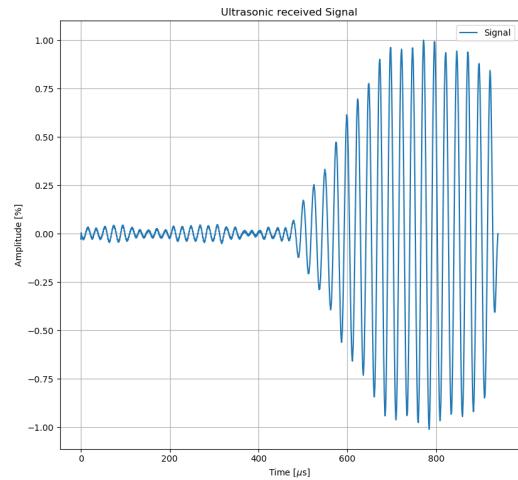


Abbildung 5.9: Signal in 16 cm Distanz

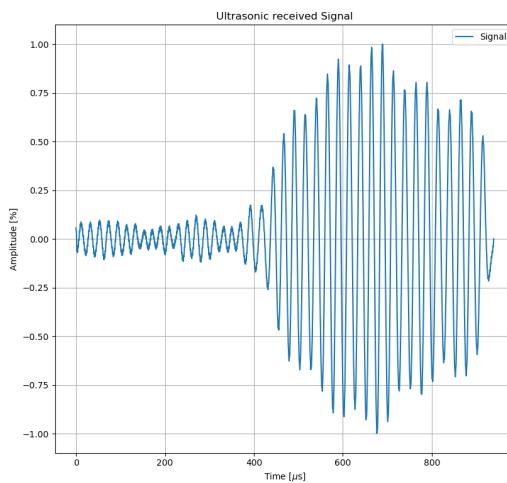


Abbildung 5.10: Signal in 35 cm Distanz

sungen sind jeweils normiert. In der 35 cm Messung musste die Aufnahme des Signals verschoben werden, damit die Signal-Kurve aufgezeichnet werden konnte. Die Begrenzung aufgrund des Speichers sorgt dafür.

Weiterhin ist es interessant zu sehen, dass der Verlauf der Kurven bei grösserer Distanz unregelmässiger werden. Versucht man nun eine Hüllkurve an eine der schwächeren Signale zu legen erhält man bei mehrfach Messungen folgendes Bild:

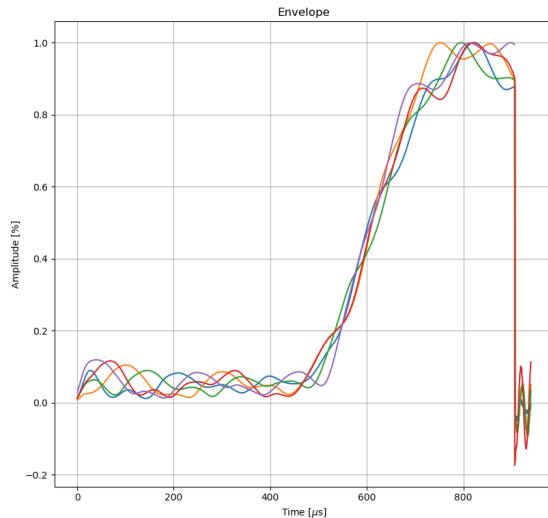


Abbildung 5.11: Hüllkurven in 16 cm Abstand

Es liegt kein Strom vor, sondern dass System befindet sich in Ruhe. Dass heisst prinzipiell muss man die identische Zeit messen. Wenn man nun den Hüllkurven-Filter anlegt, erhält man verschiedene Werte für den Beginn des Signals. Dies sorgt für eine wesentlich schlechtere Zeitauflösung, da die Ordnung des Signales unterschiedlich eingeschätzt wird.

Eine Lösung ist das Nutzen von Transistoren (z.B. einem MAX232 von TI) zur Verstärkung des Sendersignals vom MC. Weiterhin liegt aufgrund der stark hinaus stehenden Sensoren und die Rohrdurchmesser Änderungen welche auch auf Abbildung ?? zu sehen sind, ein turbulenter Strom vor. Das sorgt dafür, dass aufeinanderfolgende Messungen variieren. Folgendes Bild verdeutlicht die Problematik.

## 6 Theoretische Abhandlung

Dieses Kapitel wird auf die grundlegenden theoretischen Annahmen hinter dem verwendeten Messprinzip eingehen. Die Berechnung des Volumenstromes  $\Phi$  aus der gemessenen Zeit entspricht einer Messung der mittleren Geschwindigkeit des Vektorfeldes über die Strecke  $S$ . Die folgenden Formeln zeigen dies mathematisch:

$\vec{v}(\vec{x})$  = Reale Geschwindigkeitsprofil der Luftmoleküle

$\vec{S}$  = Richtungsvektor des Messstrecke

Damit ergibt sich folgende Formel für die Mittelung:

$$\langle \vec{v} \rangle = \frac{\int_S \vec{v} d\vec{S}}{\int_S 1 d\vec{S}} \quad (6.1)$$

In Worten dargestellt entspricht der Schätzer für den Mittelwert der Geschwindigkeit einer Projektion der lokalen Geschwindigkeitsvektoren auf die Messstrecke. Daraus wird dann der Volumenstrom berechnet. In der Messung wird die Zeit dazu verwendet diesen Mittelwert zu rekonstruieren

Im Gegensatz dazu steht die korrekte Berechnung des Volumenstromes:

$$\Phi_{theo} = \int_A \vec{v}(\vec{x}) d\vec{A} = \int_A \vec{n} \cdot \vec{v}(\vec{x}) d\vec{x} \quad (6.2)$$

Mit Querschnittsfläche  $A$  und normalen Vektor  $\vec{n}$

Offensichtlich nimmt dieser Ansatz das korrekte, reale Vektorfeld im gesamten Rohrdurchmesser in Betracht. Man kann nun einfach Turbulente Profile erdenken, wo sich diese beiden Berechnungen unterscheiden. Im trivialen Fall, dass es keinerlei Wirbel gibt und das Vektorfeld ortunabhängig ist, stimmen beide Methoden überein. Doch schon der Fall wenn man ein normales laminares Feld durch ein Rohr betrachtet ergeben konkrete Berechnungen verschiedene Ergebnisse.

Folgende Abbildung zeigt ein solches Profil:

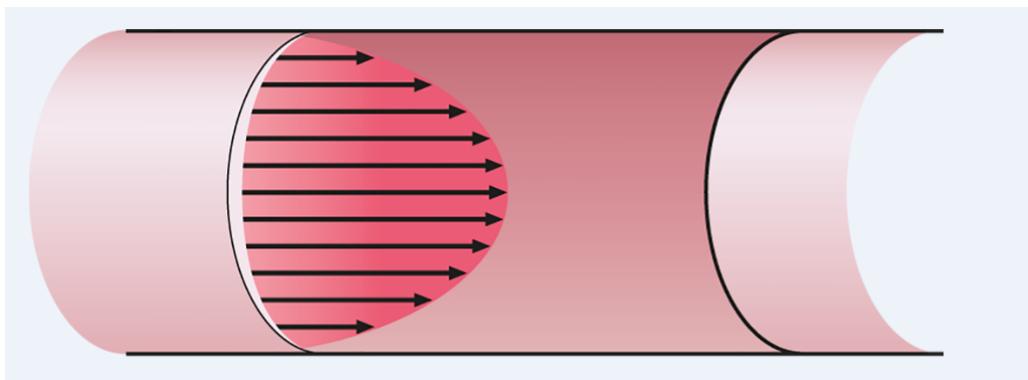


Abbildung 6.1: Laminare Strömung durch ein Rohr

Folgende Ergebnisse ergeben die Gleichungen 6.2 und 6.5:

$$\vec{v}(\vec{x}) = c \cdot \left(1 - \frac{r}{R}\right) \quad (6.3)$$

$$\Phi_{theo} = c \cdot \frac{\pi}{3} \cdot R^2 \quad (6.4)$$

$$\Phi_{mess} = c \cdot \frac{\pi}{2} \cdot R^2 \quad (6.5)$$

Dabei wird angenommen, dass der Radius des Rohres  $R$  ist, das Feld linear zum Rand symmetrisch auf null fällt und die Messstrecke  $S$  die Diagonale von oben nach unten darstellt, wobei eine Strecke von  $2 \cdot R$  auf der  $z$ -Achse (Flussrichtung) abgelegt wird.

Mit diesem wissen kann und muss ein Korrekturfaktor einbezogen werden, auch im laminaren Fall. Um dennoch korrekte Messungen, die zumindest auf Zeitmessungen beruhen, zu machen, müssen mehrere Messungen in unterschiedlichen Projektionen durchgeführt werden um entsprechend des vorhandenen Profils zu korrigieren.

Um ein Gefühl für die zu erwartende Unsicherheit der singulären Messung zu erhalten, wird hier beispielhaft die Lösung für ein Turbulentes Profil berechnet und verglichen.

## 7 Kostenschätzung

Die Kosten für den verwendeten Prototypen sind schwierig abzuschätzen. Da in Zukunft die Platine und der Mikrocontroller durch einen ASIC ersetzt werden, kann in diesem Zusammenhang nur grob geschätzt werden. Gerechnet wird mit 3 - 5 CHF für die analoge Signalverarbeitung sowie den Controller. Dagegen kann man bei der Sensorik interessanter Trends beachten. So kann schon jetzt über diverse Plattformen Ultraschall Sensoren für einen Stückpreis von 0.3 - 0.8 \$ erstanden werden. In einer solchen Preisklasse kann das Verwendeten einer solchen Messmethodik durchaus sehr kompetitiv sein.

## 8 Ergebnis

Um eine zusammenfassende Aussage über die Nutzbarkeit von Ultraschall gestützten Methoden zur Bestimmung von Volumenströmen zu machen, ist es noch etwas früh. Denn die verwendete Methodik mit nur einem Sender-paar ist gegenüber Turbulenzen und Wirbeln nicht resistent. Dennoch konnte hier gezeigt werden, mit welcher maximalen Auflösung zeitlich gerechnet werden kann. Dies hat Ultraschall als valide Alternative bewiesen. Es können ausreichend genaue Messungen gemacht werden. Die Schätzung der Kosten verspricht zumindest ein finanzielle Konkurrenz zu anderen benutzten Messmethodiken

Werden die Methodiken, welche kurz im Kapitel Theoretische Abhandlungen angerissen worden sind, implementiert, so bestehen gute Aussichten ein Produkt zu konstruieren, welches in wesentlich mehr Anwendungsfällen als Konkurrenz Produkte, in der Lage ist, korrekte Volumenstrommessungen zu machen.

## 9 Ausblick

In Anbetracht der Ergebnisse und des momentanes Standes gibt es verschiedene sinnvolle nächste Schritte.

So sollte als erstes das vom MC generierte PWM verstärkt werden. Für eine solche Aufgabe kommt ein Verstärker wie zum Beispiel der MAX232 von Texas Instruments in Frage. Dadurch kann die Intensität des Signals auch bei grösseren Distanzen aufrecht erhalten werden um eine hohe Genauigkeit der Zeit Messung zu garantieren. Um die im Rohr vorliegende Distanz zu erreichen muss die Intensität um den Faktor 4 verstärkt werden, was einer Verstärkung der PWM Spannung um 16-Fach entspricht. Da das PWM durch ein inverses schalten von Pins erzeugt wird, reicht es die jeweilige einzelne Spannung um den Faktor 8 zu verstärken. Dass heisst von den momentanen 3 Volt auf 24 V.

Eine weitere Alternative ist das Begrenzen des Abstrahlwinkels durch ein Richtelement welches man an dem Sensor anbringt. Dadurch emittiert der Sender natürlich keine Kugelwellen mehr, welches den Intensitäts Verlust nicht mehr mit dem Abstandsquadrat kollereirt sondern geringer ist.

Nun sollte der zu Beginn angesprochene bessere Versuchsaufbau realisiert werden, damit die Vorzüge dieser Methodik genutzt werden können.

Sollte sich während des Arbeitens herausstellen, dass es nicht nur an der reinen Intensität des Signals liegt, welches die Genauigkeit der Zeitmessung vorgibt, so sollte über eine alternative digitale Signalverarbeitung nachgedacht werden. Als Ansatz kann dann Frequenzmodulation des Senders + Pulskompression als Ansatz verwendet werden. Ähnlich der Praxis in der Radar-technik.

Als finale Verbesserung nicht unbedingt der Messmethodik sondern der Korrektur der berechneten Geschwindigkeiten kann über Messungen mehrerer Sensoren nachgedacht werden, welche dazu genutzt werden können um das vorhandene Geschwindigkeitsprofil zu charakterisieren und darauf basierend einen Korrektur Faktor einzubeziehen. Gerade im Falle von turbulenten Felder kann die in diesem Dokument vorgestellte Methodik zu fehlern führen.

## 10 Anhang

Die Berechnung zur alternativen Messmethodik werden hier vollzogen:

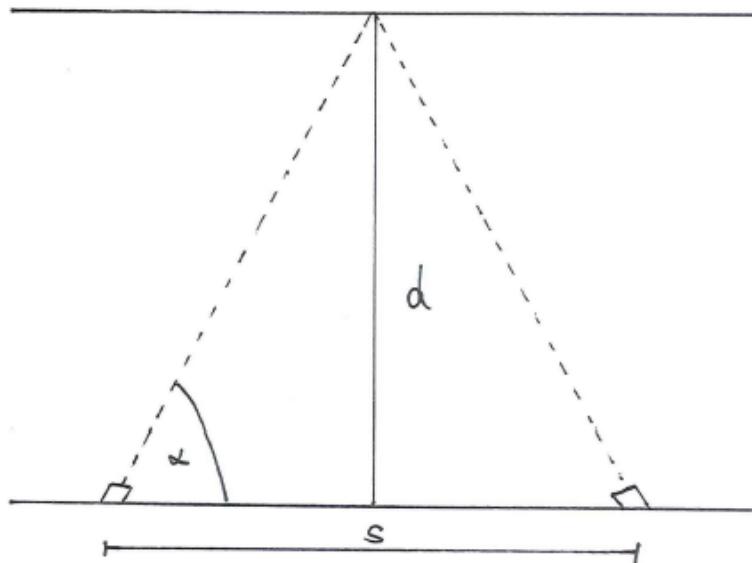


Abbildung 10.1: Veranschaulichung des alternativen Messprinzips

$$\alpha = \arctan \frac{d}{s/2} \quad (10.1)$$

$$T_{forward} = \frac{2 \cdot d}{\sin \alpha \cdot (v_s - \cos \alpha \cdot v_w)} \quad (10.2)$$

$$T_{backward} = \frac{2 \cdot d}{\sin \alpha \cdot (v_s + \cos \alpha \cdot v_w)} \quad (10.3)$$

$$v_w = \left( \frac{1}{T_{backward}} - \frac{1}{T_{forward}} \right) \cdot \frac{d}{\sin \alpha} \cdot \cos \alpha \quad (10.4)$$

Der Vorteil zeigt sich direkt, denn es gibt keine Abhängigkeit der Schallgeschwindigkeit in Luft mehr. Dies sorgt für Temperatur Unabhängigkeit.

## 11 Quellen

### Literatur

- [1] Lindab Ultralink <http://www.lindab.com/global/pro/Pages/ultralink.aspx>
- [2] Timer Tutorial [https://www.mikrocontroller.net/articles/AVR-Tutorial:\\_Timer](https://www.mikrocontroller.net/articles/AVR-Tutorial:_Timer)
- [3] A/D-Wandler <https://www.mikrocontroller.net/articles/AD-Wandler>
- [4] PWM [https://www.mikrocontroller.net/articles/AVR-Tutorial:\\_PWM](https://www.mikrocontroller.net/articles/AVR-Tutorial:_PWM)
- [5] Match Filter [https://en.wikipedia.org/wiki/Matched\\_filter](https://en.wikipedia.org/wiki/Matched_filter)

## **Temporary page!**

LATEX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LATEX now knows how many pages to expect for this document.