Entwurf einer Schaltung zur Spannungsphasenwinkelmessung in großflächigen, elektrischen Netzen

Praxisprojekt

vorgelegt von: Ashraf Ishag
Matrikel-Nr.: 11112299
Adresse: Sauerlandstr.5
51105 Köln
ashraf.ishag@hotmail.com

Gutachter: Prof. Dr.- Ing. Eberhard Waffenschmidt

Köln, 14.05.2023

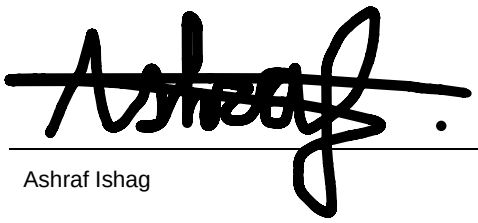
Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Köln, den 14.05.2023

Ort, Datum

Ashraf Ishag



Ashraf Ishag

Zielsetzung

Das Ziel dieser Arbeit besteht in der Entwicklung einer Hardware für ein Spannungswinkelmessgerät, das in großflächigen Messumgebungen eingesetzt werden kann. Das Praxisprojekt dient als Vorarbeit für die darauf aufbauende Bachelorarbeit, welche sich mit der Bestimmung der Lage des elektrischen Netzes durch das entwickelte Spannungswinkelmessgerät beschäftigt. Hierbei werden die Phasenverschiebungen von zwei Rechtecksignalen an verschiedenen Standorten erfasst und miteinander verglichen. Auf Basis des Phasenwinkels können dann die Leistungsflüsse, Knotenströme und Leistungsverluste zwischen Erzeuger und Verbraucher berechnet werden. Im Rahmen des Praxisprojekts wird ausschließlich auf die Hardwareentwicklung, die zugehörige Software sowie die Anforderungen an ein solches Messgerät eingegangen.

Abstract

This practical project serves as a preliminary work for a future bachelor's thesis on the design and implementation of a phasor measurement unit (PMU) using a microcontroller-based data acquisition system. The PMU is designed to measure the phase and frequency of a power system accurately based on a hardware that transforms a sinus-phase into a rectangular wave. A software will also be written to extract the time measured from the developed hardware with the help of a microcontroller.

This Practical Project involves designing and implementing the PMU using a device that detects the zero crossing of the grid. The finished hardware and software are tested in the results evaluated.

Keywords: Phasor measurement unit, PMU, microcontroller, power system monitoring, accuracy.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

Um eine Veränderung im System zu messen, müssen zwei identische Spannungsmessgeräte mit gleichen Eigenschaften zusammen als Messsystem verwendet werden. Die Spannungswinkelmessgeräte sollten an verschiedenen Messpunkten im zu messenden Netz platziert werden, um die zeitsynchronisierten Messwerte vergleichen zu können. Eine Komparator Schaltung wird verwendet, um die Zeitverschiebung zwischen zwei Rechtecksignalen zu ermitteln. Diese erzeugt im Nulldurchgang des Sinussignals einen positiven Rechteck-Impuls. Wenn der Nulldurchgang des Sinussignals (rot) erreicht ist, wird eine Sekundärspannung (U_{comp}) erzeugt, die als steigende Flanke auf 5 V transformiert wird. Diese bleibt konstant auf 5 V bis $t2$ und fällt dann auf 0 Volt ab. Gleichzeitig wird eine zweite positive Rechteckspannung (grün) mit 5 V zwischen dem Nulldurchgang $t1$ und $t3$ des Sinussignals (grün) erzeugt. Diese transformierte Rechteckspannung entspricht dem Stromversorgungsanschluss des Raspberry Pis. In **Abbildung 1-1** wird der präzise Ablauf der Umwandlung von Spannung veranschaulicht.

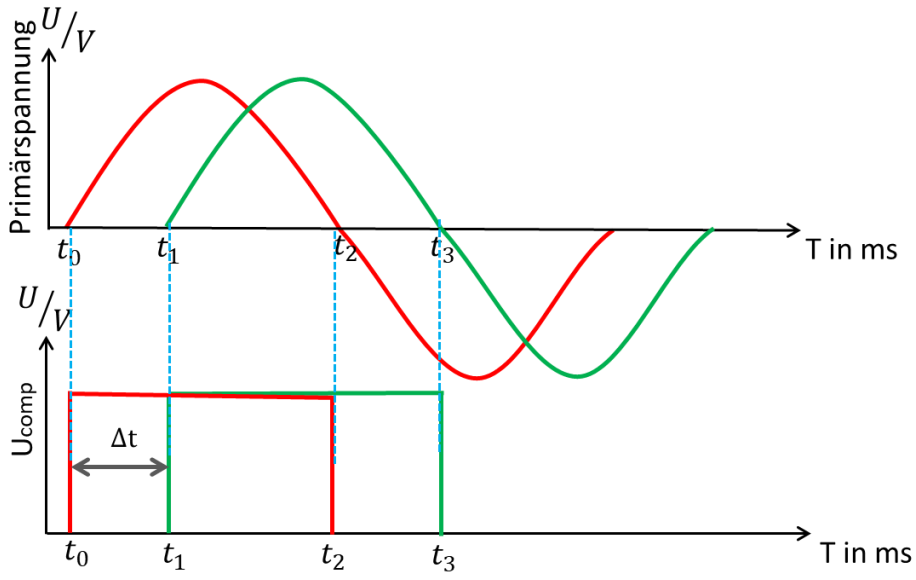


Abbildung 1-1: Spannungsumwandlung und Zeitverschiebung zwischen zwei Rechtecksignalen

Zusätzlich wird jedem verwendeten Messgerät ein hochpräziser GPS-Empfänger zugeordnet, um einen präzisen Zeitstempel zu markieren, wenn das Rechtecksignal eine Flankenänderung aufweist. Wenn die Messgeräte aktiv sind, können Informationen über die Netzwerkkonfiguration und die Netzstabilität des gesamten Systems anhand des erfassten Phasenwinkels abgeleitet werden.

$$\Delta\varphi = \Delta t * f * 360°$$

Mit: $\Delta\varphi$ Phasenwinkel in Grad°

Δt	Zeitverschiebung in ms
f	Grundfrequenz in Hz

Das Ziel dieser Arbeit besteht darin, eine Hardware zu entwickeln, die in der Lage ist, die Frequenz im Netz jederzeit zu messen. Dadurch soll es möglich sein, den Phasenwinkel zu bestimmen, indem die Messungen an zwei verschiedene Standorte gleichzeitig erfasst werden. Zur Umsetzung dieses Ziels wird eine Stoppuhr entworfen, die eine Genauigkeit von unter 1° aufweist. Die Entwicklung dieser Hardware erfordert die Auswahl und Integration geeigneter Komponenten sowie die Programmierung und Validierung der erforderlichen Software. Dazu werden verschiedene Aspekte wie die Robustheit, Zuverlässigkeit und Messgenauigkeit berücksichtigt. Um die Funktionsfähigkeit der entwickelten Hardware zu testen, werden Messungen in einem realen Netzwerk durchgeführt. Die Ergebnisse der Messungen werden ausgewertet und diskutiert, um die Leistungsfähigkeit der entwickelten Hardware zu bewerten.

Insgesamt soll diese Arbeit dazu beitragen, eine effektive und zuverlässige Methode zur Messung des Phasenwinkels in einem Netzwerk zu entwickeln, um eine bessere Steuerung und Überwachung des Netzes zu ermöglichen.

2 Stand der Forschung

Phase Measurement Units (PMUs) sind heute weit verbreitet und werden zur Messung von Strom- und Spannungswerten verwendet, um den Zustand und die Leistung von Stromnetzen zu überwachen und zu steuern. Eine der wichtigsten Aufgaben von PMUs ist die Messung der Frequenz im Stromnetz, da Änderungen in der Frequenz ein Hinweis auf Netzstörungen oder Überlastungen sein können.

Die Entwicklung von PMUs hat in den letzten Jahren große Fortschritte gemacht. Heutzutage sind PMUs kosteneffektiver und weit verbreiteter als noch vor einigen Jahren. Die Preise für PMUs können je nach Hersteller und Funktionen variieren. Einige PMUs kosten etwa 1000-3000 US-Dollar (Schweitzer Engineering Laboratories, 2023), während andere hochentwickelte PMUs mit mehreren Funktionen bis zu 10.000 US-Dollar oder mehr kosten können (Crompton Instruments, 2023). PMUs sind auch heute in vielen Teilen der Welt verfügbar. In den USA und in Europa werden PMUs häufig zur Überwachung von Stromnetzen eingesetzt. In anderen Teilen der Welt, wie zum Beispiel in China und Indien, hat die Verbreitung von PMUs in den letzten Jahren ebenfalls zugenommen (Saha & Yoo, 2023).

Ein wichtiger Faktor bei der Verwendung von PMUs ist ihre Lebensdauer. Moderne PMUs haben in der Regel eine Lebensdauer von mindestens 10 Jahren (Electric Power Research Institute, 2023), obwohl einige PMUs möglicherweise länger halten können, wenn sie regelmäßig gewartet werden. In Bezug auf die Messung der Frequenz hat die Forschung in den letzten Jahren Fortschritte gemacht. Einige Studien haben sich auf die Verbesserung der Genauigkeit und Zuverlässigkeit von PMUs bei der Messung der Frequenz konzentriert (Srinivasan & Zhang, 2023), während andere Studien sich auf die Entwicklung neuer Technologien für die Frequenzmessung durch PMUs konzentriert haben.

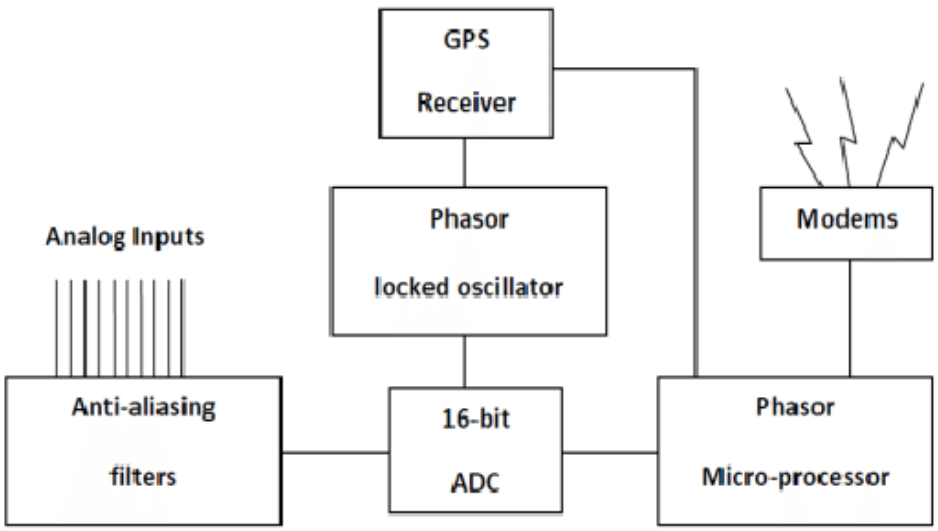


Abbildung 2-1: Beispiel eines PMU-system (Rihan, 2023)

3 Theoretische Grundlagen der Hardware

3.1 Raspberry Pi 3B+

Der Raspberry Pi 3B+ ist ein Einplatinencomputer mit einer Vielzahl von physischen Fähigkeiten, die ihn zu einem äußerst vielseitigen Gerät machen. Der Raspberry Pi verfügt über vier USB-2.0-Anschlüsse, einen Gigabit-Ethernet-Anschluss, einen 3,5-mm-Klinkenstecker für Audio und Composite-Video, einen HDMI-Ausgang, einen CSI-Kameraanschluss und einen DSI-Displayanschluss.

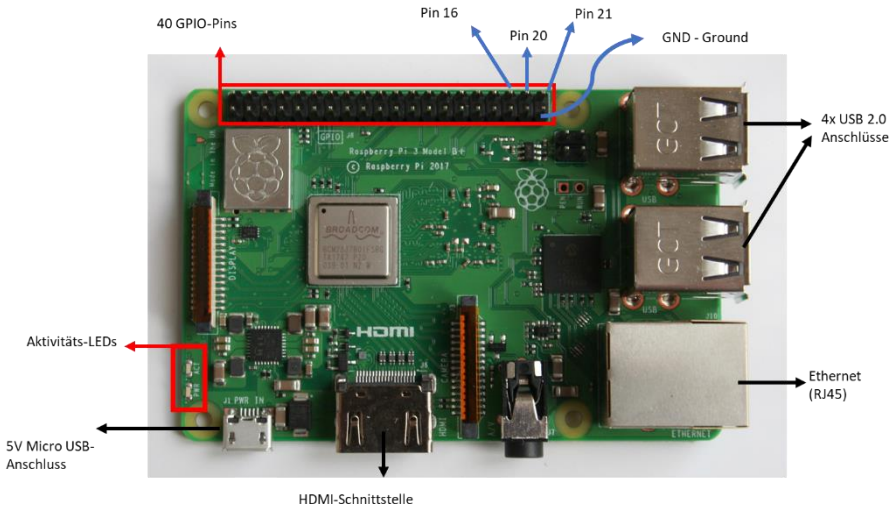


Abbildung 3-1: Raspberry Pi Modul 3B+

Darüber hinaus verfügt er über 40 GPIO-Pins, die oben am **Abbildung 3-1** dargestellt ist. Diese Pins sind in drei Gruppen unterteilt: die 3V3-Pins, die 5V-Pins und die GND-Pins. Die 3V3-Pins liefern eine Spannung von 3,3 V, während die 5V-Pins eine Spannung von 5V liefern. Die GND-Pins werden für die Masseverbindung verwendet. Zusätzlich gibt es noch Pins für die serielle Kommunikation, die Stromversorgung und die anderen Funktionen. Die Stromversorgung des Raspberry Pi 3B+ erfolgt über ein Micro-USB-Kabel, das an eine Stromquelle angeschlossen werden kann. Der Raspberry Pi benötigt eine Stromversorgung von 5 V und mindestens 2,5 A, um ordnungsgemäß zu funktionieren. Es kann auch über die 5V Pins betrieben werden, indem man die Stromquelle an der Pins setzt.

Die 40 GPIO-Pins des Raspberry Pi 3B+ sind in zwei Reihen von je 20 Pins auf beiden Seiten des Boards angeordnet. Sie sind in der Lage, digitale Signale mit einer Spannung von 3,3 V zu empfangen und zu senden und können als Eingangs- oder Ausgangspin konfiguriert werden. Die meisten dieser Pins sind in Gruppen mit ähnlichen Funktionen angeordnet.

Im Rahmen dieses Projekts werden insgesamt drei GPIO-Pins benötigt, um die gewünschten Funktionen zu realisieren. Zwei der Pins dienen dabei als digitale Inputs, während der dritte Pin als Output fungiert. Konkret handelt es sich um die GPIO-Pins

16, 20 und 21, welche sich auf der unteren rechten Seite des Raspberry Pi befinden. Weitere Details zu den genannten Schnittstellen sowie deren Anbindung und Konfiguration werden im **Kapitel 4.2** ausführlich beschrieben.

Es gibt viele Programmiersprachen, die auf dem Raspberry Pi verwendet werden können, aber ich habe mich für Python entschieden. Ein großer Vorteil von Python ist, dass es eine umfangreiche Standardbibliothek hat, die eine Vielzahl von Funktionen und Modulen zur Verfügung stellt. Ein weiterer Vorteil ist die große Entwicklergemeinschaft, die eine Vielzahl von Bibliotheken und Werkzeugen für Python zur Verfügung stellt.

Der Raspberry Pi 3B+ verfügt über einen 1,4-GHz-Quad-Core-Prozessor und 1 GB RAM, was für diesem Projekt ausreichend ist. Die Geschwindigkeit der Signalverarbeitung an den GPIO-Pins hängt von der Prozessorleistung ab und beträgt etwa 1 Million Signale pro Sekunde.

3.2 GNSS-5-Click

GNSS-5, auch bekannt als Global Navigation Satellite System (GNSS), ist ein satellitengestütztes Navigationssystem, das weltweit Geolokalisierungs- und Zeitinformationen für Nutzer bereitstellt. GNSS-5 verwendet ein Netzwerk von Satelliten, um die genaue Position eines Empfängers zu bestimmen, indem die Zeit gemessen wird, die Signale benötigen, um von den Satelliten zum Empfänger zu gelangen.

Technisch gesehen besteht GNSS-5 aus einem Empfängermodul, das mit einer externen Antenne verbunden ist, die die Satellitensignale empfängt. Das Modul enthält auch einen Prozessor, der die Signale verarbeitet und die Position und Zeit berechnet. Das GNSS-5-Modul arbeitet mit einer Betriebsspannung von 3V und enthält eine PPS-Signal-Pin, die für eine genaue Zeitreferenz verwendet wird. Das PPS-Signal ist ein Impulssignal, das einmal pro Sekunde generiert wird. Es handelt sich dabei um ein Rechtecksignal mit einer Impulsdauer von 100 Millisekunden. Um die Impulsdauer des PPS-Signals zu verkürzen, wird eine RC-Glied mit einem NAND-Gatter und einem Inverter verwenden. Dieses Schaltungskonzept wird in **Kapitel 4.1** ausführlicher behandelt.

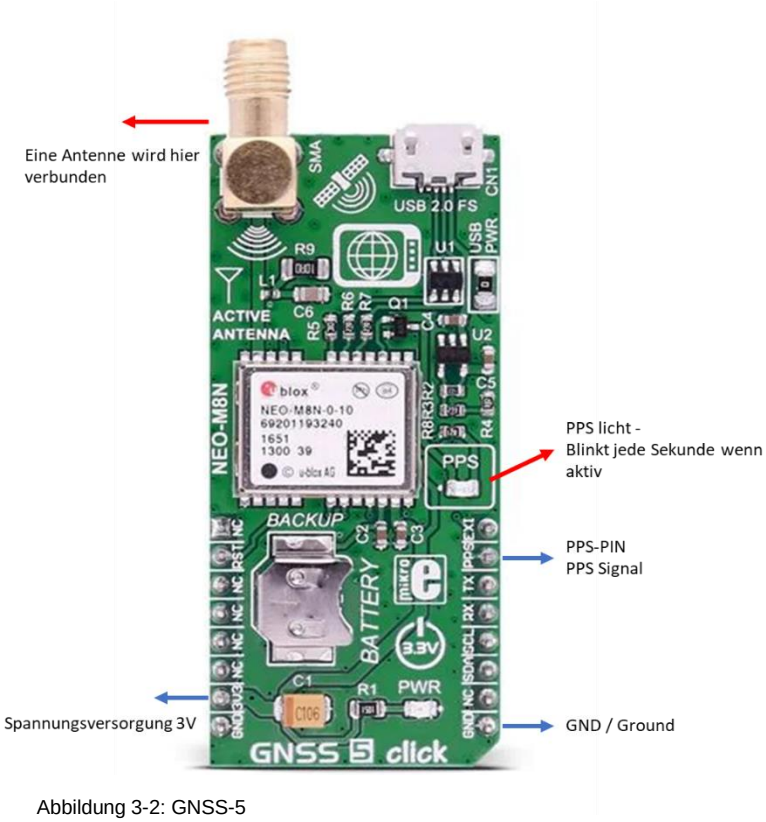


Abbildung 3-2: GNSS-5

In diesem Projekt werde ich das PPS-Signal des GNSS-5-Moduls als Reset-Signal für die Zähler verwenden. Dieses Verfahren ermöglicht es, die Zähler in synchronisation mit dem GNSS-Signal zurückzusetzen, was zu einer präziseren Erfassung der Messwerte führt.

Die Zähler werden verwendet, um die Anzahl der Impulse des PPS-Signals und anderer Signale, wie beispielweise U_{comp}, zu zählen. Durch die Verwendung des PPS-Signals als Reset-Signal kann die Anzahl der Impulse genau gezählt, um präzise Messungen zu gewährleisten. Die Pins, die uns wichtig sind, sind nochmal in **Abbildung 3-2** zu sehen.

3.3 Komparator basierte Phasennulldurchgangsmessplatine

Gegenstand dieses Abschnitts ist die von einem Master-Studenten entwickelte Komparator-Platine. Diese Platinen Schaltung ist in der Lage, aus einer einphasigen Knotenspannung von 230V (Usin) ein 5V Rechtecksignal (Ucomp) zu erzeugen und besteht aus einem Spannungsteiler, einem Komparator und einem Optokoppler zum Schutz der nachgeschalteten Komponenten.

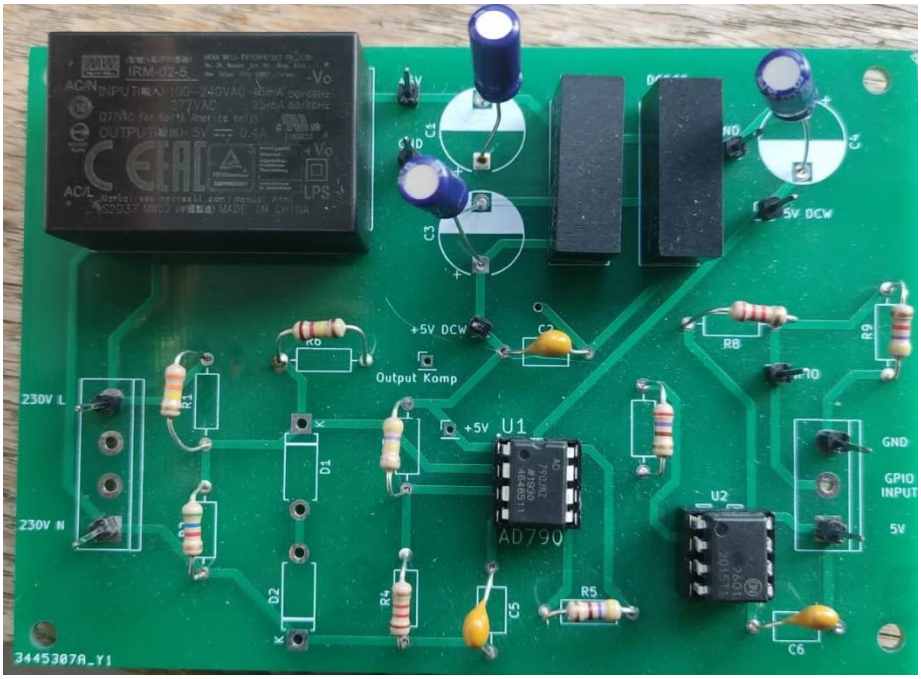


Abbildung 3-3 : Komparator-Platine

Der Spannungsteiler reduziert die hohe Eingangsspannung auf ein Niveau, das vom Komparator verarbeitet werden kann, während der Optokoppler die nachgeschalteten Komponenten vor Überspannungen und Schäden schützt. Die Durchschaltzeit der Komponenten beträgt etwa 60ns, was einem Phasenwinkel von 0,001° entspricht. **Abbildung 3-3** bietet eine visuelle Darstellung der Komparator-Platine und zeigt die verschiedenen Komponenten sowie deren Verbindung untereinander.

3.4 Counter

Für die Entwicklung eines Spannungswinkel-Messgeräts wurde der M74HC590-Zähler ausgewählt, da er eine schnelle Zählgeschwindigkeit aufweist und in der Lage ist, Echtzeit-Zählungen durchzuführen, was für die Erfassung von Messdaten in Echtzeit unerlässlich ist. Der M74HC590-Zähler zeichnet sich durch eine schnelle Zählgeschwindigkeit aus und ermöglicht dadurch eine genaue und zuverlässige Messung von Frequenzen. Der Zähler ist kosteneffizient und bietet eine hohe Genauigkeit von bis zu 0,1 Prozent. Dank seiner Erweiterbarkeit bietet der Zähler die Möglichkeit zur Anpassung an die individuellen Anforderungen des Projekts.

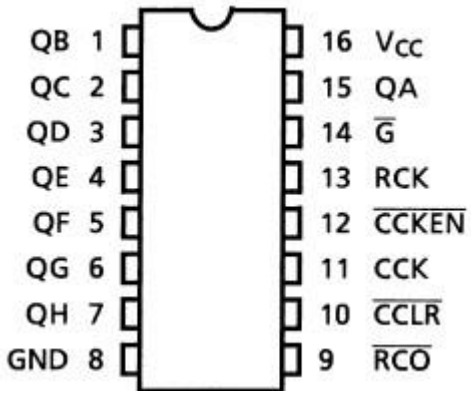


Abbildung 3-4: M74HC590 Zähler

Der M74HC590 Counter ist ein integrierter Schaltkreis, der zur Zählung von das 5V Rechtecksignal (Ucomp) verwendet wird und bis zu 8-Bit-Zahlen zählt. Die Rechteckflanke des Komparators ist am RCK-Pin angeschlossen und der Quarzoszillator ist mit dem CCK-Pin verbunden. Wenn ein rechteckiger Impuls am RCK-Pin empfangen wird, wird der aktuelle Zählwert des Zählers auf die Ausgangspins übertragen. Der Zähler erhöht dann seinen Zählwert um eins bei steigender Flanke des nächsten Taktimpulses, der am CCK-Pin empfangen wird. Dieser Prozess wiederholt sich mit jedem eingehenden rechteckigen Impuls, wodurch der Zähler bei jedem Impuls seinen Zählwert um eins erhöht. Durch die Verwendung von drei Zählern wird eine höhere Speicherkapazität erreicht, da theoretisch bis zu 50.000 Takte pro 50Hz-Wellenlänge gezählt werden können. Die Pin-Verlegung des Zählers ist im **Abbildung 3-4** deutlicher zu erkennen.

Der M74HC590-Zähler hat eine schnelle Signalreaktionszeit von weniger als 20 ns, was eine schnelle Erfassung von Signalen ermöglicht. Die maximale Eingangs- und Ausgangsstromstärke beträgt 25 mA, was in Verbindung mit einem Microcontroller wie dem Raspberry Pi 3b+ optimal ist. Pin 14 des Zählers ist mit dem Ground-Pin verbunden, um die Ausgangspins des Zählers immer zu aktivieren.

3.5 Register

Der 74HC165 ist ein 8-Bit-Parallel-in-/Serial-out-Schieberegister, das in der Entwicklung des Spannungswinkel-Messgeräts aufgrund seiner Vorteile ausgewählt wurde, darunter seine einfache Erweiterbarkeit, geringe Kosten und Fähigkeit, mehrere Eingangssignale gleichzeitig abzufragen. Das Pin-Verlegung des Registers ist in **Abbildung 3-5** dargestellt.

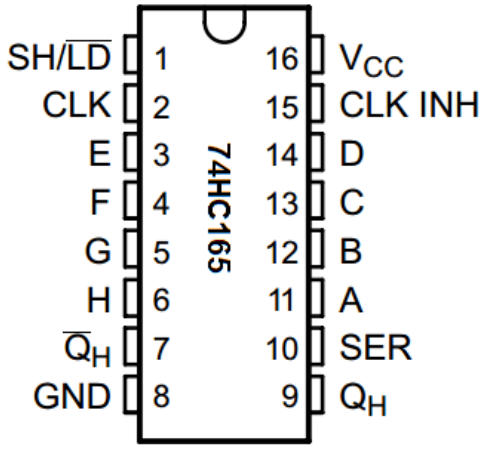


Abbildung 3-5: 74HC165 Register

Es kann 8-Bit-Werte parallel laden und dann seriell ausgeben. Zum Laden von Daten in das Register wird ein kurzer Impuls an Pin 1 (SH/LD) angelegt, der den ersten Eingangsbit-Wert in die letzte Bitposition des Registers lädt. Die restlichen Bits werden an die nächste Position im Register verschoben, bis alle 8 Bits im Register gespeichert sind. Die Daten können dann seriell ausgegeben werden, indem das Schieberegister erneut verschoben wird. Um das Schieberegister zu verschieben, wird ein Taktsignal an Pin 2 (CLK) angelegt. Bei jeder steigenden Taktflanke wird das Schieberegister um eine Position verschoben. Der aktuelle Wert des letzten Bits im Register wird auf den Serienausgang (PIN 9) ausgegeben und das nächste Bit wird an die letzte Position des Registers verschoben. Der Prozess wird so lange wiederholt, bis alle 8 Bits des Registers seriell ausgegeben wurden. Das Taktsignal wird vom Raspberry Pi erzeugt und an Pin 11 (CP) des 74HC165-Registers übertragen, um den Schieberegisterprozess zu steuern. Der Serienausgang des 74HC165-Registers wird an einem GPIO-Pin des Raspberry Pi angeschlossen, um die seriell ausgegebenen Daten zu empfangen und zu lesen.

3.6 Quarzoszillator

In diesem Abschnitt werde ich die Quarzoszillator-Komponente meines Phase-Messgeräts vorstellen und erläutern, warum ich mich für diese spezielle Quarzoszillator-Komponente entschieden habe.

Ein Quarzoszillator ist eine elektronische Komponente, die eine stabile und genaue Frequenz zur Verfügung stellt. Der Quarzoszillator arbeitet, indem er die Resonanzeigenschaften eines Quarzkristalls ausnutzt, um eine Frequenz zu erzeugen. Es basiert auf einem kristallinen Resonator, der in einem Schwingkreis mit einem Verstärker und anderen passiven Komponenten integriert ist. Der Schwingkreis wird auf die Resonanzfrequenz des Kristalls abgestimmt, wodurch eine stabile und präzise Schwingung erzeugt wird. Die Frequenz des Oszillators wird durch den Kristall selbst bestimmt, was zu einer hohen Frequenzgenauigkeit und Stabilität führt. Für das Projekt wurde ein Oszillator ausgewählt, der als Zeitgeber fungiert und die Schwingungsdauer einer Phase präzise misst, ähnlich wie eine Uhr. Um das Projektziel einer Genauigkeit von weniger als 1° Phasenwinkel zu erreichen, sollte die Zeit, in der der Quarz eine ganze Periode durchläuft, weniger als 56 Mikrosekunden betragen, gemäß das Formel:

$$1° * \frac{1}{50Hz * 360°} = 56\mu S$$

Es wurden viele verschiedene Frequenzen getestet. Je höher die Frequenz, desto kürzer sind die Wellenlängen und desto größer ist die Anzahl der Quarz-Takte. Quarze mit höheren Frequenzen können manchmal nicht vom Zähler erfasst werden. Mit steigender Quarzfrequenz steigt auch der Bedarf an Speicherplatz, um die Anzahl der gezählten Quarzperioden innerhalb einer 50-Hz-Periode zu speichern.

Letztendlich wurde ein Quarz mit einer Frequenz von 2,4576 MHz ausgewählt. Dies entspricht einer Periode von 409 nS. Die Genauigkeit des Quarzoszillators beträgt +/- 20 ppm (parts per million). Der Ausgang am Pin 8 liefert die Sinuswellenform des Oszillators. Pin 14 ist die Versorgungsspannung und wird mit +5 V DC betrieben. Pin 7 (GND) ist die Masseverbindung des Oszillators.

4 Entwicklung des Messsystems

4.1 Kurzimpulsdesign mit RC-Gliedern und Logikgattern

Das vorliegende Design im **Abbildung 4-2** ist für Anwendungen geeignet, die schnelle Reaktionen auf Pulse erfordern. Es umfasst den Einsatz von Counter- und Register-Schaltungen, die auf Pulse reagieren anstatt auf Flanken. Das PPS-Signal hat eine Pulsbreite von 100 ms, und das Reset des Counters erfolgt nach dem Ende des Pulses. Für den Register-Pin 1 werden die Werte erst nach einem 20 ms langen Puls U-comp vom Counter geholt.

Um das "Reset" und "Shift/Load" zu beschleunigen, werden neue, kürzere Pulse benötigt. Hierfür werden ein Inverter, ein OR-Gate, ein NAND-Gate, Widerstände und Kondensatoren eingesetzt. Der Inverter 74LS04 weist eine Reaktionszeit von 9 ns und eine Verzögerungszeit von 15 ns auf. Das OR-Gate 74HC32 hat eine Verzögerungszeit von 6 ns und eine Genauigkeit von 5%, während das NAND-Gate SN74LS00 eine Reaktionszeit von 9 ns und eine Verzögerungszeit von 15 ns hat.

Input A	Input B	Output Y
0	0	1
0	1	1
1	0	1
1	1	0

Tabelle 4.2: Truth-Tabelle NAND-Gate

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

Tabelle 4.1: Truth-Tabelle OR-Gate

Das PPS-Signal wird zunächst invertiert und dann durch ein RC-Glied verzögert. Das verzögerte Signal wird anschließend mit dem Originalsignal im NAND-Gate kombiniert, um das resultierende Signal PPS_{kur} zu erzeugen. Das Komparator-Signal wird eben-

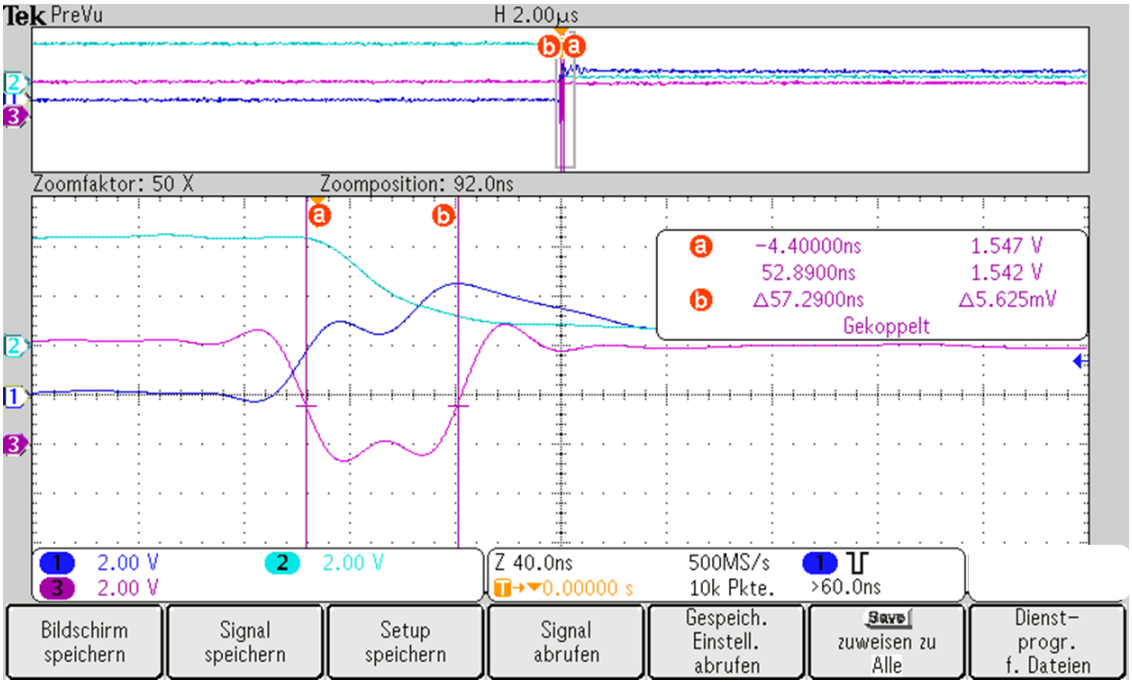


Abbildung 4-1: Messung von neuem Puls

falls invertiert und durch ein RC-Glied verzögert. Danach wird es mit dem Originalsignal im OR-Gate kombiniert, um das resultierende Signal U_{comkur} zu erzeugen.

Im **Abbildung 4-1** kann man sehen, dass der neue erzeugte Puls eine Breite von nur 57ns hat. Durch den Einsatz des beschriebenen Designs können kürzere Pulse erzeugt werden, um das Reset und Shift/Load zu beschleunigen. Dadurch wird die Effizienz des Designs erhöht, was zu einer schnelleren Reaktion auf Pulse führt.

4.1.1 Beschreibung der ausgewählten Messgrößen

Die Integration eines RC-Glieds in diese Schaltung kann die Messgrößen beeinflussen, da die Schaltung durch zusätzliche Komponenten und deren Wechselwirkungen die Eigenschaften des RC-Glieds modifizieren kann, insbesondere hinsichtlich seiner Zeitkonstante und Grenzfrequenz. Zum Beispiel kann die Eingangsimpedanz einer Schaltung die Messgrößen eines RC-Glieds beeinflussen. Wenn das RC-Glied als Teil einer Schaltung mit einer hohen Eingangsimpedanz verwendet wird, kann dies zu einer erhöhten Zeitkonstante führen, da der Kondensator langsamer auf Änderungen des Eingangssignals reagiert. Auf der anderen Seite, wenn das RC-Glied als Teil einer Schaltung mit einer niedrigen Eingangsimpedanz verwendet wird, kann dies zu einer niedrigeren Zeitkonstante führen, da der Kondensator schneller auf Änderungen des Eingangssignals reagiert.

Weitere Faktoren, die die Messgrößen eines RC-Glieds in einer Schaltung beeinflussen können, sind die Kapazitätsbelastung durch andere Komponenten in der Schaltung, die Temperatur und die Spannung an den Anschlüssen des RC-Glieds. Es ist daher wichtig, die Schaltung als Ganzes zu betrachten, um sicherzustellen, dass die Eigenschaften des RC-Glieds den Anforderungen der Schaltung entsprechen.

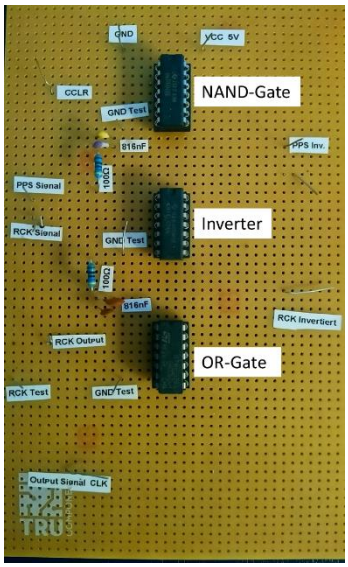


Abbildung 4-2: RC-Gliedern testen

Zur Durchführung der Tests wurde eine Lochrasterplatine verwendet, auf der verschiedene Widerstände und Kondensatoren platziert wurden. Eine Übersicht ist in **Abbildung 4-2** dargestellt. Es wurde festgestellt, dass die beste Kombination aus einem Widerstand mit einem Wert von 560 Ohm und einem Kondensator mit einem Wert von 220pF besteht, um optimale Ergebnisse zu erzielen.

4.2 Schnittstellenbeschreibung

Das entwickelte Messsystem verwendet verschiedene Bauteile, die über spezifische Schnittstellen miteinander verbunden sind. Hierzu gehören drei 74hc165-Register, die miteinander kaskadiert sind, um einen Gesamtspeicherplatz von 24 Bits zu erhalten. Dazu muss Pin 9 (Q7) des ersten Registers mit Pin 10 (DS) des zweiten Registers verbunden werden. Die Ausgangspins (QA-QH) der Counter sind mit den Eingangspins (D0-D7) des Registers verbunden.

Des Weiteren sind drei Counter (74hc590) verbunden, um ebenfalls einen 24-Bit-Speicherplatz zu erhalten. Der Datenblatt zufolge sollten diese Counters miteinander verbunden werden, indem Pin 9 (RCO) des ersten Counters mit Pin 12 (CCKEN) des zweiten Counters verbunden wird. Nach Tests hat sich jedoch gezeigt, dass bei drei Counters ein Fehler aufgetreten ist und die Verbindung anders erfolgen muss. Die Lösung bestand darin, alle Signalverläufe der Counter zu beachten und zu beobachten, wie diese Signale sich verhalten. Am Ende wurde festgestellt, dass Pin 9 (RCO) des ersten Counters mit Pin 11 (CCK) des zweiten Counters verbunden werden sollte.

Das G ist der "Output Enable Input" und muss mit Erde verbunden werden, damit die Werte immer am Ausgang der Counter zur Verfügung stehen. Das CCKEN oder Counter Clock Enable Input wird ebenfalls mit Erde verbunden. Das 5V-Rechteckkomparator-Signal kommt am Pin 13 (RCK) aller Counter an. Das Output-Signal des Quarz-Oszillators (Pin 8) ist mit Pin 11 (CCK) aller Counter verbunden. Pin 1 (NC/Enable) des Quarzes ist mit 5V verbunden, um ständig ein Signal am Ausgangspin zu erzeugen.

Das Counter erhält das verkürzte PPS-Signal-Puls am Pin 10 (CCLR), während das verkürzte Komparator-Signal einmal am GPIO 16 des Raspberry Pi und an allen Pin 1 (SH/LD) des Registers ankommt. Alle Pin 2 (CP) des Registers sind mit GPIO 20 des Raspberry Pi verbunden. Die Pins 15 (CLK INH) des Registers sind mit Erde verbunden, damit sich die Werte am Ausgang ändern können, wenn neue Werte eingehen. Schließlich ist Pin 9 (Q7) des letzten Registers direkt mit GPIO 21 des Raspberry Pi verbunden. Der Schaltplan des Messsystems, der in „Abschnitt 4.2“ dargestellt ist, veranschaulicht die Verbindungen der verschiedenen Bauteile.

4.3 Reihenfolge der Ereignisse

Die Abfolge der Ereignisse, die durchgeführt werden, um die Zeit für eine erfolgreiche Phasenmessung zu ermitteln lässt sich wie folgendes beschreiben.

Zunächst wird das PPS-Signal durch den GNSS-5 erzeugt. Dieses Signal wird dann durch das RC-Glied geführt, um einen neuen Puls zu erzeugen (PPS_{kur}). Dieser Puls wird am Pin 10 an allen Counter angeschlossen, um diese zurückzusetzen. Sobald der Counter zurückgesetzt ist, beginnt es sofort mit dem Zählen und stellt bei jedem positiven Puls aus U_{comp} Werte an den Ausgängen QA-QH bereit.

Um den Raspberry Pi über neue Werte zu informieren, wird das U_{comp}-Signal ebenfalls durch ein RC-Glied geführt, um einen neuen Puls zu erzeugen ($U_{compkur}$). Sobald dieser Puls ausgelöst wird, weiß der Register, dass es neue Werte gibt. Das Register holt sich die bereitgestellten Werte vom Counter und speichert diese.

Der Raspberry Pi hat dann etwa 20ms Zeit, um die Werte vom Register einzulesen, was mehr als ausreichend ist. Durch das $U_{compkur}$-Signal wird das Interrupt am Raspberry Pi getriggert, und das Programm wird gestartet. Nach einer Wartezeit von 10 Mikrosekunden liest das Programm das letzte Bit am GPIO 21 ein. Über GPIO 20 werden 23 Impulse erzeugt, die mit Pin 2 aller Register verbunden sind. Nach jedem Impuls werden die Bits der Register nacheinander an Pin 9 verschoben, sodass sie vom Raspberry Pi am GPIO 21 ebenfalls Bit für Bit gelesen werden können. Am Ende wandelt das Programm die Binärzahlen in die entsprechenden Zeitangaben um und gibt das Ergebnis in einem Textdokument aus, damit es ausgewertet werden kann.

4.4 Schaltplan

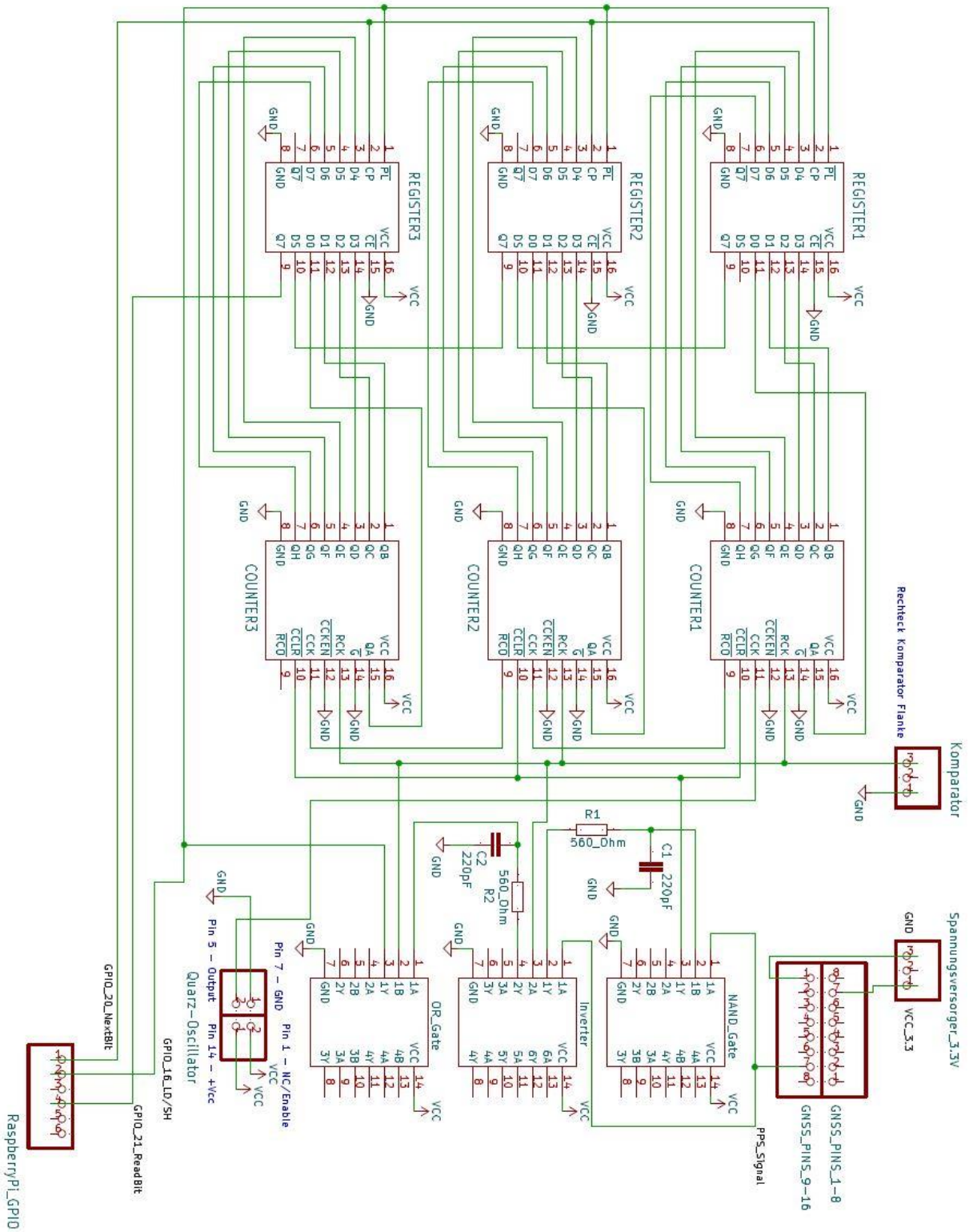


Abbildung 4-3: Schaltplan

5 Software

5.1 Entwicklungsumgebung – Python

Das Praxisprojekt wurde auf der Entwicklungsumgebung von Python unter Raspbian Linux geschrieben. Auf dem Raspberry Pi wird Python durch die integrierte Entwicklungsumgebung Thonny ausgeführt. Python bietet verschiedene Vorteile wie Plattformunabhängigkeit, eine umfangreiche Standardbibliothek, Effizienz und Geschwindigkeit. In Bezug auf Python, ist Thonny eine IDE (Integrated development environment), die speziell für die Entwicklung von Python-Code entwickelt wurde und es erleichtert, Python-Programme zu schreiben, zu testen und auszuführen.

5.2 Das Programm

```
# GPIO-Pins definieren
comp_pin = 16  # Komparator-Pin als Eingang
next_bit = 20  # nächster Bit-Pin als Ausgang
bit_pin = 21   # Bit-Pin als Eingang

# Pins als Eingänge oder Ausgänge konfigurieren
GPIO.setup(comp_pin, GPIO.IN)
GPIO.setup(next_bit, GPIO.OUT)
GPIO.setup(bit_pin, GPIO.IN)

# Leere Liste für die Binärziffern und Ergebnis definieren
list_of_digits = []
list_of_times =[]

# Funktion definieren, die Binärzahl in Fließkommazahl umwandelt
def binary_to_float(sequence):
    return sum([sequence[n] << 23-n for n in range(24)])

# Funktion definieren, die 24 Bit Daten vom Bit-Pin einliest
def read_routine(t_register_clock):
    global list_of_digits

    # Wartezeit definieren
    sleep_time = 10e-6

    # Leere Liste für die Binärziffern definieren
    digits = []

    # Binärdaten vom Bit-Pin einlesen
    for index in range(23):
        digits.append(GPIO.input(bit_pin))
        GPIO.output(next_bit, GPIO.HIGH)
        time.sleep(sleep_time)
        GPIO.output(next_bit, GPIO.LOW)
        time.sleep(sleep_time)
    digits.append(GPIO.input(bit_pin))

    # Binärziffern in eine Fließkommazahl umwandeln
    list_of_digits.append(digits)
    list_of_times.append(binary_to_float(digits) / 2.4576e3)

    # Prüfen, ob (xx) Bit Daten eingelesen wurden
    if len(list_of_digits) == 500:
        print(list_of_digits)
        print(list_of_times)
        sys.exit()

# Interrupt einrichten, der die read_routine() aufruft, wenn der Komparator-Pin eine fallende Flanke erkennt
GPIO.add_event_detect(comp_pin, GPIO.FALLING, callback=read_routine, bouncetime=1)
```

Abbildung 5-1 : Python Programm

Zunächst wird die Bibliothek "RPi.GPIO" importiert, die es ermöglicht, auf die GPIO-Pins des Raspberry Pi zuzugreifen. Die Bibliothek "time" und "sys" werden ebenfalls importiert, um Zeitfunktionen und Systemfunktionen zu nutzen.

Die Funktion "GPIO.setmode(GPIO.BCM)" legt den Modus für die GPIO-Pins fest. "BCM" bedeutet, dass die GPIO-Pins mit ihren GPIO-Nummern benannt werden, anstatt mit ihren physischen Pin-Nummern. Es werden drei GPIO-Pins eingerichtet: Pin 16 wird als Eingang für den Komparator definiert, Pin 20 als Ausgang für den nächsten Bit und Pin 21 als Eingang für die Daten. Es wird eine leere Liste "list_of_digits" definiert, in der die gelesenen Daten gespeichert werden. Eine zweite leere Liste wird erstellt, um die aus den Binärzahlen umgerechneten Ergebnisse in Millisekunden zu speichern. Die Funktion "binary_to_float(sequence)" wandelt eine Sequenz von 24 Bits in eine Fließkommazahl um. Die Funktion durchläuft jede Stelle der Sequenz und multipliziert sie mit 2^(23-n), wobei n der Stelle in der Sequenz entspricht. Die Summe aller Produkte ergibt die Fließkommazahl. Die Funktion "read_routine(t_register_clock)" wird aufgerufen, wenn der Komparator-Pin eine fallende Flanke erkennt. Innerhalb der Funktion wird zuerst eine Pause von 10 Mikrosekunden eingelegt. Dann werden 24 Bit Daten vom Bit-Pin gelesen und als Liste "digits" gespeichert. Die Liste enthält die Daten für jeden der 24 Bits in der Reihenfolge, in der sie empfangen wurden. Mit "GPIO.output(next_bit,GPIO.HIGH)" wird der nächste Bit angefordert, indem der nächste Bit-Pin auf HIGH gesetzt wird. Nach einer weiteren Pause von 10 Mikrosekunden wird "GPIO.output(next_bit,GPIO.LOW)" aufgerufen, um den Pin auf LOW zu setzen und den nächsten Bit anzufordern. Die Liste "digits" wird dann an die Funktion "binary_to_float()" übergeben, um die empfangenen Daten in eine Fließkommazahl umzuwandeln. Der resultierende Wert wird in eine Liste "list_of_times" zusammen mit dem aktuellen Zeitstempel gespeichert. Wenn 500 Bit Daten empfangen wurden, werden die Listen "list_of_digits" und "list_of_times" ausgegeben und das Programm beendet.

Schließlich wird die Funktion "GPIO.add_event_detect(comp_pin,GPIO.FALLING,callback=read_routine, bouncetime = 1)" aufgerufen, um einen Interrupt einzurichten, der die Funktion "read_routine()" aufruft, wenn der Komparator-Pin eine fallende Flanke erkennt. Der Parameter "bouncetime" definiert eine Zeit in Millisekunden, während der weitere Signaleignungen ignoriert werden, um ein Prellen zu vermeiden.

6 Anfertigung eines Platinen-Layouts

6.1 Kicad

In diesem Projekt wird die Open-Source-Software KiCad als Werkzeug zur Erstellung einer Platine verwendet. KiCad ist eine umfassende Lösung für die Erstellung von Schaltplänen, Leiterplatten-Layouts, Generierung von Stücklisten und vieles mehr. Es bietet eine Vielzahl von Tools, die zur Erstellung von Design-Schemata und PCB-Layouts benötigt werden, einschließlich des Schaltplan-Editors EESchema, des PCB-Layout-Editors PCBNew, des Gerber-Viewers GerbView und des Symbol-Editors für Schaltpläne. KiCad ist eine Open-Source-Software, die kostenlos heruntergeladen und genutzt werden kann. Es ist ein Programm für das Design von elektronischen Schaltungen und Leiterplatten, das eine umfassende Lösung für alle Aspekte des Schaltkreis-Designs bietet. KiCad besteht aus verschiedenen Modulen, die für verschiedene Aufgaben verwendet werden können. Der Schaltplan-Editor EESchema ist beispielsweise für die Erstellung von Schaltplänen und Symbolen für Schaltpläne zuständig. Der Leiterplatten-Layout-Editor PCBNew ist hingegen für das Layout der Leiterplatte und die Zuweisung der Bauteile auf der Platine zuständig. Der Gerber-Viewer GerbView wird verwendet, um das Ergebnis des PCB-Layouts zu überprüfen.

6.2 SMD-Platine: Entwurf und Test

Im Rahmen des Projekts wurde eine Platine mit Hilfe des PCB-Editors entworfen, wie in **Abbildung 6-1** und **Abbildung 6-2** zu sehen ist. Die Größe der Platine beträgt ungefähr 60mm x 50mm. Alle Bauteile auf der Platine sind SMD-Bauteile, mit Ausnahme des Quarzes.

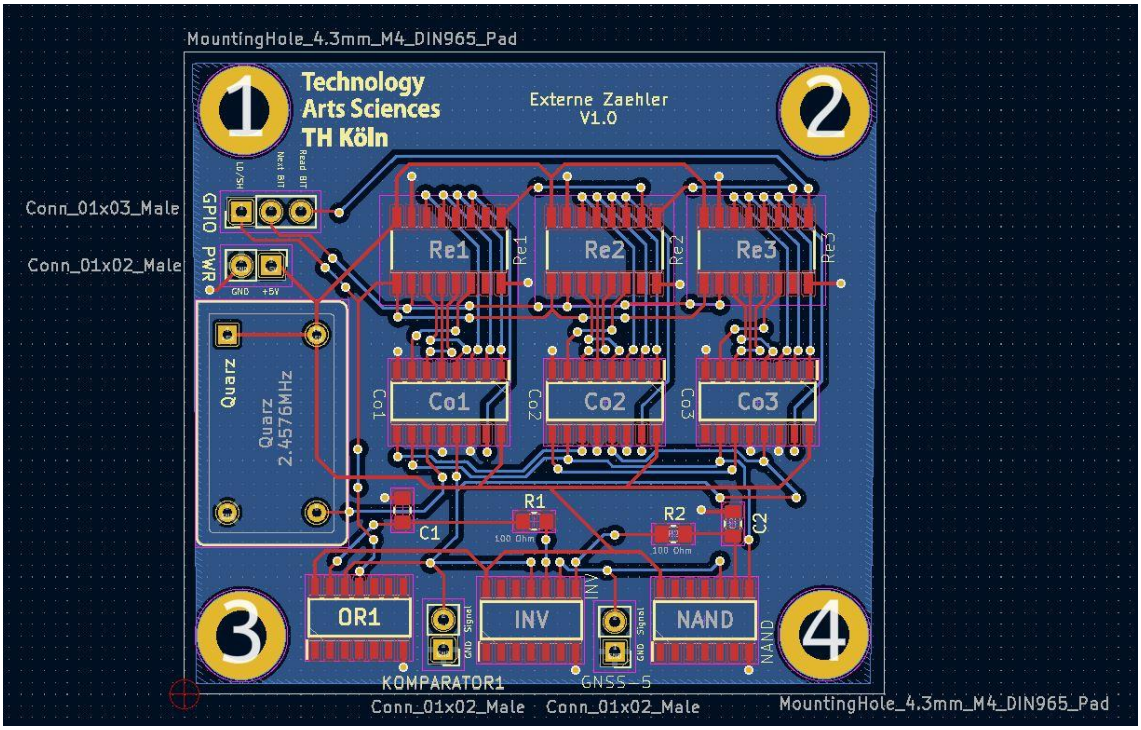


Abbildung 6-1: Die Platine wurde mit Hilfe des PCB-Editors entworfen

Bei der Entwicklung wurden verschiedene Faktoren berücksichtigt, wie beispielsweise die Größe der Leiterbahnen, die Aufteilung der Leiterbahnen und die Anzahl der Schichten. Es wurden insgesamt drei Platinen bestellt und getestet. Leider wurde bei allen Platinen ein erhöhtes Rauschen festgestellt, dessen Ursache nicht klar ist.

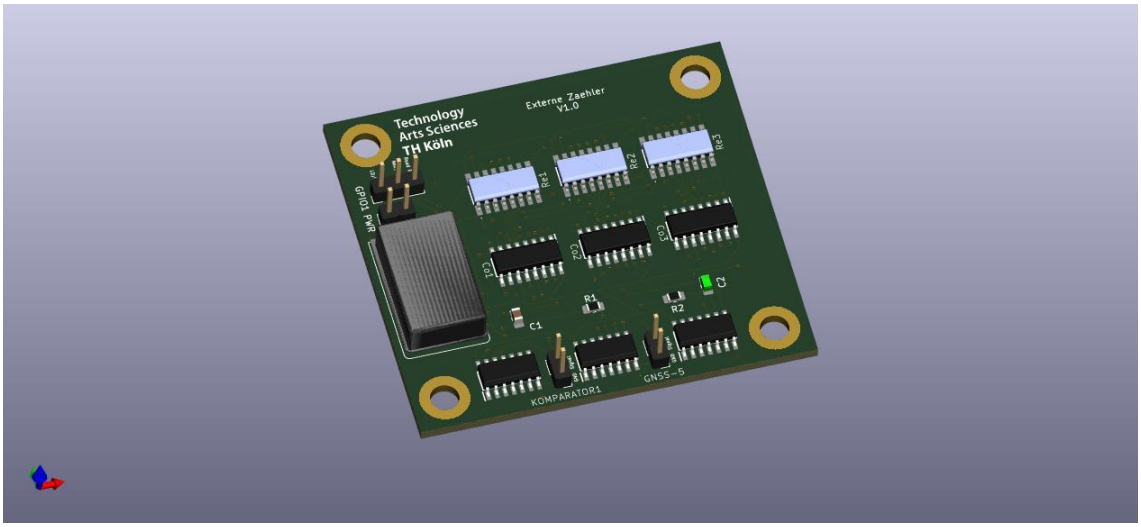


Abbildung 6-2: 3D Ansicht

Trotz des Rauschens hat die Platine alle elektronischen und elektrischen Tests, die mit KiCad durchgeführt wurden, bestanden. Es wird vermutet, dass das Rauschen durch die hohe Kompaktheit der Bauteile auf der Platine verursacht wird, da sie sehr nah beieinander angeordnet sind.

Zusammenfassend kann gesagt werden, dass die Platine erfolgreich mit dem PCB-Editor entworfen wurde, aber aufgrund des Rauschens noch Verbesserungsbedarf besteht. Weitere Untersuchungen müssen durchgeführt werden, um die Ursache des Rauschens zu identifizieren und zu beseitigen.

7 Ergebnisse

Um die Messgenauigkeit des PMU-Geräts zu testen, wurden mehrere Messungen durchgeführt, wobei jeder Messzyklus 5 Sekunden dauerte. Insgesamt wurden 500 Zeitstempel für jede Messung aufgezeichnet. Die Ergebnisse wurden dann in einem Python-Programm analysiert, um die Messdaten zu visualisieren.

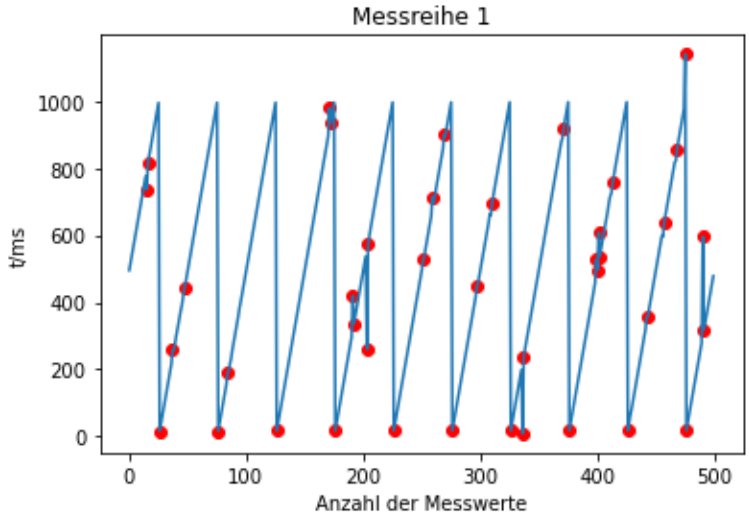


Abbildung 7-1 : Messreihe 1

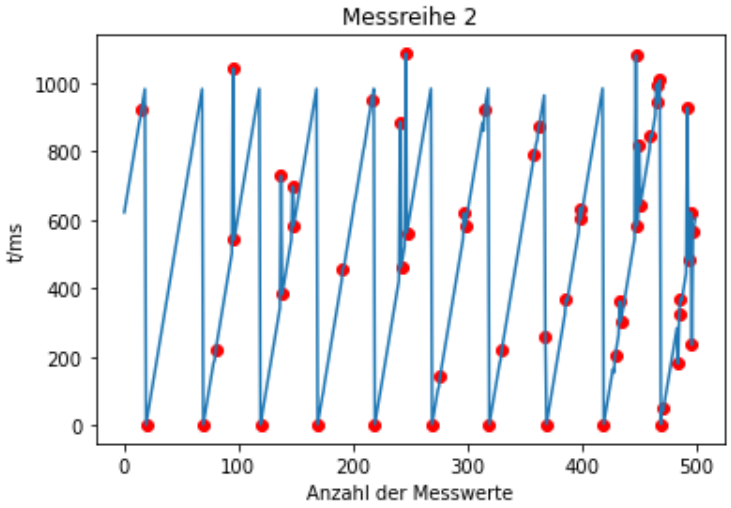


Abbildung 7-2 : Messreihe 2

Die Analyse der Ergebnisse ergab, dass die gemessenen Frequenzen für jede Messung stabil waren und eine lineare Progression der Zeit aufwiesen, wie in **Abbildung 7-1** und **Abbildung 7-2** gezeigt wird. Jedoch gab es in einigen Fällen Ausreißer Werte, die sich signifikant von den vorherigen Werten unterschieden und die Genauigkeit der Messungen beeinträchtigen könnten. Diese Ausreißer Werte wurden als rote Punkte in den Grafiken dargestellt.

Die Analyse der Ergebnisse ergab auch, dass das PMU-Gerät bei Erreichen von 1 Sekunde ein PPS-Signal empfängt, das die Daten zurücksetzt. Dies ist wichtig, um si-

cherzustellen, dass die Messungen in einem festgelegten Zeitintervall durchgeführt werden.

Zusammenfassend zeigen die Ergebnisse der Messungen, dass das PMU-Gerät in der Lage ist, die Frequenz einer gegebenen Phase mit einer durchschnittlichen Genauigkeit von 88% zu messen. Obwohl Ausreißer Werte vorhanden waren, zeigte die Analyse der Ergebnisse zeigte auch, dass die Messgenauigkeit des PMU-Geräts durch eine Optimierung des Messprogramms weiter verbessert werden könnte. Insbesondere wurde festgestellt, dass einige Messungen nach einer Zurücksetzung des Zählers durchgeführt wurden und daher vom aktuellen Programm als falsche Werte interpretiert wurden.

Insgesamt haben die durchgeführten Messungen gezeigt, dass das PMU-Gerät in der Lage ist, präzise Messungen der Frequenz einer gegebenen Phase durchzuführen, was es zu einem nützlichen Instrument für die Überwachung und Steuerung von Stromnetzen und Stromversorgungssystemen macht.

8 Fazit

Das Praxisprojekt hatte zum Ziel, eine Hardware zu entwickeln, die in der Lage ist, jederzeit die Frequenz einer Phase zu bestimmen. Die erfolgreich durchgeführte Vorbereitung bildet die Grundlage für die darauffolgende Bachelorarbeit. Die entwickelte Hardware erfüllt die gestellte Aufgabe, jedoch gibt es noch Potenzial für Verbesserungen.

Die Hardware ist in der Lage, alle Werte einer Phase von Anfang eines PPS-Signals bis zum nächsten PPS-Signal zu messen. Die Genauigkeit der gemessenen Werte ist akzeptabel, aber es besteht die Möglichkeit, noch genauere Werte zu erzeugen, indem man kleinere Impulse erzeugt. Die Software könnte auch noch weiter verbessert werden, indem beispielsweise Logarithmen eingesetzt werden, um Ausreißwerte zu erkennen und aus der Berechnung der Frequenz zu entfernen. Die Zeitstempel aus dem GNSS-5 könnten mit den erzeugten Werten verknüpft werden, um Zeit, Standort und Frequenz zusammenzuführen.

Während der Tests wurde festgestellt, dass der Raspberry Pi nicht in der Lage ist, die Werte "live" auf dem Bildschirm zu übermitteln. Die gemessenen Werte müssen daher erst in einem externen Textdokument geschrieben und dann zur Bewertung aufgerufen werden.

Zusammenfassend kann eine kostengünstige PMU-Version für ca. 200 Euro realisiert werden. Insgesamt hat das Praxisprojekt dazu beigetragen, eine grundlegende Lösung für die gestellte Aufgabe zu finden und weitere Verbesserungen aufzuzeigen, die in der Bachelorarbeit umgesetzt werden können.

9 Literaturverzeichnis

Crompton Instruments. (11. Mai 2023). Von https://www.crompton-instruments.com/pdf/pmu-smart-grid.pdf abgerufen

Electric Power Research Institute. (02. 05 2023). *Phasor Measurement Unit (PMU) Reliability*. Von https://www.epri.com/research/products/0000000000010454 abgerufen

Rihan, M. (21. April 2023). *Research Gate*. Von https://www.researchgate.net/figure/Components-of-a-Phasor-Measurement-Unit_fig1_254050131 abgerufen

Saha, T., & Yoo, S. (23. April 2023). *Phasor measurement units for power system monitoring and protection: A review. Energies, 10(10)*. Von https://doi.org/10.3390/en10101481 abgerufen

Schweitzer Engineering Laboratories. (11. Mai 2023). Von https://selinc.com/solutions/pmu/ abgerufen

Srinivasan, D., & Zhang, Y. (21. 04 2023). *IEEE Access*. Von https://doi.org/10.1109/ACCESS.2020.3001307 abgerufen