

Bachelorarbeit
Tobias Leon Thölen

Entwicklung eines drahtlosen PPG-Sensorsystems zur Bestimmung der Pulswellengeschwindigkeit

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Tobias Leon Thölen

Entwicklung eines drahtlosen PPG-Sensorsystems zur Bestimmung der Pulswellengeschwindigkeit

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Informatik Technischer Systeme*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Roman Kusche
Zweitgutachter: Prof. Dr. Peer Stelldinger

Eingereicht am: 27. März 2025

Tobias Leon Thölen

Thema der Arbeit

Entwicklung eines drahtlosen PPG-Sensorsystems zur Bestimmung der Pulswellengeschwindigkeit

Stichworte

Pulswellenanalyse, Pulswellengeschwindigkeit, Photoplethysmographie, drahtlos, ESP32

Kurzzusammenfassung

In dieser Arbeit wird die Entwicklung eines drahtlosen PPG-Sensorsystems zur Bestimmung der arteriellen Pulswellengeschwindigkeit vorgestellt. Ziel dieser Arbeit ist es, verschiedene Lösungsansätze zur Entwicklung solcher Systeme zu vergleichen und ein drahtloses, anwenderfreundliches System zu entwickeln. Idealerweise soll dieses System in der Lage sein, Pulswellen an den Handgelenken zu erfassen und daraus die Pulswellengeschwindigkeit zu bestimmen. Zu diesem Zweck wurde ein drahtloses Sensorsystem aus mehreren Sensorknoten entwickelt, das Bluetooth Low Energy als Übertragungsprotokoll einsetzt und mithilfe von Photoplethysmographie-Sensoren sowie geeigneten Synchronisationsmechanismen die Sensordaten synchron erfasst und zur Signalverarbeitung an einen PC übermittelt. Dort werden die Sensordaten durch eine in Python implementierte Signalverarbeitung verarbeitet, die unter anderem spezifische Filter und Algorithmen zur Merkmalsextraktion von PPG-Signalen einsetzt. Das drahtlose PPG-Sensorsystem wurde abschließend umfassend durch Systemtests und Probandenmessungen überprüft. Dabei zeigte sich, dass das Ziel der Pulswellenerfassung am Handgelenk aufgrund von Störfaktoren nicht vollständig erreicht wurde. Als alternativer Messansatz wurden Finger-Finger-Messungen durchgeführt und mit Finger-Zeh-Messungen verglichen. Abschließend wurden weitere Vorschläge zur Optimierung und Erweiterung des Systems gemacht.

Tobias Leon Thölen

Title of Thesis

Development of a Wireless PPG Sensor System for the Determination of the Pulse Wave Velocity

Keywords

Pulse wave analysis, Pulse wave velocity, photoplethysmography, wireless, ESP32

Abstract

This bachelor's thesis presents the development of a wireless PPG sensor system for assessing arterial pulse wave velocity. The aim of this thesis is to compare different approaches for developing such systems and to create a wireless, user-friendly solution. Ideally the system should capture pulse waves at the wrist and use them to determine pulse wave velocity. For this purpose, a wireless sensor system consisting of several sensor nodes was developed, which employs Bluetooth Low Energy as the transmission protocol and utilizes photoplethysmography sensors along with suitable synchronization mechanisms to synchronously collect sensor data and transmit it to a PC for signal processing. The sensor data is processed by a signal processing chain implemented in Python, which incorporates specific filters and algorithms for feature extraction from PPG signals. The wireless PPG sensor system was finally comprehensively tested through system tests and subject measurements. It was found that the goal of pulse wave detection at the wrist was not fully achieved due to interference factors. As an alternative measurement approach, finger-to-finger measurements were performed and compared with finger-to-toe measurements. Finally, further suggestions for optimizing and expanding the system were provided.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
1 Einleitung	1
2 Grundlagen	3
2.1 Fachliche Grundlagen	3
2.1.1 Einführung in das Herz-Kreislauf-System	3
2.1.2 Bedeutung der arteriellen Gefäßsteifigkeit	4
2.1.3 Die Pulswellengeschwindigkeit als Prädiktor für arterielle Gefäßsteifigkeit	6
2.1.4 Die Photoplethysmographie als Verfahren zur Erfassung der Puls- welle und Pulswellengeschwindigkeit	7
2.2 Drahtlose Kommunikationskonzepte	8
2.2.1 Bluetooth Low Energy	8
2.2.2 Thread	14
2.3 Synchronisationskonzepte	17
2.3.1 Definition einer Uhr	17
2.3.2 Synchronisationsansätze	19
2.3.3 Timing-Sync Protocol for Sensor Networks (TPSN)	20
2.4 Grundlagen zur Signalverarbeitung	21
3 Systementwicklung	24
3.1 Anforderungen an das Sensorsystem	24
3.2 Aufbau des Sensorsystems	28
3.3 Auswahl der Hardware	29
3.4 Entwicklung der Firmware	36
3.4.1 Entwicklung der PPG-Softwaremodule	36

3.4.2	Entwicklung der Kommunikations-Softwaremodule	39
3.4.3	Herausforderung und Implementierung der Synchronisation	45
3.5	Entwicklung der Signalverarbeitung	53
4	Funktionale Systemüberprüfung	59
4.1	Vergleich von Bluetooth Low Energy und Thread	60
4.2	Überprüfung der Synchronisation	61
4.2.1	Überprüfung der Synchronisation und Funktion der Sensorknoten mittels einer schaltbaren PDLC-Folie	61
4.2.2	Vergleich der globalen Zeitstempel auf den Sensorknoten	63
4.3	Vergleich der drei Datenaggregationsvarianten Interrupts, Timerinterrupts und Polling	65
4.4	Parameter für die PPG-Sensormodule	67
4.4.1	Einfluss der Pulsweite auf die Signale	67
4.4.2	Einfluss des ADC-Bereichs auf die Signale	69
4.4.3	Einfluss des LED-Stroms auf die Signale	69
4.4.4	Einfluss der Abtastrate auf die Signale	71
4.5	Signalverarbeitungsparameter	75
4.5.1	Untersuchung von Resampling	75
4.5.2	Untersuchung von Filtern	76
4.6	Betriebsdauer des Sensorsystems	78
5	Probandenmessungen	79
5.1	Durchführung	79
5.2	Ergebnisse	81
6	Diskussion	87
	Literaturverzeichnis	91
A	Anhang	100
A.1	Verwendete Hilfsmittel	100
	Selbstständigkeitserklärung	101

Abbildungsverzeichnis

2.1	Dargestellt ist eine Pulswelle mit einem ausgeprägten systolischen Maximum (erster Wellenberg) und einem kleineren diastolischen Maximum (zweiter Wellenberg), erfasst von einem Finger eines jungen Probanden.	5
2.2	Der Bluetooth Low Energy Protokollstapel.	9
2.3	Zeigt die Struktur eines GATT-Services mit zwei Characteristics.	13
2.4	Die Abbildung stellt den Thread-Protokollstapel dar mit den zwei unteren auf dem IEEE 802.15.4 Protokoll basierenden Ebenen und dem CoAP Protokoll für die Anwendungsebene.	14
2.5	Exemplarische Darstellung perfekt detektierter Pulswellenmerkmale.	22
3.1	Die Abbildung zeigt eine Skizze des Oberkörpers, der in diesem Bereich liegenden Arterien und die relevanten Mess- und Abstandspunkte.	25
3.2	Der schematische Aufbau des Sensorsystems als Blockdiagramm.	28
3.3	Die Abbildung zeigt den schematischen Aufbau des Sensorsystems mit den physischen Komponenten und den zur Verfügung stehenden Schnittstellen.	33
3.4	Die Abbildung zeigt die Schaltung mit den Pin-Belegungen für einen Sensorknoten.	34
3.5	Die Abbildung zeigt ein Sensorknoten-Gehäuse von Innen.	35
3.6	Die Abbildung zeigt die zwei entwickelten Sensorknoten-Armbänder.	35
3.7	Die Abbildung zeigt die entwickelten Software Module MAX30102 und PPGSensor.	38
3.8	Aufbau des BLEServerFacade-Moduls.	40
3.9	Aufbau des BLEClientFacade-Moduls.	42
3.10	Die Skizze zeigt mit den schwarzen Pfeilen auf, wo zeitliche Verzögerungen auftreten können.	45
3.11	Aufbau eines Sensorknoten-Moduls.	48
3.12	Aufbau eines Empfangsknoten-Moduls.	50
3.13	Darstellung der entwickelten Signalverarbeitungskette.	53

4.1	Einstellungen des Signalgenerators.	61
4.2	Versuchsaufbau mit den zwei Sensorknoten und der PDL-C-Folie.	62
4.3	Die erfassten Rechtecksignale durch die Sensoren.	63
4.4	Ein Ausschnitt aus der Tabelle mit den erfassten Daten des vierten Test- durchlaufs.	64
4.5	Ergebnis mit einer Pulsweite von 69 μ s.	68
4.6	Ergebnis mit einer Pulsweite von 411 μ s.	68
4.7	Ergebnis mit einem Strom von 25 mA (50 %).	70
4.8	Ergebnis mit einem Strom von 37,5 mA (75 %).	70
4.9	Rohdaten bei 50 Hz.	72
4.10	Rohdaten bei 100 Hz.	73
4.11	Rohdaten bei 1000 Hz.	73
4.12	Ein Ausschnitt des Vergleichs zwischen quadratischer und kubischer Inter- polation am Übergang von der systolischen zur diastolischen Phase einer Pulswelle (diastolische Kerbe).	76
4.13	Ein exemplarischer Ausschnitt des Vergleichs verschiedener Filtern am sy- stolischen Maximum einer Pulswelle.	77
5.1	Abgebildet sind die vier Messpositionen: oben links Zeigefinger zu Zeigefin- ger, oben rechts rechter Zeigefinger zu rechtem Zeh, unten links die oberen Seiten der Handgelenke, unten rechts die unteren Seiten der Handgelenke.	80
5.2	Zwei exemplarische Messreihen: eine für die Finger-Zeh-Messung und eine für die Messung auf der unteren Seite der Handgelenke.	82
5.3	Boxplots der Durchschnittswerte: Links der Finger-Zeh-Messungen und rechts der Zeigefinger-Zeigefinger-Messungen.	86

Tabellenverzeichnis

3.1	Zeitabstände für verschiedene Geschwindigkeiten und Distanzen [20, 6, 69].	27
3.2	Erwartungsbereiche für Pulswellengeschwindigkeiten für die entsprechende Altersgruppe [15].	27
3.3	Beispieltabelle mit Zeitstempeln und Sensordaten.	47
4.1	Testparameter für den internen Synchronisationstest mit den Zeitstempeln.	64
4.2	Testparameter für unterschiedliche Pulsweiten.	67
4.3	Testparameter für unterschiedliche ADC-Bereiche.	69
4.4	Testparameter für unterschiedliche LED-Stromstärken.	69
4.5	Testparameter für unterschiedliche Abtastraten.	71
A.1	Verwendete Hilfsmittel und Werkzeuge	100

1 Einleitung

In den letzten Jahren ist die Anzahl der Erkrankungen des Herz-Kreislauf-Systems (engl. Cardiovascular Disease, CVD) stetig gestiegen [52, 70]. Krankheiten in Bezug auf das Herz-Kreislauf-System stellen zudem eine der häufigsten Todesursachen weltweit dar [52]. Daher stehen Untersuchungs- und Überwachungssysteme für das Herz-Kreislauf-System stark im Fokus der Forschung.

Dabei hat sich die arterielle Gefäßsteifigkeit als ein wichtiger Faktor für Endorganschäden und Herz-Kreislauf-Erkrankungen herausgestellt [45]. Diese kann mit der Pulswellengeschwindigkeit (engl. pulse wave velocity, PWV) beschrieben werden, die durch intelligente Sensorsysteme erfasst werden kann.

Bisherige Systeme zur Messung und Analyse der Pulswellengeschwindigkeit werden häufig nur in klinischen Umgebungen eingesetzt. Daher stellt die Erforschung von drahtlosen Systemen, wie Fitnessarmbändern und intelligenten Armbanduhren, die diese wichtigen Parameter erfassen und auswerten können, einen wichtigen Forschungsschwerpunkt dar [60].

Vor diesem Hintergrund wird in dieser Arbeit die Entwicklung eines drahtlosen Sensorsystems vorgestellt, das mithilfe eines Photo-Plethysmographie-Sensors die Pulswellen erfasst und die Sensordaten drahtlos an einen Auswertungscomputer überträgt. Dort werden die Sensordaten durch digitale Signalverarbeitung verarbeitet und die Pulswellengeschwindigkeit als wichtiger Parameter bestimmt.

Einleitend in die Arbeit wird eine Übersicht der für das Sensorsystem relevanten Begriffe und eine Einführung in die Pulsmessung mittels Photo-Plethysmographie-Sensoren gegeben. Zudem werden grundlegende Konzepte der drahtlosen Kommunikation, der Uhrensynchronisation sowie der Signalverarbeitung für die Bestimmung der Pulswellengeschwindigkeit vorgestellt.

Anschließend wird die Entwicklung des Systems detailliert beschrieben. Dazu gehören die Auswahl der Hardware-Komponenten, einschließlich Sensoren und Mikrocontroller, sowie die Implementierung der Firmware und der Signalverarbeitung. Abschließend werden die durchgeführten Versuche erläutert und die Ergebnisse evaluiert und diskutiert. Die Arbeit schließt mit einer Diskussion ab, die einen Ausblick auf zukünftige Forschungsarbeiten und mögliche Weiterentwicklungen des Sensorsystems gibt.

2 Grundlagen

In diesem Kapitel werden die Grundlagen zur Entwicklung eines drahtlosen PPG-Sensorsystems vorgestellt. Als erstes werden medizinisch-technische Grundlagen und Begriffe erläutert, darunter die Photoplethysmographie und die Pulswellengeschwindigkeit. Daraufhin werden drahtlose Kommunikationskonzepte behandelt, darunter die Kommunikationsprotokolle Bluetooth Low Energy und Thread, sowie relevante Grundlagen zur Synchronisation von drahtlosen Sensorsystemen. Das Kapitel schließt mit einer Einführung in wesentliche Grundlagen der Signalverarbeitung ab.

2.1 Fachliche Grundlagen

In diesem Unterkapitel werden die wesentlichen medizinisch-technischen Grundlagen vorgestellt. Zunächst erfolgt eine kurze Einführung in die Funktionsweise des Herz-Kreislauf-Systems. Anschließend werden die arterielle Gefäßsteifigkeit, die Bedeutung der Pulsweite sowie die Photoplethysmographie, als Verfahren zur Erfassung der Pulsweite und Pulswellengeschwindigkeit, erläutert.

2.1.1 Einführung in das Herz-Kreislauf-System

Das Herz-Kreislauf-System (auch kardiovaskuläres System) ist essenziell für das Überleben, da es den Körper über Blutgefäße mit Sauerstoff und Nährstoffen versorgt und Stoffwechselabfälle wie Kohlenstoffdioxid abtransportiert. Zudem verteilt es Hormone und reguliert die Hautdurchblutung. Die Hauptbestandteile sind das Herz, die Blutgefäße, das Blut und die Lunge [27].

Die Blutgefäße spielen eine zentrale Rolle bei der Bewertung des Gesundheitszustands. Sie werden in zwei Hauptsysteme unterteilt: das Hochdrucksystem und das Niederdrucksystem. Zu dem Hochdrucksystem gehören die Aorta (Hauptschlagader), die großen und

kleinen Arterien sowie die Arteriolen. Diese Gefäße transportieren sauerstoffreiches Blut vom Herzen zu den Organen und Geweben. Das Niederdrucksystem umfasst die großen und kleinen Venen, die sauerstoffarmes Blut zurück zum Herzen leiten. Beide Systeme sind über die Kapillaren der Mikrozirkulation miteinander verbunden. In diesen sehr feinen Gefäßen findet der Stoffaustausch statt: Sauerstoff und Nährstoffe werden an die Zellen abgegeben, während Abfallstoffe wie Kohlenstoffdioxid aufgenommen werden [27].

Das Herz, als zentrales Organ des Systems, besteht aus zwei Herzhälften mit jeweils einem Vorhof (Atrium) und einer Kammer (Ventrikel). Sauerstoffarmes Blut wird aus der rechten Herzkammer in den sogenannten kleinen Kreislauf (Pulmonalkreislauf) gepumpt. Dort wird in der Lunge Kohlenstoffdioxid abgegeben und Sauerstoff aufgenommen. Anschließend fließt das oxygenierte Blut in die linke Herzhälfte zurück, von wo es über die linke Herzkammer in den großen Kreislauf (systemischer Kreislauf) gepumpt wird. Während dieser Pumpzyklen wird zwischen der Auswurfphase, der Systole, in der Blut aus den Herzkammern in die Blutgefäße gepumpt wird und der Entspannungs- und Füllungsphase der Herzkammern, die auch als Diastole bezeichnet wird, unterschieden [27].

Der zentrale Indikator für den Zustand des Herz-Kreislauf-Systems ist der Puls. Dieser beschreibt die Druckwelle, die durch die rhythmische Pumpbewegung des Herzens entsteht und sich über die Blutgefäße ausbreitet. Der Puls kann durch seine Frequenz (Schläge pro Minute), seinen Rhythmus (Regelmäßigkeit der Schläge) und die Eigenschaften der Pulswelle analysiert werden. Letztere liefert wichtige Informationen über den Zustand der Gefäße, insbesondere deren Steifigkeit. Zur Messung und Analyse dieser Parameter kommen verschiedene Geräte wie Pulsoximeter, Pulsuhren oder vergleichbare Geräte zum Einsatz [27, 44].

2.1.2 Bedeutung der arteriellen Gefäßsteifigkeit

Mit zunehmendem Alter kommt es zu einer Versteifung der Arterien, da diese aus einem sich abnutzenden Material Elastin bestehen und dieses brüchig wird. Dieser Zustand wird auch als Arteriosklerose bezeichnet. Eine häufige Folge der Arteriosklerose ist die arterielle Gefäßsteifigkeit (engl. arterial stiffness). In diesem Zusammenhang werden auch Begriffe wie Gefäßrigidität oder Gefäßverkalkung verwendet [6, 73].

Insbesondere größere Arterien wie die Aorta, verlieren dadurch ihre Dämpfungs- und Pufferfunktion, was zu einer erhöhten Nachlast führt - der Druck, den das Herz aufbringen muss, um Blut durch die Gefäße zu pumpen. Dies kann zu schwerwiegenden Folgen

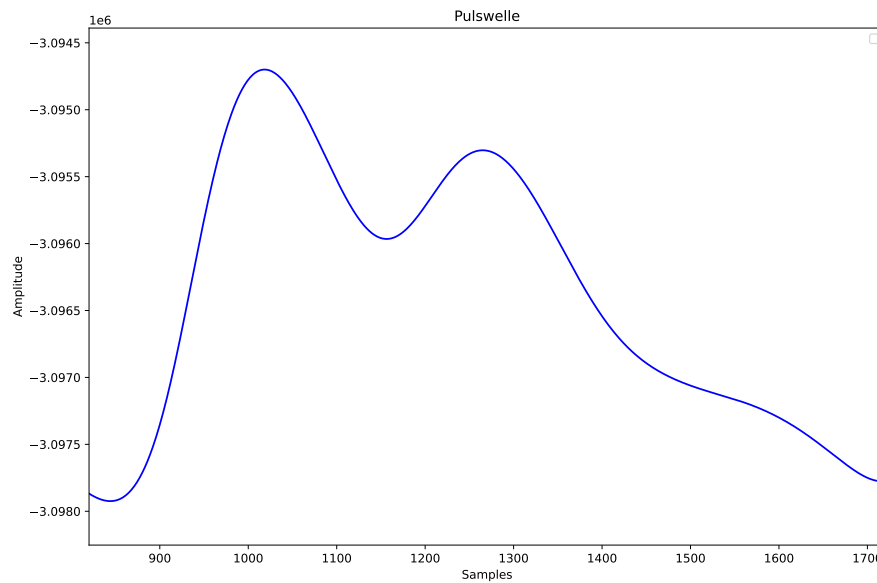


Abbildung 2.1: Dargestellt ist eine Pulswelle mit einem ausgeprägten systolischen Maximum (erster Wellenberg) und einem kleineren diastolischen Maximum (zweiter Wellenberg), erfasst von einem Finger eines jungen Probanden.

für das Herz-Kreislauf-System führen, darunter Linksherzhypertrophie, Herzinsuffizienz und ein erhöhtes Risiko für Schlaganfälle, Demenz oder Niereninsuffizienz. [6] Daher ist die Analyse der arteriellen Gefäßsteifigkeit mithilfe der Pulswelle von sehr hoher Bedeutung.

Die Pulswelle entsteht durch die rhythmische Pumpbewegung des Herzens und setzt sich aus zwei Komponenten zusammen: der initialen Welle, die vom Herzen aus durch die Arterien läuft und einer Reflektionswelle. Die Reflektionswelle entsteht durch eine Reflektion einer vorherigen initialen Pulswelle durch den Gefäßwiderstand in äußeren (Peripherie), kleineren Gefäßen (Mikrozirkulation). Zusammen formen diese beiden Wellen die Pulswelle. Die Amplitude dieser resultierenden Pulswelle wird auch als systolisches Maximum bezeichnet, der Punkt an dem der Blutdruck am höchsten ist. Bei einer fortgeschrittenen Gefäßsteifigkeit entsteht die Reflektionswelle möglicherweise früher oder stärker, dies spiegelt sich direkt in einem höheren systolischen Druck wider, was auch als Pulsdruckamplifikation bezeichnet wird. Die Abbildung 2.1 zeigt eine exemplarische Pulswelle mit ihren charakteristischen Wellenhügeln [6, 36].

2.1.3 Die Pulswellengeschwindigkeit als Prädiktor für arterielle Gefäßsteifigkeit

Mithilfe der Pulswelle können verschiedene Parameter bestimmt werden, die als Prädiktor dienen um die arterielle Gefäßsteifigkeit zu beschreiben. Einer dieser Prädiktoren ist die Pulswellengeschwindigkeit (engl. pulse wave velocity, PWV). Sie beschreibt die Geschwindigkeit, mit der sich eine Pulswelle in Metern pro Sekunde (m/s) vom Herzen durch die Arterien des Körpers ausbreitet [6, 44]. Die Pulswellengeschwindigkeit kann als Prädiktor für die arterielle Gefäßsteifigkeit eingesetzt werden, da sie in inverser Beziehung zur Dehnbarkeit der Arterien steht. Sie kann durch die Bramwell-Hill-Gleichung beschrieben werden [6, 36]. Bei jungen oder gesunden Patienten ist die Pulswellengeschwindigkeit aufgrund der geringen Gefäßsteifigkeit und der noch vorhandenen Elastizität der Gefäße typischerweise niedriger und steigt mit einem höheren Alter und der damit verbundenen höheren Gefäßsteifigkeit an. Die Pulswellengeschwindigkeit liegt bei Kindern bei ungefähr 6 m/s und kann mit höherem Alter auf über 10 m/s ansteigen. Werte von über 10 m/s können zu dem auf Endorganschäden hinweisen [44].

Die Pulswellengeschwindigkeit hängt nicht nur vom Alter ab, sondern auch von der Wahl der Messorte und dem verwendeten Messverfahren. Ein häufig verwendeter Messbereich ist die Strecke zwischen der Halsschlagader (Karotisarterie) und der Oberschenkelarterie (Femoralarterie), bekannt als cfPWV (carotid-femoral pulse wave velocity). Der Vorteil an diesen Messpunkten ist, dass die Distanz in Metern abgemessen werden kann und der Weg die wichtigste und die durch Verschleiß am meisten betroffene Hauptarterie Aorta abdeckt [6, 36]. Ein alternativer Messort ist an der Armarterie (Brachialarterie). Diese deckt jedoch nicht einen Großteil der Aorta ab und die Distanzmessung zwischen den Armen kann nur annähernd abgeschätzt werden. Darüber hinaus kann der Messort dazu führen, dass die Pulswellengeschwindigkeit in den peripheren Gefäßen höher ausfällt. Dies kommt, wie oben bereits beschrieben durch die geringere Elastizität der Peripherie zustande [36].

Zur Erfassung der Pulswelle stehen verschiedene nicht-invasive Messverfahren zur Verfügung, wie tonometrische, piezoelektrische, sonografische, oszillometrische oder auf Magnetresonanztomographie basierende Messverfahren [6]. Viele dieser Verfahren erfordern jedoch stationäre Geräte und geschultes Fachpersonal, was ihre Anwendungsmöglichkeiten einschränkt. Eine kompaktere und kostengünstigere Alternative stellt die Photoplethysmographie (PPG) dar [2]. Daher wird in dieser Arbeit die Entwicklung eines drahtlosen Sensorsystems vorgestellt, das auf der Photoplethysmographie basiert.

2.1.4 Die Photoplethysmographie als Verfahren zur Erfassung der Pulswelle und Pulswellengeschwindigkeit

Die Photoplethysmographie ist ein auf Licht basierendes und nicht-invasives Messverfahren zur Erfassung von Änderungen des Blutvolumens in der Haut. Mithilfe einer Lichtquelle, wie einer LED, wird eine Hautstelle durchleuchtet. Die Lichtabsorption in den Gefäßen, verursacht durch Unterschiede im Gehalt an gesättigtem und ungesättigtem Hämoglobin, wird anschließend von einem Photodetektor gemessen. Die Spektralbereiche zur Erfassung der Lichtreflexionen liegen typischerweise im Rotlichtbereich (660–760 nm) und im Infrarotbereich (810–940 nm) [2, 32, 48].

Die Photoplethysmographie wird außerdem in zwei gewebsoptische Messprinzipien unterteilt. Einmal der Transmissionspulsoxymetrie und der Reflexionspulsoxymetrie. Bei der Transmissionspulsoxymetrie wird das Gewebe durch eine Lichtquelle durchleuchtet und auf der gegenüber liegenden Seite die Lichtintensität durch ein Photodetektor erfasst. In der Reflexionspulsoxymetrie hingegen sind die Lichtquelle und der Photodetektor nebeneinander auf einem Sensormodul verbaut. Ein Nachteil dieser Methode besteht darin, dass der Photodetektor auch Shuntlicht detektieren kann, also Licht, das ohne Interaktion mit dem Gewebe reflektiert wird. Dennoch bietet diese Methode Vorteile wie die flexible Platzierung des Sensors an verschiedenen Hautstellen und eine geringere Anfälligkeit für Bewegungsartefakte [32, 42, 48].

Die Photoplethysmographie ist insgesamt sehr störanfällig. Neben Bewegungen und Shuntlicht können auch äußere Lichteinflüsse, geringe Änderungen an der Hardware oder Veränderungen in der Sensorbefestigung zu Störungen führen. Daher sind Filter für photoplethysmographie-basierte Sensorsysteme unerlässlich. Sie dienen dazu, niederfrequente und hochfrequente Signalanteile herauszufiltern und das resultierende Photoplethysmogramm zu bereinigen. Insbesondere ermöglichen sie die Trennung des pulsierenden Wechselanteils (engl. alternating current, AC) vom Gleichanteil (engl. direct current, DC). Der Wechselanteil entsteht durch die Blutpulsation in den Arterien und wird von der Herzfrequenz beeinflusst. Der Gleichanteil beschreibt einen konstanten Signalanteil, der auf die nicht pulsierende Hautreflexion, die Atmung (Respiration) und vasomotorische Aktivitäten zurückzuführen ist. [2, 48, 32, 42].

Aus dem Photoplethysmogramm lassen sich wichtige Schlüsselpunkte und Parameter ableiten, wie das oben bereits erwähnten systolische Maximum (engl. systolic peak) oder der Beginn der Pulswelle (engl. onset). Diese dienen als Grundlage zur Berechnung weiterer

Parameter, darunter die Pulsweite, die Pulswellengeschwindigkeit, der Augmentation Index oder die Pulswellenlaufzeit. Alle diese Werte sind entscheidende Indikatoren für den Zustand des Herz-Kreislauf-Systems. Der Fokus dieser Arbeit liegt in der Bestimmung der Pulswellengeschwindigkeit [48, 42].

2.2 Drahtlose Kommunikationskonzepte

Neben den medizinisch-technischen Grundlagen werden im Folgenden die für diese Arbeit relevanten drahtlosen Kommunikationskonzepte erläutert. Der Fokus liegt auf Protokollen aus dem Bereich der Wireless Personal Area Networks (WPANs), insbesondere der IEEE-802.15-Familie. Vorgestellt werden das Bluetooth Low Energy (BLE)-Protokoll sowie als Alternative das auf IEEE 802.15.4 basierende Thread-Protokoll. Diese Protokolle sind aktuelle und relevante Standards für die Entwicklung drahtloser Sensorsysteme (engl. wireless sensor networks, WSNs), die für geringen Energieverbrauch und kurze Übertragungsdistanzen optimiert sind [67, 40, 33].

2.2.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE), auch als Bluetooth LE bekannt, ist ein Protokoll der Bluetooth-Protokollfamilie. Seit der Bluetooth-Version 4.0 wird Bluetooth mit der Zusatzbezeichnung Low Energy versehen. Die Bluetooth-Version 4.0 hat im Gegensatz zur ihren Vorgängerversionen (Bluetooth 1.0 bis 3.0), die als Bluetooth Basic Rate (BR) und Bluetooth Enhanced Data Rate (EDR) bekannt sind, die positiven Eigenschaften von höherer Datenrate bei einem reduziertem Energieverbrauch kombiniert. Um diesen Kompromiss zu erreichen, beschränkt sich Bluetooth Low Energy auf eine geringere Reichweite von wenigen Metern. [39].

Die Spezifikation von Bluetooth Low Energy werden umfänglich in der Bluetooth Core Specification definiert. Darin werden unter anderem die verwendeten Technologien, Protokolle und Aufgaben der jeweiligen Netzwerkschichten beschrieben [7]. Die Bluetooth Core Specification definiert BLE als ein vollständiges Kommunikationsprotokoll (Full Stack), das alle Ebenen des OSI-Referenzmodells abdeckt, wie Abbildung 2.2 zeigt.

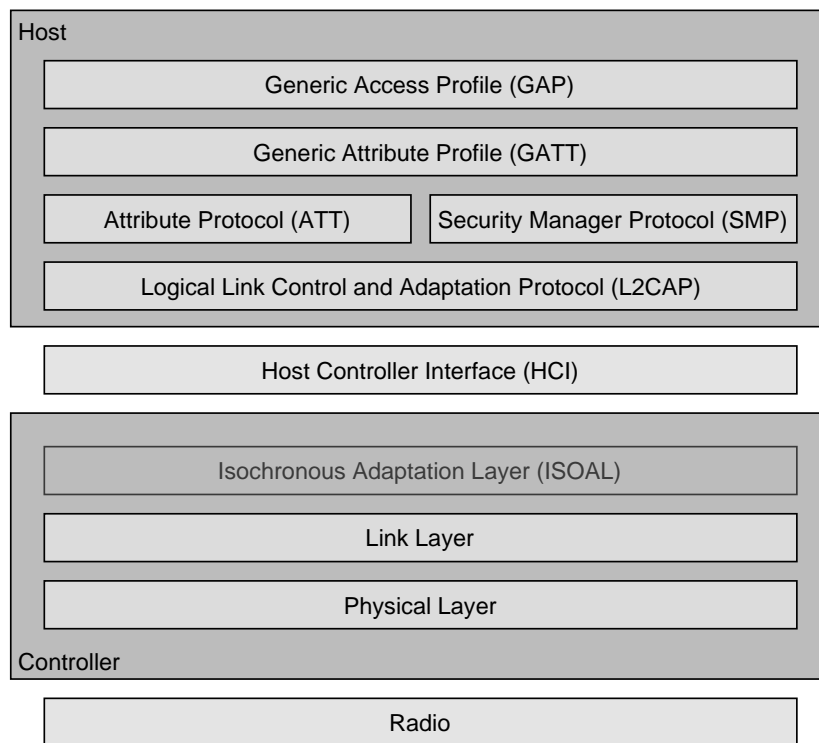


Abbildung 2.2: Der Bluetooth Low Energy Protokollstapel.

BLE-Protokollstapel

Physikalische-Ebene

Auf der physikalischen Ebene werden Eigenschaften wie das unlicenzierte 2,4-GHz-ISM-Frequenzband (Industrial Scientific Medical) und die 40 Funkkanäle definiert. Diese sind mit einer Bandbreite von jeweils 2 MHz im Frequenzbereich von 2,4 GHz bis 2,4835 GHz verteilt. Dabei werden drei Frequenzkanäle als Werbekanäle (Advertising Channels) verwendet und die restlichen 37 Kanäle als Datenübertragungskkanäle [7, 23, 65].

Auf allen Kanälen kommt die Gaussian Frequency Shift Keying (GFSK) Modulation zum Einsatz, um Funksignale zu erzeugen und zu übertragen. Durch Frequenzverschiebung (Shifting) innerhalb eines Kanals, zum Beispiel bei 2402 MHz wird die Codierung von Einsen und Nullen ermöglicht. Um Störungen zu minimieren, wird zusätzlich eine Gauss-Filterung angewendet [7]. Bluetooth Low Energy unterstützt verschiedene Modulationsvarianten, mit denen eine Datenrate von 1 Mbit/s oder 2 Mbit/s erreicht werden können [7, 23, 65].

Link-Layer-Ebene

Die Link-Layer-Ebene ist direkt mit der physikalischen Schicht verbunden und kombiniert sowohl hardware- als auch softwarebasierte Funktionen. In dieser Schicht werden zudem verschiedene Kommunikationsrollen definiert:

- **Master:** Steuert die Verbindung und koordiniert den Datenaustausch.
- **Slave:** Antwortet auf Anfragen des Masters und sendet Daten gemäß dessen Steuerung.
- **Advertiser:** Sendet regelmäßig Werbepakete (Advertising Packets), um eine Verbindung mit anderen Geräten herzustellen.
- **Scanner:** Empfängt und verarbeitet Werbepakete, um verfügbare Geräte zu erkennen.

Zusätzlich führt die Link-Layer-Schicht verschiedene Funktionen zur Sicherstellung einer zuverlässigen und sicheren Kommunikation aus, darunter:

- Preamble, Access Address und Air Protocol Framing
- Cyclic Redundancy Check (CRC) Generierung und Verifikation
- Data Whitening
- Random Number Generation
- Advanced Encryption Standard (AES) [65]

Host Controller Interface (HCI)

Das Host Controller Interface (HCI) bietet eine standardisierte Schnittstelle zur Kommunikation zwischen der Controller- und der Host-Ebene [7]. HCI definiert Kommandos und Events zur Umwandlung von Rohdaten in Datenpakete und umgekehrt [7, 65].

Logical Link Control and Adaptation Protocol (L2CAP)

Das Logical Link Control and Adaptation Protocol (L2CAP) dient dem Multiplexing verschiedener Protokolle und bildet eine Schnittstelle zwischen den oberen Protokollen ATT und SMP sowie der unteren Link-Layer-Ebene. Dabei werden für die obere Ebenen Service Data Units (SDUs) zusammengefügt (Reassembly) und für die untere Ebene Protocol Data Units (PDUs) durch Fragmentierung (Segmentation) zerteilt [7, 65].

Attribute Protocol (ATT)

Auf der Ebene des Attribute Protocols (ATT) werden zwei weitere Rollen eingeführt:

- **Client:** Fordert Daten vom Server an oder erhält diese.
- **Server:** Stellt Daten (Attribute) bereit.

Diese Rollen sind unabhängig von der separaten Master-Slave-Struktur und können flexibel eingenommen werden. Das bedeutet, dass ein Teilnehmer sowohl als Server als auch als Client agieren kann, unabhängig davon, ob dieser ein Master oder Slave ist [7, 23, 65].

Attribute sind Datenstrukturen mit eindeutigen Identifikatoren (Universally Unique Identifier, UUID), Berechtigungen und Werten. UUIDs sind entweder standardisierte 16-bit- oder benutzerdefinierte 128-bit-Identifikatoren, mit denen Attribute eindeutig adressiert werden. Die Attribute werden in Tabellen hinterlegt und es kann mithilfe ihrer Identifikatoren auf diese zugegriffen werden. Das übergeordnete Generic Attribute Profile (GATT) verwaltet die internen Daten. Für den Zugriff auf Attribute stehen vier Operationen zur Verfügung [7, 23]:

- **Reading:** Attribute-Werte auslesen.
- **Writing:** Attribute-Werte setzen.
- **Notifications:** Der Server sendet Werteänderungen an den Client (ohne Bestätigung).
- **Indications:** Der Server sendet Werteänderungen an den Client (mit Bestätigung).

[7, 23]

Generic Attribute Profile (GATT)

Das Generic Attribute Profile (GATT) definiert auf ATT basierende, höherwertige Datentypen. Die Basis dieser Datentypen bilden ein oder mehrere Services, die eine oder mehrere Characteristics gruppieren [7, 23].

In Characteristics sind Eigenschaften (Properties), Werte (Values) und Deskriptoren (Descriptors) enthalten. Die Eigenschaften beschreiben, mit welchen Operationen (read, write, notifications oder indications) auf die Werte zugegriffen werden kann. Deskriptoren enthalten Metadaten zu einer Characteristic, z. B. den Client Characteristic Configuration Descriptor, der für Server mit der Notifications-Funktion erforderlich ist [7].

Die Bluetooth Special Interest Group hat für verschiedenste Standardanwendungen Profile definiert, unter denen vordefinierte Services zu finden sind, auf die mit 16-bit UUIDs zugegriffen werden kann. Es können aber auch eigene Services und Characteristics selbst definiert werden, um Bluetooth Low Energy für neue Anwendungen einzusetzen. Diese werden mit 128-bit-UUIDs verwaltet [7, 65]. Ein Beispiel für einen GATT-Service ist ein Bluetooth-Low-Energy-Server, der Pulswellendaten bereitstellt. Abbildung 2.3 veranschaulicht dieses Szenario. Wie zu sehen ist, besitzt der Service in diesem Fall zwei Characteristics: Infrarotdaten und Rotdaten.

Generic Access Profile (GAP)

Das Generic Access Profile (GAP) dient als Brücke zwischen der Anwendungsebene und den unteren Protokollen des Bluetooth Low Energy Stacks. GAP definiert Rollen und Verfahren zur Geräteerkennung sowie zum Aufbau und zur Verwaltung von Verbindungen. Die möglichen Rollen sind:

- **Advertiser (Broadcaster)**: Sendet Daten ohne eine aktive Verbindung.
- **Scanner (Observer)**: Empfängt Daten von Advertisern, ohne eine Verbindung herzustellen.
- **Peripheral (Slave)**: Baut eine Verbindung zu einem Central auf und ermöglicht bidirektionale Kommunikation.
- **Central (Master)**: Initiiert Verbindungen zu Peripherals und Daten mit diesen austauschen.

[65, 23].

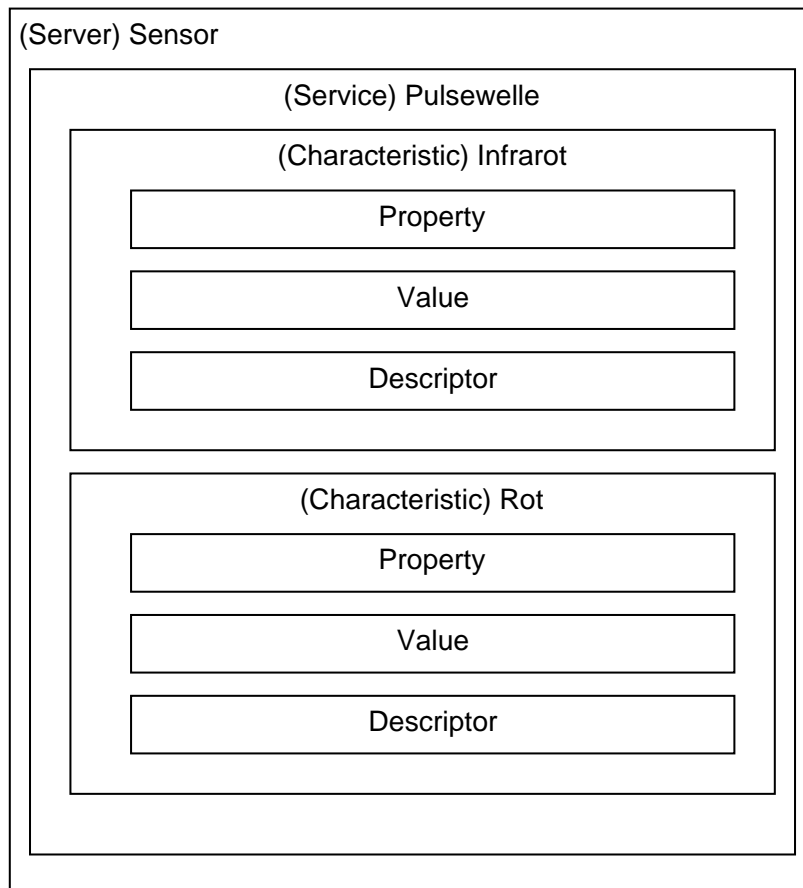


Abbildung 2.3: Zeigt die Struktur eines GATT-Services mit zwei Characteristics.

2.2.2 Thread

Thread ist ein Kommunikationsprotokoll, das für Widerstandsfähigkeit, Kosteneffizienz, einfache Integration, niedrigen Energieverbrauch und hohe Sicherheit entwickelt wurde. Zur Kommunikation verwendet es IPv6-Routing und bildet ein Mesh-Netzwerk. Die Mesh-Funktionalität ermöglicht eine Erweiterung des Netzwerks um zusätzliche Teilnehmer und eine Vergrößerung der Reichweite. Hauptsächlich kommt es im Bereich Smart Home und IoT-Geräte zum Einsatz und steht in direkter Konkurrenz zum ZigBee Protokoll [66].

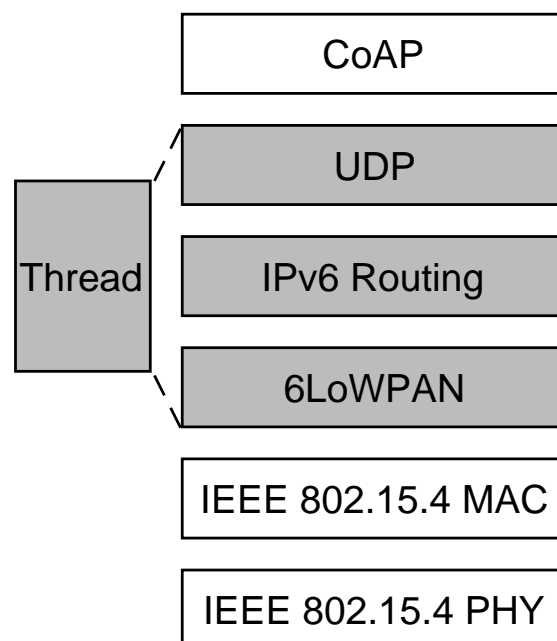


Abbildung 2.4: Die Abbildung stellt den Thread-Protokollstapel dar mit den zwei unteren auf dem IEEE 802.15.4 Protokoll basierenden Ebenen und dem CoAP Protokoll für die Anwendungsebene.

Thread ist kein Full Stack Protokoll und deckt nicht alle Ebenen des OSI-Referenzmodells ab. Es definiert von diesem nur die Ebenen Netzwerk und Transport, siehe dazu Abbildung 2.4. Die darunterliegenden Schichten basieren auf IEEE 802.15.4, welches die physikalische und die MAC-Schicht beschreibt [53].

Thread Protokollstapel

PHY (Physical) Layer

Auf der Physikalischen Ebene wird als Frequenzband 2450 MHz definiert und ist in 16 Kanäle aufgeteilt. Thread verwendet das O-QPSK-Modulations-verfahren und erreicht eine Datenübertragungsrate von bis zu 250 Kbps. Die maximale Reichweite eines Funk-signals beträgt 30 Meter. Ein Netzwerk kann jedoch mit bis zu 36 Hops und mehr als 250 Teilnehmern skaliert werden [66, 31].

MAC (Data Link) Layer

Die MAC-Schicht deckt Aufgaben, wie die zuverlässige Nachrichtenübertragung, Fehlerkorrektur und Überlastungskontrolle ab [31]. Sie unterstützt Mechanismen des IEEE 802.15.4-Standards, darunter das Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) Verfahren, zur Vermeidung von Kollisionen. Dieses stellt sicher das sich mehrere Thread-Teilnehmer das 2,4 Ghz Frequenzband teilen können und darauf warten müssen das ein Frequenzkanal frei wird. Zusätzlich kommen Acknowledgment- und Retry-Mechanismen zum Einsatz [66].

6LoWPAN

6LoWPAN steht für IPv6 over Low-power Wireless Personal Area Network und ist ein Standard der als Vermittler-Ebene zwischen der IP-Routing Ebene und der IEEE 802.15.4 MAC Ebene dient [31]. Auf dieser Ebene werden IPv6-Pakete zu IEEE 802.15.4 MAC Ebene konforme Pakete fragmentiert oder von dieser zurück zu IPv6-Paketen zusammengesetzt. Dabei werden IPv6 Header und UDP Header möglichst stark komprimiert [66].

IP Routing

Thread nutzt IPv6-Routing, wobei IP-Adressen über DHCPv6 automatisch zugewiesen werden. Jeder Router speichert Routing-Informationen in einer Tabelle, um eine zuverlässige Pfadfindung zu gewährleisten. Dabei kommen Protokolle aus der Distance-Vector-Routing-Familie zum Einsatz, die durch kompakte Nachrichten einen effizienten Austausch von Routing-Informationen ermöglichen [31].

Transport Layer/User Datagram Protocol

Als Transport Protokoll wird UDP eingesetzt, da es keinen großen overhead und einen schneller und größeren Datenfluss ermöglicht. Außerdem wird als Anwendungsprotokoll in Thread CoAP eingesetzt und dies erfordert als darunterliegendes Transportprotokoll UDP [66].

Thread Gerätetypen

Im Thread-Protokoll werden verschiedene Gerätetypen unterschieden, die jeweils spezifische Aufgaben im Netzwerk übernehmen. Diese werden im folgenden näher erläutert. Grundlegend werden die Thread-Geräte in zwei Gerätearten eingeteilt, die vollständigen Thread-Geräte (Full Thread Devices, FTDs) und die Minimalen Thread-Geräte (Minimal Thread Devices, MTDs) [47, 63].

Die vollständigen Thread-Geräte sind Geräte in einem Thread-Netzwerk, die üblicherweise eine dauerhafte Stromversorgung besitzen und Aufgaben, wie die Verwaltung des Netzwerks oder Routing übernehmen. Ihre Funkmodule sind dafür entsprechend immer aktiv. FTDs lassen sich weiter in die Gerätetypen Router, Router Eligible End Devices (REEDs) und Full End Devices (FEDs) unterteilen [47, 63].

Von diesen Gerätetypen nehmen die Router zentrale Aufgaben wahr, sie leiten die IPv6-Pakete in dem Thread-Netzwerk weiter. REEDs sind Endgeräte, die bei Bedarf zu Routern hochgestuft werden können. Dadurch übernehmen sie dynamisch Routing-Aufgaben und tragen zur Erweiterung des Netzwerks bei, beispielsweise wenn ein Router außerhalb der Reichweite eines Thread-Gerätes liegt. Die FEDs hingegen sind Endgeräte die nicht zu Routern hochgestuft werden können [47, 63].

Neben den verschiedenen Thread-End-Geräten werden die Router noch in zwei weitere besondere Router-Typen aufgeteilt, die Thread-Leader-Router und die Thread-Border-Router. Der Leader-Router wird automatisch aus den verfügbaren Routern gewählt und übernimmt zentrale Verwaltungsaufgaben, darunter die Hochstufung von REEDs zu vollwertigen Routern. Sollte der Leader ausfallen, wählt das Netzwerk automatisch einen neuen Leader, um die Stabilität zu gewährleisten und Single Points of Failure zu vermeiden [47, 63].

Die Border-Router sind Router die Nachrichten zwischen einem Thread-Netzwerk und einem externen Thread-Netzwerk, wie zum Beispiel WLAN austauschen können. Somit kann nicht nur ein großes Mesh-Netzwerk, sondern auch eine einfache Anbindung zu Cloud-Diensten oder weiteren Internet-Diensten realisiert werden [47, 63].

Die minimalen Thread-Geräte, sind in der Regel batteriebetrieben und kommunizieren nur über übergeordnete FTDs. Sie lassen sich ebenfalls weiter aufteilen in die Minimal End Devices (MEDs) und die Sleepy End Devices (SEDs). Die Minimal End Devices sind Geräte die einem Router untergeordnet sind und nur über diesen Nachrichten mit anderen Geräten austauschen können, ihr Funkmodul ist immer aktiviert. Die Sleepy End Devices hingegen können ihre Funkmodule zwischenzeitig ausschalten, wenn diese sich im Idle-Modus befinden [47, 63].

2.3 Synchronisationskonzepte

Für die Entwicklung eines drahtlosen PPG-Sensorsystems zur Bestimmung der Pulswellengeschwindigkeit ist ein grundlegendes Verständnis der Begriffe und Mechanismen zur Daten- und Zeitsynchronisation erforderlich. Deshalb werden in diesem Abschnitt wesentliche Konzepte zur Synchronisation in drahtlosen Sensornetzwerken erläutert, darunter Uhren (clock), Echtzeituhren (Real-Time Clock, RTC) sowie verschiedene Synchronisationsprotokolle.

2.3.1 Definition einer Uhr

Eine Uhr (clock) in einem eingebetteten System wird häufig als Echtzeituhr (Real-Time Clock, RTC) realisiert. Diese basiert auf einem Quarzoszillator, der mithilfe eines Schwingquarzes ein elektrisches Signal mit einer bestimmten Frequenz erzeugt. Das Signal dient dazu, einen Zähler periodisch zu inkrementieren [58].

Neben den physikalischen Echtzeituhren werden auch Softwareuhren eingesetzt, die auf diesen basieren. Sie werden in festen Zeitintervallen f der Echtzeituhr aktualisiert, was durch Interrupts gesteuert wird. Jeder Interrupt löst eine Inkrementierung eines Zählers aus, der als Softwareuhr dient. Die einzelnen Zähler Schritte werden dabei als Ticks bezeichnet [58].

In einer idealen Umgebung würde eine Uhr C für eine Maschine p exakt der absoluten Zeit t entsprechen, also $C_p(t) = t$. Die Differenz zwischen der gemessenen Zeit einer Uhr und der idealen Zeit kann als $C_p(t) - t \leq a$ beschrieben werden, wobei im Idealfall $a = 0$ gilt. Dies kommt jedoch nicht zustande, da die Echtzeituhren in der realen Umgebung unterschiedliche Frequenzen aufweisen (clock drift) und zu unterschiedlichen Zeiten starten (clock offset) [58, 71].

In diesem Zusammenhang bestimmt die clock drift rate den Rahmen, in dem die Echtzeituhr von der tatsächlichen perfekten Anfangszeit über die Zeit abdriftet. Eine gute Echtzeituhr hat typischerweise eine clock drift rate im Bereich von 10^{-6} , also einer Mikrosekunde pro Sekunde Zeitabweichung, was 60 Mikrosekunden pro Minute macht [58].

Die Zeitmessung eines Netzknotens unter Berücksichtigung von clock drift und clock offset kann mathematisch beschrieben werden als:

$$C_i(t) = \theta + f \cdot t. \quad (2.1)$$

Hierbei steht θ für den clock offset (Phasendifferenz) und f für den clock drift (Frequenzdifferenz, auch als clock skew bezeichnet). Das bedeutet, dass jede Uhr in einem Netzwerk zu einem unterschiedlichen Zeitpunkt θ startet und die Uhr mit dem Faktor f über die Zeit angepasst wird. Die Beziehung zwischen den Uhren zweier Netzknoten kann mit folgender Formel genauer beschrieben werden [71, 56]:

$$C_B(t) = \phi^B + f^{AB} \cdot C_A(t). \quad (2.2)$$

Da sich die Uhren mit der Zeit aufgrund ihrer unterschiedlichen physikalischen Eigenschaften immer weiter auseinander bewegen, ist eine regelmäßige Synchronisation erforderlich [71, 56, 51].

Neben diesen zeitlichen Abweichungen müssen in drahtlosen Sensornetzwerken zusätzlich Verzögerungen in der Nachrichtenübertragung berücksichtigt werden, die sich auf die Synchronisation auswirken.

Es können verschiedene Arten von Verzögerungen auftreten:

- **Send Delay:** Verzögerung beim Senden eines Pakets.
- **Access Delay:** Wartezeit auf den Medienzugriff.
- **Transmission Delay:** Zeit, die benötigt wird, um das Paket zu übertragen.
- **Propagation Delay:** Laufzeit des Signals durch das Medium.
- **Reception Delay:** Empfangszeit des Pakets.
- **Receive Delay:** Verzögerung bei der Verarbeitung der empfangenen Nachricht.

Daher müssen mehrere Nachrichten zur Bestimmung der Uhrenunterschiede übertragen werden [71, 56].

Zusätzlich müssen bei der Synchronisation in drahtlosen Sensornetzwerken Aspekte, wie Energieeffizienz, Skalierbarkeit, Genauigkeit, Robustheit und Kosten mit einbezogen werden. Daher werden im nächsten Abschnitt grundlegende Synchronisationsansätze und Protokolle vorgestellt, die diese Faktoren berücksichtigen [56].

2.3.2 Synchronisationsansätze

Zur Synchronisation in drahtlosen Sensornetzwerken stehen drei grundlegende Prinzipien zur Verfügung: Die Zwei-Wege-Synchronisation oder auch Sender-Empfänger-Synchronisation, die Ein-Weg-Synchronisation und die Empfänger-Empfänger-Synchronisation [71].

Bei der Zwei-Wege-Synchronisation werden Zeitstempel von und zu einem Empfänger ausgetauscht. Das bedeutet, dass sowohl der Sender als auch der Empfänger Zeitstempel beim Versenden t_0 (beim Sender), t_2 (beim Empfänger) und Empfangen t_1 (beim Sender), t_3 (beim Empfänger) aufnehmen. Häufig werden für N Runden Zeitstempel ausgetauscht und der Empfänger passt nach den Runden seine Zeit an. Der Ansatz wurde in den Protokollen wie das Timing-Sync Protocol for Sensor Networks (TPSN), Tiny-Sync and Mini-Sync und Lightweight Tree-based Synchronization (LTS) eingesetzt [71, 56].

Die Ein-Weg-Synchronisation basiert darauf, dass ein Master-Knoten seine Zeitinformationen per Broadcast an N Knoten verteilt. Die Knoten nehmen die Empfangszeit der Zeitinformationen auf und passen ihre Zeit entsprechend an. Ein Protokoll, das diesen Ansatz implementiert, ist das Flooding Time Synchronization Protocol (FTSP) [71].

Die dritte Variante ist die Empfänger-Empfänger-Synchronisation bei der ein Master-Knoten seine Zeitinformation einmalig versendet und die Empfänger-Knoten unter sich die Empfangszeiten der Nachrichten austauschen. Dieses Prinzip wurde im Reference Broadcast Synchronization (RBS) Protokoll umgesetzt [71, 56].

2.3.3 Timing-Sync Protocol for Sensor Networks (TPSN)

Im Folgenden wird das Timing-Sync Protocol for Sensor Networks (TPSN) näher erläutert. Das TPSN-Protokoll ist ein Protokoll, welches auf dem Zwei-Wege-Ansatz basiert. Das Protokoll besteht aus zwei grundlegenden Phasen: der Level Discovery Phase und der Synchronization Phase [56, 22].

In der Level Discovery Phase wird eine Baumstruktur im Netzwerk aufgebaut, indem ein Wurzel-Knoten ein Level-Discovery-Paket via Broadcast verteilt. In diesem Paket ist eine Identifikation und das Level des Senders enthalten. Dieses wird an alle Nachbarn verteilt und diese weisen sich selber ein höheres Level zu und versenden wiederum eigene Level-Discovery-Pakete an ihre Nachbarn. Dieser Prozess wird fortgesetzt, bis alle Knoten ein Level zugewiesen bekommen haben [56, 22, 51].

In der Synchronisationsphase wird eine Synchronisation von oben nach unten durch den Baum mit jeweils einem Eltern- und einem Kind-Knoten durchgeführt. Diese Phase nutzt das grundlegende Prinzip der Zwei-Wege-Synchronisation. Sie wird durch eine Time-Sync-Nachricht des Wurzel-Knotens initiiert [56, 22, 51].

Ein Kind-Knoten sendet eine Synchronisationsimpuls-Nachricht an den Eltern- oder Wurzel-Knoten und speichert dabei den Zeitstempel T_1 . Der Eltern-Knoten speichert beim Empfang dieser Nachricht den Zeitstempel T_2 und antwortet nach einiger Zeit mit einem Acknowledgement (ACK)-Paket. Das ACK-Paket enthält den Empfangszeitstempel T_2 der Synchronisationsimpuls-Nachricht sowie den Sendezeitstempel der ACK-Nachricht T_3 . Der Kind-Knoten empfängt diese Paket und speichert einen weiteren Empfangszeitstempel T_4 . Mit diesen vier Zeitstempeln kann der Kind-Knoten anschließend den Clock Drift sowie die Übertragungsverzögerung mit den folgenden Formeln berechnen [56, 22, 51]:

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}, \quad d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2} \quad [56, 22].$$

Mit diesen zwei Informationen kann der Kind-Knoten, auch Sender-Knoten genannt, seine lokale Uhr anpassen. Daher handelt es sich bei diesem Verfahren um einen Zwei-Wege- oder Sender-Empfänger-Ansatz [22].

2.4 Grundlagen zur Signalverarbeitung

Die Bestimmung der Pulswellengeschwindigkeit mit einem drahtlosen PPG-Sensorsystem erfordert nicht nur eine zuverlässige Kommunikation und Synchronisation, sondern auch eine effektive digitale Signalverarbeitung. In diesem Abschnitt werden daher die wesentlichen Begriffe und Verfahren der Signalverarbeitung vorgestellt.

Eine Signalverarbeitungskette zur Bestimmung der Pulswellengeschwindigkeit besteht aus drei Teilen: der Vorverarbeitung, der Merkmalsextraktion und einer Auswertung der Merkmale [42, 16].

In der Vorverarbeitung werden die Rohsignale von Störfaktoren bereinigt. Dazu zählen die in Kapitel 2.1.4 beschriebenen Störfaktoren. Diese Störungen können sowohl hoch- als auch niederfrequente Anteile enthalten. Zur Reduzierung von Störungen werden FIR- oder IIR-Filter sowie fortgeschrittene Verfahren wie die Wavelet-Transformation eingesetzt [42, 14, 20].

Vorherige Arbeiten haben gezeigt, dass ein Chebyshev-II-Bandpassfilter vierter Ordnung eine gute Filterleistung erzielt. Alternativ werden Butterworth-Filter erster bis vierter Ordnung vorgeschlagen [37, 46, 48]. Die Wahl des Filtertyps und der Grenzfrequenzen hat einen erheblichen Einfluss auf die Signalqualität. Insbesondere Signale, bei denen die morphologischen Eigenschaften erhalten bleiben müssen, sind von diesen Parametern stark abhängig. Dies betrifft vor allem die geometrischen Eigenschaften des Originalsignals sowie dessen erste, zweite und dritte Ableitung, ebenso wie zeitliche Merkmale [46]. Ein häufig eingesetzter Bandpassbereich liegt zwischen 0,5 Hz und 20 Hz [10, 48].

Die Extraktion von Merkmalen ist entscheidend für die Bestimmung der Pulswellengeschwindigkeit. Wichtige Merkmale sind Maxima (systolic peak), Minima (onset) sowie charakteristische Punkte wie die diastolische Kerbe (diastolic notch) und das diastolische Maximum (diastolic peak) [59, 42, 48, 14, 16].

Zusätzlich werden die erste, zweite und dritte Ableitung untersucht, um weitere Parameter zu extrahieren [42, 16]. Hierzu zählen das Maximum der ersten Ableitung sowie die Maxima und Minima der zweiten Ableitung (oft als a-, b-, c-, d- und e-Punkte bezeichnet) und der dritten Ableitung (p1 und p2). Abbildung 2.5 zeigt ein Beispiel für die detektierten Merkmale in den ursprünglichen Pulswellen sowie in deren Ableitungen [59, 42, 48, 14, 16].

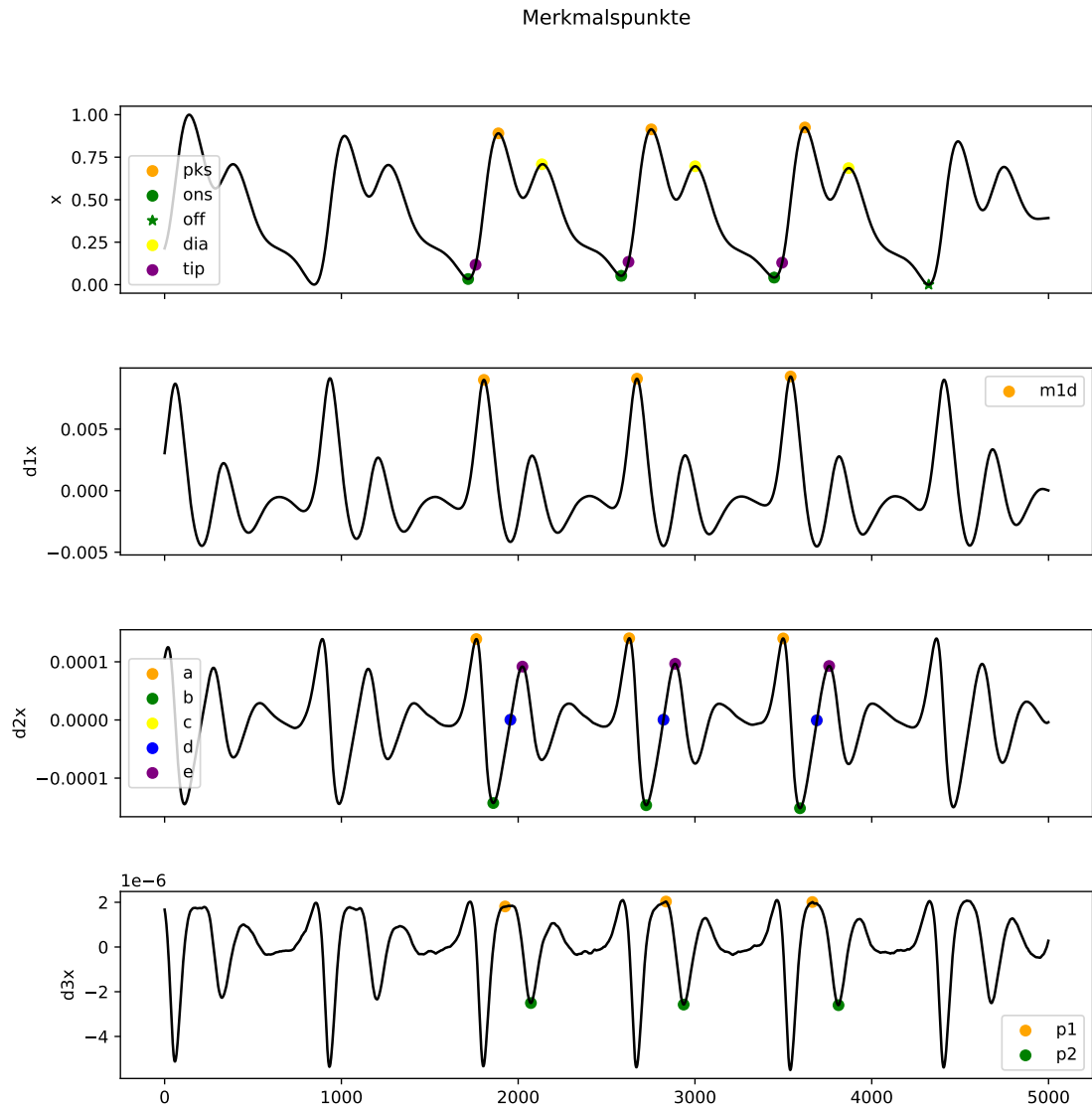


Abbildung 2.5: Exemplarische Darstellung perfekt detektierter Pulswellenmerkmale.

Um diese Merkmale zuverlässig zu identifizieren, wird häufig eine Peak-Detektion eingesetzt. Da Pulswellen durch die Herzfrequenz variieren, eignet sich eine adaptive Fenstergröße für die Peak-Detektion [14, 48]. Innerhalb eines definierten Intervalls, dem Inter-Beat-Intervall (IBI), lassen sich die relevanten Merkmale bestimmen [42, 16, 14].

Neben klassischen Merkmalen stellt die Tangenten-Intersektions-Methode einen bewährten Ansatz zur Bestimmung des Abstands zwischen zwei Pulswellen dar [8, 13]. Hierbei werden Tangenten an charakteristische Punkte der Welle angelegt, typischerweise am Minimum (engl. onset) und am Punkt der maximalen Steigung (engl. max. 1st derivative, m1d). Der Punkt, an dem sich die Tangenten schneiden, wird als Tangenten-Intersektions-Punkt (engl. tangent-intersection-point, tip) bezeichnet. Er gilt als zuverlässigerer Merkmalspunkt, da er weniger von Wellenreflexionen beeinflusst wird. Daher wird er häufig anstelle des Minimums oder der maximalen Steigung verwendet [5].

Zusätzlich zur Punkt-zu-Punkt-Analyse erlaubt diese Methode eine segmentierte Untersuchung bestimmter Pulswellenabschnitte [5]. In diesem Zusammenhang werden Verfahren wie die Kreuzkorrelation (Cross-Correlation) oder die Least Squares Difference Methode angewendet, um Ähnlichkeiten zwischen Signalabschnitten quantitativ zu bewerten [21].

Zusätzlich zur Analyse im Zeitbereich kann das Signal auch im Frequenzbereich untersucht werden. Dafür bietet sich unter anderem die Fast-Fourier-Transformation an. Im Frequenzspektrum lassen sich die relevanten Frequenzanteile identifizieren. Zudem können Abweichungen im Frequenzbereich auf untypische Pulswellen hinweisen. [14, 20, 42].

3 Systementwicklung

In diesem Kapitel wird aufbauend auf den Erkenntnissen der vorherigen Abschnitte die Entwicklung des Sensorsystems beschrieben. Dazu werden einleitend die Anforderung an das Messsystem definiert und darauf ein grundlegendes Konzept zu dem Sensorsystem vorgestellt. Anschließend wird die Auswahl der Hardware-Komponenten, die Softwareentwicklung sowie die Signalverarbeitung im Detail erläutert.

3.1 Anforderungen an das Sensorsystem

Das in dieser Arbeit entwickelte Sensorsystem soll die Pulswellengeschwindigkeit im Bereich des Aortenbogens bestimmen. Wie im vorherigen Kapitel beschrieben, ist dieser Bereich von besonderem Interesse, da die Aorta am stärksten von arterieller Versteifung betroffen ist. Die Pulswellengeschwindigkeit soll für diesen Bereich indirekt mit den abgemessenen Pulswellen der linken und rechten Armarterie (Arteria radialis und Arteria ulnaris) bestimmt werden, um ein nutzerfreundliches und portables Sensorsystem zu gewährleisten.

Die Berechnung der Pulswellengeschwindigkeit basiert auf den zuvor beschriebenen Merkmalen der Pulswelle, die aus den Signalen extrahiert werden. Mit diesen Merkmalen wird zunächst die Pulswellenlaufzeit (engl. Pulse Transit Time, PTT) zwischen der vom linken und rechten Arm erfassten Pulswellen bestimmt. Um die Pulswellengeschwindigkeit berechnen zu können muss zudem der Abstand zwischen den zwei Abzweigungen der Unterschlüsselbeinarterien (Arteria subclavia) abgeschätzt werden. Die Berechnung der Pulswellengeschwindigkeit erfolgt schließlich aus der PTT und dem geschätzten Abstand zwischen den beiden Unterschlüsselbeinarterien. Die Abbildung 3.1 veranschaulicht das Messprinzip des Sensorsystems. Sie zeigt die Platzierung der Sensorknoten an den Armarterien (Arteria radialis und Arteria ulnaris) sowie den abzuschätzenden Abstand zwischen den beiden Abzweigungen der Arteria subclavia.

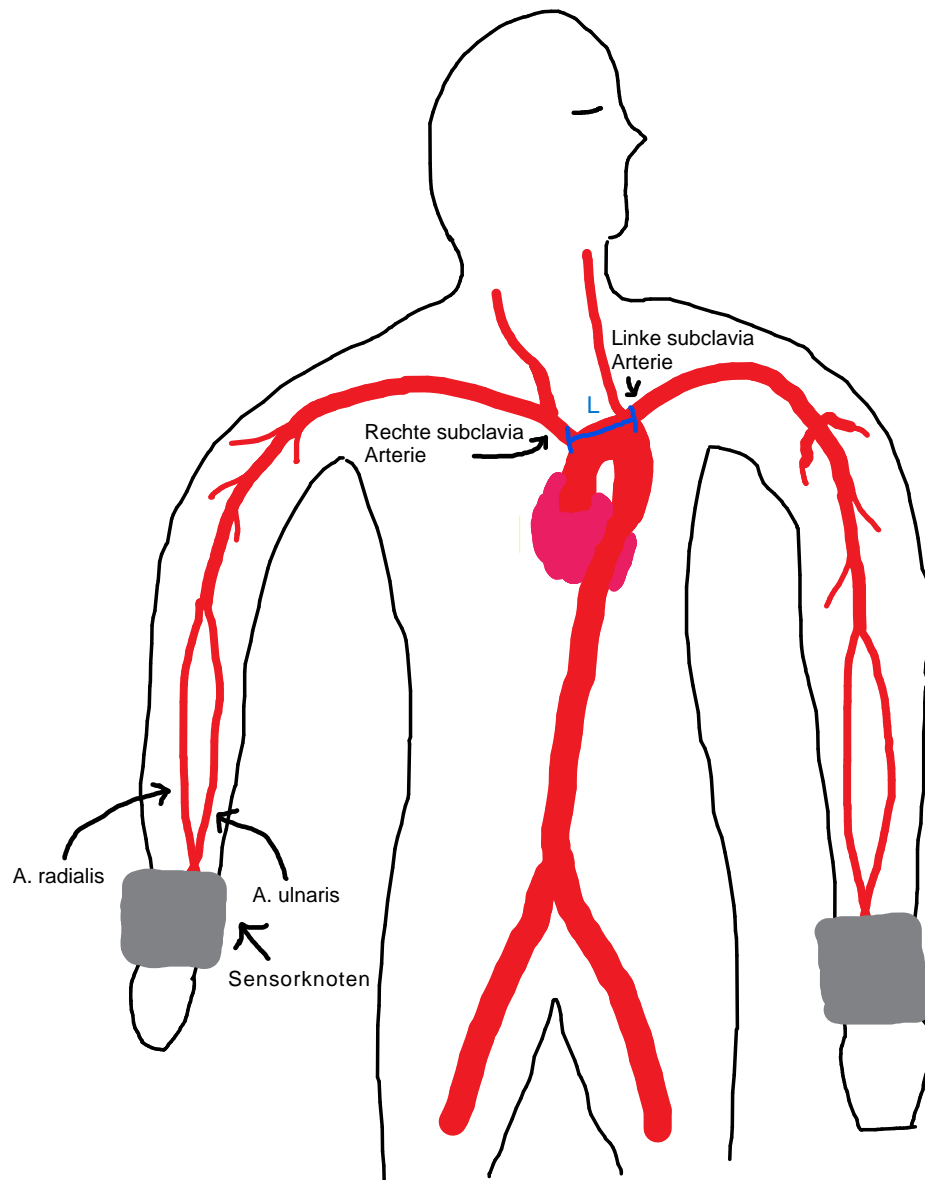


Abbildung 3.1: Die Abbildung zeigt eine Skizze des Oberkörpers, der in diesem Bereich liegenden Arterien und die relevanten Mess- und Abstandspunkte.

Für die präzise Erfassung der Pulswellen am linken und rechten Arm ist die Auswahl einer geeigneten Abtastrate entscheidend. Dazu haben vorherige Arbeiten gezeigt, dass die dominierenden Frequenzanteile einer Pulswelle bis zu 90 % im Bereich von 0,5 bis 10 Hz liegen und ein weiterer signifikanter Teil im Bereich von 0 bis 20 Hz [14]. Um das Nyquist-Shannon-Abtasttheorem einzuhalten, ist daher eine Abtastrate von mehr als 40 Hz erforderlich. Weitere Quellen empfehlen sogar höhere Abtastraten von 50 bis 100 Hz, da diese zu besseren Signalen führen, die mehr biomedizinische Merkmale enthalten [29, 10].

Damit an den beiden Armen die Pulswelle erfasst werden kann, müssen zwei Sensorknoten entwickelt werden. Hierfür muss geeignete Hardware ausgewählt werden, insbesondere ein Entwicklungsboard, das ein passendes drahtloses Kommunikationsprotokoll unterstützt. Dieses Kommunikationsprotokoll muss effizient genug sein, das heißt, es sollte sowohl energiesparend als auch möglichst latenzarm sein.

Außerdem muss ein PPG-Sensormodul ausgewählt werden, welches ebenfalls Kriterien wie Energieeffizienz und Kompaktheit erfüllt. Weitere Sensoren, beispielsweise zur Erfassung von Bewegungen, werden in dieser Arbeit nicht berücksichtigt, da der Fokus auf der Verwendung von PPG-Sensormodulen liegt.

Ergänzend sollen geeignete Softwarelösungen und Synchronisationsalgorithmen entwickelt und implementiert werden, die Verzögerungen minimieren und eine korrekte Zuordnung der Daten sicherstellen, um eine möglichst genaue Berechnung der Pulswellengeschwindigkeit zu gewährleisten.

Dabei ist insbesondere wichtig, dass sowohl die Hardware- als auch die Softwarelösung die Sensordaten schnell genug erfassen und mögliche Verzögerungen auf ein vernachlässigbares Maß begrenzen.

Die folgende Tabelle 3.1 stellt mögliche Pulswellengeschwindigkeiten und Abstände zwischen den Abzweigungen der Arterien in Beziehung zueinander. Die Einträge in den Zellen entsprechen den erwarteten Pulswellenlaufzeiten. Für eine gesunde Person mit einer potentiellen Pulswellenschwindigkeit von etwa 4 m/s, würden die Pulswellen beispielsweise etwa 5 bis 10 ms auseinander liegen [20, 6, 69].

Aufgrund dieser Anforderung muss das Sensorsystem mindestens eine zeitliche Auflösung im Millisekundenbereich erreichen. Die zu erfüllenden Erwartungsbereiche für die Pulswellengeschwindigkeiten für unterschiedliche Altersgruppen sind in Tabelle 3.2 aufgeführt.

Tabelle 3.1: Zeitabstände für verschiedene Geschwindigkeiten und Distanzen [20, 6, 69].

Geschwindigkeit (m/s)	0,02 m	0,03 m	0,04 m
4	5 ms	7,5 ms	10 ms
6	3,3 ms	5 ms	6,7 ms
10	2 ms	3 ms	4 ms
15	1,3 ms	2 ms	2,7 ms

Tabelle 3.2: Erwartungsbereiche für Pulswellengeschwindigkeiten für die entsprechende Altersgruppe [15].

Altersgruppe	Pulswellengeschwindigkeit
10-30	3,5 - 8 m/s
30-60	4 - 11 m/s
60-90	5 - 13 m/s

Ein Sensorknoten muss zudem batteriebetrieben sein und ist zunächst für Forschungs- und Entwicklungszwecke vorgesehen. Daher ist die Anforderung an die Betriebsdauer nicht sehr hoch und soll im Batteriebetrieb mindestens 1,5 Stunden gewährleisten. Diese Anforderung stellt sicher, dass Messungen über einen ausreichend langen Zeitraum durchgeführt werden können. Eine Optimierung der Batteriebetriebsdauer der Sensorknoten ist nicht Teil dieser Arbeit. Der Fokus liegt vielmehr auf dem Messprinzip und den verschiedenen Herausforderungen in der Entwicklung.

Um zudem ein kompaktes und tragbares Design für die Sensorknoten zu erreichen, welches an einer Smartwatch oder einem Fitnesstracker orientiert ist, soll entsprechend platzsparende Hardware eingesetzt werden, die in ein speziell entwickeltes Gehäuse integriert werden soll.

3.2 Aufbau des Sensorsystems

In diesem Abschnitt wird anhand der vorher definierten Anforderungen ein Systemkonzept vorgestellt. Um die Pulswellen an unterschiedlichen Arterien, wie an den zwei Armen zu erfassen, sind mindestens zwei Sensorknoten erforderlich. Aus diesem Grund wurde ein System mit drei Knoten entwickelt: Zwei Sensorknoten, die an den entsprechenden Hautstellen angebracht werden und dort mithilfe von PPG-Sensoren Pulswellen erfassen, sowie ein Empfangsknoten, der die Sensordaten drahtlos empfängt und an einen Auswertungsknoten weiterleitet. Der Empfangsknoten dient in diesem Netzwerk als Synchronisations- und Netzwerkverwaltungsinstanz und soll die Sensordaten sammeln und weiterleiten. Der Empfangsknoten soll zudem als Adapter dienen und eine Schnittstelle für den Auswertungscomputer bereitstellen. Auf einem Auswertungscomputer wird anschließend eine Datenverarbeitung durchgeführt, um die Knoten im drahtlosen Netzwerk von rechenintensiven Aufgaben zu entlasten und den Energieverbrauch zu reduzieren. Die Abbildung 3.2 zeigt den schematischen Aufbau des entwickelten, drahtlosen PPG-Sensorsystems.

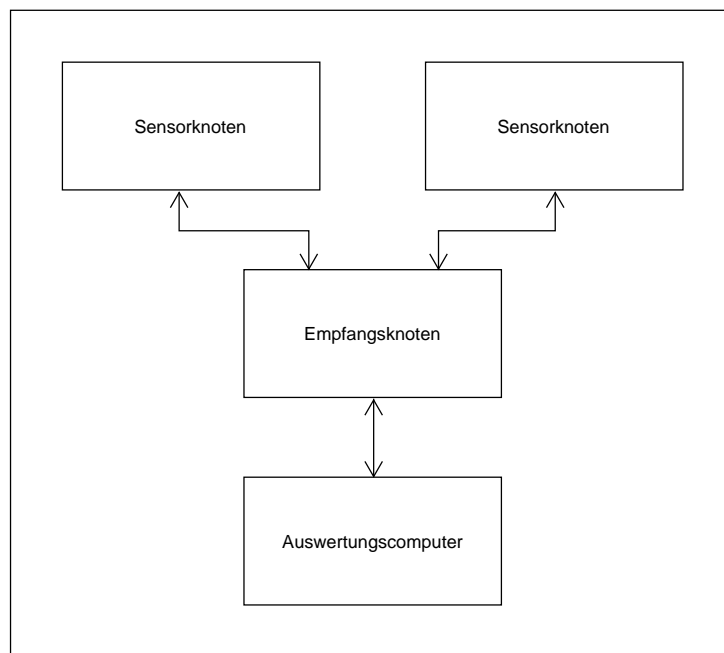


Abbildung 3.2: Der schematische Aufbau des Sensorsystems als Blockdiagramm.

3.3 Auswahl der Hardware

Die im vorherigen Kapitel beschriebene Übersicht wurde verfeinert und es wurden für die grundlegenden Komponenten in diesem Sensorsystem, Schnittstellen und konkrete Hardware-Komponenten bestimmt und ausgewählt, die die Anforderungen des Sensorsystem erfüllen. In der Entwicklung des Sensorsystems war vor allem die Auswahl von geeigneter Hardware-Komponenten für die Sensorknoten elementar, da diese drahtlos und mit einer Batterie betrieben werden sollen.

Eine essenzielle Komponente aller Knoten in diesem drahtlosen Sensorsystem ist das eingebettete Entwicklungsboard. Dieses muss eine kompakte Bauweise aufweisen, eine schnelle Datenverarbeitung ermöglichen und sowohl Kommunikationsschnittstellen für das Sensormodul als auch drahtlose Übertragungsstandards unterstützen. Weitere Auswahlkriterien waren die Verfügbarkeit und der Preis des Entwicklungsboards. Aus diesen Gründen fiel die Wahl auf das Entwicklungsboard XIAO ESP32-C6 der Firma Seeed Studio. Das Entwicklungsboard basiert auf dem leistungsfähigen System-on-a-Chip (SoC) ESP32-C6 der Firma Espressif.

Der ESP32-C6 ist ein Ultra-low-power SoC mit zwei 32-bit RISC-V Prozessoren, einem high-performance (HP) Prozessor mit 160 MHz und einem low-power (LP) Prozessor mit 20 Mhz Taktrate [17, 55]. Diese Taktraten sind in Bezug auf die erforderlichen 40 Hz Abtastrate mehr als ausreichend und daher hervorragend für das Sensorsystem geeignet. Bei dem ESP32-C6 handelt es sich jedoch nur um ein single-core System, da der low-power Prozessor nur während low-power Anwendungen zum Einsatz kommt. Ansonsten laufen die Instruktionen ausschließlich auf dem high-performance Prozessor. Dementsprechend unterstützt der ESP32-C6 keine echte Parallelität, sondern nur pseudo-parallele Ausführungen von Tasks mithilfe des kompakten und effizienten Echtzeitbetriebssystems FreeRTOS [17, 19].

Neben den schnellen Prozessoren besitzt der ESP32-C6 ein umfangreiches Radiofunk-System für die Unterstützung verschiedener Funkprotokolle, wie unter anderem Wi-Fi 6 im 2,4 GHz Band basierend auf den IEEE 802.11b/g/n/ax Standards, Bluetooth Low Energy 5.3 basierend auf dem IEEE 802.15.1 Standard, sowie Zigbee 3.0 und Thread 1.3 beide jeweils auf dem IEEE 802.15.4 Standard basierend [17, 55]. Durch die Unterstützung dieser modernen Funkprotokolle kann ein drahtloses und energieeffizientes Sensorsystem entwickelt werden. Zudem kann mithilfe dieses Radiofunk-Systems in Zukunft das Funkprotokoll für bestimmte Umgebungen und Funknetzwerke schnell angepasst werden,

ohne ein neues Entwicklungsboard auswählen zu müssen und einzusetzen. Zur Übertragung bietet das XIAO ESP32-C6 Entwicklungsboard wahlweise eine on-board Keramik Antenne mit einer Reichweite von bis zu 80 Metern oder eine Verbindungsschnittstelle für eine externe ULF-Antenne [55].

Weitere wichtige Merkmale des XIAO ESP32-C6 Entwicklungsboards sind die zur Verfügung stehenden Peripherie-Schnittstellen, die kompakte Größe und die Möglichkeit das Board direkt mit einer Batterie zu betreiben. Als Peripherie-Schnittstellen stehen dem Entwicklungsboard eine SPI-, eine UART- und eine I²C-Schnittstellen zur Verfügung, wobei es für I²C und UART jeweils noch eine low-power Schnittstelle gibt. Des weiteren bietet das Entwicklungsboard elf GPIO-Pins und sieben Analoge Pins. Ein weiteres herausragendes Merkmal ist die kompakte Größe des Entwicklungsboards von nur $21 \times 17,8$ mm, was es ebenfalls ideal für den Einsatz in diesem Sensorsystem macht [55].

Die Möglichkeit das Entwicklungsboard mit einer 3,3 V Batterie durch die entsprechenden Batterie-Pins zu versorgen, vereinfacht das Design der Sensorknoten und reduziert die Kosten des Systems, da kein externer Laderegler und Spannungsregler eingesetzt werden muss. Das Entwicklungsboard kann unabhängig von dem Batteriebetrieb jederzeit über eine USB Type-C Schnittstelle mit Strom versorgt werden und dabei gleichzeitig die Batterie mit 3,3 V geladen werden. Über die USB Type-C Schnittstelle wird zudem das Board programmiert oder direkt mit einem externen Computer ausgelesen [55].

Zur Realisierung eines Sensorknotens mussten neben dem Entwicklungsboard ein geeignetes PPG-Sensormodul und eine Batterie ausgewählt werden. Als PPG-Sensormodul wurde das GY-MAX30102-Sensormodul ausgewählt, auf diesem ist ein MAX30102 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health Sensormodul von Analog Devices (chemals Maxim Integrated Products) verbaut [3].

Das GY-MAX30102-Sensormodul zeichnet sich durch seinen günstigen Preis, seine hohe Verfügbarkeit und seine kompakte Größe von $13 \times 13 \times 3$ mm aus. Zudem existieren zahlreiche umgesetzte Projekte mit dem Sensormodul und frei verfügbare Bibliotheken, die die Implementierung in das Sensorsystem erleichtern. Das GY-MAX30102-Sensormodul vereinfacht die Inbetriebnahme und den Zugriff auf das darunterliegenden MAX30102-Sensormodul, in dem es fünf Pins bereitstellt. Einen VIN- und GND-Pin für die 3,3 V Spannungsversorgung, eine SCL- und SDA-Pin für die I²C-Schnittstelle und einen INT-Pin, um Interrupts von dem Sensor zu signalisieren. Mit einer Spannungsversorgung von

3,3 V und einer I²C-Schnittstelle lässt sich dieses Sensormodul sehr gut mit dem XIAO ESP32-C6 Entwicklungsboard verbinden.

Das MAX30102-Sensormodul integriert verschiedene Komponenten. Dazu gehören eine Infrarot-LED mit einem Wellenlängenbereich von 870 bis 900 nm sowie eine Rotlicht-LED mit einem Bereich von 650 bis 670 nm. Ein Photodetektor erfasst das reflektierte Licht dieser LEDs mit einem spektralen Erkennungsbereich von 600 bis 900 nm [3].

Außerdem verfügt das Modul über einen Filter und eine Umgebungslichtunterdrückung (engl. ambient light cancellation, ALC). Ein 18-Bit continuous-time oversampling sigma-delta ADC mit einer Sampling-Rate von 10,24 MHz digitalisiert die analogen Sensorsignale. Zur Datenübertragung stehen ein Datenregister und eine I²C-Schnittstelle mit einer Taktfrequenz von 400 kHz zur Verfügung [3].

Das Datenregister des MAX30102-Sensormoduls wird über die I²C-Schnittstelle gelesen oder beschrieben. Dabei können sowohl zahlreiche Sensoreinstellungen vorgenommen werden oder die internen Infrarot- und Rotlichtsensor Samples aus dem internen Daten-FIFO gelesen werden. Als Register stehen unter anderem Interrupt-Register, FIFO-Register und Konfigurations-Register zur Verfügung [3]. Über die Interrupt-Register kann der Interrupt-Status von anliegenden Interrupts, wie von FIFO Almost Full oder New FIFO Data Ready abgefragt werden und über Interrupt-Enable-Register aktiviert oder deaktiviert werden [3].

Von den FIFO Registern ist das FIFO Write Pointer, FIFO Read Pointer und das FIFO Data Register interessant, das erste zeigt an, wo als nächstes ein Sample in die FIFO geschrieben wird, das zweite von wo ein Sample aus der FIFO über I²C gelesen wird und das dritte Register zeigt auf den nächsten Sample der aus der FIFO gelesen werden soll. Ein Sample kann entweder 3 Byte groß sein, wenn nur die Infrarot LED aktiv ist oder 6 Byte groß sein, wenn sowohl die Infrarot- als auch Rotlicht-LED aktiv sind. Die FIFO besteht aus einer 32 Sample großen Speicherbank, also insgesamt 32 x 6 Bytes 192 Bytes an Sensordaten [3].

Neben den Interrupt- und FIFO-Registern sind auch die verschiedenen Konfigurationsregister von Interesse. Über diese können Einstellungen wie die Mittelung der Abtastwerte, der Betriebsmodus des Sensors, der ADC-Bereich, die Abtastrate des ADCs, die LED-Pulsweite, der LED-Strom sowie der Multi-LED-Modus eingestellt werden [3].

Die Sensorsamples können über 1, 2, 4, 8, 16, oder 32 Samples gemittelt werden. Es kann ein Shutdown-Modus eingestellt werden mit dem anstatt einer maximalen Stromversor-

gung von 1200 μA eine Stromversorgung von nur 10 μA erreicht wird. Das Modul kann außerdem während des Betriebs durch einen Reset-Modus auf einen Vorbetriebszustand zurückgesetzt werden, indem alle Register zurückgesetzt sind und Standardwerte beinhalten. Außerdem können der SpO₂- oder HR-Modus ausgewählt werden, bei dem entweder nur die rote LED oder beide LEDs aktiv sind. Der ADC-Auflösungsbereich kann von 2048 auf bis zu 16384 nA und die ADC-Abtastrate von 50 bis 3200 Hz eingestellt werden. Die LED-Pulsweite kann von 69 μs bis 411 μs eingestellt werden, um die Genauigkeit und den Energieverbrauch abhängig vom Use-Case zu steuern. Die Stromversorgung der LEDs kann ebenfalls gesteuert und von 0 bis 50 mA eingestellt werden [3].

Da das GY-MAX30102-Sensormodul eine Abtastrate von mindestens 50 Hz unterstützte ist dieses Sensormodul für die Detektierung der Pulswelle geeignet. Zudem ermöglicht der geringe Stromverbrauch von maximal 1200 μA für das Sensormodul sowie 50 mA für die LEDs und die Option eines Shutdown-Modus einen energieeffizienten Betrieb. Die weiteren Merkmale, wie die unterstützte I²C-Schnittstelle, die vielen Einstellungsmöglichkeiten und die Größe des Sensormoduls, sprechen ebenfalls für den Einsatz des GY-MAX30102-Sensormoduls in dem zu entwickelnden Sensorsystem. Die Abbildung 3.3 zeigt nochmals das Sensorsystem mit seinen eben definierten Hardware-Komponenten und den möglichen Schnittstellen.

Für die Sensorknoten muss zudem eine geeignete Batterie ausgewählt werden, die unter maximalen Bedingungen einen Betrieb von mindestens einer Stunde pro Sensorknoten ermöglicht. Basierend auf dem Datenblatt des ESP32-C6 wurde angenommen, dass der ESP32-C6 im vollen Wi-Fi-Sende-Modus einen maximalen Stromverbrauch von 354 mA erreicht [17]. Für das GY-MAX30102-Sensormodul wurde ein maximaler Stromverbrauch von 1200 μA für das Modul selbst sowie zweimal 50 mA für die LEDs angenommen [3]. Daraus ergibt sich ein theoretischer Gesamtstromverbrauch von 455,2 mA unter voller Auslastung. Somit musste eine Batterie mit einer Kapazität von mindestens 455,2 mAh ausgewählt werden. Da für das Entwicklungsboard eine Spannung von 3,7 V empfohlen wird, wurde ein 3,7 V Lithium-Ionen-Akku mit einer Kapazität von 700 mAh ausgewählt, welcher unter den theoretischen maximalen Bedingungen eine Betriebsdauer von

$$\text{Betriebsdauer} = \frac{700 \text{ mAh}}{455,2 \text{ mA}} = 1,53 \text{ h}$$

ermöglicht [3].

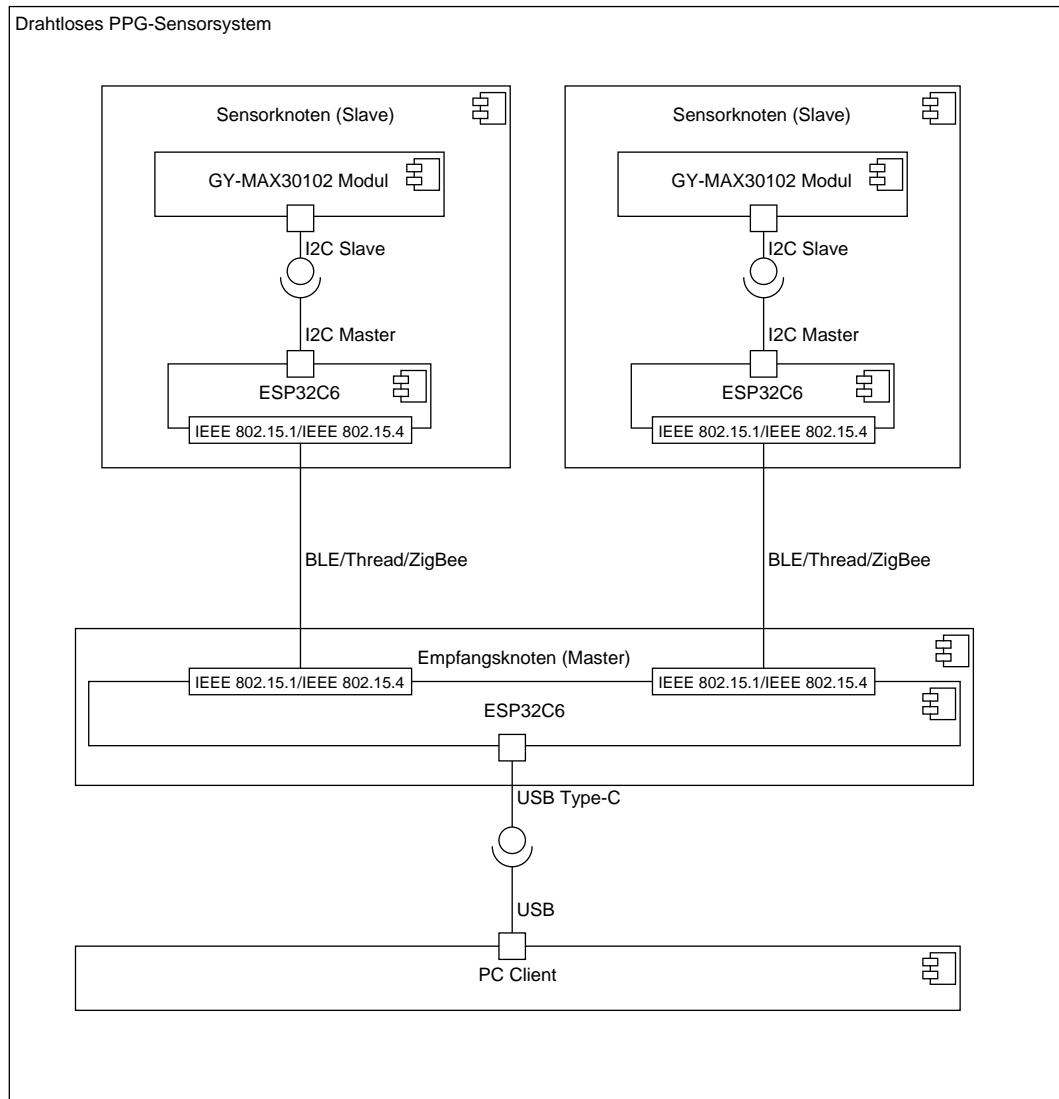


Abbildung 3.3: Die Abbildung zeigt den schematischen Aufbau des Sensorsystems mit den physischen Komponenten und den zur Verfügung stehenden Schnittstellen.

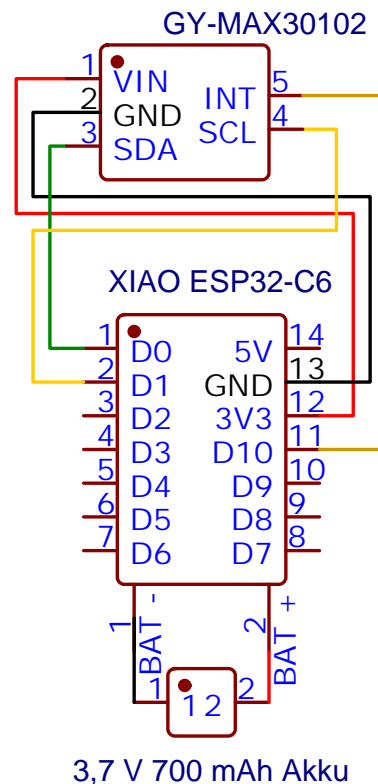


Abbildung 3.4: Die Abbildung zeigt die Schaltung mit den Pin-Belegungen für einen Sensorknoten.

Alle drei Komponenten eines Sensorknotens wurden entsprechend der in Abbildung 3.4 dargestellten Schaltung zu einem Sensorknoten verbunden. Anschließend wurden die Schaltungen mithilfe eines Multimeters auf Kurzschlüsse und fehlerhafte Verbindungen überprüft.

Zum Schutz und zur sicheren Erprobung der Schaltung wurde ein Gehäuse entwickelt, das in Abbildung 3.5 dargestellt ist. Das Design orientiert sich in Größe und Form an einer Smartwatch und gewährleistet eine optimale Unterbringung aller Komponenten. Mithilfe eines 3D-Druckers wurden für jeden Sensorknoten ein Gehäuse gefertigt und in jedem jeweils eine Schaltung verbaut. Die fertigen Sensorknoten-Armbänder sind in Abbildung 3.6 zu sehen.

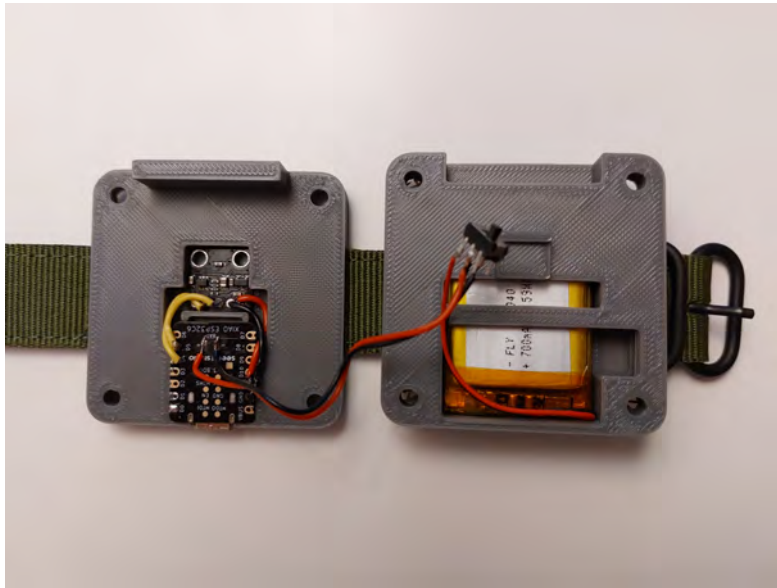


Abbildung 3.5: Die Abbildung zeigt ein Sensorknoten-Gehäuse von Innen.



Abbildung 3.6: Die Abbildung zeigt die zwei entwickelten Sensorknoten-Armbänder.

3.4 Entwicklung der Firmware

In diesem Kapitel wird die Entwicklung der Firmware beschrieben. Es wird unter anderem auf die verwendeten Entwicklungswerkzeuge, Bibliotheken und insbesondere auf den Entwicklungsprozess der Sensorknoten-Firmware eingegangen. Zudem werden die Herausforderungen und Lösungen im Zusammenhang mit dem Softwaredesign, der Kommunikation und der Synchronisation dargestellt.

Die im Kapitel 3.3 definierten Hardwarekomponenten wurden in logische Komponenten aufgeteilt. Entsprechend wurden separate Softwarelösungen für den Sensorknoten und den Empfangsknoten entwickelt. Die Firmware des Sensorknotens umfasst eine Softwarekomponente für das GY-MAX30102-Sensormodul sowie eine Kommunikationskomponente. Die Firmware des Empfangsknotens enthält nur eine Kommunikationskomponente.

Die internen Softwarekomponenten der beiden logischen Komponenten wurden iterativ mit einem bottom-up Ansatz entwickelt. Dabei wurde das GY-MAX30102-Sensormodul von einer niedrigen zu einer höheren Abstraktionsebene entwickelt. Ebenso wurden die Kommunikationskomponenten von der Ebene der Protokollspezifikation hin zu einer höheren Abstraktionsschicht entwickelt, die einen vereinfachten Zugriff auf die zugrunde liegenden Kommunikationsprotokolle ermöglicht. Dieser Entwicklungsansatz folgt dem Prinzip des Separation of Concerns, wodurch nicht nur eine klarere Abstraktion, sondern auch eine bessere Übersichtlichkeit und Wartbarkeit der Firmware gewährleistet wird. Jede Softwareebene wurde voneinander gekapselt, iterativ weiterentwickelt sowie mehrfach angepasst und separat getestet. Im Folgenden wird zuerst die Entwicklung des logischen Sensorknotens beschrieben und anschließend die Software des Empfangsknotens.

3.4.1 Entwicklung der PPG-Softwaremodule

Zur schnellen Entwicklung und einfachen Integration des GY-MAX30102-Sensormoduls in das Sensorsystem wurden die SparkFun MAX301x Particle Sensor-Bibliothek und die Wire-Bibliothek eingesetzt [57]. Konkret kam aus der SparkFun MAX301x Particle Sensor-Bibliothek das MAX30105-Softwaremodul zum Einsatz. In diesem sind einige vordefinierte Register und Funktionen, die für unterschiedliche MAX3010X-Sensormodule kompatibel sind, unter anderem für das verwendete MAX30102-Sensormodul.

Damit das Softwaremodul nahtlos in das System und vor allem in die logischen Komponenten integriert werden kann, wurde ein MAX30102-Wrapper-Modul entwickelt.

Dieses abstrahiert die Schnittstellen des MAX30105-Moduls und erweitert es um zusätzliche Funktionen, wie `getOneSampleFromFIFO` oder `getOneSampleFromFIFOSecure`. Diese Funktionen ermöglichen einen direkten und effizienten Weg zum Auslesen der Sensordaten aus dem FIFO-Speicher des MAX30102-Sensormoduls.

Die `getOneSampleFromFIFOSecure`-Funktion enthält zusätzliche Prüfmechanismen, die sicherstellen, dass die internen FIFO-Pointer unterschiedlich sind und der FIFO-Speicher vollständig geleert wird. Die `getOneSampleFromFIFO` hingegen liest die Daten direkt aus dem FIFO-Speicher heraus und es erfolgt dabei nur eine Überprüfung, ob die Sensorwerte unterschiedlich und ungleich null sind.

Neben den Funktionen zum Auslesen wurde eine `init`-Funktion implementiert, die für die Initialisierung des Sensors verantwortlich ist. In dieser Funktion wird die I²C-Kommunikation mithilfe der Wire-Bibliothek gestartet und es werden die in dem Kapitel 3.3 beschriebenen Einstellungen für die Konfigurationsregister gesetzt. Es werden der LED-Strom, der Mittelungsfaktor, der LED-Modus, die Abtastrate des Sensors, die Pulsweite der LEDs und der ADC-Bereich eingestellt.

Zu Beginn wurden verschiedene Parameter unabhängig vom eigentlichen Sensorsystem getestet, um eine optimale Einstellung des Sensormoduls zu bestimmen, siehe dazu auch Kapitel 4.4. Neben dem separaten Testen war auch das Datenblatt des MAX30102-Sensormoduls bei diesem Entwicklungsschritt entscheidend.

Folgende Konfiguration wurde bestimmt: Als Abtastfrequenz wurden am Anfang 100 Hz eingestellt. Der LED-Modus wurde auf Modus 2 gesetzt, der im Datenblatt auch als SpO₂-Modus bezeichnet wird. In diesem Modus sind sowohl die rote als auch die infrarote LED aktiv. Die Abtastrate von 100 Hz war zunächst ausreichend für die Detektion der Pulswelle und erfüllte die Mindestanforderungen. In der späteren Entwicklung konnte die Abtastfrequenz durch die Funktion `setSampleRate` jederzeit während der Laufzeit beliebig angepasst werden, um das Sensorsystem mit unterschiedlichen Abtastraten zu testen. Der LED-Modus 2 wurde gewählt, um die verschiedenen Messwerte der infrarote und rotlicht LEDs zu vergleichen. Zudem ist das Infrarotspektrum besser geeignet für die genaue Detektierung der Pulswelle und ist nur im LED-Modus 2 verfügbar.

Die Pulsweite wurde entsprechend dem MAX30102-Datenblatts eingestellt, indem eine Pulsweite von maximal 411 µs im LED-Modus 2 und bei einer maximalen Abtastrate von 400 Hz empfohlen wird [3, 57].

Der ADC-Bereich wurde auf das Maximum von 16384 nA eingestellt, um eine möglichst hohe Genauigkeit zu erzielen. Es wurde keine Mittelung der Sensorwerte durchgeführt, indem der Mittelungsfaktor auf eins gesetzt wurde.

Der LED-Strom für beide LEDs wurde auf das Maximum von 50 mA eingestellt, um eine möglichst tiefe Lichtdurchdringung der Haut zu gewährleisten. Eine kleinere Einstellung hat in dem Zusammenhang zu starken Rauschen und nicht erkennbaren Pulswellen geführt [57]. Jegliche Einstellungen wurde in dem Kapitel 4.4 genauer untersucht und ausgewertet.

Zur weiteren Abstraktion des entwickelten MAX30102-Softwaremoduls wurde eine weitere Ebene PPGSensor entwickelt. Diese soll eine noch weiter vereinfachte Schnittstelle für das MAX30102-Softwaremodul bieten. Die Hauptfunktion dieser Ebene besteht in der Konfiguration des Betriebsmodus (Polling- oder Interrupt-Modus) über die setup-Funktion. Intern sind ein Interrupt-Service-Routine und eine Polling-Funktion implementiert. Die gesamte Struktur der PPGSensor-Softwaremodule ist in folgender Abbildung 3.7 zu sehen.

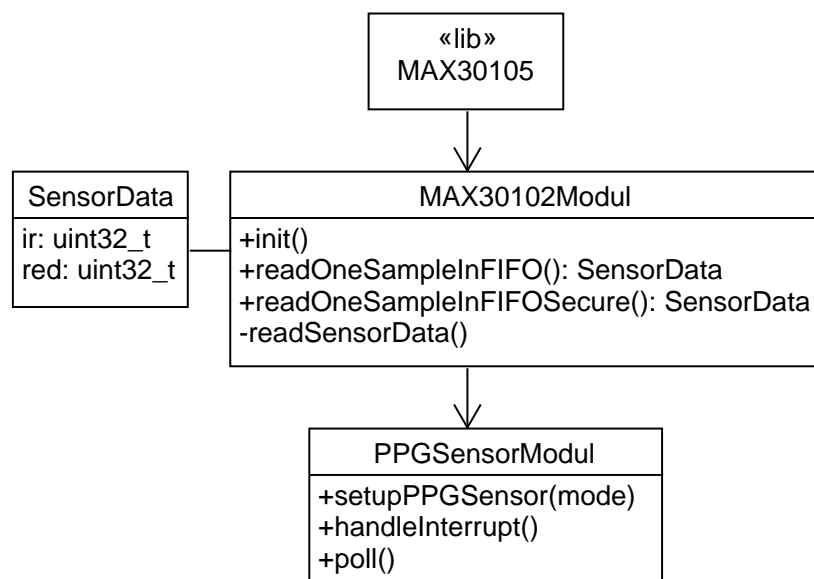


Abbildung 3.7: Die Abbildung zeigt die entwickelten Software Module MAX30102 und PPGSensor.

Die Programmierung erfolgte in der Arduino IDE mit dem Arduino ESP32 Core 3.1-Framework [4, 18]. Diese Entwicklungswerkzeuge werden von dem Hersteller der Entwicklungsboards SeedStudio auf der Webseite empfohlen [55]. Die Wahl der Arduino IDE erfolgte zudem, da sie eine einfache Bedienung und schnelle Prototypenentwicklung ermöglicht. Die IDE unterstützt dazu auch die einfache Integration von Software-Bibliotheken, wie den zuvor erwähnten Bibliotheken SparkFun MAX301x Particle Sensor- und Wire- Bibliothek.

3.4.2 Entwicklung der Kommunikations-Softwaremodule

Eine weitere zentrale Softwarekomponente aller beteiligten Netzwerknoten ist die Kommunikationskomponente. In der Entwicklung der Kommunikationsinfrastruktur wurde die Master-Slave-Architektur eingesetzt, auch als Controller-Peripheral-Architekturansatz bekannt. Da das entwickelte Sensornetzwerk nicht durch die Anzahl der Sensorknoten überlastet wird, eignet sich der Master-Slave-Ansatz gut für das Sensorsystem [50].

Mit diesem Modell wurden die entwickelten Softwarekomponenten Sensorknoten und Empfangsknoten als Slave und Master klassifiziert. Das bedeutet, dass der Empfangsknoten als Master die Verbindung zu den Sensorknoten (Slaves) initiiert und die Verbindungsinformationen intern speichert. So kann ein Empfangsknoten mit mehreren Sensorknoten verbunden werden.

Ein wichtiger Schritt in der Entwicklung bestand darin, die Master- und Slave-Komponenten innerhalb des verwendeten Kommunikationsprotokolls zu identifizieren. Das eingesetzte Protokoll ist Bluetooth Low Energy, das bereits in früheren Arbeiten im Zusammenhang mit drahtlosen Sensorsystemen zur Bestimmung der Pulswellengeschwindigkeit eingesetzt wurde [67, 8].

Für die Kommunikationskomponenten, die intern Bluetooth Low Energy verwenden, wurden zwei Softwaremodule entwickelt, die Protokolldetails abstrahieren und nur wesentliche Funktionen wie das Versenden und Empfangen bereitstellen. Dadurch wird das Facade-Softwarepattern umgesetzt. Es wurden eine BLEServerFacade und eine BLEClientFacade entwickelt, die im Folgenden beschrieben werden.

BLEServerFacade-Modul

Das BLEServerFacade-Modul erfüllt die grundlegende Funktionalität eines Bluetooth Low Energy GATT-Servers und agiert gleichzeitig als Bluetooth Low Energy Slave/Peripheral, welcher selbst keinen Aufbau einer Verbindung initiiert, sondern eine Verbindung durch werben (Advertising) ermöglicht. Die entwickelten Funktionen dieses Moduls und die eingesetzten Bluetooth Low Energy Bibliotheken sind in Abbildung 3.8 dargestellt.

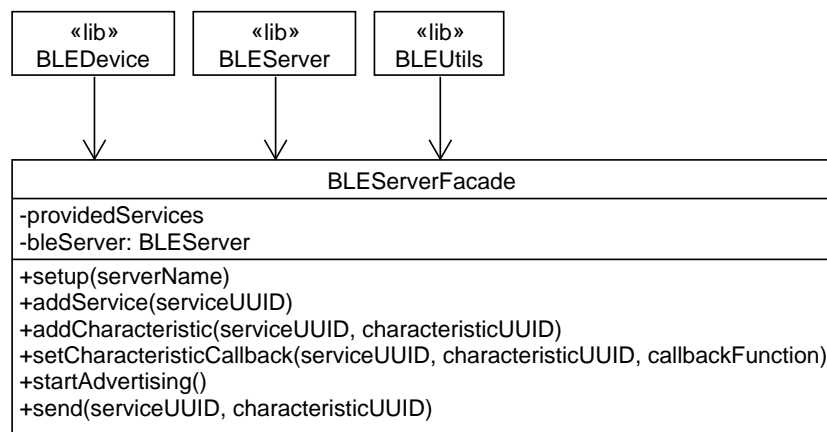


Abbildung 3.8: Aufbau des BLEServerFacade-Moduls.

Eine der wichtigsten Funktionen dieses Moduls ist die setup-Funktion. Sie initialisiert das BLE-Gerät mithilfe des BLEDevice-Moduls und erzeugt eine Server-Instanz. Außerdem werden für die Server-Instanz spezifische Server-Callbacks definiert. Zu den Callback-Funktionen zählen onConnect, in der das Verhalten nach einem Verbindungsaufbau definiert wird und onDisconnect, in der das Verhalten nach einem Verbindungsabbruch definiert wird.

Das BLEServerFacade-Modul basiert auf der Idee, dass der erstellte BLE-Server verschiedene Services mit zugehörigen Characteristics bereitstellt. Um diese zu verwalten, wurden zwei interne Datenstrukturen hinzugefügt. Entsprechend wurden die Funktionen addService und addCharacteristic implementiert, die es ermöglichen, Services und deren Characteristics hinzuzufügen. Diese Funktionen müssen vor weiteren Schritten aufgerufen werden, da der BLE-Server ansonsten keine nutzbaren Services oder Characteristics bereitstellt.

Registrierung von Callback-Funktionen

Nachdem mit den beschriebenen Funktionen erfolgreich ein BLE-Server mit den gewünschten Services und Characteristics erstellt wurde, mussten für einige Characteristics zusätzlich Reaktionen anhand von Callback-Funktionen implementiert werden. Dies ist wichtig, da eine Characteristic die write-Eigenschaft besitzen kann, die einem anderen BLE-Teilnehmer erlaubt, Werte auf dieser Characteristic zu schreiben.

Dafür wurde die Funktion `setCharacteristicCallback` hinzugefügt, mit der eine Characteristic aus der internen Datenstruktur ausgewählt und eine `onWriteCallback`-Funktion für diese registriert werden kann.

Sichtbarmachen der BLE-Services durch Advertising

Damit der BLE-Server für andere BLE-Geräte, die als Scanner agieren, sichtbar ist, wurde eine weitere Funktion, die `startAdvertising`-Funktion, hinzugefügt. Da die `BLEServerFacade` nicht nur als GATT-Server sondern auch als BLE-Peripheral agiert, muss dieser seine Services aktiv bewerben (advertising), dies erfolgt über die `startAdvertising`-Funktion. In dieser wird durch die Liste aller angebotenen Services iteriert, der entsprechende Service gestartet und die UUID des Service der Advertising-Instanz der BLE-Server-Instanz übergeben. Dadurch wird sichergestellt, dass der Server die gewünschten Services anbietet. Anschließend kann das Advertising gestartet werden.

Implementierung der send-Funktion

Die letzte Funktion die dem Softwaremodul `BLEServerFacade` hinzugefügt wurde ist die `send`-Funktion. Die `send`-Funktion dient dazu, mithilfe der Service-UUID und einer Characteristic-UUID, einen Wert auf einer Characteristic zu ändern. Ein weiterer Schritt in dieser Funktion ist der Aufruf von `notify` für die entsprechende Characteristic, um dem GATT-Client asynchron zu benachrichtigen, dass eine Änderung auf der entsprechenden Characteristic durchgeführt wurde.

Validierung des BLEServerFacade-Moduls

Zur Überprüfung der Funktionalität des BLE-Servers sowie zur Validierung der beworbenen Services und Characteristics wurde während der Entwicklung die nRF Connect App von Nordic Semiconductor eingesetzt. Die App bietet einen BLE-Scanner, der in der Umgebung nach verfügbaren BLE-Geräten sucht und mit diesen eine Verbindung aufbauen kann. Anschließend können die vom Server bereitgestellten Attribute sowie die internen GATT-Eigenschaften (Services, Characteristics) ausgelesen werden. Darüber hinaus

ermöglicht die App das Abfragen von Properties, Values und Descriptors der Characteristics sowie das Lesen (read), Ändern (write) und asynchrone Empfangen (notify) von Werten.

BLEClientFacade-Modul

Das BLEClientFacade-Modul wurde entwickelt, um die Details eines Bluetooth Low Energy GATT-Clients und eines Bluetooth Low Energy Masters zu abstrahieren. Die Abbildung 3.9 zeigt das Modul und die implementierten Funktionen zur Übersicht.

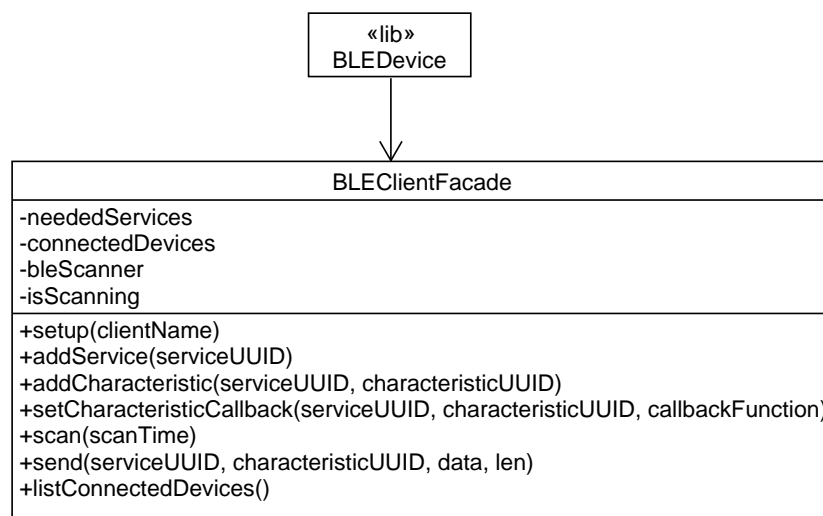


Abbildung 3.9: Aufbau des BLEClientFacade-Moduls.

Ähnlich wie im BLEServerFacade-Modul wurden auch hier Datenstrukturen für Services und Characteristics sowie die Funktionen addService, addCharacteristic und setCharacteristicCallback eingeführt. Diese Datenstrukturen und Funktionen ermöglichen es, die Services zu speichern, die der GATT-Client abonnieren möchte und Characteristics mit einer Callback-Funktion zu verknüpfen.

Zusätzlich wurde die Liste neededDevices angelegt, die vor dem Start des GATT-Clients initialisiert werden muss, da sonst keine passenden Services für die Verbindung mit einem GATT-Server gefunden werden können.

Zum Starten wurde ebenfalls eine setup-Funktion implementiert mit der ein BLE-Gerät initialisiert wird und anstelle eines BLE-Servers eine BLE-Scanner-Instanz erstellt wird. Für diese BLE-Scanner-Instanz werden in der setup-Funktion Einstellungen vorgenommen, darunter das Setzen der Advertised Device Callbacks.

Verarbeitung erkannter BLE-Geräte

Die Advertised Device Callbacks werden vom BLE-Protokoll aufgerufen, sobald ein neues BLE-Gerät vom Scanner detektiert wird. Es ist ähnlich zu dem Scanner in der nRF Connect App, es werden alle verfügbaren BLE-Geräte in der Nähe erfasst.

Für jedes erkannte Gerät wird eine Callback-Instanz ausgelöst. In dieser ist die onResult-Funktion implementiert, die überprüft, ob das erfasste Gerät die benötigten Services anbietet. Jede UUID aus der Liste der erforderlichen Services wird mit den angebotenen Services des erkannten Geräts verglichen. Falls eine Übereinstimmung gefunden wird, wird das Gerät zur Liste der verbundenen Geräte hinzugefügt.

Der eigentliche Scan-Vorgang erfolgt über die implementierte scan-Funktion. In dieser wird einmalig ein Scan-Vorgang für die Scanner-Instanz gestartet und mit einem Flag sichergestellt, dass der Scan nicht mehrfach ausgeführt wird.

Aufbau einer Verbindung zu einem BLE-Server

Nach einem durchgeführten Scan kann eine Verbindung zu den gefundenen Geräten hergestellt werden, falls diese noch nicht vorhanden sind. Dies wird für jedes erkannte Gerät mit einem isConnected-Flag überprüft. Ist das entsprechende Gerät noch nicht verbunden wird die implementierte connectToServer-Funktion aufgerufen.

In der connectToServer-Funktion wird für jedes noch nicht verbundene Gerät eine Client-Instanz erstellt und für diese Client-Instanz auch ähnlich zu dem Server eine onConnect- und eine onDisconnect-Callback-Funktionen registriert. Besonders wichtig ist, dass in der onDisconnect-Funktion das Gerät bei einem Verbindungsabbruch aus der Liste der verbundenen Geräte entfernt und alle für das Gerät reservierten Ressourcen freigegeben werden. Andernfalls kann es zu unerlaubten Speicherzugriffen kommen, die letztendlich zu Abstürzen des BLE-Clients führen.

Nach der Erstellung eines Clients und der Registrierung der Callback-Funktionen muss eine Verbindung zu dem Gerät mit dem Client Funktionsaufruf connect erfolgen. War die Verbindung erfolgreich, werden in der connectToServer-Funktion alle Services und Characteristics des verbundenen Servers in den entsprechenden Datenstruktur des verbundenen

Gerätes gespeichert. Dadurch kann der Client später auf alle Services und Characteristics eines remote BLE-Servers zugreifen.

Neben der Speicherung aller remote Services und Characteristics wird für bestimmte Characteristics, die in der Liste der erwarteten Services und Characteristics enthalten sind, eine `onNotifyCallback`-Funktion registriert. Diese Callback-Funktion wird zu Beginn mit der oben beschriebenen Funktion `setCharacteristicCallback` in der `Characteristic`-Datenstruktur hinterlegt. Die Callback-Funktion wird asynchron aufgerufen, sobald der Server mit seiner `send`-Funktion einen neuen Wert setzt und anschließend die `notify`-Funktion aufruft.

Implementierung der `send`-Funktion

Es wurde ebenfalls eine `send`-Funktion in dem Modul implementiert, um zum Beispiel ein Start-Kommando oder einen Zeitstempel zu übertragen. Dabei wird intern auf eine als Parameter übergebene `Characteristic` zugegriffen, deren Wert durch den Aufruf von `writeValue` gesetzt wird. Auch dieser Schreibzugriff kann auf der Serverseite durch die registrierte `onWriteCallback`-Funktion asynchron verarbeitet werden.

Implementierung der `listConnectedDevices`-Funktion

Zur Entwicklung und Überprüfung der Funktionalität des `BLEClientFacade`-Moduls wurden zudem die Funktion `listConnectedDevices` hinzugefügt, die von außen aufgerufen werden kann, um die aktuell verbundenen Geräte mit ihren jeweiligen Services und Characteristics auszugeben.

Alle zuvor beschriebenen Softwaremodule wurden anschließend in verschiedenen Empfangsknoten- und Sensorknoten-Modulen eingesetzt. Die konkrete Verwendung einzelner Funktionen sowie die Registrierung spezifischer Callback-Funktionen werden im folgenden Kapitel erläutert, da dies stark von den implementierten Algorithmen abhängt.

3.4.3 Herausforderung und Implementierung der Synchronisation

Während der Entwicklung der Sensor- und Empfangsknoten Firmware wurden in einem iterativen Prozess mehrere Versionen erstellt. Insbesondere im Hinblick auf eine erfolgreiche zeitliche Synchronisation der Sensordaten war dieses Vorgehen erforderlich. Ein erster wichtiger Entwicklungsschritt war die Identifizierung von Problemstellen im Sensornetzwerk. Die in Abbildung 3.10 dargestellte Skizze erwies sich dabei als hilfreich, da sie mögliche Verzögerungen visualisiert.

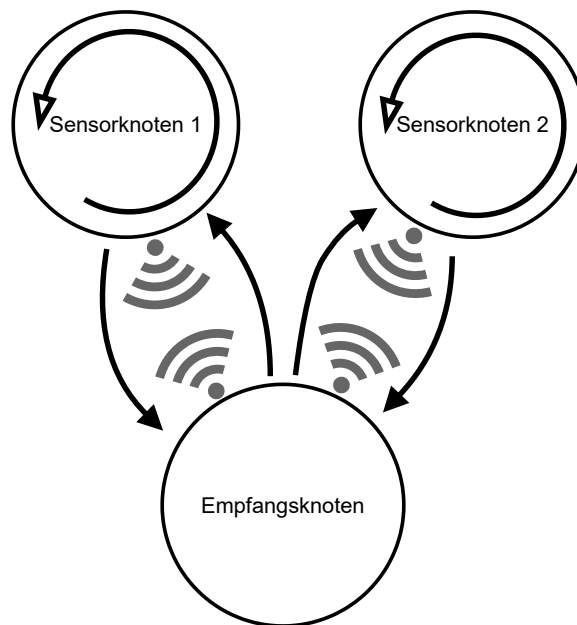


Abbildung 3.10: Die Skizze zeigt mit den schwarzen Pfeilen auf, wo zeitliche Verzögerungen auftreten können.

- Ein erstes Problem, das auftreten kann, ist eine Verzögerung vom Empfangsknoten zu den Sensorknoten.
- Ein weiteres Problem sind unterschiedlich laufende Echtzeituhren auf den Entwicklungsboards, die zu einem Clock Drift und zu unterschiedlichen Erfassungszeiten der Sensorwerte führen können.
- Das dritte mögliche Problem ist eine Verzögerung vom Sensorknoten zum Empfangsknoten hin.

Alle diese Faktoren können dazu führen, dass die Sensorwerte nicht zur erwarteten realen Zeit erfasst oder übermittelt werden, siehe dazu auch Kapitel 2.3.

Ein zusätzlicher Aspekt bei der Entwicklung der Netzwerkknoten war die Hardware-Beschränkung des ESP32C6-Entwicklungsboards. Da dieses kein symmetrisches Multiprocessing ermöglicht, mussten die Ausführungsschritte strikt sequentiell erfolgen, siehe Kapitel 3.3. Unter Beachtung dieser Punkte wurde der folgende Ablauf entwickelt:

- **Messung starten:** An alle Sensorknoten wird ein start Kommando verschickt, um eine Messung zu initiieren.
- **Sensoren auslesen:** Alle Sensorknoten nehmen separat für eine bestimmte Zeit oder eine bestimmte Anzahl an Datenpunkten Sensorwerte auf.
- **Messwerte sammeln:** Die Messwerte werden nach der Erfassung an den Empfangsknoten versendet, dieser sortiert diese und bereitet sie für die Weiterleitung an den Auswertungscomputer vor.
- **Messwerte weiterleiten:** Die Messwerte werden über die serielle Schnittstelle an den Auswertungscomputer versendet.

Im ersten Schritt wird die Messung gestartet. Dies soll durch ein start-Befehl initiiert werden. Das heißt der Empfangsknoten sendet ein start an die Sensorknoten und anschließend führen diese eine Messung für eine bestimmte Zeit durch. Hierbei muss beachtet werden, dass es, wie in der Abbildung zu sehen ist, zu Verzögerung zwischen dem Empfangsknoten und den Sensorknoten kommen kann. Wenn diese Verzögerung zwischen dem Empfangsknoten und mehreren Sensorknoten sehr unterschiedlich ist, starten die Sensorknoten entsprechend ihre Messung zu unterschiedlichen Zeiten. Diese Asynchronität führt letztendlich dazu, dass die Sensorknoten nicht nur zeitlich verzögert ihre Messung starten, sondern dadurch auch die gesamten erfassten Datenpunkte zeitlich verschoben werden.

Das Ziel besteht darin, die Messung möglichst synchron zu starten. Dies kann auf zwei Arten erreicht werden: Entweder einem Broadcast mit sehr geringer Verzögerung oder durch Uhrensynchronisation. Mit dem Ansatz der Uhrensynchronisation werden Softwareuhren auf den Sensorknoten mit der Uhr des Empfangsknotens vor dem eigentlichen Start der Messung synchronisiert. Über diese internen Softwareuhren kann nach der erfolgten Synchronisation und nach einer bestimmten Wartezeit, ein synchroner Start der Messung durchgeführt werden.

Nach dem erfolgreichen synchronen Start der Messung lesen alle Sensorknoten ihre PPG-Sensoren aus, entweder durch einen Hardwareinterrupt, einen Timerinterrupt oder aktives Polling in einer Schleife. Die Messung wird dabei bei dem Erreichen eines festgelegten Zeitpunkts oder beim Erreichen einer bestimmten Anzahl an Samples beendet.

Dazu wurde Anfangs angenommen, dass die Abtastrate auf beiden Sensorknoten gleich ist. In späteren Tests stellte sich jedoch heraus, dass diese Annahme für verschiedene Sensoren nicht zutrifft und Abweichungen in den Abtastraten auftreten können.

Daher bietet sich an genaue Zeitstempel anstelle von approximierten Zeitstempeln einzusetzen, um einen präziseren Vergleich zwischen den Messpunkten zu ermöglichen. Als Zeitstempel kann die im Synchronisationsschritt bestimmte Uhrenanpassung genutzt werden, um die aktuelle lokale Zeit auf eine gemeinsame globale Uhrzeit zwischen den Sensorknoten abzugleichen. Dies erfolgt nach der folgenden Formel:

$$\text{globaleZeit} = \text{offset} + \text{lokaleEchtzeitUhr} - \text{drift}. \quad (3.1)$$

Anschließend werden alle Sensordaten von den Sensorknoten an den Empfangsknoten übertragen. Hierbei könnten auch Verzögerungen zwischen den Sensorknoten und dem Empfangsknoten entstehen, diese sind aber für die zeitliche Reihenfolge nicht relevant, da alle Messwerte gesammelt werden und durch den Zeitstempel eindeutig sortiert werden können.

Im letzten Schritt verarbeitet der Empfangsknoten alle gesammelten Messwerte und ordnet sie mit den entsprechenden Zeitstempeln in eine Tabelle ein. In dieser Tabelle werden für jeden Sensor eine Spalte für Zeitstempel sowie die Infrarot- und Rotlichtwerte angelegt. Die Tabelle 3.3 zeigt exemplarisch erfasste Sensordaten mit zugehörigen Zeitstempeln.

Tabelle 3.3: Beispieltabelle mit Zeitstempeln und Sensordaten.

Zeitstempel-1	IR-1	ROT-1	Zeitstempel-2	IR-2	ROT-2
0,000s	125	140	0,000s	128	135
0,010s	130	145	0,010s	129	136
0,020s	128	143	0,020s	130	137
0,030s	132	147	0,030s	132	139
0,040s	131	146	0,040s	133	140

Im letzten Schritt wird diese Tabelle vollständig an den Auswertungscomputer übermittelt. Bevor diese vier Schritte nicht vollständig abgeschlossen sind, kann keine weitere Messung gestartet werden, da erst sichergestellt werden muss, dass alle Sensordaten vollständig am Auswertungscomputer angekommen sind.

Mit den zuvor dargestellten Schritten wurden ein Sensorknoten und ein Empfangsknoten entwickelt, die diese Schritte umsetzen. Zuerst wird der Sensorknoten beschrieben, gefolgt von der Umsetzung des Empfangsknotens. Beide nutzen die zuvor beschriebenen BLE-Module, der Sensorknoten zusätzlich das PPG-Sensormodul.

Umsetzung des Sensorknotens

Die folgende Abbildung 3.11 zeigt zur Übersicht die Struktur des Sensorknoten-Moduls.

Sensorknoten
bleServer: BLEServerFacade ppgSensor: PPGSensor name service irCharacteristic redCharacteristic cmdCharacteristic
doMeasurement() sendData() taskSync() taskMeasurement() taskSend()

Abbildung 3.11: Aufbau eines Sensorknoten-Moduls.

Für die Umsetzung eines Sensorknotens mit den oben beschriebenen Ansätzen und Komponenten mussten zunächst die notwendigen Komponenten PPGSensor und BLEServerFacade importiert werden. In der für die Arduino-Umgebung üblichen setup-Funktion wurden das PPG-Sensormodul sowie die BLEServerFacade-Instanz initialisiert. Dabei wurde der Modus für die Instanz des PPG-Sensormoduls auf Polling gesetzt.

Für die BLEServerFacade-Instanz wurde ein Service mit der Funktion addBLEService hinzugefügt. Diesem Service wurden zunächst mithilfe der Funktion addBLECharacteristic die Charakteristiken ir, red und cmd hinzugefügt. Die eigens definierte, 128-Bit große UUID, mit der der Service identifiziert wird, hat folgenden Aufbau:

```
12345678-1234-5678-1234-56789abcdef0
```

Die Characteristic-UUIDs folgen demselben Aufbau, wobei die letzte Ziffer jeweils inkrementell erhöht wird.

Die Idee ist, dass jeder Sensorknoten einen BLE-Service „Sensor“ anbietet und für jeden Nachrichtentyp, wie Infrarot, Rotlicht, Kommandos und weitere, eine eigene Characteristic bereitstellt.

Über den Wert (Value) der Characteristic, beispielsweise der Infrarot- oder Rotlicht-Characteristic, soll der tatsächliche Sensorwert binär kodiert übertragen werden. Das erfolgt durch die entwickelte send-Funktion.

Über die cmd-Characteristic sollen Kommandos wie das Start-Kommando oder bestimmte Einstellungen vom Empfangsknoten empfangen werden. In der zugehörigen onWrite-Callback-Funktion, die mit der setCharacteristicCallback-Funktion registriert wird, werden Befehlspakete entgegengenommen und verarbeitet. Bei einem Start-Befehl werden dann weitere Schritte, wie eine Synchronisation oder eine direkte Messung, initiiert.

Nach der Initialisierung und dem Hinzufügen der beschriebenen Characteristics wird der Server in der setup-Funktion mit der startAdvertising-Funktion gestartet.

Anschließend kann der Sensorknoten potenzielle Verbindungsanfragen vom Empfangsknoten sowie Befehle wie den Start-Befehl empfangen und verarbeiten. Damit der Sensorknoten Aufgaben ausführen kann, wurden die beiden Funktionen doMeasurement und sendData implementiert.

Die doMeasurement-Funktion liest die Sensorwerte direkt aus dem PPG-Sensormodul und speichert sie in Listen. Die sendData-Funktion dient dazu, die in den Listen gespeicherten Sensorwerte sequentiell an den Empfangsknoten zu übertragen und anschließend die Listen zu leeren.

Alle drei Schritte, der synchrone Start, die Messung und das Versenden der Sensordaten wurden als Zustandsmaschine in drei separate Tasks aufgeteilt. Im Synchronisations-Task wird auf das Erreichen der Startzeit gewartet. Anschließend wird die Messung im

Mess-Task mithilfe der doMeasurement-Funktion durchgeführt bis entweder die Messzeit abgelaufen ist oder die erforderliche Anzahl an Messwerten erreicht wurde (Abbruchbedingung für die Messung erfüllt ist). Im letzten Task werden die Sensordaten über die serielle Schnittstelle mit der sendData-Funktion an den Auswertungscomputer übertragen.

Umsetzung des Empfangsknotens

Zur Übersicht stellt die folgende Abbildung 3.12 das entwickelte Modul des Empfangsknotens dar.

Empfangsknoten
bleClient: BLEClientFacade name serviceNode1 serviceNode2 irCharacteristic redCharacteristic cmdCharacteristic
processSerialCommands() processReceivedSensorData() sendDataToSerial()

Abbildung 3.12: Aufbau eines Empfangsknoten-Moduls.

Der Empfangsknoten durchläuft, ähnlich wie der Sensorknoten, eine Initialisierungsphase in der setup-Funktion, in der die Services und Characteristics, die er abonnieren möchte, registriert werden.

Da es mehrere Sensorknoten gibt, werden diese über unterschiedliche Service-UUIDs unterschieden. Dementsprechend muss je ein Service für den ersten und den zweiten Sensorknoten hinzugefügt werden.

Da auch unbekannt ist, welche Characteristics die jeweiligen Sensorknoten anbieten, werden jeweils auch die Characteristics ir, red und cmd hinzugefügt. Für alle Characteristics werden zudem Callback-Funktionen definiert und registriert. Anschließend wird der BLE-Client mit der setupBLEClientFacade-Funktion gestartet.

In der loop-Funktion können über die definierte processSerialCommands-Funktion Befehle von der seriellen Schnittstelle an den Empfangsknoten gesendet werden. Einer dieser Befehle ist das Start-Kommando. Wird dieses erkannt, so wird ein Start-Befehl mit send an alle verbundenen Sensorknoten gesendet. Neben dem Befehl können auch die Messdauer und die Abtastrate eingestellt werden.

Nachdem eine Messung abgeschlossen ist, werden alle Sensordaten über die definierten Callback-Funktionen, wie sensorOneIrNotifyCallback oder sensorTwoRedNotifyCallback, empfangen. In diesen Funktionen werden die Sensordaten mithilfe der processReceivedSensorData-Funktion in Datenstrukturen gespeichert.

Der Abschluss der Datenübertragung wird durch das von einem Sensorknoten gesendete Kommando ready signalisiert. Da der Ablauf strikt sequentiell erfolgen muss, werden zunächst alle Sensordaten des ersten und anschließend die des zweiten Sensorknotens empfangen. Dies wird über Flags gesteuert: Ist der erste Knoten bereit (ready), wird ein Befehl an den zweiten Knoten gesendet, sodass dieser seine Sensordaten übertragen kann.

Sobald alle Sensordaten empfangen wurden, was durch das letzte ready signalisiert wird, werden diese in der zuvor beschriebenen Tabellenstruktur an die serielle Schnittstelle übertragen und von einem Auswertungscomputer empfangen. Dies erfolgt über sendDataToSerial die Funktion.

Dies ist die grundsätzliche Struktur der Umsetzung mit den Bluetooth Low Energy Modulen. Da die aktuelle Implementierung keine parallele oder Broadcast-Kommunikation erlaubt, startet der zweite Sensorknoten unvermeidlich verzögert. Um dieses Problem zu lösen, wurde in einem weiteren Entwicklungsschritt eine Uhrensynchronisation implementiert, die sich an der Synchronisationsphase des *Timing-Sync Protocol for Sensor Networks* (TPSN) orientiert.

Implementierung der Synchronisation mit TPSN

Für die Implementierung des Uhren-Synchronisationsprotokolls mussten die vier Zeitstempel t_1 , t_2 , t_3 und t_4 im Sensorknoten deklariert werden. Damit diese gleichzeitig zur berechneten Zeit starten, wurde zusätzlich eine Variable `delayAfterStart` eingeführt. Zudem wurden die globalen Variablen `drift` und `offset` hinzugefügt, die mithilfe der Formeln des TPSN-Protokolls bestimmt werden (siehe Kapitel 2.3.3).

Zur Initiierung der Synchronisation wurde die Funktion `startTimeSync` implementiert, die nach dem Empfang eines Start-Befehls aufgerufen wird. In dieser Funktion wird der erste Zeitstempel t_1 erfasst und über die neu definierte `time-sync-Characteristic` an den Empfangsknoten gesendet.

Es wurden sowohl für den Empfangsknoten, die neue `time-sync-Characteristic` als auch für den Sensorknoten, die `time-ack-Characteristic` hinzugefügt, damit jegliche Synchronisationsnachrichten separat von den Kommandos verarbeitet werden und mögliche Verarbeitungsverzögerungen vermieden werden.

Auf der Master-Knoten-Seite wird die Nachricht des jeweiligen Sensorknotens in der zugehörigen `timeNotifyCallback`-Funktion verarbeitet. In dieser Funktion wird der zweite Zeitstempel t_2 beim Empfang der Nachricht erfasst.

Da das direkte Versenden einer Antwort (ACK-Nachricht) innerhalb der Callback-Funktion zu einer Blockierung des BLE-Stacks führen könnte, wurde eine Nachrichten-Queue implementiert. Während der aktuellen `timeNotifyCallback`-Funktion wird die Rücknachricht zunächst in die Queue eingefügt.

Sobald wieder Rechenzeit zur Verfügung steht, wird diese Queue in der `loop`-Funktion mithilfe der `processQueue`-Funktion ausgelesen und alle verfügbaren Nachrichten verarbeitet. Für jede Nachricht in der Queue wird vor dem Versand der Zeitstempel t_3 (Sendezeitstempel) erfasst und der Nachricht hinzugefügt. Anschließend wird die Nachricht als `Timestamp`-Datenstruktur an den Sensorknoten gesendet.

Der Sensorknoten empfängt die ACK-Nachricht in der `onTimeSync`-Callback-Funktion. Diese Funktion ist für den Sensorknoten zentral, da sie die empfangene Nachricht ausliest und die enthaltenen Zeitstempel verarbeitet. Der letzte Zeitstempel t_4 wird erfasst, und mithilfe aller Zeitstempel werden `drift` und `offset` berechnet.

Mit den bestimmten Werten wird anschließend die lokale Echtzeituhr angepasst und eine globale Uhrzeit bestimmt. Dazu wurde die Funktion `calculateRemoteTimestamp` implementiert und die oben bereits erwähnte Formel 3.1 für den globalen Zeitstempel angewendet.

Diese Funktion ermöglicht nicht nur eine globale Zeitreferenz für einen synchronen Start, sondern auch die Erfassung synchronisierter Zeitstempel während der Messphase.

Im Synchronisations-Task wird mithilfe einer bedingten Anweisung überprüft, ob der berechnete Remote-Zeitstempel größer oder gleich dem ursprünglichen Zeitstempel vor dem Versand des Start-Befehls (t_s) plus einer definierten Verzögerung (`delayAfterStart`) ist und ob der Zeitstempel t_4 bereits erfasst wurde. Falls diese Bedingungen erfüllt sind, wird die Messung gestartet.

Inwiefern die implementierten Ansätze die Synchronisation erfolgreich gewährleisten, wird in Kapitel 4.2 beschrieben.

3.5 Entwicklung der Signalverarbeitung

Die Signalverarbeitung wurde in grundlegende Verarbeitungsschritte unterteilt, die in Abbildung 3.13 dargestellt sind. Diese Schritte basieren auf den Methoden vorheriger Arbeiten (siehe Kapitel 2.4).



Abbildung 3.13: Darstellung der entwickelten Signalverarbeitungskette.

Die Datenerfassung umfasst die Aufnahme von Sensordaten während einer Messung sowie das Einlesen der gespeicherten Sensordaten für die anschließende Auswertung.

Im Schritt der Vorverarbeitung werden die Daten auf einen einheitlichen Wertebereich normalisiert. Zudem werden potenzielle Störartefakte gefiltert, wobei zu beachten ist, dass relevante Merkmale erhalten bleiben und möglichst keine Phasenverschiebung durch den Filter entsteht.

Die Merkmalsextraktion erfolgt im dritten Schritt. Hierbei werden charakteristische Punkte der Pulswelle mittels eines Peak-Detektionsalgorithmus bestimmt. Anschließend werden einzelne Bereiche analysiert und weitere Merkmale extrahiert.

Im letzten Schritt der Indikatorenbestimmung werden die medizinisch relevanten Größen, wie die Pulswellenlaufzeit und die Pulswellengeschwindigkeit, berechnet.

Die Implementierung der Signalverarbeitung erfolgte in der Entwicklungsumgebung Visual Studio Code unter Verwendung der Programmiersprache Python. Jeder Verarbeitungsschritt wurde modular in separaten Python-Modulen implementiert. Die konkrete Implementierung der Signalverarbeitungsschritte wird im Folgenden detailliert erläutert.

Für die Signalverarbeitung wurden die folgenden Python-Bibliotheken eingesetzt:

- Pandas und NumPy zur effizienten Verarbeitung und Manipulation von Datenstrukturen [62, 25],
- SciPy für erweiterte Signalverarbeitung [68],
- Matplotlib zur Visualisierung der Daten [28],
- Neurokit2 für die Analyse und Verarbeitung von Biosignalen [41].

Zur Steuerung und zur Erfassung der Sensordaten wurde ein Python-Skript entwickelt, mit dem der Empfangsknoten gesteuert und die Daten von den Sensorknoten ausgelesen werden. Über die Software lassen sich Parameter wie Messdauer und Abtastfrequenz der Sensorknoten einstellen, ohne die Firmware neu kompilieren zu müssen.

Die erfassten Sensordaten der zwei Sensorknoten wurde während einer selbst definierten Messdauer in Sekunden über die serielle Schnittstelle eingelesen und in dem Format: "Timestamp_ID, IR_Value_ID, Red_Value_ID" in einer CSV-Datei gespeichert.

Nach Abschluss der Messung wurden die Sensordaten aus der CSV-Datei eingelesen. Mithilfe der Standardbibliothek OS wurde die Datei aus dem entsprechenden Systempfad geladen. Alle Rohdaten (Zeitstempel, Infrarot- und Rotlichtwerte) wurden dabei in separaten 2D-Arrays gespeichert, um eine strukturierte Weiterverarbeitung zu ermöglichen.

Während des Einlesevorgangs wurden die Sensordaten invertiert, um die Wellenform korrekt darzustellen, dies entspricht eine gängigen Methode in der PPG-Signal-Analyse [35].

Anschließend wurden Parameter wie die Anzahl der Datenpunkte, die Messdauer und die tatsächliche Abtastrate bestimmt. Die Zeitstempel der zwei Sensorknoten wurde zudem mithilfe der Kreuzkorrelation auf Kohärenz zueinander untersucht.

Im nächsten Schritt wurden die Infrarot- und Rotlichtdaten mit Filtern verarbeitet. Es wurden verschiedene Filter eingesetzt, um deren Einfluss auf das Signal und vor allem auf die Morphologie des Signals zu untersuchen. Aufgrund des Einfluss auf die Form des Signals kann die Auswahl der Filter die resultierenden Pulswellenlaufzeiten und Pulswellengeschwindigkeiten stark beeinflussen.

Als erster Filter wurde ein Savitzky-Golay-Filter implementiert [49, 24, 54]. Die Fenstergröße und die polynomiale Ordnung wurde durch austesten bestimmt. Als gute Parameter haben sich eine Fenstergröße von 91 und eine polynomiale Ordnung von 6 herausgestellt. Durch die Fenstergröße von 91 wird das Signal gut geglättet und durch die polynomiale Ordnung von 6 bleiben dennoch wichtige Spitzen und Merkmale erhalten. Der Filter wurde mit der SciPy-Funktion `savgol_filter` umgesetzt [54].

Zusätzlich wurden weitere Filter untersucht, darunter ein Butterworth-Hoch- und Tiefpassfilter [37, 46]. Die untere Grenzfrequenz wurde nach mehreren Durchläufen auf 0,5 Hz festgelegt, um durch Atmung oder Bewegung verursachte Schwankungen zu reduzieren [64]. Die obere Grenzfrequenz wurde nach mehreren Testläufen von 20 Hz auf 10 Hz angepasst.

Für den Filter wurde eine Ordnung von 2 gewählt, um die Phasenverschiebung möglichst gering zu halten. Sie kann in diesem System die Ergebnisse maßgeblich beeinflussen, da die Signalform für eine präzise Abstandsmessung sehr genau sein muss [35, 38].

Der Filter wurde mit der Funktion `butter` der SciPy-Bibliothek als Second-Order-Section-Filter erstellt und mit der `sosfiltfilt`-Funktion angewendet. Es ist hierbei wichtig zu betonen, dass die `filtfilt`-Variante eingesetzt wurde, da diese den Filter einmal vorwärts und einmal rückwärts auf das Signal anwendet und so eine Null-Phasen-Verschiebung erreicht wird. Alle beschriebenen Parameter und die Auswahl des Filters orientieren sich an den vorherigen Forschungsarbeiten, die im Kapitel 2.4 vorgestellt wurden [37, 46].

In einem weiteren Schritt wurde mit dem Nutzsignal und dem Störsignal das Signal-Rausch-Verhältnis (engl. Signal-to-Noise-Ratio, SNR) berechnet und ausgegeben [72, 64, 49]. Außerdem wurde mit den gefilterten Signalen eine Kreuzkorrelation durchgeführt, um deren Ähnlichkeit zu quantifizieren [42, 26].

Sowohl das Signal-Rausch-Verhältnis als auch die Kreuzkorrelation, sollen als Parameter für die Einschätzung der Signalqualität dienen und in der späteren Untersuchung des Endsystems einen möglichen Zusammenhang zwischen realistischen und unrealistischen Pulswellenlaufzeiten und Pulswellengeschwindigkeiten aufzeigen [72, 64, 21, 26, 49].

Die gefilterten Infrarot- und Rotlichtspalten wurden anschließend auf den Wertebereich von Null bis Eins mithilfe der Min-Max-Normalisierung normalisiert. Die Normalisierten Sensordaten wurden in weiteren Datenstruktur gesichert, um mit diesen separat weiter zu arbeiten.

Den nächsten entscheidende Schritt der Signalverarbeitungskette stellt die Merkmalsextraktion dar. Für die Bestimmung der Pulswellenmerkmale musste zunächst eine Peak-Detektion implementiert werden.

Zur Implementation einer Peak-Detektion stehen verschiedene Möglichkeiten zur Verfügung. Zuerst wurde eine Peak-Detektion mittels der SciPy-Funktion `find_peaks` umgesetzt. Da diese jedoch nicht zuverlässig arbeitete, wurde die für PPG-Signale optimierte Funktion `ppg_process` der Neurokit2-Bibliothek verwendet. Diese Funktion gibt eine Datenstruktur mit analysierten PPG-Signalen zurück, einschließlich der detektierten Maxima.

Neben der Peak-Detektion eignet sich diese Bibliothek auch gut um weitere Eigenschaften der Pulswellen zu untersuchen. So stellt diese Bibliothek eine Plot-Funktion zur Verfügung mit der einige zuvor analysierte Daten, wie zum Beispiel der Puls, dargestellt werden können. Dies ist ebenfalls hilfreich, um die Qualität der Signale zu überprüfen.

Nachdem die Maxima bestimmt wurden, wurden die charakteristischen Punkte der Pulswellen identifiziert. Dazu wurden das zu untersuchende Signal, die detektierten Peaks und die Abtastrate an die `fiducial_points`-Funktion übergeben und die Merkmale bestimmt. Diese Funktion basiert auf der Arbeit von Elisa Mejía-Mejía und Peter Charlton [11].

Es wurden wesentliche Merkmale der Pulswellen, wie des original Signals, der ersten, der zweiten und der dritten Ableitung extrahiert. Dazu gehören die in dem Kapitel 2.4 erwähnten Merkmale der Pulswellen. Die Funktion gibt eine Datenstruktur (Pandas DataFrame) zurück in dem für jeden Merkmalspunkt in den Spalten die identifizierten Indizes der Merkmale enthalten sind.

Im finalen Schritt wurden die Pulswellenlaufzeit und Pulswellengeschwindigkeit bestimmt. Dazu wurde zunächst zwischen jeweils zwei erfassten Merkmalspunkten die Pulswellenlaufzeit berechnet. Hierfür wurde die Funktion `calculate_pulse_transit_times` entwickelt. In dieser werden jeweils eine Merkmalspalte von den ersten und dem zweiten Sensorknoten ausgewählt. Anschließend werden die zwei miteinander in der `find_nearest_indices`-Funktion verglichen und für den zweiten Sensor der nächstgelegenen Index zurückgegeben [26].

Mit den Indizes des ersten Sensorknotens und den passenden Indizes des zweiten Sensorknotens werden anschließend die Zeitstempel aus den Zeitstempel Datenstrukturen extrahiert und die Differenz zwischen allen passenden Zeitstempeln berechnet. Das Ergebnis wird anschließend für die Weiterverarbeitung von Mikrosekunden in Sekunden umgewandelt und als Tabelle (Pandas DataFrame) zurück gegeben.

Zusätzlich wurde ein weiteres Verfahren zur Berechnung der Pulswellenlaufzeit implementiert, das sich am Diastole-Patching-Ansatz orientiert [5]. Es werden jeweils zwei Bereiche um den Tangenten-Intersektionspunkt extrahiert und mittels Kreuzkorrelation verglichen. Der Punkt mit der höchsten Korrelation wird zur Bestimmung der Pulswellenlaufzeit gewählt.

Es wurden folgende Schritte implementiert:

1. Zunächst wurden die Indizes der Tangenten-Intersektionspunkte aus der Datenstruktur der Merkmalspunkte extrahiert, ähnlich zur zuvor beschriebenen Funktion.
2. Anschließend wurden die Indizes der nächstgelegenen Punkte im zweiten Signal bestimmt.
3. Im dritten Schritt wurde von den beiden zu vergleichenden Signale die relevanten Signalbereiche und die zugehörigen Zeitstempel extrahiert. Dies geschah in einem vordefinierten Fensterbereich, das sich symmetrisch um den Tangenten-Intersektionspunkt erstreckt.
4. Im vierten Schritt erfolgte die Kreuzkorrelation und die Bestimmung des Index mit der höchsten Übereinstimmung. Dazu wurde die SciPy-Funktion `correlate` eingesetzt.
5. Darauf erfolgte die Korrektur des ursprüngliche Index des Tangenten-Intersektionspunkts um den zuvor bestimmten Index der maximalen Korrelation.

6. Schließlich wurden die zwei Zeitstempel an den korrigierten Indizes extrahiert und jeweils voneinander subtrahiert, um die Pulswellenlaufzeit zu erhalten. Die Ergebnisse wurden ebenfalls in einer Tabelle (Pandas DataFrame) gespeichert. Zudem wurden verschiedene Fenstergrößen getestet, um eine optimale Fenstergröße zu erreichen.

Am Ende der Signalverarbeitungskette erfolgte die Berechnung der Pulswellengeschwindigkeit mit der Funktion `calculate_pwvs`. Die Funktion nimmt eine Datenstruktur mit Pulswellenlaufzeiten sowie den arteriellen Abstand in Metern als Parameter entgegen und berechnet für alle enthaltenen Pulswellenlaufzeiten die Pulswellengeschwindigkeit.

Die entwickelten Methoden wurden anhand verschiedener Messungen validiert, deren Ergebnisse in Kapitel 4 beschrieben sind. Abschließend wurden die erfassten Parameter (Pulsfrequenz, Pulswellenlaufzeit, Pulswellengeschwindigkeit) ausgegeben und mit Matplotlib die Signale visualisiert, um eine detaillierte Analyse zu ermöglichen.

4 Funktionale Systemüberprüfung

Neben der Implementierung der geforderten Eigenschaften ist die Überprüfung dieser Funktionen essenziell, um die Qualität der Umsetzung und die Erfüllung der Anforderungen zu bewerten.

Dazu wird zunächst die Überprüfung der drahtlosen Anforderungen vorgestellt, wobei die Auswahl der Kommunikationsprotokolle sowie die Synchronisationstechniken eingehend analysiert werden.

Anschließend wird die Analyse der Datenerfassung betrachtet, wobei insbesondere die Wahl zwischen Interrupt-, Timerinterrupt- und Polling-Verfahren sowie die relevanten Sensorparameter untersucht werden.

Daraufhin werden die Versuche zur Signalverarbeitungskette vorgestellt, insbesondere im Hinblick auf Resampling-Methoden und die Auswahl geeigneter Filterparameter.

Das Kapitel schließt mit einem Versuch zu der Betriebsdauer des Sensorsystems ab.

In jedem Abschnitt werden die durchgeführten Experimente ausführlich beschrieben, die Ergebnisse präsentiert und anschließend in den fachlichen Kontext eingeordnet, um ihre Bedeutung in Bezug auf die Systemanforderungen einzuordnen.

4.1 Vergleich von Bluetooth Low Energy und Thread

Zur Auswahl eines geeigneten Kommunikationsprotokolls für das drahtlose Sensorsystem wurde neben einer Literaturrecherche ein experimenteller Vergleich der Sendegeschwindigkeiten sowie der Round-Trip-Time zwischen Bluetooth Low Energy und Thread durchgeführt.

Durchführung

Für beide Protokolle wurde ein Demo-Programm implementiert, das ein einzelnes Zeichen überträgt. Die Zeitmessung erfolgte in Mikrosekunden sowohl vor als auch nach dem Aufruf der Sende-Funktion. Die Differenz dieser Zeitstempel definiert die Sendegeschwindigkeit. Diese Messung ist für das Sensorsystem relevant, da das verwendete Entwicklungsboard kein Symmetric Multiprocessing unterstützt und somit kein echter Parallelbetrieb möglich ist.

Zusätzlich wurde die Round-Trip-Time bestimmt. Hierzu wurde ein Zeitstempel vor dem Senden einer Nachricht und ein weiterer nach dem Empfang der Antwort erfasst.

Für die BLE-Messungen wurde ein Testaufbau bestehend aus einem BLE-Server, der als Sensorknoten agiert und einem BLE-Client der als Empfangsknoten agiert, realisiert. Der Versuch für das Thread-Protokoll basierte auf der OpenThread-Bibliothek und bestand aus einem Thread Leader sowie einem Thread Child.

Ergebnisse

Der Versuch ergab für Bluetooth Low Energy eine ungefähre Sendedauer in einem Bereich von 800 bis 950 μ s. Nach mehreren Durchläufen wurde eine durchschnittliche Round-Trip-Time von etwa 70 ms ermittelt.

Für das Thread-Protokoll wurde eine Sendedauer von ca. 3000 bis maximal 6000 μ s gemessen. Die Round-Trip-Time betrug durchschnittlich 37 ms.

Die Messergebnisse zeigen, dass BLE eine kürzere Sendedauer aufweist, jedoch eine höhere Latenz bei der Übertragung von Daten im Vergleich zu Thread. Dennoch wurde BLE für das Sensorsystem ausgewählt, da in dieser Arbeit kein Mesh-Netzwerk realisiert wird und BLE mit einer höheren Datenrate von bis zu 1 Mbit/s eine größere Bandbreite bietet.

Der vorgestellte Synchronisationsansatz sowie die Uhrensynchronisation können jedoch zukünftig auch mit dem Thread-Protokoll in einem größeren Mesh-Netzwerk mit mehreren Sensorknoten integriert werden.

Die folgenden Synchronisationstests zeigen, dass auch Bluetooth Low Energy mit der implementierten Synchronisation die zeitlichen Anforderungen erfüllt.

4.2 Überprüfung der Synchronisation

Die Synchronisation des Sensorsystems wurde mit zwei Ansätzen überprüft mit einem externen und einem internen Verfahren. Extern wurde die Synchronisation mit einer schaltbaren Folie überprüft und intern durch den Vergleich der berechneten Remote-Zeitstempel, die jeweils durch Timerinterrupts auf den Sensorknoten bestimmt und erfasst werden.

4.2.1 Überprüfung der Synchronisation und Funktion der Sensorknoten mittels einer schaltbaren PDLC-Folie

Aufbau

Die Synchronisation wurde extern durch eine elektronisch ansteuerbare Polymer Dispersed Liquid Crystal (PDLC)-Folie überprüft. Diese Folie wurde mit einem Signalgenerator angesteuert, der ein Rechtecksignal mit einer Periodendauer von 0,5 s, einer Amplitude von 10 V und einer Frequenz von einem 1 Hz erzeugt, siehe Abbildung 4.1. Die Sensorknoten wurden mit den Sensoren nach unten auf die PDLC-Folie gelegt, wie in Abbildung 4.2 zu sehen ist.

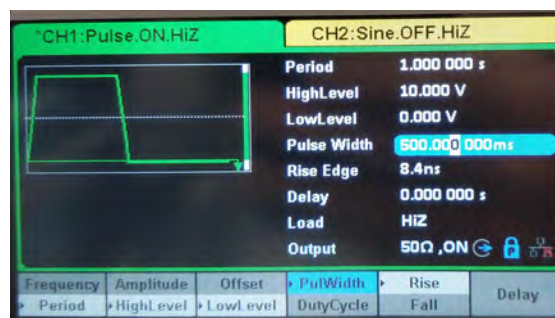


Abbildung 4.1: Einstellungen des Signalgenerators.



Abbildung 4.2: Versuchsaufbau mit den zwei Sensorknoten und der PDLC-Folie.

Durchführung

Über das Python-Skript zur Steuerung des Empfangsknotens wurden Befehle wie `set duration` oder `set samplerate` ausgeführt, um den Einfluss verschiedener Einstellungen auf die Synchronisation zu untersuchen. Als Testparameter wurden Messungen mit einer Dauer von 1, 2, 5, 10 und 30 Sekunden durchgeführt mit jeweils einer Abtastrate von 100 Hz, um die Auswirkungen unterschiedlicher Messdauern auf die Synchronisation und die Stabilität der Synchronisation über längere Zeiträume hinweg zu untersuchen. Die erfassten Messwerte wurden anschließend mit der zuvor beschriebenen Signalverarbeitung ausgewertet.

Ergebnisse

Die aufgezeichneten Signale in Abbildung 4.3 zeigen, dass die Sensorknoten synchron messen und erwartungsgemäß funktionieren. Es lassen sich rechteckige Signalfanken mit einer Periodendauer von ca. 0,5 Sekunden erkennen, die ein synchrones Verhalten aufweisen. Aufgrund von Unsicherheiten wie Messrauschen wurde ein weiterer Versuch durchgeführt, um die Synchronisation genauer zu überprüfen.

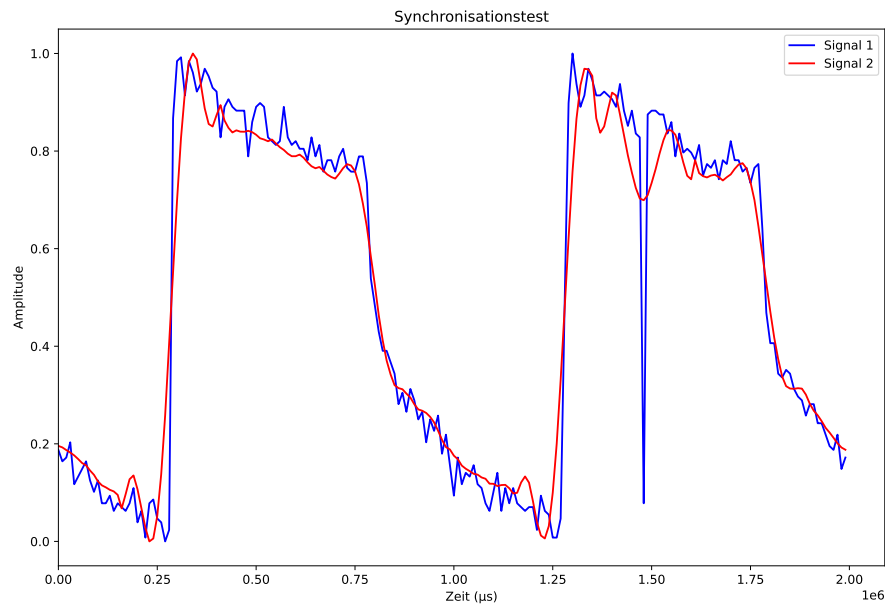


Abbildung 4.3: Die erfassten Rechtecksignale durch die Sensoren.

4.2.2 Vergleich der globalen Zeitstempel auf den Sensorknoten

Aufbau

Um die Präzision der Zeitstempel auf den Sensorknoten zu überprüfen, wurde ein weiterer Versuch durchgeführt, in dem ausschließlich Zeitstempel erfasst wurden. Zunächst wurden auf beiden Sensorknoten die Uhrenanpassungsparameter, Offset und Drift, bestimmt. Anschließend wurden auf den beiden Sensorknoten synchronisierte Echtzeituhren gestartet, die in periodischen Intervallen Interrupts auslösten. Bei jedem ausgelösten Interrupt wurde ein globaler Zeitstempel durch die `calculateRemoteTime`-Funktion berechnet und in einer Datenstruktur gespeichert. Dadurch wurde sichergestellt, dass beide Sensorknoten auf einer einheitlichen Zeitbasis arbeiten und miteinander verglichen werden konnten.

Durchführung

Die Messungen wurden mit unterschiedlichen Messdauern und Interrupt-Intervallen durchgeführt. Aufgrund begrenzter Speicherkapazitäten durfte die maximale Anzahl von 20.000 Datenpunkten, also $20.000 \text{ Datenpunkte} \times 4 \text{ Byte} \times 2 \text{ Sensorknoten} = 160.000 \text{ Byte}$, nicht überschritten werden. Es wird erwartet, dass die Zeitstempel auf den zwei Sensorknoten

sehr ähnlich sind. Außerdem wird erwartet, dass der Abstand zwischen den einzelnen Zeitstempel dem Interrupt-Intervall entspricht. Sollte es starke Abweichung zwischen den Zeitstempeln der Sensorknoten im Bereich von Mikrosekunden geben, würde dies auf eine nicht ausreichende Synchronisation hinweisen. Folgende Testbedingungen wurden untersucht 4.1:

Tabelle 4.1: Testparameter für den internen Synchronisationstest mit den Zeitstempeln.

Start Reihenfolge der Knoten	Messdauer	Intervall
Sensorknoten Empfangsknoten	1 s	50 μ s
Empfangsknoten Sensorknoten	1 s	50 μ s
Empfangsknoten Sensorknoten	1 s	50 μ s
Empfangsknoten Sensorknoten	10 s	500 μ s

Ergebnisse

Wie der Ausschnitt 4.4 für eine 10 Sekunden Messung mit Intervallen von 500 μ s zeigt, sind die Zeitstempel bis auf Abweichungen im hundertstel Mikrosekundenbereich nahezu identisch. Auf beiden Sensorknoten wurden die Zeitstempel während der 10 Sekunden gleichmäßig erhöht. In dieser Zeit konnten keine starken Abweichung zwischen den Zeitstempeln der zwei Sensorknoten erkannt werden. Stichprobenartige Überprüfungen in allen Messreihen bestätigten dieses Ergebnis. Es wurde lediglich eine konstante Abweichung von wenigen Mikrosekunden zwischen den erfassten Zeitstempeln beobachtet, wie die gelb markierten Stellen in Abbildung 4.4 zeigen. Somit erfüllt die Synchronisation die Anforderungen an eine präzise Zeitmessung im einstelligen Millisekundenbereich, die für das Sensorsystem erforderlich ist.

15000	127063235	0	0	127062509
15001	127063735	0	0	127063009
15002	127064235	0	0	127063509
15003	127064735	0	0	127064009
15004	127065235	0	0	127064509
15005	127065735	0	0	127065009
15006	127066235	0	0	127065509
15007	127066735	0	0	127066009
15008	127067235	0	0	127066509
15009	127067735	0	0	127067009
15010	127068235	0	0	127067509

Abbildung 4.4: Ein Ausschnitt aus der Tabelle mit den erfassten Daten des vierten Testdurchlaufs.

4.3 Vergleich der drei Datenaggregationsvarianten Interrupts, Timerinterrupts und Polling

Zur Auswahl der geeigneten Methode für die Sensordatenerfassung wurden drei Varianten verglichen: interruptgesteuertes Auslesen, timerinterruptgesteuertes Auslesen und aktives Polling. Ziel war es, die Methode zu identifizieren, die die zuverlässigste Datenerfassung gewährleistet.

Interrupt-gesteuertes Auslesen

Zuerst wurde überprüft, ob und in welchem Umfang das Auslesen der Sensoren über Interrupts möglich ist. Es wurde zunächst angenommen, dass dies die beste Methode sei, da nur dann ein Auslesen erfolgt, wenn ein neuer Wert anliegt.

Dafür wurde in einem separaten Programm der GPIO-Pin, an den der Interrupt-Pin des Sensors angeschlossen ist, mit der `attachInterrupt`-Funktion für die Erzeugung von Interrupts registriert. Der Sensor befindet sich standardmäßig im High-Zustand, weshalb `FALLING` als Interrupt-Bedingung festgelegt wurde.

Sobald eine Änderung auftritt und ein neuer Sensorwert in dem FIFO-Speicher des Sensors verfügbar ist, löst das Sensormodul einen `DATARDY`-Interrupt aus. Um den anliegenden Interrupt zurückzusetzen, musste entweder das Interrupt Status 1 Register oder das FIFO DATA Register ausgelesen werden [3].

Anfangs wurde dies direkt in der zugehörigen ISR durchgeführt, jedoch führte das Lesen des Registers zu einer unerwarteten langen Verzögerung, wodurch die ISR blockiert wurde und letztendlich Systemabstürze auftraten.

Eine alternative Lösung bestand darin, das Eintreffen eines neuen Interrupts über eine Task-Queue zu signalisieren und die Verarbeitung unabhängig von der ISR in einem separaten Task durchzuführen. Das Auslesen funktionierte bis zu einer Abtastrate von 50 Hz stabil. Mit höheren Abtastraten war jedoch kein fehlerfreies Auslesen mehr möglich, da die Interrupt-Register nicht zurückgesetzt wurden.

Da die direkte Interrupt-Steuerung somit nicht wie gewünscht funktionierte, wurden die beiden weiteren Varianten getestet und sich gegen einen Interrupt-gesteuertes Auslesen entschieden.

Timerinterrupt-gesteuertes Auslesen

Eine weitere Variante war das Auslesen durch einen Timerinterrupt. Die Idee bestand darin, eine Mischung aus Polling und Interrupt-Steuerung zu realisieren, indem nach Ablauf eines definierten Intervalls ein Interrupt ausgelöst wird und der Sensor unmittelbar ausgelesen wird. Dadurch sollte ein kontinuierliches Abfragen (Polling) vermieden werden.

Allerdings traten auch bei dieser Methode ähnliche Probleme wie bei der vorherigen Interrupt-Variante auf, sodass die Sensordaten ebenfalls in einem separaten Task verarbeitet werden mussten. Das Auslesen erfolgte ebenfalls über eine Auftrags-Queue. Dabei zeigte sich, dass die Interrupts zwar wie erwartet ausgelöst wurden, die Sensoren jedoch nicht immer genau zur vorgesehenen Zeit neue Daten bereitstellten. Die Ursache lag in geringfügigen Abweichungen von der erwarteten Abtastfrequenz. So betrug die tatsächliche Abtastrate des Sensorknotens 1 in einigen Durchläufen nur 988 Hz anstelle der erwarteten 1000 Hz.

Der Timerinterrupt-Ansatz wies zwei wesentliche Probleme auf: 1. Die erfasste Datenmenge innerhalb eines bestimmten Zeitraums war unzureichend. 2. Einige Aufträge in der Auftrags-Queue wurden erst nach Ablauf der vorgesehenen Zeit ausgeführt, da das Auslesen der Sensoren zu lange dauerte. Letzteres Problem führte zu Systemabstürzen, da gemeinsame Ressourcen nicht rechtzeitig freigegeben wurden, wodurch ein Watchdog-Timer-Fehler durch den ESP32C6 ausgelöst wurde.

Aktives Polling

Die letzte getestete Variante war das aktive Polling in einem Task. Dabei lief der entsprechende Task kontinuierlich in einer Schleife und wurde entweder durch das Erreichen einer bestimmten Anzahl an Sensordaten oder durch den Ablauf einer definierten Zeitspanne (gesteuert durch einen Timerinterrupt) gestoppt.

Ein Vorteil dieser Methode ist die höhere Stabilität, da der Task nur so lange Daten sammelt, bis die Abbruchbedingung erfüllt ist. Dadurch wird verhindert, dass gleichzeitig andere Tasks auf die gemeinsamen Ressourcen zugreifen, was potenzielle Abstürze vermeidet.

Ein weiterer Vorteil ist, dass exakt die Sensordaten erfasst werden, die innerhalb eines bestimmten Zeitraums von dem Sensormodul bereitgestellt werden. Ein wesentlicher Nachteil ist jedoch die hohe Systembelastung durch die kontinuierliche Abfrage-Schleife.

Obwohl Polling eine höhere Systembelastung verursacht, wurde diese Variante aufgrund ihrer Stabilität und der verlässlichen Erfassung der Sensordaten bevorzugt. Die anderen Methoden führten zu unvorhersehbaren Verzögerungen oder Systemabstürzen, was eine präzise Datenerfassung erschwerte.

4.4 Parameter für die PPG-Sensormodule

Von den zuvor beschriebenen Abfrageansätzen wurde die Polling-Variante eingesetzt und das GY-MAX30102-Sensormodul mit verschiedenen Parametern getestet. Wesentliche Einstellungen wie LED-Strom, Pulsweite, ADC-Bereich und Abtastrate beeinflussen die Signalqualität entscheidend. In allen Tests wurde keine Mittelwertbildung im Sensormodul vorgenommen (`sampleAveraging = 1`), der LED-Modus auf SpO2 (`ledMode = 2`) belassen und die Abtastrate wurde auf 100 Hz eingestellt, mit Ausnahme der Versuche zur Untersuchung der Abtastrate. Die Pulswellen wurden während der Messung sowohl am linken als auch am rechten Zeigefingers eines jungen, männlichen Probanden erfasst.

4.4.1 Einfluss der Pulsweite auf die Signale

Zunächst wurde untersucht in wieweit die Pulsweite die Intensität und Form der erfassten Signale beeinflusst. Um diesen Einfluss zu analysieren, wurden Messungen mit Pulsweiten von 69 μs und 411 μs durchgeführt (Tabelle 4.2).

Die folgenden Parameter wurden untersucht:

Tabelle 4.2: Testparameter für unterschiedliche Pulsweiten.

LED-Strom	Abtastrate	Pulsweite	ADC-Auflösung
50 mA	100 Hz	69 μs	16384
50 mA	100 Hz	411 μs	16384

Die Ergebnisse in Abbildung 4.5 und 4.6 zeigen, dass bei einer Pulsweite von 411 μs die Signalspitzen ausgeprägter sind. Eine Pulsweite von 69 μs liefert ebenfalls erkennbare Signale, jedoch können Spitzen und weitere Merkmale weniger präzise bestimmt werden. Der Nachteil einer höheren Pulsweite liegt im erhöhten Stromverbrauch [57].

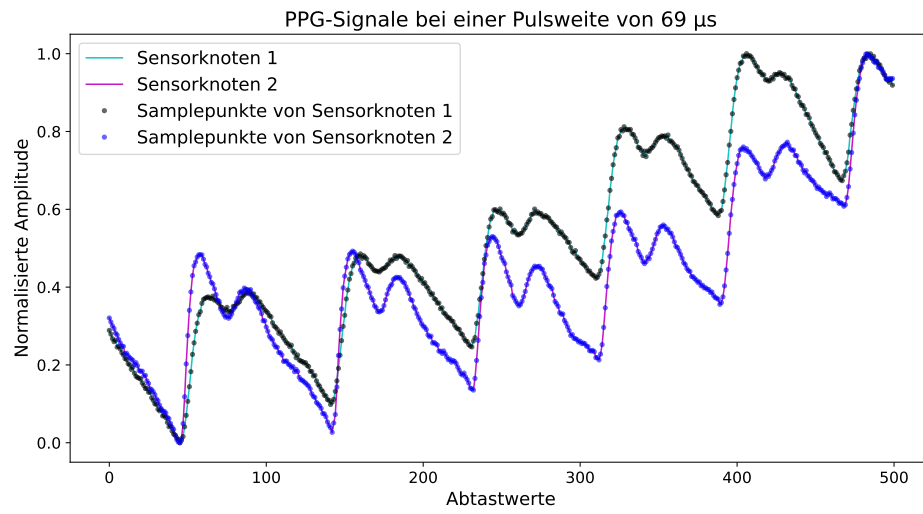


Abbildung 4.5: Ergebnis mit einer Pulsweite von 69 μs .

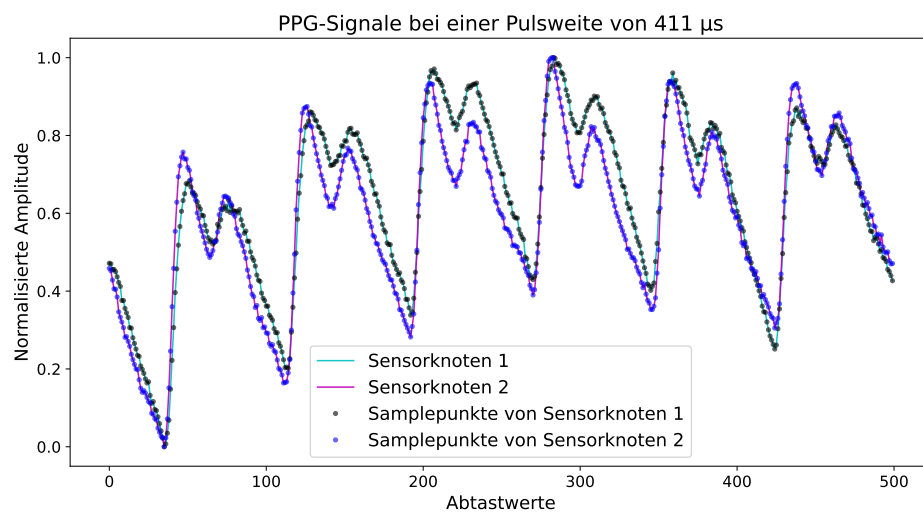


Abbildung 4.6: Ergebnis mit einer Pulsweite von 411 μs .

4.4.2 Einfluss des ADC-Bereichs auf die Signale

Der ADC-Bereich wurde in drei Stufen untersucht (Tabelle 4.3):

Tabelle 4.3: Testparameter für unterschiedliche ADC-Bereiche.

LED-Strom	Abtastrate	Pulsweite	ADC-Auflösung
50 mA	100 Hz	411 μ s	16384
50 mA	100 Hz	411 μ s	8192
50 mA	100 Hz	411 μ s	2048

Bei einer Reduzierung des ADC-Bereichs unter 16384 nA wurden keine Pulswellen mehr erkannt. Daher wurde der maximale Auflösungsbereich von 16384 nA beibehalten.

4.4.3 Einfluss des LED-Stroms auf die Signale

Um den Einfluss der LED-Stärke zu untersuchen wurden folgende zwei unterschiedliche Stromstärken unter den angegebenen Parametern untersucht (Tabelle 4.4):

Tabelle 4.4: Testparameter für unterschiedliche LED-Stromstärken.

LED-Strom (%)	LED-Strom	Abtastrate	Pulsweite	ADC-Auflösung
50 %	25 mA	100 Hz	411 μ s	16384
75 %	37,5 mA	100 Hz	411 μ s	16384

Die Messreihen in Abbildung 4.7 und 4.8 zeigen, dass ein LED-Strom von 50 mA erforderlich ist, um deutliche Pulswellen zu erfassen. Bei niedrigeren Strömen von 25 mA wurden in einigen Fällen nur Rauschen oder keine Pulswellen erkannt. Aus diesen Gründen muss für das Sensorsystem ein maximaler LED-Strom von 50 mA verwendet werden.

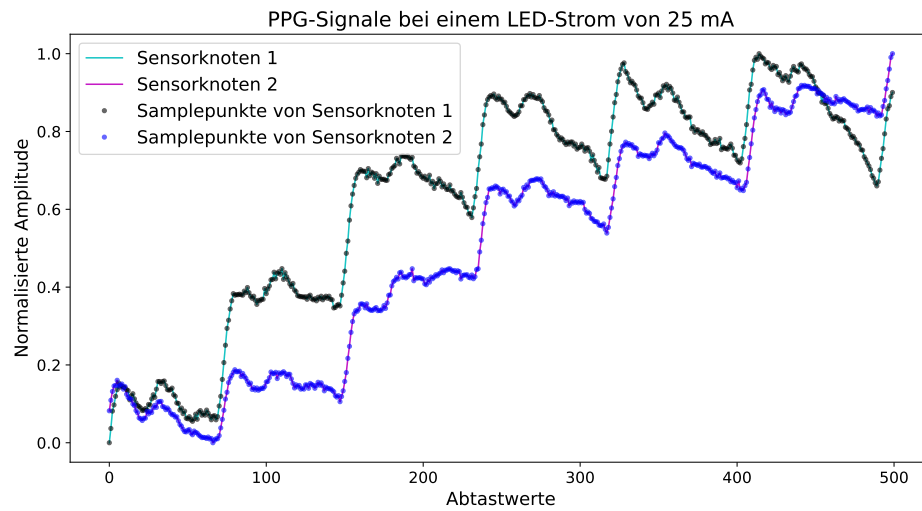


Abbildung 4.7: Ergebnis mit einem Strom von 25 mA (50 %).

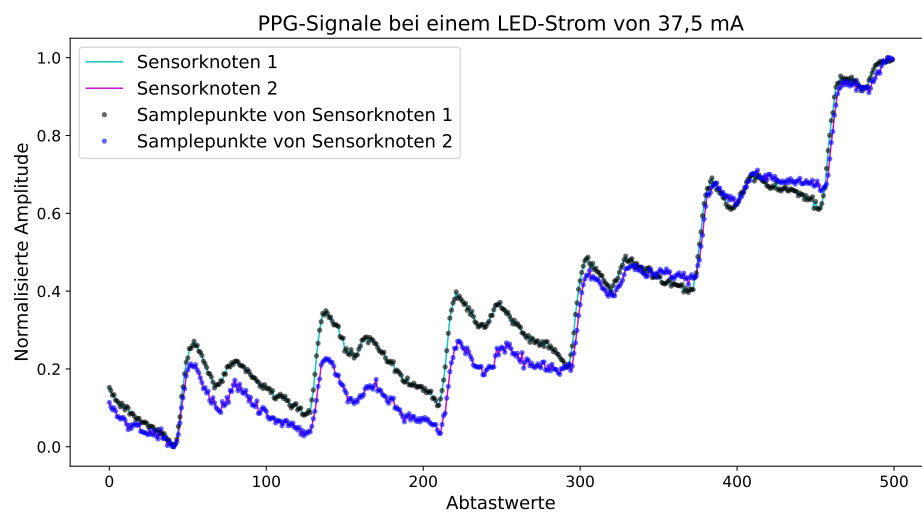


Abbildung 4.8: Ergebnis mit einem Strom von 37,5 mA (75 %).

4.4.4 Einfluss der Abtastrate auf die Signale

In diesem Unterkapitel wird der Einfluss unterschiedlicher Abtastraten gesondert betrachtet. Die Abtastrate hat einen entscheidenden Einfluss auf die durch die Signalverarbeitung resultierende Pulswellenlaufzeiten und letztendlich auf die Pulswellengeschwindigkeiten.

Ein Faktor der das Ergebnis der Pulswellenlaufzeit maßgeblich beeinflusst ist der Zeitstempel der für einen Datenpunkt festgehalten wird, da zwischen zwei Pulswellen der zeitliche Abstand für die Pulswellenlaufzeiten bestimmt werden muss.

Die Abtastrate beeinflusst diesen Abstand unmittelbar, wird beispielsweise eine Abtastrate von 100 Hz angewendet, so ist der zeitlich ideale Abstand zwischen den Datenpunkten exakt 10 ms groß. Dies wäre im Bezug zu der in der Tabelle 3.1 dargestellten Werte für die Pulswellenlaufzeiten nicht in dem gewünschten Bereich und wäre zu ungenau.

Aus diesem Grund wurde angenommen das eine höhere Abtastrate zu präziseren Ergebnissen führt. Diese Annahme wurde in den Folgenden Versuchen untersucht und die Signale mit der im Kapitel 3.5 vorgestellten Signalverarbeitungskette ausgewertet.

Durchführung

Alle Sensorparameter wurden auf die maximalen Werte eingestellt, der LED-Modus 2 beibehalten und die Mittelwertbildung in den Sensormodulen deaktiviert. Die Messungen erfolgten, wie zuvor, mit einem Probanden an beiden Zeigefingern. Untersucht wurden folgende Parameter (die Pulsweite wurde dabei entsprechend dem MAX30102-Sensormodul Datenblatt angepasst) [3]:

Tabelle 4.5: Testparameter für unterschiedliche Abtastraten.

Abtastrate	Pulsweite
50 Hz	411 μ s
100 Hz	411 μ s
200 Hz	411 μ s
400 Hz	411 μ s
800 Hz	215 μ s
1000 Hz	118 μ s
1600 Hz	69 μ s

Ergebnisse

Bereits bei einer Abtastrate von 50 Hz konnten die PPG-Signale rekonstruiert werden, wie der Abbildung 4.9 zu entnehmen ist. Zudem wurde über mehrere Durchläufe festgestellt, dass externe Einflüsse, wie der Fingerdruck auf den Sensor oder Umgebungslicht, die Signalqualität stark beeinflussen, welcher in Abbildung 4.10 betrachtet werden kann. Neben diesen Erkenntnissen wurde festgestellt, dass mit einer Erhöhung der Abtastrate entsprechend eine längere Übertragungszeit einher geht.

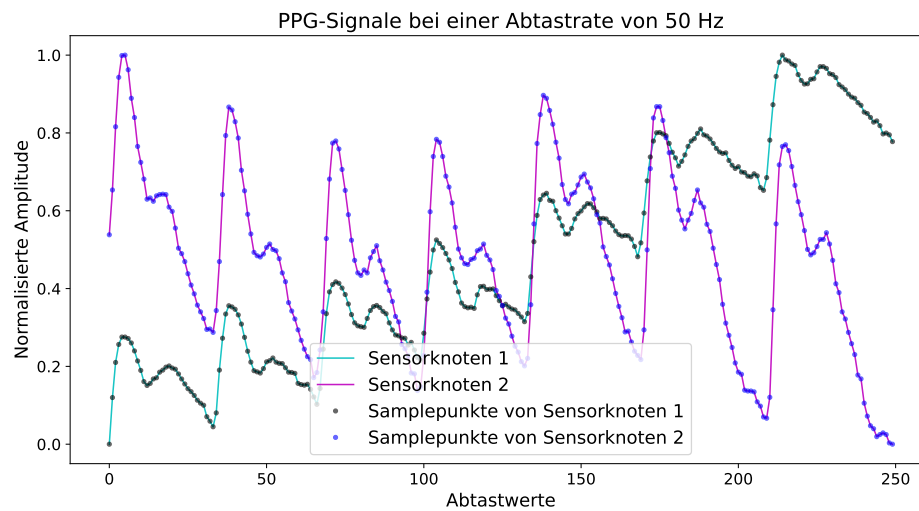


Abbildung 4.9: Rohdaten bei 50 Hz.

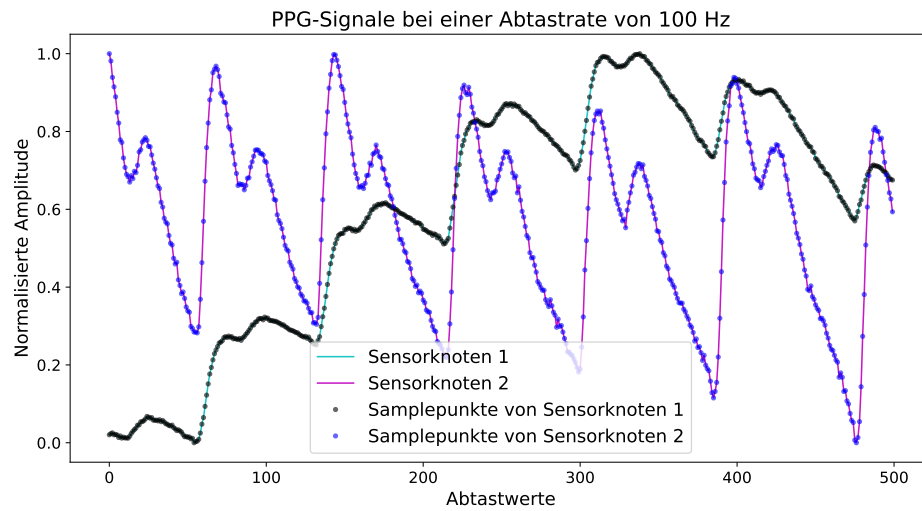


Abbildung 4.10: Rohdaten bei 100 Hz.

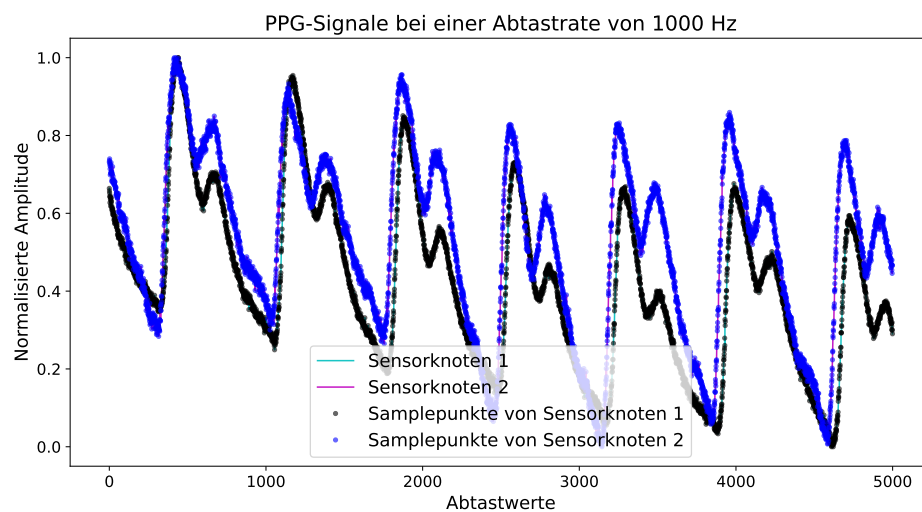


Abbildung 4.11: Rohdaten bei 1000 Hz.

Neben der Betrachtung der Rohdaten wurde der Einfluss der Abtastrate auf die durch die Signalverarbeitungskette bestimmten Pulswellengeschwindigkeiten untersucht. Dazu wurden Messungen mit Abtastraten von 100 Hz und 1000 Hz durchgeführt und miteinander verglichen.

Alle Signale wurden in die Kategorien sehr gut, gut, okay, schlecht und sehr schlecht eingeteilt. Die Einteilung basierte auf einer individuellen Betrachtung jeder einzelnen Messreihe. Eine Messreihe, bei der keine erkennbaren Pulswellen und sehr starke Schwankungen auftraten, wurde der Kategorie „schlecht“ oder „sehr schlecht“ zugeordnet. Eine Messreihe, bei der Pulswellen erkennbar waren, jedoch starke Schwankungen und Rauschen auftraten, wurde der Kategorie „okay“ zugeordnet. Messreihen, bei denen die Pulswellen deutlich erkennbar waren und nur sehr leichte Schwankungen oder Rauschen auftraten, wurden in die Kategorien „gut“ oder „sehr gut“ eingeteilt. Für den Vergleich wurden ausschließlich Signale der Kategorien „sehr gut“ und „gut“ berücksichtigt.

Als Vergleichsparameter wurden zusätzlich die Korrelation zwischen Roh- und gefilterten Sensordaten, das Signal-Rausch-Verhältnis, die Mittelwerte sowie die Standardabweichungen der Pulswellenlaufzeiten und Pulswellengeschwindigkeiten miteinbezogen.

Die Ergebnisse zeigen, dass eine geringe Abtastrate die Erwartungen nicht erfüllt. Bei 100 Hz weist die Pulswellengeschwindigkeit eine hohe Variabilität auf, während die Standardabweichungen bei 1000 Hz für alle Merkmalspunkte deutlich geringer sind. Zudem konnte festgestellt werden, dass die bei 1000 Hz ermittelten Pulswellengeschwindigkeiten deutlich näher am erwarteten Bereich für junge Probanden (4–7 m/s) liegen.

Daraus lässt sich schließen, dass eine höhere Abtastrate zu kleineren Abtastintervallen führt, wodurch die Auflösung verbessert wird (bei 1000 Hz in 1-ms-Intervallen). Daher erweist sich eine höhere Abtastrate von 1000 Hz als besser geeignet.

Allerdings geht eine höhere Abtastrate mit einem wesentlichen Nachteil einher, der signifikanten Zunahme der Datenmenge, wodurch die Übertragungszeiten und der Stromverbrauch erhöht werden. Aus diesem Grund wurde zusätzlich der Einfluss von Resampling-Methoden auf Messreihen mit 100 Hz untersucht.

4.5 Signalverarbeitungsparameter

4.5.1 Untersuchung von Resampling

Zur Untersuchung des Einflusses von Resampling-Methoden wurden nach einer Recherche eine quadratische und eine kubische Interpolation auf ausgewählte Sensordaten angewendet und die Ergebnisse miteinander verglichen [9].

Die Interpolationsvarianten wurden sowohl auf Sensordaten mit einer niedrigen Abtastrate von 50 Hz und 100 Hz als auch auf die herunter abgetasteten Sensordaten mit einer ursprünglichen Abtastrate von 1000 Hz angewendet.

Ergebnisse

Der Versuch zeigt anhand der Abbildung 4.12, dass beide Interpolationsmethoden grundsätzlich geeignet sind. Es fiel jedoch auf, dass die quadratische Interpolation weniger zur Überszillation zwischen den Datenpunkten neigt als die kubische Interpolation. Zudem ist die quadratische Interpolation aufgrund ihrer geringeren Rechenlast der kubischen Interpolation vorzuziehen [9].

Das Herunter- und anschließende Hochsamplen der ursprünglich mit 1000 Hz erfassten Sensordatenreihe hat zudem keine gravierenden Einfluss auf die Pulswellenlaufzeiten und Pulswellengeschwindigkeiten. Aufgrund dieser Ergebnisse kann das System mit einer reduzierten Abtastrate von 50 Hz oder 100 Hz betrieben werden.

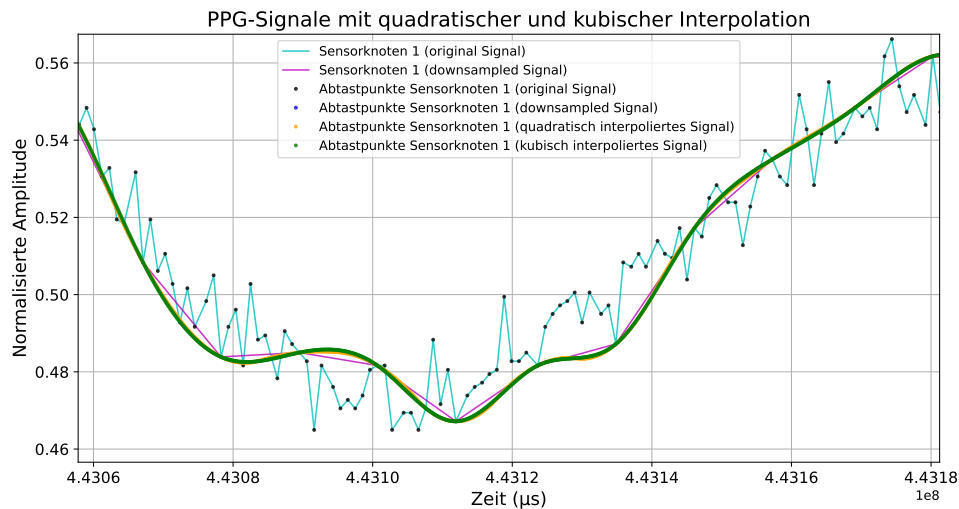


Abbildung 4.12: Ein Ausschnitt des Vergleichs zwischen quadratischer und kubischer Interpolation am Übergang von der systolischen zur diastolischen Phase einer Pulswelle (diastolische Kerbe).

4.5.2 Untersuchung von Filtern

Aufgrund der erforderlichen hohen zeitlichen Genauigkeit ist die Wahl geeigneter Filter entscheidend. Die Auswahl des richtigen Filters ist wichtig, um einen hohen Detailgrad zu erhalten, gleichzeitig jedoch auch Störfaktoren in den Signalen effektiv zu reduzieren.

Es wurden folgende Filter miteinander verglichen:

- IIR-Filter: Butterworth 2. Ordnung, Bessel 2. Ordnung, Elliptisch 2. Ordnung, Chebyshev I 2. Ordnung, Chebyshev II 2. Ordnung
- FIR-Filter: Blackman, Hann, Hamming
- Glättungsfilter: Savitzky-Golay-Filter

Dazu wurde ein PPG-Signal simuliert (mithilfe der neurokit2-Bibliothek) und auf dieses hochfrequentes Rauschen hinzugefügt. Die Ergebnisse wurden mit dem originalen PPG-Signal als Referenz graphisch verglichen. Die Grenzfrequenz wurde auf 10 Hz eingestellt.

Ergebnisse

Die Ergebnisse des Versuchs zeigen, dass IIR-Filter eine starke Filterung erreichen. Zum Chebychev II Filter kann festgestellt werden, dass dieses das Signal am stärksten verschiebt, was mit vorherigen Arbeiten übereinstimmt [35]. Der Butterworth-Filter 2. Ordnung verschiebt das Signal am wenigsten. Alle anderen getesteten Filter erreichen eine ähnliche Filterleistung und verschieben das Signal nur minimal. Ein Ausschnitt dieses Versuchs ist in der Abbildung 4.13 zu sehen.

Zum Savitzky-Golay-Glättungsfilter lässt sich sagen, dass dieser eine gute Glättung erzielt und dennoch relevante Merkmale erhalten bleiben. Es wurde tendenziell ein großes Fenster und ein Polynom niedrigen Grades für den Savitzky-Golay-Filter gewählt.

Für die Probandentests wurden abschließend sowohl der Savitzky-Golay-Filter als auch der Butterworth-Filter 2. Ordnung eingesetzt.

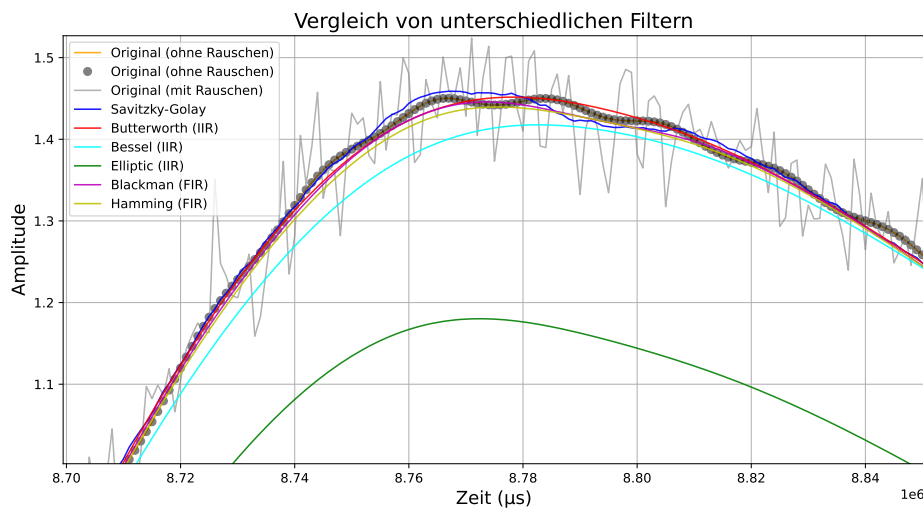


Abbildung 4.13: Ein exemplarischer Ausschnitt des Vergleichs verschiedener Filtern am systolischen Maximum einer Pulswelle.

4.6 Betriebsdauer des Sensorsystems

In diesem Versuch wurde die Betriebsdauer des Sensorsystems überprüft. Dazu wurden die Sensorknoten vollständig aufgeladen. Der Ladezustand wird über die Verbindung eines USB-Typ-C-Kabels an einen PC durch die auf dem Xiao ESP32C6 verbaute LED signalisiert. Während des Ladevorgangs blinkt die LED, ist der Akku vollständig aufgeladen, wird sie deaktiviert.

Ergebnisse

Die zwei Sensorknoten konnten mit einem vollständig aufgeladenen Akku eine Betriebszeit von über 10 Stunden erreichen. Damit erreichen die Sensorknoten mehr als die geforderte Mindestbetriebszeit von 1,5 Stunden.

Neben diesem Test wurden zu Beginn der Entwicklung der Stromverbrauch mithilfe eines Oszilloskops gemessen. Dabei wurde ein durchschnittlicher Verbrauch von etwa 80 mA bestimmt.

Weitere Tests zum Energieverbrauch während der Sende- und Empfangsvorgänge über Bluetooth Low Energy wurden nicht durchgeführt. Ebenso wurde kein Zusammenhang zwischen dem Akkustand und der Betriebsleistung untersucht, beispielsweise im Hinblick auf die ausreichende Versorgung der Komponenten wie der LEDs.

5 Probandenmessungen

Abschließend wurde das System mit verschiedenen Probanden getestet. Durch Messungen an Probanden unterschiedlichen Alters und Geschlechts konnten mehrere Messreihen erhoben und die daraus bestimmten Pulswellenlaufzeiten sowie Pulswellengeschwindigkeiten miteinander verglichen werden.

Ziel dieser Untersuchungen war es zu überprüfen, inwieweit das System die relevanten Indikatoren zuverlässig bestimmen kann und wie die ermittelten Werte im Vergleich zu den Erwartungswerten aus vorherigen Studien stehen.

5.1 Durchführung

Für die Messung wurden die Sensorparameter, bis auf die Abtastrate, auf das Maximum eingestellt. Die Abtastrate wurde auf 100 Hz eingestellt. Jede Messung hatte eine ungefähre Dauer von 5 Sekunden.

Die Messung wurde nicht zeitgesteuert beendet, sondern durch das Erreichen einer maximalen Anzahl an Datenpunkten. Bei einer Abtastrate von 100 Hz ergab dies 500 erwartete Datenpunkte. Die Zeitstempel für die Datenpunkte wurden mit der Funktion `calculateRemoteTimestamp` erfasst und direkt für jeden erfassten Datenpunkt gesichert.

In allen Messungen wurden die Probanden in einer sitzenden Position getestet. Insgesamt wurden Tests mit drei Probanden unterschiedlicher Altersgruppen durchgeführt: ein junger Proband von 24 Jahren (Altersgruppe 1), eine Probandin von 58 Jahren (Altersgruppe 2) und eine Probandin von 78 Jahren (Altersgruppe 3). Um die Funktionalität des Systems zu überprüfen und die Ergebnisse mit anderen Studien zu vergleichen, wurden Pulswellen an unterschiedlichen Körperstellen erfasst. Es wurden Messungen zwischen den zwei Zeigefingern, dem rechten Zeigefinger und dem rechten Zeh, sowie an den unteren und oberen Handgelenken durchgeführt. Die untersuchten Messpositionen sind in Abbildung 5.1 dargestellt.



Abbildung 5.1: Abgebildet sind die vier Messpositionen: oben links Zeigefinger zu Zeigefinger, oben rechts rechter Zeigefinger zu rechtem Zeh, unten links die oberen Seiten der Handgelenke, unten rechts die unteren Seiten der Handgelenke.

Zusätzlich wurde jede Messreihe mit einer individuellen Probanden-ID versehen, die in den Dateinamen integriert wurde, um eine eindeutige Zuordnung zu ermöglichen. In einigen Messreihen wurde zudem der Puls mit einer Samsung Galaxy Watch Active 2 aufgezeichnet, um zusätzlich die Qualität der Messreihen zu überprüfen.

Die Sensordaten wurden mit der im Kapitel 3.5 vorgestellten Signalverarbeitungskette verarbeitet. Dazu wurden alle Messreihen mit einem Butterworth-Hochpassfilter mit einer Grenzfrequenz von 0,5 Hz sowie einem Savitzky-Golay-Filter mit einer Fenstergröße von 91 und einem polynomialen Grad von 6 verarbeitet.

In die Berechnung der Pulswellengeschwindigkeit wurden ausschließlich Werte innerhalb eines realistischen Bereichs für die jeweilige Altersgruppe berücksichtigt. Für die ersten

beiden Probanden wurde ein Bereich von 4 bis 12 m/s gewählt, während für die letzte Probandin ein Bereich von 6 bis 15 m/s festgelegt wurde [1, 15, 12].

Die Ergebnisse der Pulswellengeschwindigkeit setzten sich aus den Maxima der Pulswellen (pks), den Tangenten-Intersektionspunkten (tip) sowie den Maxima der ersten Ableitung (m1d) zusammen.

5.2 Ergebnisse

Mit dem entwickelten System konnten Pulswellenlaufzeiten und Pulswellengeschwindigkeiten erfasst und bestimmt werden. Der Vergleich zwischen Probanden unterschiedlichen Alters zeigte deutliche Unterschiede in den Ergebnissen der Pulswellenlaufzeiten und Pulswellengeschwindigkeiten.

Allerdings zeigten sich deutliche Schwankungen zwischen den Messreihen eines einzelnen Probanden. In Messreihen mit sehr guter Signalqualität, in denen die Pulswellen klar erkennbar sind und die zu vergleichenden Pulswellen beider Messreihen eine hohe Ähnlichkeit aufweisen, entsprechen die Pulswellenlaufzeiten und Pulswellengeschwindigkeiten tendenziell den Erwartungen für die Altersklasse [1, 15, 12].

Dagegen führten große Schwankungen, starkes Rauschen und weitere Störfaktoren häufig zu abweichenden Ergebnissen. In vielen Fällen war dies auf fehlerhaft erkannte Merkmalspunkte oder deutliche Ausreißer in der Merkmalerkennung zurückzuführen.

Einordnung der Messpositionen

Die Ergebnisse an unterschiedlichen Messpositionen zeigen, dass die Messungen zwischen Zeigefinger und Zeigefinger sowie zwischen dem rechten Zeigefinger und den rechten großen Zeh gut ausgeprägte Pulswellen lieferten. Im Gegensatz dazu führten die Messungen an der oberen und unteren Seite der Handgelenke tendenziell zu stark verrauschten Ergebnissen. Besonders an der oberen Seite der Handgelenke war das Rauschen häufig am stärksten ausgeprägt. Dies deckt sich mit den Ergebnissen vorheriger Studien [24].

Als Hauptstörfaktoren wurden dabei Bewegungen, eine fehlerhafte Platzierung der Sensoren an den entsprechenden Körperstellen und Umgebungslicht identifiziert. Der Einfluss der Raumtemperatur wurde nicht untersucht.

Die Abbildung 5.2 zeigt exemplarische zwei durchgeführte Messungen einmal eine zwischen Zeigefinger und dem großen Zeh und eine Aufnahme auf der unteren Seite der Handgelenke der Probandin 1. In der ersten Grafik sind gut ausgeprägte Pulswellen zu erkennen. In der zweiten Grafik sind Pulswellen erkennbar, jedoch in Verbindung mit einer sehr starken Schwankung, wie sie häufig in den Handgelenk-Messungen aufgetreten ist.

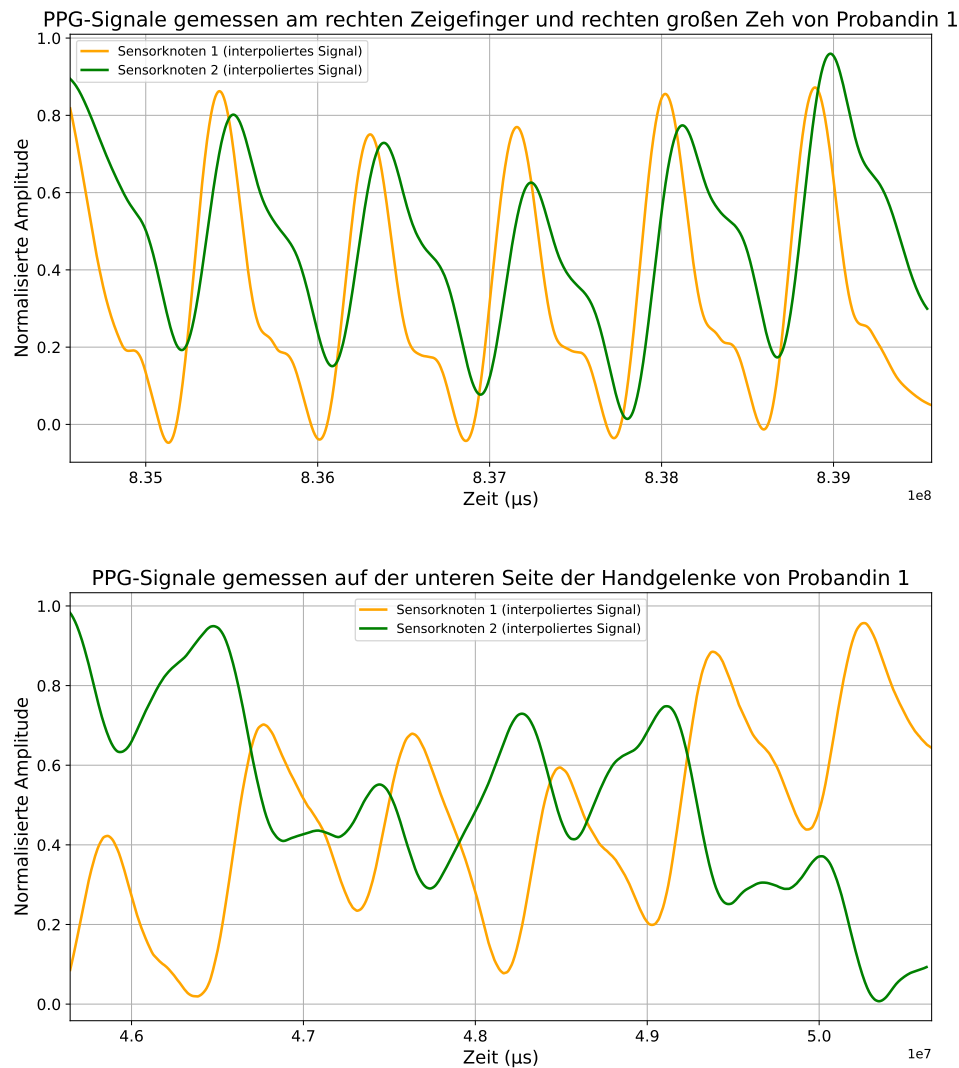


Abbildung 5.2: Zwei exemplarische Messreihen: eine für die Finger-Zeh-Messung und eine für die Messung auf der unteren Seite der Handgelenke.

Da keine Forschungsarbeiten zur Messung der Pulswellenlaufzeit oder der Pulswellengeschwindigkeit für die Positionen Zeigefinger-Zeigefinger und die zwei Seiten der Handgelenke als Referenz gefunden werden konnte, waren die Ergebnisse dieser Messpositionen schwieriger zu überprüfen.

Signalqualität und Einfluss auf die Messergebnisse

Ein eindeutiger Zusammenhang zwischen der Signal-to-Noise-Ratio und der Erkennung von Merkmalspunkten konnte nicht eindeutig nachgewiesen werden. Allerdings zeigte sich, dass Pulswellen mit einer Signal-to-Noise-Ratio von über 10 dB tendenziell gut verarbeitet wurden.

Hinsichtlich der Kreuzkorrelation ließ sich feststellen, dass eine Korrelation von meist über 80 % auf eine gute Signalqualität der Pulswellen hinweist. Eine hohe Korrelation zwischen den Pulswellen führte zudem zu Ergebnissen, die weitgehend den Erwartungen entsprachen. Oft wiesen diese Signale eine sehr ähnliche Morphologie auf, wie es in der zuvor dargestellten Abbildung 5.2 zu sehen ist.

Einordnung der Merkmalspunkte

Bei den unterschiedlichen Merkmalspunkten zeigte sich, dass die Punkte tip, m1d, a2d sowie der bestimmte Punkt durch die Diastol-Patching-Methode (pcc) überwiegend den erwarteten Werten für die Pulswellengeschwindigkeit entsprechen. Dabei fiel auf, dass tip und pcc tendenziell ähnliche Werte lieferten.

In Signalen mit guter bis sehr guter Signalqualität (Korrelation von meist über 80 % und einer SNR von über 10 dB) wurden häufig auch die Maxima (pks) und Minima (ons) zuverlässig erkannt.

Darüber hinaus zeigte sich, dass die Ergebnisse ohne Ableitung stark von Änderungen der Signalform beeinflusst wurden. Im Gegensatz dazu blieben die Merkmalspunkte der ersten und zweiten Ableitung, wie m1d oder a2d, weitgehend stabil.

Die Merkmalspunkte in den Ableitungen wurden jedoch stark von hochfrequentem Rauschen beeinflusst, sodass bei stark verrauschten Signalen Fehler in der Merkmalerkennung auftraten.

Daher wurden die Merkmalspunkte Maxima, Tangenten-Intersektionspunkte sowie die Maxima der ersten Ableitung in die Berechnung der durchschnittlichen Finger-Zeh-Pulswellengeschwindigkeit und der Finger-Finger-Pulswellengeschwindigkeit der Proban-

den einbezogen. Die Bestimmung anderer Pulswellengeschwindigkeiten musste aufgrund der starken Schwankung in den Pulswellen gesondert vorgenommen werden.

Einordnung der individuellen Probandenergebnisse

Das konsistenteste Ergebnis wurde bei Proband 0 erzielt, der mit einem Alter von 24 Jahren eine Pulswellengeschwindigkeit von 6,2 m/s aufwies. Dieses Ergebnis liegt innerhalb des erwarteten Bereichs von 4 bis 7 m/s für diese Altersgruppe [1, 15, 12].

Das Ergebnis weist zudem mit einer Standardabweichung von 0,16 m/s eine geringe Schwankung auf. Andere erfasste Messreihen haben tendenziell höhere Standardabweichungen von über 0,2 m/s. Außerdem lagen die Pulswellengeschwindigkeiten bei diesen Messungen meist in einem höheren Bereich als erwartet.

Zum Vergleich der altersabhängigen Unterschiede wurde ebenfalls die Pulswellengeschwindigkeit einer 58-jährigen Probandin untersucht. Ihr Durchschnittswert betrug 8,41 m/s und liegt damit im erwarteten Bereich von $8,46 \pm 2,86$ m/s für die Altersgruppe von 50 ± 15 Jahren [1, 15].

Dieser Wert liegt über dem Median für diese Altersgruppe und könnte auf eine Hypertonie (Bluthochdruck) hinweisen [15]. Diese Annahme wird durch die tatsächliche Diagnose der Probandin gestützt.

Auch für eine 78-jährige Probandin wurde eine Pulswellengeschwindigkeit von 10,3 m/s mit einer Standardabweichung von 0,24 m/s im qualitativ besten Ergebnis der Messreihen ermittelt. Dieser Wert liegt ebenfalls innerhalb des erwarteten Bereichs für diese Altersklasse.

Alle analysierten Werte beziehen sich auf die Pulswellengeschwindigkeit zwischen dem rechten Zeigefinger und großen Zeh. Aufgrund des größeren Abstands zwischen diesen beiden Messpunkten kann die Pulswellengeschwindigkeit in diesen Bereich am zuverlässigsten bestimmt werden.

Im Gegensatz dazu erwies sich die Bestimmung der Pulswellengeschwindigkeit zwischen den Zeigefingern als schwieriger, da der Abstand zwischen den Messpunkten deutlich kleiner ist.

Die Abbildung 5.3 zeigt eine Gegenüberstellung der Boxplots für die Messungen zwischen Zeigefinger und Zeigefinger sowie zwischen Finger und großem Zeh. Hier wird die deutlich höhere Varianz in den Messungen zwischen den Zeigefingern sichtbar.

Es ist zu beachten, dass alle Ergebnisse individuell ausgewertet wurden. Derzeit existiert keine standardisierte Methode zur Messung der Pulswellengeschwindigkeit. Während einige Studien lediglich Tangenten-Intersektionspunkte in die Bestimmung einbeziehen, berücksichtigen andere ausschließlich die Maxima oder verwenden alternative Ansätze zur Berechnung der Pulswellengeschwindigkeit. Daher ist in Zukunft die Definition von Standards in diesem Bereich unerlässlich [43].

Die Messungen zwischen den drei Probanden haben gezeigt, dass die Erkennung der Pulswellen bei sehr kleinen zeitlichen Abständen zwischen den Messpunkten besonders schwierig ist, was zu einer hohen Streuung der Ergebnisse führt. Bei den Messungen an den Handgelenken wurden nur selten Werte im erwarteten Bereich erzielt. In den meisten Fällen führten starke Schwankungen oder hochfrequentes Rauschen aufgrund einer fehlerhaften Platzierung der Sensoren zu Störungen, die nicht kompensiert werden konnten. Daher ist es in zukünftigen Arbeiten notwendig, diese Störfaktoren gezielt zu kompensieren, etwa durch den Einsatz zusätzlicher Hardware wie Beschleunigungssensoren, Drucksensoren oder ähnlichem.

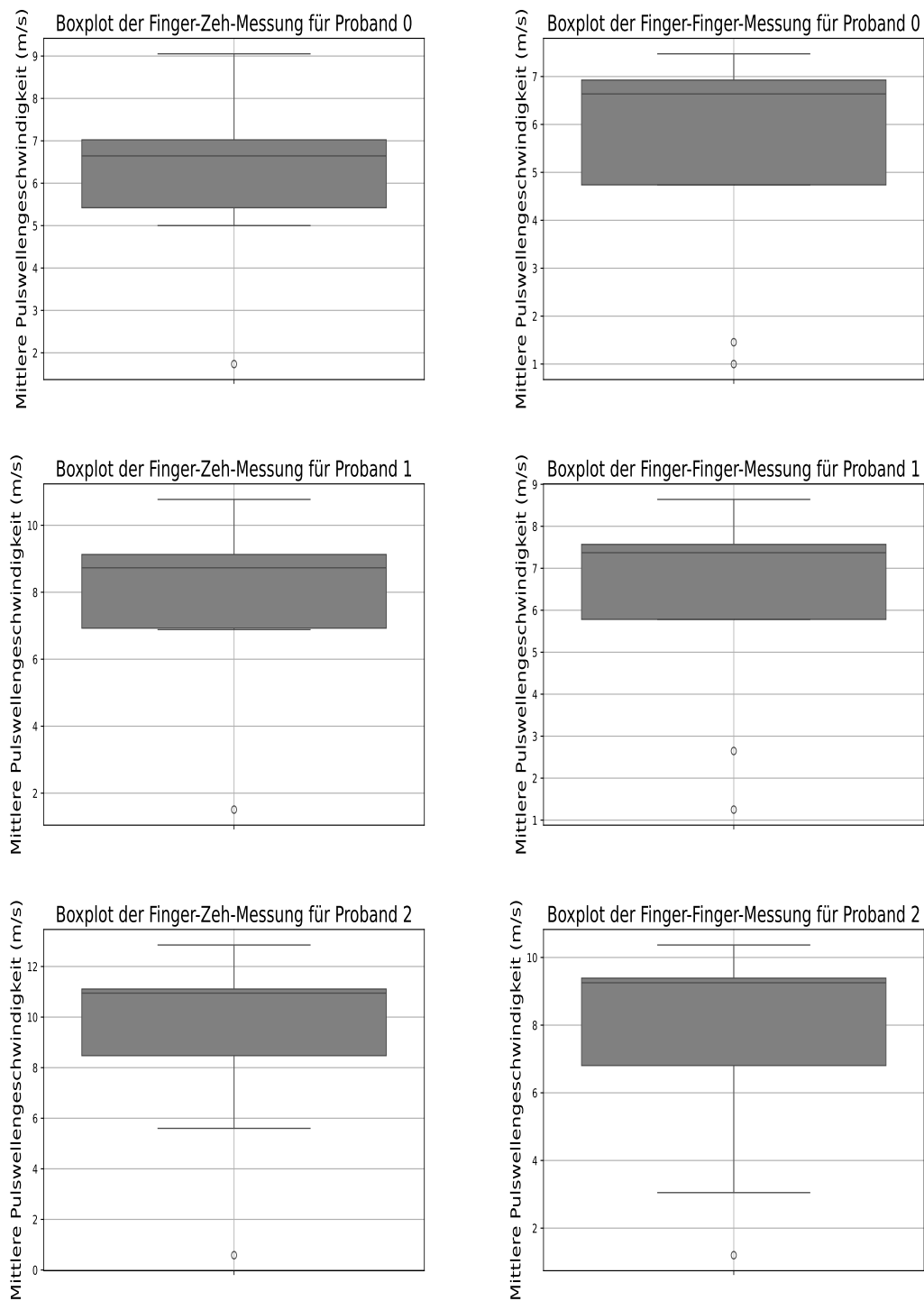


Abbildung 5.3: Boxplots der Durchschnittswerte: Links der Finger-Zeh-Messungen und rechts der Zeigefinger-Zeigefinger-Messungen.

6 Diskussion

In dieser Arbeit wurde die Entwicklung eines drahtlosen PPG-Sensorsystems zur Bestimmung der Pulswellengeschwindigkeit vorgestellt.

Einleitend wurden die theoretischen Grundlagen sowie der aktuelle Stand der Forschung in den Bereichen Pulswellenanalyse, drahtlose Kommunikationskonzepte, Synchronisationsverfahren und Signalverarbeitung erläutert.

Zunächst wurden medizinisch-technische Grundlagen zum Herz-Kreislauf-System, zur arteriellen Gefäßsteifigkeit, zur Pulswellengeschwindigkeit und zur Photoplethysmographie dargestellt. Es folgte eine detaillierte Vorstellung der Kommunikationsprotokolle Bluetooth Low Energy und Thread, die als Standards für die Entwicklung von Wireless Sensor Networks (WSNs) relevant sind.

Daraufhin wurden grundlegende Konzepte der Synchronisation behandelt, darunter die Definition einer Uhr, verschiedene Synchronisationsansätze sowie eine tiefgehende Beschreibung des Synchronisationsprotokolls Timing-Sync Protocol for Sensor Networks (TPSN).

Das Kapitel Grundlagen schloss mit einer Übersicht zu gängigen Signalverarbeitungsmethodiken ab, wobei unter anderem die Tangenten-Intersektionsmethode vorgestellt wurde.

Im darauffolgenden Kapitel Systementwicklung wurde die Entwicklung des Sensorsystems umfassend beschrieben. Dazu wurden zu Beginn das Ziel und die Anforderungen der Arbeit definiert und ein grundlegendes Systemkonzept vorgestellt.

Der Fokus lag dabei auf der Entwicklung eines benutzerfreundlichen, drahtlosen PPG-Sensorsystems, das an den Handgelenken getragen werden kann, um dort die Pulswellen zu erfassen und mit diesen die Pulswellengeschwindigkeit zu bestimmen.

Anschließend wurden die Hardware- und Softwareentwicklung detailliert erläutert. Dazu wurde zunächst die Auswahl der Hardware-Komponenten wie das GY-MAX30102-Sensormodul und das Entwicklungsboard XIAO ESP32C6 dargestellt und begründet.

Zudem wurde die Entwicklung der Softwaremodule für das PPG-Sensormodul und die Kommunikation dargestellt. Hierbei kam das Master-Slave-Pattern zum Einsatz und das Bluetooth Low Energy Protokoll wurde zur Kommunikation eingesetzt. Dazu wurde die Entwicklung der zwei Wrapper-Module BLEServerFacade und BLEClientFacade, die die Protokollspezifikation abstrahieren, beschrieben.

Folgend wurden die Herausforderungen im Zusammenhang mit der Synchronisation bei der Integration dieser Module dargelegt und als Lösung das Timing-Sync Protocol for Sensor Networks implementiert.

Darauf wurde die Signalverarbeitung vorgestellt, beginnend mit einer Übersicht der Signalverarbeitungskette. Die einzelnen Verarbeitungsschritte wurden detailliert beschrieben, insbesondere die implementierten Filter und Verfahren zur Merkmalsextraktion. Dabei wurde insbesondere der Einsatz linearphasiger Filter betont, um die ursprüngliche Signalform möglichst zu erhalten. Zur Merkmalsextraktion wurden die Peak-Detektion mit der Funktion `find_peaks` und `ppg_process` aus der NeuroKit2-Bibliothek, die `fiducial_points`-Funktion sowie ein Algorithmus basierend auf dem Diastole-Patching-Ansatz vorgestellt.

Zur Validierung des Sensorsystems wurden umfangreiche Tests durchgeführt. Dazu wurden verschiedene Experimente konzipiert und vorgestellt. Zu Beginn wurden die Kommunikationslösungen untersucht. Dabei zeigte sich, dass Bluetooth Low Energy in Kombination mit dem Synchronisationsprotokoll TPSN die zeitlichen Anforderungen von mindestens einer Millisekunde mehr als erfüllt.

Anschließend wurden verschiedene Datenaggregationsverfahren (Interrupts, Timerinterrupts, Polling) miteinander verglichen sowie der Einfluss unterschiedlicher Sensorparameter untersucht. Das Ergebnis des Vergleichs zeigte, dass mit der gegebenen Hardware Polling aufgrund seiner höheren Zuverlässigkeit die beste Aggregationsmethode darstellte. Allerdings wurde dabei auch betont, dass Polling nicht die effizienteste Lösung ist und dies in Zukunft optimiert werden sollte.

In weiteren Experimenten wurden verschiedene Sensorparameter analysiert, um deren optimale Auswahl zu begründen. Hierbei hat sich gezeigt, dass niedrige Abtastraten von 50 bis 800 Hz die erforderliche Genauigkeit des Sensorsystems nicht erreichen.

Daher wurden in weiteren Versuchen Parameter für die Signalverarbeitung untersucht, insbesondere Resampling-Methoden und Filterverfahren. Dabei zeigte sich, dass sowohl die quadratische als auch die kubische Interpolation genügen und durch sie die Anforderung an die Genauigkeit erfüllt werden.

Ein weiter Fokus lag auf der Untersuchung unterschiedlicher Filterverfahren. Hierbei erwies sich ein Butterworth-Filter 2. Ordnung als besonders geeignet, da er eine effektive Filterung bei minimaler Verzerrung des Originalsignals ermöglicht. Der Savitzky-Golay-Glättungsfilter hat sich zudem als guter Filter zur Reduzierung von Rauschen und dem gleichzeitigen Erhalt von Signalmerkmalen herausgestellt. Zur Verarbeitung der Signale wurde letztendlich ein Savitzky-Golay-Glättungsfilter angewendet und ein Butterworth-Filter 2. Ordnung zur Reduzierung von niederfrequenten Schwankungen.

In einem letzten Systemtest wurde die Betriebsdauer des Sensorsystems untersucht. Dabei wurde eine Betriebsdauer von über 10 Stunden gemessen, womit das Ziel von 1,5 Stunden deutlich übertroffen wurde.

Das gesamte System wurde abschließend mit drei Probanden getestet. Dabei zeigte sich, dass zahlreiche potenzielle Störfaktoren wie Bewegung, Shuntlicht sowie die unzureichende Befestigung der Sensoren, einen erheblichen Einfluss auf die Ergebnisse haben. Diese Herausforderungen wurden auch durch die vorgestellte Literatur bestätigt.

Somit konnte das ursprüngliche Ziel der Pulswellenerfassung an den Handgelenken aufgrund starker Schwankungen nicht vollständig erreicht werden. Allerdings wurde durch die Finger-Finger-Messungen ein vergleichbarer Messeffekt erzielt, da diese auf demselben Abstand der Schlüsselbeinarterien basieren.

Durch den Vergleich von drei Probanden unterschiedlichen Alters konnten die Pulswellengeschwindigkeiten miteinander verglichen werden. Es hat sich gezeigt, dass die Finger-Zeh-Messungen gut den Erwartungsbereich widerspiegeln, während die Finger-Finger-Messungen ebenfalls plausible Ergebnisse lieferten, jedoch mit einer höheren Variabilität.

Aus den genannten Gründen sollten zukünftige Arbeiten Mechanismen integrieren, die gezielt die genannten Störfaktoren reduzieren. Dies könnte durch den Einsatz von zusätzlichen Sensoren wie Beschleunigungssensoren, Drucksensoren oder Mikrofonen erfolgen.

Entsprechend sollten auch Algorithmen implementiert werden, die diese Störfaktoren gezielt kompensieren, beispielsweise durch Adaptive Noise Cancellation oder durch den vielversprechenden Ansatz des Wavelet-Denoisings [61].

Die Probandenmessungen und die Literaturrecherche zeigten, dass Messungen zwischen den Zeigefingern und Handgelenken bisher wenig untersucht wurden. Aus diesem Grund könnten zukünftige Studien Messungen an diesen Positionen in Kombination mit einem etablierten Referenzsystem, wie beispielsweise einem EKG, durchführen. Darüber hinaus wäre eine weitergehende Untersuchung des vorgestellten Sensorsystems mit einer größeren Anzahl an Probanden und einem Referenzsystem sinnvoll, um die Zuverlässigkeit und Genauigkeit des Systems weiter zu validieren.

Ein weiteres Forschungspotential besteht in der Entwicklung neuer Algorithmen zur Bestimmung der Pulswellengeschwindigkeit, etwa durch die Analyse des Massenschwerpunkts der Pulswelle oder durch ein Verfahren, das die anliegenden Tangenten der Pulswellen miteinander vergleicht, indem es die euklidische Distanz zwischen jedem Punkt der Tangenten berechnet [30].

Eine weitere Optimierung des Sensorsystems bestünde in der Entwicklung eines erweiterten drahtlosen Sensornetzwerks mit mehreren Sensorknoten auf Basis des vorgestellten Thread-Protokolls. In ein solches System könnten beispielsweise intelligente Kopfhörer oder Ringe integriert werden, um mehrere Messpunkte für die Pulswellenerfassung und die Bestimmung der Pulswellengeschwindigkeit zu erhalten [34].

Diese Arbeit hat einen Überblick zu der Entwicklung eines drahtlosen PPG-Sensorsystems gegeben, insbesondere die Herausforderungen bei der Entwicklung eines solchen Systems aufgezeigt und einige Lösungsansätze vorgestellt und implementiert.

Ein zukünftiges Ziel muss es sein, die noch offenen Herausforderungen zu lösen und zuverlässige drahtlose Sensorsysteme für den Alltagsgebrauch zu entwickeln, um arterielle Gefäßkrankheiten frühzeitig zu erkennen und mögliche Folgerisiken zu minimieren.

Literaturverzeichnis

- [1] ALIVON, Maureen u. a.: A novel device for measuring arterial stiffness using finger-toe pulse wave velocity: Validation study of the pOpmètre®. In: *Archives of Cardiovascular Diseases* 108 (2015), Nr. 4, S. 227–234. – URL <https://www.sciencedirect.com/science/article/pii/S1875213615000042>. – ISSN 1875-2136
- [2] ALLEN, John: Photoplethysmography and its application in clinical physiological measurement. 28 (2007), Nr. 3, S. R1. – URL <https://iopscience.iop.org/article/10.1088/0967-3334/28/3/R01/meta>. – Zugriffsdatum: 2024-12-01. – Publisher: IOP Publishing. – ISSN 0967-3334
- [3] ANALOG DEVICES INC.: *MAX30102 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*. Wilmington, MA: , 2018. – URL <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30102.pdf>. – Zugriffsdatum: 2024-12-16. – Datenblatt
- [4] ARDUINO LLC: *Arduino IDE*. Boston, MA: , 2024. – URL <https://www.arduino.cc/en/software>. – Zugriffsdatum: 2024-12-16. – Software-Dokumentation
- [5] BACHLER, M. ; MAYER, C. ; HAMETNER, B. ; WASSERTHEURER, S.: Increasing Stability of Real-Time Pulse Wave Velocity Estimation by Combining Established and New Approaches. In: *2013 8th EUROSIM Congress on Modelling and Simulation*, 2013, S. 47–51
- [6] BAULMANN, Johannes ; NÜRNBERGER, J ; SLANY, J ; SCHMIEDER, Roland ; SCHMIDT-TRUCKSÄSS, Arno ; BAUMGART, Dietrich ; CREMERIUS, P ; HESS, O ; MORTENSEN, Kai ; WEBER, T: Arterielle Gefäßsteifigkeit und Pulswellenanalyse. In: *Deutsche Medizinische Wochenschrift - DEUT MED WOCHENSCHR* 135 (2010), 03

- [7] BLUETOOTH SPECIAL INTEREST GROUP: *The Bluetooth® Low Energy Primer / Bluetooth® Technology website*. 2024. – URL <https://www.bluetooth.com/bluetooth-resources/the-bluetooth-low-energy-primer/>. – Zugriffsdatum: 2025-01-30
- [8] BURAIOLI, Irene ; LENA, Davide ; SANGINARIO, Alessandro ; LEONE, Dario ; MINGRONE, Giulia ; MILAN, Alberto ; DEMARCHI, Danilo: A New Noninvasive System for Clinical Pulse Wave Velocity Assessment: The Athos Device. In: *IEEE Transactions on Biomedical Circuits and Systems* 15 (2021), Nr. 1, S. 133–142
- [9] BÉRES, Szabolcs ; HEJJEL, László: The minimal sampling frequency of the photoplethysmogram for accurate pulse rate variability parameters in healthy volunteers. In: *Biomedical Signal Processing and Control* 68 (2021), S. 102589. – URL <https://www.sciencedirect.com/science/article/pii/S1746809421001865>. – ISSN 1746-8094
- [10] CHARLTON, P. H. ; MAROZAS, V.: Wearable Photoplethysmography Devices. In: KYRIACOU, P. A. (Hrsg.) ; ALLEN, J. (Hrsg.): *Photoplethysmography*. Elsevier, 2021
- [11] CHARLTON, Peter H. ; TEAM, based on work by Elisa Mejía-Mejía the: *Pulse detection and correction from pulsatile signals*. Online resource. June 2022. – URL <https://peterhcharlton.github.io/bsp-book/tutorial/notebooks/pulse-wave-analysis.html#fiducial-point-functions>. – Licensed under CC BY 4.0
- [12] CHEN, Yung-Sheng ; LU, Wan-An ; HSU, Ling-Yen ; KUO, Cheng-Deng: Determinants of hand pulse wave velocity and hand pulse transit time in healthy adults. In: *Scientific Reports* 14 (2024), Nr. 1, S. 10144. – URL <https://doi.org/10.1038/s41598-024-60927-5>. – ISSN 2045-2322
- [13] CHIU, Y.Christopher ; ARAND, Patricia W. ; SHROFF, Sanjeev G. ; FELDMAN, Ted ; CARROLL, John D.: Determination of pulse wave velocities with computerized algorithms. In: *American Heart Journal* 121 (1991), Nr. 5, S. 1460–1470. – URL <https://www.sciencedirect.com/science/article/pii/0002870391901539>. – ISSN 0002-8703
- [14] COURONNÉ, Robert: *Erfassung der Pulswelle am Unterarm*. Vieweg+Teubner Verlag Wiesbaden, 2012. – URL <https://doi.org/10.1007/978-3-8348-2403-5>

- [15] DIAZ, Alejandro ; TRINGLER, Matías ; WRAY, Sandra ; RAMIREZ, Agustín J. ; CABRERA FISCHER, Edmundo I.: The effects of age on pulse wave velocity in untreated hypertension. In: *The Journal of Clinical Hypertension* 20 (2018), Nr. 2, S. 258–265. – URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/jch.13167>
- [16] ELGENDI, Mohamed: On the analysis of fingertip photoplethysmogram signals. In: *Current Cardiology Reviews* 8 (2012), February, Nr. 1, S. 14–25
- [17] ESPRESSIF SYSTEMS: *ESP32-C6 Technical Reference Manual*. Shanghai: , 2024. – URL https://www.espressif.com/sites/default/files/documentation/esp32-c6_technical_reference_manual_en.pdf. – Zugriffsdatum: 2024-12-16. – Technisches Handbuch
- [18] ESPRESSIF SYSTEMS: *GitHub - espressif/arduino-esp32: Arduino core for the ESP32*. 2024. – URL <https://github.com/espressif/arduino-esp32>. – Zugriffsdatum: 2024-12-16. – GitHub Repository
- [19] ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD.: *FreeRTOS (IDF) - ESP32-C6 - — ESP-IDF Programming Guide v5.4 documentation*. 2025. – URL https://docs.espressif.com/projects/esp-idf/en/stable/esp32c6/api-reference/system/freertos_idf.html
- [20] FAN, Zhaopeng ; ZHANG, Gong ; LIAO, Simon: *Pulse Wave Analysis*. IntechOpen, 8 2011. – URL <https://www.intechopen.com/chapters/18367#>
- [21] GADDUM, N. R. ; ALASTRUEY, J. ; BEERBAUM, P. ; CHOWIENCZYK, P. ; SCHAEFFTER, T.: A Technical Assessment of Pulse Wave Velocity Algorithms Applied to Non-invasive Arterial Waveforms. In: *Annals of Biomedical Engineering* 41 (2013), Nr. 12, S. 2617–2629. – URL <https://doi.org/10.1007/s10439-013-0854-y>. – ISSN 1573-9686
- [22] GANERIWAL, Saurabh ; KUMAR, Ram ; SRIVASTAVA, Mani B.: Timing-sync protocol for sensor networks. In: *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. New York, NY, USA : Association for Computing Machinery, 2003 (SenSys '03), S. 138–149. – URL <https://doi.org/10.1145/958491.958508>. – ISBN 1581137079
- [23] GOMEZ, Carles ; OLLER, Joaquim ; PARADELLS, Josep: Overview and evaluation of Bluetooth Low Energy: an Emerging Low-Power Wireless Technology. In: *Sensors*

- 12 (2012), August, Nr. 9, S. 11734–11753. – URL <https://www.mdpi.com/1424-8220/12/9/11734>
- [24] GUPTA, Shresth ; SINGH, Anurag ; SHARMA, Abhishek: Denoising and Analysis of PPG Acquired From Different Body Sites Using Savitzky Golay Filter. In: *TENCON 2022 - 2022 IEEE Region 10 Conference (TENCON)*, 2022, S. 1–4
- [25] HARRIS, Charles R. u. a.: Array programming with NumPy. In: *Nature* 585 (2020), September, Nr. 7825, S. 357–362. – URL <https://doi.org/10.1038/s41586-020-2649-2>
- [26] HOCHEGGER, Mario: *Implementation of an optical pulse wave velocity measurement device*, Graz University of Technology, Bachelor Thesis, 09 2023. – Thesis for: Bachelor, September 2023, Research Gate
- [27] HOFMANN, Julia: *Kardiovaskuläres System*. S. 77–124. In: *Ein Physiologiemodell für Tactical Combat Casualty Care Training in mobilen Serious Games*, Springer Fachmedien Wiesbaden, 2020. – URL http://link.springer.com/10.1007/978-3-658-30202-3_6. – Zugriffsdatum: 2024-11-27. – ISBN 978-3-658-30201-6 978-3-658-30202-3
- [28] HUNTER, J. D.: Matplotlib: A 2D graphics environment. In: *Computing in Science & Engineering* 9 (2007), Nr. 3, S. 90–95
- [29] JIN, Ji ; GENG, Xingguang ; ZHANG, Yitao ; ZHANG, Haiying ; YE, Tianchun: Pulse Wave Analysis Method of Cardiovascular Parameters Extraction for Health Monitoring. In: *International Journal of Environmental Research and Public Health* 20 (2023), Nr. 3. – URL <https://www.mdpi.com/1660-4601/20/3/2597>. – ISSN 1660-4601
- [30] JIN, Wei ; CHOWIENCZYK, Phil ; ALASTRUEY, Jordi: Estimating Pulse Wave Velocity from the Radial Pressure Wave Using Machine Learning Algorithms. In: *PLoS ONE* 16 (2021), Nr. 6, S. e0245026. – URL <https://doi.org/10.1371/journal.pone.0245026>
- [31] KHATTAK, Sohaib Bin A. ; NASRALLA, Moustafa M. ; FARMAN, Haleem ; CHOUDHURY, Nikumani: Performance evaluation of an IEEE 802.15.4-Based thread network for efficient Internet of Things communications in smart cities. In: *Applied Sciences* 13 (2023), Juni, Nr. 13, S. 7745. – URL <https://www.mdpi.com/2076-3417/13/13/7745>

- [32] KLUG, Nathalie E.: *Mobile Pulsoxymetrie*. – URL <https://mediatum.ub.tum.de/1224528>. – Zugriffsdatum: 2024-12-03
- [33] KO, JeongGil ; LU, Chenyang ; SRIVASTAVA, Mani B. ; STANKOVIC, John A. ; TERZIS, Andreas ; WELSH, Matt: Wireless Sensor Networks for Healthcare. In: *Proceedings of the IEEE* 98 (2010), Nr. 11, S. 1947–1960
- [34] KUSCHE, Roman ; DICHTER, Daniel ; GOMEZ, Jean Carlos H. ; THÖLEN, Tobias: Pulse Wave Measurement With In-Ear Headphones While Music Playback. In: *IEEE Sensors Letters* 9 (2025), Nr. 1, S. 1–4
- [35] LAPITAN, Denis G. ; ROGATKIN, Dmitry A. ; MOLCHANOVA, Elizaveta A. ; TARASOV, Andrey P.: Estimation of phase distortions of the photoplethysmographic signal in digital IIR filtering. In: *Scientific Reports* 14 (2024), Mar, Nr. 1, S. 6546. – URL <https://doi.org/10.1038/s41598-024-57297-3>. – ISSN 2045-2322
- [36] LAURENT, Stephane ; COCKCROFT, John ; VAN BORTEL, Luc ; BOUTOUYRIE, Pierre ; GIANNATTASIO, Cristina ; HAYOZ, Daniel ; PANNIER, Bruno ; VLACHOPOULOS, Charalambos ; WILKINSON, Ian ; STRUIJKER-BOUDIER, Harry ; ARTERIES, on behalf of the European Network for Non-invasive Investigation of L.: Expert consensus document on arterial stiffness: methodological issues and clinical applications. 27 (2006), Nr. 21, S. 2588–2605. – URL <https://dx.doi.org/10.1093/eurheartj/ehl254>. – Zugriffsdatum: 2024-11-27. – Publisher: Oxford Academic. – ISSN 0195-668X
- [37] LIANG, Yongbo ; ELGENDI, Mohamed ; CHEN, Zhencheng ; WARD, Rabab: An optimal filter for short photoplethysmogram signals. In: *Scientific Data* 5 (2018), 5, Nr. 1. – URL <https://www.nature.com/articles/sdata201876>
- [38] LIAO, Shangdi ; LIU, Haipeng ; LIN, Wan-Hua ; ZHENG, Dingchang ; CHEN, Fei: Filtering-induced changes of pulse transmit time across different ages: a neglected concern in photoplethysmography-based cuffless blood pressure measurement. In: *Frontiers in Physiology* 14 (2023). – URL <https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2023.1172150>. – ISSN 1664-042X
- [39] LIU, Chendong ; ZHANG, Yilin ; ZHOU, Huanyu: A comprehensive study of Bluetooth low energy. In: *Journal of Physics Conference Series* 2093 (2021), 11, Nr. 1, S. 012021. – URL <https://doi.org/10.1088/1742-6596/2093/1/012021>

- [40] LIU, Wei ; SHEN, Zhao ; HU, Chao ; MENG, Max Q.-H.: Remote pulse wave monitoring system. In: *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, S. 1567–1572
- [41] MAKOWSKI, Dominique ; PHAM, Tam ; LAU, Zen J. ; BRAMMER, Jan C. ; LE-SPINASSE, François ; PHAM, Hung ; SCHÖLZEL, Christopher ; CHEN, S. H. A.: NeuroKit2: A Python toolbox for neurophysiological signal processing. In: *Behavior Research Methods* 53 (2021), feb, Nr. 4, S. 1689–1696. – URL <https://doi.org/10.3758%2Fs13428-020-01516-y>
- [42] MEJÍA-MEJÍA, E. ; ALLEN, J. ; BUDIDHA, K. ; EL-HAJJ, C. ; KYRIACOU, P.A. ; CHARLTON, P.H.: Photoplethysmography Signal Processing and Synthesis. In: KYRIACOU, P.A. (Hrsg.) ; ALLEN, J. (Hrsg.): *Photoplethysmography*. Elsevier, 2021
- [43] MENGDEN, T. ; HAUSBERG, M. ; HEISS, C. ; MITCHELL, A. ; NIXDORFF, U. ; OTT, C. ; SCHMIDT-TRUCKSÄSS, A. ; WASSERTHEURER, S.: Arterielle Gefäßsteifigkeit – Ursachen und Konsequenzen. In: *Der Kardiologe* 10 (2016), Feb, Nr. 1, S. 38–46. – URL <https://doi.org/10.1007/s12181-015-0041-5>. – ISSN 1864-9726
- [44] MIDDEKE, Martin: Pulswellen-Analyse deckt das Alter der Gefäße auf. 154 (2012), Nr. 16, S. 61–64. – URL <https://link.springer.com/article/10.1007/s15006-012-1144-6>. – Zugriffsdatum: 2024-11-26. – Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 16 Publisher: Urban and Vogel
- [45] MITCHELL, Gary F.: Arterial Stiffness and Hypertension. In: *Hypertension* 64 (2014), Nr. 1, S. 13–18. – URL <https://www.ahajournals.org/doi/abs/10.1161/HYPERTENSIONAHA.114.00921>
- [46] MORILLO, Ángel S. ; CAUSE, Joan L. ; BACIU, Vlad-Eusebiu ; DA SILVA, Bruno ; GARCIA-NARANJO, Juan C. ; STIENS, Johan: PPG EduKIT: An adjustable photoplethysmography evaluation system for educational activities. In: *Sensors* 22 (2022), 2, Nr. 4, S. 1389. – URL <https://www.mdpi.com/1424-8220/22/4/1389>
- [47] OPENTHREAD: *OpenThread*. – URL <https://openthread.io/?hl=de>
- [48] PARK, Junyung ; SEOK, Hyeon S. ; KIM, Sang-Su ; SHIN, Hangsik: Photoplethysmogram Analysis and Applications: An Integrative Review. 12 (2022), S. 808451. – URL <https://www.frontiersin.org/journals/physiology/arti>

- [cles/10.3389/fphys.2021.808451/full](#). – Zugriffsdatum: 2024-12-05. – Publisher: Frontiers. – ISSN 1664-042X
- [49] PARSAORAN, Aldrin J. ; MANDALA, Satria ; PRAMUDYO, Miftah: Study of Denoising Algorithms on Photoplethysmograph (PPG) Signals. In: *2022 International Conference on Data Science and Its Applications (ICoDSA)*, 2022, S. 289–293
 - [50] PIGNATA, Andrea ; FRA, Vittorio ; MACII, Enrico ; URGESE, Gianvito: A Time-synchronized Framework for Bluetooth Low Energy Wireless Sensor Networks. In: *2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT)*, 2024, S. 1–6
 - [51] RANGANATHAN, Prakash ; NYGARD, Kendall E.: TIME SYNCHRONIZATION IN WIRELESS SENSOR NETWORKS: A SURVEY. In: *International Journal of Ubicomp* 1 (2010), S. 92–102. – URL <https://api.semanticscholar.org/CorpusID:10271586>
 - [52] ROTH, Gregory A. u. a.: Global Burden of Cardiovascular Diseases and Risk Factors, 1990–2019: Update From the GBD 2019 Study. In: *Journal of the American College of Cardiology* 76 (2020), Nr. 25, S. 2982–3021. – URL <https://www.sciencedirect.com/science/article/pii/S0735109720377755>. – ISSN 0735-1097
 - [53] RZEPECKI, Wojciech ; RYBA, Piotr: IoTSP: Thread Mesh vs Other Widely used Wireless Protocols – Comparison and use Cases Study. In: *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2019, S. 291–295
 - [54] SAVITZKY, Abraham ; GOLAY, Marcel J.: Smoothing and differentiation of data by simplified least squares procedures. In: *Analytical chemistry* 36 (1964), Nr. 8, S. 1627–1639
 - [55] SEEED STUDIO: *Getting Started with Seeed Studio XIAO ESP32C6* / *Seeed Studio Wiki*. August 2024. – URL https://wiki.seeedstudio.com/xiao_esp32c6_getting_started/. – Zugriffsdatum: 2024-12-16. – Online-Dokumentation
 - [56] SIVRIKAYA, F. ; YENER, B.: Time synchronization in sensor networks: a survey. In: *IEEE Network* 18 (2004), Nr. 4, S. 45–50
 - [57] SPARKFUN ELECTRONICS: *SparkFun MAX3010x Pulse and Proximity Sensor Library - Arduino reference*. 2022. – URL <https://reference.arduino.cc/reference/en/libraries/sparkfun-max3010x-pulse-and-proximity-sensor-library/>

- [58] STEEN, M. van ; TANENBAUM, A. S.: *Distributed Systems*. 4. distributed-systems.net, 2023. – URL <https://www.distributed-systems.net>
- [59] SUBOH, M. Z. ; JAAFAR, R. ; NAYAN, N. A. ; HARUN, N. H. ; MOHAMAD, M. S. F.: Analysis on Four Derivative Waveforms of Photoplethysmogram (PPG) for Fiducial Point Detection. In: *Frontiers in Public Health* 10 (2022), June 30, S. 920946. – URL <https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2022.920946>. – ISSN 2296-2565
- [60] SUKESH RAO, M. ; BANGERA, Sanith C.: Design and Development of a Wearable Wristband for Measuring Pulse Transit Time (PTT) using Pressure Sensors. In: *2023 4th International Conference on Computing and Communication Systems (I3CS)*, 2023, S. 1–4
- [61] TALUKDAR, Md. Toky F. ; PATHAN, Naqib S. ; FATTAH, Shaikh A. ; QUAMRUZZAMAN, Muhammad ; SAQUIB, Mohammad: Multistage Adaptive Noise Cancellation Scheme for Heart Rate Estimation From PPG Signal Utilizing Mode Based Decomposition of Acceleration Data. In: *IEEE Access* 10 (2022), S. 59759–59771
- [62] THE PANDAS DEVELOPMENT TEAM: *pandas-dev/pandas: Pandas*. Februar 2020. – URL <https://doi.org/10.5281/zenodo.3509134>
- [63] THREAD GROUP: Thread Network Fundamentals / Thread Group, Inc. Thread Group, Inc., September 2022. – White Paper. – URL https://portal.threadgroup.org/DesktopModules/Inventures_Document/FileDownload.aspx?ContentID=633. – Zugriffsdatum: 2025-01-31
- [64] TIGGES, Timo ; ROCKSTROH, Jonas ; PIELMUŞ, Alexandru ; KLUM, Michael ; FELDHEISER, Aarne ; HUNSICKER, Oliver ; ORGLMEISTER, Reinhold: In-ear photoplethysmography for central pulse waveform analysis in non-invasive hemodynamic monitoring. In: *Current Directions in Biomedical Engineering* 3 (2017), Nr. 2, S. 587–590. – URL <https://doi.org/10.1515/cdbme-2017-0122>
- [65] TOSI, Jacopo ; TAFFONI, Fabrizio ; SANTACATTERINA, Marco ; SANNINO, Roberto ; FORMICA, Domenico: Performance Evaluation of Bluetooth Low Energy: A Systematic Review. In: *Sensors* 17 (2017), Dezember, Nr. 12, S. 2898. – URL <https://www.mdpi.com/1424-8220/17/12/2898>
- [66] UNWALA, Ishaq ; TAQVI, Zafar ; LU, Jiang: Thread: An IoT Protocol. In: *2018 IEEE Green Technologies Conference (GreenTech)*, 2018, S. 161–167

- [67] VALERIO, Andrea ; BURAIOLI, Irene ; SANGINARIO, Alessandro ; LEONE, Dario ; MINGRONE, Giulia ; MILAN, Alberto ; DEMARCHI, Danilo: A New True Wireless System for Real-Time Pulse Wave Velocity Assessment. In: *IEEE Sensors Journal* 24 (2024), Nr. 15, S. 24365–24376
- [68] VIRTANEN, Pauli u. a.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. In: *Nature Methods* 17 (2020), S. 261–272
- [69] WEBER, T. ; EBER, B. ; ZWEIKER, R. ; HORN, S. ; SOCK, S. ; GRÜNER, P. ; PICHLER, M. ; MAYER, G. ; EISSERER, G. ; MAGOMETSCHNIGG, D. ; ILLYES, M.: Pulswellengeschwindigkeit, zentraler Blutdruck und Augmentationsindex – "neue" Parameter zur Beschreibung eines Endorganschadens der arteriellen Strombahn bei Hypertonie. Pathophysiologie, Methodik, prognostische Bedeutung, Empfehlungen. In: *Journal für Hypertonie* 12 (2008), Nr. 1, S. 7–13
- [70] WOODRUFF, Rebecca C. ; TONG, Xin ; KHAN, Sadiya S. ; SHAH, Nilay S. ; JACKSON, Sandra L. ; LOUSTALOT, Fleetwood ; VAUGHAN, Adam S.: Trends in Cardiovascular Disease Mortality Rates and Excess Deaths, 2010–2022. In: *American Journal of Preventive Medicine* 66 (2024), Nr. 4, S. 582–589. – URL <https://doi.org/10.1016/j.amepre.2023.11.009>. – ISSN 0749-3797
- [71] WU, Yik-Chung ; CHAUDHARI, Qasim ; SERPEDIN, Erchin: Clock Synchronization of Wireless Sensor Networks. In: *IEEE Signal Processing Magazine* 28 (2011), Nr. 1, S. 124–138
- [72] ZAUNSEDER, Sebastian ; VEHKAOJA, Antti ; FLEISCHHAUER, Vincent ; HOOG AN-TINK, Christoph: Signal-to-noise ratio is more important than sampling rate in beat-to-beat interval estimation from optical sensors. In: *Biomedical Signal Processing and Control* 74 (2022), S. 103538. – URL <https://www.sciencedirect.com/science/article/pii/S174680942200060X>. – ISSN 1746-8094
- [73] ZINK, W. u. a.: *2.2 Kardiovaskuläres System*. 2010. Thieme Verlag, 2010. – URL <https://www.thieme-connect.de/products/ebooks/lookinside/10.1055/b-0034-32880>. – Zugriffsdatum: 2024-11-29. – Publication Title: Anästhesie bei alten Menschen. – ISBN 978-3-13-148451-2 978-3-13-184951-9

A Anhang

A.1 Verwendete Hilfsmittel

In der Tabelle A.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle A.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
L ^A T _E X	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments.
ChatGPT	Zur Verbesserung der Sprache und Erläuterung von Programm Teilen.
UMLet	Zur Erstellung von Diagrammen.
EasyEDA	Zur Erstellung von Schaltplänen.
Zotero	Zur Verwaltung der Literatur.

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original