
Entwicklung eines Systems für die Mobile Sensordatenerfassung zur Erkennung von Ganzkörpergesten in Echtzeit

Development of a system for mobile sensor data acquisition to recognize full body gestures in real time

Bachelor-Arbeit

Pascal Dornfeld

KOM-B-0645



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Elektrotechnik
und Informationstechnik
Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation
Prof. Dr.-Ing. Ralf Steinmetz

Entwicklung eines Systems für die Mobile Sensordatenerfassung zur Erkennung von Ganzkörpergesten in Echtzeit

Development of a system for mobile sensor data acquisition to recognize full body gestures in real time

Bachelor-Arbeit

Studiengang: Informatik

KOM-B-0645

Eingereicht von Pascal Dornfeld

Tag der Einreichung: 23. Juli 2019

Gutachter: Prof. Dr.-Ing. Ralf Steinmetz

Betreuer: Philipp Müller und Augusto Garcia-Agundez

Technische Universität Darmstadt

Fachbereich Elektrotechnik und Informationstechnik

Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation (KOM)

Prof. Dr.-Ing. Ralf Steinmetz

Erklärung zur Abschlussarbeit gemäß § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Pascal Dornfeld, die vorliegende Bachelor-Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Bachelor-Arbeit stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Darmstadt, den 23. Juli 2019

Pascal Dornfeld



Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation	3
1.2	Problemstellung und Beitrag	4
1.3	Überblick der Gliederung	5
2	Hintergrund	7
2.1	Sensor	7
2.1.1	Beschleunigungssensor	7
2.1.2	Gyrosensor	7
2.2	Datenübertragung zwischen Wearable und Endgerät	8
2.2.1	Vergleich verschiedener Protokolle	8
2.2.2	Bluetooth Low Energy	9
2.3	Chipkommunikation	10
2.3.1	I2C	10
2.3.2	SPI	10
2.4	Energieversorgung	11
3	Aktueller Stand	13
3.1	Texas Instruments CC2650STK	13
3.2	aconno ACNSENSA	13
3.3	Arduino Primo Core	14
3.4	Zusammenfassung und verwandte Arbeiten	14
4	Design und Planung	15
4.1	Anforderungen	15
4.2	Hauptkomponenten	15
4.2.1	MCU	15
4.2.2	IMU	16
4.3	Zusammenwirken der Komponenten	16
4.4	BLE Schnittstelle	16
4.5	Applikation des Android Smartphones	17
5	Implementation	19
5.1	Architecture	19
5.1.1	nRF52832 Software	19
5.2	Design Decisions	19
5.3	Interaction of Components	19
5.4	Summary	19
6	Evaluation	21
6.1	Goal and Methodology	21
6.2	Evaluation Setup	21
6.3	Evaluation Results	21
6.4	Analysis of Results	21

7	Conclusions	23
7.1	Summary	23
7.2	Contributions	23
7.3	Future Work	23
7.4	Final Remarks	23
	Literaturverzeichnis	23

Zusammenfassung

The abstract goes here...



1 Einleitung

1.1 Motivation

Die Nachfrage nach Wearables steigt kontinuierlich. Abbildung 1.1 zeigt, dass sich der Absatz in den letzten vier Jahren versechsfacht hat. Wearables sind Geräte, die am Körper getragen werden, um zum Beispiel mithilfe von Sensoren Daten zu erfassen [Ben19].

Ein weit verbreiteter Anwendungsfall ist die Herzfrequenzmessung beim Sport mit einem Fitnessarmband. Durch das Auswerten dieser Daten kann die Trainingsintensität in Echtzeit an die Person angepasst und somit die Effektivität gesteigert werden.

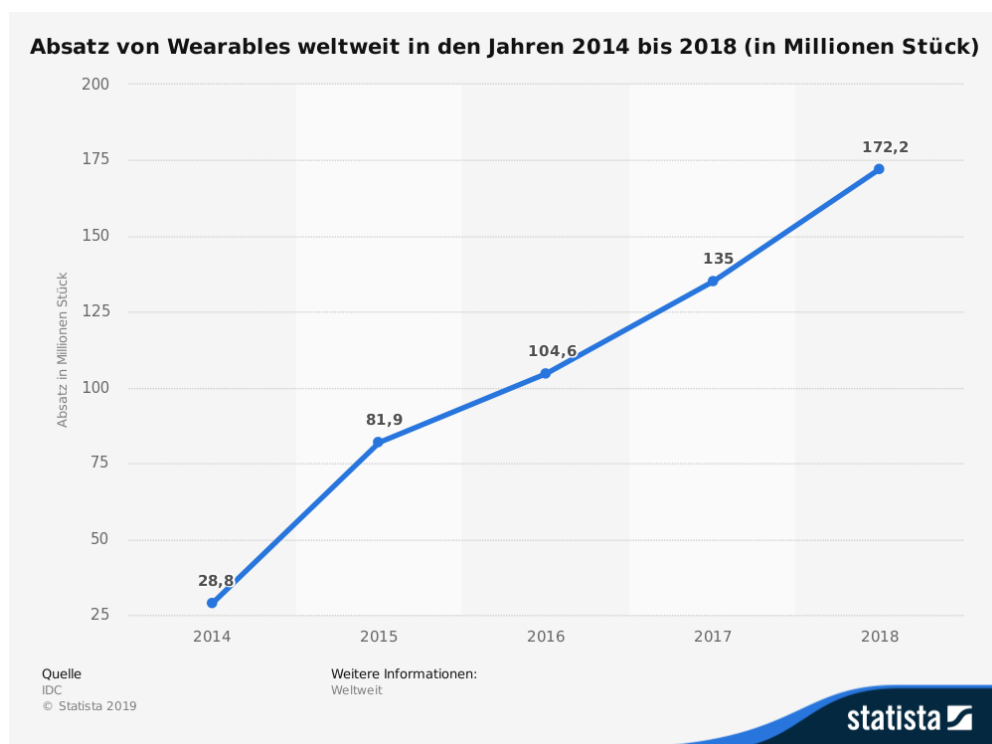


Abbildung 1.1: Absatz von Wearables [Int19]

In dieser Arbeit hingegen wird ein Wearable entworfen, das die Position und Rotation von Gelenken erfasst. Auf der Analyse dieser Daten aufbauend können dann weiterführende Anwendungsfälle entwickelt werden. Folgende sind zum Beispiel interessante Anwendungen:

- Das Erkennen von falsch ausgeführten Übungen beim Sport oder falscher Haltung beim Sitzen oder Stehen im Alltag. Dadurch können negative gesundheitliche Folgen vermieden werden.
- Eine Lösung für mobiles Motion Capturing zum Erstellen von Animationen in Filmen oder Videospielen. Mit der Anzahl der Wearables kann die Auflösung der Bewegung proportional zum Preis skaliert werden.
- Ein Echtzeitsystem zur Übersetzung von Gestensprache in gesprochene Sprache zur Kommunikation von stummen Menschen.
- Für die Bewegungserkennung in Videospielen und Virtual Reality. Abbildung 1.2 zeigt, dass die Nintendo Wii die bisher fünftmeist-verkaufte Konsole ist (Stand: Ende 2018) und damit die Geschichte der Videospielekonsolen prägt. Die Fernbedienung übernimmt dabei vor allem eine Funktion: Das

Tracken der Handposition. Zusätzlich kann man mit dem Nunchuk per Kabel einen zweiten Sensor für die andere Hand anschließen. Würde stattdessen eine Konsole mit dem hier entwickelten Wearable entworfen werden, würden die Hände beim Spielen frei bleiben, das Kabel zwischen den Händen wegfallen und es könnten weitere Gliedmaßen getrackt werden. VR-Headsets und die Xbox Kinect nutzen dagegen Kameras zur Ganzkörpergestenerkennung. Diese Lösung ist dagegen nicht mobil einsetzbar und anfällig auf Störeinflüsse wie fehlender Sichtkontakt.

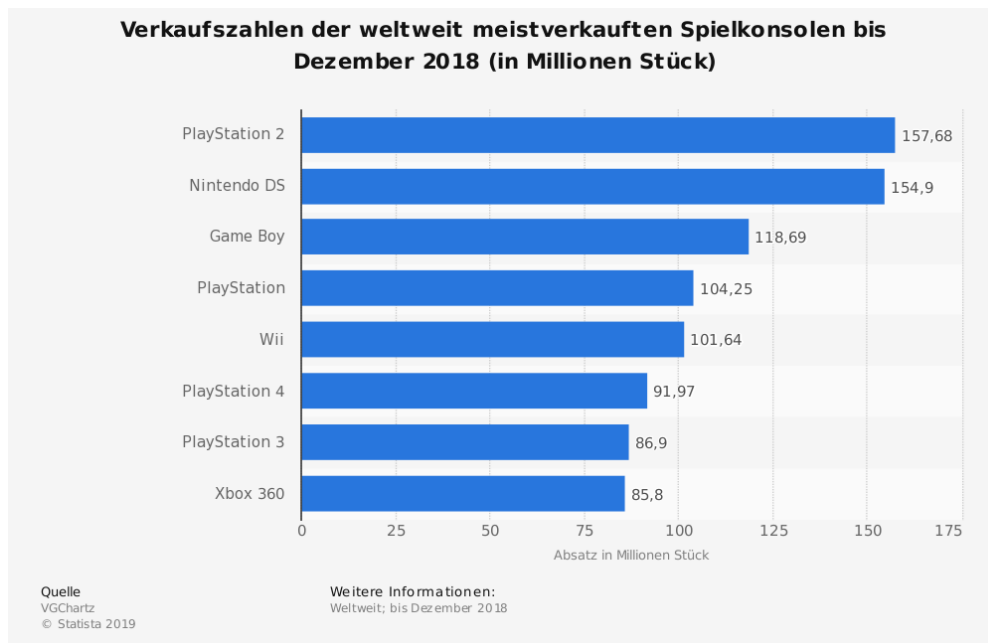


Abbildung 1.2: Verkaufszahlen der weltweit meistverkauften Spielkonsolen [VGC19]

Das Wearable kann also in vielen Bereichen Anwendung finden. Von Medizin über Kommunikation zur Produktivität und Unterhaltung.

1.2 Problemstellung und Beitrag

Folgende Ziele ergeben sich für diese Arbeit:

1. Um eine gute Mobilität zu gewährleisten, muss das Wearable klein und leicht sein, damit es auch beim Sport praktikabel bleibt.
2. Die Datenübertragung muss ausreichend schnell sein um die Datenrate zu übertragen, die für Gestenerkennung nötig ist.
3. Die Energiekapazität und der Energieverbrauch müssen verhältnismäßig zueinander abgestimmt sein. Die Batterie soll keinesfalls öfter als einmal pro Tag geladen oder gewechselt werden müssen.
4. Die eingesetzten Protokolle sollen weit verbreitet sein, damit möglichst viele Endgeräte von dem Wearable profitieren können und bestenfalls einzelne Komponenten des Wearables einfach durch Neuere ersetzt werden können.

Wegen der hohen Mobilität und Verbreitung wurde als Empfänger der Daten ein Smartphone ausgewählt, wobei durch Implementierung der Schnittstelle später auch andere Geräte genutzt werden können. Das Smartphone soll die Schnittstelle exemplarisch implementieren und die Daten sowie Statistiken zur Verbindung zur Auswertung anzeigen. Das Wearable soll bei Verbindung Sensordaten mit variabler Rate bereitstellen um Datenübertragung und Energieverbrauch bei verschiedenen Konfigurationen auswerten zu können.

1.3 Überblick der Gliederung

Zunächst werden verschiedene Protokolle, Techniken oder Komponententypen verglichen, die für das Wearable wichtig sind. Dann werden bestehende Lösungen vorgestellt und geprüft, welche Vor- und Nachteile sie bieten. Im Anschluss werden die eingesetzten Komponenten bezeichnet und beschrieben, wie sie miteinander interagieren sollen. In der Implementierung wird die Beschaltung beschrieben und es werden die Probleme benannt, die bei der Implementierung aufgetreten sind. Zum Schluss wird das System anhand der genannten Ziele evaluiert und ein Fazit gezogen.



2 Hintergrund

2.1 Sensor

Um Gelenkstellungen anzugeben wird eine Position und Rotation gebraucht. Diese wird von einem Beschleunigungssensor und einem Gyrosensor erfasst und besteht aus je 3 Achsen. Die Sensoren sind platz- und energiesparend zusammen in einer Inertialen Messeinheit (IMU) erhältlich. Im Folgenden wird ein Einblick in die Funktionsweise der beiden Sensorentypen gegeben.

2.1.1 Beschleunigungssensor

Beschleunigungssensoren können die Beschleunigung messen, die auf den Sensor einwirkt. Im Bereich der kleinen Bauteile werden zwei Techniken dafür eingesetzt.

Beim kapazitiven Sensor, der in Abbildung 2.1 illustriert ist, wird eine Masse senkrecht-federnd parallel zu einer Platte befestigt, mit der sie einen Kondensator bildet. Bei einer Beschleunigung bewegt sich die Masse zur Platte hin oder davon weg und die Kapazität des Kondensators ändert sich. Durch diese Änderung lässt sich dann der Wert für die Beschleunigung berechnen. [Ele19c]

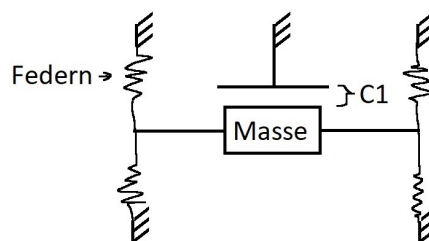


Abbildung 2.1: Kapazitiver Beschleunigungssensor. Basierend auf [sen18]

Beim piezoresistiven Sensor, der in Abbildung 2.2 dargestellt ist, wird ein Stoff mit piezoresistivem Effekt genutzt, der bei Dehnung seinen Widerstand ändert. Eine Masse wird an diesen piezoresistiven Stoff befestigt und durch die Beschleunigung wird der Stoff gedehnt, wodurch durch die Änderung des Widerstands die Beschleunigung berechnet werden kann. [Ele19a]

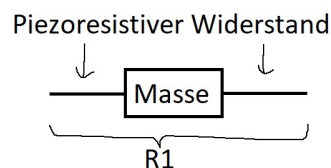


Abbildung 2.2: Piezoresistiver Beschleunigungssensor. Basierend auf [sen18]

2.1.2 Gyrosensor

Mit einem Gyrosensor, zu sehen in Abbildung 2.3, wird die Drehung gemessen. Eine Masse wird dabei senkrecht-federnd parallel zu einer Platte befestigt, sodass wieder ein Kondensator entsteht und parallel zu der Platte von außen zum Schwingen gebracht. Dieses Konstrukt gibt es ein zweites Mal in die andere Richtung schwingend. Bei einer Drehung bewegen sich die Massen durch den Coriolis Effekt.

Durch die Subtraktion der beiden Kapazitäten kann man dann die Drehgeschwindigkeit berechnen. Da die Schwingung gegensätzlich ist, bewegen sich die Massen durch den Coriolis Effekt bei einer Drehung in entgegengesetzte Richtungen und die Differenz der Kapazitäten ändert sich. Bei einer geraden Beschleunigung bewegen sich beide Massen in die selbe Richtung und die Differenz der Kapazitäten ändert sich nicht. [VPdN⁺10]

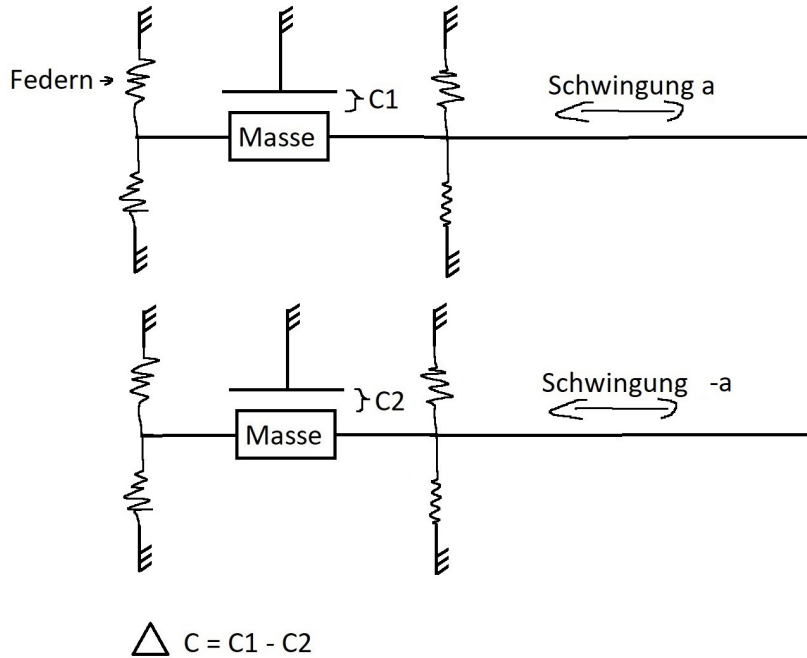


Abbildung 2.3: Gyrosensor

Da konstant eine Schwingung auf der Masse gehalten werden muss, damit der Coriolis Effekt auftritt, benötigt der Gyrosensor meist mehr Strom als der Beschleunigungssensor, was später von den Komponenten bestätigt wird.

2.2 Datenübertragung zwischen Wearable und Endgerät

Um das am Besten geeignete Protokoll zu finden, um die Sensordaten der IMU auf die Endgeräte zu übertragen, wurden verschiedene Protokolle verglichen und nach dem Ausschlussverfahren ausgewählt.

2.2.1 Vergleich verschiedener Protokolle

Eine drahtgebundene Datenübertragung ist schnell und sehr energiesparend und es sind viele verbreitete Anschlüsse spezifiziert. Die Mobilität ist dabei aber sehr stark eingeschränkt, da Kabel insbesondere beim Sport sehr stören.

Während einige drahtlose Kommunikationsprotokolle wie NFC wegen der geringen Reichweite nicht geeignet sind, sind Mobilfunkprotokolle wie LTE wegen zu hoher Reichweite ineffizient [ZJM17]. Die bekannten Protokolle WIFI und Bluetooth arbeiten unter anderem auf der 2.4-GHz Antenne, weswegen Peer-to-Peer(P2P) Protokolle auf dieser Frequenz in Tabelle 2.1 verglichen wurden.

Tabelle 2.1: P2P-Protokolle auf der 2.4-GHz Frequenz

Protokoll	Typische Anwendung	Kommentar
Wi-Fi Hotspot	Teilen der Internetverbindung in einem lokalem Netzwerk	Am Smartphone kann ein Hotspot erstellt werden, die Wearables könnten diesen suchen und sich verbinden, woraufhin z.B. über eine REST-API kommuniziert werden kann. Es kann auch die 5-GHz Frequenz genutzt werden.
Wi-Fi Direct	P2P-Kommunikation	Wie Wi-Fi Hotspot aber die Wearables bekommen keinen Internetzugriff vom Smartphone.
Bluetooth (BT)	Datenübertragung, Audiostreaming	Nicht für niedrigen Energieverbrauch ausgelegt.
BT Low Energy (BLE)	P2P-Kommunikation mit geringem Energieverbrauch, z.B. Smartwatches	Während die maximale Geräteanzahl nicht vom Protokoll spezifiziert ist, ist diese in Android auf 7 Geräte beschränkt ^a . Weiterhin ist die Anzahl zum Beispiel durch die Auslastung der Geräte und des Funkkanals begrenzt.
BT Mesh	Mesh-Kommunikation mit geringem Energieverbrauch	Nachrichten werden hierbei über das gesamte Netzwerk geflutet, wodurch die Reichweite des Netzwerkes erhöht wird und die Auslastung auf das Netzwerk verteilt wird aber der Energieverbrauch aller Knoten steigt. BLE Geräte können mit Mesh-Netzwerken kompatibel gemacht werden, indem sie sich zu einem Knoten des Netzwerks verbinden.
Thread, ZigBee	Smarthome	Nicht ohne weitere Hardware unter Android
Ant, Ant+	Sensordatenübertragung mit geringem Energieverbrauch	Nicht von allen Smartphones ohne zusätzliche Hardware unterstützt

^a Beschränkung durch GATT_MAX_PHY_CHANNEL: https://android.googlesource.com/platform/external/bluetooth/bluedroid/+/-/master/include/bt_target.h#1428

Da Bluetooth nicht für einen geringen Energieverbrauch ausgelegt ist, werden die Alternativen BLE und BT Mesh bevorzugt. Die Smarthome Protokolle funktionieren ohne zusätzlicher Hardware nicht am Smartphone. Ant und Ant+ sind zwar für die Sensordatenübertragung bei Smartphones ausgelegt, aber brauchen zusätzlich ein Dongle, wenn diese das Protokoll nicht unterstützen. Sowohl handelsübliche Laptops als auch das vorliegende Smartphone, ein Pocophone F1, unterstützt die Protokolle nicht. "BLE ist etwa 30% energieeffizienter als Wi-Fi"¹ [PPLA17], weswegen die Wi-Fi Protokolle heraus fallen. BT Mesh löst zwar das Problem der Geräteanzahlbeschränkung unter Android, allerdings auf Kosten des Energieverbrauchs. Die erweiterte Reichweite wird bei einem Wearable in der Regel nicht benötigt. Dem Ausschlussverfahren nach soll die Datenübertragung zwischen Wearable und Endgerät mit BLE stattfinden, da dieses Protokoll von fast jedem Smartphone und Laptop unterstützt wird und den Anforderungen am Besten entspricht. BT Mesh ist eine Alternative, dessen zusätzlicher Energieverbrauch zunächst evaluiert werden müsste.

2.2.2 Bluetooth Low Energy

BLE ist eine Abwandlung von Bluetooth, die für geringen Energieverbrauch optimiert ist. Geräte können dabei die Rolle vom Central oder Peripheral einnehmen, die einem Master-Slave Protokoll gleichkom-

¹ Übersetzung durch den Verfasser

men. Um eine Datenverbindung zu starten, kündigt sich die Peripheral auf den drei Advertising Frequenzen an und die Central hört diese ab. Die Parameter dafür, zum Beispiel die bis zu 31 Byte große Payload, werden im Generic Access Profile (GAP) eingestellt. Die Central kann auf die Ankündigung der Peripheral antworten und sich mit Diesem verbinden. [Ada19]

Nach dem Verbindungsaufbau gibt die Central eine Geschwindigkeit vor. Die Geschwindigkeit besteht aus einem Connection Interval, einer Slave Latency und einem Supervision Timeout. Das Connection Interval beschreibt in welchen Zeitabständen die Datenbursts passieren. Sie liegt zwischen 7.5 ms und 4 s. Die Slave Latency bestimmt, wie oft das Peripheral die Datenbursts pausieren kann um Energie zu sparen, falls keine Daten gesendet werden müssen. Die Supervision Timeout besagt, nach welcher Zeit eine Verbindung als abgebrochen gilt, wenn eine Seite nicht mehr antwortet. Später kann einerseits die Peripheral neue Geschwindigkeiten vorschlagen, die von der Central angenommen oder abgelehnt werden oder andererseits die Central neue Geschwindigkeiten von sich aus bestimmen. [Tex16b]

Bei einer Verbindung stellt die Peripheral Services zur Verfügung, die aus Characteristics bestehen. Jede Service und Characteristic hat dabei eine eindeutige UUID. Diese ist 16 Bit groß, wenn sie aus den vordefinierten Profilen besteht oder 128 Bit für eigene Definitionen. Eine Characteristic repräsentiert einen Wert und kann im Lese- und Schreibzugriff eingeschränkt werden. Die Central kann das notificate- oder indicate-Bit setzen, wodurch die Peripheral Änderungen des Wertes mitteilt, wobei beim Indicate der Empfang der Updates von der Central bestätigt werden muss. [Nor16]

2.3 Chipkommunikation

Eine IMU enthält in der Regel keine programmierbare Chiparchitektur sondern wird mit einer Microprocessor Unit (MPU) betrieben. Diese beinhaltet unter anderem eine Recheneinheit und einen Programmspeicher. Zur Kommunikation zwischen IMU und MPU kann man wiederum zwischen einigen Systemen wählen. Die verwendeten IMUs unterstützen die weit verbreiteten Master-Slave-Systeme Inter-Integrated Circuit (I2C) und Serial Peripheral Interface (SPI), die auch Beide auf der gewählten MPU laufen. Die Chipkommunikation läuft in der Regel mit einer langsameren Frequenz als die Recheneinheit. Einerseits kann die Übertragungsfrequenz von der Software emuliert werden, indem Takte der Recheneinheit übersprungen werden, wodurch die Recheneinheit nicht in den Schlafmodus gehen kann. Andererseits kann die Recheneinheit einen weiteren Chip nutzen, der mit einem Buffer die Übertragung effizient abarbeiten kann.

2.3.1 I2C

Beim I2C-Protokoll sieht die Verdrahtung Serial Clock (SCL) und Serial Data (SDA) mit Pull-Up Widerständen vor. Der Master startet eine Datenübertragung mit START indem er SDA runter zieht während SCL oben ist. Dann gibt der Master den Takt auf SCL vor. Es folgt ein Paket mit 7 Bit für Slaveadresse und 1 Bit für Read oder Write. Der Slave bestätigt den Empfang mit ACK indem es SDA für ein Bit runterzieht. Nun können 8-Bit-Datenpakete übertragen werden, die jeweils mit ACKs bestätigt werden. Zum Schluss sendet der Master STOP indem er SDA hoch zieht während SCL oben ist. [NXP14]

2.3.2 SPI

Beim SPI-Bussystem sind Chip Select (CS), Serial Clock (SCK), Master-In-Slave-Out (MISO) und Master-Out-Slave-In (MOSI), meist mit Pull-Up Widerständen, verkabelt. Der Master startet die Übertragung, indem es CS runter zieht und dann auf SCK den Takt vorgibt. Auf MISO kann der Slave dann Daten zum Master schicken und auf MOSI der Master Daten zum Slave. Um die Übertragung zu beenden wird CS vom Master wieder hochgezogen. [mic]

2.4 Energieversorgung

Zu den typischen mobil verfügbaren Spannungsquellen gehören wiederaufladbare Akkus und Einwegbatterien, die elektrische Energie chemisch speichern können. Die Gängigsten basieren auf Nickel oder Lithium, wobei die Lithiumbasierenden wegen der höheren Energiemenge bei gleichem Gewicht und Größe unter dem Gesichtspunkt der Mobilität präferiert werden. Genauso benötigen Lithiumakkus eine Ladeelektronik sowie einen Entladeschutz welche Platz benötigen und Energie verbrauchen und verlieren ihre maximale Kapazität mit zunehmenden Alter. Durch ihre maximale Spannung von 4.2 Volt könnten sie auch nicht ohne Spannungswandler mit den meisten Komponenten betrieben werden, der wiederum Effizienz einbüßen würde. [Ele19b]

Einweglithiumbatterien im Knopfzellenformat sind dagegen unkompliziert zu nutzen, haben eine geringe Selbstentladung von $< 1 \%$ Jahr, lassen sich durch ihre flache Bauweise platzsparend an eine Platine integrieren und sind mit ihrer Nennspannung von 3 V direkt mit den Komponenten nutzbar [Var14].



3 Aktueller Stand

Es existieren bereits einige Lösungen für den Anwendungsfall.

3.1 Texas Instruments CC2650STK

Darunter fällt beispielsweise das TI CC2650STK. Es besteht aus der CC2650 MCU sowie 10 Sensoren, darunter eine MPU-9250 IMU, die über I2C angeschlossen ist. Eine Auflistung der überflüssigen Komponenten, die nicht komplett abgeschaltet werden können, findet sich in Tabelle 3.1. Eine CR2032 Knopfzelle mit 230 mAh verliert dadurch $\frac{365d * 24h}{(230mAh/0.00351mA)} \approx 13.4 \%$ der Kapazität pro Jahr. Die IMU verbraucht 3.4 mA, wenn Gyrosensor bei 1 kHz und Beschleunigungssensor bei 4 kHz laufen [Inv16]. Die MCU unterstützt zwar BLE aber kein BT Mesh. Das System ist nicht mehr in Europa verfügbar. [Tex]

Tabelle 3.1: Stromverbrauch der ausgeschalteten zusätzlichen Komponenten am TI CC2650STK

Sensortyp	Stromverbrauch ausgeschaltet
Lastschalter	0.9 μA [Tex15]
Feuchtigkeitssensor	0.11 μA [Tex16a]
Drucksensor	0.1 μA [Bos18b]
Infrarottemperatursensor	2 μA [dig]
Lichtsensor	0.4 μA [Tex17]]
Gesamt	3.51 μA

3.2 aconno ACNSENSA

Das aconno ACNSENSA besteht aus einer nRF52832 MCU und 5 Sensoren, darunter eine LSM9DS1 IMU über I2C. Eine Auflistung der überflüssigen Komponenten, die nicht komplett abgeschaltet werden können, findet sich in Tabelle 3.2. Eine CR2450 Knopfzelle mit 620 mAh verliert dadurch nur $\frac{365d * 24h}{(620mAh/0.0014mA)} \approx 2 \%$ der Kapazität pro Jahr. Der Gyrosensor der IMU verbraucht 4.0 mA bei 952 Hz. Weitere 600 μA sind zusammen angegeben für den Beschleunigungssensor bei 952 Hz und Magnetometer bei 20 Hz, der sich aber ausschalten lässt [STM15]. Die MCU unterstützt BT Mesh. [aco17b] Das System ist nicht gut verfügbar. So sind bei bekannten Händler nur 5 Exemplare zu bekommen [mou]. Die Produktwebseite sowie die Datenblätter waren während über den Zeitraum dieser Arbeit teilweise nicht zugreifbar und es existieren mehrere Versionen unter dem selben Namen (vgl. Datenblatt und Produktbild bei mouser.de).

Tabelle 3.2: Stromverbrauch der ausgeschalteten zusätzlichen Komponenten am aconno ACNSENSA

Sensortyp	Stromverbrauch ausgeschaltet
Drucksensor	1 μA [aco17b]
Feuchtigkeitssensor	0.06 μA [aco17b]
Lastschalter	0.34 μA [Vis15]
Gesamt	1.4 μA

3.3 Arduino Primo Core

Der Arduino Primo Core beinhaltet ein nRF52832 MCU und 3 Sensoren, inklusive einer LSM6DS3 IMU über I2C. Eine Auflistung der überflüssigen Komponenten, die nicht komplett abgeschaltet werden können, findet sich in Tabelle 3.3. Eine CR2032 Knopfzelle mit 230 mAh verliert dadurch $365d * 24h$ $\frac{(230mAh/0.0015mA)}{365d * 24h} \approx 5.7 \%$ der Kapazität pro Jahr. Die IMU verbraucht bei 1.6 kHz nur 1.25 mA [STM17b]. Das System kann nicht mehr gekauft werden und ist auf der Produktseite als 'retired' gelistet.[ARD] [Ard16]

Tabelle 3.3: Stromverbrauch der ausgeschalteten zusätzlichen Komponenten am Arduino Primo Core

Sensortyp	Stromverbrauch ausgeschaltet
Magnetometer	1 μA [STM17a]
Feuchtigkeitssensor	0.5 μA [STM16]
Gesamt	1.5 μA

3.4 Zusammenfassung und verwandte Arbeiten

Mit den existierenden Lösungen wäre es grundsätzlich möglich das Wearable umzusetzen. Allerdings gibt es neben der problematischen Verfügbarkeit eine Limitierung auf die gegebene Hardware. So verschwenden alle Lösungen Energie und Platz durch Sensoren, die für den benötigten Anwendungsfall nicht nötig sind, auch wenn dies im Falle des aconno ACNSENSA vernachlässigbar ist. Hier ist auch zu beachten, dass zusätzlich zu dem Standbyverbrauch der Komponenten auch Leckströme durch Kondensatoren und Verluste durch Pull-Up Widerstände hinzuzufügen sind. Alle genannten Lösungen nutzen für die Kommunikation zwischen MCU und IMU das I2C Protokoll, während in einer Arbeit [MT12] ermittelt wurde, dass I2C einen doppelt so hohen Energieverbrauch wie SPI hat. Auch zwischen den IMUs gibt es gravierende Unterschiede im Energieverbrauch. So verbraucht die IMU des Arduino Prime Core weniger als ein Drittel der Energie der anderen Beiden und enthält als Einziges keinen Magnetometer. Ein Magnetometer entspricht einem elektrischen Kompass und bestimmt die Rotation in Ausrichtung zu den Erdpolen. In der Arbeit [ABCO12] wurde ein Sensorfusionsalgorithmus entwickelt, der die Genauigkeit des Gyrosensors mit den Daten eines Magnetometers verbessert. Dies kann in zukünftigen Änderungen integriert werden, falls die Genauigkeit der verwendeten IMU nicht ausreichen sollte. In dem Projekt [Cho] wird ein System zur Bewegungsaufnahme mit einer IMU entwickelt. Hierbei werden die Sensoren per Kabel miteinander verbunden. Da die Software allerdings open-source ist, könnte darauf aufbauend eine eigene Bewegungserkennungssoftware erstellt werden.

4 Design und Planung

4.1 Anforderungen

Als Komponenten werden hauptsächlich eine IMU und eine kompatible MCU, die Daten über BLE übertragen kann sowie für zukünftige Tests auch BT Mesh unterstützt, benötigt. Die Komponenten sollen ähnliche Anforderungen an die Spannungsversorgung besitzen wie eine Lithiumknopfzelle sie bereitstellen kann, damit kein zusätzlicher Spannungsregler benötigt wird. Ferner sollen die in der Einleitung festgelegten Ziele angestrebt werden.

4.2 Hauptkomponenten

Die Arbeit hat sich zunächst danach gerichtet Komponenten zu wählen, die es auf fertige Platinen gelötet gibt, da sich QFN-Komponenten händisch nicht löten lassen.

Da sich der Energieverbrauch von MCUs anhand der Datenblätter schlecht vergleichen lässt, wurde betrachtet, welche Komponenten sich oft zusätzlich auf den Platinen befinden aber viel Strom verbrauchen. In diesem Blog [mad19] wurden verschiedene Komponenten von einem Arduino Pro Mini entfernt und deren Auswirkungen auf den Stromverbrauch ermittelt. Das Entfernen der LED, die dauerhaft leuchtet um den Betrieb anzuzeigen, hat eine Einsparung von mehr als 15 % bei aktiver Nutzung bewirkt und mehr als 90 % im Schlafmodus der Recheneinheit. Das Entfernen des Spannungsreglers hat mindestens weitere 9 % bei Nutzung und 75 % im Standby eingespart. Dieser Blog [Spa] beschreibt, dass ein integrierter Programmierer zwar einen einfach zu verwendenden USB Anschluss bietet, aber gleichzeitig die Stromaufnahme verdreifachen kann. Während eine LED problemlos zu entfernen ist, wurde auf die anderen beiden Bauteile nach Möglichkeit verzichtet, da ein Entfernen auch die Funktion der Platine einschränken oder die Platine zerstören kann.

4.2.1 MCU

Die Bluetooth SIG, die das Bluetooth Protokoll spezifiziert, hat eine Liste mit geprüften Implementierungen von BT Mesh veröffentlicht [Blu]. Mit dieser Liste wurden die Herstellerseiten nach passenden verfügbaren MCUs durchsucht bis die Wahl auf die nRF52832 MCU gefallen ist. Diese war schon in den zwei fertigen Produkten von Arduino und aconno enthalten weswegen von einer hohen Bekanntheit auszugehen ist.

Der nRF52832 hat eine Spannungsanforderung von 1.7 bis 3.6 V, wodurch er sich direkt an einer Knopfzelle betreiben lässt. Das EasyDMA-Modul enthält einen Buffer um SPI- und I2C-Kommunikation abzuhandeln, während die MCU im Schlafmodus ist. Das schwächere Modell nRF52810 unterstützt BT Mesh nicht. Das stärkere Modell nRF52840 hat mehr Speicherplatz und unterstützt weitere Protokolle aber verbraucht dementsprechend mehr Energie. [Nor]

Durch den Pinabstand von 2.54 mm, wie er sich auch auf Arduinos vorfindet, ist die Platine von SparkFun sehr einfach zu verwenden. Sie enthält eine Power-LED, die sich über JP6 trennen lässt. Es existiert ein Spannungsregler, der sich überbrücken lassen müsste. Die Platine stellt eine Alternative dar, falls sich mit den kleineren Platinen wegen der Größe nicht mehr arbeiten lassen würde. Die Größe dieser Platine ist nicht direkt angegeben, aber wegen des Pinabstands mindestens 15 x 43 mm. [Spa16]

Der ACN52832 von aconno enthält weder Power-LED noch Spannungsregler sondern nur die nötigsten Komponenten. Er hat eine Größe von 20 x 24.8 mm. [aco17a]

Der BL651 von Laird enthält ebenfalls nur die nötigsten Komponenten und hat eine Größe von 10 x

14 mm [Lai19]. Es existieren noch weitere Platinen vom nRF52832 sodass eine sehr gute Verfügbarkeit gegeben ist. Sollte eine Platine nicht mehr erhältlich sein, könnte sie durch eine Andere ersetzt werden, da die Platinen nur die Pins vom nRF52832 besser zugänglich machen und die nötigen Taktgeber, Kondensatoren und Antennen bereitstellen. Für den Prototypen wurde zunächst das nRF52 DK von Nordic verwendet, da es auch den externen Programmierer für die Platinen enthält und die Pins ohne zu Löten mit Jumperkabel verbunden werden können.

4.2.2 IMU

Auf der Suche nach einer passenden IMU haben sich der STMicroelectronics LSM6DSL und Bosch BMI260 als besonders energiesparend hervorgetan. Der LSM6DSL läuft mit 1.71 - 3.6 V, verbraucht 0.4 mA im normalen Modus und 0.65 mA im Performance-Modus [STM17c]. Die zugehörige Platine STEVAL-MKI178V2 enthält keine überflüssigen Komponenten.

Der BMI260 läuft mit 1.7 - 3.6 V und verbraucht 0.685 mA. Der Sensor war zu Beginn der Arbeit angekündigt. Da er pin-kompatibel zum Vorgänger BMI160 ist, wurde mit Diesem entwickelt. Zum Ende der Arbeit sollte er durch den BMI260 getauscht werden, aber leider war er da nicht verfügbar. [Bos]

Der BMI160 läuft auch mit 1.7 - 3.6 V aber verbraucht 0.925 mA [Bos18a]. Das BMI160 Shuttle Board enthält zusätzlich einen BMM150 Magnetometer. Dieser hat im Standby einen Verbrauch von 1 μ A, kann aber mechanisch entfernt werden [Bos19].

Beide IMUs haben einen FIFO-Buffer eingebaut. Die gemessenen Daten können in dem Buffer hinterlegt werden, sodass die MCU die Daten in Bursts lesen und länger im Schlafmodus bleiben kann, da ein Wechsel der Energiemodi wiederum Energie kostet.

Mit zwei Interrupt Pins können die IMUs die MCU über bestimmte Events informieren. Das DataReady-Event benachrichtigt, wenn der Sensor neue Daten gemessen hat. Das FIFO-Watermark- oder FIFO-Threshold-Event wird geworfen, wenn die FIFO über einen eingestellten Prozentsatz gefüllt ist. Das FIFO-Full-Event zeigt an, dass die FIFO komplett gefüllt ist. Für spezielle Anwendungen unterstützen beide IMUs weitere Events, wie das Significant-Motion-Event um ein Aufheben des Sensors durch einen Nutzer zu melden.

4.3 Zusammenwirken der Komponenten

Während SPI 4 Datenverbindungen braucht, braucht I2C 2 Datenverbindungen. Da die Pins in der Software der MCU zugewiesen werden, kann für SPI verkabelt und die selbe Verdrahtung auch für I2C genutzt werden. Abbildung 4.1 stellt die Verbindungen graphisch dar.

Wenn das Wearable nicht mit einem Endgerät verbunden ist, soll es in den Schlafmodus wechseln. Über einen Knopf soll es aufgeweckt werden und mit dem Advertise beginnen. Mit Diesem soll das Advertise auch wieder abgebrochen werden und das Wearable in den Schlafmodus versetzt werden können. Wenn das Wearable mit einem Endgerät verbunden ist, soll es sich trennen lassen, indem der Knopf über einige Sekunden gedrückt gehalten wurde, damit es sich nicht versehentlich trennt.

Um dem Nutzer ein Feedback über den Status zu geben, soll eine LED verwendet werden. Diese soll beim Advertise in regelmäßigen Abständen blinken, sodass sich der Stromverbrauch nicht wie bei einer immer angeschalteten Power-LED auswirkt. Wenn eine Verbindung hergestellt wurde, soll eine zweite LED zu Beginn der Verbindung aufblinken. Wenn während einer Verbindung kurz auf den Knopf gedrückt wird, soll die zweite LED kurz aufleuchten.

4.4 BLE Schnittstelle

In einer Characteristic sollen die 3 Achsen des Beschleunigungssensors, die 3 Achsen des Gyrosensors und der Zeitstempel hinterlegt werden. Damit soll sich der Bewegungsablauf des Sensors zeitlich korrekt

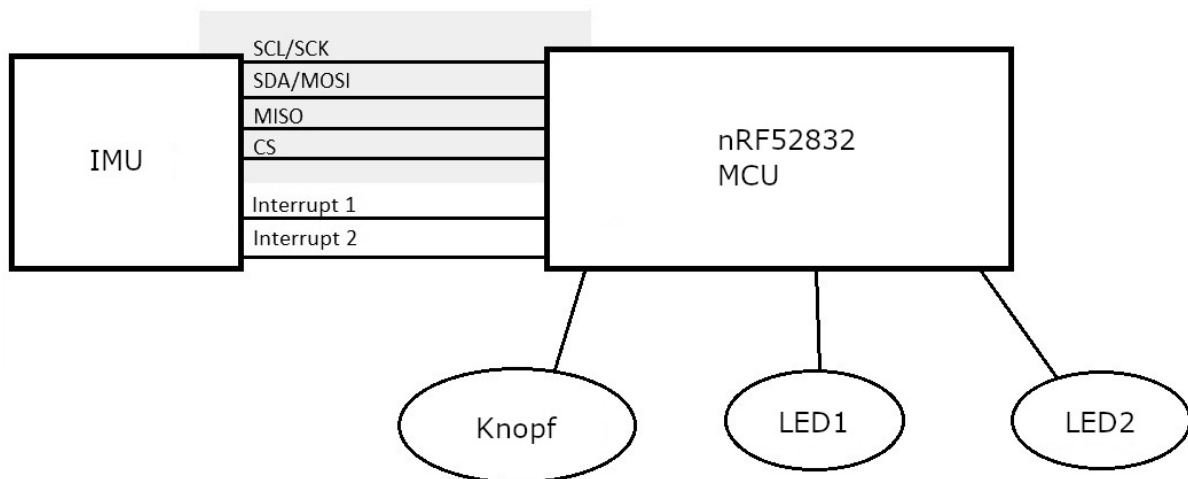


Abbildung 4.1

rekonstruieren lassen, da BLE die Daten in Burst mit Zeitabständen vom Connection-Interval übergibt und die Daten in Burst von der FIFO abgerufen werden können. Um die MCU möglichst selten aus dem Schlafmodus zu holen, soll die FIFO der IMU in den Buffer der EasyDMA geleert werden, sobald sie komplett gefüllt ist. Zusätzlich soll umgefüllt werden, wenn die MCU durch andere Interrupts die z.B. durch den BLE-Stack aufgeweckt wird. Der Wert der Characteristic soll von der MCU nur gelesen werden können, aber es soll das Notify-Bit gesetzt werden können.

Mit einer zweiten Characteristic soll sich die Abfragerate der IMU einstellen lassen. Da die Sensorabfragerate damit von der Datenübertragungsrate unabhängig ist, lässt sich die Energieaufnahme getrennt evaluieren. Während eine niedrige Sensorabfragerate mit einer hohen Datenübertragungsrate sehr geringe Latenzen der Sensordaten ermöglicht, sind hohe Sensorabfrageraten mit niedrigen Datenübertragungsraten bei weniger zeitkritischen Anwendungen zur intensiven Auswertung interessant. Damit die Verbindung aufrecht erhalten werden aber die Datenerfassung pausiert werden kann, soll die IMU über die Characteristic auch kurzzeitig ausgeschaltet werden können. Die zweite Characteristic soll von der MCU gelesen und geschrieben werden können.

4.5 Applikation des Android Smartphones

Beim Start der App soll nach den Wearables gesucht und zu den Wearables verbunden werden. Bei einer Verbindung soll das Notify-Bit der ersten Characteristic gesetzt werden, damit neue Sensordaten beim Smartphone ankommen. Es ist nicht Ziel der Arbeit eine komplette Gestenerkennung zu implementieren. Um Einschätzen zu können, ob die empfangenen Daten korrekt erfasst werden und ankommen, sollen deshalb die Sensordaten und Statistiken zur Verbindung graphisch dargestellt werden. Da es im Android SDK keine einfache Möglichkeit dazu gibt, wurde die Bibliothek Androidplot¹ genutzt. Der Wert der zweiten Characteristic soll gesetzt und die Verbindungsdatenrate geändert werden können.

¹ <http://androidplot.com/>, abgerufen am 10.07.2019



5 Implementation

5.1 Architecture

5.1.1 nRF52832 Software

Offizielle Liste ist hier: https://www.nordicsemi.com/DocLib/Content/User_Guides/getting_started/latest/UG/common/nordic_tools

- IAR: 30 Tage Demo oder beschränkte Funktion (32 Kbyte code u.A.)
- Keil uVision: beschränkte Version (32 KByte Code und Debugger). Essential Version 1330€ pro Jahr.
- MBed.org: sehr einfach, da online. weitere abstraktion wie bei arduino. code, compiler, bin-datei auf stick ziehen, fertig. leider (noch) kein mesh support.
- Project Zephyr: ist noch nicht offiziell supportet
- GCC. Diese Anleitung: <https://www.disk91.com/2017/technology/hardware/discover-nordic-semi-n>
Bug in aktueller GNU Arm Embedded Toolchain 8-2018-q4-major beim Kompilieren: C:/nrf52/arm_tools/bin/none-eabi-objcopy: _build/nrf52832_xxaa.hex 64-bit address 0x4b4fa300000000 out of range for Intel Hex file
fix: bin/arm-none-eabi-objcopy.exe mit der datei von version 7-2018-q2-update ersetzen.
Wenns läuft ists leider trotzdem mist, weil jeder Ordner von der library einzeln eingetragen werden muss. Am ende bekommt man die .hex datei und muss die rüberkopieren.
- Segger Embedded Studio: funktioniert zwar ganz einfach, aber ist nicht so geil wie eclipse mit zb themes und code folding

5.2 Design Decisions

notizen:

beim loggen mit rtt gehen gerne pakete verloren. als lösung uart nutzen und mit putty den log lesen.

problem: das debug terminal zeigt den log unsichtbar an. lösung: nutze ses v4.12: https://devzone.nordicsemi.com/f/nordic-q-a/45985/nrf_log-not-working-on-segger-embedded-studio

problem: entgegen dem datenblatt nimmt die funktion 'lsm6dsl_fifo_raw_data_get' eine 8-bit länge, womit nicht die 4kb der fifo direkt gelesen werden können. lösung: die implementierung der funktion ruft nur die funktion 'lsm6dsl_read_reg' auf, welche eine 16-bit länge nimmt. deswegen kann man hier einfach 'lsm6dsl_read_reg' mit 16-bit nutzen.

problem: spi maximal 255 bytes lesen in einer transaction wegen 8bit buffer. lösungen: - größerer chip hat 16bit buffer, dafür mehr gesamtenergieaufnahme - buffer manuell im sdk auf 16bit stellen funktioniert, ist aber gegen die hardwarespezifikation aus dem datenblatt - in einer übertragung mehrere transactions, was viel mehr codeaufwand bedeutet - mehrere transactions hintereinander

5.3 Interaction of Components

5.4 Summary



6 Evaluation

6.1 Goal and Methodology

ziel ist lange batterielebensdauer durch geringe stromaufnahme. mind ein monat bei täglichem gebrauch wäre top.

vielleicht ein paper zur erforderlichen sampling rate, dass gesten korrekt erkannt werden.
wenn noch ganz viel zeit ist, kann man noch die befestigung mit einer umfrage evaluieren

6.2 Evaluation Setup

6.3 Evaluation Results

das hier ist der stromverbrauch. das hier ist die erreichte sampling rate. so skaliert das mit der anzahl der sensoren

6.4 Analysis of Results



7 Conclusions

7.1 Summary

7.2 Contributions

7.3 Future Work

andere unkonventionelle arten zur datenübertragung untersuchen. durch vibrierende knochen reden.
btmesh implementieren und vergleichen, wie stark der energieverbrauch sich ändert. es gibt da die
library von nordic
nRF52811
eInk um verbindung anzuzeigen

7.4 Final Remarks



Literaturverzeichnis

- [ABCO12] Fatemeh Abyarjoo, Armando Barreto, Jonathan Cofino, and Francisco Ortega. Implementing a sensor fusion algorithm for 3d orientation detection with inertial/magnetic sensors. 01 2012.
- [aco17a] aconno. *Datenblatt ACN52832*, August 2017. Online erhältlich unter https://1897079276.rsc.cdn77.org/wp-content/uploads/2018/08/ACN52832_data_sheet_v1.3.pdf; abgerufen am 09. Juli 2019.
- [aco17b] aconno. *Datenblatt ACNSENSEA*, November 2017. Online erhältlich unter https://1897079276.rsc.cdn77.org/wp-content/uploads/2018/08/ACNSENSEA_datasheet.pdf; abgerufen am 07. Juli 2019.
- [Ada19] Adafruit Industries. Introduction to Bluetooth Low Energy, März 2019. Online erhältlich unter <https://cdn-learn.adafruit.com/downloads/pdf/introduction-to-bluetooth-low-energy.pdf?timestamp=1562415841>; abgerufen am 06. Juli 2019.
- [ARD] ARDUINO. ARDUINO PRIMO CORE. Online erhältlich unter <https://store.arduino.cc/arduino-primo-core>; abgerufen am 09. Juli 2019.
- [Ard16] Arduino. *Datenblatt Primo Core*, September 2016. Online erhältlich unter https://www.arduino.cc/en/uploads/Main/ARDUINO_PRIMO_CORE_V02_SCH.pdf; abgerufen am 09. Juli 2019.
- [Ben19] Bendel, Prof. Dr. Oliver. Wearables, Januar 2019. Online erhältlich unter <https://wirtschaftslexikon.gabler.de/definition/wearables-54088/version-368816>; abgerufen am 21. April 2019.
- [Blu] Bluetooth SIG. Qualified Mesh Products. Online erhältlich unter <https://www.bluetooth.com/bluetooth-technology/topology-options/le-mesh/mesh-qualified/>; abgerufen am 01. März 2019.
- [Bos] Bosch. Produktseite BMI260. Online erhältlich unter https://www.bosch-sensortec.com/bst/products/all_products/bmi260; abgerufen am 09. Juli 2019.
- [Bos18a] Bosch. *Datenblatt BMI160*, Oktober 2018. Online erhältlich unter https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMI160-DS000.pdf; abgerufen am 09. Juli 2019.
- [Bos18b] Bosch. *Datenblatt BMP280*, Januar 2018. Online erhältlich unter https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP280-DS001.pdf; abgerufen am 09. Juli 2019.
- [Bos19] Bosch. *Datenblatt BMM150*, April 2019. Online erhältlich unter https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMM150-DS001.pdf; abgerufen am 09. Juli 2019.
- [Cho] Chordata. Motion Capture system that you can build yourself. Online erhältlich unter <https://hackaday.io/project/27519-motion-capture-system-that-you-can-build-yourself>; abgerufen am 09. Juli 2019.

-
- [dig] digikey.com. TMP007 Infrared Thermopile Sensor. Online erhältlich unter <https://www.digikey.com/en/product-highlight/t/texas-instruments/tmp007-infrared-thermopile-sensor>; abgerufen am 09. Juli 2019.
- [Ele19a] Elektronik-Kompodium.de. DMS - Dehnungsmessstreifen, Juni 2019. Online erhältlich unter <https://www.elektronik-kompodium.de/sites/bau/1404151.htm>; abgerufen am 30. Juni 2019.
- [Ele19b] Elektronik-Kompodium.de. Lithium-Ionen-Akkus, Juni 2019. Online erhältlich unter <https://www.elektronik-kompodium.de/sites/bau/0810281.htm>; abgerufen am 07. Juli 2019.
- [Ele19c] Elektronik-Kompodium.de. MEMS - Micro-Electro-Mechanical Systems, Juni 2019. Online erhältlich unter <https://www.elektronik-kompodium.de/sites/bau/1503041.htm>; abgerufen am 30. Juni 2019.
- [Int19] International Data Corporation. Absatz von Wearables weltweit in den Jahren 2014 bis 2018 (in Millionen Stück), März 2019. Online erhältlich unter <https://de.statista.com/statistik/daten/studie/515723/umfrage/absatz-von-wearables-weltweit>; abgerufen am 21. April 2019.
- [Inv16] InvenSense. *Datenblatt MPU-9250*, Juni 2016. Online erhältlich unter <https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>; abgerufen am 09. Juli 2019.
- [Lai19] Laird. *Datenblatt BL651*, Februar 2019. Online erhältlich unter https://connectivity-staging.s3.us-east-2.amazonaws.com/2019-02/CS-DS-BL651%20%20v1_1.pdf; abgerufen am 09. Juli 2019.
- [mad19] madcoffee auf home-automation-community.com. ARDUINO LOW POWER - HOW TO RUN ATMEGA328P FOR A YEAR ON COIN CELL BATTERY, Februar 2019. Online erhältlich unter <http://www.home-automation-community.com/arduino-low-power-how-to-run-atmega328p-for-a-year-on-coin-cell-battery/>; abgerufen am 09. Juli 2019.
- [mic] microcontroller.net. Serial Peripheral Interface. Online erhältlich unter https://www.mikrocontroller.net/articles/Serial_Peripheral_Interface; abgerufen am 07. Juli 2019.
- [mou] mouser.com. acnSENSA. Online erhältlich unter <https://www.mouser.de/ProductDetail/aconno/acnSENSA?qs=sGAEpiMZZMve4%2FbfQkoj%252BGMseiRPTSjeOSSloCsufA0%3D>; abgerufen am 08. Juli 2019.
- [MT12] K. Mikhaylov and J. Tervonen. Evaluation of power efficiency for digital serial interfaces of microcontrollers. In *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, May 2012.
- [Nor] Nordic Semiconductor. Explore the nRF52 Series. Online erhältlich unter <https://www.nordicsemi.com/Products/Low-power-short-range-wireless>; abgerufen am 09. Juli 2019.
- [Nor16] Nordic Semiconductor. Bluetooth low energy Characteristics, a beginner's tutorial, März 2016. Online erhältlich unter <https://devzone.nordicsemi.com/nordic/short-range-guides/b/bluetooth-low-energy/posts/ble-characteristics-a-beginners-tutorial>; abgerufen am 06. Juli 2019.

-
- [NXP14] NXP Semiconductors. *I2C-bus specification and user manual*, April 2014. Online erhältlich unter <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>; abgerufen am 07. Juli 2019.
- [PPLA17] G. D. Putra, A. R. Pratama, A. Lazovik, and M. Aiello. Comparison of energy consumption in wi-fi and bluetooth communication in a smart building. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–6, Jan 2017.
- [sen18] sensoren.info. MEMS - Micro-Electro-Mechanical Systems, Januar 2018. Online erhältlich unter <https://www.sensoren.info/mikromechBesch1.png>; abgerufen am 30. Juni 2019.
- [Spa] Sparkfun. Reducing Arduino Power Consumption. Online erhältlich unter <https://learn.sparkfun.com/tutorials/reducing-arduino-power-consumption/all>; abgerufen am 09. Juli 2019.
- [Spa16] SparkFun. *Datenblatt nRF52832 Breakout*, August 2016. Online erhältlich unter https://cdn.sparkfun.com/assets/learn_tutorials/5/4/9/sparkfun-nrf52832-breakout-schematic-v10.pdf; abgerufen am 09. Juli 2019.
- [STM15] STMicroelectronics. *Datenblatt LSM9DS1*, März 2015. Online erhältlich unter <https://www.st.com/resource/en/datasheet/lsm9ds1.pdf>; abgerufen am 09. Juli 2019.
- [STM16] STMicroelectronics. *Datenblatt HTS221*, August 2016. Online erhältlich unter <https://www.st.com/resource/en/datasheet/hts221.pdf>; abgerufen am 09. Juli 2019.
- [STM17a] STMicroelectronics. *Datenblatt LIS3MDL*, Mai 2017. Online erhältlich unter <https://www.st.com/resource/en/datasheet/lis3mdl.pdf>; abgerufen am 09. Juli 2019.
- [STM17b] STMicroelectronics. *Datenblatt LSM6DS3*, August 2017. Online erhältlich unter <https://www.st.com/resource/en/datasheet/lsm6ds3.pdf>; abgerufen am 07. Juli 2019.
- [STM17c] STMicroelectronics. *Datenblatt LSM6DSL*, September 2017. Online erhältlich unter <https://www.st.com/resource/en/datasheet/lsm6dsl.pdf>; abgerufen am 09. Juli 2019.
- [Tex] Texas Instruments. CC2650STK. Online erhältlich unter <https://www.ti.com/tool/CC2650STK>; abgerufen am 07. Juli 2019.
- [Tex15] Texas Instruments. *Datenblatt TPS2291xx*, November 2015. Online erhältlich unter <https://www.ti.com/lit/ds/symlink/tps22913.pdf>; abgerufen am 09. Juli 2019.
- [Tex16a] Texas Instruments. *Datenblatt HDC1000*, Januar 2016. Online erhältlich unter <https://www.ti.com/lit/ds/symlink/hdc1000.pdf>; abgerufen am 09. Juli 2019.
- [Tex16b] Texas Instruments. Generic Access Profile (GAP), 2016. Online erhältlich unter http://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_40_00_45/docs/blestack/ble_user_guide/html/ble-stack-3.x/gap.html#; abgerufen am 06. Juli 2019.
- [Tex17] Texas Instruments. *Datenblatt OPT3001*, November 2017. Online erhältlich unter <https://www.ti.com/lit/ds/symlink/opt3001.pdf>; abgerufen am 09. Juli 2019.
- [Var14] Varta. *Datenblatt DS6450*, September 2014. Online erhältlich unter https://products.varta-microbattery.com/applications/MB_DATA/DOCUMENTS/DATA_SHEETS/DS6450.pdf; abgerufen am 07. Juli 2019.

-
- [VGC19] VGChartz. Verkaufszahlen der weltweit meistverkauften Spielkonsolen bis Dezember 2018 (in Millionen Stück), März 2019. Online erhältlich unter <https://de.statista.com/statistik/daten/studie/160549/umfrage/anzahl-der-weltweit-verkauften-spielkonsolen-nach-konsolentypen/>; abgerufen am 21. April 2019. Bearbeitet.
- [Vis15] Vishay. *Datenblatt sip32401a*, Juni 2015. Online erhältlich unter <https://www.vishay.com/docs/63705/sip32401a.pdf>; abgerufen am 07. Juli 2019.
- [VPdN⁺10] Benedetto Vigna, Fabio Pasolini, Roberto de Nuccio, Macro Capovilla, Luciano Prandi, and Fabio Biganzoli. Low cost silicon coriolis' gyroscope paves the way to consumer imu. In Evgeni Gusev, Eric Garfunkel, and Arthur Dideikin, editors, *Advanced Materials and Technologies for Micro/Nano-Devices, Sensors and Actuators*, pages 67–74, Dordrecht, 2010. Springer Netherlands.
- [ZJM17] Longhao Zou, Ali Javed, and Gabriel-Miro Muntean. Smart mobile device power consumption measurement for video streaming in wireless environments: Wifi vs. lte. pages 1–6, 06 2017.