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-type-number

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-type-number

Eingereicht von Pascal Dornfeld
Tag der Einreichung: 23. Juli 2019

Gutachter: Prof. Dr.-Ing. Ralf Steinmetz
Betreuer: Philipp Müller

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

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.

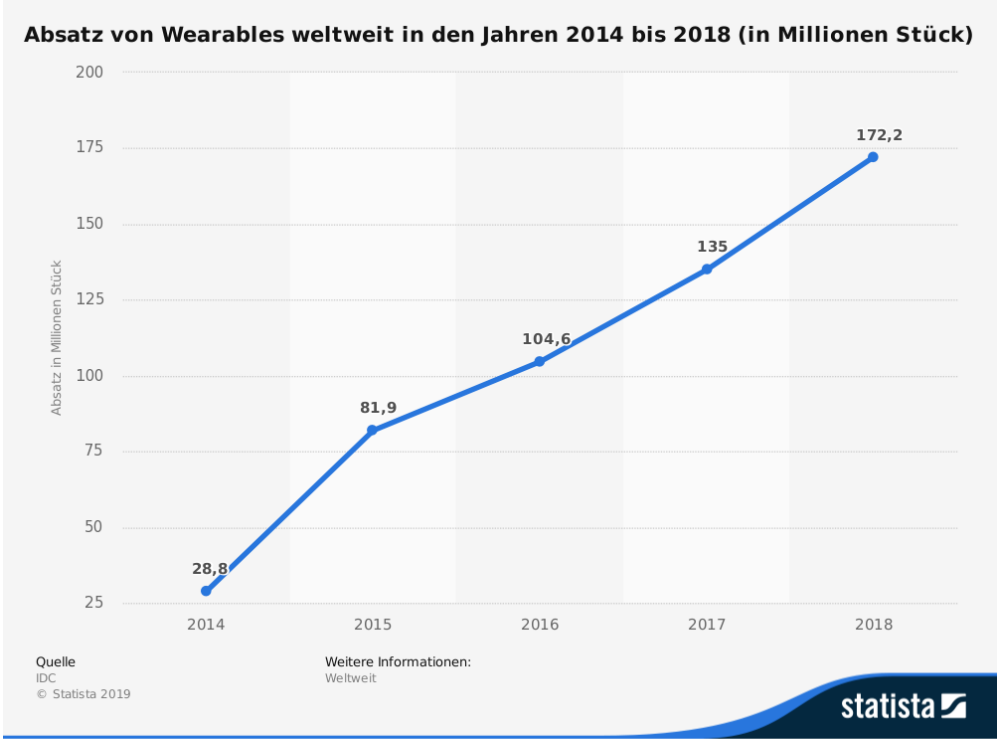

Abbildung 1.1: Absatz von Wearables [Int19]

In dieser Arbeit hingegen wird ein Wearable entworfen, dass 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 Videospielkonsolen 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.

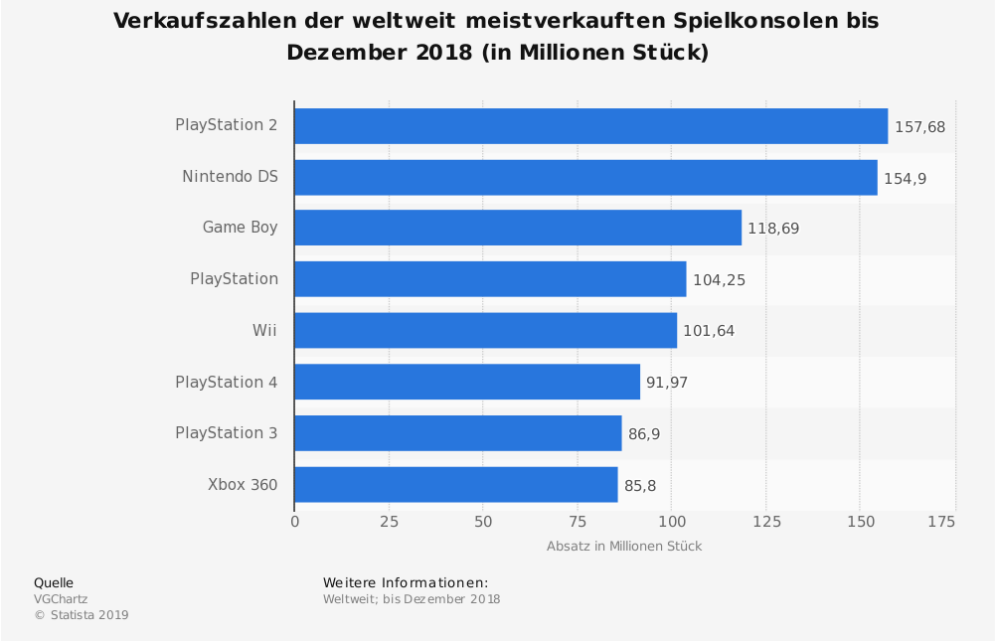

Abbildung 1.2: Verkaufszahlen der weltweit meistverkauften Spielkonsolen [VGC19]

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 wie sie miteinander interagieren sollen. In der Implementation wird die Beschaltung beschrieben und die Probleme die bei der Implementierung aufgekommen sind benannt. Zum Schluss wird das System anhand der genannten Zielen 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 getrennt erhältlich oder zusammen in einer Inertialen Messeinheit (IMU). 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.[1]

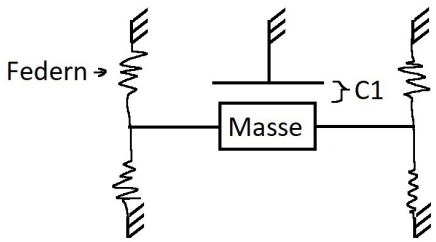

Abbildung 2.1: Kapazitiver Beschleunigungssensor. Basierend auf `https://www.sensoren.info/mikromechBeschl.png`, aufgerufen am 30.06.2019

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.[2]

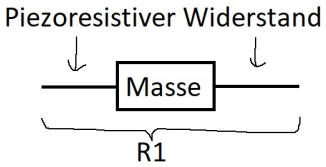

Abbildung 2.2: Piezoresistiver Beschleunigungssensor. Basierend auf `https://www.sensoren.info/mikromechBeschl.png`, aufgerufen am 30.06.2019

[1] `https://www.elektronik-kompendium.de/sites/bau/1503041.htm`, aufgerufen am 30.06.2019
[2] `https://www.elektronik-kompendium.de/sites/bau/1404151.htm`, aufgerufen am 30.06.2019

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 Dieser von außen zum Schwingen gebracht. Dieses Konstrukt gibt es ein zweites Mal in die andere Richtung schwingend. Bei einer Drehung bewegen sich die Masse 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]

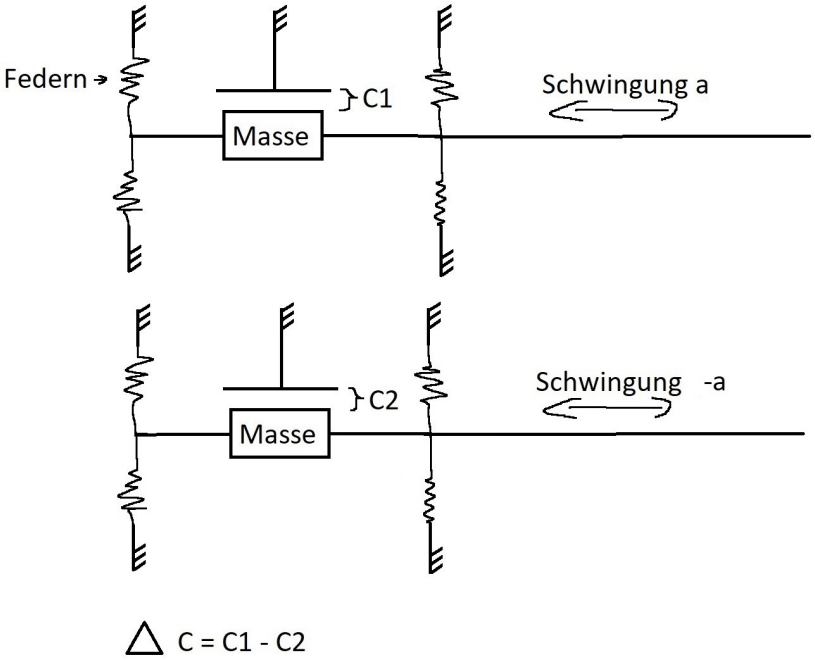

Abbildung 2.3: Gyrosensor

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 über mit z.B. USB ist auch ein weit verbreiteter Anschluss 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 vergleichen 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"[3] [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.

[3] Übersetzung durch den Verfasser

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 gleichkommen. 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.[4]
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.[5]
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.[6]

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 Protokollen wählen. Die verwendeten IMUs unterstützen die weit verbreiteten Protokolle 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 Protokollfrequenz 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 Protokolle effizient abarbeiten kann.
In [MT12] wurde ermittelt, dass I2C einen doppelt so hohen Energieverbrauch wie SPI hat. Da die Protokolle nur vorschreiben wie Daten gesendet werden, aber die Informationen implementierungsabhängig ineffizient verpackt werden können, wurden beide Protokolle mit dem Wearable getestet.

2.3.1 SPI

2.3.2 I2C

[4] https://cdn-learn.adafruit.com/downloads/pdf/introduction-to-bluetooth-low-energy.pdf?timestamp=1562415841, aufgerufen am 06.07.2019

[5] 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#, aufgerufen am 06.07.2019

[6] https://devzone.nordicsemi.com/nordic/short-range-guides/b/bluetooth-low-energy/posts/ble-characteristics-a-beginners-tutorial, aufgerufen am 06.07.2019

2.4 Energieversorgung

nickel und lithium. akku und batterie.
was ist das alles. kleiner vergleich. was wählen wir...

2.5 Verfügbare fertige Einheiten

z.b. TI CC2650STK
blabla datenblatt blabla größe blabla überladen blabla überteuert blabla standbystromverbrauch blabla "no longer available for shipment to Europe"ggwp

oder ACNSENSA
blabla datenblatt blabla standbystromverbrauch blabla eingeschränkte verfügbarkeit (datenblatt-links alle down, wtf)

3 Design / Concept

3.1 Requirements and Assumptions

wir brauchen irgendwas zum rechnen, irgendwas mit bluetooth, irgendwelche sensoren, irgendeine stromversorgung, buttons/led oder so für user, programm und sinnvolle schnittstelle auf handy, irgendwas zum befestigen von dem ganzen

3.2 System Overview

3.2.1 Bluetooth MCU

diesdas wurde angeguckt. hat sich rausgestellt, dass bluetooth und rechner in einem am effizientesten ist. das hier sind weitere Kriterien gewesen. am ende kamen die hier infrage und das hier wurde ausgewählt, weil.

3.2.2 Sensoren

gab entweder accelerometer und gyro getrennt, oder alles in einem als imu. am ende das hier ausgesucht, weil.

3.2.3 Stromversorgung

gibt diese techniken und diese formfaktoren. das hier passt am besten.

3.2.4 Android Schnittstelle

das hier wäre ganz cool als schnittstelle

3.2.5 Befestigung

... das kommt später ...

3.3 Summary

das ist der plan. am ende haben wir dann das da. lets go

4 Implementation

4.1 Architecture

4.1.1 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

4.2 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

4.3 Interaction of Components

4.4 Summary

5 Evaluation

5.1 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

5.2 Evaluation Setup

5.3 Evaluation Results

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

5.4 Analysis of Results

6 Conclusions

6.1 Summary

6.2 Contributions

6.3 Future Work

andere unkonventionelle arten zur datenübertragung untersuchen. durch vibrierende knochen reden.
btmesh implementieren und vergleichen

6.4 Final Remarks

Literaturverzeichnis

[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.

[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.

[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.

[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.

[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.

[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.