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

Titel der Arbeit (Übersetzung)

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

Titel der Arbeit (Übersetzung)

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

1	Einleitung	3
1.1	Motivation	3
1.2	Problem Statement and Contribution	4
1.3	Outline	4
2	State of the Art	5
2.1	Beschleunigungssensor	5
2.2	Gyroskop	5
2.3	Bluetooth Low-Energy	5
2.4	Bluetooth Mesh	5
2.5	Summary	5
2.6	irgendwelche interessanten papers?	5
2.7	Verfügbare fertige Einheiten	5
2.8	Summary	5
3	Design / Concept	7
3.1	Requirements and Assumptions	7
3.2	System Overview	7
3.2.1	Bluetooth MCU	7
3.2.2	Sensoren	7
3.2.3	Stromversorgung	7
3.2.4	Android Schnittstelle	7
3.2.5	Befestigung	7
3.3	Summary	7
4	Implementation	9
4.1	Architecture	9
4.1.1	nRF52832 Software	9
4.2	Design Decisions	9
4.3	Interaction of Components	9
4.4	Summary	9
5	Evaluation	11
5.1	Goal and Methodology	11
5.2	Evaluation Setup	11
5.3	Evaluation Results	11
5.4	Analysis of Results	11
6	Conclusions	13
6.1	Summary	13
6.2	Contributions	13
6.3	Future Work	13
6.4	Final Remarks	13

	Literaturverzeichnis	13
--	-----------------------------	-----------



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.

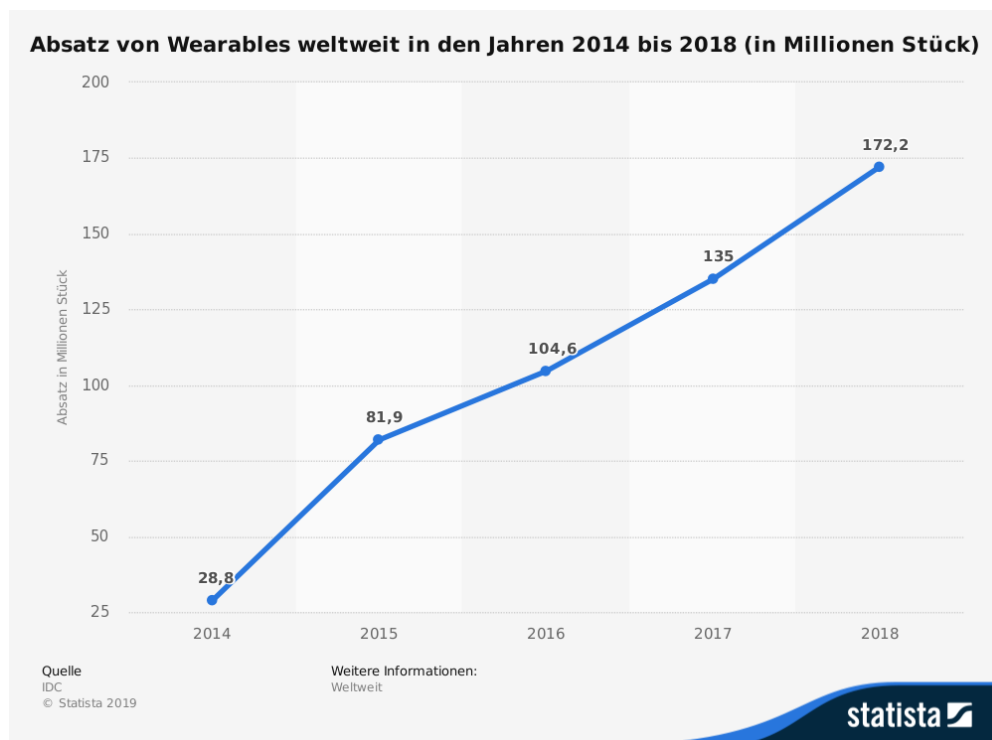


Abbildung 1.1: Absatz von Wearables [Int19]

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 Zeichensprache 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 fünfthelbsteverkaufte Konsole ist und damit die Geschichte der Videospielkonsolen prägt. Die Fernbedienung übernimmt bei ihr die gleichen Funktionen wie das hier entwickelte Wearable. Zusätzlich kann man mit dem Nunchuk einen zweiten Bewegungssensor anschließen. Das

hier entwickelte Wearable könnte dieses System erweitern und weitere Vorteile bieten, zum Beispiel, dass die Hände beim Spielen frei bleiben. Bei VR-Headset werden neben Fernbedienungen wie bei der Wii auch Kameras zur Ganzkörper-Bewegungserkennung genutzt.

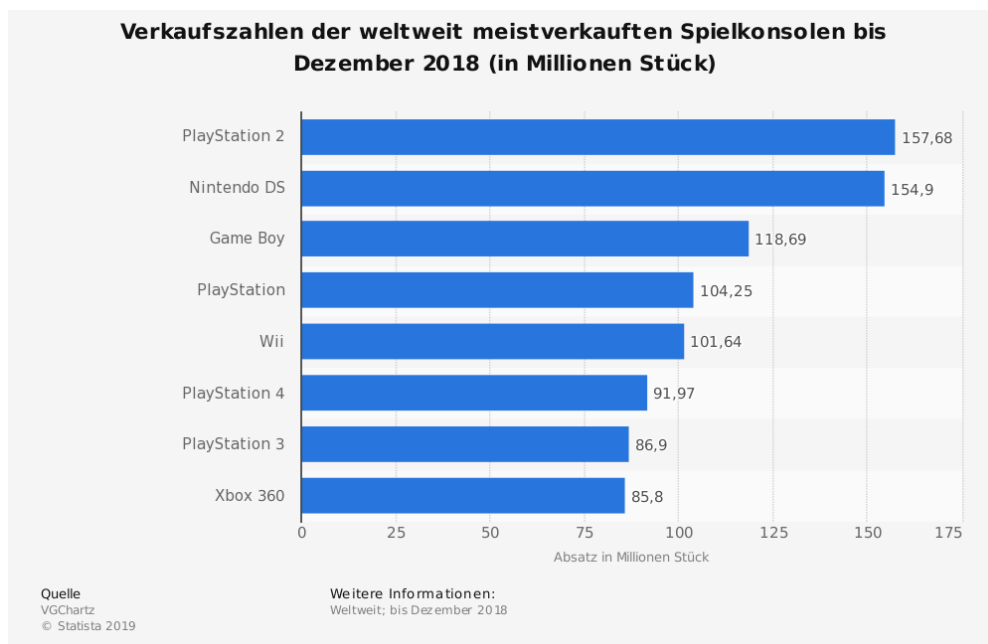


Abbildung 1.2: Verkaufszahlen der weltweit meistverkauften Spielkonsolen [VGC19]

1.2 Problem Statement and Contribution

blabla welche möglichkeiten blabla welche technik/protokolle ausgewählt blabla

1.3 Outline

blabla struktur

2 State of the Art

2.1 Beschleunigungssensor

blabla funktionsweise blabla physik blabla elektrotechnik
diesdas <https://www.youtube.com/watch?v=eqZgxR6eRjo>

2.2 Gyroskop

blabla funktionsweise blabla physik blabla elektrotechnik

2.3 Bluetooth Low-Energy

blabla funktionsweise blabla protokoll

2.4 Bluetooth Mesh

blabla funktionsweise blabla protokoll

2.5 Summary

2.6 irgendwelche interessanten papers?

irgendwas richtung "gesture detection system"?

2.7 Verfügbare fertige Einheiten

z.b. TI CC2650STK
blabla datenblatt blabla gröÙe 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)

2.8 Summary

blabla alle kacke, perfekt gibts nicht



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

dieses 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: 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

es ist perfekt. mit allem anderen gebe ich mich nicht zufrieden!

6.2 Contributions

6.3 Future Work

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.
- [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.