



---

Bachelorarbeit

**Entwicklung eines Frameworks  
zur Darstellung von  
Smartphone-Sensordaten für die  
didaktische Unterstützung von  
Programmiervorlesungen**

---

Marius Cerwenetz

Abschlussarbeit

zur Erlangung des akademischen Grades

Bachelor of Science

Vorgelegt von	Marius Cerwenetz
am	XX. Juli 2022
Referent	Prof. Dr. Peter Barth
Korreferent	Prof. Dr. Jens-Matthias Bohli

## **Schriftliche Versicherung laut Studien- und Prüfungsordnung**

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mannheim, XX. Juli 2022

---

Marius Cerwenetz

## **Zusammenfassung**

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
<b>2</b>	<b>Experimente/Aufgaben</b>	<b>3</b>
<b>3</b>	<b>Einführung</b>	<b>6</b>
<b>4</b>	<b>Diskussion und Fazit</b>	<b>8</b>
<b>5</b>	<b>Ausblick</b>	<b>9</b>
	<b>Literaturverzeichnis und Online-Quellen</b>	<b>9</b>

# Kapitel 1

## Einführung

MINT-Berufe leiden hierzulande unter einem akuten Fachkräftemangel. Das Institut der deutschen Wirtschaft ermittelte für April 2021 ein Unterangebot von 145.100 Personen [1] in 36 MINT-Berufskategorien. Digitalisierungs-Projekte geraten dadurch ins Stocken.

Nicht zuletzt er auch ein Kräftemangel in der Softwareentwicklung. Es fehlen Programmiererinnen und Programmierer. Softwareentwicklung ist gerade in der Lernphase nicht trivial und abstrakt. Unlebendige Übungsaufgaben die beispielsweise Konsolenein- und ausgaben realisieren schrecken zukünftige Programmierinnen und Programmierer eher ab als sie zu ermutigen.

Microcontroller sind bereits eine große Hilfe, da hier spielerisch kleine Projekte realisiert werden können. So können schon früh in Schulen Kinder an die Programmierung herangeführt werden. Sie lernen spielerisch kleine Programme zu entwickeln und verstehen die ihnen beigebrachten Abläufe durch schnelle Anwendung. Microcontroller sind jedoch auch mit Anschaffungskosten verbunden und für kleine Anwendungen, welche Sensoren verwenden, wird viel zusätzliches Material wie zum Beispiel Breadboards, Verbindungskabel und Erweiterungsboards benötigt. Moderne Smartphones bieten hier Abhilfe da Sie meistens mit verschiedenen Sensoren bespickt sind, wie zum Beispiel: Kompass, GPS, Microphon oder Kamera. Viele Kinder besitzen bereits mit 10 Jahren [2] ein Smartphone.

Im Rahmen dieser Arbeit soll ein Framework entwickelt werden, dass das Smartphone von Anwendern einbindet um Sie beim Programmierenlernen zu unterstützen. Dies soll in Form von Aktivitäten möglich sein. Aktivitäten starten entweder UIs auf dem Smartphone um eine Eingabe entgegenzunehmen oder senden konstant Sensordaten.

## Kapitel 2

# Experimente/Aufgaben

### Aufgaben

Dieses Kapitel enthält verschiedene Beispielaufgaben die mit dem Framework gelöst werden sollen. Die Benutzung der API dazu wird später geschildert.

#### Disco

Die LED muss ganz schnell blinken. Ablauf: `led_on`, `sleep`, `led_off` ...

#### Diebstahl-Alarm

Wenn nach dem Telefon gegriffen wird soll die LED blinken. Wenn man sich davon entfernt soll sie wieder aus sein.

#### Würfeln

Das Smartphone wird geschüttelt. Ein random Zahlenwert wird zurückgegeben. Je nachdem welches Ergebnis zurückkommt soll ein Wert auf dem Display angezeigt werden.

Ablauf: `get_random()`, `displayText`

#### Klatsch-Zähler

Der Anwender möchte wissen wie oft in einem bestimmten Zeitraum geklatscht wurde. Ein Methodenaufruf startet in der Androidanwendung eine Activity, die innerhalb des im Argument genannten Zeitraums die Anzahl der maximalen Lautstärkeamplituden des Mikrophons misst und die Anzahl zurückgibt.

#### Dreh-Zähler

Ein Nutzer möchte in einem bestimmten Zeitraum zählen wie oft das Smartphone gedreht wurde.

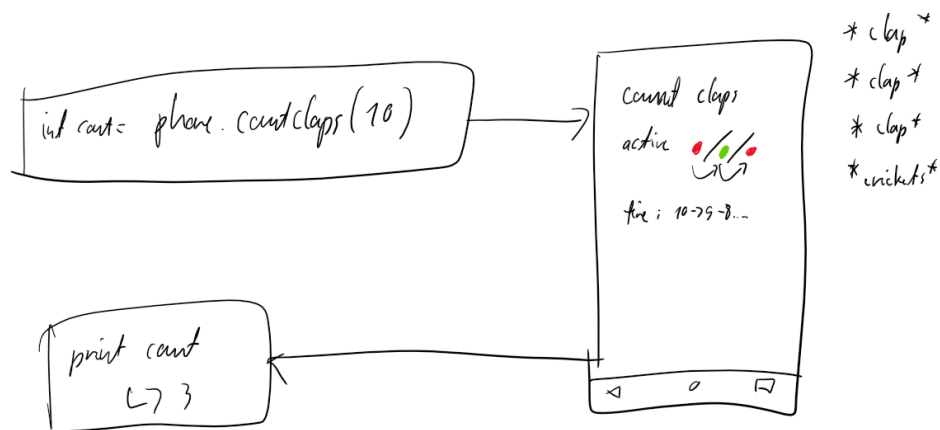


Abbildung 2.1: Klatsch-Zähler

## API

Gelöst werden sollen die Aufgaben durch das Aufrufen der API. Diese bietet die benötigten Funktionen an. Die Aufrufe sind frei miteinander kombinierbar, so dass Aufgaben erweitert werden können.

## Eingaben

### Auslesen der accelerometer-Daten

Ein User will den Wert der X, Y und Z Koordinaten des Smartphones wissen. So kann er bspw. feststellen, ob das Smartphone gerade nach unten, oben oder horizontal bewegt wurde. Kippbewegungen werden nicht detektiert.

```
1 Phone p;
2 float x_val = p.getX();
```

### Lautstärkepegelmessung

Ein User möchte den aktuellen Lautstärkepegel messen.

```
1 Phone p;
2 float volume = p.getVolume();
```

### Annäherungssensor-Messung

Ein User möchte wissen, ob ein Objekt unmittelbar vor den Annäherungssensor steht. Der Aufruf erfolgt folgendermaßen.

```
1 Phone p;
2 boolean res = p.proximity_triggered();
```

## Ausgaben

### Display-Ausgabe

Um auf dem Smartphone einen beliebigen Text anzuzeigen. Dies kann er zum Beispiel wie im folgenden Beispielcode machen.

```
1 Phone p;  
2 p.displayText("Hallo Welt");
```

### LED leuchten lassen

Ein User möchte eine LED auf dem Display ansteuern. Dies kann er so machen.

```
1 Phone p;  
2 p.led_on();  
3 p.led_off();
```

### Vibrationsauslöser

Ein Nutzer möchte das Smartphone auf Anfrage vibrieren lassen. Der Aufruf einer Methode lässt das Smartphone einmal vibrieren.

```
1 Phone p;  
2 p.vibrate();
```

### Audio-Ausgabe

Ein Nutzer möchte den Pfad zu einer Audio-Datei angeben und

```
1 Phone p;  
2 p.vibrate();
```



## Kapitel 3

# Einführung

### Nachrichtenformate

Die Nachrichten werden im JSON-Format übertragen. Für die Anwendung steht folgender Ausschnitt beispielsweise für das Starten einer bestimmten Activity.

```
1  {  
2    "issue_id" : 0,  
3    "type" : activity_request,  
4    "value" : "boolean_activity",  
5    "arguments" : []  
6  }
```

Das Type-Feld beschreibt, ob es sich bei der Nachricht um ein request oder um eine response handelt. Dieses Feld kann die folgenden Werte annehmen:

- activity\_request
- activity\_response

Ein Activity-Request fordert das Starten einer Activity mit der der Nutzer interagieren muss. Je nach Request-Art werden entweder Sensor-Responses oder Activity-Responses zurückgesendet.

### Android Anwendung

Die Android-Anwendung läuft auf dem Smartphone des Anwenders. Sie hat ein sehr schlichtes äußeres Erscheinungsbild und reagiert nur auf Anfragen der Middleware. Sie besteht aus einer Haupt-Activity und mehreren Unter-Activites die gegebenenfalls

aufgerufen werden.

In der Haupt-Activity wird ein Service gestartet und gebunden. Der Service baut eine Verbindung zu einem MQTT Topic auf einem MQTT-Server auf, reagiert auf Nachrichten und versendet gegebenenfalls selbst Nachrichten. Geht eine Nachricht ein die eine User-Interaktion verlangt wird die jeweilige Activity gestartet. Der User interagiert mit der Activity und bestätigt seine Eingabe. Die Activity bindet daraufhin den MQTT-Service und versendet die Antwort an die Middleware, wo Sie an die Bibliothek und somit an die Endanwendung auf dem PC des Users gesendet wird.

Geht eine Nachricht ein, die die Übermittlung von Sensordaten verlangt wird keine eigene Aktivität gestartet. Der Service startet in diesem Fall selbst den Ausleseprozess und übermittelt dann, je nach Anfrage, einen oder mehrere Werte. Die Verfügbarkeit der jeweiligen Sensoren wird beim Start der Anwendung ermittelt. Falls ein Sensor angefragt wird der auf dem Smartphone nicht existiert wird eine Nachricht mit einer Fehlermeldung zurückgegeben.

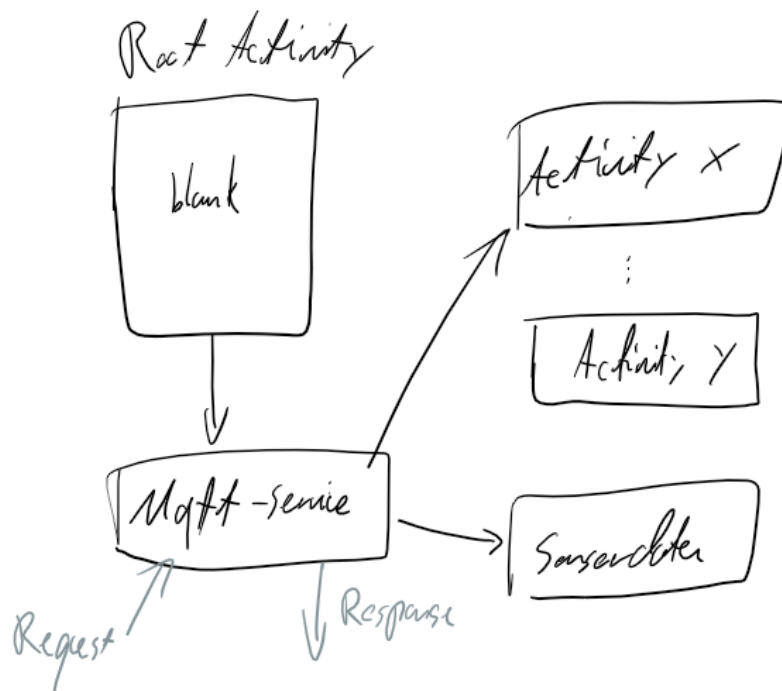


Abbildung 3.1: Android-App-Aufbau

## **Kapitel 4**

# **Diskussion und Fazit**

# Kapitel 5

## Ausblick

### Literaturverzeichnis

- [1] Koppel Oliver Anger Christina, Kohlisch Enno. Mint-frühjahrsreport 2021. mint-engpässe und corona-pandemie: von den konjunkturellen zu den strukturellen herausforderungen, gutachten für bda, bdi, mint zukunft schaffen und gesamtmittel, köln. <https://www.iwkoeln.de/studien/christina-anger-enno-kohlisch-oliver-koppel-axel-pluennecke-mint-engpaesse-und-corona-pandemie-von-den-konjunkturellen-zu-den-strukturellen-herausforderungen.html>, 2021.

### Online-Quellen

- [2] bitkom. <https://www.bitkom.org/Presse/Presseinformation/Mit-10-Jahren-haben-die-meisten-Kinder-ein-eigenes-Smartphone>. [letzter Zugriff: 07. April. 2022].