hochschule mannheim

Fakultät Informationstechnik
Studiengang Technische Informatik

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

Programmierenlernen fällt besonders am Anfang schwer. Embeddedprojekte erlauben mit vergleichsweise wenig Aufwand einen gelungen Einstieg mit effektiver Lernerfahrung. Solche Projekte benötigen allerdings viel Peripherie und Hardware. Diese benötigt wiederrum eine nicht niedrigschwellige Erfahrung zum Beispiel im Umgang mit Microcontrollern. Smartphones haben diese Nachteile nicht bieten allerdings trotzdem einen hohen Umfang an Sensoren.
In dieser Arbeit wird ein Framework erstellt, dass das Smartphone nutzt um mit dem Microcontroller kleine Softwareprojekte umzusetzen. Kleine Aufgabenstellungen mit Musterlösungen werden ausgearbeitet und mitgereicht.

Inhaltsverzeichnis

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 herangerführt werden. Sie lernen spielerisch kleine Programme zu entwickleln 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. Dieses orientiert sich ganz an bereits bestehenden System wie Arduino und Microbit. Kleine Interaktionsmöglichkeiten werden geschaffen.

Benötigt werden dafür eine Library zum Einbinden, eine Python-Anwendung und eine mobile Anwendung für Android Smartphones.

Anwendung

Die Anwendung besteht aus einem Textfeld, einer Ausgabe-LED, einer Signal LED für Nutzereingaben und zwei Buttons. Auf dem Textfeld können verschiedene Ausgaben präsentiert werden. Die Ausgbale LED kann vom Computer aus an und ausgeschaltet werden. Die Signal-LED dient dazu dem Nutzer zu signalisieren, ob sich das Smartphone gerade in einem Aufnahmeprozess befindet. Die Buttons werden verwendet um auf dem PC Steuerbefehle auszuführen.
Neben den Ein- und Ausgabemöglichkeiten im UI-Screen kann zur Eingabe auch noch das Microphon und das accelerometer genutzt werden, sofern vorhanden. Als Ausgabemöglichkeiten kommen der Vibrationsmotor sowie der Lautsprecher des Smartphones zum Einsatz.

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.

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.

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. Diese soll im Textfeld angezeigt werden.

Dreh-Zähler
Ein Nutzer möchte in einem bestimmten Zeitraum zählen wie oft das Smartphone

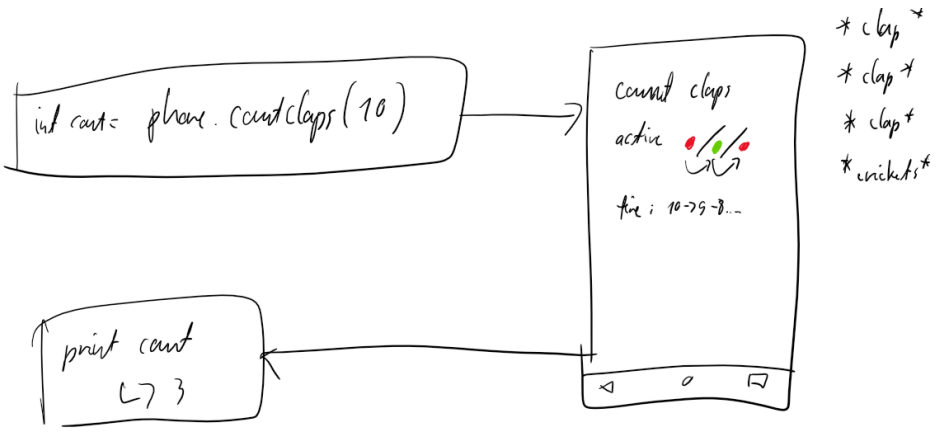

Abbildung 2.1: Klatsch-Zähler

gedreht wurde.

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.

```
Phone p;
bool triggered = p.primity_triggerd();
```

Amplituden-Spike-Messung mit Zeitraum
Ein User möchte wissen, wie oft die Lautstärke innerhalb eines angegebenen Zeitraums t ein gewisses Limit überstiegen hat. Der Aufruf könnte dabei folgendermaßen laufen.

```
Phone p;
int timeframe = 1000; //1000 ms = 1s
int num = p.getNumOfSpikes(timeframe);
```

Umdrehungsmessung mit Zeitraum
Ein User möchte wissen, wie oft das Smartphone innerhalb eines angegebenen Zeitraums t gedreht wurde. Der Aufruf sieht folgendermaßen aus.

```
Phone p;
int timeframe = 1000; //1000 ms = 1s
int num = p.getNumOfSpikes(timeframe);
```

Ausgaben

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

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

LED leuchten lassen
Ein User möchte eine LEDäuf dem Display ansteuern. Dies kann er so machen.

```
Phone p;
p.led_on();
p.led_off();
```

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

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

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

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

Kapitel 3

Architektur

Der Aufbau des Frameworks besteht aus einer mobilen Anwendung für Android-Smartphones, Bibliothek in der Programmiersprache C und einer Middleware die den Austausch Koordiniert.
Clientseitig erfolgen sämtliche Aufrufe die Sensordaten abgreifen oder Steueranfragen senden immer erst per UDP über die Middleware.
Die Middleware sammelt, sobald sich eine Android-Anwendung bei ihr anmeldet, gewisse Sensordaten und speichert Sie intern zwischen, damit Sie schnell vorrätig vorliegen. Die Speicherung erfolgt in einem seperaten Thread, der die letzten Zustände der Sensordaten hält. Werden Sensordaten abgerufen werden Sie aus der Liste entfernt. Sollten mehr als 5 Sensorwerte vorliegen werden die ältesten entfernt um die Ergebnisse aktuell zu halten.
Steueraufrufe werden beim Aufruf über ein seperates Topic versand.

Android Anwendung

Die Android-Anwendung läuft auf dem Smartphone des Anwenders. Sie besteht aus einer Haupt-Activity deren außerlicher Aufbau in 1 beschrieben ist.
In der Haupt-Activity wird ein Service gestartet und gebunden. Der Service baut eine Verbindung zu einem MQTT Server auf. Dort verbindet er sich auf zwei Topics: Einem Sensortopic auf dem mit einer einem QOS-Level 0 alle Sensordaten dauerhauft ausgetauscht werden, die vom Smartphone aus unterstützt werden. Das andere Topic dient zum Aufruf von Steuerbefehlen wie den in 2 erwähnten Funktionsaufrufen, welche mit einem QOS-Level von 2 versendet werden müssen, damit sie unter Garantie ausgeführt werden. Die Activity bindet beim Start den MQTT-Service ein. Stellt ein Client über die Bibliothek eine Anfrage wird dieser Aufruf wie z.B. eine Ausgabe auf einem Textfeld an das Smartphone wird diese Anfrage zuerst zur Middleware geleitet.

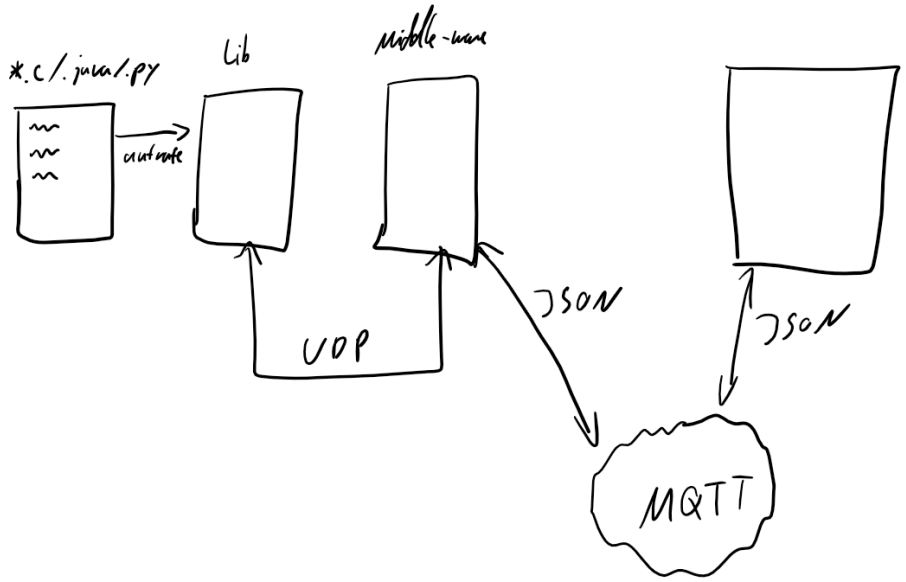

Abbildung 3.1: Android-App-Aufbau

Diese leitet die Anfrage weiter an das Smartphone, dass dann den jeweiligen Befehl ausführt. Dies beeinhaltet vor allem Ausgaben, sowie Eingaben die eine Nutzerinteraktion mit reduziertem Ergebnis. Zum Beispiel alle Eingaben die etwas in einem angegebenen Zeitraum messen. Alle anderen Sensordaten werden dauerhaft an die Middleware übertragen. 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.

Nachrichtenformate

Die Nachrichten werden im JSON-Format übertragen. Nachfolgendes Listing zeigt beispielsweise den Aufruf die LED einzusschalten.

```
1 {
2    "type" : "control_request",
3    "value" : "led on",
4    "arguments" : []
5  }
```

Folgende Nachricht beschreibt beispielsweise den übermittelten Sensorwert des accelerometers.

```
1 {
2    "type" : "acellerometer_x_value",
```

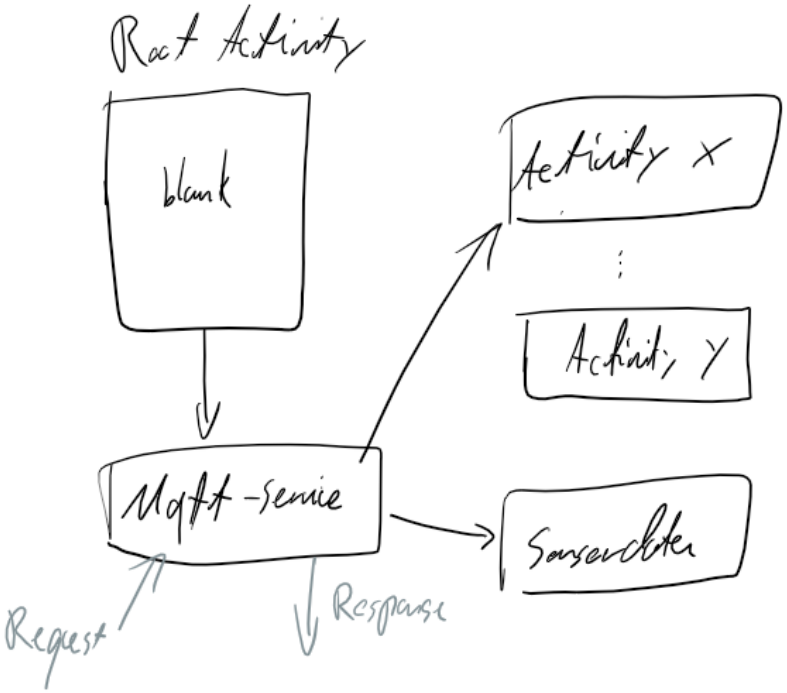

Abbildung 3.2: Android-App-Aufbau

```
3     "value" : "-0.15"
4 }
```

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 gesamtmetall, 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].