



Entwicklung eines Frameworks zur Darstellung von Smartphone-Sensordaten zur didaktischen Unterstützung von Programmierlehrveranstaltungen

Programmierenlernen birgt für Neueinsteiger Schwierigkeiten. Abstrakte Konzepte wie Schleifen, Entscheidungsbäume oder Datenstrukturen sind am Anfang nicht leicht zu verstehen. Die Motivation wird zusätzlich erschwert durch eine beträchtliche Menge an nötiger Vorarbeit um herkömmlich erscheinende Anwendungen zum Beispiel mit einer GUI zu realisieren. Frameworks wie Swing [1], QT [2] oder Tcl/Tk [3] schaffen hier zwar Abhilfe, benötigen jedoch ebenfalls eine gewisse Einarbeitungszeit. Alternativ zu herkömmlichen Anwendungen sind simple Embedded Projekte auf Microcontrollern wie zum Beispiel dem Raspberry Pi [4], Microbit[5] oder Arduino[6] auch ohne graphische Benutzeroberfläche möglich und ermöglichen einen spielerischen Umgang mit dem Programmierenlernen. Diese Lösungen und Projekte sind unterhaltender und motivierender, erfordern jedoch zusätzliche Microcontroller und gegebenenfalls ergänzende Erweiterungsboards für Sensoren oder Schnittstellen. Moderne Smartphones sind bereits mit verschiedensten Sensoren ausgestattet, mobiler als Microcontroller und meistens bei Lernenden bereits vorhanden.

Im Rahmen der Arbeit soll ein Framework entwickelt werden, was die Sensoren von Smartphones ausliest und Sie angehenden Entwicklerinnen und Entwicklern zur weiteren Verarbeitung zugänglich macht. Es soll außerdem Aktionen auf dem Smartphone auslösen die gegebenenfalls eine Interaktion mit dem Nutzer ermöglichen.

Das Framework besteht aus drei Teilen: Einer C/C++ Bibliothek, einer in Python implementierten Kontrollanwendung und einer Androidanwendung. Die Bibliothek stellt Schnittstellen in Form von Objekten und Funktionsaufrufen bereit, die zum Starten von Aktionen auf dem Smartphone dienen. Außerdem sollen über die Schnittstellen spezifizierte Sensordaten zu festeingestellten Raten automatisch ausgelesen werden. Umgesetzt werden die Aufrufe auf dem Smartphone durch eine Kontrollanwendung die sich auf dem PC der Entwicklerin bzw. des Entwicklers befindet. Diese sendet Anweisungen zum Starten von Aktionen an die Anwendung auf dem Smartphone. In der Smartphoneanwendung werden die Prozeduren ausgeführt und gegebenenfalls Sensordaten an die Kontrollanwendung zurückgesendet. Die Kontrollanwendung reicht die Sensordaten an die Bibliothek weiter, so dass Sie der Entwicklerin bzw. dem Entwickler zur weiteren Verarbeitung zur Verfügung stehen. Dieser Vorgang soll über das UDP Protokoll erfolgen, so dass die Bibliothek möglichst wenige Zeilen Code umfasst und schnell in anderen Programmiersprachen wie zum Beispiel Java umgesetzt werden kann. Die Auswahl an Sensordaten umfassen dabei alle üblichen Smartphone-Sensoren wie zum Beispiel Bewegungs-, Positions oder Umgebungssensoren. Eine Übersicht über den Ablauf ist bei Abbildung 1 zu sehen.

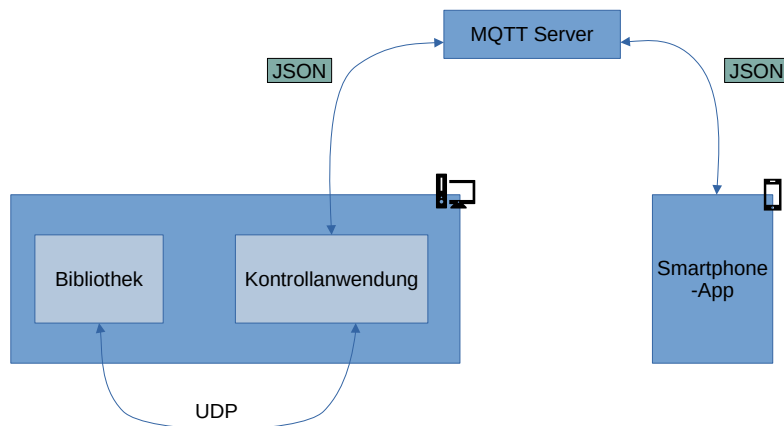


Abbildung 1: Framework-Ablauf

Zur Vermittlung der beiden Geräte wird zum Nachrichtenaustausch zwischen Kontrollanwendung und Smartphoneanwendung ein MQTT-Server verwendet. [7] Der Nachrichtenaustausch erfolgt über das JSON-Format [8] für das geeignete Nachrichtenformate definiert werden. Eine Internetverbindung des PCs und des Smartphones werden somit vorausgesetzt. Die Konfiguration der korrekten MQTT-Topics auf beiden Geräten wird dabei auf dem PC in einer Konfigurationsdatei und in der Androidanwendung intern über das Shared Preferences-Feature festgelegt.

Im Rahmen der Arbeit sollen kleine Demoanwendungsfälle zum Programmierenlernen erarbeitet werden.

Literatur

- [1] Christian Ullenboom. *Java ist auch eine Insel*. [letzter Zugriff: 15. März 2022]. URL: https://openbook.rheinwerk-verlag.de/javainsel/17_001.html.
- [2] The QT Company. *QT*. [letzter Zugriff: 15. März 2022]. URL: <https://www.qt.io/>.
- [3] Tcl Core Team. *Tcl/Tk*. [letzter Zugriff: 15. März 2022]. URL: <https://www.tcl.tk/>.



- [4] Raspberry Pi Foundation. *Raspberry Pi*. [letzter Zugriff: 15. März 2022]. URL: <https://www.raspberrypi.org/>.
- [5] Micro:bit Educational Foundation. *Microbit*. [letzter Zugriff: 15. März 2022]. URL: <https://microbit.org/>.
- [6] Arduino. *Arduino*. [letzter Zugriff: 15. März 2022]. URL: <https://www.arduino.cc/>.
- [7] Eclipse Foundation. *Mosquitto*. [letzter Zugriff: 15. März 2022]. URL: <https://mosquitto.org/>.
- [8] Tim Bray. *RFC 8259*. [letzter Zugriff: 15. März 2022]. URL: <https://datatracker.ietf.org/doc/html/rfc8259>.