Quantified Student

Huawei Link proof-of-concept supplement

Jelle Maas\*

Fontys University of Applied Sciences

Eindhoven, Netherlands

Dated: May 18, 2022

Abstract

The Quantified Student (QS for short) project focuses on helping students with their development and optimizing their performance with the help of collected data. The collected data will be shown in a dashboard where the student can see it. After which, the student can conclude where and how to improve their workflow. For example, the system can show when it is the best time to work for the student.

Table of Contents

[1 Introduction 5](#_Toc103758586)

[2 Goal 5](#_Toc103758587)

[3 Contributions 6](#_Toc103758588)

[3.1 Wiki 6](#_Toc103758589)

[3.2 Proof-of-concept 6](#_Toc103758590)

[4 Version control 7](#_Toc103758591)

[Conclusion 8](#_Toc103758592)

Table of Figures

[Figure 1 Flowchart protocol determination 6](#_Toc102984872)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description** |
| 0.1 | May 18, 2022 | J. Maas | Initial document setup and add goal, contributions and conclusion |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

Quantified Student (QS for short) Watch is a component of the Quantified Student system. The project is intended on the one hand to collect physical data from a student using smartwatches and on the other hand to provide performance insights to a student, generated from that collected data.

To achieve said goal, it is required to have proper communication with a compatible smartwatch. This communication may be used to gather, among other things, stress, and fitness data from signed up students.

The communication is setup with the help from an Android application. This application handles the communication with a compatible smartwatch, which means it can request the mentioned data from the smartwatch. The application acts as the middleman between the smartwatch and data collection endpoint.

# Goal

The goal of this proof-of-concept is to prove that communication between a custom Android application and a Huawei peripheral (smart band) is possible. Additionally, this proving also provides a first step in the development of the smartwatch data source and allows the Quantified Student team to gather biometric data for the platform.

# Contributions

## Wiki

The Huawei Link protocol was reverse engineered by decompiling the original Huawei Health mobile application. The source code was not readable at first because it was obfuscated. Therefore, the code had to be manually de-obfuscated to make it readable again. After that, it was possible to make sense of what actions the original application was taking in communicating with the Huawei peripherals. After this long exploration process, the handshake sequence and some other services have been discovered. The results of the forementioned efforts are documented in the organizational GitHub wiki and can be viewed publicly at <https://github.com/quantifiedstudent/mobile-android/wiki>.

## Proof-of-concept

The proof-of-concept focuses on implementing the handshake sequence of the Huawei Link protocol. The handshake sequence is meant to pair the peripheral with the mobile application and to set up a secure connection between them. The communication with the peripheral is handled by the built-in [Bluetooth GATT](https://developer.android.com/reference/android/bluetooth/BluetoothGatt) services from the Android SDK.

# Version control

During the development of the feature, the changes were pushed to a separate branch called *“poc/huawei\_link\_protocol”*. After the development process was finished, the changes were reviewed by the necessary reviewers via a pull request. When all reviewers approved the contributions, the proposed changes were merged with the main branch of the GitHub repository. The pull request shows all the comments and commits made during the development and can be viewed publicly at <https://github.com/quantifiedstudent/mobile-android/pull/4>.

# Conclusion

The proof-of-concept was a success and proves that the communication between a custom Android application and a Huawei peripheral is possible. With the contributions, the mobile application is now able to initiate handshakes with the Huawei peripheral, which allows the application to send and receive data such as biometrics.

Note that there is still one issue while connecting and pairing with the Huawei peripheral, which sometimes causes instability and results in being unable to gather the biometrics.

The next steps are to resolve the forementioned issues and to extend upon the functionality of the protocol.