Brief Technical Documentation of Master Thesis Project

"Facilitating Technology-based Mental Health Interventions with Mobile Virtual Reality and Wearable Smartwatches"

Use-Case Analyzing Heart Rate Variability during Slow-Breathing Relaxation Exercises

Karolinska Institutet

Stockholm University

Author: **Luis Quintero**
http://luiseduve.github.io

Master's Programme in Health Informatics:
Spring Semester 2019

Thesis project supported by:

mimerse

Table of Contents

1. Introduction

1.1. Purpose

This document contains a brief technical documentation to keep in mind when working with the project: Physiologically Adaptable Relaxation Experience through Virtual Reality (PARE-VR)

The project augments VR applications developed in Unity with access to HR and PPG data from wearable Samsung smartwatches to execute physiologically adaptable behaviors. Consists three applications that are in charge of signal acquisition, signal processing and visualization independently.

1.2. References

Visit the thesis with the information of the project in the following link: THESIS

2. System Description

2.1. Resources

HARDWARE

- Smartphone:	Samsung Galaxy S9
- VR headset:	Samsung Gear VR
- Smartwatch:	Samsung Gear Sport

SOFTWARE

- Virtual Environment: CalmPlace.

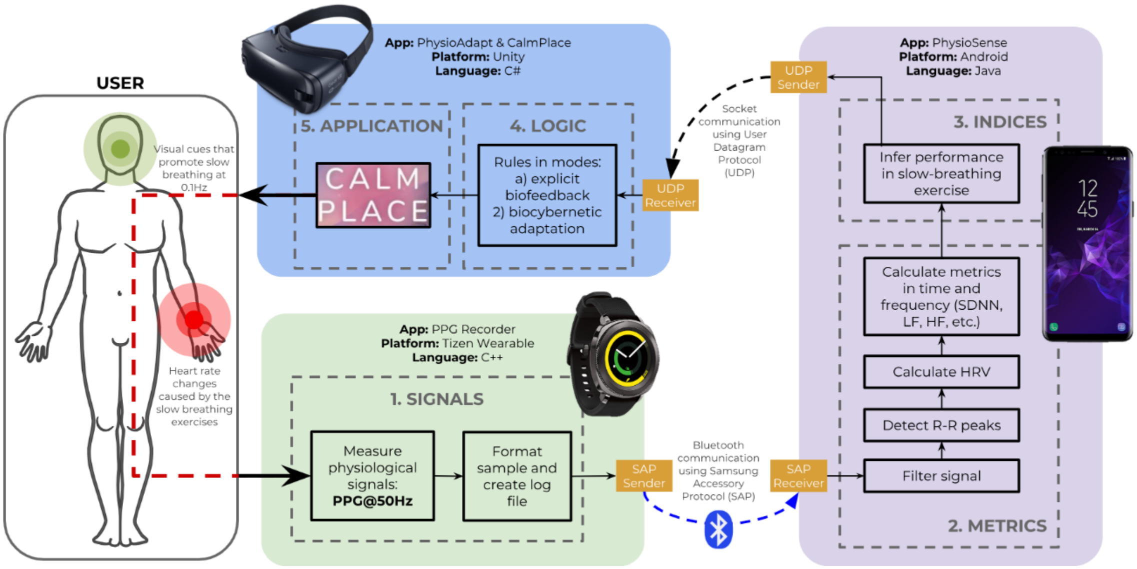
  Relaxation experience developed by Mimerse for anxiety and stress management.

- Main Project : PARE-VR

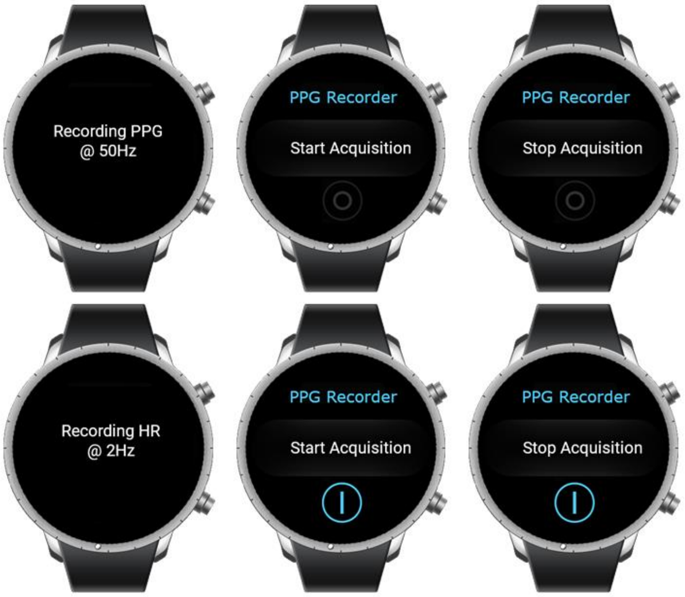
  Files with the developed applications are located in the Github repository: https://github.com/luiseduve/pare-vr

2.2. Overall Architecture

The system is composed by three applications that run independently and communicate physiological data in real-time, such as shown in the image below.


USER
Visual cues that promote slow breathing at 0.1Hz
Heart rate changes caused by the slow breathing exercises
App: PhysioAdapt & CalmPlace
Platform: Unity
Language: C#
5. APPLICATION
CALM PLACE
4. LOGIC
Rules in modes: a) explicit biofeedback 2) biocybernetic adaptation
UDP Receiver
Socket communication using User Datagram Protocol (UDP)
UDP Sender
App: PhysioSense
Platform: Android
Language: Java
3. INDICES
Infer performance in slow-breathing exercise
Calculate metrics in time and frequency (SDNN, LF, HF, etc.)
Calculate HRV
Detect R-R peaks
Filter signal
2. METRICS
App: PPG Recorder
Platform: Tizen Wearable
Language: C++
1. SIGNALS
Measure physiological signals: PPG@50Hz
Format sample and create log file
SAP Sender
Bluetooth communication using Samsung Accessory Protocol (SAP)
SAP Receiver

3. Setup of APP 1: PPG Recorder | Tizen



Application in charge of collecting heart data from the smartwatch. Enable the checkbox to read raw Heart Rate @ 2Hz or disable it to read PPG @ 50Hz and get a complete heart signal to be processed.

3.1. Environment

Application developed in Tizen Studio 3.0 as a Native C application.

3.2. Setup Tizen Studio

Download and setup Tizen Studio from:

https://developer.tizen.org/development/tizen-studio/download

Load the project located in the folder App1.

3.2.1. Companion App

The developed application needs to be paired with a cellphone to work. Therefore, it requires installation of special packages, as detailed in:

https://developer.samsung.com/galaxy-watch/develop/creating-your-first-app/native-companion/setup-sdk

- Open Tools>Package Manager
- In the main SDK tab, install the packet “2.3.2 Wearable”, and
- From the tab Extension SDK install the “Samsung Certificate Extension” and “Samsung Wearable Extension”.

3.3. Setup Smartwatch

Pair the smartwatch with the cellphone via Bluetooth to follow the initial setups.

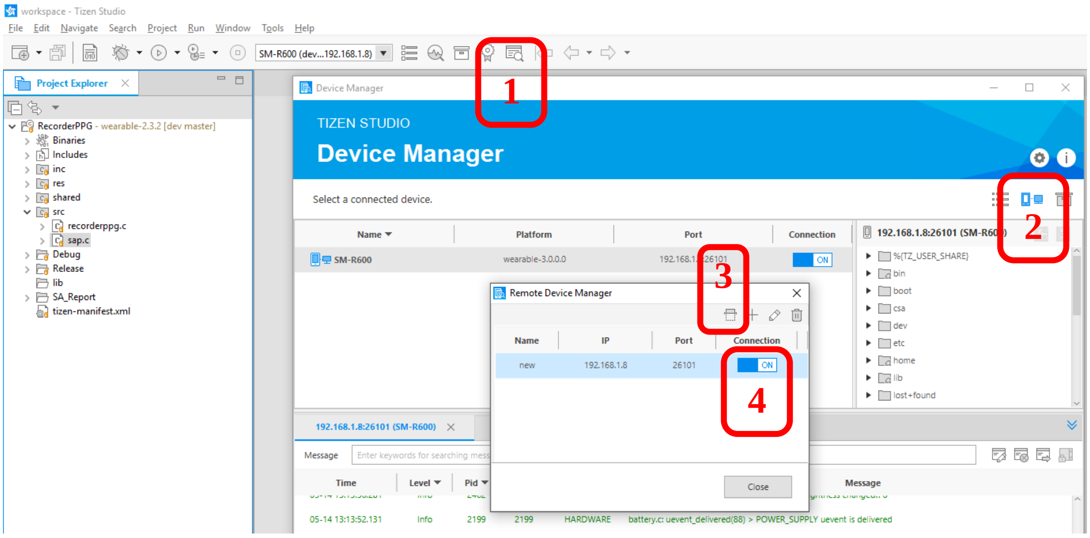
Go to Settings > About Gear > Software Information >Software Version and tap five times over the option until the message “Developer mode turned on” is shown.

Enable Debugging in the smartwatch: Settings > About Gear > Debugging

3.4. Connection of Tizen Studio with Smartwatch

Unpair the smartwatch from the cellphone, preferably turning off the Bluetooth connection in the smartphone. The smartwatch should be now in “Standalone mode” and automatically connected to WiFi, make sure that the connection in the computer and the smartwatch is the same.

In Tizen Studio. Open 1) Tools>Device Manager → 2) Remote Device Manager →3) Scan devices → 4) Connection ON



Select a connected device.

Name	Platform	Port	Connection
SM-R600	wearable-3.0.0.0	192.168.1.8:26101	ON

192.168.1.8:26101 (SM-R600)

Remote Device Manager

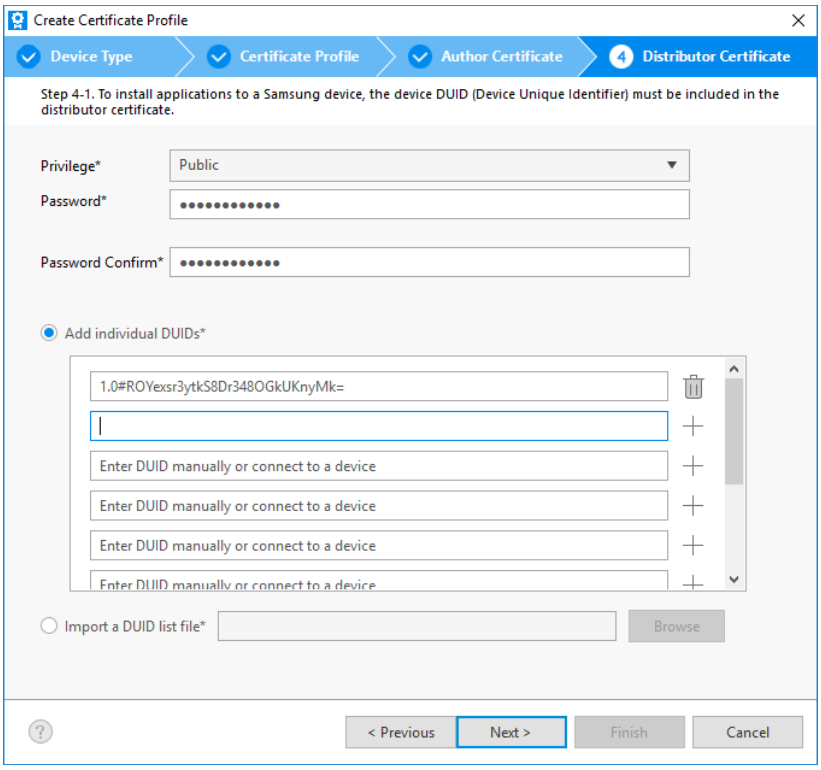
Name	IP	Port	Connection
new	192.168.1.8	26101	ON

Message: Enter keywords for searching mess

```
Time                 Level   Pid    Message
05-14 13:13:52.131   Info    2199   2199   HARDWARE   battery.c: uevent_delivered(88) > POWER_SUPPLY uevent is delivered
```

3.5. Creation of Development Certificate

To be able to deploy the application it is necessary to create a Certificate Profile, a Samsung account is required. Open Tools>Certificate Manager and create a new Samsung Certificate Profile. It is **very important** that the smartwatch is connected to Tizen Studio so that DUIDs in the last step are filled in automatically for the certificate.



Step 4-1. To install applications to a Samsung device, the device DUID (Device Unique Identifier) must be included in the distributor certificate.

Privilege*	Public
Password*	••••••••••••
Password Confirm*	••••••••••••

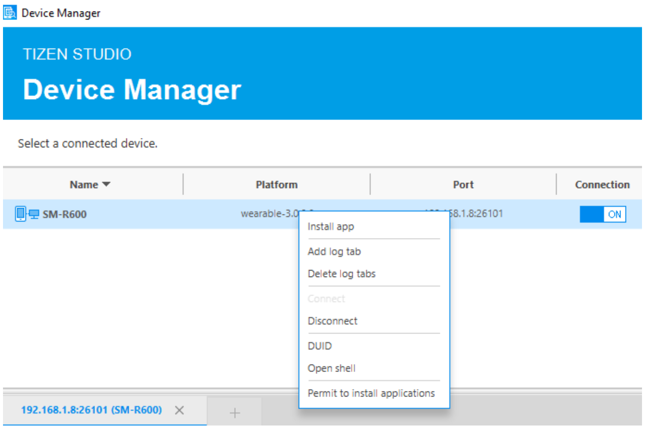
● Add individual DUIDs*

- 1.0#ROYexsr3ytkS8Dr348OGkUKnyMk=
- Enter DUID manually or connect to a device
- Enter DUID manually or connect to a device
- Enter DUID manually or connect to a device
- Enter DUID manually or connect to a device

○ Import a DUID list file*

3.6. Applying the Certificate Profile on the Smartwatch

When the certificate is created and active, it is necessary to upload it to the device. In the Device Manager, right-click the connected device and press “Permit to install applications”.



Select a connected device.

Name	Platform	Port	Connection
SM-R600	wearable-3.0	8.1.8:26101	ON

<u>Now the application can be compiled and deployed on the smartwatch!</u>

3.7. Considerations about the Source Code

3.7.1. Communication Smartwatch – Smartphone via SAP

The SAP communication was developed based on the guidelines in:

https://developer.samsung.com/galaxy-watch/develop/creating-your-first-app/native-companion/use-sap

And the example source code available in:

https://developer.samsung.com/galaxy-watch/develop/samples/companion/hello-message-native

In order to the Android application to recognize the data sent from the Tizen wearable app, it is mandatory that the APP_PROFILE_ID variable are the same in both smartphone and smartwatch applications.

For the smartwatch app two lines need to be changed. In the file “sap.c” the macro definition APP_PROFILE_ID in the header. And in "res/xml/accessoryservices.xml", the keyname “serviceProfile id”.

3.7.2. Extracting Log Files

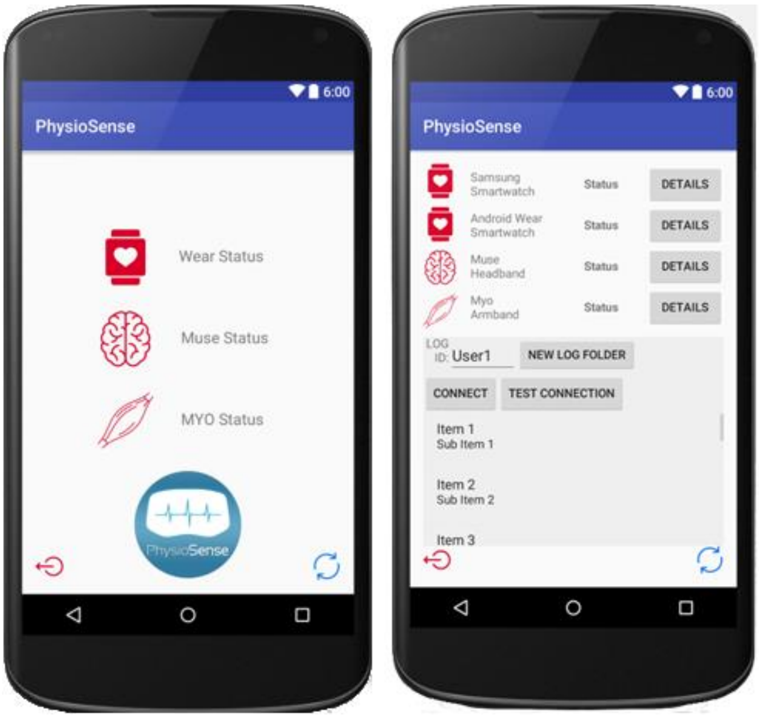
Every time a new session is started, a log file is saved with the recorded data. The files can only be extracted connecting to the Device Manager in Tizen Studio and going to the path: “opt/usr/media/”. All logs are saved in the folder “ppg_recorder_logs”. Can be extracted to the PC for offline processing by right-clicking the folder and choosing “Pull file”.

3.7.3. Native Development instead of Web Development

A native application was developed because the development of web applications did not allow to access the sensor HRM_RAW, which is needed to collect the PPG signals. See the issue in:

https://developer.samsung.com/forum/board/thread/view.do?boardName=SDK&messageId=356484&startId=zzzzz~

4. Setup of APP 2: PhysioSense.v2 | Android



Application that runs in the background and is in charge of receiving data from the smartwatch and process it in real-time. It connects through SAP protocol, receives the packets and create different log folders for the incoming data in request of the user.

4.1. Environment

Application developed in Android Studio 1.4.1 as a Java application.

4.2. Setup Smartphone

Configure developer mode on the Android phone according to the guidelines:

https://developer.android.com/studio/debug/dev-options

Install the corresponding drivers for the cellphone, in this case Samsung S9:

https://www.samsung.com/us/support/owners/product/galaxy-s9-unlocked

https://developer.samsung.com/galaxy/others/android-usb-driver-for-windows

4.3. Setup Android Studio

Open the project in the folder App2. It contains several Android and Java projects:

The package “pppgreceiverunity” is an Android library to be used in Unity (still under development), intended to replace the background application and use data processing directly from the .apk compiled in Unity.

The package “math.peakdetector” is a Java application that runs the peak detector algorithm with a test signal.

The package “app” contains the final project with the PhysioVR v2, the first version is found in: https://github.com/PhysioTools/PhysioVR

4.4. Considerations about the Source Code

4.4.1. Communication Smartwatch – Smartphone via SAP

In order to the Android application to recognize the data sent from the Tizen wearable app, it is mandatory that the APP_PROFILE_ID variable are the same in both smartphone and smartwatch applications.

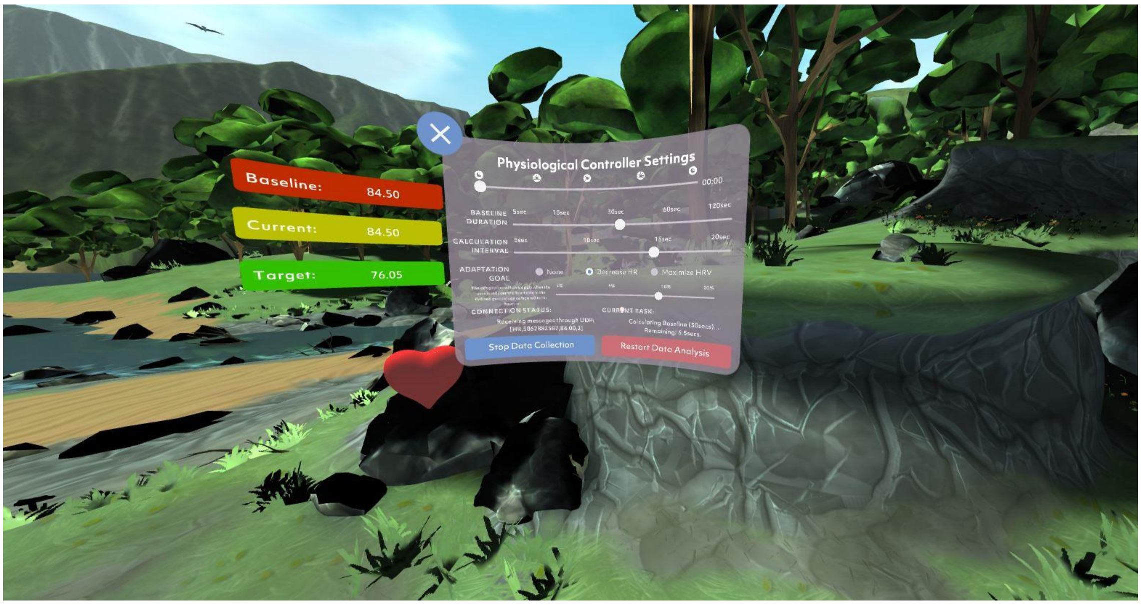
For the smartphone, in the file "res/xml/accessoryservices.xml", the keyname “serviceProfile id” has to coincide with the “serviceProfile id” in the smartwatch application.

4.4.2. Extracting Log Files

Every time a new session is started, or when the user presses “NEW LOG FILE”; a log file is saved in the path “Downloads/PhysioVR_Logs”. It contains one folder with the timestamp and the ID written in the UI. There are four files that are logged.

- signal.csv: Original data gathered from smartwatch through SAP
- processedSignal.csv: Signal after signal processing algorithm.
- peaksAndHRV.csv: Position of detected peaks and calculated HRV.
- debugLogger: Log of events and messages during peak detection.

5. Setup of APP 3: CalmPlace | Unity



Baseline: 84.50

Current: 84.50

Target: 76.05

Physiological Controller Settings

00:00

BASELINE DURATION

CALCULATION INTERVAL

ADAPTATION GOAL ○ None ● Decrease HR ○ Maximize HRV

CONNECTION STATUS: Receiving messages through UDP: [HR,5867882587,84.00,2]

CURRENT TASK: Calculating Baseline (30secs)... Remaining: 6.5secs.

Application that runs the virtual reality environment in Unity. It receives the metrics from the physiological signal and executes the adaptations in the virtual environment.

5.1. Environment

Application developed in Unity 2018.2.5f1 as a C#/Unity application.

5.2. Setup Unity

Configure Unity to compile Android application following the guides in:

https://unity3d.com/learn/tutorials/topics/mobile-touch/building-your-unity-game-android-device-testing

5.3. Configure Oculus Signature

All Gear VR applications must be signed with an Oculus Signature File (osig) during development to access low-level VR functionality on the mobile device. This signature comes in the form of an Oculus-issued file that you include in your application.

To see the instructions about how setup Unity for development with Oculus:

https://developer.oculus.com/documentation/unity/latest/concepts/unity-mobileprep/

To generate the osig file, go to: https://dashboard.oculus.com/tools/osig-generator/

When the osig file is created, put it in the Unity folder path "Assets/Plugins/Android/assets/" without renaming the osig file.

5.4. Considerations about the Source Code

5.4.1. Public source code of App3 is restricted.

The code that is available online (https://github.com/luiseduve/pare-vr/) does not include the CalmPlace environment, only the code used for data workflow and adaptation. Therefore, the visual panel for physiological control should be adapted for new Unity applications using the provided scripts.