User manual - Historiographer

Michael Brilka

June 2024

Contents

1 Introduction & Limits

The goal of the Historiographer-App is to enable long term recordings of sensor data on the Bangle.js 2. We achieve this goal by converting the data and saving the data in a custom binary format.

In the current version of the app, we make it possible to record the following sensors at a given interval:

- Heart-rate monitor, selectable intervals: 10, 20, 40, 80, 160, 200 ms
- Barometer, every 120 ms, or a multiple of it
- Accelerometer, every 80 ms, or a multiple of it
- Magnetude, every 80 ms, or a multiple of it
- Compass, every 20 ms, or a multiple of it
- GPS, every 1000 ms, or a multiple of it

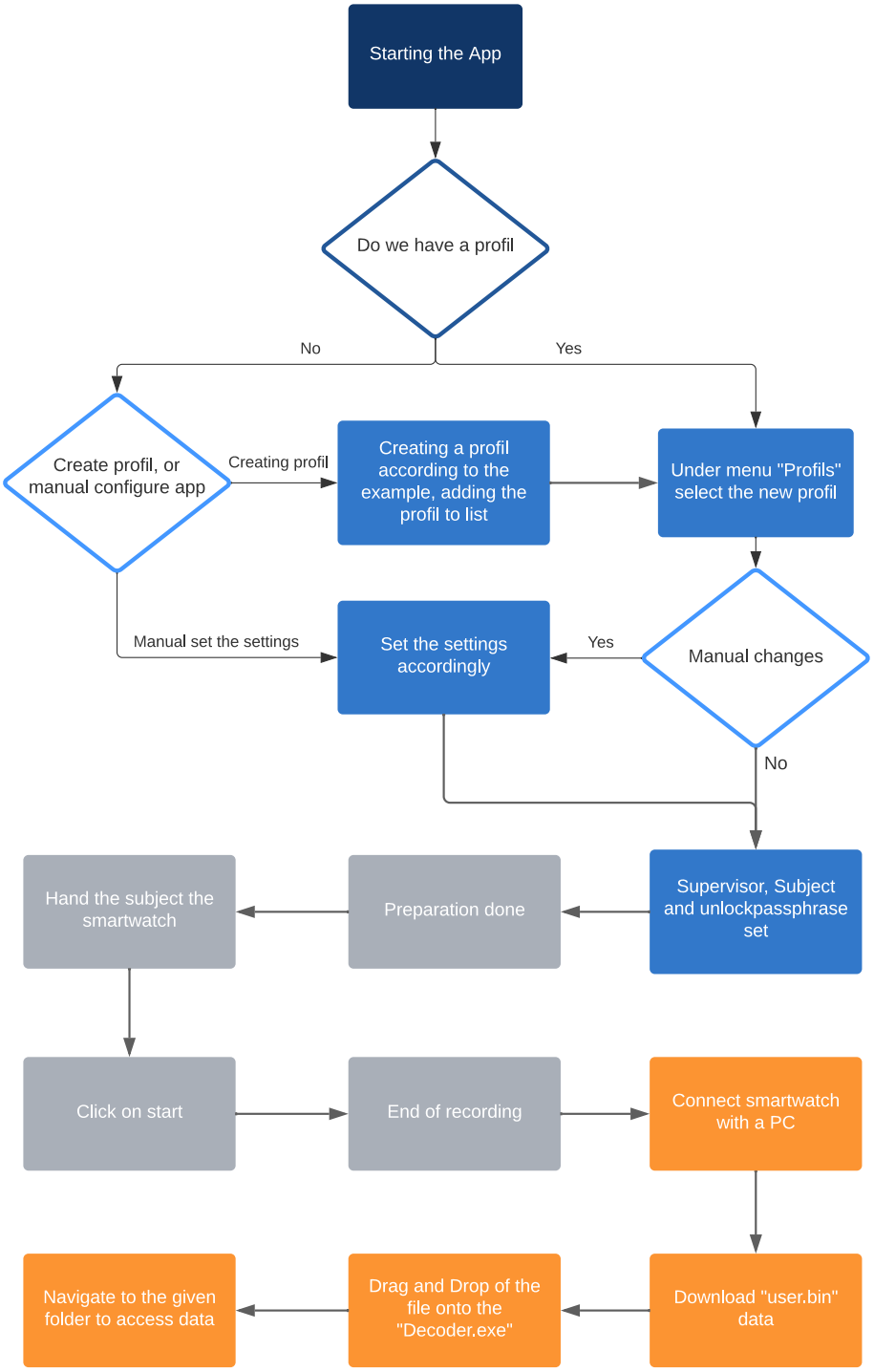
The runtime of the Historiographer-App is limited by the storage. To give an estimation for the total runtime, we developed formulas. While the complete formulas would be too complex for this user manual, we simplified it down. Our simplification includes that we assume that sensible options were chosen. We also assume that we have at least one sensor, that records its value with an interval of 255 ms or less. The last tine we leave open is the skips_ entry's of the senors. 0 indicates that all values are going to be recorded. If we instead make this number negative, it indicates that we don't record the sensor. This leaves us with the following equation[1]:

$$\begin{aligned} runtime = (&StorageFileinbits - 290bits) \\ &/(24bits/selected\_HRM \\ &+ 140bits/(120ms * (skips\_Barometer + 1)) \\ &+ 60bits/(80ms * (skips\_Accelerometer\_XYZ + 1)) \\ &+ 28bits/(80ms * (skips\_Accelerometer\_Magnitude + 1)) \\ &+ 112bits/(20ms * (skips\_Compass + 1)) \\ &+ 204bits/(1000ms * (skips\_GPS + 1))) \end{aligned} \tag{1}$$

After a successful run, we need to download and decode the data. For decoding, we use the "Decoder.exe" executable. It decodes and exports the data automatically under the name of the supervisor and subject. The decoded data then can be stored as a CSV- or XML-file.

[1] To make the equation simpler to understand, we omitted the case where skips_* is -1 or lower. Setting skips_* to -1 will result in a division by 0, which isn't allowed. If set to -2 or lower, the result will be a negative number, that will falsely increase the runtime, which isn't allowed either. Therefore, the term can be excluded during calculation.

2 Workflow


Starting the App
Do we have a profil
No
Yes
Create profil, or manual configure app
Creating profil
Creating a profil according to the example, adding the profil to list
Under menu "Profils" select the new profil
Manual set the settings
Set the settings accordingly
Yes
Manual changes
No
Hand the subject the smartwatch
Preparation done
Supervisor, Subject and unlockpassphrase set
Click on start
End of recording
Connect smartwatch with a PC
Navigate to the given folder to access data
Drag and Drop of the file onto the "Decoder.exe"
Download "user.bin" data

3 Profil

The goal of the profiles is, to allow people to immediately record data, without the need to fully understand the Historiographer-App. Therefore, a single knowledgeable worker can create profiles. These simplify the whole settings procedure down. Now only the name of the profile needs to be selected and the name of the subject must be entered. Next we can give the Bangle.js 2 to the subject and the recording can begin, by pressing the start button.

4 Manuel settings

Because not every combination of settings requires an independent profile, we also support the setting of settings. These will not override the profile. The settings compromise of:

- 1 Size of file
- 2 Size of RAM storage
- 3 Selection of sensors, and their interval
- 4 Notification of the User, regarding low storage or battery
- 5 Shutdown of sensors during low storage
- 6 Shutdown of sensors during low battery

In the following sections, we describe the settings:

4.1 Size of file

This option sets the size of the file during creation and is given in Kilobytes. The Bangle.js 2 has an 8 MB SPI-Flash storage onboard. From this, we can roughly use 7.5 MB. The available storage can be viewed in the "About" menu in the settings.

4.2 Size of RAM storage

We use the RAM storage to save values firstly there. This is due to performance reasons: we save the time to prepare to write to the main storage. The bigger the RAM storage is, the more efficient we will write the data, unfortunately we also increase the overall time for writing the data. In our tests, we came to the conclusion that a size of 768 or 1024 Bytes offers the best balance between efficient writing and amount of interruptions.

4.3 Selection of sensors, and their interval

All of the available sensors can be activated and set their recording interval independent of each other. The recording interval is always a multiple of the actual sensor speed. Optional to improve the accuracy of the interval we can set the internal update interval from 80 ms to 10 ms.

4.4 Notification of the User, regarding low storage or battery

Because the storage or battery has a limit, we implemented a two stage warning system. Here we implemented the first stage. It simply warns the user after a number of Kilobytes are used or only x% of the battery are left. The warning consists of a vibration and a notification on the display. After accepting this, we display an additional symbol on the widget for low storage and/ or battery.

4.5 Shutdown of sensors during low storage

Through the disabling the recording of the sensors, at low storage capacity, we enable that only the most important sensors will be recorded. Other widgets still can access this sensor, which enables them to still record data.

4.6 Shutdown of sensors during low battery

The shutdown of sensors, at low battery, is designed like the shutdown during low storage space. The difference is that we completely shut the sensor down. Therefore, other widgets cannot access the sensor.

5 Creation of profiles

To create custom profiles, the provided "Custom.json" can be edited. It is important though that all variables are assigned a value, otherwise the JSON parser can create an unrecoverable exception.

After the creation of a profile, we can upload the file to the Bangle.js 2. Now we need to add the new profile to the "availableConfigs.json" file. For that, we con download the existing version, add the new profile, and upload the file to the Bangle.js 2.