Software paper for submission to the Journal of Open Research Software

To complete this template, please replace the blue text with your own. The paper has three main sections: (1) Overview; (2) Availability; (3) Reuse potential.

Please submit the completed paper to: editor.jors@ubiquitypress.com

---

(1) Overview

Title

WEARDA: human activity monitoring with wearable sensors

Paper Authors

1. Dijk, van, Richard;
2. Gawehns, Daniela;
3. Bourdage, Renelle;

Paper Author Roles and Affiliations

1. Research software engineer, Leiden Institute of Advanced Computer Science, Faculty of Science, Leiden University, The Netherlands.
2. PhD candidate, Leiden Institute of Advanced Computer Science, Faculty of Science, Leiden University, The Netherlands.
3. MSc student Psychology, Faculty of Social and Behavioural Sciences, Leiden University, The Netherlands.

Abstract

We present the Wearable Sensor Data Acquisition WEARDA software package for acquisition of data from Samsung wearable devices, github.com/LiacsProjects/Wearda. WEARDA contains data collection experiences in the field of monitoring activities of persons. The software package was initially created for the Dementia project "Dementia back at the heart of the community" [0] [1]. A Samsung smart watch [1] was chosen because of its relatively open operating system, well accompanied tooling for software development, and acceptance by participants. Together with WEARDA, the watch can monitor real-life subject's activities by recording acceleration (tri-axis accelerometer), rotation velocity (tri-axis gyroscope), outdoor location trajectory (Global Positioning System), passing doors, and switching floors (barometer). Privacy rules can be applied by use of a predefined privacy circle based on GPS coordinates. Recordings are labeled as outside the privacy circle whenever the watch left this predefined area.

Keywords

Human physical activity; Smart watch; Tizen OS; Tri-axes accelerometer; Tri-axis gyroscope; GPS; Privacy geofencing.

[1] This project was a collaboration between Netherlands Institute for Health Services Research (NIVEL) Utrecht and Leiden Institute of Advanced Computer Science (LIACS), Leiden University, funded by ZONMW, with project timeline from 2018 to 2022

1 Introduction

We present the WEARDA software package running on a smart watch to monitor human activity, developed as part of the project "Dementia at the heart of the community".
This project was initiated in 2018 as a collaboration between NIVEL Utrecht and LIACS, Leiden University. One of the objectives was to study activity patterns of people living with dementia.

The chosen smart watch The Samsung Gear Fit Pro 2 smart watch, issued in 2016, was selected to monitor this human physical activity. The watch contains acceleration-, gyroscope-, barometer-, and Global Positioning System (GPS) sensors. The watch collected human physical activity when the subjects wore the watch around their wrist. The operating system of the watch is Tizen OS: an operating system with strong software development tooling, free of charge and with a rich, well documented API accompanied by many programming examples, all available online [2].

The WEARDA package The WEARDA software package is developed for the researcher. It can be used to collect human activity in a real-life environment where the subjects work or live, and used to collect human activity data in an experimental setting, where the activity behavior is known and there is a wish for annotation.

Requirements and solutions Data collection on real-life subjects is a difficult task: it needs thorough preparation, measurement execution, and error-less logistics. The following paragraphs describe these attention points in detail.
The first attention point of data collection is an efficient and error-less preparation of taking measurements. The watches can easily be configured with configuration settings created on a laptop. Watches and laptop are connected wireless, and act standalone, no other facilities needed. The watches can be configured before data collection will take place, and can store their measurement data for days of data collection without the need for a laptop to be on site.
The second attention point of data collection is an efficient and error-less measurement execution. The WEARDA software does its measurement in the background, invisible for the subject, and it cannot be stopped to prevent unintended interventions. The stored measurements on the watch can be copied to a laptop, and later removed, with Tizen software running on this laptop. Removal of the measurements can also be done with the WEARDA application installed on the watch, if there is no laptop available on site.
The third attention point of data collection is an efficient and error-less logistics. The measurements are linked to an unique identifier of the watch, and are accompanied by the (corrected) configuration settings used by that particular watch. The WEARDA software corrects invalid configuration settings: it will not block the start of the measurements, nor produce error messages, if errors are detected in the original configuration settings. With this correction mechanism, the researcher will not be confronted with the absence of measurements later, after intensive data collection efforts on site. Further, entering the person-id (an unique number attached

to a real- life subject) is designed to be done fast and error-less. It is done with a digital spinner wheel, showing all possible id's while swiping: so no individual digits must be entered with a digital, very small, keyboard, which is time consuming because it needs a solid, relaxed hand, which is not always doable on site, and therefore could induce errors quickly.

Data collection steps If we take the aforementioned attention points and related solutions into consideration, the data collection on real-life subjects steps are: create a particular configuration setting and upload this to all watches used in the study (preparation). On site, each watch is activated by the researcher with the WEARDA sensor app: remove all previous collected data, enter a person-id, start the measurement, and then let the subject wear the watch on their wrist (measurement execution). After data collection, every watch is read out by use of a laptop to copy its data, to be placed in one folder, first sorted by person-id, and after that by date time (error-less logistics).

Privacy preservation The WEARDA package contains privacy preservation functionality by configuring a GPS circle around an certain area. If the smart watch is residing outside of this circle, data is marked as being private, but still recorded because of possible inaccuracy of GPS. The data collected can best be removed after verification.

Structure of document The following section will be about the implementation and architecture of the WEARDA software. It is followed by a section regarding the quality control of the package, where the tests and the techniques are provided to ensure the high quality of the delivered software. In the availability section, details on the WEARDA software system requirements and dependencies are given, as well as the information on how to obtain the software. The last section is about the software reuse potential, and provides an overview of how to configure and run WEARDA on a Samsung watch.

2 Implementation and architecture

We developed the WEARDA software package, consisting of two main components: an interactive application that can be operated through the touchscreen of the watch, and a data collection service that runs in the background on the watch.

2.1 Sensor application

The software package contains the Tizen OS widget application "liacs.sensorapplication" visible in the list of applications with as title "Sensors" and a digital brain as icon.

Application front-end The application carries a spinner entry widget control to enter the person-id, a number between 000 and 999, and two buttons: the RESTART button to start or restart measurements, and the CLEAN button to remove all measurements from the watch. The CLEAN button can be activated by pressing it three times to prevent unintended removal when accidentally pressing the button. The application can be closed by pressing the back button on the side of the watch. This will not influence the measurements, and allows the participant to use other applications on the watch. We did not supply a STOP button because of the fear that participants would press this button unintended. Thus, the measurements can only be stopped by a full shutting down of the watch. After several measurements the battery will be depleted, and the watch shuts down automatically. While re-charging, the data can be downloaded, removed, and the watch is ready for the next participant. Shutting the watch down reduces power consumption, and thus, the charge duration will be shortened for the next experiment to start. In data collection setting where the data needs annotation, the person-id can be used as an annotation label. For example, if different types of daily activities needs labelling, such as washing dishes, cutting vegetables or biking, the person-id could be used (e.g., ID 001 = washing dishes, 002 = cutting vegetables, et cetera).

Figure 1: Left: the icon of the sensor application in the Tizen OS menu, right the sensor application with the person-id snipper, restart, and clean button.

Measurement files The measurement records are stored in comma separated values sensor files, and the file name contains the person-id, date, time, watch-id, and sensor types. This information also appears in the first line of these files. The researcher retrieves the files by use of the Smart Development Bridge (SDB): a command line tool which is part of the Tizen Studio package. It is advised to retrieve the files daily while collecting data, for example to validate the data in the evening, to prevent loss of data because watches can get lost or damaged. In principle, the watch can store 0.5 GB of data enough for 7 days of measurements 12 hours a day. So, it could be possible to monitor real-life subjects during the day for one week continuously without intervention of the researcher to download the data frequently during this period. However, the battery will be depleted after 12 hours, so the real-life subject should recharge the watch battery daily in this data collection scenario.

Files separation Sensor values can be recorded with different frequencies. The motion sensors are usually read out with high frequency 25-50 Hz, while the barometer, battery, and GPS (Global Positioning System) have lower frequencies 0.25 – 2

Hz. For every frequency, a separate file is created. In that way, there is no need for sensors with low frequency rates to be recorded many times as duplicates, if these recordings are mixed with sensors who are recorded with higher frequency rates. This separation of files, thus reduces power consumption, and memory usage, because it reduces also the number, and size of recordings. Sensor values are stored in the comma separated value file format.

Central clock The application uses one central clock. The time recordings in several sensor files can be aligned with this central time reference. It is advised to synchronize the time of the watches before the data collection. Precise time synchronization can be established automatically by connection to internet via Bluetooth or WIFI.

Configuration file The measurements are configured with a configuration file which is uploaded to the watch, and placed in an application independent, accessible file system directory. Its contents is read before starting the measurements. The configuration settings used, and enriched with the software package version, is saved in a meta file which has an identical structure as the original configuration file. This file is so an integral part of the sensor files generated by each series of measurements. Errors in the configuration file are corrected to default settings. Once a configuration file is designed, it can be uploaded to the watches with the SDB command line tool.

2.2 Sensor service

The software package contains an Tizen OS widget service "liacs.sensorservice" which runs in the background, neither visible in the list of applications nor visible via the display of the watch. This design makes continuous measurements possible since Tizen applications become idle after brief time to reduce power consumption of the display, while services become idle if the battery is almost empty.

Processing messages The sensor service processes the messages RESTART and CLEAN sent by the sensor application. After start of the watch, the sensor service is activated and waiting for these messages. After one valid message is received, the service starts the recording of the sensor values and stores it into comma separated values files. The measurement is based on the contents of the uploaded configuration file. Each RESTART message will reload the configuration file found in its uploading folder. This allows the researcher to change the configuration of watches while the measurement is still running.
Figure 2 shows a state diagram with transitions with format "event - actions", as a formal description of the states, and relevant state events between the watch application manager, the sensor application, and service.

3 Quality control

The watches were manually tested. We conducted a spinning test — watch rotating on its Gorilla glass – to validate the values of the acceleration and gyroscope sensors.

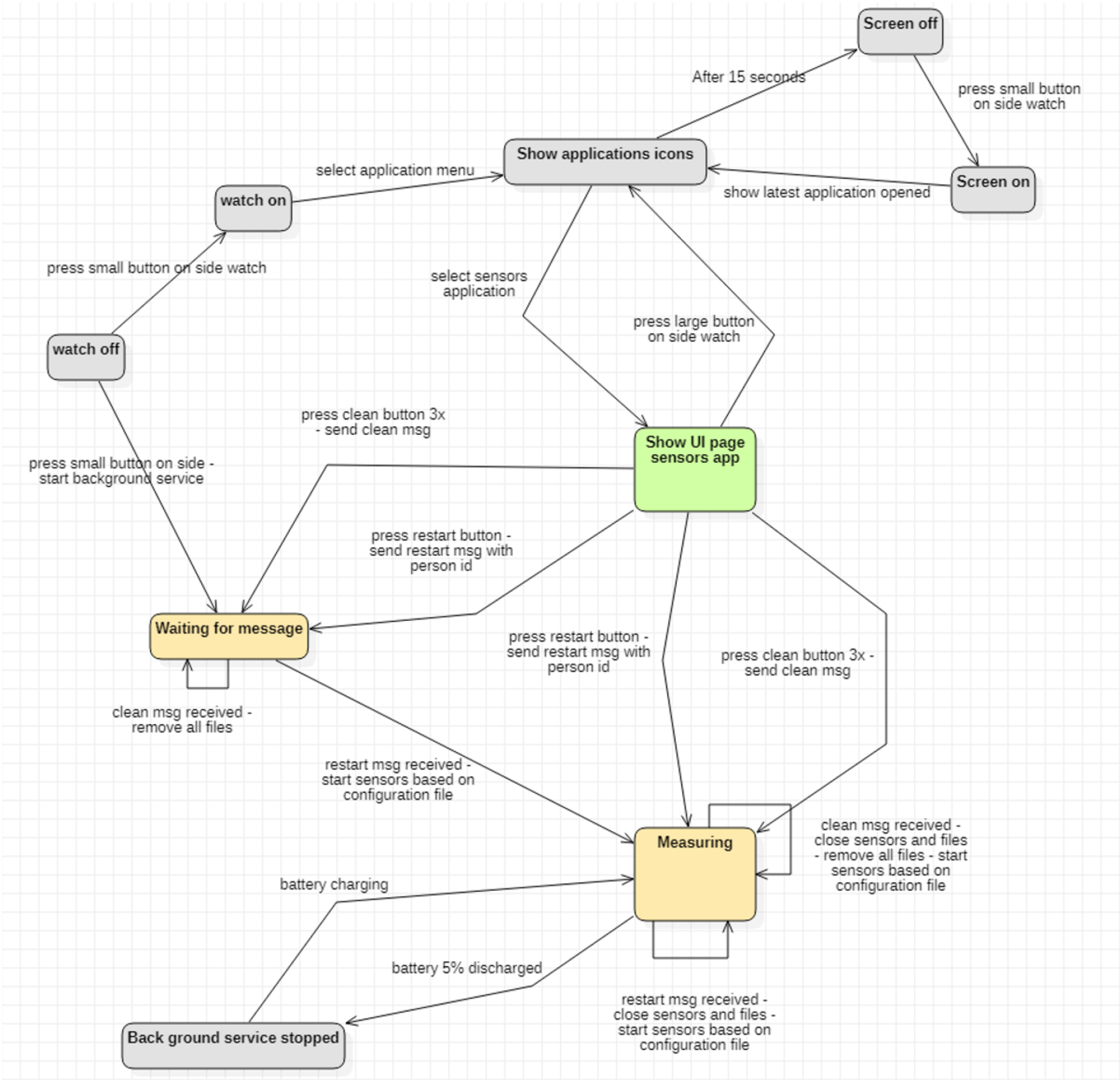

Figure 2: State diagram of sensor application (green), sensor service (orange), and the Tizen OS (grey).

The sensors were positioned a bit outside of the center of the watch.

Gyroscope and accelerometer The gyroscope values showed a decreasing trend on two of its axes, the gravity acceleration axis showed $ax = 0$, $ay = 1.2$ m/s^2, $az = 9.81$ m/s^2. The linear acceleration was not valid for high rotation velocities, but showed promising values after 32 seconds. The linear acceleration is derived from the gravity acceleration with an algorithm used by Samsung.

Bias line measurements Zero measurements or bias line measurements showed that the standard deviation of the gravitational acceleration $G = \sqrt{(ax^2 + ay^2 + az^2)}$ ranges between 0.02 - 0.036 m/s^2. Its average G ranges between 9.6 - 10.2 m/s^2. Further, the average gravitational acceleration G was dependent on the spatial orientation or spatial positioning of the watch compared to the direction of the

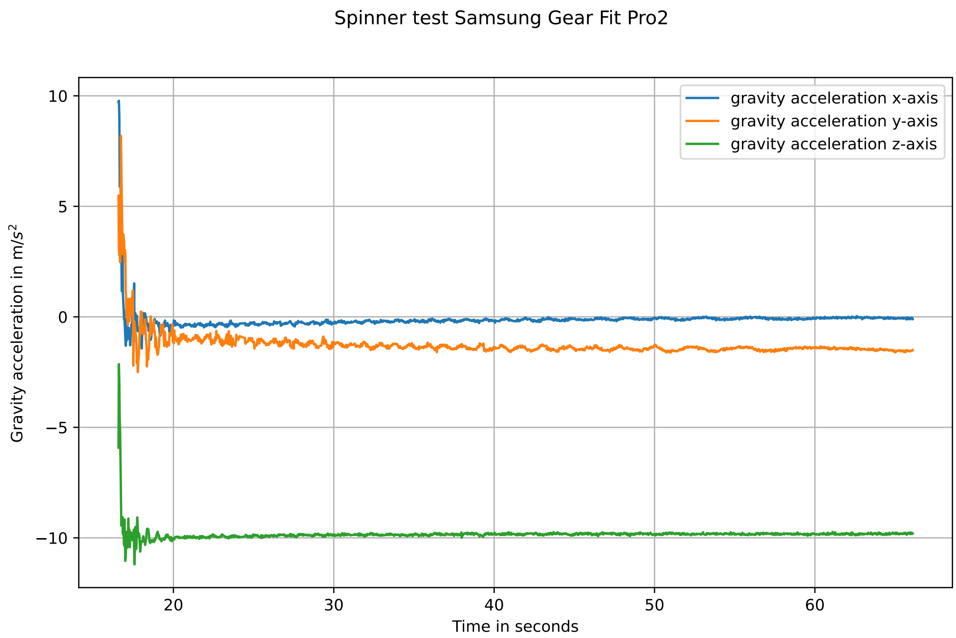

Figure 3: Watch spinning on its glass, decelerating. The oscillation is caused by the movements of the arms of the band. The graph shows the gravity acceleration per axis in m/s^2.

gravitational force or orientation of the surface of the ground as well.

Barometer and GPS The barometer and GPS were validated by comparing height differences of a GPS track with the air pressure measured [3]. An air pressure difference of 1 millibar (hPa) corresponded to 7.75 meters at sea level, temperature 10 degrees Celsius. The accuracy of the barometer is roughly 0.01 millibar (0.075 meters).

Irregular sampling rate We noticed that the Tizen OS API is producing different biased time stamps per sensor type. On top of that the sampling time appears to be irregular. To overcome this problem, we implemented a write timer which writes all sensor values known at a certain moment, to one or more sensor files. If the write timer interval is shorter then a specific sensor readout interval, all duplicate values are removed. Vice versa there will be no duplicates produced.

Writer and sensor intervals, and sample density The best intervals of the writer timer and the sensor read outs are listed in table 1. The table also contains the realized sample density which is defined as the ratio between recorded samples divided by the expected number of samples. A sample density of 1 is ideal, if it is lower then 1 the effective sample frequency is lower then the configured one, if it is larger then 1 the effective sample frequency is higher then the configured one.

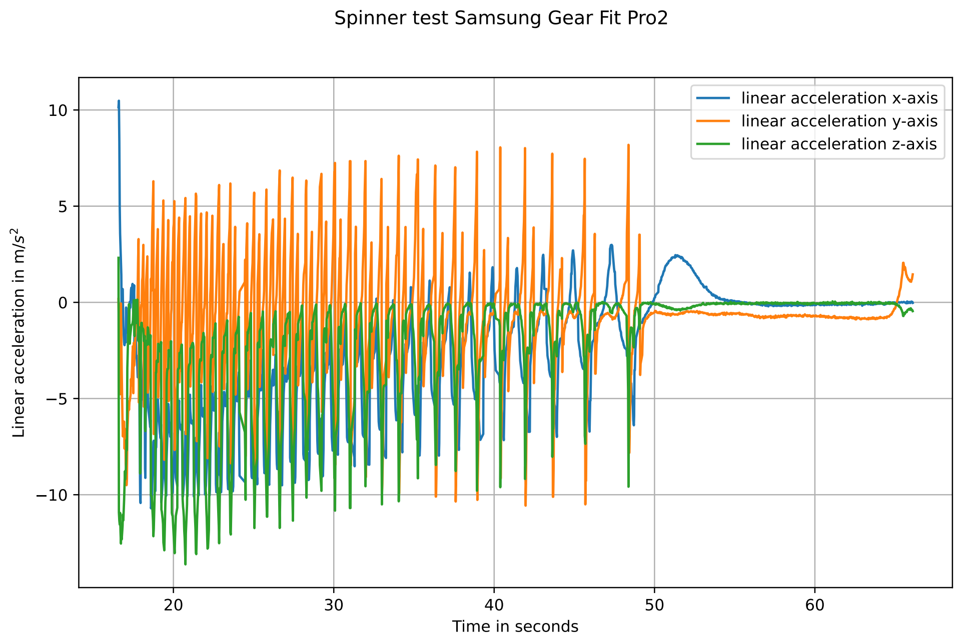

Figure 4: Watch spinning on its glass, decelerating. The oscillation is caused by the movements of the arms of the band. The graph shows the linear acceleration per axis in m$/s^2$.

None blocking error handling At last, we mention the none blocking error handling approach. Errors are corrected if possible, or at least will be bypassed to prevent sudden unexpected stops. For example, if the configuration file contains values being out of range, they will be replaced by typical or default values. For the release version of the WEARDA package, errors are not shown to the user nor is it logged. Logging in general (not only for errors) is switched off to reduce power consumption and memory use. Only, the debug version of the WEARDA package will log information about the program flow, or warnings, or errors.

Table 1: Sample density as function of write- and sensor readout interval in milliseconds.

Sample rate	Sensor interval	Writer interval	Sample density GPS off	Sample density GPS on
50Hz	10ms	20ms	0.99	0.90-0.93
50Hz	20ms	10ms	0.94	
40Hz	25ms	25ms	0.80-0.87	0.975
40Hz	10ms	25ms	0.86	
40Hz	15ms	25ms	0.94	
30Hz	15ms	33ms	0.96	
30Hz	33ms	33ms	0.97	
25Hz	20ms	40ms	0.99	
20Hz	10ms	50ms	0.96	
20Hz	25ms	50ms	0.98	
20Hz	50ms	50ms	0.93	
10Hz	100ms	100ms	0.99	

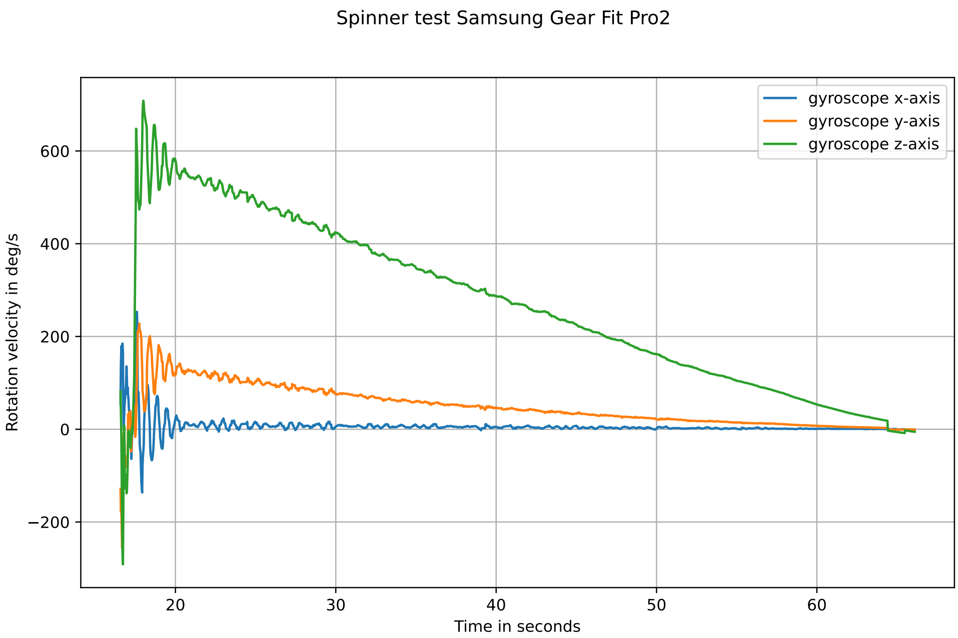

Figure 5: Watch spinning on its glass, decelerating. The oscillation is caused by the movements of the arms of the band. The graph shows the rotation velocity per axis in degrees per second.

(2) Availability

Operating system
Tizen OS developed by Samsung based on Linux OS, version 2.3.1:13

Programming language
C-programming language, Tizen API version 2.3.1:13

Additional system requirements
The software needs some disk space to store the data collected. Typical for 7 days of data collection, 12 hours a day, 50Hz sample frequency 500 MB will be sufficient.

Dependencies
None

List of contributors
Richard van Dijk, research software engineer of the LIACS software lab was the only developer of the software package. The software was reviewed by his colleague Jeremie Gobeil, also research software engineer of the LIACS Software Lab.

Software location:
Archive (e.g. institutional repository, general repository) (required)

Name: Zenodo
Persistent identifier: `https://doi.org/?/zenodo.?`

Licence: MIT license
Publisher: LIACS Software Lab, Leiden University
Version published: Sensor application v1.0.1 and sensor service v1.0.2
Date published: ?/?/21

Code repository (e.g. SourceForge, GitHub etc) (required)

Name: GitHub
Persistent identifier: https://github.com/LiacsProjects/Wearda
Licence: MIT license
Date published: 28/07/21

Emulation environment (if appropriate)

Name: None
Persistent identifier: None
Licence: None
Date published: None

Language

The language of repository, documentation, software and supporting files, are all in English.

(3) Reuse potential

While developed for the Dementia project of LIACS and Nivel, the WEARDA package was designed in a generic manner to allow reuse in other contexts. The package can be reused with minor reconfiguration for any project where another wearable device running the Tizen OS is used for collecting activity information. Items to be configured include the coordinates and radius of the privacy circle, selection of the sensors, and the polling frequency appropriate for the type of sensor being used, see table 2 for an overview of all settings.

Table 2: Configuration settings.

Field name	**Field type**	**Possible values, () = default**
Unique identifier watch	string	Example "D8F8"
Accelerometer interval	integer	0 indicates switched off, 10-1000 (25) ms
Linear accelerometer interval	integer	0 indicates switched off, 10-1000 (25) ms
Gyroscope interval	integer	0 indicates switched off, 10-1000 (25) ms
Barometer interval	integer	0 indicates switched off, 10-1000 (100) ms
GPS interval	integer	1-10 (1) seconds
GPS middle point privacy circle, latitude	float	-90.0-90.0 (52.169311) degrees
GPS middle point privacy circle, longitude	float	-180.0-180.0 (4.456711) degrees
GPS radius privacy circle	integer	0 indicates switched off, 10-1000 (100) meter
Write recordings interval	float	0.01-10.0 (0.05) seconds

For projects where similar watches are used (i.e., smart watches with a comparable set of sensors), the overall architecture of the WEARDA package is reusable, as are the optimization strategies for energy consumption, time synchronization, robustness of the measurement and storage of the data.
The wearable chosen meets the requirements set out in 2018. The development of smart watches continued, and there will be better alternatives available in terms of

battery discharge / charge duration, quality of the sensors, amount of sensor types, and software development environments such as Android Wear OS of Google. Google and the Tizen foundation recently decided to merge both their operating systems Android Wear OS and Tizen OS into a new OS called "Wear" [4]. This means that our application can also run-on Android Wear OS wearables in the future possibly with little adaption.

Acknowledgements
We acknowledge contributions from Jeremie Gobeil, Joost Visser members of the LIACS Software Lab, and Matthijs van Leeuwen group leader of the LIACS Explanatory Data Analysis group during the genesis of this project.

Funding statement
This work is partly financed by ZonMw, under project number 733050846, the hours of the LIACS Software Lab were financed by LIACS, the Leiden Institute of Advanced Computer Science.

Competing interests
The authors declare that they have no competing interests.

Data policy
The data collection of the Dementia project of LIACS and Nivel will be available to other researchers but not to the general public because of their privacy sensitive nature.

References
[0] Dementia back at the heart of the community, `https://www.universiteitleiden.nl/en/research/research-projects/data-science-research-programme/data-science-resea`

[1] Samsung Gear Fit Pro 2 manual / official specs, `https://www.samsung.com/us/business/support/owners/product/gear-fit2-pro-bluetooth/`

[2] Tizen Foundation organization, `https://www.tizen.org/`

[3] GPSVisualizer, GPS website by Adam Schneider, Portland, Oregon, United States, `https://www.gpsvisualizer.com/`

[4] Merge of Wear OS and Tizen, `https://www.theverge.com/2021/5/19/22443809/google-samsung-wear-os-tizen-things-we-know`