Hiking Band: User Manual

Holappa, Heidi Lundén, Jaakko-Juhani Rislakki, Tuomas

ELEC-E8408 Embedded Systems Development, Aalto University. Group I.

Table of contents

1 Introduction

The purpose of this document is to provide Hiking Band users the required information to successfully setup and use the Hiking Band system. The system consists of two applications: the Raspberry Pi Web Application and LilyGO T-watch smartwatch hiking application. This document contains a section for each application and a section for communication between applications.

A test plan has been included for both applications for the purpose of detailing to QA specialists how it can be verified that the application works as intended. Additional information for testing can be found from the SRS documentation. It is important to highlight, however, that the SRS may contain optional features that have not been implemented in the proof-of-concept version. All non-optional features listed in SRS SHOULD be available and optional features MAY be available.

2 LilyGo T-Watch Hiking application

The LilyGo T-Watch Hiking application is a proof-of-concept (later in this section PoC) smartwatch application for tracking hiking trips. The application uses LilyGo T-Watches BMA423 accelerometer to track step count and LilyGo T-Watches M8/M6 GPS Module. Average speed is computed by recording the start time of the hike and calulating average speed from tracked distance and hike duration. Users can also view information from past hikes and configure Bluetooth synchronization from the settings menu.

2.1 Requirements

Before getting started, make sure that you have the following hardware components:

- LilyGO T-Watch V2
- A Raspberry Pi 3 with a Linux-based OS
- A USB-A to micro-USB cable

> Tip
>
> While the LilyGo hiking application officially supports V2 of the LilyGo T-Watch smartwatch, the application MAY also work on V3 with configuration changes. The configuration changes are detailed in the installation instructions. Note that V3 is not officially supported.

2.2 Installation and setup

The following installation instructions were used during the development stage of the LilyGo Hiking application. Please pay careful attention to version numbers to ensure that installation proceeds successfully.

2.2.1 Arduino-cli and esp32 libraries

1. Install arduino-cli (v.1.1):

https://arduino.github.io/arduino-cli/1.1/

2. Install esp32 libraries (v.2.0.14)

```
arduino-cli core update-index --config-file arduino-cli.yaml

arduino-cli core install esp32:esp32@2.0.14

python3 -m pip install pyserial
```

3. Test your board

```
arduino-cli board list

Port         Protocol Type              Board Name FQBN Core
/dev/ttyUSB0 serial   Serial Port (USB) Unknown
```

2.2.2 Compilation and upload to esp32

Use the following table to make your compilation:

Device	Board/FQBN
ESP32_WROOM_32	esp32:esp32:esp32-poe-iso
LILYGO_WATCH_2020_V2	esp32:esp32:twatch
LILYGO_WATCH_2020_V3	esp32:esp32:twatch

For example for TWATCH V3:

```
DEVICE="LILYGO_WATCH_2020_V3"
FQBN=esp32:esp32:twatch
arduino-cli compile --fqbn $FQBN \
                    --build-path $(pwd)/build \
                    --build-property "build.extra_flags=-D $DEVICE -D ESP32" .
arduino-cli upload -p /dev/ttyUSB0 \
                   --fqbn esp32:esp32:esp32-poe-iso \
                   --input-dir $(pwd)/build .
```

> **Note**
>
> The device path may not be `/dev/ttyUSB0`. To verify the name of the USB-device, connect the smartwatch with the cable and use command `ls /dev/tty*`.

or

configure config.ini

```
./install.sh
```

> **Tip**
>
> The config.ini contains LilyGo T-Watch versions V2 and V3. To change the T-Watch version, change which version is uncommented. V3 is not officially supported, but both V2 and V3 T-Watches were used during development stage.
>
> - When V2 is selected, the GPS module in V2 is used.
> - With V3 distance is calculated based on an hard coded step length as detailed in the SRS.

2.2.3 Debugging

Add read and write access to usb device:

```
chmod 777 /dev/ttyUSB0
```

Read the serial:

```
picocom -b 115200 /dev/ttyUSB0
or
putty
or
screen /dev/ttyUSB0 115200
```

2.3 Tutorial

> **Important**
>
> Write this section!

2.4 Test plan

At this PoC stage the, test plan relies on manual testing. The functional requirements of the smart watch detailed in the SRS documentation can all be tested manually. A comprehensive list of testable features have been collected to the following subsections. These collections should assist the QA specialists in implementing suitable tests to verify that the functionalities work as intended.

2.4.1 Navigation

- User can navigate from main view to:
  - session view
  - past sessions view
  - settings view
- user can navigate back to main view from all other views

2.4.2 Session view

- When user presses start - button
  - application begins tracking user's movement
  - the session data is displayed on the view, including
    - step count
    - distance
    - average speed
  - the start button turns red and the button label changes to "stop"
- When user presses stop - button
  - applicatin stops tracking user's movement
  - the stop button turns blue and the button label changes to "start"

2.4.3 Past sessions view

- The past session view contains information on stored sessions.
- For each session the following information is shown:
  - date of the session
  - travelled distance
  - average speed

2.4.4 Settings view

> **!** Important
>
> TODO: Write this section!

3 Raspberry Pi Web Application

Some introductory words here.

3.1 Requirements

The web application and the installation and run scripts have been built on a Linux based Operating System. It is recommended to use the application on a Linux based Operating System.

The minimum Python version is 3.10. Versions for dependencies are listed in requirements.txt. Use of virtual environment is adviced, as detailed below in installation instructions.

3.2 Installation and setup

These instructions assume that the user is using a Linux based Operating System with a bash terminal emulator. The installation may either be done manually or by using a convenience script provided in the project repository.

3.2.1 Option 1: Manual installation

First setup the virtual environment

```
python3 -m venv venv
```

Then install dependencies

```
pip install -r requirements.txt
```

If you add new dependencies, create an updated `requirements.txt` with the following command:

```
pip freeze > requirements.txt
```

3.2.2 Option 2: Convenience script

Run the installation script with

```
./install.sh
```

3.3 Running the application

Running the application may also be done manually or by using a convenience script.

3.3.1 Option 1: Manually

To run the app use

```
flask --app src/app.py run
```

To debug:

```
flask --app src/app.py --debug run
```

3.3.2 Option 2: Convenience script

To run the app use

```
./start-app.sh
```

To debug:

```
./start-app.sh debug
```

3.4 Tutorial

> **!** Important
>
> TODO: Write this section!

3.5 Test plan

At this proof-of-concept stage the, test plan relies on manual testing. A comprehensive list of testable features have been collected to the following subsections. These collections should assist the QA specialists in implementing suistable tests to verify that the functionalities work as intended.

3.5.1 Main view

- The main view contains navigation buttons to hikes view and configuration view
- The main view additionally contains some key information from past sessions:
  - The last recorded session
  - The session with longest travelled distance
  - The session with fastest average speed
  - Averages for step count, distance, average speed and burned calories all sessions

3.5.2 Hikes view

- The hikes view contains a table of past hikes
  - data is correctly shown in the table
  - data with missing values is displayed correctly
  - empty table is displayed correctly
- Each row contains the following information
  - id
  - date
  - step count
  - distance
  - average speed
  - burned calories
- additionally each row contains a Delete button from which the selected entry can be deleted
  - pressing the delete button activates a modal in which user is asked to confirm deletion
  - after confirmation, entry is deleted and the hikes-view is re-rendered

3.5.3 Configuration view

> **!** Important
>
> TODO: Write this section!

4 Communication between devices

> **!** Important
>
> TODO: Write this section!