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 3B+ 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 Tutorial

This section introduces the basic functionalities of the LilyGo T-Watch.

2.2.1 Starting a hiking session

To start a new hiking session:

1. In main view press ‘Session view’ button

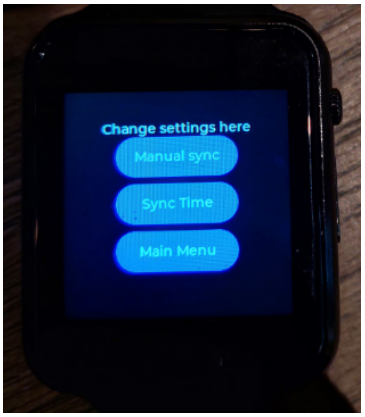

Figure 1: LilyGO main view

2. In Session view press ‘Start’ button

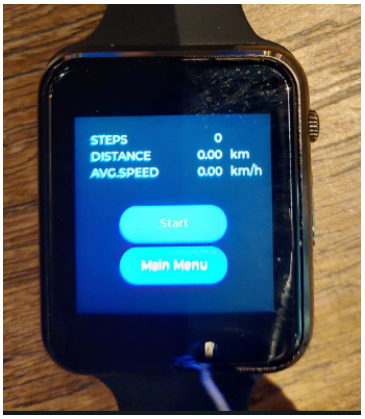

Figure 2: LilyGO session view before start button is pressed

To end a hiking session:

1. Navigate to ‘Session view’
2. Press ‘Stop’ button

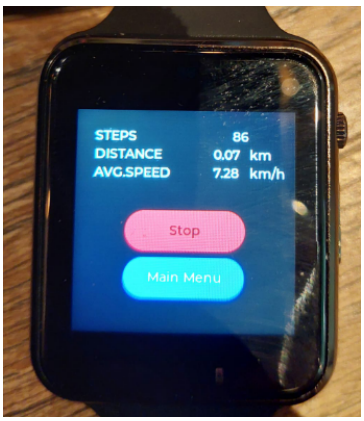

Figure 3: LilyGO session view after start button has been pressed

> **Tip**
>
> While a hiking session is active, you can:
>
> 1. Navigate to other views in the smartwatch application
> 2. Toggle the touch screen on/off with the PEK-button

2.2.2 Viewing past hiking sessions

To view past hiking sessions:

1. Navigate to past session view from the main view

Figure 4: LilyGO past sessions view

> **Note**
>
> If there are no previous session, a prompt will indicate that no hikes have been recorded. Otherwise past sessions will be listed.

2.2.3 Managing settings

> **Important**
>
> TODO

2.3 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.

2.3.1 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

2.3.2 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"

2.3.3 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
- the watch stores at maximum five past hike sessions
- if five hiking sessions have been recorded, the oldest entry will be overwritten when the next session begins

2.3.4 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 Tutorial

This section details the functionalities the web application provides once it is running locally. Navigate to the application with you browser. By default Flask applications run in port 5000 on localhost. By typing `localhost:5000`(or `127.0.0.1:5000`) to your browser, you should land on the main view.

3.2.1 Main view

The main view has two navigational buttons:

1. Button `Show all hikes` navigates to a view that shows all hikes
2. Button `Configuration` navigates to a view in which the bluetooth connection can be configures

The main view additionally visualizes selected data from past hikes:

1. Last hike taken
2. Average values for all hikes in persistent memory
3. Hike with the longest distance
4. Hike with the fastest average speed

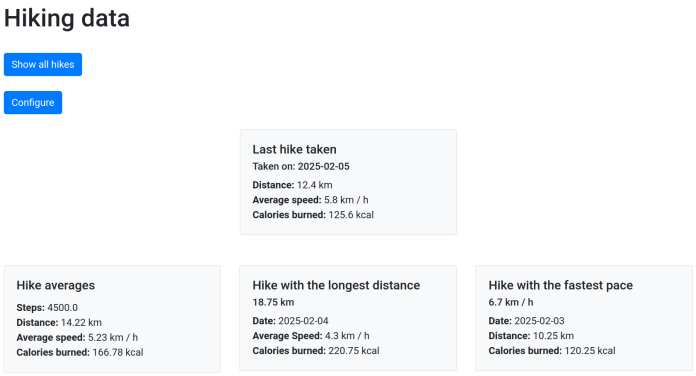

Figure 5: Web application main view

3.2.2 Past hikes view

The past hikes view includes a table that lists all past hikes in persistent memory. For each hike the following information is shown:

- ID (automatically generated for the database)
- Date

- Distance
- Steps
- Calories
- Average Speed

Each table row also includes a `Delete` button, that allows the user to delete the hike in question.

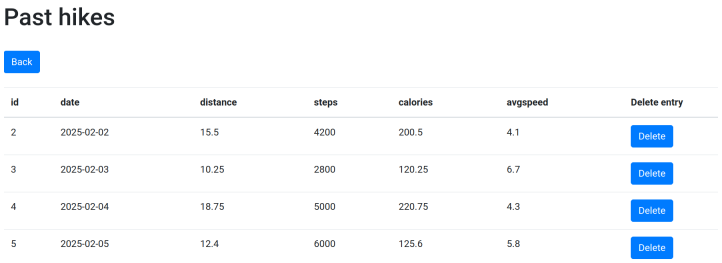

Figure 6: Web application past hikes view

Pressing Delete opens a confirmation Monad. By pressing delete, the action is confirmed. By pressing cancel, confirmation Monad is closed and no action is taken.

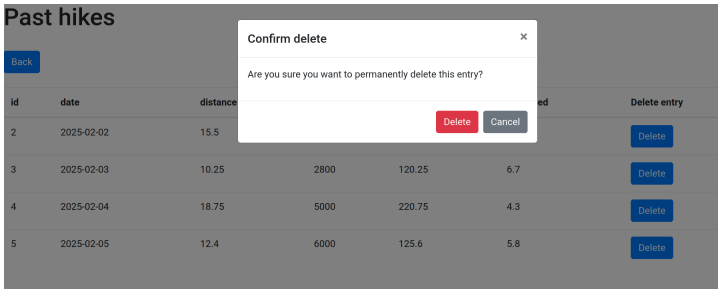

Figure 7: Web applicatoin confirm deletion

3.2.3 Configuration view

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

3.3 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.

3.3.1 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

3.3.2 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

3.3.3 Configuration view

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

4 Communication between devices

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