ToRaDes

Toolbox for Rapid Development of Smart Homes and Assistive Technologies

Requirements Specification

FABI3 - Flexible Assistive Button Interface

Benjamin Aigner (BA)
University of Applied Sciences Technikum Wien
Höchstädtplatz 6, A-1200 Vienna, Austria
Email: beni, chris
@asterics-foundation.org

Version 0.2 from 10.10.18

Gefördert von

Document Information

Issue Date	**10.10.18**
Status	**Review**
Dissemination Level	**PU**
Document ID	**functional_requirements_FABI**

ToRaDes

Toolbox for Rapid Development of Smart Homes and Assistive Technologies
This project has been funded by the Municipal Department 23 of the City of Vienna (Economic Affairs, Labour and Statistics - MA 23) in course of the 18th Call "Competence Teams for Research and Teaching".

AsTeRICS Foundation

The AsTeRICS Foundation is a non-profit organisation that promotes the development and application of Open Source Assistive Technology (AT) for people with disabilities. In course of several research projects, hardware- and software solutions have been developed at the University of Applied Sciences Technikum Wien, that can be easily adapted to individual needs and implemented at low cost. These solutions include devices for alternative computer input, augmentative and alternative communication (AAC) and environmental control systems – but also low-tech tools supporting activities of daily living. The AsTeRICS Foundation delivers these resources to users and other stakeholders.

Disclaimer

Revision History

Revision	Date	Author(s)	Description
0.1	20.04.18	BA	Created
0.2	15.05.18	BA	Minor updates

Contents

1 Introduction **5**
- 1.1 Purpose . . . 5
- 1.2 Scope . . . 5
- 1.3 Background . . . 5
- 1.4 References . . . 5

2 Functional Requirements **6**
- 2.1 Context . . . 6
- 2.2 User Requirements . . . 6
- 2.3 Data Flow Diagrams . . . 7
- 2.4 Functional Requirements . . . 8
  - 2.4.1 Functional Requirements Group 1 - general . . . 8
  - 2.4.2 Functional Requirements Group 2 - Keyboard/Mouse/Joystick actions . . . 8
  - 2.4.3 Functional Requirements Group 3 - Button processing . . . 9
  - 2.4.4 Functional Requirements Group 4 - Interfaces . . . 9
  - 2.4.5 Functional Requirements Group 5 - Configurations (Slots) . . . 10
  - 2.4.6 Functional Requirements Group 6 - Infrared remotes . . . 10

3 Other Requirements **13**
- 3.1 Interface Requirements . . . 13

4 Glossary **14**

1 Introduction

The FABI (Flexible Assistive Button Interface) allows control of a computer's mouse cursor and typing desired keyboard keys by using buttons and special/individual input methods. It can be helpful for people who cannot use standard computer input devices – enabling them to play games, surf the internet, write emails and much more.

1.1 Purpose

The FABI Interface can be actuated via dedicated buttons, momentary switches or self-made electrical contacts. FABI consists of a hardware device and a graphical software application for configuration of the desired functions.

1.2 Scope

This requirements specification is used to determine functional requirements (FR) and non-functional requirements (NFR) for FABI device. The document is used in conjunction with the medical product classification and the risk management document.

1.3 Background

The AsTeRICS Foundation is a non-profit organisation that promotes the development and application of Open Source Assistive Technology (AT) for people with disabilities. Currently available button interfaces either lack of flexibility in context of configuration by the user or in usability with many different platforms. In addition most of these assistive devices are very expensive.
FABI is designed as low-cost button interface, which can be bought as finished product or built in a DIY manner.

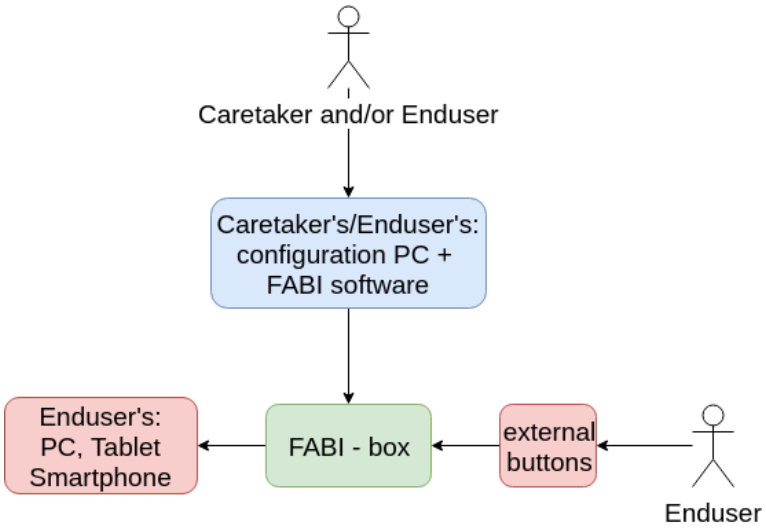
1.4 References

Note: Currently no references here.

2 Functional Requirements

2.1 Context

The context of this requirement specification is limited to the FABI box (highlighted in green), a hardware module, and to the FABI software (FABIGUI, highlighted in blue) which is used to configure the FABI box.



The external buttons are not scope of the FABI project, either they are bought off-the-shelf or they are built in a DIY manner. The PC which is controlled by the FABI box is also not scope of this document.

2.2 User Requirements

U1 - Compatibility The FABI box should work with all major smartphone and PC platforms. This includes Microsoft Windows, Apple macOS, Linux, Apple iOS, Google Android. In addition, the IR remote part should work with most available consumer electronic devices, which can be controlled via IR remotes.

U2 - Input options The FABI box should be able to trigger mouse, keyboard and joystick actions on the connected device.

U3 - Connections This box should be connected either via a USB compatible interface or via a wireless Bluetooth connection.

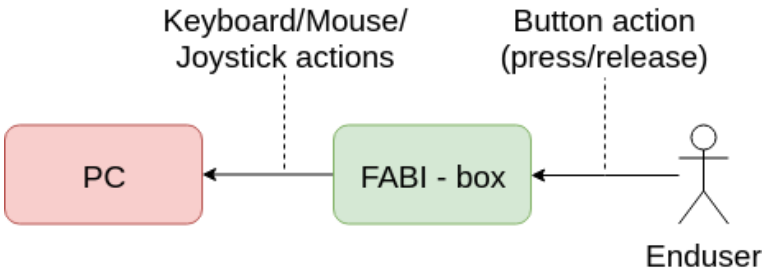
U4 - Buttons External buttons should be connected via a quasi-standard 3.5mm jack plug, which is used in nearly all available assistive buttons.

U5 - Configuration The FABI box should store the configuration, which is used by the device itself with no further interaction necessary.

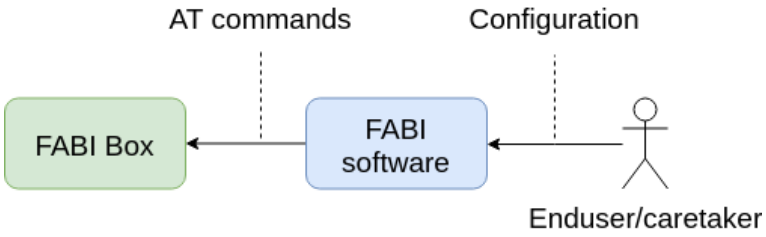
U6 - Configuration interface The FABI box configuration should be configurable via two different possibilities: on the one hand, it should be compatible with previous FABI releases and their corresponding PC GUI. On the other hand, the FABI box itself should be able to provide a WiFi hotspot, which can be used to open a configuration page via a webbrowser.

U7 - Slots The FABI box should be able to switch between different configurations, where the switching should be triggerable by the end user.

2.3 Data Flow Diagrams



During normal operation, the enduser triggers actions be pressing (and releasing) the attached buttons. The button actions are transformed to Keyboard, Mouse or Joystick actions (e.g., press left mouse button, click with left mouse buttonm, write a text, press/release a joystick button), determined by the current configuration. The triggered action should transfer HID commands to the PC/smartphone device.
These HID can be transferred either via a USB compatible interface or via a Bluetooth connection.



If the user, a caretaker or any other authorized person needs to adapt the FABI configuration, two possibilities can be used:

FABI software This configuration GUI is used for all FABI devices (also previous versions 1 and 2). This software needs a USB connection, which is used to transfer so called *AT commands* to the FABI device, which updates its configuration. The configuration user needs to connect the FABI box to his/her computer (this GUI is Microsoft Windows only), open the GUI, change the parameters and save it to the FABI box.

Web configuration The FABI device is able to open a WiFi hotspot, to which a configurator can connect to. This WiFi hotspot is only enabled by a dedicated action, avoiding any unauthorized access. This configuration possibility is available for any WiFi device which has a browser for opening the config page.

2.4 Functional Requirements

2.4.1 Functional Requirements Group 1 - general

General requirements overview.

Table 2.1: Requirements Group 1 - general

Section/ Requirement ID	Requirement definition
FR1.0	The device shall be able to translate button presses into Keyboard-/Mouse/Joystick actions [U2][FR2]
FR1.1	The device shall be controlled via assistive buttons (connected via jack plugs) [U4][FR3]
FR1.2	The device shall be connected either by a cable or a wireless interface, using standardized interfaces [U3][FR4]
FR1.3	The device shall be able to store configuration on itself and switch between them [U5][U7][FR5]
FR1.4	The device shall be configured by either a PC GUI or a Web interface [U6][FR6]
FR1.5	The device shall be able to control consumer electronics via infrared commands [U1][FR7]
FR1.6	The device shall be able to work with: Linux, Apple macOS, Microsoft Windows, Android, iOS [U1]

2.4.2 Functional Requirements Group 2 - Keyboard/Mouse/Joystick actions

Requirements related to the output possibilities for triggering mouse, keyboard and joystick actions.

Table 2.2: Requirements Group 2 - Keyboard/Mouse/Joystick actions

Section/ Requirement ID	Requirement definition
FR2.0	The device shall be able to trigger a mouse left click
FR2.1	The device shall be able to trigger a mouse double left click
FR2.2	The device shall be able to trigger a mouse right click
FR2.3	The device shall be able to trigger a mouse middle click
FR2.4	The device shall be able to trigger a mouse wheel up/down
FR2.5	The device shall be able to change the mouse wheel steps
FR2.6	The device shall be able to move the mouse cursor in X and Y axis
FR2.7	The device shall be able to write an arbitrary text on the connected device
FR2.8	The device shall be able to press/release an arbitrary keyboard key
FR2.9	The device shall be able to set different joystick axis
FR2.10	The device shall be able to press/release different joystick buttons
FR2.11	The device shall be able to modify the joystick hat position

2.4.3 Functional Requirements Group 3 - Button processing

Button related requirements, defining the abilities for handling buttons.

Table 2.3: Requirements Group 3 - Button processing

Section/ Requirement ID	Requirement definition
FR3.0	The device shall be able to use standard off-the-shelf assistive buttons
FR3.1	The device shall be able to use DIY buttons
FR3.2	The device shall use a quasi-standard 3.5mm jack plug for button connections
FR3.3	The device shall be able to recognize a button press
FR3.4	The device shall be able to recognize a button release
FR3.5	The device shall be able to process a long button press
FR3.6	The device shall be able to have different delays for press, release and idle times (anti-tremor)

2.4.4 Functional Requirements Group 4 - Interfaces

Requirements related to the communication interfaces for normal operation (not configuring the device).

Table 2.4: Requirements Group 4 - Interfaces

Section/ Requirement ID	Requirement definition
FR4.0	The device shall use a USB compatible connection for a wired operation
FR4.0.0	The device shall use a USB-HID profile for highest possible compatibility
FR4.0.1	The device shall use a USB-CDC profile for configuration
FR4.0.2	The device shall implement a so-called composite device to support HID and CDC concurrently.
FR4.0.2.0	The device shall be able to switch on or off different parts of the HID profile, enabling maximum compatibility to different specialized devices
FR4.0.3	The device shall be controlled via standardized AT commands
FR4.0.4	The device shall be able to trigger the same actions via buttons and via AT commands
FR4.1	The device shall use a Bluetooth LowEnergy connection for a wireless operation
FR4.1.0	The device shall use the HID-over-GATT (HOG) Bluetooth profile for triggering HID actions
FR4.1.1	The device shall implement the same HID commands for Bluetooth and USB
FR4.2	The device shall be powered via the USB compatible connection, even if in wireless mode.

2.4.5 Functional Requirements Group 5 - Configurations (Slots)

Requirements related to the handling of different configurations in so called slots.

2.4.6 Functional Requirements Group 6 - Infrared remotes

Requirements related to infrared (IR) remote unit (receiving and transmitting)

Table 2.5: Requirements Group 5 - Slots

Section/ Requirement ID	Requirement definition
FR5.0	The device shall be able to store a minimum of 10 different configurations
FR5.1	The device shall be able to modify the current configuration permanently
FR5.2	The device shall be able to modify the current configuration temporarily
FR5.3	The device shall be able to modify the configuration via the USB compatible interface
FR5.4	The device shall be able to modify the configuration via the WiFi interface
FR5.5	The device shall be able to assign all meaningful actions to all input modalities
FR5.6	The device shall store different properties into one configuration
FR5.6.0	The device shall store an unique name for each configuration
FR5.6.1	The device shall store the action for each triggered button (or any other input modality)
FR5.6.2	The device shall store long press / idle times (anti-tremor settings) individually for each slot
FR5.6.3	The device shall store infrared commands independent from configuration settings
FR5.7	The device shall load a configuration by name
FR5.8	The device shall load a configuration by navigating (next/previous)
FR5.9	The device shall load a configuration by number
FR5.10	The device shall be able to delete one, more or all configurations
FR5.11	The device shall be able to print out one or all configurations via USB compatible or WiFi interface

Table 2.6: Requirements Group 6 - Infrared remotes

Section/ Requirement ID	Requirement definition
FR6.0	The device shall be able record usually used 38kHz infrared remote codes
FR6.0.0	The device shall be able to store at least 50 different remote codes
FR6.0.2	The device shall be able to delete one, more or all IR commands (identified either by name or number)
FR6.0.3	The device shall be able to print the received IR command via USB compatible or WiFi interface
FR6.1	The device shall be able to replay previously recorded infrared remote sequences
FR6.1.0	A recorded IR code shall be replayed, identified by either a name or a number
FR6.1.1	The device shall be able to replay an arbitrary code sequence, with the same format as sent in FR6.0.3

3 Other Requirements

3.1 Interface Requirements

The main interface for this device is the USB compatible connection and/or the Bluetooth interface. Due to the standardized interfaces and protocols, following documents are referenced here (this device shall implement these referenced documents as they are an integral part of this document):

Device Class Definition for Human Interface Devices (HID), v1.11 This document describes the necessary requirements for a USB HID compatible device. These specifications need to be implement within the firmware. USB low-level specifications are met by the used microcontroller.

Universal Serial Bus Class Definitions for Communications Devices, v1.2 This document specifies the interface for communicate via a virtual serial port over the USB compatible interface.

Bluetooth v4.2 core specification This document specifies the Bluetooth interface. Implementation is provided by the selected microcontroller.

Bluetooth HID over GATT Profile, v1.0 Profile specification to tunnel HID commands over a Bluetooth connection. This profile is implemented by this device.

In addition to these standardized profiles/specifications, this device is implementing a command API, which is used to modify the configuration or control the device. The interface requirement is part of Doxygen documentation. The currently valid API is available in this repository in */docs/html/README__AT__command__API_8md.html* (or in raw markdown: */main/README_AT_command_API.md*).
Note: This AT command API is used for different devices. Please consult the documentation which commands are used for this device.

4 Glossary

AT (commands)	Serial command API, starting each command with text sequence "AT"
AT	Assistive Technology
API	Application Programming Interface
CDC	Class Definition for Communications (USB specification)
ESP32	Application processor by Espressif
FABI	Flexible Assistive Button Interface
GATT	Generic Attributes (Bluetooth Low Energy services)
HID	Human Interface Device
HOG	HID over GATT (Bluetooth profile)
IEEE	Institute of Electrical and Electronics Engineers
USB	Universal Serial Bus
WiFi	Synonym for wireless networking based on IEEE 802.11