

## Project application

### ***FABI (Flexible Assistive Button Interface) using Voice***

This document contains confidential information about the project FABI (Flexible Assistive Button Interface) using Voice and may only be passed on within the FH Technikum Wien project team and to the persons explicitly listed in the distribution list below.

#### Project team [alphabetical]

Name	E-mail	Telephone (optional)
Anton Altmeyer	be22x013@fh-technikum.at	+49/1628522581
Christian Tanzer	ew22b031@technikum-wien.at	
Norbert Szabo	ew21b041@technikum-wien.at	

#### Project supervisors, customers, taught [alphabetically].

Name	E-mail	Telephone (optional)
Oliver Klein	kleino@technikum-wien.at	
Alija Sabic	sabic@technikum-wien.at	

#### Document Development

Version	Description	Date
1.0	First creation of the document.	31.03.2023
2.0	Extending the "Working Package"	19.04.2023
3.0	Editing the workflow for ML model	28.05.2023
4.0	Editing the workflow for the adaption of the firmware	10.06.2023

Version	Author	File name	Last change	Print date	Page
4.0	Anton Altmeyer Norbert Szabo Christian Tanzer	Documentation FAVI	25.06.2023	25.06.2023	1 / 4

## Table of contents

Project application .....	1
FABI (Flexible Assistive Button Interface) using Voice .....	1
Project team [alphabetical] .....	1
Project supervisors, customers, taught [alphabetically]. ....	1
Document Development .....	1
Table of contents.....	2
Project short description.....	2
Goals and project delimitation.....	3
Project team and team roles .....	3
Rough schedule and priorities n.....	4

### Project short description

The open-source project FABI (Flexible Assistive Button Interface, cf. [1]) provides users with the ability to emulate mouse and keyboard interaction with a PC via momentary switches.



The current implementation(s) use the *SparkFun Pro Micro* as a microcontroller to interface the buttons and to communicate with a PC via USB (cf. [2, 3]). In addition, a bluetooth module (cf. [4]) can be used to allow for communication with a PC via Bluetooth.

Users and tinkerers can choose between two variants, (w/ and w/o Bluetooth) to allow for cheaper variants, where Bluetooth functionality is not necessary. However, using a more powerful, recent microcontroller (with WiFi and Bluetooth support, out-of-the-box) could reduce costs of the variant w/ Bluetooth and allow for extended input or control functionality, respectively, using voice commands instead of or in addition to the momentary switches.

Version	Author	File name	Last change	Print date	Page
4.0	Anton Altmeyer Norbert Szabo Christian Tanzer	Documentation FAVI	25.06.2023	25.06.2023	2 / 4

## Goals and project delimitation

Develop a variant of FAVI, called *Flexible Assistive Voice Interface* or **FAVI**, based on the *Arduino Nano RP2040 Connect/Sense* to allow for USB and Bluetooth communication w/o the necessity of an add-on module. In addition, run a machine learning (ML) model, using TensorFlow Lite, to detect pre-defined voice commands (cf. [6]), as an alternative trigger for the momentary switches (only in SW), allowing users to control the switches only via voice. Therefore, built-in microphones need to be tested and issued. In future possible infrared interfaces (for Home Applications, etc.) or Messenger API (e.g., WhatsApp, Signal, etc.) can be implemented. In addition, different commands, e.g., Wakeup or Sleep.

## Project team and team roles

Name	Main responsibility	Team Roles
Anton Altmeyer	Project management	Project manager, hardware development, software development
Norbert Szabo	Software/Hardware	Hardware development, embedded software development, user documentation
Christian Tanzer	Software/Hardware	Hardware development, embedded software development, technical documentation

AP #	Work package	A. Altmeyer	C. Tanzer	N. Szabo			Person hours Estimate
<b>1000</b>	<b>Project preparation phase</b>						<b>17,5</b>
1100	Assemble team, division of roles	R	R	R			0,5
1200	Write project proposal	R					2,0
1300	Create PSP		R				4,0
1400	Create PAP			R			1,0
1500	Project planning (meetings)			R			10,0
<b>2000</b>	<b>Project implementation phase</b>						<b>72</b>
2100	Hardware build-up of FABI device		R	R			5,0
2200	Testing the current FABI software using buttons (usage of Arduino Pro Micro MC)		R	R			4,0
2300	Flashing the FABI software on the new device (Arduino RP 2040 Sense)		R	R			3,0
2400	Testing the FABI software with breadboard using new device (Arduino Nano RP 2040 Connect)		R	R			4,0
2500	Fixing occurring errors		R	R			8,0

Version	Author	File name	Last change	Print date	Page
4.0	Anton Altmeyer Norbert Szabo Christian Tanzer	Documentation FAVI	25.06.2023	25.06.2023	3 / 4

## Project application

### FABI (Flexible Assistive Button Interface) using Voice

2600	Implementing "End-to-end tinyML audio classification" with PC Tutorial: <a href="https://blog.tensorflow.org/2021/09/TinyML-Audio-for-everyone.html">https://blog.tensorflow.org/2021/09/TinyML-Audio-for-everyone.html</a>	R					20,0
2700	Extending/Changing the FAVI firmware (functionally) → Understanding functionalities of display, neopixel and buzzer	R	R	R			5,0
2800	Extending/Changing the FAVI firmware (functionally) → Extending the functionalities of the firmware to also use it for the Arduino Nano RP2040 Connect	R	R	R			12,0
2900	Extending the ML model of audio classification for FAVI properties (Transfer Learning) → Adding more possible words to be classified	R					6,0
3000	Testing/Validating the FAVI software and device		R	R			5,0
<b>4000</b>	<b>Project follow-up phase</b>						<b>15</b>
4100	Project evaluation	R	R	R			5,0
4200	Create final presentation	R	R	R			10,0

### Rough shedule and priorities

#### Meeting 1:

By the end of April, the FABI device should be set up and the current FABI software version should have been tested on the device using Buttons. Until mid of May, different ML models will be derived and maybe by end of May tested by developing the FABI software for using voice recognition with these ML models.

#### Meeting 2:

Due to problems with implementing the ML model, the model only has an accuracy of 24% and the "Transfer Learning" part referred to issues regarding the dependencies. Therefore, the development of the model needed to be stopped and we considered focusing more on extending the firmware functionality of the display, neopixel and buzzer. We considered to use the Adafruit libraries to implement the functions for the use of the Arduino Nano RP2040 Connect.

Version	Author	File name	Last change	Print date	Page
4.0	Anton Altmeyer Norbert Szabo Christian Tanzer	Documentation FAVI	25.06.2023	25.06.2023	4 / 4