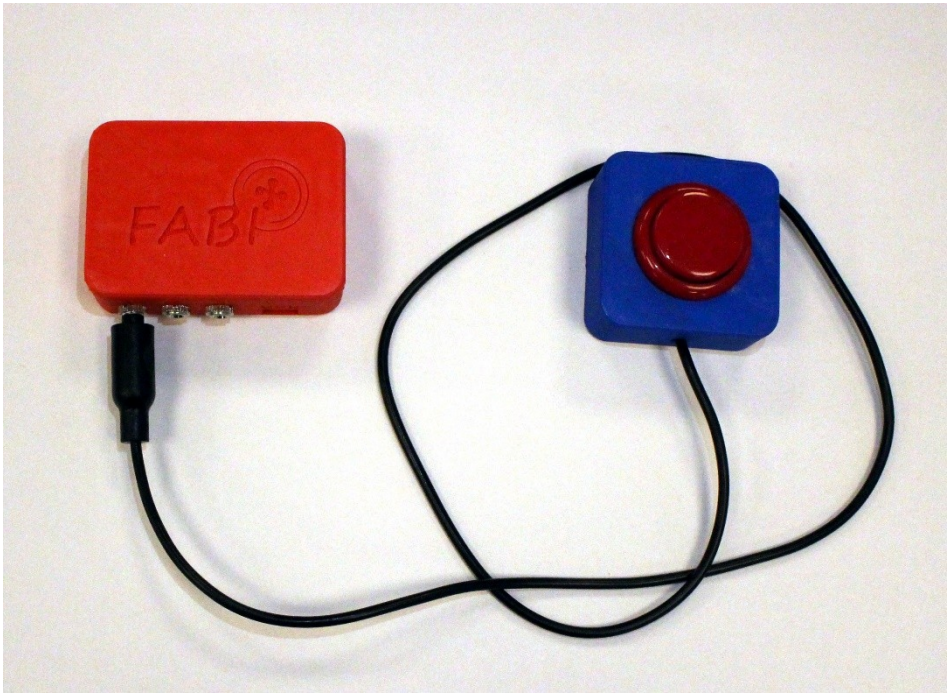
FABI – Flexible Assistive Button Interface

User Manual



AsTeRICS Foundation

office@asterics-foundation.org

www.asterics-foundation.org/fabi

Welcome to FABI

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.
The FABI Interface can be actuated via dedicated buttons, momentary switches or self-made electrical contacts. FABI consists of a hardware module (a low-cost microcontroller which behaves as a computer mouse and/or keyboard) and a graphical software application for configuration of the desired functions.

This user manual includes a "do-it-yourself" building guide for your personal FABI device, describing the necessary hardware components and the Graphical User Interface application for the configuration of the different functions. A configured FABI module can be used on any computer (Windows, Linux or Mac) without the installation of special software, because the FABI module behaves exactly like a standard mouse and keyboard combination, which is simply plugged into your computer. Nevertheless, to configure the desired functions via the FABI GUI application, it is necessary to install a driver. The installation of the driver is explained in the software section of this manual.

FABI is a free Assistive Technology module developed by the AsTeRICS Academy Project of the University of Applied Sciences, Technikum Wien. The AsTeRICS Academy Project ended in 2016. In 2017 the non-profit organization AsTeRICS Foundation was founded in order to continue the distribution and development of the researched technologies and projects.
(see http://www.asterics-foundation.org).

All software and hardware documents are free and we tried to use the most affordable components available on the market to establish the desired functionalities – making FABI the most reasonably priced flexible assistive button interface we know of!!

Contents

Building the Hardware

Building your own FABI interface for computer control is easy! Here, we will show how to build your own FABI and buttons.

A working FABI device consists of a microcontroller with USB cable and some additional electrical connections (wires, switches, buttons) – which can be mounted in a way so that they are easily accessible for the user(s).

For a detailed description of the processes, more information about the components and alternative ways to build your own FABI device please visit the FABI github (https://github.com/asterics/FABI) and refer to the more detailed manual (https://github.com/asterics/FABI/blob/master/FABI%20flexible%20assistive%20button%20interface.docx)

Required Material and Tools

For the FABI Box:

- 1x Arduino Pro Micro
- 8x 3.5mm Jack-plugs (PJ-392)
- About 60cm (24in) of wire cable
- 3D-printed casing (STL files are available here: https://github.com/asterics/FABI/tree/master/Case%20Designs/3D-printer)
- 1x Screw (M3x6 - M3x10)

For each FABI button:

- 1x push button
- 1x Jack-plug cable
- 1x 3D-printed casing

Required Tools:

- Hot glue gun
- 3D-printer (or you can order the FABI-Kit, which includes the 3D-Printed cases)
- Soldering iron and soldering tin
- Wire cutter
- Pliers

Building your own FABI Box

1- Place the Arduino micro in the casing so the USB-Micro plug faces the hole on the side of the casing.

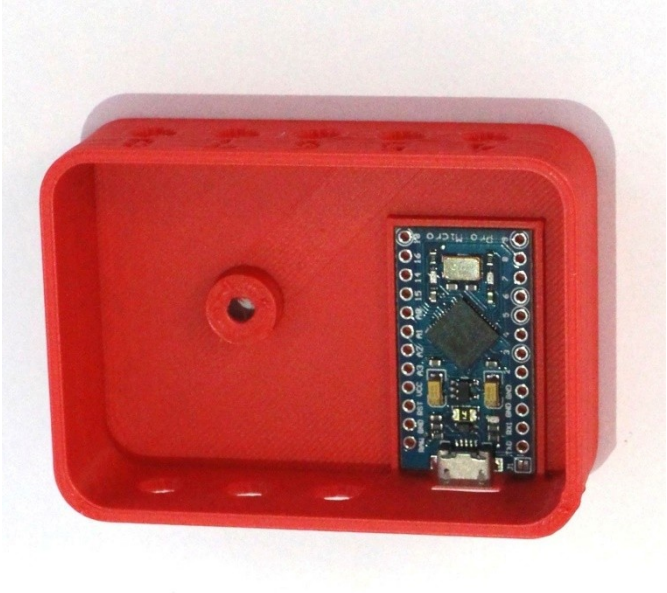


Figure 1 The Arduino Pro Micro microcontroller is placed inside the FABI box

2- Apply a generous amount of hot glue to the side of the Arduino closer to the middle (pins RAW to pin 10) as well as above the USB-Micro connector. Make sure not to apply any on the pins (TX0 to pin 9) the other side of the Arduino.

Figure 2 Applying hot glue to secure the Arduino Pro Micro

3- Mount three of the jack-plug connectors to the holes in the casing numbered 1, 2 and 3. (Some 3D-printers use support material when printing holes, therefore the jack plugs might not fit in the holes and it might be necessary to use a tool like for example a screwdriver to remove the extra material from the holes).

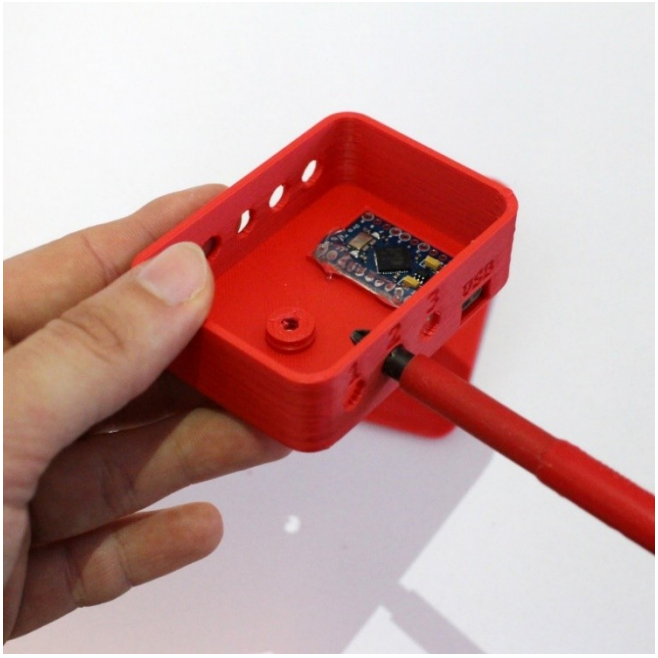


Figure 3 Removing excess material from the holes for the jack-plugs

For the next steps it is important to know that the jack plugs have three pins, which are numbered counterclockwise starting with the longest pin.

Figure 4: 3,5 mm Klinkenstecker 3-polig Stereo Panel Montage Löten Anschluss Mit Sicherungslasche Mutter, kenable Ltd
Online Access: https://www.kenable.co.uk/de/audio-kabel/audio-adapter/5674-35mm-jack-3-pole-stereo-panel-mount-solder-terminal-with-locking-nut-5055383456741.html (Last access: August 21st, 2019)

4- Stick the jack Plug connectors through the holes with the pins facing inside. and secure them with the nut. The longest of the pins (pin 1) should be closest to the bottom of the casing as shown in the picture.

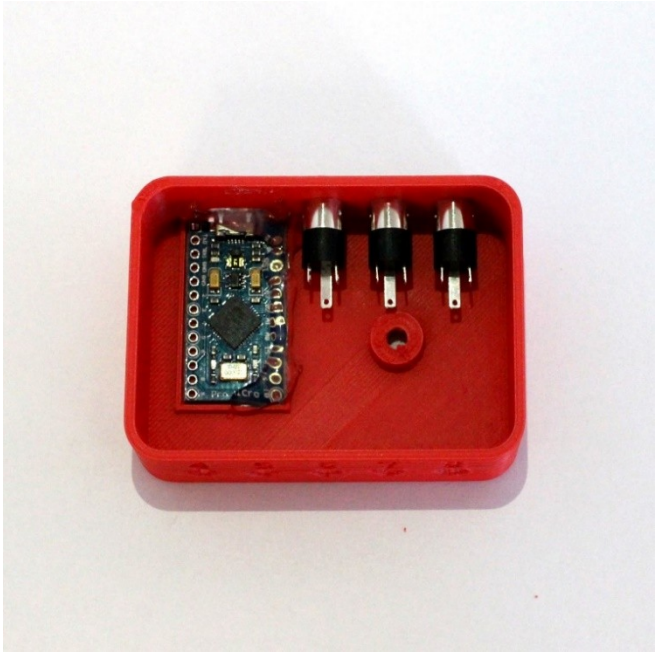


Figure 5: The first three jack-plugs are placed in FABI Box

5- Apply hot glue to secure the jack plugs.

Figure 6: The jack-plugs are secured with hot glue

6- Cut three pieces of wire; about 4,5cm, 5,5cm and 6,5cm, and remove the insulation from the ends. Solder the longest wire to pin **D2** on the Arduino, the 5,5cm cable to pin **D3** and the shortest one to pin 4. Solder the wire connected to pin **D2** on the Arduino, to pin 2 of the jack-plug which is numbered 1 on the casing. Solder the wire connected to pin **D3** on the Arduino, to pin 2 of the jack-plug labeled 2 and the wire connected to pin **D4** on the Arduino to the jack plug labeled 3 as shown in the picture.

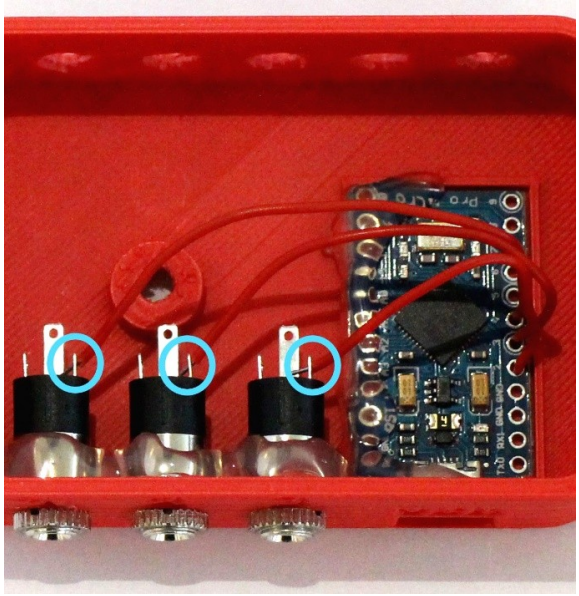


Figure 7: The wires are connected to pins 2-4 on the Arduino and pin 2 of the jack-plugs 1-3

7- Cut five wires with the lengths of about 4,5cm, 5,5cm, 6,5cm, 7,5cm and 8,5cm and remove the insulation from both ends.
Solder the shortest wire to pin **D5** on the Arduino, the next longer one to pin **D6** etc.

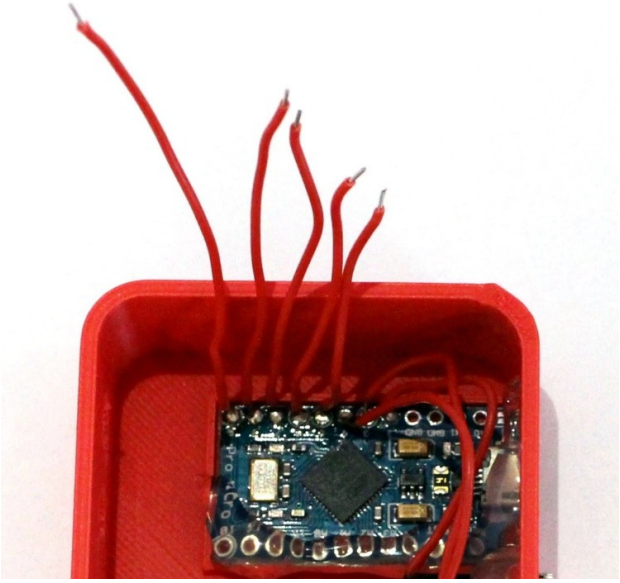


Figure 8: 5 wires are soldered to pins 5-9 on the Arduino

8- Mount the remaining five jack plugs to the FABI casing, with the longest pin (pin 1) closest to the bottom and secure them with hot glue.

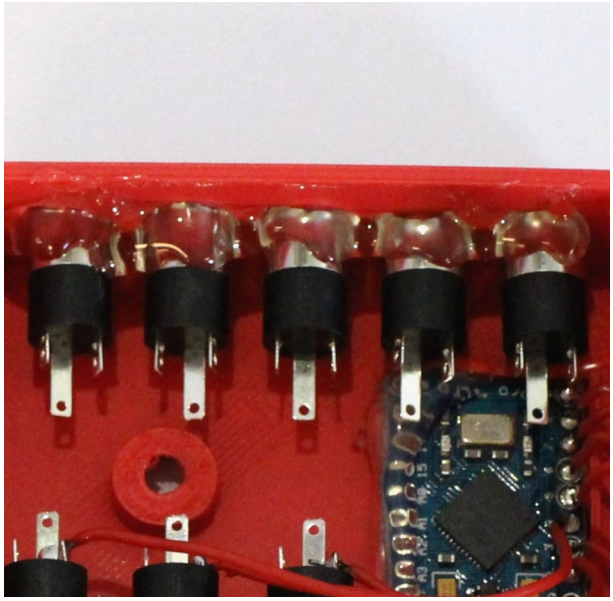


Figure 9: The jack-plugs 5-8 are placed in the FABI Box and secured with hot glue

9- Solder the cable connected to pin **D5** on the Arduino to pin 2 of the jack plug number 4, the cable connected to pin **D6** on the Arduino to jack plug number 5 and so on.

Figure 10 The wires connected to the pins 5-9 on the Arduino are soldered to pin 2 of the jack-plugs 4-8

10- Take a wire with a length of about 15cm (6in) (or use a wire and remove the entire isolation) and solder one end it to one of the Ground (GND) pins on the Arduino. Then solder it to pin 1 of each of the jack plugs starting with number 3 (as shown in the picture) and cut off the remaining wire.

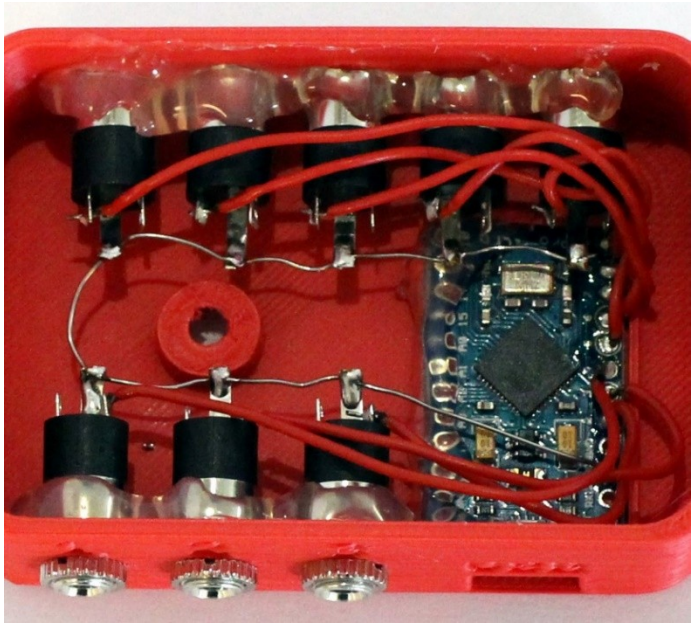


Figure 11: A wire is connected to the GND pin on the Arduino and pin 1 on the jack-plugs

Tip: You can cut the tip of the pin in half so the wire can be secured in between.

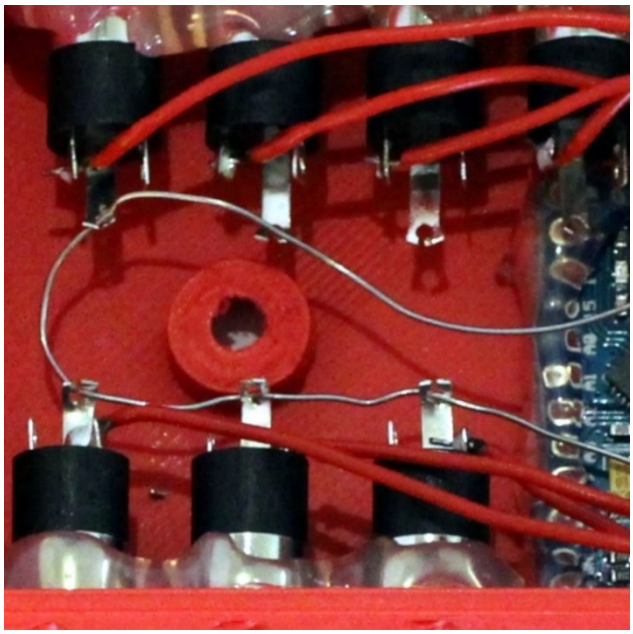


Figure 12: Securing the wire connecting the jack-plugs to the ground by putting it between the tips of the jack-plug pins

11- Use the screw to attach the top of the casing.

Figure 13: The finished FABI Box

Building your own FABI Buttons

1- Remove about 3cm (~1.2in) of the black cable insolation from jack plug cable. Remove the isolation off the tips of the yellow and the red cables. Cut off the white cable as it will not be needed.

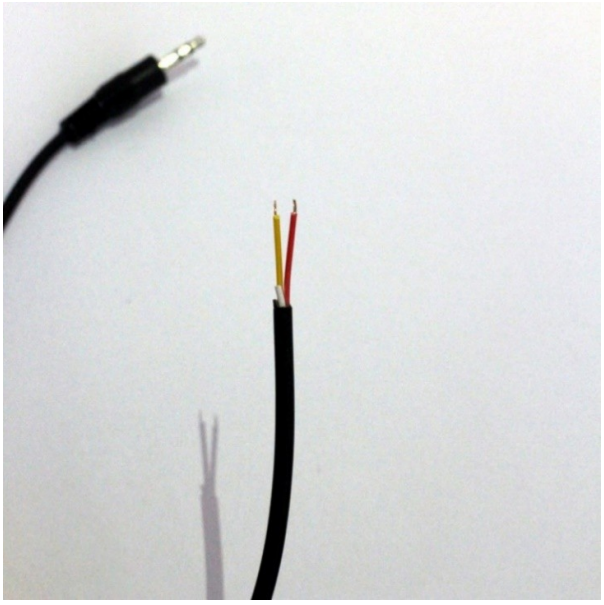


Figure 14: The insolation is removed from the tips of the jack-plug cable

2- Thread the cable through the hole on the side of the button case.

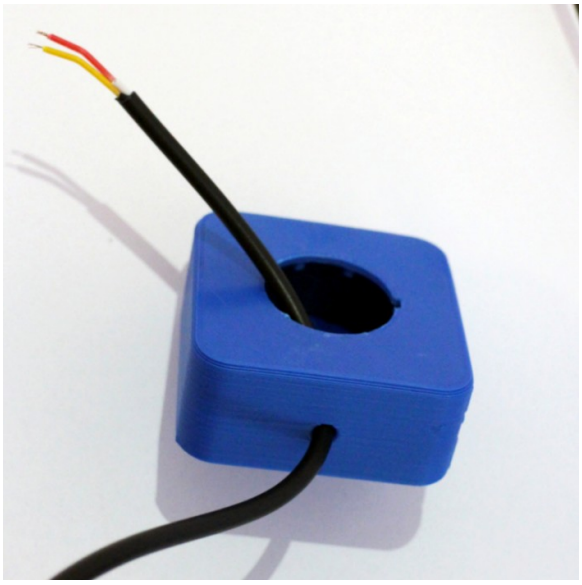


Figure 15: Step 2

3- Tie a knot in the cable approximately 5cm (2in) off the end (this will assure that strong pulling on the cable does not destroy the functionality of the button)

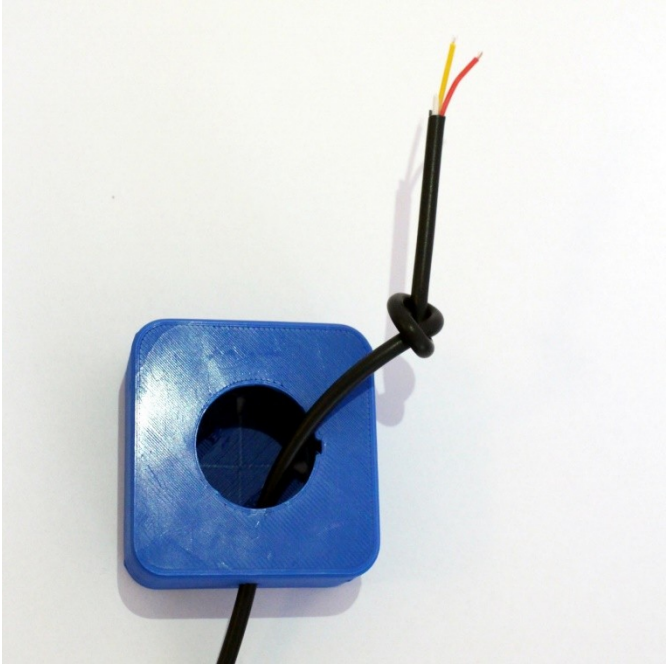


Figure 16: Step 3

4- Take the button and bend the terminals of the button to an angle of about 30° as shown in the picture.

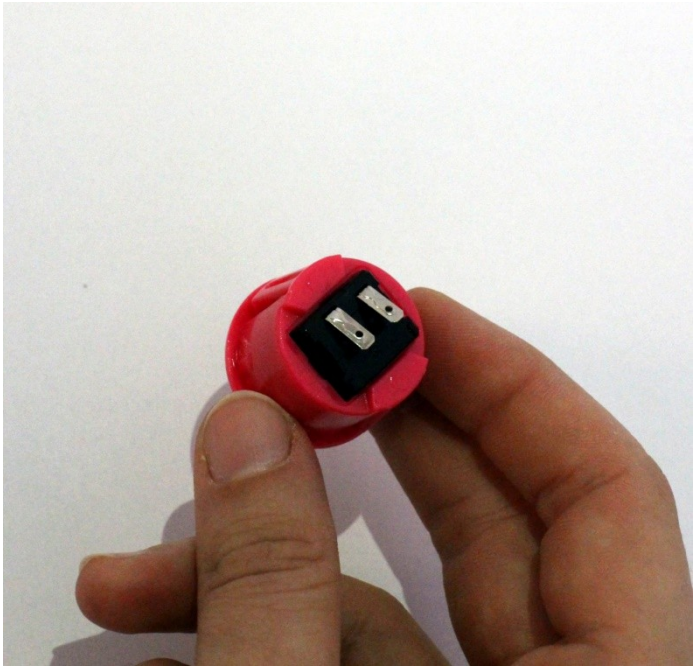


Figure 17: The terminals of the button are bent sideways so the button can fit into the box

5- Solder the red and the yellow cables to the feet of the button (it does not matter which cable is connected to which terminal)

Tip: soldering might be easier if you stick the tips of the cables into the holes in the terminal of the button.

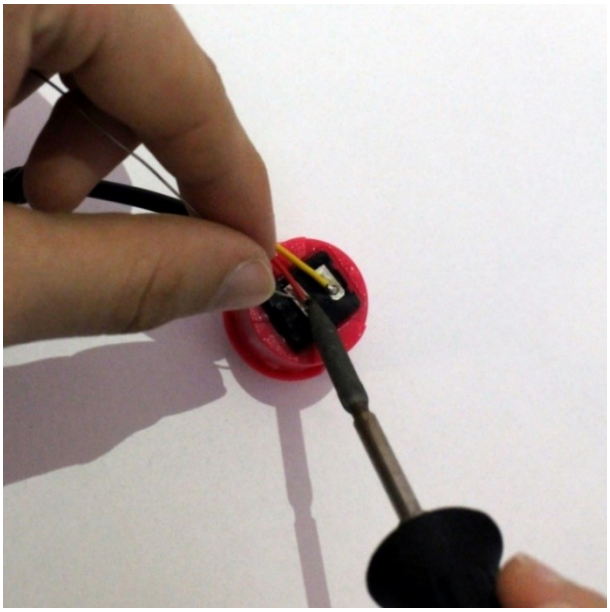


Figure 18: Connecting the red and the yellow wires of the jack-plug cable to the button

6- Push the button into the button case. Make sure the snaps on the side of the button are aligned with the hole on the side of the button case as shown in the picture.

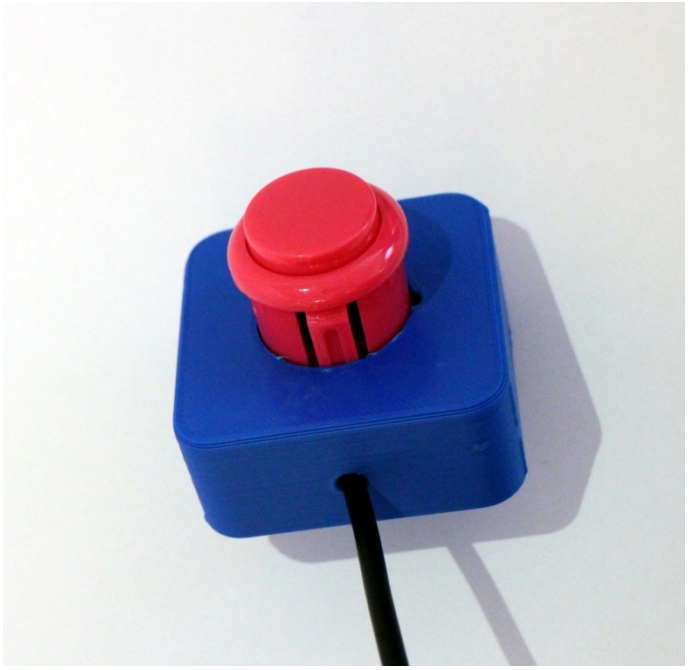


Figure 19: The correct orientation for pushing the button into the button case

Using the Software

The FABI hardware needs proper software for the configuration of the functionalities of the switches. The *FabiGUI.exe* can be downloaded from GitHub (https://github.com/asterics/FABI/releases/latest/download/FabiGUI.exe). The other files can be downloaded here: https://github.com/asterics/FABI/releases/latest/download/FABI.zip. The *FABI.zip* software package contains the following two main parts:

- **The software which runs in the microcontroller (the so-called "firmware"): It creates the desired mouse and keyboard inputs for the attached computer**
- **The FABI GUI software application which runs on a Windows PC. It communicates with the software on the microcontroller and can be used for the configuration of the desired functions of the attached switches**
- **This documentation/manuals**

This chapter will describe the necessary steps to install the software and to program the firmware into the microcontroller. Furthermore, the features of the FabiGUI application and how to define the desired behavior of the FABI device will be explained!

Installation

After downloading the *FABI.zip* software package from the above sources, extract the zip package to a folder of your choice. Try to start the *FabiGUI.exe* application. If the startup fails with an error message, the Microsoft .Net framework is not installed on your computer – in this case please download and install .Net from the following webpage: http://www.microsoft.com/en-us/download/confirmation.aspx?id=17718. In case you have any problems or questions regarding the software download or installation, feel free to write us an email: office@asterics-foundation.org

Notes for macOS & Linux

If you want to run the FabiGUI.exe on Linux or macOS, please download the Mono framework: https://www.mono-project.com/download/stable/

There might be minor differences if you use Arduino/FabiGUI.exe on Linux/macOS. Especially the COM ports are named differently (e.g., „ttyACM0“ on Linux).

Loading the firmware onto the microcontroller

To load the firmware onto the microcontroller we will use the Arduino IDE (Integrated Development Environment). You can download the Arduino IDE from http://www.arduino.cc.

After installing the Arduino IDE, attach the microcontroller (Arduino Pro Micro) via an USB micro cable to a COM Port on your computer (the following screenshots show a Windows PC, but the procedure is similar for Linux and MAC)

Connect this end to the microcontroller. Be careful not to break the connector – especially if using a cheap Chinese clone board ;-)

Connect this end to a USB port on your computer

Figure 20: Micro USB-cable

Making sure that the device is securely connected to the computer via the USB cable. To install the drivers please follow the instructions here: http://arduino.cc/en/Guide/ArduinoLeonardoMicro

Now start the Arduino IDE and open the *FabiWare.ino* file which is located in the FabiWare subfolder of the download software package. Select your **COM-port** number in the menu: **Tools-> Serial port**:

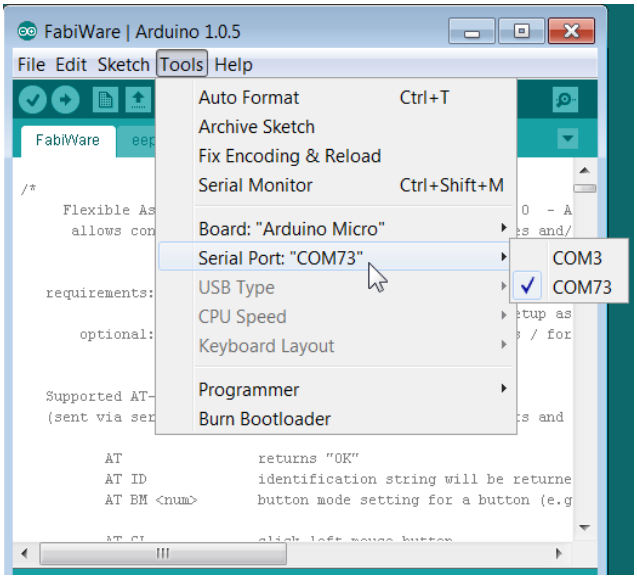


Figure 21: COM Port selection in the Arduino IDE

Then, select “**Arduino Micro**” in the menu **Tools->Board**:

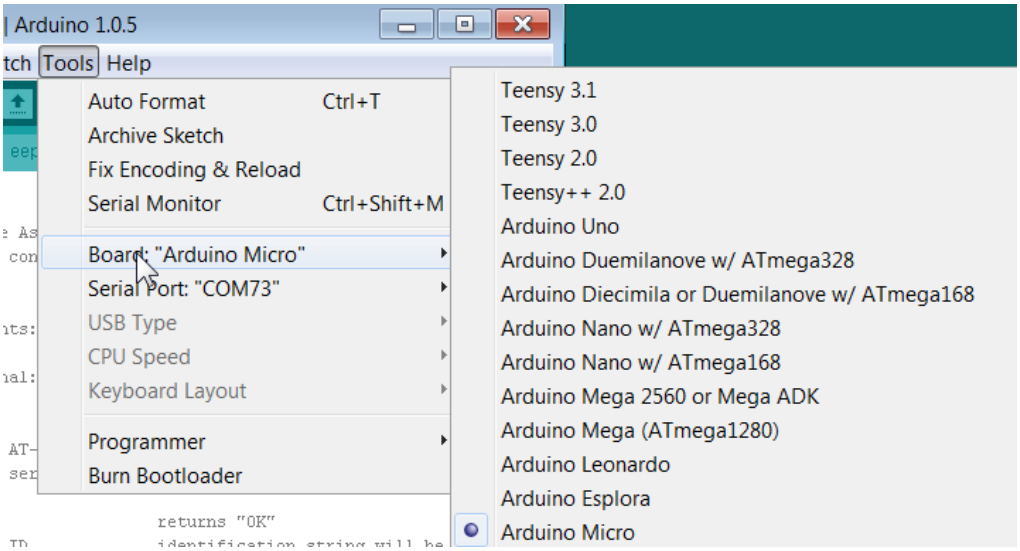


Figure 22: Select the correct microcontroller type in the board menu of the Arduino IDE

Now you can load the software onto your Arduino by clicking on the “Upload”-Button in the Arduino IDE. After several seconds, the message “done uploading” should appear.

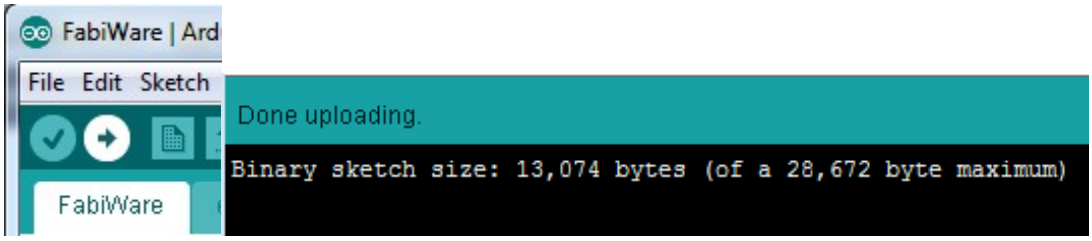


Figure 23: Using the Upload button in the Arduino IDE to compile and flash the firmware

Your computer should now perform the desired mouse- and keyboard activities, if you press the associated buttons! In case of problems please contact: office@asterics-foundation.org

Using the FABI GUI application

When the FABI firmware has been programmed into the microcontroller, you can use the FABI GUI application to configure your desired FABI functions. After starting FabiGUI.exe, the following window should open:

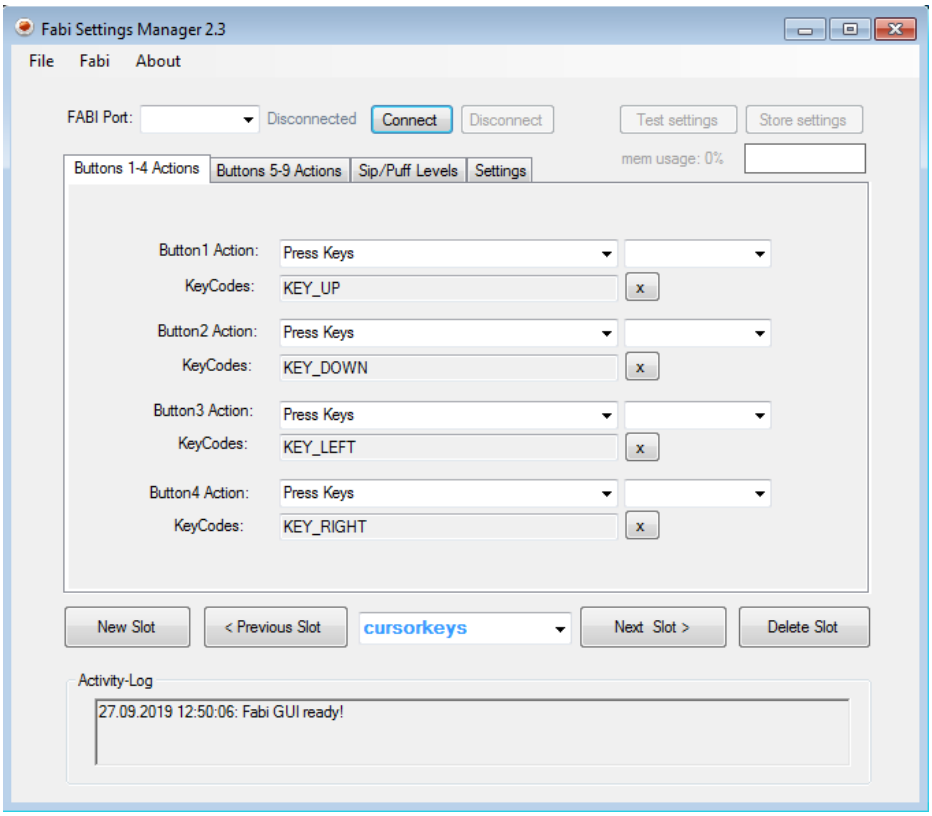


Figure 24: FABI user interface

Connecting the FABI device:

In order to be able to use the features of the FABI GUI, the FABI device must be connected to the application. To connect the device, follow the following steps:

1. Make sure your device is securely connected to your computer.
2. Select the appropriate COM port (communication port) in the combo box at the top of the application window. If the combo box does not show your noted COM Port number, please reconnect the device and wait for the COM ports to be updated, and then click on the drop menu to refresh the COM port list or restart the application.
3. Once the COM port is selected, click the “Connect” button on the right-hand side of the combo box. Once the device is connected, a confirmation message will appear in the activity log at the bottom of the application window (as shown in the picture below)

4. You can decide if you want to proceed with settings FROM the FABI device (click yes) or start with fresh settings (click no).

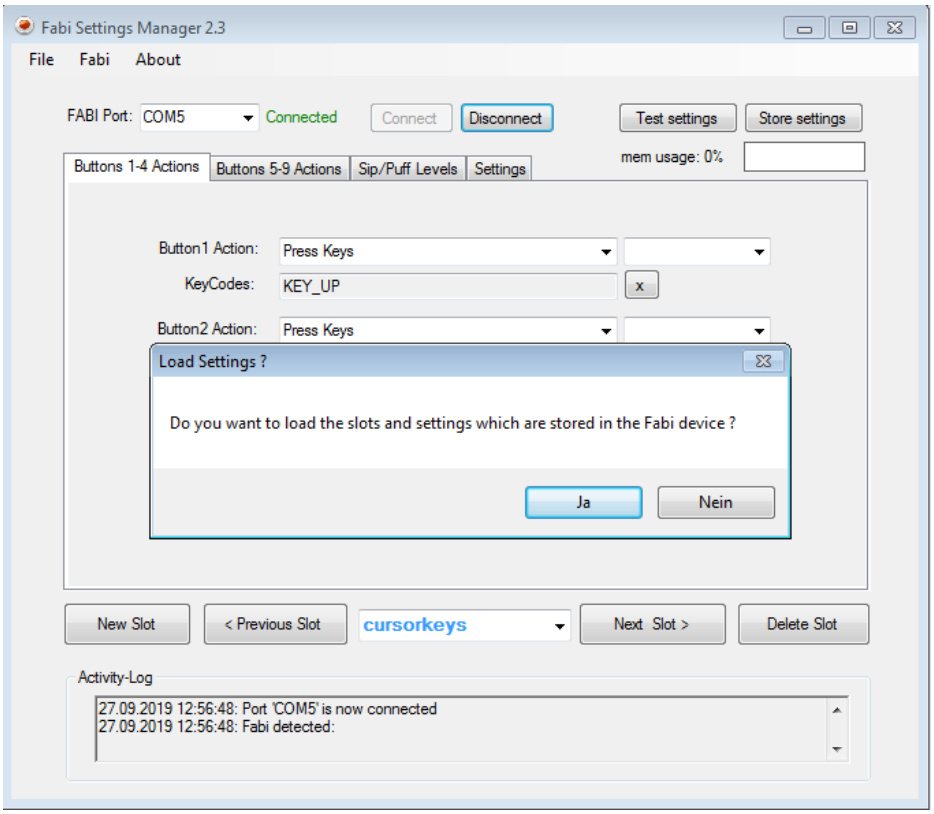


Figure 25: Program connected to the right port of the microcontroller

Port Status

The port status is located at the top / middle of the application window. It displays whether the device is currently connected or disconnected from the user interface. The functions of the user interface may only be used if the port status is “Connected”.

Activity Log

The activity log is located in the top right corner the application window. It provides messages in accordance to the use of the application.

Applying Settings

The selected functions for the (up to) 8 buttons will be activated when you click “Apply settings”. Once the settings have been applied, you will receive a confirmation message in the activity log, and you will be able to use the FABI device with the new configuration.

Attention: these settings are volatile. If you want to store settings permanently, press “Store settings”!

Saving, Loading and Clearing Slots

If you have selected Button-Function settings that you would like to use again, you can save them under a name in a memory slot of the microcontroller, so you can later re-load and use this configuration. Up to 4 memory slots are available. The configuration of the memory slots will be safed even if the microcontroller is unplugged from the USB cable/power supply. When you plug in the controller next time, the first slot will be loaded and activated.

When you save a new slot, you can give it a name that will help you remember the configuration. Click on “New Slot” to create a new slot and click into the text box for editing the name of this slot. The dropdown menu of this box also shows you the available slots which are stored in the controller.

Figure 26: Updating, Loading and Clearing Slots

If you wish to change a particular slot configuration you have saved before, expand the dropdown menu by clicking on the black arrow, and choose the desired configuration. Once you have chosen it, you can test the settings with the “Test settings” button.

If you no longer wish to use a saved configurations, you may delete one or more of them by repeat clickinging of “**Delete slot**”.

Transferring and restoring whole configurations to/from disk

The File Menu allows transferring all current slots from the FABI microcontroller into a settings (.set) file on the computer. The settings file can then be transferred to the same or another FABI device, restoring all slots and settings. Thus, multiple setups (for example for individual users or use-cases) can be kept on a computer and applied with a single click. A file chooser window will open which allows the selection of the desired filename to save or restore the settings.
Attention: When transferring settings from file to FABI, the current slots will be cleared.

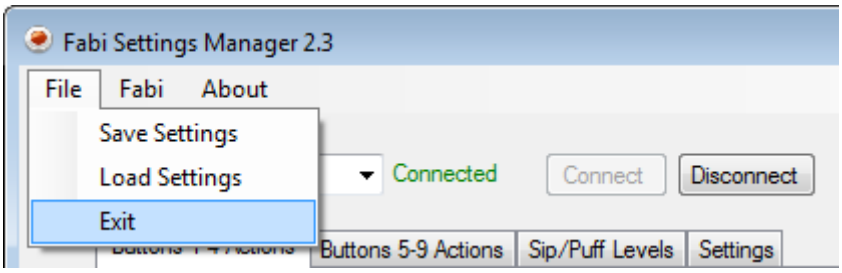


Figure 27: Transferring all slots from FABI to file and vice versa

Assigning functions for buttons or switch actions

In the FABI user interface you can associate up to 9 (in this box, only 8 connectors are used) Button/Switch signals to different computer input control functions like “Click Right Mouse Button”, “Wheel Up” or “Hold Left Mouse Button”. In the following chapter, the selection of these alternative actions is explained.

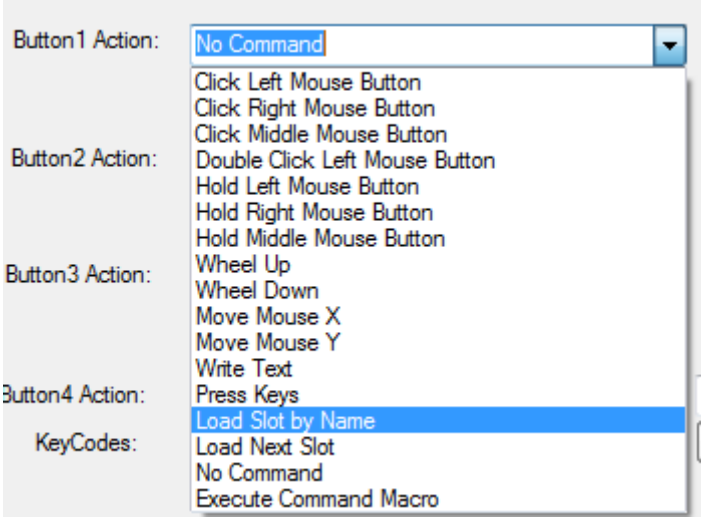


Figure 28: different alternative actions of the FABI GUI

1: No action

If “No action” is selected from the function menu, then no action will be performed when the button is pressed (yes: this also could be desired sometimes ;-))

2: Switch to next configuration

This action is only relevant if you saved multiple FABI setting configurations in the memory slots. Once you have multiple configurations saved, you can assign the action of switching between the configurations saved on different slots in the application.

3: Click left / middle / right mouse buttons

With this function you can simulate a left/middle or right mouse click with the selected button.
Note: a click consits of a press and release of a mouse button, executed both on a button press!

4: Double click left mouse button

Double clicking the left mouse button may be necessary, for example to open a file. However, producing a double click with the regular click mouse button function may not be convenient, so you can assign a double click of the left mouse button instead.

5: Hold left / middle / right mouse buttons

The click mouse button options imitate a quick mouse click, however sometimes it is necessary to continue pressing a particular mouse button (for example, when dragging a file, continuously pressing the left mouse click is necessary). For this purpose, the FABI user interfaces application allows assigning this function to the pressing of a button.
The mouse press is triggered on a button press, the mouse release on a button release.

6: Wheel Up / down

The options “Wheel Up” or “Wheel down” emulate a scroll wheel, otherwise known as the mouse wheel. Triggering the “Wheel Up” option results in upwards scrolling, while “wheel down” results in downward scrolling.

7: Mouse move X or Y

The cursor movements on the computer screen occur in both vertical and horizontal direction, where vertical movements are movements across the X axis and horizontal movements are movements across the Y axis. The “Move mouse X” and “Move mouse Y” emulate computer mouse movements and when triggered they result in mouse movements in the selected axis
These two options also require a speed parameter to indicate how quickly the cursor should move in each case. The input field for the speed parameter appears once the mouse move option is selected.
Note: use a negative value for moving to the opposite direction

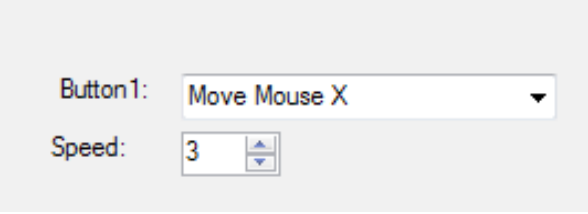


Figure 29: Screenshot of "Move Mouse X" action,
the same principle is valid for Move Mouse Y

8: Write text

The “Write text” option allows you to type a particular text excerpt each time you perform an action (for example, write “Hello” when you press the button).
When you select “Write text”, a blank text box will appear under the dropdown menu as shown below: Click on the text box and type the desired text.

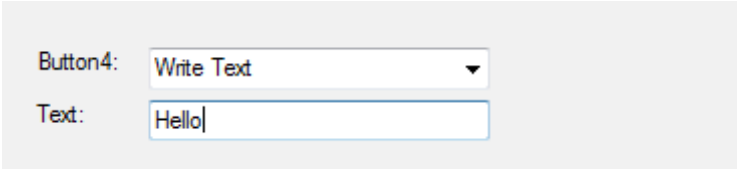


Figure 30: Screenshot of "Write Text" action

In this example, “Hello” will be written each time the button is pressed

9: Press Keys

The “Press Keys” function allows you to perform a selected key command by pressing the button.

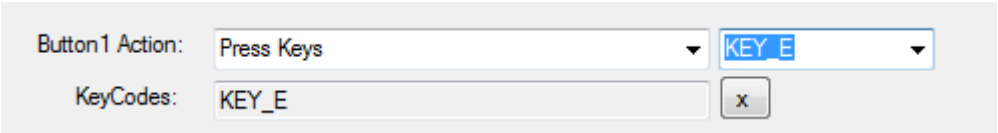


Figure 31: Screenshot of "Press Key" action

In this example, “e” will be written each time the button is pressed.
Note: use “KEY_SHIFT” for upper-case letters. It is also possible to assign multiple keys (remove all assigned keys with the small “x”).

Contact Information

The AsTeRICS Foundation:
Department of Embedded Systems
University of Applied Sciences Technikum Wien
Höchstädtplatz 6
A-1200 Wien
AUSTRIA

Webpage: https://www.asterics-foundation.org
Email: office@asterics-foundation.org

Disclaimer

The University of Applied Sciences Technikum Wien and the AsTeRICS Foundation team do not assume any warranty and liability for the functionality of the set of Assistive Technology and the correctness of the documents handed over.

Additionally, the UAS TW is not liable for any damages to health due to the use of the Assistive Technology provided. The provided software applications and hardware modules are used at own risk!

Acknowledgement

We want to thank Miriam Brenner, Fabian Schiegl and Fanny Peternell for their support in creating this manual.

The AsTeRICS Academy is partly funded by the City of Vienna,
Municipal department of Economic Affairs, Labour and Statistics (MA 23).