FABI

flexible assistive button interface

FABI - Flexible Assistive Button Interface

Construction Manual

AsTeRICS Foundation

Table of Contents

Welcome to FABI

FABI - the "Flexible Assistive Button Interface" - makes it possible to connect several momentary switches (buttons) to a computer or a tablet / smartphone via a USB socket. Pressing a button can trigger desired keys on the keyboard or carry out other mouse cursor actions. A configured FABI module can be used with any computer (Windows, Linux or Mac) without installing special software, because the FABI module behaves like a normal computer mouse or keyboard when it is connected to the computer. People for whom conventional input devices are not suitable can play computer games, surf the Internet, write e-mails and much more.

The FABI interface can be used with arcade buttons, assistive switches or self-made electrical contacts. FABI consists of a hardware module (an inexpensive microcontroller that functions as a computer mouse or keyboard) and a graphical configuration interface ("FABI-GUI") for setting the desired functions.

FABI is available as an open source kit (including the corresponding assembly instructions for the hardware) and was developed as part of the AsTeRICS Academy project at the UAS Technikum Wien. In 2017, the non-profit organization AsTeRICS Foundation was founded in order to further develop such technologies and systems and make them available to the public, see: www.asterics-foundation.org.

All software modules, the hardware design files and the documents for the instructions are available under free and open source licenses and can be used and modified free of charge. We have tried to select the most cost-effective components for the desired functions - which makes FABI the most cost-effective push button interface in the currently known universe!

About this guide

These "Do-It-Yourself" assembly instructions contain information for the assembly and commissioning of the FABI module with 3d-printed housing, and the assembly of buttons with 3d-printed housing. (Instructions for using the configuration software for are available in a separate document[1]).

[1] see https://github.com/asterics/FABI/raw/master/Documentation/FABI_GUI_Manual/de/FABI_AnwendungsAnleitung.pdf

Assembling the hardware

Building your own FABI device is easy!
Here we show you how you can assemble the FABI module and the buttons yourself.

A FABI device consists of a microcontroller, a 3d-printed housing and electrical connections to buttons that can be individually attached for users. A description of the necessary basic knowledge (e.g. soldering basics) as well as information on how buttons or housings can be manufactured yourself can be found here: https://github.com/asterics/FABI/Documentation/ConstructionManual/SelfmadeBox

Materials and tools required

For the FABI Box:

- 1x Arduino Pro Micro microcontroller
- 8x 3.5mm 3-pin jack sockets (PJ-392)
- About 60cm hookup wire
- 1x 3D printed case, the STL files are available here: https://github.com/asterics/FABI/tree/master/Case%20Designs/3D-printer
- 1x M3 screw (6-10mm)

For a button:

- 1x push button (e.g. Arcade button)
- 1x jack cable with 3.5mm stereo or mono jack plug (e.g. this jack cable on reichelt.at)
- 1x 3D printed button housing, the STL files are available here: https://github.com/asterics/FABI/tree/master/Case%20Designs/3D-printer/buttons

Required tools:

- A hot glue gun, hot glue cartridges
- A soldering iron with solder
- A side cutter
- A pincer
- Optional: 3D printer with filament
  The 3D printed components can also be ordered from the AsTeRICS Foundation, or alternative housings could be made if desired.

Assembling the FABI box

1- Place the Arduino Pro Micro microcontroller in the FABI housing so that the USB micro connector points towards the housing wall.

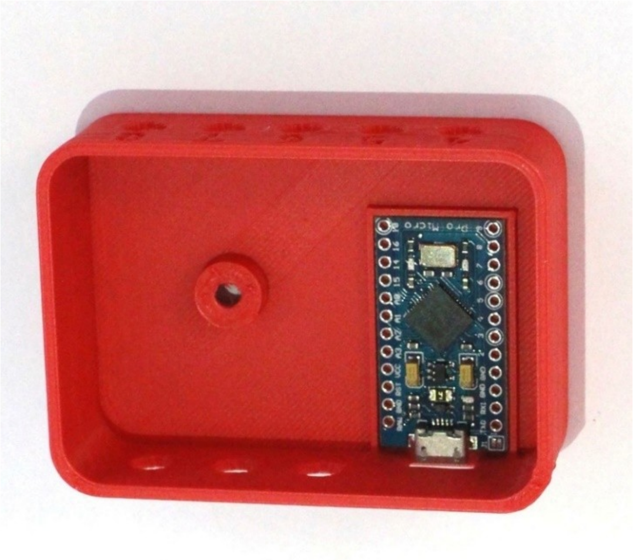


Illustration 1: The Arduino Pro Micro correctly placed in the FABI box

2- Apply hot glue to the side of the Arduino Micro located in the middle of the housing (pins RAW to pin 10) and over the USB micro connector. Make sure that no hot glue gets onto the pins on the edge of the housing (the side with pins TX0 to pin 9) and the inside of the USB port:

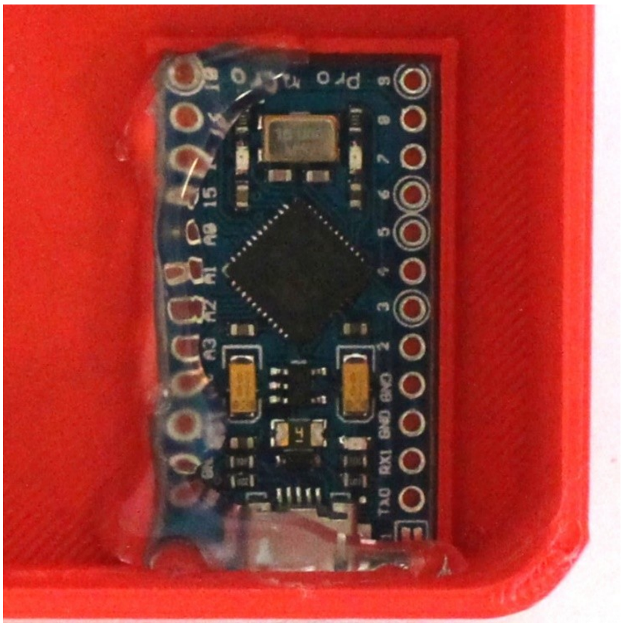


Figure 2: Hot glue was applied to fix the Arduino in the FABI box

3- Attach the first three jack sockets in the holes numbered 1, 2 and 3. (Some 3D printers use support material when printing holes, so the jack sockets may not initially fit into the holes. The easiest way to remove the support material is with a screwdriver (remove the support material by turning it slightly).

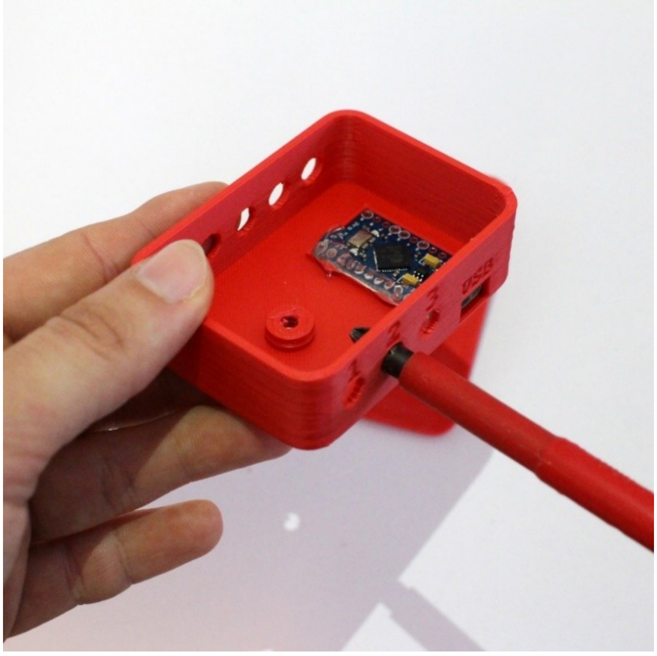


Figure 3: Removing the excess material from the holes for the jack sockets

For the next steps it is important to know that the jack sockets have three pins, these are numbered as follows:

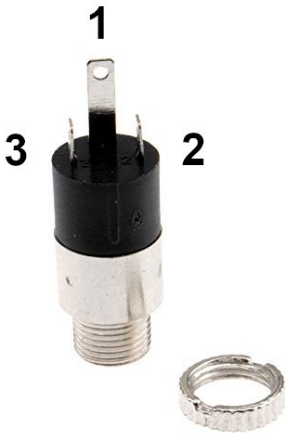


Figure 4: The pins of the jack plug [1]

[1] 3.5 mm jack plug 3-pin stereo panel assembly soldering connection with locking clip nut, 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- Insert the jack sockets into the holes from the inside and then fasten them from the outside with the locking nut. Make sure that the longest pin (Pin1) is at the bottom, as you can see in the picture:

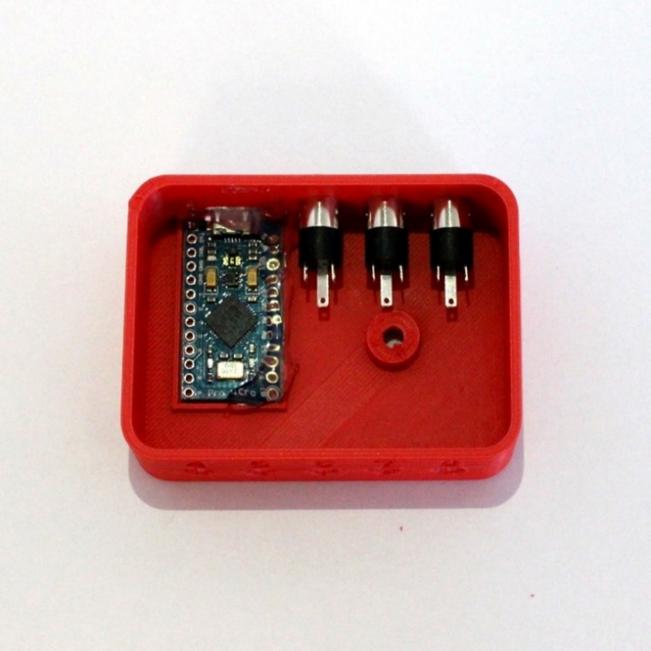


Figure 5: The first three jack sockets, attached in the FABI box

5- Fix the position of the jack sockets with hot glue.

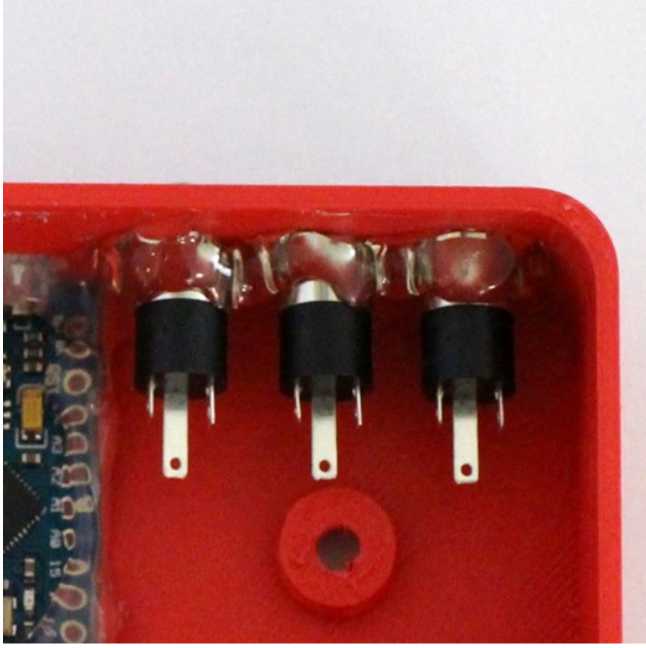


Figure 6: The first three jack sockets fixed with hot glue

6- Cut three pieces of cable about 4.5 cm, 5.5 cm and 6.5 cm in length and remove the insulation from the ends. Solder the longest cable to pin D2 of the microcontroller and the other end to pin2 of the jack socket, which is numbered 1 on the outside of the housing. Solder the next shorter cable to pin D3 and pin 2 of jack socket number 2. Solder the shortest cable to pin D4 and pin 2 of jack socket number 3:

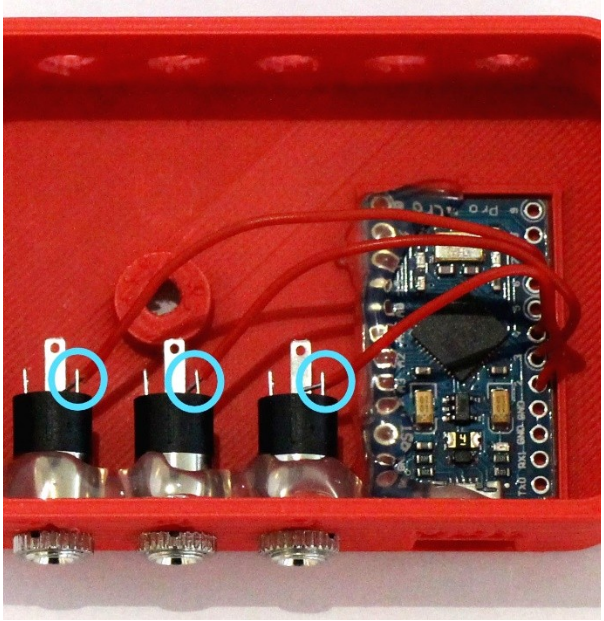


Figure 7: The wires of pins 2-4 of the microcontroller are each connected to pin 2 of the jack sockets

7- Cut five more cables about 4.5 cm, 5.5 cm, 6.5 cm, 7.5 cm and 8.5 cm in length. Solder the shortest of these cables to pin D5 of the microcontroller, the next longer to pin D6, etc.

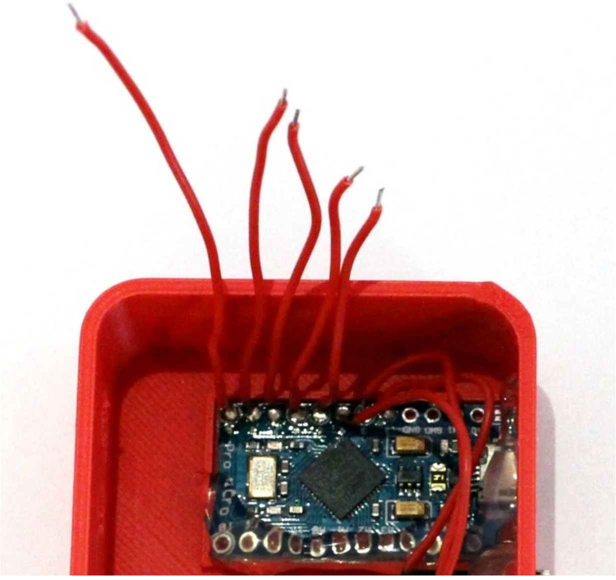


Figure 8: The other wires are connected to pins 5-9 of the microcontroller

8- Attach the five remaining jack sockets to the housing, the longest pin (pin 1) should be below again. Fix the jack sockets with hot glue.

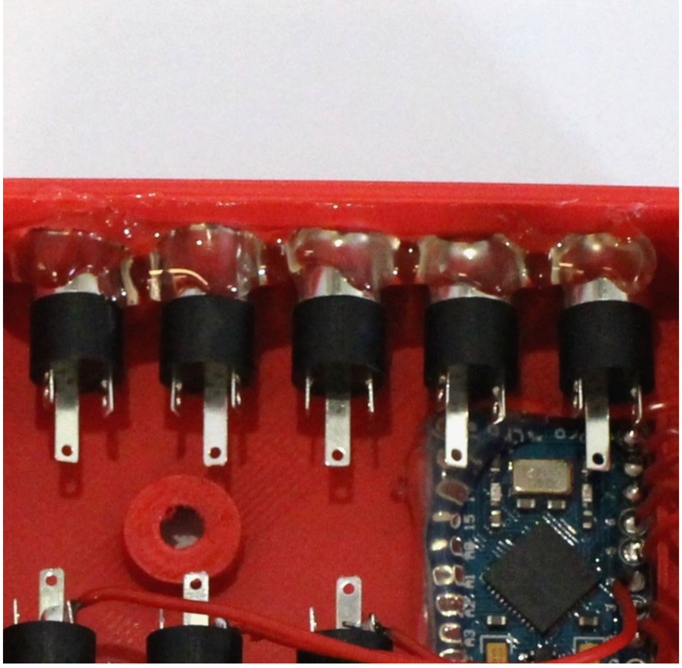


Figure 9: Hot glue fixes the jack sockets 5-8 in the FABI box

9- Solder the wire connected to pin D5 to pin 2 of jack socket number 4. Solder the wire connected to pin D6 to pin 2 of jack socket number 5, etc.

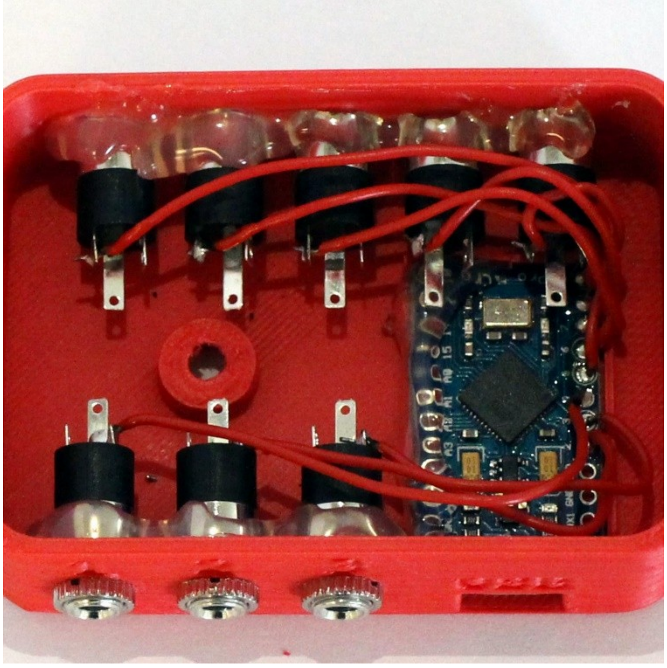


Figure 10: The wires are now connected to the jack sockets

10- Now take an uninsulated wire about 15cm long (or remove all insulation from a piece of jumper wire) and solder one end of the to one of the ground (GND) pins of the microcontroller. Then solder the wire to pin 1 of all jack sockets, starting with jack socket number 3 (as shown in the picture) and cut off the remaining remainder of the wire.

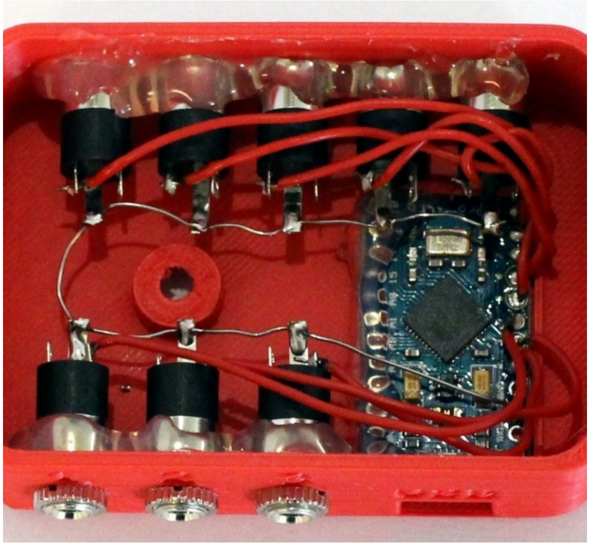


Figure 11: A wire is connected to the GND pin of the Arduino and pin 1 of the jack sockets

Tip: When soldering the wire, it can help to cut pin 1 with a wire cutter, as it is then possible to thread the wire into the hole of the pin.

11- Close the box by fastening the lid with the M3 screw.

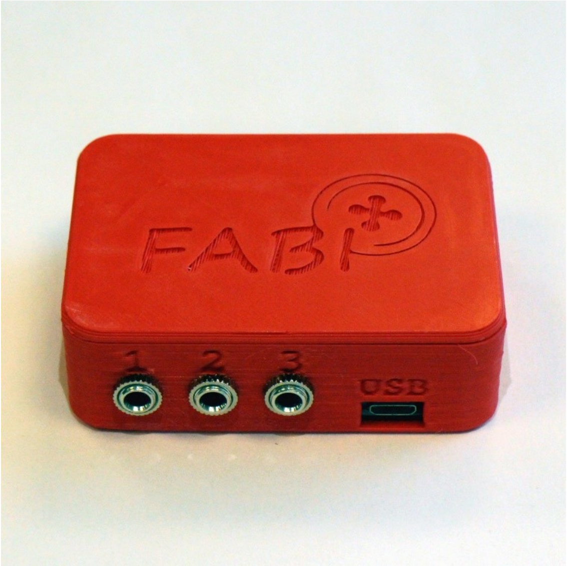


Figure 12: The finished FABI box

Assembling the buttons

1- Remove about 3cm of the outer black insulation of the jack cable.
Strip the insulation from the tips of the yellow and red wires, cut off the white wire, it is not needed for this setup.

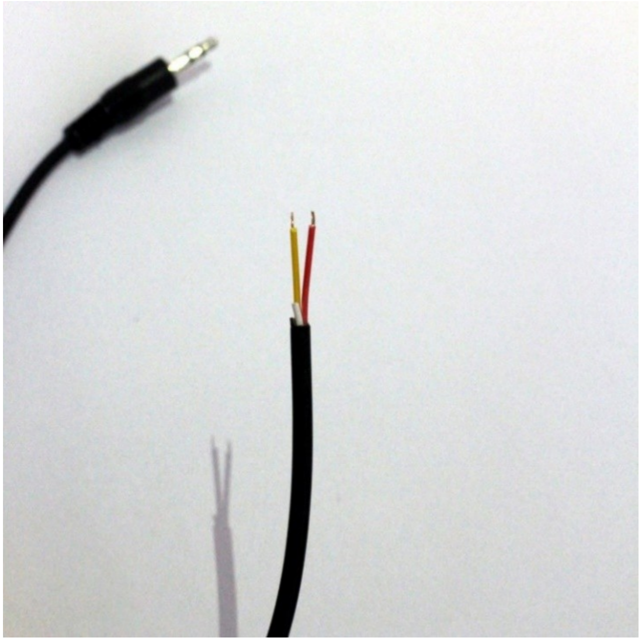


Figure 13: The insulation has been removed from the outside of the jack cable and the tips of the red and yellow cables.

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

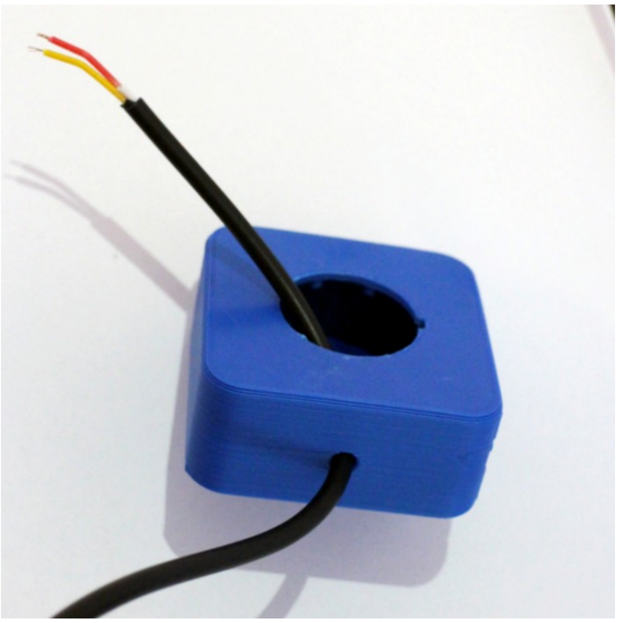


Figure 14: The jack cable was threaded through the hole on the side of the box

3- Make a knot in the jack cable about 5cm away from the end of the cable (the knot ensures that even strong pulling on the cable does not damage the functionality of the button).

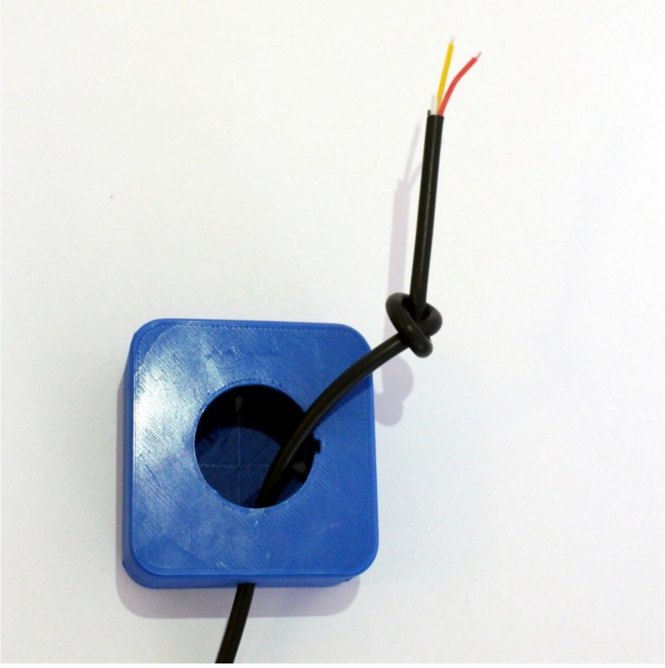


Figure 15: A knot in the jack cable

4- Bend the metal connectors of the button about 30 ° to the side as shown in the picture so that the button fits into the housing.

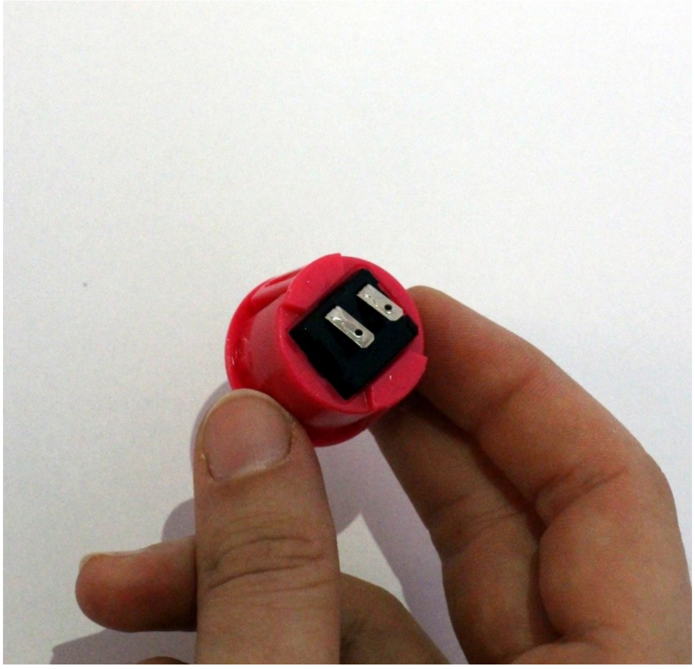


Figure 16: The connectors of the button are bent to the side

5- Solder the red and yellow wires to the metal connectors of the button. (It doesn't matter which cable is soldered to which foot.
   Tip: Putting the tips of the cables through the holes in the button's connectors can make soldering easier.

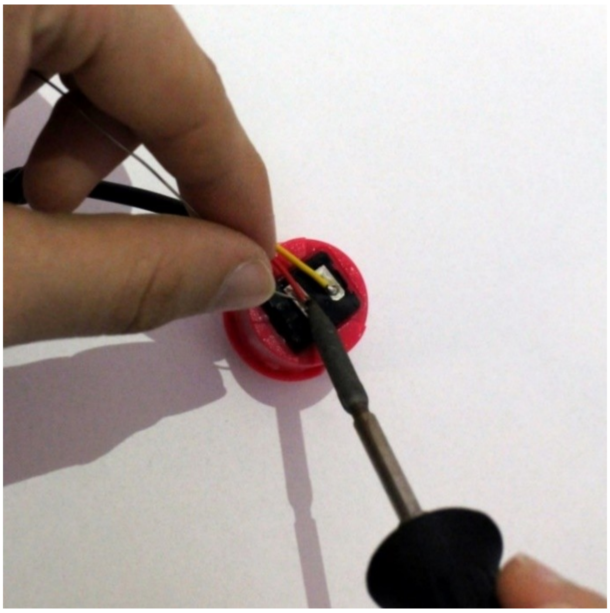


Figure 17: Connect the red and yellow cables to the feet of the button

6- Insert the soldered button into the housing, making sure that the snap hooks on the side of the button point in the direction of the hole on the side of the box. (see picture)

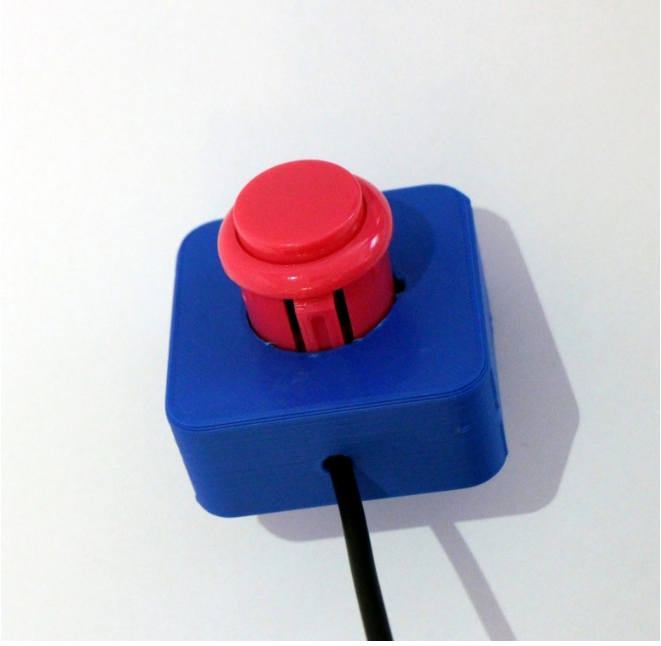


Figure 18: The correct orientation for inserting the button into the housing

Load the firmware onto the microcontroller

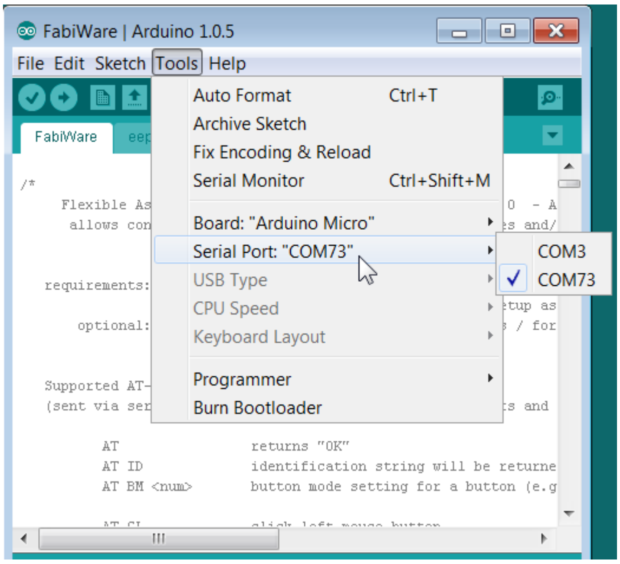
The software that runs on the microcontroller to produce the desired functions of the buttons is called "firmware". It is located in the FABI.zip file and can be downloaded from the following address: https://github.com/asterics/FABI/releases/latest

Now unzip the FABI.zip file in a directory of your choice on your PC. In order to install the firmware on the microcontroller, the Arduino IDE (Integrated Development Environment) is also required. You can download the Arduino IDE here: http://www.arduino.cc.

After installing the Arduino IDE, connect the microcontroller (Arduino Pro Micro) to your computer using a USB cable (the following screenshots show a Windows PC, but the process is similar for Mac and Linux).

Start the Arduino IDE and open the FabiWare.ino file, which is located in the “FabiWare” subfolder of the unzipped software package. Make sure that the device is securely connected to the computer with the USB cable. In the Arduino IDE in the menu "Tools-> Serial Port" select the COM port that is available there after plugging in the USB cable (see Figure 19, the COM port number shown may be different on your PC).

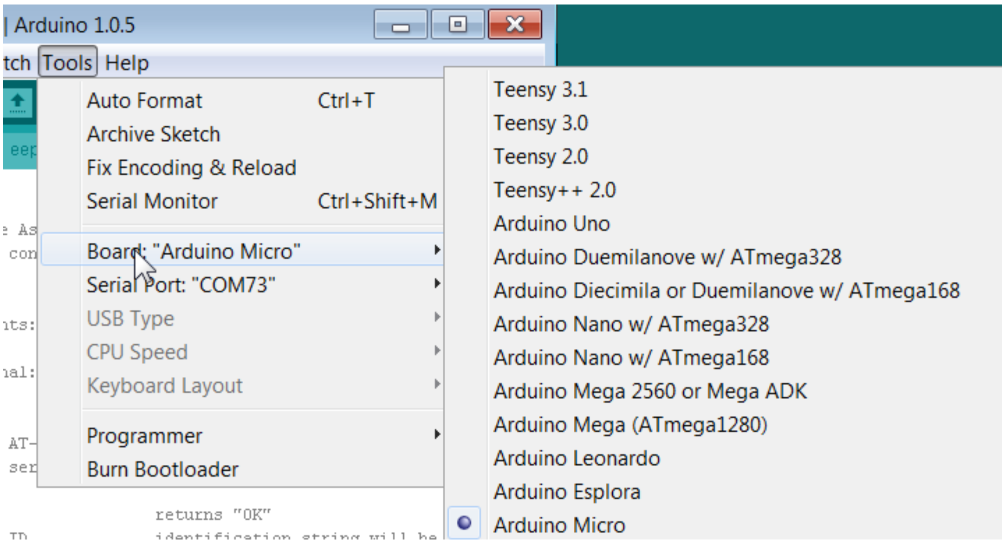
If no COM port is available, follow the instructions in the Arduino IDE manual to correctly install the driver software for the microcontroller: http://arduino.cc/en/Guide/ArduinoLeonardoMicro



```
/*
    Flexible As                                0  - A
     allows con                               es and/

  requirements:
                                              etup as
      optional:                               s / for

  Supported AT-
  (sent via ser                               es and

         AT              returns "OK"
         AT ID           identification string will be returne
         AT BM <num>     button mode setting for a button (e.g
```

Figure 19: COM port selection in the Arduino IDE

After the COM port has been correctly selected, select the entry "Arduino Micro" or "Arduino / Genuino Micro" in the Tools → Board menu:

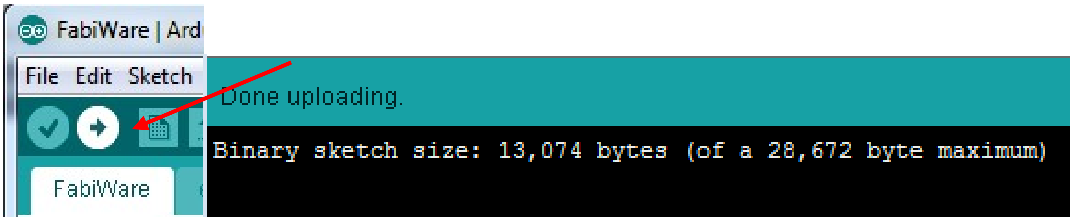


Board: "Arduino Micro"
Serial Port: "COM73"

```
returns "OK"
```

Figure 20: Select the correct type of microcontroller in the board menu of the Arduino IDE

Now you can load the software onto the microcontroller by clicking the "Upload" button in the Arduino IDE. After a few seconds the message “Done uploading” should appear.



```
Done uploading.
Binary sketch size: 13,074 bytes (of a 28,672 byte maximum)
```

Figure 21: Using the upload button of the Arduino IDE to load the firmware onto the microcontroller

The FABI-GUI software can now be used to configure the desired mouse and keyboard actions. You can find detailed instructions on how to use this software in the FABI application manual.

Contact information

AsTeRICS Foundation

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

Disclaimer of liability

The University of Applied Sciences Technikum Wien and the AsTeRICS Foundation assume no guarantee or liability for the functionality of the hardware / software modules or the correctness of the documentation.

Furthermore, the UAS Technikum Wien and the AsTeRICS Foundation are not liable for any damage to health caused by the use of the hardware / software modules provided.
The modules and information provided are used at your own risk!

Acknowledgement

We would like to thank Miriam Brenner, Fabian Schiegl and Fanny Peternell for their support in creating these instructions.

This project was financially supported by the City of Vienna (Municipal Department 23 for Economy, Labor and Statistics, MA 23) (project number 14-02, 18-04).

MA23
Wirtschaft, Arbeit Statistik

StaDt Wien