FH University of Applied Sciences TECHNIKUM WIEN

FLip Mouse

Alternative Computer Input Device

User Manual v2.10

Preface

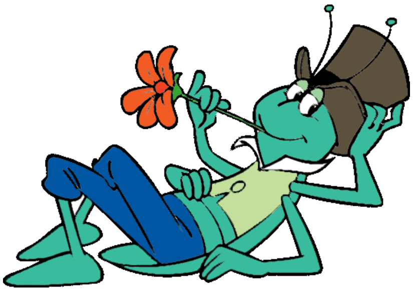
The FLipMouse is an open source Assistive Technology module which is developed at the University of Applied Sciences, Technikum Wien and delivered as a DIY-construction kit by the AsTeRICS Foundation. The FLipMouse allows people who cannot use standard computer input devices to control a computer mouse cursor or a joystick as well as typing desired keyboard keys or using infrared remote controls. The FLipMouse device can be actuated with minimal finger- or lip-movements and/or sip & puff activities via a dedicated mouthpiece.

The FLipMouse Graphical User Interface (GUI) is a software application intended for use in conjunction with the FLipMouse device. This user manual includes a description of the Graphical User Interface for the configuration of the different functions of the FLipMouse – as well as explanations how to use those features. A configured FLipMouse module can be used on any computer without installation of special software, because the FLipMouse module behaves exactly like a standard mouse, joystick or keyboard which is plugged into the computer.

Furthermore, this manual covers the installation of the FLipMouse GUI application and important guidelines for using the FLipMouse, in particular cleaning procedures and hygienic precautions.

All design files for electronics and enclosure - as well as the software source code are available as open source and are distributed via the AsTeRICS Foundation website and via GitHub. The construction kit with detailed instructions for building the FLipMouse can be purchased from Hackerspaceshop.com.

The AsTeRICS Foundation.
http://www.asterics-foundation.org

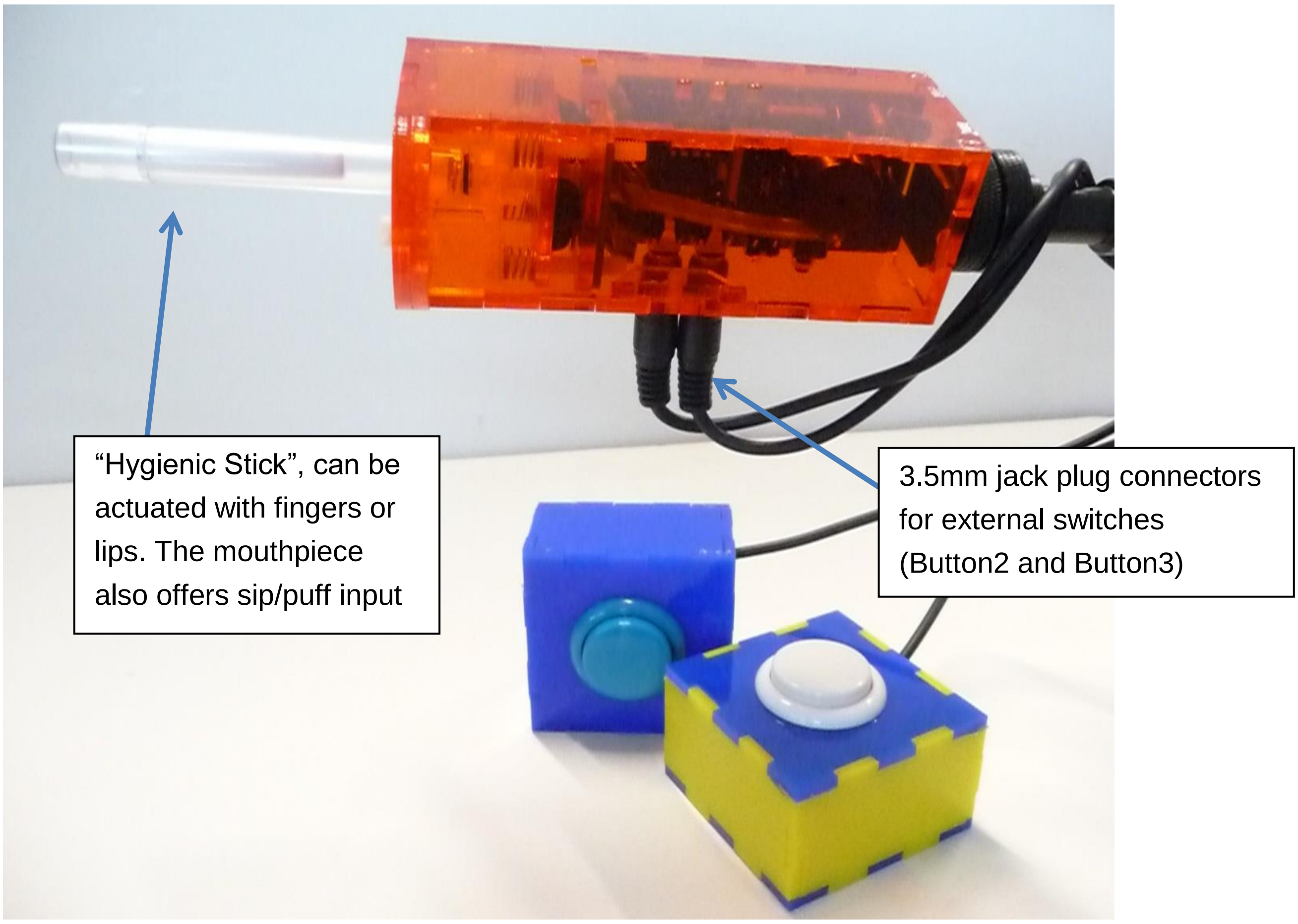


“Flip”, the vagabound grasshopper,
cartoon character © by Waldemar Bonsels

Contents

Hardware overview

The FLipMouse is available with an acrylic glass enclosure and with a 3d-printed enclosure (the DIY-Kit which is sold via Hackerspaceshop.com contains the acrylic case). Design files for the 3d-printed case are available in the Github repository. The primary method for using the FLipMouse is the transparent stick ("Hygienic Stick" or "mouthpiece") which is attached to the front of the FLipMouse. Additionally, two external momentary switches can be attached to extend interaction capabilities:



"Hygienic Stick", can be actuated with fingers or lips. The mouthpiece also offers sip/puff input

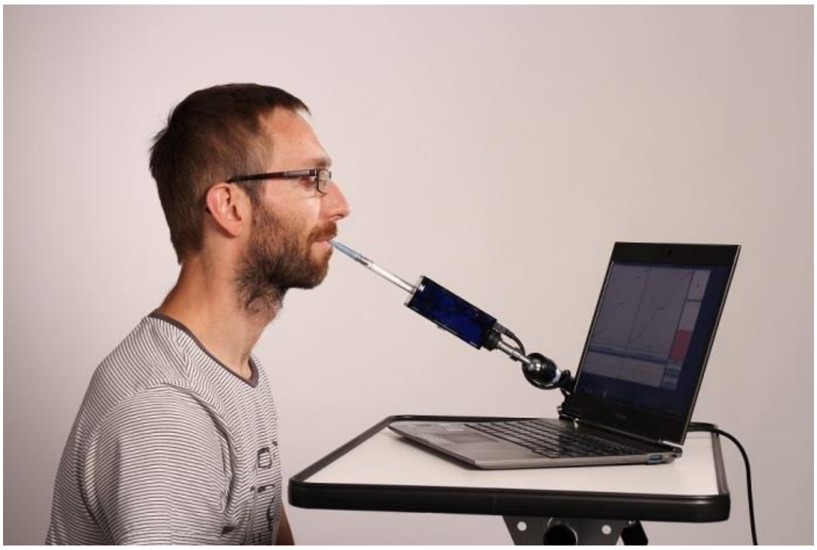
3.5mm jack plug connectors for external switches (Button2 and Button3)

FLipMouse (acrylic case version) with connected external switches. Please note that the mounting screw ("HotShoe adapter") moved to the side since version 2.5 of the FLipMouse.

The *Hygienic Stick* is made by the company *QuadJoy.* Replacement sticks can be ordered via the homepage: http://www.quadjoy.com. We want to thank Bill and Debra Street of *QuadJoy* for letting us use their mouthpiece for our FLipMouse construction kit!
Please note the cleaning instructions in the dedicated section of this manual.

Users can interact with the FLipMouse in several ways:

- by touching the mouthpiece with the lips or fingers
  and applying small forces in vertical or horizontal direction
- by increasing or reducing pressure in the mouthpiece (puffing or sipping)
- by actuating (up to) 3 momentary switches / pushbuttons
- if desired, a longer plastic tube can be attached to the pressure sensor so that the joystick can be used with a finger and a separate mouthpiece allows sip/puff control.

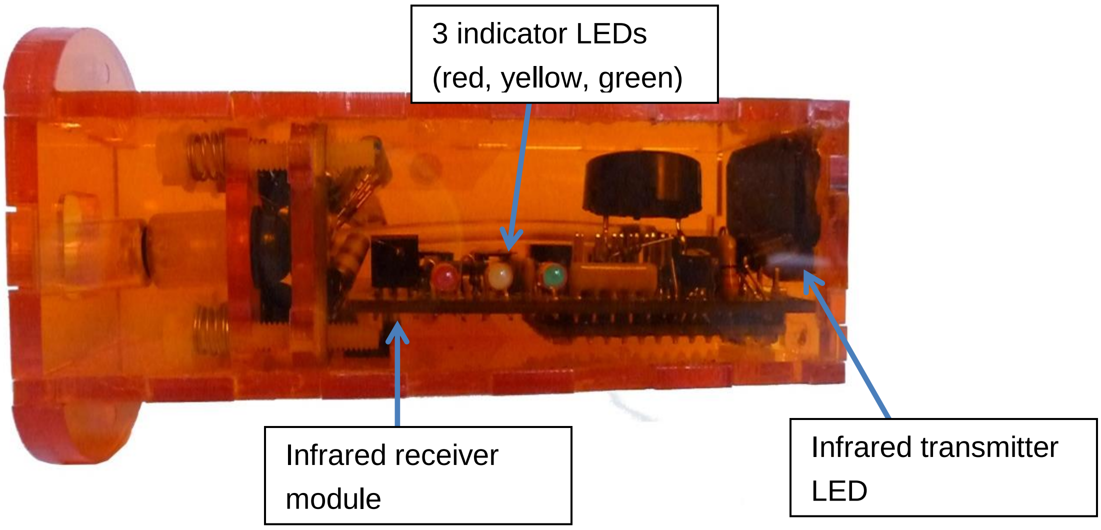


Using the FLipMouse for computer control via lip interaction

The hardware features of the FLipMouse:

- “Zero-way”-joystick / mouthpiece:
  very low forces are sufficient to create up/down/left/right movement
  - The zero-way-joystick can be used with fingers or toes etc.
  - It can also be used as a mouthpiece (actuated by lips / mouth movements)
  - If desired, sip / puff activities into the mouthpiece can trigger additional functions

- One built-in switch, e.g. to change the active configuration (functions)
- Two 3.5mm jack plug sockets for attaching external switches to trigger additional functions
- 3 Indicator Light Emitting Diodes (LEDs) for showing the active configuration, calibration procedure etc.
- Universal infrared remote control receiver and transmitter
- “Hot Shoe” Adapter for mounting to a *Manfrotto Magic Arm* or similar mounting system
- Firmware upgrade via Flipmouse GUI or Arduino (the reset button is needed for this)
- Optional add-on board for Bluetooth (e.g. to control smartphones or iOS devices)

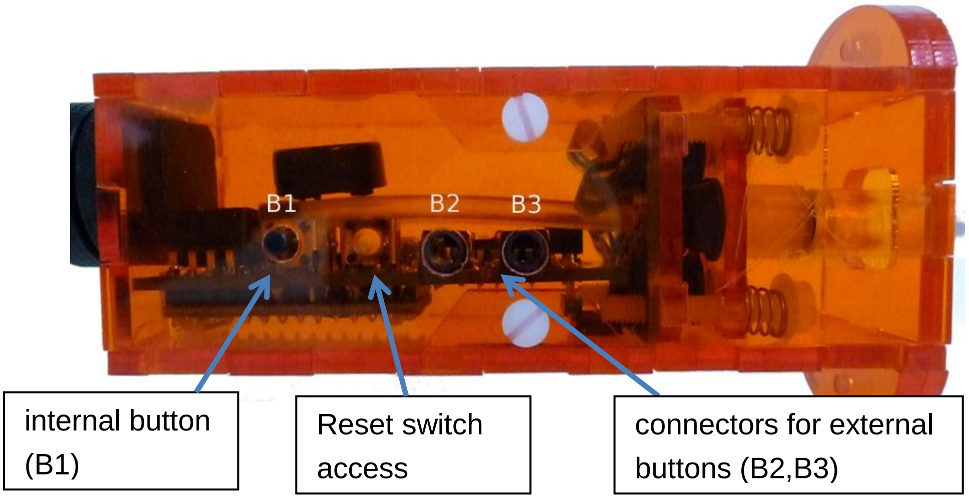


3 indicator LEDs (red, yellow, green)

Infrared receiver module

Infrared transmitter LED

On top of the FLipMouse enclosure, 3 LEDs indicate the current mode of operation (which can be changed to activate different speed settings or functional options). Furthermore, the Infrared (IR-) receiver and transmitter modules are accessible. These modules can be used for recording and replay of arbitrary infrared remote control commands (for example to change volume or channel settings of a TV-set).



internal button (B1)

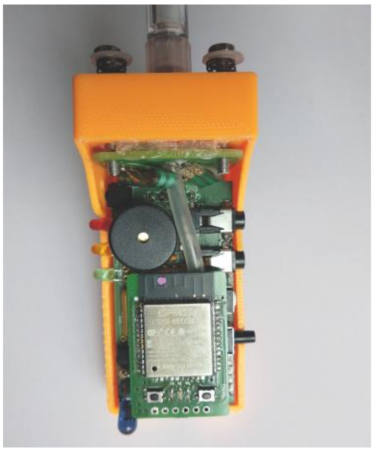
Reset switch access

connectors for external buttons (B2,B3)

The bottom of the FLipMouse gives access to a button (B1) which offers a configurable function (for example changing the operational mode). Next to this button is a hole which enables to press the reset-button of the built-in microcontroller – this is necessary when you want to update the software of the FLipMouse to a newer version.

Two 3,5mm Jack Plugs marked (B1) and (B2) allow the connection of external momentary switches.

3D-Printed Housing and Bluetooth Add-On

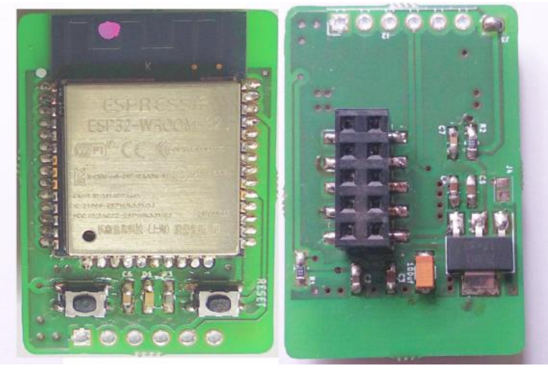


Different alternatives for the FlipMouse housing are available. The design files for the 3d-printed enclosures can be downloaded from the Github repository. The electronic components and PCB are the same as for the acrylic housing variant – and the devices offer exactly the same funcitonality. However, the construction process differs because of the mechanical connections. Thus, two construction manuals for building the FlipMouse are available:

- Construction manual for acrylic case variant and
- Construction manual for 3d-printed variant

An advantage of the 3d-printed case is the better availability of 3d-printers compared to laser cutters so that the 3d-printed case is easier to manufacture. The DIY-Kit which can be ordered from Hackerspaceshop.com contains the laser-cut acrylic parts.

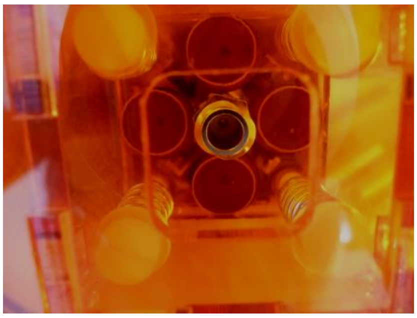
BlueTooth – Add-On Module

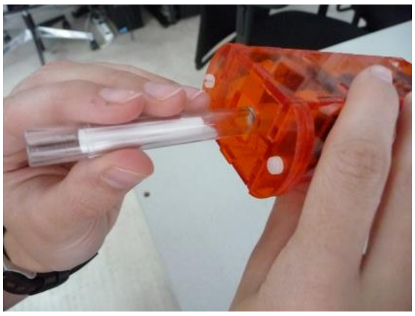


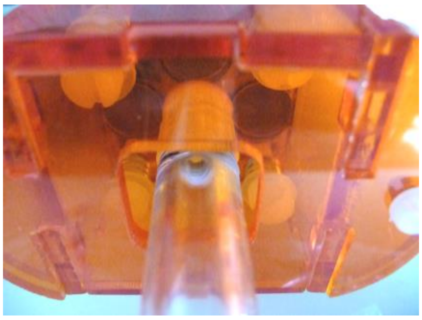
The optional BlueTooth add-on module allows connection of the FLipMouse as a Bluetooth mouse/keyboard to various devices (smartphones, tablets, IOs devices). This module is not contained in the DIY-Kit but can be built from the hardware design files and instructions available on Github (the parts must be ordered individually). If you are interested in this feature, have a look at the FLipMouse Wiki where construction is explained: https://github.com/asterics/FLipMouse/wiki

Instructions for proper mounting and use

1. Attach the "hygienic stick"

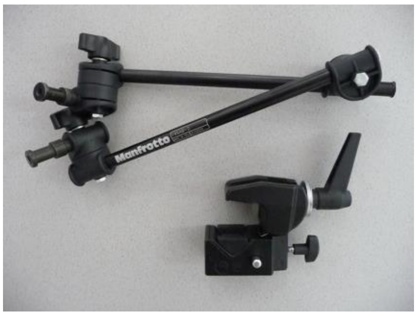


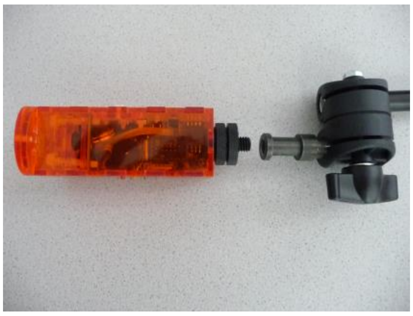


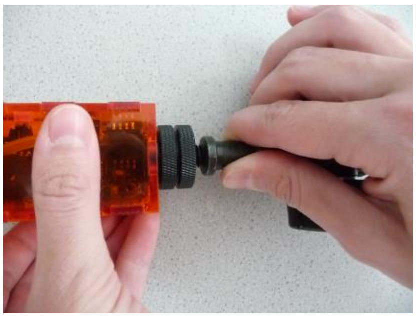


Use a clean "hygienic stick" and attach it to the front connector, turn clockwise (not too hard):

2. Mount the FLipMouse device appropriately for the user

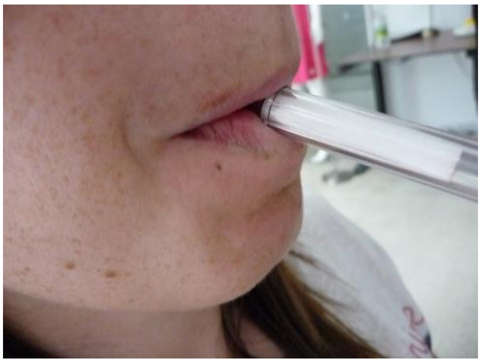






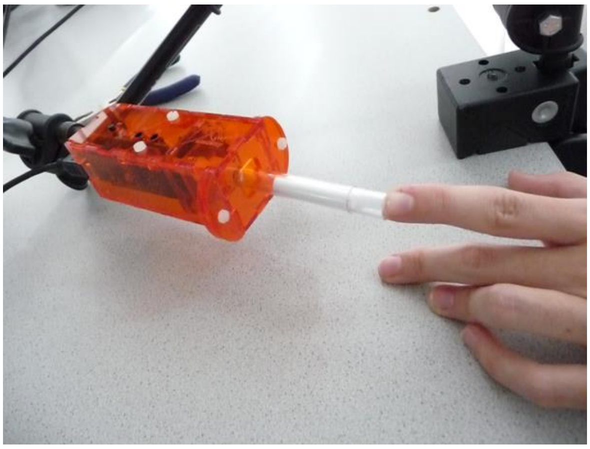
The picture shows a "Manfrotto Gelenkarm" + "SuperClamp" mount combination. You could also use the *Manfrotto Magic Arm* (http://www.manfrotto.com/magic-arm-kit) or any other kind of mounting solution which fits to the *HotShoe*-Adapter of the FLipMouse. Be careful when attaching the mount to the 3/8" screw of the *HotShoe*-Adapter: hard forces might break the acrylic enclosure of the device.

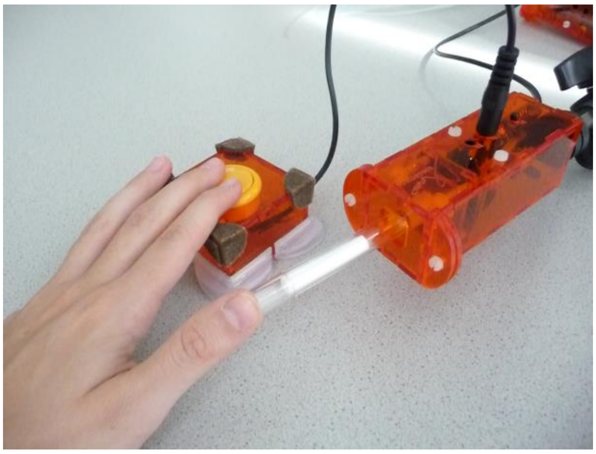
Using the FLipMouse with Mouth/Lips



Ensure that the head of the user rests comfortably. The mouthpiece should barely touch the lips. It is not useful to insert the mouthpiece into the mouth ! (This could result in uncomfortable feelings and make it harder to control the lateral movement)

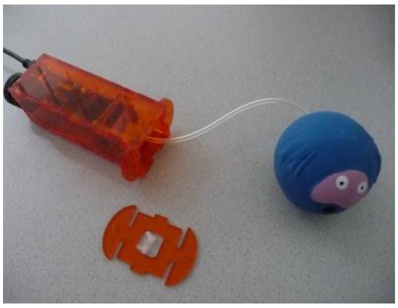
Using the FLipMouse with fingers:

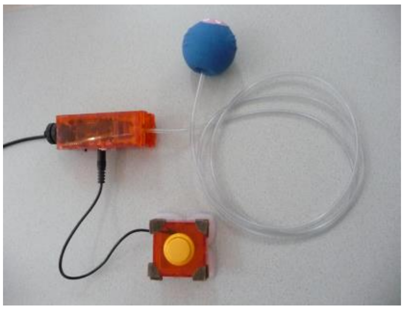


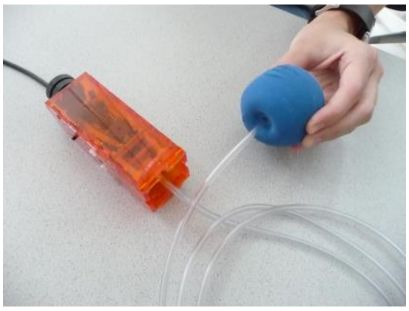


If the user desires to actuate the FLipMouse with a finger or the thumb, mount the system appropriately so that the hand is in a resting position and the finger can touch the joystick without stress. Attach one or two additional external switches if desired, and mount the switches at suitable locations (for example at the legs/toes/shoulder etc.)

Using the FLipMouse as a ball-switch interface:







If a soft button interaction method is desired, a plastic tube can be attached to the front connector of the FLipMouse using a *Luer Lock* adapter. This plastic tube can be connected to a soft ball (the picture above shows a cheap “dog toy” from a pet store). This solution could be used as a soft head-switch or shoulder-switch and can be combined with other external switches. Please note that in this case no direct control of cursor or joystick is possible.

3. Attaching the FLipMouse to a Computer, Tablet or Smart Phone

When you plug in the FLipMouse via the provided *Micro USB* cable, do not touch the mouthpiece as long as the LEDs are blinking (the initial blinking indicates the zero point calibration phase). Wait until the device is recognized by the computer, then move the joystick / mouthpiece to check correct operation.

Connect to a USB port of your computer. If using a smartphone or tablet, attach the *USB-OTG* adapter here

Connect this end of the USB micro cable to the FLipMouse

Please note that every time you apply power to the FLipMouse device (respectively when you plug it in) the zero-calibration is performed which is indicated by blinking all 3 LEDs. It is important that you do not touch the joystick / mouthpiece until the LEDs stopped blinking.

Using the FLipMouse with Android Smart Phones or Tablets

The FLipMouse should work with any operating system which supports USB HID (Mouse/Keyboard/Joystick) devices, such as Windows, Linux or MacOS. Several Android devices provide an USB port with *USB-OTG* (“on-the-go”) functionality. If your device supports “*OTG*” you can attach the FLipMouse using an *USB-OTG*-adapter (see picture below) and it will work as a normal mouse (you will get a mouse cursor) or keyboard. You can test if your Android phone or tablet features the *USB-OTG* function with the “*OTG checker*” app.

Using the FLipMouse via Bluetooth – for example with an iOS device (iPhone or iPad)

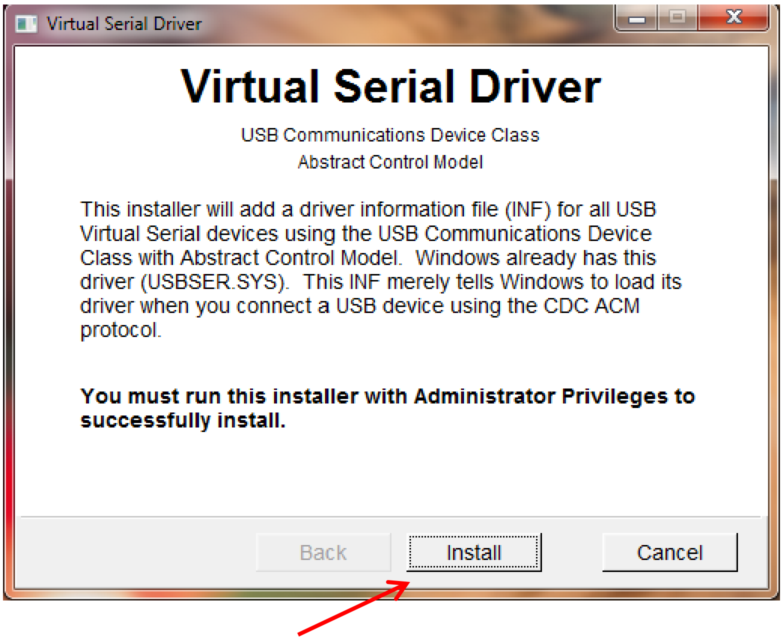
The *Bluetooth* – add-on-module for the FLipMouse allows wireless mouse/keyboard control of computers, tablets and smart phones. Furthermore, iPhones or iPads can be used via VoiceOver & switch support. If attached to PC, users may switch to Bluetooth-operation of a second device and back on demand. For pairing the Blutooth module with a new device, press the internal Button (B1) for 5 seconds until you hear a long tone. If a passphrase needs to be entered on the host device, use “1234”. Use the FlipMouse Settings manager to activate Bluetooth operation
(see section General Settings).

Installing the Settings Manager (GUI)

The *FLipMouse Settings Manager* application offers a Graphical User Interface (GUI) to change and store settings of the FLipMouse device so that all features and functions can be adapted to personal preferences and needs. The *FLipMouse Settings Manager* sends and receives information from/to the FLipMouse. This information transfer occurs through a communication port (COM port). The COM port driver must be installed before you can use the *FLipMouse Settings Manager*. Please note that the *FLipMouse Settings Manager* is currently only available for Windows and the following instructions apply the Windows operating system only:

To install the COM port driver, start the *serial_install.exe* program which is contained in the software package for the FLipMouse, which can be downloaded from:
https://github.com/asterics/FLipMouse/releases,
Alternatively, the COM port driver can be downloaded from
https://www.pjrc.com/teensy/serial_install.exe

When starting the *serial_install.exe* program, please make sure you have Administrator Privileges, and click the install button. (If your current user account does not provide Administrator Privileges, right-click the *serial_install.exe* file and select "Run as Administrator". The following window will appear:



Virtual Serial Driver

Virtual Serial Driver

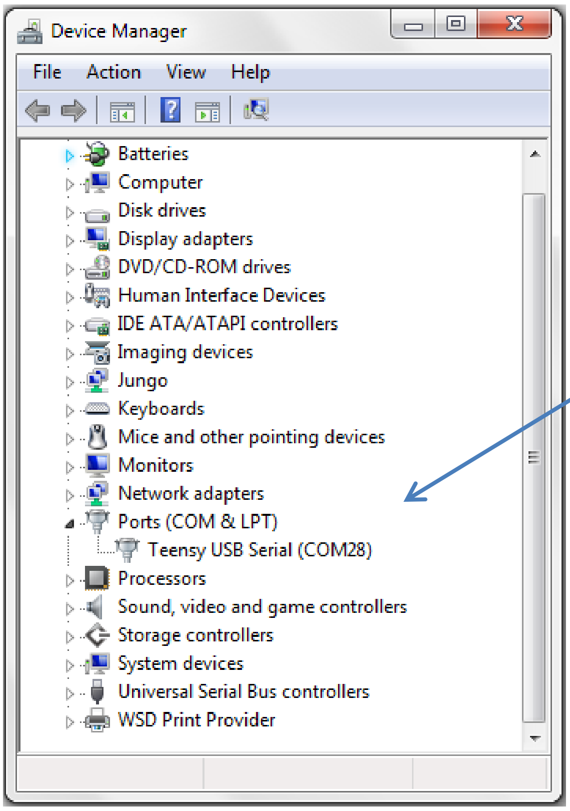
USB Communications Device Class
Abstract Control Model

This installer will add a driver information file (INF) for all USB Virtual Serial devices using the USB Communications Device Class with Abstract Control Model. Windows already has this driver (USBSER.SYS). This INF merely tells Windows to load its driver when you connect a USB device using the CDC ACM protocol.

You must run this installer with Administrator Privileges to successfully install.

Click "Install" to setup the COM-Port driver for communication between FLipMouse and host computer. After finishing installation, please (re-)connect your FLipMouse device using the micro USB cable.

After making sure that the device is securely connected to the computer, you may check if the device's COM port is successfully detected. The detection might take a couple of seconds.
To check if the COM port is detected correctly, open the *Device Manager* of the Windows operating system:



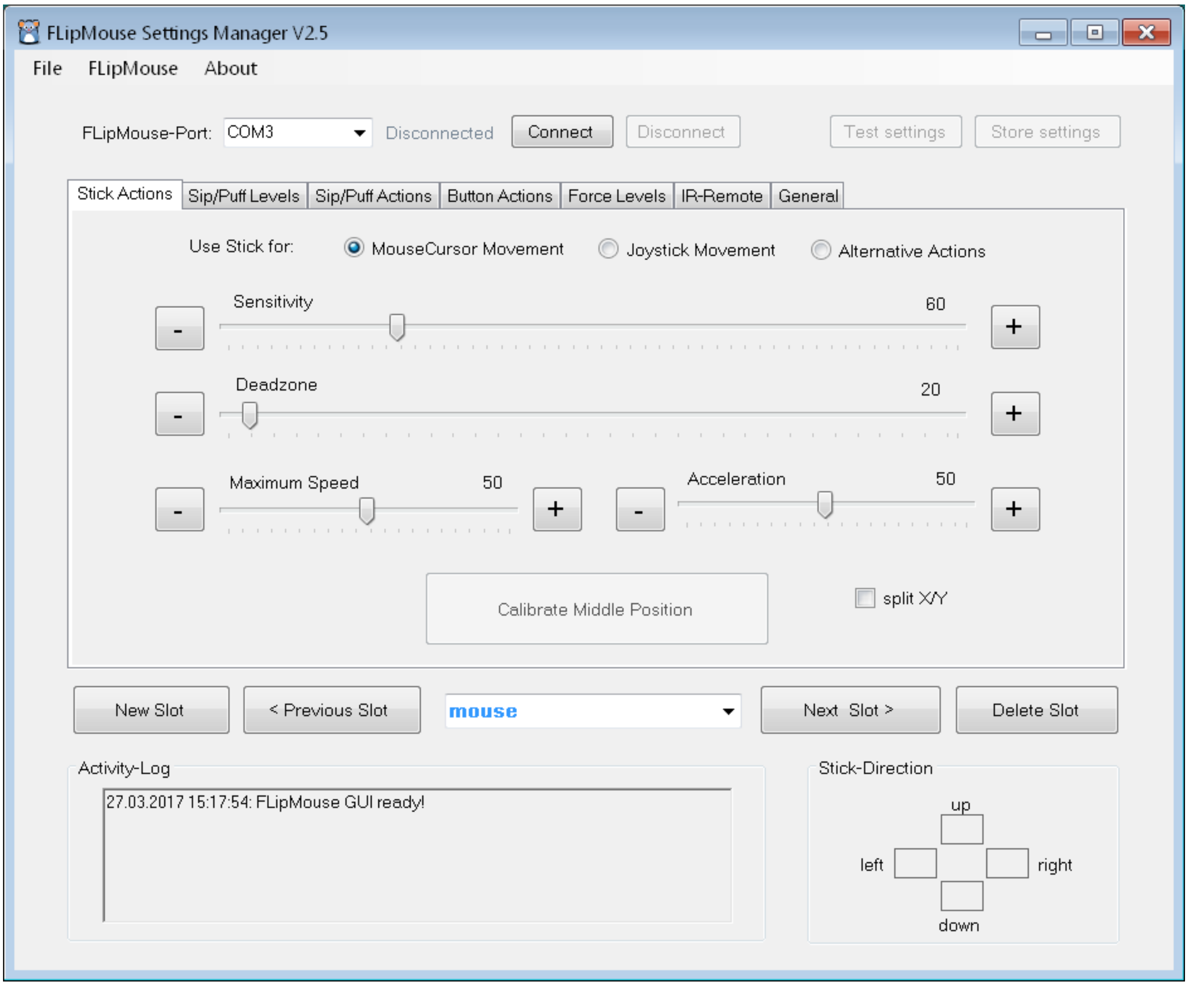
- Batteries
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Imaging devices
- Jungo
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - Teensy USB Serial (COM28)
- Processors
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
- WSD Print Provider

The COM port of your FLipMouse should appear here. The port name is in this example "COM28", however the number might differ because it is automatically suggested by your computer.

If you see a COM port as in the example above – congratulations! You are now ready to use the FLipMouse GUI. If you want to update the FLipMouse software (*Teensy* microcontroller firmware) – please refer to chapter "Firmware Update"!

The FLipMouse Settings Manager (GUI)

To use the *FLipMouse Settings Manager*, double-click the *FLipMouseGUI.exe* file.
The window of Graphical User Interface (GUI) will appear:



FLipMouse-Port: COM3 Disconnected

Use Stick for: ● MouseCursor Movement ○ Joystick Movement ○ Alternative Actions

Sensitivity 60

Deadzone 20

Maximum Speed 50

Acceleration 50

□ split X/Y

mouse

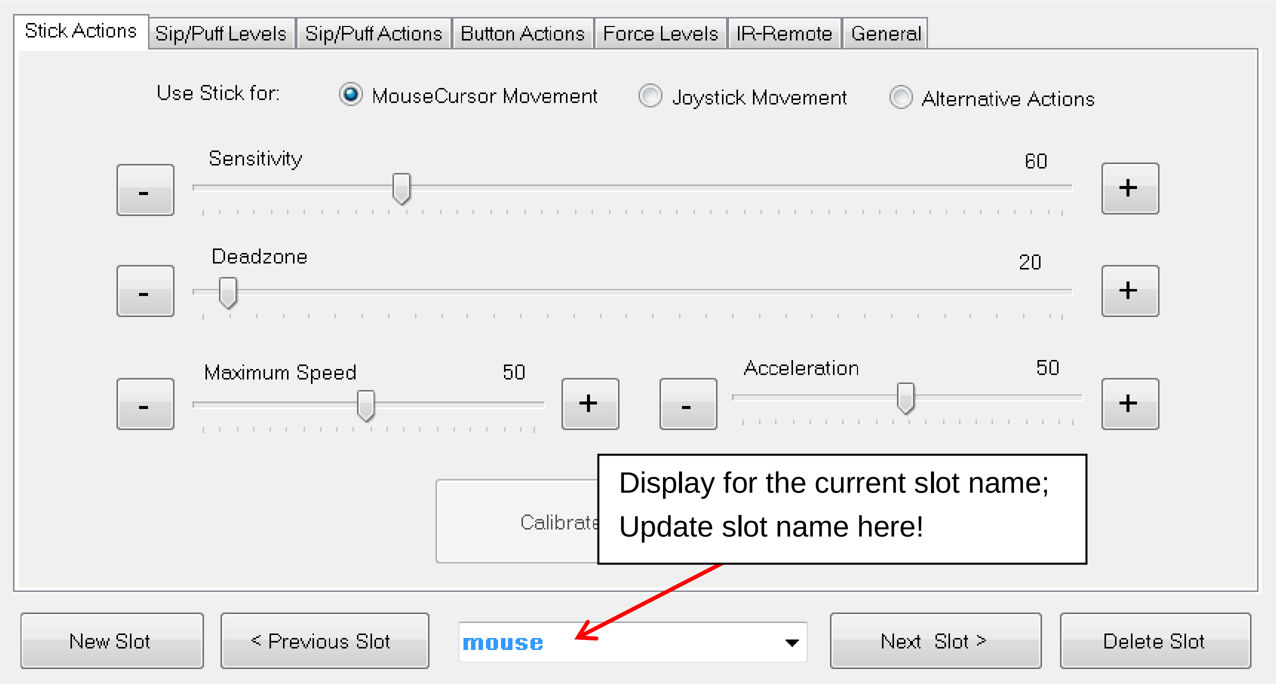
27.03.2017 15:17:54: FLipMouse GUI ready!

Operational modes – “memory slots”

The *FLipMouse Settings Manager* allows adjustment of all important settings (operational modes). The settings are stored in individual ***memory slots*** (for example one slot for fast mouse operation, one slot for slow mouse operation, one slot for keyboard key generation etc.). All settings can be stored to (or loaded) from the FLipMouse after it has been connected to the USB port of the computer. The settings can also be saved to (or loaded from) a file on your computer.
Settings which were stored into the FLipMouse memory slots stay valid also when the power supply / USB cable is removed. When the FLipMouse is plugged in the next time, the settings will be available – also if you use another computer or operating system!

Managing Slots

The Slot management bar consists of the Buttons “*New Slot*”, “*Previous Slot*”, “*Next Slot*” and “*Delete Slot*”. After starting the *FLipMouse Settings Manager*, only one default slot named “mouse” exists. When you create a slot using the “*New Slot*” button, you create a new slot **by copying the settings of the current slot**. You can assign a name that will help you remember the slot’s purpose by clicking into the drop-down box in the middle of the slot management bar:



Use Stick for: ● MouseCursor Movement ○ Joystick Movement ○ Alternative Actions

Sensitivity 60

Deadzone 20

Maximum Speed 50

Acceleration 50

mouse

Display for the current slot name;
Update slot name here!

Clicking will highlight the default text, as seen above. You may now the type the new name, as desired. Using the “*Previous Slot*” and “*Next Slot*” Buttons you can change between the existing slots. The settings of the current slot will be stored and remembered when changing to another slot. The maximum slot count is five. The active slot can be removed using the “*Delete Slot*” button.

Changing the settings of the active slot

The functional features and settings of the active slot can be changed by using the sliders and selection boxes displayed on the different tabs of the *FLipmouse Settings Manager* window. These settings tabs will be described in detail in the next chapter.

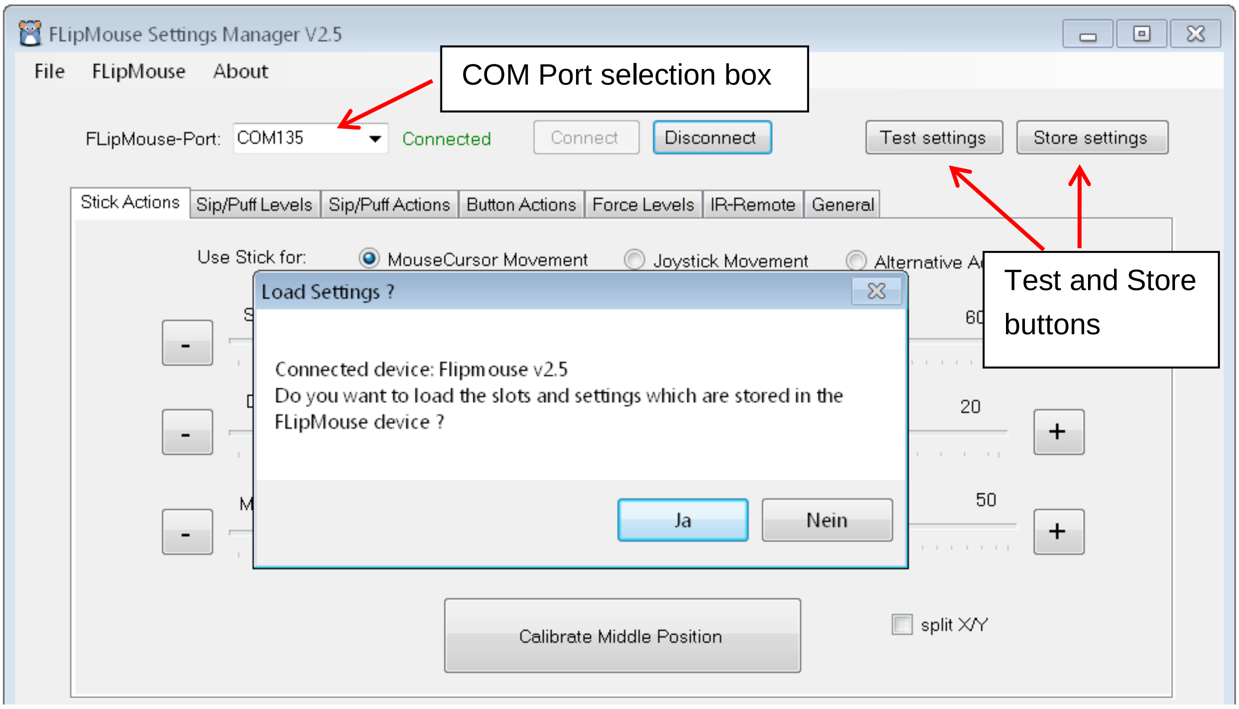
Loading and saving the configuration to/from file

The *File* Menu allows transferring all current slots from to a settings (.set) file on your computer. All settings can thus be transferred to the same or to another FLipMouse device. 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 be opened which allows selection of the desired filename for saving or loading the configuration.

Connecting the FLipMouse Device

In order to be able to use the configuration settings on the FLipMouse device, the FLipMouse must be connected to the application. To connect the device, follow these steps:

1. Make sure your device is connected to your computer via an USB port (see page 12).
2. Select the appropriate COM port (communication port) in the combo box at the top of the application window. If the combo box appears empty, it means that no port has been detected. In this case, please reconnect the device and wait for the COM ports to be updated, and then click on the drop-down menu to refresh the COM port list.
3. Once the COM port is selected, click the "*Connect*" button on the right hand side of the combo box. When the device is connected, a window will pop up, asking if you want to load the current settings from the device. If you confirm this question with "yes", the settings displayed in the GUI window will be overwritten with the settings that are currently stored in your FLipMouse. If you choose "No", the settings will be kept and could be transferred to the FLipMouse. Since Version 2.6, the older firmware versions of the FlipMouse device are detected and can automatically be updated. If this process fails see section "Firmware / GUI update", page 32.
4. After successful connection of the COM port, the port status at the top right corner of the window will change to "connected":



COM Port selection box

Test and Store buttons

Load Settings ?

Connected device: Flipmouse v2.5
Do you want to load the slots and settings which are stored in the FLipMouse device ?

Now you should see live feedback of stick movement and sensor values on the screen.

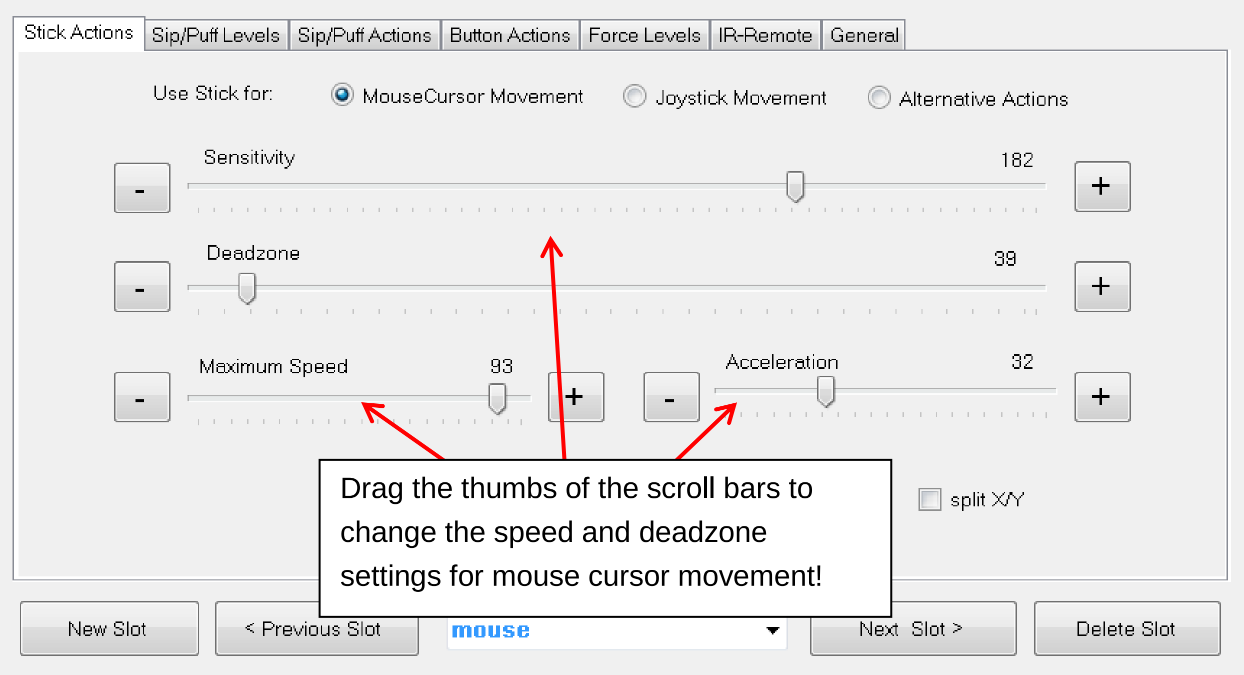
Testing and Storing current Settings

When the FLipMouse device is connected, the settings of the active slot can be applied temporarily to the device by clicking "*Test 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 FLipMouse with the new settings. To store all current settings (all slots) permanently to the FLipMouse device, press the "*Store settings*" button. Storing all slots can take several seconds.

Changing Settings and Features

Stick Actions

The *StickActions* tab is the first panel of the *FLipMouse Settings Manager* user interface. On this panel, you will be able to change the *Sensitivity, Deadzone, MaximumSpeed* and *Acceleration* of the FLipMouse stick. You may also *Calibrate* the middle position of the stick if necessary.
By default, the FLipMouse stick is created mouse cursor movements. However, stick movements can also be reassigned to other functions (such as pressing the key 'A' when the stick is pushed up) by selecting the "*Use Stick for Alternative Actions"* - which will be described later.
To continue using the FLipMouse stick for cursor movements, make sure "*Use Stick for Cursor Movement*" is selected, as in the image below. The visible scroll bar allows changing the movement behavior of the mouse cursor according to the preferences of the user.



Use Stick for: ● MouseCursor Movement ○ Joystick Movement ○ Alternative Actions

Sensitivity 182

Deadzone 39

Maximum Speed 93

Acceleration 32

□ split X/Y

Drag the thumbs of the scroll bars to change the speed and deadzone settings for mouse cursor movement!

Sensitivity

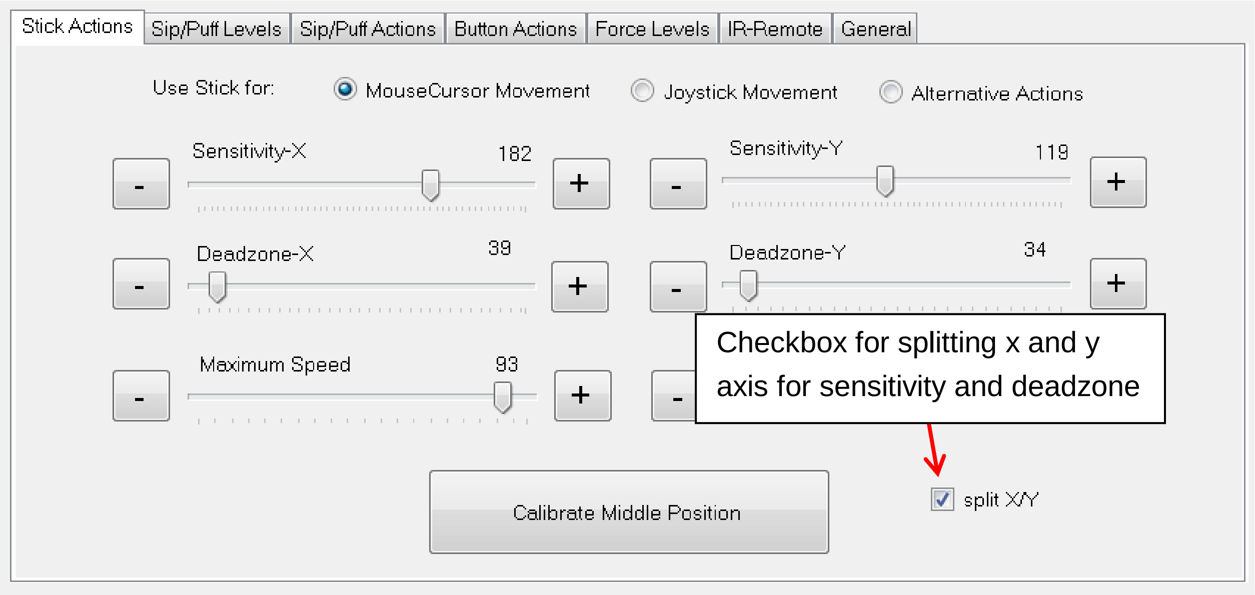
When using the stick for cursor movement, the sensitivity of the stick can be adjusted via the *Sensitivity* scroll bar. A smaller value results in lower sensitivity to applied force. To change the value, click and drag the scroll bar knob or click the bar next to the knob. You can also place the mouse cursor over the "+" or "-" buttons next to each scroll bar so the scroll bar position will decrease or increase automatically (This works for all scroll bars of the *FLipMouse Settings Manager*).

Deadzone

The *deadzone* setting defines a passive zone for the stick movements: when the *deadzone* value is low, very slight stick movements will already cause cursor movement (or execute the assigned alternative function in case "*Use Stick for Alternative Actions*" is selected – see following chapter). If the deadzone setting is too low, the cursor starts unintended drifting especially if a stronger force has been applied. In this case, increase the *deadzone* value so that cursor drifting does not occur under normal operating conditions for a given user.

Split axis control for sensitivity and deadzone:

Moving the cursor on the computer is possible both horizontally (left or right movements, also known as x axis) or vertically (up or down movements, also known as y axis). If desired, the *sensitivity*- and *deadzone*-values for horizontal or vertical movements can be changed individually. In order to do that, select "*split X/Y*", as shown below:



Use Stick for: ● MouseCursor Movement ○ Joystick Movement ○ Alternative Actions

Sensitivity-X 182

Sensitivity-Y 119

Deadzone-X 39

Deadzone-Y 34

Maximum Speed 93

Checkbox for splitting x and y axis for sensitivity and deadzone

☑ split X/Y

If you decide that splitting the horizontal and vertical movements is not necessary, simply unselect "*split X/Y*", and the *sensitivity* and *deadzone* scroll bars will go back to the default state.

Maximum Speed and Acceleration

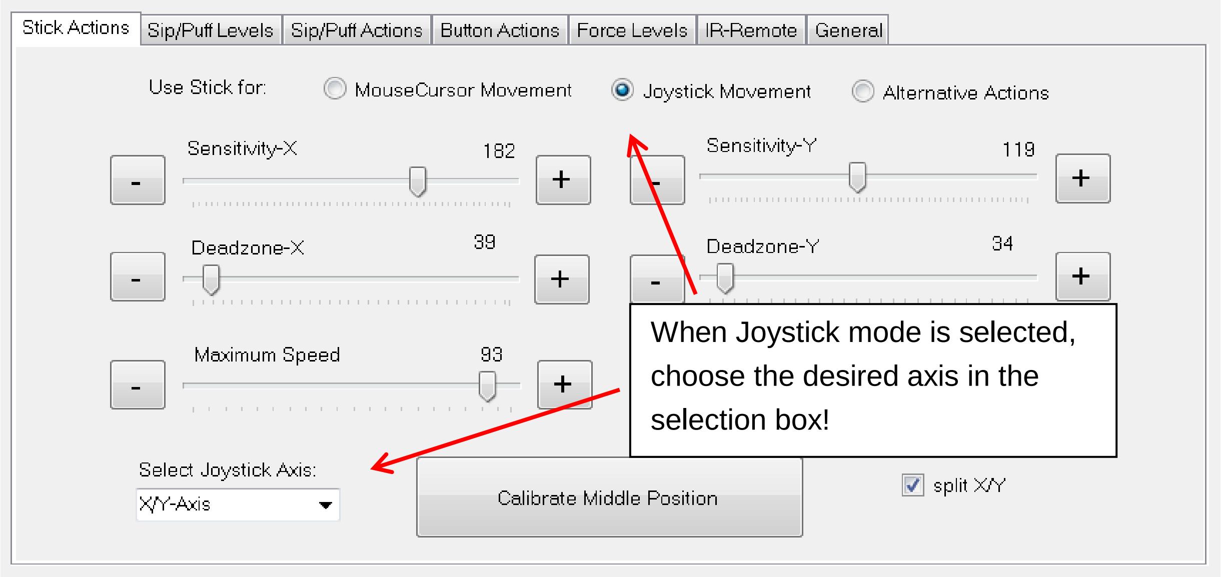
Turning the *maximum speed* setting to a low level is useful if a user desires to limit the cursor speed. The acceleration setting allows influencing the acceleration behavior of the mouse cursor: if acceleration is set to a low level, even a strong deflection of the stick will start a slow (but progressing) cursor movement which makes it easier to reach near target precisely.

Calibration of the stick’s middle position

When using the stick for cursor movement, the cursor should stay in place when the stick is not touched. Sometimes the resting position of the stick may becomes inaccurate and causes unwanted cursor movements, even if the stick is not used. If this happens, you can either increase the *deadzone* values or initiate a calibration. When the Settings Manager is active, a calibration can be initiated by pressing the “*Calibrate Middle Position*” button. Once you click this button, do not move the FLipMouse stick, otherwise the calibration will be done incorrectly. The LEDs on the device will flash and the speaker will emit a low tone, indicating the start of the calibration. This will last for about 1 second - then the calibration sequence is accomplished. Calibration can also be assigned as an action in response to button pressing, sip/puff actions, or alternative FLipMouse stick actions, so that it can be performed on demand, while using the FLipMouse device.

Using the Stick for Joystick Movement

If you select “*Use stick for Joystick Movement*”, moving the FLipMouse stick up / down / left / right will result in joystick activities. The FLipMouse will behave like a gamepad with 6 axes (*X/Y, Z/Z-Turn* and *Slider1/Slider2*). As the FLipMouse stick can only deliver 2 axes of information, the desired joystick axis can be chosen from a selection box which appears left beside the calibration button:



Use Stick for: ○ MouseCursor Movement ● Joystick Movement ○ Alternative Actions

Sensitivity-X 182

Sensitivity-Y 119

Deadzone-X 39

Deadzone-Y 34

Maximum Speed 93

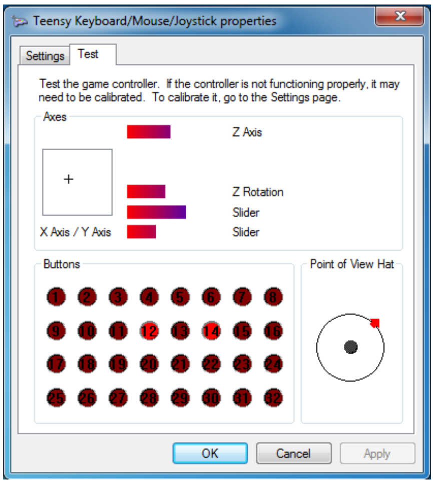
Select Joystick Axis: X/Y-Axis

☑ split X/Y

When Joystick mode is selected, choose the desired axis in the selection box!

Please note that the joystick function must be supported by software applications running on the computer (for example computer games). Microsoft Windows offers a test software called “*joy.cpl*” – you can start this utility by entering “*joy.cpl*” into the search dialog.

The joystick controller provided by the FLipMouse is called “*Serial+Keyboard+Mouse+Joystick*”. When you select this device in the “*joy.cpl*” utility and display its preferences, the live values of the joystick axis and button activities will be displayed in a window similar to the one shown here:

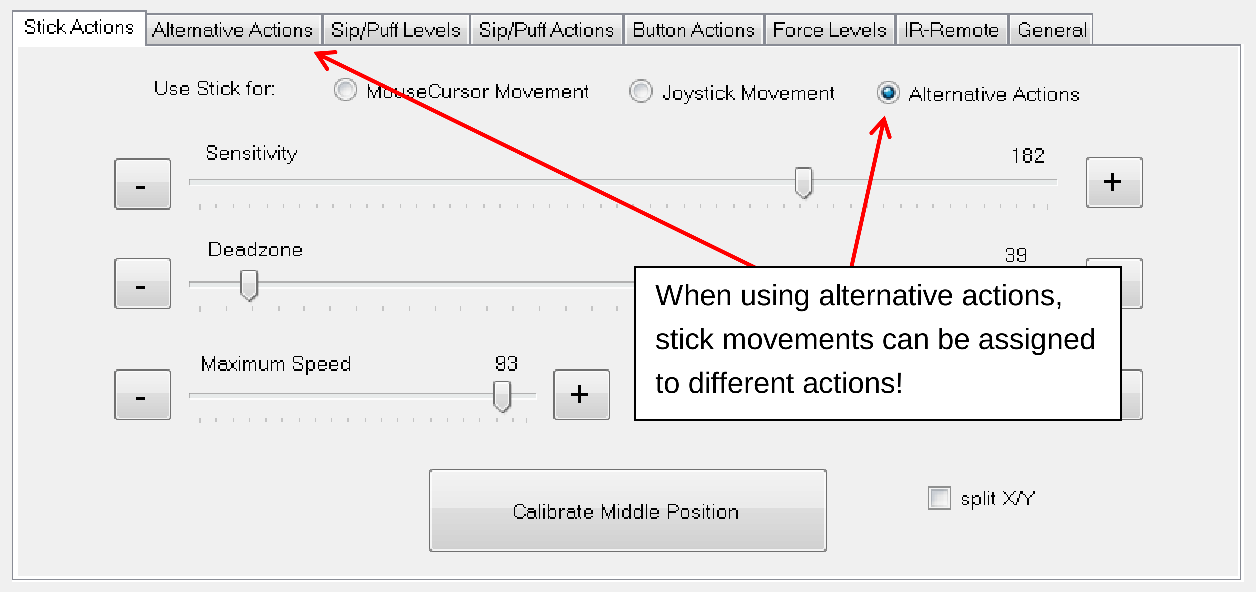


Test the game controller. If the controller is not functioning properly, it may need to be calibrated. To calibrate it, go to the Settings page.

Desired joystick button activities can be created using sip/puff or other interaction events by choosing “*Press Joystick Button*” from the action menu as described in the following section.

Using the Stick for Alternative Actions

The “*Alternative Actions*” tab appears when you select “*Use Stick for Alternative Actions*”. It allows the assignment of specific actions to FLipMouse stick activities (moving the stick up / down / left / right). For using these alternative actions instead of mouse cursor movement, change the slot (e.g. from “*mouse*” to “*keys*” or create a new slot and select the tab “Alternative actions.



Use Stick for: ○ MouseCursor Movement ○ Joystick Movement ● Alternative Actions

Sensitivity 182

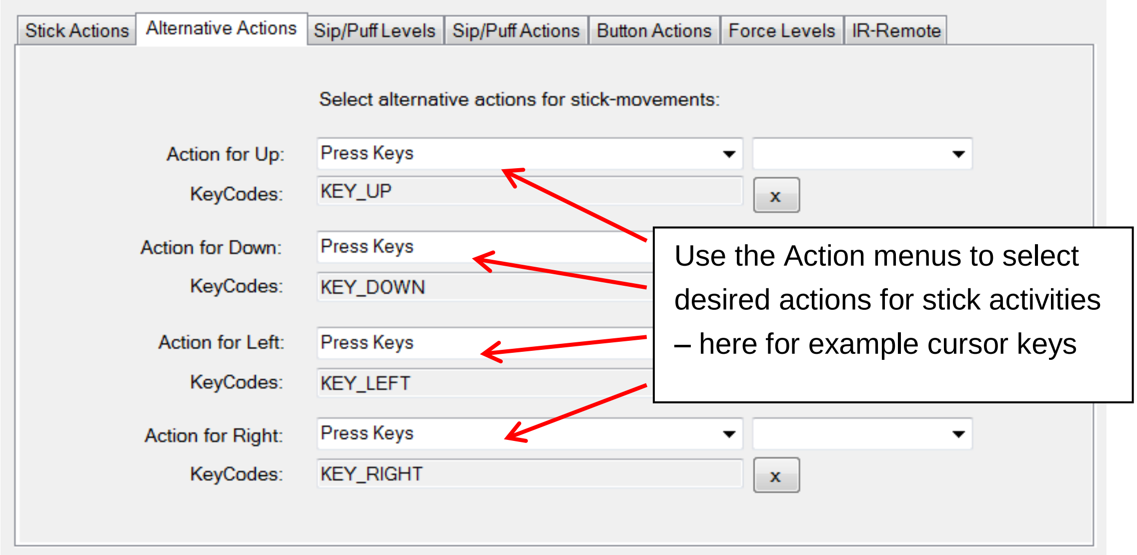
Deadzone 39

Maximum Speed 93

□ split X/Y

When using alternative actions, stick movements can be assigned to different actions!

According to the stick activities (up/down/left/right), there are four drop down menus from which you may choose an alternative action for each stick movement. By default, alternative actions for all four possible stick movements are preselected as cursor key presses.



Select alternative actions for stick-movements:

Action for Up: Press Keys
KeyCodes: KEY_UP

Action for Down: Press Keys
KeyCodes: KEY_DOWN

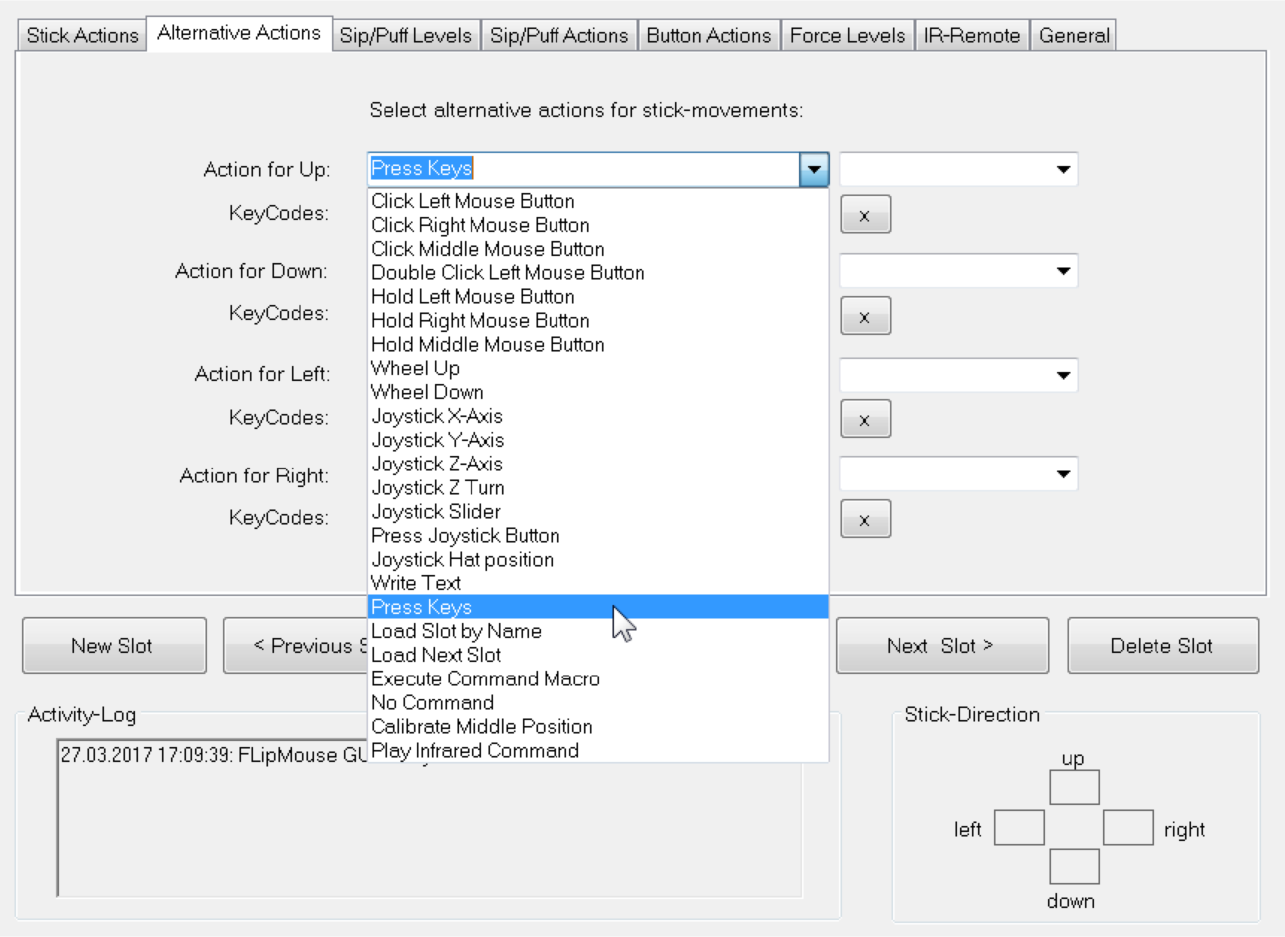
Action for Left: Press Keys
KeyCodes: KEY_LEFT

Action for Right: Press Keys
KeyCodes: KEY_RIGHT

Use the Action menus to select desired actions for stick activities – here for example cursor keys

The Action Menus

The Action menus provide different actions which can be assigned to desired user activities – for example to stick movements, sip/puff activity or external buttons. The actions can be selected via drop-down menus which are offered for all supported user activities:



Select alternative actions for stick-movements:

Action for Up: Press Keys

27.03.2017 17:09:39: FLipMouse GU

In the following, the individual actions will be briefly described.

Click Left / Middle / Right Mouse Button

Creates mouse button clicks. (The left/right or middle mouse button is pressed and released again after short time).

Double Click Left Mouse Button

Double clicking the left mouse button may be necessary in cases such as opening a file. However, producing a double click with the regular click mouse button function may not be convenient, so you may assign a double click to a desired user activity.

Hold Left / Middle / Right Mouse Button

The *hold* action creates a continued pressing of a particular mouse button (for example, when dragging a file, continuously pressing the left mouse click is necessary). The mouse button is released when the assigned user activity ends (for example when the sip/puff activity ends, or when the stick is moved back to middle position or when an external button is released).

Toggle Left / Middle / Right Mouse Button

The *toggle* action changes the state of a mouse button from *pressed* to *not pressed* and vice vera. This is useful for if a user has problems with a continuous sip/puff action which is needed for dragging etc.

Wheel Up / Down

The actions "*Wheel up*" or "*Wheel down*" emulate the mouse scroll wheel. The picture below displays an example of a scroll wheel. Triggering the "Wheel up" option results in upwards scrolling, while "wheel down" results in downward scrolling.

Joystick X/Y/Z/Slider movement

These actions can be used to send a desired value to the joystick axis. The selectable joystick axes are: X / Y / Z / Z-Turn / Slider1 and Silder2. The joystick is moved back to middle position when the associated user activity ends.

Joystick hat position

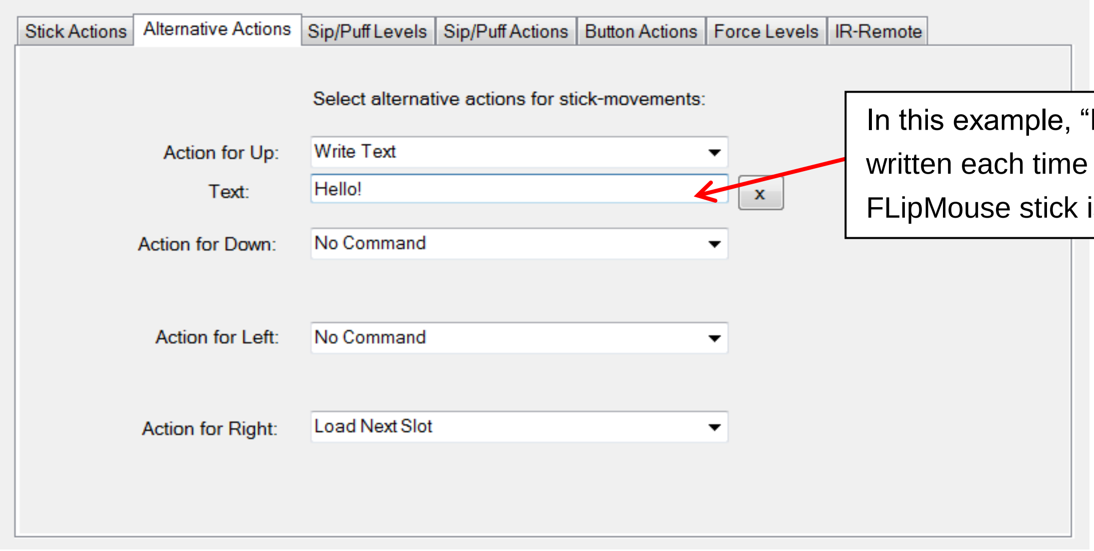
This action sets the orientation of the joystick hat (in degrees). Allowed values are:
-1, 0, 45, 90, 135, 180, 225, 270, 315. The value of -1 sets the hat to the middle (idle) position. The joystick hat is moved back to middle position when the associated user activity ends.

Press Joystick Button

This action can be used to press a desired joystick button. The FLipMouse device supports 32 buttons, so any value from 1 to 32 is allowed. The joystick button is released when the associated user activity ends.

Write Text

The "*Write text*" action allows typing a particular text/phrase whenever a user activity occurs (for example: write "Hello" whenever you move the FLipMouse stick up). Selecting this action displays a blank text box under the drop-down menu where the text can be entered, as shown below:



Select alternative actions for stick-movements:

Action for Up: Write Text

Text: Hello!

Action for Down: No Command

Action for Left: No Command

Action for Right: Load Next Slot

In this example, “Hello!” will be written each time the FLipMouse stick is moved

Press / Hold / Toggle Keys

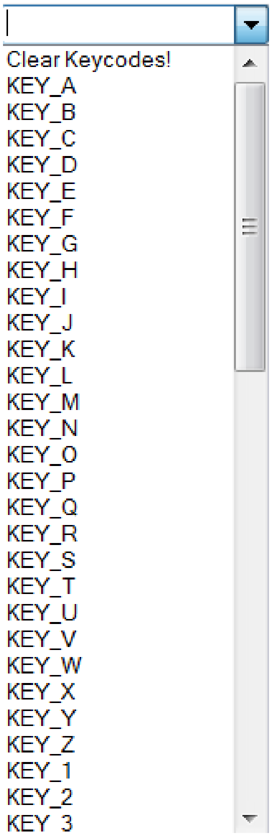
The “*Press Keys*” action allows pressing one or multiple keyboard keys. A second drop-down menu with possible key identifiers is displayed. **When a key is selected from this menu, it will be added** to the *KeyCodes* text field. Thus, multiple keyboard keys can be pressed simultaneously. The keys are released when the associated user activity ends. If you would like to remove or change the assigned keys, you must clear the currently assigned keys by pressing the “X” button next to the *KeyCodes* field.

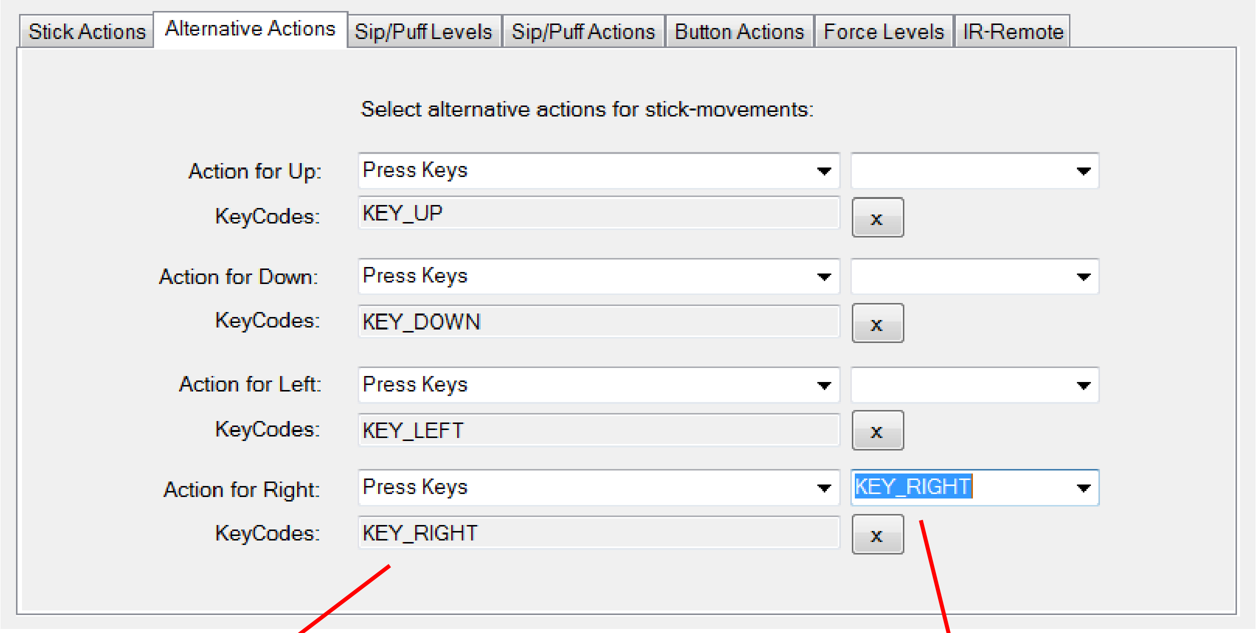
Common key combinations include:

KEY_CTRL + Z: triggers undo function
KEY_CTRL + C: triggers copy function
KEY_CTRL + V: triggers paste function
KEY_CTRL + KEY_ALT + KEY_ DELETE

The “*Hold Key*” action keeps the key pressed until the user activity ends. The “*Toggle Key*” action changes the key state from *pressed* to *not pressed* and vice versa with each user activity.

For a list of all supported key identifiers, see appendix or have a look at the FLipMouse Wiki / Github pages: https://github.com/asterics/FLipMouse/wiki/at-api.





Select alternative actions for stick-movements:

Action for Up: Press Keys

KeyCodes: KEY_UP

Action for Down: Press Keys

KeyCodes: KEY_DOWN

Action for Left: Press Keys

KeyCodes: KEY_LEFT

Action for Right: Press Keys KEY_RIGHT

KeyCodes: KEY_RIGHT

This gray text field is read-only and is automatically filled when selecting a key combination from the drop down menu on the right.

This drop down menu includes all the possible key identifiers to be added to the *KeyCodes* field.

Load Slot by Name

This action allows switching to another slot by its name. The name can be specified in the text field. This action is only relevant if you have stored multiple FLipMouse configuration slots. The LEDs indicate the slot number as described below.

Load Next Slot

This action switches to the next slot. If the last slot is already reached, triggering this action will switch to the first slot. This action is only relevant if you have stored multiple FLipMouse configuration slots. When you switch the slot, the built-in LED bulbs will change accordingly, showing the active slot.

There are three built in LEDs (red, yellow-orange, green), which will display the binary number for the slot position of the configuration that you have selected. In other words, if you saved two configurations, e.g. “Game settings” and “Mouse”, “Game settings” will be position 1 and “Mouse” will be position 2.

Additionally to the LEDs, an acoustic signal will indicate the change of the slot.

The following list shows the color codes of the LEDs for each slot position:

Position 1 will be displayed as 1 shining bulb:

🔴

Position 2 will be displayed as 1 shining bulb:

🟡

Position 3 will be displayed as 2 shining bulbs:

🔴 🟡

Position 4 will be displayed as 1 shining bulb:

🟢

Position 5 will be displayed as 2 shining bulbs:

🔴 🟢

Position 6 will be displayed as 2 shining bulbs:

🟡 🟢

Position 7 will be displayed as 3 shining bulbs:

🔴 🟡 🟢

Execute Command Macro

This action allows running a so-called *command macro* which consists of several individual actions. Thus, a desired sequence of actions can be performed which consists for example of a number of mouse moves, mouse clicks, text inputs or key presses. This action is powerful, but also a bit more complicated because the individual actions must be specified in form of so-called *AT-commands* which are separated by semicolons. For example: the following command macro will move the mouse cursor 100 steps to the left, 20 steps up, then wait 100 milliseconds and then perform a left mouse click: “MX 100; MY -20; WA 100; CL”. For a list and detailed explanation of all supported AT-commands, see appendix or hav a look at the FLipMouse Wiki / Github pages: https://github.com/asterics/FLipMouse/wiki/at-api.

No Command

If the action “*No Command*” is selected, no action will be assigned to the respective user activity.

Calibrate middle position

This action starts the calibration sequence for the middle position of the stick. Please refer to the description “Calibration of the stick’s middle position” on page 18.

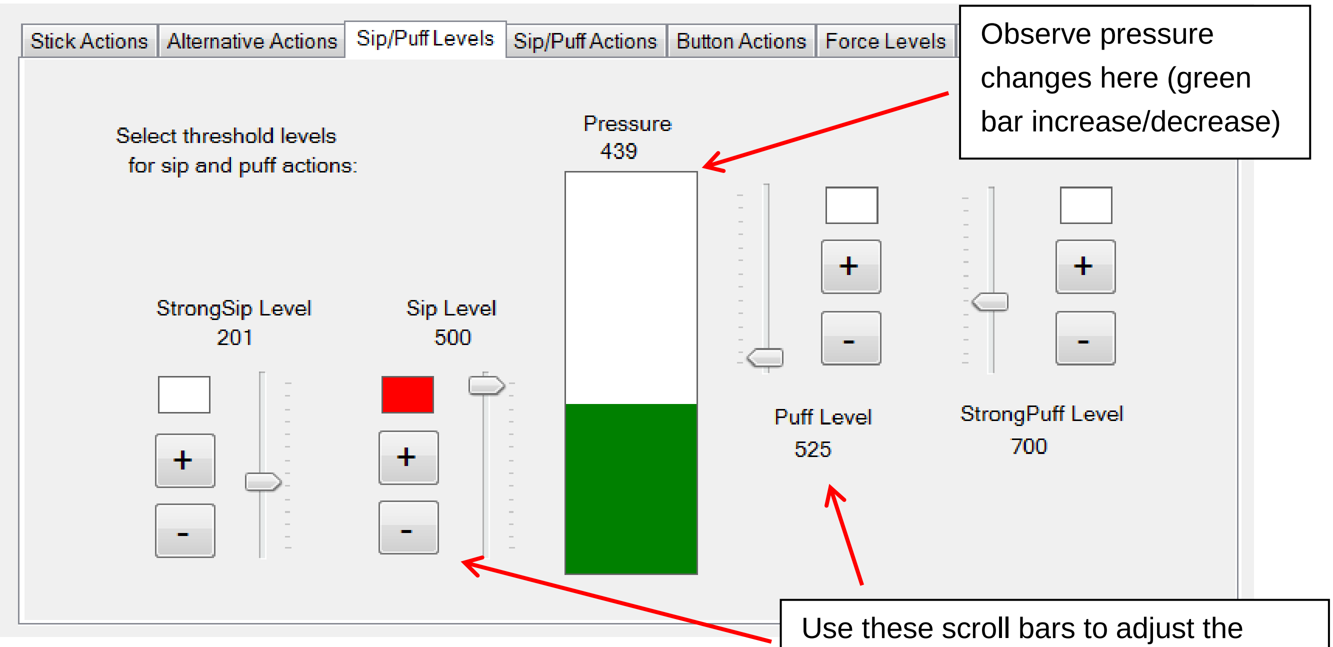
Play / Hold Infrared Command

This action plays an infrared code of given command name. The existing commands can be selected from the drop-down menu. The “play” action sends the recorded code once, whereas the “hold” action repeats the code until the user activity ends. For more information about infrared codes please refer to section “*Infrared Code Recording and Replay*”.

Sip/Puff Actions and Levels

When using the FLipMouse stick with the mouth, the user can trigger actions by sipping or puffing through the tube. The tube is connected to a pressure sensor, which outputs a value that corresponds to the detected pressure. When the user sips, the sensor value decreases, and when the user puffs the value increases.

You may assign actions for sipping or puffing using the tab “*Sip/Puff Actions*”. You can select the thresholds for sipping and puffing as desired, using the tab “*Sip/Puff Levels*”. When the FLipMouse is connected, you can monitor the pressure values and the triggering of sip / puff actions in the graphical user interface:



Select threshold levels for sip and puff actions:

Pressure 439

StrongSip Level 201

Sip Level 500

Puff Level 525

StrongPuff Level 700

Observe pressure changes here (green bar increase/decrease)

Use these scroll bars to adjust the threshold pressure for sip or puff. The red indicator shows when the pressure is reached.

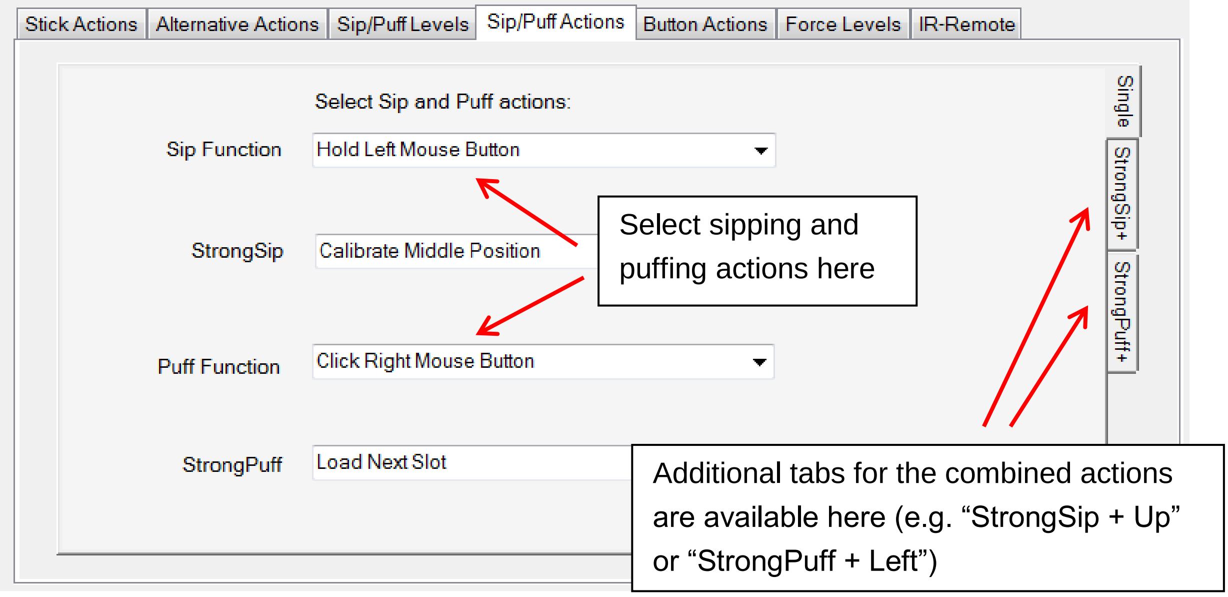
Note that the idle pressure level (without sip or puff) is around 512, and the level increases when you puff into the mouthpiece, indicating increased pressure.

StrongSip and StrongPuff

To enrich the functionality of the FLipMouse, especially for persons who cannot access external switches, the StrongSip and StrongPuff modes are available. These modes allow additional actions which can be performed via strong sip or puff activities followed by a stick movement (up / down / left / right). These actions will only be triggered once – then the mode switches back to normal. Entering StrongSip or StrongPuff mode is indicated via an acoustic signal (high tone). If the stick is moved within one second, the corresponding combined action (for example “StrongSip+Up” is triggered). If the stick is not moved within one second, the single StrongSip or StrongPuff action will be triggered. Thus, a total of 10 additional actions can be performed.

Sip and Puff Actions tabs

The actions assigned to the sip- and puff activities can be defined in the “*Sip/Puff Actions*” tab. This works in exactly the same manner as selecting alternative action for stick movements or button actions – via the drop-down action menus for the assigned functions. The Sip/Puff actions tab features three sub-tabs for the assignment of the combined “StrongSip+” and “StrongPuff+” actions which are followed by a stick movement.



Select Sip and Puff actions:

Sip Function: Hold Left Mouse Button

StrongSip: Calibrate Middle Position

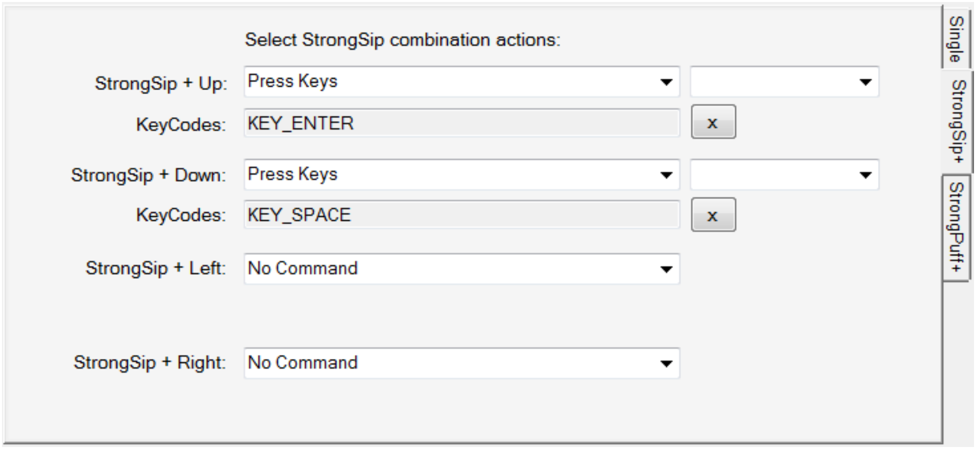
Puff Function: Click Right Mouse Button

StrongPuff: Load Next Slot

Select sipping and puffing actions here

Additional tabs for the combined actions are available here (e.g. “StrongSip + Up” or “StrongPuff + Left”)

Example of combined actions selected in the StrongSip+ tab:



Select StrongSip combination actions:

StrongSip + Up: Press Keys

KeyCodes: KEY_ENTER

StrongSip + Down: Press Keys

KeyCodes: KEY_SPACE

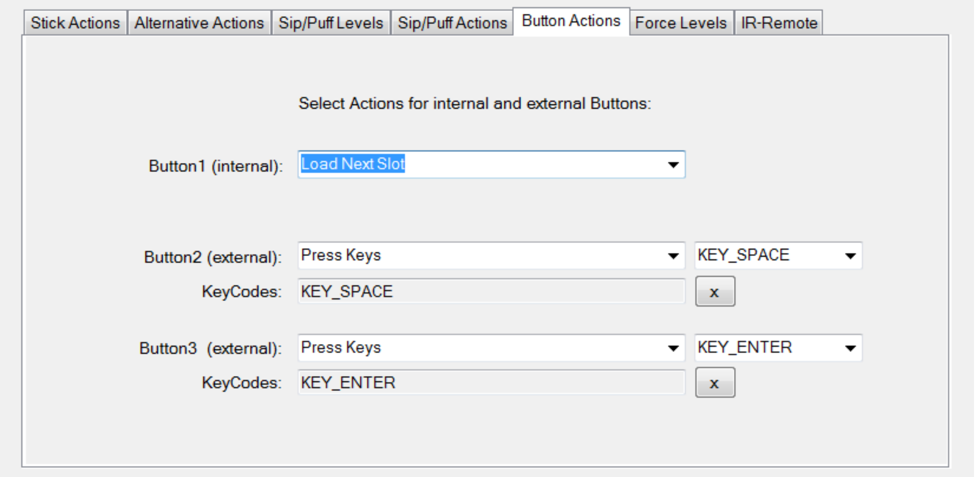
StrongSip + Left: No Command

StrongSip + Right: No Command

Buttons Actions

The FLipMouse has one built-in button which is ready to use – referred to as “*Button 1 (internal)*” in the tab “*Button Actions*”, and two 3,5mm jack plugs where two external buttons may be connected (referred as “*Button2*” and “Button3”).

These three buttons can trigger actions that are assigned in the “*Button Actions*” tab:



Select Actions for internal and external Buttons:

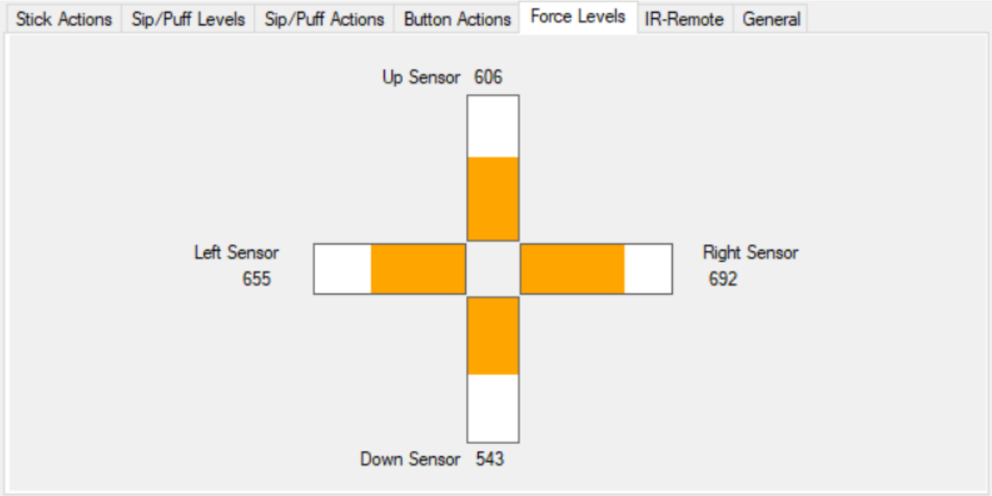
Button1 (internal): Load Next Slot

Button2 (external): Press Keys KEY_SPACE
KeyCodes: KEY_SPACE

Button3 (external): Press Keys KEY_ENTER
KeyCodes: KEY_ENTER

View Force Levels

The FLipMouse stick is attached to four force sensors – one for each movement direction (up, down, left, right). The numeric value of each of those sensors is displayed in the “*Force Levels*” tab. The live values can be monitored when the FLipMouse device is connected. Whenever the stick is moved, the sensor values change. Specifically, a movement in each direction increases the respective sensor value. The sensor values also depend on the use of the adjustment screws which change the force which is applied to the sensor pads via 4 metal springs. Therefore, when assembling the FLipMouse, the live sensor values can be used as guidance. **When not touching the stick, the values should look similar to screenshot below. All values should decrease and raise similarily when moving the stick. Significant deviation in one direction indicates a sensor problem. The sensors should be checked and might need to be replaced.**

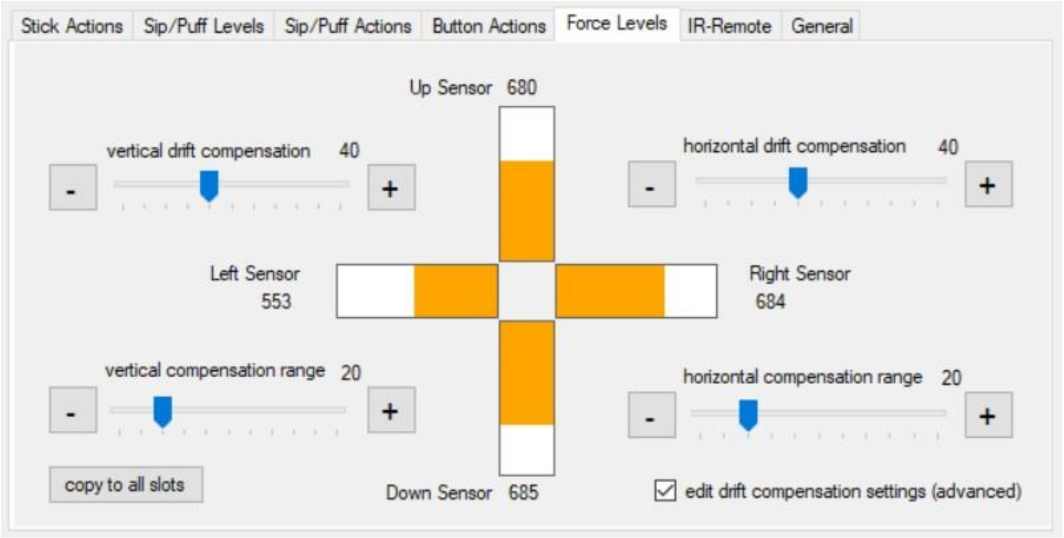


Up Sensor 606
Left Sensor 655
Right Sensor 692
Down Sensor 543

Live display of force sensor values

Drift Compensation Settings

Due to mechanical reasons, the force sensors of the FLipMouse show small inaccuracies which can result in mouse cursor drifting. This effect is critical if using very small deadzone settings (very sensitive mouse control): if you move the mouthpiece/joystick to one direction and then release it, the mouse cursor continues to slightly “drift” in this direction because the sensor values do not settle at the calibrated middle position. Since Version 2.9, the FLipMouse offers a “drift compensation” feature, where the software tries to calculate the expected drift caused by the last interaction and corrects the bias, so that the performance for small deadzones is improved. These advanced settings can be activated in the “Force Levels” Tab by enabling the checkbox “edit drift compensation settings”:



Up Sensor 680

vertical drift compensation 40

horizontal drift compensation 40

Left Sensor 553

Right Sensor 684

vertical compensation range 20

horizontal compensation range 20

Down Sensor 685

☑ edit drift compensation settings (advanced)

If “edit drift compensation settings” is enabled, four sliders appear around in the Force Levels display. The sliders on the left side control the vertical drift compensation settings (X-Axis), the sliders on the right side control the horizontal drift compensation settings (Y-Axis).

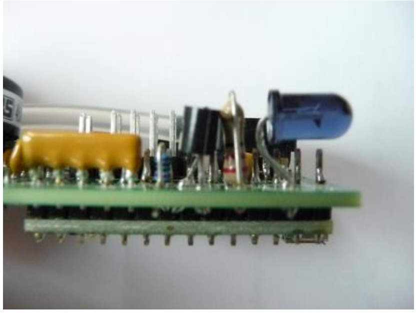
The “drift compensation” value represents a gain value (level) for the correction, whereas the “compensation range” value represents the amount of force which is taken into account for the calculation. The best values for a particular FLipMouse can be determined by experimentation.

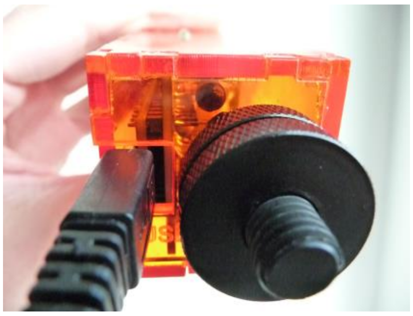
For example:

- If you push the mouthpiece upwards and after release the mouse cursor continues to drift upwards, try to increase the vertical compensation value
- If you push the mouthpiece to the left and after release the mouse cursor drifts to the right, try to decrease the horizontal compensation value.

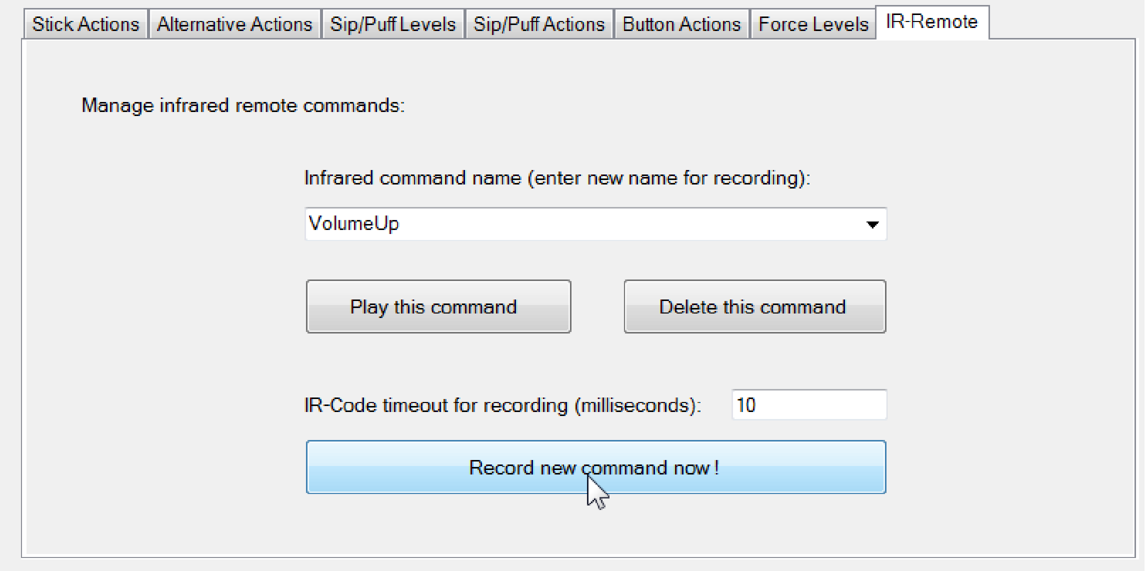
Infrared Code Recording and Replay

The FLipMouse device includes an infrared remote receiver module and a high current IR LED so that it can record and replay remote control codes of many remote controls used in consumer electronic devices. If you want to use this feature of the FLipMouse, make sure that the IR sending LED is mounted so that it points to the device you want to control (usually the IR LED points to the direction the user looks when using the FLipMouse). For instructions how to connect an external IR LED to increase the transmit power, please visit the FLipMouse Wiki pages: https://github.com/asterics/FLipMouse/wiki





To record a new IR code enter the intended command name and press “Record new command” - then point your IR remote to the side of the FLipMouse where the LEDs stick out and send the IR command (you have 10 seconds for this). The new command is now stored in the FLipMouse and can be chosen and replayed using the selection box in the GUI window. Additionally, the command can be entered as text parameter for the “Play Infrared Command” action. If the command is not correctly replayed, try to increase the IR-Code timeout – for example to 250 milliseconds.



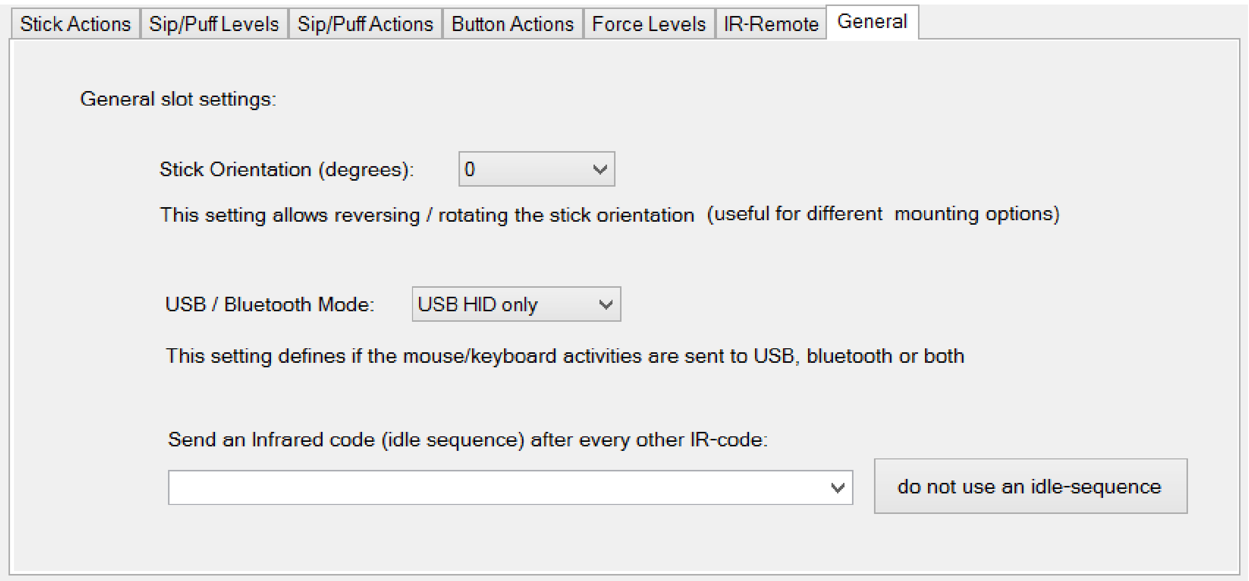
Manage infrared remote commands:

Infrared command name (enter new name for recording): VolumeUp

IR-Code timeout for recording (milliseconds): 10

General Settings

The final tab is dedicated to “General Settings” and allows changing the *Stick orientation*, the USB/Bluetooth mode and other settings of the current slot:



General slot settings:

Stick Orientation (degrees): 0

This setting allows reversing / rotating the stick orientation (useful for different mounting options)

USB / Bluetooth Mode: USB HID only

This setting defines if the mouse/keyboard activities are sent to USB, bluetooth or both

Send an Infrared code (idle sequence) after every other IR-code:

Stick Orientation

The FLipMouse is mounted using a HotShoe 3/8” mounting screw which located on the side of the FLipMouse case. The orientation can be changed according to the user’s preferences. If the FlipMouse is mounted e.g. upside-down, the stick orientation can be selected accordingly so that the up/down/left/right movements are still interpreted correctly. This combo box provides orientation settings for 0 / 90 / 180 and 270 degrees, so that any mounting orientation is possible.

USB/Bluetooth mode

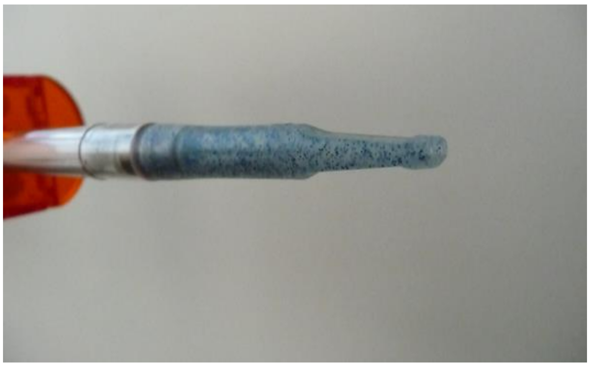
This setting is only relevant if a Bluetooth add-on module is used (see FLipMouse Wiki pages for more information: https://github.com/asterics/FLipMouse/wiki/bt-conn). Using this combo box selection, it is possible to decide if mouse / keyboard activities are sent via USB cable, via Bluetooth connection, or both. Thus, dedicated slots for USB and Bluetooth can be created, so that a user can switch e.g. from a laptop computer (where the FLipMouse is connected via cable) to a wireless connection (phone / tablet) and back.

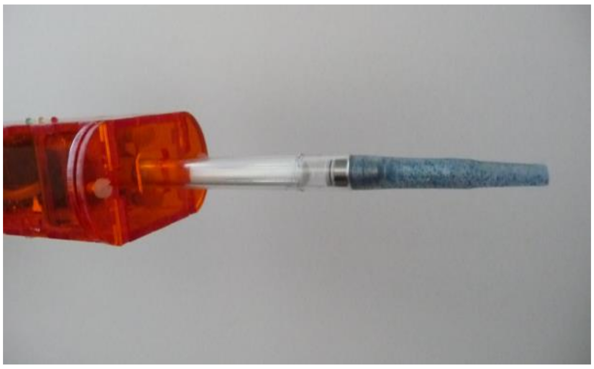
Infrared Code idle sequence

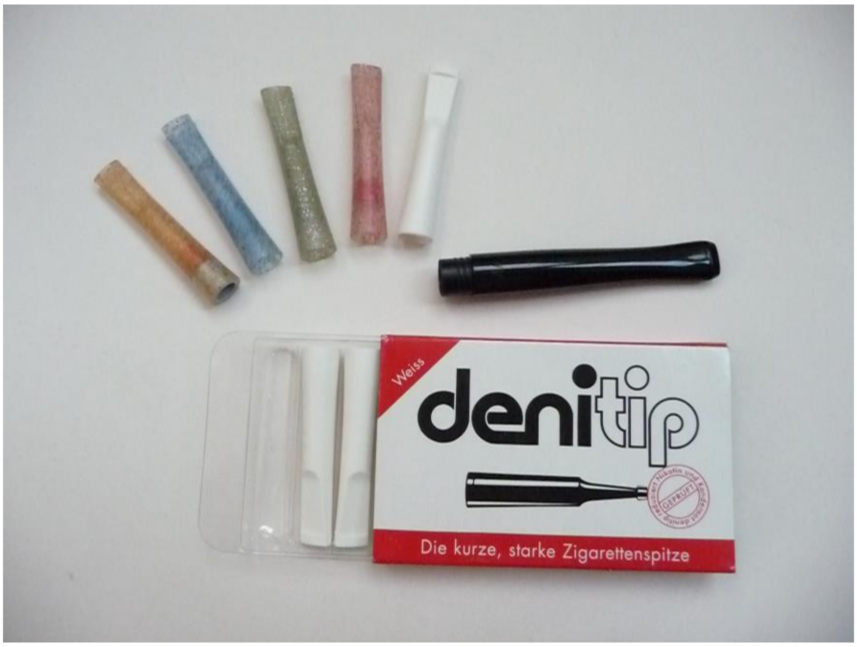
If an infrared code name is selected from this box, the according code will be appended after every other IR-code. The idle sequence can be changed per slot. This is useful for certain devices, e.g. IR-controlled toys which rely on a special IR code to indicate the idle (inactive) state.

Mod your Mouthpiece?

Several users reported that they would prefer a different shape of the mouthpiece. It is of course possible to apply your own modifications and extensions which feel comfortable to you. Here we want to give some suggestions using affordable, disposable cigarette holders. The link between the *Hygienic Stick* and the cigarette holder is established with a small brass tube. It is also possible to cut the cigarette holder and insert it into the *Hygienic Stick* so that a tight fit is ensured.







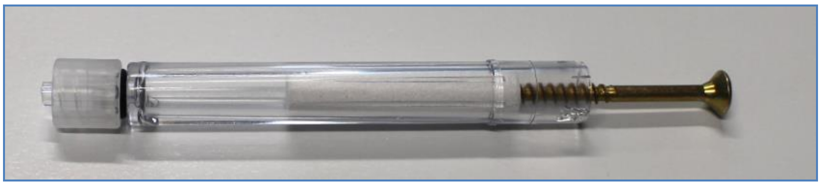
Regardless of your desired mouthpiece modifications: Always take care for a hygienic operation and apply the cleaning procedures as described in the following chapter!

Instructions for cleaning and maintenance

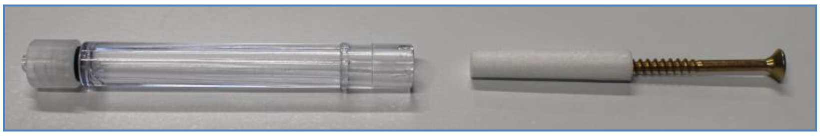
If the mouthpiece is used to actuate the cursor movements via lip/mouth movements, a careful cleaning procedure is essential to avoid bacteria and health problems! Read this section carefully and apply the needed cleaning procedures whenever necessary.

Exchanging the saliva filter:

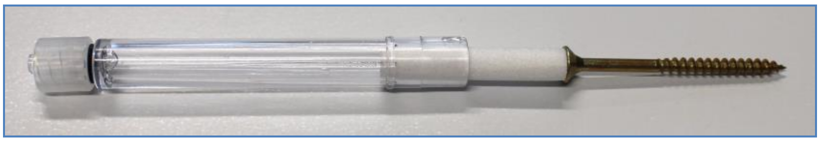
The saliva filter is placed inside the hygienic stick. Usually, no (or nearly no) saliva should enter the stick – but it strongly depends on the person using the mouthpiece and his/her condition. If saliva enters the *Hygienic Stick*, the saliva filters prevents the moisture entering the inner parts of the FLipMouse (respectively the pressure sensor). The filter can be replaced as shown below:



Removal of the saliva filter, using a normal screw to grab the filter



The removed filter



Inserting a new filter (pushing it back into the hygienic stick)

Disinfection of the mouthpiece/hygienic stick:

To disinfect a mouthpiece, use boiling water. Place the *Hygienic Stick* (without saliva filter) in boiling water for at least five minutes. Then let the *Hygienic Stick* dry before the next application.

Ordering replacement sticks and filters:

Saliva filters and *Hygienic Sticks* can be directly ordered from the *Quadjoy* online shop: https://quadjoy.com/shop. We would like to thank Bill and Debra Street for letting us use their *Quadjoy Hygienic Stick* together with our FLipMouse device – you guys are great !!

Hygienic considerations and risks

If the mouthpiece is actuated with the lips, it is important to respect the following hygienic precautions:

a. One mouthpiece must be used exclusively by one person
b. The filter tip must be replaced whenever saliva or dirt have formed
c. The mouthpiece must be cleaned whenever saliva or dirt have formed (see chapter “Cleaning and Maintenance”)

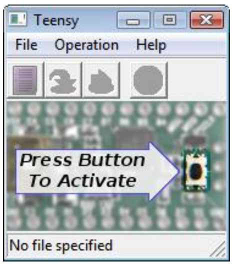
With the lack of hygiene concerning the mouthpiece, especially when the mouthpiece is used by more than one person, it is possible that

- skin rash arises
- viruses and bacteria are transmitted causing diseases
- dirt formats in the mouthpiece, which is dangerous to health.

With wrong positioning of the FLipMouse there is the risk of feeling uncomfortable after longer periods of use, and it is possible to get muscle cramps or muscle pain due to unusual head positioning – pay attention to a convenient position and schedule pauses if you work for longer time periods!

Updating the firmware / GUI software

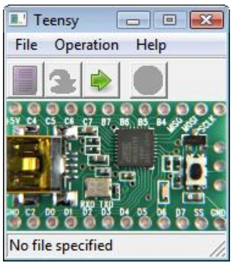
The software (firmware) which runs on the Teensy microcontroller inside the FLipMouse must match the version of the Settings Mananger (GUI) running on the PC. Since version 2.6, the Settings Manager supports automatic update of the Teensy firmware. However, the firmware can also be updated via the Arduino IDE or the Teensy loader application if the automatic update fails. Find the most recent versions in the latest release package:
https://github.com/asterics/FLipMouse/releases
Unzip the FlipMouse.zip package and start the teensy.exe program (the Teensy Loader)



Press Button To Activate

No file specified

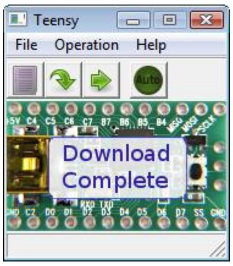
The Teensy Loader user interface should look as on the picture on the left side. If there is any security warning (related to Windows User Access Control), please ensure that the Publisher is “PJRC.COM, LLC” and click on “Run”.



No file specified

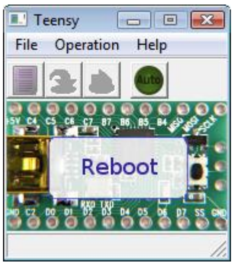
As next step, activate the “download mode” of the FLipMouse by using some pointed tool (needle, small screwdriver,…) to press the reset button which is accessible via the small hole on the bottom side. After a few seconds (driver installation), the Teensy Loader GUI should indicate connection to the FLipMouse as shown on the left side.

Choose *“File→Open HEX file”* and select the file “FLipWare.hex” from the FLipMouse folder (or location where you downloaded the release .zip file).



Download Complete

Select *“Operation→Program”*. After a few seconds, you should see the notification “Download complete”, as on the left hand side picture.
(If this step does not work in the first place, try to unplug and replug the FLipMouse and restart the Teensy Loader software.



Reboot

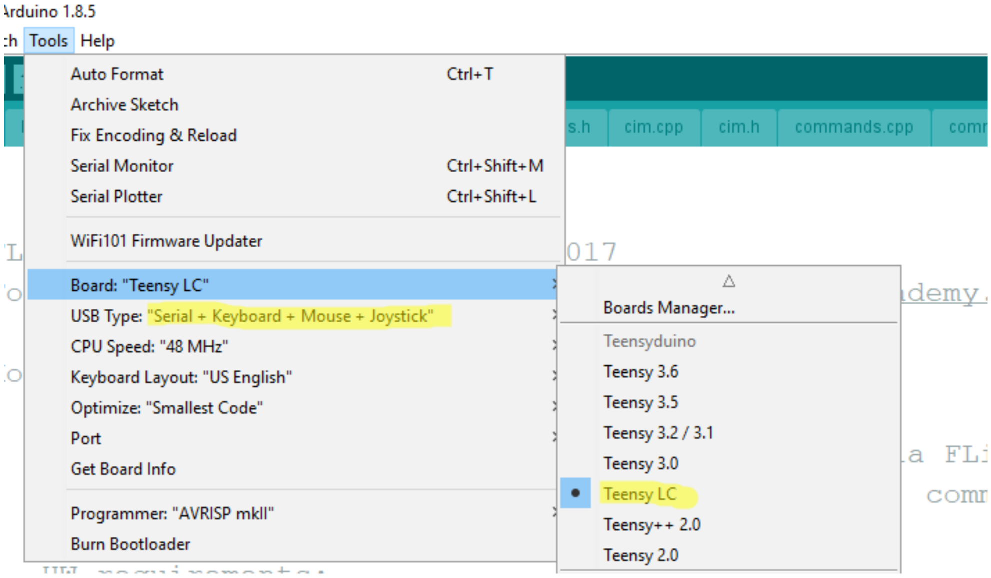
To finish the firmware installation, select *“Operation→Reboot”*.
You should read “Reboot” and the FLipMouse should beep for a short time.

To update the GUI software, copy the file “FlipMouseGUI.exe” from the zip package to the desired location (replacing the old FlipMouseGUI.exe file).

Building the Firmware

If you want to adapt the software (firmware) of the FlipMouse, follow these steps:

1. Clone the FLipMouse repository, https://github.com/asterics/FLipMouse
2. Download and install the Arduino IDE, https://www.arduino.cc/en/main/software
3. Download and install Teensyduino, https://www.pjrc.com/teensy/td_download.html
4. Open FLipWare/FLipware.ino with the Arduino IDE, see: https://github.com/asterics/FLipMouse/blob/master/FLipWare/FLipWare.ino
5. Use the following settings in Arduino IDE:
    - Tools -> Board -> Teensy LC
    - Tools -> USB Type -> "Serial + Mouse + Keyboard + Joystick"



Board: "Teensy LC"
USB Type: "Serial + Keyboard + Mouse + Joystick"
CPU Speed: "48 MHz"
Keyboard Layout: "US English"
Optimize: "Smallest Code"
Programmer: "AVRISP mkII"

Further instructions and troubleshooting

For further instruction and troubleshooting please refer to the GitHub Wiki of the FlipMouse project: https://github.com/asterics/FLipMouse/wiki

Appendix: Macro commands and Key codes

Supported Macro commands:

Abbreviation	function	example / explaination
CL	click left mouse button	presses + releases mouse button
CR	click right mouse button	
CM	click middle mouse button	
CD	click double with left mouse button	
HL	hold the left mouse button	keeps mouse button pressed until action ends
HR	hold the right mouse button	
HM	hold the middle mouse button	
RL	release the left mouse button	releases mouse button
RR	release the right mouse button	
RM	release the middle mouse button	
TL	toggle left mouse button	changes the state of the mouse button
TR	toggle right mouse button	
TM	toggle middle mouse button	
WU	move mouse wheel up	
WD	move mouse wheel down	
WS <int>	set mouse whell step size	WS 3 sets the wheel stepsize to 3 rows
MX <int>	move mouse in x direction	MX 4 moves cursor 4 pixels to the right
MY <int>	move mouse in y direction	MY -10 moves cursor 10 pixels up
RO <uint>	rotate stick orientation	RO 180 flips x any y movements of stick/mouthpiece
KW <string>	keyboard write string	KW Hello! writes "Hello!" on the keyboard
KP <string>	key press: Press keyboard keys (once). Keys are identified by keyboard shortcuts (see list below)	KP KEY_UP presses the "Cursor-Up" key KP KEY_CTRL KEY_ALT KEY_DELETE presses all 3 keys
KH <string>	key hold: hold keyboard keys	KH KEY_LEFT holds the "Cursor-Left" key
KT <string>	key toggle: toggle keyboard keys the key will remain pressed until "AT KT" command is sent again or a "AT KR" command is sent	KT KEY_DOWN toggles the "Cursor-Down" key
KR <string>	Key release	KR KEY_UP releases the "Cursor-Up" key

RA	releases all Release all currently pressed keys and mouse buttons	
JX <int>	set joystick x axis	JX 512 sets the x-axis to middle position
JY <int>	set joystick y axis	JY 1023 sets the y-axis to full up position
JZ <int>	set joystick z axis	JZ 0 sets the z-axis to lowest position
JT <int>	set joystick turn axis	JT 512 sets the rotation to middle position
JS <int>	set joystick slider position	JS 512 sets the slider to middle position
JP <int>	press joystick button	JP 1 presses joystick button 1
JR <int>	release joystick button	JR 2 releases joystick button 2
JH <int>	set joystick hat position	JH 45 sets joystick hat to 45 degrees possible values are: 0, 45, 90, 135, 180, 225, 270, 315 and -1 to set center position
MM <int>	mouse mode: cursor on (int==1) or alternative functions on (int==0)	MM 1 switches to mouse mode
CA	calibrate zero position	calibrates stick/mouthpiece middle position
IP <string>	play infrared command	IP channelUp plays infrared command “channelUp” (if this ir-command name has been recorded)
WA <int>	wait a certain number of milliseconds	WA 100 waits 100 milliseconds

Please note that certain commands for adjusting FlipMouse settings are not shown in this table (for example setting thresholds or accelerations, recording IR-commands etc.)
For a full list of commands please refer to the FlipMouse wiki: https://github.com/asterics/FLipMouse/wiki/at-api.

Supported Key Identifiers:

Letters
KEY_A KEY_B KEY_C KEY_D KEY_E KEY_F KEY_G KEY_H KEY_I KEY_J KEY_K KEY_L KEY_M KEY_N KEY_O KEY_P KEY_Q KEY_R KEY_S KEY_T KEY_U KEY_V KEY_W KEY_X
Digits
KEY_1 KEY_2 KEY_3 KEY_4 KEY_5 KEY_6 KEY_7 KEY_8 KEY_9 KEY_0
Function keys
KEY_F1 KEY_F2 KEY_F3 KEY_F4 KEY_F5 KEY_F6 KEY_F7 KEY_F8 KEY_F9 KEY_F10 KEY_F11 KEY_F12 KEY_F13 KEY_F14 KEY_F15 KEY_F16 KEY_F17 KEY_F18 KEY_F19 KEY_F20 KEY_F21 KEY_F22 KEY_F23 KEY_F24
Navigation keys
KEY_UP KEY_DOWN KEY_LEFT KEY_RIGHT KEY_TAB KEY_PAGE_UP KEY_PAGE_DOWN KEY_HOME KEY_END
Special keys
KEY_ENTER KEY_SPACE KEY_BACKSPACE KEY_DELETE KEY_INSERT KEY_ESC KEY_NUM_LOCK KEY_SCROLL_LOCK KEY_CAPS_LOCK KEY_PAUSE KEY_SEMICOLON KEY_COMMA KEY_PERIOD KEY_MINUS KEY_EQUAL KEY_SLASH KEY_BACKSLASH KEY_LEFT_BRACE KEY_RIGHT_BRACE KEY_QUOTE KEY_TILDE KEY_MENU
Keypad keys
KEYPAD_1 KEYPAD_2 KEYPAD_3 KEYPAD_4 KEYPAD_5 KEYPAD_6 KEYPAD_7 KEYPAD_8 KEYPAD_9 KEYPAD_0 KEYPAD_SLASH KEYPAD_ASTERIX KEYPAD_MINUS KEYPAD_PLUS KEYPAD_ENTER KEYPAD_PERIOD
Alternate function keys
KEY_SHIFT KEY_CTRL KEY_ALT KEY_RIGHT_ALT KEY_GUI KEY_RIGHT_GUI

Contact Information

University of Applied Sciences Technikum Wien

Faculty of Electronic Engineering
Höchstädtplatz 6
A-1200 Wien
AUSTRIA

AsTeRICS Foundation

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

Disclaimer

The University of Applied Sciences Technikum Wien and the AsTeRICS Foundation 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 Ilana Snitkovski for her support in creating this User Manual.

This project was partly funded by the City of Vienna,
Municipal department of Economic Affairs, Labour and Statistics (MA 23),
in course of the projects AsTeRICS Academy (14-02), ToRaDes (18-04)
and Wissensdrehscheibe für Barrierefreie Technologien (26-02)