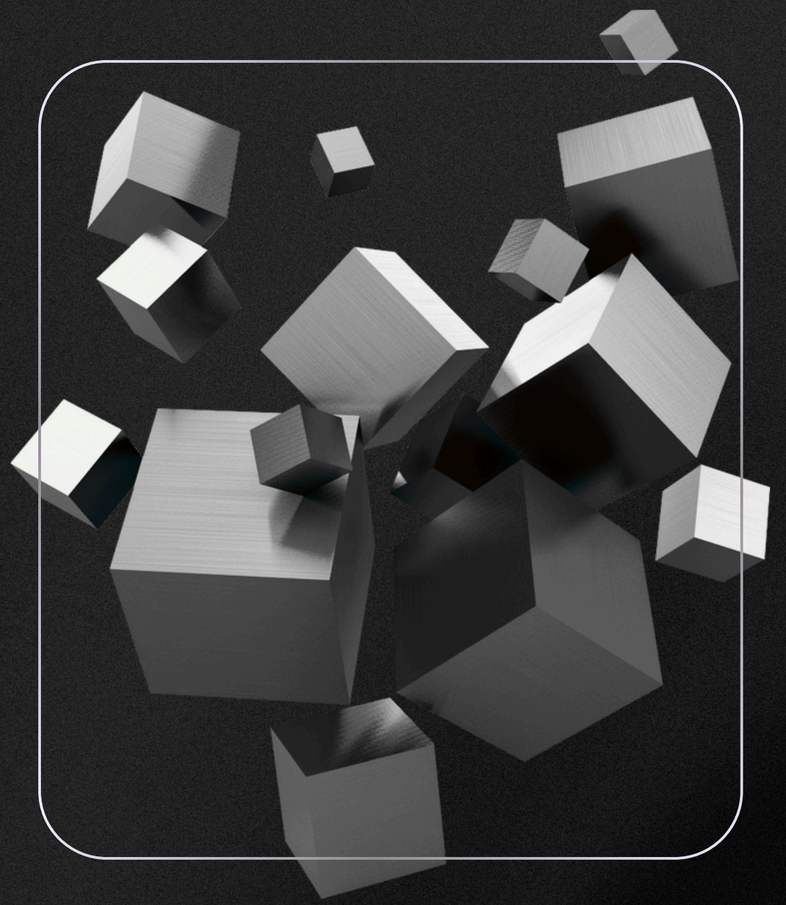
October 2025

TESI

Manual

Tying embodied sound with TESI and connecting it to DAW (digital audio workstations)



Presented by : **Tadhg Mnensa Mc Carthy, Thijs Rexwinkel, Stan Wijnen, Naiara Barrio, Thaksana Parameswaran**

List of Contents

What is TESI

Sound design often requires specific tools, technical knowledge, and experience, which can make it challenging for designers to explore. This is where the TESI (Tangible and Embodied Sound Interaction) toolkit comes in, providing an accessible way to experiment with sound as a design material.

TESI uses ESP32 boards with built-in sensors to capture interactions, OOCSI to connect and exchange data between devices, and Max to map this sensor data to sound in real time...

Importance of the TESI

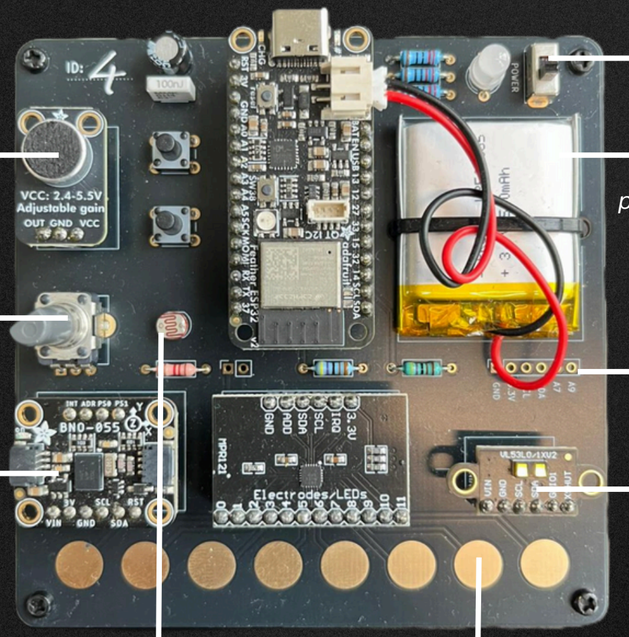
TESI makes it easier to build and test new ways of using sound as an active part of interaction design. With it, you can explore how sound can respond, adapt, and evolve based on human actions or environmental changes.

Goals of this manual

This manual was created to guide new users in setting up and working with the TESI toolkit. Our main goal is to make the system easier to understand and use, addressing challenges such as system complexity (how software, sensors, and networks interact) and accessibility (reducing barriers for those without prior experience in sound design).

Technical details of the TESI

The TESI contains multiple onboard sensors, offering versatile interaction modalities.
In the picture on the right, an overview is visible of all the available sensors on the TESI



Microphone (MAX4466)

Converts sound waves into electrical signals for audio detection or recording.

Potentiometer

A variable resistor that adjusts voltage output based on its knob position.

Motion sensor (MBO055)

Motion and orientation sensing

Power switch

Lipo battery

provides portable power to the circuits.

Headers for extra input

Distance sensor (Time of flight VL53L3CX)

Laser sensor that measures distance using reflected infrared light.

LDR (Light Dependent Resistor)

Changes resistance based on light intensity to measure brightness levels.

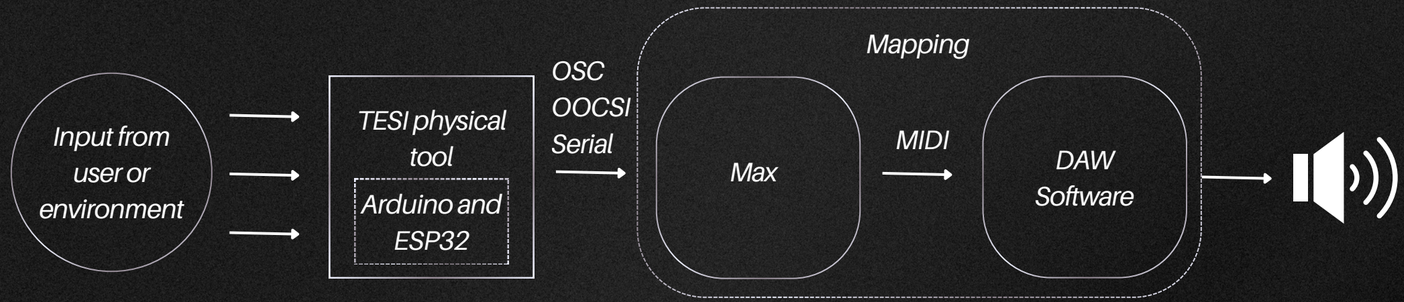
Captive touch sensors (MPR121)

Detects touch or proximity by sensing changes in electrical capacitance.

Connections to softwares

Mapping sound with the TESI tool involves several software programs working together exchanging data.

- Arduino IDE: Used to write and upload code that reads sensor inputs and sends data (like distance or touch) to the computer via serial communication.
- Max (Cycling ‘74): A visual programming environment that receives the sensor data, processes it, and converts it into MIDI.
- LoopMIDI: Creates a virtual MIDI port that lets Max send MIDI data to other software (like a DAW) inside the same computer.
- DAW (Digital Audio Workstation): Software like FL Studio or Ableton Live that receives MIDI input to trigger sounds and effects.



Oocsi and OSC

For the data communication, there are two options to be chosen from, OOCSI and OSC

OSC (Open Sound Control): Sends data over a network . This is good for real-time, flexible control between devices or computers and requires predetermined target addresses.

OOCSI: A messaging framework for interactive systems. This is better for multi-device communication and prototyping complex setups. OOCSI implements a client-server model over TCP that allows devices and software clients to publish and subscribe to named channels.. OOCSI has slightly more latency compared to OSC.

Both can be applied, but OSC is concidered a more standard approach. OOCSI is often used for prototyping and gives higher control when using multiple devices over the network.
For future application, it is up to the user which approach fits best.

Potential outlets

The TESI tool can be MIDI mapped on DAW's (Digital Audio Workstations). Some examples of possble applications are:

- *FL*
- *Ableton Live*
- *Pro tools*
- *Reaper*
- *Garage band*
- *Studio 1*

Setting up Arduino

To connect the TESI toolkit to a dedicated network, a few adjustments need to be made inside the code.

To update and upload the code, the Arduino IDE is used. After installing the latest version of the Arduino IDE, install the libraries listed and install the boards.

Download the TESI .zip file provided and extract it. Inside the file, go to: .\arduino\tesi to find and open the Arduino file.

Go to .\arduino\tesi\src to find the secrets-template.h to change the values to your network

Arduino IDE

https://www.arduino.cc/en/software/

Boards to install

ESP32 by Espressif Systems

https://learn.adafruit.com/adafruit-esp32-feather-v2/arduino-ide-setup

Libraries to install

OSC Message <<OSCMessage.h>> 3.5.8 by Adriaan Feed
Adafruit MPR121 1.1.3 by Adafruit
Adafruit BNO055 1.6.4 by Adafruit
Adafruit VL53L0X 1.2.4 by Adafruit
Adafruit VL53L1X 3.1.2 by Adafruit

Secrets.h

Rename the file from secrets-template.h to secrets.h

Change the following variables:
SECRET_SSID: Network name to connect to
SECRET_PASSWORD: Network password
SECRET_IP: IPv4 of your computer within the preferred network

tesi.ino | config.cpp | secrets.h

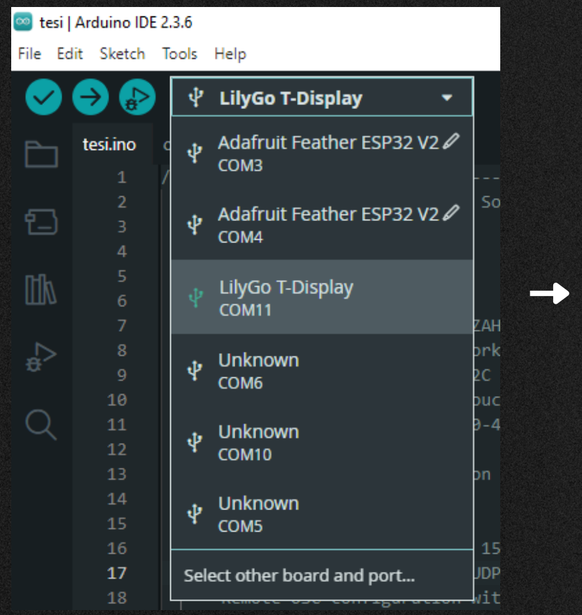
```
// secrets.h

// Contains details used for Wireless connections - must never be shared on github/gitlab etc.
// This actual file is included in .gitignore so it is never updated to avoid this.
// When creating your own project copy or rename secrets.template.h to secrets.h

#define SECRET_SSID          "YOUR_WIFI_SSD"
#define SECRET_PASSWORD      "YOUR_WIFI_Password"
#define SECRET_IP            "XXX.XXX.XXX.XXX"
#define SECRET_HOSTSERVER    "oocsi.id.tue.nl"
#define SECRET_OUTPORT       8001
#define SECRET_INPORT        9001
#define SECRET_OOCSI_NAME    "TESI_####"
#define SECRET_CHANNEL       "TESICHANNEL"

```

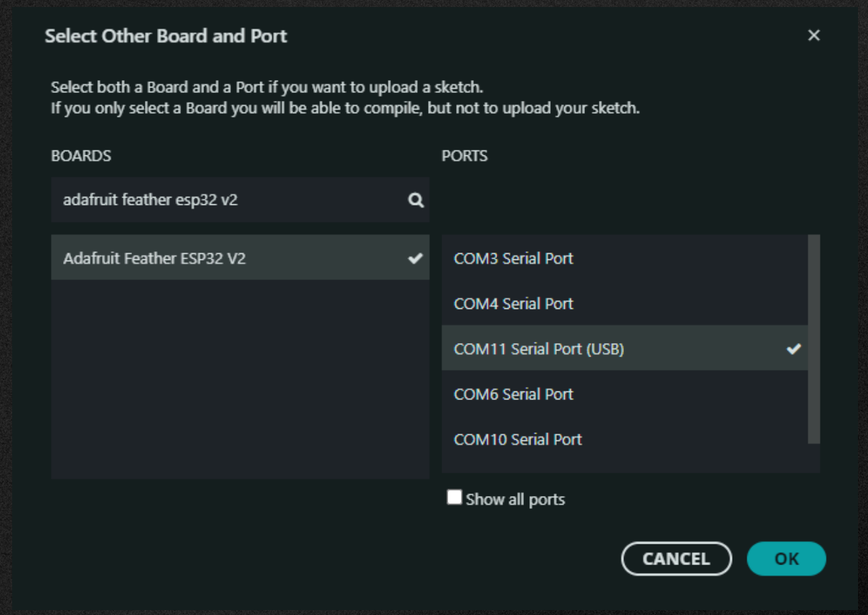
Setting up Arduino

Select the ‘Adafruit Feather ESP32 V2’ board in the board manager and upload the code by pressing the upload button

If you want to work with multiple TESI boards, find the config.ccp file in the .\arduino\tesi\src folder

Select the correct board





Connect multiple TESI devices

In the config.cpp file, line 6, the device index is assigned. When you want to connect multiple devices to the same netwerk, edit this number for each device (device one = index 1, device two = index two)

Setting up Max 9

After setting up Arduino, install Max 9 from the Cycling74 website. After the installation is complete, also isntall the necessary libraries (or packages as Max calls them).

The first three can be downloaded directly through max by scrolling down on the page and click ‘Open in Max'. Max wil automatically open and on the page you can click ‘install’. The packages are now automatically included in Max.

The last two packages you have to manually include. Click the link and download the files. Now locate Max in you directory → go to the folder ‘packages’ and unzipp the files in this folder. This is also how you can manually add the other packages if necessary.

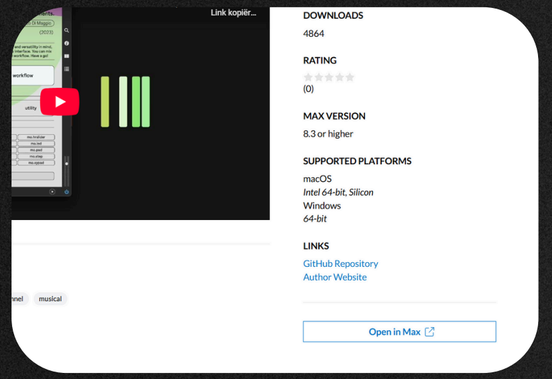
If done correctly you ‘packages’ folder should now lcontaint all five packages.

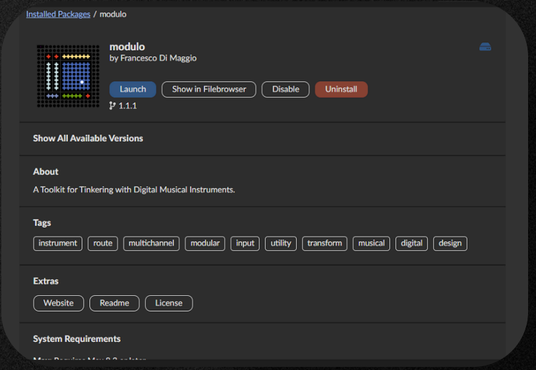
Cycling 74

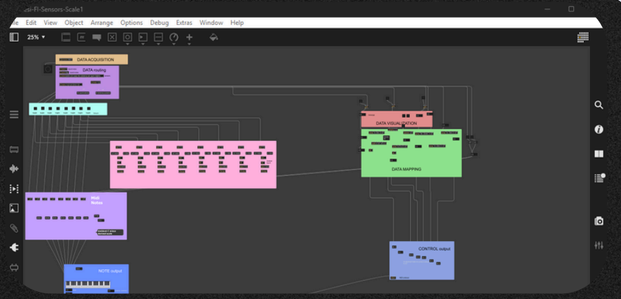
https://www.arduino.cc/en/software/

Libraries to install

Modulo [https://cycling74.com/packages/modulo]
Odot [https://cycling74.com/packages/odot]
Sadam Library [https://cycling74.com/packages/sadam-library]
FluCoMa [https://www.flucoma.org/]
Oocsi-max [https://github.com/francesco-di-maggio/oocsi-max]



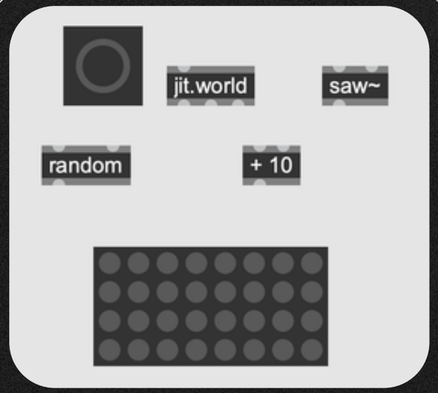




Patches

You make everything in a patch. It is an flexible sketch space to create your interactive software, in our case the mappings for sonic interaction design.

Before we start, first some basic information how Max works to give you a better understanding. Max offers you an environment where you can create your own interactive software for music., but also has also tools to work with videos and graphics. For this application we will use Max as the main mapping environment for sonic interaction design. Max is build up of the following things.

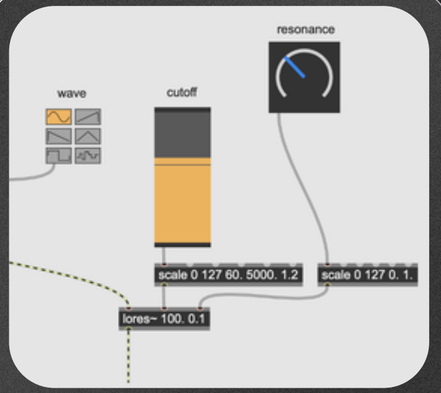


Objects

Everything you create in Max is made from an object that you define. Each object does something, for example represent hardware or generate sound waves. The most used objects have shortcuts, but eveything can be made manually made from an object.

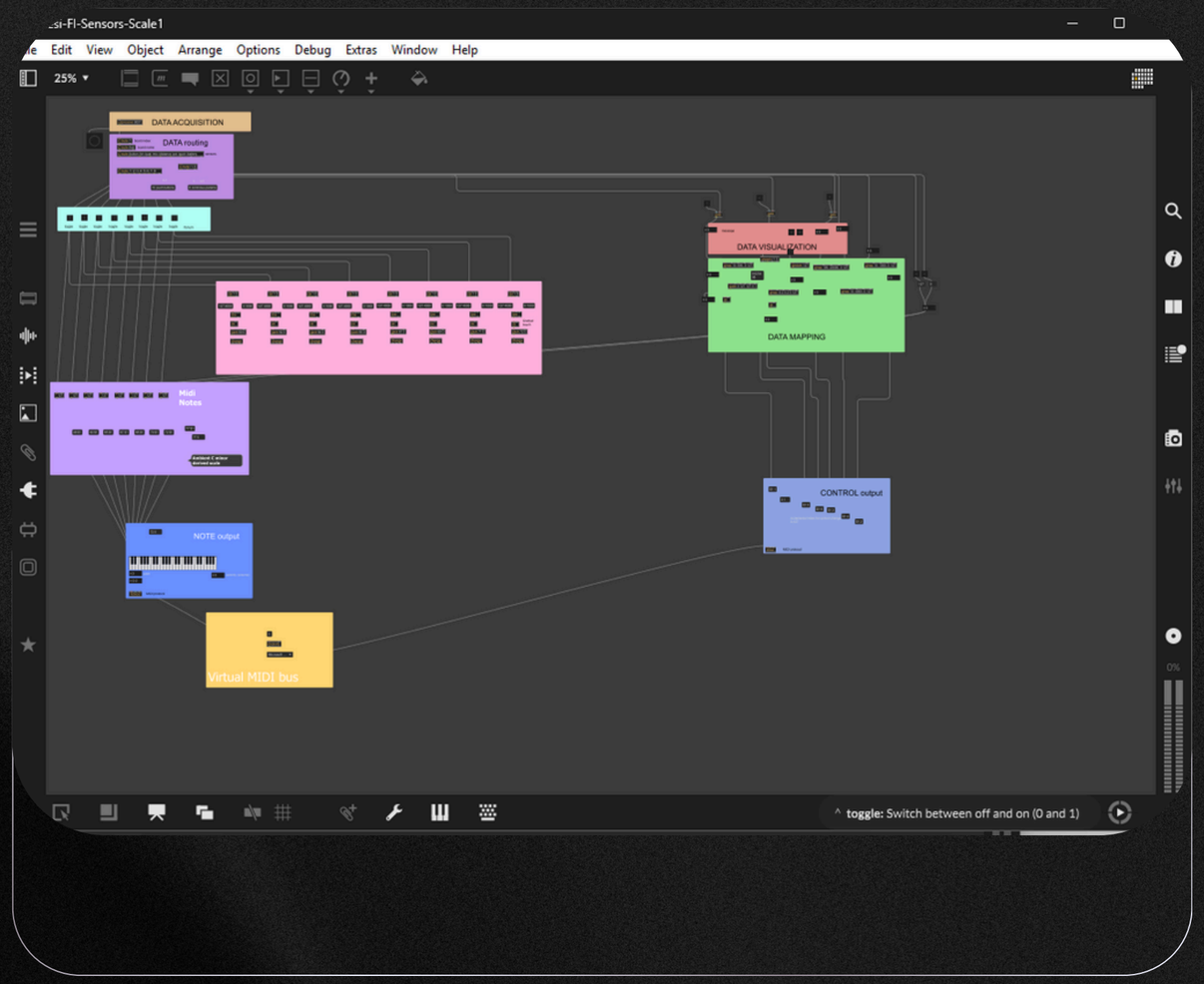
Patchcords

Patchcords connect one object to another. This connection lets object share their output with connected objects.



UI objects

A special category of objects are UI objects like sliders, dials or buttons to procide control over values or display results.



Now you understand the basics on how Max is build up, you can open your first patcher. To get you started we created a basic patch. Here we present you with some relevant basics and examples of Max and a fully worked out set up for the TESI to receive, process and map input from the sensors and convert and send it to DAW software for sound creation. Showcasing the functionalities and enabling you to start explore with TESI.

- *Download Link*

Comments in the patcher will guide you through the functionalities , working principles and possibilities. Also watch this (...) tutorial where we guide you through this patcher to get you started.

However, Max has so much to offer which is impossible to cover in this manual. Go to the Max website or watch youtube videos to explore and learn all the possibilities Max has.

- https://cycling74.com/learn
- https://docs.cycling74.com/userguide/
- Youtube videos

After some initial exploration and experimentation with Max to get yourself familiar, see the next step and try to set up and connect your prefered DAW software to make use of the full potential of the TESI system.

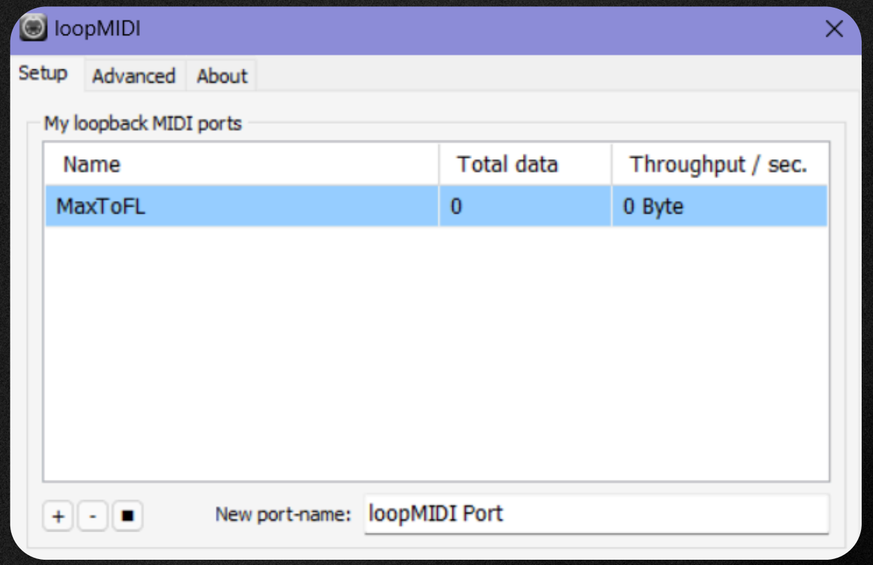
LoopMIDI

What is LoopMIDI

LoopMIDI is an open source software that creates Virtual channels to send MIDI data from Max 9 to your DAW. This is needed for Window Machines, since they do not have a native virtual MIDI port driver. You can install it for free at: https://www.tobias-erichsen.de/software/loopmidi.html

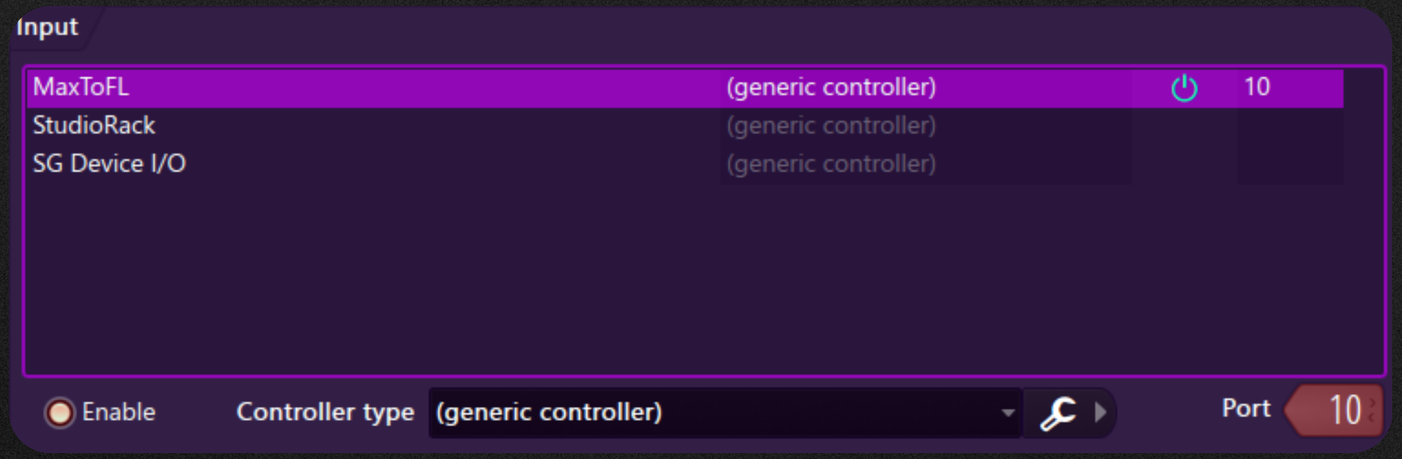
Setup

Once downloaded, open the software and create a new MIDI port by clicking the + button. Now in Max, you should be able to connect to this port in midiinfo.



LoopMIDI Advice

When you connect in Max, check to see if the Total Data is increasing when you send data. If it is not increasing, make sure your note out is also selected to the correct port. If not working, close Max and LoopMIDI, and open LoopMIDI first.



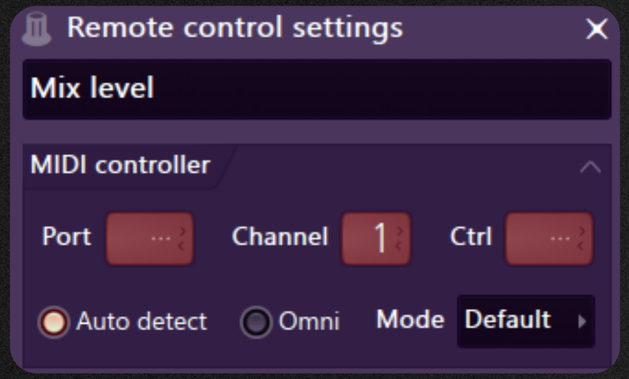
Digital Audio Workstation

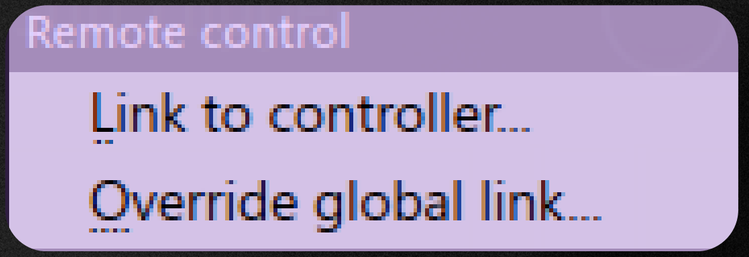
Select a DAW to work with (FL Studio was used in this example but these traits are universal). Enter your MIDI Settings, and within the Input section, enable the LoopMIDI Port you created.

Now your Max Patch is linked to your DAW. NoteOut will automatically begin playing notes on whatever VST(Virtual Studio Technology) you have selected.

To use your CtlOut data, right click on any sliders or knobs within your DAW. Click on Link to controller.

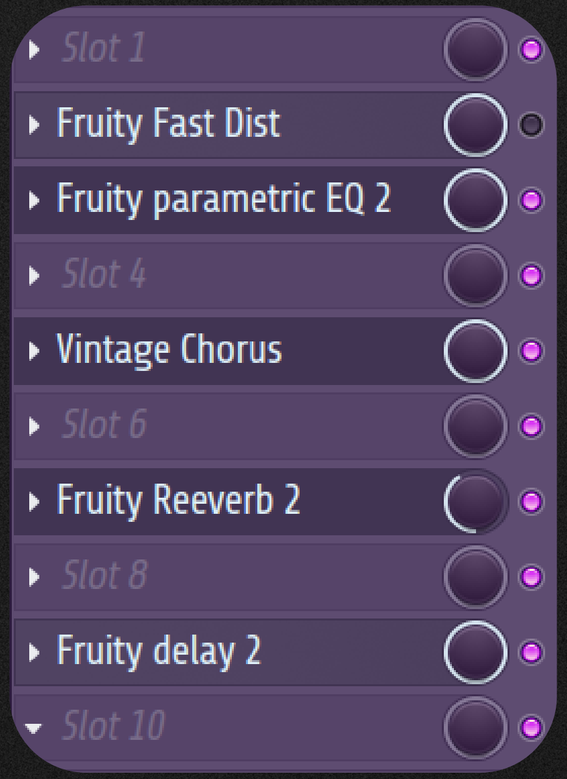
Now, in the control settings, you can either manually select the Channel and Ctrl of your data. You can also select Auto detect, which allows you to interact with a sensor on your TESI, and it will automatically link that sensor to the value of this Control.





MIDI Notes

When using sensor values, that can have wide ranges of values. Locking these values to notes of a scale, or using an arpeggiator can help the output not feel out of control, and more stablized.



Mapping TESI to VST

Use of Effects

Effects, like Reverb, Delay and Distortion can be fun effects to pair with sensors with Analog Data. You can toggle whether the effect is switched on, the strength of the effect as well as the parameters of that specific effect, all with the data from your TESI sensors.

Good luck!