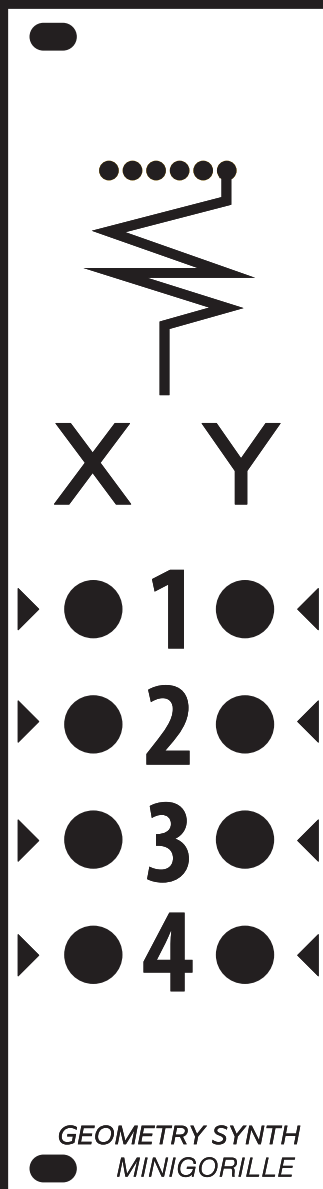


GEOMETRY SYNTHESIZEUR

MANUAL V1.1
LAST EDITED
20141109



contact@minigorille.com
#geometrysynth

The **Geometry synth** is built around an Atmega328 chip, and **communicate to the computer via a Bluetooth module**, the Bluefruit EZ link from ADAFRUIT.

The main idea behind the module is to provide an hardware that can talk to Processing, a code based environnement to create generative geometry patch that react to the incoming CV's.

The module as 6 CV in and 2 GATE in .

Every module as been hand built and hand tested in Portland, OR USA.

⚠ Please **do not feed the Geometry Synth with voltage above 10v** as you may destroy it !

👁 On some computer you may run into some issue when trying to pair the bluetooth module, be patient and try again and make sure there is not too much interference/obstacle between your computer and the module.

✓ Make sure that the computer that you'll be pairing with has **Bluetooth v2.1 or greater installed**. Nearly every Mac computer/laptop, Windows laptop has this already. If not, buy a USB Bluetooth module but check that it is V2.1 compatible and not just V2.0

⚠ Note that even if Processing can run on Linux, the GeometrySynth can only be used with Windows & MacOS this is due to the Bluetooth serial link module.

👁 **There are two LEDs, one red and one blue :**

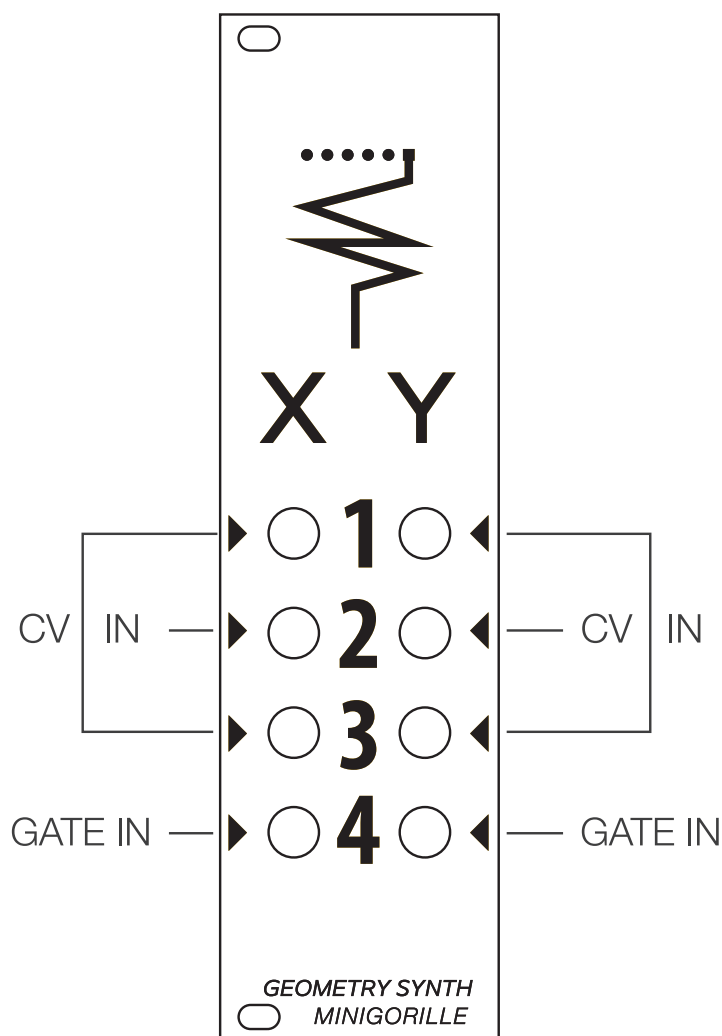
The **red Connection LED** is used to indicate the **state of the bluetooth connection/link**. If it is slow-blinking, there is no pairing for the module. If it is fast blinking, it's paired to a computer and the computer has opened a connection and is sending/receiving data.

The **blue LED** is for monitoring the data received & sent.

Knowing how to code in Processing is not required to get started as the GeometrySynth come with sketch however it's strongly recommended as it's fun and powerful, also Processing language is the same as Arduino.

The GeometrySynth is an experimental platform and when you create new patch you may crash it, it's ok force quit the program and relaunch it.

👁 Please note that if you instal an oder ArduinoBased module such as the Ardcore or a Dev Arduino Board you may have to re-launch the setup.pde Processing sketch as the communication port may have change.



ⓘ The GeoSynth has the Arduino Firmata burned on it's atmega328. This library implements the Firmata protocol for communicating with software on the host computer. For advanced user, it allows you to write custom firmware without having to create your own protocol and objects for the programming environment that you are using, and you can use the Bluetooth serial module to upload your modified Firmata.

✓ The GeoSynth run on +12v // when you plug the module look for the -12v mark on the board, double check Polarity

For video example of the GeometrySynth check :

<https://vimeo.com/98840127>
<https://vimeo.com/100014470>
<https://vimeo.com/110746216>

For information about processing :
<https://processing.org/>

To download Processing sketch :
<http://www.openprocessing.org/>

A very good book with all the basics of Processing :
Getting Started with Processing
 A Quick, Hands-on Introduction
 By Casey Reas, Ben Fry Print ISBN:978-1-4493-7980-3

SETTING UP THE MODULE

When you power the module, you should see a red LED blinking.

Go to http://www.minigorille.com/geometry_synth.html and download GEOMETRY_SYNTH package.



GEOMETRY_SYNTH.ZIP

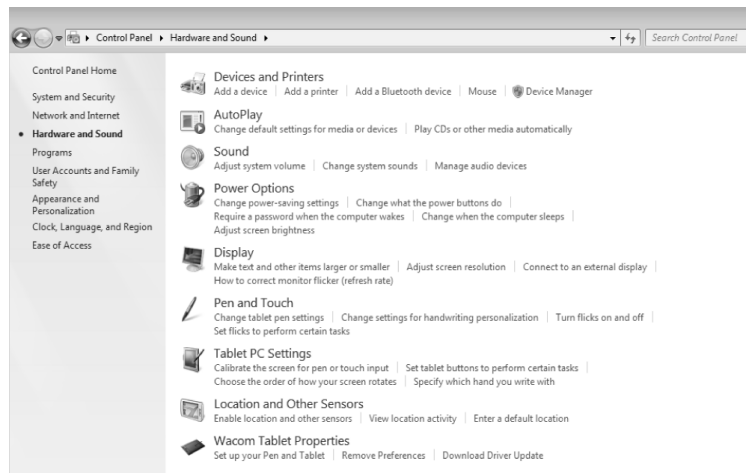
GEOMETRY_SYNTH/SKETCHES
Setup, template and geometry sketch.

GEOMETRY_SYNTH/libraries
the content of this folder need to be copy in your Processing librairie folder.

GEOMETRY_SYNTH/GeoSynth_manual.pdf

1 - PAIRING THE MODULE TO THE COMPUTER ON WINDOWS

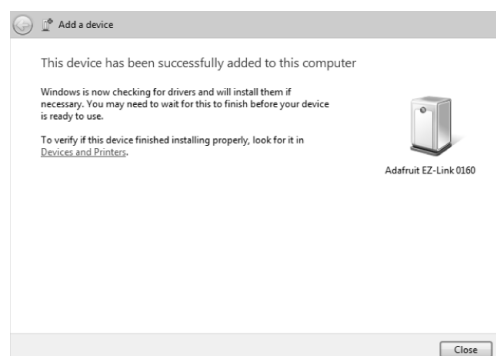
1.0 - open Devices and Printers Control Panel



1.1 - select Add Bluetooth Device, Windows will look for new devices, make sure the module is powered, you should see a new device found with the name **Adafruit EZ - Link** _ _ _ _ _ are the 4 alphanumeric characters written on the label of your Geometry synth box.



1.2 - select the device and click next, the device should now be installed. If ask for a password : 1234



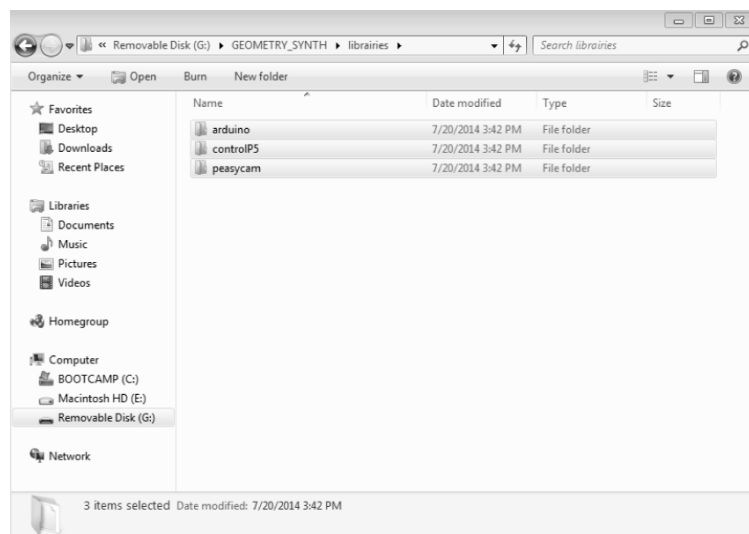
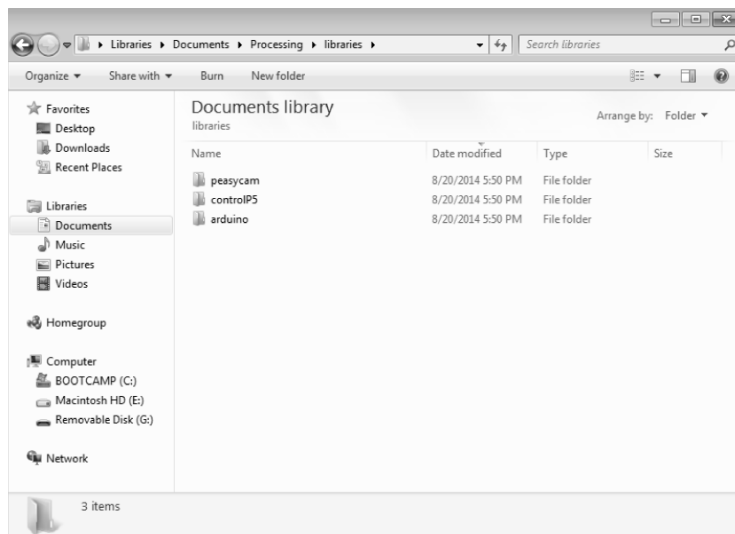
1.3 - Go to <http://www.processing.org> and download the current Processing version available.



1.4 - Once Processing has been installed on your computer, launch Processing.

Then go to your document folder and look for the processing folder, inside it you should have a libraries folder, copy the content of the libraries folder from the GEOMETRY_SYNT package to it.

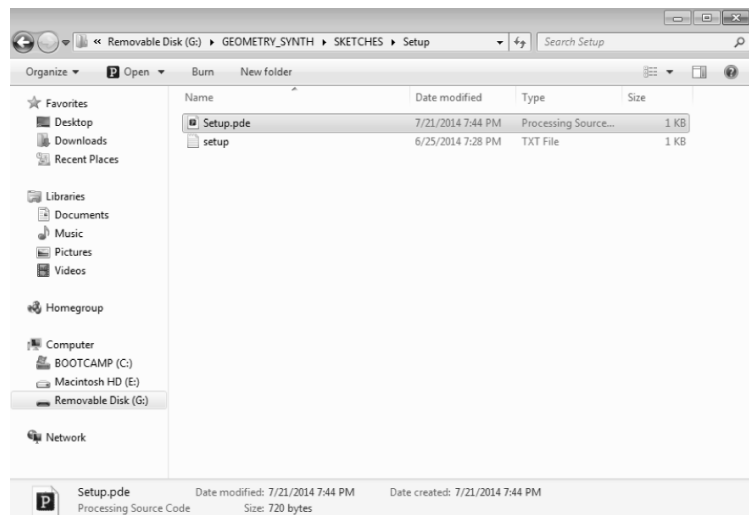
GEOMETRY_SYNT/libraries
the content of this folder needs to be copied into your Processing libraries folder.



1.5 - quit Processing and relaunch it in order to load the library.

1.6 - To finalize the setup of the GEOMETRY SYNT you need to tell Processing on which COM port the module is talking to the computer to do so go in the SKETCHES folder of the GEOMETRY_SYNT package and look for the setup folder, launch setup.pde, you should see the processing setup sketch, run it.

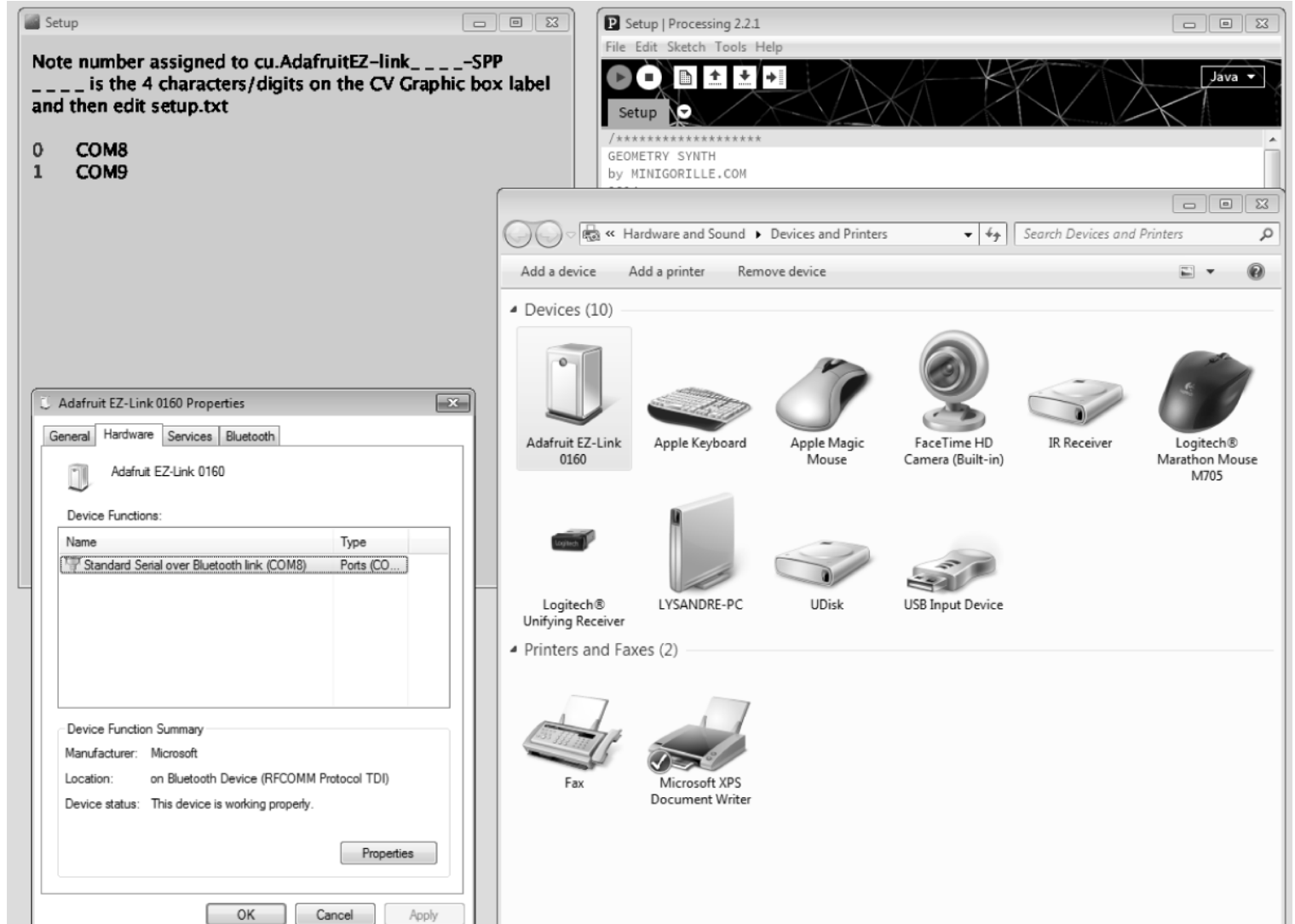
✓ To run a Processing sketch, click on the PLAY button on the TOP LEFT of the editor windows.



1.7 - Open the Device and Printers control Panel and open the EZ-Link device installed in steps 1.1 & 1.2.

Under hardware you can see what is the COM PORT assigned, in this example it is COM8, now look at the Processing sketch and note the blue number assigned to the COM port. In this example it's 0.

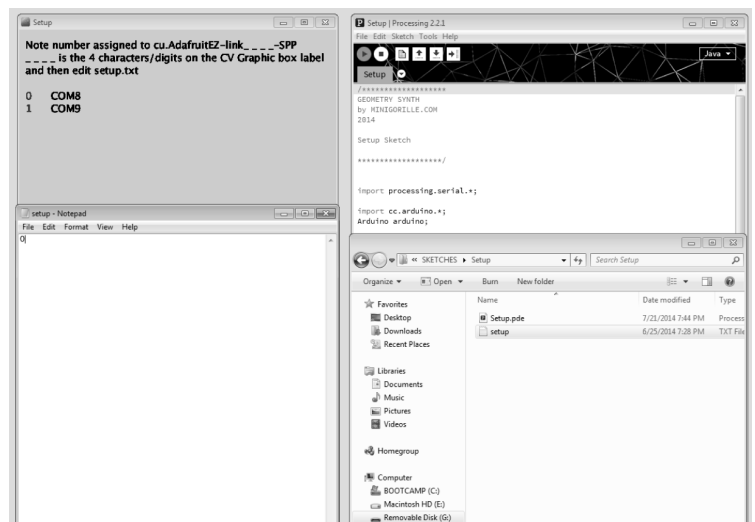
THROW BACK : COM (Communication port[1]) is the original, yet still common, name of the serial port interface on IBM PC-compatible computers. It might refer not only to physical ports, but also to virtual ports, such as ports created by Bluetooth or USB-to-serial adapters.



1.8 - you can now look for the setup.txt file in the setup folder and edit it with note pad, replace the number 0 by the blue number found in step 7 and save.

Please note that you will need to edit the setup.txt for each sketch coming in the GEOMETRY_SYNTH package as well as for every new sketch you will create in the future.

In the future, if you instal a new device that use COM Port such as an Arduino, Ardcore from Snazzy or Euroduino from Circuit Abbey, you may have to go back to step 7 and look at the new port that may have been assigned to the EZ-Link and run setup.pde and edit setup.txt



1.9 - You should now be all setup to run your first geometry sketch !

Go to the sketch folder from the GEOMETRY_SYNTH package and run 001.

✓ When you launch the demo scripts it activate Full-Screen mode, you may notice that it will take a seconde or two before something happen, this is normal and due to the fact that the computer is contacting the module via Bluetooth and opening the communication.

👁 Dont forget to edit the setup.txt before you launch the sketch ! If you forget it will crash the sketch, no big deal just force quit and edit the setup.txt with the correct port and relaunch the sketch !

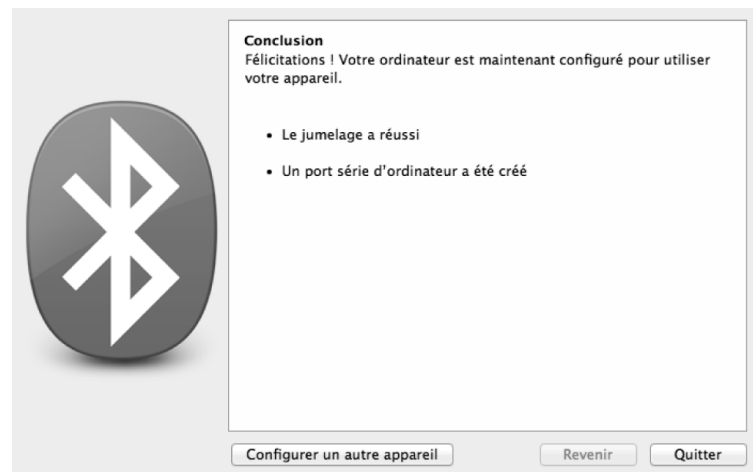
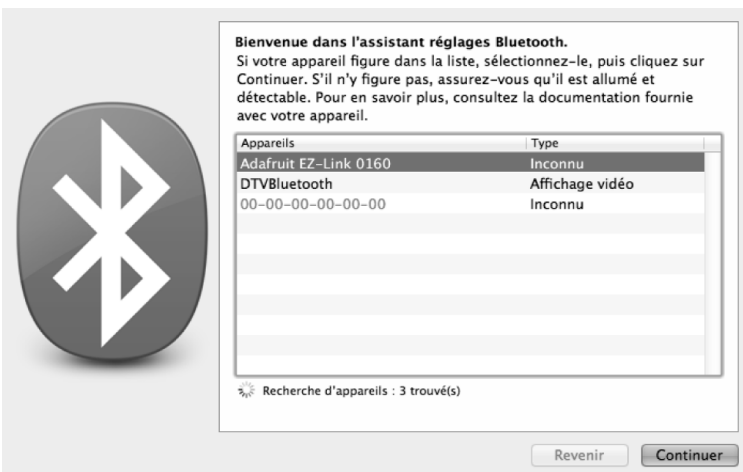
2 - PAIRING THE MODULE TO THE COMPUTER ON MacOS X

You are lucky if you have OS X as it is way easier to install the module on it !

2.0 - turn ON Bluetooth if it's OFF in the finder menu Right corner next to WIFI.

2.1 - open Bluetooth Preference panel

2.2 - It will immediately start scanning for the device, you'll see the BT address first, it will then turn into the EZ-Link name : Adafruit EZ - Link _____. _____ are the 4 alphanumeric characters written on the label of your Geometry synth box. Select it and click next.



2.3 - Go to the page 2 of this manual and follow the instruction steps **1.3 / 1.4 / 1.5 / 1.6** (download and install Processing + install librairies)

2.4 - skip step **1.7** and go to step **1.8** you dont need to worry about COM Port when you launch the setup.pde sketch the Adafruit EZ - Link _____ will appear with a number on the left, note this number and edit all the setup.txt as describe on step **1.8**.

2.5 - Launch a sketch from the example provided.

👁 Dont forget to edit the setup.txt before you launch the sketch ! If you forget it will crash the sketch, no big deal just force quit and edit the setup.txt with the correct port and relaunch the sketch !

👁 In general, troubleshooting is the identification of diagnosis of "trouble" in the management flow of a corporation or a system caused by a failure of some kind. The problem is initially described as symptoms of malfunction, and troubleshooting is the process of determining and remedying the causes of these symptoms.

NO POWER // RED LED OFF

- Check the power/polarity connection on the back of the module.

COMPUTER DONT FIND THE BLUE-TOOTH MODULE

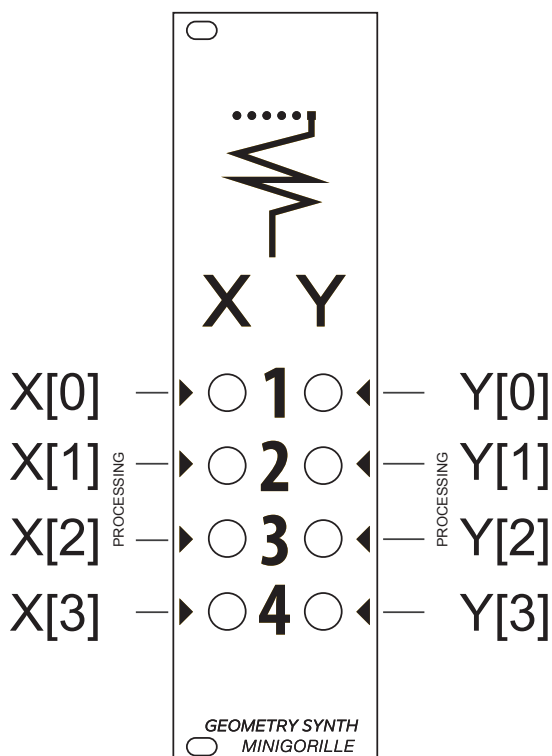
- Make sure Bluetooth V2.1 or greater is installed on your computer (Check intro for more info).
- Try to move your laptop, get closer to the module to help with the pairing process, on some laptop the bluetooth reception can be weak.
- Restart your Eurorack.
- Be patient and try a couple of times, the Bluetooth pairing process can sometime be anyoing on older computer.

FREEZE WHEN RUN A SKETCH

- Double check that you have correctly edited the setup.txt file located in the sketch folder with the correct port number find in step 1.8 or 2.4
- Re-run the setup.pde sketch to make sure the port has not change, especially if you have recently installed a new device on your Windows computer, the COM port will have change, note the new port and edit all the setup.txt text files.
- Force quit the sketch, wait a few seconde and relaunch it, sometimes when you close a sketch and re-launch right away a different one the old connection between the module and the computer has not been close properly and when the computer try to establish a new connection it failed to do so and cause the Processing sketch to crash.
- Try with one of the provided demo sketch as an error in your code may cause the Processing sketch to crash.

WHY X1 IS X[0] IN PROCESSING CODE ?

Because we have used Array to store the incoming value we have shifted the index as an Array start at 0. We will in the future update the front panel to avoid confu-sion.



```

/*****
GEOMETRY SYNTH
by MINIGORILLE.COM
2014

```

Template sketch

```

CV INPUT (0 to 5v)
X1 X2 X3
Y1 Y2 Y3

```

```

GATE INPUT
X4
Y4
*****/

```

```

import processing.serial.*;
import cc.arduino.*;
Arduino arduino;

```

```

int[ ] sPort;

```

//VARIABLES

//ASSIGN PIN TO X1 X2 X3 X4

```

int[ ]X = {0, 2, 4, 2};

```

//ASSIGN PIN TO Y1 Y2 Y3 Y4

```

int[ ]Y= {1, 3, 5, 3};

```

//valX FOR EACH CV X INPUT

```

float[ ]valX = new float[3];

```

//valY FOR EACH CV Y INPUT

```

float[ ]valY = new float[3];

```

//valXY FOR EACH SET OF CV XY INPUT

```

float[ ][ ]valXY = {valX, valY};

```

//FULL SREEN AT STARTUP

```

boolean sketchFullScreen() {
    return true;
}

```

```

void setup() {

```

// READ THE SERIAL PORT IN THE TEXT FILE setup.txt

```

String[ ] serialP = loadStrings("setup.txt");

```

```

sPort = int(split(serialP[0],','));

```

// BACKGROUND COLOR

```

background(0);
smooth();

```

//WINDOWS SIZE

```

size(displayWidth, displayHeight);

```

//SERIAL PORT NUMBER

```

arduino = new Arduino(this, Arduino.list()[sPort[0]], 57600);

```

//DECLARE X4 & Y4 PINS AS INPUT

```

arduino.pinMode(X[3], Arduino.INPUT);

```

```

arduino.pinMode(Y[3], Arduino.INPUT);

```

```

}

```

```

void draw() {

```

// READ THE INCOMING CV AND REMAP THEM

```

for (int i=0; i<3; i++) {

```

```

    valX[i] = map(arduino.analogRead(X[i]), 0, 1023, 0, width);

```

```

    valY[i] = map(arduino.analogRead(Y[i]), 0, 1023, 0, height);

```

```

}

```

```

}

```

COMMENTS HEADER

Use the comments to explain your sketch, what are the control, variables it's also were you can put credits if you use someone else sketch or part of it.

FULL SCREEN

By default the sketch will go FULLSCREEN you can remove this lines of codes if you dont want FULLSCREEN

COLOR, SMOOTH & WINDOWS SIZE

By default the sketch as a black baground and smooth enabled, if you remove the FULLSCREEN code you can specify the size of your windows, often if you try experimental code with heavy computing you may want to use a small resolution.

REMAP VALUES

By default the sketch will remap all the incoming values from 0 to width and 0 to height (pixels) but you can map each incoming valX[i] & valY[i] individually.

/*****

GEOMETRY SYNTH
by MINIGORILLE.COM
2014



IT IS NOT RECOMENDED TO CHANGE
ANY OF THE LINES OF CODES HIGH-
LIGHTED ON THIS PAGE.

Template sketch

CV INPUT (0 to 5v)

X1 X2 X3

Y1 Y2 Y3

GATE INPUT

X4

Y4

*****/

```
import processing.serial.*;
import cc.arduino.*;
Arduino arduino;
```

```
int[ ] sPort;
```

//VARIABLES

//ASSIGN PIN TO X1 X2 X3 X4

```
int[ ]X = {0, 2, 4, 2};
```

//ASSIGN PIN TO Y1 Y2 Y3 Y4

```
int[ ]Y= {1, 3, 5, 3};
```

//valX FOR EACH CV X INPUT

```
float[ ]valX = new float[3];
```

//valY FOR EACH CV Y INPUT

```
float[ ]valY = new float[3];
```

//valXY FOR EACH SET OF CV XY INPUT

```
float[ ][ ]valXY = {valX, valY};
```

//FULL SREEN AT STARTUP

```
boolean sketchFullScreen() {
    return true;
}
```

```
void setup() {
```

// READ THE SERIAL PORT IN THE TEXT FILE setup.txt

```
String[ ] serialP = loadStrings("setup.txt");
```

```
sPort = int(split(serialP[0],','));
```

// BACKGROUND COLOR

```
background(0);
```

```
smooth();
```

//WINDOWS SIZE

```
size(displayWidth, displayHeight);
```

//SERIAL PORT NUMBER

```
arduino = new Arduino(this, Arduino.list()[sPort[0]], 57600);
```

//DECLARE X4 & Y4 PINS AS INPUT

```
arduino.pinMode(X[3], Arduino.INPUT);
```

```
arduino.pinMode(Y[3], Arduino.INPUT);
```

```
}
```

```
void draw() {
```

// READ THE INCOMING CV AND REMAP THEM

```
for (int i=0; i<3; i++) {
```

```
    valX[i] = map(arduino.analogRead(X[i]), 0, 1023, 0, width);
```

```
    valY[i] = map(arduino.analogRead(Y[i]), 0, 1023, 0, height);
```

```
}
```

```
}
```

Importing required Library such as Processing Serial and Arduino.

Declaring arrays, an array is a collection of variables that are accessed with an index number.

int[] sPort is an array that will contain the Port number.

Here we are reading the setup.txt file, in order to know the port number.

It's easier to have it store in an external file so if the communication port change you dont have to open the sketch but just to edit the setup.txt text file.

Opening the communication

Decalring the Arduino Pin as Input for X3 & Y3 // X4 and Y4 on the module.