lighting installations & artists

HC Gilje: http://hcgilje.com

nonotak: https://www.nonotak.com/_TAKAMI-NAKAMOTO

404.zero: https://vimeo.com/404zero

Romain Tardy: https://romaintardy.com/The-Great-Indecision-Council-2018

Plan

Introduce the physical materials and tools

Let’s light up LED together

physical materials

Light

1. **LED strips**
   - addressable*
   - RGBW*

+ **DMX decoder**
1. DMX USB Pro
2. DMX512

2. **bulbs**
   - dimmerable*
   - non dimmerable

+ **Relay** + **Arduino**

3. spotlight

4. laser light

...

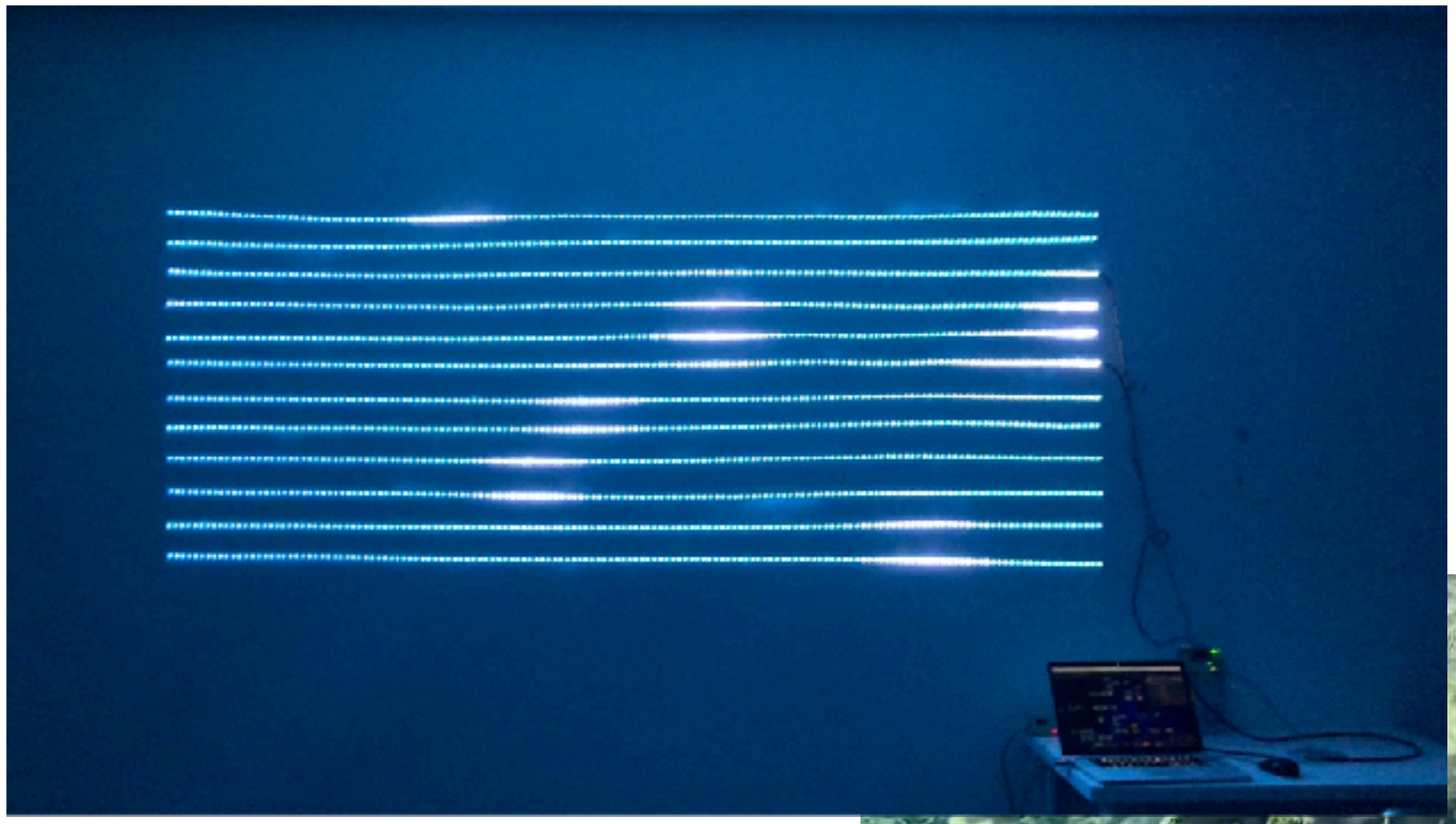
*show it in class

DMX

DMX originated as a way to set the bar for lighting manufacturers to build fixtures that would all be compatible with each other, instead of having individual control stations for each set of lighting. This gave the Audio Visual industry a huge break because it allowed them to control everything from one single source giving them more freedom and flexibility when it came to creating lighting shows.

DMX requires different components to work. First off, you need a source to create DMX. This can be achieved by a computer interface that converts USB to DMX using a DMX software, which will allow you set infinite presets, and give you full control all with the click of a mouse.

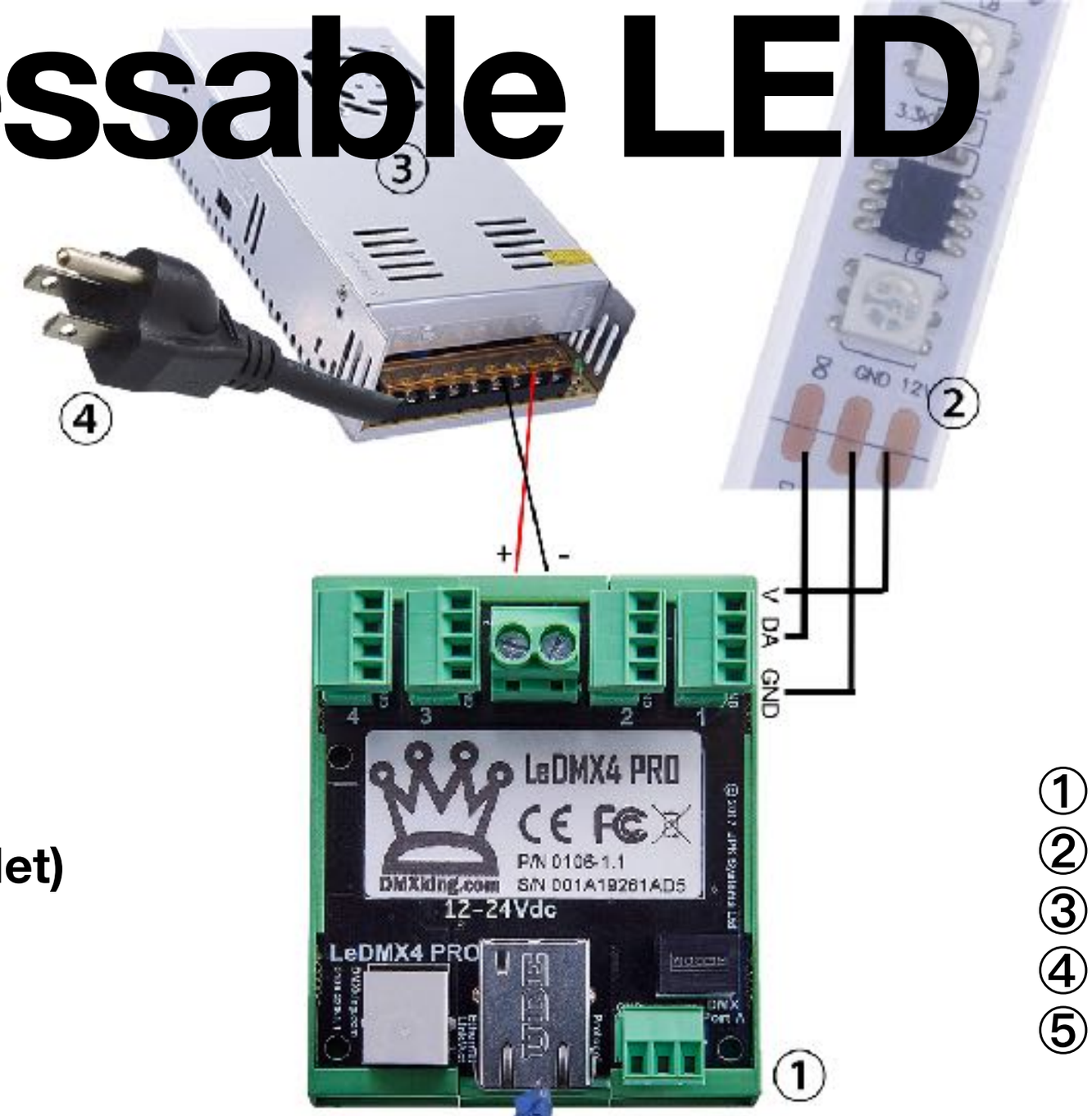
addressable LED



Pixel Mapping

build a LED matrix panel
&
map image/video on it

addressable LED



LED

↑

DMX512(via Art-Net)

↑

TouchDesigner

① **DMX 12v***
② **LED 12v***
③ **power supply 12v***
④ **power cable***
⑤ **network cable**

*click it to buy

RGBW

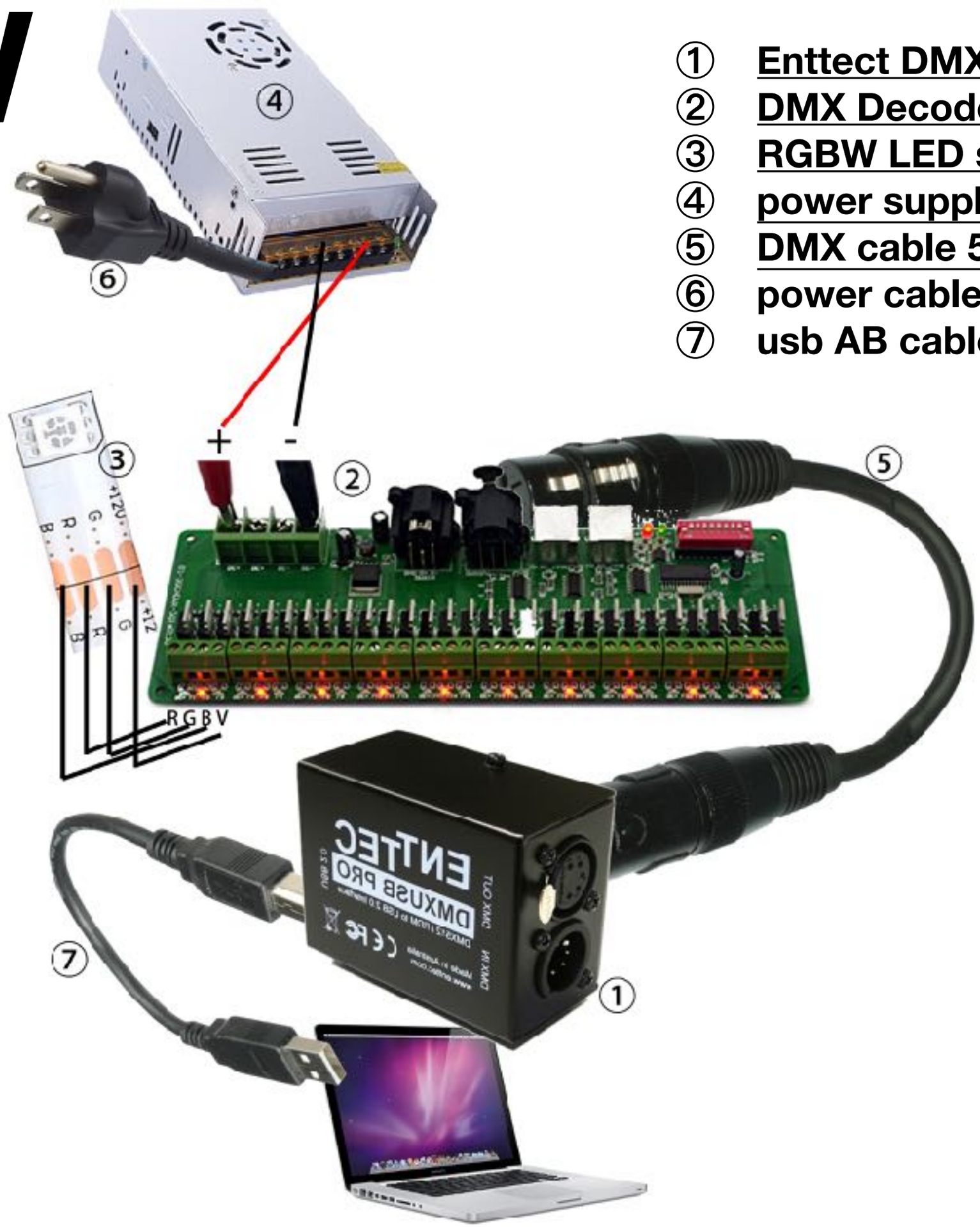
LED

↑

DMX usb Pro

↑

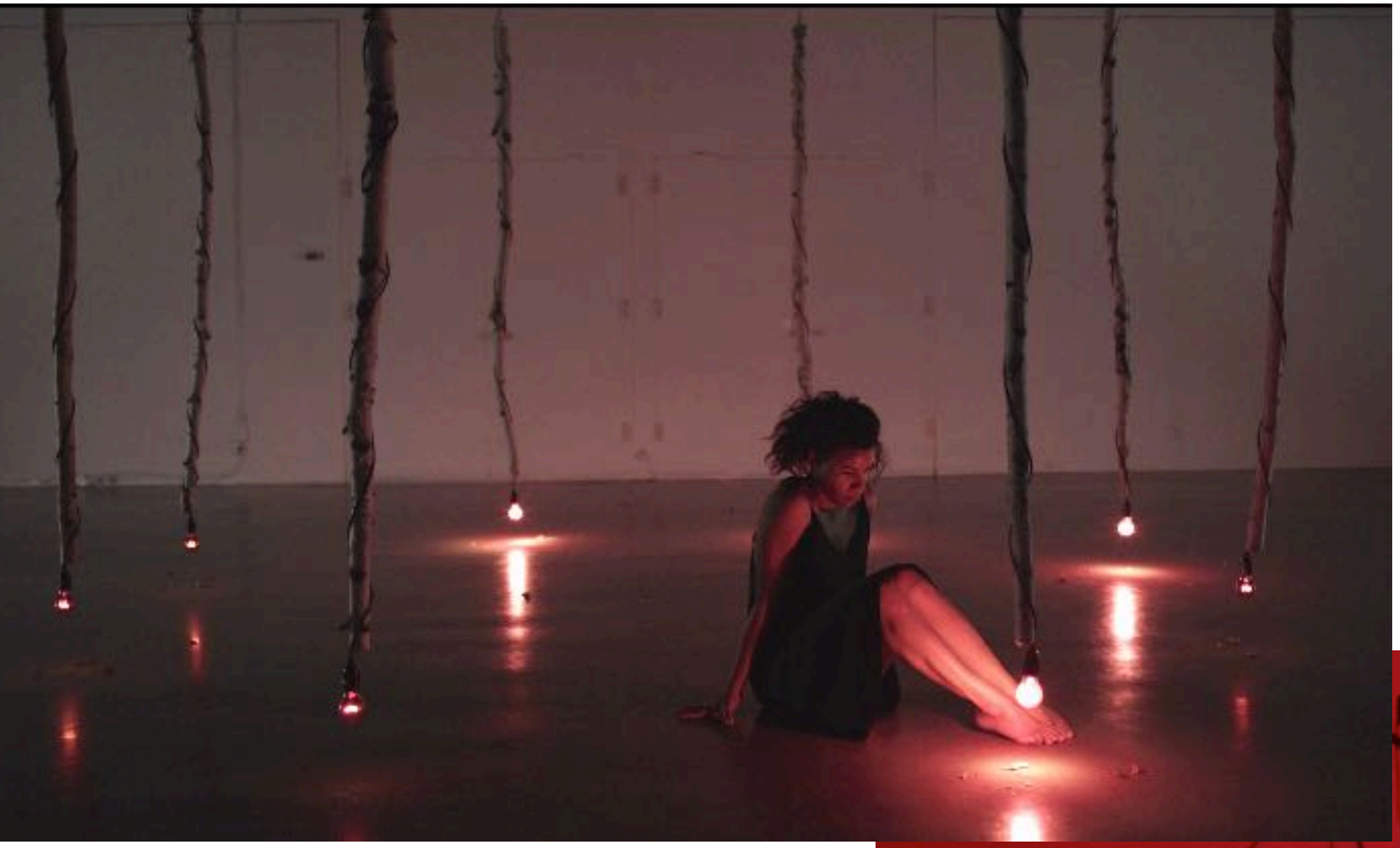
TouchDesigner

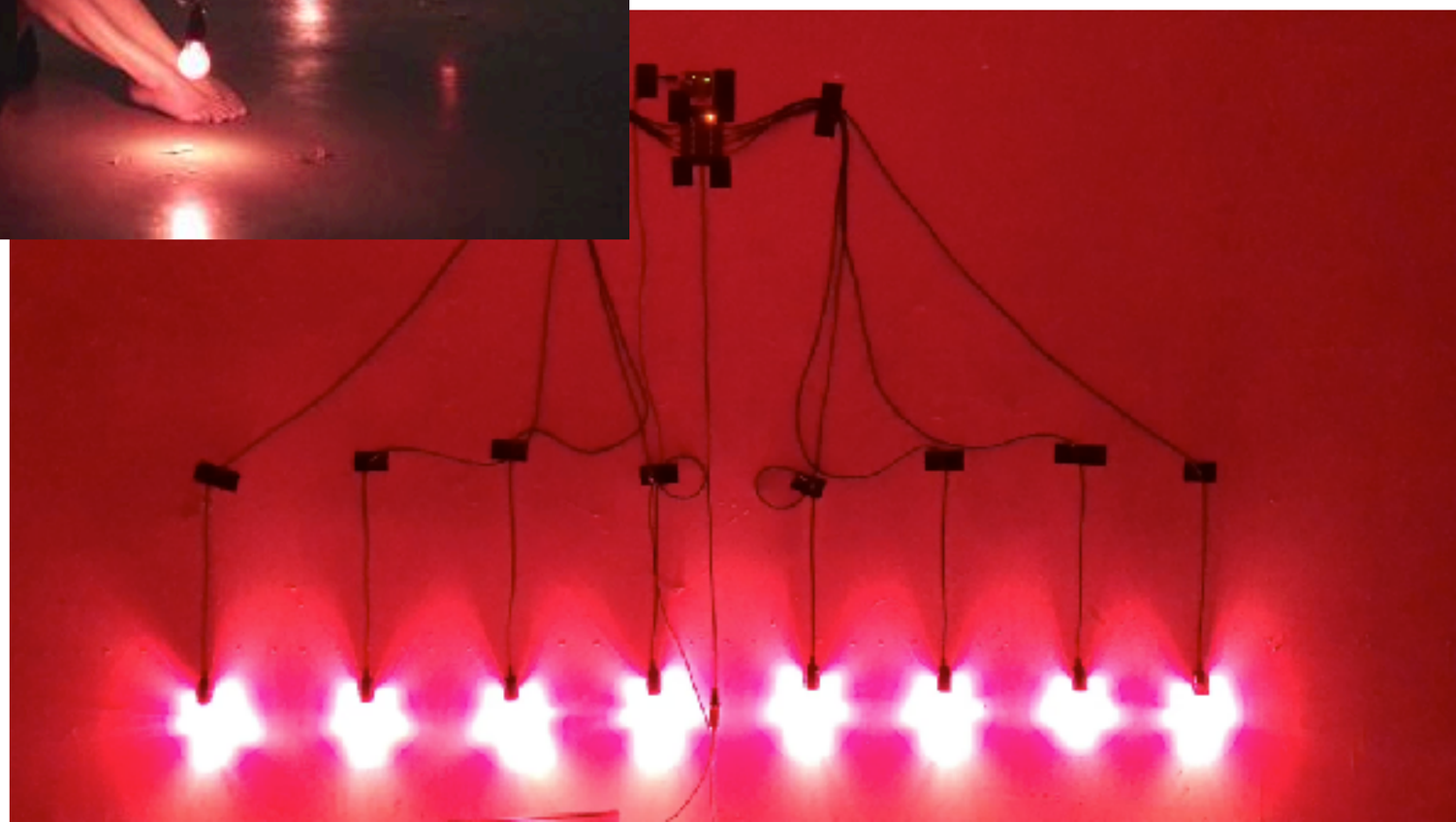


① Enttect DMX PRO*
② DMX Decoder*
③ RGBW LED strips 12v*
④ power supply 12v*
⑤ DMX cable 5 pin*
⑥ power cable
⑦ usb AB cable

*click it to buy

dimmerable bulb





dimmerable bulb

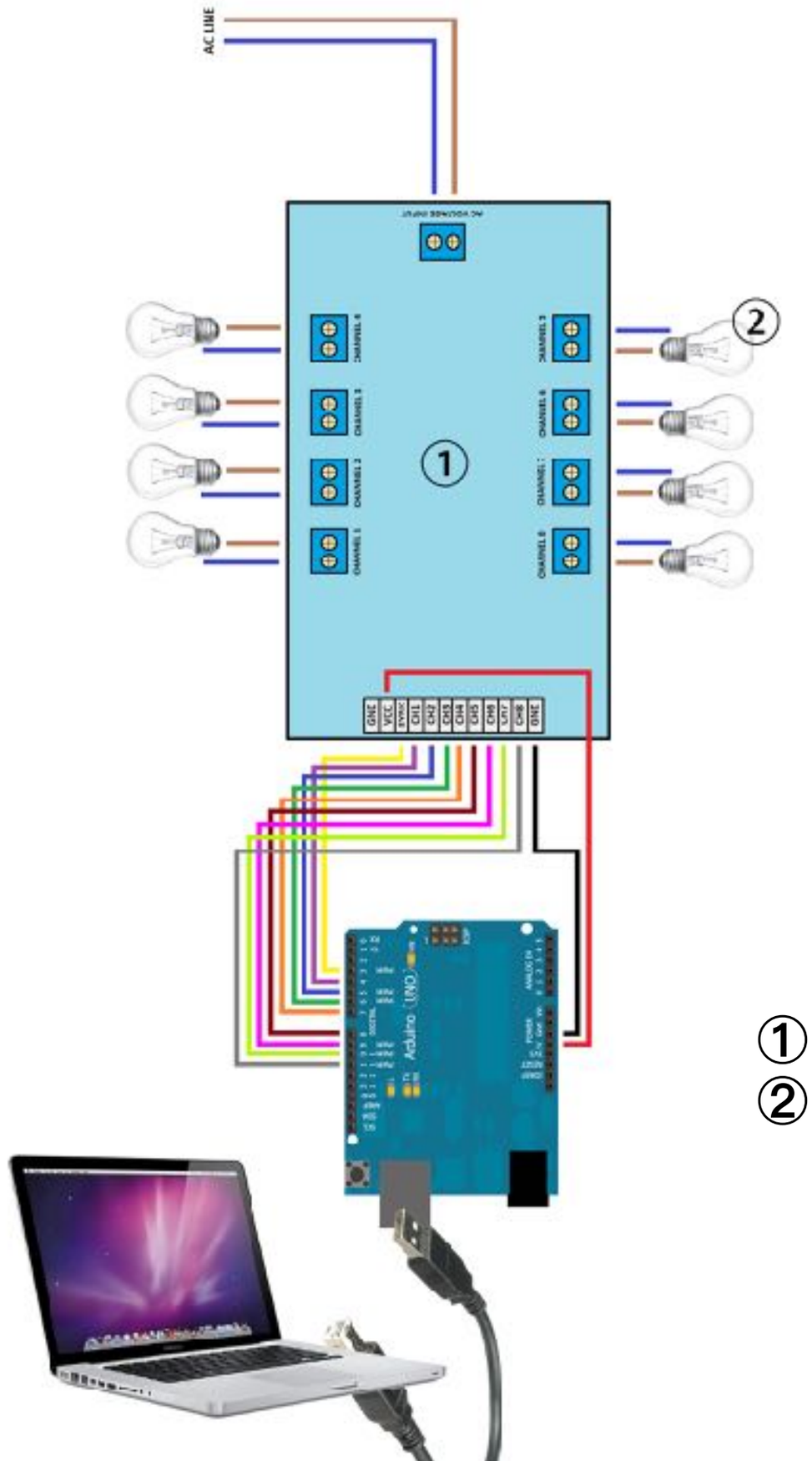
Bulbs

↑

Arduino + Relay

↑

TouchDesigner



① dimmer relay*
② dimmerable bulb*

*click it to buy

audio driven light

Get audio values and visualize in light

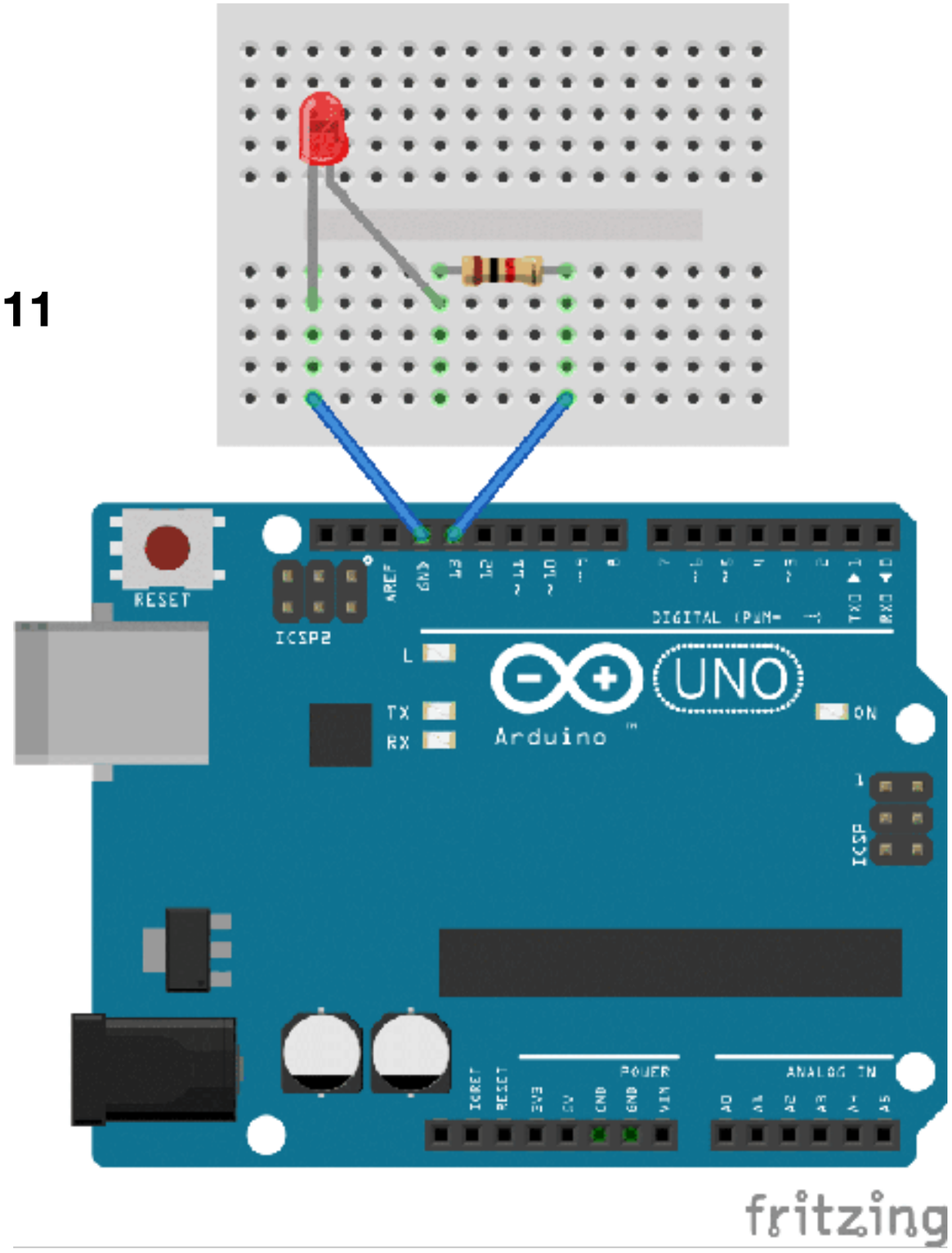
TouchDesigner

TD is a visual programming platform

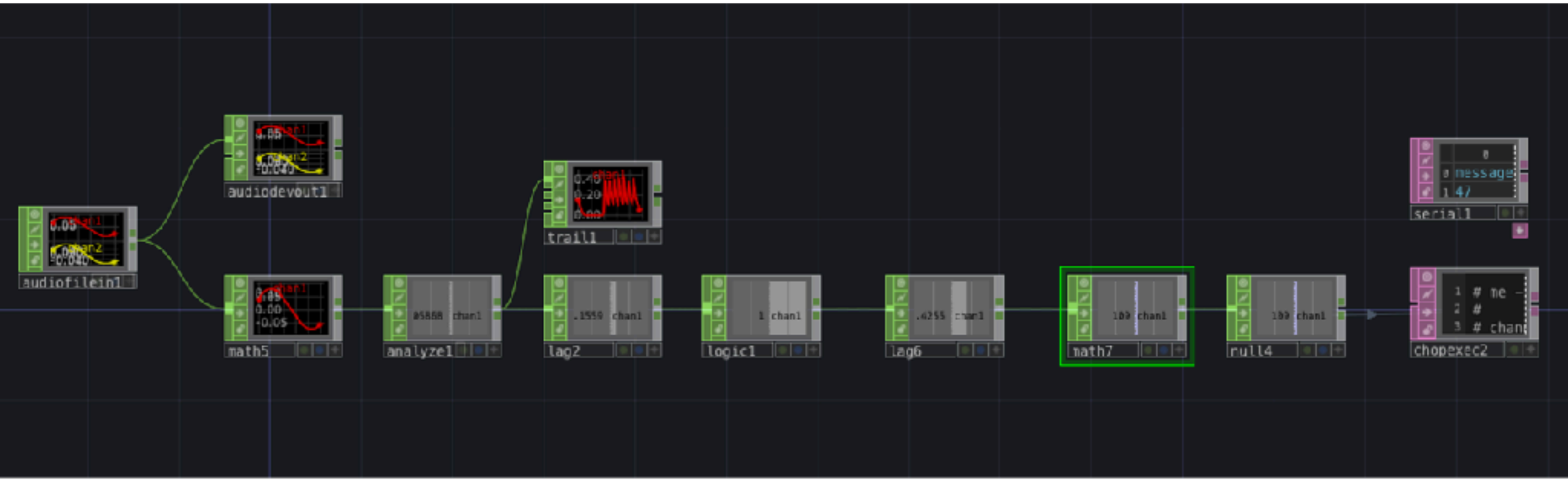
***TD download link: https://www.derivative.ca/099/Downloads/**

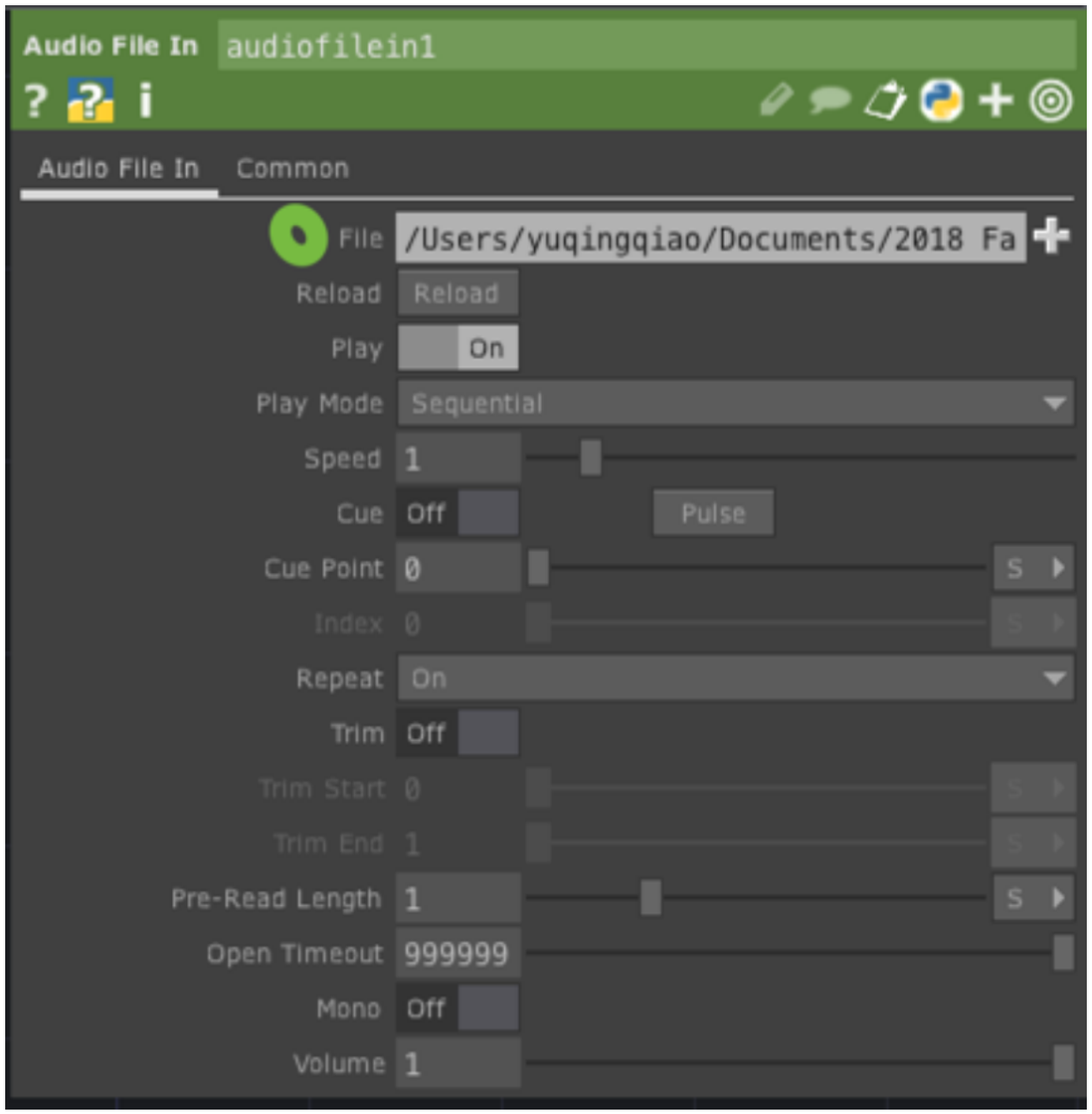
circuit

ledPin = 11

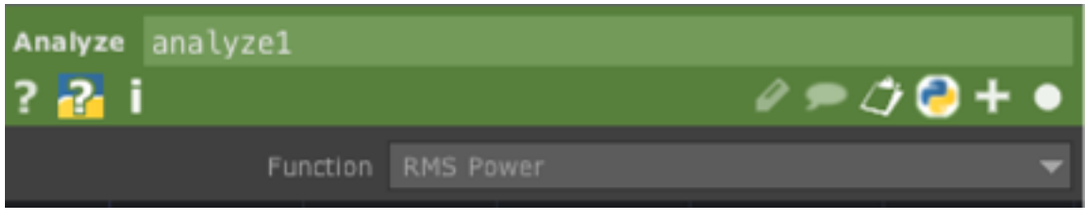


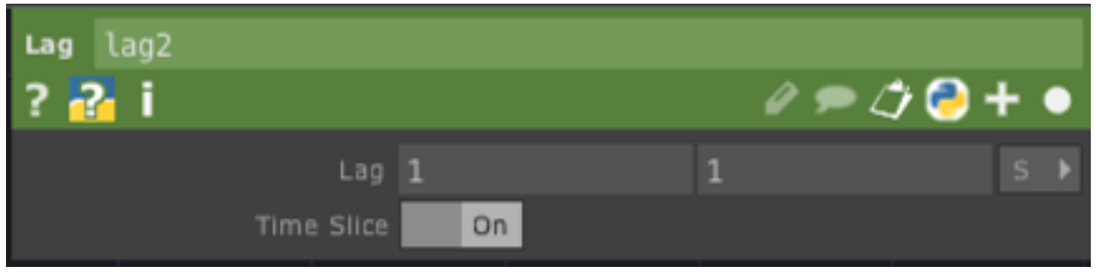
open TD

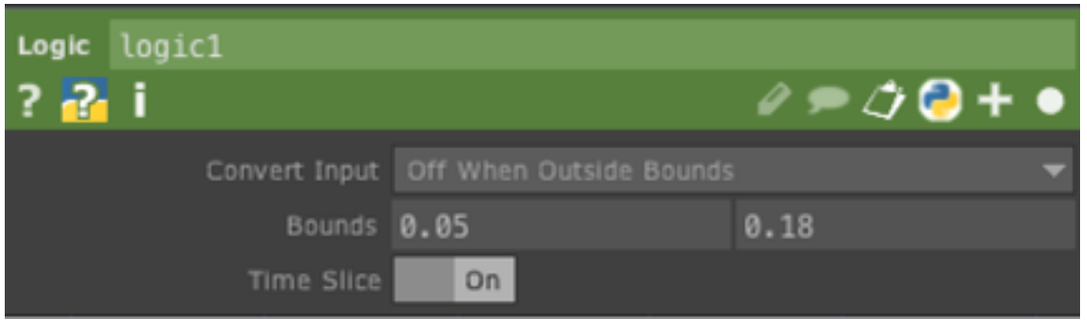

audiofilein1
audiodevout1
math5
trail1
analyze1
lag2
logic1
lag6
math7
null4
serial1
chopexec2

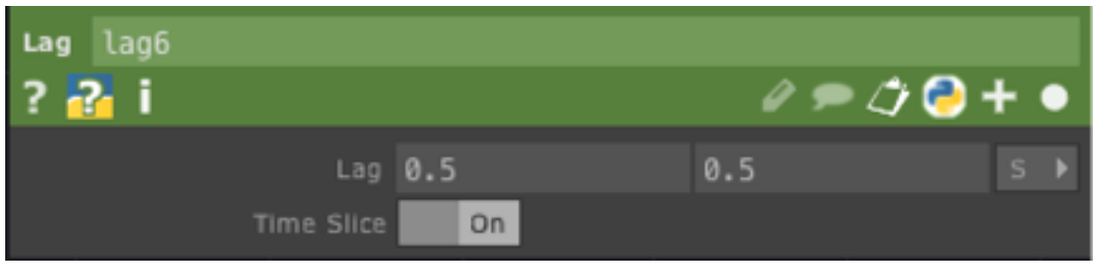

Audio File In
audiofilein1
Audio File In
Common
File
/Users/yuqingqiao/Documents/2018 Fa
Reload
Reload
Play
On
Play Mode
Sequential
Speed
1
Cue
Off
Pulse
Cue Point
0
S
Index
0
S
Repeat
On
Trim
Off
Trim Start
0
S
Trim End
1
S
Pre-Read Length
1
S
Open Timeout
999999
Mono
Off
Volume
1

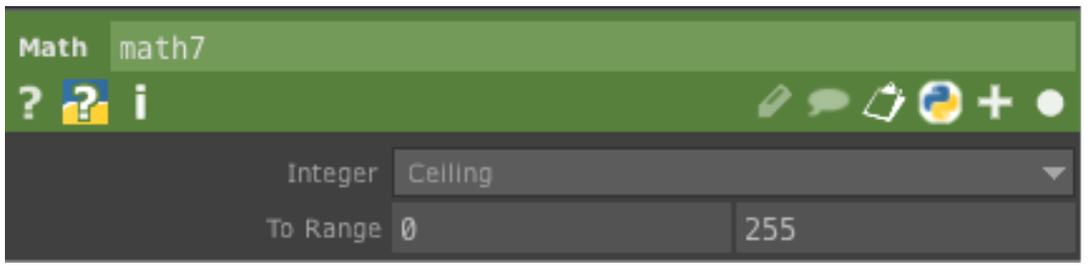

Math
math5
Combine Channels
Add

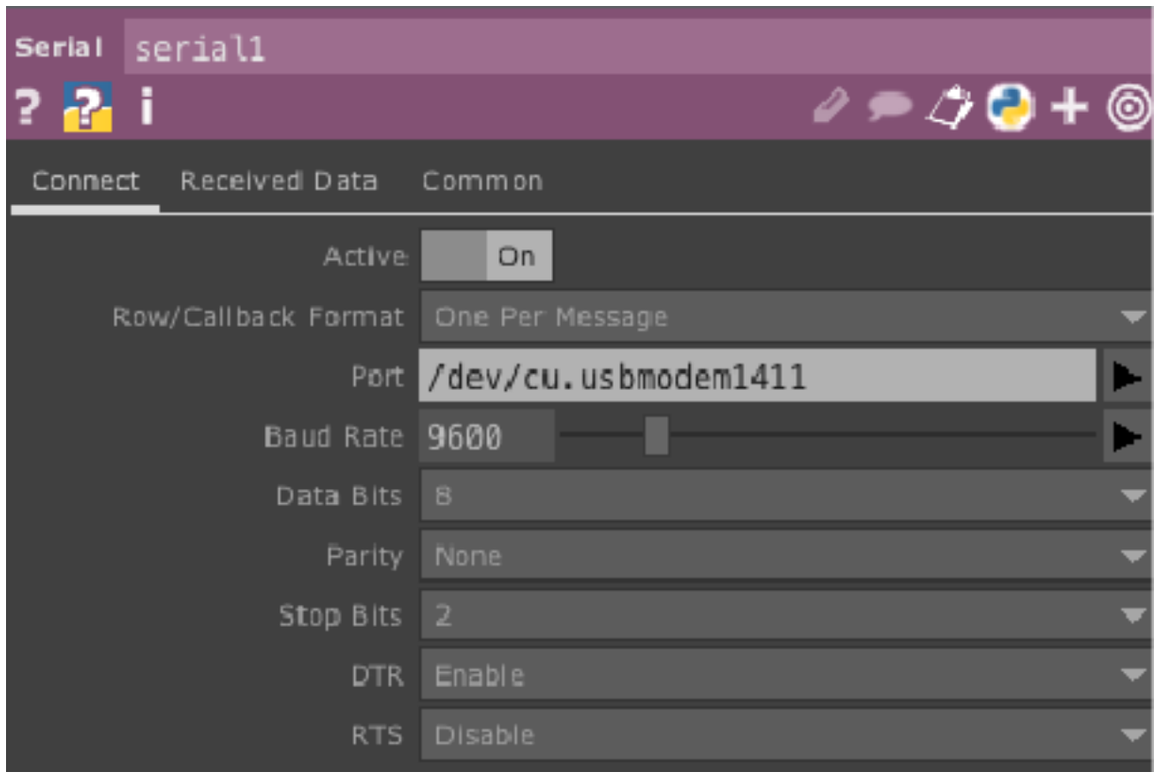

Analyze
analyze1
Function
RMS Power

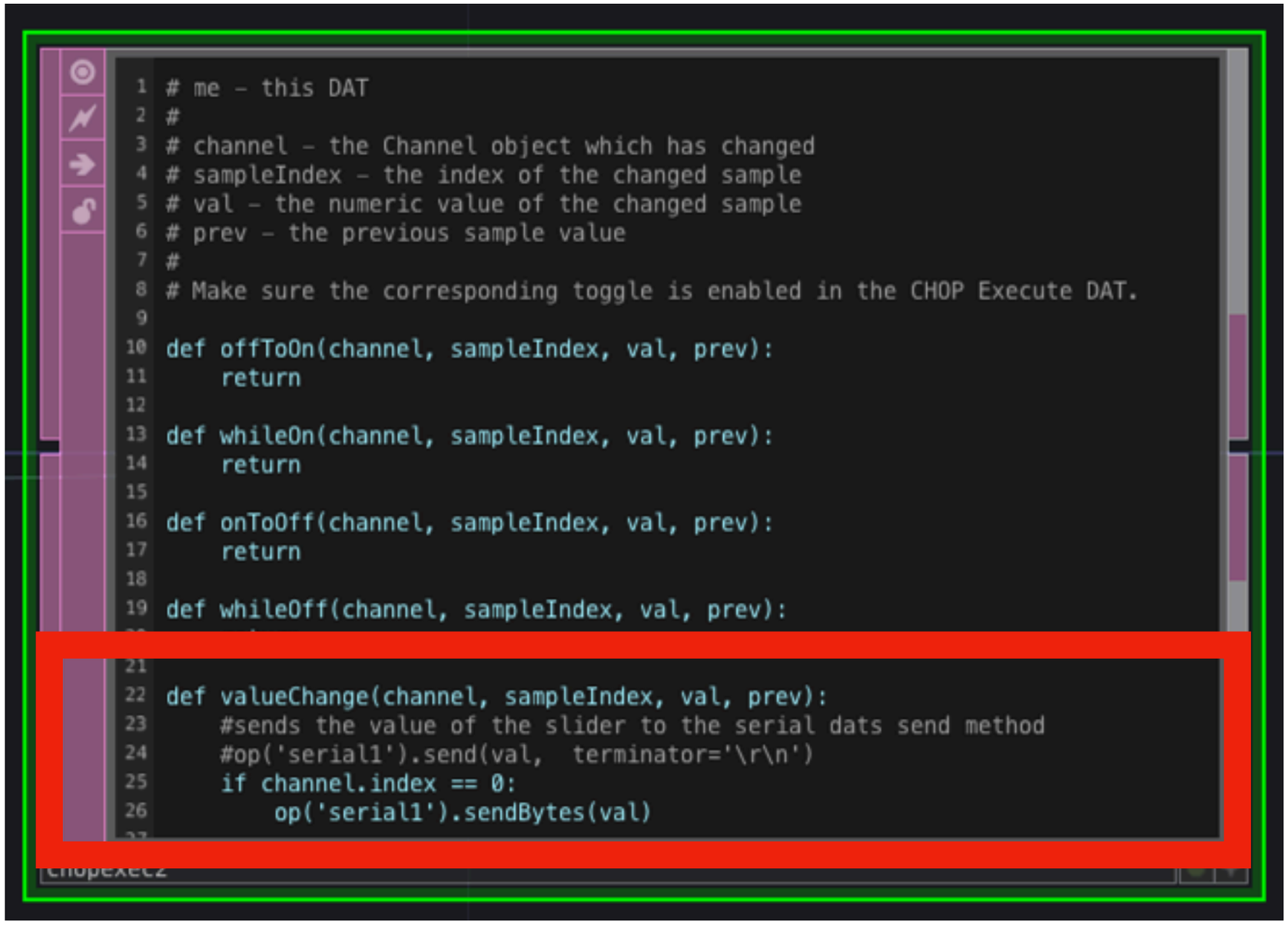

Lag
lag2
Lag
1
1
S
Time Slice
On


Logic
logic1
Convert Input
Off When Outside Bounds
Bounds
0.05
0.18
Time Slice
On


Lag
lag6
Lag
0.5
0.5
S
Time Slice
On


Math
math7
Integer
Ceiling
To Range
0
255


Serial
serial1
Connect
Received Data
Common
Active
On
Row/Callback Format
One Per Message
Port
/dev/cu.usbmodem1411
Baud Rate
9600
Data Bits
8
Parity
None
Stop Bits
2
DTR
Enable
RTS
Disable



```
# me - this DAT
#
# channel - the Channel object which has changed
# sampleIndex - the index of the changed sample
# val - the numeric value of the changed sample
# prev - the previous sample value
#
# Make sure the corresponding toggle is enabled in the CHOP Execute DAT.

def offToOn(channel, sampleIndex, val, prev):
    return

def whileOn(channel, sampleIndex, val, prev):
    return

def onToOff(channel, sampleIndex, val, prev):
    return

def whileOff(channel, sampleIndex, val, prev):


def valueChange(channel, sampleIndex, val, prev):
    #sends the value of the slider to the serial dats send method
    #op('serial1').send(val,  terminator='\r\n')
    if channel.index == 0:
        op('serial1').sendBytes(val)
```