

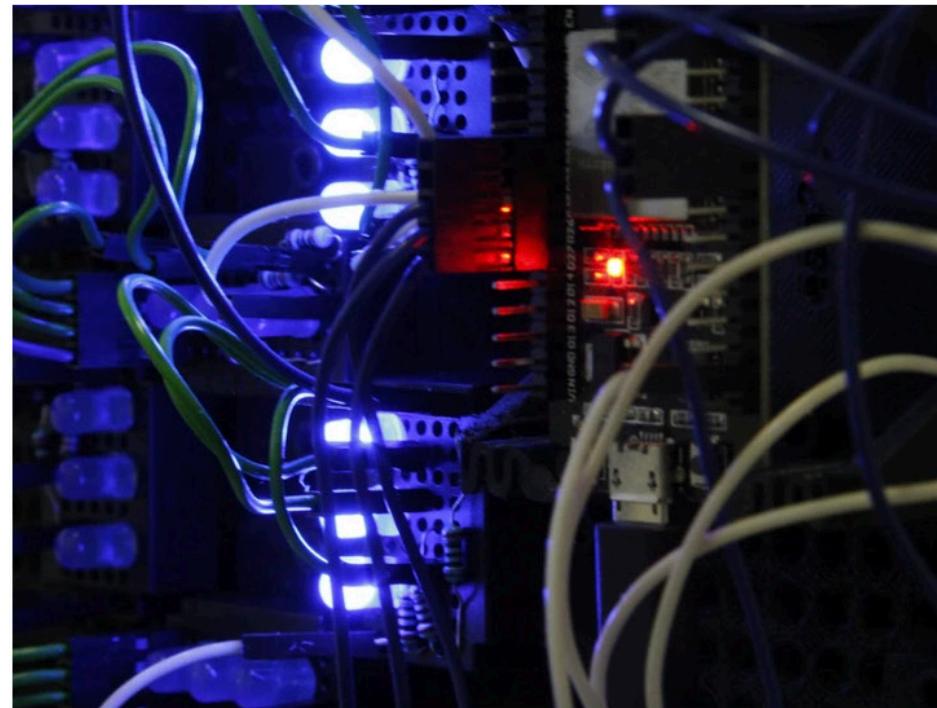
OPEN HARDWARE WORKSHOP

by Silvia Binda Heiserova

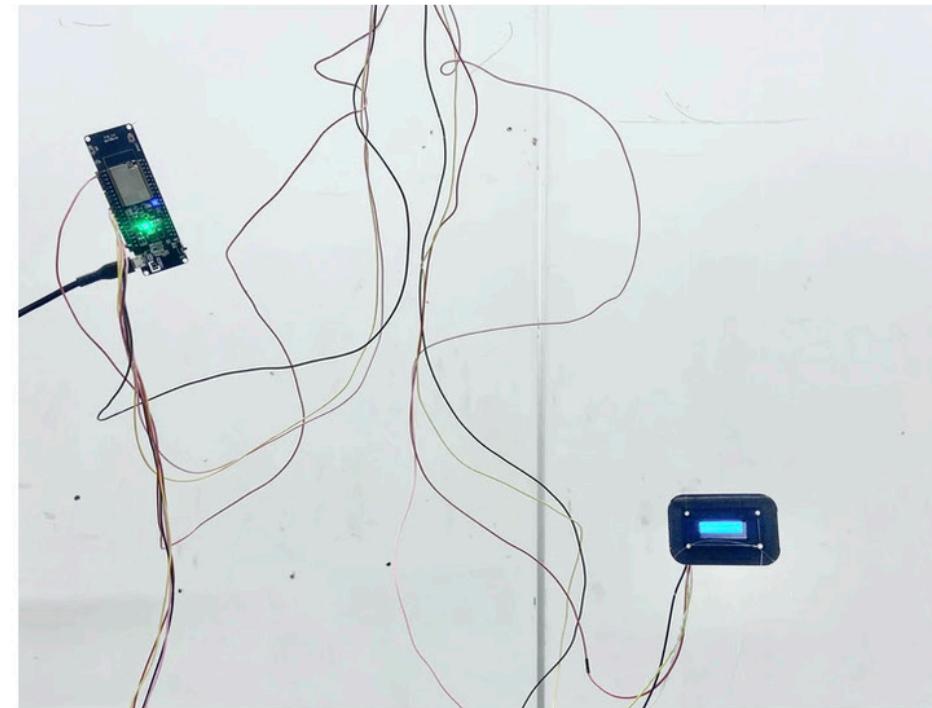


BALTAN LABORATORIES

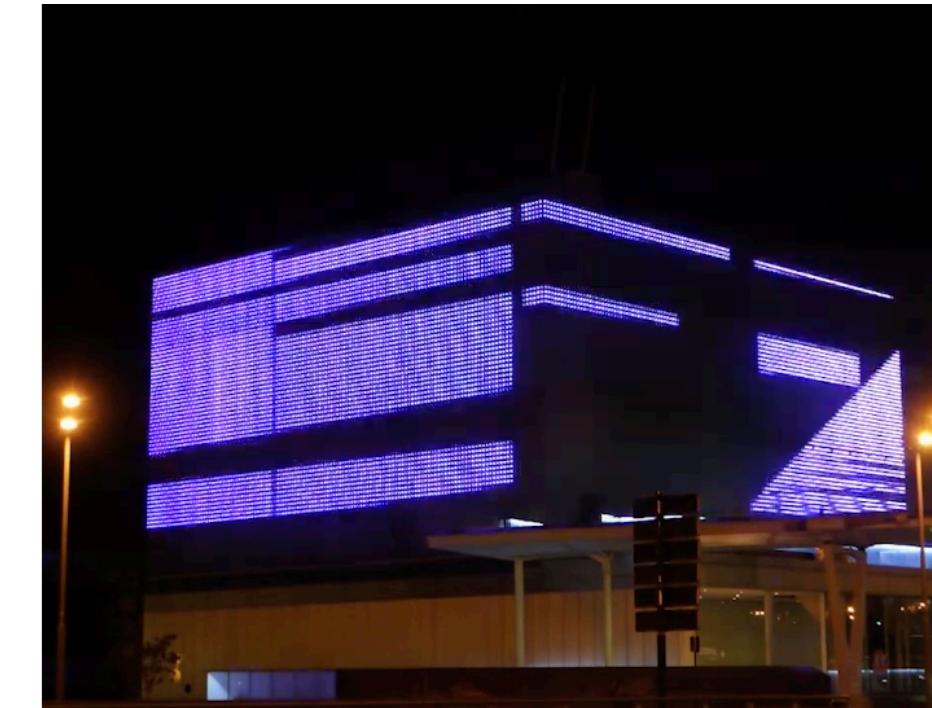
1) INTRODUCTION: Silvia



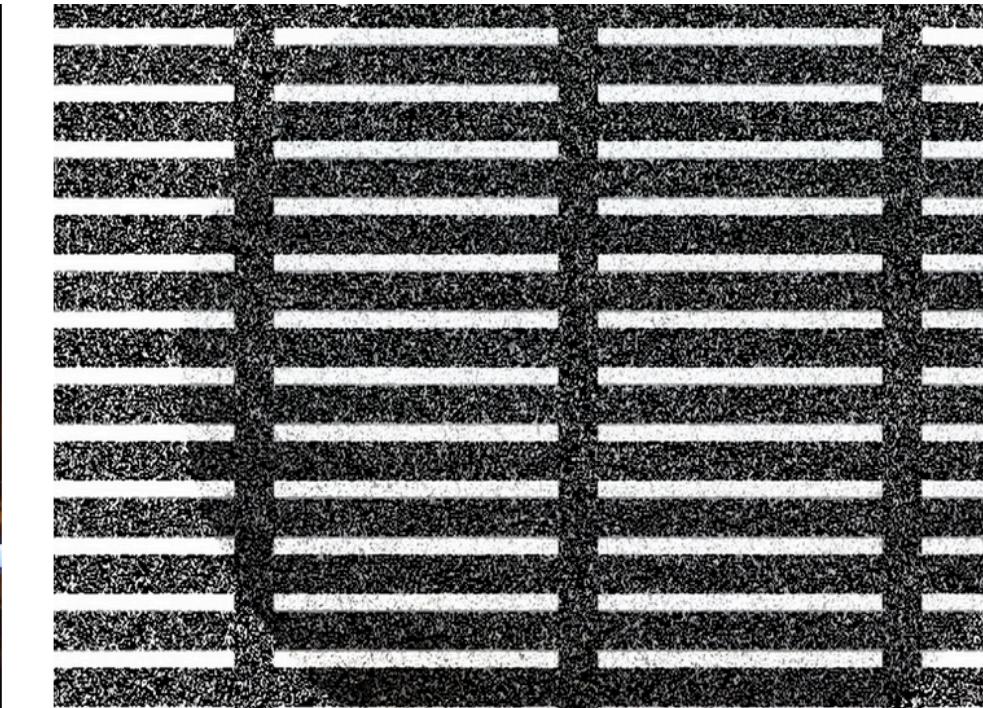
Conversatorio



#Woman Critical Interface



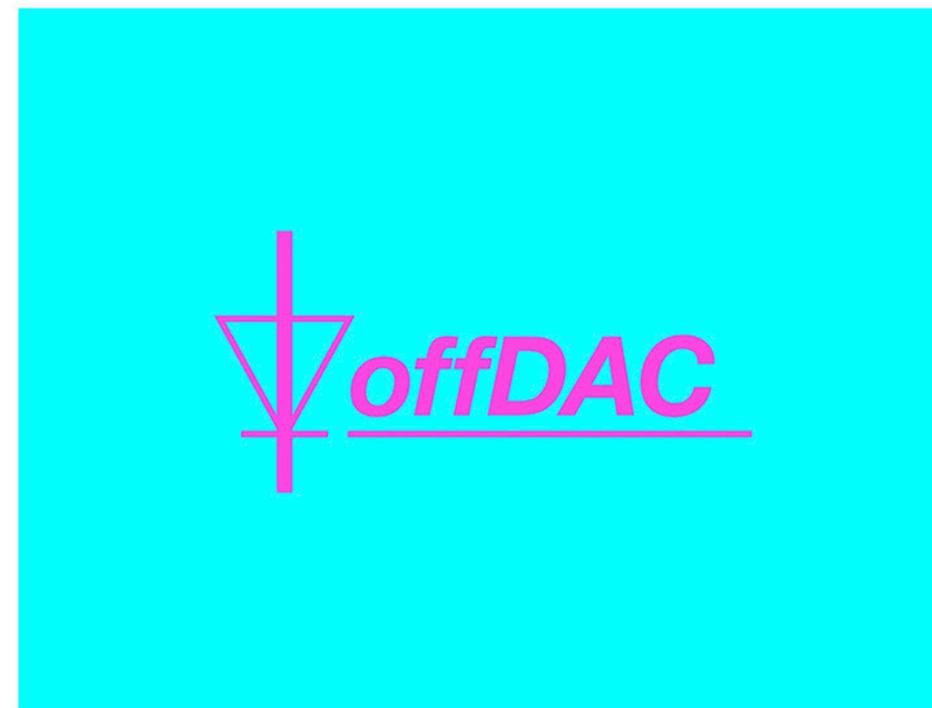
Hyperimage



TEXTO



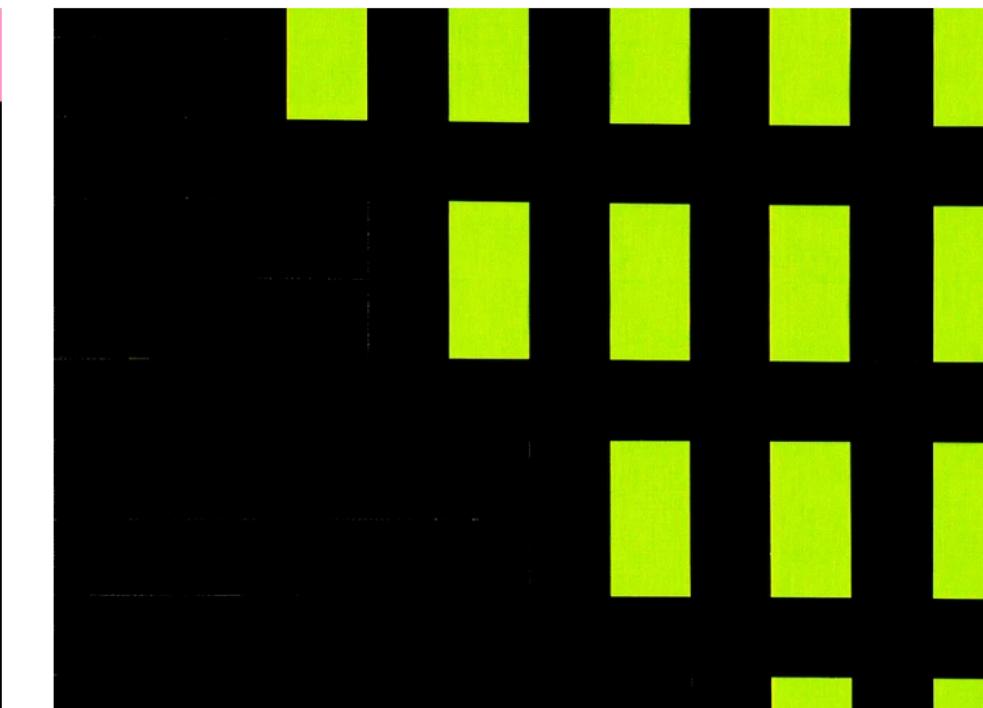
Vending Machine



offDAC

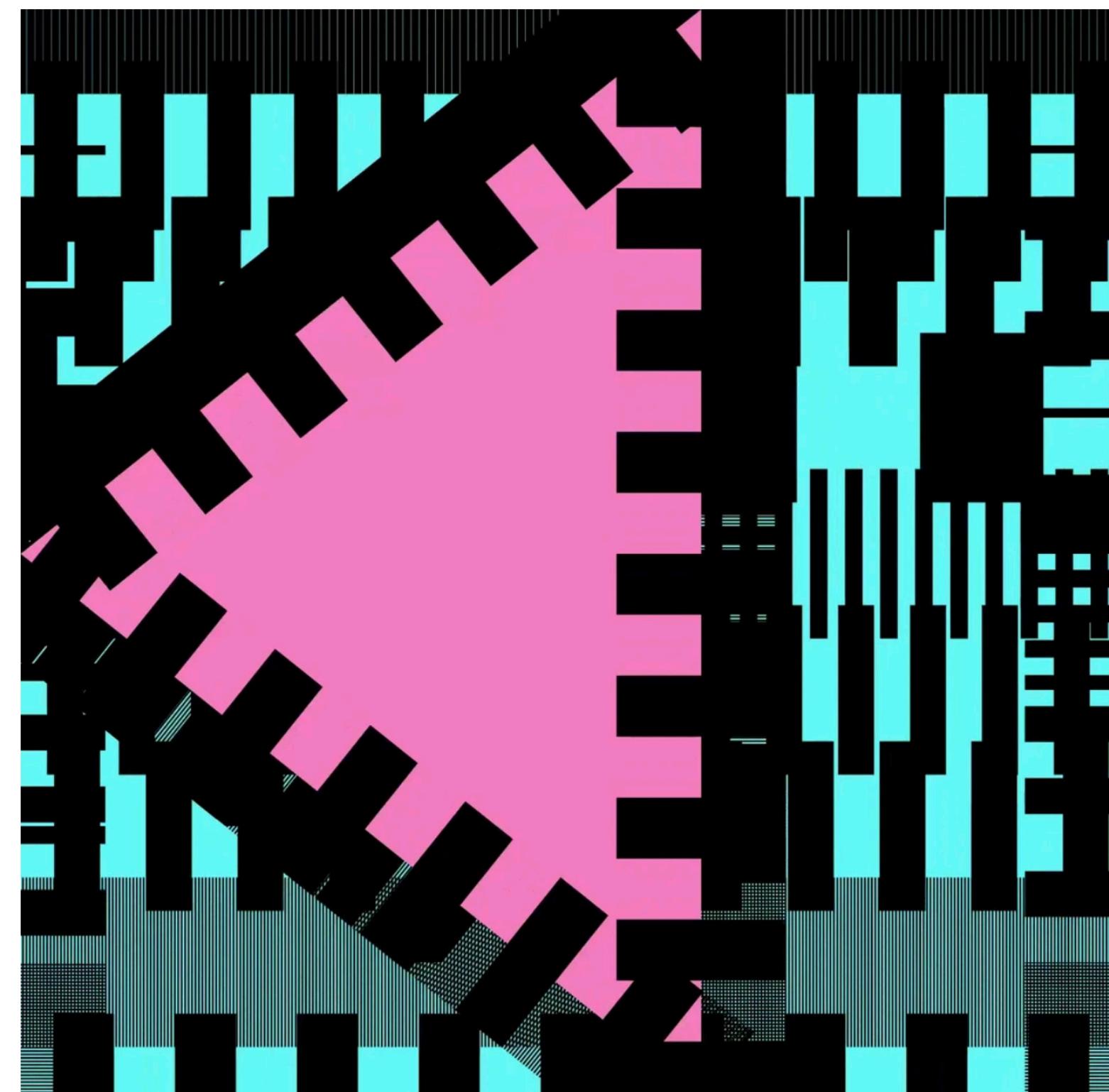
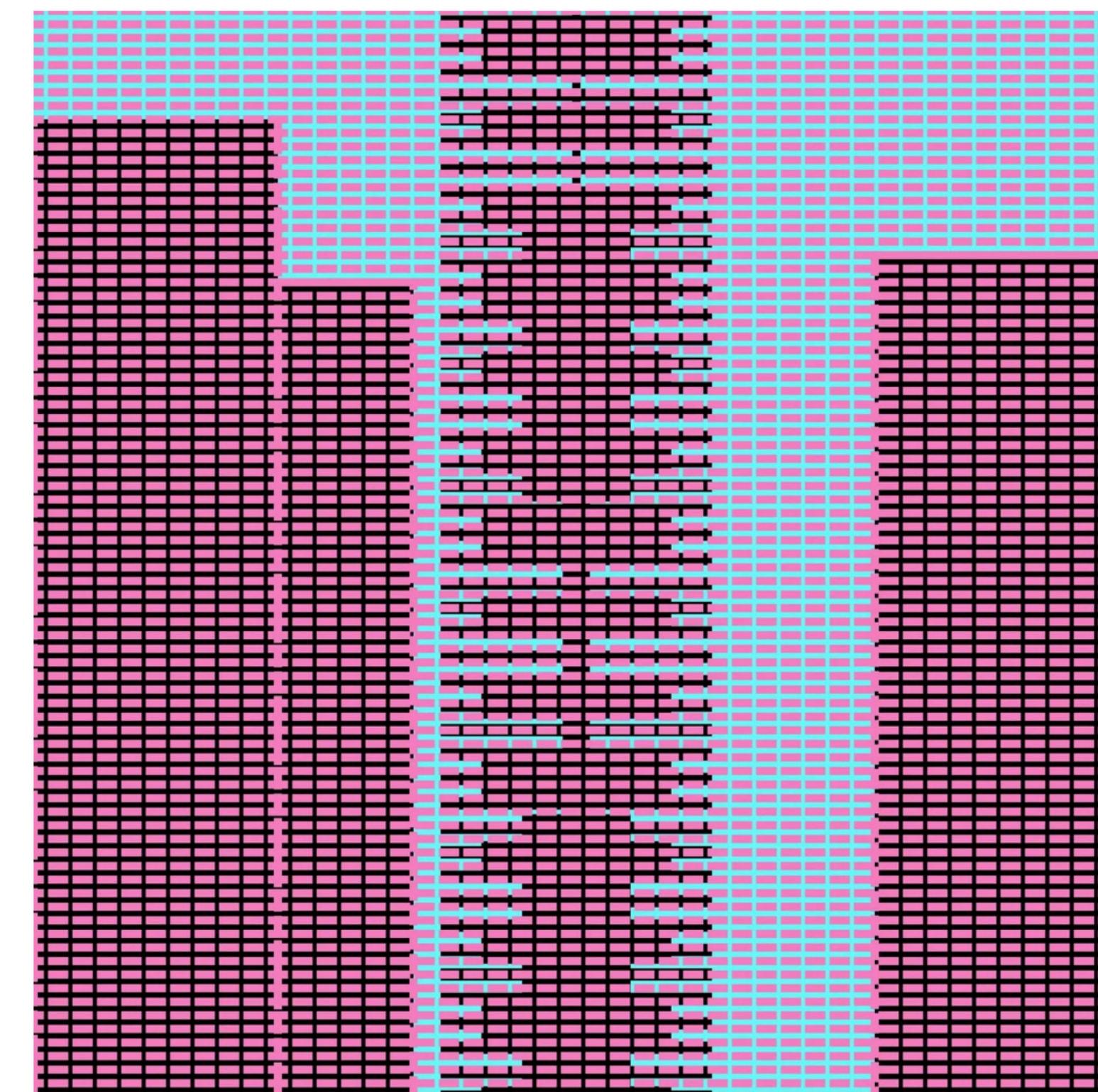
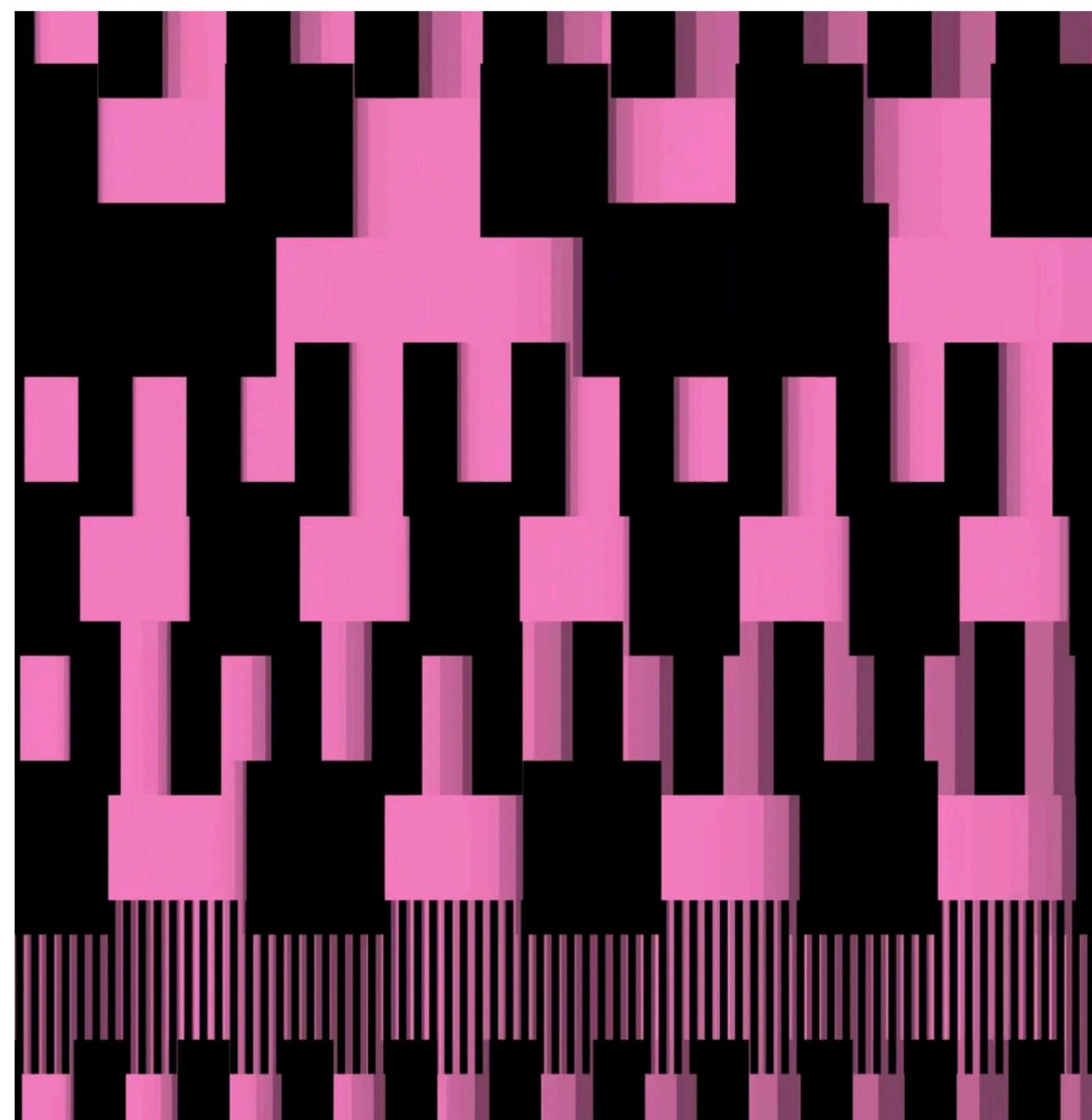


Moving Cityscapes

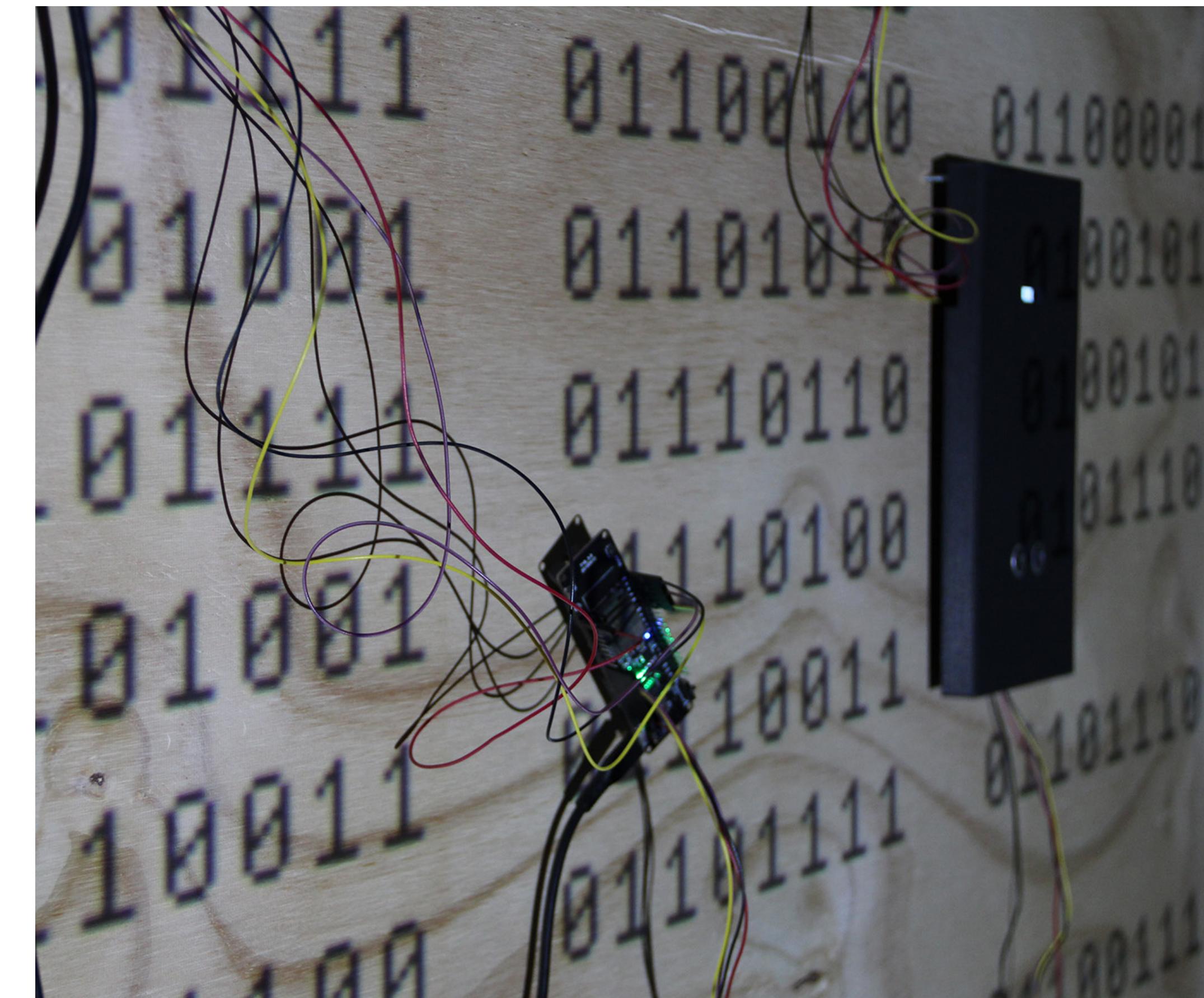


Binary Landscapes

Moving Cityscapes, 2021



#Woman Critical Interface, 2021-2023



offDAC community (since 2023)

offdac.com

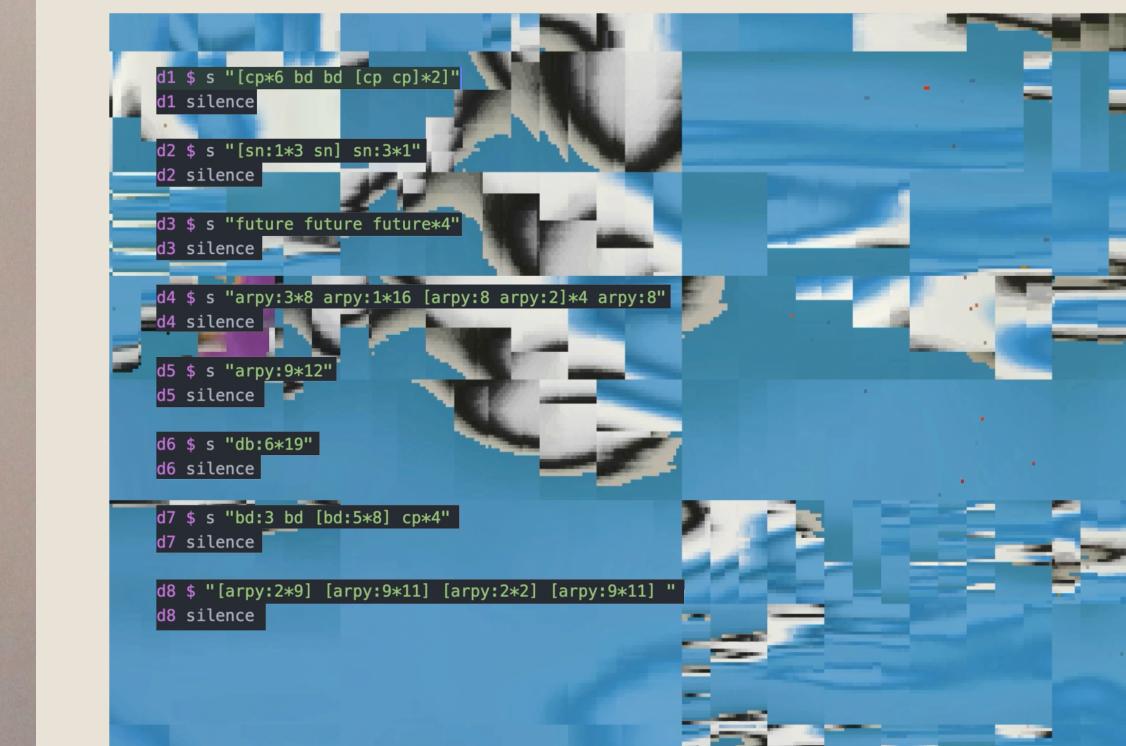


15/11/23
16:00
Etopia (Aula 3)
Zaragoza
offDAC
event 001
FREE
ENTRY

13/01/24
11:00 - 17:00
offDAC LAB
Košická 52/A, Bratislava
offDAC
event 002
Live Coding
Workshop

Live Coding

Tidal Cycles workshop

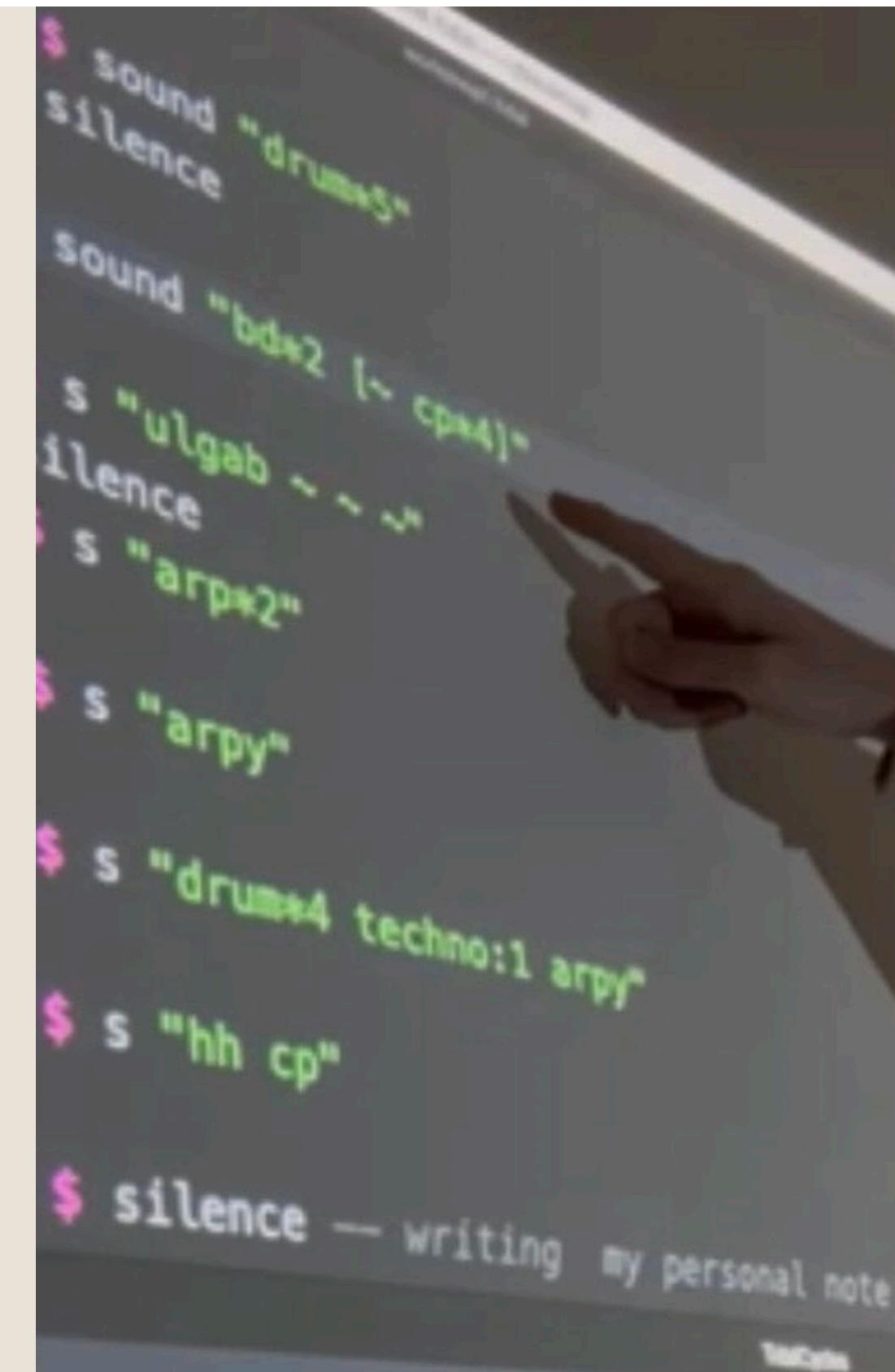


Live Coding workshop organized by offDAC with kind support from Tidal Cycles. Registration is necessary, see link in the QR code below. No previous knowledge is needed. Participation is free of charge. Snacks and beverages will be provided by offDAC. For further information see offdac.com or write us at info@offdac.com

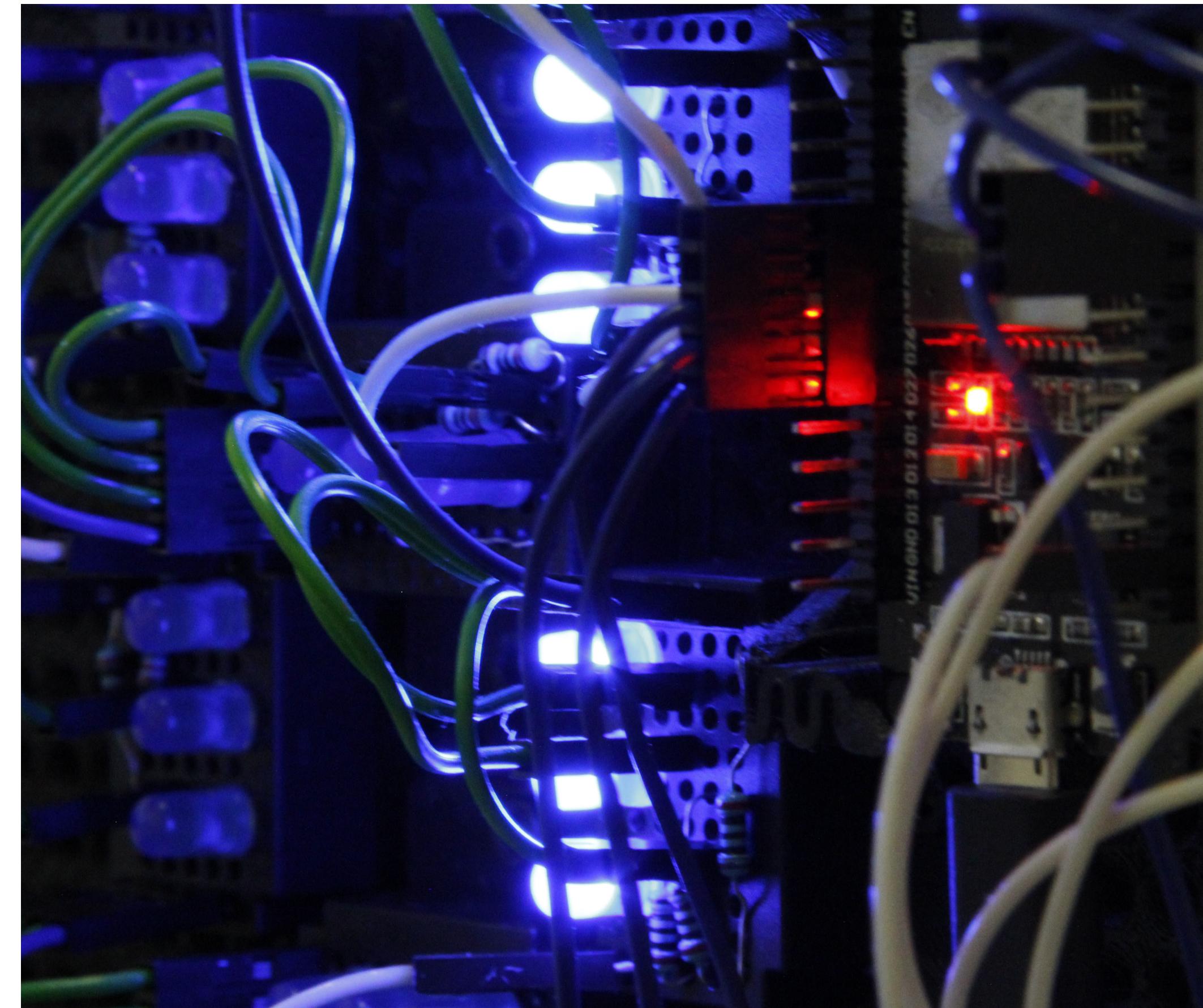
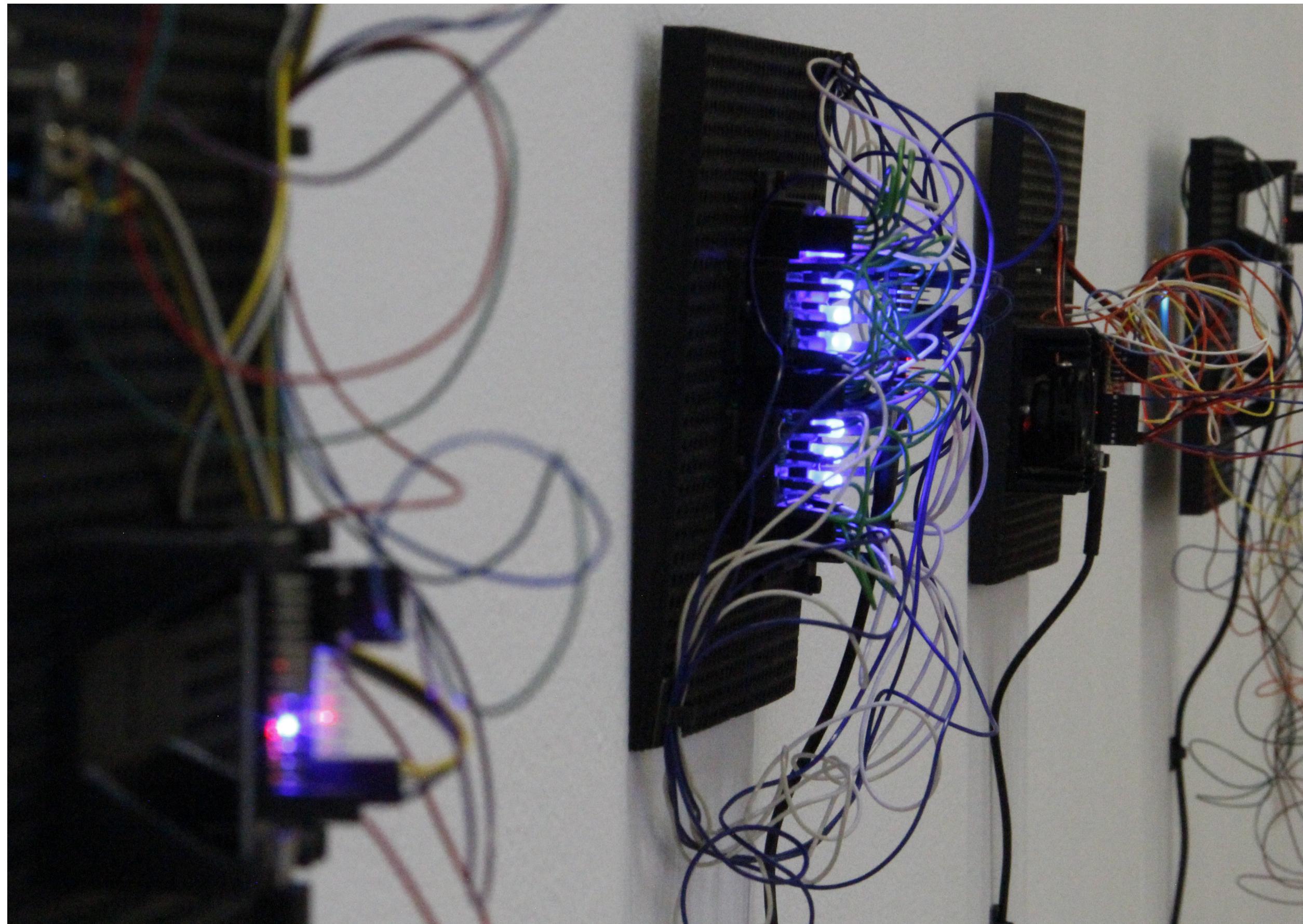
Organized by:
 offDAC

With kind support by:
 tidalcycles

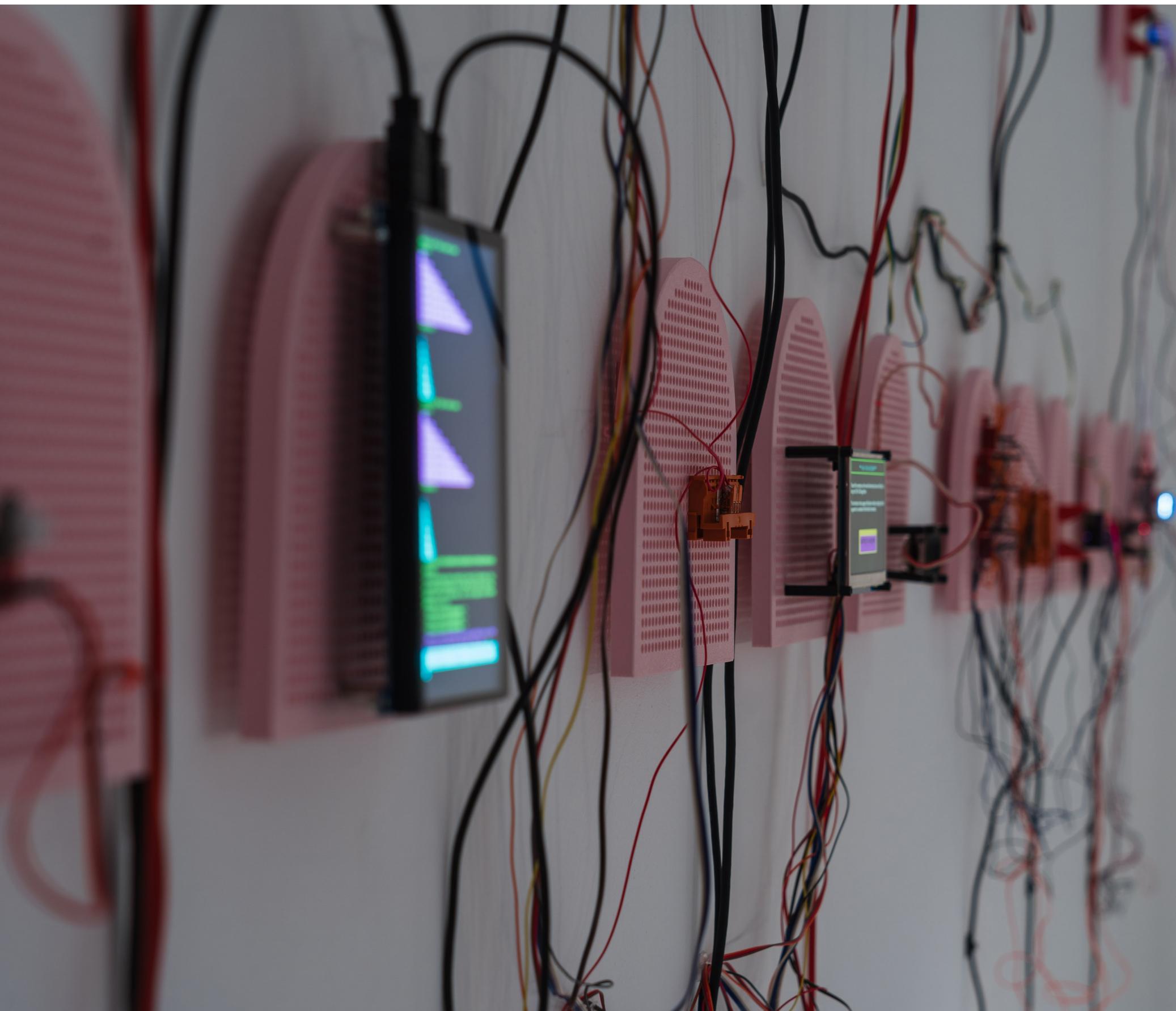
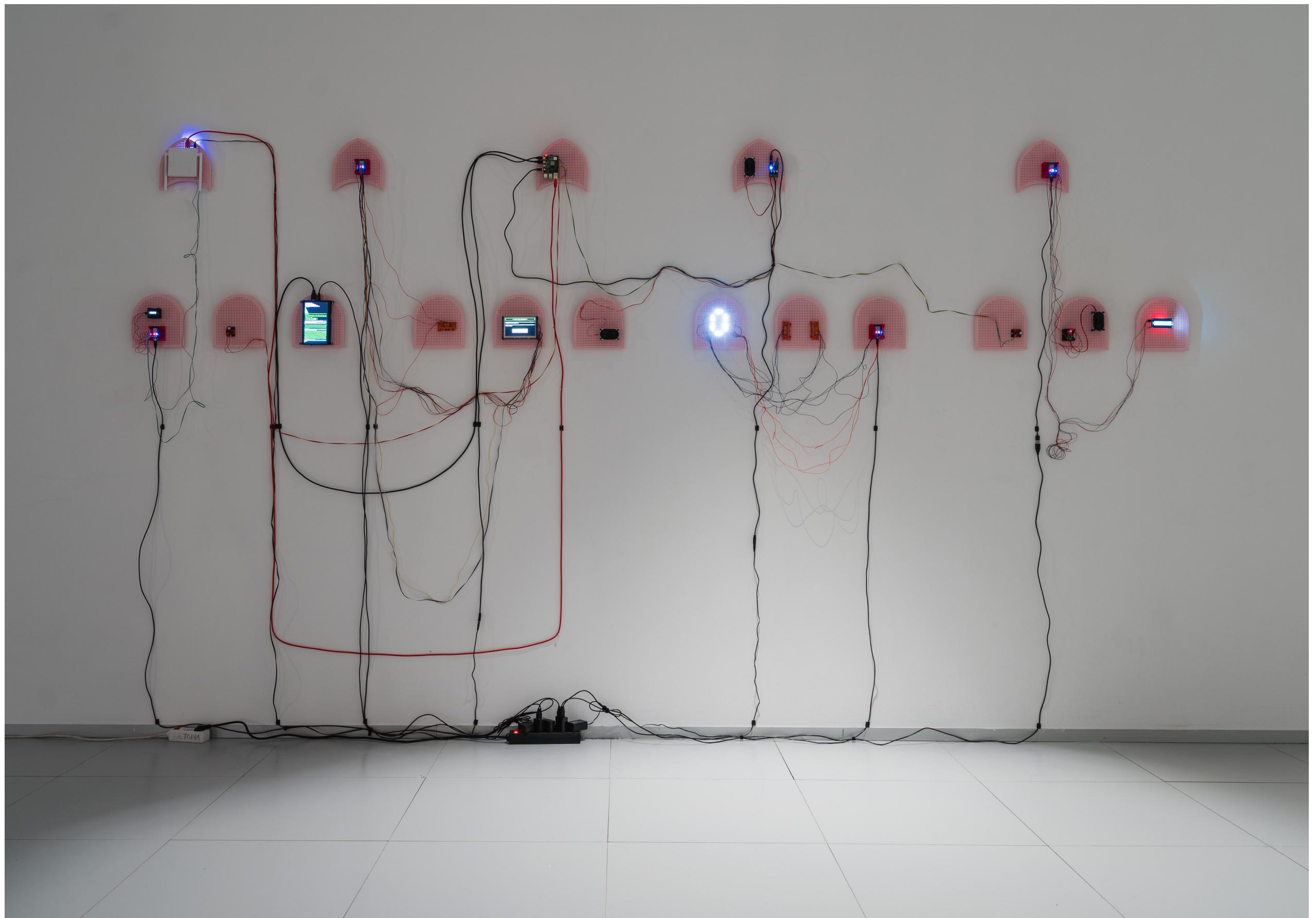
Register:

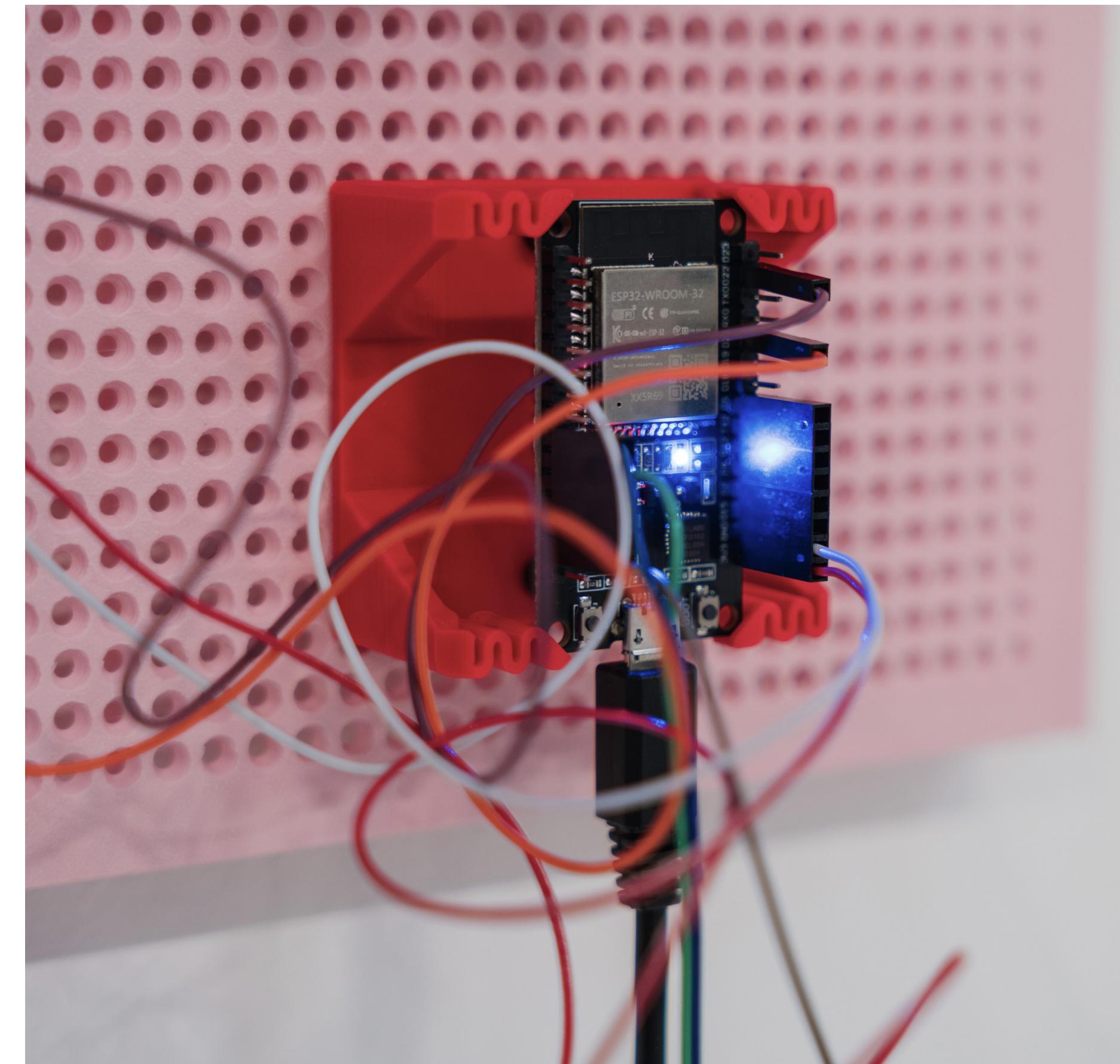
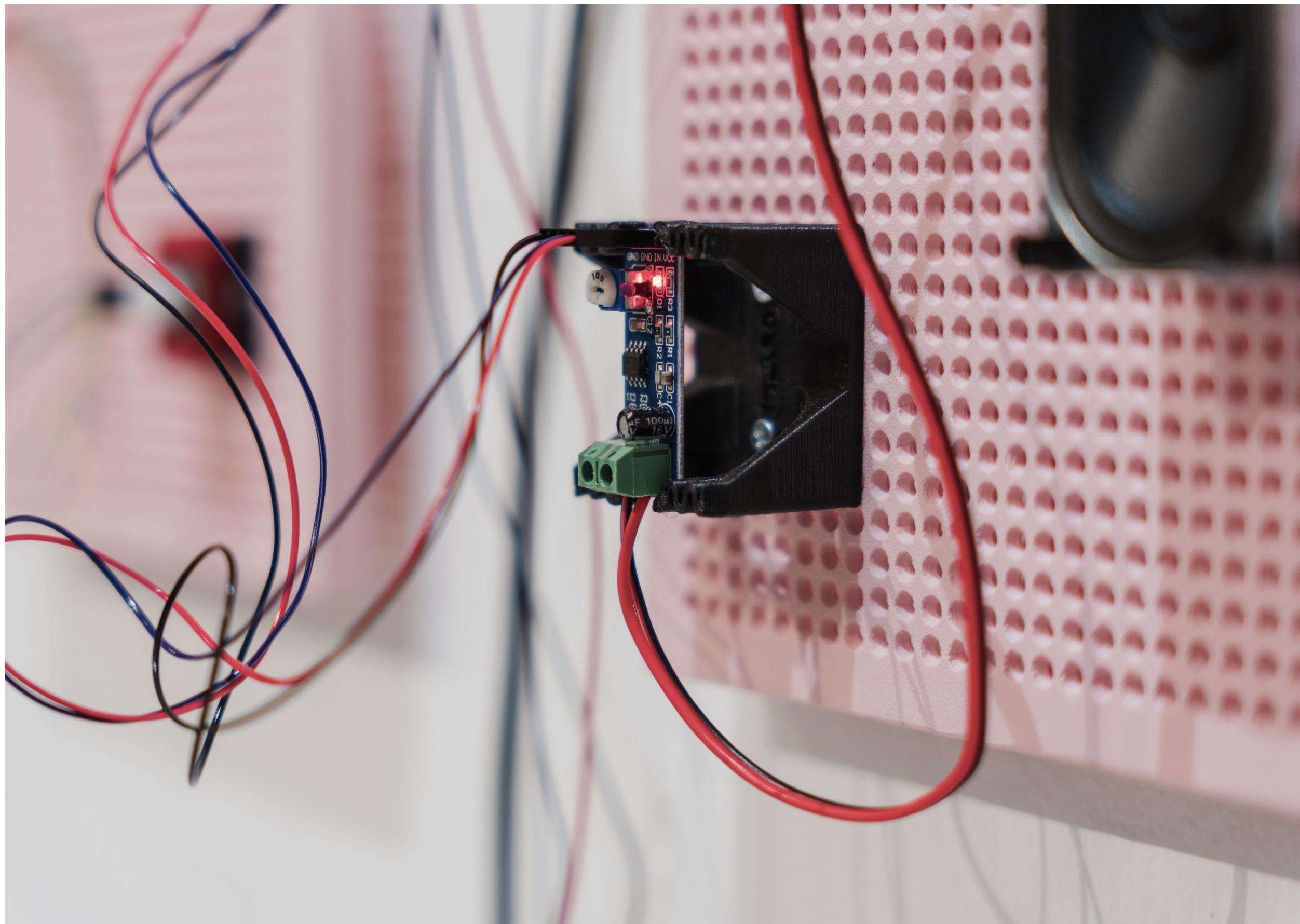
Conversatorio, 2023



Hash Breakdown, 2024



Hash Breakdown, 2024



OPEN HARDWARE PRINCIPLES

<https://opencourse.com>

“Open hardware,” or “open source hardware,” refers to the design specifications of a physical object which are licensed in such a way that said object can be studied, modified, created, and distributed by anyone.

OPEN HARDWARE PRINCIPLES

<https://opencourse.com>

- the “source code” for open hardware—schematics, blueprints, logic designs, code, etc.—is available for modification by anyone (under permissive licenses, such as MIT license)
- to provide as many people as possible the ability to construct, remix, and share their knowledge of hardware design and function

PLATFORM FOR SHARING PROJECTS

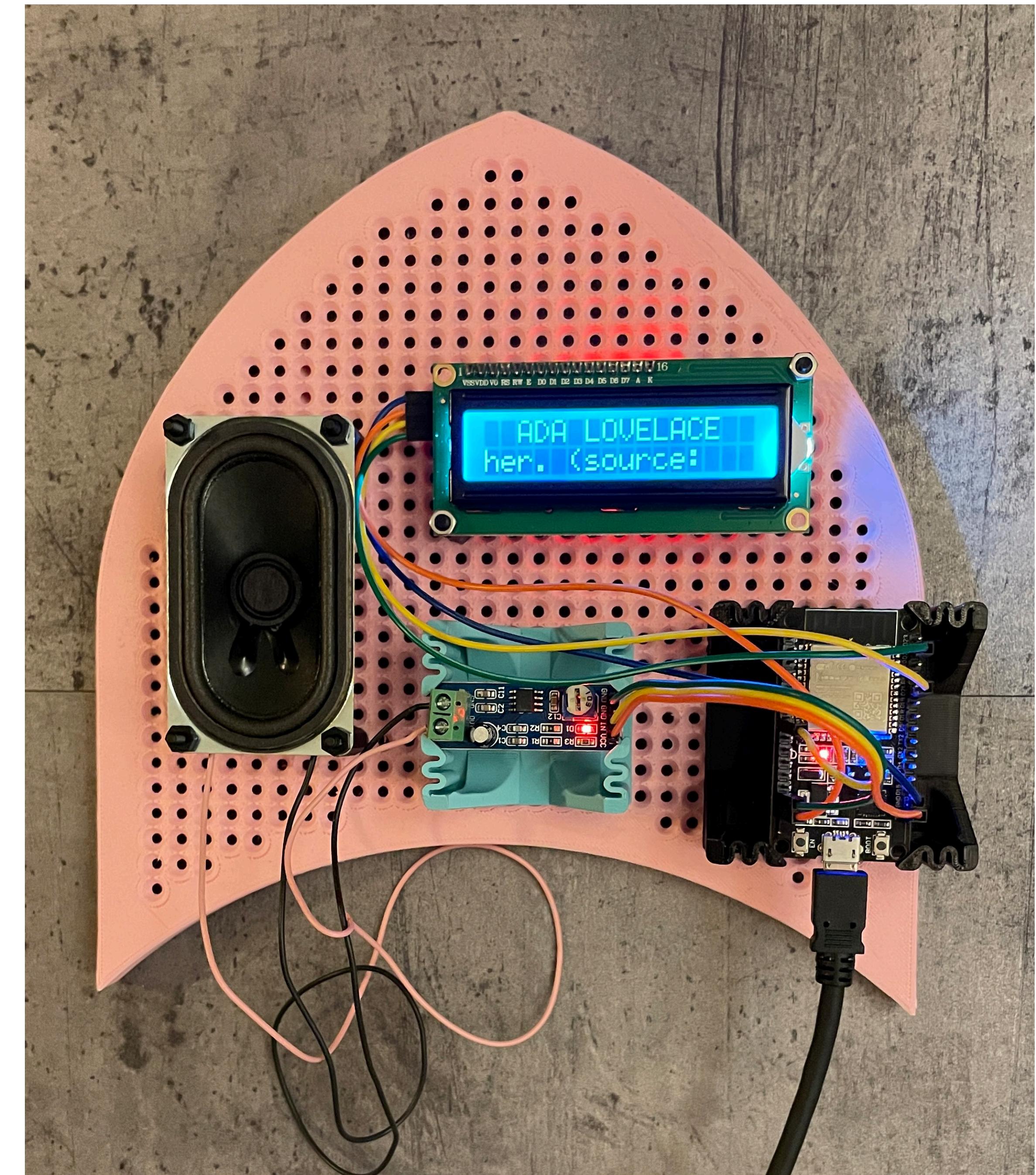


GitHub is a cloud-based platform where you can store, share, and work together with others to write code.

https://github.com/silviabinda/open_hardware_workshop_baltan

PLAN FOR TODAY IS:

**MAKE OUR
VERY OWN
HARDWARE
ARTEFACT**



PLEASE RAISE YOUR HAND WHO:

- EVER SOLDERED SOMETHING

PLEASE RAISE YOUR HAND WHO:

- EVER SOLDERED SOMETHING**
- KNOWS WHAT IS AN “ESP32”**

PLEASE RAISE YOUR HAND WHO:

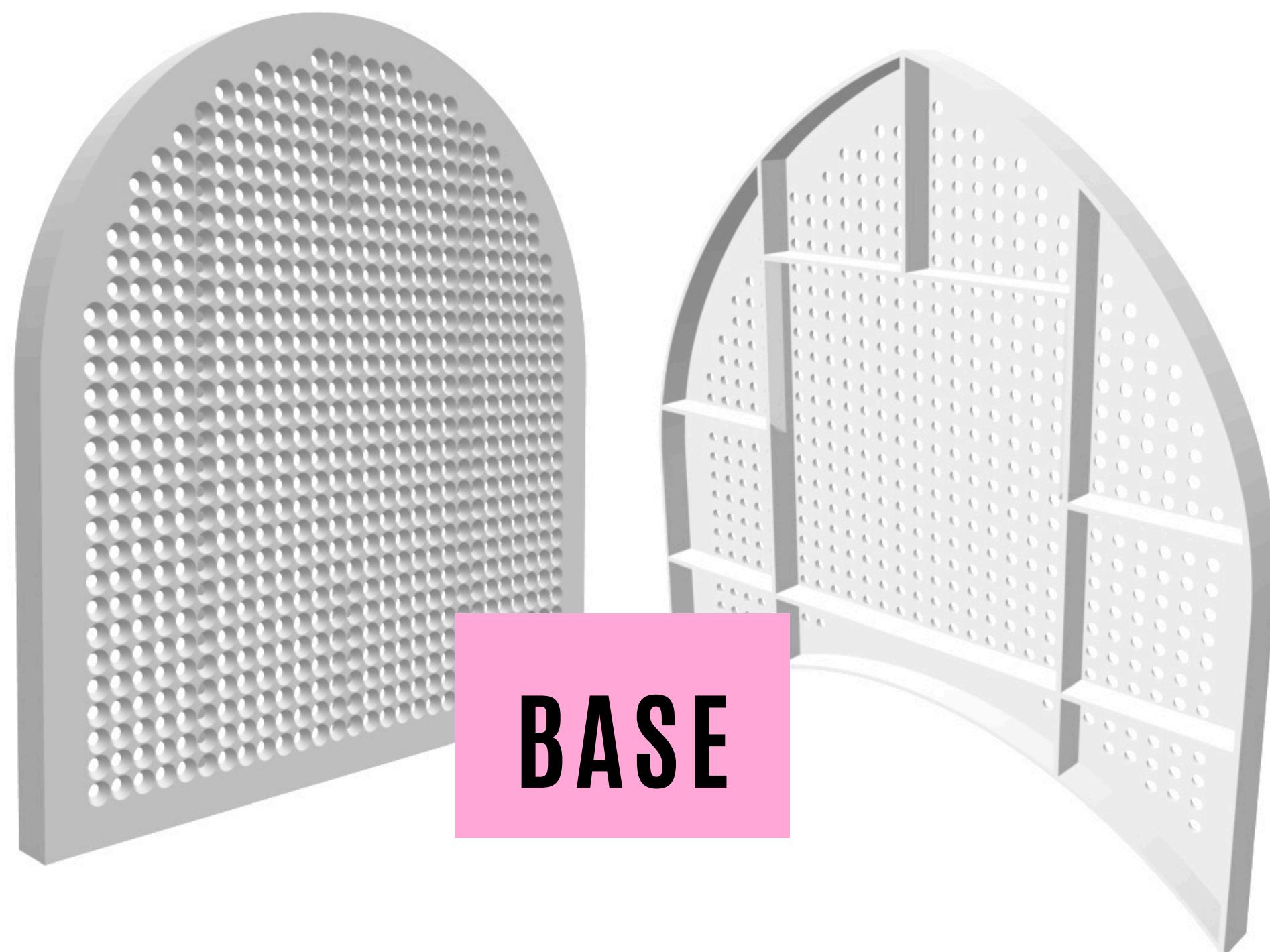
- EVER SOLDERED SOMETHING
- KNOWS WHAT IS AN “ESP32”
- EVER CODED IN PYTHON

PLEASE RAISE YOUR HAND WHO:

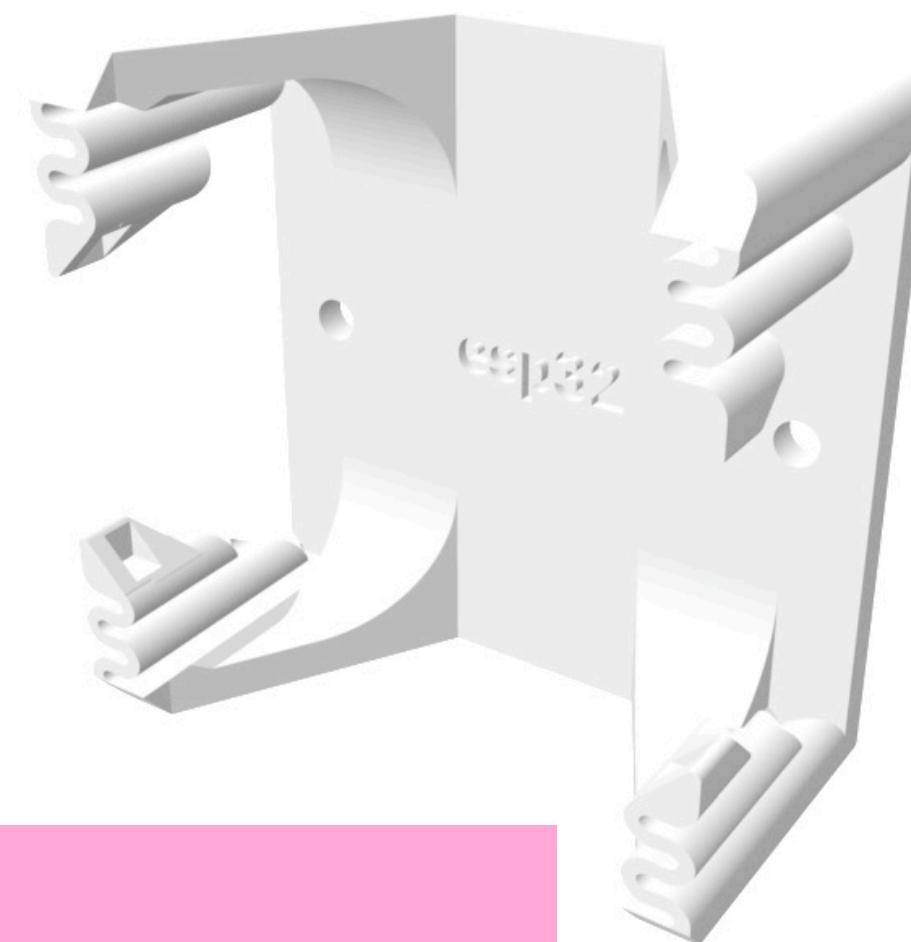
- EVER SOLDERED SOMETHING
- KNOWS WHAT IS AN “ESP32”
- EVER CODED IN PYTHON
- KNOWS WHAT A DUPONT CONNECTOR IS

Items we will use:

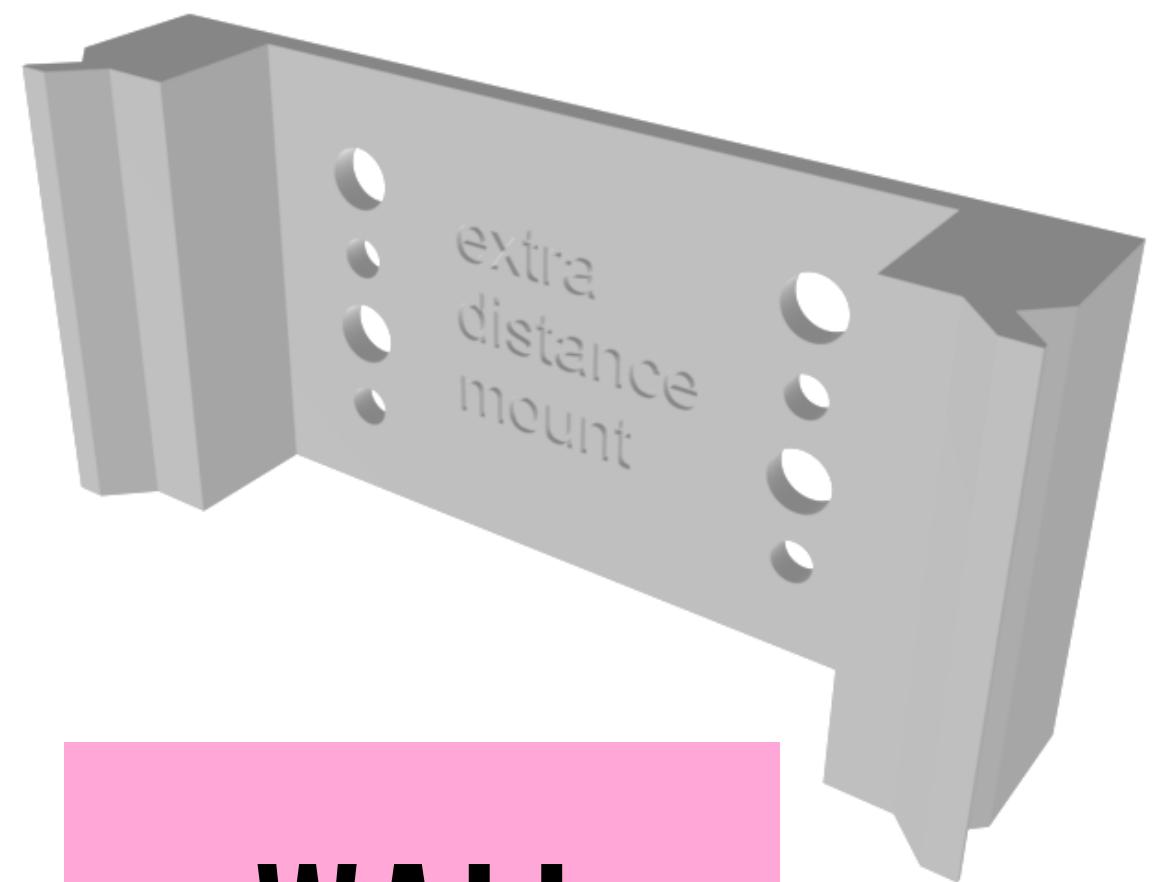
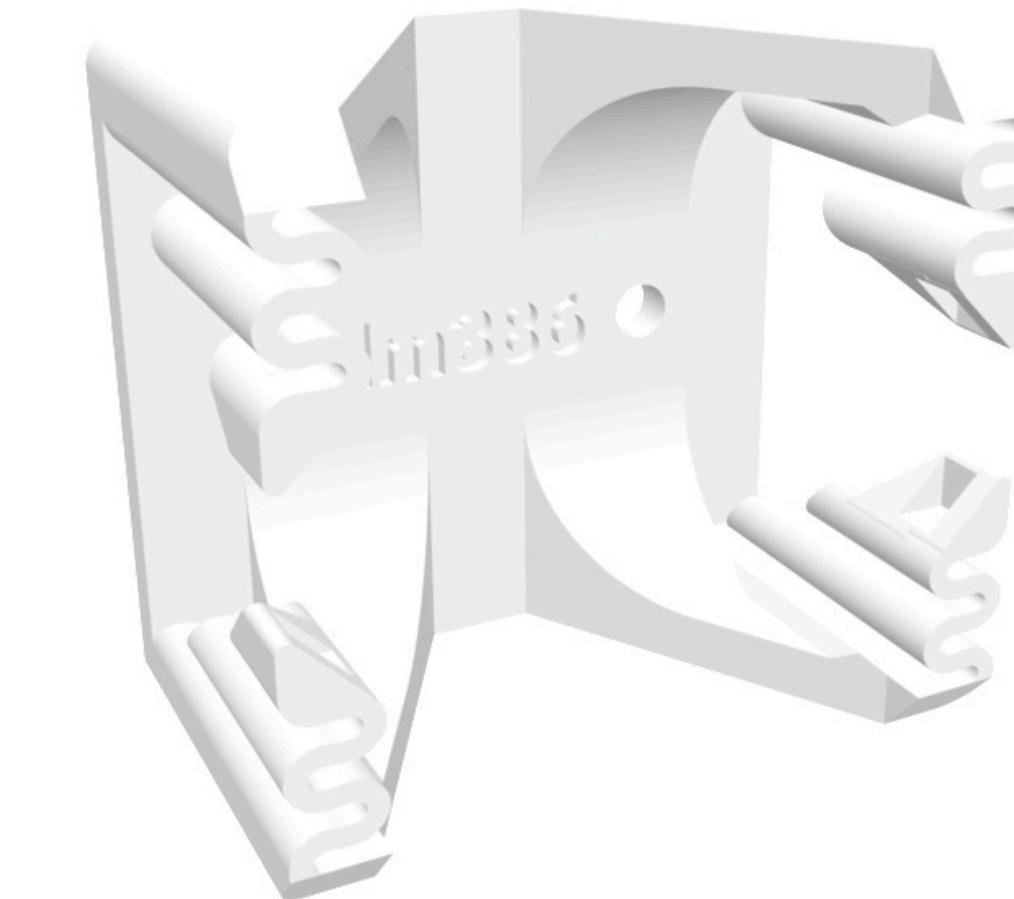
1) 3D prints



BASE



HOLDERS



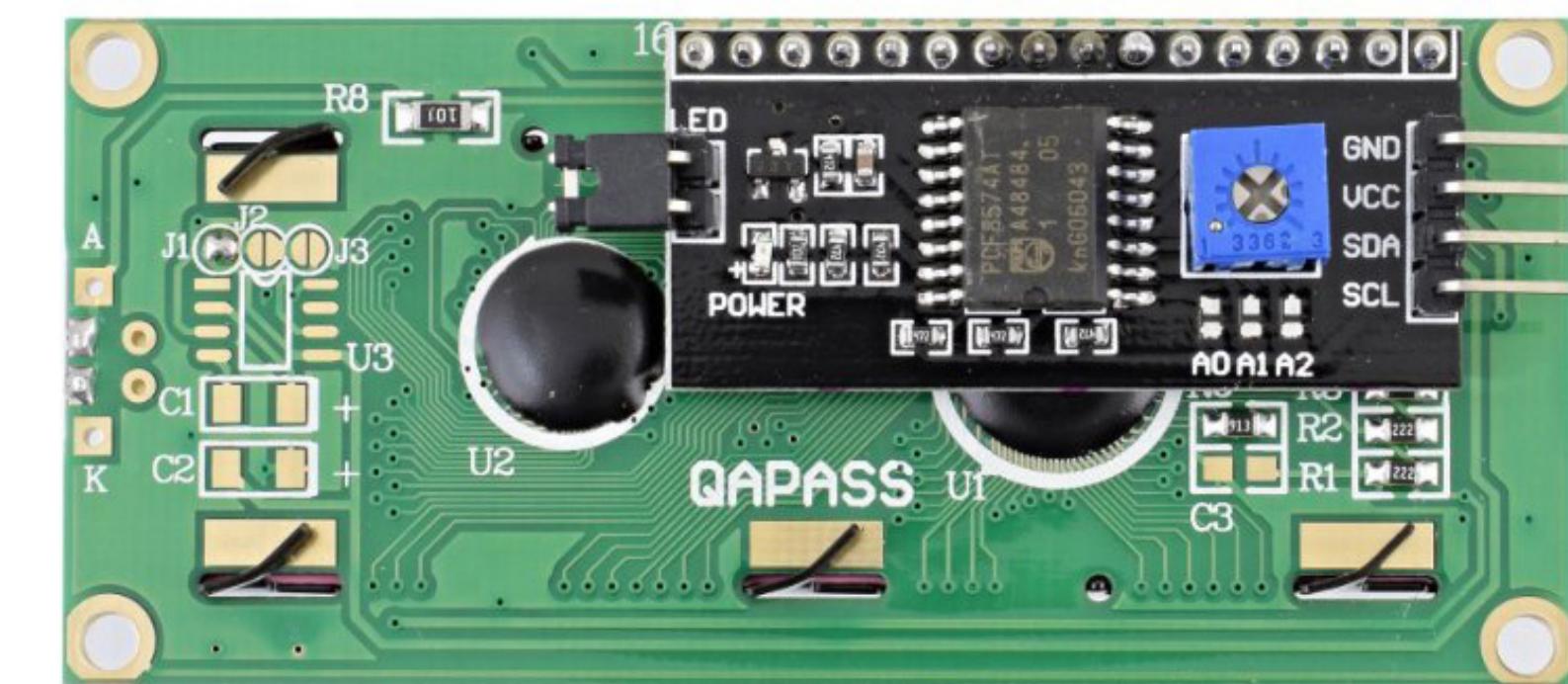
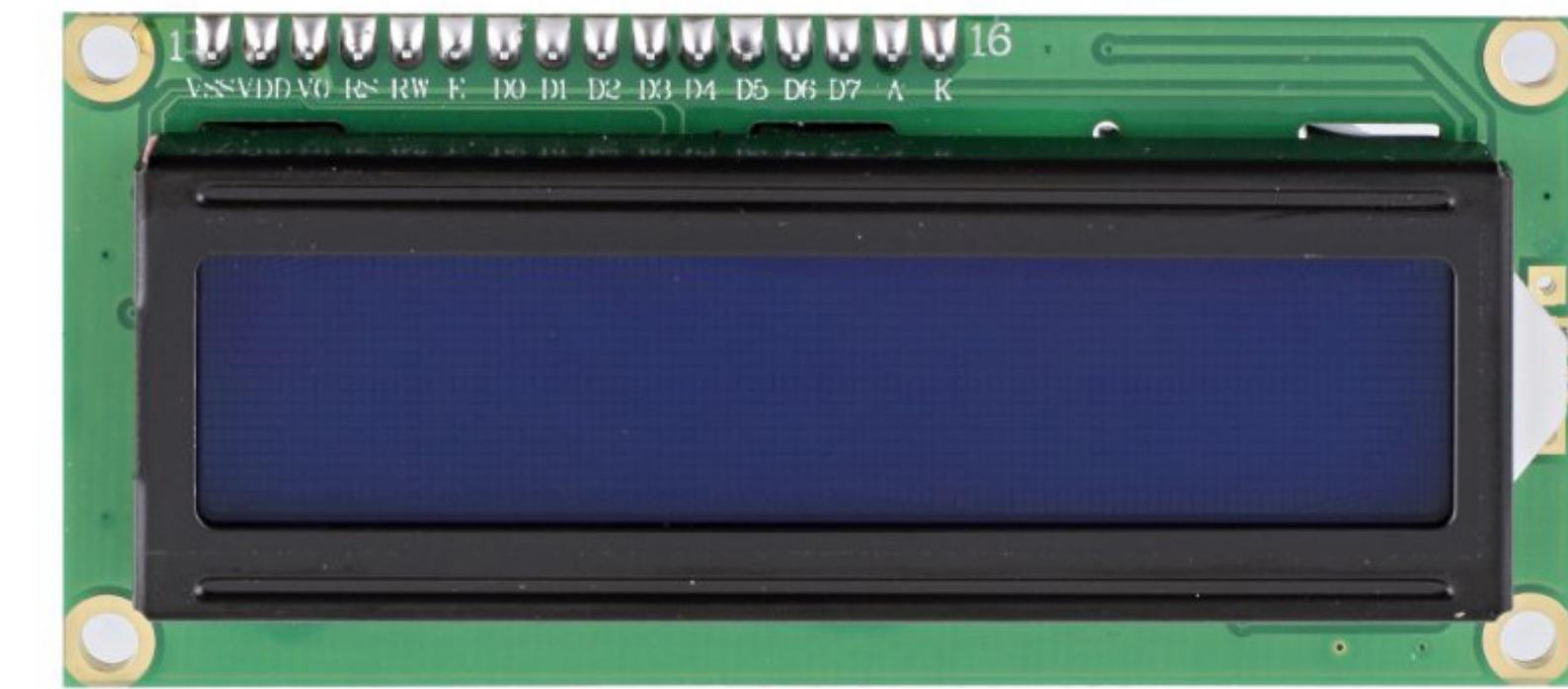
WALL
MOUNT

Items we will use:

2) ESP32 microcontroller



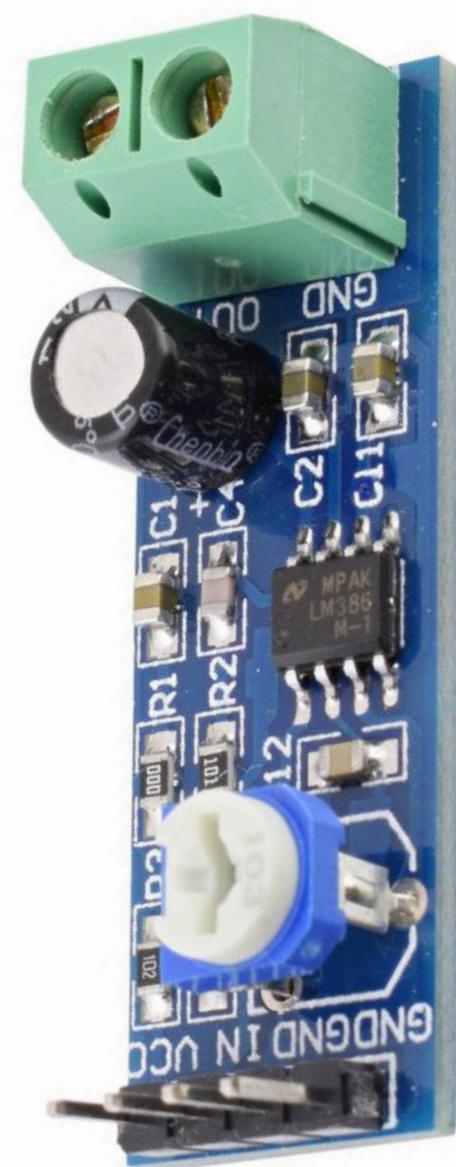
3) LCD display (16x2, I2C)



blue
or
yellow

Items we will use:

4) LM386 amplifier



5) mini speaker



Items we will use:

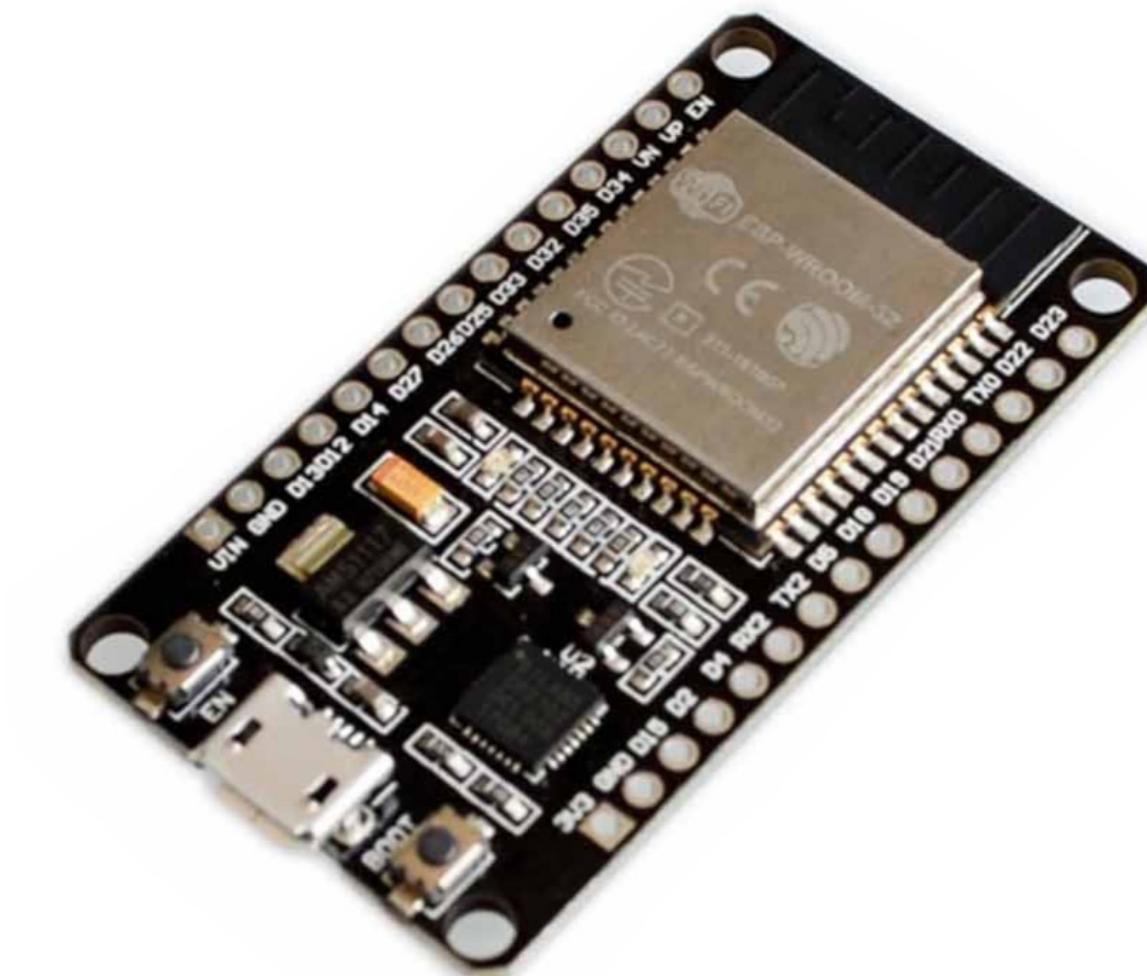
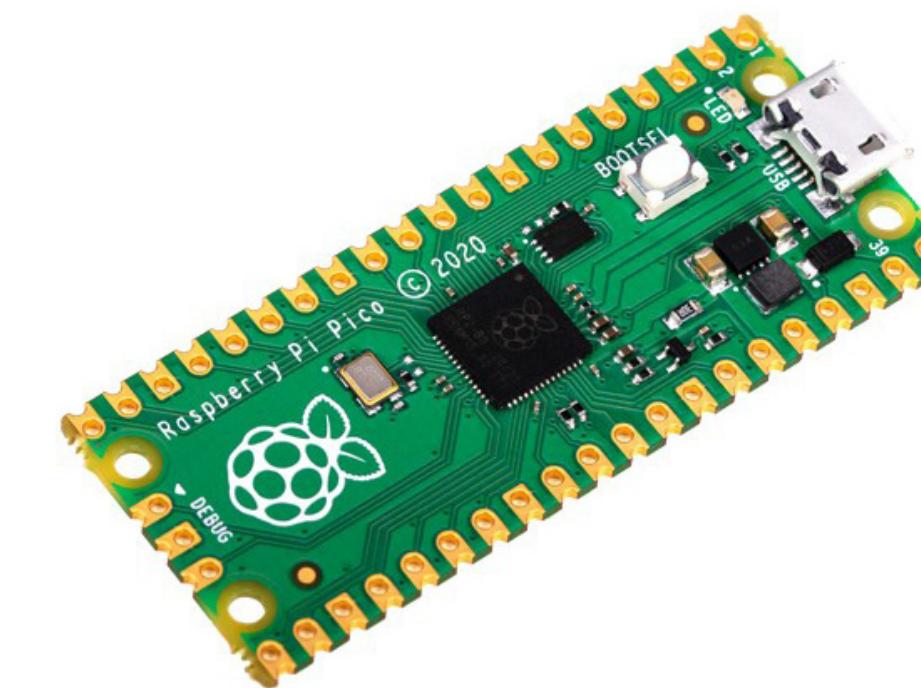
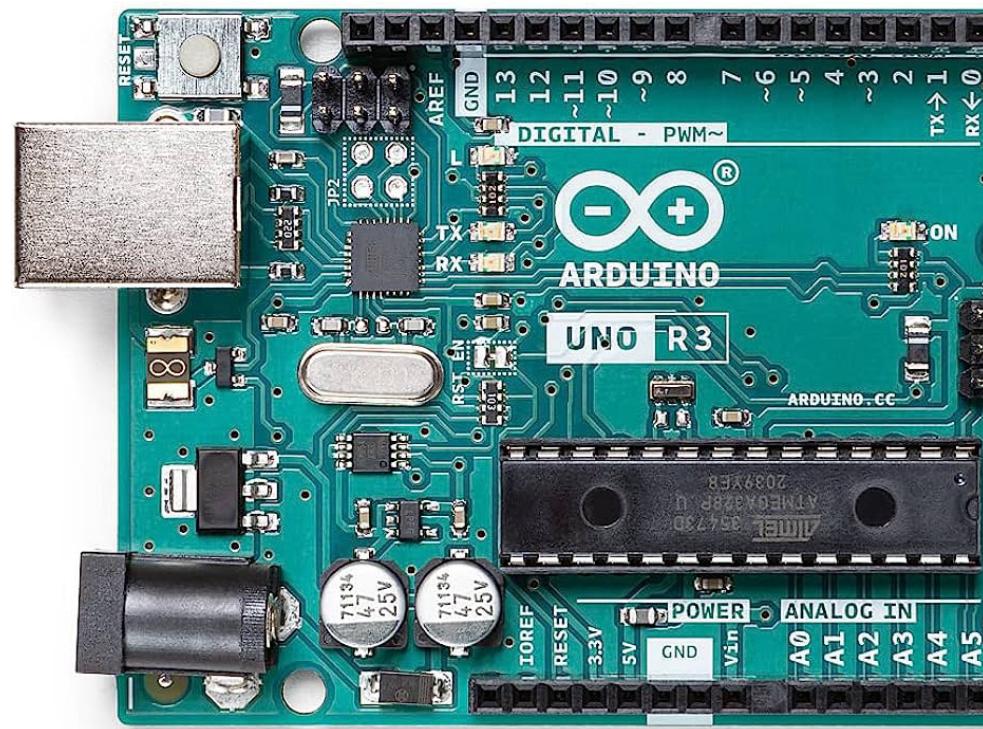
6) wires, spacers, screws, tools



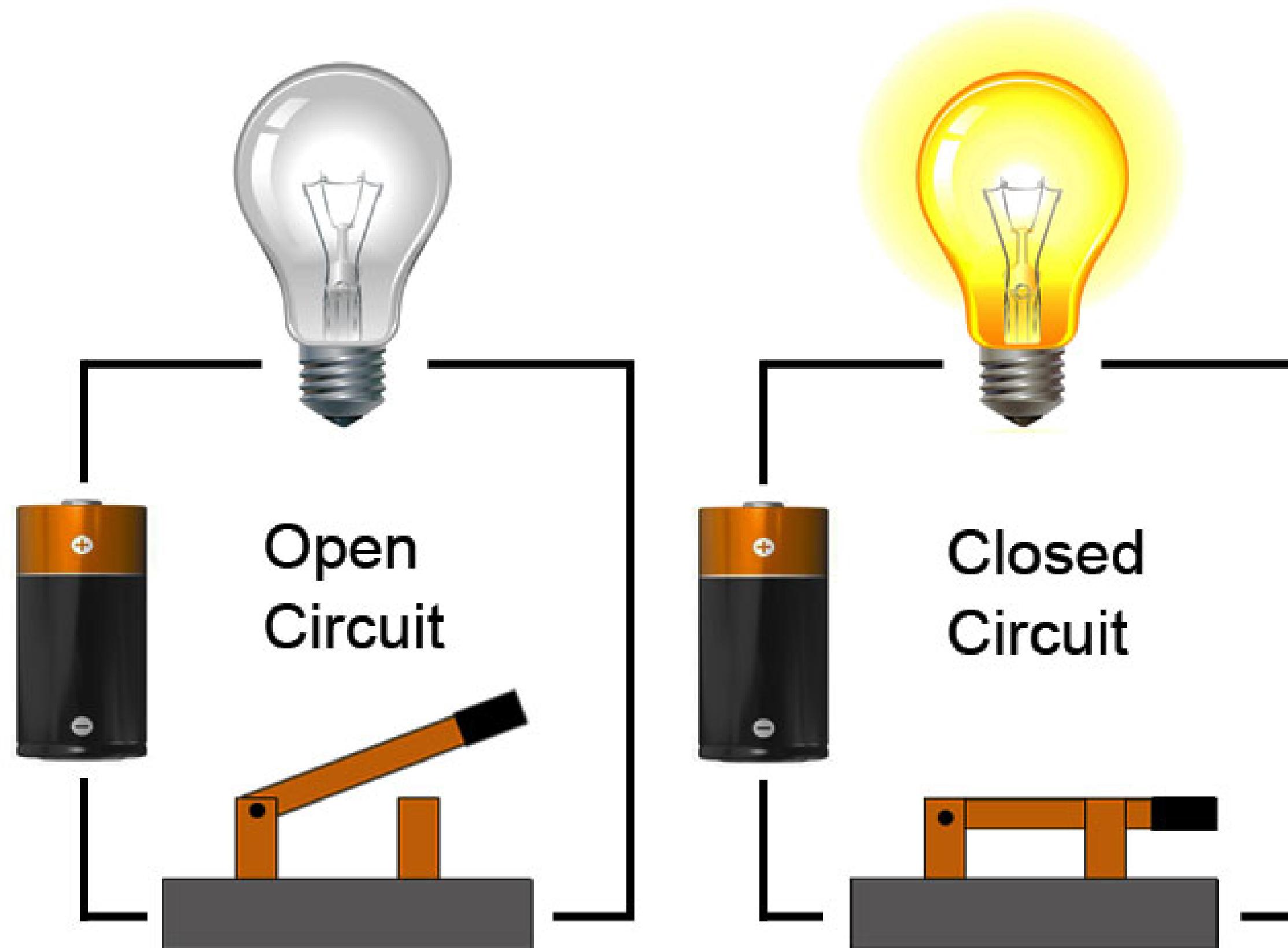
brief introduction to electronics:

MICROCONTROLLERS

- = a small, low-cost computer-on-a-chip
- designed to perform a specific set of tasks
- applications: controlling machines, sensing and monitoring devices, art, IoT
- typically consists of a central processing unit (CPU), memory, input/output (I/O) pin
- common examples = Arduino, Raspberry Pi Pico, ESP 32



OPEN AND CLOSED CIRCUITS



circuit is a complete and closed path through which electric current can flow

closed circuit would allow the flow of electricity between **power (+)** and ground (-)

open circuit would break the flow of electricity between **power (+)** and ground (-)

POSITIVE VOLTAGE AND GROUND

Positive Voltage:

“VCC”

“3V3”

“+”

Ground (carries a voltage of 0V):

“GND”

“_”

!!!! It is very important to prevent short circuits by making sure that the **positive voltage (+)** is never wired directly to ground (-) !!!!!

INSPECT THE COMPONENTS AND FIND POSITIVE VOLTAGE AND GROUND ON EACH

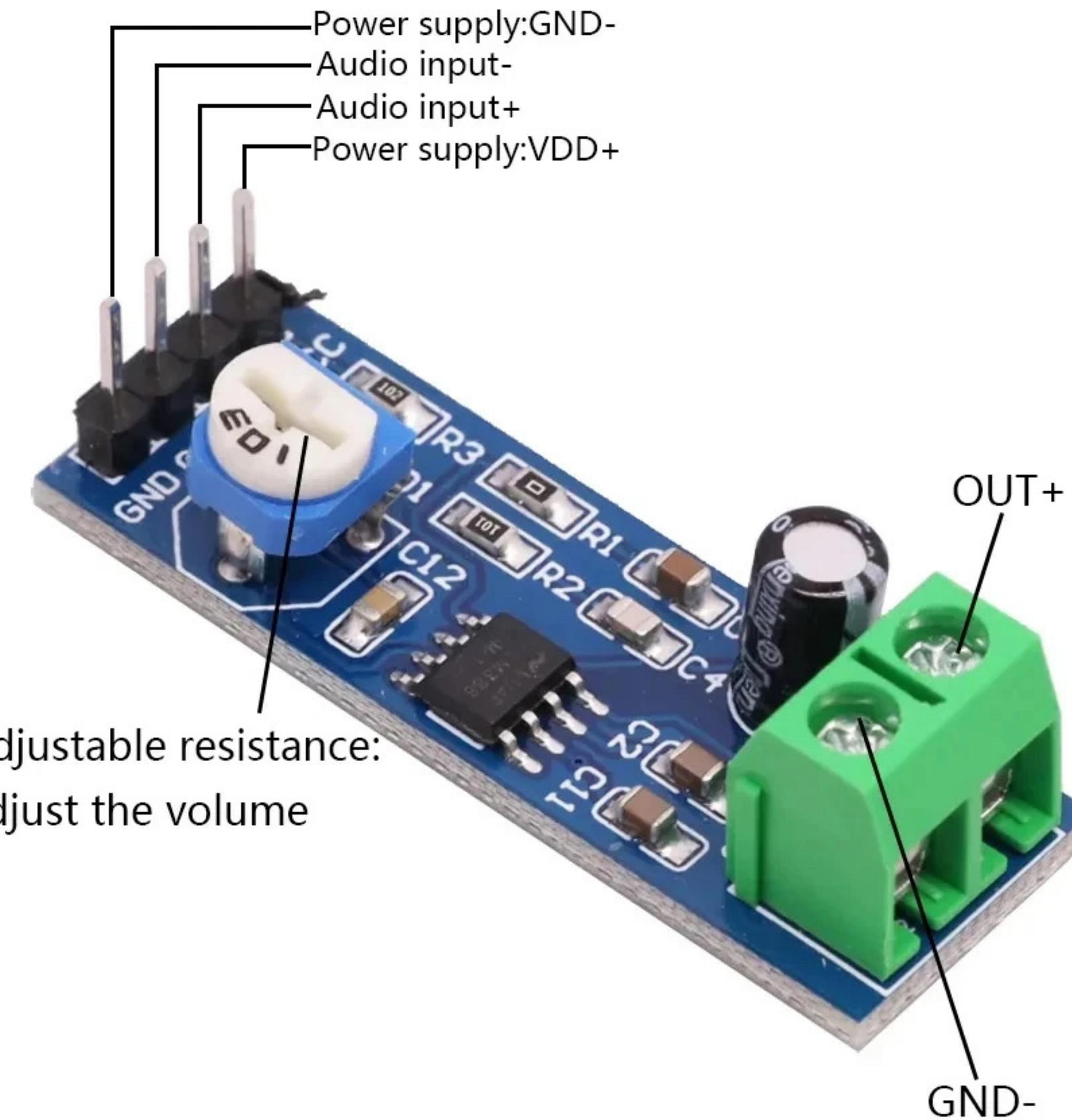
DIGITAL MULTIMETER

- = a device used to measure:
 - electric current (amps)
 - voltage (volts)
 - resistance (ohms)
- = great for troubleshooting
- = capable of measuring both AC and DC voltage



AMPLIFIER MODULE

- on-board LM386 Chip
- on-board speaker wire holder
- on-board 10K variable resistor, you can adjust the amplification volume



WIRE CUTTERS & STRIPPERS

- = designing your preferred cable length
- = essential for stripping off the insulation layer of the wire

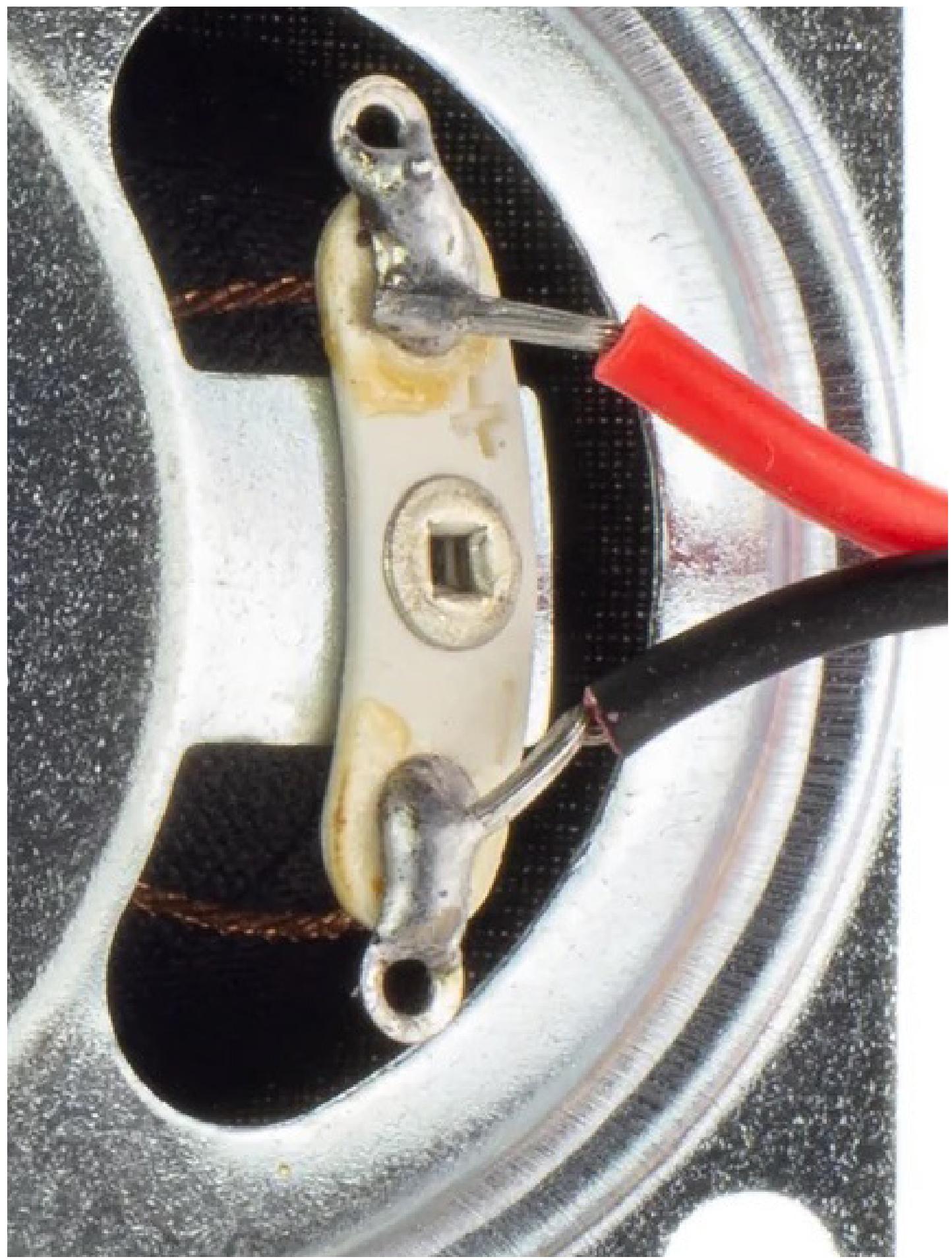
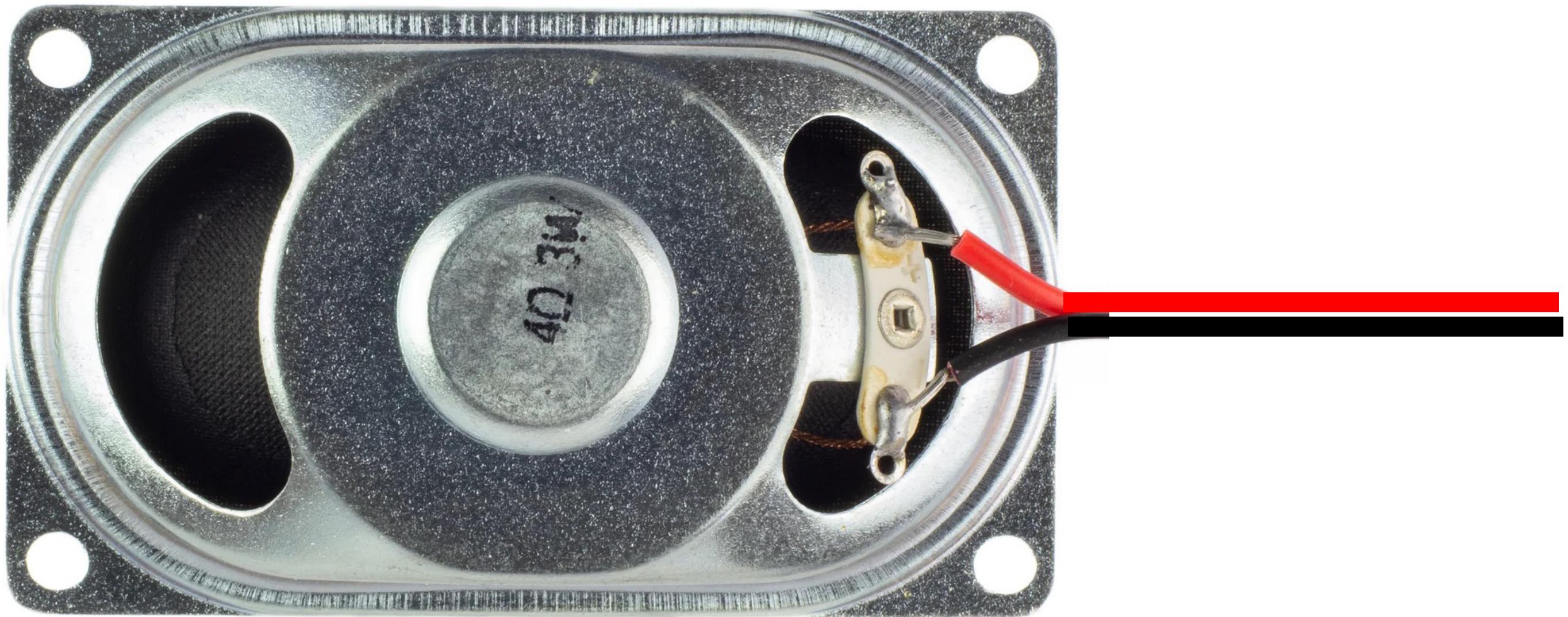


HELPING 3rd HAND

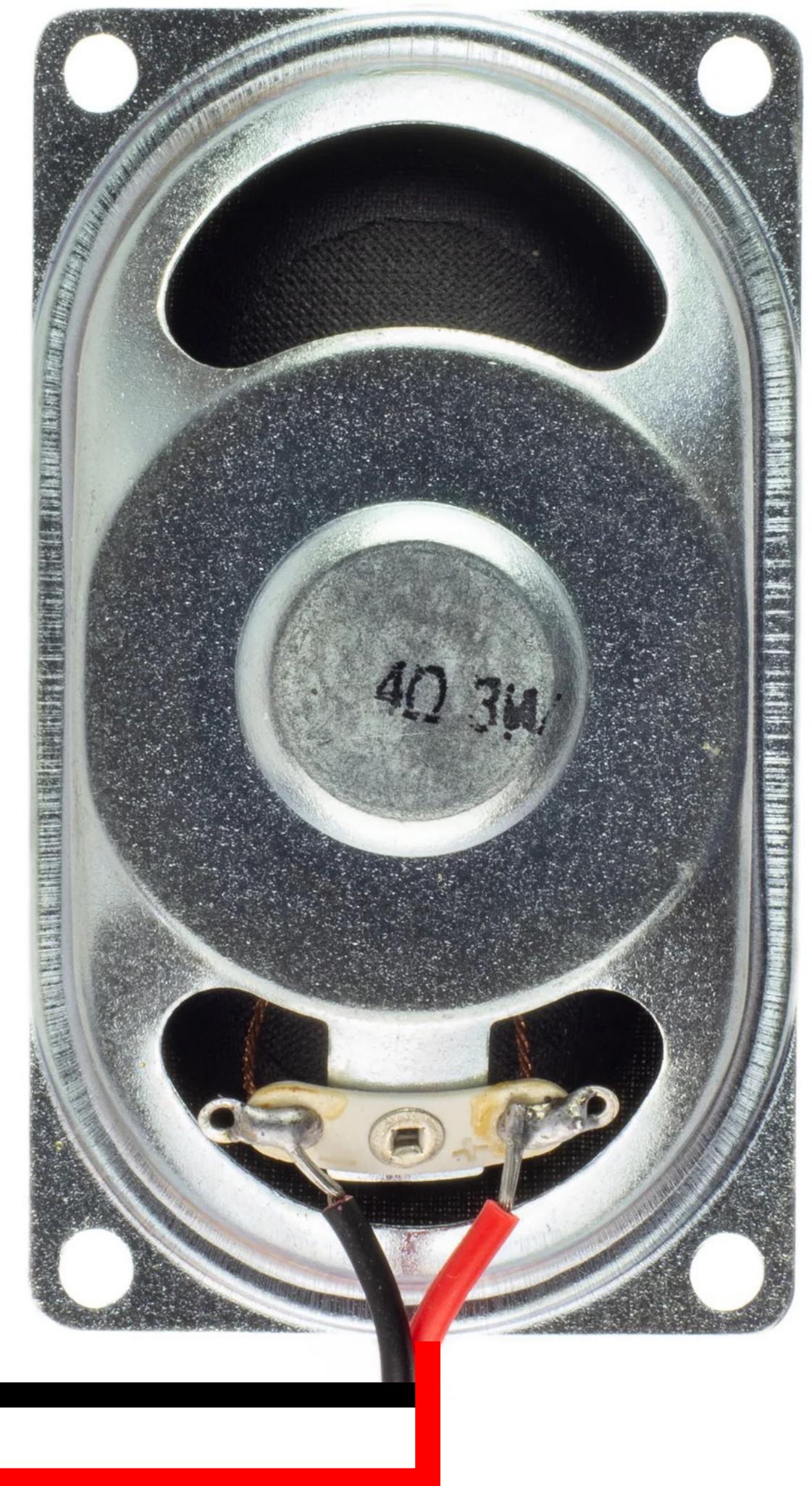
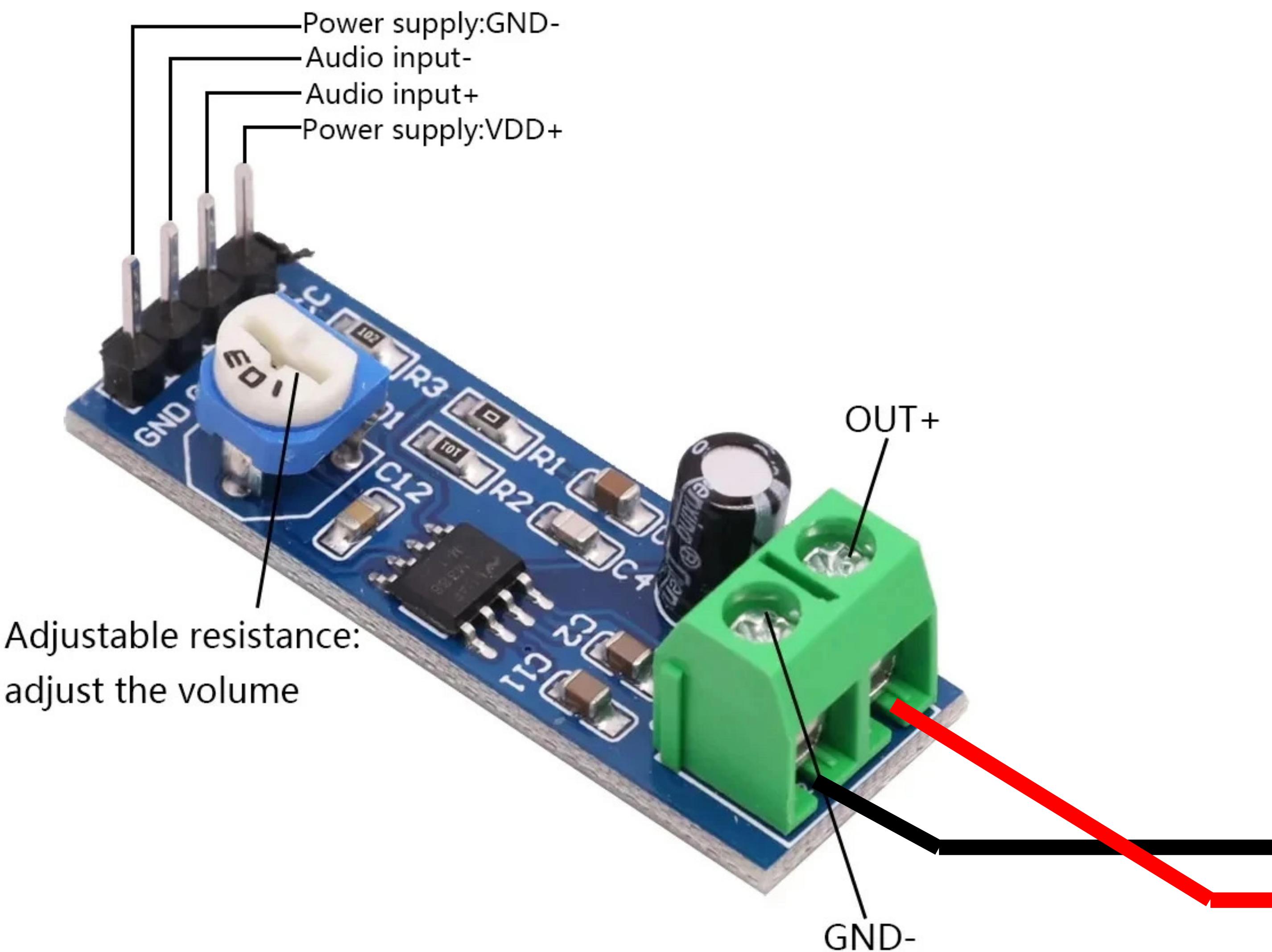
= for holding circuit boards or
wire when soldering or tinning



soldering wires to the speaker



connect the speaker to the amplifier



Soldering the PINs on ESP32

solder joint check:

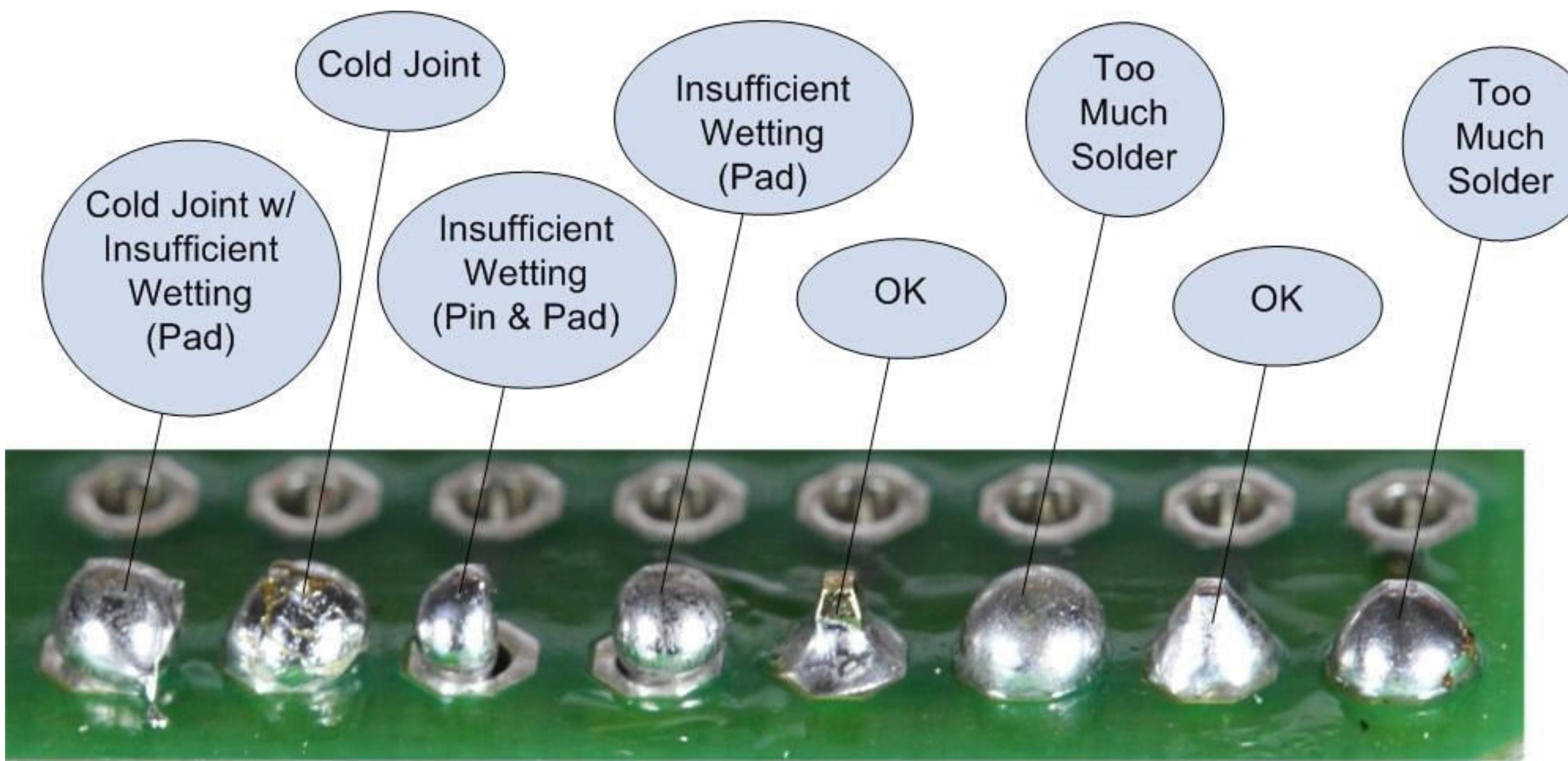
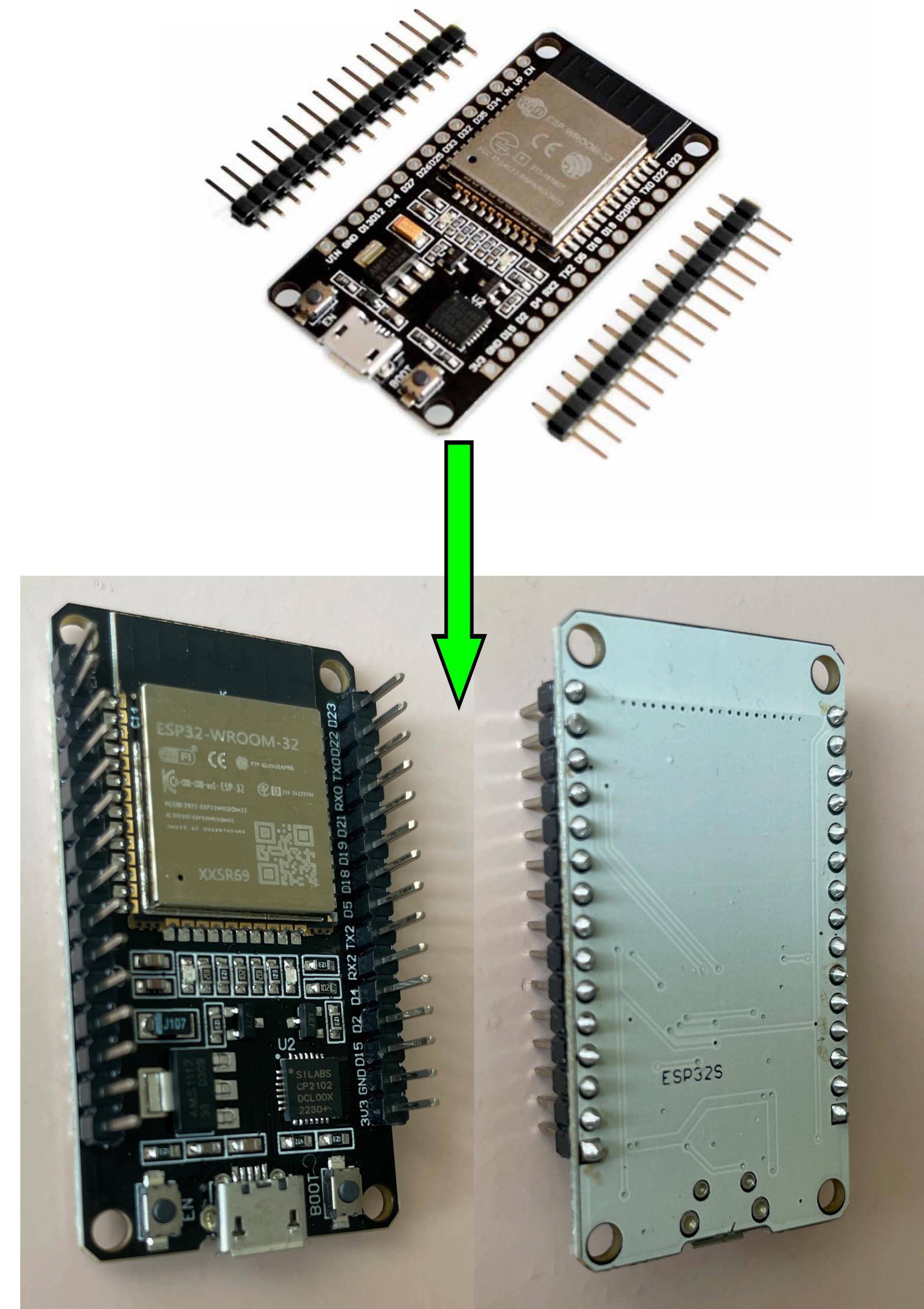


Image source: <https://learn.adafruit.com/adafruit-guide-excellent-soldering/common-problems>



Installing Thonny IDE

- Thonny = IDE for Python and MicroPython
- IDE = Integrated Development Environment
 - application that helps coding efficiently



Python = interpreted programming language

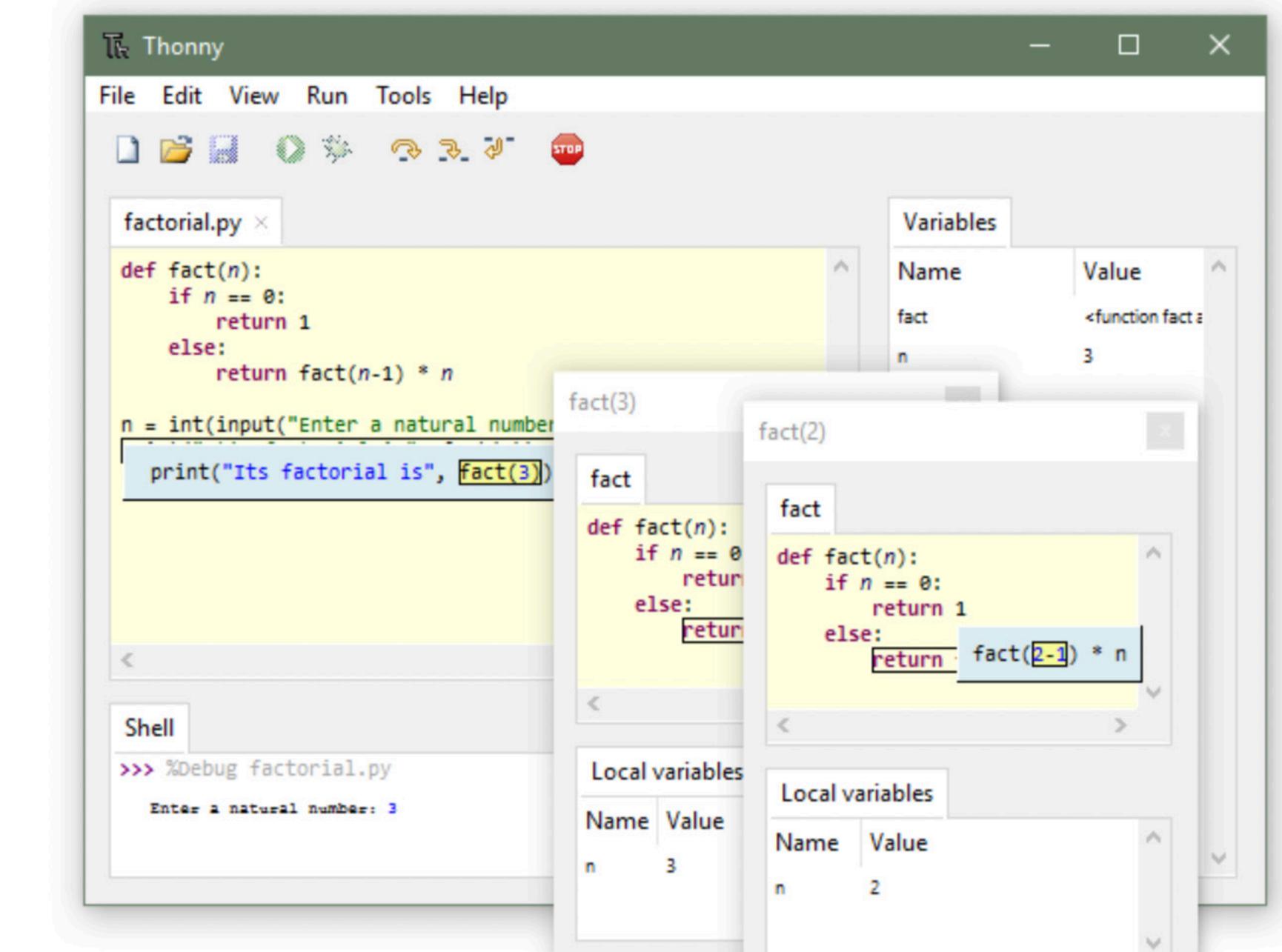
- uses significant indentation
- open source, packages <https://pypi.org>

- **MicroPython** = lightweight version of Python
 - optimized for microcontrollers

thonny.org

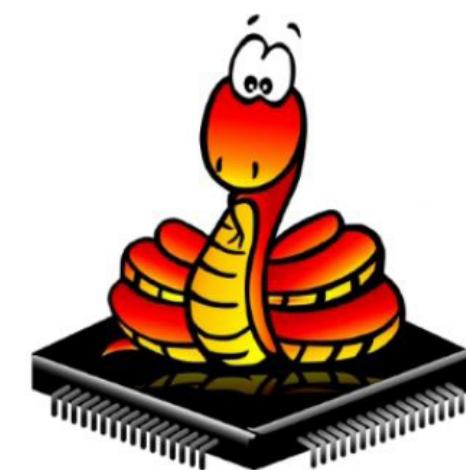
Thonny
Python IDE for beginners

 Download version [4.1.4](#) for
Windows • Mac • Linux



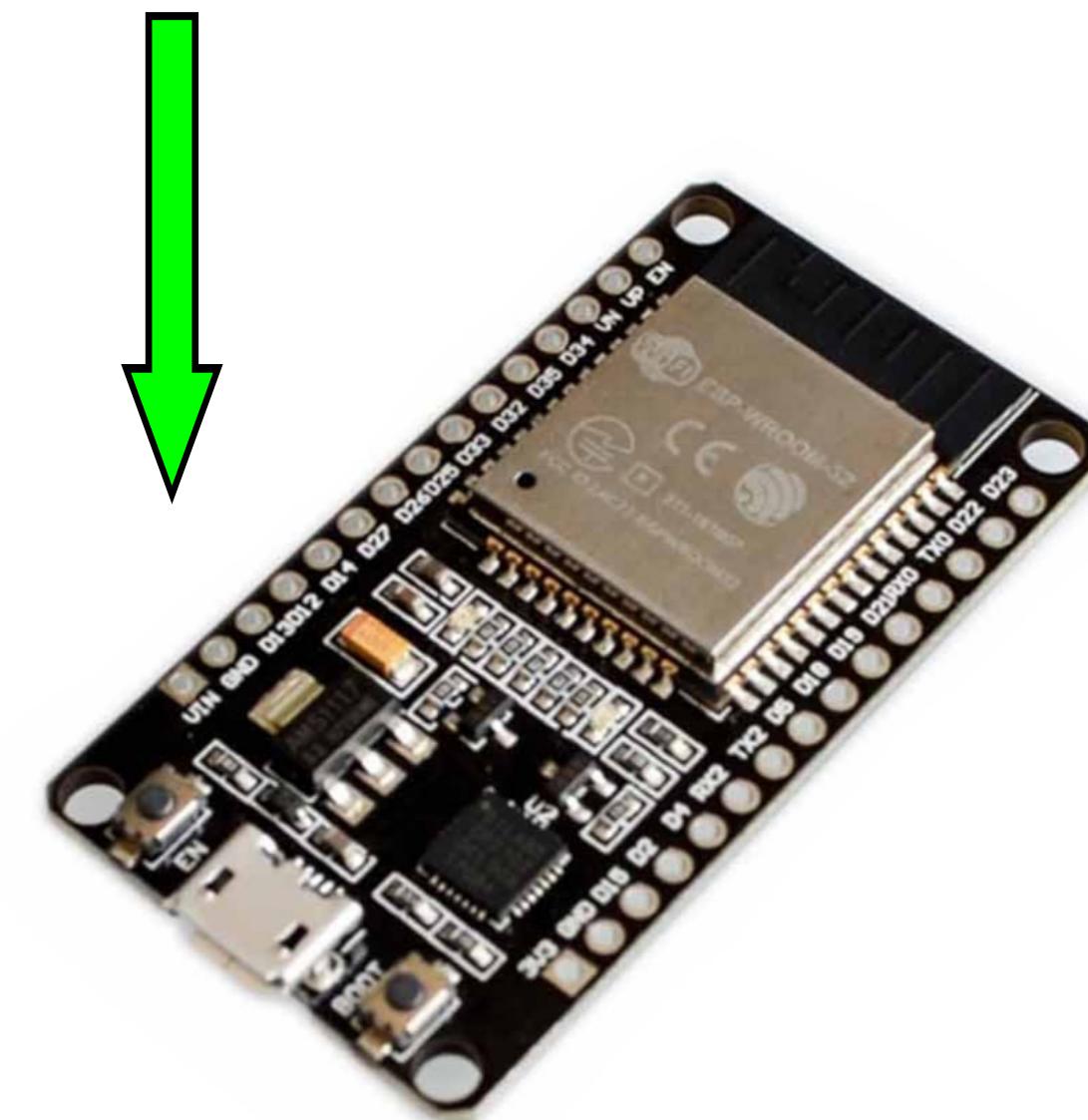
Installing MicroPython on ESP32

<https://micropython.org>



MicroPython

- 1) go to “Downloads”
- 2) select the ESP32 WROOM board from the list
- 3) download MicroPython as a .zip file



Installing MicroPython on ESP32

!!!

**WHEN INSTALLING THE FIRMWARE:
DISCONNECT ALL THE WIRES FROM THE
ESP32 PINS**

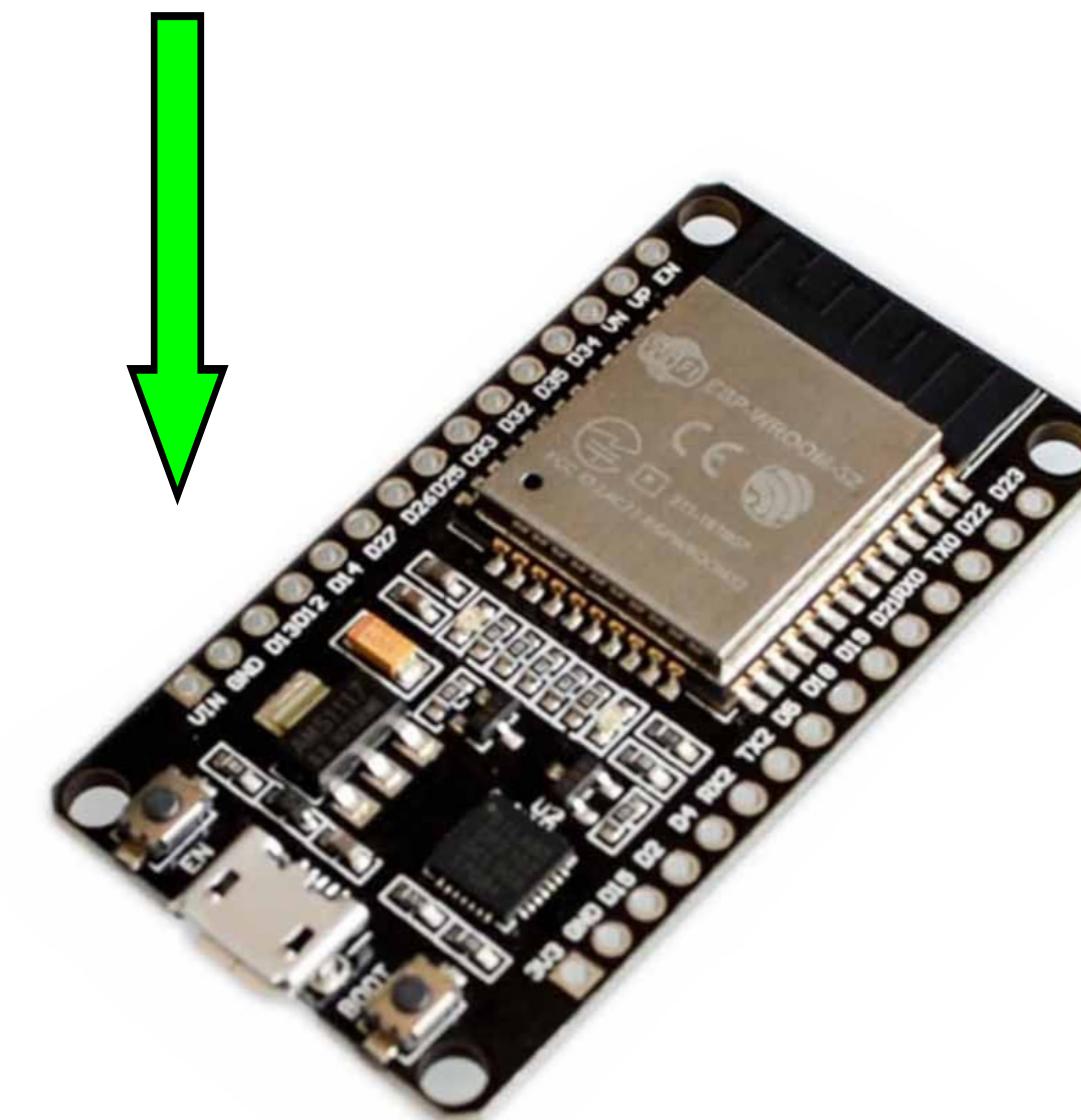
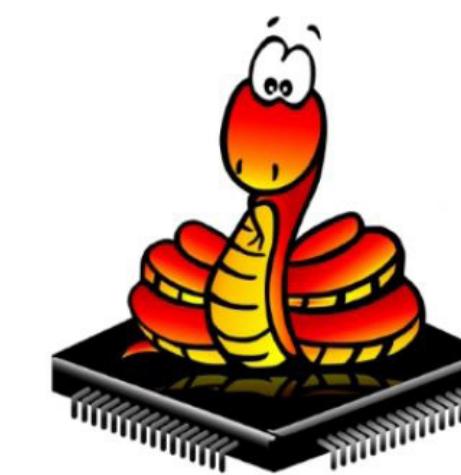
!!!

**WHEN CONNECTING WIRES / TOUCHING
PINS: ALWAYS DISCONNECT ESP32 FROM
POWER SOURCE**

!!!

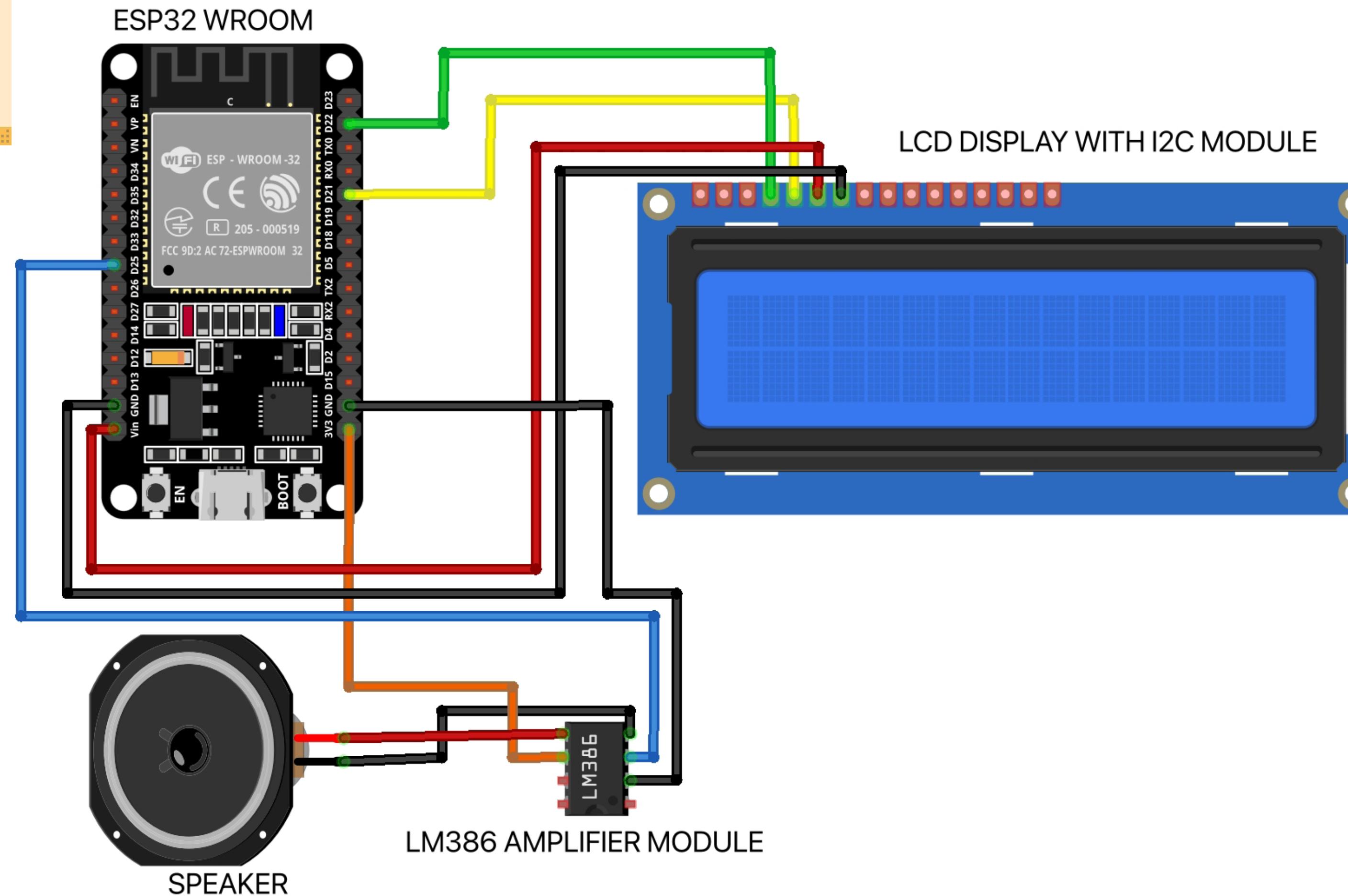


MicroPython

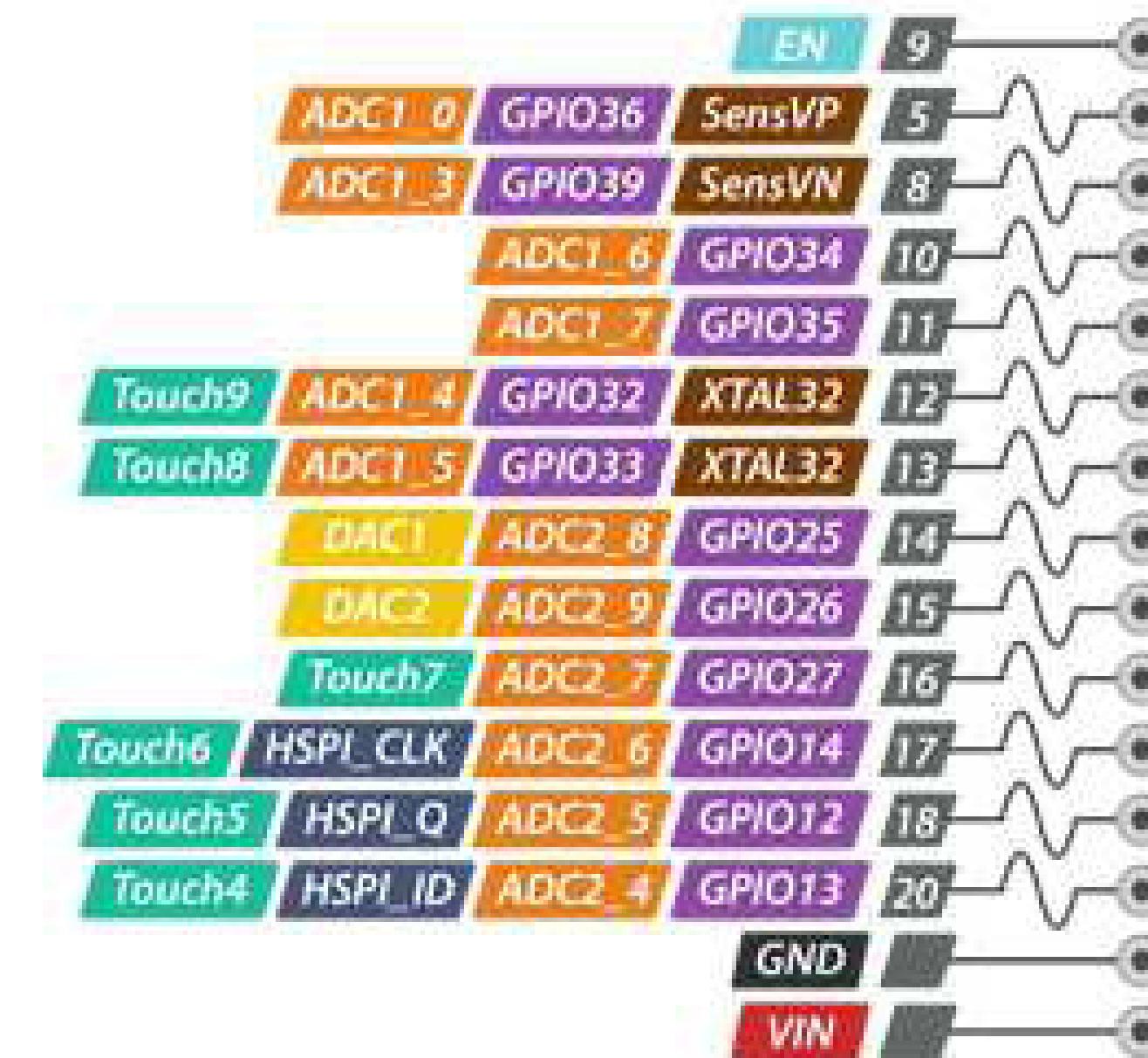


WIRING SCHEMATICS

Wiring diagram for
Open Hardware Workshop
2024 by Silvia Binda Heiserova



ESP32 PIN OUT



36	MOSI	GPIO23	V_SPI_D
39	SCL	GPIO22	V_SPI_WP
41	CLK3	GPIO1	U0_RXD
40	CLK2	GPIO3	U0_RXD
42	SDA	GPIO21	V_SPI_HD
38	MISO	GPIO19	V_SPI_Q
35	SCK	GPIO18	V_SPI_CLK
34		GPIO5	V_SPI_CS0
27		GPIO17	U2_RXD
25		GPIO16	U2_RXD
24		GPIO4	ADC2_0 / HSPI_HD / Touch0
22	CS	GPIO2	ADC2_2 / HSPI_WP / Touch2
21		GPIO15	ADC2_3 / HSPI_CS0 / Touch3
		GND	
		3.3V	

Power Control Arduino ADC SPI EN
 GND Touch GPIO DAC UART PWM

Uploading Libraries to ESP32

LIBRARIES

- = collections of pre-written code and functions
- extend the capabilities of Python/MicroPython
- provide a wide range of tools and modules
- making it easier to work on specific tasks without reinventing the wheel.

1) “i2c_lcd.py”

- = library for running the display
- copy it and save to ESP32)

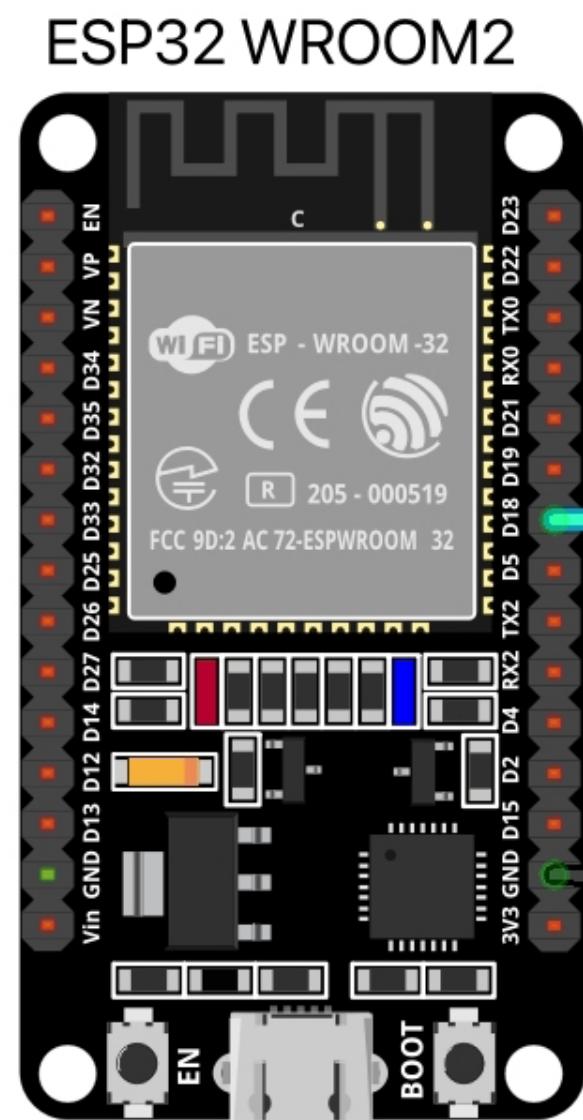
2) “lcd_api.py”

- = api for lcd display
- copy it and save to ESP32

for trying at home:

simple LED (= Light Emitting Diode)

+ 220 ohm resistor !



RESOURCES:

- <https://www.makerspaces.com/basic-electronics/>
- <https://opencourse.com>
- <https://www.instructables.com/Basic-Electronics/>
- <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-ground/>
- <https://www.circuitbasics.com/introduction-to-microcontrollers/>

CONTACT:

silviaheiserova@gmail.com