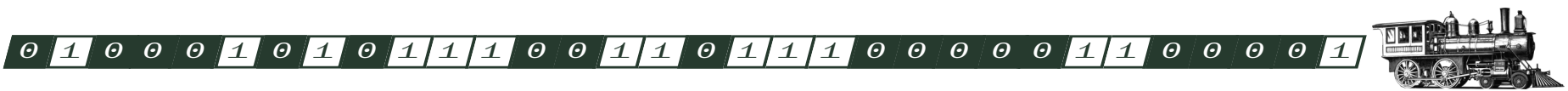
Concevoir une interface



Lucas Sifoni / Turing-Express · Espace Brut

contact@turing-express.fr

Code complet disponible le 25 octobre : github.com/lucassifoni

Site perso (articles, confs) -> lucassifoni.info
Développement indépendant -> turing-express.fr
Logiciel pour acteurs de l'archi/bâtiment -> alzo.fr
Métrologie optique -> chantepierre-optique.fr

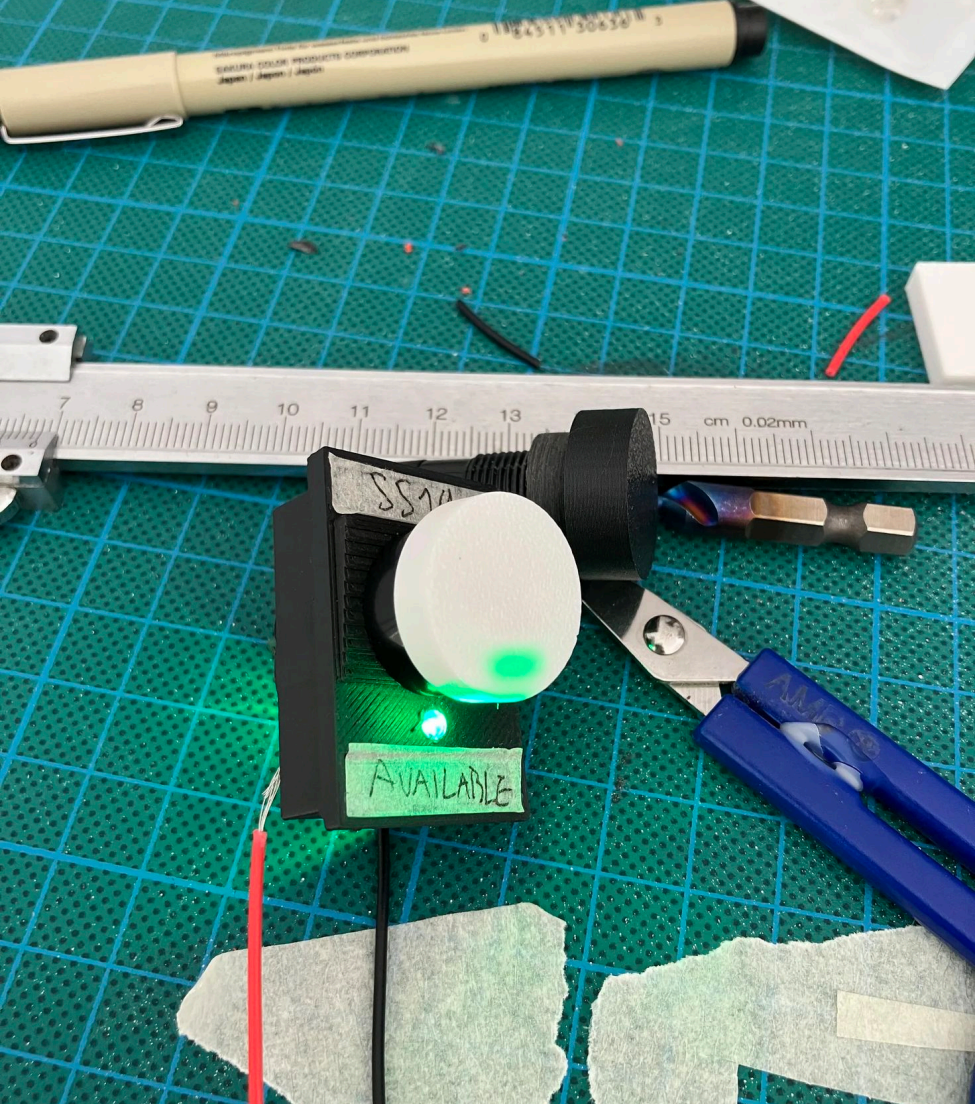
Fabrication matérielle

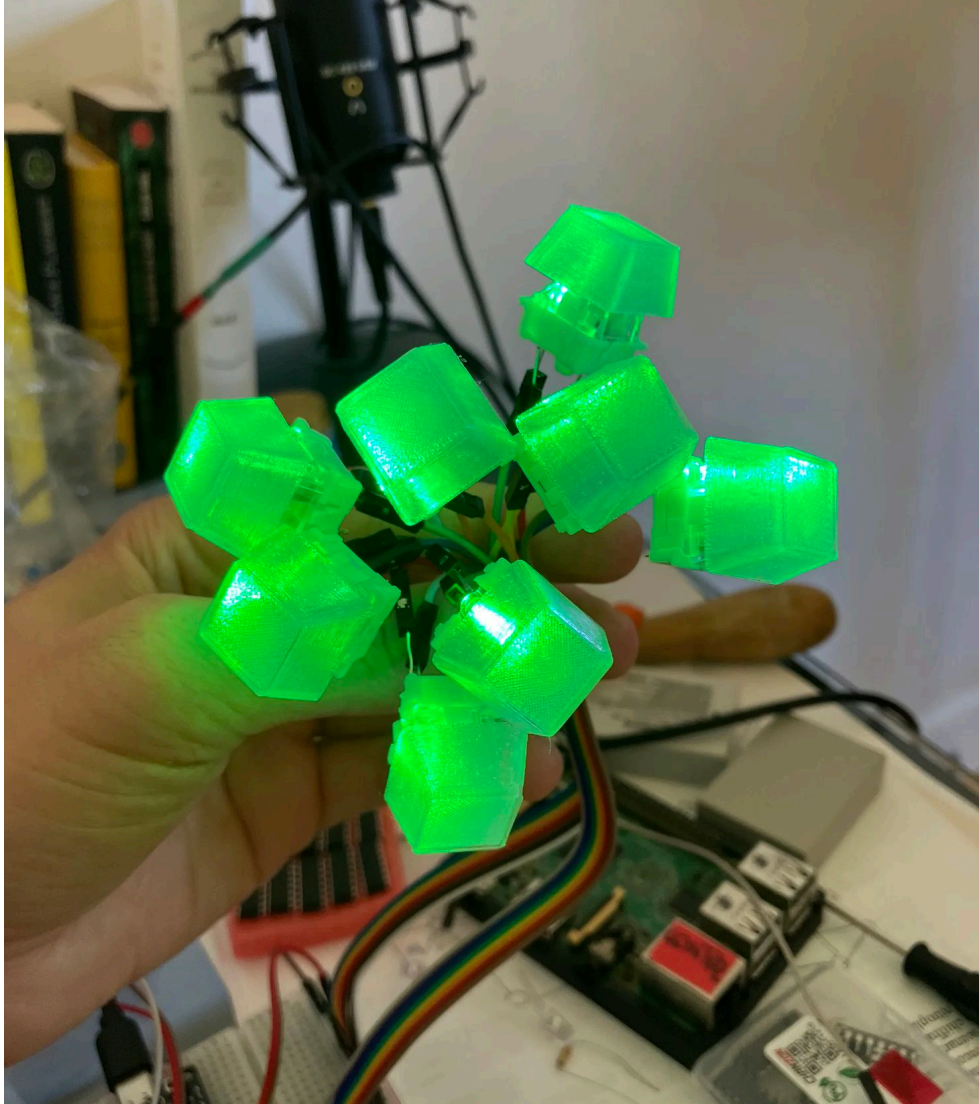
- 4 modules indépendants
- pas de PCB
- fabriquer soi-même
- impression 3D, découpe laser, composants DIP / TH et padboards
- old-school

4 modules indépendants

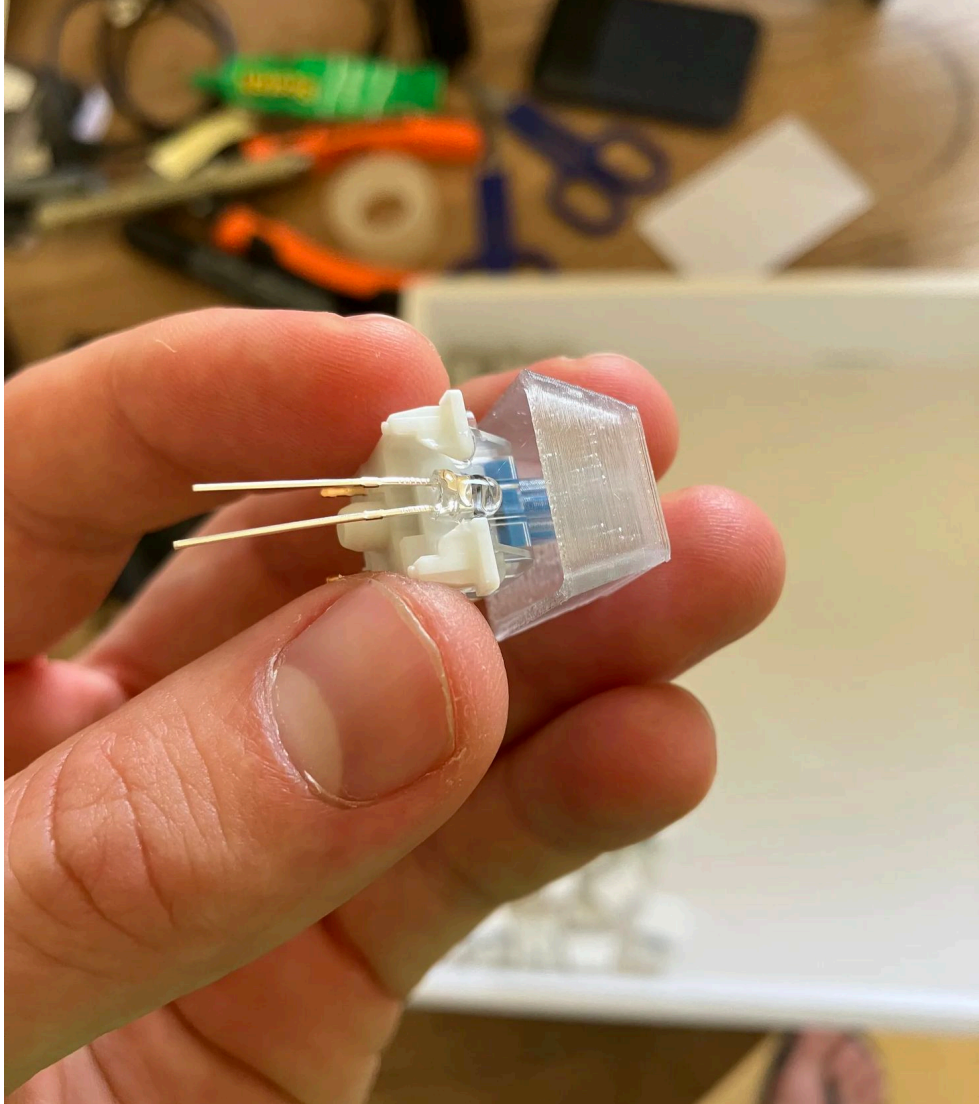


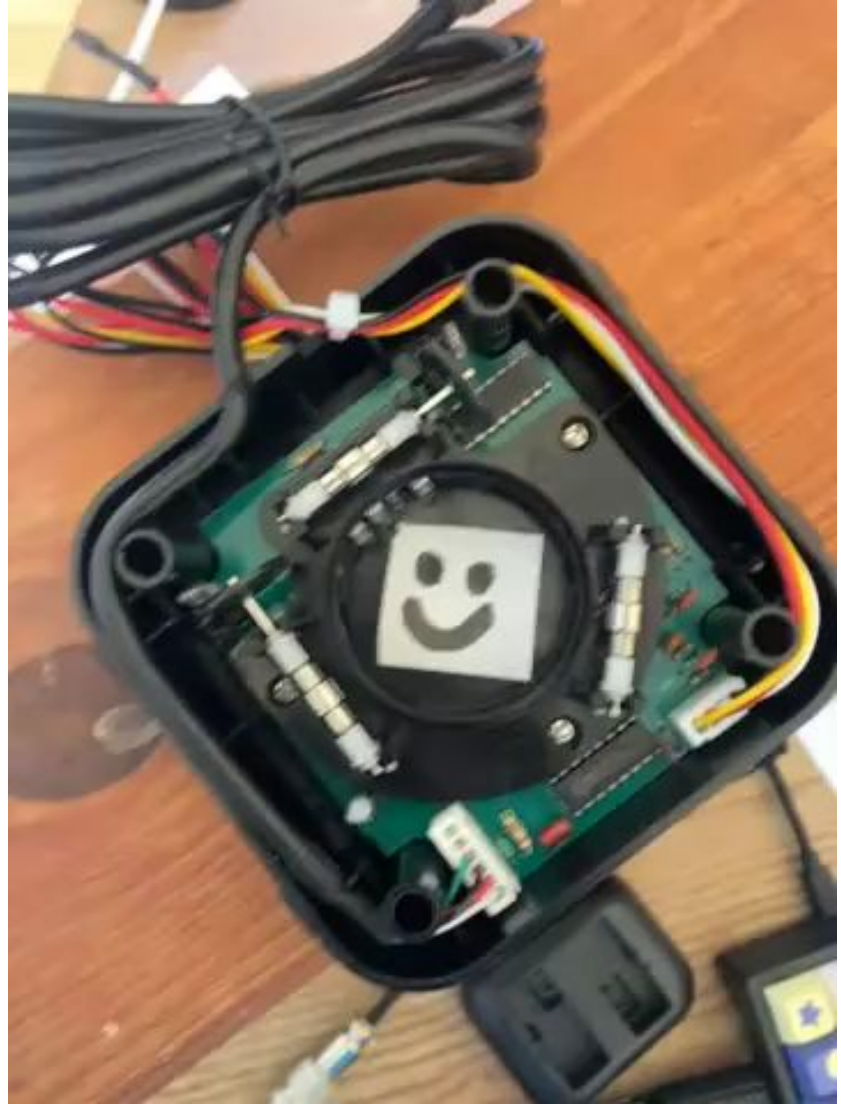

Integrated Circuit
P/N:74HC595N
PKG CASE:DIP-16
QTY:10
CHANZON
W57-1=10
10PCS ULN2803APG ULN2803A ULN2803 DI
P18 100% NEW
16

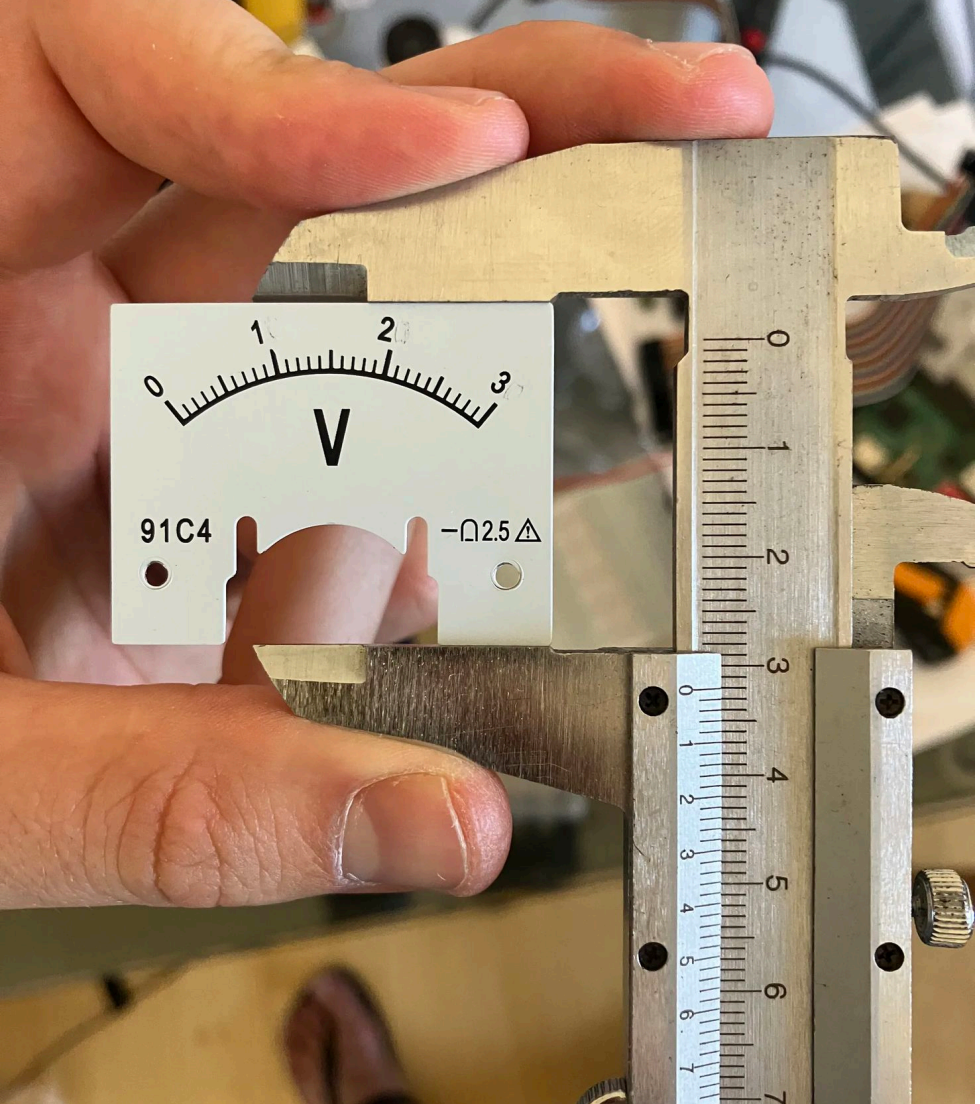

AVAILABLE

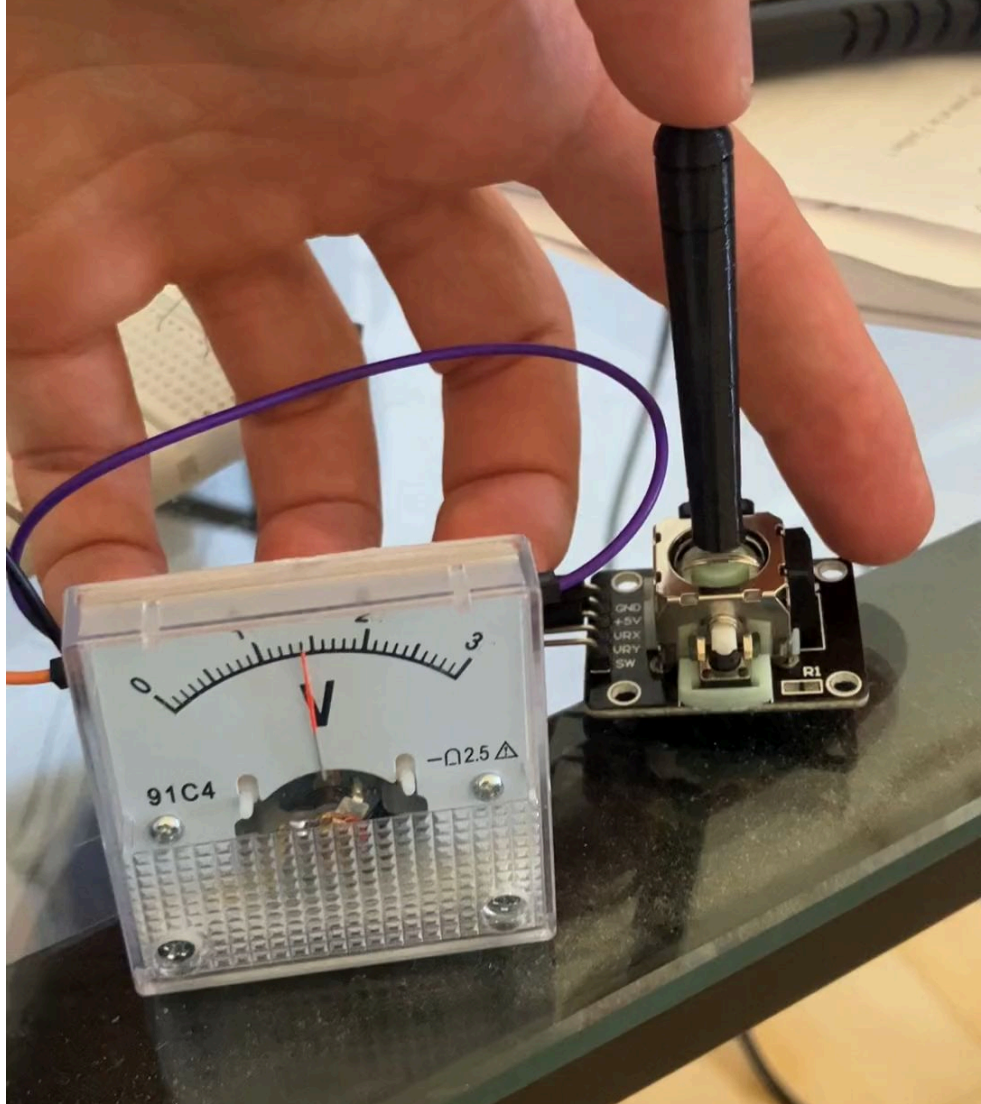


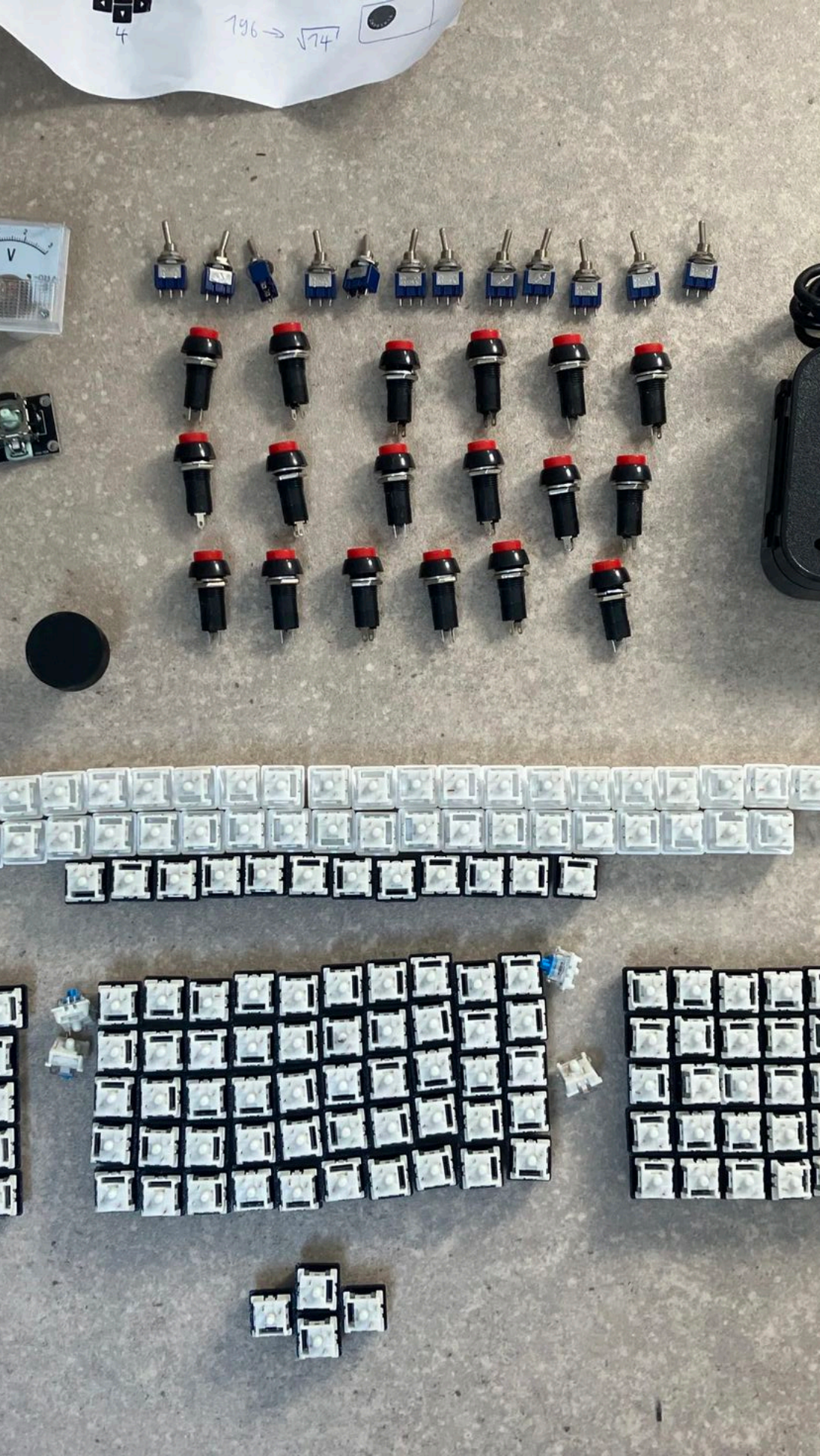




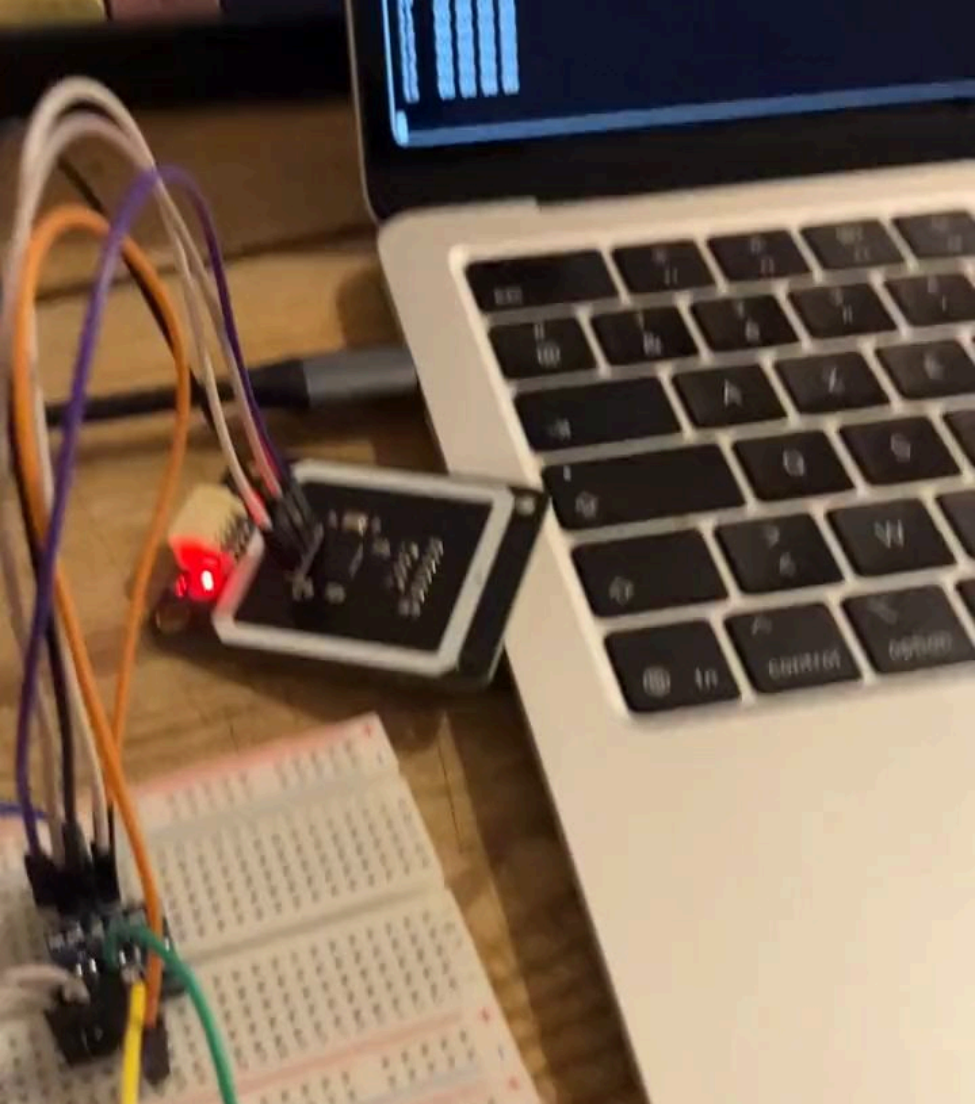


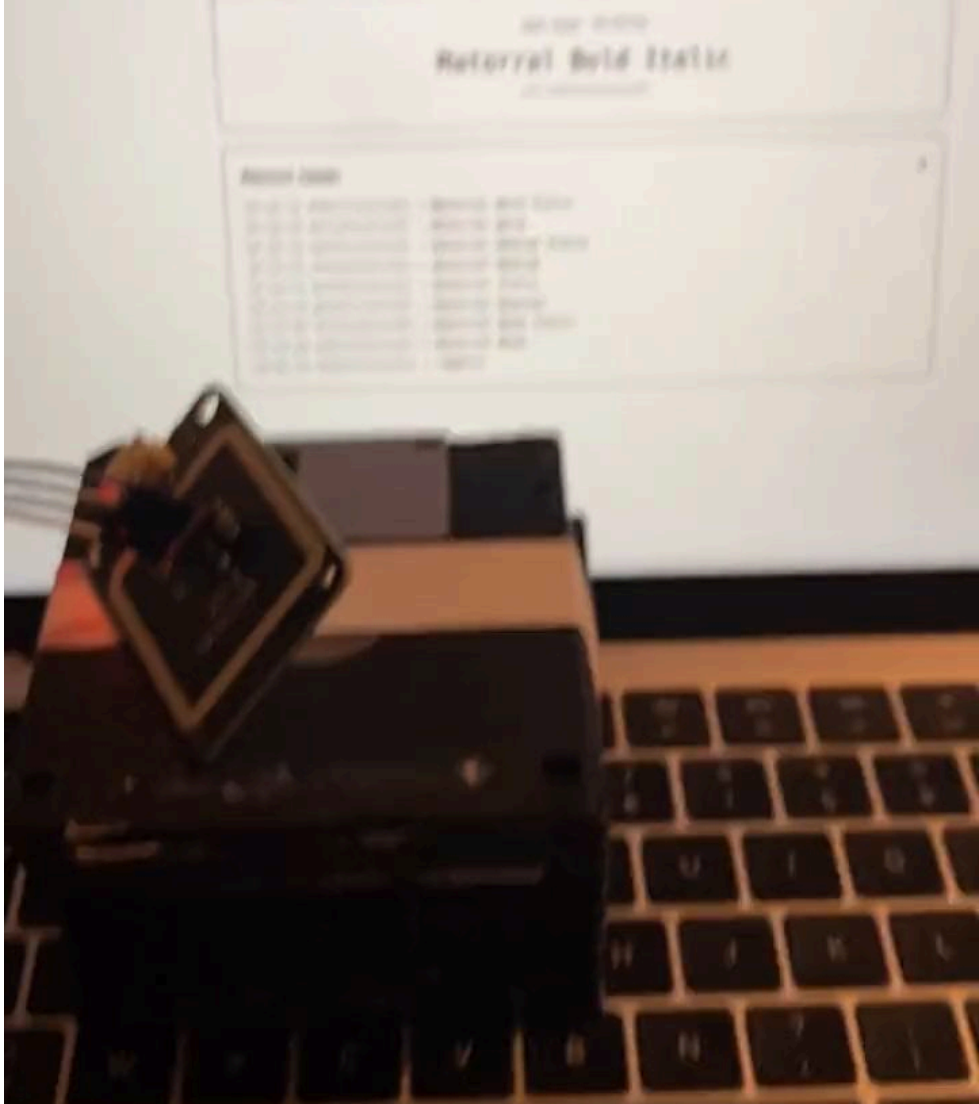

0
1
2
3
V
91C4
−Ω2.5 ⚠

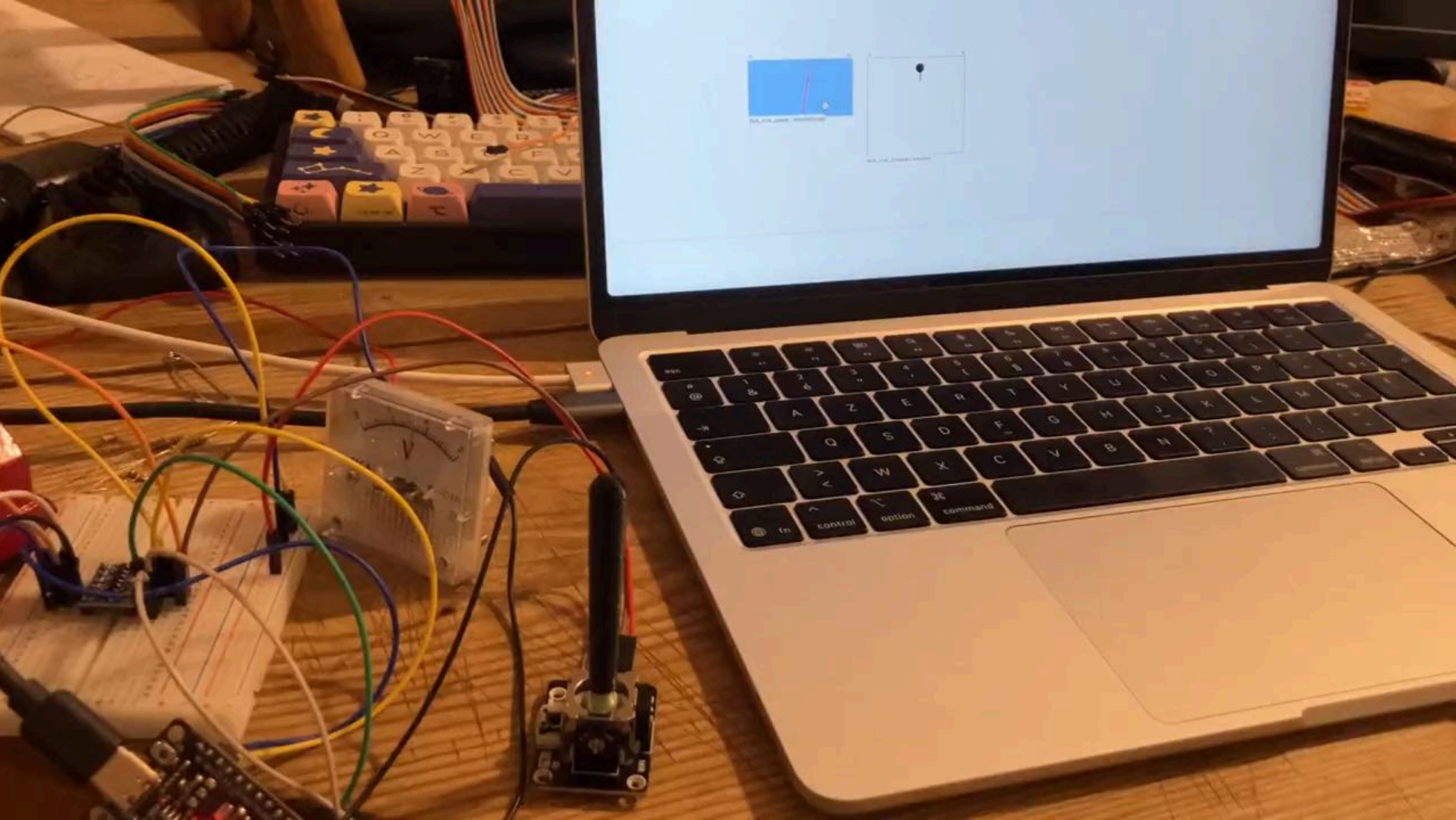

0
1
2
3
V
91C4
−Ω2.5 ⚠
GND
+5V
URX
URY
SW
R1

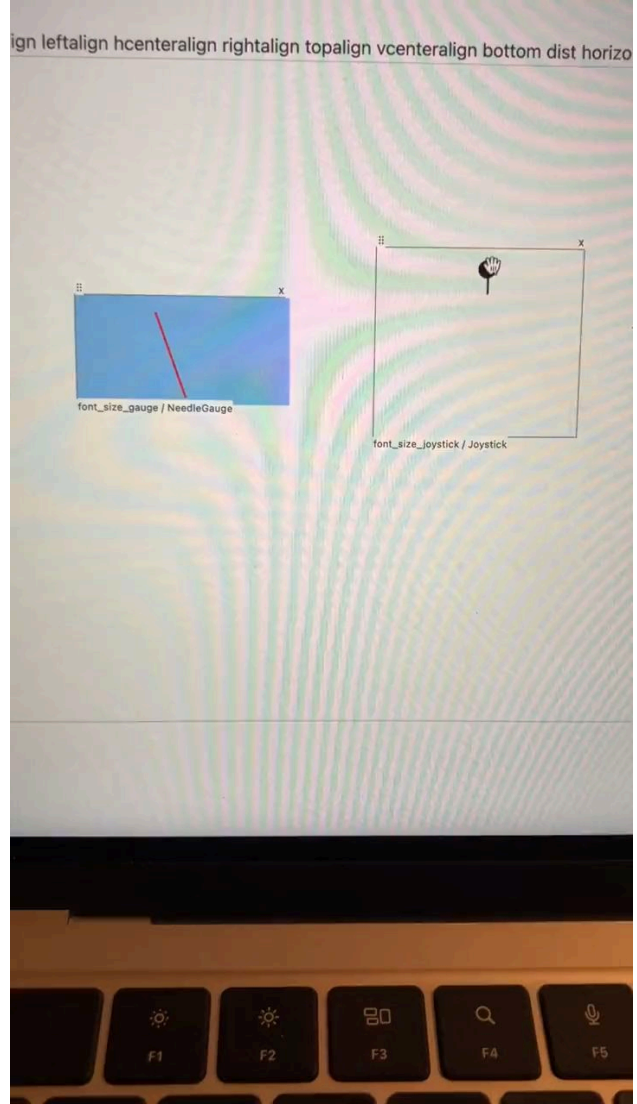

4
196 → √74
V

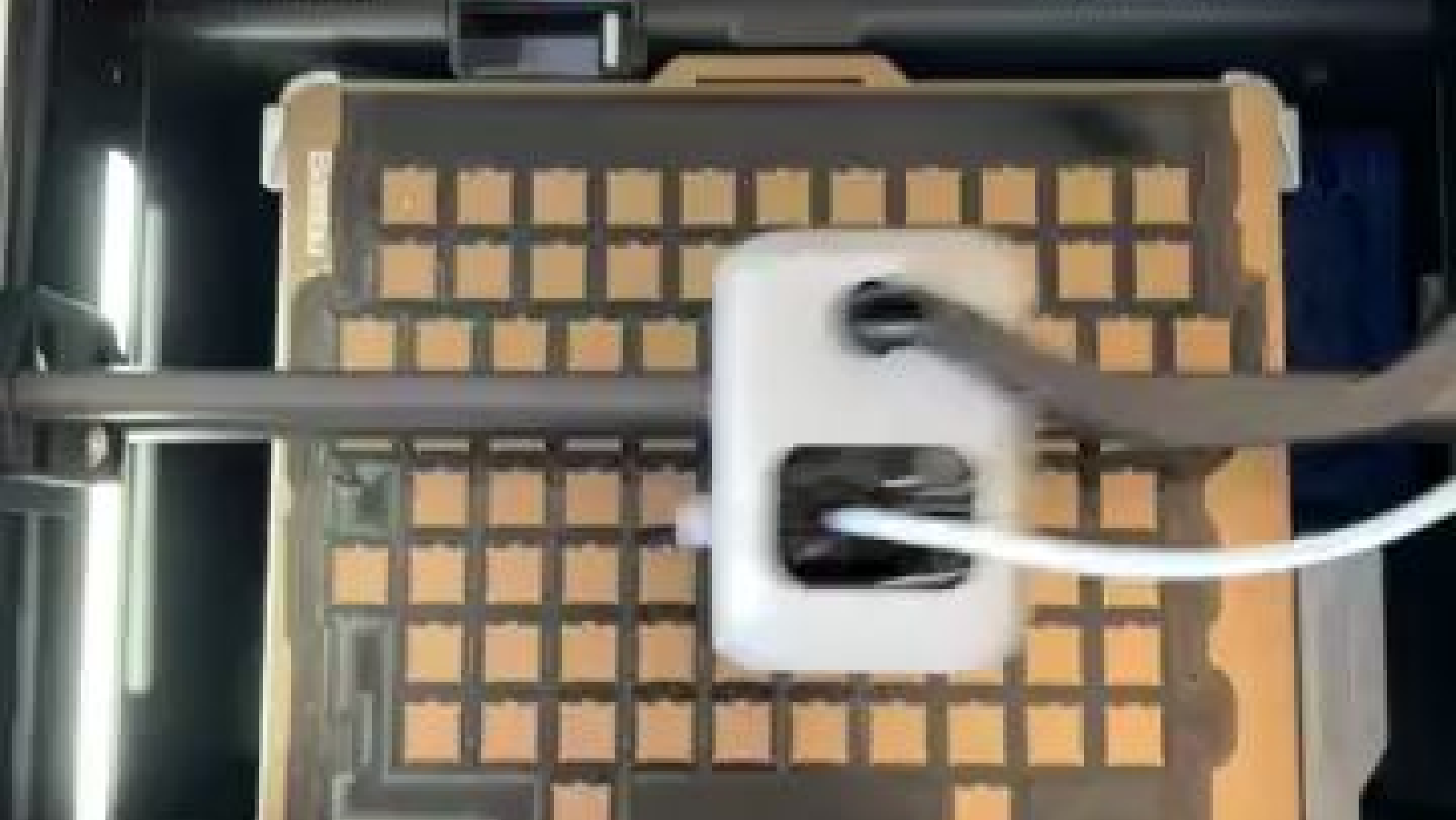




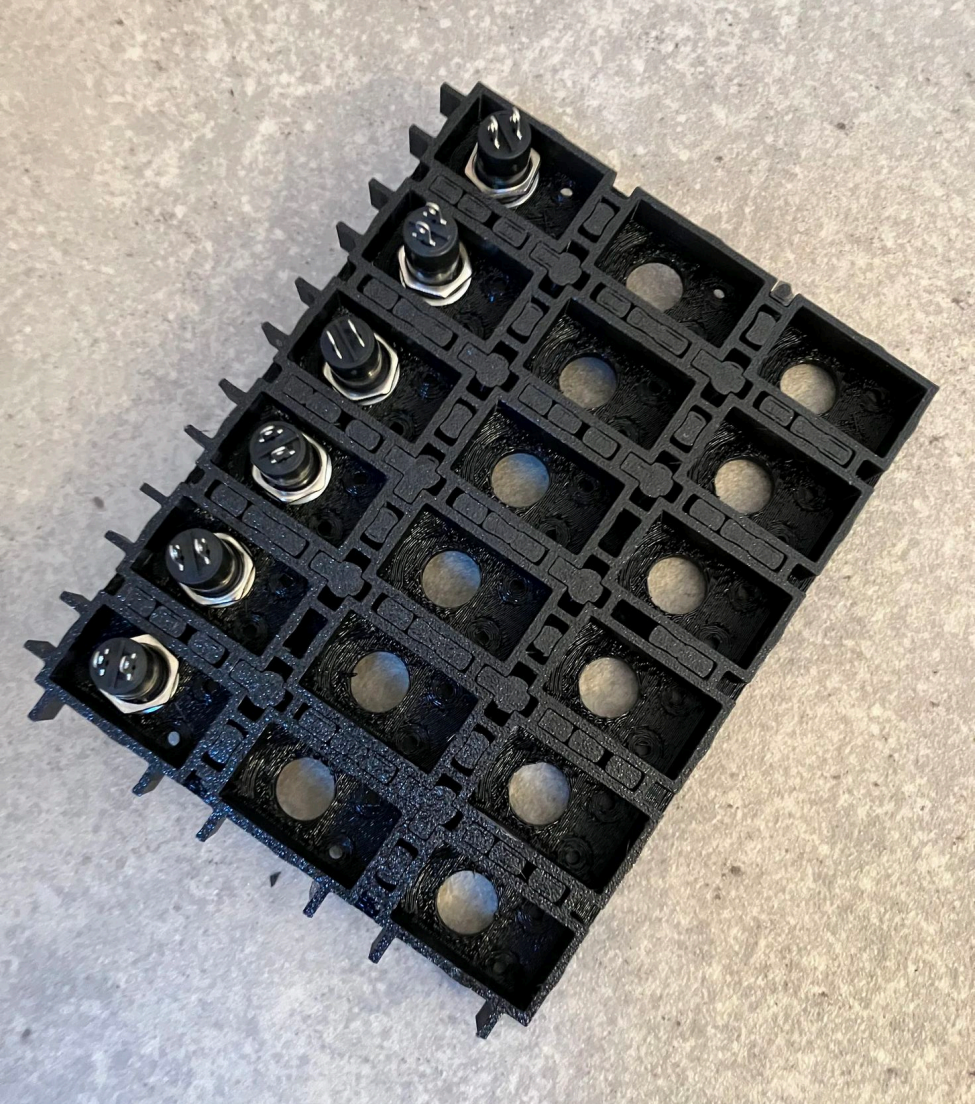


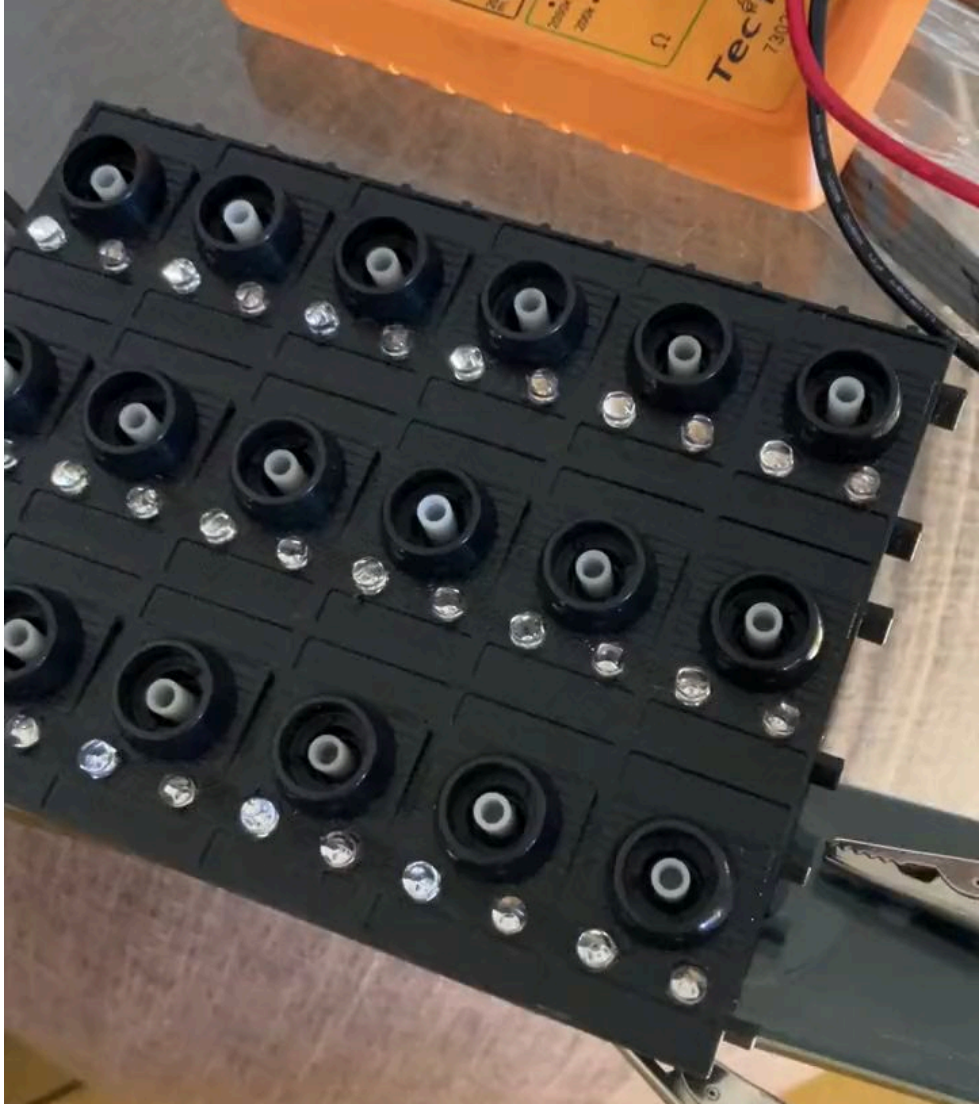

control
option
command


ign leftalign hcenteralign rightalign topalign vcenteralign bottom dist horizo
font_size_gauge / NeedleGauge
font_size_joystick / Joystick
F1
F2
F3
F4
F5





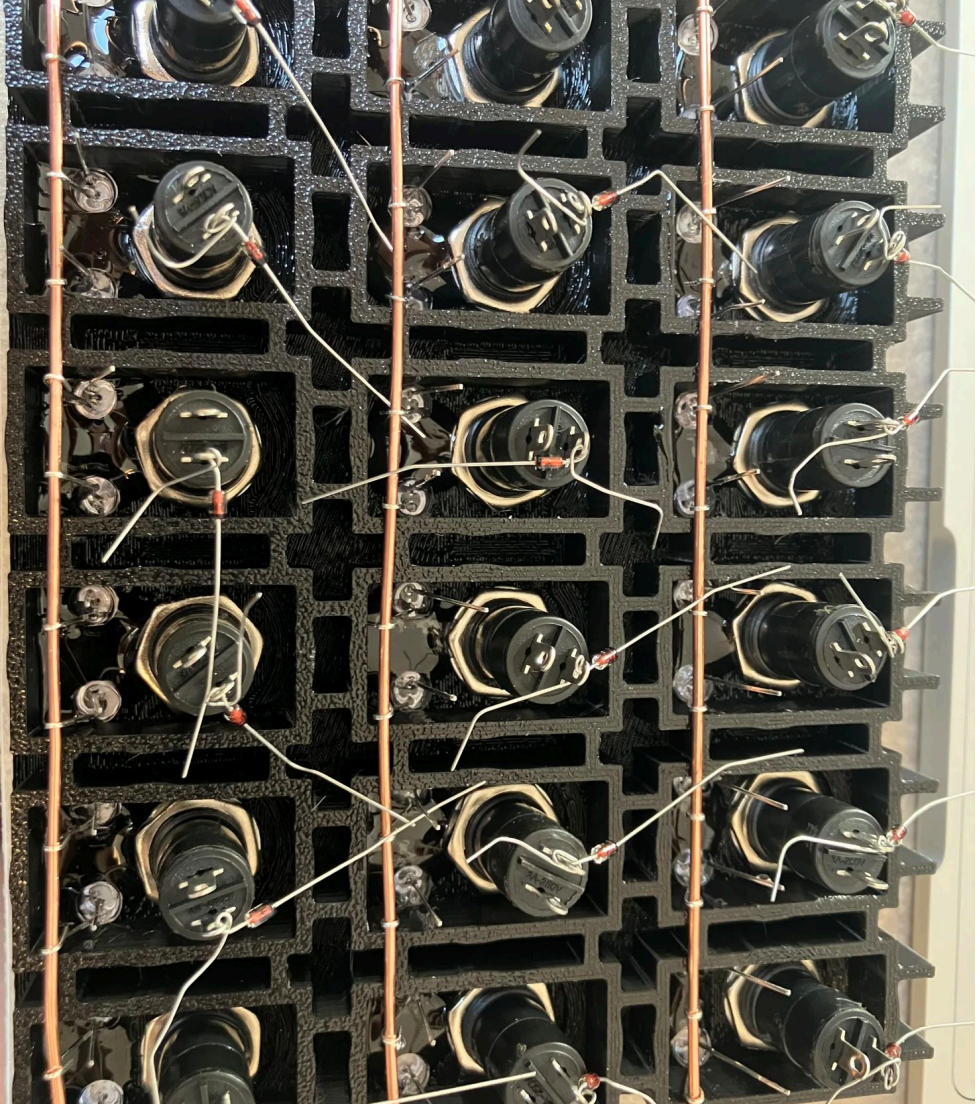


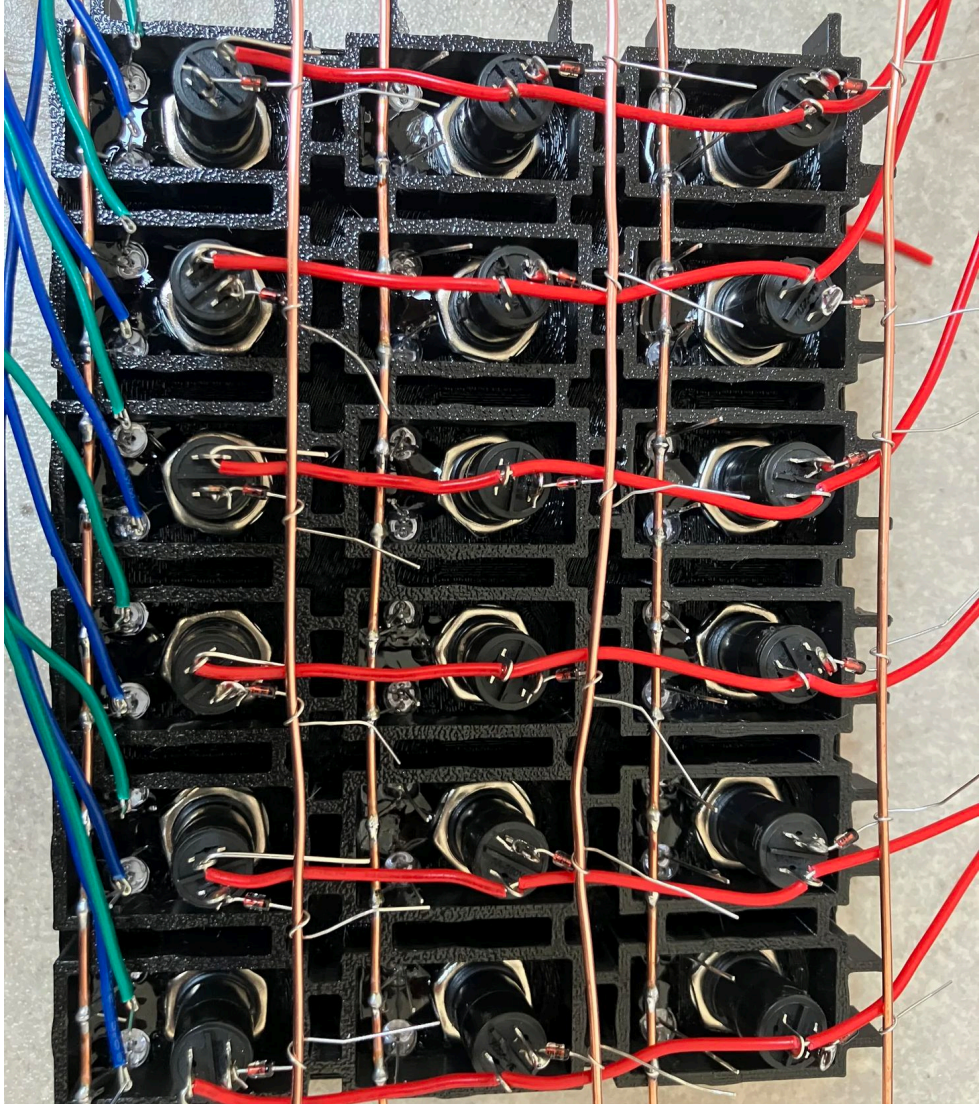




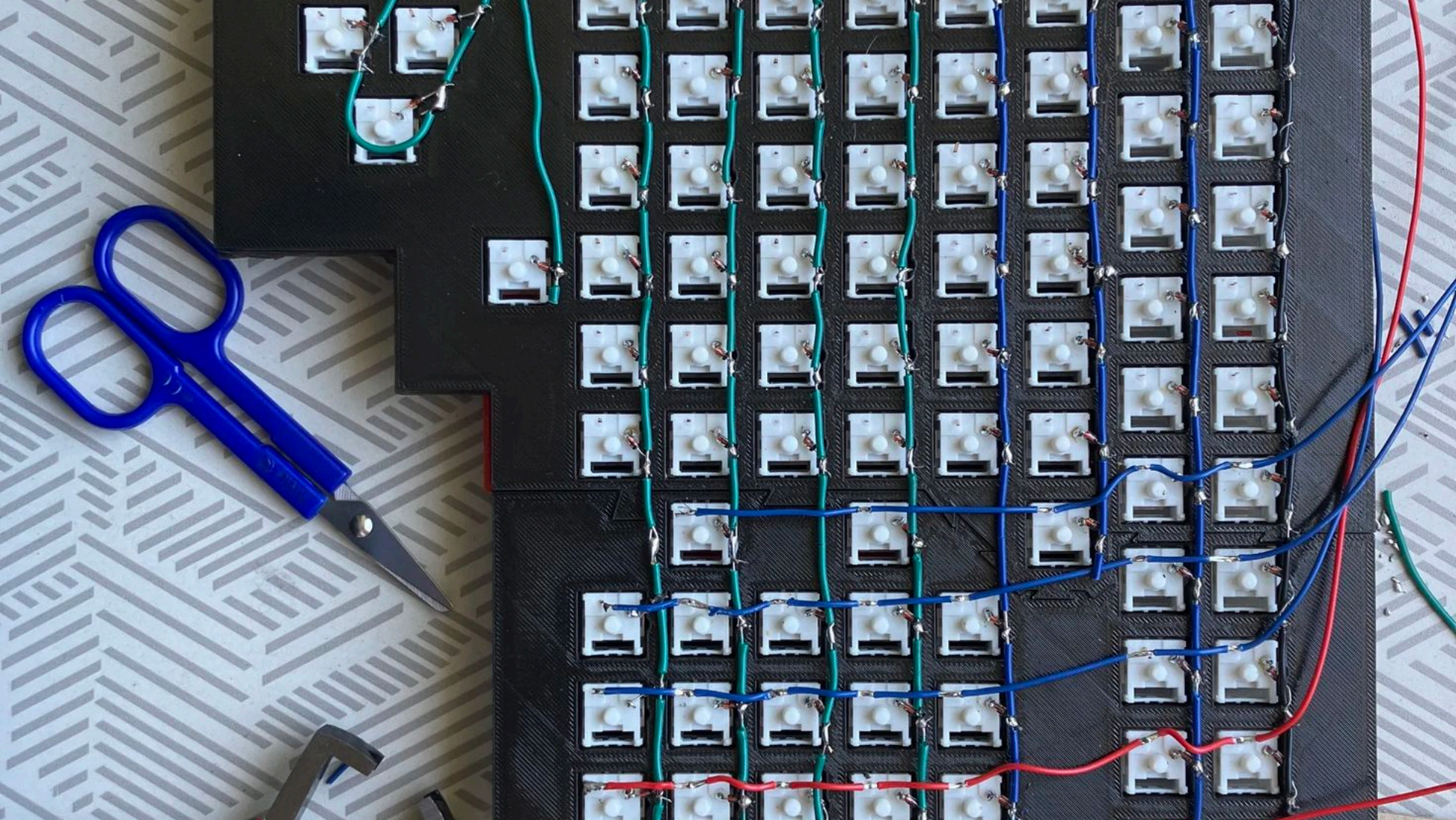


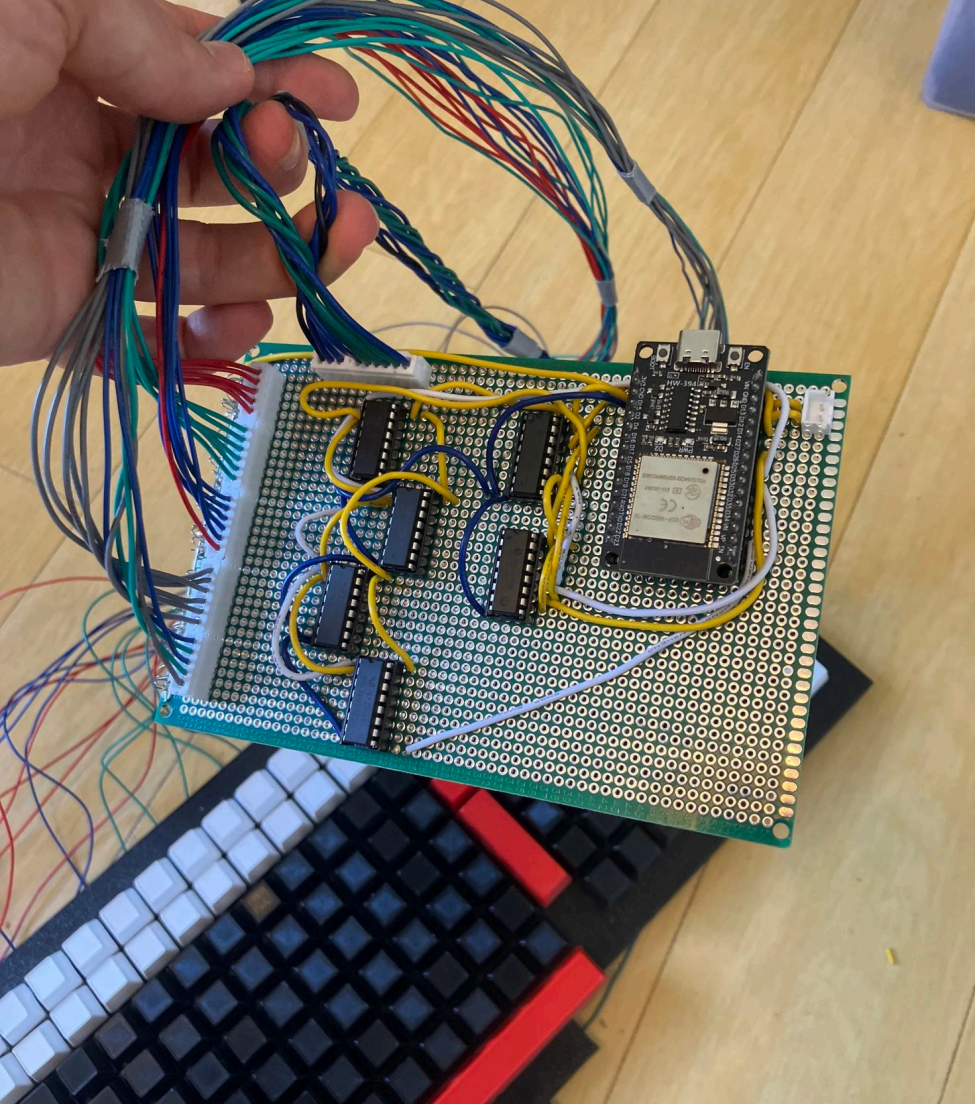


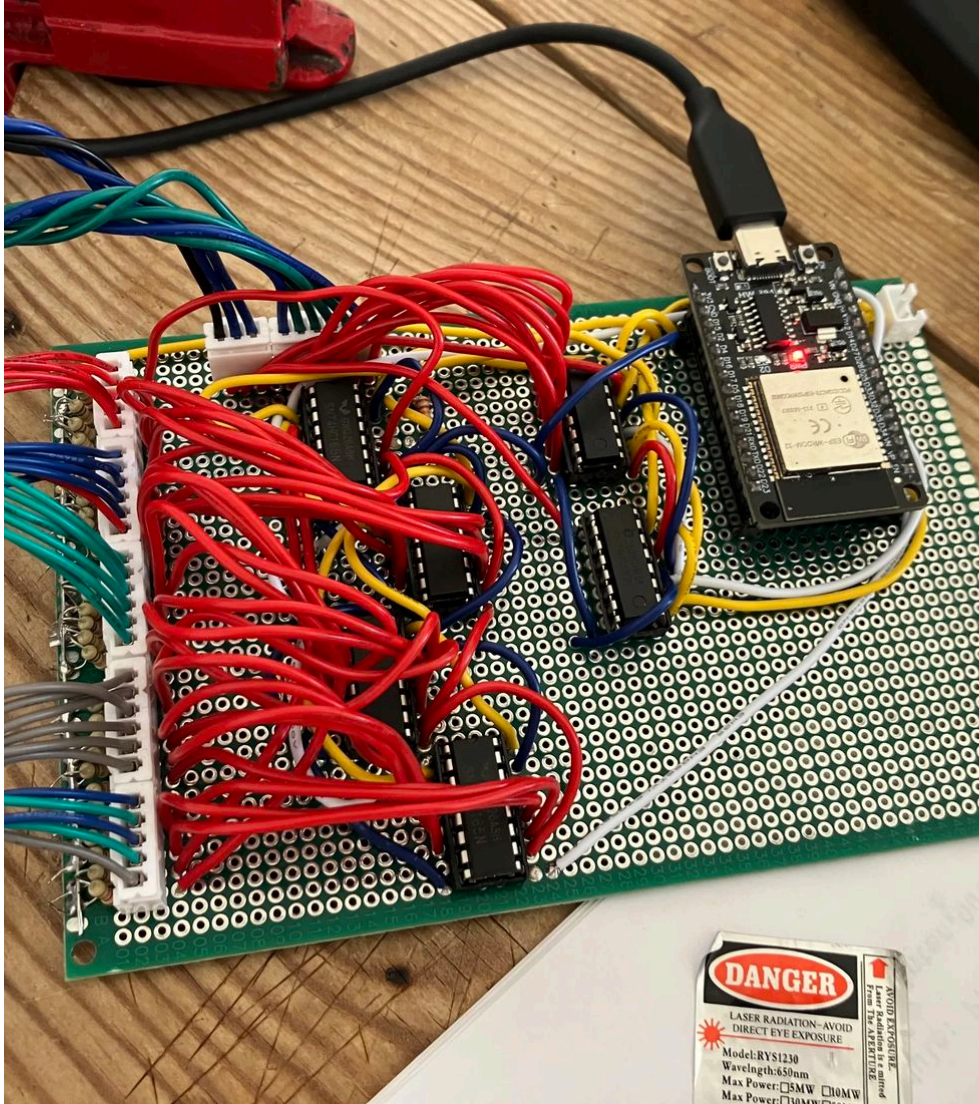





CD4049






DANGER
LASER RADIATION-AVOID DIRECT EYE EXPOSURE
Model:RYS1230
Wavelngth:650nm
Max Power:□5MW □10MW

STAND ALONE CONFIGURATION MODE MENU TABLE

NO	LED1(ON)	LED2(BLINK)	PRINTER OPERATION
1	GREEN	GREEN	PRINT CONFIGURATION INFO.
2	GREEN	ORANGE	PRINT FILE LIST
3	GREEN	RED	FACTORY RESET
4	ORANGE	GREEN	GAP – CALIBRATION MODE
5	ORANGE	ORANGE	B/M – CALIBRATION MODE
6	ORANGE	RED	START DATA – DUMP MODE
7	RED	GREEN	DEMO MODE

XPM	: DISABLE
REFLECT SENSOR	: H:2.06V, L:1.03V, D:1.03V, T:1.55V
TRANSMIT SENSOR	: H:2.03V, L:0.77V, D:1.26V, T:1.48V
CAL CHECK UNIT	: 400

CONFIGURATION CONTROL STATUS

PRINT SPEED	: COMMAND
PRINT DARKNESS	: COMMAND
PRINT WIDTH	: COMMAND
PRINT OFFSET	: COMMAND
TEAR OFF	: COMMAND

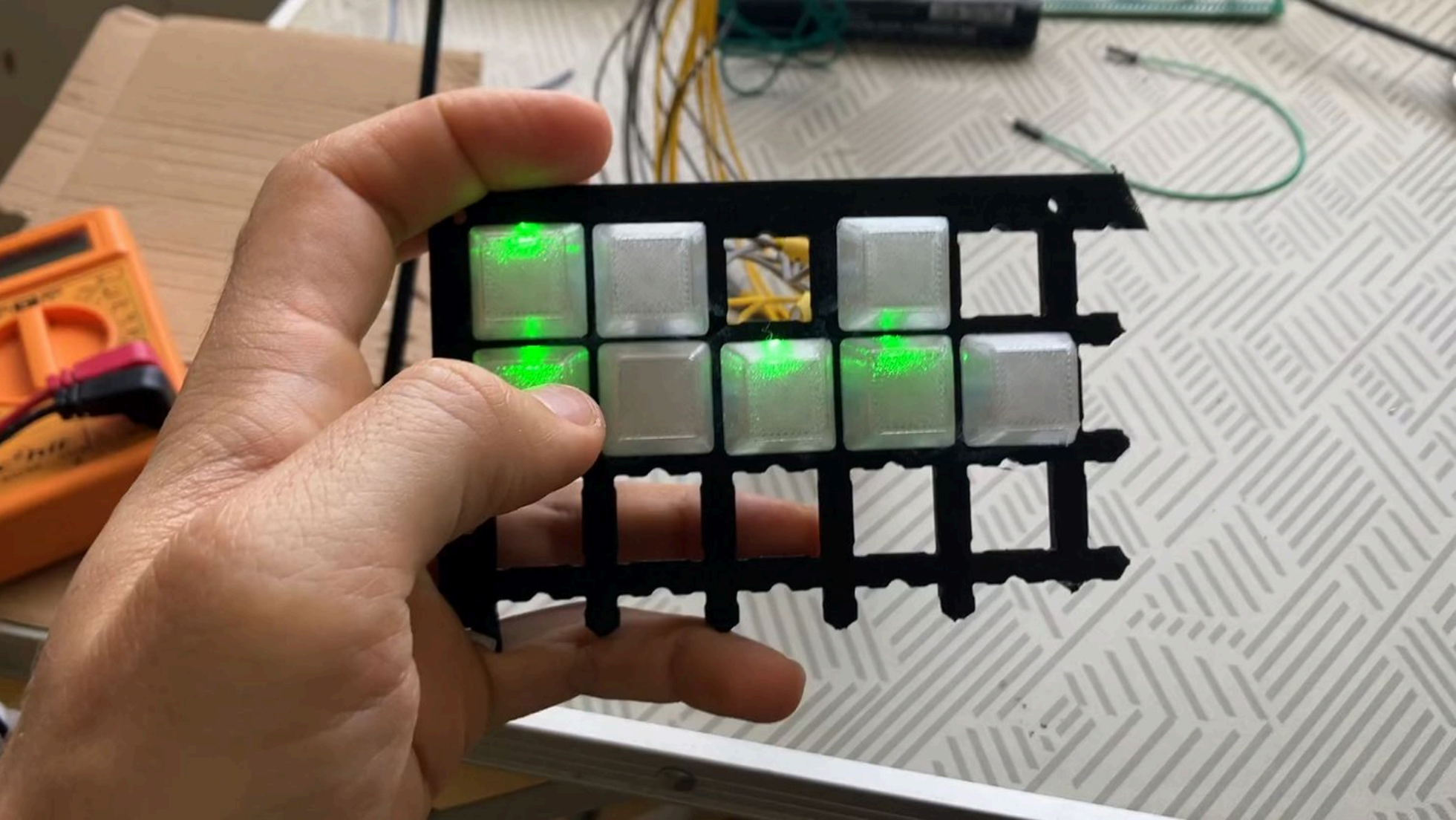
BIXOLON XD5 – 43d CONFIGURATION

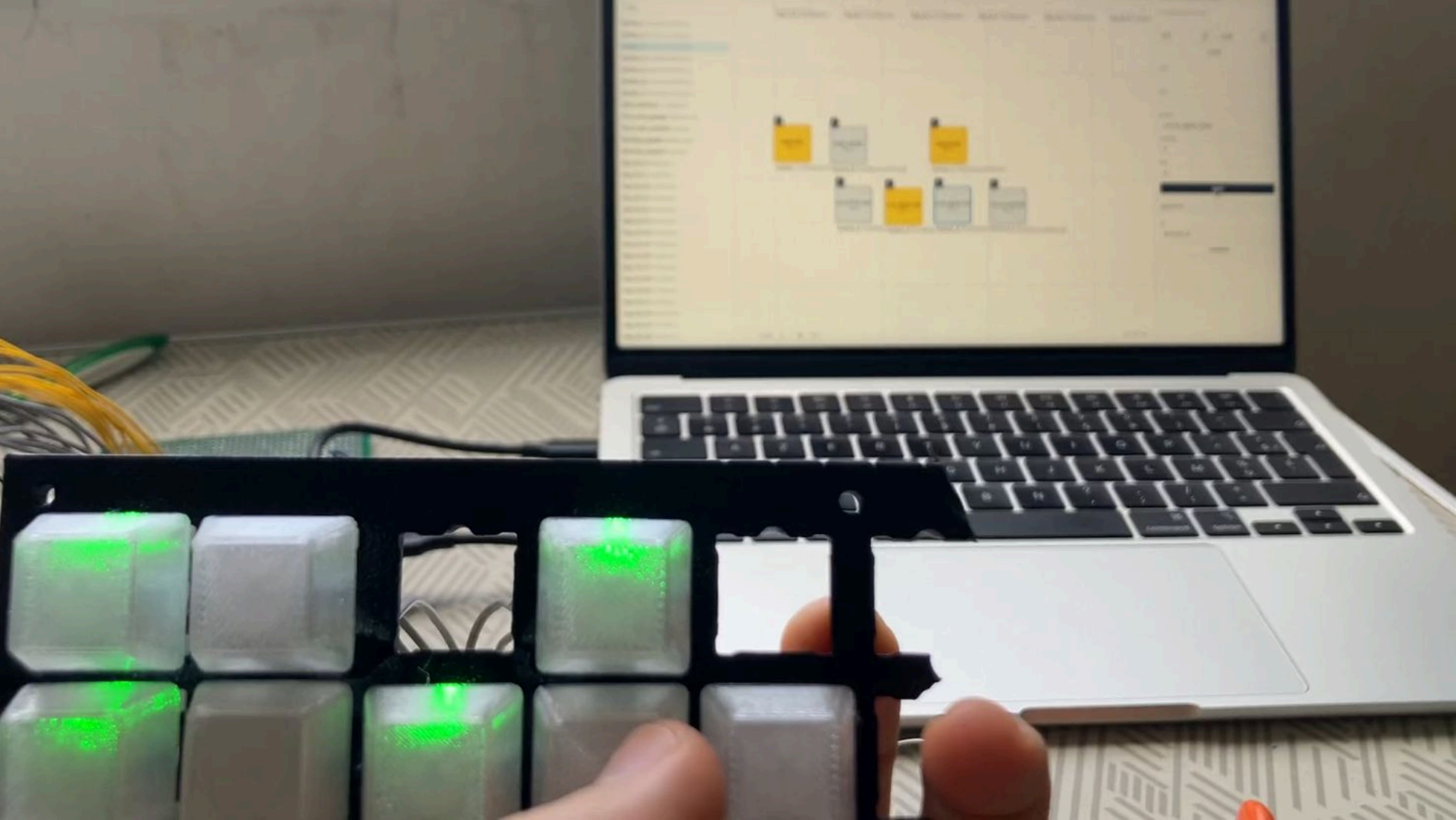
FIRMWARE VERSION	: V01.40 STD 101722
EMULATION	: SLCS & BPL – Z & BPL – E
PRINT DARKNESS	: 20/30
PRINT SPEED	: 4 IPS
TEAR OFF VALUE	: + 000 DOT
TEAR OPTION	: CUTTER
PRINT OFFSET	: + 0000
POWER – UP ACTION	: NO ACTION
MEDIA TYPE	: WEB PAPER

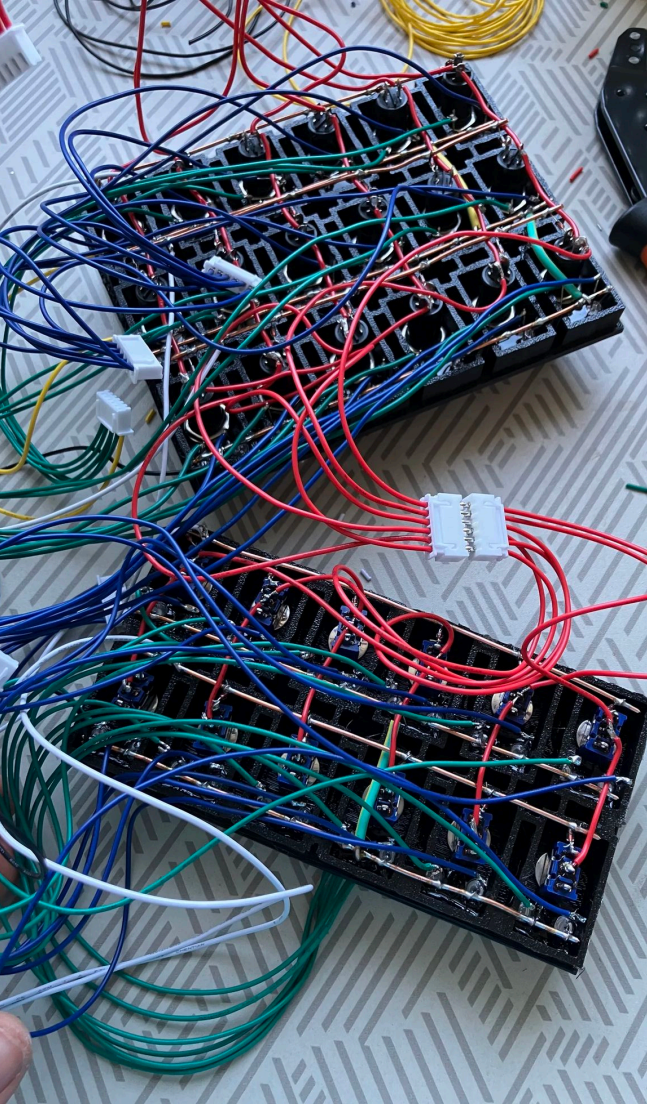
: 601 DOT
: 1232 DOT
: 115200
: NONE
: 8
: 1
: HARDWARE
: BEGWWRKA24020051
: CP – 850 (SBE. USA – 1 CHARSE
: WCP1252 – LATIN1
: U.S.A
: 1/1/11
: 0:0
: DISABLE

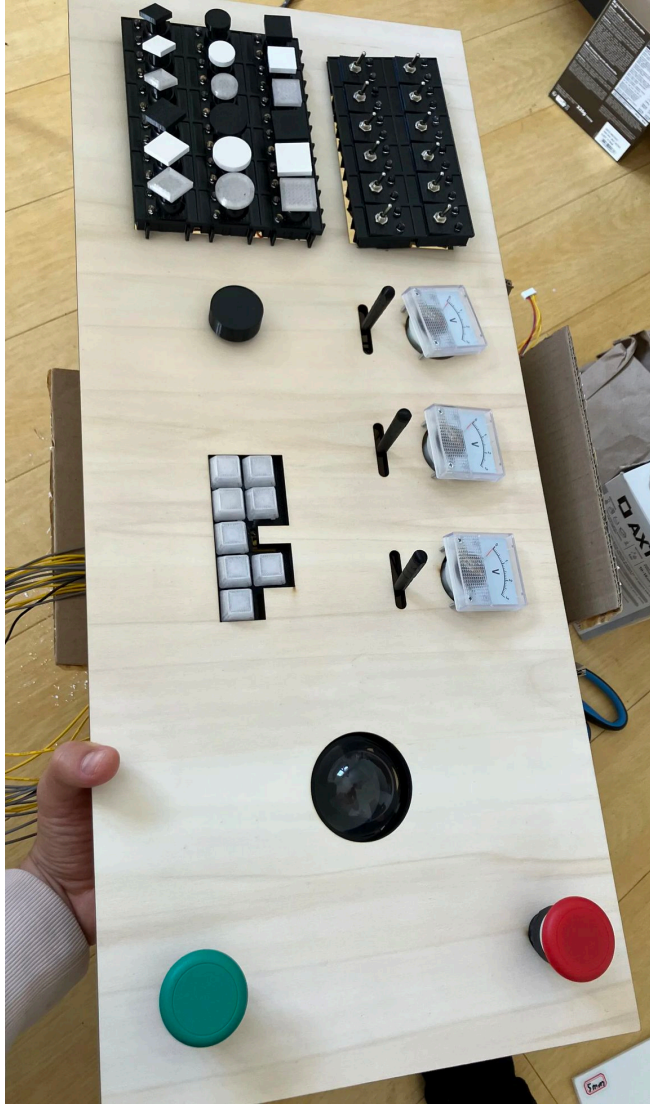
: COMMAND
: COMMAND

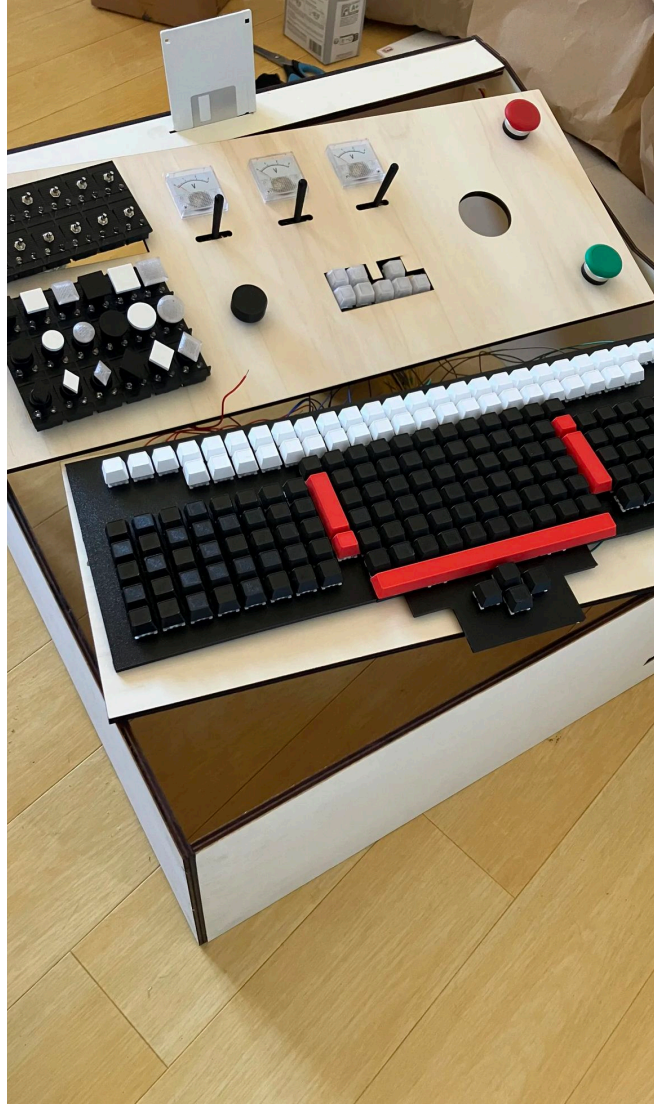








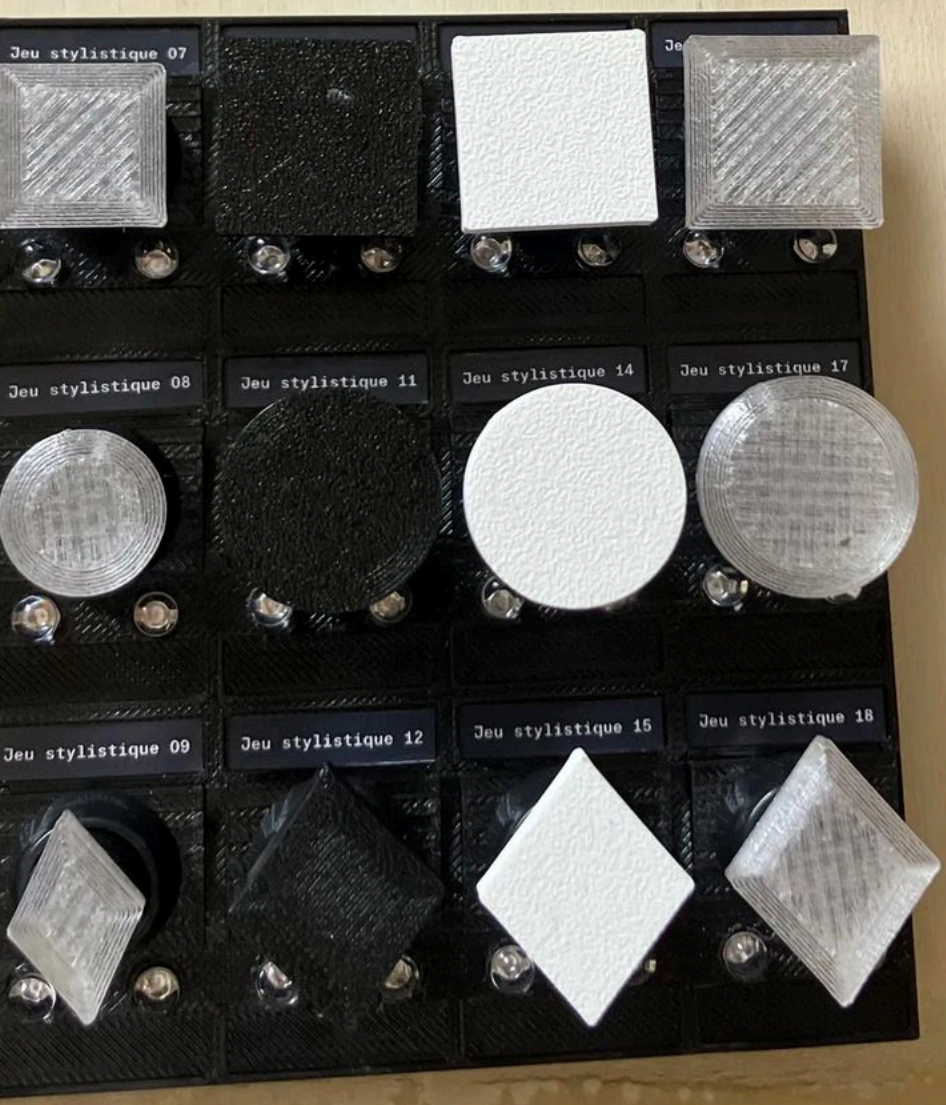






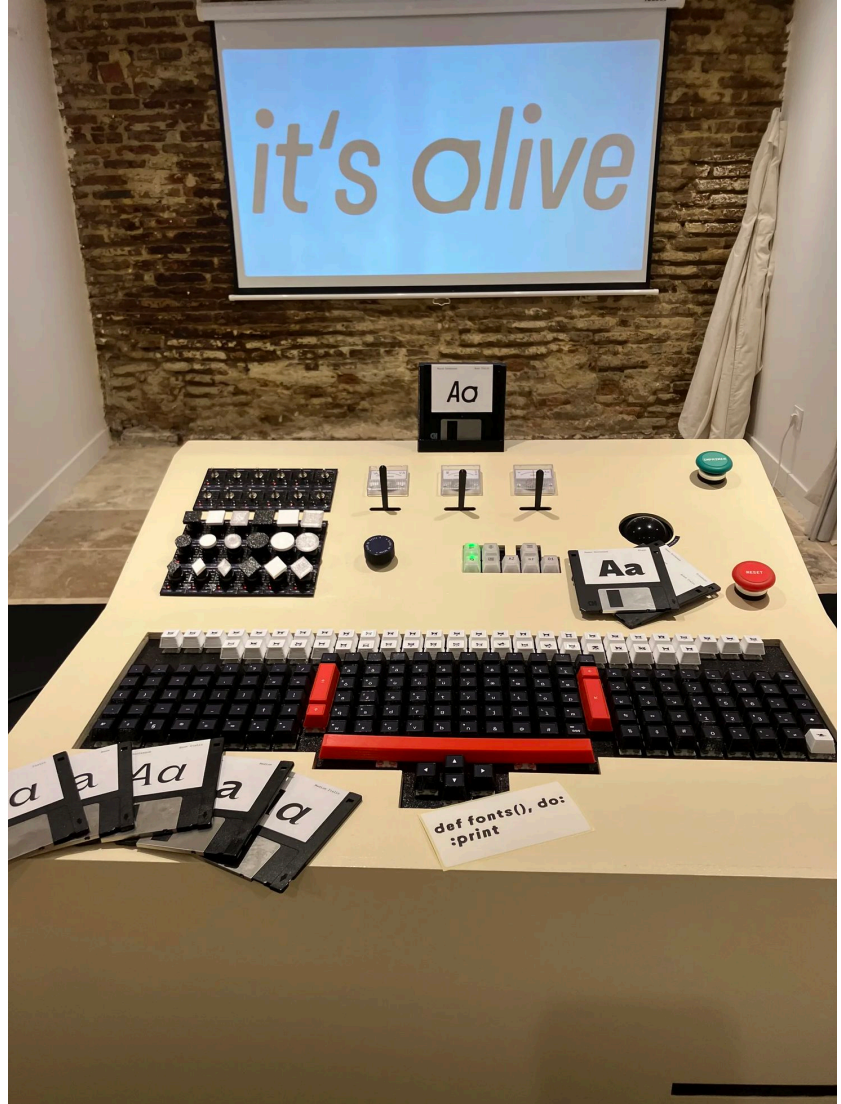



Bas-de-casse en petites capitales
Tout en petites capitales
Ponctuation adaptée à la casse
Zéro barré
Oui/Non On/Off
Oui/Non On/Off
Oui/Non On/Off
Oui/Non On/Off
a alternate
g alternate
Variantes de titrage
Lettres italiques ornées
Oui/Non On/Off
Oui/Non On/Off
Oui/Non On/Off
Oui/Non On/Off


Jeu stylistique 07
Jeu stylistique 08
Jeu stylistique 11
Jeu stylistique 14
Jeu stylistique 17
Jeu stylistique 09
Jeu stylistique 12
Jeu stylistique 15
Jeu stylistique 18



L3
L8
L2
L6
L7
L5
L0
L4
U2
L1


it's alive
Aa
Aa
def fonts(), do:
:print

Où placer la complexité ?

Du matériel sans connaissance du logiciel

- Une matrice de boutons ne sait pas à quoi elle sert
- Un contrôleur de LED ne sait pas à quoi il réagit
- Se contenter d'être une entrée (ex: un bouton) ou une sortie (ex: une LED)

Du logiciel sans connaissance de l'installation

Parti-pris :

- Décrire chaque sorte d'élément physique :
  - Qu'est-ce qu'un bouton ?
  - Qu'est-ce qu'un joystick ?
  - Qu'est-ce qu'une jauge ?
- Ne pas décrire d'actions :
  - ~~"Si je pousse le joystick de taille de texte vers la droite, la taille du texte augmente"~~
  - ~~"Si je fais tourner la trackball, un mot aléatoire remplace le texte"~~
- Du code pur et donc déterministe
- Repousser les effets au bord de l'installation
- L'installation complète est décrite par les données et non par du code

Du logiciel sans connaissance de l’installation

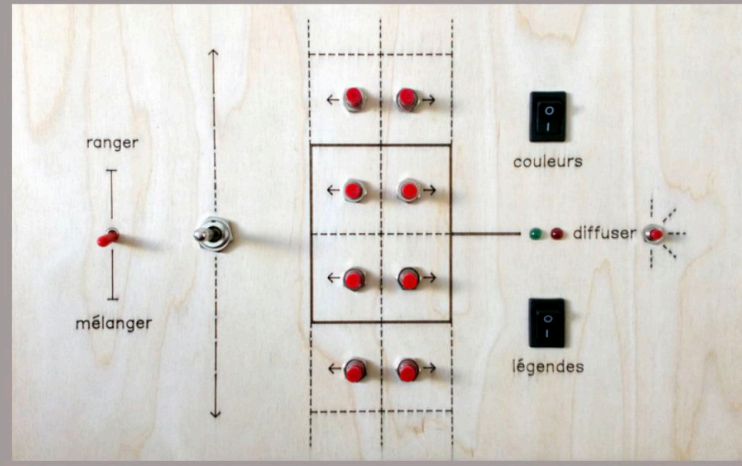
Conséquences :

- Changer un module est simple
- Changer un comportement est simple
- La manière dont les instructions arrivent à l’installation n’a pas d’importance
- Les effets de bord sont contenus

La simulation comme outil de conception

- Simuler le matériel aide à le comprendre et à le réaliser
- Le réaliser aide à affiner la simulation

∞ cycle vertueux



Panneau de contrôle

directions



image 1



image 2



image 3



image 4

scroll



doux

divers





couleurs



légendes



ordre

Scene front distance to the mirror 20000mm

20000

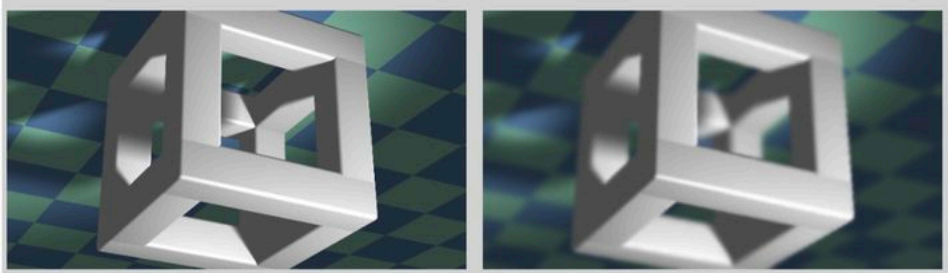
Tile size : 48.25mm / Scene width : 178.03mm / Scene depth : 354.54mm

Sensor distance to the mirror (ideal : around 408.20mm): 408.3mm

408.3

Step : Drawing done..

Redraw



Scene front distance to the mirror 5000mm

5000

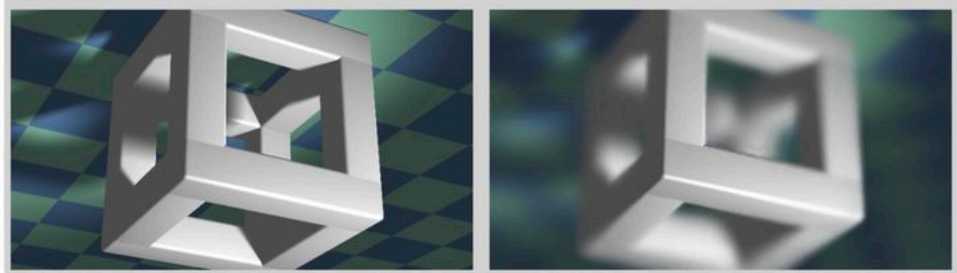
Tile size : 12.06mm / Scene width : 44.51mm / Scene depth : 88.63mm

Sensor distance to the mirror (ideal : around 434.96mm): 435.2mm

435.2

Step : Drawing done..

Redraw



Mirror radius (mm) 57

Draw tangents ? ☐

Draw normals ? ☑

Draw infinity ray fan ? ☐

Zoom 1.5

Mirror focal length (mm) 400

Rays (min 2) 4

Source distance 10000

Sensor position 416

Sensor width 3.58

Sensor height 2.02

Pixel size 0.0028

Source height 250

Some information :

Average focal plane location for on-axis object at 10000mm : 416.75mm

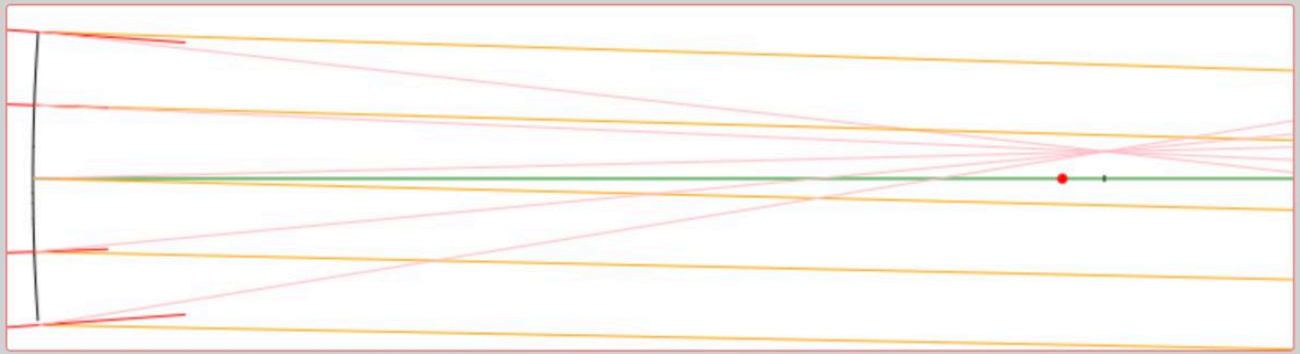
FoV : 0.49° x 0.28° (at 10000mm, that is 85.90mm x 48.47mm)

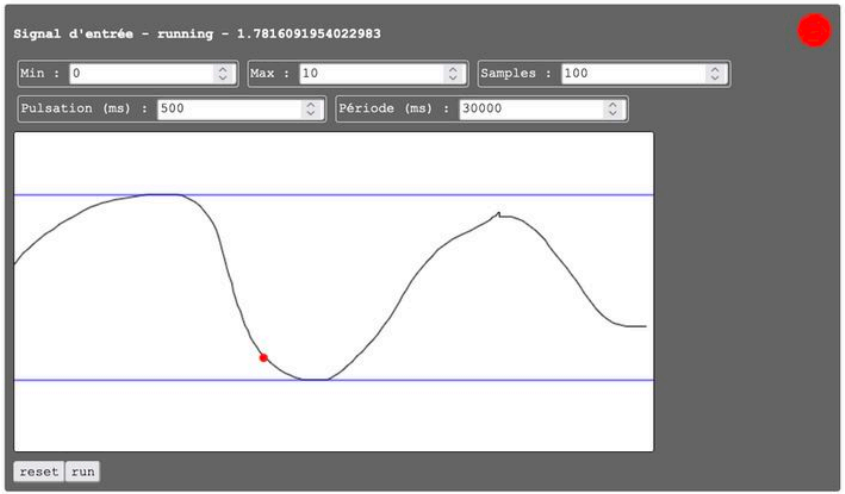
Longitudinal aberration : 0.17mm

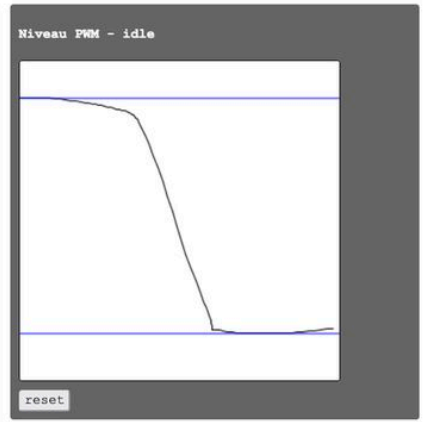
Geometric blur circle for sensor position : 0.2369mm (84.62 px wide)

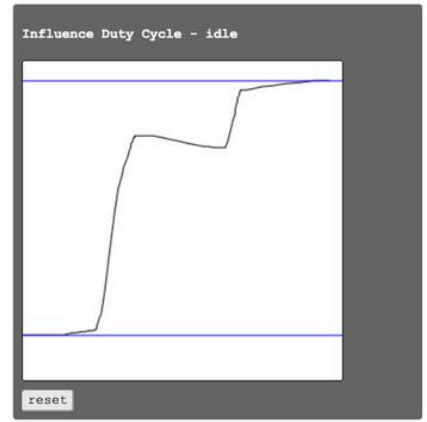
Minimal blur circle @ 550nm (airy disk diameter) : 0.0047mm (1.68px wide)

Dawes limit : 1.02 arcsec

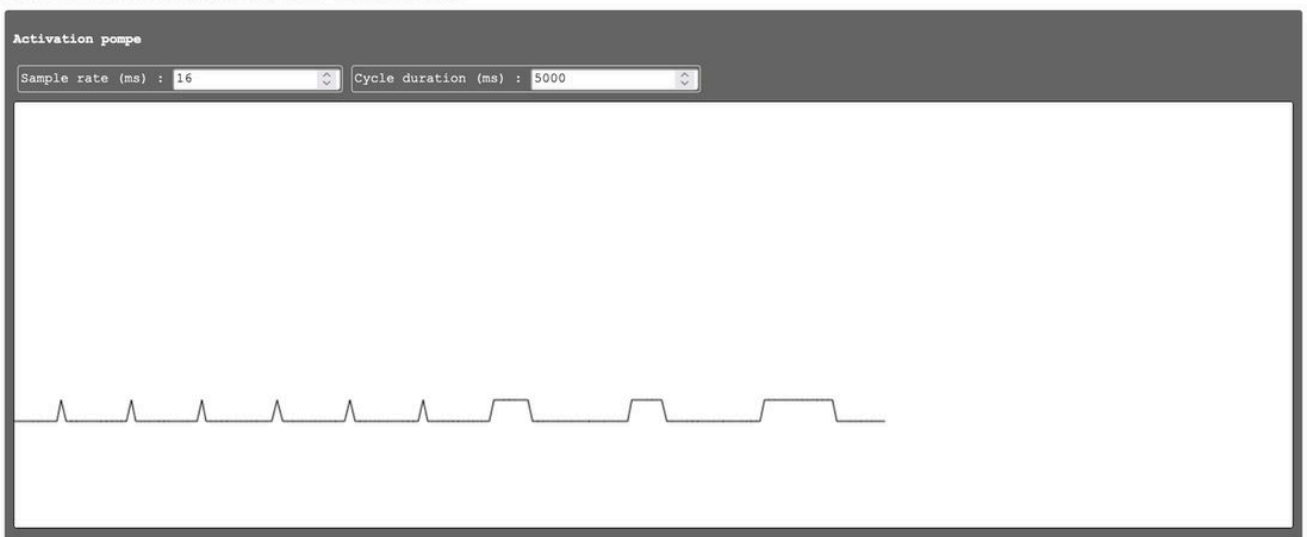








Niveau PWM 0.39095117845117855 Duty Cycle 708.2365319865321



Petit tour dans le simulateur

Points clés du système

`Composant` = `type` (état, réglages) + `instance` (adresse, traduit événements, transmet au coeur)
Virtuel + physique = une seule instance partagée
Instancier des composants = faire le pont entre logique et adressage physique

Fonction de transition / réduction

ancien état + événement = nouvel état + instructions

Quand j'appuie sur "texte à gauche", il se passe quoi ?

- J'appuie sur la touche ou bien sur sa contrepartie numérique.
- Un événement `bouton_2 appuyé` arrive dans le système.
- Il parvient à un composant appelé `bouton_2`.
- Le composant dit `je suis l'alignement à gauche` et transmet `alignement à gauche` au coeur
- `état 1 (texte centré)` + `alignement à gauche` = `état 2 (texte à gauche)` + `mise à jour des leds`
- L'affichage se met à jour avec le nouvel état
- Les instructions sont communiquées au contrôleur des LEDs

-> Les LED sont synchronisées avec un état et non réactives à une action -> Tout changement d'état est projeté vers d'éventuelles dépendances

Quand je fais rouler la trackball, il se passe quoi ?

- Je fais rouler la trackball ou bien la contrepartie numérique
- Un événement `roulement` arive dans le système
- Il parvient au composant `trackball` qui le transmet au coeur
- Avant d'entrer dans le coeur, cet état est augmenté d'un `mot aléatoire`, ex. "typography"
- `état 1 (ancien texte)` + `changer le texte (typography)` = `état 2 (typography)` + `mise à jour des leds`
- L'affichage se met à jour avec le nouvel état
- Les instructions sont communiquées au contrôleur des LEDs

-> La trackball ne sait rien du mot aléatoire, le texte et les LED ne savent rien de la trackball

Tout est indépendant

Démo Mail + Mini Brutator

contact@turing-express.fr

Code complet disponible le 25 octobre : github.com/lucassifoni

Site perso (articles, confs) -> lucassifoni.info
Développement indépendant -> turing-express.fr
Logiciel pour acteurs de l'archi/bâtiment -> alzo.fr
Métrologie optique -> chantepierre-optique.fr

