

Controllers



Tutorium CM und KK WS24



Pandora's Box ♩ = 60

4 knock

5 shake

4 shake

1

case #

open close on side off

bottom sensor

electronics

pico

+

what is happening here?

general

- in computer music context: what is a controller?
- what should it control?
- how should it control?
- what is a control data?
- what is a sensor?



protocols: recap



protocols: recap

Aspect	MIDI	OSC
Resolution	7-bit (0–127)	32-bit floating-point
Data Flexibility	Fixed message format, limited to MIDI parameters	Customizable, user-defined messages
Network Capability	Limited, mostly local (5-pin or USB)	Network-native (UDP/IP), internet-compatible
Ideal Use Cases	Simple panning, basic control setups	Complex, high-precision spatialization, interactive installations

what is a **serial** protocol? (USB)

protocols: recap

Aspect	MIDI	OSC
Resolution	7-bit (0–127)	32-bit floating-point
Data Flexibility	Fixed message format, limited to MIDI parameters	Customizable, user-defined messages
Network Capability	Limited, mostly local (5-pin or USB)	Network-native (UDP/IP), internet-compatible
Ideal Use Cases	Simple panning, basic control setups	Complex, high-precision spatialization, interactive installations

what is a **serial** protocol? (USB, UART, I²C) -> data is transmitted sequentially (bit by bit) over a single communication line or channel

types of devices

1. Traditional MIDI Controllers
2. Alternative Instrument Controllers
3. Touch and Multitouch Controllers
4. Wearable and Gestural Controllers
5. Custom DIY Controllers
6. Brain-Computer Interfaces
7. Haptic Controllers
8. Environmental and Sensor-Based Controllers
9. AI-Driven and Generative Controllers
10. Hybrid Controllers

types of devices

1. Traditional MIDI Controllers (keyboards, pads, faders, knobs)
2. Alternative Instrument Controllers (Wind Controllers, string controllers, electronic drum kits)
3. Touch and Multitouch Controllers (Tablets/Phones, Specialized Surfaces)
4. Wearable and Gestural Controllers (gloves, motion sensors, body sensors)
5. Custom DIY Controllers (Arduino or ESP32-based controllers, Modular controllers Monome, Eigenharp or digitize)
6. Brain-Computer Interfaces (Emotiv headsets, EEG-based systems)
7. Haptic Controllers (vibration motors or actuators for tactile feedback)
8. Environmental and Sensor-Based Controllers (microphones, light sensors and cameras, pressure or temperature)
9. AI-Driven and Generative Controllers (AI systems that learn from and adapt to performer behavior, Generative controllers based on randomness or procedural systems.)
10. Hybrid Controllers

types of devices



types of devices



M.Marcoll: control issues (<https://vimeo.com/250123813>)

types of devices



<https://www.youtube.com/watch?v=ieUnl7pfQH4>

digitize violin

types of devices



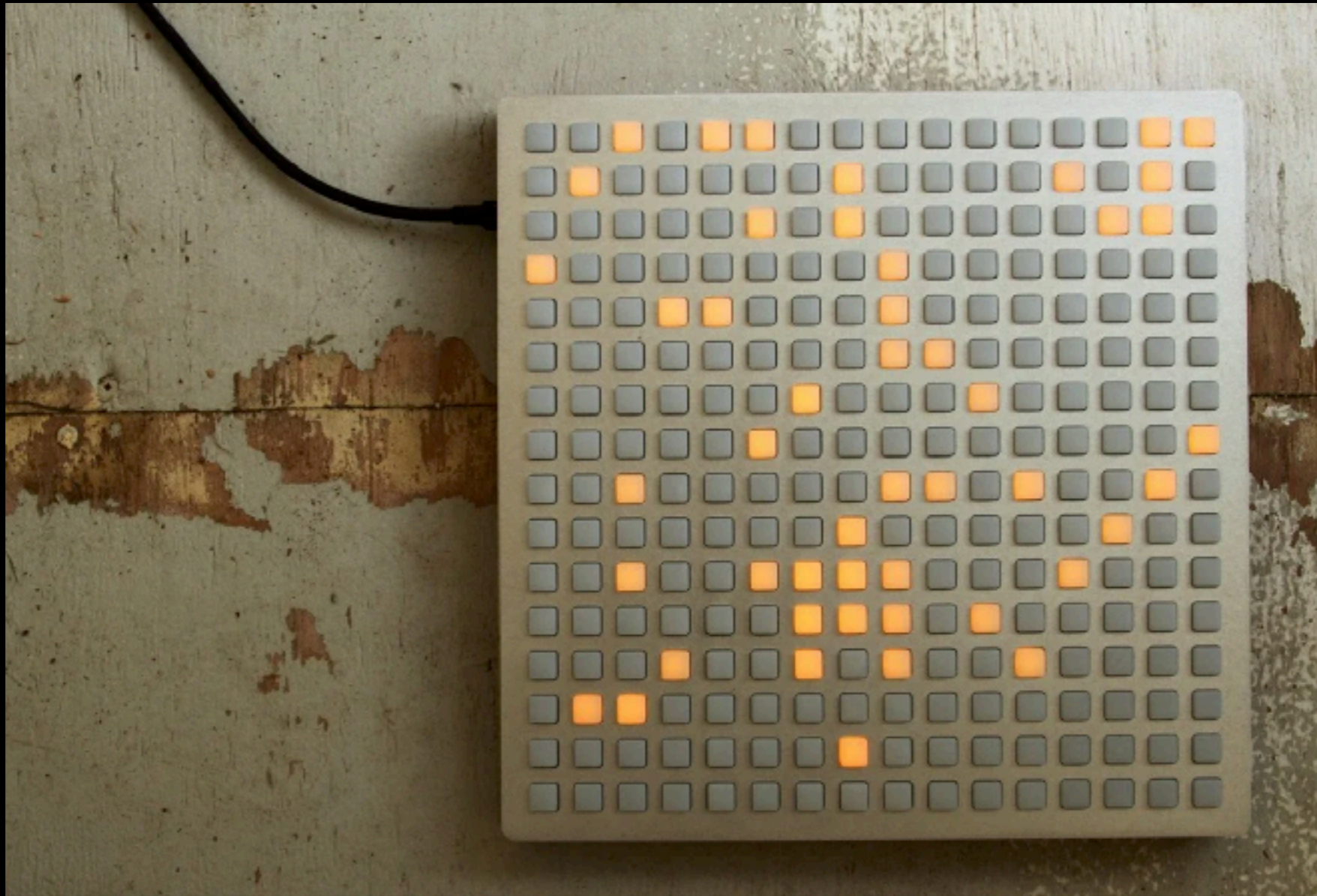
- sensel morph or similar j-play (joue): <https://www.youtube.com/watch?v=NvflQnHfbI4>

types of devices



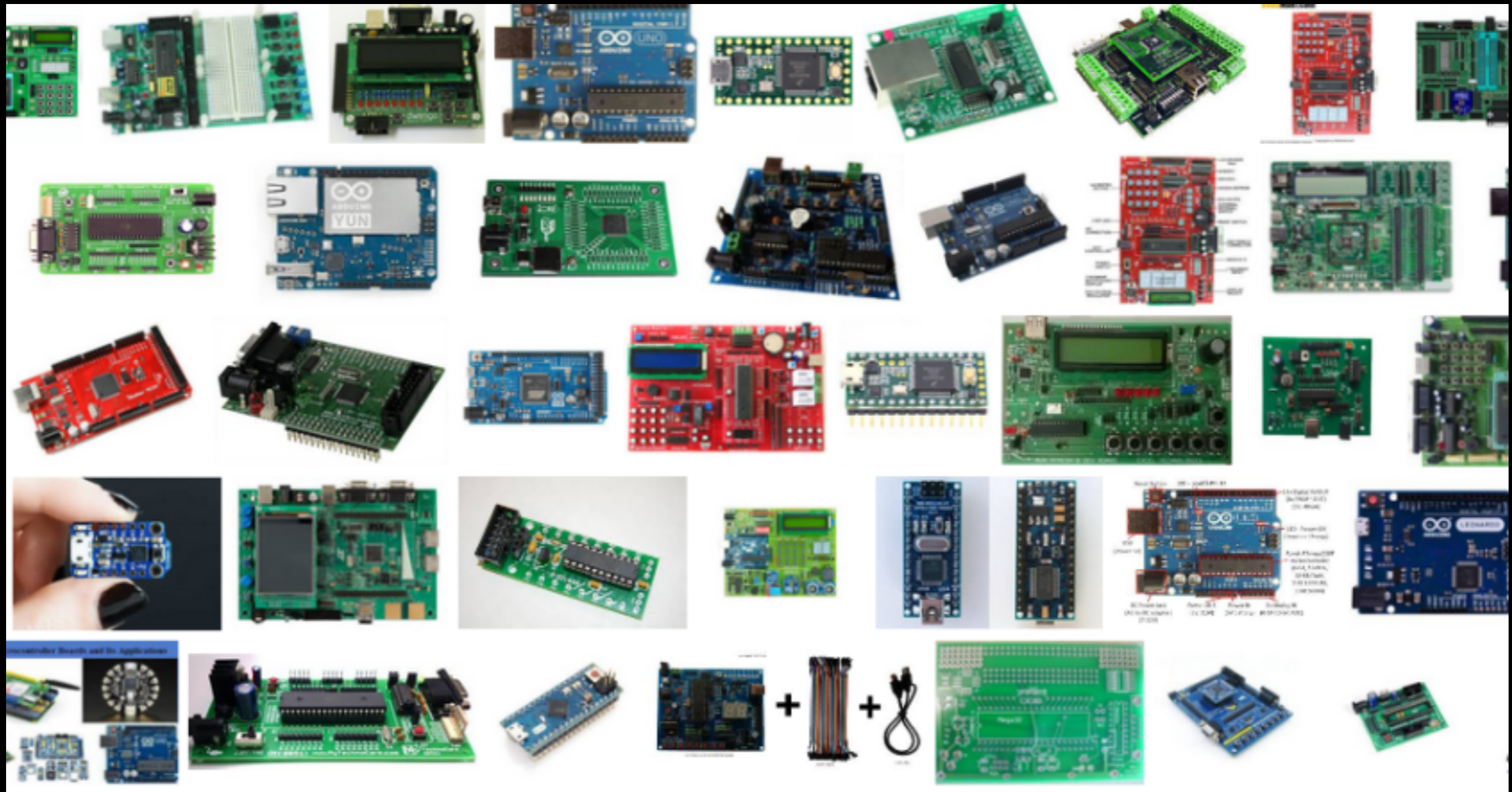
- soma lab

types of devices



- monome - chemical etudes by M.Ciciliani <https://vimeo.com/275471334>

Microcontrollers



- Arduino, ESP32, Raspberry Pi, Teensy...

Microcontrollers

1. General-Purpose Microcontrollers (arduino, raspberry pi pico, teensy) -> MIDI Controllers, Sensor Input, Audio Reactive Systems,
2. High-Performance Microcontrollers (Teensy 4.x Series, ESP32, STM32) -> Audio Synthesis, Complex Sensor Processing, Wireless MIDI/OSC
3. Internet of Things (IoT)-Enabled Microcontrollers (ESP8266, ESP32, Particle Photon) -> Collaborative Performance, Distributed Installations, IoT Music Interfaces
4. Specialized Audio Microcontrollers (Bela Boards, Axoloti Core, Teensy Audio Boards) -> Standalone Synthesizers, Audio Effects Processors, Low-Latency Interaction
5. DIY and Modular Systems (Micro:bit, Makey Makey, Adafruit Boards) -> Educational Projects, Interactive Installations, Rapid Prototypin

Microcontrollers: aspects to consider

1. Processing Power
2. Connectivity
3. I/O Requirements
4. Ease of Use

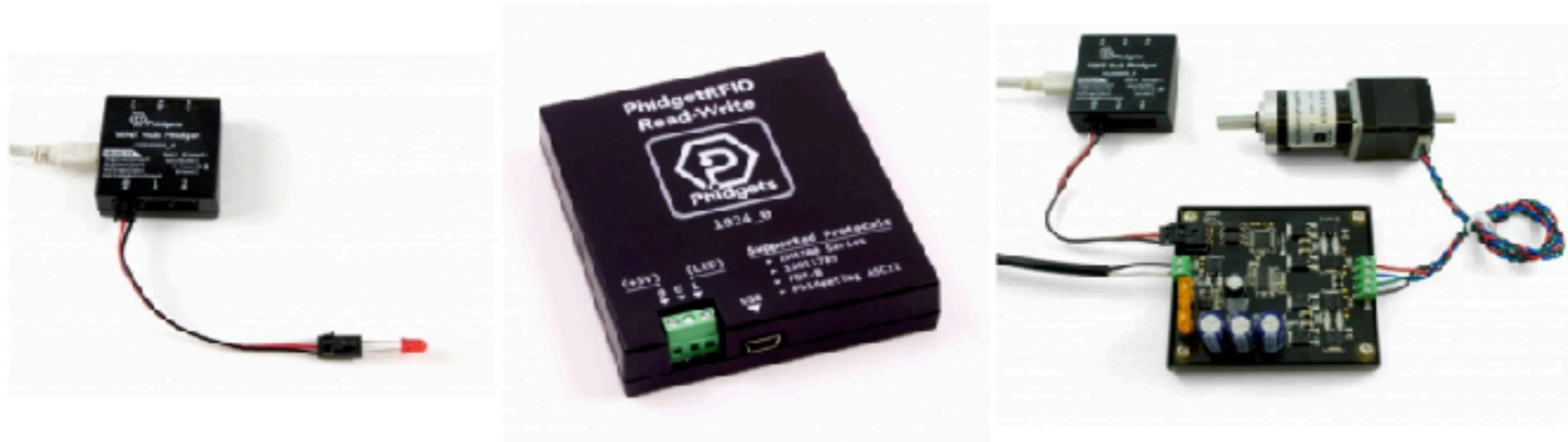
Microcontrollers

Feature	Microcontroller	Raspberry Pi
Processing Power	Low (8-bit/16-bit/32-bit, single-core)	High (multi-core ARM processor)
Memory	Small (few KB to MB)	Large (512MB to 8GB RAM)
Power Consumption	Very low	Higher (requires external power source)
OS	None or real-time operating system	Full operating system (Linux-based)
Programming	Low-level (C, C++)	High-level (Python, Java, C++)
I/O	Digital and analog I/O pins, PWM	Digital I/O, GPIO, requires extra components for analog I/O
Cost	Low (few dollars)	Moderate (typically \$5–\$70)
Use Cases	Embedded, simple controls, IoT devices	Full computing, networking, multimedia

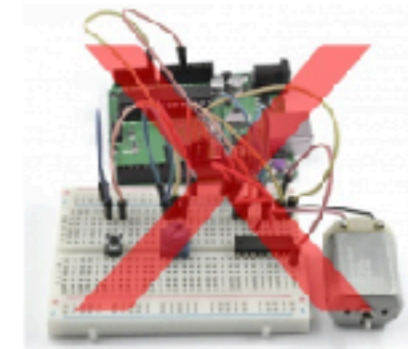
What Is A Phidget?

At Phidgets, our philosophy is that you don't have to be an electrical engineer in order to create sophisticated projects using motors, sensors and other electronics. You simply need to know how to code.

Phidgets are building-blocks for sensing and control using a computer, tablet, or phone. Phidgets enable your software application to interact with the physical world.

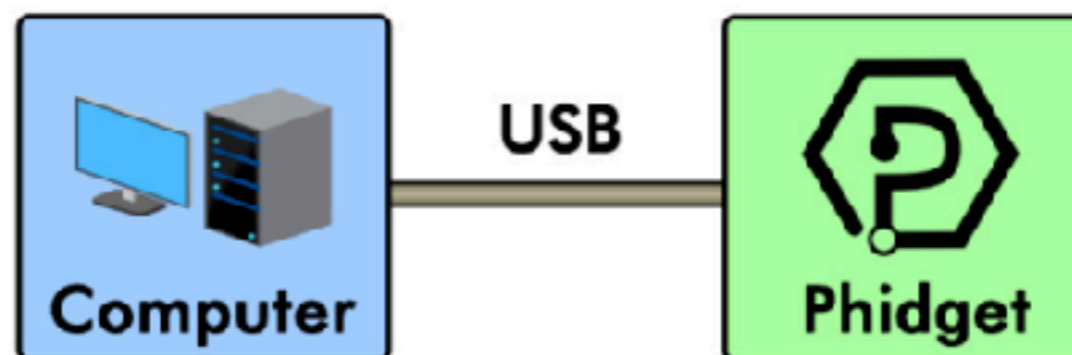


The advantage of using Phidgets is that all Phidgets can be plugged in and used in your project out of the box, without having to follow complex pinout diagrams or configure communication between the device and computer. With Phidgets, we deal with the details of getting the electronics behaving as they should, so you can focus on the programming and details of your final product.



Hardware Introduction

Phidgets connect via a USB port:



How to choose/design a suitable/
perfect controller for your piece.

BUYS A MIDI CONTROLLER

ONLY USES DRUM PADS



quickmeme.com

How to choose/design a suitable/ perfect controller for your piece.

1. Define Your Creative Goals (What role will the controller play in your music?, Expressive Needs)

2. Identify the Inputs and Outputs (knobs, sliders, buttons, touch, joystick, force, gesture, unconventional material; midi/osc, visual, haptic)

3. Choose a Platform

4. Design the Interaction

5. Build or Source Components (DIY vs. Pre-built)

6. Program the Controller

7. Prototype and Test

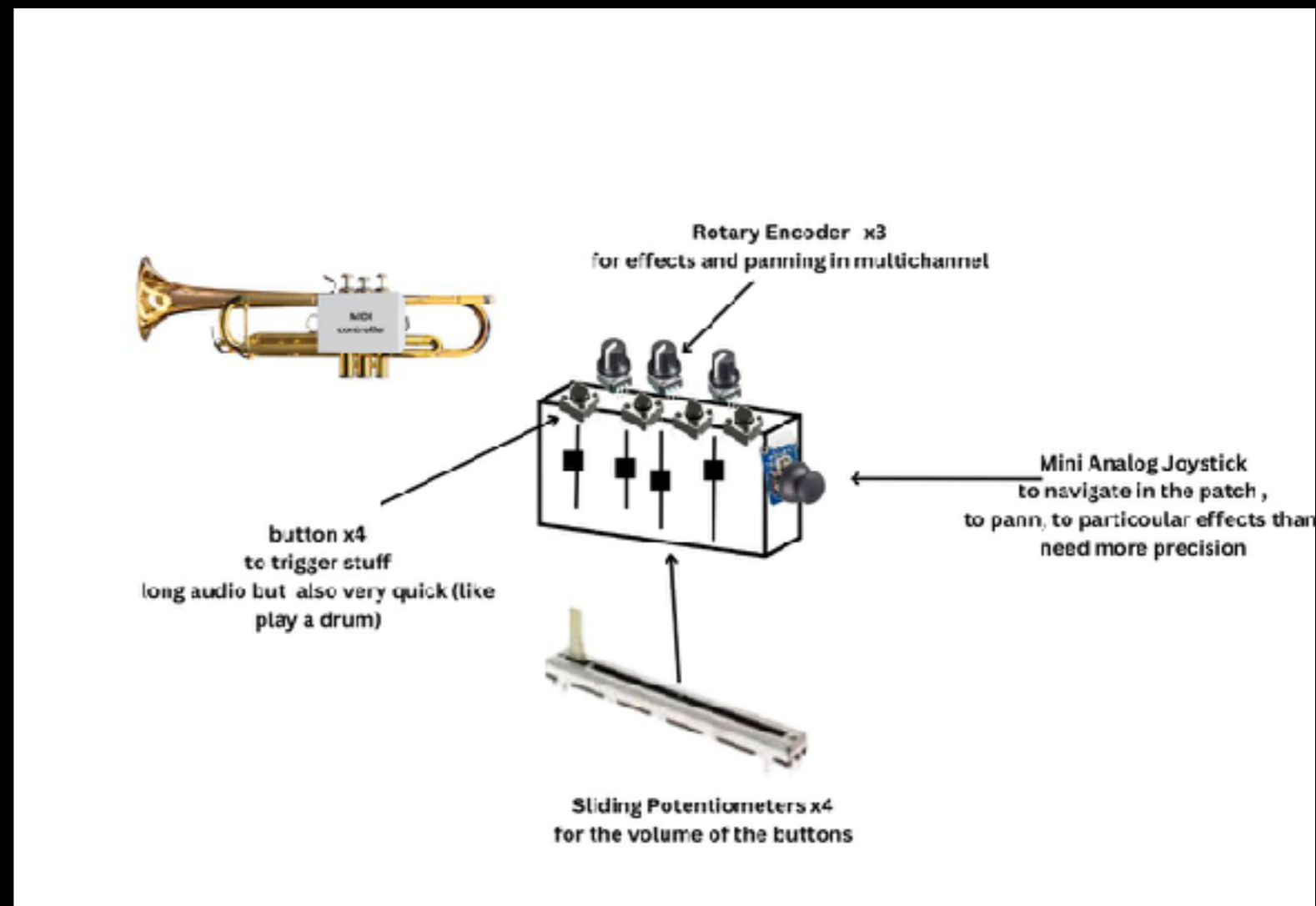
8. Consider Aesthetics and Ergonomics

9. Example Applications

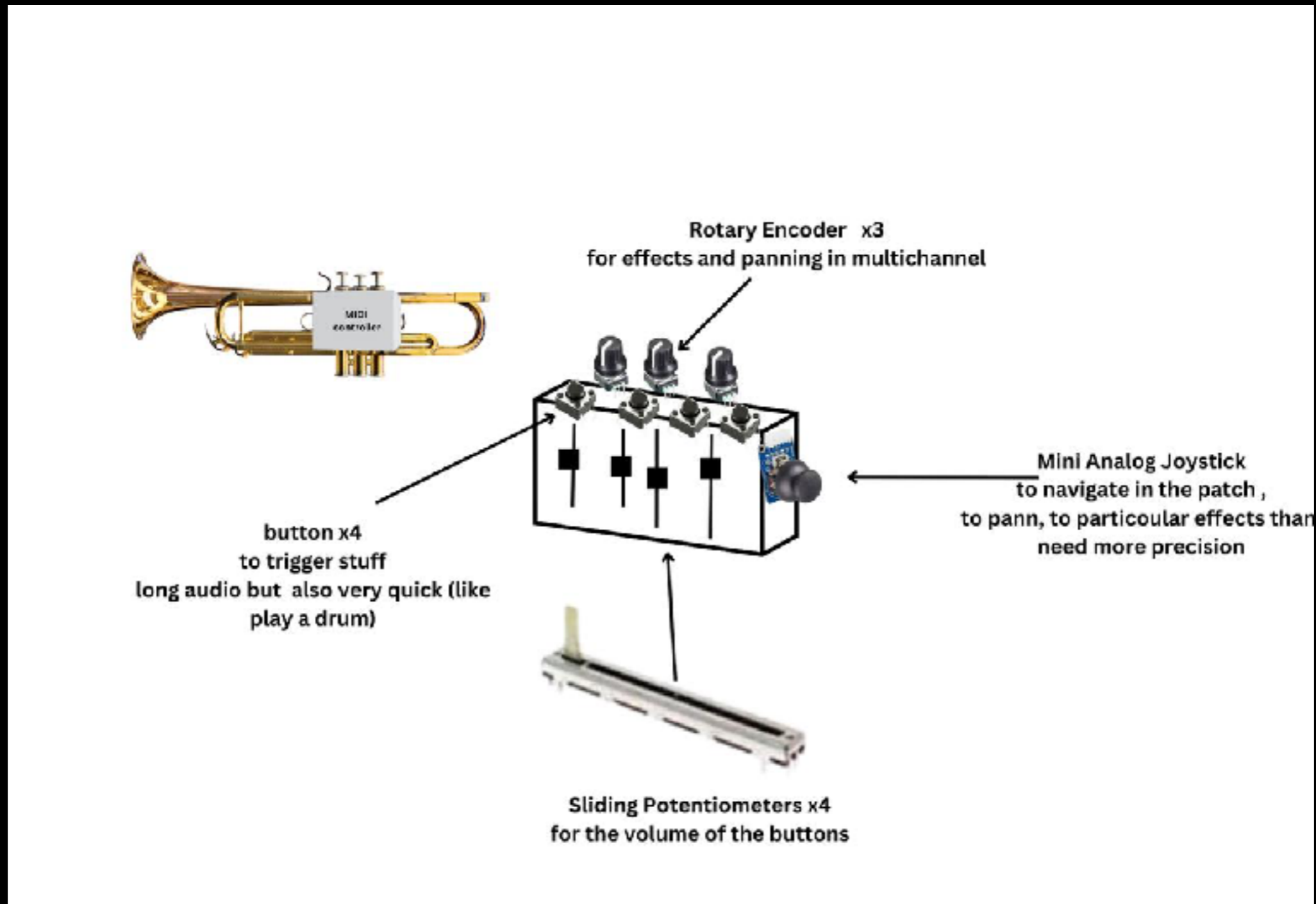
10. Keep the Music in Focus

How to choose/design a suitable/ perfect controller for your piece.

1. Define Your Creative Goals
2. Identify the Inputs and Outputs
3. Choose a Platform
4. Design the Interaction
5. Build or Source Components
6. Program the Controller
7. Prototype and Test
8. Consider Aesthetics and Ergonomics
9. Example Applications
10. Keep the Music in Focus



How to ~~choose~~/design a suitable/ perfect controller for your piece.

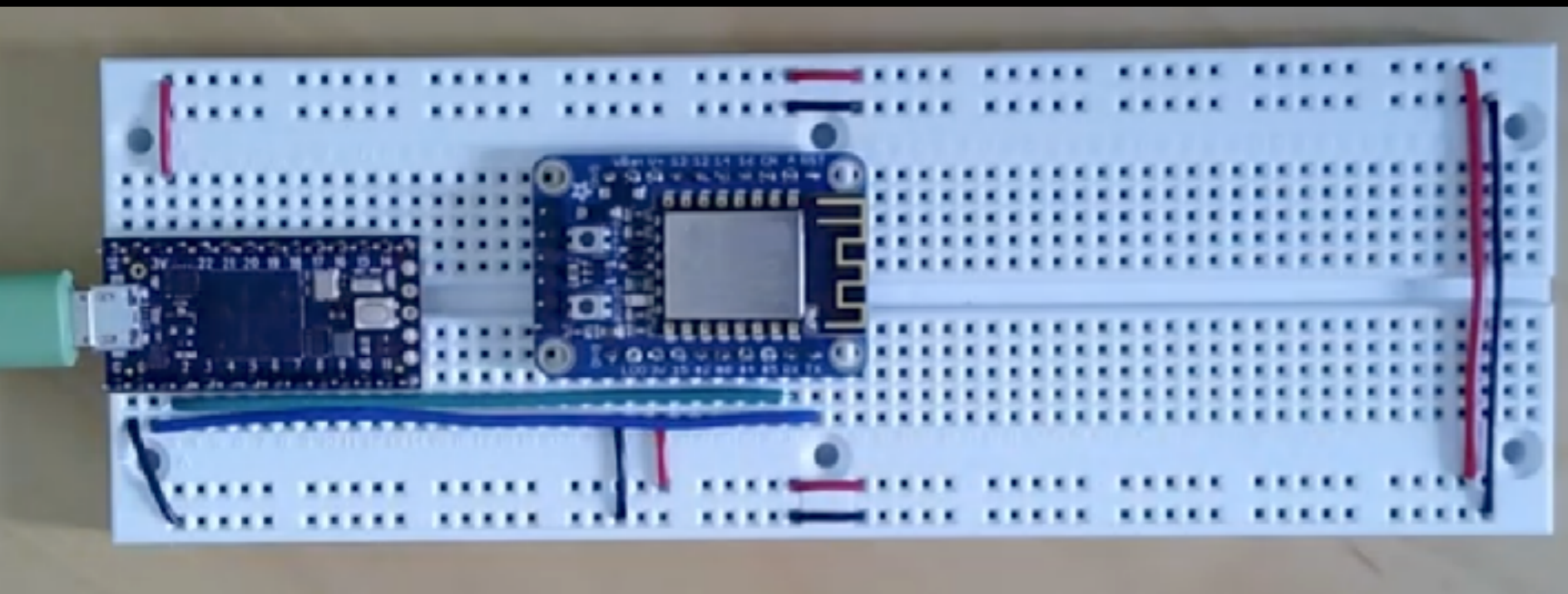


DiY controller: how to

<https://www.espressif.com/en/products/socs/esp32-s3>

<https://www.berrybase.at/espressif-esp32-s3-devkitc-1-n8r8-development-kit-8mb-psram-8mb-flash>

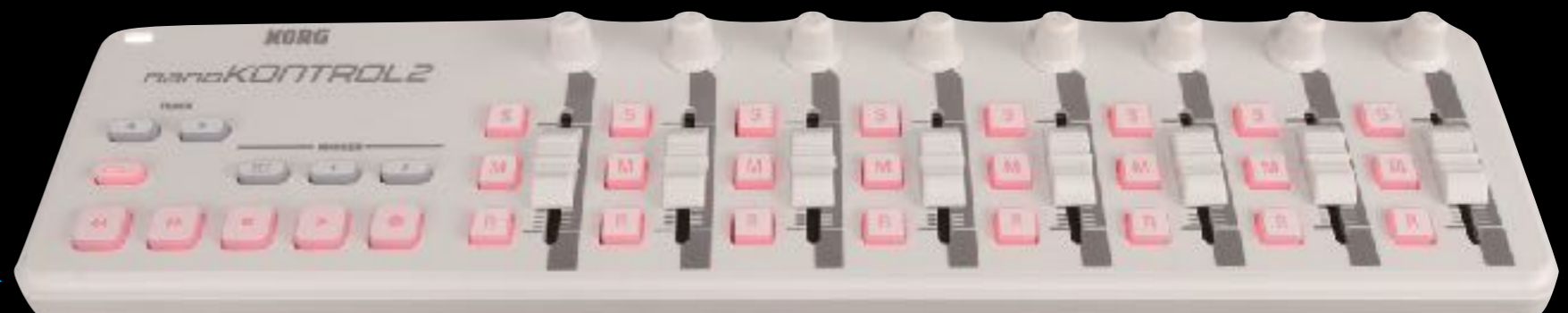
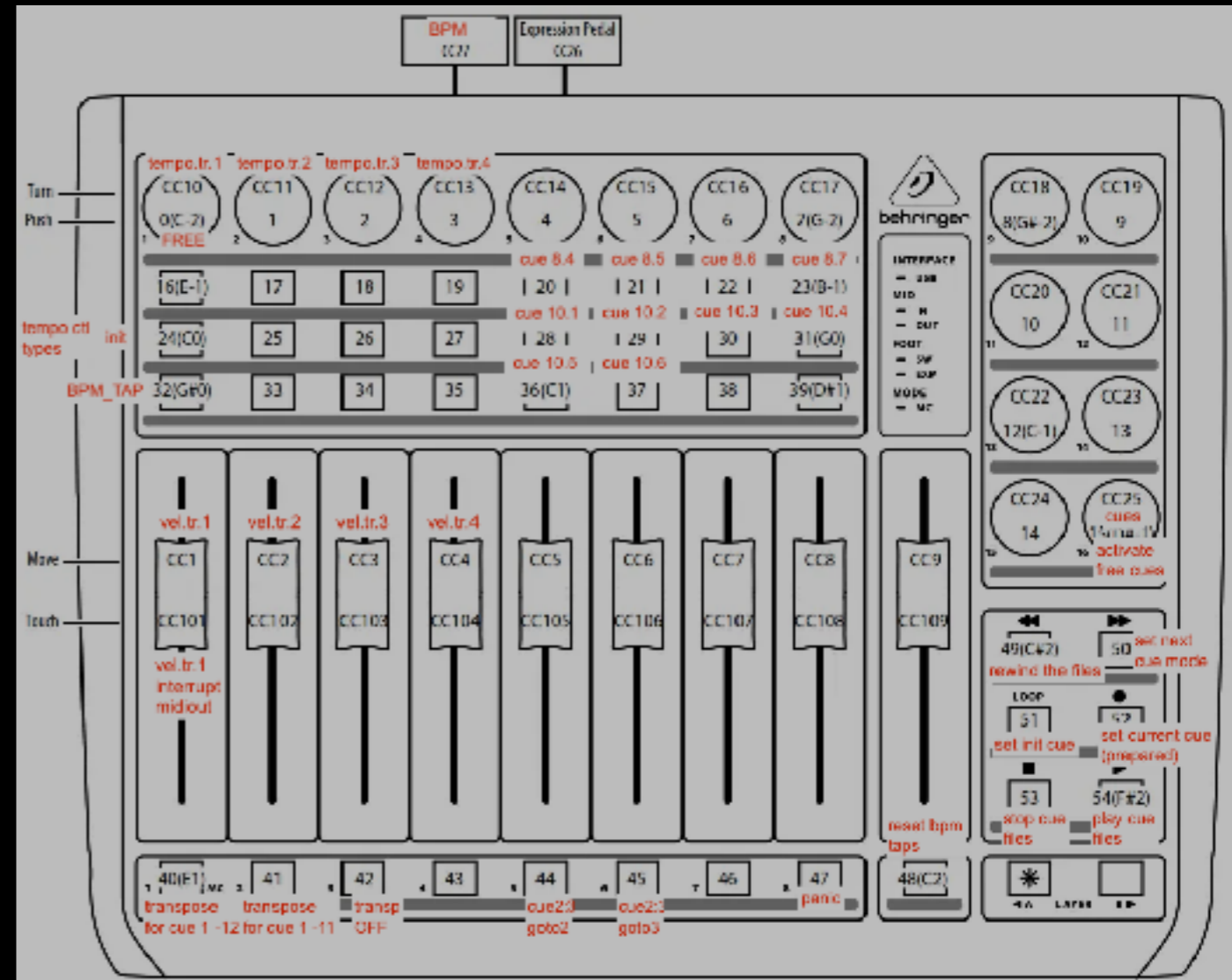
<https://www.pjrc.com/store/teensy41.html>



Powerbank 2300 slim



Mapping and remapping from controller to controller.



Types and usage of the control data.

1. Continuous Data
2. Discrete Data
3. Gesture/Movement Data
4. Random/Generative Data
5. Environmental Data
6. Biofeedback Data
7. Symbolic Data (midi/osc or text)
8. Sensor Data (Piezoelectric sensors on surfaces, Distance sensors, Touch-sensitive surfaces)

practice: record your control data along with the audio

- SuperCollider patch.

practice: multichannel spatialization

- which controller/sensor?
- data/protocol?
- where is mapping?

