

USB-Midi-Controller Workshop

Schedule

- Presentation
 - MIDI
 - Applications
 - Signalflow
 - Electronics Basics
 - Programming
 - Mapping in Software
 - Examples: Layouts, Cases
- Tinkering

MIDI

- Protocol for Musical Information
 - Note on / Note off
 - Controlchange
- Hardware specification
 - MIDI
 - USB-MIDI
 - USB-Serial with Converter

Applications

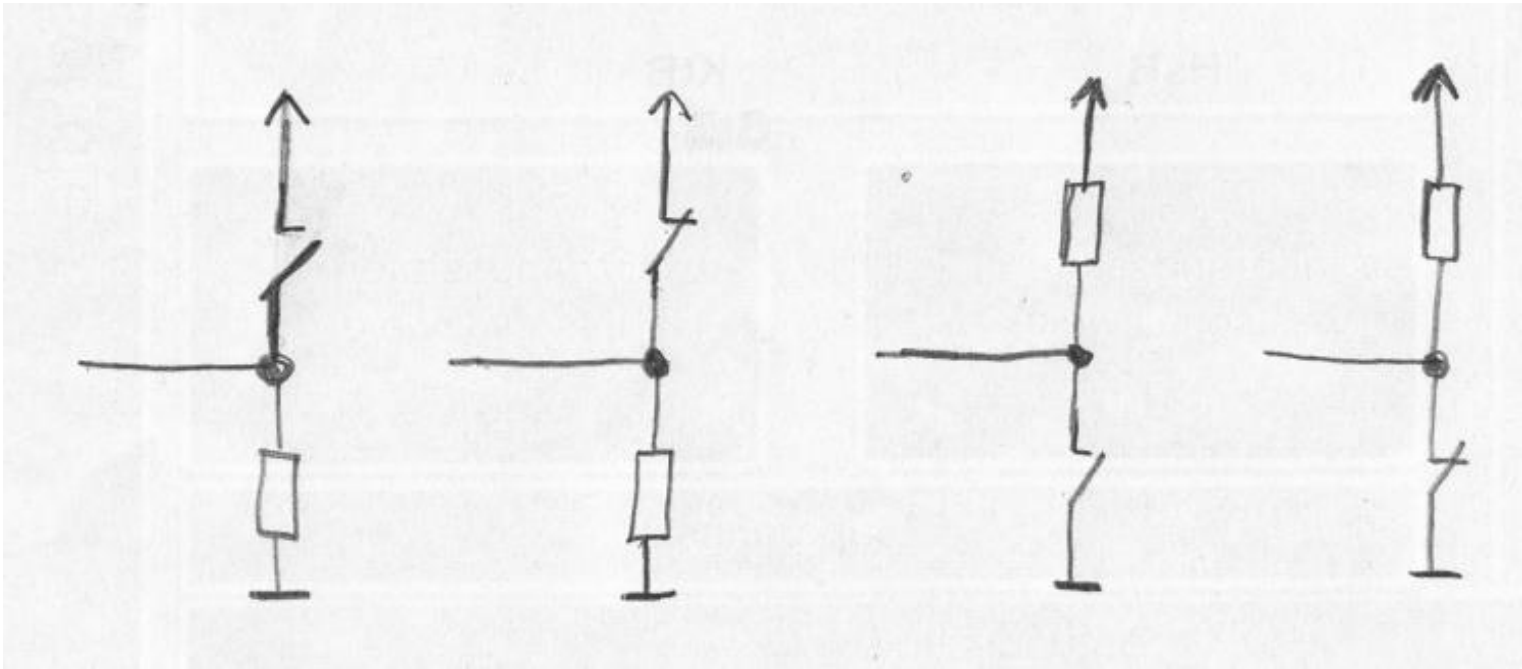
- DJ
- VJ
- Musicplayer
- ???

Signalflow

1. Detect Input with Arduino
2. Preprocessing in Arduino (Debounce etc.)
 - SerialMIDIElements-Library
3. Send MIDI-Data via USB-Serial
 - SerialMIDIElements-Library
4. Receive MIDI-Data at PC and convert to MIDI-Message for the Music-Software
 - Linux/Mac: Hairless MIDI
 - Linux: ttymidi
 - Windows: Hairless MIDI + LoopBe Virtual Midi

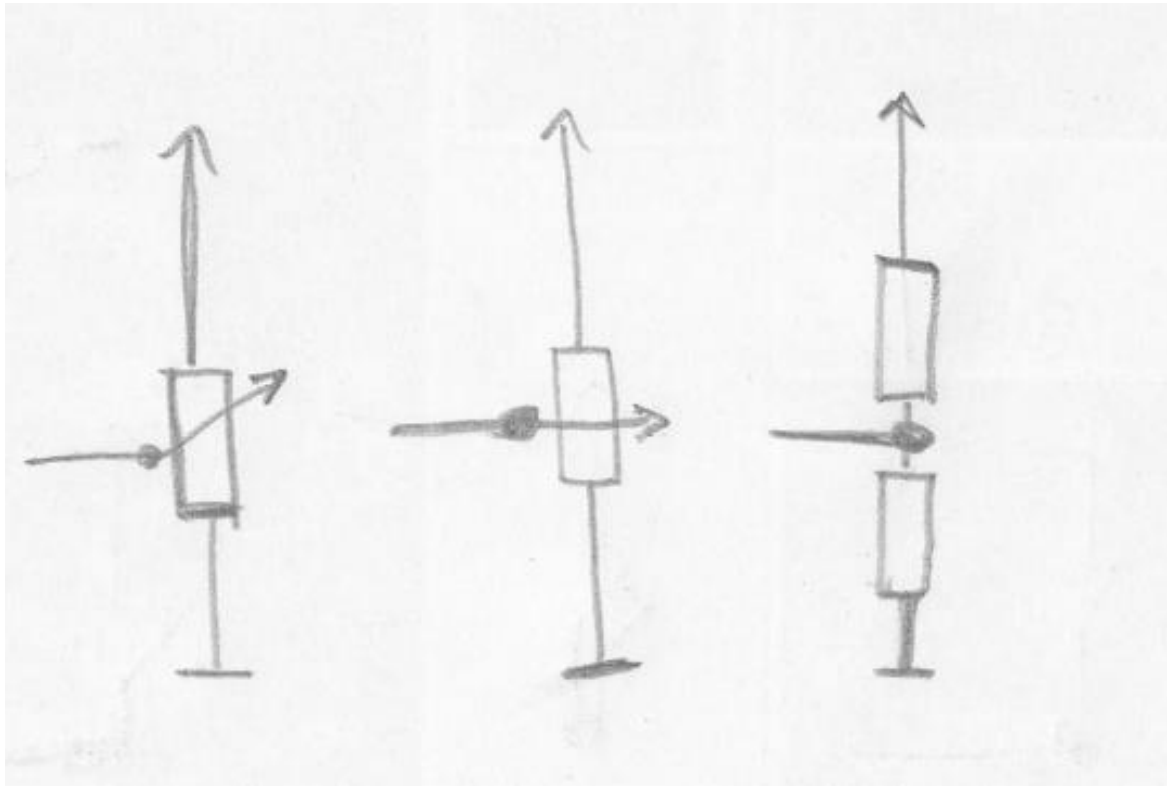
Electronics

- Buttons



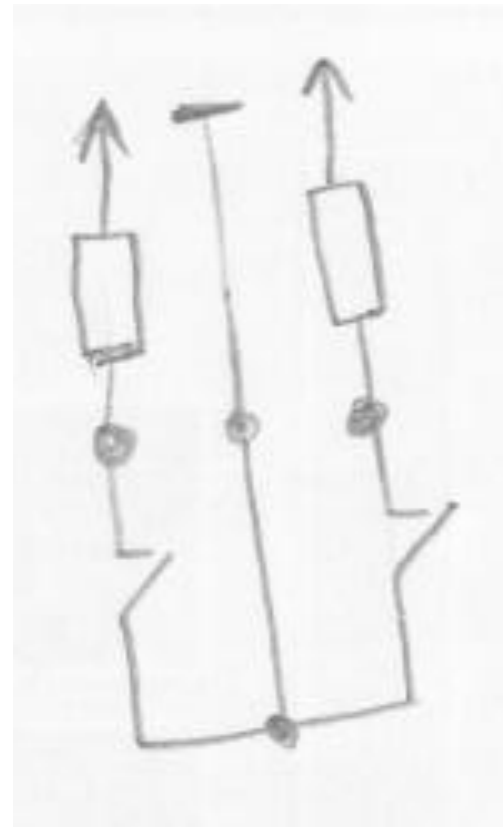
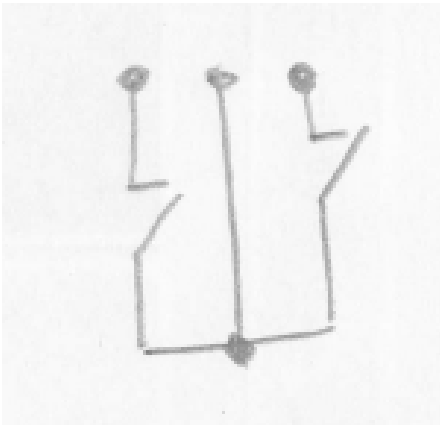
Electronics

- Potentiometers



Electronics

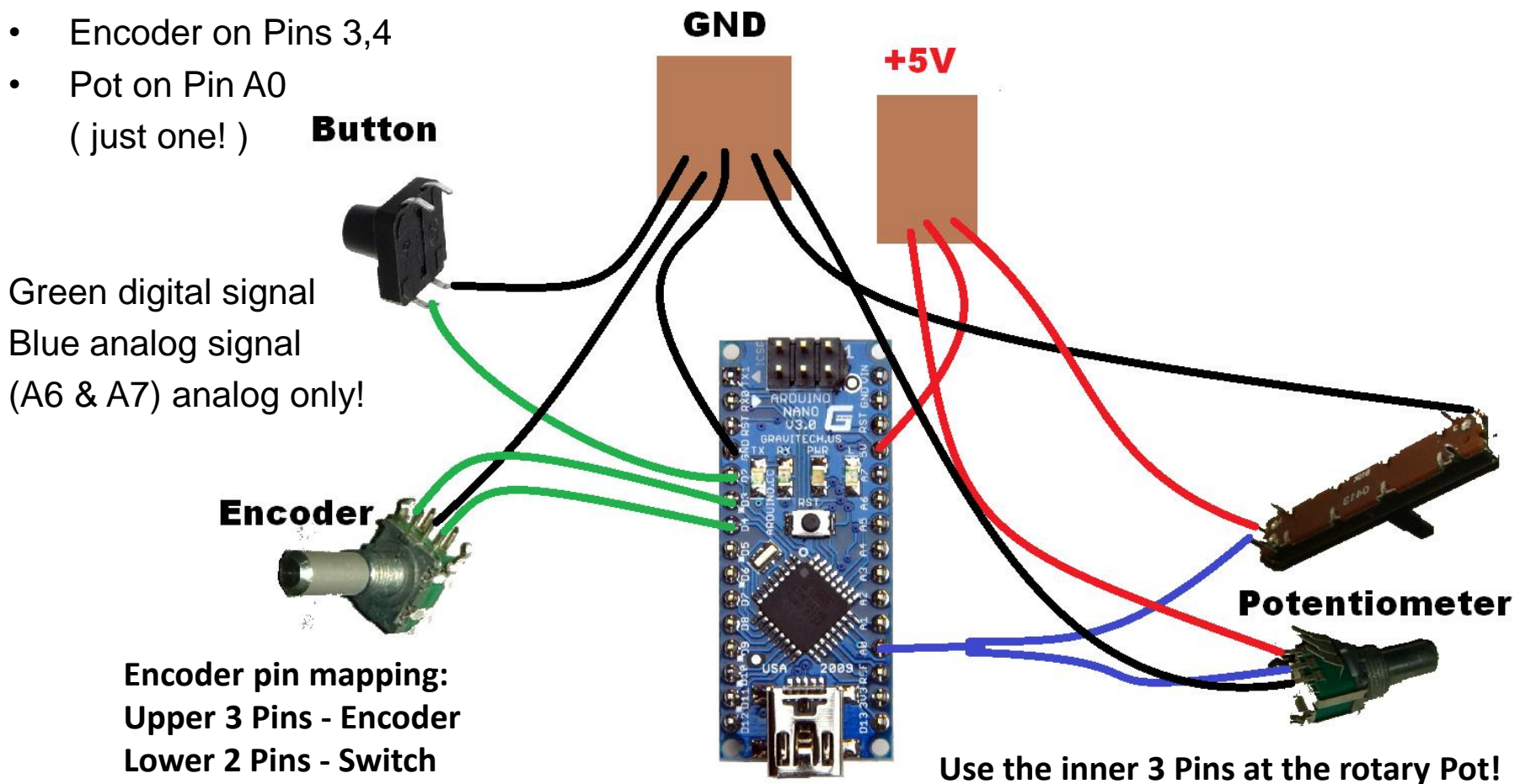
- Encoders



Electronics

- Basic-Circuit for the MIDI-Controller

- Button on Pin 2
- Encoder on Pins 3,4
- Pot on Pin A0
(just one!)



Programming

- SerialMIDIElements-Library
 - Classes which handle everything
 - Buttons, Potentiometers, Encoders
- Just three steps needed:
 1. Include MIDIElementsFSA header
 2. Setup a Button/Pot/Encoder
 3. Read specific Button/Pot/Encoder in Loop

Programming

```
#include <SerialMIDIElements.h>

boolean debug      = false;      // print to serial instead of midi
boolean secondary  = false;      // disabled secondary midi messages
int      midiChannel = 1;        // midi channel number

// setup a button on Arduino-pin 2 on for ControlChange 1
Button      but1(2,      midiChannel,1,secondary,debug);
// setup a Encoder on Arduino-pin 6 and 7 for ControlChange 11
MIDIEncoder  enc1(3,4,  midiChannel,11,secondary, debug);
// setup a Potentiometer on Arduino-pin analog0 on for ControlChange 21
Potentiometer pot1(A0,  midiChannel,21,secondary,debug);

void setup(){
    Serial.begin(115200);
}

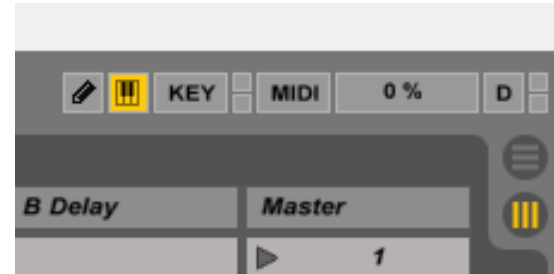
void loop(){
    // add here all the input component reads
    but1.read();
    enc1.read();
    pot1.read(); // read knob and send midi messages

}
```

Mapping

- Ableton

- Use Midi-Map Button



- Traktor

- Mapping in „Controller Manager“

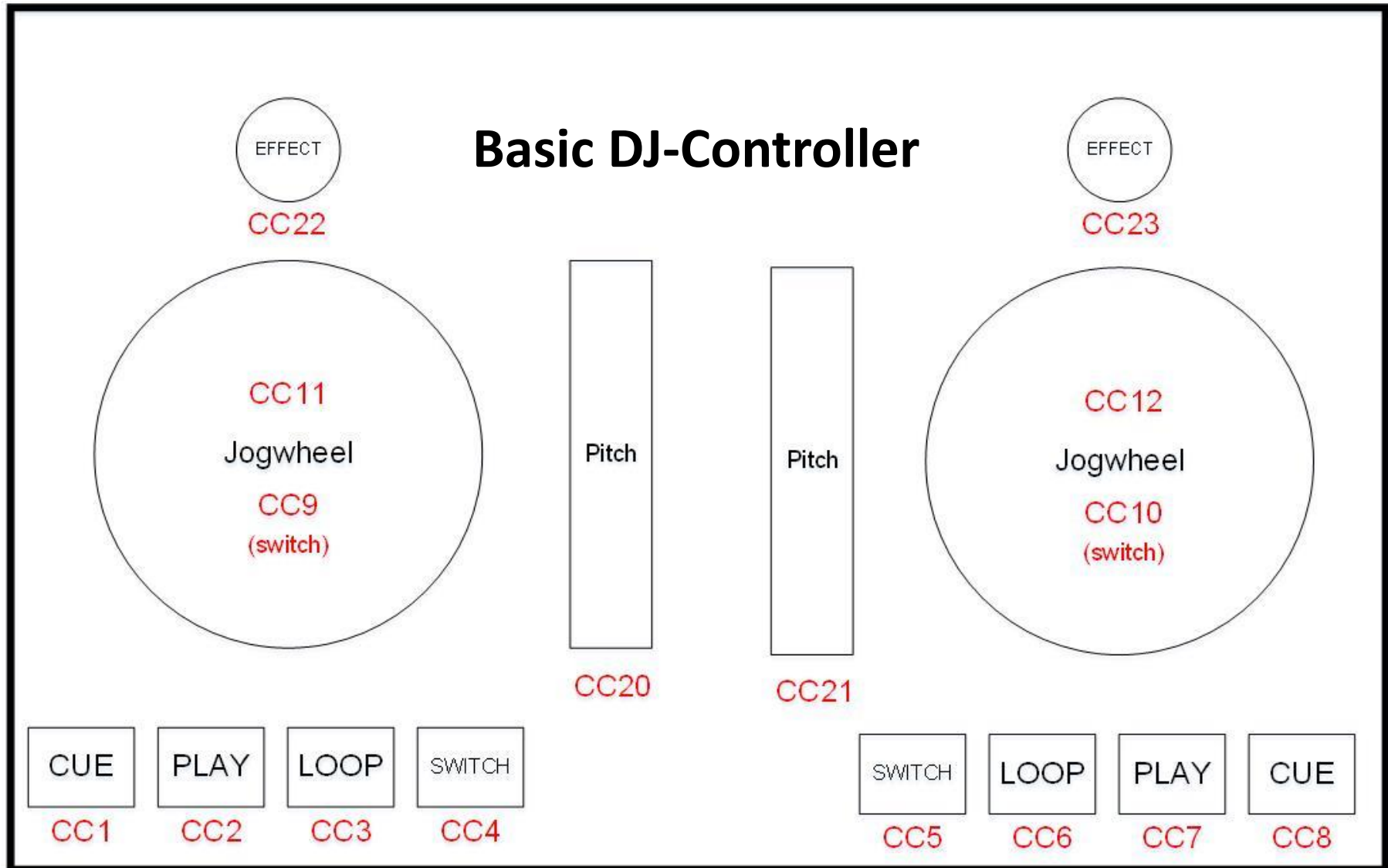


- Mixxx

- Learning-Assistant in Controller-Options

What could i build with the kit?

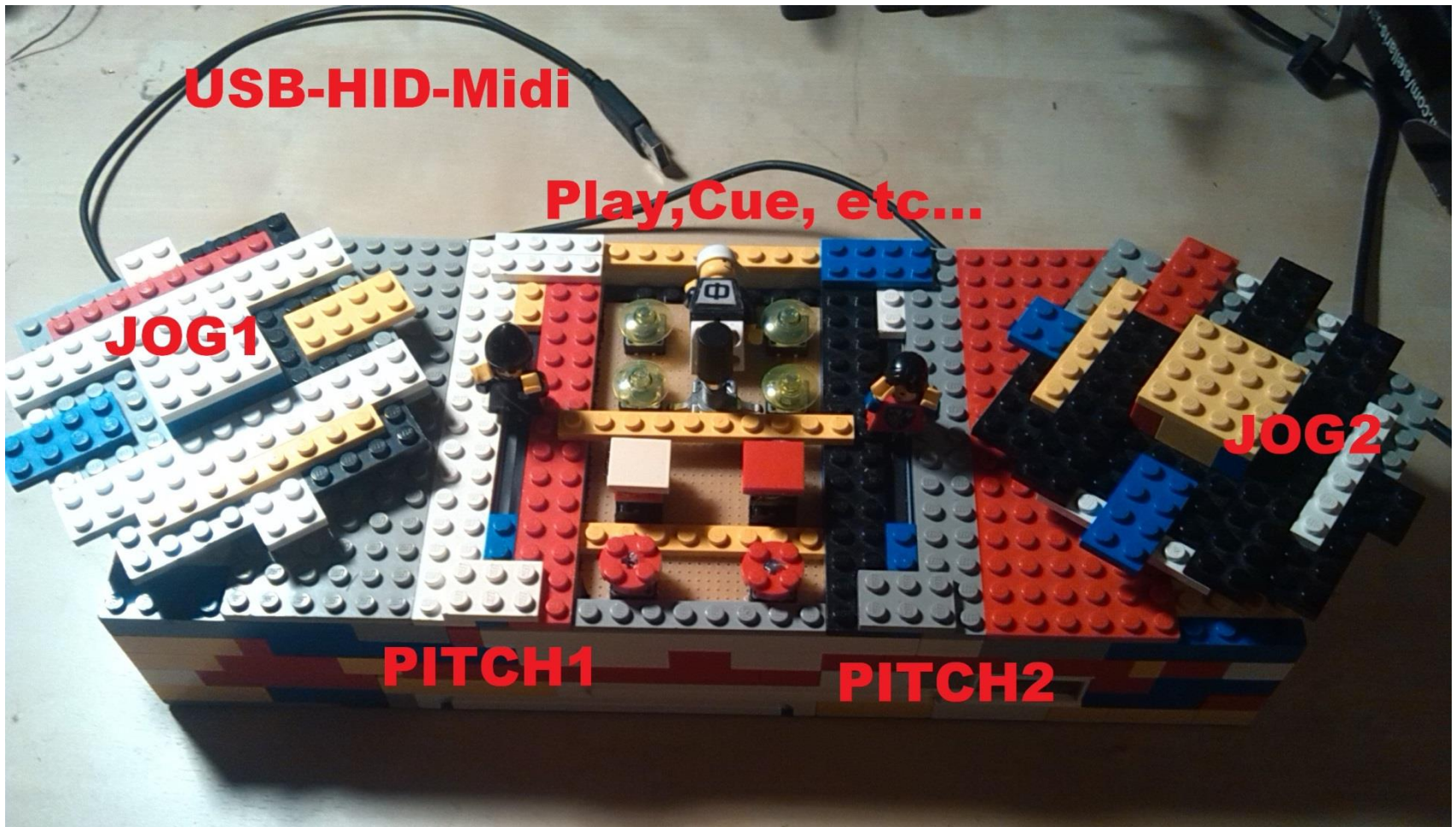
Basic DJ-Controller



Examples



Examples



USB-MIDI-Controller Tinkering

- Procedure:

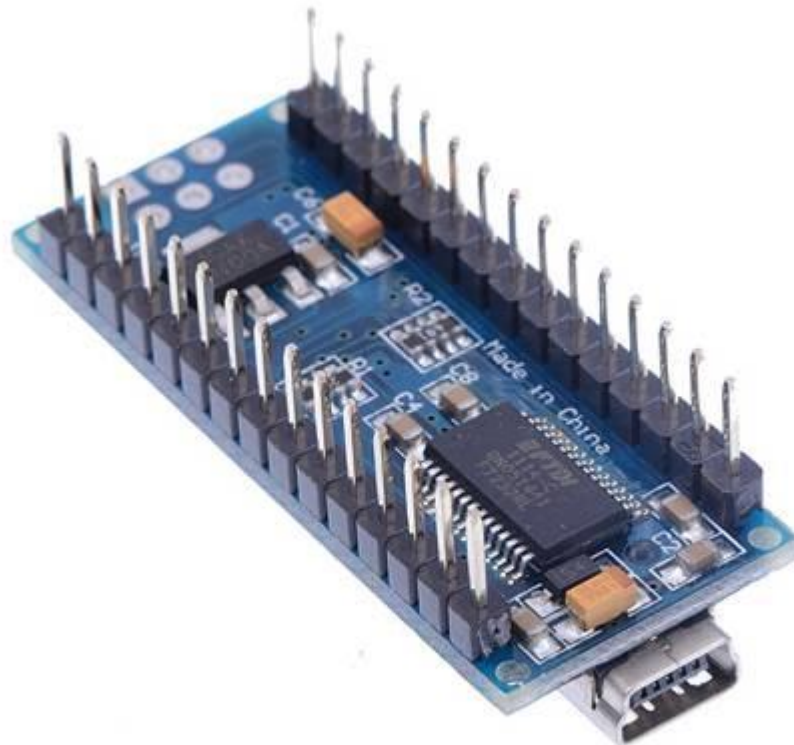
<https://github.com/julled/SerialMIDIElements>

1. Solder Pinheaders to the Arduino Nano
2. Wire one button/pot/encoder to the Arduino (see schematics)
3. Fix the +5V and GND plate with some tape at a piece of cardboard (prevent superevil short circuits...)
4. Install Arduino USB-Serial Chip-Driver CH340 (only Windows & MAC)
5. Get the SerialMIDIElements-Library from my Github and load it into the Arduino-SW
6. Load the basic-example from my Github:
SerialmidiElements_basic_example_31C3.ino
7. Install and start ttymidi / hairless-midiserial and debug input from button/pot/encoder
8. Create a virtual midi-port and use it in ttymidi / hairless-midiserial to publish midi-messages
9. Map the inputdevice to a controlelement in a MIDI-enabled SW e.g. in the MIXXX DJ-Software

Tinkering – Tips n Tricks

1. Solder the Pinheaders:

- It should look like this



Tinkering – Tips n Tricks

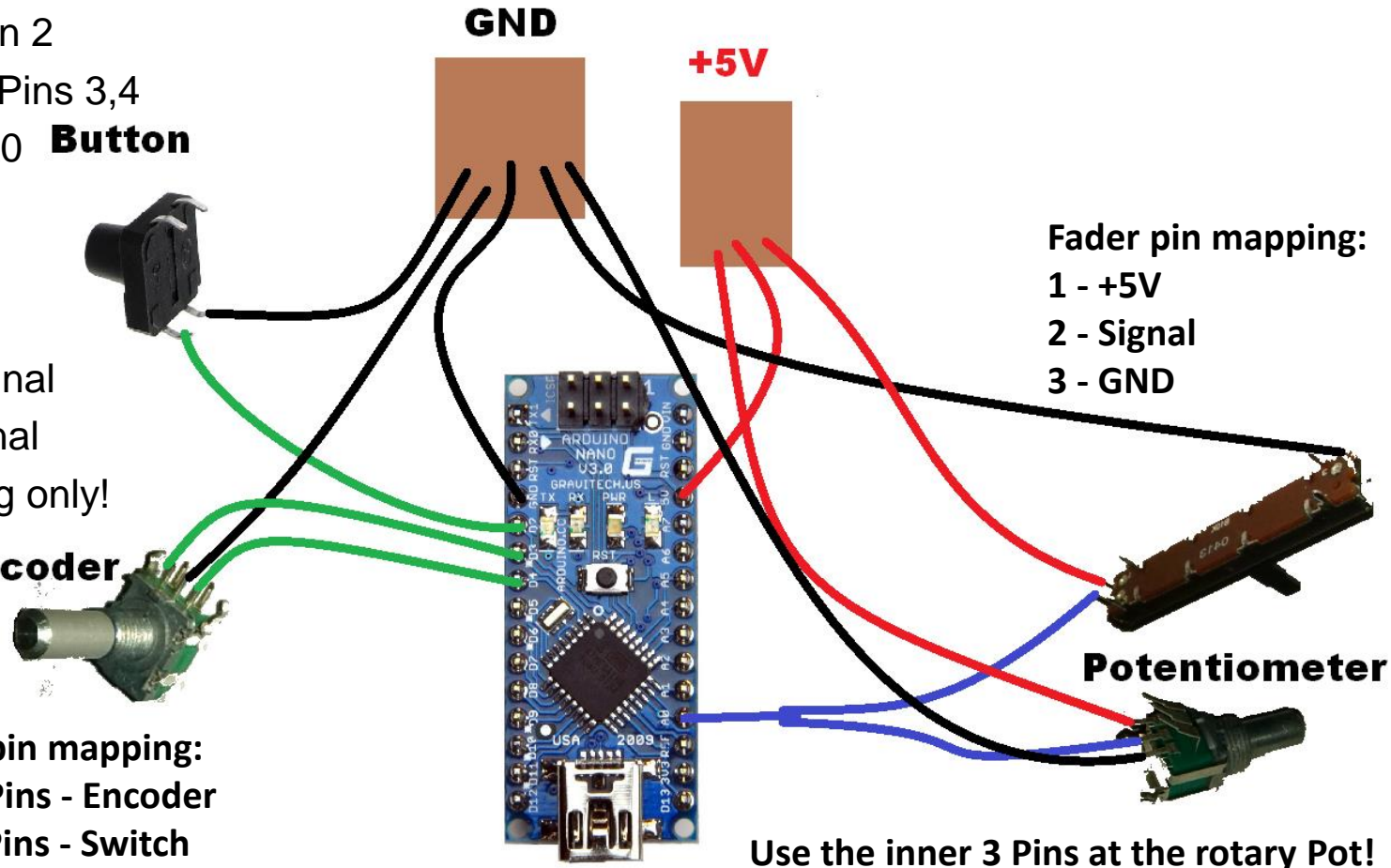
2. Wire the Basic-Circuit:

- Basic-Circuit:
- Button on Pin 2
- Encoder on Pins 3,4
- Pot on Pin A0 **Button**
(just one!)

Green digital signal
Blue analog signal
(A6 & A7) analog only!

Encoder

Encoder pin mapping:
Upper 3 Pins - Encoder
Lower 2 Pins - Switch



Tinkering – Tips n Tricks

4. Install USB-Serial Chip-Driver CH340:

- MAC OSX
 - <http://www.wch.cn/downloads.php?name=pro&proid=178>
 - On MAC OSX 10.9 & 10.10: `sudo nvram boot-args="kext-dev-mode=1"`
 - RESTART!!!
- Windows:
 - <http://www.wch.cn/downloads.php?name=pro&proid=5>
 - RESTART!!!
- Linux.
 - Normally no driver needed
 - Otherwise: <http://www.wch.cn/downloads.php?name=pro&proid=177>

Tinkering – Tips n Tricks

5. Get the SerialMIDIElements-Library:

- <https://github.com/julled/SerialMIDIElements>
- <https://github.com/julled/SerialMIDIElements/blob/master/zipped-Library/SerialMIDIElements.zip>
- Import it into Arduino: Arduino -> Sketch -> Import Library -> Add Library ... choose the SerialMIDIElements.zip

Tinkering – Tips n Tricks

6. Load Basic-Example:

- Open SerialmidiElements_basic_example_31C3.ino @ Github
- Upload Sketch to Arduino

Tinkering – Tips n Tricks

7. Install Serial-to-MIDI-Converter:

- HairlessMidi:
 - <https://projectgus.github.io/hairless-midiserial/>

OR

- ttymidi:
 - `sudo apt-get install ttymidi`

Tinkering – Tips n Tricks

8. Virtual MIDI Port:

- MAC: <http://www.johanlooijenga.com/tools/12-virtual-ports.html>
- Windows: <http://www.midiox.com/myoke.htm>
- Linux: no additional SW needed

Tinkering – Tips n Tricks

9. Mapping in MIDI-enabled SW

- e.g. MIXXX
- Download from mixxx.org
- Learning-Assistant in Controller-Options

Tinkering – Tips n Tricks

Documentation:

<https://github.com/julled/SerialMIDIElements>