

Hardware les 3 (en 4)

MIDI over USB

Teensy -> Max

Via Seriële verbinding

Arduino setup

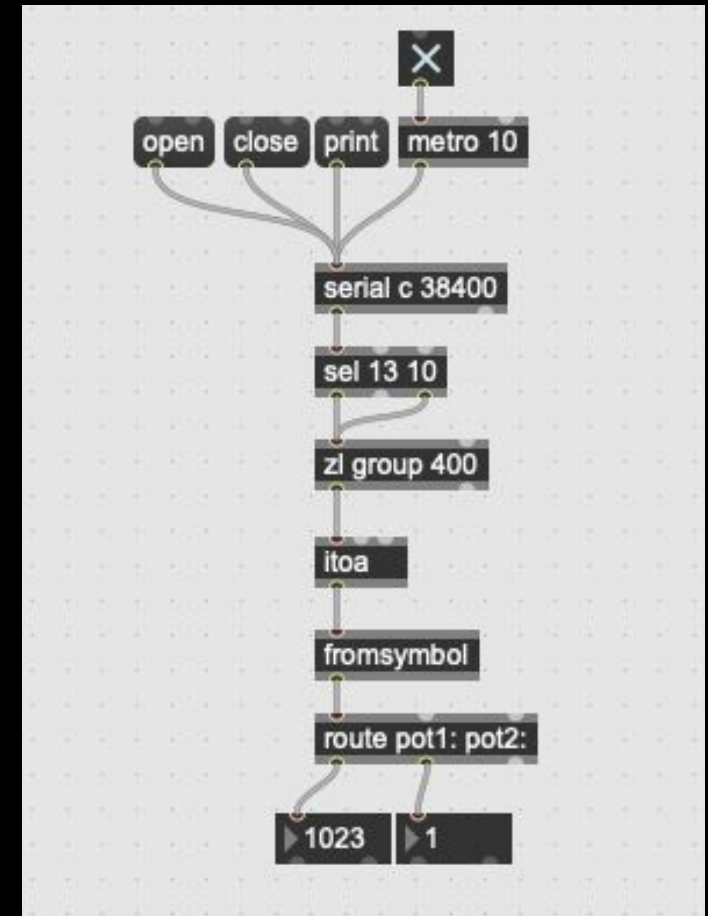
in de seriële monitor:

```
pot1: 123  
pot2: 251
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  int pot1Val = analogRead(A0);  
  int pot2Val = analogRead(A1);  
  Serial.print("pot1: ");  
  Serial.println(pot1Val);  
  Serial.print("pot2: ");  
  Serial.println(pot2Val);  
  delay(10);  
}
```



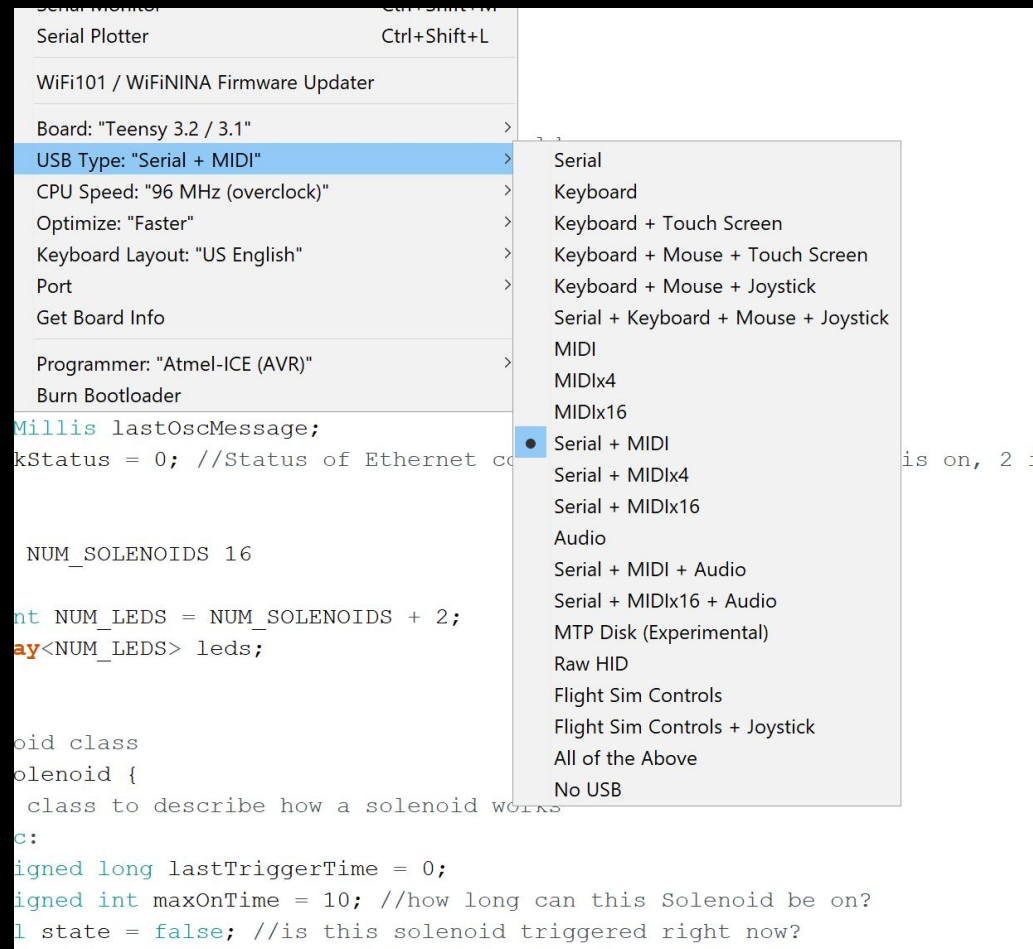
in Max:



En nu met MIDI!

(makkelijker maar minder accuraat)

USB Types



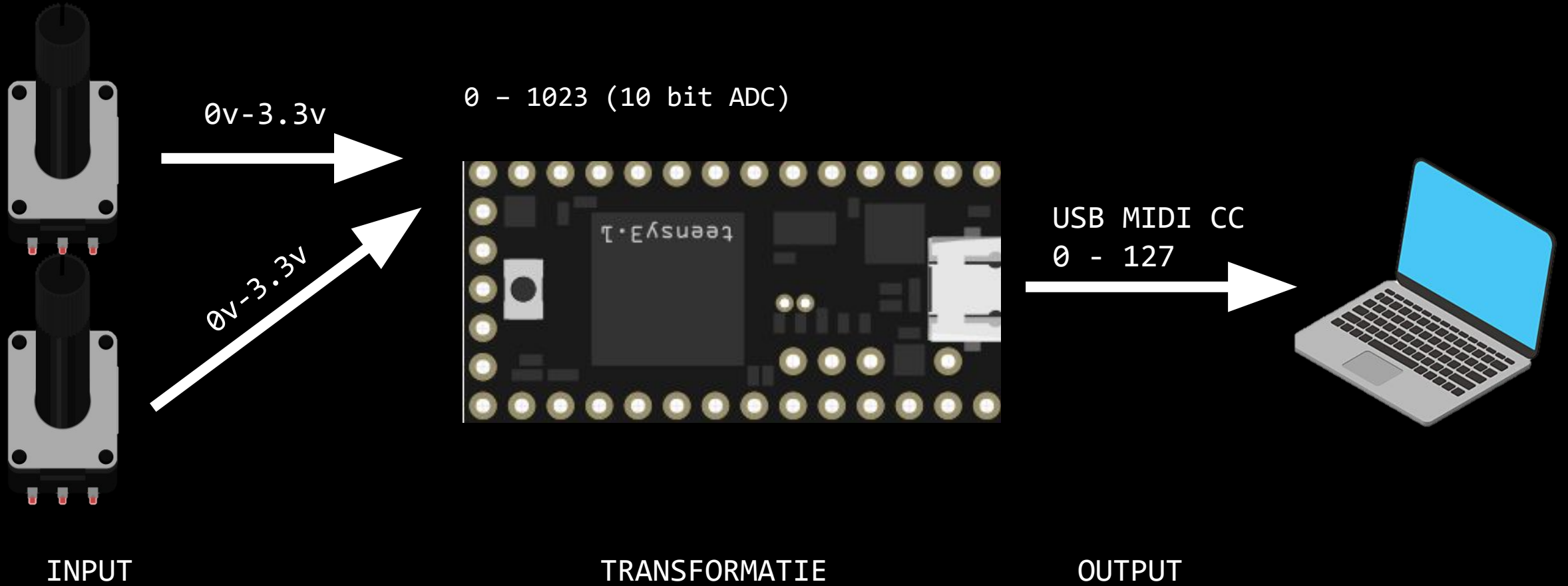
The screenshot shows a configuration window for a Teensy board. The 'USB Type' dropdown menu is open, displaying the following options:

- Serial
- Keyboard
- Keyboard + Touch Screen
- Keyboard + Mouse + Touch Screen
- Keyboard + Mouse + Joystick
- Serial + Keyboard + Mouse + Joystick
- MIDI
- MIDIx4
- MIDIx16
- Serial + MIDI
- Serial + MIDIx4
- Serial + MIDIx16
- Audio
- Serial + MIDI + Audio
- Serial + MIDIx16 + Audio
- MTP Disk (Experimental)
- Raw HID
- Flight Sim Controls
- Flight Sim Controls + Joystick
- All of the Above
- No USB

The background code in the IDE includes:

```
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFinINA Firmware Updater
Board: "Teensy 3.2 / 3.1"
USB Type: "Serial + MIDI"
CPU Speed: "96 MHz (overclock)"
Optimize: "Faster"
Keyboard Layout: "US English"
Port
Get Board Info
Programmer: "Atmel-ICE (AVR)"
Burn Bootloader
Millis lastOscMessage;
kStatus = 0; //Status of Ethernet co
NUM_SOLENOIDS 16
nt NUM_LEDS = NUM_SOLENOIDS + 2;
ay<NUM_LEDS> leds;
oid class
olenoid {
class to describe how a solenoid works
c:
igned long lastTriggerTime = 0;
igned int maxOnTime = 10; //how long can this Solenoid be on?
l state = false; //is this solenoid triggered right now?
```

MIDI Controller



MIDI berichten

- NoteOn
- NoteOff
- Control Change

Extra:

- Program Change
- After Touch
- Pitch Bend
- SysEx
- Song Position
- Clock
- Etc (zie [hier](#))

MIDI berichten

Control Change op Teensy

```
usbMIDI.sendControlChange(control, value, channel);
```

Control = index van controller (potmeter) (0 -127)

Value = positie / waarde van potmeter (0 -127)

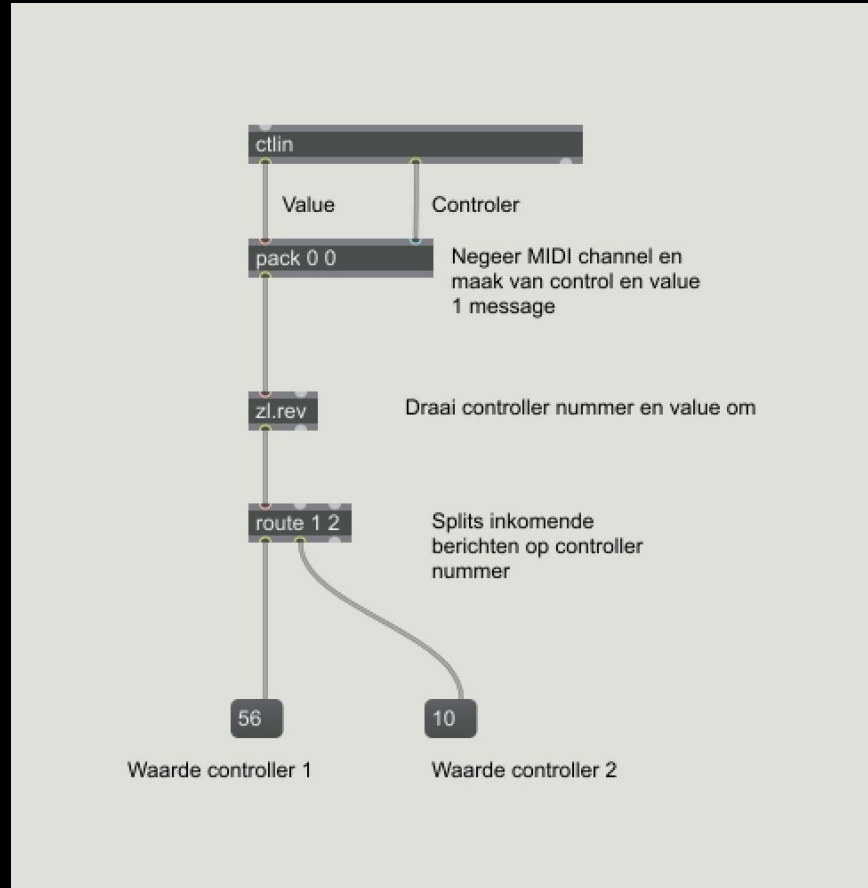
Channel = MIDI kanaal van bericht (1 - 16)

Ontvangen in Max

`ctlin`

- Broertje van `notein`
- Volgorde van `control`, `value` omgedraaid

Ontvangen in Max



EINDE LES 3

MIDI ontvangen op Teensy

Hoe kan je berichten naar je Teensy sturen over MIDI en daar iets mee doen?

MIDI binnenkrijgen op Teensy

```
void setup() {  
    Serial.begin(115200);  
    usbMIDI.setHandleControlChange(myControlChange);  
}
```

```
void loop() {  
    usbMIDI.read();  
}
```

```
void myControlChange(byte ch, byte cntl, byte val) {  
    int channel = int(channel);  
    int control = int(control);  
    int value = int(val);  
    Serial.println("Channel " + String(ch) + " controller " + String(control) + " value: " + String(value));  
}
```

Waar kan je naar luisteren op een Teensy?

```
usbMIDI.setHandleNoteOff(myNoteOff)
usbMIDI.setHandleNoteOn(myNoteOn)
usbMIDI.setHandleAfterTouchPoly(myAfterTouchPoly)
usbMIDI.setHandleControlChange(myControlChange)
usbMIDI.setHandleProgramChange(myProgramChange)
usbMIDI.setHandleAfterTouch(myAfterTouch)
usbMIDI.setHandlePitchChange(myPitchChange)
usbMIDI.setHandleSystemExclusive(mySystemExclusiveChunk);
usbMIDI.setHandleTimeCodeQuarterFrame(myTimeCodeQuarterFrame);
usbMIDI.setHandleSongPosition(mySongPosition);
usbMIDI.setHandleSongSelect(mySongSelect);
usbMIDI.setHandleTuneRequest(myTuneRequest);
usbMIDI.setHandleClock(myClock);
usbMIDI.setHandleStart(myStart);
usbMIDI.setHandleContinue(myContinue);
usbMIDI.setHandleStop(myStop);
usbMIDI.setHandleActiveSensing(myActiveSensing);
usbMIDI.setHandleSystemReset(mySystemReset);
usbMIDI.setHandleRealTimeSystem(myRealTimeSystem);
```

Maar FL Studio / Logic / Ableton / Reaper
/ Audacity / Cubase / Studio One / Reason
/ Bitwig / GarageBand / Mixcraft / Pro
Tools / Cakewalk / een MIDI learn
programma snapt hier niets van

Waarom?

Slimme MIDI sturen

- Niet altijd sturen
- Ruis filteren
- Weinig latency behouden

Niet altijd sturen

- Alleen MIDI sturen als MIDI waarde verandert

Aan het einde van loop()

```
_potValue = potValue;
```

In loop():

```
if (potValue != _potValue) {  
    //als de huidige potValue anders is dan de vorige potValue,  
    //stuur dan een MIDI CC bericht  
    usbMIDI.sendControlChange(1, potValue, 16);  
}
```

Analoge ruis

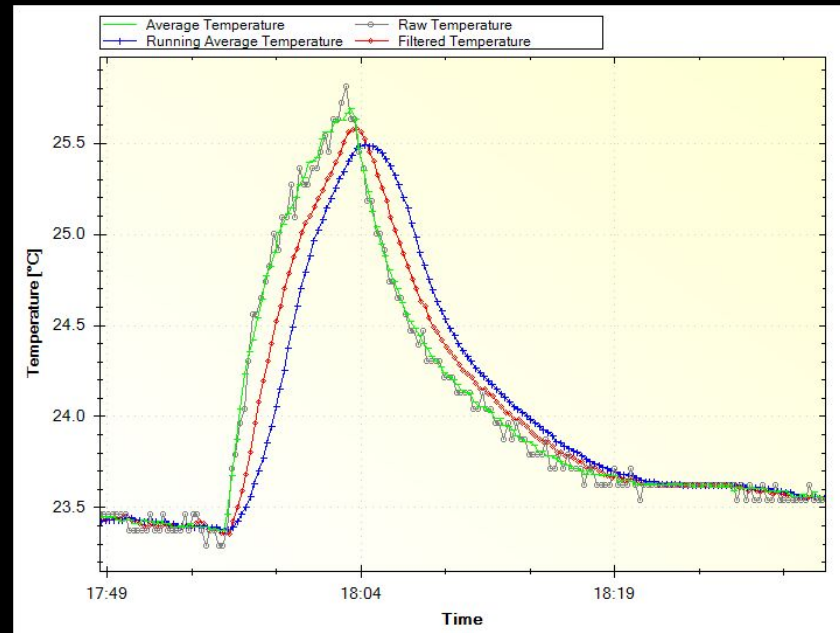
- Onontkoombaar
- USB voeding ruis
- Netstroom ruis
- Externe ruis

Analoge ruis (filteren)

- Analooq filteren is het beste
- Goedkoper en makkelijker is digitaal filteren

MAAR

digitaal filteren -> latency



Analoog ruis (filteren 2)

- Vooral snog makkelijkst in Max op te lossen
- Anders lopend gemiddelde nemen of digitaal filter
- Zie [hier voor meer info](#) of [hier](#)

MIDI USB naam

- name.c maken in nieuw tabblad

```
// To give your project a unique name, this code must be
// placed into a .c file (its own tab). It can not be in
// a .cpp file or your main sketch (the .ino file).

#include "usb_names.h"

// Edit these lines to create your own name. The length must
// match the number of characters in your custom name.

#define MIDI_NAME    {'M','y',' ','M','I','D','I'}
#define MIDI_NAME_LEN 7

// Do not change this part. This exact format is required by USB.

struct usb_string_descriptor_struct usb_string_product_name = {
    2 + MIDI_NAME_LEN * 2,
    3,
    MIDI_NAME
};
```

Demonstratie