

# NonagonBar

electronica & software

## Raspberry Pi

- OS versie: Raspbian Buster with desktop (GNU/ Linux 10, 26-09-2019 )  
<https://www.raspberrypi.org/downloads/raspbian/> (zonder recommended software)
- VNC viewer (versie: 6.20.113) <https://www.realvnc.com/en/connect/download/viewer/>
- Pure Data 0.49.0-3 (installatie: `sudo apt install puredata`)
- Pure Data: library: cyclone (`sudo apt install pd-cyclone` of: installeer met de deken-installer in de pd-gui.)

## Arduino

### Libraries:

OSC

source: <https://github.com/CNMAT/OSC>

Trill

source: <https://github.com/BelaPlatform/Trill/tree/master/lib/Arduino/Trill>

Belangrijke setup functies en variabelen:

### OSC:

IP adres pi (ethernet port)

IP adres arduino (je wijst zelf een IP adres toe aan de arduino. De eerste 3 getallen moeten overeenkomen met dat van de Pi.)

### Setup:

**prescaler**, (int) betreft de gevoeligheid van de sensor. “rule of thumb” : hogere waarden zijn beter voor grotere objecten of objecten met meer weerstand, kleinere waarden zijn beter voor proximity sensing.

**noiseThreshold**, (int 0-4096) drempelwaarde van waar iets als data te lezen.

**scanSettings**: scansnelheid van de sensor: 3 = slow, 2 = normal, 1 = fast, 0 = ultra fast

(bovenstaande waarden zijn aan te passen in in Trill.h , regel 96-104)

**readSpeed**: scan & verzend OSC bundels elke x ms.

**kThreshold**: drempelwaarde voor detectie van beweging. gebaseerd op het verschil tussen huidige sensorwaarde en de vorige gemeten sensorwaarde.

**NewDataMultiplier & oldDataMultiplier**: zie beschrijving volgende pagina

## Arduino Code:

In de arduino code wordt de sensordata uitgelezen. er gebeuren 2 belangrijke dingen in dit proces:

1 de functie  $\text{newData} = (\text{newDataMultiplier} * \text{value}) + (\text{oldDataMultiplier} * \text{sensList}[\text{index}].\text{oldData})$

deze functie werkt als een soort filter. "value" is de direct uitgelezen data, en "sensList[index].oldData" is de vorige waarde.

newDataMultiplier + oldDataMultiplier moet dus gelijkstaan aan 1.

hoe hoger de variabele newDataMultiplier, hoe meer je van de directe waarde verstuurt, en hoe sneller alles reageert.

door oldDataMultiplier hoger te zetten, kun je de waarden "smoothen";

de data die wordt verstuurd loopt langzamer naar de huidige waarde toe wanneer oldDataMultiplier hoger is.

in de praktijk worden deze waarden al snel (onbruikbaar) langzaam wanneer oldDataMultiplier wat hoger staat,

maar het kan een leuk idee zijn om zowel de directe waarden als de gefilterde waarden uit te lezen, waardoor je een soort envelope followers zou kunnen maken.

met deze reden heb ik de code met een `//@rawData` comment laten staan, wanneer je alle lines met deze comment aanzet, wordt de lijst "rawData" ook over OSC naar pd verstuurd.

het verschil tussen de vorige waarde en huidige waarde wordt berekend (`sensList[index].delta`).

als het verschil groter is dan "kThreshold", wordt er beweging geregistreerd.

(0 = geen beweging, 1 = beweging naar de sensor toe, 2 = beweging van de sensor af)

de sensordata wordt omgezet naar de volgende lijsten, die naar PD worden verstuurd met OSC:

**"Data"** : de "gefilterde" sensordata.

**"Change"** : de beweging per sensor. 0 = geen beweging, 1 = beweging naar de sensor toe 2 = beweging van de sensor af.

**"rawData"**: directe sensordata

## Pure Data:

In PD worden de waarden verder gescaled, om er audio triggers van te maken.

De lijst "Data" komt in pd binnen en gaat als eerste door de setBase functie heen: deze wordt geïnitieerd wanneer er 2 seconden geen beweging wordt gedetecteerd (input van de "Change" lijst). de huidige waarde wordt dan omgezet naar de minimumwaarde in de scaling. (de snelheid hiervan is aan te passen door setBaseCounter te veranderen)

- De minimumwaarden van de scaling staan standaard op 0 en worden telkens (per sensor) bijgewerkt door de setBase functie
- De maximumwaarden van de scaling moeten van tevoren worden ingesteld op basis van testresultaten. (te doen: automatische scaling maken voor hoogste waarden? > idee van Mark)
- de uiteindelijke scaling gebeurt in de Scale.cl patch. enkel de 3 sensoren met de hoogst inkomende waarden worden gescaled.