

Wind Instrument Building

Marije Baalman

May 10, 11, 12

Overview

- Day 1: focus on making sensors
- Day 2: focus on receiving and processing data
- Day 3: focus on connecting the data to sound and visuals

Day 2 - receiving and processing data

On the second day of the workshop, we will focus on using the Sense/Stage MiniBee platform to receive the data from the sensor, while the kite is in the air.

We will learn how to view and understand the data in *Processing*, and learn some simple methods to process the data.

The **Sense/Stage MiniBee platform** is a wireless sensing platform created for artists, based on the *Arduino* platform. The sensing board includes an accelerometer, and can be used to attach various sensors to it. The preprogrammed firmware, along with the software allows for easy configuration of the hardware, so you can immediately start receiving data. For an introduction see:

<https://docs.sensestage.eu/sensestage-v1/overview-of-the-system>

At the end of this day, participants will have a basic setup to receive and analyse data from the sensors in realtime, and have learnt some basic methods for processing the data.

Day 2 - receiving the wireless data and processing the data

- Morning: set up to receive data
 - *Introduction to Sense/Stage MiniBee platform*
 - Installation of software
 - Finish soldering for your sensors and connections
 - Testing connections
- Afternoon: process data
 - *Introduction*
 - Try out processing data
 - View output
 - *Share observations*

Sense/Stage MiniBee platform

What is the Sense/Stage MiniBee?

The Sense/Stage MiniBee platform is a wireless microcontroller to send and receive data to self-made devices.

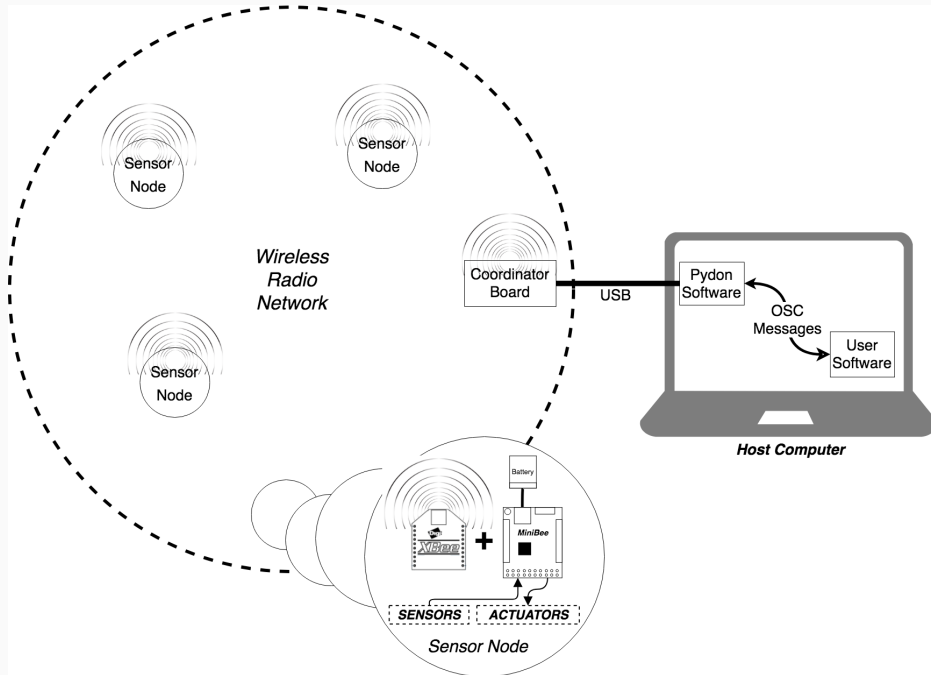
- open source
- based on the *Arduino* platform
- uses *Digi XBee*-radios for wireless communication
- provides basic firmware and software to easily configure and start receiving data from sensors quickly

What is the Sense/Stage MiniBee?

The Sense/Stage MiniBee platform is a wireless microcontroller to send and receive data to self-made devices.

- open source
- based on the *Arduino* platform
- uses *Digi XBee*-radios for wireless communication
- provides basic firmware and software to easily configure and start receiving data from sensors quickly
- you can extend the firmware yourself, while still using the provided software to send and receive data
- the hardware board has an accelerometer built-in

Setup of the wireless network



Open Sound Control (OSC)

The software *pydon* sends out messages using **Open Sound Control (OSC)**.

Open Sound Control (OSC)

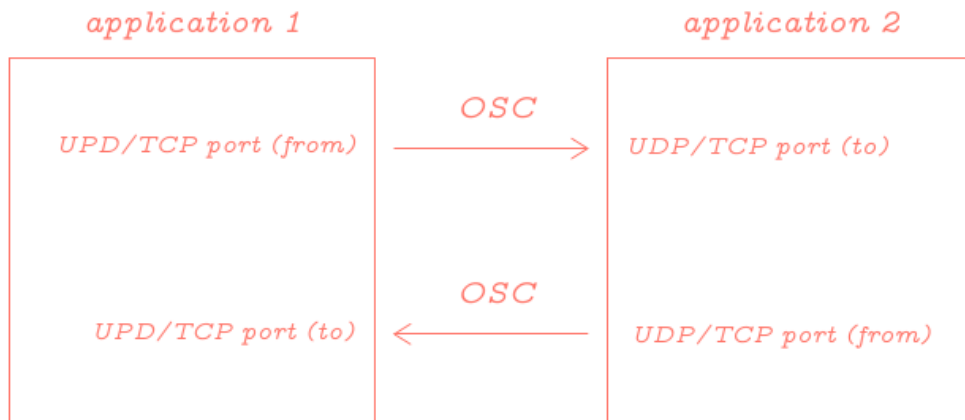
The software *pydon* sends out messages using **Open Sound Control (OSC)**.

- OSC is a protocol that can be sent over a network. It uses UDP packages.
- OSC is implemented in many creative coding environments, e.g. Processing, SuperCollider, PureData (Pd), Max/MSP, Ableton/Max4Live, TouchDesigner, Isadora, vvvv, ...

Open Sound Control (OSC)

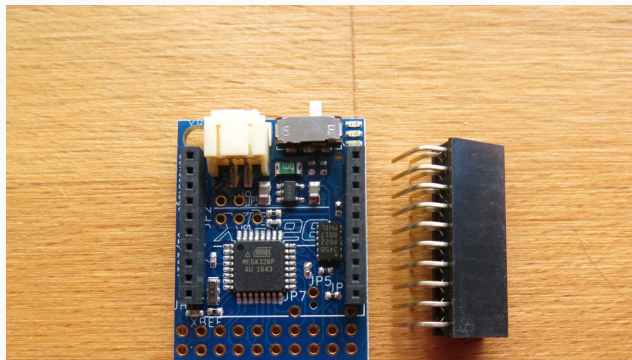
The software *pydon* sends out messages using **Open Sound Control (OSC)**.

- OSC is a protocol that can be sent over a network. It uses UDP packages.
- OSC is implemented in many creative coding environments, e.g. Processing, SuperCollider, PureData (Pd), Max/MSP, Ableton/Max4Live, TouchDesigner, Isadora, vvvv, ...



Overview of the MiniBee board

- 6 analog inputs: **A0, A1, A2, A3**, A6, A7
- 8 digital inputs or outputs: D2, D3, D5, D6 and D8 to D11
- PWM (“analog”) output, at pins D3, D5, D6, D9, D10, D11
- TWI/I2C communication: SDA (A4), SCL (A5) - to connect to digital sensors
- Serial I/O: RX, TX (to XBee)
- Power input (between 3.3V and 16V)
- Regulated power output: 3.3V, GND
- **Accelerometer** (ADXL345) on the board



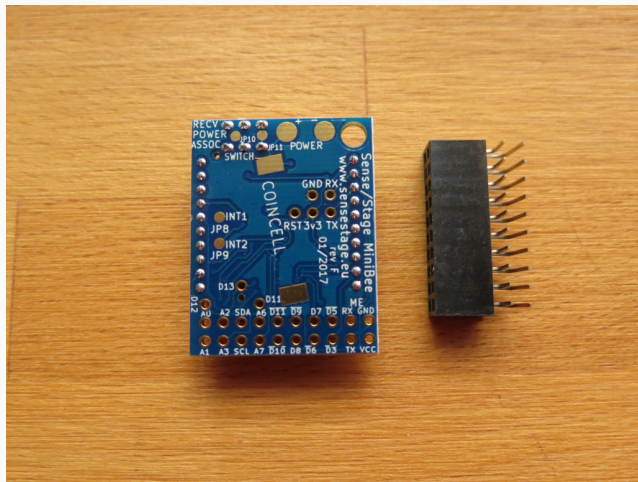
MiniBee pin out

The pin outs on the header are from left to right:

GND - RX - D5 - D2 - D9 - D11 - A6 - SDA - A2 - A0

3v3 - TX - D3 - D6 - D8 - D10 - A7 - SCL - A3 - A1

On the bottom of the board, the pin labels are also printed. *By mistake pin D2 is labeled as D7.*



Installation guide at

<https://docs.sensestage.eu/sensestage-v1/pydon-installation-guide/>

Also installation videos are available for Mac OSX and Windows

Installation guide at

<https://docs.sensestage.eu/sensestage-v1/pydon-installation-guide/>

Also installation videos are available for Mac OSX and Windows

Note

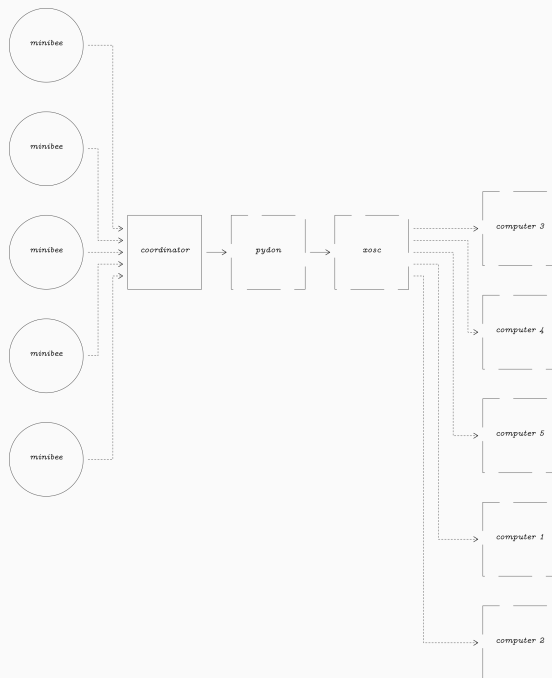
- on more recent versions of Mac OSX *python2* is not installed by default anymore. . . unfortunately, this makes the installation a bit more complex, as the software is not compatible with *python3*

```
pydoncli.py -s /dev/tty.usbserial-DARSDF300 -c /path/to/wind-  
instrument-building/minibee/kite-basic-config.xml -t  
12000
```

Arguments are:

- -s path to the serial port
- -c path to the configuration file
- -t port of the receiving application

Receiving data through my computer



Receiving data from my computer (that receives data from the MiniBees)

Alternative - if the installation of *pydon* is too complex. I can set up my computer to receive the data from the wireless sensor, and we can forward the OSC data from there to your laptop(s).

Receiving data from my computer (that receives data from the MiniBees)

Alternative - if the installation of *pydon* is too complex. I can set up my computer to receive the data from the wireless sensor, and we can forward the OSC data from there to your laptop(s).

Alternative - if the installation of *pydon* is too complex. I can set up my computer to receive the data from the wireless sensor, and we can forward the OSC data from there to your laptop(s).

To connect to the datastream:

- send OSC message 10.18.0.162 at the port 57300
 - `/XOSC/subscribe/host "10.18.0.162" 57120 <your port>`
 - `/XOSC/subscribe/tag "/minibee/data" <your port>`

```
/minibee/data id val_1 val_2 val_3 val_4 acc_x acc_y acc_z
```

For example:

```
/minibee/data 3 0.67 0.93 0.167 0.452 0.5103 0.4823 0.5032
```

Libraries to install:

- oscP5
- grafica

Code available at:

- <https://github.com/sensestage/wind-instrument-building>

Quarks to install:

- DataNetwork
- SenseWorld

Code available at:

- <https://github.com/sensestage/wind-instrument-building>

Exploring and scaling the range of data

What is the range of the data?

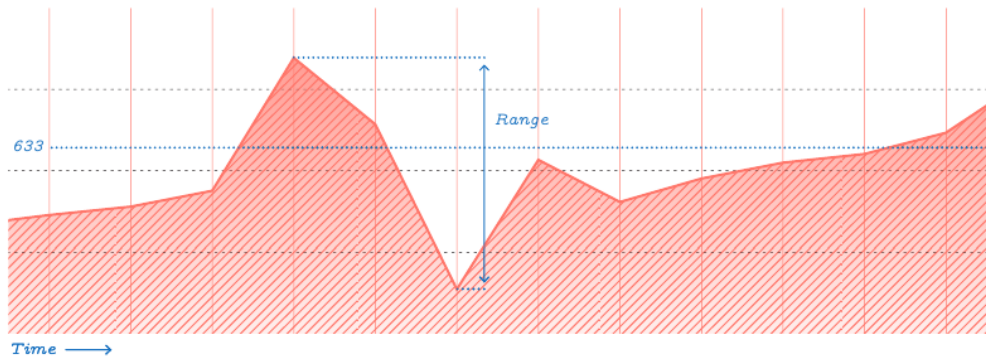


Figure 17.6 — Determining the range of the data

Scaling the data

$$V_{out} = \frac{V_{in} - V_{min}}{V_{max} - V_{min}}$$

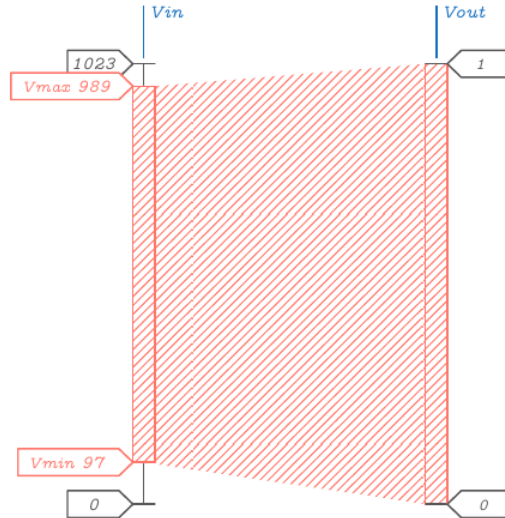


Figure 18.1 — Scaling the sensor input range (97 to 989) to a standardized range

Scaling and inverting the data

$$V_{out} = 1 - \frac{V_{in} - V_{min}}{V_{max} - V_{min}}$$

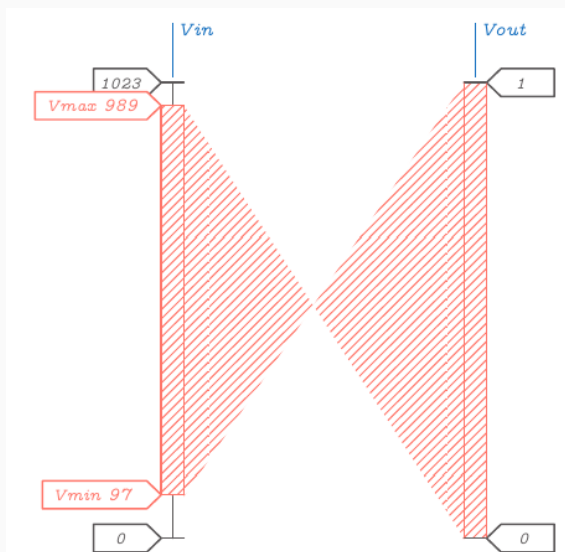


Figure 18.4 — Scaling of the sensor input range (97 to 989) with inversion to the output range (compare with figure 18.1)

<https://marijebaalman.eu>

- iii - instrument inventors initiative
 - National Taiwan Science Education Center
-
- Some images are taken from the book *Composing Interactions*
 - the research for the book *Composing Interactions* was funded by Creative Industries Fund NL