Materials in Music

Design Workshop

Jianing Zheng, Sophie Skach, Adan L. Benito

Workshop Overview

- Introduction
- Sharing session
- Introduction to Interactive Textiles
- Introducing Bela and Pure Data
- Prototyping and Making - Building sensor and connecting the sensor to Bela
- Sharing and Wrap-up

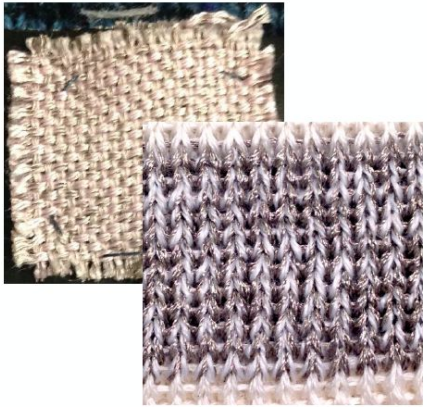
Textile Materials

- Non-conductive materials:
    - cotton, wool, linen, silk, …
    - nylon, polyester, acrylic, …
    - .
- Conductive materials:
    - copper, silver, steel, iron, …
    - carbon, graphene, …

Textile Structure

fibre / filament

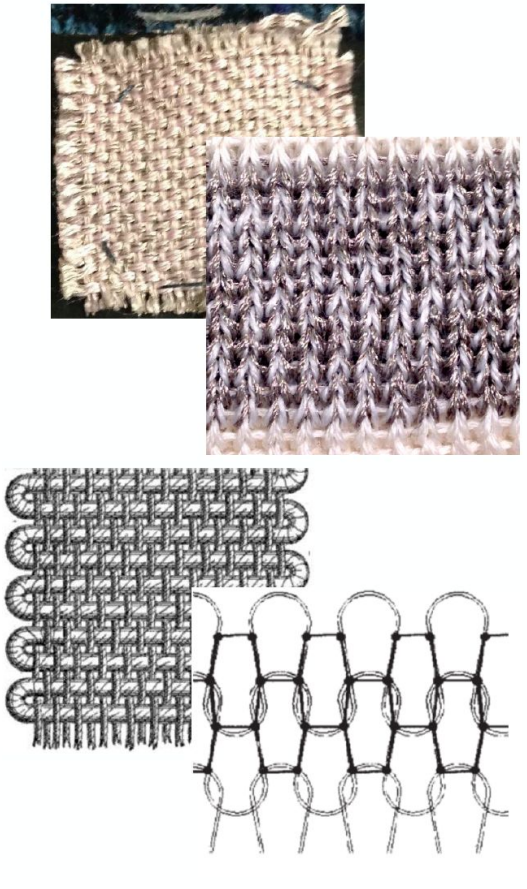
yarn / thread



fabric / textile surface

Textile Structure

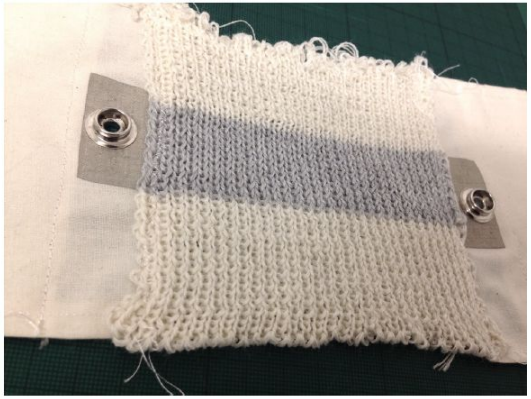
- weaving
- knitting
- felting
- braiding
- crocheting
- embroidering
- …

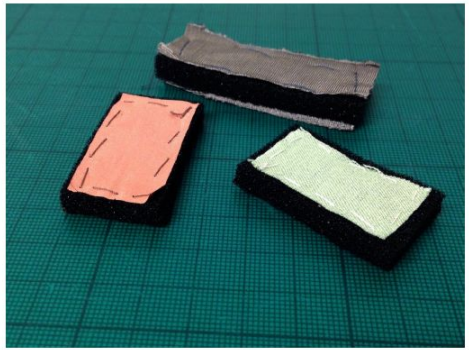


Textile Sensors

Examples

- Switches
- Capacitive sensors
- Piezo resistive sensors (pressure, stretch)
- Piezo electric sensors
- Textile speakers
- Antennas





Applications

Examples in music

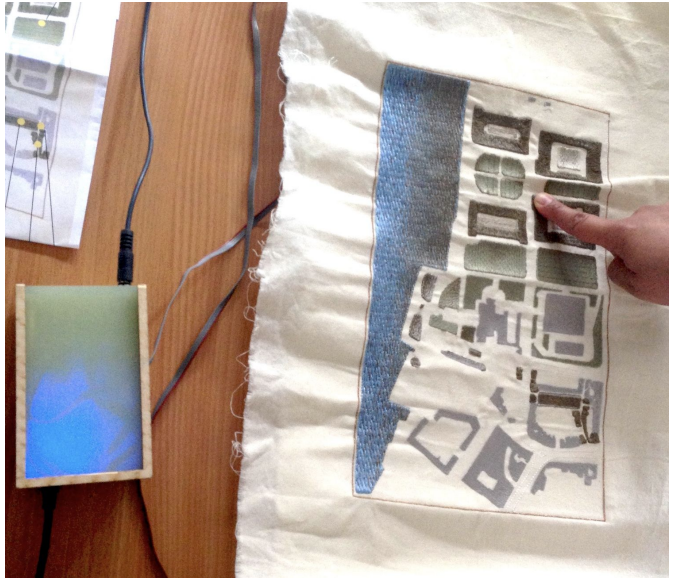
Mazi, collaborative sound interface
(Antonella Nonnis)

Soft Knit Biosensing for Augmenting Vocal Performances
(Courtney N. Reed, Sophie Skach, et al)

Applications

Examples in music

Text and Iles (Sophie Skach)



Aural Fabric (Alessia Milo)

Applications

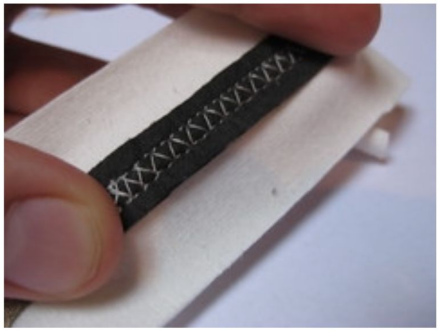
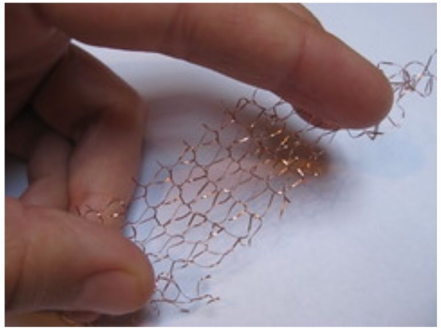
Examples in music

KnittedKeyboard II
(Irmandy Wicaksono, MIT Media Lab)

Data Gloves
(Hannah Perner Wilson, Mika Satomi, KOBAKANT)

Examples

Stretch Sensor and switch

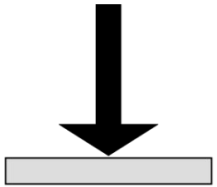

Constructed stretch sensors,
by KOBAKANT DIY Wearable Technology Documentation

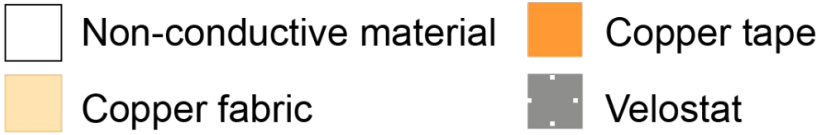


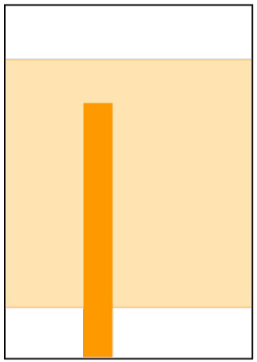
Fingertip contact switch,
by KOBAKANT DIY Wearable Technology Documentation

Pressure Sensor

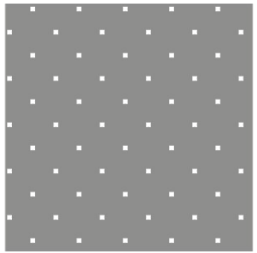
Build a pressure sensor



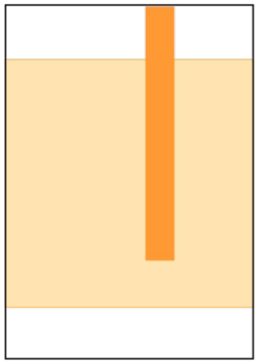




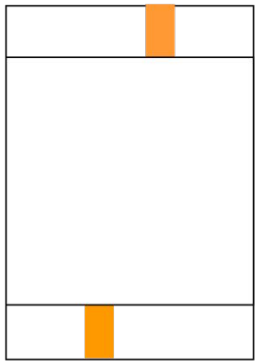
Top layer

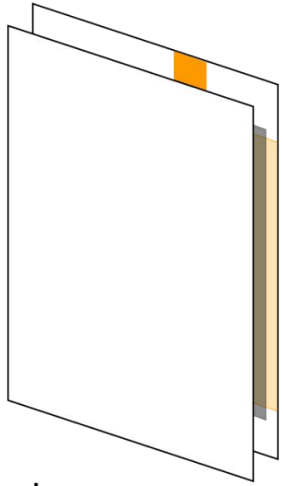


Mid layer



Bottom layer





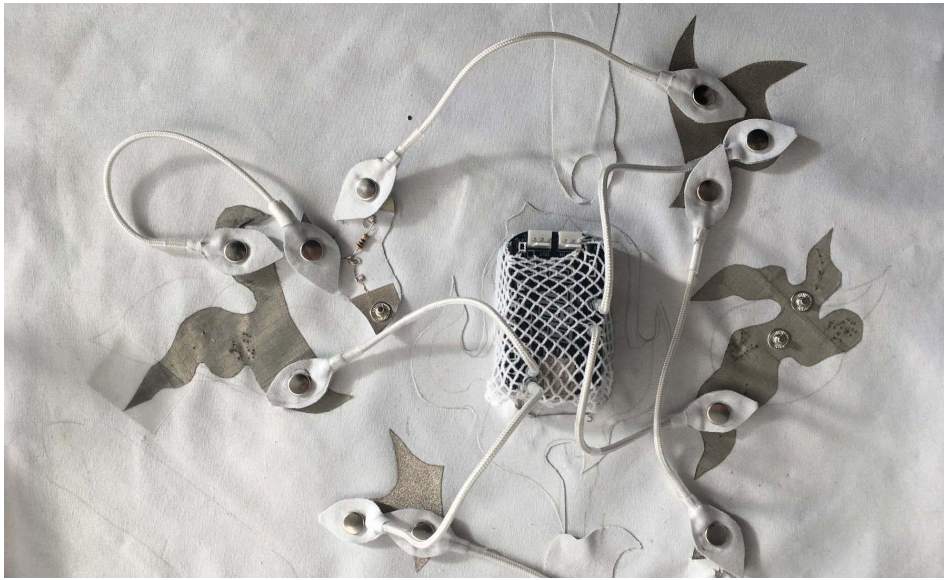
Overall look

Useful websites

- KOBAKANT. Hannah Perner Wilson and Mika Satomi. https://www.kobakant.at/DIY/
- E-Textile Summercamp Swatch Exchange: https://etextile-summercamp.org/swatch-exchange/
- Data Paulette: https://datapaulette.org/
- Bela Blog: https://blog.bela.io/e-textiles-and-embodied-interaction-with-bela/
- Embelashed Paper Sensors: https://embelashed.org/paper/paper-index.html#paper-sensors

Bela

https://bela.io/



Making music with e-textiles
(Nicola Woodham)

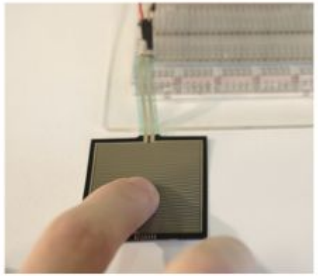
Cutting-edge e-textile research with Bela
(Sophie Skach, Rebecca Stewart, et al)

What is Bela?

Embedded computer designed for interactive audio

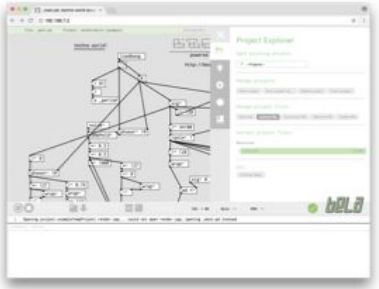
- Power of a single board Linux computer
- Connectivity of a micro-controller
- Combines the benefits of both

New approach to high-bandwidth sensor processing



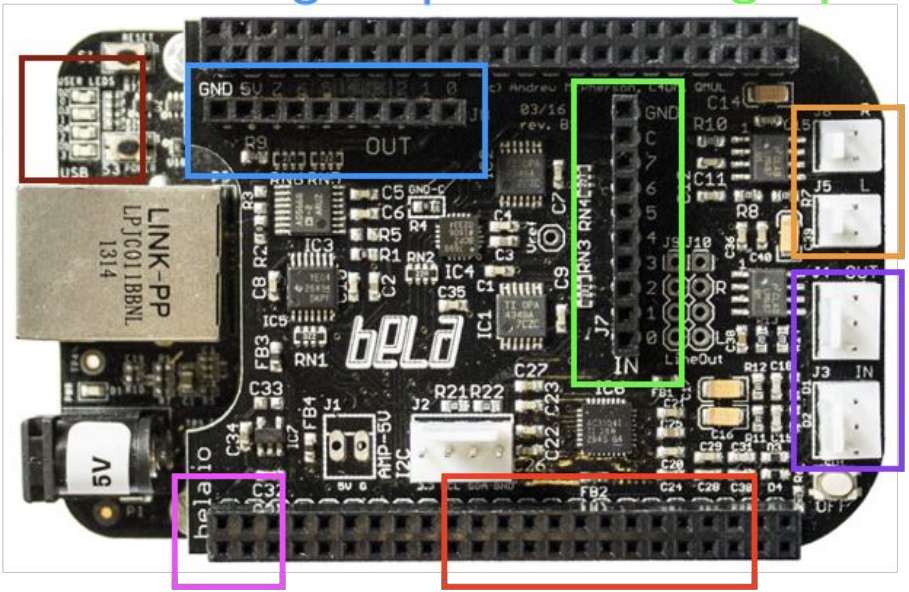
- Analog, digital I/O sampled at audio rate
- Ultra low action-sound latency
- Jitter-free alignment between audio and sensors

Open-source maker platform

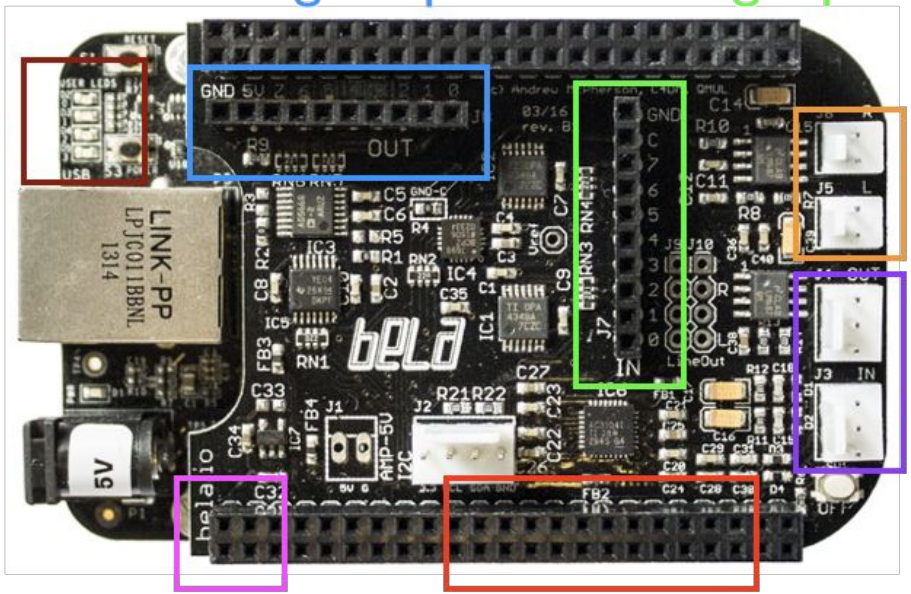


- Open hardware and software
- Targeted at musicians, artists
- Online community resources: forum, wiki, blog.

Bela Hardware



Bela Hardware



5V

Analog outputs

Analog inputs

USB

Stereo speaker amps

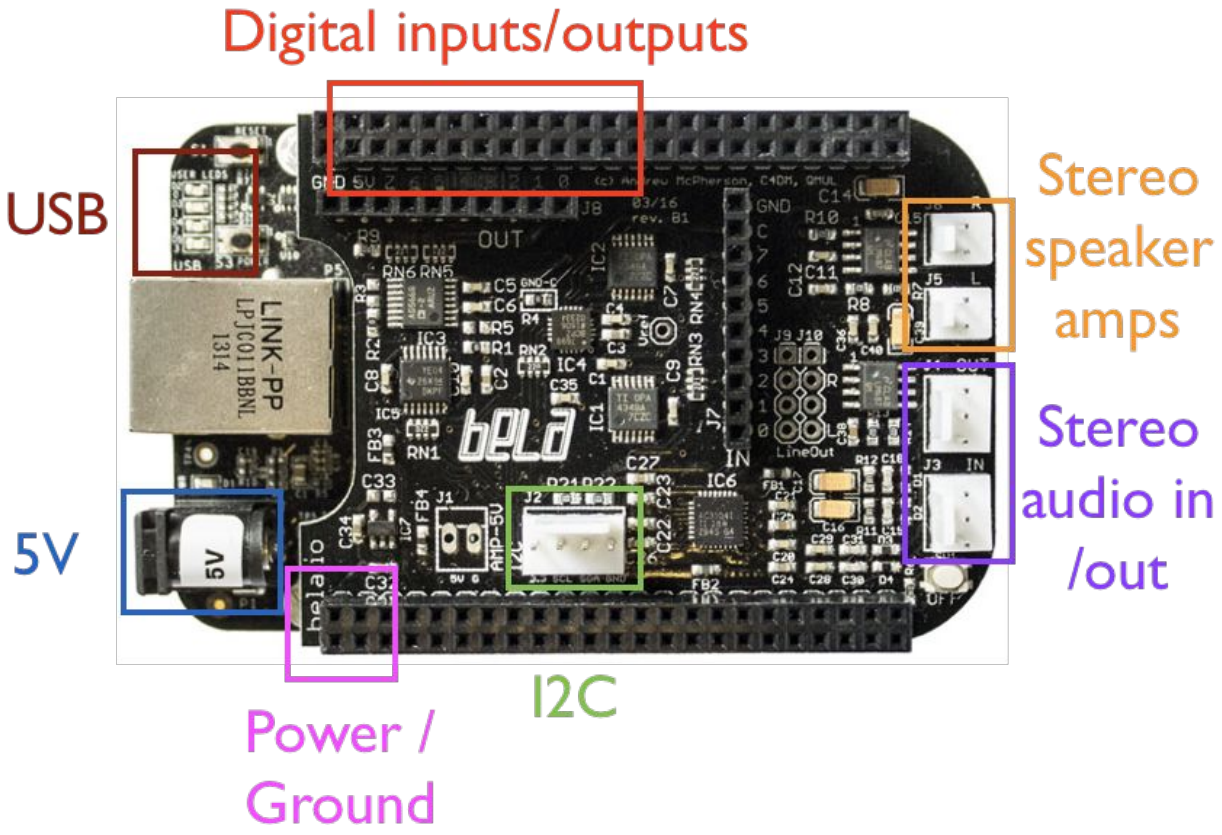
Stereo audio in /out

Power / Ground

Digital inputs/outputs

3.3V

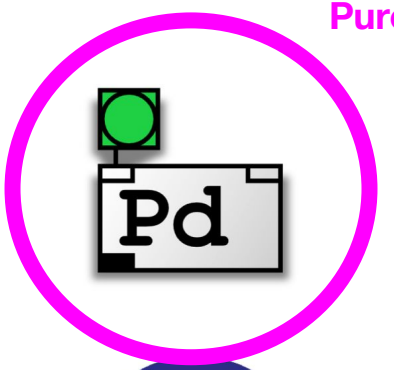
Bela Hardware



Bela is polyglot

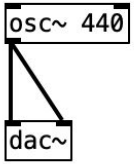
Bela is polyglot

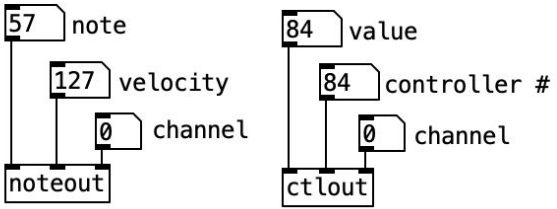
Pure Data

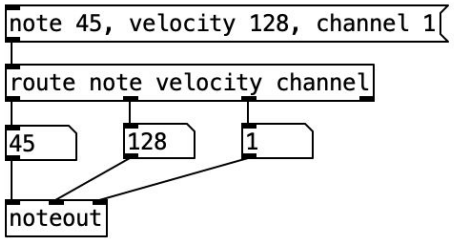


Pure Data

- Free open-source programming language
- Visual environment -> dataflow (‘patching’)
- Powerful, quick -> good for prototyping
- Several flavours: ***Vanilla***, Extended, PurrData
- Objects, messages and signals -> event-driven vs realtime





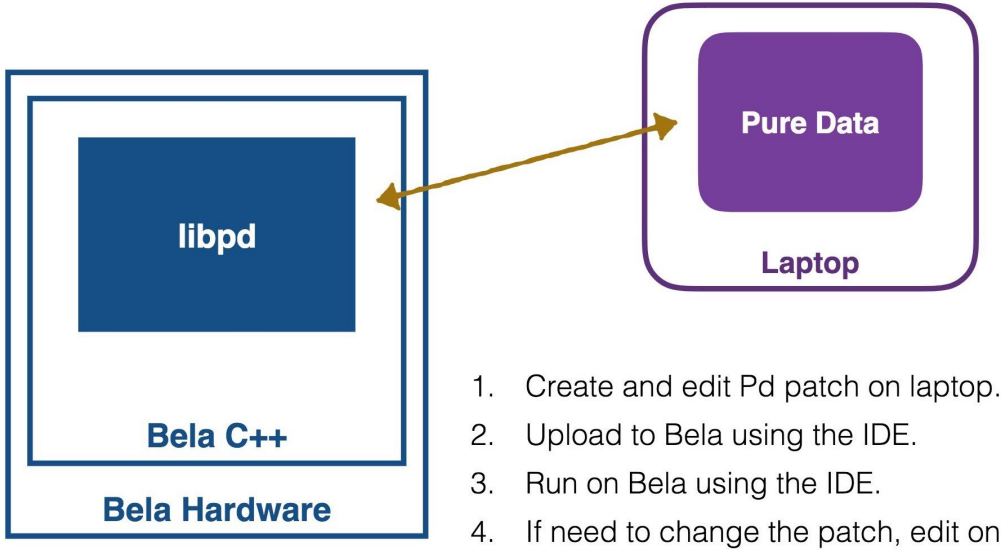


https://puredata.info/downloads

Bela & Pure Data

libpd -> library wrapper -> Pd vanilla DSP functionality -> headless!

Workflow:



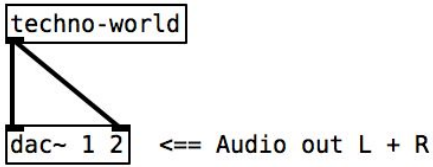
1. Create and edit Pd patch on laptop.
2. Upload to Bela using the IDE.
3. Run on Bela using the IDE.
4. If need to change the patch, edit on laptop and repeat process.

The main Pd patch of your project must always be named `_main.pd`

Bela & Pure Data

Audio I/O [audio rate]

- Stereo audio input: `[adc~ 1 2]`
- Stereo audio output: `[dac~ 1 2]`

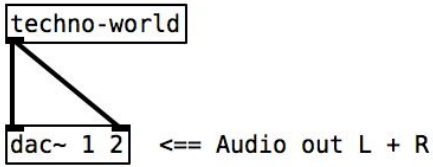


https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

Audio I/O [audio rate]

- Stereo audio input: `[adc~ 1 2]`
- Stereo audio output: `[dac~ 1 2]`



Analog I/O

- Analog inputs: `[adc~ 3 4 5 6 7 8 9 10]`
- Analog outputs: `[dac~ 3 4 5 6 7 8 9 10]`

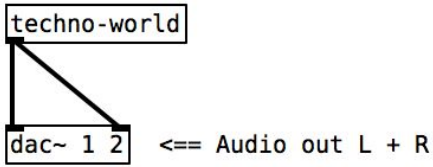
https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

Audio I/O

[audio rate]

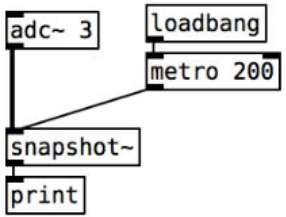
- Stereo audio input: `[adc~ 1 2]`
- Stereo audio output: `[dac~ 1 2]`



Analog I/O

[0.5 * audio rate] -> upsampled internally to audio rate

- Analog inputs: `[adc~ 3 4 5 6 7 8 9 10]`
- Analog outputs: `[dac~ 3 4 5 6 7 8 9 10]`
- Printing at slower rate:



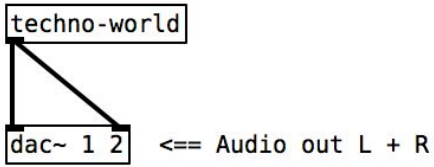
https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

Audio I/O

[audio rate]

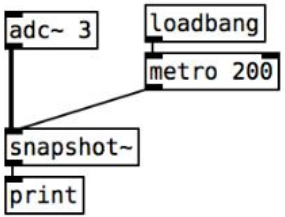
- Stereo audio input: `[adc~ 1 2]`
- Stereo audio output: `[dac~ 1 2]`



Analog I/O

[0.5 * audio rate] -> upsampled internally to audio rate

- Analog inputs: `[adc~ 3 4 5 6 7 8 9 10]`
- Analog outputs: `[dac~ 3 4 5 6 7 8 9 10]`
- Printing at slower rate:



Scope

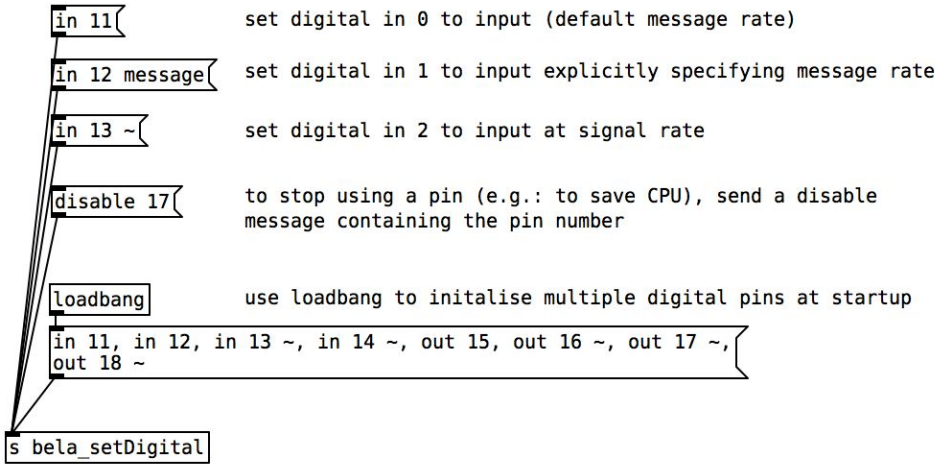
- 4 channels: `[dac~ 27 28 29 30]` → signals must be audio rate `sig~`

https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

Digital I/O [audio rate]

- Digital pins *0-16* are numbered ***11-26***
- initialised by sending messages to `bela_setDigital`

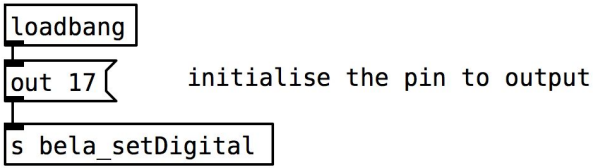


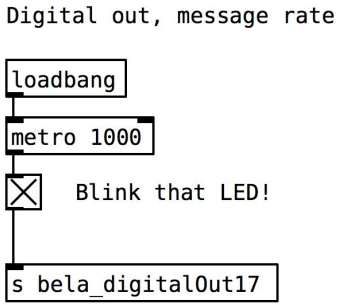
https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

Digital I/O [audio rate]

- ***Digital OUT***



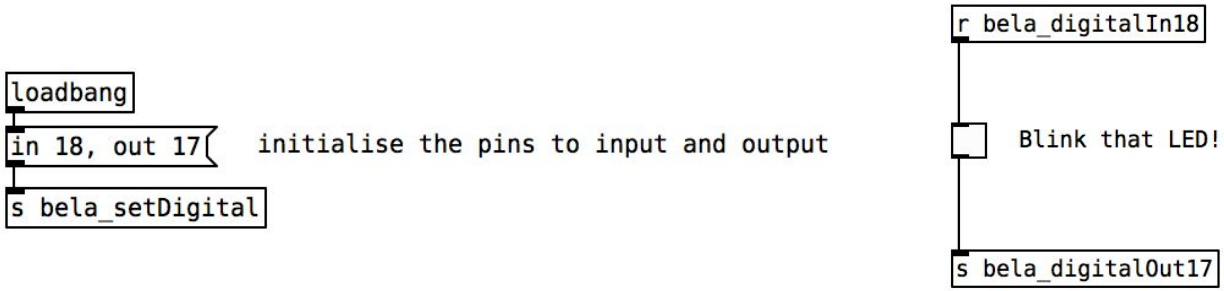


https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

Digital I/O [audio rate]

- ***Digital IN***



https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

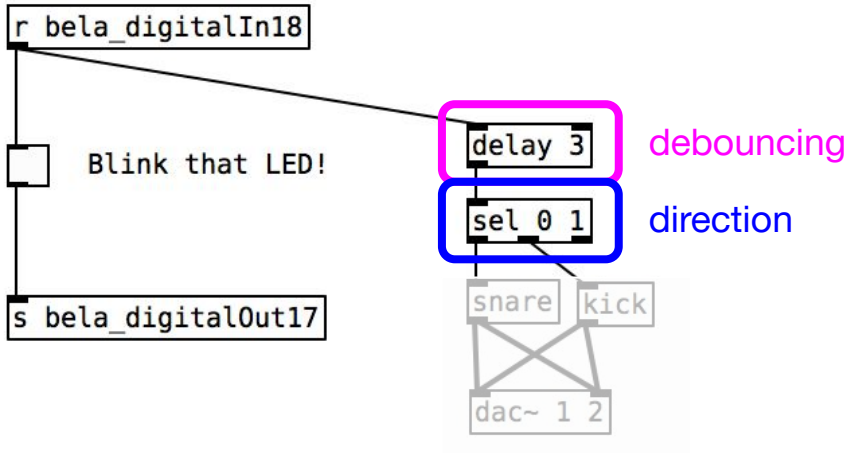
Custom render

- Allows us to write our own C++ code to interact with libpd
- Complex, big file, only part of it can be edited

https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

Digital input debouncing

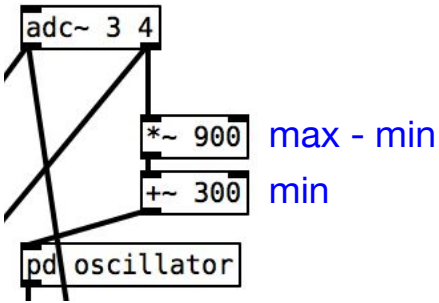


https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

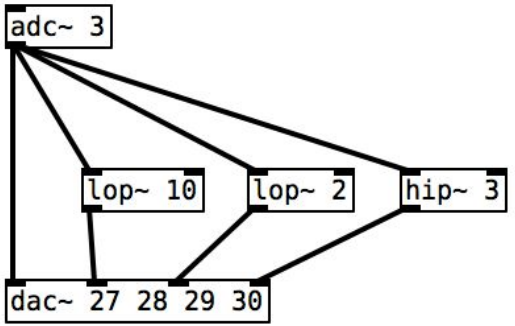
Bela & Pure Data

Analog input conditioning

- **Scaling to specific range**



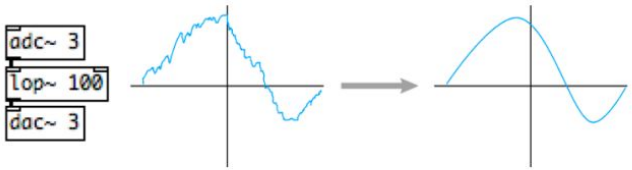
- **Filtering**



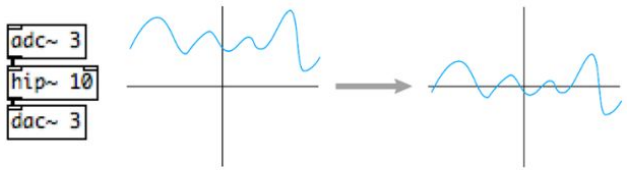
https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Bela & Pure Data

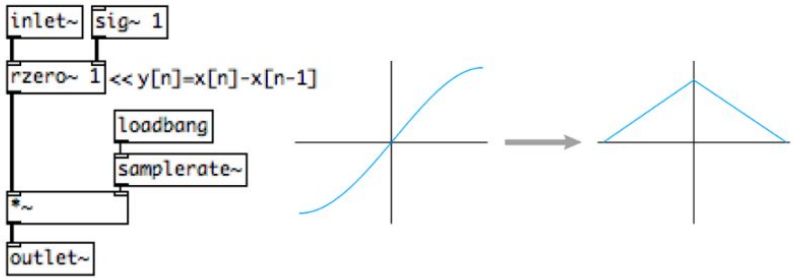
Analog input conditioning



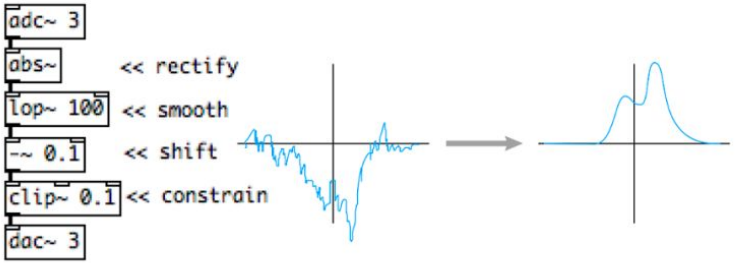
(a) Smoothing



(b) Re-centering



(c) Differentiating



(d) Thresholding

https://learn.bela.io/using-bela/languages/pure-data/#about-pure-data

Getting up & running

Required software

Firefox/Chrome browser (*recommended)

The IDE may or may not work in other browsers.

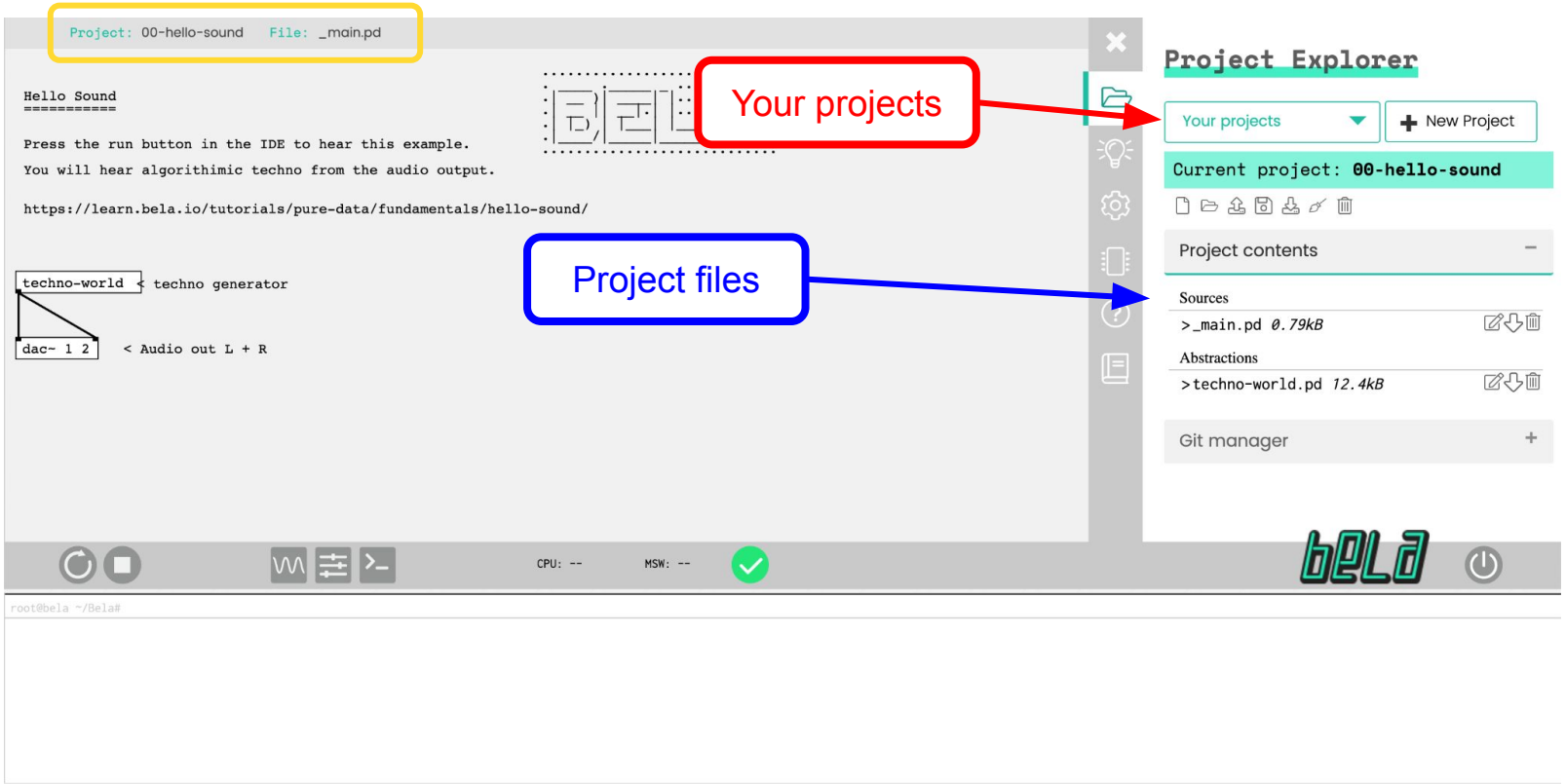
Getting up & running

1. Plug in Bela to computer (USB)
2. Launch the browser
3. After around 30s go to the url:

MacOS, Linux	http://bela.local/
Windows 10 (and older)	http://192.168.6.2
Windows 11	http://192.168.7.2

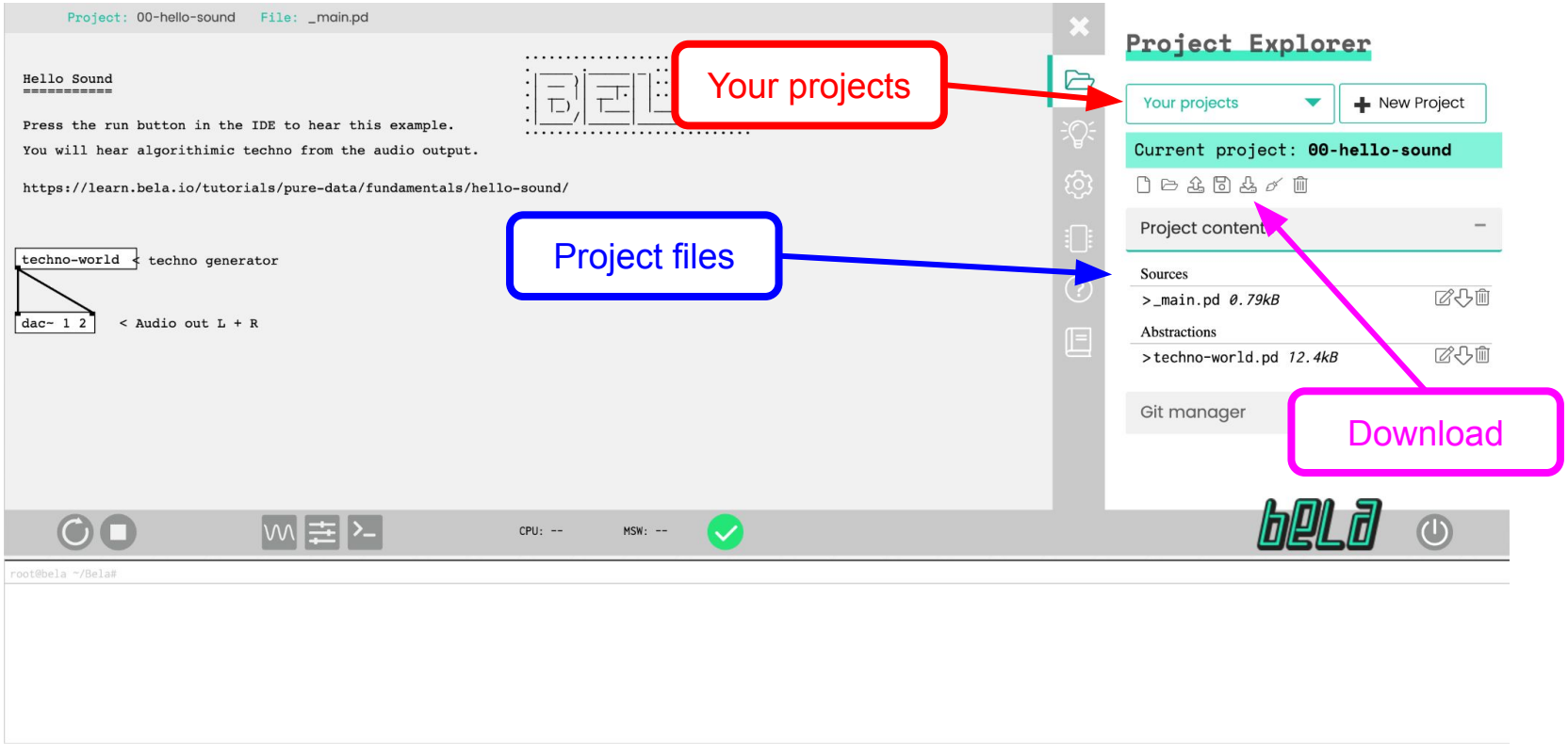
Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2



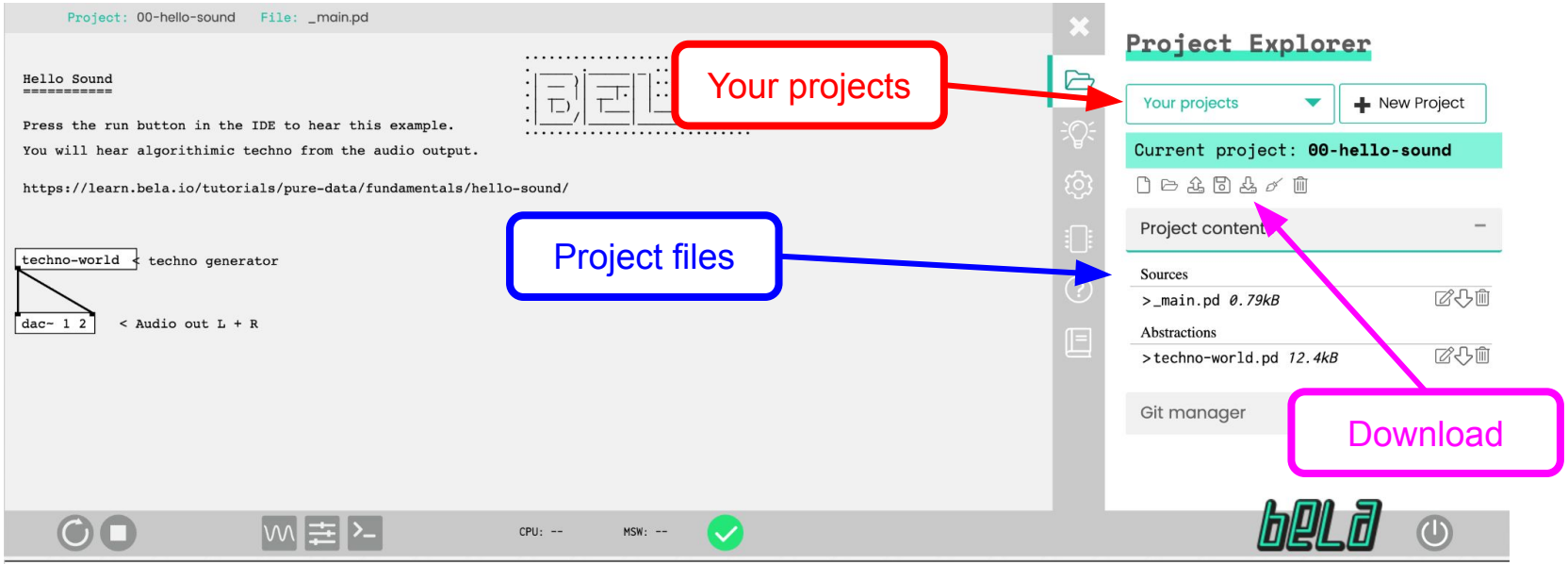
Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2



Bela IDE

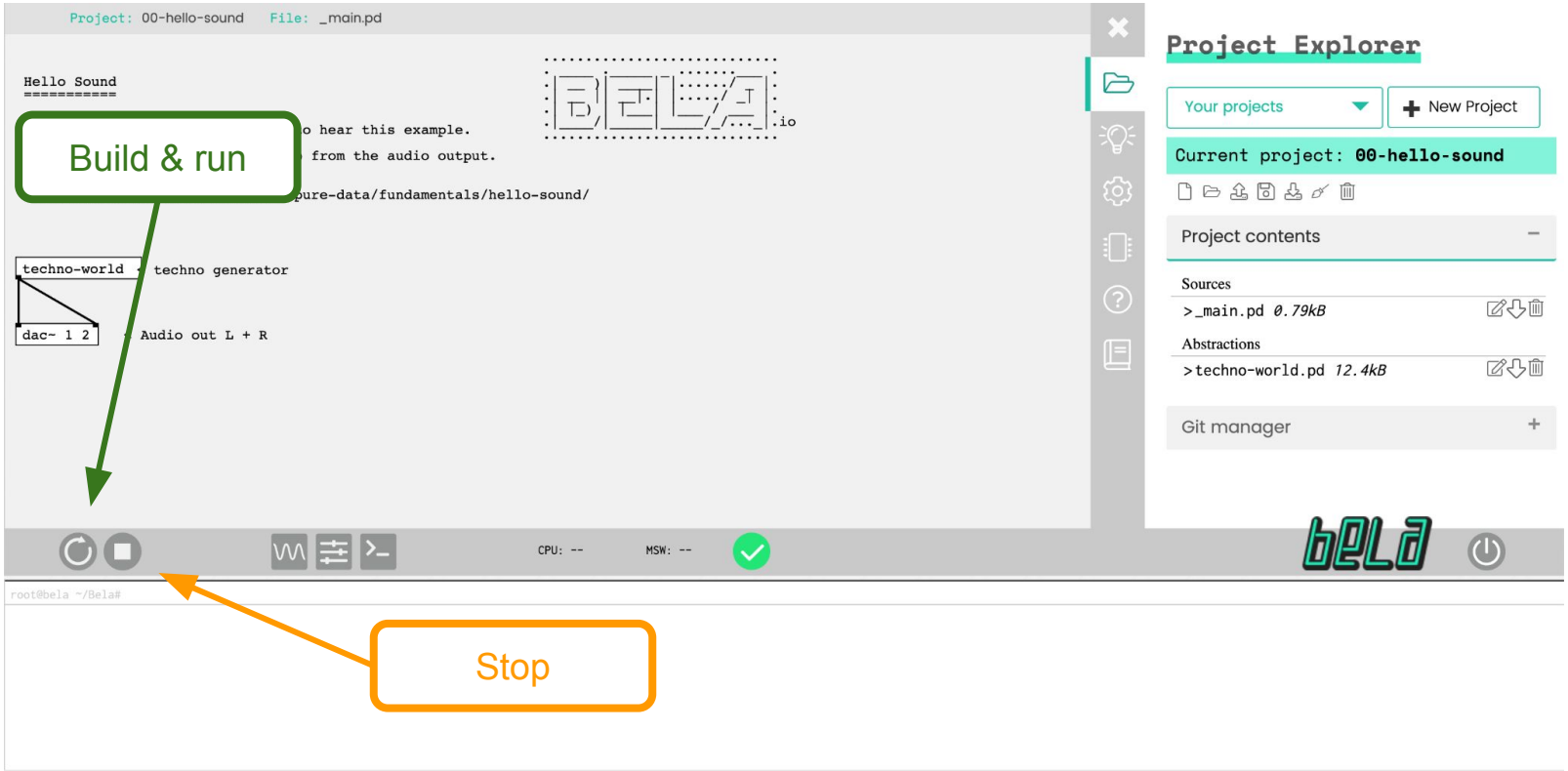
bela.local or 192.168.7.2 or 192.168.6.2



Patch connections can be visualised in the browser,
but cannot be edited here

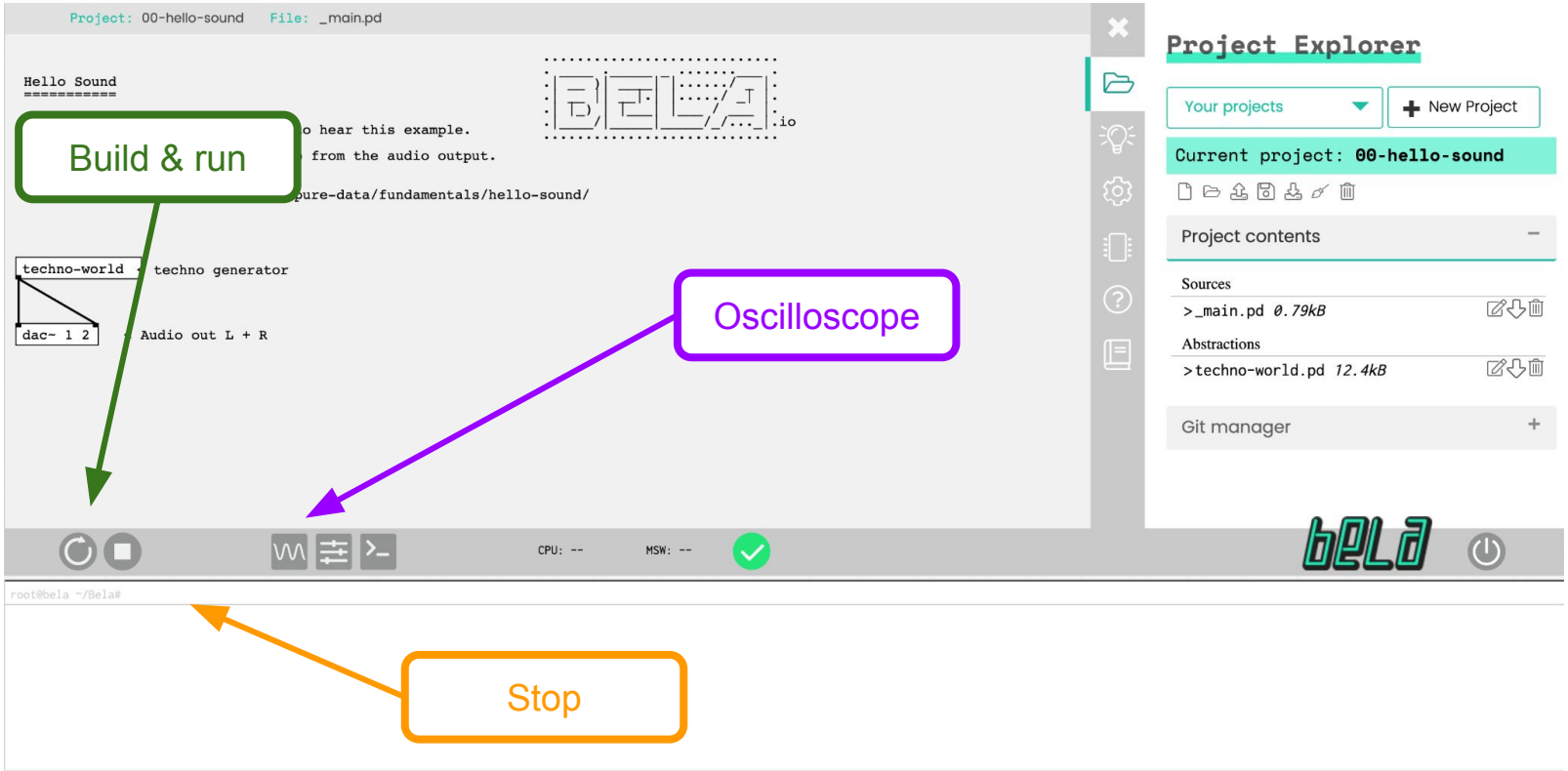
Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2



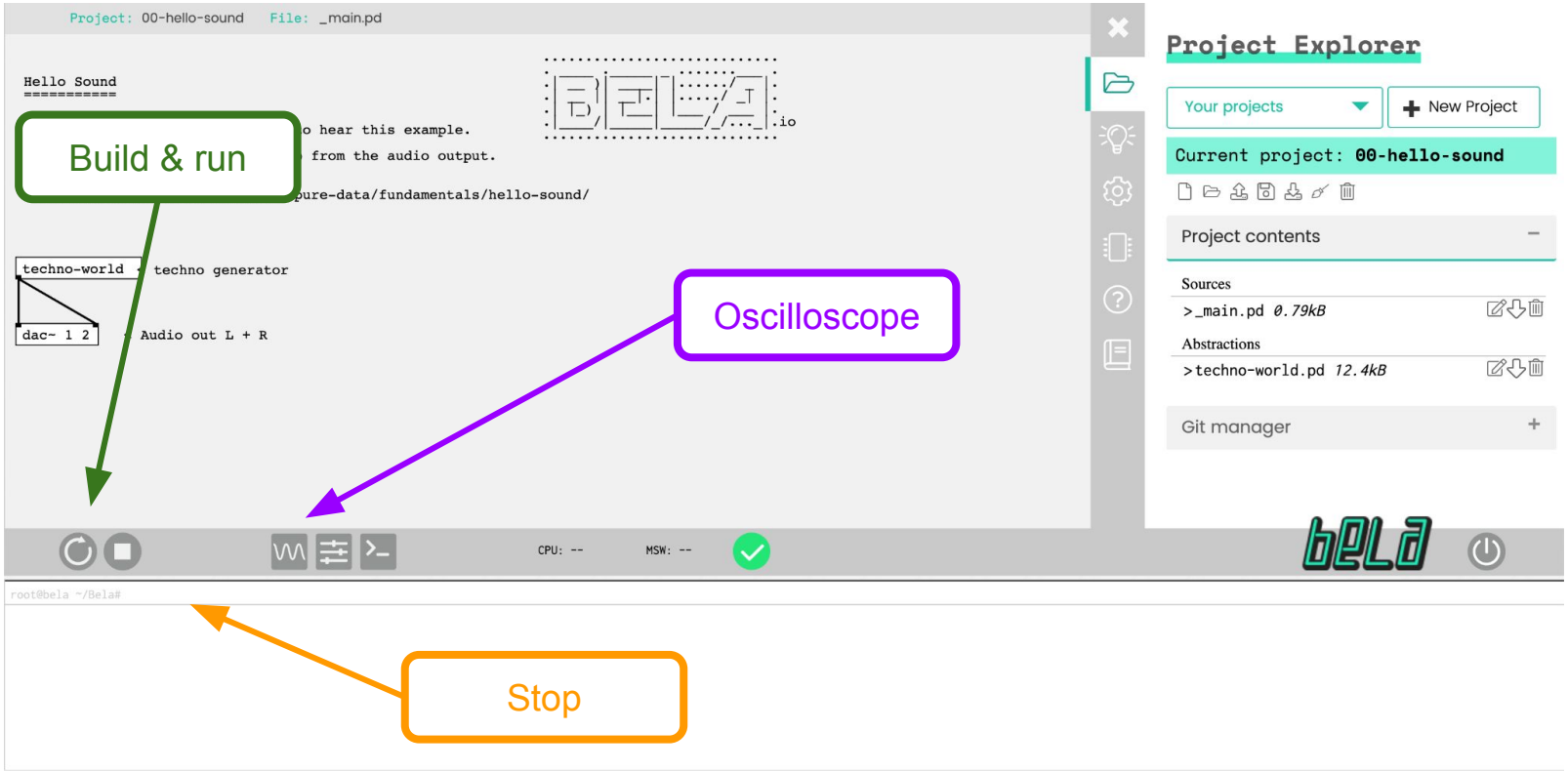
Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2



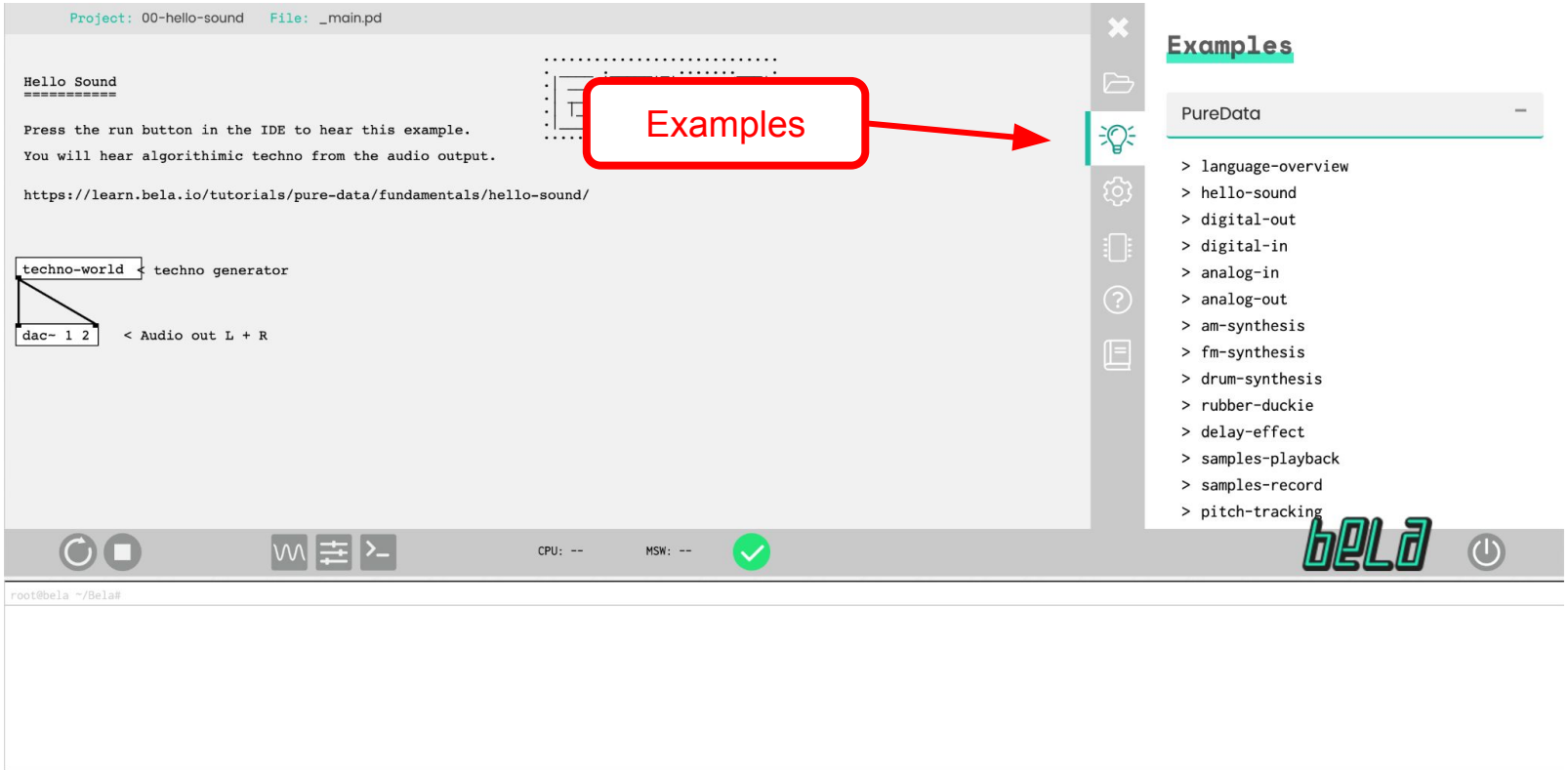
Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2



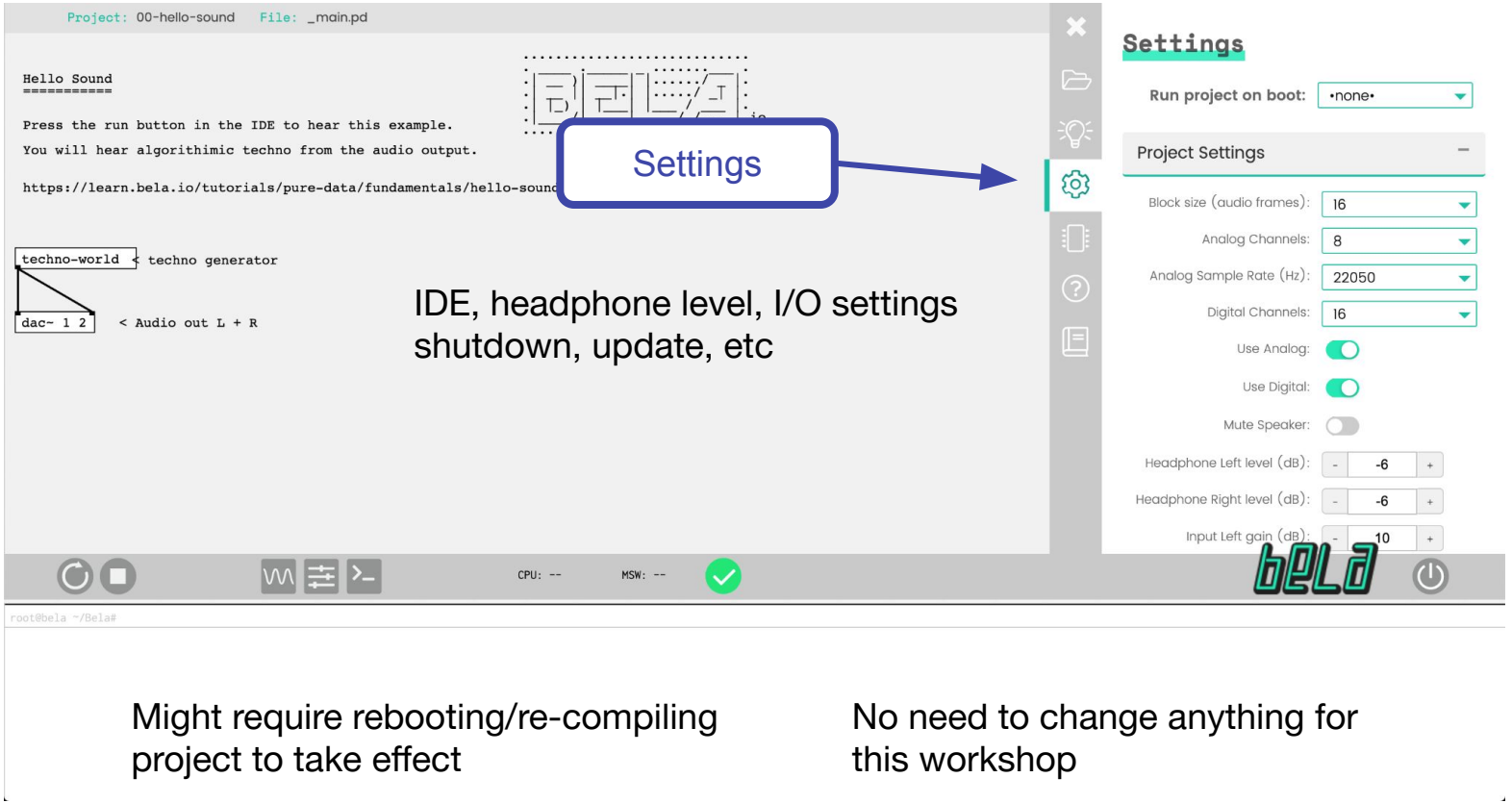
Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2



Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2

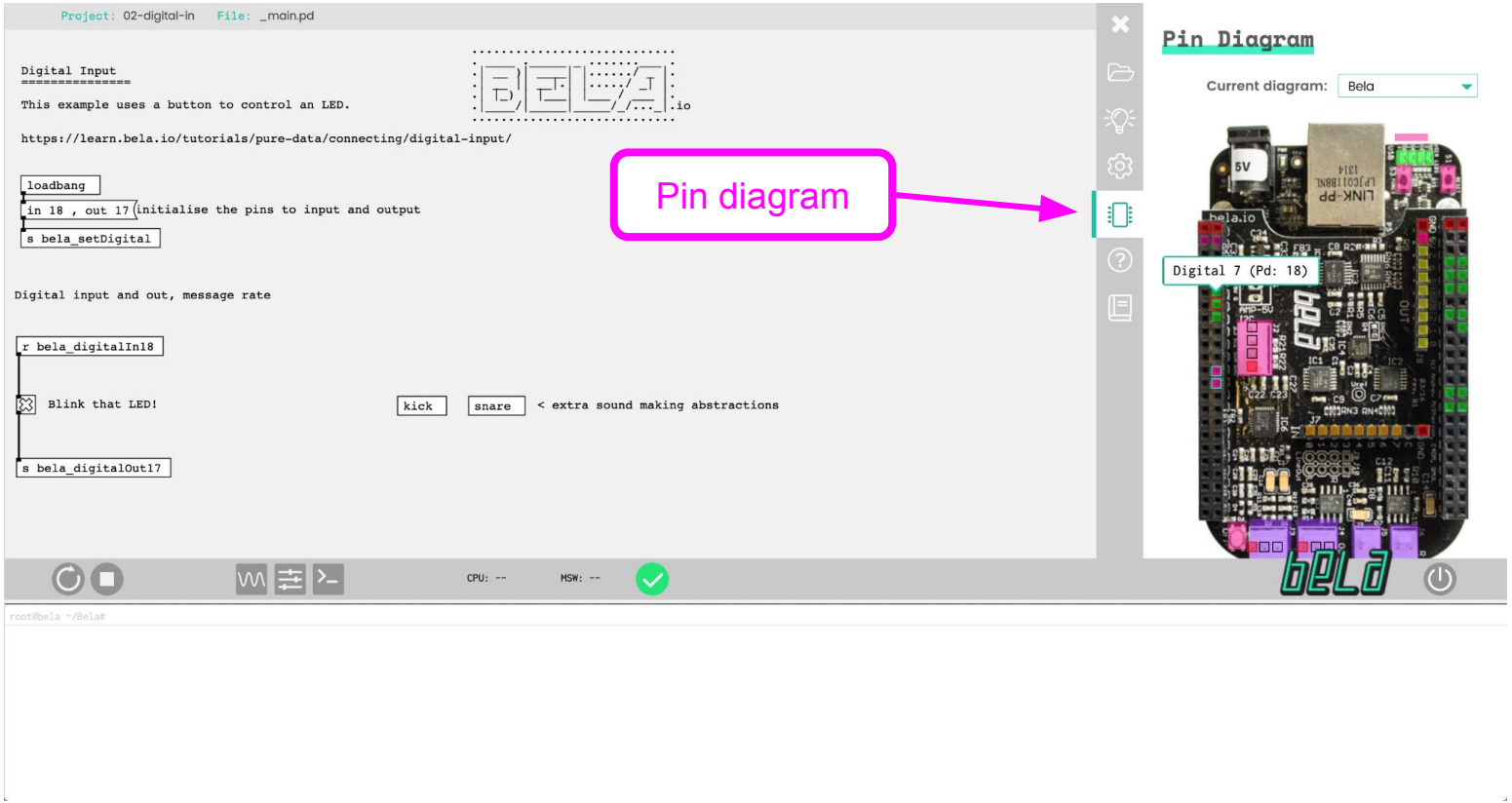


Might require rebooting/re-compiling project to take effect

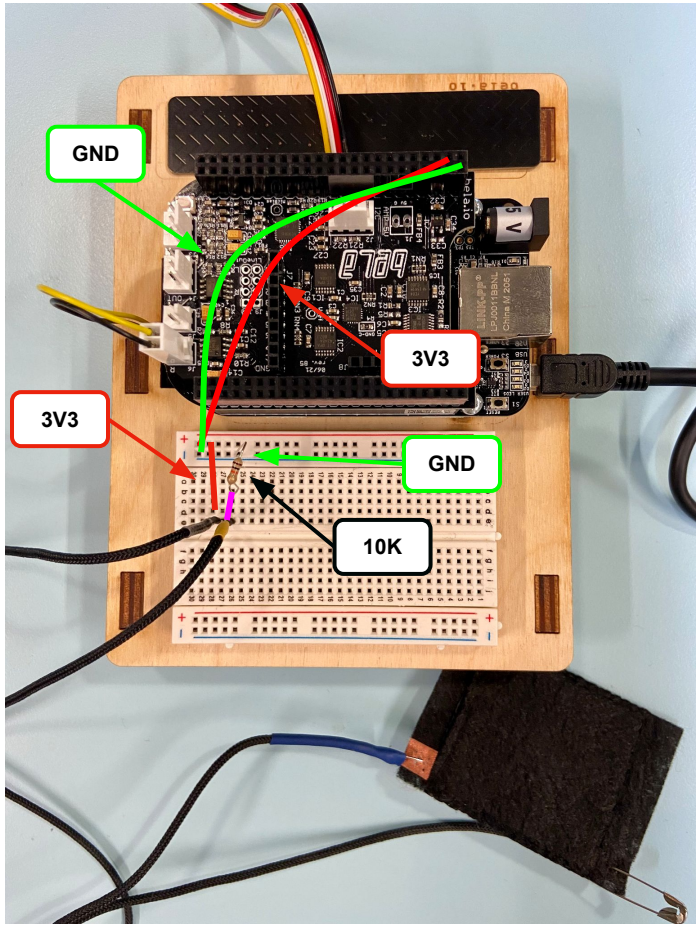
No need to change anything for this workshop

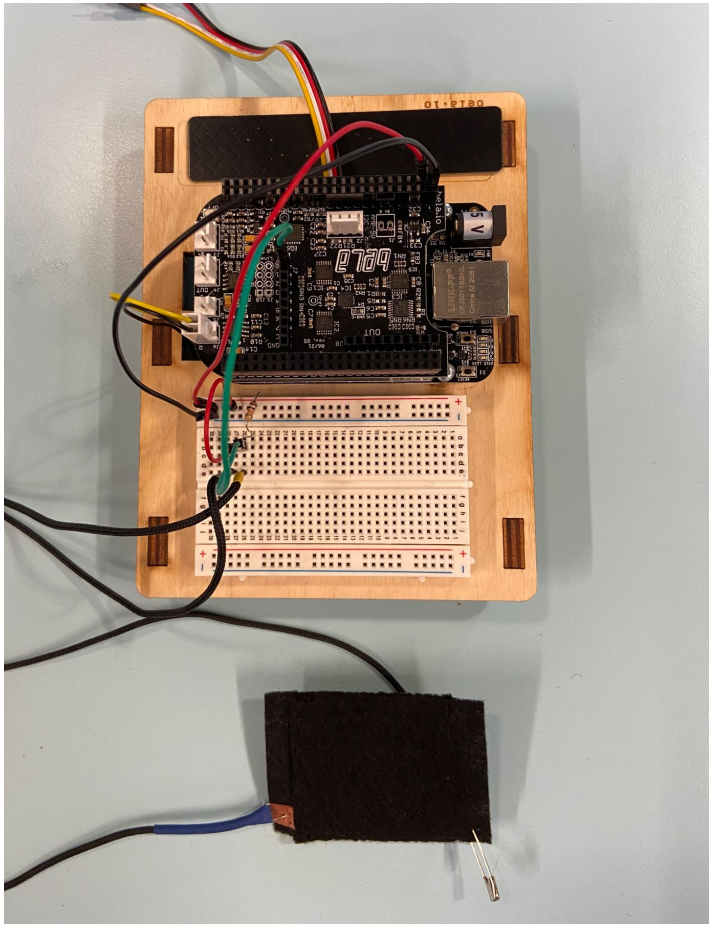
Bela IDE

bela.local or 192.168.7.2 or 192.168.6.2



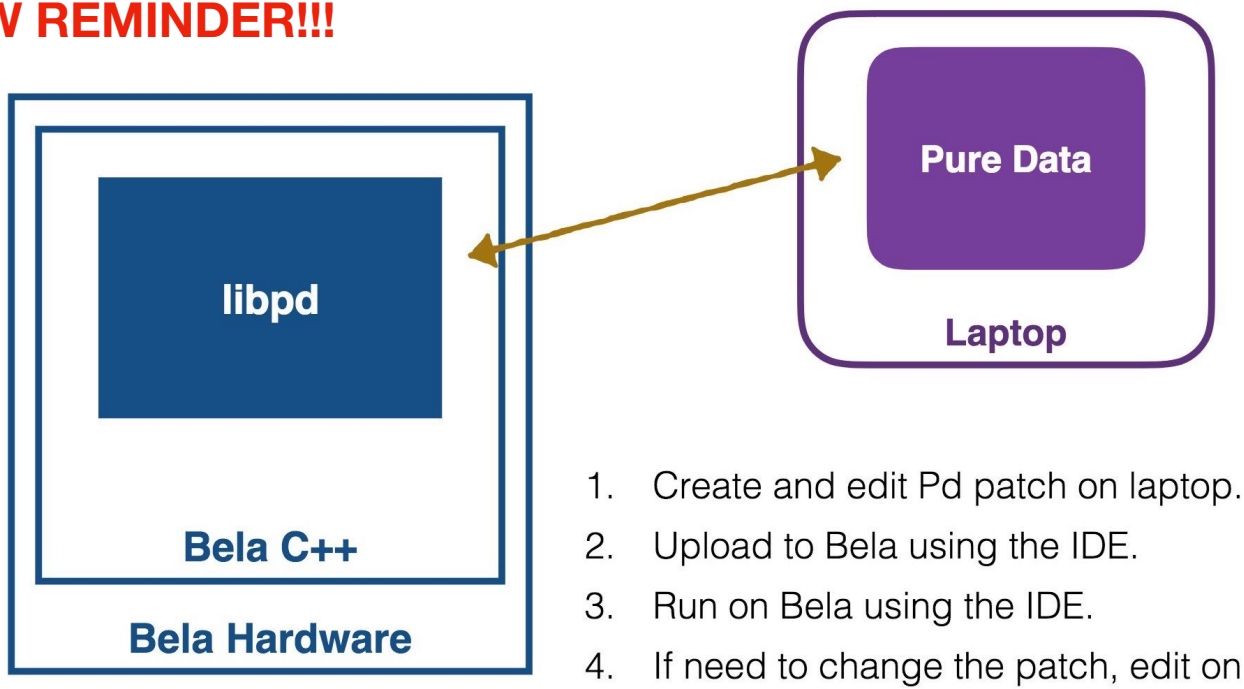
Connect your sensor to Bela!





Bela & Pure Data

WORKFLOW REMINDER!!!

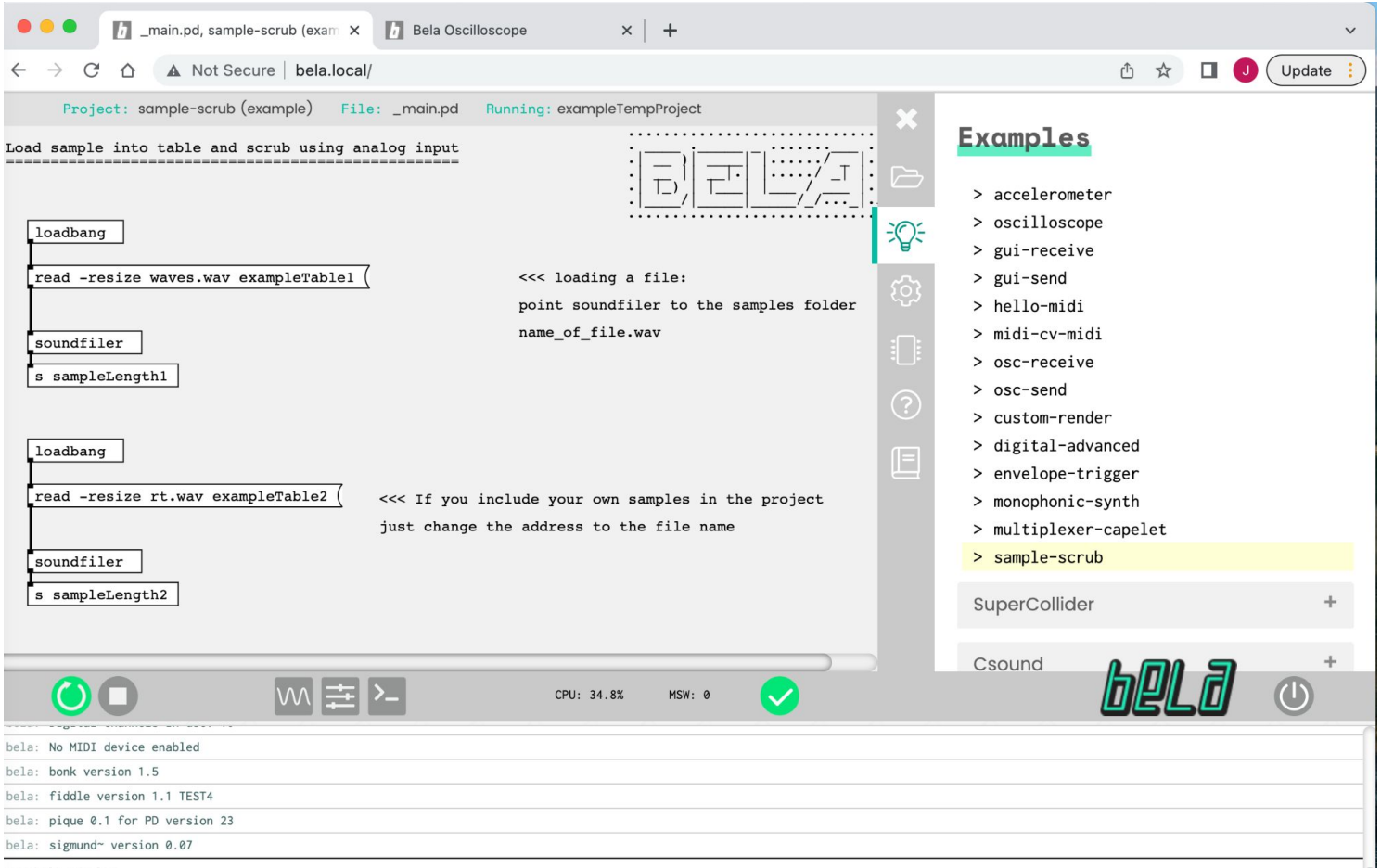


1. Create and edit Pd patch on laptop.
2. Upload to Bela using the IDE.
3. Run on Bela using the IDE.
4. If need to change the patch, edit on laptop and repeat process. (*)

The main Pd patch of your project must always be named `_main.pd`

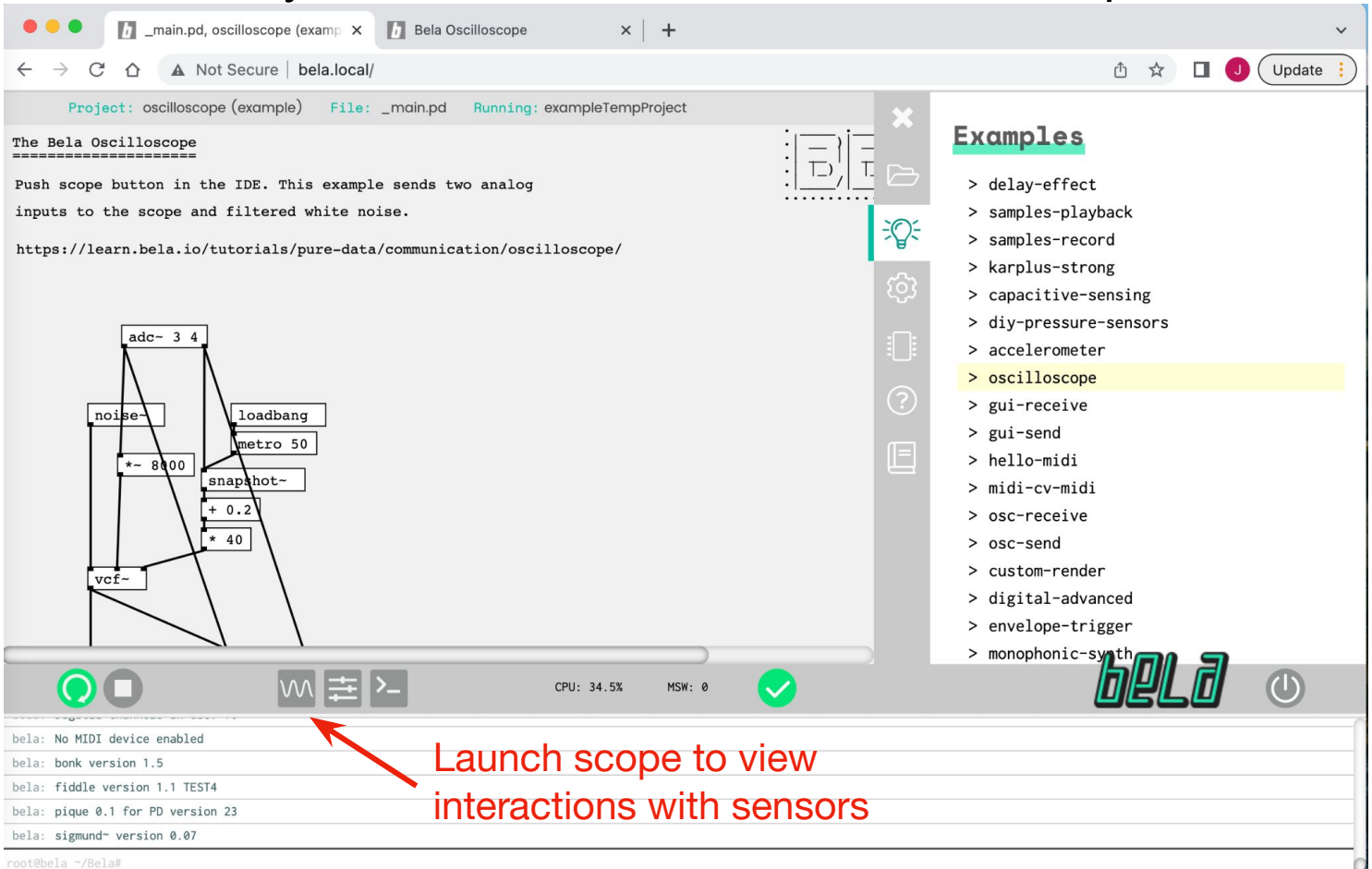
Open example Examples/PureData/sample-scrub

In IDE, you can view and run but not edit Pd patches



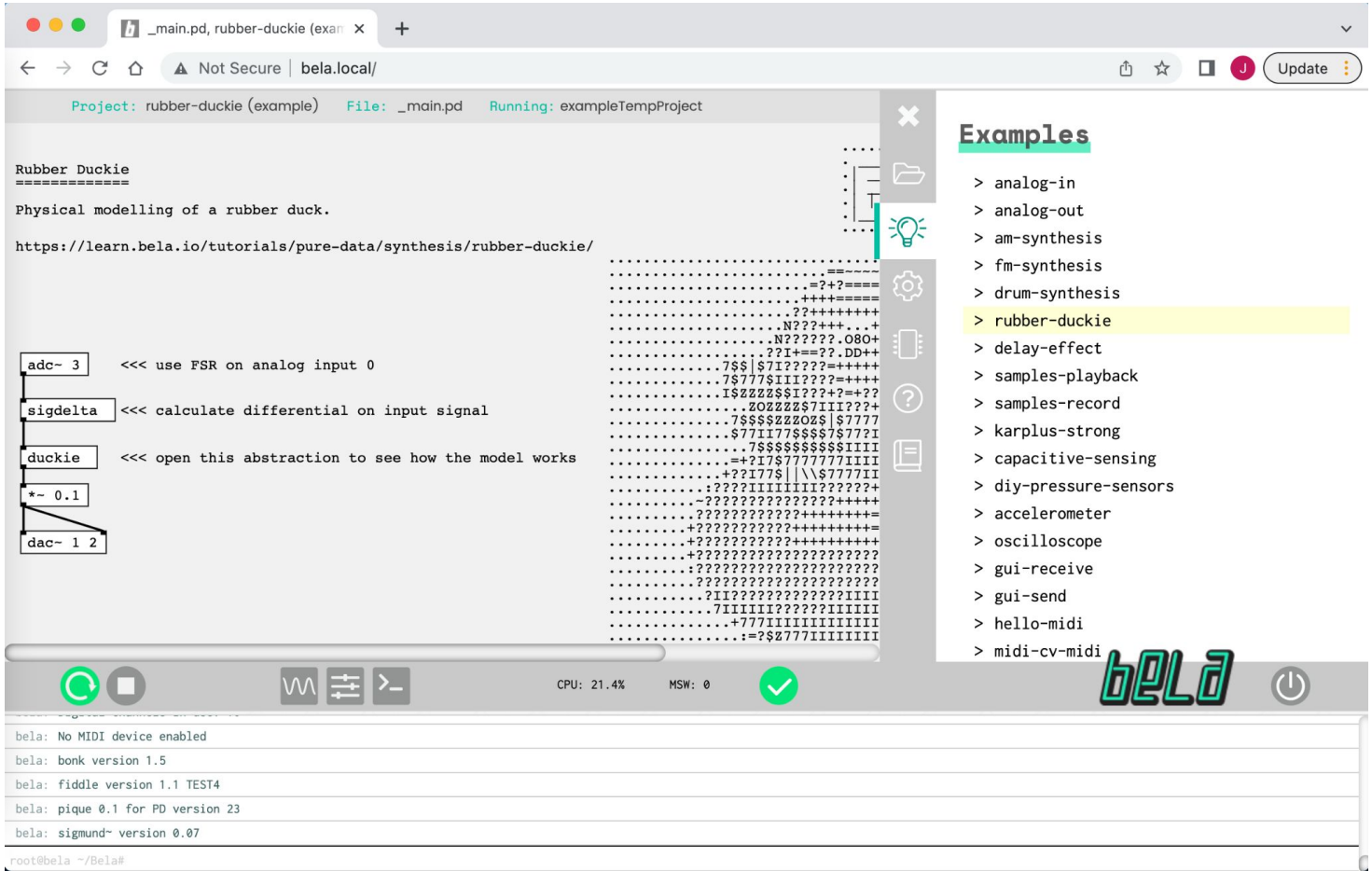
Open example Examples/PureData/Oscilloscope

In IDE, you can view and run but not edit Pd patches



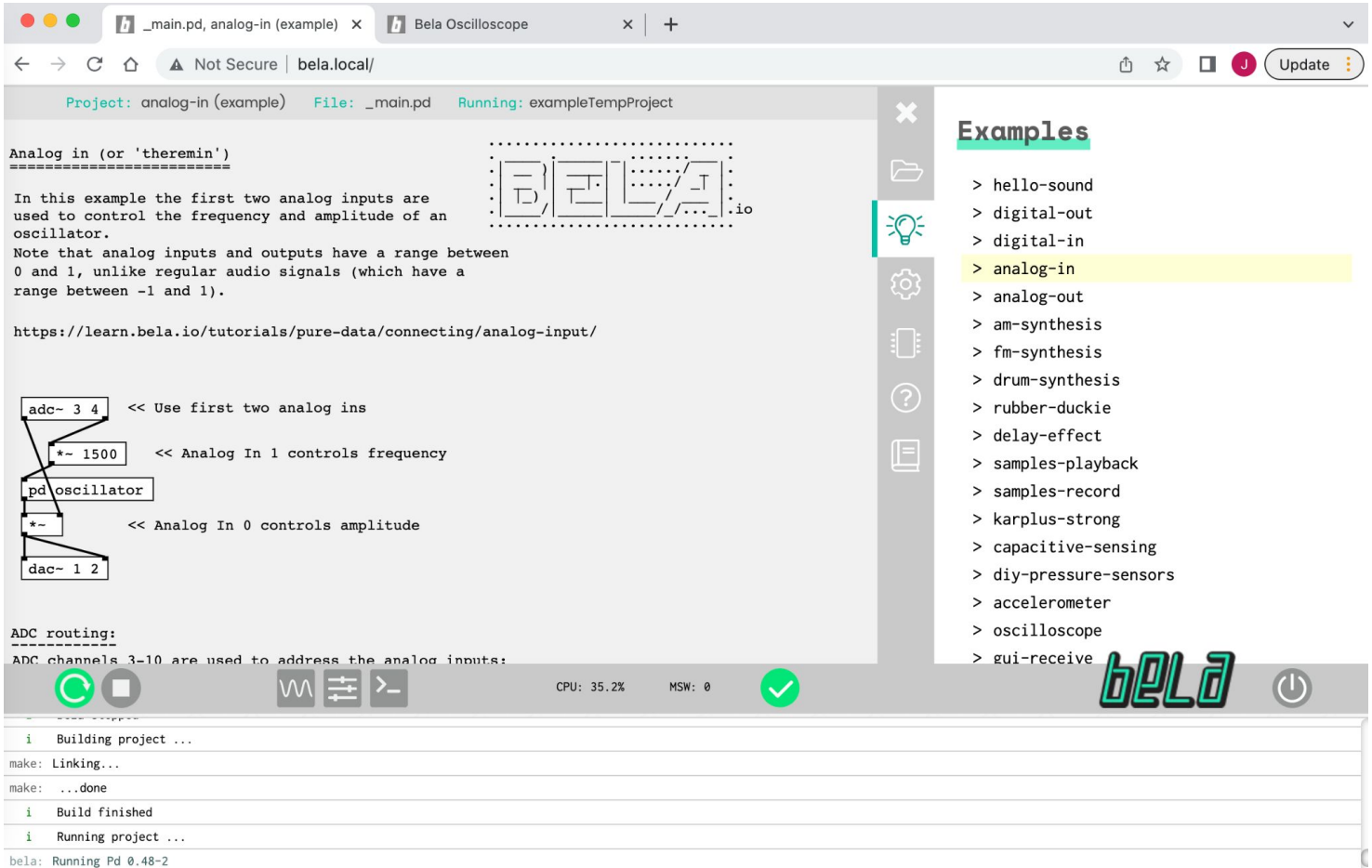
Open example Examples/PureData/rubber-duckie

In IDE, you can view and run but not edit Pd patches



Open example Examples/PureData/analog-in

In IDE, you can view and run but not edit Pd patches



Open example Examples/PureData/diy-pressure-sensors

In IDE, you can view and run but not edit Pd patches