

C++ Real-Time Audio Programming with Bela

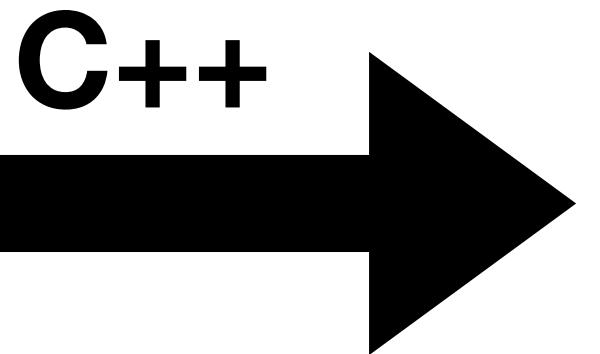
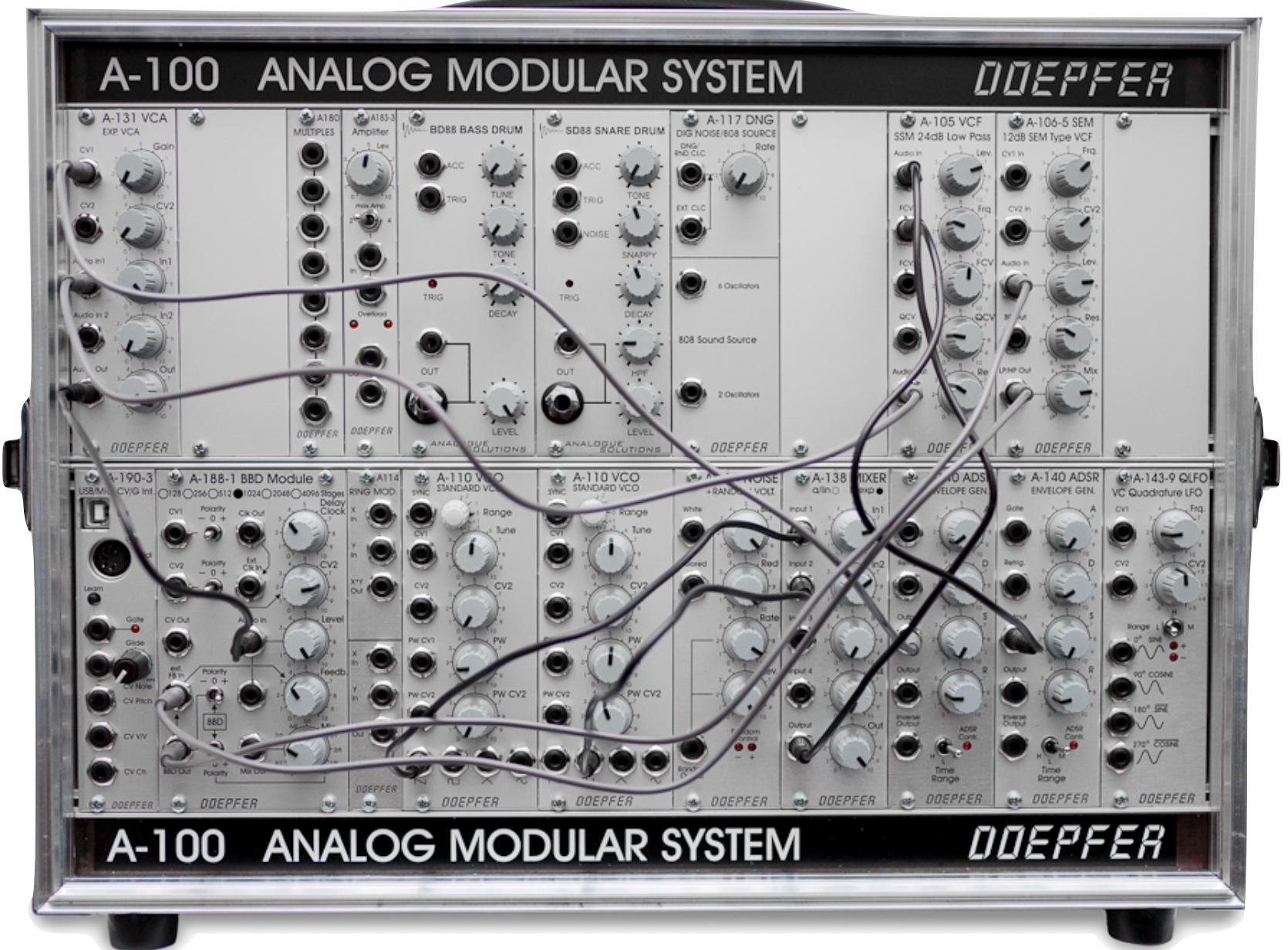
Dr Andrew McPherson

Centre for Digital Music
School of Electronic Engineering and Computer Science
Queen Mary University of London

Founder and Director, Bela

C++ Real-Time Audio Programming with Bela

Modular synthesis



Embedded hardware

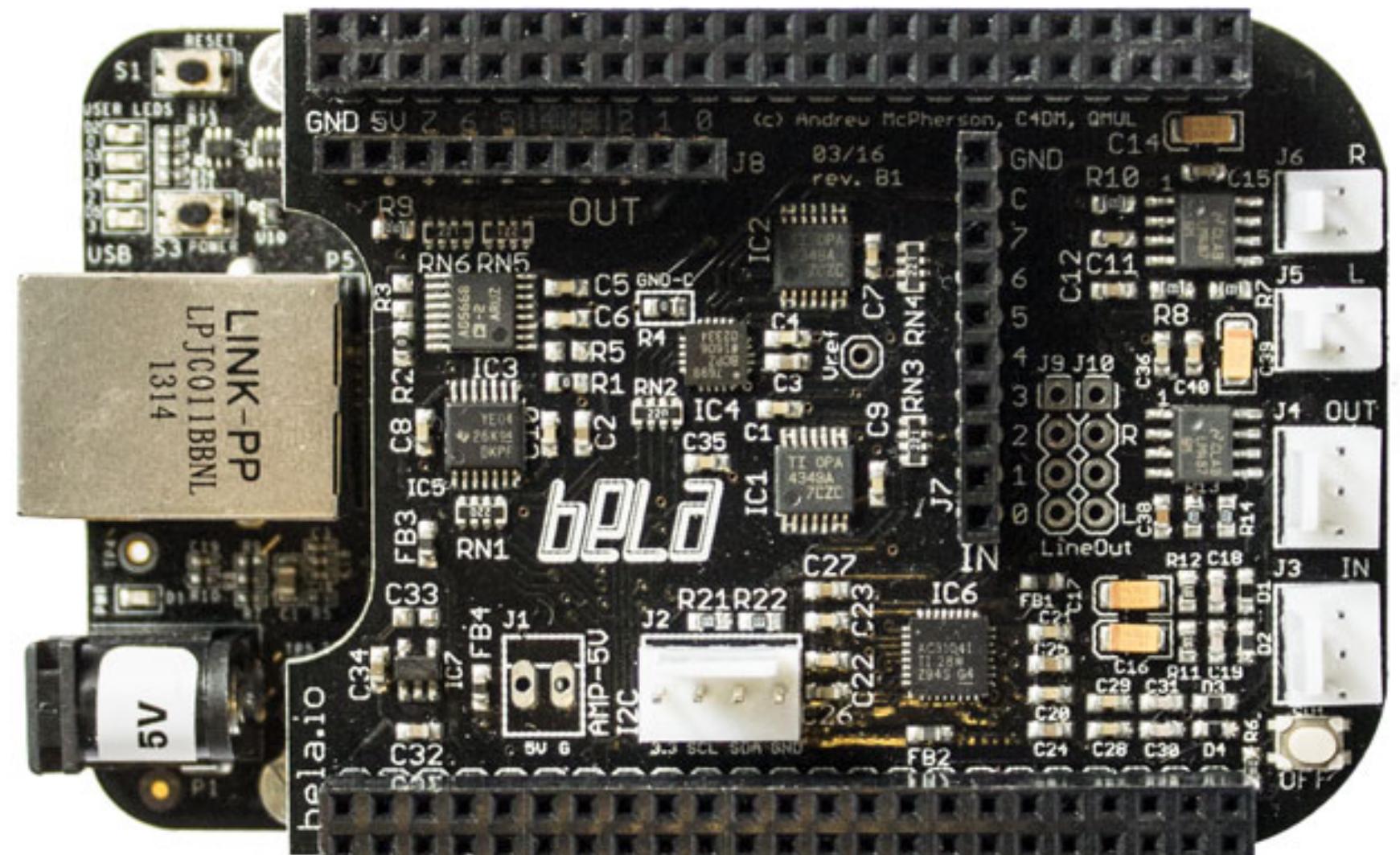


Image credit: Nina Richards, wikipedia (CC-BY 3.0)

What you'll learn

During this course, you will learn about:

- Writing **real-time audio code** in C++
- Running that code on **embedded hardware**
- Creating **musical instruments**, **audio effects** and **synth modules**
- Turning digital signal processing **theory** into working code

We assume passing familiarity with these concepts:

Signals & Systems

- Continuous & discrete time
- Sampling
- Time & frequency domain
- Aliasing

DSP

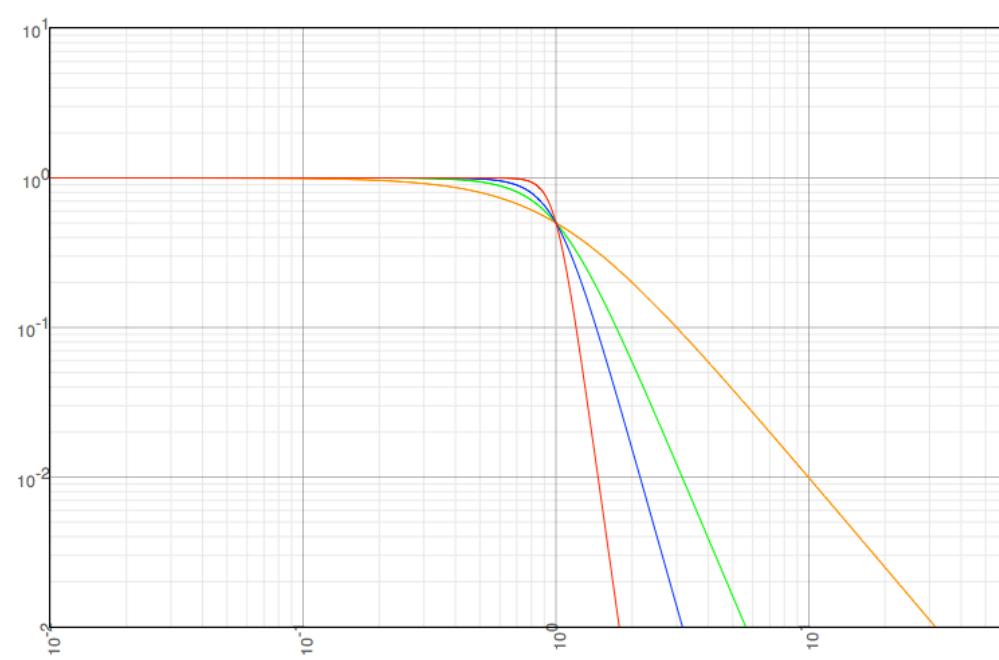
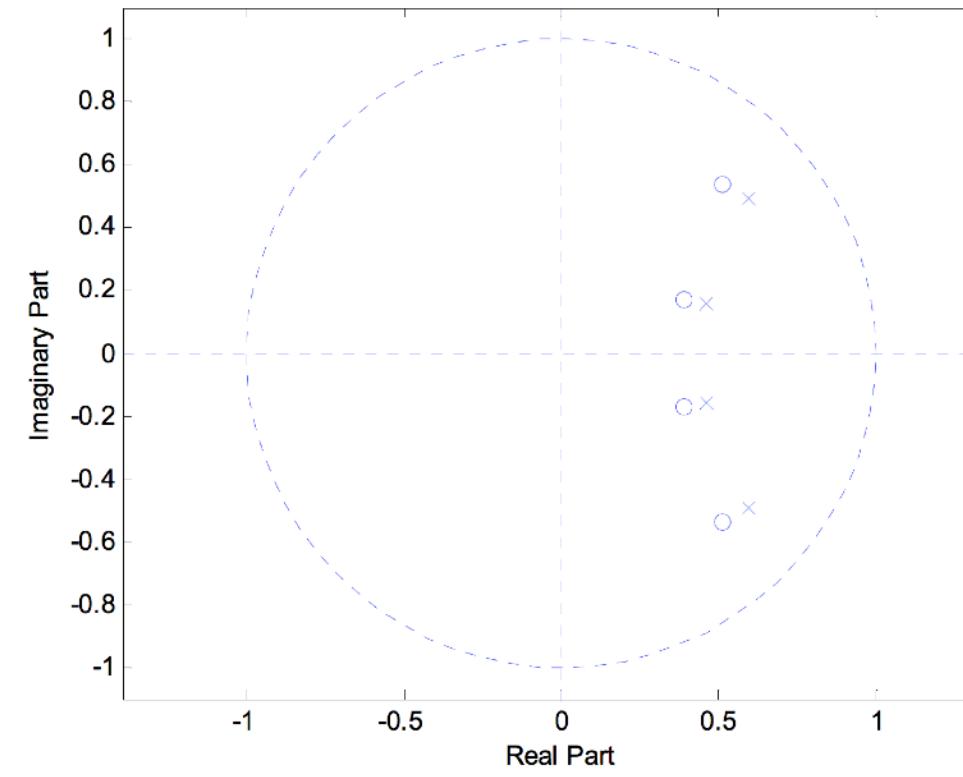
- FIR and IIR filters
- Convolution
- Fourier transform (FFT)

Programming

- Basic familiarity with programming (C, C++, Python, Java, JS etc.) is helpful but optional

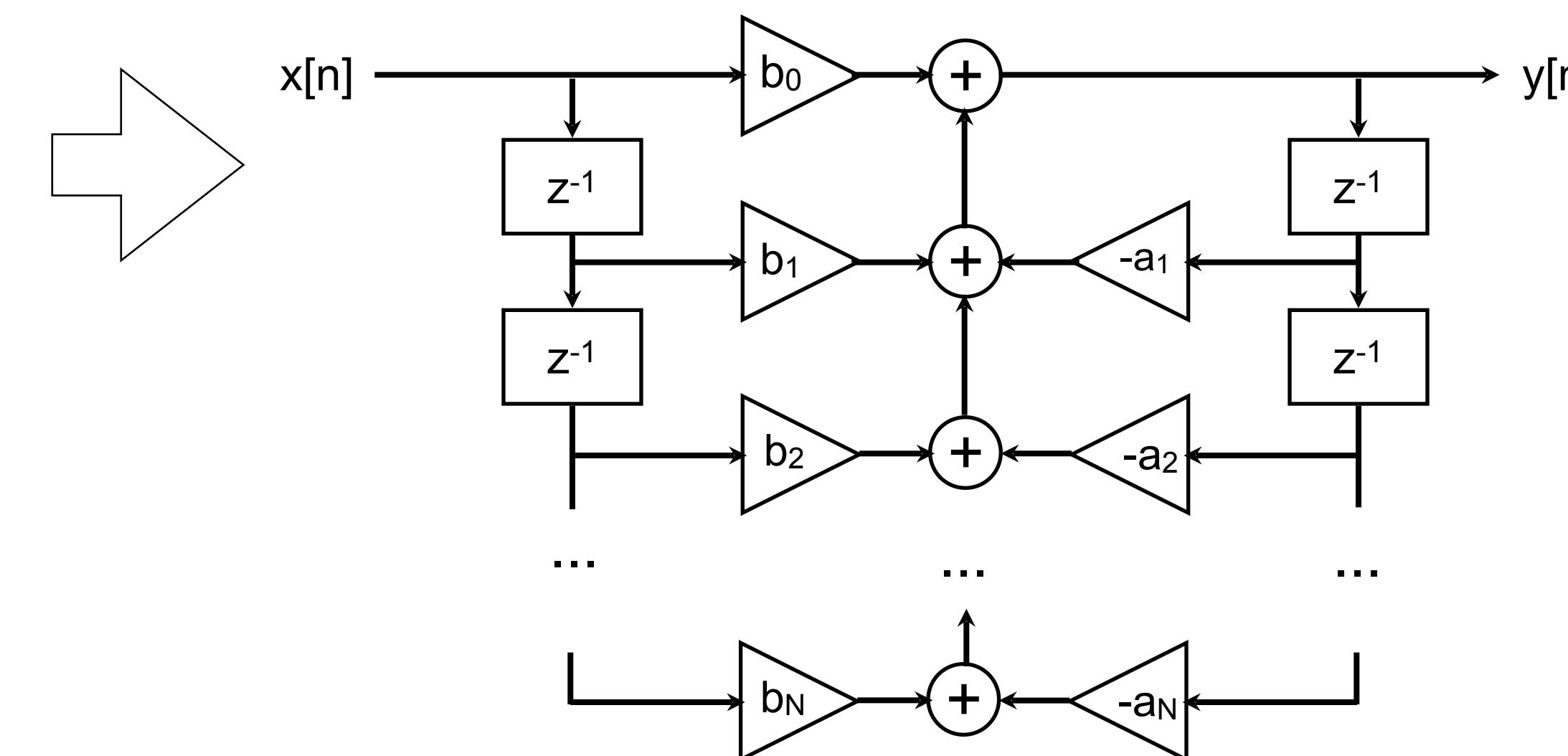
The course in a nutshell

Specifications



Design

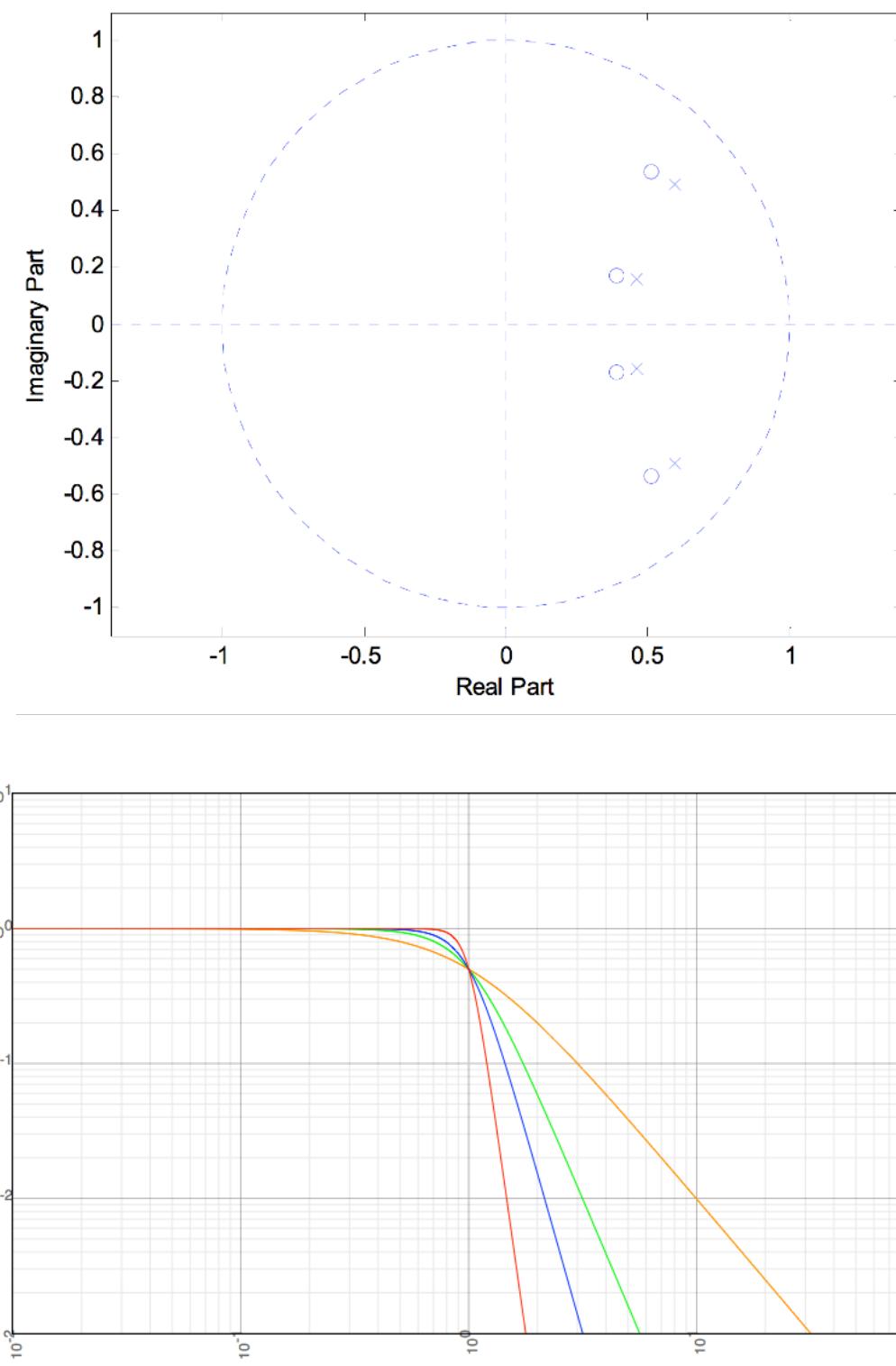
$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k]$$



Typical digital signal processing course

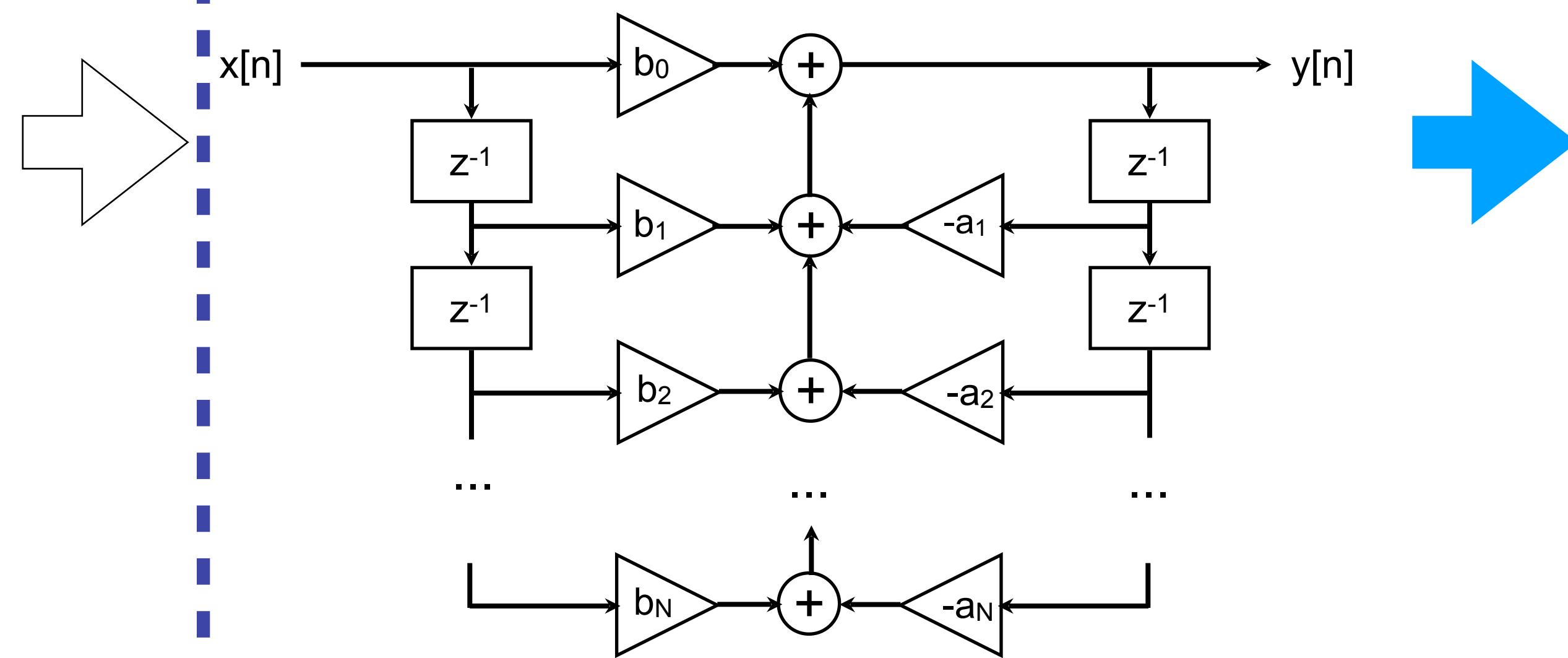
The course in a nutshell

Specifications



Design

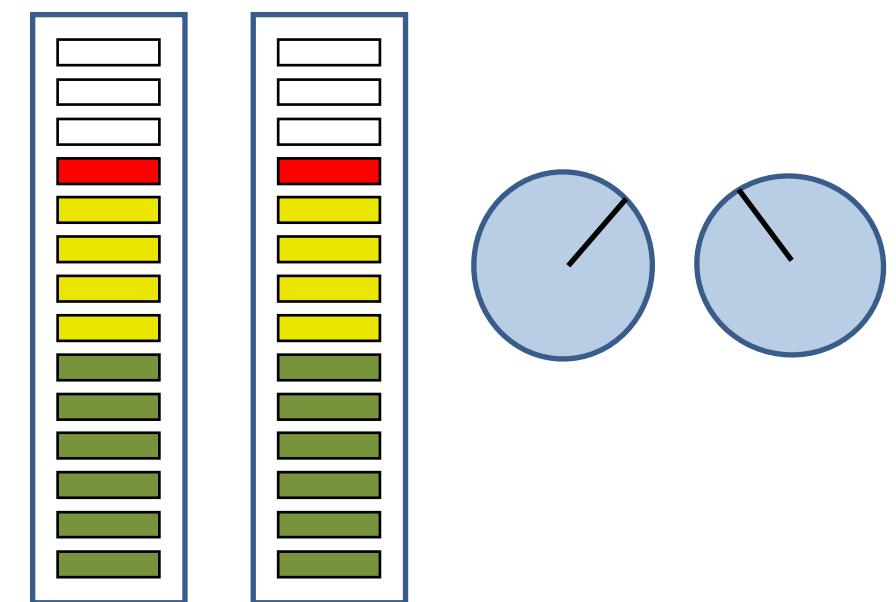
$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k]$$



This course

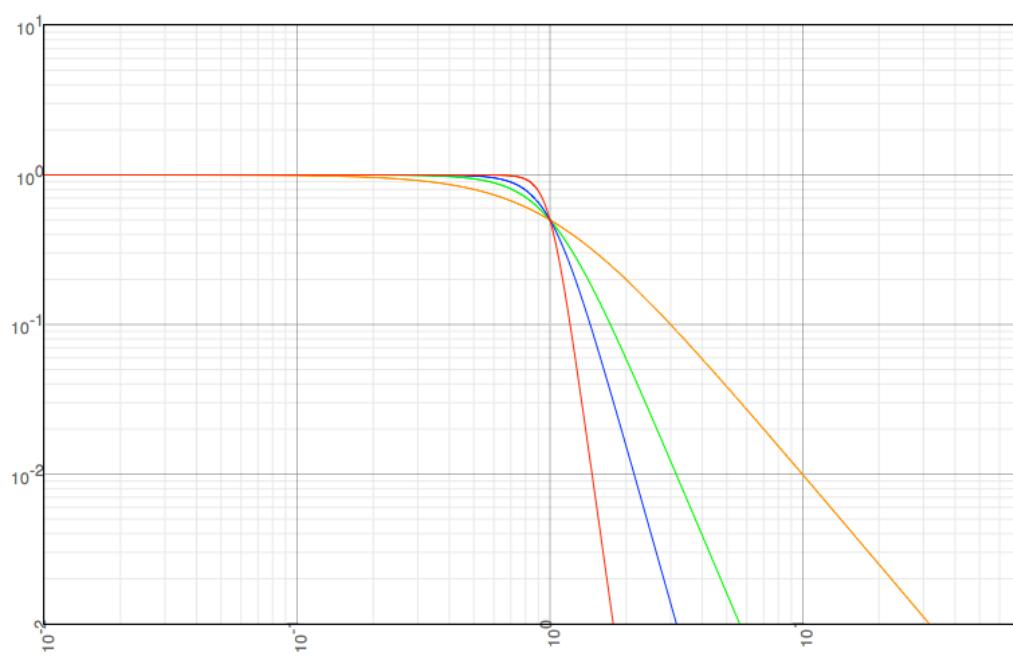
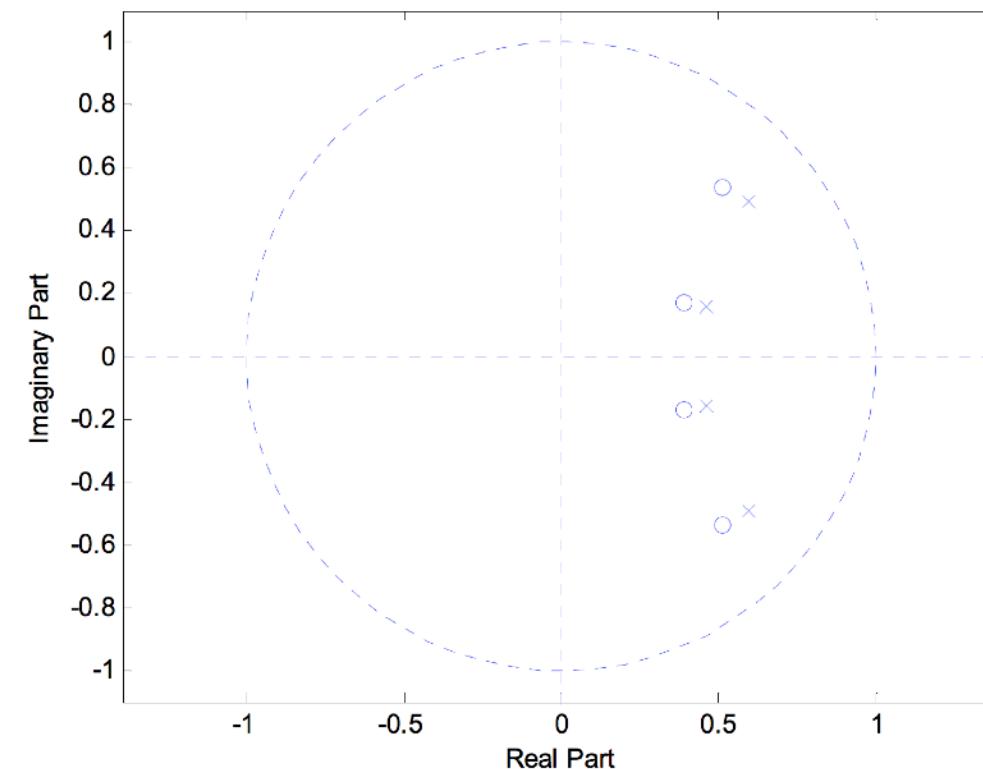
Implementation

```
void render(BelaContext *context, void * /* user data */)
{
    /* Iterate over each channel */
    for(int channel = 0; channel < numChannels; channel++)
        /* Then iterate over each sample within the frame */
        for(int n = 0; n < numSamples; n++)
            /* Calculate the sample... */
            float sample = gAmplitude *
                sin(2.0 * M_PI * (frequencies[channel] *
                    gFrequency + phase[channel]));
}
```



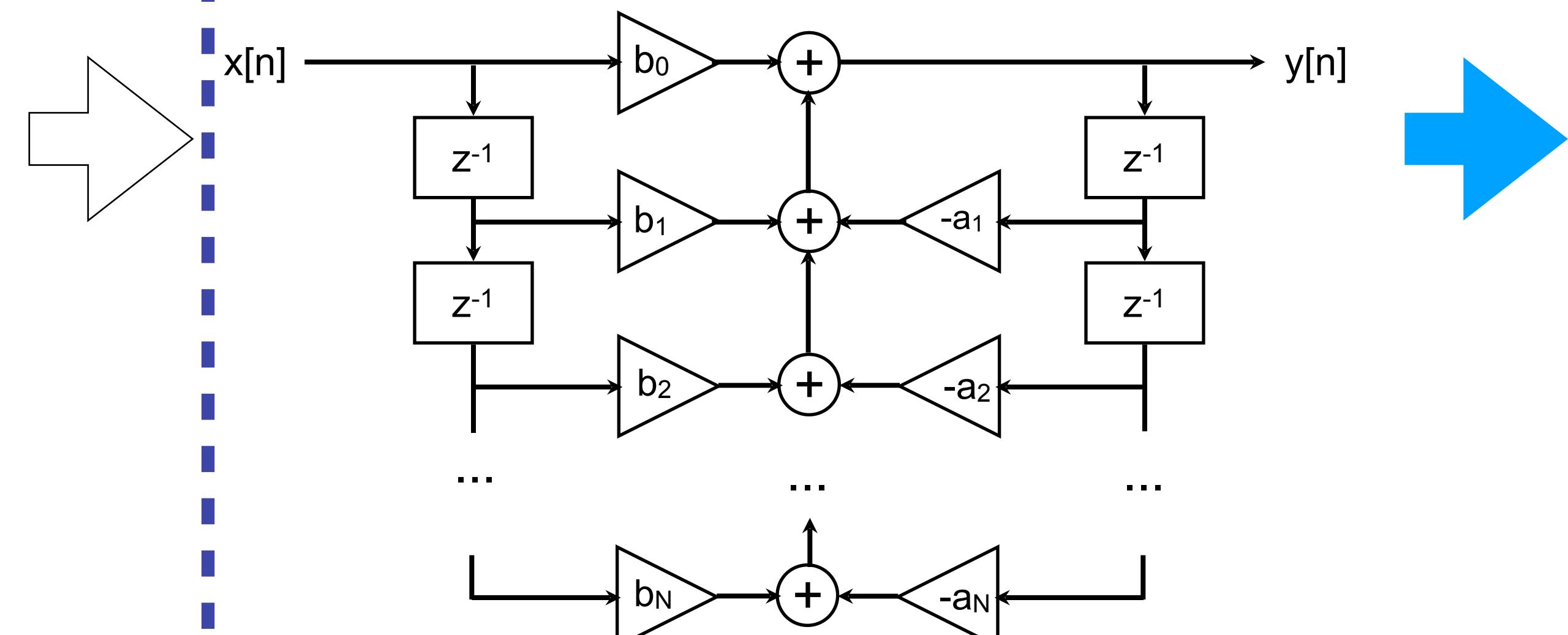
The course in a nutshell

Specifications



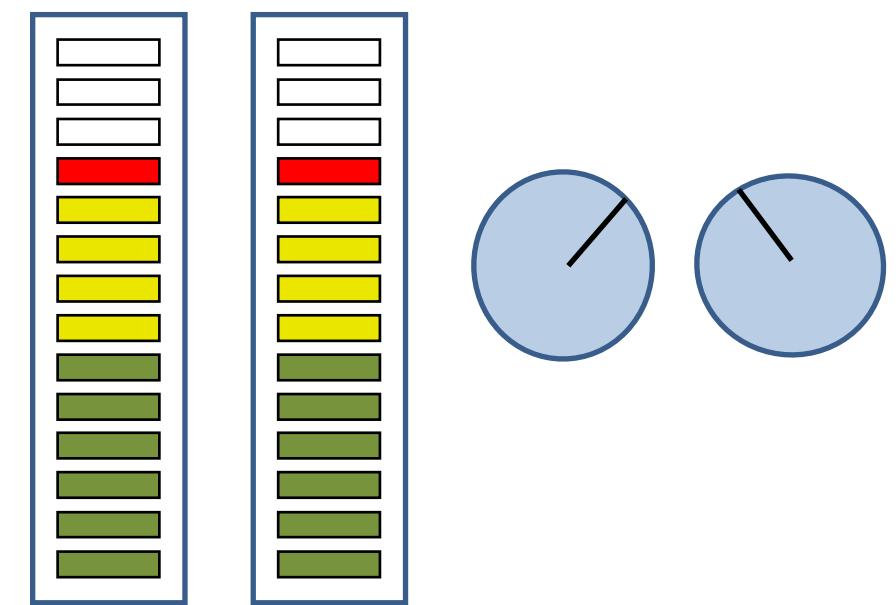
Design

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k]$$



Implementation

```
void render(BelaContext *context, void * /* user data */)
{
    /* Iterate over each channel */
    for(int channel = 0; channel < numChannels; channel++)
        /* Then iterate over each sample within the frame */
        for(int n = 0; n < numSamples; n++)
            /* Calculate the sample... */
            float sample = gAmplitude *
                sin(2.0 * M_PI * (frequencies[channel] *
                    gFrequency + phase[channel]));
}
```



- We will turn **signal processing theory** into **efficient C++ code**
- We'll use Bela to process audio and sensor signals **in real time**

The course in a nutshell

Programming topics



Music/audio topics

- Working in real time
- Buffers and arrays
- Parameter control
- Classes and objects
- Analog and digital I/O
 - Filtering
 - Circular buffers
 - Timing in real time
 - State machines
 - MIDI
- Block-based processing
 - Threads
 - Fixed point arithmetic
 - ARM assembly language

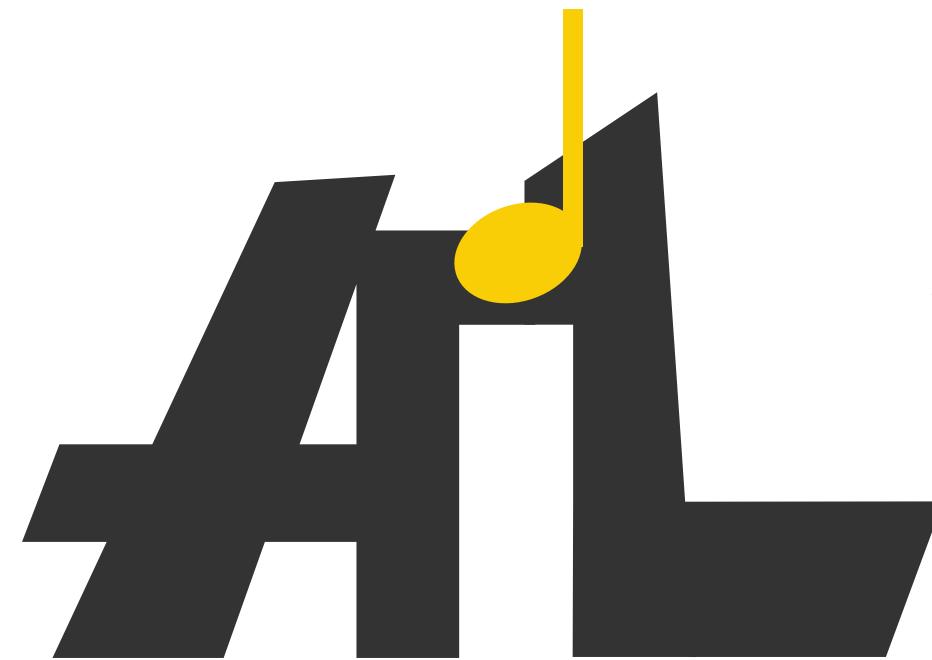
- Oscillators
- Samples
- Wavetables
- Filters
- Control voltages
- Gates and triggers
- Delays and delay-based effects
- Metronomes and clocks
- Envelopes
- ADSR
- MIDI
- Additive synthesis
- Phase vocoders
- Impulse reverb

About the lecturer



Dr Andrew McPherson

Reader (Associate Professor) in Digital Media
Centre for Digital Music (C4DM)
Queen Mary University of London



**Augmented
Instruments
Laboratory**

a research team creating, studying
and sharing new musical instruments

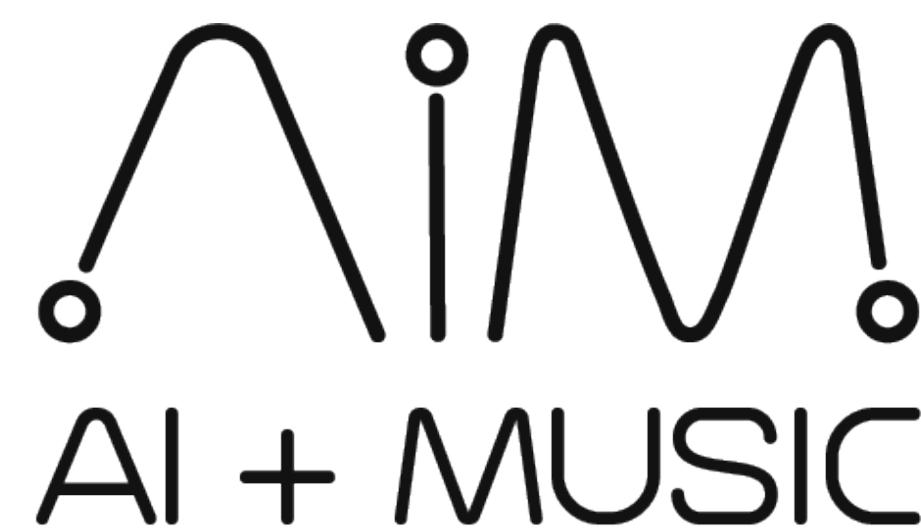
instrumentslab.org

Origins of this course



School of Electronic Engineering
and Computer Science

centre for digital music



- This course is partly based on material from my QMUL class **Music and Audio Programming**
- Part of the QMUL MSc programme in **Sound and Music Computing** and the PhD in **Artificial Intelligence and Music**
- Bela was developed partly as a result of teaching this class

Lecture 0: Setting up

What you'll learn today:

Setting up your Bela kit

Running your first projects

Using the Bela IDE

Companion materials:

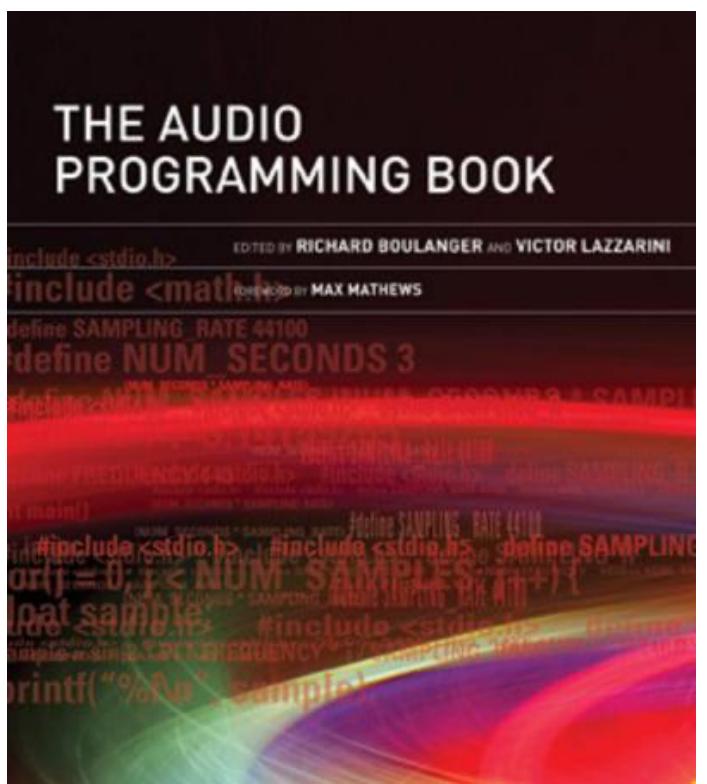
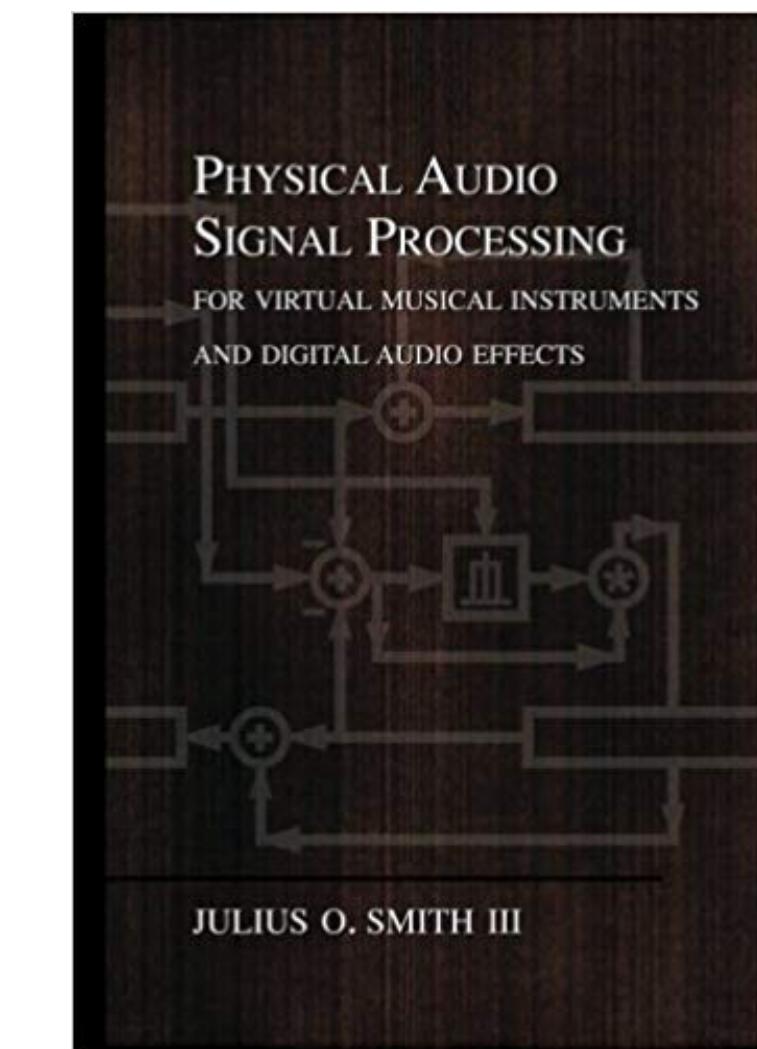
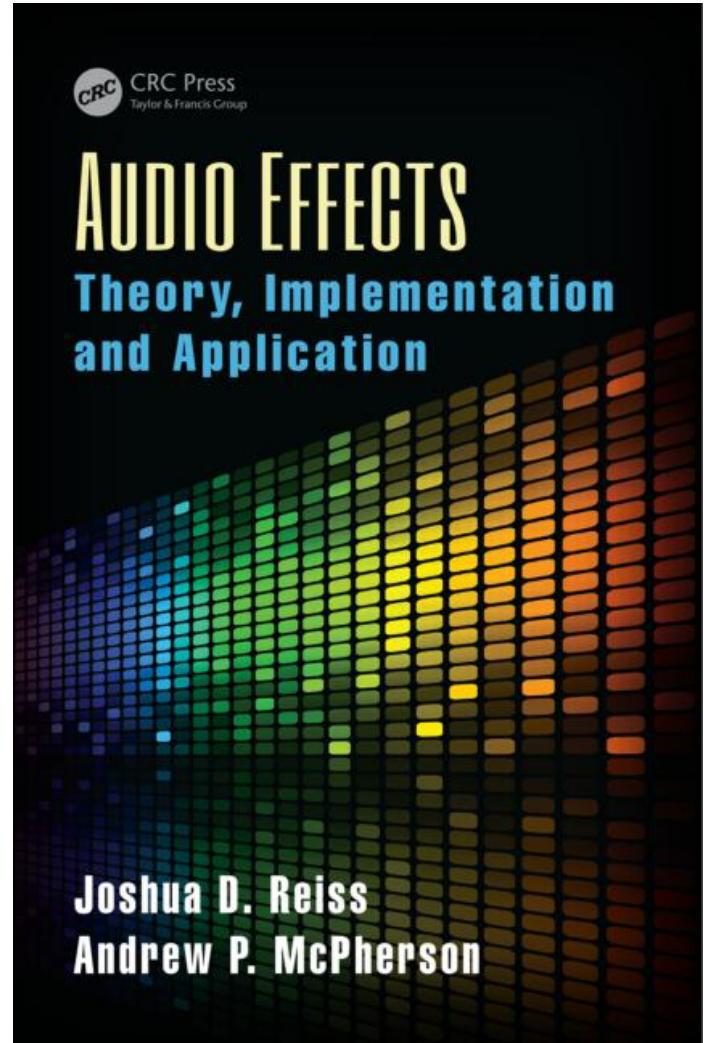
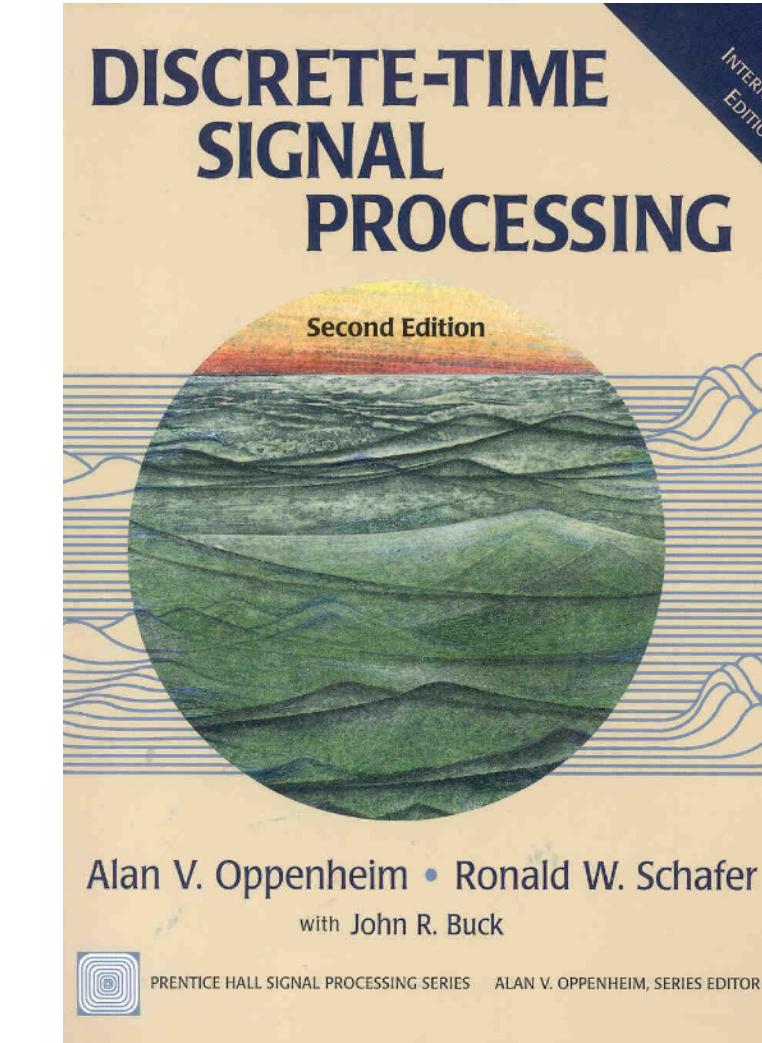
github.com/BelaPlatform/bela-online-course

Code examples

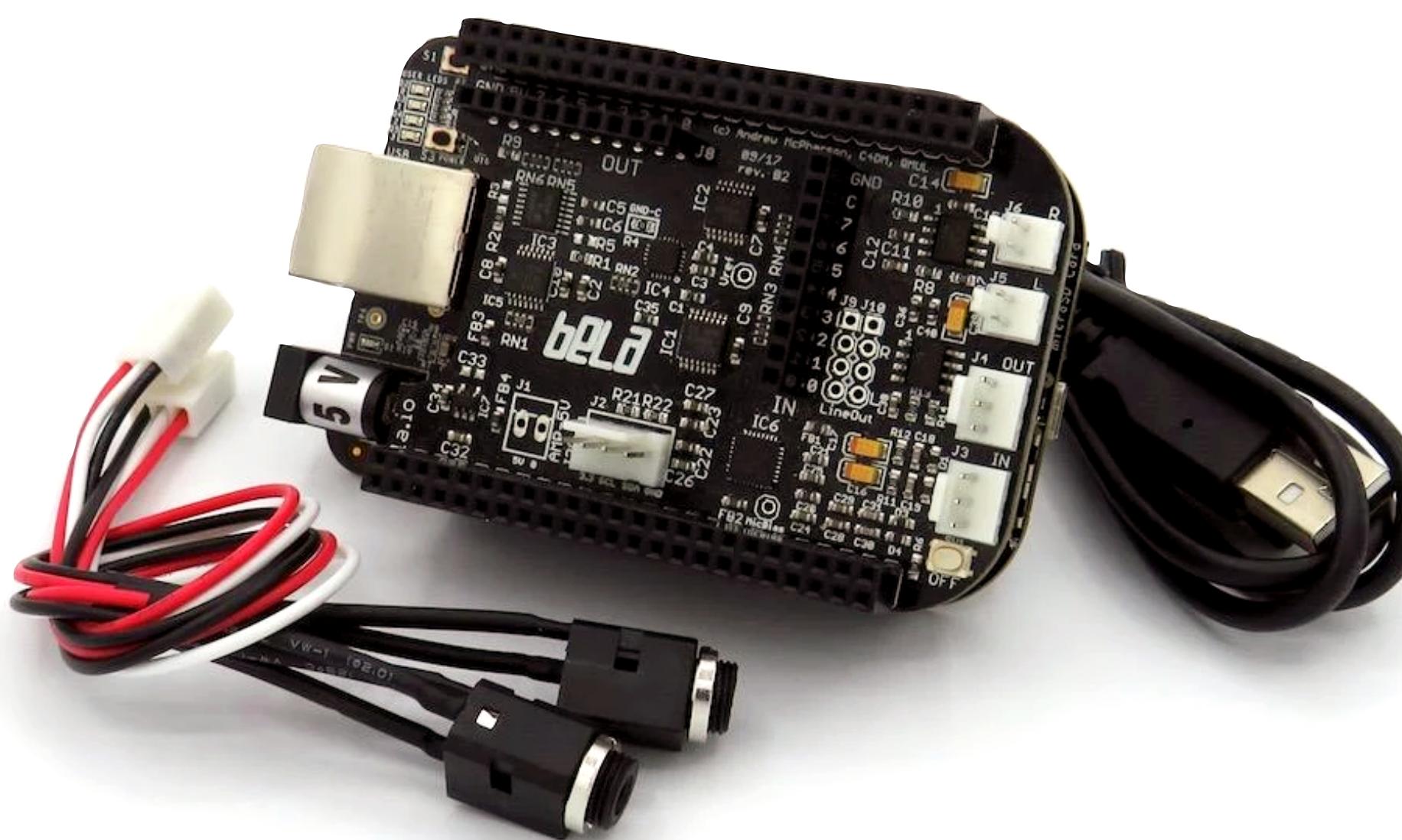
Links and references

Books and references

- There are lots of good books and online resources about DSP and audio programming. For example:
 - Oppenheim, Schafer and Buck, *Discrete-Time Signal Processing*
 - Julius O. Smith, *Physical Audio Signal Processing*
 - Richard Boulanger and Victor Lazzarini (editors), *The Audio Programming Book*
 - Reiss and McPherson, *Audio Effects: Theory, Implementation and Application*
- See the GitHub companion site for links:
github.com/BelaPlatform/bela-online-course
- We will work with hardware in this course
 - Datasheets and reference manuals are always useful
 - We will post links as we go along

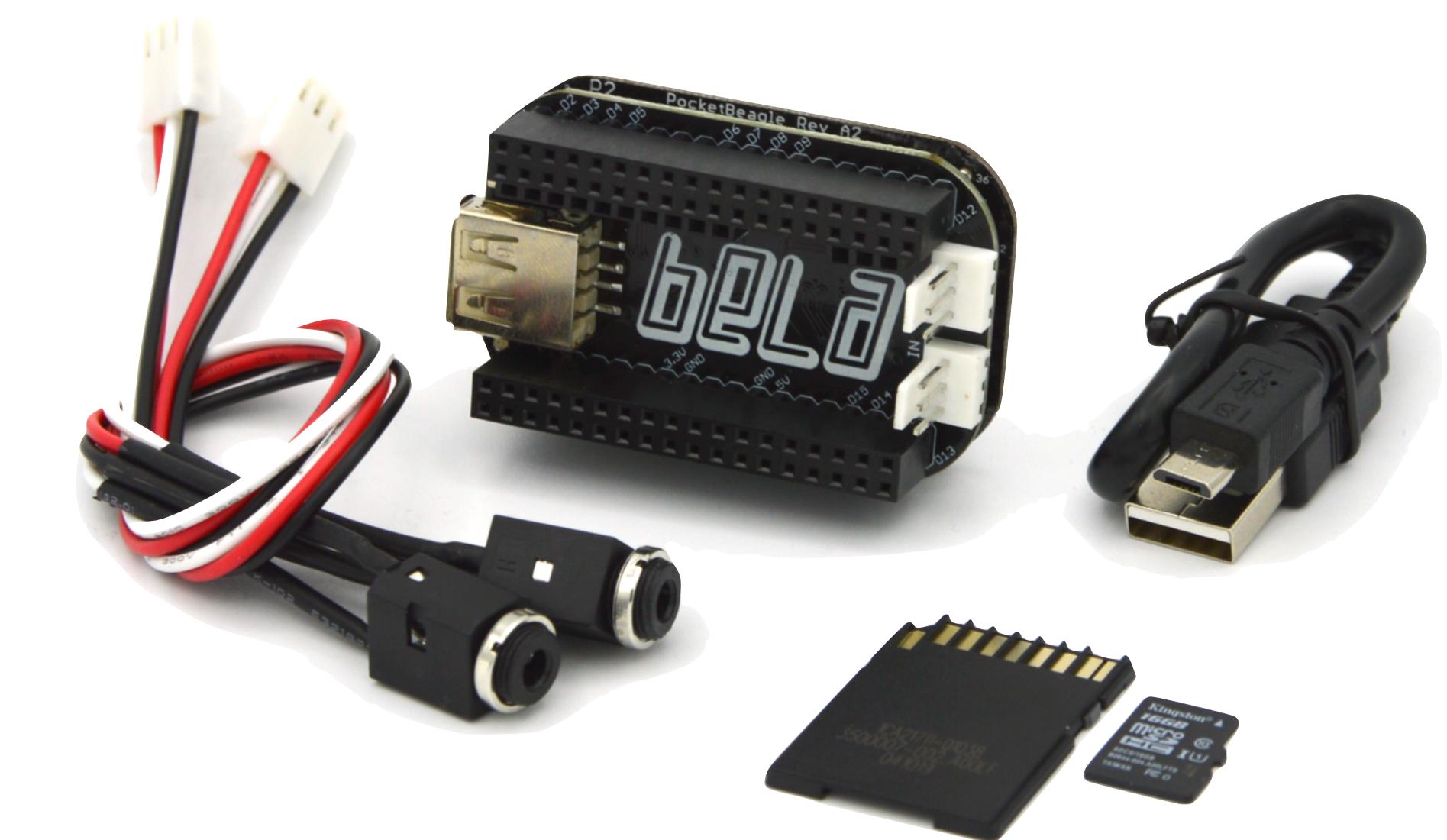


What you'll need



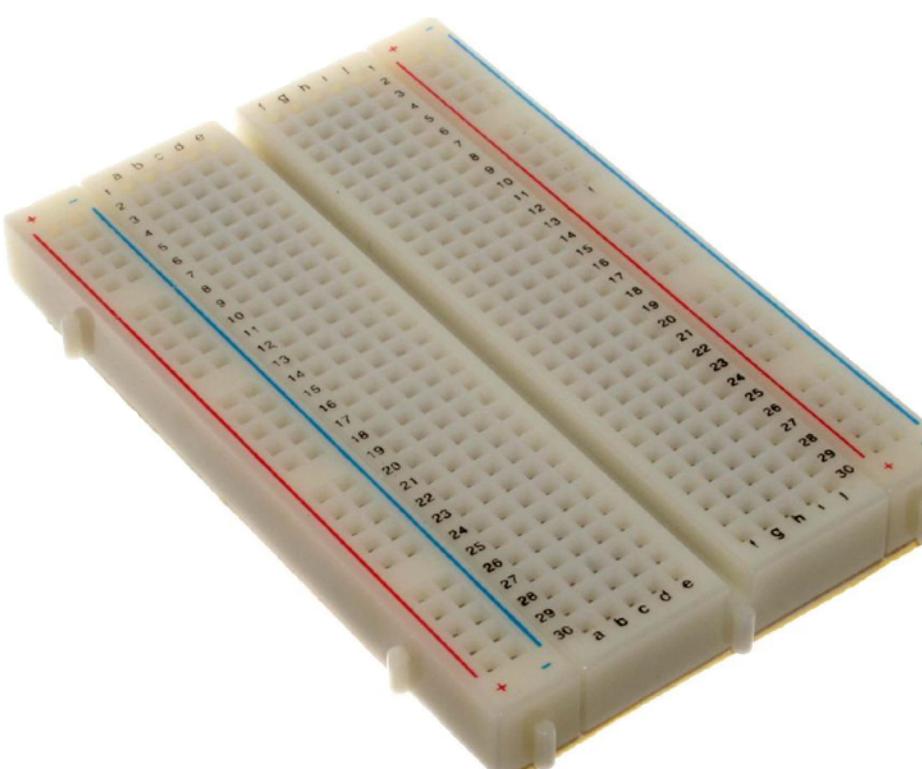
Bela Starter Kit

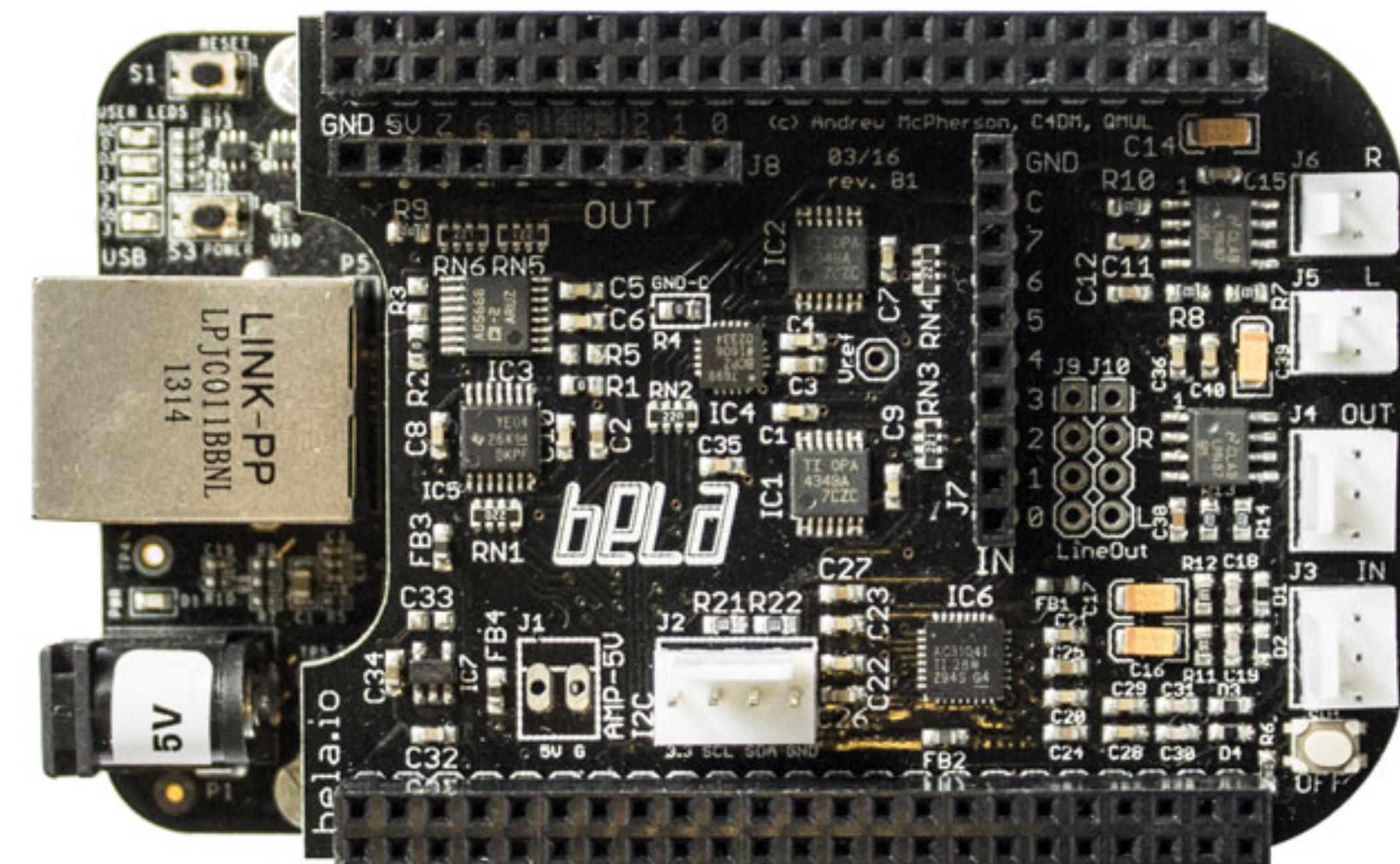
or



Bela Mini Starter Kit

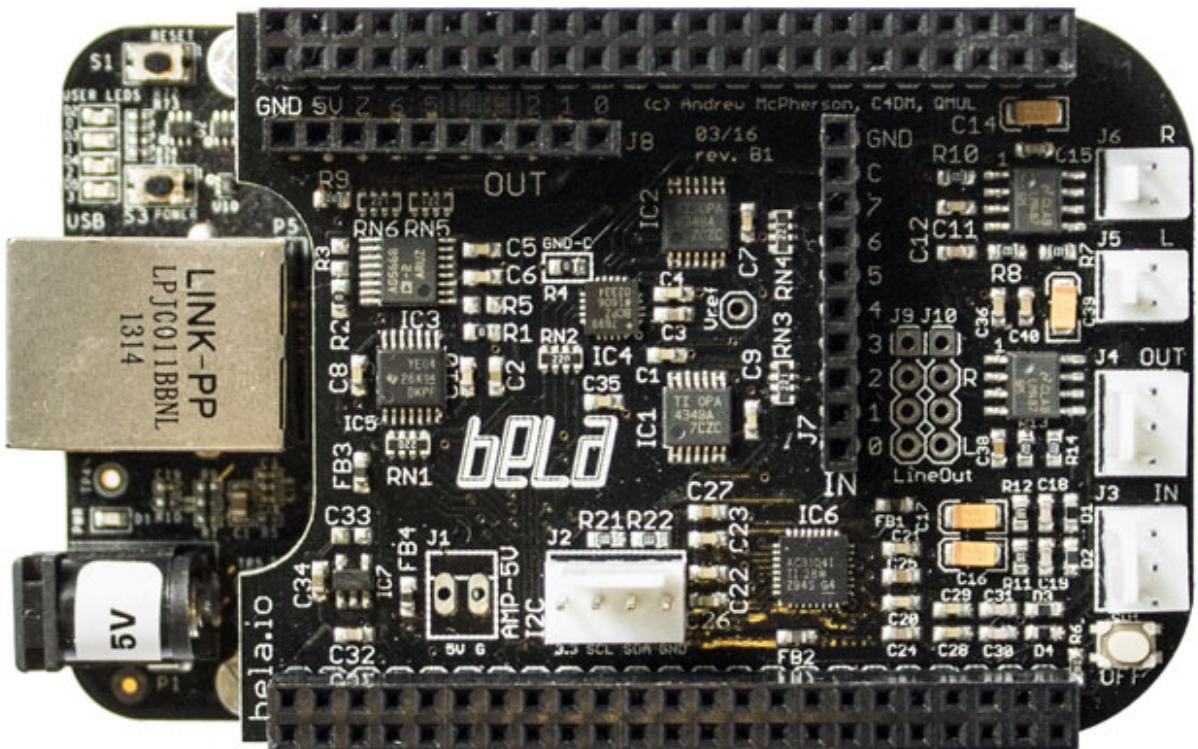
Recommended
for some lectures:





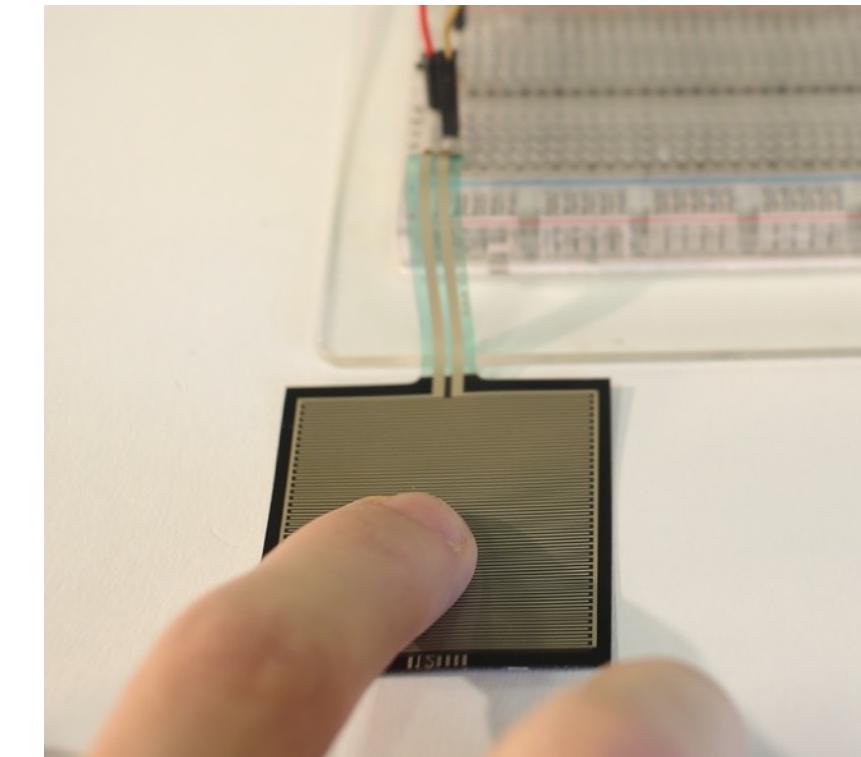
What is Bela?

Embedded computer
designed for
interactive audio



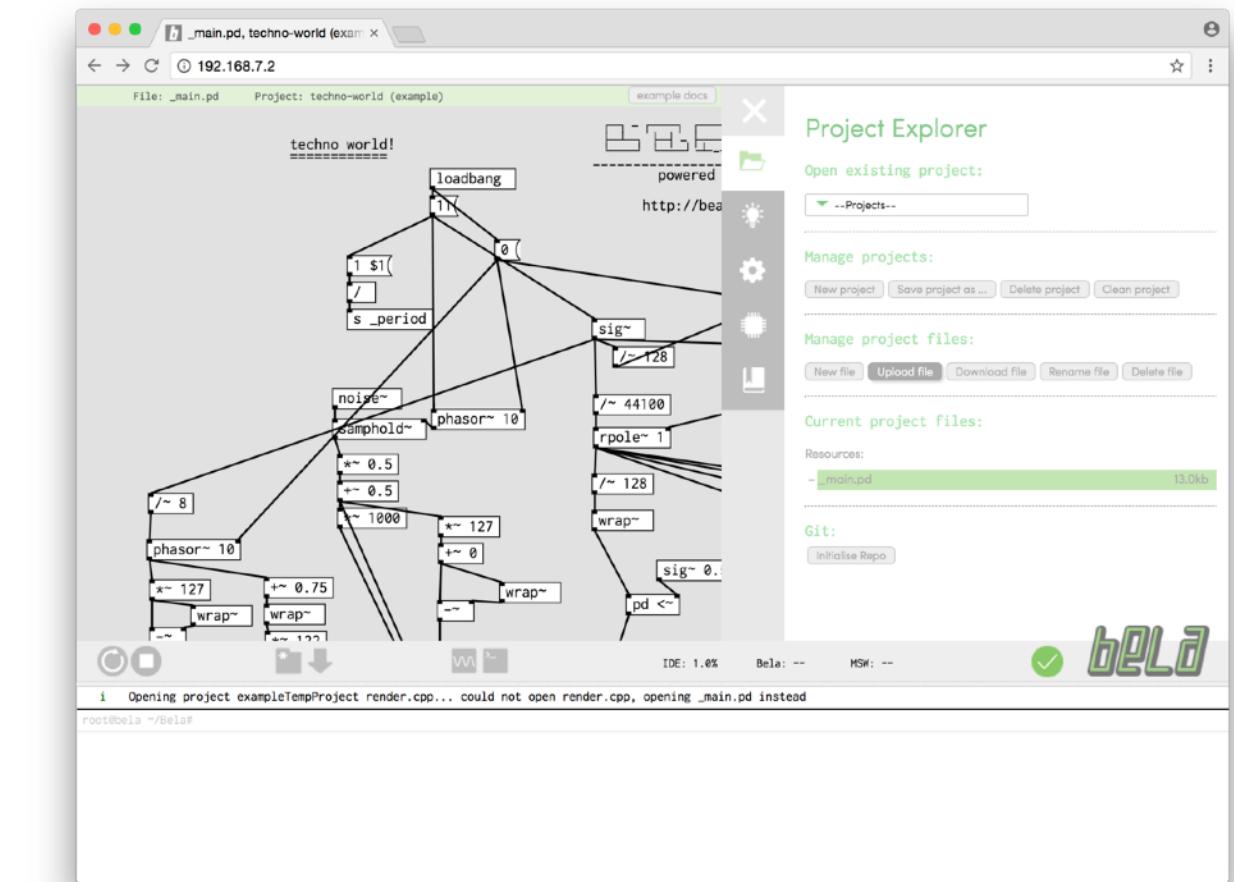
- Power of a single board computer
- Connectivity of a micro-controller
- Combines the benefits of both

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

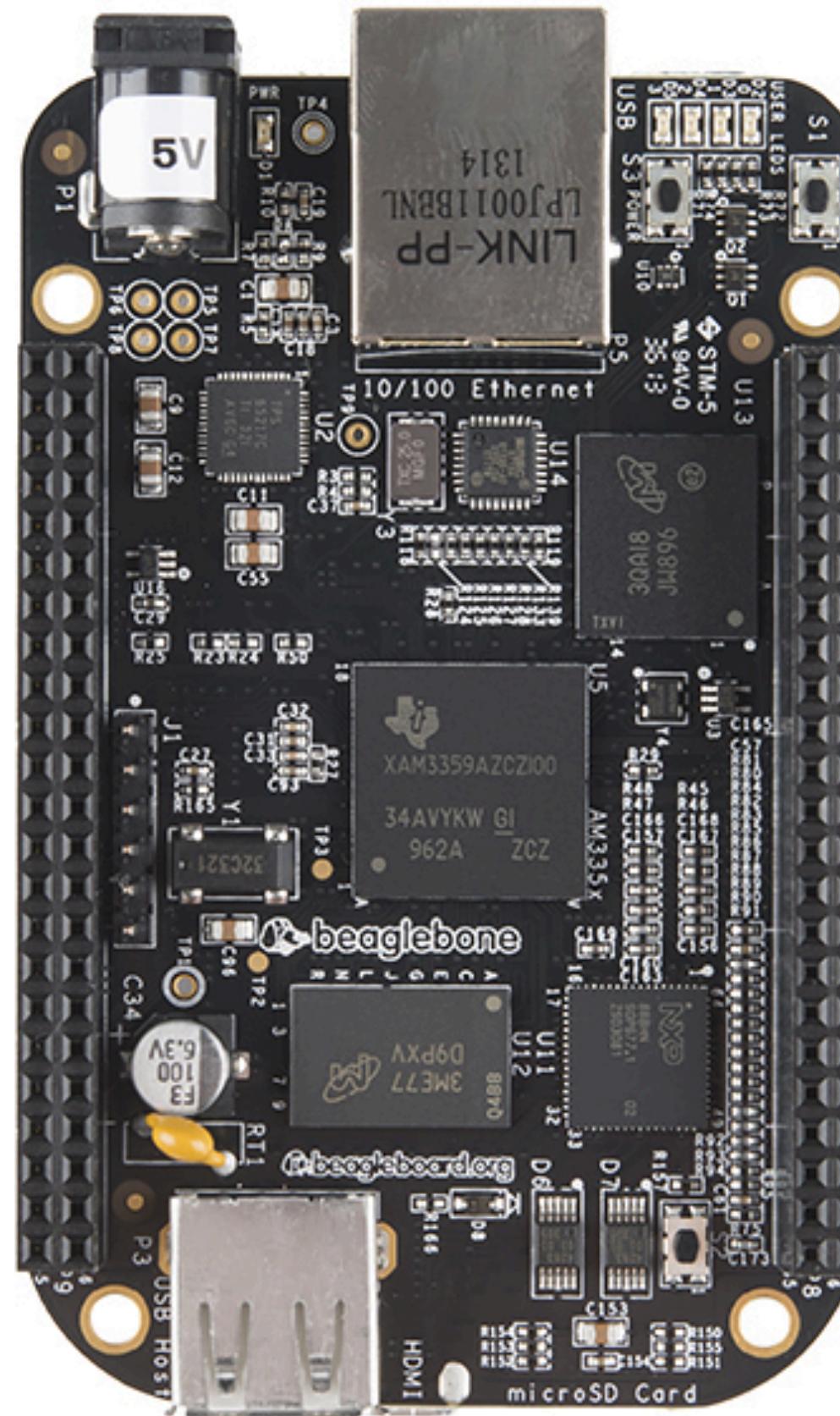


- Open-source hardware and software
- Targeted at musicians, artists, makers, designers
- Online community resources: forum, blog, knowledge base

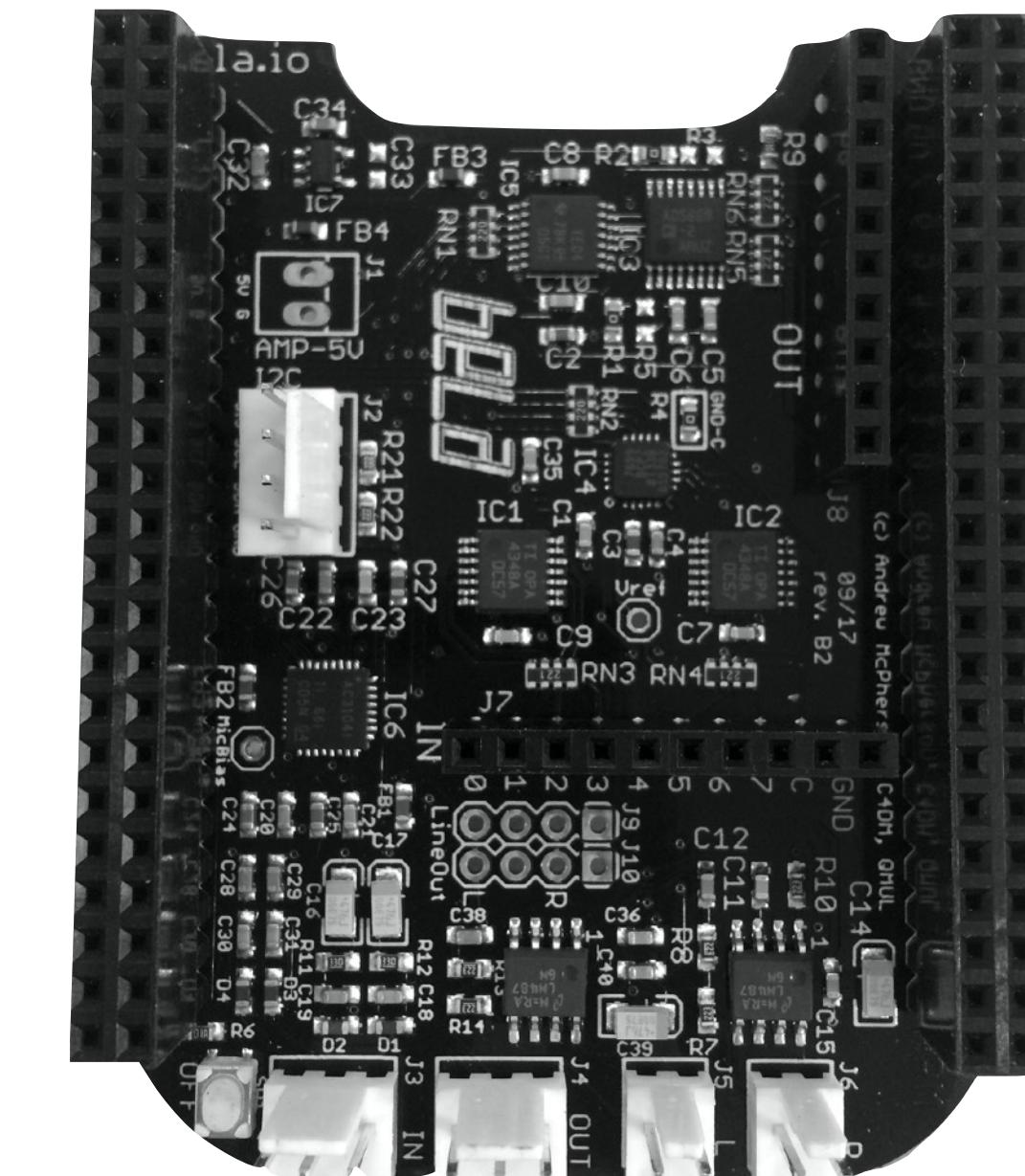
Bela hardware

BeagleBone Black →
1GHz ARM Cortex-A8
512MB RAM
NEON vector floating point
Dual PRU microcontrollers

Embedded Computer



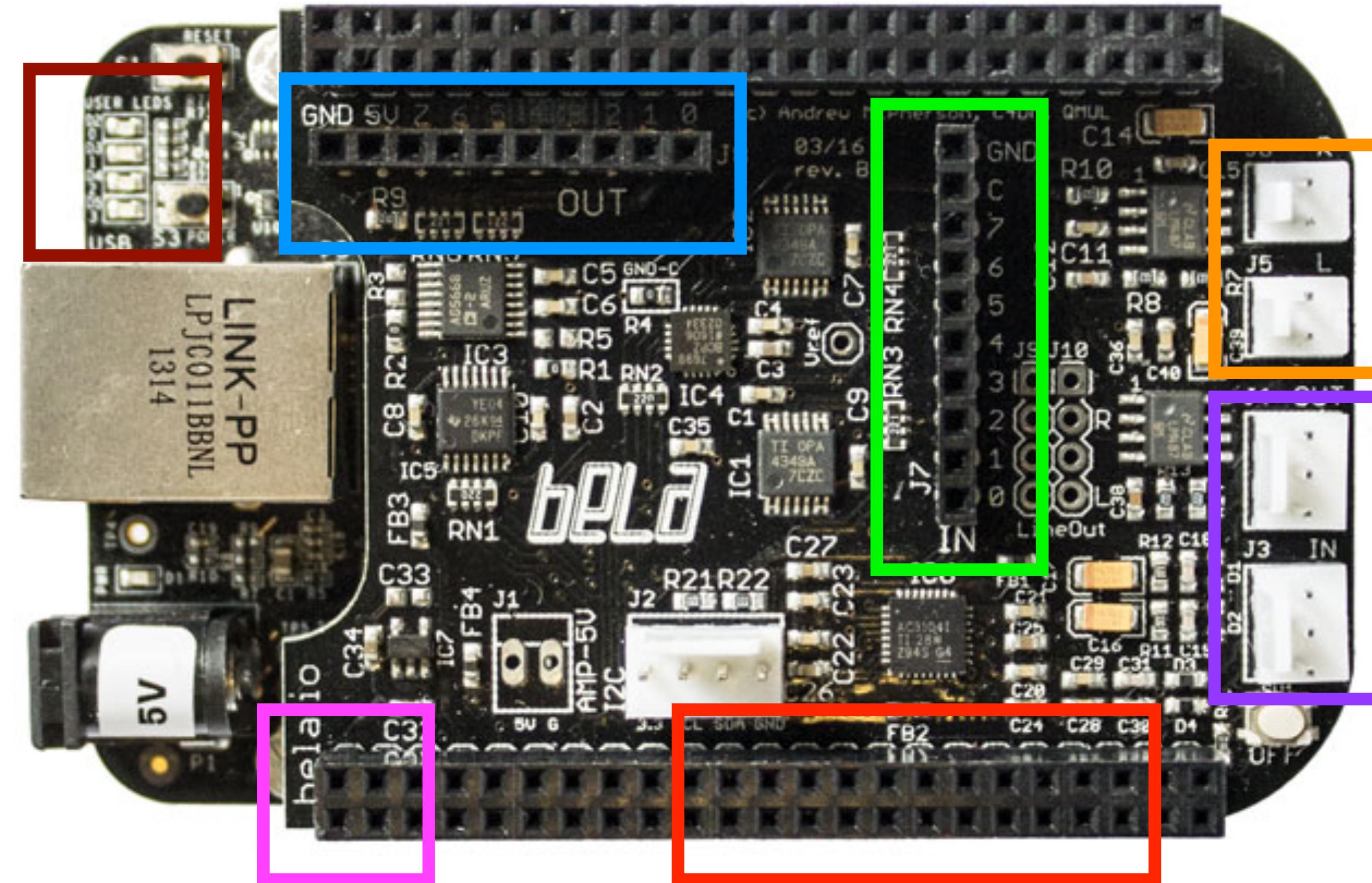
Bela Audio/Sensor Cape



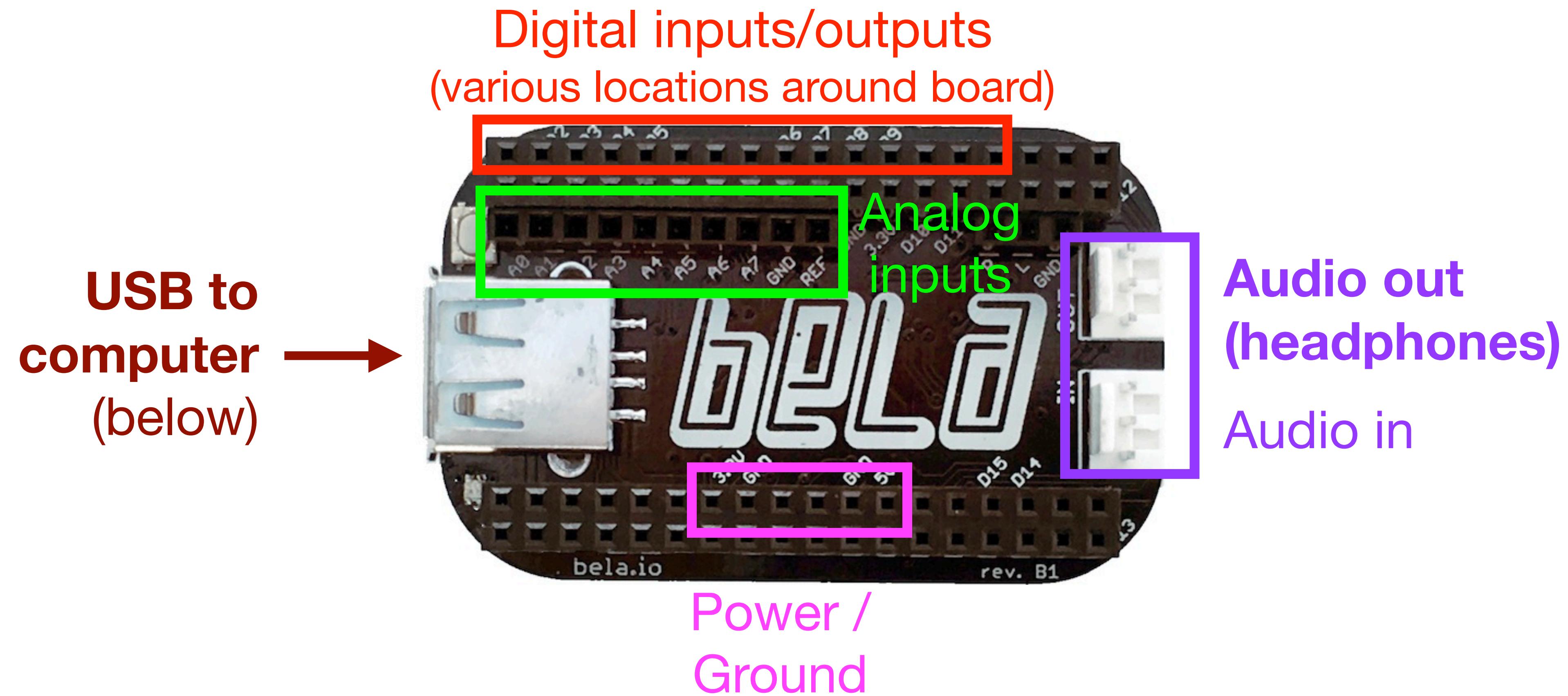
The Bela Cape
16-bit audio I/O
16-bit analog I/O
stereo 1W speaker amplifiers

Bela hardware

USB to computer

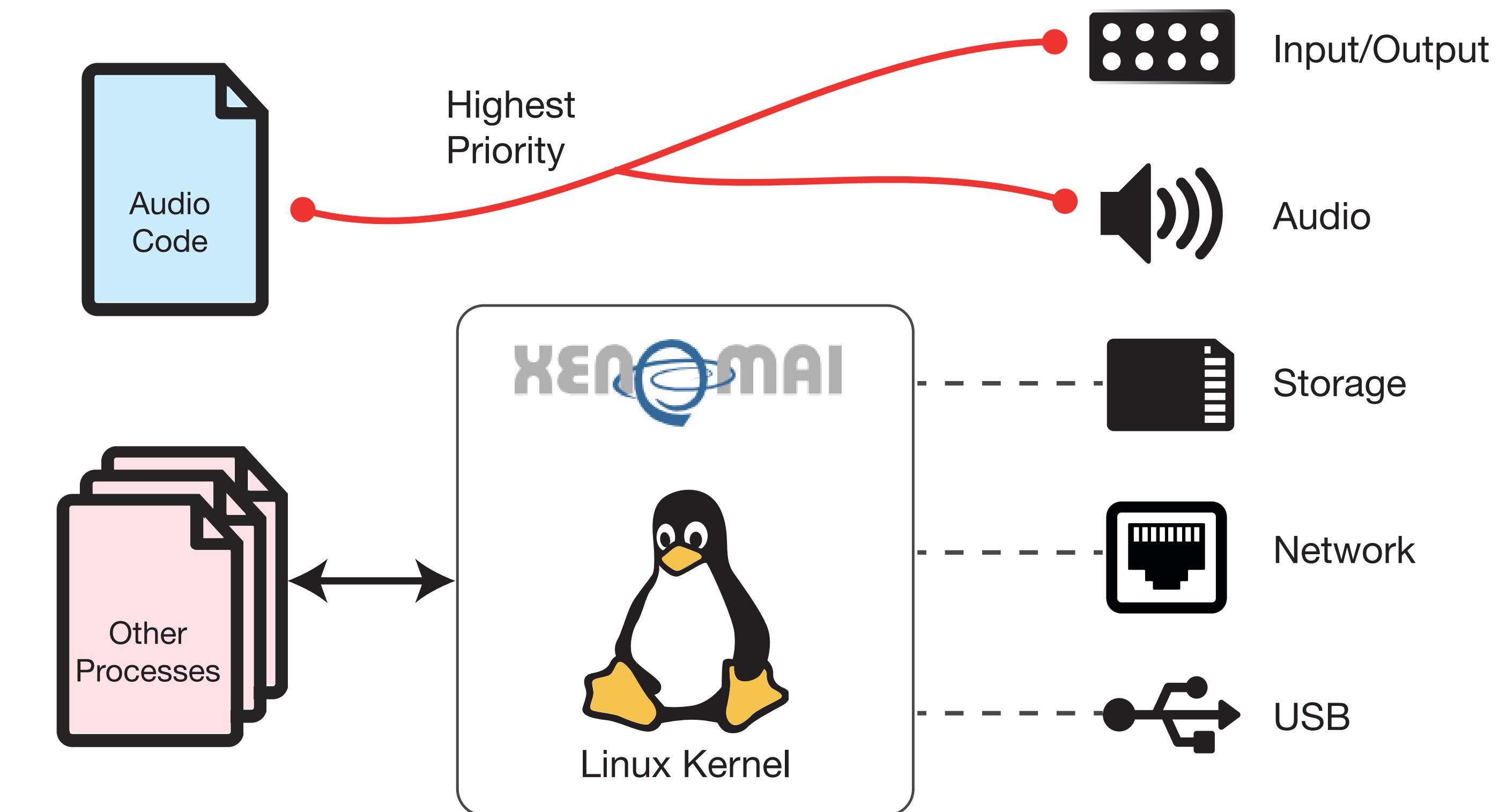


Bela Mini hardware



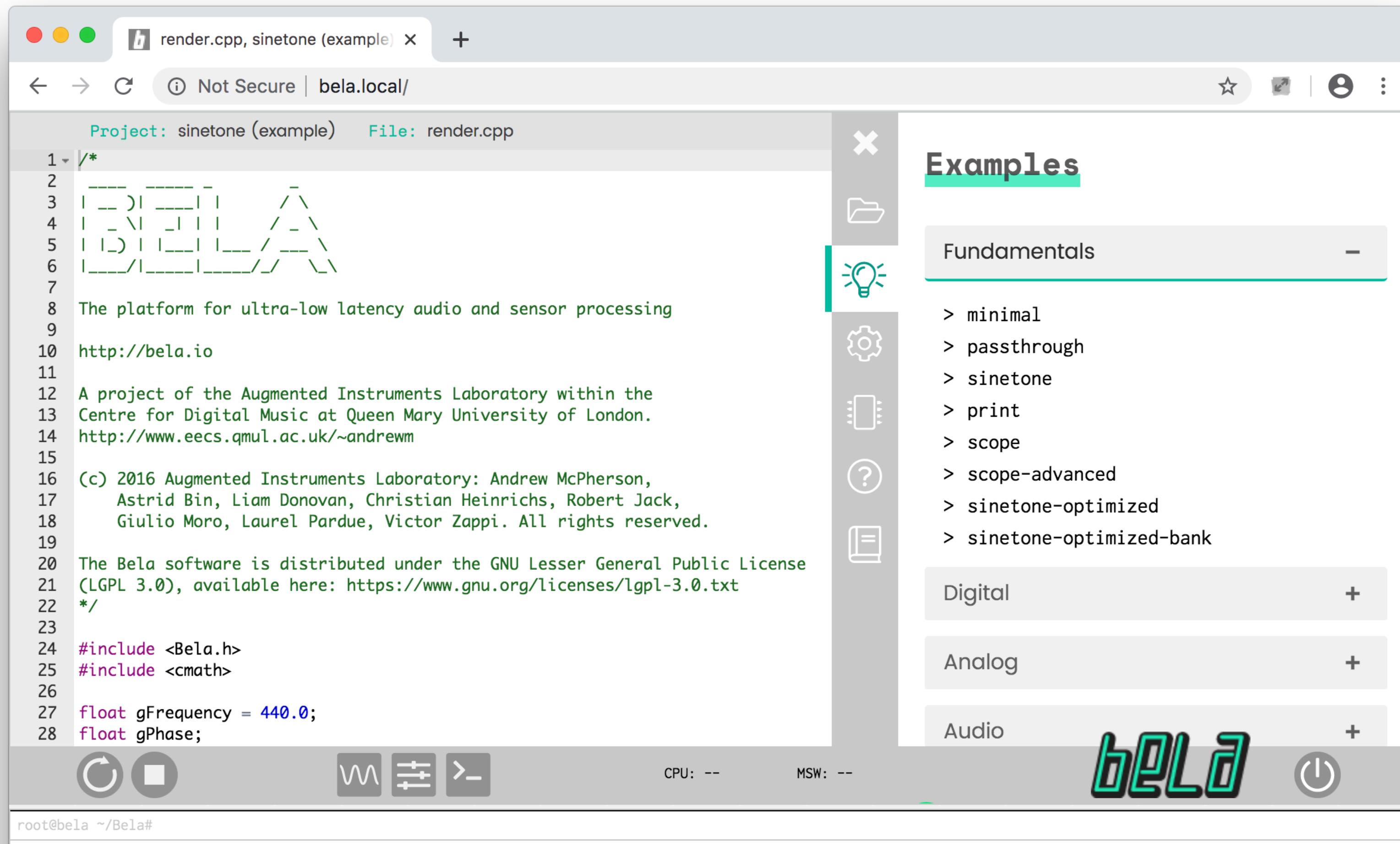
Bela software

- Bela uses the **Xenomai** real-time Linux extensions to run audio code at near **bare-metal priority**
 - Guarantees that our audio code **always runs first** on the hardware
 - Still have access to all of the capabilities of Linux
- Bela achieves **extremely low audio latency** (delay)
 - More on this later!



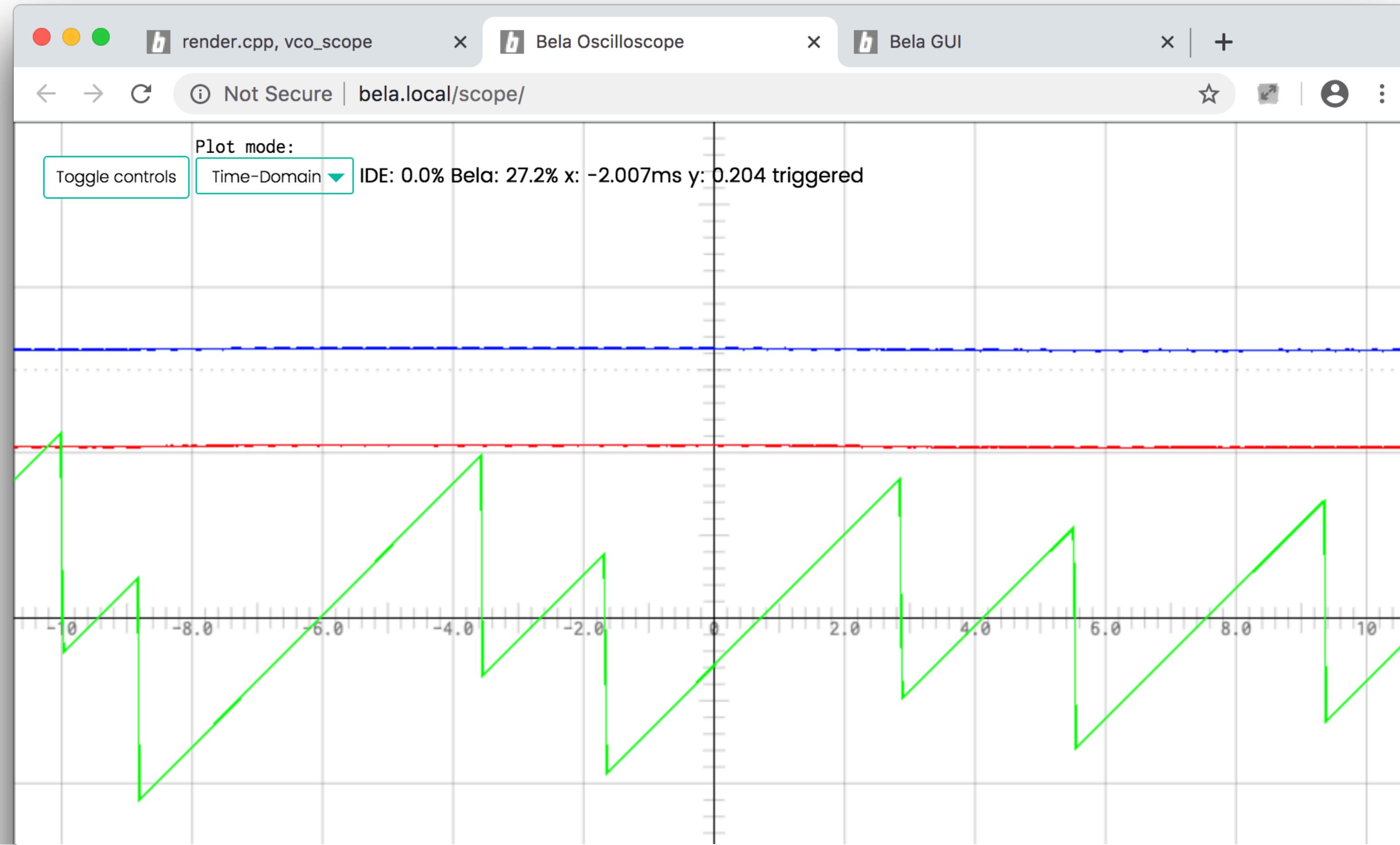
Bela software

- Bela has a built-in, browser-based IDE for developing code



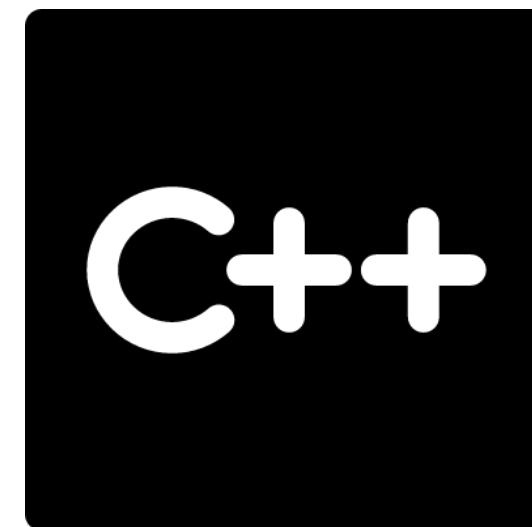
Bela software

- Bela has a built-in, browser-based IDE for developing code
 - Including a browser-based oscilloscope for visualising signals

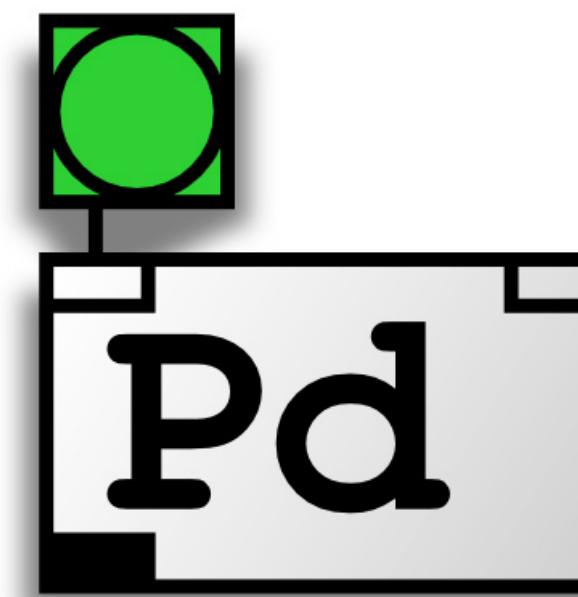


Bela software

- Bela supports a number of programming languages:



C / C++



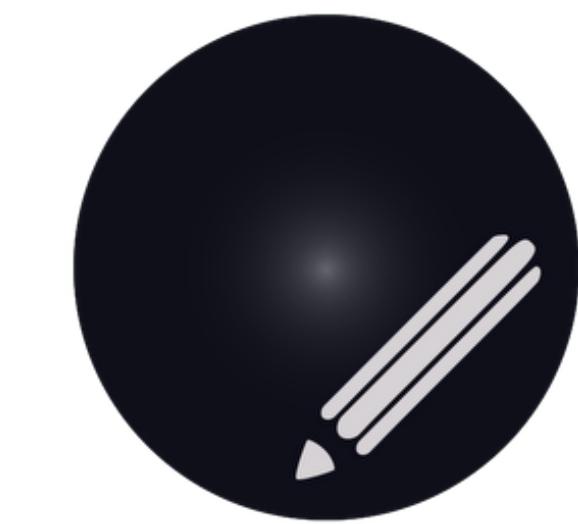
Pure Data



SuperCollider



Functional
Audio
Stream



pyo

- In this course we will be working in **C++**

- Compared to the audio-specific languages, C/C++ is closer to what the hardware is doing
- Lets us understand the low-level implementation of DSP concepts
- Possible to write very efficient code (but we still have to work at it!)

Where to find support

The Bela
Knowledge Base:

learn.bela.io

The screenshot shows a web browser window for the Bela Knowledge Base at learn.bela.io. The page features a header with the Bela logo and a search bar. Below the header, the title "Bela Knowledge Base" is highlighted in green. The main content area contains three sections: "Get Started Guide" (with a teal rocket icon), "Tutorials" (with a teal lightbulb icon), and "Using Bela" (with a teal question mark icon). Each section includes a brief description and a link to more information.

Bela Knowledge Base

This is the central hub for creating and learning with Bela.

Use the navigation menu on the left, or choose a section below:

Get Started Guide
Learn all about Bela, and get to know your hardware and software.

Tutorials
Access our tutorials in C++, Puredata, and more.

Using Bela
Articles on how to embed, update, and Bela, and our technical explainers.

Where to find support

The Bela support
forum:

forum.bela.io

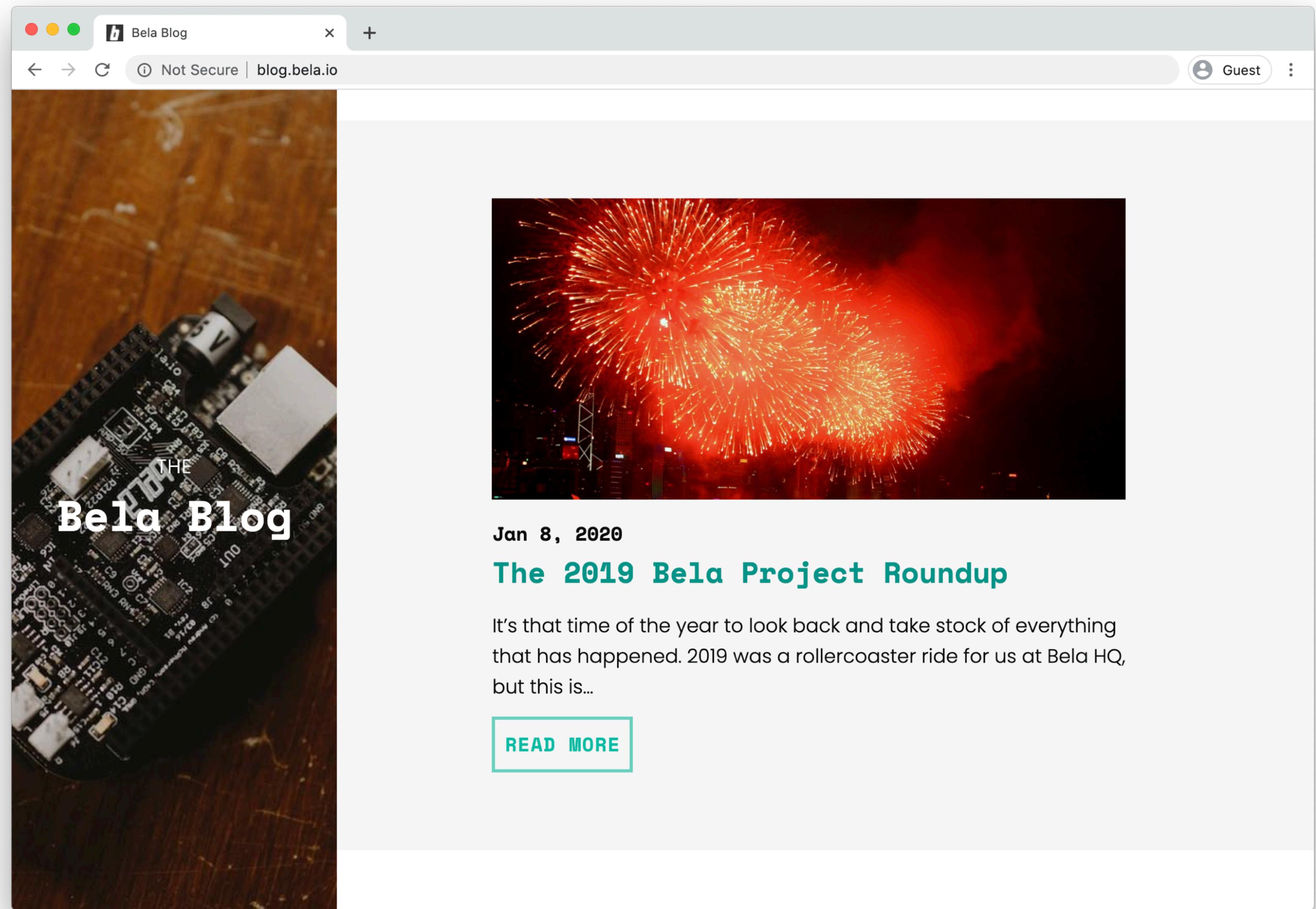
The screenshot shows a web browser window for the forum.bela.io website. The header includes the Bela logo, navigation links for Home, Guidelines, Code, Docs, and bela.io, a search bar, and options to Sign Up or Log In. The main content area is titled "Welcome to the Bela community" with a sub-section about getting support and sharing ideas. On the left, there's a sidebar with a "Start a Discussion" button, a dropdown menu set to "Latest", and a list of discussion tags: All Discussions, Tags, FAQ, Interactivity, Software, General, Getting Started, Audio, Hardware, Show and Tell, Forum, CTAG-ALSA, and Solved. The main content area displays several posts from the community:

- New IDE: GUI and shareable libraries** by djzielin (17 replies) - A post announcing a new version of the Bela codebase with a new IDE design and improved workflow.
- Bela Platform FAQ** by giuliomoro (3 replies) - A post linking to the official FAQ page and providing information about the FAQ on the Kickstarter page.
- Getting Started Instructions** by andrew (0 replies) - A post providing instructions for getting started with Bela, pointing to the GitHub wiki.
- Bela Mini: bela in miniature** by SEDur (22 replies) - A post announcing the release of the Bela Mini board.
- New blog for projects: blog.bela.io** (0 replies) - A post announcing a new blog for Bela projects.

Where to find support

The Bela blog:

blog.bela.io



Where to find support

The Bela
GitHub repository:

[github.com/
BelaPlatform/Bela](https://github.com/BelaPlatform/Bela)

More technical
documentation at
the [Wiki link](#)

The screenshot shows the GitHub repository page for 'BelaPlatform / Bela'. The repository has 2,399 commits, 43 branches, 0 packages, 0 releases, 23 contributors, and a master branch selected. The latest commit was made 19 hours ago. The repository description is 'Bela: core code, IDE and lots of fun!'. The commit list includes changes to .settings, IDE, core, examples, include, libraries, pru, resources, scripts, .cproject, and .aitianore.

Commit Details	Time Ago
giuliomoro Merge branch 'dev' into 'master'	Latest commit afae18c 19 hours ago
.settings Change include order so cobalt macro expansions are picked up correctly.	2 years ago
IDE IDE: removing # from address bar when opening side tab	2 days ago
core core: avoid people sending use dmesg logs for no reason: it's the McA...	2 months ago
examples Merge branch 'dev' into 'master'	19 hours ago
include Typo	3 months ago
libraries Gui: post-merge fixup	3 months ago
pru Cleaning GPIOs on exit: PRU writes zeros, ARM sets values to 0 if fai...	14 months ago
resources Csound: updated to 6.14	last month
scripts scripts: not copy IDE/ide-dev.js during update_board	2 months ago
.cproject Change include order so cobalt macro expansions are picked up correctly.	2 years ago
.aitianore ignored temp folder for IDE	8 months ago

Where to find support

Materials for
this course:

**github.com/
BelaPlatform/
bela-online-course**

Code
Lecture slides
Links and resources

The screenshot shows a GitHub repository page for 'BelaPlatform/bela-online-course'. The repository has 3 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The README.md file has been updated 9 days ago. The page includes sections for 'C++ Real-time Audio Programming with Bela', 'Frequently Asked Questions', and 'Where can I find this course?'. It also mentions that the course will be released weekly on the Bela YouTube channel.

Assets and slides for the free YouTube course, C++ Real-time Audio Programming with Bela, released in Spring 2020.

Branch: master ▾ New pull request Find file Clone or download ▾

disastrid Update README.md Latest commit 0f553f6 9 days ago

README.md Update README.md 9 days ago

README.md

C++ Real-time Audio Programming with Bela

This is the repository for the assets for the first free online Bela course, released in Spring 2020.

Frequently Asked Questions

Where can I find this course?

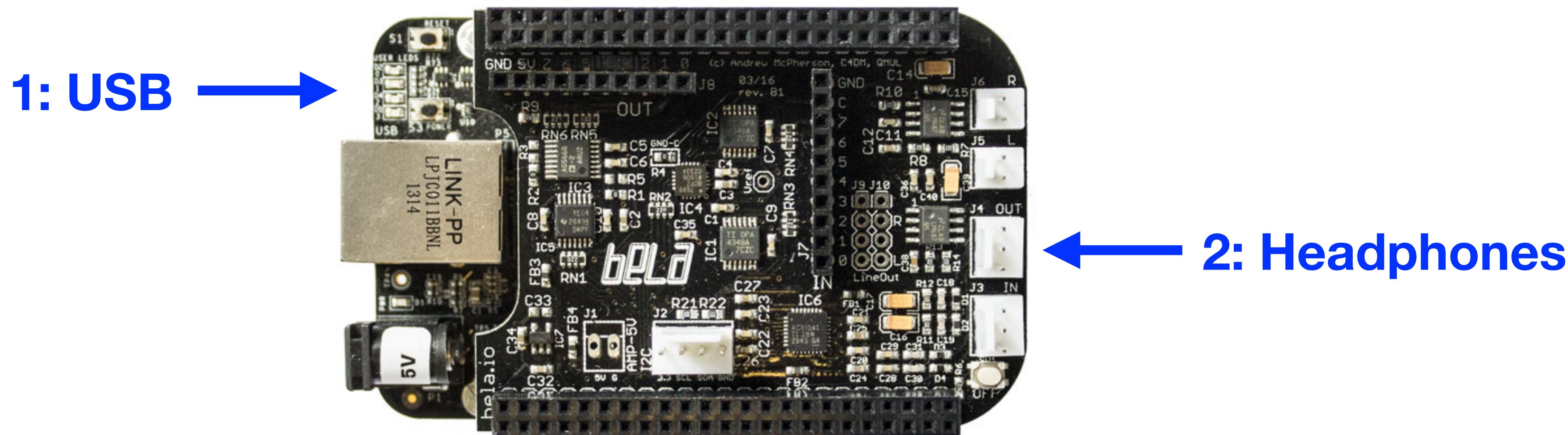
This course will be released weekly on the [Bela YouTube channel](#). Subscribe, and enable alerts so you'll know when new lectures drop.

Getting started

Getting started: connect your board

Running MacOS X Catalina (10.15)? Install drivers: learn.bela.io/drivers

1. Connect the Bela board to your computer with the USB cable
2. Connect the headphones to the audio output via the adapter cable

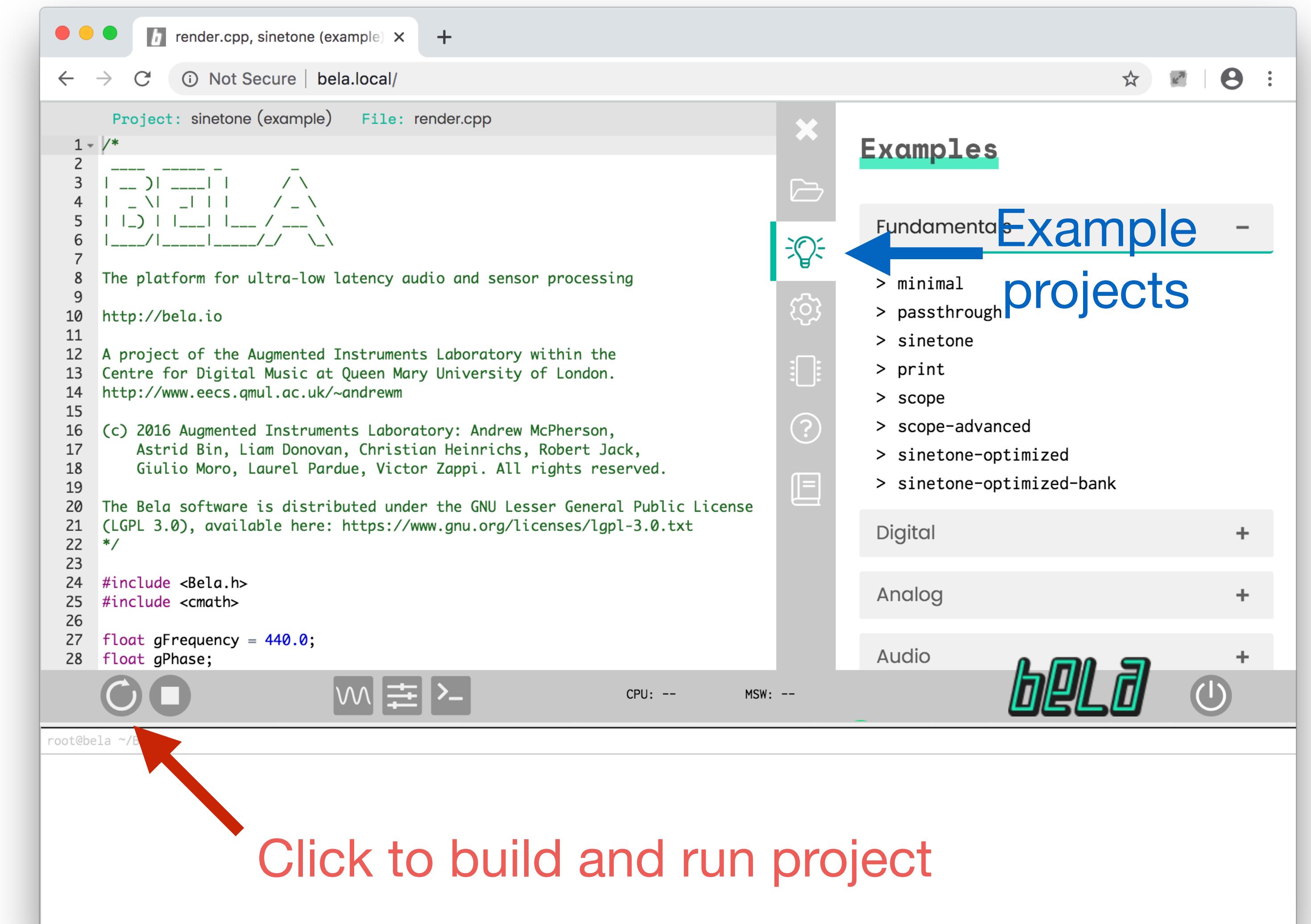


3. Watch for blinking blue lights near the USB connector; wait 15 seconds

The Bela IDE

Bring up the Bela IDE:
<http://bela.local>

or sometimes:
<http://192.168.7.2> (Linux)
or
<http://192.168.6.2> (Windows)



The Bela IDE

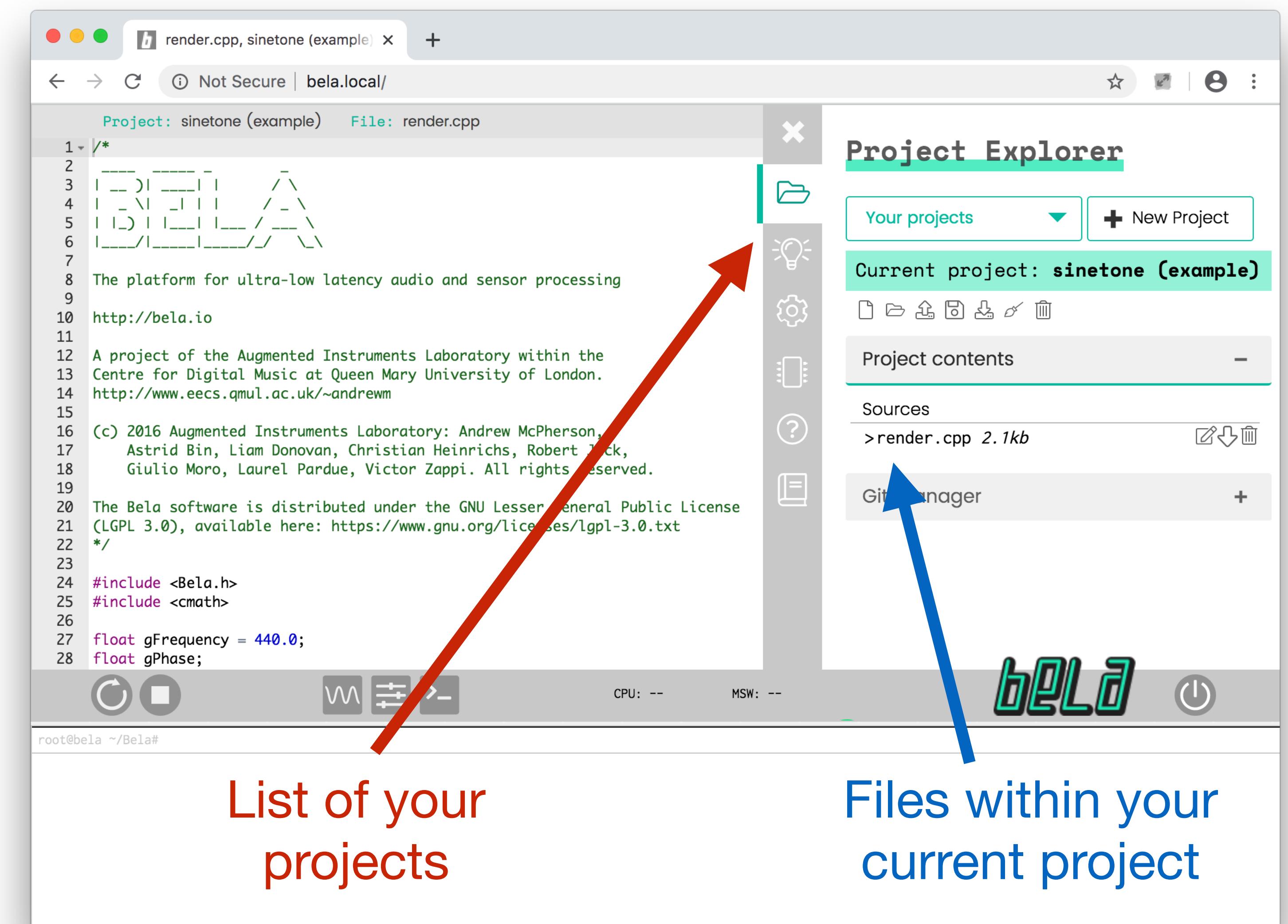
Bring up the Bela IDE:
<http://bela.local>

or sometimes:

<http://192.168.7.2> (Linux)

or

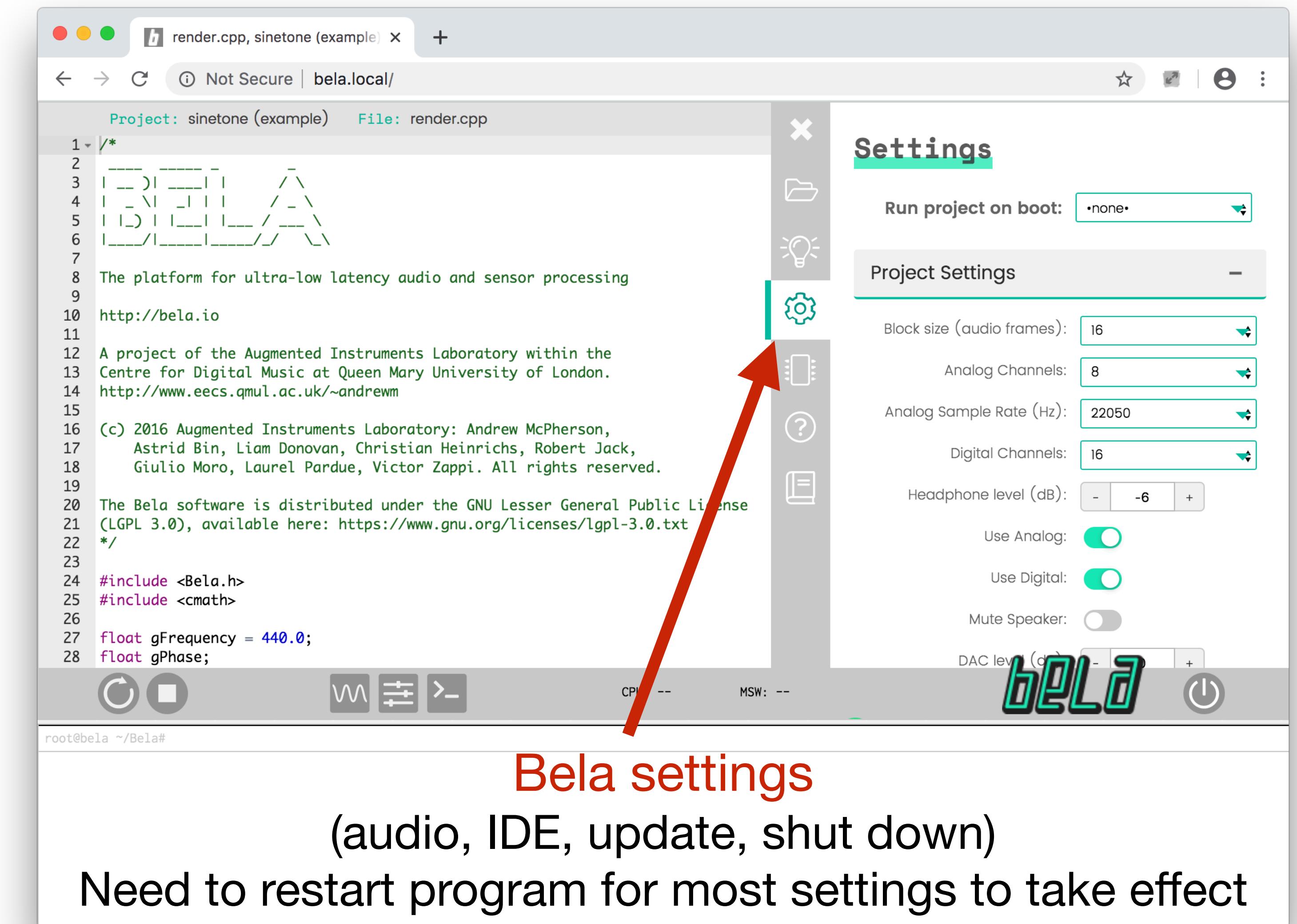
<http://192.168.6.2> (Windows)



The Bela IDE

Bring up the Bela IDE:
<http://bela.local>

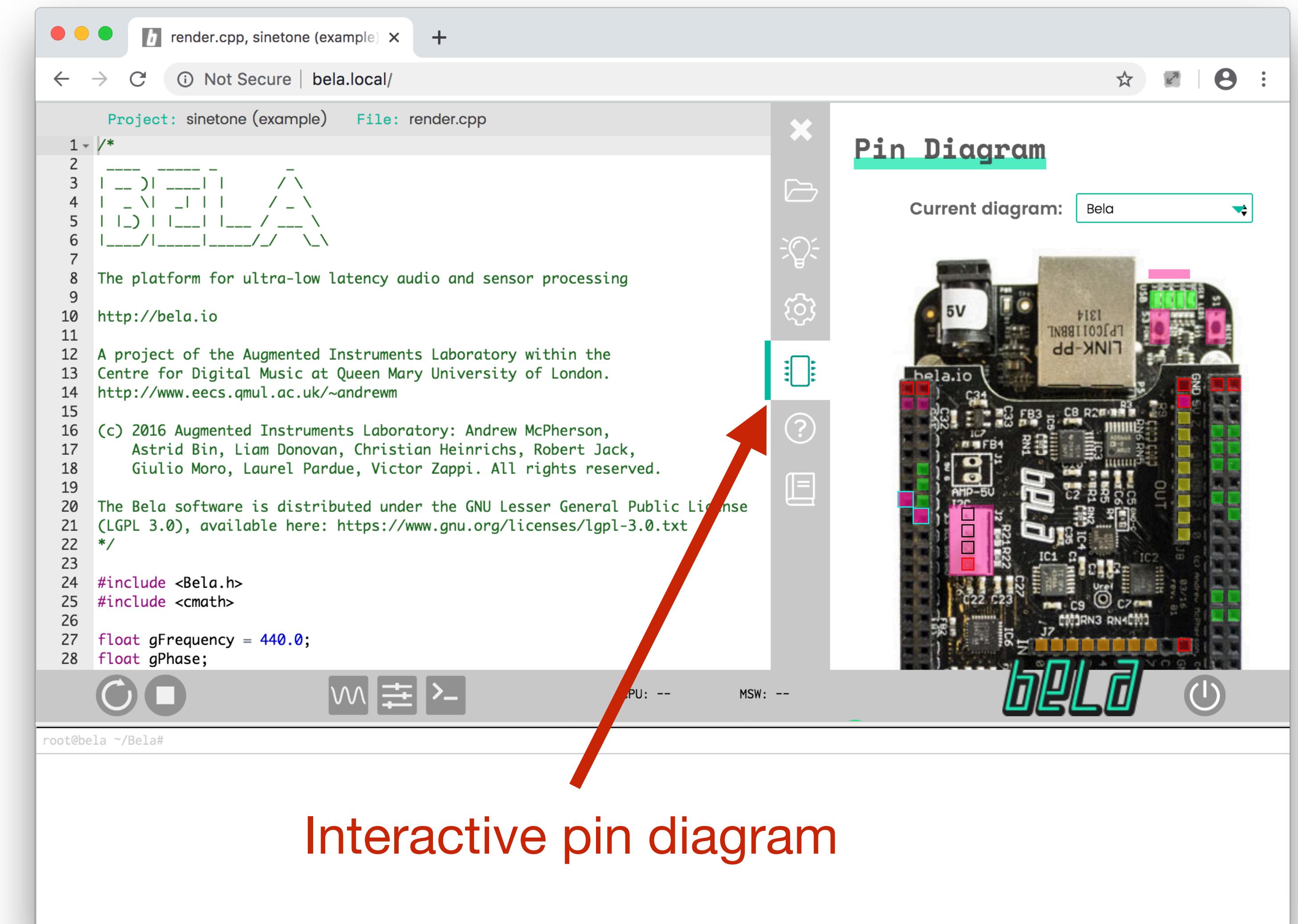
or sometimes:
<http://192.168.7.2> (Linux)
or
<http://192.168.6.2> (Windows)



The Bela IDE

Bring up the Bela IDE:
<http://bela.local>

or sometimes:
<http://192.168.7.2> (Linux)
or
<http://192.168.6.2> (Windows)



Interactive pin diagram

The Bela IDE

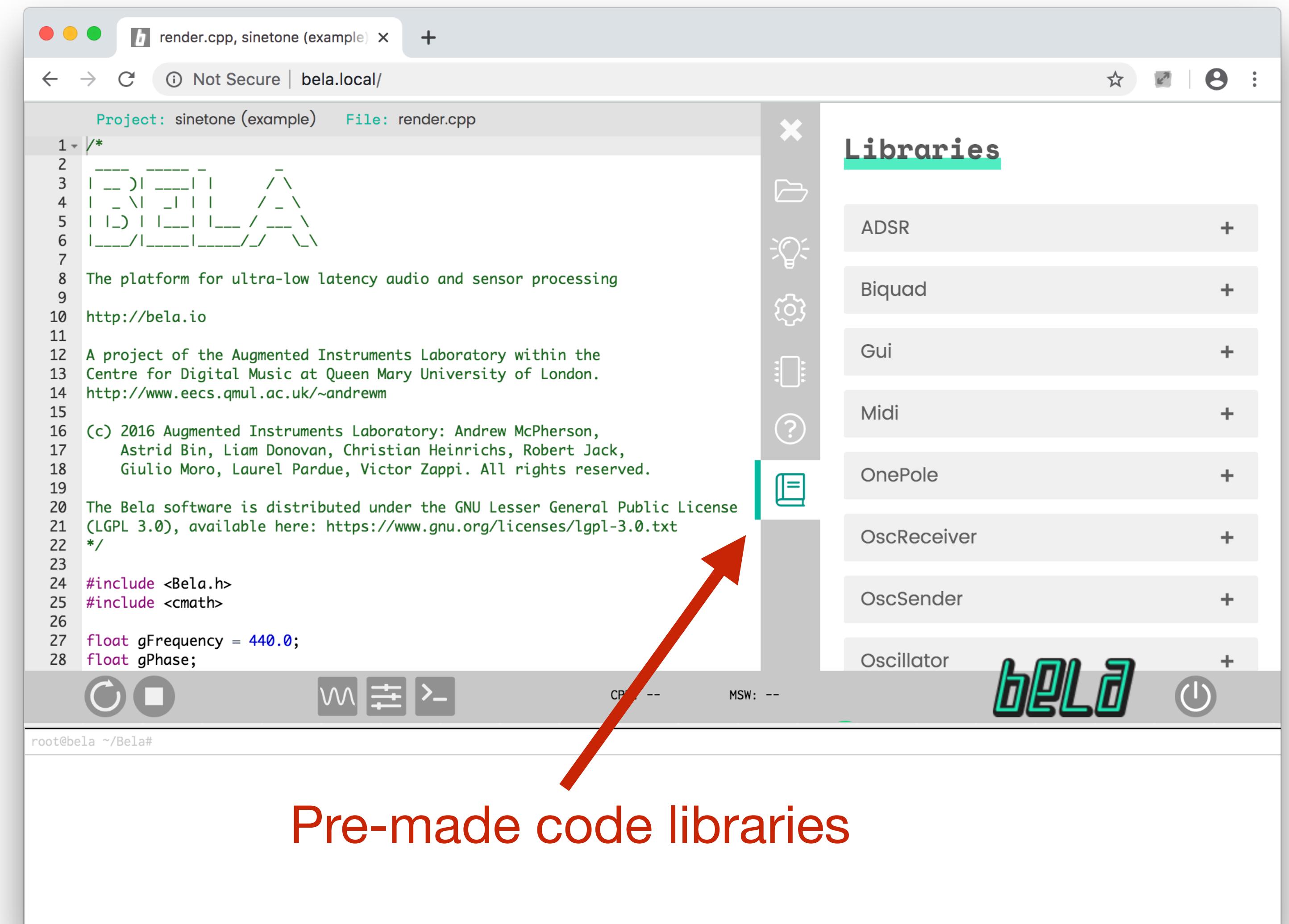
Bring up the Bela IDE:
<http://bela.local>

or sometimes:

<http://192.168.7.2> (Linux)

or

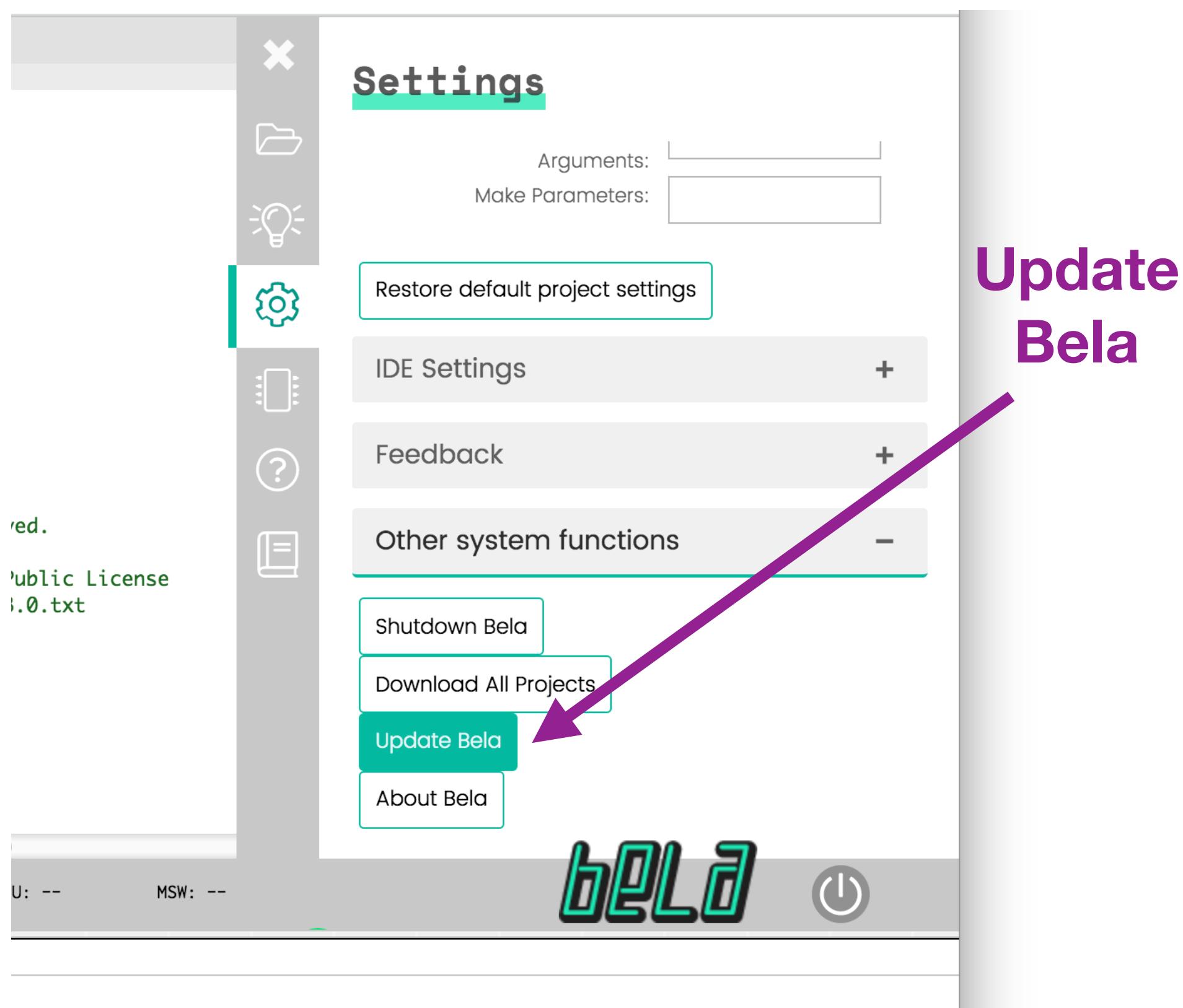
<http://192.168.6.2> (Windows)



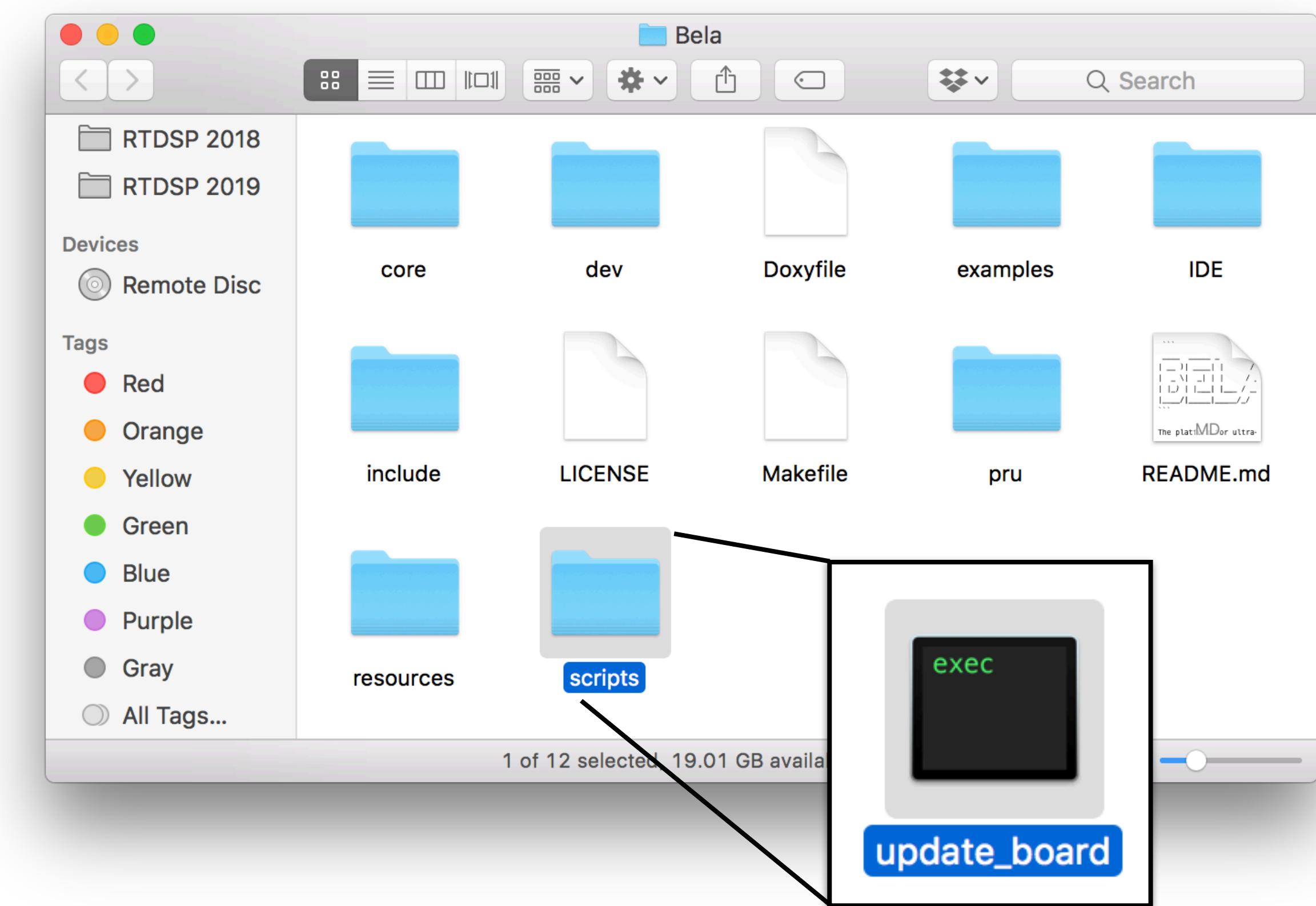
Updating your board

Updating Bela

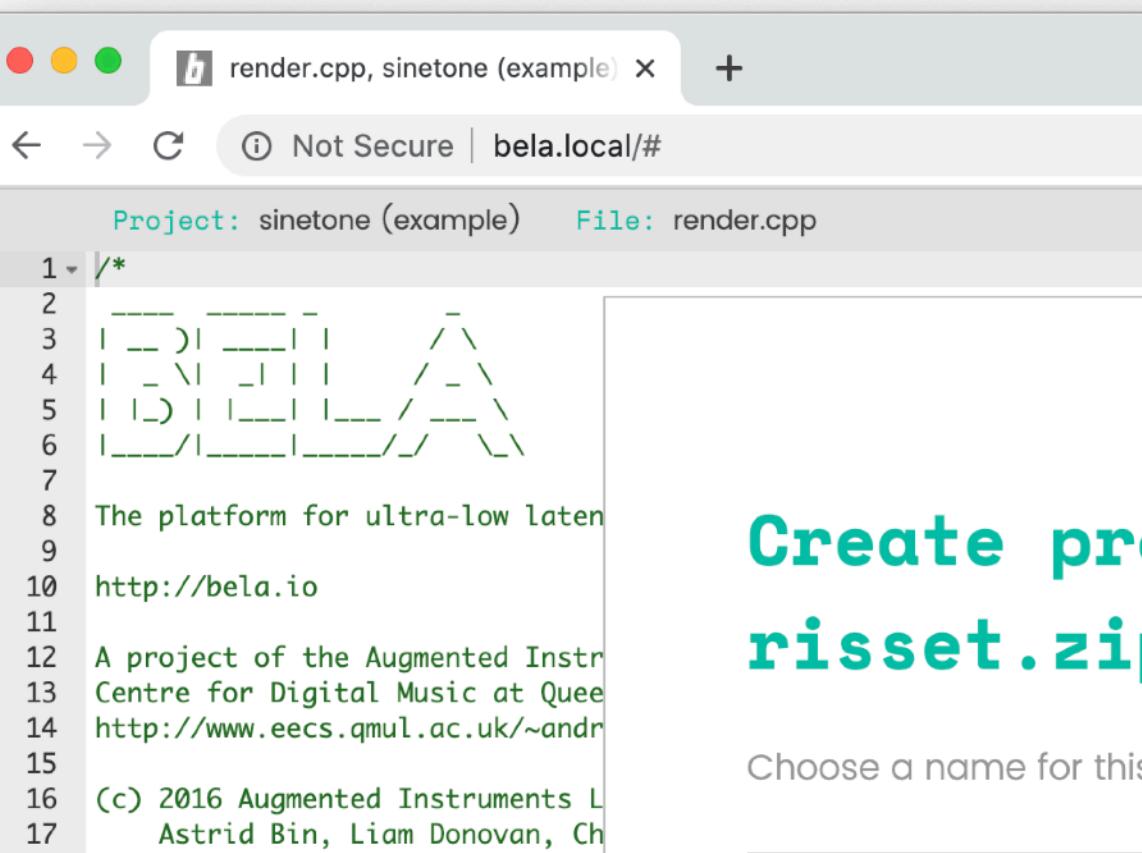
- Download the latest Bela core software from: <http://learn.bela.io/update>
- Two ways to update the board:
 - By IDE (recommended)



By Script (Mac and Linux)
unzip the archive

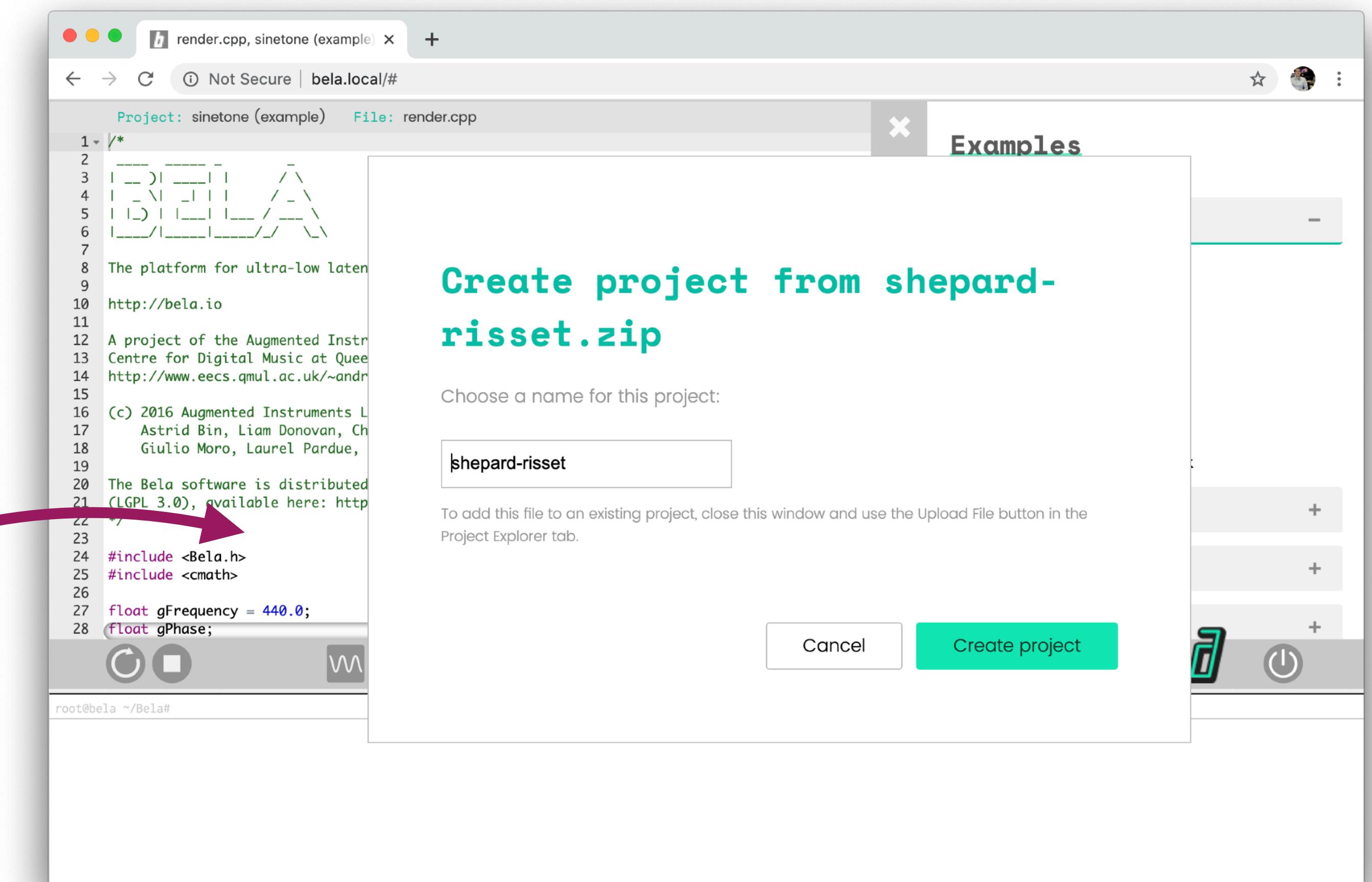


Uploading a new project

1. Download the example [shepard-risset.zip](#) from github.com/BelaPlatform/bela-online-course
 2. Drag the .zip file onto the Bela IDE window
 3. Bela will ask you to name the new project. (*if it doesn't, update to the latest software*)

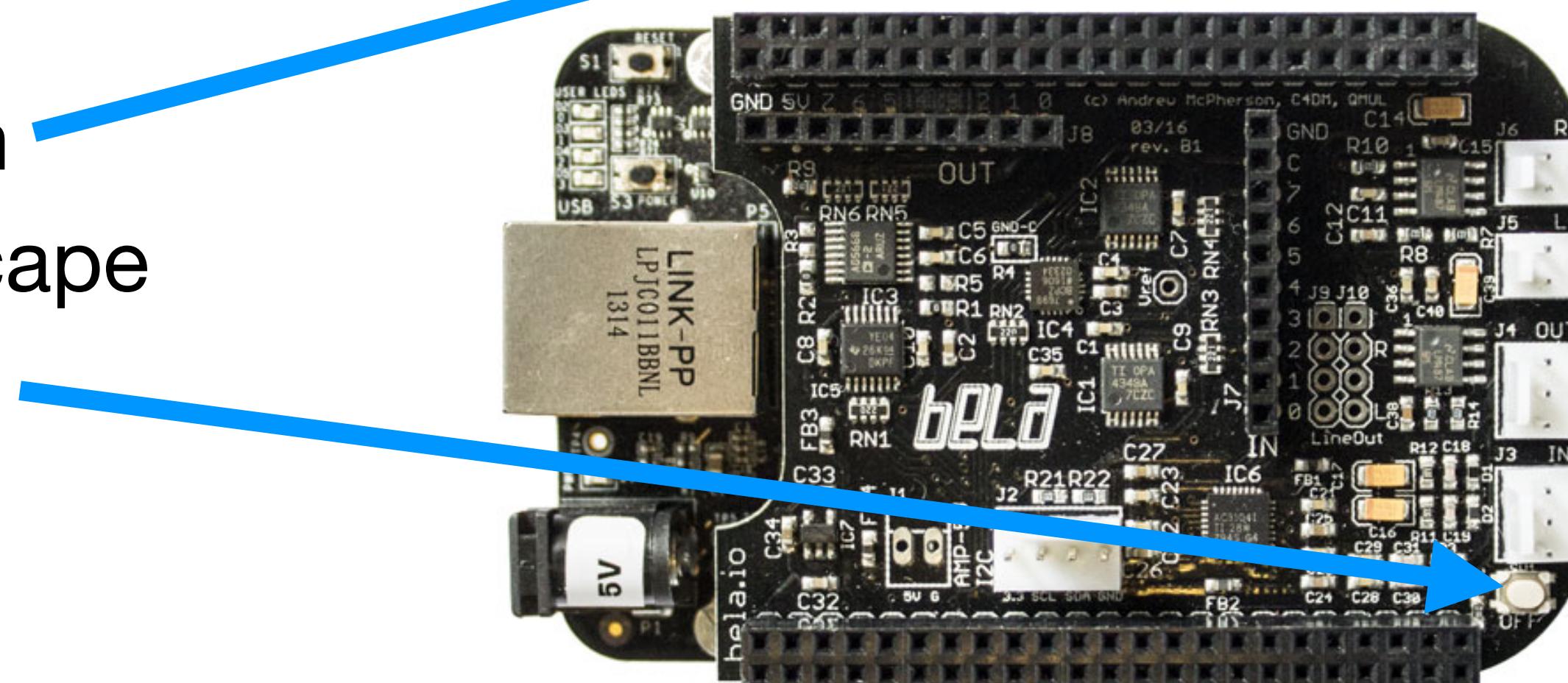
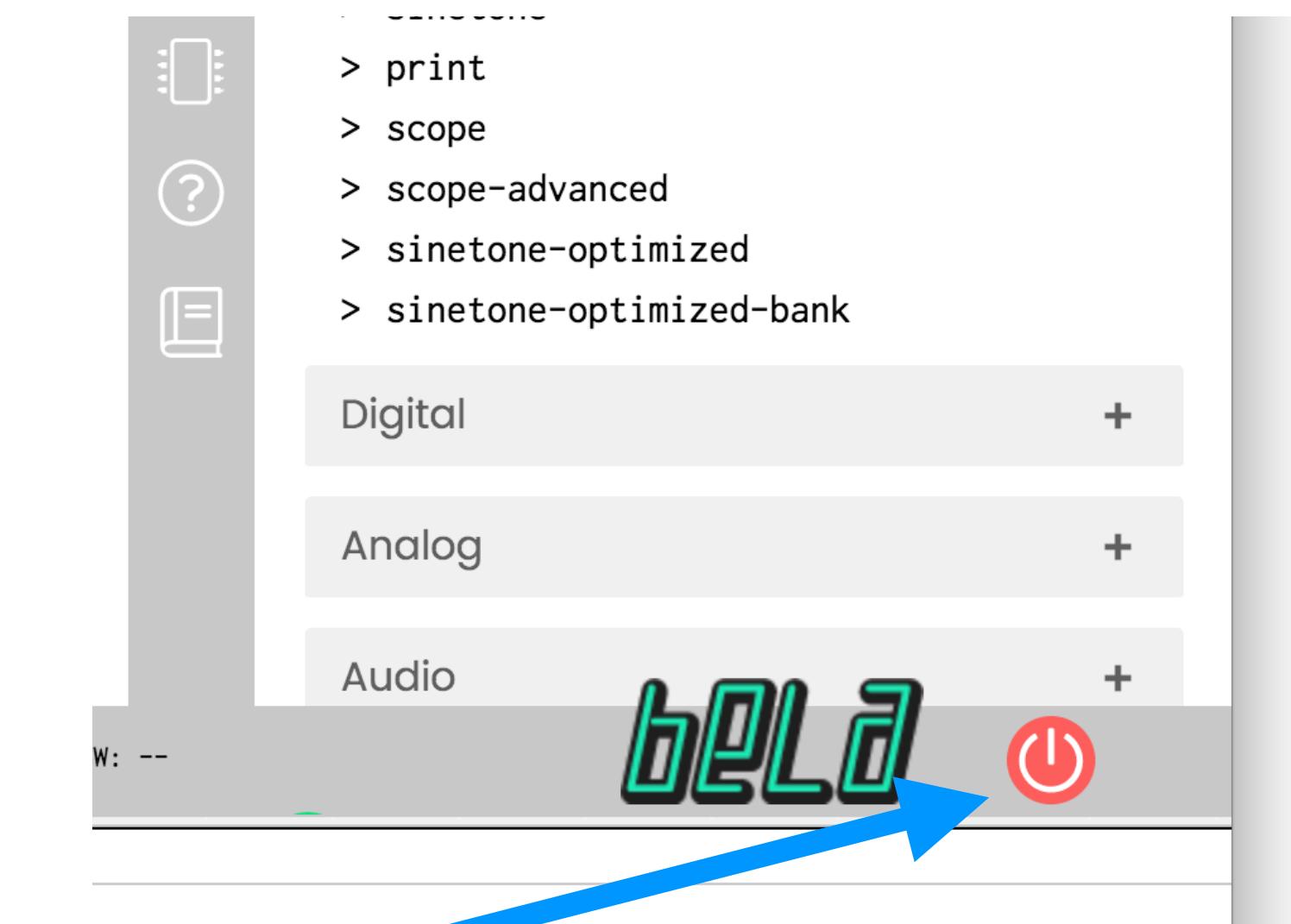
Create proj
risset.zip

Choose a name for this proj



Turning off Bela

- When you're done,
don't just pull the power!
- Bela is a full Linux computer which
needs to be shut down
- Two options:
 - In the IDE, click on the **Shutdown** button
 - Hold the white **OFF** button on the Bela cape
for several seconds



Keep in touch!

Social media:

@BelaPlatform

forum.bela.io

blog.bela.io

More resources and contact info at:

learn.bela.io/resources