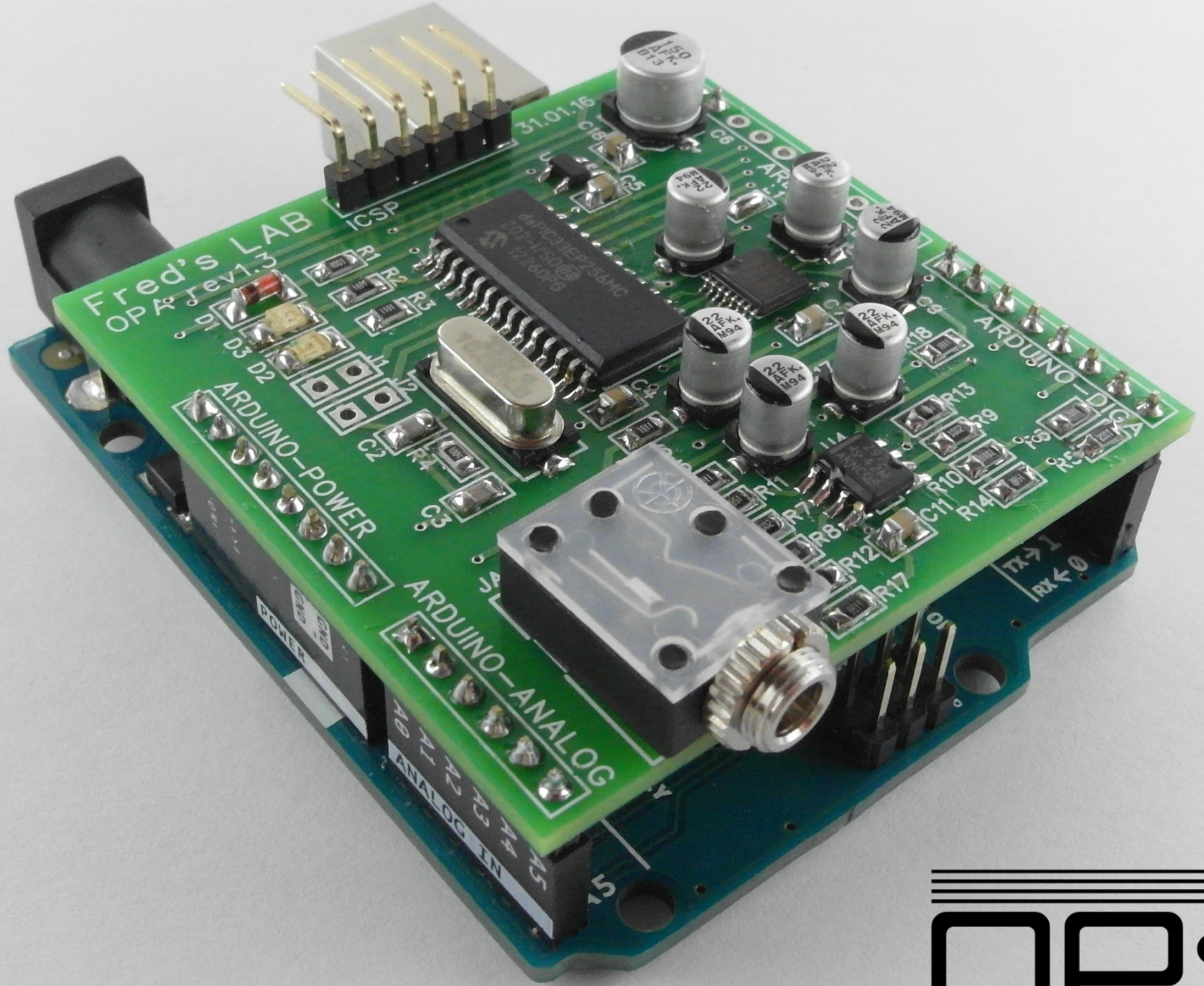


OP·A

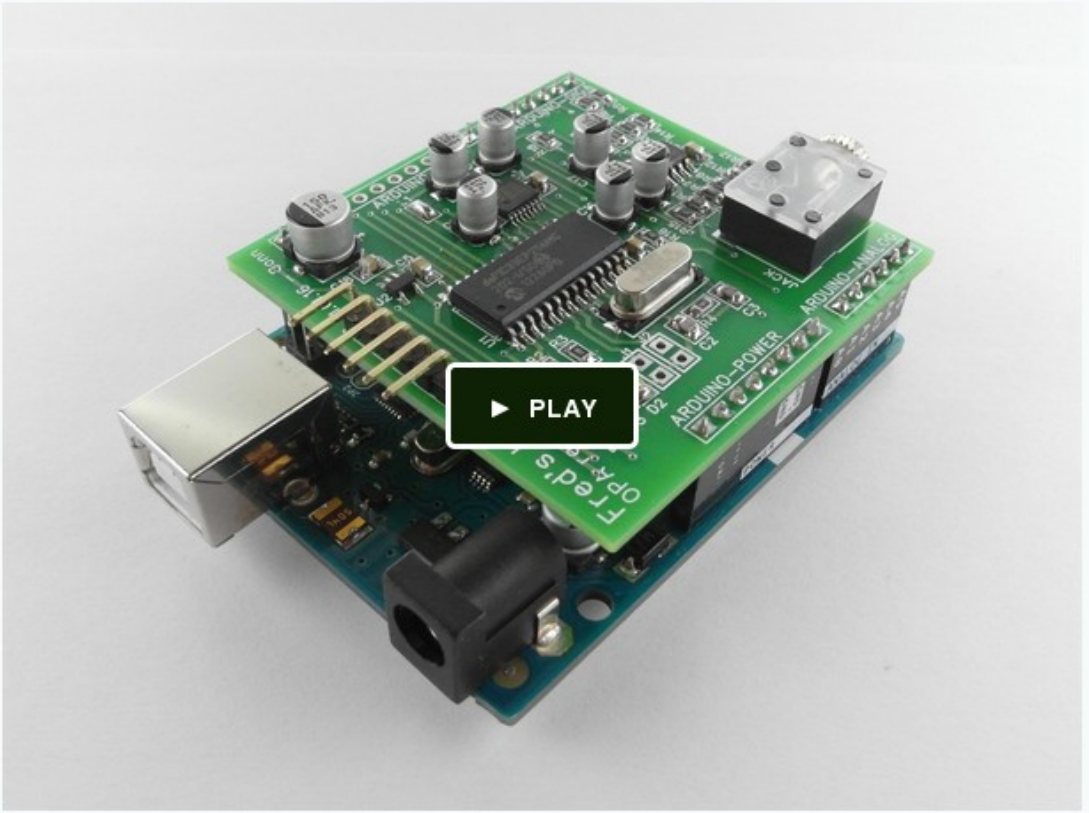
FM SOUND SHIELD

https://www.kickstarter.com/projects/1304489933/opa-multitimbral-fm-synthesizer-shield-for-arduino

KICKSTARTER Discover Start a project About us Search Projects Sign up Log in

OP·A - Multitimbral FM Synthesizer Shield for Arduino

by **Frédéric Meslin**



83
backers

€4,045
pledged of €8,000 goal

17
days to go

Back This Project ★ Remind me

This project will only be funded if at least €8,000 is pledged by Sun, Mar 27 2016 10:59 PM CET.

Bonn, Germany Sound Project We Love

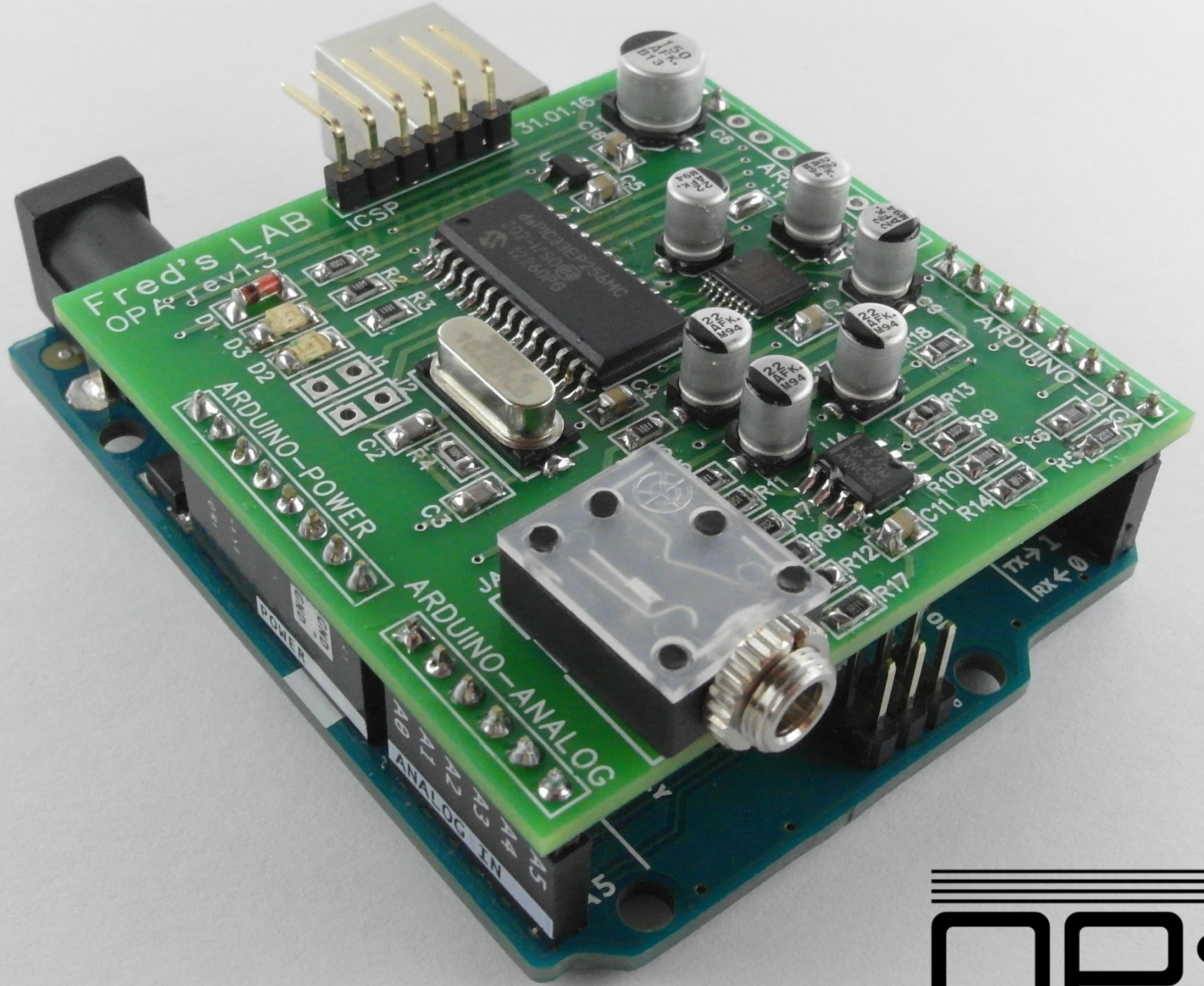
OP·A is a FM synthesizer shield for Arduino. Inspired by retro-consoles, great for chiptune music, custom instruments and art setups.

Frédéric Meslin

First created | **1 backed**
fredslab.net
See full bio Contact me

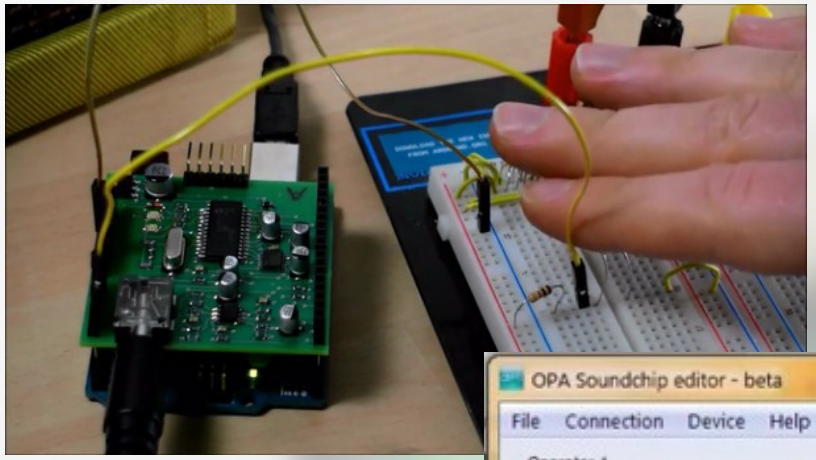
Techniquement

- 4-operator voice structure
- 14 different algorithms
- 10 voice polyphony
- 8 simultaneous instruments
- 90 internal program memory
- 16-bit high-quality stereo output
- Powered with +5V from Arduino
- 3.5mm jack line-level connector



OP·A

FM SOUND SHIELD



OPA Soundchip editor - beta

File Connection Device Help

Operator 1

Coarse Fine Pitch

Absolute

x1.00 0

Attack Decay Sustain Release Tracking

Env. Init LFOSpeed LFOAmount

224

Mute

Operator 2

Coarse Fine Pitch

Absolute

x1.00 0

Attack Decay Sustain Release Tracking

Env. Init LFOSpeed LFOAmount

0

Mute

Operator 3

Coarse Fine Pitch

Absolute

x1.00 0

Attack Decay Sustain Release Tracking

Env. Init LFOSpeed LFOAmount

0

Mute

Operator 4

Coarse Fine Pitch

Absolute

x1.00 0

Attack Decay Sustain Release Tracking

Env. Init LFOSpeed LFOAmount Feedback

0

Mute

Program

1 2 3 4

5 6 7 8

Volume Panning

Voice stealing

Init

File system:

Open ...

Save ...

Internal:

Load

Store

1 Program

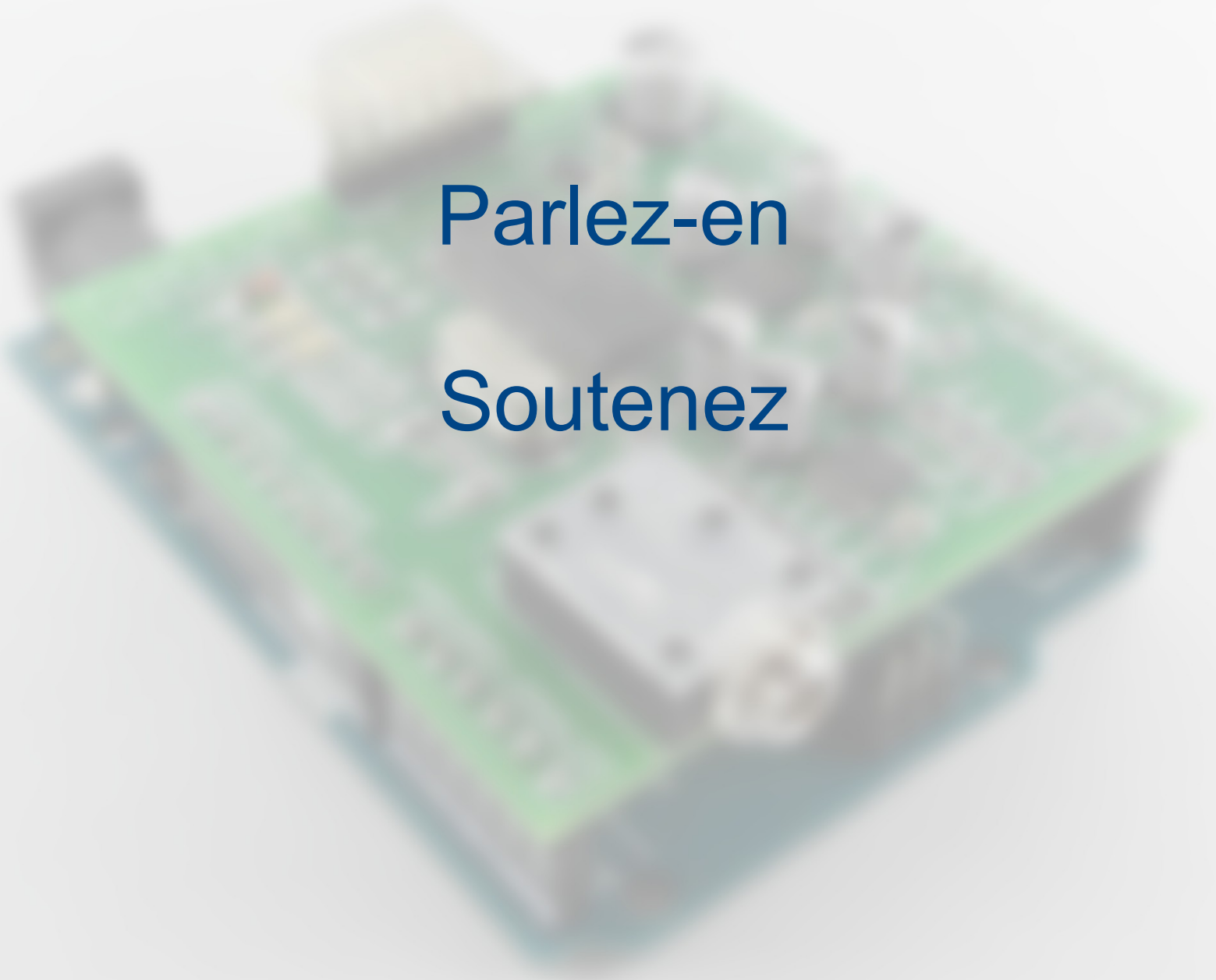
Global

Volume Tune

255 0 semi

Arduino is connected

```
/* OPA Démo HAUM */

#include "melody.h"
#include "OPA.h"

#define DETECT1 A0
#define DETECT2 A1

OPA opa;
unsigned int nb = 0;

void setup() {
  pinMode(OPA_TX, INPUT);
  pinMode(OPA_RX, INPUT);
  pinMode(OPA_SWAP, OUTPUT);
  digitalWrite(OPA_SWAP, 0); // 1 = PC

  pinMode(DETECT1, INPUT);
  pinMode(DETECT2, INPUT);
  digitalWrite(DETECT1, 0);
  digitalWrite(DETECT2, 0);

  Opa.enable();
}

void loop() {
  bool ch1 = !digitalRead(DETECT1);
  bool ch2 = !digitalRead(DETECT2);

  if (ch1 || ch2) {
    ++nb;
    if (nb > melody_length) nb=0;

    if ((melody[nb].chan == 1 && ch1) ||
        (melody[nb].chan == 2 && ch2) ||
        melody[nb].cmd == OPA_CODE_NOTEOFF) {

      if (melody[nb].cmd == OPA_CODE_NOTEOFF)
        opa.noteOff(melody[nb].chan, melody[nb].note);
      else
        opa.noteOn(melody[nb].chan, melody[nb].note);
    }

  } else {
    opa.allSoundsOff();
  }

  delay(melody[nb].delay);
}
```



Parlez-en

Soutenez

Parlez-en

Soutenez

Créez de beaux bugs… sonores