

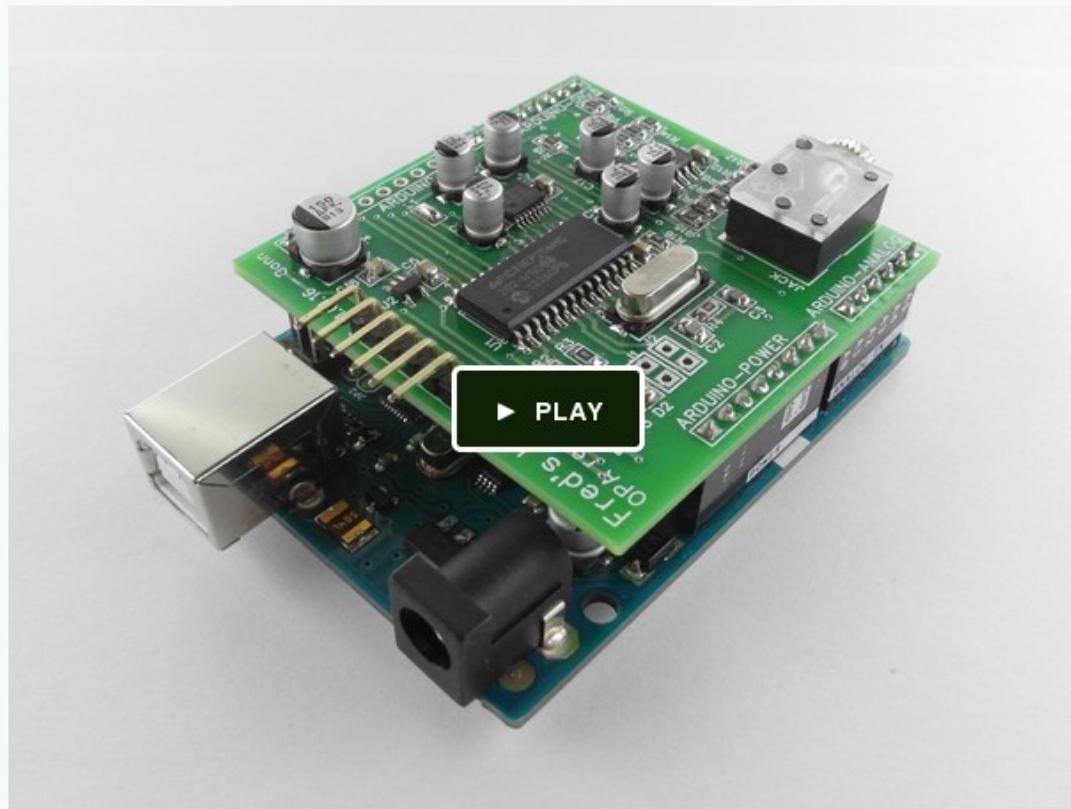
OPA

FM SOUND SHIELD

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

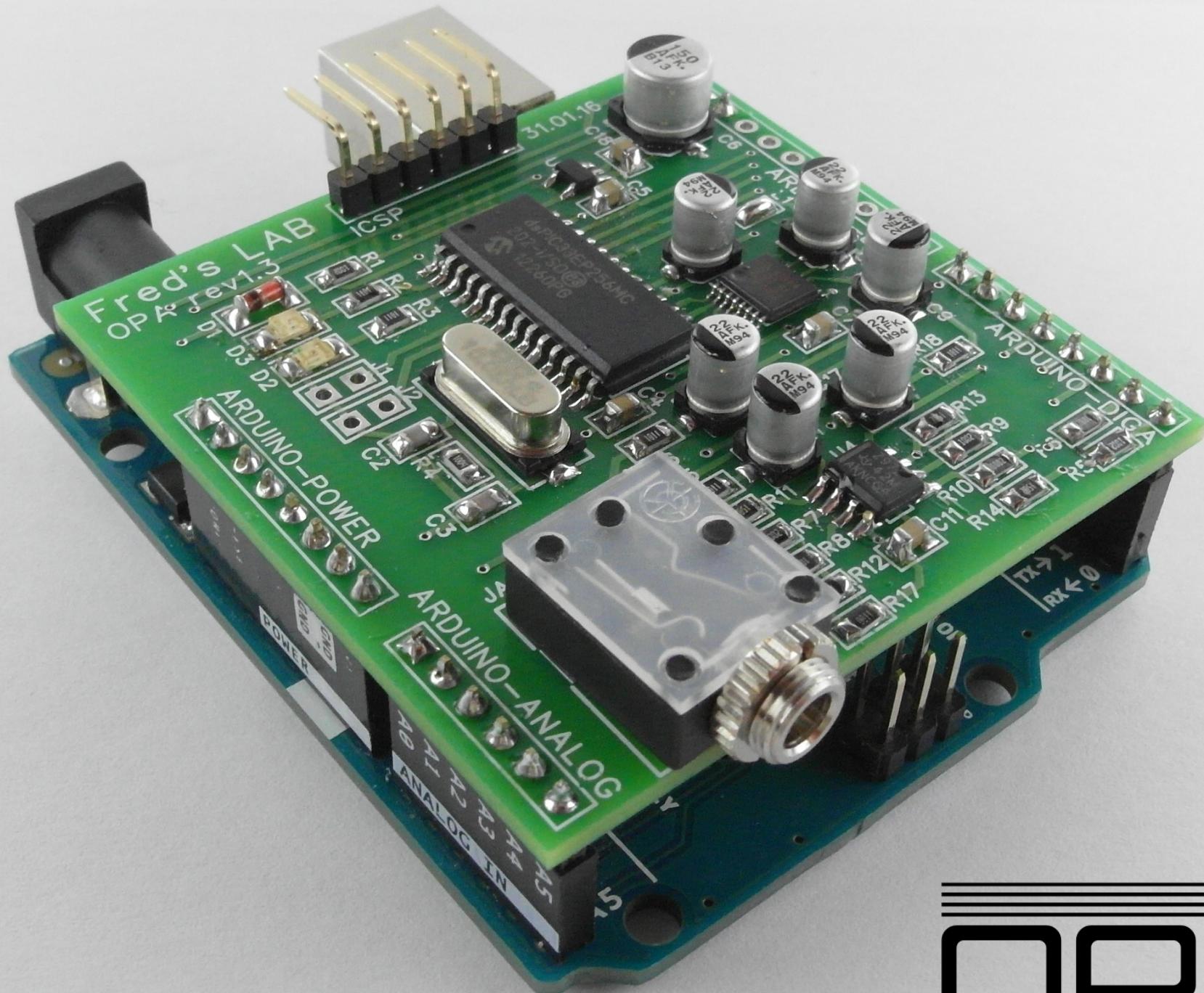
[See full bio](#) [Contact me](#)



Techniquement

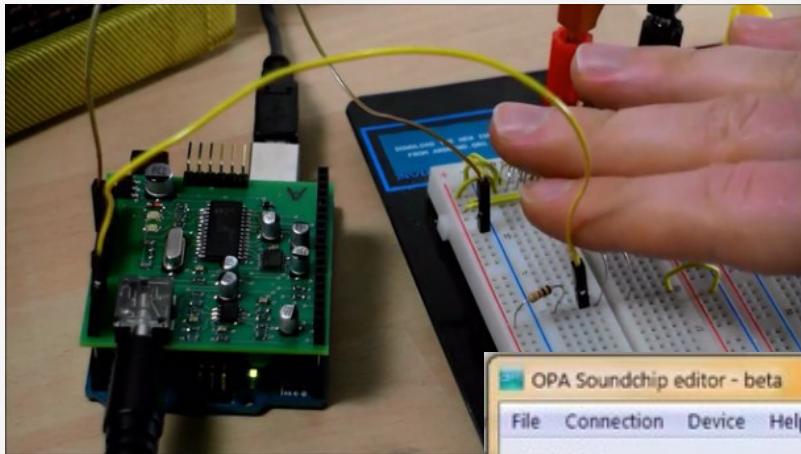
- 4-operator voice structure
- 14 different algorithms
- 10 voice polyphony
- 8 simultaneous instruments
- 90 internal program memory
- 16-bit high-quality stereo output
- Powered with +5V from Arduino
- 3.5mm jack line-level connector





OPA

FM SOUND SHIELD



OPA Soundchip editor - beta

File Connection Device Help

Operator 1

Coarse Fine Pitch
x1.00 0 Absolute

Attack Decay Sustain Release Tracking

Env. Init LFOSpeed LFOAmount

224 Mute

Operator 2

Coarse Fine Pitch
x1.00 0 Absolute

Attack Decay Sustain Release Tracking

Env. Init LFOSpeed LFOAmount

0 Mute

Operator 3

Coarse Fine Pitch
x1.00 0 Absolute

Attack Decay Sustain Release Tracking

Env. Init LFOSpeed LFOAmount

0 Mute

Operator 4

Coarse Fine Pitch
x1.00 0 Absolute

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

This screenshot shows the OPA Soundchip editor software interface. It features four operator sections (Operator 1, Operator 2, Operator 3, Operator 4) each with controls for Coarse and Fine pitch, Attack, Decay, Sustain, Release, Tracking, Env. Init, LFOSpeed, LFOAmount, and a Mute button. Each operator also has a small graphical envelope editor showing a square wave-like curve. To the right of the operators is a vertical stack of buttons for selecting programs (1-8), volume, panning, and voice stealing. Below the operators are buttons for file operations (Open, Save, Load, Store) and global settings (Volume, Tune). At the bottom left, a message indicates "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);

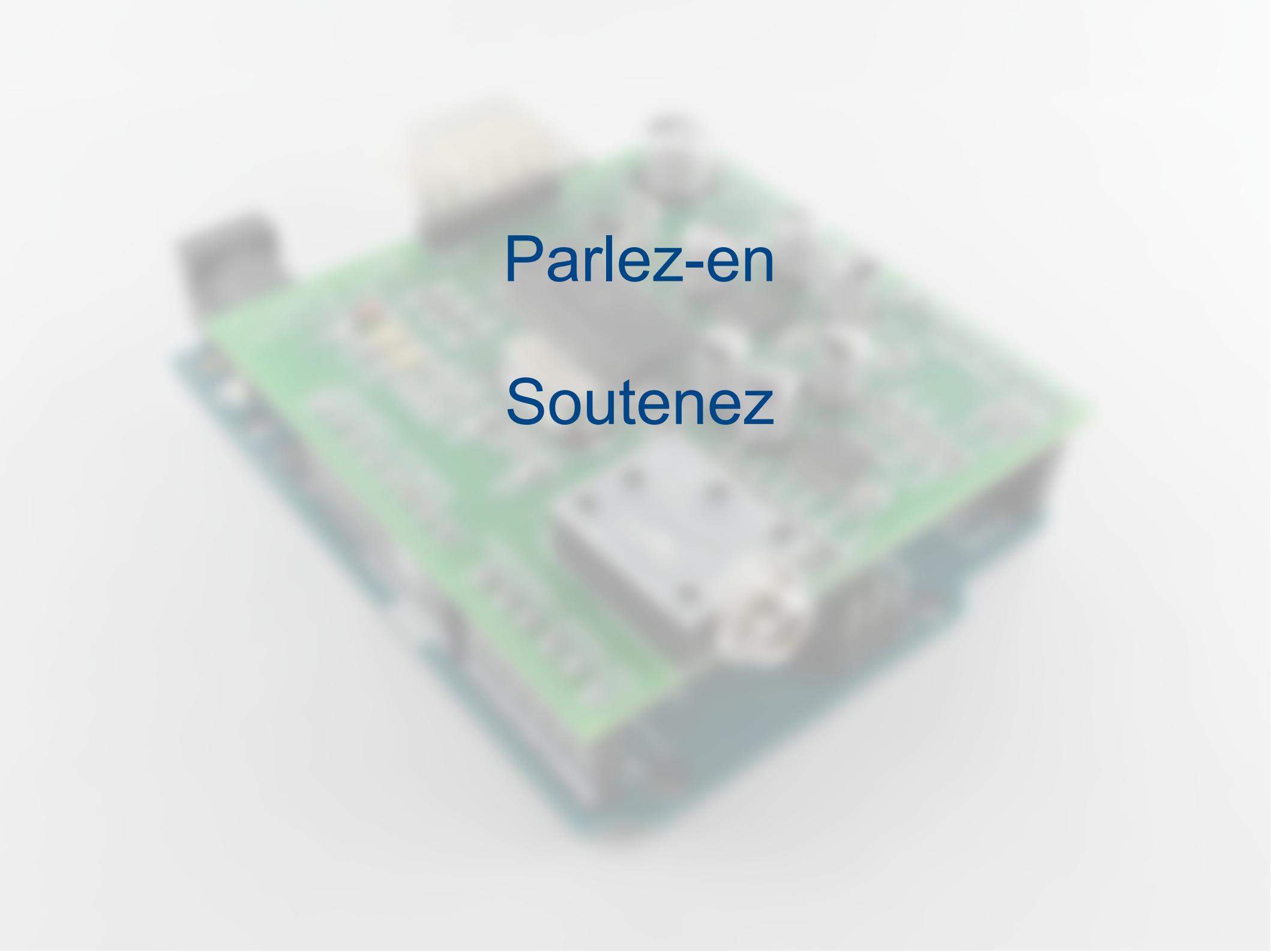
    if (ch1 || ch2) {
        ++nb;
        if (nb > melody_length) nb=0;

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

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

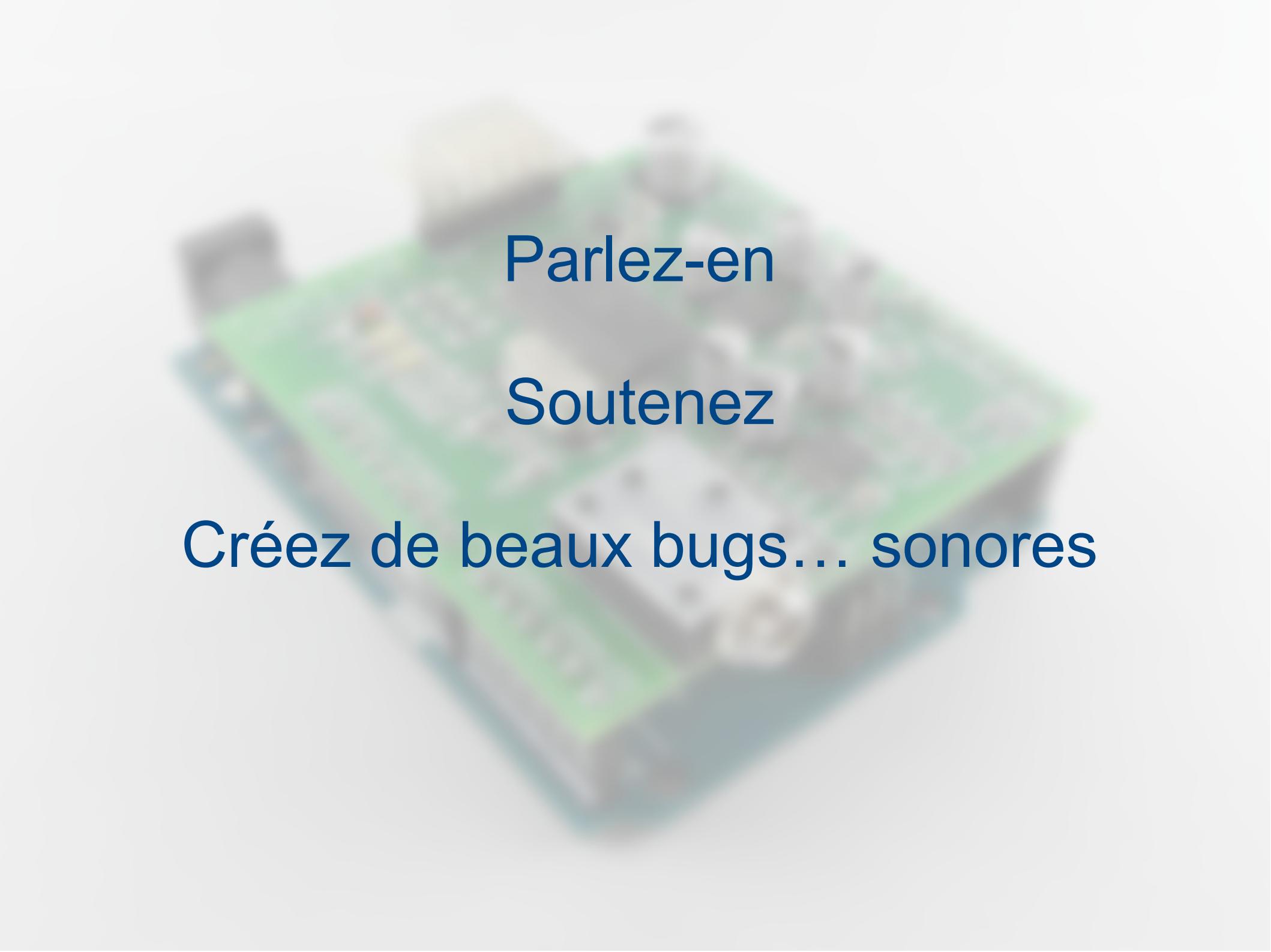
        } else {
            opa.allSoundsOff();
        }

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

A blurry background image of a person holding a green book. The book has a large, dark brown cross on its cover. The person is wearing a light-colored shirt.

Parlez-en

Soutenez



Parlez-en

Soutenez

Créez de beaux bugs... sonores