LBNL-PhotoFlute

CMLS Homework N.3

Group: **Last but not Least**

Professor: **Fabio Antonacci**
Assistant: **Marco Olivieri**

A.Y. 2021/2022

POLITECNICO
MILANO 1863

Contents

1 Introduction

LBNL-PL is a half-hardware, hald-software instrument which can be easily played with just seven photoresistors and an additional sensor. The sound is generated by Supercollider with the help of a GUI in Juce that is able to customize the sound of the software instrument. LBNL-PL has been developed with three powerful software instruments: Arduino, Supercollider and Juce.

2 Data Fetching - Arduino

A simple Arduino sketch fetches data from the seven photoresistor and the other home-made sensor. This data is printed on the serial-port only when the Arduino board program detects changes inside the current state of the circuit, those sparing computational power on both the Arduino and Supercollider sides.

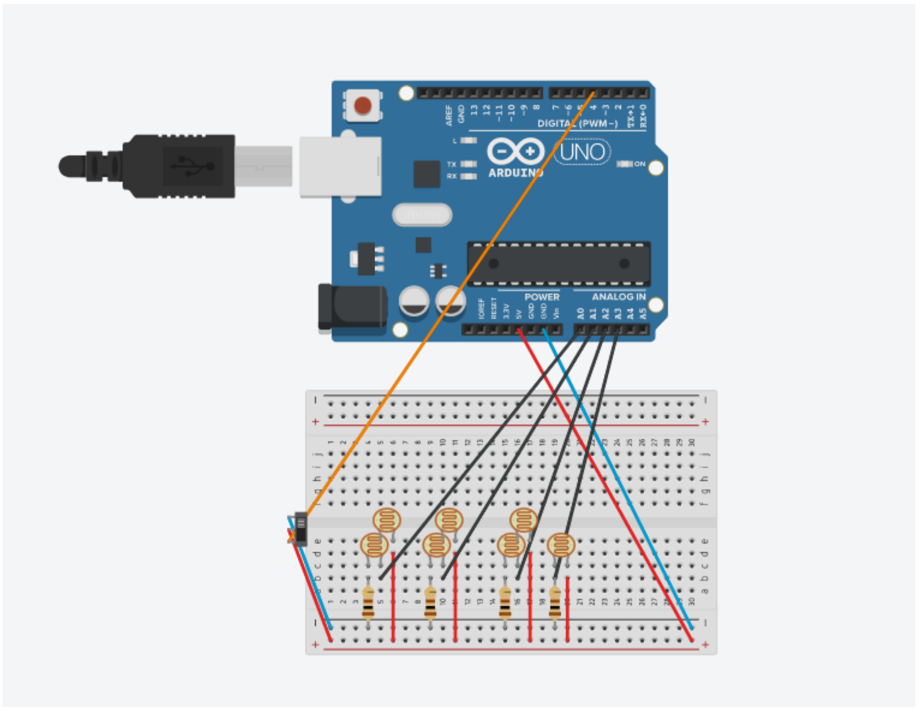

Figure 1: Circuit Scheme

2.1 The Sensors

There are a total of 8 sensors used by LBNL-PF:

- **7 Photoresistors**
- **1 Home-made sensor** constructed with aluminum foil

The photoresistors Each photoresistor controls one note of the flute, with values that are carefully calibrated to obtain a switch-like behaviour (on/off).

The alimun foil sensor The other sensor is used to switch between to distinct modes of the PhotoFlute: one which enables sharp notes and one which enables non-sharp notes.

3 Audio Software Backbone - Supercollider

The heart and mind of the application is a Supercollider file which controls each synthesizer and each parameter which modifies the overall sound.

3.1 The Notes

Each note is a synthesizer of its own which is initialized with a gate value of zero to keep it from playing sounds. Each time the user puts his finger on top of a photoresistor, the gate of the synth which corresponds to that note rises and the sound plays. When the finger is no longer on the photoresistor, the gate is put back to zero and the sound stops.

3.2 Sharp Mode

Sharp Mode (that is, the mode that lets you play sharp notes) is enabled with the aluminum-foil sensor which basically acts as an on/off switch. Let's say we have an array **of length 14**:

$$\text{NotesArray} = [\text{Synths with non-sharp freqs}, \quad \text{Synths with sharp freqs}]$$

What the sensor those is setting an offset variable to the value 7 when the Sharp Mode is on, so that Supercollider accesses the synthesizer with an index offsetted by 7, turning the flute's photoresistors to sharp notes.

3.3 Parameter Personalization

There are many global variables which are controlled via OSC Messages coming from the Juce GUI. These are used to change some aspects of the sound:

- Overall Volume
- Panning
- Synthesizer Type
- Octave of the Notes

There are a total of three Synthesizer types: flute, sax and organ.

4 Graphical User Interface - JUCE

The graphical user interface takes its cue from mobile applications, this makes it very simple and intuitive.

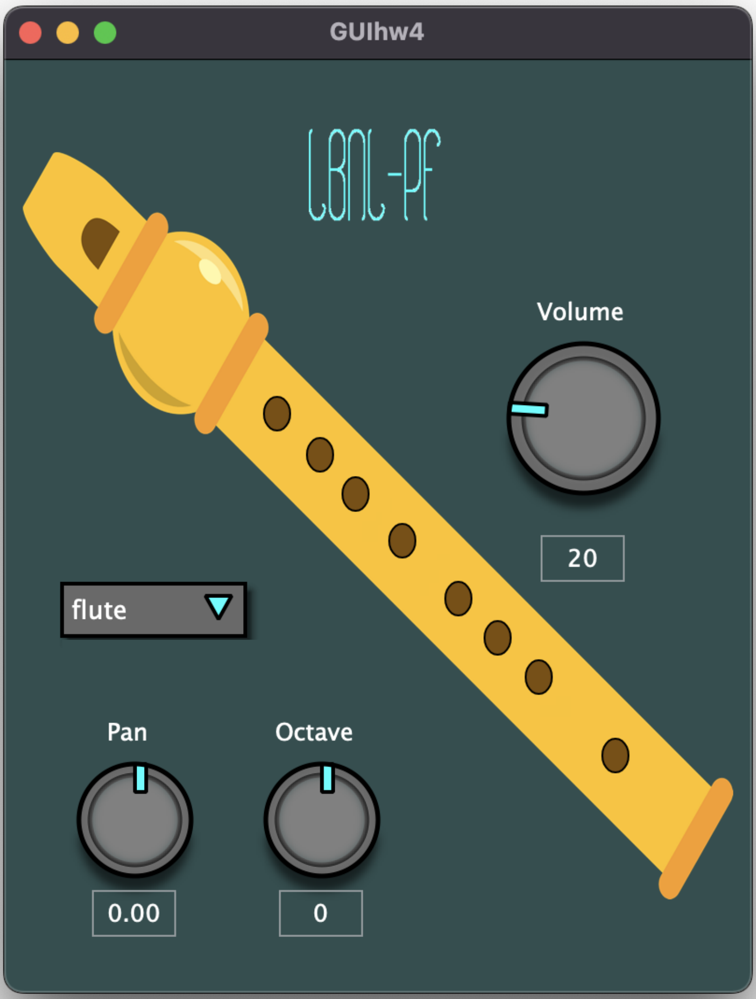

Figure 2: GUI

4.1 Components

The juce components that are inside the GUI are the following.

- `juce::ImageComponent`: for the logo image and the flute one.
- `juce::TextButton`: for the flute's holes.
- `juce::Slider`: for the volume, the pan and the octave knob.

- `juce::Label`: for the knob's title.
- `juce::ComboBox`: for the instrument decision.

4.2 Connection with SuperCollider - sending messages

In order to make the connection between JUCE and SuperCollider working properly it is necessary to create an object `juce::OSCSender` and an other `juce::DatagramSocket`.

Each time a message has to be send the OSCSender object is called with his method `send`. This takes two arguments, the `OSCAddressPattern` and all the arguments that has to be sent.

4.3 Connection with SuperCollider - receiving messages

Regarding the receiving part it is been defined a class `oscMessageReceived` which takes as input the `OSCMessage`. The purpose of this class is to determinate the notes that has to be played and by changing the status of the button it is possible to modify the GUI.

5 How Does It Work

As previously explained, LBNL-PhotoFlute is easy to use.

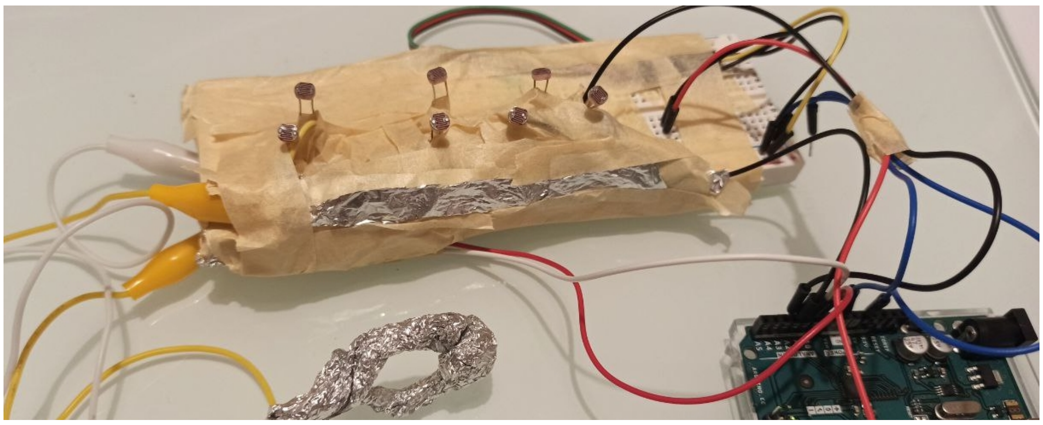

Figure 3: LBNL-PhotoFlute

Here are some general rules to keep in mind:

- Put your fingers on the photoresistors to play notes
- Use the aluminum foil sensor to toggle Sharp Mode
- Use the GUI to customize the sound
- Enjoy LBNL-PL