Politecnico di Milano

Music and Acoustic Engineering

Computer Music - Homework #3

Group: DelayLama - A.Y.: 2021-2022

Synthesthesia

June 1, 2022

Group members:

Ahmed Said
Riccardo Di Bella
Juan Braun
Sofia Parrinelli
Lea Bian

Contents

1 Problem description and introduction

1.1 Synesthesia

The creative development of the project came from the concept of synesthesia.

Synesthesia is a neurological phenomenon where one sensory experience is subconsciously accompanied by another. One of the most common type of synesthesia is chromesthesia, in which sound automatically evokes an experience of color, shape, and movement.

Using the RGB sensor and giving a particular graphical feedback we designed our interpretation of chromesthesia where the color is related to a particular harmonization.

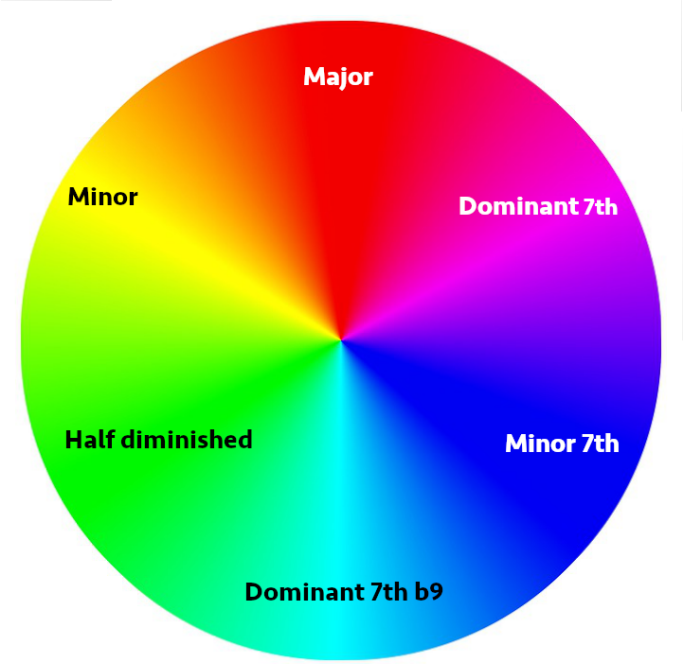

Figure 1: Proposed colour harmonization mapping

Figure 1 shows the proposed mapping between colours and harmonization. The order of the chords was selected in order to create common chords progression with adjacent colours. For example the progression II-V-I can be achieved by starting in the blue colour and moving counter clockwise through the circle.

2 Devices

2.1 RGB Color Sensor

For the color recognition the TCS34725 RGB Color Sensor was used, which measures the light in the red, green and blue wavelengths and an overall clear value, returning four digital integer values between 0 and 65535.

The sensor communicates to the Bela board via I2C communication protocol. This protocol allows bidirectional communication between the Bela and the sensor. This allows to change the working parameters of the sensor to improve the overall usability of the device. In particular, the analog gain of the sensor and the integration time can be changed.

The working parameters that performed the best are:

- Integration time = 154 ms
- Analog gain = 16

The Figure 2 shows the sensor sensitivity for different colors. The sensitivity varies according to the wavelength, reaching its maximum around the red color wavelength.

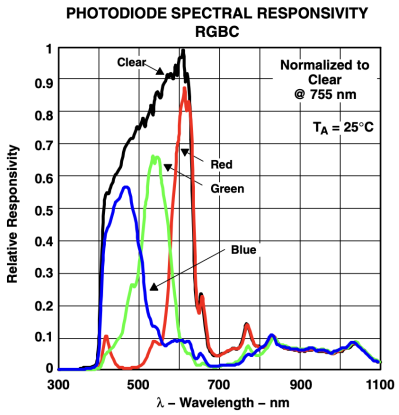

Figure 2: Sensor sensitivity

To implement the communication between the board and the sensor we exploited a version of the Arduino **Adafruit TCS34725** library adapted for the Bela platform.

2.1.1 RGB values normalization

The function `getRGB()` returns the RGB colors, normalized to the range $[0, 255]$. However, the actual bounds of the range depend on many factors (environment lighting, distance, tilting of the sensor...), thus a calibration mechanism was implemented. To avoid a manual calibration phase, the calibration is done automatically by updating the actual minimum and maximum values sensed for the three channels as new readings come in. This requires showing a selection of colors to the sensor at the startup, but it enables the sensor to work in different conditions. However, it also means that the conditions can't change significantly during the execution of the program, as the range values are not re-setted, except when the program is stopped. Thus, for greater stability of the readings, the sensor was also mounted in a fixed position inside a box, where it's not subject to changes in lighting or distance from the sensed object.

2.2 Ultrasonic Sensor

For the pitch control an ultrasonic sensor (HC-SR04) was used. This sensor is a digital sensor that measures the distance of a target object by emitting ultrasonic sound waves and receiving them after they have been reflected by an object.

It has two main pins:

- Trigger pin: by sending a high signal to this pin we tell the sensor to start sending the probing wave. This pin is connected to the digital output of the Bela.
- Echo pin: after receiving the reflected wave, this pin sends a pulse whose width is related to the wave travel time. This pin is connected to the digital input of the Bela.

The Bela library offers a `PulseIn` class, which can be used to simplify the interaction with a distance sensor. By using the `PulseIn.hasPulsed()` function we can obtain the duration of the

pulse in samples. The distance in centimeters can then be computed using the following equation:

$$d = \frac{p}{F_a}\frac{10^6}{58} \tag{1}$$

Where p is the pulse width in samples and F_a is the analog sample rate. The scaling factor $1/58$ is specified by the sensor manufacturer, which also dictates that the width must be passed in microseconds (thus the multiplication by 10^6).

To clean the noise in the readings returned by the distance sensor we employed a simple moving average filter of length 3.

This sensor need to have a power supply of 5V to operate and the digital input to the Bela cannot be larger than 3.3V for this reason the voltage divider shown in Figure 3

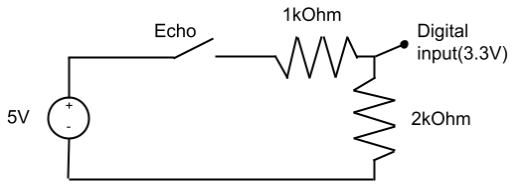

Figure 3: Voltage divider for digital input

When the sensor emits a digital output the Echo switch closes and the digital output becomes:

$$v_{out} = 5V\frac{2k\Omega}{1k\Omega + 2k\Omega} = 3.3V$$

2.3 Connection diagram

Figure 4 shows the connection diagram of the sensors to the Bela

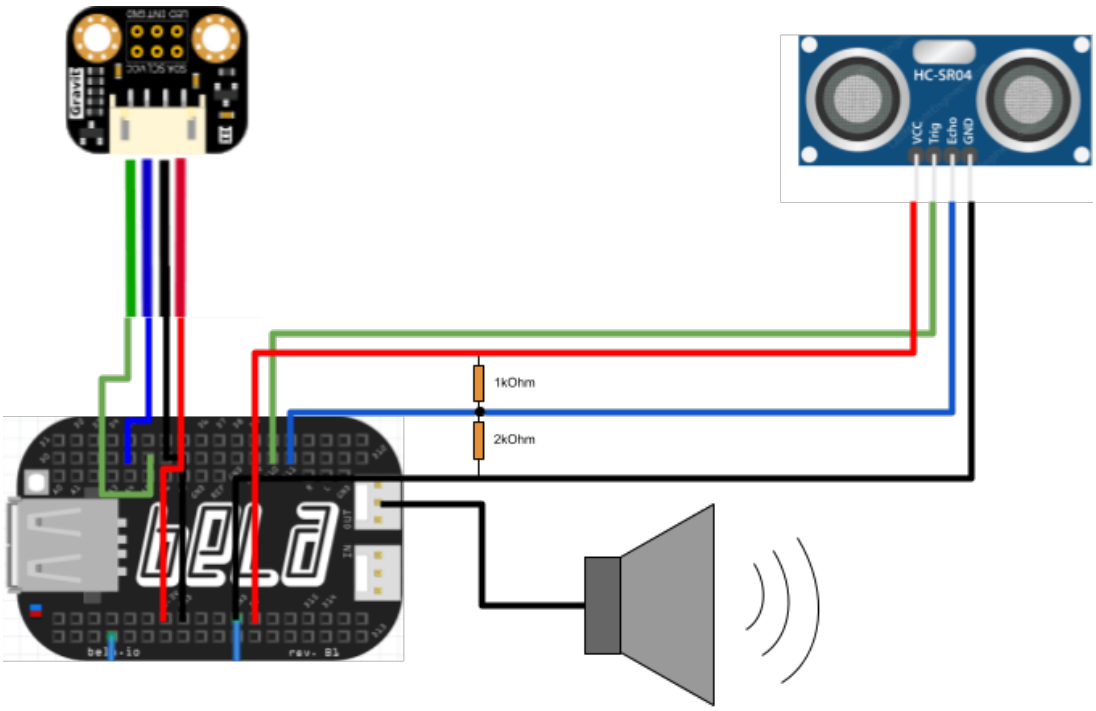

Figure 4: Connection diagram

The mapping of the pins is shown in table 1.

Bela	**Sensor**
SDC	RGB - SDC
SDA	RGB - SDA
3.3V	RGB - Vcc
GND	RGB - GND
D-10	HC-SR04 - Trigger
D-11	HC-SR04 - Echo(after voltage divider)
5V	HC-SR04 - Vcc
GND	HC-SR04 - GND

Table 1: Pin connections

3 Implementation

3.1 Pitch control

The pitch of the fundamental is controlled by the distance measured through the distance sensor. Though the distance sensor is able to measure distances within the $[2, 400]$ centimeters range, we only exploit a shorter range for practical reasons. Thus, the distance range $[2, 100]$ (cm) is mapped to the frequency range $[50, 2000]$ (Hz) in a linear fashion.

3.2 Color Sensor

After receiving the RGB values a type of chord is selected depending on the "distance" between the three values and a fixed threshold which is associated to the color. The three values must fall within a range given by the color threshold and a certain tolerance. For example, we select the color "red" if the following conditions are all satisfied:

$$\begin{aligned} |R - RED\_THRESHOLD_R| &< \epsilon \\ |G - RED\_THRESHOLD_G| &< \epsilon \\ |B - RED\_THRESHOLD_B| &< \epsilon \end{aligned} \tag{2}$$

Where ϵ is the tolerance, which can be defined relative to the full range of the RGB values. The thresholds and the tolerance used in the program were chosen through experimentation, and are listed in table 2.

	R	G	B
RED	255	0	0
GREEN	60	255	60
BLUE	0	55	255
YELLOW	150	150	40
PURPLE	120	50	150
TEAL	50	200	130

Table 2: Color thresholds

The chosen tolerance value is:

$$\epsilon = \frac{255}{4} \tag{3}$$

The association between the colors and the chords is as follows:

- Red: Major chord
- Yellow: Minor chord
- Green: Half diminished chord
- Teal: Dominant 7th b9 chord
- Blue: Minor 7th chord
- Purple: Dominant 7th

3.3 Synthesizer

To generate the output sound we used the Oscillator class provided by Bela. In particular, we sum the output from 4 separate oscillators, each one producing a triangle waveform.

This sound is then fed into a digital bandpass biquad filter, whose center frequency is controlled by RGB values. In particular, the center frequency sweeps from 80 Hz to 800 Hz, following this equation:

$$f_c = \frac{(B \times 80 + G \times 200 + R \times 800)}{R + G + B} \tag{4}$$

This results in a darker sound for colors closer to blue and a brighter sound for colors closer to red.

3.4 Integration with Processing

A one-way communication from Bela to Processing was established to give a visual feedback to the user and complete the synesthetic experience.

The communication is realized through OSC messages. The addresses of the variables sent in this way are:

- `/colours/[R,G,B]`
- `/distance/d`
- `/chord/c`

The Processing script displays three spheres with varying sizes, orbiting around a centered sphere of fixed size. The orbiting spheres correspond to the three RGB channels and their sizes are related to these values. The sphere at the center displays the "overall" sensed color. The speed of the orbiting spheres is related to the distance measured by the sensor, and thus indirectly to the frequency of the fundamental. A text on bottom, finally, displays the current type of chord played.

4 Device

As a proof of concept we present the device enclosed in a box, to give it a more appealing design and hide the circuitry inside. On the top of the box we have the *color turntable*, i.e. a wheel depicting a color gradient which can be spun by the user to select the type of chord and change the center frequency of the bandpass filter. On one side we have an opening for the distance sensor. By moving an object (preferably a wide flat surface) in front of the sensor we are able to detect the distance of the object and alter the pitch of the played chord accordingly.

Figure 5 shows the proposed enclosure

Figure 5: Proposed enclosure for the device.