Computer Music – Languages and Systems
HW3

DESIGN AND IMPLEMENTATION OF A COMPUTER MUSIC SYSTEM

Emma Coletta 214391
Federico Ferreri 221557
Lorenzo Previati 222058

PURPOSE OF THE FINAL SYSTEM:

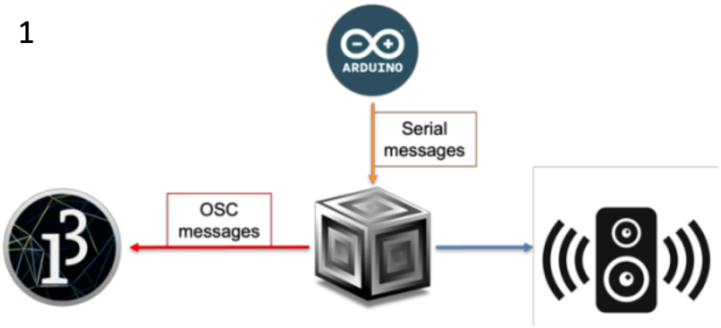
frequency modulation of a synthesizer controlled by a gray scale photosensitive sensor.

Needed tools:

- interaction unit system;
- computer music unit;
- graphical feedback unit.

For the implementation of our music system, we chose to use the first among the suggested possible schemes, involving the Arduino Uno (as our *interaction music system*), SuperCollider (as our *computer music unit*) and the exchange of OSC messages.

1



Using Arduino UNO as our control device, we share information to SuperCollider with serial messages providing the feedback events to Processing (our *graphical feedback unit*) with OSC messages.

SYSTEM DESCRIPTION

Tools:

- digital push button
- analog gray scale sensor
- Arduino UNO board

Connected to the Arduino, the digital push button has the simple task to activate or to stop our synthesizer from playing. If the value passed on to Supercollider is 0 the synthesizer is refrained from playing, otherwise the program then works on frequency modulation.

The gray scale sensor for Arduino is a composition of a photocell and an integrated white LED on board. We exploited its ability to measure the intensity of light from black to white to decide on which frequency our synthesizer would play. So, depending on the amount of light caught by the sensor, frequency modulation happens, resulting on the change of pitch of the playing synth.

Arduino board settings:

- digital push button: charged at 3.3V;
- analog gray scale sensor: charged at 5V.

Value range caught by the gray scale sensor: from 0 (when deactivated) to 1024 (max. light condition).

Frequency range for the frequency modulation: 440Hz to 880Hz.

Notes on SuperCollider code

We open the communication with the Serial Port of the computer to which the Arduino board is connected.

With the help of a routine, we store in an array the values caught by the sensor (luminosity condition) transmitted by the Arduino.

We then proceeded to define a Synth object, initialized at a frequency = 0.
As our last step, we created a control routine that sets the frequency for the playing synth.

The following portion of code sets the frequency for the playing synth, mapping between the range of values caught by the light sensor (from 0 to 1024) and a set value for the frequency of a note from central A to A one octave above it.

```
var noteFunction = { arg x; var y;
    y= case
    {x>0 && x<78.69}{440}
    {x>78.69 && x<157.38}{466.16}
    {x>157.38 && x<236.07}{493.88}
    {x>236.07 && x<314.76}{523.25}
    {x>314.76 && x<393.45}{554.37}
    {x>393.45 && x<462.14}{587.33}
    {x>462.14 && x<550.83}{622.25}
    {x>550.83 && x<629.52}{659.25}
    {x>629.52 && x<708.21}{698.46}
    {x>708.21 && x<786.9}{739.99}
    {x>786.9 && x<865.59}{783.99}
    {x>865.59 && x<944.28}{830.61}
    {x>944.28 && x<1024}{880};
};
```