

# CONVERSATION BENCH

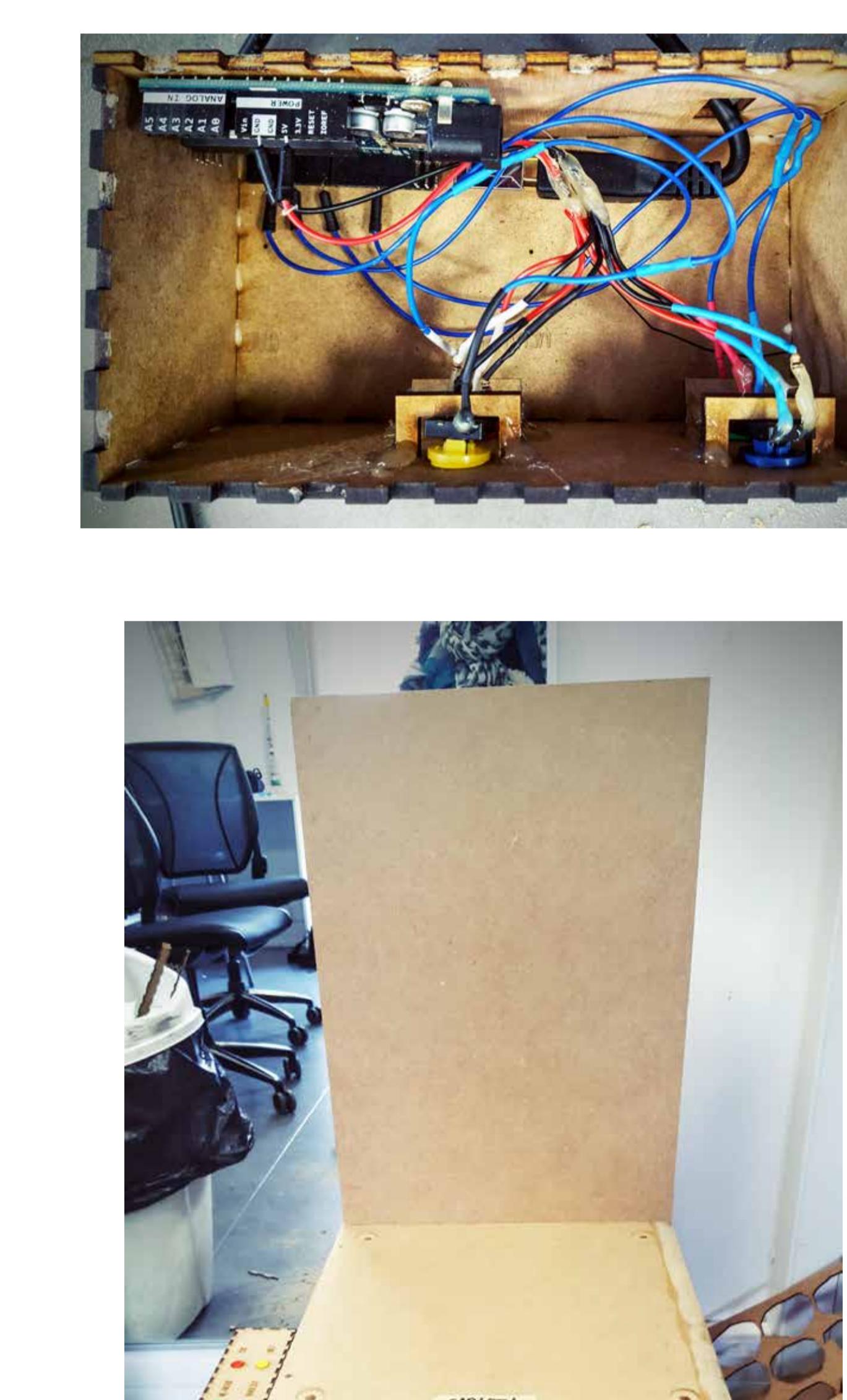
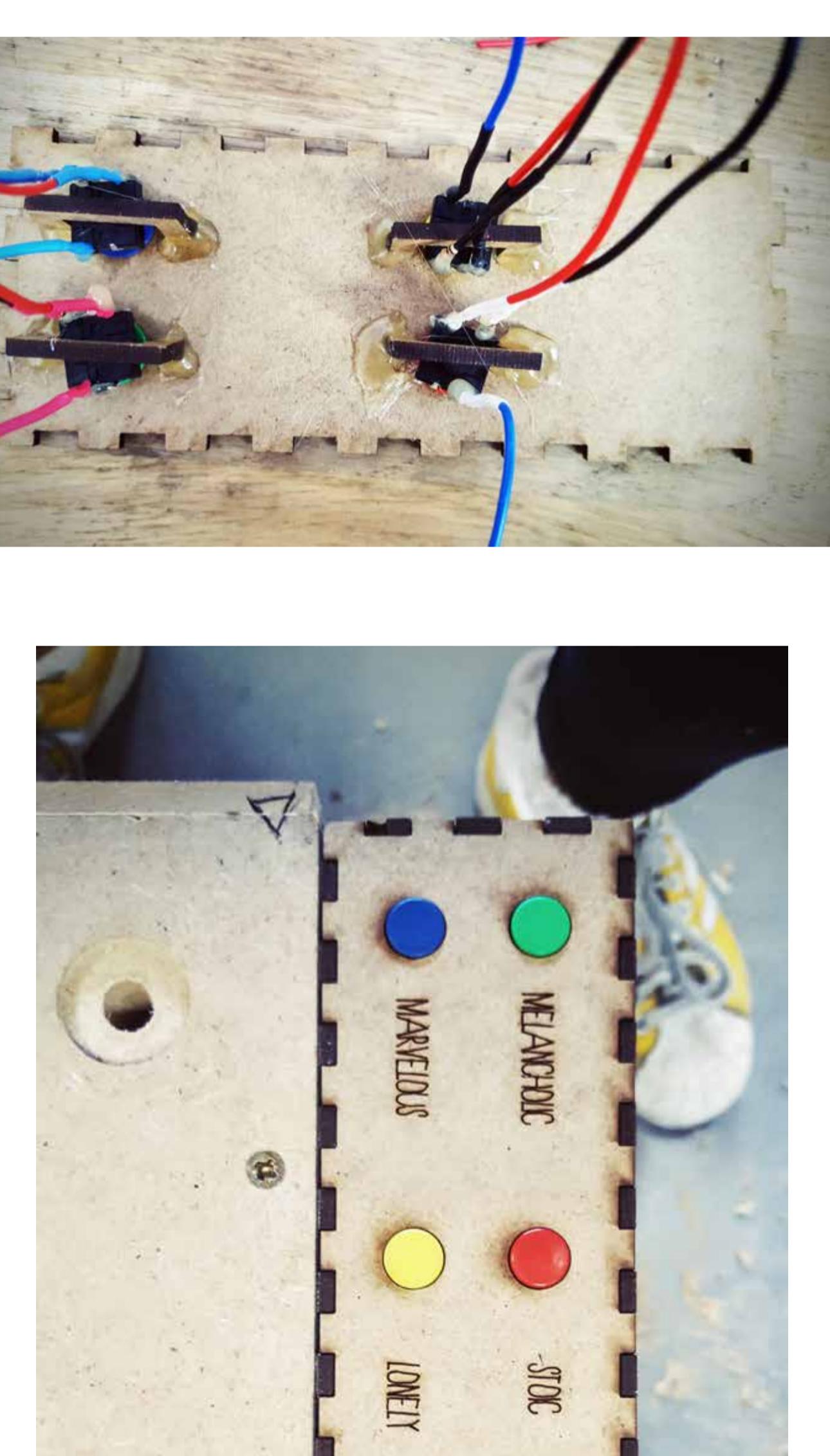


IF YOU FEEL LONELY MELANCHOLIC STOIC OR MARVELOUS

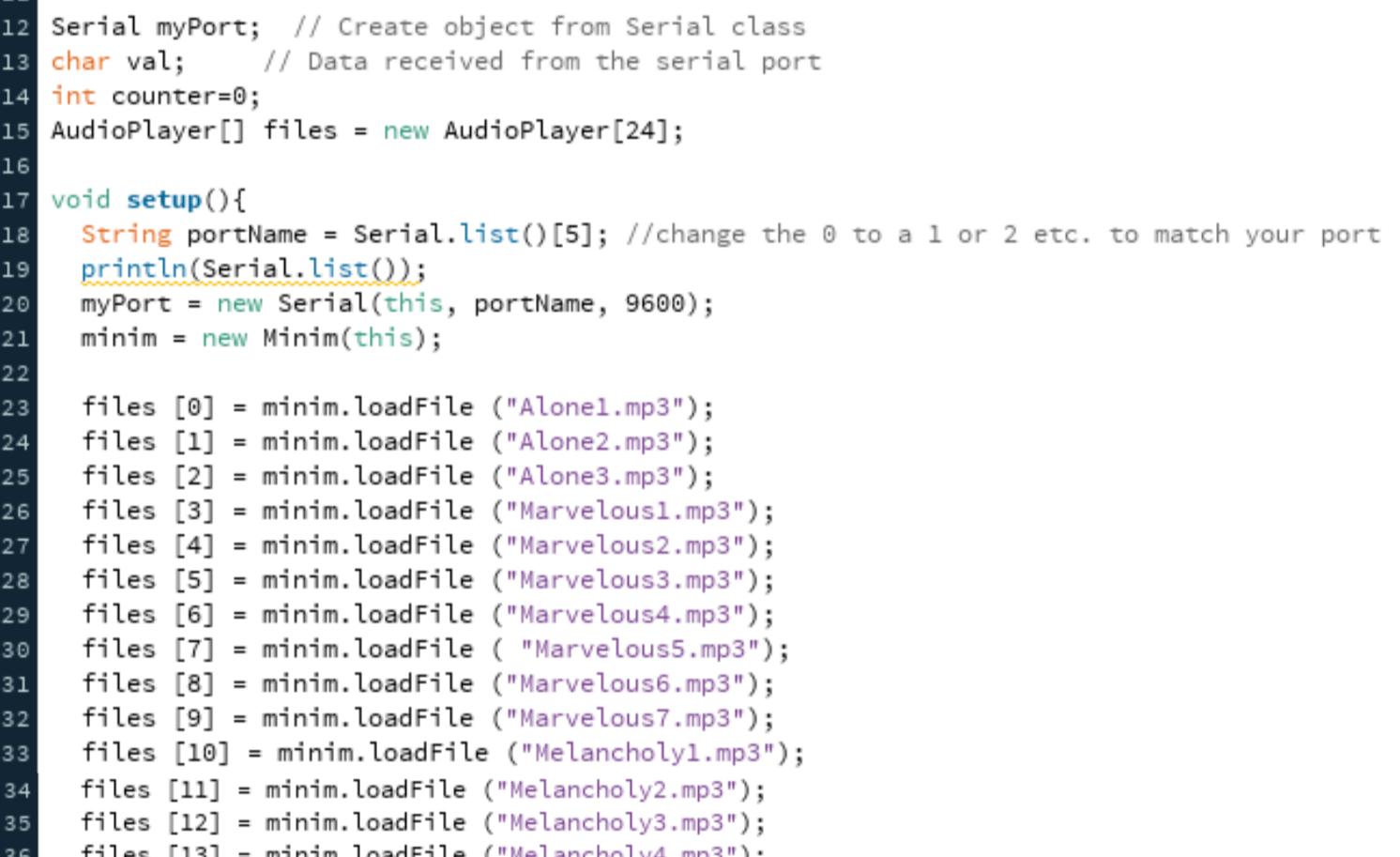
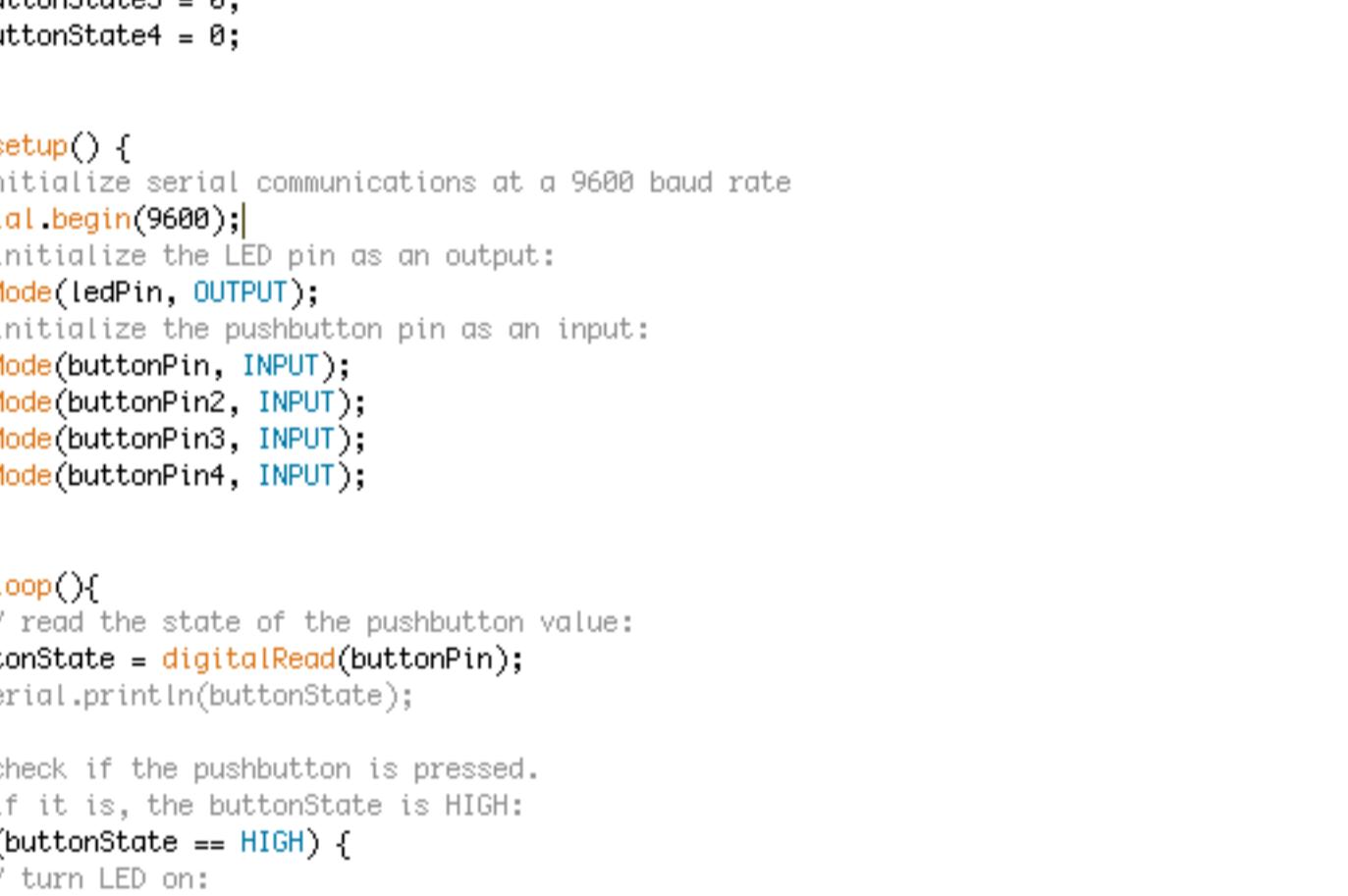
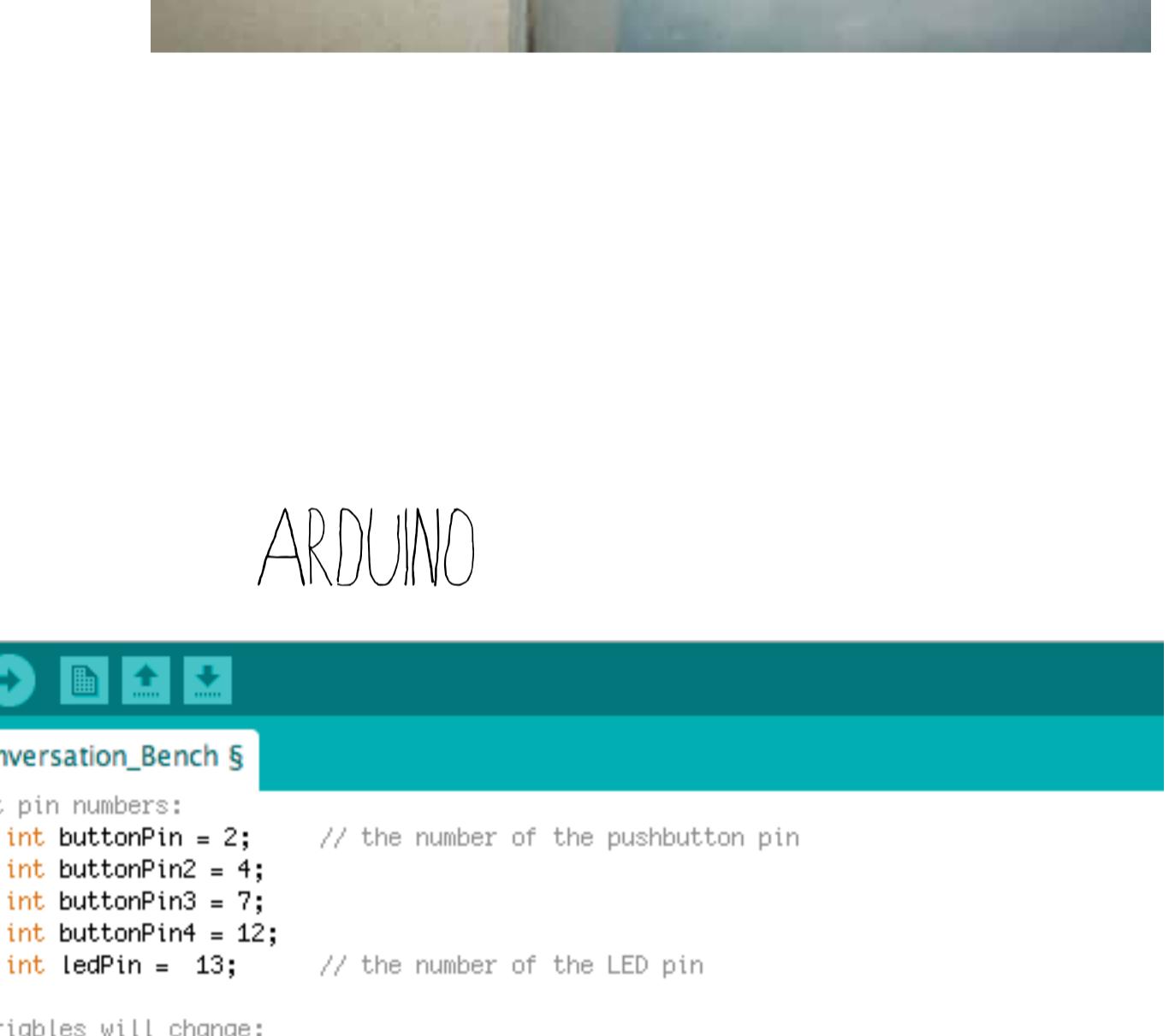
AND YOU ARE SITTING ON A BENCH

BY YOURSELF...

WHY NOT HAVE THE LATTER TALKING



## PROCESS



## CODE

### ARDUINO

```
// set pin numbers:
const int buttonPin = 2; // the number of the pushbutton pin
const int buttonPin2 = 4;
const int buttonPin3 = 7;
const int buttonPin4 = 12;
const int ledPin = 13; // the number of the LED pin

// variables will change:
int buttonState = 0; // variable for reading the pushbutton status
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;

void setup() {
  // initialize serial communications at a 9600 baud rate
  Serial.begin(9600);
  // initialize the led pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
  pinMode(buttonPin4, INPUT);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  // Serial.println(buttonState);

  // check if the pushbutton is pressed.
  // If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    Serial.println("1"); // when processing receives Serial Print, 1 = button pressed
    delay (100);

  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    Serial.println("0");
    delay (100);
  }

  /////////////////Second Button///////////////
  buttonState2 = digitalRead (buttonPin2);

  if (buttonState2 == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    Serial.println("2");
    delay (100);

  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    Serial.println("3");
    delay (100);
  }

  /////////////////Third Button/////////////
  buttonState3 = digitalRead (buttonPin3);

  if (buttonState3 == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    Serial.println("4");
    delay (100);

  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    Serial.println("5");
    delay (100);
  }

  /////////////////Fourth Button/////////////
  buttonState4 = digitalRead (buttonPin4);

  if (buttonState4 == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    Serial.println("6");
    delay (100);

  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    Serial.println("7");
    delay (100);

  }

  //send "Hello, world" over the serial port
  //Serial.println("Hello, world!");
  //wait 100 milliseconds so we don't drive ourselves crazy
  //delay(100);
}
```

### PROCESSING

```
import dfdf.minim.*;
import dfdf.minim.analysis.*;
import dfdf.minim.signals.*;
import dfdf.minim.spi.*;
import dfdf.minim.ugens.*;

import processing.serial.*;

Minim minifm;
Minim myPort; // Create object from Serial class
char val; // Create object from the serial port
int counter=0;
AudioPlayer[] files = new AudioPlayer[24];

void setup(){
  String portName = Serial.list()[5]; //change the 0 to a 1 or 2 etc. to match your port
  myPort = new Serial(this, portName, 9600);
  minifm = new Minim(myPort);

  files [0] = minifm.loadFile ("Alone1.mp3");
  files [1] = minifm.loadFile ("Alone2.mp3");
  files [2] = minifm.loadFile ("Alone3.mp3");
  files [3] = minifm.loadFile ("Marvelous1.mp3");
  files [4] = minifm.loadFile ("Marvelous2.mp3");
  files [5] = minifm.loadFile ("Marvelous3.mp3");
  files [6] = minifm.loadFile ("Marvelous4.mp3");
  files [7] = minifm.loadFile ("Marvelous5.mp3");
  files [8] = minifm.loadFile ("Marvelous6.mp3");
  files [9] = minifm.loadFile ("Marvelous7.mp3");
  files [10] = minifm.loadFile ("Melancholy1.mp3");
  files [11] = minifm.loadFile ("Melancholy2.mp3");
  files [12] = minifm.loadFile ("Melancholy3.mp3");
  files [13] = minifm.loadFile ("Melancholy4.mp3");
  files [14] = minifm.loadFile ("Melancholy5.mp3");
  files [15] = minifm.loadFile ("Melancholy6.mp3");
  files [16] = minifm.loadFile ("Melancholy7.mp3");
  files [17] = minifm.loadFile ("Melancholy8.mp3");
  files [18] = minifm.loadFile ("Melancholy9.mp3");
  files [19] = minifm.loadFile ("Stoic1.mp3");
  files [20] = minifm.loadFile ("Stoic2.mp3");
  files [21] = minifm.loadFile ("Stoic3.mp3");
  files [22] = minifm.loadFile ("Stoic4.mp3");
  files [23] = minifm.loadFile ("Stoic5.mp3");
}

void draw(){
}

void serialEvent(Serial myPort){
  if (myPort.available() > 0){ // If data is available,
    val = myPort.readChar(); // read it and store it in val
    println(val);

    if (val == '1'){
      for(int i = 0; i < files.length;i++){
        files[i].pause();
        files[i].rewind();
      }
      //println("test");
      counter = counter +1;
      switch(counter){
        case 1:
          files[2].pause();
          files[0].play();
          break;
        case 2:
          files[0].pause();
          files[1].play();
          break;
        case 3:
          files[1].pause();
          files[2].play();
          counter=6;
          break;
        case 4:
          files[2].pause();
          files[3].play();
          break;
        case 5:
          files[3].pause();
          files[4].play();
          break;
        case 6:
          files[4].pause();
          files[5].play();
          break;
        case 7:
          files[5].pause();
          files[6].play();
          break;
        case 8:
          files[6].pause();
          files[7].play();
          break;
        case 9:
          files[7].pause();
          files[8].play();
          break;
        case 10:
          files[8].pause();
          files[9].play();
          break;
        case 11:
          files[9].pause();
          files[10].play();
          break;
        case 12:
          files[10].pause();
          files[11].play();
          break;
        case 13:
          files[11].pause();
          files[12].play();
          break;
        case 14:
          files[12].pause();
          files[13].play();
          break;
        case 15:
          files[13].pause();
          files[14].play();
          break;
        case 16:
          files[14].pause();
          files[15].play();
          break;
        case 17:
          files[15].pause();
          files[16].play();
          break;
        case 18:
          files[16].pause();
          files[17].play();
          break;
        case 19:
          files[17].pause();
          files[18].play();
          break;
        case 20:
          files[18].pause();
          files[19].play();
          counter=6;
          break;
        case 21:
          files[19].pause();
          files[20].play();
          break;
        case 22:
          files[20].pause();
          files[21].play();
          break;
        case 23:
          files[21].pause();
          files[22].play();
          break;
        case 24:
          files[22].pause();
          files[23].play();
          break;
      }
    }
    //println("test");
    counter = counter +1;
    switch(counter){
      case 1:
        files[19].pause();
        files[18].play();
        break;
      case 2:
        files[18].pause();
        files[17].play();
        break;
      case 3:
        files[17].pause();
        files[16].play();
        break;
      case 4:
        files[16].pause();
        files[15].play();
        break;
      case 5:
        files[15].pause();
        files[14].play();
        break;
      case 6:
        files[14].pause();
        files[13].play();
        break;
      case 7:
        files[13].pause();
        files[12].play();
        break;
      case 8:
        files[12].pause();
        files[11].play();
        break;
      case 9:
        files[11].pause();
        files[10].play();
        break;
      case 10:
        files[10].pause();
        files[9].play();
        break;
      case 11:
        files[9].pause();
        files[8].play();
        break;
      case 12:
        files[8].pause();
        files[7].play();
        break;
      case 13:
        files[7].pause();
        files[6].play();
        break;
      case 14:
        files[6].pause();
        files[5].play();
        break;
      case 15:
        files[5].pause();
        files[4].play();
        break;
      case 16:
        files[4].pause();
        files[3].play();
        break;
      case 17:
        files[3].pause();
        files[2].play();
        break;
      case 18:
        files[2].pause();
        files[1].play();
        break;
      case 19:
        files[1].pause();
        files[0].play();
        break;
      case 20:
        files[0].pause();
        files[1].play();
        break;
    }
  }
}
```