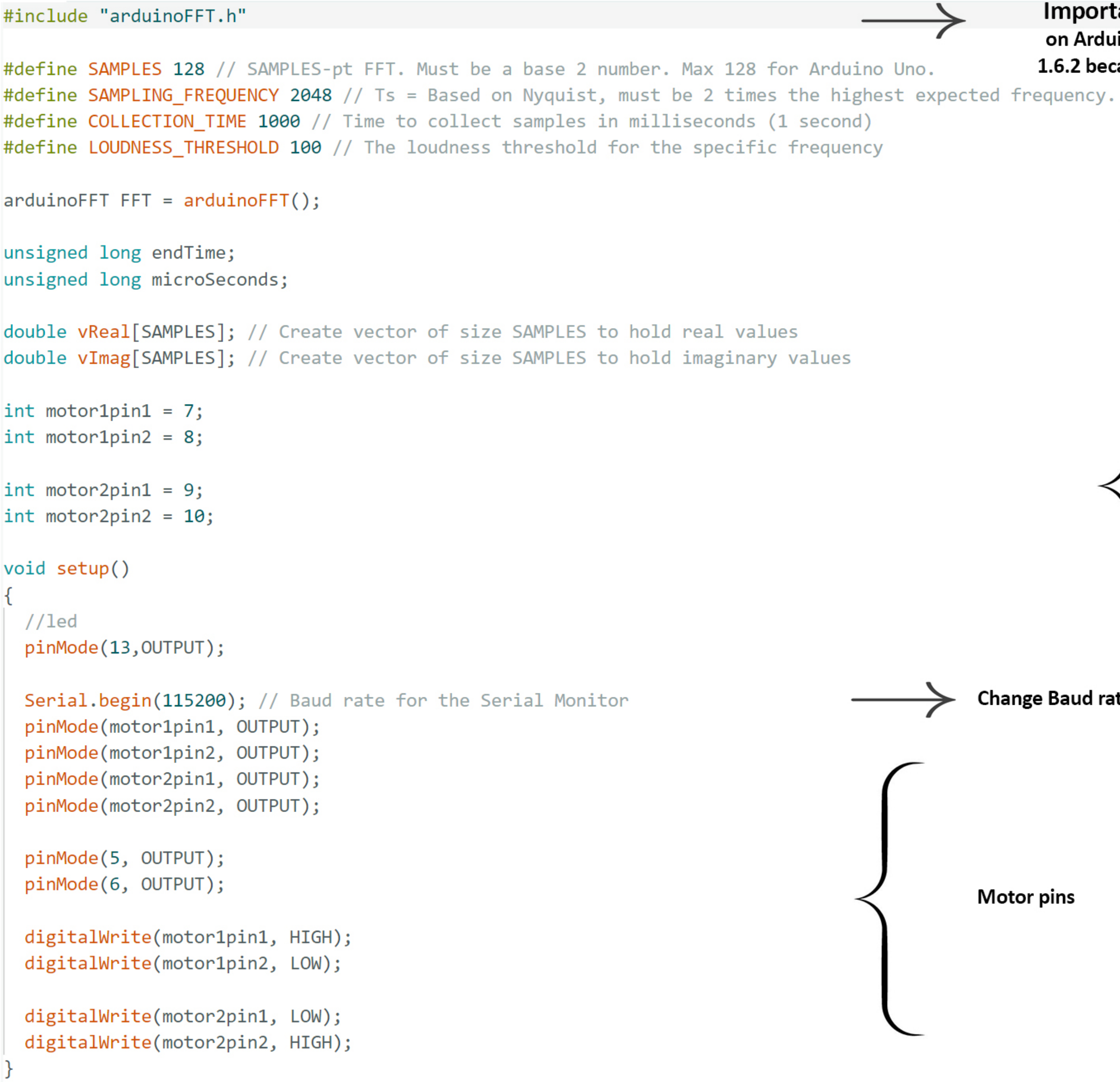


```
#include "arduinoFFT.h"

#define SAMPLES 128 // SAMPLES-pt FFT. Must be a base 2 number. Max 128 for Arduino Uno.
#define SAMPLING_FREQUENCY 2048 // Ts = Based on Nyquist, must be 2 times the highest expected frequency.
#define COLLECTION_TIME 1000 // Time to collect samples in milliseconds (1 second)
#define LOUDNESS_THRESHOLD 100 // The loudness threshold for the specific frequency

arduinoFFT FFT = arduinoFFT();

unsigned long endTime;
unsigned long microSeconds;

double vReal[SAMPLES]; // Create vector of size SAMPLES to hold real values
double vImag[SAMPLES]; // Create vector of size SAMPLES to hold imaginary values

int motor1pin1 = 7;
int motor1pin2 = 8;

int motor2pin1 = 9;
int motor2pin2 = 10;

void setup()
{
  //led
  pinMode(13,OUTPUT);

  Serial.begin(115200); // Baud rate for the Serial Monitor
  pinMode(motor1pin1, OUTPUT);
  pinMode(motor1pin2, OUTPUT);
  pinMode(motor2pin1, OUTPUT);
  pinMode(motor2pin2, OUTPUT);

  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

  digitalWrite(motor1pin1, HIGH);
  digitalWrite(motor1pin2, LOW);

  digitalWrite(motor2pin1, LOW);
  digitalWrite(motor2pin2, HIGH);
}
```

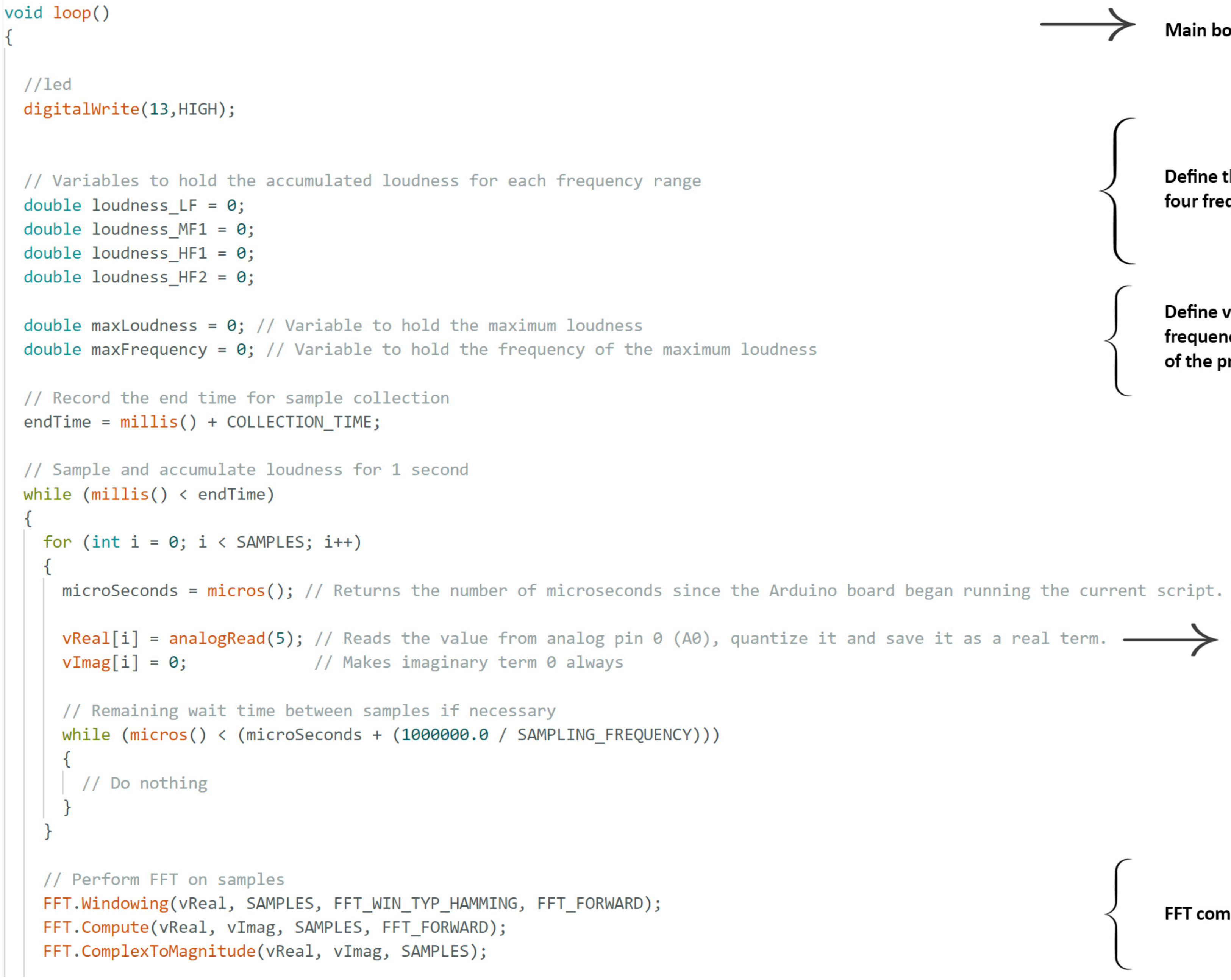
→ **Important:** **Install arduinoFFT library on Arduino IDE, make sure it's version 1.6.2 because 2.0.2 (new version) didn't work for us**

Constants: Change the number of samples (then sampling_frequency) if you have a more powerful microcontroller. Also, you can change collection time (and maybe delay at the end of the code) and loudness threshold if needed

Other varilbes and constants

→ **Change Baud rate on the Arduino IDE to 115200**

Motor pins



```
void loop()
{

  //led
  digitalWrite(13,HIGH);


  // Variables to hold the accumulated loudness for each frequency range
  double loudness_LF = 0;
  double loudness_MF1 = 0;
  double loudness_HF1 = 0;
  double loudness_HF2 = 0;

  double maxLoudness = 0; // Variable to hold the maximum loudness
  double maxFrequency = 0; // Variable to hold the frequency of the maximum loudness

  // Record the end time for sample collection
  endTime = millis() + COLLECTION_TIME;

  // Sample and accumulate loudness for 1 second
  while (millis() < endTime)
  {
    for (int i = 0; i < SAMPLES; i++)
    {
      microSeconds = micros(); // Returns the number of microseconds since the Arduino board began running the current script.

      vReal[i] = analogRead(5); // Reads the value from analog pin 0 (A0), quantize it and save it as a real term.
      vImag[i] = 0;             // Makes imaginary term 0 always

      // Remaining wait time between samples if necessary
      while (micros() < (microSeconds + (1000000.0 / SAMPLING_FREQUENCY)))
      {
        // Do nothing
      }
    }

    // Perform FFT on samples
    FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
    FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
    FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);
```

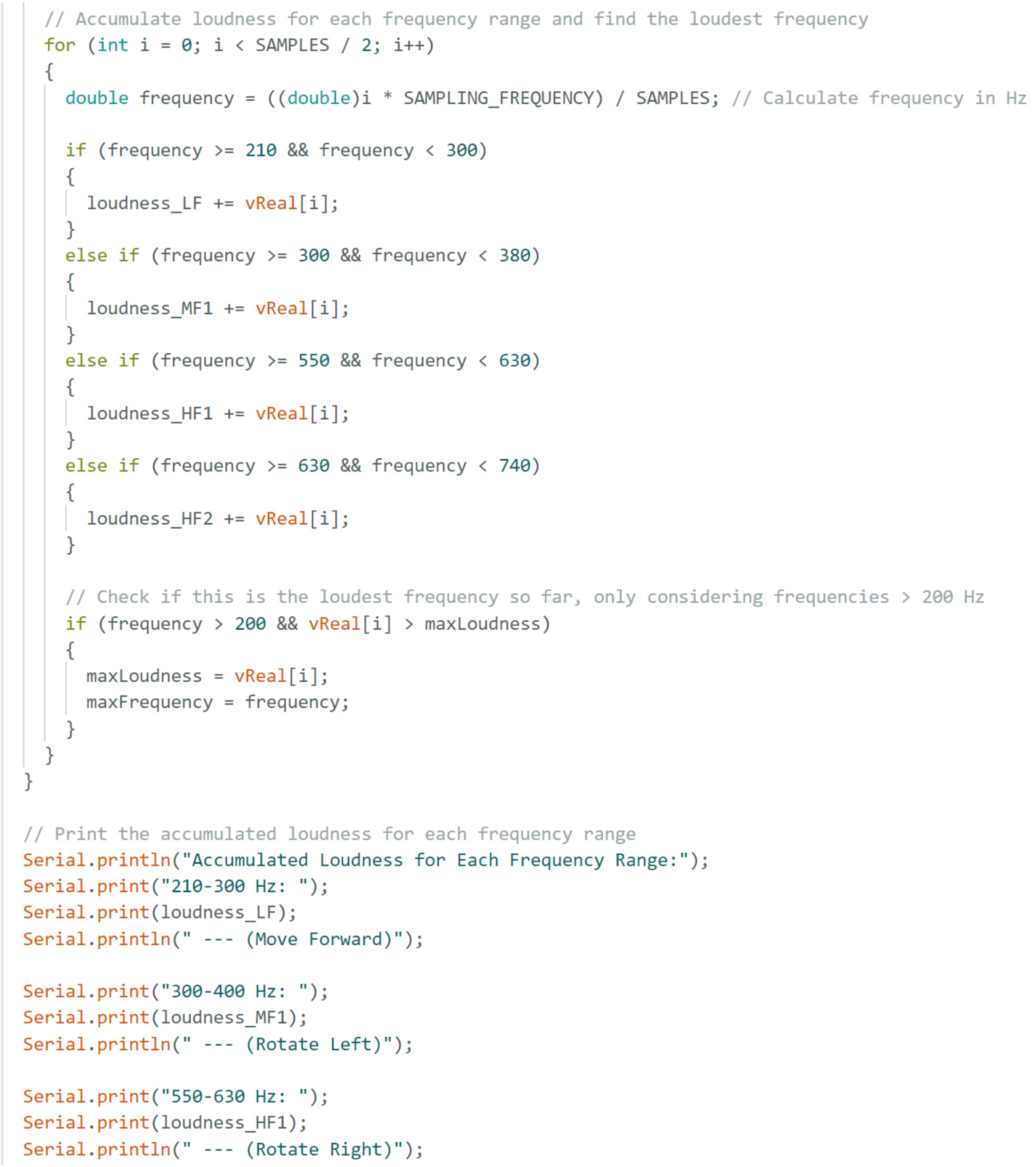
Main body of the code

Define the accumulated loudness for the four frequency ranges

Define variables for the max loudness in the frequencies. It is not needed for this version of the project

VERY IMPORTANT: Change the analog pin based on your circuit scheme

FFT computation

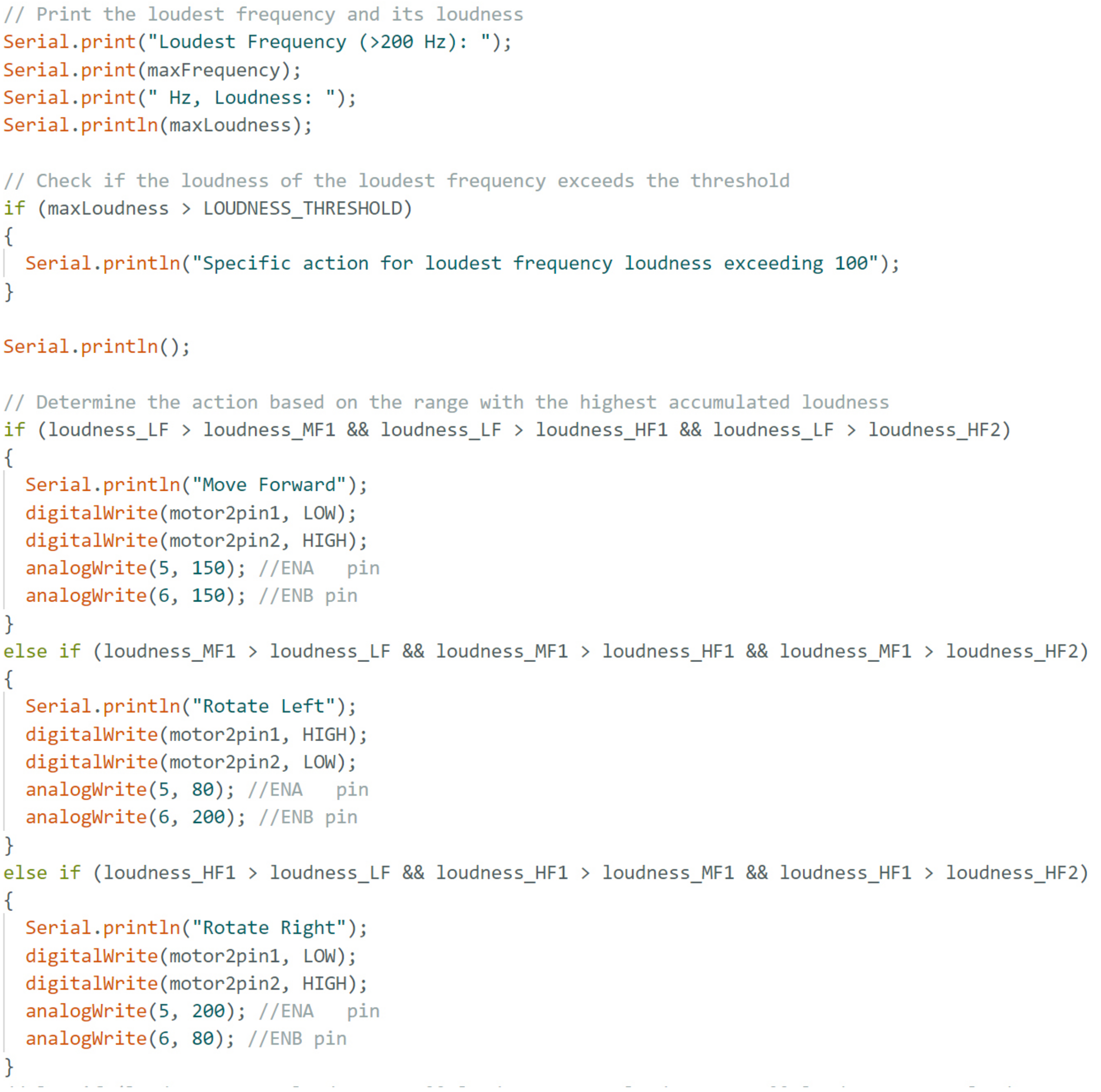


```
  // Accumulate loudness for each frequency range and find the loudest frequency
  for (int i = 0; i < SAMPLES / 2; i++)
  {
    double frequency = ((double)i * SAMPLING_FREQUENCY) / SAMPLES; // Calculate frequency in Hz

    if (frequency >= 210 && frequency < 300)
    {
      loudness_LF += vReal[i];
    }
    else if (frequency >= 300 && frequency < 380)
    {
      loudness_MF1 += vReal[i];
    }
    else if (frequency >= 550 && frequency < 630)
    {
      loudness_HF1 += vReal[i];
    }
    else if (frequency >= 630 && frequency < 740)
    {
      loudness_HF2 += vReal[i];
    }

    // Check if this is the loudest frequency so far, only considering frequencies > 200 Hz
    if (frequency > 200 && vReal[i] > maxLoudness)
    {
      maxLoudness = vReal[i];
      maxFrequency = frequency;
    }
  }
}

// Print the accumulated loudness for each frequency range
Serial.println("Accumulated Loudness for Each Frequency Range:");
Serial.print("210-300 Hz: ");
Serial.print(loudness_LF);
Serial.println(" --- (Move Forward)");

Serial.print("300-400 Hz: ");
Serial.print(loudness_MF1);
Serial.println(" --- (Rotate Left)");

Serial.print("550-630 Hz: ");
Serial.print(loudness_HF1);
Serial.println(" --- (Rotate Right)");
```

Calculation of the accumulated loudness for the four frequency ranges

Calculation of the loudest frequency

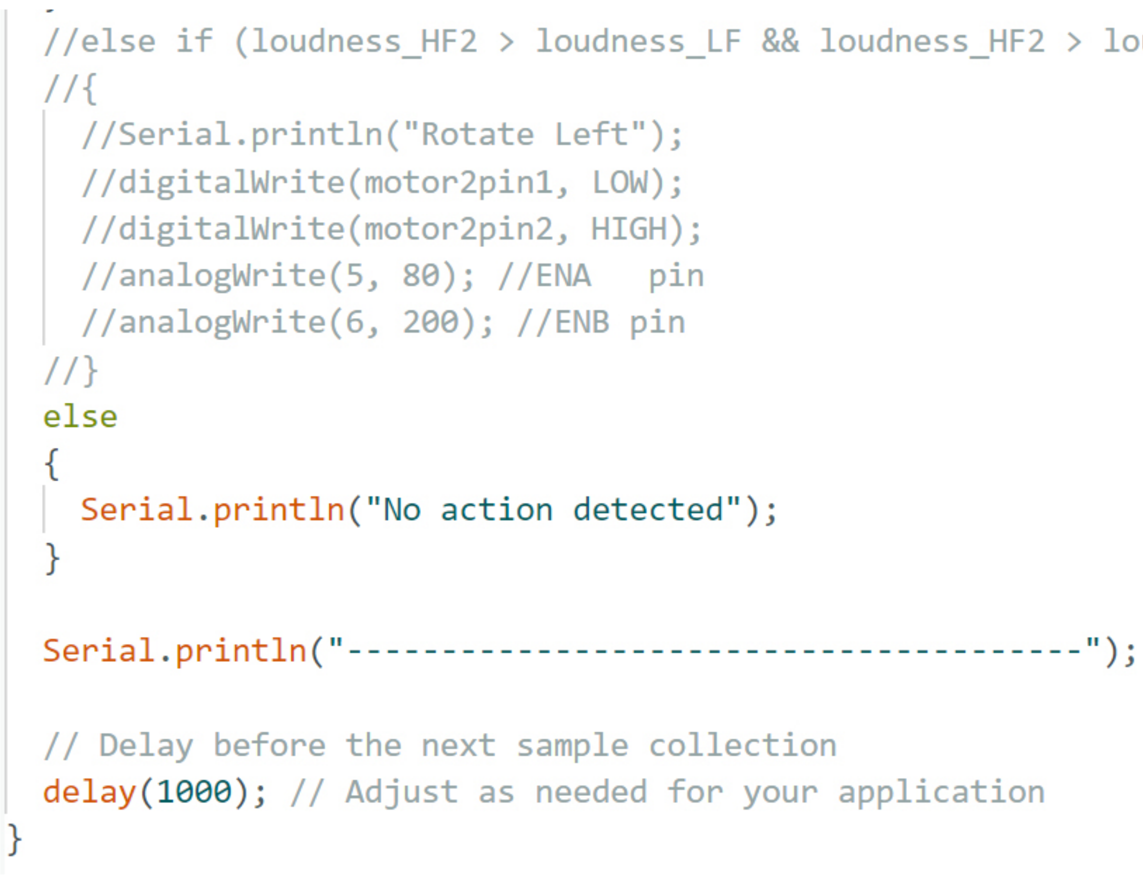
Print the results, Open Serial Monitor to see the numbers. The Arduino board needs to be connected to your laptop to see the results



```
// Print the loudest frequency and its loudness
Serial.print("Loudest Frequency (>200 Hz): ");
Serial.print(maxFrequency);
Serial.print(" Hz, Loudness: ");
Serial.println(maxLoudness);

// Check if the loudness of the loudest frequency exceeds the threshold
if (maxLoudness > LOUDNESS_THRESHOLD)
{
  Serial.println("Specific action for loudest frequency loudness exceeding 100");
}

Serial.println();

// Determine the action based on the range with the highest accumulated loudness
if (loudness_LF > loudness_MF1 && loudness_LF > loudness_HF1 && loudness_LF > loudness_HF2)
{
  Serial.println("Move Forward");
  digitalWrite(motor2pin1, LOW);
  digitalWrite(motor2pin2, HIGH);
  analogWrite(5, 150); //ENA   pin
  analogWrite(6, 150); //ENB pin
}
else if (loudness_MF1 > loudness_LF && loudness_MF1 > loudness_HF1 && loudness_MF1 > loudness_HF2)
{
  Serial.println("Rotate Left");
  digitalWrite(motor2pin1, HIGH);
  digitalWrite(motor2pin2, LOW);
  analogWrite(5, 80); //ENA   pin
  analogWrite(6, 200); //ENB pin
}
else if (loudness_HF1 > loudness_LF && loudness_HF1 > loudness_MF1 && loudness_HF1 > loudness_HF2)
{
  Serial.println("Rotate Right");
  digitalWrite(motor2pin1, LOW);
  digitalWrite(motor2pin2, HIGH);
  analogWrite(5, 200); //ENA   pin
  analogWrite(6, 80); //ENB pin
}
```

Print the result of the loudest frequency (if is higher than the threshold) on Serial Monitor

If conditions, translation of the result of accumulated loudness of each frequency range to the corresponding movement. The loudest range determines the movement of .that moment
IMPORTANT: You can change the speed of the motors here



```
  //else if (loudness_HF2 > loudness_LF && loudness_HF2 > loudness_MF1 && loudness_HF2 > loudness_HF1)
  //{
    //Serial.println("Rotate Left");
    //digitalWrite(motor2pin1, LOW);
    //digitalWrite(motor2pin2, HIGH);
    //analogWrite(5, 80); //ENA   pin
    //analogWrite(6, 200); //ENB pin
  //}
  else
  {
    Serial.println("No action detected");
  }

  Serial.println("---------------------------------------");

  // Delay before the next sample collection
  delay(1000); // Adjust as needed for your application
}
```

You can change the delay here, and may need to change it in the beginning of the code