

```

#include <FastLED.h>
int sound=A0;
const int sampleWindow = 10; // Sample window width in mS (50 mS =
20Hz)

const int samplespeed2=100;
const int samplespeed3=samplespeed2*2;
unsigned int sample;
    int speed1=0;
    int speed2=0;
    int speed3=0;
    int previous2=0;
    int previous3=0;

// How many leds in your strip?
#define NUM_LEDS 14

// For led chips like Neopixels, which have a data line, ground, and
power, you just
// need to define DATA_PIN. For led chipsets that are SPI based (four
wires - data, clock,
// ground, and power), like the LPD8806 define both DATA_PIN and
CLOCK_PIN
#define DATA_PIN 8
#define CLOCK_PIN 5

// Define the array of leds
CRGB leds[NUM_LEDS];

void setup() {
    Serial.begin(9600);
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);
}

void loop() {
    float s=analogRead(A0);
    Serial.print("señal ");
    Serial.println(s);
    s=map(s,480,550,0,255);
    Serial.print("MAP ");
    Serial.println(s);
    //Serial.println(s_st);
    for (int i=0;i<=NUM_LEDS;i++){
        // Turn the LED on, then pause
        leds[i] = CRGB::Black;
        FastLED.show();}
        // delay(50);
// for (int i=0;i<=NUM_LEDS;i++)
//{
//while(s!=5&& s>5)

```

```

// {
  if(s==3 && s<10){
    for (int i=0;i<=NUM_LEDS;i++){
      leds[i] = CRGB(80+s,150,s);
      FastLED.show();}}

  else if (s>10 && s<15){
    for (int i=0;i<=NUM_LEDS;i++){
      // Now turn the LED off, then pause
      leds[i] = CRGB(0+s,(150-s),250);
      FastLED.show();}}

    else if (s==15 && s<30){
      for (int i=0;i<=NUM_LEDS;i++){
        // Now turn the LED off, then pause
        leds[i] = CRGB(s,30,(255-s));
        FastLED.show();}}

    else if (s>30 && s<60){
      for (int i=0;i<=NUM_LEDS;i++){
        // Now turn the LED off, then pause
        leds[i] = CRGB((s+20),100,s);
        //leds[i] = CRGB::HotPink;
        FastLED.show();}}

    else if (s>60 && s<100){
      for (int i=0;i<=NUM_LEDS;i++){
        // Now turn the LED off, then pause
        leds[i] = CRGB(0+s,(70),200);
        //leds[i] = CRGB::BlueViolet;
        FastLED.show();}}

//   else if (s>60 && s<70){
//     for (int i=0;i<=NUM_LEDS;i++){
//       // Now turn the LED off, then pause
//       //leds[0] = CRGB(s+50,0,(100+s));
//       leds[i] = CRGB::RosyBrown;
//       FastLED.show();}}
//
//   else if (s>70 && s<80){
//     for (int i=0;i<=NUM_LEDS;i++){
//       // Now turn the LED off, then pause
//       //leds[0] = CRGB((70+s),0,s);
//       leds[i] = CRGB::DarkOrange;
//       FastLED.show();}}
//
//   else if (s>80 && s<90){
//     for (int i=0;i<=NUM_LEDS;i++){
//       // Now turn the LED off, then pause
//       //leds[0] = CRGB(20+s,(0),s);

```

```

// leds[i] = CRGB::DeepPink;
// FastLED.show(); } }
//
// else if (s>90 && s<100){
//   for (int i=0;i<=NUM_LEDS;i++){
//     // Now turn the LED off, then pause
//     //leds[0] = CRGB(s+10,s,(90+s));
//     leds[i] = CRGB::Cyan;
//     FastLED.show();} }

    else if (s>100 && s<150){
      for (int i=0;i<=NUM_LEDS;i++){
        // Now turn the LED off, then pause
        leds[i] = CRGB((50+s),(50+s),s);
        FastLED.show();}}

//   else if (s>110 && s<120){
//     //
//     for (int i=0;i<=NUM_LEDS;i++){
//       // Now turn the LED off, then pause
//       leds[i] = CRGB(0,(50+s),(50+s));
//       FastLED.show();}}
//
//   else if (s>120 && s<130){
//     for (int i=0;i<=NUM_LEDS;i++){
//       //
//       // Now turn the LED off, then pause
//       leds[i] = CRGB((50+s),s,(50+s));
//       FastLED.show();}}
//
//   else if (s>130 && s<140){
//     for (int i=0;i<=NUM_LEDS;i++){
//       //
//       // Now turn the LED off, then pause
//       leds[i] = CRGB((45+s),(45+s),s);
//       FastLED.show();}}
//
//   else if (s>140 && s<150){
//     for (int i=0;i<=NUM_LEDS;i++){
//       // Now turn the LED off, then pause
//       leds[i] = CRGB(s,(45+s),(45+s));
//       FastLED.show();}}

    else if (s>150 && s<200){
      for (int i=0;i<=NUM_LEDS;i++){
        // Now turn the LED off, then pause
        leds[i] = CRGB((45+s),0,(45+s));
        FastLED.show();}}

    else if (s>200 && s<250){

```

```

    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB((30+s),(30+s),s);
FastLED.show();}}

    else if (s>250 && s<300){
    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB(s,(30+s),(30+s));
FastLED.show();}}

    else if (s>300 && s<350){
    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB((30+s),s,(30+s));
FastLED.show();}}

    else if (s>350 && s<450){
    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB((20+s),(20+s),s);
FastLED.show();}}

    else if (s>450 && s<550){
    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB(s,(20+s),(20+s));
FastLED.show();}}

    else if (s>550 && s<650){
    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB((20+s),s,(20+s));
FastLED.show();}}

    else if (s>650 && s<750){
    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB((0+s),(255-s),2+s);
FastLED.show();}}

    else if (s>750 && s<800){
    for (int i=0;i<=NUM_LEDS;i++){
// Now turn the LED off, then pause
leds[i] = CRGB((10+s),(10+s),s);
FastLED.show();}}

// else if (s>240 && s<250)
// {
// for (int i=0;i<=NUM_LEDS;i++){

```

```

// // Now turn the LED off, then pause
// leds[i] = CRGB(0,(5+s),(5+s));
// FastLED.show();
//   }}
//
//   else if(s>250 && s<256)
//   {
//     for (int i=0;i<=NUM_LEDS;i++){
// // Now turn the LED off, then pause
// leds[i] = CRGB((0+s),s,(0+s));
// FastLED.show();
//   }}

//   else
//   {
// // Now turn the LED off, then pause
// leds[0] = CRGB(0,0,255);
// FastLED.show();
//   }
  delay(100);
  // s=0;
//}
//}

  unsigned long startMillis= millis(); // Start of sample window
  unsigned long startMillis2= millis(); // Start of sample window
  unsigned long startMillis3= millis(); // Start of sample window

  unsigned int peakToPeak = 0; // peak-to-peak level

  unsigned int signalMax = 0;
  unsigned int signalMin = 1024;

  // collect data for 50 mS
  while (millis() - startMillis < sampleWindow)
  {
    sample = analogRead(A0);
    if (sample < 1024) // toss out spurious readings
    {
      if (sample > signalMax)
      {
        signalMax = sample; // save just the max levels
      }
      else if (sample < signalMin)
      {
        signalMin = sample; // save just the min levels
      }
    }
  }
  peakToPeak = (signalMax - signalMin)*3; // max - min = peak-peak
  amplitude

```

```
double volts = (peakToPeak * 5.0); // convert to volts
int speed1 = constrain(peakToPeak, 0, 255)*1.5;
if (startMillis2 - previous2 >= samplespeed2)
{ speed2=speed1;
previous2=startMillis2;
}
  if (startMillis3 - previous3 >= samplespeed3)
{speed3=speed1;
previous3=startMillis3;
}

//Serial.print("velocita1");
//Serial.println(s);
//
//  Serial.print("velocita1");
//  Serial.println(speed1);
//  Serial.print("velocita2");
//  Serial.println(speed2);
//  Serial.print("velocita3");
//  Serial.println(speed3);

  analogWrite(13,speed1);
  analogWrite(12,speed2);
  analogWrite(11,speed3);

}
```