#include <Adafruit\_NeoPixel.h>

#define LED\_PIN 0

#define LS\_PIN 2

//#define NUM\_PIXEL 12

// Parameter 1 = number of pixels in strip

// Parameter 2 = pin number (most are valid)

// Parameter 3 = pixel type flags, add together as needed:

//   NEO\_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)

//   NEO\_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)

//   NEO\_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)

//   NEO\_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(60, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

void setup() {

  strip.begin();

  strip.show(); // Initialize all pixels to 'off'

  pinMode (LS\_PIN, INPUT);

}

void loop() {

  int sensorValue = analogRead(LS\_PIN);

  if (sensorValue >300 ){

    colorWipe(strip.Color(255, 0, 0), 50); // Red

    colorWipe(strip.Color(0, 255, 0), 50); // Green

    colorWipe(strip.Color(0, 0, 255), 50);

  }else

  if(sensorValue >150)

    rainbow(20);

 else

   rainbowCycle(20);

}

/\*

void loop {

  // Some example procedures showing how to display to the pixels:

  colorWipe(strip.Color(255, 0, 0), 50); // Red

  colorWipe(strip.Color(0, 255, 0), 50); // Green

  colorWipe(strip.Color(0, 0, 255), 50); // Blue

  rainbow(20);

  rainbowCycle(20);

}

\*/

// Fill the dots one after the other with a color

void colorWipe(uint32\_t c, uint8\_t wait) {

  for(uint16\_t i=0; i<strip.numPixels(); i++) {

      strip.setPixelColor(i, c);

      strip.show();

      delay(wait);

  }

}

void rainbow(uint8\_t wait) {

  uint16\_t i, j;

  for(j=0; j<256; j++) {

    for(i=0; i<strip.numPixels(); i++) {

      strip.setPixelColor(i, Wheel((i+j) & 255));

    }

    strip.show();

    delay(wait);

  }

}

// Slightly different, this makes the rainbow equally distributed throughout

void rainbowCycle(uint8\_t wait) {

  uint16\_t i, j;

  for(j=0; j<256\*5; j++) { // 5 cycles of all colors on wheel

    for(i=0; i< strip.numPixels(); i++) {

      strip.setPixelColor(i, Wheel(((i \* 256 / strip.numPixels()) + j) & 255));

    }

    strip.show();

    delay(wait);

  }

}

// Input a value 0 to 255 to get a color value.

// The colours are a transition r - g - b - back to r.

uint32\_t Wheel(byte WheelPos) {

  if(WheelPos < 85) {

   return strip.Color(WheelPos \* 3, 255 - WheelPos \* 3, 0);

  } else if(WheelPos < 170) {

   WheelPos -= 85;

   return strip.Color(255 - WheelPos \* 3, 0, WheelPos \* 3);

  } else {

   WheelPos -= 170;

   return strip.Color(0, WheelPos \* 3, 255 - WheelPos \* 3);

  }

}