```
#include <Adafruit_NeoPixel.h>
#include <SparkFun_APDS9960.h>
#include "Wire.h"

#define NEOPIXED_CONTROL_PIN 7
#define NUM_LEDS 30 // EDIT YOUR led LIGHTS
#define APDS9960_INT 2
#define LED_SPEED_STEP_INTERVAL 150

Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_LEDS, NEOPIXED_CONTROL_PIN, NEO_RBG + NEO_KHZ800);
SparkFun_APDS9960 apds = SparkFun_APDS9960();


unsigned long lastLedChangeTime = 0;
short ledDirection = 0;
short colorSelection = 0;
byte ledStates[] = {150, 100, 70, 50, 40, 30, 10, 2, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int isr_flag = 0;


void setup()
{
   Serial.begin(9600);
   Serial.println("Program started");
   strip.begin();
   pinMode(APDS9960_INT, INPUT);
```

```
  attachInterrupt(0, interruptRoutine, FALLING);
  if ( apds.init() && apds.enableGestureSensor(true)) {
      Serial.println("APDS-9960 initialization complete");
  } else {
     Serial.println("Something went wrong during APDS-9960 init!");
  }
  lastLedChangeTime = millis();
  Serial.println("Init succesfully");
}

void loop()
{
  if( isr_flag == 1 ) {
    detachInterrupt(0);
    handleGesture();
    isr_flag = 0;
    attachInterrupt(0, interruptRoutine, FALLING);
  }
  animateLeds();
}

void interruptRoutine()
{
  isr_flag = 1;
}

/**
```

```
 * This will handle gestures from the APDS9960 sensor
 * Up and Down gestures will call toggleColor function
 * Left and Right gestures will change the led animation
 */
void handleGesture() {
   if ( !apds.isGestureAvailable() ) {
      return;
   }
   switch ( apds.readGesture() ) {
     case DIR_UP:
       Serial.println("UP");
       toggleColor();
       break;
    case DIR_DOWN:
       Serial.println("DOWN");
       toggleColor();
       break;
     case DIR_LEFT:
       ledDirection = 1;
       Serial.println("LEFT");
       break;
     case DIR_RIGHT:
       ledDirection = -1;
       Serial.println("RIGHT");
       break;
     case DIR_FAR:
       ledDirection = 0;
```

```
        Serial.println("FAR");
        break;
    }
}

/**
 * Change current leds color
 * Each time this function is called will change the leds state
 */
void toggleColor()
{
    if (colorSelection == 0) {
        colorSelection = 1;
    } else if (colorSelection == 1) {
        colorSelection = 2;
    } else {
        colorSelection = 0;
    }
}

/**
 * The animation will run after LED_SPEED_STEP_INTERVAL millis
 * First the rotateLeds function is called, then the leds colors are set using the strip api
 */
void animateLeds()
{
    if (millis() - lastLedChangeTime < LED_SPEED_STEP_INTERVAL) {
```

```
      return;
    }
    rotateLeds();
    for (int i=0; i < NUM_LEDS; i++) {
      strip.setPixelColor(i, getColor(ledStates[i]));
      strip.show();
    }
    lastLedChangeTime = millis();
}

/**
 * Using a secondary array "intermediateLedStates", leds intensities are animated
 * First the values from "ledStates" are copied on "intermediateLedStates" like so
 * let's sat the "ledStates" array is {100, 80, 60, 0, 0, 0} and the ledDirection is 1
 * then after this function is called "ledStates" array is {0, 100, 80, 60, 0, 0} simulating a rotation effect
 */
void rotateLeds()
{
    byte intermediateLedStates[NUM_LEDS];
    for (int i=0; i < NUM_LEDS; i++) {
      intermediateLedStates[i] = 0;
    }
    for (int i=0; i < NUM_LEDS; i++) {
      if (ledDirection == 1) {
        if (i == NUM_LEDS -1) {
          intermediateLedStates[0] = ledStates[i];
        } else {
```

```
            intermediateLedStates[i + 1] = ledStates[i];
          }
        } else {
          if (i == 0) {
            intermediateLedStates[NUM_LEDS - 1] = ledStates[i];
          } else {
            intermediateLedStates[i - 1] = ledStates[i];
          }
        }
    }
    for (int i=0; i < NUM_LEDS; i++) {
        ledStates[i] = intermediateLedStates[i];
    }
}

uint32_t getColor(int intensity)
{
    switch (colorSelection) {
        case 0:
          return strip.Color(intensity, 0, 0);
        case 1:
          return strip.Color(0, intensity, 0);
        default:
          return strip.Color(0, 0, intensity);
    }

}
```