



SmartElex RGB and Gesture Sensor - APDS-9960

Touchless gestures are the new frontier in the world of human-machine interfaces. By swiping your hand over a sensor, you can control a computer, microcontroller, robot, etc. The Avago APDS-9960 offers ambient light and color (as clear, red, green, and blue) measuring, proximity detection, and gesture sensing.



Board Overview

Pin Descriptions

The APDS-9960 breakout board provides 6 pins to provide power to the sensor and I²C bus.

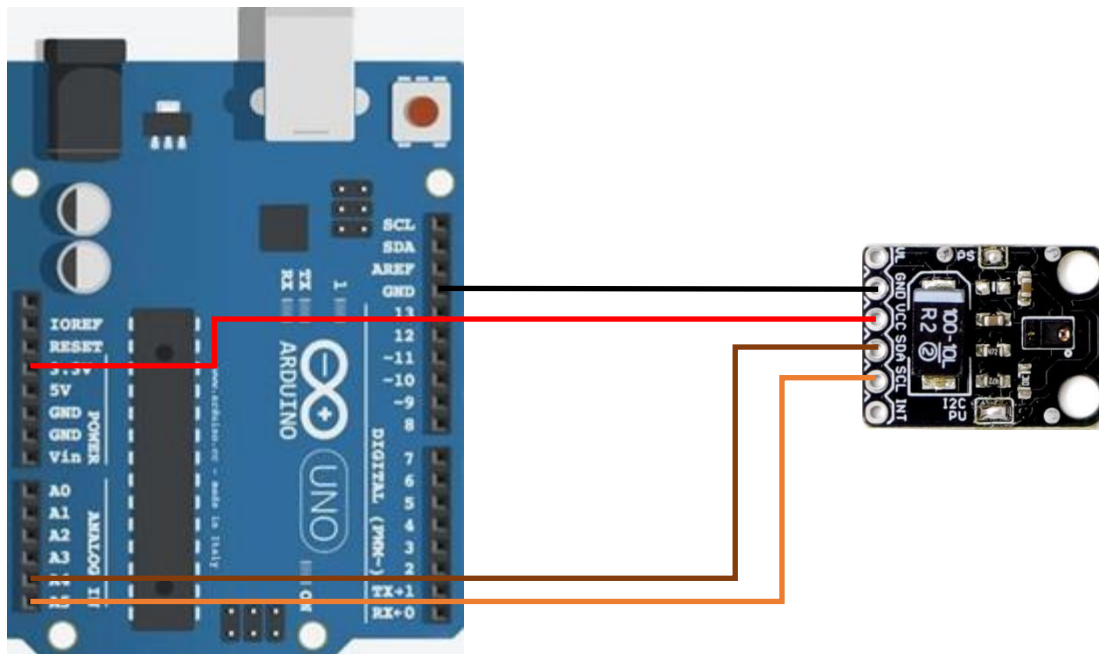
Pin Label	Description
VL	Optional power to the IR LED if PS jumper is disconnected. Must be 3.0 - 4.5V
GND	Connect to ground.
VCC	Used to power the APDS-9960 sensor. Must be 2.4 - 3.6V
SDA	I ² C data
SCL	I ² C clock
INT	External interrupt pin. Active LOW on interrupt event

Setting the Jumpers

On the front of the breakout board are 2 solder jumpers:

- **PS** -- This jumper connects the power supplies of the sensor and IR LED (also located on the APDS-9960) together. When the jumper is closed (i.e. connected), you only need to supply power to the VCC pin to power both the sensor and the IR LED. If the jumper is open, you need to provide power to both the VCC (2.4 - 3.6V) and VL (3.0 - 4.5V) pins separately. This jumper is **closed** by default.
- **I²C PU** -- This is a 3-way solder jumper that is used to connect and disconnect the I²C pullup resistors. By default, this jumper is **closed**, which means that both SDA and SCL lines have connected pullup resistors on the breakout board. Use some solder wick to open the jumper if you do not need the pullup resistors (e.g. you have pullup resistors that are located on the I²C bus somewhere else).

Connect the Breakout Board



Note that we are leaving VL on the breakout board unconnected.

IMPORTANT: You must use 3.3V! If you try to use a 5V power supply or 5V I²C communications, you risk damaging the APDS-9960. If you are using a 5V microcontroller, then you need to have some kind of level shifting.

Arduino

APDS9960

A5(SCL)	SCL
A4(SDA)	SDA
3.3V	VCC
GND	GND

Arduino Library Installation

To use the APDS-9960, you will need some supporting software. Arduino has created an library that makes the APDS-9960 easy to use. GO to library manager → search for “**Arduino_APDS9960**” & install it.

Gesture Sensing Example

Load the GestureSensor Example

Open up the Arduino program and select File → Examples → Arduino_APDS9960 → GestureSensor.

Important Note: If the sensor is not detecting gestures then uncomment the ‘APDS.setGestureSensitivity();’ line & adjust the sensitivity to 95.

for setGestureSensitivity(..) a value between 1 and 100 is required.

Higher values make the gesture recognition more sensitive but less accurate(a wrong gesture may be detected). Lower values makes the gesture recognition more accurate but less sensitive (some gestures may be missed).

Default is 80

```
// APDS.setGestureSensitivity(80);
```

```
/*  
APDS-9960 - Gesture Sensor  
  
This example reads gesture data from the on-board APDS-9960 sensor of the  
Nano 33 BLE Sense and prints any detected gestures to the Serial Monitor.
```

Gesture directions are as follows:

- UP: from USB connector towards antenna
- DOWN: from antenna towards USB connector
- LEFT: from analog pins side towards digital pins side
- RIGHT: from digital pins side towards analog pins side

The circuit:

- Arduino Nano 33 BLE Sense

This example code is in the public domain.

```
*/  
  
#include <Arduino_APDS9960.h>  
  
void setup() {  
  Serial.begin(9600);  
  while (!Serial);  
  
  if (!APDS.begin()) {  
    Serial.println("Error initializing APDS-9960 sensor!");  
  }  
  
  // for setGestureSensitivity(..) a value between 1 and 100 is required.  
  // Higher values make the gesture recognition more sensitive but less accurate  
  // (a wrong gesture may be detected). Lower values makes the gesture recognition  
  // more accurate but less sensitive (some gestures may be missed).  
  // Default is 80  
  //APDS.setGestureSensitivity(80);  
  
  Serial.println("Detecting gestures ...");  
}  
void loop() {  
  if (APDS.gestureAvailable()) {  
    // a gesture was detected, read and print to Serial Monitor  
    int gesture = APDS.readGesture();  
  
    switch (gesture) {  
      case GESTURE_UP:  
        Serial.println("Detected UP gesture");  
        break;  
  
      case GESTURE_DOWN:  
        Serial.println("Detected DOWN gesture");  
        break;  
  
      case GESTURE_LEFT:  
        Serial.println("Detected LEFT gesture");  
    }  
  }  
}
```

```
        break;

    case GESTURE_RIGHT:
        Serial.println("Detected RIGHT gesture");
        break;

    default:
        // ignore
        break;
}
}
}
//////////////////////////////////////////////////////////////////END//////////////////////////////////////////////////////////////////
```

You should see a messages noting that "Detecting gestures ...".

Hover your hand 4 to 8 inches (10 to 20 cm) above the sensor but off to one side (i.e. not directly above the sensor). While maintaining the same height, swipe your hand over the sensor (into and then immediately out of range of the sensor). If you move too fast, the sensor will not recognize the gesture.

Gestures will appear on the serial monitor, which indicate the direction of the swipe.

Serial Monitor



Supported Gestures

Here is a list of the currently supported gestures. Make sure each gesture begins outside of the range of the sensor, moves into the range of the sensor, and ends outside the range of the sensor.

Gesture	Description
UP	A swipe from the bottom of the board to the top and out of range of the sensor. Make sure that your wrist/arm is not in the sensor's range at the end of the swipe!
DOWN	A swipe from the top of the board to the bottom and out of range of the sensor.
LEFT	A swipe from the right side of the board to the left and out of range of the sensor.
RIGHT	A swipe from the left side of the board to the right and out of range of the sensor.