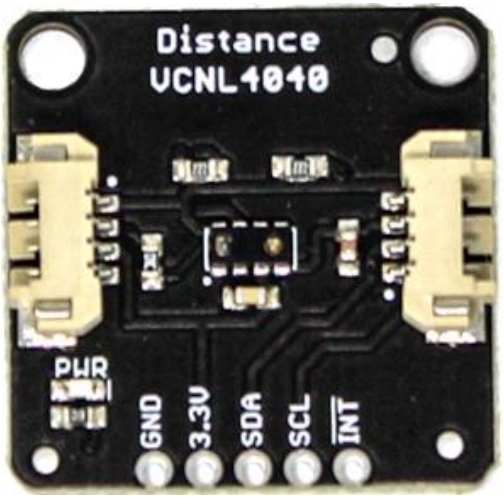
SmartElex Proximity Sensor Breakout - 20cm, VCNL4040

The SmartElex Proximity Sensor is a simple IR presence and ambient light sensor utilizing the VCNL4040. This sensor is excellent for detecting if something has appeared in front of the sensor; detecting objects qualitatively up to **20cm** away. This means you can detect if something is there, and if it is closer or further away since the last reading, but it's difficult to say it is 7.2cm away.



We often see this type of sensor on automatic towel dispensers, automatic faucets, etc. However, the VCNL4040 has **no dead zone** and can read all the way up to the face of the sensor. The Proximity VCNL4040 sensor is a great digital alternative to the popular analog PIR sensors. The Proximity Sensor also has an ambient light sensor built it which is excellent if you need a digital light sensor.

Hardware Overview

Note: If you want to do anything outside of what is covered in this tutorial or the example code, please refer to the VCNL4040 datasheet for exact details on the sensor functionality.

Power LED

There is a power status LED to help make sure the board is getting power. You can power the board either through the *polarized* **connector** or the **breakout pins** (**PWR** and **GND**) provided. This system is meant to use **3.3V**, be sure that you are **NOT** using another voltage when using the breakout.

VCNL4040 Sensor

The VCNL4040 is a **proximity sensor** (PS), **ambient light sensor** (ALS), and a high power **IR emitter** (IRED) integrated into a single package. The ambient light sensor and proximity sensor operate in a parallel structure. The combination of the two sensors along with the IR emitter form the proximity detector.

The sensor has an operating voltage range from **2.5V** to **3.6V**. It is recommended that you use this board with the intended **3.3V** of the connect. The VCNL4040 also includes excellent temperature compensation. For any details not covered in this guide, please refer to the datasheet.

Note: This sensor is excellent for detecting qualitative changes (if an object is there or moved since the last reading).

IR Emitter

The IR emmiter is immune to red glow with a **940 nm** wavelength (for those discrete applications) and the sink current is programmable (**200 mA default**). For more details on the IR emitter, please refer to the datasheet.

Ambient Light Sensor (ALS)

The VCNL4040 offers a **16-bit** high resolution ALS with ±10 % accuracy and is immune to fluorescent light flicker. With an ambient light sensing capability down to 0.01 lux/step, the VCNL4040 works well under a low transmittance lens design (dark lens). In addition, the patented FiltronTM technology offers great background light cancellation capability (including sunlight) without utilizing microcontroller resources.

Proximity Sensor (PS)

The proximity sensor features smart persistence, which prevents the misjudgment of proximity sensing but also keeps a fast response time. In active force mode, a single measurement can be requested for more design flexibility and/or power saving.

I^2C Address

The sensor has a single slave address **0x60** (HEX) of 7-bit addressing, following I^2C protocol.

Connections

The simplest way to use the Proximity Sensor is through the connector. The connectors are polarized for the I^2C connection and power. (**They are tied to their corresponding breakout pins.*)However, the board also provides five labeled breakout pins. You can connect these lines to the I^2C bus of your microcontroller and power pins (**3.3V** and **GND**). The interrupt pin is broken out to use for triggered events.

Interrupt Pin

High and low interrupt thresholds can be programmed for both the ambient light sensor and proximity sensor, allowing the component to use a minimal amount of the microcontrollers resources. Adjustable persistence sets up the amount of consecutive hits required before an interrupt event occurs, to prevent false triggers.

The VCNL4040 also supports an easy to use proximity detection logic mode, that triggers when the PS high threshold is exceeded and automatically resets the interrupt pin when the proximity reading falls beneath the PS low threshold.

An interrupt can be cleared by reading data out from the **INT_Flag register** (resetting it to "0") and the INT pin is then reset to high. For more details on the interrupt, please refer to the datasheet.

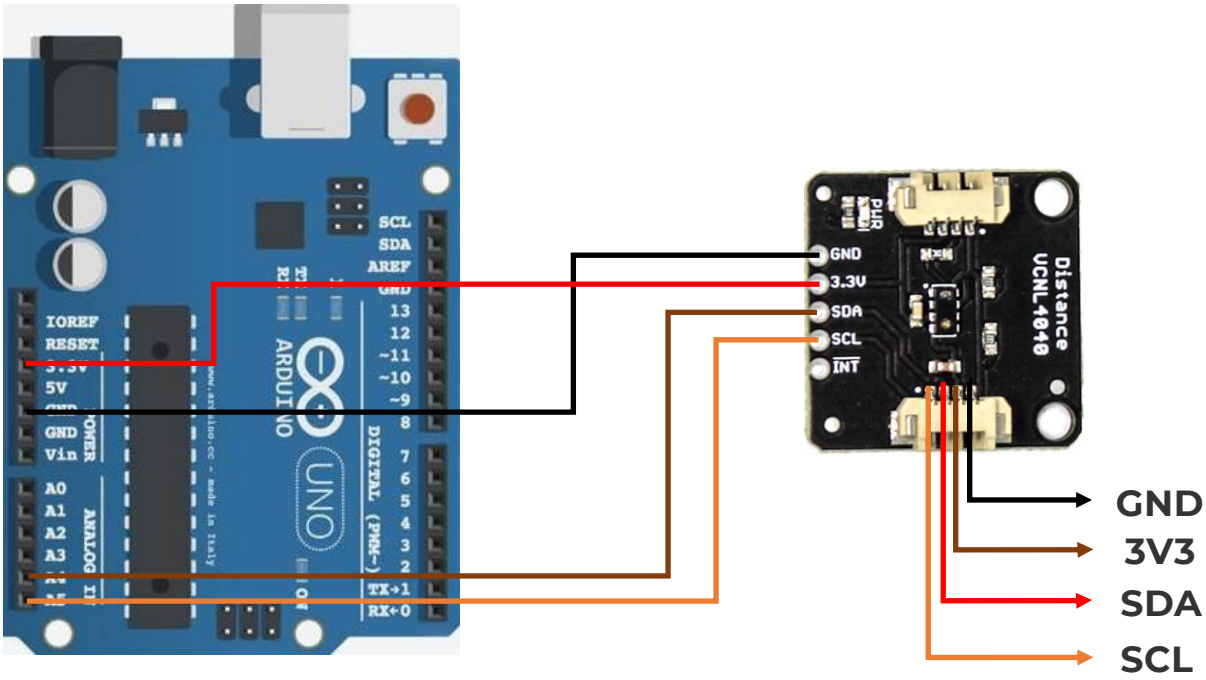
Interrupt Jumper

Cutting the **INT** jumper will remove the **10 kΩ** pull-up resistor from the INT pin.

I^2C Pull-up Jumper

Cutting the **I^2C** jumper will remove the **2.2 kΩ** pull-up resistors from the I^2C bus. If you have many devices on your I^2C bus you may want to remove these jumpers.

WIRING



Arduino	VCNL4040
A5(SCL)	SCL
A4(SDA)	SDA
3.3V	3V3
GND	GND

Arduino Library

Adafruit has written an Arduino library to test this sensor. You can grab it from the Arduino Library Manager by searching for **'Adafruit VCNL4040'**.

Example: vcnl4040_test

```
#include <Adafruit_VCNL4040.h>

Adafruit_VCNL4040 vcnl4040 = Adafruit_VCNL4040();

void setup() {
  Serial.begin(115200);
  // Wait until serial port is opened
  while (!Serial) { delay(1); }

  Serial.println("Adafruit VCNL4040 Config demo");

  if (!vcnl4040.begin()) {
    Serial.println("Couldn't find VCNL4040 chip");
    while (1);
  }
  Serial.println("Found VCNL4040 chip");

  //vcnl4040.setProximityLEDCurrent(VCNL4040_LED_CURRENT_200MA);
  Serial.print("Proximity LED current set to: ");
  switch(vcnl4040.getProximityLEDCurrent()) {
    case VCNL4040_LED_CURRENT_50MA: Serial.println("50 mA"); break;
    case VCNL4040_LED_CURRENT_75MA: Serial.println("75 mA"); break;
    case VCNL4040_LED_CURRENT_100MA: Serial.println("100 mA"); break;
    case VCNL4040_LED_CURRENT_120MA: Serial.println("120 mA"); break;
    case VCNL4040_LED_CURRENT_140MA: Serial.println("140 mA"); break;
    case VCNL4040_LED_CURRENT_160MA: Serial.println("160 mA"); break;
    case VCNL4040_LED_CURRENT_180MA: Serial.println("180 mA"); break;
    case VCNL4040_LED_CURRENT_200MA: Serial.println("200 mA"); break;
  }

  //vcnl4040.setProximityLEDDutyCycle(VCNL4040_LED_DUTY_1_40);
  Serial.print("Proximity LED duty cycle set to: ");
  switch(vcnl4040.getProximityLEDDutyCycle()) {
    case VCNL4040_LED_DUTY_1_40: Serial.println("1/40"); break;
    case VCNL4040_LED_DUTY_1_80: Serial.println("1/80"); break;
    case VCNL4040_LED_DUTY_1_160: Serial.println("1/160"); break;
    case VCNL4040_LED_DUTY_1_320: Serial.println("1/320"); break;
  }

  //vcnl4040.setAmbientIntegrationTime(VCNL4040_AMBIENT_INTEGRATION_TIME_80MS);
  Serial.print("Ambient light integration time set to: ");
  switch(vcnl4040.getAmbientIntegrationTime()) {
    case VCNL4040_AMBIENT_INTEGRATION_TIME_80MS: Serial.println("80 ms"); break;
    case VCNL4040_AMBIENT_INTEGRATION_TIME_160MS: Serial.println("160 ms"); break;
    case VCNL4040_AMBIENT_INTEGRATION_TIME_320MS: Serial.println("320 ms"); break;
```

```
    case VCNL4040_AMBIENT_INTEGRATION_TIME_640MS: Serial.println("640 ms"); break;
  }


  //vcnl4040.setProximityIntegrationTime(VCNL4040_PROXIMITY_INTEGRATION_TIME_8T);
  Serial.print("Proximity integration time set to: ");
  switch(vcnl4040.getProximityIntegrationTime()) {
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_1T: Serial.println("1T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_1_5T: Serial.println("1.5T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_2T: Serial.println("2T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_2_5T: Serial.println("2.5T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_3T: Serial.println("3T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_3_5T: Serial.println("3.5T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_4T: Serial.println("4T"); break;
    case VCNL4040_PROXIMITY_INTEGRATION_TIME_8T: Serial.println("8T"); break;
  }

  //vcnl4040.setProximityHighResolution(false);
  Serial.print("Proximity measurement high resolution? ");
  Serial.println(vcnl4040.getProximityHighResolution() ? "True" : "False");

  Serial.println("");

}

void loop() {

  Serial.print("Proximity:"); Serial.println(vcnl4040.getProximity());
  Serial.print("Ambient light:"); Serial.println(vcnl4040.getLux());
  Serial.print("Raw white light:"); Serial.println(vcnl4040.getWhiteLight());
  Serial.println("");

  delay(500);
}
////////////////////////////////////////END/////////////////////////////////////////
```

Despite being a qualitative sensor, you will notice that the output does vary with the distance of the object, surface color, and reflectivity. You can also use this setting to estimate the field of view of the sensor. The maximum output value is 65535, closest to the sensor, and the minimum (0-1) being the furthest the sensor can read.