



## Qwiic Proximity Sensor (VCNL4040) Hookup Guide

### Introduction

The Qwiic 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. If you need quantitative distance readings for exact distances, check out the SparkFun 2 meter and 4 meter Time of Flight (ToF) sensors with mm accuracy.



SparkFun Proximity Sensor Breakout - 20cm,  
VCNL4040 (Qwiic)

© SEN-15177

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 SparkFun Proximity VCNL4040 sensor is a great digital alternative to the popular analog PIR sensors. The Qwiic Proximity Sensor also has an ambient light sensor built in which is excellent if you need a digital light sensor for your next glitter cannon.

This board is one of our many Qwiic compatible boards! Simply plug and go. No soldering, no figuring out which is SDA or SCL, and no voltage regulation or translation required!

### Required Materials

The Qwiic Proximity Sensor does need a few additional items for you to get started. *You may already have a few of these items, so feel free to modify your cart as necessary.*



SparkFun RedBoard Qwiic

● DEV-15123

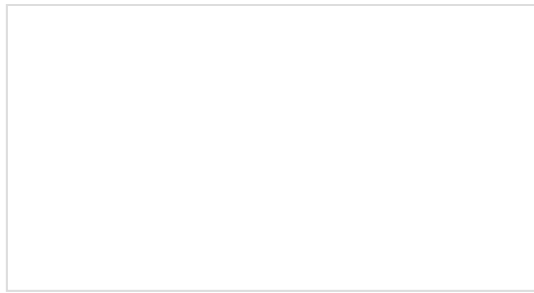


SparkFun Qwiic Cable Kit

● KIT-15081

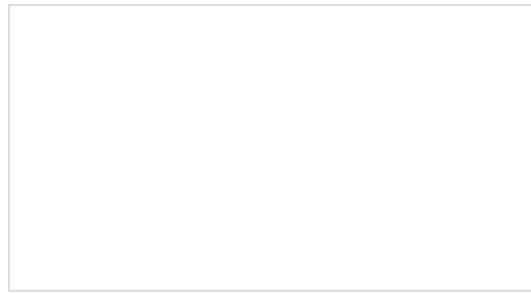
### Suggested Reading

If you're unfamiliar with jumper pads or I<sup>2</sup>C be sure to checkout some of these foundational tutorials.



## I2C

An introduction to I2C, one of the main embedded communications protocols in use today.



## How to Work with Jumper Pads and PCB Traces

Handling PCB jumper pads and traces is an essential skill. Learn how to cut a PCB trace, add a solder jumper between pads to reroute connections, and repair a trace with the green wire method if a trace is damaged.

If you aren't familiar with the Qwiic system, we recommend reading here for an overview.



*Qwiic Connect System*

## 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.

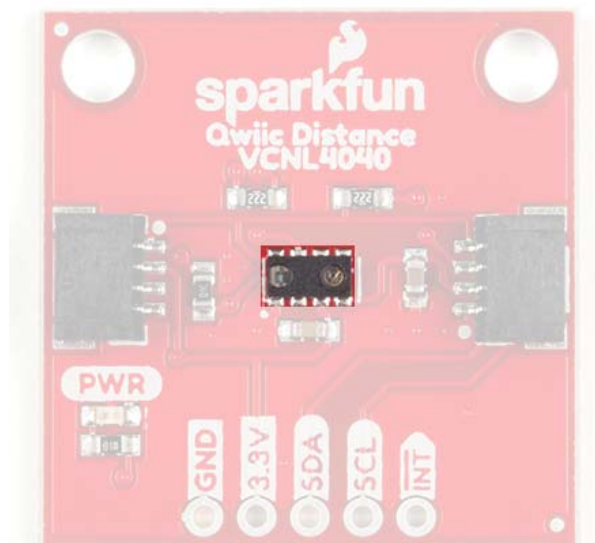
### Power LED

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



### 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 Qwiic connect system. 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). If you need quantitative distance readings, like exact distances, check out the SparkFun 2 meter and 4 meter Time of Flight (ToF) sensors with mm accuracy.

## IR Emitter

The IR emitter 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  $\pm 10\%$  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 Filtron™ 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.

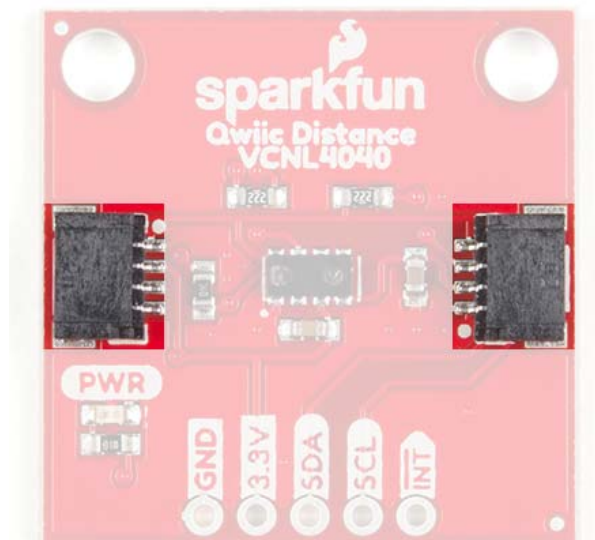
## Qwiic or I<sup>2</sup>C

### I<sup>2</sup>C Address

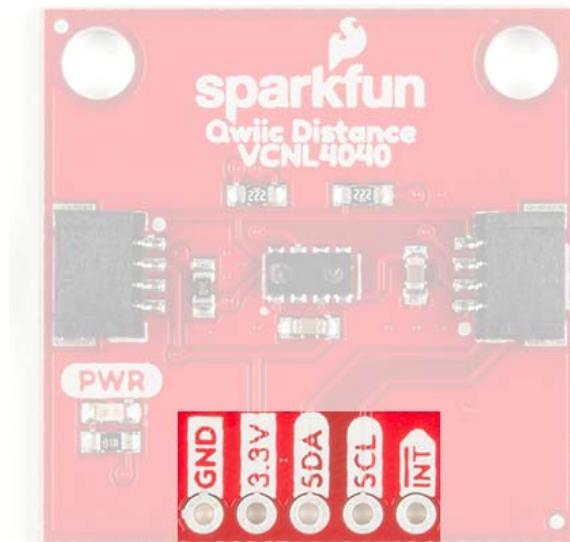
The sensor has a single slave address **0x60** (HEX) of 7-bit addressing, following I<sup>2</sup>C protocol.

## Connections

The simplest way to use the Qwiic Proximity Sensor is through the Qwiic connect system. The connectors are polarized for the I<sup>2</sup>C 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<sup>2</sup>C bus of your microcontroller and power pins (**3.3V** and **GND**), if it doesn't have a Qwiic connector. 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.



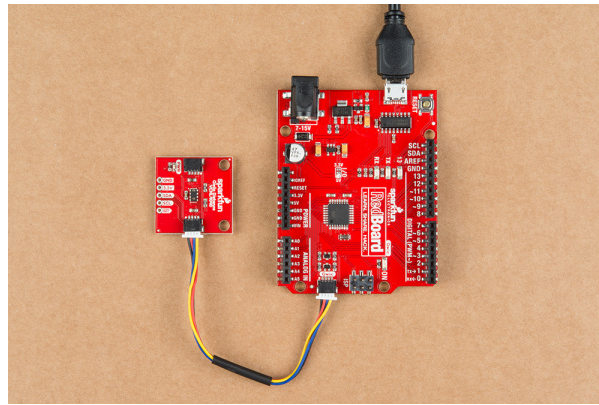
### I<sup>2</sup>C Pull-up Jumper

Cutting the I<sup>2</sup>C jumper will remove the **2.2 k $\Omega$**  pull-up resistors from the I<sup>2</sup>C bus. If you have many devices on your I<sup>2</sup>C bus you may want to remove these jumpers. Not sure how to cut a jumper? Read [here!](#)



### Hardware Assembly

With the Qwiic connector system, assembling the hardware is simple. All you need to do is connect your Qwiic Proximity Sensor to the RedBoard Qwiic with a Qwiic cable. Otherwise, you can use the I<sup>2</sup>C pins, if you don't have a Qwiic enabled microcontroller. Just be away of your input voltage and any logic level shifting you may need to do.



## Arduino Library

**Note:** This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. If you have not previously installed an Arduino library, please check out our installation guide.

We've written an Arduino library to flex every feature of this sensor. You can grab it from the Arduino Library Manager, from the GitHub repository, or use the download button below.

[DOWNLOAD THE VCNL4040 LIBRARY \(ZIP\)](#)

## Library Functions

The Arduino library is commented and the functions should be self-explanatory. However, below is a detailed list of the available library functions.

**Note:** It is recommended that users begin with the examples in the following section before diving head first into the library functions. The library examples demonstrate how to setup the more basic features of the VCNL4040 without having to dig through the datasheet. Once users have become familiar with the basic setups and have gone through the datasheet thoroughly, it should be easier to follow the library functions below.

- `.begin` - Check communication with sensor and set it to default I<sup>2</sup>C Wire library settings
- `.isConnected` - Checks connection
- `.setLEDCurrent` - Sets the IR LED sink current to one of 8 settings (75, 100, 120, 140, 160, 180, and 200 mA).
- `.setIRDutyCycle` - Sets duty cycle of the IR LED (40 max.)
- `.setProxInterruptPersistence` - Sets the Prox interrupt persistence value.  
The PS persistence function (PS\_PERS, 1, 2, 3, 4) helps to avoid false trigger of the PS INT. It defines the amount of consecutive hits needed in order for a PS interrupt event to be triggered.



`.setAmbientInterruptPersistence` - Sets the Ambient interrupt persistence value

The ALS persistence function (ALS\_PERS, 1, 2, 4, 8) helps to avoid false trigger of the ALS INT. It defines the amount of consecutive hits needed in order for a ALS interrupt event to be triggered.

`.setProxIntegrationTime` - Sets the integration time for the proximity sensor

`.setAmbientIntegrationTime` - Sets the integration time for the ambient light sensor

`.powerOnProximity` - Power on the prox sensing portion of the device

`.powerOffProximity` - Power off the prox sensing portion of the device

`.powerOnAmbient` - Power on the ALS sensing portion of the device

`.powerOffAmbient` - Power off the ALS sensing portion of the device

`.setProxResolution` - Sets the proximity resolution to 12 or 16 bit

`.enableAmbientInterrupts` - Enables ALS interrupt

`.disableAmbientInterrupts` - Disables ALS interrupt

`.enableSmartPersistence` - Turn on smart persistence

`.disableSmartPersistence` - Turn off smart persistence

To accelerate the PS response time, smart persistence prevents the misjudgment of proximity sensing, but also keeps a fast response time.

`.enableActiveForceMode` - Enables active force mode

`.disableActiveForceMode` - Disable active force mode

An extreme power saving way to use PS is to apply PS active force mode. Anytime host would like to request one proximity measurement, enable the active force mode. This triggers a single PS measurement, which can be read from the PS result registers. VCNL4040 stays in standby mode constantly.

`.takeSingleProxMeasurement` - Sets trigger bit so sensor takes a force mode measurement and returns to standby.

`.enableWhiteChannel` - Enables the white measurement channel

`.disableWhiteChannel` - Disables the white measurement channel

`.enableProxLogicMode` - Enables the proximity detection logic output mode

`.disableProxLogicMode` - Disables the proximity detection logic output mode

When this mode is selected, the INT pin is pulled low when an object is close to the sensor (value is above high threshold) and is reset to high when the object moves away (value is below low threshold). Register: PS\_THDH / PS\_THDL define where these threshold levels are set.

`.setProxCancellation` - Sets the proximity sensing cancellation value - helps reduce cross talk with ambient light

`.setALSHighThreshold` - Value that ALS must go above to trigger an interrupt

`.setALSLowThreshold` - Value that ALS must go below to trigger an interrupt

`.setProxHighThreshold` - Value that Proximity Sensing must go above to trigger an interrupt

`.setProxLowThreshold` - Value that Proximity Sensing must go below to trigger an interrupt

- .getProximity - Read the Proximity value
- .getAmbient - Read the Ambient light value
- .getWhite - Read the White light value
- .getID - Read the sensors ID
- .setProxInterruptType - Enable four proximity interrupt types
  - .isClose - Interrupt flag: True if prox value greater than high threshold
  - .isAway - Interrupt flag: True if prox value lower than low threshold
  - .isLight - Interrupt flag: True if ALS value higher than high threshold
  - .isDark - Interrupt flag: True if ALS value lower than low threshold
- .readCommand - Reads two consecutive bytes from a given 'command code' location
- .writeCommand - Write two bytes to a given command code location (8 bits)

## Arduino Examples

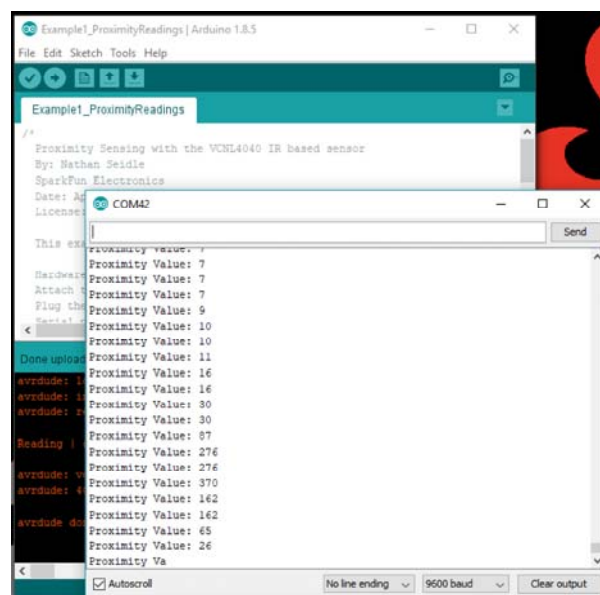
**Note:** This section is an example of using the Qwiic Proximity Sensor and the RedBoard Qwiic with the Arduino IDE. It is not intended to be a guide for using I<sup>2</sup>C devices with the Arduino IDE.

Please use the following links and the internet for a better understanding of I<sup>2</sup>C and how it works in the Arduino IDE:

- A tutorial on I<sup>2</sup>C.
- An in-depth overview of the Wire (I<sup>2</sup>C) library.

### Example 1: Proximity Readings

This example outputs the IR Proximity Value from the VCNL4040 sensor in the Serial Monitor.





```

Example3_AmbientLight | Arduino 1.8.5
File Edit Sketch Tools Help
Example3_AmbientLight
/*
  Proximity Sensing with the VCNL4040 IR based sensor
  By: Nathan Seidle
  SparkFun Electronics
  Date: April 2015
  License: CC BY-NC-ND 4.0
*/
//
// This example code is in the public domain.
//
// Point to the sensor at a light source.
// The read() function will return the ambient light level.

// Setup
//-----
void setup() {
  // Serial port for debugging and data
  Serial.begin(9600);
}

// Loop
//-----
void loop() {
  // Read the ambient light level
  int ambientLightLevel = read();

  // Print the ambient light level
  Serial.println(ambientLightLevel);
}

```

Ambient light level: 845  
 Ambient light level: 845  
 Ambient light level: 834  
 Ambient light level: 834  
 Ambient light level: 784  
 Ambient light level: 784  
 Ambient light level: 771  
 Ambient light level: 771  
 Ambient light level: 822  
 Ambient light level: 822  
 Ambient light level: 836  
 Ambient light level: 836  
 Ambient light level: 835  
 Ambient light level: 835  
 Ambient light level: 786  
 Ambient light level: 786  
 Ambient light level: 756  
 Ambient light level: 756  
 Ambient light level: 823  
 Ambient light level: 823

Again the sensor is qualitative, but you can test how the angle of incident from the light source affects the sensor readings.

#### Example 4: All Readings

This example outputs IR, ambient and white light readings to the terminal. Along with proximity and ambient light sensing the VCNL4040 has a 'white light' sensor as well.

```

Example4_AllReadings | Arduino 1.8.5
File Edit Sketch Tools Help
Example4_AllReadings
/*
  Proximity Sensing with the VCNL4040 IR based sensor
  By: Nathan Seidle
  SparkFun Electronics
  Date: April 2015
  License: CC BY-NC-ND 4.0
*/
//
// This example code is in the public domain.
//
// Along with proximity and ambient light sensing, the VCNL4040 has a 'white light' sensor as well.
// The read() function will return a struct containing the proximity, ambient light, and white light readings.
// The read() function will return the ambient light level.

// Setup
//-----
void setup() {
  // Serial port for debugging and data
  Serial.begin(9600);
}

// Loop
//-----
void loop() {
  // Read the sensor data
  ProxData_t proxData = read();

  // Print the sensor data
  Serial.println("Prox value[" + String(proxData.proxValue) + "] Ambient light level[" + String(proxData.ambientLightLevel) + "] White level[" + String(proxData.whiteLevel) + "]");
}

```

Prox value[5] Ambient light level[624] White level[802]  
 Prox value[5] Ambient light level[712] White level[901]  
 Prox value[4] Ambient light level[787] White level[985]  
 Prox value[4] Ambient light level[879] White level[1102]  
 Prox value[5] Ambient light level[848] White level[1056]  
 Prox value[7] Ambient light level[745] White level[924]  
 Prox value[15] Ambient light level[471] White level[622]  
 Prox value[20] Ambient light level[259] White level[342]  
 Prox value[19] Ambient light level[204] White level[379]  
 Prox value[10] Ambient light level[498] White level[443]  
 Prox value[6] Ambient light level[728] White level[920]  
 Prox value[5] Ambient light level[782] White level[984]  
 Prox value[5] Ambient light level[773] White level[971]  
 Prox value[4] Ambient light level[767] White level[960]  
 Prox value[5] Ambient light level[766] White level[959]  
 Prox value[5] Ambient light level[802] White level[994]  
 Prox value[5] Ambient light level[830] White level[1039]  
 Prox value[4] Ambient light level[823] White level[1028]  
 Prox value[5] Ambient light level[749] White level[933]  
 Prox value[6] Ambient light level

#### Example 5: Advanced Settings

This example shows how to use different Wire ports, fast I<sup>2</sup>C, and various advanced settings that are supported by the library. Please, refer to the datasheet and library functions listed above for more details.

## Resources and Going Further

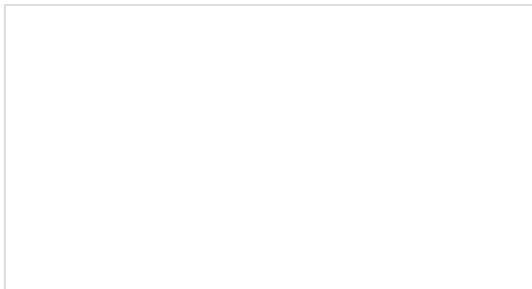
For more information, check out the resources below:

- [GitHub Product Repo](#)
- [Schematic \(PDF\)](#)
- [Eagle Files \(ZIP\)](#)
- [VCNL4040 Datasheet](#)
- [SparkFun VCNL4040 Arduino Library \(ZIP\)](#)
- [SparkFun VCNL4040 Arduino Library GitHub Repository](#)
- [SFE Product Showcase](#)

Need help getting started with Arduino and I<sup>2</sup>C? Check out these resources here:

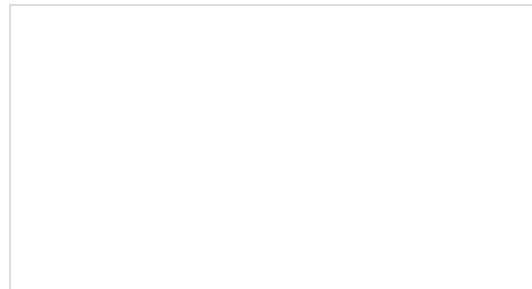
- [Arduino I<sup>2</sup>C Scanner Example](#)
- [Arduino Wire Library Reference Page](#)
- [Arduino Wire Library \(In-Depth\) Reference](#)

Need some inspiration for your next project? Check out some of these other Qwiic product tutorials:



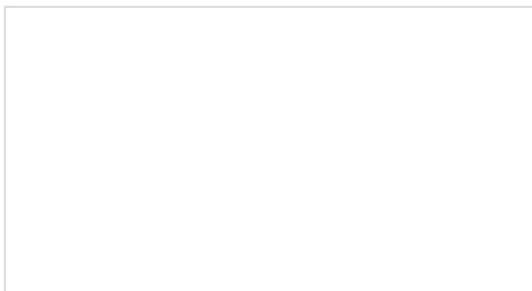
### Using the PSoC 6 Pioneer Board with the Pioneer IoT Add-on Shield

Cypress's PSoC 6 Pioneer Board is a development tool for the powerful PSoC 6 processor. In this tutorial, we'll show you how to use the PSoC 6 Pioneer Board along with the SparkFun/Digi-Key Pioneer IoT Add-on Shield to send data to a Raspberry Pi via BLE or WiFi.



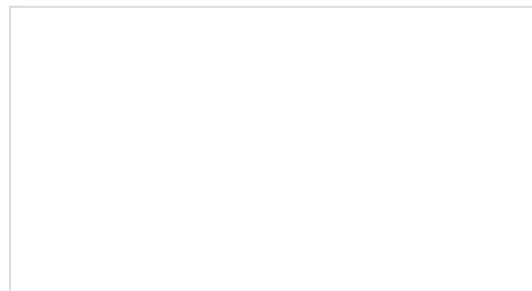
### Qwiic Distance Sensor (RFD77402) Hookup Guide

The RFD77402 uses an infrared VCSEL (Vertical Cavity Surface Emitting Laser) TOF (Time of Flight) module capable of millimeter precision distance readings up to 2 meters. It's also part of SparkFun's Qwiic system, so you won't have to do any soldering to figure out how far away things are.



### LTE Cat M1/NB-IoT Shield Hookup Guide

The SparkFun LTE CAT M1/NB-IoT Shield equips your Arduino or Arduino-compatible microcontroller with access to data networks



### TFMini - Micro LiDAR Module (Qwiic) Hookup Guide

The TFMMini is a ToF (Time of Flight) LiDAR sensor capable of measuring the distance to an

across the globe. This shield adds wireless, high-bandwidth cellular functionality to your IoT project!

object as close as 30 cm and as far as 12 meters! The TFMMini allows you to easily integrate LiDAR into applications traditionally reserved for smaller sensors such as the SHARP GP-series infrared rangefinders. With the added Qwiic feature, you can quickly connect to the sensor via I2C!