TFMini - Micro LiDAR Module (Qwiic) Hookup Guide

Introduction

Heads up! This tutorial is for the **Qwiic** enabled TFMini. Serial data is output via I^{2}C. If you are using the TFMini that outputs serial data via UART [SEN-14588], please refer to the TFMini - Micro LiDAR Module Hookup Guide.

The TFMini is a ToF (Time of Flight) LiDAR sensor capable of measuring the distance to an object as close as 30 cm and as far as 12 meters! The TFMini allows you to 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! In this tutorial, you will learn how to connect to the TFMini using an Arduino microcontroller with the Qwiic system.
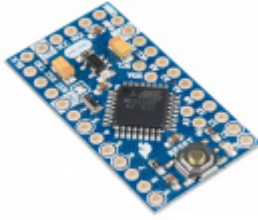

TFMini - Micro LiDAR Module (Qwiic)
◉ SEN-14786

Required Materials

To follow along with this tutorial, you will need the following materials. You may not need everything though depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary.

Arduino Pro Mini 328 - 3.3V/8MHz
◎ DEV-11114

Break Away Headers - Straight
◎ PRT-00116

SparkFun Beefy 3 - FTDI Basic Breakout
◎ DEV-13746

USB micro-B Cable - 6 Foot
◎ CAB-10215

Qwiic Cable - Breadboard Jumper (4-pin)
◎ PRT-14425

SparkFun Qwiic Adapter
◎ DEV-14495

Tip: You could also use SparkX BlackBoard. It's basically a RedBoard with additional features. The board includes the same AP2112K 3.3V voltage regulator that is populated on the Beefy 3 and Qwiic connector on the board. This would reduce the amount of components and time soldering headers to the board. Additionally, a USB port can provide up to about 500mA so the TFMini may want to pull more power when auto ranging for longer distances.

Keep in mind that SparkX products are rapidly produced to bring you the most cutting edge technology as it becomes available. SparkX products are tested but come with no guarantees. Live technical support is not available for SparkX products.

SparkFun BlackBoard
◉ SPX-14669

Tools

Depending on your setup, you will need a soldering iron, solder, and general soldering accessories.

Solder Lead Free - 100-gram Spool
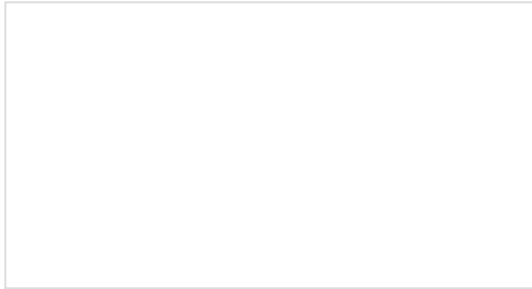◉ TOL-09325

Weller WLC100 Soldering Station
◉ TOL-14228

Suggested Reading

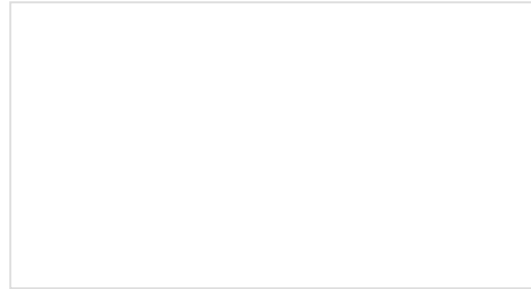If you aren't familiar with the Qwiic system, we recommend reading here for an overview.

Qwiic Connect System

We would also recommend taking a look at the following tutorials if you aren't familiar with them.
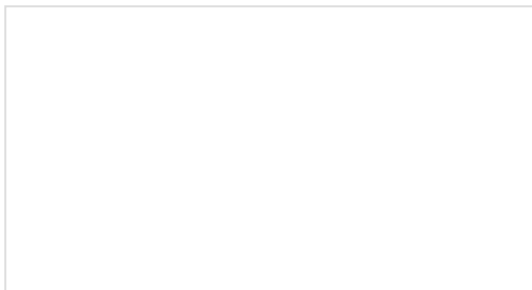
Logic Levels
Learn the difference between 3.3V and 5V devices and logic levels.

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

Using the Arduino Pro Mini 3.3V
This tutorial is your guide to all things Arduino Pro Mini. It explains what it is, what it's not, and how to get started using it.

Hardware Overview

Note: This product does not use laser light for ranging. Instead it contains an LED and optics. Many such systems are being marketed under the name "LiDAR," although it may be more appropriate to think of this device as a "Time-of-Flight Infrared Rangefinder". It differs significantly from traditional IR rangefinders in that it uses ToF to determine range and not triangulation — as is performed by the Sharp GP-series devices.

The sensor works by sending a modulated near-infrared light out. The light that is reflected from the object returns to the sensor's receiver. The distance between the two can be converted using the sensor by calculating the time and phase difference. The distance measured may vary depending on the environment and the reflectivity of object.

Input Power

According to the datasheet (pg 1), the input voltage is **5V**. In this tutorial, we will be using the included boost converter by applying a 3.3V input voltage from the Qwiic side to boost power to 5V on the TFMini side.
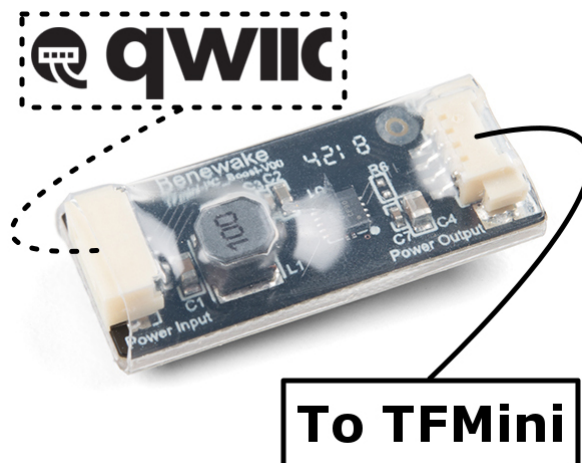
Current Draw: According to the datasheet, TFMini may pull up to **~800mA** at peak current. You may want to consider providing a sufficient power supply when using the sensor in a project.

⚡ **Warning:** Since the Qwiic system uses 3.3V for power, the Qwiic enabled TFmini with boost circuit can exceed the maximum output rating of the 3.3V voltage regulator on the RedBoard powered by Arduino. This causes 3.3V voltage regulator to enter thermal shutdown and the RedBoard to restart. As a result, you will not be able to read the TFMini's output on an Arduino serial monitor.
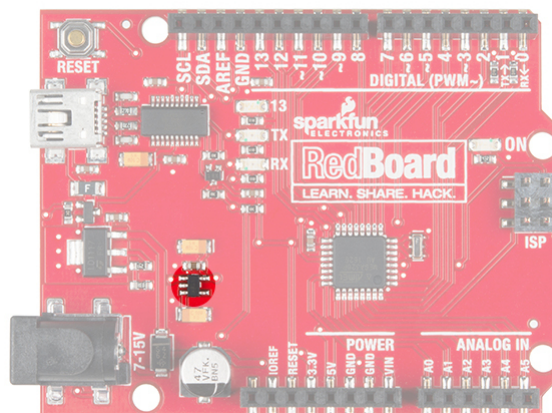

Make sure to use either the Beefy 3 with a 3.3V Arduino Pro Mini, BlackBoard, or the LD1117V33 3.3V voltage regulator to provide sufficient current for the TFMini.

Logic Levels

While the sensor can be powered at 5V, the I2C pins are only **3.3V logic**. Make sure to use a logic level converter when reading the sensor with a 5V microcontroller.

Pinout

There is a marking next to the polarized connector to indicate the polarity as "**J1**" as indicated in the image below. This is useful when referencing sensor's pinout. As opposed to the original TFMini, the green and white wires for the Qwiic enabled TFMini uses an I^2C serial. The default address of the TFMini is **0x10**.
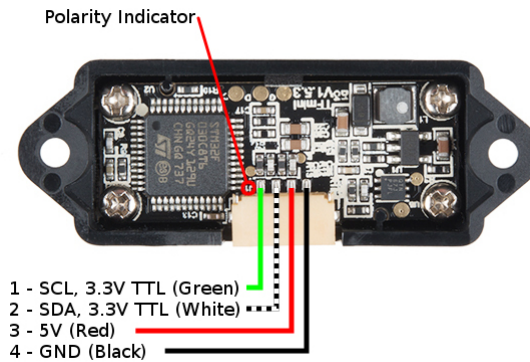

| Pin Number	Wire Color	Qwiic TFMini Pinout	Wire Color
1	Green	SCL (**3.3V TTL**)	Yellow
2	White	SDA (**3.3V TTL**)	Blue
3	Red	**5V**	Red
4	Black	GND	Black

Hardware Hookup

If you haven't yet assembled your 3.3V Pro Mini, now would be the time to head on over to that tutorial to solder the header pins. Once soldered, connect the power and I^2C pins between the Arduino Pro Mini 3.3V/8MHz and Qwiic adapter.

Arduino Pins	Qwiic Adapter Pins	Wire Color
A5 (SCL)	SCL (**3.3V TTL**)	Yellow
A4 (SDA)	SDA (**3.3V TTL**)	Blue
3.3V	**3.3V**	Red
GND	GND	Black

Then connect the Qwiic cable that was included in the **Power Input** side between the Qwiic adapter and the boost circuit. On the other side, insert the second cable that was included in the **Power Output** side between from the boost circuit to the TFMini. The connectors are different on each side of the boost converter so it should not be Finally, power the circuit up with a micro-B USB cable and the Beefy 3. The connection should look like the image below.
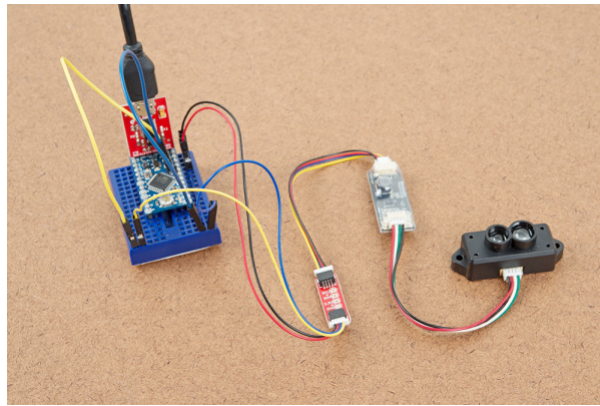
> **Tip:** Looking to reduce the number of components? Try using the SparkX BlackBoard is an alternative to connecting to the Qwiic enabled TFMini. The trade off is that the BlackBoard is larger than the Arduino Pro Mini.
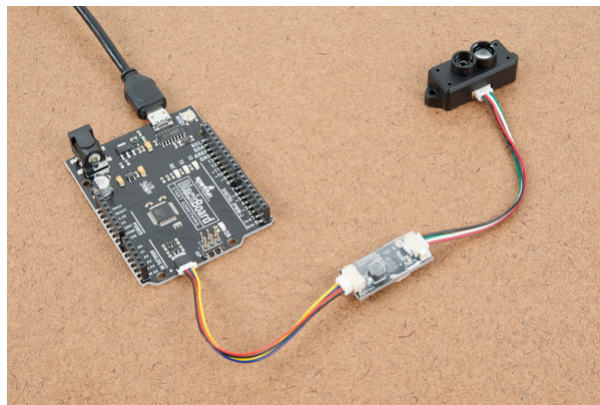
Example Code

> **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.

Grab a micro-B USB cable and connect the Arduino to your computer. Copy, paste, and upload the code below. Make sure to use the correct COM port and board selection.

```
/*
  LidarTest.ino
  Written in collaboration by Nate Seidle and Benewake

  Example sketch for the Qwiic Enabled TFMini
  (https://www.sparkfun.com/products/14786)
*/

#include <Wire.h>

uint16_t distance = 0; //distance
uint16_t strength = 0; // signal strength
uint8_t rangeType = 0; //range scale
/*Value range:
 00 (short distance)
 03 (intermediate distance)
 07 (long distance) */

boolean valid_data = false; //ignore invalid ranging data

const byte sensor1 = 0x10; //TFMini I2C Address

void setup()
{
  Wire.begin();

  Serial.begin(115200);
  Serial.println("TFMini I2C Test");
}


void loop()
{
  if (readDistance(sensor1) == true)
  {
    if (valid_data == true) {
      Serial.print("\tDist[");
      Serial.print(distance);
      Serial.print("]\tstrength[");
      Serial.print(strength);
      Serial.print("]\tmode[");
      Serial.print(rangeType);
      Serial.print("]");
      Serial.println();
    }
    else {
      //don't print invalid data
    }
  }
  else {
    Serial.println("Read fail");
  }
  delay(50); //Delay small amount between readings
}
```

```cpp
//Write two bytes to a spot
boolean readDistance(uint8_t deviceAddress)
{
  Wire.beginTransmission(deviceAddress);
  Wire.write(0x01); //MSB
  Wire.write(0x02); //LSB
  Wire.write(7); //Data length: 7 bytes for distance data
  if (Wire.endTransmission(false) != 0) {
    return (false); //Sensor did not ACK
  }
  Wire.requestFrom(deviceAddress, (uint8_t)7); //Ask for 7 bytes

  if (Wire.available())
  {
    for (uint8_t x = 0 ; x < 7 ; x++)
    {
      uint8_t incoming = Wire.read();

      if (x == 0)
      {
        //Trigger done
        if (incoming == 0x00)
        {
          //Serial.print("Data not valid: ");//for debugging
          valid_data = false;
          //return(false);
        }
        else if (incoming == 0x01)
        {
          Serial.print("Data valid:    ");
          valid_data = true;
        }
      }
      else if (x == 2)
        distance = incoming; //LSB of the distance value "Dist_L"
      else if (x == 3)
        distance |= incoming << 8; //MSB of the distance value "Dist_H"
      else if (x == 4)
        strength = incoming; //LSB of signal strength value
      else if (x == 5)
        strength |= incoming << 8; //MSB of signal strength value
      else if (x == 6)
        rangeType = incoming; //range scale
    }
  }
  else
  {
    Serial.println("No wire data avail");
    return (false);
  }

  return (true);
}
```

Once uploaded, try moving an object in front of the sensor to test. In the example below, a third hand was used to hold the TFMini when detecting an object at a certain distance away from the sensor. Since the sensor is not able to detect an object when less than 11.8 inches (or 30cm = 0.3m) away, the object under test was placed at 20 inches and 30 inches.

Qwiic Enabled TFMini Reading an Object at 20 Inches

Qwiic Enabled TFMini Reading an Object at 30 Inches
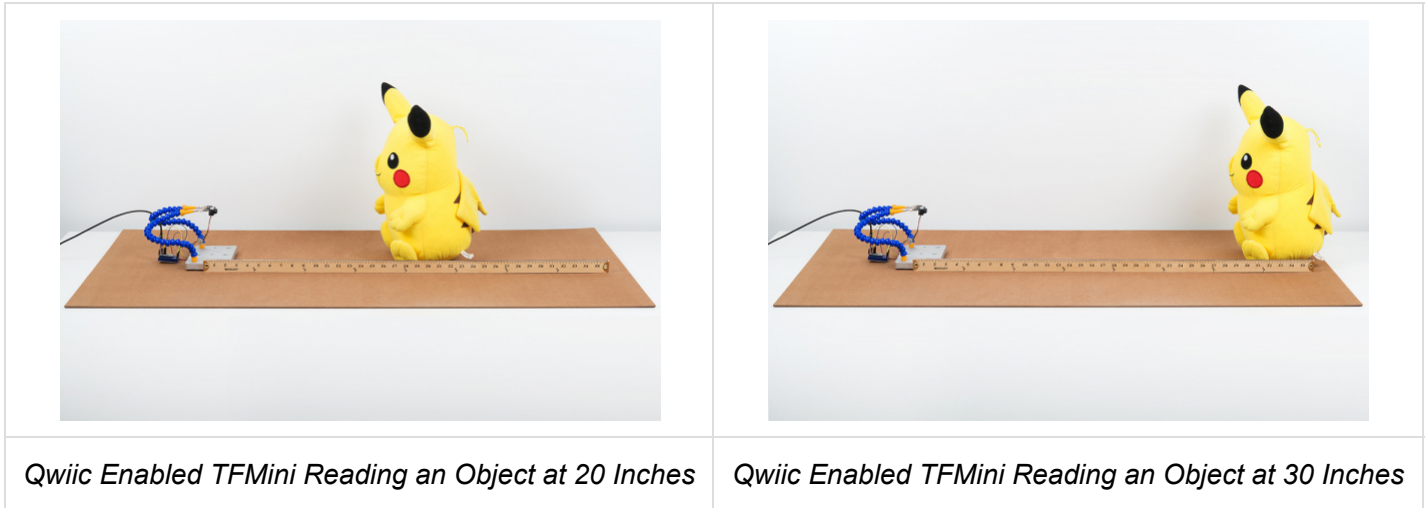
Opening the serial monitor at **115200**, you may see an output similar to the values printed below. Using a yard stick, the values responded as expected when moving an object between 30 inches and 20 inches. The Qwiic enabled TFMini indicated that the object was at an "intermediate distance".

```
TFMini I2C Test
Data valid: Dist[72]   strength[274]   mode[3]
Data valid: Dist[72]   strength[275]   mode[3]
Data valid: Dist[73]   strength[267]   mode[3]
Data valid: Dist[72]   strength[265]   mode[3]
Data valid: Dist[71]   strength[275]   mode[3]
Data valid: Dist[70]   strength[284]   mode[3]
Data valid: Dist[68]   strength[305]   mode[3]
Data valid: Dist[67]   strength[311]   mode[3]
Data valid: Dist[65]   strength[329]   mode[3]
Data valid: Dist[65]   strength[335]   mode[3]
Data valid: Dist[65]   strength[341]   mode[3]
Data valid: Dist[65]   strength[361]   mode[3]
Data valid: Dist[64]   strength[367]   mode[3]
Data valid: Dist[64]   strength[383]   mode[3]
Data valid: Dist[64]   strength[387]   mode[3]
Data valid: Dist[64]   strength[386]   mode[3]
Data valid: Dist[64]   strength[385]   mode[3]
Data valid: Dist[64]   strength[384]   mode[3]
Data valid: Dist[64]   strength[386]   mode[3]
Data valid: Dist[64]   strength[394]   mode[3]
Data valid: Dist[64]   strength[398]   mode[3]
Data valid: Dist[64]   strength[400]   mode[3]
Data valid: Dist[64]   strength[402]   mode[3]
```

# Resources and Going Further

Now that you've successfully got your Quick enabled TFMini up and running, it's time to incorporate it into your own project! For more on the TFMini, check out the links below:
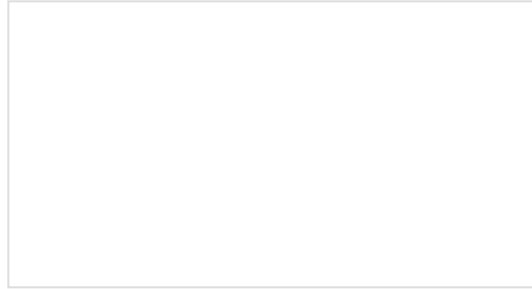
- Datasheet (PDF)
- BeneWake
    - Frequently Asked Questions
    - TFMini Downloads

Need some inspiration for your next project? Check out some of these related tutorials:
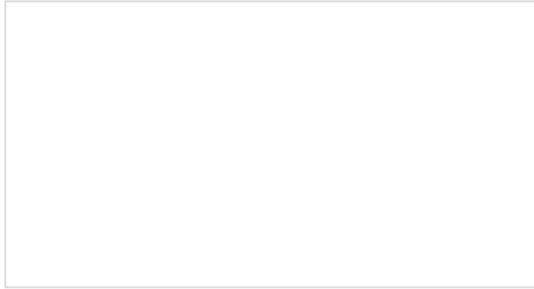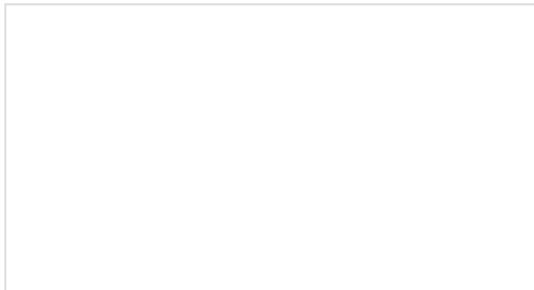
## VL6180 Hookup Guide
Get started with your VL6180 based sensor or the VL6180 breakout board.

## Building an Autonomous Vehicle: The Batmobile
Documenting a six-month project to race autonomous Power Wheels at the SparkFun Autonomous Vehicle Competition (AVC) in 2016.

## LIDAR-Lite v3 Hookup Guide
A tutorial for connecting the Garmin LIDAR-Lite v3 or the LIDAR-Lite v3HP to an Arduino to measure distance.
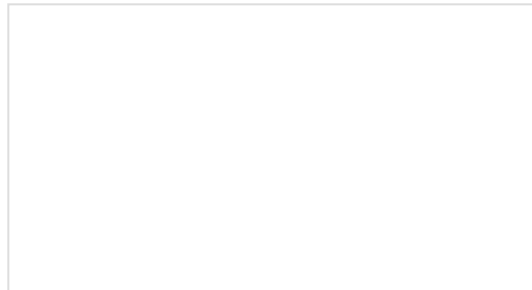
## Qwiic Distance Sensor (RFD77402) Hookup Guide
The RFD77402 uses an infrared VCSEL (Vertical Cavity Surface Emitting Laser) TOF (Time of Flight) module capable of millimeter precision distance readings up to 2 meters. It's also part of SparkFun's Qwiic system, so you won't have to do any soldering to figure out how far away things are.