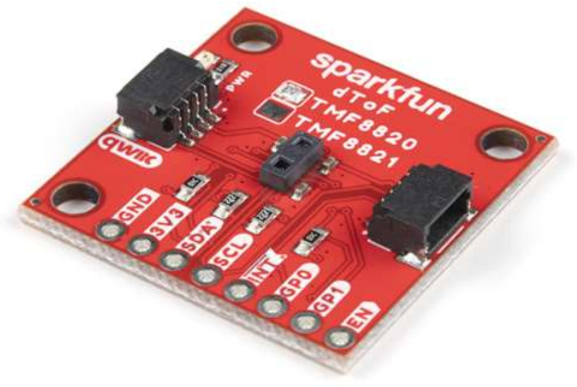
Qwiic dToF Imager (TMF882X) Hookup Guide

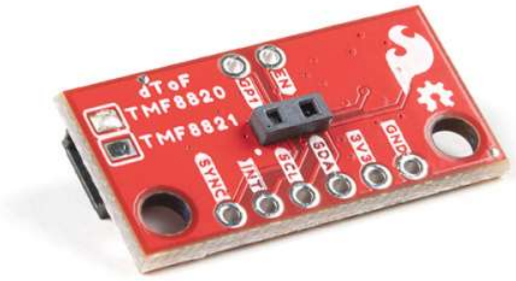
Introduction

The SparkFun Qwiic dToF Imager - TMF8820/TMF8821 are a direct time-of-flight (dToF) sensors that includes single modular package with associated Vertical Cavity Surface Emitting Laser (VCSEL) from AMS. The dToF device is based on Single Photon Avalanche Photodiode (SPAD), time-to-digital converter (TDC) and histogram technology and achieves 5000 mm detection range. Due to its lens on the SPAD, the TMF8820 supports 3x3 multizone output data while the TMF8821 supports 3x3, 4x4, and 3x6 multizone output data. The lens on each dToF Imager provides a very wide, dynamically adjustable, field of view. A multi-lens-array (MLA) inside the package above the VCSEL widens up the FoI (field of illumination). All processing of the raw data is performed on-chip and the TMF8820/TMF8821 provide distance information together with confidence values on its I^{2}C interface. The high performance on-chip optical filter blocks most of the ambient light, and enables distance measurements in dark and sunlight environments.

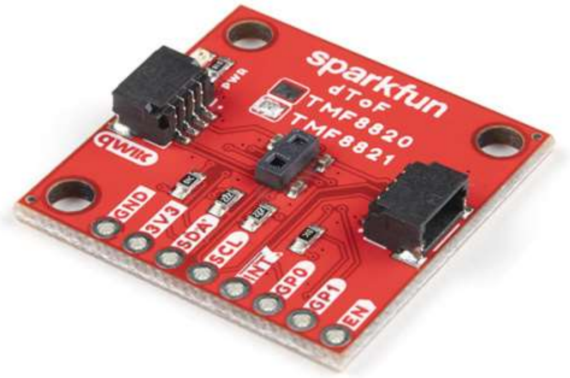
These sensors are great for projects that such as distance measurement for camera autofocus - Laser Detect Autofocus - LDAF (mobile phone), presence detection (computing and communication), object detection and collision avoidance (robotics), and light curtain (industrial).



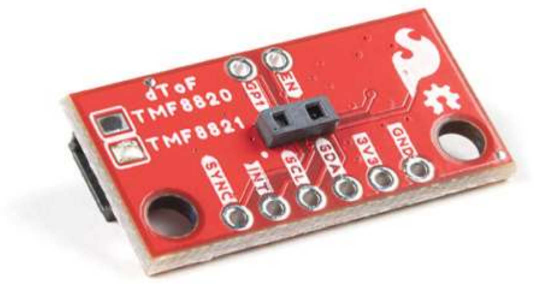
SparkFun Qwiic dToF Imager - TMF8820
◉ SEN-19036



SparkFun Qwiic Mini dToF Imager - TMF8820
◉ SEN-19218



SparkFun Qwiic dToF Imager - TMF8821
◉ SEN-19037



SparkFun Qwiic Mini dToF Imager - TMF8821
SEN-19451

Required Materials

Note: We recommend a microcontroller with enough flash to run your program code. Sorry, Uno's (or any development board using the ATmega328P) are out. But didn't you want an excuse to try out something new? We recommend choosing either an Artemis Thing Plus or an ESP32 Thing Plus board as your development board.

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. Note that the following wishlist includes the RedBoard Artemis and the TMF8821. Depending on your application, you can adjust the cart for a different processor board or sensor version.

Qwiic dToF Imager TMF8821 Wishlist SparkFun Wish List	
	SparkFun Qwiic dToF Imager - TMF8821 SEN-19037
	Qwiic Cable - 50mm PRT-14426
	SparkFun RedBoard Artemis DEV-15444
	Reversible USB A to C Cable - 2m CAB-15424

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.

Hookup Guide for the SparkFun RedBoard Artemis

Get started with the RedBoard Artemis - all the functionality of the SparkFun Artemis module wrapped in the familiar Uno R3 footprint

Installing Board Definitions in the Arduino IDE

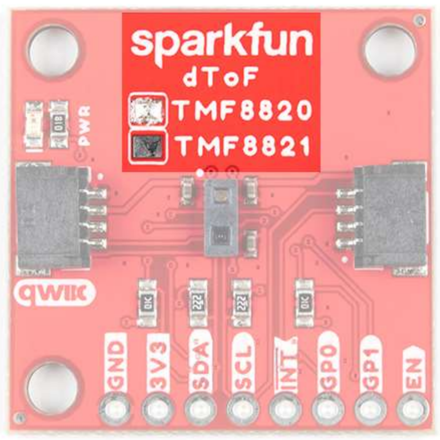
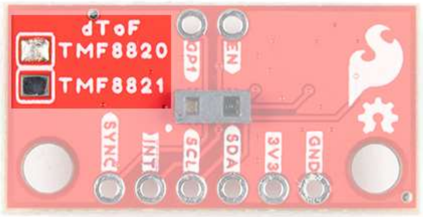
How do I install a custom Arduino board/core? It's easy! This tutorial will go over how to install an Arduino board definition using the Arduino Board Manager. We will also go over manually installing third-party cores, such as the board definitions required for many of the SparkFun development boards.

Hardware Overview

First, let's check out some of the characteristics of the TMF8820 and TMF8821 we're dealing with, so we know what to expect out of the board. Below is a comparison table for both sensors taken from the datasheet. Both are the pretty much the same except for the zone operation. Typically, the each board is powered at **3.3V** via the Qwiic connector.

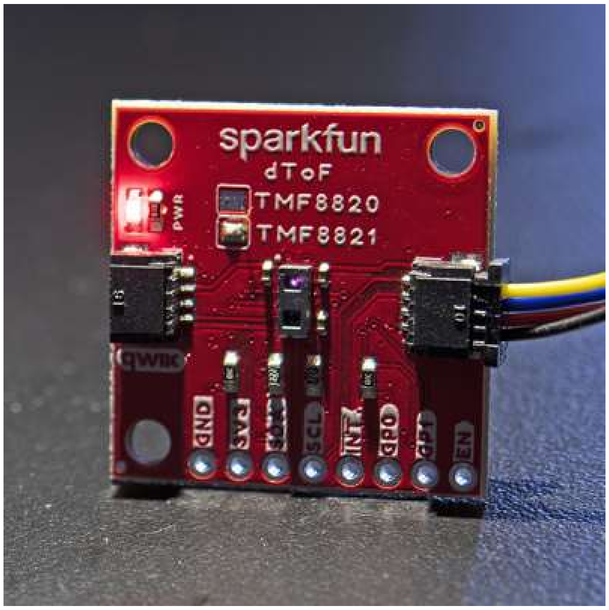
Characteristic	**TMF8820**	**TMF8821**
Operating Voltage	2.7V to 3.6V, **typically 3.3V via Qwiic Connector**	
I/O Voltage	1.62V to 3.3V, typically 3.3V via Qwiic Connector	
Current Consumption (Standby)	8µA	
Current Consumption (Active)	57mA	
Measurement Range	10mm to 5000mm, better accuracy detects reliably closest object	
Zone Operation	3x3	3x3, 4x4 and 3x6
Light Source	Class 1 940nm VCSEL	
I^{2}C Address	0x41	
Field of View	up to 63°	
Max Read Rate	up to 30 Hz	
Operating Temperature	-30°C to 70°C	

The layout for both the Qwiic dToF Imagers standard and mini sizes are the same. The only difference is the IC that is populated on the boards. The boards can be distinguished by the solder blob on the top side of the board. Below shows the image of the TMF8820 populated boards for the standard and mini size.

	
Qwiic dToF Imager - TMF8820/TMF8821 - Label	*Qwiic Mini dToF Imager - TMF8820/TMF8821 - Label*

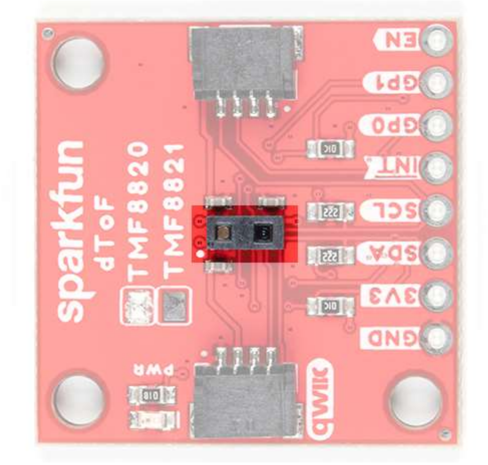
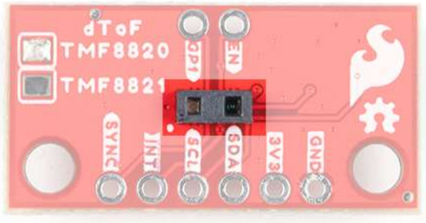
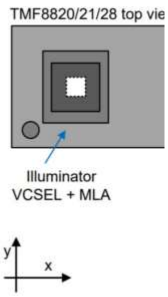
Illuminator and Receiver

The TMF8820 and TMF8821 consists of an illuminator (VCSEL + MLA) and receiver (lens + SPADs). The illuminator emits a infrared laser at a frequency of 940nm. By taking a smartphone camera or DSLR out, you should be able to see the IR through the illuminator by aiming the camera at an angle when the board is powered and running the example code! The internal processor (ARM M0+ ®) executes the ams algorithm to calculate the target distance of the object.

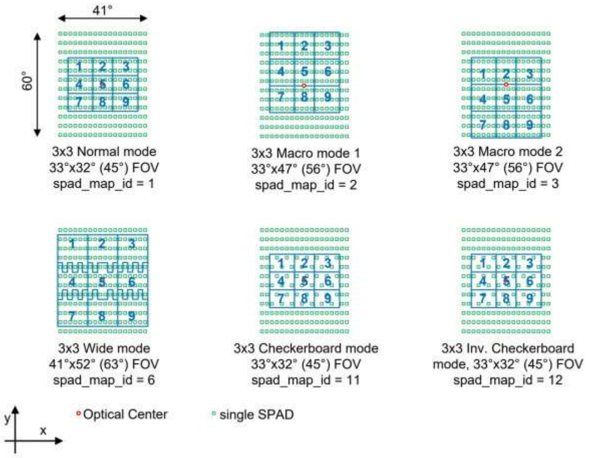


> **Note:** CLASS 1 LASER PRODUCT CLASSIFIED IEC 60825-1 2014.

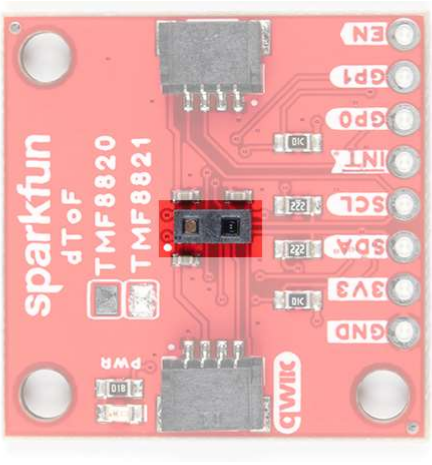
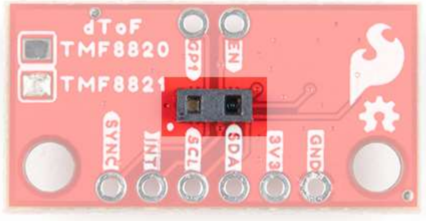
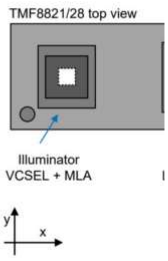
The orientation of the TMF8820/8821 IC's illuminator (VCSEL + MLA) and receiver (lens + SPAD) can be referenced by the IC's polarity marker. You'll need to orient the board based on the marker for your project's needs. While the image below shows the TMF8820, this applies to both the TMF8820 and TMF8821 when using a 3x3 SPAD. The TMF8820 is limited to only a 3x3 SPAD.

 EN GP1 GP0 INT SCL SDA 3V3 GND sparkfun dToF TMF8820 TMF8821 PWR qwiic	 dToF TMF8820 TMF8821 GP1 EN SYNC INT SCL SDA 3V3 GND	 TMF8820/21/28 top vie Illuminator VCSEL + MLA y x
TMF8820 - IC	*TMF8820 - IC*	*(3*

Below are SPAD map configurations for the 3x3 modes taken from the datasheet. Note that the datasheet recommends that users use the checkerboard SPAD masks for high ambient light conditions.



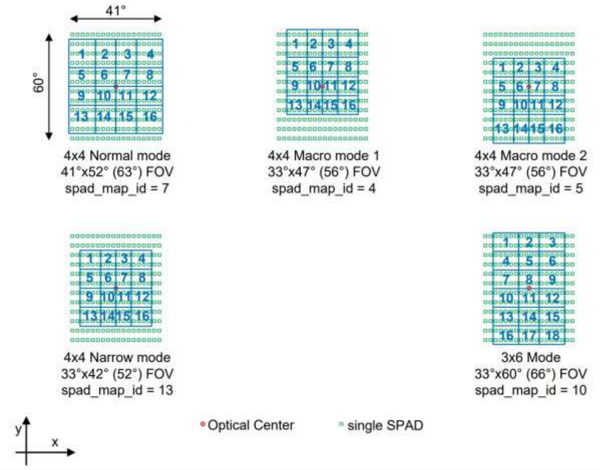
The orientation of the TMF8821 IC's illuminator (VCSEL + MLA) and receiver (lens + SPAD) can be reference by the IC's polarity marker. You'll need to orient the board based on the marker for your project's needs. The image below shows the TMF8821 and only applies to TMF8821 when using a 4x4 and 3x6 SPAD. The TMF8821 can support a 3x3, 4x4, or 3x6 SPAD. Make sure to configure the IC to set the size of the SPAD.

 EN GP1 GP0 INT SCL SDA 3V3 GND sparkfun dToF TMF8820 TMF8821 PWR qwiic	 dToF TMF8820 TMF8821 GP1 EN SYNC INT SCL SDA 3V3 GND	

TMF8821 - IC	*TMF8821 - IC*	*TMF8* *(4*

Note: When using the TMF8821 configured as a 4x4 zone, please note that the zones for 4x4 operating mode are presented in zones 1-8 and 10-17; the result for zone 9 and 18 is not used. This is stated from the datasheet in the section for the "Performance in 4x4 Operating Mode" This feature is only available in certain versions of the IC such as the TMF8821.

Below are SPAD map configurations for the 4x4 and 3x6 modes taken from the datasheet. These configurations are possible with the TMF8821.

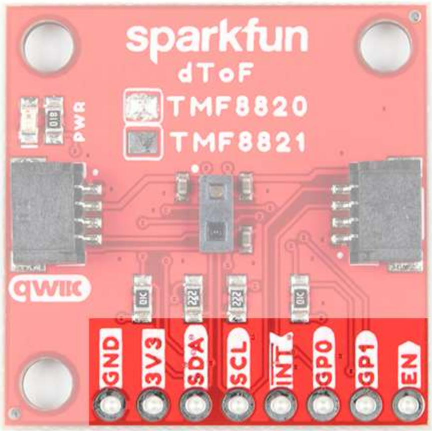
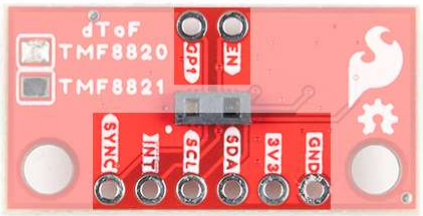


Broken Out Pins

The pins that are broken out are listed out as follows. Note that the pins are rearranged for the Mini version. Most likely you will be using the Qwiic cable to access the sensor. However, you can still solder header pins or wires to the PTHs.

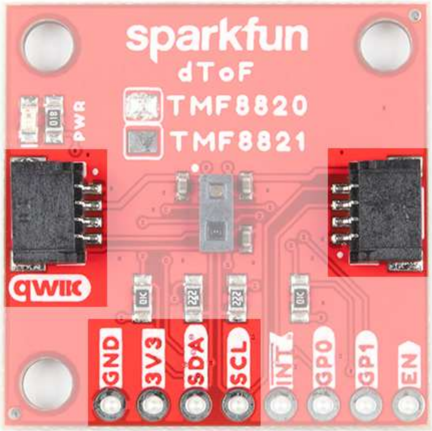
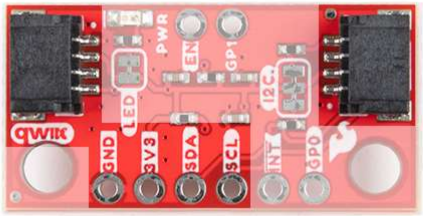
Pin	**Description**	**Direction**
GND	Ground	In
3.3V	Power	In
SDA	Data	In/Out
SCL	Clock	In/Out
$\overline{\text{INT}}$	Interrupt, goes low when data is ready.	Out

GP0/SYNC	General purpose input/output. This pin can also be used to connect to a SYNC signal to interrupt the TMF8820/TMF8821 if the high power illuminator is operating or sync the sensor to a camera operation. Make sure to not On SYNC assertion, the VCSEL is immediately switched off (typically after 10 µs), on SYNC de-assertion the VCSEL operation is resumed.	In/Out
GP1	General purpose input/output.	In/Out
EN	Enable input active high; setting to low forces the device into shutdown and all memory content is lost; this is connected to 3.3V via a 10kΩ pull-up resistor when not being used	In

 sparkfun dToF TMF8820 TMF8821 PWR qwiic GND 3V3 SDA SCL INT GP0 GP1 EN	 dToF TMF8820 TMF8821 GP1 EN SYNC INT SCL SDA 3V3 GND
Qwiic dToF Imager - TMF8820/TMF8821 - Breakout Pins	*Qwiic Mini dToF Imager - TMF8820/TMF8821 - Breakout Pins*

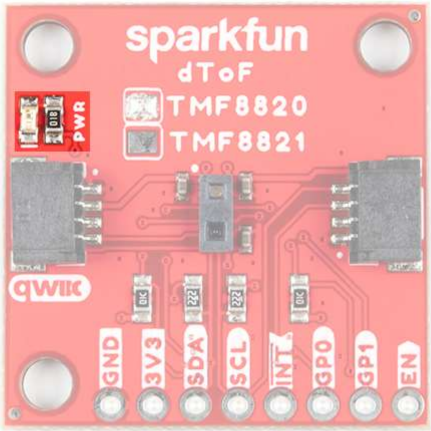
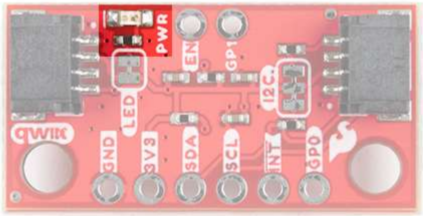
Qwiic and I^{2}C

The breakout boards include 2x Qwiic connectors to easily access the I^{2}C data lines and power. Note that the standard size uses the right angle Qwiic connecgtors The Qwiic ecosystem is made for fast prototyping by removing the need for soldering. All you need to do is plug a Qwiic cable into the Qwiic connector and voila! The I^{2}C address for each sensor is **0x41** as stated earlier.

 sparkfun dToF TMF8820 TMF8821 PWR qwiic GND 3V3 SDA SCL INT GP0 GP1 EN	 PWR EN GP1 LED I2C qwiic GND 3V3 SDA SCL INT GP0
Qwiic dToF Imager - TMF8820/TMF8821	*Qwiic Mini dToF Imager - TMF8820/TMF8821*

LED

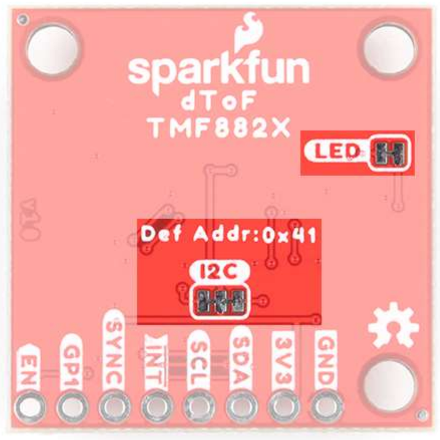
The power LED will light up when the board is powered. To disable, cut the trace on the back of the board.

 sparkfun dToF TMF8820 TMF8821 PWR qwiic GND 3V3 SDA SCL INT GP0 GP1 EN	 PWR EN GP1 LED I2C qwiic GND 3V3 SDA SCL INT GP0
Qwiic dToF Imager - TMF8820/TMF8821 - LED	*Qwiic Mini dToF Imager - TMF8820/TMF8821 - LED*

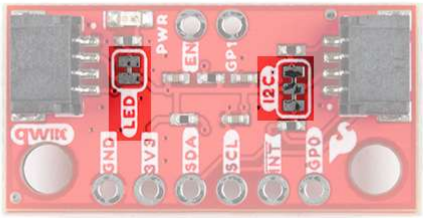
Jumper

There are two jumpers on the back of the board. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.

- **I^{2}C** - This three way jumper labeled I^{2}C connects two 2.2kΩ pull-up resistors to the I^{2}C data lines by default. If you have many devices on your I^{2}C data lines, then you may consider cutting these.
- **LED** - This connects to the power LED by default. Cut the trace to disable the LED.



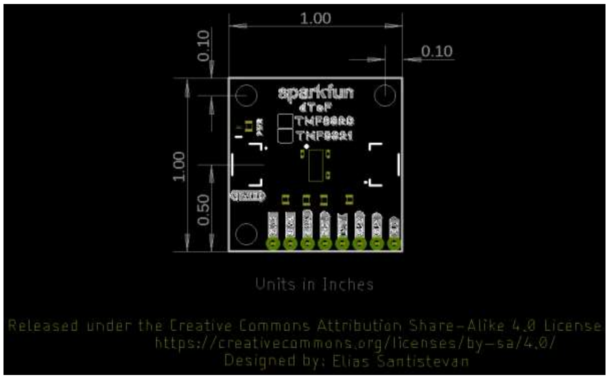
Qwiic dToF Imager - TMF8820/TMF8821 - Jumpers



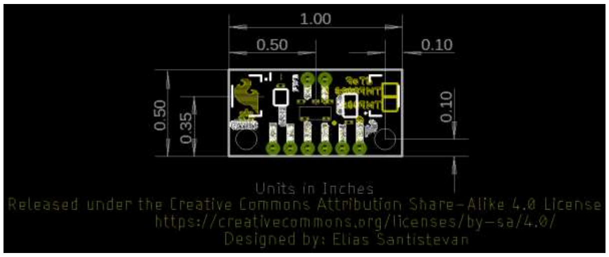
Qwiic Mini dToF Imager - TMF8820/TMF8821 - Jumpers

Board Dimensions

The Qwiic dToF Imager TMF8820/TMF8821 use the standard Qwiic size 1.0"x1.0". The mini uses versions has a footprint that is half the size 0.5" x 1.0".


1.00
0.10
0.10
1.00
0.50
sparkfun
dToF
TMF8820
TMF8821
Units in Inches
Released under the Creative Commons Attribution Share-Alike 4.0 License
https://creativecommons.org/licenses/by-sa/4.0/
Designed by: Elias Santistevan

Qwiic dToF Imager - TMF8820/TMF8821 - Board Dimensions


1.00
0.50
0.10
0.50
0.35
0.10
Units in Inches
Released under the Creative Commons Attribution Share-Alike 4.0 License
https://creativecommons.org/licenses/by-sa/4.0/
Designed by: Elias Santistevan

Qwiic Mini dToF Imager - TMF8820/TMF8821 - Board Dimensions

Hardware Hookup

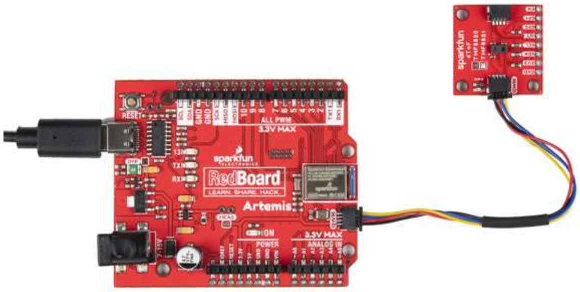
Note: We recommend a microcontroller with enough flash to run your program code. Sorry, Uno's (or any development board using the ATmega328P) are out. But didn't you want an excuse to try out something new? We recommend choosing either an Artemis Thing Plus or an ESP32 Thing Plus board as your development board.

The following Arduino-compatible processor boards are compatible with the Qwiic dToF TMF8820/TMF8821.

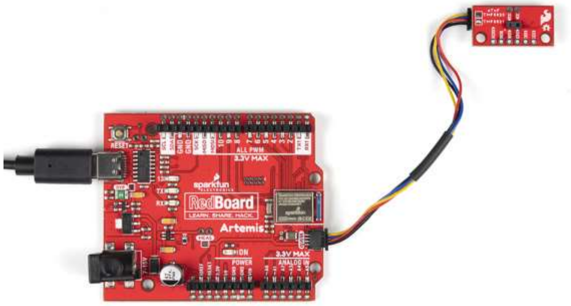
- Artemis
- SAMD21/51
- ESP32
- Teensy
- STM32

- nRF52840

To connect your Qwiic dToF Imager, insert a Qwiic cable between an Arduino-compatible development board and the sensor. Then insert the associated cable for power and programming the microcontroller. In this case, we used the RedBoard Artemis and a USB type C cable. If you're going to be soldering to the through hole pins, then just attach lines to power, ground, and the I^2C data lines to the microcontroller of your choice. Make sure to orient the sensor with respect to the TMF8820/TMF8821's SPADs and your application. In this case, the sensor was rotated 90° counterclockwise for reference.



For the Qwiic Mini versions, you would follow the same steps to connect to the sensor. The only difference is that the board is smaller. The image below shows the RedBoard Artemis connecting to the sensor. However, you could use an Arduino-compatible development board that has a smaller form factor. Make sure to orient the sensor with respect to the TMF8820/TMF8821's SPADs and your application.



Software Installation

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 IDE, library, or board add-on, please review the following tutorials.

- Installing the Arduino IDE
- Installing an Arduino Library
- Installing Board Definitions in the Arduino IDE

If you've never connected an CH340 device to your computer before, you may need to install drivers for the USB-to-serial converter. Check out our section on How to Install CH340 Drivers" for help with the installation.

The SparkFun TMF882X dToF Arduino library can be downloaded with the Arduino library manager by searching '**SparkFun TMF882X dToF Arduino Library**' or you can grab the zip here from the GitHub repository to manually install:

SPARKFUN TMF882X DTOF ARDUINO LIBRARY (ZIP)

Note: For the API reference, check out the following link to the GitHub Page.

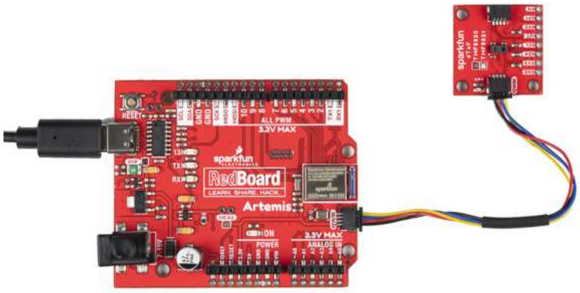
SPARKFUN TMF882X DTOF ARDUINO LIBRARY: API REFERENCE

Arduino Examples

Below are a few examples that are highlighted from the Arduino library.

Example 1: Basic

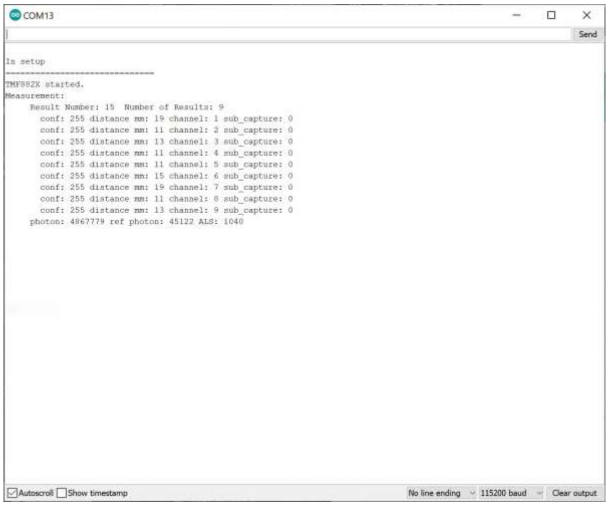
The following example uses the RedBoard Artemis. However, you could also use any Arduino-compatible microcontroller listed earlier in this tutorial if it has enough memory.



After installing the library, open the sketch in Arduino: **File** > **Examples** > **SparkFun Qwiic TMF882X Arduino Library** > **Example-01_Basic**. The following code was copied from the Arduino Library for your convenience. If you have not already, select your Board (in this case the **RedBoard Artemis**), and associated COM port. Upload the code to the board.

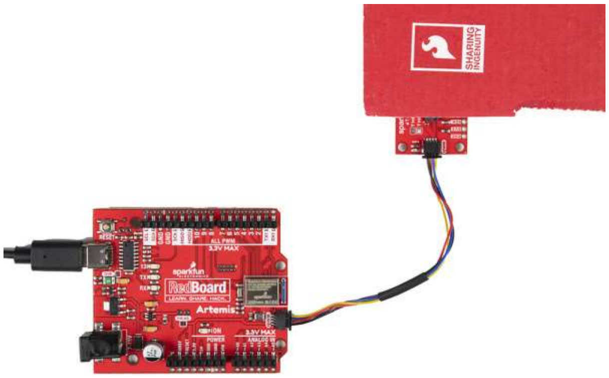
```
/*

  Example-01_Basic.ino

  This demo shows a basic use of a TMF882X device. The device is connected to,
  and a single reading is taken for each loop iteration.

  Supported Boards:

   SparkFun Qwiic dToF Imager - TMF8820        https://www.sparkfun.com/products/19036
   SparkFun Qwiic Mini dToF Imager - TMF8820   https://www.sparkfun.com/products/19218
   SparkFun Qwiic Mini dToF Imager - TMF8821   https://www.sparkfun.com/products/19451
   SparkFun Qwiic dToF Imager - TMF8821        https://www.sparkfun.com/products/19037

  Written by Kirk Benell @ SparkFun Electronics, April 2022

  Repository:
     https://github.com/sparkfun/SparkFun_Qwiic_TMF882X_Arduino_Library

  Documentation:
     https://sparkfun.github.io/SparkFun_Qwiic_TMF882X_Arduino_Library/

  SparkFun code, firmware, and software is released under the MIT License(http://opensource.org/licenses/MIT).
*/

#include "SparkFun_TMF882X_Library.h" //http://librarymanager/All#SparkFun_Qwiic_TMPF882X

SparkFun_TMF882X  myTMF882X;

// Structure to hold the measurement results - this is defined by the TMF882X SDK.

static struct tmf882x_msg_meas_results myResults;

void setup()
{

    delay(1000);
    Serial.begin(115200);
    Serial.println("");

    Serial.println("In setup");
    Serial.println("==============================");

    // Initialize the TMF882X device
    if(!myTMF882X.begin())
    {
        Serial.println("Error - The TMF882X failed to initialize - is the board connected?");
        while(1);
    }else
        Serial.println("TMF882X started.");
    // The device is now ready for operations
}

void loop()
{
    delay(2000);

    // get a Measurement
    if(myTMF882X.startMeasuring(myResults))
    {
        // print out results
        Serial.println("Measurement:");
        Serial.print("     Result Number: "); Serial.print(myResults.result_num);
        Serial.print("  Number of Results: "); Serial.println(myResults.num_results);

        for (int i = 0; i < myResults.num_results; ++i)
        {
            Serial.print("       conf: "); Serial.print(myResults.results[i].confidence);
            Serial.print(" distance mm: "); Serial.print(myResults.results[i].distance_mm);
            Serial.print(" channel: "); Serial.print(myResults.results[i].channel);
            Serial.print(" sub_capture: "); Serial.println(myResults.results[i].sub_capture);

        }
        Serial.print("     photon: "); Serial.print(myResults.photon_count);
        Serial.print(" ref photon: "); Serial.print(myResults.ref_photon_count);
        Serial.print(" ALS: "); Serial.println(myResults.ambient_light); Serial.println();

    }

}
```

Open the Arduino Serial Monitor set to **115200 baud**. Place an object like your finger or a flat, rigid object in front of the dToF Imager's IC. You should see something similar in the output below. The readings show a high confidence value [`.confidence` values are between 0 and 255, with 255 being the highest] and the distance measurements in millimeters [`.distance_mm`] for each channel [`.channel`]. The output also provides information about the total photons received [`.photon_count`], reference photon [`.ref_photon_count`], and tutorial ambient light [`.ambient_light`]received by the channels.

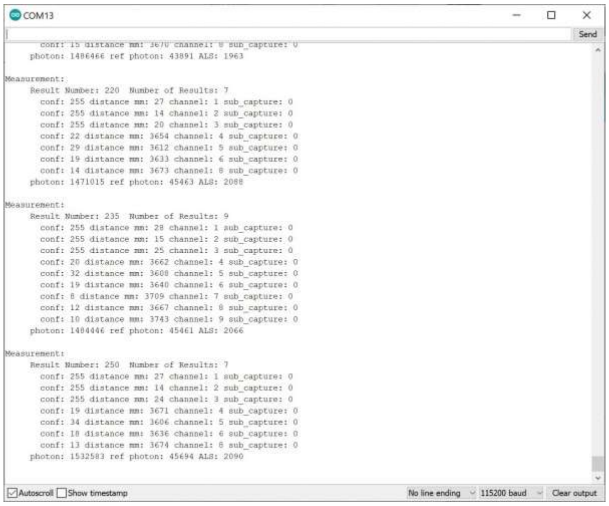


```
In setup
------------------------------
TMF882X started.
Measurement:
     Result Number: 15  Number of Results: 9
       conf: 255 distance mm: 19 channel: 1 sub_capture: 0
       conf: 255 distance mm: 11 channel: 2 sub_capture: 0
       conf: 255 distance mm: 13 channel: 3 sub_capture: 0
       conf: 255 distance mm: 11 channel: 4 sub_capture: 0
       conf: 255 distance mm: 11 channel: 5 sub_capture: 0
       conf: 255 distance mm: 15 channel: 6 sub_capture: 0
       conf: 255 distance mm: 19 channel: 7 sub_capture: 0
       conf: 255 distance mm: 11 channel: 8 sub_capture: 0
       conf: 255 distance mm: 13 channel: 9 sub_capture: 0
     photon: 4867779 ref photon: 45122 ALS: 1040
```

Try positioning the object so that channel 1 through 3 is covered and close to the TMF8820/TMF8821. In this case, we used a piece of cardboard to partially cover the sensor. Make sure to orient the board with respect to the SPADs as shown in the image below to easily view the changes in the output.



With the Serial Monitor still open, should see something similar to the output below. You will notice that the sensor was able to detect an object over channels 1 through 3 with a high confidence. The values for the other channels that were not covered by the cardboard showed a low confidence and longer distance values. The TMF8820/TMF8821 may not display a result if the SPAD does not receive a reflected pulse.



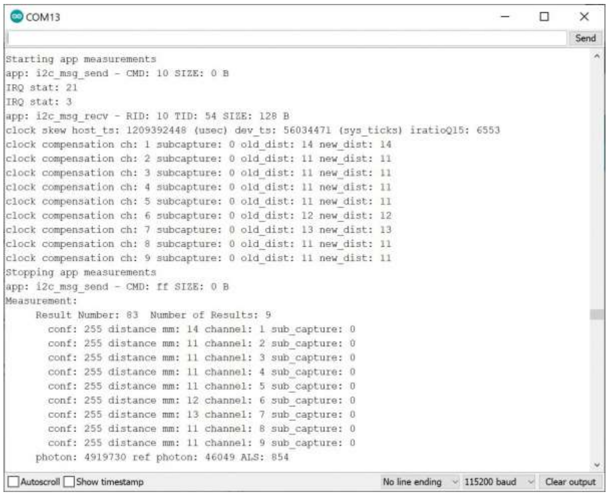
```
       conf: 15 distance mm: 3670 channel: 8 sub_capture: 0
     photon: 1486466 ref photon: 43891 ALS: 1963

Measurement:
     Result Number: 220  Number of Results: 7
       conf: 255 distance mm: 27 channel: 1 sub_capture: 0
       conf: 255 distance mm: 14 channel: 2 sub_capture: 0
       conf: 255 distance mm: 20 channel: 3 sub_capture: 0
       conf: 22 distance mm: 3654 channel: 4 sub_capture: 0
       conf: 29 distance mm: 3612 channel: 5 sub_capture: 0
       conf: 19 distance mm: 3633 channel: 6 sub_capture: 0
       conf: 14 distance mm: 3673 channel: 8 sub_capture: 0
     photon: 1471015 ref photon: 45463 ALS: 2088

Measurement:
     Result Number: 235  Number of Results: 9
       conf: 255 distance mm: 28 channel: 1 sub_capture: 0
       conf: 255 distance mm: 15 channel: 2 sub_capture: 0
       conf: 255 distance mm: 25 channel: 3 sub_capture: 0
       conf: 20 distance mm: 3662 channel: 4 sub_capture: 0
       conf: 32 distance mm: 3608 channel: 5 sub_capture: 0
       conf: 19 distance mm: 3640 channel: 6 sub_capture: 0
       conf: 8 distance mm: 3709 channel: 7 sub_capture: 0
       conf: 12 distance mm: 3667 channel: 8 sub_capture: 0
       conf: 10 distance mm: 3743 channel: 9 sub_capture: 0
     photon: 1484446 ref photon: 45461 ALS: 2066

Measurement:
     Result Number: 250  Number of Results: 7
       conf: 255 distance mm: 27 channel: 1 sub_capture: 0
       conf: 255 distance mm: 14 channel: 2 sub_capture: 0
       conf: 255 distance mm: 24 channel: 3 sub_capture: 0
       conf: 19 distance mm: 3671 channel: 4 sub_capture: 0
       conf: 34 distance mm: 3606 channel: 5 sub_capture: 0
       conf: 18 distance mm: 3636 channel: 6 sub_capture: 0
       conf: 13 distance mm: 3674 channel: 8 sub_capture: 0
     photon: 1532583 ref photon: 45694 ALS: 2090
```

Example 5: Verbose

With the library installed, open the sketch in Arduino: **File** > **Examples** > **SparkFun Qwiic TMF882X Arduino Library** > **Example-05_Verbose**. The following code was copied from the Arduino Library for your convenience. If you have not already, select your Board (in this case the **RedBoard Artemis**), and associated COM port. Upload the code to the board.

```
/*

  Example-05_Verbose.ino

  The TMF882X Arduino library uses the TMF882X Software Development Kit (SDK) from
  AMS to interface with the sensor.

  The AMS SDK is able to print out informational messages during normal operation, as
  well as debug messages. This example shows how to enable those messages and direct
  them to a Serial device for output.

  Supported Boards:

   SparkFun Qwiic dToF Imager - TMF8820        https://www.sparkfun.com/products/19036
   SparkFun Qwiic Mini dToF Imager - TMF8820   https://www.sparkfun.com/products/19218
   SparkFun Qwiic Mini dToF Imager - TMF8821   https://www.sparkfun.com/products/19451
   SparkFun Qwiic dToF Imager - TMF8821        https://www.sparkfun.com/products/19037

  Written by Kirk Benell @ SparkFun Electronics, April 2022

  Repository:
     https://github.com/sparkfun/SparkFun_Qwiic_TMF882X_Arduino_Library

  Documentation:
     https://sparkfun.github.io/SparkFun_Qwiic_TMF882X_Arduino_Library/

  SparkFun code, firmware, and software is released under the MIT License(http://opensource.org/licenses/MIT).
*/

#include "SparkFun_TMF882X_Library.h"    //http://librarymanager/All#SparkFun_Qwiic_TMPF882X

SparkFun_TMF882X  myTMF882X;

static struct tmf882x_msg_meas_results myResults;

void setup()
{

    delay(1000);
    Serial.begin(115200);
    Serial.println("");

    Serial.println("In setup");
    Serial.println("==============================");

    // The underlying TMF882X SDK can output a wide variety of information during
    // normal operation. It's very verbose.
    //
    // Enable this output as part of this demo.
    //
    // Pass in our output device - Serial
    myTMF882X.setOutputDevice(Serial);

    // Enable Info messages
    myTMF882X.setInfoMessages(true);

    // Enable Debug mode. Set this before calling begin to get initialization debug
    // information.

    myTMF882X.setDebug(true);

    if(!myTMF882X.begin())
    {
        Serial.println("Error - The TMF882X failed to initialize - is the board connected?");
        while(1);
    }else
        Serial.println("TMF882X started.");


}

void loop()
{
    delay(2000);

    // get a myResultsurment
    if(myTMF882X.startMeasuring(myResults))
    {
        // print out results
        Serial.println("Measurement:");
        Serial.print("     Result Number: "); Serial.print(myResults.result_num);
        Serial.print("  Number of Results: "); Serial.println(myResults.num_results);

        for (uint32_t i = 0; i < myResults.num_results; ++i)
        {
            Serial.print("       conf: "); Serial.print(myResults.results[i].confidence);
            Serial.print(" distance mm: "); Serial.print(myResults.results[i].distance_mm);
            Serial.print(" channel: "); Serial.print(myResults.results[i].channel);
            Serial.print(" sub_capture: "); Serial.println(myResults.results[i].sub_capture);

        }
        Serial.print("     photon: "); Serial.print(myResults.photon_count);
        Serial.print(" ref photon: "); Serial.print(myResults.ref_photon_count);
        Serial.print(" ALS: "); Serial.println(myResults.ambient_light); Serial.println();

    }

}
```

Open the Arduino Serial Monitor set to **115200 baud**. The output is similar to the Basic example. However, there is additional information displayed. This information is useful for those debugging with the TMF8820 or TMF8821. The measurements were taken when an object was covering the IC completely.



```
Starting app measurements
app: i2c_msg_send - CMD: 10 SIZE: 0 B
IRQ stat: 21
IRQ stat: 3
app: i2c_msg_recv - RID: 10 TID: 54 SIZE: 128 B
clock skew host_ts: 1209392448 (usec) dev_ts: 56034471 (sys_ticks) iratioQ15: 6553
clock compensation ch: 1 subcapture: 0 old_dist: 14 new_dist: 14
clock compensation ch: 2 subcapture: 0 old_dist: 11 new_dist: 11
clock compensation ch: 3 subcapture: 0 old_dist: 11 new_dist: 11
clock compensation ch: 4 subcapture: 0 old_dist: 11 new_dist: 11
clock compensation ch: 5 subcapture: 0 old_dist: 11 new_dist: 11
clock compensation ch: 6 subcapture: 0 old_dist: 12 new_dist: 12
clock compensation ch: 7 subcapture: 0 old_dist: 13 new_dist: 13
clock compensation ch: 8 subcapture: 0 old_dist: 11 new_dist: 11
clock compensation ch: 9 subcapture: 0 old_dist: 11 new_dist: 11
Stopping app measurements
app: i2c_msg_send - CMD: ff SIZE: 0 B
Measurement:
     Result Number: 83  Number of Results: 9
       conf: 255 distance mm: 14 channel: 1 sub_capture: 0
       conf: 255 distance mm: 11 channel: 2 sub_capture: 0
       conf: 255 distance mm: 11 channel: 3 sub_capture: 0
       conf: 255 distance mm: 11 channel: 4 sub_capture: 0
       conf: 255 distance mm: 11 channel: 5 sub_capture: 0
       conf: 255 distance mm: 12 channel: 6 sub_capture: 0
       conf: 255 distance mm: 13 channel: 7 sub_capture: 0
       conf: 255 distance mm: 11 channel: 8 sub_capture: 0
       conf: 255 distance mm: 11 channel: 9 sub_capture: 0
     photon: 4919730 ref photon: 46049 ALS: 854
```

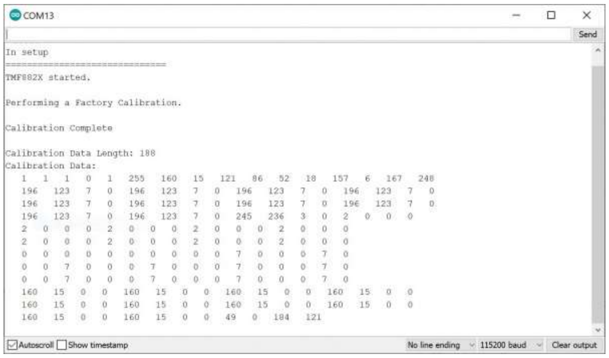
Example 8: Factory Calibration

This example is suggested when placing the TMF8820 and TMF8821 its final application and when adjusting the SPAD mask selection. As suggested in the datasheet under **7.3 Calibration**, "*the calibration test shall be done in a housing with minimal ambient light and no target within 40 cm in field of view of the TMF8820/21.*"

With the library installed, open the sketch in Arduino: **File** > **Examples** > **SparkFun Qwiic TMF882X Arduino Library** > **Example-08_FactoryCal**. The following code was copied from the Arduino Library for your convenience. If you have not already, select your Board (in this case the **RedBoard Artemis**), and associated COM port. Upload the code to the board.

```
/*

  Example-08_FactoryCal.ino

  This example shows how to peform a Factory Calibration on the connected
  TMF882X device. Details on the calibration and it's use are contained in
  the TMF882X datasheet.

  Supported Boards:

   SparkFun Qwiic dToF Imager - TMF8820        https://www.sparkfun.com/products/19036
   SparkFun Qwiic Mini dToF Imager - TMF8820   https://www.sparkfun.com/products/19218
   SparkFun Qwiic Mini dToF Imager - TMF8821   https://www.sparkfun.com/products/19451
   SparkFun Qwiic dToF Imager - TMF8821        https://www.sparkfun.com/products/19037

  Written by Kirk Benell @ SparkFun Electronics, April 2022

  Repository:
     https://github.com/sparkfun/SparkFun_Qwiic_TMF882X_Arduino_Library

  Documentation:
     https://sparkfun.github.io/SparkFun_Qwiic_TMF882X_Arduino_Library/

  SparkFun code, firmware, and software is released under the MIT License(http://opensource.org/licenses/MIT).
*/


#include <SparkFun_TMF882X_Library.h>    //http://librarymanager/All#SparkFun_Qwiic_TMPF882X

SparkFun_TMF882X  myTMF882X;

void setup()
{

    delay(1000);
    Serial.begin(115200);
    Serial.println("");

    Serial.println("In setup");
    Serial.println("==============================");

    if(!myTMF882X.begin())
    {
        Serial.println("Error - The TMF882X failed to initialize - is the board connected?");
        while(1){}
    }else
        Serial.println("TMF882X started.");


    Serial.println();
    Serial.println("Performing a Factory Calibration.");
    Serial.println();
    // Perform a factory calibration of the connected device.

    // First set some config parameters to support the calibration
    struct tmf882x_mode_app_config tofConfig;
    if (!myTMF882X.getTMF882XConfig(tofConfig))
    {
        Serial.println("Error - unable to get device configuration.");
        while(1){}
    }

    // Change the APP configuration
    //  - set the reporting period to 500 milliseconds
    //  - set the iterations to 4,000,000 (4M) to perform factory calibration
    tofConfig.report_period_ms = 500;
    tofConfig.kilo_iterations = 4000;

    if (!myTMF882X.setTMF882XConfig(tofConfig))
    {
        Serial.println("Error - unable to set device configuration.");
        while(1){}
    }

    struct tmf882x_mode_app_calib factoryCal;

    // Peform the calibration
    if (!myTMF882X.factoryCalibration(factoryCal))
    {
        Serial.println("Error - Factory Calibration Failed.");
        while(1){}
    }

    // Output the calibration
    Serial.println("Calibration Complete");
    Serial.println();
    Serial.print("Calibration Data Length: ");
    Serial.println(factoryCal.calib_len);

    Serial.println("Calibration Data:");
    for (int i = 0; i < factoryCal.calib_len; i++)
    {
        Serial.print("   "); Serial.print(factoryCal.data[i]);

        if ( (i + 1) % 16 == 0 )
            Serial.println();
    }
    Serial.println();

}

void loop()
{
    delay(2000);

    // Nothing - just here for the calibration example
}
```

Open the Arduino Serial Monitor set to **115200 baud**. You will see an output similar to the image below. You will need to follow the datasheet to calibrate the sensor properly.



```
In setup
==============================
TMF882X started.

Performing a Factory Calibration.

Calibration Complete

Calibration Data Length: 188
Calibration Data:
   1    1    1    0    1   255   160   15   121   86   52   18   157   6   167   248
   196   123   7   0   196   123   7   0   196   123   7   0   196   123   7   0
   196   123   7   0   196   123   7   0   196   123   7   0   196   123   7   0
   196   123   7   0   196   123   7   0   245   236   3   0   2   0   0   0
   2   0   0   0   2   0   0   0   2   0   0   0   2   0   0   0
   2   0   0   0   2   0   0   0   2   0   0   0   2   0   0   0
   0   0   0   0   0   0   0   0   0   0   7   0   0   0   7   0
   0   0   7   0   0   0   7   0   0   0   7   0   0   0   7   0
   0   0   7   0   0   0   7   0   0   0   7   0   0   0   7   0
   160   15   0   0   160   15   0   0   160   15   0   0   160   15   0   0
   160   15   0   0   160   15   0   0   160   15   0   0   160   15   0   0
   160   15   0   0   160   15   0   0   49   0   184   121
```

Example 9: SPAD Map

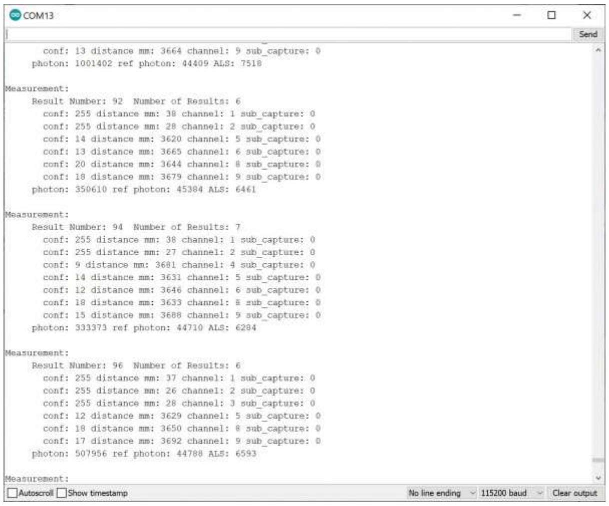
Note: You will need to load the SPAD configuration (i.e. `SPAD_MAP_ID` each time the TMF8820/TMF8821 is powered up. Make sure to also calibrate the sensor after adjusting the SPAD map.

With the library installed, open the sketch in Arduino: **File** > **Examples** > **SparkFun Qwiic TMF882X Arduino Library** > **Example-09_SPADMap**. The following code was copied from the Arduino Library for your convenience. If you have not already, select your Board (in this case the **RedBoard Artemis**), and associated COM port. Upload the code to the board.

```
/*

  Example-09_SPADMap.ino

  The Optical performance of the TMF882X is controled by a SPAD (Single Photon
  Avalanche Photodiode) Map.

  SPAD Maps are set using a SPAD Map ID, which are detailed in the TMF882X datasheet.

  This example shows how to determine the current SPAD Map on the device and change
  it to a desired map.

  Supported Boards:

   SparkFun Qwiic dToF Imager - TMF8820        https://www.sparkfun.com/products/19036
   SparkFun Qwiic Mini dToF Imager - TMF8820   https://www.sparkfun.com/products/19218
   SparkFun Qwiic Mini dToF Imager - TMF8821   https://www.sparkfun.com/products/19451
   SparkFun Qwiic dToF Imager - TMF8821        https://www.sparkfun.com/products/19037

  Written by Kirk Benell @ SparkFun Electronics, April 2022

  Repository:
     https://github.com/sparkfun/SparkFun_Qwiic_TMF882X_Arduino_Library

  Documentation:
     https://sparkfun.github.io/SparkFun_Qwiic_TMF882X_Arduino_Library/

  SparkFun code, firmware, and software is released under the MIT License(http://opensource.org/
licenses/MIT).
*/

#include <SparkFun_TMF882X_Library.h>    //http://librarymanager/All#SparkFun_Qwiic_TMPF882X

static struct tmf882x_msg_meas_results myResults;


SparkFun_TMF882X  myTMF882X;

// What SPAD map to change to

#define NEW_SPAD_MAP 2

void setup()
{

    delay(1000);
    Serial.begin(115200);
    Serial.println("");

    Serial.println("In setup");
    Serial.println("==============================");
    if(!myTMF882X.begin())
    {
        Serial.println("Error - The TMF882X failed to initialize - is the board connected?");
        while(1){}
    }else
        Serial.println("TMF882X started.");


    // Let's change the SPAD map in use on this device.
    //
    // Get the current SPAD Map ID
    int spadMap =  myTMF882X.getCurrentSPADMap();
    Serial.println();
    Serial.print("Current SPAD Map ID: ");
    Serial.println(spadMap);

    // Now switch
    Serial.println("Switching SPAD Map to ID 2 - 3x3 Macro 1 off center");
    Serial.println();

    if (!myTMF882X.setCurrentSPADMap(NEW_SPAD_MAP))
    {
        Serial.println("Error -  Failed to set the SPAD Map - halting");
        while(1){}
    }

    // Let's make sure it worked
    spadMap =  myTMF882X.getCurrentSPADMap();

    if(spadMap != NEW_SPAD_MAP)
    {
        Serial.println("Error -  Failed to set the SPAD Map - halting");
        while(1){}
    }

    Serial.print("The new SPAD Map ID: ");
    Serial.println(spadMap);

    // Now set some config parameters to support the spad map
    struct tmf882x_mode_app_config tofConfig;

    if (!myTMF882X.getTMF882XConfig(tofConfig))
    {
        Serial.println("Error - unable to get device configuration.");
        while(1){}
    }

    // Change the APP configuration
    //  - set the reporting period to 500 milliseconds
    tofConfig.report_period_ms = 500;

    if (!myTMF882X.setTMF882XConfig(tofConfig))
    {
        Serial.println("Error - unable to set device configuration.");
        while(1){}
    }

}

void loop()
{
    delay(2000);

    // get a myResultsurment
    if(myTMF882X.startMeasuring(myResults))
    {
        // print out results
        Serial.println("Measurement:");
        Serial.print("     Result Number: "); Serial.print(myResults.result_num);
        Serial.print("  Number of Results: "); Serial.println(myResults.num_results);

        for (int i = 0; i < myResults.num_results; ++i)
        {
            Serial.print("       conf: "); Serial.print(myResults.results[i].confidence);
            Serial.print(" distance mm: "); Serial.print(myResults.results[i].distance_mm);
            Serial.print(" channel: "); Serial.print(myResults.results[i].channel);
            Serial.print(" sub_capture: "); Serial.println(myResults.results[i].sub_capture);

        }
        Serial.print("     photon: "); Serial.print(myResults.photon_count);
        Serial.print(" ref photon: "); Serial.print(myResults.ref_photon_count);
        Serial.print(" ALS: "); Serial.println(myResults.ambient_light); Serial.println();

    }
}
```

This example allows you to select the operating mode of the SPAD. The datasheet under "**8.5.17 SPAD_MAP_ID Register**" provides a list of acceptable values and modes. Note that certain values are reserved and zone configurations may not be available to use for the TMF8820 (i.e. limited to 3x3) or TMF8821 (i.e. limited to 3x3, 4x4, 3x6).

Once uploaded, open the Arduino Serial Monitor at **115200 baud** and cover channels 1 through 3. The spad_map_id was changed to `2`. The readings are similar to example 1 when the SPAD was using the default 3x3 Normal Mode. However, SPAD uses a different FOV and slightly offset.

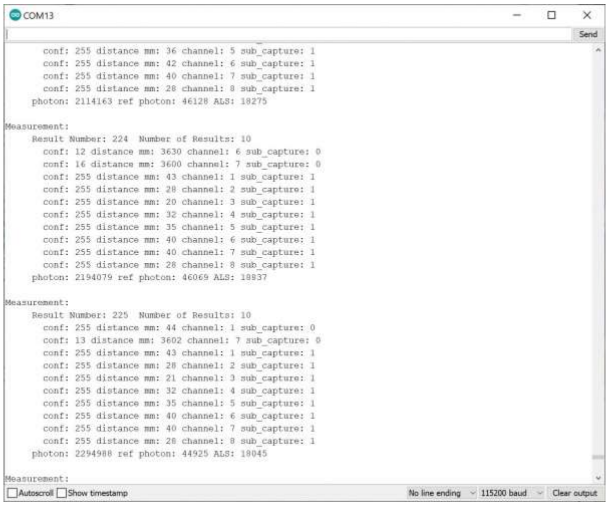


```
      conf: 13 distance mm: 3664 channel: 9 sub_capture: 0
    photon: 1001402 ref photon: 44409 ALS: 7518

Measurement:
    Result Number: 92  Number of Results: 6
      conf: 255 distance mm: 38 channel: 1 sub_capture: 0
      conf: 255 distance mm: 28 channel: 2 sub_capture: 0
      conf: 14 distance mm: 3620 channel: 5 sub_capture: 0
      conf: 13 distance mm: 3665 channel: 6 sub_capture: 0
      conf: 20 distance mm: 3644 channel: 8 sub_capture: 0
      conf: 18 distance mm: 3679 channel: 9 sub_capture: 0
    photon: 350610 ref photon: 45384 ALS: 6461

Measurement:
    Result Number: 94  Number of Results: 7
      conf: 255 distance mm: 38 channel: 1 sub_capture: 0
      conf: 255 distance mm: 27 channel: 2 sub_capture: 0
      conf: 9 distance mm: 3681 channel: 4 sub_capture: 0
      conf: 14 distance mm: 3631 channel: 5 sub_capture: 0
      conf: 12 distance mm: 3646 channel: 6 sub_capture: 0
      conf: 18 distance mm: 3633 channel: 8 sub_capture: 0
      conf: 15 distance mm: 3688 channel: 9 sub_capture: 0
    photon: 333373 ref photon: 44710 ALS: 6284

Measurement:
    Result Number: 96  Number of Results: 6
      conf: 255 distance mm: 37 channel: 1 sub_capture: 0
      conf: 255 distance mm: 26 channel: 2 sub_capture: 0
      conf: 255 distance mm: 28 channel: 3 sub_capture: 0
      conf: 12 distance mm: 3629 channel: 5 sub_capture: 0
      conf: 18 distance mm: 3650 channel: 8 sub_capture: 0
      conf: 17 distance mm: 3692 channel: 9 sub_capture: 0
    photon: 507956 ref photon: 44788 ALS: 6593

Measurement:
```

☐ Autoscroll ☐ Show timestamp   No line ending   115200 baud   Clear output

If you have a TMF8821, try adjusting the SPAD to 4x4 normal mode with a 41°x52° FoV by changing spad_map_id to `7` (i.e. change the line `#define NEW_SPAD_MAP 2` to `#define NEW_SPAD_MAP 7`). Then upload the code to the board.

Open the Arduino Serial Monitor at **115200 baud** and try to cover channels 9 through 16. You will see an output similar to the image below indicating that the respective channels were covered.



```
      conf: 255 distance mm: 36 channel: 5 sub_capture: 1
      conf: 255 distance mm: 42 channel: 6 sub_capture: 1
      conf: 255 distance mm: 40 channel: 7 sub_capture: 1
      conf: 255 distance mm: 28 channel: 8 sub_capture: 1
    photon: 2114163 ref photon: 46128 ALS: 18275

Measurement:
    Result Number: 224  Number of Results: 10
      conf: 12 distance mm: 3630 channel: 6 sub_capture: 0
      conf: 16 distance mm: 3600 channel: 7 sub_capture: 0
      conf: 255 distance mm: 43 channel: 1 sub_capture: 1
      conf: 255 distance mm: 28 channel: 2 sub_capture: 1
      conf: 255 distance mm: 20 channel: 3 sub_capture: 1
      conf: 255 distance mm: 32 channel: 4 sub_capture: 1
      conf: 255 distance mm: 35 channel: 5 sub_capture: 1
      conf: 255 distance mm: 40 channel: 6 sub_capture: 1
      conf: 255 distance mm: 40 channel: 7 sub_capture: 1
      conf: 255 distance mm: 28 channel: 8 sub_capture: 1
    photon: 2194079 ref photon: 46069 ALS: 18837

Measurement:
    Result Number: 225  Number of Results: 10
      conf: 255 distance mm: 44 channel: 1 sub_capture: 0
      conf: 13 distance mm: 3602 channel: 7 sub_capture: 0
      conf: 255 distance mm: 43 channel: 1 sub_capture: 1
      conf: 255 distance mm: 28 channel: 2 sub_capture: 1
      conf: 255 distance mm: 21 channel: 3 sub_capture: 1
      conf: 255 distance mm: 32 channel: 4 sub_capture: 1
      conf: 255 distance mm: 35 channel: 5 sub_capture: 1
      conf: 255 distance mm: 40 channel: 6 sub_capture: 1
      conf: 255 distance mm: 40 channel: 7 sub_capture: 1
      conf: 255 distance mm: 28 channel: 8 sub_capture: 1
    photon: 2294988 ref photon: 44925 ALS: 18045

Measurement:
```

☐ Autoscroll ☐ Show timestamp   No line ending   115200 baud   Clear output

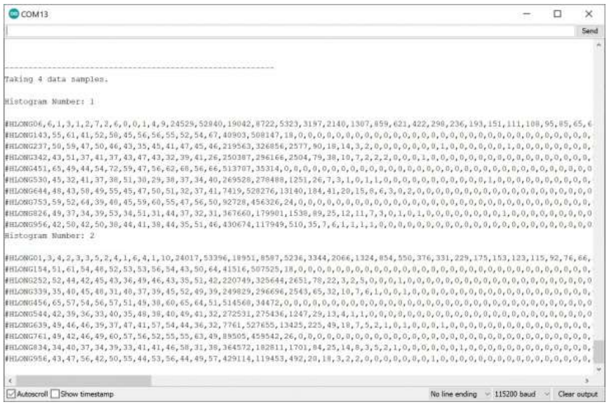
Note: For a custom SPAD map, try checking out **Example-10_CustomSPADMap** and the associated header file to define your own custom SPAD configuration. For more information, check out the datasheet.

Example 11: Histogram

With the library installed, open the sketch in Arduino: **File** > **Examples** > **SparkFun Qwiic TMF882X Arduino Library** > **Example-11_Histogram**. The following code was copied from the Arduino Library for your convenience. If you have not already, select your Board (in this case the **RedBoard Artemis**), and associated COM port. Upload the code to the board.

```
/*

  Example-11_Histogram.ino

  This example shows how to enable and recieve raw histogram data from the
  connected TMF882X device

  Supported Boards:

   SparkFun Qwiic dToF Imager - TMF8820        https://www.sparkfun.com/products/19036
   SparkFun Qwiic Mini dToF Imager - TMF8820   https://www.sparkfun.com/products/19218
   SparkFun Qwiic Mini dToF Imager - TMF8821   https://www.sparkfun.com/products/19451
   SparkFun Qwiic dToF Imager - TMF8821        https://www.sparkfun.com/products/19037

  Written by Kirk Benell @ SparkFun Electronics, April 2022

  Repository:
     https://github.com/sparkfun/SparkFun_Qwiic_TMF882X_Arduino_Library

  Documentation:
     https://sparkfun.github.io/SparkFun_Qwiic_TMF882X_Arduino_Library/

  SparkFun code, firmware, and software is released under the MIT License(http://opensource.org/
licenses/MIT).
*/

#include "SparkFun_TMF882X_Library.h"  //http://librarymanager/All#SparkFun_Qwiic_TMPF882X

SparkFun_TMF882X  myTMF882X;

#define NUMBER_OF_SAMPLES_TO_TAKE  4

int nSample = 0;

// For our histogram printout
#define MAX_BIN_LEN 128

// Define our histogram callback function

void onHistogramCallback(struct tmf882x_msg_histogram *myHistogram)
{

    nSample++;

    Serial.print("Histogram Number: ");
    Serial.println(nSample);

    uint8_t zone_count = 0;
    for (int tdc_idx = 0; tdc_idx < myHistogram->num_tdc; ++tdc_idx)
    {
        // Histogram tag for zones, #HLONG01,#HLONG02....
        Serial.println();
        Serial.print("#HLONG");
        Serial.print(zone_count++);

        for (int bin_idx = 0; bin_idx < myHistogram->num_bins; ++bin_idx) {

            Serial.print((unsigned long)myHistogram->bins[tdc_idx][bin_idx]);

            if ((bin_idx + 1) == MAX_BIN_LEN)
            {
                Serial.println();
                Serial.print("#HLONG");
                Serial.print(zone_count++);

            } else if ((bin_idx + 1) % MAX_BIN_LEN != 0)
                Serial.print(",");
        }
    }
    Serial.println();
}

void setup()
{

    delay(500);
    Serial.begin(115200);
    Serial.println("");


    if(!myTMF882X.begin())
    {
        Serial.println("Error - The TMF882X failed to initialize - is the board connected?");
        while(1){}
    }

    // set our call back function that handles histograms
    myTMF882X.setHistogramHandler(onHistogramCallback);

    // Set our delay between samples  - 1 second - note it's in ms
    myTMF882X.setSampleDelay(1000);

    // First config parameter to enable output of histogram data.

    struct tmf882x_mode_app_config tofConfig;
    if (!myTMF882X.getTMF882XConfig(tofConfig))
    {
        Serial.println("Error - unable to get device configuration.");
        while(1){}
    }

    // Change the APP configuration
    //  - set the reporting period to 500 milliseconds
    //  - Enable Histogram mode
    tofConfig.report_period_ms = 500;
    tofConfig.histogram_dump = 1;

    if (!myTMF882X.setTMF882XConfig(tofConfig))
    {
        Serial.println("Error - unable to set device configuration.");
        while(1){}
    }
}

void loop()
{
    delay(2000);

    // get a measurment
    // Have the sensor take 4 measurements, the results are sent to the above callback

    Serial.println("----------------------------------------------------------");
    Serial.print("Taking ");
    Serial.print(NUMBER_OF_SAMPLES_TO_TAKE);
    Serial.println(" data samples.");
    Serial.println();

    nSample=0;
    myTMF882X.startMeasuring(NUMBER_OF_SAMPLES_TO_TAKE);

    Serial.println("----------------------------------------------------------\n\n");

}
```

Open the Arduino Serial Monitor at **115200 baud** to view the raw histogram data for your TMF8820 and TMF8821. Note that the example code is currently set to take a `4` data samples at a time. You should see an output similar to the image below if you are covering the sensor.



```
----------------------------------------------------------
Taking 4 data samples.

Histogram Number: 1

#HLONG06,6,1,3,1,2,7,2,6,0,0,1,4,9,24529,52840,19042,8722,5323,3197,2140,1307,859,621,422,298,236,193,151,111,108,95,85,65,6
#HLONG143,55,61,41,52,58,45,56,56,55,52,54,67,40903,508147,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG237,50,59,47,50,46,43,35,45,41,47,45,46,219563,326856,2577,90,18,14,3,2,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
#HLONG342,43,51,37,41,37,43,47,43,32,39,41,26,250387,296166,2504,79,38,10,7,2,2,2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG451,65,49,44,54,72,59,47,56,62,68,56,66,513707,35314,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
#HLONG530,45,32,41,37,38,51,30,29,38,37,34,40,269528,278488,1251,26,7,3,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
#HLONG644,48,43,58,49,55,45,47,50,51,32,37,41,7419,528276,13140,184,41,20,15,8,6,3,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
#HLONG753,59,52,64,39,48,45,59,60,55,47,56,50,92728,456326,24,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG826,49,37,34,39,53,34,51,31,44,37,32,31,367660,179901,1538,89,25,12,11,7,3,0,1,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
#HLONG956,42,50,42,50,38,44,41,38,44,35,51,46,430674,117949,510,35,7,6,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Histogram Number: 2

#HLONG01,3,4,2,3,3,5,2,4,1,6,4,1,10,24017,53396,18951,8587,5236,3344,2066,1324,854,550,376,331,229,175,153,123,115,92,76,66,
#HLONG154,51,61,54,48,52,53,53,56,54,43,50,64,41516,507525,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG252,52,44,42,45,43,36,49,46,43,35,51,42,220749,325644,2651,78,22,3,2,5,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
#HLONG339,35,40,45,48,31,48,37,39,45,52,49,39,249829,296696,2543,65,32,10,7,6,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG456,65,57,54,56,57,51,49,38,60,65,64,51,514568,34472,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
#HLONG544,42,39,36,33,40,35,48,38,40,49,41,32,272531,275436,1247,29,13,4,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
#HLONG639,49,46,46,39,37,47,41,57,54,44,36,32,7761,527655,13425,225,49,18,7,5,2,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG761,49,42,46,49,60,57,56,52,55,55,63,49,89505,459542,26,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG834,34,40,37,34,39,33,41,41,46,58,31,38,364572,182811,1701,84,25,14,8,3,5,2,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
#HLONG956,43,47,56,42,50,55,44,53,56,44,49,57,429114,119453,492,20,18,3,2,2,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

☑ Autoscroll □ Show timestamp No line ending 115200 baud

More Examples!

This tutorial highlights a few examples listed in the Arduino Library to get you started. Additional examples can be found in the library.

SPARKFUN QWIIC TMF882X ARDUINO LIBRARY: EXAMPLES

Troubleshooting

Not working as expected and need help?

If you need technical assistance and more information on a product that is not working as you expected, we recommend heading on over to the SparkFun Technical Assistance page for some initial troubleshooting.

SPARKFUN TECHNICAL ASSISTANCE PAGE

If you don't find what you need there, the SparkFun Forums are a great place to find and ask for help. If this is your first visit, you'll need to create a Forum Account to search product forums and post questions.

CREATE NEW FORUM ACCOUNT LOG INTO SPARKFUN FORUMS

Resources & Going Further

Now that you've successfully got your Qwiic dToF Imager up and running, it's time to incorporate it into your own project! For more information, check out the resources below:

- SparkFun Qwiic dToF Imager - TMF8820
  - Schematic (PDF)
  - Eagle Files (ZIP)
  - Board Dimensions (PDF)
- SparkFun Qwiic Mini dToF Imager - TMF8820
  - Schematic (PDF)
  - Eagle Files (ZIP)
  - Board Dimensions (PDF)
- SparkFun Qwiic dToF Imager - TMF8821
  - Schematic (PDF)
  - Eagle Files (ZIP)
  - Board Dimensions (PDF)
- SparkFun Qwiic Mini dToF Imager - TMF8821
  - Schematic (PDF)
  - Eagle Files (ZIP)
  - Board Dimensions (PDF)
- TMF882X
  - Datasheet (PDF)
  - Driver User Guide (PDF)
  - Host Driver Communication (PDF)
- Arduino Library
  - API Reference
- GitHub Hardware Repo
- SFE Product Showcase

Need some inspiration for your next project? Check out some of these other tutorials using sensors.

Bark Back Interactive Pet Monitor

Monitor and interact with pets through this dog bark detector project based on the Raspberry Pi!

PIR Motion Sensor Hookup Guide

An overview of passive infrared (PIR) motion detecting sensors, and how to hook them up to an Arduino.

Simultaneous RFID Tag Reader Hookup Guide

A basic guide to getting started with the RFID Tag Reader breakout and how to read and write multiple RFID tags over multiple feet!

Air Quality Sensor - SGP40 (Qwiic) Hookup Guide

Get started measuring indoor air quality with the SparkFun Air Quality Sensor - SGP40 (Qwiic) Hookup Guide.