SparkFun Temperature Sensor - STTS22H (Qwiic) Hookup Guide

none

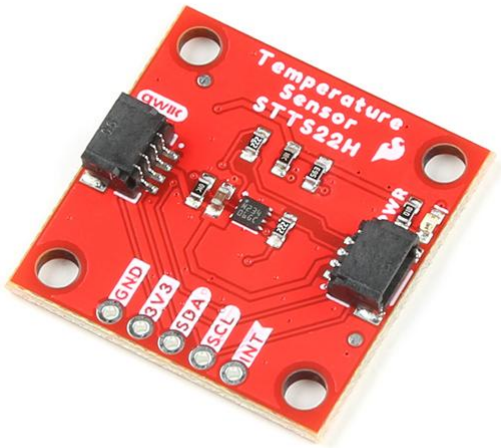
None

Table of contents

1. Hookup Guide

1.1 Introduction

The SparkFun Temperature Sensor - STTS22H (Qwiic) and the SparkFun Micro Temperature Sensor - STTS22H (Qwiic) are Qwiic enabled breakout boards based on the ultralow-power, high-accuracy, digital temperature sensor STTS22H from ST Microelectronics. Thanks to its factory calibration the STTS22H offers high-end accuracy performance over the entire operating temperature range, reaching as low as ±0.5 °C without requiring any further calibration at the application level.

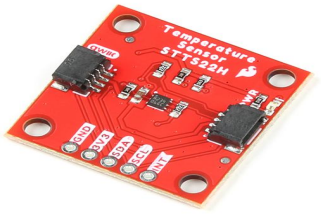


SparkFun Temperature Sensor - STTS22H (Qwiic)

SparkFun Micro Temperature Sensor - STTS22H (Qwiic)

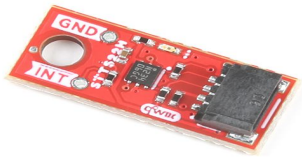
Required Materials

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.



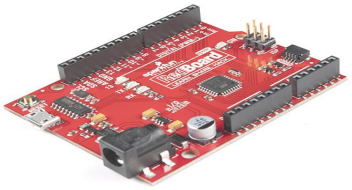
SparkFun Temperature Sensor - STTS22H (Qwiic)

SEN-21262



SparkFun Micro Temperature Sensor - STTS22H (Qwiic)

SEN-21273

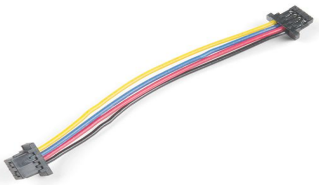


SparkFun RedBoard Qwiic

DEV-15123

USB micro-B Cable - 6 Foot

CAB-10215



Qwiic Cable - 50mm

PRT-14426

Suggested Reading

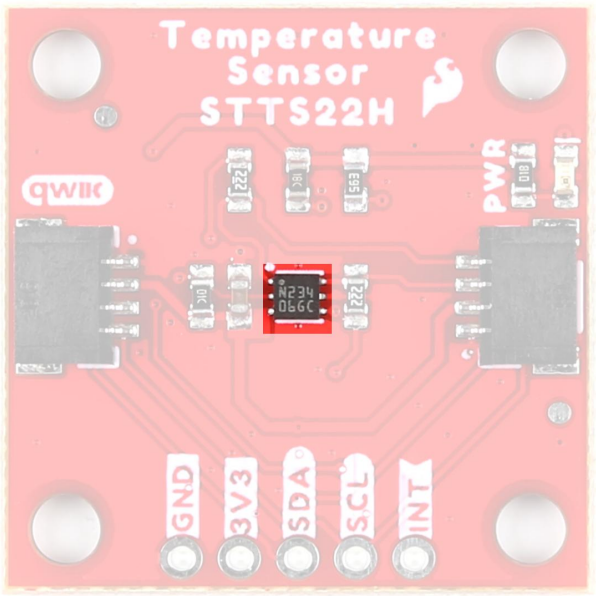
If you aren't familiar with the Qwiic system, we recommend reading here for an overview.

Qwiic Connect System

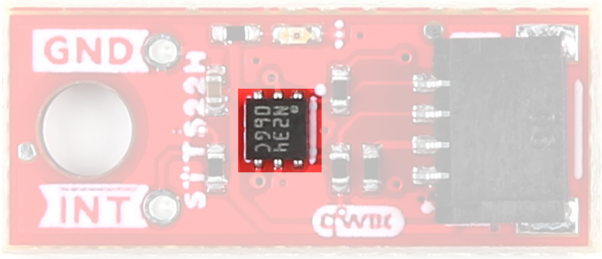
1.2 Hardware Overview

STTS22H

The STTS22H is an ultralow-power, high-accuracy, digital temperature sensor offering high performance over the entire operating temperature range. Thanks to its factory calibration the STTS22H offers high-end accuracy performance over the entire operating temperature range, reaching as low as ±0.5 °C without requiring any further calibration at the application level. The sensor operating mode is user-configurable and allows selecting between different ODRs (down to 1 Hz) or the one-shot mode for battery saving. In one-shot mode, the sensor current consumption falls to 1.75 µA. For more information, refer to the datasheet.



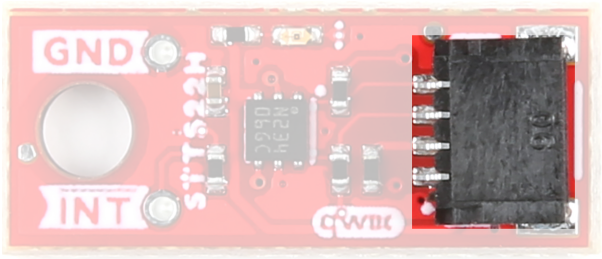
STTS22H

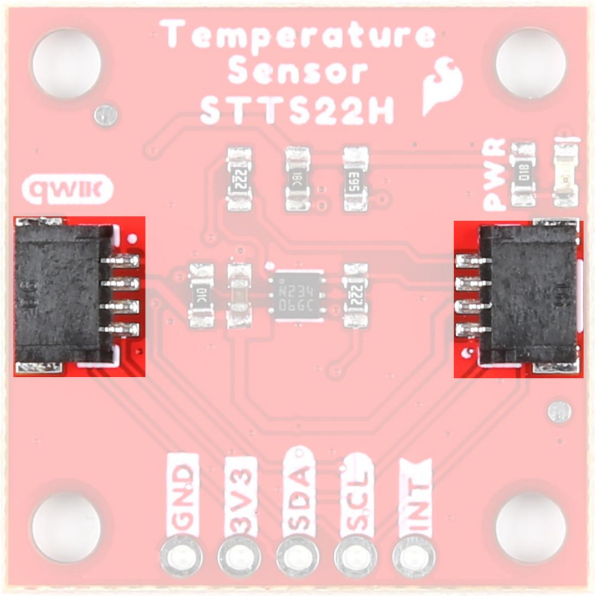


STTS22H on Micro

Qwiic Connectors

There are two Qwiic connectors on either side of the SparkFun Temperature Sensor - STTS22H to provide power and I^2C connectivity simultaneously. The Micro version has a single Qwiic connector that again provides power and I^2C connectivity. The I^2C address of both boards is **0x3C** by default, but the 1x1 board has 3 other addresses the board can be configured to use, while the Micro has 1 other address available.



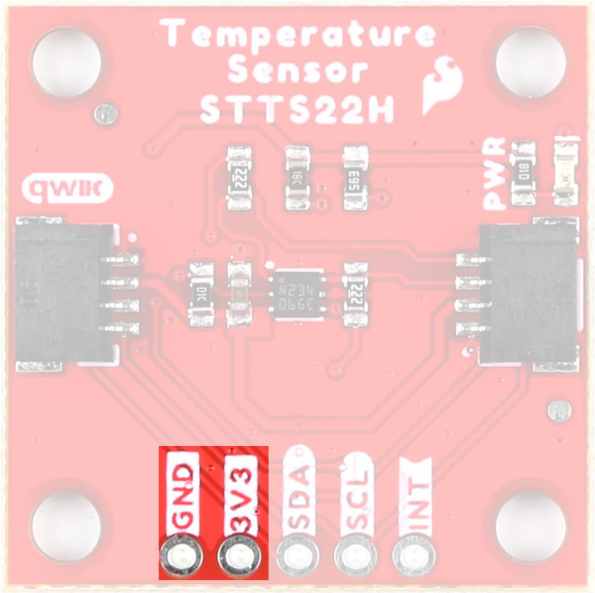


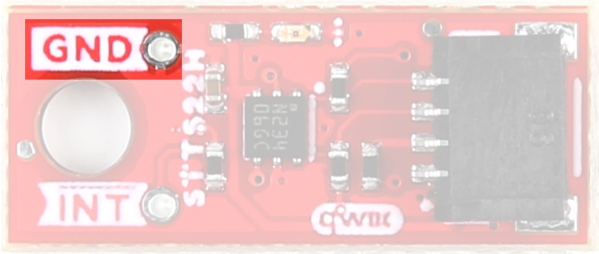
Qwiic Connectors

Qwiic Connector on Micro

Power

Ideally, power will be supplied via the Qwiic connector(s). Alternatively, power can be supplied through the header along the bottom side of the board labeled `3V3` and `GND`. The input voltage range should be between **1.5**-**3.6V**. The Micro version has a single Ground Pin available.





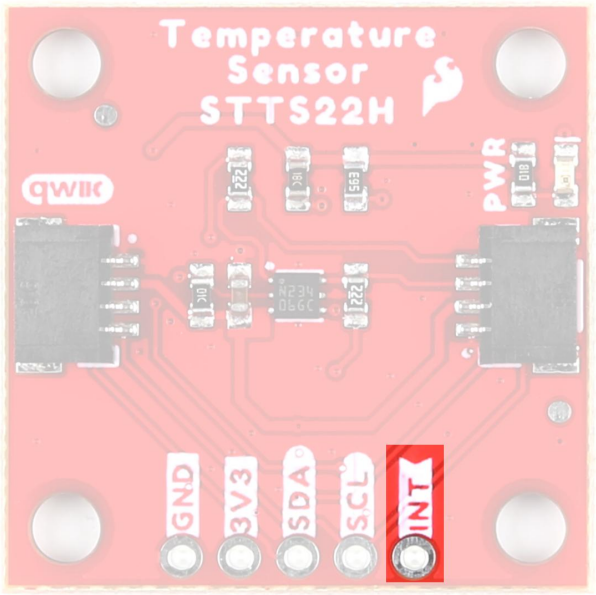
3.3V & GND Pins

GND Pin on Micro

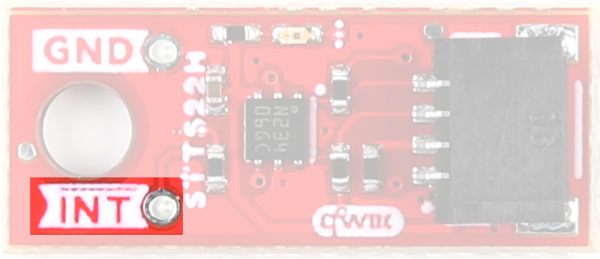
⚡ **Note:** There is no onboard voltage regulation on either of these boards. If you choose to provide power via the plated through holes, ensure that your voltage does not exceed **5.5V**.

Interrupt Pin

An interrupt pin is available to signal the application whenever the selectable high or low threshold has been exceeded.



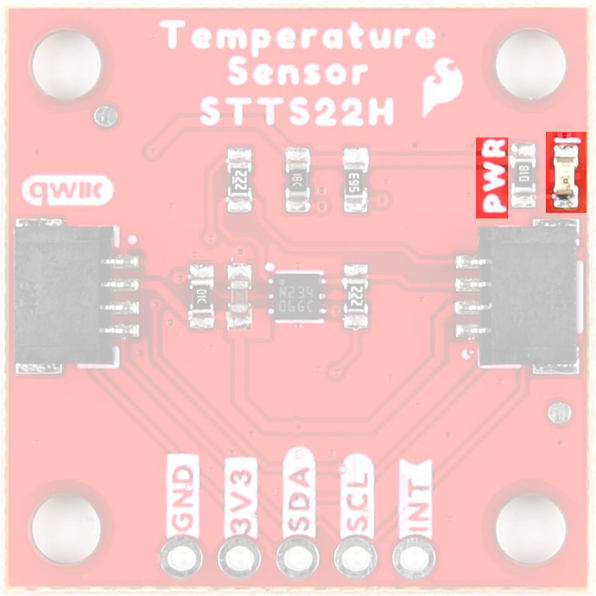
Interrupt Pin

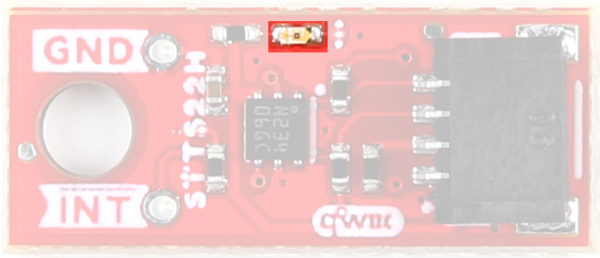


Interrupt Pin on Micro

Power LED

Hopefully this is self-explanatory, but this LED lights up when power is supplied to the board.

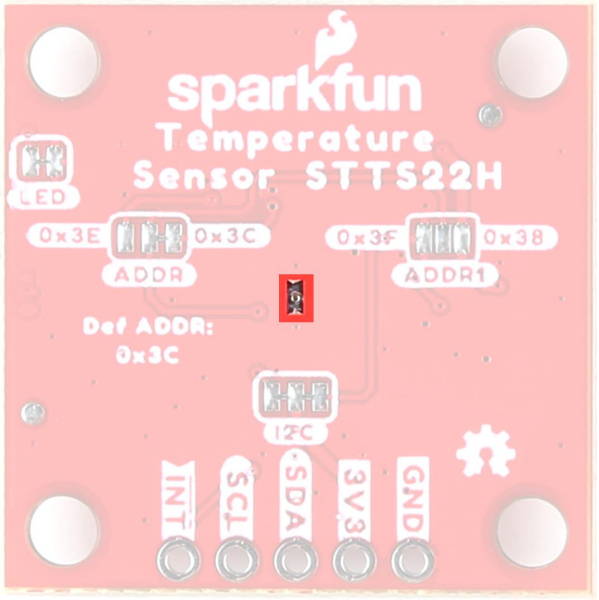


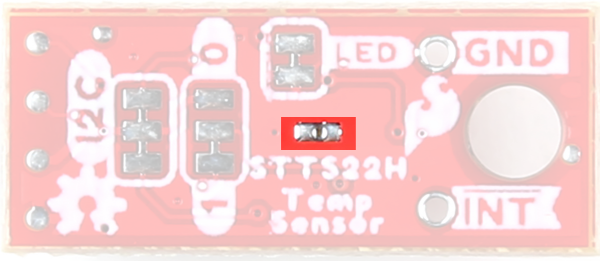


Power LED *Power LED on Micro*

Exposed Pad

There's an extra pad on the bottom side of each board that allows for the most accurate possible readings.



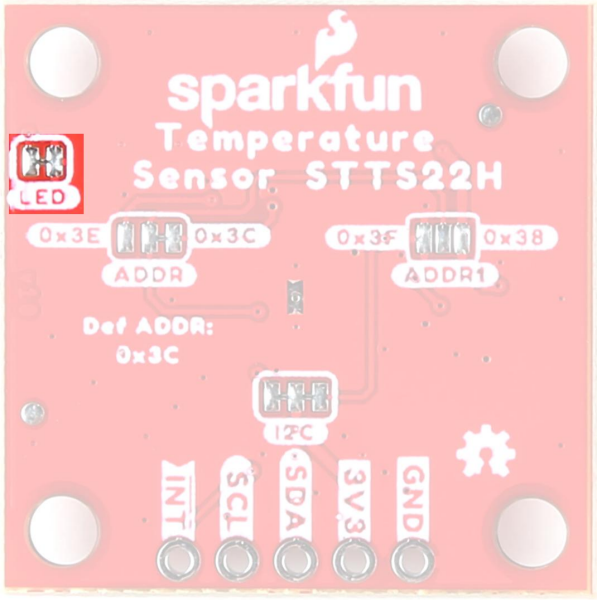


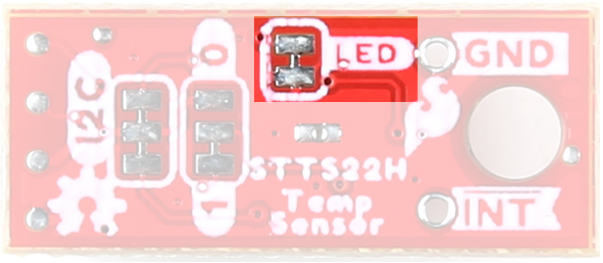
Exposed Pad *Exposed Pad on Micro*

Jumpers

LED JUMPER

If power consumption is an issue, cut this jumper to disable the LED on the front of the board.





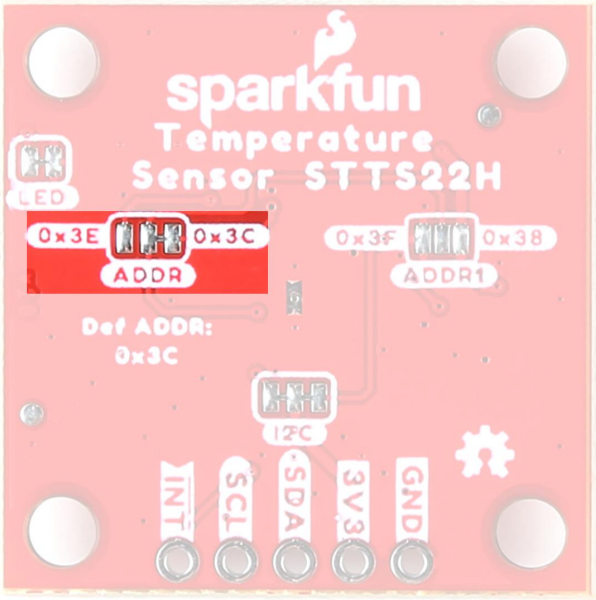
Power LED Jumper

Power LED Jumper on Micro

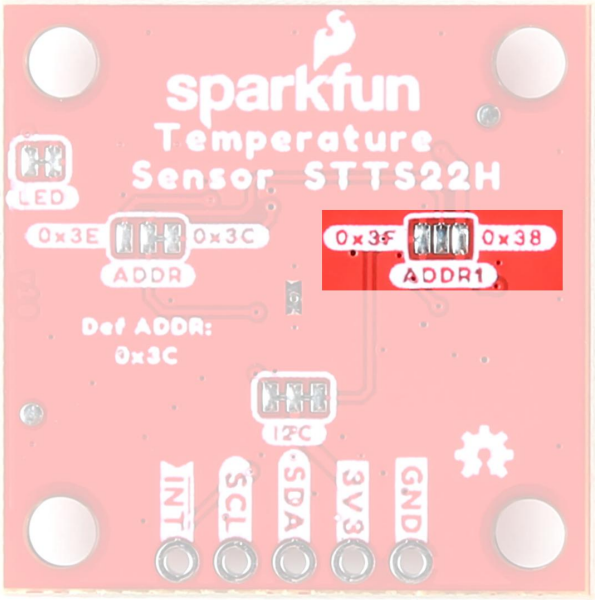
ADDRESS JUMPERS

The 1x1 board has two address jumpers available. The default I^{2}C address is **0x3C**. By cutting various trace combinations, there are three other I^{2}C addresses available.

ADDR	
R8(15K)	0x3C (Default)
R7(56K)	0x3E
VDD	0x38
GND	0x3F
OPEN	Undefined

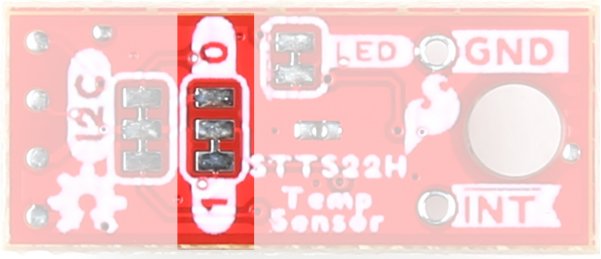


Address Jumper



Address Jumper 1

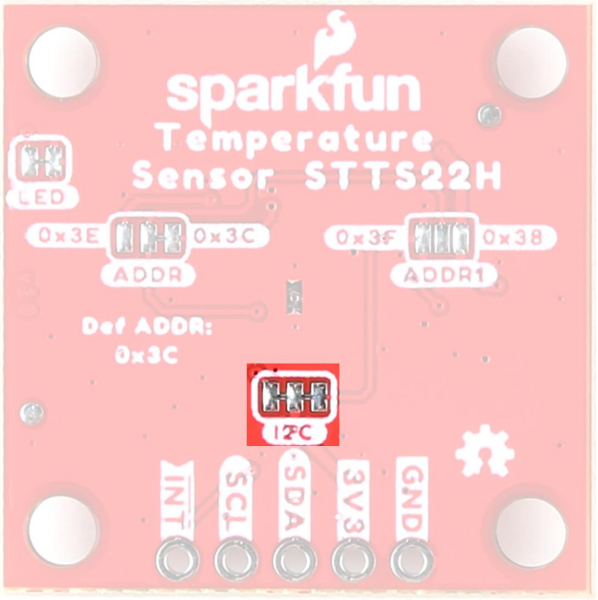
The Micro version of this board has a single address jumper that affords the ability to change the I^{2}C address from **0x3C (Default)** to 0x38.

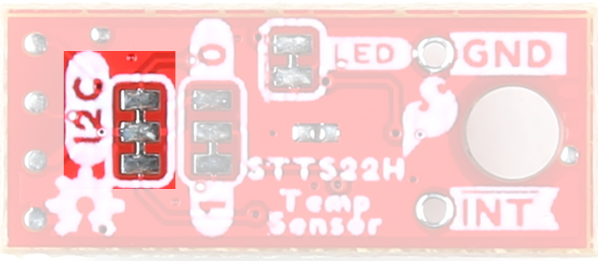


Address Jumper on Micro

I^{2}C JUMPER

These boards are both equipped with pull-up resistors. If you are daisy-chaining multiple Qwiic devices, you will want to cut this jumper; if multiple sensors are connected to the bus with the pull-up resistors enabled, the parallel equivalent resistance will create too strong of a pull-up for the bus to operate correctly. As a general rule of thumb, disable all but one pair of pull-up resistors if multiple devices are connected to the bus. To disable the pull up resistors, use an X-acto knife to cut the joint between the two jumper pads highlighted below.

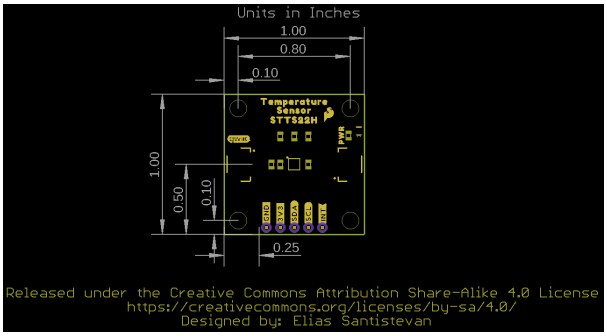




I²C Jumper *I²C Jumper on Micro*

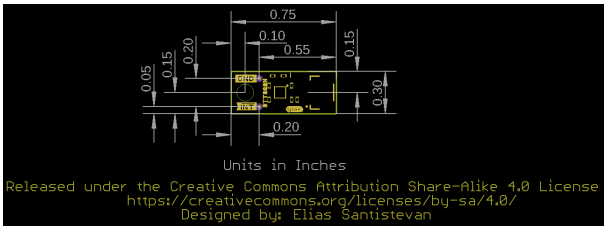
Board Outline

The standard Temperature Sensor STTS22H Breakout measures 1" x 1".



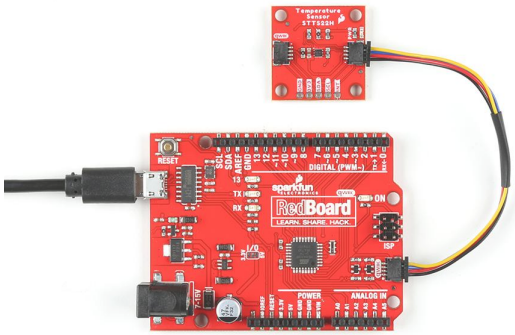
Board Outline of 1" x 1"

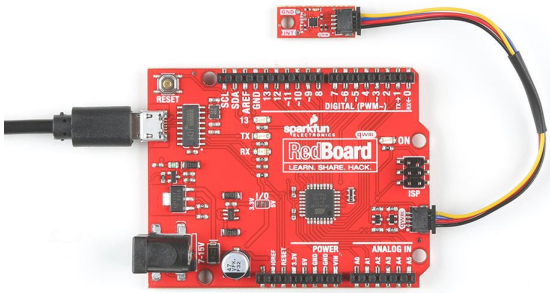
The Micro Temperature Sensor STTS22H Breakout measures 0.75" x 0.3".



Board Outline of Micro

The delightful thing about our Qwiic System is that it makes hooking up your project as easy as plug and play. Pop one end of your Qwiic connector into the controlling board and plug the other end of your Qwiic connector into your STTS22H Temperature Sensor board! Voila!





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.

SparkFun has written a library to work with the SparkFun Temperature Sensor - STTS22H (Qwiic). You can obtain this library through the Arduino Library Manager by searching for "STTS22H". Find the one written by SparkFun Electronics and install the latest version. If you prefer downloading libraries manually, you can grab them from the GitHub Repository.

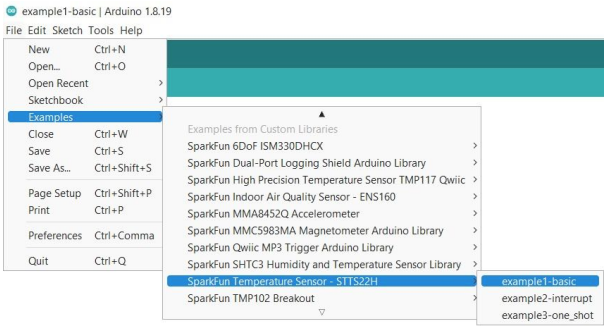
SparkFun STTS22H Temperature Sensor Arduino Library (ZIP)

1.3 Examples

Example 1: Basic Readings

Now that we've got our library installed and our hardware all hooked up, let's look at some examples.

This first example just does some basic measurements. To find Example 1, go to **File > Examples > SparkFun Temperature Sensor - STTS22H > example1-basic**:

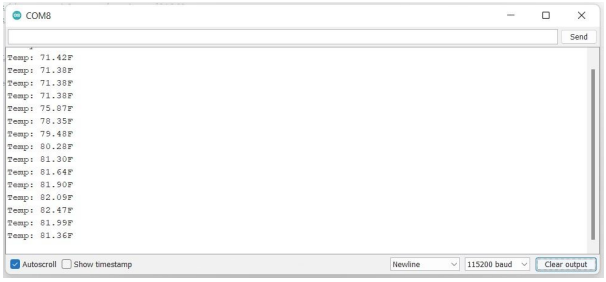


Alternatively, you can copy and paste the code below to a shiny new Arduino file:

Example 1 Arduino Code

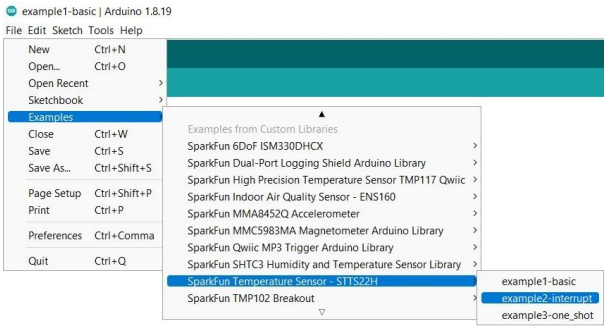
```
/*
example1-basic.ino

This example shows basic data retrieval from the SparkFun Temperature Sensor - STTS22H.

Output Data Rates:

STTS22H_POWER_DOWN
STTS22H_ONE_SHOT
STTS22H_1Hz
STTS22H_25Hz
STTS22H_50Hz
STTS22H_100Hz
STTS22H_200Hz

Written by:
Elias Santistevan @ SparkFun Electronics December, 2022

Products:
   SparkFun Temperature Sensor - STTS2H             https://www.sparkfun.com/products/21262
   SparkFun Micro Temperature Sensor - STTS2H       https://www.sparkfun.com/products/21051

Repository:
     https://github.com/sparkfun/SparkFun_STTS22H_Arduino_Library

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

*/

#include <Wire.h>
#include "SparkFun_STTS22H.h"

SparkFun_STTS22H mySTTS;

float temp;

void setup()
{

    Wire.begin();

    Serial.begin(115200);

    if( !mySTTS.begin() )
    {
        Serial.println("Did not begin.");
        while(1);
```

Once you've got your code uploaded, open up a Serial Monitor and check out your output.



Example 2: Interrupts

Once the library is installed, go ahead and open up **File->Examples->SparkFun Temperature Sensor - STTS22H > example2-interrupt**:



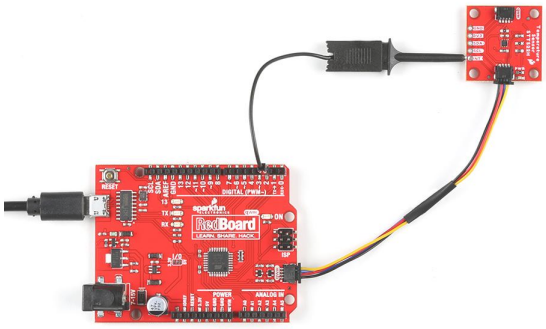
Make sure to select your board (SparkFun RedBoard) and COM port before hitting upload to begin experimenting with the air quality sensor. Alternatively, you can copy and paste the code below into a nice new Arduino sketch:

Example 2 Arduino Code

```
/*
example2_basic.ino

This example desmonstrates how to set temperature thresholds to trigger an interrupt.

Output Data Rates:

STTS22H_POWER_DOWN
STTS22H_ONE_SHOT
STTS22H_1Hz
STTS22H_25Hz
STTS22H_50Hz
STTS22H_100Hz
STTS22H_200Hz

Written by:
Elias Santistevan @ SparkFun Electronics December, 2022

Products:
   SparkFun Temperature Sensor - STTS2H             https://www.sparkfun.com/products/21262
   SparkFun Micro Temperature Sensor - STTS2H       https://www.sparkfun.com/products/21051

Repository:
     https://github.com/sparkfun/SparkFun_STTS22H_Arduino_Library

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

*/

#include <Wire.h>
#include "SparkFun_STTS22H.h"

SparkFun_STTS22H mySTTS;

float temp;

// These values are in Farenheit
float interruptHighValue = 90.5;
float interruptLowValue = 42.0;

int tempInterrupt = 1;

void setup()
{

    Wire.begin();

    Serial.begin(115200);

    pinMode(tempInterrupt, INPUT);

    if( !mySTTS.begin() )
    {
        Serial.println("Did not begin.");
        while(1);
    }

    Serial.println("Ready");

    // Other output data rates can be found in the description
    // above. To change the ODR or mode, the device must first be
    // powered down.
    mySTTS.setDataRate(STTS22H_POWER_DOWN);
    delay(10);
    mySTTS.setDataRate(STTS22H_25Hz);

    // Enables incrementing register behavior for the IC.
    // It is not enabled by default as the datsheet states and
    // is vital for reading the two temperature registers.
    mySTTS.enableAutoIncrement();


    // Set interrupts for both lower and higher thresholds.
    // Note: These functions accept Farenheit as their arguments.
    // Other functions for different units just below.
    mySTTS.setInterruptLowF(interruptLowValue);
    mySTTS.setInterruptHighF(interruptHighValue);

    //mySTTS.setInterruptLowC(interruptLowValue);
    //mySTTS.setInterruptHighC(interruptHighValue);

    //mySTTS.setInterruptLowK(interruptLowValue);
    //mySTTS.setInterruptHighK(interruptHighValue);

    delay(100);
}

void loop()
{

    // Checking if data ready is not necessary when output is set higher
    // than 1Hz.
    mySTTS.getTemperatureF(&temp);

    // Temperature in different units can be retrieved
```

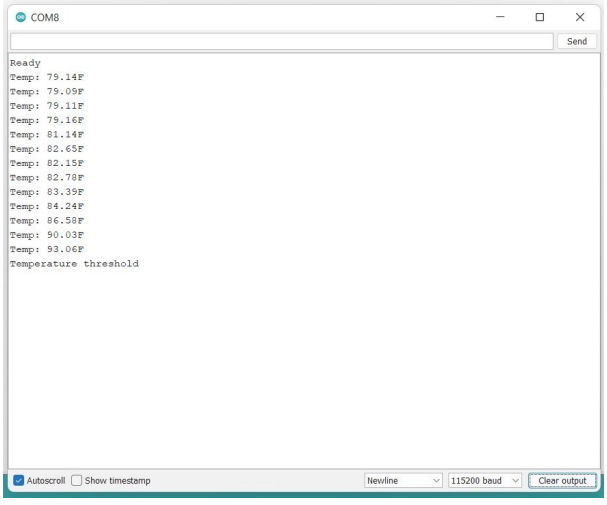
```
    // using the following functions.

    //mySTTS.getTemperatureC(&temp);
    //mySTTS.getTemperatureK(&temp);

    Serial.print("Temp: ");
    Serial.print(temp);
    Serial.println("F");

    if( digitalRead(tempInterrupt) == LOW )
    {
        Serial.println("Temperature threshold");
        while(1);
    }

    // delay = 1/ODR
    delay(40);

}
```

Note that depending on which processor board you are using, you may need to alter the Interrupt Pin. Since we're using a RedBoard here, our Interrupt Pin is 2 (`ensInt = 2`). Also, in this example, we've used an IC hook with a pigtail to connect the Interrupt Pin to the RedBoard pin 2, but you can also solder headers to the STTS22H Temperature Sensor so you can use the interrupt pin. Your hardware hookup should look something like the following:

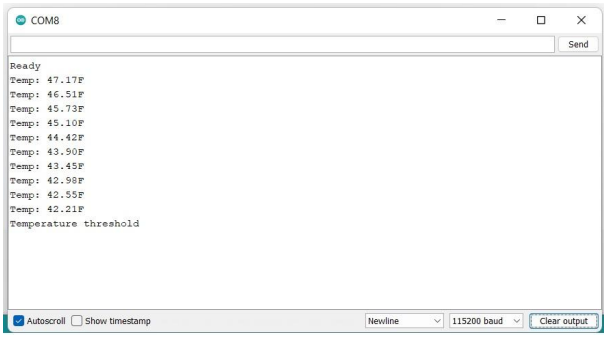


Once you've got your code uploaded, open up a Serial Monitor and check out your output.

If you have a look at the code, you'll notice that we've set our upper threshhold to 90.5 degrees F, and our lower threshhold to 42 degrees F. I held the sensor in front of a heater to hit the upper threshhold:



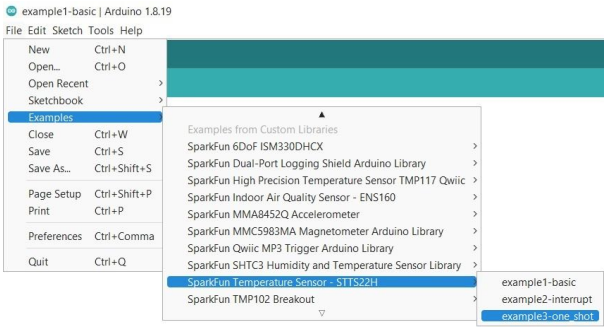
The lower threshhold was reached by sticking the sensor in a plastic bag and then putting that plastic bag into ice water:



Example 3: One Shot

The One-Shot operating mode of the STTS22H allows for the temperature measurement to be made and then the device puts itself in a power-down condition. In one-shot mode, the sensor current consumption falls to 1.75 µA, though the full breakout board will draw a bit higher due to the LED &etc.

Go ahead and open up **File->Examples->SparkFun Temperature Sensor - STTS22H ->example3-one_shot**. Make sure to select your board (SparkFun RedBoard) and COM port before hitting upload to begin experimenting with the air quality sensor.



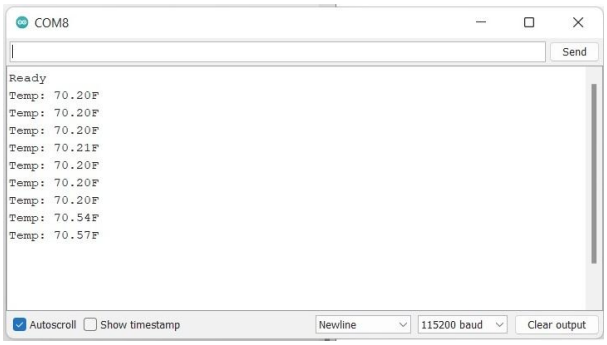
Alternatively, you can copy and paste the code below into a nice new Arduino sketch:

Example 3 Arduino Code

```
/*
example3-one_shot.ino

This example shows basic data retrieval using the "one-shot" feature i.e. - get the temp
now feature.

Output Data Rates:

STTS22H_POWER_DOWN
STTS22H_ONE_SHOT   < -------- This one.
STTS22H_1Hz
STTS22H_25Hz
STTS22H_50Hz
STTS22H_100Hz
STTS22H_200Hz

Written by:
Elias Santistevan @ SparkFun Electronics December, 2022

Products:
   SparkFun Temperature Sensor - STTS2H             https://www.sparkfun.com/products/21262
   SparkFun Micro Temperature Sensor - STTS2H       https://www.sparkfun.com/products/21051

Repository:
     https://github.com/sparkfun/SparkFun_STTS22H_Arduino_Library

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

*/

#include <Wire.h>
#include "SparkFun_STTS22H.h"

SparkFun_STTS22H mySTTS;

float temp;

void setup()
{

    Wire.begin();

    Serial.begin(115200);

    if( !mySTTS.begin() )
    {
        Serial.println("Did not begin.");
        while(1);
    }

    Serial.println("Ready");

    // Other output data rates can be found in the description
    // above. To change the ODR or mode, the device must first be
    // powered down.
    mySTTS.setDataRate(STTS22H_POWER_DOWN);
    delay(10);
    // Force new reading, temp sensor will power down after conversion.
    mySTTS.setDataRate(STTS22H_ONE_SHOT);

    // Enables incrementing register behavior for the IC.
    // It is not enabled by default as the datsheet states and
    // is vital for reading the two temperature registers.
    mySTTS.enableAutoIncrement();

    delay(100);
}

void loop()
{

    // Temp sensor will power down automatically after single read.
    if( mySTTS.dataReady() )
    {

        mySTTS.getTemperatureF(&temp);

        // Temperature in different units can be retrieved
        // using the following functions.

        //mySTTS.getTemperatureC(&temp);
        //mySTTS.getTemperatureK(&temp);

        Serial.print("Temp: ");
        Serial.print(temp);
        Serial.println("F");

        // Wait 10 seconds for until we initiate another read.
        delay(10000);

        // Enable another reading.
        mySTTS.setDataRate(STTS22H_ONE_SHOT);

    }
```

```
    // Demonstrative delay.
    delay(100);

}
```

Once you've got your code uploaded, open up a Serial Monitor and check out your output. You should see something like the following:



This really isn't all that exciting until you measure the current consumption!

2. Hardware Resources

Now that you've successfully got your STTS22H Temperature Sensor up and running, it's time to incorporate it into your own project! For more information, check out the resources below:

SparkFun Temperature Sensor - STTS22H (Qwiic):

- Schematic (PDF)
- Eagle Files
- Board Outline (PNG)

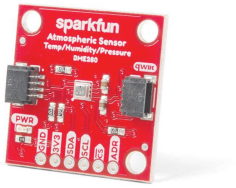
SparkFun Micro Temperature Sensor - STTS22H (Qwiic):

- Schematic (PDF)
- Eagle Files
- Board Outline (PNG)

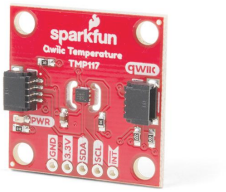
General Information:

- Datasheet(PDF)
- Qwiic Info Page
- STTS22H Arduino Library
- GitHub Hardware Repo

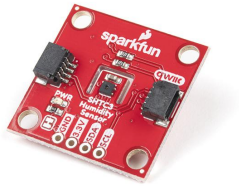
Or check out other Qwiic Sensor Tutorials:



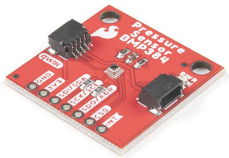
Qwiic Atmospheric Sensor (BME280) Hookup Guide



Qwiic TMP117 High Precision Digital Temperature Sensor Hookup Guide



SparkFun Humidity Sensor Breakout - SHTC3 (Qwiic) Hookup Guide



Qwiic Pressure Sensor (BMP384) Hookup Guide

3. Support

Note

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**