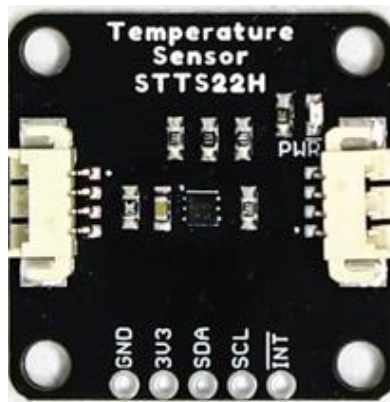




SmartElex Temperature Sensor - STTS22H

The STTS22H breakout board 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 $\pm 0.5\text{ }^{\circ}\text{C}$ without requiring any further calibration at the application level.



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 $\pm 0.5\text{ }^{\circ}\text{C}$ 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 $1.75\text{ }\mu\text{A}$. For more information, refer to the datasheet.

Connectors

There are two connectors on either side of the Temperature Sensor - STTS22H to provide power and I²C connectivity simultaneously. The I²C address of the board is **0x3C** by default, but the board has 3 other addresses the board can be configured to use.

Power

Ideally, power will be supplied via the 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.

⚡ 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.

Power LED

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

Exposed Pad

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

LED Jumper

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

Address Jumpers

The 1x1 board has two address jumpers available. The default I²C address is **0x3C**. By cutting various trace combinations, there are three other I²C addresses available.

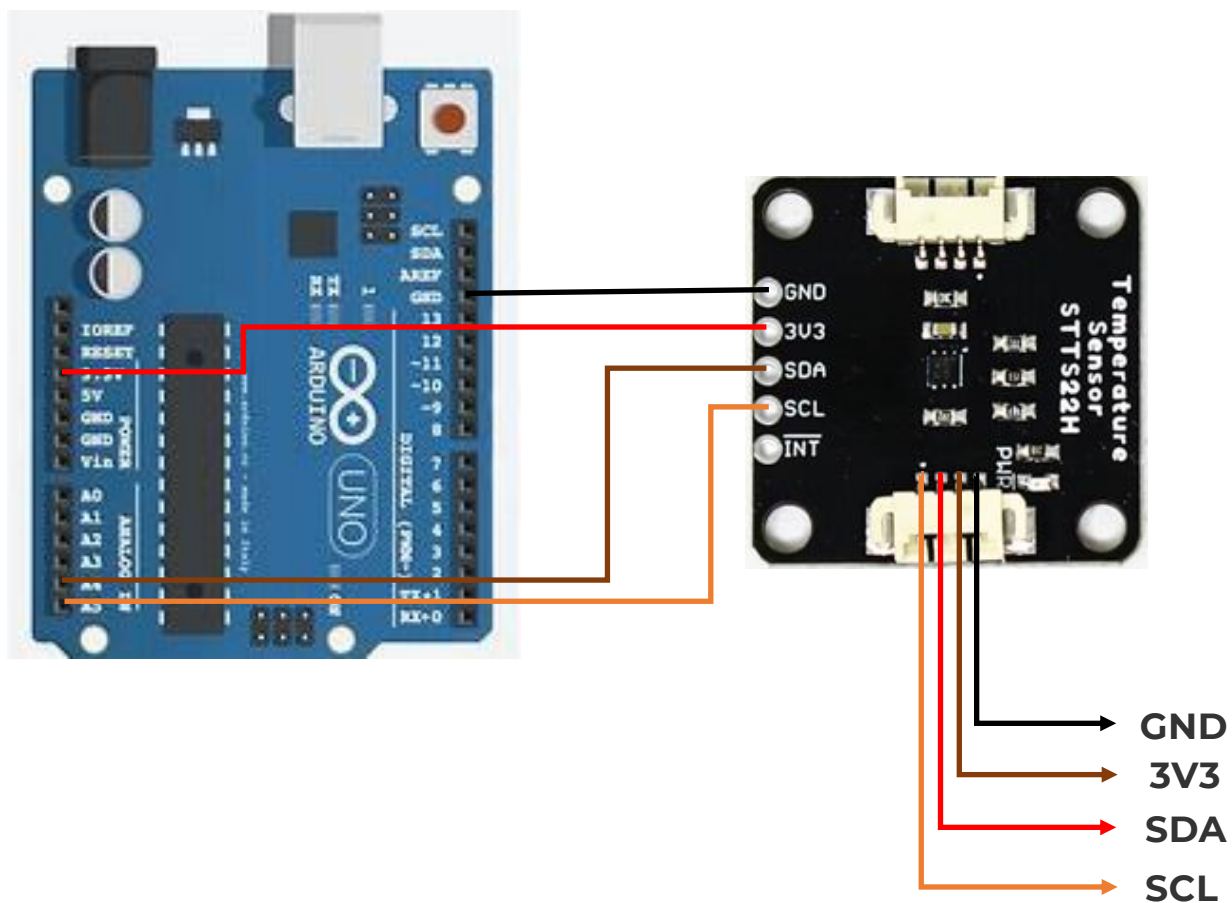
ADDR	
R8(15K)	0x3C (Default)
R7(56K)	0x3E

VDD	0x38
GND	0x3F
OPEN	Undefined

The Micro version of this board has a single address jumper that affords the ability to change the I²C address from **0x3C (Default)** to 0x38.

Wiring

Connecting the STT22H to Arduino:



Arduino	STT22H
SCL(A5)	SCL
SDA(A4)	SDA
3.3v	3V3
GND	GND

Examples

Example 1: Basic

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

```
#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);
mySTTS.setDataRate(STTS22H_1Hz);
// Enables incrementing register behavior for the IC.
// It is not enabled by default as the datasheet states and
// is vital for reading the two temperature registers.
mySTTS.enableAutoIncrement();
delay(100);
}
void loop()
{
// Only use data ready for one-shot mode or 1Hz output.
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");
```

```
}  
  
// delay = 1/ODR  
  
delay(1000);  
  
}
```

Once you're ready to go, go ahead and hit the upload button (the right facing arrow button under the "Edit" menu item). Once your code is uploaded, open the Serial Monitor and you'll see information start printing out.

