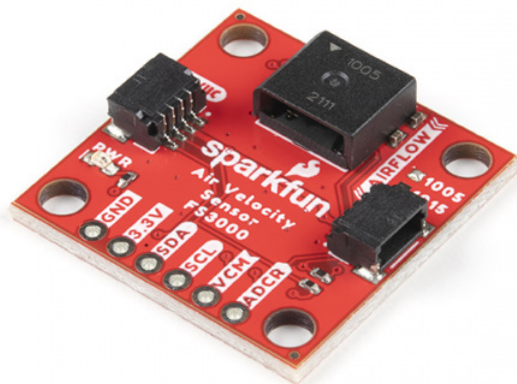


Air Velocity Sensor Breakout - FS3000 Hookup Guide

Introduction

Need to keep track of the airflow in your data center or around your servers? How about making sure your HVAC and air control systems are functioning at full capacity? Or, if you're more fun, what about figuring out how fast your RC airplane is going? Well, the new SparkFun Air Velocity Sensor Breakout - FS3000 can help you with all that and more! It's super easy, super Qwiic to hookup, and super fun to play with. Let's have a look!



SparkFun Air Velocity Sensor Breakout - FS3000-1005 (Qwiic)

© SEN-18377




Product Showcase: SparkFun Air Velocity Sensor Br...





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.

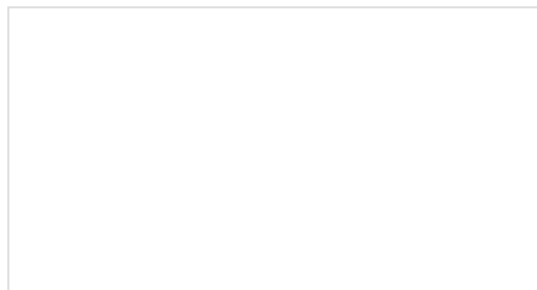
| Air Velocity Sensor Breakout - FS3000 Wish List SparkFun Wish List | |
|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
|  | SparkFun Air Velocity Sensor Breakout - FS3000-1005 (Qwiic) SEN-18377 |
|  | Qwiic Cable - 100mm PRT-14427 |
|  | SparkFun RedBoard Qwiic DEV-15123 |

Suggested Reading

If you aren't familiar with the Qwiic system, take a look here for an overview.

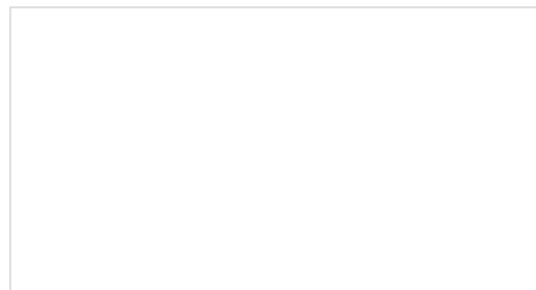


We also recommend taking a look at the following tutorials if you aren't familiar with the concepts covered in them.



What is an Arduino?

What is this 'Arduino' thing anyway? This tutorial dives into what an Arduino is and along with Arduino projects and widgets.



Installing Arduino IDE

A step-by-step guide to installing and testing the Arduino software on Windows, Mac, and Linux.



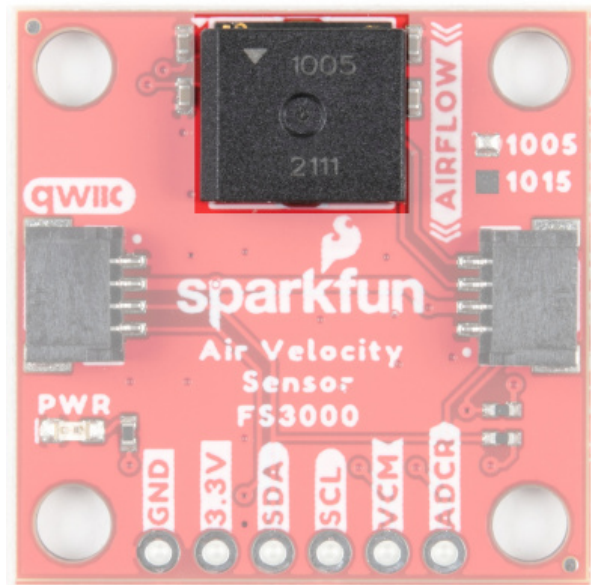
Logic Levels

Learn the difference between 3.3V and 5V devices and logic levels.

Hardware Overview

FS3000

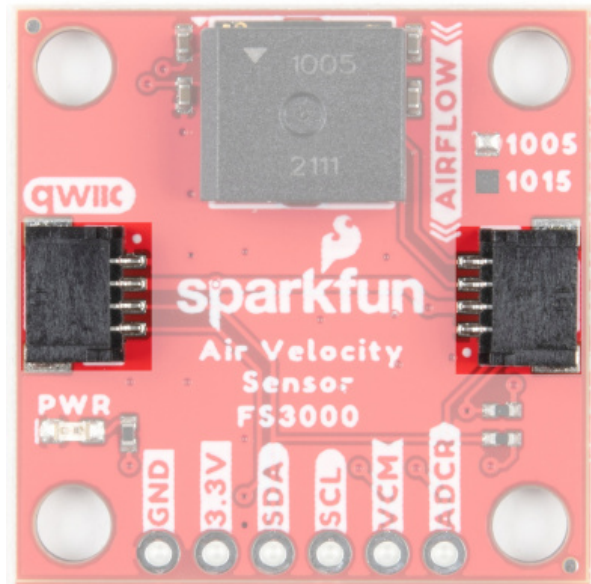
The FS3000 is a surface-mount type air velocity module utilizing a MEMS thermopile-based sensor. It features a digital output with 12-bit resolution and comprises a “solid” thermal isolation technology and silicon carbide coating to protect it from abrasive wear and water condensation. For more in depth information on this chip, please refer to the datasheet.



Qwiic Connectors

Our Qwiic Ecosystem makes sensors pretty much plug and play. There are two Qwiic connectors on either side of the Qwiic Air Velocity Sensor board to provide power and I²C connectivity simultaneously.

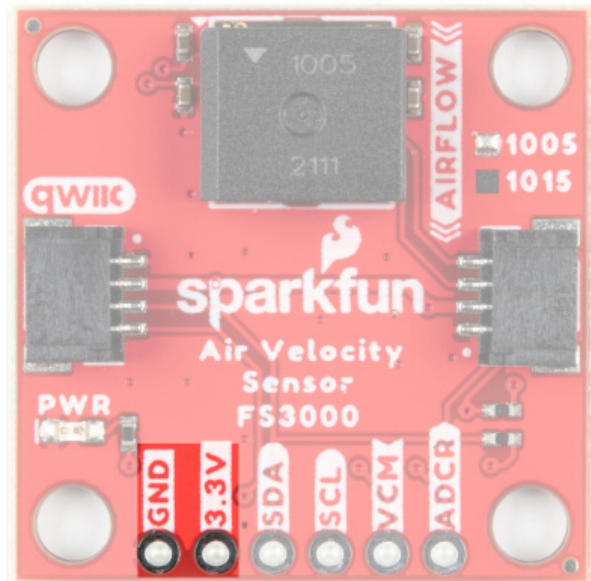
The I²C address of the board is **0x28**.



Pins

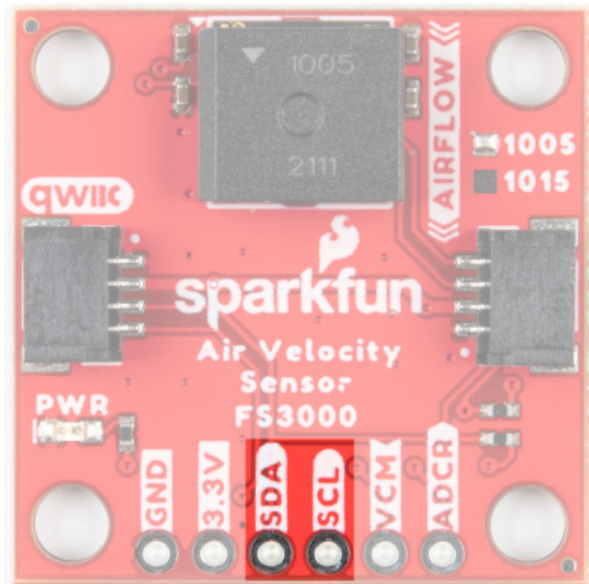
Power

Ideally, power will be supplied via the Qwiic connectors on either side of the board. 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 **2.7-3.3V**. The usual current draw is around 10mA.



I²C

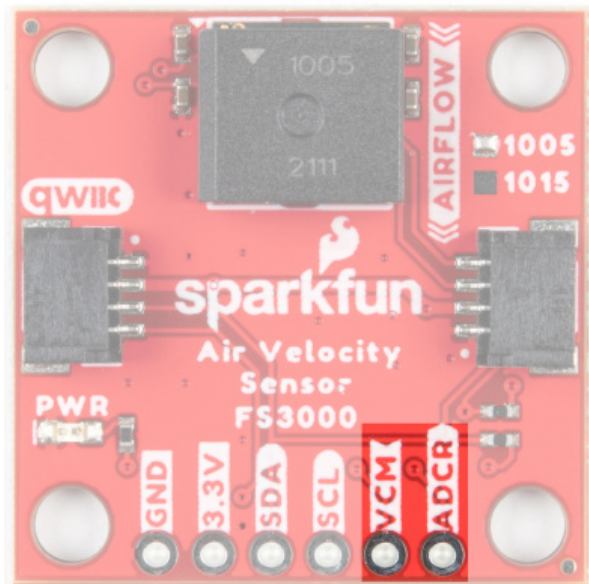
The I²C pins break out the functionality of the Qwiic connectors. Depending on your application, you can connect to these pins via the plated through holes for SDA and SCL.



VCM and ADCR

VCM is an output from the sensor that provides a common bias voltage. With a power supply voltage at 3.3V (usual for Qwiic), the VCM pin will output 1.25V. The current datasheet does not provide any more information about this pin, however, it is most likely a bias voltage used with some sort of analog gain stage internal to the sensor (prior to it's internal ADC). This can be useful for unique more advanced projects that may benefit from having a known bias voltage tied to another analog system.

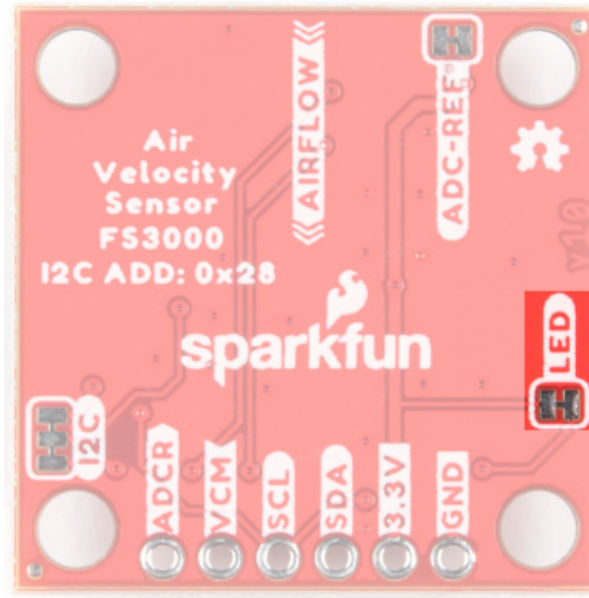
ADCR stands for ADC Reference. This is an input to the FS3000 which allows you to provide a reference voltage for the sensor's internal ADC. Note, you must first cut the "ADC-REF" jumper (on the bottom side of the board) before providing a custom voltage to this pin. By default, it is connected to VDD.



Jumpers

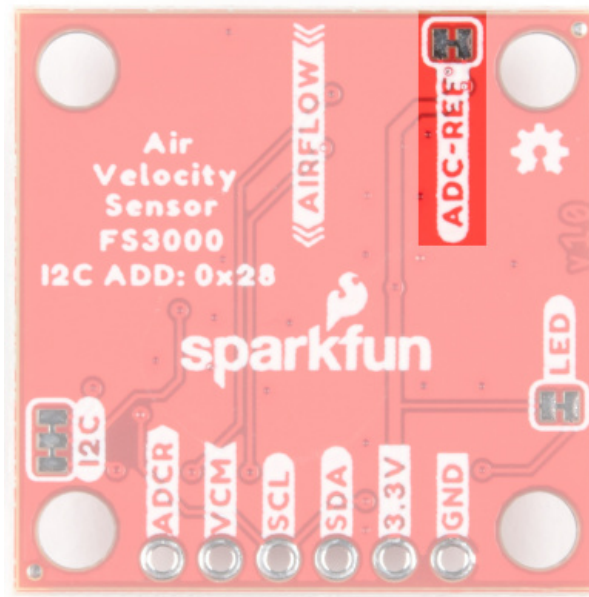
LED

If power consumption is an issue, cutting this jumper will disable the Power LED on the front of the board.



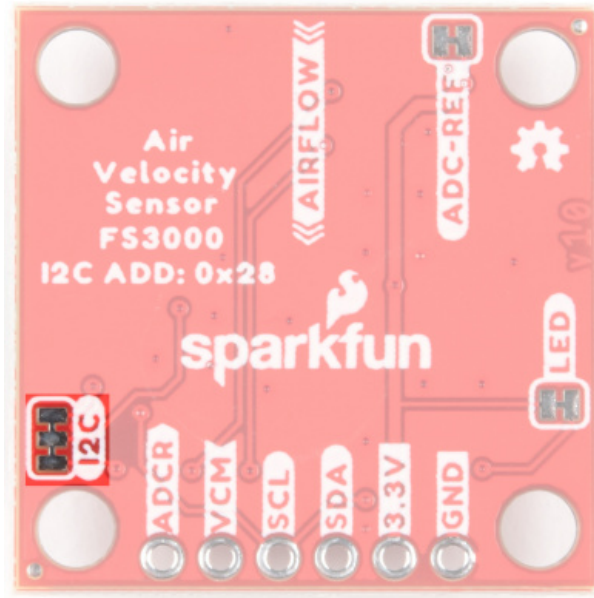
ADC-REF

Cut the ADC-REF jumper to provide your own custom ADC reference voltage on the ADCR header pin. By default (jumper closed), it is connected to VDD.

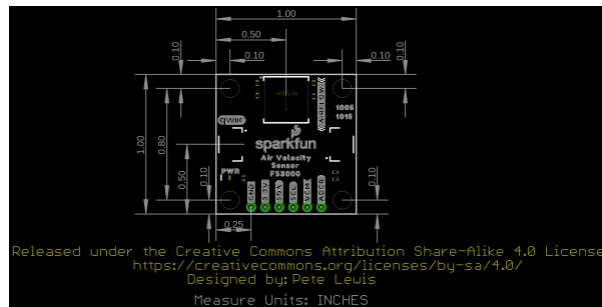


I2C

The Sparkfun Air Velocity Sensor has pull up resistors attached to the I²C bus; 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. If you need to disconnect the pull up resistors they can be removed by cutting the traces on the corresponding jumpers highlighted below.

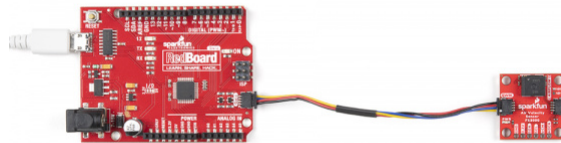


Board Outline



Hardware Hookup

Using the Qwiic system, assembling the hardware is simple. All you need to do is connect your Air Velocity Sensor Breakout to your chosen development board with a Qwiic cable or adapter cable. Otherwise, you can use the I²C pins broken out if you do not have a Qwiic connector on your development board or if you do not want to use a Qwiic connection. If you are not using a Qwiic-enabled board, make sure your input voltage and logic are either running at **3.3V** or you are shifting the logic level from whatever logic your controller runs at to **3.3V**.



Software Setup and Programming

Note: Make sure you are using the latest stable 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 Qwiic Air Velocity Sensor. You can obtain this library through the Arduino Library Manager by searching for "FS3000". 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 or by clicking on the button below.

SPARKFUN AIR VELOCITY SENSOR BREAKOUT - FS3000 (QWIIC) ARDUINO LIBRARY
GITHUB

Readings

The FS3000 has a response time of 125ms. In the provided example, we are only taking readings from the sensor once per second, so this is well above the response time. If you wish to take a higher frequency of readings, make sure to add in at least a 125ms delay in between each read.

To calculate or understand airflow and how it relates to diameter, you can use this tool.

Functions

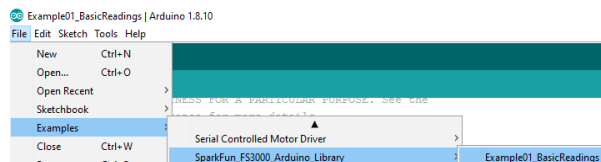
Below is a list of the functions that can be used with the Air Velocity Sensor, along with a description of what each function does and how to use it.

- **FS3000()** - Base constructor
- **begin()** -- Initialize the sensor. Returns false if sensor is not detected.
- **isConnected()** -- Returns true if I2C device ack's
- **readRaw()** -- Read from sensor, checksum, return raw data (409-3686)
- **readMetersPerSecond()** -- Read from sensor, checksum, return m/s (0-7.23)
- **readMilesPerHour()** -- Read from sensor, checksum, return mph (0-33ish)

Note: If you are using a board with an Apollo core, please note that 100KHz on the Apollo 2.1.1 doesn't currently work. You will need to use 400KHz, or jump back to Apollo 2.1.0.

Example Code

Once you've installed the FS3000 library, you should see **File > Examples > SparkFun_FS3000_Arduino_Library > Example01_BasicReadings** to open the example sketch.



Alternatively, you can copy and paste the code below into a fresh Arduino sketch.


```
/******
```

```
Example_01_BasicReadings.ino
```

```
Read values of air velocity from the FS3000 sensor, print them to terminal.
```

```
Prints raw data, m/s and mph.
```

```
Note, the response time on the sensor is 125ms.
```

```
SparkFun FS3000 Arduino Library
```

```
Pete Lewis @ SparkFun Electronics
```

```
Original Creation Date: August 5th, 2021
```

```
https://github.com/sparkfun/SparkFun\_FS3000\_Arduino\_Library
```

```
Development environment specifics:
```

```
IDE: Arduino 1.8.15
```

```
Hardware Platform: SparkFun Redboard Qwiic
```

```
SparkFun Air Velocity Sensor Breakout - FS3000 (Qwiic) Version: 1.0
```

```
Artemis Redboard @ 400KHz (Core v2.1.0)
```

```
(note, v2.1.1 has a known issue with clock stretching at 100KHz)
```

```
Do you like this library? Help support SparkFun. Buy a board!
```

```
SparkFun Air Velocity Sensor Breakout - FS3000 (Qwiic)
```

```
https://www.sparkfun.com/products/18377
```

```
Hardware Connections:
```

```
Use a qwiic cable to connect from the Redboard Qwiic to the FS3000 breakout (QWIIC).
```

```
You can also choose to wire up the connections using the header pins like so:
```

```
ARDUINO --> FS3000
```

```
SDA (A4) --> SDA
```

```
SCL (A5) --> SCL
```

```
3.3V --> 3.3V
```

```
GND --> GND
```

```
This program is distributed in the hope that it will be useful,
```

```
but WITHOUT ANY WARRANTY; without even the implied warranty of
```

```
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
```

```
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
```

```
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

```
*****/
```

```
#include <Wire.h>
```

```
#include <SparkFun_FS3000_Arduino_Library.h> //Click here to get the library: http://librarymanager/All#SparkFun\_FS3000
```

```
FS3000 fs;
```

```
void setup()
```

```
{
```

```

Serial.begin(115200);
Serial.println("Example 1 - Reading values from the FS3000");

Wire.begin();

if (fs.begin() == false) //Begin communication over I2C
{
  Serial.println("The sensor did not respond. Please check wiring.");
  while(1); //Freeze
}
Serial.println("Sensor is connected properly.");
}

void loop()
{
  Serial.print("FS3000 Readings \tRaw: ");
  Serial.print(fs.readRaw()); // note, this returns an int from 0-3686

  Serial.print("\tm/s: ");
  Serial.print(fs.readMetersPerSecond()); // note, this returns a float from 0-7.23

  Serial.print("\tmph: ");
  Serial.println(fs.readMilesPerHour()); // note, this returns a float from 0-16.17

  delay(1000); // note, response time on the sensor is 125ms
}

```

Set your Board and Serial Port, and then upload the sketch to your Arduino. Then **open the serial monitor**. Make sure your **baud rate** is set to *115200*. You'll begin to see output, including raw data and translated miles per second and miles per hour.

```

COM11
Example 1 - Reading values from the FS3000
Sensor is connected properly.
FS3000 Readings      Raw: 431      m/s: 0.05      mph: 0.10
FS3000 Readings      Raw: 432      m/s: 0.05      mph: 0.11
FS3000 Readings      Raw: 428      m/s: 0.04      mph: 0.09
FS3000 Readings      Raw: 405      m/s: 0.00      mph: 0.00
FS3000 Readings      Raw: 428      m/s: 0.04      mph: 0.09
FS3000 Readings      Raw: 2292     m/s: 3.49      mph: 7.82
FS3000 Readings      Raw: 474      m/s: 0.14      mph: 0.31
FS3000 Readings      Raw: 440      m/s: 0.07      mph: 0.15
FS3000 Readings      Raw: 442      m/s: 0.07      mph: 0.16
FS3000 Readings      Raw: 422      m/s: 0.03      mph: 0.06
FS3000 Readings      Raw: 431      m/s: 0.05      mph: 0.10

```

Note that peak in the middle of the readings? That's where I pointed the sensor at the fan!

Troubleshooting

Need help?

If your product is not working as you expected or you need technical assistance or information, head on over to the SparkFun Technical Assistance page for some initial troubleshooting.

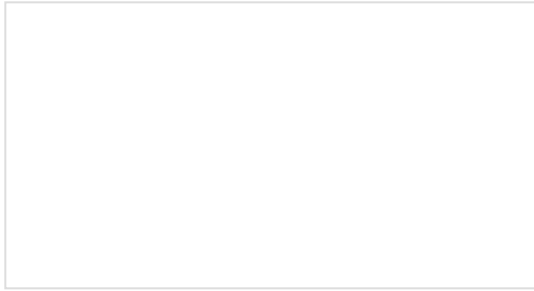
If you don't find what you need there, the SparkFun Forums are a great place to find and ask for help. If this is your first visit, you'll need to create a Forum Account to search product forums and post questions.

Resources and Going Further

For more on the Air Velocity Sensor Breakout, check out some of the links below:

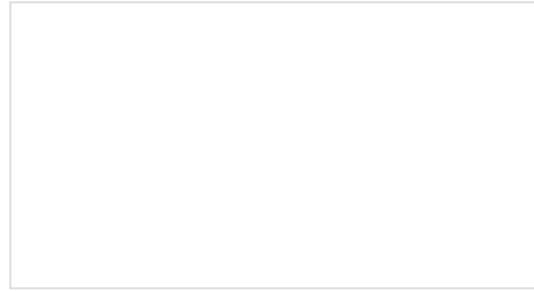
- [Schematic \(PDF\)](#)
- [Eagle Files \(ZIP\)](#)
- [Board Dimensions](#)
- [Datasheet \(PDF\)](#)
- [GitHub](#)
- [Arduino Library GitHub](#)

Need some inspiration for your next project? Check out some of these related tutorials:



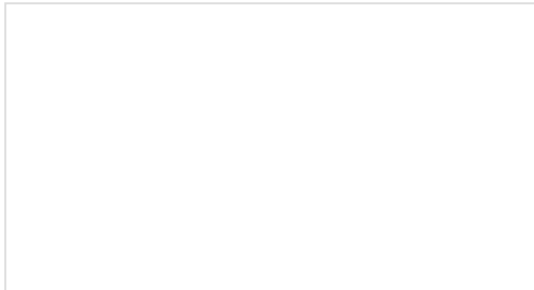
SparkFun GPS Breakout - XA1110 (Qwiic) Hookup Guide

Figure out where in the world you are with the Qwiic SparkFun GPS Breakout - XA1110.



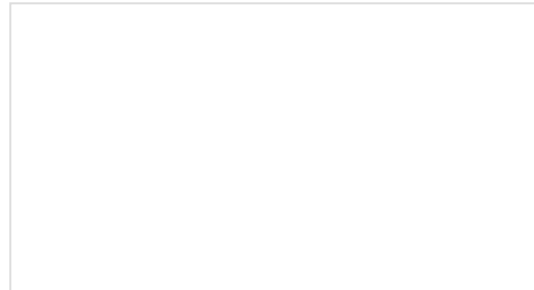
Qwiic Real Time Clock Module (RV-1805) Hookup Guide

Find out what time it is, even after the power's been out on your project for a while with the Qwiic Real Time Clock (RTC) module.



Pi Servo pHAT (v2) Hookup Guide

This hookup guide will get you started with connecting and using the Pi Servo pHAT on a Raspberry Pi.



Qwiic EEPROM Hookup Guide

Running out of memory? Don't forget to check out the Qwiic EEPROM!