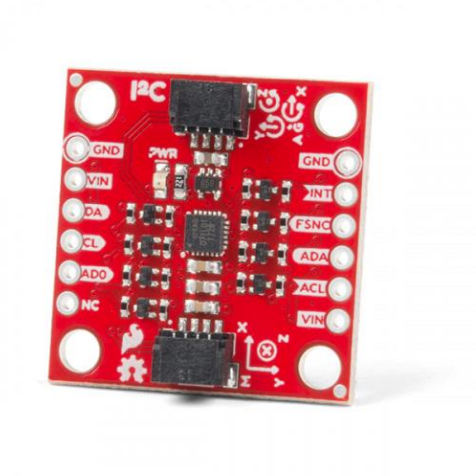
SparkFun 9DoF IMU (ICM-20948) Breakout Hookup Guide

Introduction

The SparkFun 9DoF IMU Breakout incorporates all the amazing features of Invensense's ICM-20948 into a Qwiic-enabled breakout board replete with logic shifting and broken out GPIO pins for all your motion sensing needs. The ICM-20948 itself is an extremely low powered, I^{2}C and SPI enabled 9-axis motion tracking device that is ideally suited for smartphones, tablets, wearable sensors, and IoT applications. Featuring a 3-Axis Gyroscope with four selectable ranges, a 3-Axis Accelerometer, again with four selectable ranges, a 3-axis compass with a wide range to ±4900 µT, and an on-board Digital Motion Processor, this little breakout can even detect the motion of invisibility cloaks. Not really. Just checking to see if you were still with me. But it *is* pretty amazing. Check it out:



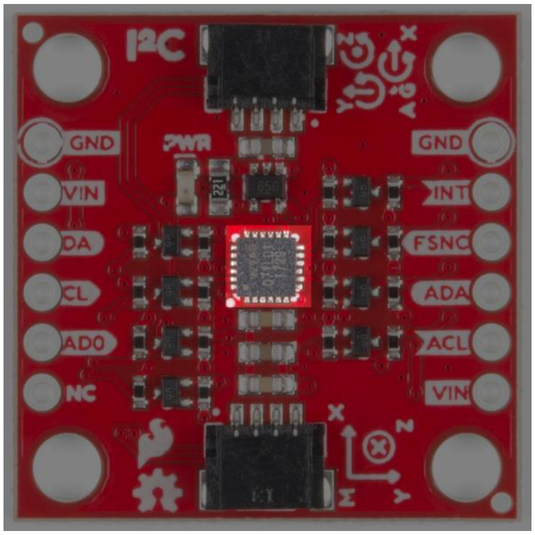
In this hookup guide, we'll connect our sensor up to our Esp32 Thing Plus microcontroller and run a quick (Qwiic) example to get you up and running with this fantastic board!

Hardware Overview

We've put a lot of effort into making this the most useful and versatile breakout for the ICM 20948. Let's take a closer look at all the special parts.

Sensor

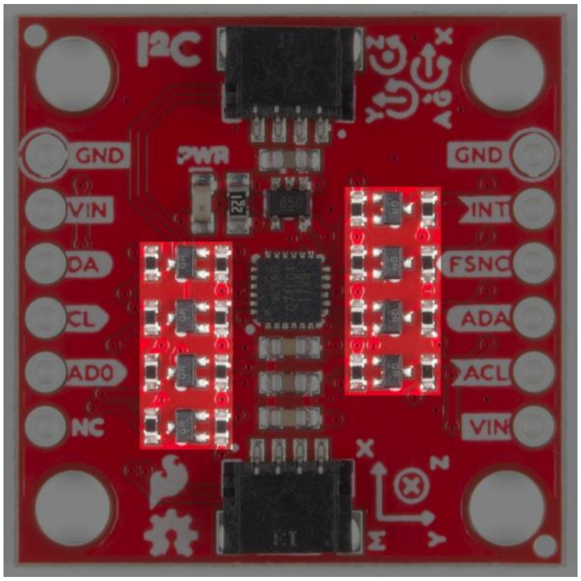
At the heart of the board (metaphorically and geometrically) is the ICM 20948 from Invensense. This puppy packs the ability to measure up to 10 unique values (3 axes of acceleration, rotational rate, and magnetic strength data as well as an on-board temperature sensor). The sensor is placed dead-center between the four 4-40 stand-off mounting holes to drastically simplify computation in dynamics.



ICM 20948 Sensor

Level Shifters

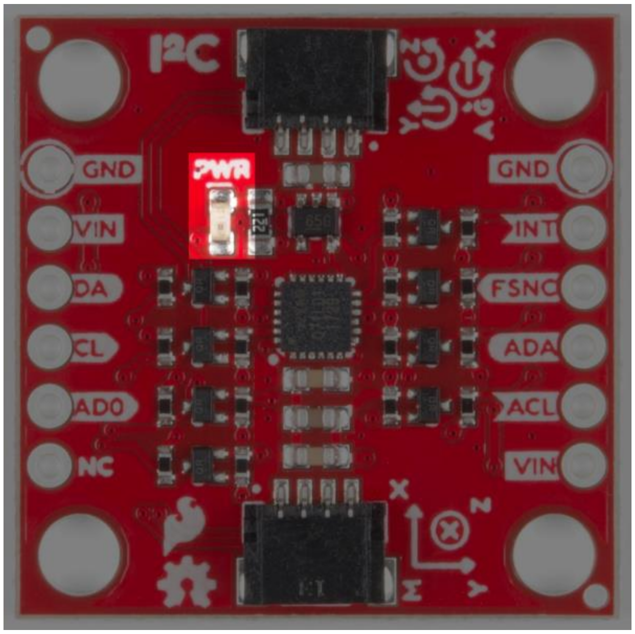
The ICM is a fickle fellow - optionally allowing a **3.3V** supply voltage but requiring I/O to work at 1.8V. This is just the price we pay for amazing technolojay (hey that rhymes). Since there aren't many popular development boards that run at the voltage of the future we've added high speed level shifting to each and every IO pin. These cool MOSFETS allow for bi-directional voltage translation up to the maximum SPI speed of the ICM - 7MHz - which will allow you to make inertial measurements with fantastic temporal resolution. Feel free to use the ICM IO anywhere from 1.8V to 5.5V!



TXS0108 Modules

Power

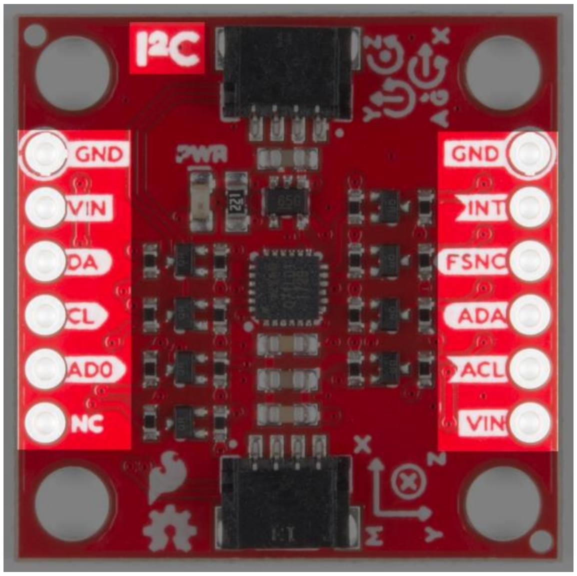
Input power on this board should be between **1.8-5.5V**. The ICM is riding the wave of 1.8V level devices so we've included a built-in regulator to make it easy to interface with 3.3V or 5V microcontrollers. There is an LED on the front of the board that will light up when the board is powered correctly. You can disable the LED functionality by cutting the LED jumper on the back of the board. This is described in the ***Jumpers*** section below.



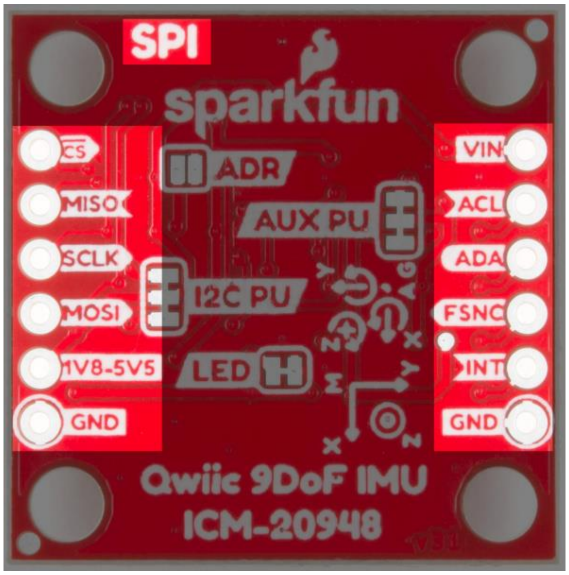
Power LED

GPIO

For flexibility, we've broken out functional pins for both I^{2}C and SPI. There are no modifications required to switch between I^{2}C mode and SPI out of the box, but if the 'ADR' jumper on the back is closed SPI will be unavailable.



I²C Pin Labels



SPI Pin Labels

Jumpers

Look at all those jumpers on the back of the board! Here's what they do:

Pullup Jumpers

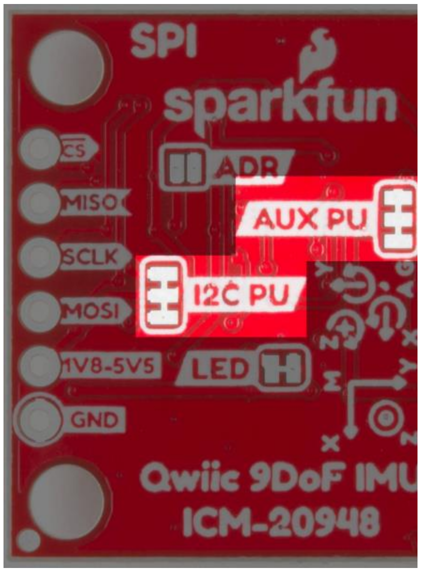
- **I²C Pullup** - Does nothing, the pullups are not populated on the board because the TXS0108 has them built-in
- **Aux Pullup** - Cut these jumpers to disconnect the pullup resistors from the auxiliary I²C bus

LED Jumper

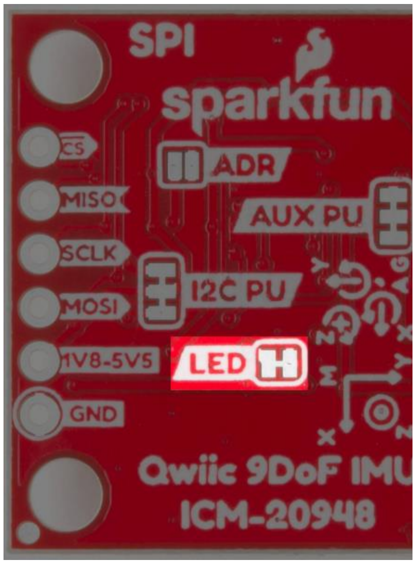
- Cutting this jumper allows you to disable the LED functionality on the front of the board.

Address Jumper

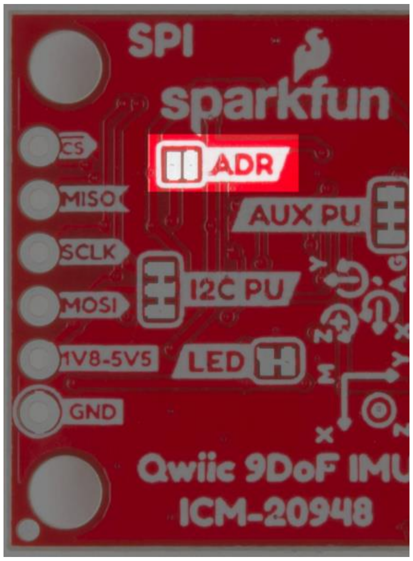
- When open (default) the address of the ICM is 0x69 and it is possible to use SPI communication. When soldered closed the address changes to 0x68. Closing the jumper prevents you from using SPI.



Pullup Jumpers



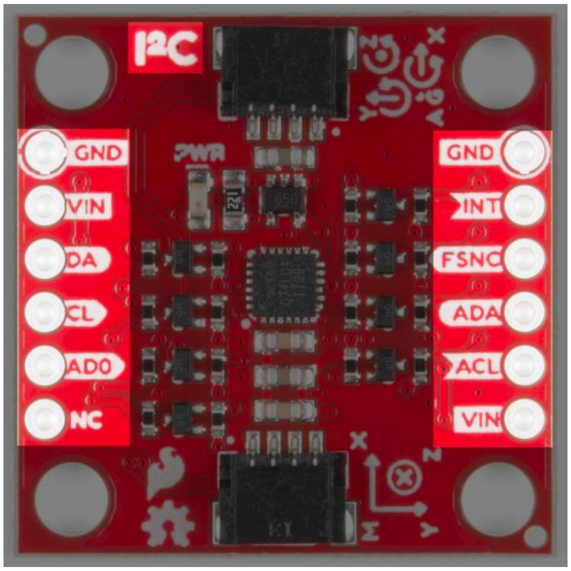
LED Jumper



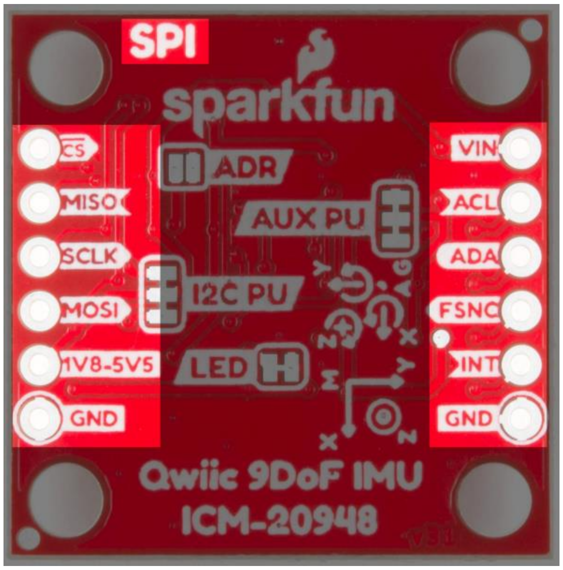
I^{2}C Address Jumper

Hardware Hookup

we've organized them by function, and provided lots of labels. You'll first notice that one side has the text 'I^{2}C' and the other side says 'SPI.' The labels on either side are those that apply to that kind of communication.



I^{2}C Pin Labels



SPI Pin Labels

Next, you'll see that on the left side are the main connections to the host microcontroller. When connecting I^{2}C you'll have a 'No Connect' pin that serves as the chip select when using SPI. As noted before, there are no modifications required to switch between I^{2}C mode and SPI out of the box. However if the 'ADR' jumper is closed SPI will be unavailable.

On the right side are connections to external sensors that can be controlled by the ICM, as well as the 'INT' and 'FSYNC' interrupt pins. The auxiliary I^{2}C bus pins are level shifted to/from the 'VIN' level that you supply.

Breakout Board Pin Functions (SPI)

Breakout Pin	Arduino Uno	Esp32 Thing Plus	Microcontroller Pin Requirements
MOSI	11	18	Data output of chosen SPI port
SCLK	13	5	Clock output of chosen SPI port
MISO	12	19	Data input of chosen SPI port
CS	2	2	An output pin to select the ICM for SPI

Breakout Board Pin Functions (I^{2}C)

Breakout Pin	Arduino Uno	Esp32 Thing Plus	Microcontroller Pin Requirements
DA	SDA	23	Data line of chosen I^{2}C port
CL	SCL	22	Clock line of chosen I^{2}C port
AD0	-	-	Optional - use to control I^{2}C address from software

Breakout Board Pin Functions (Auxiliary I^{2}C and Interrupts)

Breakout Pin	Arduino Uno	Esp32 Thing Plus	Microcontroller Pin Requirements
ADA	-	-	Data line of auxiliary I^{2}C bus
ACL	-	-	Clock line of auxiliary I^{2}C bus
FSYNC	-	-	Optional - synchronize measurements with a signal out from the microcontroller
INT	-	-	Optional - respond to configurable interrupts in from the ICM