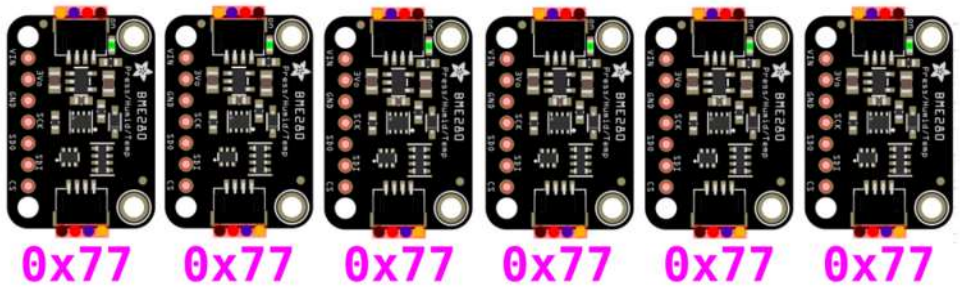
Overview



A previous guide discussed working with I2C devices in a general way, covering various topics. If you're new to I2C and need a broader overview, be sure to read that guide first, here's the link:

Working with I2C Devices

This guide goes more in depth on working with ***multiple*** copies of the same I2C device, which most likely have the **same I2C address**. Getting this general configuration working seems to be a common source of confusion.

As mentioned in the guide linked above, every I2C device on the I2C bus needs to have a unique address. If you've only used a single I2C device, you may not even have realized this. Most drivers are written such that they use, without any further input from the user, a predefined default address. This may also have even been the case for using multiple, but different, I2C devices. In that case, each device's default address was different, so there was nothing else to do in terms of coding. Everything was just plug-and-play simple.

However, things get more interesting when address conflicts start happening. This is essentially unavoidable when attempting to use **multiple copies of the same I2C device**. As mentioned in the other guide, there are three main ways of dealing with this:

- Use an alternate address (if device allows)
- Use an I2C channel multiplexer if alternate address(es) not possible (recommended)
- Use alternate I2C ports (if desperate)

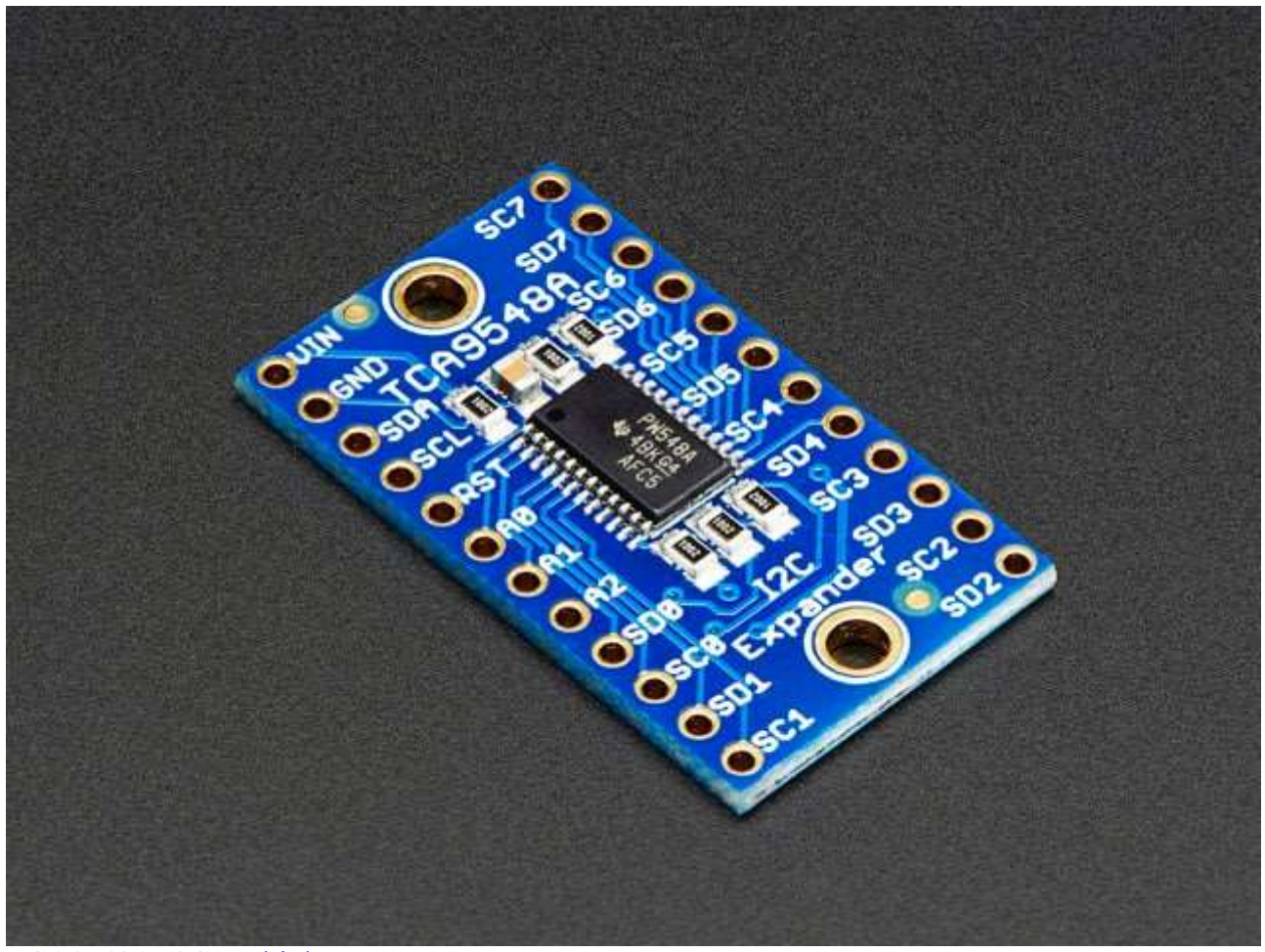
With any of these approaches, there is more work to be done. What is the alternate address? How do you actually "set" it? How do you change code to accommodate the alternate address? How do you incorporate using an I2C channel multiplexer? etc.

This guide aims to help answer those questions.

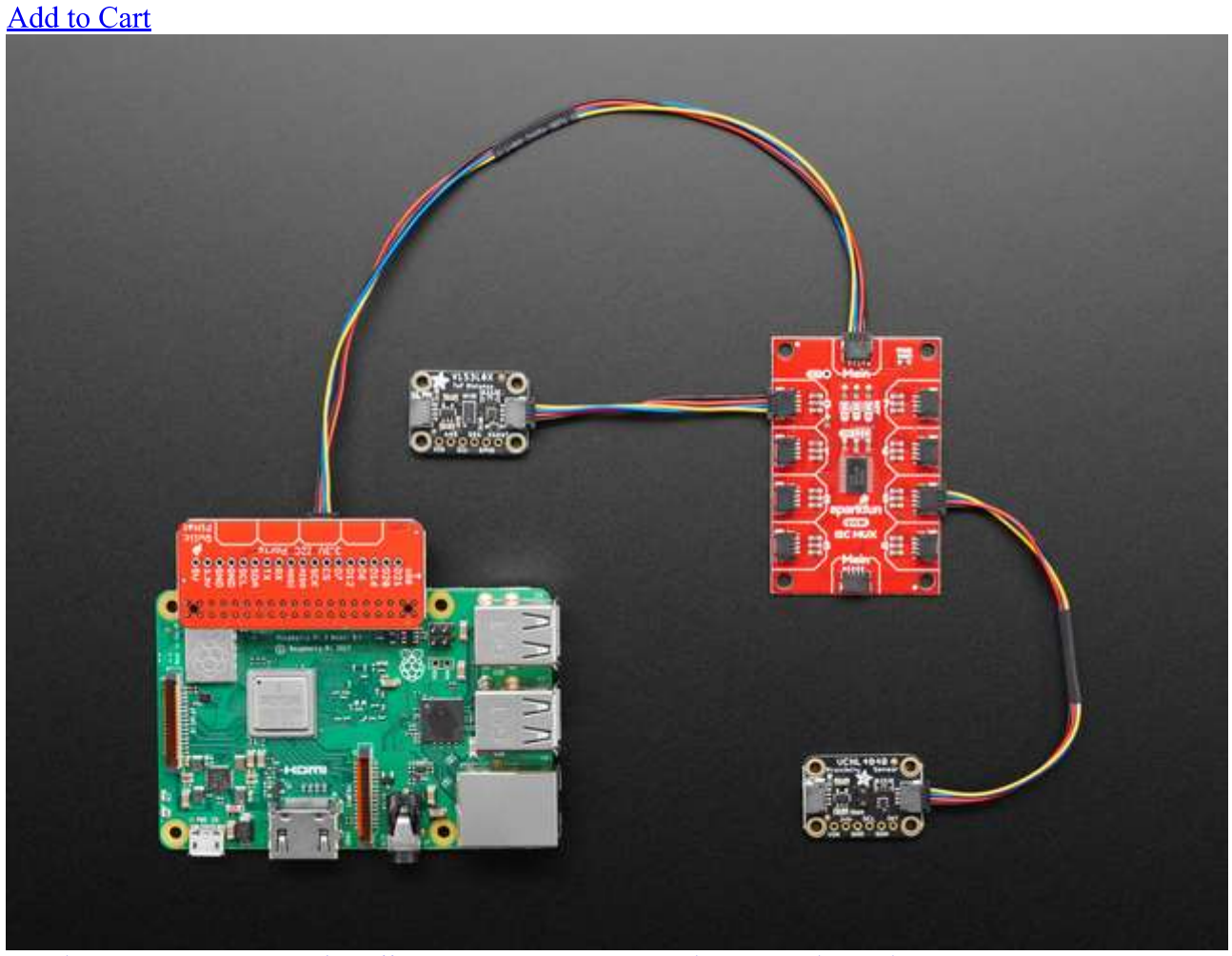
Hardware

This guide uses the Adafruit BME280 breakout in the examples as a representative I2C device. However, there is nothing special about this particular sensor. So it's not really "required hardware". The information here should generally apply to **any** I2C breakout.

When it comes to an I2C multiplexer, the TCA9548A is currently the best option. It's available in a couple of different breakouts:



[TCA9548A I2C Multiplexer]
You just found the perfect I2C sensor, and you want to wire up two or three or more of them to your Arduino when you realize "Uh oh, this chip has a fixed I2C address, and from...
$6.95
In Stock
[Add to Cart]



[SparkFun STEMMA QT / Qwiic TCA9548A Mux Breakout - 8 Channel]
Do you have too many sensors with the same I2C address? Put them on the SparkFun Qwiic Mux Breakout to get them all talking on the same bus! The Qwiic Mux Breakout...
Out of Stock
[Out of Stock]

The [Adafruit TCA9548A breakout]is used in this guide. However, the SparkFun breakout has the benefit of not needing any soldering. Everything can be connected using STEMMA QT cables.

Also, an [ItsyBitsy M4] is shown in the wiring diagrams. However, the information in this guide should apply to **any** Arduino or CircuitPython board with an I2C port. Which is pretty much most of them.