Setup Raspberry Pi:
Used Canakit Raspberry Pi 4
Straightforward setup - connected to room WiFi

Setting up sensors:
https://learn.adafruit.com/adxl343-breakout-learning-guide/assembly
Pre-solder:

Installed Azure CLI:
https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-macos

Followed this to get raspberry pi set up as a device
https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-raspberry-pi-kit-node-get-started

But, stopped at section which sends message using node js
Instead, used the python SDK:
https://github.com/Azure/azure-iot-sdk-python

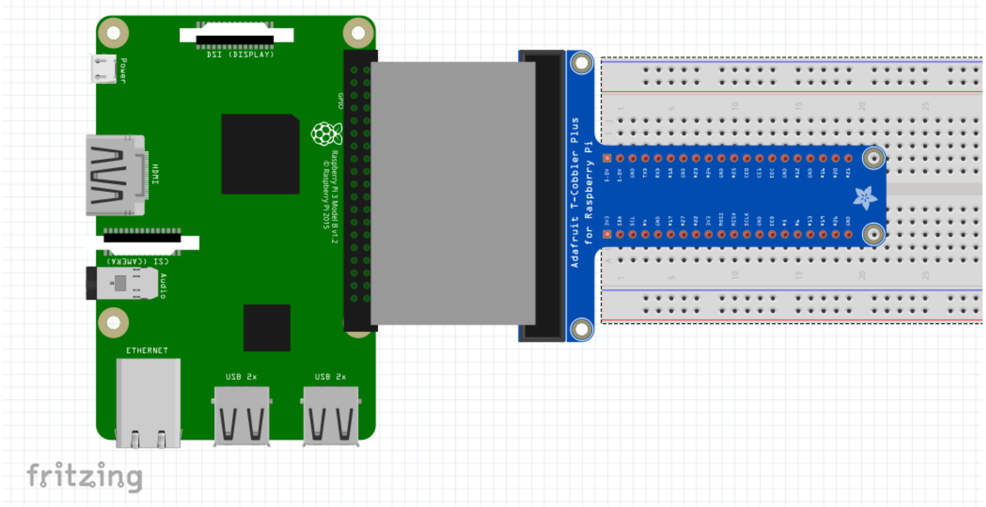
Sent dummy message from Pi to Azure:
Quick start guide here:
https://github.com/Azure/azure-iot-sdk-python/tree/main/azure-iot-device/samples

Basic order of guides:

1. Install and use IoT explorer
2. https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-macos
3. Send telemetry using SDK (only for setting up iot hub, device registration)
4. But, use python SDK on the Pi: https://github.com/Azure/azure-iot-sdk-python
5. Manage cloud device messaging with Azure IoT Tools for VS Code Use VScode extension to easily view messages / manage devices
6. https://learn.adafruit.com/adxl343-breakout-learning-guide/circuitpython Set up sensor and read data in python

Setting up the Raspberry Pi

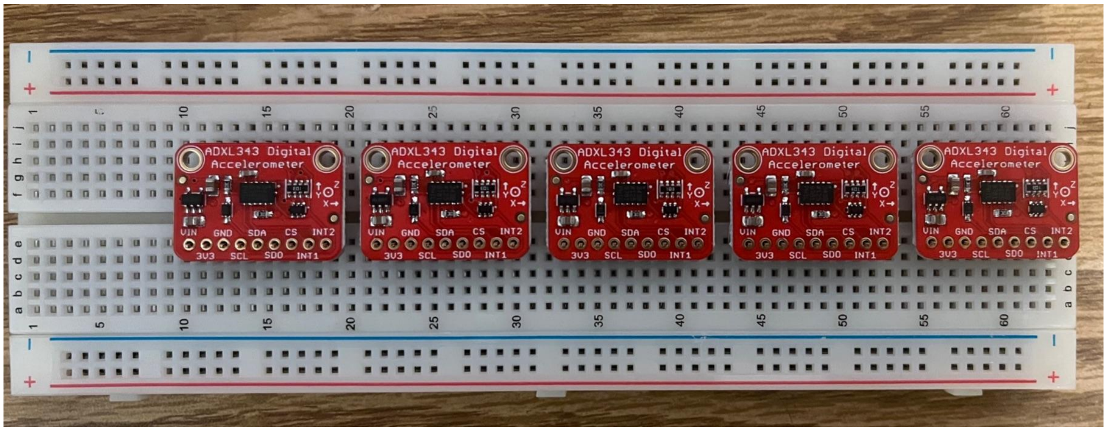
1. https://mikefrobbins.com/2020/01/22/initial-setup-of-a-canakit-raspberry-pi-4-4gb-starter-kit/ - Great guide to get the Pi4 setup and connected to a monitor
2. Connect the adafruit T-cobbler to the Pi and the breadboard



   This will make wiring easier because the pins will be clearly labelled
3. Install an IDE like https://thonny.org

Connecting multiple sensors

1. Solder the sensors (https://learn.adafruit.com/adxl343-breakout-learning-guide/assembly) for all 5



2. Install necessary CircuitPython (version of micropython that is able to interface with hardware) https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/installing-circuitpython-on-raspberry-pi
3. Enable 2 more I2C buses
4. Add 1K8 pullup resistors to the new I2C ports
5. Connect the sensors:
    a. For every I2C bus, connect two sensors, one of which has the SDO port wired to HIGH to use an alternate address
6. Run “test.py” file - see all 5 sensor readings show up

Communicating with Azure

1. Install and use IoT explorer
2. https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-macos
3. Send telemetry using SDK (only for setting up iot hub, device registration)
4. But, use python SDK on the Pi: https://github.com/Azure/azure-iot-sdk-python
5. Manage cloud device messaging with Azure IoT Tools for VS Code Use VScode extension to easily view messages / manage devices

^TODO combine together, include python scripts that merges previous step and the following:

Visualizing as a graph:

1. Send telemetry using PNP conventions using code here: https://github.com/Azure/azure-iot-sdk-python/blob/main/azure-iot-device/samples/pnp/simple_thermostat.py
2. Read from iothub and display in node using this guide https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-web-apps and my modified code.