Pololu AltIMU-10 v5 Usage with RPI

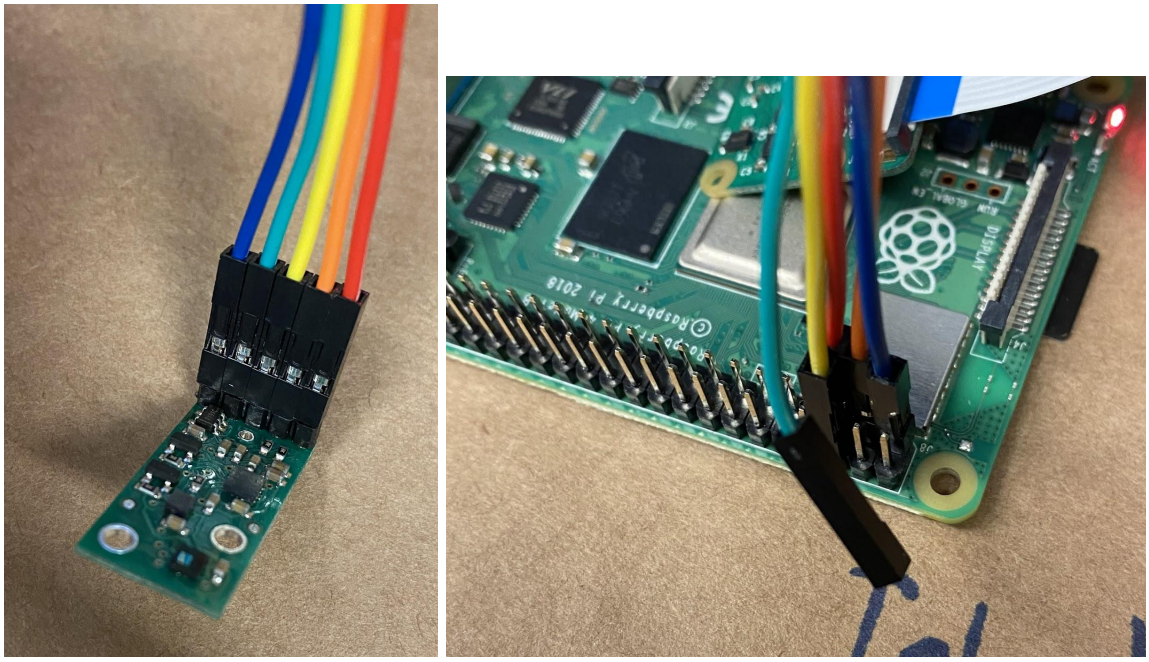
Introduction:

This is a basic tutorial on how we used the Pololu AltIMU-10 v5 with our python scripts on our Raspberry Pi 4. There are two sections to follow: hardware and software. Hardware covers the pin configuration of the IMU and RPI. Enabling I2C on RPI walks through enabling I2C on physical pins 3 and 5 of the RPI. Software covers the necessary files, imports, initialization, and functions to use the IMU data in a python script.

Hardware:

There are 5 pins on the IMU:

1. VDD: Positive voltage pin. We will use this to power the IMU with 3V. Connect this to a 3v3 Power pin on the RPI, we generally use physical pin 1.
2. VIN: Another voltage in pin. This pin will be unused for our purposes.
3. GND: Ground pin. Connect this to a Ground pin on the RPI, we generally use physical pin 6.
4. SDA: I2C data line. This is the line that allows the RPI and IMU to communicate and allows us to read data from the IMU. Connect this to physical pin 3 of the RPI.
5. SCL: I2C clock line. This line uses the RPI processor clock as a clock for the I2C communication. Connect this to physical pin 5 of the RPI.



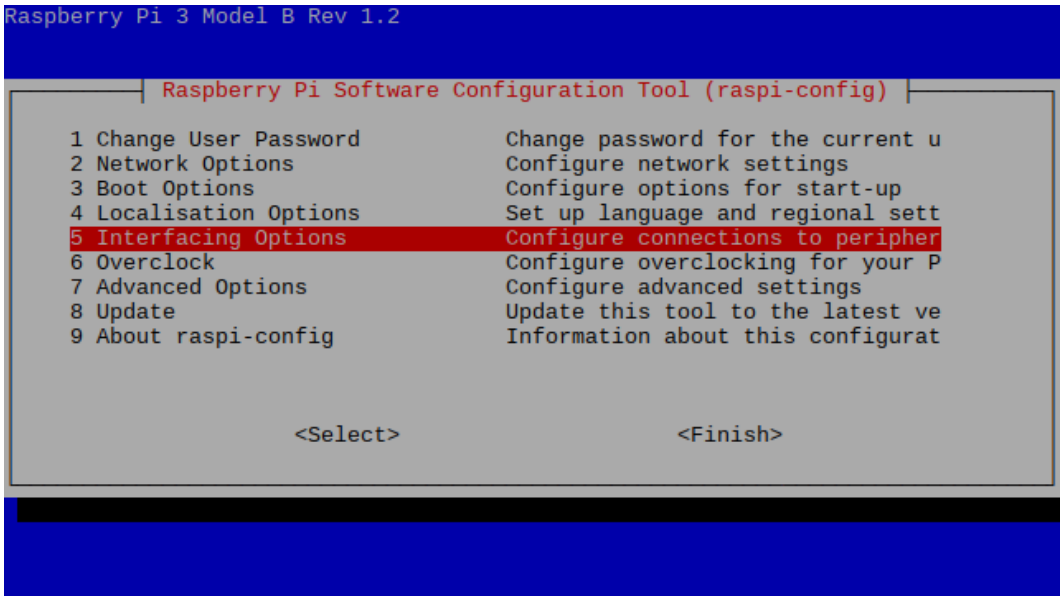
Connections to the IMU on the left with the corresponding connections to the RPI on the right

Enabling I2C on RPI:

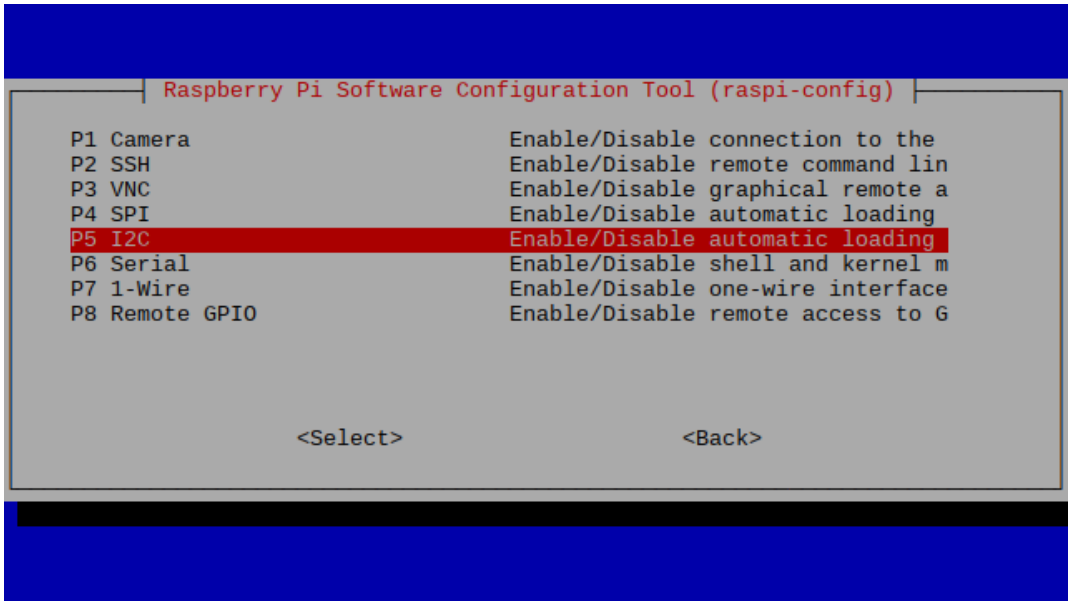
To use I2C on physical pins 3 and 5 on the RPI, we must enable I2C. In the terminal, run

```
sudo raspi-config
```

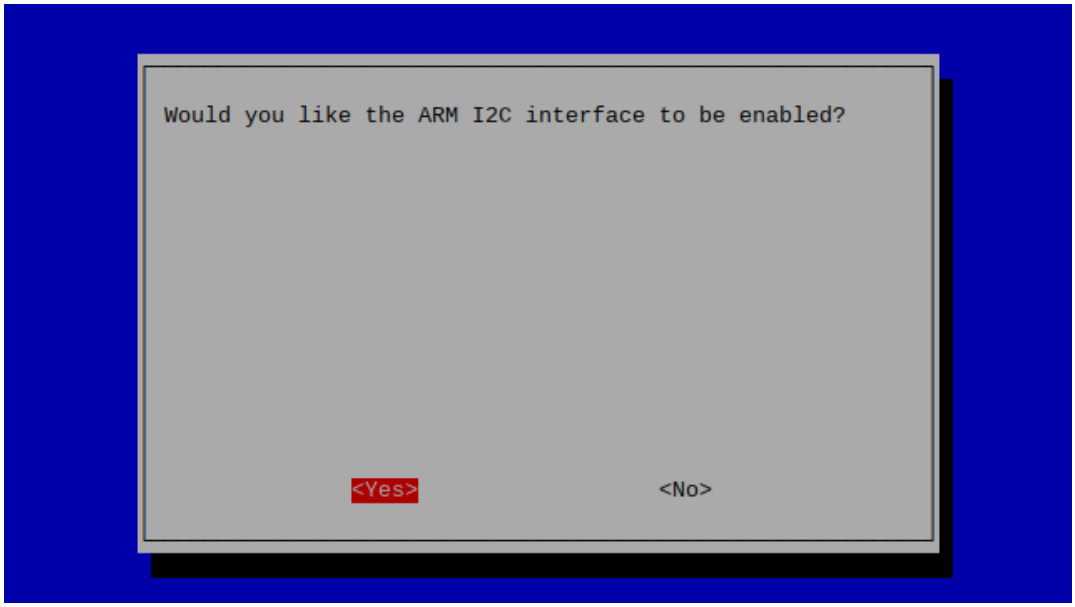
This launches the raspi-config utility. From there, select Interfacing Options:



Then select I2C:



Finally, select <Yes> to enable I2C on the RPI:



Software:

1. **Required Files:** These files must be in the same folder as your python script that will use the data from the IMU. They are in the gitlab repo.
    a. constants.py
    b. i2c.py
    c. lis3mdl.py
    d. lsm6ds33.py
    e. lps25h.py

2. **Imports:** To use the data from the IMU in your python script you must have these imports at the top of your script.

```
from constants import *         # Includes addresses on I2C bus
from lsm6ds33 import LSM6DS33   # Accel & Gyro (+ temp)
from lis3mdl import LIS3MDL     # Magnetometer (+ temp)
from lps25h import LPS25H       # Barometric Pressure & Temperature
```

3. **Initialization:** We need to initialize the IMU and enable the sensors we want to read from.

```
imu = LSM6DS33()        # Accelerometer and Gyroscope
imu.enable()

magnet = LIS3MDL()      # Magnetometer
magnet.enable()

baro = LPS25H()         # Barometric and Temperature
baro.enable()
```

4. **Functions:** These are the functions that we used to read data from the IMU:
    a. imu.getGyroscopeDPS()
    b. imu.getAccelerometerMPS2()
    c. magnet.getMagnetometerRaw()
    d. baro.getBarometerMillibars()
    e. baro.getAltitude()
    f. baro.getTemperatureCelsius()

    There are more functions for reading data in the files that are listed in section 1.