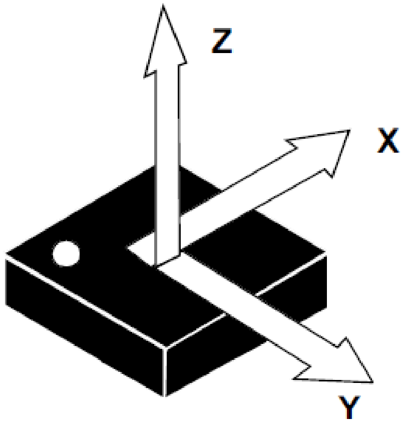
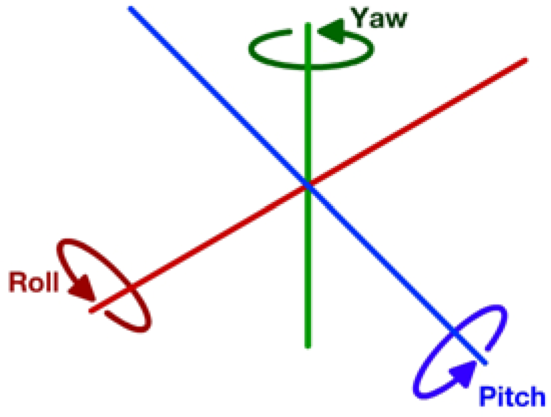
Week 1

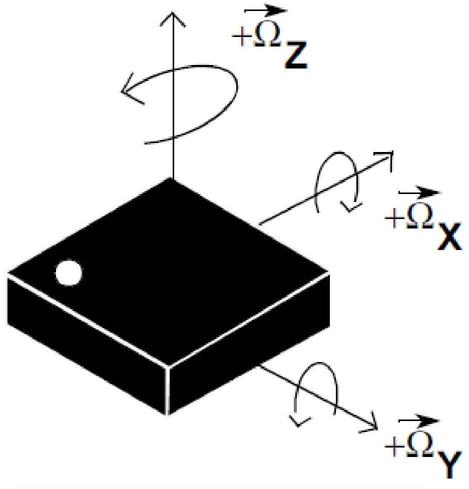
Inertial Measurement Unit (IMU)

IMU Intro

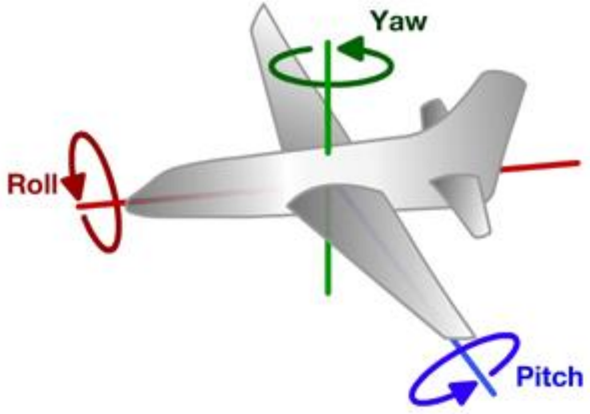


Accelerometer





Gyro



Today: IMU

- Goal: measure quad pose
- Wire IMU to PI
- Getting gyro / accel data
- Convert raw data to useful format
- Getting roll and pitch angle from Accel.
- Gyro and Angle Calibration – Milestone1

Electronics

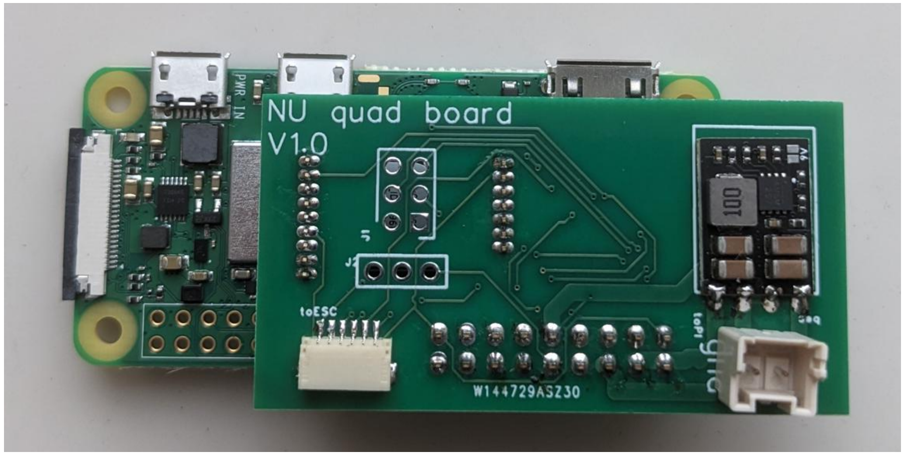
- Main electronics consist of:
  - Pi zero 2W
  - BMI088 IMU

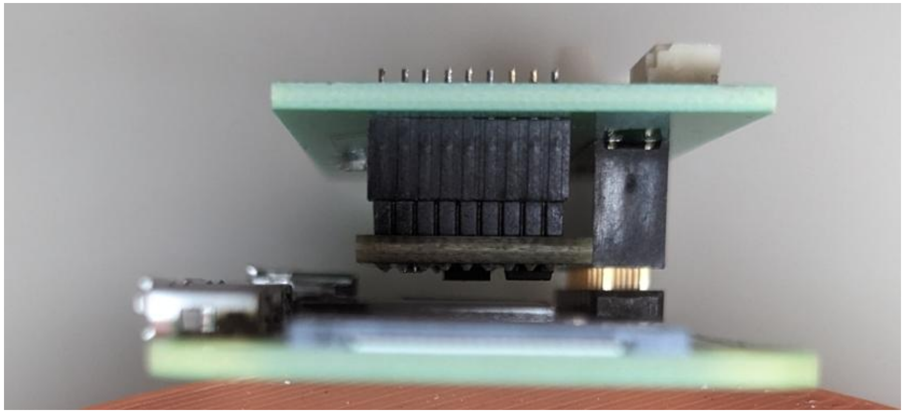
- Power only in the port marked “PWR”

- <u>Don’t let IMU touch PI</u>
  - You will make me sad if you destroy IMU :(
  - You will make me sad if you destroy the PI :(

- <u>Don’t let IMU or PI touch any metal surface</u>
  - You will make me sad if you destroy IMU :(
  - You will make me sad if you destroy the PI :(

- Don’t unplug/plug boards without checking with me, you can damage them.
  - You will make me sad if you destroy IMU :(
  - You will make me sad if you destroy the PI :(





Setup

1. Place sd card in PI
2. When ready to power (after checking with me) attach usb cable to “power” port and to computer.
3. After about 15 seconds of power, you should be able to see a ssid of “quadxy” where xy corresponding to the number with the sd card
4. Connect with password “quadxyxy”
5. Dhcp should be set as “automatic”

More Setup

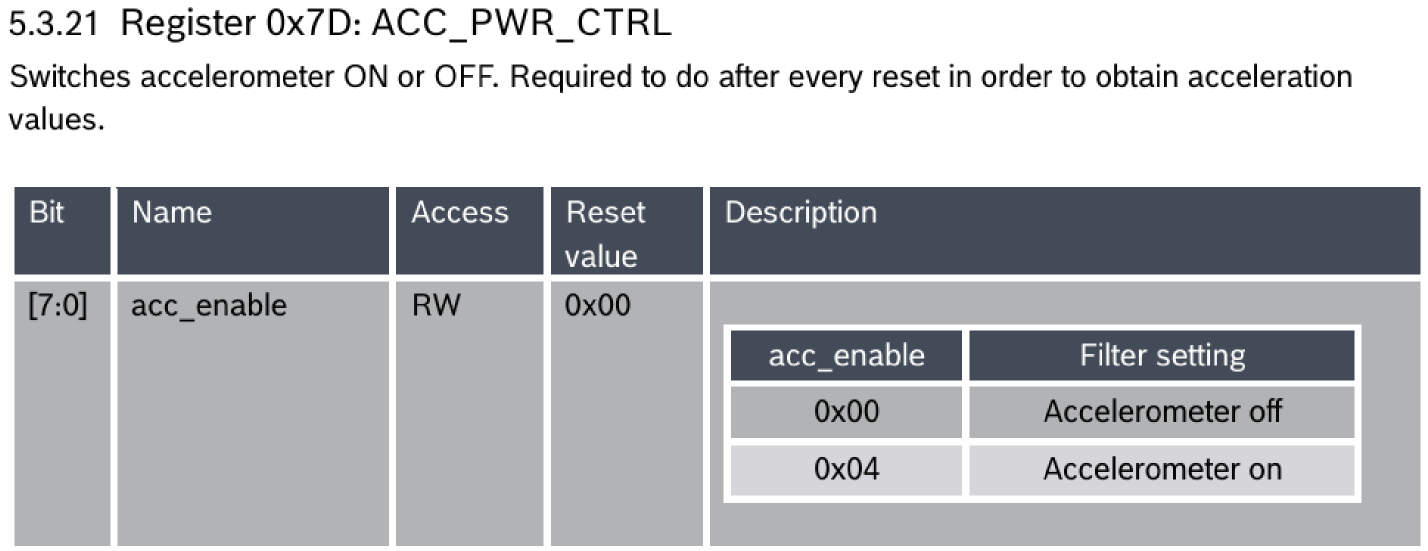
- Ssh into pi at
  - 10.42.0.1 user:pi password:raspberry
- Windows users suggested to use mobaxterm program (need to download)
  - Mac maybe Xquartz??
  - Can transfer in putty command line:

    ```
    pscp -pw raspberry week1_student.cpp pi@ 10.42.0.1 :/home/pi/flight_controller/student.cpp
    ```
- Keep all your files in the flight controller directory
- Test imu is connected by running :
  - i2cdetect –y 1
  - You should detect something connected at address 19 and 69
- Compile code
  - gcc –o name name.cpp –lwiringPi –lm
  - (type this, don’t copy/paste)
- Run code
  - ./name

DEMO

I2C interface

- Each device has an address
  - Accelerometer is at 0x19
  - Gyro is at 0x69
- Read/write memory locations
- Example from data sheet



5.3.21 Register 0x7D: ACC_PWR_CTRL

Switches accelerometer ON or OFF. Required to do after every reset in order to obtain acceleration values.

<table>
<tr><th>Bit</th><th>Name</th><th>Access</th><th>Reset value</th><th>Description</th></tr>
<tr><td>[7:0]</td><td>acc_enable</td><td>RW</td><td>0x00</td><td>
<table>
<tr><th>acc_enable</th><th>Filter setting</th></tr>
<tr><td>0x00</td><td>Accelerometer off</td></tr>
<tr><td>0x04</td><td>Accelerometer on</td></tr>
</table>
</td></tr>
</table>

Reading from I2C

- Connect to I2C address:
  - int accel_address=wiringPiI2CSetup (0x19) ;
  - Details in week1 student code
- Raw Z accelermoter values are at locations 0x16,0x17
- Read it :
  - Uint16_t az=wiringPiI2CReadReg16(accel_address,0x16)
  - take 2’s complement
- Now values range from -32768 to +32767 … what does this mean?

Convert to “G’s”

- What is our acceleration range?
  - Hint look at address 0x41 in datasheet.
  - For example if default Range is +-2 g
  - -32768 to +32767 corresponds to -2 to +2 g

Convert to degrees / sec

- What is our sensed angular velocity range?
  - Hint look at address 0x0f in datasheet
  - Set range to 1000 degrees/second
  - So -32768 to +32767 corresponds to -1000 to +1000 dps

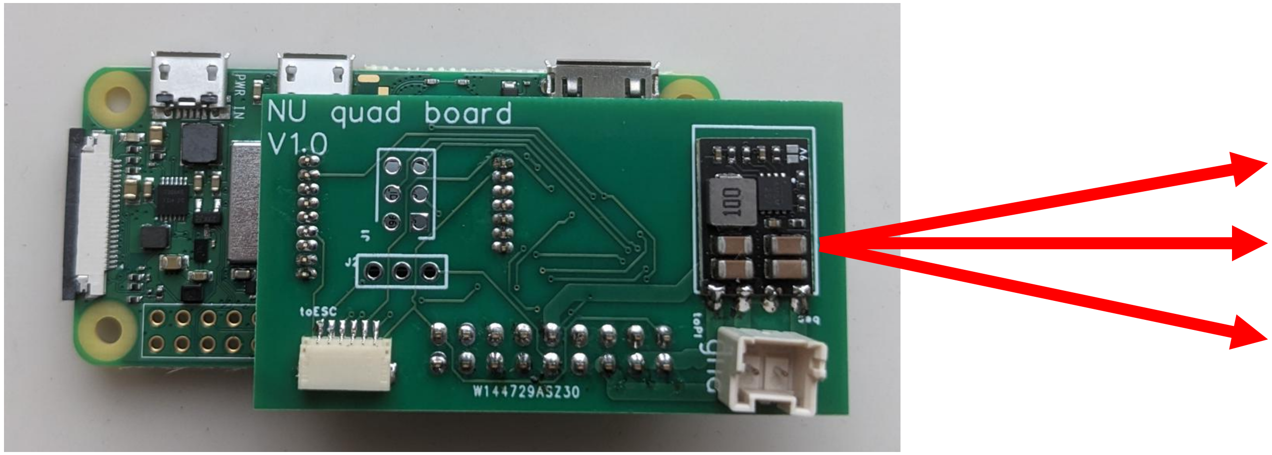
Drift

- When stationary, what is the gyro output supposed to be?

Drift

- When stationary, what is the gyro output supposed to be?
- Find average stationary value over 1000 samples in calibrate_imu() subtract from gyro output in read_imu() so stationary value is very close to zero.
- Do this for all 3 gyro DOF
- Use x_gyro_calibration, y_gyro_calibration, z_gyro_calibration global variables

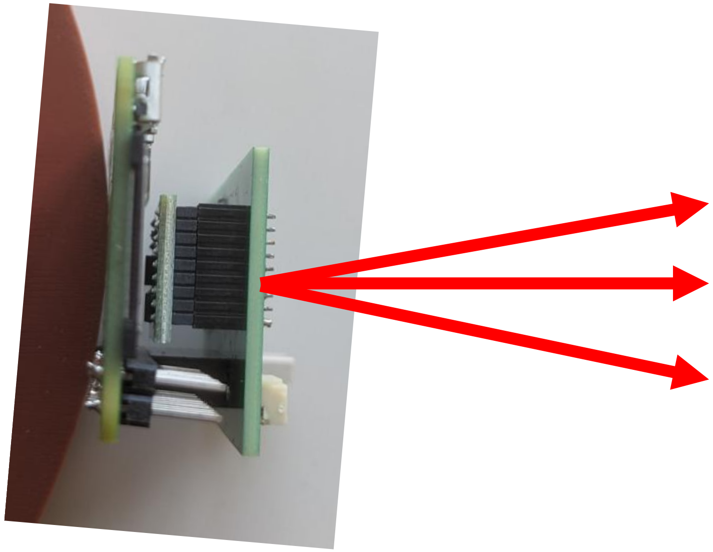
Pitch and roll



Pitch=small +

Pitch=0

Pitch=small -



Roll=small -

Roll=0

Roll=small +

Getting Pitch and Roll

- Get acceleration in X,Y,Z;
- Use Y,X values to compute pitch
  - Hint:atan2() then convert to degrees
    - Check signs, gyro and accel should match, i.e. if gyro is +, accel angle should be getting bigger
    - If using math, also include the –lm flag
- Use Z,X values for rotation on roll

Pitch and Roll Calibration

- What are pitch and roll values supposed to be when flat on table?

Pitch and Roll Calibration

- What are pitch and roll values supposed to be when flat on table?
- Find average flat value in calibrate_imu() over 1000 samples, subtract from pitch and roll values in read_imu() so stationary value computed by the accel. is very close to zero.
  - Note this is angle, not acceleration values
- Use roll_calibration, pitch_calibration global variables

Milestone 1

- Print calibrated values to screen; 3 gyro values, roll, pitch (use printf) every call of read_imu() sample. Print the values continuously.
- Make format readable
  - All in one line / reading, nice spacing
    - Hint use %10.5f to help with spacing