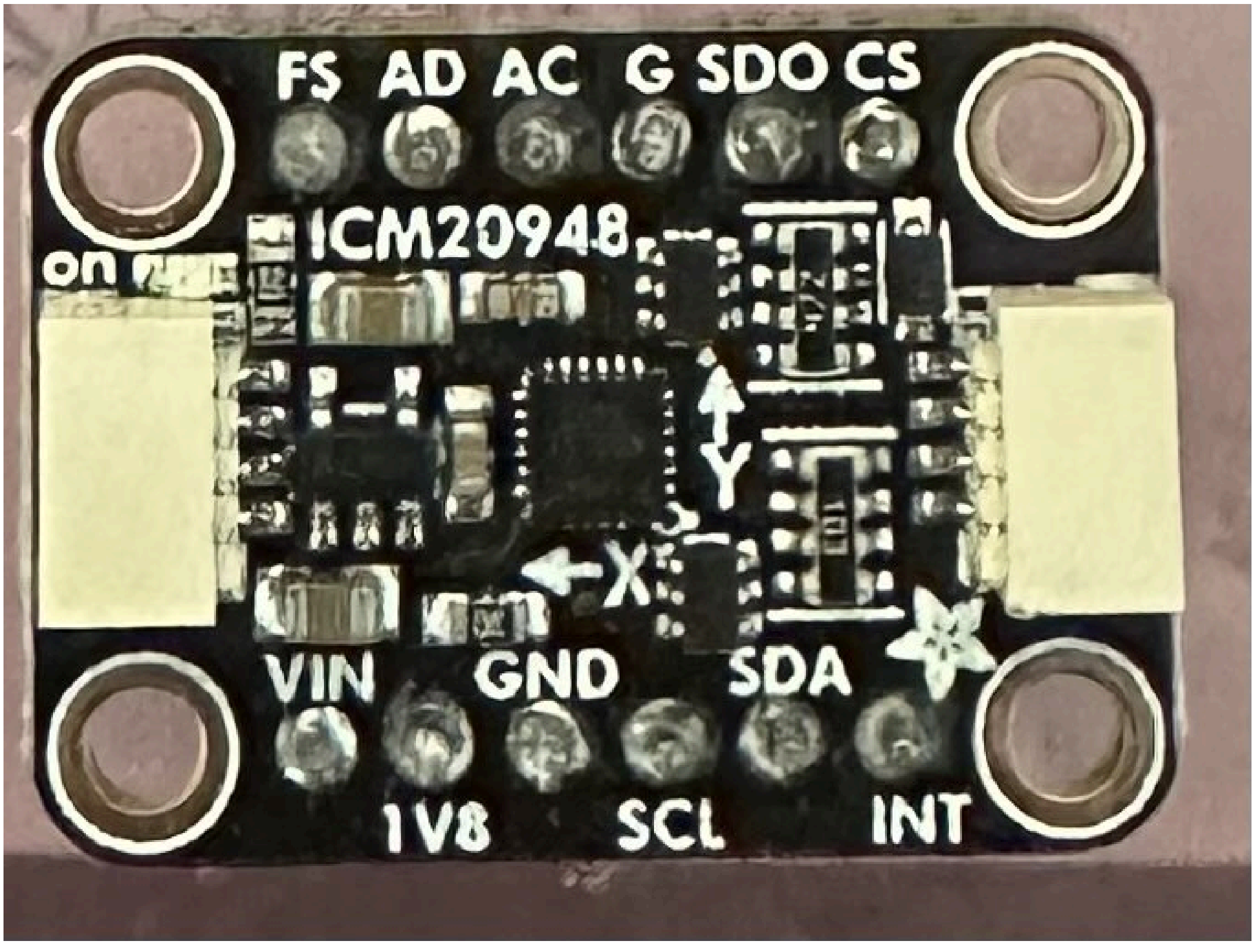
The Adafruit ICM 20948 is an accelerometer, magnetometer, and gyroscope, making it a 9 Degree of Freedom sensor. This guide will show you how to wire and receive the values being read by the sensor through the use of i2c on the BeagleBone.

Datasheet:
https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf

Wiring

- Wire GND to P9.01 (black wire)
- VIN to P9.03 (white wire)
- SCL to P9.17(green wire)
- SDA to P9.18 (blue wire)
- The wiring image can be seen below



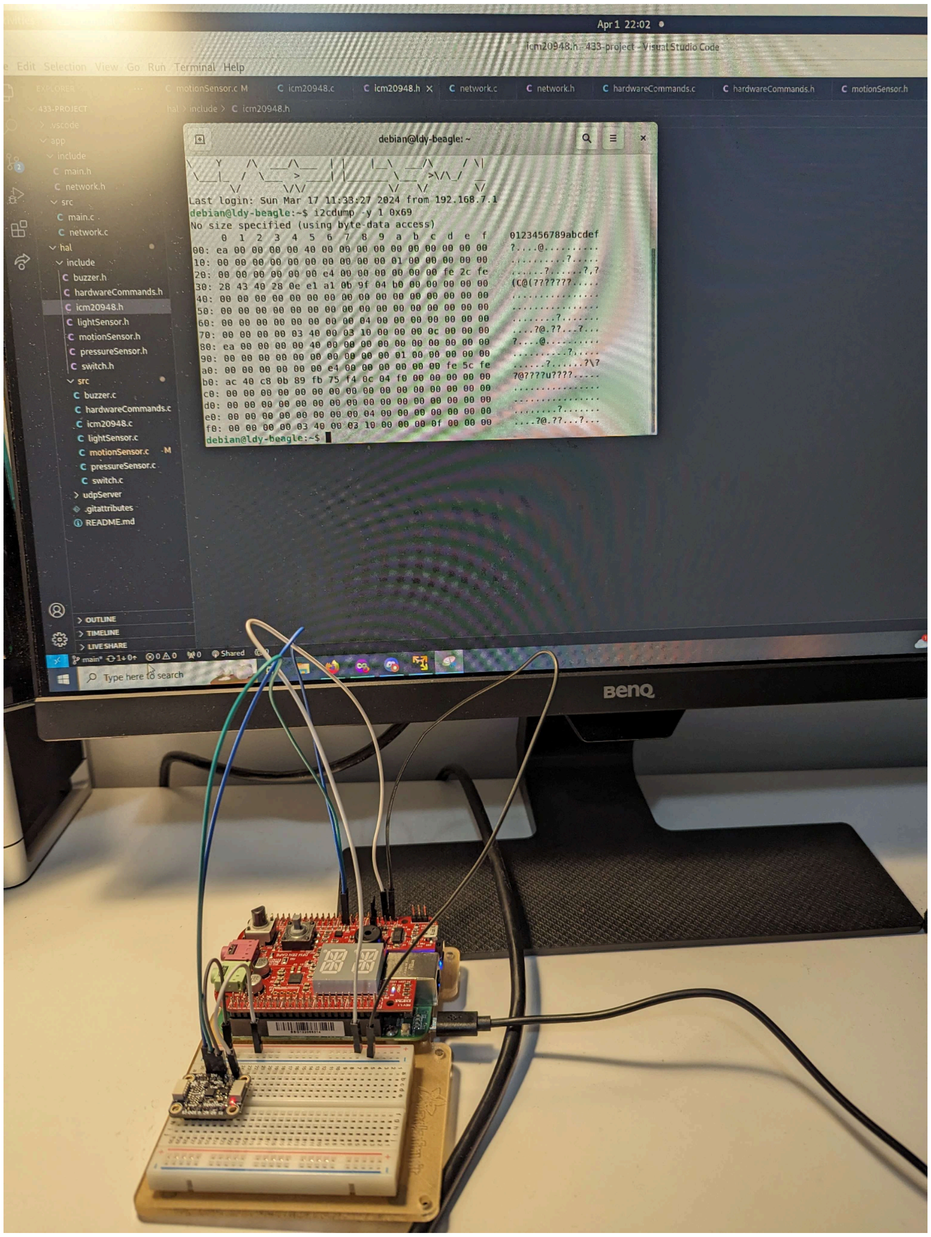
Configure P9.17 and P9.18 for I2C

- config-pin P9_18 i2c
- config-pin P9_17 i2c
- **Note:** this will reset everytime the BeagleBone is shutdown and must be configured on reboot for the user to be able to read the values.

Turn on the device by running i2cset -y 1 0x69 0x06 0x01

- 0x06 is the PWR_MGMT_1 address
- Doing this wakes the chip from sleep mode and turns off low power mode of the power management chip so that the device would become fully functional
- **Note:** this command should be used after every multibyte read or the values returned will become 0 (unknown reason)

The accelerometer and gyroscope values should now be available by using the command: i2cdump -y 1 0x69

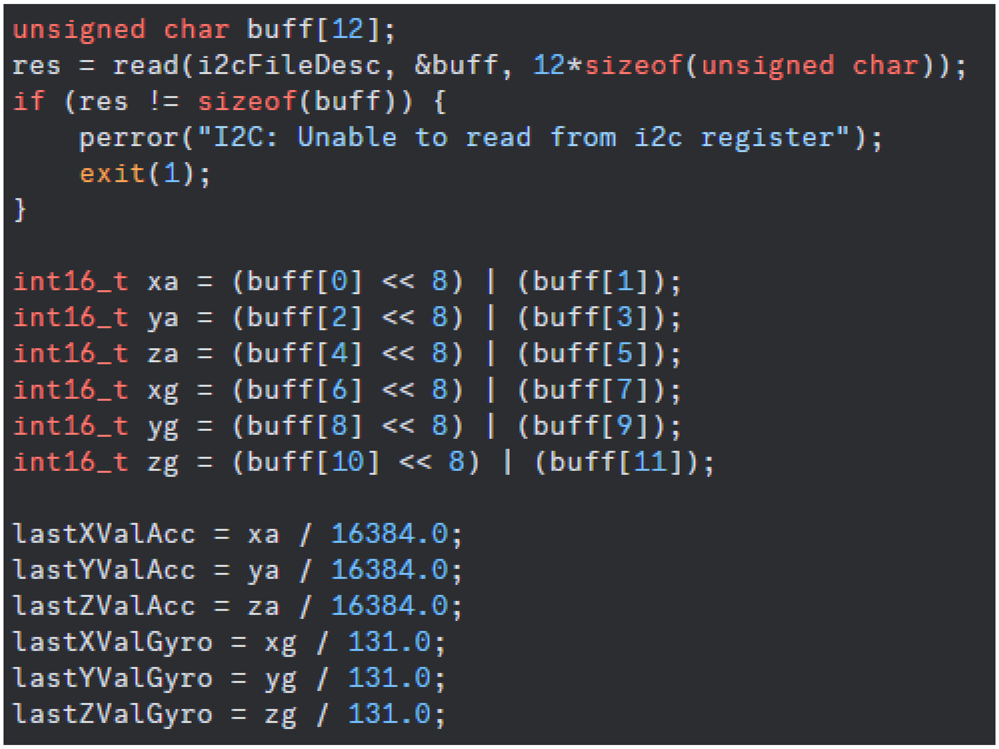


```
Last login: Sun Mar 17 11:33:27 2024 from 192.168.7.1
debian@ldy-beagle:~$ i2cdump -y 1 0x69
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: ea 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00    ?....@..........
10: 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00    ..........?.....
20: 00 00 00 00 00 00 e4 00 00 00 00 00 00 fe 2c fe    ......?......?,?
30: 28 43 40 28 0e e1 a1 0b 9f 04 b0 00 00 00 00 00    (C@(???????.....
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
60: 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00    ........?.......
70: 00 00 00 00 03 40 00 03 10 00 00 00 0c 00 00 00    ....?@.??...?...
80: ea 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00    ?....@..........
90: 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00    ..........?.....
a0: 00 00 00 00 00 00 e4 00 00 00 00 00 00 fe 5c fe    ......?......?\?
b0: ac 40 c8 0b 89 fb 75 f4 0c 04 f0 00 00 00 00 00    ?@????u????.....
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
e0: 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00    ........?.......
f0: 00 00 00 00 03 40 00 03 10 00 00 00 0f 00 00 00    ....?@.??...?...
debian@ldy-beagle:~$
```

The readings are in Big Endian from 0x2d through 0x38 which are in order of accelerometer x to z values and gyroscope x to z values. Since these registers are one after another, we can do a multibyte read by adding 0x80 to 0x2d.

Obtaining the values

- After reading the I2c Register with the buff of 12 you can obtain the accelerometer x value by bit shifting buff[0] by 8 and then applying the or operation with the mask of buff[1]. Similarly the y accelerometer value is obtained via buff[4] being shifted to the left by 8 and then applying the mask buff[2] with the or operator. And finally the z accelerometer value lies on the buff[4] being shifted by 8 and having the mask of buff[5] with the or operator.
- For the gyroscope the x value lies on buff[6] shifted by 8 with the or operator using mask buff[7], y gyroscope is on buff[8] with or operator using mask buff[9] and finally the gyroscope z lies on buff10[] being shifted by 8 and combined with the mask buff[11] with the “or” operator.
- Since the default sensitivity scale factor is 0, the values read is in terms of LSB/g (least significant bits/grams) which is why you have to then divide by 16384 to get the acceleration in G’s.
- Similarly the gyroscope scale factor is defaulted to 0, the values read is in terms of LSB/dps (least significant bits/degrees per second). To convert the data to degrees per second we have to divide the returned value by 131.
  - To know more about the sensitivity scale factor for both gyroscope and accelerometer respectively, refer to pages 11 to 12 on the datasheet provided above.



```
unsigned char buff[12];
res = read(i2cFileDesc, &buff, 12*sizeof(unsigned char));
if (res != sizeof(buff)) {
    perror("I2C: Unable to read from i2c register");
    exit(1);
}

int16_t xa = (buff[0] << 8) | (buff[1]);
int16_t ya = (buff[2] << 8) | (buff[3]);
int16_t za = (buff[4] << 8) | (buff[5]);
int16_t xg = (buff[6] << 8) | (buff[7]);
int16_t yg = (buff[8] << 8) | (buff[9]);
int16_t zg = (buff[10] << 8) | (buff[11]);

lastXValAcc = xa / 16384.0;
lastYValAcc = ya / 16384.0;
lastZValAcc = za / 16384.0;
lastXValGyro = xg / 131.0;
lastYValGyro = yg / 131.0;
lastZValGyro = zg / 131.0;
```