



MAKE EVERYTHING SMART

# SMART TECHNOLOGT

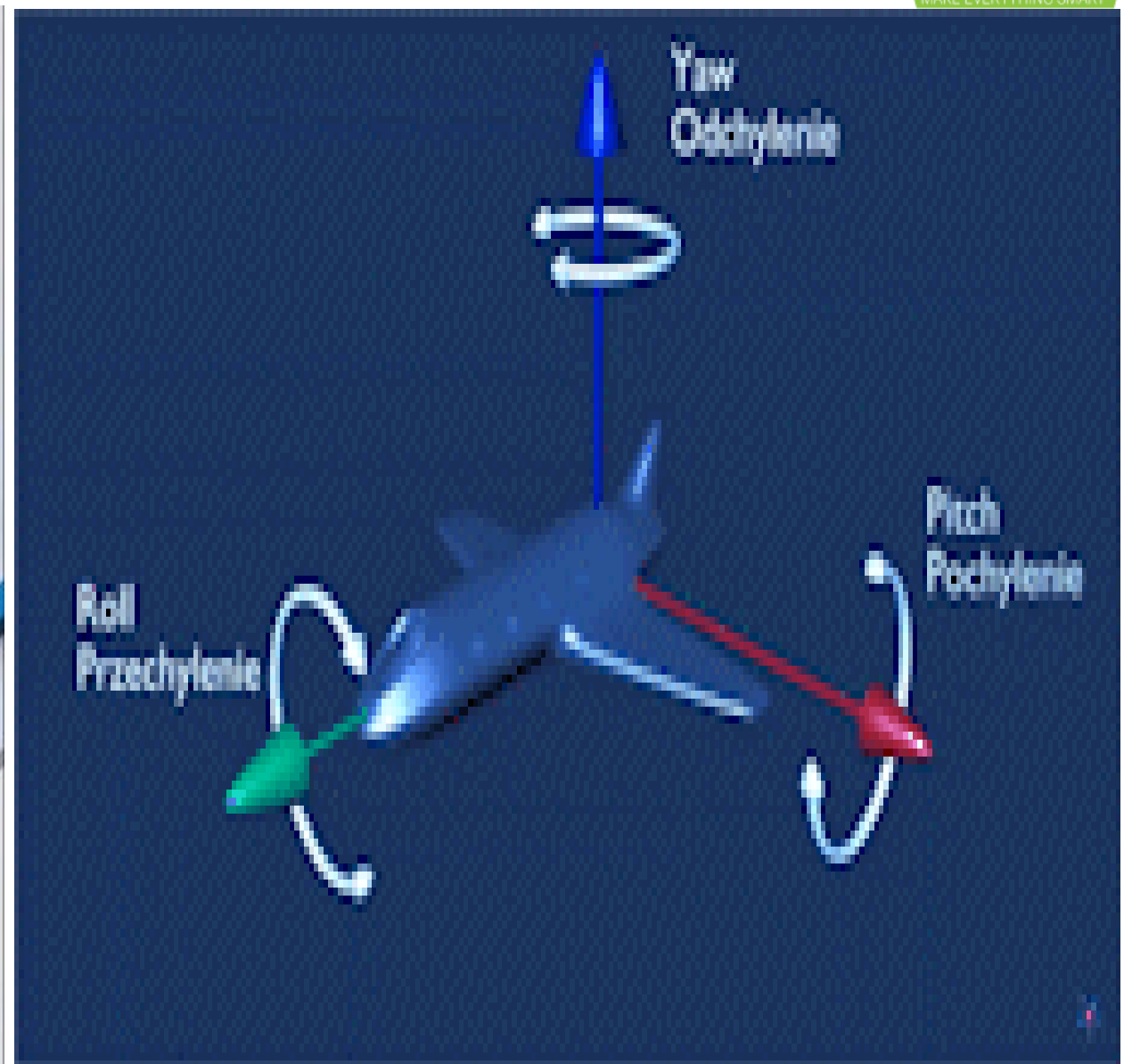
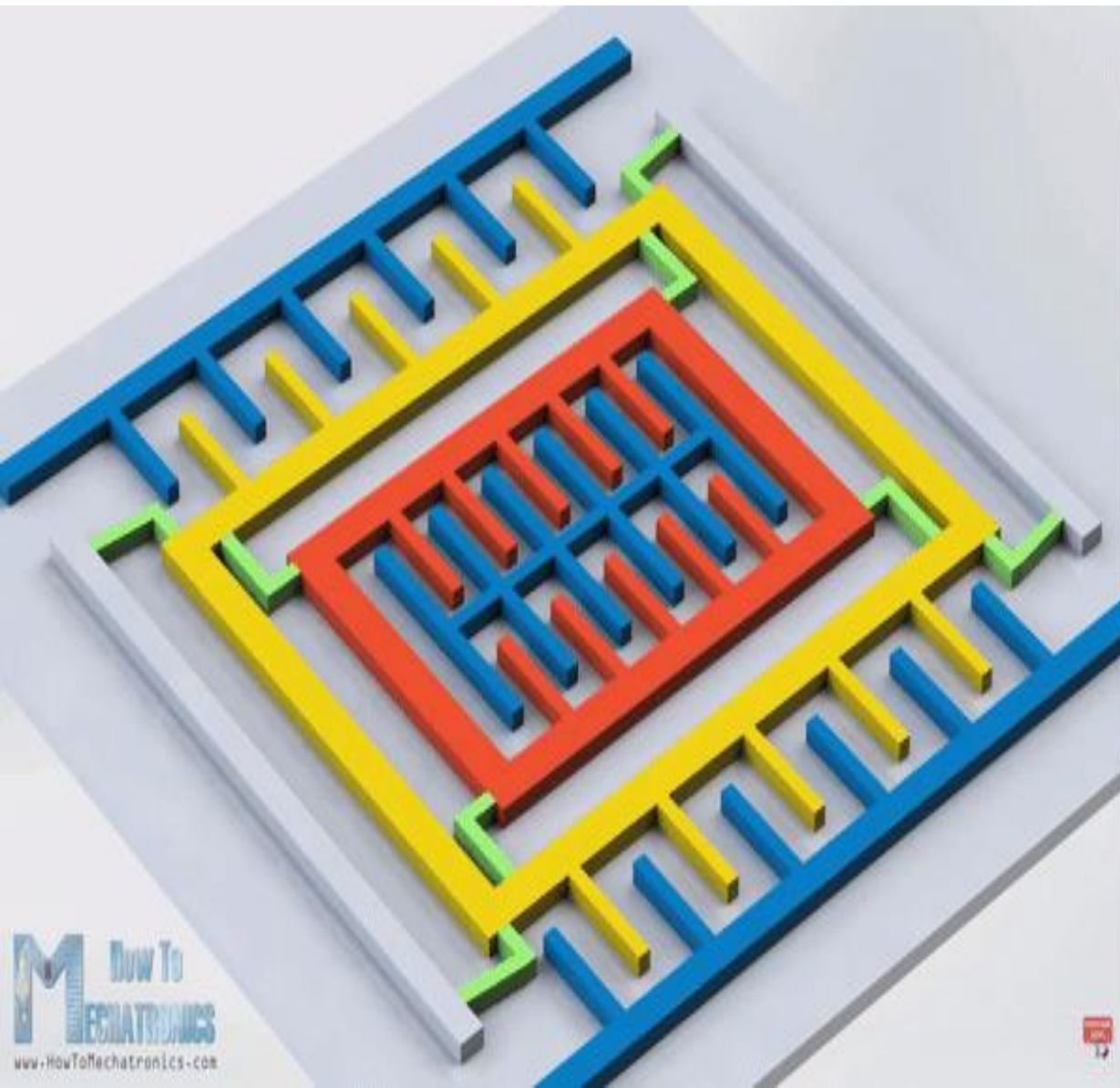


The slide features a white background with large green geometric shapes. On the left, a large green arrow points towards the center. On the right, a large green arrow points away from the center. In the bottom right corner, there is a cluster of several smaller green triangles of varying shades and orientations. A thin green horizontal line runs across the middle of the slide, passing behind the text.

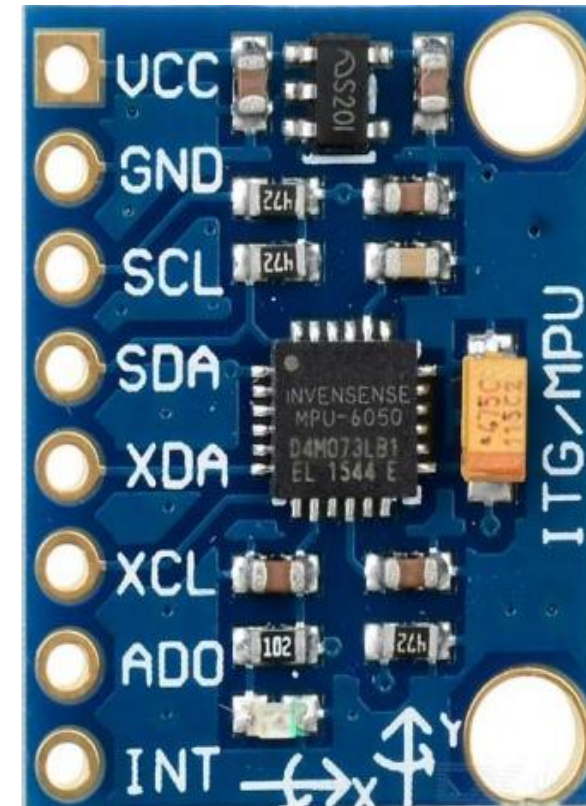
# LECTURE

9

- MPU 6050 Module



- 1) **VCC** → Connect it to the 3.3V or 5V output of the Arduino.
- 2) **GND**
- 3) **SCL** → is a I2C Clock pin. This is a timing signal supplied by the Bus Master device. Connect to the SCL pin on the Arduino.
- 4) **SDA** → is a I2C Data pin. This line is used for both transmit and receive. Connect to the SDA pin on the Arduino.
- 5) **XDA** → is the external I2C data line. The external I2C bus is for connecting external sensors.
- 6) **XCL** → is the external I2C clock line.
- 7) **AD0** → allows you to change the internal I2C address It can be used if the module is conflicting with another I2C device, or if you wish to use two MPU6050s on the same I2C bus. When you leave the AD0 pin unconnected, the default I2C address is 0x68<sub>HEX</sub> and when you connect it to 3.3V, the I2C address becomes 0x69<sub>HEX</sub>.



- The I2C Interface

The module uses the I2C interface for communication with the Arduino. It supports two separate I2C addresses: 0x68HEX and 0x69HEX. This allows two MPU6050s to be used on the same bus or to avoid address conflicts with another device on the bus.

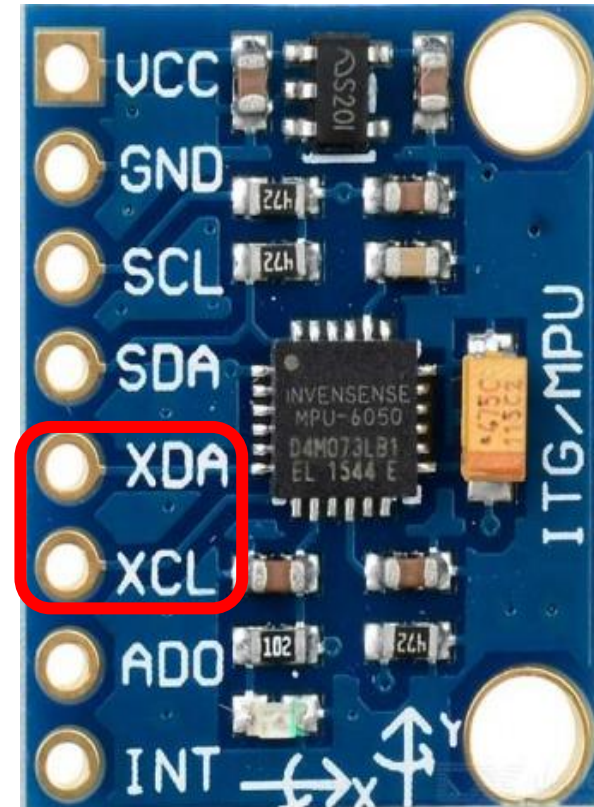
The ADO pin determines the I2C address of the module. This pin has a built-in 4.7K pull-down resistor. Therefore, when you leave the ADO pin unconnected, the default I2C address is 0x68HEX and when you connect it to 3.3V, the line is pulled HIGH and the I2C address becomes 0x69HEX.



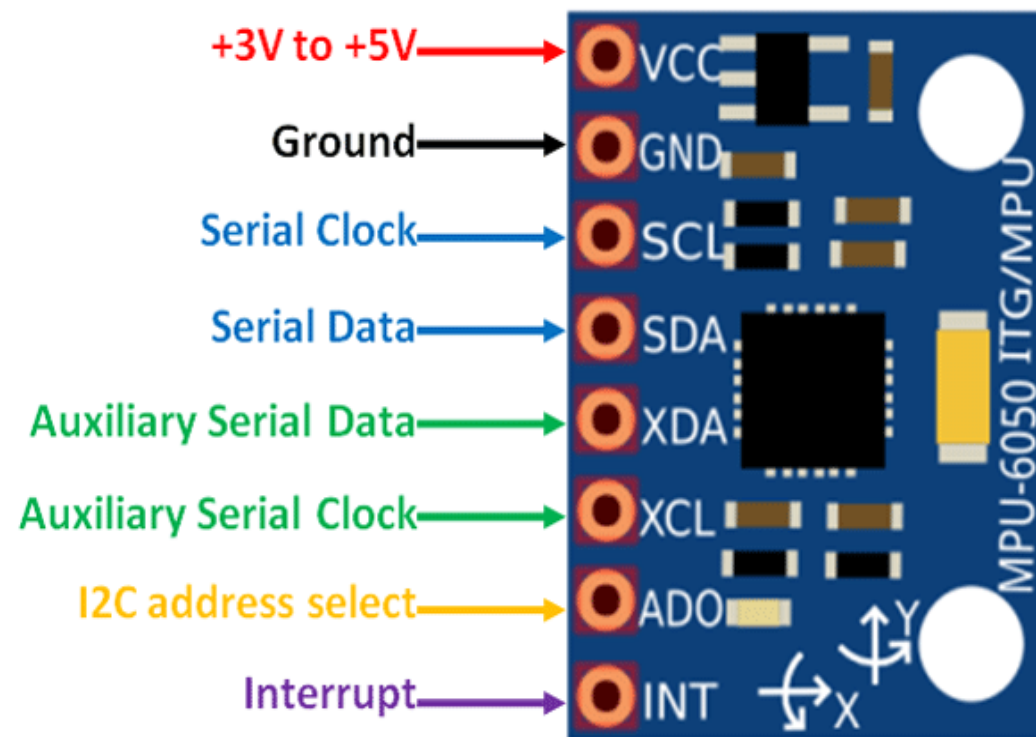
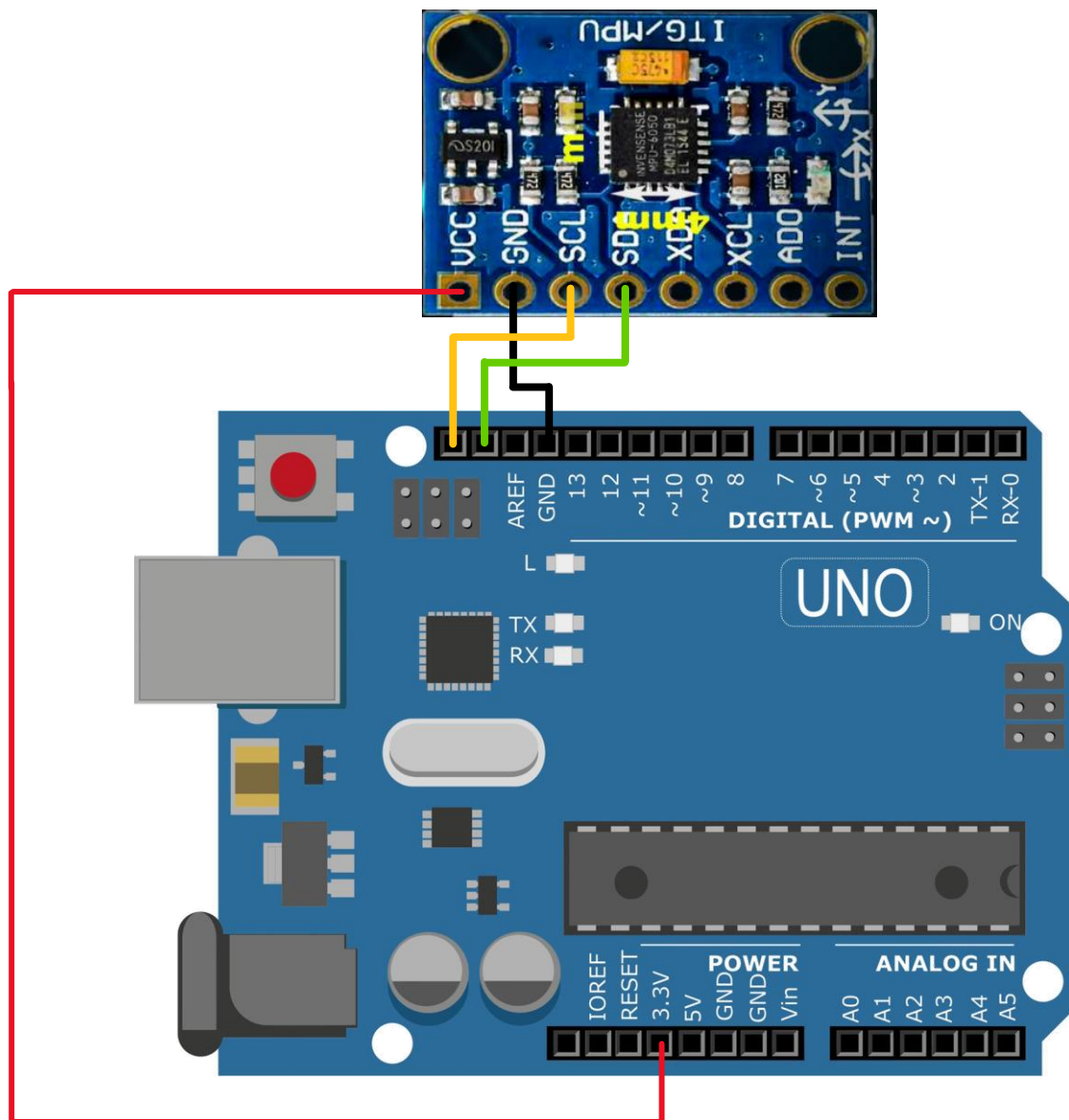


- **Adding External Sensors**

To increase the level of accuracy even further, the MPU6050 module provides a feature for connecting external sensors. These external sensors are connected to the MPU6050 via a second I2C bus (XDA and XCL), which is completely independent of the main I2C bus. This external connection is usually used to attach a magnetometer, which can measure magnetic fields on three axes. By itself, the MPU6050 has 6 Degrees of Freedom (DOF), three each for the accelerometer and the gyroscope. Adding a magnetometer adds an extra three DOF to the sensor, making it 9 DOF.



M

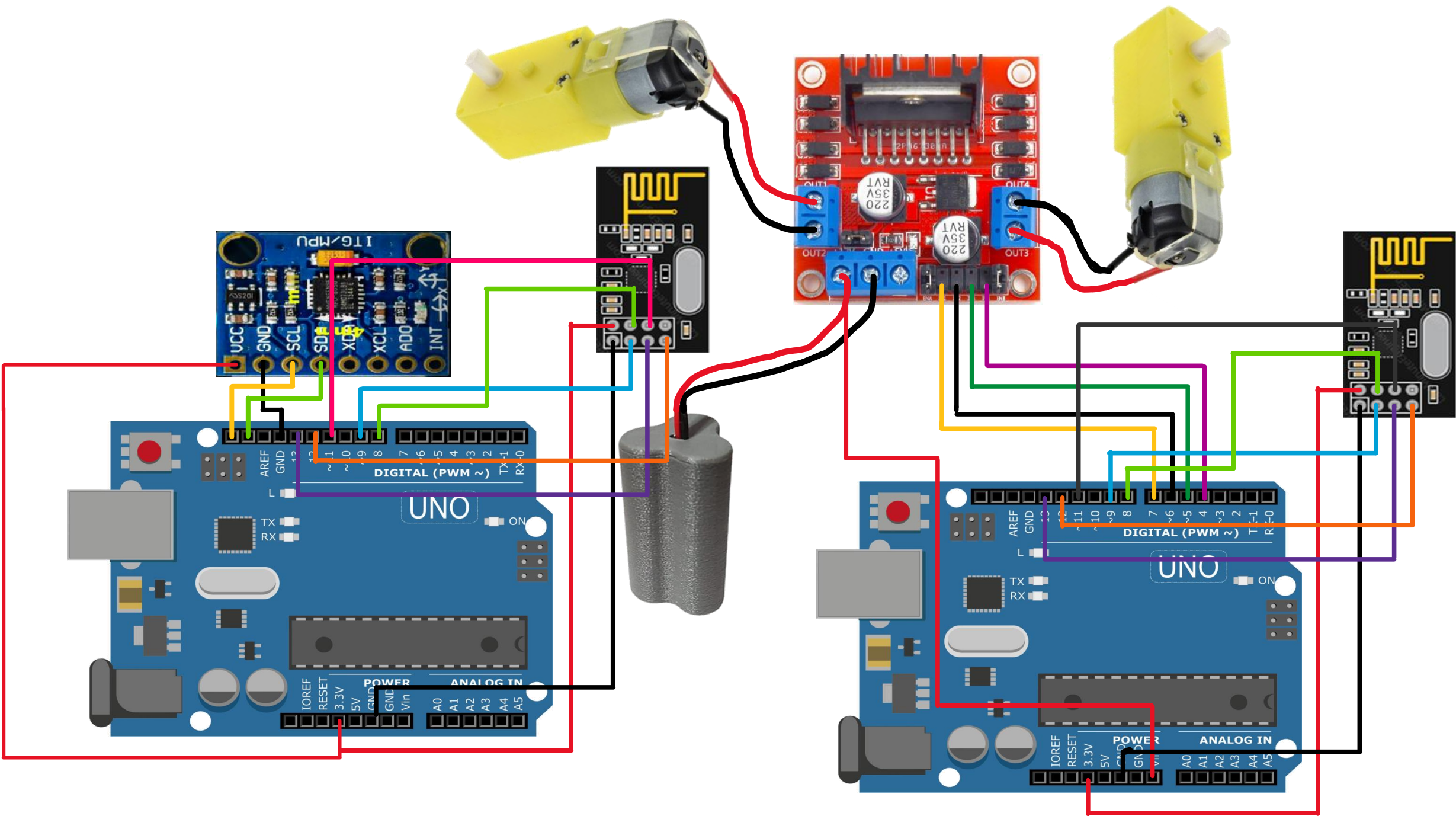


```
#include <Wire.h>
#include <MPU6050.h>
MPU6050 mpu;
void setup()
{ mpu.begin(); Serial.begin(9600);}
void loop()
{ Vector normAccel = mpu.readNormalizeAccel();

  int pitch = -(atan2(normAccel.XAxis, sqrt(normAccel.YAxis*normAccel.YAxis +
normAccel.ZAxis*normAccel.ZAxis))*180.0)/M_PI;
  int roll = (atan2(normAccel.YAxis, normAccel.ZAxis)*180.0)/M_PI;

  Serial.print("Pitch = ");
    Serial.print(pitch);
    Serial.print("      Roll = ");
    Serial.println(roll);
    delay(200);
}
```





- **Master Code**



**[https://drive.google.com/drive/folders/1cYc68HKOrz912mWYTWQ5C\\_BVZdjaPg3r?usp=sharing](https://drive.google.com/drive/folders/1cYc68HKOrz912mWYTWQ5C_BVZdjaPg3r?usp=sharing)**

- **Slave code**

**<https://drive.google.com/drive/folders/1DINDeUUfEdtvvtKMVfrZZG9iqW-QoYvtl?usp=sharing>**

**THANKS**  
**FOR**  
**COMING**

