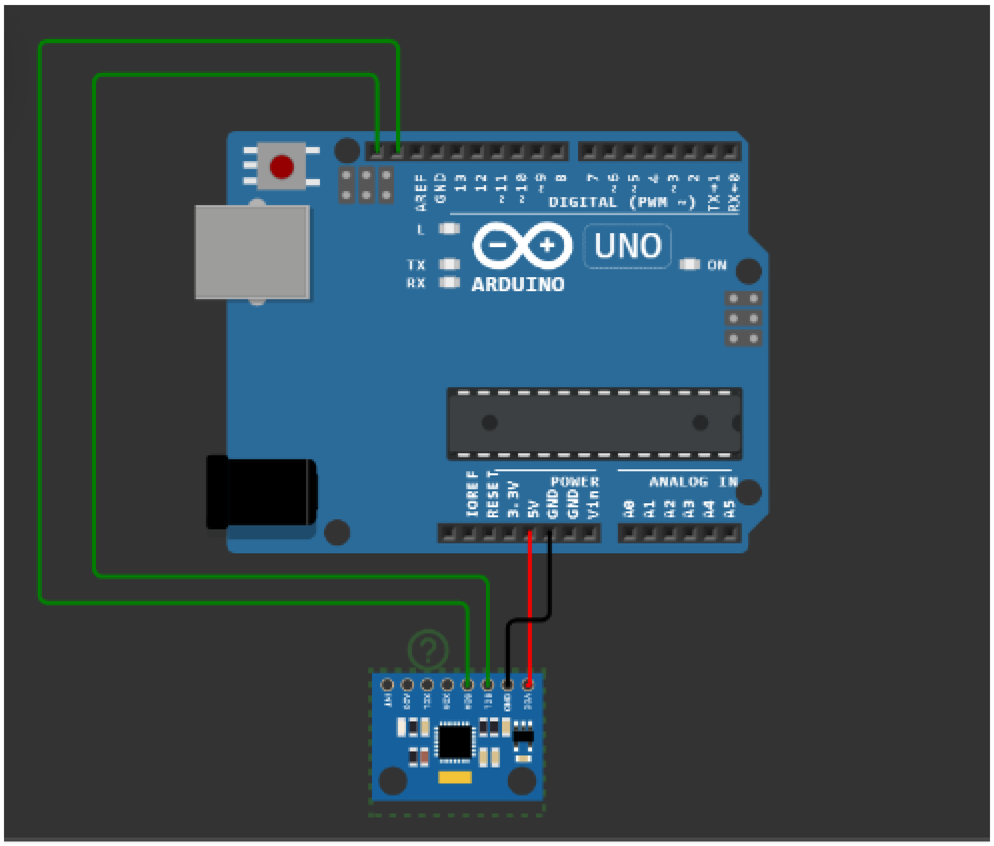
Task 6.1



```
#include <Wire.h>
#define MPU6050_ADDR  0x68
#define PWR_MGMT_1  0x6B
#define GYRO_CONFIG  0x1B
#define ACCEL_XOUT_H  0x3B
#define ACCEL_YOUT_H  0x3D
#define ACCEL_ZOUT_H  0x3F
#define GYRO_XOUT_H  0x43
#define GYRO_YOUT_H  0x45
#define GYRO_ZOUT_H  0x47

// MPU6050 variables
int acceleration_x, acceleration_y, acceleration_z;
int gyroX, gyroY, gyroZ;
float yawAngle;

void setup() {
  Serial.begin(9600);

  // Initialize I2C communication
  Wire.begin();
  // Configure gyroscope range
  Wire.beginTransmission(MPU6050_ADDR);
  Wire.write(GYRO_CONFIG);
  Wire.write(0x00);  // Set full-scale range to ±250 degrees per second
  Wire.endTransmission(true);
}

void loop() {
  // Read accelerometer data
  Wire.beginTransmission(MPU6050_ADDR);
  Wire.write(ACCEL_XOUT_H);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_ADDR, 6, true);
  acceleration_x = Wire.read() << 8 | Wire.read();
  acceleration_y = Wire.read() << 8 | Wire.read();
  acceleration_z = Wire.read() << 8 | Wire.read();

  // Read gyroscope data
  Wire.beginTransmission(MPU6050_ADDR);
  Wire.write(GYRO_XOUT_H);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_ADDR, 6, true);
  gyroX = Wire.read() << 8 | Wire.read();
  gyroY = Wire.read() << 8 | Wire.read();
  gyroZ = Wire.read() << 8 | Wire.read();

  // Convert raw gyroscope values to degrees per second
  float gyroScale = 250.0 / 32767.0;  // ±250 degrees per second
  float deg_x = gyroX * gyroScale;
  float deg_y = gyroY * gyroScale;
  float deg_z = gyroZ * gyroScale;

  // Calculate yaw angle
  yaw_Angle += (deg_z / 1000.0) * (180.0 / 3.141592);  // Integration over time

  // Print yaw angle
  Serial.print("Yaw Angle: ");
  Serial.println(yaw_Angle);
  delay(1000);
}
```

If the Sensor is surrounded by a noisy environment:

you can use a low-pass filter to reduce the effect of high-frequency noise on the yaw angle measurement. A low-pass filter allows only low-frequency signals to pass through while attenuating higher frequencies.

The gyroscope and accelerometer can be sampled between 1-8kHz, but the limitations of the SD module will restrict the sample rate down to 500 Hz - which suffices for many applications (full datasheet for MPU6050 here).

It's important to note that the MPU6050 sensor has an onboard Digital Motion Processor (DMP), which can handle the sensor fusion and provide filtered orientation outputs directly. Utilizing the DMP can simplify the process of retrieving the yaw angle while also providing more accurate results by combining data from the accelerometer, gyroscope, and magnetometer sensors.