Class: MRE 320 Sensors and Actuators

MRE 320 Individual Project

IMU SENSOR Milestone #1

Name: Yuchen Xia, Haibo Zhao

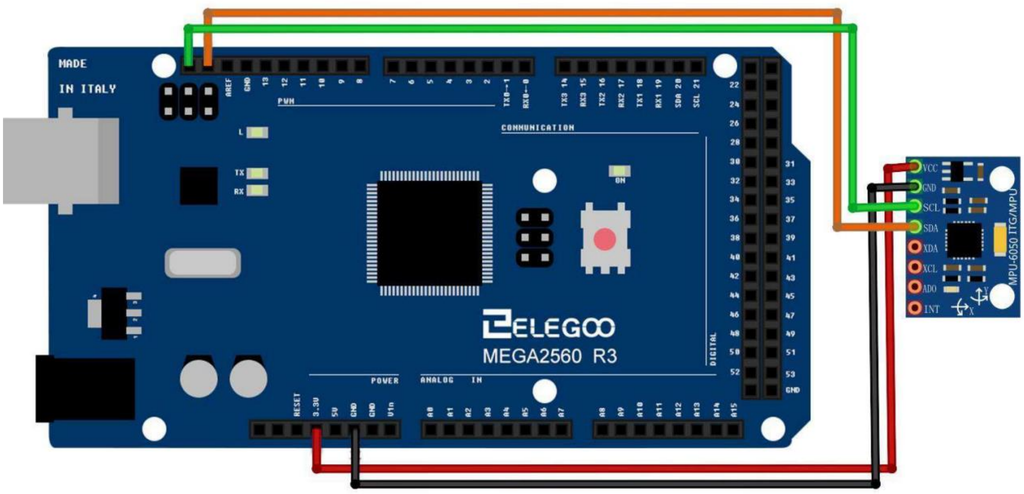
Professor: Zhang Mingshao

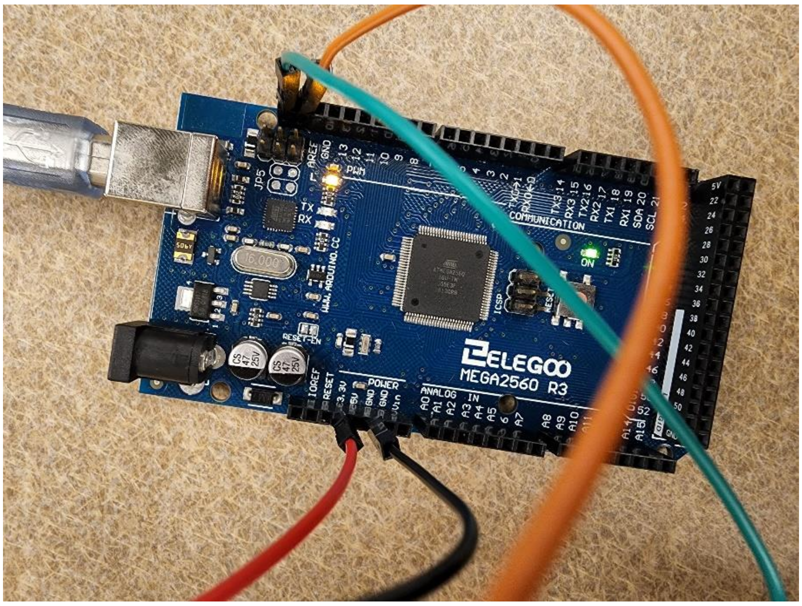
September 20, 2022

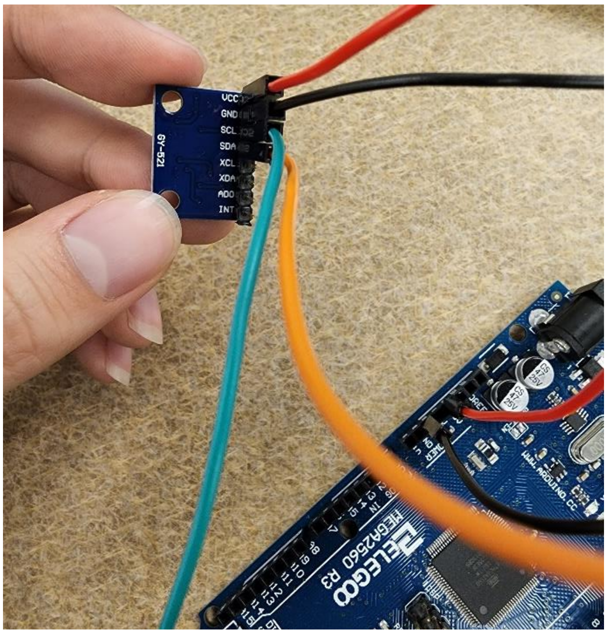
Part 1. Summary of findings in sensor study

The sensor we studied is a GY-521 module equipped with an MPU6050 chip, which is also one of the best IMU sensors. The GY-521 module features a 3-axis gyroscope, a 3-axis accelerometer, a digital motion processor (DMP), and a temperature sensor. as it contains 16-bits analog to digital conversion hardware for each channel. It can capture acceleration and angular velocity in x, y, z directions at the same time. The chip uses I2C (Inter Integrated Circuit) protocol for communication. we can see that this module has 8 IO ports, VCC is used to connect to the power supply(3～5V), GND is connected to the ground. SCL, SDA and XCL, XDA are two groups of two IIC communication buses. AD0 is the slave address setting pin of module IIC, INT is the interrupt output pin. IMU sensors are an integral part of many electronic products today such as the framework for game production and applications, location-based services, points of interest, handset and portable gaming, motion-based game controllers, Wearable sensors for health, fitness and sports etc.

Connection







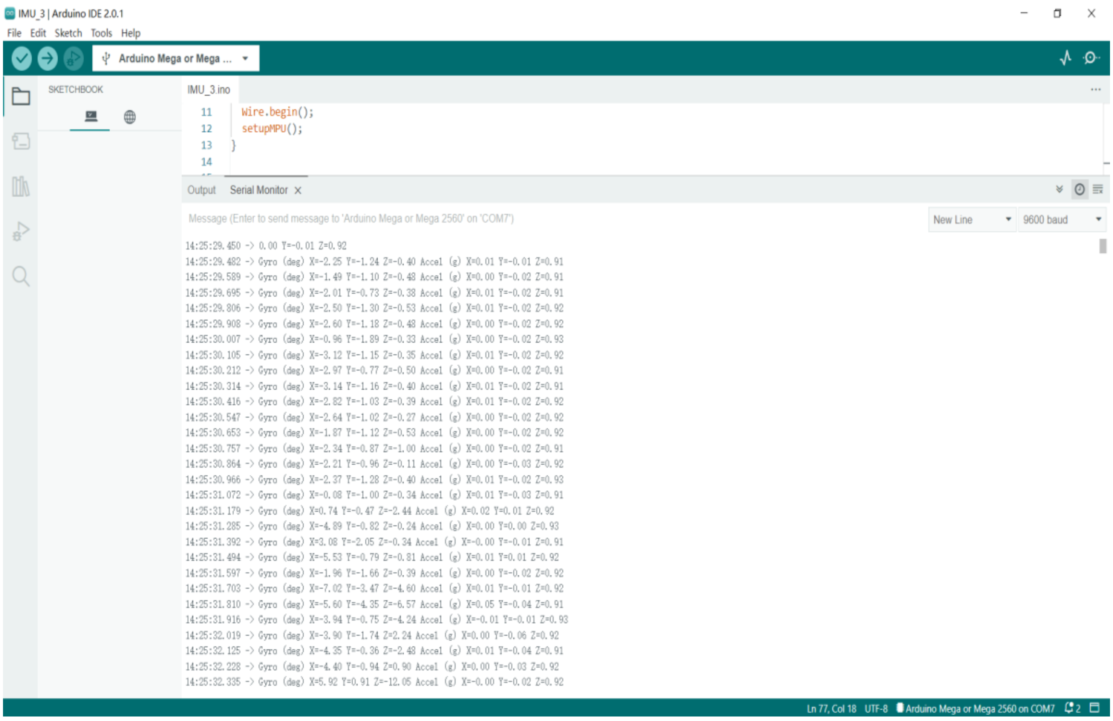
Arduino code

```
#include <Wire.h>

long accelX, accelY, accelZ;
float gForceX, gForceY, gForceZ;

long gyroX, gyroY, gyroZ;
float rotX, rotY, rotZ;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  setupMPU();
}


void loop() {
  recordAccelRegisters();
  recordGyroRegisters();
  printData();
  delay(500);
}

void setupMPU(){
```

```
  Wire.beginTransmission(0b1101000); //This is the I2C address of the MPU (b1101000/b1101001 for AC0 low/high datasheet sec. 9.2)
  Wire.write(0x6B); //Accessing the register 6B - Power Management (Sec. 4.28)
  Wire.write(0b00000000); //Setting SLEEP register to 0. (Required; see Note on p. 9)
  Wire.endTransmission();
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x1B); //Accessing the register 1B - Gyroscope Configuration (Sec. 4.4)
  Wire.write(0x00000000); //Setting the gyro to full scale +/- 250deg./s
  Wire.endTransmission();
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x1C); //Accessing the register 1C - Acccelerometer Configuration (Sec. 4.5)
  Wire.write(0b00000000); //Setting the accel to +/- 2g
  Wire.endTransmission();
}

void recordAccelRegisters() {
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x3B); //Starting register for Accel Readings
  Wire.endTransmission();
  Wire.requestFrom(0b1101000,6); //Request Accel Registers (3B - 40)
  while(Wire.available() < 6);
  accelX = Wire.read()<<8|Wire.read(); //Store first two bytes into accelX
  accelY = Wire.read()<<8|Wire.read(); //Store middle two bytes into accelY
  accelZ = Wire.read()<<8|Wire.read(); //Store last two bytes into accelZ
  processAccelData();
}

void processAccelData(){
  gForceX = accelX / 16384.0;
  gForceY = accelY / 16384.0;
  gForceZ = accelZ / 16384.0;
}

void recordGyroRegisters() {
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x43); //Starting register for Gyro Readings
  Wire.endTransmission();
  Wire.requestFrom(0b1101000,6); //Request Gyro Registers (43 - 48)
  while(Wire.available() < 6);
  gyroX = Wire.read()<<8|Wire.read(); //Store first two bytes into accelX
  gyroY = Wire.read()<<8|Wire.read(); //Store middle two bytes into accelY
  gyroZ = Wire.read()<<8|Wire.read(); //Store last two bytes into accelZ
  processGyroData();
}
```

```
void processGyroData() {
  rotX = gyroX / 131.0;
  rotY = gyroY / 131.0;
  rotZ = gyroZ / 131.0;
}

void printData() {
  Serial.print("Gyro (deg)");
  Serial.print(" X=");
  Serial.print(rotX);
  Serial.print(" Y=");
  Serial.print(rotY);
  Serial.print(" Z=");
  Serial.print(rotZ);
  Serial.print(" Accel (g)");
  Serial.print(" X=");
  Serial.print(gForceX);
  Serial.print(" Y=");
  Serial.print(gForceY);
  Serial.print(" Z=");
  Serial.println(gForceZ);
}
```

Screenshot



Part 2. Testing plans

1.Error measurement

For the gyroscope, fix the IUM sensor on the breadboard, and connect a rotary encoder under it to measure the angular velocity of the rotating breadboard, and analyze the error by comparing the reading with the IMU sensor. For an accelerometer, the acceleration can be obtained by further deriving the angular velocity with respect to time t.

2.Resolution

Resolution is the minimum input of angular velocity that gyroscope can identify.

To measure the resolution, input the minute angular velocity to the gyro, and see the change in the output.

3. Accuracy

Establish a coordinate system and calculate the error between the measured displacement value and the actual displacement value. Then find the maximum error of the measurement, the general formula for calculating the accuracy is the absolute value of the maximum allowable error divided by the measurement range times 100%, and then use the formula to calculate the accuracy

4.Sensitivity

The sensitivity of a nonlinear sensor can be expressed as the change in input voltage divided by the change in acceleration or the change in angular velocity or the change in temperature, which can be expressed by the slope of the fitted value line in a small area of the sensor image.