Blue wire- com 5 - slave

Black -com 4 - master

Caliberation code:

```
#include <Wire.h>

#include <MPU6050.h>


MPU6050 mpu;


void setup() {

 Serial.begin(115200);

 Wire.begin();

 mpu.initialize();


 if (!mpu.testConnection()) {

  Serial.println("MPU6050 connection failed");

  while (1);

 }


 Serial.println("Calibrating gyro and accel, keep the sensor still...");

 delay(1000);


 mpu.CalibrateAccel(6);

 mpu.CalibrateGyro(6);

 Serial.println("Calibration complete.");


 mpu.PrintActiveOffsets();

}


void loop() {}
```

```
slave:  caliberation: 1840.00000,-2713.00000,    6364.00000,     -24.00000,       -16.00000,       -64.00000
master :caliberation: -358.00000,          1653.00000,     7164.00000,     133.00000,       -53.00000,       -13.00000
```

master:

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

// Master sensor values
int16_t axM, ayM, azM, gxM, gyM, gzM;
float masterAngle = 0.0;

// Slave sensor values
float slaveAngle = 0.0;
int16_t axS = 0, ayS = 0, azS = 0, gxS = 0, gyS = 0, gzS = 0;

// Relative angle
float relativeAngle = 0.0;

void setup() {
  Serial.begin(115200);   // Read angle from Slave (via Serial)
  Wire.begin();          // I2C for MPU6050

  mpu.initialize();

  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed");
    while (1);
  }

  // Calibration for Master MPU6050
  mpu.setXAccelOffset(1936);
  mpu.setYAccelOffset(-2689);
  mpu.setZAccelOffset(6364);
  mpu.setXGyroOffset(-24);
  mpu.setYGyroOffset(-19);
  mpu.setZGyroOffset(-63);

  Serial.println("Master Arduino Ready");
```

```
}

void loop() {
  // --- Read Master MPU6050 data ---
  mpu.getMotion6(&axM, &ayM, &azM, &gxM, &gyM, &gzM);

  // Compute Master angle (pitch from accelerometer)
  masterAngle = atan2(axM, sqrt((float)ayM * ayM + (float)azM * azM)) * 180.0 / PI;

  // --- Read Slave Data from Serial ---
  if (Serial.available()) {
    String data = Serial.readStringUntil('\n');
    data.trim();

    if (data.startsWith("slave:")) {
      data.remove(0, 6); // Remove "slave:"
      int index = 0;
      float values[7];  // angle, ax, ay, az, gx, gy, gz

      while (data.length() > 0 && index < 7) {
        int commaIndex = data.indexOf(',');
        String value = (commaIndex == -1) ? data : data.substring(0, commaIndex);
        values[index++] = value.toFloat();
        data = (commaIndex == -1) ? "" : data.substring(commaIndex + 1);
      }

      slaveAngle = values[0];
      axS = values[1]; ayS = values[2]; azS = values[3];
      gxS = values[4]; gyS = values[5]; gzS = values[6];
    }
  }

  // --- Compute Relative Angle ---
  relativeAngle = masterAngle - slaveAngle;

  // --- Print All Data ---
  Serial.println("====== RELATIVE ELBOW ANGLE SYSTEM ======");
  Serial.print("Master Angle: "); Serial.print(masterAngle, 2);
  Serial.print("° | Slave Angle: "); Serial.print(slaveAngle, 2);
  Serial.print("° | Relative: "); Serial.print(relativeAngle, 2); Serial.println("°");

  Serial.println("---- Master Sensor ----");
  Serial.print("Accel (ax,ay,az): "); Serial.print(axM); Serial.print(", ");
  Serial.print(ayM); Serial.print(", "); Serial.println(azM);
  Serial.print("Gyro (gx,gy,gz): "); Serial.print(gxM); Serial.print(", ");
  Serial.print(gyM); Serial.print(", "); Serial.println(gzM);

  Serial.println("---- Slave Sensor ----");
  Serial.print("Accel (ax,ay,az): "); Serial.print(axS); Serial.print(", ");
  Serial.print(ayS); Serial.print(", "); Serial.println(azS);
  Serial.print("Gyro (gx,gy,gz): "); Serial.print(gxS); Serial.print(", ");
  Serial.print(gyS); Serial.print(", "); Serial.println(gzS);

  Serial.println("========================================\n");

  delay(250);
}
```

Slave:

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

void setup() {
  Serial.begin(115200);   // Send to Master
  Wire.begin();
  mpu.initialize();

  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed");
    while (1);
  }
  // Calibration values for Slave
  mpu.setXAccelOffset(-686);
  mpu.setYAccelOffset(1299);
  mpu.setZAccelOffset(7042);
  mpu.setXGyroOffset(135);
  mpu.setYGyroOffset(-60);
  mpu.setZGyroOffset(-16);
}

void loop() {
  int16_t ax, ay, az;
  int16_t gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  // Calculate pitch angle
  float angle = atan2(ax, sqrt((float)ay * ay + (float)az * az)) * 180.0 / PI;

  // Send data as a single line
  Serial.print("slave:");
  Serial.print(angle, 2); Serial.print(",");
  Serial.print(ax); Serial.print(",");
  Serial.print(ay); Serial.print(",");
  Serial.print(az); Serial.print(",");
  Serial.print(gx); Serial.print(",");
  Serial.print(gy); Serial.print(",");
  Serial.println(gz);

  delay(200);
}
```