Daniel Kloppe,

Reno Hoffman,

Haizi Cao,

Haoxuan Li

Sensor Measurement Proposal

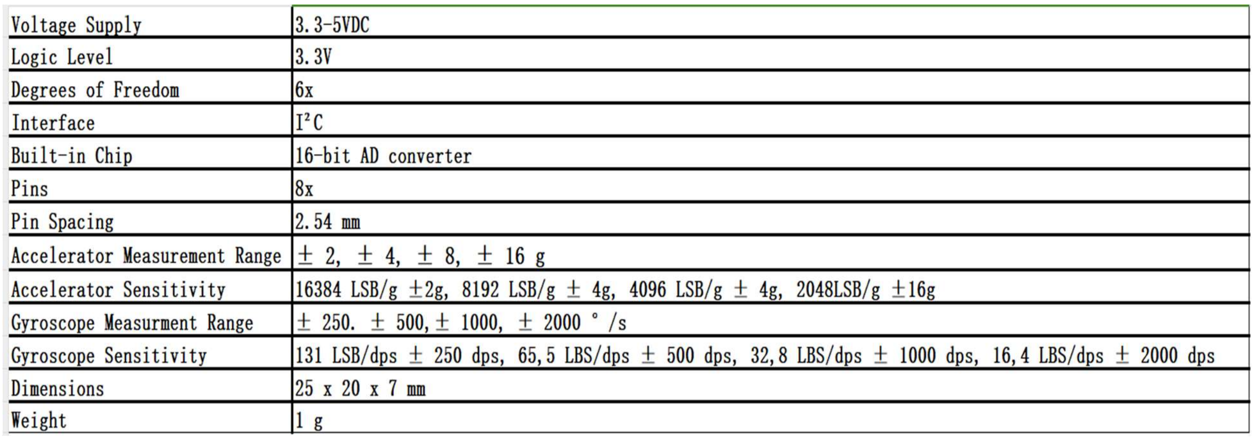
09/24/23

The format will cover all things about one sensor then will move on to the next.

MPU-6050

The MPU-6050 tracks motion. It uses a 3-axis accelerometer which uses piezo-electric effects to track the acceleration acting on it in unit vectors. This includes acceleration from motion and gravity. It also uses a 3-axis Gyroscope to measure the change in angle over time around the unit axes.

To measure the static characteristics of the MPU-6050, acceleration and orientation must be considered. For static state acceleration, the drift and static error can be measured. The drift can be measured by placing the sensor in a stationary position and observing deviations in the acceleration over time. The static error can be measured by observing the error between the acceleration of the sensor while stationary and its output. This can be repeated with the sensor in varying orientations. The values will be compared to a level and the angles will be oriented with the aid of a protractor. To obtain expected values, trigonometry and physics will be. To check the static characteristics of the gyroscope in the sensor, the same characteristics can be measured. The drift will be found by leaving the sensor in a stationary position and observing the variation from zero over time. The static error can then be measured by once again leaving the sensor in a stationary position and finding the absolute error from the observed values. Other applicable characteristics will be measured in a similar manner.



Voltage Supply	3.3-5VDC
Logic Level	3.3V
Degrees of Freedom	6x
Interface	I²C
Built-in Chip	16-bit AD converter
Pins	8x
Pin Spacing	2.54 mm
Accelerator Measurement Range	± 2, ± 4, ± 8, ± 16 g
Accelerator Sensitivity	16384 LSB/g ±2g, 8192 LSB/g ± 4g, 4096 LSB/g ± 4g, 2048LSB/g ±16g
Gyroscope Measurment Range	± 250. ± 500,± 1000, ± 2000 ° /s
Gyroscope Sensitivity	131 LSB/dps ± 250 dps, 65,5 LBS/dps ± 500 dps, 32,8 LBS/dps ± 1000 dps, 16,4 LBS/dps ± 2000 dps
Dimensions	25 x 20 x 7 mm
Weight	1 g

Figure 1 - MPU6050 Specifications

MPU-6050 _ Code/Hardware

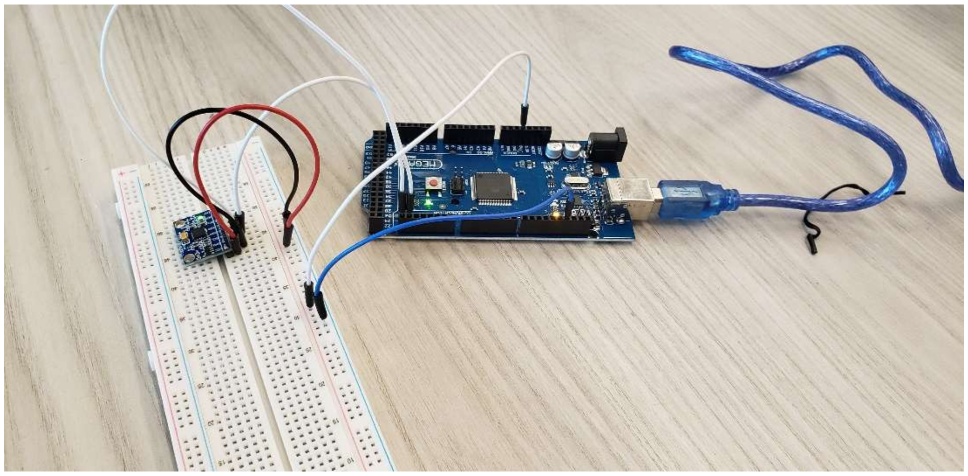
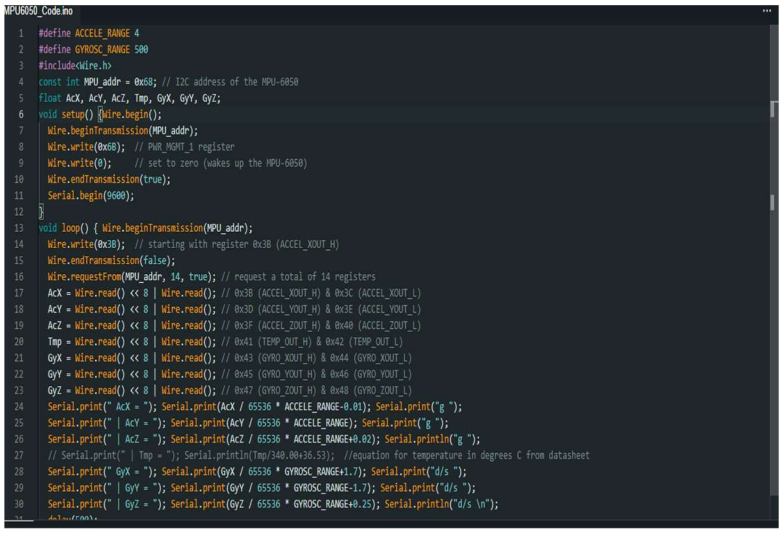
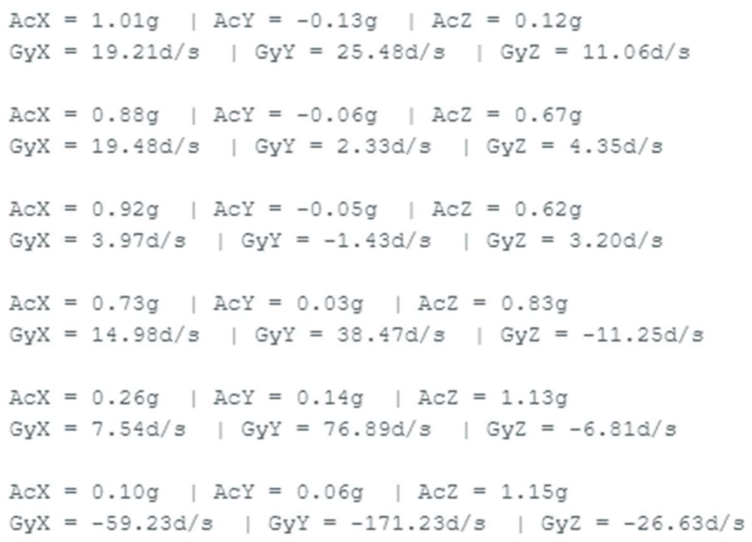


Figure 2 - MPU6050 Hardware



```
MPU6050_Code.ino
#define ACCELE_RANGE 4
#define GYROSC_RANGE 500
#include<Wire.h>
const int MPU_addr = 0x68; // I2C address of the MPU-6050
float AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
void setup() {Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);  // PWR_MGMT_1 register
  Wire.write(0);     // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
  Serial.begin(9600);
}
void loop() { Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr, 14, true); // request a total of 14 registers
  AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp = Wire.read() << 8 | Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX = Wire.read() << 8 | Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY = Wire.read() << 8 | Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ = Wire.read() << 8 | Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
  Serial.print(" AcX = "); Serial.print(AcX / 65536 * ACCELE_RANGE-0.01); Serial.print("g ");
  Serial.print(" | AcY = "); Serial.print(AcY / 65536 * ACCELE_RANGE); Serial.print("g ");
  Serial.print(" | AcZ = "); Serial.print(AcZ / 65536 * ACCELE_RANGE+0.02); Serial.println("g ");
  // Serial.print(" | Tmp = "); Serial.println(Tmp/340.00+36.53);  //equation for temperature in degrees C from datasheet
  Serial.print(" GyX = "); Serial.print(GyX / 65536 * GYROSC_RANGE+1.7); Serial.print("d/s ");
  Serial.print(" | GyY = "); Serial.print(GyY / 65536 * GYROSC_RANGE-1.7); Serial.print("d/s ");
  Serial.print(" | GyZ = "); Serial.print(GyZ / 65536 * GYROSC_RANGE+0.25); Serial.println("d/s \n");
```

Figure 3 - MPU6050 Code



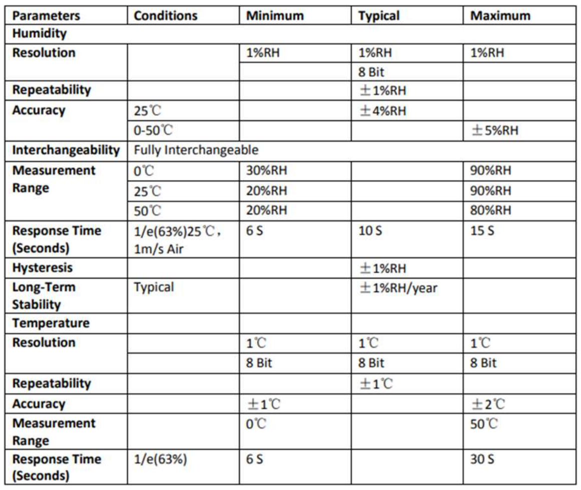
```
AcX = 1.01g  | AcY = -0.13g  | AcZ = 0.12g
GyX = 19.21d/s  | GyY = 25.48d/s  | GyZ = 11.06d/s

AcX = 0.88g  | AcY = -0.06g  | AcZ = 0.67g
GyX = 19.48d/s  | GyY = 2.33d/s  | GyZ = 4.35d/s

AcX = 0.92g  | AcY = -0.05g  | AcZ = 0.62g
GyX = 3.97d/s  | GyY = -1.43d/s  | GyZ = 3.20d/s

AcX = 0.73g  | AcY = 0.03g  | AcZ = 0.83g
GyX = 14.98d/s  | GyY = 38.47d/s  | GyZ = -11.25d/s

AcX = 0.26g  | AcY = 0.14g  | AcZ = 1.13g
GyX = 7.54d/s  | GyY = 76.89d/s  | GyZ = -6.81d/s

AcX = 0.10g  | AcY = 0.06g  | AcZ = 1.15g
GyX = -59.23d/s  | GyY = -171.23d/s  | GyZ = -26.63d/s
```

Figure 5 - MPU6050 Static test Data

The DHT11

The DHT11 is a simple, inexpensive digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin. From findings and basic overview, calibration of the specific DHT11 was minimal.

Test proposal -
To measure almost all the static characteristics of the DHT11 a variety of different environments will be simulated and placing the sensor next to different objects that will affect the temperature and humidity of the readings from the sensor. These readings will be compared to both the datasheet and an additional temperature and humidity measuring device that will serve as a control when compared to the sensor. The DHT will be placed in colder environments that will be compiled to find the sensitivity through graphing the measured against the control. Resolution will be similarly tested however it will be looking for the smallest change detectable be the sensor. All other static characteristics will be tested similarly with comparison to the data spreadsheet, the control measuring device, and the different simulated environments. Control will use Acurite indoor temperature and humidity thermometer.



Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃，1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

- Humidity sensor measures indoor humidity from 1% - 99% RH
- Temperature sensor measures from -4 to 158 degrees Fahrenheit (-20 to 70 Celsius Celsius)
- Helpful humidity meter indicator lets you know if your home is in a safe range
- Measurements are updated every 10 seconds
- Enhanced accuracy: +/- 0.5°F, +/- 2% RH with manual calibration option
- Displays high and low records for the previous 24 hours and for all-time
- Totally wireless tabletop, wall-mountable, and magnet-mountable design

Figure 6 – DHT11 _ Datasheet Specifications/ Acurite Datasheet

DHT11 _ Code/Hardware

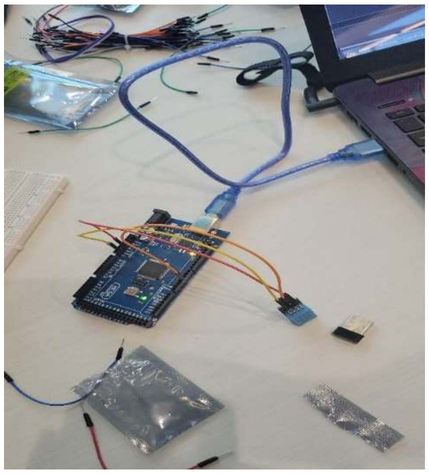
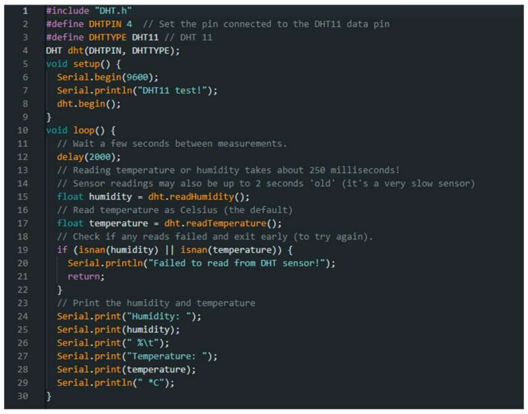
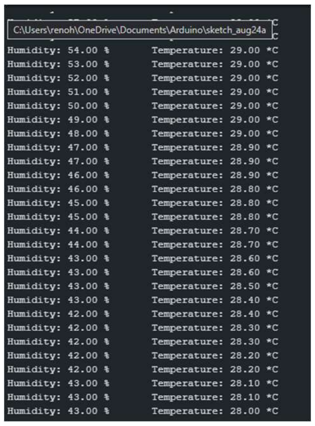


Figure 7 – DHT11_Hardware


```
#include "DHT.h"
#define DHTPIN 4  // Set the pin connected to the DHT11 data pin
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println("DHT11 test!");
  dht.begin();
}
void loop() {
  // Wait a few seconds between measurements.
  delay(2000);
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (it's a very slow sensor)
  float humidity = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float temperature = dht.readTemperature();
  // Check if any reads failed and exit early (to try again).
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // Print the humidity and temperature
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" *C");
}
```
Figure 8 – DHT11_Code



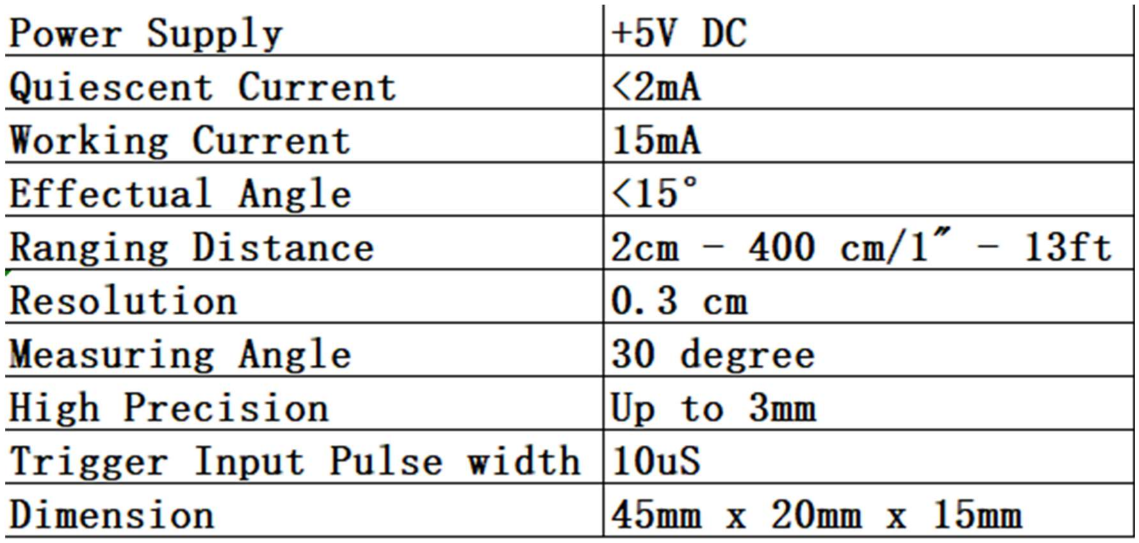
```
Humidity: 54.00 %       Temperature: 29.00 *C
Humidity: 53.00 %       Temperature: 29.00 *C
Humidity: 52.00 %       Temperature: 29.00 *C
Humidity: 51.00 %       Temperature: 29.00 *C
Humidity: 50.00 %       Temperature: 29.00 *C
Humidity: 49.00 %       Temperature: 29.00 *C
Humidity: 48.00 %       Temperature: 29.00 *C
Humidity: 47.00 %       Temperature: 28.90 *C
Humidity: 47.00 %       Temperature: 28.90 *C
Humidity: 46.00 %       Temperature: 28.90 *C
Humidity: 46.00 %       Temperature: 28.80 *C
Humidity: 45.00 %       Temperature: 28.80 *C
Humidity: 45.00 %       Temperature: 28.80 *C
Humidity: 44.00 %       Temperature: 28.70 *C
Humidity: 44.00 %       Temperature: 28.70 *C
Humidity: 43.00 %       Temperature: 28.60 *C
Humidity: 43.00 %       Temperature: 28.60 *C
Humidity: 43.00 %       Temperature: 28.50 *C
Humidity: 43.00 %       Temperature: 28.40 *C
Humidity: 42.00 %       Temperature: 28.40 *C
Humidity: 42.00 %       Temperature: 28.30 *C
Humidity: 42.00 %       Temperature: 28.30 *C
Humidity: 42.00 %       Temperature: 28.20 *C
Humidity: 42.00 %       Temperature: 28.20 *C
Humidity: 43.00 %       Temperature: 28.10 *C
Humidity: 43.00 %       Temperature: 28.10 *C
Humidity: 43.00 %       Temperature: 28.00 *C
```

Figure 9 – DHT11_Output Data

Ultrasonic Ranging Module

The theory of an ultrasonic sensor is to emit a sound wave of a certain frequency at one end of the sensor. When this sound wave contacts the surface of an object, it will rebound back to the receiver, thereby obtaining the distance between the sensor and the measured object. Ultrasonic ranging module provides 2cm (about 0.79 in) - 400cm (about 13.12 ft) non-contact measurement function, and the ranging accuracy can reach to 3mm (about 0.12 in). It can ensure that the signal is stable within 5m, and the signal is gradually weakened after 5m, till the 7m position disappears.

For this part we will find the range, static error, and precision of ultrasonic ranging module. To measure the range, firstly we plan to put an object in front of the ultrasound sensor. We will increase the distance between it and the object gradually (Stop measuring until normal data cannot be displayed). Then we will observe the changes in the measured data and record it down. To get static error, we will first record the data measured between the maximum range and the minimum range and then calculate the error. To get the Precision, we can calculate it by the standard deviation with a set of readings of the sensor for the same input.



Power Supply	+5V DC
Quiescent Current	<2mA
Working Current	15mA
Effectual Angle	<15°
Ranging Distance	2cm – 400 cm/1″ – 13ft
Resolution	0.3 cm
Measuring Angle	30 degree
High Precision	Up to 3mm
Trigger Input Pulse width	10uS
Dimension	45mm x 20mm x 15mm

Figure 10 - Ultrasonic Ranging Module Specifications

Ultrasonic _ Code/Hardware

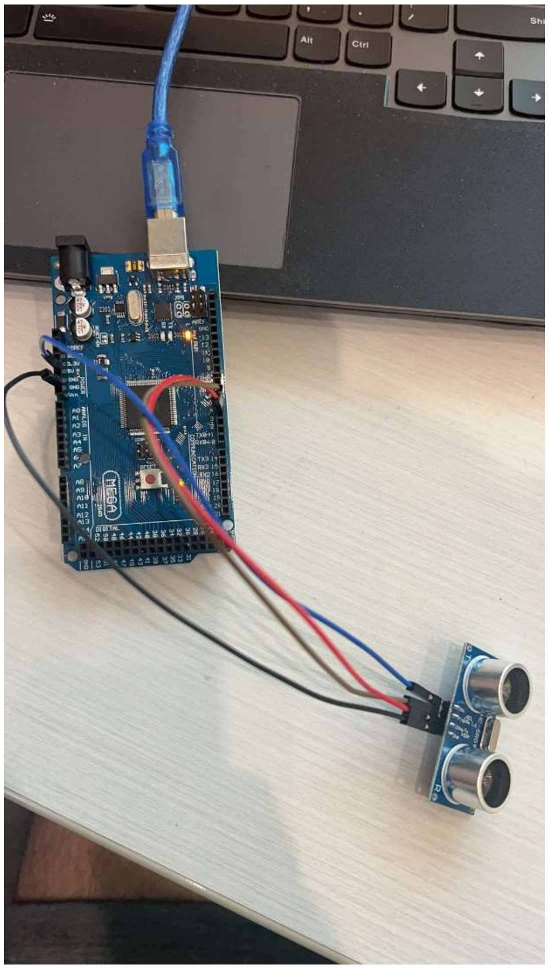
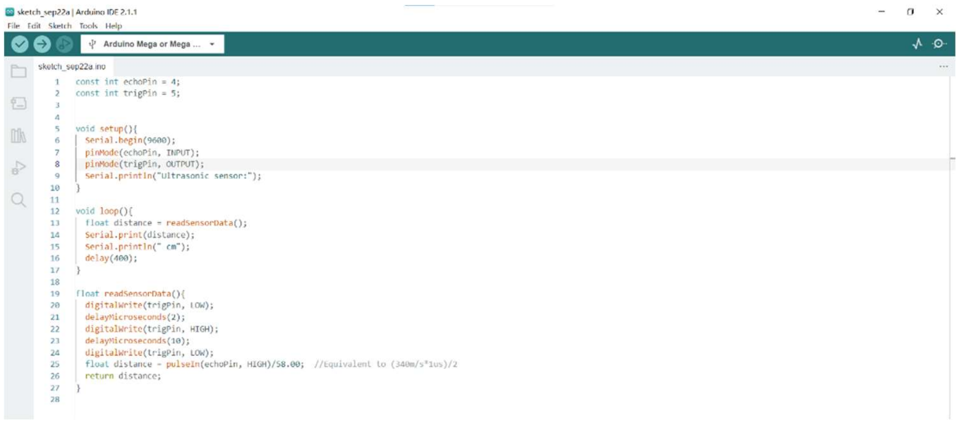
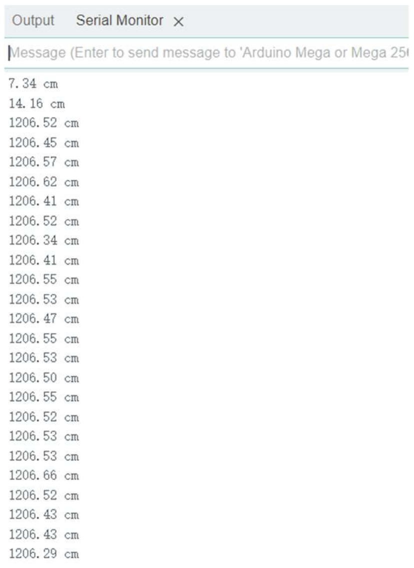


Figure 11 - Ultrasonic Ranging Module Hardware


sketch_sep22a | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Mega or Mega ...
sketch_sep22a.ino

```
const int echoPin = 4;
const int trigPin = 5;


void setup(){
  Serial.begin(9600);
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  Serial.println("Ultrasonic sensor:");
}

void loop(){
  float distance = readSensorData();
  Serial.print(distance);
  Serial.println(" cm");
  delay(400);
}

float readSensorData(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  float distance = pulseIn(echoPin, HIGH)/58.00;  //Equivalent to (340m/s*1us)/2
  return distance;
}

```

Figure 12 - Ultrasonic Ranging Module Code



Output Serial Monitor ×

Message (Enter to send message to 'Arduino Mega or Mega 25

```
7.34 cm
14.16 cm
1206.52 cm
1206.45 cm
1206.57 cm
1206.62 cm
1206.41 cm
1206.52 cm
1206.34 cm
1206.41 cm
1206.55 cm
1206.53 cm
1206.47 cm
1206.55 cm
1206.53 cm
1206.50 cm
1206.55 cm
1206.52 cm
1206.53 cm
1206.53 cm
1206.66 cm
1206.52 cm
1206.43 cm
1206.43 cm
1206.29 cm
```

Figure 13 - Ultrasonic Ranging Module Output specifications