

---

# SENSOR DAN AKTUATOR

## BAGIAN II

---

Modul IV Praktikum Sistem Tertanam



INSTITUT TEKNOLOGI SUMATERA  
LAMPUNG SELATAN  
2022

## I. Daftar Isi

<b>I. Daftar Isi</b>	<b>1</b>
<b>II. Persiapan Praktikum</b>	<b>2</b>
<b>III. Pendahuluan</b>	<b>4</b>
A. Sensor IMU	4
B. Sensor Ultrasonik	4
<b>IV. Percobaan</b>	<b>6</b>
A. Range Measurement	6
B. Inertial Measurement	8
<b>V. Challenge</b>	<b>13</b>
A. Know One's Limits	13
B. Spatial Awareness	13
<b>VI. Referensi</b>	<b>15</b>

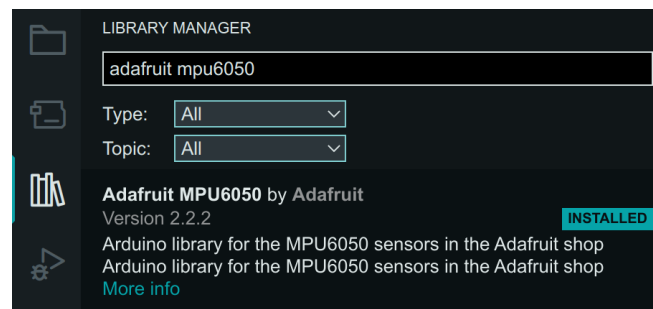
## II. Persiapan Praktikum

### A. Alat & Bahan

1. ESP32
2. Arduino IDE
3. Kabel jumper
4. Breadboard
5. Sensor jarak ultrasonik (HC-SR04)
6. Sensor IMU 6-DOF (MPU-6050)
7. Sensor suhu (DHT-11)
8. 3 buah LED (merah, kuning, hijau) dan 1 buah buzzer
9. Berbagai benda tambahan (**lihat bagian challenge A**)

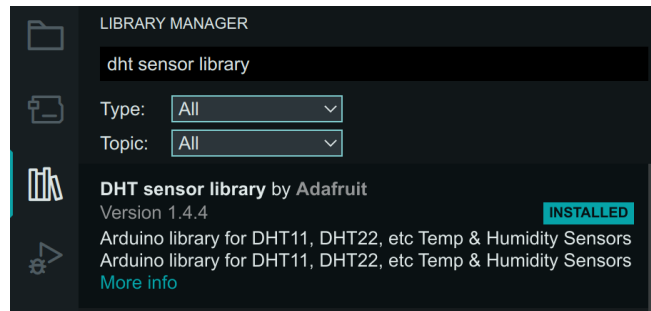
### B. Library Arduino IDE

1. Adafruit MPU6050
  - a) *Download library Adafruit MPU6050* melalui [link berikut](#) atau cari dan *install* langsung melalui “Library Manager” pada Arduino IDE



Gambar 1 Library “Adafruit MPU6050” pada Library Manager

- b) Pastikan dependensi *library* Adafruit MPU6050 terpenuhi semua dengan meng-*install* juga *library* lainnya seperti **Adafruit BusIO**, **Adafruit Unified Sensor Driver**, **Adafruit GFX Library**, dan **Adafruit SSD1306**. Tahap ini dapat dilakukan secara otomatis melalui Library Manager dengan memilih opsi “Install all” saat memasang *library* Adafruit MPU6050
2. DHT sensor library
  - a) *Library* ini telah di-*install* pada praktikum minggu sebelumnya. Bila belum terpasang, silahkan *download library Adafruit DHT sensor library* melalui [link berikut](#) atau cari dan *install* langsung melalui “Library Manager” pada Arduino IDE



Gambar 2 Library “DHT sensor library” pada Library Manager

- b) Pastikan *library* [Adafruit Unified Sensor Driver](#) juga telah terpasang sebagai dependensi dari **Adafruit DHT sensor library** (terpasang otomatis bila memilih opsi “Install all”)

### III. Pendahuluan

#### A. Sensor IMU

Sensor IMU (*Inertial Measurement Unit*) pertama kali digunakan sekitar tahun 1930 untuk kebutuhan penerbangan [1]. Sensor IMU umumnya terdiri dari 2 komponen utama yaitu *accelerometer* (untuk mengukur akselerasi atau percepatan) dan *gyroscope* (untuk mengukur kecepatan angular) yang masing-masing memiliki nilai pada sumbu X, Y, dan Z [2]. Seiring berkembangnya zaman, sensor IMU menjadi semakin efisien, murah, dan berukuran kecil dengan adanya MEMS (*microelectromechanical systems*) [3]. Oleh karena ukuran dan efisiensinya, sensor IMU banyak digunakan pada benda dalam kehidupan sehari-hari seperti *handphone* (kompas, *pocket detection*, dsb), *headset VR* dan *drone* (orientasi), serta kamera (stabilisasi gambar/video).



**Gambar 3** Sensor IMU digunakan pada headset VR untuk orientasi arah [\[source\]](#)

Beberapa contoh proyek menarik yang memakai sensor IMU antara lain:

1. [Relativ - Build Your Own VR Headset for \\$100](#)
2. [Using Ninjutsu Hand Signs \(Naruto\) to Unlock a Safe](#)
3. [Level Platform Using Accelerometer](#)
4. [Punch Activated Arm Flamethrowers \(Real Firebending\)](#)
5. [How to Make a Wristband Pedometer Based on BMI160 Motion](#)

#### B. Sensor Ultrasonik

Sensor ultrasonik merupakan salah satu tipe sensor jarak yang dapat bekerja lewat udara tanpa kontak secara langsung [4]. Frekuensi gelombang yang umum digunakan pada sensor ultrasonik adalah 40,000 Hz [5]. Agar dapat

berfungsi, sensor ultrasonik membutuhkan 2 buah komponen penting yaitu transmitter (untuk mengirim sinyal) dan receiver (untuk menerima sinyal) [6], atau gabungan keduanya (transceiver). Pada level kehidupan sehari-hari, sensor ultrasonik banyak diimplementasikan pada produk elektronik seperti *drone* dan *roomba* (*collision detection*), keran dan *hand dryer* (menyala otomatis), dll. Meski begitu, sensor ultrasonik yang terkonfigurasi khusus dapat digunakan untuk beberapa hal lainnya seperti untuk USG (ultrasonografi) dan pengukuran kedalaman laut.



**Gambar 4** USG menggunakan ultrasonik untuk mendeteksi bayi dalam kandungan [[source](#)]

Beberapa contoh proyek menarik yang memakai sensor jarak antara lain:

1. [Social Distancing Radar](#)
2. [Touchless Automatic Motion Sensor Trash Can](#)
3. [Arduino Radar with Processing](#)
4. [Wash-A-Lot-Bot! A DIY Hand Washing Timer](#)
5. [Low Power Water Level Sensor for LoRaWAN/The Things Network](#)

## IV. Percobaan

### A. Range Measurement

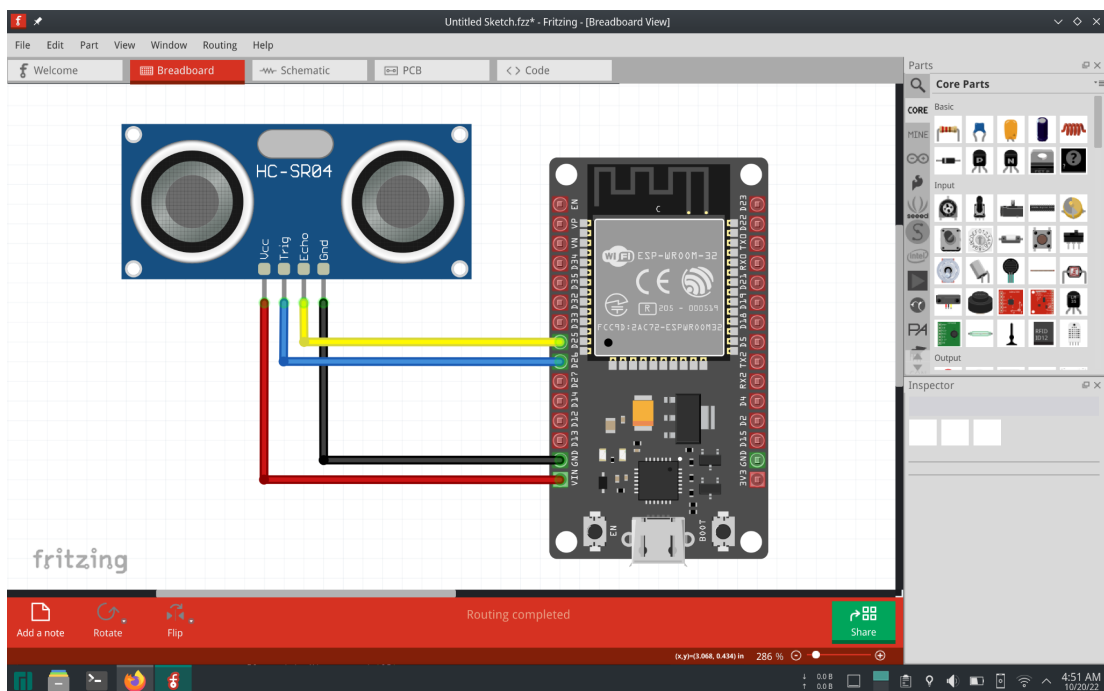
Pada percobaan ini, kita akan mencoba mengukur jarak dari sensor dengan menggunakan sensor ultrasonik HC-SR04 (atau sensor serupa).

#### Persiapan

Pada percobaan ini, tidak diperkenankan menggunakan *library* tambahan untuk mengukur jarak sensor ultrasonik. Meski begitu, untuk penggunaan diluar praktikum, *library* yang populer digunakan adalah [NewPing](#).

#### Rangkaian dan Kode

Mula-mula buatlah rangkaian seperti gambar berikut.



Gambar 5 Rangkaian untuk percobaan HC-SR04

Selanjutnya, tuliskan kode yang tertera di bawah ini pada Arduino IDE.

```

1. // Sambungkan pin 25 (digital) ke pin ECHO pada HC-SR04
2. #define ECHO_PIN 25
3. // Sambungkan pin 26 (digital) ke pin TRIG pada HC-SR04
4. #define TRIG_PIN 26
5.
6. void setup() {
7.   Serial.begin(9600);
8.
9.   // Set pin output dan input
10.  pinMode(TRIG_PIN, OUTPUT);
11.  pinMode(ECHO_PIN, INPUT);
12.
13.  // Reset kondisi TRIG_PIN bila default-nya bernilai HIGH
14.  digitalWrite(TRIG_PIN, LOW);

```

```

15. }
16.
17. void loop() {
18.   // Set TRIG_PIN agar memancarkan gelombang
19.   digitalWrite(TRIG_PIN, HIGH);
20.   // Matikan kembali setelah jeda waktu tertentu
21.   delayMicroseconds(10);
22.   digitalWrite(TRIG_PIN, LOW);
23.
24.   // Hitung durasi pancar gelombang (dalam mikrosekond)
25.   long duration = pulseIn(ECHO_PIN, HIGH);
26.   // Hitung jarak sebenarnya dari sensor ke benda
27.   int distance = duration * 0.034 / 2;
28.   // Tampilkan jarak pada serial monitor
29.   Serial.println((String) "Jarak: " + distance + " cm");
30.   delay(500);
31. }

```

Upload kode ke mikrokontroler dan letakkan benda di depan sensor ultrasonik. Dengan mengecek “Serial Monitor” pada Arduino IDE, bagaimana hasilnya?

### Pertanyaan / Asprak Bertanya

1. Bagaimana sensor/gelombang ultrasonik bekerja? Bila salah satu lubang sonar pada sensor tertutup apa yang akan terjadi?
2. Apakah sensor ultrasonik yang kalian pakai dapat menembus permukaan air? Mengapa demikian?

Dapat dicoba pada air dalam gelas/gayung/ember

3. Berdasarkan pengamatan, berapa jarak maksimal yang dapat diukur sensor ultrasonik kalian? Apakah hal tersebut sesuai dengan *datasheet* sensor?

Sebutkan juga model sensor yang dipakai (bila bukan HC-SR04), misal HY-SRF05 atau US-100

4. Mengapa menggunakan “delayMicroseconds(10)” pada kode? Adakah pengaruh terhadap jarak minimal/maksimal bila delay tersebut diubah?

HINT: Baca dokumentasi pulseIn dan rumus jarak/kecepatan suara di udara, berikan bukti perhitungan bila dirasa perlu

5. Sensor ultrasonik merupakan salah satu tipe sensor jarak yang banyak digunakan. Apa perbedaan sensor ultrasonik dengan tipe sensor jarak lainnya?

Contoh sensor jarak selain ultrasonik misalnya sensor induktif, dll. Perbedaan boleh ditulis dari segi cara kerja, kelemahan, atau kelebihan

## B. Inertial Measurement

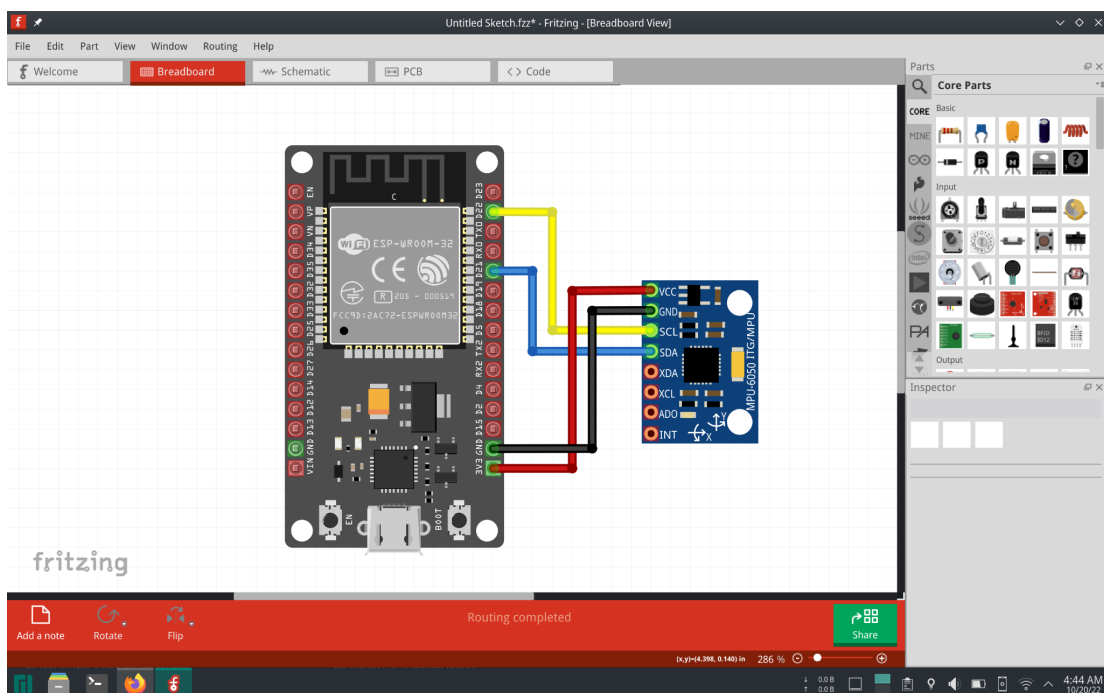
Pada percobaan ini, kita akan mencoba mengukur percepatan dan kecepatan angular dengan menggunakan sensor IMU 6-DOF MPU-6050.

### Persiapan

Install terlebih dahulu *library* [Adafruit MPU6050](#) seperti yang telah dijelaskan di bagian awal modul beserta *library* dependensinya (Adafruit BusIO, Adafruit Unified Sensor Driver, Adafruit GFX Library, dan Adafruit SSD1306).

### Rangkaian dan Kode

Mula-mula buatlah rangkaian seperti gambar berikut. **Cek datasheet/pinout** mikrokontroler masing-masing untuk letak pasti pin SCL dan SDA.



Gambar 6 Rangkaian untuk percobaan MPU-6050

Selanjutnya, tuliskan kode yang tertera di bawah ini pada Arduino IDE.

```

1. #include <Adafruit_MPU6050.h>
2. #include <Adafruit_Sensor.h>
3. #include <Wire.h>
4.
5. Adafruit_MPU6050 mpu;
6. float ax_offset, ay_offset, az_offset;
7. float gx_offset, gy_offset, gz_offset;
8. void calibrate_mpu();
9.
10. void setup(void) {
11.   Serial.begin(115200);
12.
13.   // Inisialisasi sensor
14.   while (!mpu.begin()) {

```

```
15.     Serial.println("Tidak dapat menemukan MPU-6050!");
16. }
17.
18. mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
19. Serial.print("Rentang accelerometer diset ke ");
20. switch (mpu.getAccelerometerRange()) {
21. case MPU6050_RANGE_2_G:
22.     Serial.println("+2G");
23.     break;
24. case MPU6050_RANGE_4_G:
25.     Serial.println("+4G");
26.     break;
27. case MPU6050_RANGE_8_G:
28.     Serial.println("+8G");
29.     break;
30. case MPU6050_RANGE_16_G:
31.     Serial.println("+16G");
32.     break;
33. }
34.
35. mpu.setGyroRange(MPU6050_RANGE_500_DEG);
36. Serial.print("Rentang gyroscope diset ke ");
37. switch (mpu.getGyroRange()) {
38. case MPU6050_RANGE_250_DEG:
39.     Serial.println("+ 250 deg/s");
40.     break;
41. case MPU6050_RANGE_500_DEG:
42.     Serial.println("+ 500 deg/s");
43.     break;
44. case MPU6050_RANGE_1000_DEG:
45.     Serial.println("+ 1000 deg/s");
46.     break;
47. case MPU6050_RANGE_2000_DEG:
48.     Serial.println("+ 2000 deg/s");
49.     break;
50. }
51.
52. mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
53. Serial.print("Filter bandwidth diset ke ");
54. switch (mpu.getFilterBandwidth()) {
55. case MPU6050_BAND_260_HZ:
56.     Serial.println("260 Hz");
57.     break;
58. case MPU6050_BAND_184_HZ:
59.     Serial.println("184 Hz");
60.     break;
61. case MPU6050_BAND_94_HZ:
62.     Serial.println("94 Hz");
63.     break;
64. case MPU6050_BAND_44_HZ:
65.     Serial.println("44 Hz");
66.     break;
67. case MPU6050_BAND_21_HZ:
68.     Serial.println("21 Hz");
```

```

69.     break;
70. case MPU6050_BAND_10_HZ:
71.     Serial.println("10 Hz");
72.     break;
73. case MPU6050_BAND_5_HZ:
74.     Serial.println("5 Hz");
75.     break;
76. }
77.
78. // Kalibrasi sensor (harap diamkan sensor)
79. calibrate_mpu();
80. Serial.println("");
81. delay(5000);
82. }
83.
84. void loop() {
85. // Ambil hasil bacaan dari sensor
86. sensors_event_t a, g, temp;
87. mpu.getEvent(&a, &g, &temp);
88.
89. // Tampilkan hasil bacaan
90. Serial.print("Percepatan aX: ");
91. Serial.print(a.acceleration.x - ax_offset);
92. Serial.print(", aY: ");
93. Serial.print(a.acceleration.y - ay_offset);
94. Serial.print(", aZ: ");
95. Serial.print(a.acceleration.z - az_offset);
96. Serial.println(" m/s^2");
97.
98. Serial.print("Rotasi gX: ");
99. Serial.print(g.gyro.x - gx_offset);
100. Serial.print(", gY: ");
101. Serial.print(g.gyro.y - gy_offset);
102. Serial.print(", gZ: ");
103. Serial.print(g.gyro.z - gz_offset);
104. Serial.println(" rad/s");
105.
106. Serial.print("Temperatur: ");
107. Serial.print(temp.temperature);
108. Serial.println(" degC");
109.
110. Serial.println("");
111. delay(500);
112. }
113.
114. void calibrate_mpu() {
115. Serial.println("\nLetakkan MPU-6050 di bidang datar yang stabil!");
116. for (int i = 7; i > 0; i--) {
117.     Serial.println((String) "Kalibrasi dalam " + i + " detik...");
118.     delay(1000);
119. }
120.
121. Serial.println("Menjalankan proses kalibrasi (harap tunggu!");
122.

```

```

123.   int total_loop = 3000;
124.   float total_ax = 0, total_ay = 0, total_az = 0;
125.   float total_gx = 0, total_gy = 0, total_gz = 0;
126.
127.   for (int i = 0; i < total_loop; i++) {
128.       sensors_event_t a, g, temp;
129.       mpu.getEvent(&a, &g, &temp);
130.
131.       total_ax += a.acceleration.x;
132.       total_ay += a.acceleration.y;
133.       total_az += a.acceleration.z;
134.
135.       total_gx += g.gyro.x;
136.       total_gy += g.gyro.y;
137.       total_gz += g.gyro.z;
138.   }
139.
140.   // Rata-rata hasil keseluruhan untuk menghasilkan offset
141.   ax_offset = total_ax / total_loop;
142.   ay_offset = total_ay / total_loop;
143.   az_offset = total_az / total_loop - 9.8;
144.
145.   gx_offset = total_gx / total_loop;
146.   gy_offset = total_gy / total_loop;
147.   gz_offset = total_gz / total_loop;
148.
149.   // Tampilkan hasil offset
150.   Serial.print("\nOffset aX: ");
151.   Serial.print(ax_offset);
152.   Serial.print(", aY: ");
153.   Serial.print(ay_offset);
154.   Serial.print(", aZ: ");
155.   Serial.print(az_offset);
156.   Serial.println(" m/s^2");
157.
158.   Serial.print("Offset gX: ");
159.   Serial.print(gx_offset);
160.   Serial.print(", gY: ");
161.   Serial.print(gy_offset);
162.   Serial.print(", gZ: ");
163.   Serial.print(gz_offset);
164.   Serial.println(" rad/s");
165. }

```

Link kode secara utuh: <https://pastebin.com/raw/7PQLkcV0>.

Harap diamkan MPU-6050 dengan posisi sejajar (jangan miring-miring) dengan alas sensor (meja, lantai, dsb) saat proses kalibrasi.

Upload kode ke mikrokontroler dan gerakkan sensor MPU-6050 ke berbagai arah. Dengan mengecek "Serial Monitor" pada Arduino IDE, bagaimana hasilnya?

### **Pertanyaan / Asprak Bertanya**

1. Bandingkan nilai temperatur yang dihasilkan oleh MPU-6050 dan DHT-11 dengan nilai temperatur sekitar yang didapatkan termometer/HP kalian/internet, manakah diantara kedua sensor yang lebih akurat?

Sebutkan juga nilai ketiga temperatur yang didapatkan

2. Apa itu 6-DOF dan bagaimana cara kerja accelerometer dan gyroscope?

NOTE: Pada gyroscope juga terdapat 3 istilah gerakan yaitu roll, pitch, dan yaw yang masing-masing bisa disebutkan perbedaannya

3. Mengapa nilai percepatan sumbu X dan Y pada accelerometer mendekati 0 sedangkan nilai pada sumbu Z jauh diatas 0?

HINT: Baca mengenai inersia serta energi kinetik/potensial

4. Accelerometer (dan gyroscope) umumnya memberikan nilai dalam 3 dimensi yang berbeda (X, Y, dan Z), bagaimana cara menggabungkan nilai ketiganya menjadi 1 kesatuan/arah (1 dimensi) saja?

HINT: Baca mengenai vektor serta teori pitagoras

5. Menurutmu, selain untuk mengukur percepatan ( $m/s^2$ ), apakah sensor IMU dapat digunakan/dimodifikasi untuk mengukur kecepatan biasa ( $m/s$ )?

OPSIONAL: Bonus poin apabila dapat membuktikan dalam contoh kasus dan rumusan

## V. Challenge

### A. Know One's Limits

Bagaimana pengaruh berbagai permukaan/bentuk benda terhadap pembacaan sensor ultrasonik? Bandingkan hasilnya dengan penggaris/alat ukur biasa!

Untuk membantu menjawab *challenge* ini, dapat dilakukan pengetesan jarak pada berbagai contoh permukaan benda abnormal seperti berikut (sertakan bukti gambar):

1. Cermin, metal (permukaan reflektif dengan sudut pantul tidak lurus atau diatas  $0^\circ$ ; tes terhadap berbagai besaran sudut)
2. Plastik, kaca (permukaan transparan yang tidak terlalu tebal dan tipis)
3. Api (yang dinyalakan di depan lubang transmitter ultrasonik)
4. Boneka/bantal/peliharaan berbulu lebat, sapu/sikat lantai dalam posisi vertikal (atau permukaan berbulu/tidak rata lainnya)
5. Saringan air, *pen holder* (atau permukaan berongga/berlubang lainnya)

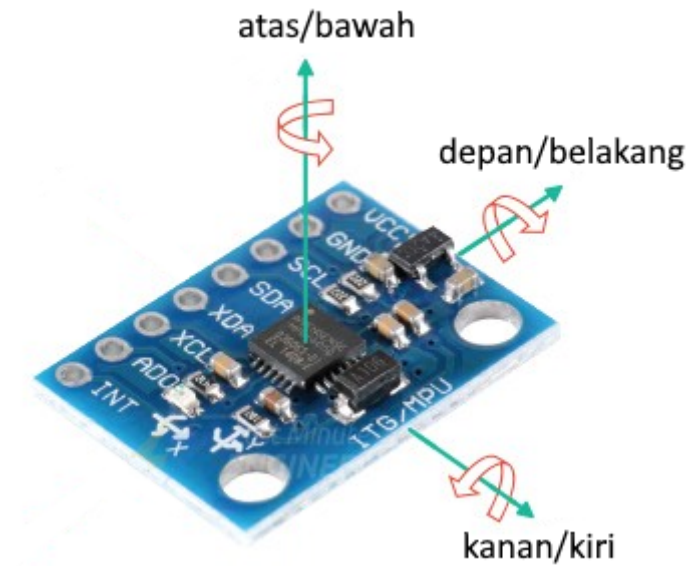
NOTE: Cukup diwajibkan memilih 3 dari 5 opsi yang ada. Untuk benda per opsinya, cukup gunakan 1 benda saja yang dianggap telah memenuhi kriteria (misal bila ada kaca tidak perlu mencoba pada plastik lagi). Lalu, percobaan pada permukaan air juga diperlukan untuk menjawab "asprak bertanya".

### B. Spatial Awareness

Seperti yang telah kalian ketahui, *accelerometer* dan *gyroscope* mempunyai nilai pada 3 sumbu X, Y, dan Z. Namun, gerakan dan arah apa saja yang mempengaruhi nilai-nilai tersebut?

Untuk membantu menjawab *challenge* ini, dapat dilakukan pengetesan terhadap berbagai gerakan/arah berikut (bukti boleh berupa gambar atau link video demonstrasi):

1. Bila *accelerometer* bergerak ke atas (secara cepat), maka lampu LED merah menyala. Bila bergerak ke bawah, maka buzzer juga ikut menyala
2. Bila *accelerometer* bergerak ke kanan (secara cepat), maka lampu LED kuning menyala. Bila bergerak ke kiri maka buzzer juga ikut menyala
3. Bila *accelerometer* bergerak ke depan (secara cepat), maka lampu LED hijau menyala. Bila bergerak ke belakang maka buzzer juga ikut menyala
4. Bila *gyroscope* berputar di atas/bawah (secara cepat), maka lampu LED merah dan kuning menyala. Bila arah putaran berkebalikan (negatif) maka buzzer juga ikut menyala
5. Bila *gyroscope* berputar di kanan/kiri (secara cepat), maka lampu LED merah dan hijau menyala. Bila arah putaran berkebalikan (negatif) maka buzzer juga ikut menyala
6. Bila *gyroscope* berputar di depan/belakang (secara cepat), maka lampu LED kuning dan hijau menyala. Bila arah putaran berkebalikan (negatif) maka buzzer juga ikut menyala



**Gambar 7** Gambaran letak putaran saat pengetesan *gyroscope*

NOTE: Untuk bukti gambar/video di laporan dapat disertakan 3 contoh saja, tidak wajib semua arah/gerakan didokumentasikan. Namun, kesimpulan tetap mencakup keseluruhan gerakan/arah (misal gerakan ke atas mewakili nilai pada sumbu apa, dst). Bila sensor tidak akurat atau sulit ditebak juga dapat dituliskan pengamatannya pada bagian analisis.

## VI. Referensi

- [1] H. Zhao and Z. Wang, "Motion Measurement Using Inertial Sensors, Ultrasonic Sensors, and Magnetometers With Extended Kalman Filter for Data Fusion," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 943–953, May 2012, doi: 10.1109/JSEN.2011.2166066.
- [2] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications," *IJSPS*, pp. 256–262, 2013, doi: 10.12720/ijsp.1.2.256-262. [Online]. Available: <http://www.ijsp.1.2.256-262>. [Accessed: Oct. 20, 2022]
- [3] M. Tanenhaus, D. Carhoun, T. Geis, E. Wan, and A. Holland, "Miniature IMU/INS with optimally fused low drift MEMS gyro and accelerometers for applications in GPS-denied environments," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, Apr. 2012, pp. 259–264, doi: 10.1109/PLANS.2012.6236890.
- [4] N. A. Latha, B. R. Murthy, and K. B. Kumar, "Distance Sensing with Ultrasonic Sensor and Arduino," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 2, no. 5, Dec. 2016 [Online]. Available: <https://www.ijariit.com/manuscript/distance-sensing-ultrasonic-sensor-arduino/>. [Accessed: Oct. 20, 2022]
- [5] M. Ishihara, M. Shiina, and S. Suzuki, "Evaluation of Method of Measuring Distance Between Object and Walls Using Ultrasonic Sensors," *Journal of Asian Electric Vehicles*, vol. 7, no. 1, pp. 1207–1211, 2009, doi: 10.4130/jaev.7.1207.
- [6] V. A. Zhmud, N. O. Kondratiev, K. A. Kuznetsov, V. G. Trubin, and L. V. Dimitrov, "Application of ultrasonic sensor for measuring distances in robotics," *J. Phys.: Conf. Ser.*, vol. 1015, p. 032189, May 2018, doi: 10.1088/1742-6596/1015/3/032189. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1015/3/032189>. [Accessed: Oct. 20, 2022]