

Funcionamento do Sensor

Instrumentação Eletrônica para Engenharia

Felipe Costa Gomes, Davi Dantas Lopes

Engenharia Eletrônica

Universidade de Brasília - Faculdade do Gama

Gama, Distrito Federal

E-mails: 190012757@aluno.unb.br , 170031934@aluno.unb.br

I. INTRODUÇÃO

O MPU-6050 é um dispositivo compacto que oferece rastreamento de movimento integrado de seis eixos, proporcionando seis graus de liberdade. Ele une um giroscópio de três eixos, um acelerômetro de três eixos e um processador de movimento digital em um pacote de dimensões $24 \times 4 \times 9 \text{ mm}$. Com um barramento I^2C , é possível conectar uma bússola externa de três eixos para obter uma saída de fusão de movimento completa com nove eixos. Além disso, o MPU-6050 possui três conversores analógico-digital de 16 bits para digitalizar as saídas do giroscópio e outras três para digitalizar as saídas do acelerômetro [1]. Para garantir um rastreamento preciso de movimentos lentos e rápidos, o dispositivo ainda conta com um giroscópio e um acelerômetro programáveis.

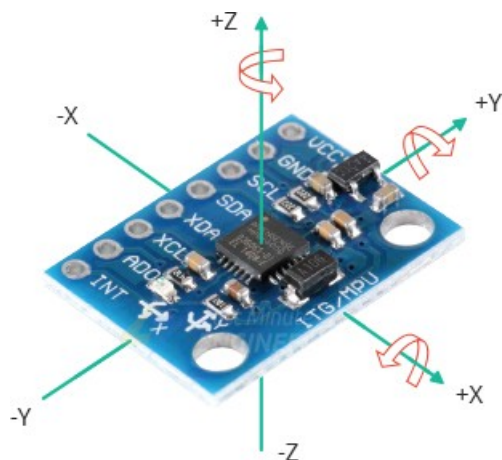


Figura 1. A MPU-6050 representa os 6 graus de liberdade por meio de 6 eixos, sendo 3 dedicados ao giroscópio e outros 3 dedicados ao acelerômetro.

II. FUNCIONAMENTO DO ACELERÔMETRO

A. Princípios de Funcionamento

Os sensores de movimento são projetados para medir a taxa de mudança de posição, localização ou deslocamento de um objeto. Se a posição de um objeto em função do tempo é representada por $x(t)$, então a primeira derivada fornece a velocidade do objeto $v(t)$, se uma direção também é especificada. Se a velocidade do objeto está mudando, então a primeira derivada da velocidade resulta na aceleração, que

é também a segunda derivada da posição. Dessa forma, a equação para a velocidade do objeto é dada por:

$$v(t) = \frac{dx(t)}{dt} \quad (1)$$

Sendo assim, a equação para a aceleração do objeto é dada por:

$$a(t) = \frac{dv(t)}{dt} = \frac{d^2x(t)}{dt^2} \quad (2)$$

O sensor de movimento primário é o acelerômetro, que mede a aceleração $a(t)$ de um objeto. Integrar as equações 1 e 2 permite determinar a velocidade e a posição do objeto [2]. Portanto, o acelerômetro é capaz de fornecer informações sobre aceleração, velocidade ou posição. A aceleração em um acelerômetro é geralmente expressa pela comparação com a aceleração devido à gravidade na superfície da Terra, que é de aproximadamente $9,81 \text{ m/s}^2$.

Existem vários processos físicos que podem ser utilizados para desenvolver um sensor de aceleração. Acelerômetros com base em massas rotativas podem ser encontrados, assim como projetos que combinam a lei de Newton da aceleração da massa e a lei de Hooke da ação da mola [2].

B. Sistema Massa-mola Simplificado

A segunda lei de Newton estabelece que a aplicação de uma força em um objeto produzirá um movimento cuja aceleração é diretamente proporcional à sua massa. Logo:

$$F = m \cdot a \quad (3)$$

A lei de Hooke estabelece que, se uma mola de constante k estiver esticada de uma posição de equilíbrio Dx , então haverá uma força na mola dada por:

$$F = k \cdot dx \quad (4)$$

Na Fig.2, é ilustrado o princípio básico do acelerômetro, sendo possível observar uma massa que desliza livremente sobre uma base, estando conectada a esta por meio de uma mola em seu estado natural não deformado [2]. Quando uma aceleração é aplicada ao conjunto, a mola se estende para fornecer a força necessária para acelerar a massa. Essa relação pode ser descrita pela combinação das equações de Newton e de Hooke, resultando em:

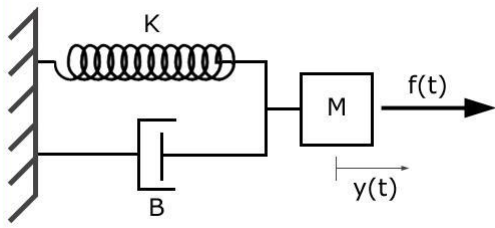


Figura 2. Sistema básico de um acelerômetro do tipo massa-mola.

$$m \cdot a = k \cdot \Delta x \quad (5)$$

Onde:

k = Constante da mola $[N/m]$;

Δx = Extensão da mola $[m]$

m = Massa $[kg]$

a = Aceleração $[m/s^2]$

A partir da equação 5, é viável medir a aceleração ao reduzi-la a uma medida de extensão linear da mola, isto é:

$$a = \frac{k}{m} \cdot \Delta x \quad (6)$$

É possível notar que a equação 6 só serve para casos estáticos [2].

C. Acelerômetro em Sistemas Estáticos e Não Estáticos

Existem vários tipos de acelerômetros disponíveis para medir a aceleração, tais como cristais piezoelétricos, capacitores diferenciais, sensores piezoresistivos, servos-acelerômetros, entre outros [2]. No entanto, a maioria desses dispositivos opera com o mesmo princípio básico do sistema massa-mola simplificado, descrito anteriormente. Os acelerômetros podem ser divididos em dois tipos em termos de funcionalidade: aqueles que são capazes de medir a aceleração estática, como a aceleração devida à gravidade, e aqueles que são insensíveis a esse tipo de aceleração. Para estudos cinéticos no corpo humano, os acelerômetros piezoresistivos ou capacitivos são os mais utilizados [2][3].

Os acelerômetros piezoresistivos são construídos com um arranjo de microestruturas, no qual molas de polissilício, que têm sua resistência elétrica afetada pela aceleração, são dispostas em uma configuração em ponte de Wheatstone. Quando uma aceleração externa é aplicada, a deformação dessas molas desequilibra a ponte, gerando um sinal de saída proporcional à intensidade da aceleração [2].

Os acelerômetros capacitivos tipicamente usam uma geometria diferencial (conhecida como capacitor diferencial) para reduzir os efeitos não lineares. Nessa configuração, a placa central móvel está acoplada à massa sísmica e as placas externas são fixas. A aceleração aplicada causa o movimento relativo das placas. Esse deslocamento resulta na variação da capacitância equivalente do sistema, expressa pela Equação 7,

em que A é a área de cada placa, ϵ é a constante dielétrica do material isolante e d é o afastamento inicial entre as placas. Essa variação de capacitância é, então, convertida em tensão pelo condicionador de sinais para produzir o sinal de saída do acelerômetro [2].

$$\Delta C = \frac{2 \cdot A \cdot \epsilon}{d^2} \cdot \Delta d \quad (7)$$

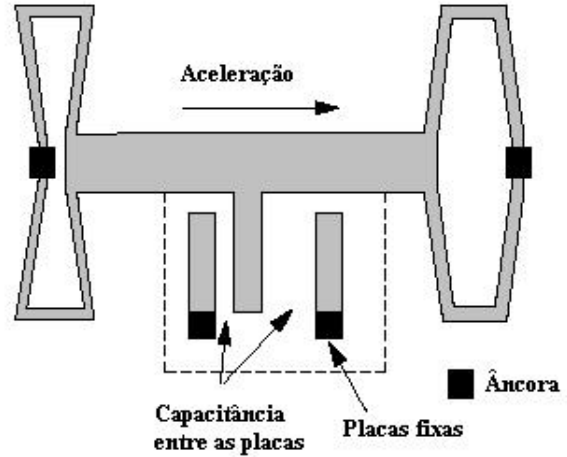


Figura 3. Diagrama de funcionamento do acelerômetro capacitivo.

O acelerômetro do MPU-6050 é fabricado por meio da tecnologia MEMS (*Micro-Electro-Mechanical Systems*), que consiste na integração de elementos mecânicos, como sensores e atuadores, com componentes eletrônicos em um único substrato de silício.

D. Especificações do acelerômetro o MPU-6050

Características do MPU-6050
O acelerômetro possui três eixos e é capaz de fornecer saída digital. Além disso, sua faixa de fundo de escala é programável e pode ser ajustada para $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$.
Os conversores analógico-digital (ADCs) integrados de 16 bits permitem a amostragem simultânea dos acelerômetros sem a necessidade de um multiplexador.
Corrente operacional normal do acelerômetro é de $500\mu A$
Corrente do modo de acelerômetro de baixa potência: $10\mu A$ a 1.25Hz, $20\mu A$ a 5 Hz, $60\mu A$ a 20 Hz, $110\mu A$ a 40 Hz
Possui detecção e sinalização de orientação
Detecção de toque
Interrupções programáveis
Interrupção high-g e autoteste do usuário

Vejamos abaixo uma tabela com especificações do acelerômetro. com $V_{DD} = 2.375V - 3.46V$, VLOGIC (MPU-6050 only) = $1.8V \pm 5\%$ ou V_{DD} , $TA = 25^\circ C$ [1].

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		g	
	AFS_SEL=1		±4		g	
	AFS_SEL=2		±8		g	
	AFS_SEL=3		±16		g	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/g	
	AFS_SEL=1		8,192		LSB/g	
	AFS_SEL=2		4,096		LSB/g	
	AFS_SEL=3		2,048		LSB/g	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes		±50		mg	1
	Z axis		±80		mg	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35		mg	
	Z axis, 0°C to +70°C		±60		mg	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		µg/√Hz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		260	Hz	
OUTPUT DATA RATE						
	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT						
			32		mg/LSB	

Figura 4. Tabela com dados da funcionalidade do acelerômetro.

E. Extração dos dados e Visualização na tela do computador no tempo de execução do código

O acelerômetro tem como função medir a aceleração, que é a taxa de variação de velocidade do objeto. Ele pode detectar forças estáticas, como a gravidade ($9,8 \text{ m/s}^2$), ou forças dinâmicas, como vibrações e movimentos. O MPU-6050 é capaz de medir a aceleração nos eixos x , y e z .

Para obter as leituras de aceleração (x, y, z) do acelerômetro, foi necessário utilizar uma ESP32, uma Protoboard e fios de *jumper*. A extração dos dados de aceleração foi realizada por meio da implementação de um código na plataforma da IDE Arduino e sua posterior transferência para o sistema embarcado por meio do barramento serial I^2C .

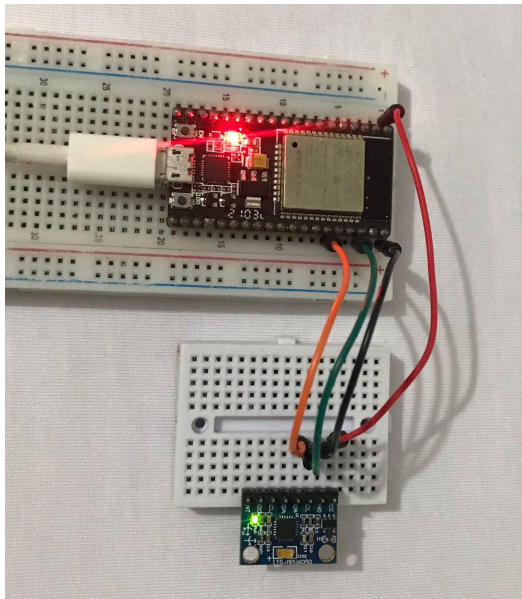


Figura 5. O circuito de conexão entre o sistema embarcado e a MPU-6050 foi utilizado para extrair dados de aceleração. O jumper vermelho foi utilizado para conectar o VDD, o jumper preto para o GND, o jumper laranja para o SDA e o jumper verde para o SCL.

Para elaboração do código, anexo IV.I, foram utilizadas as bibliotecas Adafruit_MPU6050 e Adafruit Sensor [4]. Após a inclusão dessas bibliotecas, um objeto "mpu" foi criado para interagir com o sensor. A taxa de transmissão do monitor foi definida como 115200 e o sensor foi inicializado. Em seguida, foi necessário configurar a faixa de medição do acelerômetro e do giroscópio, bem como a largura de banda do filtro. Dentro do loop, os valores do sensor foram obtidos e as leituras de aceleração foram impressas para cada eixo (x, y, z), com as medidas em metros por segundo quadrado. A leitura da velocidade angular do giroscópio também foi realizada para cada eixo (x, y, z), com as medidas em radianos por segundo (rad/s). Novas leituras do sensor foram exibidas a cada 500 milissegundos [5].

COM4	
Acceleration X: 0.88, Y: 0.59, Z: 9.27 m/s^2	
Acceleration X: 0.84, Y: 0.53, Z: 9.27 m/s^2	
Acceleration X: 0.89, Y: 0.55, Z: 9.24 m/s^2	
Acceleration X: 0.88, Y: 0.53, Z: 9.25 m/s^2	
Acceleration X: 0.79, Y: 0.53, Z: 9.28 m/s^2	
Acceleration X: 0.93, Y: 0.56, Z: 9.24 m/s^2	
Acceleration X: 0.91, Y: 0.52, Z: 9.24 m/s^2	
Acceleration X: 0.80, Y: 0.51, Z: 9.26 m/s^2	

Figura 6. Dados coletados a partir da porta serial no Arduino IDE.

III. CONCLUSÕES

Idealmente, em um objeto estático, a aceleração no eixo z é igual à aceleração gravitacional e deve ser zero nos eixos x e y . Entretanto, ao extrair os dados, observou-se que a aceleração nos eixos x e y apresentou pequenas variações com valores próximos de zero. Essas variações podem ser atribuídas ao valor típico de tolerância de calibração inicial zero-g, que apresenta variação de $\pm 50 \text{ mg}$, além de fatores de não linearidade que interferem e abrangem 0,5% do valor da medida final, e da sensibilidade limitada do dispositivo em relação ao que está sendo medido.

REFERÊNCIAS

- [1] InvenSense, "Mpu-6000 and mpu-6050 product specification revision 3.4," <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>, 2015, acesso em: 06 mai. 2023.
- [2] Y. Bruxel, "Sistema para análise de impacto na marcha humana," Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, 12 2010.

- [3] M. Mathie, A. Côté, N. Lovell, and B. Celler, "Use of a triaxial accelerometer in unsupervised monitoring of human movement," in *World Congress on Medical Physics and Biomedical Engineering*, Sydney, August 2003.
- [4] A. Industries, "Adafruit mpu6050," https://github.com/adafruit/Adafruit_MPU6050, 2019, acesso em: 06 mai. 2023.
- [5] S. Santos, "Esp32 with mpu-6050 accelerometer, gyroscope and temperature sensor (arduino)," <https://randomnerdtutorials.com/esp32-mpu-6050-accelerometer-gyroscope-arduino/>, 2021, acesso em: 06 mai. 2023.

IV. ANEXOS

A. Código

```

1 #include <Adafruit_MPU6050.h>
2 #include <Adafruit_Sensor.h>
3 #include <Wire.h>
4
5 Adafruit_MPU6050 mpu;
6
7 void setup(void) {
8     Serial.begin(115200);
9     while (!Serial)
10         delay(10);
11
12     Serial.println("Adafruit MPU6050 test!");
13
14     if (!mpu.begin()) {
15         Serial.println("Failed to find MPU6050 chip");
16         while (1) {
17             delay(10);
18         }
19     }
20     Serial.println("MPU6050 Found!");
21
22     mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
23     Serial.print("Accelerometer range set to: ");
24     switch (mpu.getAccelerometerRange()) {
25         case MPU6050_RANGE_2_G:
26             Serial.println("+2G");
27             break;
28         case MPU6050_RANGE_4_G:
29             Serial.println("+4G");
30             break;
31         case MPU6050_RANGE_8_G:
32             Serial.println("+8G");
33             break;
34         case MPU6050_RANGE_16_G:
35             Serial.println("+16G");
36             break;
37     }
38     mpu.setGyroRange(MPU6050_RANGE_500_DEG);
39     Serial.print("Gyro range set to: ");
40     switch (mpu.getGyroRange()) {
41         case MPU6050_RANGE_250_DEG:
42             Serial.println("+ 250 deg/s");
43             break;
44         case MPU6050_RANGE_500_DEG:
45             Serial.println("+ 500 deg/s");
46             break;
47         case MPU6050_RANGE_1000_DEG:
48             Serial.println("+ 1000 deg/s");
49             break;
50         case MPU6050_RANGE_2000_DEG:
51             Serial.println("+ 2000 deg/s");
52             break;
53     }
54
55     mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);
56     Serial.print("Filter bandwidth set to: ");
57     switch (mpu.getFilterBandwidth()) {
58         case MPU6050_BAND_260_HZ:
59             Serial.println("260 Hz");

```

```

        break;
        case MPU6050_BAND_184_HZ:
            Serial.println("184 Hz");
            break;
        case MPU6050_BAND_94_HZ:
            Serial.println("94 Hz");
            break;
        case MPU6050_BAND_44_HZ:
            Serial.println("44 Hz");
            break;
        case MPU6050_BAND_21_HZ:
            Serial.println("21 Hz");
            break;
        case MPU6050_BAND_10_HZ:
            Serial.println("10 Hz");
            break;
        case MPU6050_BAND_5_HZ:
            Serial.println("5 Hz");
            break;
    }
    Serial.println("");
    delay(100);
}

void loop() {
    sensors_event_t a, g, temp;
    mpu.getEvent(&a, &g, &temp);

    Serial.print("Aceleracao X: ");
    Serial.print(a.acceleration.x);
    Serial.print(", Y: ");
    Serial.print(a.acceleration.y);
    Serial.print(", Z: ");
    Serial.print(a.acceleration.z);
    Serial.println(" m/s^2");

    Serial.println("");
    delay(500);
}

```

Listing 1. Código para obter os dados do acelerômetro.