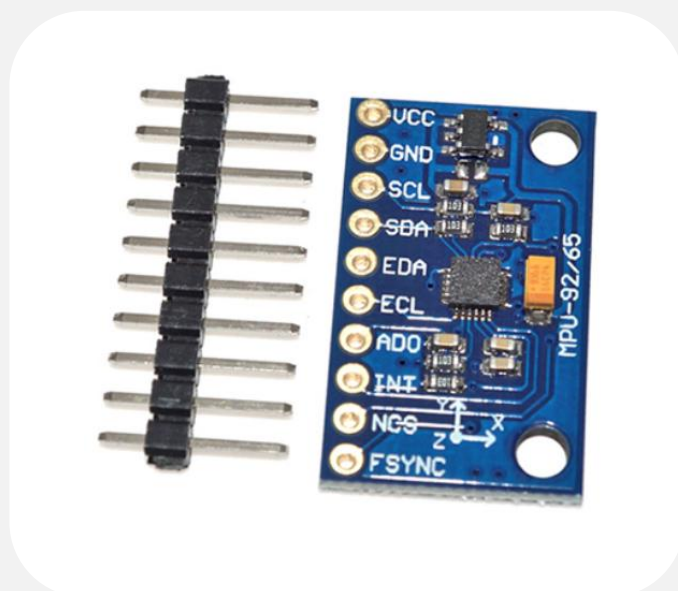


SENSOR IMU GY-9250

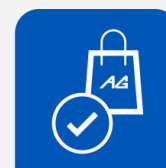
OKY3231-7



Productos
evaluados por
**ingenieros
calificados**



**Garantía y
seguridad** en
cada producto



Experiencia de
compra en la
calidad como
sello distintivo

Descripción

El OKY3231-7 es un módulo sensor IMU (de 9 ejes) que utiliza el sensor MPU-9250, ideal para proyectos educativos y aplicaciones en juguetes, robótica, drones y algunos dispositivos portátiles. Alimentado entre 3V y 5V, cuenta con un regulador LDO interno y soporta comunicación IIC/SPI. Ofrece salidas de datos de alta precisión a través de su convertidor AD de 16 bits con giroscopio, aceleración y campo magnético en rangos configurables.

Características

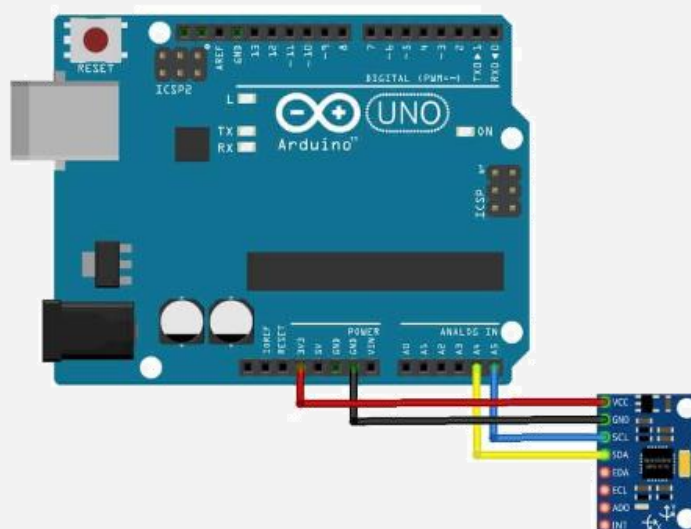
Parámetro	Descripción
Chip	MPU-9250
Alimentación	3V - 5V (regulador interno de bajo dropout)
Comunicación	I2C/SPI
Convertidor AD	16 bit
Rango de Giroscopio	± 250 , ± 500 , ± 1000 , ± 2000 °/s
Rango de aceleración	$\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$
Rango de campo magnético	$\pm 4800\mu T$
Espaciado de pines	2.54mm

Dimensiones	15mm x 25mm
Regulador LDO	3.3V
Grados de libertad (DoF)	9
Registros de interrupción SPI	20MHz

Diagrama de conexión con Arduino:

Alimentamos el módulo desde Arduino mediante GND y 5V y conectamos el pin SDA y SCL de Arduino con los pines correspondientes del sensor.

- Alimentación: 5V - VCC
- Tierra: GND - GND
- Intercambio de datos: SDA - A4
- Señal de reloj: SCL - A5



El módulo utiliza la interfaz I2C para comunicarse con la MCU. Admite dos direcciones I2C diferentes; 0x68 y 0x69. Eso permite usar dos dispositivos en el mismo bus o en caso de que haya un conflicto de dirección con otro dispositivo en el bus.

El pin ADO determina la dirección I2C a usar. Este pin tiene una resistencia pull-down de 4.7K incorporada en el módulo. Si el pin se deja desconectado, la línea se reducirá y la dirección I2C predeterminada será 0x68. Para seleccionar 0x69, conecte el pin ADO a 3.3V.

La dirección I2C auxiliar con pines etiquetados como EDA y ECL son un bus I2C controlado por el MPU-9250 para que pueda comunicarse directamente con otros sensores para que pueda obtener información adicional para sus cálculos internos.

Ejemplo de código en Arduino para realizar la lectura de los valores RAW

Los raw son las lecturas iniciales proporcionadas por el acelerómetro, giroscopio y magnetómetro. Estos valores se presentan típicamente en forma de números binarios que corresponden a las mediciones físicas de movimiento, rotación y campos magnéticos.

```
#include "Wire.h" // Librería para poder comunicarse con dispositivos por I2C
```

```
const int MPU_ADDR = 0x68; // Dirección I2C del MPU-6050. Si AD0(0x69) HIGH, de lo contrario I2C(0x69) es LOW
```

```
int16_t a_x, a_y, a_z; // Variables para el acelerómetro  
int16_t g_x, g_y, g_z; // Variables para el giroscopio  
int16_t m_x, m_y, m_z; // Variables para el magnetómetro  
char tmp_str[7]; // Variable temporal con 7 caracteres
```

```
// Función para convertir un int16 a una cadena de texto
```

```
char* convert_int16_to_str(int16_t i) {  
    sprintf(tmp_str, "%6d", i); // Convierte el número a una cadena de texto con 6 dígitos  
    return tmp_str;  
}
```

```
void setup() {  
    Serial.begin(9600); // Inicializa la comunicación serie a 9600 baudios  
    Wire.begin(); // Inicializa la comunicación I2C  
    Wire.beginTransmission(MPU_ADDR); // Comienza la comunicación con el dispositivo en la dirección I2C especificada  
    Wire.write(0x6B); // Selecciona el registro de configuración de la fuente de alimentación del MPU-6050  
    Wire.write(0); // Escribe un 0 en el registro, encendiendo el MPU-6050  
    Wire.endTransmission(true); // Termina la transmisión  
}
```

```
void loop() {  
    Wire.beginTransmission(MPU_ADDR);  
    Wire.write(0x3B); // Selecciona el registro 0x3B (ACCEL_XOUT_H) para empezar la lectura de los datos del acelerómetro  
    Wire.endTransmission(false); // Mantiene la conexión activa  
    Wire.requestFrom(MPU_ADDR, 7 * 2, true); // Solicita 14 registros desde el MPU-6050
```

```
// "Wire.read()<<8 | Wire.read();" lee dos bytes (alto y bajo) y los combina en un solo valor de 16 bits
```

```
    a_x = Wire.read() << 8 | Wire.read(); // Lectura del registro del acelerómetro: 0x3B (A_XOUT_H) y 0x3C (A_XOUT_L)
```

```
a_y = Wire.read() << 8 | Wire.read(); // Lectura del registro del acelerómetro: 0x3D
(A_YOUT_H) y 0x3E (A_YOUT_L)
```

```
a_z = Wire.read() << 8 | Wire.read(); // Lectura del registro del acelerómetro: 0x3F
(A_ZOUT_H) y 0x40 (A_ZOUT_L)
```

```
int16_t t = Wire.read() << 8 | Wire.read(); // Lectura del registro de temperatura: 0x41
(T_OUT_H) y 0x42 (T_OUT_L)
```

```
g_x = Wire.read() << 8 | Wire.read(); // Lectura del registro del giroscopio: 0x43
(G_XOUT_H) y 0x44 (GYRO_XOUT_L)
```

```
g_y = Wire.read() << 8 | Wire.read(); // Lectura del registro del giroscopio: 0x45
(G_YOUT_H) y 0x46 (GYRO_YOUT_L)
```

```
g_z = Wire.read() << 8 | Wire.read(); // Lectura del registro del giroscopio: 0x47
(G_ZOUT_H) y 0x48 (GYRO_ZOUT_L)
```

```
m_x = Wire.read() << 8 | Wire.read(); // Lectura del registro del magnetómetro
```

```
m_y = Wire.read() << 8 | Wire.read(); // Lectura del registro del magnetómetro
```

```
m_z = Wire.read() << 8 | Wire.read(); // Lectura del registro del magnetómetro
```

```
// Impresión de la información
```

```
Serial.print("aX = "); Serial.print(convert_int16_to_str(a_x));
```

```
Serial.print(" | aY = "); Serial.print(convert_int16_to_str(a_y));
```

```
Serial.print(" | aZ = "); Serial.print(convert_int16_to_str(a_z));
```

```
Serial.print(" | gX = "); Serial.print(convert_int16_to_str(g_x));
```

```
Serial.print(" | gY = "); Serial.print(convert_int16_to_str(g_y));
```

```
Serial.print(" | gZ = "); Serial.print(convert_int16_to_str(g_z));
```

```
Serial.print(" | mx = "); Serial.print(convert_int16_to_str(m_x + 200)); // Ajuste de 200
para el valor del magnetómetro en x
```

```
Serial.print(" | my = "); Serial.print(convert_int16_to_str(m_y - 70)); // Ajuste de -70
para el valor del magnetómetro en y
```

```
Serial.print(" | mz = "); Serial.print(convert_int16_to_str(m_z - 700)); // Ajuste de -700
para el valor del magnetómetro en z
```

```
Serial.println();
```

```
// Espera 1 segundo para la siguiente lectura
```

```
delay(1000);
```

```
}
```

Enlace externo: cómo utilizar el MPU-9250

Robojax. (2018, 10 julio). How to use MPU-9250 Gyroscope, Accelerometer, Magnetometer for Arduino [Video]. YouTube.
<https://www.youtube.com/watch?v=mzwovYcozvl>

AG Electrónica SAPI de CV
 República de El Salvador 20 Piso 2, Centro
 Histórico, Centro, 06000 Ciudad de México,
 CDMX
 Teléfono: 55 5130 7210

Realizó

Adrián Jesús Beltrán Cruz

Revisó

Ing. Jesús Daniel Ibarra Noguez

Fecha

03/07/2024

