

SENSOR IMU (UNIDAD DE MEDICION INERCIAL) DE BAJO CONSUMO

SKU12476



Descripción

El sensor IMU de 10 DOF es ideal para el monitoreo de movimiento, así como para la detección y medición de la posición, altura y temperatura.

Características

Cuenta con dos sensores internos MPUS9250 y BMP280

MPU9250: giroscopio de 3 ejes, acelerómetro de 3 ejes y brújula/magnetómetro de 3 ejes.

BMP280: sensor de presión barométrica.

Especificaciones:

- Voltaje: 3.3 V ~ 5.5 V (regulación de voltaje interno con baja caída de voltaje)
- Acelerómetro
 - Resolución: 16 bit
 - Rango de medición (configurable): ± 2 , ± 4 , ± 8 , ± 16 g
 - Corriente de funcionamiento: 450uA
- Giroscopio
 - Resolución: 16 bit
 - Rango de medición (configurable): ± 250 , ± 500 , ± 1000 , $\pm 2000^\circ/\text{seg}$
 - Corriente de funcionamiento: 3,2 mA
- Brújula/magnetómetro

- Resolución: 14 bits
- Rango de medición: $\pm 4800\mu\text{T}$
- Corriente de funcionamiento: 280uA
- Sensor de presión barométrica
 - Resolución barométrica: 0,0016 hPa
 - Resolución de temperatura: 0,01 °C
 - Rango de medición: 300~1100 hPa (altitud: +9000 m ~ -500 m)
 - Precisión relativa barométrica (700 hPa~900 hPa, 25 °C~40 °C): $\pm 0,12$ hPa (± 1 m)
 - Corriente de funcionamiento (frecuencia de actualización de 1 Hz, modo de consumo ultrabaja): 2,8 uA

Definición de pines

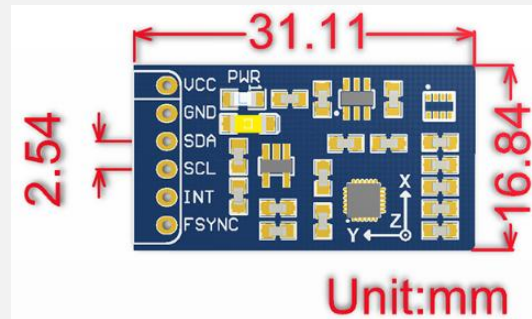


PIN	SÍMBOLO	DESCRIPCIÓN
1	VCC	Fuente de alimentación de 3.3 V ~ 5.5 V
2	GND	Tierra
3	SDA	Pin de datos I2C
4	SCL	Pin de reloj I2C
5	INT	Salida de interrupción digital MPU9250
6	FSYNC	Señal de sincronización de cuadros MPU9250

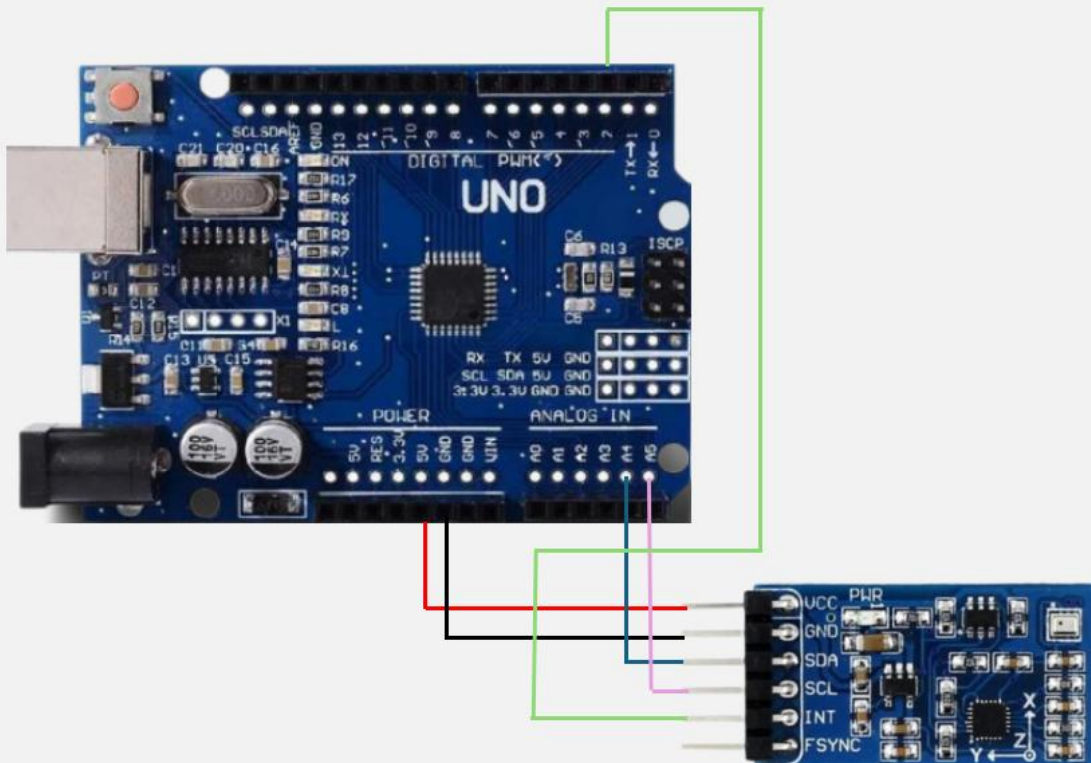
Aplicaciones

- Drones (Quadcopter)
- Robots de equilibrio
- Navegación inercial en interiores
- Altímetro barométrico

Dimensiones



Ejemplo de conexión



Conexiones

Para conectar los sensores al Arduino Uno:

- **VCC** -> **5V**
- **GND** -> **GND**
- **SCL** -> **A5** (Pin SCL en Arduino Uno) (comparte con el MPU9250 y BMP280)
- **SDA** -> **A4** (Pin SCL en Arduino Uno) (comparte con el MPU9250 y BMP280)
- **INT** -> **Pin 2** (opcional, si usas interrupciones)

- **SYNC** -> Puedes dejarlo desconectado o usar un pin digital como el **Pin 4** si lo necesitas, podrías conectar el SYNC a un pin digital de tu elección, por ejemplo pin 4, y gestionar las señales manualmente en tu código.

Código

Primero, instala la librería **Hideakitai MPU9250** y la librería **Adafruit BMP280** si no las tienes.

1. En el **IDE de Arduino**, ve a **Herramientas -> Gestionar bibliotecas**.
2. Busca e instala:
 1. **MPU9250 by hideakitai**
 2. **Adafruit BMP280**



```
#include <Wire.h>
#include <MPU9250.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>

// Instancia para el sensor BMP280
Adafruit_BMP280 bmp;
// Instancia para el MPU9250
MPU9250 mpu;
// Configuraciones de pines (ajustar si usas otros pines)
#define BMP280_ADDR 0x76 // Dirección I2C del BMP280
void setup() {
  Serial.begin(115200);
  Wire.begin();
  // Inicializar MPU9250
  if (!mpu.setup(0x68)) { // Dirección I2C del MPU9250
    Serial.println("Error inicializando MPU9250.");
    while (1);
  }
  Serial.println("MPU9250 inicializado correctamente.");
```

```
// Inicializar BMP280
if (!bmp.begin(0x76)) {
  Serial.println("Error inicializando BMP280/BME280 en 0x76.");
  if (!bmp.begin(0x77)) {
    Serial.println("Error inicializando BMP280/BME280 en 0x77.");
    while (1);
  }
}
Serial.println("BMP280 inicializado correctamente.");

// Configurar BMP280
bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,
  Adafruit_BMP280::SAMPLING_X2, // Oversampling de temperatura
  Adafruit_BMP280::SAMPLING_X16, // Oversampling de presión
  Adafruit_BMP280::FILTER_X16, // Filtro
  Adafruit_BMP280::STANDBY_MS_500);
}

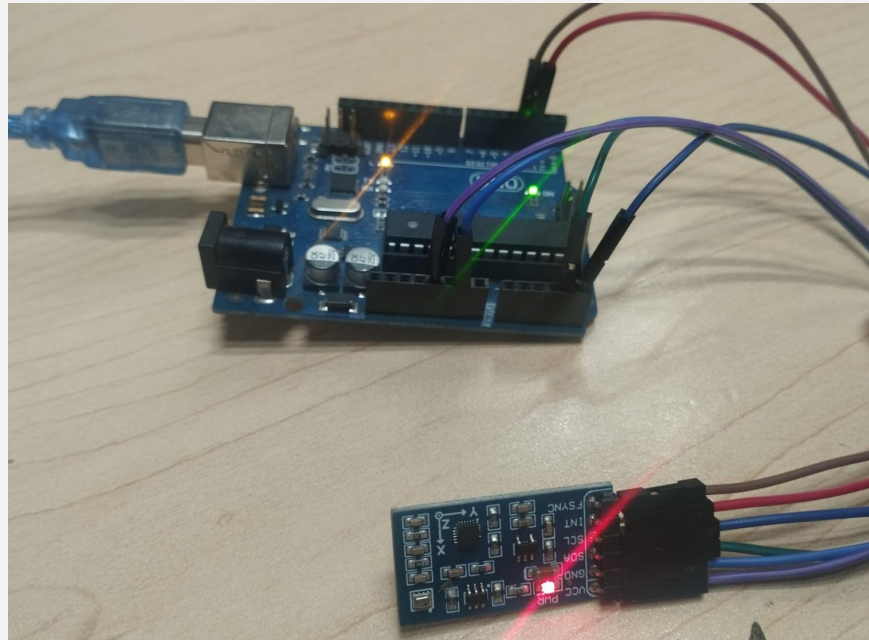
void loop() {
  // Leer acelerómetro, giroscopio y magnetómetro del MPU9250
  if (mpu.update()) {
    Serial.print("Acelerómetro [g]: X = ");
    Serial.print(mpu.getAccX());
    Serial.print(" Y = ");
    Serial.print(mpu.getAccY());
    Serial.print(" Z = ");
    Serial.println(mpu.getAccZ());

    Serial.print("Giroscopio [dps]: X = ");
    Serial.print(mpu.getGyroX());
    Serial.print(" Y = ");
    Serial.print(mpu.getGyroY());
    Serial.print(" Z = ");
    Serial.println(mpu.getGyroZ());

    // Mantener las lecturas del magnetómetro originales
    Serial.print("Magnetómetro: X = ");
    Serial.print(mpu.getMagX());
    Serial.print(" Y = ");
    Serial.print(mpu.getMagY());
    Serial.print(" Z = ");
    Serial.println(mpu.getMagZ());
  } else {
    Serial.println("Error al actualizar los datos del MPU9250.");
  }
  // Leer presión y temperatura del BMP280
  Serial.print("Presión: ");
  Serial.print(bmp.readPressure());
  Serial.println(" Pa");

  Serial.print("Temperatura: ");
  Serial.print(bmp.readTemperature());
  Serial.println(" °C");
  delay(1000); // Pausa de 1 segundo antes de la siguiente lectura
}
```


Resultados



```

bme280.ino
1  #include <Wire.h>
2  #include <Adafruit_Sensor.h>
3  #include <Adafruit_BMP280.h> // O usa Adafruit_BME280 si es BME280
4
5  Adafruit_BMP280 bmp; // Usa Adafruit_BME280 si es un BME280
6
7  void setup() {
8      Serial.begin(115200);
9      Wire.begin();
10
11     if (!bmp.begin(0x76)) {
12         Serial.println("Error inicializando BMP280/BME280 en 0x76.");
13         if (!bmp.begin(0x77)) {
14             Serial.println("Error inicializando BMP280/BME280 en 0x77.");
15             while (1);
16         }
17     }
18     Serial.println("BMP280/BME280 inicializado correctamente.");
19 }
20
21 void loop() {
22     Serial.print("Presión: ");
23     Serial.print(bmp.readPressure());
24     Serial.println(" Pa");
25
26     Serial.print("Temperatura: ");

```

Salida Monitor Serie x

Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM5') Nueva línea ▾

```

Temperatura: 26.17 °C
Presión: 77980.73 Pa
Temperatura: 26.17 °C
Presión: 77980.32 Pa
Temperatura: 26.18 °C
Presión: 77980.80 Pa
Temperatura: 26.19 °C
Presión: 77982.20 Pa
Temperatura: 26.20 °C
Presión: 77982.28 Pa
Temperatura: 26.22 °C
Presión: 77984.39 Pa
Temperatura: 26.21 °C

```

```

74 }
75

Salida Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM5')
18:15:16.489 -> MPU9250 inicializado correctamente.
18:15:16.489 -> Error inicializando BMP280.

```

```

69 Serial.print("Presion: ");
70 Serial.print(bmp.readPressure());
71 Serial.println(" Pa");

Salida Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM5') Nueva línea 11
17:19:50.362 -> Giroscopio [dps]: X = -2.08 Y = 1.07 Z = -0.73
17:19:50.362 -> Magnetómetro [uT]: X = 0.00 Y = Z = 0.00
17:19:50.362 -> Presión: 78041.36 Pa
17:19:50.362 -> Temperatura: 27.48 °C
17:19:51.376 -> Acelerómetro [g]: X = -0.79 Y = 0.46 Z = 0.39
17:19:51.376 -> Giroscopio [dps]: X = -2.08 Y = 0.79 Z = -0.79
17:19:51.376 -> Magnetómetro [uT]: X = 0.00 Y = Z = 0.00
17:19:51.376 -> Presión: 78041.25 Pa
17:19:51.376 -> Temperatura: 27.54 °C
17:19:52.394 -> Acelerómetro [g]: X = -0.79 Y = 0.46 Z = 0.39
17:19:52.394 -> Giroscopio [dps]: X = -2.20 Y = 1.10 Z = -0.85
17:19:52.394 -> Magnetómetro [uT]: X = 0.00 Y = Z = 0.00
17:19:52.394 -> Presión: 78040.53 Pa
17:19:52.394 -> Temperatura: 27.60 °C
17:19:53.401 -> Acelerómetro [g]: X = -0.79 Y = 0.46 Z = 0.39
17:19:53.401 -> Giroscopio [dps]: X = -2.08 Y = 0.49 Z = -0.61
17:19:53.401 -> Magnetómetro [uT]: X = 0.00 Y = Z = 0.00
17:19:53.401 -> Presión: 78040.44 Pa
17:19:53.401 -> Temperatura: 27.65 °C

Lín. 63, col. 14 Arduino Uno en
Carretera cerrada en... ESP

```

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	Alan Huerta Zavala
Revisó	Ing. Jessica López Morales
Fecha	07/11/2024

