



ARDUINO 10DOF SENSOR MODULE GY-87

The **GY-87** is a popular 10DOF (Degrees of Freedom) sensor module that combines multiple sensors into a single board, making it ideal for projects requiring motion tracking, orientation sensing, and environmental monitoring. It is commonly used with microcontrollers like Arduino, ESP32, or Raspberry Pi.

Key Features of the GY-87 Module

1. **MPU6050:**
 - 6-axis Motion Tracking Sensor (3-axis gyroscope + 3-axis accelerometer).
 - Provides data for orientation, motion, and vibration.
 - Includes a Digital Motion Processor (DMP) for advanced calculations.
 2. **HMC5883L:**
 - 3-axis Magnetometer (compass).
 - Measures magnetic fields to determine heading or direction.
 3. **BMP180:**
 - Barometric Pressure Sensor.
 - Measures atmospheric pressure, altitude, and temperature.
 4. **10DOF Capability:**
 - Combines data from the gyroscope, accelerometer, magnetometer, and barometric sensor for 10 degrees of freedom.
-

Pinout of the GY-87 Module

The GY-87 module typically has the following pins:

- **VCC:** 3.3V or 5V power input.
- **GND:** Ground.
- **SCL:** I2C clock line.
- **SDA:** I2C data line.



ARDUINO 10DOF SENSOR MODULE GY-87

- **XDA**: Auxiliary I2C data line (for external sensors).
 - **XCL**: Auxiliary I2C clock line (for external sensors).
 - **AD0**: Address select pin for MPU6050.
 - **INT**: Interrupt pin for MPU6050.
-

Connecting GY-87 to Arduino

Here's how to connect the GY-87 to an Arduino:

GY-87 Pin Arduino Pin

VCC	3.3V or 5V
GND	GND
SCL	A5 (or SCL)
SDA	A4 (or SDA)

Note: The GY-87 can operate at both 3.3V and 5V logic levels, but ensure your Arduino and GY-87 are compatible.

Arduino Code for GY-87

To read data from the GY-87, you'll need libraries for the MPU6050, HMC5883L, and BMP180. Here's an example:

1. Install the required libraries:
 - **MPU6050**: Use the MPU6050_tockn or I2Cdev library.
 - **HMC5883L**: Use the Adafruit_HMC5883_Unified library.
 - **BMP180**: Use the Adafruit_BMP085 library.
2. Example Code:

```
cpp
Copy
#include <Wire.h>
#include <MPU6050_tockn.h>
```



ARDUINO 10DOF SENSOR MODULE GY-87

```
#include <Adafruit_HMC5883_U.h>
#include <Adafruit_BMP085.h>

MPU6050 mpu6050(Wire);
Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified(12345);
Adafruit_BMP085 bmp;

void setup() {
  Serial.begin(9600);

  // Initialize MPU6050
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);

  // Initialize HMC5883L
  if (!mag.begin()) {
    Serial.println("Could not find a valid HMC5883 sensor!");
    while (1);
  }

  // Initialize BMP180
  if (!bmp.begin()) {
    Serial.println("Could not find a valid BMP180 sensor!");
    while (1);
  }
}

void loop() {
  // Read MPU6050 data
  mpu6050.update();
  Serial.print("Accel: ");
  Serial.print(mpu6050.getAccX()); Serial.print(", ");
  Serial.print(mpu6050.getAccY()); Serial.print(", ");
  Serial.print(mpu6050.getAccZ()); Serial.println();

  Serial.print("Gyro: ");
  Serial.print(mpu6050.getGyroX()); Serial.print(", ");
  Serial.print(mpu6050.getGyroY()); Serial.print(", ");
  Serial.print(mpu6050.getGyroZ()); Serial.println();

  // Read HMC5883L data
  sensors_event_t event;
  mag.getEvent(&event);
  Serial.print("Mag: ");
  Serial.print(event.magnetic.x); Serial.print(", ");
  Serial.print(event.magnetic.y); Serial.print(", ");
```

ARDUINO 10DOF SENSOR MODULE GY-87

```
Serial.print(event.magnetic.z); Serial.println();

// Read BMP180 data
Serial.print("Temp: "); Serial.print(bmp.readTemperature()); Serial.println(" *C");
Serial.print("Pressure: "); Serial.print(bmp.readPressure()); Serial.println(" Pa");
Serial.print("Altitude: "); Serial.print(bmp.readAltitude()); Serial.println(" m");

delay(1000);
}
```

Applications of the GY-87

- **Drones and UAVs:** For flight stabilization and navigation.
- **Robotics:** For orientation and motion tracking.
- **Weather Stations:** For altitude and pressure measurements.
- **Virtual Reality (VR):** For head tracking and motion sensing.

Tips for Using the GY-87

1. **Calibration:** Calibrate the MPU6050 and HMC5883L for accurate readings.
2. **I2C Addresses:** Ensure there are no address conflicts with other I2C devices.
3. **Power Supply:** Use a stable power source to avoid noise in sensor readings.

