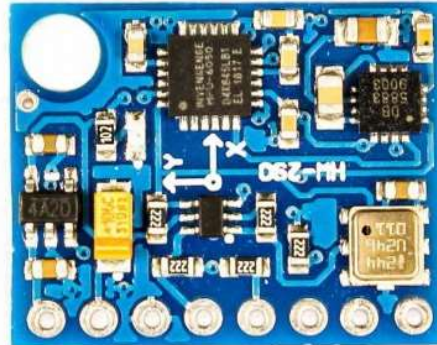


GY-87 10DOF MPU6050 HMC5883L BMP180



The GY-87 is a popular sensor module that integrates multiple sensors on a single board. It includes:

1. **MPU6050:** A 6-axis gyroscope and accelerometer. This sensor measures acceleration and rotation in three dimensions, which is useful for applications involving motion detection and orientation.
2. **HMC5883L:** A 3-axis digital magnetometer (or compass). This sensor measures the magnetic field strength in three dimensions, allowing for magnetic field detection and heading calculation.
3. **BMP180:** A barometric pressure sensor. This measures atmospheric pressure and can be used to estimate altitude or for weather monitoring.

Here's a brief overview of how each sensor is typically used:

- **MPU6050:** Often used for inertial measurement units (IMUs) in robotics, drones, and motion tracking systems. It provides data on acceleration and angular velocity, which can be used to determine orientation and movement.
- **HMC5883L:** Used for compass applications to determine the direction relative to the Earth's magnetic field. It can be used in navigation systems and for orientation purposes in various devices.
- **BMP180:** Commonly used in weather stations and altitude measurement systems. It can provide data on air pressure, which can be converted into altitude or used for weather forecasting.

Interfacing with the GY-87

1. **Connection:** The GY-87 board typically uses I2C communication, so you'll connect it to your microcontroller (like an Arduino or ESP32) using the SDA (data line) and SCL (clock line) pins, along with power and ground.
2. **Libraries and Code:** You'll need libraries for each sensor to read data. For Arduino, libraries like MPU6050, HMC5883L, and BMP180 are available and make interfacing easier.
3. **Calibration:** Calibration might be necessary for accurate readings. For example, the MPU6050 might require calibration to correct for biases in the gyroscope and accelerometer readings.
4. **Data Fusion:** If you're using all sensors together, you may need to fuse the data for more accurate results. For example, combining accelerometer and magnetometer data to get a more accurate heading or combining barometric pressure with altitude data.

Would you like specific code examples or further details on interfacing with this module? See next Page:

To get started with the GY-87 sensor module using an Arduino, you'll need to install and use libraries for each sensor: MPU6050, HMC5883L, and BMP180. Below, I'll provide a basic example of code that initializes each sensor and reads some data from them.

Required Libraries

1. **MPU6050:** You can use the MPU6050 library by Jeff Rowberg.
2. **HMC5883L:** The HMC5883L library by Andrew S. or Adafruit_HMC5883_U from Adafruit.
3. **BMP180:** The Adafruit BMP085 Unified library by Adafruit.

You can install these libraries via the Arduino Library Manager or download them from their respective GitHub repositories.

Sample Code

```
cpp
Copy code
#include <Wire.h>
#include <MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
#include <Adafruit_HMC5883_U.h>

// Create instances of each sensor class
MPU6050 mpu;
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified();
Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified();

// Variables to store sensor data
float temperature, pressure, altitude;
Vector magData;

void setup() {
  // Start serial communication
  Serial.begin(9600);

  // Initialize MPU6050
  Wire.begin();
  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed");
    while (1);
  }
  Serial.println("MPU6050 initialized");

  // Initialize BMP180
  if (!bmp.begin()) {
    Serial.println("BMP180 initialization failed");
    while (1);
  }
  Serial.println("BMP180 initialized");

  // Initialize HMC5883L
  if (!mag.begin()) {
    Serial.println("HMC5883L initialization failed");
    while (1);
  }
  Serial.println("HMC5883L initialized");
}
```

```

}

void loop() {
  // Read MPU6050 data
  int16_t ax, ay, az;
  int16_t gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  // Convert to g (acceleration) and degrees per second (gyroscope)
  float ax_g = ax / 16384.0;
  float ay_g = ay / 16384.0;
  float az_g = az / 16384.0;
  float gx_dps = gx / 131.0;
  float gy_dps = gy / 131.0;
  float gz_dps = gz / 131.0;

  Serial.print("MPU6050: ");
  Serial.print("Ax: "); Serial.print(ax_g); Serial.print(" g ");
  Serial.print("Ay: "); Serial.print(ay_g); Serial.print(" g ");
  Serial.print("Az: "); Serial.print(az_g); Serial.print(" g ");
  Serial.print("Gx: "); Serial.print(gx_dps); Serial.print(" dps ");
  Serial.print("Gy: "); Serial.print(gy_dps); Serial.print(" dps ");
  Serial.print("Gz: "); Serial.print(gz_dps); Serial.println(" dps");

  // Read BMP180 data
  sensors_event_t event;
  bmp.getEvent(&event);
  if (event.pressure) {
    pressure = event.pressure;
    bmp.getTemperature(&temperature);
    altitude = bmp.pressureToAltitude(pressure);

    Serial.print("BMP180: ");
    Serial.print("Temperature: "); Serial.print(temperature); Serial.print(" C ");
    Serial.print("Pressure: "); Serial.print(pressure); Serial.print(" hPa ");
    Serial.print("Altitude: "); Serial.print(altitude); Serial.println(" m");
  } else {
    Serial.println("BMP180: Error reading pressure data");
  }

  // Read HMC5883L data
  if (mag.getEvent(&magData)) {
    Serial.print("HMC5883L: ");
    Serial.print("X: "); Serial.print(magData.x); Serial.print(" uT ");
    Serial.print("Y: "); Serial.print(magData.y); Serial.print(" uT ");
    Serial.print("Z: "); Serial.print(magData.z); Serial.println(" uT");
  } else {
    Serial.println("HMC5883L: Error reading magnetometer data");
  }

  // Wait a bit before the next loop
  delay(500);
}

```

Key Points:

1. **Initialization:** Each sensor is initialized in the `setup()` function. Check if each sensor is connected and initialized properly.
2. **Reading Data:** Data is read from each sensor in the `loop()` function and printed to the serial monitor. Data from the MPU6050 is converted to g-forces and degrees per second. The BMP180 data is converted to temperature, pressure, and altitude. The HMC5883L provides magnetometer readings in microteslas (uT).
3. **Libraries:** Ensure you have the necessary libraries installed. You can manage libraries through the Arduino IDE by going to **Sketch > Include Library > Manage Libraries**.

This basic example should help you get started with integrating and reading data from the GY-87 sensor module. Adjust and expand the code based on your specific project needs.