Code example: LSM6DSO32 accelerometer and gyroscope

Mercury V1 manual · v1.0

https://www.altimetercloud.com/support/mercuryv1/code-accelerometer-gyro/

This example shows how you can connect to the LSM6DSO32 IMU (Accelerometer & Gyroscope) and read the values. You need to ensure you turn on the sensors using the VACC pin as per this code.

Once programmed, press the power button to restart the Mercury then connect a Serial monitor.

> **Using Arduino IDE?** Our online programmer includes Mercury_Pins.h by default so the pin names work without issue. If you are using Arduino IDE or another programmer, copy the Mercury_Pins.h tab content and paste it into the top of your program.

```
/*
 * Mercury LSM6DSO32 Example
 * Reads accelerometer and gyroscope data from the LSM6DSO32 IMU
 * and outputs to Serial every 500ms.
 *
 * Wiring: Connect LSM6DSO32 to the I2C port (SDA/SCL)
 * I2C address: 0x6B (SDO/SA0 HIGH on Mercury board)
 *
 * Library: Adafruit LSM6DS (auto-installed by compiler)
 */

#include "Wire.h"
#include "Adafruit_LSM6DS.h"
#include "Adafruit_LSM6DSO32.h"
#include "Mercury_Pins.h"

Adafruit_LSM6DSO32 imu;

void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("LSM6DSO32 IMU Example");
  Serial.println("====================");

  // Power on the sensor rail
  pinMode(VACC, OUTPUT);
  digitalWrite(VACC, HIGH);
  delay(100);

  // Start I2C on Mercury pins
  Wire.begin(SDA, SCL);

  // Initialise the IMU at address 0x6B
  if (!imu.begin_I2C(0x6B)) {
    Serial.println("ERROR: LSM6DSO32 not found at 0x6B!");
    Serial.println("Run the I2C Scanner example to check your address.");
    while (1) delay(100);
  }
```

```
    Serial.println("LSM6DSO32 connected at 0x6B");

    // —— Accelerometer setup ——
    // Range options: 4G, 8G, 16G, 32G
    imu.setAccelRange(LSM6DSO32_ACCEL_RANGE_16_G);

    // Data rate options: 12.5, 26, 52, 104, 208, 416, 833, 1660, 3330, 6660 Hz
    imu.setAccelDataRate(LSM6DS_RATE_104_HZ);

    // —— Gyroscope setup ——
    // Range options: 125, 250, 500, 1000, 2000 degrees per second
    imu.setGyroRange(LSM6DS_GYRO_RANGE_250_DPS);

    // Data rate (same options as accelerometer)
    imu.setGyroDataRate(LSM6DS_RATE_104_HZ);

    Serial.println("Accel: 16G range | Gyro: 250 dps range | Rate: 104Hz");
    Serial.println();
}

void loop() {
    // Read all sensor data in one call
    sensors_event_t accel, gyro, temp;
    imu.getEvent(&accel, &gyro, &temp);

    // Accelerometer (m/s^2)
    Serial.printf("Accelerometer (X/Y/Z) m/s2: %+7.2f  %+7.2f  %+7.2f",
        accel.acceleration.x,
        accel.acceleration.y,
        accel.acceleration.z);

    // Separator
    Serial.print("    |    ");

    // Gyroscope (degrees per second)
    Serial.printf("Gyro (X/Y/Z) dps: %+7.2f  %+7.2f  %+7.2f\n",
        gyro.gyro.x,
        gyro.gyro.y,
        gyro.gyro.z);

    delay(500);
}

#pragma once
/*
 * Mercury (ESP32-C6) Pin Definitions
 * Board-specific GPIO assignments
 */

// —— Status LED (NeoPixel) ——
#define LEDPOWER      3    // NeoPixel power (drive HIGH to enable)
#define LED           2    // NeoPixel data signal

// —— I2C Bus ——
#define SDA           21   // I2C data
#define SCL           22   // I2C clock

// —— Sensor Power ——
```

```
#define VACC         20   // Sensor power rail (drive HIGH to enable)

// —— General Purpose Ports ——
#define GP06         6    // GP06 port
#define GP07         7    // GP07 port

// —— High Current Output ——
#define OUT1         5    // High current output (e.g. pyro / relay)

// —— Battery Bar LEDs ——
#define BL1          4    // Battery LED 1 (lowest)
#define BL2          14   // Battery LED 2
#define BL3          15   // Battery LED 3
#define BL4          18   // Battery LED 4
#define BL5          19   // Battery LED 5 (highest)

// —— Indicators ——
#define DISK         8    // Disk activity LED

// —— Analogue / Detection ——
#define BATIN        0    // Battery voltage (1:1 divider)
#define USBDETECT    1    // USB power detect (HIGH = USB present)
#define BUTTON       9    // BUTTON on the board, boot button but can be used
```