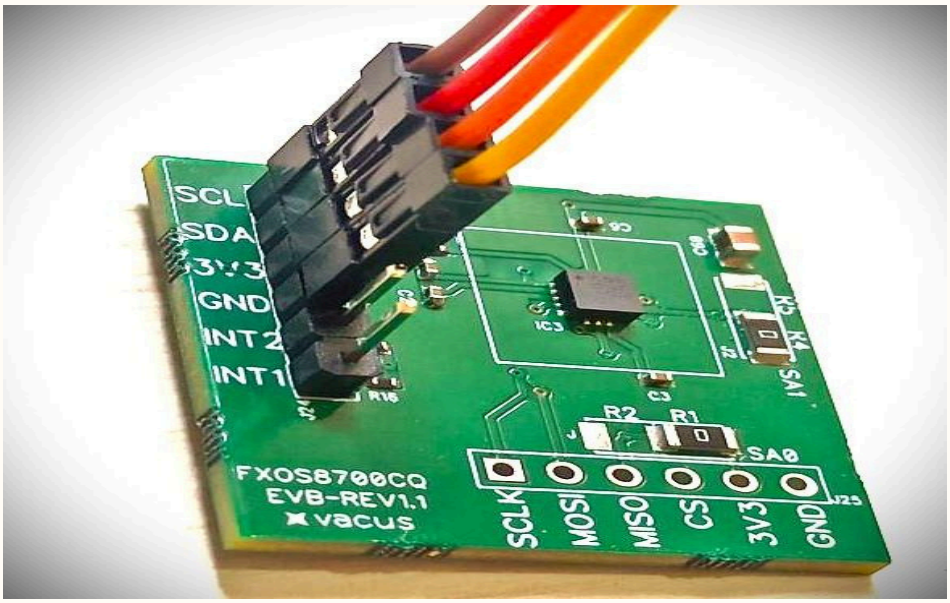
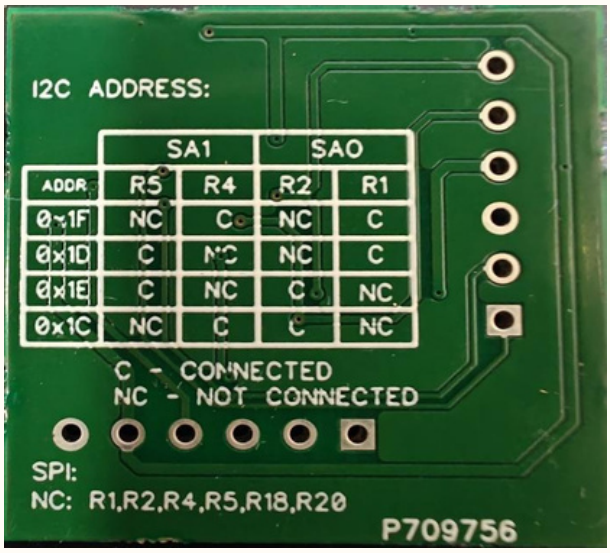
FXOS8700CQ 6-Axis Sensor evaluation board Integration & Setup Guide

The FXOS8700CQ is a 6-axis sensor combining a 3-axis accelerometer and a 3-axis magnetometer





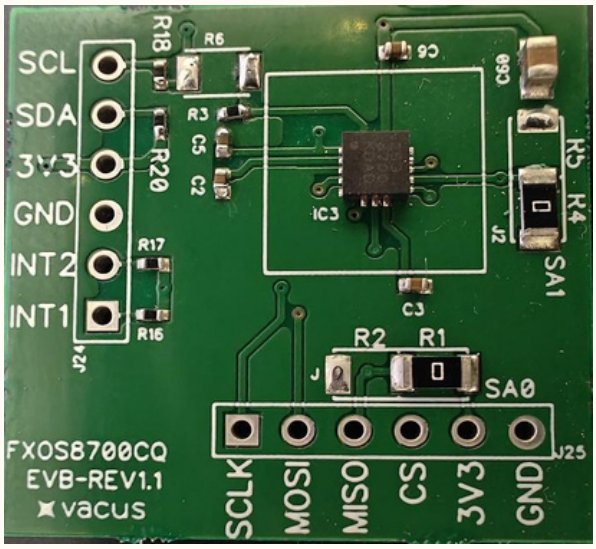


Fig : Image of FXOS8700CQ sensor module

Table of Content

1. Overview

The **FXOS8700CQ** is a **6-axis sensor** from NXP combining a 3-axis accelerometer and 3-axis magnetometer. It supports I²C and SPI, but here we use **I²C**.

1.1 General description

FXOS8700CQ is a small, low-power, 3-axis linear accelerometer and 3-axis m agnetometer combined into a single package. The device features a selectable I 2C or *point-to-point* SPI serial interface with 14-bit accelerometer and 16-bit m agnetometer ADC resolution along with smart-embedded functions. F XOS8700CQ has dynamically selectable acceleration full-scale ranges of ±2 *g*/±4 *g* /±8 *g* and a fixed magnetic measurement range of ±1200 µT. Output data rates (ODR) from 1.563 Hz to 800 Hz are selectable by the user for each sensor. I nterleaved magnetic and acceleration data is available at ODR rates of up to 400 H z. FXOS8700CQ is available in a plastic QFN package, and it is guaranteed to operate over the extended temperature range of –40 °C to +85 °C.

1.2 Features and benefits

- Complete 6-axis, e-compass hardware solution.
- 1.95 V to 3.6 V VDD supply voltage, 1.62 V to 3.6 V VDDIO voltage.
- ±2 *g*/±4 *g*/±8 *g* dynamically selectable acceleration full-scale range.
- ±1200 µT magnetic sensor full-scale range.
- Output data rates (ODR) from 1.563 Hz to 800 Hz for each sensor and up to 400 Hz when operated in hybrid mode with both sensors active.
- Low noise: < 126 µg/√Hz acceleration noise density at 200-Hz bandwidth, < 100 nT/√Hz magnetic noise density at 100-Hz bandwidth.
- 14-bit ADC resolution for acceleration measurements.
- 16-bit ADC resolution for magnetic measurements.
- Low power: 240 µA current consumption at 100 Hz and 80 µA at 25 Hz with both sensors active.
- Embedded programmable acceleration event functions.
- Freefall and motion detection.
- Transient detection.
- Vector-magnitude change detection.
- Pulse and tap detection (single and double).
- Orientation detection (portrait/landscape).
- Embedded programmable magnetic event functions.
- Threshold detection.
- Vector-magnitude change detection.
- Autonomous magnetic min/max detection.
- Autonomous hard-iron calibration.
- Programmable automatic ODR change using auto-wake and return-to-sleep functions to save power. This function works with both magnetic and acceleration event interrupt sources.
- 32-sample FIFO for acceleration data only.
- Integrated accelerometer self-test function.

1.3 Applications

- Security: motion detection, door opening, smart home applications, robotics, and unmanned aerial vehicles (UAVs) with electronic compass (e-compass) function.
- Medical applications: patient monitoring, fall detection, and rehabilitation.
- E-compass in mobile devices, tablets, and personal navigation devices.
- User interface (menu scrolling by orientation change, tap detection for button replacement).
- Orientation detection (portrait/landscape: up/down, left/right, back/front orientation identification).
- Augmented reality (AR), gaming, and real-time activity analysis (pedometry, freefall, and drop detection for hard disk drives and other devices).
- Power management for mobile devices using inertial and magnetic event detection.
- Wearable devices: motion detection, activity monitoring, sports monitoring, context awareness, and shock and vibration monitoring (mechatronic compensation, shipping, and warranty usage logging).

2. Hardware Requirements

2.1 ESP32 development board
2.2 FXOS8700CQ sensor module

3. Wiring Diagram

3.1 For I2C Communication:-

FXOS8700CQ Sensor Board Pin	ESP32 Pin	Description
VDD	3.3V	Power supply
GND	GND	Ground
SDA	GPIO21	I^2C Data
SCL	GPIO22	I^2C Clock
SA0(1)	VCC	Address bit 0
SA1(1)	VCC	Address bit 1
INT1/INT2	OPTIONAL	Interrupt Output

Table - 01

If you want to change the I2C Address, please follow the below Table - 02

	SA1		SA0	
ADDR	R5	R4	R2	R1
0x1F	NC	C	NC	C
0x1D	C	NC	NC	C
0x1E	C	NC	C	NC
0x1C	NC	C	C	NC

Table - 02

C- Connected, NC- Not Connected

3.2 For SPI Communication:

Please remove all the resistors mentioned below to enable SPI communication properly
NC – R1, R2, R4, R5, R18, R20.

4. Software Requirements

4.1 Arduino IDE
4.2 ESP32 board support installed (via Boards Manager)
4.3 Library: Adafruit FXOS8700

*(Install via Library Manager: search **"Adafruit FXOS8700"**)*

5. How to Install the Adafruit FXOS8700 Library in Arduino

5.1 Follow these simple steps to install the Adafruit FXOS8700 library in the Arduino IDE:

Step 1: Open the Arduino IDE

- Launch the Arduino software on your computer.

Step 2: Open the Library Manager

- Click on the **Tools menu** in the top navigation bar.
- Select **Manage Libraries**... from the dropdown.
  You'll see the Library Manager window open (as shown in the image - 01).

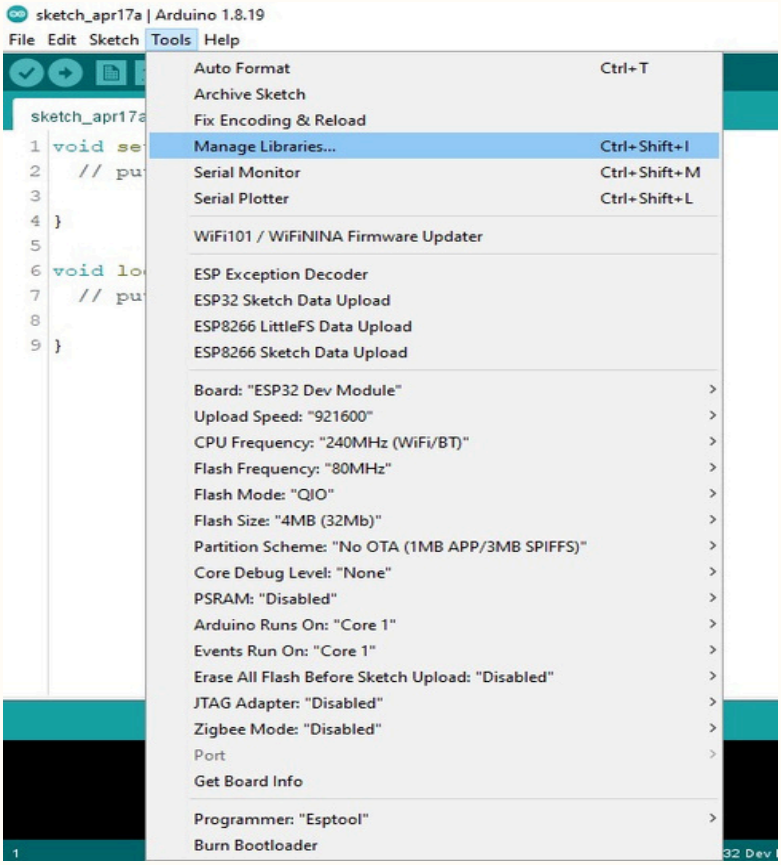


Image - 01

Step 3: Search for the FXOS8700 Library

- In the Library Manager's search bar (top right), type **FXOS8700**.
- Wait for the list to populate (see example in the image - 02).

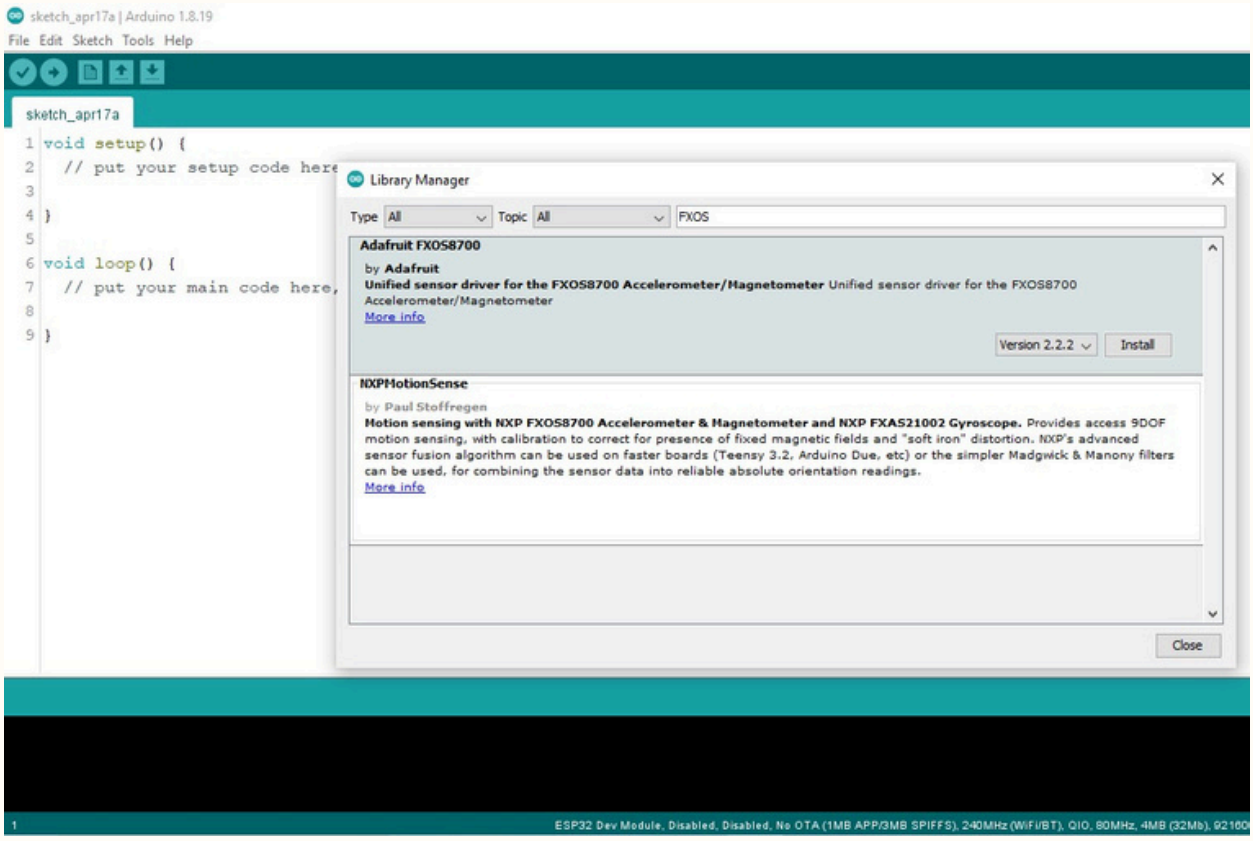


Image - 02

Step 4: Install the Adafruit FXOS8700 Library

- Find the **Adafruit FXOS8700** library in the search results.
- Click the **Install** button next to it.

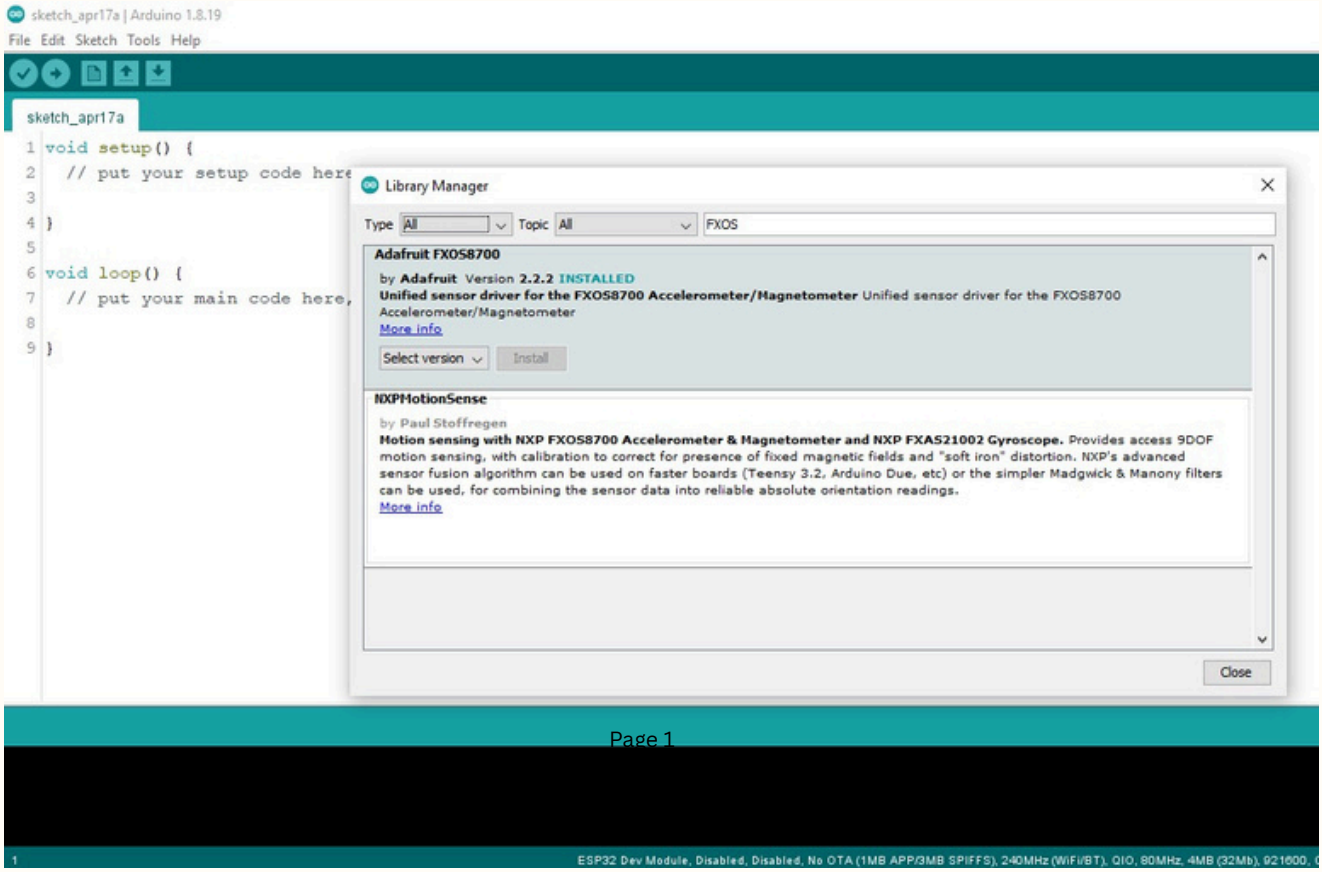


Image - 03

Once installed, the library will be ready to use in your Arduino projects!

6. Sample Arduino Code

6.1 How to Use Example Code for the Adafruit FXOS8700 in Arduino

After installing the Adafruit **FXOS8700 library**, follow these steps to load and run the example code:

Step 1: Open the Example Code

- In the Arduino IDE, go to the **File menu**.
- Hover over **Examples** in the dropdown.
- Scroll through the list and find **Adafruit FXOS8700**.
- Click on the **sensorapi** example (as shown in the image - 04).

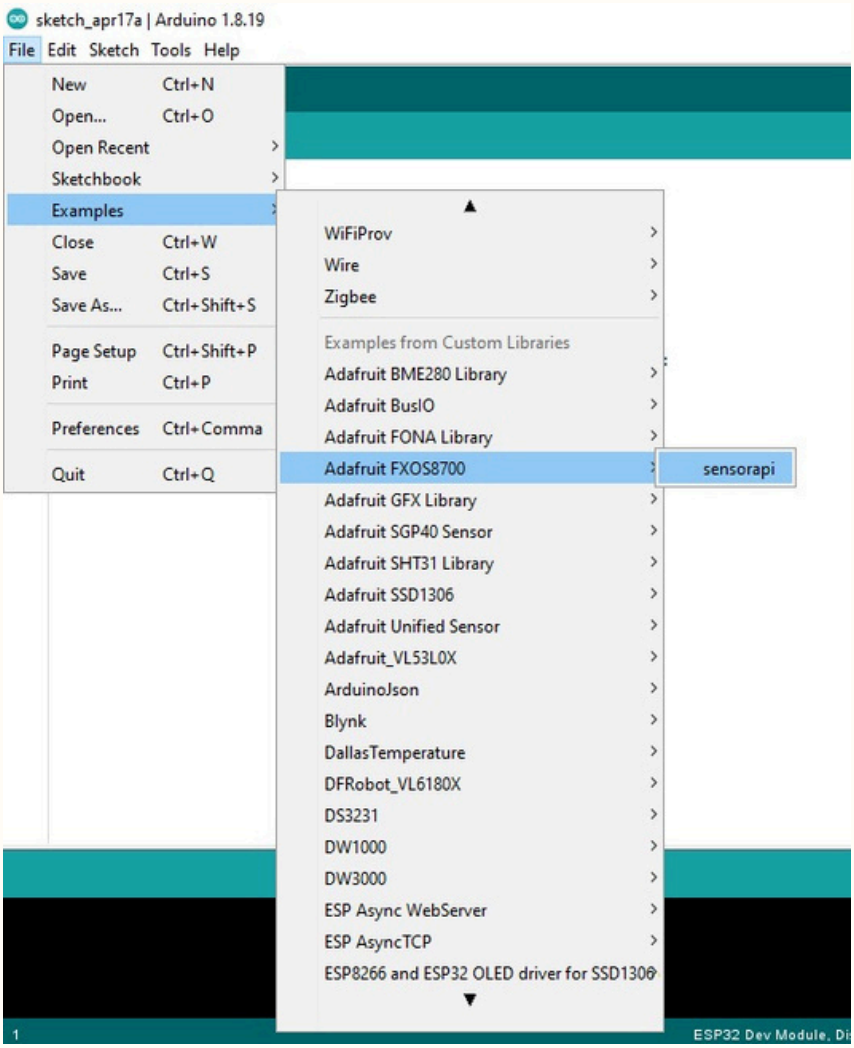


Image - 04

Image - 05

Note: Depending on your library version, the example name or folder structure may vary slightly.

Step 2: Connect the Hardware

- Plug your **ESP32 module** into your computer via USB.
- Make sure your **FXOS8700CQ sensor** is wired to the ESP32 correctly according to the connection table (you can include or refer to this table here).

Step 3: Select the Correct Port and Board

- In the Arduino IDE, go to **Tools > Board** and select your specific ESP32 board.
- Then go to **Tools > Port** and choose the correct **COM port** for your ESP32 (as shown in the image - 06).

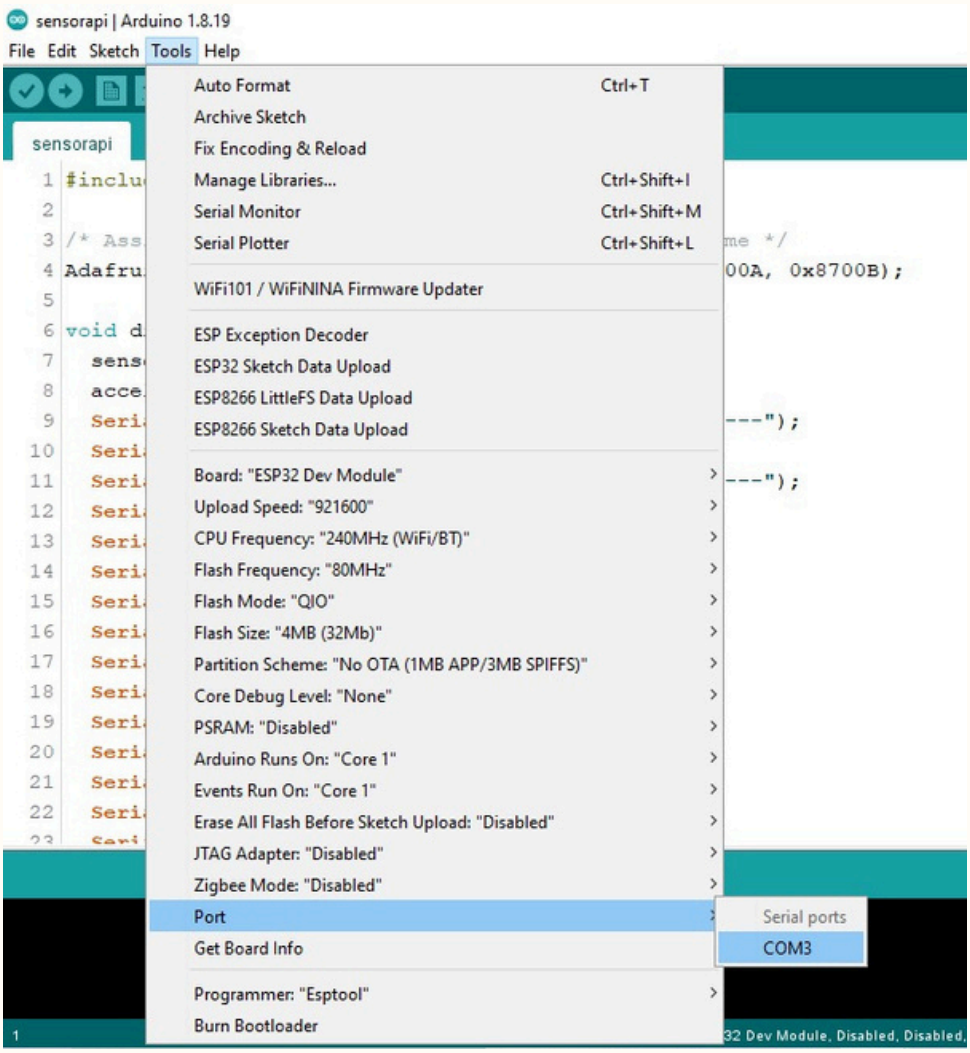


Image - 06

Step 4: Upload the Code

- Click the **Upload button** in the Arduino IDE (the right arrow icon).
- Wait for the code to **compile and upload** to your board.

Step 5: View the Output

- After uploading, open the **Serial Monitor** from the Tools menu.
- Set the **baud rate** to match the one used in the example code (usually **9600**, but check the code to confirm).
- You should now see real-time sensor data, including all **6 axes:**
    1. 3 axes of **magnetometer data**
    2. 3 axes of **accelerometer data** (See the image - 07 for an example output.)

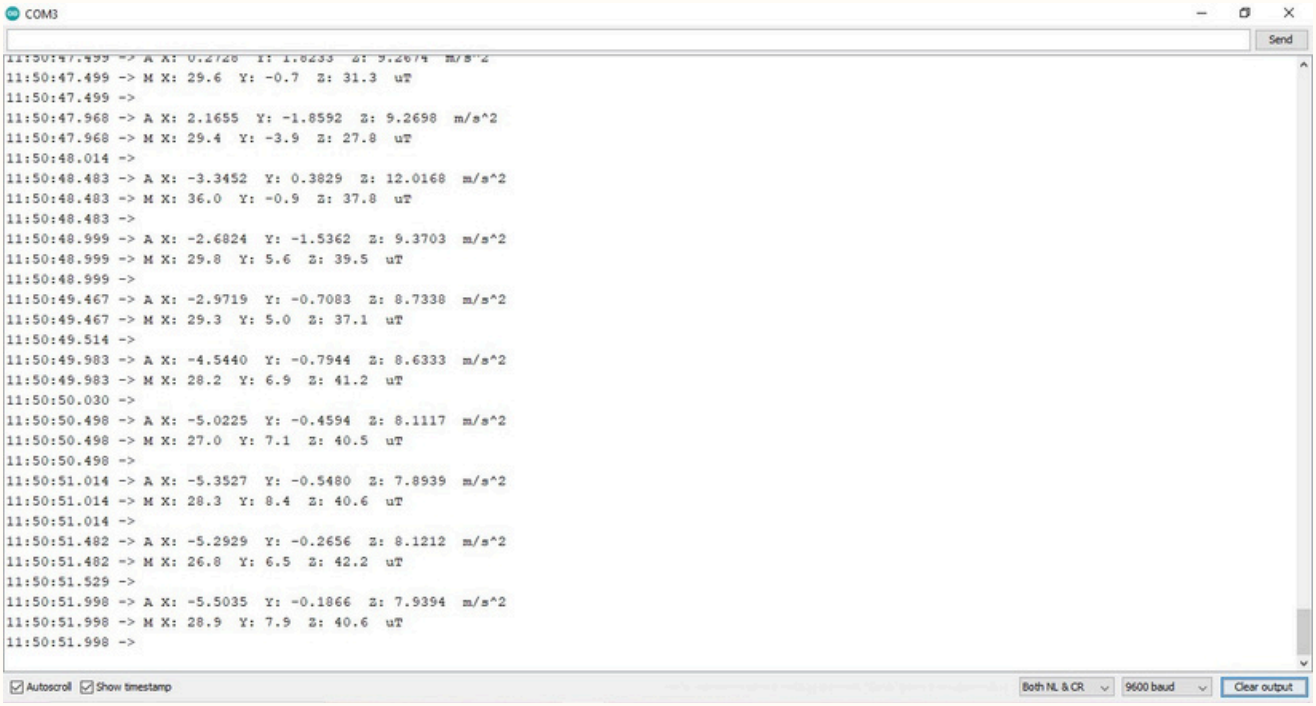


Image - 07

7. References

7.1 FXOS8700CQ Datasheet – NXP

7.2 Adafruit FXOS8700 Library GitHub