GY-521 Module (MPU6050)

Author: Roaida Binta Ali

Overview

The GY-521 is a compact breakout board that features the MPU6050 sensor — a 6-DOF (Degrees of Freedom) motion tracking device. It integrates:

- A 3-axis accelerometer
- A 3-axis gyroscope
- An I2C communication interface
- Onboard power regulation, pull-up resistors, and a status LED

It is widely used in motion sensing, robotics, drones, and wearables.

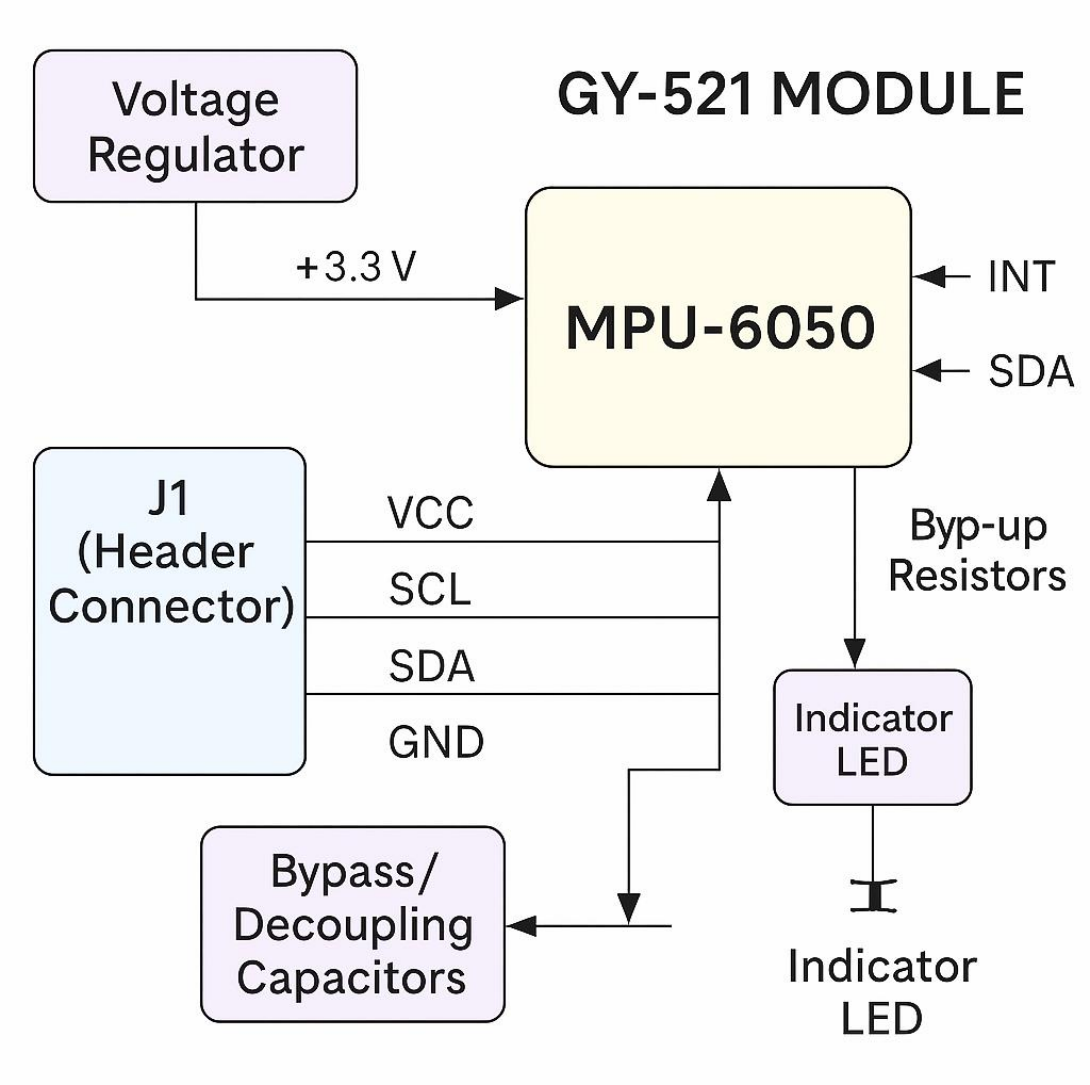
It includes:

Component	**Purpose**
MPU6050 chip	The core sensor (3-axis accelerometer + 3-axis gyroscope)
Voltage regulator (e.g., HX9193-33)	Converts 5V input to 3.3V for MPU6050
Pull-up resistors on SDA/SCL (R1, R2)	Required for proper I2C communication
Bypass capacitors (C1, C2, etc.)	Filter noise on power supply lines
Header pins	Easy connection to breadboard or wires

Why use GY-521 instead of raw MPU6050?

GY-521 Module	**Raw MPU6050**
Easy to use, ready-made	Needs SMD soldering
Has voltage regulation	Needs external regulator
Breadboard friendly	Not breadboard compatible
Comes with I2C pull-ups	Needs external components

Block Diagram



Uses of Components of GY-521

Component	Function
MPU6050 (U1)	Core sensor – 3-axis accelerometer + 3-axis gyroscope
Voltage Regulator (U2: HX9193-33)	Converts 5V input to 3.3V for MPU6050
Pull-up Resistors (R1, R2)	Required for SCL and SDA & stable I2C communication
Decoupling Capacitors (C1–C4)	Reduce voltage noise and ensure stable power These protect the MPU6050 from voltage spikes and interference.
LED (LED1)	Power indicator (turns ON when powered)
Header Pins (J1)	The 8-pin header lets you connect the GY-521 to your microcontroller.

MPU6050 Sensor Chip:

The MPU6050 is a sensor chip with:

- Accelerometer: Measures linear acceleration (X, Y, Z axes)
- Gyroscope: Measures angular velocity or rotation rate (X, Y, Z axes)
- Digital Motion Processor (DMP): Optional built-in processor to filter and combine data

What is I2C

I2C (Inter-Integrated Circuit) is a 2-wire serial communication protocol used by MPU6050 to communicate with a microcontroller.

I2C Line	Function
SCL	Serial Clock Line – provides timing
SDA	Serial Data Line – transfers data

- Supports multiple devices on the same bus
- MPU6050 uses 7-bit I2C address:
    - 0x68 (default)
    - 0x69 (if AD0 pin is pulled HIGH)

Pinout of GY-521

Pin	Function
VCC	Input power (3.3V–5V)
GND	Ground
SCL	I2C Clock
SDA	I2C Data
XDA/XCL	Aux I2C for other sensors
AD0	I2C address selector
INT	Interrupt output (e.g., motion detect)

LED Indicator

- LED1 is a red SMD LED
- Connected to 3.3V output via 2kΩ resistor (R4)

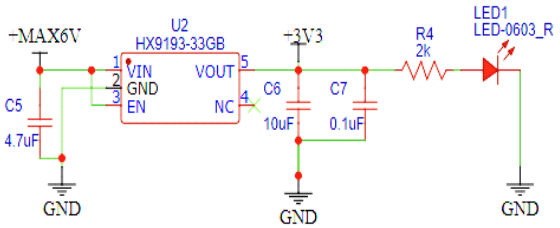
- Lights up when the board is properly powered
- Helpful for debugging power issues

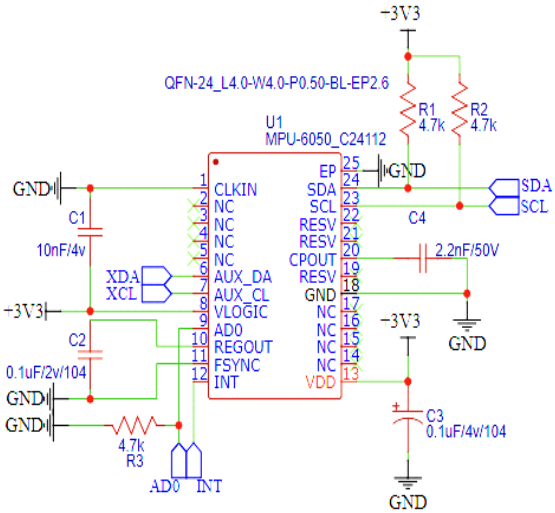
Applications

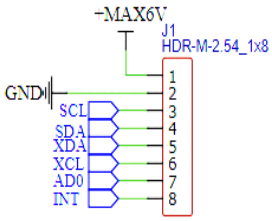
Field	Use Case
🤖 Robotics	Balancing robots, path tracking, motion detection
🚁 Drones	Flight control, stabilization, orientation
🕹️ Gaming	Tilt-based controllers, gesture recognition
👟 Wearables	Step counters, motion analysis
📱 Phones/Tablets	Screen rotation, motion-based UI

References

- MPU6050 Datasheet
- GY-521 Board Schematics







I2C Protocol — Complete Summary & MPU6050 Application

I2C (Inter-Integrated Circuit), pronounced "I-squared-C" or "I-two-C," is a serial communication protocol that allows multiple devices to communicate over just two wires:

- SDA — Serial Data Line (bi-directional data)
- SCL — Serial Clock Line (clock signal from master)

It uses a master-slave architecture where one master controls the bus and multiple slaves respond.

Basic Features

Feature	Description
Wires Used	2 (SDA and SCL)
Speed	Standard: 100 kbps, Fast: 400 kbps, High: 3.4 Mbps
Number of Devices	Multiple (up to ~128 devices)
Communication	Master-slave
Addressing	7-bit or 10-bit device addresses

How I2C Communication Works

1. Master sends a START condition.
2. Master sends slave address.
3. Slave sends ACK (acknowledge).
4. Master sends or requests data bytes, one at a time.
5. Master sends STOP condition to end communication.

Real-Life Analogy

Think of I2C like a school intercom:

- The principal (Master) calls a student (Slave) by name (address).
- Only the called student responds.
- All share the same two wires, but only one talks at a time.

Why Use I2C?

- Simple wiring: only two wires needed for many devices.
- Multi-device support: many sensors and modules can share the same bus.

- Widely supported: nearly all microcontrollers have I2C support.
- Reasonable speed: faster than UART in many cases.

I2C Compared to Other Protocols

Feature	I2C	SPI	UART
Wires	2	4	2
Devices	Many	Few	2
Speed	Medium	High	Medium
Master/Slave	Yes	Yes	No (peer-to-peer)

I2C in MPU6050 (Gyroscope + Accelerometer)

- MPU6050 has a built-in hardware I2C interface.
- Uses two I2C pins:
    - Pin 22 — SCL (clock input from master)
    - Pin 23 — SDA (bi-directional data line)
- Connect these pins to microcontroller I2C pins with pull-up resistors (4.7kΩ) to +3.3V or +5V.
- MPU6050's AD0 pin selects its I2C address (usually 0x68 or 0x69).
- Through I2C, the master (e.g., Arduino) can:
    - Configure sensor settings (registers)
    - Read acceleration, gyro, temperature data
    - Receive interrupt signals

Typical MPU6050 to Arduino I2C Connection

MPU6050 Pin Connects To Arduino Uno Pin

SDA (23) A4

SCL (22) A5

Summary Notes

- I2C = Inter-Integrated Circuit, a simple 2-wire serial protocol.
- Uses SDA (data) and SCL (clock) lines.

- Supports one master, multiple slaves.
- Devices have 7-bit or 10-bit addresses.
- Speeds: 100 kHz (Standard), 400 kHz (Fast), 3.4 MHz (High-Speed).
- MPU6050 has built-in I2C for easy integration.
- Pull-up resistors are required on SDA and SCL.
- Common devices using I2C: MPU6050, EEPROMs, OLED displays, RTCs.