**How to Connect and Read Data from MPU6050**

**Connection:**

* Plug MPU6050 into breadboard
* Connect MPU6050 to Arduino via wire connection:
  + VCC pin → 5V
  + GND pin → GND
  + SDA pin → A4 analog pin (Arduino)
  + SCL pin → A5 analog pin (Arduino)

**Setup:**

* Install and open Arduino IDE
* Install necessary libraries
  + Go to “Sketch” → ‘Include Library” → “Manage Libraries”
  + Search for “Adafruit MPU6050” and install it
* Include libraries in Sketch
  + Include all necessary libraries in the Arduino Sketch. This is done by adding the following lines at the beginning of the Sketch:
    - **#include <Wire.h>**
    - **#include <Adafruit\_MPU6050.h>**
    - **#include <Adafruit\_Sensor.h>**
  + These libraries provide functions for communicating with the MPU sensor and interpreting its data
* Instantiate object
  + After including the libraries, you need to instantiate the MPU6050 object in the Sketch:
    - **Adafruit\_MPU6050 mpu;**
  + This creates an instance of the MPU6050 object named “**mpu**” that you can use to interact with the sensor
* Establish serial communication
  + Add the “**Serial.begin()**” at the start of the setup to initialize the serial communication
* Set ranges and filter bandwidth on “**mpu**”
  + Set the ranges for the gyroscope and accelerometer, and filter bandwidth:
    - **mpu.setAccelerometerRange(MPU6050\_RANGE\_8\_G);**
    - **mpu.setGyroRange(MPU6050\_RANGE\_500\_DEG);**
    - **mpu.setFilterBandwidth(MPU6050\_BAND\_5\_HZ);**

**Reading Data:**

* Create variables to store sensor data
  + Create variables of type “**sensors\_event\_t**”, to store sensor readings for accelerometer (“**a**”), gyroscope (“**g**”), and temperature (“**temp**’):
    - **sensors\_event\_t a, g, temp;**
* Read the sensor data
  + Call the “**getEvent()**” function of the MPU6050 object (“**mpu**”) to read the sensor data:
    - **mpu.getEvent(&a, &g, &temp);**
  + The function takes three pointers as arguments, each pointing to a “**sensors\_event\_t”** structure where the data will be stored. It retrieves the latest sensor readings and stores them in the variables “**a**”, “**g**”, “**temp**”
* Print accelerometer data
  + Print the accelerometer data using “**Serial.print();**”
    - **Serial.print(a.acceleration.x);**
    - **Serial.print(a.acceleration.y);**
    - **Serial.print(a.acceleration.z);**
* Print gyroscope data
  + Print the gyroscope data using “**Serial.print();**”
    - **Serial.print(g.gyro.x);**
    - **Serial.print(g.gyro.y);**
    - **Serial.print(g.gyro.z);**