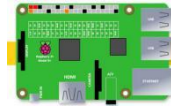
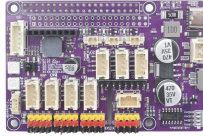



Lesson 18 Reading Data from MPU6050

18.1 Overview

In this lesson, we will learn how to interface the MPU6050 sensor with a Raspberry Pi using the Adept Robot HAT V3.2. The MPU6050 is a widely used 6 - axis motion tracking device, and we will explore how to read acceleration and angular velocity data from it. This knowledge is useful for various applications such as robotics, motion - controlled projects, and inertial measurement systems.

18.2 Required Components

Components	Quantity	Picture
Raspberry Pi	1	
Adept Robot HAT V3.2	1	
MPU6050	1	

18.3 Principle Introduction

MPU-6050 is the world's first integrated 6-axis Motion Tracking device. It integrates a 3-axis MEMS gyroscope, a 3-axis MEMS accelerometer, and an expandable digital motion processor (DMP), and can connect a third-party digital sensor like magnetometer, with an I2C interface. After expansion, it can output a 9-axis signal through its I2C or SPI interface (SPI interface is only

available in MPU-6000). MPU-60X0 can also be connected to non-inertial digital sensors (e.g., pressure sensor) through its I2C interface.

MPU-6050 uses three 16-bit ADCs (analog-to-digital converter) respectively for the gyroscope and accelerometer, and converts the measured analog quantity into an exportable digital quantity. The gyroscope can measure angular velocity, and the accelerometer do acceleration. To accurately track fast and slow motion, the measuring range of the sensor is adjustable: range of the gyroscope is ± 250 , ± 500 , ± 1000 , $\pm 2000^\circ/\text{sec}$ (dps, or degrees per second), and that of the accelerometer is ± 2 , ± 4 , ± 8 , $\pm 16g$ (gravitational acceleration).

An on-chip 1024-byte FIFO (first in first out) helps reduce system power consumption. A 400kHz I2C interface is used for the communication with all device registers. In addition, a temperature sensor, an oscillator with a $\pm 1\%$ variation in the working environment, and a programmable low-pass filter (LPF) are embedded on the chip.

As for power, the MPU-6050 module supports a VCC's voltage operation range of 3-5VDC.



18.3.1 How the MPU6050 Works

Acceleration Measurement: The 3 - axis accelerometer in the MPU6050 measures the acceleration forces acting on the device along the X, Y, and Z axes. These forces can be due to gravity (when the device is stationary or moving in a gravitational field) or other external forces (such as acceleration during motion). The measured acceleration values are converted into digital form by the 16 - bit ADCs for further processing.

Angular Velocity Measurement: The 3 - axis gyroscope measures the angular velocity of the device's rotation around the X, Y, and Z axes. This is useful for detecting rotational motion, such as the turning of a robotic arm or the rotation of a vehicle.

DMP Function: The Digital Motion Processor (DMP) in the MPU6050 can perform complex calculations on the raw sensor data. It can, for example, combine the acceleration and angular

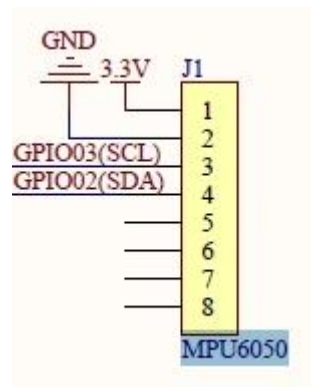
velocity data to provide more accurate orientation information, reducing the computational load on the Raspberry Pi.

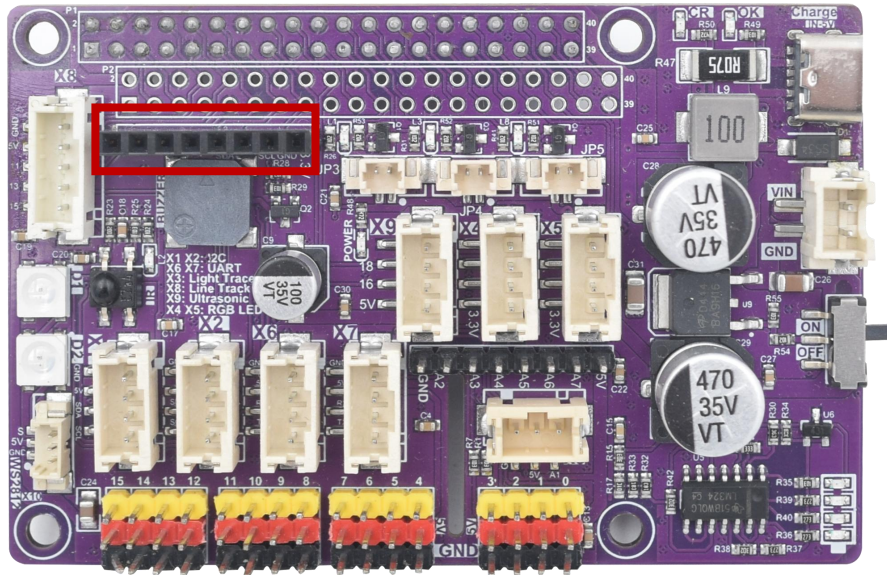
I2C Communication: The MPU6050 communicates with the Raspberry Pi via the I2C interface. The SCL (Serial Clock) and SDA (Serial Data) lines are used to send commands to the MPU6050's registers and receive data from it. The I2C protocol allows for efficient communication between the two devices, enabling the Raspberry Pi to read sensor data and configure the MPU6050's settings.

PINS of Raspberry Pi	MPU6050
GPIO03(SCL)	SCL
GPIO02(SDA)	SDA
3.3V	VCC
GND	GND

18.4 Wiring Diagram

Connect the MPU6050 module to the MPU6050 port on the Adeept Robot HAT V3.2, as shown below:





18.5 Demonstration

1. **Remotely log:** Remotely log in to the Raspberry Pi terminal.
2. **Navigate to the Program Folder:** Enter the following command in the terminal and press Enter to access the folder where the program is located:

```
cd Adeept_PiCar-Pro/Examples/12_MPU6050/
```

```
pi@raspberrypi:~ $ cd Adeept_PiCar-Pro/Examples/12_MPU6050/
pi@raspberrypi:~/Adeept_PiCar-Pro/Examples/12_MPU6050 $
```

3. **View Directory Contents:** Type "ls" in the terminal and press **Enter**. This will display all the files in the current directory, ensuring that the "**Mpu6050.py**" file is present:

```
ls
```

```
pi@raspberrypi:~ $ cd Adeept_PiCar-Pro/Examples/12_MPU6050/
pi@raspberrypi:~/Adeept_PiCar-Pro/Examples/12_MPU6050 $
```

4. **Run the Program:** Enter the command below and press Enter to start the **Mpu6050.py** program:

```
sudo python3 Mpu6050.py
```

```
pi@raspberrypi:~/Adeept_PiCar-Pro/Examples/12_MPU6050 $ sudo python3 Mpu6050.py
X=-0.167, Y=-0.124, Z=9.767
X=-0.198, Y=-0.100, Z=9.829
X=-0.151, Y=-0.148, Z=9.832
X=-0.160, Y=-0.081, Z=9.798
X=-0.201, Y=-0.085, Z=9.828
X=-0.150, Y=-0.102, Z=9.791
X=-0.114, Y=-0.082, Z=9.839
X=-0.222, Y=-0.113, Z=9.845
X=0.010, Y=-0.099, Z=9.751
X=-0.119, Y=-0.146, Z=9.828
```

5. **Observation and Termination:** After successfully running the code, you can see the values of gravitational acceleration at X, Y, and Z axes.

When the value of X is 0, it indicates that the X-axis direction of the MPU6050 is horizontal. If the value of X is positive, it means that the X-axis of the MPU6050 is offset upward. If the value of X is negative, it implies that the X-axis of the MPU6050 is offset downward. When the value of X is equal to the gravitational acceleration value (approximately 9.8), it represents that the X-axis direction of the MPU6050 is upward and perpendicular to the ground. When the value of X is -9.8, it shows that the X-axis direction of the MPU6050 is downward and perpendicular to the ground. The situation of the Y-axis is similar to that of the X-axis.

Minor data deviation may happen. and when you want to terminate the running program, you can press the "**Ctrl+C**" shortcut key on the keyboard.

18.6 Code

Complete code refer to [Mpu6050.py](#)

```
01  #!/usr/bin/env/python
02  # File name   : Mpu6050.py
03  # Website    : www.Adeept.com
04  # Author     : Adeept
05  # Date      : 2025/03/11
06
07  from mpu6050 import mpu6050
08  import time
09
10  sensor = mpu6050(0x68)
11  def mpu6050test():
12      x = 0
```

```
13 y = 0
14 z = 0
15 for i in range(0,10):
16     accelerometer_data = sensor.get_accel_data()
17     x = x + accelerometer_data['x']
18     y = y + accelerometer_data['y']
19     z = z + accelerometer_data['z']
20     print('X=%.3f, Y=%.3f, Z=%.3f'%(x/10.0,y/10.0,z/10.0))
21     time.sleep(0.3)
22
23 if __name__ == "__main__":
24     try:
25         while True:
26             mpu6050test()
27     except:
28         pass
29
```

Code explanation

Mpu6050.py

Initialization Stage:

An instance of the mpu6050 class is created with the I2C address 0x68. This address is used to communicate with the MPU6050 sensor over the I2C bus.

Loop Control Process:

Stage 1: An infinite while loop is entered. Inside this loop, the mpu6050test function is called repeatedly to continuously collect and print the average acceleration data from the MPU6050 sensor.

Stage 2: Exception handling: A try - except block is used to catch any exceptions that may occur during the execution of the code. If an exception occurs, the program simply passes without taking any further action.

This code initializes the MPU6050 sensor, defines a function to collect and calculate the average acceleration data, and then enters an infinite loop to continuously perform data collection and printing operations.