

Got angle and angular velocity (DMP algorithm)

Got angle and angular velocity (DMP algorithm)

[DMP algorithm](#)

[Quaternion solution](#)

[Quaternion to Euler angle](#)

[Quaternion](#)

[Euler angle](#)

[Conversion formula](#)

[Solution code](#)

[Software code](#)

The tutorial introduces the use of DMP algorithm to calculate the attitude angle.

The MPU6050 module can detect the motion data of the three-axis acceleration, the three-axis gyroscope, and the temperature data.

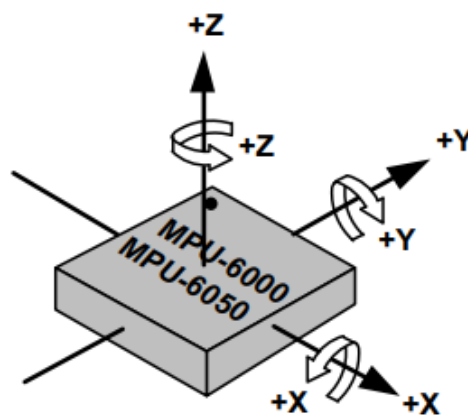
DMP algorithm

DMP is the motion engine inside the MPU6050, the full name of which is Digital Motion Processor. It directly outputs quaternions, which can reduce the workload of the peripheral microprocessor and avoid tedious filtering and data fusion.

Motion Driver is a software package for its motion sensors from Invensense. It is not all open source. The core algorithm part is compiled into a static link library for ARM processors and MSP430 processors, and is suitable for sensors such as MPU6050, MPU6500, MPU9150, and MPU9250.

Quaternion solution

Using the MPU6050 driver library provided by InvenSense, the raw data will be converted into quaternion output. We need to calculate the quaternion and finally get the attitude data (Euler angle): heading angle (Yaw), roll angle (Roll) and pitch angle (Pitch).



Quaternion to Euler angle

Quaternion

Quaternion is a mathematical tool used to represent three-dimensional rotation, but it is not very intuitive to observe.

Quaternions consist of four real numbers, and the basic representation is as follows:

$$q = q_0 + q_1 * i + q_2 * j + q_3 * k$$

Euler angle

Euler angle is a way to describe the rotation state of an object in three-dimensional space. It is usually composed of three angle values: yaw, roll, and pitch, which is convenient for observing posture.

Conversion formula

```
pitch = asin(-2 * q1 * q3 + 2 * q0 * q2)
roll = atan2(2 * q2 * q3 + 2 * q0 * q1, -2 * q1 * q1 - 2 * q2 * q2 + 1)
yaw = atan2(2 * (q1 * q2 + q0 * q3), q0 * q0 + q1 * q1 - q2 * q2 - q3 * q3)
```

Solution code

MPU6050 DMP outputs the quaternion after attitude settlement, in q30 format, magnified 2^{32} times ($q30 = 2^{32} = 1073741824$).

```
void Read_DMP(void)
{
    unsigned long sensor_timestamp;
    unsigned char more;
    long quat[4];

    dmp_read_fifo(gyro, accel, quat, &sensor_timestamp, &sensors, &more); //Read DMP data
    if (sensors & INV_WXYZ_QUAT )
    {
        //Quaternion
        q0=quat[0] / q30;
        q1=quat[1] / q30;
        q2=quat[2] / q30;
        q3=quat[3] / q30;
        Roll = asin(-2 * q1 * q3 + 2 * q0 * q2)* 57.3; //Calculate the roll angle
        Pitch = atan2(2 * q2 * q3 + 2 * q0 * q1, -2 * q1 * q1 - 2 * q2 * q2 + 1)* 57.3;
        //Calculate the pitch angle
        Yaw = atan2(2 * (q1 * q2 + q0 * q3), q0 * q0 + q1 * q1 - q2 * q2 - q3 * q3) * 57.3; //Calculate the yaw angle
    }
}
```

Software code

Balance car PID control basics: 08-13 tutorial only provides one project file.

Product supporting materials source code path: Attachment → Source code summary → 3.PID_Course → 08-13.Balanced_Car_PID