

6.3 MPU6050 calibrates the direction of motion

1. Learning goal:

Get gyroscope data, calibrate the car's movement status, and let the car go straighter.

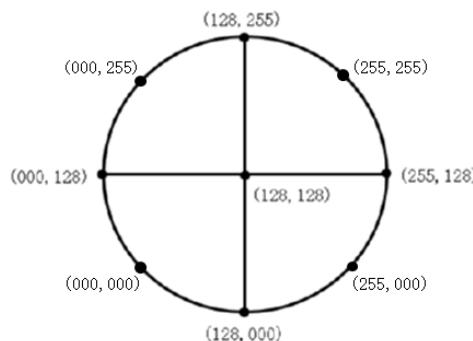
2. Experimental phenomena:

Because of the individual differences of the motor, the car will not go straight when going forward. In this course, we use the gyroscope to check the attitude angle of the car, so that the car can maintain a fixed attitude angle motion as much as possible.

3. Analysis:

3.1 The APP analog rocker sends data, the X axis gradually increases from left to right (0~255), the Y axis gradually increases from bottom to top (0~255), and the middle position is (128,128). Based on this feature, we recreate a control mode function to handle the situation of the analog rocker control car.

The specific code can be found in the MPU6050_DMP.ino source code provided in the data.



4. About code

4.1 Define MPU6050 variable for storing data

```
// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte frc
uint8_t devStatus; // return status after each device operat
uint16_t packetSize; // expected DMP packet size (default is 4
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorInt16 aa; // [x, y, z] accel sensor measure
VectorInt16 gy; // [x, y, z] gyro sensor measurem
VectorInt16 aaReal; // [x, y, z] gravity-free accel s
VectorInt16 aaWorld; // [x, y, z] world-frame accel se
VectorFloat gravity; // [x, y, z] gravity vector
float euler[3]; // [psi, theta, phi] Euler angle containe
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll conta

volatile bool mpuInterrupt = false; // indicates whether MPU inte
void dmpDataReady()
{
    mpuInterrupt = true;
}
```

4.2 New create PID related parameter.

```

/*=====
PID Function
The PID function is used in mainly
control applications. PIDCalc performs one iteration o
algorithm.

While the PID function works, main is just a dummy pro
a typical usage.
=====

typedef struct
{
    float SetPoint;    //Desired value
    float Proportion; //Proportional Const
    float Integral;   //Integral Const
    float Derivative; //Derivative Const
    floatLastError;  //Error[-1]
    float PrevError; //Error[-2]
    float SumError;   //Sums of Errors
} PID;
}

/*PID calculation section*/
PID omega_PID = {0, 0.4, 0, 0.1, 0, 0, 0};

float omega_Work = 0;

```

4.3 New create PID calculate function

$$PID=U_k+K_P \cdot [E(k)-E(k-1)] + K_I \cdot E(k) + K_D \cdot [E(k)-2E(k-1)+E(k-2)]$$

```

float PIDCal_car(float NextPoint)
{
    float dError, Error;
    Error = omega_PID.SetPoint - NextPoint;
    omega_PID.SumError += Error;
    dError = omega_PID.LastError - omega_PID.PrevError;
    omega_PID.PrevError = omega_PID.LastError;
    omega_PID.LastError = Error;

    double omega_rad = omega_PID.Proportion * Error
                    + omega_PID.Integral * omega_PID.SumError
                    + omega_PID.Derivative * dError;

    if (omega_rad > M_PI / 6)
        omega_rad = M_PI / 6;
    if (omega_rad < -M_PI / 6)
        omega_rad = -M_PI / 6;
    return -omega_rad;
}

```

4.4 MPU6050 initialization

```

void initMPU6050()
{
    mpu.initialize();

    // verify connection
    bool mpu_state = mpu.testConnection();

#ifdef ENABLE_DEBUG
    Serial.println(F("Testing device connections..."));
    Serial.println(mpu_state ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));

    // load and configure the DMP
    Serial.println(F("Initializing DMP..."));
#endif

    devStatus = mpu.dmpInitialize();

    // supply your own gyro offsets here, scaled for min sensitivity
    mpu.setXGyroOffset(51);
    mpu.setYGyroOffset(8);
    mpu.setZGyroOffset(21);
    mpu.setXAccelOffset(1150);
    mpu.setYAccelOffset(-50);
    mpu.setZAccelOffset(1060);

    // make sure it worked (returns 0 if so)
    if (devStatus == 0)
    {
        // Calibration Time: generate offsets and calibrate our MPU6050
        mpu.CalibrateAccel(6);
        mpu.CalibrateGyro(6);

#ifdef ENABLE_DEBUG
        Serial.println();
        mpu.PrintActiveOffsets();
        // turn on the DMP, now that it's ready
        Serial.println(F("Enabling DMP..."));
#endif
    }

    mpu.setDMPEnabled(true);

    // enable Arduino interrupt detection
#ifdef ENABLE_DEBUG
    Serial.print(F("Enabling interrupt detection (Arduino external interrupt "));
    Serial.print(digitalPinToInterrupt(INTERRUPT_PIN));
    Serial.println(F(")..."));
#endif

    attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();

    // set our DMP Ready flag so the main loop() function knows it's okay to use it
#ifdef ENABLE_DEBUG
    Serial.println(F("DMP ready! Waiting for first interrupt..."));
#endif

    dmpReady = true;
}

```

```

    // get expected DMP packet size for later comparison
    packetSize = mpu.dmpGetFIFOPacketSize();
    digitalWrite(LED_PIN, LOW);      // LED D9 is on
}
else
{
// ERROR!
// 1 = initial memory load failed
// 2 = DMP configuration updates failed
// (if it's going to break, usually the code will be 1)
#ifndef ENABLE_DEBUG
    Serial.print(F("DMP Initialization failed (code "));
    Serial.print(devStatus);
    Serial.println(F(")"));
#endif

    digitalWrite(LED_PIN, HIGH);     //LED D9 is off
}
}

```

4.5 Get MPU6050 data, The output data will be saved in the array ypr.

```

void mpu6050_getdata()
{
    // if programming failed, turn off LED9 and don't try to do anything
    if (!dmpReady)
    {
        digitalWrite(LED_PIN, HIGH); // turn off LED9
        return;
    }

    // reset interrupt flag and get INT_STATUS byte
    mpuInterrupt = false;
    mpuIntStatus = mpu.getIntStatus();

    // get current FIFO count
    fifoCount = mpu.getFIFOCount();

    // check for overflow (this should never happen unless our code is too inefficient)
    if ((mpuIntStatus & _BV(MPU6050_INTERRUPT_FIFO_OFLOW_BIT)) || fifoCount >= 1024)
    {
        // reset so we can continue cleanly
        mpu.resetFIFO();
        fifoCount = mpu.getFIFOCount();
    }
}

```

```

else if (mpuIntStatus & _BV(MPU6050_INTERRUPT_DMP_INT_BIT))
{
    // wait for correct available data length, should be a VERY short wait
    while (fifoCount < packetSize)
        fifoCount = mpu.getFIFOCount();

    // read a packet from FIFO
    mpu.getFIFOBytes(fifoBuffer, packetSize);

    // track FIFO count here in case there is > 1 packet available
    // (this lets us immediately read more without waiting for an interrupt)
    // fifoCount -= packetSize;

    // display Euler angles in degrees
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    mpu.dmpGetAccel(&aa, fifoBuffer);

    digitalWrite(LED_PIN, LOW); //turn on LED9
}
}

```

4.6 Obtain the MPU6050 data in the loop function, PID calculation and then store it in omega_Work

```

mpu6050_getdata();
omega_Work = PIDCal_car(ypr[0]);

```

4.7 New create control car rocker_carRun() function, x and y are the input analog rocker values, spin for rotation and yaw for yaw angle.

```

void rocker_carRun(int x, int y, int spin, float yaw)
{
    int speed_yaw_b = 0, speed_yaw_f = 0;
    //Debounce
    if (y >= 200 && abs(x - 128) <= 50)
    {
        x = 128;
    }
    else if (x >= 200 && abs(y - 128) <= 50)
    {
        y = 128;
    }

    if (abs(x - 128) < 30 && abs(y - 128) < 30)
    {
        x = 128;
        y = 128;
        omega_PID.SetPoint = ypr[0];      //Set initial yaw angle
    }
    //Adjust the speed according to the angle
    else if (abs(x - 128) > 30 && abs(y - 128) < 30)
    {
        speed_yaw_f = CarSpeedControl * sin(yaw);
        speed_yaw_b = speed_yaw_f;
    }
    else if (abs(x - 128) < 30 && abs(y - 128) > 30)
    {
        speed_yaw_f = CarSpeedControl * sin(yaw);
        speed_yaw_b = 0;
    }
}

```

```

int speed_X = map(x, 0, 255, -CarSpeedControl, CarSpeedControl);
int speed_Y = map(y, 0, 255, -CarSpeedControl, CarSpeedControl);
int spin_A = map(spin, -1, 1, -30, 30); //Fixed rotation speed of 30

//Motor speed
int speed_L1 = speed_Y + speed_X + spin_A - speed_yaw_f;
int speed_L2 = speed_Y - speed_X + spin_A - speed_yaw_b;
int speed_R1 = speed_Y - speed_X - spin_A + speed_yaw_f;
int speed_R2 = speed_Y + speed_X - spin_A + speed_yaw_b;

//Drive car
carRun(speed_L1, speed_R1, speed_L2, speed_R2);
delay(20);
}

```

5. Compiling and downloading code

5.1 Open the MPU6050_DMP.ino program in the course data, select the serial port and click upload directly (the Omniduino car must first be connected to the computer via the USB data cable).

5.2 If there is an error like the following, it means that the library file is missing. Please copy the library file provided by the omniduino omnibus to the library file directory compiled by arduinolDE.

please refer to 【3.Development Environment Construction】 ---- 【3.4 Add additional library files】

```

Adafruit_PWMSServoDriver.h: No such file or directory

CarRun:2:10: error: Adafruit_PWMSServoDriver.h: No such file or directory

#include <Adafruit_PWMSServoDriver.h>

^

compilation terminated.

exit status 1
Adafruit_PWMSServoDriver.h: No such file or directory

```

5.3 If the compilation passes normally, but the following error occurs during uploading, the reason may be that the wrong serial port or the serial port is occupied.

```

An error occurred while uploading the sketch
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039
An error occurred while uploading the sketch
avrdude: ser_open(): can't open device "\.\COM32": The system can

```

Solution: Open the device manager to see if there is a serial port with CH340 tag. If not, please restart the Omniduino car, then, re-plug the USB cable or replace a USB cable; If there is a serial port number, we need to close the other serial port or assistant software, avoid serial port occupation, and then re-select the serial port to ArduinoIDE 【Tool】-->【Port】.

6.4 After clicking the upload button, the upload is always displayed, but it can't be uploaded successfully for a long time.

Problem uploading to board. See <http://www.arduino.cc/en/Guide/Troubleshooting#upload> for suggestions

avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 4 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 5 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 6 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 7 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 8 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 9 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 10 of 10: not in sync: resp=0xec
Problem uploading to board. See <http://www.arduino.cc/en/Guide/Troubleshooting#upload> for suggestions

[Copy error messages](#)

Because the uploading program and the WIFI camera communication is realized through the serial port, when the serial port is occupied by the WIFI camera, and the program cannot be uploaded.

Solution:

- ①Unplug the USB cable, turn off the power of the car, wait for the D2 indicator to go out.
- ②Then, plug in the USB data cable. At this time, your mobile phone should not connect the WiFi signal of the car.
- ③You can upload the program to the car according to the normal steps.
- ④After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. The corresponding experimental phenomenon will appear.
(Tip: If you upload APP control program. After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. Mobile phone connect the car to the WIFI signal, and then open the APP to control.)