6.9 Car self-stabilizing mode。

1. Learning goal

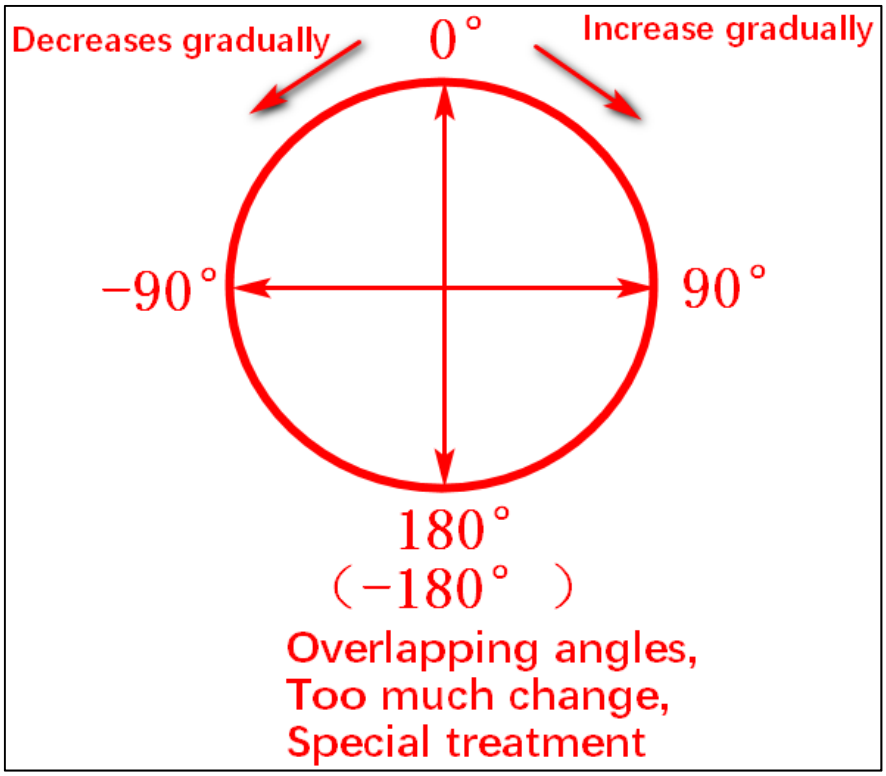
Obtain gyroscope data, record the gyroscope angle of the cart itself, and when external forces change the cart angle, the cart can adjust angle by itself.

2. Experimental phenomenon:

Record the current gyroscope angle when the car is turned on, when we manually change the angle of the car, the car will automatically start the motor, rotate towards the recorded direction, and stop within the allowed range of the recorded angle. The custom key function can modify the record of the car Gyroscope direction.

3. Experimental phenomena:

The initial angle of the gyroscope is zero. When it is rotated to the left, the angle value is negative; when it is rotated to the right, the angle value is positive. It will coincide at 180 degrees and -180 degrees, so the angle output different near 180 degrees will be very large, so we need to do special treatment for this situation, otherwise the car will always rotate and not stop automatically.



4. **Code analysis**

4.1 New create PID adjustment related parameters.

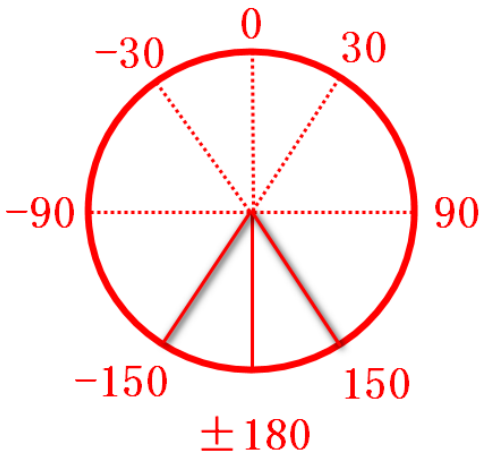
```
float PIDCal_Stabilize(PID pid, float nowValue);

/*==============================================
PID Calculation section
==============================================*/
PID alpha_PID = {0, 5, 0, 0.1, 0, 0, 0};
float alpha_Work = 0;
```

4.2 Create a new PID calculation function, and perform special processing between 150 degrees to 180 degrees and -180 degrees to -150 degrees according to the actual situation. First, we need to discuss the case where the gyroscope's alpha angle range is between -180 degrees and -150 degrees.

The X in the comment actually represents the angle value pid.SetPoint recorded by the gyroscope, and Y represents the current value of the gyroscope angle nowValue.

After X determines the value range, the value range of Y can be divided into two sections, the left part is rotated counterclockwise, and the right part is rotated clockwise.

①we can subdivide into multiple areas to discuss.



```
float PIDCal_Stabilize(PID pid, float nowValue)
{
  float dError, Error;
  Error = pid.SetPoint - nowValue; //Deviation
  //The following is a special case: the initial angle of the gyroscope is zero, it turns to th
  //If the following treatments are not added, it will cause the car to stop by 180 degrees. If
  if (pid.SetPoint < -(M_PI * 5/6) || pid.SetPoint > (M_PI * 5/6))
  {
    if (pid.SetPoint < 0) // -180 < X < -150
    {
      if ((M_PI + pid.SetPoint) < nowValue || pid.SetPoint > nowValue) //Clockwise
      {
        if (nowValue < 0) // -180 < Y < X
        {
          Error = 2 * M_PI + pid.SetPoint + nowValue;
        }
        else // Y > pi-X
        {
          Error = 2 * M_PI + pid.SetPoint - nowValue;
        }
      }
      else if ((M_PI + pid.SetPoint) > nowValue && pid.SetPoint < nowValue) //Counterclockwise
      {
        Error = pid.SetPoint - nowValue;
      }
```

②Discuss the case of alpha angle values between 150 and 180 degrees

```
  else // 150 < X < 180
  {
    if (nowValue < (-M_PI + pid.SetPoint) || pid.SetPoint < nowValue) //Counterclockwise
    {
      if (pid.SetPoint < nowValue)
      {
        Error = -(nowValue - pid.SetPoint);
      }
      else if (abs(pid.SetPoint) > abs(nowValue))
      {
        Error = -(2 * M_PI - pid.SetPoint + nowValue);
      }
      else
      {
        Error = -(pid.SetPoint + nowValue);
      }
    }
    else if ((-M_PI + pid.SetPoint) < nowValue && pid.SetPoint > nowValue) //Clockwise
    {
      Error = pid.SetPoint - nowValue;
    }
  }
}
```

③Other cases and PID calculation

```
  else
  {
    Error = pid.SetPoint - nowValue;    //Deviation
  }

  pid.SumError += Error;          //Integral
  dError = pid.LastError - pid.PrevError; //Current differential
  pid.PrevError = pid.LastError;
  pid.LastError = Error;

  double pid_value = pid.Proportion * Error
              + pid.Integral * pid.SumError
              + pid.Derivative * dError;

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

4.3 New created radian is converted into an angle function, which can be used for automatic stop adjustment later.

```
float Radians_To_Degrees(float a)
{
  return (a * 180 / M_PI); //angle
}
```

4.4 Key function, when you press button to refresh the recorded gyroscope angle.

```
void keyscan()
{
  int val;
  val = digitalRead(KEY_PIN);
  if (val == LOW)
  {
    delay(10);
    val = digitalRead(KEY_PIN);
    while (val == LOW)
    {
      val = digitalRead(KEY_PIN);
      if (val == HIGH)
      {
        whistle();
        alpha_PID.SetPoint = ypr[0];
        return;
      }
    }
  }
}
```

4.5 Get the MPU6050 data in the loop function, and then store it in alpha_Work through PID calculation. The offset is calculated mainly because there is some deviation in the value calculated by the gyroscope sensor. If the offset is relatively small (less than 3), you don't need to adjust the car to avoid the problem that the car's motor always rings.

```
void loop()
{
  keyscan();
  mpu6050_getdata();
  // Serial.println(ypr[0] * 180 / M_PI);
  alpha_Work = PIDCal_Stabilize(alpha_PID, ypr[0]);

  int nowDegree = abs(Radians_To_Degrees(ypr[0]));
  int point = abs(Radians_To_Degrees(alpha_PID.SetPoint));
  int offset = abs(nowDegree - point);

  if (offset >= 3)
  {
    int speed_spin = 50 * sin(alpha_Work);
    carRun(-speed_spin, speed_spin, -speed_spin, speed_spin);
  }
  else
  {
    carRun(0, 0, 0, 0);
  }
}
```

4.6 If you feel that the car's self-stabilization function is turning too slowly (the speed is too small), you need to increase the speed value.

Method:

You can slightly increase 50 in the speed_speed = 50 * sin (alpha_Work) statement.

Note:

The range of sin (alpha_Work) has been limited in the PID calculation, with a maximum of -0.5 and a minimum of 0.5. After the value of 50 is increased, the number of shocks may increase. You need to adjust the above PID parameters:

```
/*=======================================
PID Calculation section
=======================================*/
PID alpha_PID = {0, 5, 0, 0.1, 0, 0, 0};
```

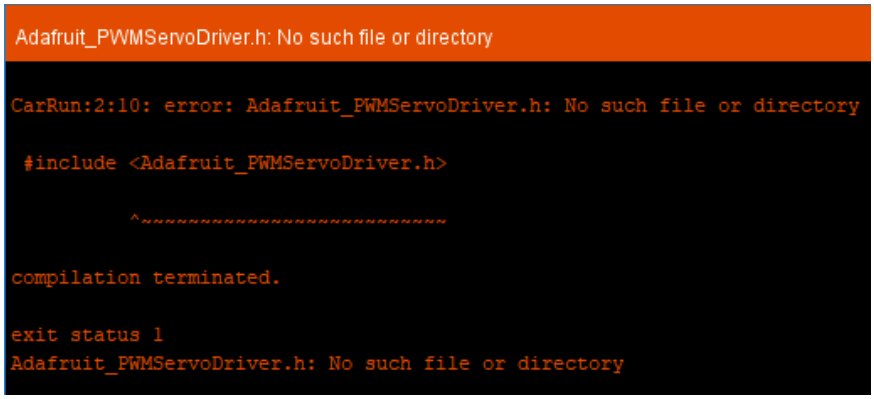
5. Compiling and downloading code

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

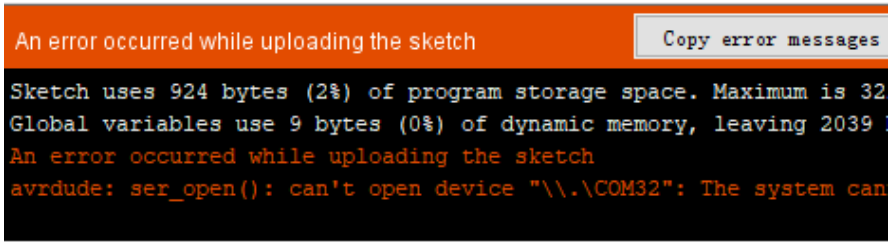
5.2 If there is an error like the following, it means that the library file is missing. Please copy the library file provided by the Omniduino omnibus to the library file directory compiled by arduinoIDE.

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

files】

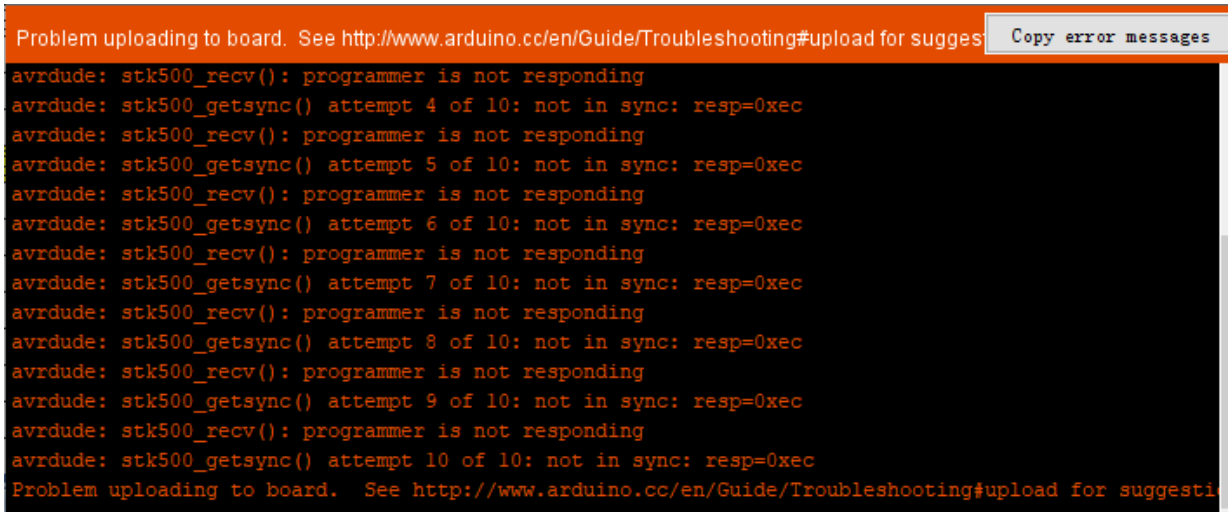


5.3 If the compilation passes normally, but the following error occurs during uploading, the reason may be that the wrong serial port or the serial port is occupied.



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】.

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



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.)