Gyroscope calibration straight-line walk

This tutorial is a comprehensive experiment combining multiple peripherals. You can understand individual peripherals before conducting this experiment.

1. Software-Hardware

- **KEIL5**
- **MSPM0G3507 Robot Development Board**

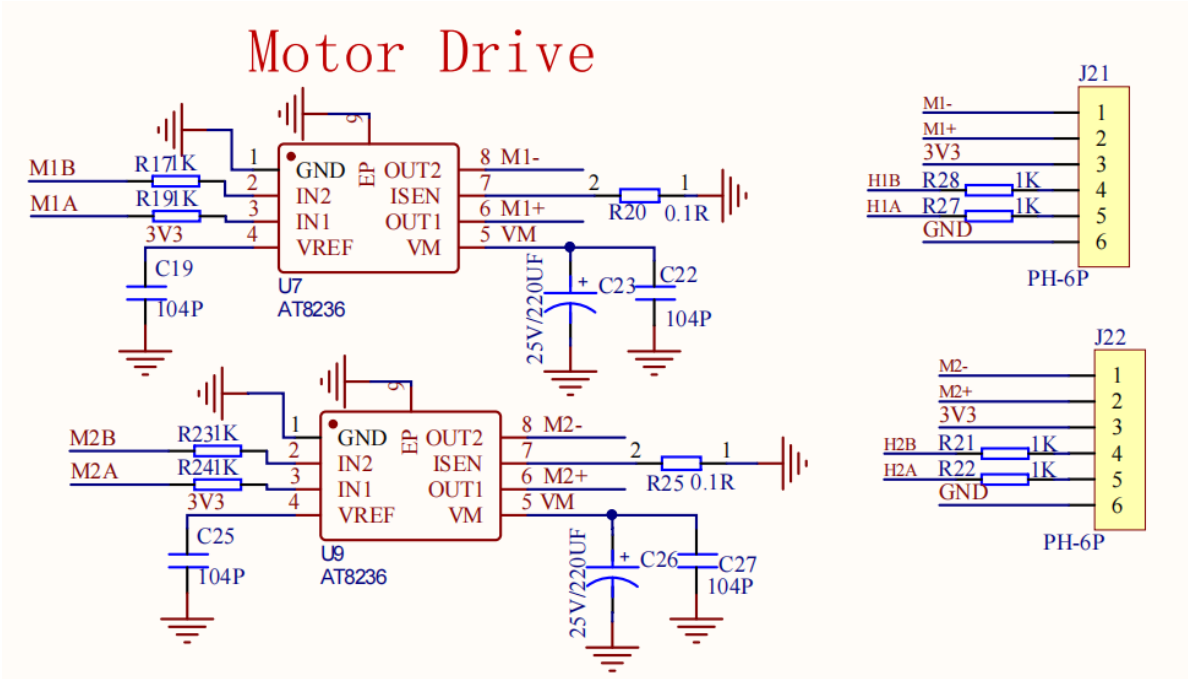
  MPU6050 module: external

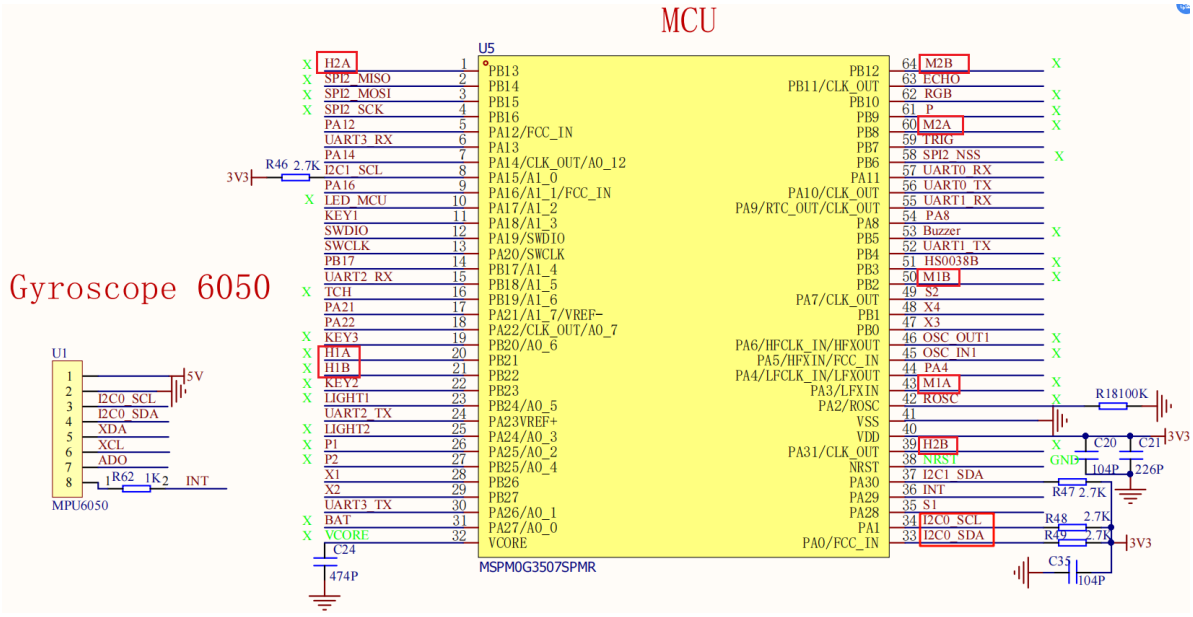
  OLED screen: external
- **Type-C data cable or DAP-Link**

  For program download or simulation to the development board

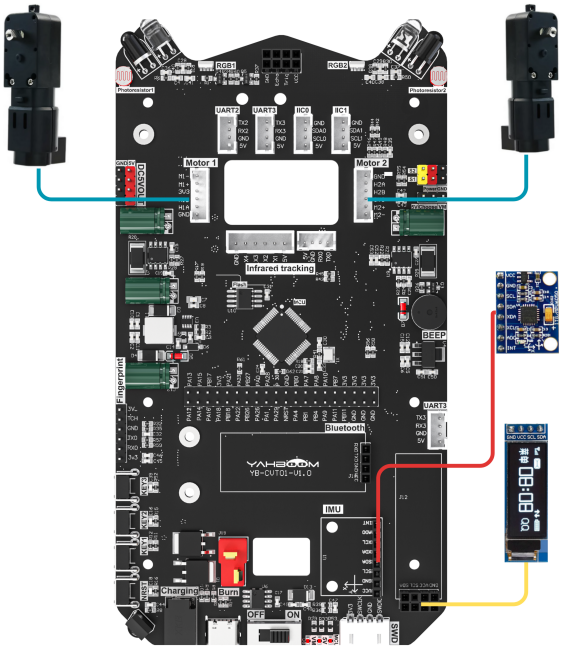
2. Brief Principle

2.1 Hardware Schematic Diagram



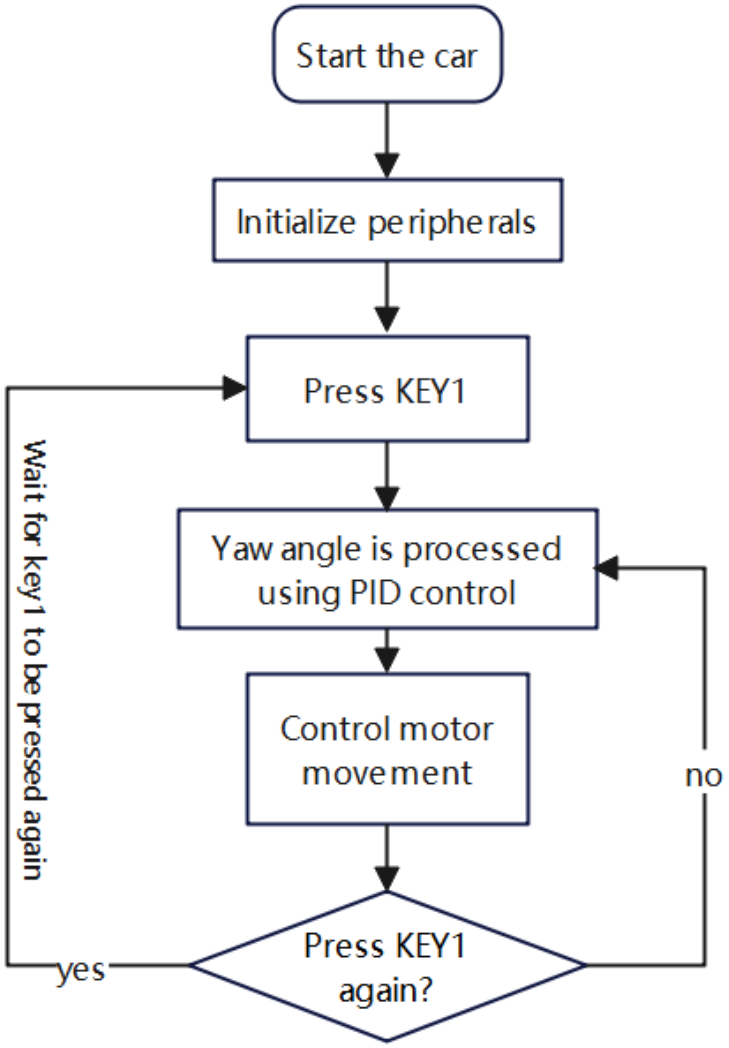


2.2 Physical Connection Diagram



2.3 Control Principle

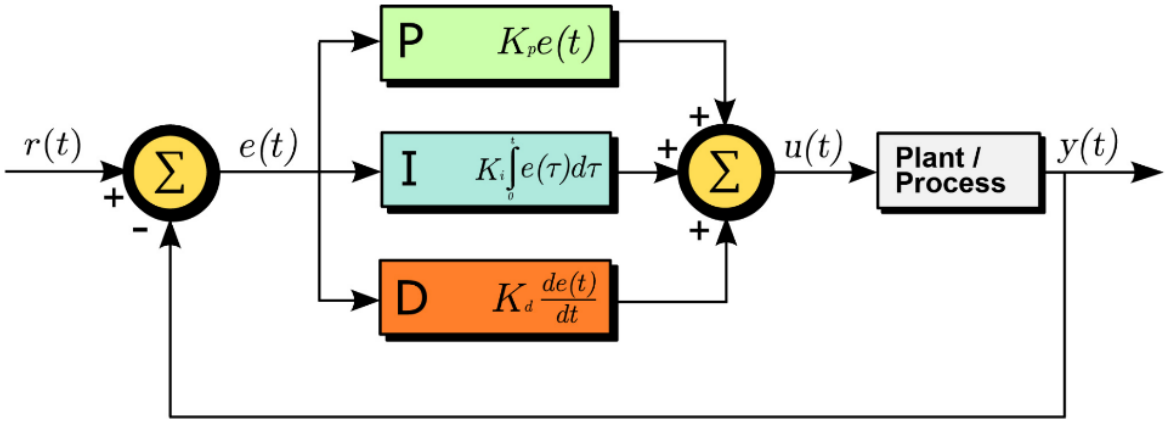
Flowchart:



Module	Function
MPU6050	Obtain fused data and transmit data to the microcontroller

Principle:

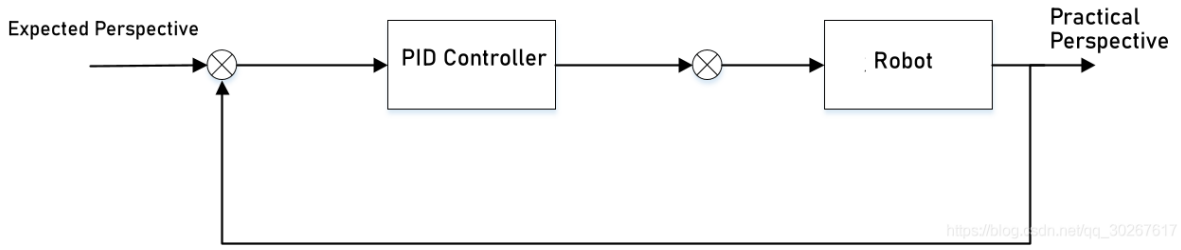
PID Controller, composed of proportional unit P, integral unit I and differential unit D. Through the setting of three parameters Kp, Ki and Kd. PID controller is mainly suitable for systems whose basic linearity and dynamic characteristics do not change with time.



r(t) setpoint, reference, is the expected process value or set value (SP); y(t) output, process variable, is the measured process value, output value (PV); e(t) error, is the deviation; u(t) control effort, is the control amount;

The significant characteristic of PID controller is the ability to utilize the influence of proportional, integral and differential control terms on controller output for precise and optimal control.

For attitude angle control, we want the robot to follow the given input for a given attitude angle. According to the single-stage PID approach, it should be like this:



3. Main Functions

Mainly introduces the functional code written by users, **for detailed code, you can open the project files we provide and view the source code in the Bsp folder**.

> **Function: BSP_Loop**

Function Prototype	**void BSP_Loop()**	
Function Description	Use button, press once to switch between straight-line walking and stopping	
Input Parameters	None	
Output Parameters	None	

> **Function: mpu6050_Line**

Function Prototype	**void mpu6050_Line()**	
Function Description	PID combined with mpu6050 controls straight-line walking	
Input Parameters	None	
Output Parameters	None	

> **Function: Get_CalibratedAngles**

Function Prototype	void Get_CalibratedAngles()
Function Description	Get calibrated angles
Input Parameters	None
Output Parameters	None

> Function: Dir_PID

Function Prototype	float Dir_PID(float error)
Function Description	Angle loop PID control
Input Parameters	Target angle
Output Parameters	Output angle

> Function: PID_Location_Calc

Function Prototype	float PID_Location_Calc(PID_t *pid, float actual_val)
Function Description	Speed loop: positional PID calculation method
Input Parameters	Target speed
Output Parameters	Output speed

4. Experimental Phenomenon

After successfully downloading the program, install the MPU6050 module and OLED screen on the expansion board. Then you can power on the car.

After power on, we need to wait for the mpu6050 value to stabilize. At this time, we can observe the OLED screen. After initialization is complete, the OLED will display ok, then we press the key1 button on the expansion board, and the car can start straight-line movement. If we adjust the car direction midway, the car will also self-adjust to walk straight.

If we want to stop the car movement, we need to press the key1 button again. If we want the car to move again, the same applies (need to press the key1 button).