中国科学技术大学

《嵌入式原理系统及应用》

课程设计报告



**基于FreeRTOS的GD32自平衡独轮车设计**

|  |  |
| --- | --- |
| 姓名： | 陈泽昊 |
| 学号： | BC23010002 |

**摘 要**

本课程设计的目标是通过动量轮控制和车轮控制实现独轮车的自平衡。作为第一版本的小车，结构方面存在很大的问题。由动量轮控制的左右平衡可以实现，但是由于驱动车轮部分的连接件是3d打印件，电机驱动车轮会发生打滑。因此目前还无法实现自平衡。

课程设计使用的主控为GD32F470（嘉立创梁山派），在PC平台（keil5）上进行开发，使用FreeRTOS实时操作系统进行多任务管理。

报告包括以下内容，

硬件部分：包括所使用硬件介绍、控制板pcb设计和自平衡独轮车结构设计。

软件部分：包括无刷电机控制、陀螺仪角度数据读取、FreeRTOS任务管理以及串级pid控制。

实验部分：包括单独的无刷电机控制和陀螺仪数据读取。

**关键词：**GD32、FreeRTOS、串级pid

目录

[第1章 自平衡独轮车硬件设计 2](#_Toc170048010)

[1.1 使用元件介绍 2](#_Toc170048011)

[1.1.1 GD32F470（嘉立创梁山派） 2](#_Toc170048012)

[1.1.2 直流无刷电机 3](#_Toc170048013)

[1.1.3 陀螺仪 3](#_Toc170048014)

[1.1.4 电源 3](#_Toc170048015)

[1.2 PCB控制板设计 3](#_Toc170048016)

[1.3 机械结构设计 4](#_Toc170048017)

[第2章 软件设计 5](#_Toc170048018)

[2.1 独立外设控制 5](#_Toc170048019)

[2.1.1 直流无刷电机控制 5](#_Toc170048020)

[2.1.2 陀螺仪读取 6](#_Toc170048021)

[2.1.3 USART与电脑通信（上位机显示） 7](#_Toc170048022)

[2.2 FreeRTOS任务管理 9](#_Toc170048023)

[2.2.1 sensorsTask传感器数据处理任务 9](#_Toc170048024)

[2.2.2 attitudeTask姿态控制任务 10](#_Toc170048025)

[2.2.3 平衡串级pid控制 11](#_Toc170048026)

[第3章 实验部分 13](#_Toc170048027)

[3.1 电机控制 13](#_Toc170048028)

[3.2 陀螺仪姿态读取 13](#_Toc170048029)

[总结体会 15](#_Toc170048030)

# 第1章 自平衡独轮车硬件设计

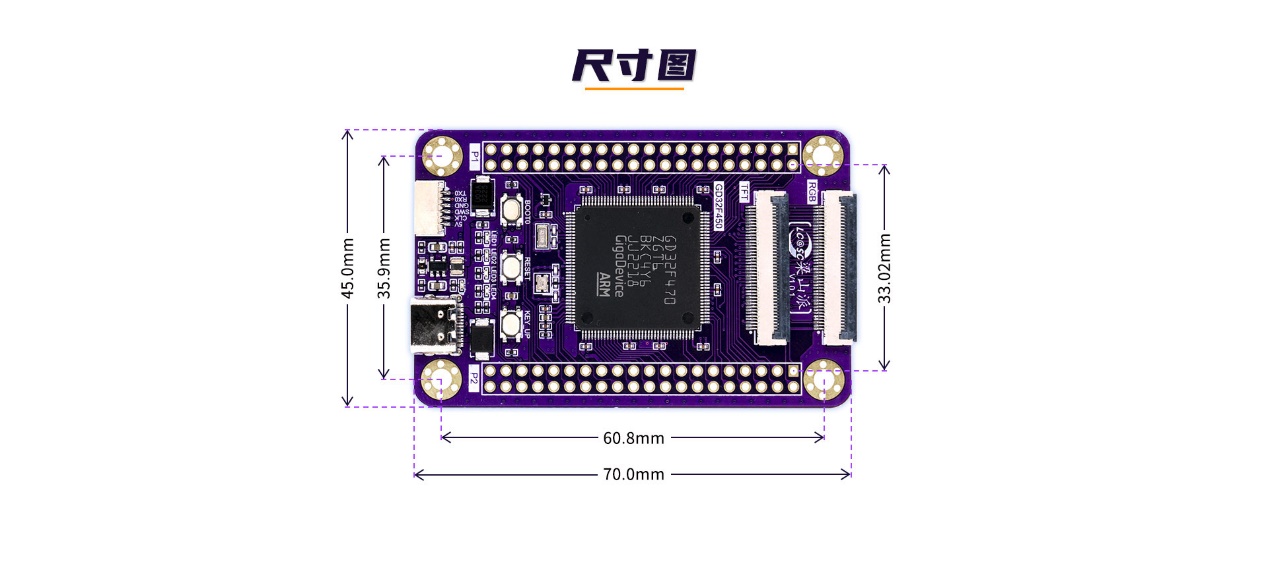
## 1.1 使用元件介绍

### 1.1.1 GD32F470（嘉立创梁山派）

考虑到价格和性能，在嘉立创梁山派上进行独轮车的开发。

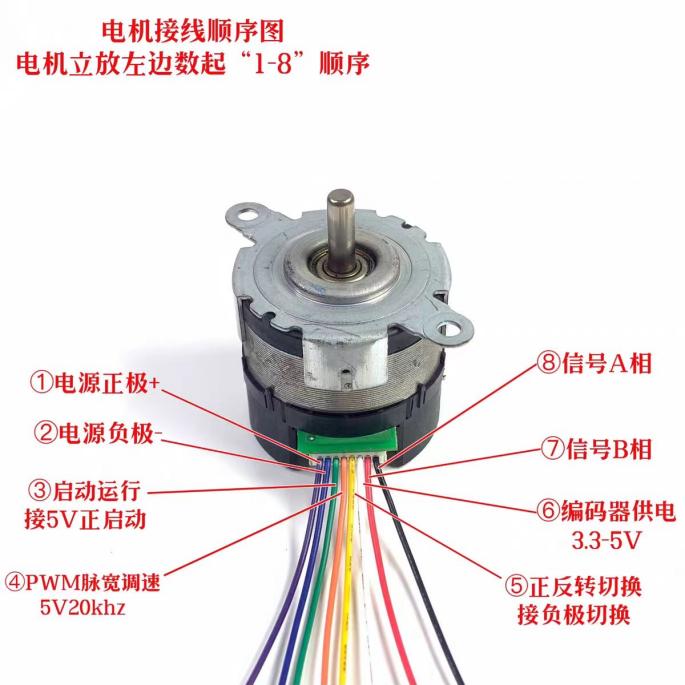
梁山派使用了国产的GD32F470ZGT6作为主控芯片。GD32F470ZGT6的硬件参数如下所示：

|  |  |
| --- | --- |
| **立创·梁山派开发板基本硬件参数** | |
| **CPU及Flash特性** | 高达240MHz Flash高达1024K  SRAM高达512K |
| **片内外设特性** | 高达3个12位ADC； 高达2个12位DAC； 高达8个通用16位定时器，2个基本定时器和2个增强型定时器； 高达6个SPI、3个I2C、4个USART、2个I2S、2个CAN、1个USBFS+HS、1个ETH、1个SDIO、1个Camera、1个IPA、1个EXMC/SDRAM |
| **板子特性** | 采用TYPE-C，USB提供5V电源 引出SWD&串口调试接口，配有DAPlink仿真器方便程序下载仿真与串口调试。 板载RGB565、8080并行接口并外扩SDRAM方便点屏。 板载USB2.0FS/HS。 引出68个可编程IO方便项目扩展。 |



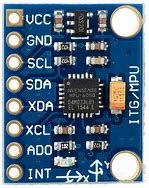
### 1.1.2 直流无刷电机

选用Nidec的直流无刷电机，额定电压24V，集成了100线的AB相编码器。



### 1.1.3 陀螺仪

选用MPU6050六轴角速度和加速度传感器。使用内置DMP模块进行姿态解算，通过软件I2C读取姿态欧拉角和角速度原始值。



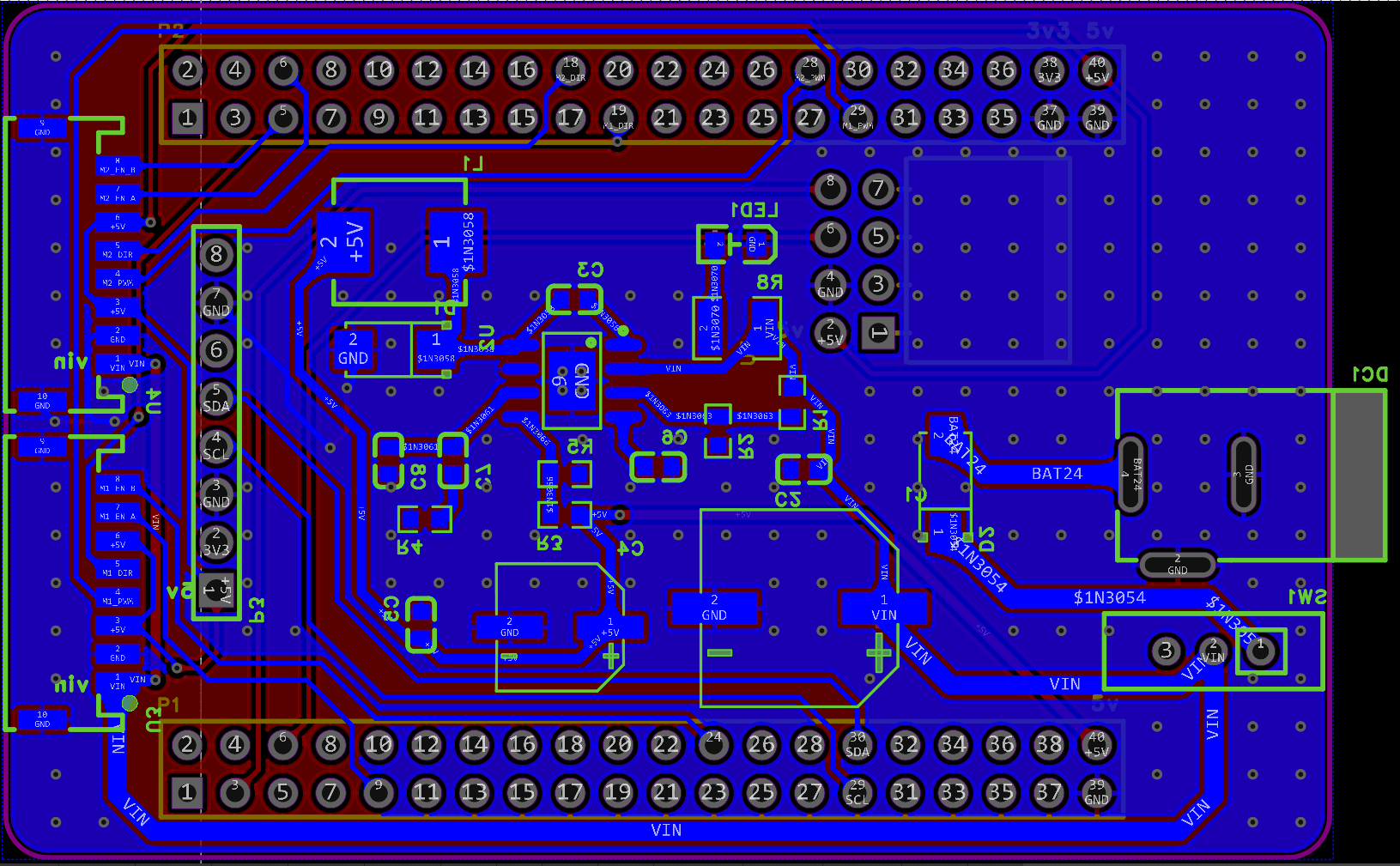
### 1.1.4 电源

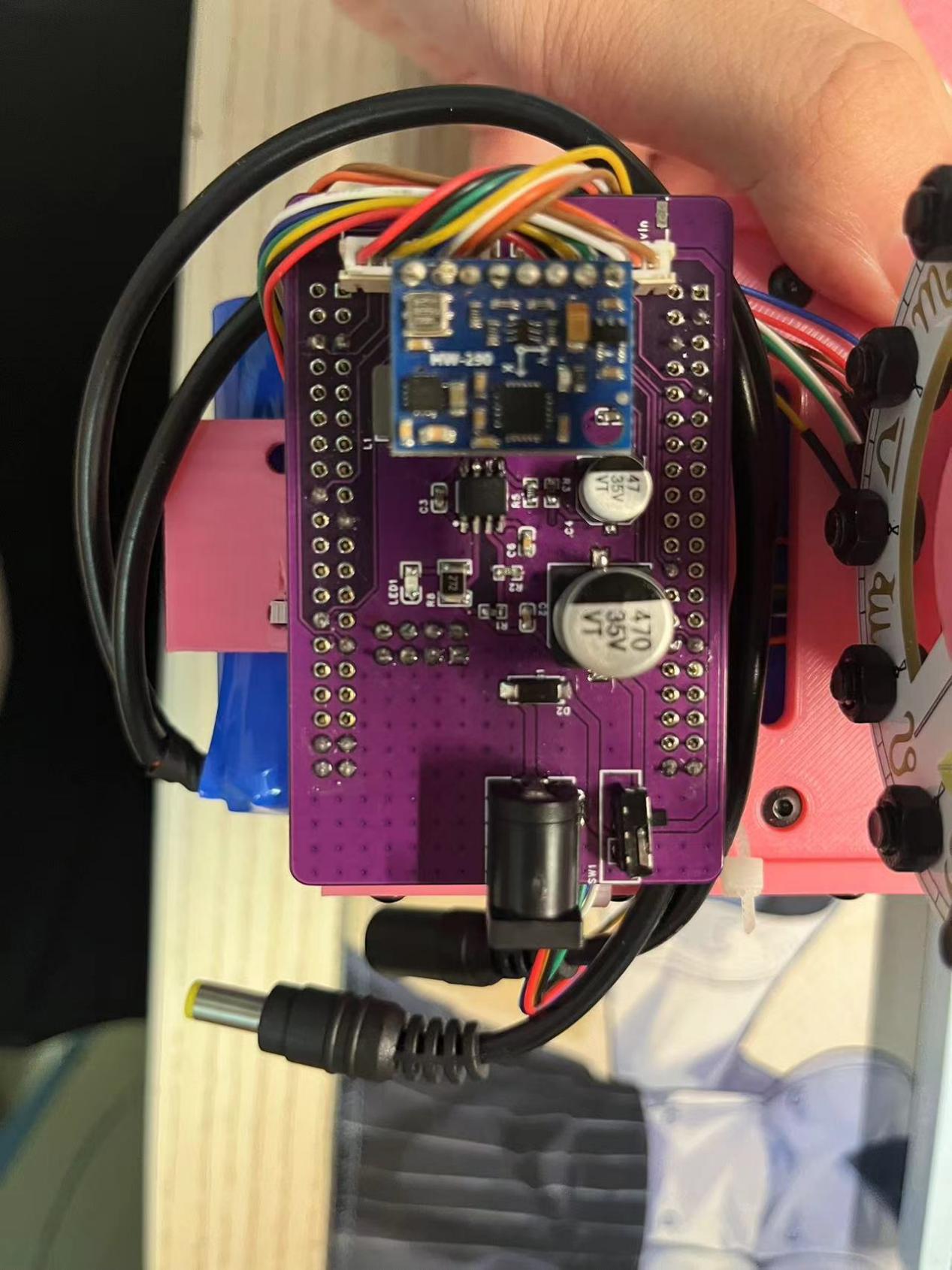
选用的24V电池，由6节18650锂电池串联而成，容量为2600mah，支持最大功率120W。

## 1.2 PCB控制板设计

设计的PCB控制板包含以下内容：

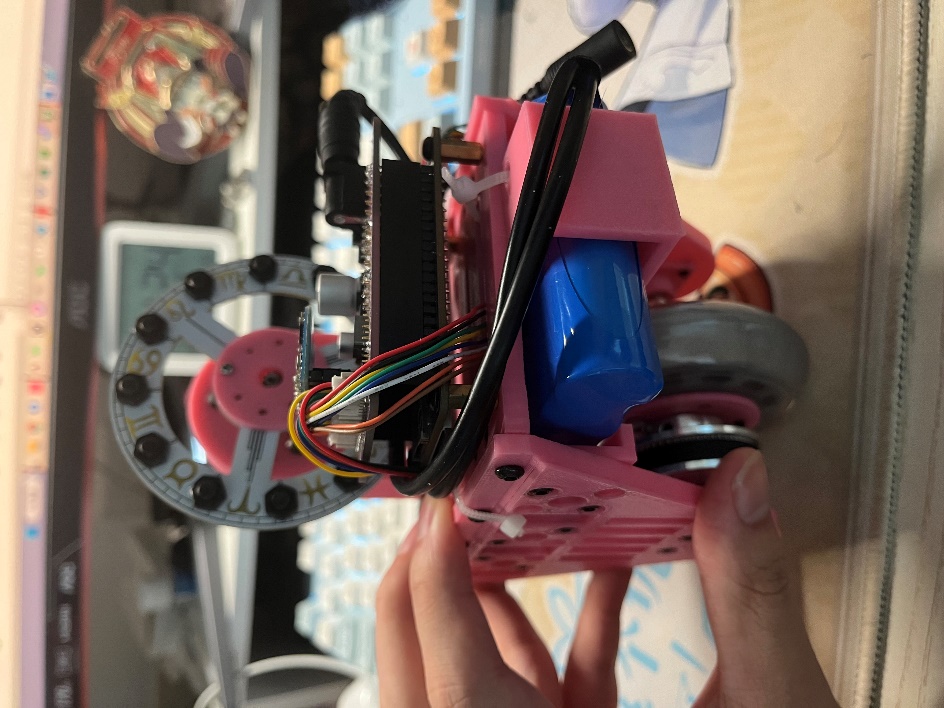
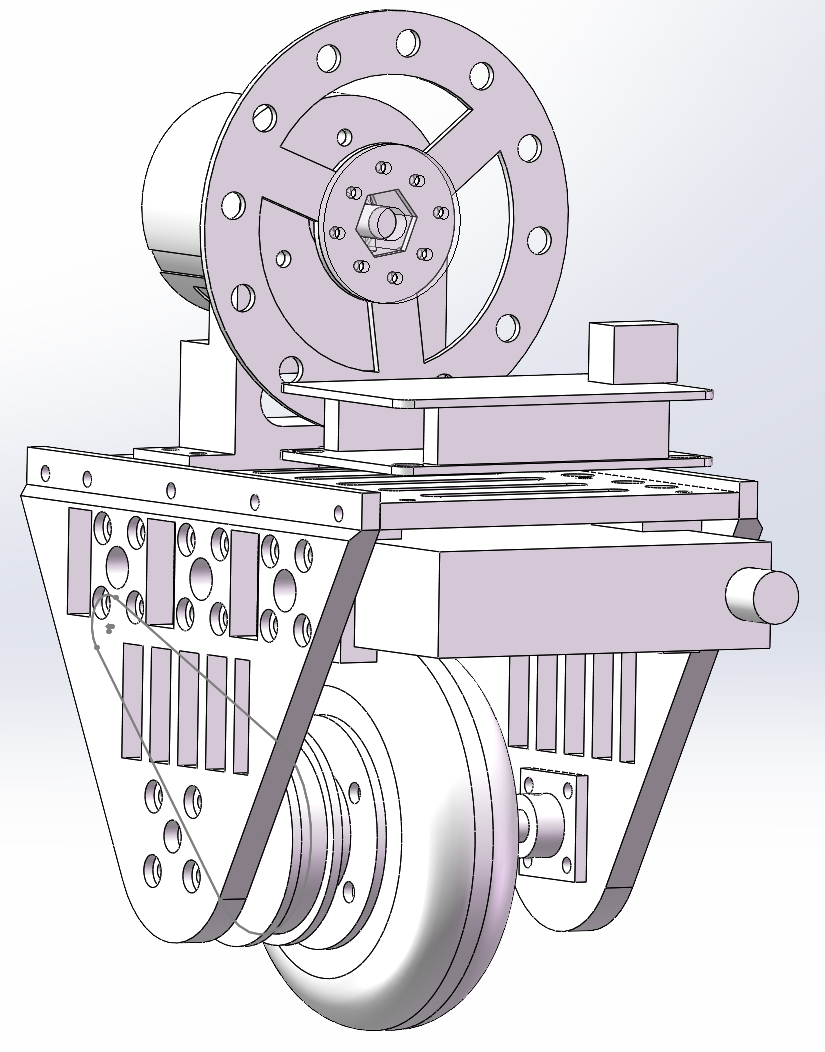
1. 供电部分：电源24V直接给电机供电，通过降压电路将电源电压降为5V给GD32供电。
2. 接口部分：通过2.54mm排插直接与GD32主控板连接，预留陀螺仪和2个无刷电机接口。





## 1.3 机械结构设计

在solidworks中对独轮车机械结构进行设计，相关结构件3D打印出来：



# 第2章 软件设计

## 2.1 独立外设控制

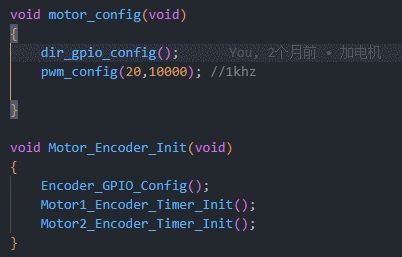
### 2.1.1 直流无刷电机控制

（1）电机转动底层控制

直流无刷电机控制方式如下（以单个电机为例）：

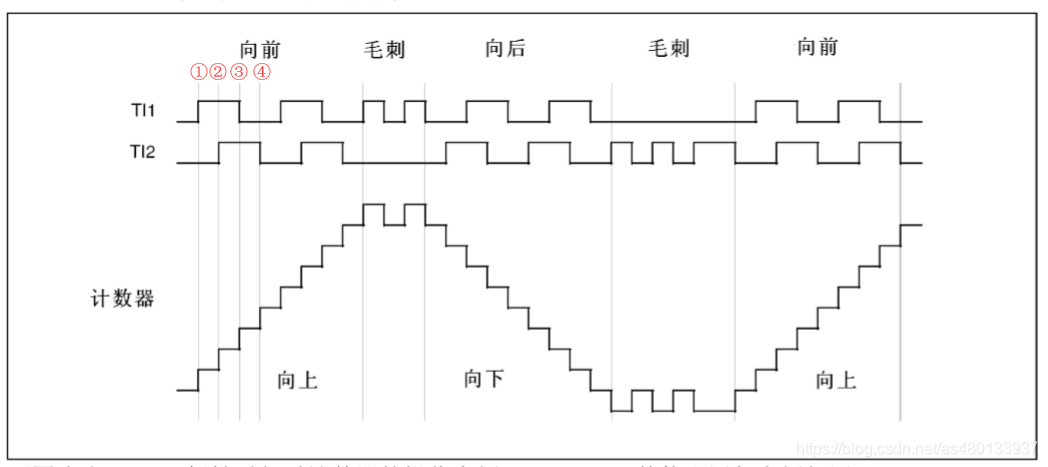
|  |  |  |
| --- | --- | --- |
| **电机控制线** | **控制方式** | |
| 启动运行（刹车线） | 直接接5V正极 | |
| PWM脉宽调速 | 定时器比较输出 | timer4，PA1 |
| 电机转动方向线 | GPIO输出 | PC4 |
| 编码器A相 | 定时器正交解码 | timer1\_CH0，PA15 |
| 编码器B相 | timer1\_CH1，PB3 |

使用电机前，首先对定时器和所使用的GPIO口进行初始化：



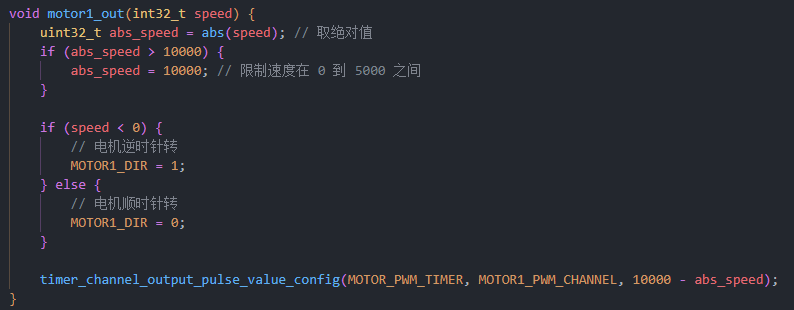
脉宽调速：设置timer4时钟预分频值20，计数重装值10000，得到输出频率1khz的PWM波；

编码器：设置timer1为解码模式2，通过检测编码器AB相上升沿和下降沿计数。一个脉冲信号会触发四次计数，设置定时器重装载值(100-1)\*4。



（2）电机转速pid闭环控制

首先对电机控制函数进行封装：



参数speed范围-10000-10000，speed符号正负对应电机正反转，speed绝对值大小控制电机转速快慢。

闭环控制函数中，需要对电机转速实际值进行计算，并将实际转速作为pid控制函数的输入，得到电机转速的输出.



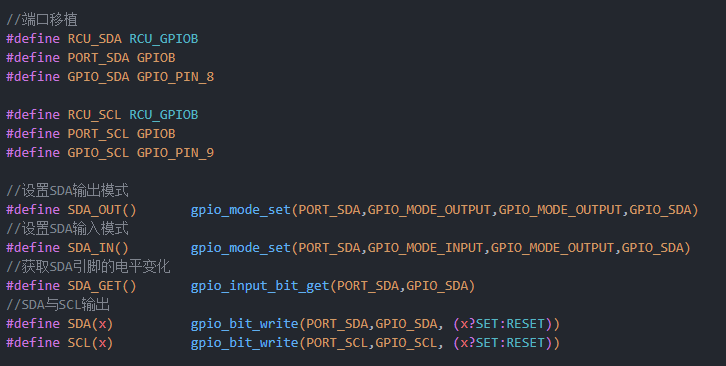
在电机测试任务中，对电机转速的闭环控制进行验证，控制频率为50hz：



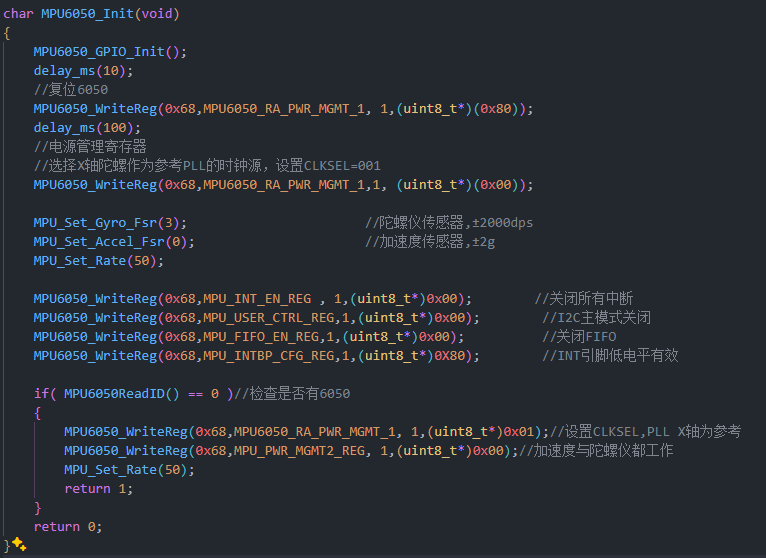
### 2.1.2 陀螺仪读取

GD32与MPU6050之间通过I2C进行通信。

在课程设计中，通过两个GPIO口的输入和输出作为I2C的SCL和SDA信号线，模拟I2C通信的时序来实现软件I2C。



使用传感器前，先对MPU6050进行初始化，设置角速度传感器量程为±2000dps，加速度传感器量程为±2g：



随后即可使用内置的DMP处理单元对姿态欧拉角进行读取。

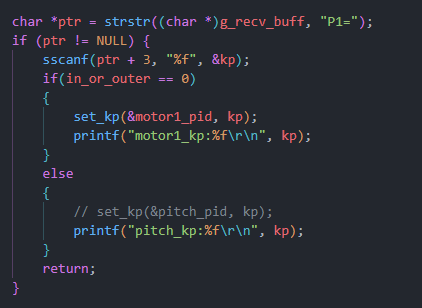
### 2.1.3 USART与电脑通信（上位机显示）

GD32通过串口通信将数据发送给电脑，可以进行方便的pid参数调试。

使用USART1，波特率设置为115200。GD32发送数据到电脑端之间通过printf函数即可；GD32接收电脑端数据需要在串口中断处理函数中进行：

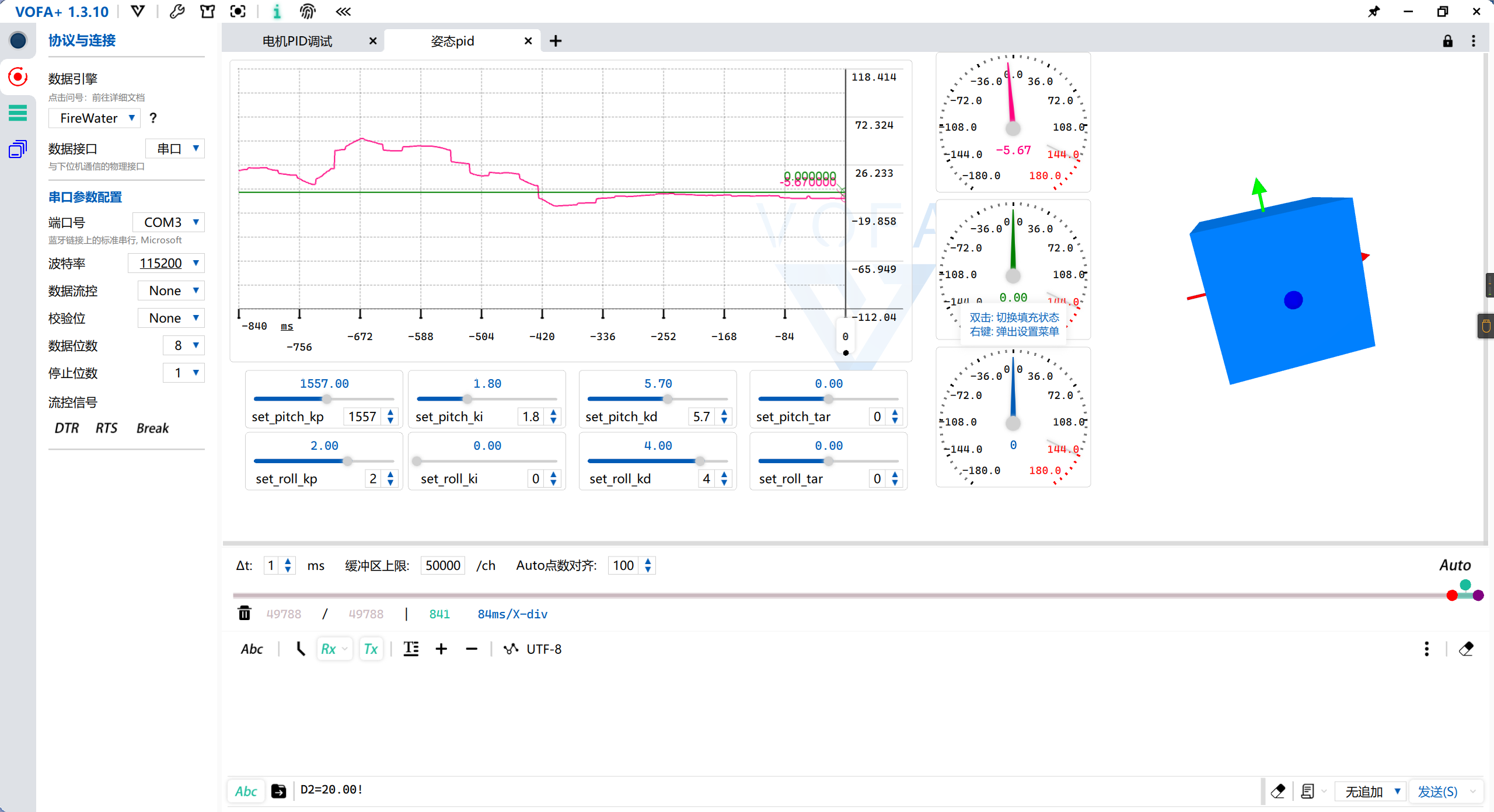


当GD32串口接收到数据时，接收中断标志不为0，进行接收数据的处理。首先将接收到的数据保存到缓冲区数组g\_recv\_buff。当数据传输完成，识别到结束标志位“!”，在Get\_Data()函数中对缓冲区数据进行处理：



上位机发送的数据格式为“P1=kp!”，将接收到的kp提取出来，并设置为相应的pid参数。

上位机软件选用了VOFA+，通信协议简单，可以满足需求。



## 2.2 FreeRTOS任务管理

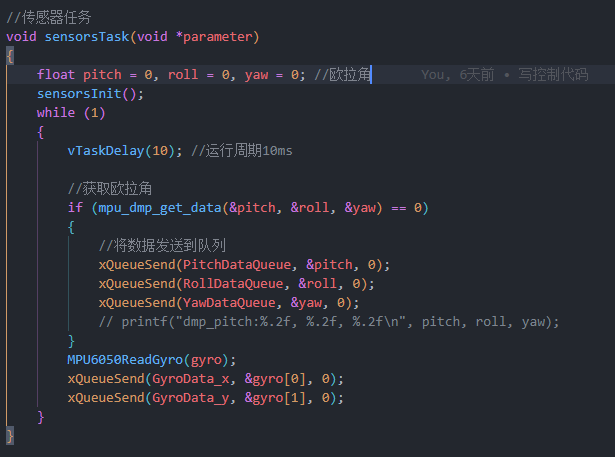
在课程设计中，共创建了3个任务。LED\_Task用于控制led灯闪烁，指示程序运行状态；attitudeTask用于控制独轮车姿态平衡控制；sensorsTask用于处理陀螺仪数据，得到姿态欧拉角。

所创建的3个任务遵循优先级越高，运行频率越高的原则。LED\_Task优先级最低，程序正常运行时，led指示灯每秒闪烁一次；attitudeTask和sensorsTask优先级一致，运行频率为100HZ。



### 2.2.1 sensorsTask传感器数据处理任务

首先对MPU6050进行初始化，在while(1)循环中，设置10ms延时，保证100hz运行频率，随后将获取到的姿态欧拉角和角速度值发送到创建的消息队列中。



### 2.2.2 attitudeTask姿态控制任务

首先对姿态控制的角度环和速度环pid参数和电机驱动进行初始化，在while(1)循环中，设置运行延时为5ms，通过定义时间戳tick来控制传感器数据获取、角度环计算、速度环计算的周期。



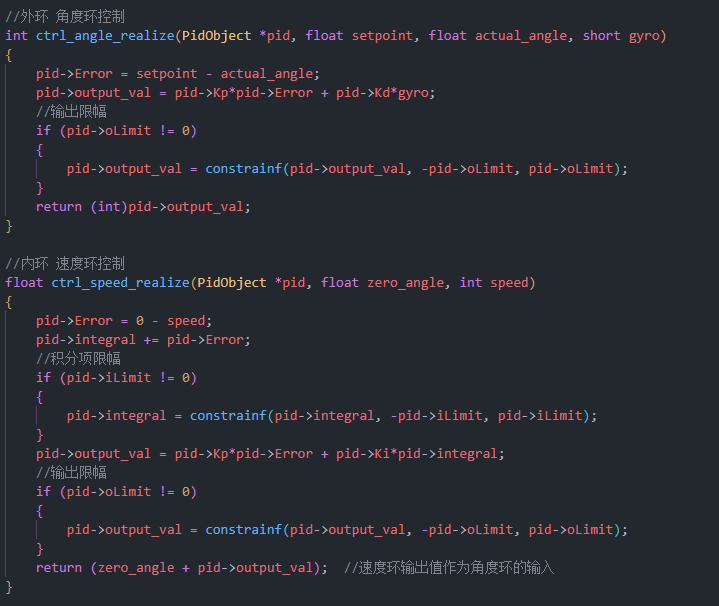
### 2.2.3 平衡串级pid控制

所使用的串级pid公式如下所示：

其中内环为动量轮电机速度环，外环为姿态角度环。为电机速度实际值，为机械零点，为姿态角度实际值，为计算得到的电机控制量。

以动量轮左右平衡为例，当独轮车向一侧偏移机械零点时，速度环输出增大角度环输出，从而得到更快的响应；当独轮车逐渐调整，靠近机械零点时，动量轮pwn输入减小使得speed变小，从而速度环输出减小。将内环输出看做增大外环传感器输入，从而增大外环响应速度。

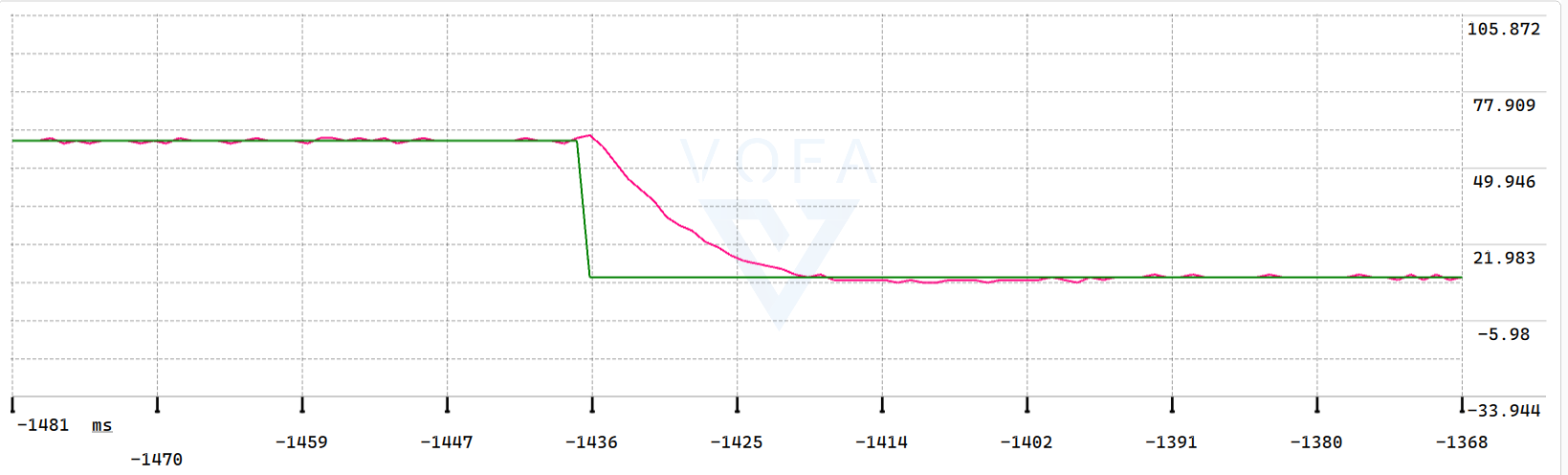
速度环控制中需要p、i环节，角度环控制中需要p、d环节。



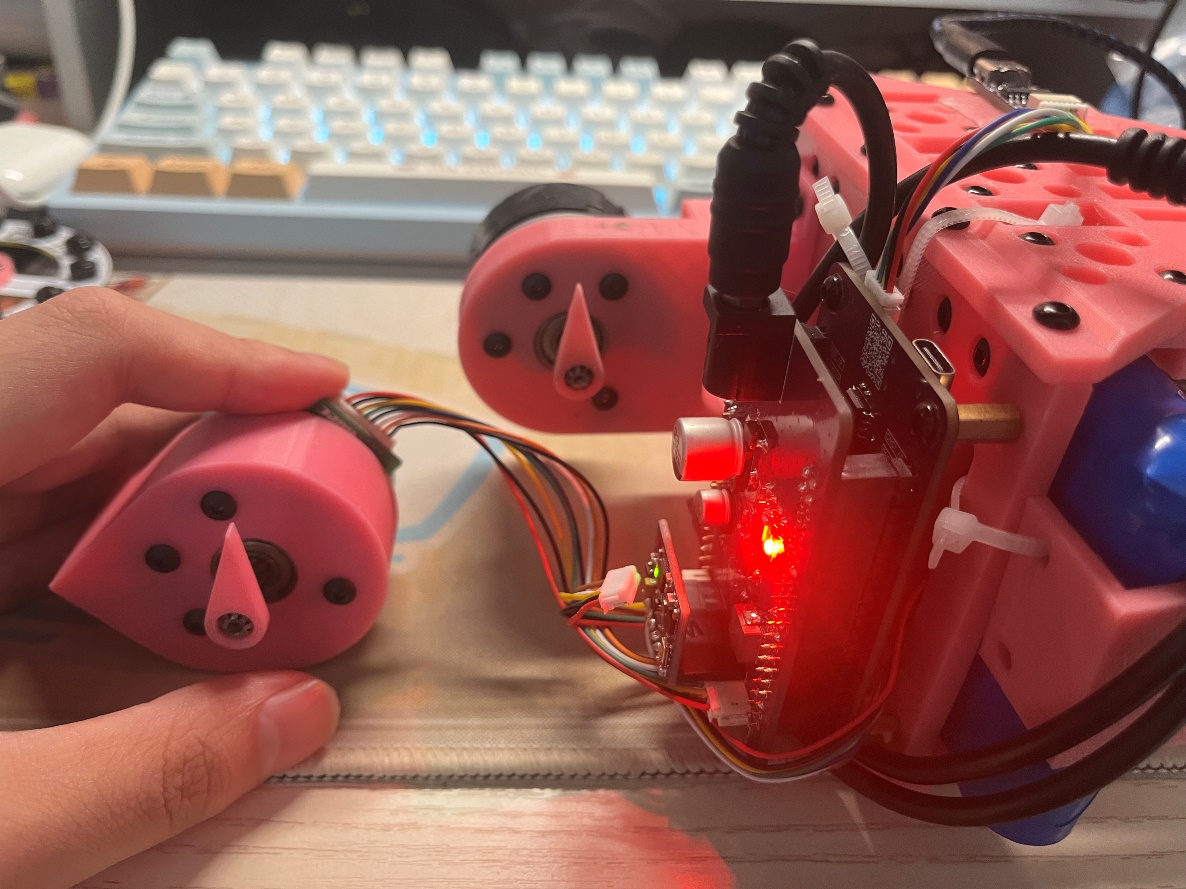
# 第3章 实验部分

### 3.1 电机控制

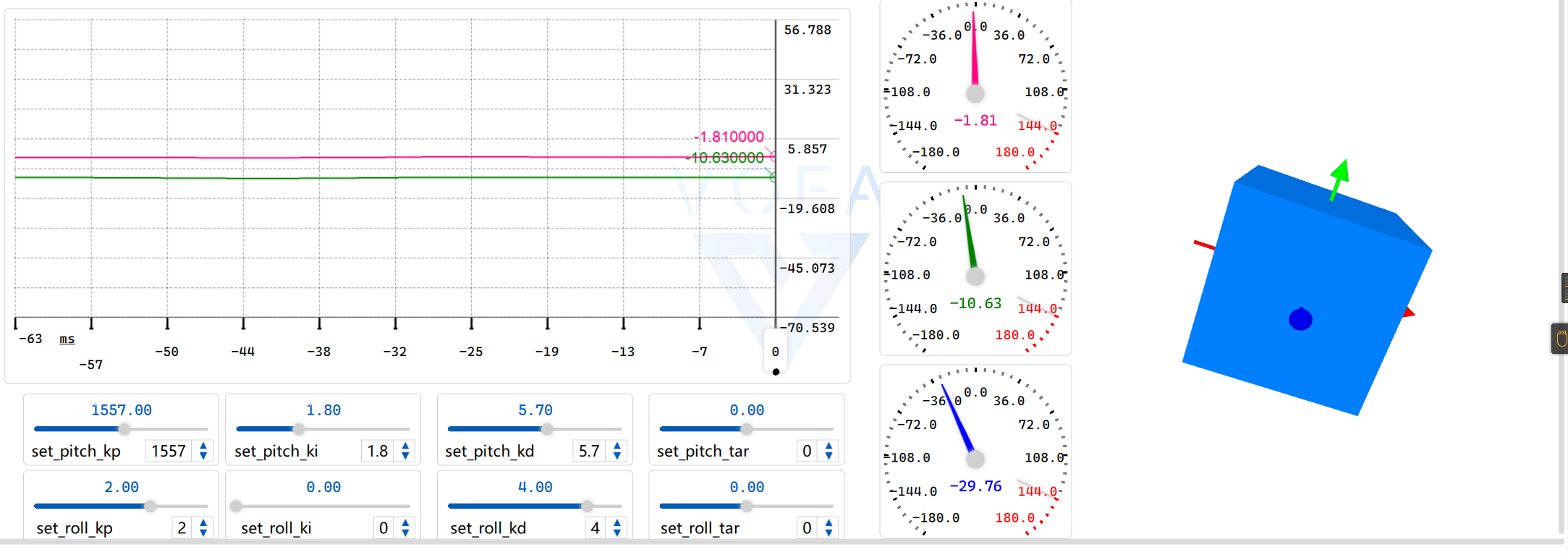
电机转速闭环控制：转速即为在电机测试任务中，一个周期内（20ms）电机编码器变化，设置期望值从60到10。Pid参数为：kp=7.0，ki=1.7，kd=0.4，可以观察到响应比较迅速。



双电机同步控制：



### 3.2 陀螺仪姿态读取



平衡控制所需要的roll和pitch角度都较为可以准确的读出，pitch俯仰角上有-10.5度的偏移误差。

# 总结体会

之前使用stm32进行单片机开发，使用的都是裸机编程，通过这学期的嵌入式系统课程，对操作系统有了更深的理解。虽然课上讲的的ucos-II，课程设计使用的是FreeRTOS，实际使用下来，感觉底层的逻辑是一致的。在移植FreeRTOS过程中遇到的问题，比如FreeRTOS配置文件FreeRTOSConfig.h中的一些内容，因为不太理解，所以刚开始系统不能正常运行，通过听课得到了解决。

画板子过程中使用嘉立创EDA，用起来还是很方便的，同时每个月可以免费打板，对做点小项目非常友好。

目前自平衡控制还实现不了，使用串级pid对动量轮进行控制，在左右平衡上可以坚持1-2s，之后对模型再进行改善，将驱动车轮部分的连接件换成金属的，再进行自平衡调试。