梯队队员第一次培训

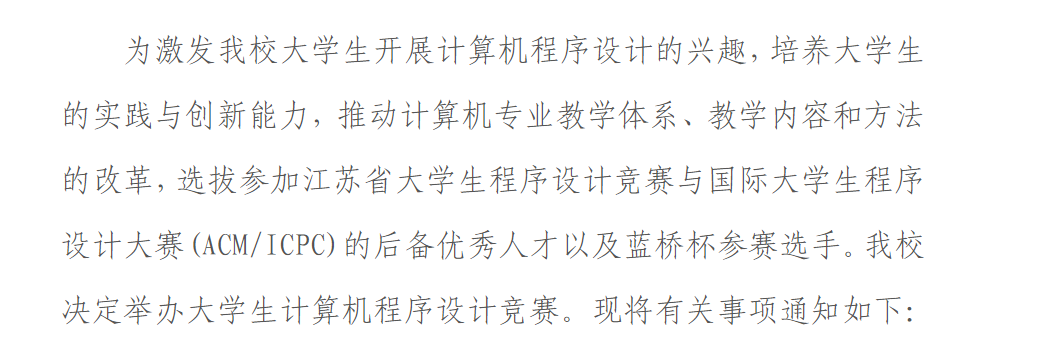
1. 梯队队员考核

首先是周期任务，第二次任务完成情况不好，为此，我组织了一场培训，但是没有完成的还得补上，第三次任务截止12.20号，具体任务为使用can通信控制3508或者6020电机并且收到反馈信息，第二次任务有些人不明白具体要做啥，考虑到第三次任务，任务改为遥控器发送数据，单片机dbus接收数据，使用遥控器串口通信控制电机。

以下为任务完成情况，有问题提出来。

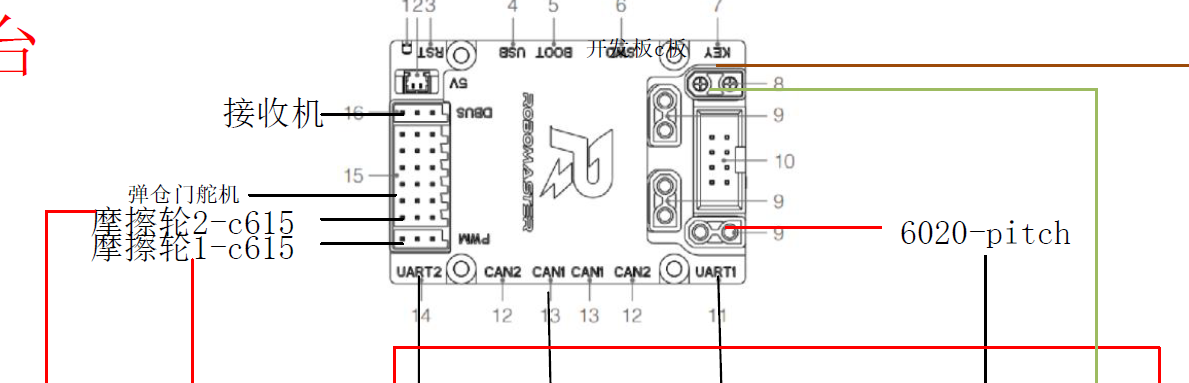


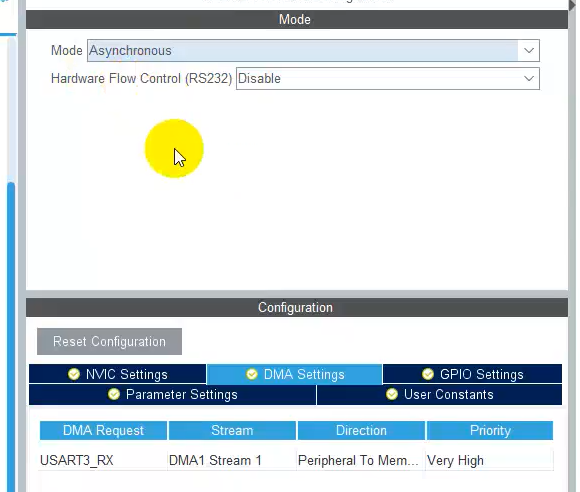
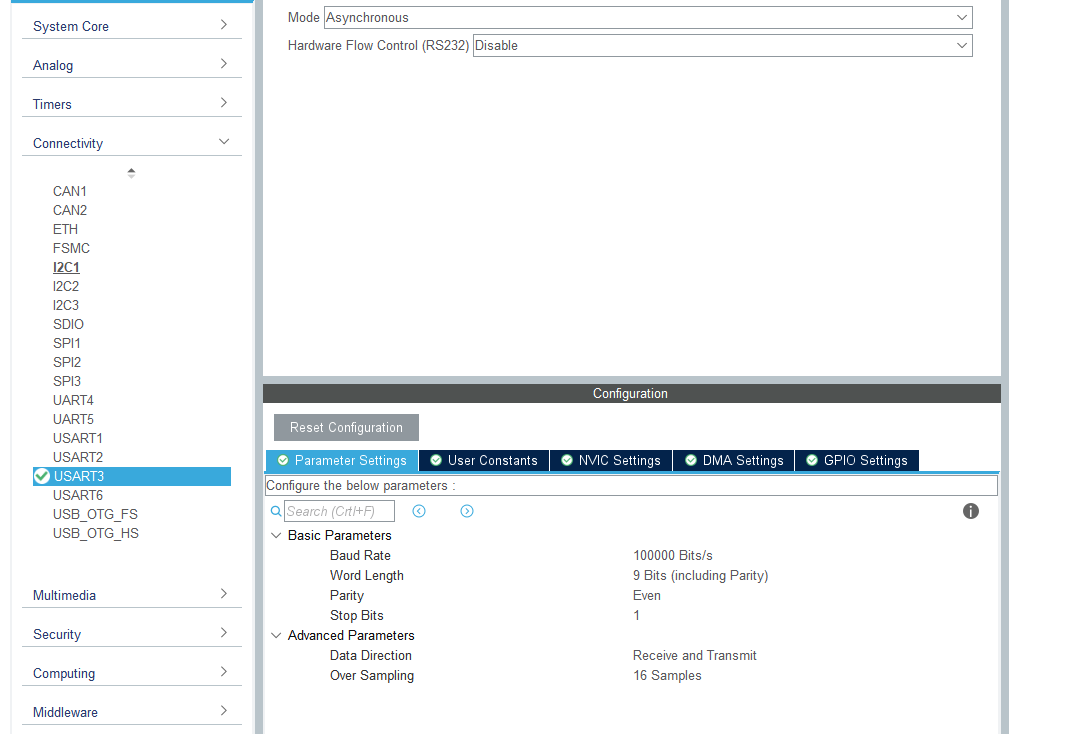
还有一个江苏大学程序设计大赛，个人赛，大家尽量参加，c语言是基础。



1. 串口通信

中断DMA+遥控器



## ②代码

main.c

/\* USER CODE BEGIN 2 \*/  
 HAL\_UART\_Receive\_DMA(&huart3,sbus\_rx\_buffer,18);  
 //HAL\_TIM\_PWM\_Start(&htim1,TIM\_CHANNEL\_1);  
 /\* USER CODE END 2 \*/

/\* USER CODE BEGIN 4 \*/  
void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*UartHandle)  
{  
 RC\_Ctl.rc.ch0 = (sbus\_rx\_buffer[0] | (sbus\_rx\_buffer[1] << 8))& 0x07ff; //确保11位  
 RC\_Ctl.rc.ch1 = ((sbus\_rx\_buffer[1] >> 3) | (sbus\_rx\_buffer[2] << 5))& 0x07ff;  
 RC\_Ctl.rc.ch2 = ((sbus\_rx\_buffer[2] >> 6) | (sbus\_rx\_buffer[3] << 2) | (sbus\_rx\_buffer[4] << 10))& 0x07ff;  
 RC\_Ctl.rc.ch3 = ((sbus\_rx\_buffer[4] >> 1) | (sbus\_rx\_buffer[5] << 7))& 0x07ff;  
 RC\_Ctl.rc.s1 = ((sbus\_rx\_buffer[5] >> 4) & 0x000C) >> 2;  
 RC\_Ctl.rc.s2 = ((sbus\_rx\_buffer[5] >> 4) & 0x0003);  
   
  
 RC\_Ctl.mouse.x = sbus\_rx\_buffer[6] | (sbus\_rx\_buffer[7] << 8);  
 RC\_Ctl.mouse.y = sbus\_rx\_buffer[8] | (sbus\_rx\_buffer[9] << 8);  
 RC\_Ctl.mouse.z = sbus\_rx\_buffer[10] | (sbus\_rx\_buffer[11] << 8);  
 RC\_Ctl.mouse.press\_l = sbus\_rx\_buffer[12];  
 RC\_Ctl.mouse.press\_r = sbus\_rx\_buffer[13];  
 RC\_Ctl.key.v = sbus\_rx\_buffer[14] | (sbus\_rx\_buffer[15] << 8);  
  
}  
/\* USER CODE END 4 \*/

main.h

/\* USER CODE BEGIN ET \*/  
typedef struct  
{  
 struct  
 {  
 unsigned short ch0;  
 unsigned short ch1;  
 unsigned short ch2;  
 unsigned short ch3;  
 unsigned char s1;   
 unsigned char s2;  
 }rc;  
   
  
 struct  
 {  
 unsigned short x;  
 unsigned short y;  
 unsigned short z;  
 unsigned char press\_l;   
 unsigned char press\_r;  
 }mouse;  
   
 struct  
 {  
 unsigned short v;  
 }key;  
  
}RC\_Ctl\_t;  
/\* USER CODE END ET \*/

# 2串口

## ①无中断 无dma 轮询模式

HAL\_UART\_Transmit(&huart1,(uint8\_t\*)"hello",5,0xFFFF);  
HAL\_Delay(1000);

收发

uint8\_t buf[5];  
 HAL\_UART\_Receive(&huart1,buf,3,0xFFFF);  
 HAL\_UART\_Transmit(&huart1,buf,3,0xFFFF);

## ②中断

/\* USER CODE BEGIN PV \*/  
uint8\_t buffer[5];  
/\* USER CODE END PV \*/

/\* USER CODE BEGIN 0 \*/  
void HAL\_UART\_TxCpltCallback(UART\_HandleTypeDef \*huart)  
{  
   
}  
void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*huart)  
{  
 HAL\_UART\_Transmit\_IT(&huart1,buffer,3);  
 HAL\_UART\_Receive\_IT(&huart1,buffer,3);  
}  
/\* USER CODE END 0 \*/

/\* USER CODE BEGIN 2 \*/  
 HAL\_UART\_Receive\_IT(&huart1,buffer,3);  
 /\* USER CODE END 2 \*/

## ③DMA

1.IT改为DMA

2.空闲中断 不定长接收

void USART1\_IRQHandler(void)  
{  
 /\* USER CODE BEGIN USART1\_IRQn 0 \*/  
 if(\_\_HAL\_UART\_GET\_FLAG(&huart1,UART\_FLAG\_IDLE) != RESET)  
 {  
 \_\_HAL\_UART\_CLEAR\_IDLEFLAG(&huart1);  
 HAL\_UART\_DMAStop(&huart1);  
 uint8\_t len = 1024-\_\_HAL\_DMA\_GET\_COUNTER(huart1.hdmarx);  
 HAL\_UART\_Transmit\_DMA(&huart1,buffer,len );  
 HAL\_UART\_Receive\_DMA(&huart1,buffer,1024 );  
 }  
 /\* USER CODE END USART1\_IRQn 0 \*/  
 HAL\_UART\_IRQHandler(&huart1);  
 /\* USER CODE BEGIN USART1\_IRQn 1 \*/  
  
 /\* USER CODE END USART1\_IRQn 1 \*/  
}

main.c

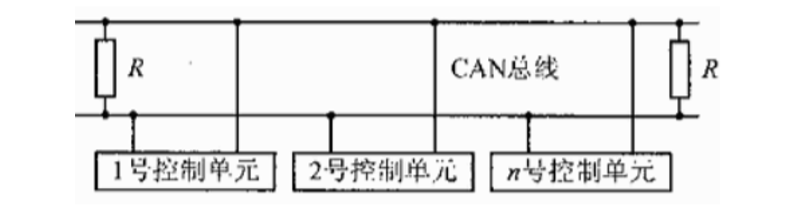
HAL\_UART\_Receive\_DMA(&huart1,buffer,1024);  
 \_\_HAL\_UART\_ENABLE\_IT(&huart1,UART\_IT\_IDLE);

3.can总线

CAN 是控制器域网 (Controller Area Network, CAN) 的简称，是由研发和生产汽车电子产 品著称的德国 BOACH 公司开发，并最终成为国际标准（ISO11898），CAN 是国际上应用

最广泛的现场总线之一。 在北美和西欧，CAN 总线协议已经成为汽车计算机控制系统和嵌 入式工业控制局域网的标准总线，并且拥有以 CAN 为底层协议专为大型货车和重工机械车 辆设计的 J1939 协议。

CAN 总线由 CAN\_H 和 CAN\_L 两根线构成，各个设备一起挂载在总线上。



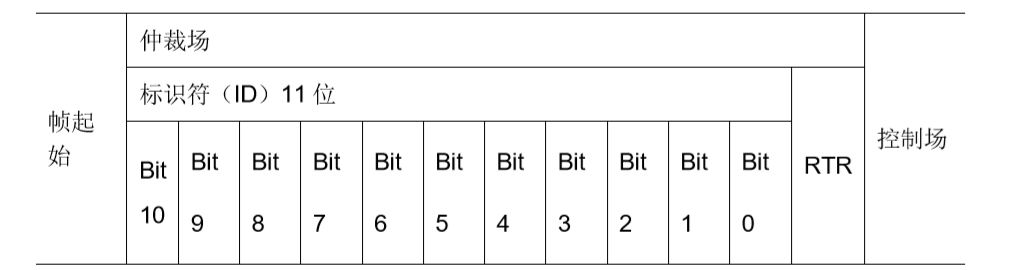
RoboMaster 系列电机也采用 CAN 协议进行通信，CAN 协议比较复杂，一个完整的数据帧

由下图中的各个部分组成：



这里重点介绍 CAN 的仲裁场和数据场的内容。每一个挂载在 CAN 总线 上的 CAN 都有一个自己独属的 ID，每当一个设备发送一帧数据时，总线其他设备会检查这个 ID 是否是自己需要接收数据的对象，如果是则接收本帧数据，如果不是则忽略。

ID 存储在数据帧最前头的仲裁场内，CAN 的 ID 分为标准 ID 和拓展 ID 两类，标准 ID 长度 为 11 位



在通过 ID 判断本帧数据可以接收后，控制场中的 DLC 规定了本帧数据的长度，而数据场内 的数据的大小为 8 Byte，即 8 个 8 位数据。CAN 总线的一个数据帧中所需要传输的有效数 据实际上就是这 8Byte。



# C620电调的CAN通信

在配套的C620手册中找到CAN协议描述



这是电调接收报文格式，即如果要发送数据给 1 号到 4 号电调，控制电机的输出电流，从而 控制电机转速时，需要按照表中的内容，将发送的 CAN 数据帧的 ID 设置为 0x200，数据域 中的 8Byte 数据按照电调 1 到 4 的高八位和第八位的顺序装填，帧格式和 DLC 也按照表中内容进行设置，最后进行数据的发送

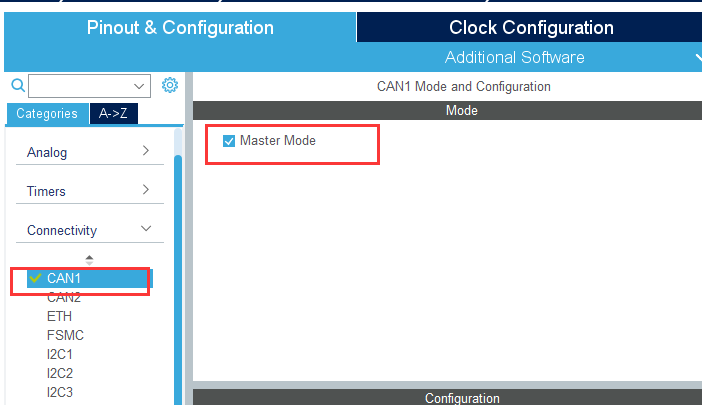
下表是C620向32发送的报文，32需要按下面的格式进行解析。

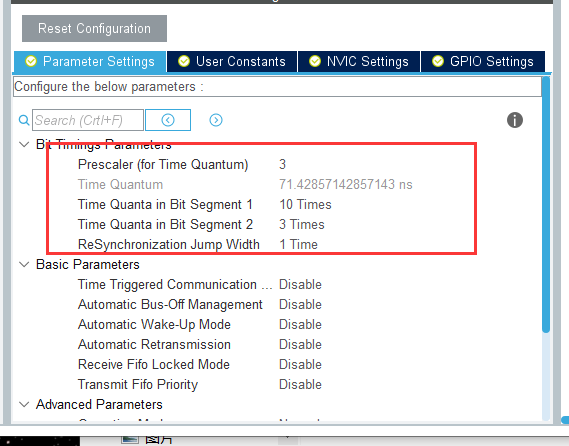


可以发现按这样的格式发送，一帧完整的CAN报文就完全可以将所需要的数据发送出去，不许将数据拆分成许多不同的数据帧，对比遥控器的串口每次只能发送一个字节的数据，一帧完整的数据需要发送18个字节，减少了数据拼接时丢帧的可能。

首先根据接收到的 ID 判断究竟接收到的是哪个电调发送来的数据，手册中规定 1 号电调 ID 为 0x201，2 号为 0x202，3 号为 0x203，4 号为 0x204。判断完数据来源之后，就可以按照手册中的数据格式进行解码，通过高八位和第八位拼接的方式，得到电机的转子机械角度，转子转速，转矩电流，电机温度等数据。

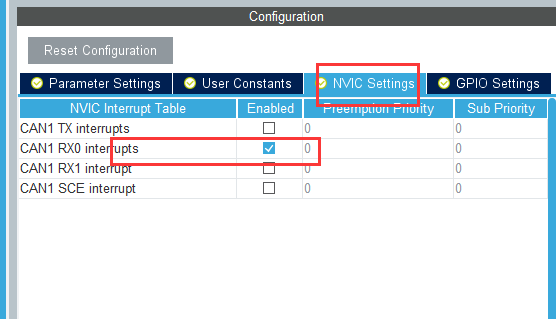
# CAN 在 cubeMX 中的配置





CAN的通信速率按照C620手册要求的1Mbps配置

71.42857142857143𝑛𝑠 ∗(10 +3+1) = 1000𝑛𝑠 = 1𝑢𝑠=1mbps



开启对应的CAN接收中断