关于遥控器和NUC掉线

2022年1月26日-27日 电控组 陈卓勋

1. 背景

遥控器掉线，但是数据依然没有被改变，是上一次解析时的数据。在遥控器离线时，机器人会失控。对于自动模式，NUC数据控制机器人云台时，这个问题更加可怕。

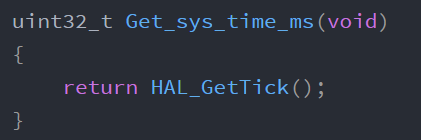
1. 解决思路

在遥控器数据解析和NUC数据解析相关的C源文件内，分别使用一个变量记录上一次解析数据的时间。并且提供接口供别的任务获取这个时间。我们新建一个监视任务，这个任务1000ms运行一次，获取当前时间和数据包上一次解析时间。如果时间差超过2000ms，就判断掉线了，使用函数接口将遥控器数据或者NUC数据设置为无效。在这个机制下，每次在中断来临且成功解析后，需要更新当前解析时间，解析完成后还需要使用函数接口将数据设置为有效。

1. 代码实现

（1）关于获取系统时间：[(17条消息) STM32使用FREERTOS获取系统时间\_xian\_z的博客-CSDN博客\_freertos 获取系统时间](https://blog.csdn.net/xian_z/article/details/113044362)

这个时间最大为49天，远大于比赛时间。

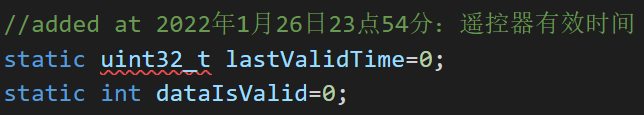


获得以ms为单位、从开机到现在经过的时间。

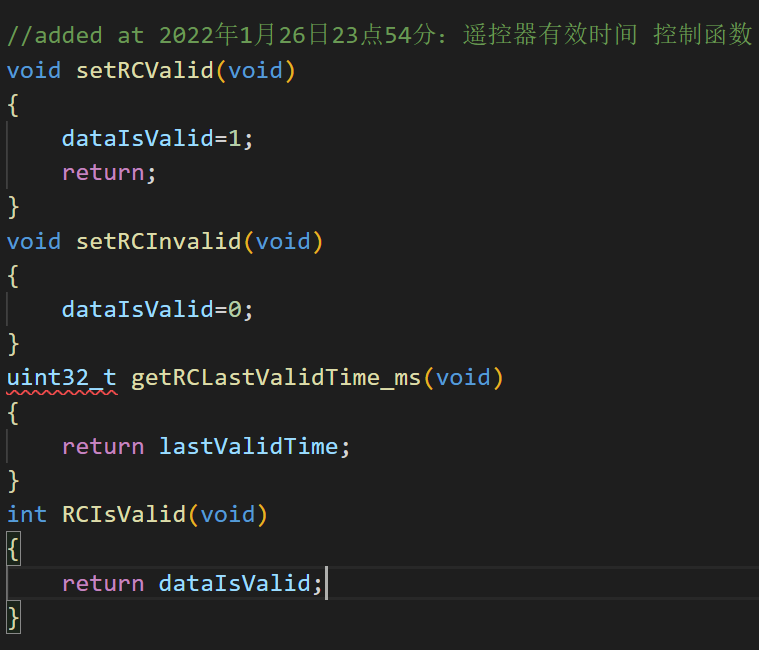
不妨把它的声明写在monitor\_task.h里面

（2）更改遥控器数据解析相关函数。

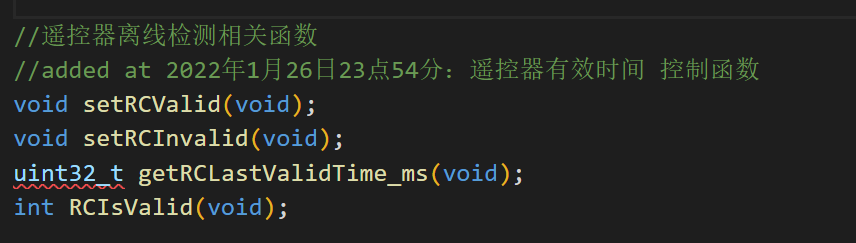
a) 创建变量 static int dataIsValid=0, static uint32 lastValidTime\_ms用来记录是否有效和最后一次时间。



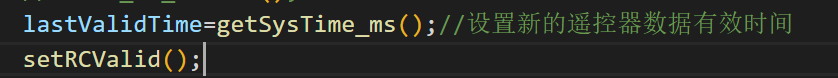
b) 在c文件中增加 setRCValid函数、setRCInvalid函数、RCIsValid函数、getLastValidTime\_ms函数。



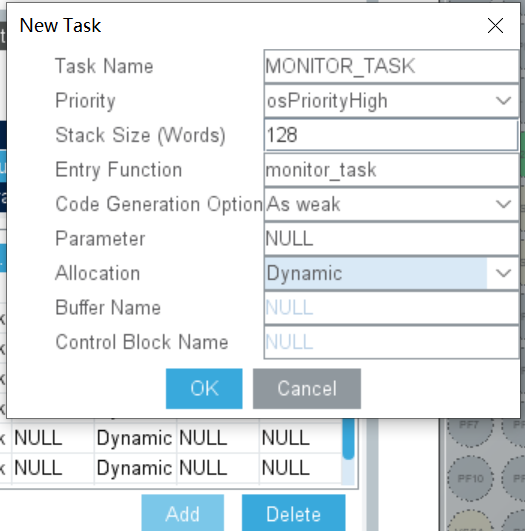
c) 在h文件中增加这三个函数的声明



d) 在遥控器数据从串口缓冲区到遥控器命令解析函数中增加setRCValid和改变lastRCValidTime\_ms

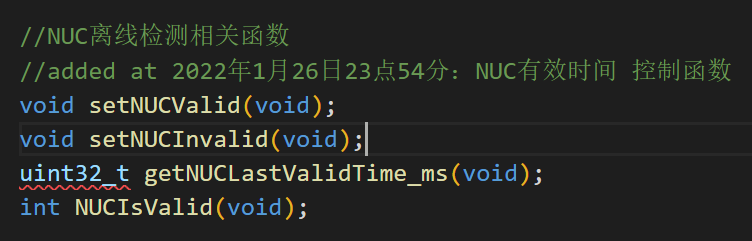


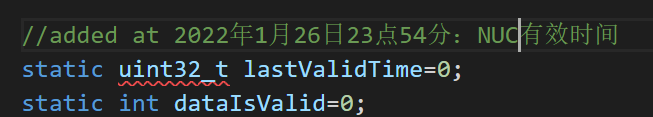
（3）新建一个监视任务，在cubeMX里设置。



（4） 更改底盘、云台等机构的控制逻辑，让运动机构在遥控器掉线时停止运动。在NUC掉线时立即自动切换到手动模式。

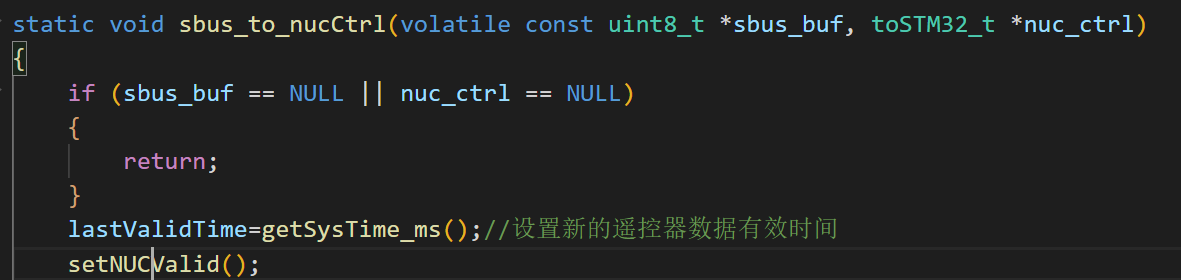
（5） 编写监视任务函数，代码里，1000ms执行一次，判断当前时间超过上次时间2000ms及以上时，设置数据无效。





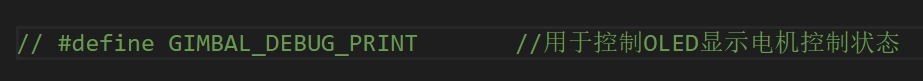






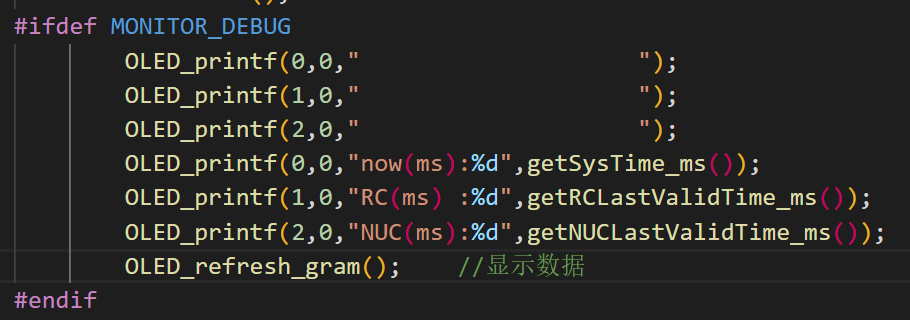
1. 模块测试方法
2. 单独测试检测模块在OLED屏幕上输出当前时间、遥控器、NUC上一次有效的时间。

（为了避免其他模块的干扰，我们关闭云台OLED debug）



在任务里新增





就可以用OLED测试了。

最终的代码：（哇，这次发现，如果直接复制代码而不是截图过来，发现就可以让大家直接复制粘贴代码进行测试了。好耶，那以后就都复制代码吧！

#include "monitor\_task.h"

#include "cmsis\_os.h"

#include "remoteControl.h"  //用来加载数据的上一次有效时间和设置无效

#include "nucCommu.h"       //用来加载数据的上一次有效时间和设置无效

#include "OLED.h"

#define MONITOR\_DEBUG       //控制是否开启monitor\_task的debug

#define CHECK\_TIME 1000     //一秒钟都没有更新数据：认为掉线了。时间阈值 单位ms

uint32\_t getSysTime\_ms(void)

{

    return HAL\_GetTick();

}

void monitor\_task(void)

{

    while (1)

    {

        if(getSysTime\_ms()-getRCLastValidTime\_ms()>CHECK\_TIME)//一秒钟都没有更新数据：认为遥控器掉线了

            setRCInvalid();//设置遥控器数据无效

        //对于NUC数据同样处理

        if(getSysTime\_ms()-getNUCLastValidTime\_ms()>CHECK\_TIME)

            setNUCInvalid();

        #ifdef MONITOR\_DEBUG

                OLED\_printf(0,0,"                    ");

                OLED\_printf(1,0,"                    ");

                OLED\_printf(2,0,"                    ");

                OLED\_printf(0,0,"now(ms):%d",getSysTime\_ms());

                OLED\_printf(1,0,"RC(ms) :%d",getRCLastValidTime\_ms());

                OLED\_printf(2,0,"NUC(ms):%d",getNUCLastValidTime\_ms());

                OLED\_refresh\_gram();    //显示数据

        #endif

        osDelay(1000);//这个时间足够长，可以输出后就让OLED刷新

    }

}

1. 与底盘控制逻辑联合测试。

（首先需要修改底盘控制模式）

1. 结果
2. OLED屏幕显示结果

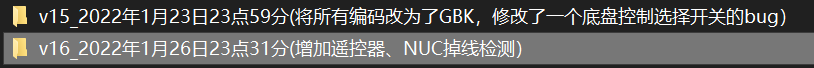
|  |  |
| --- | --- |
| 上电，不开启遥控器 | 开启遥控器 |
| 关闭遥控器 | 再次开启遥控器 |

1. 掉线控制测试结果

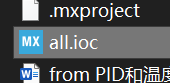
（先设计底盘控制模式，小陀螺（通过宏配置为准小陀螺）、跟随云台、不跟随（测量旋转圈数防止线断裂）等）

附：操作过程

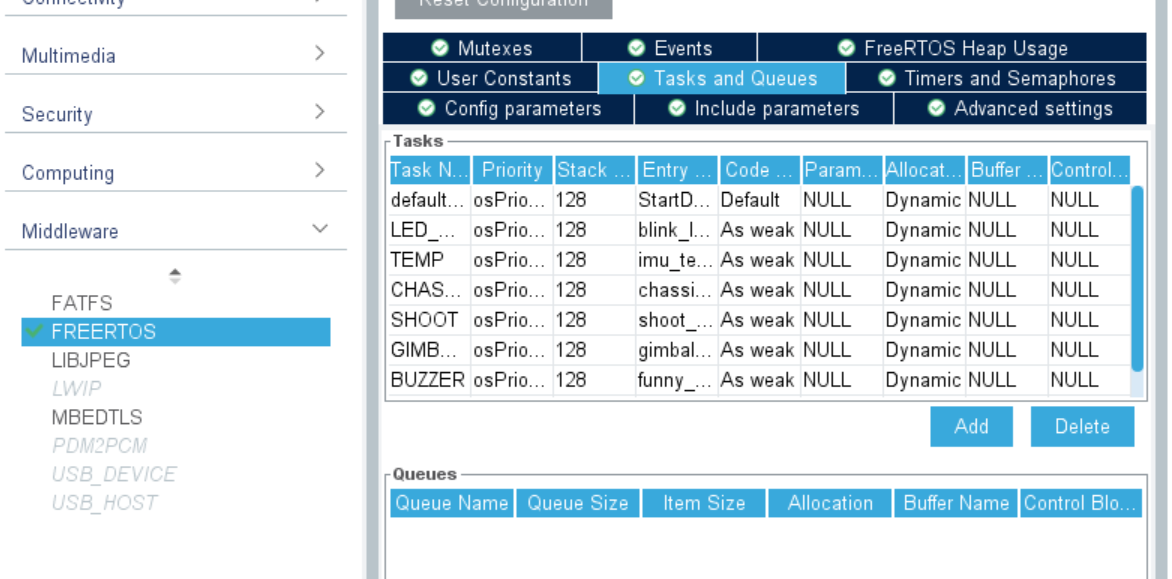
复制一份程序，作为新版本

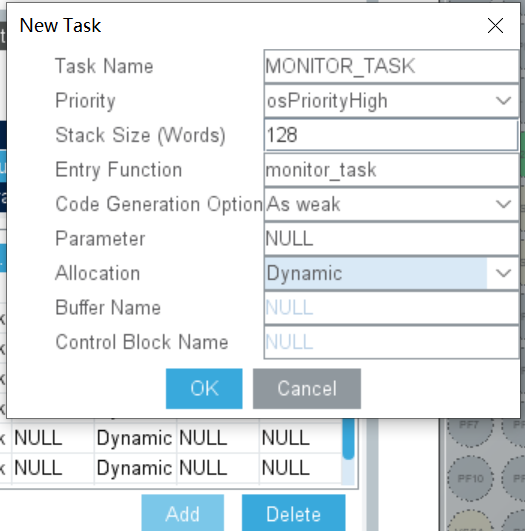


打开新版本的cubemx文件



点击add添加任务



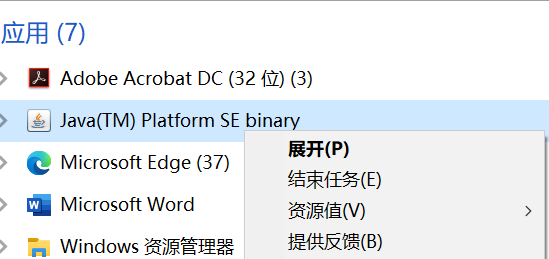


生成工程文件。

中途cubeMX挂了。

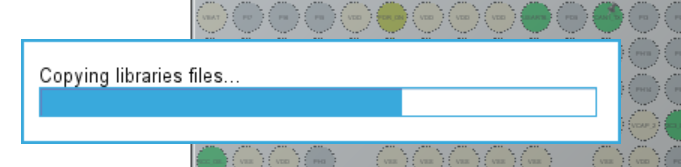
不管，结束任务，重新打开并生成。

结束的是java这个哈



第二次还是崩了。

发现不对劲，每次都会崩掉。Copy 不下去



即便管理员身份也不行。

好吧，不管了，直接写代码。

记得在μvision工程里添加源文件和头文件路径

（最后测试OLED输出发现能正常工作，好耶）