滤除控制量。根据两次采样的差值合理性判断。当两次ECD采样差距过大时，说明出现了过中央零的情况。根据差值的正负判断旋转的方向

//计算ECD单位下的角速度。ECD/ms

        // 需要对反馈量进行过滤。当5ms的范围内，获取的两个ECD相差过大时（超过了6000）

        // 说明超过了中心0，只需要对反馈量过滤就可以了

        delta\_ecd=temp->equalPresentECD-temp->equalPreviousECD;

        if(delta\_ecd>DELTA\_ECD\_THRESHOLD)   //顺时针方向越过零点

            temp->presentSpeedEqualECD=(delta\_ecd-ECD\_FULL\_ROUND)/(temp->presentTime\_ms-temp->lastTime\_ms);

        else if (delta\_ecd<-DELTA\_ECD\_THRESHOLD)    //逆时针越过零点

            temp->presentSpeedEqualECD=(delta\_ecd+ECD\_FULL\_ROUND)/(temp->presentTime\_ms-temp->lastTime\_ms);

        else

            temp->presentSpeedEqualECD=delta\_ecd/(temp->presentTime\_ms-temp->lastTime\_ms);

调试时，出现问题。

LED闪烁速率很慢，应该是实时性出现问题。需要测量每个任务的执行时间。

执行时间均小于1ms

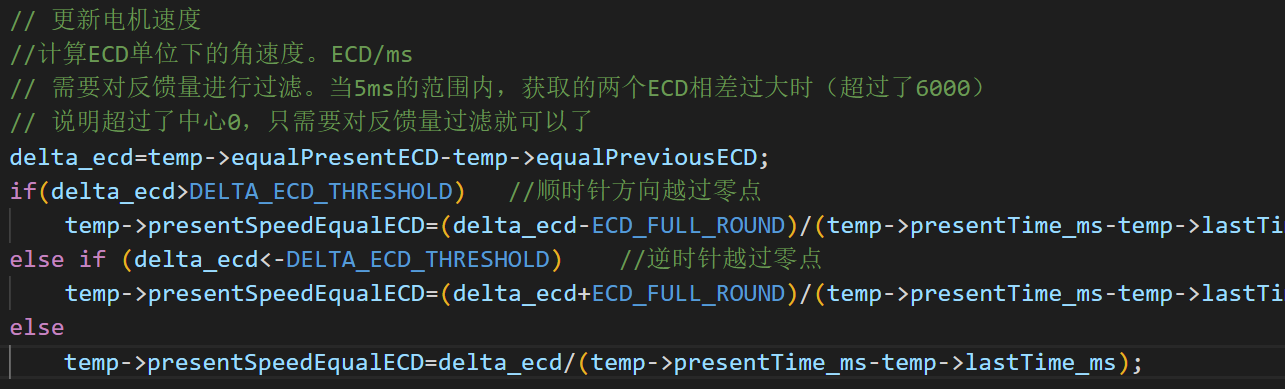
应该是出现了内存溢出的问题

最后完成了，是比较的问题。

机器人能够知道自己的云台转过了多少圈了。

新问题，云台不受控制orz

云台疯转



输出参量，动不动就是天文数字

找到计算过程，强制转换类型，得到正确结果

编译器自己可能会有一些问题。可能没有按照标准C的约定来。那么，我们要用一些括号来避免编译器出错。

delta\_ecd=temp->equalPresentECD-temp->equalPreviousECD;

        if(delta\_ecd>DELTA\_ECD\_THRESHOLD)   //顺时针方向越过零点

            temp->presentSpeedEqualECD=((fp32)(delta\_ecd-ECD\_FULL\_ROUND))/(temp->presentTime\_ms-temp->lastTime\_ms);

        else if (delta\_ecd<-DELTA\_ECD\_THRESHOLD)    //逆时针越过零点

            temp->presentSpeedEqualECD=((fp32)(delta\_ecd+ECD\_FULL\_ROUND))/(temp->presentTime\_ms-temp->lastTime\_ms);

        else

            temp->presentSpeedEqualECD=((fp32)delta\_ecd)/(temp->presentTime\_ms-temp->lastTime\_ms);

好了，输出正常数量级的数字了

输出变量，发现在计算速度时，计算出的很慢。

修改sen，

单位ecd/s

正常转动时，ECD速度在0-2的数量级内。通过延长任务时间获得更好的精度

Safe\_ratio过小。是错误处理了1-r与r

太多错了，把freertos搞崩了。不知道哪里的东西把操作系统的任务调度给弄没了。

需要如何设置每个函数在操作系统中的优先级？各个算法是否存在内存溢出？每个模块会有什么问题？

那就重新从头开始写吧。

Xs，输出一直会是一个很怪的数字，不知道为什么。

一堆尝试后，来了一把操作。

if(switch\_is\_down(rc\_p->rc.s[0]))

            PID\_clear(&(pyControl[PITCH\_INDEX\_IN\_PY].controlSpeedByVoltage\_pid));

        PID\_calc(&(pyControl[PITCH\_INDEX\_IN\_PY].controlSpeedByVoltage\_pid),pyControl[PITCH\_INDEX\_IN\_PY].presentSpeedEqualECD,pyControl[PITCH\_INDEX\_IN\_PY].wantedSpeedECD);

    pyControl[PITCH\_INDEX\_IN\_PY].giveCurrent=(int16\_t)pyControl[PITCH\_INDEX\_IN\_PY].controlSpeedByVoltage\_pid.out;

用遥控器手动清除PID，然后就有比较正常的输出了。

过于奇怪，突然转起来，不妨重新开始写。

要让每一个模块单独测试输入输出。

每一个任务开始写时，要先写debug用的函数和部分

云台重新设计

1. 前言

为了满足比赛需求，需要使用键盘控制机器