基于ESP32的自平衡小车

目标：两轮自平衡小车

实现小车基本的稳定和移动

暂定蓝牙控制

材料：

1. N20直流减速电机(带码盘+轮子) \* 2
2. TB6612电机驱动模块
3. ESP32-wroom
4. MPU6050(GY-521)
5. MP1584EN降压模块
6. 7.4V锂电池
7. 切割亚克力板(支架)
8. 小零件
   1. 开关
   2. LED
   3. 洞洞板
   4. 排线，排针
   5. 电机固定件
9. 可选(bonus)：
   1. IIC\_OLED屏
   2. 蜂鸣器
   3. 超声波传感器(避障)
   4. 9g舵机(起身)

技术难点：

1. 陀螺仪和加速度计的理解
2. MPU6050(GY521)的使用
3. 卡尔曼滤波的理解和应用
4. PID的理解，使用，和参数整定
5. 使用中断的编码盘

制作过程：

1. 硬件
2. 软件

代码函数实现：

1. TB6612驱动电机
2. 码盘测速
3. 使用GY-521获得姿态数据
4. 对GY-521数据卡尔曼滤波
5. 使用PID算法整合输入-控制输出，形成闭环
6. 连接蓝牙实现控制
7. \*NVS的使用

NOTE：

1. 在实现中断处理代码时，最好让ISR仅对中断进行响应，然后把实际的处理（可能包含时间较长的操作）交给主循环来做
2. 引脚6-11连接到了SPI Flash，使用它们可能会影响烧写(15和2，12注意)(16,17)
3. TB6612的VMM引脚一定要连接
4. 尽量多的考虑问题。。。
5. Motor函数中的Speed输入，一定要是int类型，隐式的类型转换会出错，使得电机异常转动
6. String和arduino Json配合使用时一定要使用char接收蓝牙接收数据否则将无法解析字符串数据(本质是以为，串口传输的数据都是字节，需要进行强制类型转换)
7. PID参数一定要注意正负，特别是D，P和I的符号容易理智的判断，而D不容易，当你发现加入D参数后，小车的震荡反而更大了，那么很有可能是符号问题。正确的D可以减小震荡，而反向的，只会增加震荡。

每日进度：

1. 10日开始
2. 中间未记录
   1. 电路错误实现
   2. 实现编码盘数据读取但未连接中断
   3. 实现GY-521的数据读取并进行卡尔曼滤波，但未理清陀螺仪输出数据
   4. 实现TB6612控制N20减速电机
   5. 使用7.4V锂电池，并使用降压模块至5V
3. 16日重新规划电路
4. 19日进行PID调节实现站立