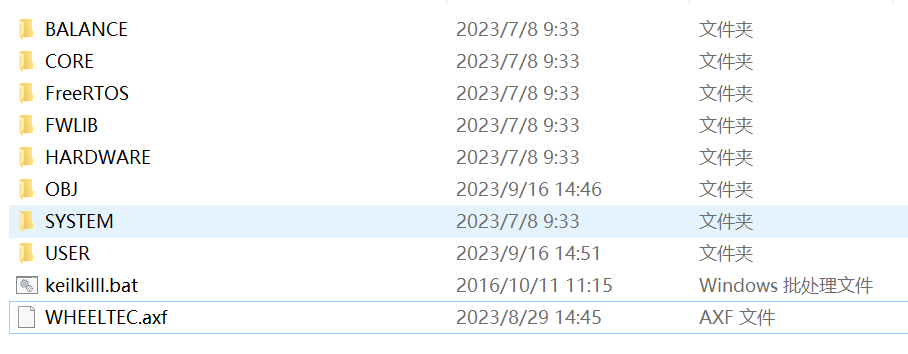
#### 智能车底盘stm32源码说明

智能车底盘控制源码为keil5的标准工程文件，目录结构如下：

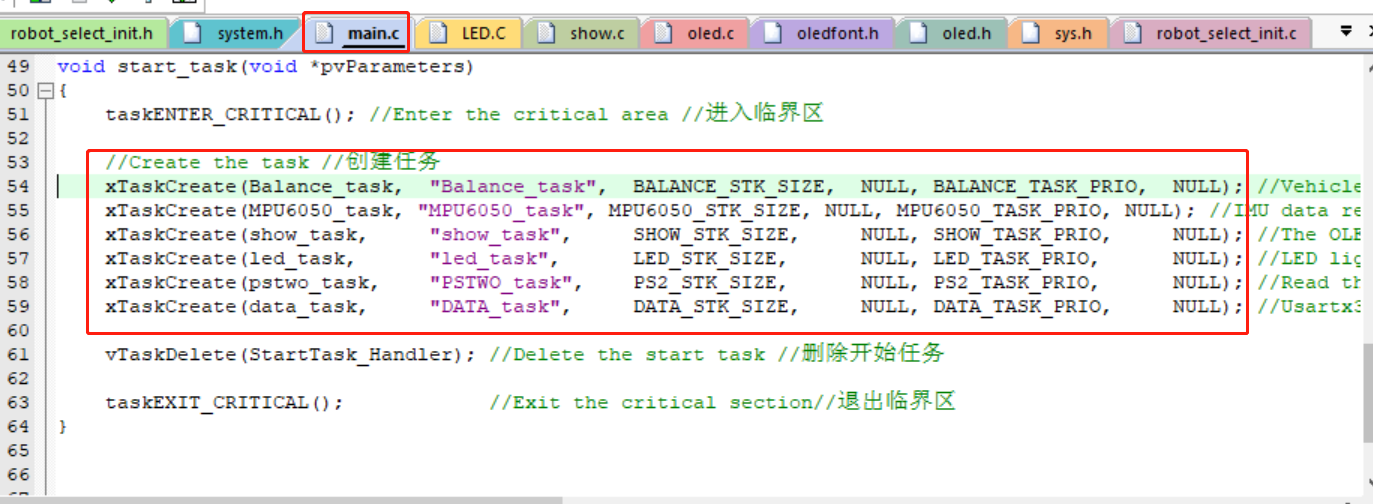


具体每个文件夹含义可参考keil工程相关内容。

打开USER文件夹中拓展名为.uvprojx 文件可以直接打开工程。

源码分析：

工程中函数入口文件main.cpp 中，以任务的形式创建多个并行任务，执行不同的功能：

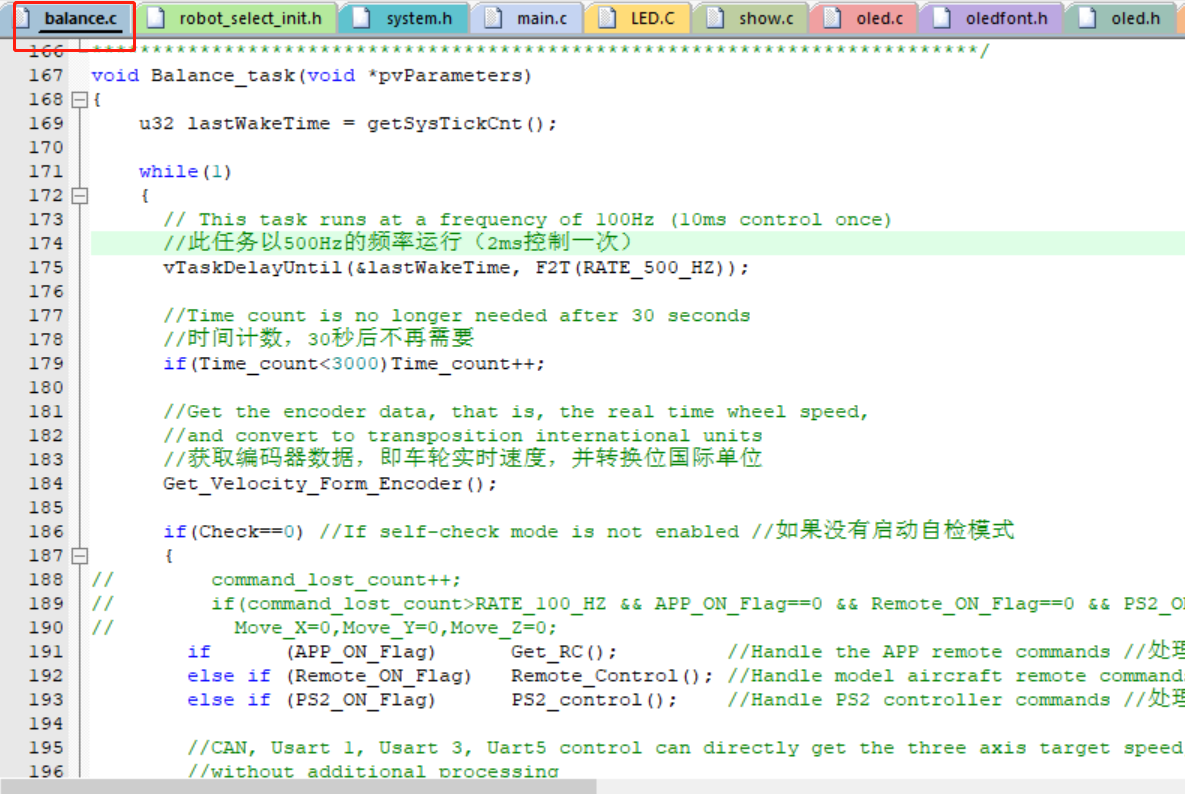


按顺序分别包含：电机控制、MPU6050 IMU传感器读取、oled屏显示、led灯控制、手柄信息处理以及串口信息处理等任务。

创建任务过程中根据任务的重要性设定好优先级，当几个任务并行工作时程序会优先处理优先级更高的任务。

各部分源码根据其功能性，在逻辑及代码量上有明显区别，这里以底盘电机控制为例，进行介绍，其余部分感兴趣可以读取代码，**代码中包含大量注释可以帮助理解。**

Balance.c文件为底盘控制逻辑实现的主文件：

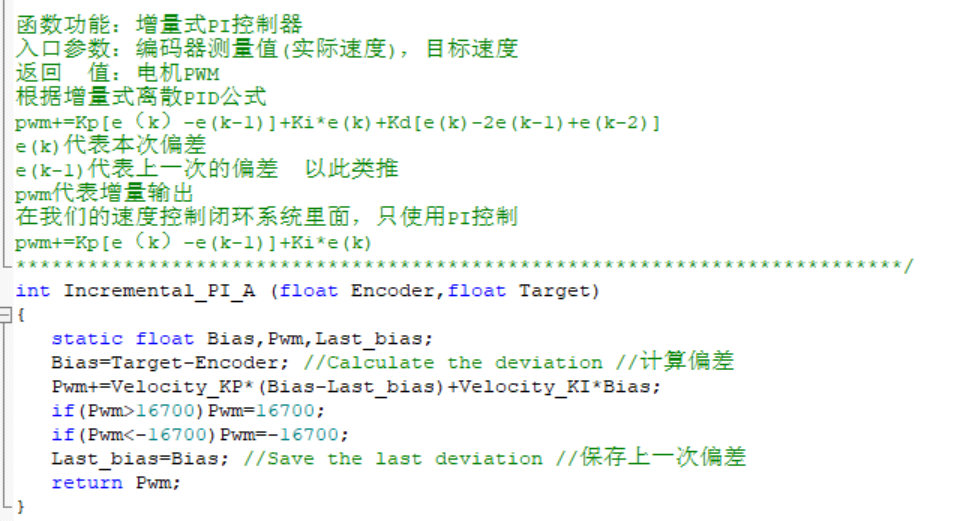


任务的执行实体函数如上（balance task），函数实现：

①编码器读取（Get\_Velocity\_Form\_Encoder）用于速度反馈完成闭环控制，并

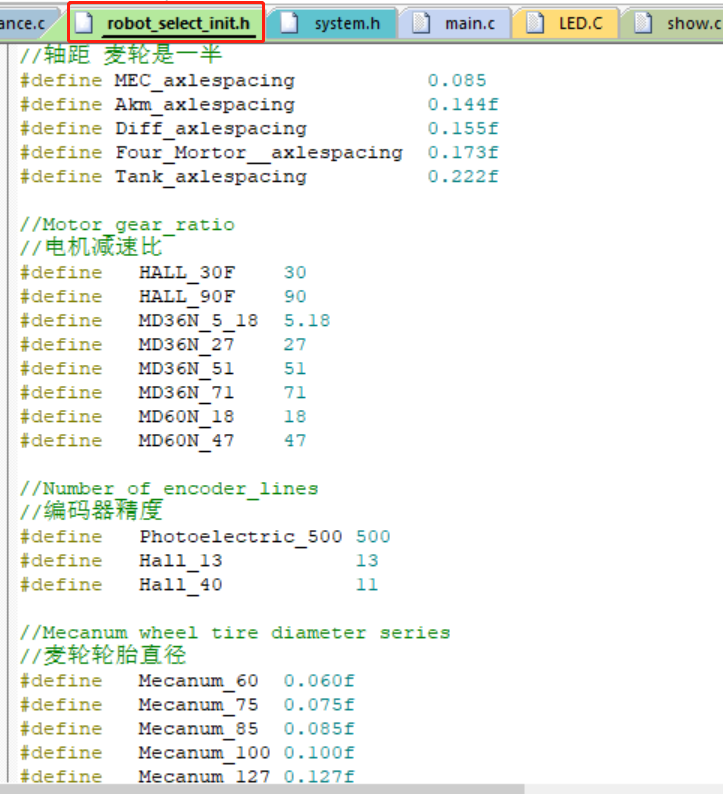
②接收手柄、远程控制型号或串口信息，根据不同车轮类型及运动学模型对信息进行处理。

③将处理后的信息与实际读取到的编码器信息通过PID计算得到可直接控制电机速度的pwm信号。



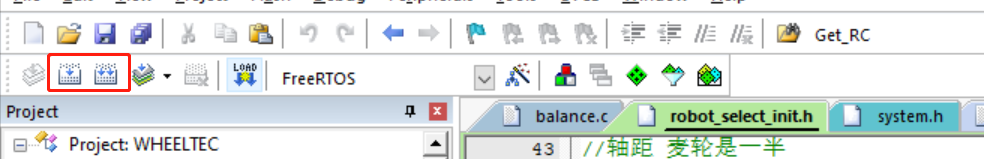
电机运动学及PID相关理论知识可以自行查询，具体实现流程可实际查看代码。

需要注意的是，根据不同的车轮类型、车轮直径及电机编码器线数、减速比需要根据实际值调整参数，才能正常完成小车控制，参数设定文件为：robot\_select\_init.h：



修改对应宏定义即可。

编译下载过程：



点击上图按钮，编译完成后会在主目录的OBJ目录下生成总的.axf二进制文件，之后打开烧写程序以FLYMCU下载到目标板中（现用平台：stm32F403）。



连接上串口后，点击搜索串口完成串口自动选定，之后点击开始编程即可。