

目录

简介:1

软件框架:2

代码规范:2

创建子任务3

简介:

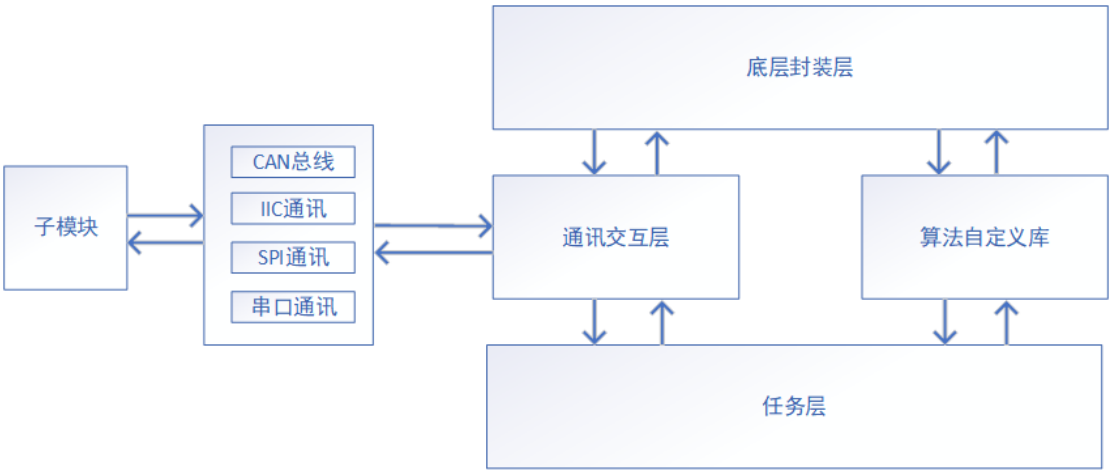
```
# Robot_software

##编译环境
STM32CubeMx 版本: 4.27.0
固件库版本: stm32cube_fw_f4_v1210.zip

##注意事项
基于 Doxygen 的 C/C++注释原则
软件框架分为底层封装层、通讯交互层、任务层

##结构描述
├─ Drivers/CMSIS           //CMSIS 封装函数文件
├─ Drivers/STM32F4xx_HAL_Driver      //HAL 库函数文件
├─ Application/User           // 配置文件
├─ Middlewares/FreeRTOS      //FreeRTOS 系统文件
├─ Application/MDK-ARM       //stm32 启动文件
├─ Application/UserBsp       //底层函数定义文件（根据）
├─ Application/RiserLib      //自定义库文件
|   ├─ remote_ctrl.c         //遥控定义函数
|   ├─ pid.c                 //pid 函数
|   ├─ delay.c               //延时函数
|   └─ definedmath.c         //数学计算函数
├─ Application/Intera //获取反馈信息任务
├─ Application/Comm      //通讯协议文件（Intera、Comm 负责与外部交互，模块化，数据接口）
├─ Application/AppCtrl   //任务文件
└─ IMU // 姿态传感器
```

软件框架：



代码规范：

文件头：

```
/**
 * @file chassis_task.c
 * @version 1.0
 * @date Oct,19th 2018
 *
 * @brief 底盘姿态任务文件
 *
 * @author
 *
 */
```

函数头：

```
/**
 * @brief chassis_task
 * @param
 * @attention
 * @note
 */
```

注意事项：

- 1.缩进统一用空格，不能用 tab 键缩进。
- 2.所有的{}统一占用一行。

创建子任务

- ① 首先在 stm32cubemx 里配置好相关硬件
- ② 在通讯交互层文件夹 Application/UserBsp 加入通讯协议的配置文件;
例如 bsp_can.c
在任务层文件夹 Application/AppCtrl 加入子任务文件;
例如 comm_task.c
在算法自定义库文件夹加入自定义算法文件;
例如 pid.c
- ③ 在 main.c 加入子任务初始化函数
- ④ 然后再 freertos.c 里创建子任务

```
osThreadDef(canTask, can_msg_send_task, osPriorityAboveNormal, 0, 512);  
can_msg_send_task_t = osThreadCreate(osThread(canTask), NULL);
```