嵌入式软件/电控部分

# 嵌入式软件主要理论

## 开发板的选择

本机器人选用了RoboMaster开发板C型作为该机器人电控部分的控制核心。RoboMaster开发板C型采用了高性能的STM32F407芯片，支持两路can输出和uart输，能够满足与机器人各外设及MiniPC等之间的通信；内部集成的IMU模块及恒温加热装置，能够在使用中减少IMU测量数据的温漂误差。与过往赛季使用的RoboMaster开发板A型相比，该开发板在满足机器人所有控制、通信及电源需求的情况下还具有尺寸小巧、易安装、电气保护完善等特点。综合以上原因，选择RoboMaster开发板C型，并基于STM32F407平台进行嵌入式软件的开发。

## 嵌入式软件开发的主要工具、库

### HAL和代码生成器

为了避免繁琐的寄存器操作，使得机器人的嵌入式软件系统开发能够更加专注于机器人控制逻辑的设计以及相关算法的开发与实现，我们选择了ST公司开发的基于STM32系列MCU的HAL硬件抽象层库并基于此进行嵌入式软件系统的开发。为了能够快速便捷地使能并设置单片机各外设模块的参数，我们使用STM32Cube MX代码生成器软件，在图形界面中对单片机的各项外设进行使能并设置相关参数，最终生成含有对应IDE的工程。

### Arm Math

由于使用的STM32F4系列芯片属于ARM Cortex-M4F构架，其具有的独立FPU单元可以大大提升MCU对于数学运算的性能。为了充分发挥该FPU的运算性能，我们调用了ArmMath库，对嵌入式软件系统中所涉及到的数学计算进行优化。

### IDE

本机器人嵌入式软件系统采用ARM Keil v5作为IDE进行开发，并使用该IDE进行代码的下载和仿真调试。另外，由于VS Code在代码阅读，代码自动补全以及代码检查方面的优势，我们在代码的编写和阅读过程中还会使用VS Code+ Keil Assistant扩展进行开发。

# 代码及系统设计

## 嵌入式系统框图



## 代码结构

在本机器人的嵌入式软件设计中，本着“低耦合，高聚合”的原则，基于STM32HAL库，我们的代码结构可以分为以下四个部分：外设驱动部分、算法部分、通信部分以及控制逻辑部分。

### 图形用户界面, 应用程序 描述已自动生成文本 描述已自动生成外设驱动部分

外设驱动部分主要包含了机器人及开发板板载各外设元件的驱动程序。驱动程序调用 HAL库提供的基础外设（如CAN、UART、GPIO等）操作API，并根据电机、遥控等不同外设元件的技术参数及协议特点进行传入数据的解算、填充相应的状态结构体以及输出数据的打包发送。通过调用该部分API，我们可以获取机器人外设（如电机、超级电容控制板等）的各种状态信息，并对它们发送相应的指令，从而实现对机器人外设的控制。

（上图为超级电容控制板的状态结构体以及相应的驱动程序API）

### 算法部分

算法部分主要包括在机器人控制、外设驱动以及传感器反馈数据的解算中经常用到的算法程序，例如计算电机输出电流大小的PID闭环、计算自瞄预测的增强卡尔曼滤波算法等。这些程序不依赖于任何特定的硬件和库，直接调用API传入参数便能使用。

图形用户界面, 文本

中度可信度描述已自动生成

### 通信部分

图形用户界面, 文本, 应用程序

描述已自动生成通信部分包括主控板与上位机MiniPC以及与裁判系统之间的通信相关的代码。里面含有自定义的通信协议的定义，接收数据的校验，解码以及发送数据的打包。此外还有机器人操作界面自定义UI的相关程序。

与mini PC通信的通信协议如下：

1. 手机屏幕截图

   中度可信度描述已自动生成由控制板发送至Mini PC的数据
2. 由 Mini PC发送给控制板的数据

图形用户界面, 应用程序

描述已自动生成

### 控制逻辑部分

该部分为机器人控制的核心部分。我们采用了FreeRTOS操作系统，并运用一个中控函数来对机器人的所有状态量进行维护。来自于控制器以及上位机的数据作为传入的控制量对机器人当前状态以及相关状态参数进行修改；改变后的状态量通过xxx\_interface传入到底盘、云台、发射机构等子模块的状态结构体中，对各模块的运动参数（如底盘前后/左右速度，自旋速度，云台姿态等）进行修改。运动参数通过xxx\_calculate函数进行进一步的解算得到每一个电机的位置或速度设定值，最后通过PID闭环反馈算法计算出每一个电机的电图形用户界面, 应用程序

中度可信度描述已自动生成流输出，通过can总线将指令发送到每个电机。



## 算法说明

在机器人的嵌入式软件系统开发中，我们主要运用了PID算法，基于增强卡尔曼滤波的自瞄预测算法，以及下坠预测算法。

### PID算法

PID算法被广泛应用于对电机运动位置和速度的精确控制，借助于电机上的编码器反馈数据以及IMU模块反馈的姿态信息，以及给定的目标值，我们可以建立一个完整的闭环控制。

此即为位置式PID的计算公式，err为目标值与反馈值的差，由此式即可由反馈速度和目标速度计算出输出的电流值。

若需要对位置进行控制，则可以在速度环之上再添加一位置环，即由反馈位置和目标位置通过PID计算出输出的速度值，再将其作为目标速度与反馈速度进行PID计算，最后得出输出电流值。

### 自瞄预测算法

自瞄预测算法用于自瞄开启时预估目标的运动速度，结合自身弹丸速度以及目标距离计算出能够击打到目标所需的提前量，对击打哨兵有着重要的作用。

自瞄预测算法的原理分为两个部分，首先是计算目标信息：在一定周期内，根据视觉传回数据计算目标的角度和速度，对角度和速度进行二阶卡尔曼滤波融合得出更可靠的目标信息。然后根据滤波得到的角度和位置信息，通过调参的方式得到不同情况下能够击打成功的比例系数，将其乘以目标速度作为提前量的预测值加到给定的角度上去，从而计算出在预测模式下机器人云台的目标姿态。

### 下坠补偿算法

在机器人尤其是英雄机器人发射弹丸的时候，由于弹丸自身重力的影响会导致弹丸的轨迹呈抛物线，因此，在自瞄模式下需要根据弹丸的初速度进行下坠补偿，才能让弹丸更加精确地命中目标。

在进行大量的数据测量，并且通过Matlab拟合之后发现弹丸下坠补偿模型接近于线性模型即：

其中y为pitch轴的下坠角度，x为视觉传感器中计算得到的机器人与目标之间的距离。k和b为参数，需要通过手动的射击测试得到。与此同时，为了避免视觉回传的pitch轴增量数据抖动，采用了一阶低通滤波为进行消抖。最终，我们可以得到加入了下坠补偿后的pitch轴目标角度的计算公式

式中，为滤波系数，m为由鼠标输入的控制量，可以实现在一定范围内的手动微调。

# 附：键位操作

1. 右键单独按进出自瞄，此时的自瞄Yaw 和pitch都是小电脑发送的

2. 右键长按进入自瞄，此时的自瞄模式pitch轴由电脑发送，yaw轴通过操作手手动给

3. 左键为射击键

4. R键为小陀螺

5. X 键为云台跟随底盘用于爬坡

6. V键即为卖血模式，解除枪口热量限制

7. B为开关红点

8. F键为扭腰，点击可进行对弹丸的闪避

9. Z键为坦克模式，实现了云台和底盘的分离控制

10. 遥控器左侧档杆拨到中间打开摩擦轮

11. Ctrl+G 键进行软重启