**负熵之舟战队**

一、项目概述

1.1 比赛任务分析

· 赛事性质：资源争夺型机器人对抗赛

· 核心目标：设计1-2台机器人，在限定时空内最大化收集和占有道具资源

· 胜负判定：最终持有道具数量多者获胜

1.2 设计理念与创新点

· 整体策略：快速搜集+自重防守

· 创新体现：

1.2.1机械：1).利用橡皮筋的弹性进行收集小球，再利用小球的自重和收集装置一定的坡度使得小球滚入储存装置中

2).创新的扇叶滚筒设计以及可输送小球的篮子，再加上独特的三层储存装置，使得小球易搜集的同时在储存装置中不易掉出

1.2.2 硬件：模块化机身，可拆卸结构，零件损坏时能秒换，适配不同任务模块的快速转换

二、系统架构设计

2.1 硬件系统组成

主控系统：STM32F4系列

执行机构：TT马达×4 + 舵机×2

控制输入：PS2手柄

机械结构：弹丸收集机构 + 底盘运动系统

2.2 软件架构（C语言）

c

// 系统主循环框架

int main(void) {

HAL\_Init();

SystemClock\_Config();

PS2\_Init(); // PS2手柄初始化

Motor\_Init(); // 电机驱动初始化

Servo\_Init(); // 舵机初始化

while(1) {

PS2\_Data\_Process(); // 手柄数据处理

Motion\_Control(); // 运动控制

Ammo\_Handler(); // 弹丸处理

}

}

2.3 机器人选型理由

· 基于比赛场地大小：2400mm\*2400mm的正⽅形区域

· 基于资源分布特性：弹丸为直径约17mm的⻩绿⾊塑㬵球道具，随机位于场地的任何位置

· 基于战术需求：携带弹丸数更多者胜出

三、关键技术实现

3.1 机械结构设计

外形展示与尺寸规范

· 结构设计：采用弹性底面和扇叶型滚筒的前置收集装置，软布传输和可移动篮子传输装置，以及挡板型储存装置

· 尺寸验证：通过直尺测量，机器人尺寸符合规则要求

· 材料选择：3D打印件+铝合金框架，重量小于2.5kg兼顾强度与重量

收集机构设计迭代

第一辆小车

· V1.0问题：普通铲子状收集后如果没有即使送回储存装置，容易造成二次掉出

· V2.0改进：采用橡皮筋作为底面，其余面封口，留下一个进入收集装置的孔

· V2.0问题：全封口设计重量太大，使得小车容易重心不稳

· 最终方案：加大开口，上下均为开口设计以减小自重，用轻质橡皮筋防止小球的掉落

第二辆小车

· V1.0问题：橡皮筋滚筒设计，无法将小球导入纯储存装置中，并且后期收集比较费力

·V 2.0改进：用滚筒加扇叶的形式地面的小球运输到储存装置中

动系统选型

· 电机对比：直流电机 vs 步进电机 vs 伺服电机

· 最终选择：直流电机

结构稳定性测试

· 测试方法：从200mm高度自由下落，底面朝下

· 测试结果：连续3次测试无零件脱落

· 结构优化：采用榫卯设计使得各个零件连接稳固

3.2动力系统控制

单一马达控制（C语言实现）

c

// TT马达控制函数

void TT\_Motor\_Control(uint8\_t motor\_id, int16\_t speed) {

if(motor\_id >= MAX\_MOTORS) return;

// 速度限制

speed = (speed > MAX\_SPEED) ? MAX\_SPEED : speed;

speed = (speed < -MAX\_SPEED) ? -MAX\_SPEED : speed;

// 设置PWM输出

\_\_HAL\_TIM\_SET\_COMPARE(&htim, motor\_channels[motor\_id],

ABS(speed));

// 设置方向

HAL\_GPIO\_WritePin(dir\_ports[motor\_id], dir\_pins[motor\_id],

(speed >= 0) ? GPIO\_PIN\_SET : GPIO\_PIN\_RESET);

}

舵机控制实现

c

// 舵机角度控制

void Servo\_Set\_Angle(uint8\_t servo\_id, uint8\_t angle) {

uint16\_t pulse\_width = SERVO\_MIN + (angle \* SERVO\_SCALE);

\_\_HAL\_TIM\_SET\_COMPARE(&htim, servo\_channels[servo\_id], pulse\_width);

}

3.3 底盘运动控制

PS2手柄通信协议

c

// PS2数据解析

typedef struct {

uint8\_t left\_x, left\_y;

uint8\_t right\_x, right\_y;

uint16\_t buttons;

} PS2\_Data\_t;

void PS2\_Data\_Process(void) {

if(PS2\_ReadData(&ps2\_data) == PS2\_OK) {

// 解析摇杆数据控制底盘

int16\_t vx = (int16\_t)ps2\_data.left\_x - 128;

int16\_t vy = (int16\_t)ps2\_data.left\_y - 128;

Chassis\_Control(vx, vy);

}

}

全向底盘运动算法

c

// 四轮差速运动模型

void Chassis\_Control(int16\_t vx, int16\_t vy) {

// 运动学逆解

float wheel\_speeds[4];

wheel\_speeds[0] = -vx - vy + rotation; // 前左

wheel\_speeds[1] = vx - vy + rotation; // 前右

wheel\_speeds[2] = -vx + vy + rotation; // 后左

wheel\_speeds[3] = vx + vy + rotation; // 后右

// 设置四个TT马达速度

for(int i = 0; i < 4; i++) {

TT\_Motor\_Control(i, (int16\_t)wheel\_speeds[i]);

}

}

3.4 弹丸收集系统

收集机构设计

· 机构类型：按压式（小车1）、滚筒式（小车2）

· 驱动方式：马达控制，舵机控制

四、难点突破与迭代优化

4.1 电机同步控制问题

难点：四个TT马达速度不一致导致轨迹偏差

解决方案：

c

// 电机速度校准

void motor\_speed\_calibration(void) {

for(int i = 0; i < 4; i++) {

actual\_speeds[i] = read\_encoder\_value(i);

speed\_compensation[i] = TARGET\_SPEED / actual\_speeds[i];

}

}

4.2 弹丸收集可靠性

难点：不同位置弹丸的收集成功率差异大

解决方案：优化收集机构开口角度，增加导向结构

五、测试验证体系

5.1 测试方案设计

· 单元测试：各模块功能验证

· 集成测试：系统联调测试

· 场景测试：模拟比赛环境压力测试

六、战队日志

第1周：基础框架搭建

1.1 机器人基础外形设计

开发历程

第1-2天：概念设计与建模

· 快速确定四轮差速底盘+前置滚筒收集和按压收集两种方案

· 使用Fusion 360进行简化三维建模，重点保证结构强度

第3-4天：快速原型制作

第5-7天：结构优化

· 尺寸测量：在270×180×240mm之内

· 200mm跌落测试：优化连接件设计，通过测试

· 采用3D打印+标准件组合，加快制作速度

1.2 单一马达驱动系统

开发历程

第3-4天：驱动电路搭建

· 使用现成的L298N电机驱动模块

· 简化电路设计，直接使用开发板PWM输出

第5-6天：基础控制实现

第7天：功能验证

· 完成单个TT马达和舵机的基础转动测试

· 验证控制响应时间<100ms

第2周：运动控制系统

2.1 底盘基础运动

开发历程

第8-9天：PS2通信快速实现

第10天：单一方向运动

第11-12天：全向运动集成

第13-14天：运动优化

· 完成PS2手柄控制全底盘运动

· 实现基础移动功能，响应正常

2.2 多马达协同控制

开发历程

第12-13天：基础同步控制

第14天：功能验证

· 四个TT马达同时转动测试通过

· 基础同步功能实现

第3周：弹丸收集系统

3.1 弹丸收集机构

开发历程

第15-16天：收集机构快速实现

· 采用简化的滚筒式收集机构

· 快速组装和调试

第17天：单弹丸收集

第18-19天：功能优化

· 单弹丸收集测试：成功率85%

· 优化收集角度和时序

第20天：多弹丸连续收集

第21天：系统测试

· 10颗弹丸连续收集测试：成功率80%，时间35秒

· 满足基础功能要求

4.1 系统完整测试

开发历程

第22天：功能全面测试

· 外形展示：完整机器人绕车展示

· 马达运动：单个和多个马达测试

· 底盘运动：PS2手柄控制验证

· 尺寸规范：直尺测量确认

· 弹丸收集：单颗和10颗收集测试

· 结构稳定性：200mm跌落测试

第23天：问题修复

· 修复弹丸收集机构的卡顿问题

· 优化电机同步性能

第24天：性能优化

第25天：文档整理与提交

· 整理所有测试数据和代码

· 准备最终提交材料

第26天：简单实现

测试总结

基础功能完成情况

功能项 状态 备注

外形展示 ✅ 完整机器人展示

单一马达转动 ✅ TT马达和舵机

底盘水平运动 ✅ PS2手柄控制

多马达转动 ✅ 4个TT马达

尺寸规范 ✅ 控制在270mm×180mm×240mm以内

单弹丸收集 ✅ 成功率85%

多弹丸收集 ✅ 10颗成功率80%

结构稳定性 ✅ 通过跌落测试

开发经验总结

时间管理策略

1. 并行开发：机械、电路、软件同步进行

2. 快速原型：优先实现基础功能，再优化

3. 模块化设计：各功能独立开发测试

技术要点

· 采用成熟的电机驱动方案，减少调试时间

· 简化控制算法，保证基础功能稳定性

· 优先确保所有基础功能达标

附录

源代码结构

/src

├── main.c # 主程序

├── motor\_control.c # 电机控制

├── ps2\_controller.c # 手柄控制

└── ammo\_system.c # 弹丸收集

硬件清单

· STM32F103C8T6核心板

· L298N电机驱动模块×2

· TT马达×4 + 舵机×2

· PS2手柄接收器

八、附录

8.1 代码核心片段

```cpp

// 提供关键算法代码，附详细注释