

第十七届全国大学生 智能汽车竞赛

技术报告



学校：湖北工业大学

队伍名称：蓝电YYDS Car

参赛队员：李欣泽 梁耀

刘思进 张海良

带队教师：詹云峰 黄周

关于技术报告和研究论文使用授权的说明

本人完全了解全国大学生智能汽车竞赛关保留、使用技术报告和研究论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名： 李欣洋 梁耀
张海良 刘思进

带队教师签名： 黄周 詹云峰

日 期： 2022年8月15日

摘 要

全国大学生智能汽车竞赛是以智能汽车为研究对象的创意性科技竞赛，是面向全国大学生的一种具有探索性的工程实践活动，是教育部倡导的大学生科技竞赛之一。

本文将介绍湖北工业大学蓝电YYDS Car的队员们在设计智能车过程中的成果，包括智能车的整体方案、机械、硬件、算法等方面的介绍。

本次比赛采用了组委会官方所提供的完全模型组车模（I车模），车模上搭载的硬件平台中控制部分是TC212单片机完成，识别和判断由百度提供的Edgeboard计算卡来完成。Edgeboard 采用摄像头对赛道信息进行检测，判断赛道元素，从而准确地通过每一个赛道元素；TC212控制部分采用ICM20602六轴传感器为辅，从而实现对车身和赛道的精准判断。Edgeboard与TC212通信采用串口通信，以数据帧格式进行数据传输，完成赛规要求的运行任务。

目录

第一章 引言	1
1.1 背景	1
1.2 组别任务分析与车模制作方案	1
第二章 机械结构设计	2
2.1 传感器的设计与安装	2
2.2 Edgeboard 与电池的固定	3
2.3 小结	4
第三章 硬件电路设计	5
3.1 硬件电路总体框架	5
3.2 主控板设计	6
3.3 PCB 设计	10
第四章 软件程序设计	11
4.1 车模控制方法	11
4.2 赛道图像处理算法	11
第五章 上位机及开发软件	13
5.1 软件开发工具	13
5.2 硬件开发工具	13
5.3 上位机调试工具	14
参考文献	15
附录：算法代码	16

第一章 引言

1.1 背景

智能车是一种高新技术密集型的新型汽车，它涵盖的范围广泛包括模式识别、传感器技术、自动化控制实现、电力电子技术、计算机技术等多个领域。在国际上已经形成智能汽车研究、设计、开发、竞赛的热潮。

本竞赛以“立足培养，重在参与，鼓励探索，追求卓越”为指导思想，旨在促进高等学校素质教育，培养大学生的综合知识运用能力、基本工程实践能力和创新意识，激发大学生从事科学研究与探索的兴趣和潜能,倡导理论联系实际、求真务实的学风和团队协作的人文精神，为优秀人才的脱颖而出创造条件。

近年来，人工智能掀起热潮，驾驶技术也在不断向着智能化的方向发展，全国大学生智能汽车竞赛顺应着科技发展趋势，十六届智能车也顺应时代潮流，开展了多个新组别，给大学生们在人工智能等前沿科学技术应用方面提了动手实践的宝贵机会。

1.2 组别任务分析与车模制作方案

我队于今年一月开始准备并参加第十七届全国大学生智能汽车竞赛百度完全模型组别，我们使用Edgeboard作为计算平台，以摄像头作为主传感器，通过OpenCV计算出赛道边线，使用百度的paddlepaddle识别标志，学习、识别标志，串口输出到TC212单片机，TC212完成对于车辆编码器、电机驱动、舵机的控制，利用连到电机上面的编码器作为反馈信号，然后通过闭环控制来实现电机的平稳运转。

第二章 机械结构设计

2.1 传感器的设计与安装

2.1.1 摄像头的安装

摄像头我队选用的是低照度摄像头（KS2A418），该摄像头具有1080P/60帧的高帧速，适合高速情况下采集图像。

为最大化减轻整车重量并降低车辆质心，摄像头的固定件采用3D打印材料，将摄像头支撑在一定高度，有较高的定位、安装精度。



图 2.1 摄像头安装

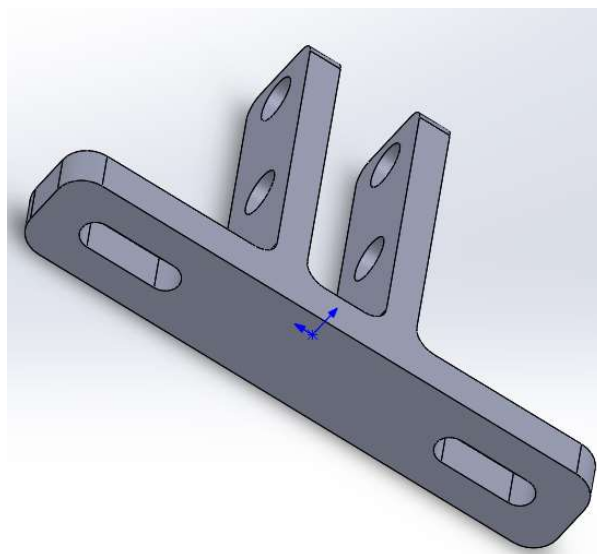


图 2.2 摄像头固定结构件

2.1.2 编码器的安装

编码器采用官方车模原装的编码器，用于作为速度闭环控制的真实测量值。编码器在后面的电机转动下跟随转动，可以较为准确的测量电机所转的圈数，提供给TC212来读取。

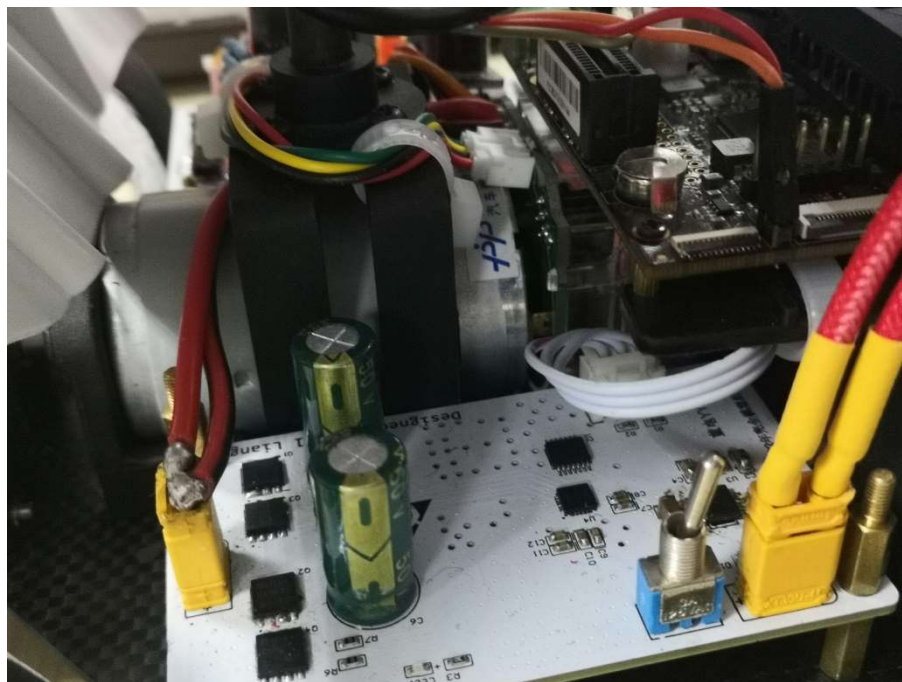


图 2.3 编码器安装

2.1.3 姿态传感器的安装

车辆的姿态传感器采用ICM20602六轴传感器模块。姿态传感器的安装位置直接影响到车辆的直立和转向控制，其安装位置应尽可能贴近车辆质心，所以将姿态传感器安装在了主控板的底部。

2.2 Edgeboard 与电池的固定

2.2.1 Edgeboard 的安装

为使车辆质心尽可能集中，将Edgeboard安装在了车辆的中心位置，在一定程度上降低了车辆质心，提升了车辆的极限性能、稳定性。

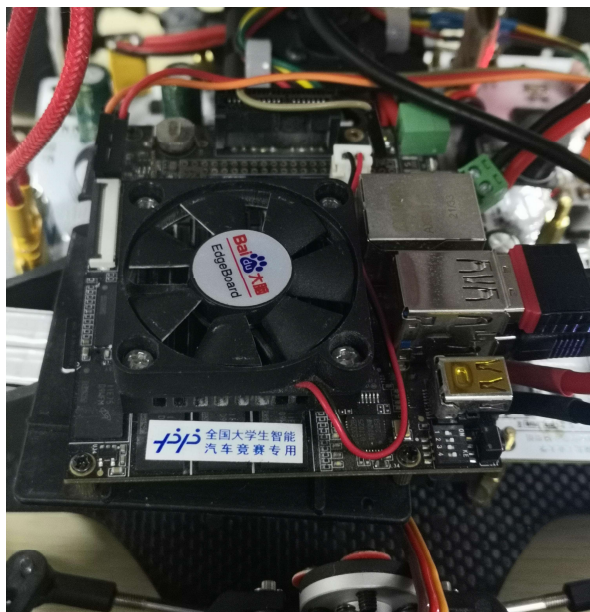


图 2.4 Edgeboard 安装

2.2.2 电池位置

我队选择将电池放置在车模提供的电池槽内，电池槽设计在车模的中前方，和车辆后面的电机质量相平衡，可以增加车辆的稳定性。



图 2.5 电池的安装

2.3 小结

我队车模的机械结构设计都是为了在不断地降低质心，集中质心。合理分配车前后的质量，可以很好地提升车辆的稳定性、动力性、灵敏性。

第三章 硬件电路设计

3.1 硬件电路总体框架

从最初进行硬件电路设计时就既定了系统的设计目标：可靠、高效、简洁，在整个系统设计过程中严格按照规范进行。

可靠性是系统设计的第一要求。我们对电路设计的所有环节都进行了电磁兼容性设计，做好各部分的接地、屏蔽、滤波等工作，将高速数字电路与模拟电路分开，使本系统工作的可靠性达到了设计要求。

高效与实用性是指本系统的各模块能充分完美的实现相应的功能。车模采用双电机驱动，我们设计了单独的驱动芯片组成驱动器，瞬间驱动电流最大可以达到几十安培。

简洁是指在满足了可靠、高效的要求后，为了尽量减轻车模的负载，降低模型车的重心位置，应使电路设计尽量简洁，尽量减少元器件使用数量，缩小电路板面积，使电路部分重量轻，易于安装，在设计完原理图后，注重PCB板的布局，优化电路的走线，整齐排列元器件，最终做到电路板的简洁。

下面是硬件设计整体框架图：

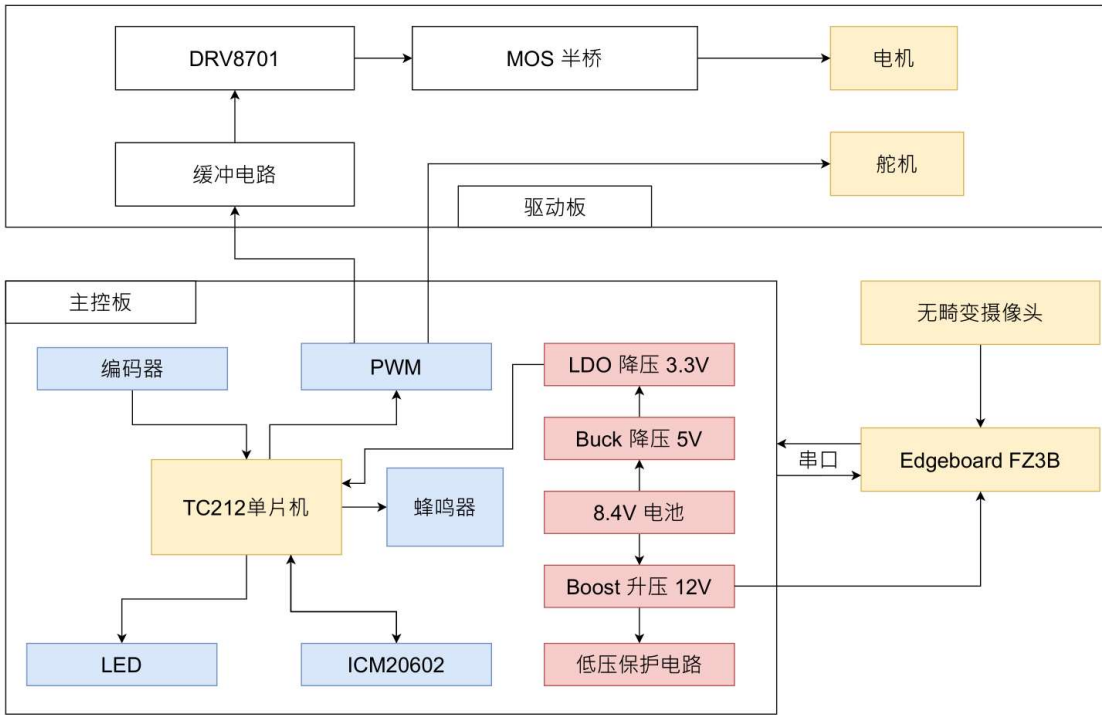


图 3.1 硬件整体框架图

3.2 主控板设计

3.2.1 电源管理模块

本系统中需要稳定提供的电压有5V和3.3V，5V用于给舵机、陀螺仪传感器、无线串口模块供电，3.3V用于给陀螺仪、单片机芯片供电。

5V稳压电路采用的是TPS5430，TPS5430是TI公司最新推出的一款性能优越的DC/DC开关电源转换芯片。TPS5430具有良好的特性，其各项性能及主要参数如下：

高电流输出：3A（峰值4A）；

宽电压输入范围：5.5~36V；

高转换效率：最佳状况可达95%；

宽电压输出范围：最低可以调整降到1.221V；

内部补偿最小化了外部器件数量；

固定500kHz转换速率；

有过流保护及热关断功能；

具有开关使能脚，关状态仅有17uA静止电流；

其内部具有软启动，与其他同类型直流开关电源转换芯片相比，TPS5430的转换效率更高。

5V稳压电路设计如下：

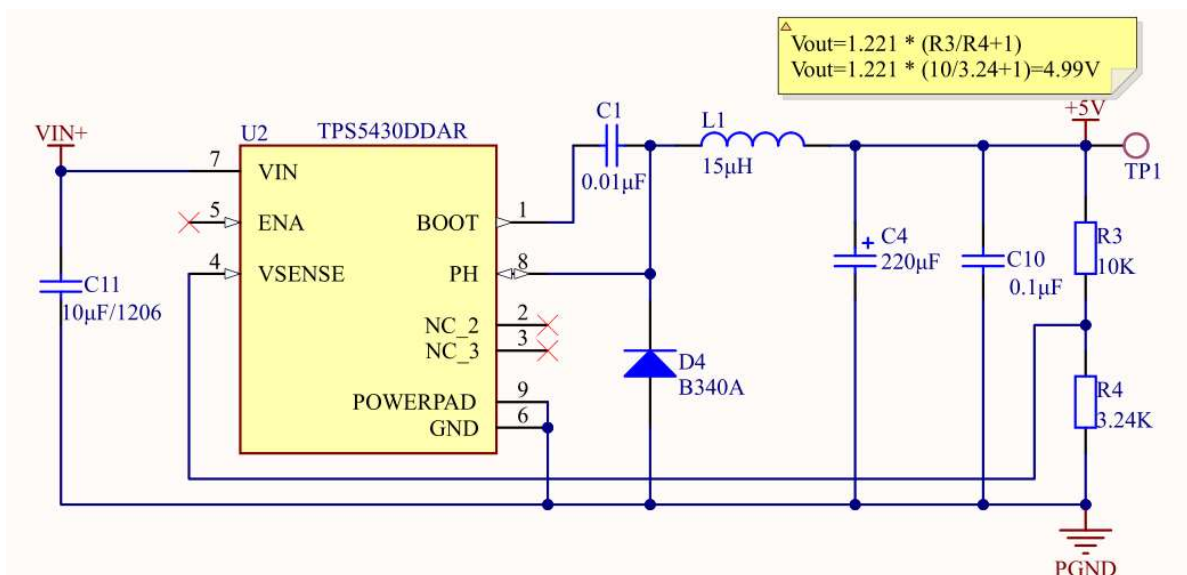


图 3.2 5V 稳压电路

3.3V稳压电路采用的是RT9013-33GB，RT9013-33GB是一款3.3V固定输出高性

能低压差(LDO)线性稳压器, 具有极高的PSRR和超低压差。该器件的静态电流低至25 μ A, 进一步延长了电池寿命。该器件还可配合低ESR陶瓷电容器使用, 从而减少电源应用所需的电路板空间, 非常适合手持无线设备应用。该器件在关断模式下的功耗为0.7 μ A (典型值), 并且具有小于40 μ s的快速导通时间。其他特性包括超低压差, 高输出精度, 限流保护, 高纹波抑制比。

3.3V稳压电路如下:

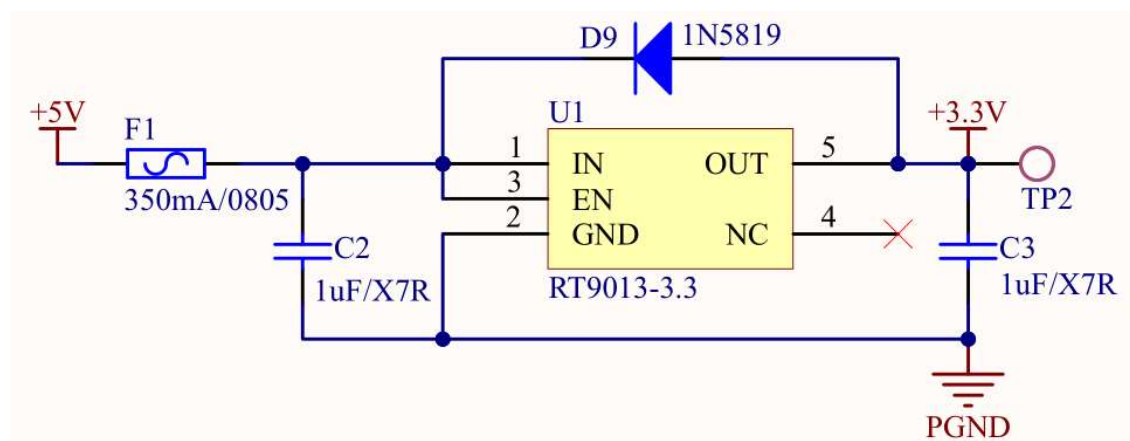


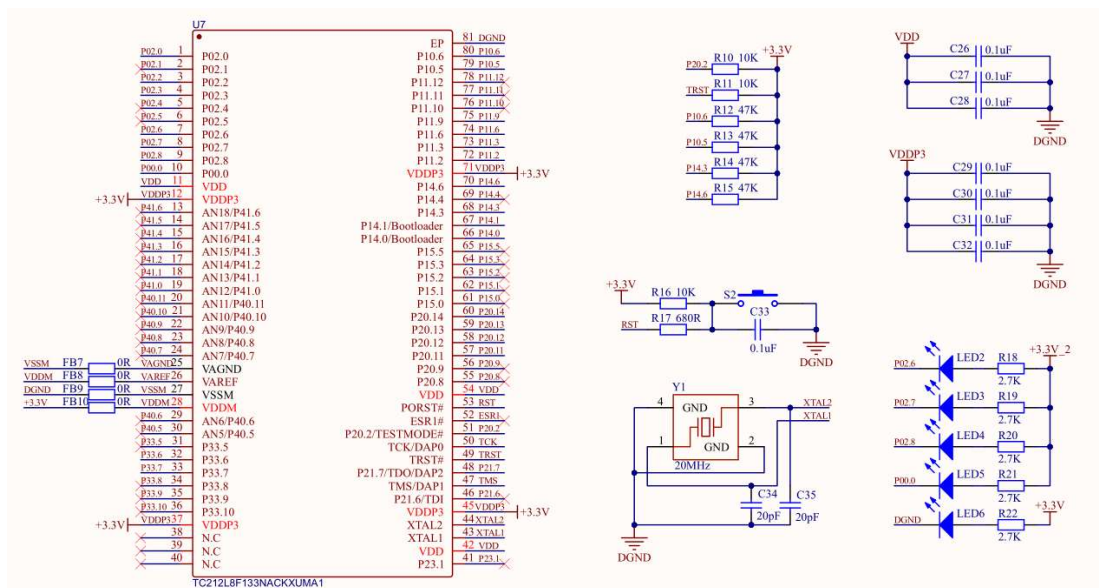
图 3.3 3.3V 稳压电路

3.2.2 单片机最小系统电路设计

本系统采用的主控MCU为英飞凌的TC212, 该芯片是英飞凌推出的适用于汽车和工业应用的强大 AURIX™ 微控制器。该芯片为使用TriCore, 具有 DSP 功能, 最高工作频率可达133MHz, 具有高达 0.5MB 的闪存、64KB EEPROM 和高达 56KB 的 RAM, 具有 ECC 保护, 具有丰富的增强型I/O 端口和外设。

该芯片具有高集成度, 降低复杂度, 节省大量成本, 采用全新单电源方案, 拥有同类产品中极其优秀的功耗, 同时节省外部电源成本, 在性能、封装、存储空间和外围设备方面可以扩展, 为跨平台设计方案提供灵活度。

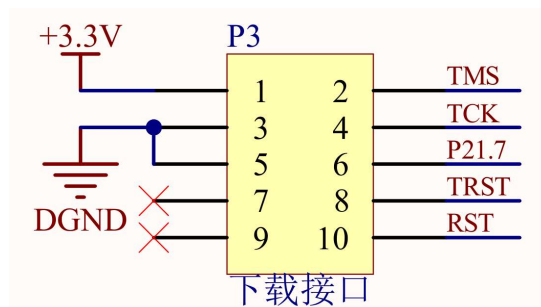
本作品中单片机最小系统电路设计如下:



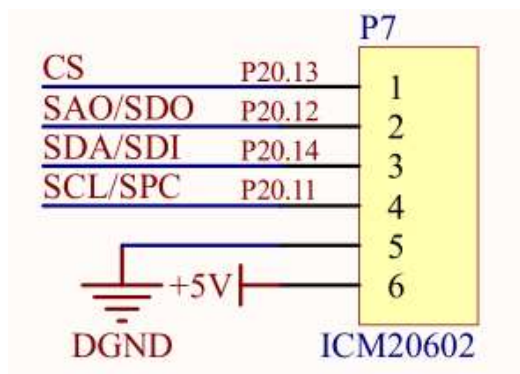
3.2.3 传感器接口电路设计

我们在主控电路板上设计了而许多传感器接口，方便于连接各种传感器来搭建一个完整的智能车系统。

(1) 下载调试接口



(2) 陀螺仪传感器接口



(3) 编码器及电机信号接口

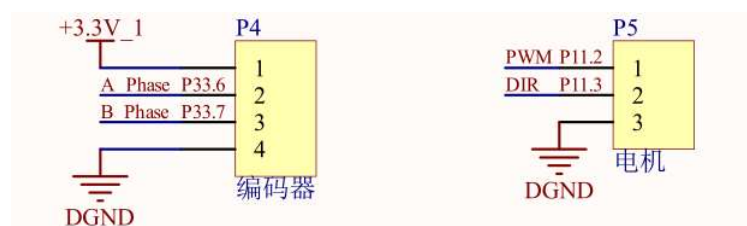


图 3.7 编码器及电机信号接口

3.2.4 电机驱动电路

我们采用的是H桥式电机驱动方式，采用TI公司推出的DRV8701芯片来驱动4个外部N通道金属氧化物半导体场效应晶体管以达到驱动直流车模电机的效果，DRV8701是一款采用4个外部N通道MOSFET的单路H桥栅极驱动器，主要用于驱动12V至24V双向有刷直流电机。

该器件可通过PH/EN（DRV8701E）或PWM（DRV8701P）接口轻松连接控制器电路。内置的感测放大器能够实现可调节的电流控制。这款栅极驱动器内置有DRV8701采用9.5V V_{GS}栅极驱动电压来驱动FET。所有外部FET的栅极驱动电流均可通过IDRIVE引脚上的单个外部电阻进行配置。

该器件内置以下保护功能：欠压锁定，电荷泵故障，过流关断，短路保护，前置驱动器故障以及过热保护。故障情况通过nFAULT引脚指示。

驱动电路设计如下：

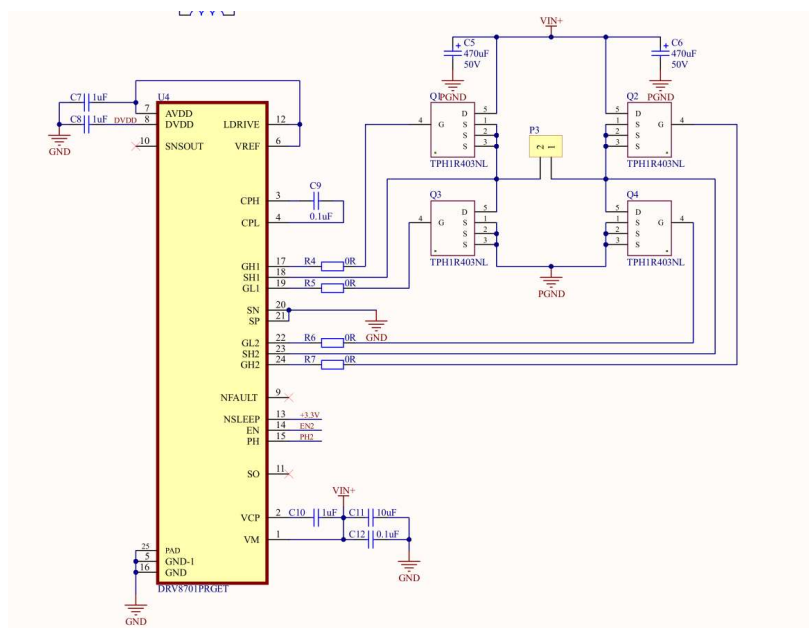


图 3.8 电机驱动电路

3.3 PCB 设计

综合考虑驱动性能、车身重心、维修难易等因素，我们选择将电路板模块化设计，采用主控板、驱动分离的方法。

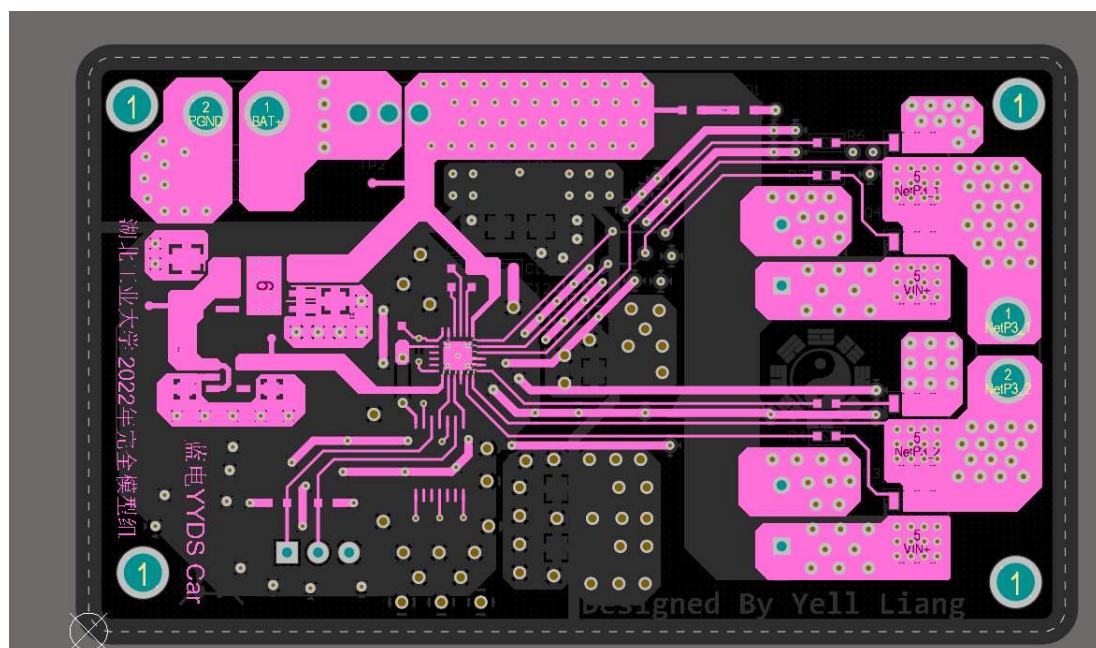


图 3.9 电机驱动板

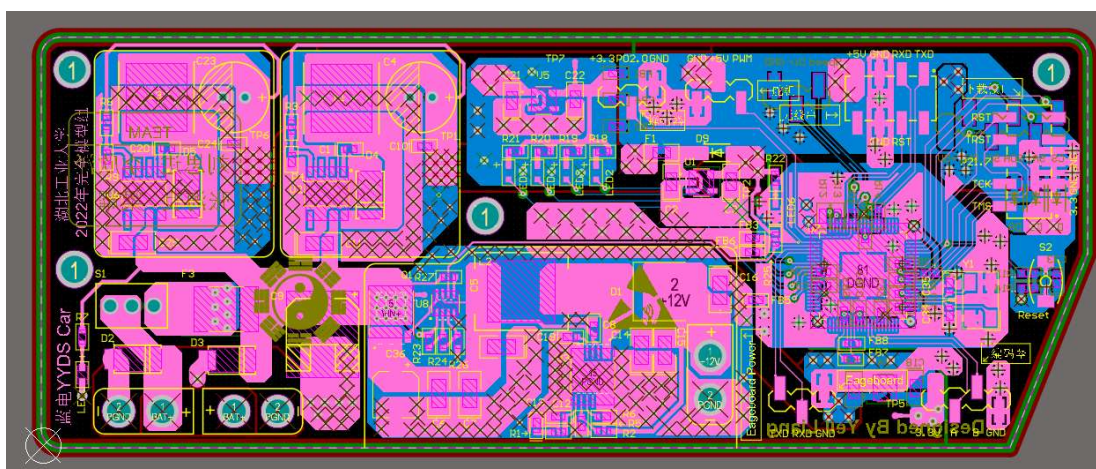


图 3.10 主控板

第四章 软件程序设计

4.1 车模控制方法

4.1.1 速度控制算法

采用PID算法进行速度控制，以编码器读值作为当前真实速度，Edgeboard解算出的当前速度作为预设速度，PWM占空比作为输出。

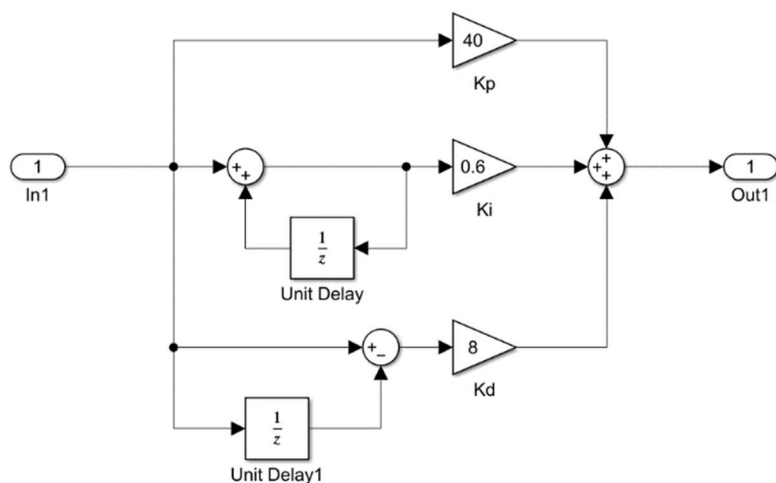


图 4.1 速度控制流程图

4.1.2 转向控制算法

采用PID算法进行转向控制，Edgeboard解算出的转向角度和IMU的车体姿态计算出预设转向角度，PWM占空比作为输出。

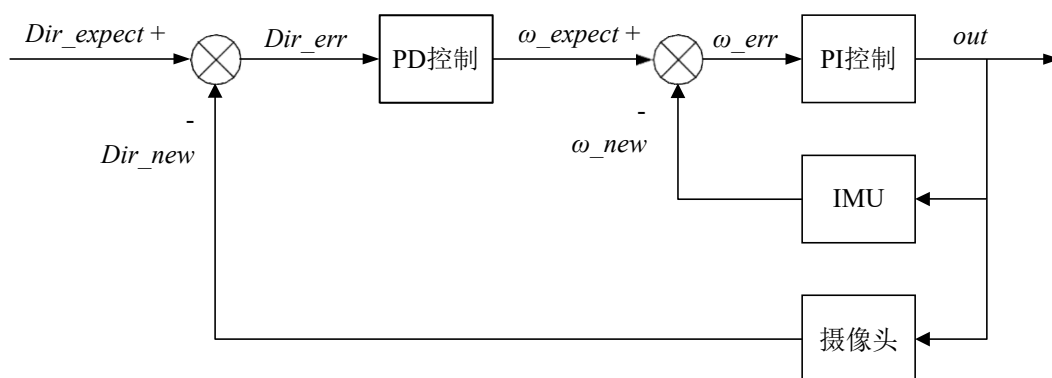


图 4.2 转向控制流程图

4.2 赛道图像处理算法

4.2.1 单线偏移的目标轨迹规划

我们使用基于单线偏移的目标轨迹规划和识别方案来确定预锚点来设定小车的目标路径。

在去畸变和透视变换后得到俯视视角下的赛道边线，由于赛道的宽度是固定的，将左边线向右侧“平移”一半宽度，或将右边线向左侧“平移”一半宽度即可得到中线。对于左边线上的每一个点，首先计算此处的切线，然后将该点沿切线法向右侧移动一半宽度即可得到中线上的一点。在寻线时，通过大量利用切换巡左中线和巡右中线的方式来获得稳定的巡线。

4.2.2 基于角点识别赛道元素

我们使用基于角点来识别赛道的元素，在此之前，已经完成了对赛道边线的识别，得到了稳定且等距的边线前提下，直接取每个点前后20cm构成一个三点，利用三角函数解算出 θ 的角度，根据 θ 的角度来判断直角、锐角等，根据当前的图片中角点的数量来判断赛道元素，例如含有多个 90° 的角点的出入库元素。

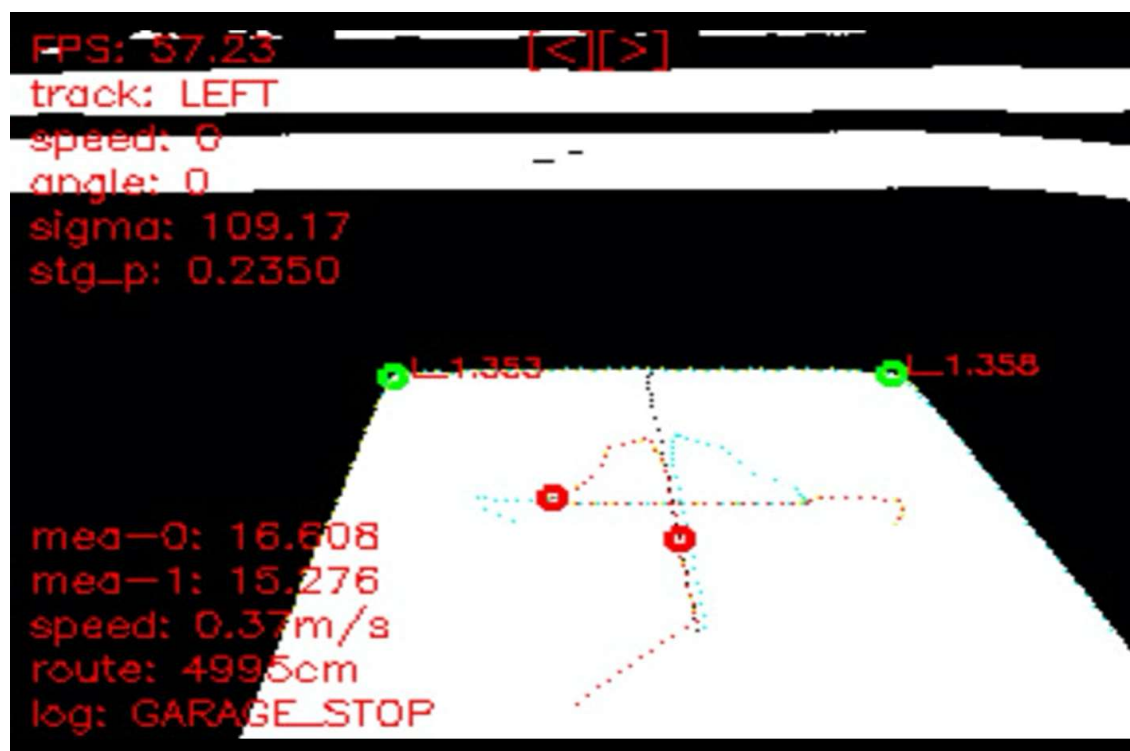


图 4.3 赛道元素提取效果图

第五章 上位机及开发软件

5.1 软件开发工具

由于本次比赛的微控制器限定了使用英飞凌的处理器，我们的控制部分采用的是英飞凌的TC212，为了更好的进行软件调试，开发环境选择了英飞凌官方的AURIX Development Studio，不仅可以在线编译代码，还可以Debug进行在线调试代码：

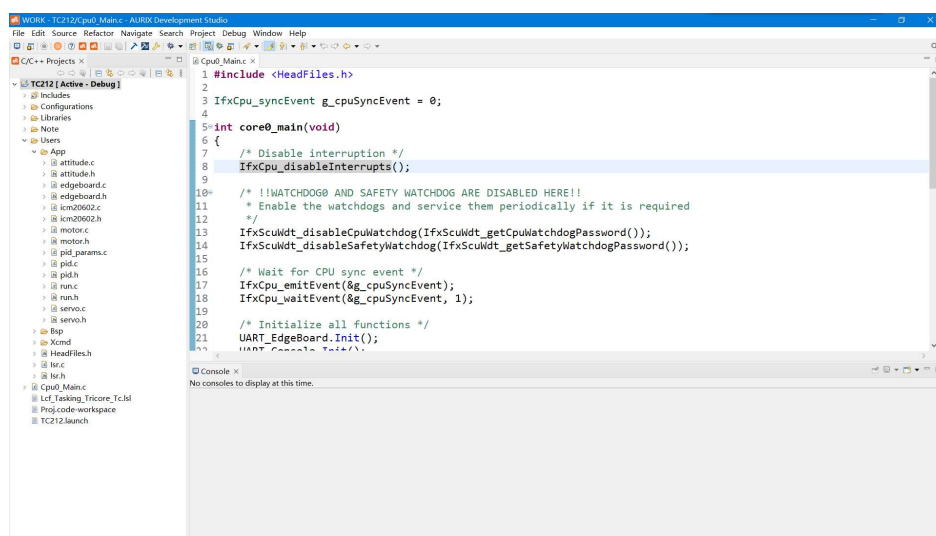


图 5.1 ADS 软件开发环境界面

由于本次比赛限定了使用百度的Edgeboard完成摄像头的处理计算，我们采用Visual Studio Code Server 来远程登录到 Edgeboard 开发板并完成代码的编写、编译、运行、调试。

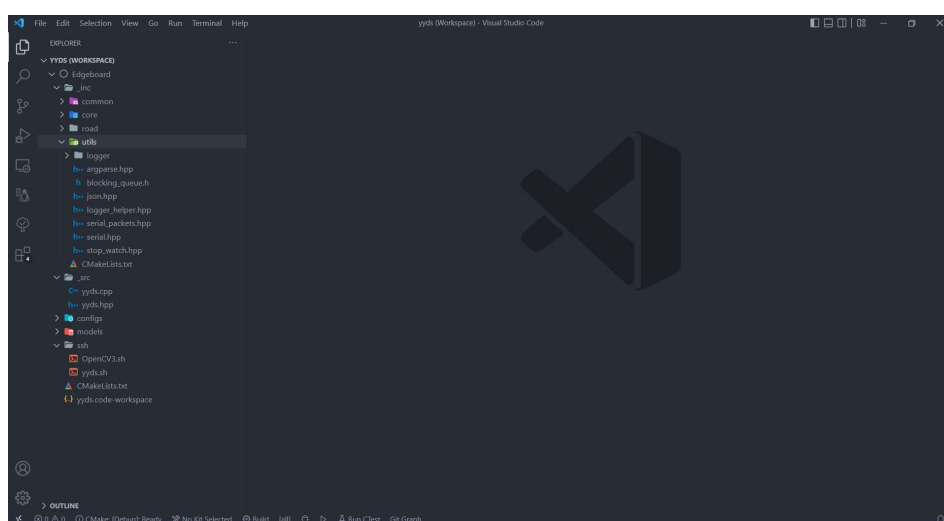


图 5.12 VSCoDe 软件开发环境界面

5.2 硬件开发工具

原理图与PCB设计软件我们使用Altium Designer，这套软件通过把原理图设计、电路仿真、PCB绘制编辑、拓扑逻辑自动布线、信号完整性分析和设计输出等技术的完美融合，为设计者提供了全新的设计解决方案，使设计者可以轻松进行设计，熟练使用这一软件使电路设计的质量和效率大大提高。

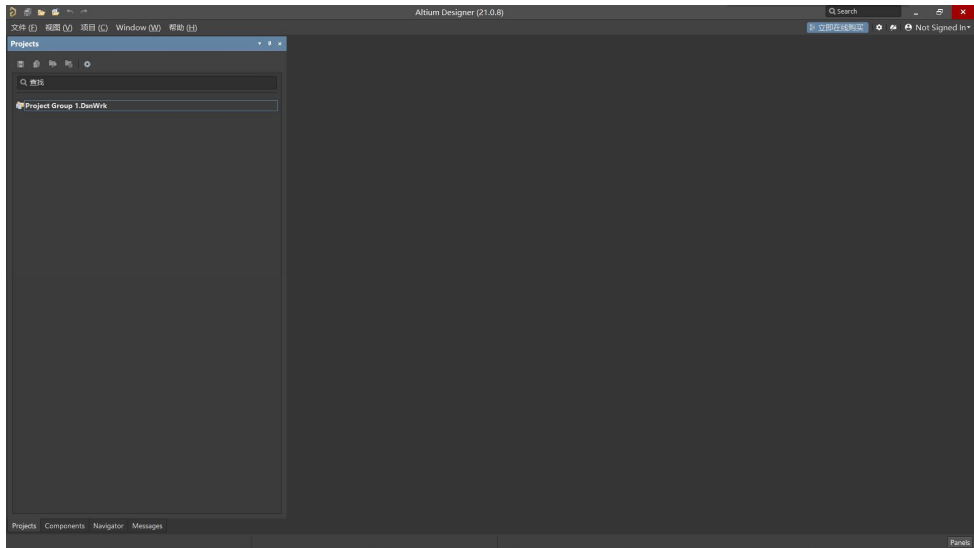


图 5.2 Altium Designer 界面

5.3 上位机调试工具

为了能够观察各个变量的给定和输出波形，以更方便的进行调试分析，我队使用C#开发了一款上位机，功能非常全面，可以帮助我们完成PID参数调试，而且还有美观的波形显示和命令行内与TC212交互。

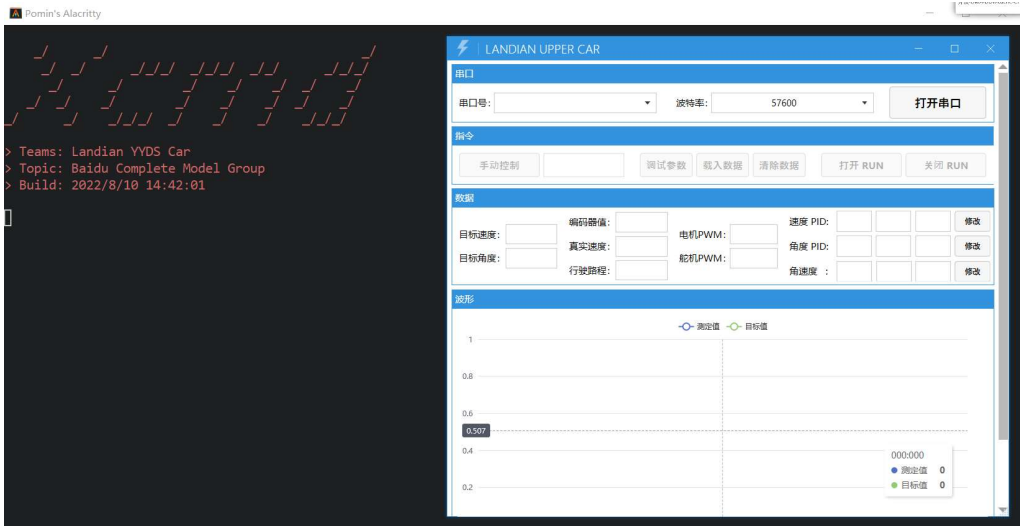


图 5.3 我队开发的调试助手界面

参考文献

- [1]王兆安.电力电子技术[M].北京：科学技术出版社，2011.
- [2]颜循进.恩智浦杯全国智能车竞赛之PID调节[J].科技风,2020(10):263.
- [3]林华,徐宇宝,汪鑫,吴瑶,邬志军,曹昌勇.摄像头与电磁导航智能车的设计与双车控制研究[J].佳木斯大学学报(自然科学版),2020,38(04):90-94.
- [4]卓晴.黄开胜.邵贝贝．学做智能车 [M]．北京：北京航空航天大学出版社．2007.
- [5]康华光．电子技术基础.模拟部分 [M]．北京：高等教育出版社．2006.
- [6]卓晴.王盼宝.马莉.智能车制作.北京：清华大学出版社. 2017.

附录：算法代码

```
#include "yyds.hpp"

#undef LOG_TAG
#define LOG_TAG "Main"

ImageProcess imgprocess; // 图像处理
logger_helper logger;    // 日志

std::shared_ptr<CaptureInterface> capture = nullptr; // 线程 -> 摄像头
std::shared_ptr<Detection> detection = nullptr;      // 线程 -> 目标检测
std::shared_ptr<Display> display = nullptr;          // 线程 -> 图像显示及视频录制
std::shared_ptr<SerialDriver> serial = nullptr;      // 线程 -> 串口接收及发送

/* 主函数 */
int main(int argc, char const *argv[]) {
    /* 数据定义 */
    bool _windows, _video; // 命令行参数
    int ret = 0;           // 初始化返回

    char fps_buffer[16]; // 帧率图像显示缓存
    double fps_time = 0; // 帧率计数

    cv::Mat frame_origin; // 摄像头原始图像
    cv::Mat frame_imgpro; // 图像处理输出

    Payload_t payload; // 数据包 (EB -> TC212)
    Stopwatch stopwatch; // 计时器

    /* 命令行参数 */
    auto args = util::argparser("yyds car command line parameters explained.");
    args.set_program_name("yyds")
        .add_help_option()
        .add_option("-l", "--launch", "launch!!!")
        .add_option("-w", "--windows", "show windows")
        .add_option("-v", "--video", "save video")
        .parse(argc, argv);
    windows = args.has_option("--windows");
    video = args.has_option("--video");

    /* 日志初始化 */
    logger.init logger();

    /* 程序退出信号 */
    signal(SIGINT, callbackSignal);

    /* 图像处理 */
    ret = imgprocess.init("../configs/control.json", (_windows || _video));
    CHECK(0 == ret) << "../configs/control.json can not found !" << LOG_END;

    /* 摄像头 */
    capture = std::make_shared<CaptureInterface>();
    ret = capture->init();
    CHECK(0 == ret) << "device video can not open !" << LOG_END;
```

```

/* 目标检测 */
detection = std::make_shared<Detection>();
ret = detection->init("../configs/usbcamera.json");
CHECK(0 == ret) << "../configs/usbcamera.json can not found !!" << LOG_END;

/* 窗口 */
display = std::make_shared<Display>(_windows, _video);
ret = display->init();
CHECK(0 == ret) << "windows or video can not create !!" << LOG_END;

/* 串口 */
serial = std::make_shared<SerialDriver>("/dev/ttyPS1", BaudRate::BAUD_230400);
ret = serial->open();
CHECK(0 == ret) << "/dev/ttyPS1 can not open !!" << LOG_END;

/* 启动多线程 */
capture->start();
detection->start();
display->start();

/* 开跑 */
while (capture->_loop) {
    // 开始计时
    stopwatch.tic();

    // 获取图像
    capture->get(frame_origin);

    // 目标检测
    detection->run(frame_origin);

    // 图像处理
    imgprocess.process(frame_origin, frame_imgpro, payload);

    // 发送指令
    if (capture->_launch) {
        serial->sendPack(payload);
    }

    // 显示图像
    sprintf(fps_buffer, "FPS: %.2f", 1000 / fps_time);
    cv::putText(frame_imgpro, (std::string)fps_buffer, Point(5, 10),
        cv::FONT_HERSHEY_SIMPLEX, 0.4, cv::Scalar(0, 0, 255));
    display->put(frame_imgpro.clone());

    // 停止计时
    fps_time = stopwatch.toc();
}

/* 结束程序 */
callbackSignal(0);
exit(EXIT_SUCCESS);
}

#undef LOG_TAG

```

```

#define LOG_TAG "System"

/* 系统信号回调函数: 系统退出 (Ctrl+C) */
void callbackSignal(int signum) {
    /* 防止多次退出系统 */
    static uint8_t is_exit = 0;
    is_exit += 1;

    if (is_exit == 1) {
        /* 结束线程 */
        capture->stop();
        detection->stop();
        display->stop();
        serial->close();

        /* 结束日志 */
        log_i("Exit Code: %d", signum);
        log_i("Finish !!!");
    }
}

```