

# 第十四届全国大学生“恩智浦”杯 智能汽车竞赛创意组 技术报告



学    校：    武汉理工大学

队    名：    996. icu

参赛队员：    岳鹏宇

                余晨阳

                郭盼根

                常文杰

                沙佳辉

指导老师：    邹斌  曲帅锋

## 关于技术报告和研究论文使用授权的说明

本人完全了解第十四届“恩智浦”杯全国大学生智能汽车竞赛关于保留、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，比赛组委会和“恩智浦”半导体公司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： 沙佳辉 岳鹏宇  
郭昕桐 余晨阳 常文杰

带队老师签名： 邹文武

日期： 2019.07.31

# 目 录

摘要 .....	I
1 引言 .....	1
2 系统整体设计 .....	2
2.1 系统整体设计思路 .....	2
2.2 系统设计框图 .....	2
2.3 系统通讯方案 .....	3
3 机械系统设计与实现 .....	4
3.1 车体结构设计及搭建材料选择 .....	4
3.2 高处摄像头（车顶部扫描用）防抖设计 .....	6
3.3 棋子拾取模块设计 .....	7
3.3.1 棋子吸取方案 .....	7
3.3.2 缓冲结构设计 .....	9
3.4 障碍板摆放装置设计 .....	9
4 硬件系统设计与实现 .....	11
4.1 主控电路设计 .....	11
4.2 最小系统电路设计 .....	12
4.3 外设接口电路设计 .....	13
4.4 输入辅助模块 .....	13
4.5 输出模块电路设计 .....	13
4.6 继电器模块电路设计 .....	14
4.7 驱动电路设计 .....	14
5 图像处理与数字识别 .....	16
5.1 处理器选型及参数 .....	16
5.2 AprilTag 的识别以及信息处理 .....	16
5.3 棋子与挡板识别 .....	17
5.4 车身位置确定及姿势控制 .....	17
6 运动控制系统设计 .....	19
6.1 运动控制程序设计总体思路 .....	19
6.2 运动控制流程设计 .....	19
6.3 车模精确运动实现 .....	19
6.3.1 车模测速 .....	20
6.3.2 车模角度环 .....	20
6.3.3 车模直线行驶 .....	22
6.4 车模运动控制 .....	22

6.5 运动方向控制 .....	23
6.6 路径规划方案 .....	24
6.7 车模调试 .....	25
6.7.1 开发工具.....	25
6.7.2 调试工具.....	25
7 预赛阶段针对八皇后系统的特殊设计 .....	26
7.1 棋盘防滑及棋子吸盘设计 .....	26
7.2 场地边缘识别棋子特殊情况处理.....	26
7.3 双摄切换方案.....	26
8 决赛阶段针对对抗策略的特殊设计 .....	28
8.1 特殊的挡板盒子设计及挡板的相关改进 .....	28
8.2 棋盘坐标变换 .....	28
8.3 处理器选择 .....	29
8.4 算法设计 .....	29
9 其他特殊设计 .....	32
9.1 摄像头稳定结构设计 .....	32
9.2 主板等电路板固定.....	34
9.3 摄像头（底部）安装.....	34
参考文献.....	36
附录 .....	37
1 主控单元部分程序源码 .....	37
2 电路原理图 .....	39

## 摘要

本智能车系统以恩智浦公司的 MK60FX512ZVLQ15 作为车模运动控制芯片，使用基于 NXP i.MX.RT 的可编程图像识别模块进行标签、棋盘以及挡板的识别，使用恩智浦 i.MX6Q 开发板运行 Minimax 与剪枝算法，完成下棋策略计算任务。主控芯片接收来自处理器的信息，结合 PID 控制电机运动，通过控制算法控制四个麦克纳姆轮以不同的速度与方向旋转，以精确控制车模向任意方向移动。通过电磁铁，舵机、舵机摆臂的配合，利用曲柄连杆机构很好的完成了棋子的搬运以及障碍的放置。整个系统由图像检测模块、运动控制模块、电机及驱动模块、物体搬运模块组成。经过多次实验与调试，该系统很好的完成了比赛任务。

**关键词：**图像检测、Minimax 与剪枝算法、PID 控制、麦克纳姆轮

# 1 引言

全国大学生智能车竞赛是一项以“立足培养、重在参与、鼓励探索、追求卓越”为指导思想，面向全国大学生开展的具有探索性的工程实践活动。它以在特定赛道上能自主行驶且具有优越性能的智能汽车这类复杂工程问题为任务，鼓励大学生组成团队，综合运用多学科知识，提出、分析、设计、开发并研究智能汽车的机械结构、电子线路、运动控制和经济成本等问题，激发大学生从事工程技术开发和科学研究探索的兴趣和潜能，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。

新一代人工智能技术成为现今国家经济发展新的推动力，它为自主无人系统提供了更强的能力，包括复杂动态的场景感知与理解、精准实时定位以及面向复杂环境的自适应智能导航等。第十四届全国大学生智能汽车竞赛创意组别之一室内棋类对弈组，传承第十三届的比赛模式，及“步步为营”的棋类对弈形式，并对比赛规范做了修正。创意比赛要求参赛队伍设计一辆车模能够通过车载摄像头完成赛场内线条、标签的检测与辨认、几何物体识别与搬运。通过机械、电子和控制算法完成预赛和决赛两个阶段的目标排序任务和对抗挑战任务。

竞赛环境是位于室内的六米见方平坦的地面上铺设蓝色广告布背景上，喷涂有八乘八的棋盘格。在棋盘格内有 April Tags,从第 1 格至 64 格，每隔一格放置一个标签。预赛阶段的任务是将参赛队伍提供的八块棋子，或者八块障碍板块，由组委会统一随机摆放在棋盘格内。参赛队伍的车模进行场地内，通过自主移动板块，使得板块在棋盘内满足“八皇后问题”约束。

决赛阶段是是对抗比赛。按照预赛排位进行分组对抗。进行对抗的两个队伍，分别在场左右两侧底线对称中心位置摆放自己的圆形皇后棋子，然后轮流交替上场，完成自己棋子的移动以及为对方设置障碍。对抗比赛将按照三轮两胜制完成晋级。具体方案与信标组的对抗比赛的过程相同。

本技术报告主要包括机械设计方案、硬件设计方案、软件设计方案、创新点说明等，详尽地阐述了我们的设计方法，具体表现有机械结构的创新设计以及控制算法的独特想法。本次方案的设计凝聚了我们队伍的智慧与汗水，是共同努力努力的结果。在准备比赛的过程中，我们电路设计，软件编程，系统调试，机械搭建等方面的能力均得到了提升，其中多次遇到困难，但我们不曾想过放弃，一直坚持下来，在每一次调试 Bug 中，找出自己的错误，逐渐突破自己，收获了宝贵的财富。

## 2 系统整体设计

### 2.1 系统整体设计思路

创意组智能车车模由我们自己搭建完成。主体骨架采用铝合金材料（2020 铝材）搭建，再辅以环氧树脂板及部分结构采用 3D 打印件搭建完成。以恩智浦公司的 MK60FX512ZVLQ15 作为车模运动控制芯片，使用基于 NXP i.MX.RT 的可编程图像识别模块进行标签、棋盘以及挡板的识别，使用恩智浦 i.MX6Q 开发板运行 Minimax 与剪枝算法，完成下棋策略计算任务。本系统可以分为三部分，分别是车模运行控制、图像处理识别、机械设计。车模运行控制包括路径规划、下棋走法运算，车模运动控制。路径规划采用离车模最近原则规划，而车模运动控制采用车位置环为外环、电机速度环为内环，控制直流电机作用在麦克纳姆轮实现运动，并分别将整个棋盘看做是一个坐标系来控制车模在坐标系中任意移动。图像处理为标签识别，棋盘识别，障碍识别三部分。使用基于 NXP i.MX.RT 的可编程图像识别模块，基于 Openmv 库，可以很好的完成这部分任务。机械设计分为移动模块，棋子拾取与释放模块，挡板摆放模块和其他模块四个部分。通过自行设计并通过实验，该系统能够很好的完成给定任务。

### 2.2 系统设计框图

根据上述的设计思路，可将整个系统分为：电源模块、主控单元、图像处理模块、运动策略计算模块、输出模块，为此设计出如图 1 设计框图。

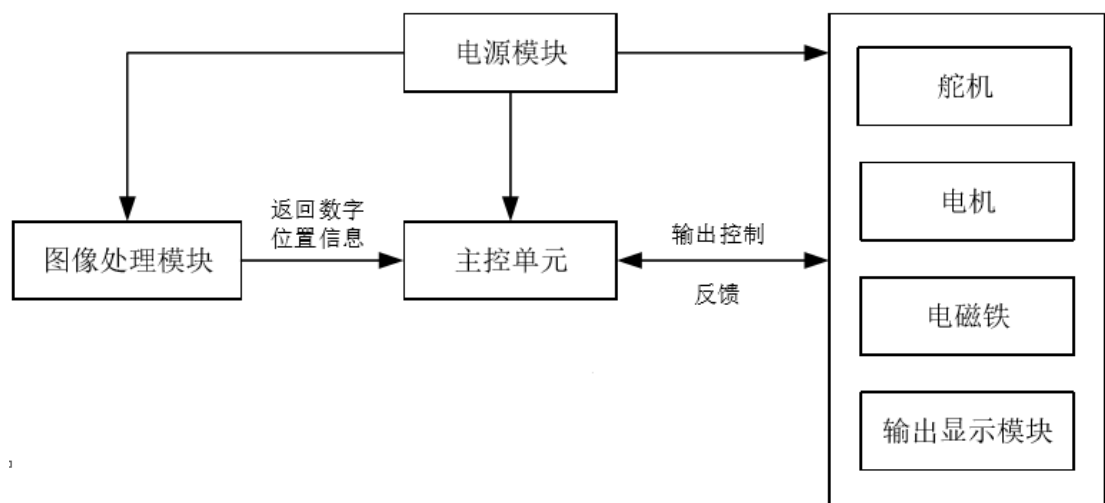


图 1 设计框图

## 2.3 系统通讯方案

本系统使用了两块 K60，四个摄像头以及一个恩智浦 i.MX6Q 开发板。为保证系统稳定的运行，我们需要设计合理的通讯方案。

我们的通讯全部采用串口通讯，通过自己编写通信协议，可以保证安全稳定的传输。

主核与副核都分别接收一个摄像头传过来的数据，并分别测量两路编码器计数值。副核向主核发送路径偏差与编码器计数值。主核同时还要接收两个可编程图像识别模块发过来的位置信息，通过路径算法输出信息作用到电机与舵机。在决赛中，主核还要向开发板发送棋盘信息。开发板经过计算后输出下一步移动信息返回给主核。

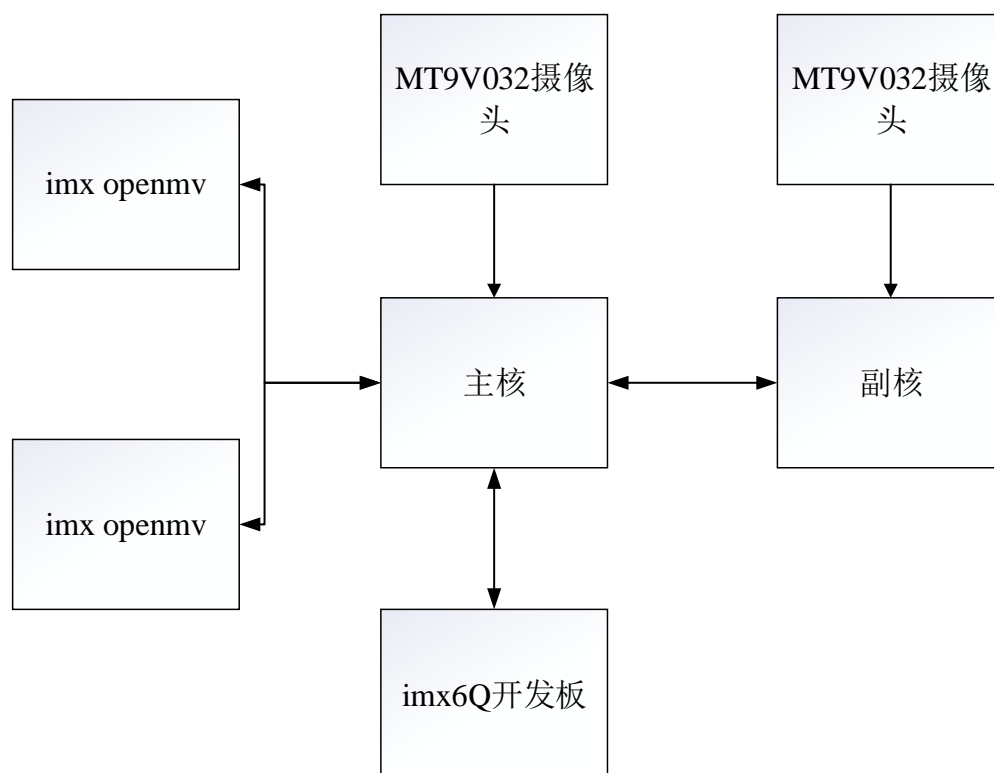


图 2 通讯框图



### 3 机械系统设计与实现

通过仔细阅读规则文件，我们熟悉了“我们要干什么”以及“有哪些限制”。之后通过小组成员们的讨论，我们确定了能够实现这些基本功能并在比赛中能占到优势的机械结构。之后，通过建模软件对所有零件进行 1:1 建模，再将所有零件在 CATIA 软件装配设计中进行装配。这样，就能在装配设计中提前发现设计中的干涉，尺寸不合理等问题，从而节省车身制作时间。装配好的车模如图 3 所示。

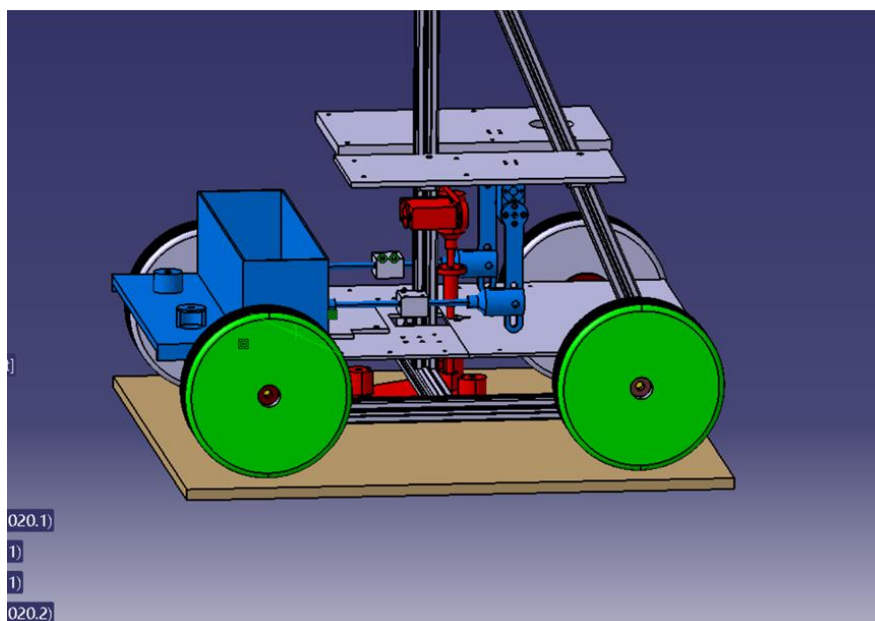


图 3 3D 模型图

对于所用到的零件，如果属于标准件，则按要求尺寸进行购买，如属于非标准件，则选择手工加工方式或 3D 打印，综合成本与零件尺寸公差要求选择加工方式，能够保证加工出来的零件符合使用要求。由于提前进行了建模与装配设计，在零件安装阶段很少出现干涉现象。

#### 3.1 车体结构设计及搭建材料选择

车体结构骨架由 2020 铝型材搭建。首先 2020 铝是标准件，与其配合的零件都可通过购买得到，省去许多加工的步骤；其次使用铝合金材料搭建车身骨架能够做到稳固牢靠而又比钢材轻，有较好的比强度。骨架搭建如图 4 所示。



图 4 车体骨架图

麦克纳姆选择方面，吸取学长的经验教训，我们选择了 152mm 直径的大直径麦轮。选择小麦轮的劣势在于同等电机减速比的情况下，小麦轮在电机占空比达到最高时，车速还是较低，无法在比赛中获取速度优势。因而我们选择了大直径麦轮，并通过将电机装在铝合金底座上部的方式，达到降低车身重心的目的，提高了小车的稳定性。

电机的选择方面，由于使用麦克纳姆轮，必须配置四个电机。我们通过网上寻找，最终确定使用 MD36 电机，如图 5 所示。该电机一大优势就是自带 500 线 AB 相光电编码器，无需外接测速装置，且内部自带行星减速齿轮实现减速增矩作用，输出转矩较大。



图 5 MD36 电机

MD36 电机主要技术参数如表 1.

表 1 MD36 电机技术参数

额定电压	24VDC
空载转速	630rpm
额定转速	450rpm
额定扭矩	7.2kg·cm
空载电流	0.3A
额定电流	2.3A
堵转电流	7A

最后，使用雕刻机加工环氧树脂板作为底板及核心板等电路板承载件，并使用 3D 打印技术制作完成一些结构复杂且对尺寸形状公差有较高要求的非标准零件，最后搭建完成整车如图 6、7 所示。

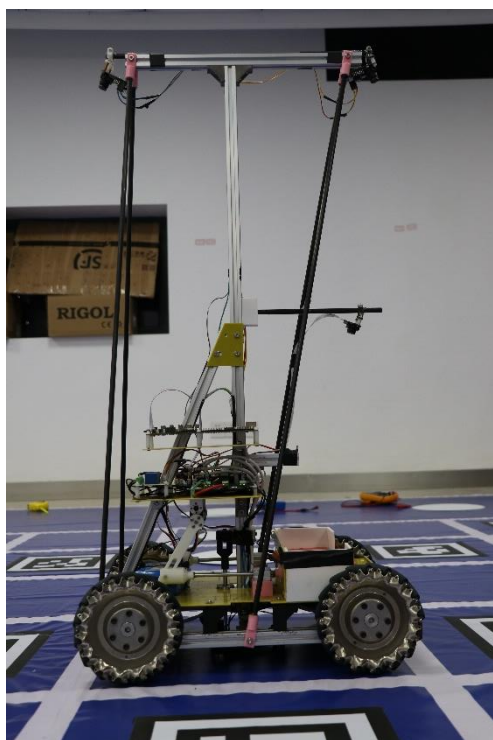


图 6 整体图

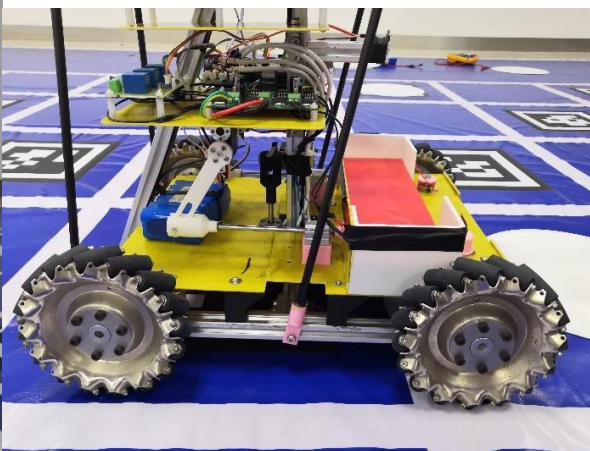


图 7 局部图

### 3.2 高处摄像头（车顶部扫描用）防抖设计

如图 4 所示，在小车顶部用来扫描用的摄像头，由于扫描需要，其固定位置离地面很高，这样就无可避免地带来了摄像头抖动的问题，在最开始的设计中，我们单独使用一根铝杆伸高再连接一横放的铝杆来放置摄像头，之后实际跑动过程中发现顶部摄像头抖动幅度过大，几乎无法识别。因此，我们对次进行了改善，为了兼顾整车质量降低和重心下移，我们没有采用直接使用多根铝杆伸高固定的方式，而是仿照斜拉索桥的原理，使用四根斜拉碳杆，从而既达到了稳定的目的，

又不增高车身质量和重心。

### 3.3 棋子拾取模块设计

创意组预赛与决赛中，均需要对棋子进行吸取，移动和摆放。根据规则中提供的方案，我们决定采用铁质棋子并使用电磁铁进行吸取和摆放棋子。为了简化机构，我们采用依次拾取、摆放的原则，从而减去棋子存储模块。

#### 3.3.1 棋子吸取方案

通过电磁铁的通电与断电可以实现棋子的吸取与释放。电磁铁的通电断电采用继电器来实现。本设计选用电磁铁为 DC-12V 供电，吸力可达 3kg,而白色棋子的质量仅为 1.6kg,因此电磁铁所能提供的吸力远远满足吸取白色棋子的质量。图 8 所示为使用的电磁铁实物图及其参数



图 8 电磁铁实物图及其参数

棋子吸取过程中采用三个磁铁同时作用实现棋子吸取，从而保证棋子姿位符合预期。搭载磁铁的吸盘如图 9 所示



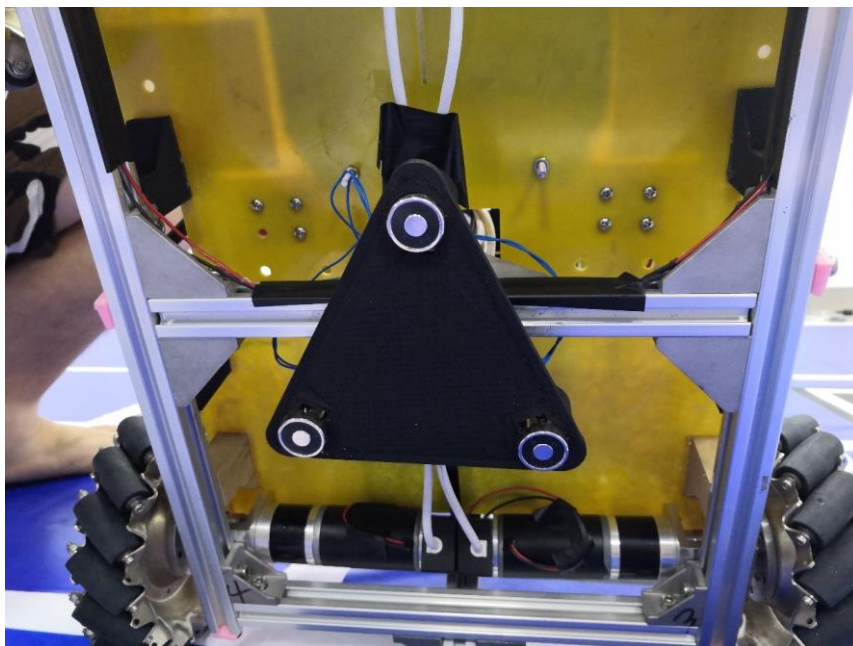


图 9 棋子吸盘

而棋子吸盘的上下运动的实现由舵机加曲柄连杆机构实现。使用舵机加曲柄连杆机构，相较于直线电机，其优点在于：第一，结构简单轻便，重量较直线电机大大减小。其二，由于舵机能够通过控制占空比精确控制转动角度，相较于直线电机为了确定吸盘位置还要附加一个电位计，结构上大大简化了。第三，舵机反应速度灵敏，吸取棋子速度非常快，通常舵机转过一个 60 度的角度时间都是毫秒级的，因此在拾取棋子时能够在非常短的时间内完成

舵机加曲柄连杆机构如图 10.

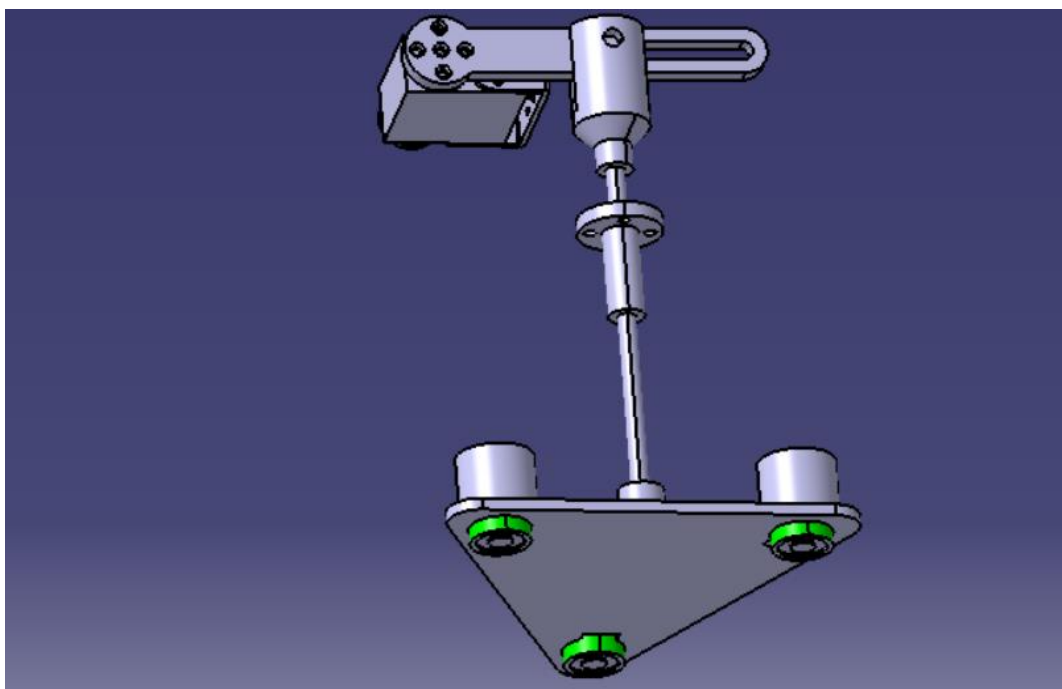


图 10 棋子吸取模块 3D 模型

### 3.3.2 缓冲结构设计

在吸取棋子时，由于电磁铁必须与棋子完全接触其吸力才能发挥最大的作用，为能使电磁铁与棋子完全接触，决定在电磁铁的支撑板和电磁铁之间设置弹簧，舵机臂下摆足够角度带动吸盘下移足够长度，由于弹簧的弹性作用，电磁铁与棋子可以完全接触，同时不会因直线电机伸出较长而顶起车模，影响整车行驶。缓冲结构主要靠弹簧实现，如图 11 所示。



图 11 缓冲结构

### 3.4 障碍板摆放装置设计

在创意组决赛阶段，需要车辆能够摆放挡板。由于规则要求挡板在摆放后能完全盖住赛道 5cm 白条，为保障摆放的质量，并防止由于偏斜而导致的为能盖住白条的情况，最终选择 90mm\*300mm 大小的挡板，同样的我们选择电磁铁吸取挡板和释放挡板。

根据规则，我们创造性地选择了一个可以同时储存多块障碍版并可自动依次摆放障碍板地机构，其工作原理类似于手枪上膛。原理如图 12，实物如图 13。

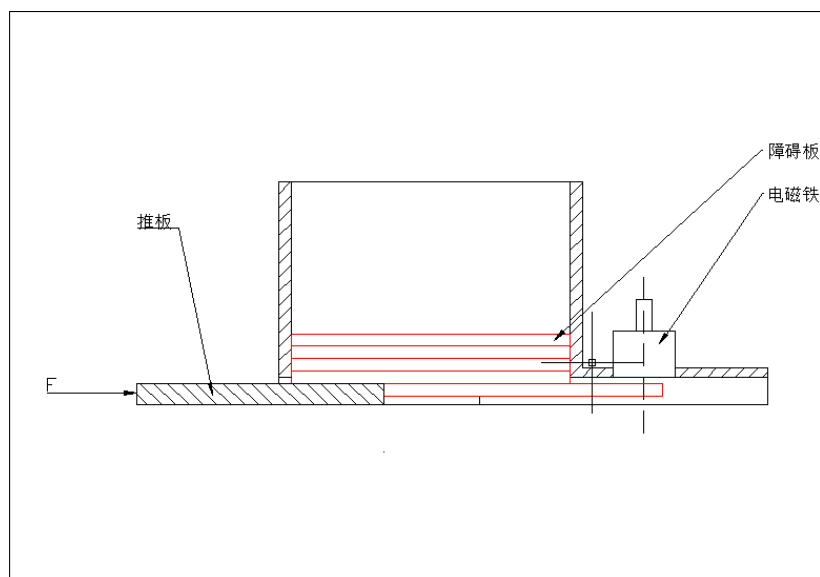


图 12 挡板放置原理图

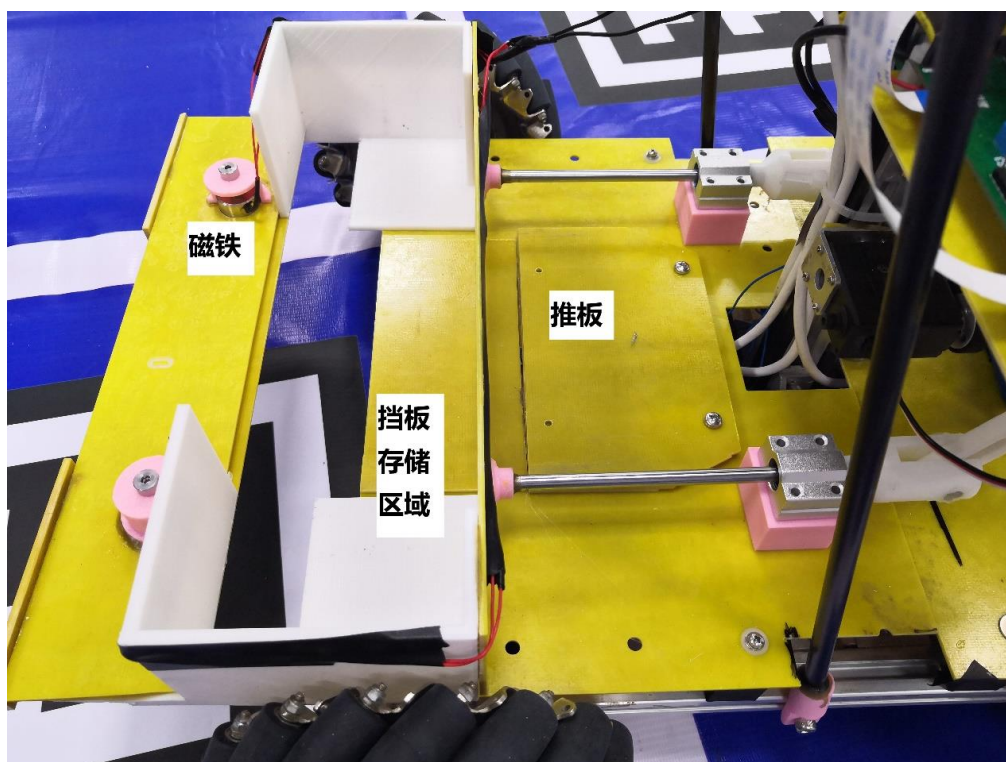


图 13 挡板放置实物图

如图 12，将多块挡板放于图示黄白色挡板盒子中，当舵机带动挡板盒子后移时，由于底板上突起一块推板（图中挡板盒子右侧），在挡板盒子后移过程中会将一块挡板推动到前方待摆放区域（即粉色圆柱状区域下方）被磁铁吸住，当挡板盒子向前移动到白线上方后，磁铁断电，挡板掉落在白线上方。之后挡板被往回拉，重复上一次动作。这样挡板便可以一次添加多次放置，省去了对弈中放置挡板的时间，从而获得一定优势。

## 4 硬件系统设计与实现

本系统按照组委会设计要求，采用官方指定的 MK60FX512ZVLQ15，为了保证单片机供电稳定、不受外围电路的影响，单独使用一片 TPS7333 稳压芯片来供电，因为 TPS7333 的低纹波电压不会对单片机信号产生外界干扰，经实际测试，单片机可以稳定正常的工作，不会出现供电不稳导致的复位和跑飞等现象。再使用一片 AMS1117-3.3 芯片来供给 IIC、摄像头、驱动芯片和 3.3V 预留。系统中如蓝牙、继电器模块、OLED 显示屏、电机编码器、蜂鸣器、OPENMV 等则使用一片 LM2596-5 芯片来进行供电，同时给 TPS7333 和 AMS1117-3.3 提供 5V 电压，LM2596 可以最大输出 3A 的电流，完全满足这些外设要求。同时由于使用多个舵机进行棋子搬运的需要，考虑舵机工作电流大于 0.6A，再使用一片 LM2596-5 来单独供给舵机工作，为防止电流过大对主控板产生电磁干扰，使用 TPS5450 耐流值高达 5A 的外接模块来解决。本系统采用双 K60 单片机，解读四路编码器。继电器模块是自己使用覆铜板焊接调试，未体现在原理图中，经实际检验电路运行正常。

驱动电路则是使用 BTN7971B 芯片构成驱动电路，为四个直流电机提供驱动能力。主控电路和驱动电路的原理图见附录。

### 4.1 主控电路设计

5V 供电采用的是 LM2596-5 降压稳压芯片，输出为 5V，最大输出电流可以达到 3A，宽输入电压 1.2V-37V，150KHz 内部固定频率，低功耗，具有热关闭和电流限制保护。基于该芯片优良性能，本系统采用两片 LM2596-5 为系统提供 5V 电源，电路原理图如图 14 所示。

3.3V 供电是由 TPS7333 和 AMS1117-3.3 产生。为保证单片机的稳定正常工作，不受电源纹波干扰，采用 TPS7333 单独供电，该芯片输出电流在 0-500MA，完全满足单片机的需要。IIC、摄像头、驱动芯片和 3.3V 预留则使用 AMS1117-3.3，该芯片线性度好、效率高、发热少、价格低，虽有纹波但是可以满足系统需要。电源输入采用的锂电池 24V，TPS7333 和 AMS1117-3.3 都是低压降芯片，故使用降压得到的 5V 来稳压到 3.3V，以保护电路。电路原理图见图 15。



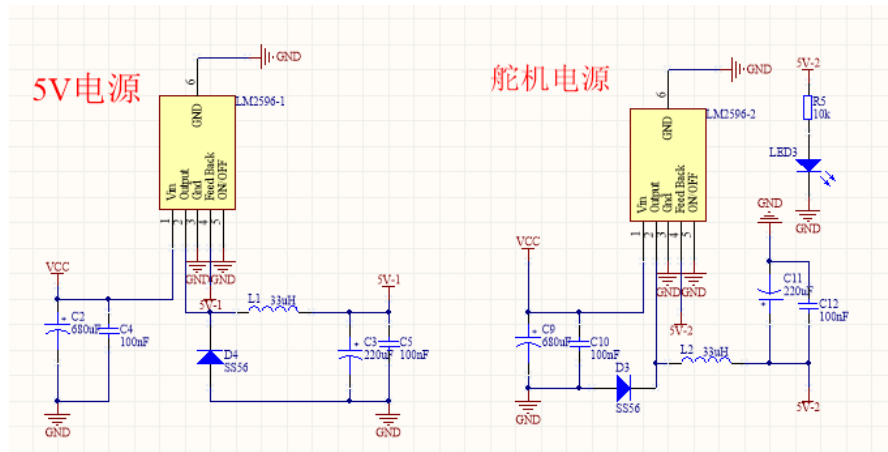


图 14 LM2596-5V 稳压电源电路原理图

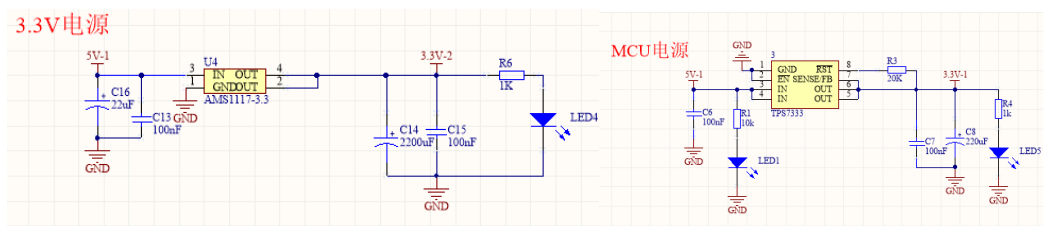


图 15 TPS7333 和 AMSA1117-3.3 电路原理图

## 4.2 最小系统电路设计

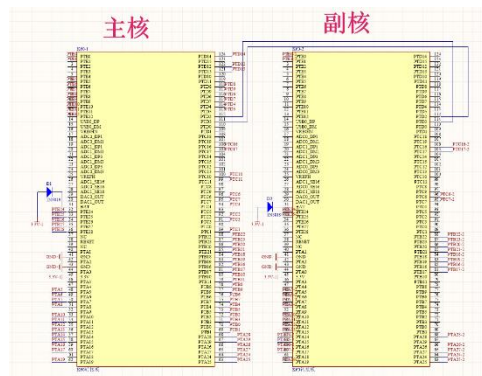


图 16 MK60 最小系统电路原理图

最小系统是整個系統工作最基礎、最核心的部分，它的穩定性決定整個系統的性能，所以設計必須十分穩定。由於 MK60FX512ZVLQ15 芯片價格昂貴，且設計需要逐步完善，因此選用核心板來降低成本，可以多次利用。設計的最小系統電路原理圖和實物分別如上圖所示。

注：本系統所使用的藍牙和 OLED 是自行設計的，未曾使用核心板自帶接口，見圖 17 和 19。

### 4.3 外设接口电路设计

为方便调试和传送数据，本系统设计了无线蓝牙接口和停车检测模块，并设计了 ADC 采集接口用于反馈舵机工作状态的信息，并预留了 IIC 接口。电路原理图见图 17。

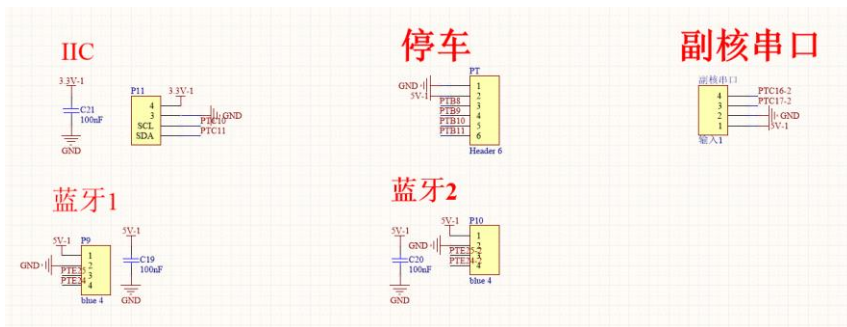


图 17 外设接口电路设计

### 4.4 输入辅助模块

在预赛和平时调试中，棋子位置是固定的，为了防止反复下载程序带来的不必要麻烦，故设计矩阵键盘手动输入，五向开关可以在调试时修改参数以节约时间，四路拨码开关设置多种工作模式，依据现场具体情况合理选择最优模式；电路原理图见图 18。

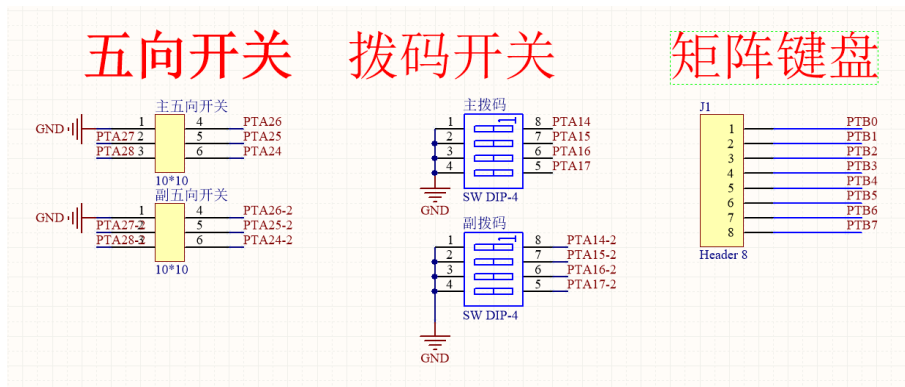


图 18 输入辅助模块电路原理图

### 4.5 输出模块电路设计

为方便平时调试，采用两块 OLED 进行人机交互，极大提高调试的效率和准确性；同时增加蜂鸣器报警电路，以便在车模运行过程中在关键位置报警来判断运行状况，电路原理图见图 19。

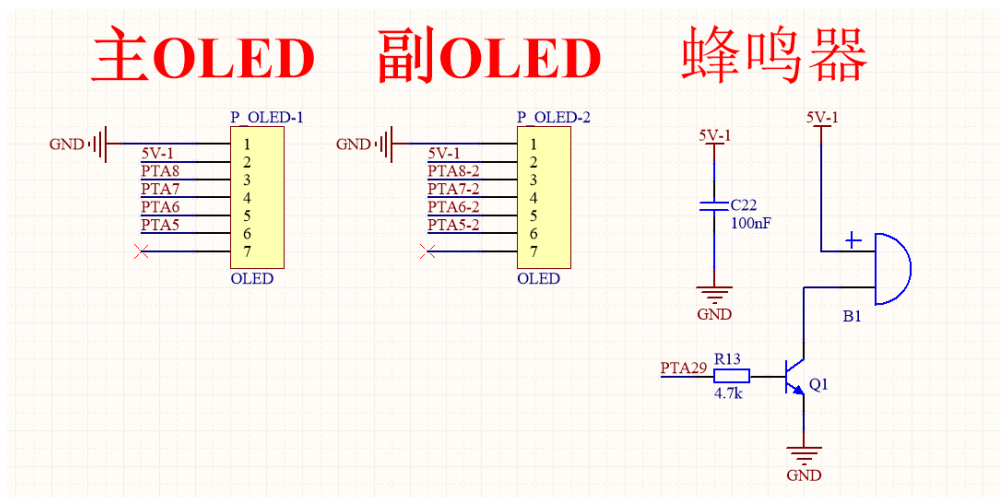


图 19 输出模块电路原理图

## 4.6 继电器模块电路设计

利用继电器的常开常闭和耐流值较大的特性，将其作为保护开关使用，供电回路接入三极管，控制三极管栅极电平即可控制继电器的常开常闭状态，电磁铁与常开触点串联，使用 24V 供电，依据电磁感应定律，吸力较大。电磁铁工作时电流较大，使用继电器可以有效地防止舵机瞬间启动的大电流击毁单片机，起到隔离保护的作用。电路原理图见图 20。

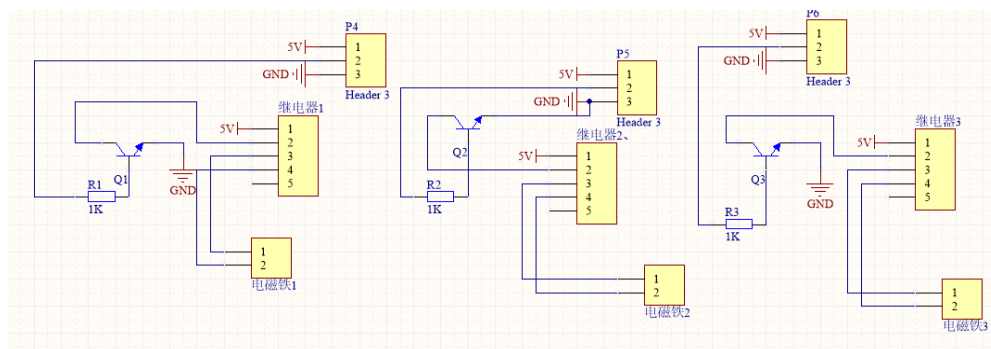


图 20 继电器模块电路原理图

## 4.7 驱动电路设计

驱动电路设计采用的是 BTN7971B 芯片，相较于 MOS 管驱动电路来说，BTN7971B 的外围电路简单，电路运行也比 MOS 管更加稳定，极少出现被静电击穿的情况，虽然成本较高，但其综合性能高于 MOS 管，出于稳定性的考量，决定使用 BTN7971B。

BTN7971B 是专用于电机驱动的大电流半桥高度集成芯片，内部集成一个 P 沟道高边 MOSFET 和一个 N 沟道底边 MOSFET 和一个驱动 IC。P 沟道 MOSFET

省去电荷泵的需要，减小了 EMI，集成驱动 IC 具有逻辑电平输入、电流诊断、斜率调整、死区时间和过温、过压、过流及短路保护功能，使用两片 BTN7971B 构成驱动 H 全桥驱动电路，以达到控制电机正反转的功能。电路原理图见图 21.

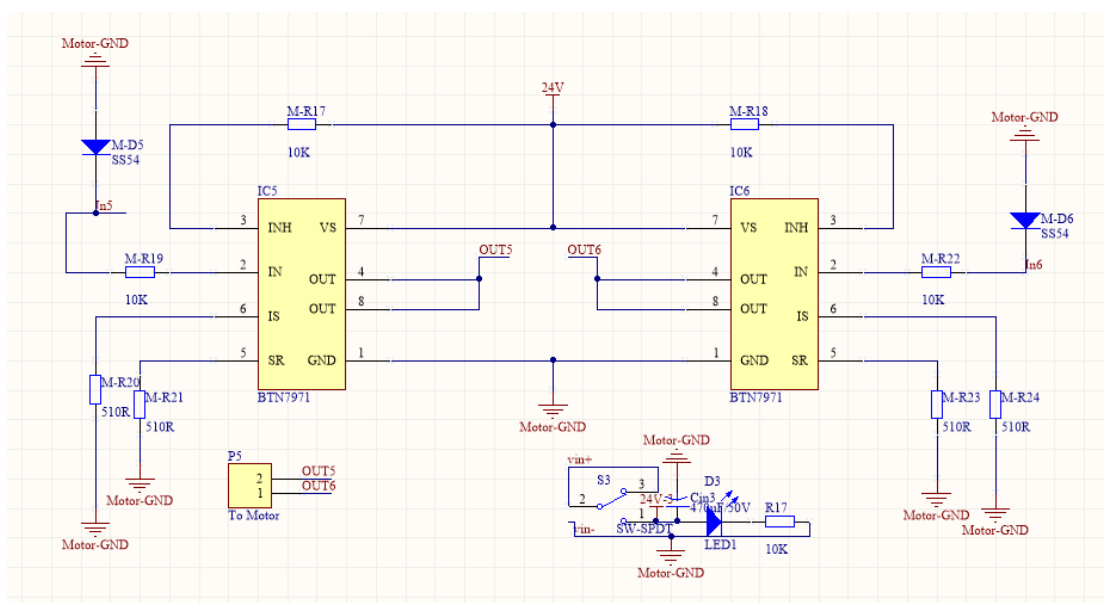


图 21 驱动电路原理图

BTN7971B 是大电流驱动电路芯片，为防止电机停转不及时引起电流倒灌烧毁单片机，采用电压隔离芯片 74LVC245 进行信号隔离，以提高电路安全性能。电路原理图见图 22。

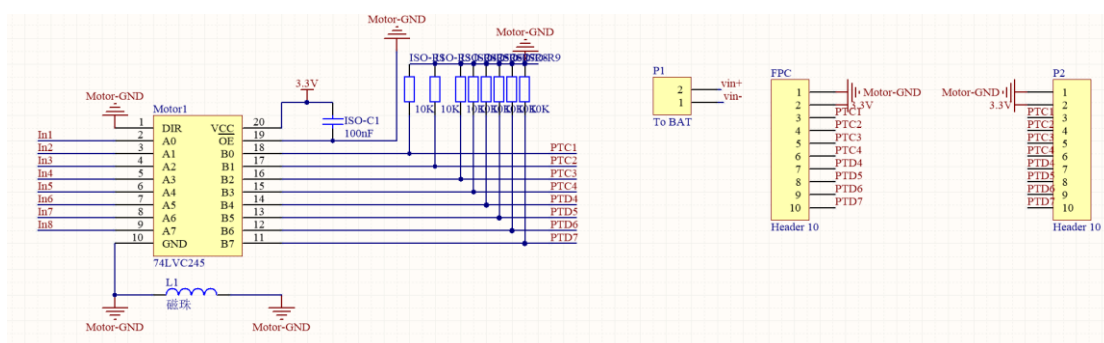


图 22 驱动隔离芯片原理图

## 5 图像处理与数字识别

已知竞赛环境由棋盘格构成，棋盘格中分布有 AprilTag 标签。比赛要求不可在场地外再铺设其他的辅助定位设备，因此控制系统需要识别 AprilTag 的信息来对于小车的位置信息以及综合信息进行判断。同时需要通过计算机视觉技术对于整个棋盘进行扫描，对于图像信息进行处理完成棋盘信息的录入工作。使得控制端通过处理过的图像信息进行处理。

### 5.1 处理器选型及参数

由于比赛要求参赛作品中的微控制器必须为 NXP 公司的微控制器系列产品。所以我们选择 i.MX RT1062 可编程视觉模块。其采用 OpenMV 的固件，内置有一些图像处理算法。其基本参数如表 2 所示。

表 2 可编程视觉模块基本参数

主控芯片	NXP i.MX RT1062, 600MHz Cortex-M7, 1MB RAM, 4MB Flash
感光芯片	OmniVision OV7725, 最高分辨率 VGA (640x480)
系统固件	OpenMV3 M7 Cam 集成了 Micropython 运行环境
通信	SPI, I2C, UART
镜头	M12 2.8mm 焦距

由于处理器的内存较小，在使用过程中如果使用像素较高的图像进行处理会导致内存泄露问题，因此我们将图片大小设置为 QQVGA，并关闭摄像头的自动白平衡和自动增益控制功能来增加摄像头在不同场景中的稳定性。

初始化部分代码:

```
sensor.set_framesize(sensor.QQVGA)
sensor.set_auto_gain(False)
sensor.set_auto_whitebal(False)
```

### 5.2 AprilTag 的识别以及信息处理

AprilTag 是一个视觉基准库，在 AR，机器人，相机校准领域广泛使用。通过特定的标志（与二维码相似，但是降低了复杂度以满足实时性要求），可以快速地检测标志，并计算相对位置。对于 AprilTag 的检测步骤分以下三步。

(1)检测线段：计算每个像素的梯度方向和大小，通过聚类将梯度大小方向相似的聚集到一起。

(2)检测正方形：整个任务转化为一个递归的、深度为 4 的深度优先搜索。第一层时，依次把每条线段作为起始线段。第二层至第四层，去找和这个起始线段终点（默认每条线段都是逆时针方向）足够近的线段。

(3)单应性及外估计：利用 Direct Linear Transform (DLT) 算法计算单应性矩阵。

在使用 AprilTag 时我们通过三种数据对于车的决策进行控制，分别为 AprilTag 的图像坐标位置、AprilTag 相对于小车的绝对坐标位置、AprilTag 的旋转角度信息。

### 5.3 棋子与挡板识别

因为棋盘的背景为蓝色，亮度较低，而棋子为白色，亮度较高。所以我们采用灰度图来对图像进行处理，不仅可以节省开发板的内存空间，同时加快处理速度。为了减少阳光、灯光对于灰度值判断的影响，通过减少快门时间，减少入光量来减少灰度阈值选取的误判。

由于采用寻找圆形的算法会消耗较多的计算资源，当棋子在棋盘中被摄像头捕捉到时存在一块较大的白色色块区域，通过选取合适的阈值对于色块的大小和像素进行判断可以很好的完成对于棋子的判断，并且扫描速度相对于寻找圆形的速度有较大的提高。

对于棋子位置的判断，我们通过获取图片中色块的中心图像坐标和色块周围的 AprilTag 的坐标对于棋子在棋盘中的位置进行判断，如图 23 所示。完成棋盘位置的扫描工作。对于挡板的位置判断，我们通过对于红色色块的寻找并结合周围的 AprilTag 的信息，录入挡板位置信息。



图 23 寻找棋子

### 5.4 车身位置确定及姿势控制

在完成任务的过程中，最重要的是完成智能车的行驶过程中位置识别。在采用陀螺仪对于车声姿态进行矫正的时候，由于电子设备的电磁影响，陀螺仪定

位的精度和速度受到干扰。我们结合 AprilTag 返回的标签旋转信息与陀螺仪的信息相结合，改善智能车旋转动作消耗时间，同时精确确定智能车相对于棋盘的旋转角度。

结合摄像头镜头焦距得到的 AprilTag 相对于小车的现实坐标可以用于对智能车当前的行驶位置进行确定。将获得现实坐标系进行标定与拟合后，根据多个 AprilTag 标签的返回信息可以精确确定其在棋盘坐标系中的方位，由于摄像头远处的画面存在镜头畸变的问题，影响返回坐标的精度，所以我们搜索最近的 Tag 来进行小车方位的确定。在行驶至目标区域后，再结合棋盘边界的白色边框对于小车在棋盘内的位置进行精确定位。

定位部分代码:

```
rotation=180*tag.rotation()/math.pi  
x=int(128+tag.x_translation()*10)  
y=int(128+tag.y_translation()*10)
```

## 6 运动控制系统设计

### 6.1 运动控制程序设计总体思路

本系统的主要任务是移动棋子或挡板准确到达棋盘指定位置。我们以棋盘建立直角坐标系。通过摄像头确定车身当前位置，根据路径规划算法和下棋策略算法得出车子想要到达的位置，计算出  $x$  轴、 $y$  轴的偏差即可控制麦克纳姆轮运动到指定位置。

八皇后问题中，路径规划算法基本思想是先将目前棋子所在位置与所有八皇后正解比较，选择重合多且路径相对较少的正解，然后再根据离车模当前位置最近的棋子或解的原则，选择最近的路径，并且给  $X$  方向和  $Y$  方向的路径距离乘某个系数，使得车模更愿意走某个方向的路径。因为由于机械原因，车模走不同方向的速度极限值不同。

步步为营下棋策略算法将在下文中详细介绍。

综上所述，本系统设计的关键在于图像识别与路径规划、速度控制。速度控制是基础，速度控制是否精确直接关系着能否完赛。而路径规划可以决定车模能否按照规则以最短时间完成比赛。

### 6.2 运动控制流程设计

比赛规则要求车模必须自主运行，不能人为控制，这就要求我们需要设计一套可靠的运动流程以保证能够按照策略要求执行特定的动作。本车模设计有三种运动模式：扫描模式，下棋模式与放障碍模式。

在预赛任务中，车模先扫描一遍棋盘，在获得满足条件的棋盘位置后，跳出扫描模式进入下棋模式，运行路径规划算法。

在决赛中，车模得到运动指令，在扫描完棋盘获得棋子与障碍位置后，回到起点等待上位机给出下一步运动指令，不同的任务分别对应移动棋子与放置障碍。

### 6.3 车模精确运动实现

为使车模定位准确、快速。我们需要精确的知道车模当前的状态，包括速度，车身角度和车身位置。



### 6.3.1 车模测速

由于单块 K60 只有两路正交解码，满足不了对四个麦克纳姆轮测速的要求，所以我们采用双核方案。在运动过程中，副核通过串口，高频率地向主核发送两个轮子的转速。通过上位机对比主核测得的车速和副核发过来的车速，可以看到两个数据最大只相差 2。而这种误差正是由机械原因造成的，所以双核测速的方案切实可行。

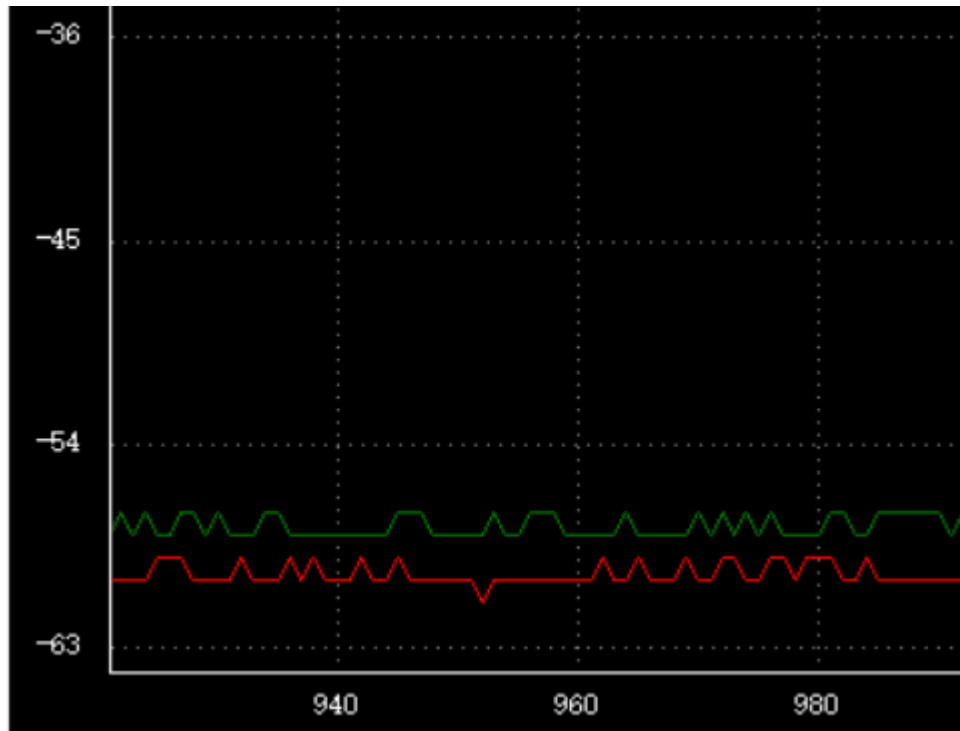


图 24 测速图

### 6.3.2 车模角度环

对车模精确运动影响最大的一个方面就是车身偏航角。在车模调试前期，我们计划用陀螺仪检测车模的偏航角。但是陀螺仪测得的偏航角存在很大的漂移，才尝试了多款陀螺仪都没有取得理想的效果后，我们决定采用从标签中获得角度信息。在实际的调试过程中，这种方法比较可靠，能够满足控制车身角度的要求。我们设置了角度环，通过  $pd$  控制，使车身在运动过程中保持正直，不随便乱晃。同时还要有足够的灵敏度，在放置棋子与挡板的过程中能够快速调整车身。角度环如下：

```
void SysCtrl_Angle_PID(YesNo_t En_Turn)
{
    if(En_Turn==Yes)
```

```

{
    PID_Update_Sample(&SysPID[Head_Angle], actual_angle , (YesNo_t)Yes);
    turn_over=0;
    if(Camera_H_B_Flag==0)
    {
        if(right_ninth_ed==1)
            SysPID[Head_Angle].set=Angle_to_Ctro_H_t;
        else
            SysPID[Head_Angle].set=Angle_to_Ctro_H;
        if(HeadCamera_Get==1)
            SysPID[Head_Angle].sample=Get_Angle_H/100;
        else
            SysPID[Head_Angle].sample=SysPID[Head_Angle].set;
    }
    else if(Camera_H_B_Flag==1)
    {
        if(right_ninth_ed==1)
            SysPID[Head_Angle].set=Angle_to_Ctro_B_t;
        else
            SysPID[Head_Angle].set=Angle_to_Ctro_B;
        if(BackCamera_Get==1)
            SysPID[Head_Angle].sample=Get_Angle_B/100;
        else
            SysPID[Head_Angle].sample=SysPID[Head_Angle].set;
    }

    PID_Update_Error(&SysPID[Head_Angle], (YesNo_t)Yes);

    Dof_PID_Update_Calculate(&SysPID[Head_Angle],SysPID[Head_Angle].EnPid_Out);
}
}

```

### 6.3.3 车模直线行驶

车模在扫描过程中需要保持直线运动，且车模在放置挡板过程中需要更高的精度。所以我们采用两个 MT9V032 摄像头，分别检测车身前方与左侧的直线。棋盘格横平竖直，对检测直线有很大的方便，但是棋盘、挡板与标签会造成很大的干扰。常用的直线检测有霍夫变换，可以很容易的完成直线检测任务，但是霍夫变换运行时间长，在 K60 中运行并不现实。根据标签白边比边线更细且标签内黑白色块交替存在，可以滤去标签的干扰。得到竖边线可以求出中线，是车身沿直线行驶，检测到横边线可以控制车身与棋盘格的距离，提高放置挡板与棋子的精度。得到的图像与边线如图 25 所示：



图 25 棋盘格及边线二值化图

## 6.4 车模运动控制

实现车模定点运动有两种方法：

(1)通过编码器累计脉冲，测得车模实际运行的距离。以精确控制车模运动固定的距离。

(2)通过摄像头实时返回车身位置，比较车身当前位置与目的位置的偏差，控制车模运动到预定位置。

比较这两种方法的优缺点，编码器测距简单易行，但是难以精确获得对应位置的设定距离；实时检测车身的位置可以非常准确的定点停车，但是摄像头视野有限，远距离运动难以实现。

综上，我们决定结合上述两种方法。先通过编码器粗略的运动到目标点附近，然后再通过摄像头实时矫正车模位置。经过反复实验与调试，达到了理想的效果。

我们的车模运动共有四种状态。分别为：加速状态、定点状态、矫正状态与执行任务状态。车模默认为加速状态，在收到移动指令后，车模以最大加速度完成加速，在以最大速度行驶过程中，离目标一段距离后开始做减速运动；当距离降到设定值，开始进入定点状态，车模利用标签信息开始调整车身以准确停在方格内；满足一定的精度要求后，开始进入矫正状态，这一步对挡板放置尤其重要。

通过车身上安装的两个摄像头检测到的边线对车身精确控制；最后执行特定的任务，吸取棋子、放置棋子或者放置挡板，执行框图见图 26。

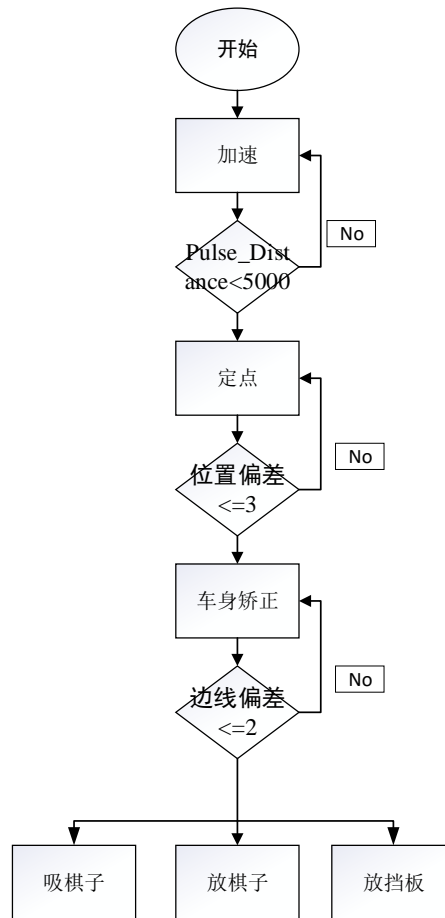


图 26 车模运动控制框图

## 6.5 运动方向控制

由于车模车轮使用的是麦克纳姆轮，由麦克纳姆轮的特性可以非常容易控制车模实现前、后、左、右、顺时针旋转、逆时针旋转的简单运动，然后再由这些简单的运动合成任意方向的复杂运动，便于控制。车模运动方向的控制是根据车模当前位置与目标位置决定的，由车模当前位置与目标位置可知  $X$  方向的距离和  $Y$  方向的距离以及直线距离  $L$ ，然后即可由车模的实际速度与方向推导出车模  $X$ 、 $Y$  方向的分速度（总速度/ $X$  方向分速度=直线距离  $L$ / $X$  方向的距离，总速度/ $Y$  方向的分速度=直线距离  $L$ / $Y$  方向的距离），将分速度与偏航角控制的旋转速度相叠加得到四个车轮相应的速度，然后通过  $PI$  控制器最终输出到电机上。

## 6.6 路径规划方案

“八皇后约束”即八个棋子在棋盘上的位置不在同一行、同一列、同一斜线，若利用单片机实现此解析耗时久，所以将所有满足八皇后的解（92 种）全部存储到数组中，程序可以直接查询，节省了计算量和时间。车模扫描棋盘获得棋子位置，程序将棋子的位置与所有八皇后正解比较选出重合度最高的解（至少一种但不仅限一种）。然后程序在模拟所有入场位置（棋盘边缘）下，计算车模移动的总距离，并选出最短的距离对应的入场位置作最佳入场位置，并在 OLED 上显示。根据入场位置计算车模移动距离的方法为：根据入场位置算出离入场位置最近的棋子并记录距离，再根据棋子位置算出最近的正解位置并记录距离，再记录当前车模位置和最近的棋子的距离，如此往复循环直到将所有不重合棋子归位。再根据所记录距离之和从而比较得出路径最近的解并记录此路径用于车模移动，演示如图 27 所示。本系统路径规划的根本原则为离当前车模位置最近者优先。

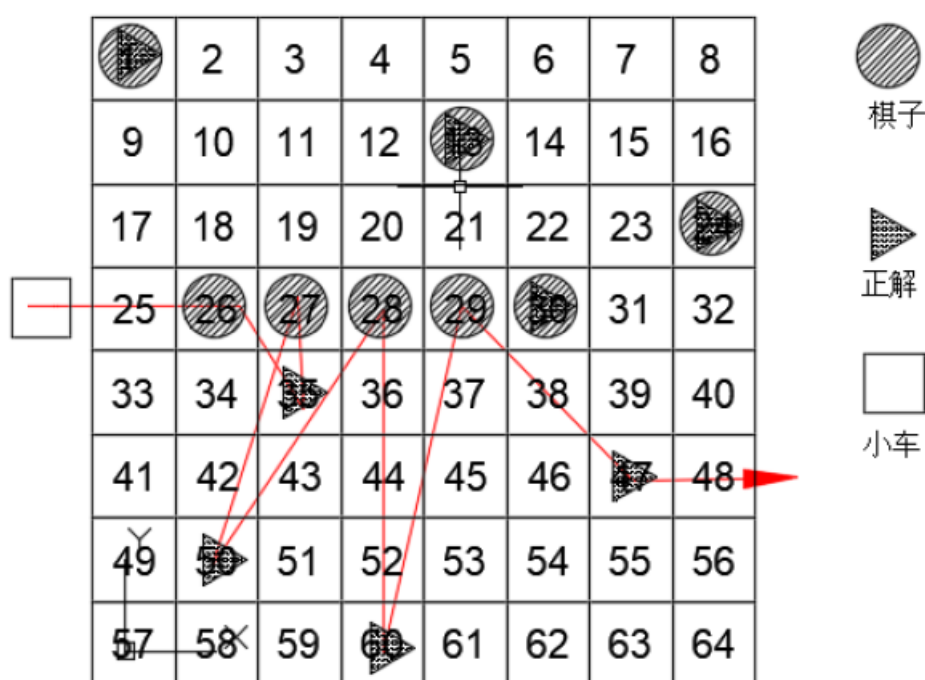


图 27 路径规划演示

## 6.7 车模调试

### 6.7.1 开发工具

程序的开发是在组委会提供的 IAR Embedded Workbench 下进行的，包括源程序的编写、编译和链接，并最终生成可执行文件。IAR Systems 是全球领先的嵌入式系统开发工具和供应商。公司成立于 1983 年，提供的产品和服务涉及到嵌入式系统的设计、开发和测试的每一个阶段，包括：带有 C/C++ 编译器和调试器的集成开发环境(IDE)、实时操作系统和中间件、开发套件、硬件仿真器以及状态机建模工具。IAR Embedded Workbench（简称 EW）的 C/C++ 交叉编译器和调试器是今天世界最完整的和最容易使用专业嵌入式应用开发工具。EW 对不同的微处理器提供一样直观用户界面。EW 今天已经支持 35 种以上的 8 位/16 位 32 位 ARM 的微处理器结构。EW 包括：嵌入式 C/C++ 优化编译器，汇编器，连接定位器，库管理员，编辑器，项目管理和 C-SPY 调试器中。使用 IAR 的编译器最优化最紧凑的代码，节省硬件资源，最大限度地降低产品成本，提高产品竞争力。

### 6.7.2 调试工具

通过 IAR 软件的在线调试功能，我们可以实时观测数据。并且通过单步运行，设置断点，运行到指定命令等功能，可以很好的解决 bug。合理高效的使用 IAR 软件给我们的调试带来了事半功倍的效果。

我们通过 NRF 无线发射接收模块将数据发送到上位机。NRF 模块相比蓝牙传输速度更快，连接更稳定。给我们的调试带来了很大的方便。

我们使用 FreeCars 上位机观测数据。FreeCars 上位机有虚拟示波器功能，且能观看图像。方便好用。

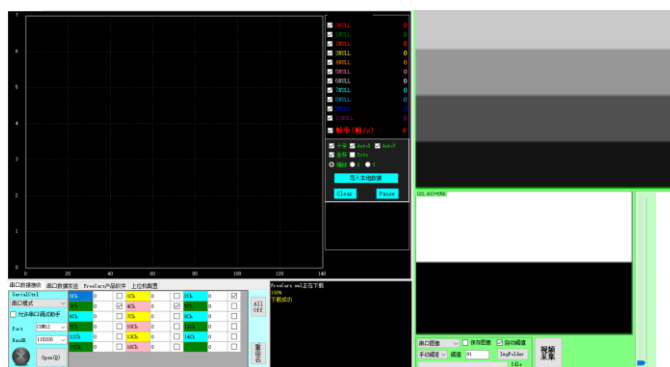


图 28 Freecars 上位机

## 7 预赛阶段针对八皇后系统的特殊设计

### 7.1 棋盘防滑及棋子吸盘设计

在预赛过程中，主要任务便是摆放棋子八皇后问题。在摆放棋子过程中，由于路径规划问题，有时无可避免地小车会轧过棋子，而由于我们的小车很重，轧过铁质薄棋子的时候会将棋子碰动。因此我们将棋子底部贴胶带，因为胶带的材质特性，当车子轧上棋子后，棋子与场地地面产生很大摩擦力，不易使棋子移动。

其次，棋子吸盘上的缓冲机构能够保证棋子吸盘能够足够长度地下压，保证棋子被吸盘吸住。并且使用舵机来带动吸盘，能够做到准确而灵敏地控制。

### 7.2 场地边缘识别棋子特殊情况处理

如果在任务过车中车模到达边缘位置时，若只用一个摄像头则无法实现获得精确位置，除非让车模转向或者摄像头转向，但这样会延长完成任务时间。所以本系统采用两个摄像头前后放置，若车模到达上边沿则此时用后面摄像头即可，若在下边沿则用前面摄像头即可，双摄像头实物如图 29。

当车模运行到棋盘左上角，如图 29 位置。此时无法通过下面两个摄像头检测两竖边线来控制车身位置。因为摄像头角度偏下，因此此时可以通过检测棋盘格的横线来控制车身位置。

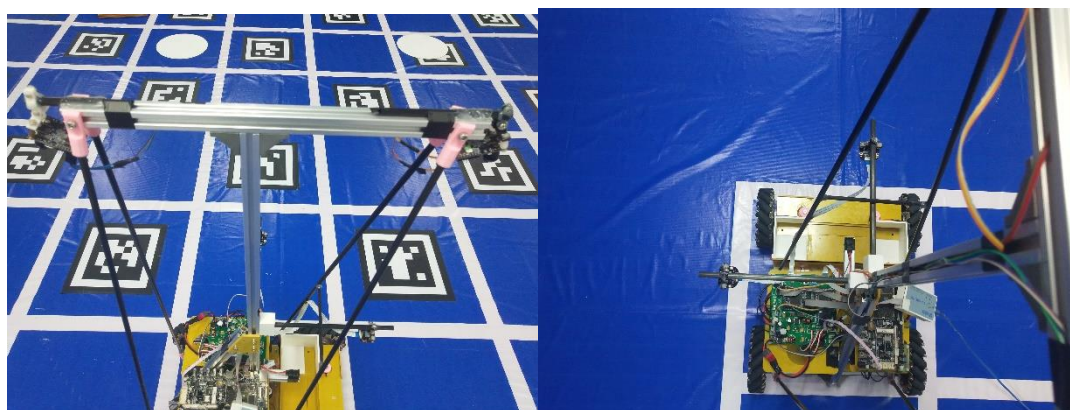


图 29 双摄切换及边线检测

### 7.3 双摄切换方案

正如上文中所介绍的，如果不希望车身转向的话，我们必须使用前后两个摄

像头。在正确、合适的时机切换摄像头，即选择使用哪个摄像头的数据便成了我们需要注意的问题。

我们切换摄像头的方案为：

- 1、 在车运动之前，执行一次摄像头切换函数。根据当前的位置确定使用哪个摄像头。
- 2、 在车快要到达目标点之前，再执行摄像头切换函数。根据目标点的位置再选择使用哪个摄像头。
- 3、 在一个摄像头没有数据时，立马切换使用另一个摄像头。

通过实际验证，这种方案保障了车模的稳定运行。



## 8 决赛阶段针对对抗策略的特殊设计

### 8.1 特殊的挡板盒子设计及挡板的相关改进

首先挡板盒子地特点在前文已经详细叙述了。而要使挡板盒子工作不卡壳，就要保证足够地精度。并且挡板厚度也是主要的影响因素。如果挡板厚度太大，那么必然会很重并且车子一碰就会跑，甚至阻碍车子移动。而如果挡板厚度过小，则推出时要保证非常小的误差，推板稍高一些则一次将推出两块挡板。最终我们选定 2mm 厚并在中部切割一个空格的设计，材料使用环氧树脂版。这种方案推出十分流畅且重量也不是很重，符合预期，设计如图。



图 30 挡板成品图

### 8.2 棋盘坐标变换

在决赛中，为了缩短下棋时间，在下完一步棋后，车模不可避免地需要按照最短路径驶出 8x8 棋盘范围外。为了做到车模可控并且精准运动。我们将原来 8x8 的坐标系扩充为 10x10 坐标。这样棋子在边线之外一格距离内都有对应的坐标位置。在对车模的控制上带来了很大的方便。

由于摄像头扫描传过来的数据是以 8x8 棋盘为基础的。我们设置函数，可以使两种数据相互转化。转换公式为：

```
source_temp.x=source.x+1;
source_temp.y=source.y+1;
num = (source_temp.x+source_temp.y*10+1);
```

其中：source.x, source.y 为 8x8 坐标；source\_temp.x, source\_temp.y 为 10x10 坐标；num 为转换后棋格对应的数字。

### 8.3 处理器选择

考虑到需要较高的主频及内存，我们选择了荣品公司的一款 IMX6Q 开发板搭载 Ubuntu 平台运行步步为营算法。

PRO6Q 开发板-CPU 主控为 imx6q 飞思卡尔 Freescale cortex-A9 四核处理器，主频 1HZ 广泛应用于工控车载医疗行业的评估板，邮票孔焊接设计稳定性更强，标配 2GB DDR3 内存及 16GB EMMC 存储，板载 Android 4.4 系统，可选系统 Ubuntu12.04、Linux+QT4.8。板载功能齐全，支持 RGB/LVDS/MIPI/HDMI 显示、以太网、IR、双路 CAN BUS、SATA、TF 卡扩张、SIM 卡、蓝牙 WIFI、摄像头、GPS、库仑计、可搭配 5 寸 10 寸屏、4G 模块、6000mA 电池等。

性能配置：

处理器---采用 Freescale Cortex-a9 四核 i.MX6Q 1GHz 处理器，性能强悍

存储配置---2GB 内存单通道 32bit 数据总线 DDR3，16GB 固态硬盘高速 eMMC4.5 存储

### 8.4 算法设计

一般解决博弈类问题的自然想法是将格局组织成一棵 Game Tree，Tree 的每一个节点表示一种格局，而父子关系表示由父格局经过一步可以到达子格局。

Game tree 的大小主要由两个数字决定

特征深度  $d$ ，这代表了游戏典型的走棋步数

分支因子  $b$ ，这代表了游戏典型的每步走棋的可行走法

可表示为  $b^d$ ，对步步为营游戏该值是个天文数字

Minimax 也不例外，它通过对以当前格局为根的格局树搜索来确定下一步的选择。而一切格局树搜索算法的核心都是对每个格局价值的评价。

每一个棋盘的状态都会对应一个分数。双方将会轮流下棋。轮到我方下子时，我会选择分数最高的状态；而对方会选择对我最不利的状态。可以这么认为，每次我都需要从对手给我选择的最差（min）局面中选出最好（max）的一个，这就是这个算法名称 minimax 的意义。

Minimax 不找理论最优解，因为理论最优解往往依赖于对手是否足够愚蠢，Minimax 中我方完全掌握主动，如果对方每一步决策都是完美的，则我方可以达到预计的最小损失格局，如果对方没有走出完美决策，则我方可能达到比预计的最悲观情况更好的结局。总之我方就是要在最坏情况中选择最好的。

由于 Minimax 非常完美的理论特性，后续的改进几乎都是基于 Minimax

的框架。既然 **Minimax** 的复杂度主要取决于特征深度  $d$  和分支因子  $b$ 。减少分支因子  $b$  的主要方式有两个：

(1)剪枝。剪枝就是去除掉某些明显劣势的走法。最著名的是 **Alpha-beta** 剪枝，它充分利用了 **Minimax** 算法的特点，并且仍然可以得到和 **Minimax** 相同的结果（也就是不是近似），是首选的优化。另外也可以用估值函数减掉某些非常坏的走法的分支。

(2)采样估计。其思路是蒙特卡洛(monte-carlo)模拟。我们可以先预估出一个先验概率分布  $P$ ，然后再根据  $P$  采样部分路径用作估计。这种手段对分支因子的减少比较激进，在围棋等分支因子较大的棋类中较常用。**Alpha** 系列采用的就是此类方法。

每一个节点都会由  $\alpha$  和  $\beta$  两个值来确定一个范围  $[\alpha, \beta]$ ， $\alpha$  值代表的是下界， $\beta$  代表的是上界。每搜索一个子节点，都会按规则对范围进行修正。

**Max** 节点可以修改  $\alpha$  值，**min** 节点修改  $\beta$  值。

如果出现了  $\beta \leq \alpha$  的情况，则不用搜索更多的子树了，未搜索的这部分子树将被忽略，这个操作就被称作剪枝。

本算法设计的 AI 分为 3 个级别

(1) Easy- RandomAIPlayer - which plays random (valid) moves -

(2) Medium- NoLookAIPlayer - which evaluates the optimal move to play without looking ahead

(3) Hard- MultipleLookAIPlayer - which evaluates the optimal move to play with multiple looks ahead

AI 评分取决于两个因素：

1.该玩家距离目标的距离

2.他们剩下的障碍

可表示为  $eval = weight1 * wallsLeft - weight2 * distanceToExit$

基于这种思想，我们在 Ubuntu 上编写了 java 程序，其中估值函数和 **Alpha-beta** 剪枝算法分别如图 31 和图 32。

```

protected int awesomeness(Board board, Player player) {
    int awesomeness = 0;

    awesomeness += board.getWallsLeft(player) * wallsLeftWeight();

    Space otherSide;
    if (player.equals(board.players._1())) otherSide = board.player2Start;
    else otherSide = board.player1Start;
    ArrayList<Space> exits = new ArrayList<Space>();
    for (int i = 1; i <= 7; i++) exits.add(new Space(i, otherSide.row()));
    board.graph.fillNodeDistances(exits);
    int distance = board.graph.node[board.getSpace(player).row()-1][board.getSpace(player).col()-1].distanceToExit;
    awesomeness -= distance * distanceWeight();

    if (player.equals(board.winner())) awesomeness = Integer.MAX_VALUE/2;

    return awesomeness;
}

```

图 31 估值函数设计

```

/*
private int negamax(Board board, int alpha, int beta, int depth) {
    i++;
    if (board.currentPlayer().equals(board.winner())) return Integer.MAX_VALUE;
    else if (board.players.other(board.currentPlayer()).equals(board.winner())) return Integer.MIN_VALUE + 1;
    else if (depth <= 0 || System.currentTimeMillis() - startTime > timeOut) return board.currentPlayer().minMax() * evaluate(board);

    Board newBoard;
    List<Move> potentialMoves = board.currentPlayer().allMoves(board);
    Collections.sort(potentialMoves, new MinimaxComparator(board.getSpace(board.players.other(board.currentPlayer()))));
    int evaluation;
    for (Move move : potentialMoves) {
        try {
            newBoard = board.clone();
            newBoard.makeMove(move);
            evaluation = -negamax(newBoard, -beta, -alpha, depth - 1);
            if (evaluation > alpha) alpha = evaluation;
            if (alpha >= beta) return alpha;
        }
        catch (Exception e) {}
    }

    return alpha;
}
*/

```

图 32 Alpha-beta 剪枝算法

对 Random AIPlayer, weight1,weight2 均为 0.

对 NoLookAIPlayer, 可以将 weight1,weight2 设置为 1, 5

对 MultipleLookAIPlayer, 最大可设置考虑之后的 3 步

程序中调整的一组  $\alpha\beta$  剪枝算法设置的值:

$$\alpha = -2147483647, \beta = 2147483647 \quad (0.1)$$

经测试效果较好

## 9 其他特殊设计

### 9.1 摄像头稳定结构设计

顶部扫描用摄像头安装在顶部横杆上，使用标准件连接摄像头与碳素杆。考虑到摄像头很轻，不用使用螺钉之类的也可以固定，故而将碳素杆用 AB 胶连接到顶部横杆上。如图 33 所示。

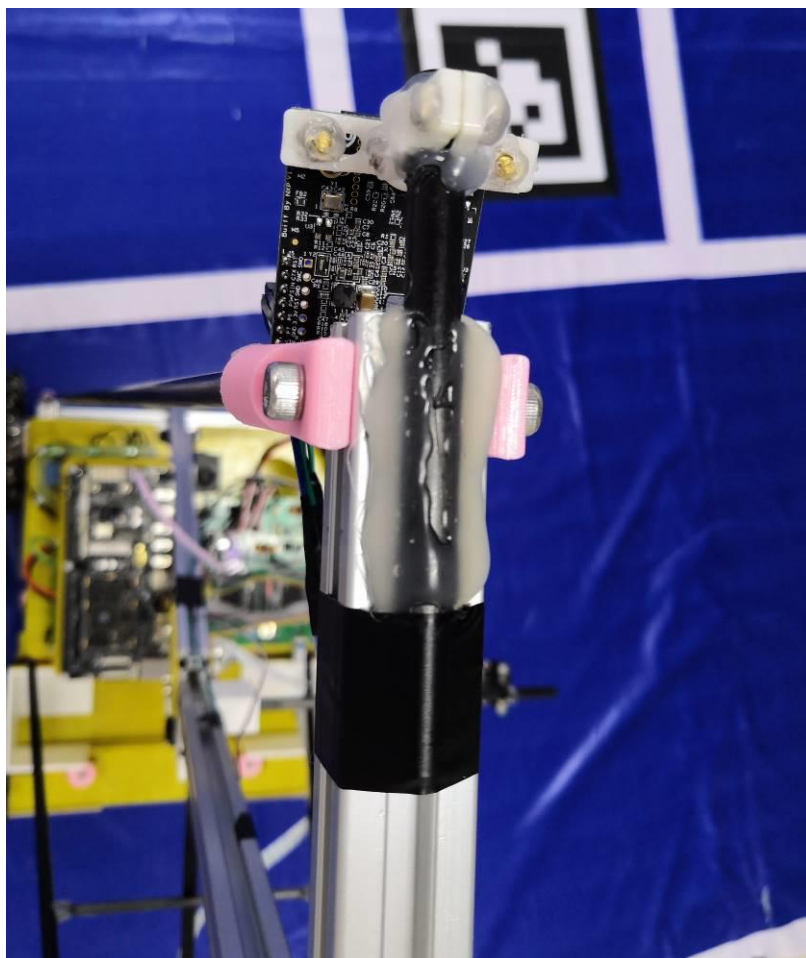


图 33 碳素杆与横杆固定图

顶部摄像头与横杆为一体，使用 AB 胶固连后刚度高，在防抖上有较佳作用。且对于运输来说，在运输过程中可单独将顶部横杆拆下。由于顶部横杆与车体竖杆采用标准件—2040 铝角接机件连接，拆卸方便且安装时可以百分百还原，对摄像头位置影响低，不需要再次调节摄像头。因此这样的模块化安装非常有利。

此外，顶部横向稳定杆与车体直接采用 4 根碳素杆斜拉连接，如图 34 所示。

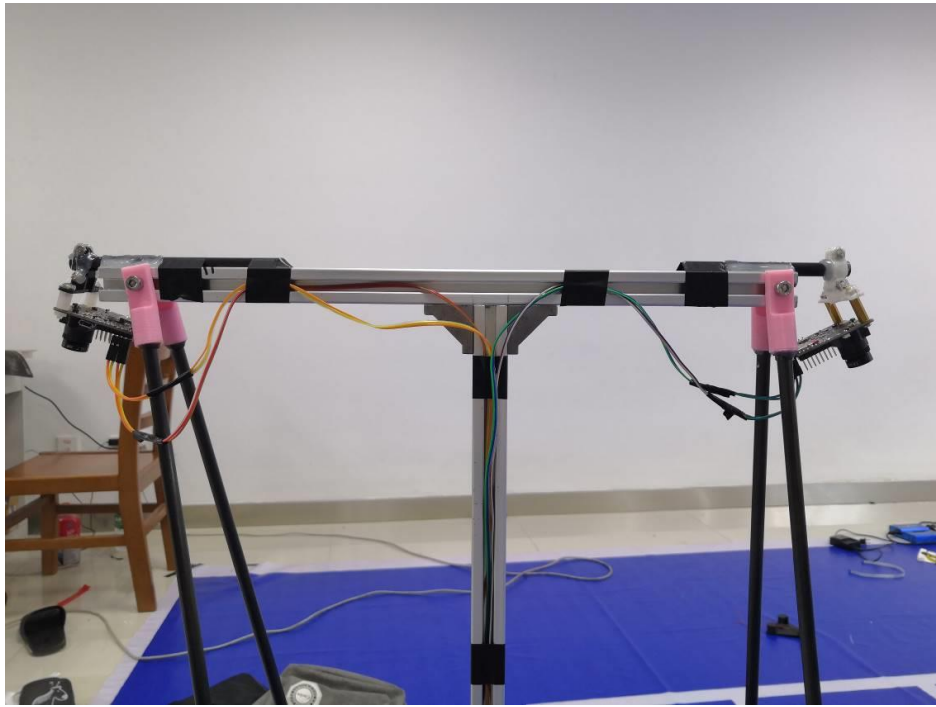


图 34 稳定结构

这样的连接方式为稳定性带来了很大的益处。

而碳素杆与 2020 铝之间的连接则采用 3D 打印件，刚度高，连接可靠且可拆卸，如图 35 所示。

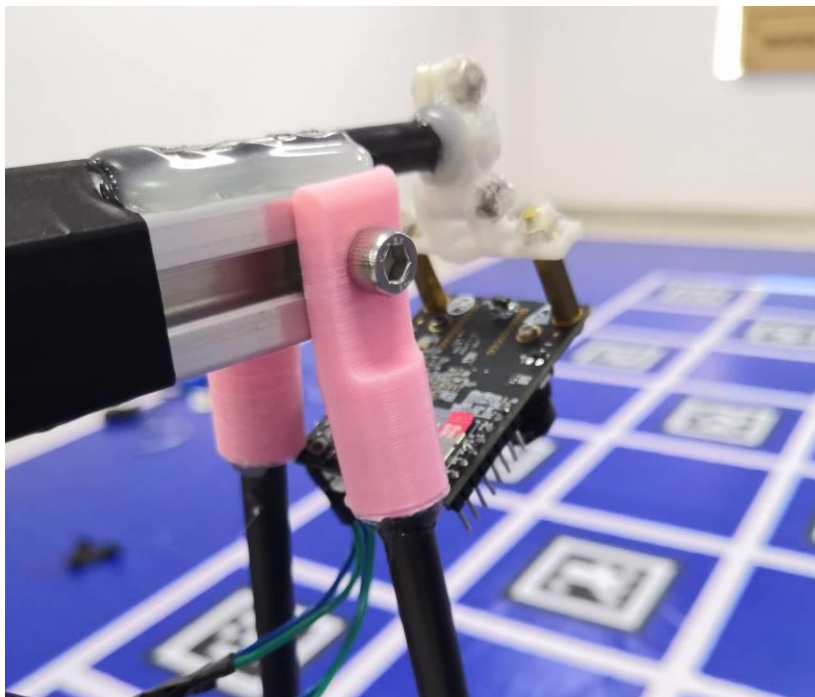


图 35 连接件图



## 9.2 主板等电路板固定

主板等电路板与车体连接均采用多过渡连接。首先按照电路板安装孔的尺寸及位置，使用雕刻机制作出用于安装电路板的环氧树脂板，在环氧树脂板上预留与车体相连的孔位，之后将环氧树脂板与车体固定，再将电路板安装于环氧树脂板上，整体安装如图 36 所示，使用这种安装方式使得线路布置和连接都有较好的可靠性。

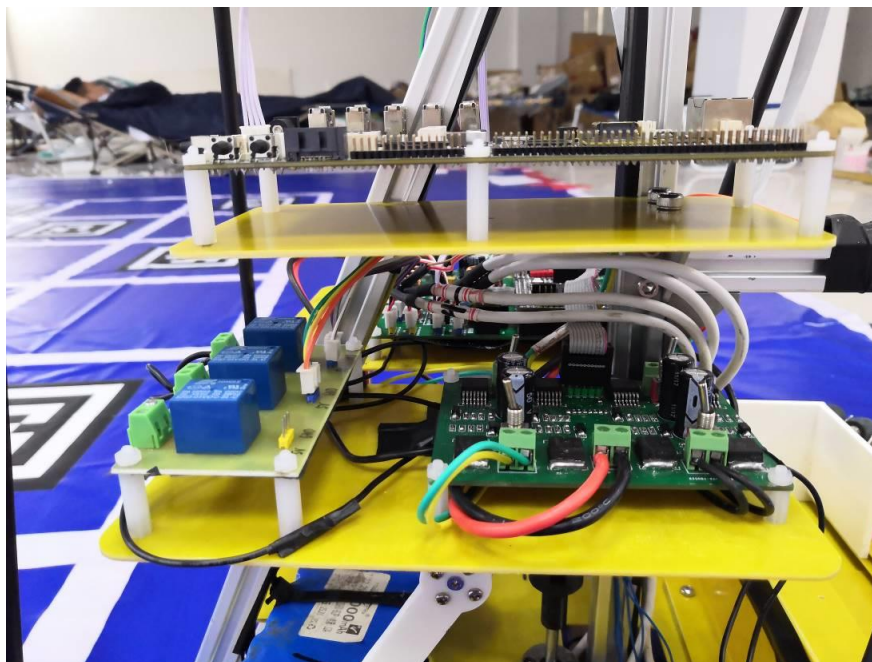


图 36 电路板安装图

## 9.3 摄像头（底部）安装

摄像头与碳素杆连接后，再将碳素杆与车体连接，具体见图 37：

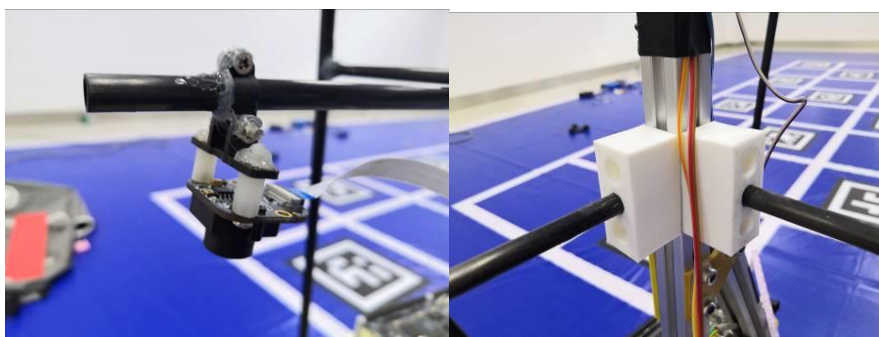


图 37 四摄安装

其中摄像头与碳素杆之间的连接采用标准件连接。而碳素杆与车体的连接则采用 3D 打印件。摄像头与碳素杆的连接处螺钉拧紧后，使用热熔胶粘连一圈，

增加可靠性防止掉落。

而碳素杆与车体的连接亦属于可拆卸连接，在运输时两个总钻风摄像头可与车体分离运输。而安装时又可以保证恢复原来的位置。



## 参考文献

- [1] 刘金琨. 先进 PID 控制 MATLAB 仿真[M]. 北京：电子工业出版社. 2011.
- [2] 卓晴，黄开胜.学做智能车[M].北京:北京航空航天大学出版社.2007
- [3] 杨光. 机械设计原理[M]. 北京:高等教育出版社.2002.
- [4] 周立青，黄根春，电子系统综合设计，北京：电子工业出版社，2017.2
- [5] 徐明远，刘增力.MATLAB 仿真在信号处理中的应用[M]. 西安:电子科技大学出版社.2007.
- [6] 于海生.计算机控制技术[M] 北京：机械工业出版社 2007.
- [7] 谢克明，王柏林.自动控制原理[M] 北京：电子工业出版社，2004

## 附录

### 1 主控单元部分程序源码

```
#include "headfile.h"
extern int WangToNum;
uint8 test_flag=0;
uint8 speed_mode = 0;
xy_t start_car_set={0,0};
xy_t start_car[28] = {  {0,0},{0,1},{0,2},{0,3},{0,4},{0,5},{0,6},{0,7},
                        {7,0},{7,1},{7,2},{7,3},{7,4},{7,5},{7,6},{7,7},
                        {1,0},{2,0},{3,0},{4,0},{5,0},{6,0},
                        {1,7},{2,7},{3,7},{4,7},{5,7},{6,7},
};
extern int16 Y_Speed,X_Speed;
int main(void)
{
    get_clk();           //配置时钟
    camera_init();       //摄像头初始化
    DisableInterrupts;
    freecars_init();     //上位机初始化
    Communication_init();//通信初始化
    Encoder_Init();      //编码器初始化
    Moto_Init();         //电机初始化
    InitServo();         //舵机初始化
    InitRelay();         //继电器初始化
    GPIO_init();
    initOLED();         //oled 初始化

    //参数初始化
    InitSpeedWithDistance();
    PID_Parameter_Init();           //速度环、位置环参数更新
    // Car_EnterTo();
```

```

S_s[0].x=4;
S_s[0].y=0;
S_s[1].x=4;
S_s[1].y=1;
S_s[2].x=4;
S_s[2].y=0;
optimal_n=7;
now_position.x=5;
now_position.y=1;
TransferToNum8t10(S_s[0],&WangToNum);
CarEnterOk = 1;
car_move_state=Car_chessing;
EnableInterrupts;
sysTick_delay_ms(2500);
ISR_init();           //时间中断初始化

for(;;)
{
//      car_move_state=Car_chessing;
//      car_move_state=Car_placing;
    En_Update_PID=Yes;

    if(gpio_get(B8))
    {
        for(;;)
        {
            En_Update_PID=No;
            if(gpio_get(B9)) break;
        }
    }

//      Moto_Set_Pwm(LH,-3000);
//      Moto_Set_Pwm(RH,-3000);
//      Moto_Set_Pwm(LB,-3000);
//      Moto_Set_Pwm(RB,-3000);
    push(0,WangToNum);

```

```

push(1,HeadCamera_Get);
push(2,X_Speed);
push(3,Y_Speed);
push(4,right_ninth_ed);
//    push(5,SysPID[position].sample);
tofreecars();

if(mt9v032_finish_flag)
{

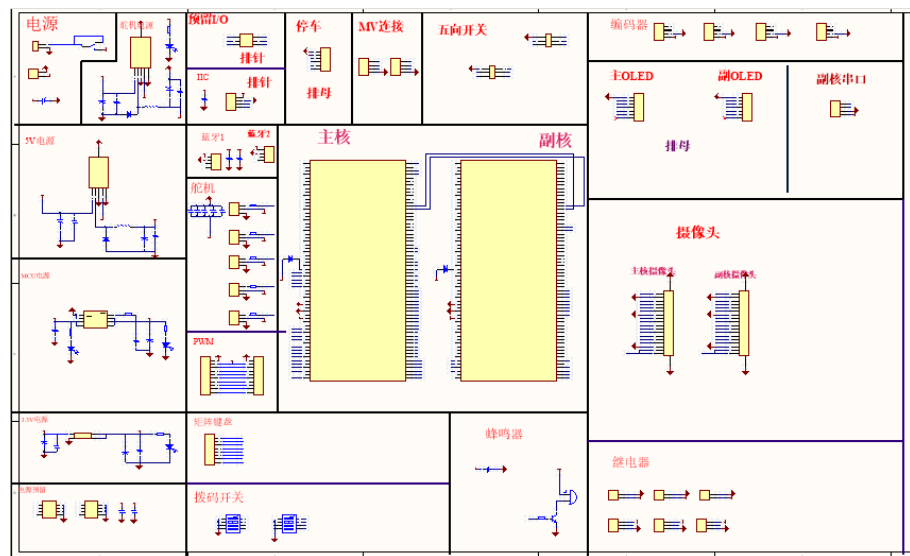
    mt9v032_finish_flag = 0;
    edge_extract();
    freecars_img(*temp,ROW,COL);

}

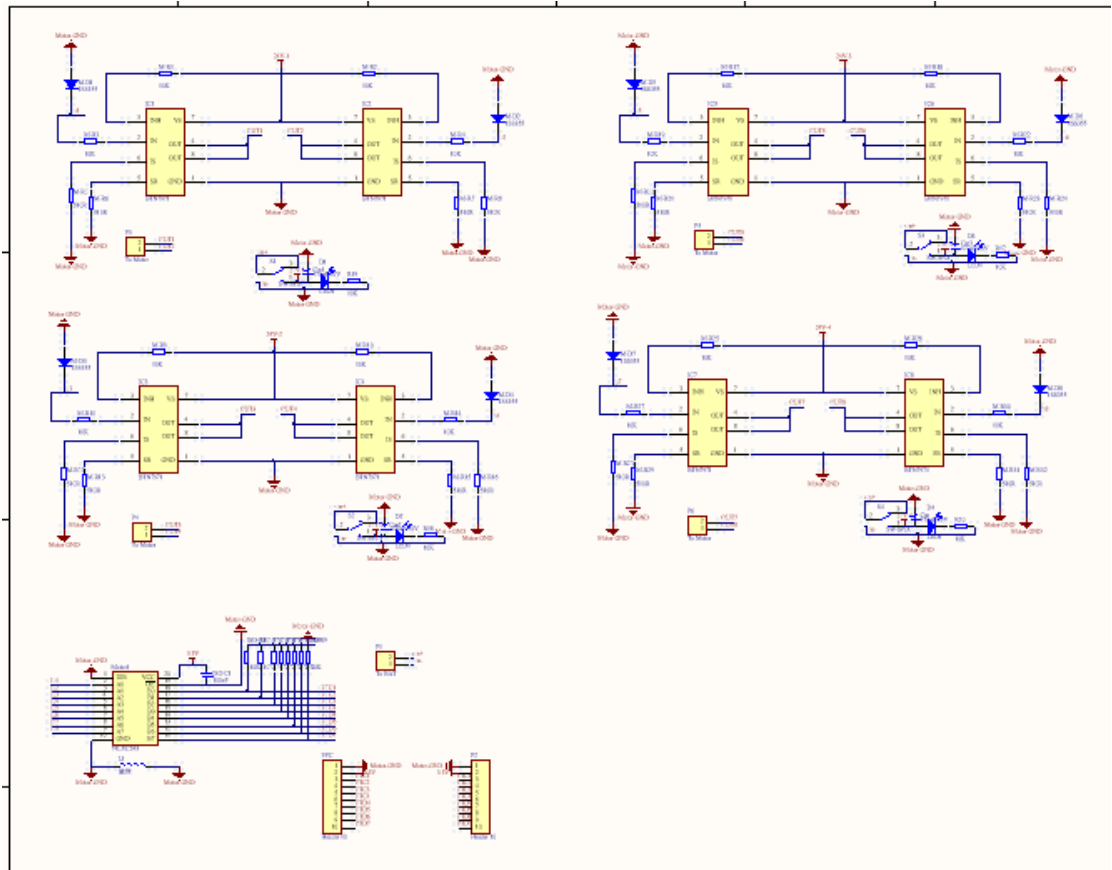
}

}
    
```

## 2 电路原理图



主控电路原理图



驱动电路原理图