

# 目录

## 第一章 总体方案设计

### 1.1 战车结构的总体设计

### 1.2 整体车模

### 1.3 主要模块介绍

### 1.4 算法总体思路

## 第二章 车身机械结构调整与优化

### 2.1 车辆尺寸以及重量

### 2.2 电池选择与安装

### 2.3 车头结构

### 2.4 车身结构

### 2.5 上台问题解决及优化

### 2.6 传感器选择与位置摆放

## 第三章 车体控制软件及算法

### 3.1 自身位置确定

### 3.2 台上设计思路

### 3.3 AprilTag 码识别

### 3.4 台下设计思路

## 第四章 开发工具、调试过程说明

### 4.1 软件编译环境

### 4.2 调试过程

# 第一章 总体方案设计

## 1.1 战车结构的总体设计

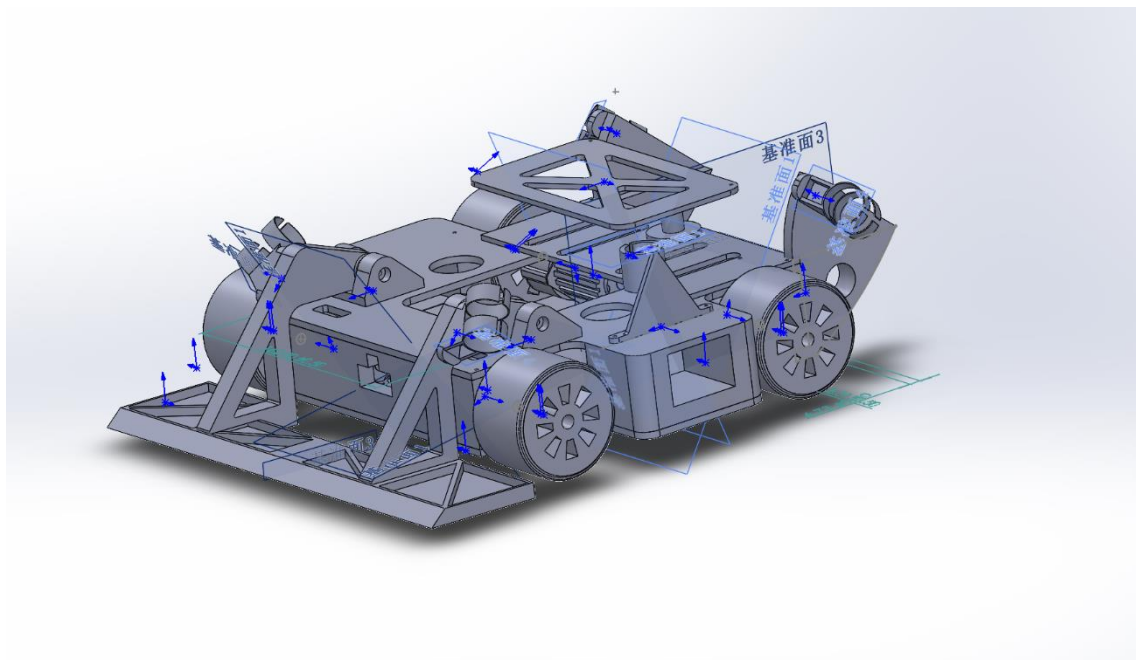
在对战车结构的设计时，我们先给主要的电子设备如电机、驱动、主控模块和传感器等配件在 SolidWorks 中绘制了等比例的 3D 模型。再通过配件的尺寸设计车身的尺寸并合理规划设备安放位置。为了提升上台的成功率我们选择了较大的前后轴距，使轮胎处于车身主体的最前端和最后段，避免上台时因为车身轮子无法着地被卡住。在设计传感器安装位置时我们用槽代替孔，让传感器的位置与角度在不改变车身设计的前提下有着一定的调试范围。为了提升上台以及对抗时战车的推力，我们选用了规则允许范围内直径最大的硅胶胎，从根本上解决了战车打滑的问题，进一步提升了上台的成功率。

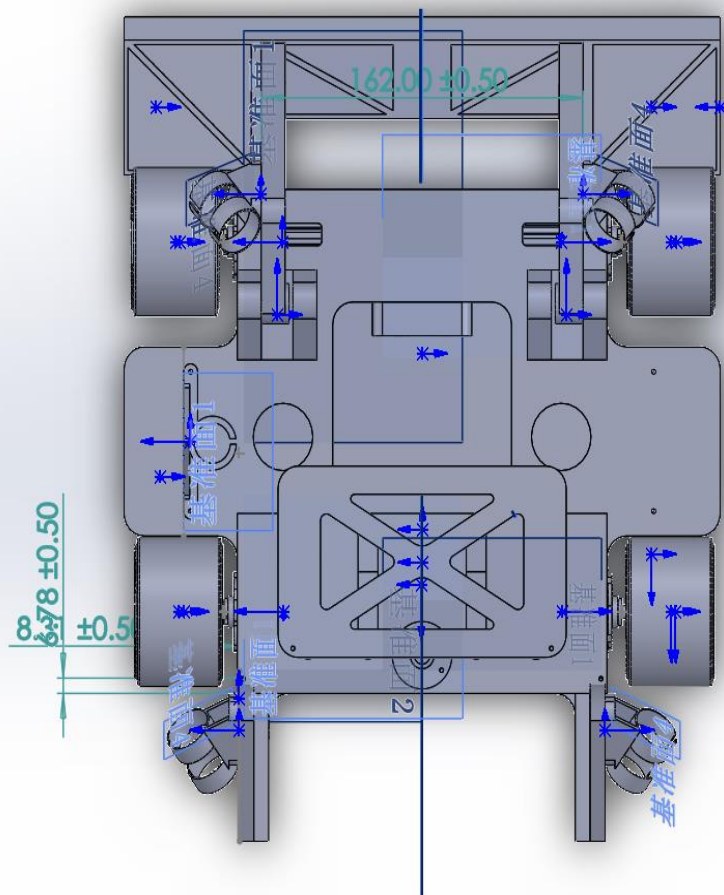
重量:规则范围内，在能保障动力足够使战车上台的前提下，车身重量越重越好，更大的重量可以让车辆获得更大的摩擦力，更容易推动目标物品。

尺寸:尺寸大小应当在规则允许范围内尽量做大做扁,最大限度降低机器人的整体重心,才可以使机器人在擂台上更好地行走，在对抗中拥有更好的稳定性

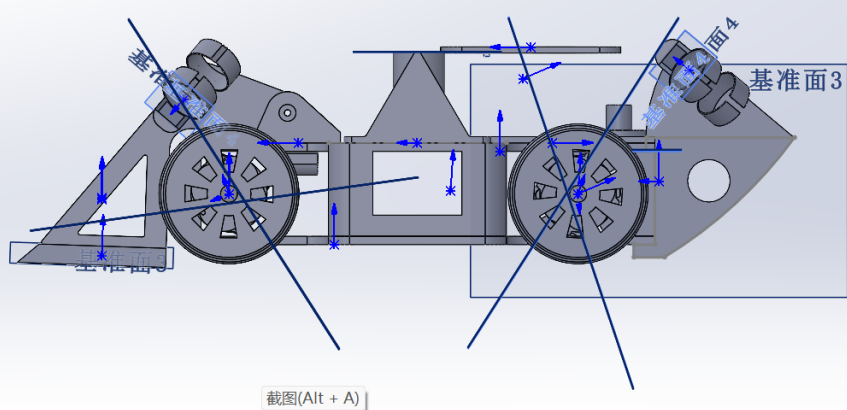
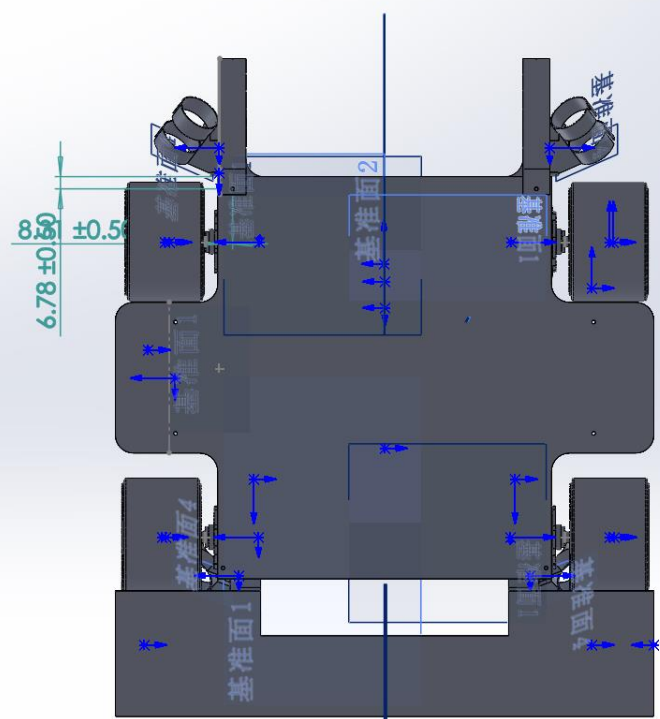
## 1.2 整体车模

整车模型自己设计，以下是通过 solidwork 完成的 3D 模型：

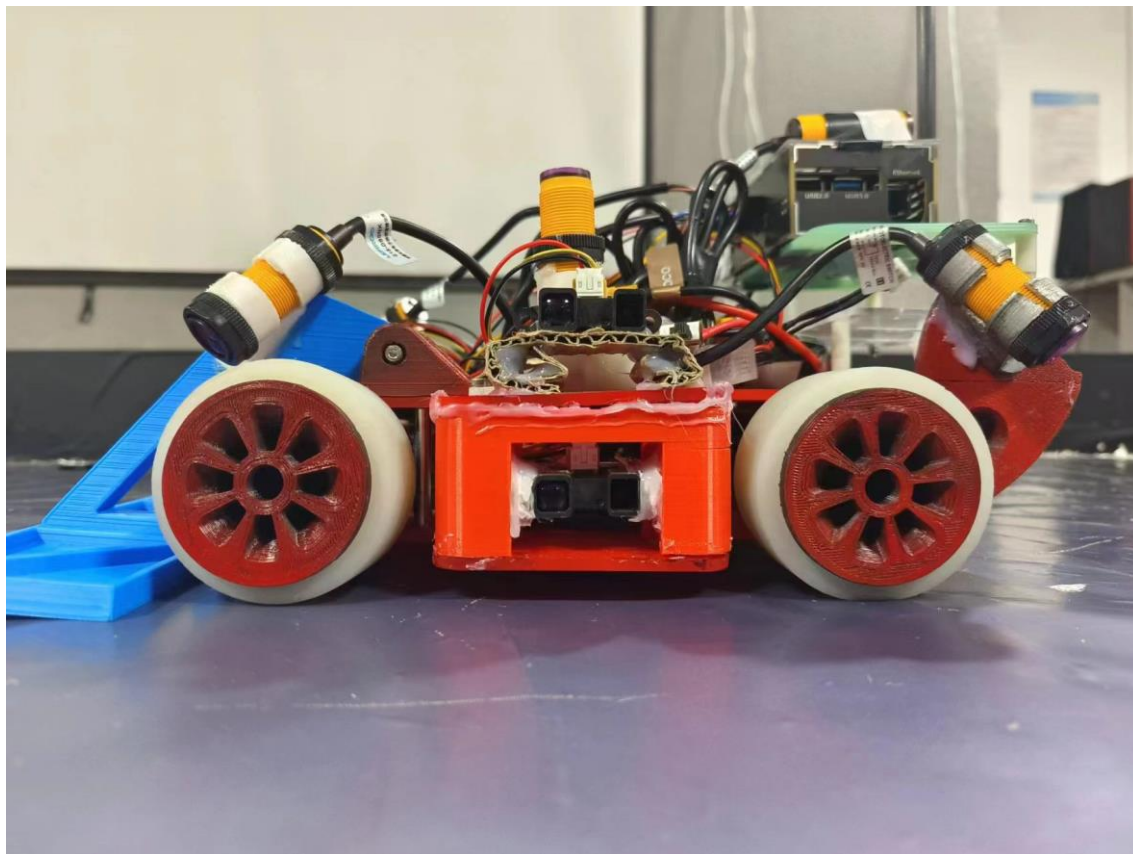


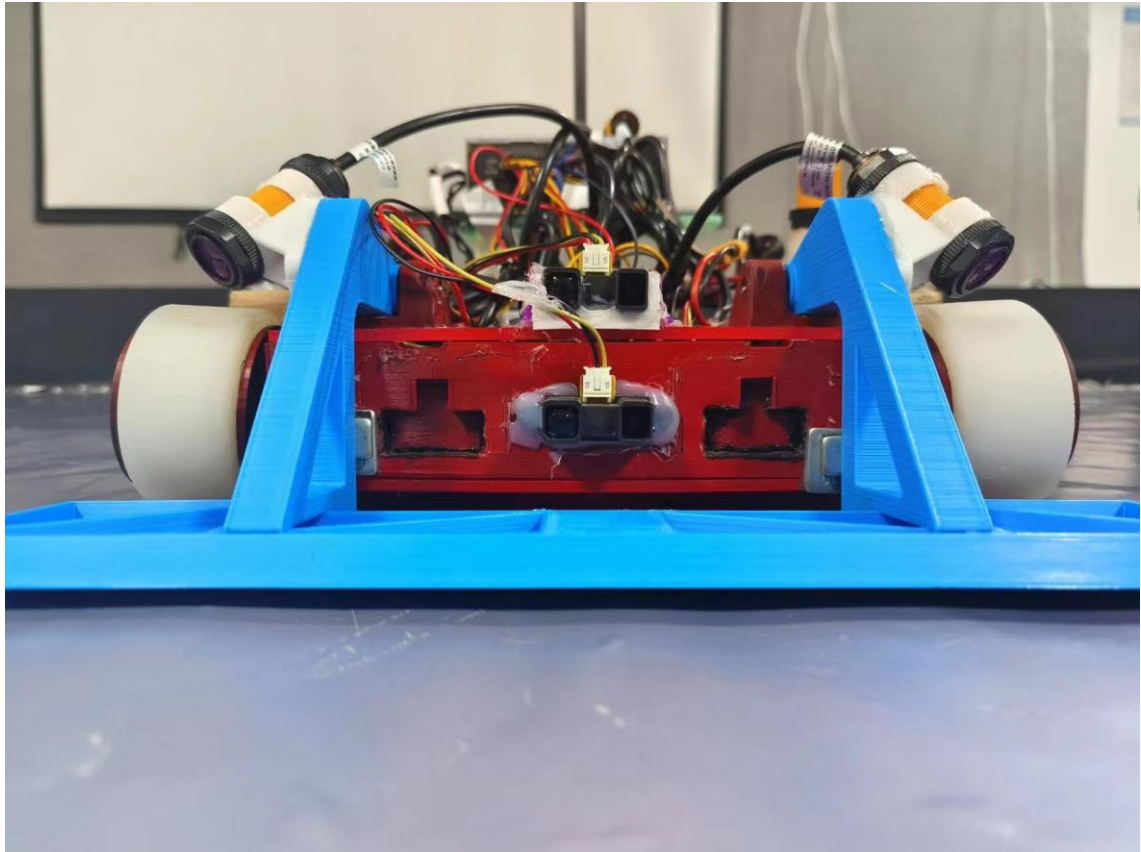


\* →



以下为实物图：





### 1.3 主要模块介绍

车体模块分为：台下定位模块，  
防掉台模块，  
攻击模块，  
供电模块，  
视觉处理模块。

### 1.4 算法总体思路

首先挥手启动，让小车全速上台。

之后通过上面四个传感器的值检测自己是否在台上，若不在台上重新上台。反之，通过下方四个传感器以及摄像头进行检测周围是否

有物体，没有则开启漫游。有物体检测物体的类型，并根据类型进行不同的反应。

小车最上面一层还有四个传感器，来检测平台边缘，防止掉台

在第三章中进行了最为详情的介绍，在 3.5 算法思路改进章节中阐述了我们对轮式战车的思考与改进。

## 第二章 车身机械结构调整与优化

### 2.1 车辆尺寸以及重量

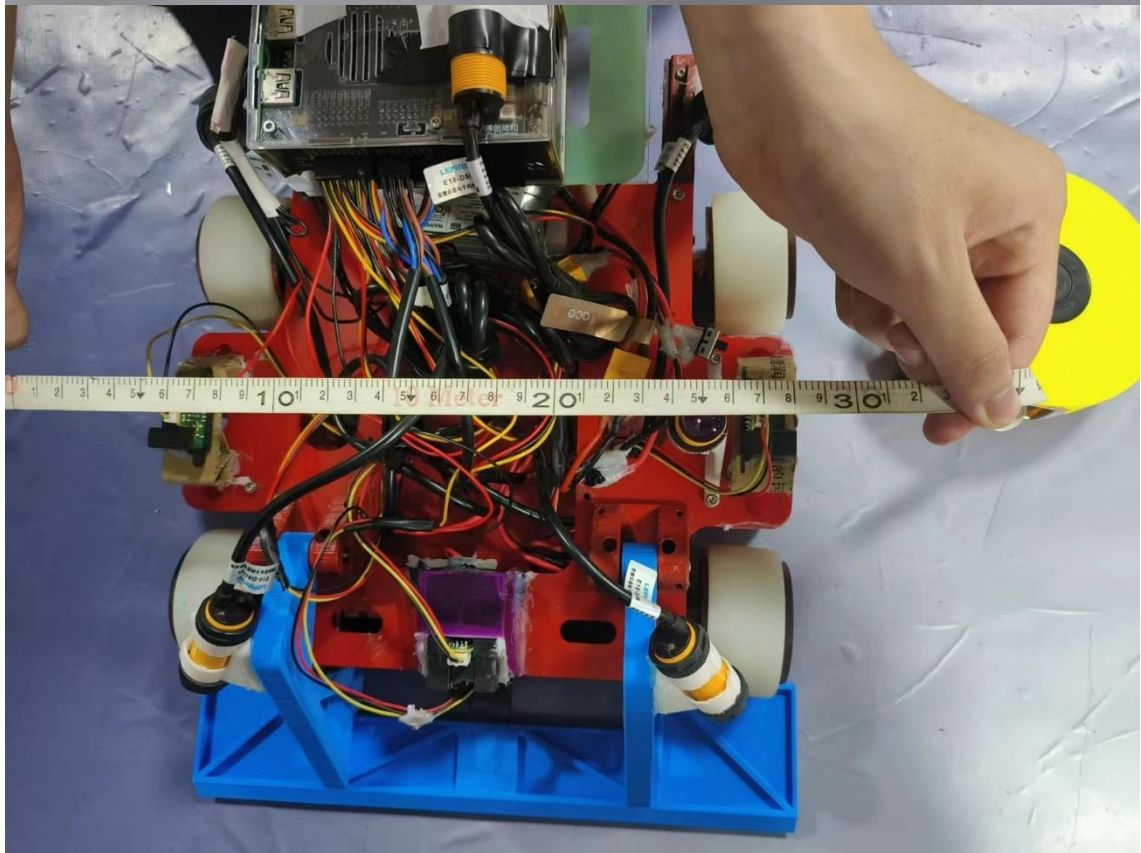
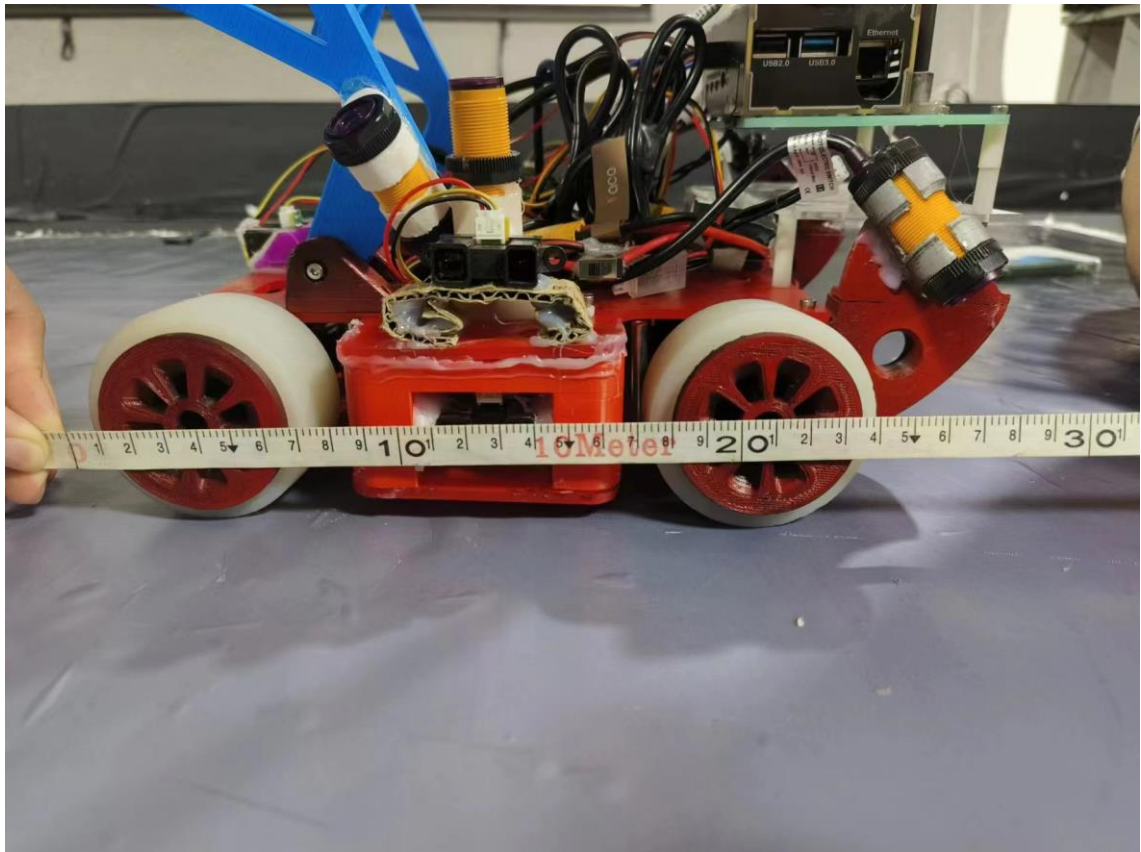
长：30cm

宽：29cm

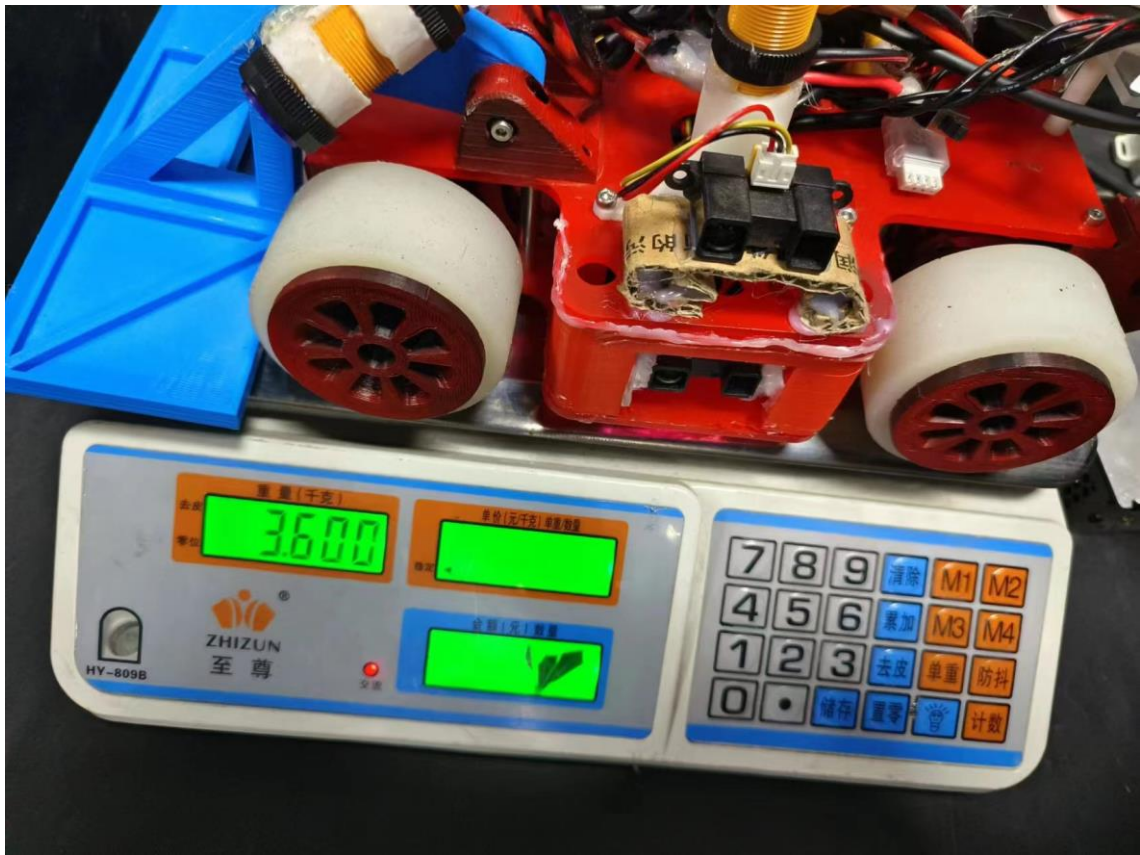
重量：3.65kg

如图：





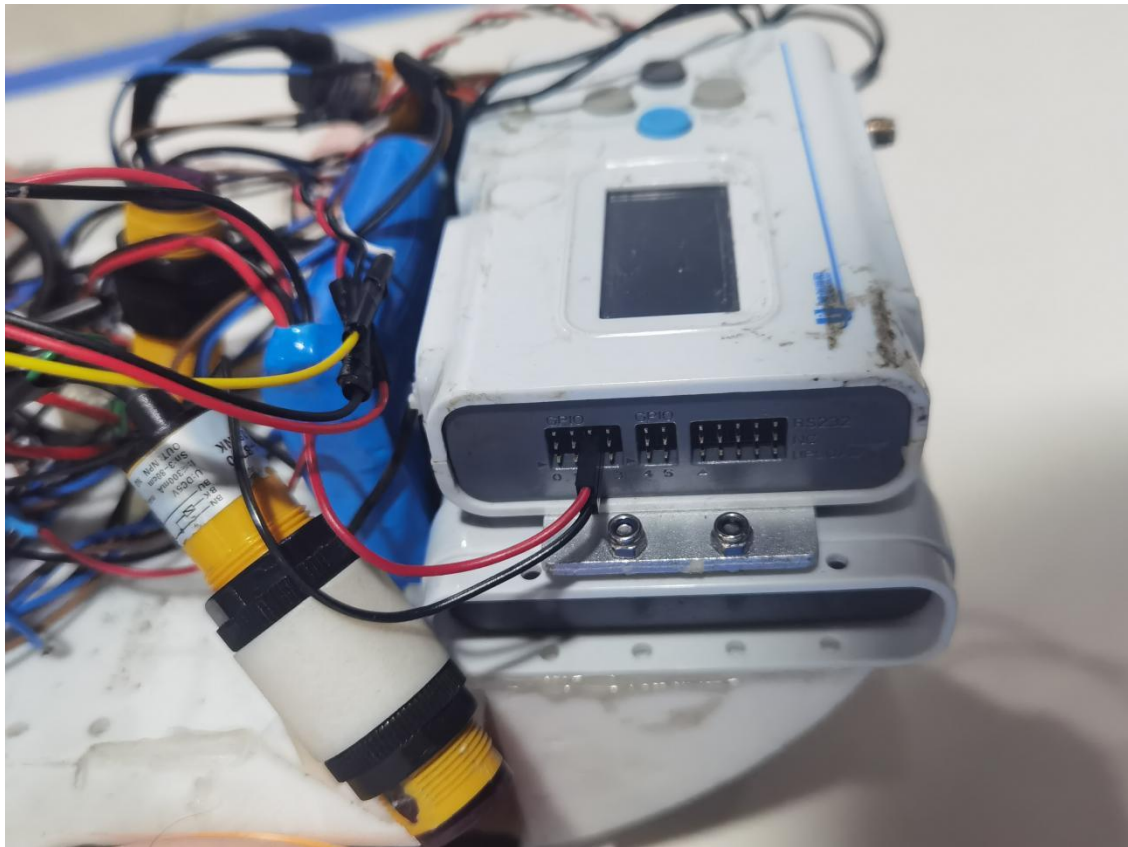




## 2.2 电池选择与安装

车身使用了两块电池，一块为官方提供电池，主要为传感器和主控供电，另一块为 12v 的锂电池，主要为驱动与电机供电。

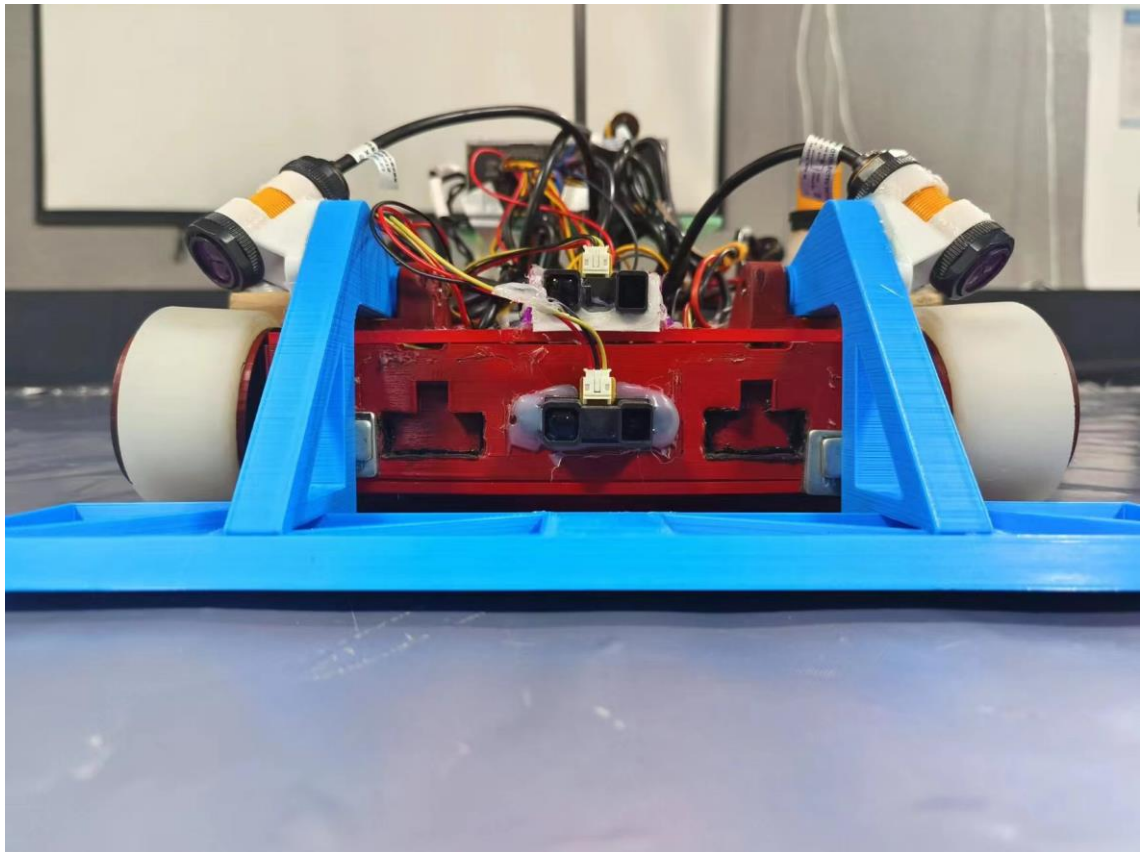
电池安装位置位于车体尾部位置，将车体整车重心位置靠后，提高上台效率和稳定性，优化上台功能，同时方便充电与开关驱动电源。



### 2.3 车头结构

本车对车头处进行了较大部分的更改，灵感来自于赛车，其不仅有良好的抗撞击能力，同时拥有强大的功能性，相比于传统车头，本车头设计有以下特点：

1. 车头自然下垂时低于底盘，保证与在场地行进时铲子与地面契合，防止被对手铲子铲起。
2. 在车头接近轮胎部分采用 U 型缺口，保证车轮运行的稳定性。
3. 考虑到赛制中有抢方块的规则，由此设计了如赛车般流线型的车头让机器人加速效率更高，更灵活，更容易掌握先机。
4. 车身与铲子连接部分结构轻便，不会遮挡摄像头视野、影响摄像头工作。



## 2.4 车身结构

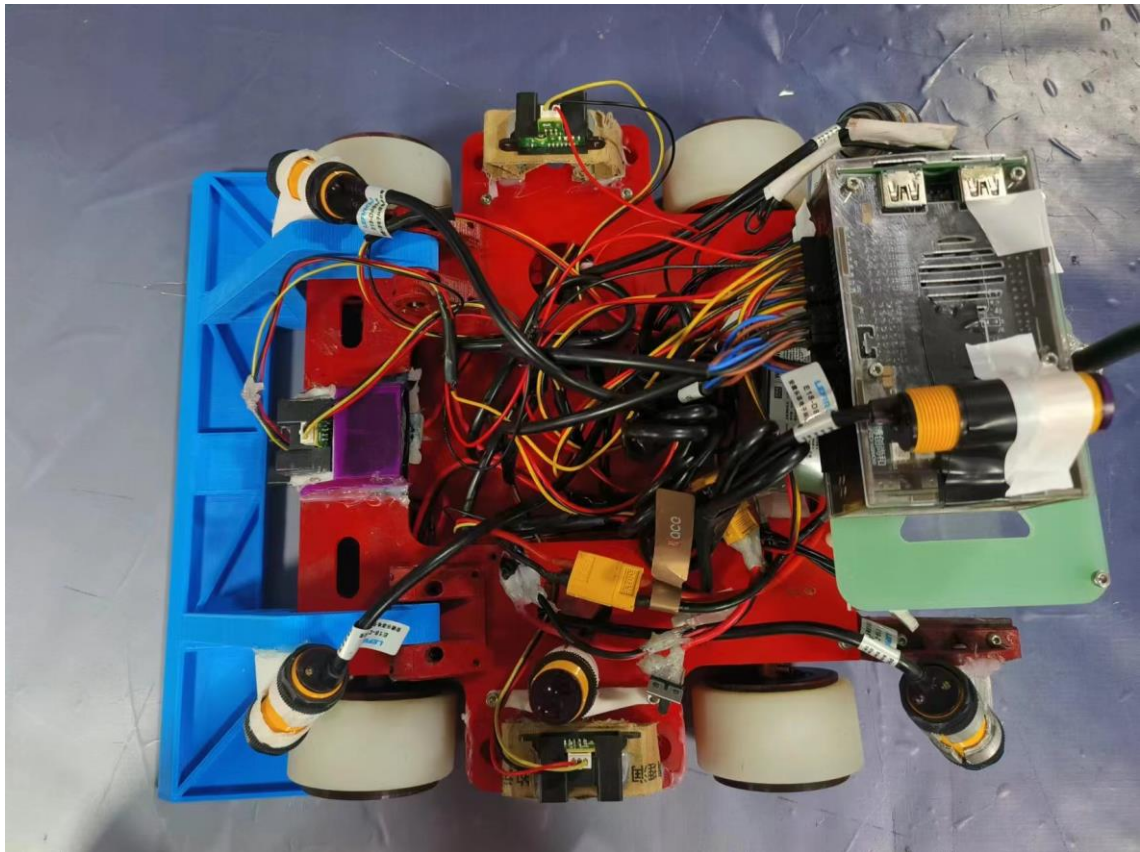
车体材料整体采用 PVC 3D 打印材质，拥有较大的稳定性与牢固性。上下板支撑结构为四个方向，使其支撑面积更大，不易断裂坍塌。

在设计过程中采用标准件方法设计机器人，使整车由多个标准组件拼装而成，有利于比赛中车辆损坏的快速替换，而且便于制造装配、采购和更好的扩展性。

本车采用四驱轮移动机构，每个驱动轮独立驱动，同时轮胎采用双驼峰橡胶轮胎，可以产生较大的驱动力。

对于主控、驱动、电源的位置放置，采取重心偏后原则，将他们放置于辅助上台模块之上，使得重心后移，便于上台。本车上台方式采取后方倒退上台(详见 2.5)对重心位置有较高的要求，因此将较多部件放置于后侧，便于上台。





## 2.5 上台问题解决及优化

因为比赛规则中没有坡道的设置，所以机器人要想登上 60mm 高的擂台，就对机器人的结构提出了更高的要求，要为登台设计专门的机构，实现机器人的自主登台。根据对比赛场地的分析，登上 60mm 高的擂台方式经过多次失败与尝试，有如下几种：

### 1. 直接登台式

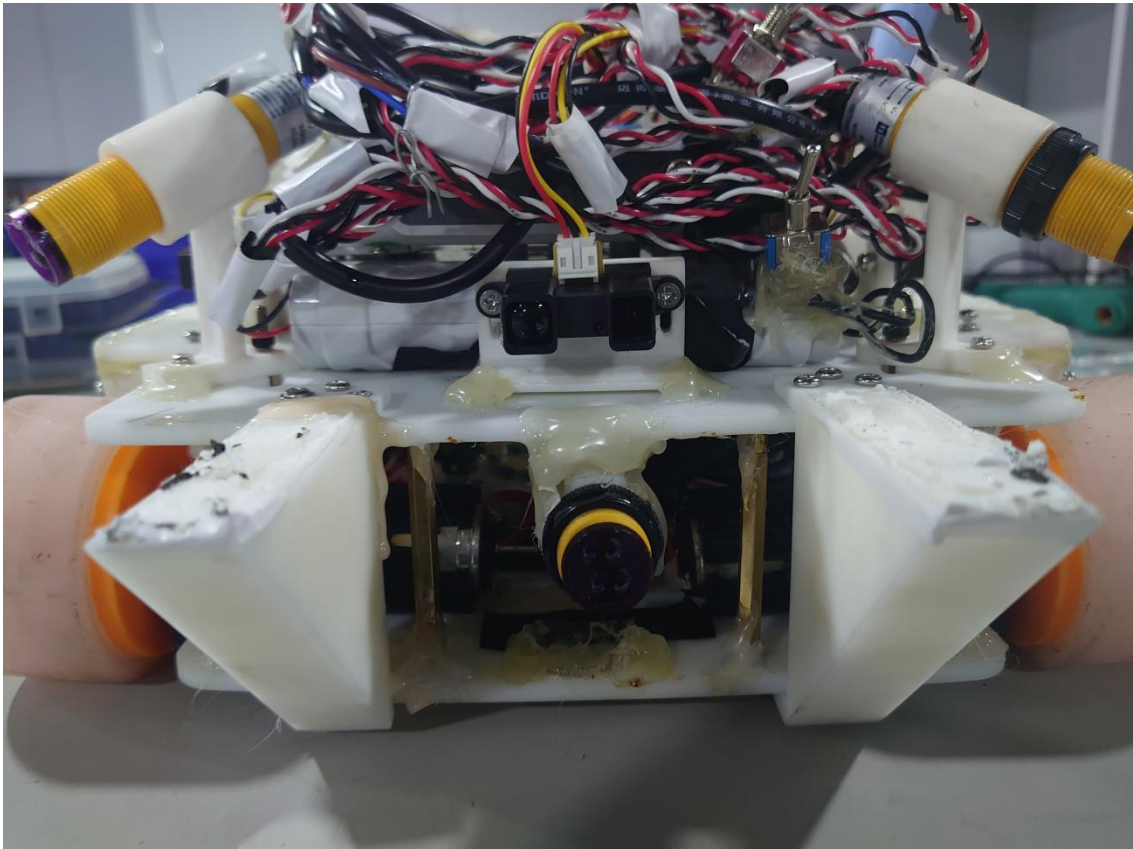
可以使用半径大于 60mm 的车轮或者使用履带结构，这样机器人可以直接走上擂台。这种方式的优点是上台很迅速，缺点是提高了机器人的重心，在比赛当中容易被对方推下台，进攻上灵活性较低。

### 2. 攀爬式

可以利用攀爬机构，使机器人爬上擂台，例如使用舵机，在机器人的前后各安装 2 个舵机，利用舵机转动一定的角度，驱动前后的挡板，构成攀爬结构使擂台机器人爬上擂台。这种方式的优点是不改变机器人的驱动结构，进攻灵活、重心低防御力强。缺点是需要安装四个舵机，而机器人的大小是有限制

的，四个舵机会很占用机器人的空间，并且舵机的增加，又使得机器人出现故障的几率增加，上台速度较慢。

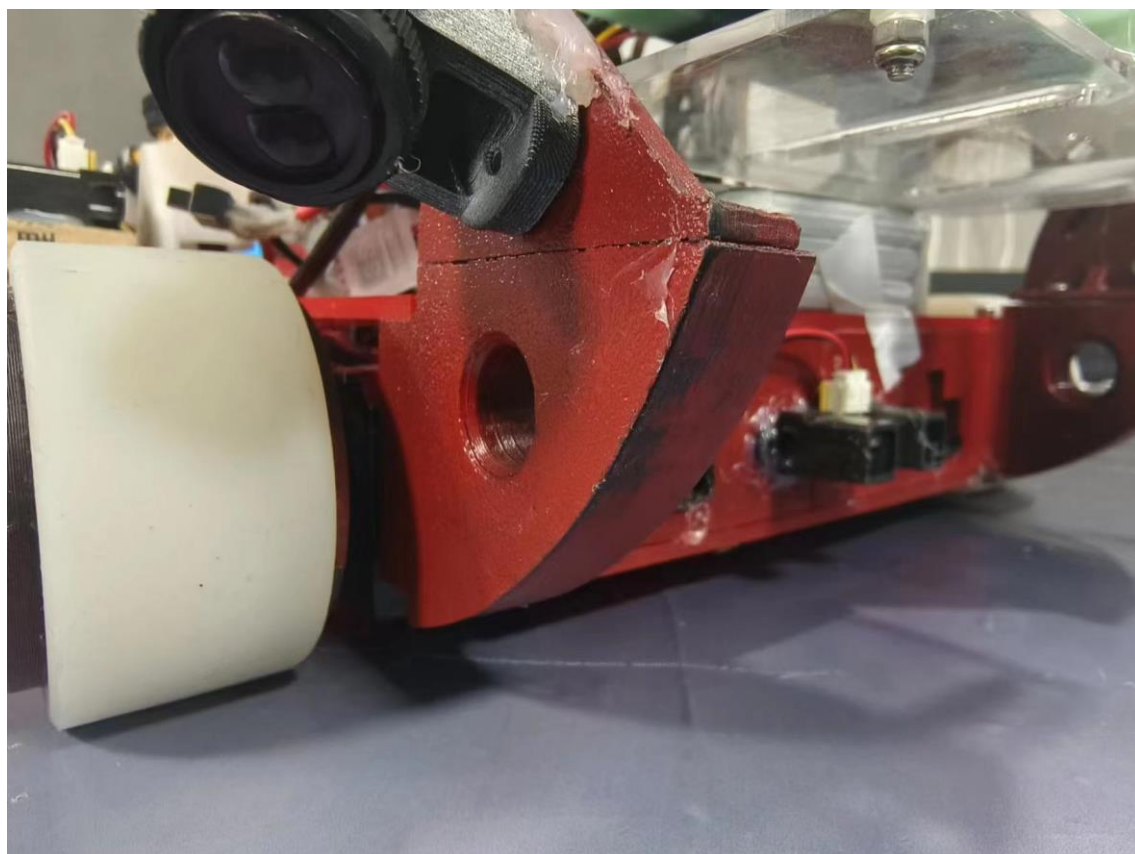
对比上述两种登台方式各自的优缺点和实际实验，我们首先采用出一种新的上台装置，上台装置为两个倒三角形斜劈，安装在机器人的尾部，如下图所示：



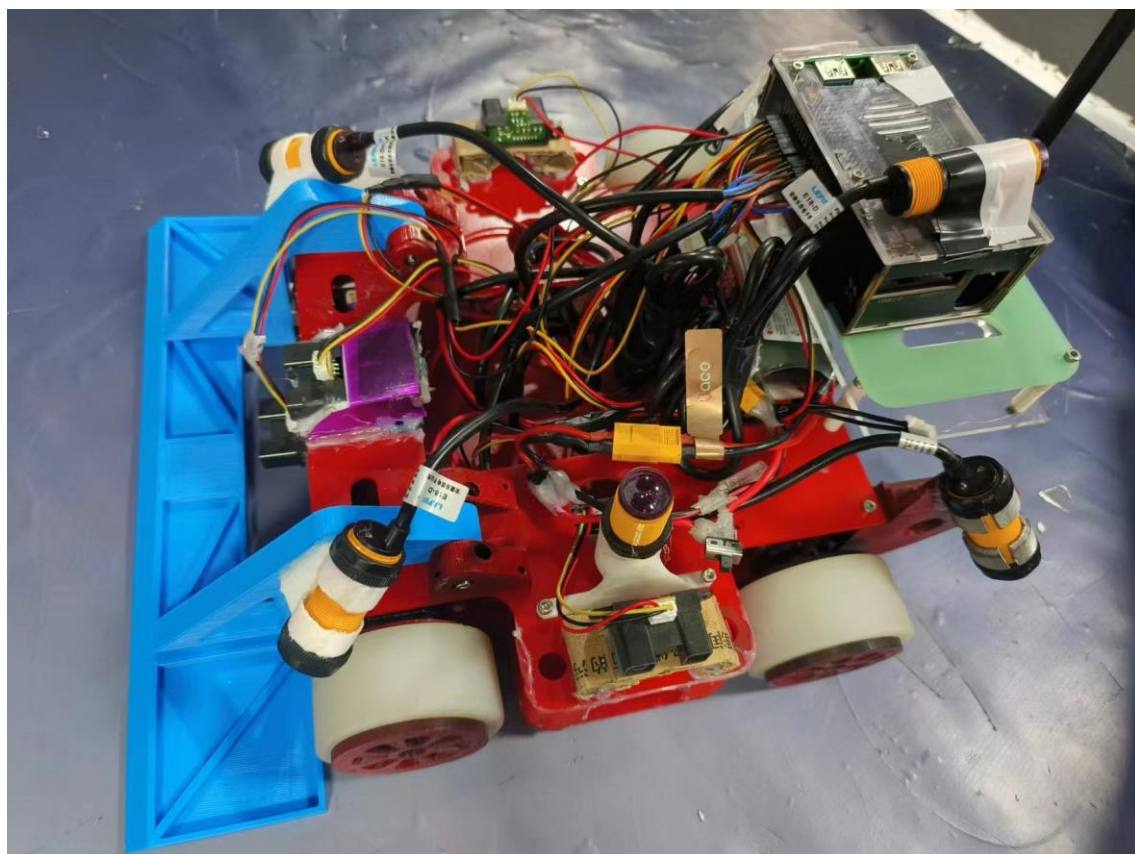
但是效果并不理想，登台时车底接触台面时导致四轮悬空而无法上台，我们选择改善了底板结构，让四轮不再悬空；但是在随后的登台过程发现机器人动力不足，我们就采用增加配重的方法使重心后移，虽然有几次上台成功，但是效率和稳定性并不理想。

我们经过小组讨论后决定再次从结构出发，通过减少配重，提高车速，将控制器置于车尾来后移重心，一次性完成登台。最后我们选择了如下图所示的圆弧形斜劈，在该种斜劈接触阶面时可以把车尾抬得更高，使车前轮充分接地，提供充足动力从而稳定有效的完成上台。





不但如此，我们也更换直径更大的车轮以提高上台成功率：





红外测距传感器返回距离模拟量用于多角度自主上台：

## 2.6 传感器选择与位置摆放

为实现防止掉台的功能，我们选择了四个红外对管放在左前、右前、左后和右后的位置，在快触碰到擂台边缘时可以及时反映，传回数字信号。

为稳定精准的在台下定位成功，在传感器分布策略上进行了高于擂台和低于擂台的区分，在车身的左侧与前方均在高于擂台处出配有红外测距传感器，同时在低于擂台处配备了红外对管数字量传感器。

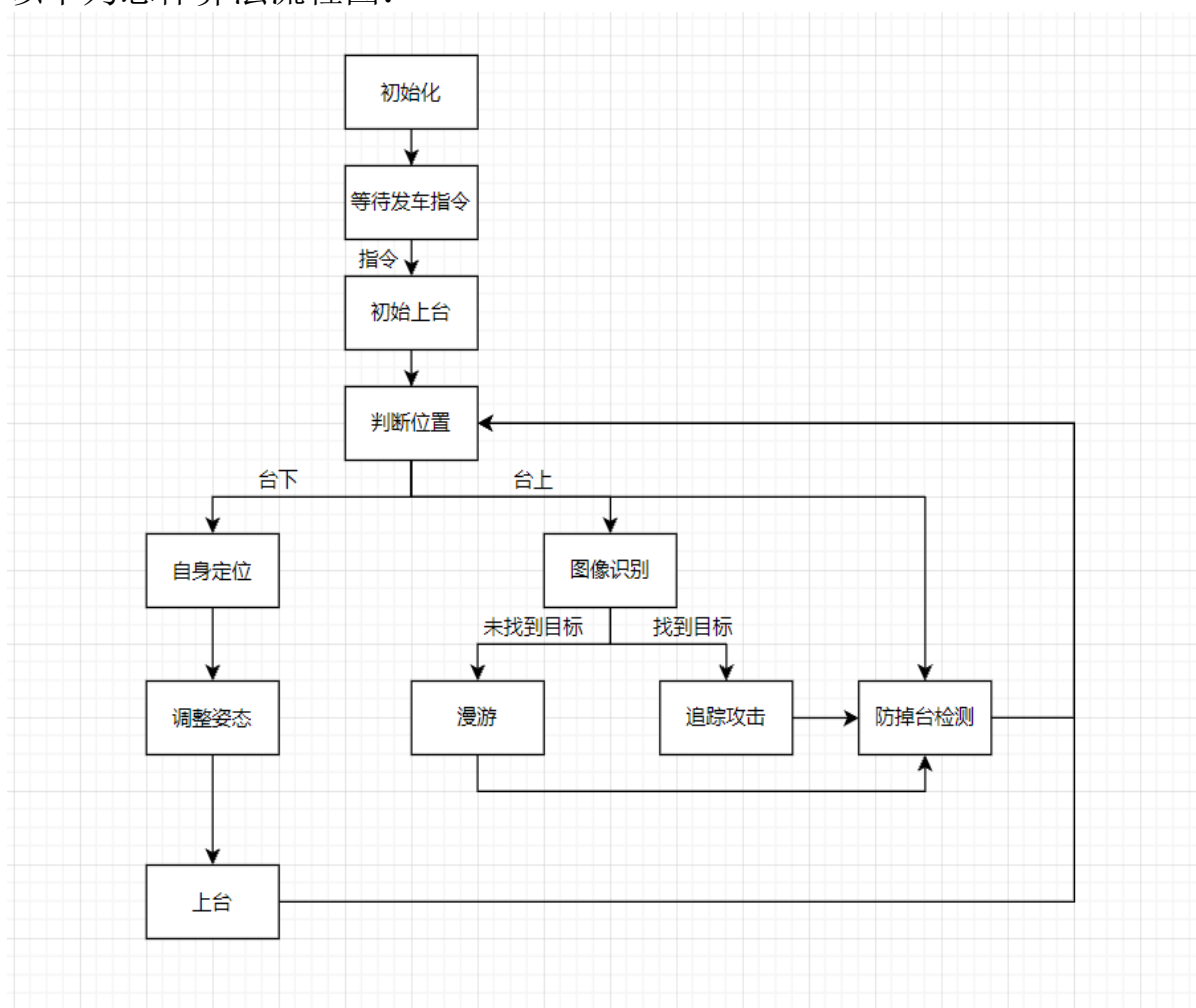
关于图像处理方面，选用了树莓派自带的库来进行 AprilTag 码的识别，并将其安装于车头位置，可以做到长前瞻与无阻挡。

另外，对于战车的软启动，我们加装了一个竖直的光电管（见 2.4 节图片），通过近距离的挥手实现软启动。

此种传感器配方法可以实现对于每一个情况做出相应的判断和相应的行动。多时段调节，复杂情况调节。

## 第三章 车体控制软件及算法

以下为总体算法流程图：



### 3.1 自身位置确定

通过上层的红外对管来监测是否到达边缘，通过下层的红外对管与红外模拟量传感器来判断是否已经掉台，并调整车体姿态。

具体有检测台上台下、检测边缘、在台下检测朝向、检测敌人的具体位置、调整车身对准敌人（目标方块）等。

### 3.2 台上设计思路

应用基础标准组件的函数封装，自己另设代码函数。整体框架源于车模状况的实时监测。将策略进行了模块化的细分：台上边缘检测、漫游寻敌、锁定攻击等众多模块。再通过定时器进行稳定的车体自身状态检测，从而达到反应灵敏和可以应对众多突发情况。

通过摄像头来进行 AprilTag 码识别，寻找需要推下的方块，为在刚寻找到目标时抢先敌方车辆行动，加快车辆响应占据战斗抢夺的主动权，我们选择在锁定之后就以满足速进攻。

对于敌方与中立方块，我方采取全速进攻的策略，对于己方方块，我方采取放弃防守的策略，因为最好的防御就是进攻。

由于传感器的局限性，我们限定了其搜索范围，出现一定几率的未找到目标的状态。此时，我们进行了设计了巡游状态进行弥补与防御，车辆以较低的速度开始按照我们设计的路线漫游，等待发现目标方块再进行攻击。

### 3.3 AprilTag 码识别

本次我们使用摄像头和树莓派里面相关的库来进行 AprilTag 码的识别，当识别码的 id 为 0（障碍）或者 1（能量）时，会进行坐标的读取、转换与传输，使控制器收到指令，控制车体向方块攻击。此次我们选择了忽视 id 为 2 的方块（敌方），因为经过测试，本车的得分效率很高，目的就是尽可能去自己得分，以进攻换防守。

### 3.4 台下设计思路

通过底部红外与上方红外传感器相结合，测定小车现在的位置和方向，不同的情况实行不同的代码，最终使小车背面朝向台子，全速冲向台子。

## 第四章 开发工具、调试过程说明

### 4.1 软件编译环境

程序的开发是在 pycharm 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。

PyCharm 是一种 Python IDE（Integrated Development Environment，集成开发环境），带有一整套可以帮助用户在使用

Python 语言开发时提高其效率的工具，比如调试、语法高亮、项目管理、代码跳转、智能提示、自动完成、单元测试、版本控制。

树莓派项目的一个核心思想是 Python 编程语言的使用。Python 允许树莓派的拥有者将项目扩展到令人难以置信的规模。

Python 是一个解释型的面向对象的、跨平台的编程语言。良好的可靠性、清晰的语法和易用性，使它成为最流行的编程语言之一。

Python 是一个优雅的、功能强大的语言。

通过 pycharm 连接树莓派并使用 python 语言进行设计，完全满足项目的开发。

## 4.2 调试过程

开发和调试过程中用到的主要开发和调试工具：pycharm。

使用写代码与实际情况相结合来调节参数，使小车功能达到预期目标。