

2025 睿抗机器人开发者大赛 (RAICOM2025)

四足多模态巡检 国赛技术方案报告

队伍名称：萝卜快跑

队长：高赫

队员：喻明、徐启轩、罗天豪、杨昊

指导老师：左少华、石富文

学校：华东师范大学

目录

一、 作品概述	4
1. 方案简介	4
2. 作品目标与阐述	5
3. 系统架构与组成	5
二、 比赛程序	7
1. 国赛赛规解读	7
(1) 物资与机械臂要求	7
(2) 视觉与运动要求	7
三、 问题分析与解决	9
1. 机械臂	9
(1) 分析与解决方案	9
(2) 可行性分析	9
2. 物资与载物台	10
(1) 分析与解决方案	10
(2) 可行性分析	11
3. 视觉算法	12
(1) 分析与解决方案	12
(2) 可行性分析	12
4. 运动控制	13
(1) 分析与解决方案	13
(2) 可行性分析	13
四、 技术方案	15

1. 机械臂	15
2. 物资与载物台	19
(1) 物资识别	19
(2) 载物台设计	19
3. 机器视觉处理	21
(1) 外接摄像头设计	21
(2) 获取原始图像 (origin_img)	21
(3) 图像二值化	22
(4) 获取轮廓与轮廓特殊值	22
(5) 识别 ArUco Tag	23
4. 机器人动作处理	24
(1) 状态机的使用	24
(2) 高层运动接口调用	26
5. 程序架构	27
6. 程序优化与硬件改进	28
(1) 程序优化	28
(2) 硬件优化	29
五、总结与感想	30

一、作品概述

1. 方案简介

四足机器人作为移动机器人，在物资运送的场景中具有其他机器人无法比拟的优势。我们团队使用杭州宇树科技有限公司生产的先进四足机器人：Unitree-legged-Go2-edu，在有幸参加本次高水准比赛。

为了呈现出更加完美的比赛结果，我们团队历经数月，构思并设计出了适合于本次比赛的技术方案。经过实测，依据本方案，四足机器人在规定的赛道中表现优良。因此本文能够作为合适的技术报告呈现给各位评委老师，关于本技术方案的组成结构如下：

1. **比赛程序部分：**将解读本次国赛的赛规，着重分析其中难点，相应的解决方案将放在后续问题分析与解决部分；
2. **问题分析与解决部分：**将回应分析的难点，具体介绍相应的解决方案，并且分析解决方案的可行性；
3. **方案实现部分：**将作为本技术方案的主体，详细说明如何实现解决方案。包括机械臂、机器视觉处理、任务规划、运动控制处理四个部分。其中附有简化的源码简要说明；
4. **抗干扰处理部分：**将说明如何保证机器狗在比赛中的稳定性；
5. **结果感想部分：**将阐释本团队从 2025 年 3 月组建初期到 8 月期间参加省赛并备赛国赛，至今 5 个多月的开发与研究感想。

2. 作品目标与阐述

针对于比赛要求，我们团队在机械臂、视觉与运动等方面在长期的调试下，总结出以完美完成比赛为基本目标，缩短完赛时长为进阶目标的核心思路，以下是关于本作品的大致阐述：

1. **机械臂方面**：核心思路是接受上位机指令后执行固定的动作。通过合理且强针对性的载物台的设计，使得机械臂能够顺利夹取和卸载物资。
2. **视觉方面**：核心思路是通过视觉算法得出赛道左右轮廓。根据轮廓计算的特征点与特征值是运动的核心判断依据。
3. **动作方面**：核心思路是借助轮廓的特征值，利用 **PID** 实现运动控制。同时，考虑到视野限制和环境光线干扰，有时会有无法获得赛道线的情况，在部分区域采用“动作固定，但结束标志不固定”的开发思路，从而合理减少算法复杂性并且增加程序的稳定性。

3. 系统架构与组成

四足机器人背部挂载了一个 Orin Nano，与四足机器人内部电脑有线连接，用于和四足机器人内部电脑进行 DDS 通信，同时在四足机器人脖颈处固定一个**摄像头**，接至 Orin Nano，视角朝正下方，用于读取赛道信息；采用激光切割制作的**物资盒**以及**第三方机械臂**，放置于机器人背部，接至 Orin Nano，用于完成物资夹取与卸载任务。

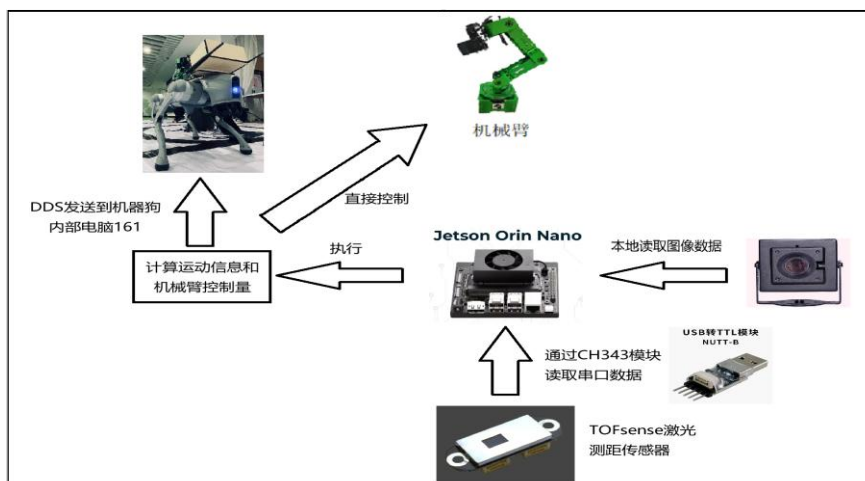


图 1.3.1 总体系统架构

本团队的总体程序系统架构如上图 1.3.1 所示，所有程序均被部署在 **Orin Nano** 上，进行所有的视觉处理、机械臂动作控制与四足机器人运动速度的计算；**Orin Nano** 计算得到的四足机器人运动速度将通过四足机器人 SDK 提供的高层运动接口发送给四足机器人内部电脑，从而控制四足机器人运动。

本团队成员相关知识储备量仍旧薄弱，本技术方案报告中若仍存有错误与不足，敬请各位评委老师批评指正。

二、比赛程序

1. 国赛赛规解读

(1) 物资与机械臂要求

- ◆ “可以在四足机器人背部搭载一个机械抓取装置用来抓取和 卸载物资。结构、尺寸不限，末端必须具备爪型结构。可外接电 池独立供电。不能配备算力模块，给四足机器人增加额外的算力。……选手需自行制作四种物资，每种一个。物资不限制颜色和材质，单个物资需通体使用同一种材质。物资上可贴有标签，标签尺寸 \leq 物资任一表面，标签内容不限，数字、ArUco Tag 均可。”

解读：不同物资形状不同，在载物台上的状态难以保持稳定，利用机械臂抓取的动作也不同。机械臂投放物资时，物资可能发生滚动甚至滚离投放区域，导致投放失败。这要求物资、载物台、机械臂以及机器人姿态四者高度协调，才能提高投放的成功率。而为防止物资滚动，物资本身可以通过材料设计、重心偏移减少物资本身的滚动，以及采用轻质软材料有利于机械臂的夹取。

(2) 视觉与运动要求

- ◆ “四足机器人沿黑色道路行走，行走至放置区时将物资卸载。”

解读：团队采用的四足机器人 Go2 虽然自带了头部的摄像头，但是能够获取到的地面图像信息有限，虽也可用于运动和特征识别，但是要求大幅降低机器人的运动速度。因此需要为机器人外加一个角度合适的摄像头。考虑到本赛题中四足机器人需要通过较为狭窄的避障区，为了减小四足机器人的转弯半径，本团队选择在四足机器人下方脖颈处固定一个

摄像头，这样既能方便获取地面赛道信息，又能避免增加过多的转弯半径。而赛规允许在倾倒区附近贴上 ArUco Tag，可以使用其判断进入放置区。

三、问题分析与解决

1. 机械臂

(1) 分析与解决方案

为了提高物资装载与投放的效率，本队选用一款由幻尔科技生产的智能机械臂，它由 6 个 HTD-35H 总线舵机串联构成，臂展接近 35cm，使得

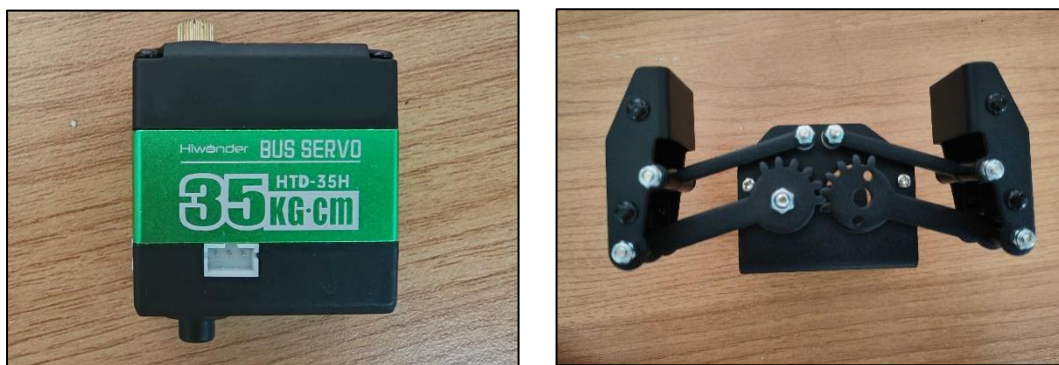


图 3.1.1 HTD-35H 总线舵机和金属宽口夹子实物图

机械臂整体具有较高的自由度和工作半径，为物资装载与投送任务的实现提供了便利。

由于比赛中的四种物资均形状规则且对称性良好，本队选用一只通过齿轮联动的金属宽口夹，如图 3.1.1 所示，既能保证夹取时物资受力均匀不打滑，又能在任务中充分发挥其金属的坚固耐用和齿轮的精准灵敏的优点。

(2) 可行性分析

在机械臂的动作调试过程中，本队使用上位机记录每个舵机的角度

参数，通过串口通信，能够通过 python 程序控制舵机。

2. 物资与载物台

(1) 分析与解决方案

为了让物资更好适配机械臂和物资盒，实现精准的物资运送，其材料的选择与识别尤为重要。需综合考虑物资的重量、硬度、是否易抓取、是否便于贴 ArUcoTag 等因素。

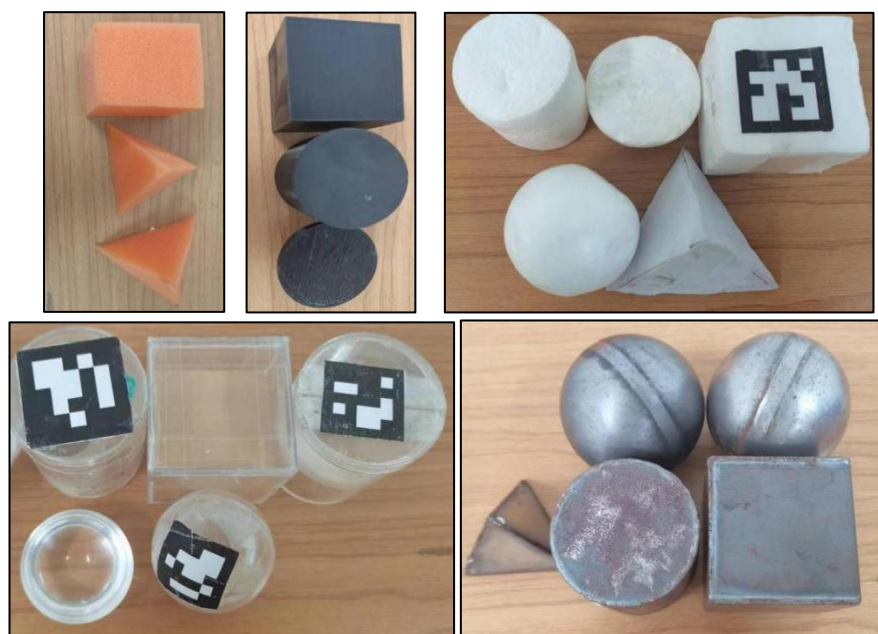


图 3.2.1 测试时选用的各种物资

在测试过程中，我们尝试使用过如图 3.2.1 所示的多种材料，包括但不限于泡沫、海绵、3D 打印 PLA 塑料、空心金属、偏置重心亚克力组合结构、EVA 材料。

而为了提高机械臂在夹取和装载物资时的效率和准确性，本队精心设计了一款新型的物资装载盒，如图 3.2.2 为物资盒的零件设计图与实物图。



图 3.2.2 物资盒的零件设计图与实物图

(2) 可行性分析

由于各类材料各有优缺，综合考虑实际测试，选取海绵这一材质，如图 3.2.3 所示，以此保证物资投送系统能够稳定抓取物资且不易脱落，投放成功率较高。



图 3.2.3 最终团队选择的物资

3. 视觉算法

(1) 分析与解决方案

团队采用 OpenCV 开源库作为主要算法基础，python 作为开发语言。

算法方面，采用 python OpenCV 开源库作为视觉算法基础，Canny 算法与八邻域算法作为主要计算方法。具体算法流程如下图 3.3.1 所示：

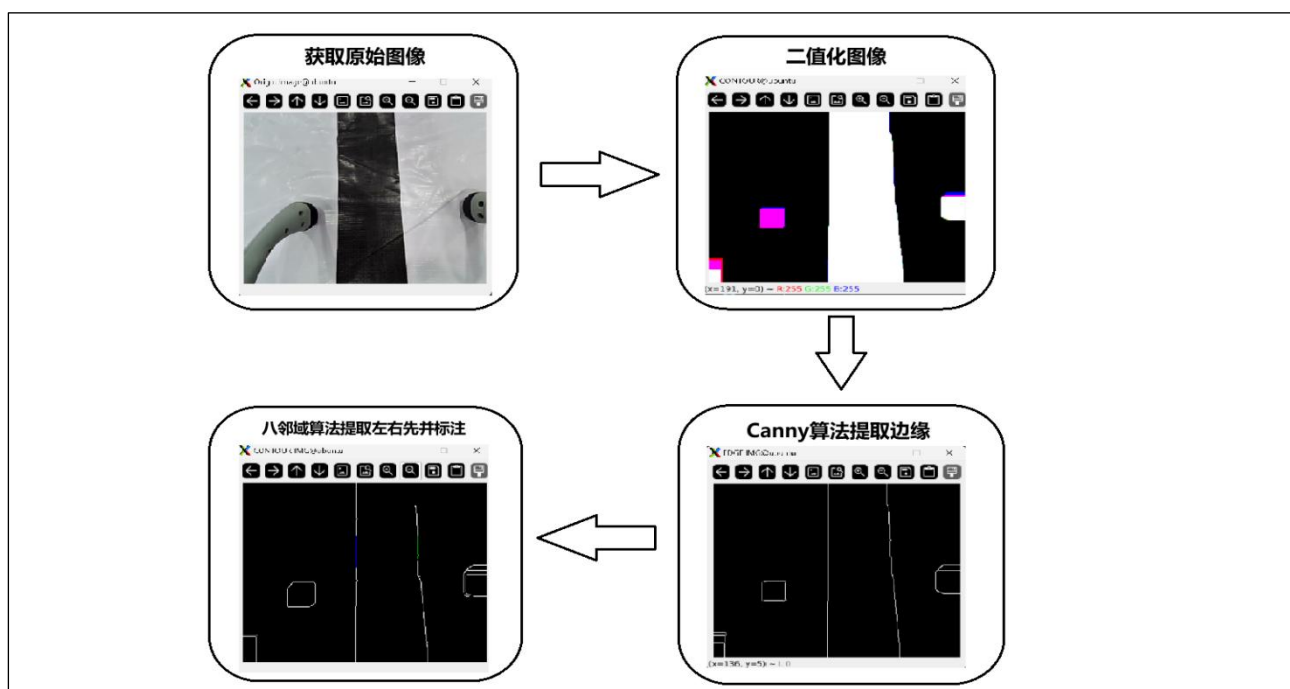


图 3.3.1 视觉算法图像处理流程

(2) 可行性分析

经过视觉算法处理后的左、右线条，是两个以点为元素的列表，通过对两个列表的处理，可得出用于运动计算的特征值，包括左右两条线 x 坐标的平均值和斜率，从而实现视觉与运动的结合。

4. 运动控制

(1) 分析与解决方案

动作函数的计算部分将于视觉部分一同在 Orin Nano 上执行，计算的结果将通过高层运动控制接口 `sport_client.Move(vx, vy, yaw_speed)` 发送给四足机器人内部电脑，内部电脑接收后将直接执行相应的动作指令。为保证机器人能在道路上行走，我们采用 PID 控制算法。考虑到赛事地图中曲线与直线混合较多，一套 PID 参数并不能保证运动稳定性，为此，团队采用**状态机转换机制**，在不同状态中将执行不同动作，并且使用不同 PID 参数。

状态机的实现如下所示：每条状态转换路径（trajectory）都有对应的执行的函数。当状态转换完成后，执行该函数。

通过对本次比赛场地地图的分析，划分出了六个主要状态：起点巡线和避障状态、进入放置区状态、进入待检区状态、忽略待检区状态、上台阶状态、终点巡线状态。

上述六个状态恰好对应了比赛过程中的所有情况。其中，环岛、避障、上台阶等状态又被分为了多个不同的微状态，用来适应运动控制算法的需要。

(2) 可行性分析

由于 GO2 背部拓展坞内部硬件版本问题，团队选择使用外接 Orin Nano 作为计算平台，其算力足够完成视觉数据的处理、运动速度计算和机械臂运动指令控制，且符合赛规中对算力的限制要求。

算法方面，PID 算法是最经典的运动算法，通过 PID 控制，可以使机

器人在直线与曲线上都有良好的运动表现。

同时，四足机器人在比赛期间会经过许多特殊的位置，通过状态机之间的定向转换，能够保证四足机器人在每个位置的运动都能符合预期。

四、技术方案

1. 机械臂

为了方便给机械臂供电和传递动作指令，缩小延迟和空间占用，将若干杜邦线、电源供电线和 USB 与 TTL 转换器焊接为一体，如图 4.1.1 所示，集成供电、信号传输为一体，与树莓派 Pi 通过 USB 直接相连，有助于提高机械臂的系统性，也十分利于调试和安装。

机械臂舵机 5 上放置第一个测距模块，用以在舵机 1 脉宽为 400~600 的范围内读取距离，判定距离最小值为目标物块的位置，锁定对应的舵机 1 脉宽。但是，由于环境噪声因素，有时测距模块会读取异常极大/极小值。对此，使用滑动窗口滤波策略，使用步长为 3 的窗口，求解平均值，即可很好地规避噪声干扰。在机械臂舵机 2 前方放置第二个测距模块，用以在锁定舵机 1 脉宽正确位置后测量深度信息。对于目标物块，横向偏移量为 0，垂直落差已知，深度值由第二个测距模块读取。因此，目标物块的空间位置信息被唯一确定，唯一需要的是机械臂的逆运动学求解。下面来求解幻尔机械臂的逆运动学方程：

根据幻尔机械臂的连轴姿态，可以将逆运动学的问题简化为下图：

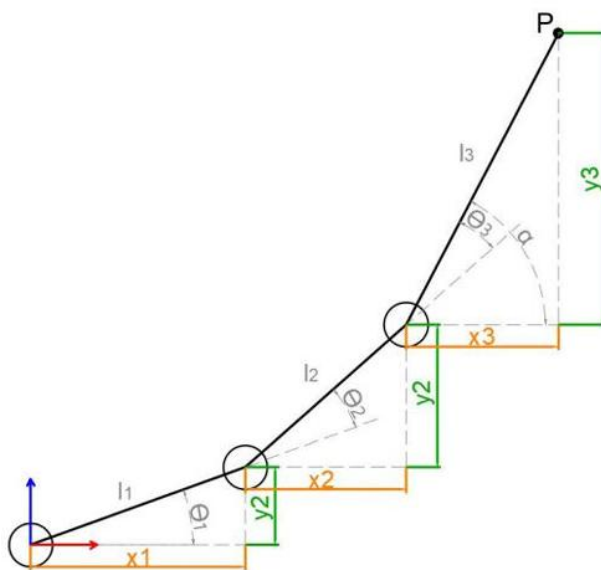


图 4.1.1 幻尔机械臂简化模型

可以发现，影响机械臂 z 轴深度的变量为 θ_1 ， θ_2 及 θ_3 ，因此，任务化为求解 θ_1 ， θ_2 和 θ_3 。可以列出方程：

$$\begin{cases} x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{cases}$$

其中 x 为深度， y 为物体高度， $l_1 \sim l_3$ 可以由 D-H 参数表求得，均为已知量。由于两个方程对应有 3 个未知量，方程有无数多组解，下面用一种容易以 Python 语言封装的方式求解这个方程：

将方程平方相加：

$$x^2 + y^2 = \sum_{i=1}^3 l_i^2 + 2 \sum_{1 \leq i < j \leq 3} l_i l_j \cos(\theta_j + \theta_{j+1} + \dots + \theta_i)$$

展开后化简得：

$$x^2 + y^2 = l_1^2 + l_2^2 + l_3^2 + 2l_1 l_2 \cos \theta_2 + 2l_1 l_3 \cos(\theta_2 + \theta_3) + 2l_2 l_3 \cos \theta_3$$

整理为关于 θ_2 的方程：

$$A \cos \theta_2 + B \sin \theta_2 = C$$

其中：

$$A = 2l_1(l_2 + l_3 \cos \theta_3)$$

$$B = -2l_1 l_3 \sin \theta_3$$

$$C = x^2 + y^2 - (l_1^2 + l_2^2 + l_3^2 + 2l_2 l_3 \cos \theta_3)$$

引入相位角 $\phi = \arctan\left(\frac{B}{A}\right)$ ，解得：

$$\theta_2 = \phi \pm \arccos\left(\frac{C}{\sqrt{A^2 + B^2}}\right) + 2k\pi$$

对每个 θ_2 解，重组方程为：

$$\begin{cases} x = A \cos \theta_1 - B \sin \theta_1 \\ y = A \sin \theta_1 + B \cos \theta_1 \end{cases}$$

其中：

$$A = l_1 + l_2 \cos \theta_2 + l_3 \cos(\theta_2 + \theta_3)$$

$$B = l_2 \sin \theta_2 + l_3 \sin(\theta_2 + \theta_3)$$

解得：

$$\theta_1 = \arctan\left(\frac{Ay - Bx}{Ax + By}\right)$$

解存在的必要条件为：

$$\left| \frac{C}{\sqrt{A^2 + B^2}} \right| \leq 1$$

当满足条件时，通常存在两组解。

X 横向偏移	Y 高度偏移	Z 深度偏移	x^2+z^2	舵机 1	舵机 2	舵机 3	舵机 4
0.1077	0.09	0.3702	0.14864733	455	230	370	500
0.0223	0.09	0.3732	0.13977553	500	230	400	480
0.056	0.09	0.3558	0.12972964	480	280	360	430
0.0204	0.09	0.344	0.11875216	510	325	240	510
-0.0203	0.09	0.344	0.11874809	540	280	350	480
0.0838	0.09	0.3302	0.11605448	460	300	280	500
0.0219	0.09	0.3305	0.10970986	500	300	300	475
-0.0005	0.09	0.3265	0.1066025	520	300	275	500
0.0332	0.09	0.3155	0.10064249	540	300	300	450
0.0471	0.09	0.3106	0.09869077	490	290	280	460
0.0088	0.09	0.294	0.08651344	515	350	250	430
0.0585	0.09	0.2692	0.07589089	480	350	250	400
0.0579	0.09	0.2543	0.0680209	480	380	200	400
0.0077	0.09	0.258	0.06662329	540	350	250	400
-0.0062	0.09	0.254	0.06455444	525	400	200	400

由于上述方程存在无穷多组解，而这些解中有大部分仅仅是在数学层面成立，在实际情境中并不可行。因此可以对 $\theta_1 \sim \theta_3$ 加入阈值，限定遍历的范围。

表格中所展示的是我们采集的部分展示舵机脉宽与目标位置对应关系的数据集。坐标原点取在机械臂舵机 2 的中心处。

从表格中的信息不难发现，舵机 2 的脉宽数据与 $x^2 + z^2$ 之间存在着近似线性的映射关系，我们可以根据这一近似的关系，将目标距离 $x^2 + z^2$ 进行粗略的分段，每一段对应一个确定的舵机脉宽。

因此，问题转换为已知 θ_1 的值，求解 θ_2 和 θ_3 ：

$$x = l_1 \cos \varphi + l_2 \cos(\theta_2 + \varphi) + l_3 \cos(\theta_3 + \theta_2 + \varphi)$$

$$y = l_1 \sin \varphi + l_2 \sin(\theta_2 + \varphi) + l_3 \sin(\theta_3 + \theta_2 + \varphi)$$

其中 φ 可以由 $x^2 + z^2$ 以及舵机 2 的脉宽-角度对于关系直接确定, θ_2 和 θ_3 是待求解的机械臂关节角度。为了便于求解, 定义中间变量 A, B:

$$\begin{cases} A = x - l_1 \cos \varphi \\ B = y - l_1 \sin \varphi \end{cases}$$

简化后的系统:

$$\begin{cases} A = l_2 \cos(\theta_2 + \varphi) + l_3 \cos(\theta_3 + \theta_2 + \varphi) \\ B = l_2 \sin(\theta_2 + \varphi) + l_3 \sin(\theta_3 + \theta_2 + \varphi) \end{cases}$$

将两个方程平方并相加:

$$A^2 + B^2 = l_2^2 + l_3^2 + 2l_2l_3 \cos \theta_3$$

解得:

$$\cos \theta_3 = \frac{A^2 + B^2 - l_2^2 - l_3^2}{2l_2l_3}$$

存在条件:

$$(l_2 - l_3)^2 \leq A^2 + B^2 \leq (l_2 + l_3)^2$$

令 $\alpha = \theta_2 + \varphi$:

$$\begin{cases} \cos \alpha = \frac{A(l_2 + l_3 \cos \theta_3) + Bl_3 \sin \theta_3}{D} \\ \sin \alpha = \frac{B(l_2 + l_3 \cos \theta_3) - Al_3 \sin \theta_3}{D} \end{cases}$$

其中 $D = l_2^2 + l_3^2 + 2l_2l_3 \cos \theta_3$ 。

最终解:

$$\theta_2 = \arctan \left(\frac{\sin \alpha}{\cos \alpha} \right) - \varphi$$

可以发现, 方程可以存在两组解, 分别对应 θ_3 和 $-\theta_3$, 但在工程实际

中，应取 $-\theta_3$ 。因为对于 θ_3 ，仅存在数学意义。代入舵机角度-脉宽的对应关系，即可完成逆运动学求解的优化方案。至此，完成了对于幻尔机械臂的逆运动学求解。

2. 物资与载物台

(1) 物资识别

由于机械臂采用测距模块进行抓取任务，故而对物资的识别没有要求。本组区分场地物资、起始物资的方案是使用将物资放入两个独立的物资盒中进行区分

(2) 载物台设计

本队的物资盒采用厚度为 5mm 的木制结构切割并拼装组成，物资盒实物如图所示。其具体采用传统布局，主体采用四周围挡和内部倾角组成。

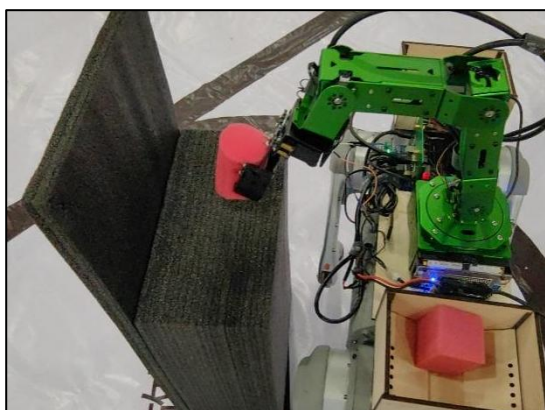


图 4.2.1 机械臂夹取物资

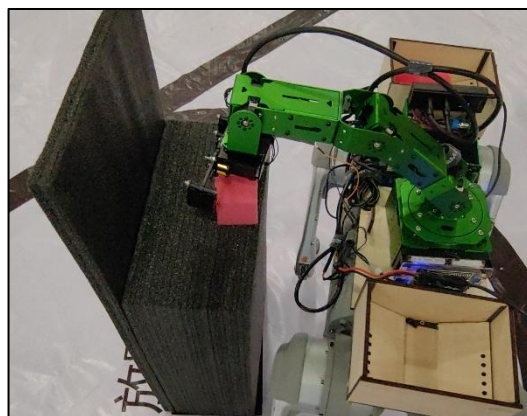


图 4.2.2 机械臂投放物资

盒内两侧倾角与水平底板大约呈一定角度的夹角，既能防止物资滑落，又能提供一个足够的坡度，使物资在机器狗行进过程中自然地保持

在盒子内中心位置；

两个盒子采用前后布局，考虑了本组使用的较多设备，故而，采用这种布局能灵活放置算力平台与其他配件，如拓展坞、电池等，且能与固定底板的螺丝一起，对算力平台起到纵向固定的作用，防止因机械狗运动时的振动导致模块接口松动，增加系统稳定性。

为了将机械臂与物资盒完美融合成一个系统，选用了一块木制底板，通过 Go2 背部拓展坞左右两条导轨将“机械臂——物资盒——算力平台”整合平台安装在机械狗背部，如图 4.2.3 所示。同时将物资盒通过纳米胶固定在前后两端，木板中间开孔用于固定连结线材，形成了一套稳定的系统，如图 4.2.4 所示。

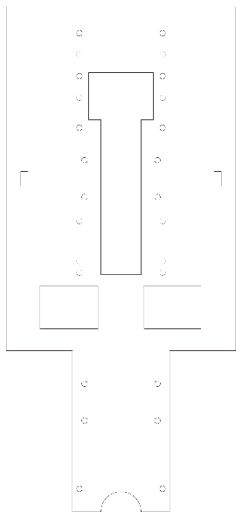


图 4.2.3 木制底板设计图

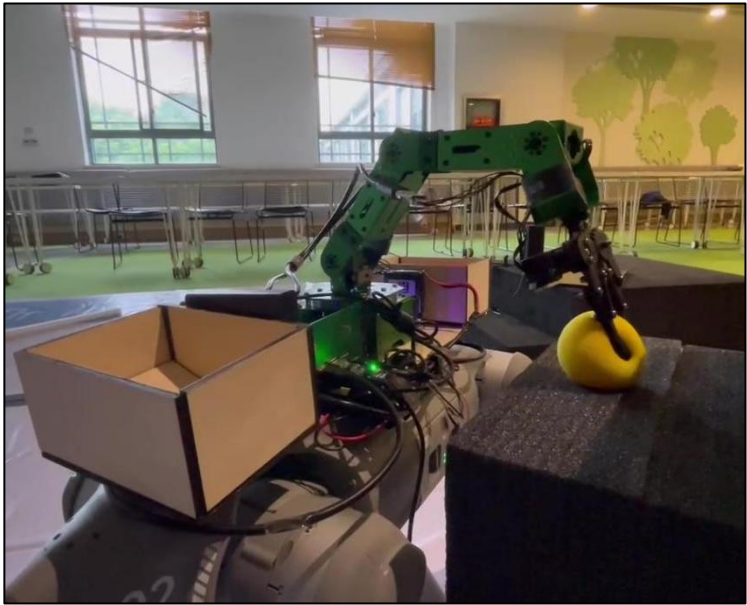


图 4.2.4 物资装载系统实物图

这样的设计不仅优化了充分考虑了平台设备整合的空间利用率，在有限的空间内，整合了组内所有需要的设备，并且相互之间干扰较小、设备运行稳定、易于维修检查，同时，使用类似于自锁结构的固定螺丝，使得平台在机械臂、机械狗运动下晃动较小，保证机夹取精度。

3. 机器视觉处理

(1) 外接摄像头设计

外加摄像头目的是为了获取更好视野，因此采用最为简单的高清无畸变摄像头模块即可，如图 4.3.1 所示。



图 4.3.1 本队所选择外接的摄像头

为获取足够的宽度和长度的前方视野，同时避免被外界环境干扰，团队决定采用正下方的视角。

(2) 获取原始图像 (origin_img)

调用 OpenCV 的 VideoCapture 方法，使用 V4L2 后端，将图像大小设置为 352×288 ，将 V4L2 获取的图像作为原始图像。

图 4.3.2 和图 4.3.3 展示了机器人所处的一个赛道的典型位置，以及在相应位置获取的原始图像。



图 4.3.2 机器人所处位置



图 4.3.3 摄像头捕获原始图像

(3) 图像二值化

由于本次赛题需要识别的线条为黑色，能和外部的白色背景形成很好的区分度，因此不需要进行 HSV 色域转换后掩膜，而是直接对原始图像二值化，便可以得到相对较为平滑的掩膜图像，具体的二值化处理流程为：首先对原始图像进行高斯滤波，去除噪点，之后设定二值化阈值对图像进行二值化，最后进行腐蚀膨胀，即可得到二值化图像，如图 4.3.4 所示。

(4) 获取轮廓与轮廓特殊值

将（3）二值化后得到的图像作为 Canny 检测的对象，得到边缘后，利用八邻域生长的方法得到左、右轮廓。

其中 Canny 算法可直接调用 OpenCV 库，效果如下图 4.3.5 所示：

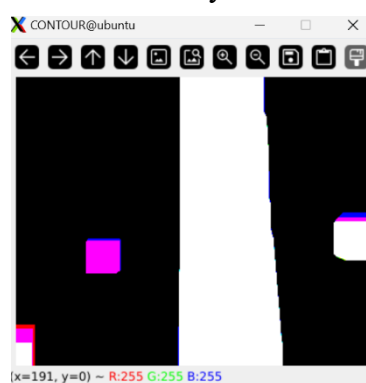


图 4.3.4 二值化所得图像

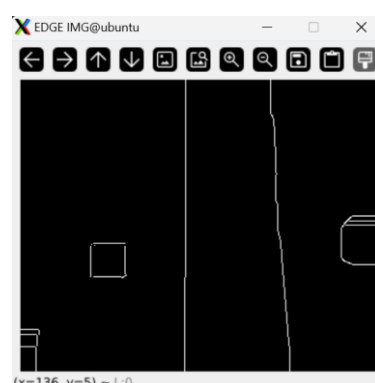


图 4.3.5 Canny 检测所得图像

得到边缘图像前，可先得出最长白列，以左、右最长白列作为起点搜索，利用八邻域生长算法，可以将左、右两边线条分类为两条轮廓，轮廓的数据类型为列表，在此情况下，顺序为从下到上。

得到两轮廓后，计算左、右轮廓的特征值。**特征值**包括左右轮廓各自中间部分的 x 坐标平均值（`even_x`）、靠近上方35%的平均斜率（`even_agl`）。得到的结果如下图 4.3.6 所示：

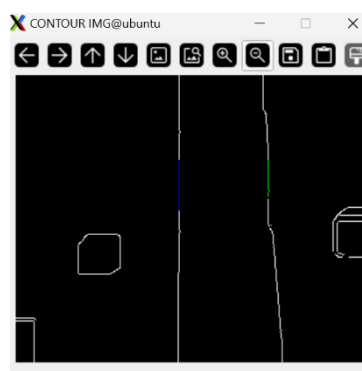


图 4.3.6 八邻域生长与特征计算所得图像

（5）识别 ArUco Tag

调用 OpenCV 的 `DetectMarkers` 方法，对输入图像进行识别。识别到 ArUco Tag 后将存储当前 ArUco Tag 的全局变量进行改变，从而实现状态机的切换。

4. 机器人动作处理

四足机器人的底层运动部分在内部 161 电脑上，在本团队方案中，将内部 161 电脑的控制部分视作下位机，将四足机器人背部挂载的 Orin Nano 视作上位机。上位机和下位机通过网线进行连接，基于 DDS 通信协议通过宇树官方提供的运动控制高层状态接口中将上位机计算得到的运动数据发送给下位机。

（1）状态机的使用

状态机初始化为起点巡线和避障的开始状态，在该状态下，将指令机器狗向左移动离开起点。当视野中蓝色区域靠左时，会切换为直行状态，在后腿超过起点障碍时，将向右移动一段距离，直到视野中出现黑色线条，随后开始在赛道上巡线运动。

赛道上不同区域会进入不同的状态，此处主要介绍处理较为棘手的状态。

1. 避障区的进入

处理：考虑到在避障区周围障碍与环境的对比度不高，直接使用视觉识别将带来很大困难，因此本团队选择在四足机器人右背部加装一个激光测距模块，用来实时测量四足机器人距离右侧障碍的距离，从而通过 PID 调整四足机器人在避障弯道的位姿，实际测试发现四足机器人应更加靠近左侧障碍板，才能确保四足机器人在 L 型弯道转弯时不容易碰到障碍。

动作：四足机器人将沿着右侧避障区通道前进，同时根据右测距的数据实时调整四足机器人的侧向速度，在到达 L 型弯道尽头机器人将同时向左转弯并向前行走，通过设置合适的转弯时间，即可全程不碰到任

何障碍通过避障区，如下图 4.4.1。



图 4.4.1

2. 放置区的进入

识别：利用 ArUco Tag 识别。在该状态难点主要在于如何确保每次四足机器人都能以一个相对合适的位置抓取放置平台上的物资，这需要建立在一个较为稳定的巡线 PID 基础上，再通过设定合理的定时时间，将四足机器人移动到指定位置。之后在四足机器人向左移动的过程中，应固定四足机器人侧向速度速度小一点以减小定时带来的误差，并适当给一个旋转速度，有利于保证机械臂与物资之间的距离适中。

动作：在识别到 ArUco Tag 以后，将沿着左线进入环岛，当到达放置平台正前方时，固定四足机器人向左移动一定距离，在抓取场地物资并卸好物资篮中的物资后，四足机器人向右移动一定距离，之后继续沿左线行走。

3. 上台阶

处理：由于本赛题要求四足机器人四条腿都位于台阶上，且全程不从台阶上摔落，因此准确识别到四足机器人何时在台阶平台上且如何以

一个合理的位姿上台阶是问题的关键。本团队选择调用四足机器人内部高层状态接口获取四足机器人的欧拉角。在上台阶过程中，四足机器人的俯仰角(pitch)绝对值逐渐增大，当四足机器人即将到达台阶顶面时，四足机器人的俯仰角绝对值逐渐减小，当该俯仰角绝对值减小到某一设定阈值后，将该条件作为四足机器人位于台阶顶面的条件。实际测试发现四足机器人上台阶过程中容易向右偏，若初始上台阶位置不当四足机器人容易从台阶右方摔落，因此在上台阶之前的巡线过程中应使得四足机器人稍微位于线的左侧一点。

动作：在沿着黑线偏左侧从正面上台阶，到达台阶顶面后固定四足机器人的转向速度，之后从正面下台阶，如下图 4.4.2。

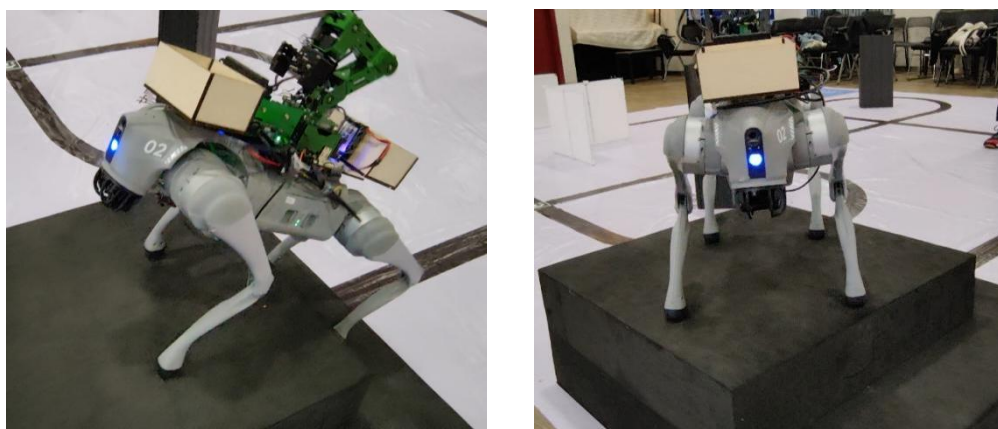


图 4.4.2

(2) 高层运动接口调用

为了对机器人进行运动控制，Orin Nano 会调用下面的高层运动接口：

`sport_client.move(float vx, float vy, float vyaw)`：用于控制四足机器人的移动速度，其中 `vx` 为前进速度，`vy` 为侧向速度，`vyaw` 为旋转速度。

`sport_client.SwitchGait(int d)`: 用于切换四足机器人的步态，其中 `d` 为步态枚举值，为了获得四足机器人运动的稳定性，本团队选择全程选择正向爬楼梯模式(`d=3`)。

为了获取四足机器人的运动状态，Orin Nano 会调用以下高层状态接口，通过订阅"rt/sportmodestate"话题即可获取到当前四足机器人的欧拉角。

5. 程序架构

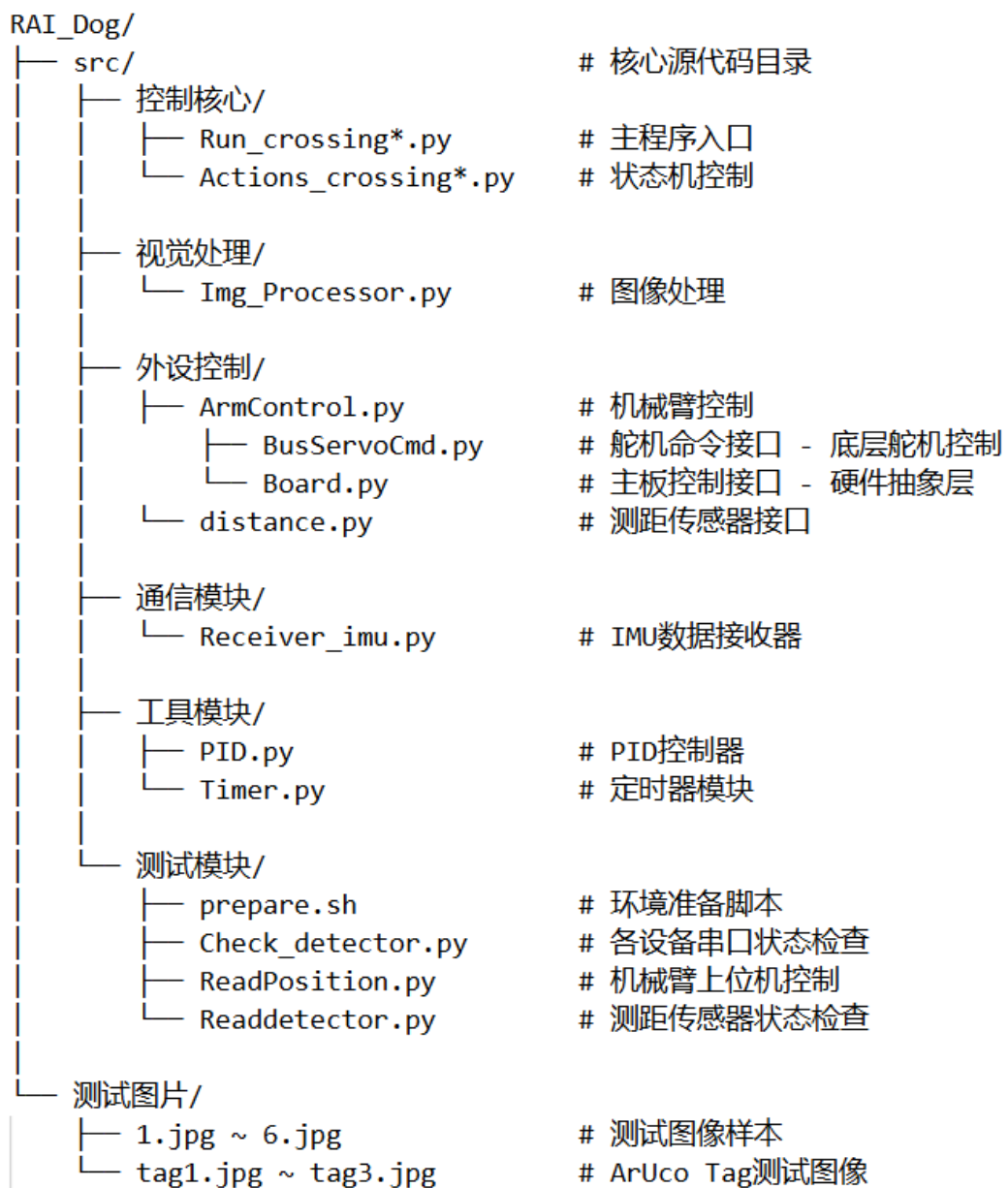


图 4.5.1 程序架构图

为保证程序的可维护性，我们对程序的结构进行了细致的设计，相关的源码已作为附件发送至负责人的邮箱中。具体程序架构图 4.5.1 所示。

6. 程序优化与硬件改进

在经历省赛后，暴露了一些问题，我们清晰认识到自己代码的鲁棒性和硬件的稳定性有待提高，于是在省赛结束后不久，我们便立刻潜心投入到新一轮开发中，我们分别对程序以及硬件做了优化：

(1) 程序优化

1. 执行速度提升

在省赛阶段我们更需要关注完整跑完全部流程，而要在国赛阶段崭露头角，我们不仅需要有跑完全程的能力，更需要提升速度。

首先，我们对程序整体运行做了优化：省赛时对一号、二号待检区的区分通过在要进入的待检区入口贴 ArUco Tag，再调用摄像头识别 tag 值决定是否进入。但是，考虑到识别 ArUco Tag 需要时间，并且存在识别错误的风险，因此，我们选择采用更简单高效的方式，将原先的代码拆分为 Run_crossing1 与 Run_crossing2 代码供比赛时选择。

同时，删除了部分对代码运行冗余的运行语句，提升四足机器人在一些精度要求不高区域的前进速度。我们也优化了机械臂抓取的代码，提升抓取效率。

2. 健壮性提升

在省赛阶段的调试环节，我们也暴露出许多由于硬件异常而产生的问题。多数情况下，硬件的异常会直接导致运行中的程序报错，从而停止运行。

为了提升程序整体的健壮性，我们加入了测试脚本，确保每一次运行前硬件连接无误。我们还在代码主体部分中引入了 `try` 语句，确保硬件错误状态下程序还能以默认值平稳运行。

（2） 硬件优化

在硬件方面，我们换用了一块连接更为牢固的 `hub`。同时，我们将所有测距模块的串口换为了 `CH343` 驱动方式，相较于原来的 `CH340` 驱动，新的驱动模块更加稳定，容易维护。

考虑到现有的硬件架构中传感器模块众多，主程序中调用频繁，容易出现串口编号错误。于是，针对现有的串口连接方式，我们编写了一套 `udev` 规则，从而提升抗干扰能力。

五、总结与感想

在准备睿抗机器人开发者大赛 CAIR 赛道的物资运送比赛过程中，我们从初始构想到最终成果展示，经历了从无到有的过程。每一次挫折都化为前进的动力，作为下一步的优化方向。通过对赛制的深入分析，我们团队将任务细分并明确分工：

1. 视觉模块：利用机器狗的外接摄像头，精准识别赛道中的复杂元素；
2. 物资模块：设计并优化机械臂及物资盒，以确保高效夹取与投送物资；
3. 运动模块：调试机器狗的步态和相应的 PID 参数，提升其运动稳定性。

这种高效的协作模式不仅帮助我们层层分解了比赛的技术要求，也显著提升了每位成员的专业能力和工作效率。

经过数月的反复调试与技术迭代，我们的四足机器人成功完成了物资识别、运送与投放任务。在此过程中，我们攻克了诸如四足机器人步态稳定性、机械臂的高精度夹取与投送、外接摄像头视野、多串口协同工作等关键技术问题，具体来说：

1. 尝试机器狗的不同步态，最大限度减少行走间抖动对视觉识别的干扰；
2. 改进机械臂夹取算法，确保物资夹取与放置的高精度；
3. 调整外接摄像头的倾斜角度，优化机器狗的视野以满足视觉算法需求。

通过解决这些问题，我们深刻体会到理论知识与实际应用之间的差距，并学会了如何在有限时间内优化算法，提升机器人性能。

在调试的过程中，机器狗曾出现过故障，这对我们的工作进度造成了不小的影响。然而，宇树公司的工程师们在关键时刻提供了专业的支持，他们帮助我们迅速诊断并修复了问题，还与我们分享了许多宝贵的技术经验和调试技巧。此外，宇树公司提供的四足机器人和开源代码，也为我们的开发工作奠定了坚实基础，这些宝贵的支持和指导让我们受益匪浅。在此，我们对杭州宇树科技有限公司及工程师们表示由衷的感

谢。

同时我们也衷心感谢睿抗机器人开发者大赛，为大学生技术创新搭建了展示平台，为年轻工程师提供了展示才华的机会。团队成员携手共进，攻克技术难题，这些经历将成为我们学习生涯中最宝贵的财富之一。

此次比赛还使我们认识到，随着机器人技术的飞速发展，四足机器人的应用前景愈发广阔，尤其在物流配送和应急救援等领域。国家和社会的进步愈发依赖于高科技成果的广泛应用。我们坚信，通过不懈努力和 innovation，四足机器人将在更多领域大放异彩，助力社会进步。

最后，再次向睿抗机器人开发者大赛的主办方及宇树科技表达诚挚的感谢。此次比赛不仅让我们收获了知识和经验，更激发了我们对未来科技探索的热情。我们期待在未来的日子里，能有更多机会参与此类盛会，为我国乃至全球的科技创新贡献力量！