



北京市八一学校



FTC 27570

荣臻开物队

Bayi Embodied AI Robotic

工程笔记

BEAR

目录

- 一、学校介绍
- 二、团队成员与分工：精诚协作，各司其职
- 三、任务策略：扬长避短，高效得分
 - (一) 策略研究
 - (二) 底盘
 - (三) 前伸滑轨
 - (四) 升降滑轨
- 五、得分方案：高效稳定，灵活应变
 - (一) 自动阶段得分方案
 - (二) 手动阶段得分方案
- 六、程序
 - (一) 视觉
 - (二) 手动
 - (三) 自动
- 七、在社区的分享与交流

一、学校介绍

北京市八一学校，一所由聂荣臻元帅亲手缔造，充满红色基因与光荣传统的学府。学校秉承“科学引领，全面发展”的教育理念，致力于培养具备卓越科技素养和创新能力的未来人才。校园科技氛围浓厚，尤其重视国防科技人才的早期培养，为国家输送了大批优秀毕业生。

2016 年，国家主席习近平亲临母校视察，并对我校科技社团工作做出重要指示，极大地鼓舞了师生，也使我校科技教育受到社会各界的广泛关注与认可。同年，我校与中国航天等单位深度合作，由中学生自主设计并全程参与研制的中国首颗科普卫星——“八一少年行”卫星成功发射升空。历经八年发展，“八一”系列科普卫星工程已成功发射四颗卫星，后续研制任务已规划至“八一 11 星”，充分彰显了我校在科技创新教育领域的领先地位和深厚积淀。

二、团队成员与分工：精诚协作，各司其职

27570 是一支由多名学生组成的富有活力与创造力的团队。团队成员分工明确，各有所长，在程序、结构、外联、文创设计和 3D 建模等领域协同合作，共同推进项目进展。

- **程序组：**李牧洲（自动阶段程序），蒋金成（手动阶段程序），葛翊扬
- **结构组：**侯沐宸，杨梓皓，肖浩天
- **外联组：**丁致灏，辛若溪，崔艺冰
- **文创设计：**肖浩天
- **3D 建模组：**侯沐宸，蒋金成，葛翊扬，虞梦飞

三、任务策略：扬长避短，高效得分

（一）策略研究

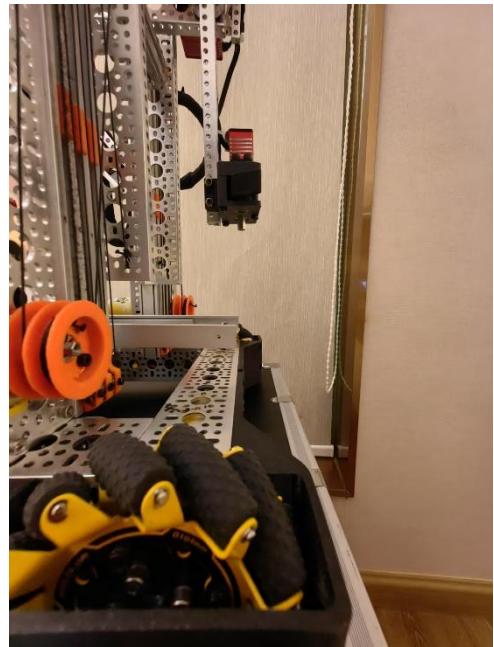
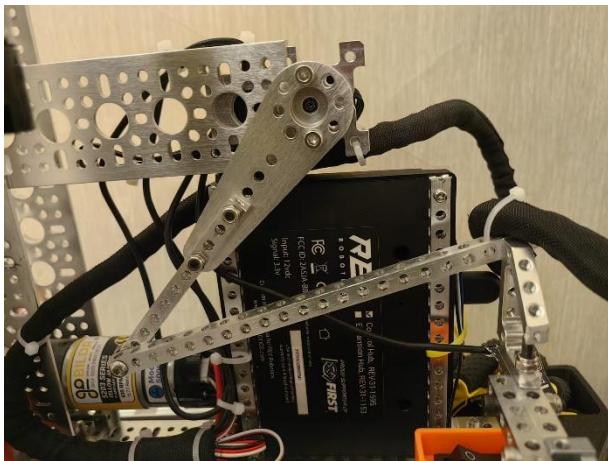
在深入研读比赛规则后，我们团队充分分析了自身优势与劣势。考虑到我们拥有扎实的 3D 打印技术基础和优秀的程序设计能力，并结合对场地和得分效率的分析，我们制定了以**自动阶段 Chamber 高效挂标本**为核心策略的方案。

我们发现，自动阶段将标本挂在 High Chamber 的得分效率远高于 High Basket，而手动阶段则相反。因此，我们果断选择 Chamber 作为自动阶段的主要得分目标，同时设计了兼顾 High Basket 的框型结构，以确保在与队友目标冲突时仍具备一定的得分能力。

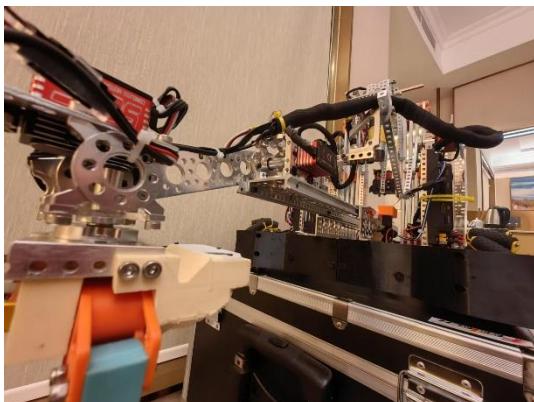
滑轨方案选择： 经过对比赛规则和各队伍方案的深入研究与对比，我们最终选择了**前后双滑轨方案**。

- **前伸滑轨：** 为确保稳定性和速度，经过对比测试，我们选择了**滑轨竖直放置和连杆驱动方案**。

- **升降滑轨：**为保证协同工作和动力充足，后滑轨采用硬连接，双电机驱动方案。



观察到：不管怎么抓取都要有一个爪子抬起来越过杠进入方形区域抓取的动作，所以我们直接选择用一节滑轨，在滑轨末端加装小臂，因为小臂有一定长度，对于抓取范围并没有很多的影响，同时直接转动小臂也可以更快完成越杠的动作，还无形中增加了自由度和与框交接的便携度。对于后滑轨的升降方案，我们选择了最传统也是最简便效率最高的方案，绕线滑轮升降，这样可以保证滑轨的长度和稳定性以及可控性。

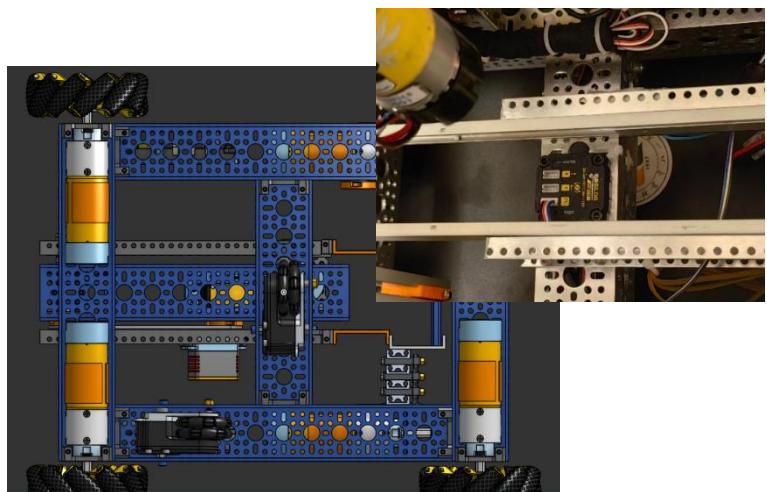


我们采用了 **EDP** 工程流程，以确保我们的工程不断迭代，高效解决每个问题。



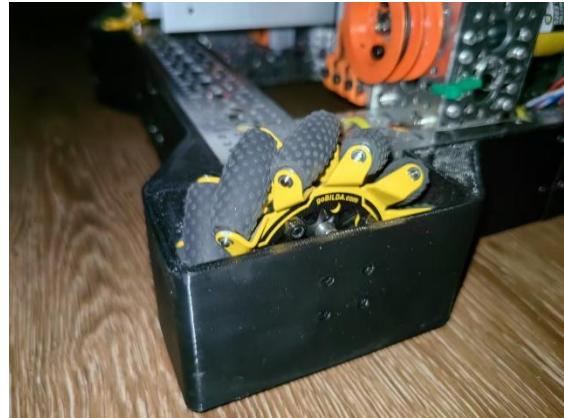
(二) 底盘

使用四个 11 大单位 C 型梁，符合正常配重车型，加装 2 个编码器和一个 IMU，并将其放置车辆的旋转中心。



第一次用 PLA—CF 这种含碳材料制作全包围但是因为碳件到一定程度会脆所以用 TPU 材质制作了全包围，这种材质压力承受强，还可以形变后恢复原状，同时可以吸收一部分撞击力不至于撞击力太大把车弹开，为我们撞到中心区域同时完成挂样本的策略保护

(三) 前伸滑轨

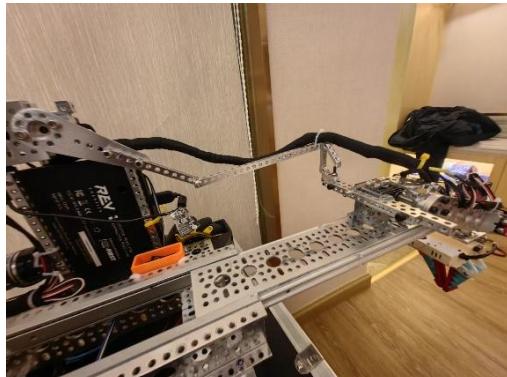


1. 滑轨长度：为最大限度提升抓取范围，并减少摩擦、增加稳定性，我们在底盘允许范围内选用了**最长滑轨**，确保机器人能够触及场地中央区域抓取物块。

2. 滑轨连接：

- 滑轨间连接：**为克服单滑轨承重不足的问题，我们采用**双滑轨协同工作方案**，并使用**10 单位长 C 型梁**进行简洁高效的连接，竖直放置的滑轨和水平连接的 C 型梁也为后续搭建预留了充足的孔位。

用



- 滑轨与车体连接：**由于竖直滑轨无法直接连接车体，我们选用**两根等长 C 型梁**作为刚性连接件，连接车体前梁和滑轨，梁的长度恰好从车前梁延伸至车体中部，保证连接的稳固性。

3. 滑轨动力来源：我们创新性地采了**连杆驱动**而非传统的绕线方式。

- 优势分析：**连杆驱动方案在单节滑轨情况下效率更高，且占用空间更小。连杆机构能有效放大力矩，提供稳定可靠的动力传输。相较于皮带传动可能打滑、链条传动可能跳齿等问题，连杆驱动在正常工况下运动传递更稳定，且能承受较大载荷，结构坚固。这对于需要伸缩抓取机构的机器人至关重要。

(四) 升降滑轨

为优化机器人升降装置的稳定性及重心分布，我们设计了一套基于 3D 打印的滑轨套件。该装置采用双侧对称设计，使用 4 节 35 厘米滑轨，并通过 3D 打印绕线轮实现升降功能，与前部爪子无缝衔接。

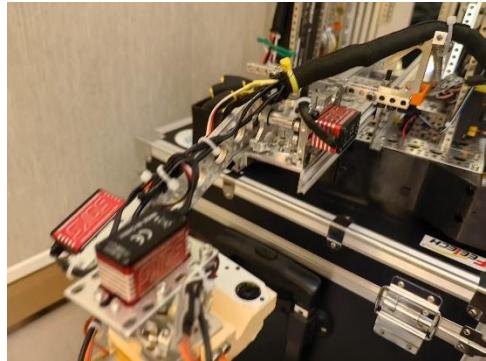
因为原来的滑轨通过铝合金 U 型梁固定两颗电机，提供稳固的动力输出，同时优化了整车重心，使滑轨收起时更贴近底盘，提升车辆平稳性。线路沿滑轨布置，并采用固定扣减少。



经过测试，滑轨运行平稳，协同作业精准，耐久性良好。我们优化了绕线轮齿形结构，并增加缓冲垫片以吸收振动。未来将探索轻量化材料，进一步提升性能。该设计显著增强了机器人在比赛中的表现。

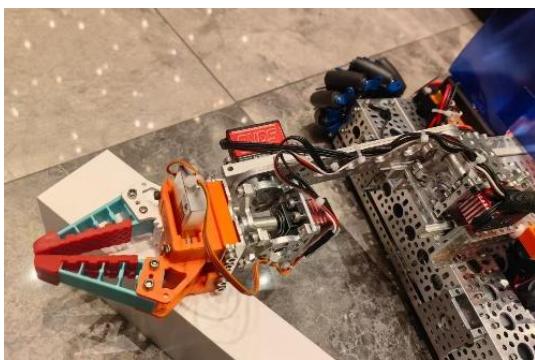
(四) 小臂和抓取爪

小臂设计：通过观察分析，我们发现所有方案都需要抬升爪子以越过取物区的横梁，除非爪子长度足够。因此，我们在滑轨末端创新性地加装了**小臂**，赋予爪子竖直方向的自由度，轻松越过横梁障碍。小臂的加入不仅对抓取范围影响甚微，反而因其长度优势，加速了越梁动作的完成，并提升了机构的自由度和与框体交互的便捷性。



抓取爪迭代： 我们对抓取爪方案进行了两次迭代。

- 初版爪：**设计小巧，能适应极限角度抓取，但夹持力不足，易损伤舵机。
- 二版爪：**加宽爪体，提升了夹持力，但高度受限，极限角度抓取能力下降。



•**终版爪 (鳍条爪)：**最终，我们选择了**鳍条爪**方案。该爪具有极高的适应性，能完美贴合本赛季样本形状。适当施力即可稳固扣住整个物块。我们在大众模型基础上进行了优化，将原平行开合结构改为**筷子式尖端抓取**，实现了仅夹持物块单边（无论长边或短边）即可可靠回收滑轨，极大地提升了抓取效率和适应性。

靠回收滑轨，极大地提升了抓取效率和适应性。

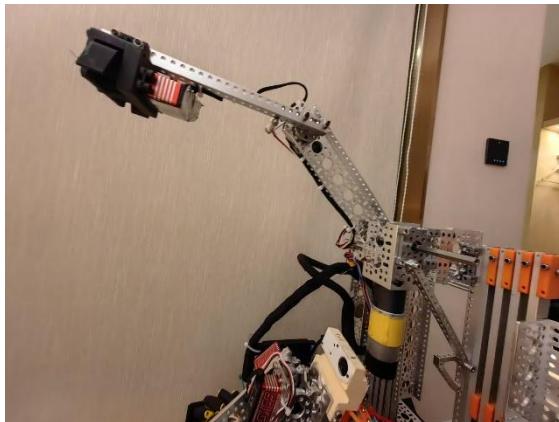
五、得分方案：高效稳定，灵活应变

(一) 自动阶段得分方案

原方案：在后升降滑轨上加装爪子，抓取标本后旋转 180° ，再降下后滑轨完成 Chamber 挂载动作。

新方案 (优化后)：资格赛检验表明，原方案上线较低，自动阶段 Chamber 得分效率是核心。因此，我们摒弃转向动作，追求快速挂载。我们创新性地提出利用车体撞击中心区域同时完成挂载的方案。

- **臂式挂载机构：**借鉴优化现有方案，我们设计了一款新型**两节臂式挂载机构**。



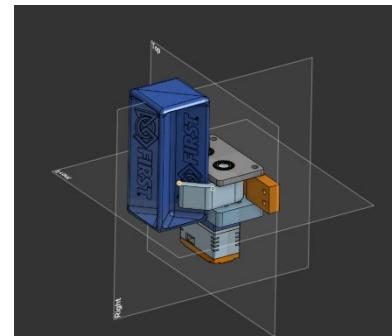
- **一级臂驱动：**采用 19.2:1 减速比电机 驱动一级臂，通过齿轮进一步减速，提供足够扭矩，承担大行程转动和定位任务。
- **二级臂驱动：**采用大扭矩、高速度舵机 驱动二级臂，与一级臂程序协同，精准控制臂的运动轨迹，确保

臂能准确到达指定位置。

- **双臂优势：**两节臂组合解决了单节臂抓样本不好控制、样本钩子方向不匹配等问题，总行程接近 270°，完美适应样本挂载需求。

自动抓取优化：原方案采用快速走线抓取地面物块，但受车体误差影响，稳定性不足。经过多次尝试，我们优化为更少走线的方案：

- **滑道式中转：**抓取地面物块后，将其放置于车体中央的滑道中，物块通过滑道滑至车后，推回观察区，再进行挂载 Chamber 的循环。
- **优势：**充分发挥了鳍条爪的适应性和车体中心区域无干扰的优势，无需转向，避免了走线不准带来的误差，提升了自动抓取的稳定性和效率。
- **视觉：**我们同样在自动阶段中加入视觉，在一开始挂上第一个块时伸出前滑轨开始识别，当挂杠的爪子松开后则立刻调用基于视觉的自动驾驶在潜水区抓取一个样本并且带回观察区准备得分。
- **爪子：**一款与标准舵机适配，全含碳打印材料，与标本内侧完美贴合。



(二) 手动阶段得分方案

手动阶段，我们依托 Road Runner 运动库，实现了场上任意位置的大臂复位，并设计了基于三次样条曲线的自动驾驶放框得分程序，大幅提升了手动阶段的得分效率和操作便捷

六、程序

(一) 视觉:

我们的视觉程序旨在开发一个鲁棒且高效的自动化手动程序，核心目标是实现：

- **自动抓取：**完成从方块检测到最终抓取的完整自动化流程。
- **鲁棒性：**程序应能在不同光照条件和场地环境下稳定运行，并具备一定的错误处理能力。

FTC 竞赛中，自动化能够显著提高队伍的得分效率和竞争力。视觉引导的**自主抓取流程**可以显著提高**效率和一致性**，特别是在场地环境复杂或光照条件变化的情况下。

视觉处理管道：

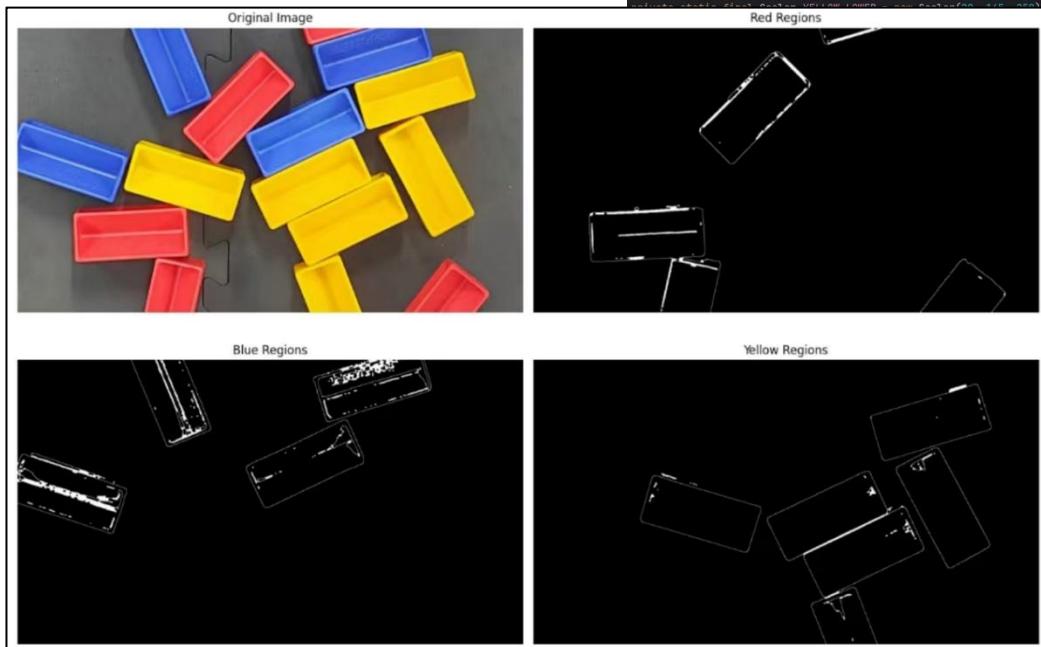
- **自动化方面：**完全自动化地完成**图像分析**和**目标检测**，无需人为干预，实现机器人的**自主视觉感知**。

- 颜色空间转换 (RGB to HSV): 将 RGB 空间转换为 HSV 空间。HSV 更符合人类对颜色的感知，且在光照变化下，色调 (Hue) 分量相对稳定，有利于颜色分割的鲁棒性。转换公式涉及非线性变换，例如：

- $V = \max(R, G, B)$
- $S = (V - \min(R, G, B)) / V \text{ (if } V \neq 0 \text{) else } 0$
- H = 条件分支函数，根据 RGB 的大小关系计算色调角度。

- 颜色阈值分割：基于颜色范围进行图像分割，提取特定颜色区域。
- 轮廓检测：膨胀+腐蚀算法，通过膨胀识别到的边缘（边缘由于光照不是完整的）将断续的边连接，随后再腐蚀

```
private static final Scalar RED_LOWER_1 = new Scalar(170, 100, 100);
private static final Scalar RED_UPPER_1 = new Scalar(180, 255, 255);
private static final Scalar RED_LOWER_2 = new Scalar(0, 100, 100);
private static final Scalar RED_UPPER_2 = new Scalar(10, 255, 255);
private static final Scalar BLUE_LOWER = new Scalar(100, 125, 220);
private static final Scalar BLUE_UPPER = new Scalar(130, 255, 255);
```



成连续的轮廓，掩码如图

- 使用边缘检测和轮廓跟踪算法，在二值化图像中提取轮廓。用于目标识别和定位。
 - 最小外接矩形：使用外接矩形拟合轮廓，得到物体的姿态信息。
 - 世界坐标计算：基于透视投影原理，将图像像素坐标转换为世界坐标。
像素坐标 (u, v) 与世界坐标 (X, Y, Z) 之间存在以下关系：
 - $u = f * X / Z + c_x$
 - $v = f * Y / Z + c_y$
 其中 f 为焦距， (c_x, c_y) 为主点坐标。在本程序中，假设方块位于同一深度平面 (Z 恒定)，简化为二维坐标转换，
`pixelsToCmRatio` 是焦距和深度共同作用下的比例系数。
 - 遇到的问题：
 - 噪声和误检：图像噪声、阴影、反光等因素导致误检，需要进行多特征融合的目标验证。面积阈值 (`MIN_AREA_THRESHOLD_PIXELS`) 和尺寸限制 (`MIN_REGION, SIDE_MAX_SIZE`) 是我们采用的滤波方法
 - 运动指令平滑性：由于舵机抖动，结果直接用于控制会导致运动抖动。
移动平均滤波是简单的低通滤波器，可以滤除高频噪声，平滑运动指令。
移动平均滤波是计算滑动窗口内数据的平均值，公式如下：
- $y[n] = (1/M) * \sum(x[i]), i = n-M+1 \text{ to } n$
- 其中 $y[n]$ 是第 n 个平滑后的输出， $x[i]$ 是输入序列， M 是窗口大小。

机器人运动控制模块

- 功能：负责运动控制，集成 Road Runner 实现精确路径规划和运动控制。
- 意义：是程序执行视觉指令，完成自主导航和抓取的执行机构。
- 由于摄像头安装位置较低，识别到的样本大概率不是完整的，于是加入了面积检测，检测到小面积的（意味着无法准确抓取）样本首先执行接近动作。

```
try {
    Vector2d targetPosition = new Vector2d(
        x: drive.pose.position.x + closestCube.centerXCm * 0.39370,
        y: drive.pose.position.y - closestCube.centerYCm * 0.39370
    );
    Action approachMovement = drive.actionBuilder(drive.pose)
        .splineToConstantHeading(targetPosition, tangent: 0)
        .build();
    Actions.runBlocking(approachMovement);
    telemetry.addLine( lineCaption: "Road Runner 接近动作完成." );
    currentAutoState = AutoState.GRABBING; // 接近完成后自动进入抓取状态
}
```

舵机控制模块：

- 舵机角度偏移量计算：`servoPositionOffset` 的计算
 $(\text{SERVO_ANGLE_COEFFICIENT} * \text{DIRECTION_MULTIPLIER_HENG} * \text{clawAngle_deviation} + \text{ANGLE_OFFSET_HENG})$ 建立视觉检测到的方块角度偏差与舵机角度调整量之间的线性关系。
`SERVO_ANGLE_COEFFICIENT` 是角度偏差到舵机角度的比例系数。

程序的创新性：

- **视觉引导的自主抓取：**实现了从方块检测、接近、定位、抓取的完整自动化流程，实现了真正的自主抓取功能，摆脱了对场地标记物或人为引导的依赖。
- **智能融合：**将视觉系统和 Road Runner 运动库结合，利用视觉提供目标信息，利用 Road Runner 实现路径规划和运动控制，充分发挥了各自的优势。
- **自主决策能力：**程序根据检测结果自主决策，如选择目标颜色、调整角度等。
- **移动平均滤波：**移动平均滤波平滑视觉指令，提高了机器人运动的平稳性。

通过持续的改进和优化，我们相信我们的视觉程序将在 FTC 竞赛中发挥重要作用，并为未来更高级的机器人自主系统奠定坚实的基础。

(二) 手动

比赛用手动程序目标是构建一个高效、可靠、且易于驾驶员掌控的 TeleOp 程序：

- **竞争优化：**程序考虑了比赛策略和操作效率，例如预设位置、快捷操作等。
- **复杂机构的协同控制：**设计直观且高效的控制方案，让驾驶员能协调控制机构。本程序力求在控制复杂性和操作便捷性之间取得平衡。
- **高精度运动控制：**程序需要提供灵敏且可预测的底盘控制，并结合场域坐标系转换，方便驾驶员进行精确定位。

机器人驾驶控制模块 (driveRobot()):

- **无头模式的实现：**
 - **场域坐标系转换：**为使驾驶员操作直观，程序实现了场域坐标转换。驾驶员摇杆输入，希望对应机器人在场地上相对于环境的左右平移和前后运动。但机器人底盘运动基于**机器人自身坐标系**。因此需要通过**旋转变换矩阵**将摇杆输入从场域转换到载具坐标系。转换公式如下：
 - $[rotX] = [\cos(-\theta) \ -\sin(-\theta)] [x]$
 - $[rotY] = [\sin(-\theta) \ \cos(-\theta)] [y]$

其中 (x, y) 是摇杆在场域坐标系下的输入， $(rotX, rotY)$ 是转换到机器人坐标系下的输入， θ 是机器人航向角 (robotHeading)，实现了精确的无头模式。

执行机构控制模块：

- 能够灵活操作机器人的各种执行机构，完成抓取、放置、挂钩等比赛任务。
- **控制方面：**采用按键映射的方式，将按键输入与执行机构的运动指令关联。滑轨升降机构采用状态机管理和预设位置，简化驾驶员的操作。
 - **直流电机编码器控制：**滑轨升降机构使用直流电机和编码器进行位置闭环控制。程序通过 `setTargetPosition()` 和 `setMode` 将电机设置为**位置控制模式**，电机根据编码器反馈，自动调整功率，使电机位置精确跟踪目标位置。位置控制的 **PID 控制算法** 在电机驱动器内部实现。

控制特点：

- 手动遥控为主，辅助自动化：**程序以手动遥控为主，核心控制逻辑由驾驶员操作完成。但在部分机构控制（如滑轨升降）中，使用了状态机和预设位置等辅助自动化功能，简化驾驶员的操作。
- 场域坐标系转换的驾驶控制：**引入场域坐标系转换实现无头模式，提高驾驶操控的直观性和便捷性。
- 半自动操作潜力：**位姿记录功能为后续扩展半自动操作提供了基础。

手动阶段基于 roadrunner 的自动驾驶程序：

```
drive.actionBuilder(drive.pose)
    //初次抓快构建三次样条曲线之前复位执行机构
    .stopAndAdd(new ChamberDrive5Park.ArmMotorAction((DcMotorEx) bigArmMotor, false, p: 0))
    .stopAndAdd(new ChamberDrive5Park.ServoAction(backArmServo, BACK_ARM_RESET_POSITION))
    .stopAndAdd(new ChamberDrive5Park.MotorAction((DcMotorEx) Left_Hanging_Motor, (DcMotorEx) Right_Hanging_Motor, p: 0))
    .stopAndAdd(new ChamberDrive5Park.ServoAction(backgrap, p: 0))
    .splineToSplineHeading(recordedCatchPose, tangent: 0)
    .waitSeconds( t: 0.2)
    .stopAndAdd(new ChamberDrive5Park.ServoAction(backgrap, p: 0.6))
    .build();
```

这一部分用于在场地的任何位置将大臂恢复为初始状态获取得分标本

这两行定义了**两个常量**，可以直接通过设置第一个数改变挂多少个标本，也就是启动几个循环，第二个常量是偏移量，由于 GoBilda 坐标系横轴为 Y 且正方向向左，所以这个常量设置了每一次的挂块位置

```
for (int i = 0; i < numberChamberPoints; i++) {
    Vector2d ChamberPoseInConstantHeading = new Vector2d(
        x: current_chamber_pose.position.x - 0,
        y: current_chamber_pose.position.y + 5 * 0.39370
    );
    drive.actionBuilder(drive.pose)
        //在挂块之前伸出大臂，抬高滑轨，张开后爪
        .stopAndAdd(new ChamberDrive5Park.MotorAction((DcMotorEx) Left_Hanging_Motor, (DcMotorEx) Right_Hanging_Motor, LIFT_UP_POSITION))
        .waitSeconds( t: 0.1)
        .stopAndAdd(new ChamberDrive5Park.ServoAction(backArmServo, BACK_ARM_SET_POSITION))
        .waitSeconds( t: 0.1)
        .stopAndAdd(new ChamberDrive5Park.ArmMotorAction((DcMotorEx) bigArmMotor, false, targetMotorPosition))
        .splineToConstantHeading(ChamberPoseInConstantHeading, tangent: 0)//挂杆位置
        .waitSeconds( t: 0.1)
        .stopAndAdd(new ChamberDrive5Park.ServoAction(backgrap, p: 0))//打开后爪
        .waitSeconds( t: 0.1)
        //在抓块之前复位
        .stopAndAdd(new ChamberDrive5Park.ArmMotorAction((DcMotorEx) bigArmMotor, false, p: 0))
        .stopAndAdd(new ChamberDrive5Park.ServoAction(backArmServo, BACK_ARM_RESET_POSITION))
        .stopAndAdd(new ChamberDrive5Park.MotorAction((DcMotorEx) Left_Hanging_Motor, (DcMotorEx) Right_Hanging_Motor, p: 0))
        .splineToConstantHeading(CatchPoseInConstantHeading, tangent: 0)
        .waitSeconds( t: 0.1)
        .stopAndAdd(new ChamberDrive5Park.ServoAction(backgrap, p: 0.6))
        .build();
```

如图是完整的基于 for 循环的手动阶段样本的得分自动，每一次向右偏移 5cm

对于放框，我们有手动阶段自动通过构建三次样条曲线自动驾驶并且自动得分：

```
Actions.runBlocking(drive.actionBuilder(drive.pose).
    splineToSplineHeading(recordedPoseBasket, tangent: 0)
    .build());
```

(三) 自动

能力: 5 + park

1. 驱动引擎

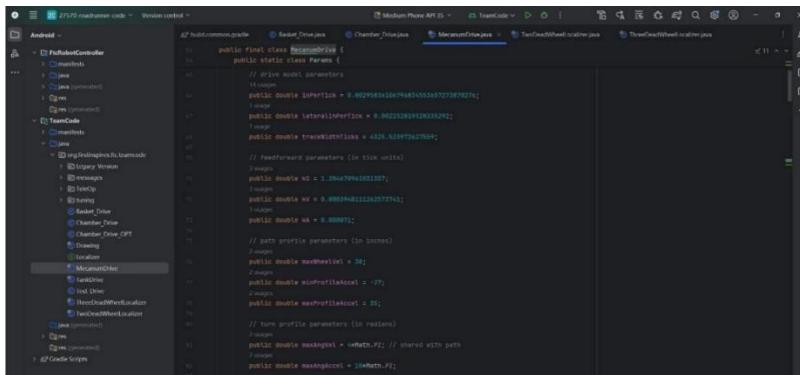
我们使用了广泛应用的 Roadrunner1.0 搭建自动程序的核心，Roadrunner 可以帮助我们在自动阶段更好的将整车的性能展现出来。通过 Roadrunner 搭建的底盘程序，可以实现高效的曲线走线，曲线走线同时进行航向调整，以及运用增益系数使我们的机器能够动态调整位置和航向及时纠错。在动态矫正下的机器能够实现即使受到外力撞击也能够及时回到预定航线上。

2. 外置编码器

本队使用了 goBILDA 的 odometry pod (Swing-arm)。通过 dead-wheel 定位会比麦轮直接定位要精准很多，可以保证在自动阶段高速运行的时候也有准确的定位。

3. 程序思路

通过 Roadrunner 教学网站上的示例，我们自己构建了三个动作控制函数：PatientMotorAction、PatientServoAction 和 MotorAction。这两个函数会控制在动作完成之后进行下一个操作，而 MotorAction 函数则满足底盘走线和滑轨动作并行，节省上升和下降的时间。

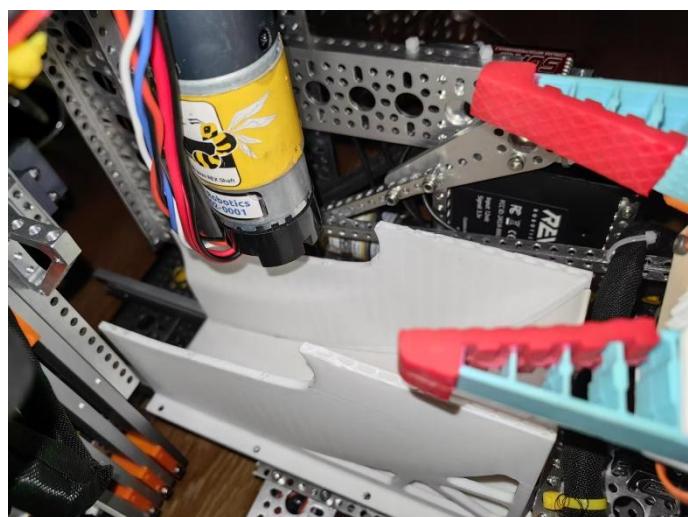


The screenshot shows the Android Studio interface with the project tree on the left and a code editor on the right. The code editor displays a Java file named 'RobotDrive.java'. The code defines a class 'RobotDrive' with static parameters for drive motors, lateral drive, and track width. It also includes feedforward parameters for the drive and lateral drive, as well as profile parameters for the turn profile.

```
public final class RobotDrive {
    public static class Params {
        // drive motor parameters
        public double leftDrivePower = 0.002980310679983445367273879279;
        ...
        public double lateralDrivePower = 0.00221891200330292;
        ...
        public double trackWidthInches = 4321.629773647768;
        ...
        // feedforward parameters (in cm units)
        ...
        public double x0 = 1.384679646051807;
        ...
        public double y0 = 0.00034611123273741;
        ...
        public double z0 = 0.000971;
        ...
        // path profile parameters (in inches)
        ...
        public double maxVelocity = 34;
        ...
        public double microVelocity = 77;
        ...
        public double maxProfileAccel = 30;
        ...
        // turn profile parameters (in radians)
        ...
        public double maxAngle = 4.0Math.PI; // stored with path
        ...
        public double maxAngleAccel = 16.0Math.PI;
    }
}
```

4. 策略思路

我们在自动阶段通过快速走线的方式抓取地面上的块达到 5 + park 的效果但是这并不稳定，走线会因为车各种误差导致不稳定。在经过多次尝试后我们选择了一种走线更少的方式：抓取地上的块放到车中间的滑道中，使块通过滑道滑到车后推回观察区再进行挂样本至 chamber 的循环，充分发挥了鳍条爪子的优势和车中心无干扰，无需转向的优势避免了因走线而产生的大量误差。

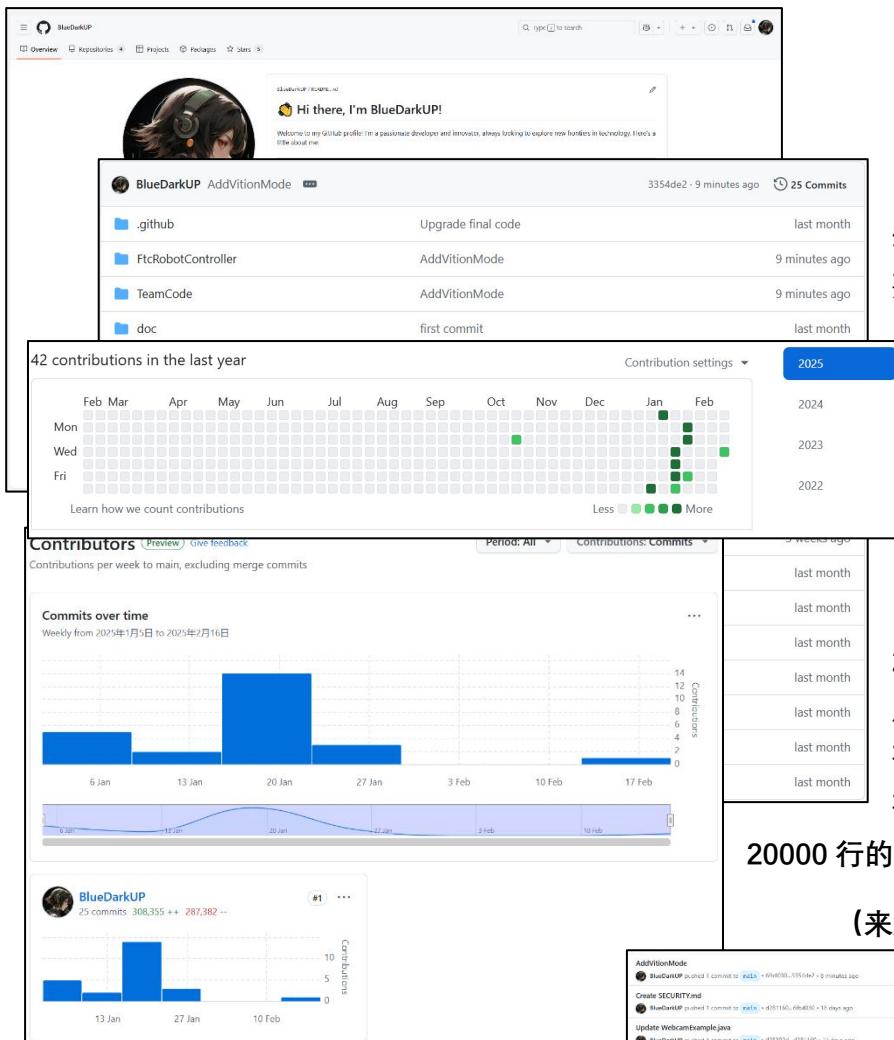


七、在社区的分享与交流：

我们的程序员（用户 BlueDarkUP，2022 年入站 Github）

将代码以 FTC 官方提供的 BSD-3-Clause-Clear 证书开源在了 Github 中的仓库

“BlueDarkUP/FTC-27570”



这其中包括我们的自动程序，手动程序和一系列驱动，总计约 300MB

我们所有仓库总共提交了 42 次修改

自开源起仓库发布了 25 次更改，从第一代程序新增了 308,355 行代码，修改了 287,382 行代码，相当于新增了约 20000 行的功能

(来源 GitHub 统计功能)

开源的内容包括我们完整的创新成果，如视觉识别，基于 RR 的自动驾驶。每个代码都带有详细的注释，并且我们程序的每一次提交都带有更改的解释

AddVitionMode	BlueDarkUP pushed 1 commit to FRC + 61e4030...195cde4 + 5 minutes ago	...
Create SECURITY.md	BlueDarkUP pushed 1 commit to FRC + 43f1160...fbfb453 + 15 days ago	...
Update WebcamExample.java	BlueDarkUP pushed 1 commit to FRC + d153502...d28196 + 22 days ago	...
remove unused files	BlueDarkUP pushed 1 commit to FRC + aef1bd6...d5550a + 22 days ago	...
REMOVE WRONG FILES	BlueDarkUP pushed 1 commit to FRC + f4a50bb...aef156a + 28 days ago	...
Update CubeDetectionTeleOp.java	BlueDarkUP pushed 1 commit to FRC + 1fd442...fcd7db + 21 days ago	...
upgrade CubeDetectionTeleOp	BlueDarkUP pushed 1 commit to FRC + c1da296...1fb7f4a + 2x days ago	...
add visual processing!	BlueDarkUP pushed 1 commit to FRC + 9179f6d...351a9b + 2x days ago	...
Add comments to SingleStickWithRR.java	BlueDarkUP pushed 1 commit to FRC + 1ef0282...957076a + 26 days ago	...
Update README.md	BlueDarkUP pushed 1 commit to FRC + 42b2121...12cf529 + 27 days ago	...
Update README.md	BlueDarkUP pushed 1 commit to FRC + f996232...425d201 + 27 days ago	...
Update README.md	BlueDarkUP pushed 1 commit to FRC + 9029e0b...9991007 + 27 days ago	...
Update README.md	BlueDarkUP pushed 1 commit to FRC + 502f944...0028000 + 27 days ago	...
Update README.md	BlueDarkUP pushed 1 commit to FRC + 7a22b2b...322d944 + 27 days ago	...
Update SingleStickWithRR.java	BlueDarkUP pushed 1 commit to FRC + e8f7wR...7h2jWlw + 27 days ago	...
Update README.md	BlueDarkUP pushed 1 commit to FRC + e8f7wR...7h2jWlw + 27 days ago	...
Update README.md	BlueDarkUP pushed 1 commit to FRC + e8f7wR...7h2jWlw + 27 days ago	...

我们的 README.md 包含详细的代码使用方案，取得的奖项，赞助商，我们的 Discord 网址和 YouTube, Facebook, Instagram 主页网址，每个网站上都发布了我们的设计经验和创新成果

The image is a collage of screenshots from various platforms showcasing the team's work and achievements:

- GitHub README:** A screenshot of the team's GitHub repository page. It includes sections for "Our Code" (Android Studio and Roadrunner), "Key Features" (Android Studio, Roadrunner, optimization), and "Recent Achievements" (Design Award - 1st Place, Control Award - 1st Place).
- Join Our Community:** A screenshot of the team's Discord invite page.
- Acknowledgments:** A screenshot of the team's acknowledgments page, thanking Feetech.
- Contact Us:** A screenshot of the team's contact information page, listing YouTube, Facebook, and Instagram handles.
- Tips:** A screenshot of the team's tips page, encouraging users to modify and improve their code.
- Facebook Profile:** A screenshot of the team's Facebook profile for "Ni Ming".
- Instagram Profile:** A screenshot of the team's Instagram profile for "yangzh0916".
- YouTube Channel:** A screenshot of the team's YouTube channel for "Allen Yang". The channel has 16 subscribers and 10 videos. Recent uploads include "A look back at our journey to the top" (123 views, 17 hours ago), "PID algorithm by dashboard" (923 views, 1 month ago), "Our robot's first score!" (443 views, 2 months ago), and "Our robot's first autonomous program" (408 views, 2 months ago).

我们同样在各种交流会中分享自己的参赛经验，从别的队伍中学习思路和创新点。也通过很多队伍解决了困扰我们许久的 GoBilda 外置编码器无法正常使用的问题，使我们也学会了 RR 的使用和底盘的 PID 控制。在北京地区的 FTC 社区中，我们参与了西城科技馆、十一学校、人大附中等队伍组织的线下技术交流活动。我们希望通过 FTC 竞赛，帮助更多队伍，开始了解机器人，学习机器人，爱上机器人。



Happy coding, and may the best robot win! 🤖🚀

特别鸣谢 | 赞助商：

深圳飞特模型有限公司



深圳飞特模型有限公司

中国航天科技集团



中国航天科技集团公司
China Aerospace Science and Technology Corporation