

ROBOMASTER 2019
冬令营 第二组答辩

团队分工与合作

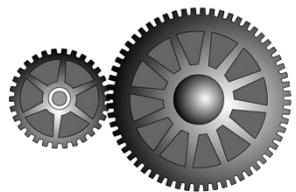
方向划分

- 机械：黄炫宇，罗宇城，姜何飞飞
- 嵌入式：杨锬瀚，刘梓熙
- 算法：刘腾渊，杨子超

再 | 细分
↓

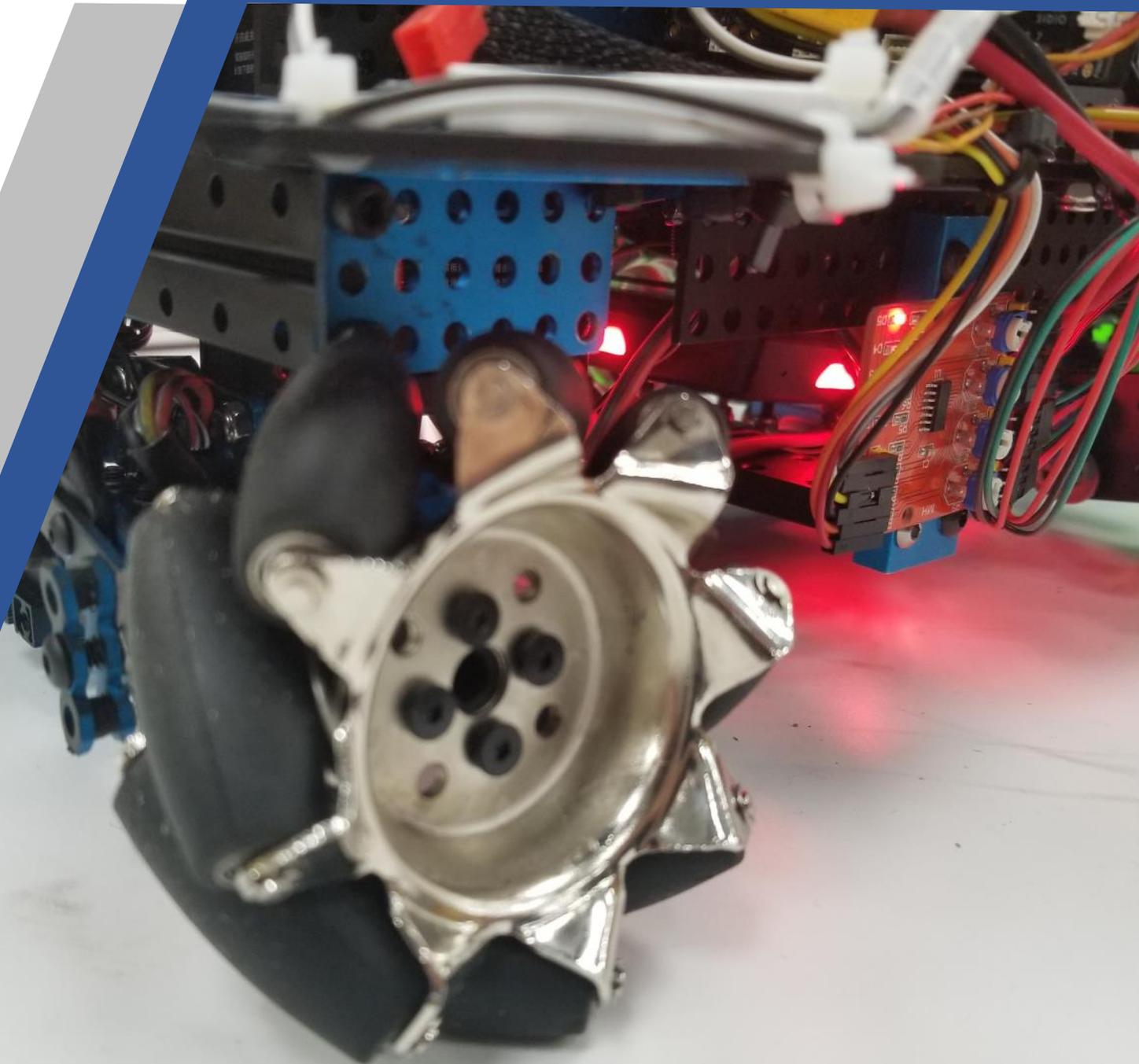
- 黄炫宇，罗宇城：解密机器人及底盘设计
- 姜何飞飞：外设与结构搭建
- 杨锬瀚：底盘控制与自动化实现
- 刘梓熙：通讯协议与机器人执行器设备
- 刘腾渊：视觉与上位机数据可视化设计
- 杨子超：寻路算法与通讯对接

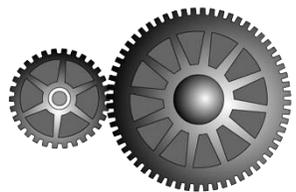




机械

- 实际需求总体分析
- 整体设计
- 问题及其解决方案
- 迭代



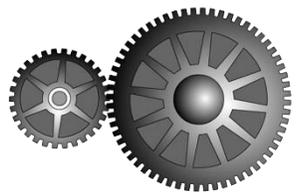


机械

- 需求分析
- 解密机器人
- 工兵机器人
- 对接装置

夹取物块的效率以及解码速度是机械在比赛中脱颖而出的关键。

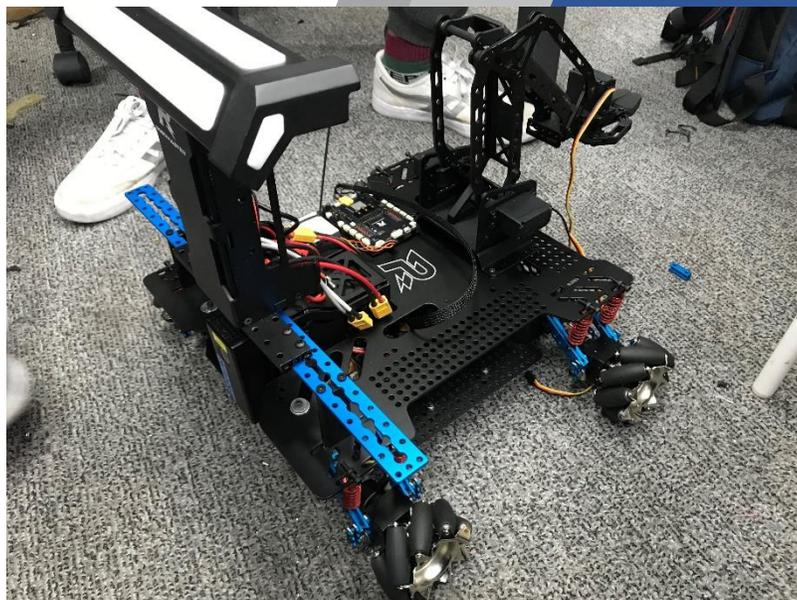




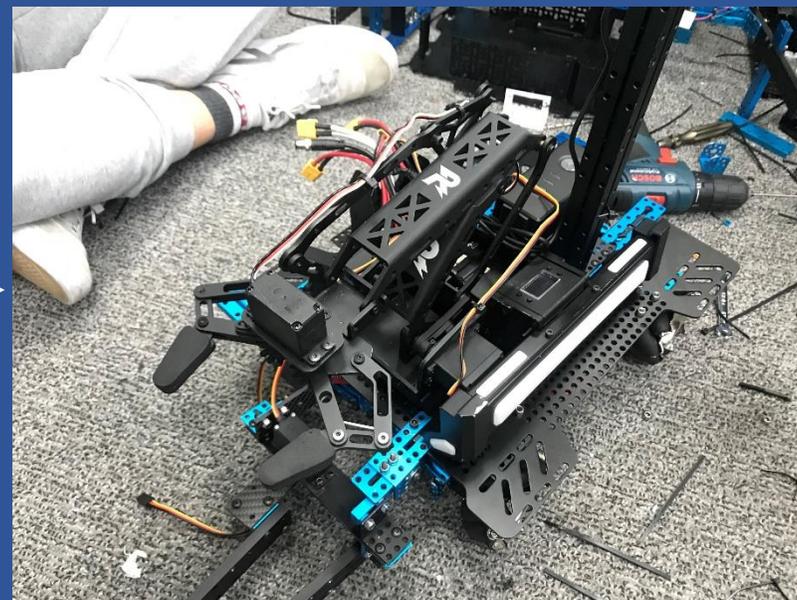
机械

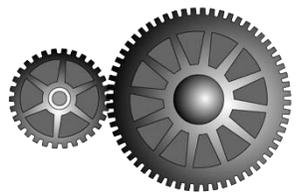
- 设计
- 工兵机器人

底盘设计与迭代



迭代

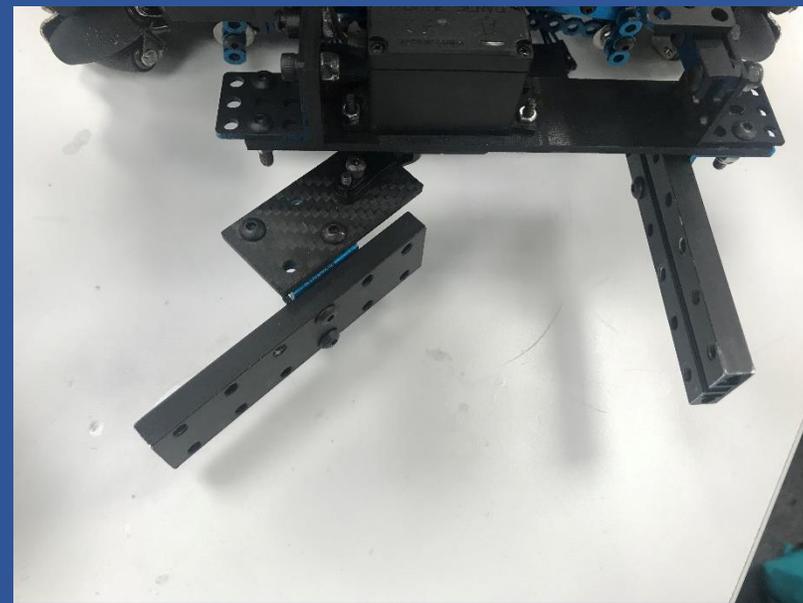
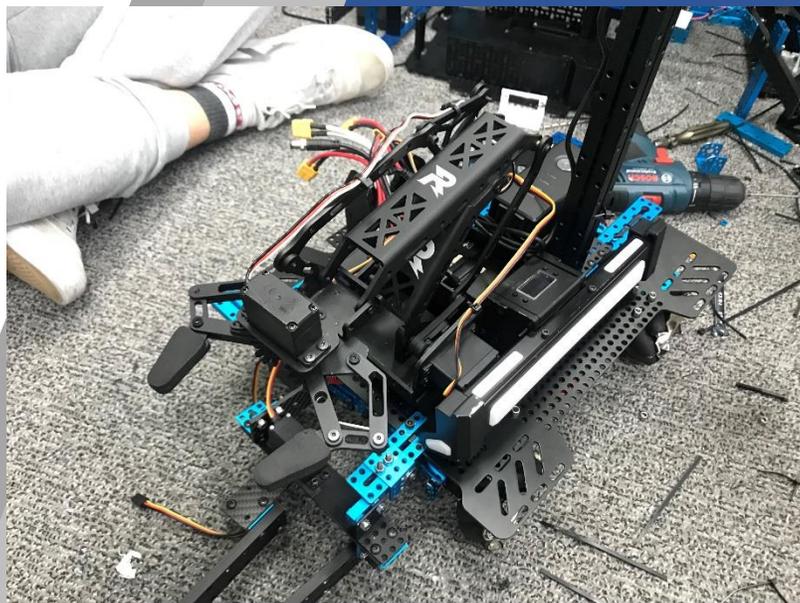


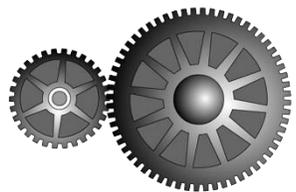


机械

- 设计
- 工兵机器人

执行器的设计

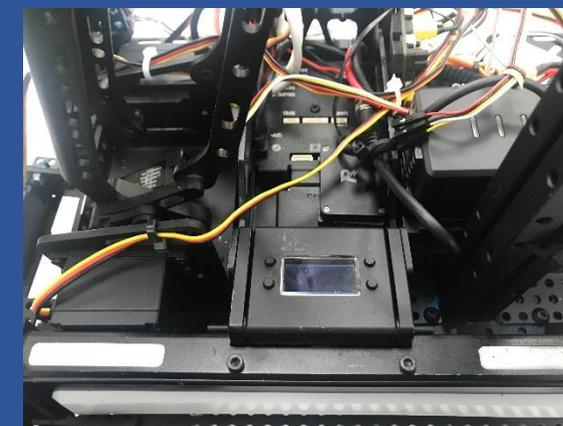
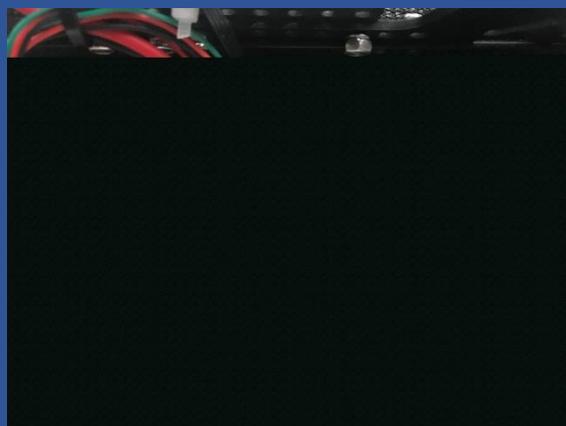
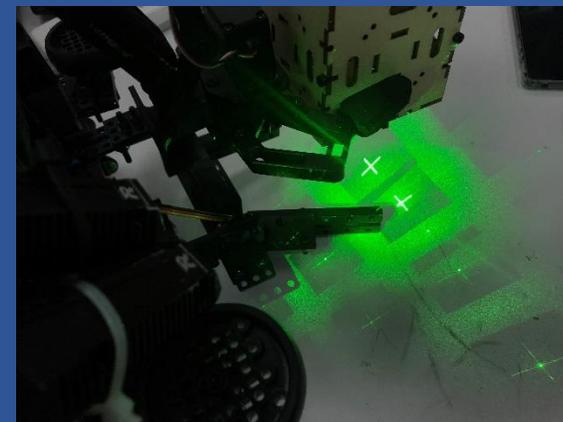
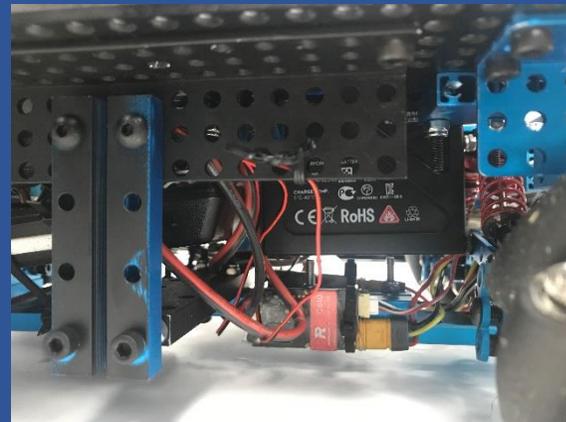


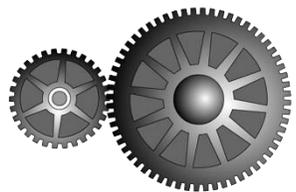


机械

- 设计
- 工兵机器人

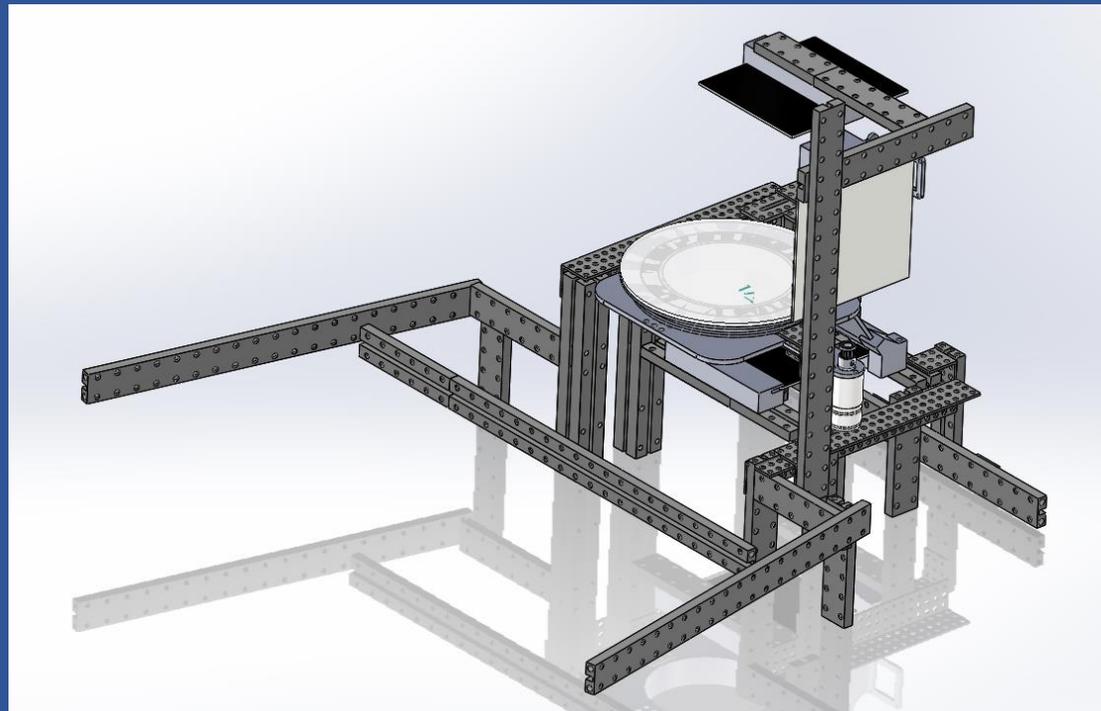
考虑整体——专注细节





机械

- 设计
-解密机器人



需求 → 方案论证 → 制作 → 改进与迭代



ROBOMASTER



嵌入式

- 项目需求的分析
- 底盘控制的设计
- 问题与解决方案
- 迭代
- 总结





嵌入式

- 项目需求的分析

有监控的自动化



ROBOMASTER



嵌入式

- 项目需求的分析

有监控的自动化



简单任务：底盘在空间内的运动

→ 自动

复杂任务：方块的夹取与规避其他机器人

→ 手动

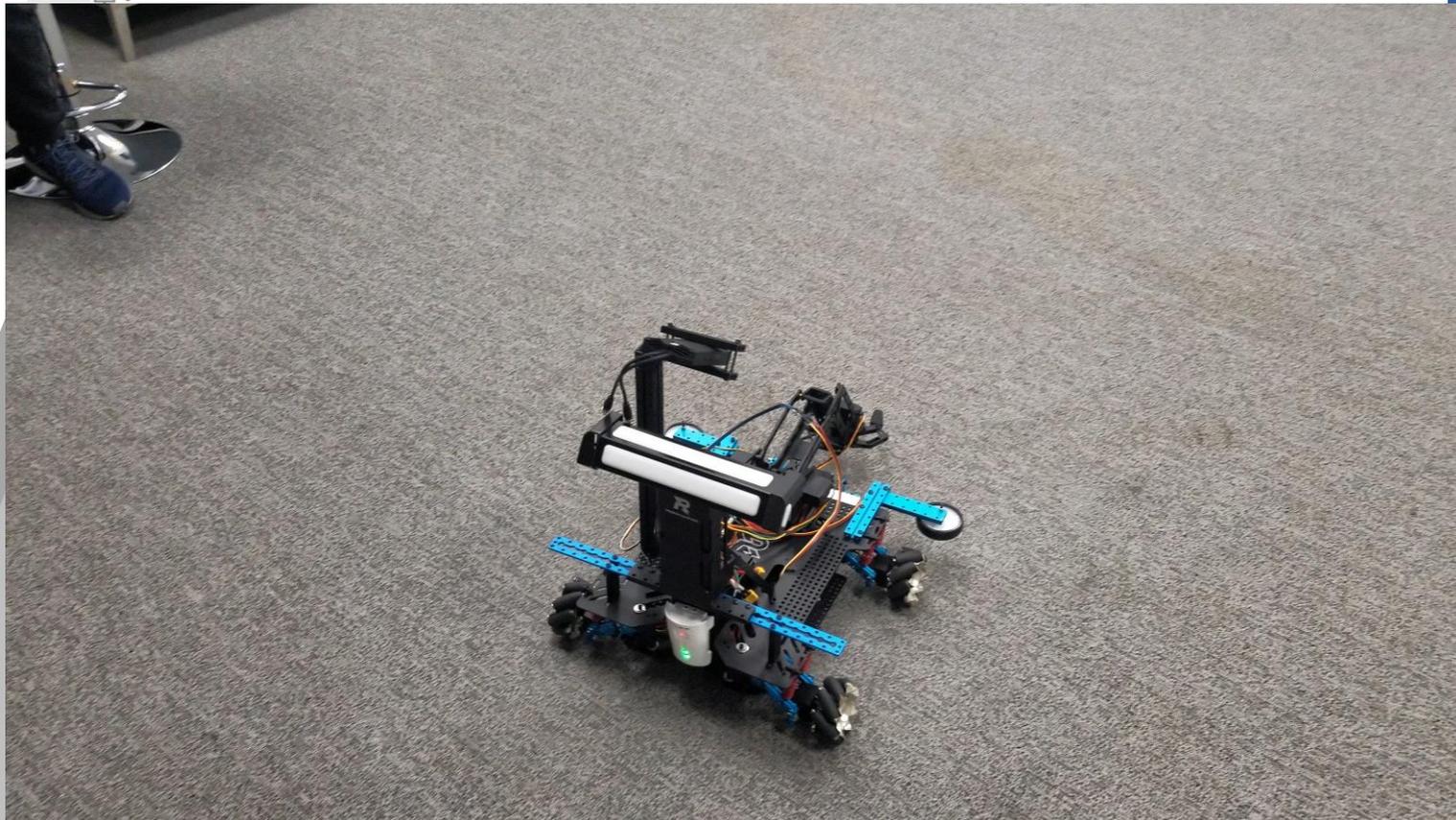




嵌入式

- 底盘控制
- 底盘的运动学里程计

```
375 void chassis_motor_position_pidc2(float coordinate_x, float coordinate_y)
376 {}
```



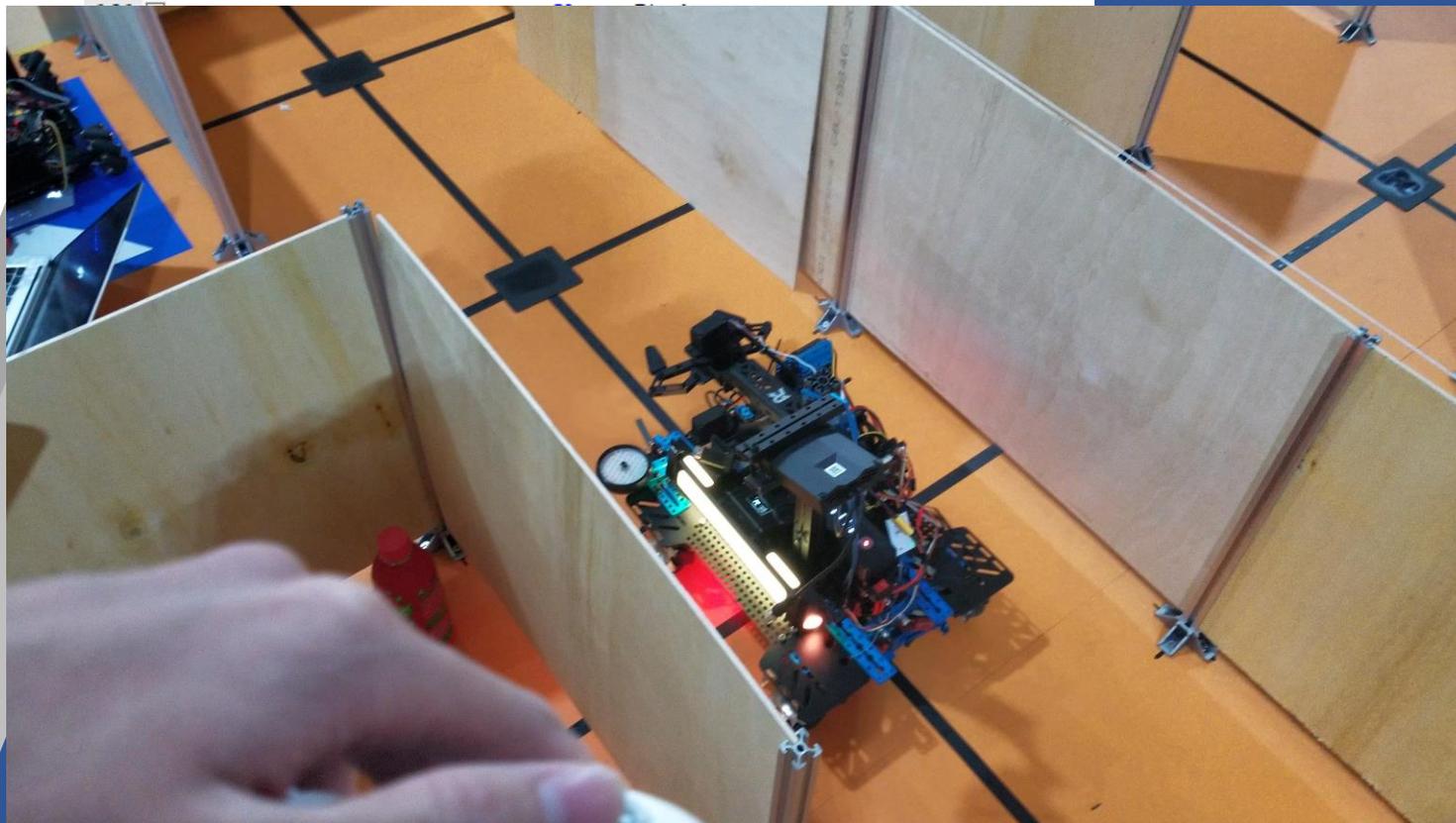
```
404 }
405 }
406 chassis_close_loop_calculate();
407 send_chassis_moto_current(chassis_moto_current);
408 }
```





嵌入式

- 底盘控制
 - 底盘的运动学里程计
 - 巡线设计与合作
 - 旋转角定位



```
98 void chassis_angle_control(int angle_goal,  
99                             int angle_error,  
100                            int angle_state,  
101                            int angle_ref)
```

```
130         w_speed == w_speedmax,  
131     }  
132 }  
133 if(fount_road.state[0] == 1 && fount_road.state[3])  
134 {  
135     angle_state = 0;  
136     chassis.vw = 0;  
137 }  
138 }  
139 chassis_moto_speed_calc(0,0,chassis.vw,chassis_moto_speed_ref);  
140 chassis_close_loop_calculate();  
141 send_chassis_moto_current(chassis_moto_current);  
142 }
```





嵌入式

- 底盘控制
 - 底盘的运动学里程计
 - 巡线设计与合作
 - 旋转角定位
 - RFID位置定位
 - 技术结合自动化实现
 - 执行器控制
 - 通讯对接





算法

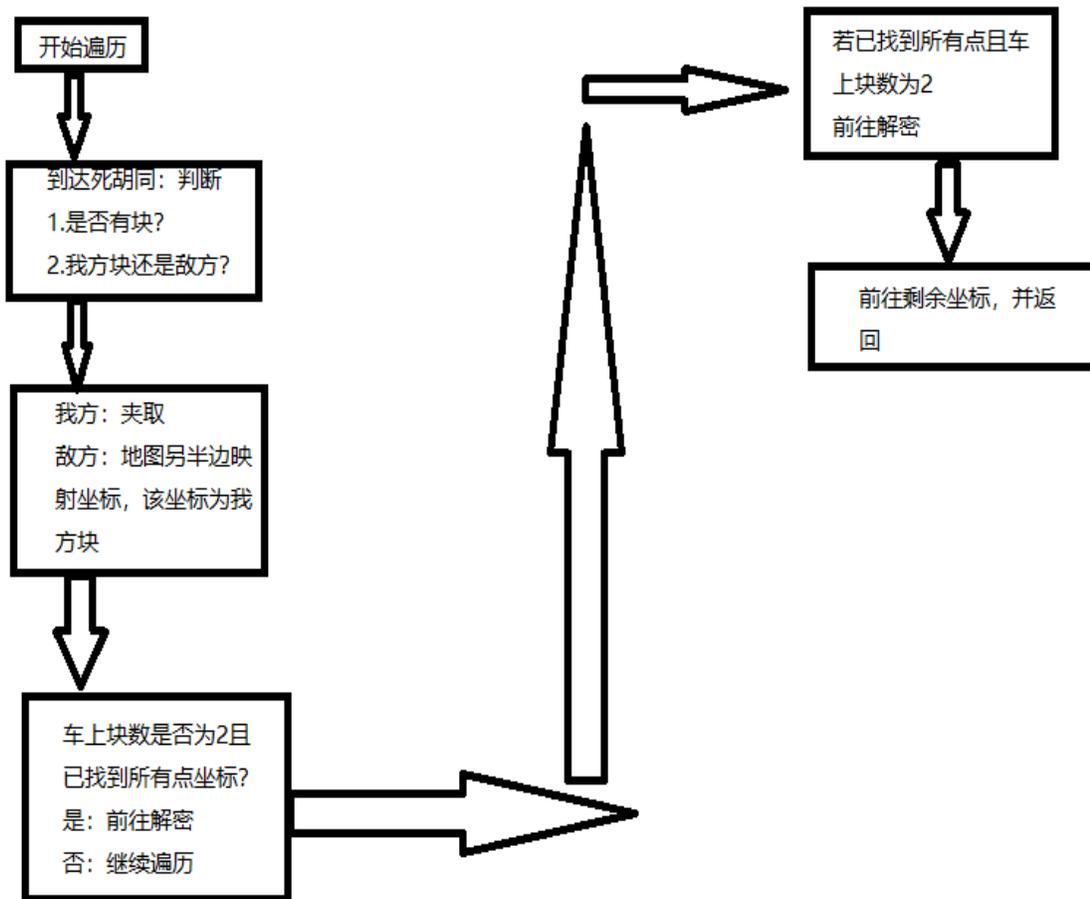
- 需求分析
- 发现问题，提出问题
- 解决问题
- 设计思路
- 总结





算法

- 需求分析
- 发现问题，提出问题
- 解决问题





算法

- 需求分析
- 发现问题，提出问题
- 解决问题

- 1、概率
- 2、密度
- 3、万有引力
- 4、梯度下降法 & Meanshift





算法

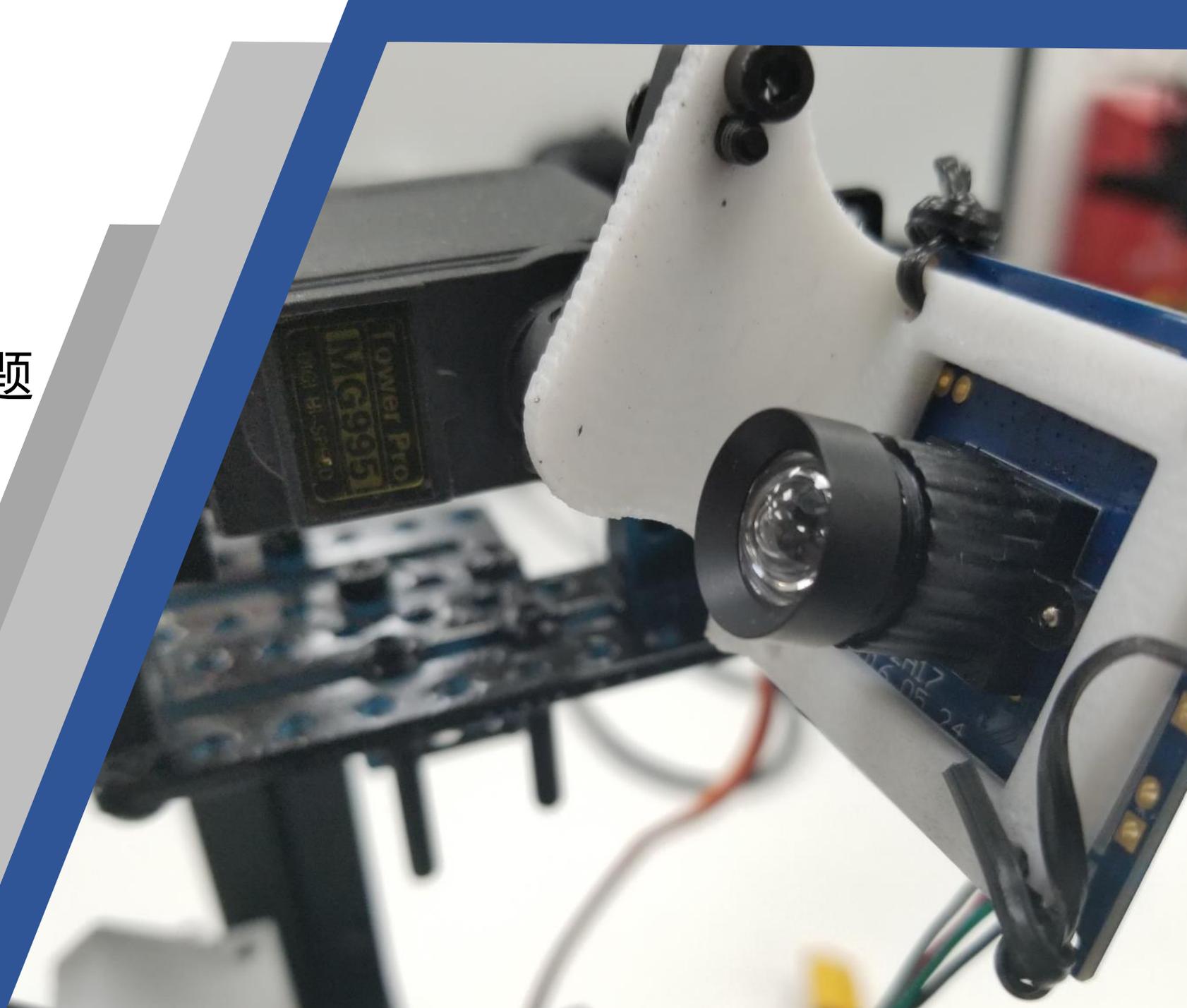
- 需求分析
 - 发现问题，提出问题
 - 解决问题
- 4个未知点位=1个已知点位（概率实时变动）
 - 发现了3个密文，直接抹除所有未知点
 - M/R^2
 - R 为该点位到目前位置的最短距离
 - 计算坡度，并根据坡度移动
 - 当出现多个方向坡度相同时，寻找到那个方向下一格的坡度；
 - 如果还相似，随机一个方向移动
 - 发现敌方块，直接无视掉，同时在对应位置生成我方已知点
 - 发现我方块，直接拿取，同时在对应位置抹除点位
 - 至高原则，拿了两个块，直接去终点

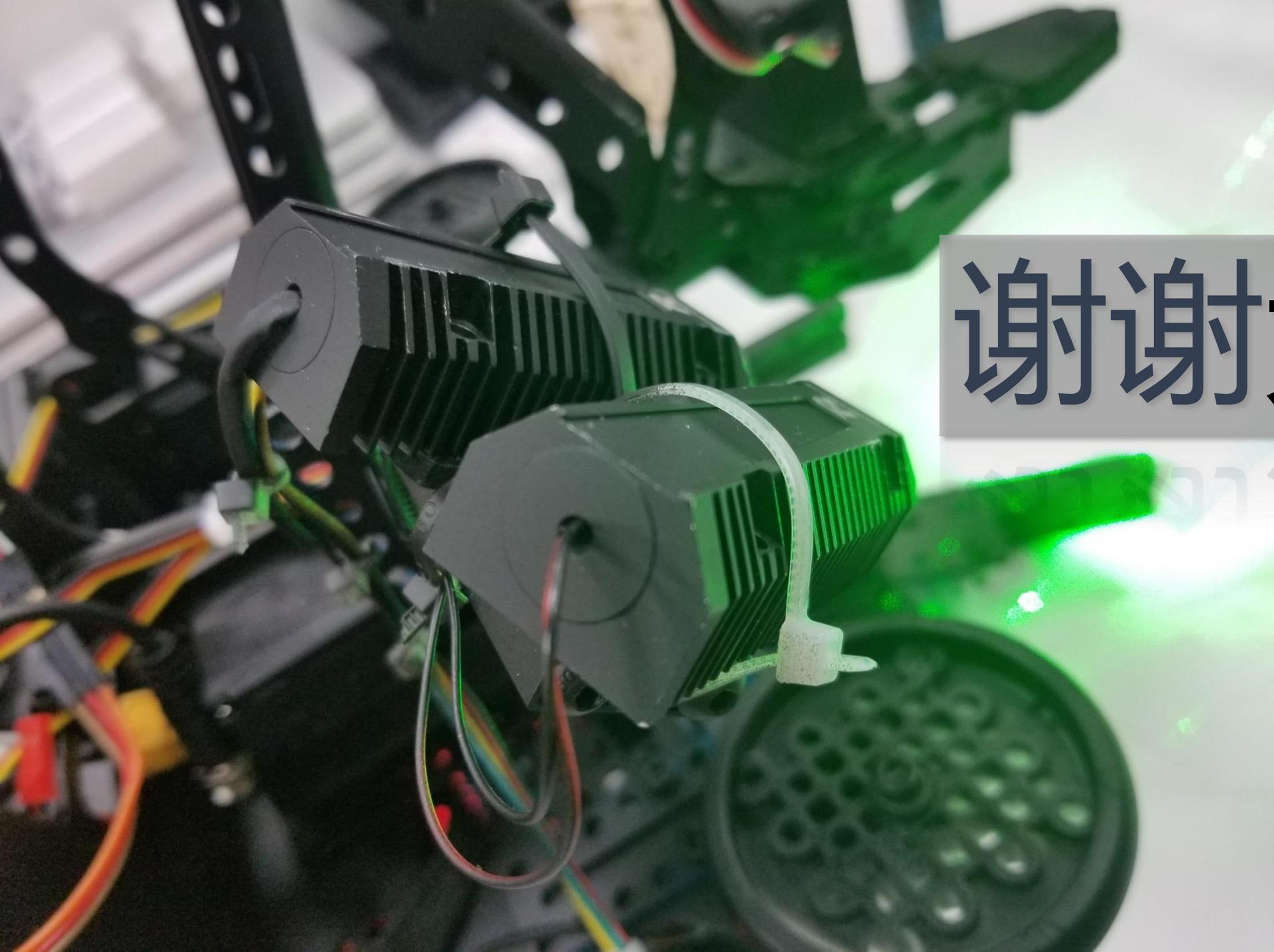




算法

- 需求分析
- 发现问题，提出问题
- 解决问题
- 设计思路
- 总结





谢谢大家