

方案答辩

2019 RM Winter Camp

Group 13

机 械：覃董学 俞沐开 吴泓学
嵌入式：冯小亚 朱彦臻 林 兴
算 法：何伊亮

01 需求分析

核心任务

1 寻线与路线计算

- 沿场地黑线行走保证路径准确
- 旋转门路线优化计算与选择

4 扫描与识别

- 物块放置平台设计
- 如何扫描6个面
- 前进时如何识别物块

识物与避障 2

- 物块识别与对方车辆、墙壁的避让

夹取与放置 3

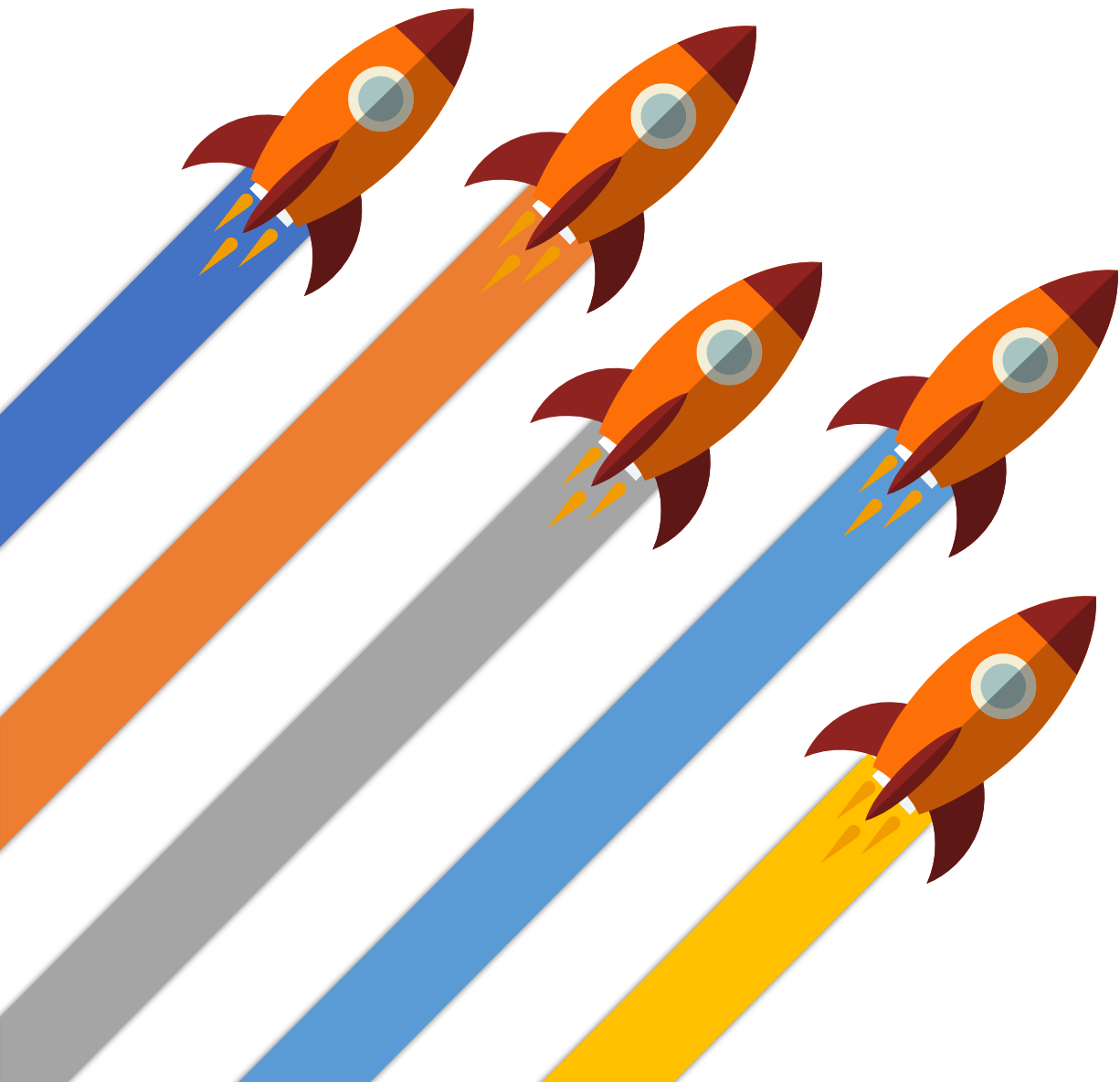
- 物块夹取，储存与放置





02 方案设计

解决策略



01

工兵机器人设计及解密机器人翻块模块设计

02

地图简化，旋转门处理，

03

算法与嵌入式通信交互

04

卡尔曼滤波器，光电寻迹，PID算法

工兵机器人设计

01

考虑官方底盘无RFID放置口，重新设计了机器底盘，以便放置RFID及寻线、图传等各类装置

02

03

01

利用GM3508电机控制机械臂抬升，利用舵机控制夹子开口

02

舵机、限位板做简单储块装置，做坡道使限位板打开时物块可轻易倒出

03

利用3D打印机打印突触块，增大夹取时的摩擦力，防止物块滑落

解密机器人翻块设计

Consideration

如何能让滑块稳定放置；如何实现滑块的稳定翻面；扫完的滑块如何处理

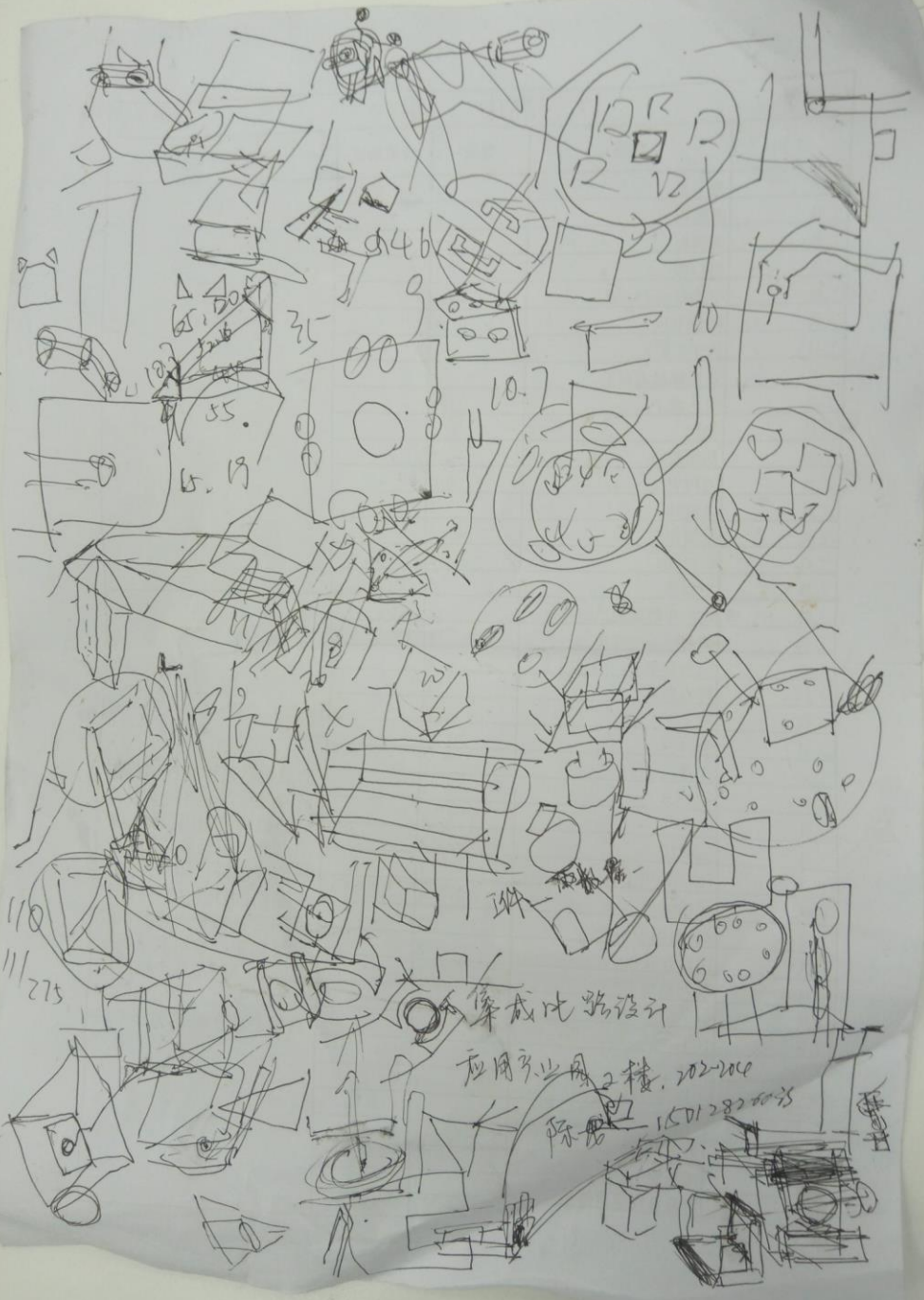
Suggestion

用亚克力板做斜坡完成物块定位；用气动装置完成翻块；参考垃圾车倒垃圾将解密完的物块扔出

Expected Outcome

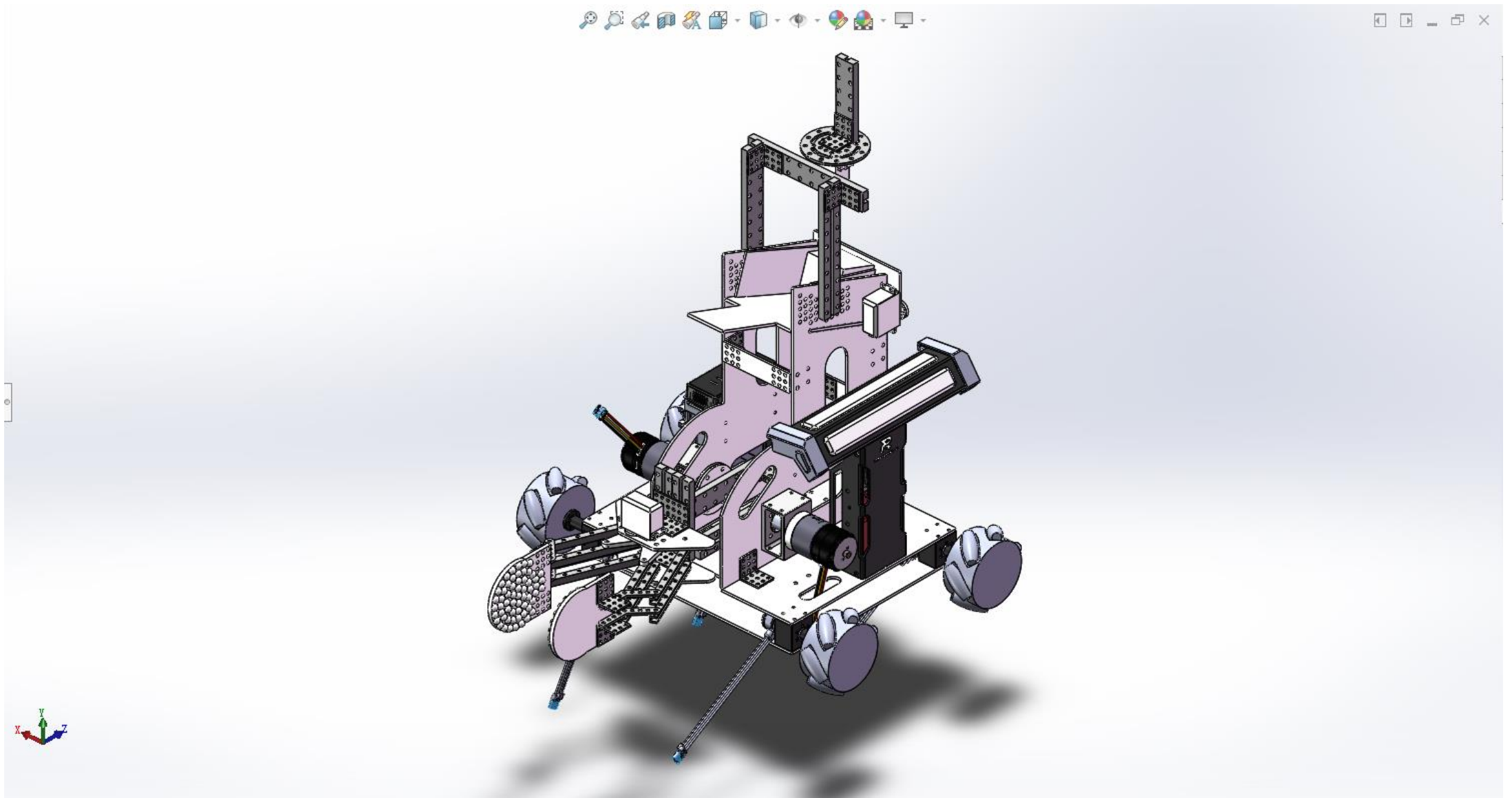
- ✓ 滑块从工兵机器人倒出后准确到达旋转平台
- ✓ 翻块装置保证六个面扫描
- ✓ 丢块动作保不影响下一物块

机械设计

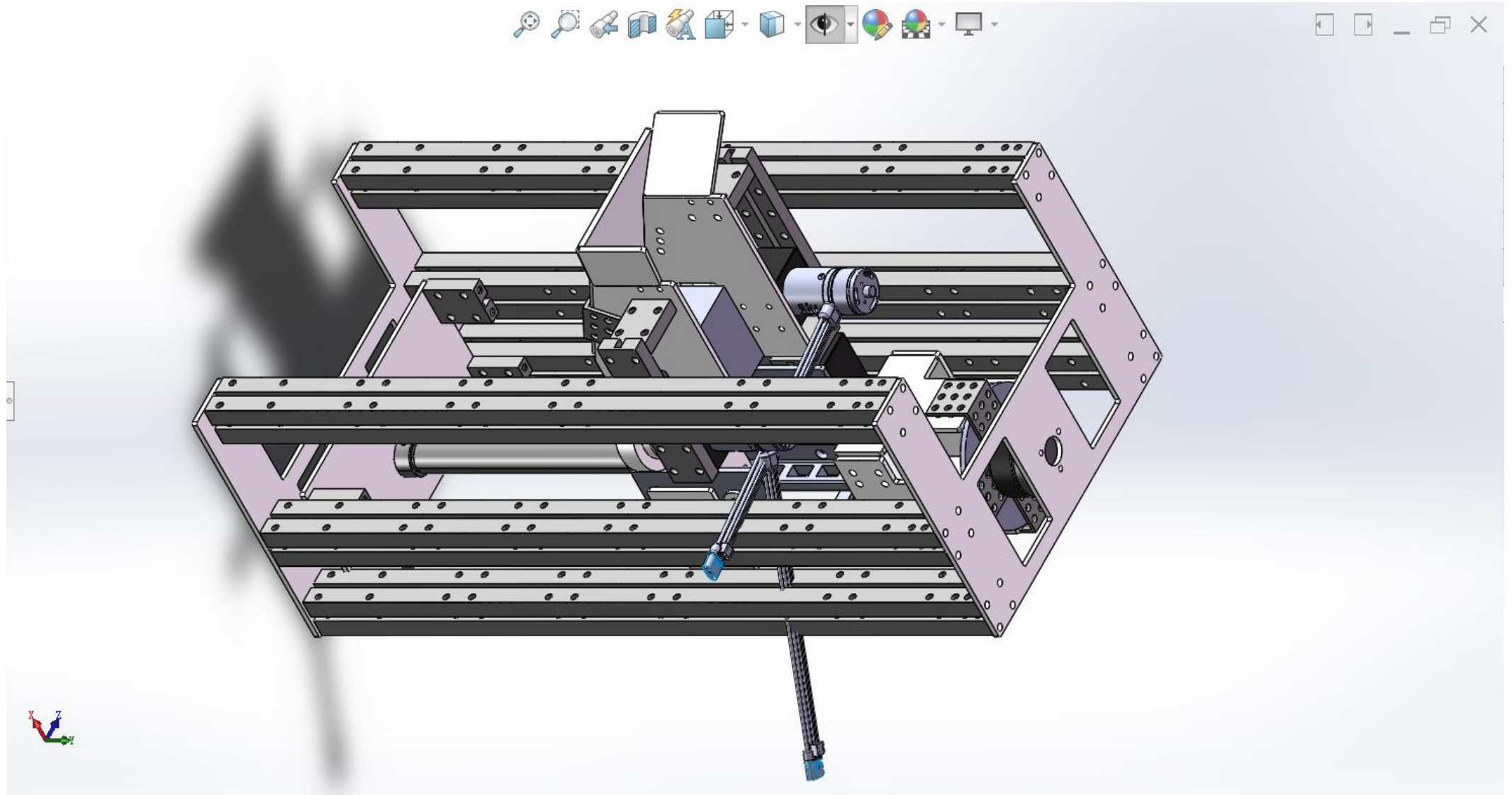


方案讨论

机械设计



机械设计



嵌入式需求

上下位机之间的通信用433无线串口模块，约定了通信协议；
下位机与妙算之间的通信：同样使用了串口通信，使用了USB转串口模块

寻迹：使用八路红外传感器进行寻迹；
返回机器人的当前位置给算法组：通过光电和RFID接收机器人当前位置再通过串口实现下位机与妙算的通信；

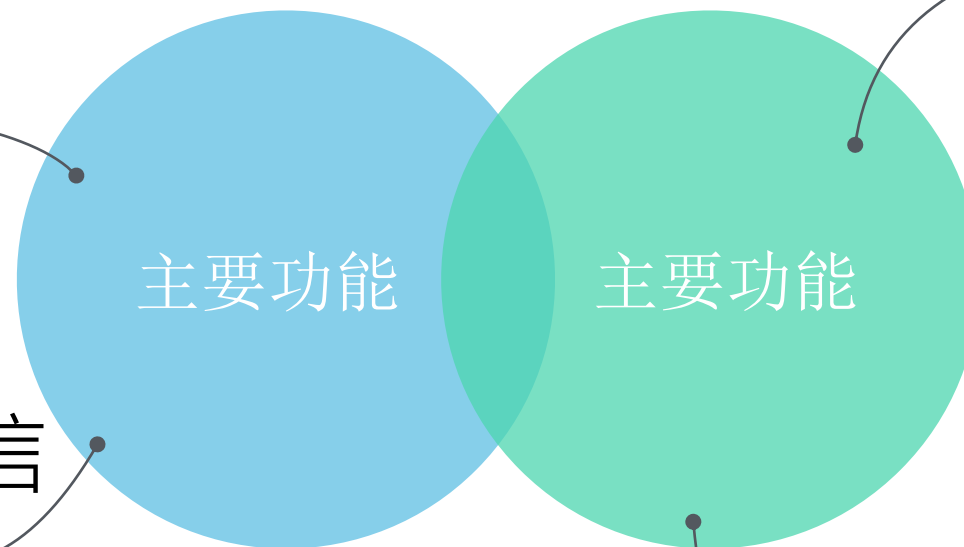
抓块放块：使用了位置环PID调整机械臂位置，使用了PWM波控制舵机进行抓取；实现了一键抓取与放置

嵌入式——通信协议

上位机与主控板的通信

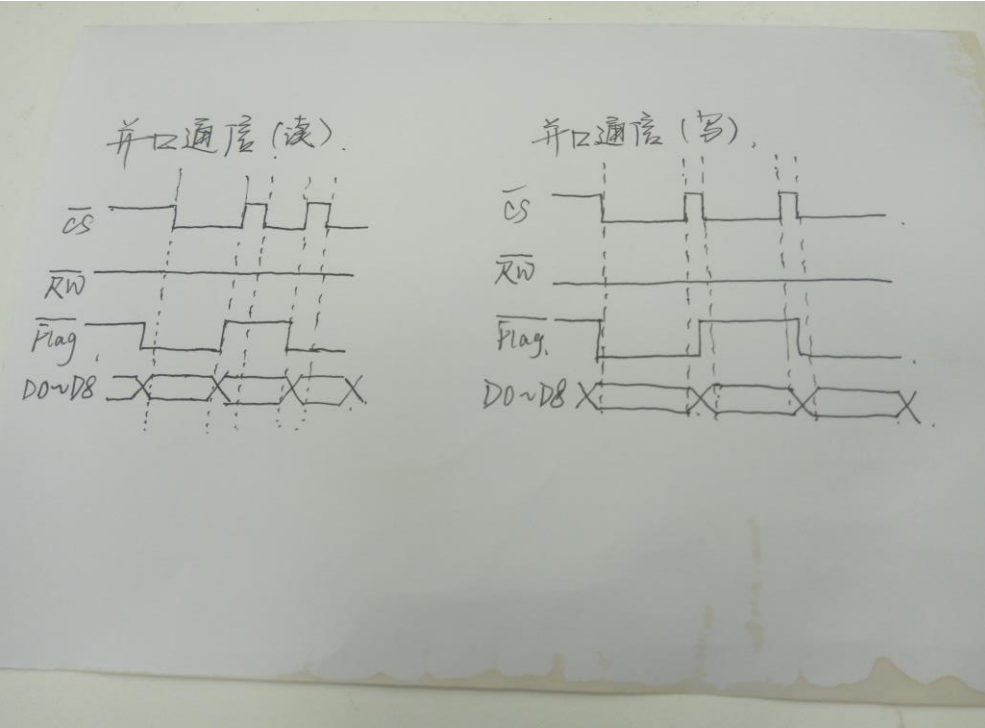
协处理器与主控板的通信

妙算2与主控板的通信



.....

协处理器与主控板的通信（协处理器后来烧掉了，但是我还是想提一下）



采用了“共享内存”的方式，在两个处理器中都开了一个大小为128字节的数组，开启一个进程进行数组内数据的同步。这样一来，各个处理器都只需要操作数组就可以进行数据的交换了。

使用并口通信，共用了11根线，其中8根数据线，1根标志线，1根使能线，1根读写线。

← 时序图

妙算2与主控板的通信

因为妙算2的串口通信要求频率不能太高，所以没有采用协处理器的那种共享内存的方式。这里我们采用了定长数据包的方式进行传输。

需要传输的数据有：坐标(x,y)、摄像头朝向等

帧格式如右

MSB.

第一帧

0	0	Position_X	Posi
---	---	------------	------

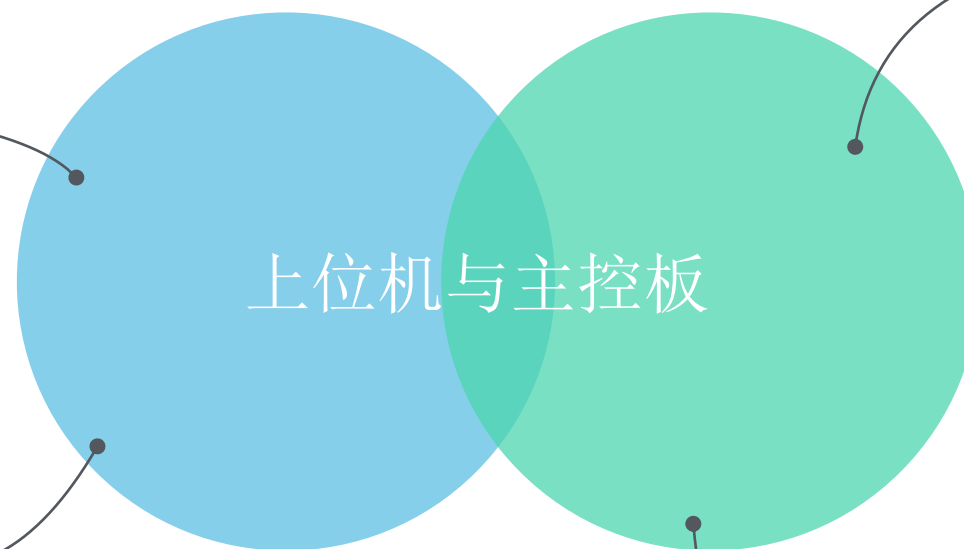
第二帧

0	1	tion_Y	Camera_reserve
---	---	--------	----------------

嵌入式——通信协议

上位机与主控板只需要进行简单的数据交换，所以没有设计特别的协议。上位机的数据直用“abcdefg....”这些单字节的ASCII码实现

下位机回传的数据采用如右格式:



Motor_(Motor1
Speed)_(Motor2
Speed)_(Motor3
Speed)_(Motor4
Speed)_End

Coordinate_(Position_X)_(Position_Y)_End

旋转门的处理

01

时间的估计：把运动的过程简化成以下三类：直走的速度，横走的速度，速度方向切换花费的额外速度。速度以 时间/格子 来表示。

02

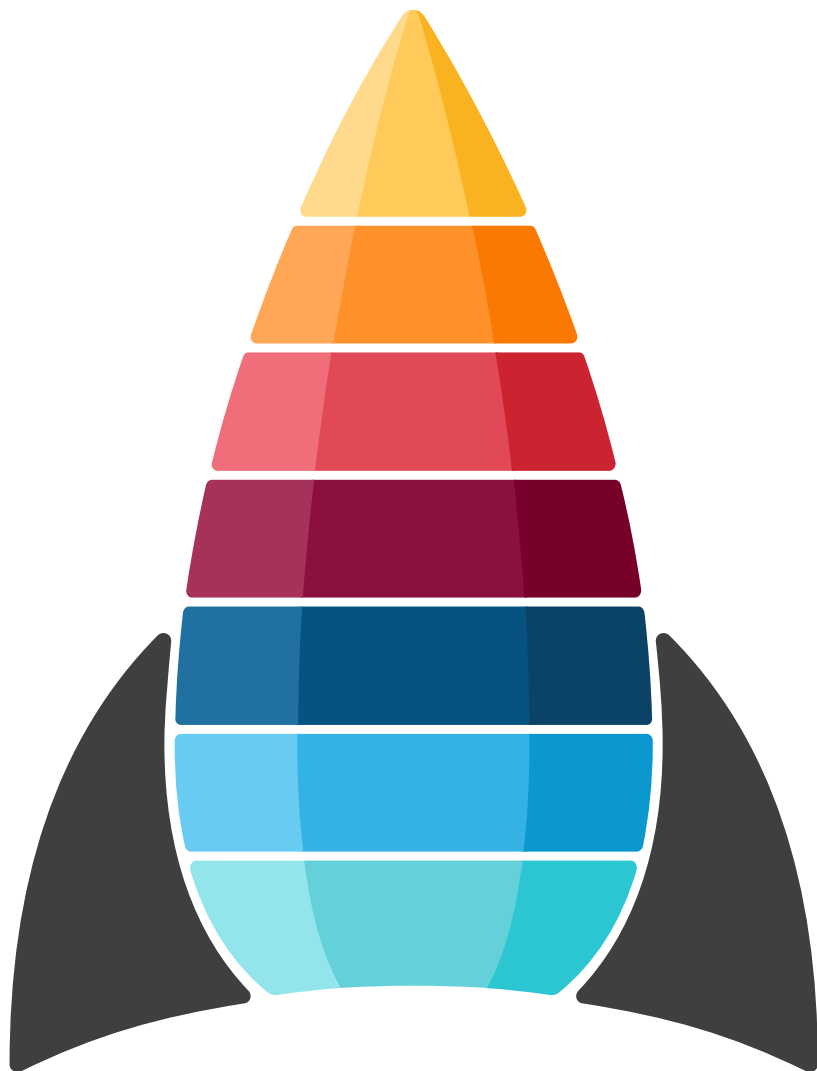
路径规划算法的选择：SPFA。它能方便地处理上述的简化方式和计算出需要在旋转门前等待的时间。

03

规划的方法：把所有旋转门的一种状态作为一张图，在每张图上分别跑一次这样的 SPFA，然后取最优的路径。



第二版规则的变化



割点大量减少，允许可视化地图和全程高速率图传。相撞问题变得可以解决。

图是对称的，方块放置位置也是对称的。需要想好如何利用对称信息。

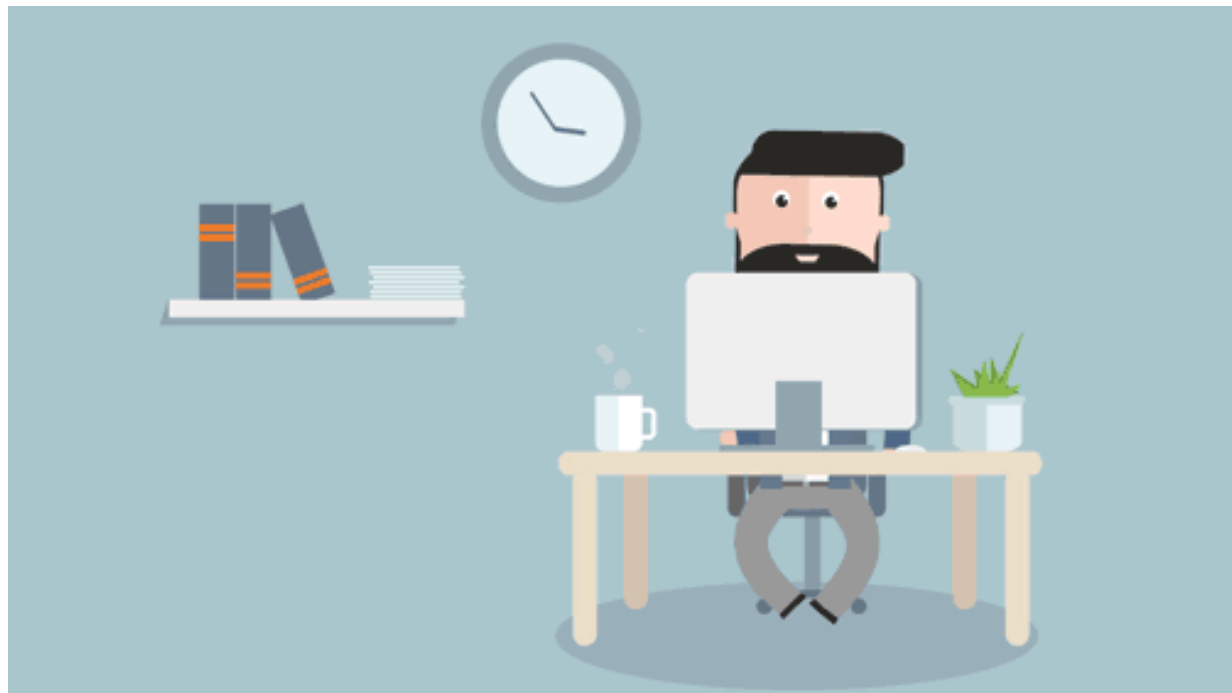
增加了手动操作时间的限制，要尽可能把操作“一键化”。

解决方案

基本策略：

半自动。

事先规划好遍历顺序，算法实时规划路径，根据运行过程中得到的信息更改决策。块的抓取和放置、信息的提供由操作手来完成



解决方案

相撞问题的解决：

将要相撞时，由操作手发送“将要相撞”的信息，然后算法自动将当前点与前方点设置为不连通，5 秒后恢复。

对称信息的利用：

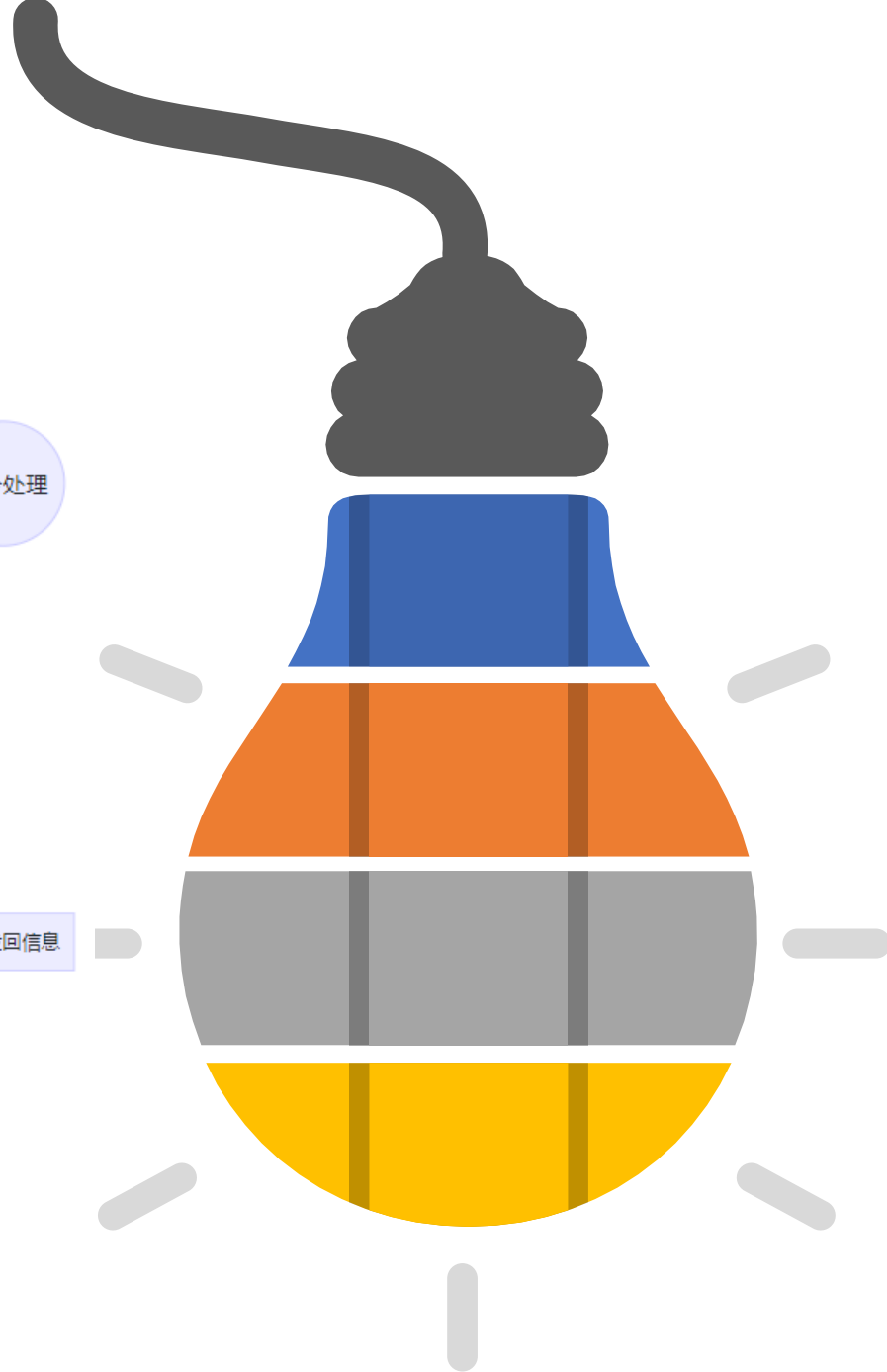
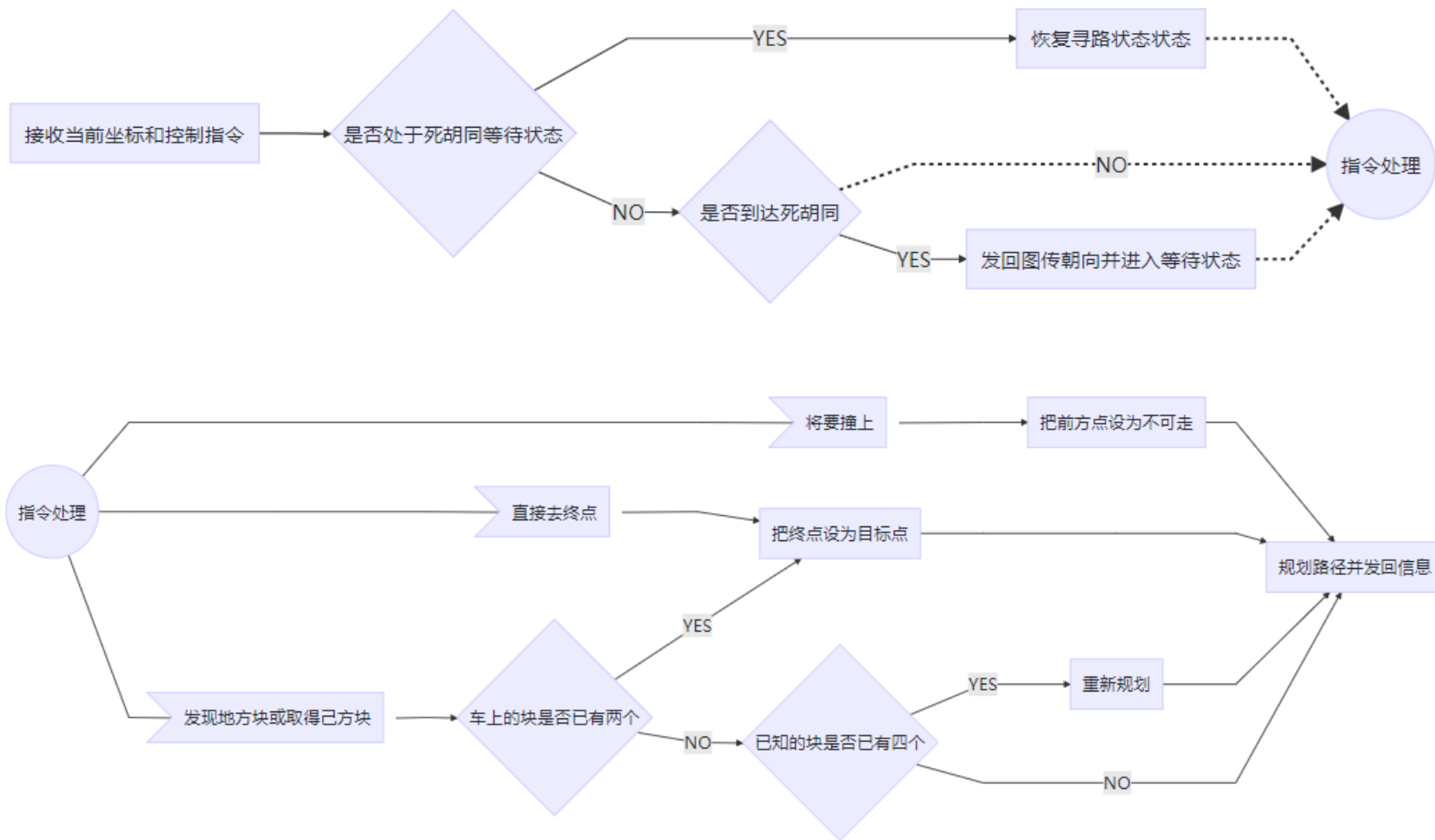
操作手发送“发现敌方方块”和“已抓取己方方块”的信息。当车上已有两个方块时就直接去终点。当所有方块的信息都已知时就直接全排列规划剩余方块的最优遍历路径和顺序。



减少手动操作的时间：

这里有嵌入式的功劳，如方块抓取的一键化。发送信息也是一键。在到达可能存在方块的点时会等待 3 秒给操作手反应。此外我还提供了“一键去终点”这种指令。

程序框图



1

**INTRO-
DUCTION**

2

FLOW

3

OBJECTIVES

4

BENEFITS

5

BARRIERS

Q

&

A