**《机器人编程实践》学生实验报告**

**学院** 数计学院 **专业** 计算机科学与技术 **班级** 计科3班

**姓名** 周吉瑞 **学号** 20190521340 **日期** 2021/05/04

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称：** | 机器人编程实践 |  |  |
| **实验名称：** | 实验五 移动射击 | | |
| **指导老师：** | 孙建勇 |  |  |

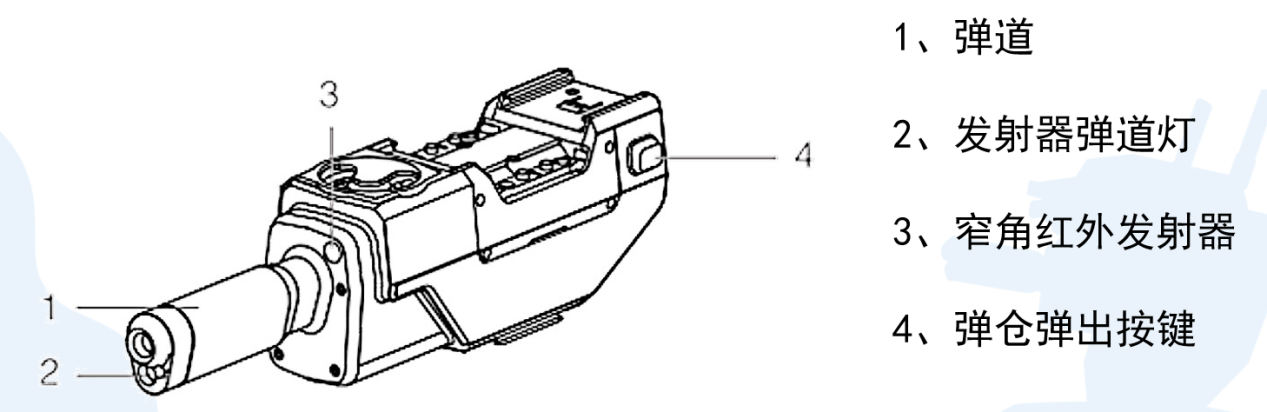
**目录**

1. **实验记录及总结**
2. **理论学习与总结**
3. **实践任务与设计**
4. **方案实践与记录**
5. **拓展任务**
6. **实践任务与设计（1）**
7. **方案实践与记录（1）**
8. **实践任务与设计（2）**
9. **方案实践与记录（2）**

## 实验记录及总结

1. **理论学习与总结**
2. **硬件学习**

发射器结构：

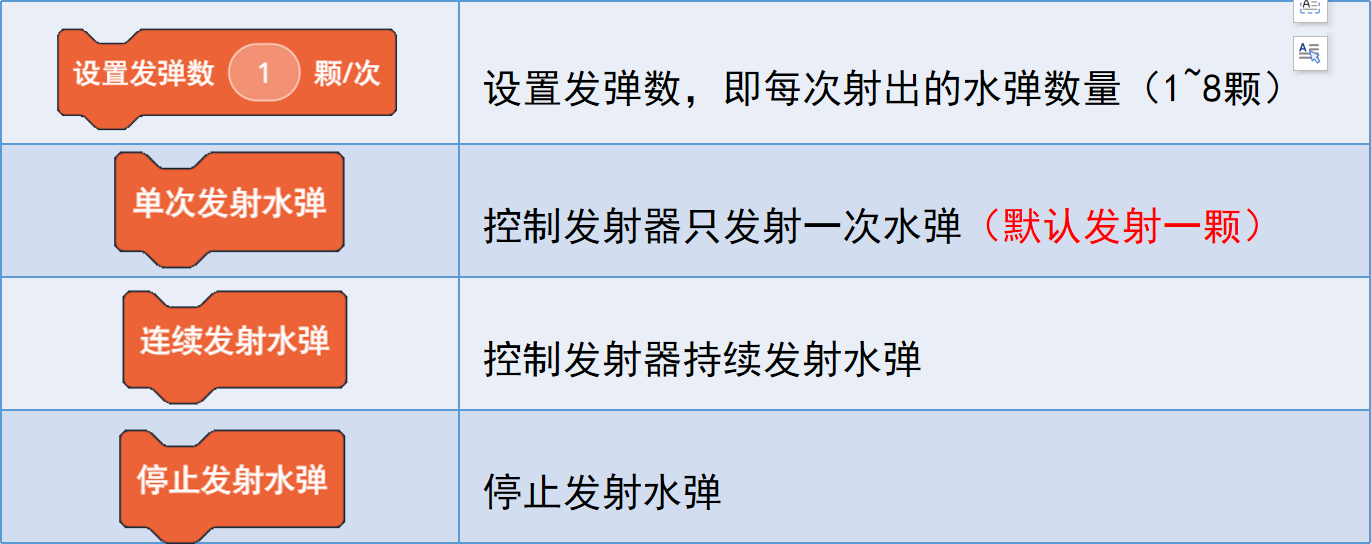


发射器原理：

气动是利用撞击作用或转动作用产生的空气压力使其运动或作功，气动就是以压缩空气为动力源，带动机械完成伸缩或旋转动作。因为是利用空气具有压缩性的特点，吸入空气压缩储存，空气便像弹簧一样具有了弹力，然后用控制元件控制其方向，带动执行元件的旋转与伸缩。

1. **编程模块**

发射器模块——水晶弹



发射器模块——红外光束

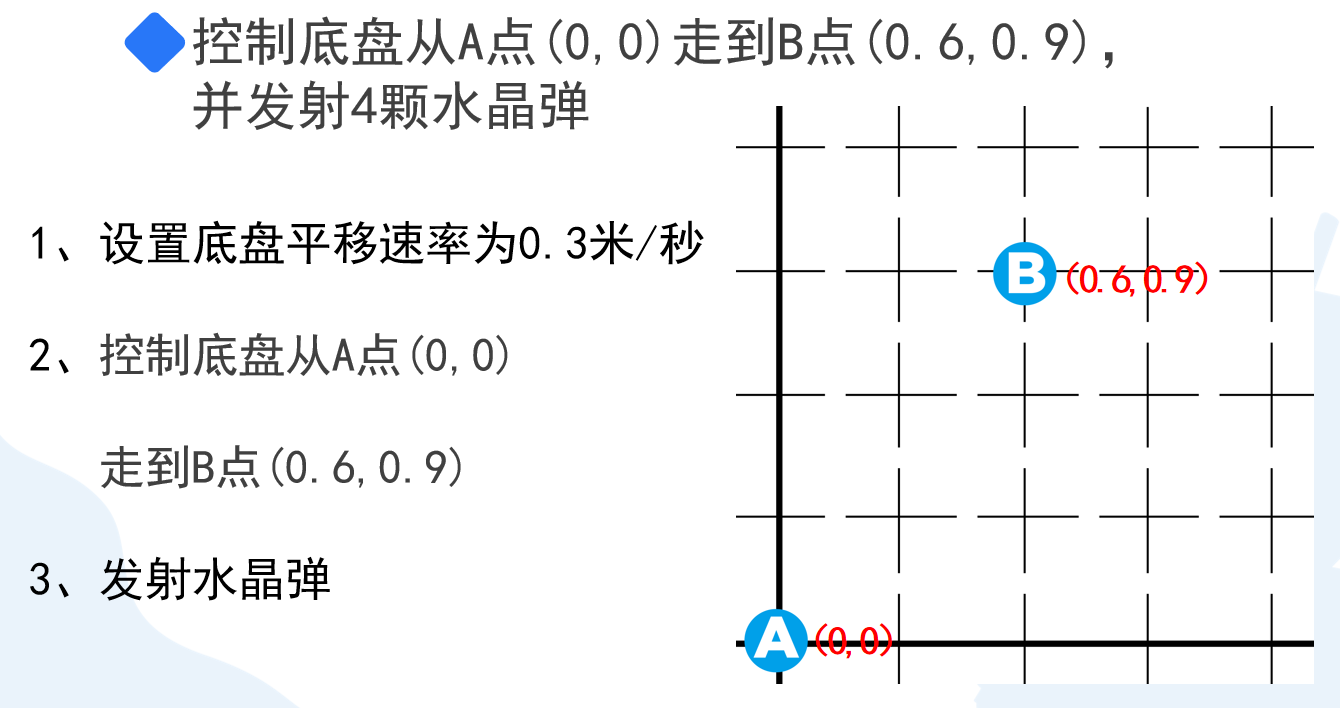


底盘模块

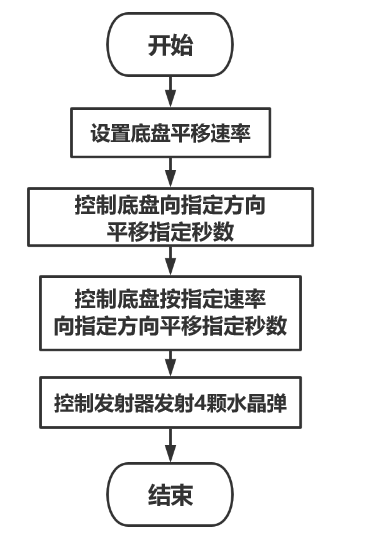


**2、实践任务与设计**

**（1）任务**



**（2）方案设计**



【方案1】

- 设置底盘平移速率

- 控制底盘平移

- 设置每次发弹数量

- 单次发射水弹

【方案2】

- 设置底盘平移速率

- 控制底盘平移

- 重复单次反射水弹

1. **方案实践与记录**

【方案1】



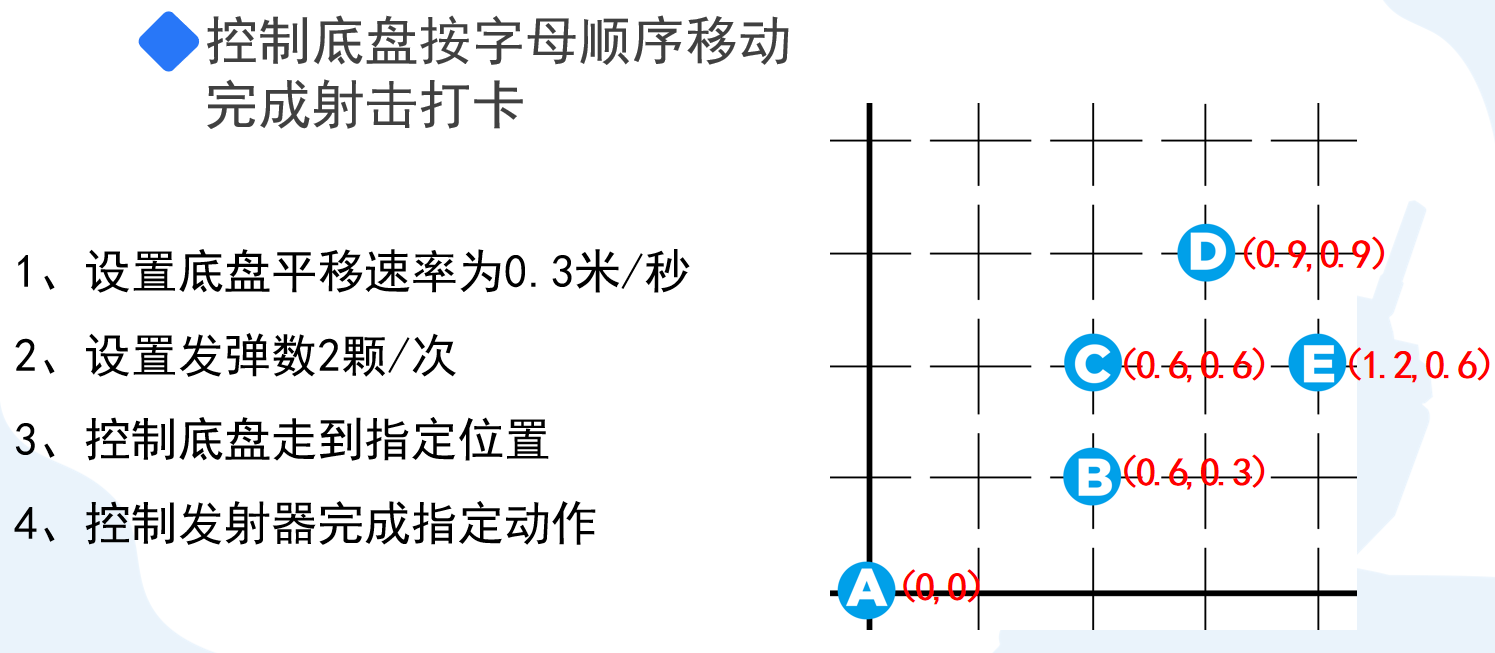
【方案2】



## 二、拓展任务

**1、实践任务与设计（1）**

1. **任务**



1. **方案设计**

【理论分析】

如何才能从一个点移动到另一个点呢？

设：起点为（x，y）终点为（a，b）

那么从起点移动到终点的步骤应该为：

<1>、x轴方向上移动a-x个长度单位（a-x为负数时为向左移动，否则为向右移动）

<2>、y轴方向上移动b-y个长度单位（b-y为负数时为向下移动，否则为向上移动）

【伪代码】

// 伪代码（C语言风格）

main() {

    setBullet(2);   // 设置每次的发弹数

    setSpeed(0.3);  // 设置底盘平移速率

*double* x **=** 0, y **=** 0;           // 坐标点

*double* temp\_x **=** 0, temp\_y **=** 0; // 零时保存坐标点

*double* a **=** 0, b **=** 0;           // 上一个坐标点

    // 每次输入一个坐标

**while** (scanf("%lf%lf", **&**x, **&**y)) {

        temp\_x **=** x;     // 用零时变量将坐标初始值占时保存

        temp\_y **=** y;     // 用零时变量将坐标初始值占时保存

        x **=** x **-** a;      // 去往该点的x轴移动单位=此点x坐标-上一个点x坐标

        y **=** y **-** b;      // 去往该点的y轴移动单位=此点y坐标-上一个点y坐标

        // 控制底盘在x轴上的移动

**if** (x **>=** 0) {

            move(90, abs(x));   // 如果x为正，那么底盘向90度方向移动绝对值x个长度单位

        } **else** {

            move(**-**90, abs(x));  // 如果x为负，那么底盘向-90度方向移动绝对值x个长度单位

        }

    // 控制底盘在y轴上的移动

**if** (y **>=** 0) {

            move(0, abs(y));    // 如果y为正，那么底盘向0度方向移动绝对值y个长度单位

        } **else** {

            move(**-**180, abs(y)); // 如果y为负，那么底盘向-180度方向移动绝对值y个长度单位

        }

a **=** temp\_x;     // 保存该点的x值

        b **=** temp\_y;     // 保存该点的y值

        launchOnce();   // 单次发射水弹

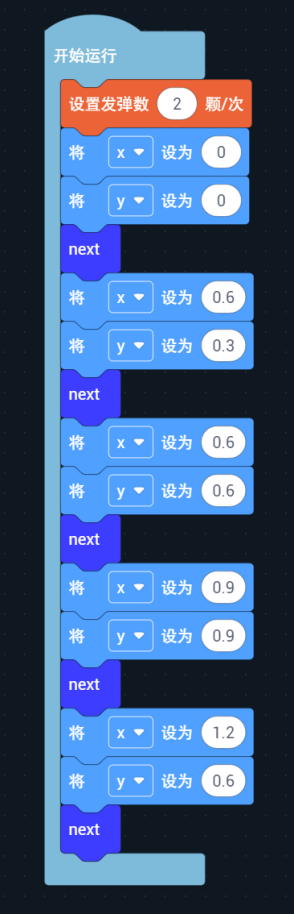
        wait(3);        // 等待3秒

    } // 一个坐标的移动与射击完成，进入下一个坐标

}

**2、方案实践与记录（1）**

【开始运行模块】 【函数next模块】 【函数temp1模块】



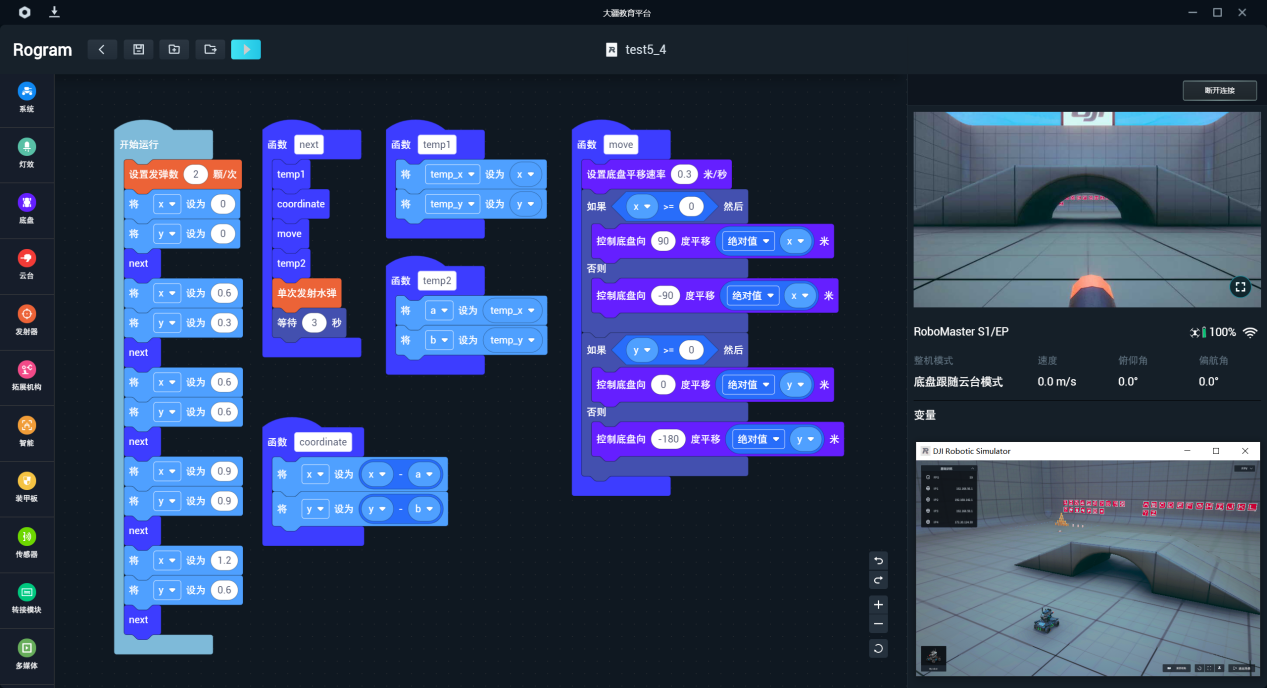
【函数temp2模块】 【函数coordinate模块】



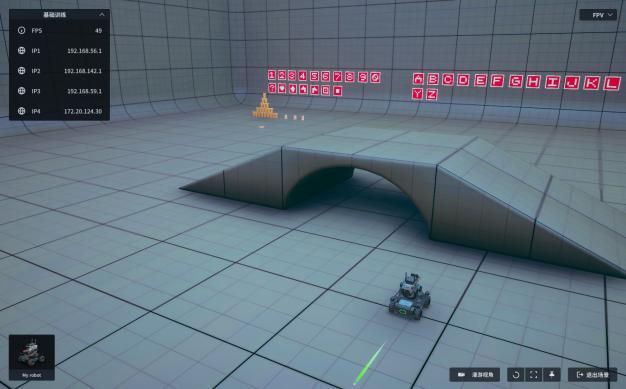
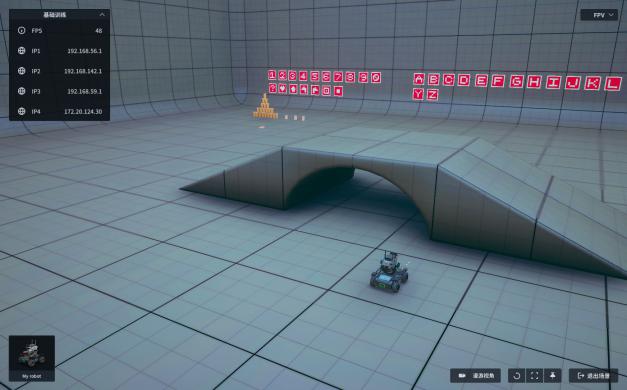
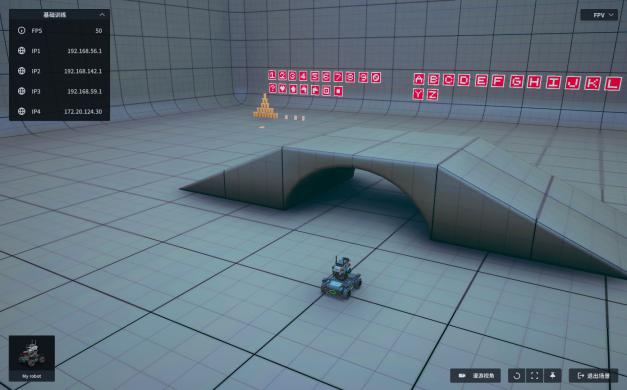
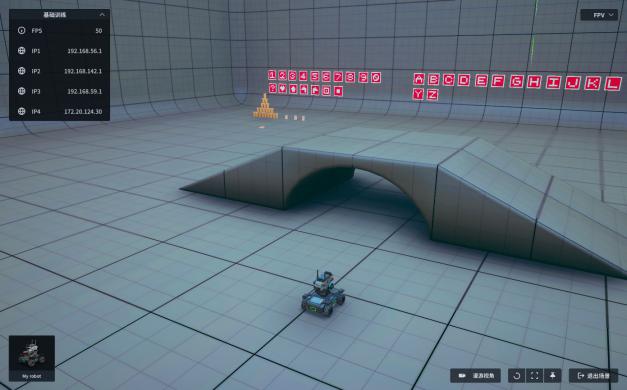
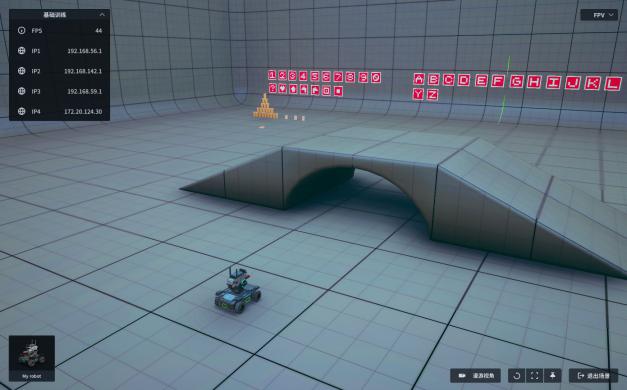
【函数move模块】



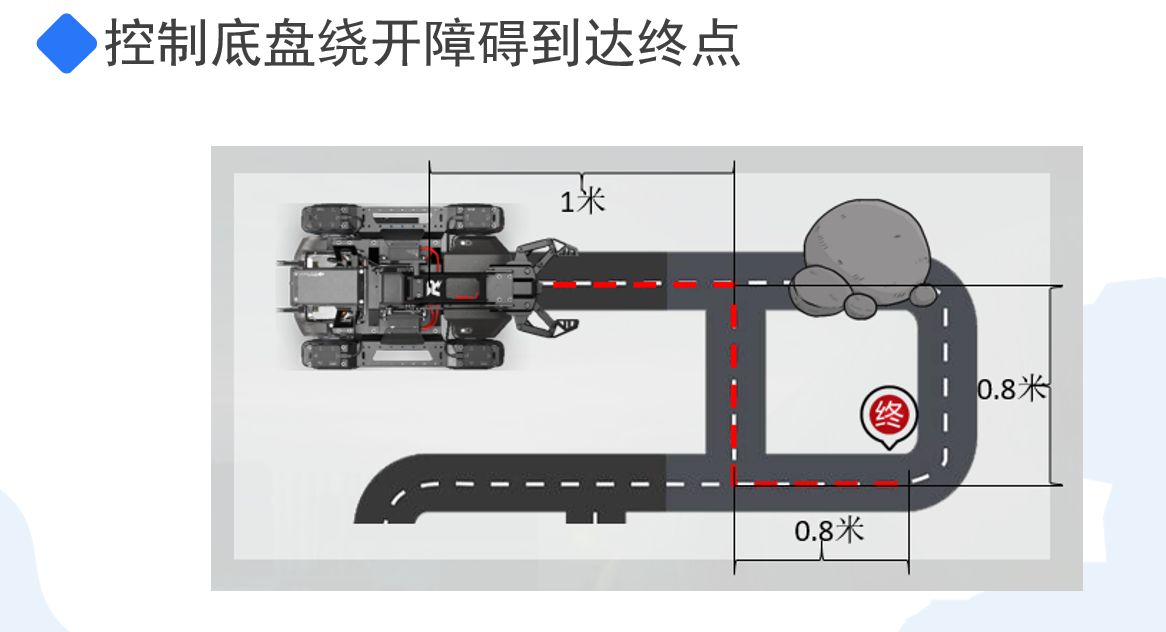
【程序总览】



【运行片段截图】



1. **实践任务与设计（2）**



**4、方案实践与记录（2）**

