

《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

一、实验记录及总结

1. 理论学习与总结

2. 实践任务与设计

3. 方案实践与记录

二、拓展任务

1. 实践任务与设计（1）

2. 方案实践与记录（1）

3. 实践任务与设计（2）

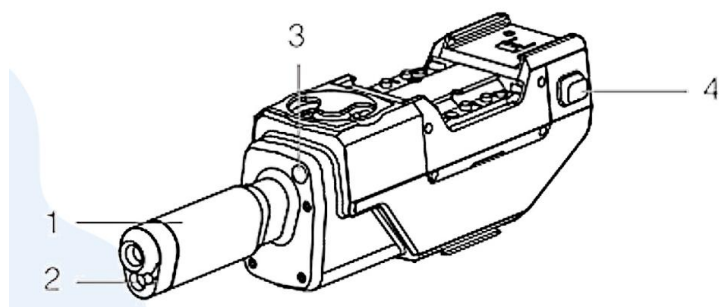
4. 方案实践与记录（2）

一、实验记录及总结

1、理论学习与总结

（1）硬件学习

发射器结构：



1、弹道

2、发射器弹道灯

3、窄角红外发射器

4、弹仓弹出按键

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。

(2) 编程模块

发射器模块——水晶弹

设置发弹数 1 颗/次	设置发弹数，即每次射出的水弹数量（1~8颗）
单次发射水弹	控制发射器只发射一次水弹（默认发射一颗）
连续发射水弹	控制发射器持续发射水弹
停止发射水弹	停止发射水弹

发射器模块——红外光束

设置红外发射频率 1 次/秒	设置红外光束的发射频率
单次发射红外光束	控制发射器只发射一次红外光束
连续发射红外光束	控制发射器持续发射红外光束
停止发射红外光束	停止发射红外光束

底盘模块

设置底盘平移速率 0.5 米/秒

控制底盘向 0 度平移

控制底盘以 0.5 米/秒向 0 度平移





控制底盘以指定的平移速率向指定方向平移

2、实践任务与设计

(1) 任务

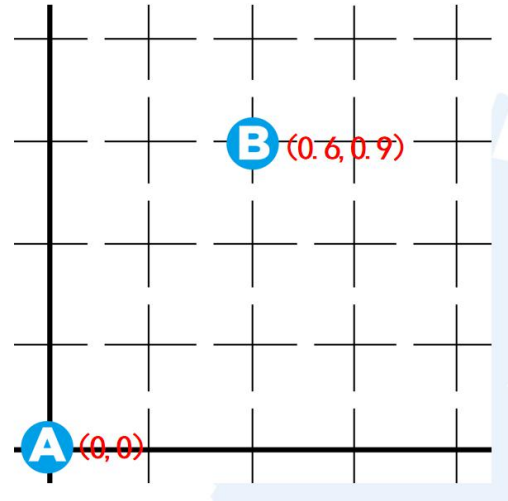
◆ 控制底盘从A点(0, 0)走到B点(0.6, 0.9),
并发射4颗水晶弹

1、设置底盘平移速率为0.3米/秒

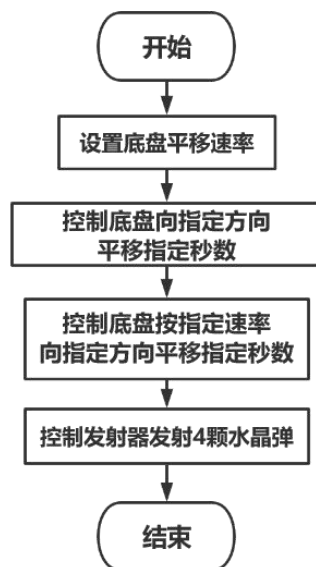
2、控制底盘从A点(0, 0)

走到B点(0.6, 0.9)

3、发射水晶弹



(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】

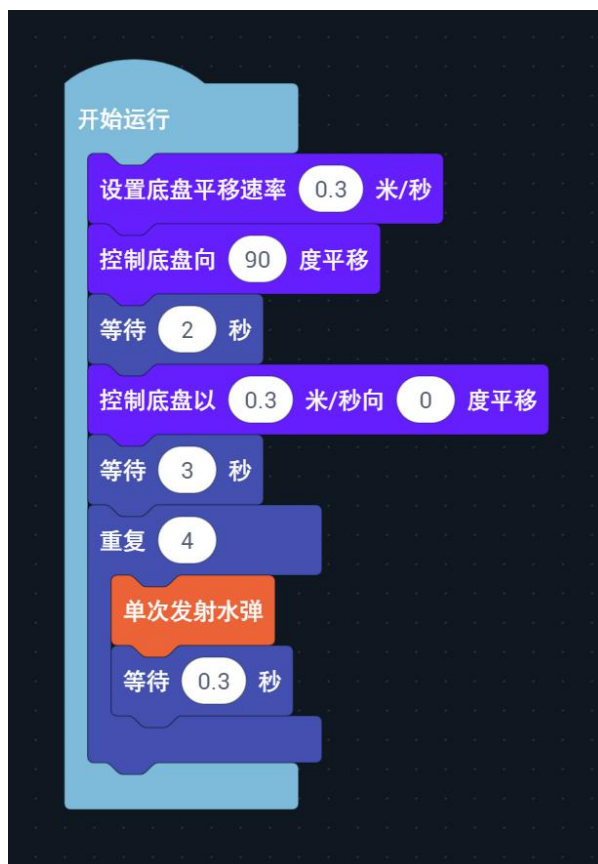
- 设置底盘平移速率
- 控制底盘平移
- 重复单次反射水弹

2、方案实践与记录

【方案 1】



【方案 2】



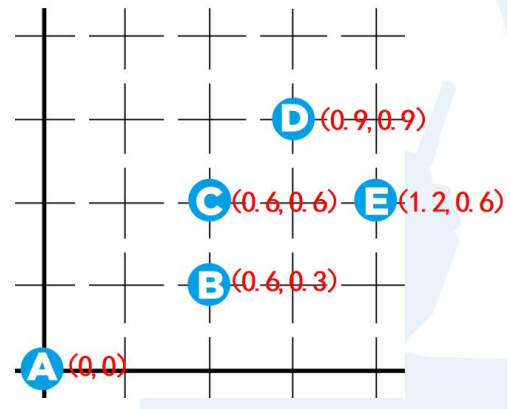
二、拓展任务

1、实践任务与设计（1）

（1）任务

◆ 控制底盘按字母顺序移动 完成射击打卡

- 1、设置底盘平移速率为0.3米/秒
- 2、设置发弹数2颗/次
- 3、控制底盘走到指定位置
- 4、控制发射器完成指定动作



（2）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暂时保存
        temp_y = y;              // 用零时变量将坐标初始值暂时保存

        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】



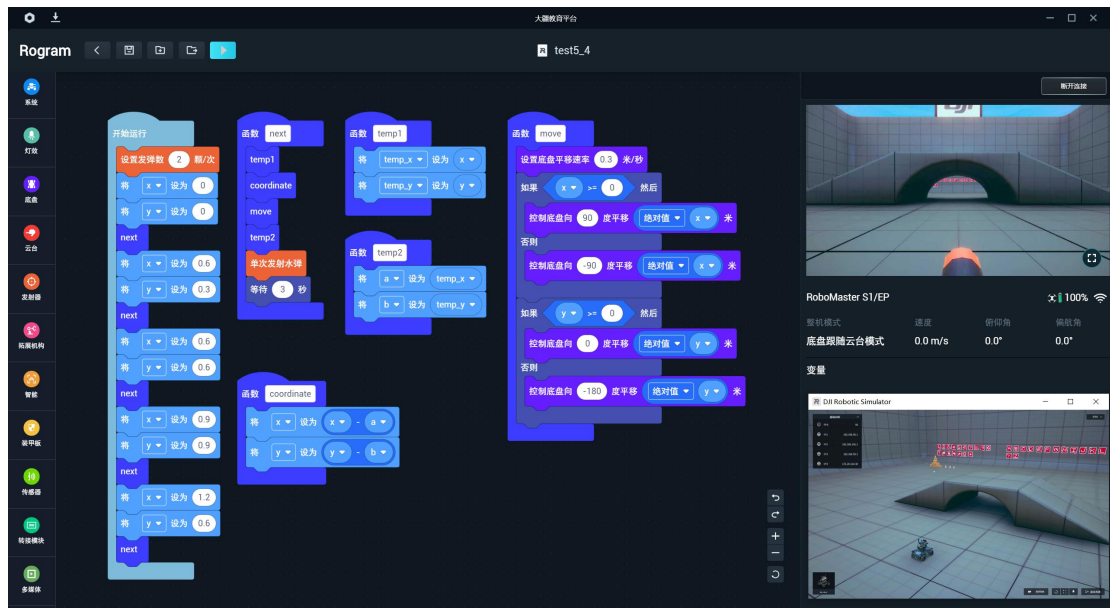
【函数 coordinate 模块】



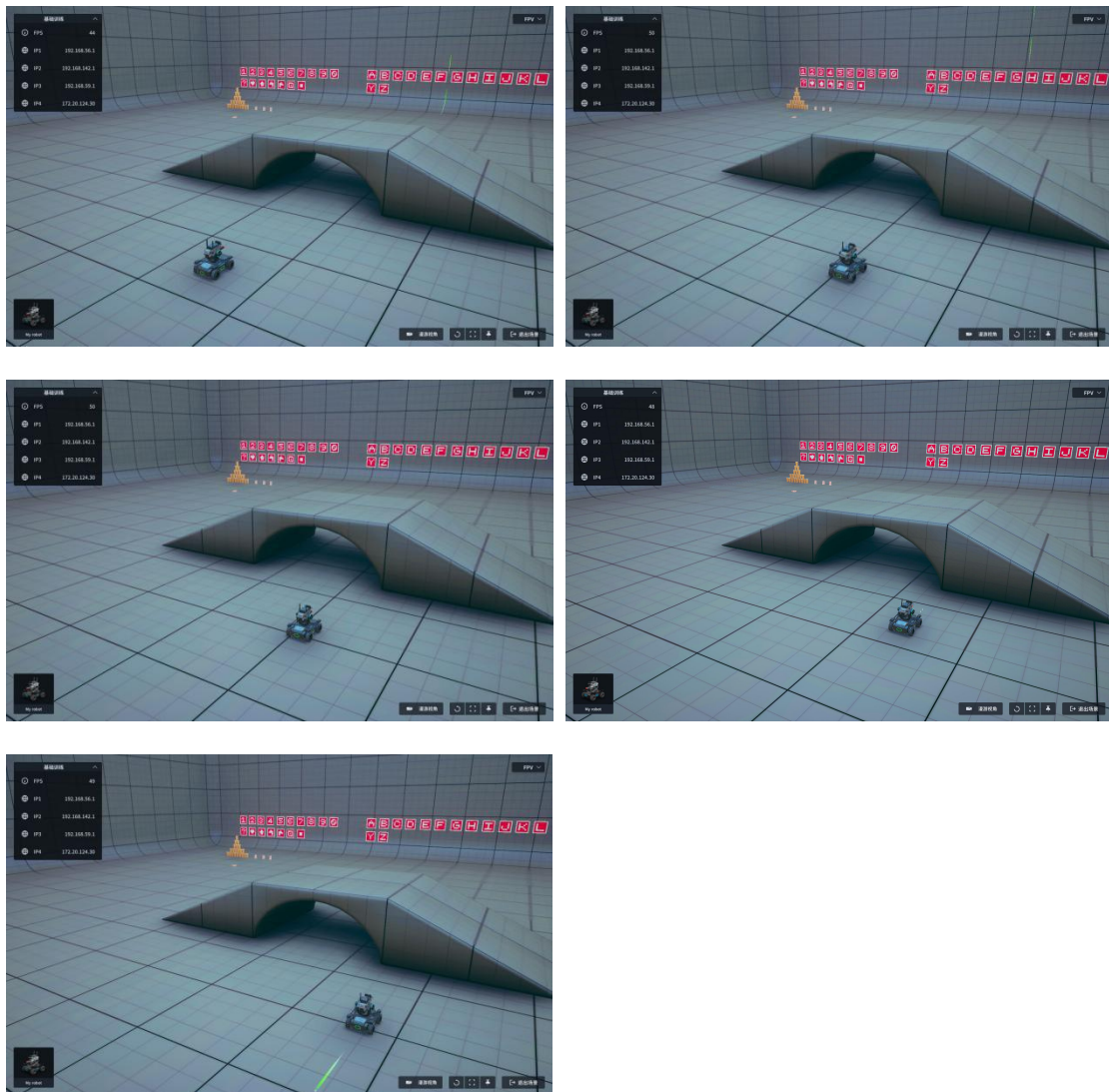
【函数 move 模块】



【程序总览】

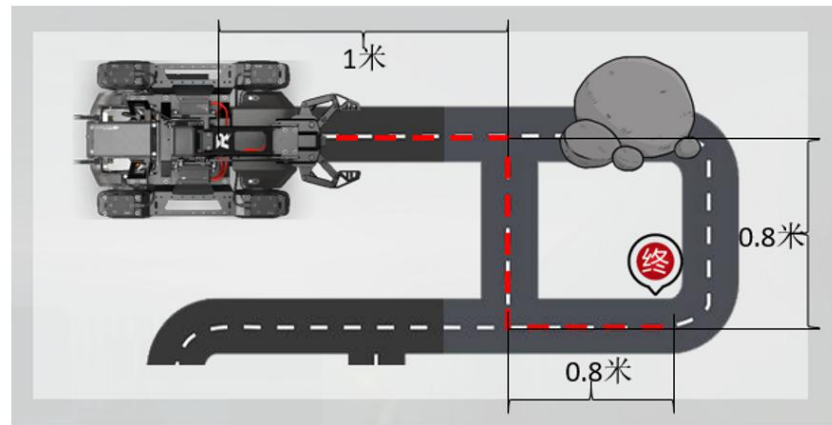


【运行片段截图】



3、实践任务与设计（2）

◆ 控制底盘绕开障碍到达终点



4、方案实践与记录（2）

