

前 言.....	4
<b>第 1 章 预备知识.....</b>	<b>5</b>
1.1 虚拟机器人的结构与功能 .....	6
1.1.1 身体结构 .....	6
1.1.2 感觉器官 .....	6
1.1.3 编程语言 .....	8
1.2 VJC1.5 仿真版能做什么 .....	9
1.2.1 为机器人编写程序 .....	9
1.2.2 创建环境 .....	9
1.2.3 仿真运行 .....	9
1.3 光盘的安装与启动 .....	10
1.3.1 安装 .....	10
1.3.2 启动 .....	10
1.4 如何使用“帮助”电子文档 .....	11
<b>第 2 章 初试身手.....</b>	<b>14</b>
2.1 歌曲 .....	15
2.2 边唱边跳 .....	18
2.3 走正方形 .....	19
2.4 走六边形 .....	19
2.5 走五角星 .....	20
2.6 走圆形 .....	24
2.7 奥运五环 .....	25
2.8 进门比赛 .....	26
2.9 一笔画 .....	27
2.10 求和 .....	27
2.11 四则混合运算 .....	29
2.12 整型数与浮点数 .....	29
2.13 霓虹灯 .....	30
2.14 汽车 .....	31
2.15 迎宾机器人 .....	31
2.16 回廊 .....	31
2.17 动态字符 .....	32
2.18 波浪 .....	33
2.19 落地扇 .....	33
2.20 等比求和 .....	34
<b>第 3 章 稳步前进.....</b>	<b>35</b>

3.1	打招呼 .....	36
3.2	叫人起床 .....	42
3.3	彩色地毯 .....	45
3.4	百米赛跑 .....	45
3.5	计算走过的距离 .....	46
3.6	秒表 .....	49
3.7	风扇定时 .....	49
3.8	找房门 .....	51
3.9	醉汉 .....	55
3.10	自由行走 .....	56
3.11	趋光的飞蛾 .....	57
3.12	鼯鼠 .....	59
3.13	沿线走 .....	60
3.14	越障跑 .....	64
3.15	太阳能电池 .....	64
3.16	电子琴 .....	65
3.17	穿越隧道 .....	65
3.18	舞台 .....	66
3.19	哈雷彗星 .....	66
<b>第4章 挑战自我 .....</b>		<b>67</b>
4.1	填字游戏 .....	68
4.2	台球 .....	70
4.3	寻宝 .....	73
4.4	篝火晚会 .....	74
4.5	弹球游戏 .....	77
4.6	随机走 .....	78
4.7	沿墙走 .....	79
4.8	点歌台 .....	82
4.9	回音壁 .....	82
4.10	查找最大值 .....	83
4.11	排序 .....	84
4.12	最大公约数 .....	85
4.13	跳房子游戏 .....	89
4.14	找地雷 .....	90
<b>第5章 我能赢 .....</b>		<b>91</b>

5.1	走迷宫 .....	92
5.2	游北京 .....	93
5.3	长跑比赛 .....	94
5.4	救援 .....	95
5.5	灭火比赛 .....	96
<b>附 录</b>	<b>.....</b>	<b>99</b>
附录 A.	子 程 序 .....	99
附录 B.	如何设置机器人型号 .....	103
附录 C.	不同型号的虚拟机器人碰撞传感器差异 .....	104
附录 D.	真实的能力风暴机器人功能一览表 .....	104

# 前 言

本书介绍了机器人的初步知识，包括机器人的身体结构、感觉系统和思维方式。与一般教科书不同的是：所有这些知识是在一系列的机器人项目中逐步展开的。读者在做机器人项目的过程中，将自主建构机器人的知识体系。建构机器人知识体系的过程是新颖有趣的，给人的印象也会是深刻的。

本书还详细讲解了 VJC 编程语言。VJC 编程语言又称为机器人 C 语言，是一种易学易用的编程语言。读完该书，再去学标准的 C 语言程序设计，就易如反掌了。

近年来，建构主义的教育理念得到了广泛的认可。大家知道，要在学习知识的过程中应用建构主义的方法，是很需要一番努力的，因为用于建构的素材和情境并非总是那么唾手可得的。本书介绍的智能机器人，由于它的技术先进性、学科综合性、自主体验性、趣味性和不断更新性，可以称得上是实现建构主义教育理念的一个优秀平台。通过做项目来建构机器人的知识体系，可以抛开繁琐的理论书籍，在学中用，又在用中学，一切都将显得那么轻松和自然。

本书的读者可以是中小学校的青少年，也可以是任何年龄的机器人爱好者。中小学校的科技老师也可以用它做教材。

# 第1章 预备知识

在本章中，我们对能力风暴智能机器人作一个简单介绍，让读者初步了解机器人的结构、传感器和编程语言。本章还讲述了光盘的安装使用方法，并给出了 VJC1.5 仿真版软件中“帮助”电子文档的详细目录。在读完本章内容之后，相信读者就能方便地使用 VJC1.5 仿真版了。



## 1.1 虚拟机器人的结构与功能

### 1.1.1 身体结构

在 VJC1.5 中，有五种型号的机器人：AS-M、AS-MII、AS-UII、AS-InfoX 和 AS-InfoM。图 1 - 1 是虚拟机器人的外形。虚拟机器人的身体结构跟真实的能力风暴智能机器人是相似的。本书中的范例采用的机器人型号均为 AS-M。设置机器人型号的方法参见附录 B。



图1 - 1 虚拟机器人外形

### 1.1.2 感觉器官

能力风暴智能机器人的感觉功能，是由机器人的传感器实现的。能力风暴机器人身上安装的传感器有以下几种：

#### 1. 红外传感器

红外传感器由红外发射器和红外接收器两部分组成。一旦程序中发出“红外测障”的指令，红外发射器就开始发射红外线。红外线遇到障碍物会反射回来，被红外接收器接收，从而机器人就能判断出障碍物所在的方位。虚拟机器人能够检测左、右、前三个方向的障碍物，跟真实的能力风暴智能机器人基本相同。发射红外线的虚拟机器人如图 1 - 2 所示。



图1 - 2 机器人发射红外线示意图

## 2. 光敏传感器

光敏传感器能够检测光线的强度，检测到的值为 0 ~ 255 之间的整数。光线越亮，检测到的数值越小；光线越暗，检测到的数值越大。此项功能跟真实的能力风暴智能机器人是一样的

## 3. 碰撞传感器

虚拟机器人的碰撞传感器能够检测左、右、前、后四个方向的碰撞，型号为 AS-InfoX 的机器人只能检测前、左、右三个方向的碰撞。如前所述，虚拟机器人共分五种，它们的区别主要就体现在碰撞检测功能上。所以读者在打开程序时，无论是例程，还是自己编写保存的程序，如果涉及到“碰撞检测”，必须先设置相应的机器人型号（参见附录 B），然后再打开程序、仿真运行。例如：打开例程“自由行走(M).flw”时，由于名称后面带有 M，故须先将机器人型号设置为 AS-M，然后再打开程序文件、仿真运行。打开例程“自由行走(Info).flw”时，最好先将机器人型号设置为 AS-InfoM，然后再打开程序文件、仿真运行。

在碰撞检测功能方面，虚拟机器人跟真实的能力风暴智能机器人有较大差别（参见附录 C、D）。

## 4. 声音传感器

声音传感器能够检测声强大小，检测到的值为 0 ~ 255 之间的整数。数值越小，声音越低；数值越大，声音越响。此项功能跟真实的能力风暴智能机器人是一样的。

## 5. 地面灰度传感器

地面灰度传感器能够检测地面的灰度，检测到的值为 0 ~ 255 之间的整数。地面颜色越深，灰度值越大；地面颜色越浅，灰度值越小。此项功能跟真实的能力风暴智能机器人基本相同。

### 1.1.3 编程语言


能力风暴智能机器人的编程语言是 VJC1.5，由机器人 C 语言（简称 JC）和流程图编程语言两部分组成。读者使用仿真版的时候，可以用流程图编程。在用流程图编程的同时，VJC1.5 自动生成 JC 代码，便于读者观察、解释或修改程序，参见图 1 - 3。在图 1 - 3 中，左边是流程图，右边是 JC 代码。在流程图编辑界面中，点击工具栏上的“JC 代码”快捷按钮，即可看到当前程序的 JC 代码。



图1 - 3 机器人走正方形的 VJC 程序



## 1.2 VJC1.5 仿真版能做什么

### 1.2.1 为机器人编写程序

在 VJC1.5 仿真版中，可以用流程图编写机器人程序，同时自动生成 JC 代码。但不能直接用 JC 代码为机器人编写程序。

在仿真版中编写的程序对真实的能力风暴机器人也是适用的。

### 1.2.2 创建环境

可以自由创建环境，是 VJC1.5 仿真版的一个突出优点，这往往是使用真实的机器人时所做不到的。在 VJC1.5 仿真版中，读者可以利用光源、声源、障碍物和彩色图带创建各种环境，如房间、迷宫、排雷场地等。下面的图 1 - 4 就是用仿真版创建的一个环境 “篝火晚会场地”，场地左下方是虚拟机器人。



图1 - 4 篝火晚会场地

### 1.2.3 仿真运行

仿真运行就是进入仿真环境，让虚拟机器人按照程序指令进行运动。编好程序、创建

环境之后，就可以仿真运行了。因此，在仿真版中，读者能够直观地看到机器人运行的效果。

## 1.3 光盘的安装与启动

### 1.3.1 安装

可通过如下方式安装“VJC1.5 仿真版”软件，两种方式任选一种。

1. 放入“VJC1.5 仿真版”软件安装光盘，Windows 将自动启动“VJC1.5 仿真版”安装向导。

2. 打开“我的电脑/光盘”，双击安装光盘中的 Setup.exe 程序，即可启动安装向导。

安装时，读者可根据安装向导的提示一步步操作。安装成功后，图形化交互式 C 语言开发库以及范例会自动安装到指定目录（缺省路径时，安装在“C:\Program File\VJC1.5 仿真版”文件夹中）。

### 1.3.2 启动

启动“VJC1.5 仿真版”软件的方式有如下两种：

1. 双击 Windows 桌面上的“VJC1.5 仿真版”图标，进入流程图编辑界面。

2. 点击 Windows 菜单“开始/程序/VJC1.5 仿真版/VJC1.5 仿真版”，进入流程图编辑界面。

进入流程图编辑界面之后，即可为机器人编写程序。编好程序，可以仿真运行。仿真的方法有以下两种：

a) 在菜单栏中选择“工具(T)”下的子菜单“仿真当前程序”。

b) 在工具栏中点击“仿真”快捷按钮，如图 1-5 所示。

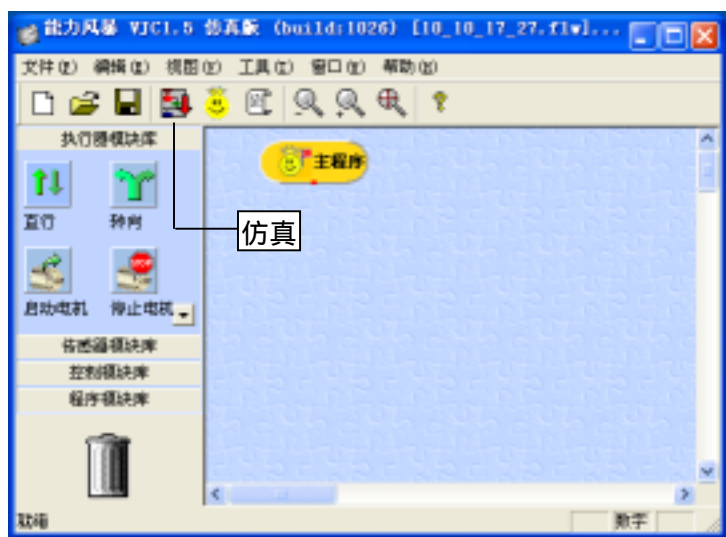


图1 - 5 流程图编辑界面

## 1.4 如何使用“帮助”电子文档

“VJC1.5 仿真版”的帮助电子文档详细介绍了流程图编程和创建环境的方法，是读者学用仿真版的好帮手。进入流程图编辑界面后，选择“帮助”菜单下的“帮助主题”，即可打开此电子文档。

下面的图 1 - 6 是“帮助”电子文档的详细目录，读者可以与电子文档对照使用。

## 图形化编程指南

- [? 简介](#)
- [? 安装和启动](#)
  - [? 系统要求](#)
  - [? 安装](#)
  - [? 启动](#)
  - [? 退出](#)
  - [? 卸载](#)
- [? 流程图编辑界面](#)
  - [? 菜单栏](#)
  - [? 工具栏](#)
  - [? 模块库区](#)
  - [? 流程图程序生成区](#)
  - [? JC代码显示区](#)
  - [? 垃圾箱](#)
- [? 编程与仿真快速入门](#)
  - [? 编程思路](#)
  - [? 编程](#)
  - [? 仿真](#)
- [? 流程图编程方法](#)
  - [? 模块的基本操作](#)
    - [? 模块的新增与连接](#)
    - [? 模块的移动](#)
    - [? 模块的删除](#)
    - [? 模块的复制](#)
  - [? 图形模块的使用方法](#)
    - + [? 执行器模块](#)
    - + [? 传感器模块](#)
    - + [? 控制模块](#)
    - + [? 程序模块](#)
  - [? 子程序调用](#)
    - [? 为什么要用子程序](#)
    - [? 系统子程序](#)
    - [? 自定义子程序](#)
  - [? 变量](#)
    - [? 变量百宝箱](#)
    - [? 变量赋值](#)
    - [? 引用变量](#)
    - [? 变量百宝箱的应用](#)
  - [? 模块库与机器人型号](#)

## 仿真工具简介

- ? 主界面简介
- ? 仿真环境进入与退出
- ? 程序仿真运行
- [-] ? 环境编辑
  - ? 障碍物
  - ? 图带和地毯
  - ? 声源
  - ? 光源
  - ? 清除环境
- [-] ? 机器人初始设置
  - ? 显示/不显示轨迹
  - ? 保留/保留原轨迹
  - ? 初始角度设置
  - ? 红外传感器设置
- [-] ? 环境文件管理
  - ? 新建环境文件
  - ? 保存环境文件
  - ? 加载环境文件
  - ? 删除环境文件
  - ? 导出环境文件
  - ? 导入环境文件
- [-] ? 例程说明
  - ? 如何运行例程
  - [+] ? 典型例程解析
  - ? 其它例程简介
  - ? 注意事项

图1 - 6 “帮助”电子文档的目录

## 第2章 初试身手

本章及以后各章的内容是以机器人项目的形式编排的，既便于初学者入门，也便于已有基础的读者参考。因为智能机器人是一个综合性的知识载体，涉及到机械、电子、软件、仿生学等众多学科，如果要详细介绍，恐非本书的篇幅所能容纳，而且也不适于初学者阅读，所以我们采用了机器人项目的形式供大家学习。

读书固能学到知识，走路做事也能心有所得。以智能机器人为伙伴，做做机器人项目，既轻松有趣，又能逐渐揭开机器人的奥秘，在不知不觉中进入智能机器人的科学殿堂。这正是我们撰写本书的期望所在。

机器人，顾名思义，就是象人的机器。本章主要介绍了机器人的执行器，即机器人的喇叭、显示屏、运动系统，相当于人的口、语言、腿脚。读者只要边看边做，很快就能熟悉机器人了。



## 2.1 歌曲

项目内容：找一段乐谱，用流程图编一首歌曲，然后仿真运行。

相关模块：发音，或者调用子程序。

难度等级：

### ◆ 程序设计

“歌曲”程序涉及模块较少，只要按照乐谱设置“发音”模块，然后将各模块依次连接起来即可。通过这个项目的学习，可掌握模块的参数设置方法，以及流程图编程方法。

下面是编程的步骤：

a) 在流程图编辑界面中，点击“执行器模块库”中的“发音”模块，将其拖到流程图生成区，连接在主程序的下面（图 2 - 1）。



图2 - 1 发音模块

b) 在“发音”模块上右击鼠标，打开参数设置对话框（图 2 - 2），在其中选择音频和时间。例如：图 2 - 2 中选择的音频为 1，时间为四分之一音符。

在编程时，您也可以选择“自定义”，直接输入音频值和时间值。

如果选择了“休止符”，则机器人将不发音。休止的时间可以选择四分之一音符等，也可以选择“自定义”，然后输入时间值。



图2 - 2 “发音”模块的参数设置对话框

c) 编辑完成的歌曲如图 2 - 3 所示。



图2 - 3 歌曲程序示意图

#### ◆ 仿真运行

d) 流程图编好后，点击工具栏中的“仿真”快捷按钮（图 2 - 4），进入仿真环境主界面（图 2 - 5）。



图2 - 4 工具栏





图2 - 5 仿真环境主界面

e) 在仿真环境主界面中，点击“运行”键，可以得到一个虚拟机器人。将虚拟机器人放到中间的运行区，它就会自动执行程序。在本例中，机器人则将演奏所编的歌曲。

◆ 调用子程序

调用子程序的方法参见附录 A。

## 2.2 边唱边跳

项目内容：用流程图编程，使得机器人能够一边唱歌，一边跳舞。

相关模块：直行、转向、发音。

难度等级：

项目解析：本项目可以通过调用系统子程序“边唱边跳”来完成（图 2 - 6），调用子程序方法可参见附录 A。



图2 - 6 边唱边跳

不过，我们建议读者最好还是用“直行”、“转向”、“发音”等模块编一个自己风格的“边唱边跳”程序，然后仿真运行。

设置“直行”、“转向”模块的参数，只要在模块上右击鼠标，打开参数设置对话框，输入速度和时间就可以了。

仿真运行时，选择“显示轨迹”，可以更清楚地看到机器人跳舞的姿态。

## 2.3 走正方形

项目内容：为机器人编程，使之能走一个正方形。走的时候显示轨迹。



图2 - 7 走正方形

相关模块：直行、转向。

难度等级：

项目解析：编写“走正方形”程序可参照图 1 - 1 编写，项目的难点在于要反复调整“转向模块”中转向时间和速度，以便机器人正好转  $90^\circ$ 。“直行”、“转向”模块的参数设置方法可参考“帮助”电子文档“执行器模块”。

## 2.4 走六边形

项目内容：为机器人编程，使之能走一个六边形。走的时候显示轨迹。要求使用“多次循环”模块。

相关模块：转向、直行、多次循环。

难度等级：

项目解析：所有的程序都能用“顺序结构”、“选择结构”以及“循环结构”来表示。本项目中的“循环结构”是程序设计中重要的结构，故须认真掌握。“多次循环”模块在“控制模块库”中。将“多次循环”模块拖到流程图生成区后，在模块上右击鼠标，打开参数设置对话框，即可输入循环次数。另外要注意的是，使用多次循环模块的时候，须将待循环的内容放在循环体内部。

## 2.5 走五角星

项目内容：为机器人编程，使之能走出一个五角星的形状。走的时候显示轨迹。要求使用“条件循环”模块。

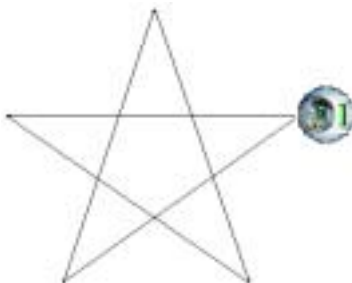


图2 - 8 五角星

相关模块：直行、转向、条件循环、计算。

难度等级：

### ◆ 程序设计

通过该项目的学习可熟悉 VJC1.5 仿真版中引用变量、变量百宝箱的概念以及“条件循环”和“计算”模块的使用。“条件循环”模块在“控制模块库”中。

从图 2 - 8 可以看到，要画出一个五角星，需重复五次“前进、右转”的动作，右转的角度为  $144^{\circ}$ 。五角星的大小由前进的距离所决定。

在本例中，要求使用“条件循环”模块，我们可以将其中的条件表达式设置为

整型变量一 < 5

在程序中，让整型变量一从 0 开始变化，每循环一次，就令整型变量一增加 1。当整型变量一等于 0、1、2、3、4 时，条件表达式都能满足；当整型变量一等于 5 时，条件表达式不再满足。于是循环将进行 5 次，然后就跳出循环，程序结束。参见图 2 - 9。

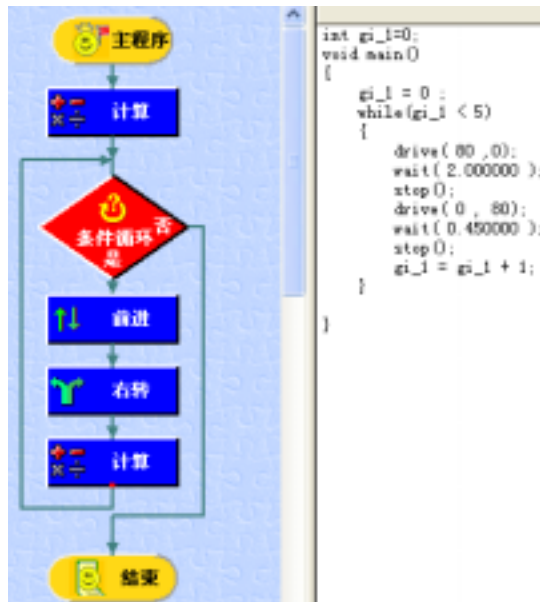


图2 - 9 五角星例程

#### ◆ 参数设置

最初的“计算”模块：计算表达式为

整型变量一 = 0

“前进”模块：速度 80，时间 2.0 秒。可自行调整。

“右转”模块：相对速度 100，时间 0.4 秒。可自行调整。

“条件循环”模块：条件表达式为

整型变量一 < 5

循环体中的“计算”模块：计算表达式为

整型变量一 = 整型变量一 + 1

第二个“计算”模块参数设置的方法如下面 a) - d)所述：

a) 在“计算”模块上右击鼠标，打开参数设置对话框（图 2 - 10）；



图2 - 10 计算模块的参数设置对话框

- b) 在图 2 - 10 中，选择右上角的“引用变量”，这时会出现变量百宝箱（图 2 - 11）。



图2 - 11 变量百宝箱

- c) 在变量百宝箱中点击一下“整型变量一”，在整型变量一右边就会出现一把小钥匙，表明整型变量一被选中了。点击“确认”。
- d) 写出完整的计算表达式“整型变量一 = 整型变量一 + 1”，如图 2 - 12 所示。



图2 - 12 计算表达式

### ◆ 仿真运行

程序编好后，就可以仿真运行了。点击工具栏中的“仿真”快捷按钮，进入仿真环境主界面，选择“显示轨迹”，然后点击“运行”键，得到一个虚拟机器人，将虚拟机器人放在运行区，就能看到机器人走出的轨迹。

计算机执行程序的过程如下：首先将“整型变量一”赋值为 0，然后进入条件循环。每循环一次，机器人前进一段距离，并右转一个角度，画出五角星的一条边，同时整型变量一增加 1。故整型变量一从 0 依次变为 1、2、3、4。当整型变量一等于 5 的时候，条件表达式不再满足，条件循环中止，程序就结束了。

## 2.6 走圆形

项目内容：为机器人编程，使之能走一个圆形。走的时候显示轨迹。

相关模块：启动电机、延时等待、停止电机。



图2 - 13 走圆形

难度等级：

项目解析：设置本项目是为了学习“启动电机”模块使用方法。

“启动电机”模块可用于分别设定左电机功率和右电机功率，并且还可以启动扩展电机。在本例中，我们可以将左电机功率设小一点，右电机功率设大一点，机器人就能逆时针走圆弧线了（图2 - 14）。



图2 - 14 启动电机模块的参数设置对话框

注意：在 VJC1.5 仿真版中，如果将左电机功率和右电机功率设为一正一负，则机器人将原地旋转。

“启动电机”模块和“延时等待”、“停止电机”模块配合使用，可以实现“直行”和“转向”模块的所有功能。



例如：对比下面的两个图，可以看出它们是等价的，这只要简单计算一下就可以知道。因为  $drive(a, b)$  函数给出的电机功率为：左电机功率  $= a + b$ ，右电机功率  $= a - b$ 。而  $motor(1, n_1)$  给出左电机功率  $n_1$ ， $motor(2, n_2)$  给出右电机功率  $n_2$ 。



图2 - 15 “前进”模块的等价形式

## 2.7 奥运五环

项目内容：为机器人编程，使之走出一个奥运五环。走的时候显示轨迹。

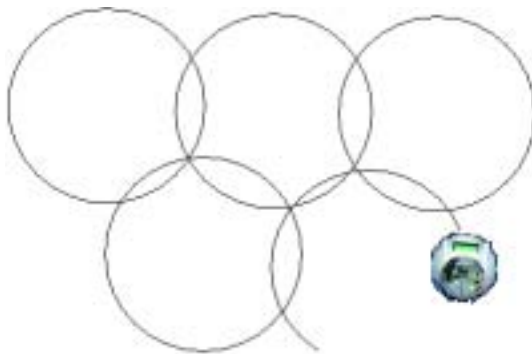


图2 - 16 奥运五环

相关模块：启动电机、延时等待、停止电机、转向。

难度等级：

项目解析：设置本项目为了学习加固“启动电机”模块的使用方法，同时熟悉仿真环境中“显示轨迹”和“保留原轨迹”设置方法。

小技巧：在仿真环境中通过点击鼠标右键可调整机器人的初始角度。

## 2.8 进门比赛

项目内容：大家见过自动化工厂中的自动导引小车（一种运货机器人）吗？自动导引小车能自动从一个工位运货至另一个工位。机器人进门这个活动，就是模拟了这个情景。参见图 2 - 17。

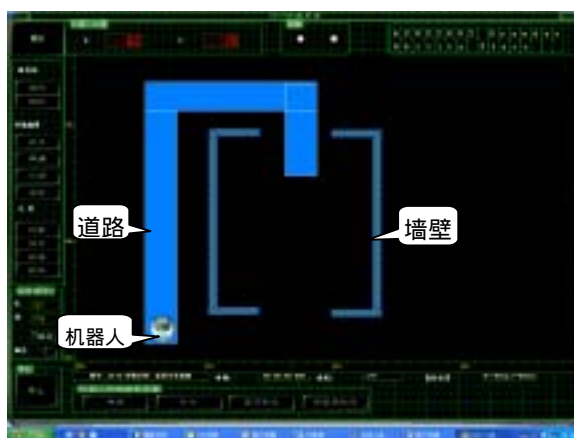


图2 - 17 机器人进门

相关模块：直行、转向。

难度等级：

项目解析：设置本项目的目的是学习仿真系统中创建环境方法。

创建环境 车间，如图 2 - 17 所示。墙壁由障碍物构成，道路由图带构成。

创建环境的方法参见“帮助”电子文档 “环境编辑”。环境创建好之后，需保存起来，以便于反复使用，参见“帮助”电子文档 “保存环境文件”和“加载环境文件”。

## 2.9 一笔画

项目内容：为机器人编程，使之在场地上画出一支铅笔的形状。

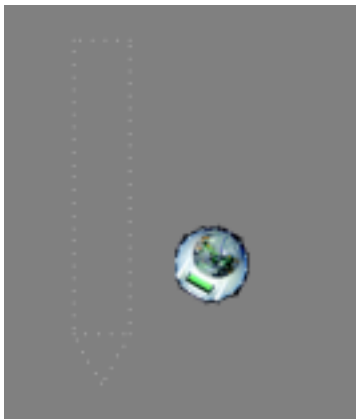


图2 - 18 一笔画

相关模块：直行、转向、启动电机、延时等待、停止电机。

难度等级：

项目解析：为了完成“一笔画”，需要反复调整各模块参数，才能达到比较理想的效果。通过本项目可对已掌握的知识进行巩固加深。在编程前，最好先确定机器人画铅笔图案的起点。

掌握了一笔画铅笔的方法后，可尝试着画花朵、水壶等简单图案，熟练后甚至还可以画出东方明珠等复杂的图案。

## 2.10 求和

项目内容：计算  $1 + 2 + 3 + \dots + 100$ ，并将计算结果显示出来。可以使用多次循环，也可以使用条件循环。

相关模块：计算、显示、多次循环或条件循环。

难度等级：

项目解析：计算机最初的功能就是强大的计算能力，能力风暴机器人所使用的 CPU 采用摩托罗拉 68HC11 芯片，也具有较弱的计算功能，本项目及以下几个与数学计算相关的项目主要是为了锻炼编程思维能力。

相关知识：机器人的显示屏

能力风暴机器人的液晶显示屏可以显示两行信息，每行 16 个字符。显示信息通常为英文、数字或符号，但不能显示中文。

显示信息也可以是变量的值。显示变量值的时候，在“显示”模块上右击鼠标，打开参数设置对话框，选择“引用变量”，就会弹出一个变量百宝箱，可以选择其中的任意一个变量作为显示对象，如图 2 - 19 所示。



图2 - 19 显示模块的参数设置

虚拟机器人的显示屏在仿真环境主界面的右上角。

## 2.11 四则混合运算

项目内容：计算，并显示计算结果。

1. 
$$\frac{3+5 \times (6-1)}{7}$$

2. 
$$\frac{30 \times (4+5) + 18 \times (6-2) - 15 \times (7+8)}{3 \times (3 \times (10+6) - 9)}$$

相关模块：计算。

难度等级：

项目解析：本项目主要目的在于培养编程思维能力。

小技巧：在变量百宝箱中，有若干类变量，除了三个整型变量之外，声音变量、亮度变量、红外变量、地面变量和碰撞变量都可以作为整型变量使用。

## 2.12 整型数与浮点数

项目内容：分别用整型变量和浮点变量计算  $\frac{12}{5}$ ，并显示计算结果。

相关模块：计算、显示。

难度等级：

项目解析：通过本项目可了解整型数和浮点数的区别。

小知识：整型数和浮点数

整型数是指正整数、负整数和零，如-6、0、32等。

浮点数是指带有有限位小数的有理数，如-10.8、0.00、25.01等。

整数既可以是整型数，也可以是浮点数，例如255是整型数，而255.0则是浮点数。

在VJC1.5仿真版中，整型数与整型数相除，得到的商是一个整型数，并且计算结果中的小数部分将被忽略。例如：用整型变量计算时， $100 \div 3 = 33$ 。

浮点数与浮点数相除，得到的商是一个浮点数，计算结果中的小数部分将保留下来。例如：用浮点变量计算时， $100.0 \div 3.0 = 33.333333333333$ 。

如果读者想更深入地了解整型数和浮点数，可查阅“C语言程序设计”方面的书。

## 2.13 霓虹灯

项目内容：设计机器人眼睛的状态，使之就象霓虹灯一样，灯光闪烁，并且有色彩变化。

相关模块：眼睛、延时等待、永远循环。

难度等级：

项目解析：通过本项目可学习“永远循环”模块使用方法，同时了解怎样控制机器人眼睛。较复杂的程序在编写好后要经过不断调式才能按设计意图运行，机器人的眼睛除了增加编程趣味性，还可用于程序调试。

相关知识：永远循环

打开“控制模块库”，可以看到“永远循环”模块。程序中使用“永远循环”模块的时候，需将待循环的内容放在循环体内（图 2 - 20）。

机器人执行程序的时候，一旦进入永远循环，将一直重复循环体的内容，而不会跳出循环去执行后面的指令。用“永远循环”编写程序的时候，应始终注意它的这个特点。

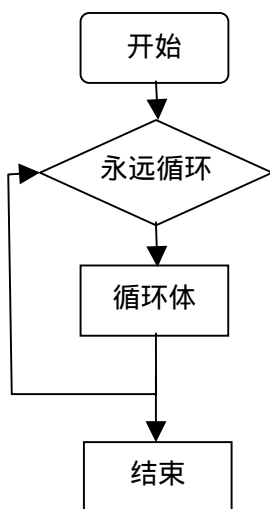


图2 - 20 永远循环的使用方法

## 2.14 汽车

项目内容：用机器人做一个汽车。前进、后退、转弯、划弧线、停止、亮车灯等动作任意组合。车灯可用机器人的眼睛代替。

相关模块：直行、转向、启动电机、延时等待、停止电机、眼睛。

难度等级：

## 2.15 迎宾机器人

项目内容：设计一个迎宾机器人，除了身体会动作，还会唱歌、显示欢迎信息、眼睛发光。

相关模块：直行、转向、眼睛、显示、发音、永远循环。

难度等级：

## 2.16 回廊

项目内容：用障碍物和彩色图带设计一个弯曲的走廊，让机器人从走廊中走过去。创建环境的方法参见“帮助”电子文档 环境编辑。

相关模块：直行、转向。

难度等级：

项目解析：要顺利走出回廊，需反复调节“直行”和“转向”模块的参数。以后我们学习了机器人的红外传感器和碰撞传感器，就可用更好的方法完成这个项目。

## 2.17 动态字符

项目内容：机器人显示字符“ABCD”，并且字符从左向右运动。



图2 - 21 仿真界面中的显示屏

相关模块：显示、延时等待、永远循环。

难度等级：

项目解析：在本例中，可以这样来设计程序：用 13 个显示模块，分别将“ABCD”显示在不同部位，如图 2 - 21/22 所示。其中，前一个“显示”模块与后一个“显示”模块之间插一个“延时等待”模块，延时等待的时间取 0.2 秒钟。当程序运行的时候，就会产生动态的效果。

A	B	C	D											
	A	B	C	D										
		A	B	C	D									
			A	B	C	D								
				A	B	C	D							
					A	B	C	D						
						A	B	C	D					
							A	B	C	D				
								A	B	C	D			
									A	B	C	D		
										A	B	C	D	
											A	B	C	D

图2 - 22 动态字符的设计方法



## 2.18 波浪

项目内容：用字符“\*”做成波浪的形状，并且看起来波浪在不停地起伏。

相关模块：显示、延时等待、永远循环。

## 2.19 落地扇

项目内容：用机器人做一个电风扇，来回摆动着吹。

难度等级：

项目解析：设置这个项目主要为了知道风扇的使用方法。

相关知识：机器人的风扇

能力风暴智能机器人本身没有风扇，风扇可以自己安装，如图 2 - 23 所示。

在 VJC1.5 仿真版环境中，风扇通常由扩展电机驱动，在程序中调用“启动电机”模块进行控制。进行参数设置时，在对话框中选中“扩展电机”即可（图 2 - 23）。

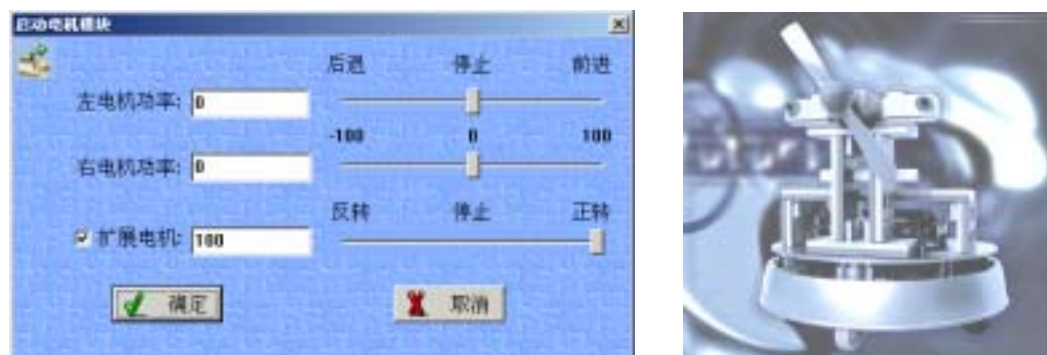


图2 - 23 选择“扩展电机”驱动风扇

## 2.20 等比求和

项目内容：计算  $4 + 4 \times 3 + 4 \times 3^2 + \dots + 4 \times 3^7$ ，并显示计算结果。

相关模块：计算、多次循环、显示。

难度等级：

项目解析：通过本项目进一步掌握“多次循环”模块使用方法。为求取项目中等比数列的值可用采用下面介绍的公式进行计算，也可将计算式改写成易编程实现的形式。建议读者设计多种方法完成该项目。

小知识：等比数列是形如

$$a, ad, ad^2, ad^3, ad^4, ad^5, ad^6, \dots$$

的序列，其中是  $a$  等比数列的首项， $d$  是等比数列的公比。由于等比数列有明显的规律，故对等比数列的前  $n$  项进行求和是很简便的。如  $d$  等于 1，则  $S_n = na$ 。如  $d$  不等于 1，令

$$S_n = a + ad + ad^2 + \dots + ad^{n-2} + ad^{n-1}$$

等式两边同乘以公比  $d$ ，得到

$$dS_n = ad + ad^2 + ad^3 + \dots + ad^{n-1} + ad^n$$

再用上一式减去下一式，得到：

$$S_n - dS_n = a - ad^n$$

即可得到

$$S_n = \frac{a(1-d^n)}{1-d}$$

## 第3章 稳步前进

在本章中，读者将接触到智能机器人的感觉系统。智能机器人的感觉是由传感器实现的。能力风暴智能机器人装备的传感器有红外传感器、光敏传感器、碰撞传感器、声音传感器、光电编码器、地面灰度传感器等 6 种。读者通过做本章的机器人项目，就能熟悉机器人的传感器工作原理。



## 3.1 打招呼

项目内容：用圆形障碍物当作人，机器人每当走到“人”跟前的时候，都会叫一声，打个招呼，然后侧身走开。

相关模块：红外测障、直行、转向。

创建环境 在机器人运行场地中，添加几个圆形障碍物（图 3 - 1）。

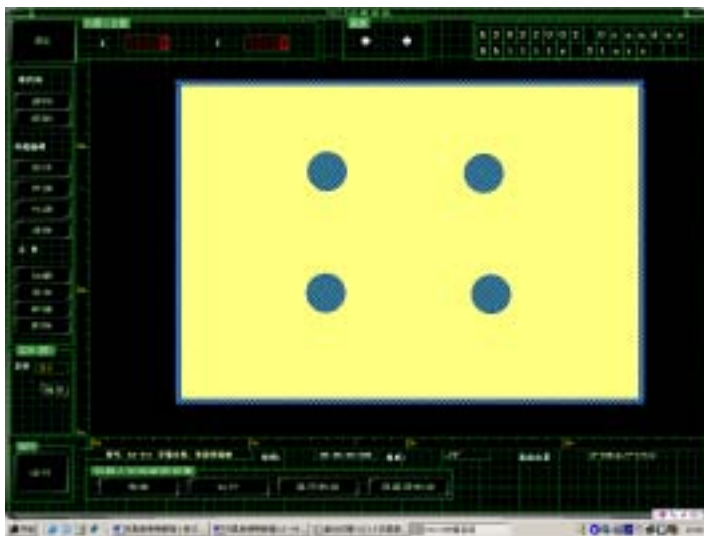


图3 - 1 打招呼场地

难度等级：

### ◆ 程序设计

机器人探测障碍物，用的是红外传感器。VJC1.5 仿真版中，机器人红外传感器的探测半径为 0~30，可以根据需要在此范围内调整（参见帮助电子文档 红外传感器设置）。

编程时，可以让机器人一边走一边进行红外检测。如果检测到有障碍物，就发音，意为看到人了，打个招呼。然后转向，继续走。机器人可以探测到前、左、右三个方向的障碍物。如果检测到障碍物在前方，就右转；障碍物在左方，就右转；障碍物在右方，就左转；否则，就前进。参见图 3 - 6。

### ◆ 参数设置

前进模块：速度 80，时间 0.1 秒。

“红外测障”模块：变量选择“红外变量一”。

三个“条件判断”模块的条件表达式为

红外变量一 == 前
红外变量一 == 左
红外变量一 == 右

设置“条件判断”模块参数的时候，首先打开参数设置对话框（图 3 - 2）。



图3 - 2 “条件判断”模块的对话框

在对话框中点击“整型变量一”，随之会弹出变量百宝箱（图 3 - 3）。



图3 - 3 变量百宝箱

在变量百宝箱中点击“红外”图标，红外变量一、红外变量二、红外变量三就会出现在变量百宝箱中（图3-4）。



图3-4 红外变量

点击选中红外变量一，确认，红外变量一就会出现在条件判断模块的参数设置对话框中，



图3-5 条件判断表达式

此时就可以设置以上列表中的条件判断表达式了（图3-5）。

由例程（图3-6）可以看出，在前、左、右方都没有障碍物的情况下，循环的周期几乎就由“前进”模块的时间参数所决定。每循环一次，程序发出一次红外测障的指令，机器人的红外测障系统就开启一次。故循环的周期越短，机器人的反应就越灵敏；循环的周期越长，机器人的反应就越迟钝。在仿真版中，建议循环周期取为0.1秒或0.05秒。

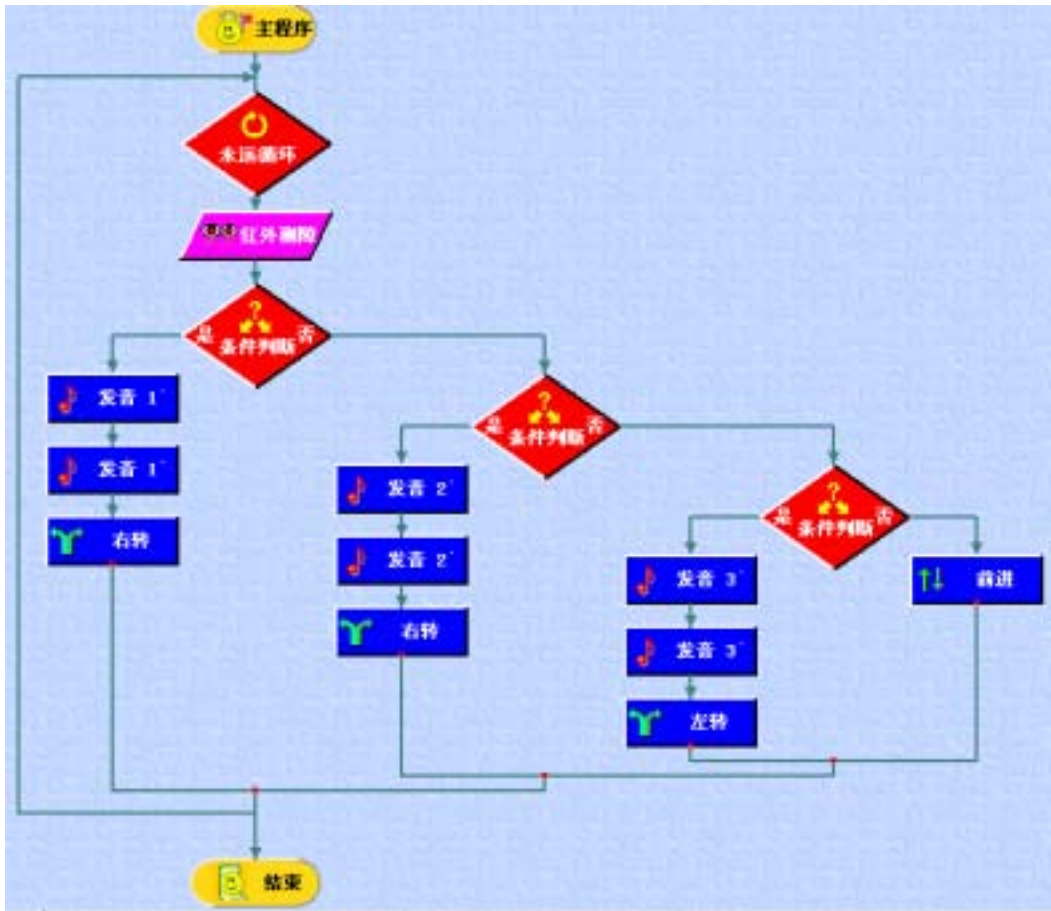


图3 - 6 打招呼例程之一

◆ 仿真运行

仿真运行的时候，在机器人运行场地中创建一个房间，里面放一些圆形障碍物。让机器人在房间里活动。也可以加载“单房间场地”，然后进入环境。

◆ 单功能模块与多功能模块

红外测障模块除了有检测障碍物的功能，还有条件判断的功能。右击模块，打开参数设置对话框，可以看到中间有“检测完成后，进行条件判断”的字样（图3 - 7）。

在这排文字上点击一下，就会出现如图3 - 8所示的对话框，在其中可以设置条件表达式。

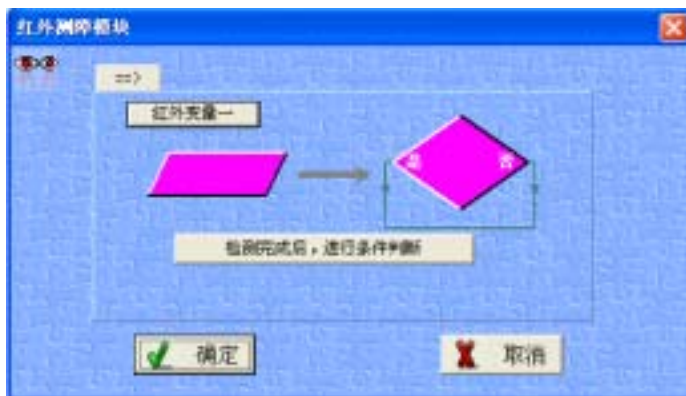


图3 - 7 红外测障模块的参数设置对话框之一



图3 - 8 红外测障模块的参数设置对话框之二

设置完成后，点击“确定”按钮，“红外测障”模块的形状就由平行四边形变成了菱形（图 3 - 9）。平行四边形模块称为单功能模块，而菱形模块则称为多功能模块，因为它具有红外检测与条件判断两种功能。

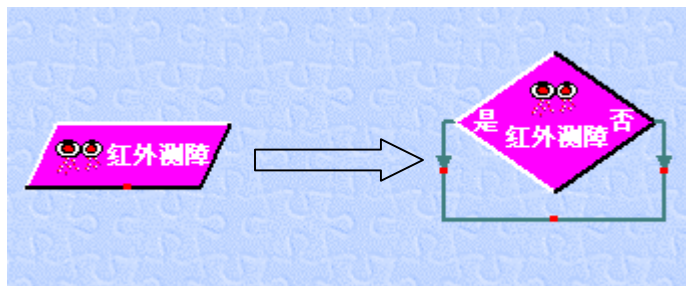


图3 - 9 单功能模块转变为多功能模块



于是我们可以将图 3 - 6 的程序等价地写成如下形式 ( 图 3 - 10 ):

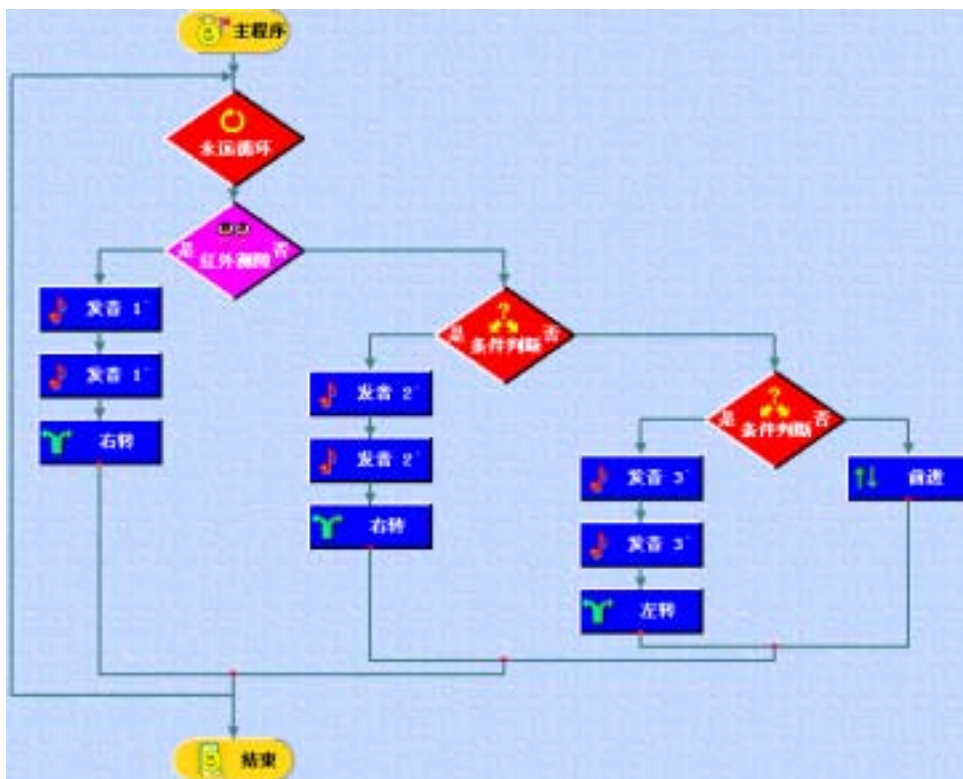


图3 - 10 打招呼例程之二

可见一个多功能模块相当于一个单功能模块和一个条件判断模块的组合。

除了“红外测障”之外，传感器模块库中的其它模块都具有两种功能，既可以作为单功能模块，又可以作为多功能模块，详见帮助电子文档 传感器模块。

## 3.2 叫人起床

项目内容：夜晚天黑的时候，机器人静止不动。一旦天亮了，机器人就动起来，并发出声音、眼睛发光，持续一段时间才停下来。

相关模块：亮度检测、发音、转向、条件循环。

难度等级：

### ◆ 程序设计

首先设计一个亮度检测小程序，第一个“亮度检测”模块参数设置为方式“左”，变量选择“亮度变量一”，第二个“亮度检测”模块参数设置为方式“右”，变量选择“亮度变量二”。“显示”模块两行显示信息分别为“亮度变量一、亮度变量二”。延时等待 0.1 秒（图 3 - 11）。



图3 - 11 亮度检测

将以上程序仿真运行。在运行场地上添加光源，光源半径设为 180，渐变值 30。然后将机器人放在光圈中和光圈外（图 3 - 12），观察一下机器人左右眼检测到的亮度值。

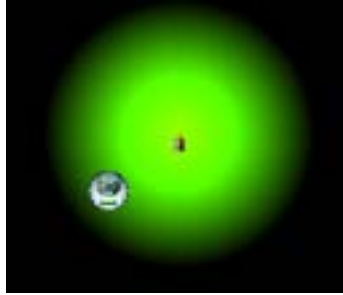


图3 - 12 亮度检测界面

为了达到叫人起床的目的，需要检测环境光的亮度。假设天黑的时候，环境光的亮度大于 200，而天亮的时候，环境光的亮度小于 200。当天黑的时候，机器人就一直反复检测环境光的亮度。一旦检测到环境光的亮度小于 200，表明天亮了，机器人就开始动作，转动身体、发音、眼睛放光。在程序中，机器人检测环境光的亮度的过程可以用条件循环来实现（图 3 - 13）。

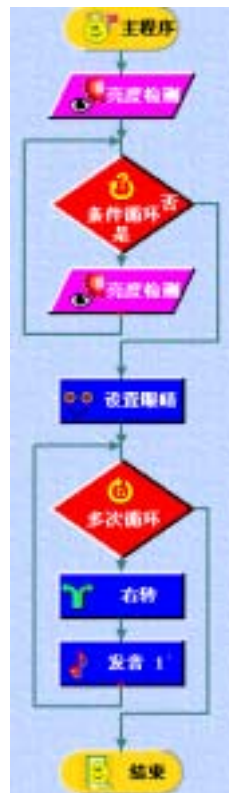


图3 - 13 叫人起床例程

## ◆ 参数设置

“亮度检测”模块：亮度检测模块可以分别检测左眼亮度、右眼亮度、平均亮度、以及左右眼亮度之差（图 3 - 14）。

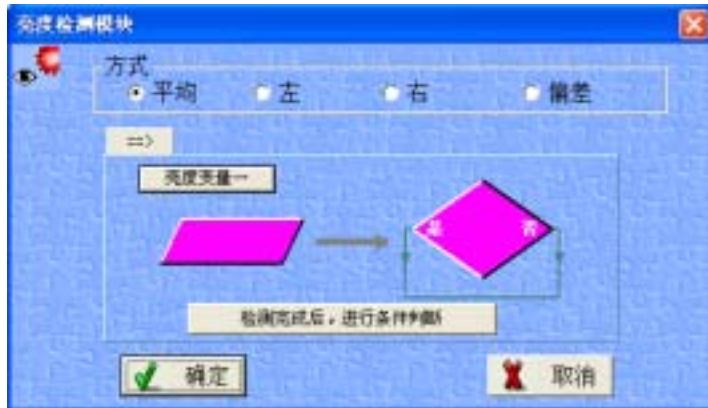



图3 - 14 亮度检测模块的对话框

在本例中，我们选择的是“平均”方式。变量选择“亮度变量一”，即将检测到的亮度值保存在“亮度变量一”中。在程序执行的时候，每检测一次，就会将前次的亮度值覆盖，所以“亮度变量一”中保存的总是最近一次检测到的亮度值。

“条件循环”模块：条件表达式设置为

亮度变量一 ≥ 200

于是当环境光的亮度大于 200 的时候，条件表达式成立，表明天还黑着，机器人就继续检测。当亮度小于 200 的时候，表明天亮了，此时条件表达式不再满足，条件循环中止，机器人就转而去执行条件循环后面的指令。

由于程序开始时默认“亮度变量一”的初始值等于零，我们在进入条件循环之前先调用一次亮度检测模块，以获取环境光的当前值，并将其赋给“亮度变量一”，可以避免程序错误地跳过条件循环。观察程序的 JC 代码，可以更清楚地看到这一点。在流程图编辑界面中，点击工具栏上的“JC 代码”快捷按钮, 即可看到当前程序的 JC 代码。

其余模块：略。

## ◆ 仿真运行

在机器人运行场地中添加光源，设置光源半径为 180，渐变值为 30。先将机器人放在黑暗的地方，然后再将机器人放在光圈中。观察是否能达到叫人起床的效果。

### 3.3 彩色地毯

项目内容：机器人在彩色地毯上行走，遇到不同的颜色能发出不同的声音。

相关模块：地面检测、直行、转向、发音、永远循环、条件判断。

难度等级：

创建环境 彩色地毯。参见帮助电子文档/环境编辑/图带和地毯。

项目解析：“地面检测”模块用于对地面的灰度进行检测，检测到的值为 0~255 之间的整型数，数值越大，地面颜色越深；数值越小，地面颜色越浅。在正式编程之前，先编写一个地面灰度检测小程序（参见图 3 - 22），检测一下地毯不同部分的灰度。然后再构思本项目的解决方案。

### 3.4 百米赛跑

项目内容：开始时，机器人站在跑道的起点处等待起跑。一旦听到发令枪响，就立刻沿着跑道向终点冲过去。在跑的过程中机器人不能越出跑道。



图3 - 15 百米赛跑场地

相关模块：地面检测、声音检测、条件循环、直行。

难度等级：

项目解析：在很多比赛中要求声控启动，当机器人听到声音后才能运动。这个项目主要演示在场地中添加声源，具体方法可参见“帮助”电子文档/环境编辑/声源。仿真运行时，先把机器人放在跑道的起点处，然后添加声源。

### 3.5 计算走过的距离

项目内容：计算机器人走过的距离。

相关模块：直行、转向、转角检测、显示。

难度等级：

#### ◆ 程序设计

在本例中，我们让机器人先走一段直线，接着走一段弧线，然后再走一段直线。最后，调用两个转角检测模块，分别检测左轮和右轮的转角，并显示出来，例程如图 3 - 16 所示。



图3 - 16 转角检测例程

#### ◆ 参数设置

“转角检测”模块：打开“转角检测”模块的对话框，如图 3 - 17 所示。



图3 - 17 转角检测模块的参数设置对话框之一

在图 3 - 17 对话框中选择“左轮、编码器检测”，就会弹出如下对话框：



图3 - 18 转角检测模块的参数设置对话框之二

在其中选择“转角变量一”，确定。

用同样的方法，将第二个“转角检测”模块的参数设置为“右轮、编码器检测、转角变量二”。

“显示”模块：选择“引用变量”，在两行中分别显示“转角变量一”和“转角变量二”。

#### ◆ 仿真运行

仿真运行时，显示轨迹，可以看到机器人走过一条曲线，最后在右上角的显示屏上显示出左右轮的转角数。

机器人的轮子转一圈，机器人的光电编码器接收到 33 个脉冲，转角计数为 33。真实的能力风暴机器人轮子的直径  $d = 64\text{mm}$ 。您可以根据检测到的左右轮平均转角数，用如

下公式计算出机器人走过的平均距离：

$$\text{距离} = \frac{\text{平均转角数}}{33} \times \pi d$$

您还可以在以上例程中添加一段程序，计算机器人走过的距离，并将计算结果显示出来。不过，用上述公式计算出的结果适用于真实的机器人，与仿真界面上的标尺不一定吻合。

#### 小知识：光电编码器

光电编码器是靠发射与接收红外光来工作的，由码盘和光电编码模块组成。码盘安装在机器人的轮子上，随轮子一起转动。光电编码模块则装在轮子的支架上。码盘上共有66个辐条，黑白相间。光电编码模块同时具有发射与接收红外线的功能，参见图3-19。

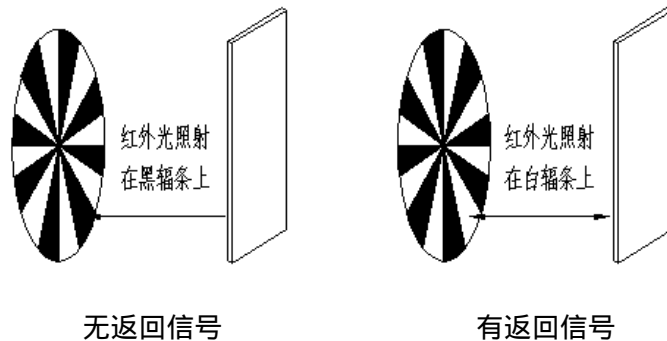


图3-19 光电编码器的工作原理图

从图中可以看出：红外线照射在黑色辐条上没有反射信号，因为红外光大部分已经被黑色辐条吸收；当红外线照射在白色辐条上有反射信号，因为红外光在白色辐条上反射强烈。于是，当机器人运动时，光电编码器就能接收到脉冲信号。轮子每转一圈，共产生33个脉冲。光电编码器有计数功能，能累加接收到的脉冲数。



## 3.6 秒表

项目内容：让机器人在显示屏上显示时间，可以当作秒表来使用（图 3 - 20）。

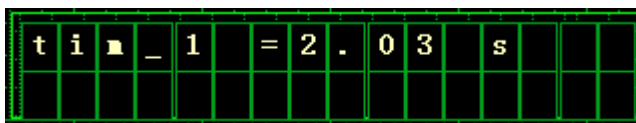


图3 - 20 显示时间

相关模块：系统时间、显示、永远循环。

难度等级：

项目解析：在本书中很多项目可用于机器人比赛，比如走迷宫、灭火。进行机器人比赛，就需要计算所花的时间。VJC1.5 仿真版提供了“系统时间”模块，能满足程序设计的需要。

## 3.7 风扇定时

项目内容：用机器人做成一个电风扇，来回摆动着吹。并且有电风扇的定时功能，吹一段时间就停下来。

相关模块：转向、启动电机、停止电机、系统时间、条件循环。

难度等级：

### ◆ 程序设计

要让风扇定时，则须调用“系统时间”模块。在程序中，控制风扇运行的时间可以用条件循环来实现。程序一开始，首先启动扩展电机，然后进入条件循环，条件表达式为：

时间变量一 < 10.0000

条件循环的内容为：左转、右转、检测系统时间，并将它在显示屏上显示出来。一旦检测到系统时间超过 10 秒钟，就跳出条件循环，停止扩展电机，程序结束。参见图 3 - 21。

### ◆ 参数设置

“启动电机”模块：选择“扩展电机”，而将左右电机的功率均设置为零。

“条件循环”模块：条件表达式为

时间变量一 < 10.0000

“系统时间”模块：变量选择“时间变量一”。

“停止电机”模块：停止扩展电机。

#### ◆ 仿真运行

点击“仿真”快捷按钮，进入仿真环境主界面，点击“运行”键，将虚拟机器人放到运行区就可以了。运行时可以看到右上角显示屏上有时间显示。

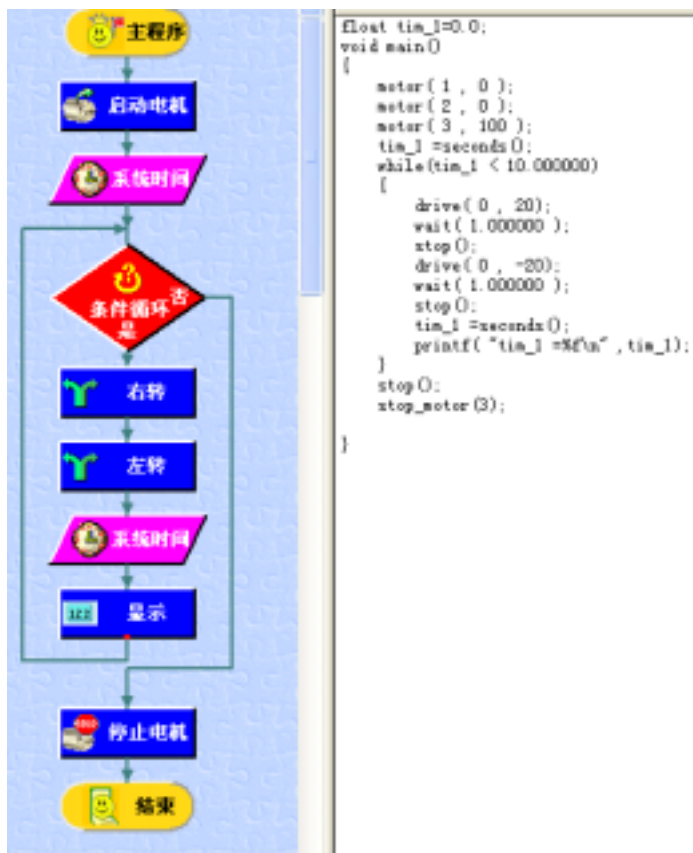


图3 - 21 风扇定时例程

## 3.8 找房门

项目内容：把机器人放在一个房间里，机器人能自己找到房门，并从房门走出去。

相关模块：红外测障、转向、直行、地面检测。

难度等级：

创建环境 单房间场地。墙壁由障碍物构成，门口的标志是一道白色图带，门可以开在任意一边。如图 3 - 25 所示。

### ◆ 程序设计

先编一个地面检测程序，检测一下地面灰度并显示出来。将机器人分别放在黑色的地面和白色的图带上，观察它们的灰度值。参考值为：黑色地面 255，白色图带 0。“地面检测”模块使用的变量是“地面变量一”，“显示”模块显示的信息为“地面变量一”，延时等待 0.1 秒。参见图 3 - 22。

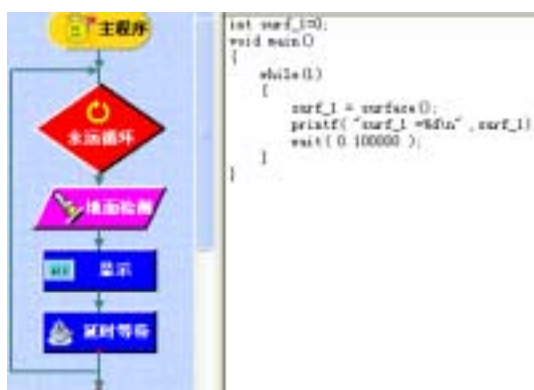


图3 - 22 地面灰度检测

机器人找房门的方式如图 3 - 23 所示。机器人一边前进一边进行红外测障，如果发现前方有障碍物，就转向  $90^{\circ}$ ，继续前进和红外测障。机器人运行的过程表现为“前进、转向、前进、转向……”的循环。在运行过程中，如果发现了门口的白线，就跳出条件循环，前进一段距离，走出门口，程序结束。机器人无论从什么地方出发，如果前进、转向的动作重复了 5 次，还没有发现门口的白线，就换一个方向进行搜索。示范程序如图 3 - 24 所示。

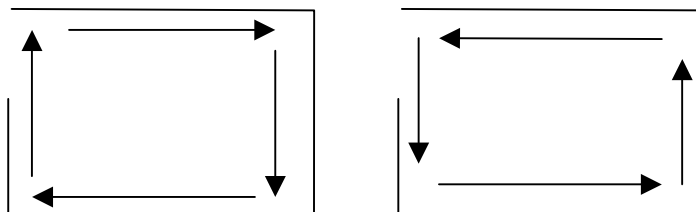


图3 - 23 机器人找房门的方式

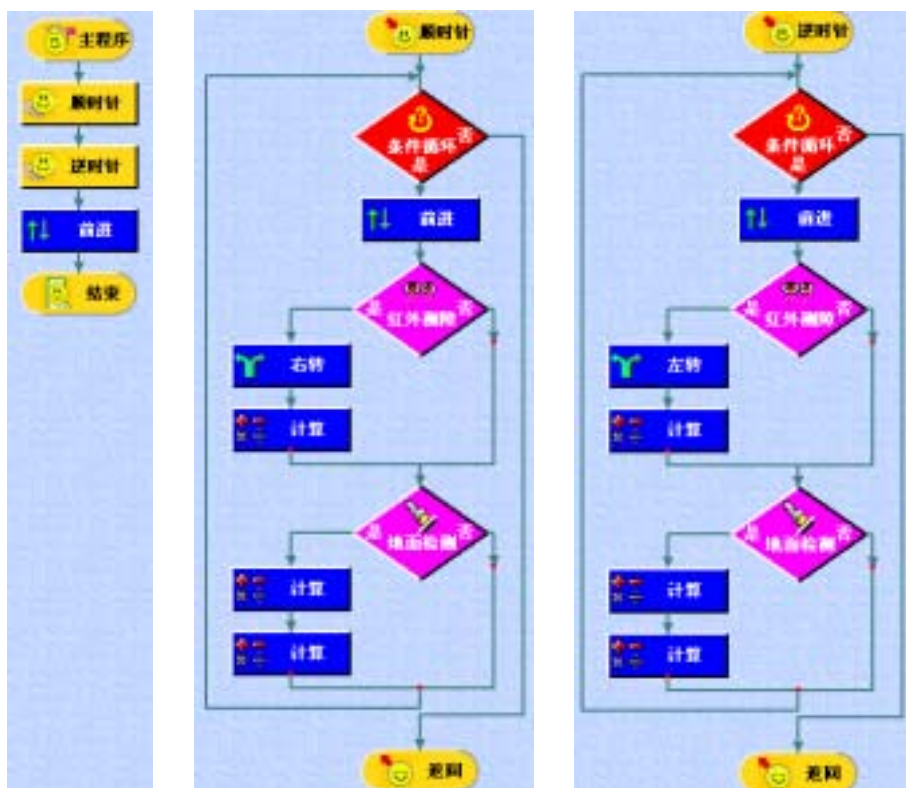


图3 - 24 找房门例程

### ◆ 参数设置

主程序：包含“顺时针”子程序、“逆时针”子程序和“前进”模块。“前进”模块的作用是在发现门口的白线后，让机器人向前移动一段距离，走出房门。其参数设置为“速度 80，时间 1.000 秒”。

子程序：

a) “顺时针”子程序

“条件循环”模块：条件表达式为

$$\text{整型变量一} < 5$$

“红外测障”模块：条件表达式为

$$\text{红外变量一} == \text{前}$$

“地面检测”模块：条件表达式为

$$\text{地面变量一} < 180$$

“前进”模块：速度 50，时间 0.1 秒。

“右转”模块：速度 79，时间 0.3 秒，右转  $90^0$ 。如果角度不准确，可调整参数。

第一个“计算”模块：计算表达式为

$$\text{整型变量一} = \text{整型变量一} + 1$$

第二个“计算”模块：计算表达式为

$$\text{整型变量一} = 5$$

第三个“计算”模块：计算表达式为

$$\text{整型变量二} = 5$$

b) “逆时针”子程序

“条件循环”模块：条件表达式为

$$\text{整型变量二} < 5$$

“红外测障”模块：条件表达式为

$$\text{红外变量一} == \text{前}$$

“地面检测”模块：条件表达式为

$$\text{地面变量一} < 180$$

“前进”模块：“速度 50，时间 0.1 秒”。

“左转”模块：速度-79，时间 0.3 秒，左转  $90^0$ 。如果角度不准确，可调整参数。

第一个“计算”模块：计算表达式为

$$\text{整型变量二} = \text{整型变量二} + 1$$

第二个“计算”模块：计算表达式为

$$\text{整型变量一} = 5$$

第三个“计算”模块：计算表达式为

$$\text{整型变量二} = 5$$

### ◆ 仿真运行

仿真运行时，首先创建一个房间，门可以开在任意一边，门口添加一条白色图带。然后将机器人放在房间中就可以了。参见图 3 - 25。

在例程中，机器人先执行“顺时针”子程序，进行顺时针搜索。顺时针搜索时，每当发现前方有障碍物，机器人就右转  $90^{\circ}$ ，同时第一个计算模块的整型变量一增加 1。整型变量一达到 5 的时候，说明机器人已经顺时针走过一圈以上，这时条件循环表达式不再满足，于是条件循环中止，机器人转而执行“逆时针”子程序。

“逆时针”子程序的运行过程与上述过程是类似的。

在执行子程序的过程中，一旦机器人检测到了门口的白线，子程序中的第二个“计算”模块、第三个“计算”模块就将整型变量一、整型变量二都赋值为 5，这时两个子程序的条件循环表达式均不再满足，于是机器人就跳出或跳过子程序，转而去执行后面的“前进”模块，走出房门。至此，整个程序就完成了。

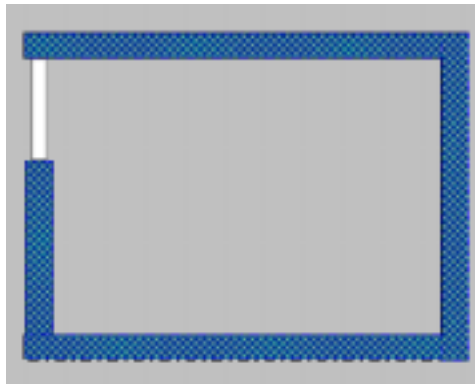


图3 - 25 找房门场地

## 3.9 醉汉

项目内容：机器人模仿醉汉的姿态，跌跌撞撞地走。如果碰到人或障碍物，就转过身，继续走。

相关模块：直行、转向、碰撞检测。

难度等级：

创建环境 设置一个房间，在房间里摆一些障碍物。

小知识：碰撞传感器

在此，我们简要介绍机器人的碰撞传感器，以型号为 AS-MII 的真实机器人为例。碰撞传感器是感知碰撞信息的传感器。在能力风暴智能机器人的左前、右前、左后、右后部位设置有四个碰撞开关，它们与碰撞环共同构成了碰撞传感器(图 3 - 26)。

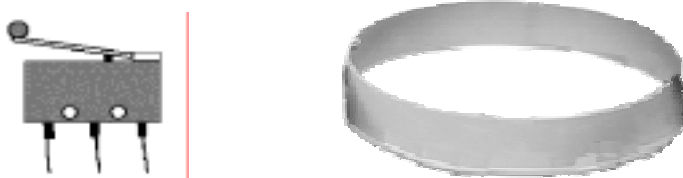


图3 - 26 碰撞开关及碰撞环

碰撞环与机器人底盘柔性连接，在受力后与底盘产生相对位移，触发固连在底盘上的碰撞开关，使之闭合。从而机器人就能感知来自不同方向的碰撞。参见图 3 - 27。

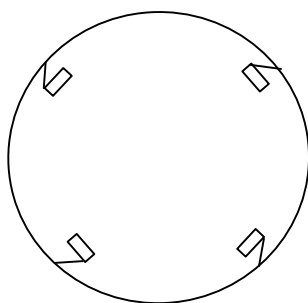


图3 - 27 机器人的碰撞传感器示意图

根据接收到的开关信号，机器人能够判别八个方向的碰撞(见图 3 - 28)。

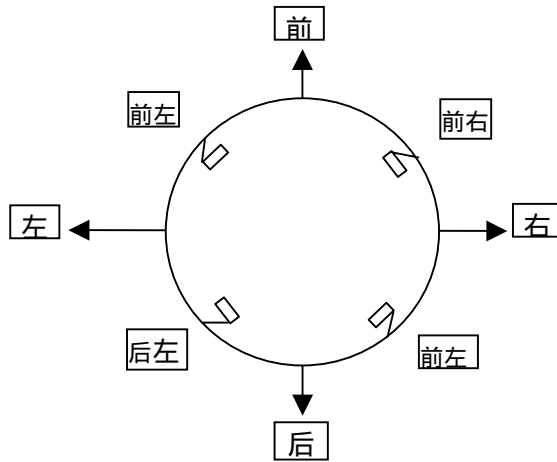


图3 - 28 碰撞方位

在 VJC1.5 仿真版中，虚拟机器人至多只能识别前、后、左、右四个方向的碰撞。

### 3.10 自由行走

项目内容：机器人在空间漫步，有时候会停下来，欣赏欣赏风景，唱唱歌，然后又继续走。遇到障碍物能自动避开，但不会撞上去。

相关模块：转向、直行、红外测障、碰撞检测、条件判断、发音。

难度等级：

创建环境 自由创意。



## 3.11 趋光的飞蛾

项目内容：机器人模仿飞蛾，一旦发现光，就会向着光走过去。

相关模块：亮度检测、启动电机、延时等待、停止电机、永远循环、条件判断。

难度等级：

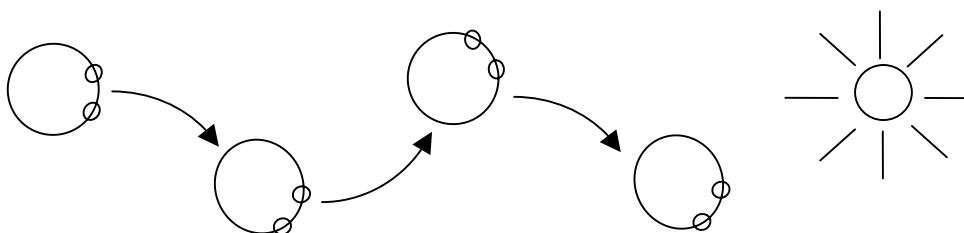


图3 - 29 机器人趋光的方式

### ◆ 程序分析

问：机器人如何才能朝着光源走过去？

答：可以采用比较机器人左右眼的亮度值的方法。如果左眼的亮度值小于右眼的亮度值，说明机器人的左眼接收到的光线较强，离光源较近；而机器人的右眼接收到的光线较弱，离光源较远。这时就让机器人逆时针划一段弧线，目的是让机器人的右眼离光源更近些。类似地，如果右眼的亮度值小于左眼的亮度值，则说明机器人的左眼离光源较远，这时就让机器人顺时针划一段弧线。通过逆时针划弧线和顺时针划弧线的交替，机器人就可以逐步接近光源了。参见图3 - 29。

问：机器人怎样才不会撞上蜡烛？

答：如果机器人左眼或右眼的亮度值小于某一个值，比如20，就停止电机。这样，机器人走到蜡烛跟前就会停下来，而不会撞上蜡烛。参见图3 - 30的例程。

### ◆ 参数设置

第一个“亮度检测”模块：方式选择“左”，变量选择“亮度变量一”。

第二个“亮度检测”模块：方式选择“右”，变量选择“亮度变量二”。

第一个“条件判断”模块：条件表达式为

条件一：亮度变量一<20。条件二：有效，亮度变量二<20，条件逻辑关系“或”。

第二个“条件判断”模块：条件表达式为

亮度变量一 < 亮度变量二。

其下的“启动电机”模块：左电机功率 20，右电机功率 80。

第三个“条件判断”模块：条件表达式为

亮度变量二 < 亮度变量一。

其下的“启动电机”模块：左电机功率 80，右电机功率 20。

“延时等待”模块：时间设为 0.1 秒。

#### ◆ 仿真运行

仿真运行时，在运行场地中央添加光源，光源半径设为 180，渐变值设为 30。然后将虚拟机器人放在光圈的边缘，机器人就会向光源中心走过去。走到蜡烛跟前，机器人会自动停下来。

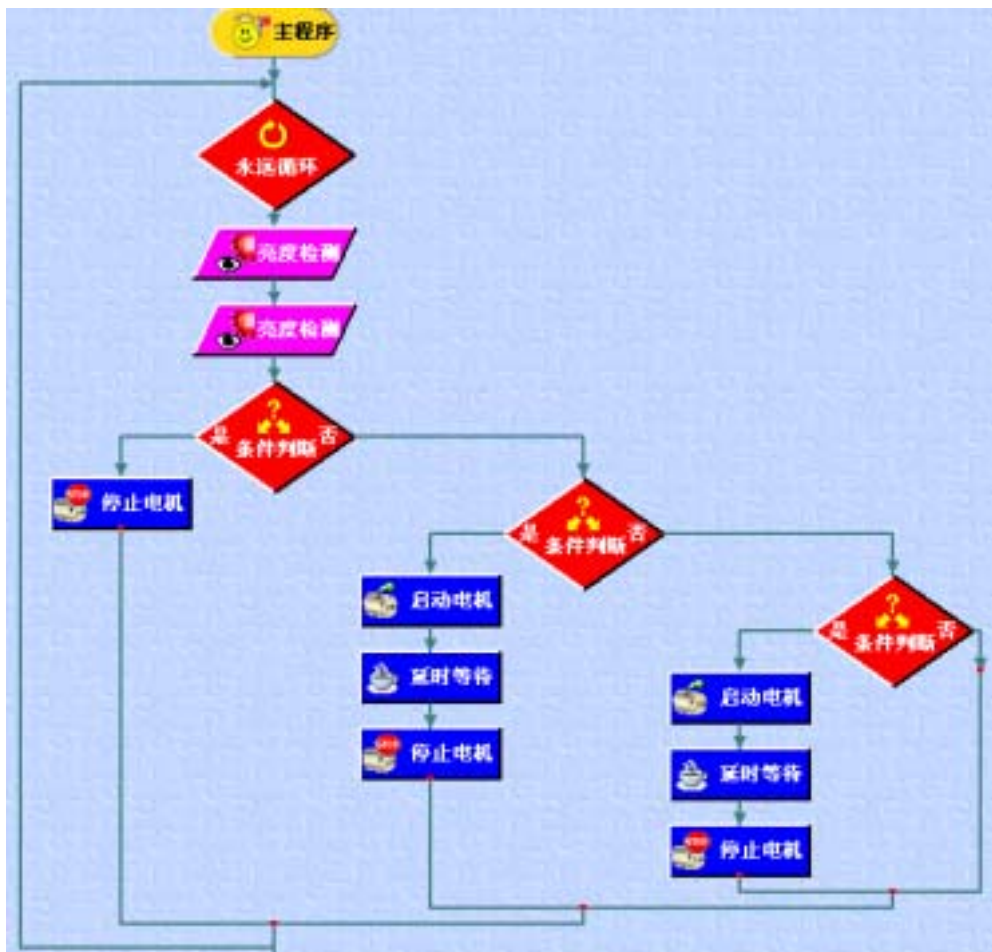


图3 - 30 趋光的飞蛾例程

## 3.12 鼯鼠

项目内容：机器人模仿鼯鼠，在地洞中来回走动，如发现光就转身钻进地洞里躲起来。参见图 3 - 31。

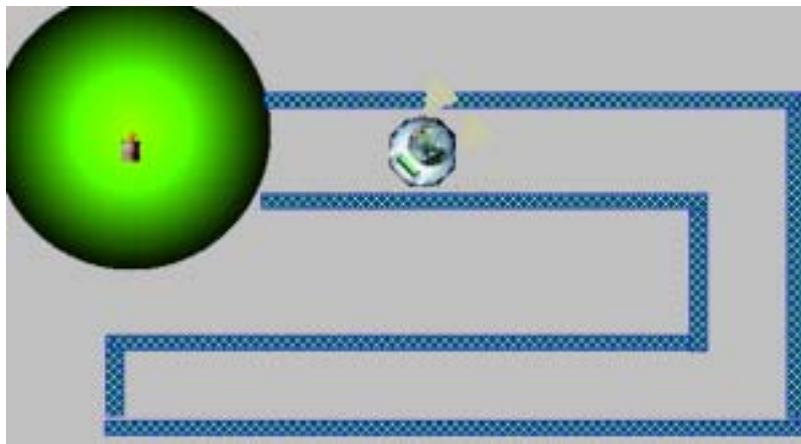


图3 - 31 鼯鼠场地

相关模块：亮度检测、红外测障、碰撞检测、启动电机、延时等待、停止电机、永远循环、条件判断。

难度等级：

### 3.13 沿线走

项目内容：让机器人沿着彩色线条，从一端走到另一端。

相关模块：地面检测、直行、转向、条件判断、永远循环。

难度等级：

创建环境 彩色线条，如图 3 - 32 所示：

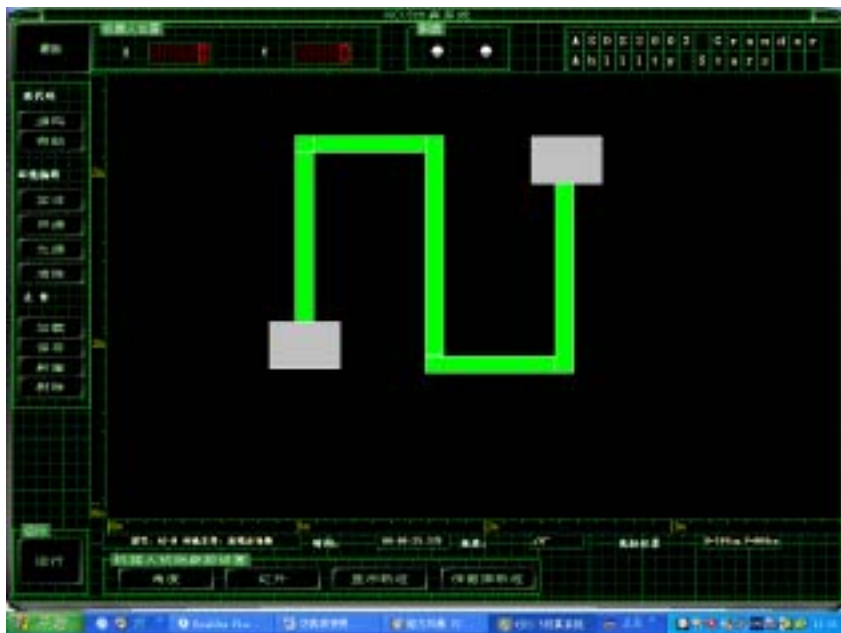


图3 - 32 沿线走场地

#### ◆ 程序分析

在设计沿线走程序之前，先检测一下地面灰度（参见图 3 - 22），灰度参考值为：灰色的矩形框 63，绿色的线条 170。

机器人从起点走到终点，可以分三步进行：第一步出发，第二步循线走，第三步到达。主程序如图 3 - 33 所示。

第一步出发，即从左边的灰色的矩形框走到绿色的线条上。程序如图 3 - 33 所示，首先检测一下出发点的地面灰度，然后进入条件循环。条件循环的条件表达式为

地面变量一 == 63
-------------

机器人边前进边检测。由于绿色线条的灰度值为 170，如果检测到绿色线条，条件循环表达式不再满足，机器人就跳出条件循环，开始循线走。

第二步循线走，机器人沿着绿色的线条向前走，一直走到终点为止，这就是主程序中条件循环的内容，其循环体即为“循线走”子程序（参见图 3 - 34）。条件循环的条件表达式为

$$\text{地面变量一!} = 63$$

此表达式满足就继续循线走，否则就跳出循环。故机器人将反复执行循线走的指令，直至到达终点、检测到灰色的矩形框为止。

“循线走”子程序是根据绿色线条的特点来设计的。由于绿色线条均为直线，且转角均为直角，故机器人循线向前走的时候，绿色线条必定在机器人的前方、左方或右方，且三者必居其一。我们让机器人一边走一边进行地面检测，如果检测到地面灰度值为 170，就继续前进并检测。如果检测到地面灰度值不等于 170，说明前方已经不是绿色的线条了，就左转 90 度并检测地面灰度，如果检测到地面灰度值还不等于 170，说明左方也没有绿色的线条，就右转 180 度并检测地面灰度。一旦检测到地面灰度等于 170，说明发现绿色线条了，就继续向前走。

第三步到达，机器人如果检测到地面灰度为 63，说明已经到达终点了，就前进一段距离，进入灰色的矩形框，唱一首歌曲，程序结束。参见图 3 - 35。

#### ◆ 参数设置

主程序中的“条件循环”模块：条件表达式为

$$\text{地面变量一!} = 63$$

出发子程序中的“条件循环”模块：条件表达式为

$$\text{地面变量一} = = 63$$

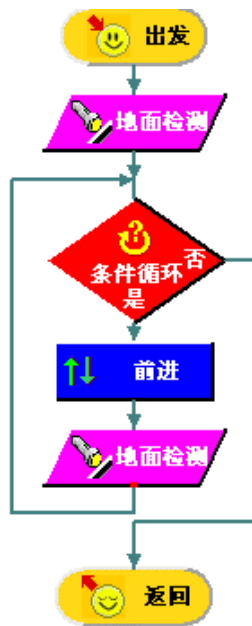
循线走子程序中的“条件判断”模块：条件表达式为

$$\text{地面变量一} = = 170$$

左转 90 度，右转 180 度，通过调整速度或时间参数实现。

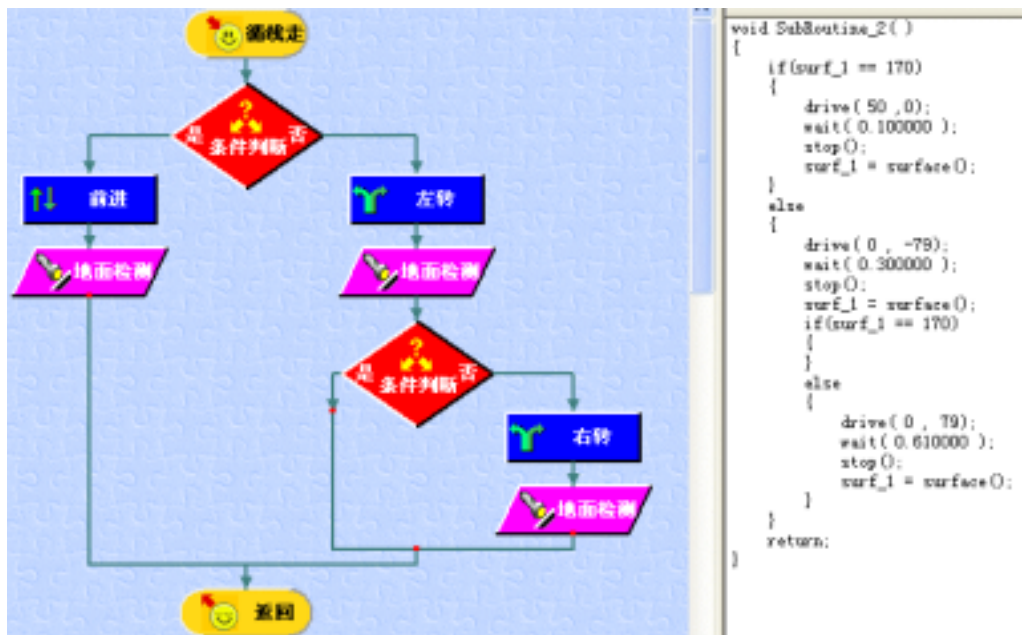


主程序



子程序“出发”

图3 - 33 沿线走例程



```

void Subroutine_2()
{
  if(surf_1 == 170)
  {
    drive(50,0);
    wait(0.100000);
    stop();
    surf_1 = surface();
  }
  else
  {
    drive(0,-79);
    wait(0.300000);
    stop();
    surf_1 = surface();
    if(surf_1 == 170)
    {
    }
    else
    {
      drive(0,79);
      wait(0.610000);
      stop();
      surf_1 = surface();
    }
  }
  return;
}
  
```

图3 - 34 子程序“循线走”



图3 - 35 子程序“到达”

#### ◆ 仿真运行

进入仿真界面后，首先创建沿线走的环境，然后将机器人放在起点即可。

### 3.14 越障跑

项目内容：在跑道上有一些障碍物，机器人的任务是听到发令枪后，以最快的速度从起点跑到终点，途中要避免障碍物。



图3 - 36 越障跑示意图

相关模块：转向、直行、红外测障、条件判断、地面检测、永远循环。

难度等级：

项目解析：这个项目与“短跑比赛”类似，但增加了难度，障碍物检测可用“红外测障”模块，也可用“碰撞检测”模块。跑道可以自行设计。

### 3.15 太阳能电池

项目内容：用机器人模拟太阳能电池。当有光照着机器人的时候，机器人的眼睛就亮起来，就象太阳能电池点亮灯泡一样。

相关模块：亮度检测、眼睛、永远循环、条件判断。

难度等级：

仿真运行：在机器人运行场地中间，添加光源，光源半径 180，渐变值 30。然后将机器人放在光圈中即可。



## 3.16 电子琴

项目内容：用机器人模拟电子琴，当前后左右四个方向发生碰撞时，能发出不同的音符。

相关模块：直行、转向、碰撞检测、永远循环、条件判断。

难度等级：

创建环境 在机器人运行场地中设置一些障碍物。

## 3.17 穿越隧道

项目内容：用机器人模拟火车穿隧道的情景。进入隧道时减速，同时打开照明灯。出隧道后加速，并发出长鸣。

相关模块：直行、亮度检测、永远循环、条件判断。

难度等级：

仿真运行：创建环境 隧道，如下图所示：

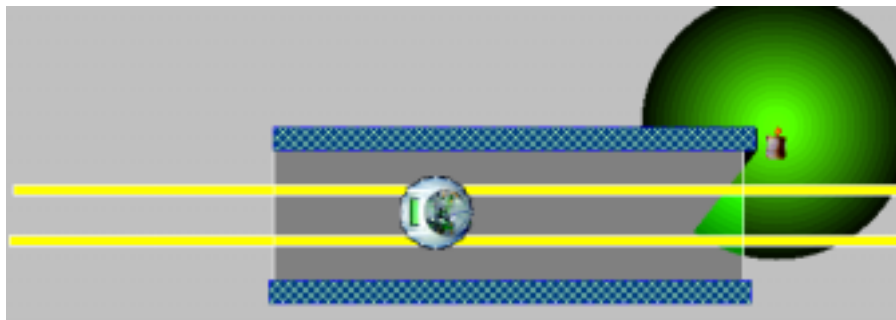


图3 - 37 穿越隧道场地

### 3.18 舞台

项目内容：用彩色图带设计一个舞台，机器人在上面载歌载舞，但不会跑到外面去。

相关模块：地面检测、转向、直行、永远循环、条件判断、发音。

难度等级：

创建环境 舞台。

### 3.19 哈雷彗星

项目内容：在机器人运行场地上，添加一个光源，机器人从远方过来，绕着光源转一个弧度，然后又向远方走去，就象彗星经过地球一样。如图 3 - 38 所示：

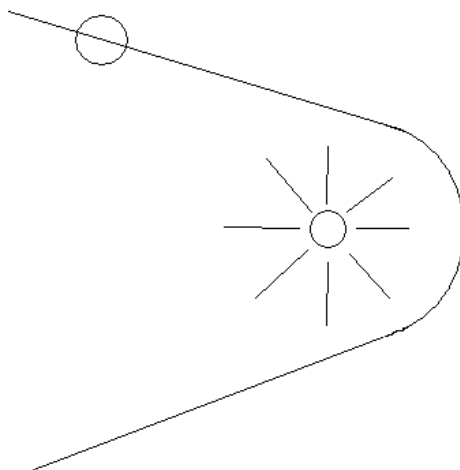


图3 - 38 哈雷彗星示意图

相关模块：亮度检测、条件判断、直行、转向、启动电机、延时等待、停止电机。

难度等级：

仿真运行：在机器人运行场地中间，添加光源。将机器人放在合适的位置，即可运行。

## 第4章 挑战自我

在本章中，我们向大家介绍一些综合性项目。每个项目可能涉及到多个传感器，或需要进行比较复杂的行为控制。机器人也能象人一样，有感觉，会思考问题并解决问题，不信您就试试吧。读者可以先学习范例，然后自主设计其余项目的解决方案。在做完本章的项目之后，您就会对机器人的智能有初步的了解。



## 4.1 填字游戏

项目内容：在机器人运行场地上，有若干个颜色各别的圆形图带，分别代表不同的英文字母。请您设计一个程序，开始时让机器人显示一个英文单词，但缺少一个字母。如果将机器人放在代表正确字母的图带上，机器人就会显示出完整的单词，并高兴地唱歌、眼睛放光。如果将机器人放在代表其他字母的图带上，机器人则会发出另一种声音，提示“错误”。

相关模块：地面检测、条件判断、显示、发音。

难度等级：

创建环境 在机器人运行场地上设置若干个圆形图带，按照灰度值从小到大的顺序排列，分别代表英文字母 abcdefg.....如图 4 - 1 所示。

### ◆ 程序分析

- a) 首先编一个地面灰度检测的程序，用以检测圆形图带的灰度。
- b) 其次进入仿真环境主界面，创建填字游戏场地。
- c) 然后编写填字游戏的程序。在例程（图 4 - 2）中，显示的单词是 doctor，缺少字母 c。填字游戏场地中三排圆形图带的灰度值分别为 0、43、63、85、106、127、128、149、170、191、202、213、234、255，依次代表 abcdefghijklmn 等 14 个字母。故程序运行时，将机器人放在第一排第三个图带上，就可以得到正确的结果，显示出完整的单词，并唱歌。而放在其他的图带上则会提示错误。



图4 - 1 填字游戏场地

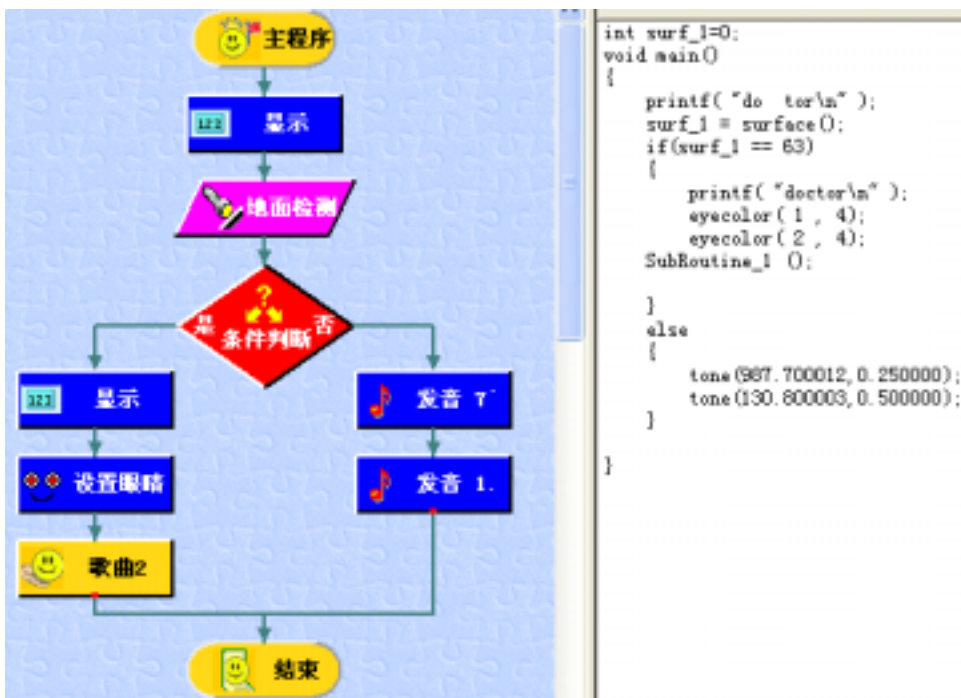


图4 - 2 填字游戏例程

## 4.2 台球

项目内容：在一个空房间里，机器人不停地撞过来撞过去，就象台球一样。房间里可以有一些障碍物。机器人反弹的角度应尽可能合理。

相关模块：碰撞检测、直行、转向、永远循环、条件判断。

难度等级：

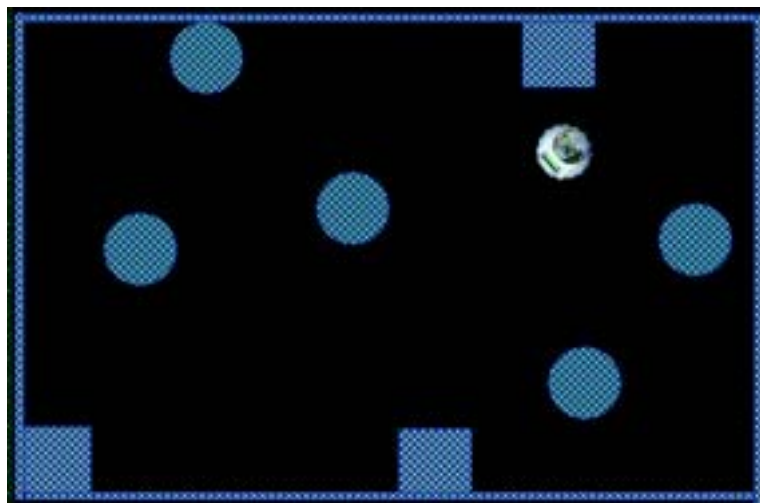


图4 - 3 台球场地示意图

### ◆ 程序分析

要让机器人象台球一样运动，则当前方发生碰撞时，机器人应向后方运动；当后方发生碰撞时，机器人应向前方运动；当左方发生碰撞时，机器人应右转 90 度，然后继续向前运动；当右方发生碰撞时，机器人应左转 90 度，然后继续向前运动；当前后左右都没有发生碰撞时，机器人就保持向前运动的状态。参见图 4 - 4 和图 4 - 5。

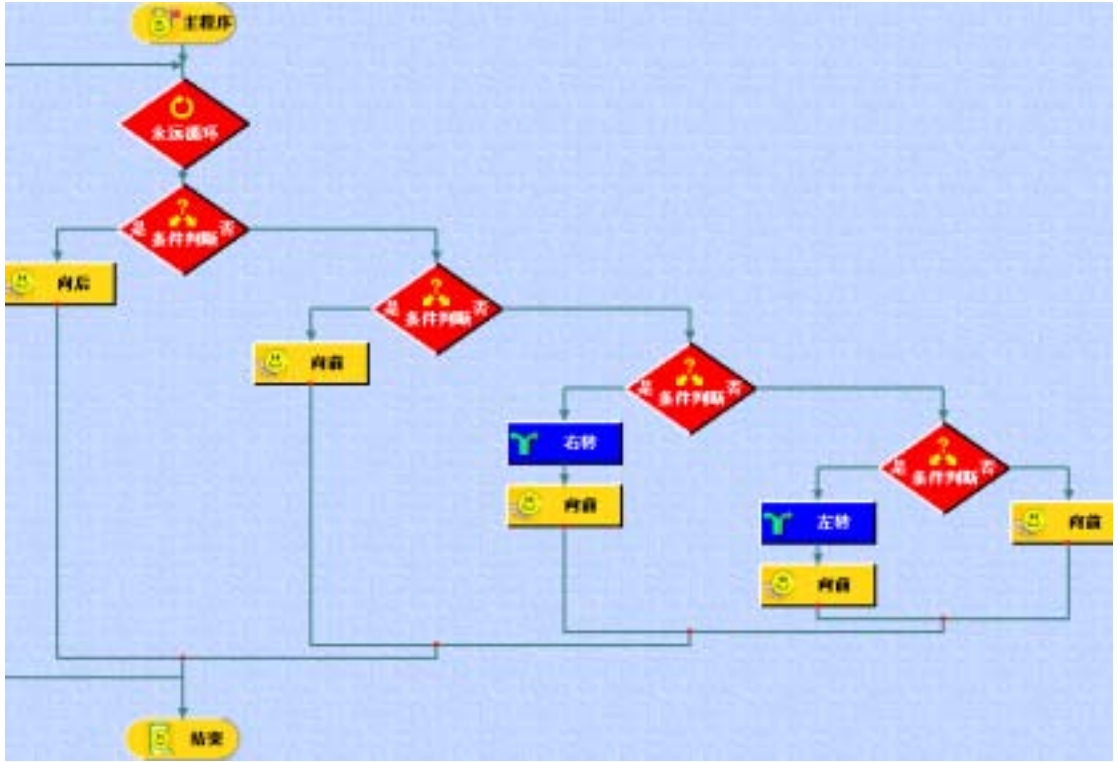


图4 - 4 “台球”主程序



图4 - 5 “台球”子程序

## ◆ 参数设置

### a) 主程序

“条件判断”模块：按照从上至下的顺序，四个条件判断模块的参数设置如下表所示。

条件表达式		
碰撞变量一	= =	前
碰撞变量一	= =	后
碰撞变量一	= =	左
碰撞变量一	= =	右

“左转”模块：速度 80，时间 0.3 秒。

“右转”模块：速度 80，时间 0.3 秒。

### b) “向前”子程序

“前进”模块：速度 100，时间 0.05 秒。

“碰撞检测”模块：选择变量“碰撞变量一”。

“条件循环”模块：条件表达式为

碰撞变量一	= =	无
-------	-----	---

### c) “向后”子程序

“后退”模块：速度-100，时间 0.05 秒。

“碰撞检测”模块：选择变量“碰撞变量一”。

“条件循环”模块：条件表达式为

碰撞变量一	= =	无
-------	-----	---

## ◆ 仿真运行

仿真运行时，创建一个单房间场地，里面放一些障碍物，如图 4 - 3 所示。然后将机器人放到房间里，机器人就会象台球一样运动起来。



## 4.3 寻宝

项目内容：在一个空旷的山洞里，地中埋藏着一个宝物（以彩色圆形图带表示）。机器人从入口处进去寻找宝物。如果找到了宝物，就从出口处出来，否则就一直在里面寻找，出口处有灯光标志。

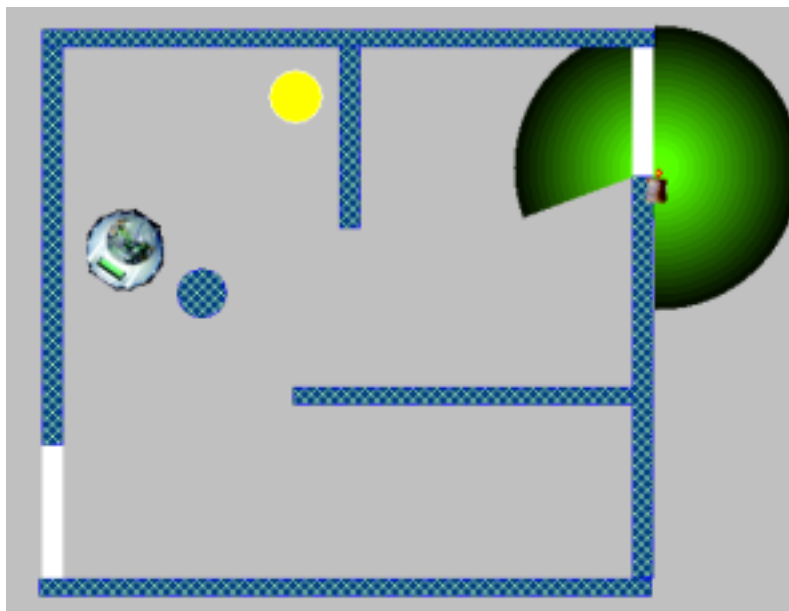


图4 - 6 寻宝场地

相关模块：红外测障、地面检测、条件判断、发音、永远循环。

难度等级：

创建环境 设计一个山洞，地中埋藏着一个宝物，如图 4 - 6 所示。

## 4.4 篝火晚会

项目内容：在本项目中，机器人参加一个篝火晚会，围绕着光源边唱边跳。机器人距离光源不能太远、也不能太近。参见图 4 - 7。

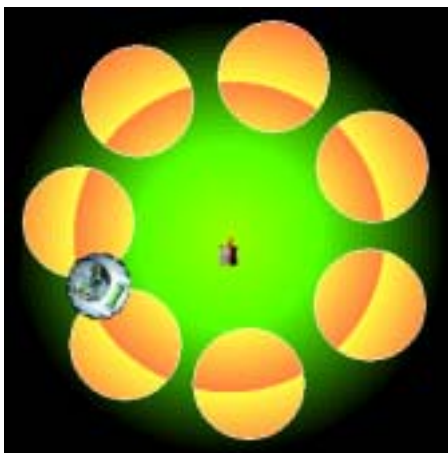


图4 - 7 篝火晚会场地

相关模块：亮度检测、直行、转向、条件判断、永远循环。

难度等级：

### ◆ 程序分析

在本例中，我们让机器人绕着光源顺时针转圈。为了使机器人跟光源保持合适的距离，需要反复检测右眼的亮度，以便及时调整机器人的位置。如果检测到的亮度满足

50	亮度变量一	120
----	-------	-----

机器人就环绕光源运动。如果亮度变量一小于 50，说明离光源太近了，机器人就左转并前进，以远离光源一点。如果亮度变量一大于 120，说明离光源太远了，机器人就右转并前进，以靠近光源一点。调整到合适的位置就唱一段歌曲。参见图 4 - 8 的例程。

### ◆ 参数设置

#### A. 主程序

“亮度检测”模块：方式“右”，变量选择“亮度变量一”。

“显示”模块：显示信息“亮度变量一”。

“条件判断”模块：条件表达式为

条件一：亮度变量一 < 50，条件二：有效，亮度变量一 > 120，条件逻辑关系：与。

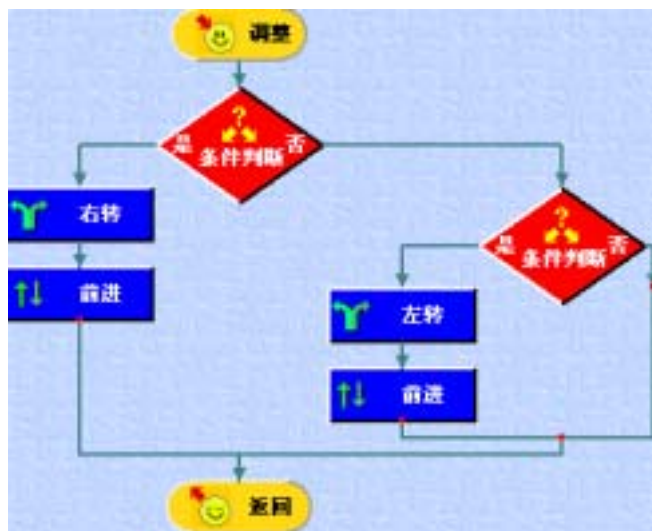
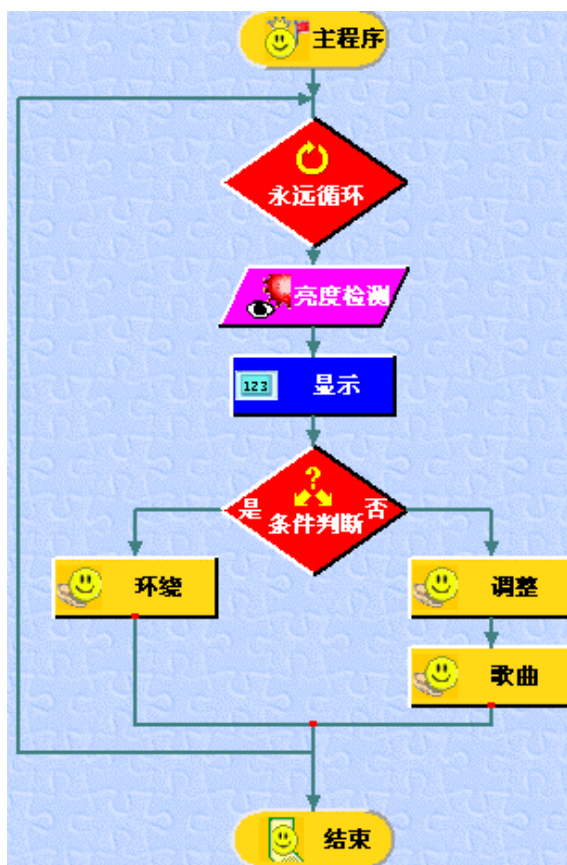


图4 - 8 篝火晚会主程序及其子程序

### B. 子程序“环绕”

“启动电机”模块：左电机功率 100，右电机功率 80。

“延时等待”模块：时间 0.1 秒。

### C. 子程序“调整”

“条件判断”模块：条件表达式如下表所示

参数 位置	条件表达式		
	上	亮度变量一	>
下	亮度变量一	<	50

“左转”模块：速度 20，时间 0.1 秒。

“右转”模块：速度 20，时间 0.1 秒。

“前进”模块：速度 40，时间 0.1 秒。

### D. 子程序“歌曲”

“亮度检测”模块：方式“右”，变量选择“亮度变量一”。

“条件判断”模块：条件表达式为

条件一：亮度变量一 > 50，条件二：有效，亮度变量一 < 120，条件逻辑关系：与。

注：以上参数均可以根据实际情况进行调整。

### ◆ 仿真运行

仿真运行时，在机器人运行场地中添加光源。光源半径设置为 180，渐变值设置为 30。然后将机器人放在光圈中间，机器人就会环绕着光源载歌载舞了。

## 4.5 弹球游戏

项目内容：在一个房间里有若干柱子（圆形障碍物），房间门口用白色图带表示。机器人在房间里运动，象皮球一样弹来弹去，检测到柱子下面的圆形彩色图带可得分（红色 10 分，黄色 8 分，蓝色 6 分，绿色 4 分，紫色 2 分）。看机器人在弹出房间以前能得几分，并且将分数显示出来。

相关模块：碰撞检测、直行、转向、计算、显示、条件循环、地面检测。

难度等级：

创建环境 设计一个空房间，里面有几个圆形障碍物，放在圆形彩色图带上。如图 4 - 9 所示。

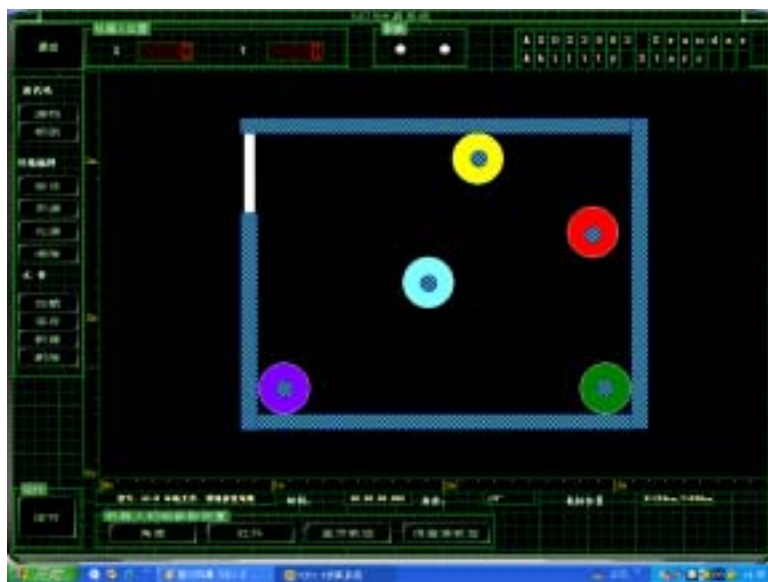


图4 - 9 弹球游戏场地

## 4.6 随机走

项目内容：机器人随机行走，有时前进，有时后退，有时左转，有时右转。行走的距离和转弯的角度要有变化，强调随机性。

相关模块：直行、转向、计算、永远循环、条件判断、碰撞检测、红外测障。

难度等级：

### ◆ 程序分析

要实现随机行走，须考虑以下几个方面：

- a) 避免碰撞。机器人在行走的过程中，如果发生了碰撞，则后退一点，并掉转方向。
- b) 红外避障。机器人边行走边进行红外探测，如果发现左、右、前任一方向有障碍物，就转向。
- c) 前进。设置前进的参数时，将时间取为 0.05 秒或 0.1 秒，可以使红外探测和碰撞检测的周期较短，从而令机器人具有较高的灵敏度。
- d) 优先级。优先级的设置为：避免碰撞的优先级最高；其次红外避障；前进的优先级最低，如果没有发生碰撞，也没有发现障碍物，才前进。
- e) 随机性。随机性体现在行走的距离和方向上，距离应有时长有时短，方向应忽左忽右、忽前忽后。以方向的随机性为例，我们可以通过“计算”模块，计算机器人转向的次数。如果右转了 3 次，就左转一次；如果左转了 4 次，就右转一次；如果前进了 5 次，就后退一次，等等。
- f) 机器人的运行环境。可以加载环境 迷宫、单房间场地或标准灭火场地等（参见帮助电子文档/环境文件管理）。也可以自己创建一个环境。

## 4.7 沿墙走

项目内容：机器人沿着墙壁行走（采用左手规则或者右手规则）。

相关模块：启动电机、延时等待、停止电机、红外测障、碰撞检测、直行、转向、永远循环。

难度等级：

### ◆ 程序分析

沿墙走，顾名思义，即机器人靠近墙壁行走。采用左手规则时，墙壁在机器人的左边；而采用右手规则时，墙壁在机器人的右边。下面以左手规则为例加以说明。



图4 - 10 沿墙走示意图

如图 4 - 10 所示，要实现沿墙走，机器人可以采取划弧线的方法，边前进边检测墙壁。机器人如果发现前方有障碍物，说明正对着墙壁了，就右转约 90 度；如果发现左方有障碍

物，说明机器人左侧对着墙壁了，就右转约 60 度；否则就划弧线前进。有时候机器人会撞上墙壁，为此机器人还需具有处理碰撞的功能。在程序中，我们让机器人一旦检测到碰撞，就后退一点，并右转约 60 度。沿墙走的例程如图 4 - 11 所示。

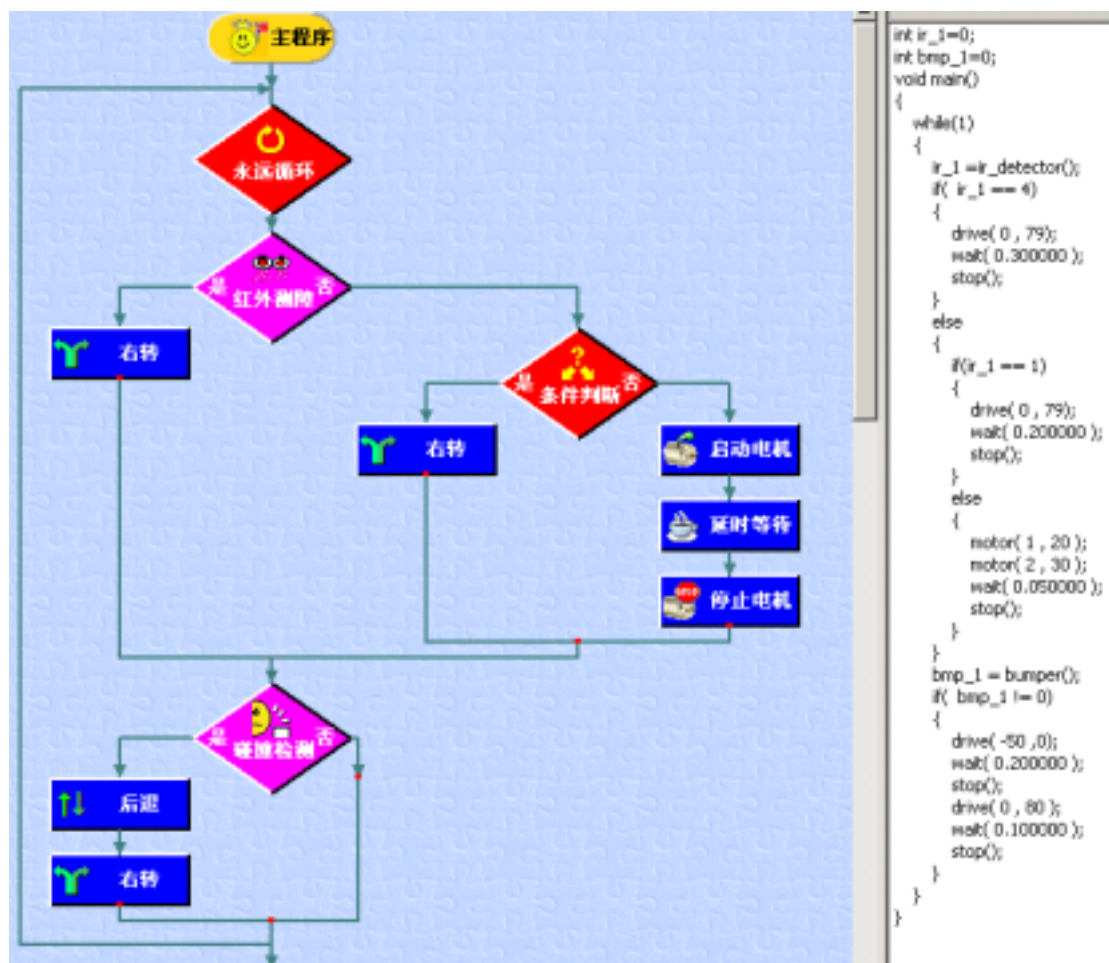


图4 - 11 沿墙走例程

### ◆ 参数设置

“红外测障”模块：检测完成后，进行条件判断，条件表达式为

红外变量1 == 左

“条件判断”模块：条件表达式为

红外变量1 == 前

“启动电机”模块：左电机功率 20，右电机功率 30。



“延时等待”模块：时间 0.05 秒。

“碰撞检测”模块：条件表达式为

碰撞变量！= 无

其余模块的参数自行设置，并需反复调整，直至满意为止。

#### ◆ 仿真运行

仿真运行时，加载单房间场地、标准灭火场地、走迷宫场地均可。读者也可以自己创建一个场地，让机器人在其中运行。

## 4.8 点歌台

项目内容：用机器人做一个点歌台，当前、后、左、右四个方向发生碰撞时，能分别唱不同的歌曲。歌曲以子程序的形式调用。

相关模块：直行、转向、碰撞检测、永远循环、条件判断、子程序。

难度等级：

创建环境 在机器人运行场地中设置一些障碍物。

## 4.9 回音壁

项目内容：机器人根据检测到的声音强度，发出相近的声音。比如检测到声音强度为 180，就发音 4（音频为 174.6）。

相关模块：声音检测、条件判断、发音、永远循环。

难度等级：

仿真运行：在机器人运行场地中添加声源。

## 4.10 查找最大值

项目内容：给出一组数，让机器人从中找出最大值，并显示出来。

相关模块：计算、条件判断、条件循环、显示。

难度等级：

项目解析：“查找最大值”项目需要设计算法。下面给出了一个简单的例程，你也可以设计自己的算法。还可以自己设计类似的项目，然后自己编程实现。

本例中给出了六个整型数 5、8、-12、3、64、81，由六个计算模块分别赋给整型变量一、整型变量二、整型变量三、亮度变量一、亮度变量二、亮度变量三（见图 4 - 12 中的前六个蓝色矩形“计算”模块）。查找最大值的过程实际上是条件判断的过程（见图 4 - 12 中的红色菱形“条件判断”模块），我们选用整型变量一与其它变量逐一进行比较，如果整型变量一小于与之相比较的变量，就将后者赋给整型变量一（见图 4 - 12 中条件判断下的“计算”模块），否则整型变量一保持不变。最后得到的整型变量一就是最大值，将它显示出来就可以了（见图 4 - 12 中最后的“显示”模块）。对照图 4 - 12 流程图右边的 JC 代码可以更清楚地理解上述过程。

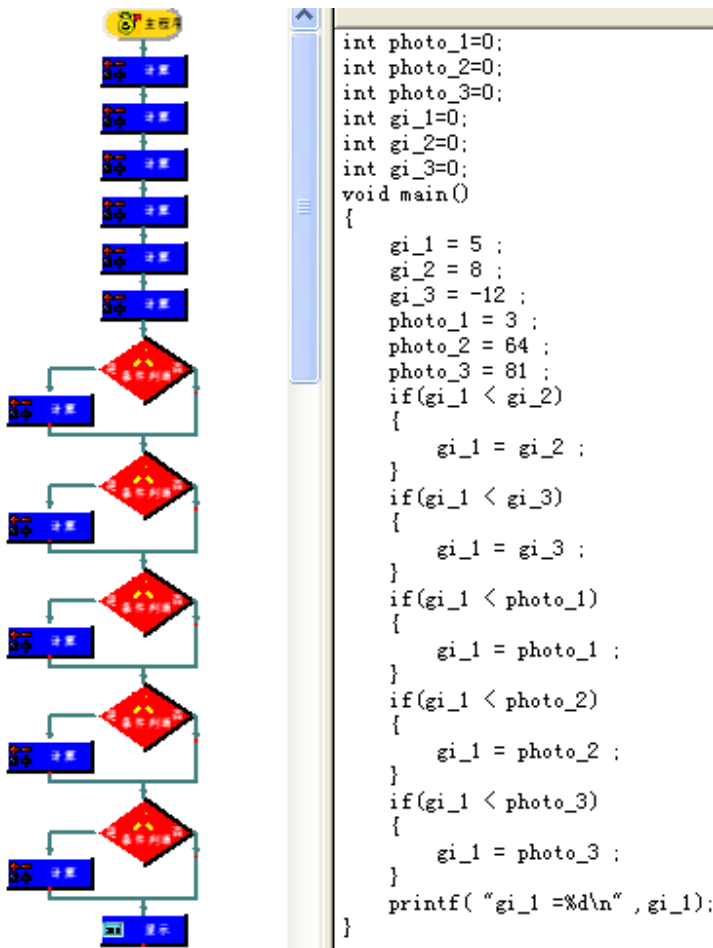


图4 - 12 查找最大值例程

## 4.11 排序

项目内容：给出一组数（例如 3 个整数），让机器人将其从小到大排列，并依次显示出来。

相关模块：计算、条件判断、条件循环、显示。

难度等级：

项目解析：“排序”是计算机数据结构中重要的内容，通过 VJC1.5 仿真版能完成较简单的数据排序，通过“排序”项目，能锻炼自己程序设计能力。

## 4.12 最大公约数

项目内容：求两个正整数的最大公约数，并显示出来。

相关模块：计算、条件判断、条件循环、显示。

难度等级：

### ◆ 程序分析

求两个正整数的最大公约数，可用辗转相除法（参见后面的小知识）。编程计算时，须注意到 C 语言中整型数的特点，两个整型数相除得到的商仍为整型数，小数部分被忽略。参考程序如如图 4 - 13 所示。

### ◆ 参数设置

条件循环之前的四个“计算”模块：计算表达式如下表所示

整型变量一	=	255 (被除数)
整型变量二	=	60 (除数)
整型变量三	=	整型变量一 ÷ 整型变量二 (得到商)
整型变量三	=	整型变量一 - 整型变量二 × 整型变量三 (得到余数)

“条件循环”模块：条件表达式为

整型变量三 != 0
------------

循环体中的四个“计算”模块：计算表达式为

整型变量一	=	整型变量二 (除数升级为被除数)
整型变量二	=	整型变量三 (余数升级为除数)
整型变量三	=	整型变量一 ÷ 整型变量二 (得到商)
整型变量三	=	整型变量一 - 整型变量二 × 整型变量三 (得到余数)

“显示”模块：显示信息为“整型变量二”。

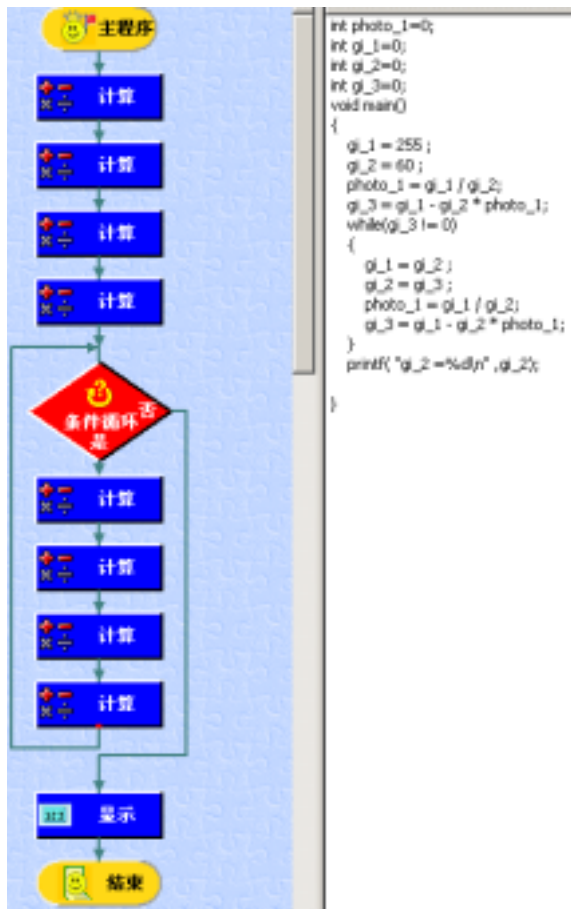


图4 - 13 求最大公约数例程

小知识：最大公约数和辗转相除法

两个正整数的最大公约数，是指能整除这两个整数的最大的整数。例如：18 和 45 的最大公约数是 9。

求最大公约数的标准方法是辗转相除法。设有两个整数  $A_0$  和  $A_1$ ，且  $A_0 \geq A_1$ ，它们的最大公约数为  $d$ ，用辗转相除法求最大公约数  $d$  的算法如下表所示：

表达式 步骤	被除数	=	除数	×	商	+	余数
1	$A_0$	=	$A_1$	×	$b_1$	+	$A_2$
2	$A_1$	=	$A_2$	×	$b_2$	+	$A_3$
3	$A_2$	=	$A_3$	×	$b_3$	+	$A_4$
$n-2$	$A_{n-3}$	=	$A_{n-2}$	×	$b_{n-2}$	+	$A_{n-1}$
$n-1$	$A_{n-2}$	=	$A_{n-1}$	×	$b_{n-1}$	+	$A_n$
$n$	$A_{n-1}$	=	$A_n$	×	$b_n$	+	0

第 1 步，以  $A_0$  作被除数， $A_1$  作为除数，得到的商为  $b_1$ ，余数为  $A_2$ 。第 2 步，将  $A_1$  升级为被除数，将  $A_2$  升级为除数，得到商为  $b_2$ ，余数为  $A_3$ 。其余的步骤依此类推。从第 1 步开始，在每一步的表达式中，由于被除数和除数都能被  $d$  整除，故余数（= 被除数 - 除数 × 商）也能被  $d$  整除。

在上述过程中，由于除数小于被除数，余数又小于除数，故余数越变越小。又由于余数不能小于  $d$ ，否则会导致矛盾，于是最后必定可以得到余数为零，如下式所示：

$$A_{n-1} = A_n \times b_n + 0$$

此即为表中第  $n$  步的式子，该式中的除数  $A_n$  就是所要求的最大公约数  $d$ ，即

$$d = A_n$$

【证明】 由于第  $n$  步中  $A_n$  整除  $A_{n-1}$ ，从表格倒推上去可知， $A_n$  也能整除

$$A_{n-2}、A_{n-3}、\dots、A_2、A_1、A_0$$

故  $A_n$  是  $A_0$  和  $A_1$  的一个公因子, 而  $d$  是  $A_0$  和  $A_1$  的最大公因子, 于是

$$A_n \leq d$$

又如前所述,  $d$  能整除表格中所有的被除数和除数,  $d$  当然也能整除  $A_n$ , 于是

$$d \leq A_n$$

从而

$$d = A_n$$

证毕。

【例】 求 253 和 77 的最大公约数。

步骤如下表所示, 求得的最大公约数  $d = 11$ 。

表达式 序号	被除数	=	除数	×	商	+	余数
1	253	=	77	×	3	+	22
2	77	=	22	×	3	+	11
3	22	=	11	×	2	+	0



## 4.13 跳房子游戏

项目内容：在运行场地中，有一组方格，每个方格代表一个房间。在第一个房间里有声源。开始的时候机器人放在第一个房间里，放置的位置是随机的。机器人自动检测声音的强度，然后进行计算，将声音的强度除以7，求得余数为0~6之间的一个随机数。最后，机器人根据得到的随机数跳到相应的房间里。得到的随机数越大越好。

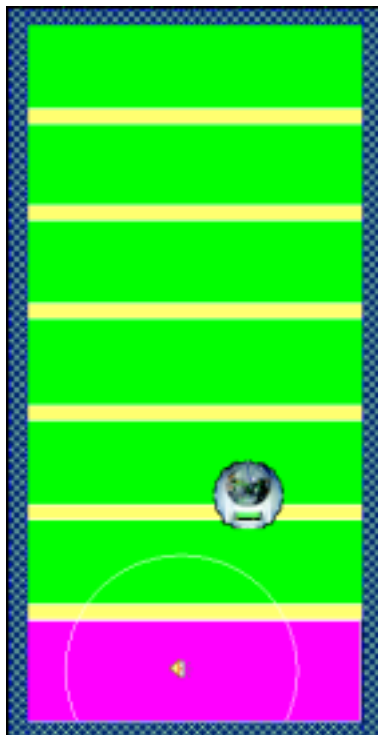


图4 - 14 跳房子场地

相关模块：声音检测、地面检测、转向、直行、条件循环、计算、发音。

难度等级：

项目解析：“跳房子”是小孩子常玩的一种活动，现在我们要在通过计算机来模拟这种游戏。“跳房子游戏”涉及到声音检测、地面灰度检测和计算，项目有点难，但做起来还是很有意思的。

## 4.14 找地雷

项目内容：在机器人运行场地中分布着一些地雷，地雷用红色圆形图带表示。白色圆形图带是出发点，黄色圆形图带是终点。机器人的任务是寻找地雷，找到地雷的时候发出警报。最后，统计所找到的地雷个数，并显示出来。

相关模块：地面检测、直行、转向、发音、计算、条件循环。

难度等级：

项目解析：要完成“找地雷”项目有一定难度，不但要有较好的编程能力，还需要一定的调试能力，更需要耐心和细心和恒心。

创建环境 找地雷场地，参考图如下。设计好之后，保存起来，以便于加载使用。

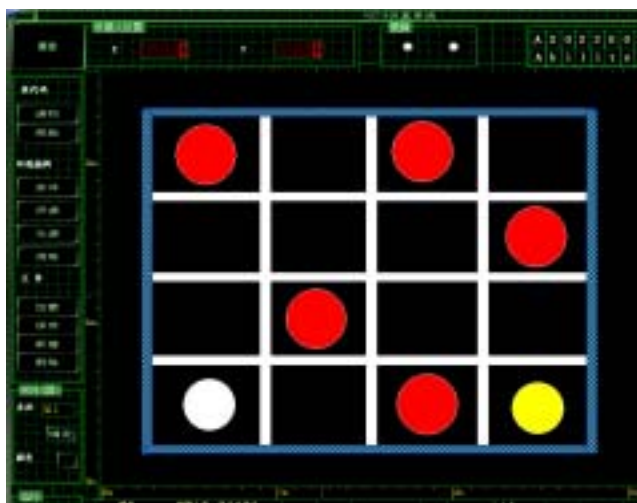


图4 - 15 找地雷场地

## 第5章 我能赢

在本章中，我们向读者介绍几个经典的机器人比赛项目，这些项目都是近年来国内国际机器人比赛经常采用的。在 VJC1.5 仿真版中，由于软件功能的限制，要达到国际赛事的水平是困难的，但读者可以从了解机器人比赛的大概情况，并演练初步的技巧。



## 5.1 走迷宫

**项目内容：**第一步，运用 VJC1.5 仿真版中的障碍物、图带、光源和声源，设计一座迷宫，参见图 5 - 1。第二步，为机器人编制程序，让机器人从迷宫的入口进去，从出口处走出来。在图 5 - 1 中，左下是机器人的入口，右下是机器人的出口。

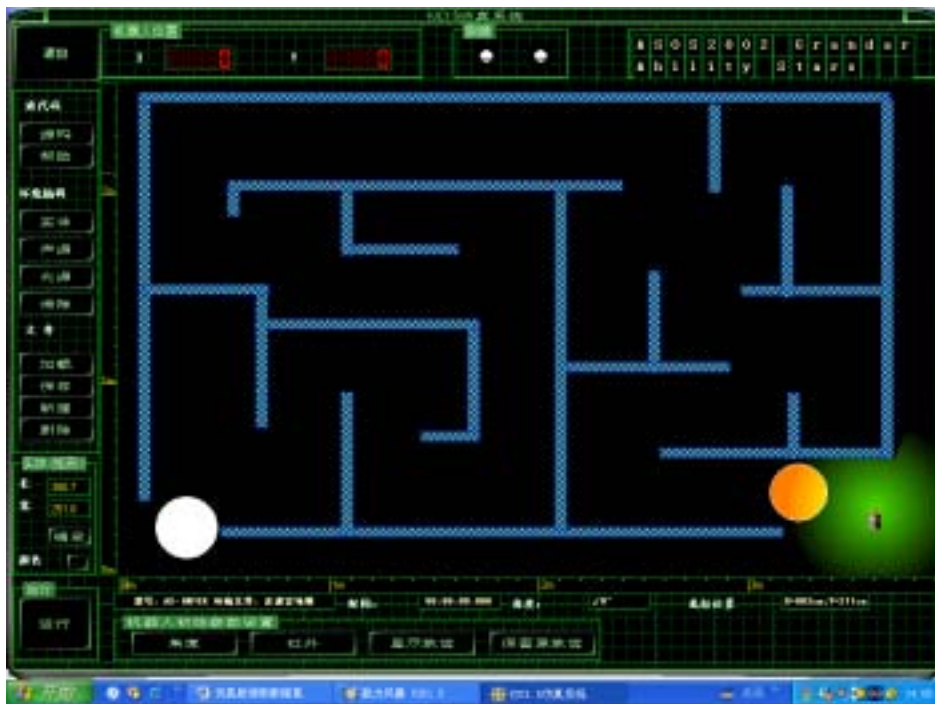


图5 - 1 机器人走迷宫场地

**难度等级：**

本项目可以设立以下奖项：

1. 最佳迷宫奖 该奖项授给迷宫设计最有创意的同学。
2. 最快速度奖 该奖项授给以最少时间走出迷宫的同学。
3. 最佳算法奖 该奖项授给程序设计思想最优秀的同学。

## 5.2 游北京

项目内容：机器人游北京的场地如下图所示。在场地中有十三个景点，机器人每到达一个景点就可以得到 10 分，重复到达的景点不加分。得分最多者获胜。

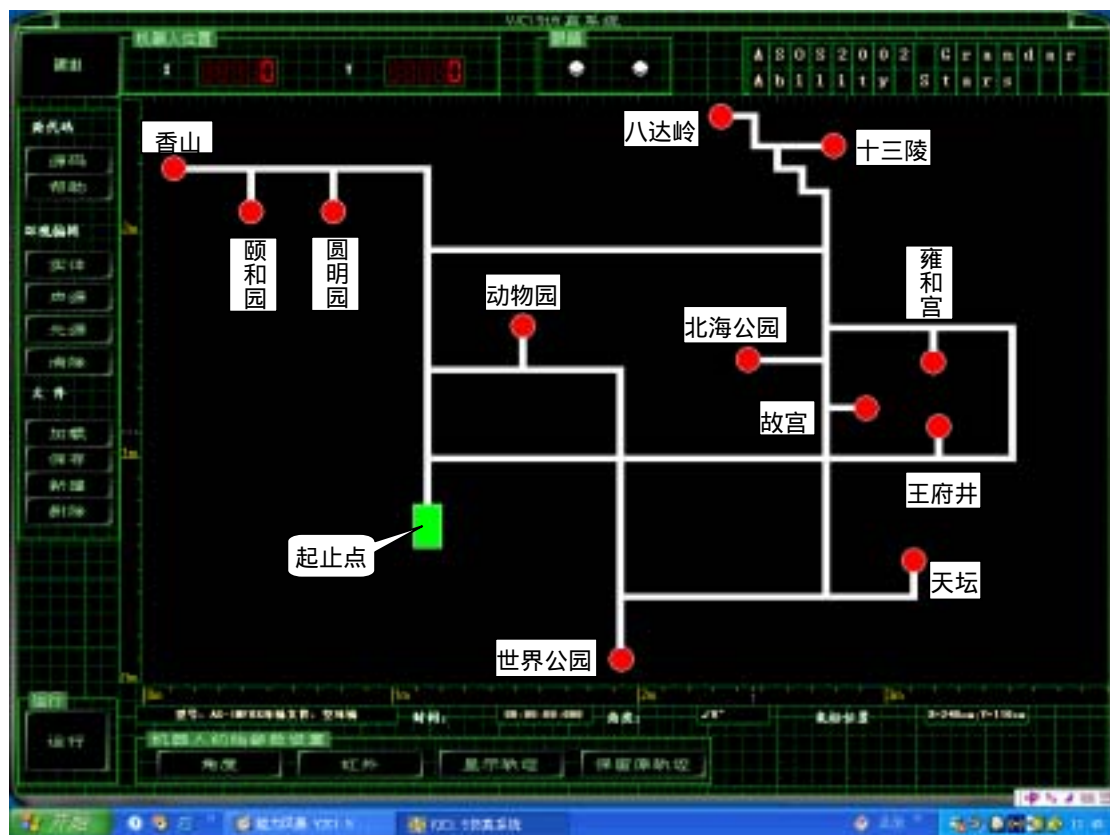


图5 - 2 机器人游北京场地

难度等级：

项目解析：本项目用通过调用子程序可简化程序结构。如觉得项目太难，可按照自己的意图修改场地，比如在交叉路口增加各种颜色的标志。

## 5.3 长跑比赛

项目内容：长跑比赛场地示意图如下，跑道周长为 400 米，要求机器人从起点（桔红色图带）出发沿跑道前进，跑完 1000 米到达终点（绿色图带），显示跑完所花的时间。本项目采用声控启动。

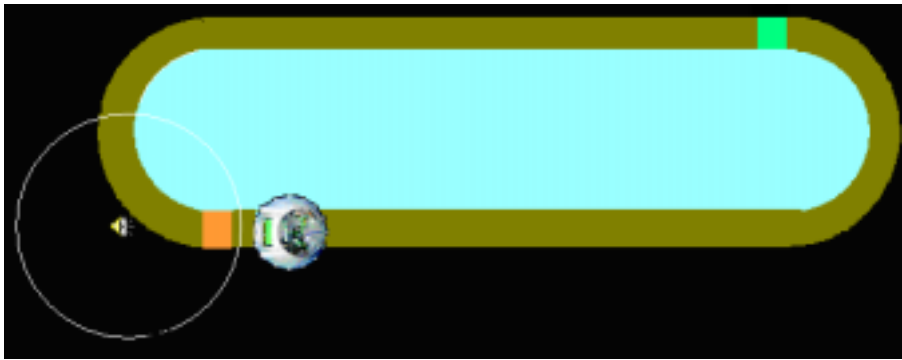


图5 - 3 长跑比赛场地

难度等级：

项目解析：可分别编写 800 米和 1000 米程序，比谁跑的时间最短。

## 5.4 救援

**项目内容：**救援场地示意图如下。这是一个公寓的某一层，有四个房间和一个走廊。房间和走廊由障碍物制成。白色的线条是门，黄色的小圆圈代表人。线条和“人”均由图带制成。

假设发生了突发事件，要求机器人从左边的门进入该楼层，将困在里面的人救出来，最后从右边的门（红色）出来。机器人如果找到人，就叫两声，即可得 10 分。得分最多者获胜。重复发现不加分。

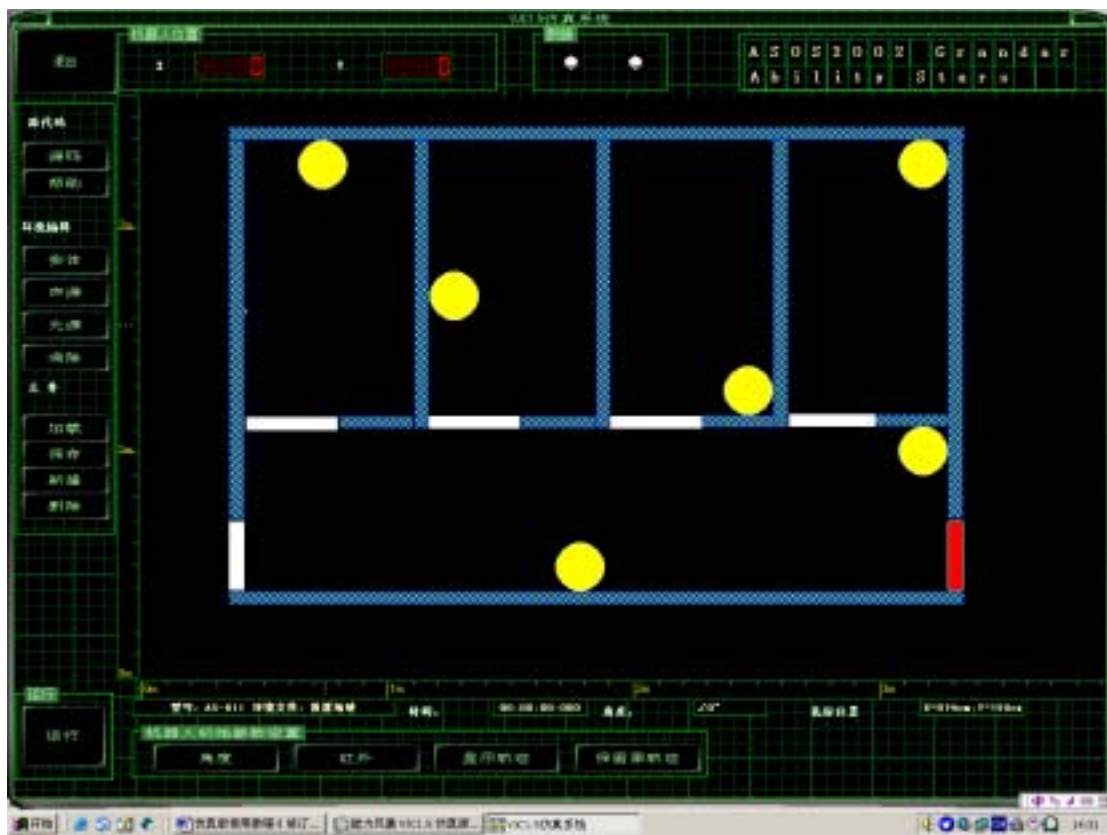


图5 - 4 机器人救援场地

难度等级：

## 5.5 灭火比赛

**项目内容：**灭火场地示意图如下，共有四个房间，其中光源代表火焰，白色的圆圈是出发点，白色的线条是房门。机器人的目标是进入房间去寻找火焰，找到火焰后，启动风扇将火焰吹灭。灭火所用时间最短的机器人获胜。

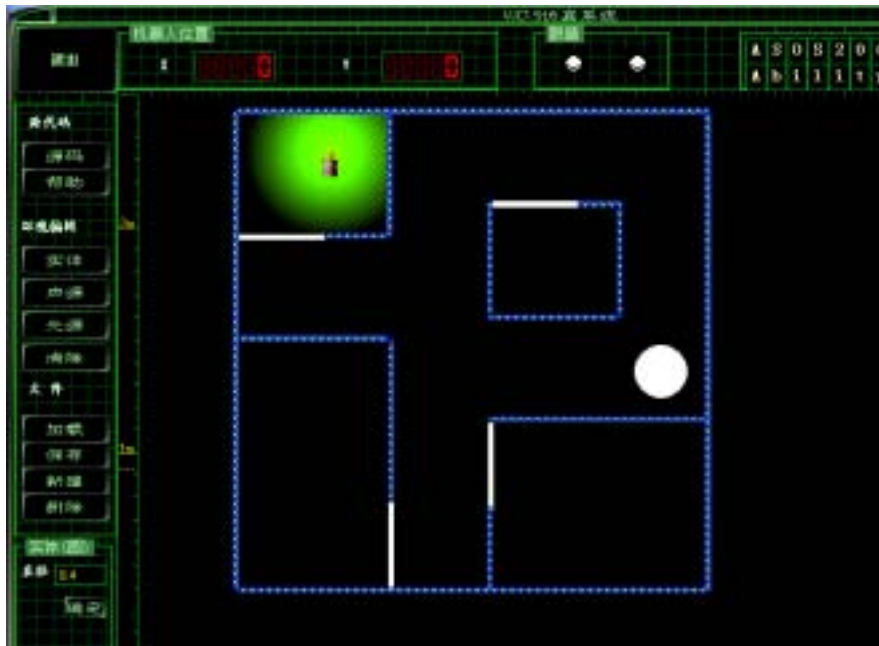


图5 - 5 标准灭火场地

注：建议读者自己创建环境 灭火场地。而不用 VJC1.5 仿真版中的标准灭火场地。创建环境的时候，应使房间和走廊尽可能宽大，以便于机器人活动，墙壁宽度取 6~10。

难度等级：

项目解析：灭火比赛是经典的机器人比赛项目。下面我们分析一下做该项目的思路。

### a) 搜索房间的策略

要完成灭火的工作，机器人首先须设法进入各个房间。搜索房间的策略推荐如下：

- (1) 固定路线法 主要使用“直行”和“转向”模块，机器人走直线，转弯角度均为 90 度。只要反复调试机器人前进的距离，就可以分别进入各个房间。



(2) 随机搜索法 参见 4.6 节“随机走”。

(3) 沿墙走 参见 4.7 节“沿墙走”。

◆ 寻找火焰的方法

参见 3.11 节“趋光的飞蛾”。

◆ 灭火的方法：

机器人在趋光的时候，如果靠近光源了，就停止前进，启动风扇灭火。

◆ 流程图

下面给出机器人灭火程序的流程图，供读者参考。

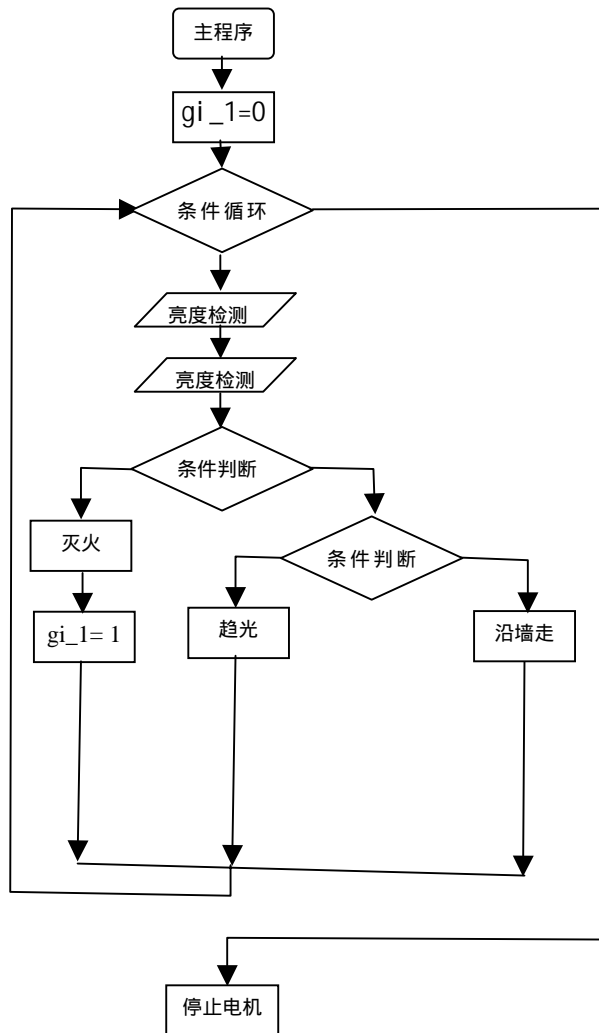


图5 - 6 灭火程序示意图

在图 5 - 6 中，参数可以进行如下设置：

上面一个“亮度检测”模块：方式：左，变量选择“亮度变量一”。

下面一个“亮度检测”模块：方式：右，变量选择“亮度变量二”。

“条件循环”模块：条件表达式为

$$gi\_1 = 0$$

上面一个“条件判断”模块：条件表达式为

条件一：亮度变量一 < 20 ； 条件二： 亮度变量二 < 20；条件逻辑关系：或。

下面一个“条件判断”模块：条件表达式为

条件一：亮度变量一 < 250 ； 条件二： 亮度变量二 < 250；条件逻辑关系：或。

# 附 录

## 附录A. 子 程 序

调用子程序，可以简化主程序的结构。当主程序较长或某个程序段需反复使用的时候，调用子程序就会显得特别方便。

在 VJC1.5 编程软件中，子程序分为以下两种：

### ◆ 系统子程序

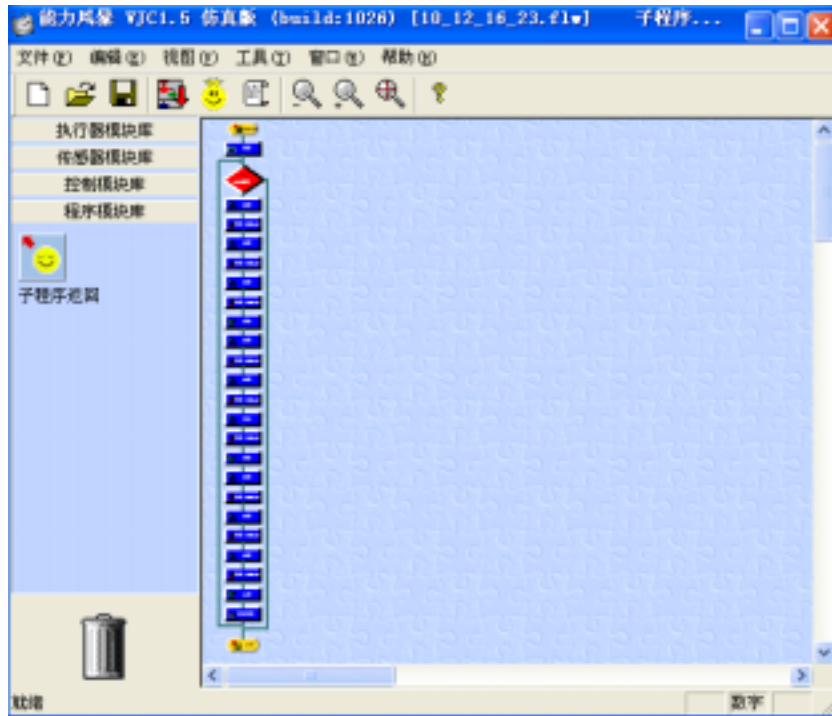
系统子程序是指 VJC1.5 仿真版自带的子程序模板，可直接调用。调用系统子程序的操作方法如下所述：

点击“程序模块库”中的“新建子程序”模块，在弹出的对话框中选择需要的系统子程序模板，比如“边唱边跳”，如附图 1 所示：




附图2 新建子程序对话框

点击“确定”按钮，就可进入该子程序的编辑窗口。在子程序的编辑窗口中，会出现“边唱边跳”子程序的流程图（附图 2），这时你可以根据需要修改子程序流程图。



附图3 子程序的编辑窗口

修改完毕，点击工具栏中的“主程序”快捷按钮，切换到主程序窗口，在程序模块库中就会有刚才修改好的子程序模块“边唱边跳”，如附图 3 所示：



附图4 子程序模块

子程序模块和其它模块一样，可以拖到流程图生成区，用于为机器人编写程序。在编写程序的过程中，如果要对子程序进行修改，可以在子程序模块上右击鼠标，进入编辑状态，编辑完毕再回到主程序窗口即可。

在系统子程序中，还可以使用“自定义”模板，自己编写一个新的子程序。

#### ◆ 其它程序中的子程序

调用其它程序中的子程序，是指打开以前保存的程序文件，调用里面的子程序，操作方法如下：

点击“程序模块库”中的“新建子程序”模块，在弹出的对话框中选择“其它程序”模板，如附图 4 所示：



附图5 新建子程序对话框

此时会弹出一个对话框，如附图 5 所示：



附图6 新建子程序对话框

在对话框中，选择你所需要文件夹以及其中的程序文件，比如文件夹“例程”中的“标准场地灭火示例”，点击“打开”按钮，“标准场地灭火示例”中的子程序就会出现在“新建子程序”目录中，如附图6所示：



附图7 其它程序中的子程序

这时可以和调用系统子程序一样调用这些子程序，如前所述。

## 附录B. 如何设置机器人型号

当程序中含有“碰撞检测”模块的时候，打开程序文件之前，必须先配置机器人型号，否则程序就不能正确地仿真运行。在 VJC1.5 仿真版中，有五种型号的机器人，参见 1.1 节。程序文件原来在何种机器人型号下编写，还须在何种机器人型号下打开。

为 VJC1.5 仿真版配置型号的步骤如下：(这里以 AS-MII 机器人为例，其它型号可以类似配置。)

- (1) 打开流程图编辑界面 (参见图 1 - 5)。
- (2) 单击菜单栏中的“工具 (T)”选项卡，在下拉菜单中单击“设置选项 (O)”，随之会弹出一个设置对话框。
- (3) 在设置对话框中“机器人型号”下拉框中单击“AS-MII”。
- (4) 单击“确定”按钮，退出设置。
- (5) 关闭窗口，回到桌面，重新打开流程图编辑界面，为机器人配置的型号“AS-MII”就生效了。

在 VJC1.5 仿真版中，当程序中不含有“碰撞检测”模块的时候，则不需配置机器人型号。

## 附录C. 不同型号的虚拟机器人碰撞传感器差异

特 征 型 号	方 向 值			
	前	后	左	右
InfoX	4	无	1	2
InfoM	4	8	1	2
AS-M	3	12	5	10
AS-MII	3	12	5	10
AS-UII	3	12	5	10

## 附录D. 真实的能力风暴机器人功能一览表

型号	InfoX	InfoM	AS-M	AS-MII	AS-UII
功能					
红外检测					
亮度检测					
碰撞检测					
声音检测					
转角检测					
地面检测					
系统时间					
电池电量					
眼 睛					
扩展电机					