



LAWRENCE TECHNOLOGICAL UNIVERSITY  
**ROBOFEST**

**常规赛—保龄球赛**

**EV3 编程 EV3机器人**

**教程**



# 工作坊目录

- **常规赛——保龄球赛RoboBowl简介**
- **SPbot介绍**
- **用机器人去解决RoboBowl挑战**



# 常规赛—保龄球赛

- 观看比赛视频  
见网站
- 关键任务
  - 测量矩形宽度
  - 计算障碍瓶3, 4的位置
  - 沿着桌子边的方向前行
  - 确定“发射区”
  - 确定障碍瓶位置
  - 发射网球
  - 返回“基地”

\*请注意抓网球是这个工作坊的范围之外的内容



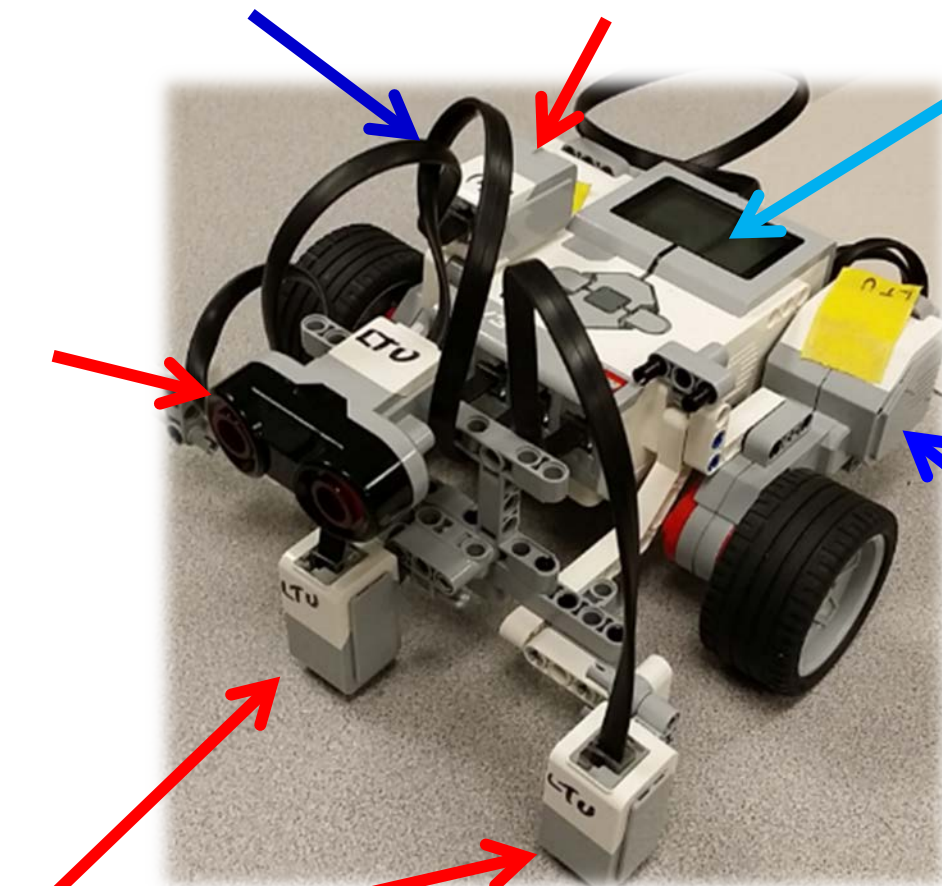
# SPbot

右电机: C

触控传感器

EV3 控制器

声纳传感器



左电机: B

颜色传感器



# 任务 1

**向前移动，当到达矩形时停止**

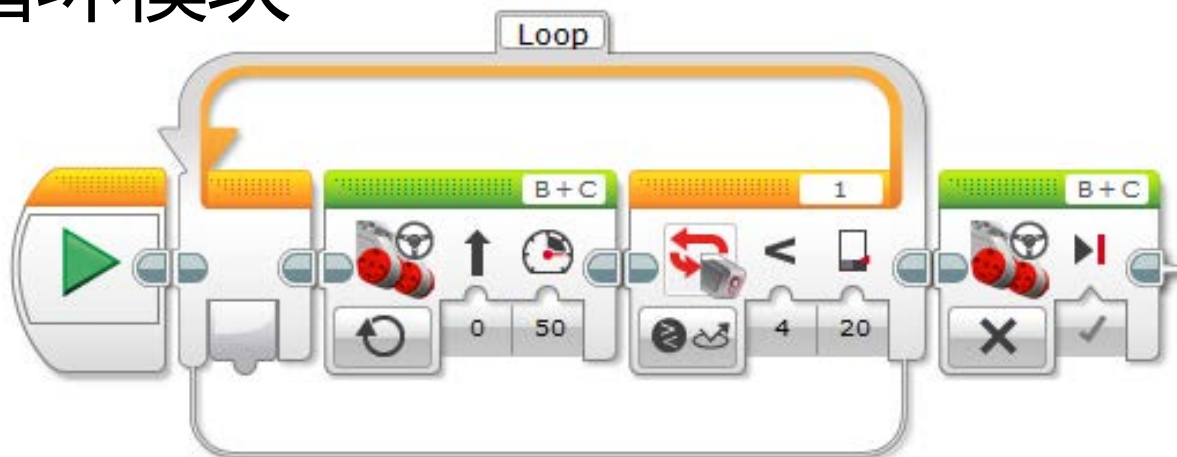


# 任务 1: 方案

- 使用等待模块



- 使用循环模块





## 任务 2

# 测量矩形的宽度

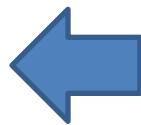
观看视频：见网站



# 测量距离

- 确定机器人前行的距离

通过测量行进中车轮转动的  
数量来计算距离



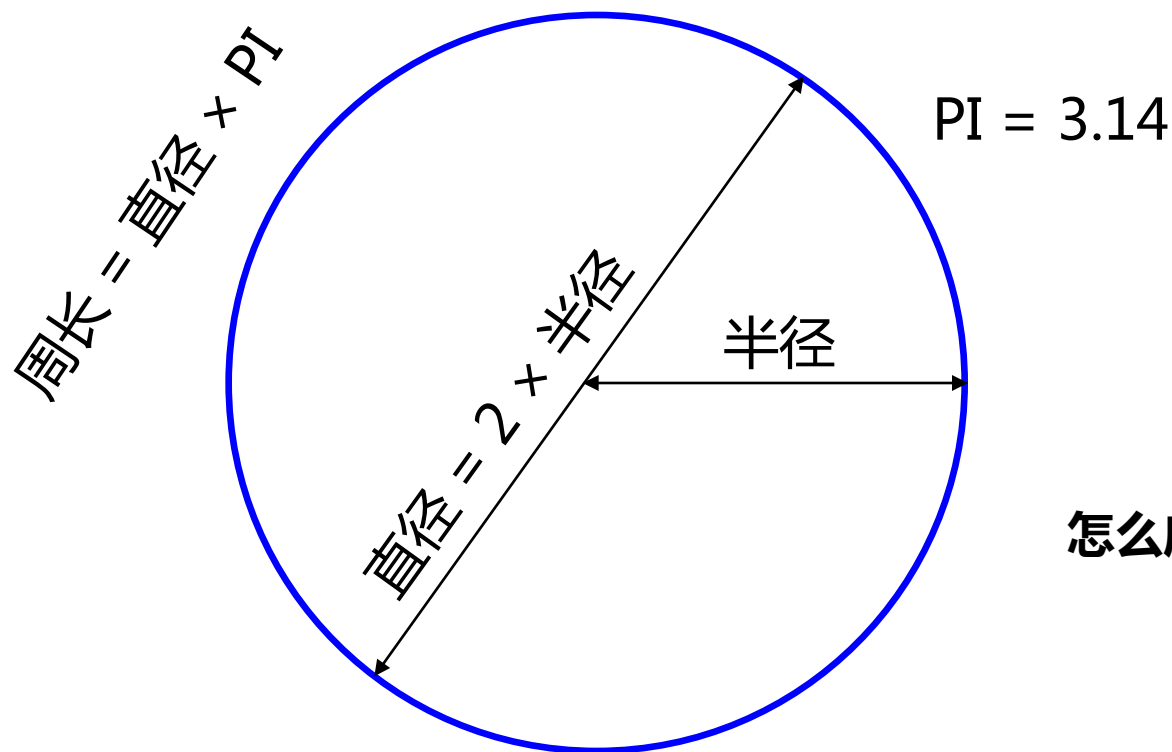
距离





# 测量距离

- 利用车轮的几何形状

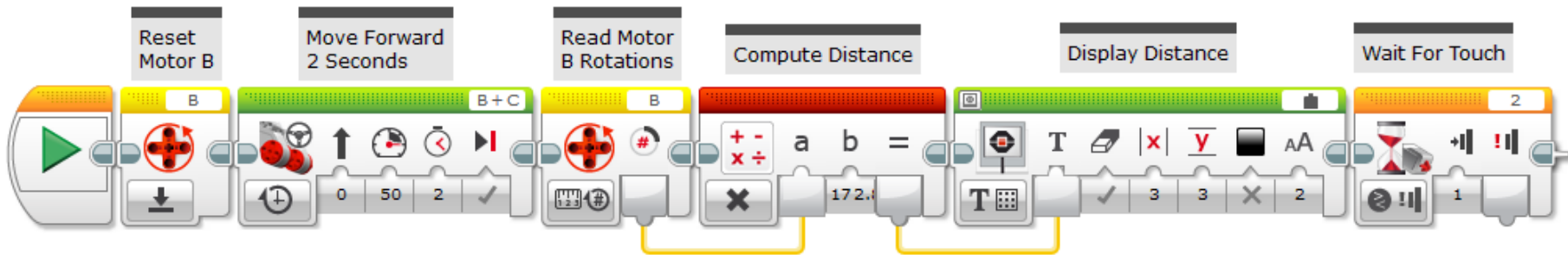


怎么应用这些信息?



# 测量距离

- 车轮每转一圈, 机器人将移动 (车轮直径) x (PI)
- 距离 = (车轮直径) x (PI) x (转动圈数)



程序: MeasureDistance.ev3



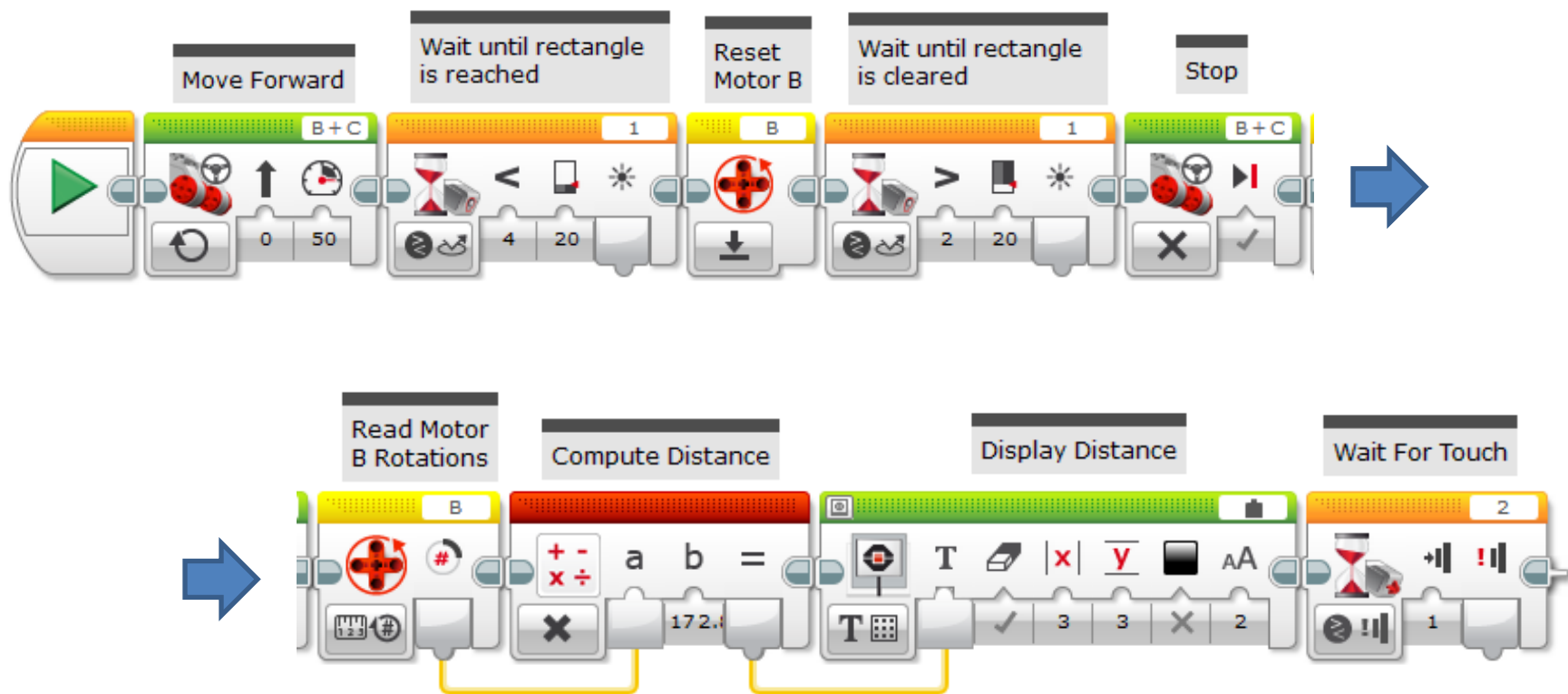
# 测量矩形宽度

- 一种可行方案:
  - 前进，直到发现黑色矩形
  - 复位电机旋转传感器
  - 前进，直到矩形的另一端被发现
  - 读取电机旋转数
  - 计算移动距离



# 测量矩形宽度

- 测量矩形宽度实例



程序: MeasureRectangle.ev3



## 任务 3

计算障碍瓶3, 4的位置



# EV3 数学运算

- 可以运用 数学模块来计算障碍瓶3，4的位置
- 例如，假设

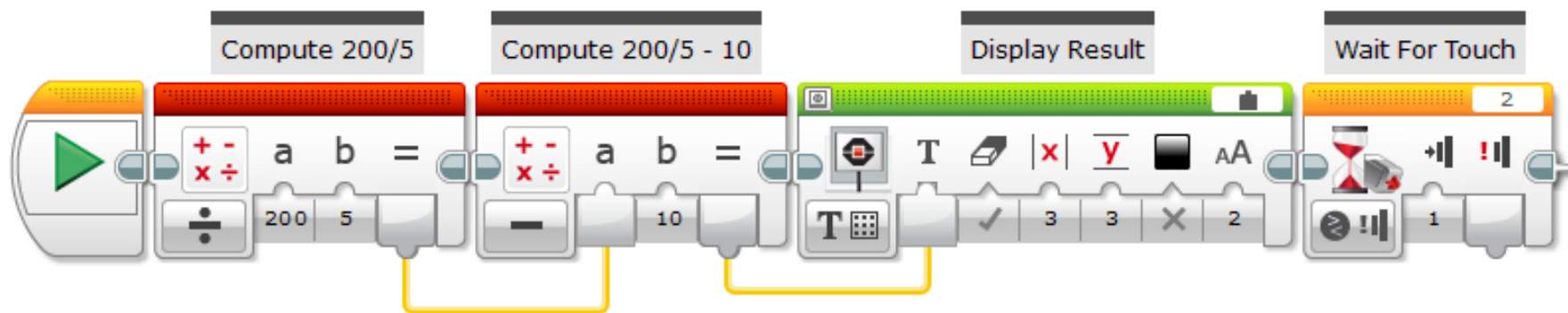
$$d = \frac{x}{5} - 10 \text{ mm}$$

- 怎么用机器人来计算这个呢？



# EV3 数学运算

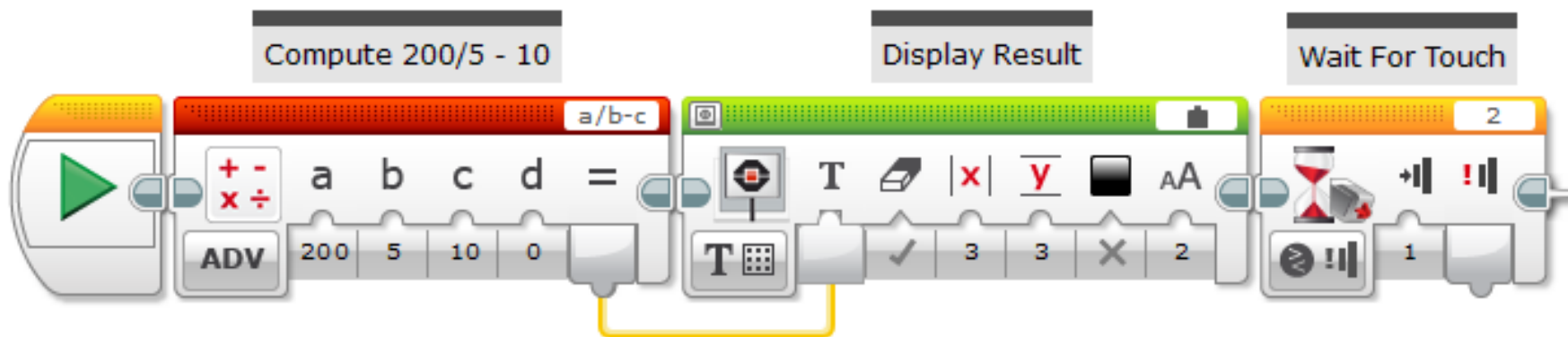
- 可以运用简单的 数学模块
  - 假设  $x = 200\text{mm}$





# EV3 数学运算

- 这里有另一个方案，需要使用 advanced blocks







## 任务 4

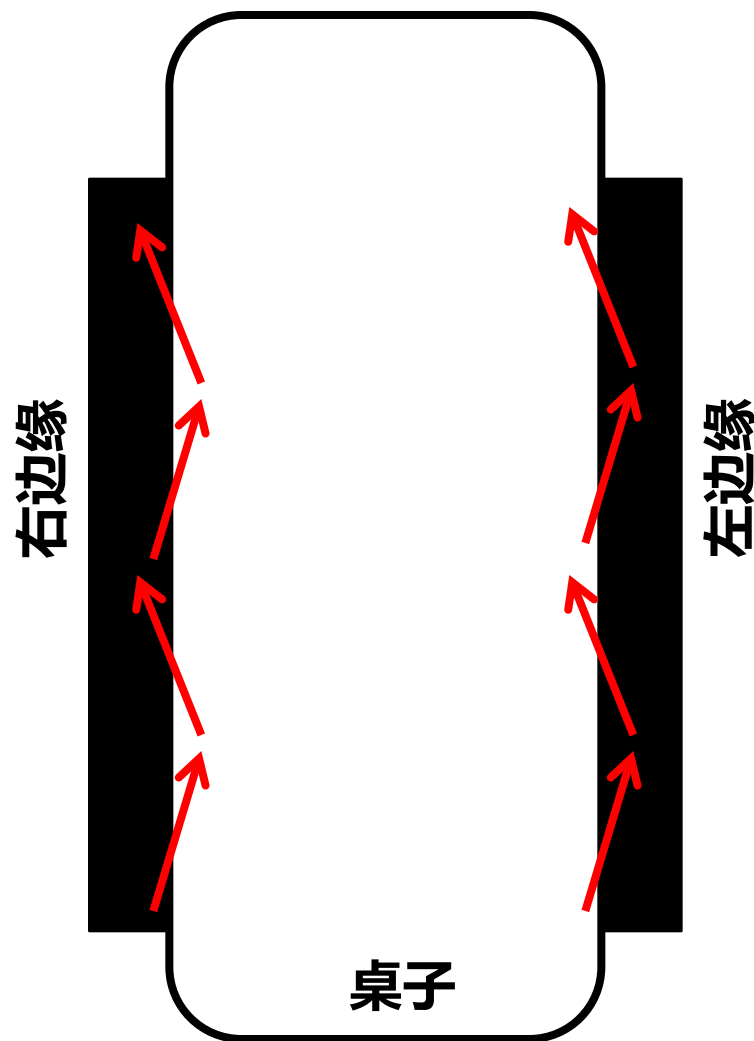
# 沿着桌子边的方向前行

观看视频：见网站



# 沿着桌子边缘前行

- 使用锯齿形方法，沿着桌子的边缘前行
- 遵循边缘前行也被认为是循线
- 我们需要确定机器人什么时候是在桌面或脱离桌面

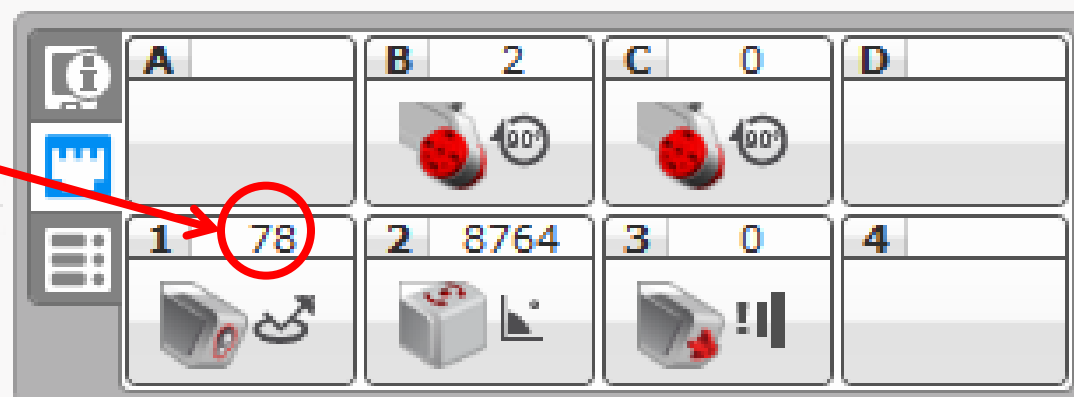




# 沿着桌子边缘前行

- 获得色彩传感器值，以确定当机器人是在桌面上还是脱离桌面。我们将使用颜色传感器反射光强度模式。
  - Off table = \_\_\_\_\_
  - On table = \_\_\_\_\_
  - On foil = \_\_\_\_\_

颜色传感器  
读数



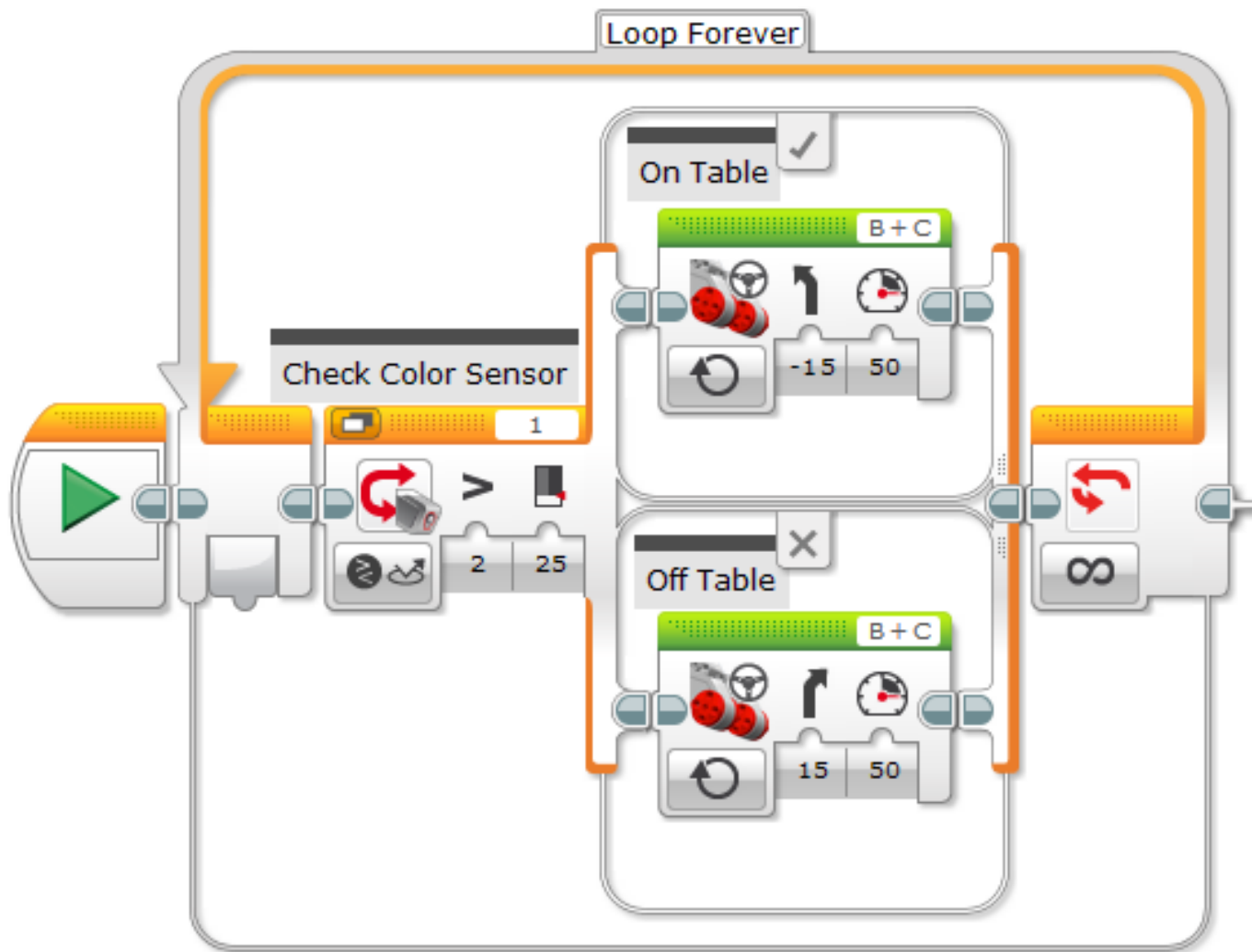


# 沿着桌子边缘前行

- 光电传感器设置实例
  - Off table = 10
  - On table = 40
  - Median threshold =  $(10+40)/2 = 25$
- 两个条件
  - Light sensor reading  $> 25$ . On table.
  - Light sensor reading  $< 25$ . Off table.



# 简单的循线算法

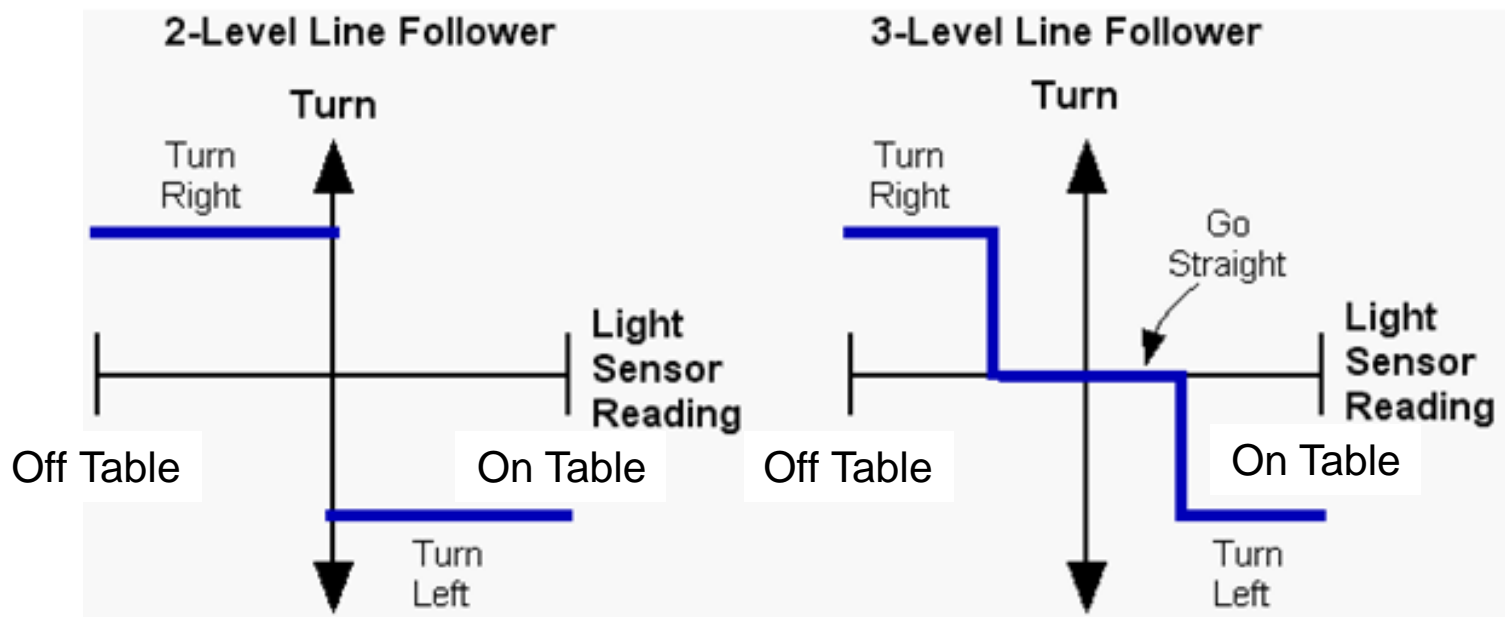


程序: LineFollowZZ.ev3



# 如何提升循线算法

- 曲折的方法可能会导致颠簸响应
- 为了提高响应速度，可以使用3段式循线的方式（如下图所示的概念）





# 任务 5

## 确定“发射区”

观看视频：见网站



# 确定“发射区”

- 假设机器人在“禁止发射区”，它必须到达“发射区”发射网球射击障碍瓶
- 一种方式用于确定“发射区”是寻找并穿过区分“禁止发射区”和“发射区”的黑线
- 这个概念被称作“过线”



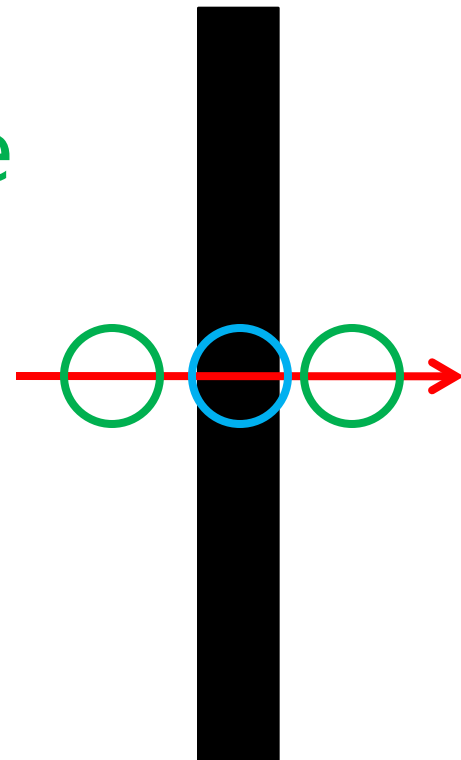


# 过线

- 如何用机器人探测黑线?
  - 直线前进然后探测....

- Off Line = waiting to cross line
- On Line = reached line
- Off Line = crossed line

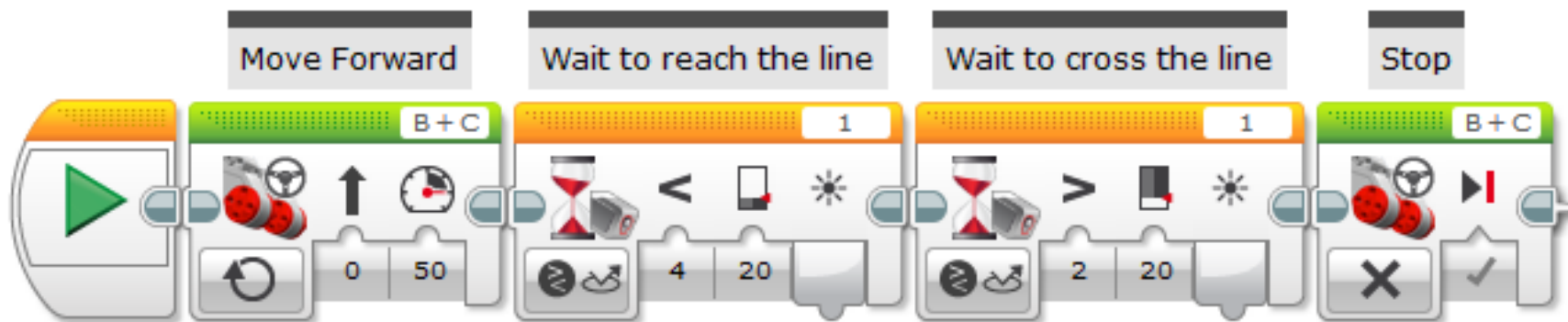
过线





# 过线

- 实例





# 任务 6

## 确定障碍瓶位置

观看视频：见网站



# 确定障碍瓶位置

- 两种方式
  - 给定距离循线
  - 自旋直到超声传感器探测到障碍瓶



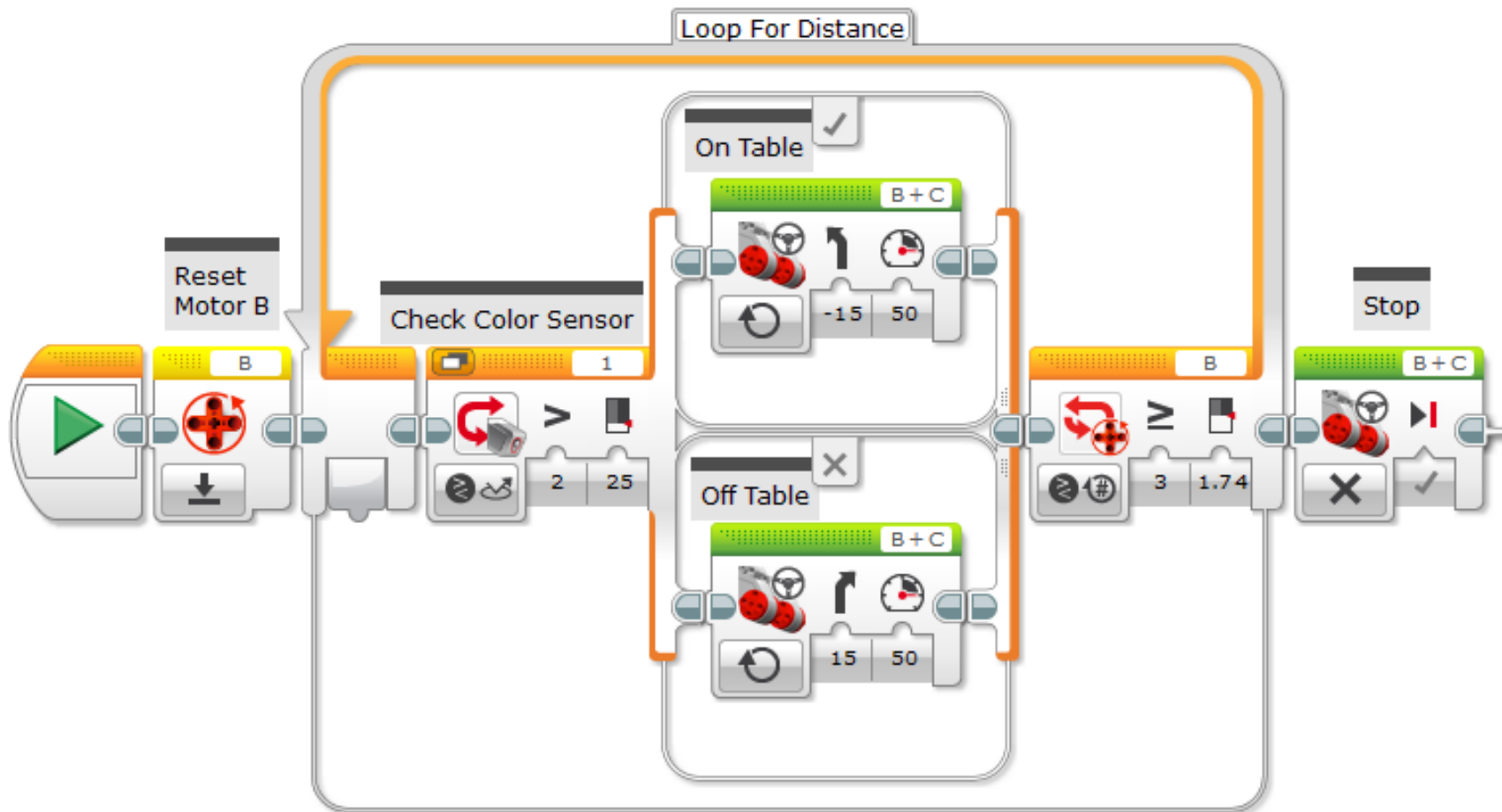
# 给定距离循线

- 首先，需要知道有多长的距离需要行进
  - 假设机器人需要前行 30 cm
  - 30 cm需要机器人的车轮转多少圈?
    - 距离 = (车轮直径) x (PI) x (车轮旋转圈数)
    - 解出 “车轮旋转圈数” ....
    - 车轮旋转圈数 =  $30 \text{ cm} / [(5.5 \text{ cm}) \times (\text{PI})] = 1.74$  圈



# 给定距离循线

- 实例

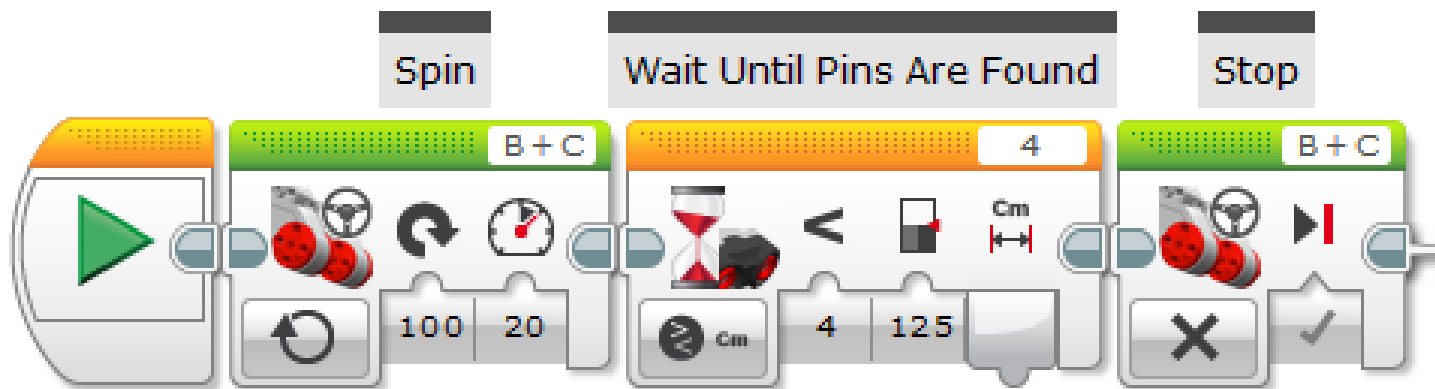


程序: TravelDistance.ev3



# 自旋搜寻

- 机器人自我旋转，直到它用超声传感器“看到”障碍瓶



程序: SpinSearch.ev3



# 任务 6

## 发射网球





# 发射网球

- 此项任务不在本教程说明范围里
- 顺利完成前面任务的机器人将发射网球



# 任务 7

## 返回“基地”

观看视频：见网站



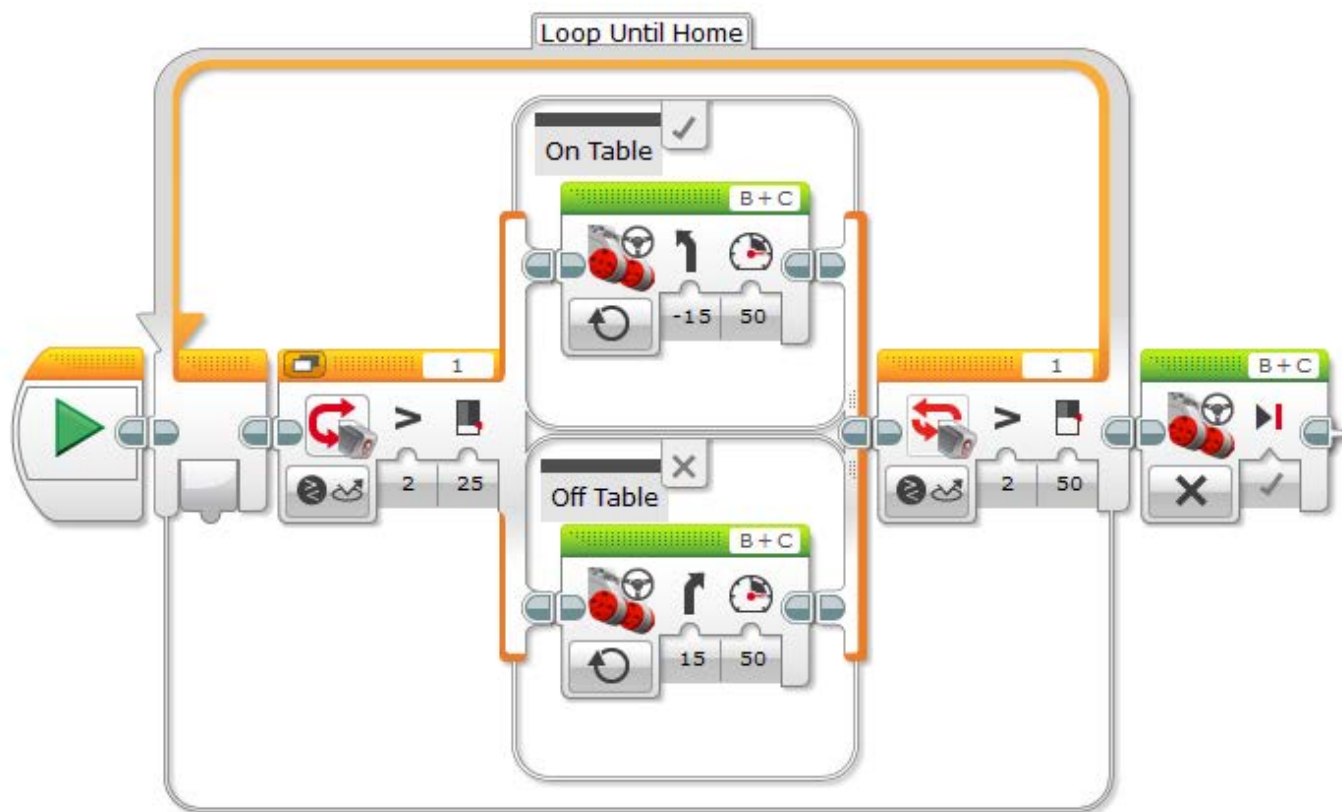
# 返回“基地”

- 一种返回“基地”的方式是循线直到“基地”被探测到
- 同“线”的探测原理一样，可以探测到“基地”
  - On table / off home base = 40
  - On home base = 60
  - Median threshold =  $(40+60)/2 = 50$



# 返回“基地”

- 两个条件
  - Light sensor reading  $> 50$ . On home base.
  - Light sensor reading  $< 50$ . Off home base.



程序: FindHome.ev3



LAWRENCE TECHNOLOGICAL UNIVERSITY

# ROBOFEST

