OSEPP
Create with no limits

BASIC 1

Arduino Robotic

Robotic Kit

编程指南

避障 & 巡线机器人

www.osepp.com cn.osepp.com

目录

1 - 入门:

在本学习指南中，我们有两个软件可以选择,它们都能实现同样的功能。

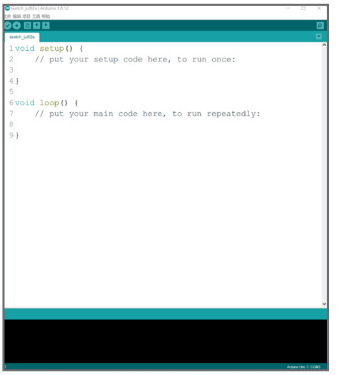
基于拖放操作的图形化编程软件oseppBlock IDE ,或者基于文本的编程软件Arduino IDE 。这些程序使机器人编程变得有趣且易于学习。

选择您的软件，让我们开始吧！

oseppBlock IDE

oseppBlock IDE 适合初学者，是一个图形化的编程软件,基于拖-放操作编程。它设计为使用文本输入编程之前的入门级软件，帮助初学者了解程序结构和语法。

如果您是编程的初学者，我们建议使用 oseppBlock 软件。



Arduino IDE

Arduino IDE 是基于文本输入的编程软件。如果你有一些文本编程的经验，那么 Arduino IDE是适合你的。

2-机器人

I - 什么是机器人?

机器人是一种机电设备,它能够对环境做出反应,做出自主的决定或动作,以完成特定的任务。

这意味着遥控车或遥控机器人不会被视为机器人,因为它们无法感知环境。它完全由人类控制。

II - 机器人能做什么?

机器人无处不在,它们可以通过执行编好的程序来完成一系列任务。它们已经存在于我们的生活中,从打扫房屋到在仓库中进行工业应用。

扫地机器人可以清洁你的房子

想让你的地板保持干净而不动一根手指吗?那就让机器人替你做吧。家庭中最流行的机器人之一是扫地机器人。这个机器人设计的程序就是为你打扫卫生,清洁你的地板和为地毯除尘。

机器人擅长做繁重重复的任务

有没有想过淘宝、京东这些企业的仓库里，机器人如何来整理你的包裹，并迅速交付货物给你？

有成千上万的机器人在那里为他们工作，如果你走进他们的仓库，你会惊讶于这些机器人的能力。

III - 机器人教育的重要性如何？

随着科学技术在现代生活各个方面的发展越来越迅速，每个人都应该尽可能多地了解、设计电子编程和集成电路，以保持竞争力。这就是为什么机器人技术在各级教育中变得越来越重要。

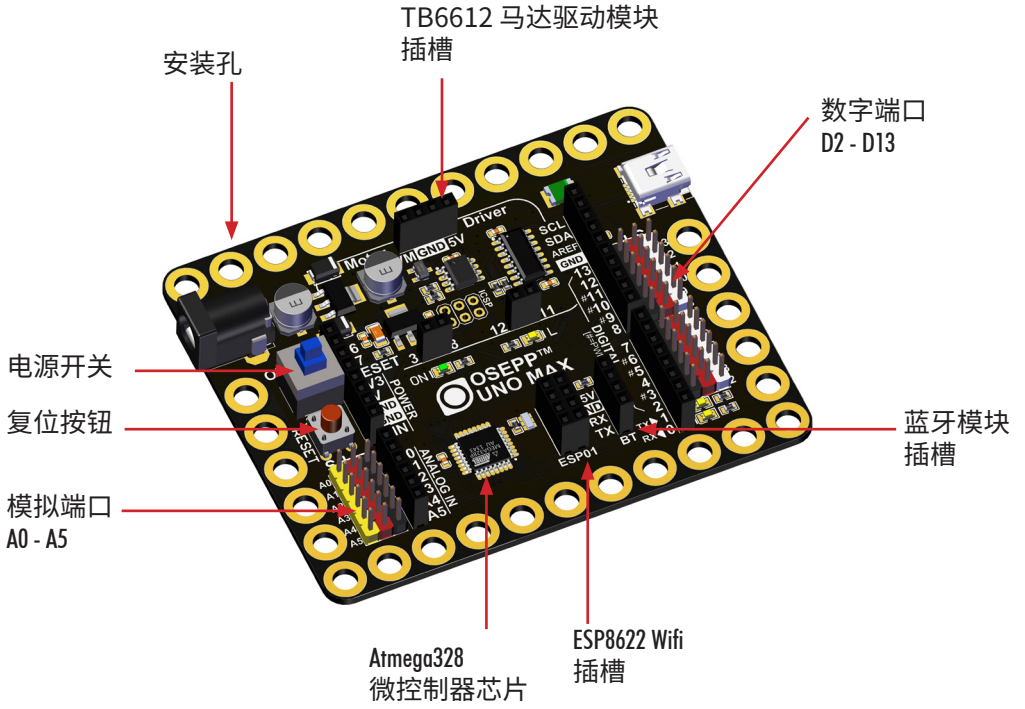
3 - 微控制器 - 机器人的大脑

I - 什么是微控制器?

微控制器是机器人的大脑，基本上是一台小型并且便宜的微型计算机系统。现今使用的大多数电子设备都有某种形式的微控制器。高度集成的微控制器内编程为飞机、家用电器、汽车、手机、玩具等各种应用系统，当然还有作业机器人的任务程序。

在构建机器人时，您需要使用微控制器。它用于感应来自现实世界的输入和基于输入做出自动化控制输出。由Arduino.cc创建的Arduino UNO是众多流行的微控制器平台之一 。它是一个开源平台，主要基于AVR的微控制器芯片Atmega328。

Uno Max 主控板



II - OSEPP Uno Max

OSEPP Uno Max 基于 Arduino UNO 设计，是 Arduino UNO 的兼容版本，专门设计用于机器人。具有许多额外的功能。我们在控制板边缘添加了安装孔，便于机器人组装和传感器连接固定。Uno Max 具有 14 个数字 I/O 端口，6 个 PWM 端口和 8 个模拟 I/O 端口。

此板还配备了马达驱动、蓝牙和 wifi 无线通信插槽。

特征：

- ATmega328P 微控制器芯片
- CH340C USB串口桥芯片
- TT6612 马达驱动模块插槽
- 蓝牙模块插槽
- Wifi 模块插槽

驱动程序：CH340 驱动程序

Uno Max 使用 CH340 芯片 CDC 驱动程序。您可能不需要下载此驱动程序，因为大多数现代操作系统都是内置的。

如果您使用的是较旧的 PC 系统并遇到驱动程序问题，您可以从这里安装 CH340 驱动程序：http://www.wch.cn/download/CH341SER_EXE.html

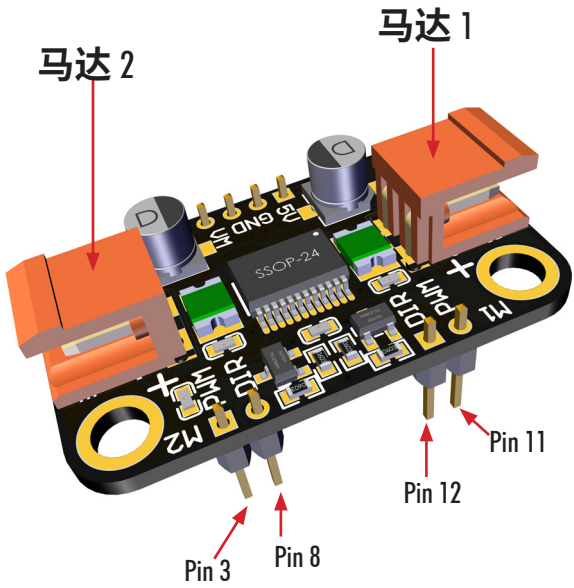
4 - 马达驱动器

I - 什么是马达驱动器?

马达驱动器是一种电流放大器:马达驱动器的作用是将小电流控制信号,转换为大电流信号,来驱动大马达。

II - 为什么机器人需要马达驱动器?

由于微型控制器输出的只是小电流的控制信号,因此需要马达驱动模块。微型控制器有GPIO引脚,可以帮助你旋转一个小马达,但它不能提供足够大的电流来驱动更大的马达,比如我们的机器人使用的直流齿轮减速马达。马达驱动模块是专门为驱动大电流马达而制造的。



III - OSEPP TB6612 马达驱动器模块。

OSEPP TB6612 马达驱动模块可在 1.2A（3.2A 峰值）的恒定电流下控制最多两个直流马达。此驱动器具有足够的功率来驱动大多数中等尺寸的直流金属齿轮马达。每个马达连接到 Uno Max 板上的两个端口。

- 马达 1 连接到端口 11 和端口 12。
- 马达 2 连接到端口 8 和端口 3。

为什么一个马达连接到两个端口?

一个端口用于控制旋转速度，另一个端口用于控制旋转方向。

- 马达 1 - 端口 11 用于速度，端口 12 用于方向控制。
- 马达 2 - 端口 3 用于速度，端口 8 用于方向控制。

IV - 如何控制马达的速度?

马达转速由 PWM（脉冲宽度调制）控制。它通过改变提供给马达的平均电压来工作。

PWM 通常用于控制 LED 的亮度和控制马达的速度。我们的微控制器有 6 个 PWM 端口：（3，5，6，9，10，11）

5 - 安装 - oseppBlock IDE

oseppBlock IDE 是一个易于使用的编程软件，基于图形化、拖-放操作。如果您是编程的新朋友，那么 oseppBlock IDE 最适合您。

尽管oseppBlock可以实现文本IDE的大部分功能，但我们仍然建议您阅读oseppBlock自动生成的Arduino代码，这正是oseppBlock与Mixly（米思齐）、mBlock等图形编程软件的不同之处。
oseppBlock致力于帮助您熟悉文本程序的语法和结构，而不是替代。为此，oseppBlock中积木样式都尽量保持与文本编程中的格式一致。

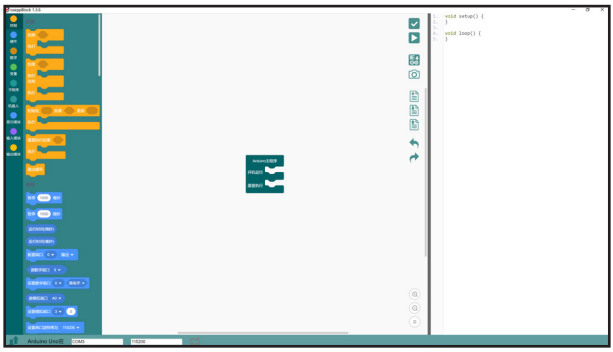
I - 下载 oseppBlock IDE：

1. 从这里下载最新的 oseppBlock IDE：
https://cn.osepp.com/software/oseppblock.html

注意:选择与操作系统相匹配的版本

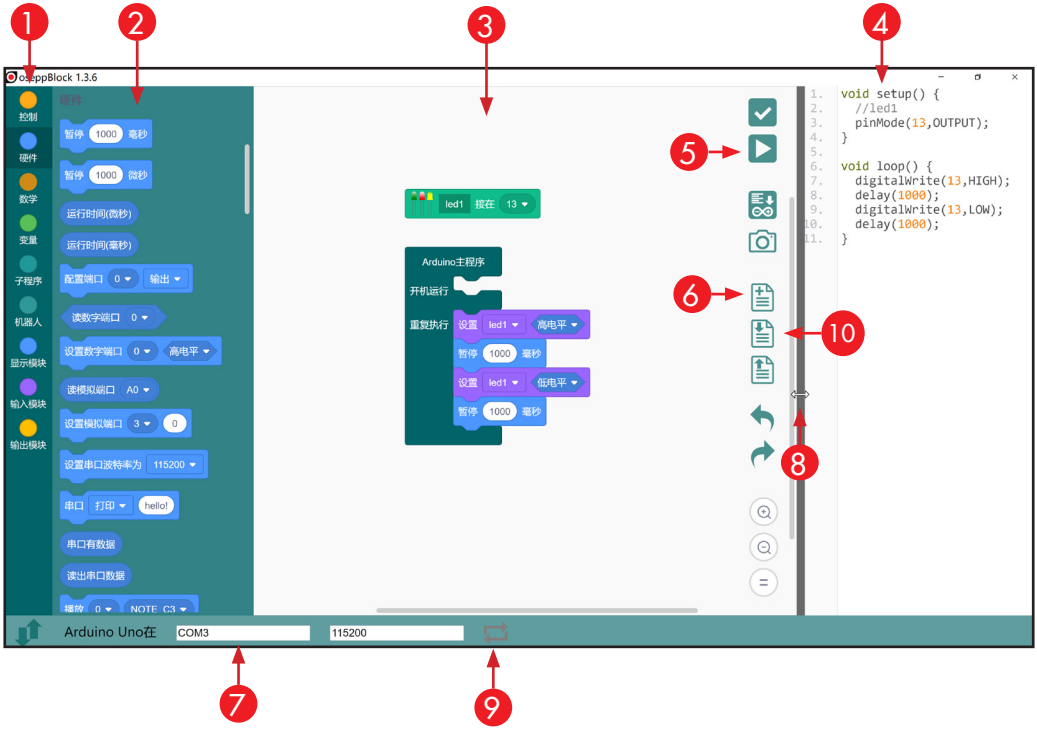
- **Windows操作系统(Win7及以上)**
  - 32位
  - 64位
- **Linux操作系统**
  - 32位
  - 64位
- **OS X(苹果电脑)**
- **Raspberry Pi(树莓派)**
- 在线尝试

2. 下载完成后，将 oseppBlock-xyz.zip 文件放置到所需的工作文件夹路径，然后解压缩它。
3. 如果您不确定该如何操作，请查看网址：
https://cn.osepp.com/tutorial/robopro/oseppblock_ide_install.html
4. 打开 oseppBlock-xyz 文件夹，然后双击 oseppBlock 应用程序文件。
5. 将显示下面的 oseppBlock 程序窗口。



II - 快速了解 oseppBlock IDE：

1. 目录区:快速定位需要的积木
2. 积木区: 提供积木原型,可拖放到工作区
3. 工作区:积木搭建程序区域
4. Arduino代码区：实时将积木转换成Arduino代码并显示在代码区。

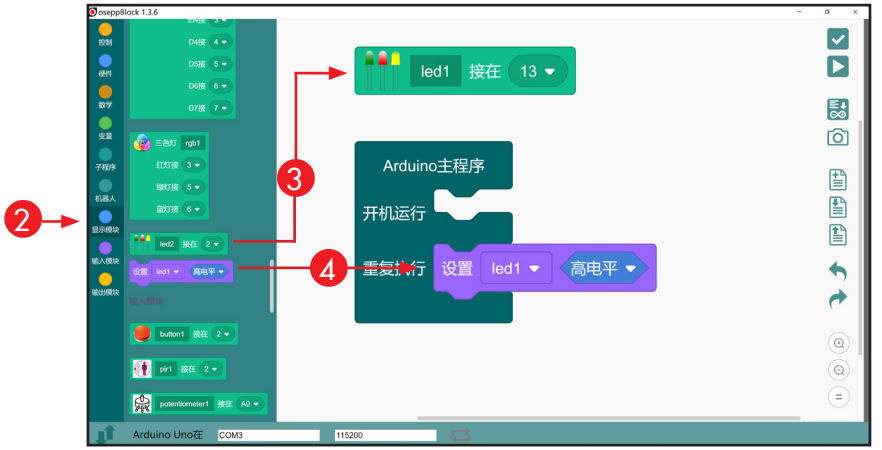


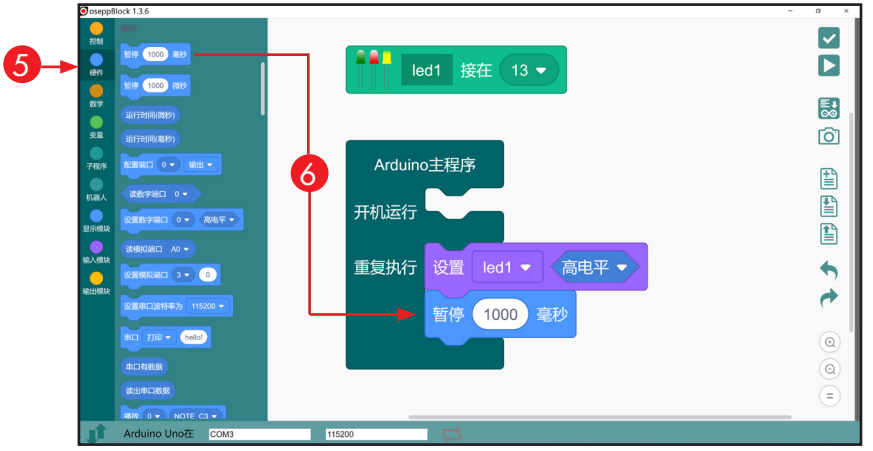
5. 单击此处上传您的程序到Uno Max
6. 单击此处清空工作区，以创建新程序
7. 单击此处选择连接微控制器的串行端口号
8. 您可以通过使用鼠标拖动来调整代码区的大小、关闭/打开。
9. 单击此处连接到串行端口，并打开串行监视器窗口。
10. 单击此处保存积木。(输入文件名和 .obp 例如：Blink.obp)

III - 闪烁 LED

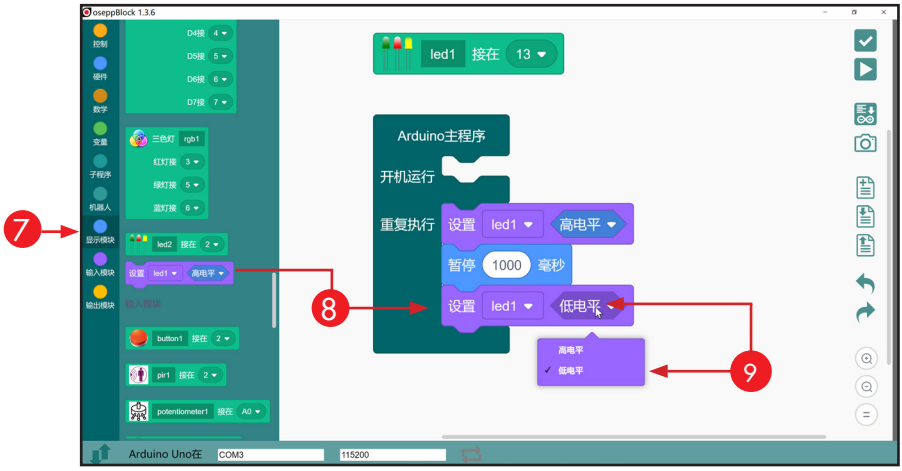
在本章节中，我们将向您展示如何编写简单的LED闪烁程序并加载到您的微控制器。

1. 打开 oseppBlock 程序
2. 单击 **显示模块** 菜单
3. 将 **led1模块** 积木拖动到工作区域
4. 拖动 **设置led1** 积木块到重复执行槽

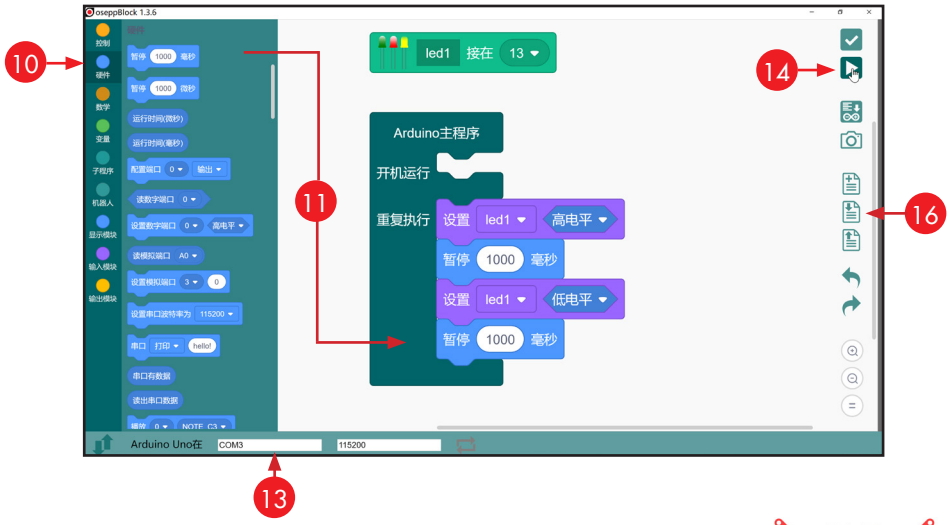


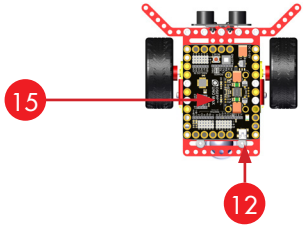


5. 点击 **硬件** 菜单
6. 选择 **暂停1000毫秒** 积木并拖动到图示位置



7. 点击 **显示模块** 菜单
8. 选择 **设置led1** 并拖动到图示位置
9. 点击图示位置展开下拉列表，选择“**低电平**”





10. 点击 **硬件** 菜单
11. 选择 **暂停1000毫秒** 积木并拖放就位
12. 通过 usb 电缆将微型控制器连接到 PC
13. 单击此处选择串行端口（COM[X]）。
14. 单击“上传”按钮上传程序。（并稍等片刻）
15. 程序上传到微控制器上后，您会看到图示的LED在闪烁。
16. 单击此处保存文件。（使用 .obp 作为文件后缀名，例如：blink.obp）

6 - 使用oseppBlock编程机器人

前进和停止

一个没有任何指令的机器人只是一块硬件，只是呆在那里，什么也不做。很像没有司机的车。硬件是程序的载体，程序是一组指令，使机器人能够执行某些任务。如果你想开车，就需要学习基本的驾驶技能。那我们要让机器人动起来，就要学会基本的驱动程序指令。

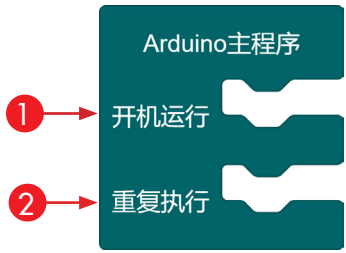
I -如何让机器人动起来?

我们将从编程的基础知识开始，向您展示如何驱动机器人前进和停止。

II - 需要学习什么代码?

Arduino 程序

Arduino程序也被称为草图(sketch)。它是Arduino 用于驱动硬件的程序。也是上载到微控制器的代码单元，并在微控制器上运行。基本的 Arduino 程序由两个函数组成：



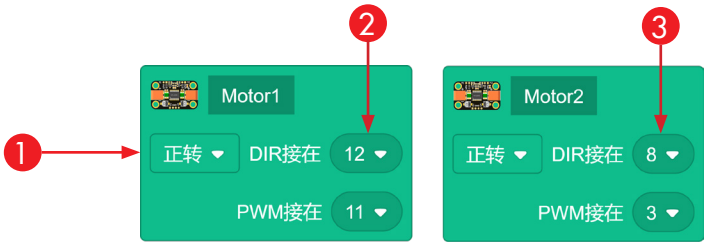
1. setup()
2. loop()

您可以在开机运行(setup)或重复执行(loop)函数中放置代码。

“**setup**”函数中的代码仅在每次主板接通电源或按下复位（reset）按键时执行一次，主要用于定义端口和一些初始化工作。

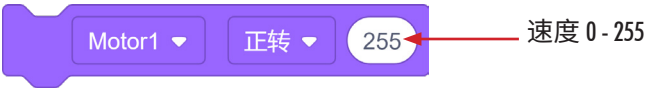
“**loop**”函数中的代码将周而复始地一直循环执行,这将是实现功能的主要地方。

马达模块积木和设置马达积木：



马达模块 积木位于**机器人**目录菜单中，其主要功能是定义马达的线路连接。Uno Max 可控制多达 6 个马达。请记住，每个马达模块由 2 个端口组成，一个端口用于方向控制，另一个端口用于旋转速度，该端口需要连接到 PWM 端口。

1. 从下拉列表中选择正转/反转,调整马达的默认方向
2. 默认情况下，马达 1 连接到端口 12 和 11
3. 默认情况下，马达 2 连接到端口 8 和端口 3



设置马达 积木用于驱动马达前进、后退和停止。此模块还允许您将马达的速度设置为 0 到 255，使马达产生可以调节的旋转速度。

暂停 积木位于**硬件**目录菜单中，用于暂停程序一段时间，以毫秒为单位（1000 毫秒等于一秒钟）。

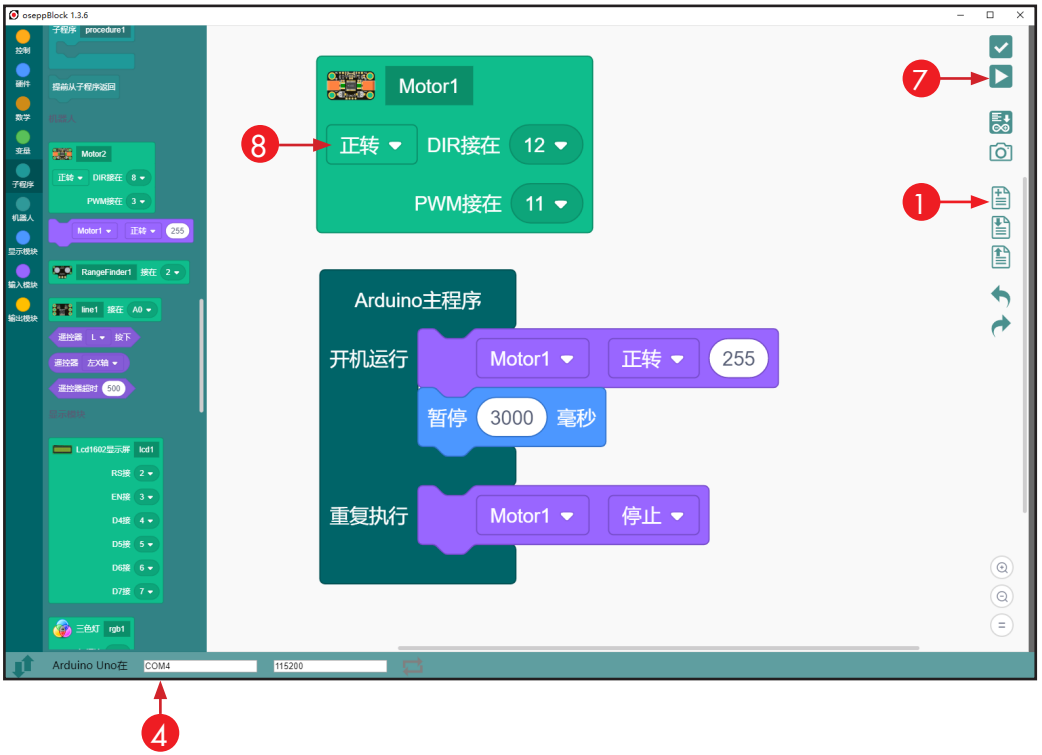
III - 搭建程序

在此程序中，我们将驱动马达前进 3 秒钟并停止。

编写此程序只需要 3 个步骤：
1. 设置马达前进
2. 等待 3 秒钟
3. 将马达设置为停止

IV - 在 oseppBlock 中编程：

1. 创建新工作区
2. 搭建图中的程序



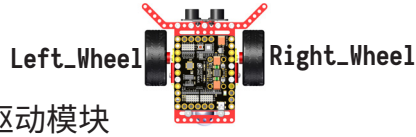
3. 通过 usb 电缆将机器人连接到 PC
4. 单击此处选择 COM 端口。(COM[X])
5. 打开控制板上的电源开关
6. 保持机器人悬空，并避免接触到轮子
7. 单击上传按钮上传程序。

将程序加载到机器人上后，您应该会看到机器人一个马达旋转3 秒钟，然后停止。您可以按控制器上的复位（reset）按钮来重复此过程。

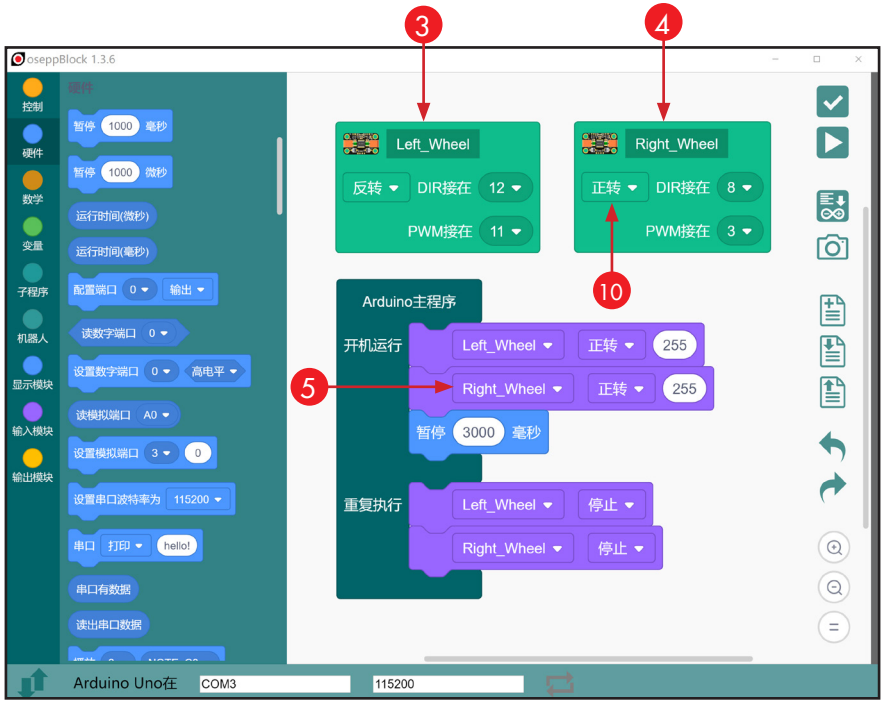
8. 如果马达旋转的方向不是驱动机器人向前移动的方向，请在马达模块积木中选择“反转”选项。
9. 再次上传程序 - 此时马达旋转方向应该是驱动机器人前进。
10. 记下马达位于机器人的那一侧（左或右）

V - 在 oseppBlock 中编程：

现在，您将需要编程另外一个马达做同样的事情。



1. 从上一个程序继续
2. 搭建下图中的程序
3. 根据之前的运行结果来正确命名马达驱动模块
   Left_Wheel位于机器人的左侧
4. Right_Wheel位于机器人的右侧。
5. 从下拉菜单中选择“Right_Wheel”



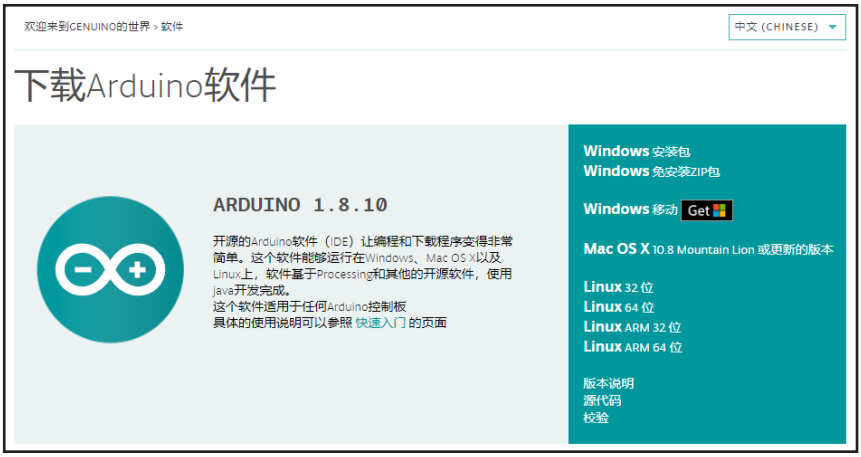
6. 通过 usb 电缆将机器人连接到 PC
7. 打开控制板上的电源开关
8. 保持机器人悬空，并避免接触到轮子
9. 单击上传按钮上传程序。
   将程序加载到机器人上后，您应该会看到两个车轮向前旋转 3 秒钟，然后停止。将机器人放在地上，然后按下复位（reset）按钮，看看机器人如何运行。
10. 如果机器人在原地旋转，这意味着您的一个车轮向后旋转。请在马达驱动模块积木上调整方向并再次上传程序。直到机器人变为前进。

7 - 安装 - Arduino IDE

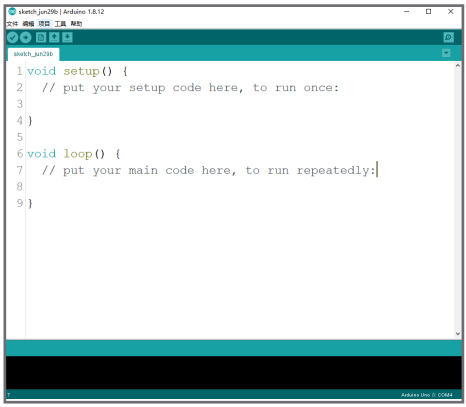
您可以使用 Arduino IDE 基于文本编程软件来实现同样的功能。如果您还没有准备好使用文本编程，请跳过此部分以及后续教程中的Arduino IDE编程章节。

I - 下载Arduino IDE：

1. 从这里下载最新的 Arduino 环境：https://www.arduino.cc/cn/Main/Software



2. 下载完成后，将 arduino-xyz.zip 文件放置到所需的工作文件夹路径，然后解压缩它。如果您不确定该如何操作，请查看网址：https://cn.osepp.com/tutorial/robopro/arduino_ide_install.html
3. 打开 arduino-xyz 文件夹并双击 arduino 应用程序文件。
4. 将显示下面的 Arduino 程序窗口。



II - 安装 oseppRobot 库：

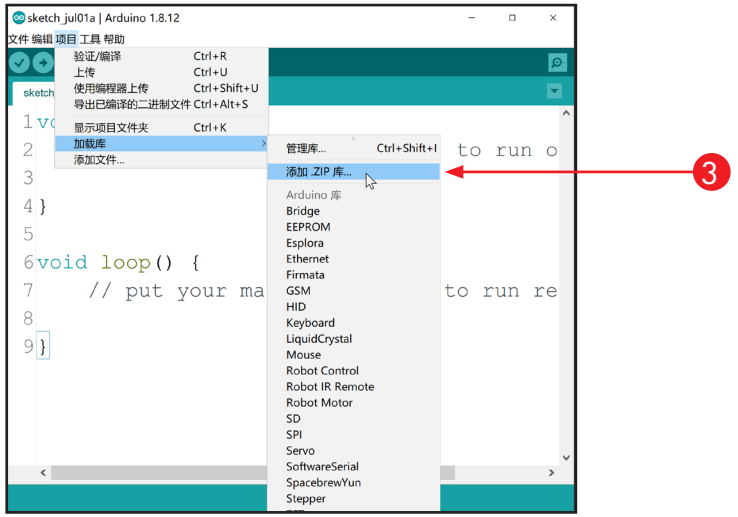
您将需要下载 oseppRobot 库并安装到您的 Arduino 环境中。

什么是程序库?

库是已编译并存储在单个文件中的相关代码片段的集合。该文件可以链接到您的代码，以便您访问库中的代码。

使用程序库好处主要是，您不必重复编写同样的代码，来实现现有的程序库中的功能。这使得代码更短，更易读。

1. 从这里下载 oseppRobot 库 zip 文件：
   https://cn.osepp.com/download/oseppRobot.zip
2. 打开Arduino IDE
3. 在文件菜单中，选择: 项目 > 加载库 > 添加.zip库 。
4. 找到并单击 oseppRobot. zip 文件，将 oseppRobot zip 文件作为库添加到 Arduino 环境。



我们将在下一教程中向您展示如何将 oseppRobot 库包含到您的程序中。

8 - 使用 Arduino IDE 编程机器人

运行和停止

在上一教程中，我们学习了如何在oseppBlock编程机器人前进和停止。在本节教程中，我们将向您展示如何在 Arduino 程序中编程。

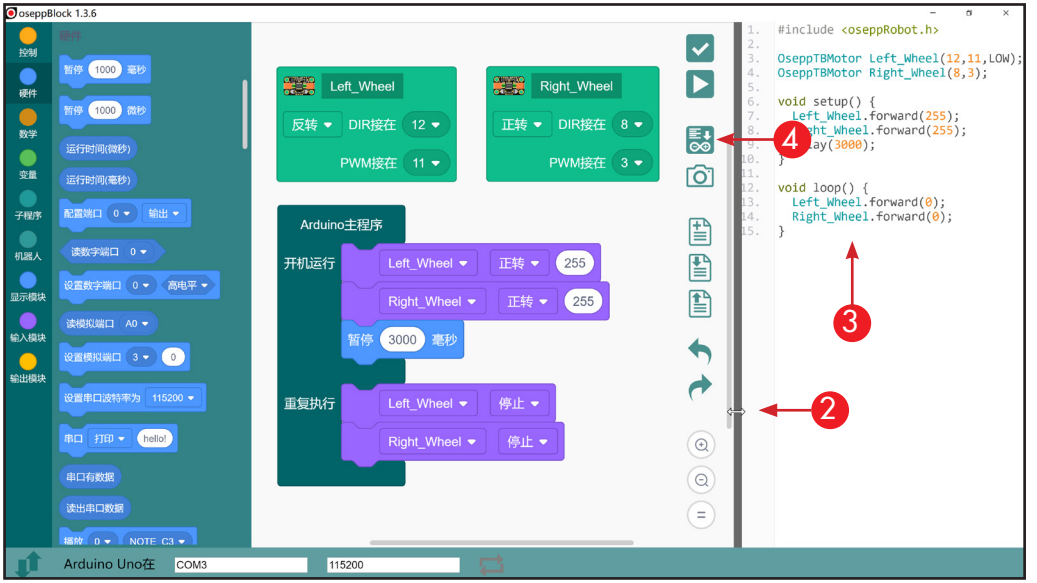
I - 如何在 Arduino IDE环境中对机器人进行编程?

Arduino IDE环境中的编程与在 oseppBlock 环境中的编程大致相同。主要区别在于一个是基于拖放操作，另一个是文本输入。

如果尚未安装 oseppRobot 库，请安装该库。(请参阅第 18 页)

II - 将”前进和停止”程序加载到 Arduino IDE 中：

1. 在 oseppBlock 中打开上一个程序
2. 用光标将 Arduino 文本代码窗口面板向左滑动
3. 在这里，你会看到你的Arduino代码
4. 点击此处将Arduino代码保存到文件，或者使用快捷键复制代码。



5. 打开 Arduino IDE
6. 通过 **#include <oseppRobot.h>** 引入 oseppRobot 库
7. 定义 **OseppTBMotor Left_Wheel** 和 **Right_Wheel**
8. 粘贴或输入程序的其余部分

9. 通过 usb 电缆将机器人连接到 PC
10. 打开机器人控制板上的电源开关
11. 单击：工具>开发板>Arduino UNO 选择您的控制板类型
12. 单击：工具>端口>COM [X] 选择连接控制板的端口号
13. 单击“上传”按钮上传程序
14. 上传完成后，您将看到“上传成功”消息

将程序加载到机器人上后，您应该会看到两个车轮向前旋转 3 秒钟，然后停止。将机器人放在地上，然后按下重置按钮，看看机器人如何运行。

15. 如果机器人在原地旋转，这意味着您的一个车轮向后旋转。使用“**LOW**”或者“**HIGH**”调整车轮方向，然后再次上传程序。

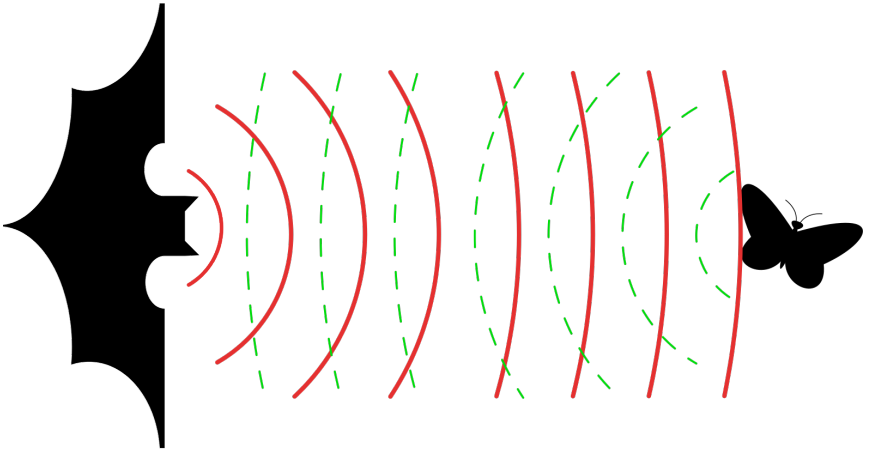
9 - 超声波传感器

超声波传感器使机器人能够像我们用眼睛一样看到周围的世界。

I - 什么是超声波传感器?

超声波传感器是一种使用超声波测量物体距离的仪器。该传感器能提供有关机器人周围环境的信息,非常适合用于测量物体的远近。

II - 超声波传感器如何工作?



你知道蝙蝠如何在晚上飞行和狩猎吗?

蝙蝠的视力很差。蝙蝠发出尖锐的叫声,然后听回音以确定其周围环境。超声波传感器对物体的检测方式与蝙蝠相同,主要依靠回声定位技术。

超声波传感器通过发出高于人类听觉范围的频率的声波来工作。传感器的发射器发出一阵超声波，接收器充当麦克风来接收超声波。传感器通过测量超声波脉冲发送和接收之间花费的时间来计算与目标的距离。

此传感器的工作原理很简单。它以 40kHz 的速度发出超声波脉冲，通过空气传输，如果有障碍物或物体，它会反弹回传感器。通过计算花费的时间乘以声音传播的速度，就可以算出距离。

III - OSEPP 超声波模块

OSEPP Range-finder超声波传感器能够测量 2cm 至 300cm 范围内的对象，但在某些环境中，此范围可能受到限制。由于超声波测距依赖于声波，因此任何吸收或偏转声音的物体，如柔软物体的或倾斜的表面可能影响测量精度。

超声波传感器上有 3 个 端口：S（信号）、+（电源正极）和 -（电源负极）。

10 - 使用超声波传感器检测距离：

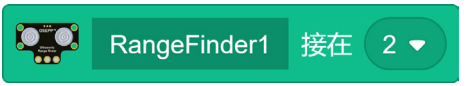
I - 机器人怎么知道物体有多远？

OSEPP RobotPro机器人配备了超声波测距传感器，使我们的机器人能够判断物体有多远。

在本教程中，我们将向您展示如何使用超声波传感器测量物体的远近，并将此数据通过串行通信显示在我们的 PC 显示器中。

II - 我们可以学习哪些新代码？

超声波模块积木



此模块位于**机器人**目录菜单中，超声波模块连接到控制器2号端口上，请根据您的实际接线来设置它。使用**超声波测量**积木来启动测量并得到测量结果。

测量结果以毫米为单位，表示超声波模块与测量对象之间的距离。

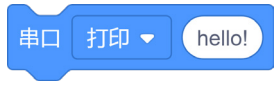
串行通信

串行通信用于计算机和微控制器之间通信的协议。通过串行通信，我们可以做两件事：我们可以发送命令或接收数据。在我们的案例中，我们将从超声波传感器接收数据，并将这些数据发送到我们的 PC。

设置串口积木：

设置串口 积木位于 **硬件** 菜单中。它用于设置数据速率，以比特（bit）/秒为单位，用于串行数据传输。为了与串口监视器通讯，请使用设置串口积木初始化串行通信，并确保两者使用相同的波特速率。

串口打印积木：



使用 **串口打印** 积木向计算机发送传感器数据或文本消息。

串口打印并换行积木：

使用该积木打印传感器数据或文本，之后串口监视器会将光标移动到下一行的行首。

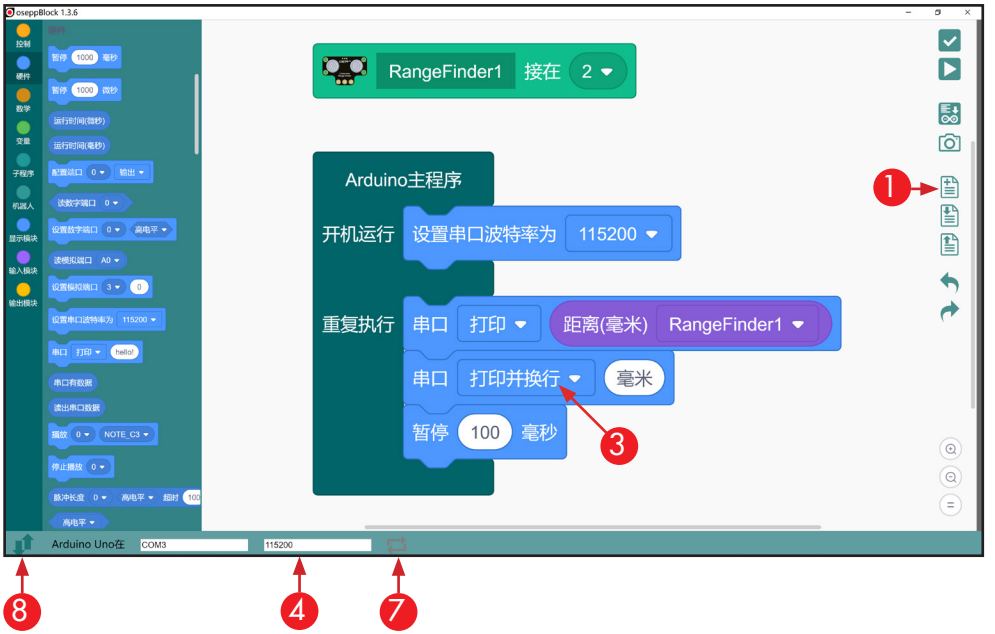
III - 搭建程序：

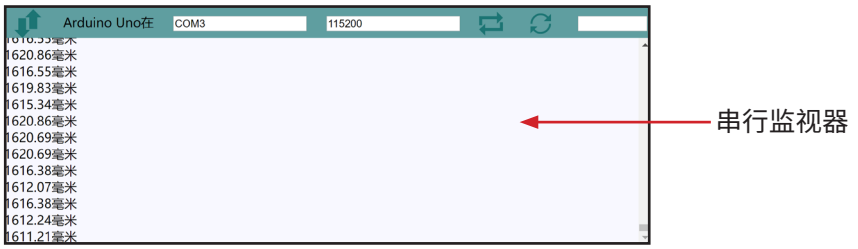
在此程序中，我们将向您展示超声波测距传感器如何测量距离。我们将检测到超声波传感器前面最近物体的距离（以毫米为单位），并发送到PC。

1. 使用 **超声波模块** 积木声明超声波模块接在2号端口
2. 使用 **设置串口** 积木初始化串行通信端口
3. 使用 **超声波测量** 积木得到距离，并使用 **串口打印** 积木发送
4. 使用 **串口打印并换行** 积木发送消息和换行
5. 添加 100 毫秒的暂停时间

IV - oseppBlock IDE 编程：测量距离

1. 启动新项目
2. 构建下面的程序
3. 从下拉列表中选择“打印并换行”



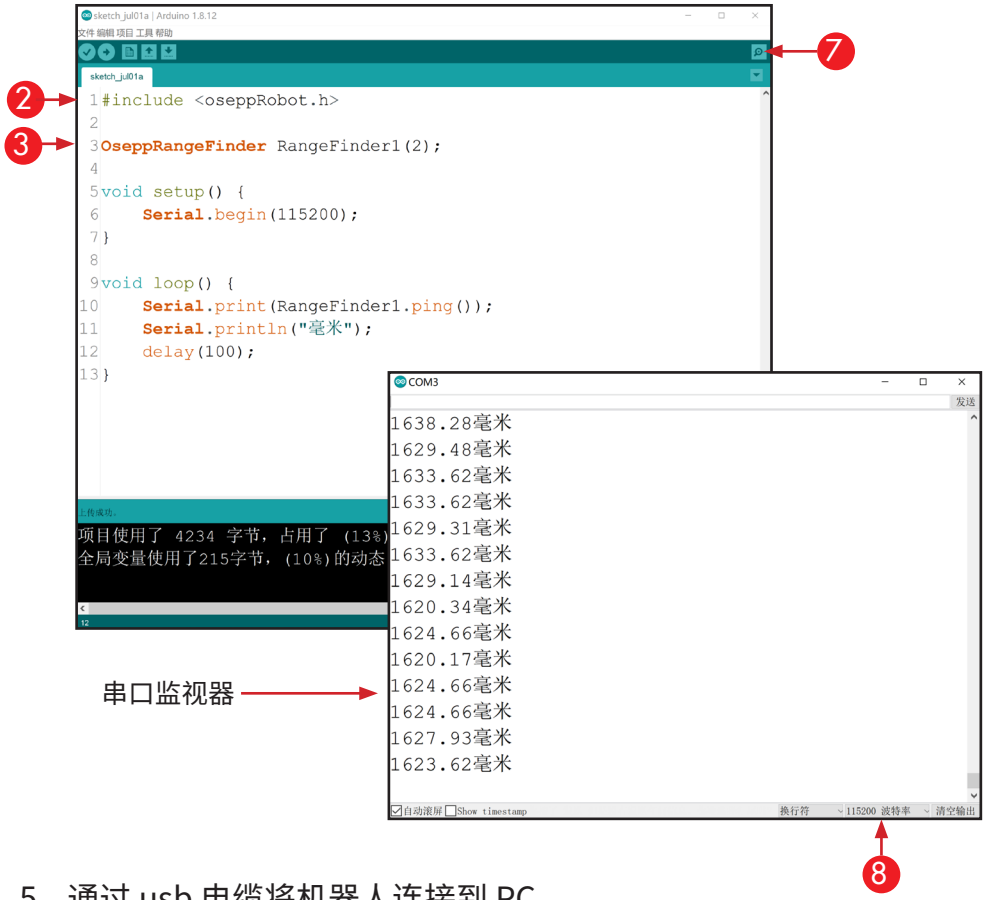


4. 选择与 **设置串口** 积木中相同的波特率，即 115200
5. 通过 usb 电缆将机器人连接到 PC
6. 上传程序
7. 单击此处打开/关闭串行监视器
8. 要折叠/展开串行监视器，请单击此处

将程序加载到微控制器上后，在超声波传感器前面移动测量物体，即可通过串行监视器看到控制器发送的消息：测量物体的距离。

V - Arduino IDE 编程：测量距离

1. 启动新项目
2. 引入 oseppRobot 库：**#include <oseppRobot. h>**
3. 定义 **OseppRangeFinder RangeFinder1(2);**
4. 输入其余代码



5. 通过 usb 电缆将机器人连接到 PC
6. 上传程序
7. 单击此处打开串行监视器或在菜单栏：工具>串行监视器
8. 在串口监视器中选择与**Serial.begin**中相同的波特率，即 115200

将程序加载到微控制器上后，在超声波传感器前面移动测量物体，即可通过串行监视器看到控制发送的消息：测量物体的距离。

11 - 避障机器人

前进和转弯

I - 我们如何创建一个避障机器人?

现在,我们已经学会了如何检测对象有多远,那我们就可以轻松地创建一个具有条件语句(**if/else**)的自主机器人。如果机器人检测到指定距离内的障碍物则转弯,否则向前移动。

II - 我们可以学习哪些新代码?

如果-执行(If/else)语句

此积木位于**控制**目录菜单中。

如果（if）语句将检查条件,如果条件成立,就执行对应的代码块。

如果-否则（if/else）,如果 if 语句中的条件成立,将执行 if 语句的代码块。否则执行else的代码块。

在Arduino语言中,逻辑结果用“True”/ “False”来表示,
“True” 通常会翻译为: 成立、 真、 是。
“False”通常会翻译为:不成立、 假、 否。
它们都代表逻辑结果,通常用于控制程序流程。
在后面的教程中,我们将统一使用True和False来表示逻辑结果。
当你看到True时,你可以理解为:成立、真、是,
False可以理解为:不成立、假、否等同义词。

比较运算

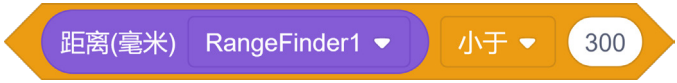
此积木位于 **数学** 目录菜单中。

比较运算符用于比较符号两边的两个表达式值。判断它们是否满足某种关系。比较的结果可以是“True”或“False”

可以从下拉列表中选择 6 个比较运算。

1. 等于(`a == b`):运算符两边的值a、b相等?
2. 不等于(`a != b`):运算符两边的值a、b不相等?
3. 小于(`a < b`):运算符左边的值a小于右边的值b?
4. 大于(`a > b`):运算符左边的值a大于右边的值b?
5. 小于等于(`a <= b`):运算符左边的值a小于或者等于右边的值b?
6. 大于等于(`a >= b`):运算符左边的值a大于或者等于右边的值b?

例子:



如果超声波检测到的距离小于 300mm
则表达式结果为True,否则表达式结果为False

III - 搭建程序

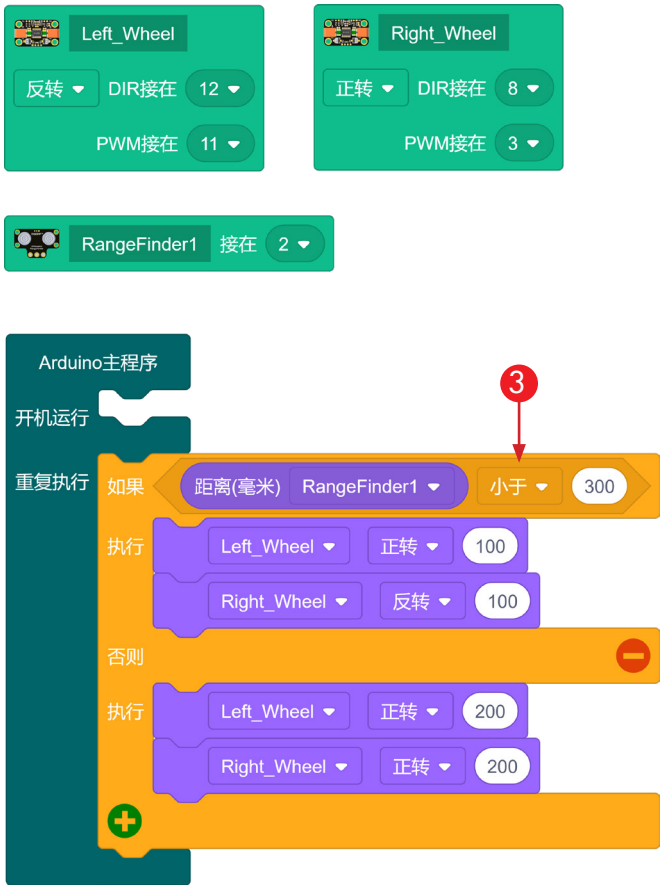
在此程序中,我们将使用比较运算符测试障碍物距离是否小于 300 毫米远。如果结果为True,那么我们可以通过控制一个车轮向前,另一个向后转动来编程马达以让机器人转身。

此程序中只有 3 个步骤:

1. 测试 - 超声波检测到的距离小于 300 mm?
2. 如果结果为True,机器人转身离开以避免碰撞
3. 如果结果为False,机器人可以继续向前行驶

IV - oseppBlock 中的程序：避障

1. 新建工作区
2. 构建以下代码块
3. 在此处拖入**比较运算**积木



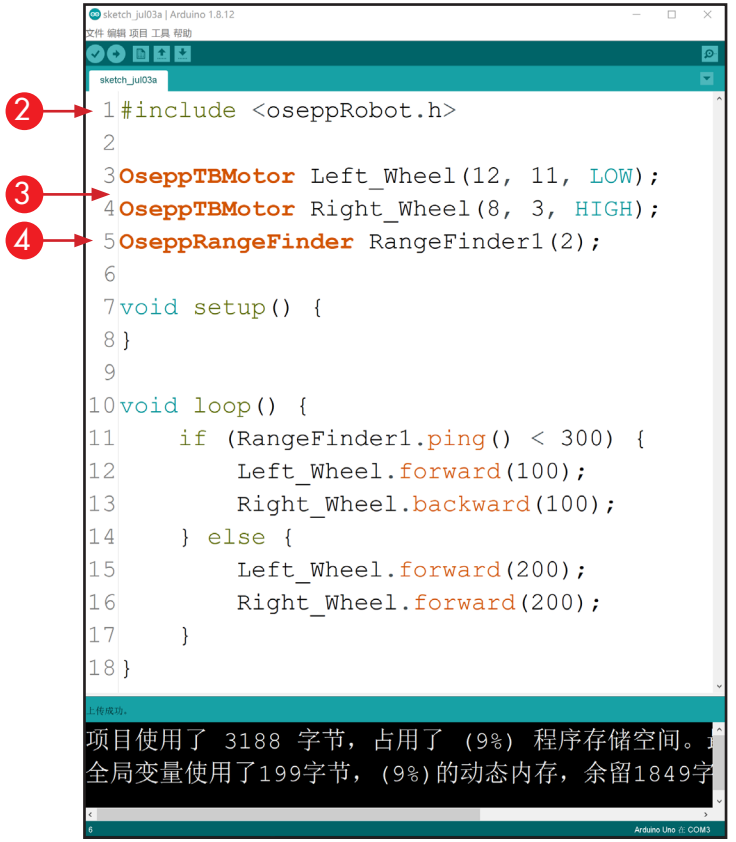
4. 通过 usb 电缆将机器人连接到 PC
5. 上传程序

将程序加载到机器人上后，将机器人放在地上并打开电源。你应该看到机器人在房间里漫游，并避让它看到的任何物体。

6. 如果机器人没有像程序预期那样工作，请检查电池电量是否充足。

第五步 - Arduino IDE 中的程序：避障

1. 新建项目
2. 引入 **oseppRobot** 库
3. 定义两个 **OseppTBMotors**
4. 定义 **OseppRangeFinder RangeFinder1**
5. 输入余下的程序代码



6. 通过 usb 电缆将机器人连接到 PC
7. 上传程序

将程序加载到机器人上后，将机器人放在地上并打开电源。你应该看到机器人在房间里漫游，并避让它看到的任何物体。

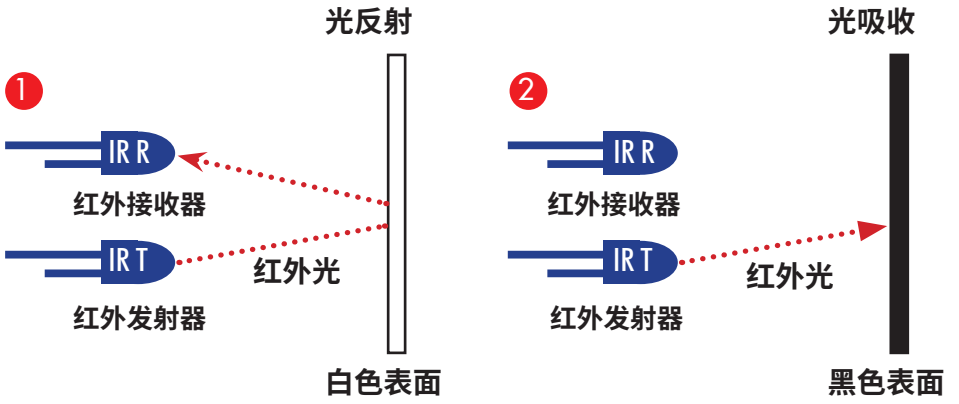
12 - 红外反射传感器

I - 什么是红外反射传感器?

红外反射传感器是机器人用于检测物体明/暗对比度的传感器。这使得它特别适合用于检测白色平面上的黑线或黑暗平面上的白线。

红外反射传感器是一种常见的机器人传感器，用于许多机器人应用，如创建一个巡线机器人，迷宫机器人，检测赛场边界，等等。

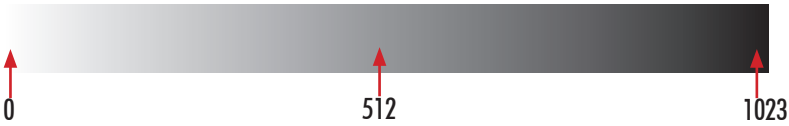
II - 红外反射传感器如何工作?

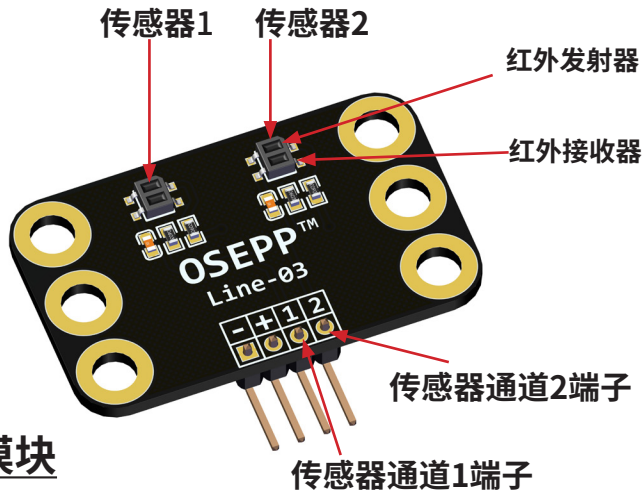


传感器有 2 个光电二极管，一个将红外光照射到对象表面，另一个光电二极管接收从物体表面反射回来的红外光。它接收的光量会导致电压变化。通过检测这种变化的电压，我们可以确定物体的对比度。

1. 当红外光照射到白色物体表面时，光被反射回来从而被接收二极管接收。白色表面没有吸收光线，传感器得到较低的输出值(接近0)。

2. 当红外光照射黑色物体表面时，光被表面吸收，接收二极管接收不到光线。黑色表面物体将光线全部吸收，传感器输出最高的值(1023)。





III - OSEPP 巡线模块

OSEPP 巡线模块配备 2 组红外反射传感器（传感器 1 和传感器 2）。每个传感器都有一个红外发射器和一个红外接收器。

OSEPP 巡线模块是模拟传感器，这意味着它返回 0 到 1023 范围内的值，具体取决于从传感器接收的红外光线的被吸收了多少。

为了获得最佳效果，使用线路传感器时，将传感器安装在离测量表面 1/8 - 1/4 英寸或 3mm - 6mm 之间。避免阳光照射也同样重要，这些措施会让传感器读数尽量保持准确。

13 - 检测路线

I - 机器人如何检测路线?

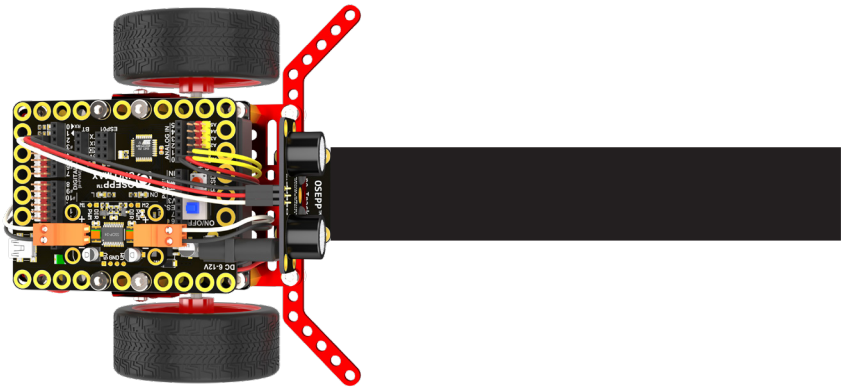
从上一节中我们知道巡线传感器是模拟传感器，根据传感器接收的红外光强度，返回从 0 到 1023 的模拟值。我们还知道，当检测吸收光线的黑色物体时，它将得到一个较高的值。如果读数为 800 或更高，那么我们可以确定检测到黑线。

让我们编写一个程序，通过串行通信打印传感器检测到的数据值。

II - 你需要的东西

使用标准尺寸的电工胶带，粘贴在光滑的地板上。

* 确保传感器 通道1 连接到微控制器的A0端口。

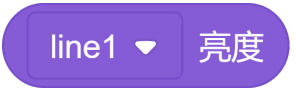


III - 我们可以学习哪些新代码?

巡线传感器模块 积木

巡线传感器模块积木位于**机器人**目录菜单中。在使用红外反射传感器硬件时使用此积木。请确认红外反射传感器正确连接到A0端口。

巡线传感器测量 积木



此积木测量 0 至 5 伏之间的电压，并将其转换为 0 到 1023 之间的模拟值。使用此积木来读取测量表面的值。

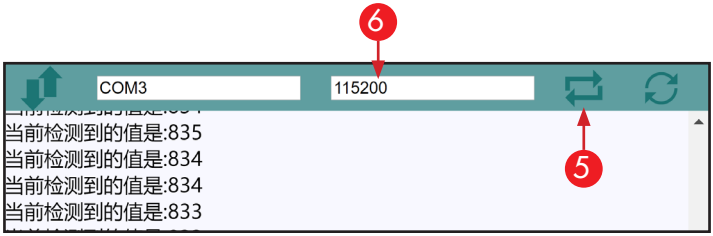
IV - 搭建程序

在此程序中，我们将打印传感器 通道1 检测到的值。

搭建程序的步骤：
1. 拖放红外反射传感器模块到工作区
2. 使用 **设置串口** 积木初始化串行通信端口
3. 使用 **串口打印** 积木打印文本“**当前检测到的值是：**”
4. 使用 **串口打印并换行** 积木打印 **巡线传感器测量** 积木值
5. 添加 100 毫秒的暂停

V - oseppBlock 中的程序：检测路线

1. 启动新工作区
2. 构建以下程序代码

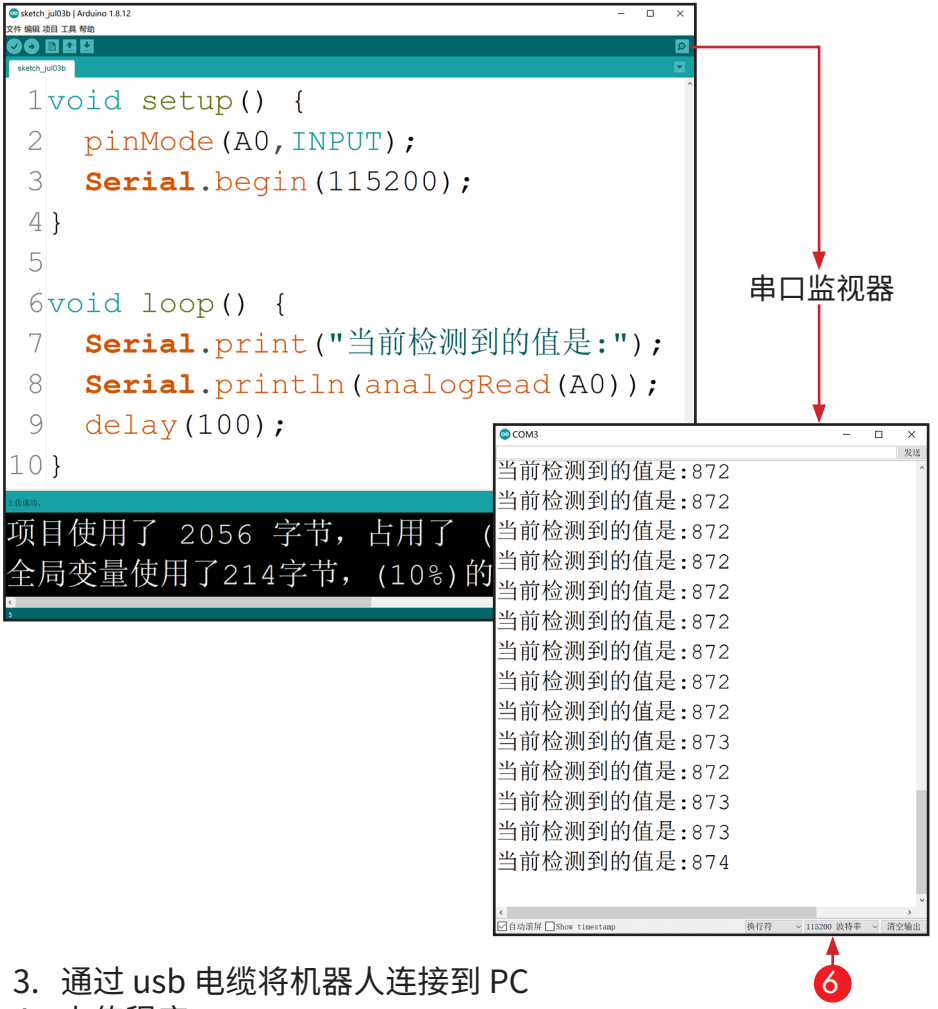


3. 通过 usb 电缆将机器人连接到 PC
4. 上传程序
5. 打开串口监视器
6. 确保选择与**设置串口**积木中相同的波特率，即 115200

将程序加载到机器人上后，将机器人传感器放在黑色胶带上方移动。如果值读数大于 800，则已检测到胶带。假设地板不是黑色，您的地板表面的读数应小于 800。

VI - Arduino IDE 中的程序：检测路线

1. 启动新项目
2. 输入下方的程序代码。



3. 通过 usb 电缆将机器人连接到 PC
4. 上传程序
5. 打开串行监视器 - 单击工具+串行监视器
6. 串口监视器中选择与**Serial.begin**相同的波特率，即 115200

将程序加载到机器人上后，将机器人传感器放在黑色胶带上方移动。如果值读数大于 800，则已检测到胶带。假设地板不是黑色，您的地板表面的读数应小于 800。

14 - 电子围栏

I - 场景：

你的机器人在房子里散步，有时会走出你的视线。那怎样才能让机器人仅在指定区域中活动呢？

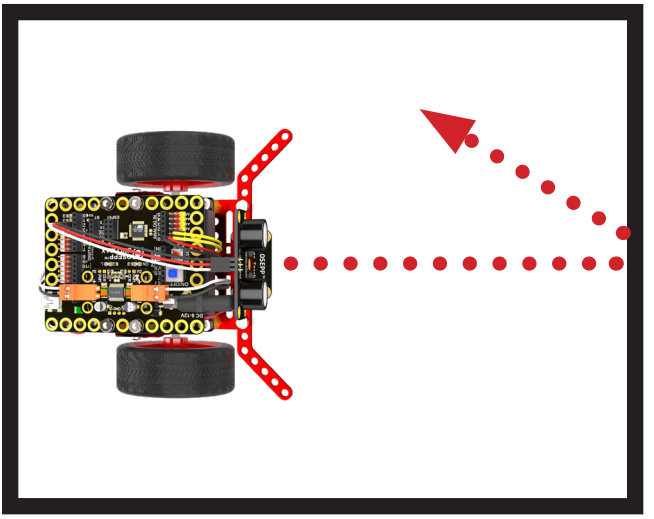
II - 我们如何设置机器人的围栏？

现在，机器人已经学会了如何检测黑色胶带，我们可以用黑色胶带创建边界线，让机器人只在该区域内活动。当机器人感应到边界线时，它会转身，避免越过边界线。

III - 您需要的东西：

选择一片空旷的地板，用标准的黑色电工胶带围出围栏区域。只要机器人能够区分出胶带和地板，您就可以在任何颜色地板上创建围栏。

* 确保巡线传感器 通道2 连接到微控制器A1端口。



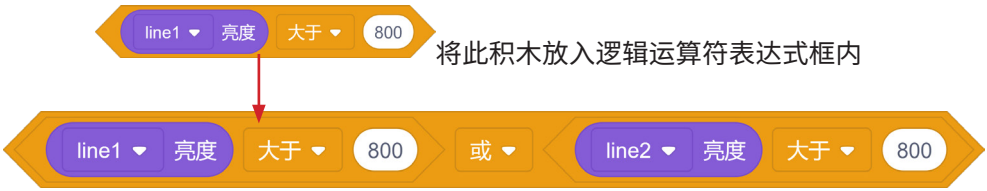
IV - 我们可以学习哪些新代码?

逻辑“或”运算符

逻辑运算符计算两个表达式的逻辑关系。有 2 个主要逻辑运算符:“与运算” 和 “或运算”。本节将重点介绍逻辑 “或” 运算符。

此积木可以在 **数学** 目录菜单中找到。选择逻辑“与”运算符,然后从下拉列表中选择“或”。.

例子:



(巡线传感器 通道1 值大于 800) **或** (通道2 值大于 800)
表达式结果就为True。

(表达式1) || (表达式2)

只要**表达式1**、**表达式2**其中有一个为True,逻辑或运算的结果就是True,如果两个表达式都是False,运算的结果为False。

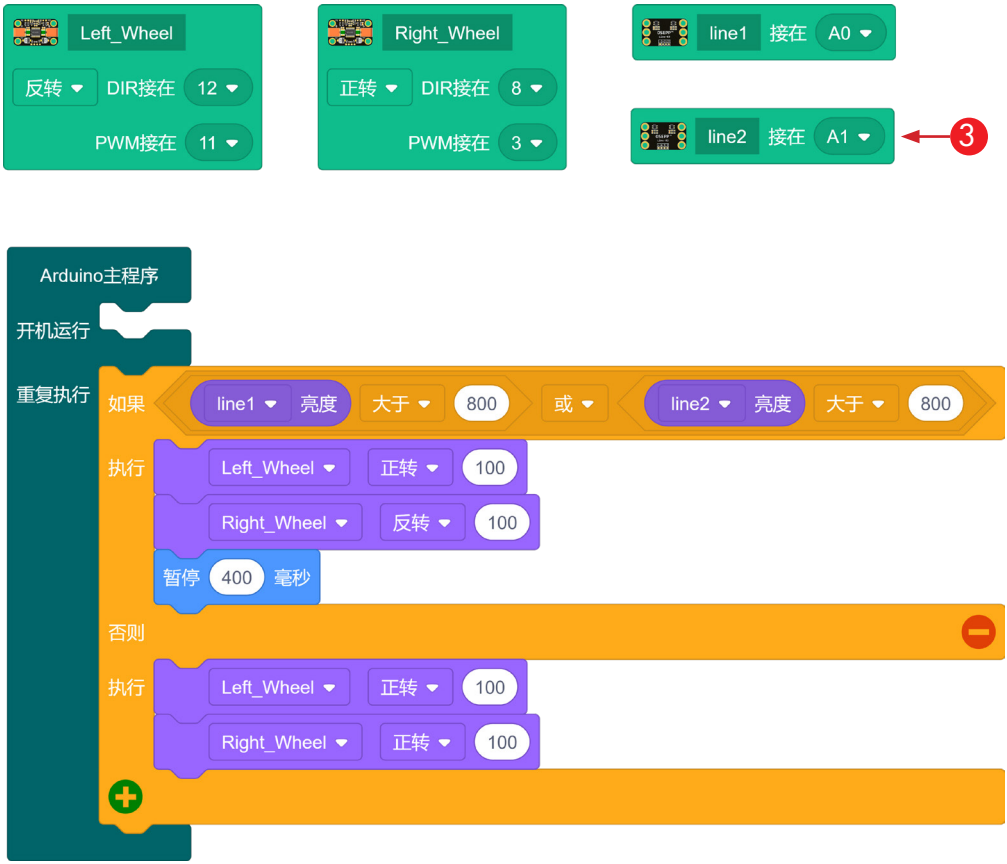
V - 搭建程序

在此程序中,我们将向您展示如何使用逻辑“或”运算符来避开边界线。

以下是此程序的 3 个主要步骤:
1. 测试 - 传感器1或传感器2检测到边界线?
2. 如果结果为 True,则转身,并持续 400 毫秒
3. 否则继续前进

VI - oseppBlock 中的程序：电子围栏

1. 启动新工作区
2. 复制以下代码块
3. 选择并拖动巡线模块通道1和巡线模块通道2积木到工作区

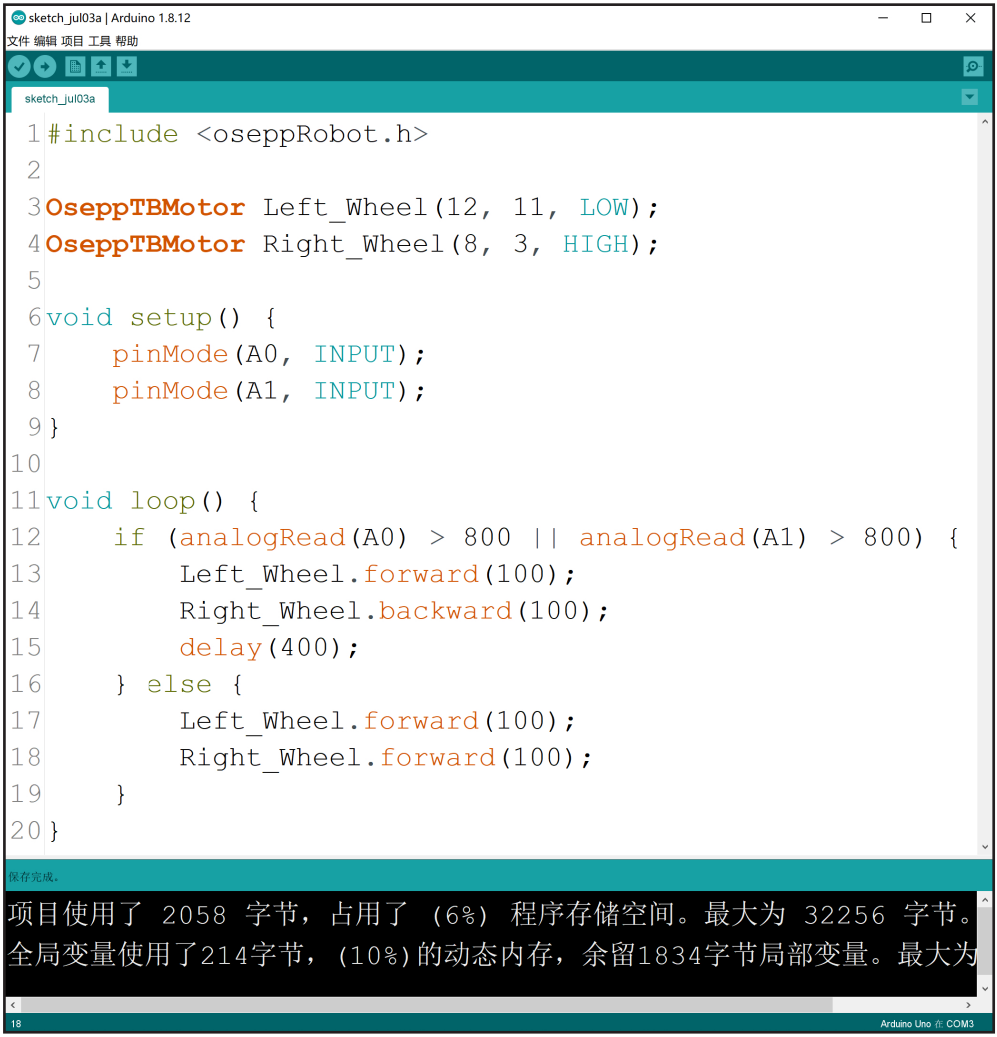


4. 通过 usb 电缆将机器人连接到 PC
5. 上传程序

加载程序后，将机器人放在边界线内并打开电源。机器人将只在边界线内漫游。

VII - Arduino IDE 中的程序：电子围栏

1. 启动新项目
2. 输入以下代码。



3. 通过 usb 电缆将机器人连接到 PC
4. 上传程序

加载程序后，将机器人放在边界线内并打开电源。机器人将只在边界线内漫游。

15 - 巡线机器人

I - 场景：

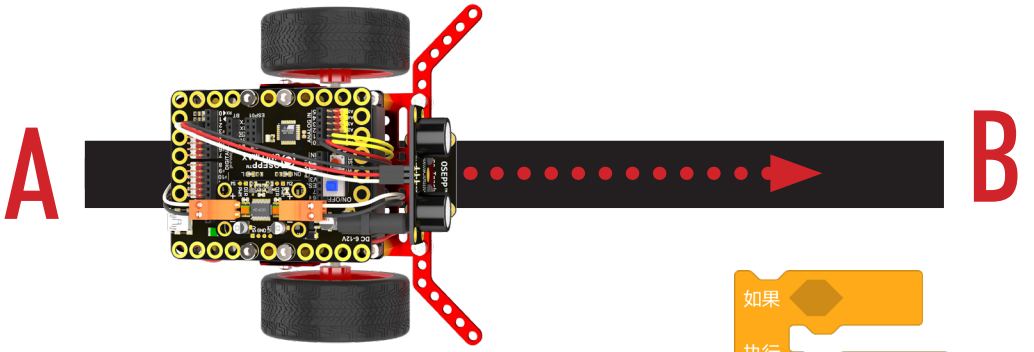
您的机器人正在指定的区域愉快地散步。现在，你有一个机器人的差事，你希望机器人在区域A到区域B之间往返。

II - 我们如何编程机器人从A区旅行到B区？

我们可以创建一个巡线机器人，能够检测线路，并保持行驶在轨道上。

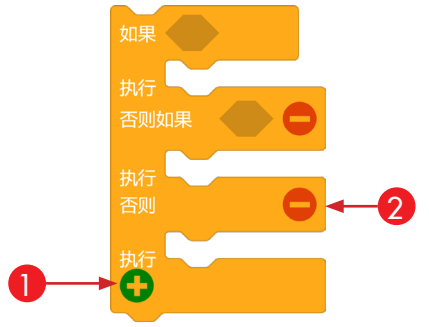
III - 您需要的东西：

选择一片光滑的地板，使用黑色电工胶带，粘贴出一条直线或曲线。



III - 我们可以学习哪些新代码？

否则-如果（else if）



1. 单击图示绿色按钮以添加新的否则-如果分支。
2. 单击分支上的红色按钮删除该分支。

否则-如果（else if）语句是在如果（if）语句之后的另一个附带条件的分支。

如果/否则-如果（If/else if）格式的语句按顺序判断条件，直到遇到条件结果为True的分支，这时将运行其关联的代码块，然后程序跳到整个结构之后的代码。如果所有分支的条件都被证明为False，则执行默认的否则（else）代码块。

没有默认代码时，else分支可以省略。
否则-如果（else if）的分支数量是没有限制的。

逻辑“与”运算符

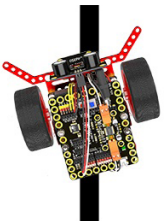
（表达式1）&&（表达式 2）
逻辑“与”运算符仅在表达式1 和表达式2 都为 true 时才给出 true。如果表达式1 或表达式2 或两者都是False，则结果false。

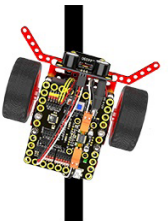
IV - 搭建程序?

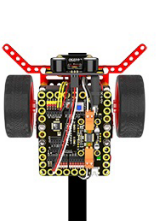
在此程序中，我们将向您展示如何使用逻辑和运算符巡线。根据传感器的输入，机器人会尝试通过向右侧调整或向左侧调整来跟踪线路。

编写此程序有 7 个步骤：
1. 测试 - 如果左侧传感器和右侧传感器都检测到线路
2. 继续前进
3. 否则，如果右侧传感器检测到线路
4. 向右侧调整机器人
5. 否则，如果左侧传感器检测到线路
6. 向左调整机器人
7. 否则原地转身

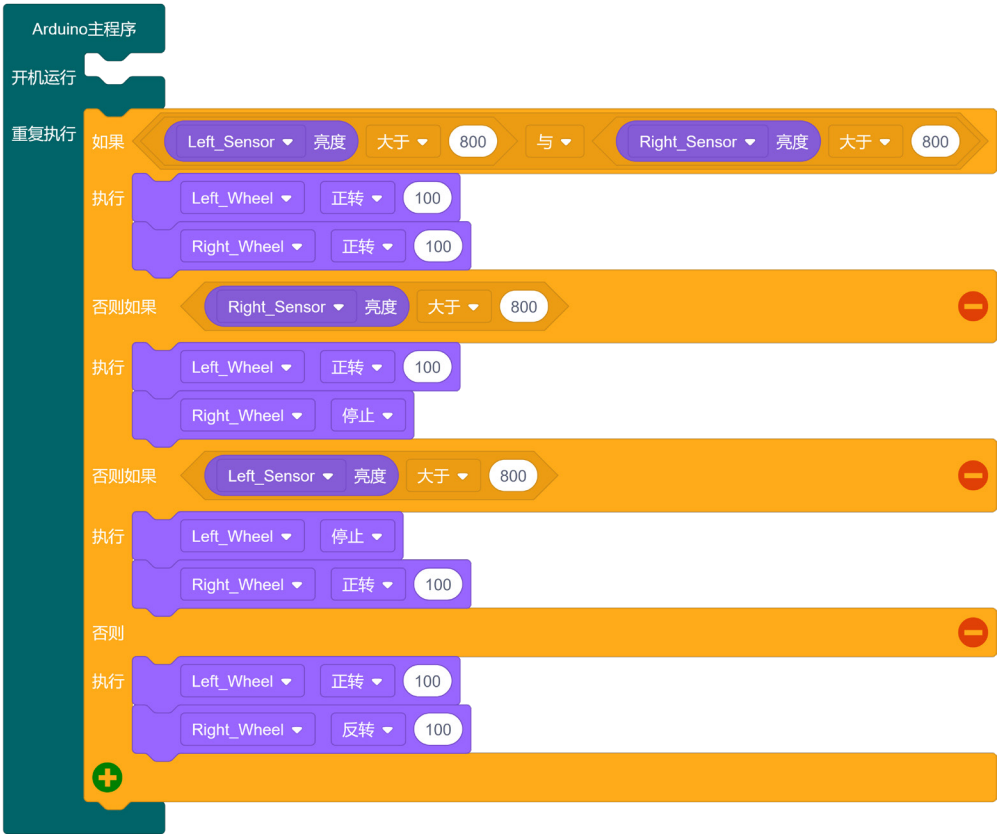

前进

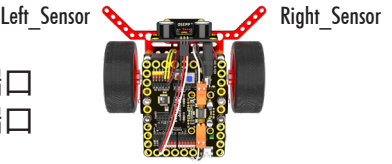

向右调整


向左调整


原地转身

V - 程序 oseppBlock：巡线





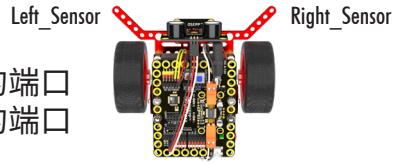
1. 根据传感器连线修改左侧传感器对应的端口
2. 根据传感器连线修改右侧传感器对应的端口

将程序加载到机器人上后，将机器人放在黑线上。你会看到你的机器人跟着线路前进，并保持在轨道上。

提示：检查传感器连线和电池电量。

VI - Arduino IDE：巡线

```
1  #include <oseppRobot.h>
2
3  OseppTBMotor Left_Wheel(12, 11, LOW);
4  OseppTBMotor Right_Wheel(8, 3);
5
6  const int Left_Sensor = A0;
7  const int Right_Sensor = A1;
8
9  void setup() {
10     pinMode(A0, INPUT);
11     pinMode(A1, INPUT);
12 }
13
14 void loop() {
15     if (analogRead(Left_Sensor) > 800 && analogRead(Right_Sensor) > 800) {
16         Left_Wheel.forward(100);
17         Right_Wheel.forward(100);
18     } else if (analogRead(Right_Sensor) > 800) {
19         Left_Wheel.forward(100);
20         Right_Wheel.forward(0);
21     } else if (analogRead(Left_Sensor) > 800) {
22         Left_Wheel.forward(0);
23         Right_Wheel.forward(100);
24     } else {
25         Left_Wheel.forward(100);
26         Right_Wheel.backward(100);
27     }
28 }
```



第6行 - 根据传感器连线修改左侧传感器对应的端口
第7行 - 根据传感器连线修改右侧传感器对应的端口

将程序加载到机器人上后，将机器人放在黑线上。你会看到你的机器人跟着线路前进，并保持在轨道上。

提示：检查传感器连线和电池电量。

16 - 巡线与避障

I - 方案：

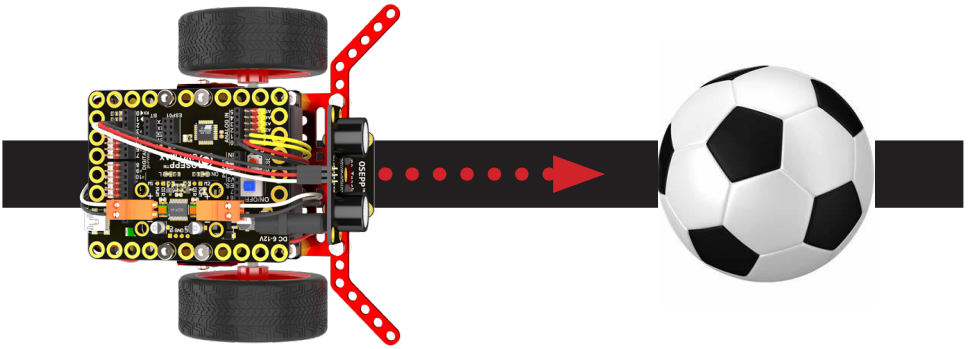
你的机器人正沿着从 A 区到 B 区来回奔跑，突然一个足球在线轨道中间翻滚。你的机器人对它视而不见……

机器人该怎么办？

II - 我们如何编程的机器人，以避免这种碰撞？

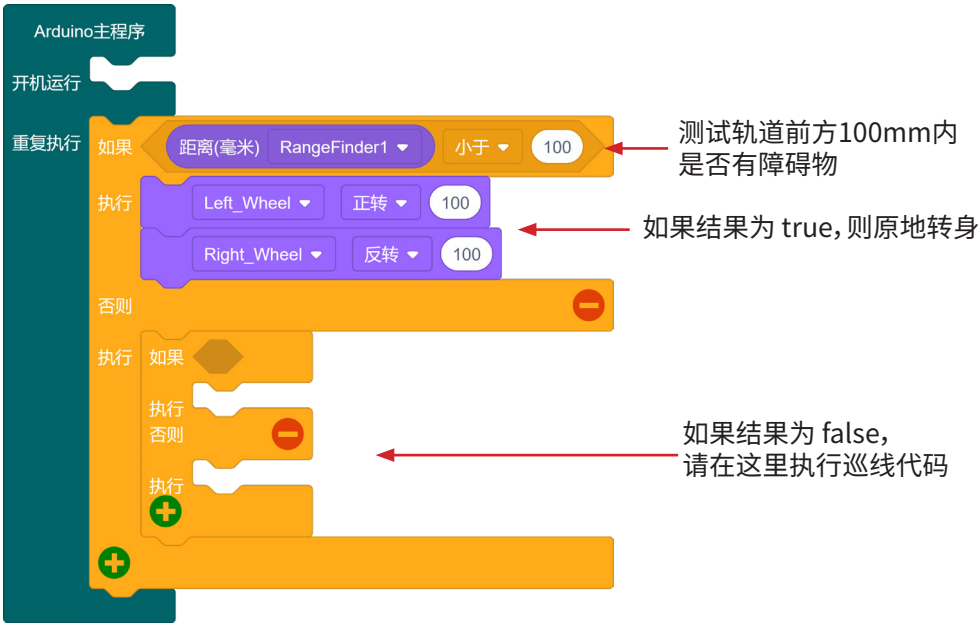
为了避免碰撞，我们必须用超声波传感器将机器人的避障功能还回去。如果它看到道路上的障碍，它会转身。

将障碍物放在轨道中间，对机器人进行编程。



III - 我们可以学习哪些新代码?

我们将应用我们学到的所有代码,并将其应用于此程序。我们将向您展示如何将避障机器人和巡线机器人的功能组合到一个程序文件中。此外,我们将向您展示如何将多个 if/else 组合在一起。
例子:

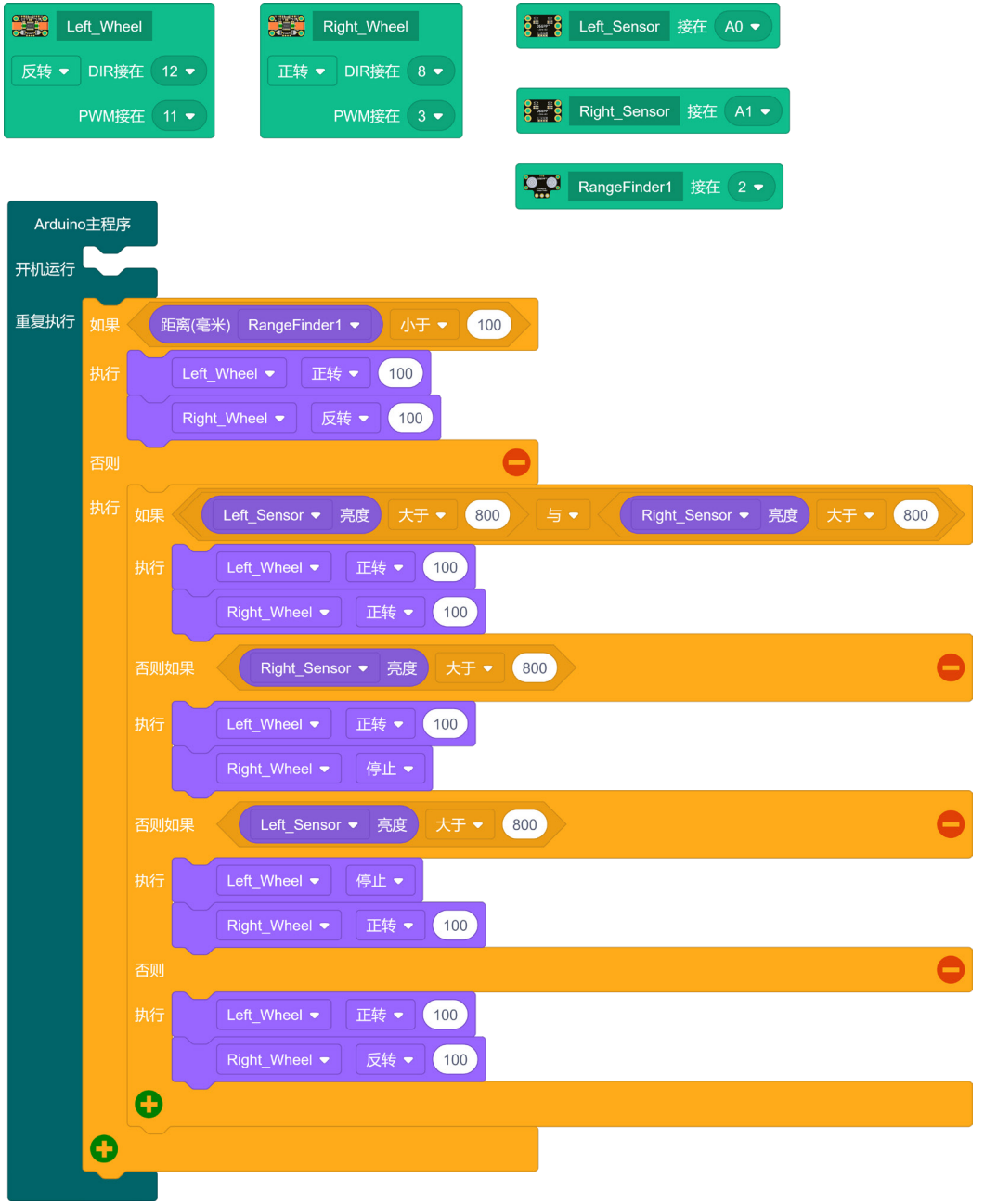


III - 搭建程序

从上一个程序继续,并添加另一个 if/else 语句。
- 如果障碍物距离小于 100 毫米远,机器人原地转身。

编写此程序有 9 个步骤:
1. 测试 - 如果障碍物距离小于 100 毫米?
2. 原地转身
3. 测试 - 如果左侧巡线传感器和右侧巡线传感器同时检测到线路?
4. 继续前进
5. 否则,如果右侧巡线传感器检测到线路
6. 向右侧调整机器人
7. 否则,如果左侧巡线传感器检测到线路
8. 向左侧调整机器人
9. 否则原地转身

IV - oseppBlock：巡线与避障



将程序加载到机器人上后，将机器人放在轨道上。机器人将尝试沿轨道前进，并在检测到轨道上的障碍物时转身。

步骤 V - Arduino IDE：巡线与避障

```
#include <oseppRobot.h>

OseppTBMotor Left_Wheel(12, 11, LOW);
OseppTBMotor Right_Wheel(8, 3);
OseppRangeFinder RangeFinder1(2);

const int Left_Sensor = A0;
const int Right_Sensor = A1;

void setup() {
    pinMode(A0, INPUT);
    pinMode(A1, INPUT);
}

void loop() {
    if (RangeFinder1.ping() < 100) {
        Left_Wheel.forward(100);
        Right_Wheel.backward(100);
    } else {
        if (analogRead(Left_Sensor) > 800 && analogRead(Right_Sensor) > 800) {
            Left_Wheel.forward(100);
            Right_Wheel.forward(100);
        } else if (analogRead(Right_Sensor) > 800) {
            Left_Wheel.forward(100);
            Right_Wheel.forward(0);
        } else if (analogRead(Left_Sensor) > 800) {
            Left_Wheel.forward(0);
            Right_Wheel.forward(100);
        } else {
            Left_Wheel.forward(100);
            Right_Wheel.backward(100);
        }
    }
}
```

将程序加载到机器人上后，将机器人放在轨道上。机器人将尝试沿轨道前进，并在检测到轨道上的障碍物时转身。

17 - 下一步是什么?

我们学习了以下内容:

- 如何驱动机器人前进并停止
- 如何创建避障机器人
- 如何创建机器人电子围栏
- 如何创建巡线机器人
- 如何创建巡线避障结合的机器人

接下来呢?

嗯,还有很多东西可以探索。我们可以探索控制机器人的不同方法,也可以尝试向机器人添加功能。

由于此机器人是基于 Arduino,您可以轻松地添加在 Arduino 社区中找到的任何类型的 Arduino 兼容功能。例如:

您可以添加一些传感器,并使用传感器数据来更改机器人的行为。您可以添加一个蓝牙模块,并尝试在手机上的蓝牙控制,或者您可以添加一个wifi模块通过网络来控制机器人!

通过编程扩展您的知识,使用机器人创建有趣的项目:

1. 创建一个相扑机器人,挑战朋友相扑机器人战斗。
2. 创建一个巡线车队和别人比赛。
3. 创建迷宫求解机器人。

更多教程请查看: https://cn.osepp.com/tutorial/robopro/

保持联系

与我们保持联系，获取最新的产品信息、免费赠品和竞赛。这也是让我们知道你在想什么的一个很好的方式。我们喜欢从客户那里听到我们做得正确，但最重要的是，我们如何改进我们的产品和服务！

https://www.facebook.com/OSEPP.ArduinoCompatible

https://twitter.com/DIY_w_OSEPP

Subscribe by going to our website

支持

我们在 OSEPP 的工程师团队不能保证我们销售的每一种产品都完美无缺，我们一路上都会犯错。因此，我们付出了很多努力，确保所有投诉、评论和疑虑得到及时处理，让客户满意！

我们努力提供业内”最佳售后服务支持”

以下是您联系我们的联系方式：

Technical Support	support@osepp.com
General Inquiries	info@osepp.com
Sales Related	sales@osepp.com