

乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人 编程教学活动 第二版



目录

简介	3
课程	11
评估	16



自动泊车

24-32



安全倒车

33-42



自动大灯

43-53



线检测

54-65



目标探测

66-76



解锁一辆汽车

77-88



巡航控制

89-100



漫游机车

101-112



智能交叉路口

113-115

简介

乐高®教育开发的这本教材可帮助教师围绕无人驾驶汽车相关的科技话题，设计并准备激动人心的课堂教学。

我们旨在通过引导学生在真实情境下编制程序解决方案，来帮助他们开发计算思维技能。



目标群体

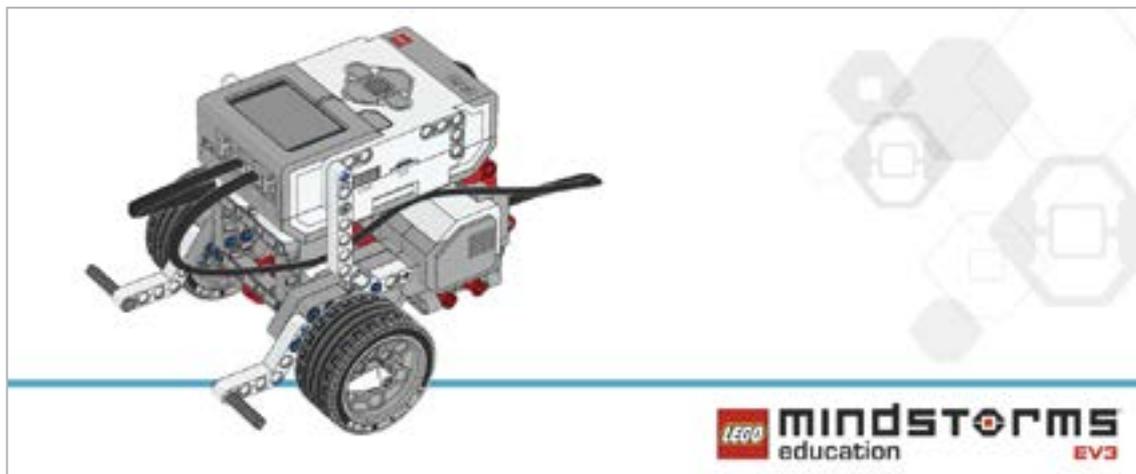
这本教材采用直接教学、探索和教程三者相结合的方式，帮助教师向学生们介绍计算机编程的相关概念，其中教程内容基于乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人软件和编程应用程序。

该教材面向 6-8 年级的学生，稍作调整既可用于更高年级的学生。

课程结构

这些课程主要通过探索实际生活中与无人驾驶汽车相关的主题，阐释一系列重要的计算机编程概念。通过这些课程的学习，学生的设计和计算思维技能将得到充分培养。这本教材包含许多跨学科内容，涉及科学、数学、工程设计技术等多个学科领域。

所有课程均基于 Robot Educator 基本模型及其扩展。可选用任何驱动基底(driving base)设计。向学生们提供丰富多样的搭建体验，是一种提升他们设计技能的很好途径。



每门课程均包含一个关于程序结构和设计的讨论环节。

Robot Educator教程

每门课程都需要学生很好地理解EV3软件或编程应用程序。这一教学目标可通过让学生们学习基础Robot Educator教程来实现。借助直接教学和探索的教学方法，确保学生掌握编程技能，理解必需的知识，从而能够顺利开展课程中的大部分任务。

活动流程

可通过多种方式在您的课堂上开展EV3编程活动。该教材中所涵盖的编程概念就是按照难度递增的方式安排的。当然，您可以最适合自己课堂的方式来安排教学内容。您还可根据自己的时间，自由调整课程。

每节课大概持续：

- 60分钟，使用预先搭建好的机器人，专门学习编程
- 90分钟 (2x45分钟)，包括搭建、记录和分享
- 135分钟 (3x45分钟)，包括搭建、记录、分享，以及基于文本程序的比较。

基于文本的编程

针对该系列活动，乐高®教育选择ROBOTC作为基于文本的编程语言的示例。您也可选择其他与EV3机器人兼容的基于文本的编程语言。

本教材的主要目标不在于指导教师和学生如何使用基于文本的编程语言，而在于说明EV3程序与基于文本的解决方案在可视化方面的等效性。想了解更多关于ROBOTC的信息，请访问：<http://www.robotc.net/>

在可能的情况下，ROBOTC程序可以准确实现EV3程序的功能。然而，由于两种编程语言的本质区别，故在运用过程中会稍存在一些内在差异。在相应的ROBOTC程序中，任何被识别的差异部分都会以绿色文字突显出来。

EV3编程应用程序

每个EV3编程活动均可通过EV3软件和EV3编程应用程序进行（平板电脑仅能搭载基本编程功能）。下表列出了哪些课程可由示例代码实现，这些代码均来自各自的源程序。

课程	自动泊车	安全倒车	自动大灯	线检测	目标探测	解锁一辆汽车	巡航控制	漫游机车
EV3 软件	X	X	X	X	X	X	X	X
EV3 机器人编程应用程序	X	X	X	X	X			X
Robot C	X	X	X	X	X	X	X	X

EV3编程应用程序中尚不包含完成解锁一辆汽车和巡航控制课程所需的变量模块和阵列模块。

科学、计算思维、编程

科学和工程领域在人类早期就已出现，相比之下，计算机科学属于新兴学科。尽管如此，这个新兴的学科不仅影响着我们探索科学和工程的方式，而且影响着我们的生活方式。

计算机科学属于STEM（科学、技术、工程和数学）学科，同时具有科学、技术、工程和数学的属性。所有STEM学科都有助于培养学生的思维方式和终身技能。这些技能包括提出问题、设计解决方案和交流结果的能力。

计算思维也是这些技能中的一种。这是人们的一种思考方式，每个人都可利用它来解决问题。计算思维可被描述为一系列技能的集合，而其中就包括算法思维。“代码”或“编程”可用于描述构建算法的行为。

因此，编程是STEM学科背景下培养学生计算思维的一种手段。

STEM学科

科学、技术、工程、数学、计算机科学

培养思维方式和终身技能

1. 提出问题并解决问题。
2. 使用模型。
3. 设计原型。
4. 探究。
5. 分析和解释数据。
6. 运用计算思维。

- a. 分解
- b. 抽象
- c. 算法思维（代码）
- d. 评估
- e. 概括

7. 利用证据进行论证。
8. 获取、评估和交流信息。

什么是计算思维?

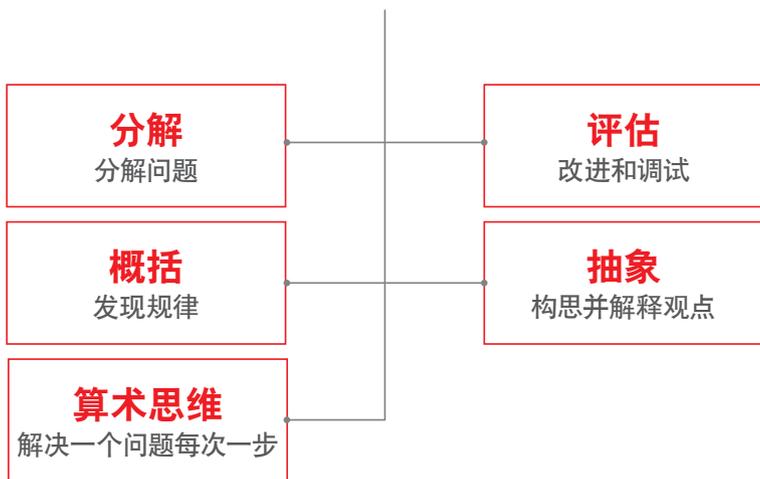
“计算思维”这个词最先由 Seymour Papert 提出, 后由周以真教授 (Jeannette Wing) 推广。她将计算思维定义为:

“计算思维过程涵盖问题界定及相应解决方案的提出, 这样就可以借助一种信息处理工具, 将这些解决方案以一种可被有效实施的形式呈现出来。” (Wing, 2011)

计算思维可应用于各种领域和环境, 以及日常生活。科学、工程和数学领域都涉及计算思维技能。这些技能可以定义为以下能力:

计算思维

我们解决问题的方式



分解

分解是指把问题分成较小的部分, 从而简化问题的能力。通过这样, 问题会变得更易于向别人解释, 或被分解成具体的任务。问题分解之后通常需要进行概括。

例如: 度假时, 可将准备工作 (或项目) 分解为多个子任务: 预订机票、预定宾馆、打包行李等。

概括 (发现规律)

概括能力是指发现任务中已知的部分, 或是在其他地方已经了解过的部分。这常会使设计算法变得更加简单。

例如: 交通灯的工作模式就是不断重复一系列相同的操作。

算法思维 (编程)

算法思维是指设计一系列有序步骤来解决问题的能力。

示例一：按照菜谱做菜时，我们会按照一系列步骤烹饪饭菜。

示例二：使用计算机时，我们可以通过编程来设计一系列需要计算机完成的动作。

评估或调试

它是指检验原型是否能按预期运行的能力，还指当原型不能按照预期运行时，发现待改进之处的能力。

计算机编程人员也借此来发现和更正程序中的错误。

示例一：做菜时，我们会不时尝尝，看看味道是否适宜。

示例二：查找书面作业中的拼写错误和遗漏的标点符号，对其进行修改，从而实现整篇作业的顺利阅读。

抽象化

抽象化是指在解释问题或解决方案时去掉不重要细节的能力。也就是说，将想法概念化的能力。

例如：描述一辆自行车时，我们仅需着重描述它的一些细节。我们可能会描述它的型号和颜色，如果对方确实对自行车感兴趣，我们可再进行更细致地描述。

培养计算思维技能的过程

运用工程设计流程

工程师们运用设计流程来寻找问题的解决方案。他们会通过多个阶段来引导自己找出解决方案。每个阶段都会用到或培养他们的某些技能。这些技能就是我们所谓的“计算思维”。

学生在完成EV3编程活动的过程中会采用类似流程：

明确问题

向学生们展示一个主题，该主题可将他们引入一个问题或一种需要改善的情况。有时，可将问题阐释的尽量详细些。为了使问题更容易解决，可以将问题分解成较小的部分。学生可通过简单的方式明确问题，并确定一些成功标准，来培养自身的“分解”技能。

换言之：

- 学生能否用自己的语言解释问题？
- 学生能否描述如何判断自己的解决方案是否成功？
- 学生能否将问题分解成多个更小、更便于处理的部分？

规划

学生应花一些时间构想问题的不同解决方案，然后制定一份详细的计划来执行其中一种方案。为达到这一目的，他们需要确定一系列实施步骤。通过识别任务中可能见过的部分，可培养他们的“概括”技能。

换言之：

- 学生能否列出一系列需要编程实现的动作？
- 学生能否从现有程序中识别出他们可以使用的部分？
- 学生能否重新使用程序中的部分代码？

尝试

接下来，每位学生都需要确定自己的最终解决方案。在这个阶段，他们需要使用图形化编程语言激活乐高®模型。当学生对自己的想法进行编程时，可培养自身的“算法思维”技能。

换言之：

- 学生能否将问题的解决方案编写成一个程序？
- 学生能否利用序列、循环、条件语句等？

修改

学生要根据自己的程序和模型是否符合成功标准来评估自己的解决方案。通过利用自己的评估技能，他们可确定是否需要更改、修改、调试或改进程序的某些部分。

换言之：

- 学生是否在他们的程序中运用迭代？
- 学生是否修复了程序中的问题？
- 学生能否判断解决方案是否与问题相关？

沟通

学生要向同学们展示自己最终的解决方案，并解释自己的解决方案为何符合成功标准。通过详略适当地解释自己的解决方案，他们可以培养自身的抽象化和沟通技能。

换言之：

- 学生能否解释解决方案中最重要的部分？
- 为使自己的解决方案更易理解，学生是否提供了足够的细节？
- 学生能否清楚说明他们的解决方案为何能够满足成功标准？

通过编程培养计算思维

为了培养学生们的算法思维，将向他们介绍一些编程原理。在设计解决方案的过程中，他们会组织一系列动作和结构，实现为自己的模型赋予生命。

学生会用到的最常见编程原理包括：

1. 输出

输出是由学生所编写程序控制的一系列信号。MINDSTORMS 中的输出示例包括声音、光线、显示及开启和关闭电机等。

2. 输入

输入是计算机或设备所接收的信息。可以通过传感器以数字或文本值的形式进行输入。例如，传感器检测或测量一些数值（如距离），然后将那些值转换为数字输入信号，使它们可以在程序中使用。

3. 事件（等待）

学生可以让程序等待特定事件发生后，再继续执行后续动作。程序可以等待一定时间，或是等待传感器检测到特定事物。

4. 循环

学生可对动作进行编程，使其无限重复或重复特定次数。

5. 函数

函数是指特定情况下可整体使用的一组动作。例如，一组可进行光线闪烁的模块被统称为“闪烁函数”。

6. 条件

对于那些只能在特定情况下才可执行的动作，学生可借助条件来对其编程。在程序中创建条件意味着：如果该条件一直没有被满足，那么其中一部分程序将永远不会被执行。例如，如果触动传感器被按压，电机将开启；如果被松开，电机将停止。

课程标准匹配

在美国及许多其他国家中，教育界和工商界领袖们正积极号召将更多的编程体验引入 K-12 课堂。计算机科学教师协会 (CSTA) 于 2011 年建立了一套计算机科学标准，并于 2016 年进行了修订。

出于这种考虑，乐高® 教育开发出了该系列课程，借助乐高 MINDSTORMS® EV3 机器人来帮助 5-8 年级的学生学习这个抽象的科目。



课程	学习目标	编程概念	核心的EV3编程所涵盖模块
自动泊车	<ul style="list-style-type: none"> - 了解算法能够执行一系列指令 - 通过对比轮式机器人不同的移动方式, 探究输出的概念 	<ul style="list-style-type: none"> - 设计可自动安全泊车且无需驾驶员介入的汽车 	<ul style="list-style-type: none"> - 移动槽 - 等待 - 触动传感器 - 程序块状态灯
安全倒车	<ul style="list-style-type: none"> - 深入了解运用算法能够执行一系列指令 - 加深对输出的理解 	<ul style="list-style-type: none"> - 为汽车设计一些功能, 可帮助提高倒车安全性 	<ul style="list-style-type: none"> - 移动转向 - 等待 - 程序块按钮 - 声音 - 显示
自动大灯	<ul style="list-style-type: none"> - 探究输入的概念及其控制方式 - 探究等待函数的概念 	<ul style="list-style-type: none"> - 为汽车设计一些功能, 可帮助提高夜间驾驶安全 	<ul style="list-style-type: none"> - 等待 - 颜色传感器 - 循环 - 循环中断
线检测	<ul style="list-style-type: none"> - 探究循环的概念 - 了解开关语句的概念, 以及如何利用它进行“真”和“伪”运算 	<ul style="list-style-type: none"> - 设计一些防止驾驶员打瞌睡的方法, 以提高驾驶安全性, 避免事故 	<ul style="list-style-type: none"> - 等待 - 颜色传感器 - 循环 - 切换 - 循环中断
目标探测	<ul style="list-style-type: none"> - 加深对循环的理解 	<ul style="list-style-type: none"> - 设计一些方法可避免车辆与路间物体相撞的事故 	<ul style="list-style-type: none"> - 等待 - 超声波传感器 - 循环 - 切换
解锁一辆汽车	<ul style="list-style-type: none"> - 了解简单的布尔逻辑(如“与”、“或”和“非”), 以及其在电路和编程中的应用 - 运用几种输入组合 	<ul style="list-style-type: none"> - 设计一种方式, 可利用密码来防止汽车被盗 	<ul style="list-style-type: none"> - 超声波传感器 - 程序块按钮 - 逻辑 - 切换 - 循环
巡航控制	<ul style="list-style-type: none"> - 使用变量模块来存储信息 - 编写多层程序 - 创建功能模块(我的模块) 	<ul style="list-style-type: none"> - 设计一个巡航控制程序来帮助驾驶员, 减轻他们的驾驶压力 	<ul style="list-style-type: none"> - 触动传感器 - 循环 - 切换 - 变量 - 数学 - 我的模块
漫游汽车	<ul style="list-style-type: none"> - 合理使用数据结构, 如线性表、表格和数组。 - 深入了解布尔逻辑及其在电路和编程中的应用 - 使用变量模块来存储信息 - 使用阵列运算模块 	<ul style="list-style-type: none"> - 设计一辆无人驾驶小汽车, 可十分安全地行驶在街道上 	<ul style="list-style-type: none"> - 变量 - 程序块按钮 - 循环 - 阵列运算 - 我的模块
项目	<ul style="list-style-type: none"> - 学生们可融汇和运用在前面编程课程中学到的知识。 - 针对现实生活中的问题和实物系统, 设计、使用和评估相应的解决方案 	<ul style="list-style-type: none"> - 设计一辆可以安全穿过交叉路口的无人驾驶汽车 	<ul style="list-style-type: none"> - 以上都是

课程列表

《新一代科学教育标准》(NGSS)	自动泊车	安全倒车	自动大灯	颜色与线检测	目标探测	解锁一辆汽车	巡航控制	漫游机车
<p>● = addresses standard</p> <p>◐ = partially addresses standard</p>								

MS.工程设计

MS-ETS1-1.	确定一个设计问题的标准和限制，并针对性地提出问题和足够精确地界定问题，以确保获得成功的解决方案，同时还要考虑相关的科学原理及对人类和自然环境的潜在影响，这些均会限制可能的解决方案。	●	●	●	●	●	●	●
MS-ETS1-2.	利用一个系统化流程来评估一些竞争性设计方案，从而确定这些方案对目标问题标准和限制的满足程度	●	●	●	●	●	●	●
MS-ETS1-3.	通过分析测试的数据来确定几个设计方案的相同点和不同点，可从中获取每种设计的最佳特征，而这些特征可被纳入新的解决方案，使该方案更好地满足成功的条件	●	●	●	●	●	●	●
MS-ETS1-4.	开发一个模型，其可通过对提议物体、工具或流程进行重复测试和修改，生成相关数据，基于此可获得一个最优的设计			●	●	●	●	●

NGSS 标准框架下的科学和工程练习

练习 1	提出问题（科学）和确定问题（工程）	●	●	●	●	●	●	●
练习 2	开发和使用模型	●	●	●	●	●	●	●
练习 3	规划并进行探究							
练习 4	分析和解释数据							
练习 5	运用数学和计算思维	●	●	●	●	●	●	●
练习 6	构思阐释内容（科学）和设计解决方案（工程）	●	●	●	●	●	●	●
练习 7	通过证据进行论证							
练习 8	获取、评估和交流信息	●	●	●	●	●	●	●

课程列表

计算机科学教师协会 (CSTA) 标准 (2016年)		自动泊车	安全倒车	自动大灯	颜色与线检测	目标探测	解锁一辆汽车	巡航控制	漫游机车
● = addresses standard ◐ = partially addresses standard									
算法与编程									
2-A-2-1	征求和综合同伴的反馈信息, 进行程序开发和改进	●	●	●	●	●	●	●	●
2-A-7-2	比较那些用于解决同一问题的不同算法, 比较时可参考它们的速度、明晰度和大小 (例如, 解决同一问题可采用不同算法, 但其中一种可能会比其它的快)。(说明: 不要求学生量化这些不同之处)	●	●	●	●	●	●	●	●
2-A-7-3	借鉴代码或以该代码作为开发基础时, 要提供合理的出处	●	●	●	●	●	●	●	●
2-A-7-4	讲解算法实施的流程并预测运算的结果。(说明: 可采用自然语言、控制流程图、代码中的注释和伪代码来表达算法)	●	●	●	●	●	●	●	●
2-A-5-5	设计、开发和展示计算产品 (computational artifact), 例如可通过独立或协作方式解决社会问题的移动应用程序	●	●	●	●	●	●	●	●
2-A-5-6	以独立或协作方式开发多个序列的程序, 这些序列自身包含多个嵌入式循环和子程序。(说明: 在这一水平, 学生可能需要使用基于模块和/或基于文本的编程语言)			●	●	●	●	●	●
2-A-5-7	创建代表不同类型数据的变量, 并操作它们的值。						●	●	●
2-A-4-8	定义和使用程序, 这些程序可以隐藏一项任务的复杂性, 并可以被再次使用去解决相似的任务。(说明: 学生可以使用和修改带有多个参数的程序, 但无须去创建它们)								●
2-A-3-9	将一个问题分解成多个较小的部分, 并针对每个部分创建解决方案	●	●	●	●	●	●	●	●
2-A-6-10	采用重复式设计流程 (如确定问题、产生想法、构建、测试和改进解决方案) 来解决问题, 包括独立或协作方式	●	●	●	●	●	●	●	●
计算系统									
2-C-7-11	证明被选择用于完成一项任务的软硬件的合理性 (例如, 比较平板电脑和台式电脑的性能, 选择哪些传感器和平台用于机器人搭建或开发一个移动应用程序)								
2-C-4-12	分析设备的计算组件与其性能之间的关系。(说明: 计算系统不仅包括计算机, 还包括汽车、微波炉、智能手机、交通灯和闪存盘)	●	●	●	●	●	●	●	●
2-C-6-13	利用一个系统化流程来确定单个设备或连接设备中问题的根源 (例如, 遵循故障排除流程图, 改动软件以查看硬件能否正常工作, 重启设备, 检查连接, 更换工作组件)	◐	◐	◐	◐	◐	◐	◐	◐

课程列表

计算机科学教师协会 (CSTA) 标准 (2016年)		自动泊车	安全倒车	自动大灯	颜色与线检测	目标探测	解锁一辆汽车	巡航控制	漫游机车
● = addresses standard ◐ = partially addresses standard									
数据和分析									
2-D-7-14	描述存储数据的不同格式如何体现其质量与大小之间的权衡。(说明: 比较音乐、文本和/或图像格式的示例 数据)								
2-D-7-15	解释那些利用计算工具解决问题的流程, 这些流程用于收集、转换和分析数据 (如利用应用程序或电子表格收集数据, 决定使用或忽略哪些数据, 并选择一种可视化方法)	◐	◐	◐	◐	◐	◐	◐	◐
2-D-5-16	修正计算模型使其能够更准确地反映真实世界的系统 (如生态系统、流行病、思想传播)	●	●	●	●	●	●	●	●
2-D-4-17	采用不同的编码方案表示数据 (如二进制、Unicode、摩尔斯电码、速记、学生创建的代码)								
计算的影响									
2-I-7-18	针对利用数据和信息对人类分类、预测行为并根据那些预测提出建议的现象, 就其消极和积极影响进行总结 (例如定制化搜索结果或基于过往浏览历史的定向广告可以节省搜索时间, 同时还可以减少选项)								
2-I-7-19	解释计算机科学如何促进创新及几乎所有事业和学科的发展	◐	◐	◐	◐	◐	◐	◐	◐
2-I-1-20	提供计算产品和设备如何影响人们健康与幸福的实例, 包括积极和消极方面	●	●	●	●	●	●	●	●
2-I-1-21	描述互联网以何种方式影响全球交流与合作								
2-I-1-22	描述与计算设备和网络相关的伦理问题 (例如平等可入、安全性、黑客、知识产权、版权、知识共享许可和剽窃)								
2-I-6-23	重新设计一个计算产品, 以消除向大众普及的障碍 (如使用图像标题、高对比色和/或大号字体)								◐
网络与互联网									
2-N-7-24	总结与弱密码、无加密措施、不安全交易和持久性数据相关的安全风险						◐		
2-N-7-25	模拟信息如何以数据包的形式传送到互联网与网络上的多个设备上。						◐		

评估

您可以通过多种方式对学生在课程中的进度进行了解和评估。这部分内容涉及一种观察评估表，您可借助它就学生的计算思维技能提升情况，向他们提供反馈。



学生主导的评估

鼓励学生讲述自己的学习过程。通过学生手中的文档工具，向他们提供分享自己思维、观点和反思的机会。学生可使用文本、视频、图像、草图笔记或其它创意性媒介记录他们的想法。允许学生选择最适合自己的工具，以获取和分享他们独特的思维、创造及学习过程。

教师指导评估

培养学生的科学、工程和计算思维技能需要充足的时间和相应的反馈。就像在设计环节中，学生需要知道“失败乃成功之母”一样，评估应当向学生就他们的成功之处和需要改进的地方提供反馈。以问题为导向的学习不在于成功或失败，而在于成为一名主动的学习者，不断地构想和尝试新的想法。

可通过多种方式为学生提供反馈，从而帮助他们提升自己的技能。为辅助 EV3 编程活动，我们还提供了一些评价标准的范例，标准内容涉及：

- 观察学生的行为、反应和策略
- 就学生的思维过程提出问题

由于学生通常会被分成小组，故可以分别针对小组和个人提供反馈。

观察评估表

下页章节提供了计算思维评估表的使用范例。针对每位学生或每个小组，您可以使用观察评估表进行：

- 评估学生在每一个步骤中的表现
- 提供建设性反馈意见，以帮助每位学生取得进步

评估表分为以下几个等级：



1.铜 (初级)

对于内容知识及其理解与应用, 以及/或针对特定主题的连贯思维表达方面, 学生尚处于初级水平。



2.银 (中级)

学生能够运用一些基础知识 (如专业术语) 进行表达, 但尚无法应用内容知识或充分理解教材中的概念。



3.金 (熟练)

学生已完全掌握相关内容和概念, 并能够熟练运用所学习的主题、内容或概念。但是缺乏讨论和运用实验要求以外知识的能力。



4.铂金 (高级)

学生对概念和观点的认识达到另一个层次, 能将相关概念应用于其他情况, 还可对所学的知识进行有机组合和应用, 并运用于开阔思维的讨论中。

分解评价标准

	铜	银	金	铂金	备注
运用你自己的语言描述问题。	 学生不能运用他们自己的语言描述问题。	 经过提示，学生能够运用自己的语言描述问题。	 学生能够运用他们自己的语言描述问题。	 学生能够运用自己的语言描述问题并开始将问题分解成较小的部分。	
描述你如何判断自己是否成功找到了问题的解决方案。	学生无法描述成功标准。	经过提示，学生能够成功标准。	学生能够描述成功标准。	学生能够描述成功标准。	
描述如何将问题分解成较小的部分。	学生不能分解问题。	经过提示，该学生能够将问题分解成较小的部分。	学生能够将问题分解成较小的部分。	学生能够将问题分解成较小的部分并可说明各部分之间的联系。	

概括评价标准

	铜	银	金	铂金	备注
请描述你曾从程序库(或其它地方)调用过哪个程序,并解释原因。	 <p>学生不能说出自己使用过的程序及相应调用原因。</p>	 <p>经过提示,学生能够识别出自己调用过的程序。</p>	 <p>学生能够说出自己调用过的程序及相应原因。</p>	 <p>学生能够详细地说出自己调用过的程序及所做过的修改。</p>	
观察学生能否发现规律,或运用之前学过的概念。	学生不能发现规律,或是运用之前学过的概念。	经过提示,学生能够发现规律,或运用之前学过的概念。	学生能够发现规律,或运用之前学过的概念。	学生能够发现规律,或运用自己的概念。	

算法思维评价标准

铜	银	金	铂金	备注
 <p>学生不能描述一系列需实现的动作。</p>	 <p>经过提示，学生能够描述一系列需要实现的动作。</p>	 <p>学生能够描述一系列需要实现的动作。</p>	 <p>学生能够详细列出一系列需要实现的动作，以协助自己进行程序编制。</p>	
<p>请描述一系列需要编程实现的动作。</p>	<p>经过提示，学生可以描述该程序。</p>	<p>学生能够描述该程序。</p>	<p>学生能够描述该程序，并可详细描述程序的每个部分。</p>	
<p>请描述你是如何为自己的解决方案编程的。</p>	<p>经过提示，学生能够描述自己解决方案中用到的编程原理。</p>	<p>学生能够描述自己解决方案中用到的编程原理。</p>	<p>学生能够充分理解并描述自己解决方案中用到的编程原理。</p>	
<p>请描述你的解决方案中用到的编程原理（例如输出、输入、事件、循环等）。</p>				

评估评价标准

	铜	银	金	铂金	备注
请描述在执行程序时发生了什么,并说明这是否符合你的预期。	 学生不能描述发生了什么。	 经过提示,学生能够描述发生了什么,并可将其与预期作比较。	 学生能够描述发生了什么,并可将其与预期作比较。	 学生能够描述发生了什么,可将其与预期作比较,并且找到了解决方案。	
请描述你是如何修复程序中的问题的。	学生不能描述自己是如何修复问题的。	经过提示,学生能够描述自己是如何修复问题的。	学生能够描述自己是如何修复问题的。	学生能够非常详细地描述自己是如何修复问题的。	
请描述你的解决方案是如何与问题联系起来的。	学生不能描述自己的解决方案是如何与问题联系起来的。	经过提示,学生能够描述自己的解决方案是如何与问题联系起来的。	学生能够描述自己的解决方案是如何与问题联系起来的。	学生能够非常详细地描述自己的解决方案是如何与问题联系起来的。	
请描述你所尝试过的不同解决方案。	学生不能描述自己尝试过的不同解决方案。	经过提示,学生能够描述自己尝试过的不同解决方案。	学生能够描述自己尝试过的不同解决方案。	学生能够描述自己尝试过的不同解决方案,并可逐一解释每种方案行不通的原因。	

抽象化评价标准

	铜	银	金	铂金	备注
请描述你的解决方案中最重要的一部分。	 <p>学生不能描述自己解决方案中的任何部分。</p>	 <p>经过提示，学生能够描述自己的解决方案。</p>	 <p>学生能够描述自己的解决方案。</p>	 <p>学生能够描述自己的解决方案，并可重点说明解决方案中最重要的一部分。</p>	
请描述你的解决方案中的重要细节。	 <p>学生不能描述关于自己解决方案的任何细节。</p>	 <p>经过提示，学生能够描述与自己解决方案相关的细节。</p>	 <p>学生能够描述与自己解决方案相关的细节，但其中一些是非重要细节。</p>	 <p>学生能够描述解决方案中的重要细节。</p>	
请描述你的解决方案是如何满足最初设计标准的。	 <p>学生不能描述自己的解决方案是如何满足最初设计标准的。</p>	 <p>经过提示，学生能够描述自己的解决方案是如何满足最初设计标准的。</p>	 <p>学生能够描述自己的解决方案是如何满足最初设计标准的。</p>	 <p>学生能够非常清晰地描述自己的解决方案是如何满足最初设计标准的。</p>	



自动泊车

设计可自动安全泊车且无需驾驶员介入的汽车。

学习目标

学生将:

了解算法能够按序执行一系列指令
通过对比轮式机器人不同的移动方式探究输出的概念

词汇

输出、算法、伪代码、调试

年级

6-8

学科

工程、STEM (科学、技术、工程和数学)、编程

课程持续时间

45-90分钟

难度

初学者

标准

《新一代科学教育标准》(NGSS)
MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. / MS-ETS1-4

计算机科学教师协会 (CSTA) 标准
2-A-2-1 / 2-A-7-1 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-3-9 /
2-A-6-10 / 2-C-4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高®MINDSTORMS®头脑风暴教育 EV3 机器人核心套装
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC软件 (可选)

自动泊车

联系 (5分钟)

围绕下列问题进行展开课堂讨论:

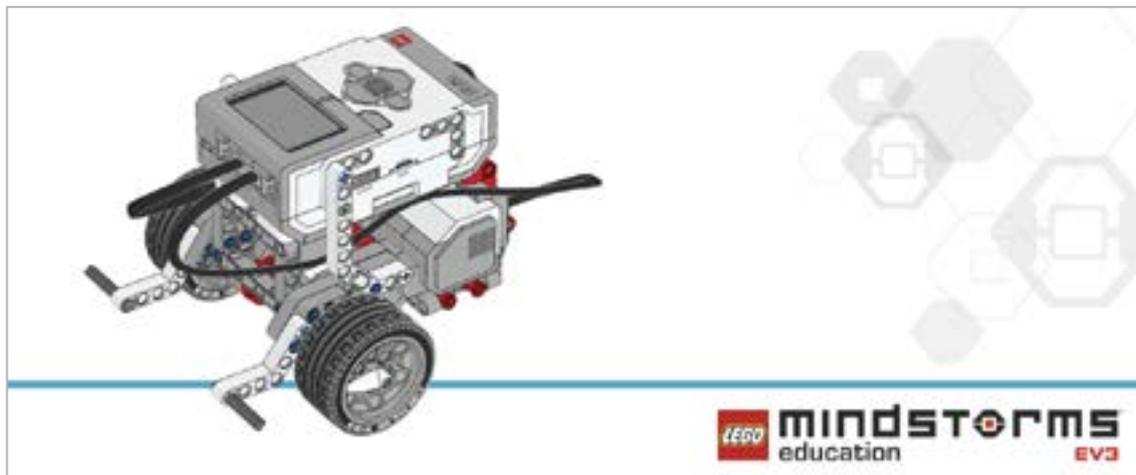
- 无人驾驶汽车的工作原理是什么?
- 怎样才能确保无人驾驶汽车的安全?
- 无人驾驶汽车需要进行哪些类型的动作?

允许学生选择最适合自己的获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式, 将自己的想法记录下来。

建构 (15到30分钟)

搭建

学生将会构建 Robot Educator 基本模型, 该模型是一款基础的带轮机器人。



在对机器人进行编程之前, 先让学生对搭建的系统进行检查:

- 连接线是否正确地从电机连接到端口 B 和 C?
- 机器人的轮子是否正确安装?
- A 的轮子能否自由旋转?

程序

让学生在EV3编程环境中新建一个项目。

让他们创建一个程序进行练习，需实现让机器人转向三次。

鼓励学生探索使带轮机器人移动的不同方式。让他们描述改变所使用每个模块的参数对机器人的影响。

可能的解决方案

文件名称：CODING-01.EV3 (Tab:1)



三点调头

1.启动程序

2.转动驱动基底并在1.5秒后停止。

3.使驱动基底向左转向并在1秒后停止。

4.使驱动基底向前移动3秒。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

备注

可让学生参阅 Robot Educator 教程以获取更多帮助。

在EV3机器人软件中：

Robot Educator > Basics > Straight Move

Robot Educator > Basics > Curved Move

Robot Educator > Basics > Tank Move

在EV3机器人编程应用程序中：

Robot Educator > Curved Move

活动准备

在分派下一个任务之前，请确保您已标定机器人的行进路径，并为任务开展预留了足够空间。最好能让学生在宽大的工作台或地板上开展任务。

反思 (35分钟)

让学生选择一个或所有这些无人驾驶情境，并对其编程：

- 平行停车
- 斜角停车
- 非字形停车

对于每种解决方案，他们应该采用不同的编程画布。

允许学生选择最适合自己的获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称: CODING-01.EV3 (Tab:2)



自动平行停车

- 1.启动程序
- 2.以所需速度沿直线前进
- 3.等待1秒
- 4.反转电机, 转动车轮1.5圈
- 5.转向另一个方向时, 反转电机, 转动车轮1.5圈
- 6.向后直线移动0.5圈
- 7.向前直线行进1圈

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分层教学部分

让学生创建一个程序, 模拟停车时适用的各种警示灯 (例如, 使用 EV3 程序块状态灯来模拟倒车警示灯)。

可能的解决方案

文件名称: CSACTIVITY1.EV3 (Tab:3)



模拟倒档及倒档警示灯

- 1.启动程序
- 2.以所需速度沿直线前进
- 3.等待1秒
- 4.打开灯 (倒档警示灯)
- 5.反转电机, 转动车轮1.5圈
- 6.转向另一个方向时, 反转电机, 转动车轮1.5圈
- 7.向后直线移动0.5圈
- 8.向前直线行进1圈

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分享

允许学生选择最适合自己的工具，以获取和分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

要求一到两个小组展示他们的程序。讨论成功之处和需要改进的方面。

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案，这样他们可以比较不同的编程语言。

重要提示：

下文提供了一个可能的解决方案，其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权，故就用户体验和技术方面的质量问题，不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>，获取所需的全部安装信息。当您使用其他编程语言完成活动时，我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-01_1.C (Three Point Turn)

```
#pragma config(StandardModel, "EV3_REMBOBOT")
/*
使用定时和转向功能实现一次三点调头。
*/

task main()
{
    //向右转并在1.5秒后停止。
    setMotorSpeed(motorB, 75);
    setMotorSpeed(motorC, 30);
    sleep(1500);

    //向左侧倒车并在1秒后停止。
    setMotorSpeed(motorB, -30);
    setMotorSpeed(motorC, -75);
    sleep(1000);

    //现在车身已于道路平行，并朝相反方向。开车离去。
    setMotorSpeed(motorB, 50);
    setMotorSpeed(motorC, 50);
    sleep(3000);
}
```

本课程的程序解决方案可从此处下载:

<http://www.legoeducation.com>



下一步是什么?

安全倒车



自动泊车

学生学习单

设计可自动安全泊车且无需驾驶员介入的汽车。

学生学习单

联系

确保你能回答下列问题：

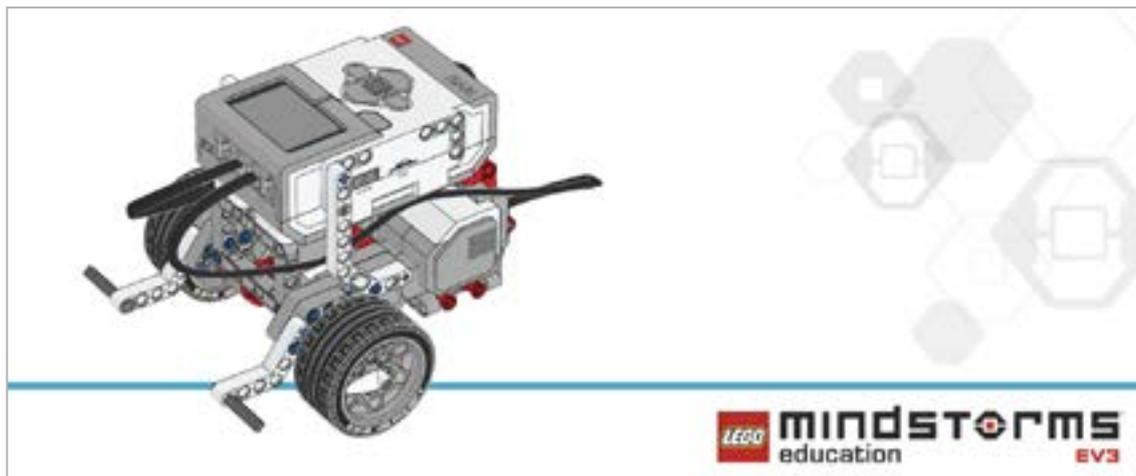
- 无人驾驶汽车的工作原理是什么？
- 怎样才能确保无人驾驶汽车的安全？
- 无人驾驶汽车需要进行哪些类型的动作？

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



程序

编制一个程序，可实现将机器人以不同的方式转动三次。

思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

反思

从下列无人驾驶情境选择一种, 并对其进行编程:

- 平行停车
- 斜角停车
- 非字形停车。

思考你所了解的情况, 然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学

创建一个程序, 模拟展示停车时适用的各种警示灯(例如, 使用程序块状态灯来模拟倒车警示灯)。

分享

思考你所了解的情况, 然后进行记录。创意性地录制和展示你的想法、创作和发现。

拓展

探究用于本次课程的基于文本的编程解决方案, 并对比这些采用不同编程语言的解决方案。



安全倒车

为汽车设计一些功能,可帮助提高倒车安全性。

学习目标

学生将:

深入了解运用算法能够按序执行一系列指令
加深对输出的理解

词汇

输入、输出、算法、伪代码、调试、等待

年级

6-8

学科

工程、STEM (科学、技术、工程和数学)、编程

课程持续时间

45-90分钟

难度

中等

标准

《新一代科学教育标准》(NGSS)

MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. / MS-ETS1-4

计算机科学教师协会 (CSTA) 标准

2-A-2-1 / 2-A-7-1 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-3-9 /
2-A-6-10 / 2-C-4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高®MINDSTORMS®头脑风暴教育EV3机器人核心套装。
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC 软件 (可选)

安全倒车

联系(5分钟)

在本次课程中, 学生将使用程序块状态灯和显示块对自己的轮式机器人进行编程, 以模拟倒车时向行人、其他驾驶员和乘客发出的可视化信号。学生还会使用触动传感器模拟前进档和倒档。

围绕下列问题进行展开课堂讨论:

- 汽车或其它机动车辆倒车时会出现什么危险?
- 你所在区域区域因倒车而引起事故的统计数据如何?
- 机动车辆如何实现更安全地倒车?
- 当汽车倒车时, 其外部发生了什么可让行人和其他驾驶员知悉该车将如何行驶。

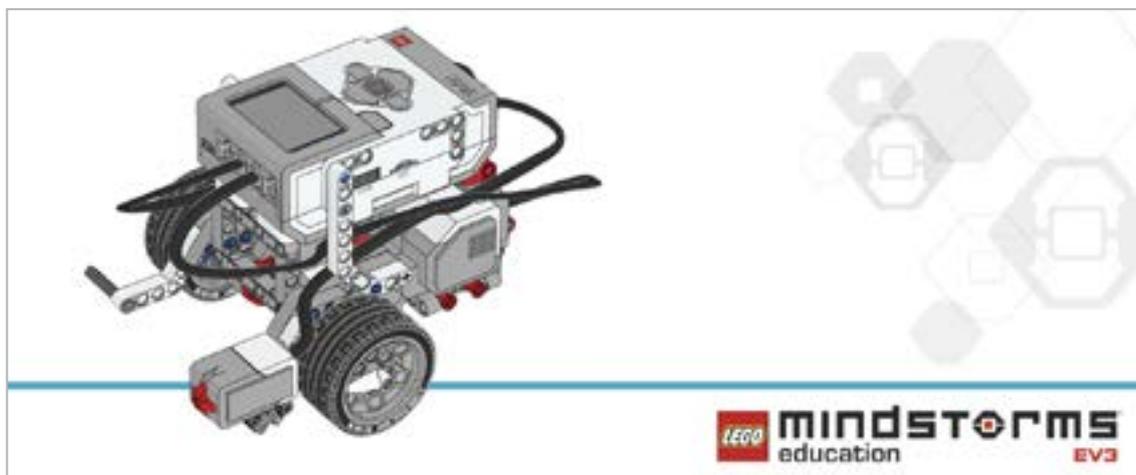
让学生简要讨论汽车倒车时会发什么。讨论一辆汽车的倒车灯如何警示其它汽车驾驶员和行人它将如何行驶。向学生讲解, 仪表盘上的各种指示器可以向驾驶员和乘客传递信息, 如汽车挂的哪个档、行驶方向等。

允许学生选择最适合自己的获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式, 将自己的想法记录下来。

建构 (15到30分钟)

搭建

学生将会构建 Robot Educator 基本模型, 然后添加触动传感器。



在对机器人进行编程之前,先让学生对搭建的系统进行检查:

- 连接线是否正确地从电机连接到端口 B 和 C?
- 机器人的轮子是否正确安装?
- 机器人的轮子能否自由旋转?
- 连接线是否正确地从触动传感器连接到端口1?

程序

让学生对自己的机器人进行编程,以模拟倒车及指示灯的使用。

让学生将触动传感器作为保险杠或制动器,以实现让机器人转向。

允许学生选择最适合自己的获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称: CODING-02.EV3 (Tab:1)



碰撞与倒车

- 1.启动程序。
- 2.以所需速度沿直线向前行驶。
- 3.等待触动传感器被“碰撞”(模拟汽车从前进档切换到倒档)。
- 4.停止电机。
- 5.等待1秒。
- 6.启动“倒车灯”(EV3程序块上的琥珀色指示灯)。
- 7.反向行驶2秒。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

备注

您可以借助 EV3 软件来演示程序块状态灯和拥有三种不同模式的触动传感器（即按压、松开和碰撞）。可让学生参阅 Robot Educator 教程以获取更多帮助。

在 EV3 机器人软件中：

- Robot Educator > Hardware > Brick Status Light
- Robot Educator > Hardware > Brick Display
- Robot Educator > Hardware > Touch Sensor

活动准备

在分派下一个任务之前，请确保您已为任务开展预留了足够空间。最好能让学生在宽大的工作台或地板上开展任务。

反思 (35分钟)

介绍将 EV3 程序块上的按钮作为“倒”档。

要求学生思考屏幕如何可作为全自动汽车的一部分。讨论屏幕可以哪些方式将重要信息传递给汽车乘客。

学生可通过下列方式编制程序，模拟汽车的前进档和倒档，从而巩固之前学过的知识：

- 使用触动传感器作为保险杠以停止机器人。
- 使用程序块按钮使机器人向后运动。
- 使机器人就其动作发出指示。

允许学生选择最适合自己的获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称：CODING-02.EV3 (Tab:2)



碰撞并等待倒档

1. 启动程序。
2. 以所需速度沿直线向前行驶。
3. 等待触动传感器被“碰撞”（模拟汽车从前进档切换到倒档）。
4. 停止电机。
5. 等待程序块按钮被按压（模拟汽车从前进档切换到倒档）。
6. 启动“倒车灯”（EV3程序块上的琥珀色指示灯）。
7. 反向行驶2秒。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

分层教学部分

让学生将 EV3 程序块作为仪表盘指示器。

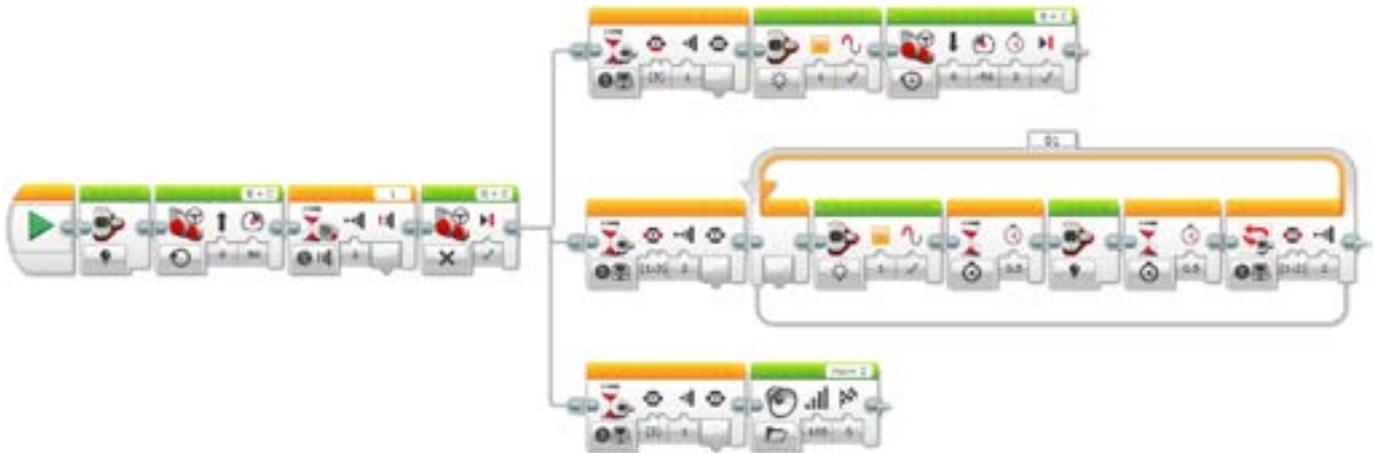
他们将会进一步编制程序，通过显示块把 EV3 程序块屏幕引入，这样轮式机器人在进行前进或后退动作时会触发屏幕，显示出机器人前进方向的箭头。

让学生组合其它信号，例如：

- 转向灯（向左或向右转向时发出闪烁光线）
- 喇叭

可能的解决方案

文件名称：CODING-02.EV3 (Tab:3)



碰撞并等待带指示灯的倒档

- 1.启动程序。
- 2.关闭 EV3 程序块上的指示灯。
- 3.以所需速度沿直线向前行驶。
- 4.等待触动传感器被“碰撞”（模拟汽车从前进档切换到倒档）。
- 5.停止电机并同步进行3个任务。

任务1

- 6.等待向下的程序块按钮被按压（模拟汽车从前进档切换到倒档）。
- 7.启动“倒车灯”（EV3 程序块上的琥珀色指示灯）。
- 8.反向行驶2秒。

任务2

- 9.等待的向左或向右的程序块按钮被“碰撞”。
- 10.启动 EV3 程序块上的琥珀色指示灯。
- 11.等待 0.5 秒。
- 12.关闭 EV3 程序块上的指示灯。
- 13.等待 0.5 秒。
- 14.重复步骤 x 至 x，直到向左或向右的程序块按钮再次被“碰撞”。

任务3

- 15.等待返回程序块按钮（模拟喇叭）受到按压。
- 16.让 Horn 2 发声。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

分享

允许学生选择最适合自己的工具，以获取和分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

要求一到两个小组展示他们的程序，以及他们的轮式机器人如何运动。

整个小组内有多少种不同的解决方案？针对给定的问题，对比多种可能的解决方案。

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案，这样他们可以比较不同的编程语言。

重要提示：

下文提供了一个可能的解决方案，其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权，故就用户体验和技术方面的质量问题，不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>，获取所需的全部安装信息。当您使用其他编程语言完成活动时，我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-02_1.C

```
#pragma config(StandardModel, "EV3 _ REMBOT")
/*
创建一个程序, 实现机器人持续向前移动, 直至按压到触动传感器。随后, 机器人向后运动, 并闪烁橙色LED灯。
*/
task main()
{
    //将电机速度设定为50%。
    setMotorSpeed(motorB, 50);
    setMotorSpeed(motorC, 50);
    //等待触动传感器受到按压。
    while(getTouchValue(touchSensor) ==0)
    {
        sleep(10);
    }
    //将电机速度设定为0%或关闭电机。等待1秒。
    setMotorSpeed(motorB, 0);
    setMotorSpeed(motorC, 0);
    sleep(1000);
    //EV3橙色LED灯开始闪烁。
    setLEDColor(ledOrangeFlash);
    //将电机速度设定为-50% (反向移动)。等待2秒。
    setMotorSpeed(motorB, -50);
    setMotorSpeed(motorC, -50);
    sleep(2000);
}
```

本课程的程序解决方案可从此处下载:
<http://www.legoeducation.com>



下一步是什么?

自动大灯



安全倒车

学生学习单

为汽车设计一些功能，可帮助提高倒车安全性。

学生学习单

联系

确保你能回答下列问题：

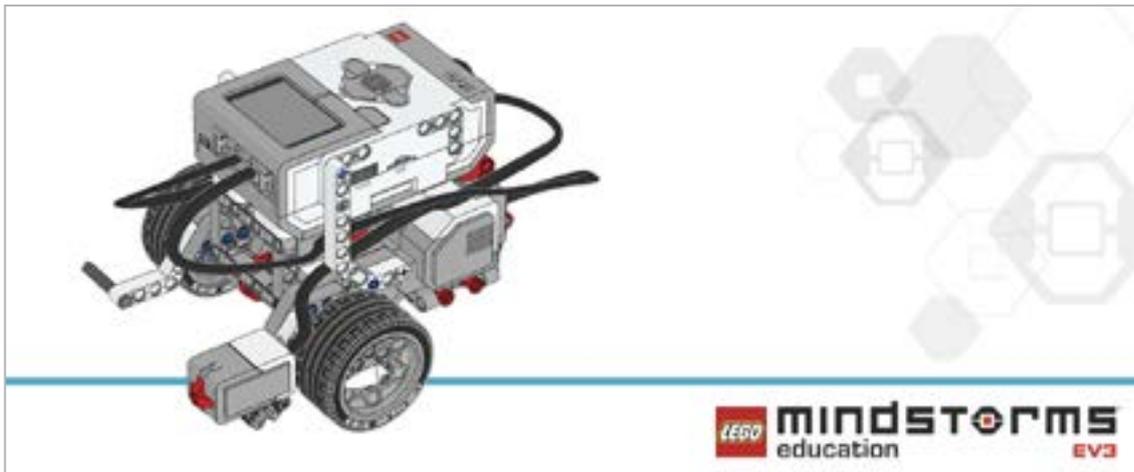
- 汽车或其它机动车辆倒车时会出现什么危险？
- 你所在区域因倒车而引起事故的统计数据如何？
- 机动车辆如何实现更安全地倒车？
- 当汽车倒车时，其外部发生了什么可让行人和其他驾驶员知悉该车将如何行驶。

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



程序

编写一个程序，可驱动轮式机器人向前移动，其按压到触动传感器后转为向后移动。

你的轮式机器人应该配有倒车警示灯
使用程序块状态灯来模拟倒车警示灯。

考虑将这些模块应用于你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

反思

拓展程序, 让你的轮式机器人拥有更多功能:

- 使用触动传感器作为保险杠以停止机器人。
- 使用程序块按钮使机器人向后运动。
- 使机器人就其动作发出指示。

考虑将该模块应用于你的解决方案:



思考你所了解的情况, 然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学部分

将新的信号加入你的程序, 例如:

- 转向灯 (向左或向右转向时发出闪烁光线)
- 喇叭

分享

思考你所了解的情况, 然后进行记录。创意性地录制和展示你的想法、创作和发现。

请思考以下问题:

你如何能够改进自己的程序?

你是否使用很多模块? 是否存在一种更有效的程序构建方式?

拓展

探究用于本次课程的基于文本的编程解决方案, 并对比这些采用不同编程语言的解决方案。



自动大灯

为汽车设计一些功能，可帮助提高夜间驾驶安全。

学习目标

学生将：

探究输入的概念及其控制方式
探究等待函数的概念

词汇

输入、输出、伪代码、等待、颜色传感器、调试、环境光线、循环、布尔逻辑、并行编程

年级

6-8

学科

工程、STEM（科学、技术、工程和数学）、编程

课程持续时间

45-90分钟

难度

中等

标准

《新一代科学教育标准》(NGSS)
MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. /MS-ETS1-4.

计算机科学教师协会 (CSTA) 标准
2-A-2-1 / 2-A-7-2 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-5-6 /
2-A-3-9 / 2-A-6-10 / 2-C-7-11 / 2-C-4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高®MINDSTORMS®头脑风暴教育EV3机器人核心套装。
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC软件（可选）。
直接光源（如手电筒）

自动大灯

联系 (5分钟)

- 夜间驾驶存在哪些危险?
- 我们如何界定的环境光线的概念?
- 在驾驶员夜间驾驶过程中, 如何能够帮助他们?

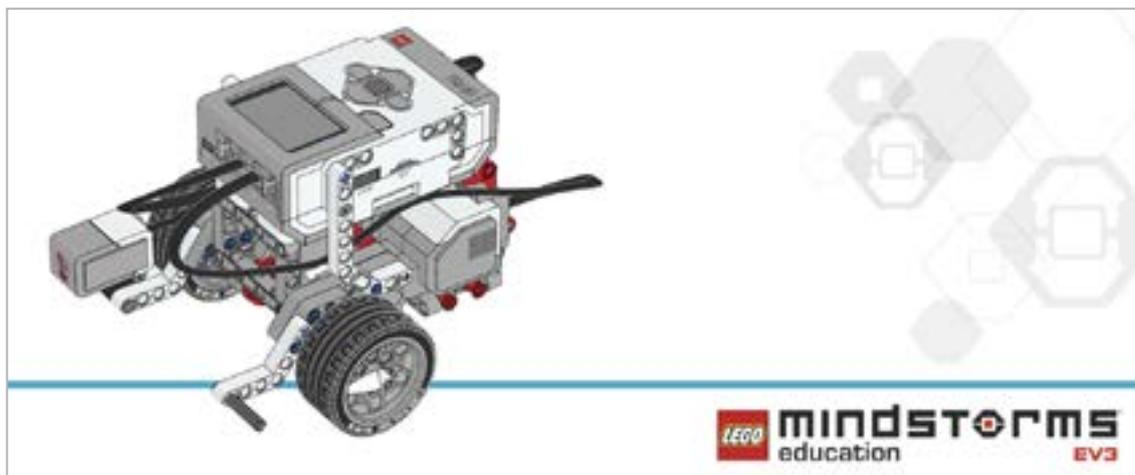
告知学生, 他们将在本次课程中探索环境光线的变化如何触发程序的响应。此外, 通过决策来控制程序就是布尔逻辑运用的一个实例。

允许学生选择最适合自己的获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式, 将自己的想法记录下来。

建构 (15到30分钟)

搭建

学生将会构建 Robot Educator 基本模型, 然后添加颜色传感器, 并将其面向前方。



在对机器人进行编程之前, 先让学生对搭建的系统进行检查:

- 连接线是否正确地从电机连接到端口 B 和 C?
- 机器人的轮子是否正确安装?
- 机器人的轮子能否自由旋转?
- 连接线是否正确地从颜色传感器连接到端口 3?

程序

让学生在EV3编程环境中新建一个项目。

向学生介绍颜色传感器，并展示它的三种模式：

- 颜色模式，可以识别乐高®系统颜色
- 环境光强度模式，可以测量光线强度
- 反射光强度模式，用来测量从颜色传感器发射并从附近表面反射回的光线的强度

让学生编写程序，使颜色传感器处于环境光强度对比模式，以对自动大灯进行模拟。

学生可以采用各种方法来模拟前大灯，例如：

- 在EV3程序块显示屏上显示一个电灯泡的图像
- 使用EV3程序块上的颜色

允许学生选择最适合自己的获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称: CODING-03.EV3 (Tab:1)



自主环境光线检测

- 1.启动程序。
- 2.等待环境光强度降低，直至低于15。
- 3.在EV3程序块显示屏上显示一只电灯泡。
- 4.等待5秒钟。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

备注

可让学生参阅 Robot Educator 教程以获取更多帮助。

在EV3机器人软件中：

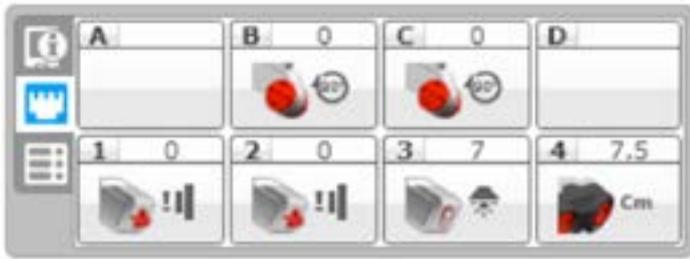
Robot Educator > Basics > Loop

Robot Educator > Hardware > Color Sensor – Light

Robot Educator > Beyond Basics > Multitasking

Robot Educator > Hardware > Brick Display

向学生演示如何在 EV3 软件中使用端口视图获取环境光线的读数 (如显示环境光强度的端口视图, 端口3)。



告知学生, 他们将探究循环是如何工作的, 以及如何将其加入他们的程序。
向学生讲解循环的概念, 以及如何利用循环使程序进行无限运行, 直至手动停止。

活动准备

让学生将手电筒照射到颜色传感器上, 以造成环境光线亮度的改变。

反思 (35分钟)

学生将创建一个程序, 可对环境光强度的变化做出响应, 并可以更加接近地模拟自动大灯。当环境变暗时, 程序将需要打开灯, 而当环境变亮时, 需要关闭灯光。

向学生强调, 他们需要对自己程序中的环境光强度设置进行持续监控和调试, 以确保程序正常运行。

可能的解决方案

文件名称: CODING-03.EV3 (Tab:2)



自主环境光线检测的重复实现

- 1.启动程序。
- 2.等待环境光强度降低, 直至低于15。
- 3.在EV3程序块显示屏上显示一只电灯泡。
- 4.等待环境光强度增强, 直至高于15。
- 5.重置EV3程序块显示屏
- 6.一直重复步骤2至5。

本课程的程序解决方案可从此处下载:
<http://www.LEGOeducation.com>

分层教学部分

学生将采用多任务/并行编程来模拟自动和手动照明开关。

他们可以借助一个触动传感器或程序块按钮来模拟手动开关。

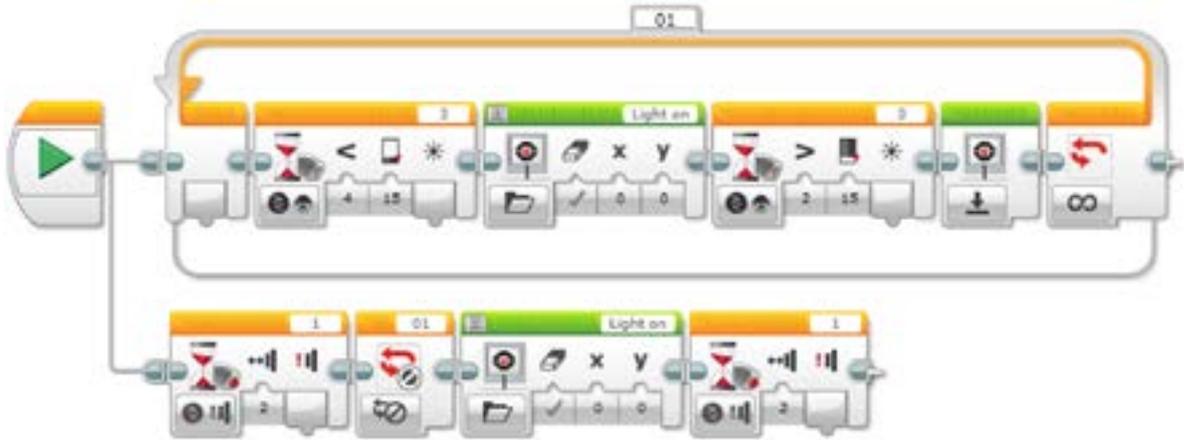
通过使用触动传感器或程序块按钮进行手动操作，可向学生介绍布尔逻辑的应用。

向学生演示如何从开始模块拖出一条线以创建一个并行程序（如多任务）。

向学生展示循环中断模块，并说明我们可以利用该模块终止并行程序中某个循环内任何正在运行的语句。

可能的解决方案

文件名称：CODING-03.EV3 (Tab:3)



自主环境光线检测的手动控制实现

1.启动程序。

任务1

- 2.等待环境光强度降低，直至低于15。
- 3.在EV3程序块显示屏上显示一只电灯泡。
- 4.等待环境光强度增强，直至高于15。
- 5.重置EV3程序块显示屏
- 6.一直重复任务1的步骤2至5。

任务2

- 7.等待触动传感器被“碰撞”（模拟模拟手动灯光开关）。
- 8.中断名为“01”的循环（覆盖自动程序）。
- 9.在EV3程序块显示屏上显示一只电灯泡。
- 10.等待触动传感器被“碰撞”（光线将会因程序结束而消失）。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

分享

允许学生选择最适合自己的工具，以获取和分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

要求一到两个小组展示他们的程序。

要求学生就他们的期望与程序实际运行的结果进行分享。

并询问他们程序运行的结果是否为他们带来了一些惊喜。

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案，这样他们可以比较不同的编程语言。

重要提示：

下文提供了一个可能的解决方案，其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权，故就用户体验和技术方面的质量问题，不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>，获取所需的全部安装信息。当您使用其他编程语言完成活动时，我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-03_1.C

```
#pragma config(Sensor, S1, touchSensor, sensorEV3 _ Touch)
#pragma config(Sensor, S4, sonarSensor, sensorEV3 _ Ultrasonic)
#pragma config(Motor, motorB, rightMotor,tmotorEV3 _ Large, PIDControl,
driveRight, encoder)
#pragma config(Motor, motorC, leftMotor, tmotorEV3 _ Large, PIDControl,
driveLeft, encoder)
```

```
/*
```

创建一个程序, 能够实现: 当环境光线的强度低于一定程度时, 可利用颜色传感器自动打开灯。

```
*/
```

```
task main()
{
    //等待环境光强度增强, 直至低于15。
    while(getColorAmbient(colorSensor) >= 15)
    {
        //等待, 无需操作。
        sleep(1);
    }
    //在液晶屏上显示“亮灯”图像。
    drawBmpfile(0, 127, "Light on");
    //停留5秒。
    sleep(5000);
}
```

本课程的程序解决方案可从此处下载:
<http://www.LEGOeducation.com>



下一步是什么? 线检测



自动大灯

学生学习单

为汽车设计一些功能，可帮助提高夜间驾驶安全。

学生学习单

联系

确保你能回答下列问题：

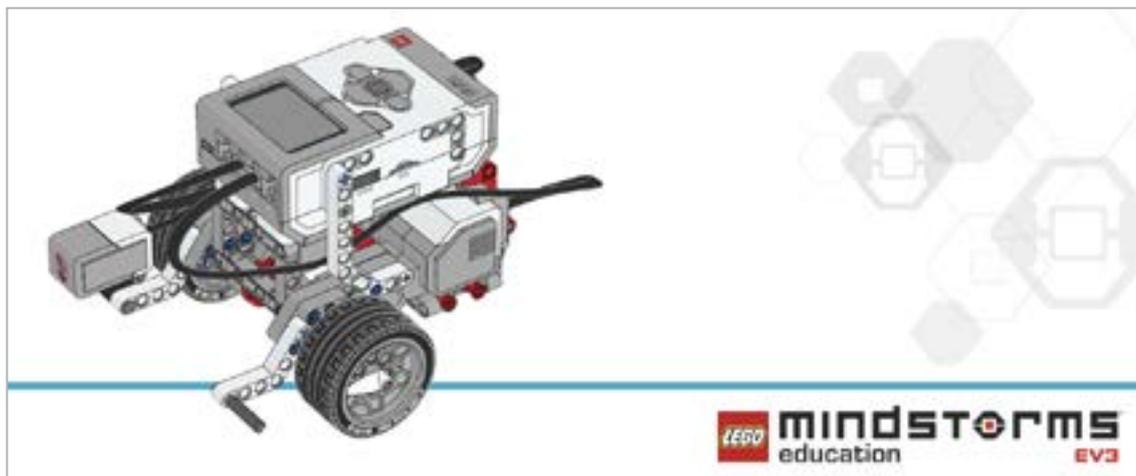
- 夜间驾驶存在哪些危险？
- 我们如何界定的环境光线的概念？
- 在驾驶员夜间驾驶过程中，如何能够帮助他们？

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



程序

编制一个程序，以模拟一辆汽车的自动大灯。

- 为EV3程序块显示屏寻找一张电灯泡图像，该程序块可引入你的程序。
- 使用颜色传感器来触发你的“电灯泡”，使其打开。
- 为了正确调整程序，可从端口视图中获取环境光线的读数。

考虑将这些模块应用于你的解决方案：

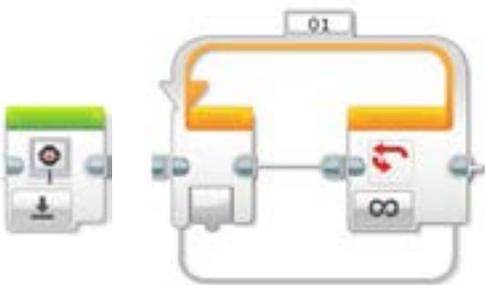


思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

反思

既然已成功将自动大灯开启，还需要进一步拓展你的程序，以实现当环境光线变亮时关闭自动大灯。为实现这一过程，你需要创建一个程序，其可以自我重复，这样你就无需一直重启它。

考虑将这些模块加入你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学部分

创建一个程序，可为你的汽车自动大灯提供更多控制功能，从而实现灯光的手动开启或关闭。许多现代车辆都具备此项功能，可允许驾驶员覆盖自动程序。

- 你可能需要通过平行编程或多任务的方式来探索程序构建的新途径。
- 你可以使用触动传感器来模拟手动开关。
- 你可能需要借助循环中断模块来覆盖自动控制。

考虑将这些模块加入你的解决方案：



思考你所了解的情况，然后进行记录。创意性地录制和展示你的想法、创作和发现。

分享

请思考以下问题：

你经历了哪些挑战？有没有什么惊喜？你如何能够改进自己的程序？

你的程序能够更简洁吗？你是否使用很多模块？是否存在一种更有效的程序构建方式？

你的程序如何能够应用到现实生活中？

拓展

探究用于本次课程的基于文本的编程解决方案，并对比这些采用不同编程语言的解决方案。



线检测

设计一些防止驾驶员打瞌睡的方法，以提高驾驶安全性，避免事故。

学习目标

学生将：

探究循环的概念

了解开关语句的概念，以及如何利用它进行“真”和“伪”运算

词汇

输入、输出、算法、伪代码、等待、颜色传感器、调试、环境光线、反射光线、循环、布尔逻辑、开关

年级

6-8

学科

工程、STEM (科学、技术、工程和数学)、编程

课程持续时间

45-90分钟

难度

中等

标准

《新一代科学教育标准》(NGSS)

MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. /MS-ETS1-4.

计算机科学教师协会 (CSTA) 标准

2-A-2-1 / 2-A-7-2 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-5-6 / 2-A-3-9 / 2-A-6-10 / 2-C-7-11 / 2-C-4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高®MINDSTORMS® 头脑风暴教育 EV3 机器人核心套装
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC 软件 (可选)

线检测

联系 (5分钟)

告知学生，他们将再次使用颜色传感器。学生可利用反射光线强度编写一个程序，实现轮式机器人沿着给定轨迹行进，这样会加深他们对传感器如何对光反应的理解。

告知他们，需要将利用颜色传感器让轮式机器人更具自主性，以模拟无人驾驶汽车如何对交通灯作出响应。

他们将创建一个程序，实现让自己的轮式机器人沿着给定的轨迹行驶。

围绕下列问题进行展开课堂讨论：

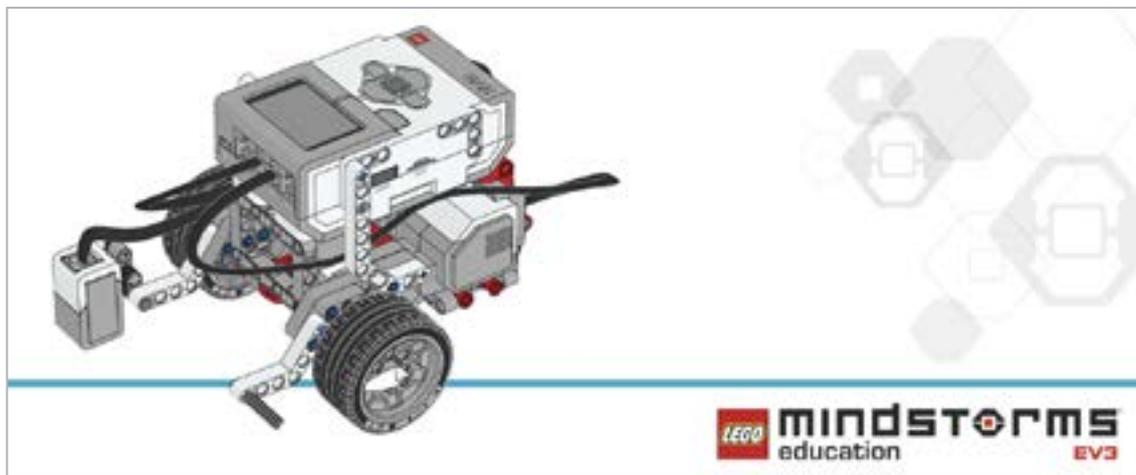
- 无人驾驶汽车可以对不同的交通灯信号做出反应吗？
- 如果驾驶员在开车时睡着了会怎么样？
- 我们如何能够检测到驾驶员正在打瞌睡？

允许学生选择最适合自己获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式，将自己的想法记录下来。

建构 (15到30分钟)

搭建

学生将会构建 Robot Educator 基本模型，然后添加颜色传感器，并将其面向地面。



在对机器人进行编程之前，先让学生对搭建的系统进行检查：

- 机器人的轮子能否自由旋转？
- 连接线是否正确地从颜色传感器连接到端口 3？

程序

让学生在EV3编程环境中新建一个项目。

学生通过对轮式机器人编程,使其沿着一条路径行进,并在一块红色积木前停下,借此他们可以探索颜色传感器的功能——识别乐高®积木颜色。

让学生模拟车辆在交通灯前的行为,使他们的轮式机器人可对一系列绿色和红色信号做出反应。将他们的程序放入一个循环中,以应对一条道路上多个“交通灯”的情况。

允许学生选择最适合自己获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

备注

可让学生参阅Robot Educator教程以获取更多帮助。

在EV3机器人软件中:

- Robot Educator > Basics > Straight Move
- Robot Educator > Basics > Curved Move
- Robot Educator > Hardware > Color Sensor – Color
- Robot Educator > Beyond Basics > Loop
- Robot Educator > Basics > Stop at Line
- Robot Educator > Beyond Basics > Switch

- 学生将会需要借助等待模块来完成该任务。需要指出,等待模块可被配置为被多种或仅一种颜色触发。在这种情况下,学生将会创建一个程序,实现当颜色传感器检测到红色时,停止电机。
- 向学生指出,为了使颜色传感器最有效地对他们所选择的颜色(红色和绿色)作出反应,他们需要确保不要选择其它任何颜色。
- 还需告知学生,他们需要探究开关语句如何运行,以及如何将它们纳入自己的程序。
- 告知学生,他们需要探究循环语句如何运行,以及如何将它们纳入自己的程序。

活动准备

使用 EV3 机器人核心套装中的横梁（乐高科技系列）模拟红灯和绿灯。将横梁放在工作台上，可让颜色传感器在机器人通过它们时检测到它们。



可能的解决方案

文件名称: CODING-04.EV3 (Tab:1)



红光检测

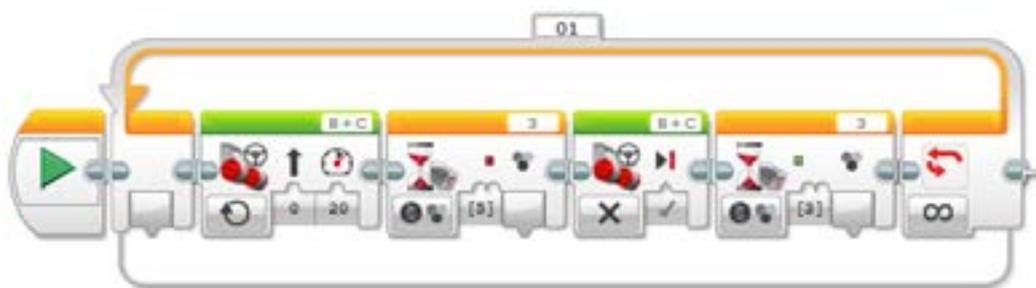
- 1.启动程序。
- 2.启动电机 B 和 C (向前移动)。
- 3.等待颜色传感器检测到红色。
- 4.停止电机。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

可能的解决方案

文件名称: CODING-04.EV3 (Tab:2)



在循环中实现红色和绿色检测

- 2.启动电机 B 和 C (向前移动)。
- 3.等待颜色传感器检测到红色。
- 4.停止电机。
- 5.等待颜色传感器检测到绿色。
- 6.一直重复步骤 2 至 5。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

反思 (35分钟)

学生应使用颜色传感器的同一功能来辨识机器人是否越线。如果你采用白色或浅色表面,则需要选用粗(约2厘米或3/4英寸)的深色线;若采用深色表面,则需要选用白色线。

让学生模拟机器人越线时向驾驶员发出的警报信号。新式汽车一般都具备这项功能。

可能的解决方案

文件名称: CODING-04.EV3 (Tab:3)



在循环中实现线检测

- 1.启动程序。
- 2.启动电机 B 和 C (以一定弧度靠近该线)。
- 3.等待颜色传感器检测到黑色,然后启动任务1和2。

任务1

- 4.播放声音“Horn 1”。

任务2

- 5.启动电机 B 和 C (以一定弧度离开该线)。
- 6.等待颜色传感器检测到白色。
- 7.一直重复步骤2至6。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分层教学部分

学生将创造一辆可沿线行进的自动化无人驾驶汽车。

让学生探索一辆无人驾驶车辆如何自行沿公路或路径行进。

需要向学生介绍切换模块，该模块将会在循环内运行。

向他们讲解，切换模块可用于程序的自主运行，通过它就可以实现轮式机器人的自主运动。

还需要说明，切换模块可用于控制程序的流程，而使用触动传感器的默认切换模块本身就是布尔逻辑运用的一个经典实例。

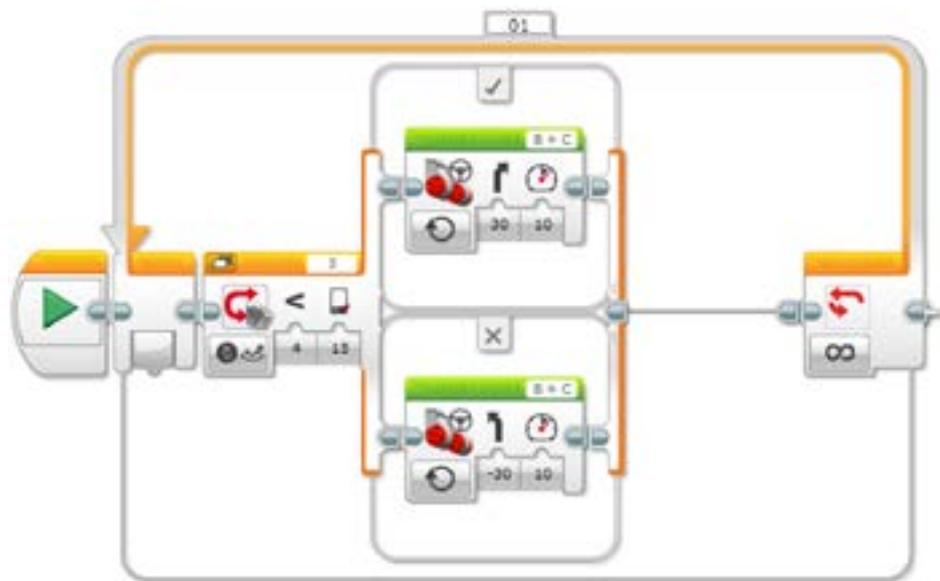
展示如何将切换模块转换为颜色传感器，并解释如何利用触发点产生真/伪陈述（观察切换模块，并关注触发点上方的程序流程如何进行一个动作，或者下方的流程如何进行另一个动作）

需要指出，为了创建巡线程序，他们需要让轮式机器人沿轨迹做摇摆运动。换言之，轮式机器人会根据自身是否越线（即触发器），一直进行向左或向右转向调整。可从网上找到一个合适的视频，作为这个现象的实例向学生展示需要指出，在这次挑战中移动转向模块需要被设置为“On”，而不是“On for”（如秒数、度数或圈数）。

轮式机器人可以巡线运动后，能否把它改进的更像汽车（即沿直线运动而不是摆动运动）？

可能的解决方案

文件名称：CODING-04.EV3 (Tab:4)



在循环中实现巡线功能

- 1.启动程序。
- 2.启动电机 B 和 C（以一定弧度靠近该线）。
- 3.等待颜色传感器检测到黑色。
- 4.启动电机 B 和 C（以一定弧度离开该线）。
- 5.等待颜色传感器检测到白色。
- 6.一直重复步骤 2 至 5。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

备注

学生将再次使用颜色传感器，但这次他们将需要对其进行编程，使其能够对反射光线强度作出响应。为了判断输入等待模块的数值，他们需要从端口视图中获取环境光线的读数。

在颜色非常浅的（或白色）表面上，最适合使用黑色或蓝色胶带。

您需要花一些时间讲解开关语句的概念，以及它如何是布尔逻辑的一个范例。

基于上述活动内容，可以添加第二个颜色传感器，并与巡线和交通灯程序组合，来拓展模拟自动客运服务系统，就像机场中的列车系统。

分享

允许学生选择最适合自己的工具，以获取和分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

本次课程涵盖了很多新概念，以及EV3软件的多个新模块。可利用这段时间回顾课程内容，确保学生能够完全理解。

要求一到两个小组展示他们的程序。

要求学生就他们的期望与程序实际运行的结果进行分享。

并询问他们程序运行的结果是否为他们带来了一些惊喜。

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案，这样他们可以比较不同的编程语言。

重要提示：

下文提供了一个可能的解决方案，其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权，故就用户体验和技术方面的质量问题，不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>，获取所需的全部安装信息。当您使用其他编程语言完成活动时，我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-04_1.C

```
#pragma config(Sensor, S3, colorSensor, sensorEV3_Color, modeEV3Color_Color)

#pragma config(Motor, motorB, leftMotor, tmotorEV3_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorC, rightMotor, tmotorEV3_Large, PIDControl, driveRight, encoder)
```

```
/*
```

创建一个程序,可驱动机器人持续向前移动,直至颜色传感器发现红色。

然后机器人停止。

```
*/
```

```
task main()
{
//将电机速度设定为20% (向前移动)。
    setMotorSpeed(motorB, 20);
    setMotorSpeed(motorC, 20);

//持续循环直至颜色传感器发现红色。
    while(getColorName(colorSensor) == colorRed)
    {
//持续行进直至颜色传感器发现红色。
        sleep(10);
    }

//将电机速度设置为0% (停止)。
    setMotorSpeed(motorB, 0);
    setMotorSpeed(motorC, 0);
}
```

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>



下一步是什么?

检测物体



线检测

学生学习单

设计一些防止驾驶员打瞌睡的方法，以提高驾驶安全性，避免事故。

学生学习单

联系

确保你能回答下列问题：

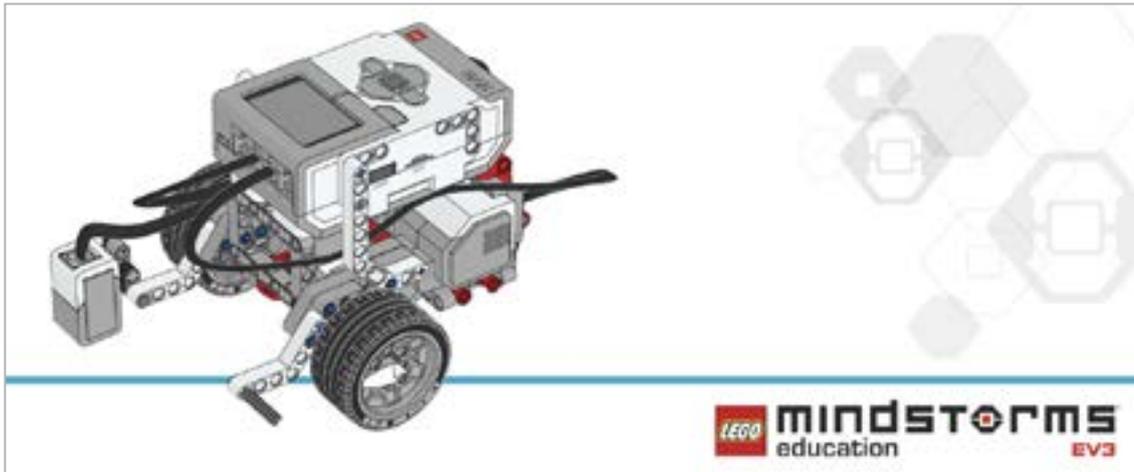
- 无人驾驶汽车可以对不同的交通灯信号做出反应吗？
- 如果驾驶员在开车时睡着了会怎么样？
- 我们如何能够检测到驾驶员正在打瞌睡？

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



程序

无人驾驶汽车需要自动对交通灯进行识别并做出反应。

创建一个程序，使机器人在红灯时停止。

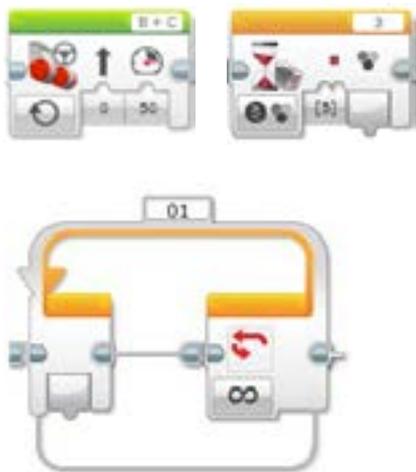
改进程序，使机器人在距交通灯适当距离时停止。

确保机器人仅对红色做出响应。

成功后，对机器人再次编程，实现当灯光由红转绿时使其继续前进。

利用颜色传感器和切换模块进行决策（布尔逻辑）。这两个模块将允许机器人根据检测到的颜色进行选择。

考虑将这些模块应用于你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

反思

模拟驾驶员在开车时睡着了会发生什么，当汽车越线时你的机器人会发出警报声音。新式汽车一般都具备这项功能。

对机器人进行编程，使其实现这项功能。

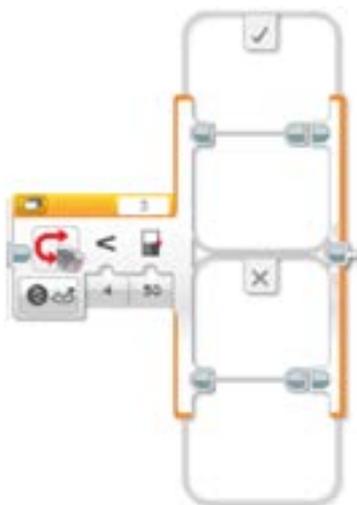
思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学部分

对机器人进行编程，使其沿着给定路线自动驾驶。你需要创建一个程序，可以识别深色线（或白线）并对其做出反应。你将会创建一个巡线程序，机器人可实现沿线运动，而不偏离。

你需要不断调试程序，以实现让机器人尽可能流畅地沿线行进。

考虑将这些模块加入你的解决方案：



对你的学习内容反思和记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分享

思考你所了解的情况，然后进行记录。创意性地录制和展示你的想法、创作和发现。

请思考以下问题：

你经历了哪些挑战？有没有什么惊喜？你如何能够改进自己的程序？

你的程序能够更简洁吗？你是否使用很多模块？

是否存在一种更有效的程序构建方式？

你的程序如何能够应用到现实生活中？

拓展

探究用于本次课程的基于文本的编程解决方案，并对比这些采用不同编程语言的解决方案。



目标探测

设计一些方法可避免车辆与路间物体相撞的事故。

学习目标

学生将:

加深对循环的理解

词汇

输入、输出、算法、伪代码、调试

年级

6-8

学科

工程、STEM (科学、技术、工程和数学)、编程

课程持续时间

45-90分钟

难度

中等

标准

《新一代科学教育标准》(NGSS)

MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. /MS-ETS1-4

计算机科学教师协会 (CSTA) 标准

2-A-2-1 / 2-A-7-2 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-5-6
/ 2-A-3-9 / 2-A-6-10 / 2-C-7-11 / 2-C-4-12 / 2-D-5-16 /
2-I-1-20

所需材料

乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人核心套装
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC 软件 (可选)

目标探测

联系 (5分钟)

围绕下列问题展开课堂讨论:

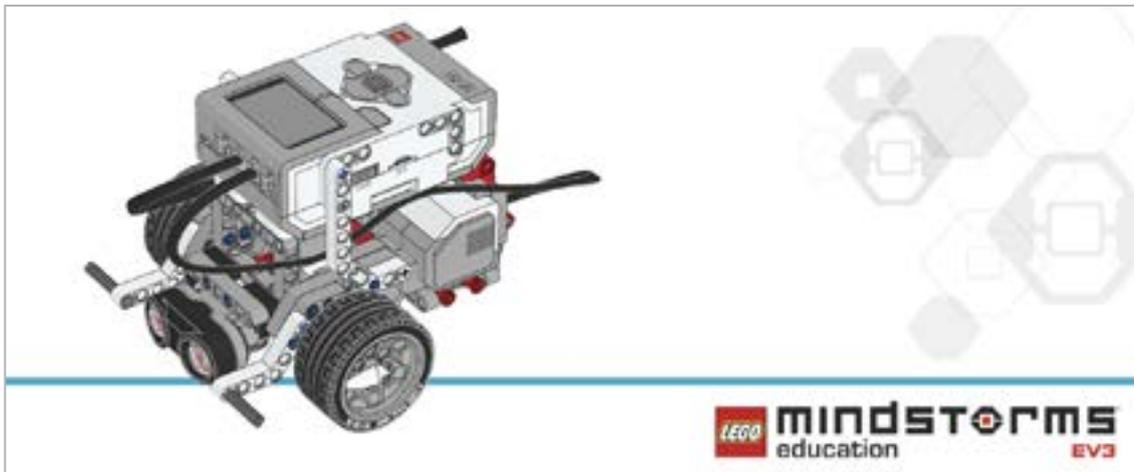
- 在哪种驾驶状况下, 小汽车会撞上障碍物?
- 为了避免与障碍物相撞, 哪些关键因素需要我们的关注?
- 是什么原因导致了高密度区域的塞车?

允许学生选择最适合自己获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式, 将自己的想法记录下来。

建构 (15到30分钟)

搭建

学生将会构建 Robot Educator 基本模型, 然后添加超声波传感器, 并使其面向前方。



在对机器人进行编程之前, 先让学生对搭建的系统进行检查:

- 连接线是否正确地从电机连接到端口 B 和 C?
- 机器人的轮子是否正确安装?
- 机器人的轮子能否自由旋转?
- 连接线是否正确地从超声波传感器连接到端口 4?

程序

介绍超声波传感器的简单用法。

演示等待模块以及如何将其用于超声波传感器。

让学生对轮式机器人进行编程, 能够实现机器人在向前 (或向后) 移动过程中检测任何可能出现的障碍物。

学生将会创建一个程序, 实现让机器人根据超声波传感器测量的距离停止在某个特定点上。

允许学生选择最适合自己获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称: CODING-05.EV3 (Tab:1)



探测物体并停止

- 1.启动程序。
- 2.开启两个电机, 并将速度设为 50。
- 3.等待超声波传感器探测到自己与障碍物间的距离小于 20 厘米。
- 4.关闭两个电机。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

备注

可让学生参阅 Robot Educator 教程以获取更多帮助。

在EV3机器人软件中:

Robot Educator > Basics > Straight Move

Robot Educator > Basics > Stop at Object

活动准备

在分派下一个任务之前, 请确保您已标定机器人的行进路径, 并为任务开展预留了足够空间。最好能让学生在宽大的工作台或地板上开展任务。

让学生尝试使用超声波传感器检测不同的物体。其中一个物体可以由乐高® 积木拼砌的长方体。



反思 (35分钟)

在驾车期间,当驾驶员看到一个物体时,他们会先减缓车速,然后停车。

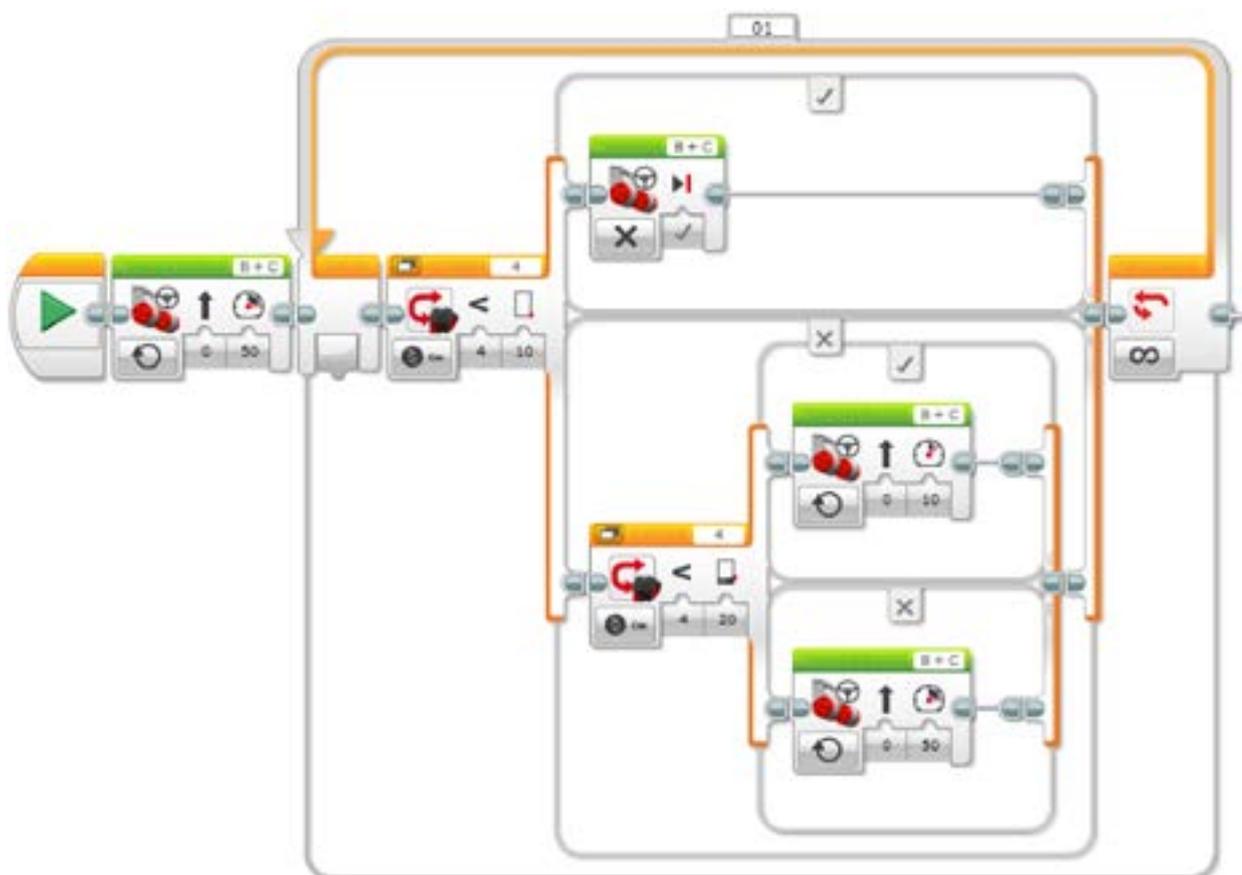
让学生对自己的机器人进行编程,使其具有相同的行为。

如果超声波传感器:

- 探测到自己距离物体不足10厘米,让机器人停止前进
- 探测到机器人距离物体10-20厘米,让机器人减速
- 未探测到任何物体,继续以全速前进

可能的解决方案

文件名称: CODING-05.EV3 (Tab:2)



探测物体并做出反应

- 1.启动程序。
- 2.开启两个电机,并将速度设为50。
- 3.如果超声波传感器探测到自己距离障碍物不足10厘米,关闭两个电机。
否则
- 4.如果超声波传感器探测到自己距离障碍物小于20厘米,将两个电机的速度设为10。
否则
- 5.开启两个电机,并将速度设为50。
- 6.一直重复步骤3至7。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分层教学部分

聚集所有小组。

指导学生将他们的机器人放置在一条垂直线上,并在机器人之间留有不同间距(就像交通堵塞一样)。让他们同时启动自己的程序,然后观察会发生什么。

要求学生改进程序,能够实现所有机器人以相同速度持续前进,并在机器人之间留有相等的间距(就像公路上正常行进的车流)。

分享

允许学生选择最适合自己的工具,以获取和分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

讨论编程效率的概念。

整个小组内有多少种不同的解决方案?针对给定的问题,对比多种可能的解决方案。

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案,这样他们可以比较不同的编程语言。

重要提示:

下文提供了一个可能的解决方案,其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权,故就用户体验和技术方面的质量问题,不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>, 获取所需的全部安装信息。当您使用其他编程语言完成活动时,我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-01_1.C

```
#pragma config(Sensor, S4,      sonarSensor,      sensorEV3 _ Ultrasonic)
#pragma config(Motor,  motorB, leftMotor,      tmotorEV3 _ Large,
PIDControl, driveLeft, encoder)
#pragma config(Motor,  motorC, rightMotor,      tmotorEV3 _ Large,
PIDControl, driveRight, encoder)
```

```
/*
```

创建一个程序,可驱动机器人持续向前移动,直至超声波传感器发现物体。
然后机器人停止。

```
*/
```

```
task main()
{
    //将电机速度设定为50% (向前移动)。
    setMotorSpeed(motorB, 50);
    setMotorSpeed(motorC, 50);
    while(getUSDistance(sonarSensor) < 20)
    {
        //持续前进,直到超声波传感器的探测值小于20厘米。
    }

    //一旦超声波传感器的探测值小于20厘米。
    //将电机速度设定为0% (停止)。
    setMotorSpeed(motorB, 0);
    setMotorSpeed(motorC, 0);
}
```

本课程的程序解决方案可从此处下载:
<http://www.legoeducation.com>



下一步是什么?

解锁一辆汽车



目标探测

学生学习单

设计一些方法可避免车辆与路间物体相撞的事故。

学生学习单

联系

确保你能回答下列问题：

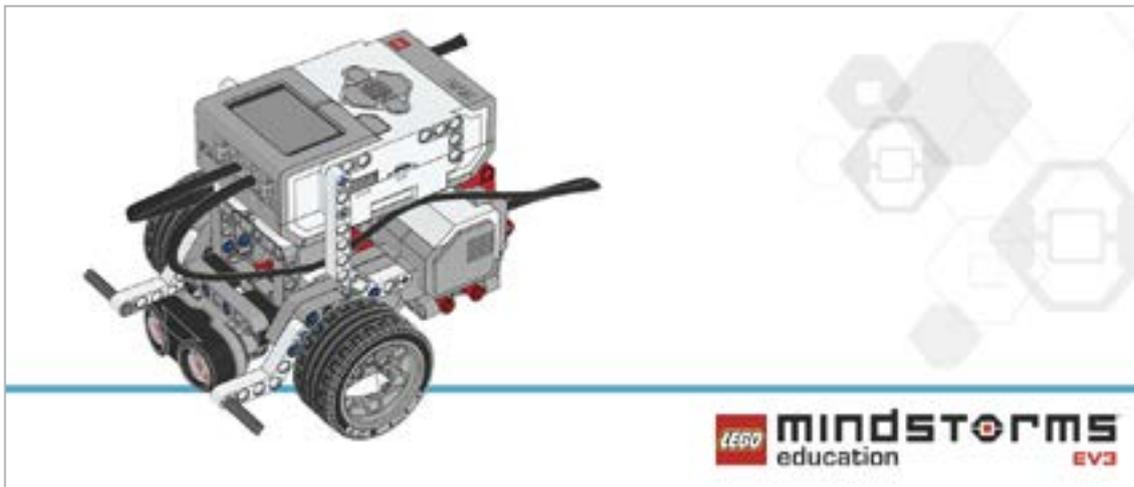
- 在哪种驾驶状况下，小汽车会撞上障碍物？
- 为了避免与障碍物相撞，哪些关键因素需要我们的关注？
- 是什么原因导致了高密度区域的塞车？

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



程序

对机器人进行编程，使其在向前（或向后）移动过程中能够检测任何可能出现的障碍物。当探测到自己距离障碍物小于20厘米时，让机器人停止前进。

考虑将这些模块应用于你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

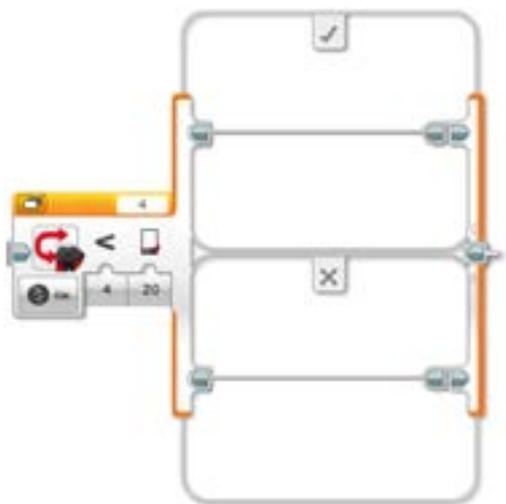
反思

在驾车期间,当驾驶员看到一个物体时,他们会先减缓车速,然后停车。
对轮式机器人编程,实现相同的过程。

如果超声波传感器:

- 探测到自己距离物体不足10厘米,让机器人停止前进
- 探测到机器人距离物体10-20厘米,让机器人减速
- 未探测到任何物体,继续以全速前进

考虑将这些模块加入你的解决方案:



思考你所了解的情况,然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学部分

将所有小组召集到一起。

将所有机器人放置在一条垂直线上,并在机器人之间留有不同间距(就像交通堵塞一样)。

让所有人同时启动他们的程序,然后观察会发生什么。

改进程序,使其实现让所有机器人以相同速度持续前进,并在机器人之间留有相等的间距(就像公路上正常行进的车流)。

思考你所了解的情况,然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分享

请思考以下问题:

“编程效率”的含义是什么?

整个班级内有多少种不同的解决方案? 针对给定的问题, 对比多种可能的解决方案。

拓展

探究用于本次课程的基于文本的编程解决方案, 并对比这些采用不同编程语言的解决方案。



解锁一辆汽车

设计一种方式，可利用密码来防止汽车被盗。

学习目标

学生将：

了解简单的布尔逻辑（如“与”、“或”和“非”），以及其在电路和编程中的应用
使用几种输入的组合。

词汇

输入、输出、伪代码、布尔逻辑、代数逻辑、开关、真、伪、循环

年级

6-8

学科

工程、STEM（科学、技术、工程和数学）、编程

课程持续时间

45-90分钟

难度

高级

标准

《新一代科学教育标准》(NGSS)
MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. /MS-ETS1-4

计算机科学教师协会 (CSTA) 标准
2-A-2-1 / 2-A-7-2 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-5-6 / 2-A-5-7 / 2-A-3-9 / 2-A-6-10 / 2-C-7-11 / 2-C-4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人核心套装
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC 软件（可选）

解锁一辆汽车

联系 (5分钟)

你知道现在可以不用钥匙启动一辆车了吗?“车钥匙”或遥控器可遥控启动汽车发动机。驾驶员只需按下按钮和制动器(或离合器)的组合,即可启动汽车。在本次课程中,学生将学习如何利用传感器和代数逻辑的组合实现上述过程,启动他们的轮式机器人。

围绕下列问题进行展开课堂讨论:

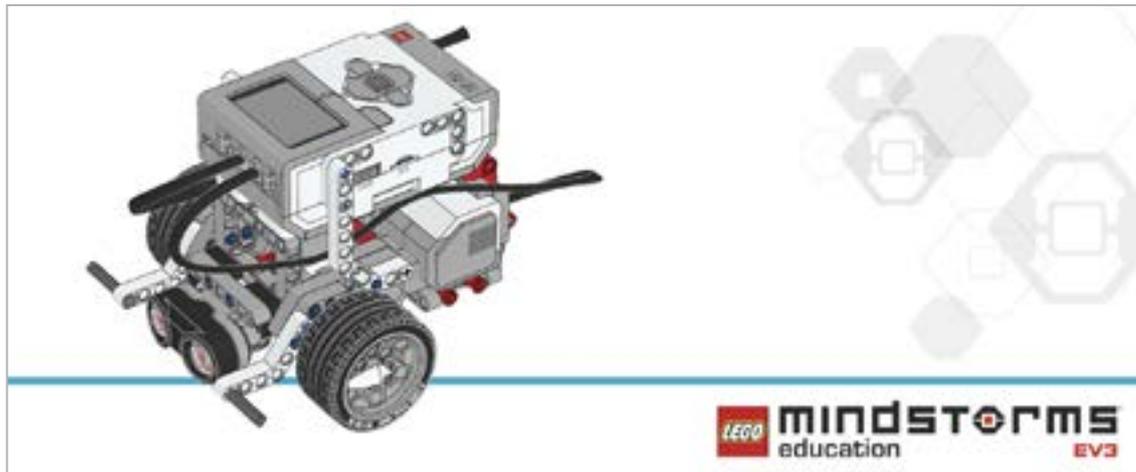
- 电子钥匙的工作原理是什么?
- 我们如何确保每把钥匙只能打开一辆汽车?

允许学生选择最适合自己获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式,将自己的想法记录下来。

建构 (15到30分钟)

搭建

学生将会构建 Robot Educator 基本模型,然后添加超声波传感器。



在对机器人进行编程之前,先让学生对搭建的系统进行检查。

- 连接线是否正确地从电机连接到端口 B 和 C?
- 机器人的轮子是否正确安装?
- 机器人的轮子能否自由旋转?
- 连接线是否正确地从超声波传感器连接到端口 4?

程序

让学生在EV3编程环境中新建一个项目。

让他们对自己的机器人进行编程, 可实现对驾驶员进行存在检测, 然后由驾驶员按下点火按钮。在本示例中:

- 当驾驶员在车内时, 超声波传感器用于模拟驾驶员衣兜里的车钥匙。这常被称为“进入泡泡”。
- 程序块按钮用于开启汽车。
- 程序块显示屏用作显示信息的仪表盘。

学生实现让他们的轮式机器人显示信息后, 再要求他们将自己的程序放入循环中, 以实现重复显示信息。

推荐的程序将两个输入放入了一个序列。

允许学生选择最适合自己获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称: CODING-06.EV3 (Tab:1)



驾驶员存在检测和汽车点火 (依次进行)

- 1.启动程序。
- 2.等待超声波传感器探测到自己距离障碍物小于5厘米。
- 3.在程序块显示屏上显示一个欢迎信息。
- 4.等待向上按钮受到按压。
- 5.在程序块显示屏上显示一个点火信息。
- 6.等待3秒, 然后结束程序。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

备注

可让学生参阅 Robot Educator 教程以获取更多帮助。

在EV3机器人软件中:

- Robot Educator > Beyond Basics > Logic
- Robot Educator > Basics > Stop at Object
- Robot Educator > Beyond Basics > Loop
- Robot Educator > Beyond Basics > Switch
- Robot Educator > Beyond Basics > Data Wires
- Robot Educator > Beyond Basics > Sensor Blocks

反思 (35分钟)

学生将必须同时使用两个输入。

让学生创建一个程序, 实现仅当正确的启动组合被激活时才可授予机器人的控制权。

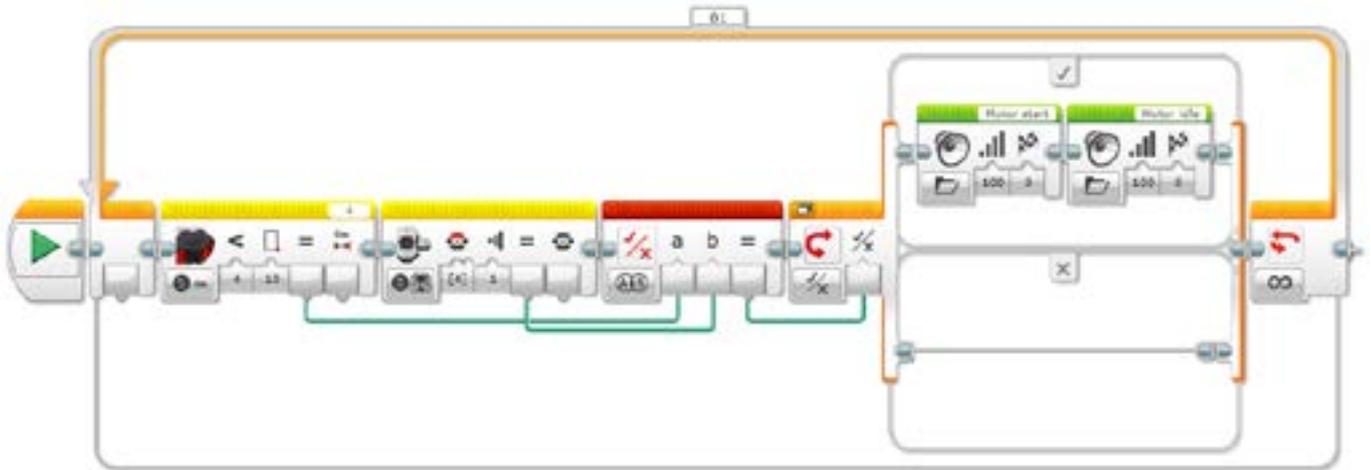
让学生反思他们的第一个程序。向他们提问如何可以创建一个程序来激活机器人。指导他们使用逻辑运算模块。并向他们解释, 逻辑运算模块的表现形式与维恩图类似。指出维恩图的众多条件, 以及其与EV3软件之间的关联性。讲解该模块时, 您可能需要解释下图, 并说明相应结果是如何产生的。

模式	使用的输入	结果
 AND	A, B	如果A和B (或两者) 均为“真”, 则结果为“真”; 如果A和B均为“伪”, 则结果为“伪”
 OR	A, B	如果A和B (或两者) 均为“真”, 则结果为“真”; 如果A和B均为“伪”, 则结果为“伪”
 XOR	A, B	如果A和B其中之一为“真”, 则结果为“真”; 如果A和B均为“伪”, 则结果为“伪”
 NOT	A	如果A为“伪”, 则结果为“真”, 如果A为“真”, 则结果为“伪”

EV3软件的逻辑运算模块可根据模块中设定的标准产生真/伪输出。例如, 只有当A和B均为“真”时, A和B才可以输出“真”的结果。

可能的解决方案

文件名称: CODING-06.EV3 (Tab:2)



通过同时满足两个条件来启动一辆汽车

- 1.启动程序。
- 2.当超声波传感器探测到自己距离障碍物小于4厘米时, 发送一个“真”结果。
- 3.向上按钮受到按压时, 程序块按钮发送出“真”结果。
- 4.在向逻辑开关发送一个“真”消息之前, 逻辑运算模块需要等待收到两个“真”消息。
- 5.逻辑开关对比从逻辑运算模块收到的消息。
如果为“真”, 则EV3程序块会播放两个声音。
如果为“伪”, 没有任何反应。
- 6.一直重复步骤2至5。

本课程的程序解决方案可从此处下载:
<http://www.LEGOeducation.com>

分层教学部分

让学生设计一个程序, 模拟一个需要满足三个条件的密码。确保学生了解, 为使汽车启动, 所有三个条件必须为真。例如:

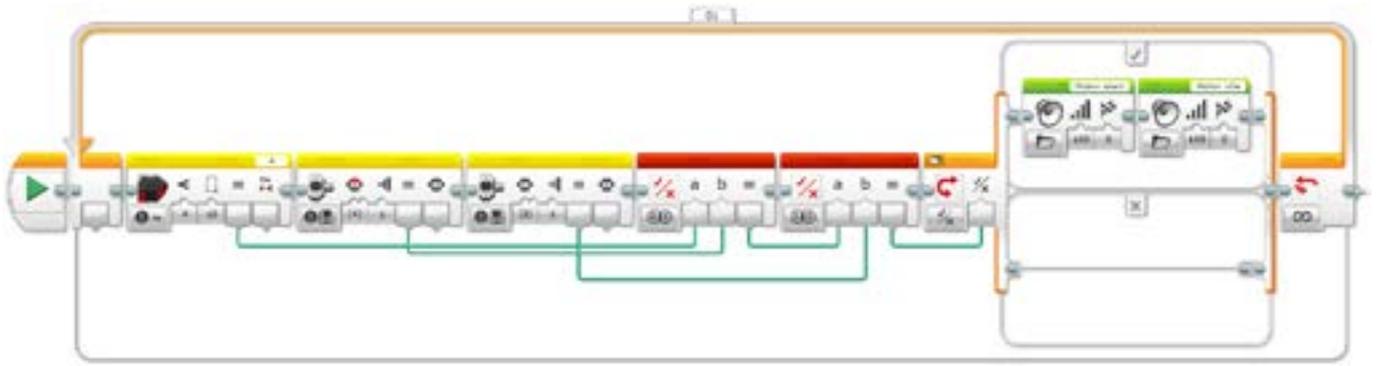
- 车钥匙或遥控器本身需要在车内。
- 制动器或离合器需要被按下。
- 点火按钮必须被按下。

学生可以连接触动传感器来模拟离合器或制动器。

让学生设计一个程序, 用于实现如果上述三个启动“汽车”的条件不能被全部满足, 则显示一条错误信息。

可能的解决方案

文件名称: CODING-06.EV3 (Tab:3)



通过同时满足三个条件来启动一辆汽车

- 1.启动程序。
 - 2.当超声波传感器探测到自己距离障碍物小于4厘米时,发送一个“真”结果。
 - 3.当顶部按钮受到按压时,程序块按钮发送一个“真”结果。
 - 4.当返回按钮受到按压时,程序块按钮发送一个“真”结果。
 - 5.在向逻辑开关发送一个“真”消息之前,逻辑运算模块需要等待收到两个“真”消息(来自超声波传感器和触动传感器的信息)。
 - 6.在向逻辑开关发送一个“真”消息之前,逻辑运算模块需要等待收到两个“真”消息(来自逻辑运算模块和程序块按钮)。
 - 7.逻辑开关对比从逻辑运算模块收到的消息。
如果为“真”,则EV3程序块会播放两个声音。
如果为“伪”,没有任何反应。
- 6.一直重复步骤2至7。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分享

允许学生选择最适合自己的工具,以获取及分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

学生如何发现并使用数据线? 要求他们分享自己的想法。指出数据线可以随意移动,也可以被整理,它对保证程序的干净整洁十分重要。

对等待模块和逻辑运算进行讨论。向学生讲解,等待模块通常一次只与一个传感器搭配使用,而逻辑运算模块则允许用户同时使用多个传感器。

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案，这样他们可以比较不同的编程语言。

重要提示：

下文提供了一个可能的解决方案，其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权，故就用户体验和技术方面的质量问题，不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>，获取所需的全部安装信息。当您使用其他编程语言完成活动时，我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-06_1.C

```
#pragma config(StandardModel, "EV3_REMBOT")
/*
创建一个程序,用以展示遥控点火如何实现。超声波传感器欢迎驾驶员。触动传感器激发点火。
*/

task touchTask()
{
    //等待触动传感器受到按压。
    while(getTouchValue(touchSensor) == 0)
    {
        //不操作。
        sleep(10);
    }
    //在液晶屏上显示文字。
    eraseDisplay();
    displayCenteredBigTextLine(4, "Ignition");
    //显示文字3秒钟。
    sleep(3000);
}
```

```
task main()
{
    //启动第二个任务, 开始监控触动传感器 。
    startTask(touchTask);
    //等待声纳传感器探测到自己距离障碍物小于或等于5厘米。
    while(getUSDistance(sonarSensor) >= 5)
    {
        //不操作。
        sleep(10);
    }
    //在液晶屏上显示文字。
    eraseDisplay();
    displayCenteredBigTextLine(4, "Welcome");
    //显示文字3秒钟。
    sleep(3000);
}
```

本课程的程序解决方案可从此处下载:
<http://www.LEGOeducation.com>



下一步是什么?

巡航控制



解锁一辆汽车

学生学习单

设计一种方式，可利用密码来防止汽车被盗。

学生学习单

联系

确保你能回答下列问题：

电子钥匙的工作原理是什么？

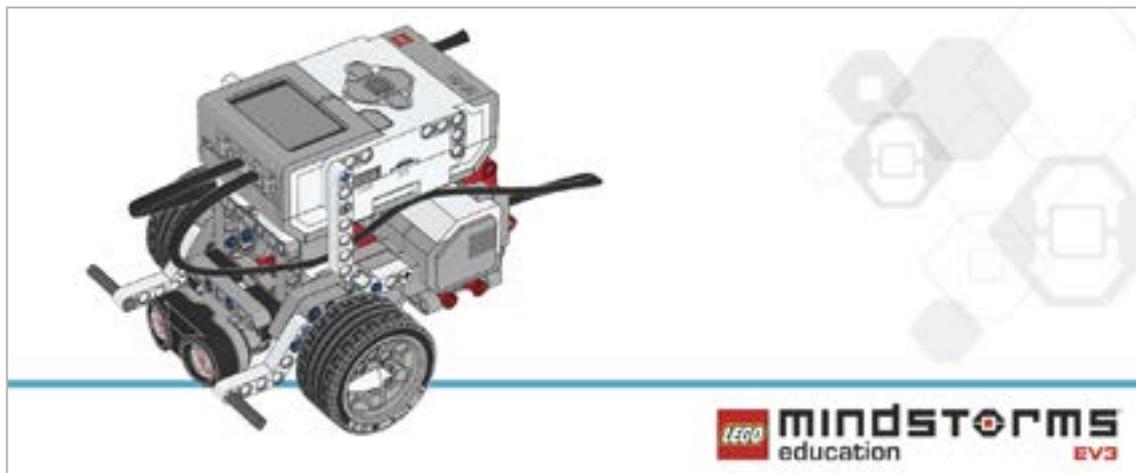
我们如何确保每把钥匙只能打开一辆汽车？

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



程序

为机器人创建一个无钥匙进入系统。

当一系列传感器组合被激活时，你的汽车驱动程序就会被执行。

对机器人进行编程，实现：当超声波传感器检测到物体时，显示“欢迎”字样；接着当触动传感器受到按压时，显示“点火”字样。

考虑将这些模块应用于你的解决方案：



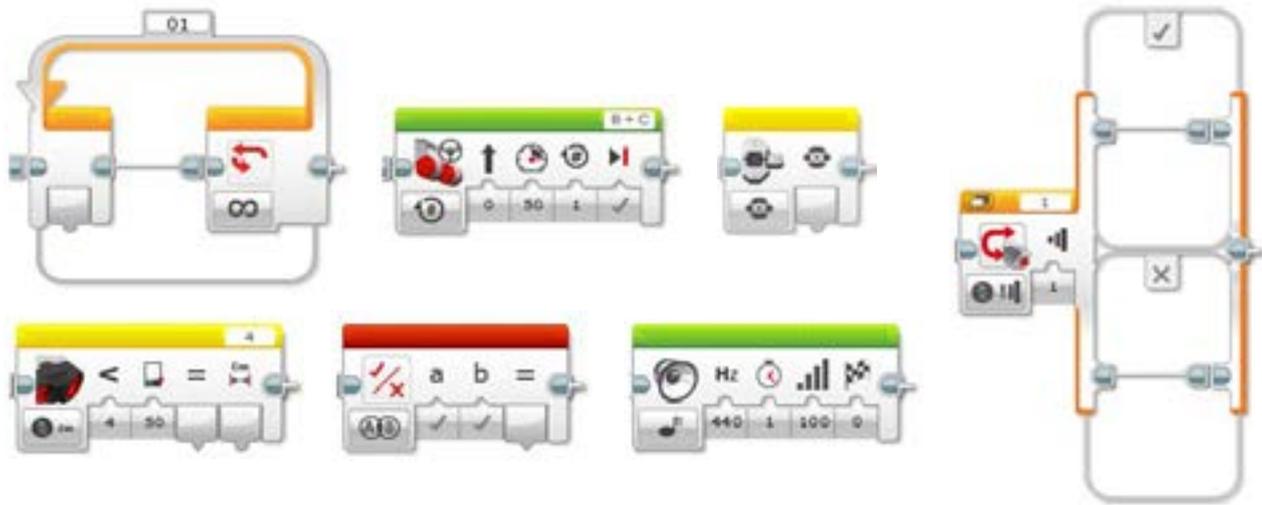
对你的学习内容反思和记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

反思

你将需要使用多种不同的输入和逻辑运算，以确保两个输入能够一起向另一个模块提供信息。思考无钥匙汽车的启动原理。对于该任务：

- 触动传感器用来点火。
- 超声波传感器用来检测车内的钥匙。
- 为使机器人启动，两个传感器需要被正确激活。
- 利用机器人的声音输出来表示它已经启动。

考虑将这些模块加入你的解决方案：



针对该任务，你将会使用传感器模块（黄色）来为逻辑运算模块创建逻辑。每个传感器模块都将被用于产生“真/伪”输出。

对你的学习内容反思和记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学部分

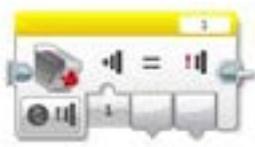
现在需要对机器人进行编程，实现：只有三个不同传感器的条件均被满足时，机器人才能“启动”。

对于该任务：

- 触动传感器用来点火。
- 超声波传感器用来检测车内的钥匙。
- 程序块按钮被用作制动器/离合器。

每个逻辑运算模块可以接收两个输入。如果需要三个输入,我们该怎么办?考虑使用两个逻辑运算模块来实现这一目标。两个传感器需要接入第一个逻辑运算模块。该模块的输出与第三个输入(传感器)一起接入下一个逻辑运算模块。所得结果会传递给切换模块。

考虑将这些模块加入你的解决方案:



思考你所了解的情况, 然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分享

请思考以下问题:

你对使用多个数据线的感受如何?

你的程序能够更简洁吗? 你是否使用很多模块?

是否存在一种更有效的程序构建方式?

你的程序如何能够应用到现实生活中?

通过可视化编程比较基于文本的编程, 哪个更容易学习?

如果还没有尝试两种编程方法, 请尝试另一种方法, 看哪种更高效。

拓展

探究用于本次课程的基于文本的编程解决方案, 并对比这些采用不同编程语言的解决方案。



巡航控制

设计一个巡航控制程序来协助驾驶员，减轻他们的驾驶压力。

学习目标

学生将：

使用变量模块来存储信息
编写多层程序
创建功能模块（我的模块）

词汇

输入、输出、伪代码、变量、常量、循环、等待、电机、触动传感器

年级

6-8

学科

工程、STEM（科学、技术、工程和数学）、编程

课程持续时间

45-90分钟

难度

高级

标准

《新一代科学教育标准》(NGSS)
MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. /MS-ETS1-4

计算机科学教师协会 (CSTA) 标准
2-A-2-1 / 2-A-7-2 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-5-6 / 2-A-5-7 / 2-A-3-9 / 2-A-6-10 / 2-C-7-11 / 2-C-4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人核心套装
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC软件（可选）

巡航控制

联系 (5分钟)

在本次课程中, 您将向学生介绍变量。学生将利用此信息为他们的轮式机器人添加巡航控制功能。按压触动传感器可以增加机器人的行进速度。

围绕下列问题进行展开课堂讨论:

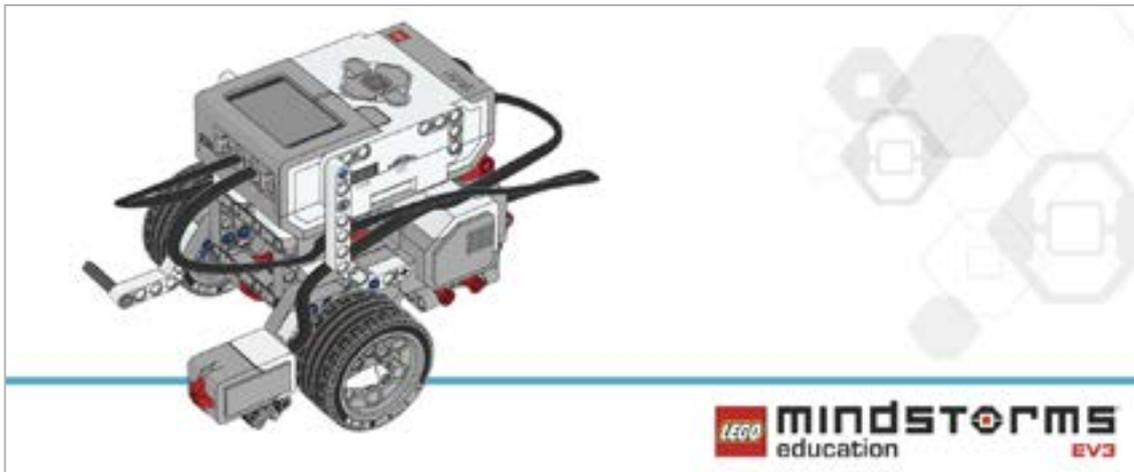
- 驾驶过程中哪些因素会使驾驶员感到压力?
- 在长时间驾驶过程中, 我们如何能帮助提高驾驶员的安全性?

向学生提问, 他们认为方向盘上的加号和减号有什么作用。讨论这两个符号如何调整汽车的速度。允许学生选择最适合自己获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式, 将自己的想法记录下来。

建构 (15到30分钟)

搭建

学生将会构建 Robot Educator 基本模型, 然后添加两个触动传感器。



告知学生, 今天的任务是利用两个触动传感器来控制并保持他们机器人的行进速度。

在对机器人进行编程之前, 先让学生对搭建的系统进行检查:

- 连接线是否正确地从电机连接到端口 B 和 C?
- 机器人的轮子是否正确安装?
- 机器人的轮子能否自由旋转?

程序

让学生在EV3编程环境中新建一个项目。

加深学生对变量模块的理解。向学生讲解，变量是一种编程模块，它可以存储数据（文本、逻辑、数字或阵列），在程序运行过程中这些数据可随时被重新赋值。

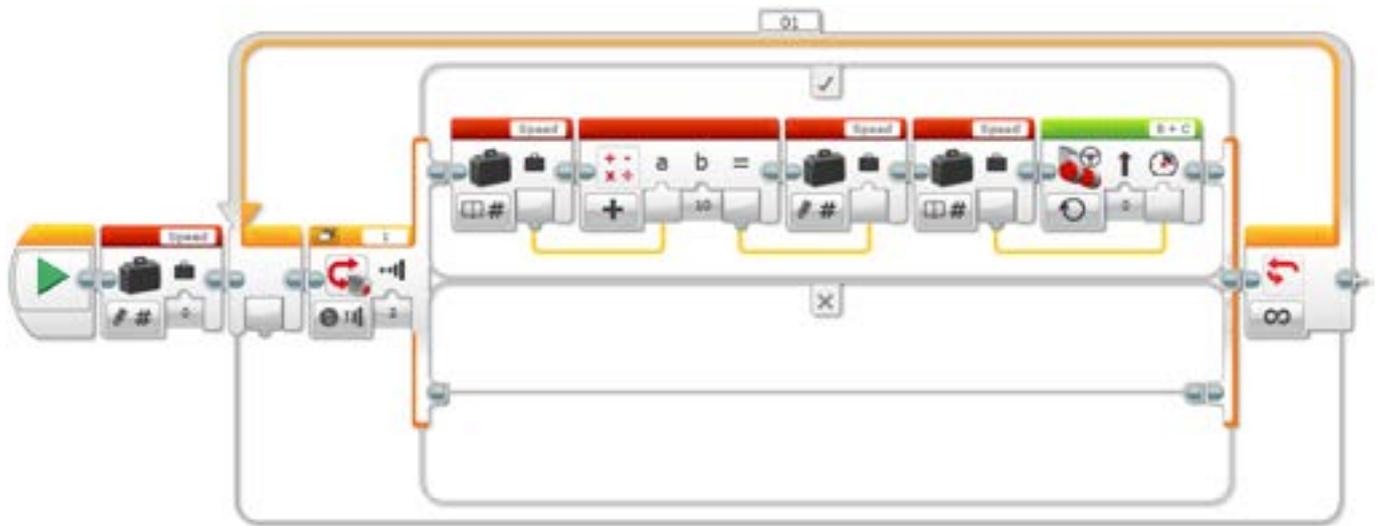
并介绍，使用数学、文本或阵列运算模块时，上述模块会被读取，然后被写入。

学生需要对自己的机器人进行编程，以实现机器人在移动时，可通过按压触动传感器来为它加速。控制触动传感器（等待）的程序块应放在一个循环中。

允许学生选择适合自己获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称: CODING-07.EV3 (Tab:1)



通过变量增加速度

- 1.启动程序。
- 2.创建一个名为“速度”的变量并赋予其0值。
- 3.如果触动传感器受到按压:
 - a.读取“速度”变量
 - b.使变量读取值增加10
 - c.将结果写入“速度”变量
 - d.读取“速度”变量
 - e.以设定的速度值启动电机B和C, 该速度值即为“速度”变量中存储的数值
- 否则
(不操作)
- 4.一直重复步骤3a至3e。

本课程的程序解决方案可从此处下载:
<http://www.LEGOeducation.com>

备注

可让学生参阅 Robot Educator 教程以获取更多帮助。

在EV3机器人软件中:

- Robot Educator > Beyond Basics > Variables
- Robot Educator > Tools > My Blocks
- Robot Educator > Basics > Straight Move
- Robot Educator > Beyond Basics > Multitasking
- Robot Educator > Beyond Basics > Loop
- Robot Educator > Beyond Basics > Data Wires
- Robot Educator > Beyond Basics > Math – Basic

告知学生, 他们今天将会使用变量模块。

您可能希望向学生讲解一个常量与一个变量的区别。

- 一个常量可被程序反复不断地调用。这些固定值只有在程序停止运行时才可以被用户编辑。
- 变量是一个程序存储数值的方式, 而这些数值将会被该程序调用。不同之处在于, 变量的值可在程序运行过程中被反复不断地覆盖。

您可能需要将常量模块与移动转向模块和显示块放在一起展示。

向学生展示如何使用变量模块, 然后让他们创建一个其中包含变量的程序。

反思 (35分钟)

编写第一段程序, 实现触动传感器被按压时, 机器人加速, 然后让学生思考, 如何拓展程序可使机器人减速。

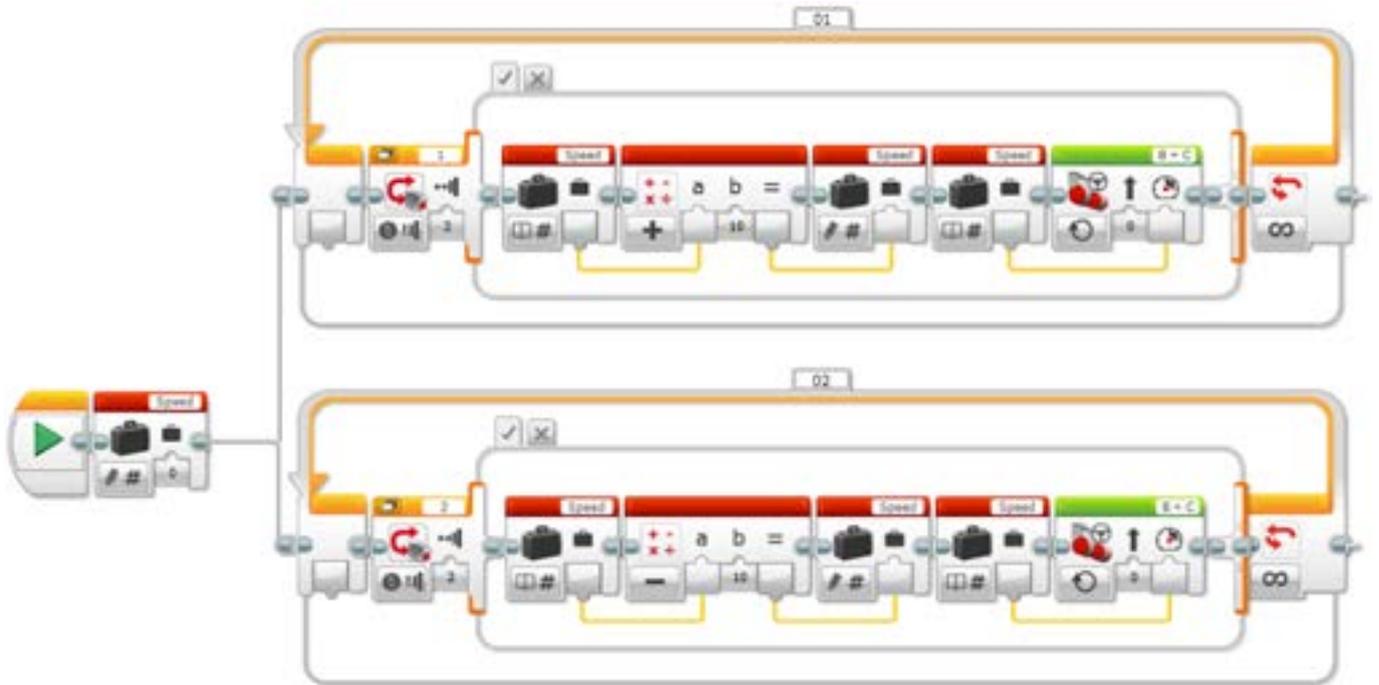
一种解决方案是引入第二个无限循环, 与首个编程练习中使用的循环类似。该循环应使用不同的触动传感器端口 (增添另一个传感器)。

数学模块应被改为减法运算而不是加法。

就多任务方面需要提醒学生, 他们在本次课程中需要用到这方面的知识。

可能的解决方案

文件名称: CODING-06.EV3 (Tab:2)



通过变量增加和降低速度

- 1.启动程序。
- 2.创建一个名为“速度”的变量模块并赋予其0值,然后启动两个任务。

任务1

- 3.如果触动传感器1受到按压:
 - a.读取“速度”变量
 - b.使变量读取值增加10
 - c.将结果写入“速度”变量
 - d.读取“速度”变量
 - e.以设定的速度值启动电机B和C,该速度值即为“速度”变量中存储的数值

否则

(不操作)

- 4.一直重复步骤3a至3e。

任务2

- 5.如果触动传感器2受到按压:
 - a.读取“速度”变量
 - b.将变量读取值减去10
 - c.将结果写入“速度”变量
 - d.读取“速度”变量
 - e.以设定的速度值启动电机B和C,该速度值即为“速度”变量中存储的数值

否则

(不操作)

- 6.一直重复步骤5a至5e。

本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分层教学部分

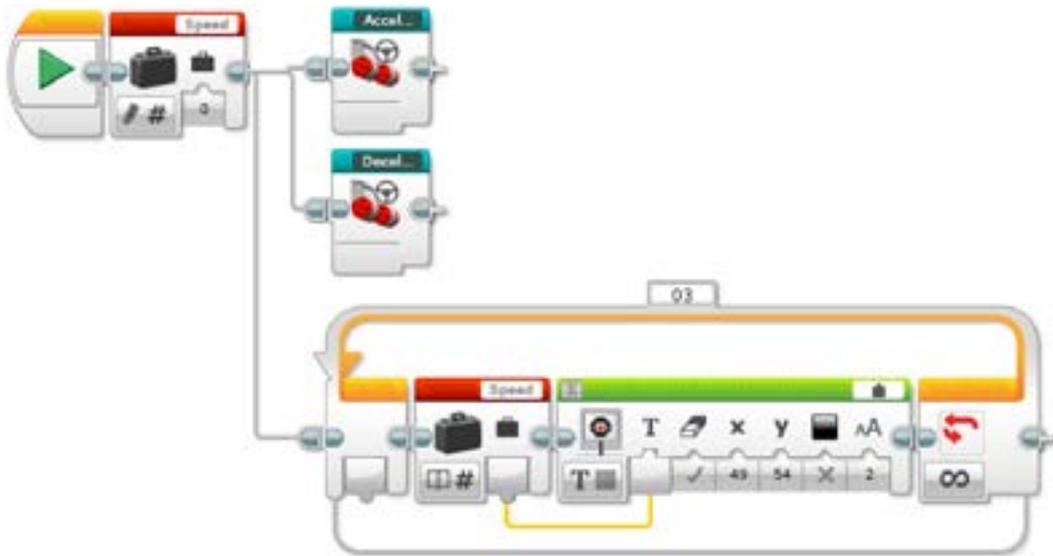
现在学生已实现通过按压一个（或两个）按钮对轮式机器人进行加速和减速，他们可以扩展自己的程序，实现显示自己机器人的移动速度。学生将会使用显示块来显示变量模块的存储值。

本次课程中将会学到的一个新技巧是创建我的模块。在下面的解决方案中，可以看到两个这样的模块。我的模块允许用户创建他们已经完成的子程序。

在下面的例子中，我们已经提取了加速和减速循环，并相应创建了两个我的模块。进行此操作出于两个原因：首先，可以节省空间；其次，这些子程序可被应用于其它程序。在EV3软件的Robot Educator部分可以找到关于我的模块的教程。

可能的解决方案

文件名称：CODING-06.EV3 (Tab:3)



通过变量和显示实现增加和降低速度

- 1.启动程序。
- 2.创建一个名为“速度”的变量模块并赋予其0值，然后启动三个任务。

任务1

- 3.启动我的模块“加速”。

任务2

- 4.启动我的模块“减速”。

任务3

- 5.读取“速度”变量。
- 6.显示“速度”变量的存储值。
- 7.一直重复步骤5至6。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

分享

允许学生选择最适合自己的工具, 以获取和分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案, 这样他们可以比较不同的编程语言。

重要提示:

下文提供了一个可能的解决方案, 其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权, 故就用户体验和技术方面的质量问题, 不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>, 获取所需的全部安装信息。当您使用其他编程语言完成活动时, 我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-07_1.C

```
#pragma config(Sensor, S1,          touchSensor,    sensorEV3 _ Touch)
#pragma config(Sensor, S2,          touchSensor2,   sensorEV3 _ Touch)
#pragma config(Motor,  motorB,                rightMotor,     tmotorEV3 _ Large,
PIDControl, driveRight, encoder)
#pragma config(Motor,  motorC,                leftMotor,      tmotorEV3 _ Large,
PIDControl, driveLeft, encoder)

/*
创建一个程序, 可通过按压触动传感器来增加机器人的速度。
*/
```

```

task main()
{
    //创建一个整数型变量以存储速度值。
    int speed = 0;
    //一直重复我们的控制循环。
    while(true)
    {
        //当我按压触动传感器的按钮时。
        if(getTouchValue(touchSensor) == 1)
        {
            // 将我们的“速度”变量增加10
            if(speed < 100) speed = speed + 10;
            //创建一个循环, 等待触动传感器按钮被松开。
            while(getTouchValue(touchSensor) == 1)
            {
                //等待按钮被松开。
                sleep(10);
            }

            //将电机B和电机C的速度值赋给“速度”变量
            setMotorSpeed(motorB, speed);
            setMotorSpeed(motorC, speed);
        }
    }
}

```

本课程的程序解决方案可从此处下载：
<http://www.LEGOeducation.com>



下一步是什么？

漫游机车



巡航控制

学生学习单

设计一个巡航控制程序来协助驾驶员，减轻他们的驾驶压力。

学生学习单

联系

确保你能回答下列问题：

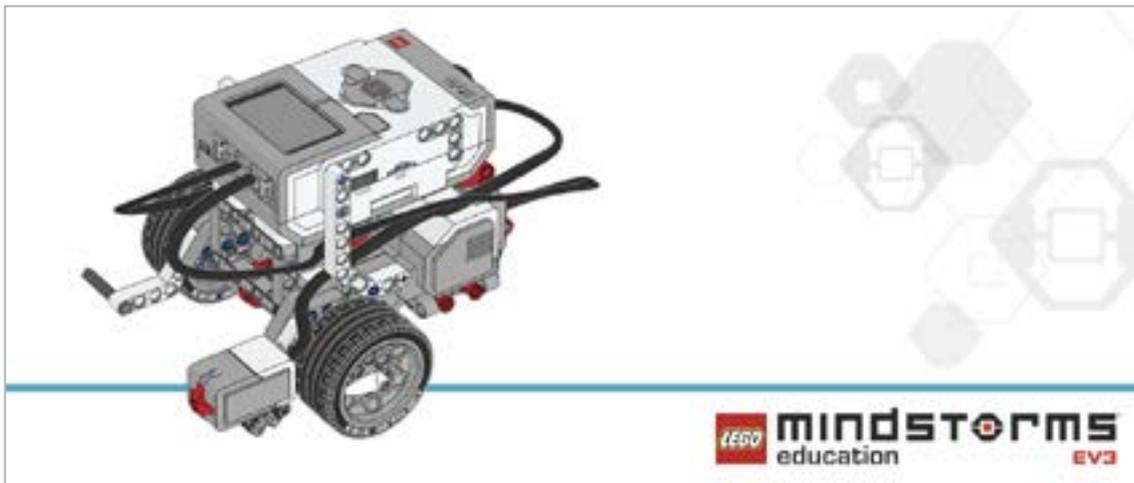
- 驾驶过程中哪些因素会使驾驶员感到压力？
- 在长时间驾驶过程中，我们如何能帮助提高驾驶员的安全性？

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



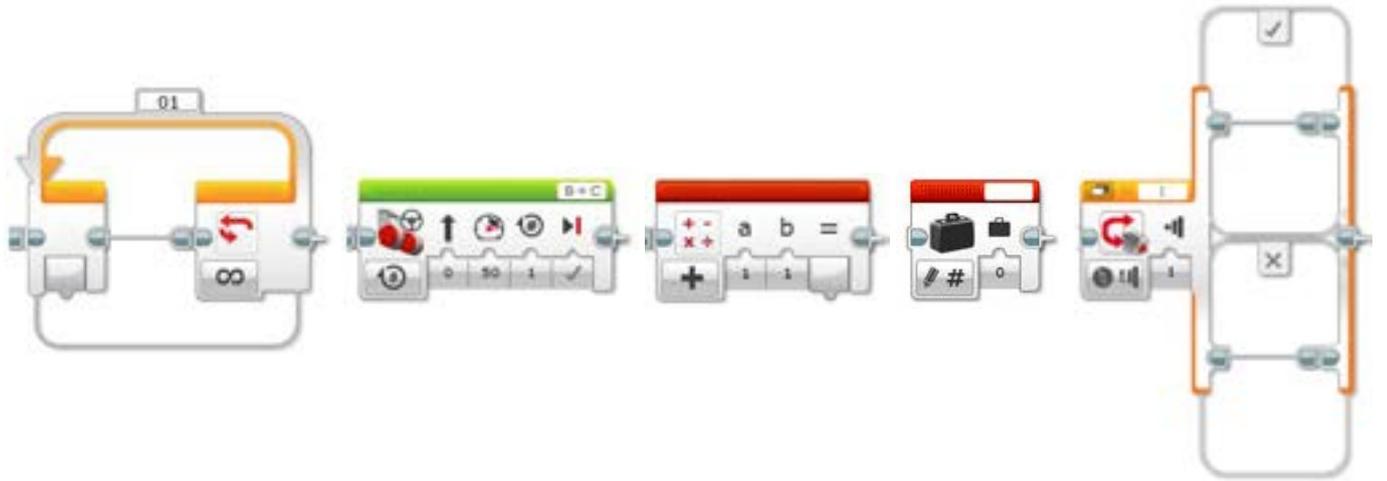
程序

为机器人创建一个巡航控制程序，就像今天许多汽车中都配有的那种。你需要使用两个触动传感器来模拟汽车方向盘上的巡航控制按钮。

触动传感器被按压时，汽车会以每次增量为10的速度加速。

使用变量模块来定义这个可以被“设定”的速度。确保移动转向模块的模式被设置为“On”，而不是“On for”（如秒数、度数或圈数）。

考虑将这些模块应用于你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

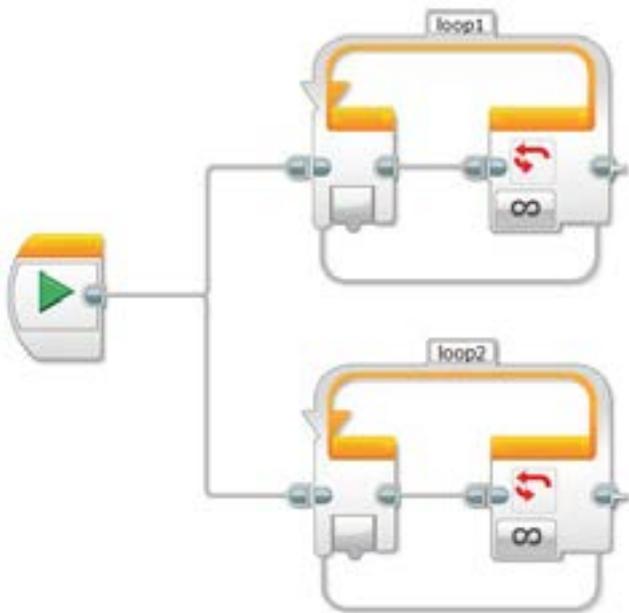
反思

改进你的程序：当触动传感器被按压时，汽车会以每次减少量为10的速度减速。

这可以通过添加第二个循环和一个切换模块来实现。

触动传感器将会被加入第二个循环，并且数学模块应被设置为减法运算，而不是加法。要注意，你将采用多任务方式，有两条程序线同时运行。

考虑将这些模块加入你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学部分

改进程序：将机器人的速度（电机功率）显示在EV3程序块显示屏上。
考虑将这些模块加入你的解决方案：



我的模块这时可能会派上用场，因为该模块可以为编程界面节省空间，并且子程序（我的模块）可再次被用于你所编制的其它程序。如果想要学习如何使用我的模块，可向你的老师求助。

思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分享

请思考以下问题：

你经历了哪些挑战？你的程序有没有为你带来什么惊喜？

你的程序能够更简洁吗？你是否使用很多模块？

是否存在一种更有效的程序构建方式？

你的程序如何能够应用到现实生活中？

拓展

探究用于本次课程的基于文本的编程解决方案，并对比这些采用不同编程语言的解决方案。



漫游机车

设计一辆无人驾驶汽车，可十分安全地行驶在街道上。

学习目标

学生将：

- 合理使用数据结构，如线性表、表格和数组。
- 加深理解布尔逻辑（如“与”、“或”和“非”），以及其在电路和编程中的一些应用
- 使用变量模块来存储信息
- 使用阵列运算模块

词汇

输入、输出、伪代码、变量、布尔逻辑、数组

年级

6-8

学科

工程、STEM（科学、技术、工程和数学）、编程

课程持续时间

45-90分钟

难度

高级

标准

《新一代科学教育标准》(NGSS)
MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. /MS-ETS1-4

计算机科学教师协会 (CSTA) 标准

2-A-2-1 / 2-A-7-2 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-5-6
/ 2-A-5-7 / 2-A-4-8 / 2-A-3-9 / 2-A-6-10 / 2-C-7-11 / 2-C-
4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人核心套装
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC软件（可选）

漫游机车

联系 (5分钟)

本次课程主要涉及数组，以及如何利用这些数组控制学生轮式机器人的动作。学生将会探索数组的工作原理，以及为什么它们在计算机编程中如此重要。他们还将学习如何将一个数组加入自己的程序。

围绕下列问题进行展开课堂讨论：

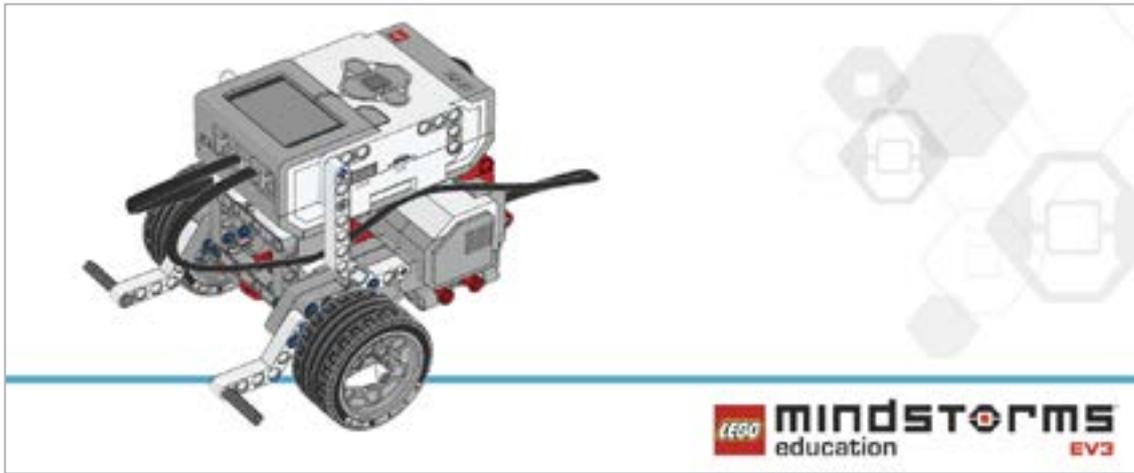
- 无人驾驶汽车的工作原理是什么？
- 怎样才能确保无人驾驶汽车的安全？
- 无人驾驶汽车需要进行哪些类型的动作？
- 可通过转动 GPS 设备中的方向，将此次编程任务与编程转向联系起来。让学生分组讨论，他们如何利用转动方向对自己机器人的转向进行编程，从而可让机器人沿着路径行进。

允许学生选择最适合自己的获取及分享想法的工具。鼓励他们以文本、视频、图像、草图笔记或其它创意性媒介的方式，将自己的想法记录下来。

建构 (15 到 30 分钟)

搭建

学生将会构建 Robot Educator 基本模型。



在对机器人进行编程之前，先让学生对搭建的系统进行检查：

- 连接线是否正确地从电机连接到端口 B 和 C？
- 机器人的轮子是否正确安装？
- 机器人的轮子能否自由旋转？

程序

向学生讲解，他们将通过 EV3 程序块上的按钮对自己的机器人进行编程，让机器人按照记录的一系列指令进行运动，上述按钮的参数设置如下：

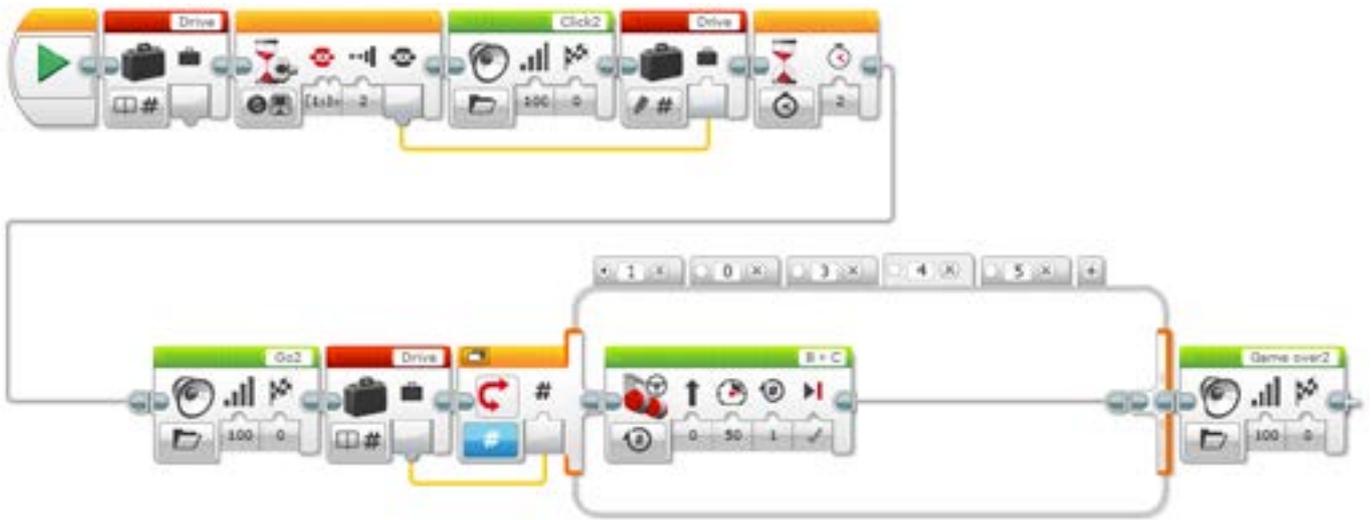
- 向上按钮受到碰撞时，机器人向前移动30厘米
- 向下按钮受到碰撞时，机器人向后移动30厘米
- 向左按钮受到碰撞时，机器人向左转动90度
- 向右按钮受到碰撞时，机器人向右转动90度

他们的程序应分为两个部分：第一部分，收集序列（如向前、向前、向左、向前、向右）；第二部分，使机器人按照上述顺序移动。

一开始，他们应该对程序进行限制，只在数组中记录一个数据（一个仅包含一个数据的数组是一个变量）。允许学生选择最适合自己的获取及分享伪代码的工具。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

可能的解决方案

文件名称：CODING-08.EV3 (Tab:1)



记录一个动作让机器人移动

- 1.启动程序。
- 2.创建一个名为“驾驶”的变量模块。
- 3.等待一个程序块按钮被碰撞。
- 4.播放声音“Click 2”。
- 5.记录变量“驾驶”中被按压按钮对应的数值。
- 6.等待2秒。
- 7.播放声音“G02”。
- 8.读取存储在变量“驾驶”中的数值，并将该值发送给开关语句。
- 9.数字开关：
 - a.如果Drive = 1，机器人向左转弯。
 - b.如果Drive = 3，机器人向右转弯。
 - c.如果Drive = 4，机器人直线向前移动两圈（轮子）。
 - d.如果Drive = 5，机器人直线向后移动两圈（轮子）。
- 10.播放声音“Game Over 2”。

本课程的程序解决方案可从此处下载：

<http://www.LEGOeducation.com>

备注

可让学生参阅 Robot Educator 教程以获取更多帮助。

在 EV3 机器人软件中:

- Robot Educator > Beyond Basics > Arrays
- Robot Educator > Tools > My Blocks
- Robot Educator > Beyond Basics > Loop
- Robot Educator > Beyond Basics > Switch
- Robot Educator > Beyond Basics > Variables
- Robot Educator > Basics > Straight Move
- Robot Educator > Basics > Curved Move

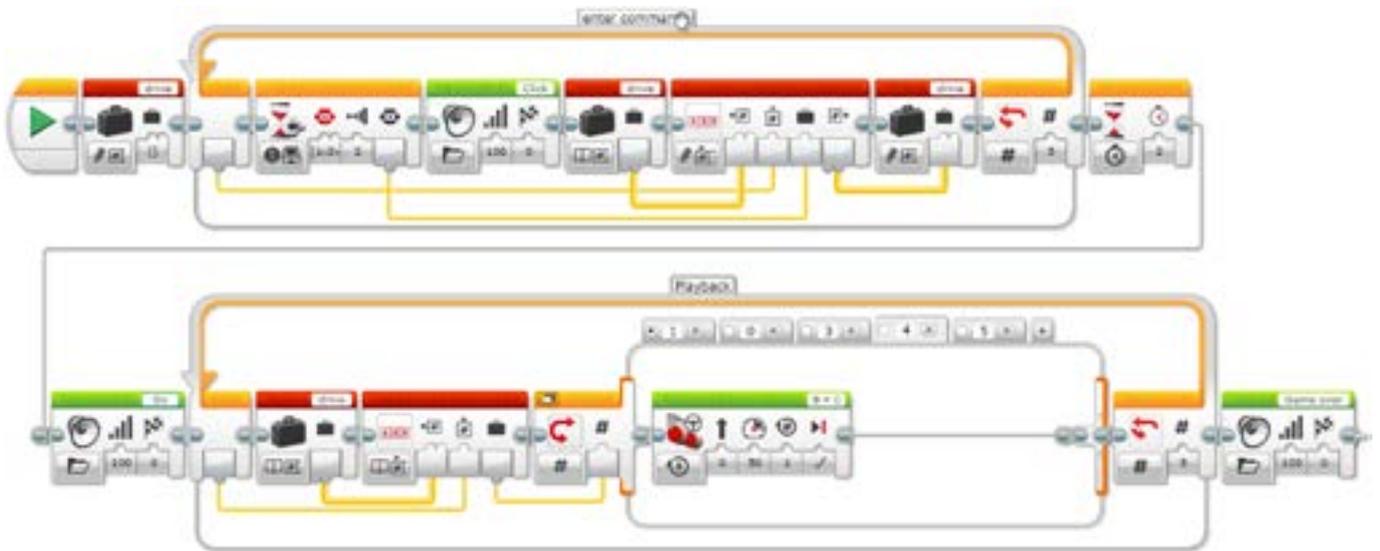
阵列运算模块用来存储一系列数据。它常被描述为一张表格, 具体表现为一行多列。

反思 (35 分钟)

让学生改进自己的程序, 利用数组记录自己机器人的五个动作。

可能的解决方案

文件名称: CODING-08.EV3 (Tab:2)



本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分层教学部分

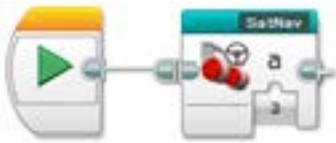
向学生演示如何创建一个我的模块，然后为其 添加一个参数。

请注意，编程面板中会增加一个新选项卡，其会显示我的模块中所包含的内容。创建好我的模块后，用户需要将“a”参数与程序的两个输入连接起来。我们的程序中将会包含两个循环模块。这样就会实现将数据直接传递给我的模块，而不是其所包含的程序。

需要向学生指出，这种方式可允许他们轻松输入所需要的步数。确保学生充分理解，在创建的我的模块中添加一个参数可允许他们向该模块输入数据。

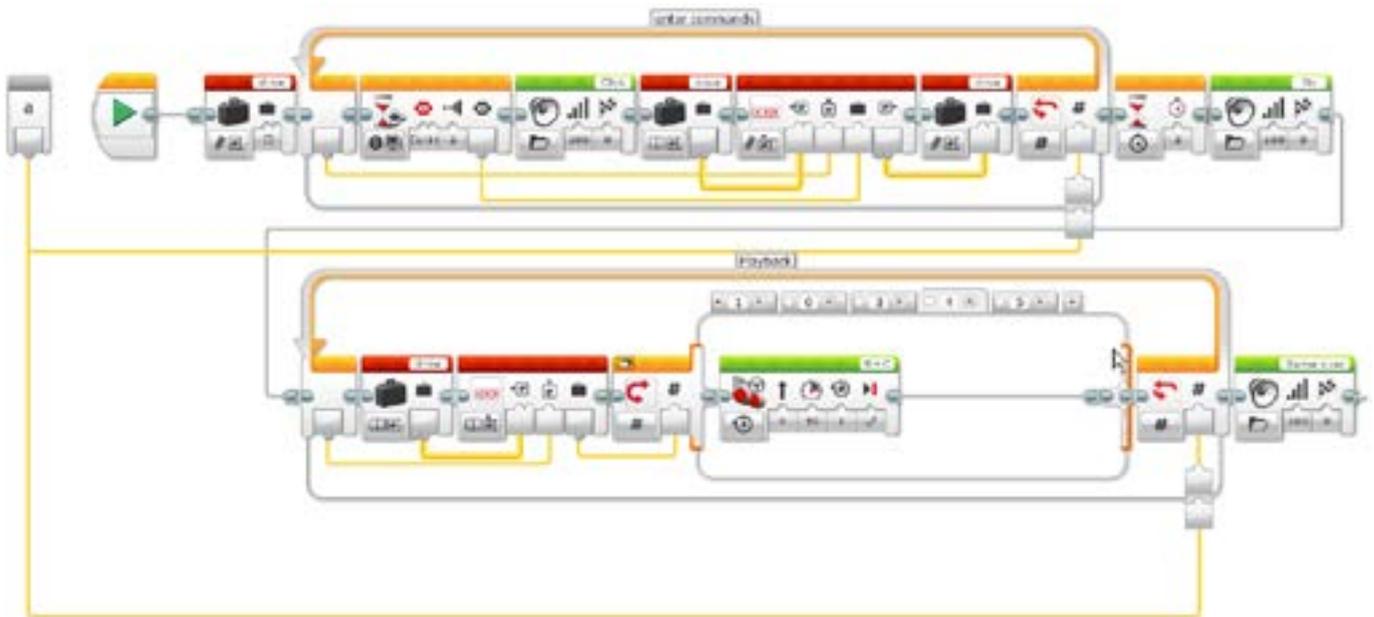
可能的解决方案

文件名称: CODING-08.EV3 (Tab:2)



可能的解决方案

文件名称: CODING-08.EV3 (Tab: SATNAV)



本课程的程序解决方案可从此处下载:

<http://www.LEGOeducation.com>

分享

允许学生选择最适合自己的工具，以获取和分享他们的创造、独特的思维及学习过程。鼓励他们使用文本、视频、图像、草图笔记或其它创意性媒介。

将学生聚集起来，分享他们编程的成功经验。询问他们如何能够改进原有程序。

创建一个我的模块，并使其带有一个参数，能否让学生更方便地编制合适的步骤数？

要求学生考虑替代解决方案。他们能否构想出新的编程方式，让机器人在房间内四处移动？

评估条件

可在“评估”章节内参看用以评估计算思维的具体评价标准。

拓展 (45 分钟)

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案，这样他们可以比较不同的编程语言。

重要提示：

下文提供了一个可能的解决方案，其采用了基于文本的编程语言 ROBOTC。您也可以选择使用其它任何与乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人兼容的基于文本的编程语言。

乐高教育不拥有 ROBOTC 平台的所有权，故就用户体验和技术方面的质量问题，不能提供任何支持或保证。请访问 ROBOTC 网站 <http://www.robotc.net/>，获取所需的全部安装信息。当您使用其他编程语言完成活动时，我们始终建议您重新安装正式版的乐高 MINDSTORMS EV3 程序块固件。

可能的解决方案

文件名称: CODING-08_1.C

```
#pragma config(StandardModel, "EV3 _ REMBOT")
```

```
/*
```

利用 EV3 程序块上的按钮为机器人编程，使其能够四处移动。可向 EV3 程序块输入 5 条命令。（向左按钮=1，向右按钮=3，向上按钮=4，向下按钮=5）

```
*/
```

```
int drive[5];
```

```
task main()
```

```
{
```

```
    for(int i = 0; i < 5; i++) //i = i + 1
```

```

{
  while(getButtonPress(buttonAny) == 0)
  {
    //等待任何按钮被按下。
  }
  if(getButtonPress(buttonLeft) == 1)      drive[i] = 1;
  else if(getButtonPress(buttonRight) == 1) drive[i] = 3;
  else if(getButtonPress(buttonUp) == 1)   drive[i] = 4;
  else if(getButtonPress(buttonDown) == 1) drive[i] = 5;
  playSoundFile("Click");
  while(bSoundActive)
  {
    sleep(10);
  }
  while(getButtonPress(buttonAny) == 1)
  {
    //等待所有按钮被松开。
    sleep(10);
  }
}
sleep(2000);
playSoundFile("Go");
while(bSoundActive)
{
  sleep(10);
}
for(int i = 0; i < 5; i++)
{
  if(drive[i] == 1)
  {
    //向左转向代码。
    moveMotorTarget(motorC, 360, 50);
    waitUntilMotorStop(motorC);
  }
  else if(drive[i] == 3)
  {
    //向右转向代码。
    moveMotorTarget(motorB, 360, 50);
    waitUntilMotorStop(motorB);
  }
  else if(drive[i] == 4)
  {
    //向前代码。

```

```
        moveMotorTarget(motorB, 360, 50);
        moveMotorTarget(motorC, 360, 50);
        waitUntilMotorStop(motorB);
        waitUntilMotorStop(motorC);
    }
    else if(drive[i] == 5)
    {
        //向后代码。
        moveMotorTarget(motorB, -360, -50);
        moveMotorTarget(motorC, -360, -50);
        waitUntilMotorStop(motorB);
        waitUntilMotorStop(motorC);
    }
}

playSoundFile("Game over");
while(bSoundActive)
{
    sleep(10);
}
}
```

本课程的程序解决方案可从此处下载:
<http://www.LEGOeducation.com>



下一步是什么?

智能交叉路口



漫游机车

学生学习单

设计一辆无人驾驶汽车,可十分安全地行驶在街道上。

学生学习单

联系

确保你能回答下列问题：

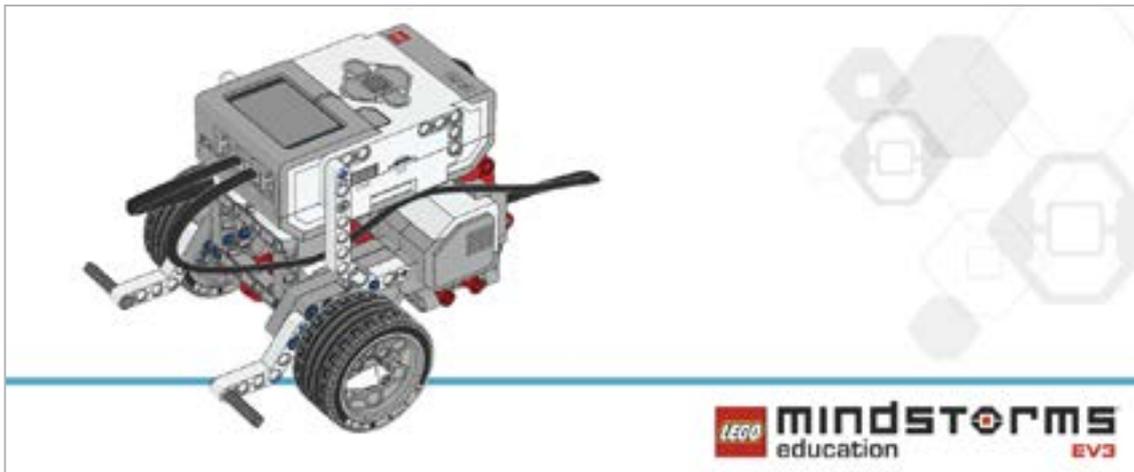
- 无人驾驶汽车的工作原理是什么？
- 怎样才能确保无人驾驶汽车的安全？
- 无人驾驶汽车需要进行哪些类型的动作？

思考你所了解的情况，然后进行记录。运用你自己的语言描述问题。创意性地录制你的发现与想法。

建构

搭建

从构建这款模型开始。



程序

为机器人创建一个包含数组的程序，可实现其利用 EV3 程序块按钮在房间内四处移动。四个程序块按钮可以用作方向控制单元（即左、右、后退和前进）。

开始时，将程序限制为一条命令。

提示1：你的程序将会包括两个不同的阶段：

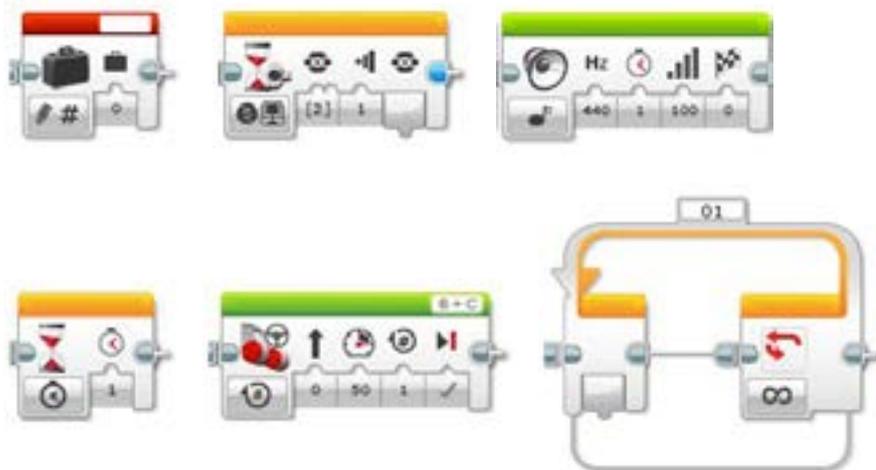
1. 收集数据
2. 使用那些数据

提示2：为实现上述两个阶段，本次课程需要用到两个循环模块。

提示3：使用变量模块通常需要三个步骤：

1. 读取变量模块
2. 向该模块添加信息
3. 写入模块以保存新数据

考虑将这些模块应用于你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

反思

改进程序，采用数组记录机器人的五个动作。

考虑将这些模块加入你的解决方案：



思考你所了解的情况，然后进行记录。描述你进行该任务的伪代码。创意性地录制你的发现与想法。

分层教学部分

创建一个我的模块,可轻松编辑程序的步数。

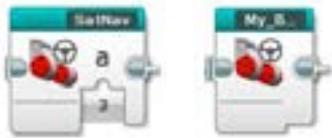
改动运动步骤的数量(从当前的5个到其它数量)需要编辑程序中的两个循环模块。通过创建一个我的模块(带有一个参数),可显著简化程序编辑过程。我的模块可轻松、明晰地改变循环的次数。

提示1: 在创建我的模块时,要突出显示那些需包括的模块,但不包含开始模块。

提示2: 如果需要在稍后阶段输入一些参数,请确保一个参数已添加至我的模块,具体如下图所示。创建该模块时可使用“+”键。

提示3: 该参数必须与程序中某个模块的输入相连。在我们的例子中,包含两个循环。

考虑将这些模块加入你的解决方案:



分享

思考你所了解的情况,然后进行记录。创意性地录制和展示你的想法、创作和发现。

请思考以下问题:

你经历了哪些挑战?在本次课程中有没有一些惊喜?

你的程序能够更简洁吗?你是否使用很多模块?

是否存在一种更有效的程序构建方式?

你的程序如何能够应用到现实生活中?

拓展

探究用于本次课程的基于文本的编程解决方案,并对比这些采用不同编程语言的解决方案。



智能交叉路口

设计一辆可以安全穿过交叉路口的无人驾驶汽车。

学习目标

学生将:

针对现实生活中的问题和实物系统, 设计、使用和评估相应的解决方案

词汇

设计概要、原型、标准

年级

6-8

学科

工程、STEM (科学、技术、工程和数学)、编程

课程持续时间

120+分钟

难度

高级

标准

《新一代科学教育标准》(NGSS)

MS-ETS1-1. / MS-ETS1-2. / MS-ETS1-3. /MS-ETS1-4

计算机科学教师协会 (CSTA) 标准

2-A-2-1 / 2-A-7-2 / 2-A-7-3 / 2-A-7-4 / 2-A-5-5 / 2-A-5-6 / 2-A-5-7 / 2-A-4-8 / 2-A-3-9 / 2-A-6-10 / 2-C-7-11 / 2-C-4-12 / 2-D-5-16 / 2-I-1-20

所需材料

乐高® MINDSTORMS® 头脑风暴教育 EV3 机器人核心套装
乐高 MINDSTORMS EV3 机器人软件或编程应用程序
ROBOTC 软件 (可选)

头脑风暴

让学生参看设计概要, 并针对问题构思多个可能的解决方案。
他们打算把哪些功能添加到自己的拼砌作品和编制的程序中?
他们是否需要更改自己机器人的物理设计?
让他们记下自己所有的想法及草图。

选择最佳解决方案

鼓励学生权衡自己每个想法的优点和缺点。然后他们应该最终敲定一个设计, 以继续进行下一步。

建构

搭建机器人并对解决方案进行编程

让学生搭建机器人并对其编程, 以解决问题。

反思

测试和分析

让学生测试和评估他们解决方案的效率。

检查和修正

让学生持续改进他们的解决方案, 直到满足设计标准。

交流

鼓励学生分享其学习过程。通过学生手中的文档工具, 向他们提供分享自己思维、观点和反思的机会。

拓展

使用基于文本的编程语言进行编程

让学生探索基于文本的编程解决方案, 这样他们可以比较不同的编程语言。

