



Autonomous Parking

Design cars that can park themselves safely without driver intervention.

Learning Objectives

Students will:

Understand that algorithms are capable of carrying out a series of instructions in order

Explore the concept of Outputs by compare different ways in which a wheeled robot can move

Vocabulary

Output, algorithm, pseudocode, debug

Grades

6-8

Subjects

Engineering, STEM, Coding

Duration

45-90 Minutes

Difficulty

Beginner

Standards

NGSS

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

CSTA

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

Materials Needed

LEGO® MINDSTORMS Education EV3 core set
LEGO MINDSTORMS EV3 Software or Programming app
ROBOTC software (optional)

Autonomous Parking

CONNECT (5 minutes)

Ignite a classroom discussion around the following questions:

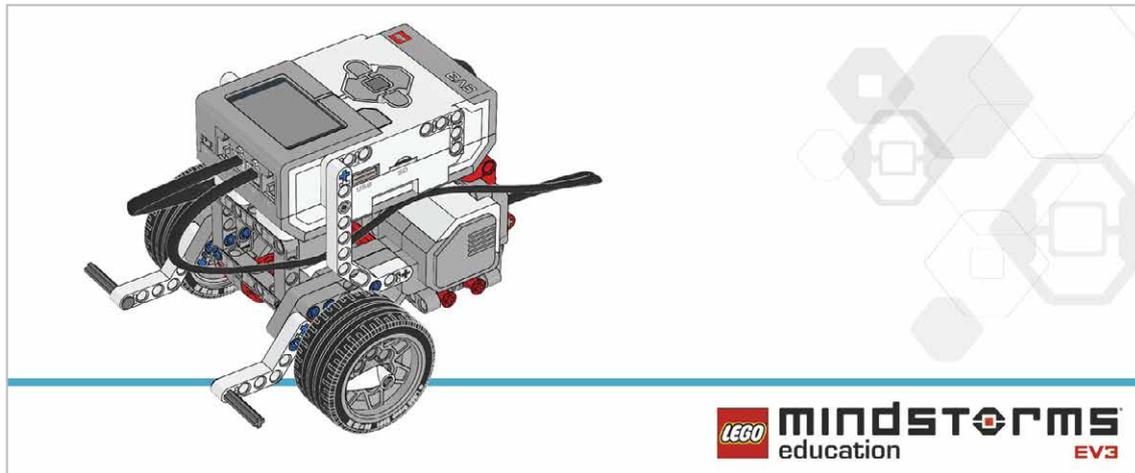
- How do autonomous cars work?
- What would it take to ensure that autonomous cars are safe?
- What types of movements do autonomous cars need to perform?

Allow the students to select the tool(s) they find most appropriate for capturing and sharing their ideas. Encourage them document their thoughts using text, videos, images, sketchnotes, or another creative medium.

CONSTRUCT (15 to 30 minutes)

Build

Students will construct the Robot Educator base model, which is a basic wheeled robot.



Have the students perform the following building check before they program their robots:

- Are the wires correctly connected from the motors to ports B and C?
- Are the wheels correctly installed?
- Are the wheels rotating freely?

Program

Have the students begin a new project in the EV3 programming environment. Have them practice by creating a program that will make the robot turn three times. Encourage the students to explore different ways of making the wheeled robot move. Have them describe the effect(s) of altering the parameters of each of the blocks that they use.

POSSIBLE SOLUTION

FILENAME: CODING-01.EV3 (Tab:1)



THREE POINT TURN

1. Start the Program
2. Turn the driving base and stop after 1.5 seconds.
3. Turn the driving base left and stop after 1 second.
4. Move the driving base forward for 3 seconds.

Program solutions for this lesson are available for download at:
<http://www.LEGOeducation.com>

Note

Refer students to the Robot Educator Tutorials for further assistance.

In the EV3 Software :

Robot Educator > Basics > Straight Move

Robot Educator > Basics > Curved Move

Robot Educator > Basics > Tank Move

In the EV3 Programming App :

Robot Educator > Curved Move

Set up

Before assigning the next task, make sure that you have marked the path the robots must follow and that there is enough space to complete the task. It is a good idea to have students work on a big table or on the floor.

CONTEMPLATE (35 minutes)

Have the students choose one or all of these autonomous driving scenarios to program:

- Parallel parking
- Angle parking
- Perpendicular parking

They should use a different Programming Canvas for each solution.

Allow the students to select the tool(s) they find most appropriate for capturing and sharing their pseudocode. Encourage them to use text, videos, images, sketchnotes, or another creative medium.

POSSIBLE SOLUTION

FILENAME: CODING-01.EV3 (Tab:2)



PARALLEL PARKING IN AUTONOMOUS MODE

1. Start the Program
2. Drive forward in a straight line at the desired speed
3. Wait for 1 second
4. Reverse motor rotation while turning for 1,5 rotation
5. Reverse motor rotation while turning the other way for 1,5 rotation
6. Drive backward in a straight line for 0,5 rotation
7. Drive forward in a straight line for 1 rotation

Program solutions for this lesson are available for download at:
<http://www.LEGOeducation.com>

Differentiation Option

Have the students create a program that simulates appropriate warning lights while parking (e.g., use the EV3 Brick Status Light to display reverse warning lights).

POSSIBLE SOLUTION

FILENAME: CSACTIVITY1.EV3 (Tab:3)



SIMULATING REVERSE GEAR AND REVERSE WARNING LIGHTS

1. Start the Program
2. Drive forward in a straight line at the desired speed
3. Wait for 1 second
4. Turn light on (reverse light)
5. Reverse motor rotation while turning for 1,5 rotation
6. Reverse motor rotation while turning the other way for 1,5 rotation
7. Drive backward in a straight line for 0,5 rotation
8. Drive forward in a straight line for 1 rotation

Program solutions for this lesson are available for download at:
<http://www.LEGOeducation.com>

Share

Allow the students to select the tool(s) they find most appropriate for capturing and sharing their creations, unique thinking, and learning process. Encourage them to use text, videos, images, sketchnotes, or another creative medium.

Ask one or two groups to demonstrate their programs. Discuss what works well and what could be improved.

Assessment Opportunity

Specific rubrics for assessing computational think can be found under 'Assessment'.

CONTINUE (45 minutes)

Using text-based programming

Have the students explore text-based programming solutions so they can compare different programming languages.

IMPORTANT

The following is a possible solution using the text-based programming language ROBOTC. You may choose to use any other LEGO MINDSTORMS Education EV3 compatible text-based programming languages.

LEGO Education has no ownership of the ROBOTC platform and does not provide any support or guarantee of the quality of the user experience and technology used. All required set up information is provided by ROBOTC at <http://www.robotc.net/>. We recommend always to reinstall the official LEGO MINDSTORMS EV3 Brick firmware when you finish using other programming languages.

POSSIBLE SOLUTION

FILENAME: CODING-01_1.C (Three Point Turn)

```
#pragma config(StandardModel, "EV3_REMBOT")
/*
Create a three point turn using timing and steering.
*/

task main()
{
    //Turn to the right and stop after 1.5 seconds.
    setMotorSpeed(motorB, 75);
    setMotorSpeed(motorC, 30);
    sleep(1500);

    //Reverse to the left and stop after 1 second.
    setMotorSpeed(motorB, -30);
    setMotorSpeed(motorC, -75);
    sleep(1000);

    //You should now be straight and facing the other way. Drive off.
    setMotorSpeed(motorB, 50);
    setMotorSpeed(motorC, 50);
    sleep(3000);
}
```

Program solutions for this lesson are available for download at:
<http://www.legoeducation.com>



WHAT IS NEXT?

Reversing safely