



Roaming Vehicles

Design an autonomous car that is safe enough to drive on the streets.

Learning Objectives

Students will:

- Make appropriate use of data structures such as lists, tables and arrays
- Extend Boolean logic (such as AND, OR and NOT) and some of its uses in circuits and programming
- Use the Variable Block to store information
- Use the Array Operations Block

Vocabulary

Input, output, pseudocode, variable, Boolean logic, array

Grades

6-8

Subjects

Engineering, STEM, Coding

Duration

45-90 Minutes

Difficulty

Advanced

Standards

NGSS

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

CSTA

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

Materials Needed

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

Roaming Vehicles

CONNECT (5 minutes)

This lesson will focus on arrays and how they can be used to control the actions of the students' wheeled robots. Students will discover what arrays are, how they work, and why they are important in computer programming. They will also learn how to incorporate an array into their program.

Ignite a classroom discussion around the following questions:

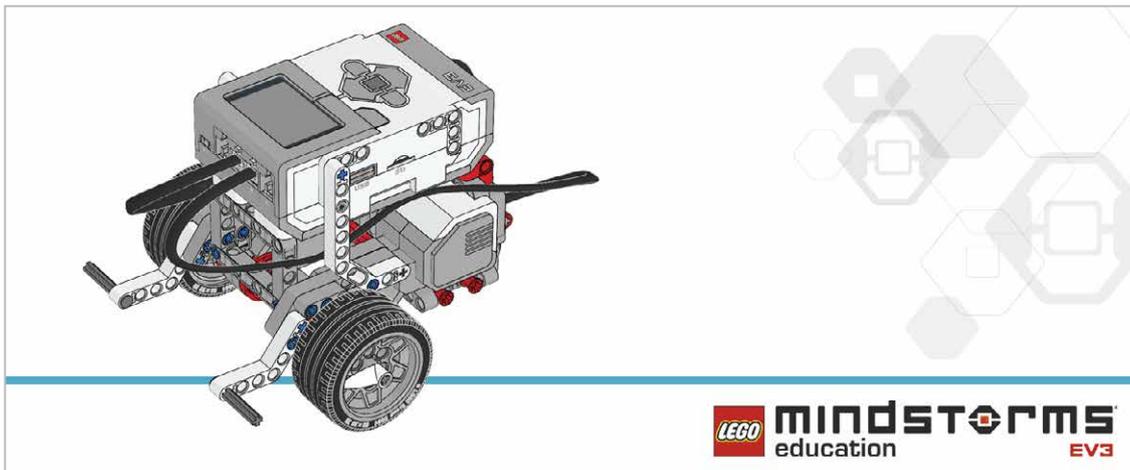
- How do autonomous cars work?
- What would it take to ensure that autonomous cars are safe?
- What types of movements do autonomous cars need to perform?
- Relate this programming task to programming turn by turn directions from a GPS device. Ask the students to discuss, in groups, how they could program turn by turn directions into their robot to make it move around a course.

Allow the students to select the tool(s) they find most appropriate for capturing and sharing their ideas. Encourage them to 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.



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

Explain to the students that they are going to program their robot to move according to a recorded set of instructions given to it through the buttons on the EV3 Brick according to these parameters:

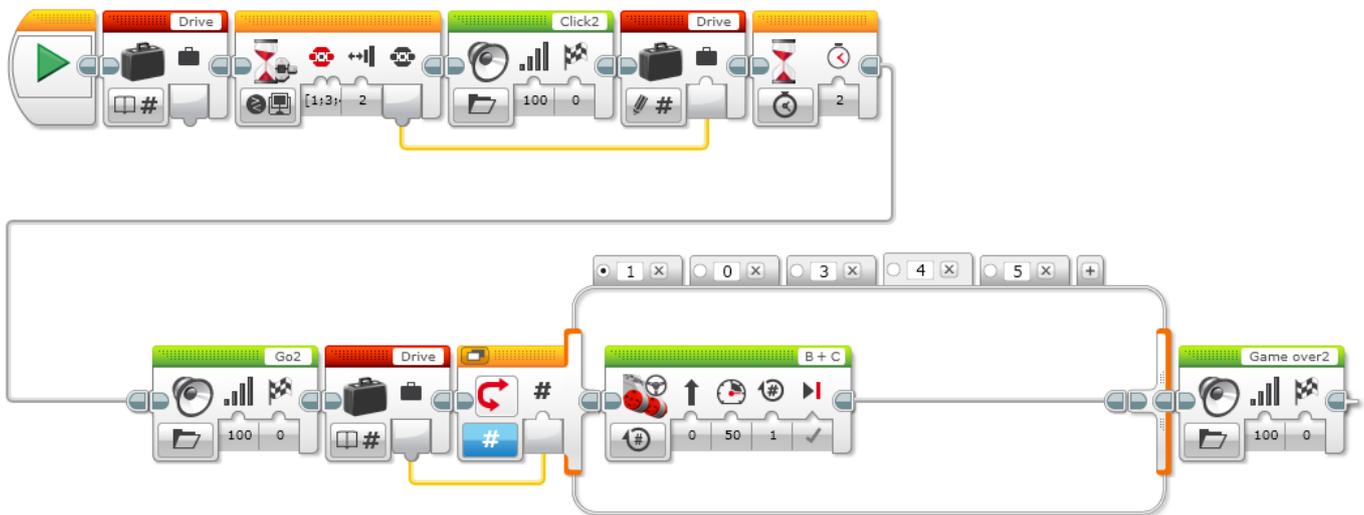
- Up Button bumped, the robot moves forward 30 cm
- Down Button bumped, the robot moves backward 30 cm
- Left Button bumped, the robot turns 90 degrees left
- Right Button bumped, the robot turns 90 degrees right

Their program should be divided into two sections: the first is to gather the sequence (e.g., forward, forward, left, forward, right), and the second is to make the robot move according to the sequence. To start with, they should limit the program to recording one item in the array (an array composed of one item is a variable).

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-08.EV3 (Tab:1)



RECORDING ONE ACTION TO MAKE THE ROBOT MOVE

1. Start the program.
2. Create a Variable Block called "Drive".
3. Wait for a Brick Button to be bumped.
4. Play sound "Click 2".
5. Record the numerical value of the pressed button in the variable "Drive".
6. Wait for 2 seconds.
7. Play sound "G02".
8. Read the number stored in variable "Drive" and send the value to a switch.
9. Numeric switch:
 - a. If Drive = 1, curve turn the robot left.
 - b. If Drive = 3, curve turn the robot right.
 - c. If Drive = 4, move the robot straight forward for 2 rotations of the wheels.
 - d. If Drive = 5, move the robot straight backward for 2 rotations of the wheels.
10. Play sound "Game Over 2".

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 > 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

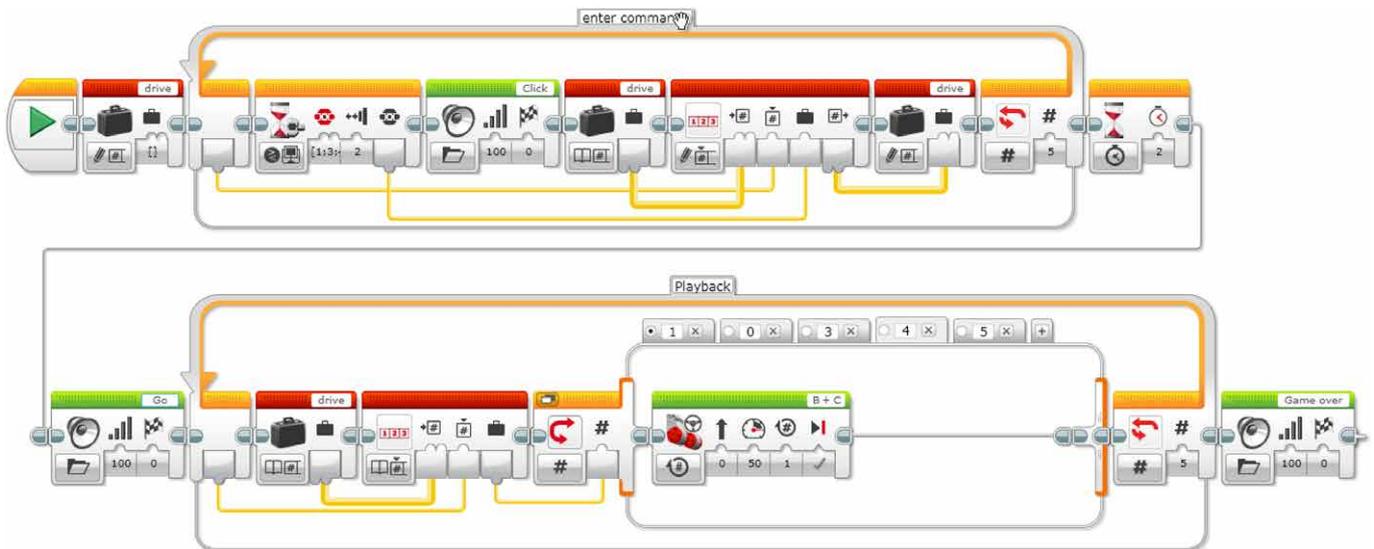
The Array Operations Block is used to store a sequence of data. It is often described as a table consisting of one row with multiple columns.

CONTEMPLATE (35 minutes)

Ask the students to improve their program by using an array to record five actions for their robot.

POSSIBLE SOLUTION

FILENAME: CODING-08.EV3 (Tab:2)



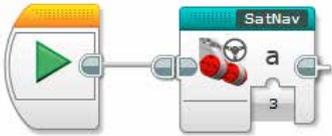
Program solutions for this lesson are available for download at:
<http://www.LEGOeducation.com>

Differentiation Option

Demonstrate to the students how to create a My Block and then add a parameter to it. Notice that a new tab is added to the Programming Palette showing what is contained within the My Block. To complete the My Block, the user needs to connect the “a” parameter to two inputs within the program. In our program, this will be the two Loop Blocks. This allows data to be directly entered into the My Block, rather than the program contained within it. Point out that this allows students to easily enter their desired number of steps. Make sure that the students understand that adding a parameter in the My Block Builder allows them to enter data into the My Block that is created.

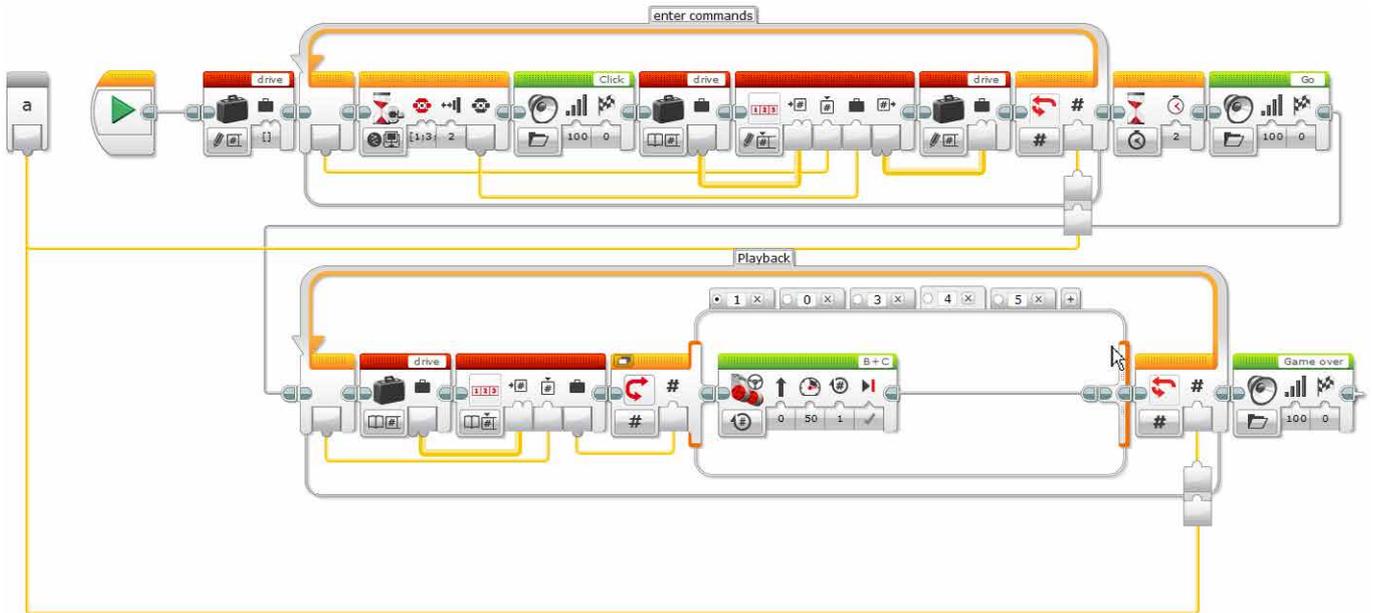
POSSIBLE SOLUTION

FILENAME: CODING-08.EV3 (Tab:2)



POSSIBLE SOLUTION

FILENAME: CODING-08.EV3 (Tab: SATNAV)



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.

Bring the students together to share their programming successes. Ask how they could have improved their programs.

Does creating a My Block with a parameter make it easier for the students to program the right number of steps?

Ask the students to consider alternative solutions. Can they come up with a new way of programming the robot to move around the room?

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-08_1.C

```
#pragma config(StandardModel, "EV3_REMBOT")
/*
Use the buttons on the EV3 brick to program the robot to move around.
5 commands can be entered into the EV3 brick. ( Left Button = 1, Right
Button = 3, Up Button = 4, Down Button = 5)
*/
int drive[5];
task main()
{
  for(int i = 0; i < 5; i++) //i = i + 1
```

```

{
  while(getButtonPress(buttonAny) == 0)
  {
    //Wait for Any Button to be pressed.
  }
  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)
  {
    //Wait for All Buttons to be released.
    sleep(10);
  }
}
sleep(2000);
playSoundFile("Go");
while(bSoundActive)
{
  sleep(10);
}
for(int i = 0; i < 5; i++)
{
  if(drive[i] == 1)
  {
    //Turn Left Code.
    moveMotorTarget(motorC, 360, 50);
    waitUntilMotorStop(motorC);
  }
  else if(drive[i] == 3)
  {
    //Turn Right Code.
    moveMotorTarget(motorB, 360, 50);
    waitUntilMotorStop(motorB);
  }
  else if(drive[i] == 4)
  {
    //Forward Code.

```

```
        moveMotorTarget(motorB, 360, 50);
        moveMotorTarget(motorC, 360, 50);
        waitUntilMotorStop(motorB);
        waitUntilMotorStop(motorC);
    }
    else if(drive[i] == 5)
    {
        //Backward Code.
        moveMotorTarget(motorB, -360, -50);
        moveMotorTarget(motorC, -360, -50);
        waitUntilMotorStop(motorB);
        waitUntilMotorStop(motorC);
    }
}

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

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



WHAT IS NEXT?

Autonomous Intersection