

# Lesson 1: Let's move!

**Subject:** STEAM

**Duration:** 45 minutes

**Grade(s):** 5<sup>th</sup> and up

**Difficulty:** Beginner

## ★ Lesson Objectives

*By the end of this lesson, students will be able to:*

- Drive the mBot2 with precision.

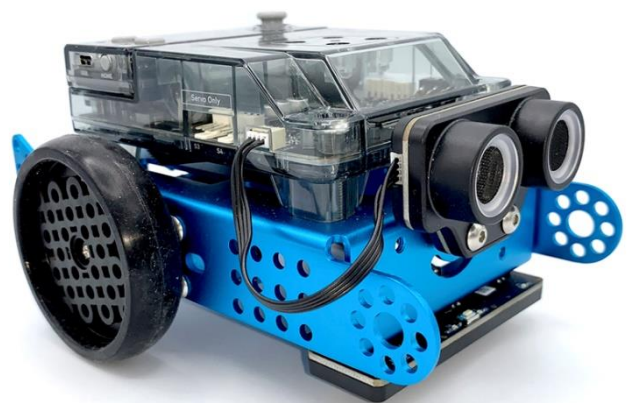
## ★ Overview

Robots are autonomous machines that replace human effort; they can perceive their environment and perform computer programs to make decisions and take actions. A computer program is a set of instructions and conditions run by a computer to perform a task. You may think you have never seen or used a robot, but that chance is slim. Robots can take many different shapes and have different capabilities depending on their context. For example, robots are present in households, such as the vacuum cleaner robot or a lawn mowing robot; you can also find them in factories assembling products. We have even sent robots to other planets.

## 🔗 Focus

*By the end of the lesson, students will know:*

- What movements can mBot2 do
- What different programming blocks you can use to make the mBot2 move



## Pre-lesson Checklist

What do you need?

- PC or laptop (with USB output) with the mBlock software installed, the web version (also for ChromeBook), or a tablet with the mBlock app installed
- The mBot2 with a CyberPi
- A USB-C cable or Makeblock Bluetooth dongle
- A3 paper
- Pens

## Lesson plan

Duration	Content
5 minutes	<b>1. Warming up</b> <ul style="list-style-type: none"><li>• Sensors and data in everyday life.</li><li>• What is the CyberPi?</li></ul>
10 minutes	<b>2. Hands-on</b> <ul style="list-style-type: none"><li>• Getting acquainted with the different programming blocks of the mBot2.</li></ul>
25 minutes	<b>3. Trying out</b> <ul style="list-style-type: none"><li>• Programming your own robot through a self-made maze.</li></ul>
5 minutes	<b>4. Wrap-up</b> <ul style="list-style-type: none"><li>• Showtime: show what you did with your robot in a fun, short movie for later discussion.</li><li>• If your teacher allows, share the end result on social media with the hashtag #mBot2moves</li><li>• Reflection: What are you most proud of? What would you like to improve about your robot?</li></ul>

## 1. Warming up (5 min)

### Step 1: Warming up

This step consists of two parts:

1. Robots in everyday life
2. Getting to know the mBot2

#### 1. Robots in everyday life

We have robots in many different places. Some people have them in their homes and others work with them every day. Robots can take many different shapes and have different capabilities depending on their context. Can you think of three robots that you have to deal with (regularly) in everyday life?

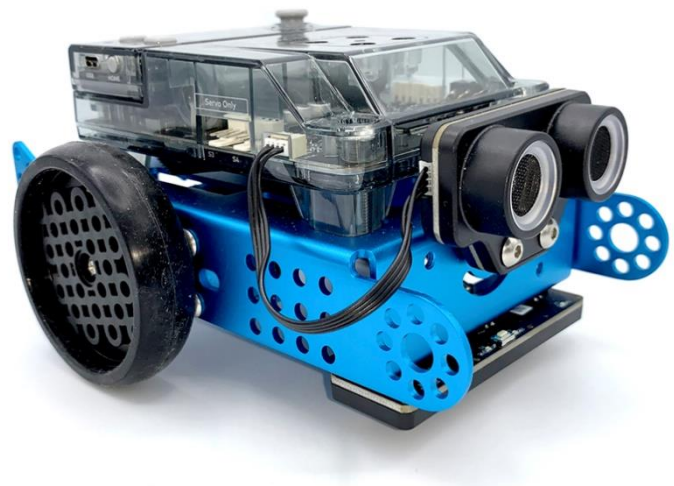


There are probably many more robots that have been developed that you have never seen. Search the Internet for three more robots that you can use in everyday life.

There is a good chance that you will have to deal with robots later on your professional or personal life. There are robots working in many different companies. Usually, these robots perform tasks that are repetitive, that need to be done with great accuracy, or that are dangerous. In this lesson we will be working mainly on programming a robot to move accurately. Can you think of three professions or businesses where it is very important for a robot to work very accurately?

## 2. Getting to know the mBot2

The mBot2 is a programmable robot equipped with different components that allow it to sense, act and communicate with its environment. The mBot2 is intended for students to learn about Computer Science and Technology, including ways that you can use a robot in real life. The mBot2 can be programmed with the help of mBlock. In mBlock, programming can be as easy as dragging and assembling blocks. As you will see in this lesson, you can, for instance, program the mBot2 to drive through a maze - with your help (not autonomously). mBot2 comes with a pair of special motors, which can record the rotation of the axle and therefore the speed and distance the robot has travelled. They are called encoder motors due to the sensor that is built-in (encoder). This type of motors allows to control specific parameters such as the rotation angle and speed of the motors. You will learn how to make use of these motors in the following steps.



## 2. Hands-on (10 min)

### Step 2: Hands-on

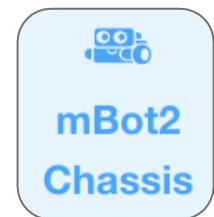
This step consists of two parts:

1. Getting acquainted with the different programming blocks of the mBot2.
2. Recreating and testing some programming examples to control the mBot2.

### 1. Getting acquainted with the different programming blocks of the mBot2

As you observed previously, robots can be used in many different ways. If you have thought of a purpose for a robot, most likely it means that you know what the robot needs to do to help you, and some of these tasks might need to be executed with certain precision.

When you start programming the mBot2, you will notice that there are many different coding blocks you can use to make the robot move. You will find these blocks in mBlock 5 in the category 'Chassis'. These coding blocks are blue.



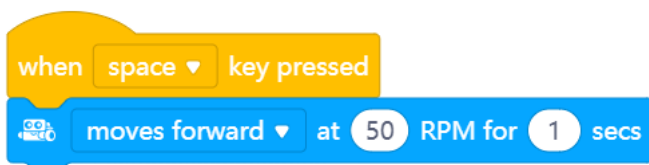
Below you can see some examples of the coding blocks needed to make the robot move. Some of them look similar, yet they all do something different. For this lesson, we will work in Live mode. Make sure the correct mode is selected in the software. The differences between Live mode and Upload mode will be explained in lesson 2.

#### Code block:

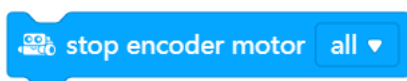


This code block allows you to move the mBot2 forward, backward, left and right at a specific rotational speed of the wheels and for a number of seconds.

The example below shows how you can make the mBot2 move for two seconds at a speed of 50 rotations per minute. This is useful, for example, when the mBot2 needs to push a load forward.

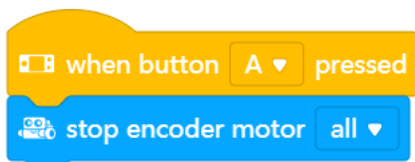


#### Code block:

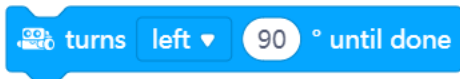


This code block allows you to make your robot stop moving. This can be a useful block during a testing phase. If a program does not work as expected you can use this block to make the robot stop immediately.

For example, you can make the robot stop when the A button of the CyberPi is pressed. To do this, use the programming example below.

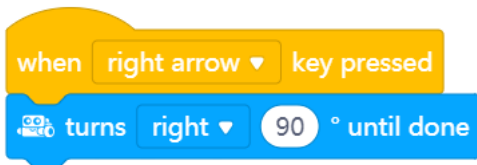
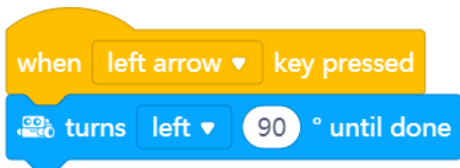


### Code block:



With this code block you can make the mBot2 rotate by a number of degrees, and you can choose whether the rotation should be to the left or to the right.

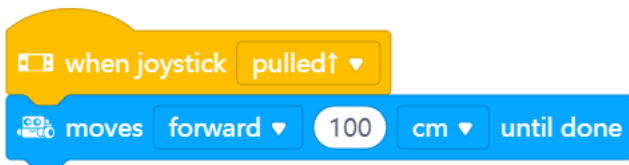
When the programming example below is set, the mBot2 can be controlled by using the arrow keys. When pressing the right arrow, the mBot2 will turn 90 degrees to the right, when pressing the left arrow, it will turn 90 degrees to the left.



### Code block:



With this code block you can make the mBot2 move forward or backward a certain distance. When you use the programming example below, the mBot2 will move 100 cm forward. To start this, move the joystick of the CyberPi upwards.








### Code block:

 encoder motor EM1  rotates at 50 RPM, encoder motor EM2  rotates at 50 RPM



You can also control the motors of the mBot2 independently, and this is one of the blocks which you can use for that.

In the example below, the mBot2 will move in a curve for 3 seconds before it stops. Notice that the value for one of the motors is negative; this is because the motors are mounted in opposite directions, so in order to move the robot in one direction, you need to make one of the motors rotate the opposite way. What would happen if both wheels rotate at 40 RPM?

 when button B  pressed

 encoder motor EM1  rotates at 50 RPM, encoder motor EM2  rotates at -30 RPM



wait 3 seconds




 stop encoder motor all 



### Code block:

 encoder motor (1) EM1   rotated angle (°)

Because the motors of mBot2 can measure their speed and rotation, you can also read these values. The speed measured is the rotational speed in Rounds per Minute ( $360^\circ = 1$  round; in science, it is more common to use "degree per second" ( $1 \text{ RPM} = 6^\circ \text{ per second}$ )). In mBlock 5, if you tick the box next to this block, you can read the values on the stage above the panda. The programming example below changes the values of the rotation back to zero, and then orders the mBot2 to move forward for one second. What value do you read on the stage after the robot moved?

 when joystick middle pressed 

 reset encoder motor all   rotated angle

 moves forward  at 50 RPM for 1 secs

## 2. Recreating and testing some programming examples to control the mBot2.

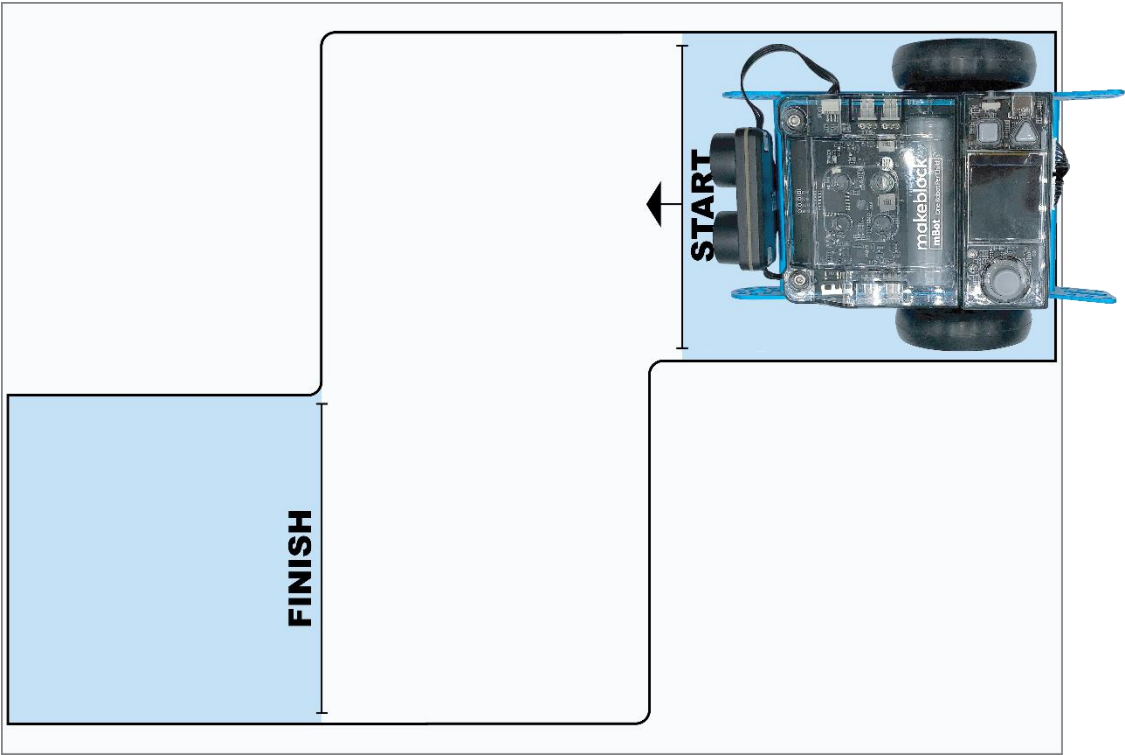
Below you can see some pairs of blocks. Try out each pair side by side. What happens? What are the differences between the code blocks?

	Option 1	Option 2
1		
2		
3		

## 3. Trying out (25 min)

### Step 3: Trying it out

It is now time to put your mBot2 to work. Take an A3 paper and draw a maze that your mBot2 has to drive through. Don't make it too complicated and take into account the width of the robot. The mBot2 doesn't have to find the right route by itself. The mBot2 just needs to drive the route that you program. Need inspiration? Below you can see an example.





Use the knowledge you gained in 'Step 2' of this lesson. Of course, you can do plenty of experimenting yourself with the different programming examples in mBlock5.

When doing this assignment, it is helpful to use the following step-by-step plan.

	Explanation
Step 1: What do you want to do?	<ul style="list-style-type: none"> <li>• What route do you want the mBot2 to drive?</li> <li>• What parts does the maze consist of?</li> <li>• How long are the distances the robot needs to travel?</li> <li>• Does the robot also need to make turns? Which way and by how many degrees?</li> </ul>
Step 2: What do you need?	<ul style="list-style-type: none"> <li>• What do you need in addition to the mBot2?</li> </ul>
Step 3: What code blocks do you need to make the mBot2 drive?	<ul style="list-style-type: none"> <li>• How are you going to make the mBot2 drive?</li> <li>• What code blocks will you use?</li> <li>• Make a brief description on how your program works (pseudocode/natural language, flowchart or UML)</li> <li>• If you need further explanation, you can discuss with your fellow students, the teacher, or do a research on the topic. There is help available for every code block in mBlock as well.</li> </ul>
Step 4: Testing and implementation	<ul style="list-style-type: none"> <li>• Is the first version ready? Test it! During the testing round, write down areas of improvement.</li> <li>• Work on the improvement points until your mBot2 drives through your maze without errors.</li> </ul>

Is it possible to get the mBot2 to drive through the maze without a hitch? Then challenge yourself with a more difficult maze. Or have one of your classmates work out a route for you.

## 4. Wrap-up (5 min)

### Step 4: Wrap-up

Did you succeed in driving the robot through the maze completely without error?

In this lesson you have been introduced to robots in everyday life and to the mBot2. You know how to control the movements of mBot2 and which programming blocks you can use for this.

It is now time for a short reflection. Think on your own and discuss with the group:

- What do you think turned out well?
- What could be better?
- Which parts of the lesson did you find easy and which did you find more difficult?
- What would you like more explanation about?
- Who could help you with that?