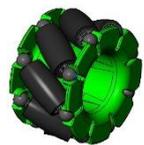


## 2-wheel robot



If you have built the model using the building instructions, you have probably already noticed two new components – the Omniwheels and the support wheel.

Actuators and technical accessories installed in the model:

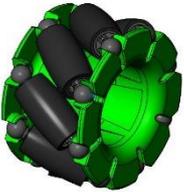
| Mini Motor 6-9V   | Gears   | Omnwheel   | Support wheel   |
|---|---|--|---|
|  |  |  |  |

You can click an image for an explanation of the actuator or technical accessory.

The “2-wheel robot model” is divided into 2 programming tasks:

|   |  |
|---|--|
| <p><b>Task 1</b></p> <p>2-wheeled_robot_1.rpp</p> | <p><u>Programming level 3</u></p> <p>The model should travel forward for a set time - 3 seconds - then remain stopped for 1 second, and then travel backwards for 2 seconds, then stop.</p>  |
| <p><b>Task 2</b></p> <p>2-wheeled_robot_2.rpp</p> | <p><u>Programming level 3</u></p> <p>Just as in the 1st task, the model should travel forwards for some distance, then turn right. This function should be carried out 4 times. Then the same function should be completed in the other direction. Finally, the program will stop.</p> |

## Task 1



Before starting to program, read the chapter “Basic information - Omniwheels” It provides important information you will need to program the model.

**Important:** While you are programming, you must pay attention to the direction of rotation of the wheels. Rotation to the left means the wheels will turn forwards, and to the right means that they will turn backwards.

Subroutines are used in the following program. Therefore, it would be helpful to read chapter 4 in the ROBO Pro Help first.

Now you can start with the 1st task.

To do so, start the ROBO Pro program as usual and check to ensure programmer level 3 is switched on.



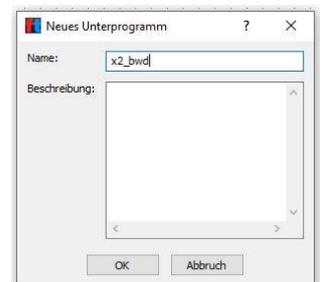
Begin with the “Start” command. And now, you will encounter a new programming feature.

Program sequences that repeat over and over again are written into something called a “subroutine”, which can be inserted into the program as needed.



If you want to create a subroutine, choose “Create new subroutine” in the tool bar.

A dialogue field will appear where you can enter a name, such as “x2\_bwd”. Press OK to confirm the entry and access the programming screen to create subroutines.



A subroutine always starts with the command “Subroutine input”.

This command call is found under Element group – Program elements – “Subroutines I/O”

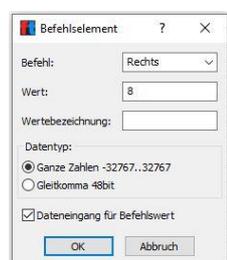
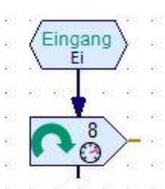


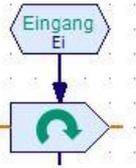
A subroutine is always ended with the command “Subroutine output”.

The subroutine starts with the input command. Then a command like the one shown follows. Read about the command in the ROBO Pro Help section.



Drag the command from Program elements – “Commands” under the input command and connect it to this command. Open the dialogue field for the command and activate the “Data input for command value”.

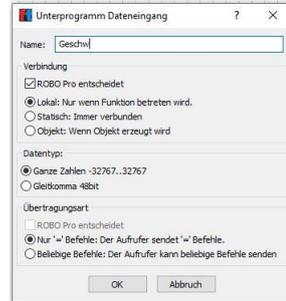




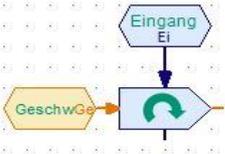
The command display will change, and you will see an orange input on the left side of the command.



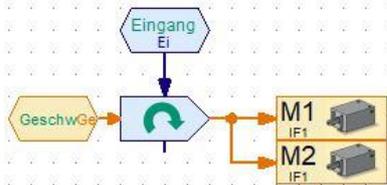
You have certainly noticed that the directional arrow is also shown with the display above. The value for motor speed is missing, and is set via the left command input. To do so, insert the command “Subroutine data input” from the subroutine I/O.



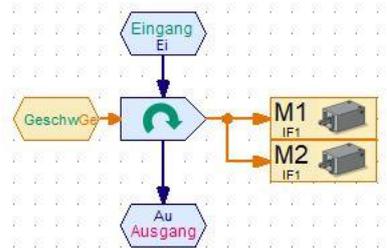
Change the name in the field, for instance to “Speed” for speed, then end it by pressing OK.



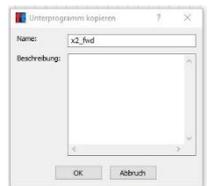
The right output is assigned 2 motor commands M1 and M2. Insert these into the program from Program elements – “Inputs, outputs”.



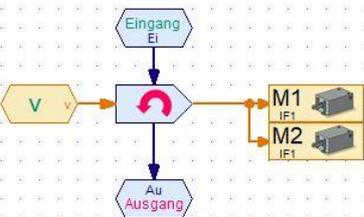
Then insert the command “Subroutine output” as the last command.



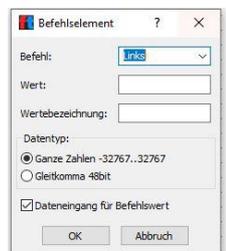
Now, create the subroutine to cause the motors to turn to the left (forward). To do so, remain in subroutine creation and activate the button “Copy current subroutine”. Change the file name in the dialogue field, for instance to “x2\_fwd”.



The last change you need to make is the direction of rotation. You will change it from “right” to “left”.



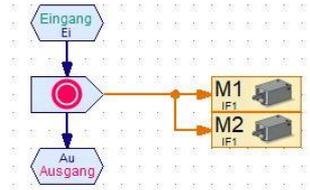
Then, this subroutine is complete.



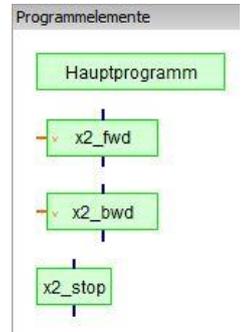
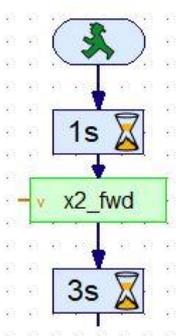
The two subroutines are available as tabs in the program bar beside the “Main program” tab.



Before you continue programming the main program, create the subroutine shown here to stop the motors.

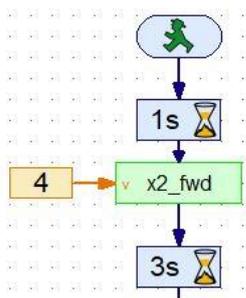


Use the program bar to switch to “Main program”.

You have already inserted the start command. The vehicle should travel forward for 3 seconds after a wait time of 1 second. To do so, insert the subroutine “x2\_fwd” after the wait time, and then a wait command of 3 seconds. Insert the subroutine via Element groups – Loaded programs.

Insert a “Constant” at the value input of the subroutine, and select the value “4” for the speed of the two motors. Then they will run at a moderate speed that can be changed as needed.



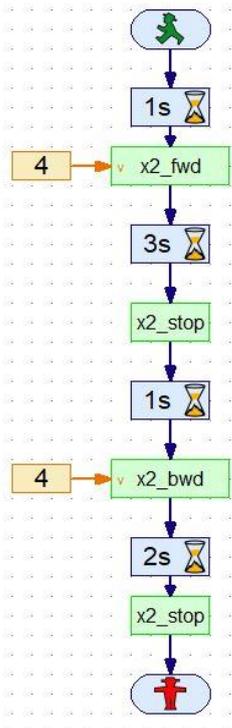
Now the motors will stop, then a wait time of 1 second follows, then backwards travel for 3 seconds. Then the motors will stop once again, and the program is ended with the Stop command.

Insert the relevant modules into the program and save it under the name

**“2-wheeled\_robot\_1.rpp”**

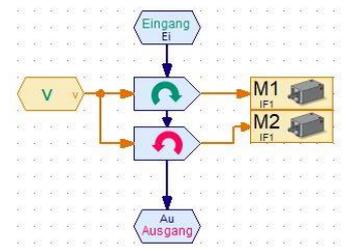
on your computer. Then test the program with your model. If everything is programmed correctly, the model will travel forward for 3 seconds and then back for 3 seconds, then stop.

Look at the next task, and think about which commands or subroutines you can use from the 1st task, and which new subroutines you will need to create.

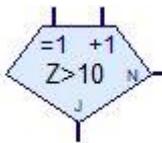


## Task 2

First, create a subroutine with the name “x2\_rotR” for “Rotation right”. It is important that both motors are activated individually. M1 turns to the right and M2 to the left.

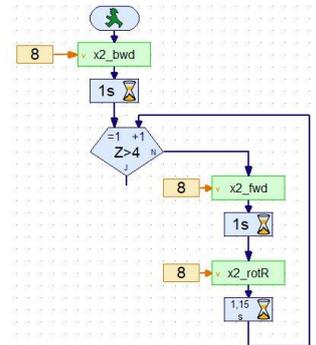


Use “Copy subroutine” to create the program once again, and assign it the name x2\_rotL for rotation to the left. You only need to change the running direction of the two motors. M1 to the left and M2 to the right.



You will use the command “FOR loop” from the basic elements to complete the loop 4 times.

Read about the command and its connections in the ROBO Pro Help section. Insert the command as shown into your program. You only need to change the number of passes to 4.



Once the value >4 is reached, the program continues via the output “J” and the program section begins where the vehicle travels in the other direction. The Stop command for the motors is inserted at the “J output” of the two FOR loops.

You can save the program under the name

**“2-wheeled\_robot\_2.rpp”**

Test the program by loading it to your controller.

Maybe you can find some other ways to control the model as well.

