

## Omniwheels base model



You will use this model to learn more fundamentals of programming, and to learn about how to use four omniwheels and how they work.

Actuators and technical accessories installed in the model:

Mini Motor	Gears	Omniwheel
		

An explanation of the components is provided on the introductory page.

The “Omniwheels base model” is divided into 2 programming tasks:

<b>Task 1</b>  Omniwheels_Base_Model_1.rpp	<u>Programming level 1</u> The model travels straight for 1s, then laterally to the left for 1.4s, back for 1s, laterally to the right for 1.4s, turns to the left for 3s and then turns to the right for 3s. Then it stops.
<b>Task 2</b>  Omniwheels_Base_Model_2.rpp	<u>Programming level 3</u> The model should travel in all possible driving directions. To do so, the driving directions should be written in subroutines for later programming tasks.

## Task 1

Since only 2 wheels were used in the previous models, unfortunately you cannot use the subroutines you have already created – you will just have to create a new one.

Before you start programming, read the explanation of the Omniwheels on the introductory page once again. The illustrations show you which motors are activated when the model turns in a certain direction, and which direction of rotation they need to be activated in.

The program is very simple, and is just a sequence of motor commands, separated by the run time.

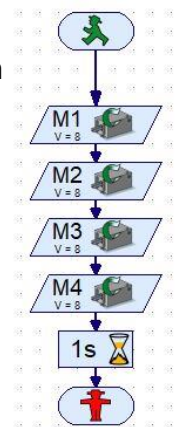
The first part of the travel sequence is shown in this part of the program. If you end this part of the program with the program stop command, you can test it out.

Then, add the other command sections in sequence.

Test out the program and save it under the name

**“Omniwheels\_Base\_Model\_1.rpp”**

on your computer.



## Task 2

Now, it's time for the second task. In this task, you will create a program similar to the one in the first task – but all driving manoeuvres will be saved in subroutines.

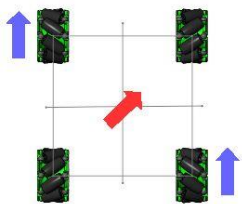
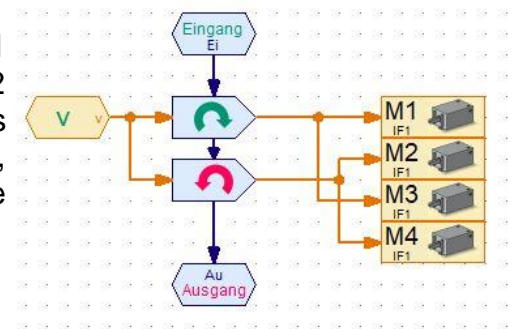
The following table shows the individual subroutines and the names you can create them under.

<b>x4_rotL</b>	<b>x4_rotR</b>	<b>x4_fwd</b>	<b>x4_bwd</b>	<b>x4_sideL</b>
Rotation left	Rotation right	Travel forward	Travel backward	Travel sideways to the left
<b>x4_sideR</b>	<b>x4_diagFL</b>	<b>x4_diagBL</b>	<b>x4_diagBR</b>	<b>x4_diagFR</b>
Travel sideways to the right	Travel forward diagonally, left	Travel backward diagonally, left	Travel backward diagonally, right	Travel forward diagonally, right

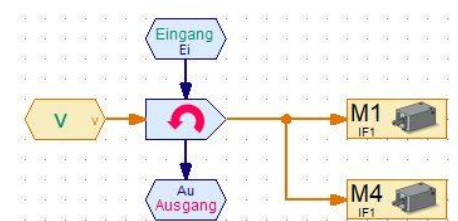
You must ensure that all four motors are always addressed at the same time in the subroutine.

Since you are already familiar with the programming, here are just a couple of tips.

Look at the program below “x4\_rotL”. In this rotation, the M1 and M3 pairs of motors have to turn to the right, and the M2 and M4 pairs have to turn to the left. The rotational speed is the same for all of the motors. In order to turn to the right, you only need to change the direction of rotation of the motors.



If you want the model to travel diagonally to the right, only 2 motors are required as shown in the image. These will turn in the same direction (to the left) and at the same speed.



Once all subroutines have been created, you can insert them into the main program. The table shows one possible sequence with times for how long each subroutine should last. The rotational speed is the same for all of the subroutines (8). The entire program runs in an endless loop.

x4_rotL	x4_rotR	x4_fwd	x4_sideL	x4_bwd
1s	1s	1s	1s	1s
x4_sideR	x4_diagFL	x4_diagBL	x4_diagBR	x4_diagFR
1s	1s	1s	1s	1s

Test out the program and save it under the name

## “Omniwheels\_Base\_Model\_2.rpp”

on your computer.

