Function model

With this model, you will learn some initial fundamentals of programming, and learn about the elements installed in the model.

If you have built the model using the building instructions, then you have already noticed a few important components. In addition to the fischertechnik components, you will use so-called “actuators” such as the encoder motor, and an LED to control a light barrier.

Actuators and sensors installed in the model.

Encoder motor	LED	Ultrasound sensor	Phototransistor

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

The “Function model” model is divided into 5 programming tasks:

Basic settings	<u>Learning level 1</u> Basic settings in the program, defining the controller configuration
Task 1	<u>Learning level 1</u> Start the encoder motor. It should run for 5 seconds, then stop again.
Task 2	<u>Learning level 1</u> The light barrier should be queried. If it is interrupted, the motor will start and run for 5 seconds. Then it will switch off again and wait until the light barrier is interrupted once again.

Task 3	<u>Learning level 1</u> The ultrasound sensor should be queried. If the value falls below a specific threshold value, the motor will start. If the threshold value is exceeded, the motor will stop.
Task 4	<u>Learning level 2</u> The motor should change its speed depending on the distance between an object and the ultrasound sensor.
Task 5	<u>Learning level 2</u> The encoder motor should not run for a specific amount of time, but rather be controlled through a number of measured pulses.

Controller configuration

Before starting the actual programming, you must first define which sensors and actuators are connected to the TXT 4.0 Controller in the controller configuration. This is described in the building instructions.

First, start the “ROBO Pro Coding” program. A start screen will appear.

Select “NEW PROJECT” here.

In the next window, choose whether the different special functions such as camera, display or both are required.

Selecting “empty” is sufficient for the basic tasks.

Leer

Now, complete the selection by pressing the “NEXT” button.

A window will appear where different options for your new project should be entered. First, you must assign it a name.

Here, for example, you assign the name “Function model”. Since you want to work with the “graphic programming”, you can conclude the input by clicking “Create”.

The programming screen will now appear; you are already familiar with it from the “Programming fundamentals” chapter.

Important: Start your TXT 4.0 Controller.
Connect the Controller and ROBO Pro Coding.

Once this step has been completed successfully, activate “Controller configuration”.

The screen will jump from the main programming screen to the

screen for controller configuration and display a selection menu, where you should first select “Controller”. Then the symbol of the TXT 4.0 Controller will be displayed.

Simply use your mouse to drag the icon to the right and into the configuration screen.
The display will change, and all available connections on the TXT 4.0 Controller will appear.

The next step is to define what should be connected to the inputs of the TXT 4.0 Controller. Use the building instructions for the model to do so. As you can see from the instructions, you will need:

- 1 encoder motor
- 1 phototransistor
- 1 light barrier LED
- 1 ultrasound distance sensor

The wiring diagram shows you which inputs and outputs these are connected to on the TXT 4.0 Controller.

First, define the required sensors. These are the phototransistor and the ultrasound sensor.

Click the selection “Input” to open the selection window and display all available sensors.

Drag these to the “Input” docking point using your mouse. You will also need to change the connection configuration there. The ultrasound sensor is changed from “I1” to “I6” and the phototransistor from “I2” to “I8”.

You will define the motor output in the next step. To do so, activate “Motor” and drag the “Encoder motor” command to the “Motor” output. The motor remains connected at “M1”.

Since the encoder motor is pulse controlled in the last task, you will need to add a pulse counter to the configuration at the input point “Counter/Digital Input”.

To do so, open the “counter” window and drag the block to the docking point. Click the small selection window and select “M1”.

To conclude the configuration, the LED for the light barrier must be defined form the “Output” selection. This is defined at the output connection, at output “03”.

The controller configuration is now complete, and you can start the

Interface test

to manually test the connected sensors and actuators.

To do so, switch from the controller configuration to the “Main program”.

You can use the interface test to test the model's function.

To do so, activate the “Interface test” button in the blue command line. The following screen will appear.

It is divided into 4 areas.

- Servo motor area
- Inputs area
- Outputs area
- Counter area.

First, look at the “Inputs” area. The two sensors - the “phototransistor” and “ultrasound distance sensor” are displayed here.

The value for the distance sensor changes when you simulate an obstacle with your hand.
Test it out for yourself.

If you want to test the phototransistor, you first have to slide slider 03 in the “Outputs” window to the right. This switches on the light barrier LED. The value for the phototransistor is switched to “1”. If you interrupt the light beam, the value switches to “0”.

Then, test out the connected encoder motor. To do so, drag the controller from “M1” to the right. The motor will run at full speed.

You can use the “cw” and “ccw” switch buttons to change the direction in which the motor runs.

When the motor is running, the display for the digital counter changes in the “Counter” window. The evaluation of this value is required for later tasks.

The last window we have to discuss is the “Servo motors” window. You can use the slider to test out an installed servo motor, but only later, when a servo motor is installed in a model. The displayed value is the centre setting of the servo.

Now, switch over to the programming screen “Main program”.

Function model – Task 1 – Learning level 1

The task is to make the encoder motor start, run for 5 seconds, and then stop after the program is started. Important: The program should **not** run in an endless loop.

To do so, drag the green command “repeat continuously – do” from the “Program start” command block and move it back to the command selection using your mouse.

Now, you need a command from the “Motor” block.

You can dock it at the free space in the “Program start” command.

Now, you need a wait command, since the motor should only run for 5 seconds. You can find this command in the “Util” block as the command “wait – s 1”. Insert the command under the motor command and change the value “1” to “5”.

Finally, you must insert a command to stop the motor. This is also found under the “Motor” block.

Dock the command under the wait time. The program is now complete, and you can try it out.

To do so, activate the “Start program” button. The program is now transmitted to the Controller and started. The motor will run for 5 seconds, then switch off again.

Function model – Task 2 – Learning level 1

The task is as follows: After the program starts, the vehicle waits until the light barrier installed in the model is interrupted.
Then the motor will start, run for 5 seconds, and stop. The program is inserted in a so-called endless loop so that the program does not have to be repeatedly restarted. This is found in the “Loops” group.

The first step is to insert the command to activate the LED between the program start and

“Continuous...”. To be certain the phototransistor will switch correctly, insert a wait time of 1 seconds after this.

To do so, insert the “wait” command from the “Util” group. The set time can be left at one second. All other commands are written in the endless loop.

The model should wait until the light barrier is interrupted. To do so, you will need the “wait until” command from the “Util” group.

Dock the command “is phototransistor ... light” from the “Input” group to it. Change the entry “light” to “dark”.

The next 3 commands are identical to the 1st task. Install a motor. It should run for 5 seconds, then stop again. Then, the program should wait for another action by the light barrier.

Test the program using “Start program”.

Function model – Task 3 – Learning level 1

The task is as follows: After the program starts, the model should wait until a measurement by the ultrasound sensor falls below a set distance value – then the motor should start. If the threshold value is exceeded, the motor will stop.

The program is integrated in an endless loop here as well.

Two new commands from the “Logic” group are required

for this task. What is this group, generally?

Boolean logic is a simple mathematical system with two values:

- **true**
- **false**

Logic blocks in ROBO Pro Coding are generally used to program conditions and loops.

Add the logic command “if ... then” to your program.

For “if” the system queries what distance is measured on the ultrasound sensor. If the distance is less than (<)10 (cm), the motor should start at “then”. The command for the ultrasound sensor is found in the “Input” group.

You can go ahead and test out the subprogram. Since the program will not shut down by itself, you can end it by clicking the button.

Now, you need to add the second logic query to the “if – then” logic. To do so, click the symbol:

A selection window will appear. Drag the command “Otherwise if” under the “if” in the right white window using your mouse.

The new query is also integrated into the block in the program. You can now insert a second ultrasound query. If the distance to be measured is greater (>) 10, the motor should stop

This task would also be completed, and you can test the program.

Function model – Task 4 – Learning level 2

The task is as follows: After the program starts, the model should wait until a measurement by the ultrasound sensor falls below a set distance value – then the motor should start at full speed. If the value is exceeded, the motor will run more slowly.

This program is also integrated into an endless loop.

You can actually structure the program as above. The only change you need to make is, instead of “stop motor”, insert “set motor” once again and simply reduce the rotational speed.

You can test out this task now as well. The program is ended using the button you have used in the past.

Function model – Task 5 – Learning level 2

The task is as follows: After the program starts, the vehicle waits until the light barrier is interrupted. If this is the case, the encoder motor should run for a specific length of time. This is not defined using the “wait-s” command, but rather via a specific number of pulses measured at the controller counter input. If this specified number has been reached, the motor should stop. The counter is reset to 0.

The first part of the program is already provided under task 3a. Switch on the LED, wait time of 1 second and then the query of the light barrier.

Insert the “set motor ...” command from the “Motor” block. Ensure you use the encoder command.

Next, you must wait until the counter has reached the value of 200, for instance. First, insert the command “wait until” and dock the command “is counter xxx value=” from the Counter block. Change the value to 200.

Then, insert the command “Stop motor” from the “Motor block and the command “reset counter” under it from the “Counter” block.

This completes the final task for this model, and you can test out the program.