

## Forklift




You will use this model to learn some more programming fundamentals. You will probably be familiar with most of the fischertechnik components you will use from previous tasks.

Actuators, sensors, and technical accessories installed in the model.

Mini Motor	Gears	Button	Omniwheel	IR track sensor
				

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

The “forklift” model is divided into 2 programming tasks:

<b>Task 1</b>  Forklift_1.rpp	<u>Programming level 3</u> The forklift is positioned in front of the pallet. When the program is started, the forklift picks up the pallet, turns 180° and sets the pallet back down.
<b>Task 2</b>  Forklift_2.rpp  	<u>Programming level 3</u> The pallet is positioned at “A”. The forklift is positioned on the line in the centre, facing towards “A”. It will then travel along the line until it ends (at the end of the line, it will complete a short correction) and then pick up a pallet at “A”. Then it will turn around on the line, travel back along the line until the track ends at “B”, and set down the pallet at “B”. Then the program is complete.

## Task 1

You can use several subroutines from previous programming tasks to solve this first task.

You will need the subroutines “x2\_stop”, “x2\_fwd”, “x2\_bwd” and “x2\_rotR”.

Start the main program with the start command, followed by a “command element” from the “commands” element group.

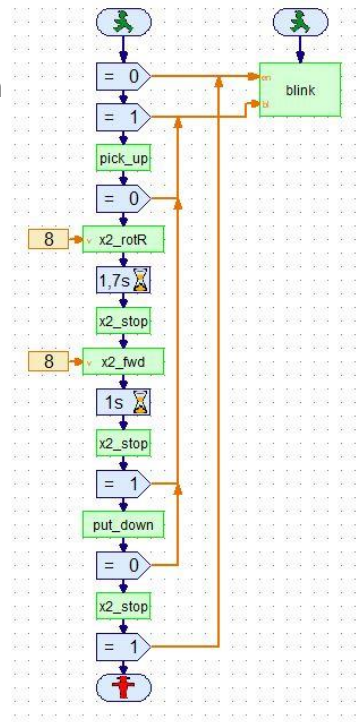
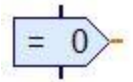
The overall program runs through one time. You will see the main program with associated subroutines on the right side.

For the flashing function, drag the command element underneath it with the value “1”.

As you can see, both command elements have another connection which will be used later to control the flashing function.

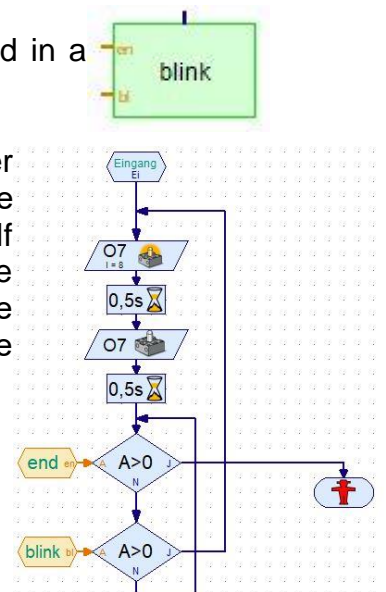
After the command element comes the subroutine to pick up the pallet, “pick\_up”.

First, create the subroutine for the flashing light. The flashing light should always be active when the forklift is moving. Assign it the name “flash”.



What should happen in this subroutine? The LED should be controlled in a flashing rhythm of 0.5s.

Once the first flashing rhythm is complete, the program queries whether the value at the input “End” is >0 (1). If this is the case (“J”), the subroutine continues to the stop command and ends the subroutine. If the value is not >0 (0), the second branch is queried for the input value “flash”. If the value here is >0 (“J”), the flashing process will start once again. If the value is not >0, it will continue at “end” and the subroutine will be ended.



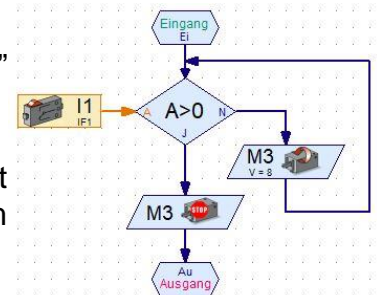
pick\_up

Add the subroutine "pick\_up" (pick up the pallet) to the program. First, all motors should be stopped. Do so using the subroutine "x2\_stop".

fork\_down

Then, you need to create another subroutine with the name "fork\_down" (fork down until limit switch I1 is reached).

You are already familiar with the individual commands, so they do not need to be explained here. You should also understand the program sequence without needing an explanation.



Insert the subroutine "x2\_fwd" as well, with a travel time (wait command) of "0.5s". You will need to experiment a little here - the fork should be under the pallet. Then the motors must be stopped using the command "x2\_stop".

The fork is now under the pallet and needs to lift it. To do so, use the subroutine "fork\_up1".

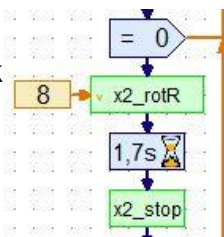
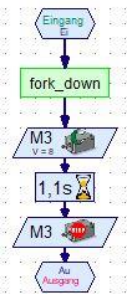
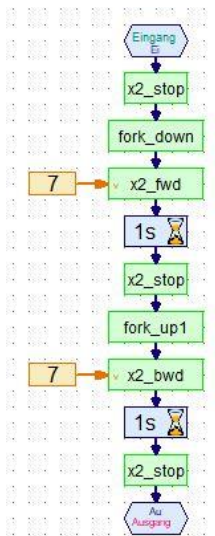
You can also experiment with the value for the wait command here.

You must complete the subroutine "pick\_up" as shown on the screen. Insert the subroutines "x2\_bwd" and "x2\_stop" accordingly.

Before inserting the subroutine "x2\_rotR", switch off the LED.

Then insert the subroutine "x2\_rotR". Rotation stops after 1.7s.

Then the forklift will travel forward for 1s. Then it will set the pallet back down. This is executed through the "put\_down" subroutine.



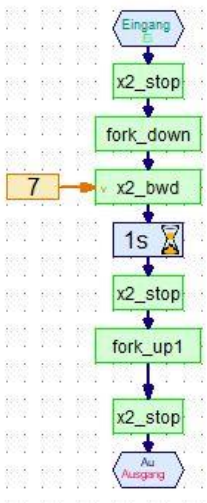
Create the subroutine using the image at the left, and insert it into the main program.

Remember to activate the flashing function before the subroutine "put\_down" and then switch it back off.

Test out the program and save it under the name

**"Forklift\_1.rpp"**

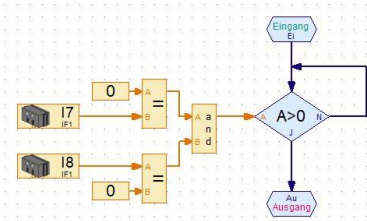
on your computer.



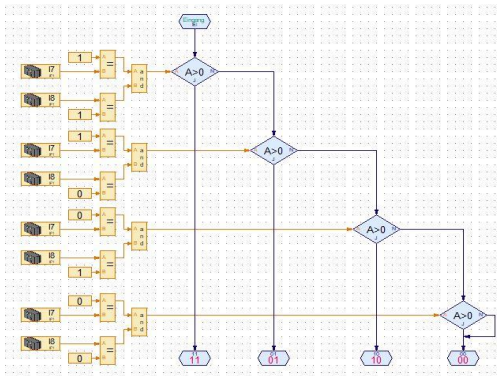
## Task 2

In the second task, the forklift picks up a pallet, travels along a certain obstacle course, then sets the pallet back down again at the end. The figure shows the main program. You will probably already be familiar with some of the subroutines.

Here is a brief explanation:



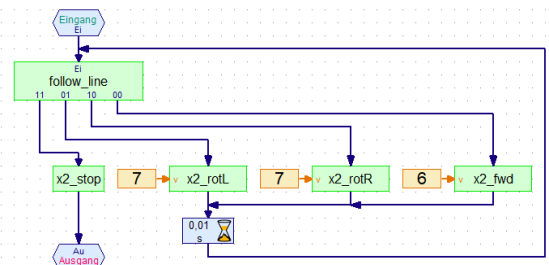
The subroutine “wait\_line\_detected” queries whether the track sensor is on the black line. If this is the case, the forklift will travel along the line until the track sensor leaves the black line (pick up pallet).



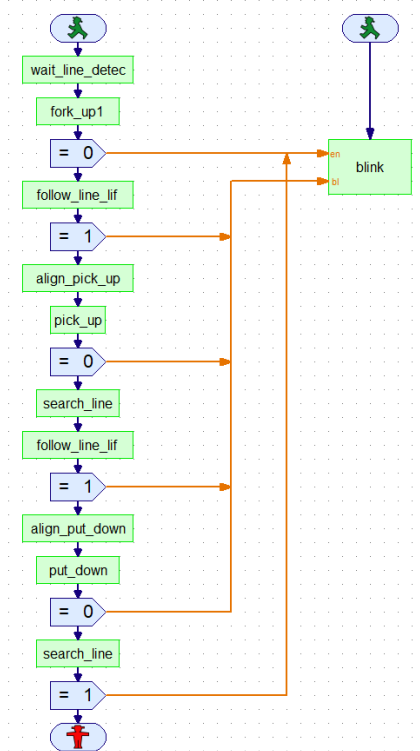
To follow a line, the system must query which IR sensor is not on the black line. There are 4 possibilities. Both sensors outside of black, both sensors on black, IR7 on black or IR8 on black. This query is completed in the subroutine “follow\_line”.

The determined values are sent to the subroutine “follower\_line\_lifter” and evaluated there.

For instance, if IR7 or IR8 is not on the black line, then the model must be aligned to the right or left through short rotational movements. The evaluation must ensure that the forklift stops at the end of the line. IR7 and IR8 have a value of “0”.



Once the forklift has reached the end of the line, it must align itself so that it can pick up and put down the pallet straight. The subroutines “align\_pick\_up” and “align\_put\_down” are designed for this purpose. The model will turn to the outermost edge of the black line, then in the opposite direction for a certain period of time, so that it is standing straight on the line. Design this program based on the example program, and add in the subroutines you are already familiar with.



At the end of this program, the forklift must turn by 180° and find the black line.

The subroutine “search\_line” is used to do so.

You are already familiar with the subroutines used here, so you can create the program.

OK - that was a complex programming task.

Test out the program and save it under the name

**“Forklift\_2.rpp”**

on your computer.

Complete examples of these two programs are also available in  
C:\Programme(x86)\ROBOPRO\Beispielprogramme\ROBOTICS Smarttech\

Disassemble the model and turn to the next task.

