

# Programming Calliope mini

## Content

[> Home](#)

Inhalt:

- [Transferring a program to the Calliope mini](#)
- [NEPO](#)
- [Selection Menu](#)
- [Sequence connection](#)
- [Parameter](#)
  - [Input - Parameter Input](#)
  - [Output - Return parameters](#)
- [Contextmenu](#)
- [Datatypes](#)
- [Block Categories \(Beginner, Expert\)](#)
- [Block descriptions](#)
- [Example Programs](#)

Programming with NEPO® follows a simple principle. The blocks are interconnected and executed by the robot according to their order. This principle is called »sequential operation«. Only blocks that are connected with each other are executed while running the program. The blocks that are connected together and to the start block are also called a sequence or »task«.

To connect blocks with each other, they have sequence connections and/or connectors. Depending on the mode of a block, the number of the connectors may vary.

Programming with NEPO is really easy. Every program begins with the »start« block.

This block is always added by default in the Open Roberta Lab workspace. The first block that you will execute is connected with the »sequence connector« of the »start« block. The sequence connector is a triangle at the bottom of the block and this triangle will become yellow as soon as a suitable block is in close proximity.



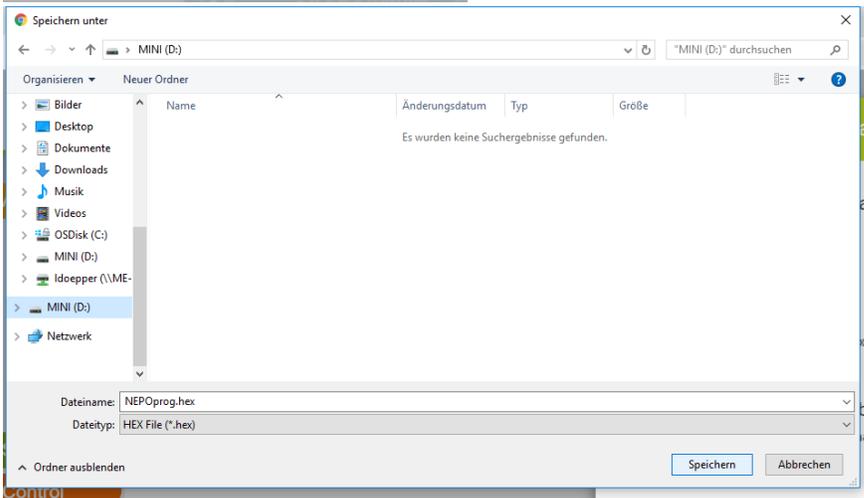
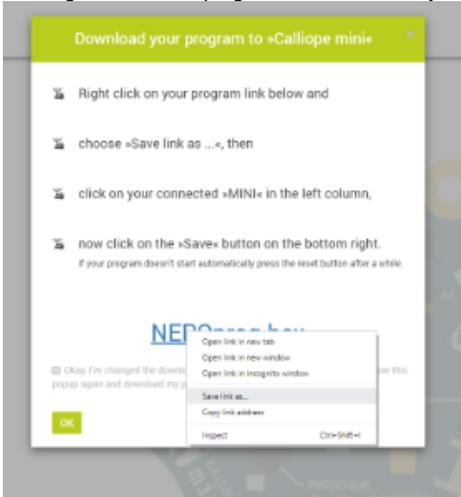
+ program-start  show sensor data

## Transferring a program to the Calliope mini

Once you have completed your first program and want to see the result, you must transfer your program to Calliope mini or Calli:bot. To do this, follow the steps below:

1. First press the  »Start on Calliope mini« button in the Open Roberta Lab.

- Then right-click on the program name. This always ends with ".hex".



- Press "Save as ..." and save your program on your Calliope mini. You will find it under the name "**Mini**".
- Now the yellow LED of your Calliope mini flashes very fast until the program is transferred. As soon as the program has been transferred, it will start immediately on your Calliope mini.
  - If you are using an older Calliope mini version 2016, you will need to press the "Reset" button on your Calliope mini once after the program has finished transferring to start the program.

## NEPO

NEPO is the name of our graphical programming language. It is easy and fun to learn programming. NEPO can also be used to program and interact with other robot systems.

Perhaps NEPO seems familiar to you. When designing NEPO we were inspired by existing concepts from [scratch](#) and [blockly](#).

The graphical NEPO blocks have different components and thus different attributes. A block always represents a certain functionality. Its function can easily be determined by its block category, for example »sensors«. Features of the graphical blocks are:

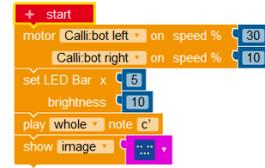
## Selection Menu

In the selection menu you can choose between different functions for the block, such as in block »drive« between forward and reverse.



# Sequence connection

The blocks are interconnected using the sequence connection. A program consists of all interconnected blocks. The blocks will be executed according to their order, from top to bottom. A NEPO block can have more than one sequence connection.



# Parameter

## Input - Parameter Input

Depending on the block different values can be passed to a block. The type of the value can be identified by the colored compounds (called »connectors« ) of a block. In the chapter »block category« you can see the different values a block can have. Blocks can only be connected when the colors of the input and the output connector match.

## Output - Return parameters

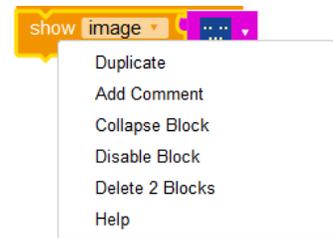
A block may optionally also pass a value (only one) to another block. These blocks have colored output connectors. Blocks can only be connected when the colors of the input and the output connector match.

# Contextmenu

By rightclicking a block its contextmenu can be opened.

Several actions can be performed, using the menu:

- Duplicate (copies the block and connected input parameters)
- Add comment (adds a comment to this block)
- Inline Inputs (changes the appearance of the block)
- Collapse Block (minimizes the block)
- Disable Block (deactivates the block so that it won't be executed)
- Delete x Blocks (deletes the selected and all connected blocks)

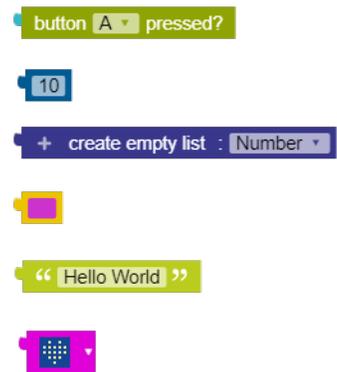


# Datatypes

There are seven different input and output types within NEPO. These types are as follows:

1. Logical value (light blue)
2. Number (dark blue)
3. String/Text (green)
4. Color (yellow)
5. List (purple)
6. Connection (pink)
7. Image (magenta)

 The type »List« can contain a Logical, Number, String or Color type



# Block Categories (Beginner, Expert)

A block-category groups blocks which have similar functionality. For example, the block-category "sensors" contains all blocks which control the sensors of your robot.

The Open Roberta Lab offers two choices when selecting existing blocks:

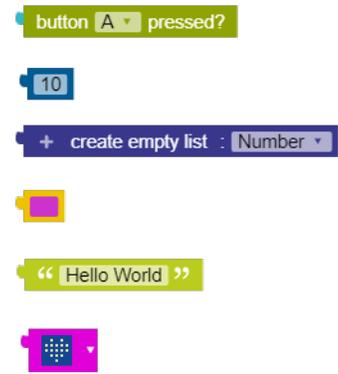
- Beginner
- Expert

The selection "Beginners" is the default selection. This category includes the main program blocks which are most important for programming a robot.

Choosing "Expert " includes the three additional block-categories "Lists", "Functions", and "Messages" and additional NEPO blocks in other block categories.

NEPO offers eleven block categories:

Block category	Beginner	Expert
Action	x	x
Sensor	x	x
Control	x	x
Logic	x	x
Math	x	x
Text	x	x
Lists		x
Colours	x	x
Variables	x	x
Functions		x
Messages		x



## Block descriptions

You can find the different blocks and their descriptions, ordered by categories [here](#).

For each block there is a detailed description of its functions, the different configurations, its parameters and return values.

## Example Programs

In the section »Do you know how ...« there are a number of different sample programs. There is a brief description and a »model solution« for each program. Each example program is also described with the following criteria:

**Criteria:**

- robot
- motors
- sensors
- block category
- difficulty
- note

The following programs are available:

- Driving a square
- Cornering
- Controlled program termination
- LED status display
- Own blocks, macros, methods
- Line follower

