



zenon
by COPA-DATA

zenon manual

zenon Science Package for Lego Mindstorms

v.8.10



© 2019 Ing. Punzenberger COPA-DATA GmbH

All rights reserved.

Distribution and/or reproduction of this document or parts thereof in any form are permitted solely with the written permission of the company COPA-DATA. Technical data is only used for product description and are not guaranteed qualities in the legal sense. Subject to change, technical or otherwise.

Contents

1	Welcome to COPA-DATA help.....	5
2	zenon Science Package for LEGO MINDSTORMS.....	5
3	zenon Science Package for Lego MINDSTORMS 2.0	7
3.1	Installation	7
3.1.1	zenon installation	7
3.1.2	Demo projects.....	8
3.1.3	Install firmware	8
3.1.4	Dealing with already existing firmware.....	10
3.1.5	Establish a Bluetooth connection with the Controller	12
3.2	Create a connection between zenon Logic and straton.....	13
3.2.1	Add NXT driver in zenon Logic	14
3.2.2	Connect NXT Controller	17
3.2.3	Controller basic settings	18
3.3	straton - Develop applications for NXT	20
3.3.1	Use NXT communication ports	21
3.3.2	Functions.....	29
3.4	Connect MINDSTORMS 2.0 Runtime with zenon	32
3.5	Questions and answers.....	34
3.5.1	Error message during the installation of the zenon Science Package.....	35
3.5.2	Activate original LEGO firmware.....	36
3.5.3	LEGO Firmware Loader does not find the Controller	37
3.5.4	zenon Logic Workbench driver does not communicate with the Controller.....	37
3.5.5	stratonNG driver does not communicate with the Controller.....	38
3.6	Use of complex variables	38
4	zenon Science Package for LEGO MINDSTORMS EV3.....	41
4.1	Requirements.....	41
4.2	Installation	42
4.3	Create a zenon Logic project.....	43
4.4	EV3 configuration in zenon Logic.....	45
4.4.1	Define sensor type	46
4.4.2	Create variables	47
4.4.3	Blocks and Functions function.....	48
4.5	Connect EV3 to zenon Logic Editor.....	58

4.6 First steps in the demo project	59
4.7 COPA-DATA Forum	64

1 Welcome to COPA-DATA help

ZENON VIDEO-TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel (https://www.copadata.com/tutorial_menu). The tutorials are grouped according to topics and give an initial insight into working with different zenon modules. All tutorials are available in English.

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com.

PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at support@copadata.com.

LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com.

2 zenon Science Package for LEGO MINDSTORMS

The **zenon Science Package for LEGO MINDSTORMS** enables the programming of LEGO MINDSTORMS NXT and MINDSTORMS EV3 Brick. Programs are created in zenon Logic and executed on the LEGO Controller with straton. zenon Logic is the programming environment in accordance with IEC 61131-3 included in zenon.

This instruction takes you through the installation and programming for the automation of a LEGO MINDSTORMS NXT 2.0 or MINDSTORMS EV3 Brick with zenon. Only information with regard to the operation with zenon Logic/straton is contained in this documentation. Please find basic information on the programming and operation of LEGO MINDSTORMS objects in the LEGO documentation.



All necessary software is included on the installation medium for the **zenon Science Package for LEGO MINDSTORMS**. This package does not contain any LEGO hardware, components or controls.

For the configuration of LEGO hardware with zenon use the following software:

- ▶ zenon: Software for monitoring and controlling which contains the integrated zenon Logic.
- ▶ zenon Logic Workbench: Programming environment for programs running on the embedded straton firmware.
- ▶ straton: Firmware for the LEGO Controller.

Please find all required applications and information on the installation media. These should already have been created in your zenon folder on the computer. The most important folders are:

- ▶ *FIRMWARE*: This folder contains the firmware you can load onto your LEGO MINDSTORMS Controller with the **LEGO Firmware Loader**.
- ▶ *USB_DRIVER*: This folder contains the INF file for the USB-to-serial driver.



Information

LEGO, MINDSTORMS, NXT and EV3 are trademarks and products of the company LEGO.

3 zenon Science Package for Lego MINDSTORMS 2.0

The **zenon Science Package for LEGO MINDSTORMS 2.0** enables the programming of LEGO MINDSTORMS NXT 2.0 objects. Programs are created in zenon Logic and executed on the LEGO Controller with straton. zenon Logic is the programming environment in accordance with IEC 61131-3 included in zenon.

3.1 Installation

For using the **zenon Science Package for LEGO MINDSTORMS** the following is required:

- ▶ zenon from version 7.00 SP0 on
- ▶ Firmware (on page 8) for the LEGO Controller

If you have already installed zenon 7.00 or higher on your computer you will also have a license for using the **zenon Science Package for LEGO MINDSTORMS**. If you haven't installed zenon yet install the zenon version from the installation medium (on page 7). This is a limited License (on page 7) for using LEGO MINDSTORMS 2.0.

3.1.1 zenon installation

In order to install zenon in the current version:

1. Connect the installation medium to the computer.
2. If the installation does not start automatically, select **start.exe**.
3. Select zenon 8.10.
4. Follow the instructions of the installation wizard.

Note: If you already have a zenon version from 7.00 on no installation is required.

3.1.1.1 zenon Science Package License

The **zenon Science Package for LEGO MINDSTORMS 2.0** is included in the standard zenon license. If this version is installed on a system without zenon it will run with a zenon Science Package License. With this license it is also possible to create and edit zenon projects within the scope of the licensed modules.

This license contains:

- ▶ 128 variables
- ▶ Extended Trend Starter Edition
- ▶ Historian Starter Edition
- ▶ VBA

- ▶ zenon Logic Workbench
- ▶ straton firmware

Note: This license is executable for 90 minutes. The Editor must then be closed and re-started.

3.1.2 Demo projects

You can also get two demo projects in the COPA-DATA Forum (www.copadata.com/forums) (<http://www.copadata.com/forums>):

- ▶ MINDSTORMS_DEMO: Gives an overview over sensors and controller. Use this project to become familiar with LEGO MINDSTORMS 2.0 and its programming.
- ▶ LEGO_DEMO_SHOOTERBOT: A sample project for the **Shooterbot**. This can be adjusted to your needs.

Load the working area with both projects into zenon:

1. right-click on the current workspace
2. in the context menu working space select -> **Restore backup...**
3. Go to the working area backup on the installation medium (*.wsb)
4. select the file and click on **OK**

To open projects:

1. in the working space go to node **zenon Logic (IEC 61131-3)**
2. select the node by left-clicking on it
3. select the project **mindstorms**
4. click on the symbol **open** in the toolbar or double-click on the project in order to open the project in the zenon Logic Workbench

3.1.3 Install firmware

In order to be able to use your LEGO MINDSTORMS 2.0 with zenon you have to load straton firmware on the Controller. This chapter will show you how to:

- ▶ straton - load firmware on the Controller
- ▶ update already existing (on page 10) straton firmware or replace other already existing firmware
- ▶ create a USB connection (on page 11) with the Controller with the COPA-DATA USB driver
- ▶ establish a Bluetooth connection (on page 12) with the Controller

STRATON - LOAD FIRMWARE ON THE CONTROLLER

In order to load straton firmware on the LEGO MINDSTORMS NXT 2.0 Controller the LEGO MINDSTORMS NXT 2.0 software is required, which was delivered with your LEGO MINDSTORMS NXT 2.0.



Attention

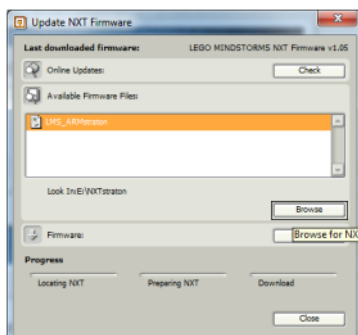
If on the Controller there already is a different firmware than the one delivered by LEGO please proceed as described in section **Dealing with already existing firmware** (on page 10).

In order to load the firmware:

1. Start the LEGO MINDSTORMS NXT 2.0 software.
2. In the **Tool** menu, select -> **Update NXT Firmware**.

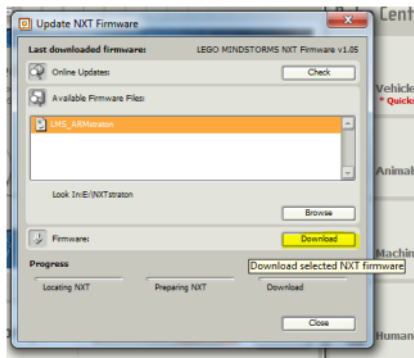


3. Ensure that the correct format is used: ***.rfw**.
If the file suffix is still ***.bin**, change this to ***.rfw**.
Correct file name: **STRATON_NXT_103_2011_08_10.rfw**.
You can find this on the installation medium in the path: *AdditionalSoftware\Science Package - Firmware\NXT2*.
4. Click on the **Browse** button.
5. Select the folder containing the firmware.



6. Start the NXT Controller.
To do so, press the orange key.
7. Connect the Controller via USB with the computer.

8. Click on the button **Download** to load the firmware on the Controller.



9. Remove the USB cable.
10. The Controller starts with the updated firmware



Information

In order to update the loaded straton later on or replace it again with firmware by LEGO follow the instructions in section **Dealing with already existing firmware** (on page 10).

3.1.4 Dealing with already existing firmware

If firmware not provided by LEGO is already present on the Controller this has to be set to the **Firmware-Load-Mode** in order to be able to install other firmware. This mode is also required if the straton firmware is supposed to be updated or LEGO firmware loaded again. In order to set the Controller to **Firmware-Load-Mode**:

1. Press for at least 5 seconds until you hear a quiet clicking noise
 - ▶ both the orange start key and
 - ▶ the Rest button on the back of the device.

You can best reach the Rest button with a pointed object.



2. Switch on the NXT Controller by pressing the orange key.



3. The NXT Controller starts in the **Firmware-Load-Mode**. You can hear a regular sound.
4. Start the download of the desired firmware.
To do so, a connection via USB (on page 11) or Bluetooth (on page 12) must be established. For the USB connection with zenon/straton the COPA-DATA driver is required.



Attention

The **Firmware-Load-Mode** can only be closed by downloading a firmware.

3.1.4.1 USB connection without LEGO firmware

If the firmware applied is not provided by LEGO probably the corresponding driver must be installed. In this case the USB Controller is not recognized during the connection with the computer. To establish a USB connection:

1. Open the **System control -> System -> Device manager**.
2. Double-click on **Unknown device**.
3. In the tab **Driver** click on **Update driver**.
4. The dialog for selecting the driver is opened.
5. Select **Search the computer for driver software**.
6. Select the folder containing the file **Mindstorm_straton.inf**.
7. A warning regarding the driver manufacturer is displayed.
8. Select **Install driver nevertheless**.
9. The device can now communicate via USB.

3.1.5 Establish a Bluetooth connection with the Controller

If the computer is provided with a Bluetooth port the connection with the Controller can also be established via Bluetooth. In order to configure Bluetooth for the connection between computer and Controller:

1. Switch on the LEGO Controller by pressing the orange key.



2. Press the orange key again.
3. Select *Settings* with the right gray arrow key.



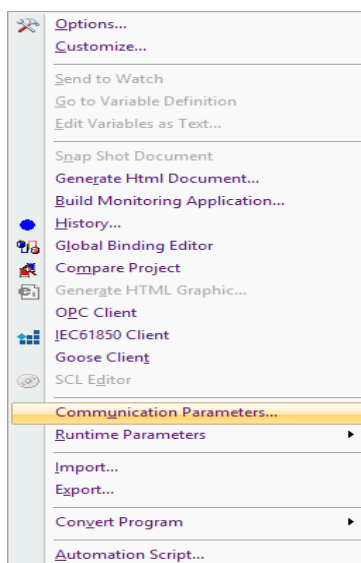
4. Confirm selection by pressing the orange key.
5. Go to the menu for Bluetooth settings by using the gray keys.
 - a) Activate Bluetooth.
 - b) Assign a Bluetooth name.
 - c) Set the Bluetooth status to *visible*.
 - d) Leave the Bluetooth menu by clicking on the dark gray key.
6. Change to menu *Connections*.
7. Select *Bluetooth* using the gray keys.
8. Now the Controller can be found via Bluetooth.
9. Open the Bluetooth device manager on the computer.
 - a) Select **Add device**.
 - b) Windows will search for visible Bluetooth devices.
 - c) Select your MINDSTORMS NXT Controller from the list and click on **Next**.
 - d) The MINDSTORMS Controller requires to enter a password for establishing the connection:
1234_ is suggested. Confirm the password by pressing the orange keys.
 - e) The password defined in the MINDSTORMS 2.0 Controller must also be entered on the computer.
 - f) Click on **Next**.
10. The connection is being established and the drivers are being installed.

11. Double-click on the created Bluetooth device.
The installed device and the assigned COM port are displayed. Connect the zenon Logic Workbench with this COM port or connect zenon with the **stratonNG** driver.
In order to adjust the port:
 - a) Click on properties.
 - b) Click on the **Advanced** button.
 - c) Change the port number according to your specifications.

3.2 Create a connection between zenon Logic and straton

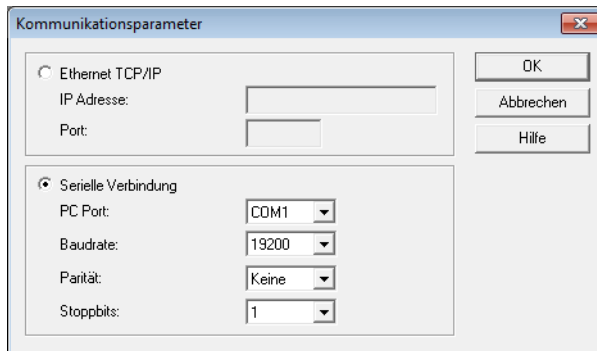
In order to create a connection between zenon Logic and straton:

1. Start zenon. The project **MINDSTORMS** is opened.
If you would like to open a different project or workspace open the workspace for the project via the context menu for the workspace.
If you would like to create a new project:
 - a) select the file -> Project new
 - a) create a new zenon Logic project by first right-clicking on the node zenon Logic and then selecting in the project tree **zenon Logic Project new...**
2. Open the zenon Logic project by double-clicking in the Workbench.
3. In the zenon Logic Workbench click on **Tools -> communication parameter**.



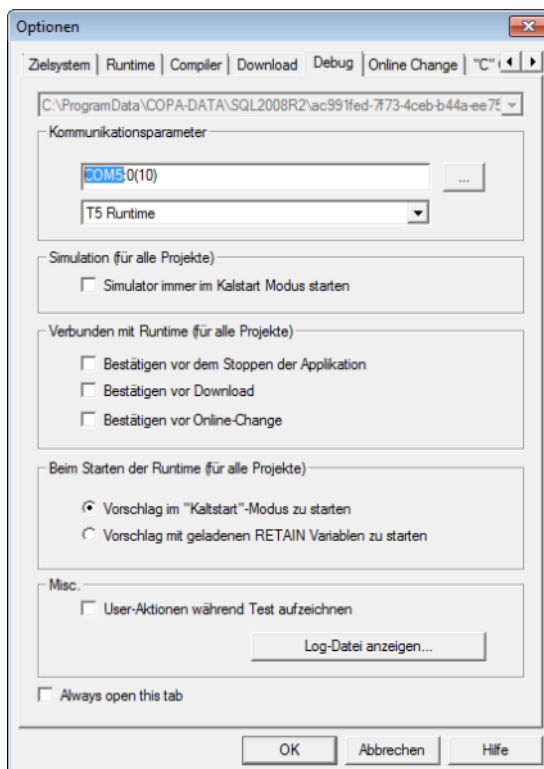
4. The dialog **Options** with the tab **Debug** is opened in order to configure the connection.

- Click on the button ... in order to open the dialog for the settings of the serial communication.



- Configure the serial setting corresponding to the settings for the LEGO Controller.

Note: In this dialog only COM ports 1 to 4 can be selected. However, in the dialog **Options** tab **Debug** you can enter the COM port directly in to the field **communication parameter**.



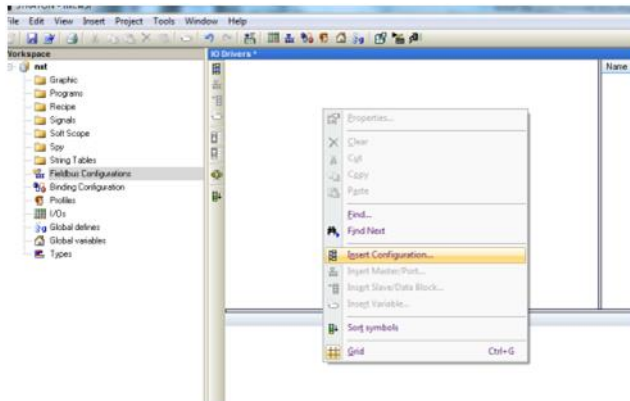
- You will thus have configured the connection between zenon Logic and straton.

3.2.1 Add NXT driver in zenon Logic

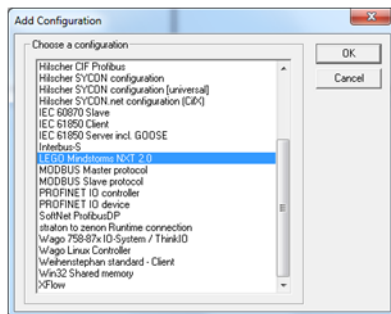
In order to add the NXT driver in zenon Logic:

- In the project tree in the zenon Logic Workbench double-click on **Fieldbus configuration**.

2. In the window I/O driver click on the symbol or the command in the context menu for **Add configuration**.



3. The dialog for configuration is opened.



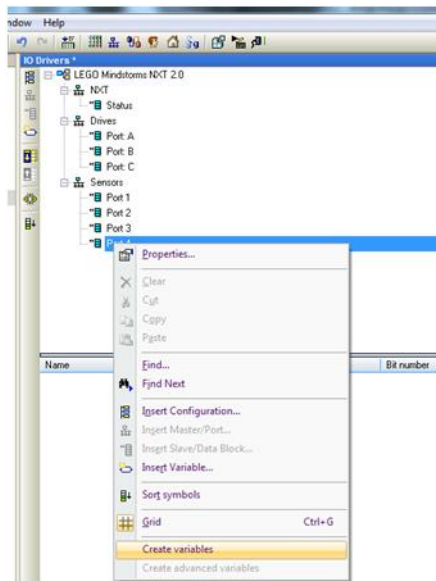
4. Select the LEGO MINDSTORMS NXT 2.0.
5. There are three sections:
 - ▶ NXT (Status)
 - ▶ Drives
 - ▶ Sensors

Clicking on the plus (+) before the section opens a subarea. For each of these sections predefined variables can easily be created.

CREATE VARIABLES

Create the required variables:

1. Right-click on the **NXT**, **Drives** or **Sensors** branch.



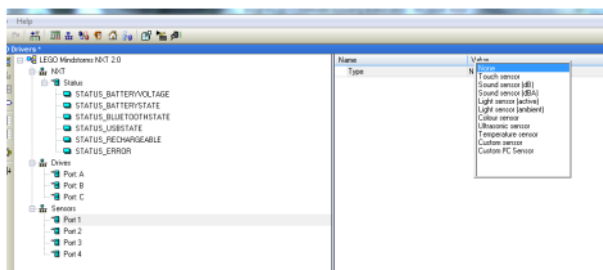
2. Select **Create variable** in the context menu

DRIVES

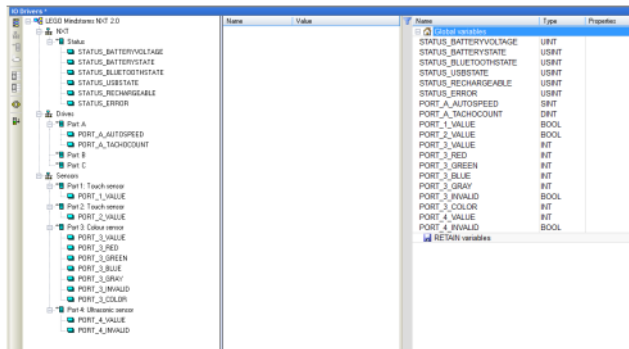
There are two versions: **easy** and **advanced**. As a start it is best to select **easy** and then **Create variable**. Please find more information on the use of *Advanced Variables* in section Use of complex variables (on page 38).

SENSORS

Before creating a variable in section **Sensors** a sensor type in field **Type** must be selected. To do so, double-click on the field **Value**.



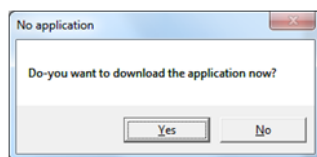
CONFIGURATION EXAMPLE:



3.2.2 Connect NXT Controller

In order to connect the NXT Controller:

1. On the Controller select **USB** as connection medium:
 - a) Press the orange key.
 - a) Use the gray keys to go to the menu *Settings* and open it by pressing the orange keys.
 - b) Select *Connection*.
 - c) Select **USB**.
(If you would like to connect via Bluetooth select **Bluetooth**.)
2. Compile the zenon Logic project:
 - a) Right-click on **mindstorms** in the project tree.
 - b) Select **Compile project** in the context menu.
3. Go online:
 - a) Select the entry **Online** in the zenon Logic menu **Project**.
 - b) The request for download is opened:



- c) Click on **Yes**.

- b) **Don't save:** The application is saved without saving data or configuration on the flash memory.
- c) **Cancel:** Saving is cancelled and the main menu opened.

During the next start the data saved last (straton application, retain data etc.) are loaded.

STRATON MENU

With the straton menu you can start or stop the loaded straton application. In order to open the menu select the entry **straton** in the main menu.

STATISTICS

Statistics displays information on the currently running application. In order to display the corresponding data select the entry *Statistics* in the main menu. The following is being displayed:

- ▶ the name of the loaded application
- ▶ code size in kB
- ▶ database size in kB
- ▶ the execution time of the straton cycle
- ▶ the cycle latency in microseconds

MENU SETTINGS

In this menu the following is configured and displayed:

- ▶ *Versions:* Information on the version used.
- ▶ **Bluetooth:** Bluetooth activation or deactivation. Name and contacts can be managed.
 - ▶ *Name:* Enter a name for the Bluetooth device. The name should be unique in the Bluetooth network.
 - ▶ *On/Off:* Activates/deactivates Bluetooth at the Controller.
 - ▶ *Visibility:* Determines if the device is visible for other Bluetooth devices.
 - ▶ *Search:* Starts search for Bluetooth devices. In order to add a device to the Bluetooth contacts select it from the list of devices found.
 - ▶ *Contacts:* Manage contacts. You can remove contacts (devices which may be connected) or establish connections (e.g. for pairing).

CONNECTION

Select the connection to be used by the straton communication server. Connection for the communication with the zenon Logic Workbench. The communication server also includes a MODBUS

slave. The zenon driver for straton must also be connected with the communication server. Possible options are:

- ▶ **USB**
- ▶ **Bluetooth:** incoming Bluetooth connection
- ▶ **RS-485:** RS-485 port with the parameters *115200, E, 8, 1*
Attention: Using RS-485 deactivates the sensor port 4. Please also bear in mind that the straton communication server protocol can only be used point-to-point.
- ▶ *None:* The communication server is not used.

VOLUME

Defines the volume of the built-in speakers. Valid values are 0 to 4, with:

- ▶ 0: switched off
- ▶ 4: Maximum

3.3 straton - Develop applications for NXT

The following restrictions apply for straton applications on LEGO MINDSTORMS 2.0:

STRATON RUNTIME:

- ▶ Maximum size of code: 26 kB
- ▶ Maximum size of straton database: 20 kB

Note: Size of code and database can be checked in the menu **Statistics**.

ZENON LOGIC WORKBENCH

- ▶ Only COM ports 1 - 9 can be used. If a USB device or a Bluetooth COM port was assigned a higher port number than 9 this has to be manually configured to a port number lower than 10.
- ▶ In the Workbench only ports COM1 – COM4 can be selected. However, port numbers can be entered directly into the parameter string, e.g. **COM9:19200,N,8,1**. (See also section Establish connection between zenon Logic and straton (on page 13).)

Note: In case of a USB or Bluetooth connection it is sufficient to enter the port, e.g. **COM5**



Information

Please find general information on the programming of LEGO MINDSTORMS NXT 2.0 in the LEGO documentation.

3.3.1 Use NXT communication ports

NXT objects contain three communication ports which can be addressed by the straton application.

With the menu **Settings** (on page 18) one of these connections is selected for the communication server. This connection can subsequently not be used out of the straton application (e.g. by a MODBUS Master or a function block). However, if a MODBUS Slave configuration was added a MODBUS Slave can use the configured port.

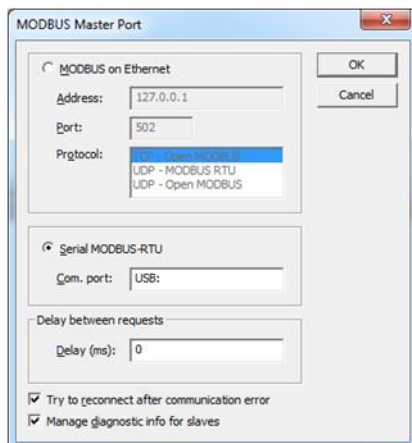
Please find further information on the use of the serial port in the straton program and on the MODBUS I/O driver in the zenon Logic documentation. To do so, in the Workbench press **F1** or use the menu **Help**.

All ports which are not being used by the communication server can be used by the straton application and are addressed as follows:

3.3.1.1 USB

The NXT component can be connected with a computer via the USB port. NXT is recognized as serial connection by the computer.

The USB port can be addressed by the straton application by selecting USB as COM port.



3.3.1.2 Bluetooth

The Bluetooth port can be used in two different ways:

1. For incoming connections:
in this case the connection is established from a remote device, e.g. a computer. In this case the string **BT** is sufficient in order to use the Bluetooth port.

2. For outgoing connections:

NXT is configured in order to establish a connection with another Bluetooth device, for instance another NXT object. In this case use a string beginning with BT and ending with the name of the device, for instance: **BT:NXT**.

Note: The device to be addressed must already be contained in the Bluetooth contacts. In order to manually add a device to the contacts start the Search (on page 18) and manually establish the connection once. For the first contact a pairing is probably required.

3.3.1.3 RS-485 Port (serial)

NXT objects also contain a serial RS-485 port. It is however shared with sensor 4. Thus only the port or the sensor can be used.

The RS-485 port is addressed via character string **HS**. With this the port is opened with the help of the following parameter per default:

- ▶ **Baud rate:** 115200 bit/s
- ▶ **Parity:** *even*
- ▶ **Data bits:** 8
- ▶ **Stop bit:** 1

ADAPT PARAMETER

If you want to adapt the communication parameters, add them after the colon separated by commas. The order:

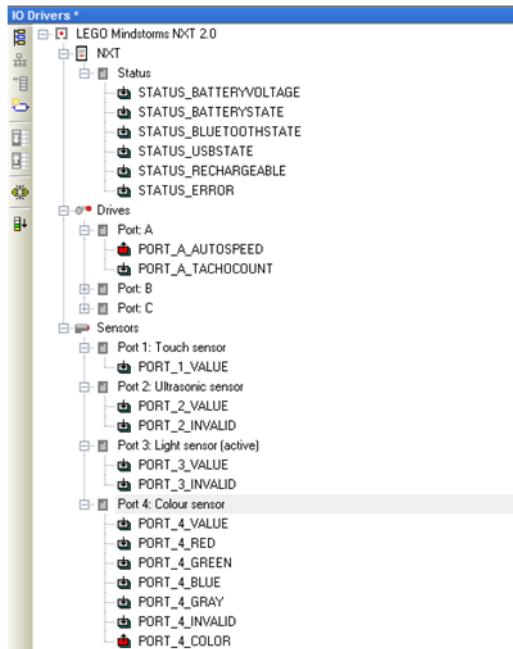
- ▶ **Baud rate** in bits/s
- ▶ **Parity:** *N* (none), *E* (even), *O* (odd), *M* (mark) and *S* (space)
- ▶ **Data bits**
- ▶ **Stop bit**

For example: **HS:19200,N,8,1**

Note: If you use the RS-485 port with MODBUS or the straton protocol, always set **Data Bits** to 8.

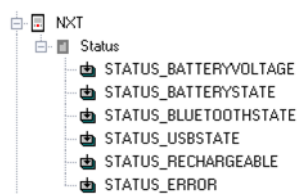
3.3.1.4 Fieldbus

With straton NXT it is also possible to use the MODBUS Slave and MODBUS Master configuration. Please find information on MODBUS in the zenon Logic documentation. To do so, in the Workbench press **F1** or use the menu **Help**. The use of NXT sensor and motor ports are configured in the LEGO LEGO MINDSTORMS NXT 2.0 configuration.



3.3.1.4.1 Status variables

The following status variables are available:



- ▶ STATUS_BATTERYVOLTAGE: current battery charge in millivolt.
- ▶ STATUS_BATTERYSTATE: Battery capacity.
Value: 0 to 4 (as displayed on the NXT display)
- ▶ STATUS_BLUETOOTHSTATE: Bluetooth status
Values:
 - ▶ 0: on
 - ▶ 1: visible
 - ▶ 2: connected

- ▶ 3: connected and visible
- ▶ 4: off
- ▶ STATUS_USBSTATE: USB status
Values:
 - ▶ 0: disconnected
 - ▶ 1: connected
 - ▶ 2: operating
- ▶ STATUS_RECHARGEABLE: Status for rechargeable batteries
Values:
 - ▶ 0: No rechargeable batteries existing
 - ▶ 1: rechargeable batteries existing
- ▶ STATUS_ERROR: Error code

3.3.1.4.2 Drive Variables

The following variables are available for the drive:



- ▶ PORT_x_AUTOSPEED (SINT, Output): Defines the motor speed.
Values: from *-100* to *100*
 - ▶ 0: Motor off
 - ▶ *-100*: Maximum speed backward
 - ▶ *100*: Maximum speed forward
- ▶ PORT_x_TACHOCOUNT (DINT, Input): Contains absolute number of motor revolutions in angular degrees. 360° are one complete revolution per second. Valid values: from *-2,147,483,648* to *2,147,483,647*.

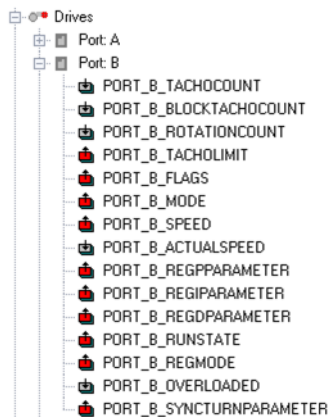
COMPLEX VARIABLES

More complex drive operations are configured with more complex variables. Select them in the entry **Create advanced variables** in the context menu.



Attention

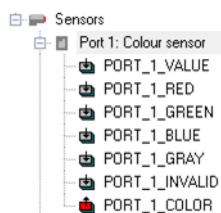
Do not use the AUTOSPEED variable and the Advanced Variables at the same time. Using them at the same time may result in an undefined behavior of the drive.



Please find a description on how to configure these variables in section Using complex variables (on page 38).

3.3.1.4.3 Color sensor variables

The following variables are available for the color recognition:



- ▶ PORT_x_COLOR (INT, Output): Defines the sensor mode.

- ▶ 0: Passive mode (measure brightness)
- ▶ 1: Red is being measured.
- ▶ 2: Green is being measured.
- ▶ 3: Blue is being measured.
- ▶ 4: All colors are being measured

- ▶ PORT_x_VALUE (INT, Input)

If the mode is *passive*, *red*, *green* or *blue*:

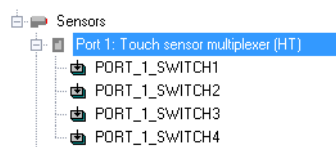
- ▶ A value between 0 and 255 defines the color intensity.

If the mode is *all colors*:

- ▶ 1: black
- ▶ 2: blue
- ▶ 3: green
- ▶ 4: yellow
- ▶ 5: red
- ▶ 6: white
- ▶ PORT_x_RED: In the mode *all colors* the intensity of the color red is measured.
- ▶ PORT_x_GREEN: In the mode *all colors* the intensity of the color green is measured.
- ▶ PORT_x_BLUE: In the mode *all colors* the intensity of the color blue is measured.
- ▶ PORT_x_GRAY: Brightness is measured

3.3.1.4.4 Touch Sensor Multiplexer (HT)

The following variables are available for the touch sensors:



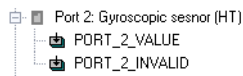
When using the Hitechnic Touch Sensor Multiplexer up to four sensors can be connected with a NXT port:

- ▶ PORT_x_SWITCH1: Status of sensor 1
- ▶ PORT_x_SWITCH2: Status of sensor 2
- ▶ PORT_x_SWITCH3: Status of sensor 3
- ▶ PORT_x_SWITCH4: Status of sensor 4

3.3.1.4.5 Gyroscopic sensor (HT)

The LEGO MINDSTORMS 2.0 NXT Gyro Sensor gives information on the number of motor revolutions and the direction of the revolutions. The revolutions are measured in angular degrees. 360° are one complete revolution per second.

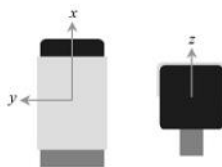
After the start of the straton application the sensor is automatically calibrated during connection. To do so, the current zero value is calculated. This takes about 200 ms. During this period of time the sensor must not be moved.



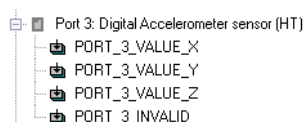
- ▶ PORT_x_VALUE: Number of angular grades per rotation second
- ▶ PORT_x_INVALID: Is *TRUE* if:
 - ▶ the sensor is being calibrated
 - ▶ the value cannot be read

3.3.1.4.6 Digital Acceleration Sensor (HT)

The LEGO MINDSTORMS 2.0 NXT acceleration sensor includes a 3-axis accelerometer measuring acceleration in the three axes **x**, **y** and **z**. The acceleration is displayed within the scope of -2g to +2g with approximately 200 steps per g.



The acceleration sensor can also be used to measure the gradient in three axes. This is possible because gravitation is perceived as acceleration. If the sensor is motionless in the normal horizontal position the x-axis and y-axis, being horizontal, will be close to zero, while the z-axis' value will be close to 200, corresponding to 1g. If the sensor is then inclined the other axes will also perceive gravity and the z-axis' value will decrease. Since gravity is divided among the three part vectors the gradient of the sensor can be determined.



- ▶ PORT_x_VALUE_X: Acceleration in direction of the x-axis
- ▶ PORT_x_VALUE_Y: Acceleration in direction of the y-axis
- ▶ PORT_x_VALUE_Z: Acceleration in direction of the z-axis
- ▶ PORT_x_INVALID: Set to *TRUE* if the data of the sensor cannot be read.

3.3.1.4.7 Digital Compass Sensor (HT)

The LEGO MINDSTORMS 2.0 NXT Compass Sensor contains a magnetic digital compass to measure the earth's magnetic field and calculate a heading angle. The Compass Sensor can be connected to a NXT port with a NXT standard cable and uses the digital I2C communication protocol. The current heading is calculated to the next integral angular degree.

The HiTechnic Compass Sensor is influenced by local magnetic interferences, as is any other magnetic compass, too. Metal objects such as motors, batteries or cables may cause magnetic interferences. Local magnetic interferences may cause the compass to produce a heading deviating several degrees from the current magnetic heading. This effect is called compass deviation. In order to correct this deviation the HiTechnic Compass Sensor has a built-in calibration function correcting deviations and saving these values in the compass.

The calibration is optional and usually not required during normal operation. In order to minimize the need to calibrate install the compass in a distance of at least 10 to 15 cm (4 - 6 inches) to NXT and the NXT motors.

In order to calibrate the compass in the program:

- ▶ Select **calibration mode**.
- ▶ Program your NXT object so that it rotates within a very narrow circle.
- ▶ It must rotate between 1½ and 2 times (more than 360 degrees).
- ▶ It must complete a full turn in approximately 20 seconds.

The compass will maintain the calibration settings until another calibration is made, even though it is being unplugged.

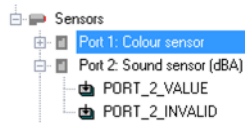


- ▶ PORT_x_VALUE: Current heading in degrees.
0: North
- ▶ PORT_x_INVALID: Set to *TRUE* if the data of the sensor cannot be read.
- ▶ PORT_x_CAL: (Display) Controls the calibration mode.
 - ▶ *TRUE*: change to calibration mode
 - ▶ *FALSE*: change to measuring mode.

Note: In the calibration mode the sensor provides invalid data.

3.3.1.4.8 Other sensors

The following additional variables for sensors are available:

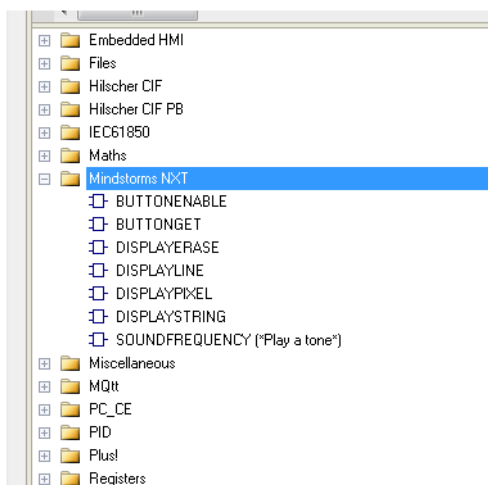


- ▶ *PORT_X_VALUE*: sensor value, depends on sensor
 - ▶ Sound dB
 - ▶ Sound dBA
 - ▶ Ultrasonic: distance in centimeters (cm)
 - ▶ Temperature: temperature in Celsius centigrade (°C/10)
- ▶ *PORT_X_INVALID*:
 - ▶ *TRUE*: sensor is in error status

3.3.2 Functions

In order to select functions these must be located in the corresponding folder. Copy the *Mindstorms NXT* folder from the *HWDEF* folder on the installation medium into the *HWDEF* folder of your straton installation.

For example: %ProgramData%\COPA-DATA\zenon8.10711\straton\HWDEF\Mindstorms NXT



The following functions are available:

- ▶ **BUTTONENABLE** (on page 30): straton takes over control of NXT keys.
- ▶ **BUTTONGET** (on page 30): Transmits information on which key was pressed.

- ▶ **DISPLAYERASE** (on page 31): Erases screen content.
- ▶ **DISPLAYLINE** (on page 31): Draws a line.
- ▶ **DISPLAYPIXEL** (on page 31): Draws a pixel.
- ▶ **DISPLAYSTRING** (on page 31): Writes a string.
- ▶ **SOUNDFREQUENCY** (on page 32): Sound output.

3.3.2.1 BUTTONENABLE

BOOL BUTTONENABLE (BOOL ENABLE)

During the activation with *ENABLE* and status *TRUE* the control is being passed on to straton using the keys on the NXT object. The key status can be read with *BUTTONGET* (on page 30).

Note:

- ▶ The menu **System** on the NXT object is deactivated in this case.
- ▶ Activating this function is only possible if the menu on the NXT object is inactive.
- ▶ This status can be interrupted, for instance by a Bluetooth pairing dialog. In this case *BUTTONGET* (on page 30) returns the value *-1*.

Response value:

- ▶ *TRUE*: Activity successfully completed.

3.3.2.2 BUTTONGET

SINT BUTTONGET

Return values:

- ▶ *-1*: Keys cannot be read by straton. The menu **System** on the NXT object is activated.
- ▶ *0*: no keys pressed.
- ▶ *1*: left key pressed.
- ▶ *2*: **Entry key** (orange key) pressed.
- ▶ *3*: right key pressed.
- ▶ *4*: **Leave** (dark gray key) pressed.

3.3.2.3 DISPLAYERASE

BOOL DISPLAYERASE

Erases the screen contents.

Response value:

- ▶ *TRUE*: Activity successfully completed.

3.3.2.4 DISPLAYLINE

BOOL DISPLAYLINE (USINT X1,USINT Y1,USINT X2, USINT Y2, BOOL C)

Draws a line from point **X1,Y1** to point **X2,Y2**.

Values for C:

- ▶ *TRUE*: draws a line
- ▶ *FALSE*: deletes the line

Response value:

- ▶ *TRUE*: Activity successfully completed.

3.3.2.5 DISPLAYPIXEL

BOOL DISPLAYPIXEL (USINT X, USINT Y, BOOL C)

Draws a point with the coordinates **X,Y**.

Values for C:

- ▶ *TRUE*: draws a point
- ▶ *FALSE*: deletes the point

Response value:

- ▶ *TRUE*: Activity successfully completed.

3.3.2.6 DISPLAYSTRING

BOOL DISPLAYSTRING(USINT X, USINT Y, STRING TEXT)

Writes a string beginning with the indicated XY position.

Response value:

- ▶ *TRUE*: Activity successfully completed.

3.3.2.7 SOUNDFREQUENCY

BOOL SOUNDFREQUENCY(UDINT FREQ, UDINT DUR, USINT VOL)

Plays a tone with the indicated frequency in Hz during the defined length in milliseconds. The volume (VOL) is indicated in a range of 1 to 4. It is limited by the configured maximum volume (on page 18) in the menu **Volume**.

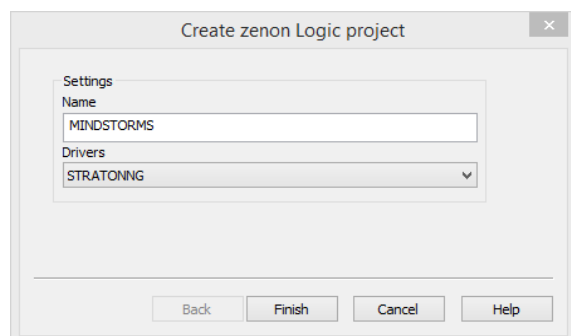
Response value:

- ▶ *TRUE*: Activity successfully completed.

3.4 Connect MINDSTORMS 2.0 Runtime with zenon

For creating a new zenon Logic project in zenon it is recommended to use the **stratonNG** driver. This supports serial communication.

When creating a new zenon Logic project in the dialog define the name of the project and the driver used.



Select *STRATONNG* from the drop-down list.

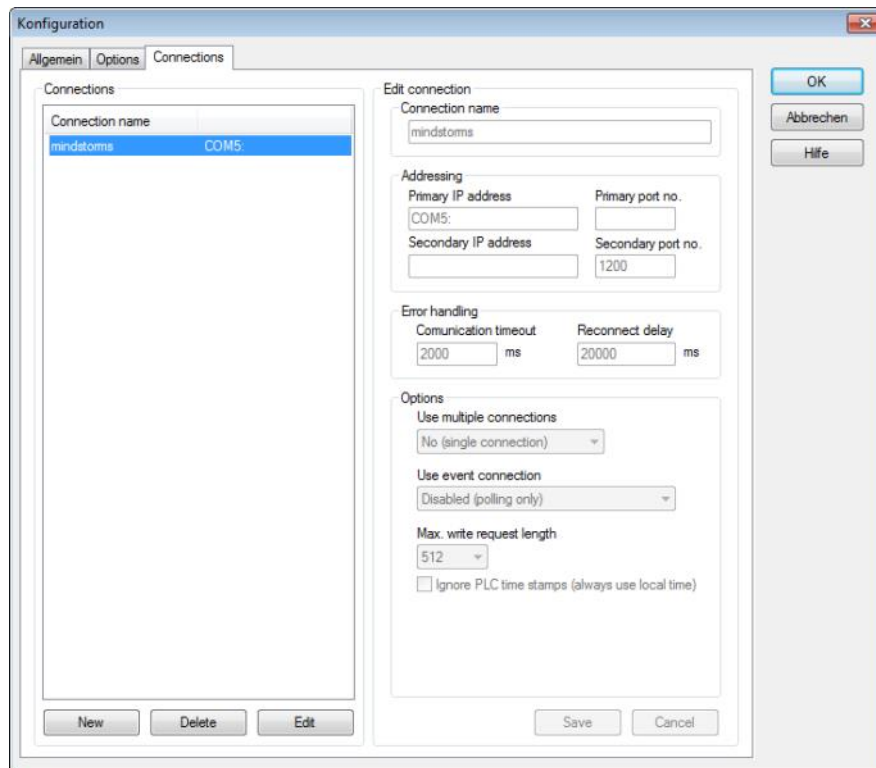
Note: The selection of the driver cannot subsequently be changed again.

CONFIGURATION OF STRATONNG DRIVER

To configure the driver:

1. in the zenon project tree select the node **Variables -> Driver**

2. highlight the stratonNG driver
Note: this is displayed with the name defined for the project, e.g. **straton: mindstorms**
3. select **Driver configuration** in the context menu
4. the dialog for the driver configuration is opened
5. open the tab Connections



To edit the driver:

1. highlight the connection name
2. click on **Edit**
3. configure the link
4. click on **Save** and then **OK**

Configuration of the driver for the connection within the scope of LEGO MINDSTORMS 2.0:

Option	Description
Connection name	Must correspond to the project name. If the name is different from that of the project this will result in errors during the connection with the Controller.
Addressing	In the communication with straton NXT via USB or Bluetooth it is sufficient to set the port without further settings, e.g. COM5 :

Option	Description
Primary IP address	<p>Entering COM port.</p> <p>Entries beginning with COM and ending with a colon are interpreted as COM ports.</p> <p>Attention: The colon (:) must be present at the end of the port definition. Otherwise it is not possible to establish a connection.</p> <p>If no further parameters are indicated the standard configuration of the COM port is used:</p> <ul style="list-style-type: none"> ▶ Baud rate: 115200 bit/s ▶ Parity: even ▶ Data bits: 8 ▶ Stop bit: 1 <p>If you want to adapt the communication parameters, add them after the colon separated by commas. The order:</p> <ul style="list-style-type: none"> ▶ Baud rate in bits/s ▶ Parity: N (none), E (even), O (odd), M (mark) and S (space) ▶ Data bits ▶ Stop bit <p>For example: COM5:19200,N,8,1</p> <p>Limitations: If a COM port is indicated it is not possible to use:</p> <ul style="list-style-type: none"> ▶ Event connection ▶ Multiple connections (more than one connection with a driver)

The documentation of these settings refers to the communication in the scope of LEGO MINDSTORMS 2.0. Please find details on the general configuration of the **stratonNG** driver in the online help (F1) in chapter **stratonNG** (<stratonNG.chm::/27853.htm>).

3.5 Questions and answers

The most common questions regarding installation and operation of LEGO MINDSTORMS with straton and the **zenon Science Package**.

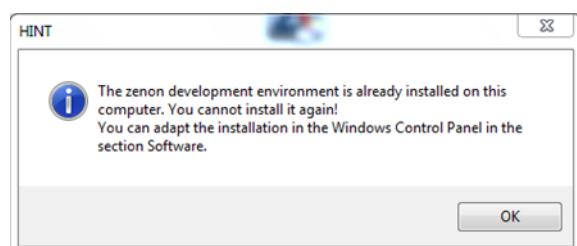
Question	Answer
Why is an error message displayed during installation?	Probably there is already a zenon installation present in the system.

Question	Answer
	<p>Use the existing installation.</p> <p>Details: Error message during the installation of the zenon Science Package (on page 35).</p>
How do I change back to the original LEGO firmware?	<p>Changing back corresponds to the proceeding during the installation of the straton firmware if other firmware already exists.</p> <p>Details: Activate original LEGO firmware (on page 36).</p>
How can I proceed if the LEGO Firmware Loader does not find the Controller?	<p>Change the USB port or re-install its driver.</p> <p>Details: LEGO Firmware Loader does not find the Controller (on page 37).</p>
How do I proceed if the zenon Logic Workbench driver does not communicate with my Controller?	<p>This may have several causes. Check the settings and if all devices are online.</p> <p>Details: zenon Logic Workbench driver does not communicate with the Controller (on page 37)</p>
How do I proceed if the stratonNG driver does not communicate with my Controller?	<p>This may have several causes. Check the communication parameters, the Runtime and the project.</p> <p>Details: stratonNG driver does not communicate with the Controller (on page 38).</p>

Further questions and answers can also be found in the COPA-DATA user forum:
www.copadata.com/forums/ (<http://www.copadata.com/forums/>).

3.5.1 Error message during the installation of the zenon Science Package

An error message is displayed during the installation of the **zenon Science Package for LEGO MINDSTORMS** which refers to an already existing installation.



This zenon version has already been installed on your system. You don't have to install zenon and can use the existing version. Import the demo projects (on page 8) from the installation medium and continue with Install firmware (on page 8).

3.5.2 Activate original LEGO firmware

If firmware not provided by LEGO is already present on the Controller this has to be set to the **Firmware-Load-Mode** in order to be able to install other firmware. This mode is also required if the straton firmware is supposed to be updated or LEGO firmware loaded again. In order to set the Controller to **Firmware-Load-Mode**:

1. Press for at least 5 seconds until you hear a quiet clicking noise
 - ▶ both the orange start key and
 - ▶ the Rest button on the back of the device.You can best reach the Rest button with a pointed object.



2. Switch on the NXT Controller by pressing the orange key.



3. The NXT Controller starts in the **Firmware-Load-Mode**. You can hear a regular sound.
4. Start the download of the desired firmware.
To do so, a connection via USB (on page 11) or Bluetooth (on page 12) must be established. For the USB connection with zenon/straton the COPA-DATA driver is required.



Attention

The **Firmware-Load-Mode** can only be closed by downloading a firmware.

DOWNLOAD OF THE LEGO FIRMWARE:

1. Connect the Controller via USB with the computer.
2. Start the LEGO MINDSTORMS NXT 2.0 software.

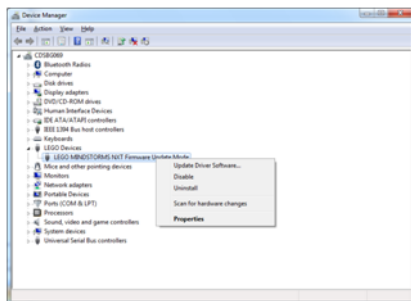
3. In the **Tools** menu, select **Update NXT Firmware**.
4. The original LEGO MINDSTORMS NXT firmware is preselected.
5. Click on the **Download** button.
6. The Controller restarts with the original LEGO firmware.

3.5.3 LEGO Firmware Loader does not find the Controller

If the **LEGO Firmware Loader** does not find the Controller connect the USB cable with a different USB port. By doing so the USB driver is installed for the Firmware Loader mode.

If this does not solve the problem deinstall the driver. To do this:

1. Open the device manager in the system control while the Controller is connected.
2. Select the LEGO MINDSTORMS **NXT Firmware Update Mode** driver.
3. Right-click on it.



4. Select **Deinstall** in the context menu.
5. Remove the USB cable from the LEGO MINDSTORMS 2.0 Controller and reconnect it.
6. The driver is automatically reinstalled.

3.5.4 zenon Logic Workbench driver does not communicate with the Controller

The communication with the Controller is a point-to-point communication via a virtual serial port. It is only possible to connect one application at a time.

If the communication fails there might be several reasons. The most frequent include:

1. The stratonNG driver and zenon are still online.
2. The COM port settings are incorrect. Select the correct COM port settings in the menu **Communication parameters** (on page 18).

3. The connection settings for the Controller are not set to the correct medium (USB or Bluetooth). Select the currently used connection.

3.5.5 stratonNG driver does not communicate with the Controller

The communication with the Controller is a point-to-point communication via a virtual serial port. It is only possible to connect one application at a time.

If the communication fails there might be several reasons. The most frequent include:

1. The zenon Logic Workbench is still online.
2. The COM port settings are incorrect.
3. The straton Runtime has not been started on the LEGO MINDSTORMS 2.0 Controller.
In order to start the Runtime:

- a) Press the orange key.
- b) As soon as you see the entry straton press the orange button again.
- c) Select **Start** and press the orange key.

You can tell that the Runtime is running when the square symbol next to the battery status is rotating.

4. The straton project was not downloaded to the Controller.
If a project was downloaded please check if it is the correct project.
5. The symbol names of the variable have not been activated.
6. The connection name (on page 32) of the **stratonNG** driver is not identical with the name of the zenon Logic project.
7. The connection settings for the Controller are not set to the correct medium (USB or Bluetooth). Select the currently used connection.

3.6 Use of complex variables

This section explains how to control a motor with complex variables.

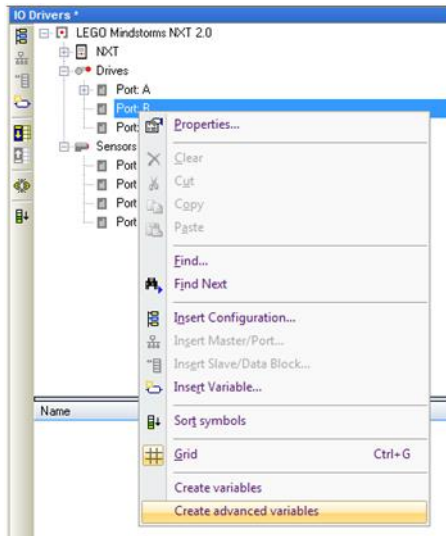


Information

The use of complex variables is only recommended for experienced users!

With an additional programming you can make the NXT motors move by a predefined angle. This is required in order to engineer very precise movements. For engineering:

1. Create all variables by right-clicking on the port and selecting **Create advanced variables** in the context menu.



2. You receive 15 variables.

The variables relevant for you are:

- ▶ *PORT_n_FLAGS*
 - ▶ Bit mask from which variables should be written.
- ▶ *PORT_n_TACHOLIMIT*
 - ▶ The relative limit for the degrees with which the motor is supposed to rotate.
 - ▶ **Attention:** This value is always positive, even if the motor rotates backwards.
- ▶ *PORT_n_MODE*
 - ▶ Mode the motor is supposed to use.
 - ▶ Does not refer to forward or backward or increasing or decreasing speed, but to internal commands.
- ▶ *PORT_n_SPEED*
 - ▶ Motor speed.
 - ▶ Range: -100 (backward) to 100 (forward)
- ▶ *PORT_n_RUNSTATE*
 - ▶ How the motor is supposed to run.
 - ▶ Physical properties (set speed instantly, Ramp UP or Ramp DOWN)



Attention

All variables must be written in a cycle!

That means: The variables cannot be forced from the zenon Logic Workbench.

STRUCTURE

Internally, NXT uses a structure. Create all variables of this structure by right-clicking on the port and selecting **Create advanced variables** in the context menu. This structure is written in every cycle. However, since **PORT_n_FLAGS** is set by straton to 0 after each cycle nothing else will happen.

The variable **PORT_n_Flags** is a bit mask for those variables which are relevant in this cycle for the structure straton queries. In order to give a **Move n degree** command set the **PORT_n_FLAGS** variable to a bit mask [ST]:

```
#define UPDATE_MODE      16#01  //The PORT_n_MODE variable is updated
#define UPDATE_SPEED     16#02  //The PORT_n_SPEED variable is updated
#define UPDATE_TACHO_LIMIT 16#04  //The PORT_n_TACHOLIMIT variable is updated

//Set FALGS to update all variables
PORT_B_FLAGS := any_to_usint(UPDATE_MODE or UPDATE_SPEED or UPDATE_TACHO_LIMIT);
```

It is then defined that the data for **PORT_n_MODE**, **PORT_n_SPEED** and **PORT_n_TACHOLIMIT** are available. Which values must these variables have?

For **PORT_n_MODE** also a bit mask is required. In principle always both must be set if you would like to create a forward movement [ST]:

```
#define MOTORON 16#01 //Has nothing todo with the physical state. Needed for some reason
#define BRAKE 16#02 //Has nothing todo with the physical state. Needed for some reason

//Set the MODE
PORT_B_MODE := any_to_usint(MOTORON or BRAKE);
```

Now you have set the bit mask on which variables are supposed to be updated as well as some internal flag parameters.

Now we can focus on the more interesting part: The rotation mode of the motor - **PORT_n_RUNSTATE**. No bit mask is available in this case. The values are exclusive.

```
#define MOTOR_RUN_STATE_IDLE 16#00 //Motor idles / does nothing
#define MOTOR_RUN_STATE_RUNNING 16#20 //Motor turnes

PORT_B_RUNSTATE := any_to_usint(MOTOR_RUN_STATE_RUNNING);
```

And finally: **PORT_n_SPEED** And **PORT_n_TACHOLIMIT**:

```
//Full force 360 degree foreward turn
PORT_B_SPEED      := 100;
PORT_B_TACHOLIMIT := 360;
```




Attention

Make absolutely sure that these values are not written in every cycle. Otherwise the rotation command is executed in every cycle. However, you still have to write all variables in every cycle. Block the writing of these values.

4 zenon Science Package for LEGO MINDSTORMS EV3

The **zenon Science Package for LEGO MINDSTORMS EV3** enables the programming of LEGO MINDSTORMS EV3 objects.

Preparation:

zenon 7.20 already installed and installations medium available.

If you have not installed zenon, you have to install **zenon Science Package for LEGO MINDSTORMS**, version 7.20 or higher.

4.1 Requirements

Hardware:

- ▶ Netgear WNA1100 Wi-Fi Dongle
- ▶ LEGO MINDSTORMS EV3 with firmware version 1.03E or 1.03H or higher.

Software:

- ▶ zenon from version 7.20 on
- ▶ LEGO MINDSTORMS EV3 Runtime (available on the installation medium in the path `\AdditionalSoftware\Science Package - Firmware\EV3`)
- ▶ Bricx Command Center (available at <http://bricxcc.sourceforge.net/> (<http://bricxcc.sourceforge.net/>))

FIRMWARE FOR LEGO MINDSTORMS EV3

The firmware on the LEGO MINDSTORMS EV3 must be at least version 1.03E or 1.03H or higher.

To check the version of the firmware, select **Brick Info** in the **Settings** tab of LEGO MINDSTORMS EV3. You can download the most recent firmware (*.bin file) from the LEGO website.

Use the LEGO software to update the firmware.

4.2 Installation

The installation comprises:

- ▶ zenon Science Package
- ▶ EV3 Setup
- ▶ zenon Logic Editor and Runtime

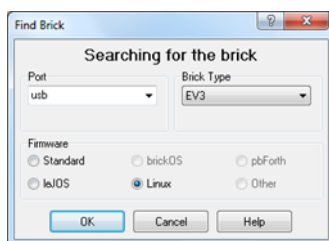
If you have already installed zenon 7.20 or higher, you can execute the EV3 setup immediately.

INSTALLATION OF THE ZENON SCIENCE PACKAGE

Start the installation from the zenon installation medium and follow the notes in the installation assistant. You can also download the zenon Science Package from the COPA-DATA home page. This installation already includes zenon Logic.

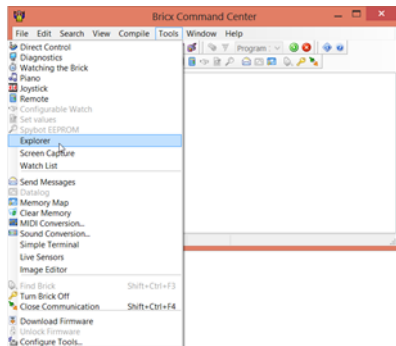
INSTALLATION OF STRATON RUNTIME ON LEGO EV3

1. Install the **Bricx Command Center**.
You can find the program at http://bricxcc.sourceforge.net/test_releases (http://bricxcc.sourceforge.net/test_releases).
2. Download a current **test_release**.
test_release2013007.zip is used in these instructions.
3. Connect the LEGO MINDSTORMS EV3 to the PC with a USB cable.
4. Open **Bricx CC: Tools -> Find Brick**
5. Configure the connection:

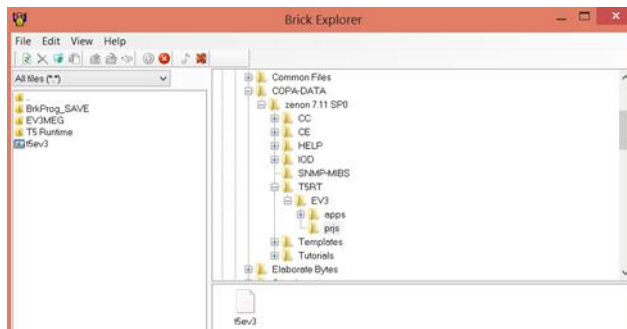


- ▶ **Port:** *usb*
- ▶ **Brick Type:** *EV3*
- ▶ **Firmware:** *Linux*

6. Open **Brick Explorer**: Tools -> Explorer.



7. Create a new folder called **T5 Runtime**.
8. Open, on the installation medium, the folder **T5 Runtime** in the following path: `\AdditionalSoftware\Science Package – Firmware\EV3\apps`.
9. Copy the files contained therein and add these to your **T5 Runtime**- folder in the **Brick Explorer**.
10. Open the `\AdditionalSoftware\Science Package – Firmware\EV3\prjs` folder on the installation medium.
11. Copy the file there called **t5ev3** to the folder in which the **T5 Runtime** folder is also located.



- On the LEGO EV3 Brick, the new entry **T5 Runtime** is now available.



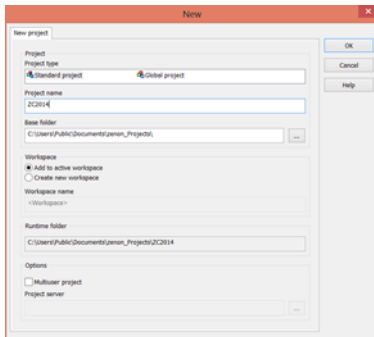
4.3 Create a zenon Logic project

zenon Logic contains a fully-integrated configuration tool for the LEGO MINDSTORMS EV3, as well as a library with Functions and Function Blocks.

The first stage is to create a zenon project and a zenon Logic project in this.

CREATE A ZENON PROJECT

1. Open the zenon Editor
2. Select **File -> New project**.
3. The dialog to create a project is opened.

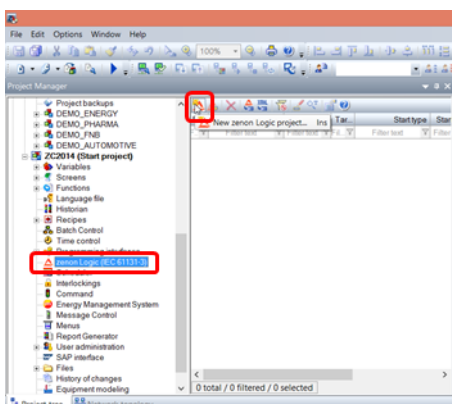


4. Enter a name for the project.
5. Click **OK**.
 - ▶ The wizard to create the project is started.
 - ▶ Close the wizard.

Alternatively, you can also open the supplied demo project.

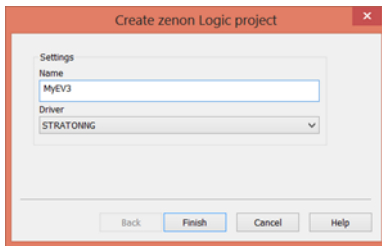
CREATE A ZENON LOGIC PROJECT

1. In the zenon Editor, go to **zenon Logic (IEC 61131-3)**.
2. In the context menu or in the tool bar of the detail view, select **zenon Logic New project**.



3. The dialog to create a project is opened.
4. Enter a name for the project.

- Click **Fertigstellen**.

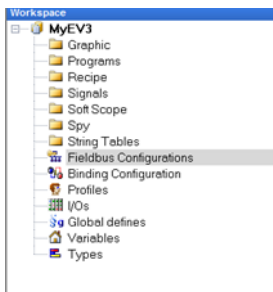


4.4 EV3 configuration in zenon Logic

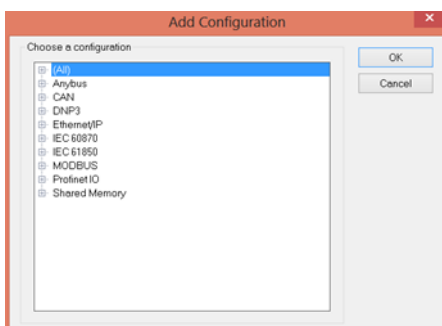
Runtime administers an allocation table, which contains the LEGO MINDSTORMS EV3 Inputs and Outputs. A tool for configuration is integrated into the zenon Logic Workbench.

To start the configuration:

- In the zenon Logic Workbench, double click on **Field bus configuration**.

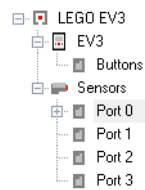


- The **I/O driver** window is opened.
- Right click on a free position in this window.
- Select **Add configuration** in the context menu.
- The **Add configuration** window is opened.



- Select the **All** node.
- Select **LEGO EV3**.
- Click on **OK**.

The configuration is created in the tree view:



LEGO EV3 CONFIGURATION

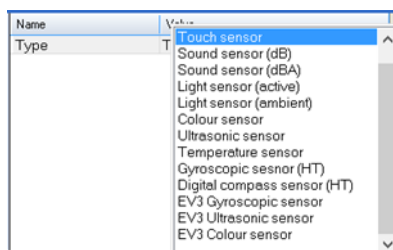
- ▶ EV3
 - ▶ Buttons
- ▶ Sensors
 - ▶ Port 0
 - ▶ Port 1
 - ▶ Port 2
 - ▶ Port 3

4.4.1 Define sensor type

You can define the type of sensor for each port in the Sensors group.

To do this:

1. Double-click on the port.
2. The configuration window will open
3. Double-click on **Type**.
4. The selection list is opened.
5. Select the desired *type*.



6. Close the dialog by clicking on **OK**.

4.4.2 Create variables

You can create variables automatically or connect them manually.

CREATE VARIABLES AUTOMATICALLY

1. Right-click on the port.
2. Select **Create variables** in the context menu.
3. All required variables are created automatically.

Note: If the sensor type is set to *None*, no variables are created.

CONNECT VARIABLES MANUALLY

To connect variables manually:

1. Right-click on the port.
2. Select **Add variables** in the context menu
3. The dialog for configuration is opened.
4. Clicking on a value opens the selection list of the values that are possible for this type.

CONFIGURE VARIABLES

Parameters	Description
Symbol	Selection of a variable in accordance with IEC 61131-3 syntax.
Area	Range.
Format	Data type: <ul style="list-style-type: none"> ▶ 32 bit float ▶ Signed 16 bit integer ▶ Signed 32 bit integer ▶ Signed 8 bit integer ▶ Single bit ▶ Unsigned 16 bit integer ▶ Unsigned 32 bit integer ▶ Unsigned 8 bit integer
Offset	Is not used.

Parameters	Description
Bit Number	Bit number.

DATATYPES

You can connect variables with any desired data type to the LEGO EV3 I/Os. Runtime automatically converts the values to the data type of the variable.

Note: STRING variables are not supported.

4.4.3 Blocks and Functions function

Some Function Blocks and Functions are provided in the library to control LEDs, sounds, LCD and motors of the LEGO EV3.

The following Function Blocks are described in more detail:

- ▶ **LED**
- ▶ **Sound**
- ▶ **Motor**

4.4.3.1 LED

Set LED pattern:

Parameters	Description
SetLed	Creates an LED pattern
[IN] LED	Index of the LED pattern to be applied: <ul style="list-style-type: none"> ▶ 0: All LEDs off. ▶ 1-9: Corresponding LED pattern.
[OUT] Q	Current LED pattern.

4.4.3.2 Sound

Set sound playback.

General:

Parameters	Description
PlaySnd	Play back sounds.
[IN] ENABLE	Switch on sound playback: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Sounds are played back. To activate playback, the setting must have previously been <i>FALSE</i>. ▶ <i>FALSE</i>: Sounds are not played back.
[IN] FREQ	Frequency of the sound.
[IN] DURATION	Duration of the signal.
[IN] VOLUME	Volume <ul style="list-style-type: none"> ▶ Minimum: 0 ▶ Maximum: 100
[OUT] PLAYING	<i>TRUE</i> , if sounds are played back.

System sounds:

Parameters	Description
PlaySysSnd	Play back system sound.
[IN] ENABLE	Switch on sound playback: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Sounds are played back. To activate playback, the setting must have previously been <i>FALSE</i>. ▶ <i>FALSE</i>: Sounds are not played back.
[IN] SND	Index of the system sound.
[IN] VOLUME	Volume. <ul style="list-style-type: none"> ▶ Minimum: 0 ▶ Maximum: 100
[OUT] PLAYING	<i>TRUE</i> , if a system sound is played back.
Symbol	Variable name in accordance with IEC 61131-3 syntax.
Area	<ul style="list-style-type: none"> ▶ Data Exchange ▶ Data Status.
Format	Data type:

Parameters	Description
	<ul style="list-style-type: none"> ▶ 32 bit float ▶ Signed 16 bit integer ▶ Signed 32 bit integer ▶ Signed 8 bit integer ▶ Single bit ▶ Unsigned 16 bit integer ▶ Unsigned 32 bit integer ▶ Unsigned 8 bit integer
Offset	Is not used.
Bit Number	Bit number.

4.4.3.3 Motor

Set motor.

MOTOR INFORMATION

Motor information:

Parameters	Description
MotorInfos	Call up motor information.
[IN] MOTOR	Index of the motor: 1 - 4
[OUT] SPEED	Current speed.
[OUT] TACHO	Current number of revolutions to display the distance covered.

RUN MOTOR

Run a motor:

Parameters	Description
Rotate	Run a motor.
[IN] MOTOR	Index of the motor: 1 - 4

Parameters	Description
[IN] START	Start motor: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	Force to be applied: <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop. ▶ 1 to 255: Forward.
[OUT] RUNNING	<i>TRUE</i> if the motor is running.
[OUT] SPEED	Current speed.

Run several motors:

Parameters	Description
Rotate2	Run several motors.
[IN] MOTORS	Index of the motor numbers. <ul style="list-style-type: none"> ▶ Motor 1: 1 ▶ Motor 2: 2 ▶ Motor 3: 4 ▶ Motor 4: 8 <p>Example: To address Motors 1 and 2, enter 3 (1 + 2).</p>
[IN] START	Start motor: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	Force to be applied: <p>-255 to -1: Backwards.</p> <p>0: Stop.</p> <p>1 to 255: Forward.</p>
[OUT] RUNNING	<i>TRUE</i> if the motor is running.

Parameters	Description
[OUT] SPEED	Current speed.

ROTATION ANGLE

Rotation angle for a motor:

Parameters	Description
RotateAngel	Rotation angle for a motor.
[IN] MOTORS	Index of the motor: 1 - 4
[IN] START	Start motor: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	Force to be applied: <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop. ▶ 1 to 255: Forward.
[IN] ANGLE	Angle in degrees.
[IN] BRAKE	Brake. <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is braked at the end. ▶ <i>FALSE</i>: Motor is not braked.
[OUT] RUNNING	<i>TRUE</i> , if the motor is running.
[OUT] SPEED	Current speed.

Angle of rotation for several motors:

Parameters	Description
RotateAngel2	Angle of rotation for several motors.
[IN] MOTORS	Index of the motor numbers. <ul style="list-style-type: none"> ▶ Motor 1: 1 ▶ Motor 2: 2 ▶ Motor 3: 4

Parameters	Description
	<ul style="list-style-type: none"> ▶ Motor 4: 8 <p>Example: To address Motors 1 and 2, enter 3 (1 + 2).</p>
[IN] START	<p>Start motor:</p> <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	<p>Force to be applied:</p> <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop. ▶ 1 to 255: Forward.
[IN] ANGLE	Angle in degrees.
[IN] BRAKE	<p>Brake.</p> <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is braked at the end. ▶ <i>FALSE</i>: Motor is not braked.
[OUT] RUNNING	<i>TRUE</i> , if the motor is running.
[OUT] SPEED	Current speed.

SPEED

Speed for a motor:

Parameters	Description
RotateCount	Run a motor for a defined number of counter impulses.
[IN] MOTORS	Index of the motor: 1 - 4
[IN] START	<p>Start motor:</p> <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	<p>Force to be applied:</p> <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop.

Parameters	Description
	<ul style="list-style-type: none"> ▶ 1 to 255: Forward.
[IN] COUNT	Number of rotation impulses.
[IN] BRAKE	Brake. <i>TRUE</i> : Motor is braked at the end. <i>FALSE</i> : Motor is not braked.
[OUT] RUNNING	<i>TRUE</i> , if the motor is running.
[OUT] SPEED	Current speed.

Speed for several motors:

Parameters	Description
RotateCount2	Run several motors for a defined number of counter impulses.
[IN] MOTORS	Index of the motor numbers. <ul style="list-style-type: none"> ▶ Motor 1: 1 ▶ Motor 2: 2 ▶ Motor 3: 4 ▶ Motor 4: 8 Example: To address Motors 1 and 2, enter 3 (1 + 2).
[IN] START	Start motor: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	Force to be applied: <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop. ▶ 1 to 255: Forward.
[IN] COUNT	Number of rotation impulses.
[IN] BRAKE	Brake. <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is braked at the end. ▶ <i>FALSE</i>: Motor is not braked.

Parameters	Description
[OUT] RUNNING	<i>TRUE</i> , if the motor is running.
[OUT] SPEED	Current speed.

RUNNING TIME

Running time for a motor:

Parameters	Description
RotateTime	Run a motor for a defined time period.
[IN] MOTORS	Index of the motor: 1 - 4
[IN] START	Start motor: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	Force to be applied: <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop. ▶ 1 to 255: Forward.
[IN] SECONDS	Time that the motor runs, in seconds.
[IN] BRAKE	Brake. <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is braked at the end. ▶ <i>FALSE</i>: Motor is not braked.
[OUT] RUNNING	<i>TRUE</i> , if the motor is running.
[OUT] SPEED	Current speed.

Speed for several motors:

Parameters	Description
RotateTime2	Run several motors for a defined time period.
[IN] MOTORS	Index of the motor numbers. <ul style="list-style-type: none"> ▶ Motor 1: 1 ▶ Motor 2: 2

Parameters	Description
	<ul style="list-style-type: none"> ▶ Motor 3: 4 ▶ Motor 4: 8 <p>Example: To address Motors 1 and 2, enter 3 (1 + 2).</p>
[IN] START	<p>Start motor:</p> <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is started. To start, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Motor is not started.
[IN] POWER	<p>Force to be applied:</p> <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop. ▶ 1 to 255: Forward.
[IN] SECONDS	Time that the motors run, in seconds.
[IN] BRAKE	<p>Brake.</p> <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Motor is braked at the end. ▶ <i>FALSE</i>: Motor is not braked.
[OUT] RUNNING	<i>TRUE</i> , if the motor is running.
[OUT] SPEED	Current speed.

MOTOR POWER OUTPUT

Define the power output for a motor:

Parameters	Description
SetPower	Define the power output for a motor whilst it is turning.
[IN] MOTORS	Index of the motor: 1 - 4
[IN] SET	<p>Define power output:</p> <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Power output is defined. To do this, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Power output is not changed.
[IN] POWER	<p>Force to be applied:</p> <ul style="list-style-type: none"> ▶ -255 to -1: Backwards.

Parameters	Description
	<ul style="list-style-type: none"> ▶ 0: Stop. ▶ 1 to 255: Forward.

Define power output for several motors:

Parameters	Description
SetPower2	Run several motors for a defined time period.
[IN] MOTORS	Index of the motor numbers. <ul style="list-style-type: none"> ▶ Motor 1: 1 ▶ Motor 2: 2 ▶ Motor 3: 4 ▶ Motor 4: 8 Example: To address Motors 1 and 2, enter 3 (1 + 2).
[IN] SET	Define power output: <ul style="list-style-type: none"> ▶ <i>TRUE</i>: Power output is defined. To do this, the setting must have previously been set to <i>FALSE</i>. ▶ <i>FALSE</i>: Power output is not changed.
[IN] POWER	Force to be applied: <ul style="list-style-type: none"> ▶ -255 to -1: Backwards. ▶ 0: Stop. ▶ 1 to 255: Forward.

STOP MOTOR

Stop a motor:

Parameters	Description
StopMotor	Stop a motor.
[IN] MOTOR	Index of the motor: 1 - 4
[IN] STOP	Motor is stopped.

Stop several motors:

Parameters	Description
StopMotor2	Stop several motors.

Parameters	Description
[IN] MOTORS	<p>Index of the motor numbers.</p> <ul style="list-style-type: none"> ▶ Motor 1: 1 ▶ Motor 2: 2 ▶ Motor 3: 4 ▶ Motor 4: 8 <p>Example: To address Motors 1 and 2, enter 3 (1 + 2).</p>
[IN] STOP	Motors are stopped.

4.5 Connect EV3 to zenon Logic Editor

LEGO MINDSTORMS EV3 and zenon Logic communicate via Wi-Fi.

To establish communication:

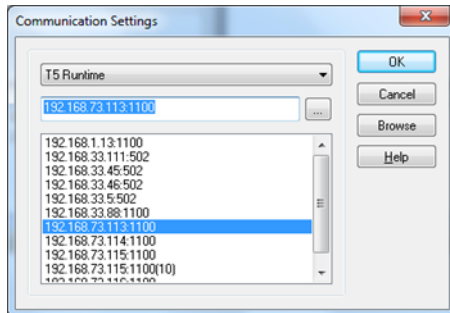
1. Insert the **Netgear WNA1100 Wi-Fi Dongle** into the USB port of the Brick.
2. Switch on the Wi-Fi on the LEGO MINDSTORMS EV3.
To do this:
 - a) Open the **Settings** tab (tool symbol).
 - b) Activate the **Wi-Fi**.
3. Connect to your network.



You can find further information about Wi-Fi in your LEGO MINDSTORMS EV3 instructions.

4. Define the IP address of the LEGO MINDSTORMS EV3 in zenon Logic Editor.
To do this:
 - a) Open the **Tools** menu.
 - b) Select the **Communication parameters** command.
 - c) The dialog to select the IP address is opened.

- d) Select the IP address of the EV3. (See also the First steps in the demo project (on page 59) chapter.)



Note:

- You can find the correct IP address in the Brick in the **Settings** tab (tool symbol) under **Brick Info**.
 - Runtime uses port *1100*.
5. Start the T5 Runtime on the LEGO MINDSTORMS EV3.
To do this:
- a) Select, in the **App** tab (arrow symbol) of the Brick, **T5 Runtime**.
 - b) Press the middle switch on the Brick.
 - c) Starts the Runtime.



You can now create a new program in the zenon Logic Workbench to control your LEGO MINDSTORMS EV3 EV3 with the help of the program.

4.6 First steps in the demo project

You can try out the basic functions and configuration of LEGO MINDSTORMS EV3 with zenon and zenon Logic with the supplied demo project.

To do this:

1. Open the demo project called **EV3_DEMO** in zenon Editor.

2. Connect the sensors and motors to the corresponding ports.

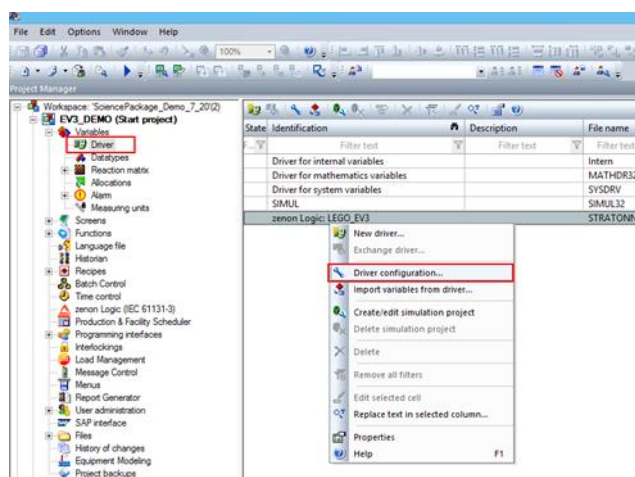


3. Configure the **zenon Logic: LEGO_EV3** driver.

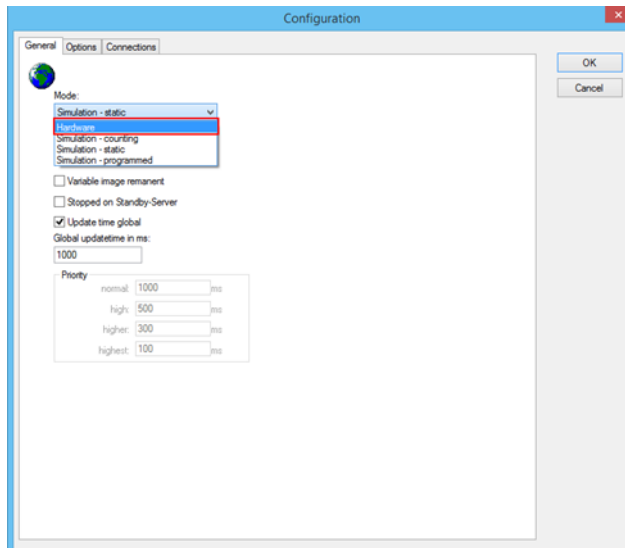
CONFIGURING THE DRIVER

To configure the driver:

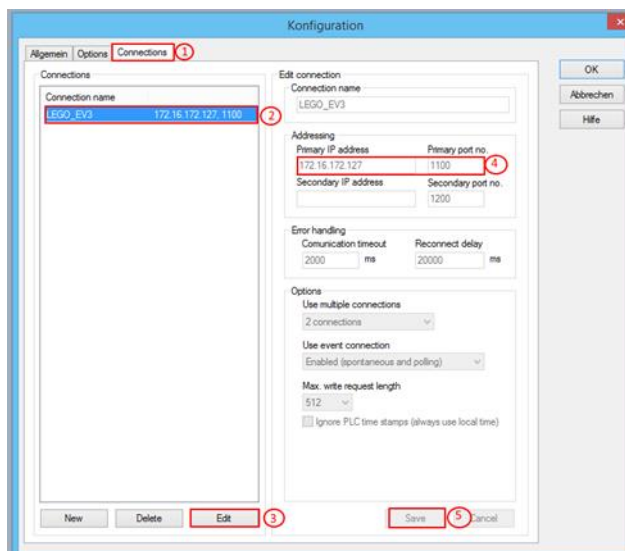
1. In the zenon project, go to the **Driver** node (below **Variables**).
2. Right-click on the driver **zenon Logic: LEGO_EV3**.
3. Select **Driver configuration** in the context menu



4. The configuration window will open



5. In the **General tab**, select the *Hardware* entry for the **Mode** option.
6. Switch to tab **Connections** (1).

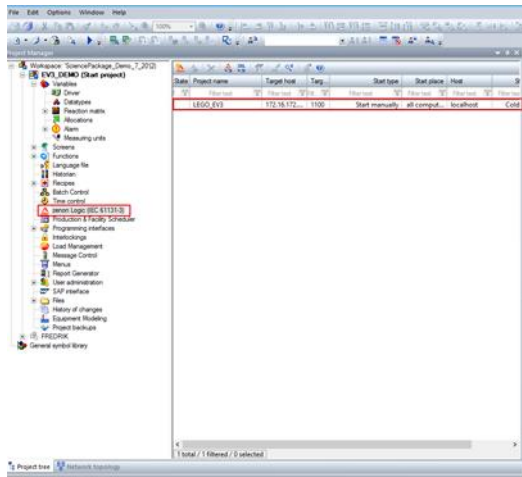


- a) Highlight the **LEGO_EV3** connection (2).
- b) Click on **Edit** (3).
- c) For the **Primary IP address** option, enter the IP address of your Brick.
You can find the address in the Brick in the **Settings** tab (tool symbol) under **Brick Info**.
- d) Change the **Primary port no** option (4) to **1100**.
- e) Confirm the input by clicking **Save** (5).
7. Close the window by clicking on **OK**.

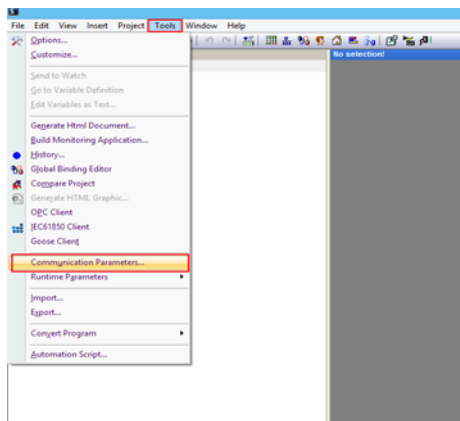
CONFIGURE ZENON LOGIC PROJECT AND START RUNTIME

Once you have configured the driver, open the zenon Logic project.
To do this:

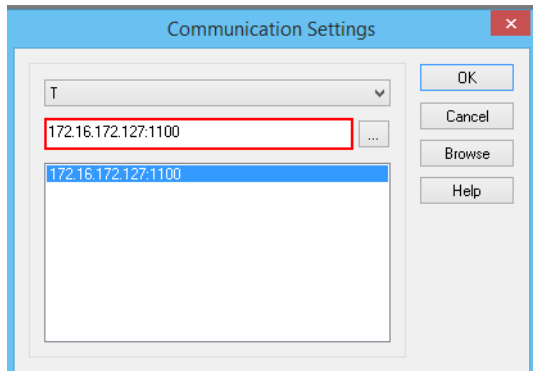
1. In the zenon Editor, go to **zenon Logic (IEC 61131-3)**.
2. Double click on the project



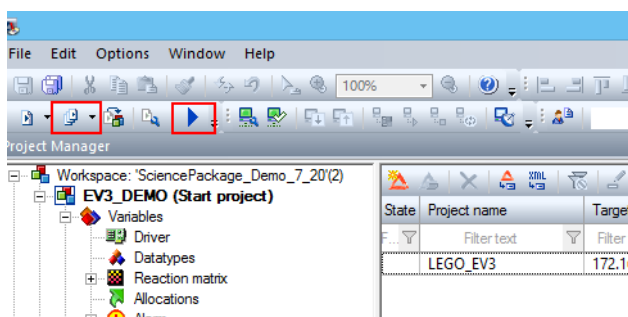
3. Open, in zenon Logic Workbench, the **Tools** menu.
4. Click on **Communication parameters**.



5. The dialog to configure communication parameters is opened.
(See also Connect EV3 to zenon Logic Editor (on page 58) chapter.)



6. Enter the IP address of your LEGO MINDSTORMS EV3 Brick and its port number.
Format: *IP address port number*.
You can find the address in the Brick in the **Settings** tab (tool symbol) under **Brick Info**.
Port number: *1100*.
7. Use straton Runtime to connect to the EV3.
to do this:
 - ▶ Open the menu **Project**
 - ▶ Select the **Online** entry.
 - ▶ Confirm the question after downloading the project with **OK**.
 - ▶ Load the project into the EV3.
(Alternatively: **Project -> Download application**.)
8. Switch to the zenon Editor
9. In the **Runtime files** tool bar, click on **Create all Runtime files**.
10. Click on **Start Runtime**.



EV3 Runtime is started.

4.7 COPA-DATA Forum

For questions in relation to **zenon Science Package for LEGO MINDSTORMS**, you can also receive information and support in the Science Package Thread in the COPA-DATA Forum:

www.copadata.com/forums (<http://www.copadata.com/forums>).