

All proprietary names and product designations are brand names or trademarks registered to the relevant title holders.

## **Services**

For service and support, please contact your local sales organisation.

Contact details: [Eaton.com/contacts](https://www.eaton.com/contacts)  
Service page: [Eaton.com/aftersales](https://www.eaton.com/aftersales)

## **Original Document**

The Dutch version of this training manual is the original.

## **Translations of the Original document**

All non-Dutch language versions of this training manual are translations of the original document  
Trainingshandboek easySoft 8 GH135 NL 3<sup>rd</sup> edition 12/23.

1st Edition 2026, redaction date October 2024

© 2019 by Eaton Industries (Netherlands) B.V.

All programs have been drawn up and carefully tested by us to the best of our knowledge. These are intended as examples. Company Eaton is not liable for any errors.

All rights reserved. No part of this manual may be reproduced or processed, multiplied, or distributed using electronic systems, in any form whatsoever (print, photocopy, microfilm, or any other method), without the written permission of company Eaton.

Subject to change.

# Table of contents

|                                             |     |
|---------------------------------------------|-----|
| Chapter 1 Introduction .....                | 4   |
| Chapter 2 Example program .....             | 5   |
| 2.1 Create project.....                     | 5   |
| 2.2 Save program .....                      | 18  |
| 2.3 Simulation .....                        | 20  |
| 2.4 Send program to easyE4 .....            | 24  |
| 2.5 View online.....                        | 38  |
| 2.6 Oscilloscope .....                      | 40  |
| 2.7 Timer .....                             | 44  |
| 2.8 Markers and marker words.....           | 52  |
| 2.9 Counter .....                           | 59  |
| 2.10 Create a function block.....           | 63  |
| 2.11 Export function block.....             | 77  |
| 2.12 Program in Function Block Diagram..... | 81  |
| 2.13 Add I/O-module .....                   | 88  |
| 2.14 Call function block again .....        | 92  |
| 2.15 Visualization.....                     | 96  |
| 2.16 Alarm screen .....                     | 114 |
| 2.17 Assign fixed IP-address .....          | 118 |
| 2.18 Webserver.....                         | 123 |
| Chapter 3 Example programs.....             | 128 |
| 3.1 Format SD-card .....                    | 128 |
| 3.2 Boot logo.....                          | 130 |
| 3.2 Data storage .....                      | 131 |
| 3.3 Easy network.....                       | 140 |
| 3.4 Default settings .....                  | 157 |
| Notes .....                                 | 158 |

# Chapter 1 Introduction

The easyE4 was developed for small and medium-sized control systems. These can be control systems in both mechanical engineering and installations. Due to its compact design, the easyE4 is suitable for installation in distribution systems and is therefore ideal for the installer.

The basic easyE4 can be expanded with a maximum of 11 extensions. This results in 188 digital I/Os with relay or transistor outputs, or 70 analog I/Os, or a mix of all of these. The extensions are available in different supply voltages and can be used interchangeably in a single control system.

Should new functionalities for the easyE4 become available in the future, they can be easily added by means of a firmware update.

The MicroSD card is not only for storing various programs, but can also be used for data logging.

The Ethernet port is used for programming, connecting control panels, and creating an easyE4 network. A total of 8 easyE4 devices can be included in an Ethernet network. These are connected to one another by means of an Ethernet switch.

Although programming on the device itself is possible, the software offers so much user convenience that it is almost indispensable. EasySoft 8 is the premier software environment for programming the easyE4 control relay from Eaton.

The easyE4 features 4 programming languages. In addition to Easy Device Programming (EDP), the easyE4 now also features the languages Ladder (LD), Function Block Diagram (FBD), and Structured Text (ST).

In this brief manual, the programming of the easyE4 series is explained using the easySoft 8 software, utilizing simple, small tasks. For these tasks, an easy-E4-DC-12TC1 and the easy-E4-DC-16TE1 extension are used.

Although easySoft 8 is much more extensive than described here, this manual can provide you with a good start in beginning to program in easySoft 8.

We wish you much enjoyment with your easyE4.

## Chapter 2 Example program

### 2.1 Create project

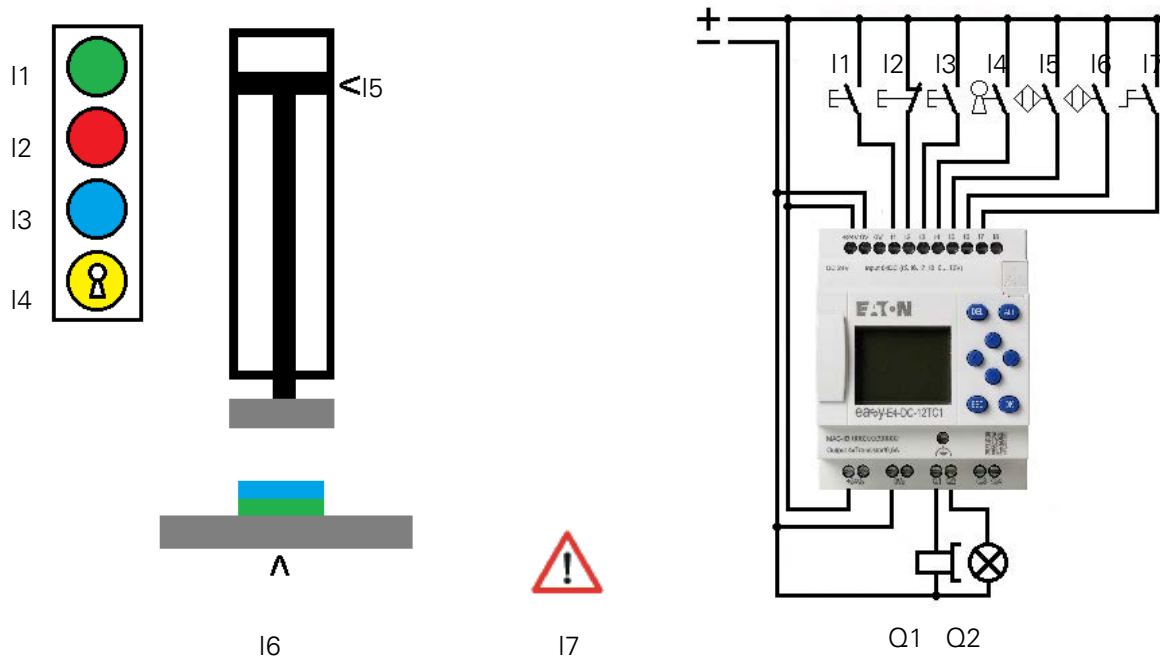


Fig.1

In the example above, a small press must be controlled. The cylinder is driven downwards with the 'Start' (I1) input by means of the 'CylinderPressing' (Q1) valve. However, that is only permitted if a product to be processed is present, which is detected by the 'ProductPlaced' (I6) sensor. The position of the press is monitored by the 'CylinderUp' sensor (I5).

The press must press for a certain time, after which the cylinder retracts again. If 'Stop' (I2) is pressed, the press opens immediately.

If an 'Error' (I7) fault occurs, the press must also open. The 'PressErrorLight' output (Q2) indicates that the press has a fault.

The fault can be reset with 'Reset' (I3) as soon as the fault is no longer present.

The number of press movements must be counted. The counter can be reset with the 'ResetCounter' (I4) key switch.

| Inputs |               | Outputs |                               |
|--------|---------------|---------|-------------------------------|
| I1     | Start         | Q1      | CylinderPressing (monostable) |
| I2     | Stop          | Q2      | PressErrorLight               |
| I3     | Reset         |         |                               |
| I4     | ResetCounter  |         |                               |
| I5     | CylinderUp    |         |                               |
| I6     | ProductPlaced |         |                               |
| I7     | Error         |         |                               |

We will build and test this application step-by-step in this manual.

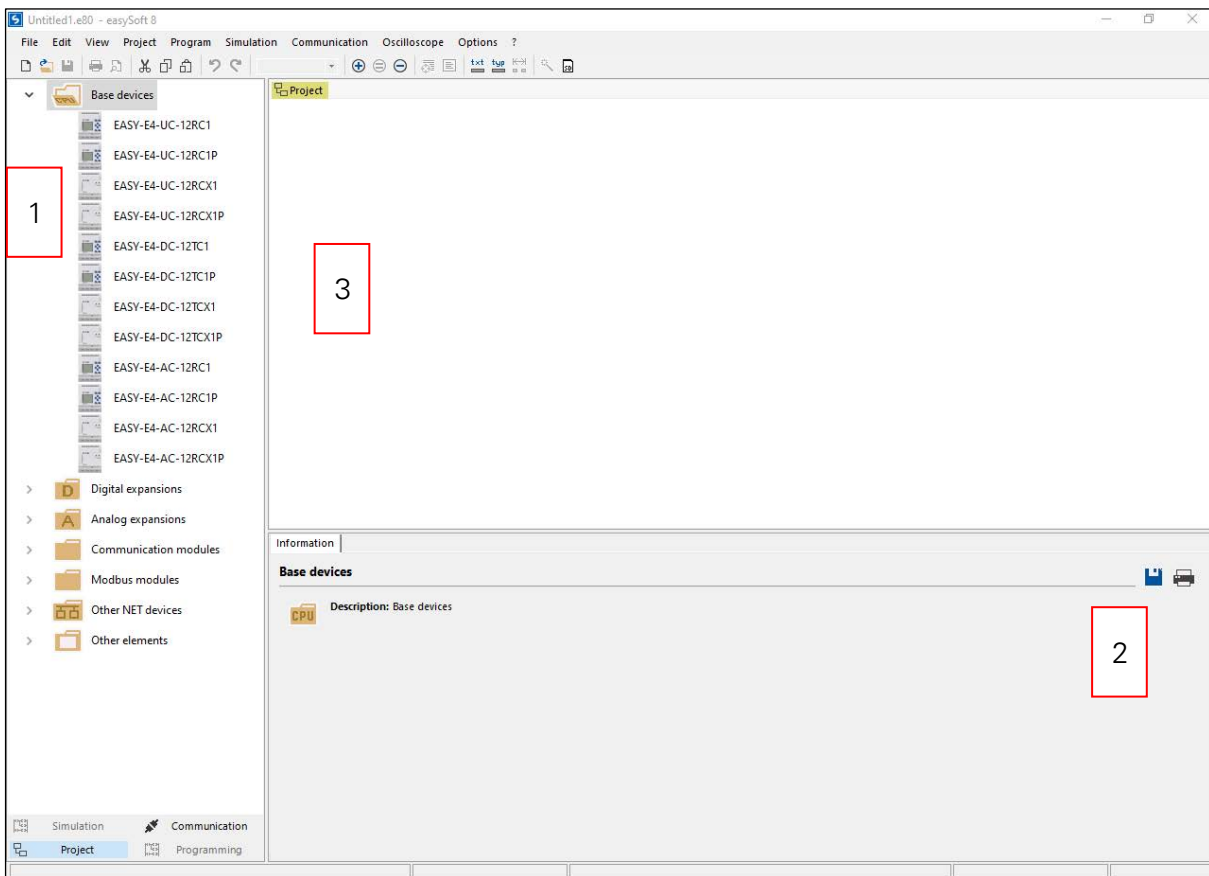


Fig. 2

After easySoft 8 has been opened, the above screen appears.

The screen is divided into 3 parts: the Toolbox [1], the Properties field [2], and the Worksheet [3].

To the bottom left of the toolbox are the various menus that can be used for work.

|               |                                                       |
|---------------|-------------------------------------------------------|
| Simulation    | => Simulating the program without an easyE4           |
| Communication | => Setting up the communication between PC and easyE4 |
| Project       | => Assembling and Parameterizing the Hardware         |
| Programming   | => Writing the program                                |

The press' instruction is created with the EASY-E4-DC-12TC1.

- ▶ Go to the toolbox and open the 'Base devices' with '>' if necessary.

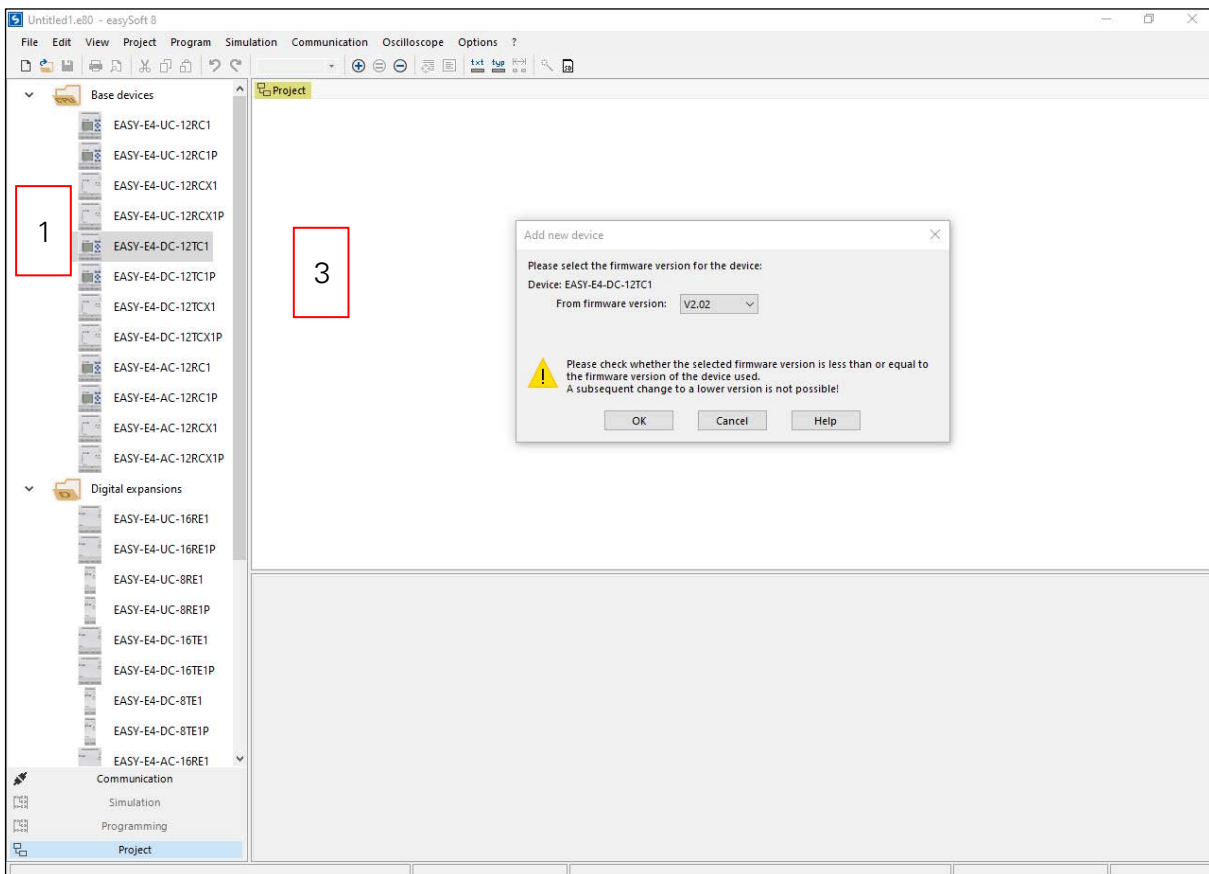


Fig. 3

First, the correct easy device must be selected, in this case an EASY-E4-DC-12TC1.

- Select the EASY-E4-DC-12TC1 by selecting it with the left mouse button and dragging it to the Worksheet [3].

A window appears in which the firmware (FW) must be chosen. The FW determines, inter alia, which functionalities are available in the easyE4.

The FW depends on the hardware. For instance, FW V2.00 can be applied from hardware (HW) version 08 onwards. The HW number is located on the left side of the easyE4 beneath the text 'Moeller Series'.

In this example, FW V2.02 and HW 08 are used.

- Select this and click "OK".

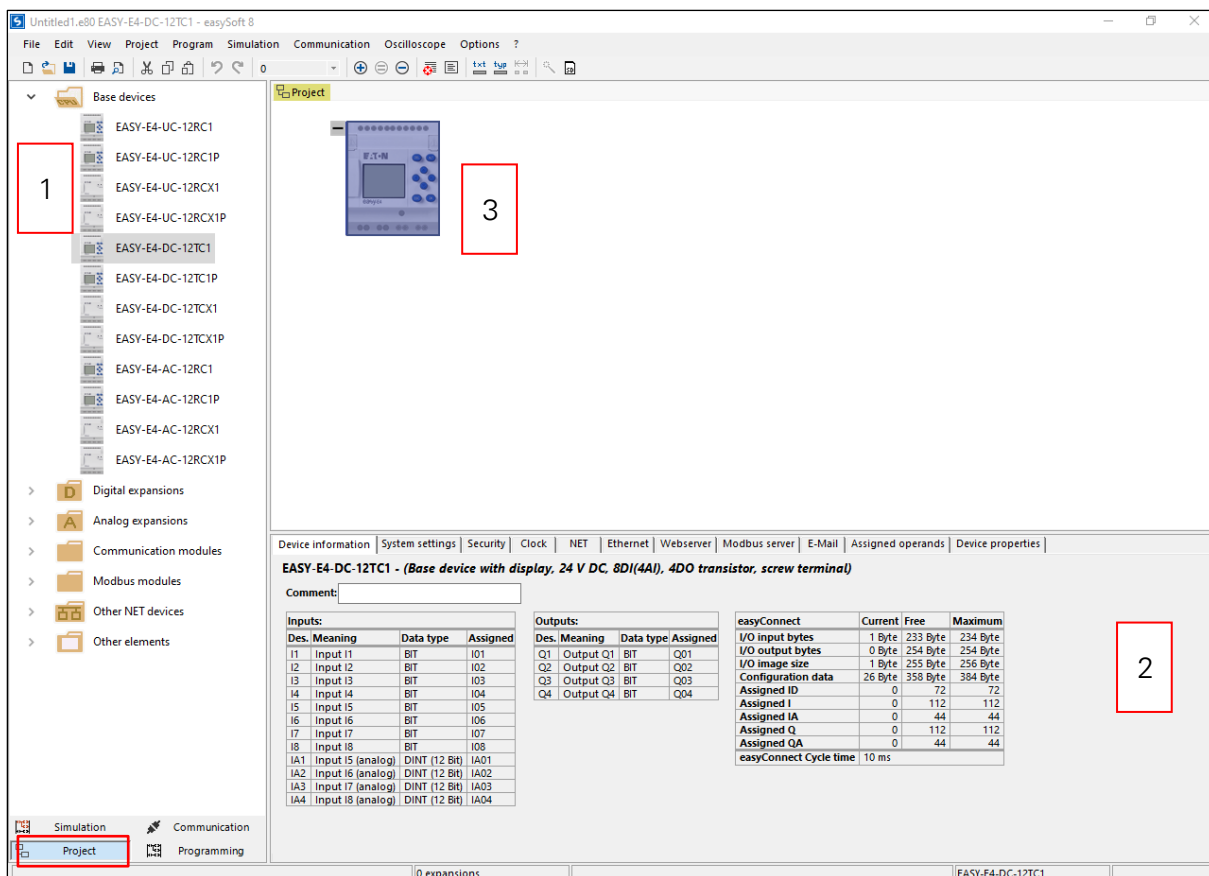


Fig. 4

In the Properties field [2] window, multiple tabs become visible.

- Select "Device information".

The information related to the device becomes visible.

## Create project

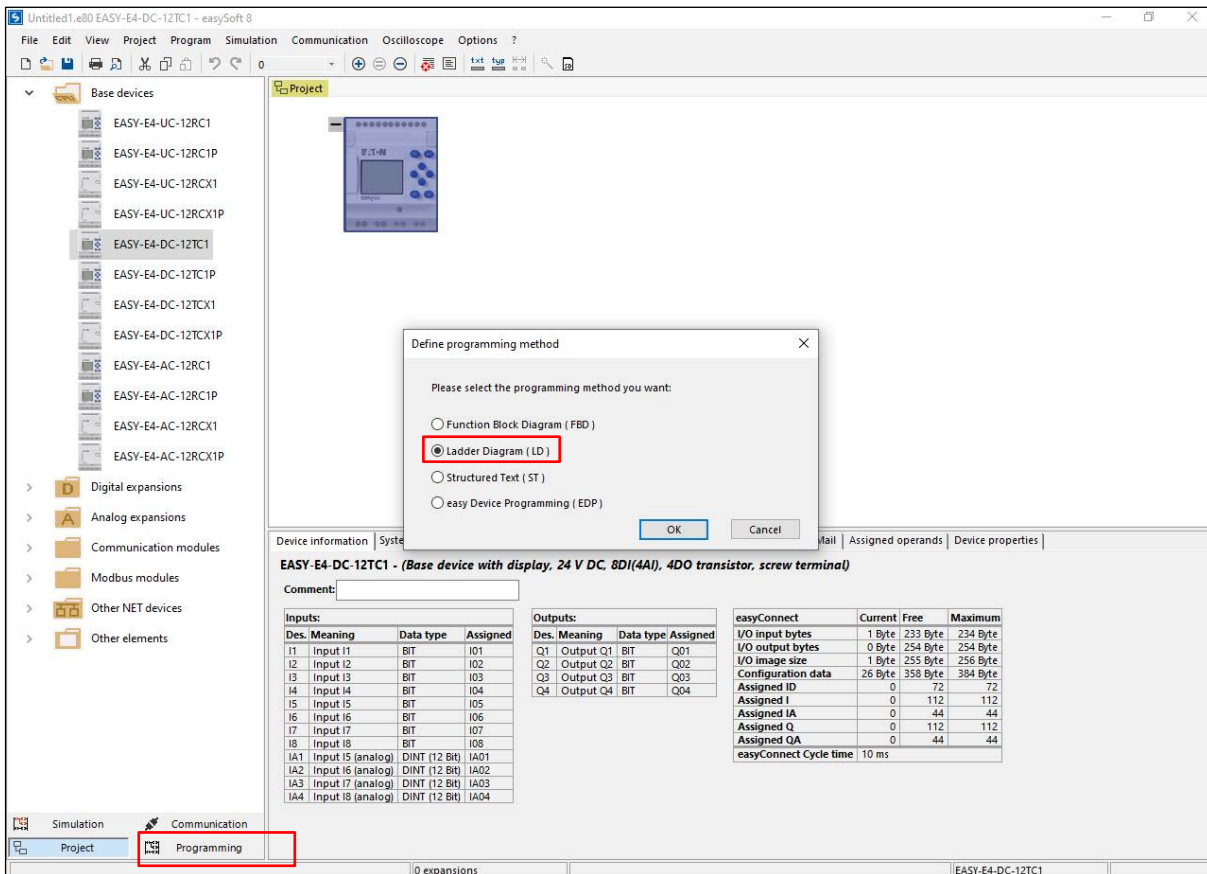


Fig. 5

Now that the Easy device has been selected, the program can be written.

- Go to the "Programming" menu.

The "Define programming method" window is displayed. In this example, the program is written in Ladder Diagram (LD).

- Select this and click "OK".

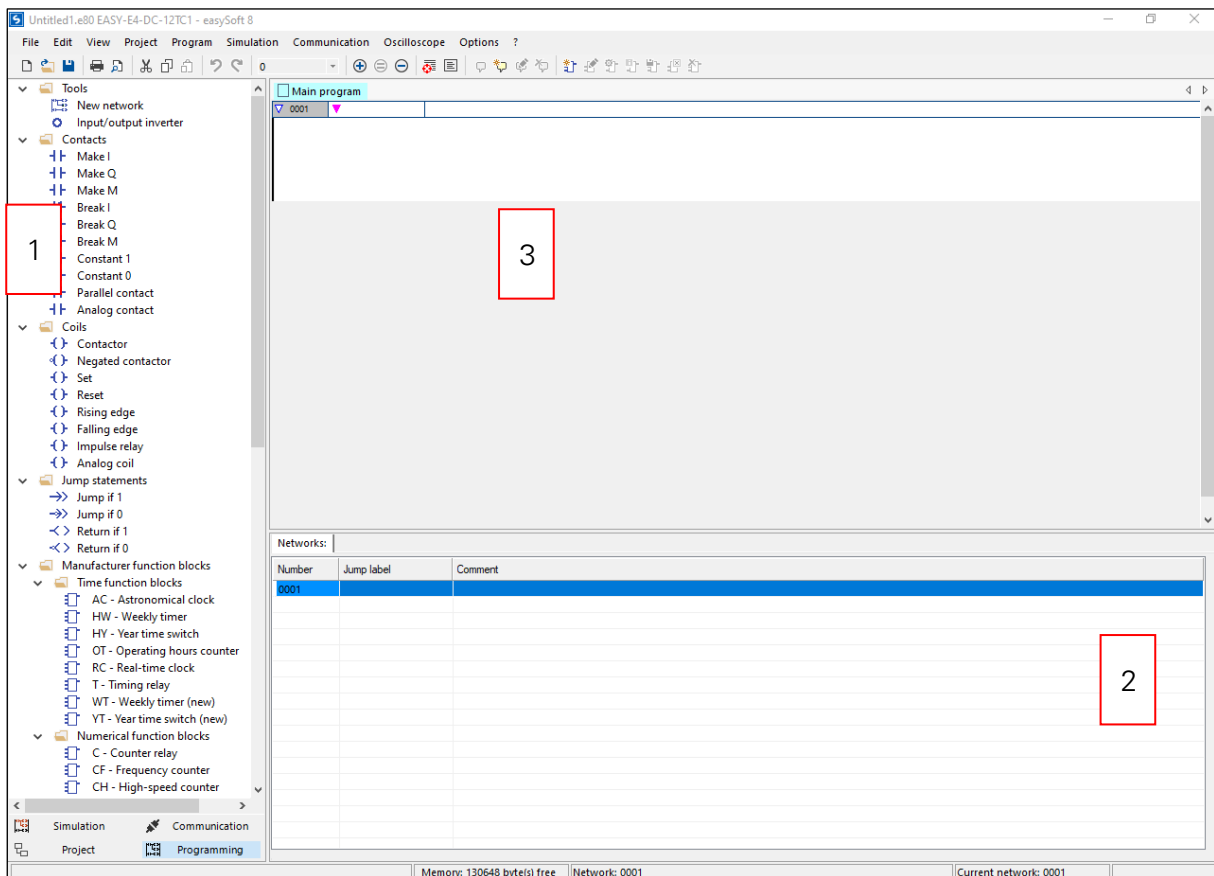


Fig. 6

A new window appears with the same layout as just now.  
Toolbox [1], the Properties field [2] and the Worksheet [3].

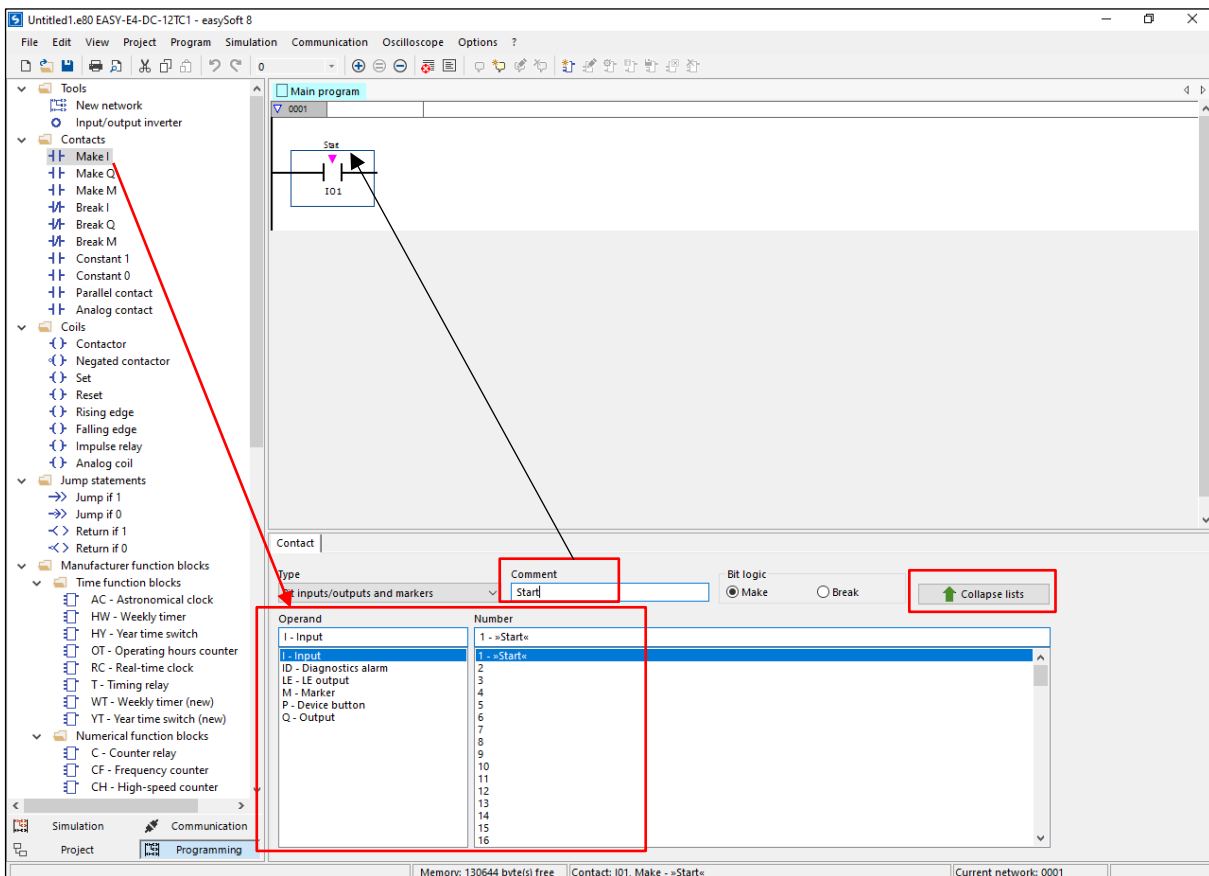


Fig. 7

The basic control of the press consists of a classic start/stop circuit.

- Drag a "Make" contact to the worksheet.
- In the "Properties" field, select ➔ "I-Input" under Operands and under "Number" ➔ "1".

A name is assigned to the inputs under 'Comment'. In this example, the name "Start". The name 'Start' now also appears next to the input on the worksheet.

To see the entire list of Operands and Numbers, it can be expanded/collapsed with the button 'Expand lists / Collapse lists'.

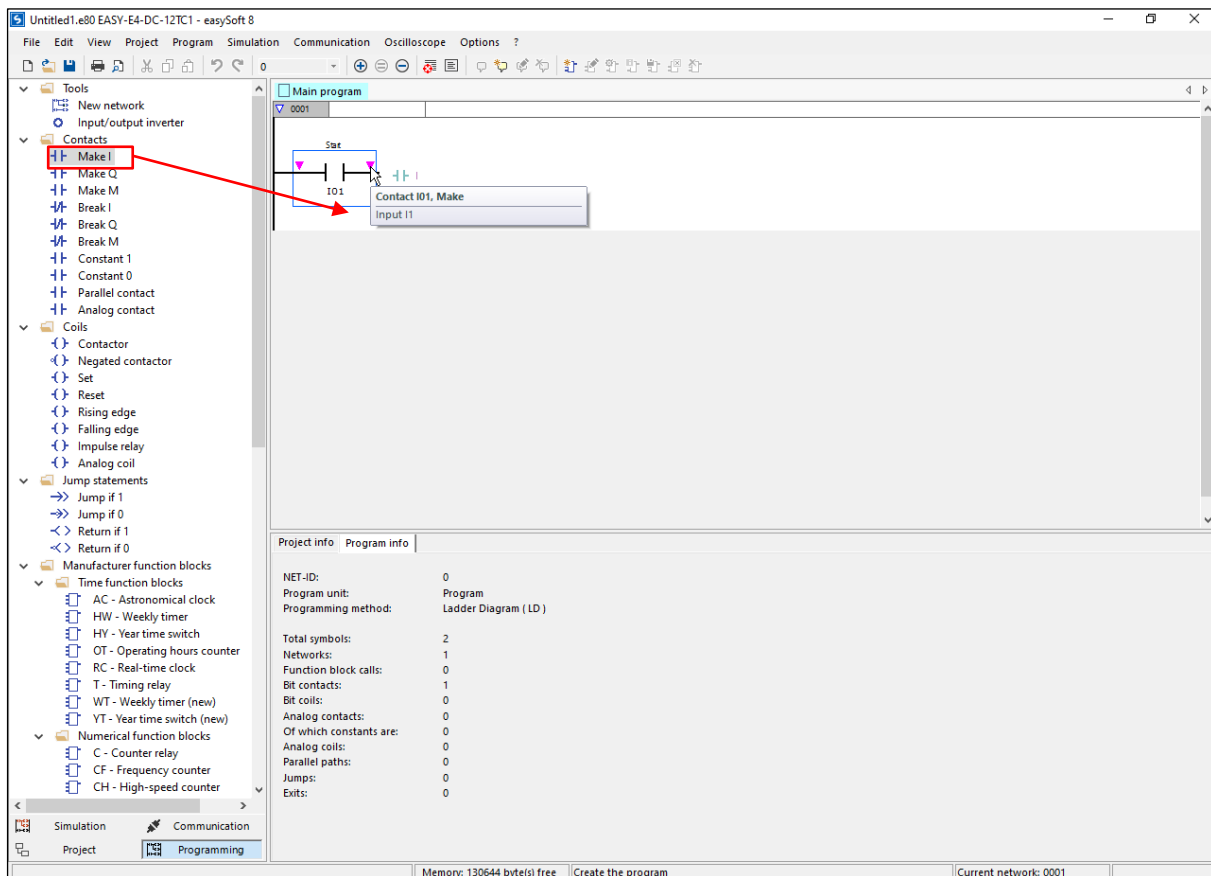


Fig. 8

- Drag a new 'Make' contact onto the worksheet and place it next to 'I1' where the pink triangle is located.

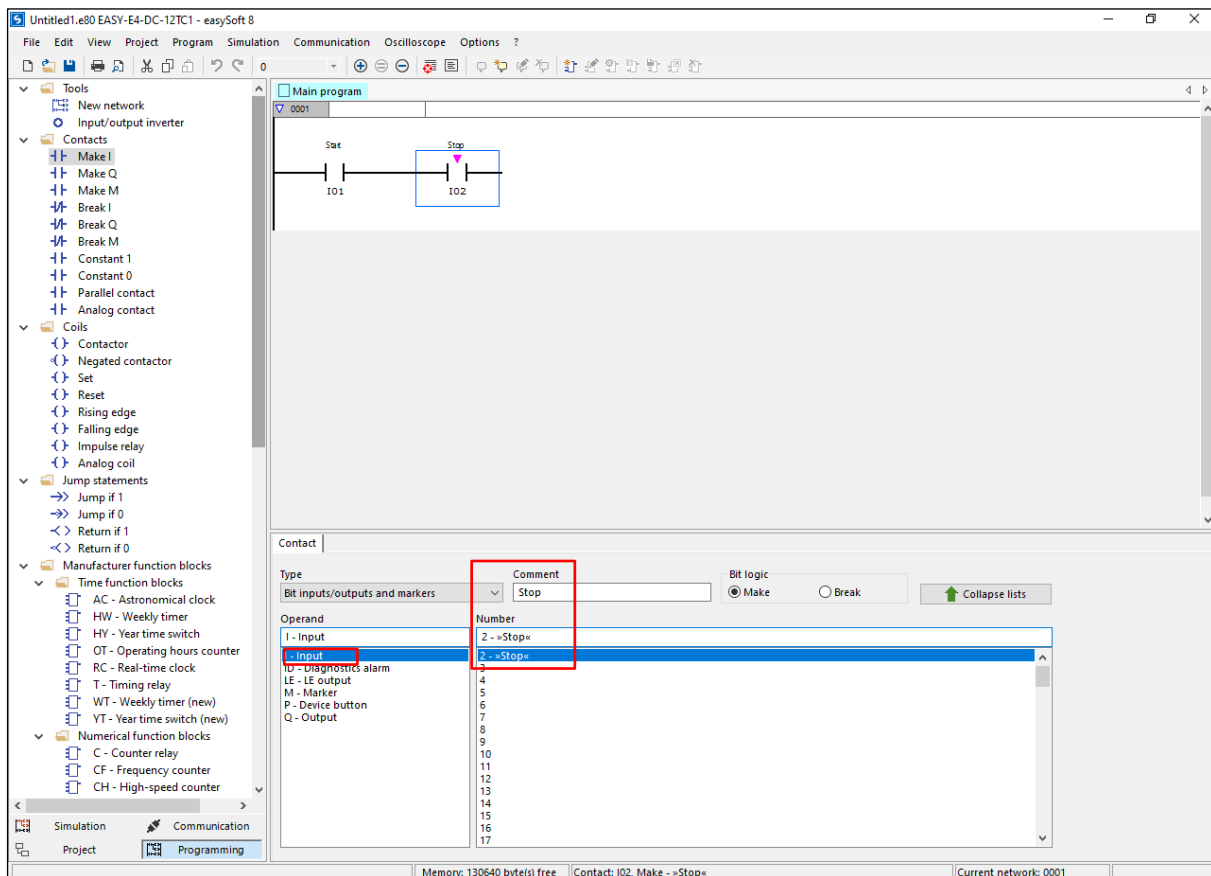


Fig. 9

The software places an 'I1' contact by default.

- Change this to 'I2' and give it the name 'Stop'.

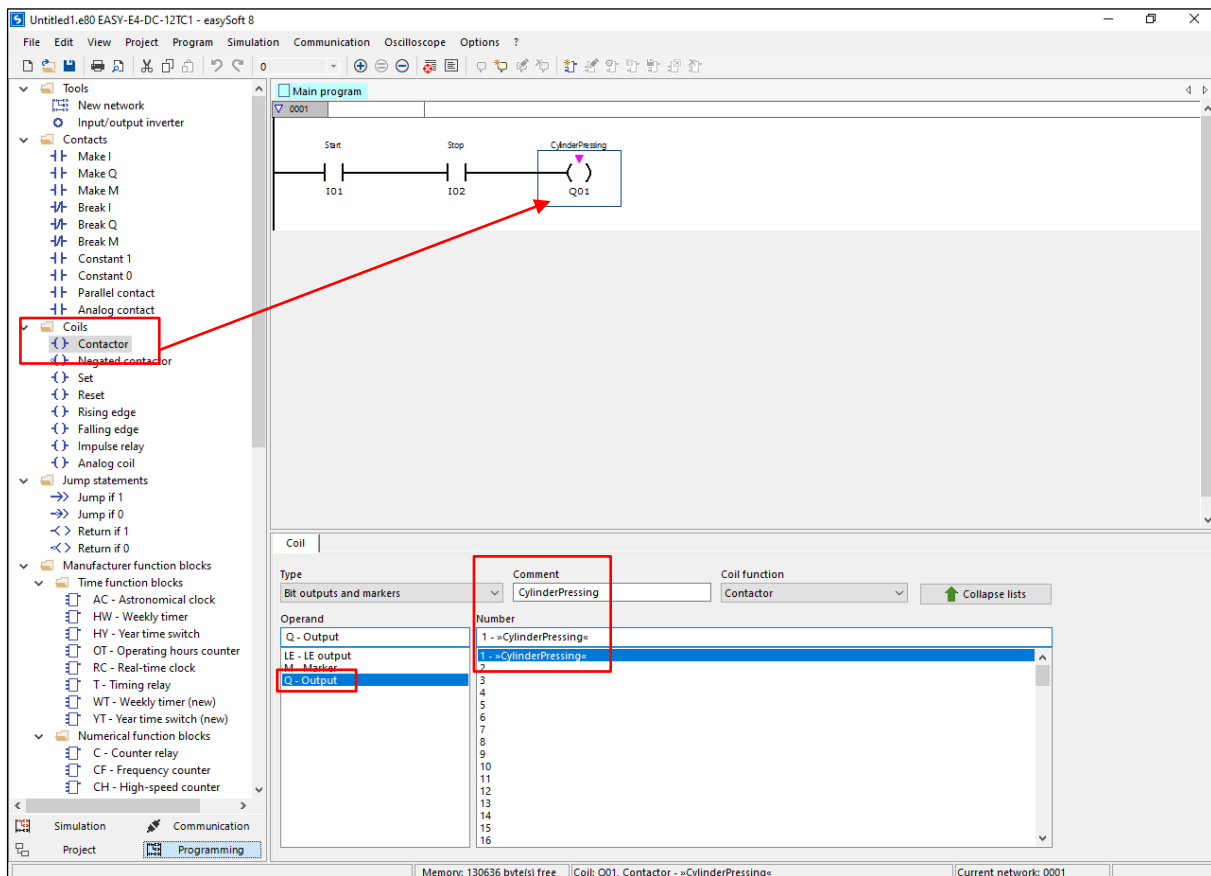


Fig. 10

The output can now be controlled. The outputs, or 'Contactors,' are available under 'Coils'.

- Drag a 'Contactor' onto the worksheet and place it next to 'Stop'.
- Select 'Q1' and give it the name 'CylinderPressing'.

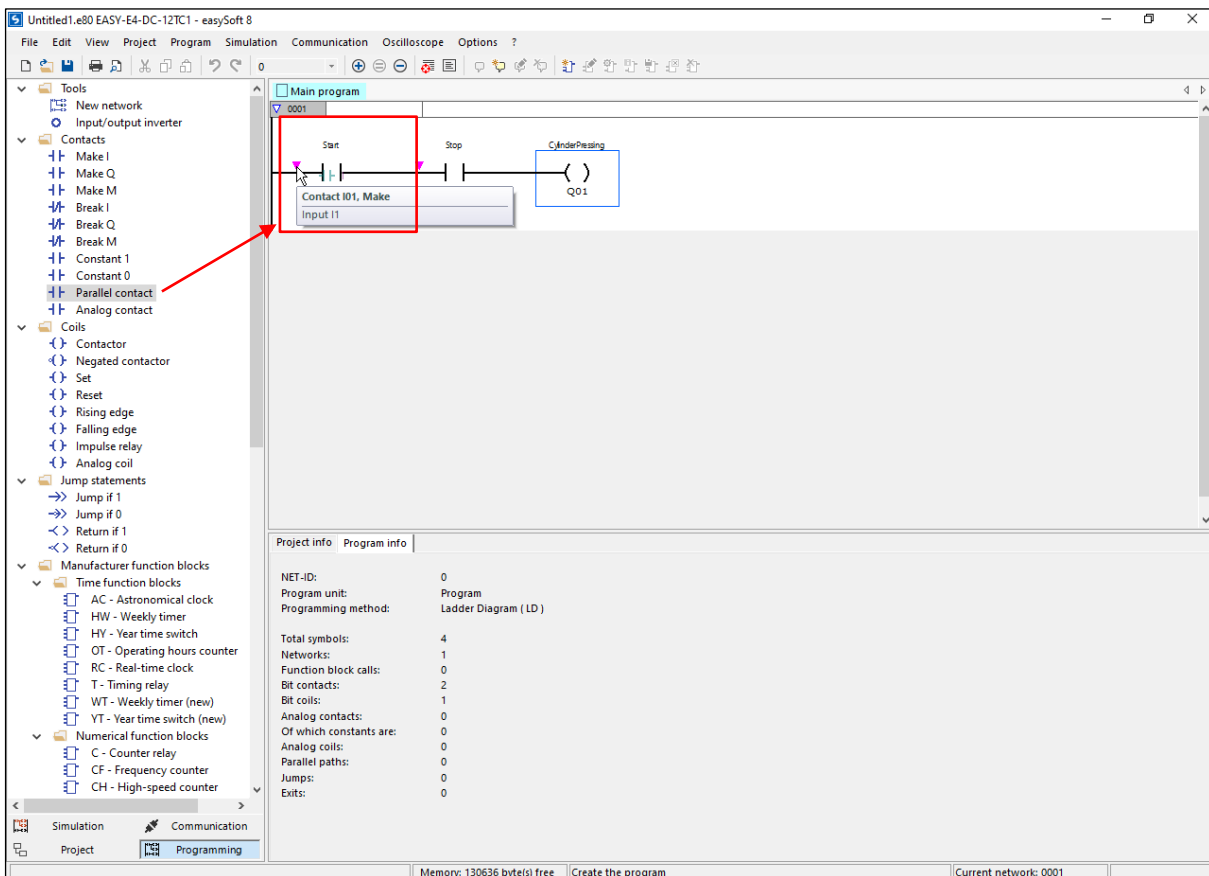


Fig. 11

The physical 'Stop' push button is always implemented with a break contact for cable break safety. In this example, there will therefore always be a 24V signal on input I2. It must therefore be programmed as a 'make contact' in the program.

If input I1 is made high using the 'Start' button, then output Q1 also becomes high. However, if 'Start' goes low, then Q1 also drops out again. A self-holding contact of Q1 must therefore still be placed across 'Start'.

- Drag a "Parallel contact" to the workspace and place it on the pink triangle to the left of "Start".

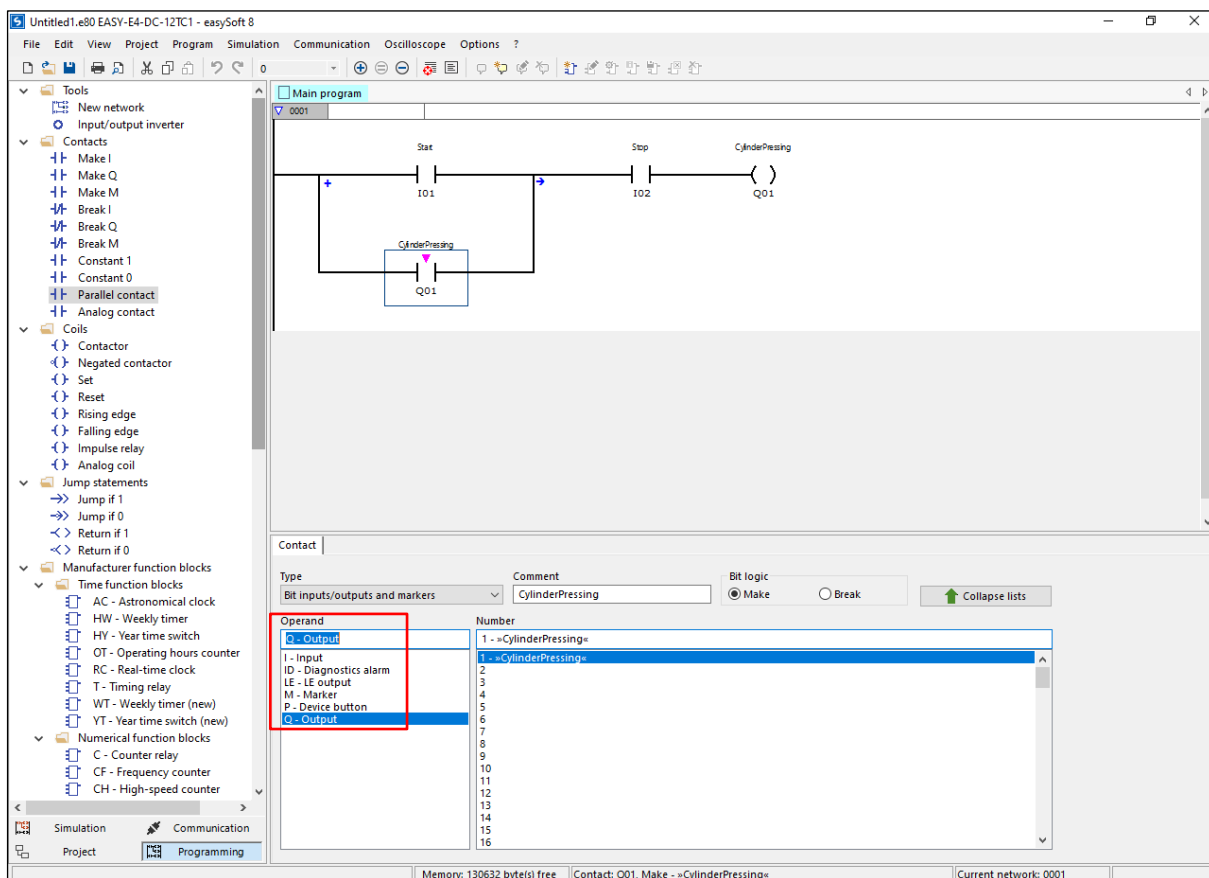


Fig. 12

By default, the contact is assigned the operand I1.

- Select "Q-Output" at "Operand".

Now the Q1 contact with the name 'CylinderPressing' is placed in parallel across 'Start'.

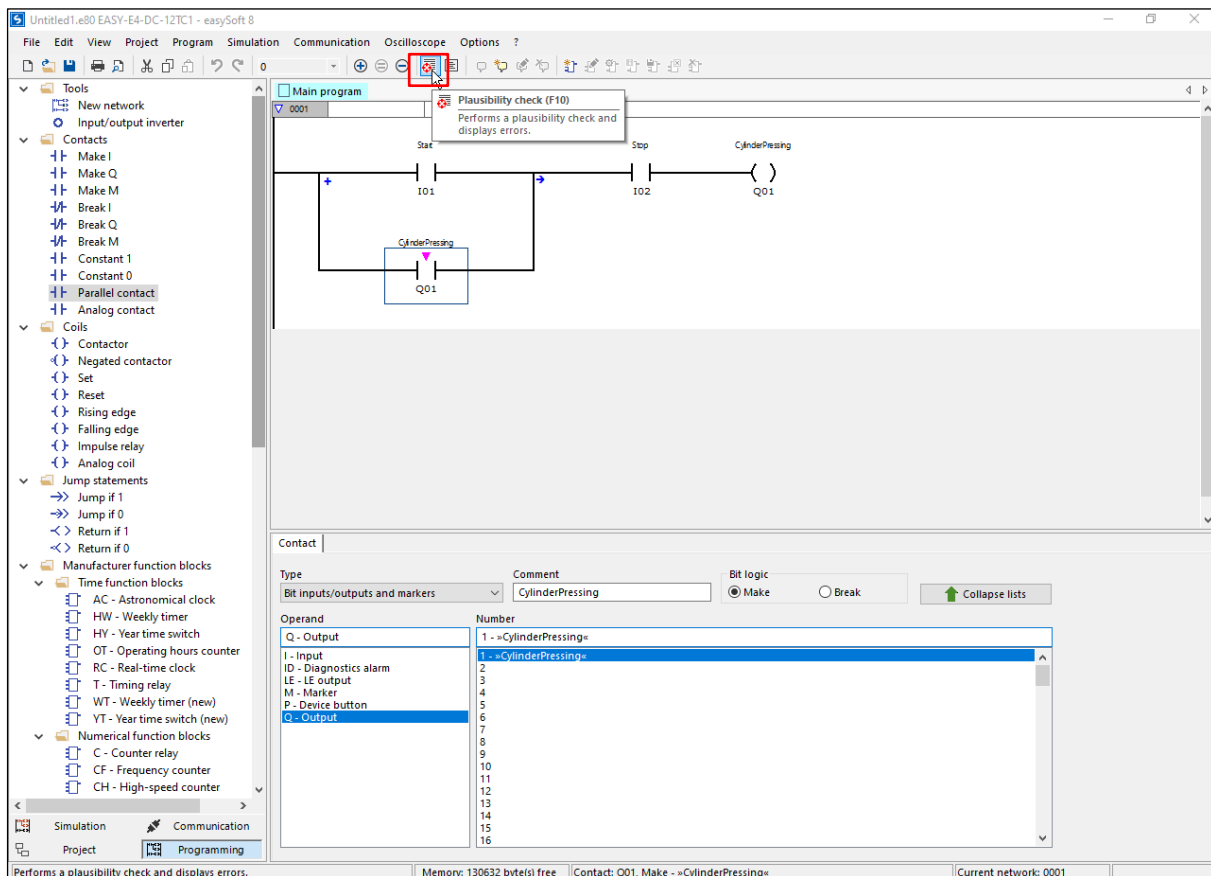


Fig. 13

The program is now ready and can be checked for errors.

- Click on "Plausibility check (F10)".

If no errors are found, the window appears as shown here.

"Plausibility check (F10)" can also be found under the "Project" menu.

## 2.2 Save program

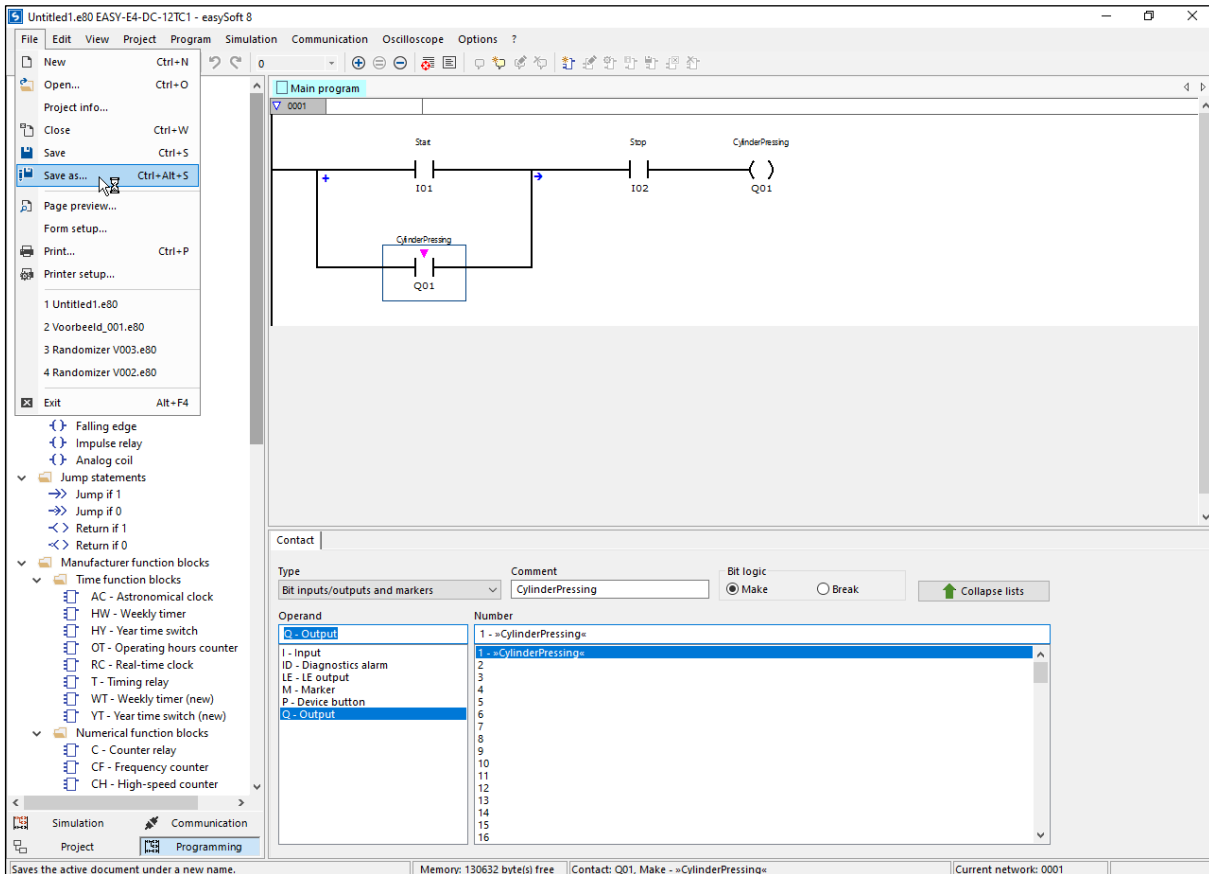


Fig. 14

It is advisable to save the program regularly.

- First, create a folder on the desktop with the name 'EasyTraining'.
- Go to "File" in the menu bar and select "Save as...".

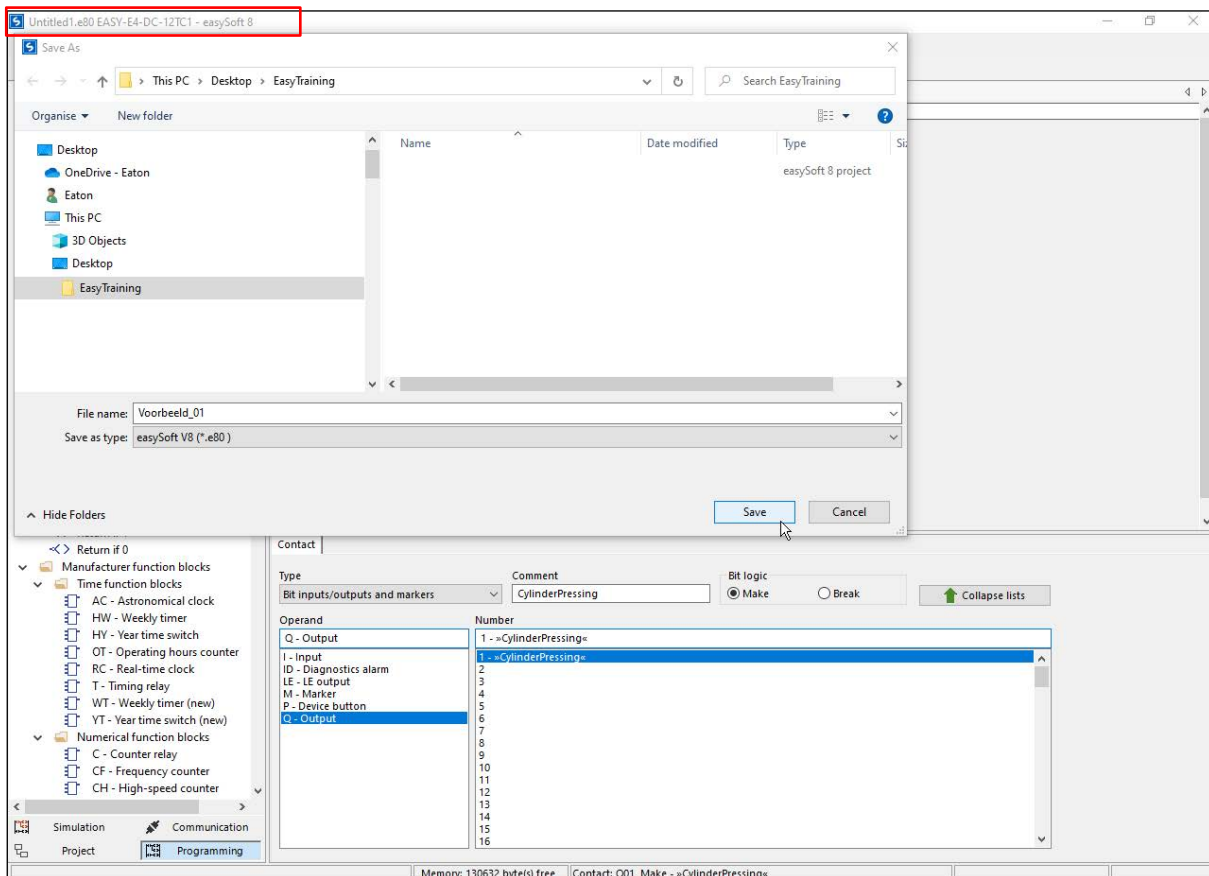
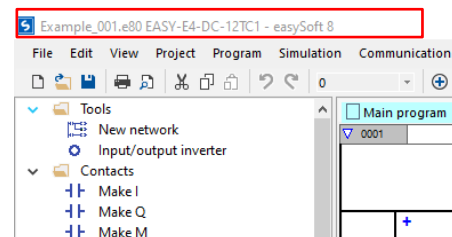


Fig. 15

- ▶ Select the folder 'EasyTraining' on the desktop.
- ▶ Give the program the name 'Example\_001'.
- ▶ And click on "Save".

The program has now been saved.

At the top of the screen, the name of the screen has changed from 'Untitled' to 'Example\_001.e80'.



## 2.3 Simulation

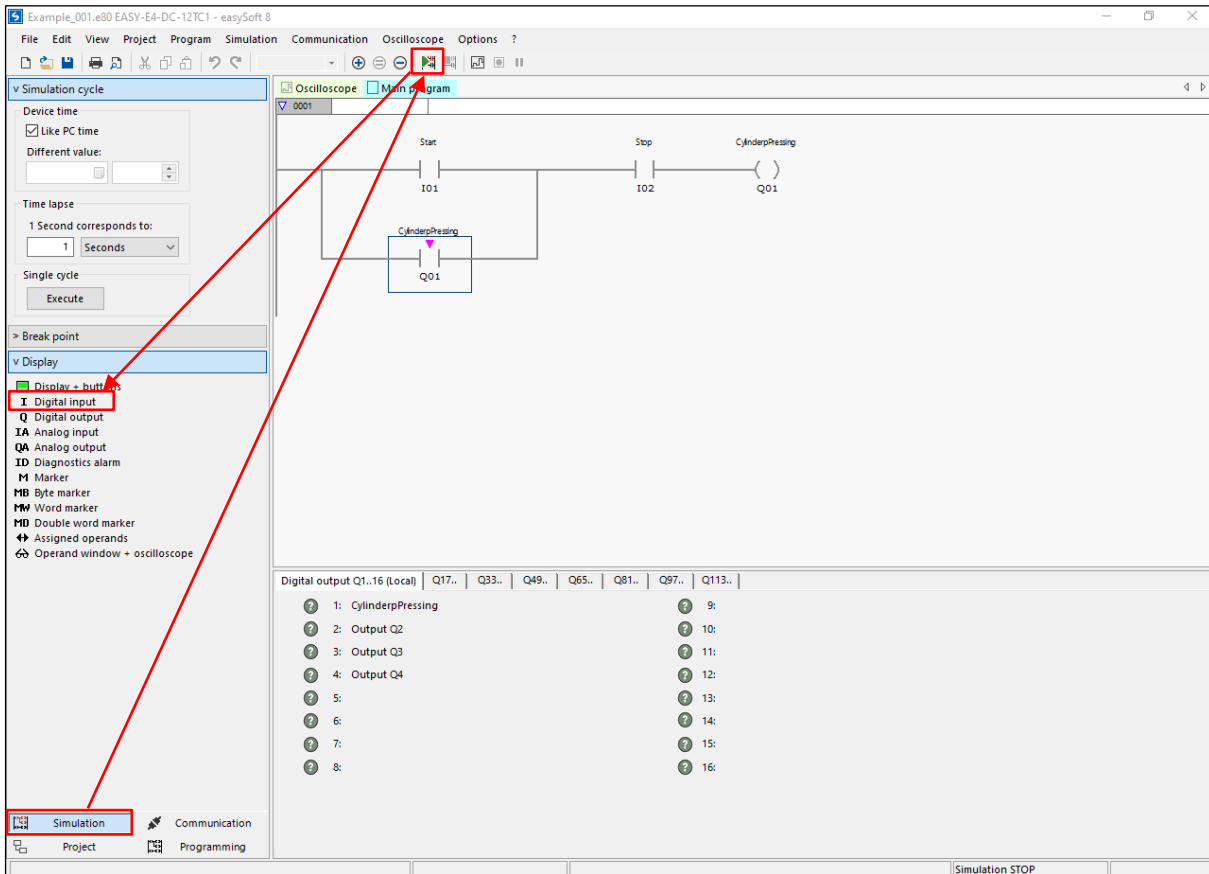


Fig. 16

Once the program has been checked and there are no programming errors, the program can be simulated to see if it operates as desired.

- ▶ Go to the "Simulation" menu. Click on "
- ▶ Start simulation (F9)" in the menu bar.
- ▶ Open the "Display" menu in the toolbox and select "I Digital input".

The inputs are now shown with the corresponding texts.

The closed circuits light up red.

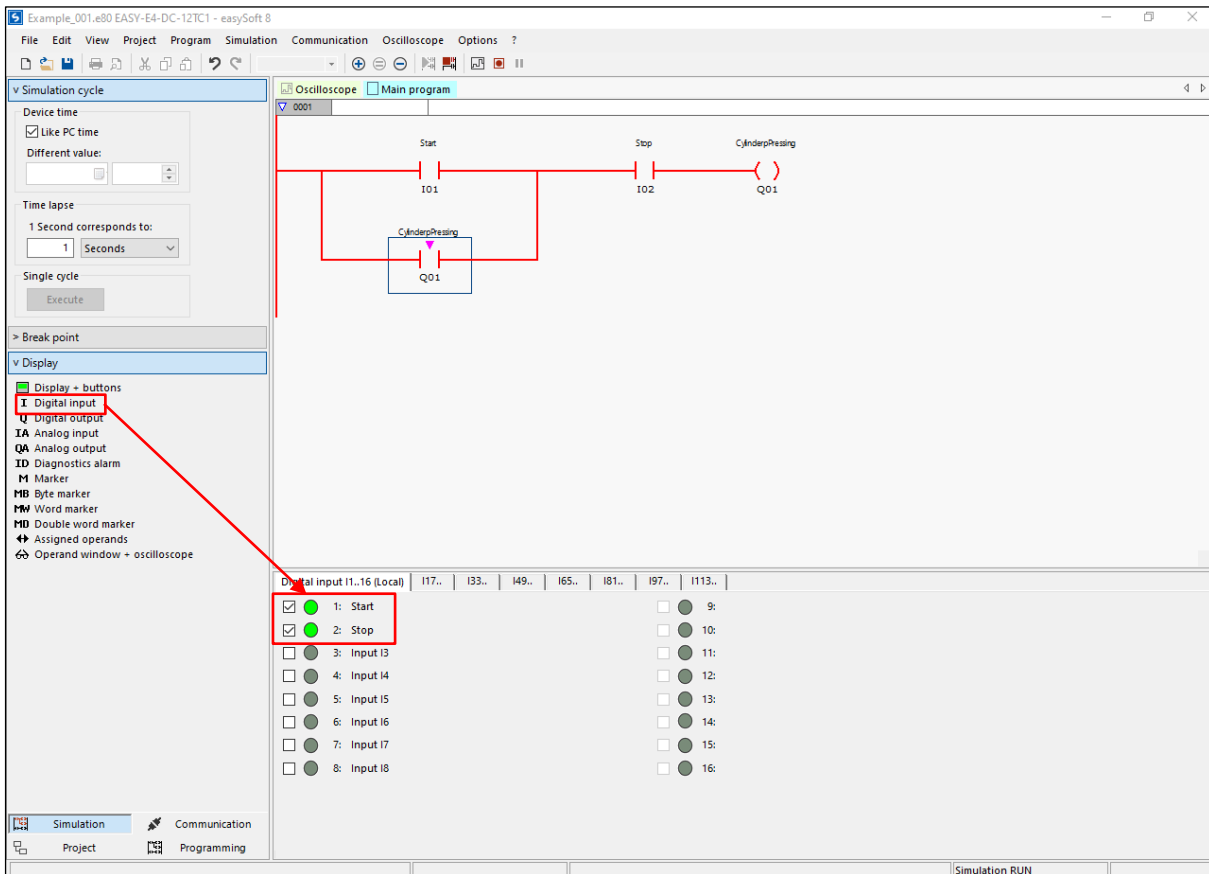


Fig. 17

- Make the 'Start' and 'Stop' inputs high by checking the box next to the input.

It can be seen in the LD (Ladder Diagram) that the entire circuit is closed, after which the 'CylinderPressing' output becomes high. The (auxiliary) contact of the 'CylinderPressing' output also becomes high.

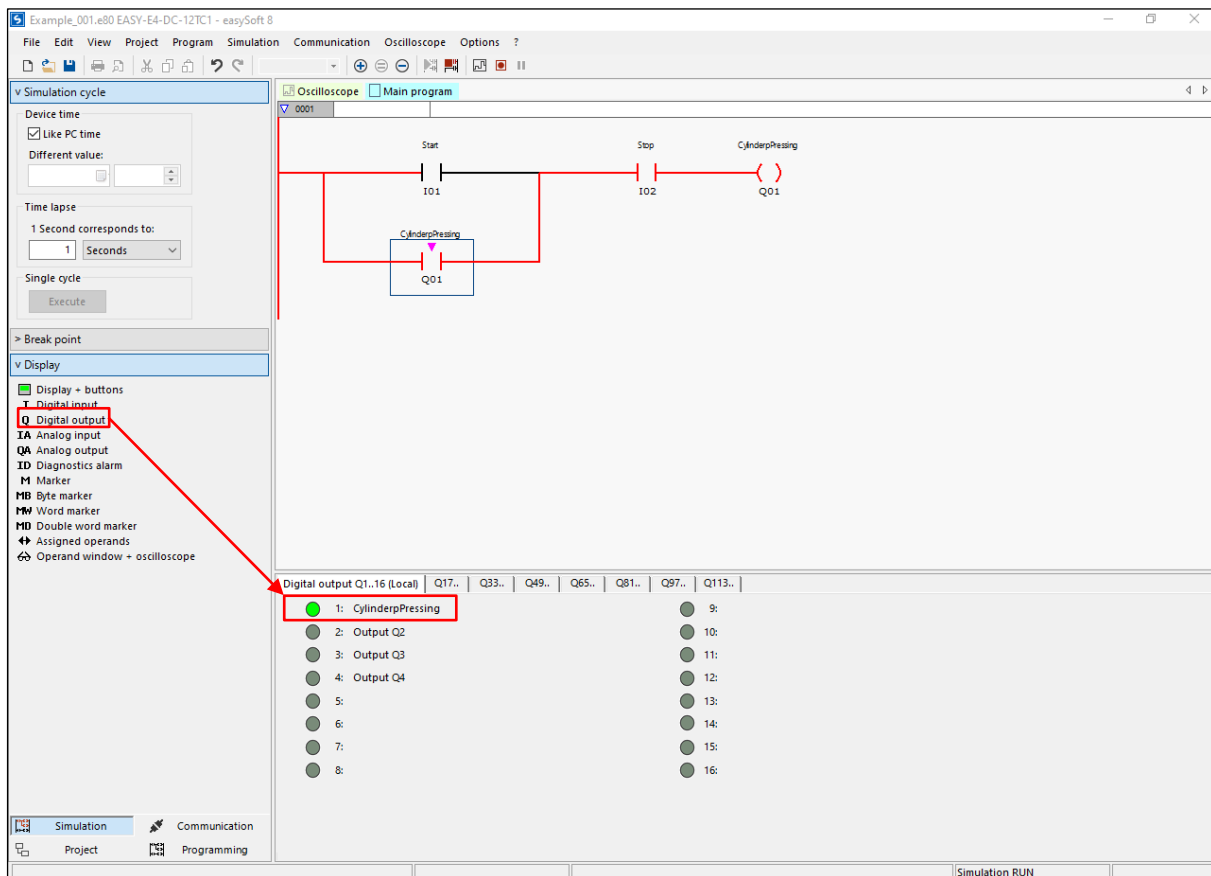


Fig. 18

- Make 'Start' low by removing the check mark next to 'Start' again.

The output Q01 remains high.

- To see the status of all outputs, go to 'Q Digital output' under 'Display'.

Here it can be seen that output Q01 'CylinderPressing' is high.

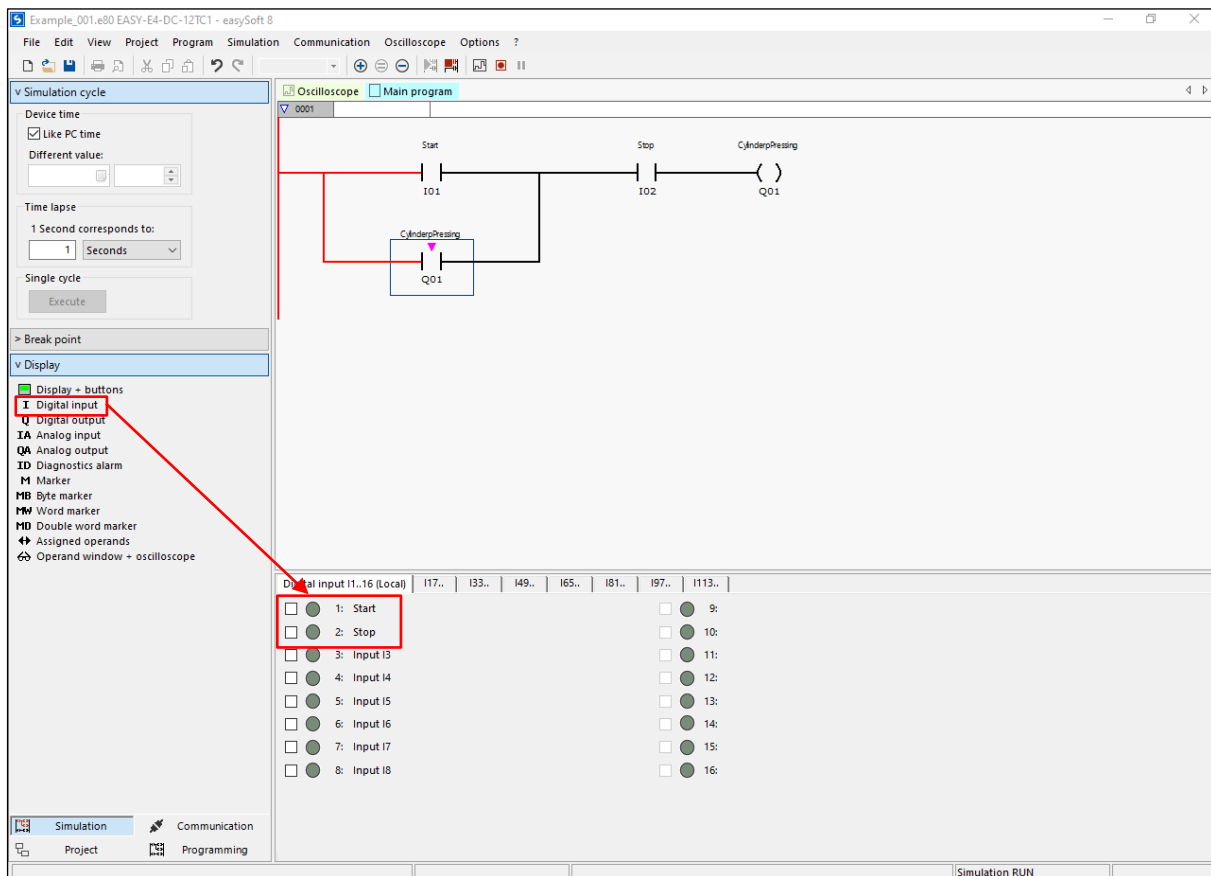


Fig. 19

- Go back to "Digital input" under "Display".
- Now also make 'Stop' low.

Here it can be seen that output Q01, 'CylinderPressing,' becomes low again.

## 2.4 Send program to easyE4

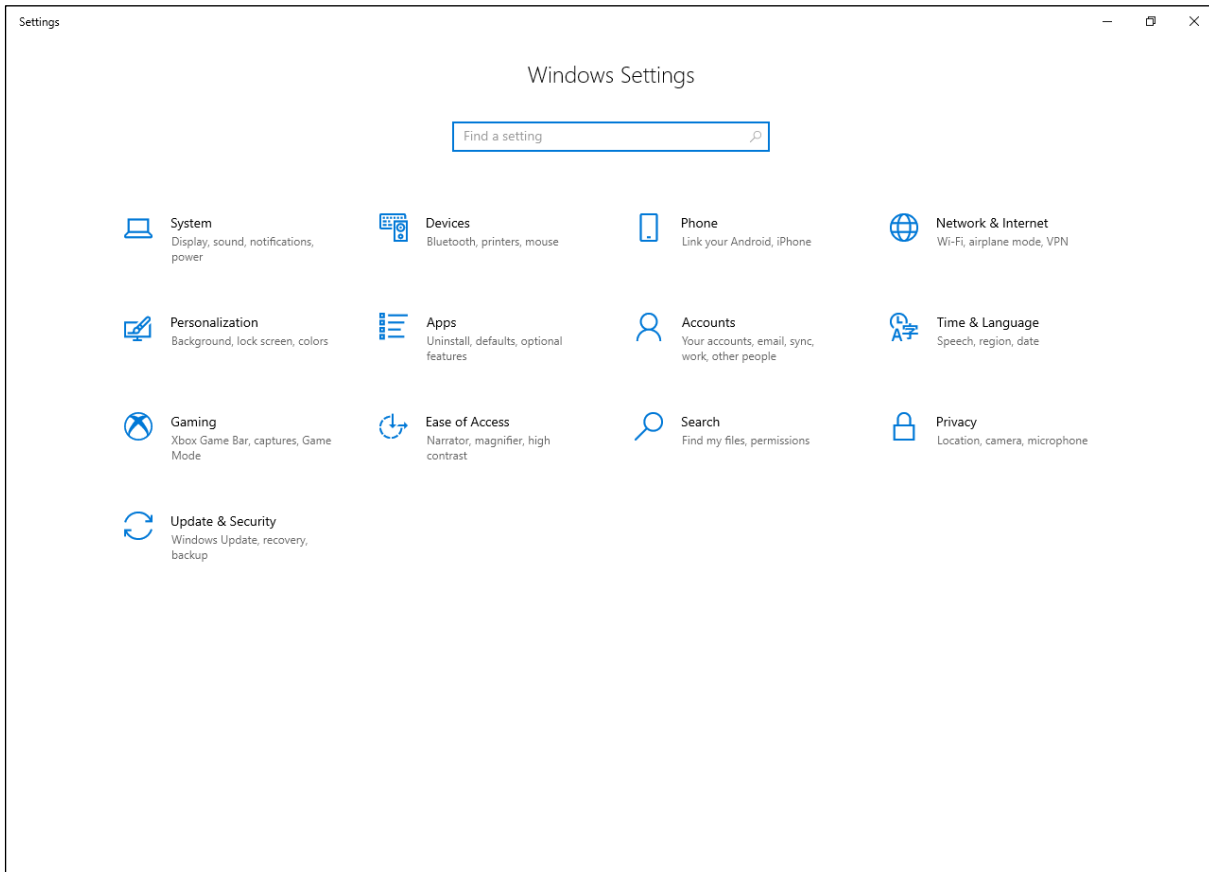


Fig. 20

The program has been tested and is ready to be transferred to the easyE4.

Before a program can be sent, the PC and the easyE4 must be able to communicate with each other. If an easyE4 is new, it will have been assigned a random IP address. This address is shown at the bottom of the display.

The PC must therefore also be set to 'Obtain an IP address automatically'.

- Open on the PC via "Settings" → "Network & Internet".

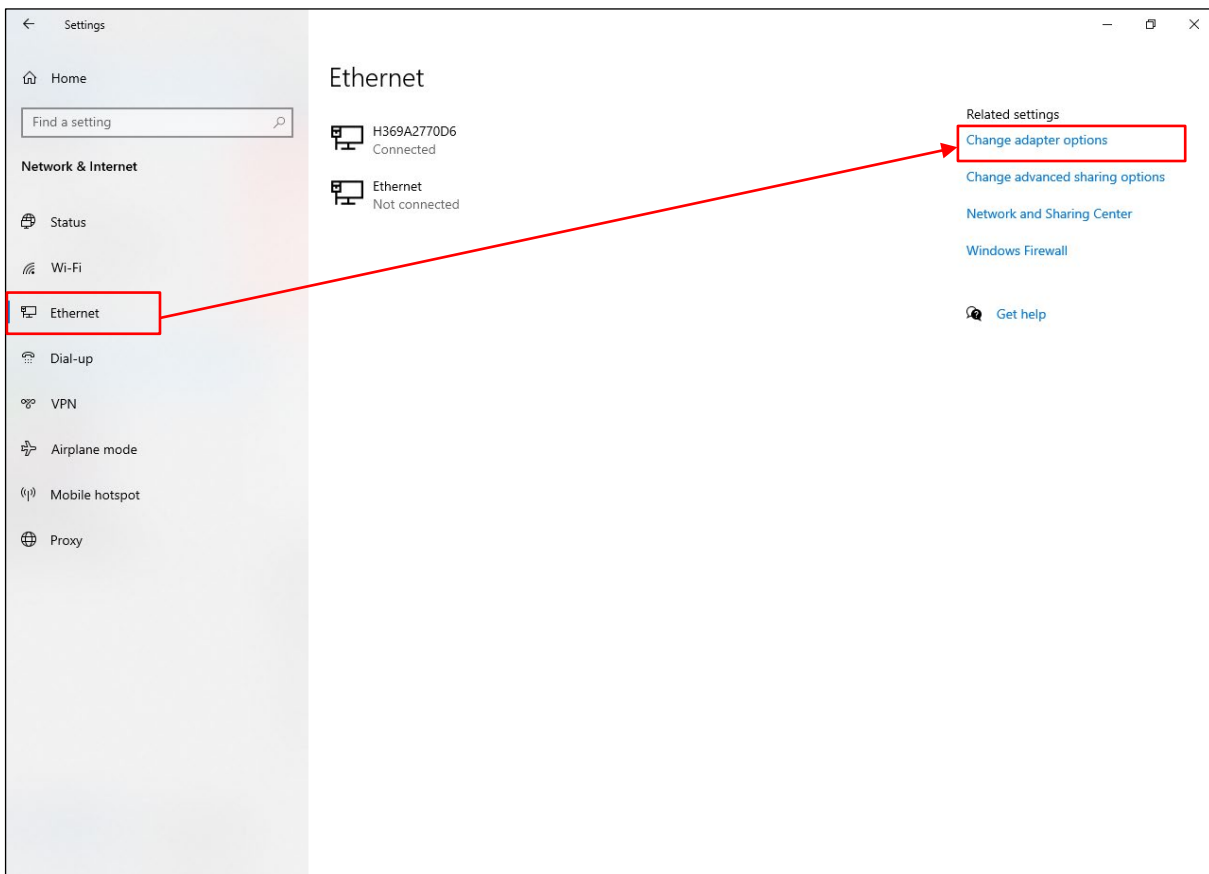


Fig. 21

- ▶ Select “Change adapter options” under “Ethernet”

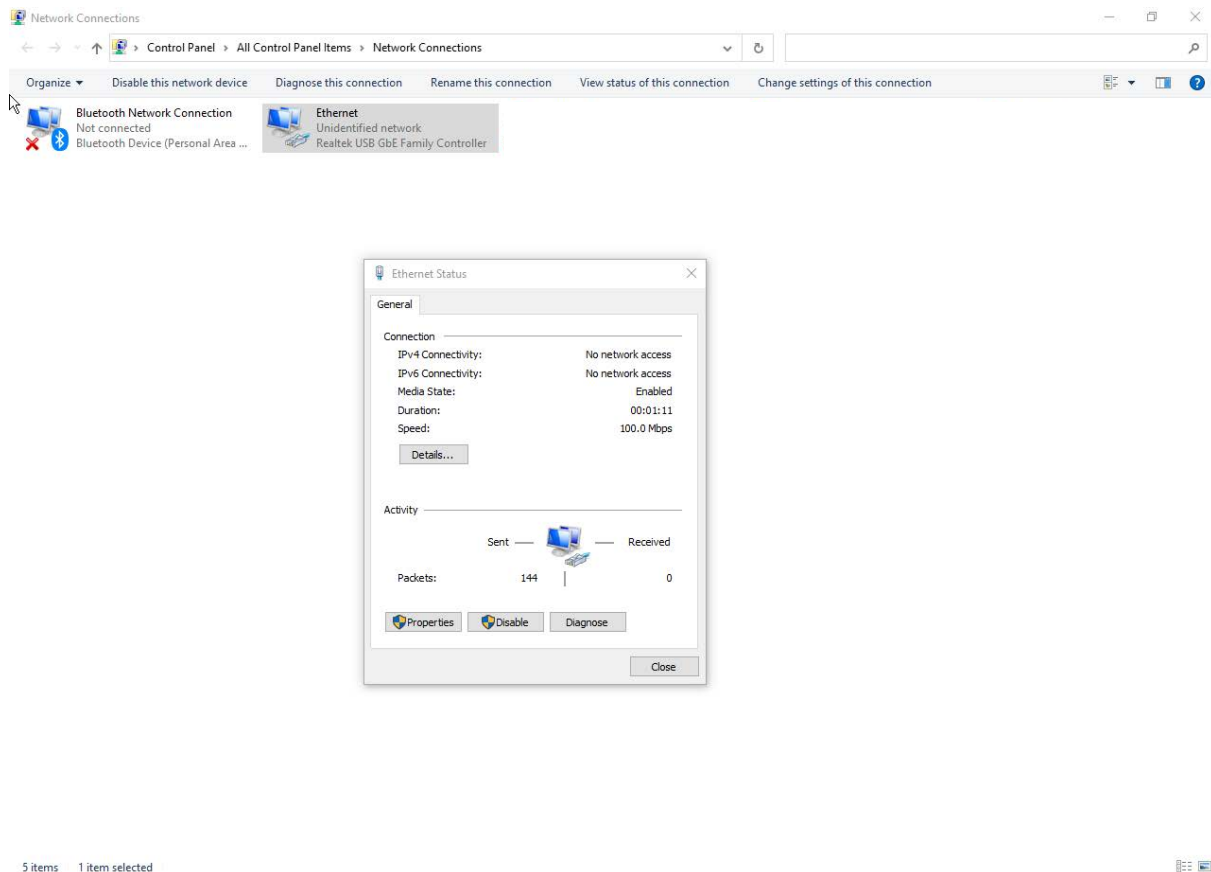


Fig. 22

- Double-click on the adapter to be used to open it.

This example has the name "Ethernet". The "Ethernet Status" window is opened.

- Here, click on "Properties".

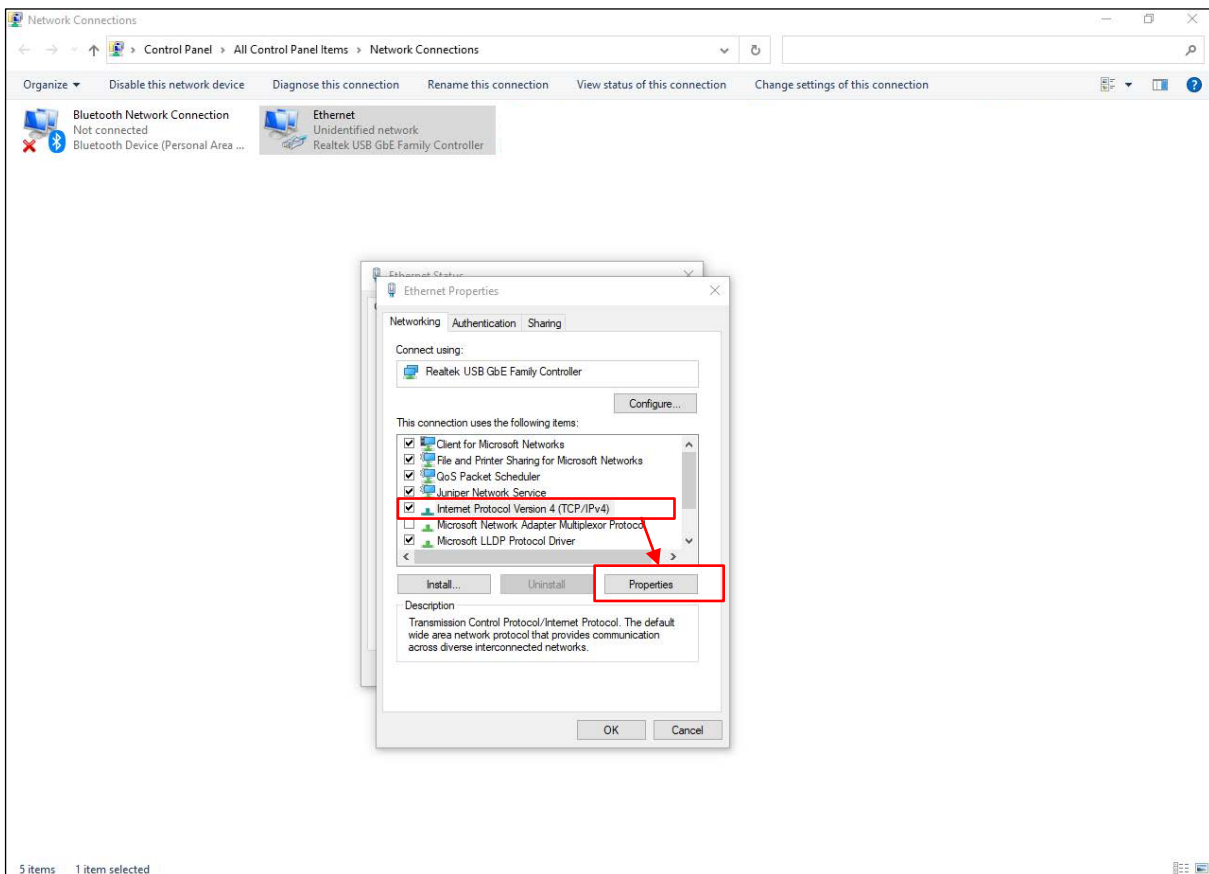


Fig. 23

- Select 'Internet Protocol Version 4 (TCP/IPv4)' here and click on 'Properties'.

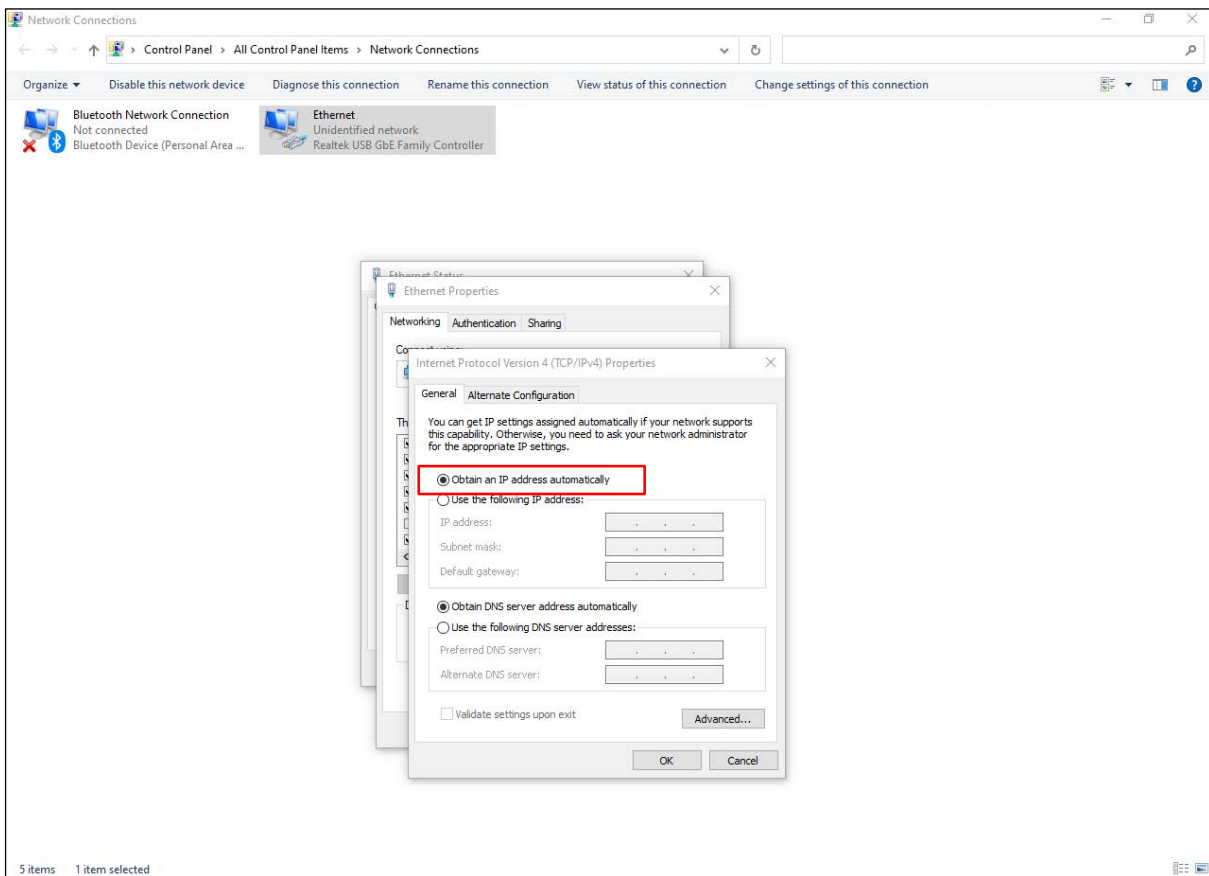


Fig. 24

- ▶ In this window, 'Obtain an IP address automatically' must be selected.
- ▶ Confirm this with 'OK,' after which all of this can be closed.

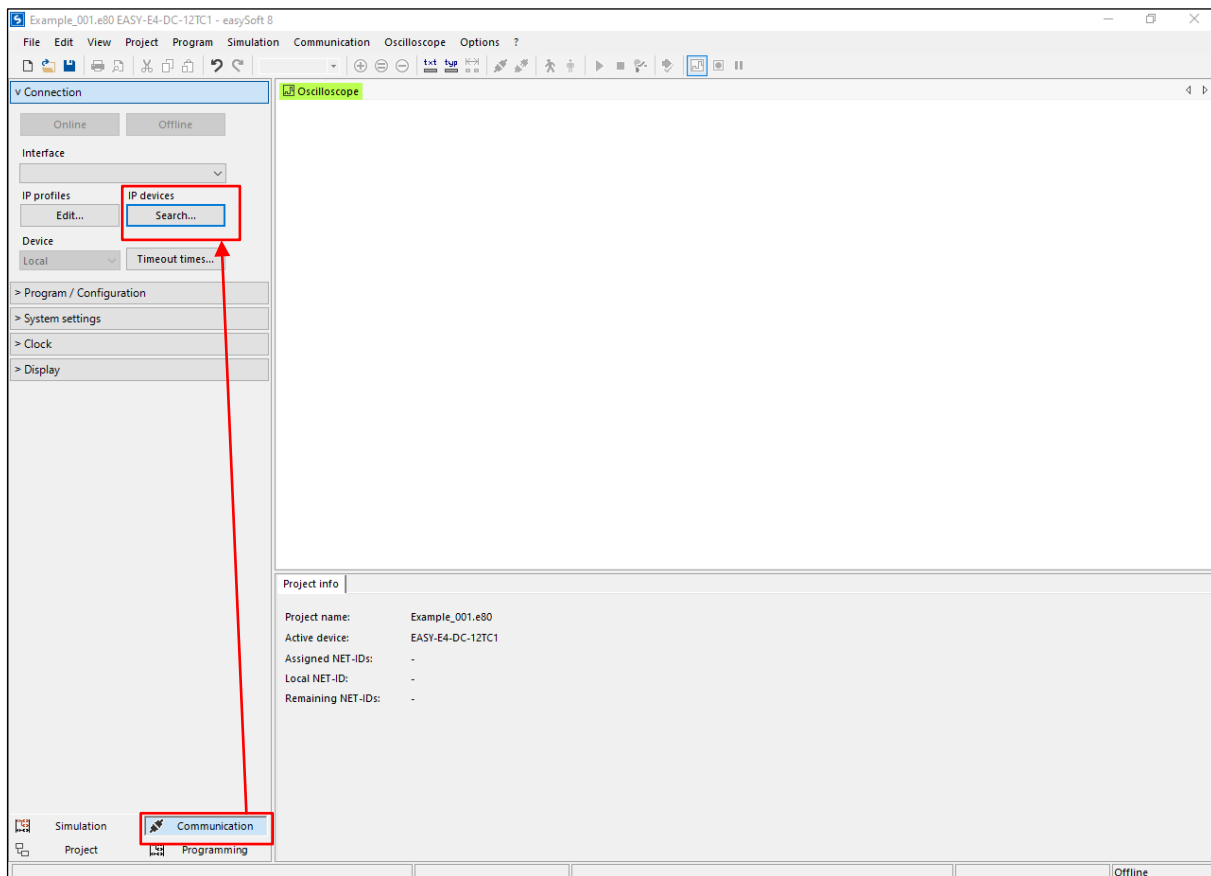


Fig. 25

- In easySoft 8, go to "Communication".
- And select "Search".

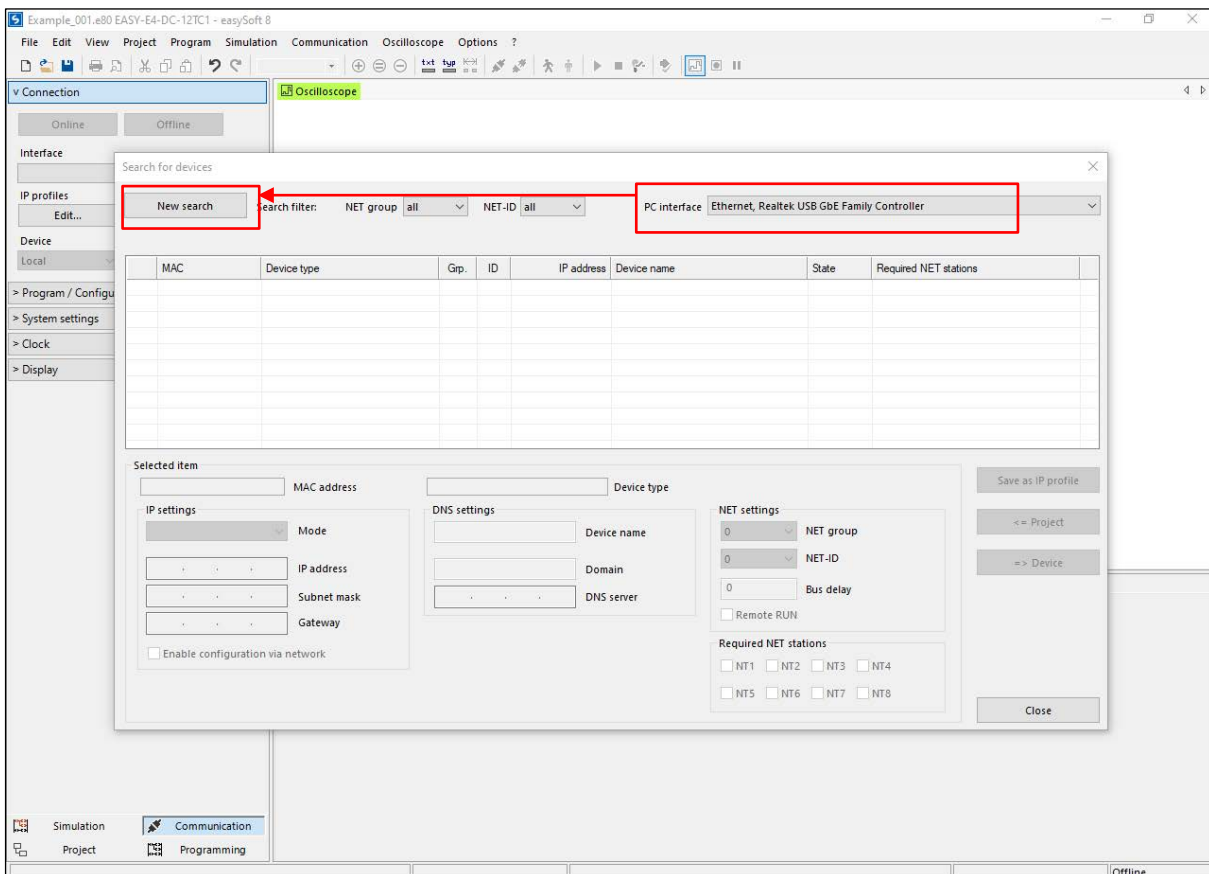


Fig. 26

- Ensure that the correct PC interface is selected and then click on 'New search'.

The software will now search for the easyE4 that is connected.

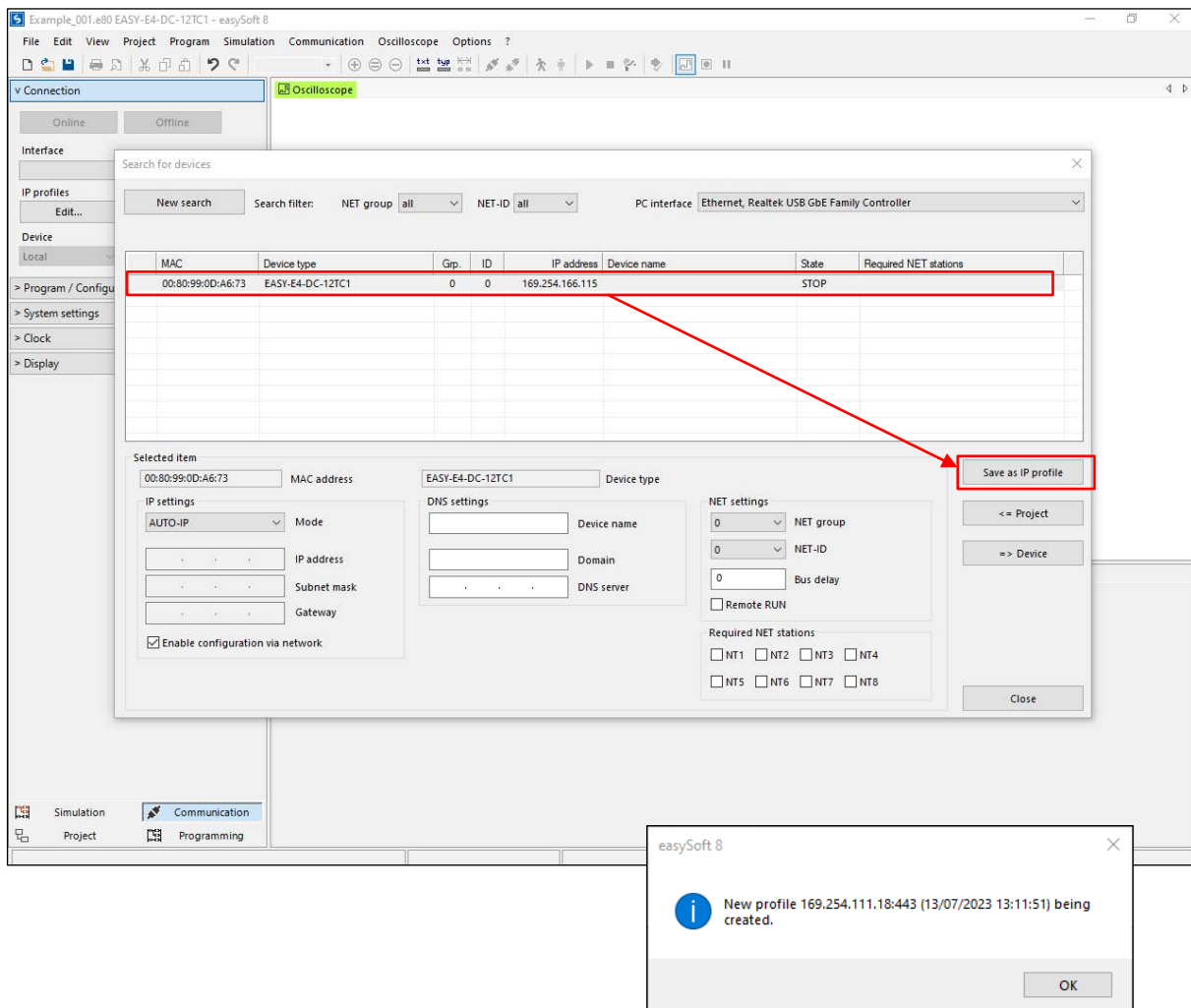


Fig. 27

The screen will now display the MAC ID and the IP address of the connected easyE4.

The MAC ID is also stated on the front of the easyE4 housing.

- Select the found address and click on 'Save as IP profile'

A window appears with the notification that a new profile has been created.

- Confirm with OK.
- Click "Close" to close the window.

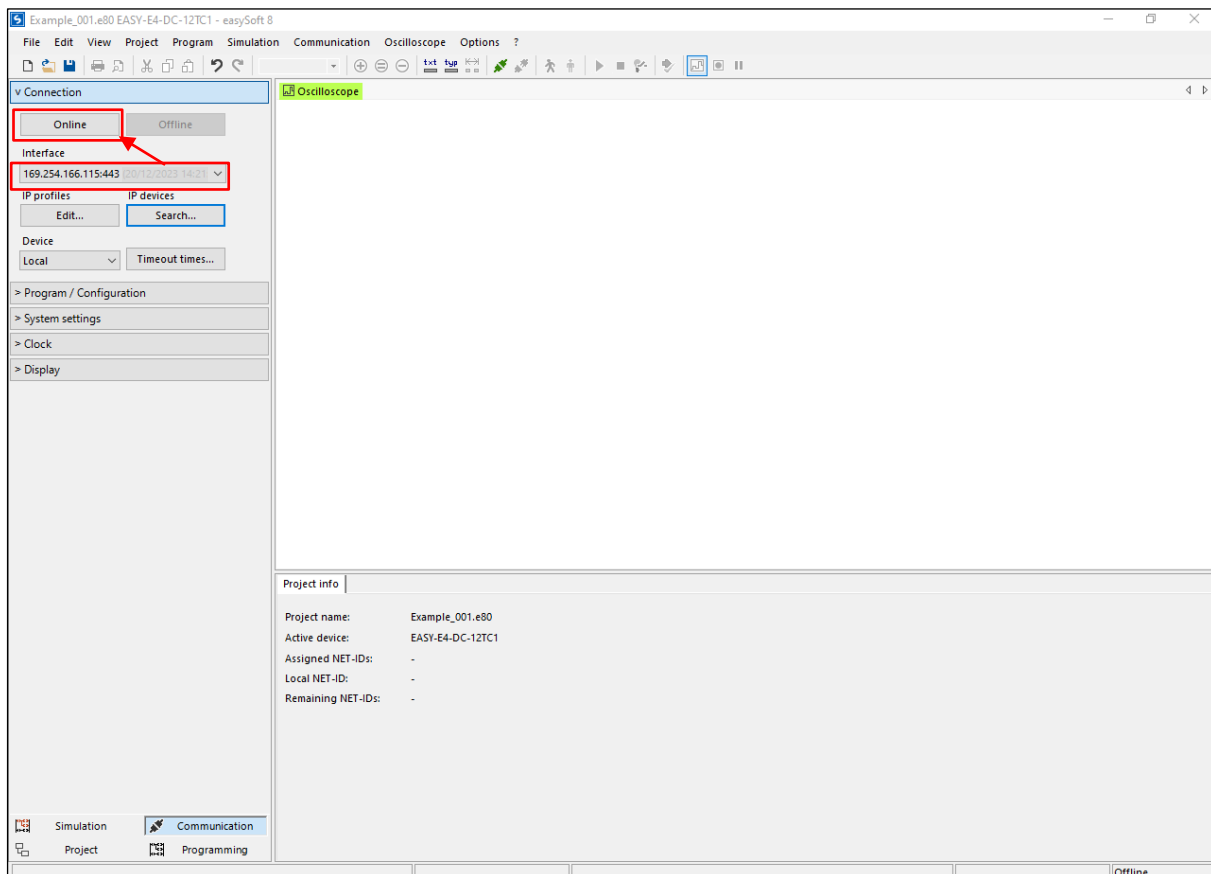


Fig. 28

The IP address that the easyE4 also has is now displayed under 'Interface'. This can be verified on the display of the easyE4 itself.

- Click on "Online" to establish the connection.

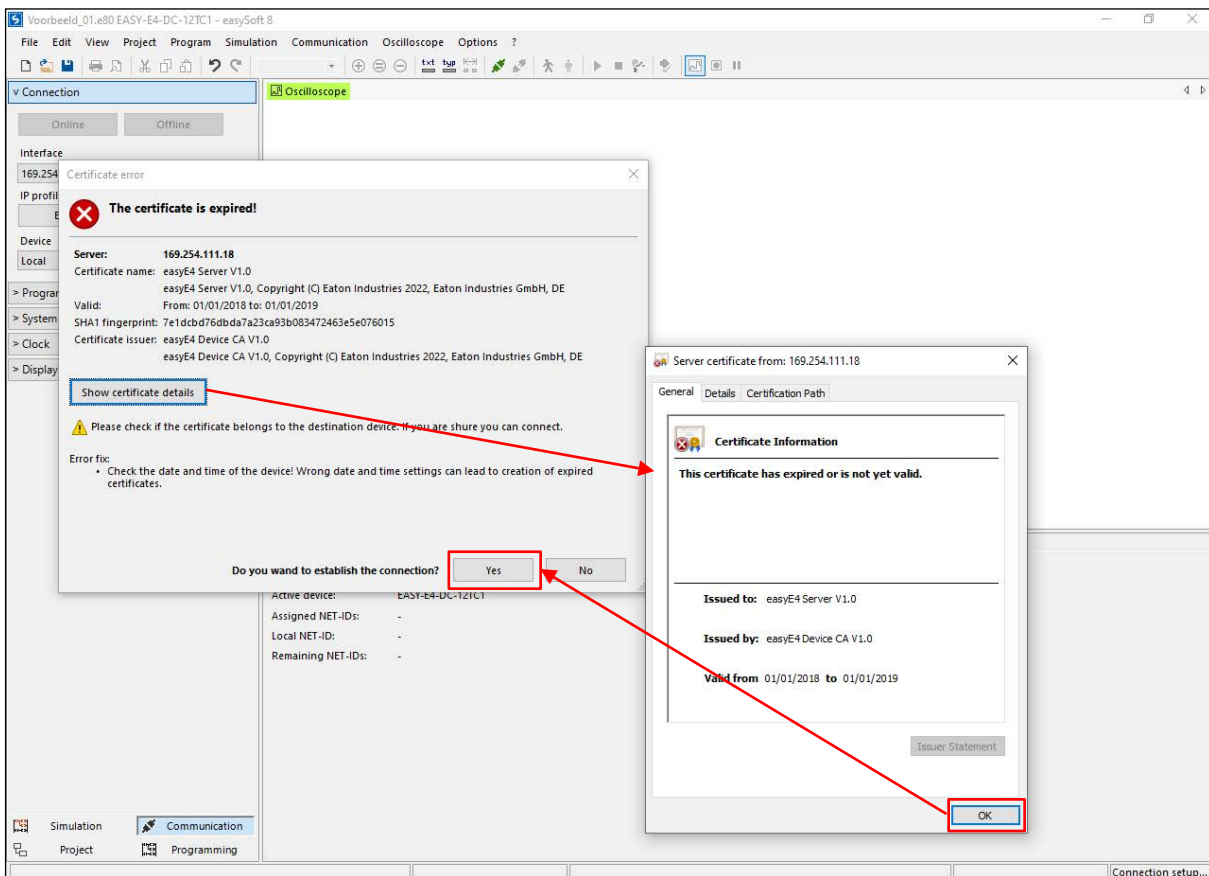


Fig. 29

A window appears with the notification that the certificate has expired.

- Click on "Show certificate details" for more information.

Here it can be seen that the date of the security certificate has expired or was never created.

- Open this window again by clicking on "OK".
- Confirm by clicking "Yes" that a connection must after all be established.

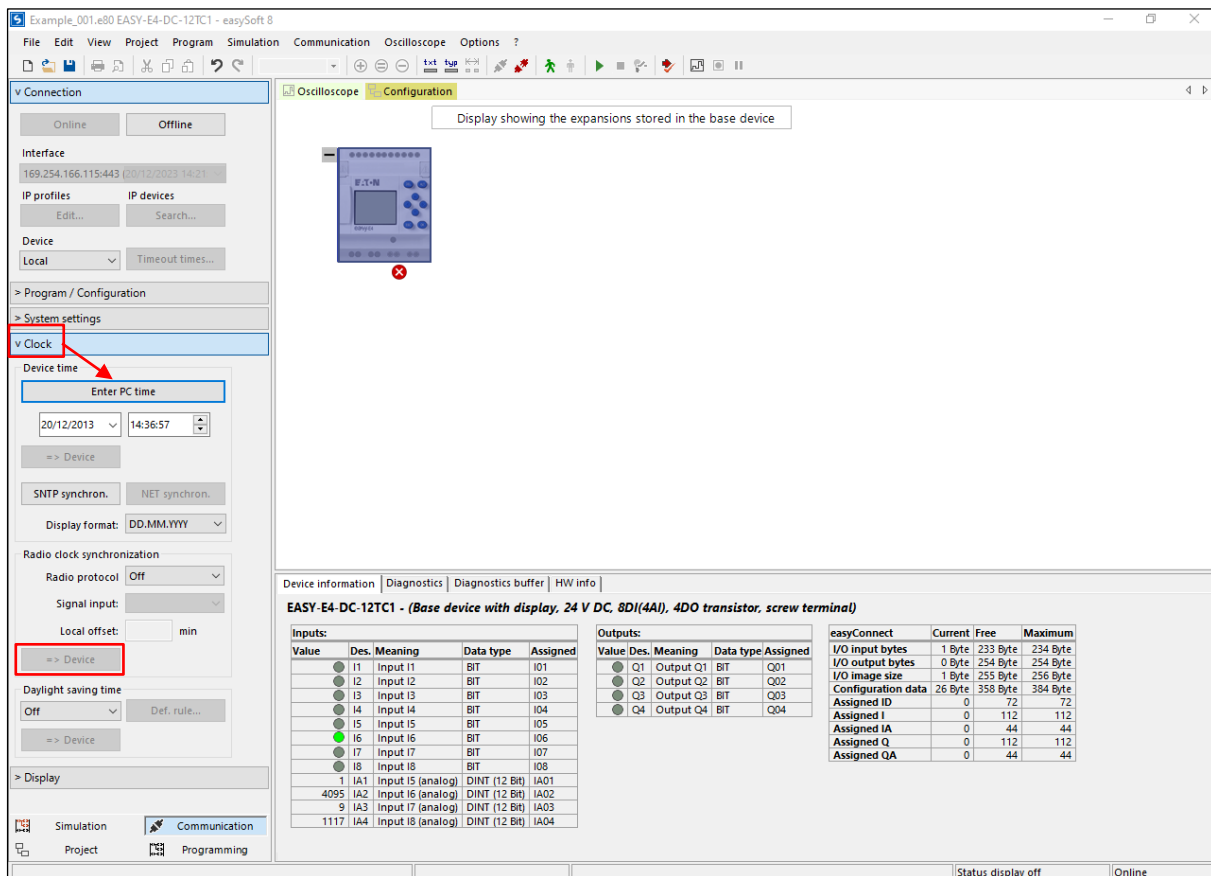


Fig. 30

This notification that the certificate has expired can be removed by setting the date and time of the easyE4 to the current time.

- Click "Clock" and select "Enter PC time".

The time of the PC is now taken over.

- Send the correct time to the easyE4 with the "=> Device" button.

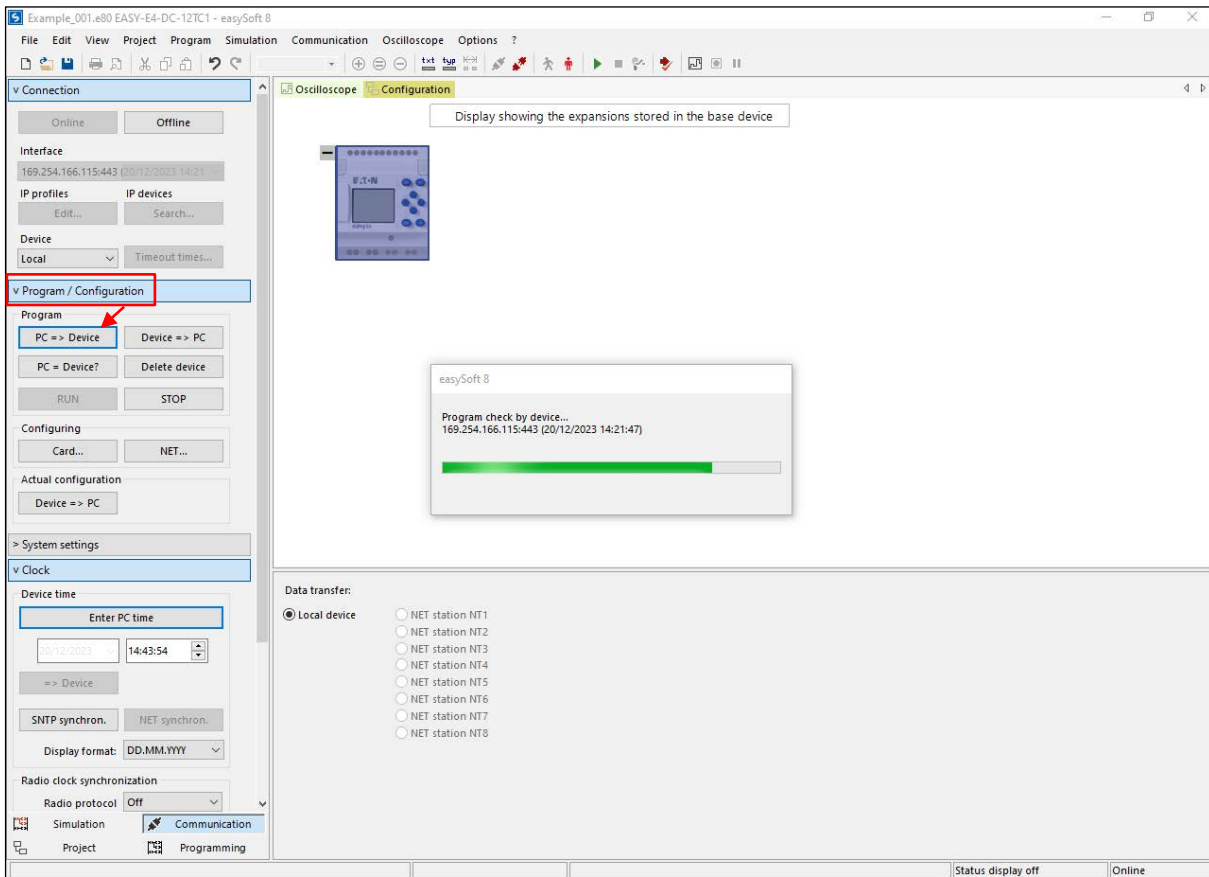


Fig. 31

- Go to 'Program / Configuration' and click on 'PC => Device'

The program will now be transferred to the easyE4.

If there is already an active program in the easyE4, the following message will be displayed. (next page).

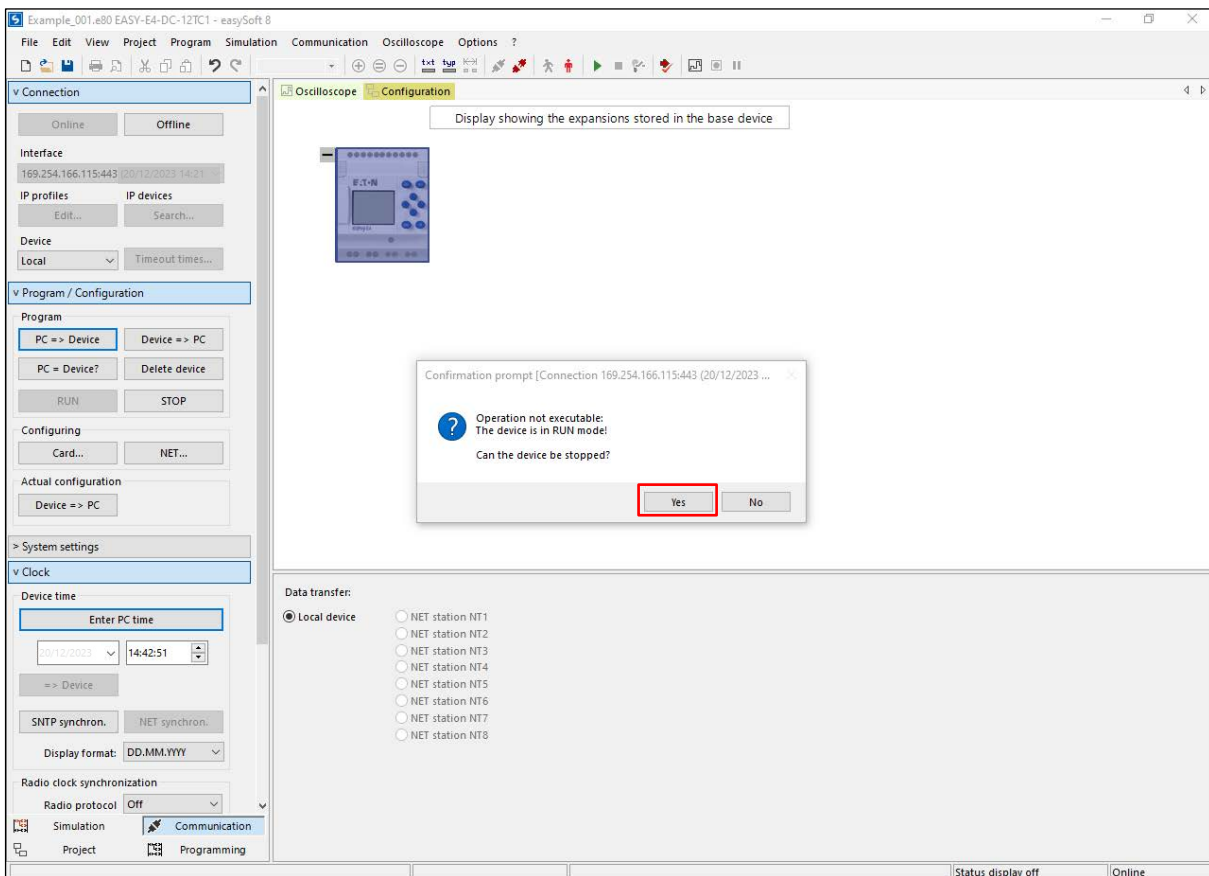


Fig. 32

Here the software indicates that transfer is not possible because the easyE4 is in 'RUN' mode. Because the easyE4 is not running in an operational process, the question 'Can the device be stopped?' can be answered "Yes". The easyE4 goes into STOP mode, after which the program is transferred.

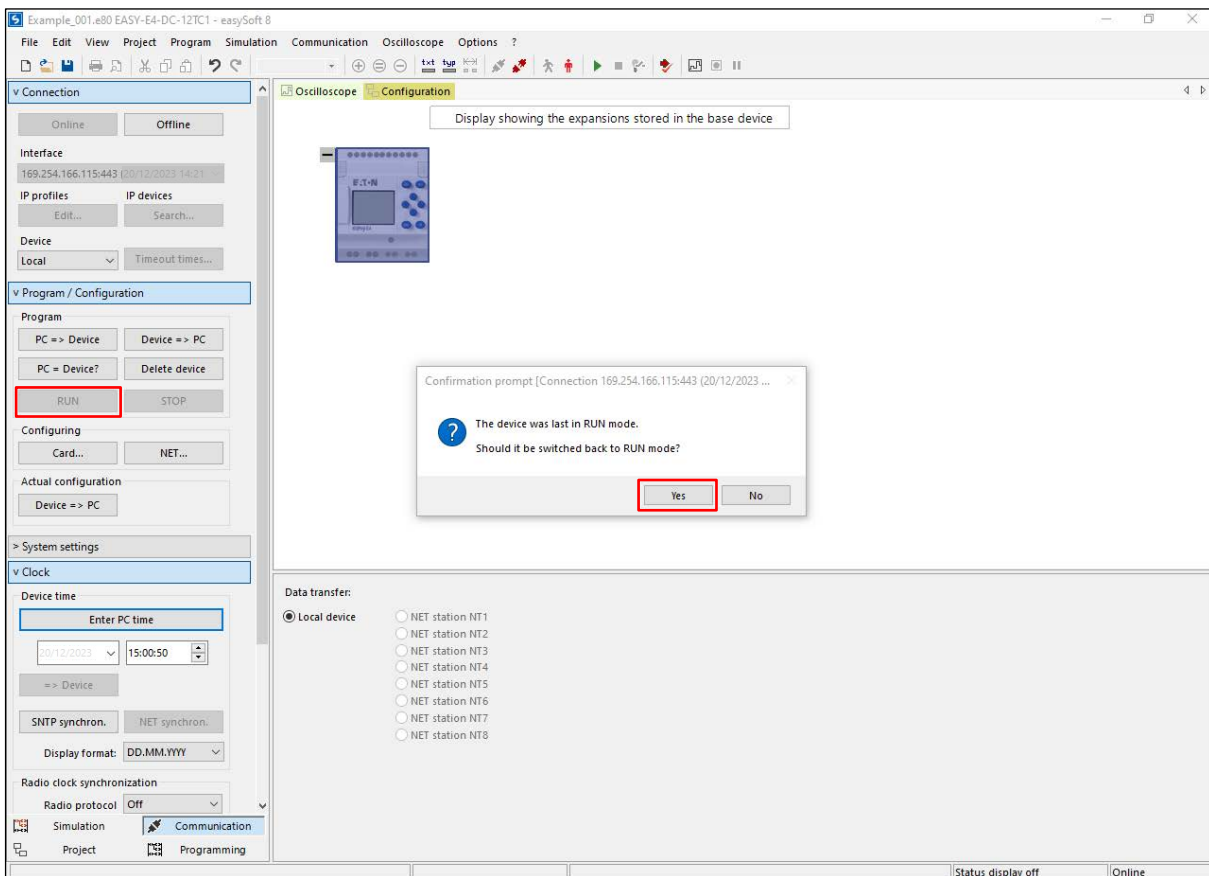


Fig. 33

After sending, the question appears whether the easyE4 should be started again. This can be answered "Yes".

If there was no program in the easyE4, then the easyE4 must be set to RUN mode using the 'RUN' button.

The program in the easyE4 has started and can be tested.

## 2.5 View online

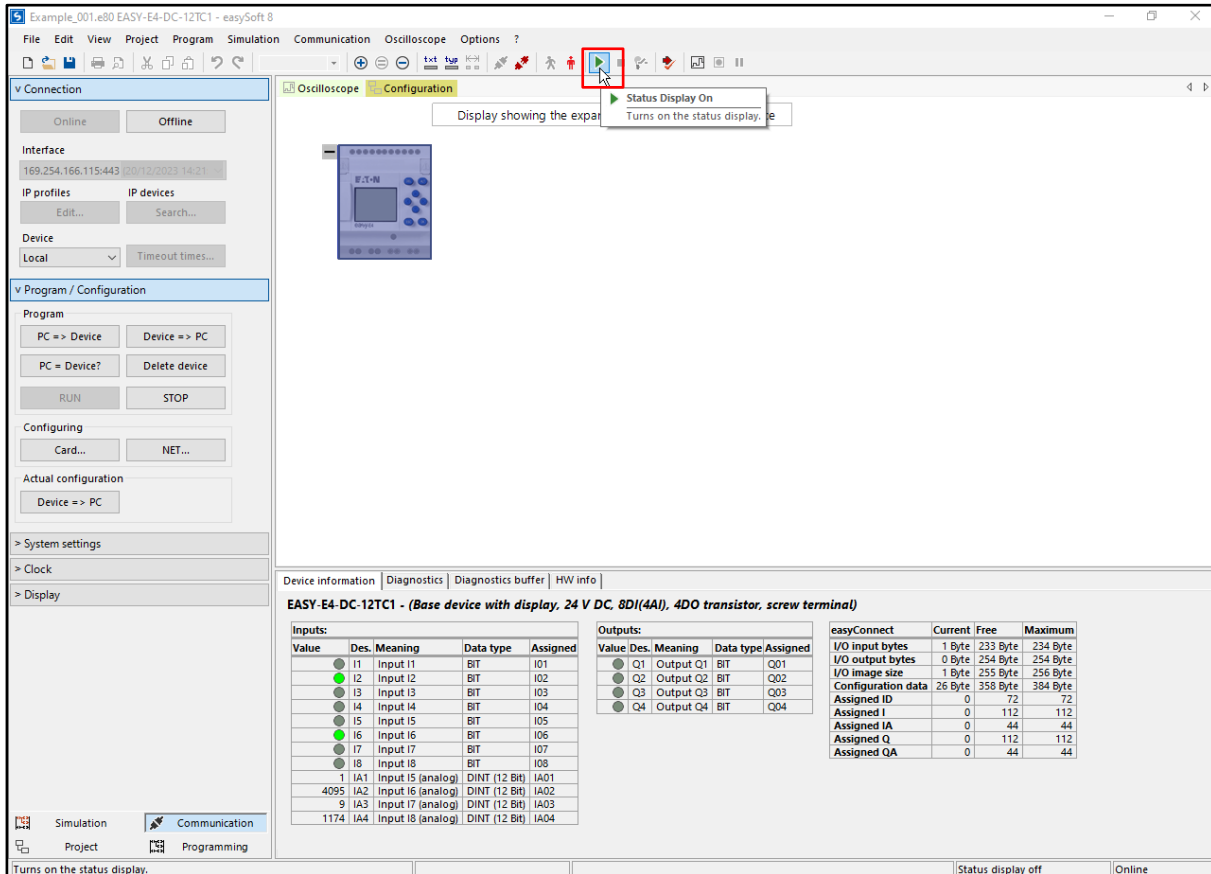


Fig. 34

Now that the connection has been established with the easyE4, you can view it online.

- For this, click on the "Status Display On" button.

The program is displayed, and the circuits that are active are colored red.

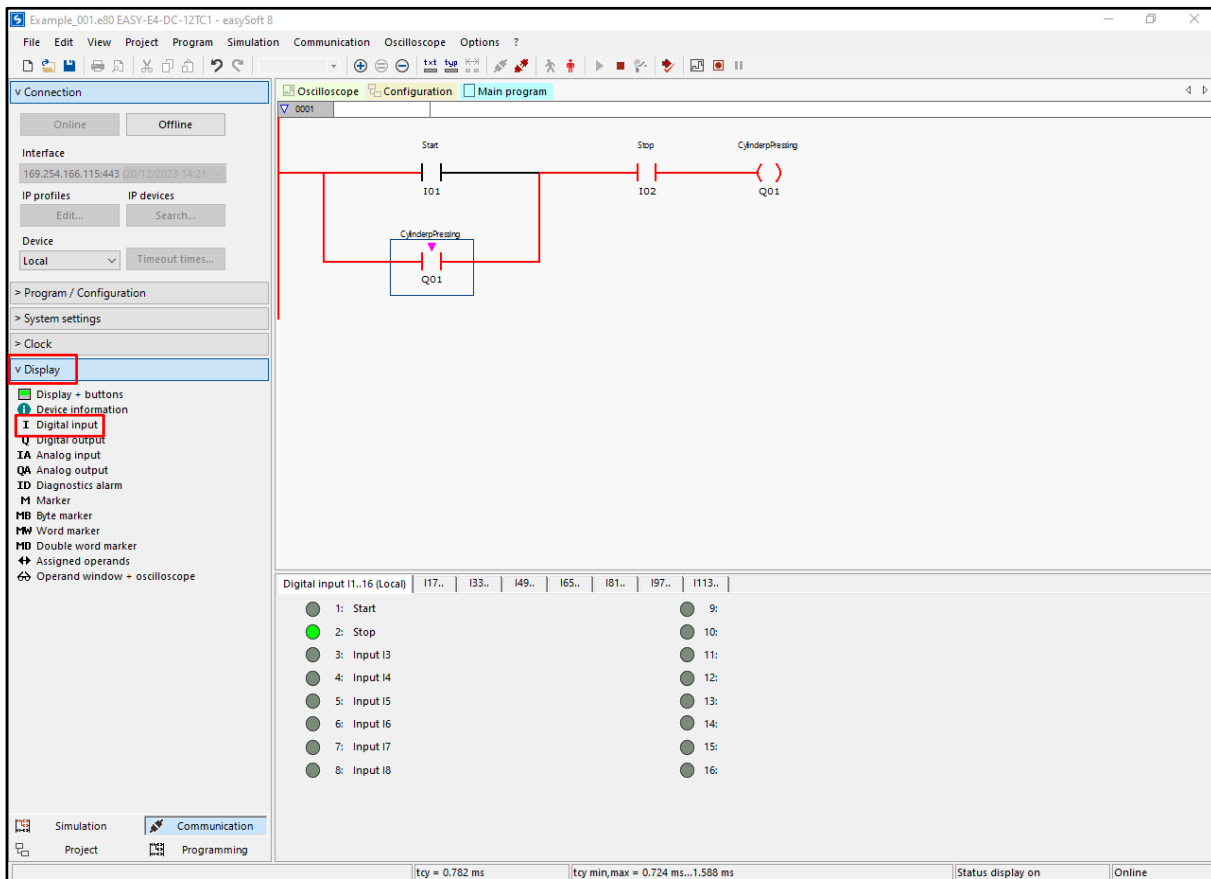


Fig. 35

Here above it can be seen that the 'CylinderPressing' output (Q1) is high, just like the 'Stop' input (I2).

- To be able to see the various inputs, go to 'Display' and select 'I Digital input'.
- Test the program by making the various inputs high.

## 2.6 Oscilloscope

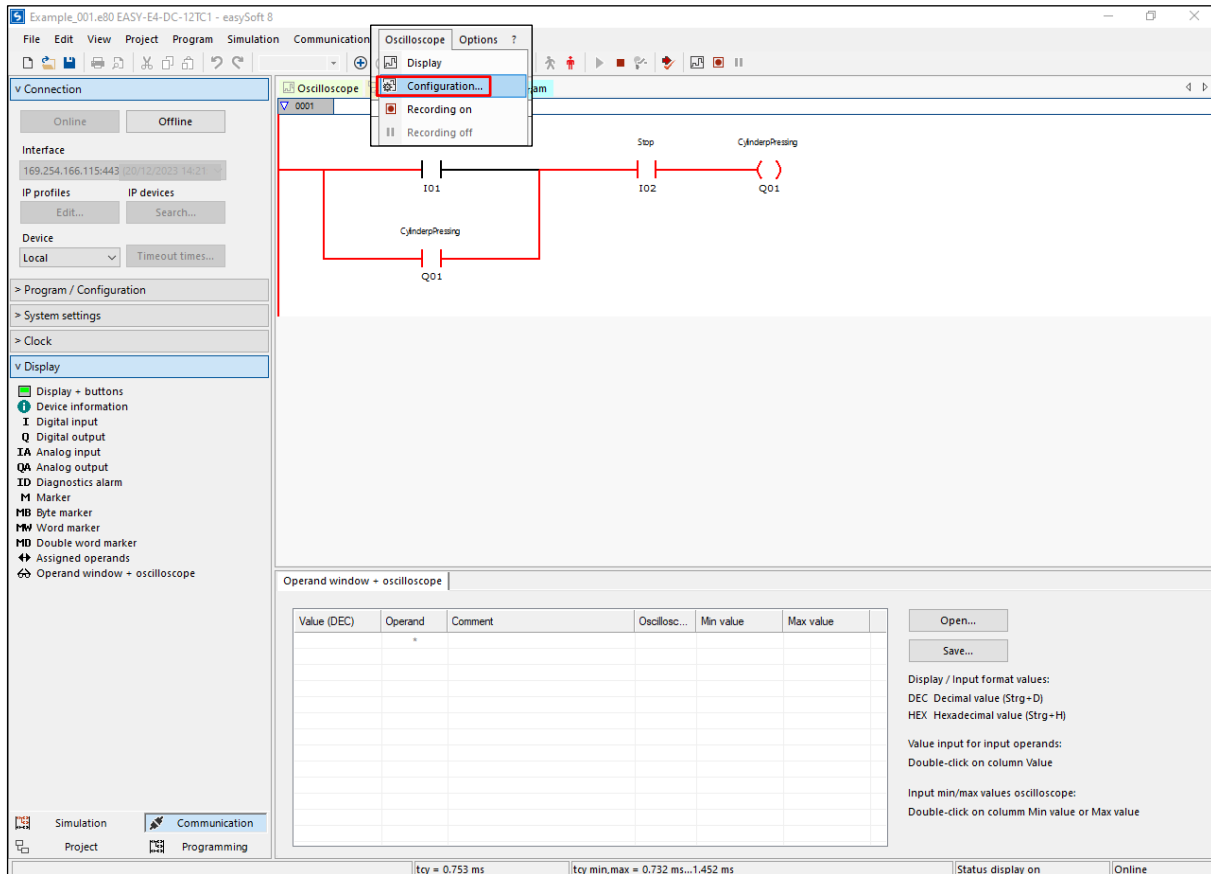


Fig. 36

With the oscilloscope function, you can, inter alia, create a variable list where the status of the various variables can be viewed online. The status of the variables can also be displayed in a timeline using the oscilloscope.

- ▶ Connect with the easyE4 (Online).
- ▶ Go to "Oscilloscope" in the menu and select "Configuraton...".

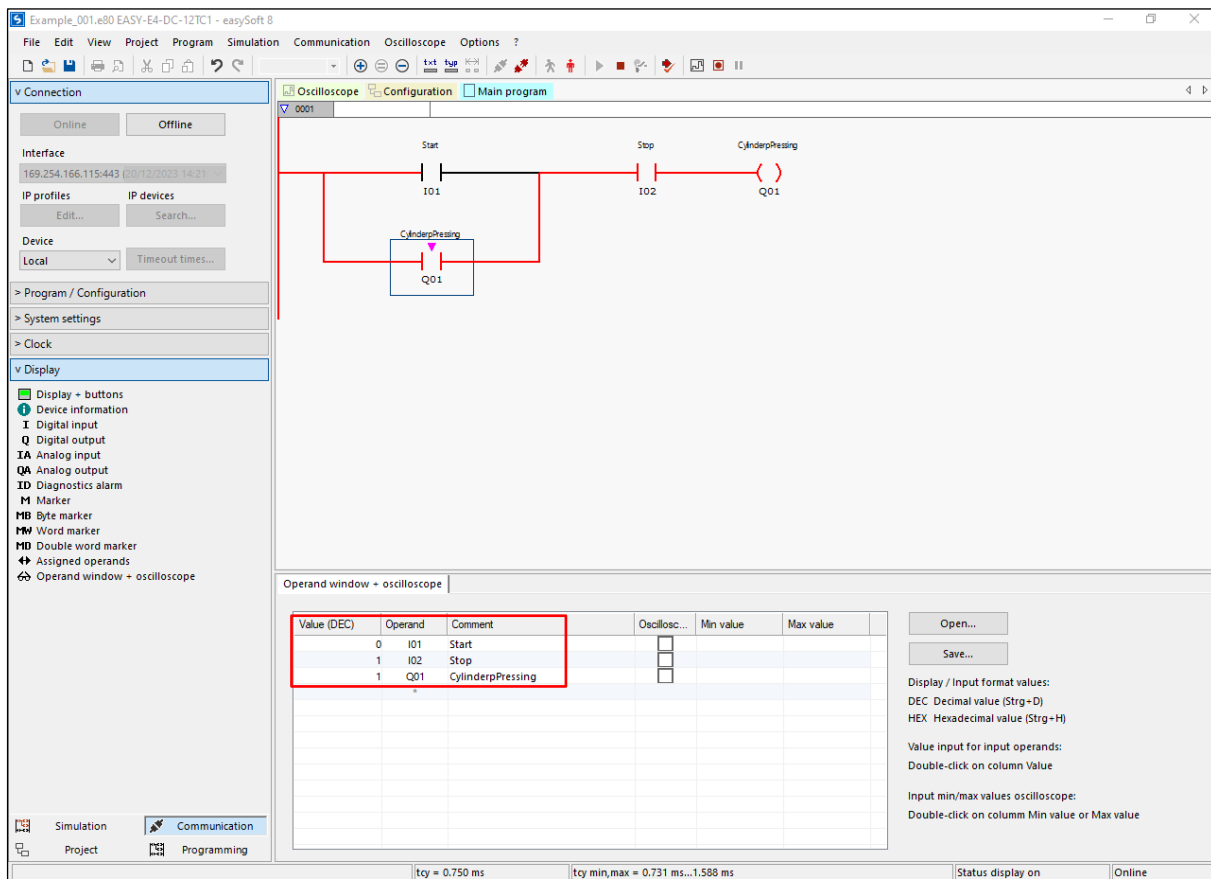


Fig. 37

In the properties field, the variables that need to be displayed can be entered in the 'Operand' column. In this example it is I1, I2 and Q1. The comments that have been assigned are displayed directly here. In this example, I2 and Q1 are high. This is shown in the "Value" column.

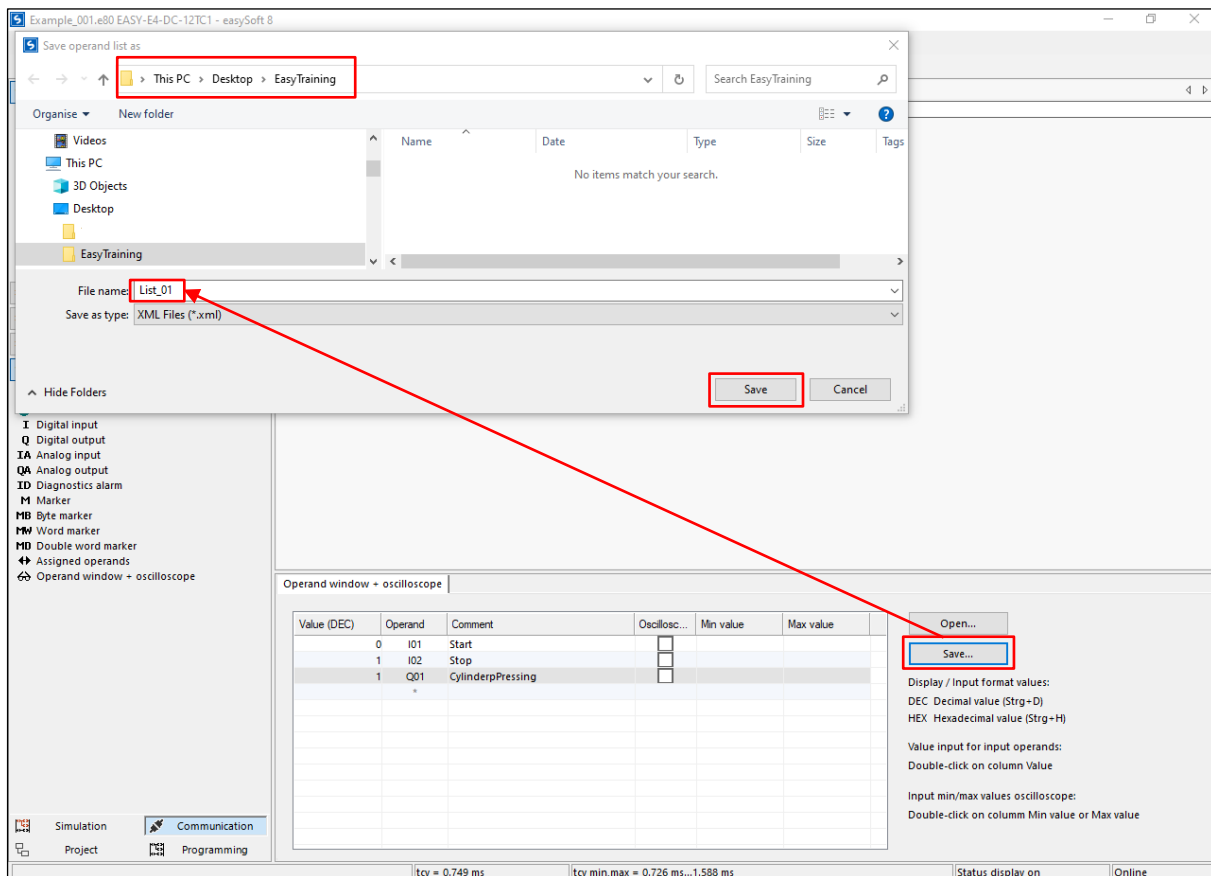


Fig. 38

This variable list can be saved for future use.

- Click on "Save" for this.
- Save the list with the name 'List\_01' in the EasyTraining folder.

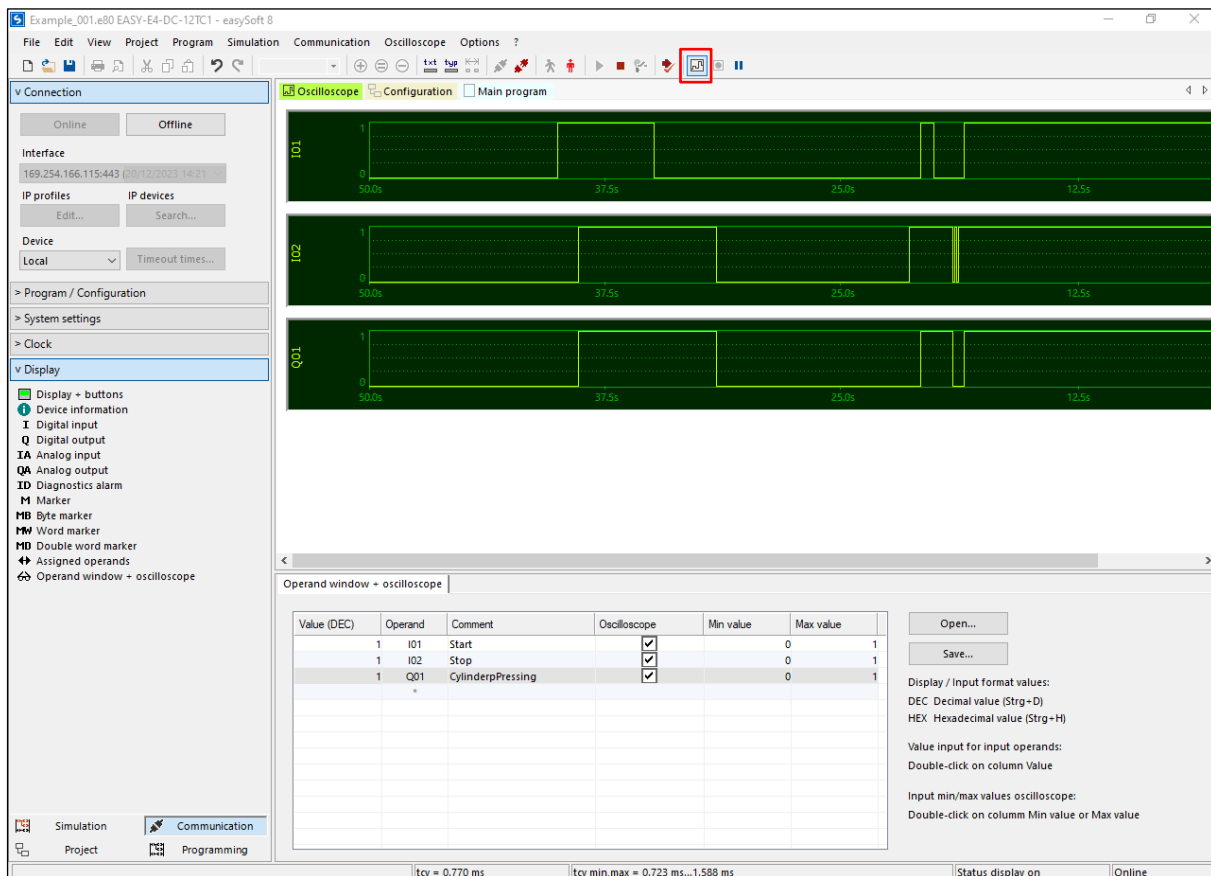


Fig. 39

To display the progress of the various signals, they can be checked in the 'Oscilloscope' column.

- Click on the "Oscilloscope" button in the menu bar.

The progression of the inputs and outputs is now displayed.

With the  pause button the view can be stopped. The view can be restarted with the  record button.

## 2.7 Timer

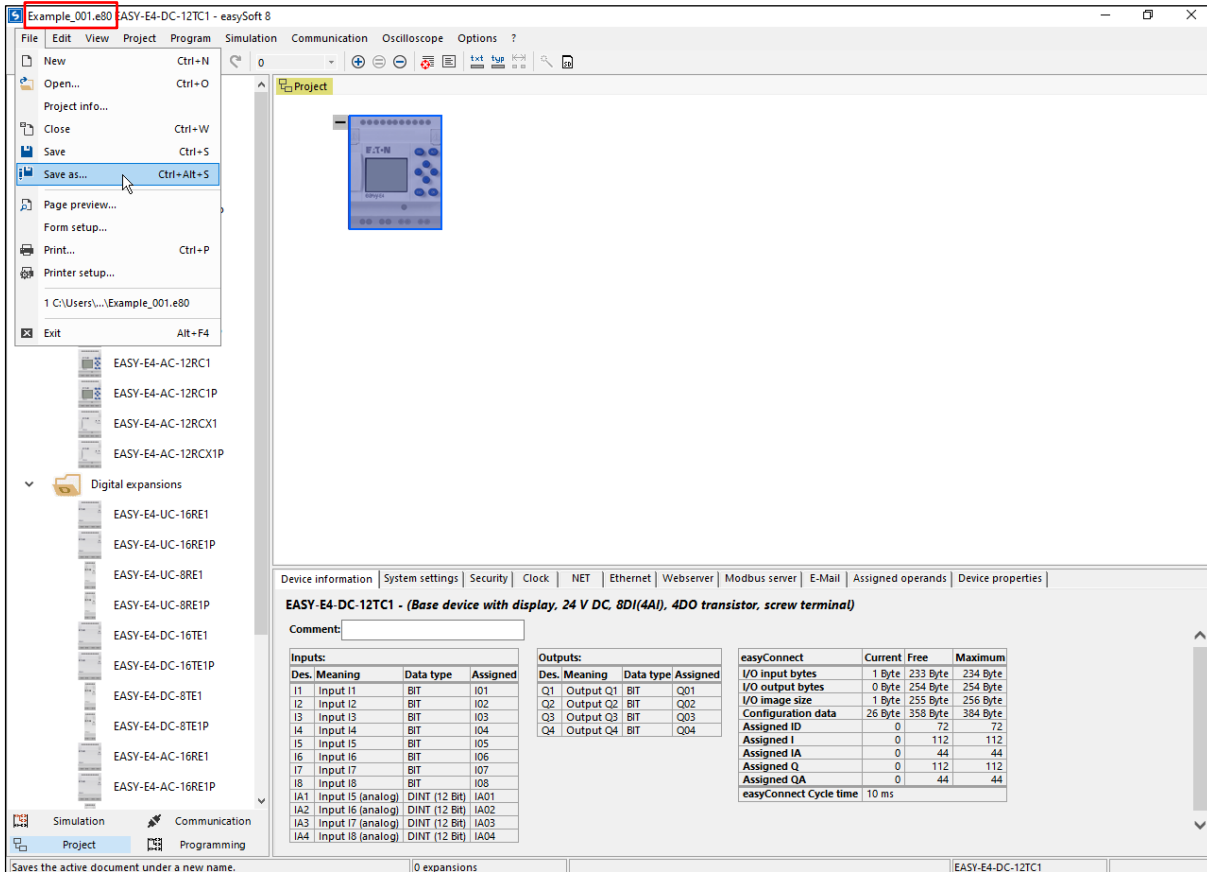


Fig. 40

The product must be pressed for a certain period of time. After 'Start' has been pressed, a timer begins running. After this time, the cylinder must retract again.

**Tip!** If the program is going to be modified, it is advisable to save the current program under a different name.

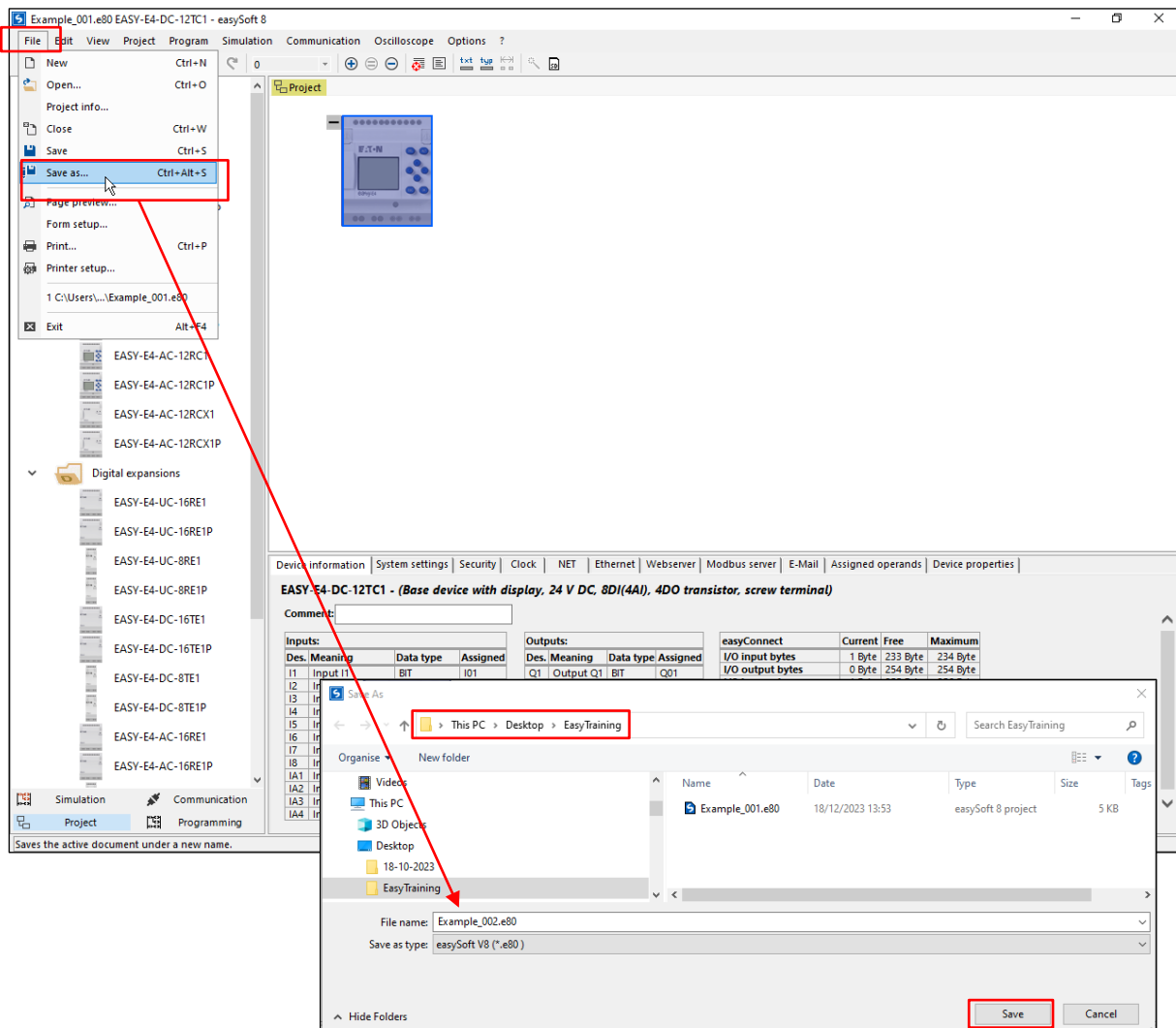


Fig. 41

- Go to the "File" menu bar. Select "File save as..."
- Give the program the name "Example\_002" and click on "Save".

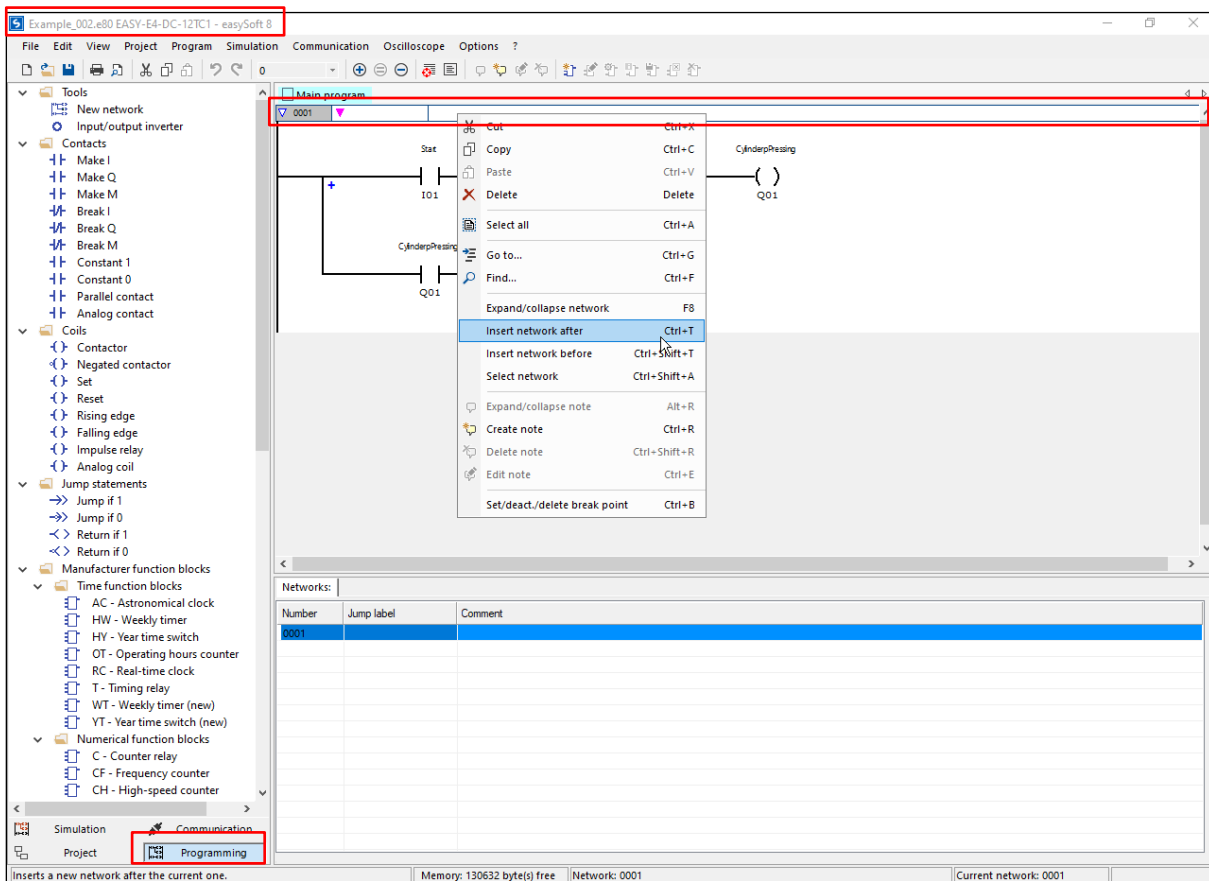


Fig. 42

- Select the "Programming" menu.

First, a programming rung must be added.

- Select the last line. In this example, line "001". This rung is now outlined in blue.

Right click on the mouse. A new screen is opened. Select "Insert network after" here. A new rung (002) is now added after the existing rung.

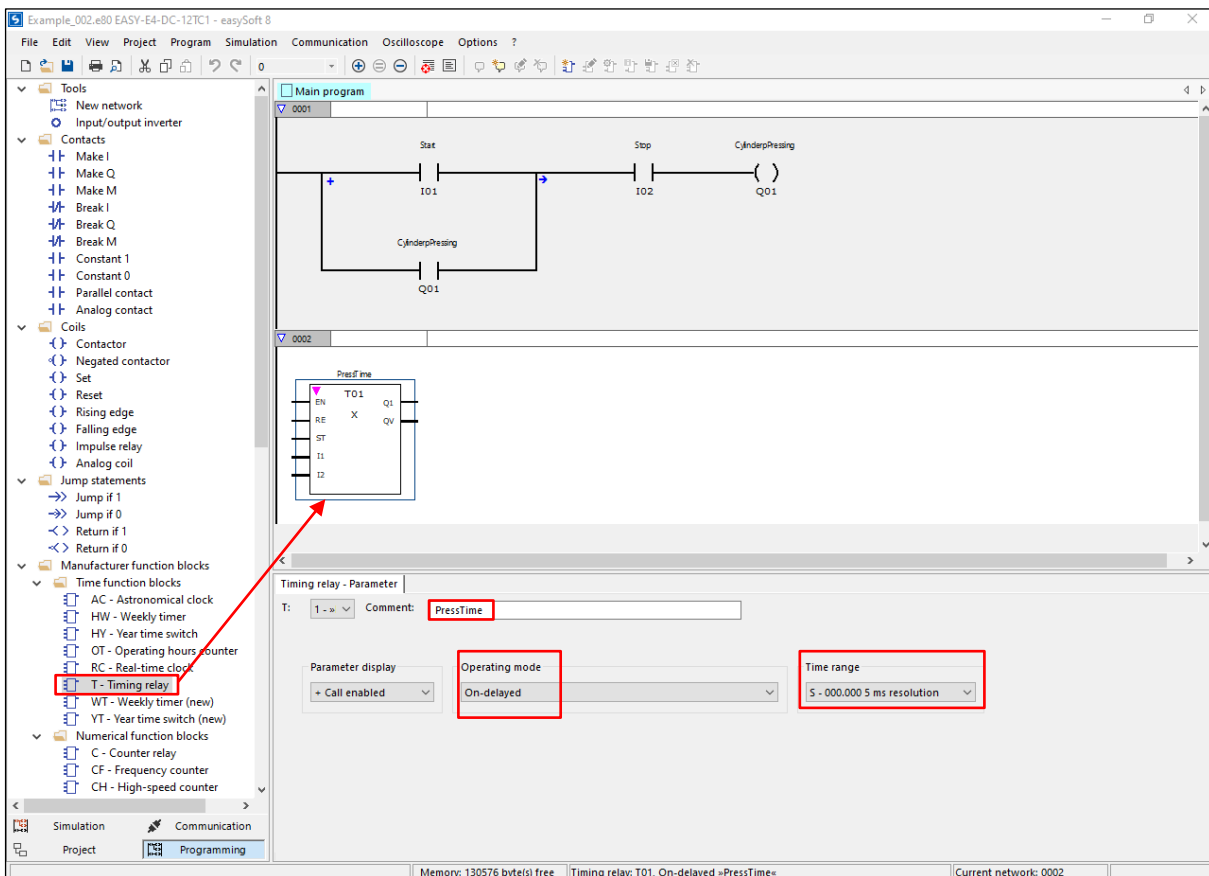


Fig. 43

For the time determination, a time relay, or timer, is used.

- Drag a 'T-Timing relay' from the toolbox to the new rung and give it the name 'PressTime'.

The 'Operating mode' indicates what type of timer it should be. In this example, it must be a time delay relay with delayed pull-in (ON-delay).

- Under "Operating mode", select ➔ "On-delayed".
- At "Time range" select "S-000.000 5ms resolution".

This means that the amount of time is indicated in seconds and milliseconds.

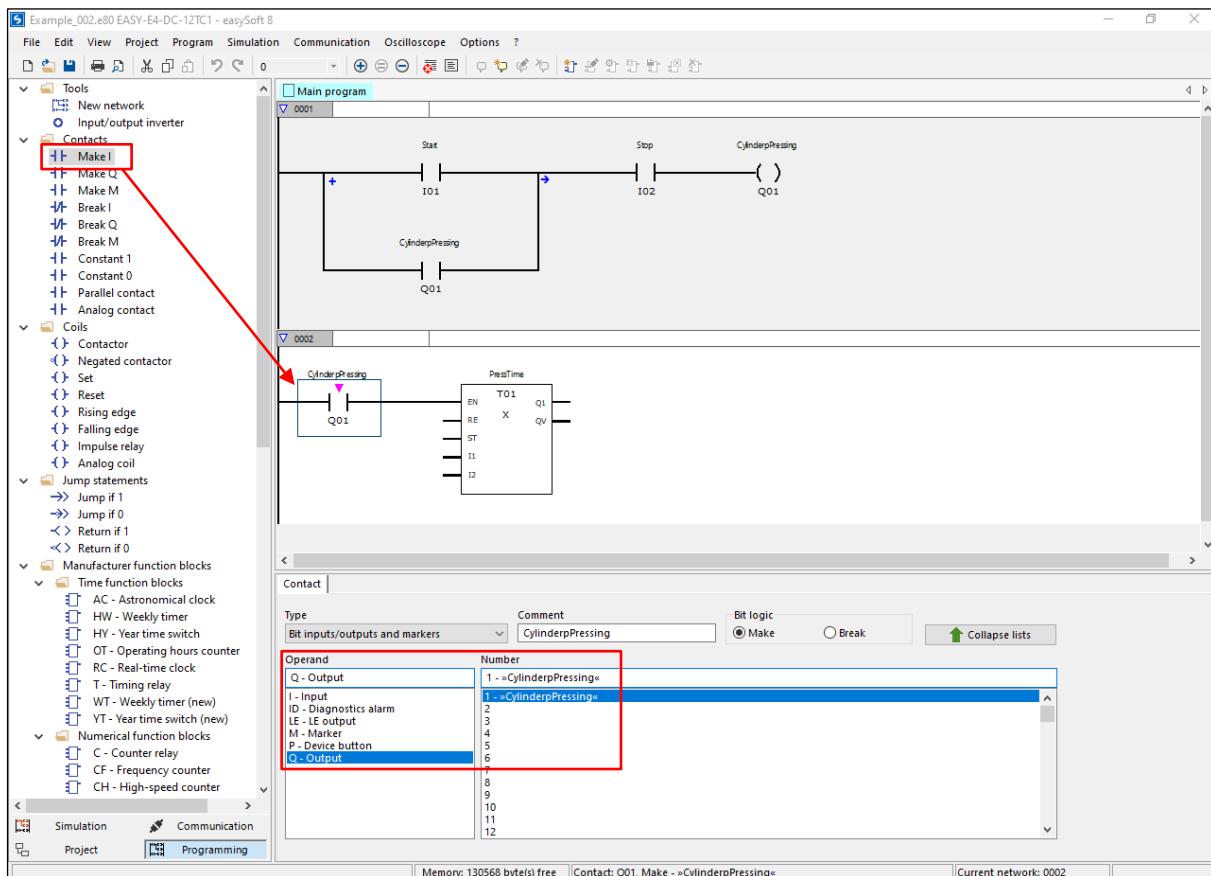


Fig. 44

The timer must be started by making the 'EN' input high.

- Drag a 'Make' contact to the 'EN' input of the timer.

The timer is started by the contact of output 'CylinderPressing' = 'Q1'.

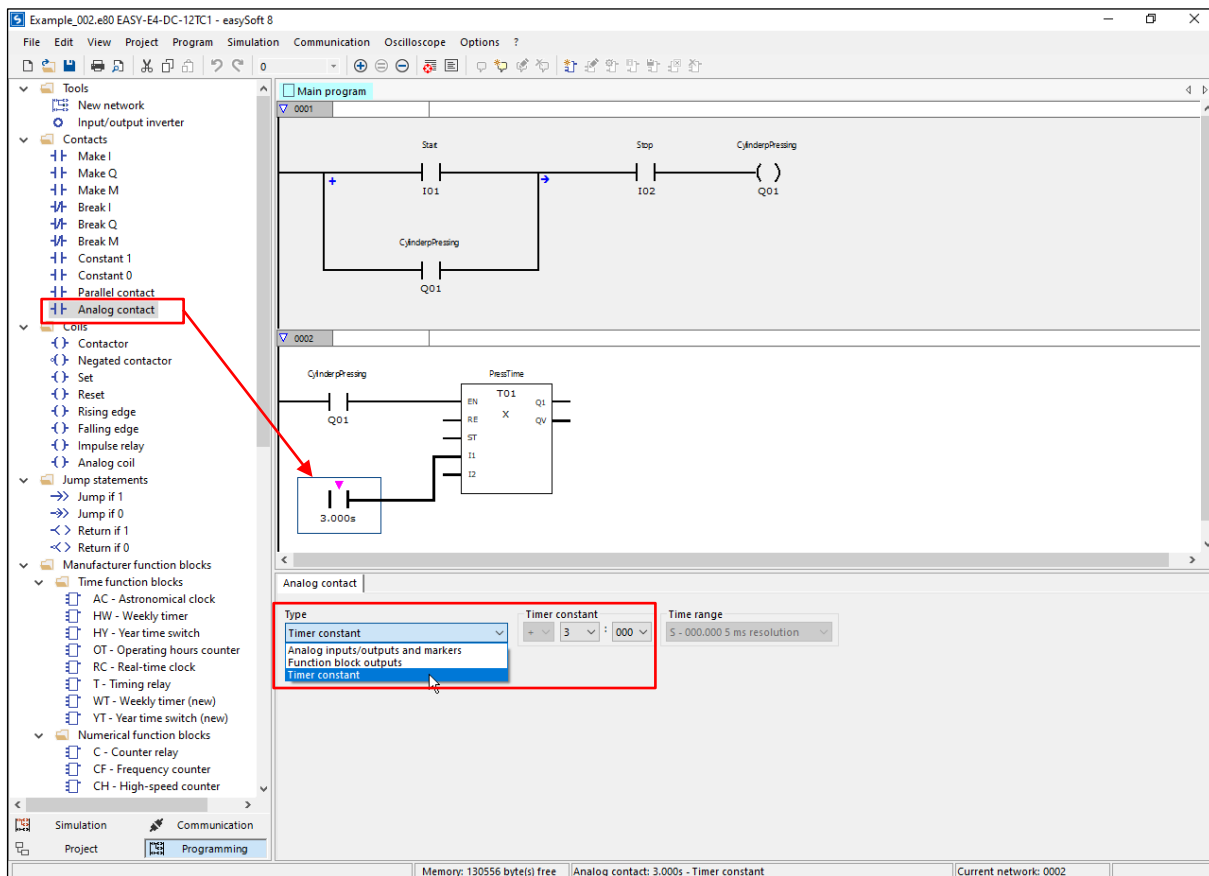


Fig. 45

The amount of time must also be assigned to the timer. This is a number.

- Drag an 'Analog contact' to 'I1' of the 'PressTime' timer.
- Select "Timer constant" at "Type".

The amount of time is entered under "Timer constant". In this example 3 seconds and 0 milliseconds.

Nothing is connected to the 'Q1' output of the timer.

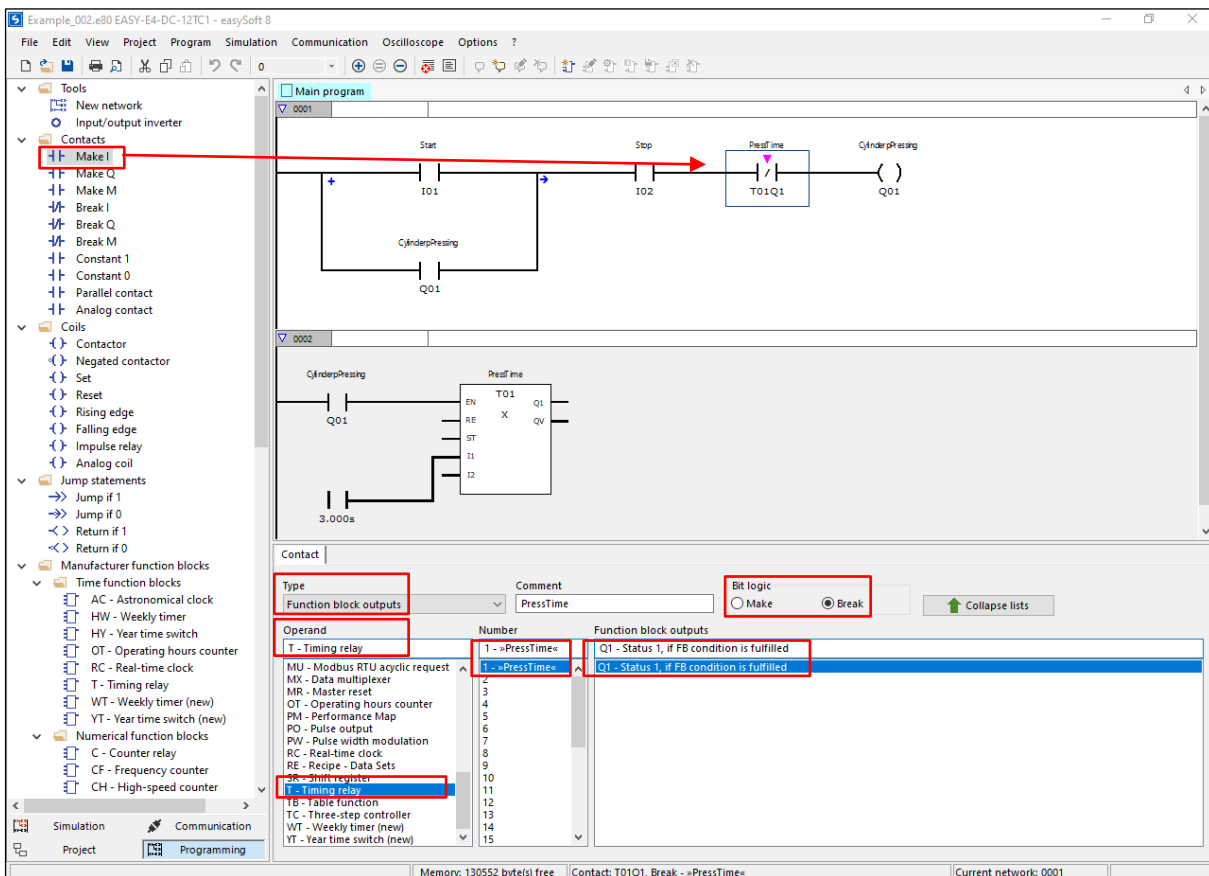


Fig. 46

The output of the 'PressTime' timer can be directly incorporated as a contact in the motor circuit.

- Place a 'Make' contact in series with the 'Stop' contact.
- At "Type", select ➔ "Function block outputs".

The type of function block can now be selected under 'Operand'.

- Here, select timer "T – Timing relay".

The correct timer can be selected under 'Number'. That is Timer T1 = "PressTime".

For the 'Function block outputs,' only output Q1 of the timer named 'PressTime' can be selected.

As soon as the time has elapsed, the circuit must be interrupted. Select "break" at "Bit logic". A diagonal line will appear in the contact.

- This means that the signal is inverted.

Output Q1 of 'PressTime' becomes high after the time has elapsed (time delay relay with delayed pull-in/ON-delay). Because output Q1 of 'PressTime' is inverted, it will interrupt the circuit to Q1 = 'CylinderPressing'.

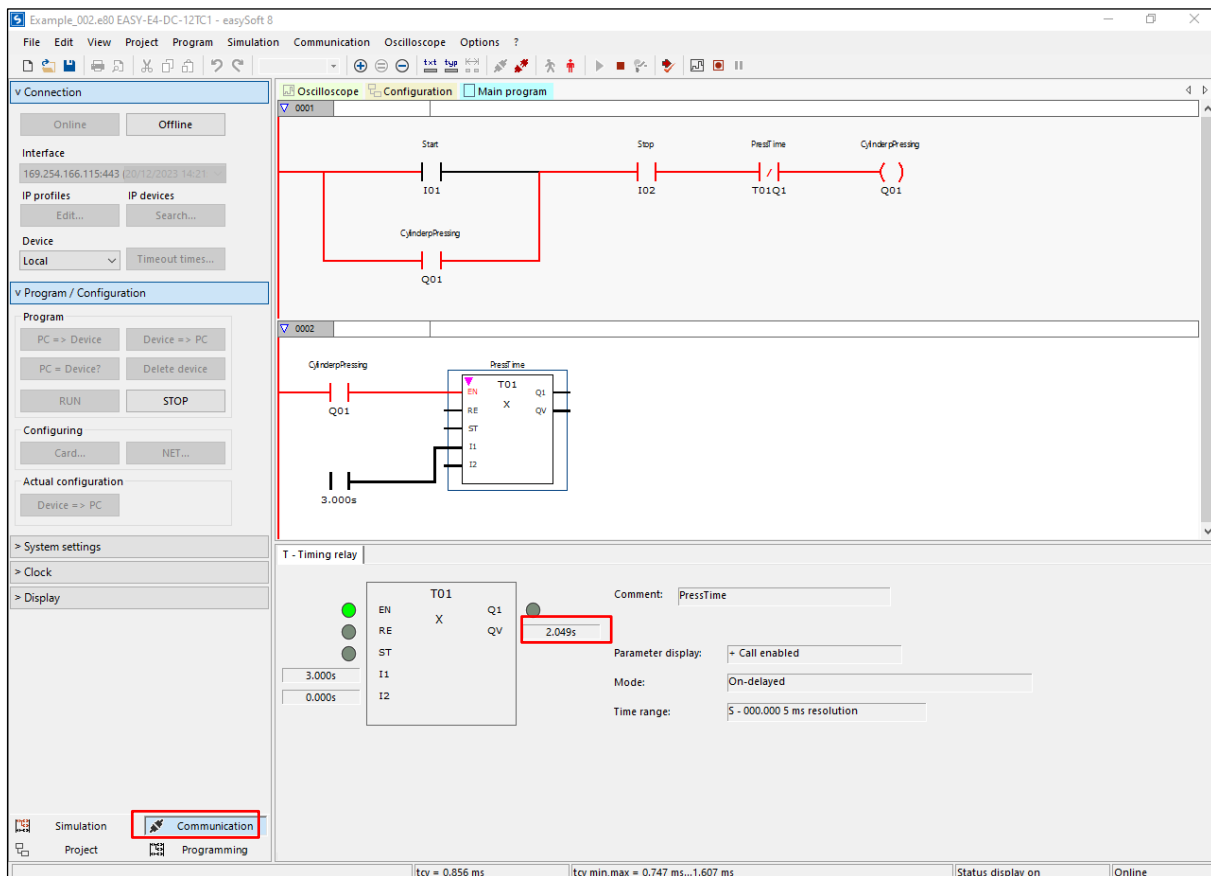


Fig. 47

Transfer the program to the PLC via the 'Communication' menu and check the operation. View via "Status display on".

Select function block 'T01' in the program. In the properties field, the timer is displayed with the current data. The 'EN' input is high, and the timer is running. The elapsed time is shown at the 'QV' output.

As soon as the time has elapsed, the contact is opened, after which the cylinder is no longer controlled.

## 2.8 Markers and marker words

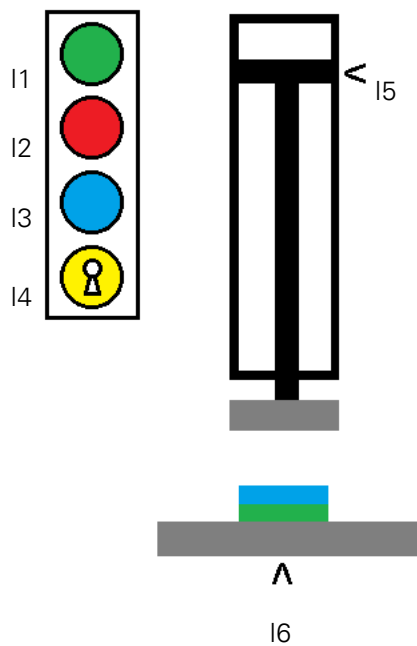


Fig. 48

The cylinder is now driven out by pressing 'Start'. However, the cylinder may only be extended if the cylinder is fully retracted (I5 = CylinderUp). And the cylinder may only press if there is a product present that needs to be pressed (I6 = ProductPlaced). In other words, after pressing, the product must first be removed and a new product must be placed. Therefore, it must be remembered whether the product currently present has been pressed or still needs to be pressed.

To remember something, a memory is usually used. This is called a marker. Such a memory then works with a Set and a Reset condition. A memory is indicated by the letter "M" for marker.

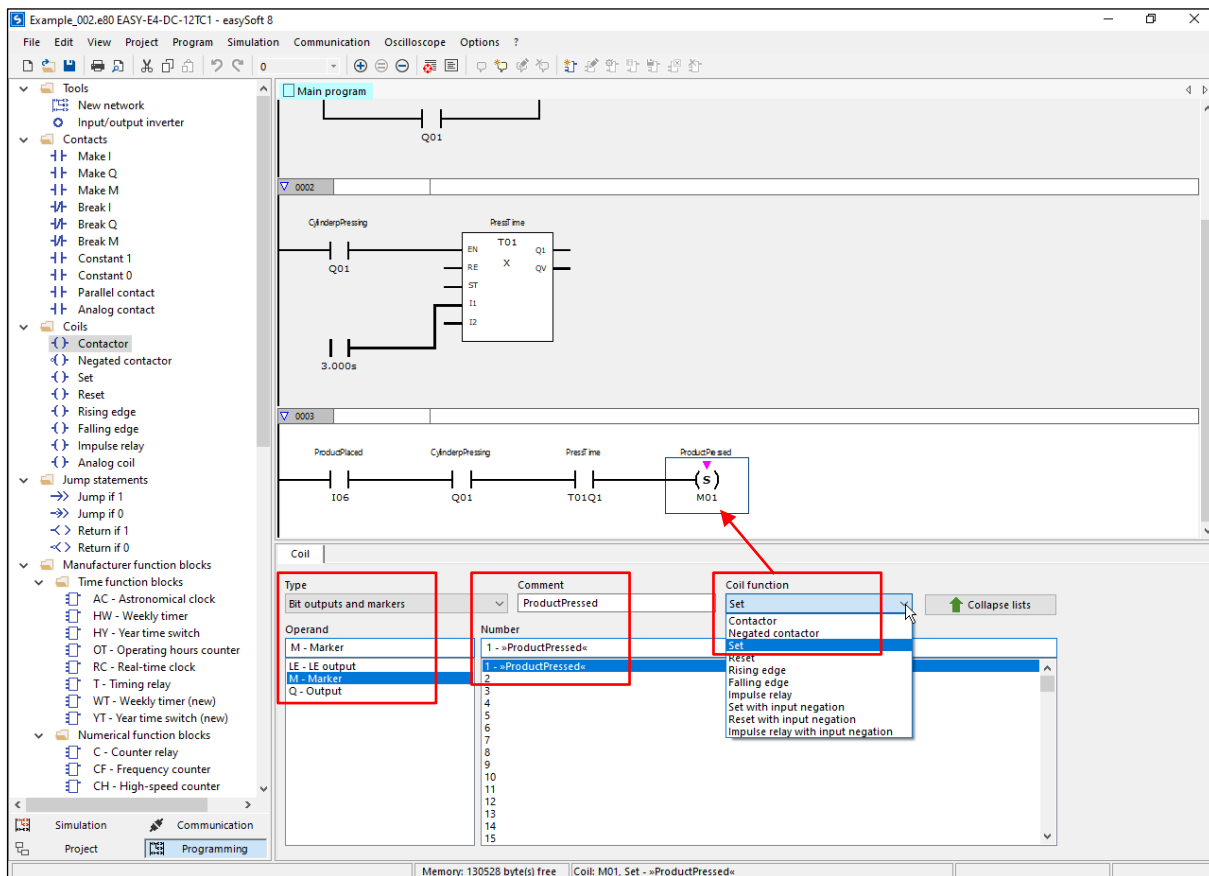


Fig. 49

### Set memory:

The "ProductPressed" (M1) memory must be set as:

- the product is in position "ProductPlaced" (I6)
- and if output "CylinderPressing" (Q1) is activated
- and if the PressTime expired (T1)
- Go to the "Programming" menu.
- Add a new rung (CTRL+T).
- Drag the "Make" contacts into the rung to be programmed.
- Drag a "Contactor" from the "Coils" section to the end of the circuit.

At "Type" select the "M – Marker".

- Under "Number," select number "1" and name it "ProductPressed".

For "Coil function," "Set" must be selected.

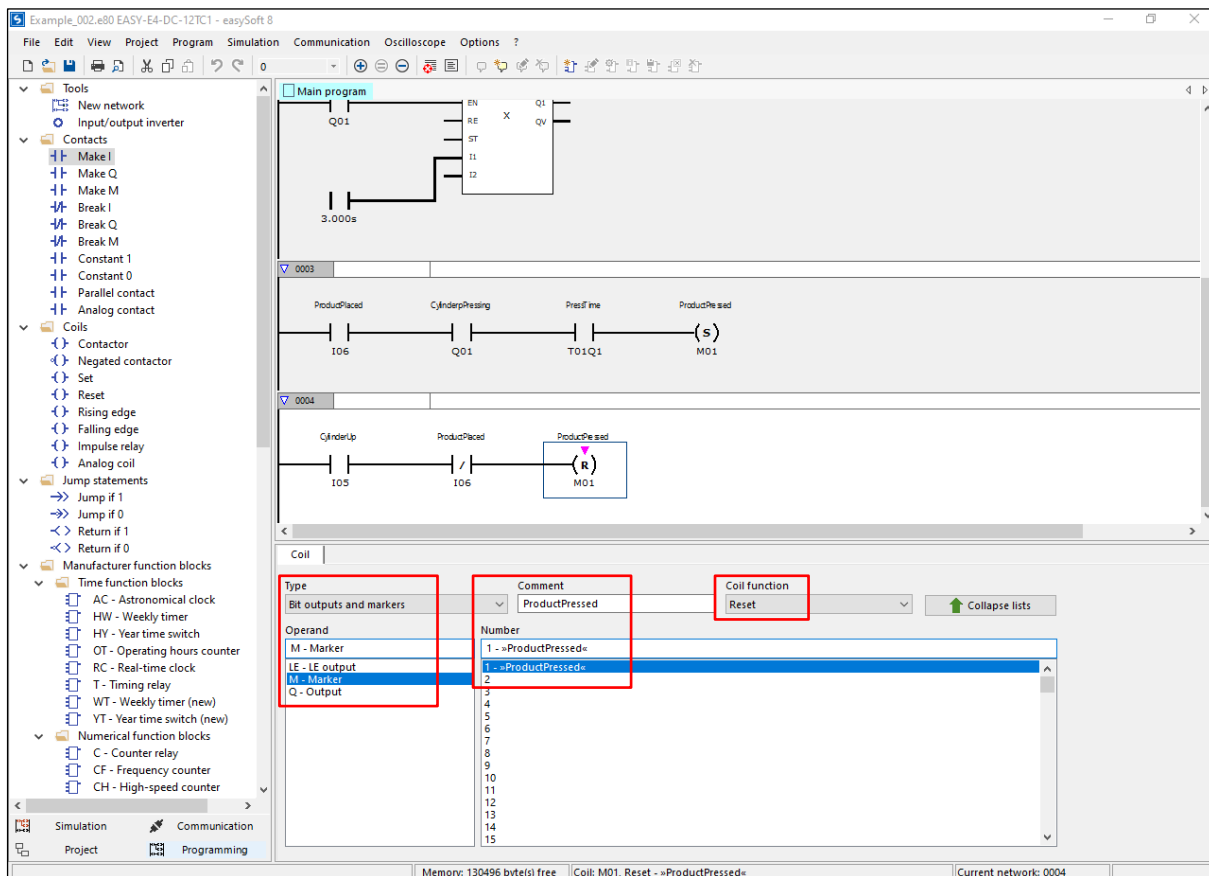


Fig. 50

## Reset memory:

The memory is reset as:

- when the cylinder is up again "CylinderUp" (I5)
  - and when the product has been removed. "ProductPlaced" (I6) is low
- Add a new rung (CTRL+T).

Program the abovementioned functionality.

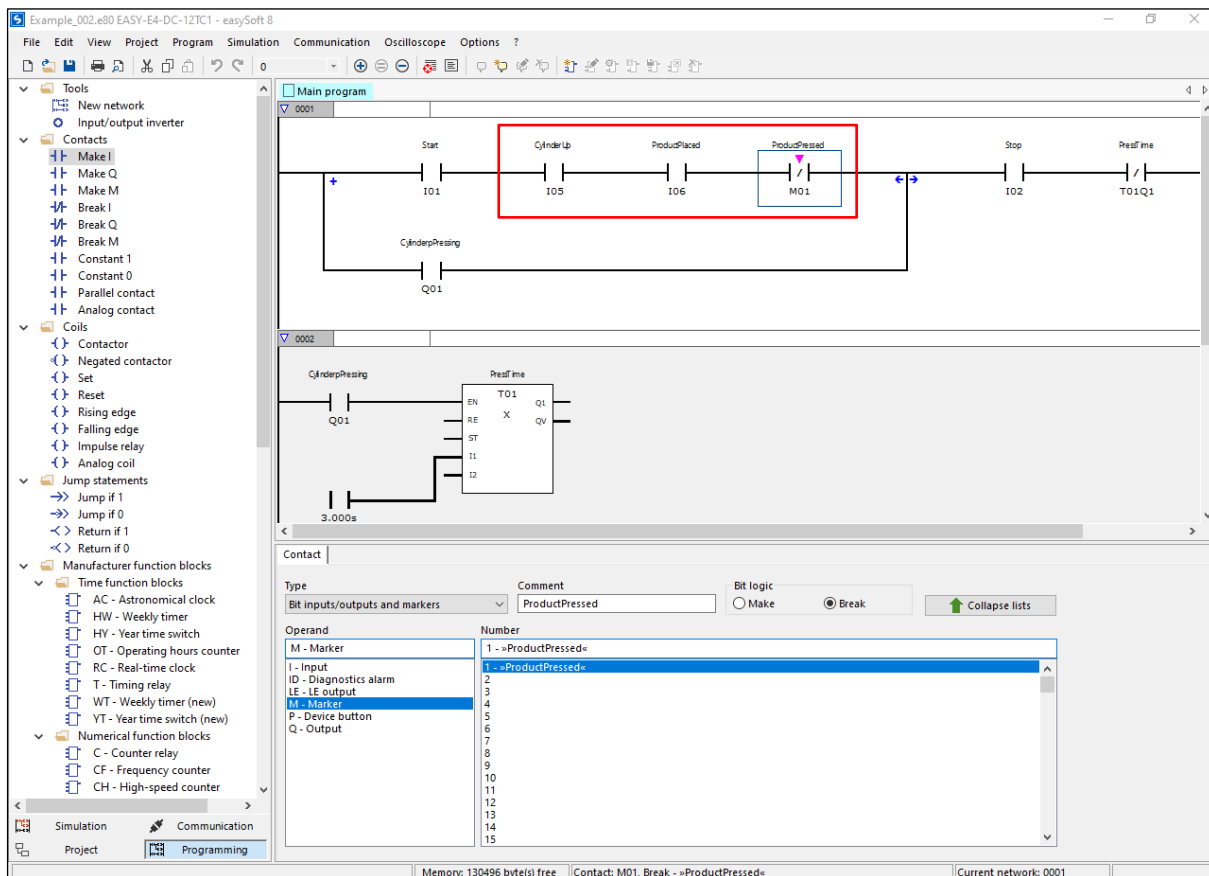


Fig. 51

In the diagram on rung 1, the contact "CylinderUp" (I5), the contact "ProductPlaced" (I6), and the memory "ProductPressed" (M1) are added as the starting condition.

The marker M1=ProductPressed must be low (off) to be able to start again, hence the "break" contact (normally closed contact).

- Save the program.
- Upload the program to the easyE4 and test the function.

What exactly are memories/storage in a control system and what you should pay attention to.

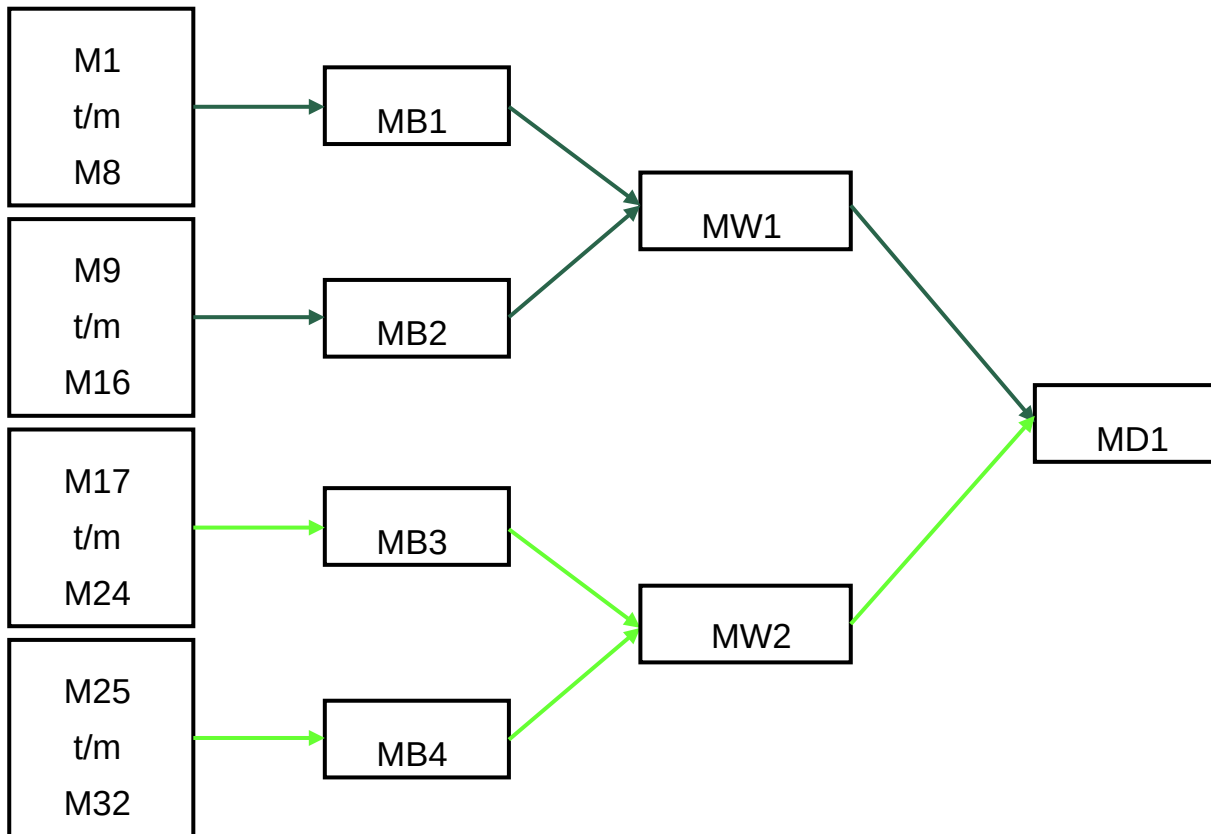


Fig. 52

In the easyE4 series, you can use:

|    |                    |         |
|----|--------------------|---------|
| M  | = Marker           | = Bit   |
| MB | = Marker byte      | = Byte  |
| MW | = MarkerWord       | = Word  |
| MD | = MarkerDoubleWord | = DWord |

A marker is a memory/storage unit in the control system that, just like an output, can assume a status of 0 or 1.

A Marker Byte consists of 8 Markers.

A MarkerWord consists of 2 MarkerBytes and thus has 16 Markers.

A MarkerDoubleWord consists of 2 MarkerWords and thus  $2 \times 16 = 32$  Markers.

An overview and structure of the markers is on the following page.

|       |            |            |            |            |            |            |            |            |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|
| Bit   | 64 .. 57   | 56 .. 49   | 48 .. 41   | 40 .. 33   | 32 .. 25   | 24 .. 17   | 16 .. 9    | 8 .. 1     |
| Byte  | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          |
| Word  | 4          |            | 3          |            | 2          |            | 1          |            |
| Dword | 2          |            |            |            | 1          |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Bit   | 128 .. 121 | 120 .. 113 | 112 .. 105 | 104 .. 97  | 96 .. 89   | 88 .. 81   | 80 .. 73   | 72 .. 65   |
| Byte  | 16         | 15         | 14         | 13         | 12         | 11         | 10         | 9          |
| Word  | 8          |            | 7          |            | 6          |            | 5          |            |
| Dword | 4          |            |            |            | 3          |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Bit   | 192 .. 185 | 184 .. 177 | 176 .. 169 | 168 .. 161 | 160 .. 153 | 152 .. 145 | 144 .. 137 | 136 .. 129 |
| Byte  | 24         | 23         | 22         | 21         | 20         | 19         | 18         | 17         |
| Word  | 12         |            | 11         |            | 10         |            | 9          |            |
| Dword | 6          |            |            |            | 5          |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Bit   | 256 .. 249 | 248 .. 241 | 240 .. 233 | 232 .. 225 | 224 .. 217 | 216 .. 209 | 208 .. 201 | 200 .. 193 |
| Byte  | 32         | 31         | 30         | 29         | 28         | 27         | 26         | 25         |
| Word  | 16         |            | 15         |            | 14         |            | 13         |            |
| Dword | 8          |            |            |            | 7          |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Bit   | 320 .. 313 | 312 .. 305 | 304 .. 297 | 296 .. 289 | 288 .. 281 | 280 .. 273 | 272 .. 265 | 264 .. 257 |
| Byte  | 40         | 39         | 38         | 37         | 36         | 35         | 34         | 33         |
| Word  | 20         |            | 19         |            | 18         |            | 17         |            |
| Dword | 10         |            |            |            | 9          |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Bit   | 384 .. 377 | 376 .. 369 | 368 .. 361 | 360 .. 353 | 352 .. 345 | 344 .. 337 | 336 .. 329 | 328 .. 321 |
| Byte  | 48         | 47         | 46         | 45         | 44         | 43         | 42         | 41         |
| Word  | 24         |            | 23         |            | 22         |            | 21         |            |
| Dword | 12         |            |            |            | 11         |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Bit   | 448 .. 441 | 440 .. 433 | 432 .. 425 | 424 .. 417 | 416 .. 409 | 408 .. 401 | 400 .. 393 | 392 .. 385 |
| Byte  | 56         | 55         | 54         | 53         | 52         | 51         | 50         | 49         |
| Word  | 28         |            | 27         |            | 26         |            | 25         |            |
| Dword | 14         |            |            |            | 13         |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Bit   | 512 .. 505 | 504 .. 497 | 496 .. 489 | 488 .. 481 | 480 .. 473 | 472 .. 465 | 464 .. 457 | 456 .. 449 |
| Byte  | 64         | 63         | 62         | 61         | 60         | 59         | 58         | 57         |
| Word  | 32         |            | 31         |            | 30         |            | 29         |            |
| Dword | 16         |            |            |            | 15         |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Byte  | 72         | 71         | 70         | 69         | 68         | 67         | 66         | 65         |
| Word  | 36         |            | 35         |            | 34         |            | 33         |            |
| Dword | 18         |            |            |            | 17         |            |            |            |
|       |            |            |            |            |            |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Byte  | 512        | 511        | 510        | 509        | 508        | 507        | 506        | 505        |
| Word  | 256        |            | 255        |            | 254        |            | 253        |            |
| Dword | 128        |            |            |            | 127        |            |            |            |
|       |            |            |            |            |            |            |            |            |
|       |            |            |            |            |            |            |            |            |
| Word  | 512        |            | 511        |            | 510        |            | 509        |            |
| Dword | 256        |            |            |            | 255        |            |            |            |

Fig. 53

## Elementary Data Types

Following is a list of the various elementary data types. These data types are independent of the **programming language** you select.

| Type/(description)  | Length in bits | Format                    | Value range                         | Example  |
|---------------------|----------------|---------------------------|-------------------------------------|----------|
| BOOL/(Bit)          | 1              | Binary (bool)             | 0/1, FALSE/TRUE                     | TRUE (1) |
| BYTE/(Byte)         | 8              | Decimal number (unsigned) | 0...255                             | 128      |
| WORD/(Word)         | 16             | Decimal number (unsigned) | 0 - 65535                           | 1023     |
| DWORD/(double word) | 32             | Decimal number (signed)   | -2 147 483 648...<br>+2 147 483 647 | - 65535  |

Fig. 54

A marker can only assume the value of 0 or 1. However, a byte consists of 8 bits and can therefore contain a binary value that can be interpreted as a decimal value. For this, you have to count binary.

For example 0000 0111=7 and 1010 0101=165.

In the table above, the value ranges of a bit (marker), byte, word, and double word are listed.

**Tip!** Prevent unwanted double assignment of markers.

Example:

Suppose M1 is used to control an output and has the value 1.

If the value 132 is now written into MB1 (MarkerByte 1) later in the program, the following binary code will be in MB1 = **1**000 010**0**. The red **0** represents M1 and the red **1** represents M8. This thus overrides M1. The output will therefore not be controlled.

## 2.9 Counter

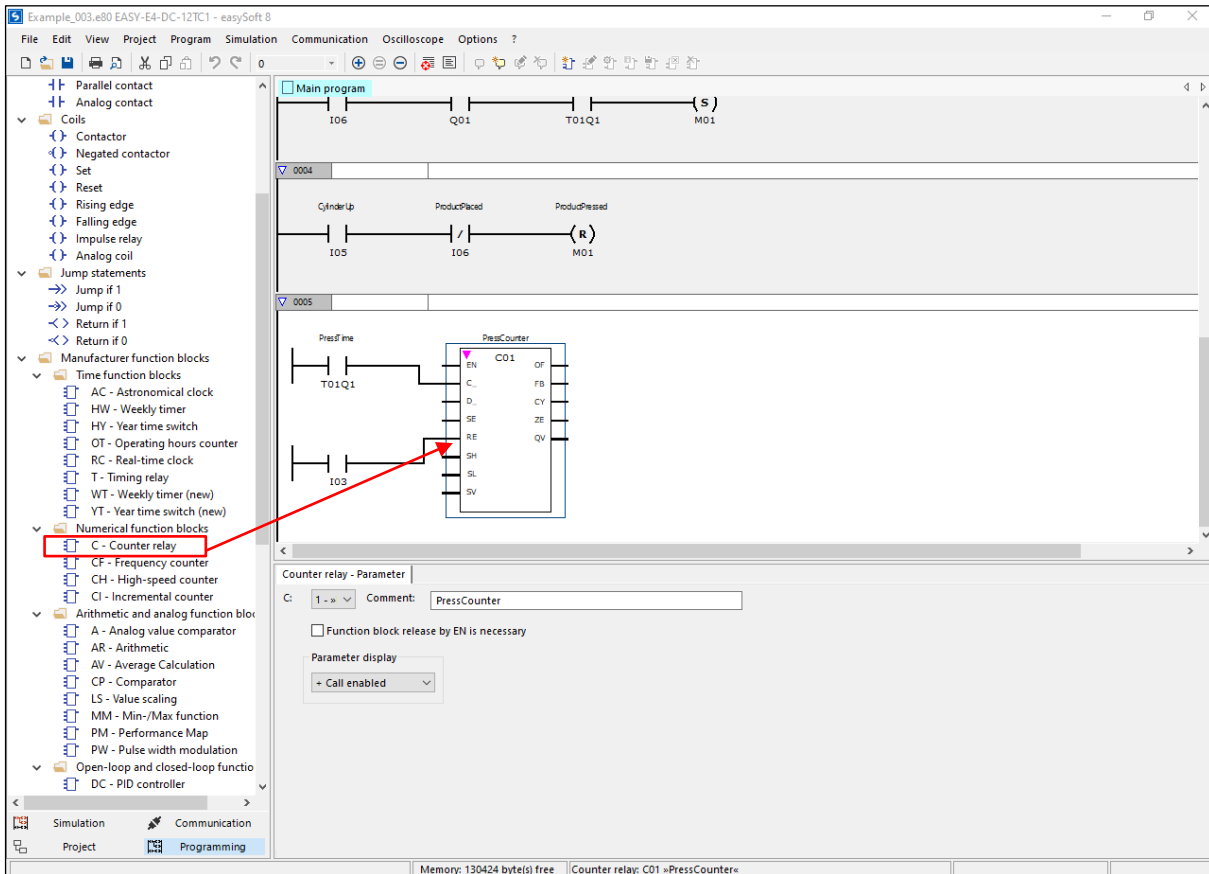


Fig. 55

In connection with maintenance, the number of press movements must be counted. A pressing is counted as soon as the press time (PressTime) has elapsed. The maintenance technician can reset the counter to 0 using a key switch "ResetCounter" (I4).

- Save the first program under a different name.

This prevents the current working version from being overwritten, and the extension of the PressCounter (C1) can be programmed.

Create a new rung.

- Drag a "C - Counter relay" to the program rung and give this counter the name "PressCounter".
- Place a "Make" contact on the "C\_" input of the counter and link it to output Q1 of the "PressTime".

On the counter, this is the input where the count pulse is offered. With every pulse, the value in the counter increases.

- Place a "Make" contact on the "RE\_" input of the counter and link it to the "ResetCounter" input (I4).

On the counter, this is the input where the reset pulse is offered. With this input, the counter is set to 0.

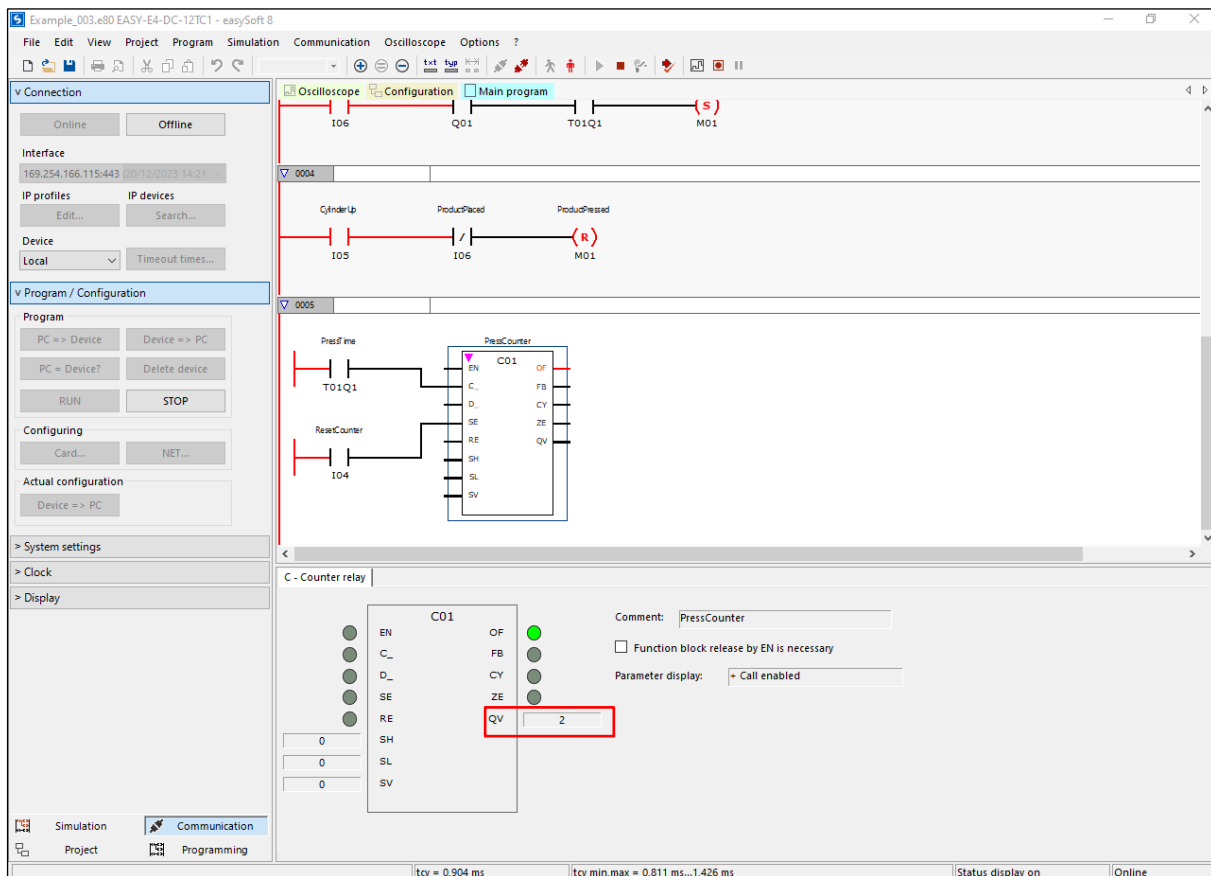


Fig. 56

- Save the program, upload the program to the easyE4 and test the function.

In this image, when monitoring online, you can see that a pressing has occurred twice.

## 2.10 Error

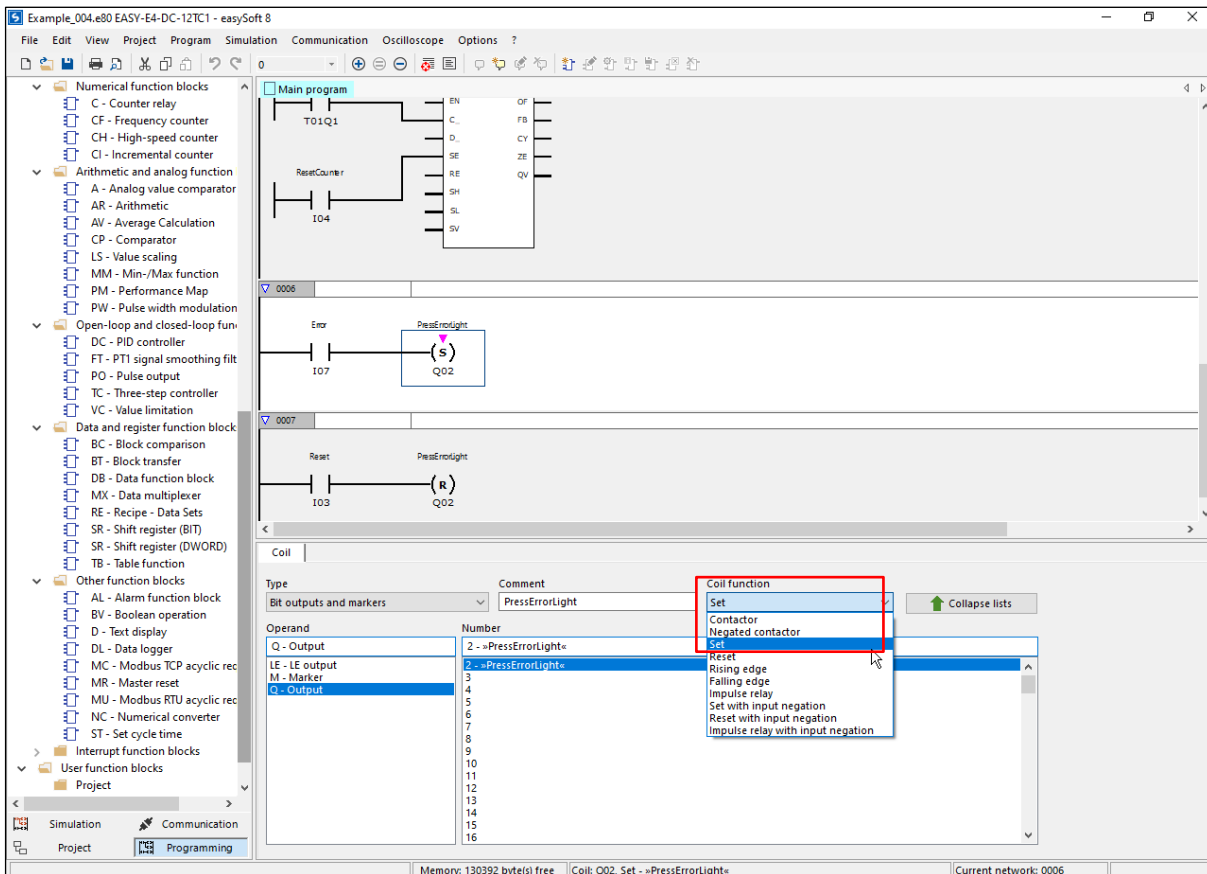


Fig. 57

An error can also occur in the process. The error is signaled with input "Error" (I7). If this error occurs, the output "PressErrorLight" (Q2) is set. As soon as the "Error" is low again, the malfunction can be reset with the "Reset" input (I3).

If the error is high, the pressing is stopped immediately and the cylinder moves up.

- Save the program under a different name and program the desired functionality in the easyE4.

In the figure above, it can be seen that output Q2 is not simply controlled, but is set and reset.

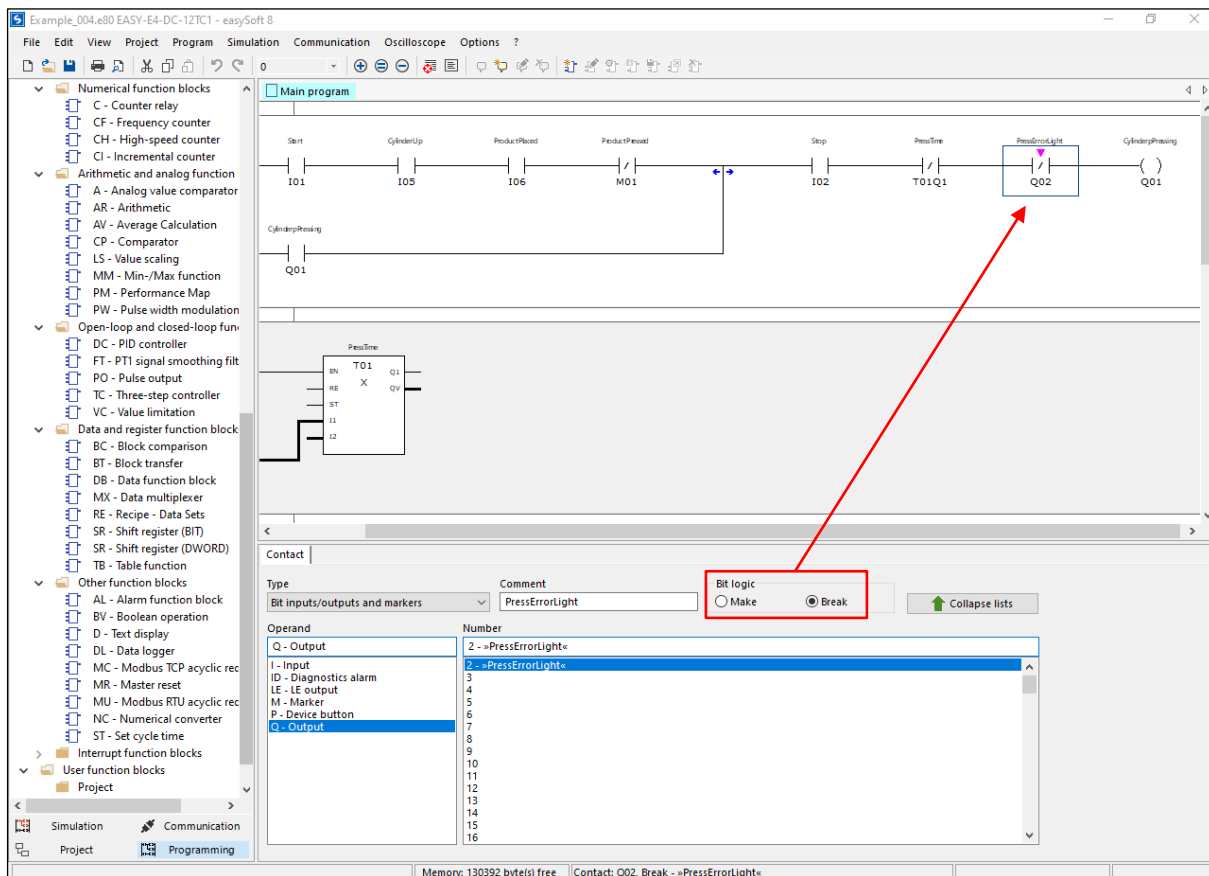


Fig. 58

The error is used to stop the pressing. For this, the circuit diagram for controlling the "CylinderPressing" must be adjusted.

The error message Q2 is included as an inverted contact in line 1 when controlling "CylinderPressing" (Q1). As soon as the error is present, the cylinder is no longer controlled.

- Save the program, upload the program to the easyE4 and test the function.

## 2.10 Create a function block

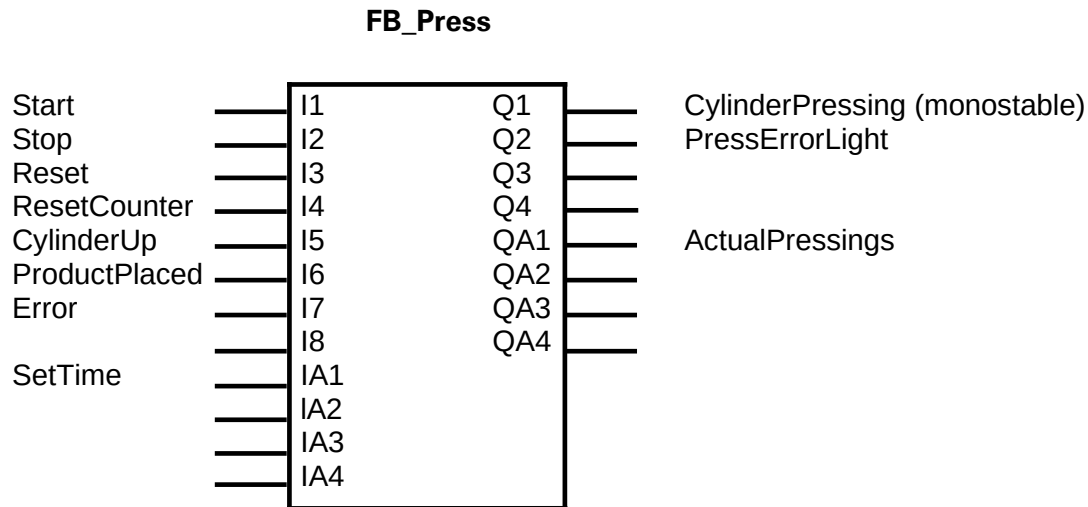


Fig. 59

The control of the press has been tested and appears to be functional. The desire, however, is to control multiple presses with the same program. This could be done by copying the program lines and adjusting the used I/O. An alternative is to create a function block (FB) yourself with this functionality.

A user-created FB is just like, for example, a Timer that is already present in the software. The Timer has inputs and outputs and a piece of program that handles the time function. This piece of program isn't programmed every time, but is called up more often. The programmer does not have access to the piece of program in the Timer.

With a User Function Block, the programmer determines the inputs and outputs and the program that must be programmed within this FB. It is sensible to first create a diagram of the desired I/O on the FB.

The digital I/O of the press is already known

| Inputs |               | Outputs |                               |
|--------|---------------|---------|-------------------------------|
| I1     | Start         | Q1      | CylinderPressing (monostable) |
| I2     | Stop          | Q2      | PressErrorLight               |
| I3     | Reset         |         |                               |
| I4     | ResetCounter  |         |                               |
| I5     | CylinderUp    |         |                               |
| I6     | ProductPlaced |         |                               |
| I7     | Error         |         |                               |

However, the FB is built in such a way that the desired press time and the number of pressings that have been performed can also be displayed.

| Analog inputs |         | Analog outputs |                 |
|---------------|---------|----------------|-----------------|
| IA1           | SetTime | QA1            | ActualPressings |

Furthermore, some digital and analog inputs and outputs have been allocated to the FB as reserve. It is not possible to subsequently modify the I/O of the FB unless a new version is created.

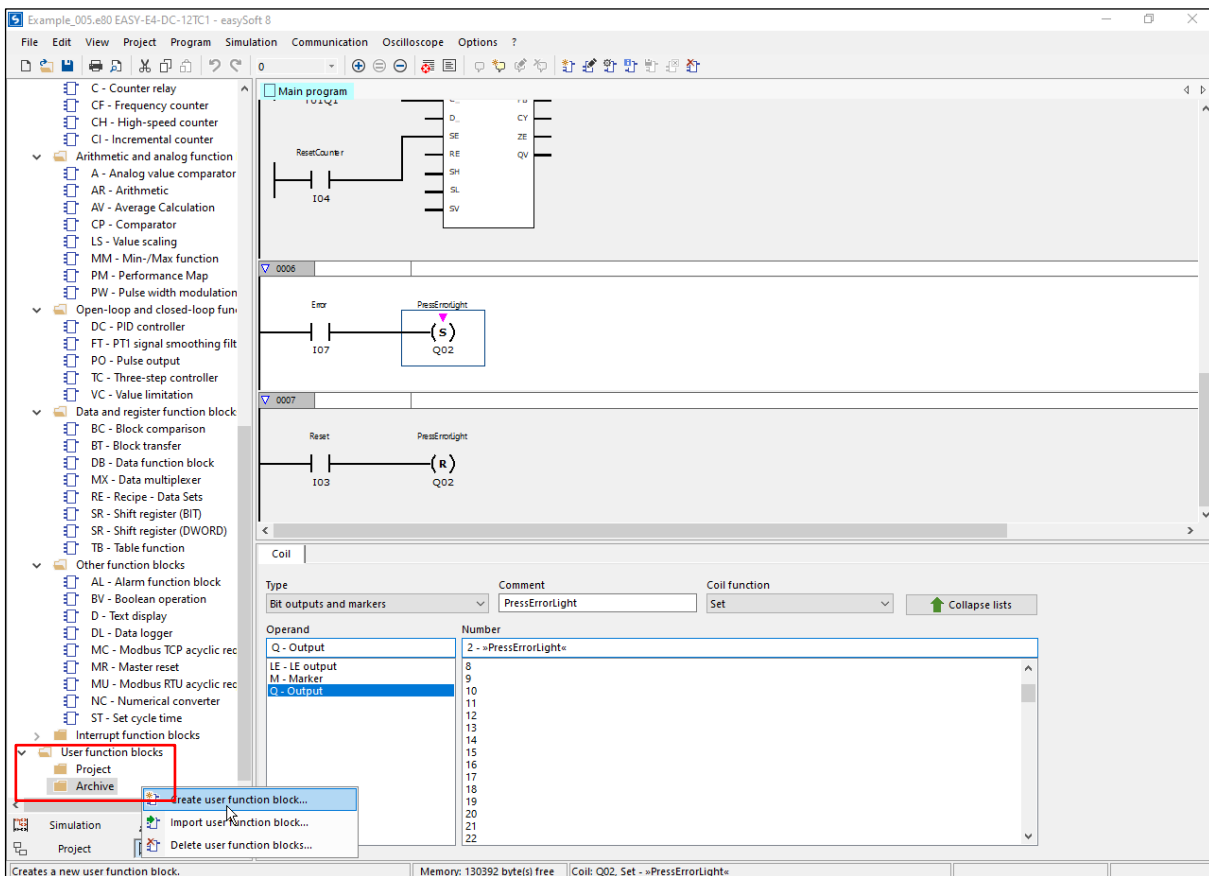


Fig. 60

- Go to "Programming".
- In the left of the tool box go to "User function blocks".

Here you have to choose between "Project" and "Archive". A User Function Block under "Project" can only be used in the current project. To also be able to use the custom-made User Function Block in other projects, it can be placed under "Archive".

- Choose "Archive" now.
- Open this folder with the right mouse button.
- Select the "Create user function block" here.

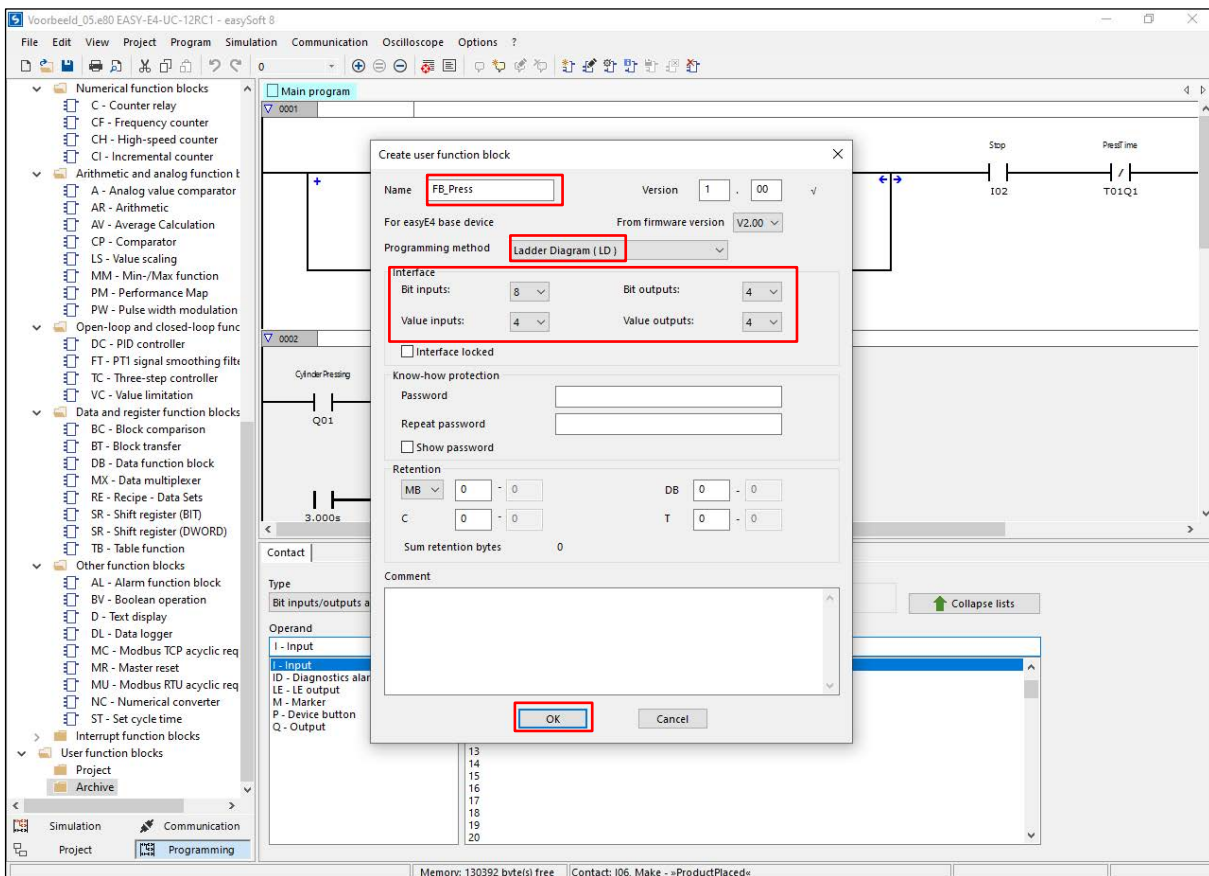


Fig. 61

Give the FB the name "FB\_Press" as devised in the graphical representation of the FB.

- Choose Ladder Diagram (LD) as the programming language.

Under "Interface", select the number of inputs and outputs. These have already been determined in the graphical example of the FB.

- Bit inputs = 8
- Value inputs = 4
- Bit outputs = 4
- Value outputs = 4
- Click "OK" to close.

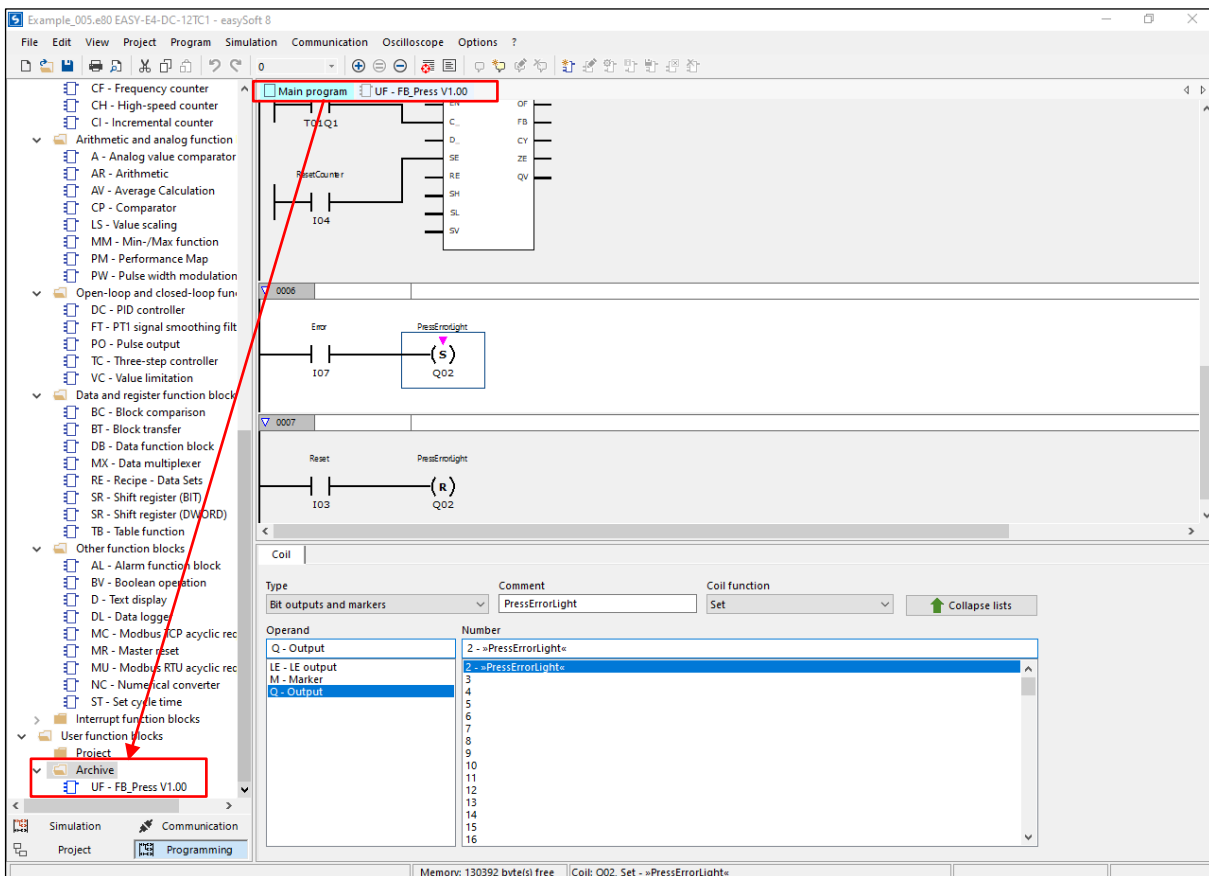


Fig. 62

The UF - FB\_Press V1.00 tab has appeared on the workspace next to the "Main program" tab.

- Click "Main program".

In the bottom left of the toolbox, the FB named "UF - FB\_Press V1.00" has appeared in the "Archive" folder. The FB is now available. However, the FB does not yet contain a program. The program is still in the "Main program".

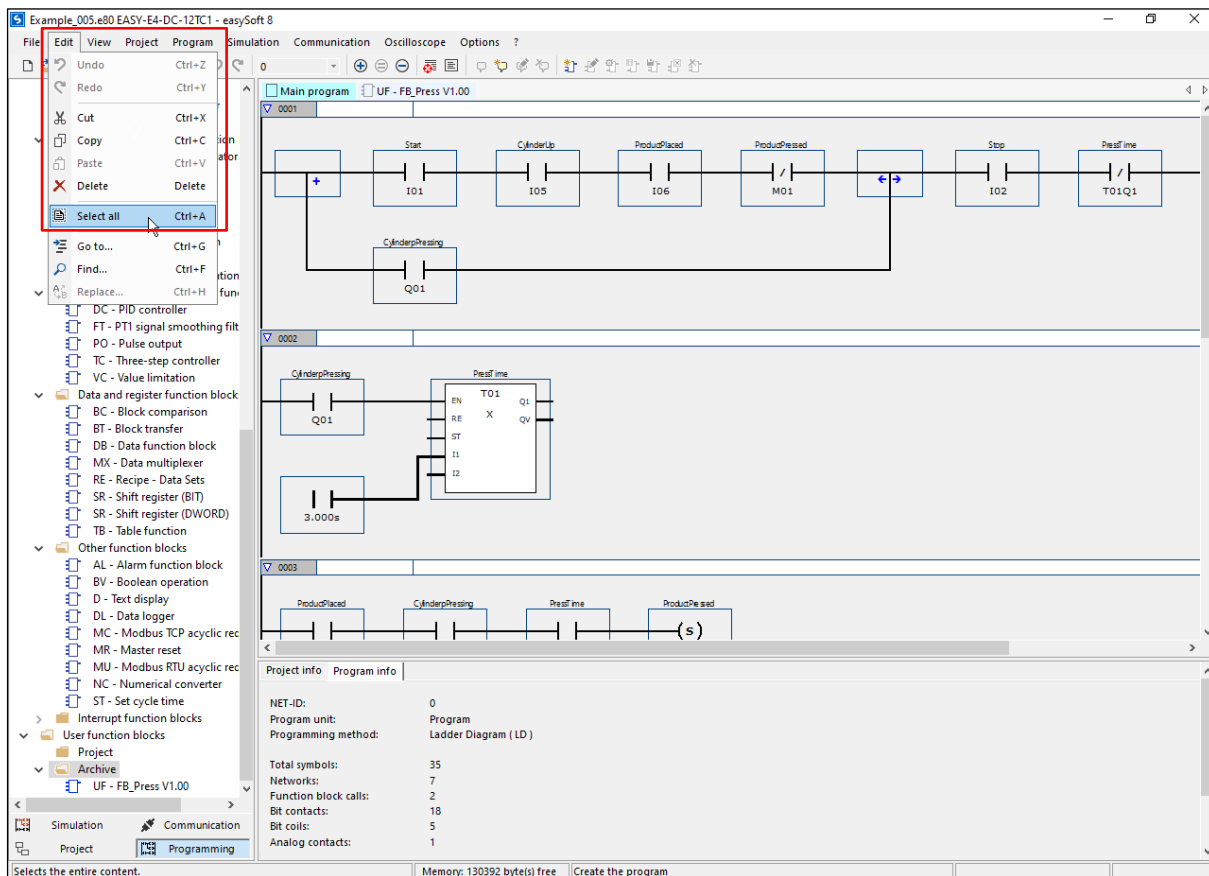


Fig. 63

- Go to "Main program".
- In the menu bar, go to "Edit" and select "Select all" (Ctrl+A) here.

Everything that has been programmed on the workspace is outlined with a blue border.

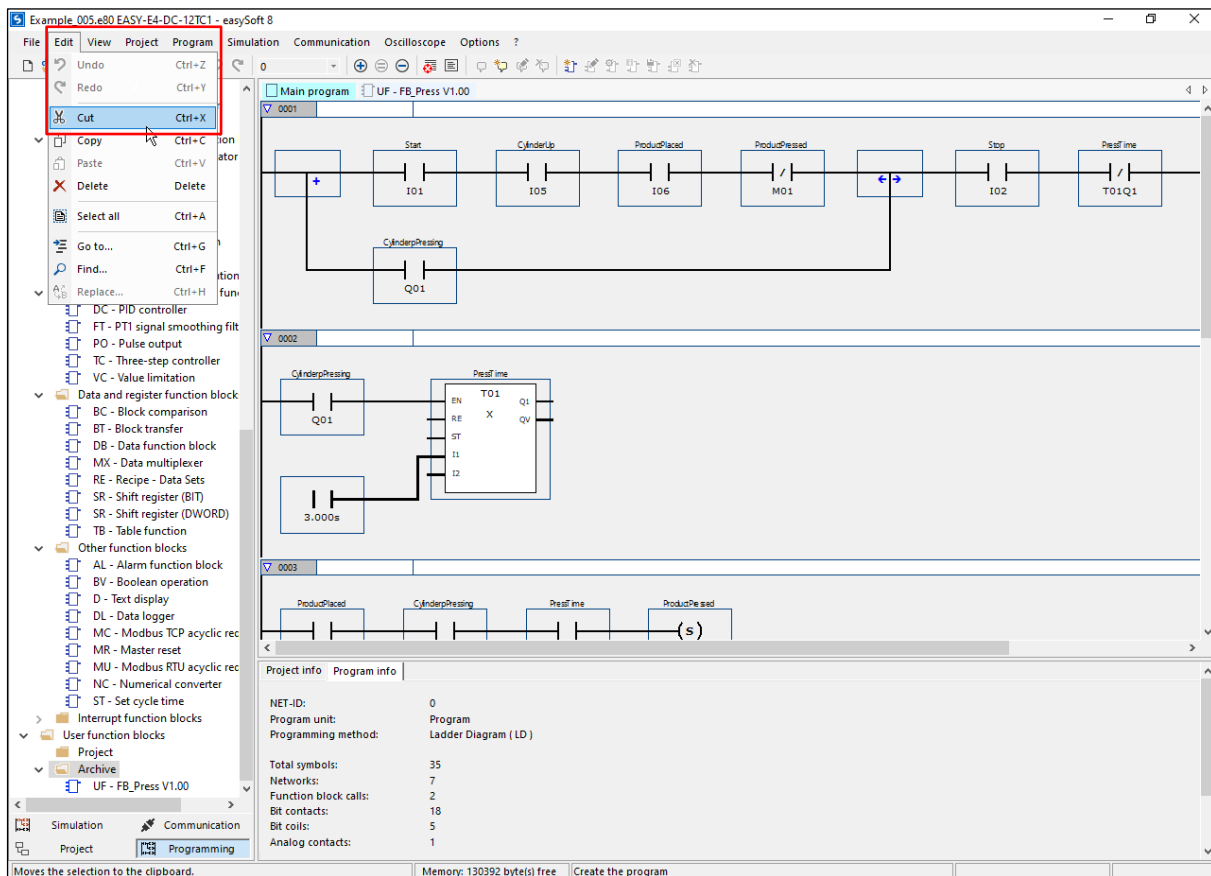


Fig. 64

The entire program in "Main program" can be cut out.

- In the menu bar, go to "Edit" and select "Cut" (Ctrl+X) here.

The entire program is cut from "Main program".

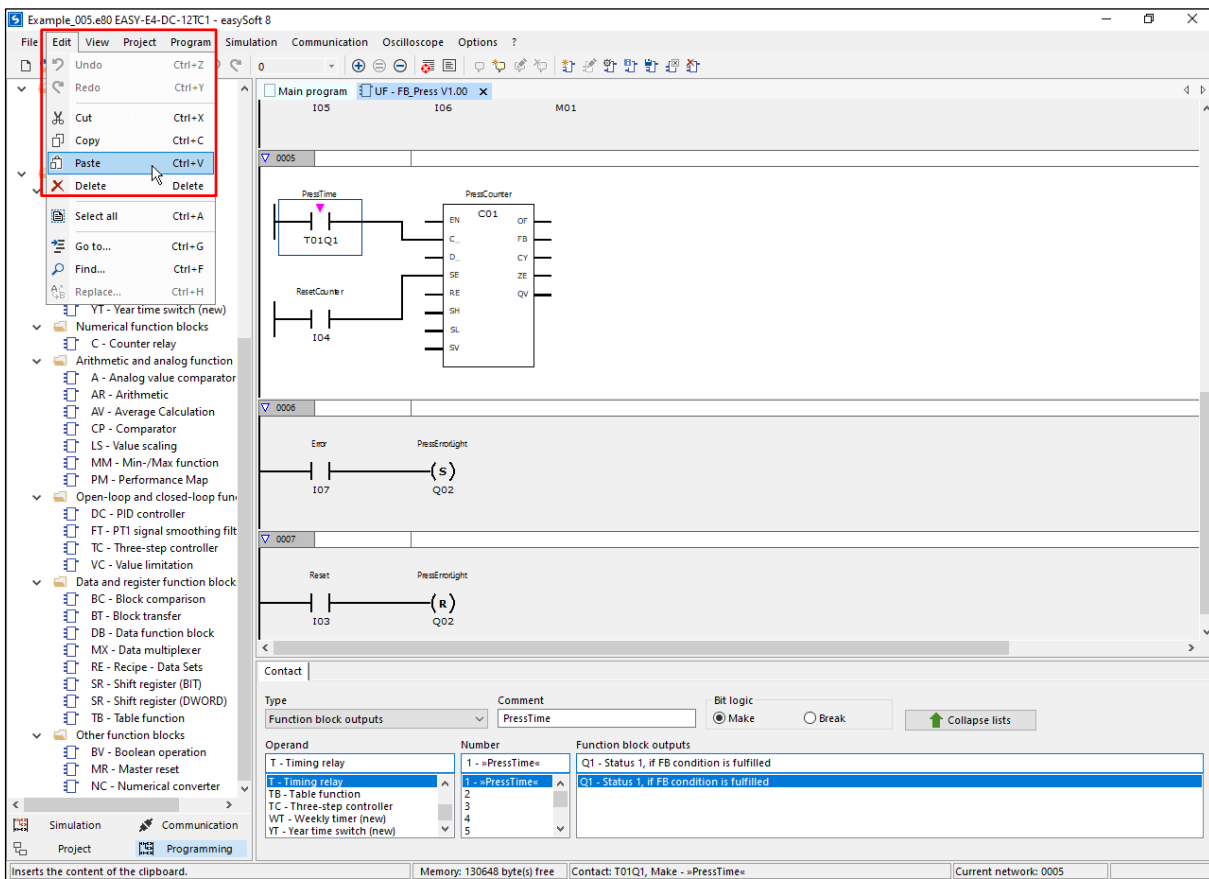


**Fig. 65**

- Go to the "UF – FB\_Press V1.00" tab.

An empty worksheet is displayed.

- In the menu bar, go to "Edit" and select "Paste" (Ctrl+V) here.

The cut program is now pasted into the FB "UF – FB\_Press V1.00".

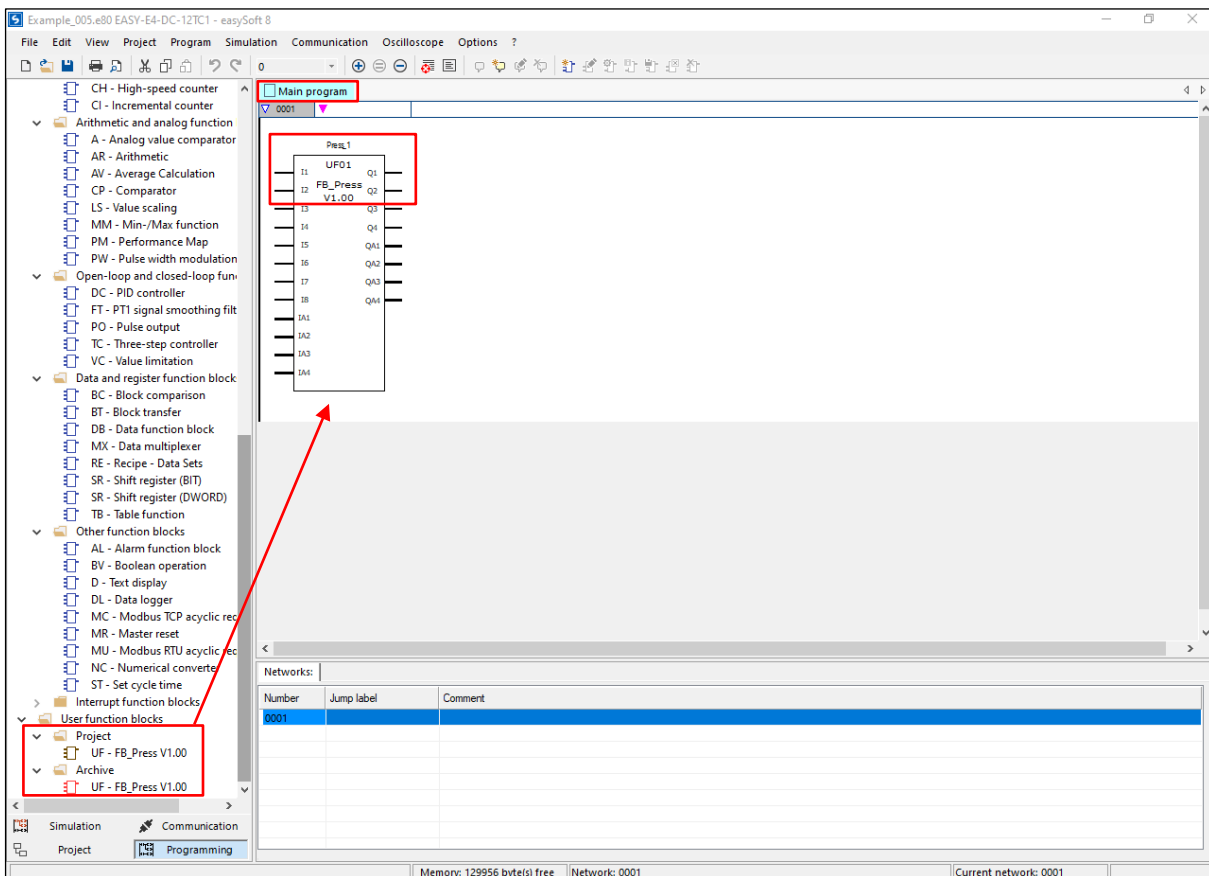


Fig. 66

- Go to the "Main program" tab.

There are no program rungs left here.

- Under "User function block", drag the custom-made FB named "UF – FB\_Press V1.00" onto the workspace.
- Name the FB "Press\_1".

The inputs and outputs only have an designation with I1, Q1, IA1, and so on.

By hovering the mouse over the inputs and outputs, the name belonging to the respective input or output appears.

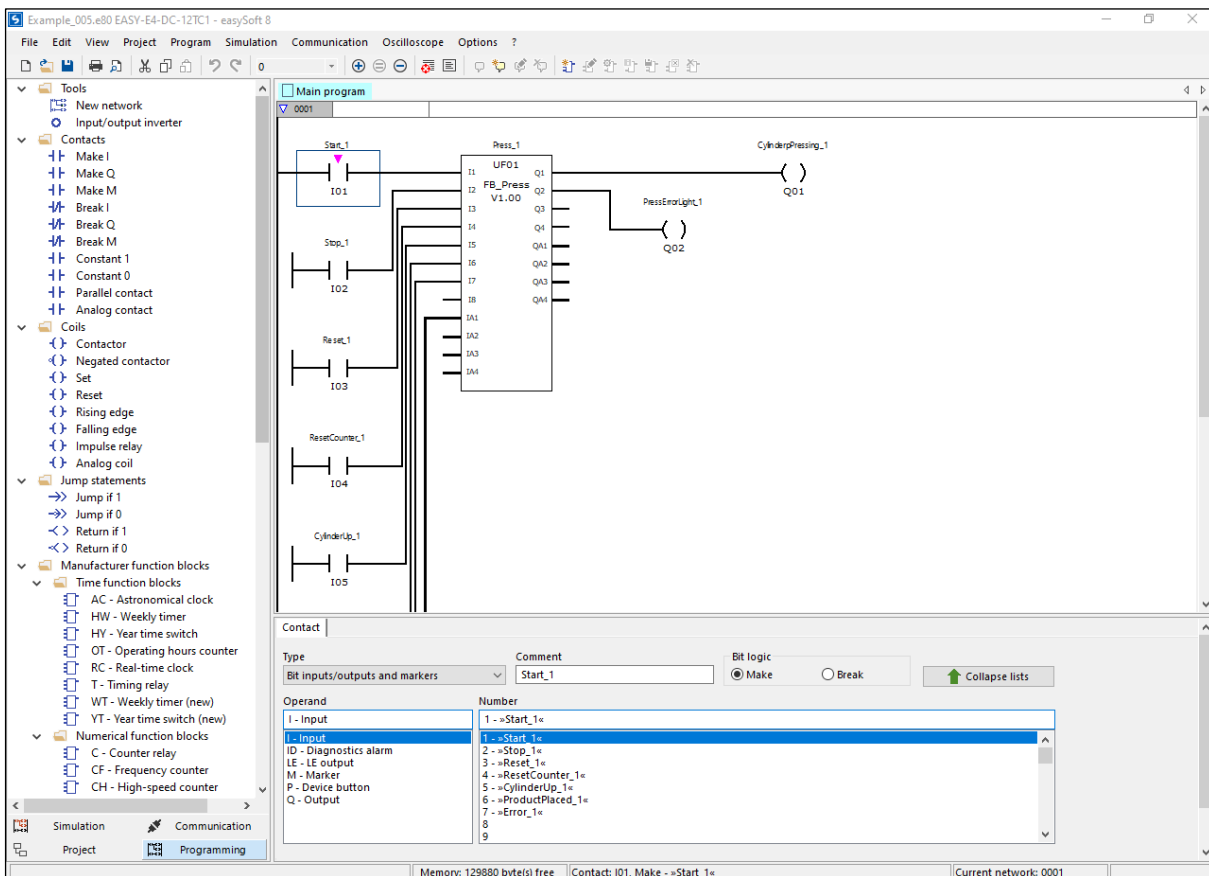


Fig. 67

The FB can now be provided with the correct I/O, just like a timer or counter.

Input I1 of the FB\_Press is named "Start". This is because this name was already assigned to I1 in the "Main program" and was copied over.

If multiple presses are controlled by the system, it is advisable not to just name the inputs **and outputs on the easyE4 "Start" but, for example, "Start\_1"** to indicate that this "Start" belongs to the control of press 1.

- Give the inputs and outputs on the easyE4 itself the following designations.

| Inputs |                 | Outputs |                                 |
|--------|-----------------|---------|---------------------------------|
| I1     | Start_1         | Q1      | CylinderPressing_1 (monostable) |
| I2     | Stop_1          | Q2      | PressErrorLight_1               |
| I3     | Reset_1         |         |                                 |
| I4     | ResetCounter_1  |         |                                 |
| I5     | CylinderUp_1    |         |                                 |
| I6     | ProductPlaced_1 |         |                                 |
| I7     | Error_1         |         |                                 |

In the FB itself, the names should therefore **NOT** be adjusted.

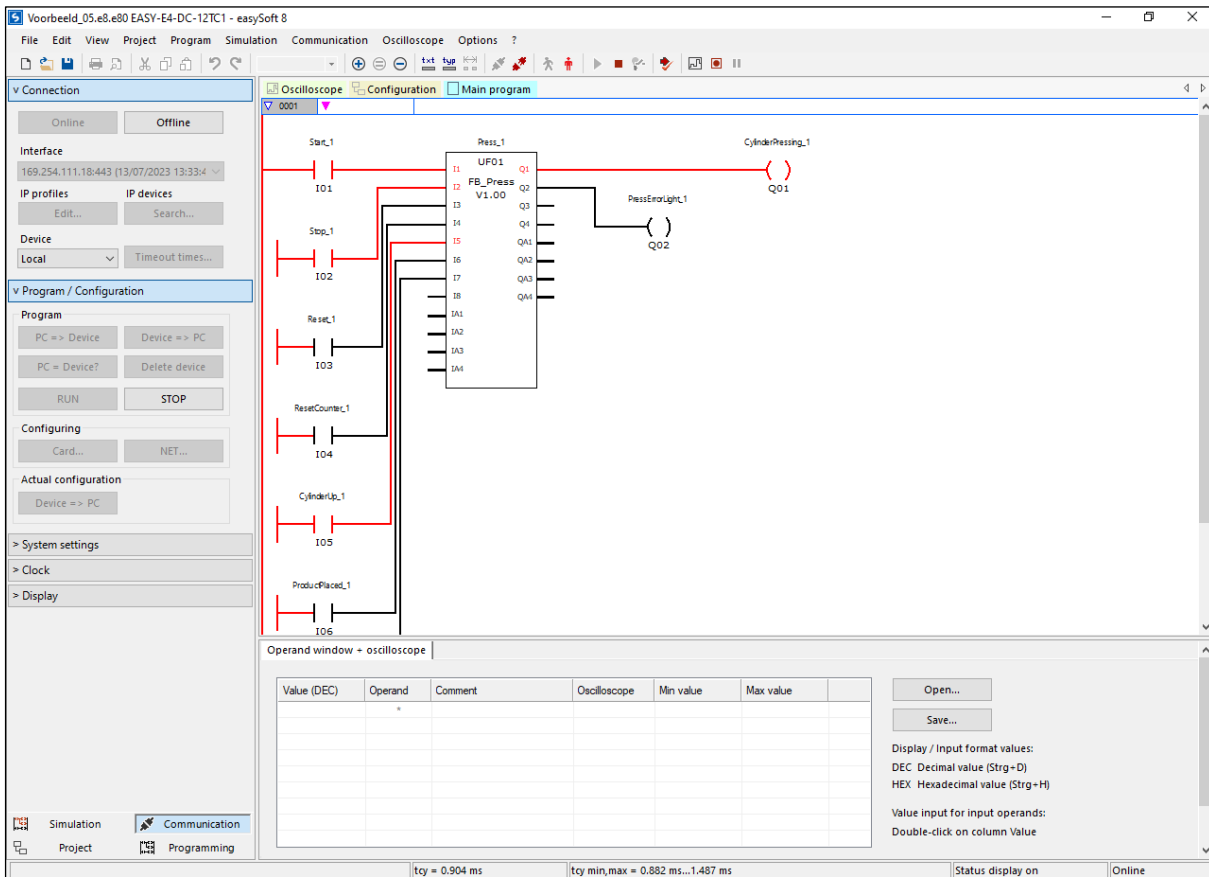


Fig. 68

- Upload the program to the easyE4 and test the function.

This will now work in exactly the same way, whereby the entire program is processed in the FB.

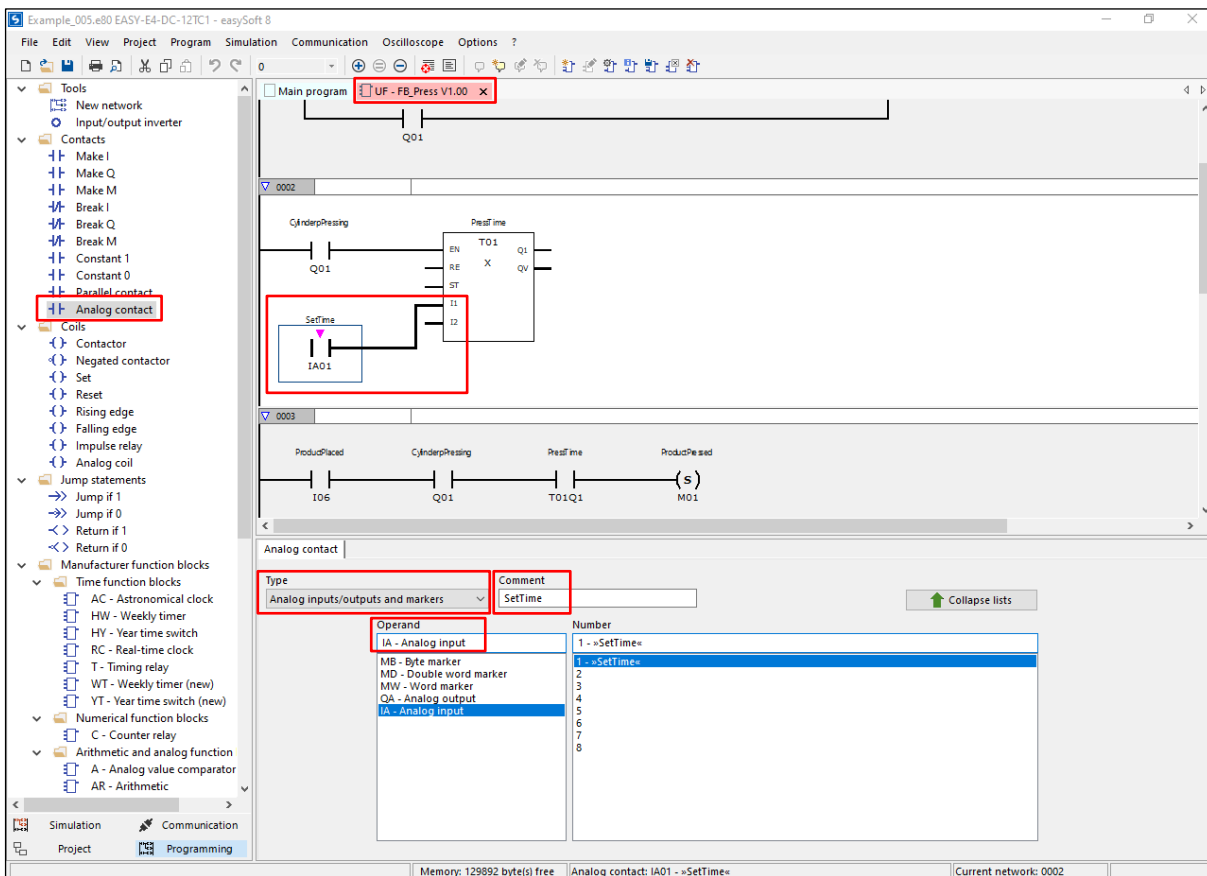


Fig. 69

The press time is now a fixed time. It must now be possible to adjust this. The analog input IA1 of FB\_Press is used for this.

- Go to the "UF – FB\_Press V1.00" tab.
- Replace the fixed time on the "PressTime" timer in FB\_Press with analog input 1 (IA1) and name it "SetTime".

If the tab "UF – FB\_Press V1.00" is not visible, then go to the toolbox and double-click on "UF – FB\_Press V1.00" under "Archive".

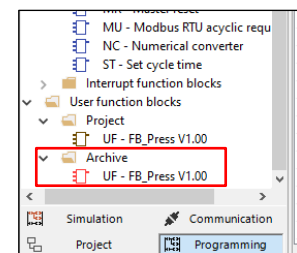


Fig. 69A

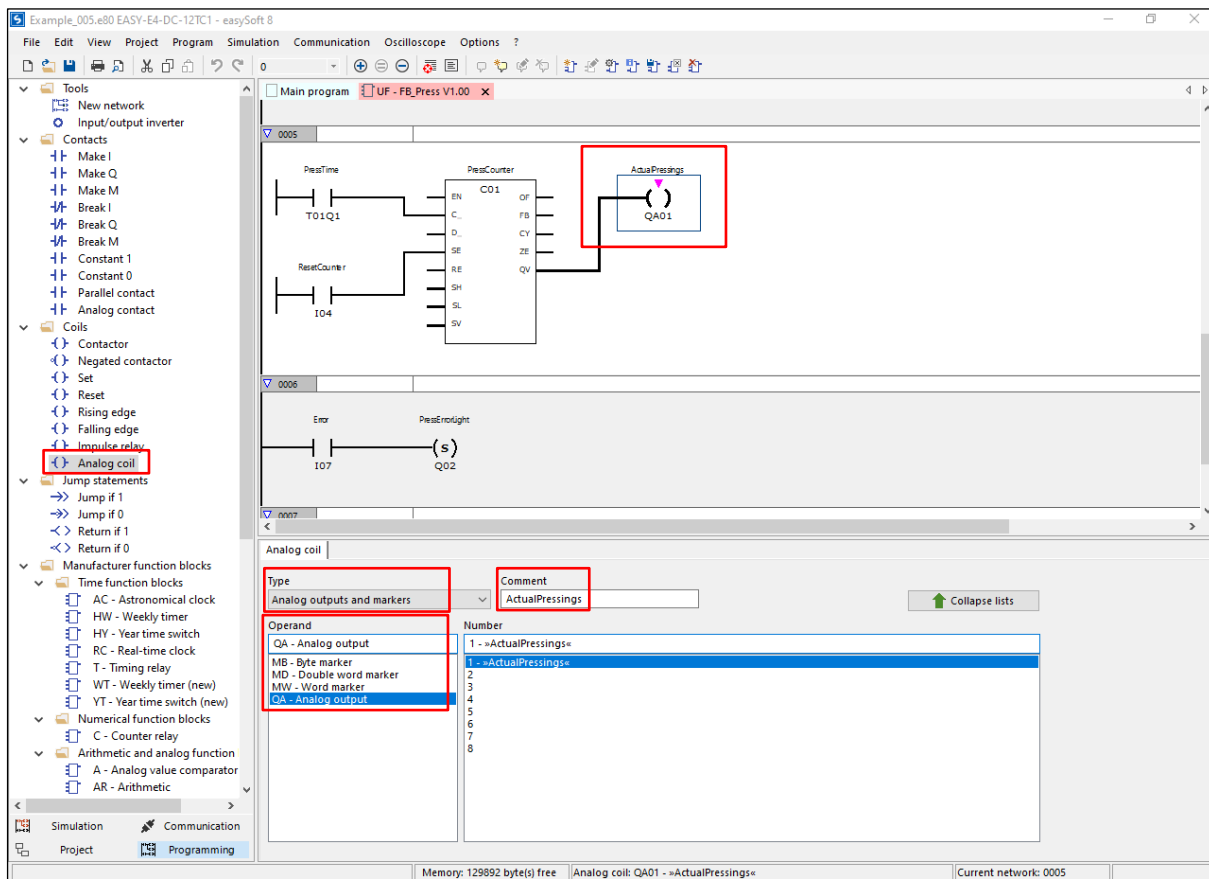


Fig. 70

The number of pressings is now only counted but is not yet linked to an output of the FB.

- Place an "Analog coil" on output QV of "PressCounter" and name it "ActualPressings".

The number of pressings is now available on output QA1 of "FB-Press".

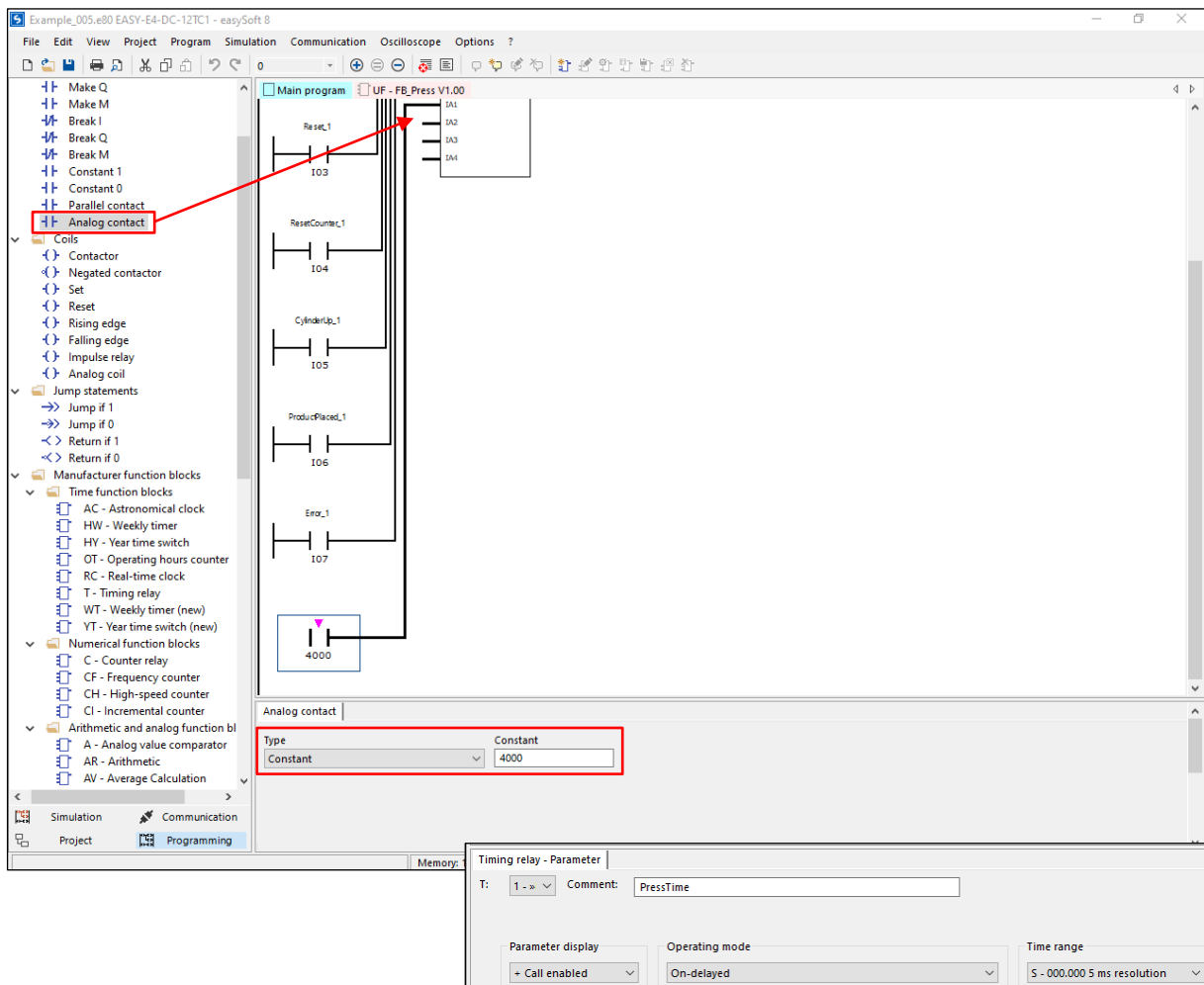


Fig. 71

- Go to "Main program" and place an analog contact on input IA1 of "Press\_1".
- Select "constant" for type because a fixed value is placed on the timer.
- Give this the value 4000.

The value 4000 corresponds to 4000 ms. In this program, a timer with the time value seconds and milliseconds has been chosen.

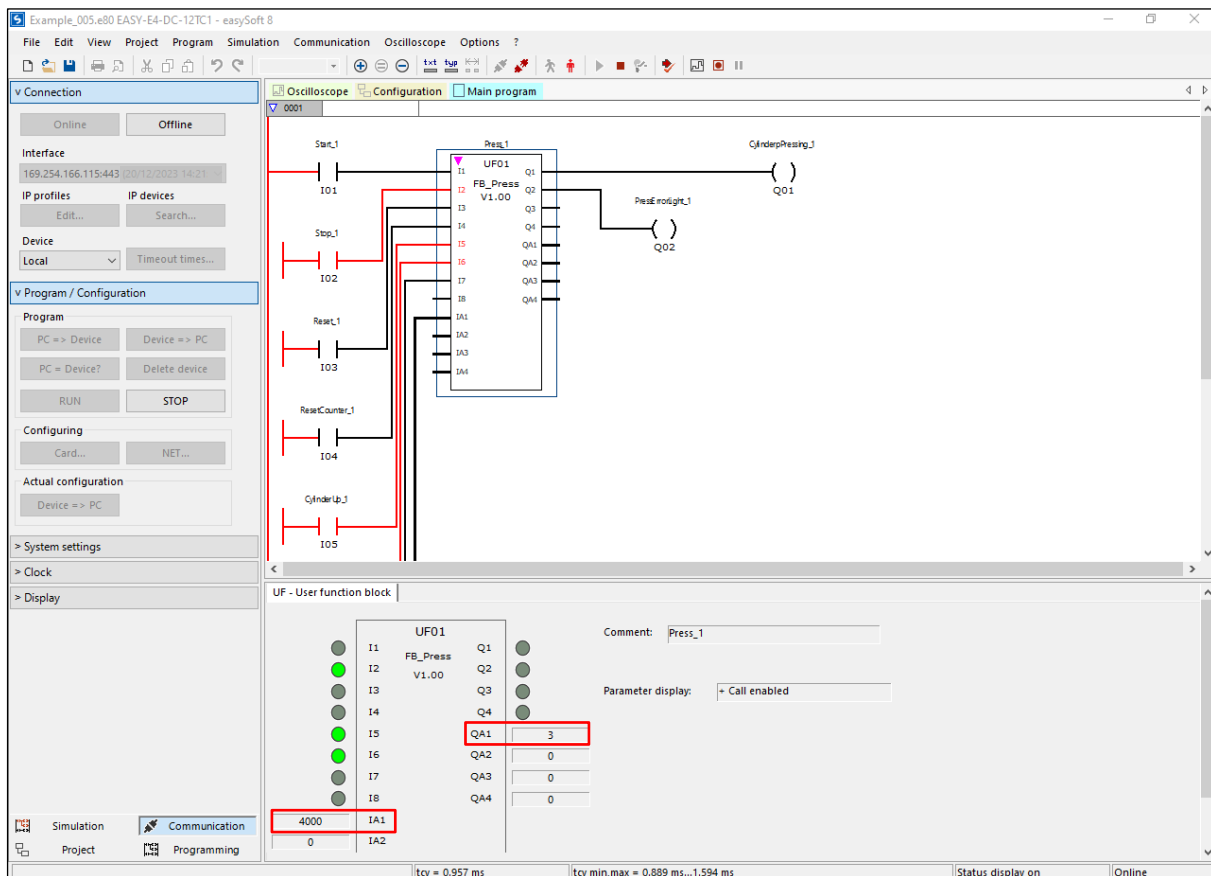


Fig. 72

- Save the program and send it to the easyE4.

Test the function. The press time is now 4 seconds. The number of pressings is made visible via output QA1 of "Press\_1".

## 2.11 Export function block

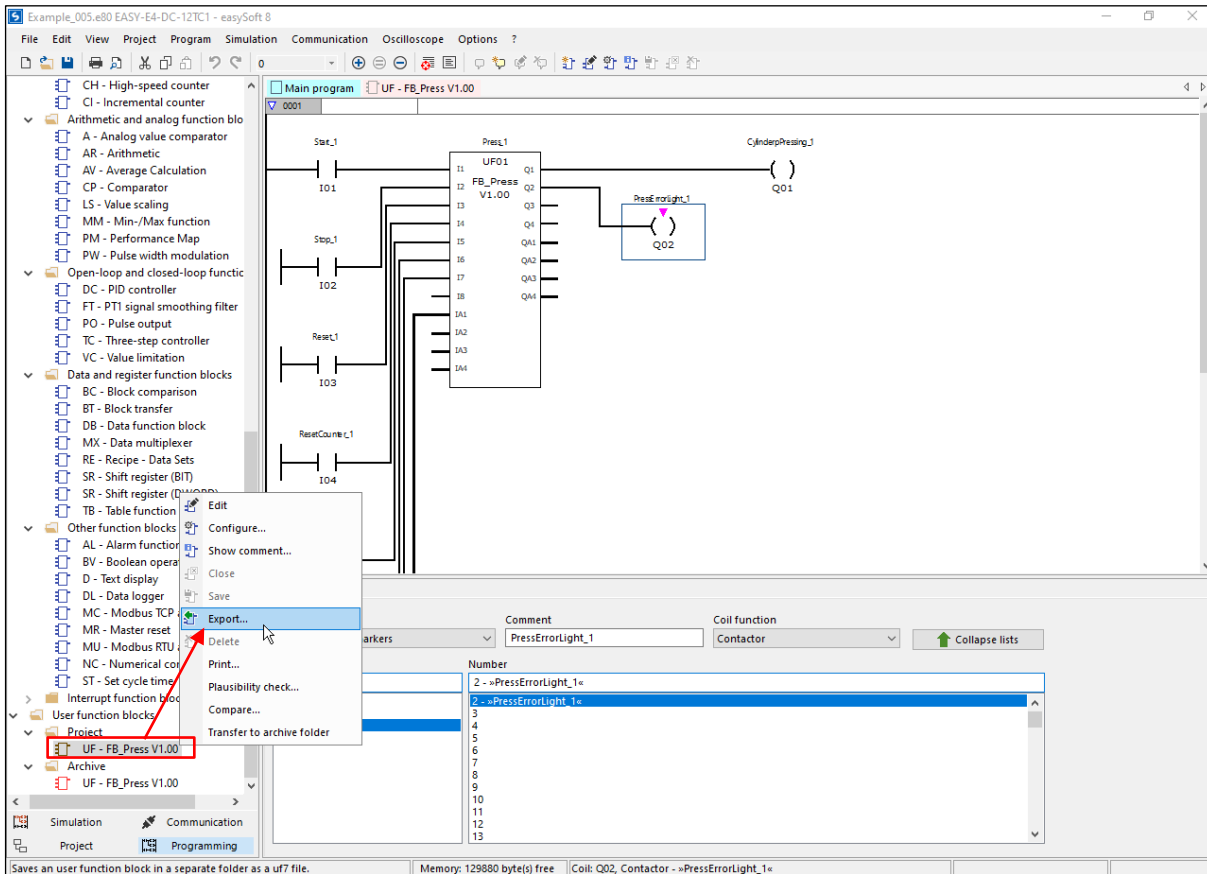


Fig 73

To use an FB in another program, the FB must be exported.

- Go to the "Programming" menu and to the "Main program" tab.
- Select the relevant FB under "Project" and right-click on it. Select "Export...".

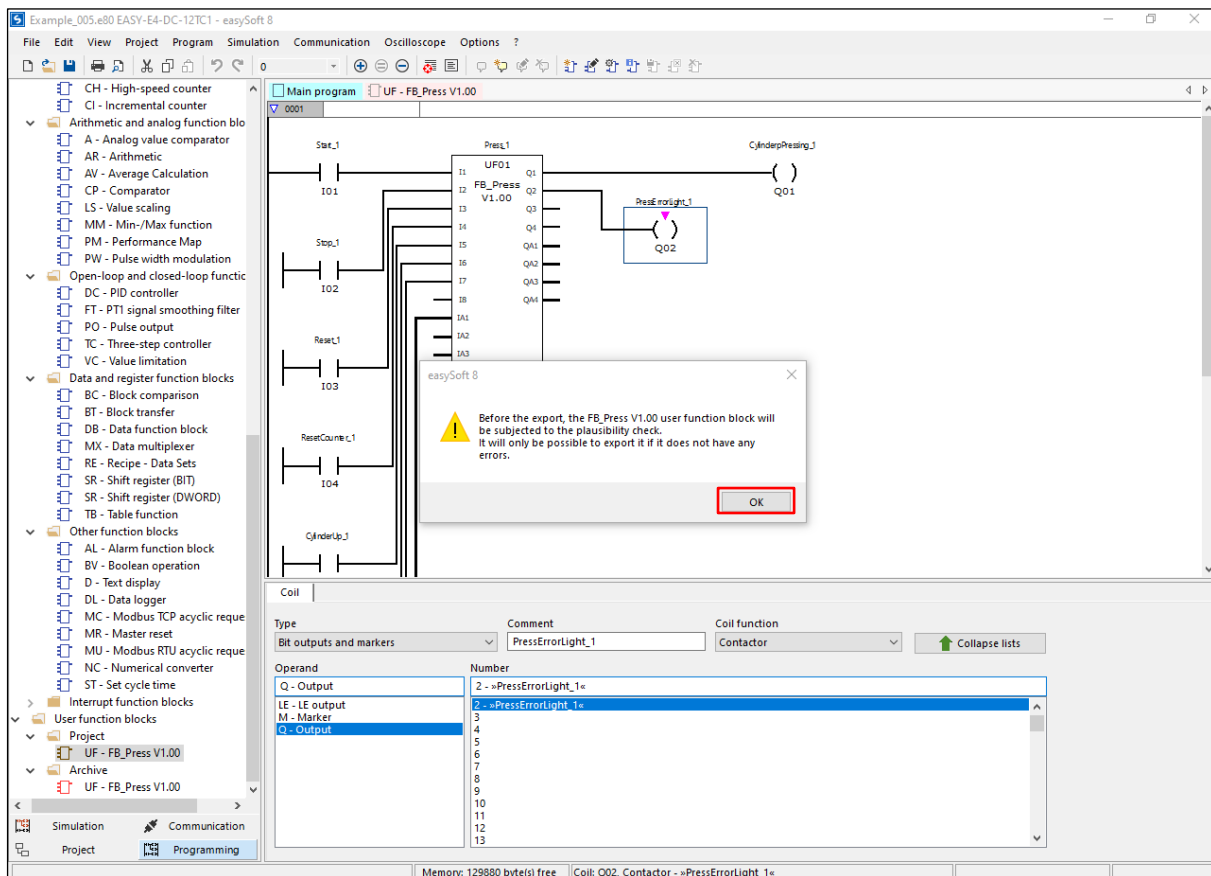


Fig. 74

The first screen that now appears indicates that the FB is being checked for syntax errors.

- Confirm this with "OK".

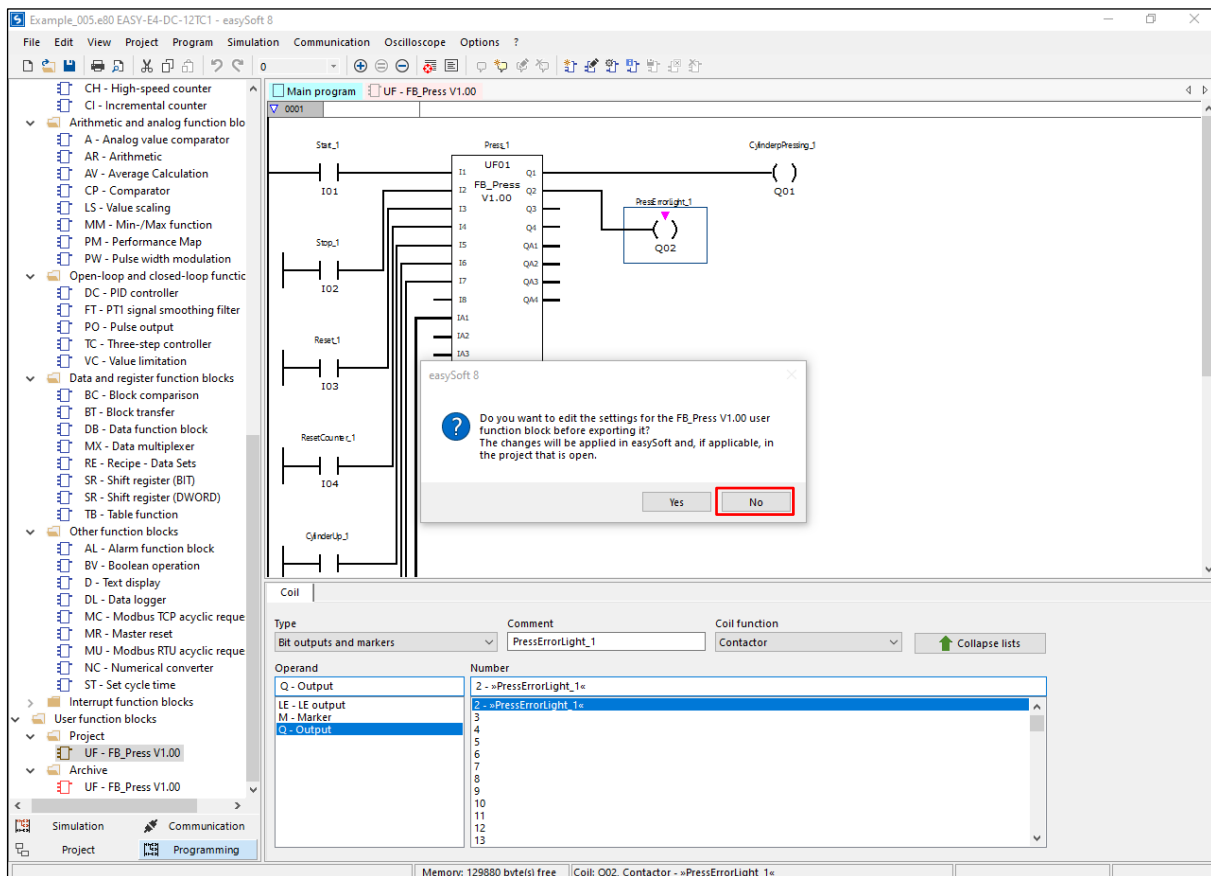


Fig. 75

The second screen that appears asks if anything still needs to be modified in the FB. If nothing needs to be modified, as in this example, you can confirm this with “No”.

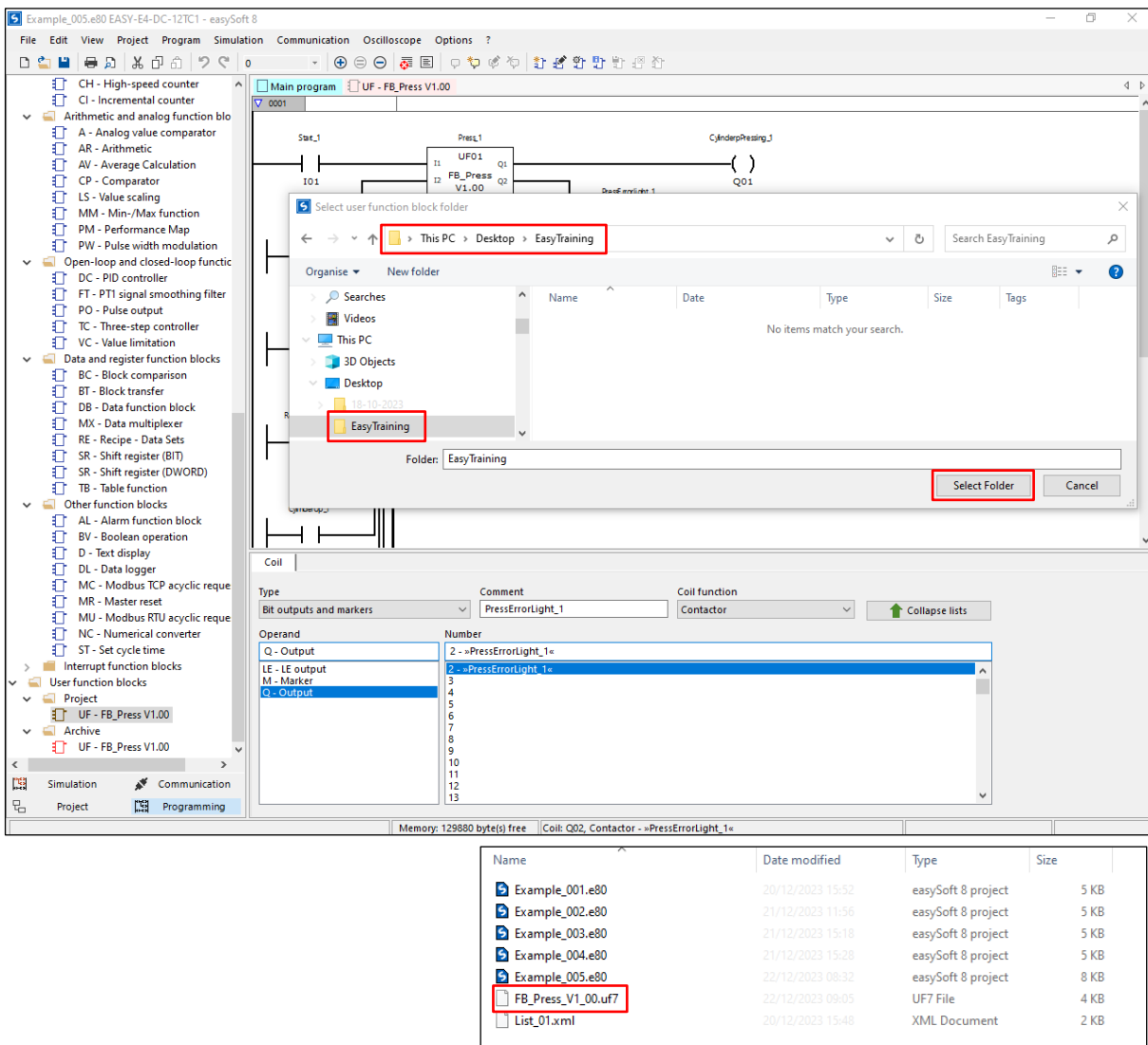


Fig. 76

The final screen asks where the FB should be saved.

- Select the "EasyTraining" folder. And click on "Select Folder".

The folder now contains not only the programs that have been created, but also the FB named "FB\_Press\_V1\_00".

It is now possible to import this FB into other programs.

## 2.12 Program in Function Block Diagram

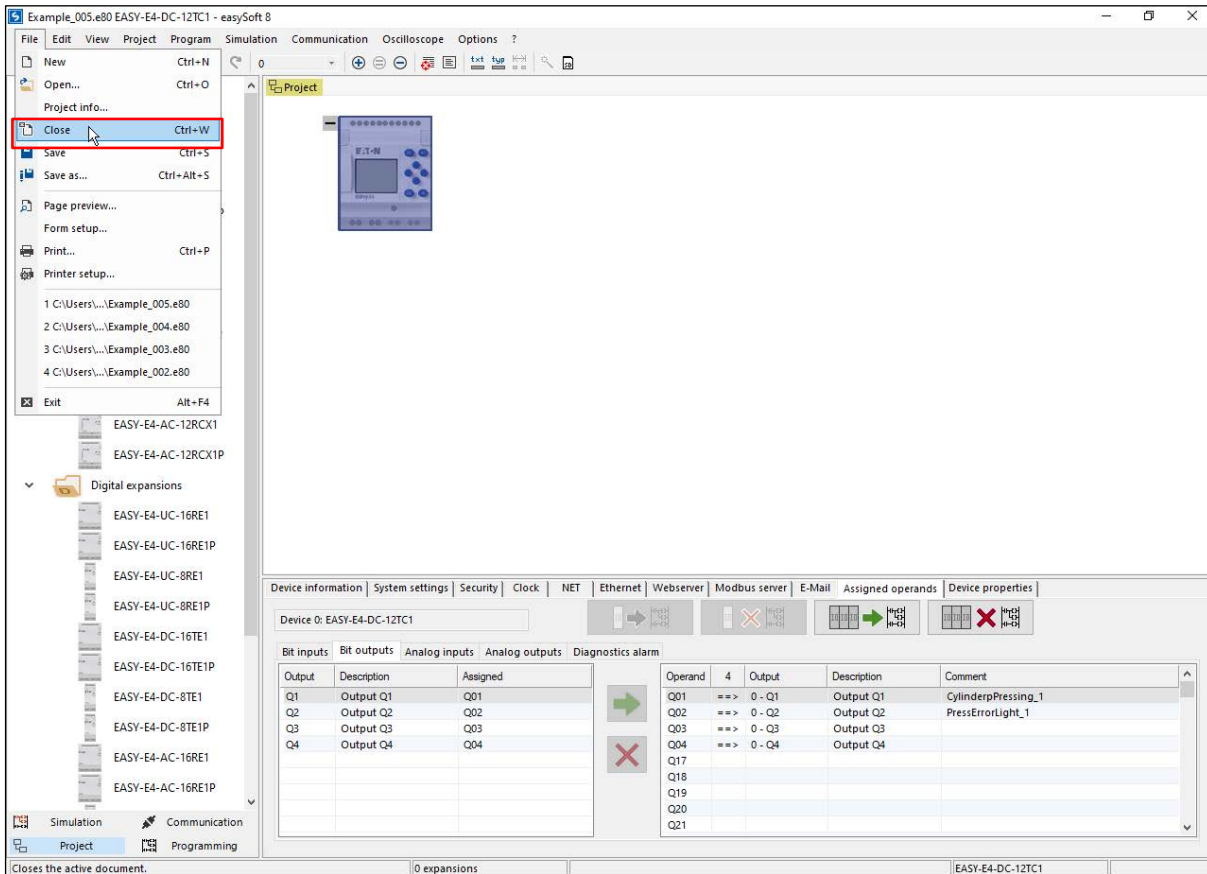


Fig. 77

The program and the FB are written in LD (Ladder Diagram). The easyE4 can, however, be programmed with multiple programming languages, including FBD (Function Block Diagram).

- In the menu bar, go to "File" → "Close" and close the current program.

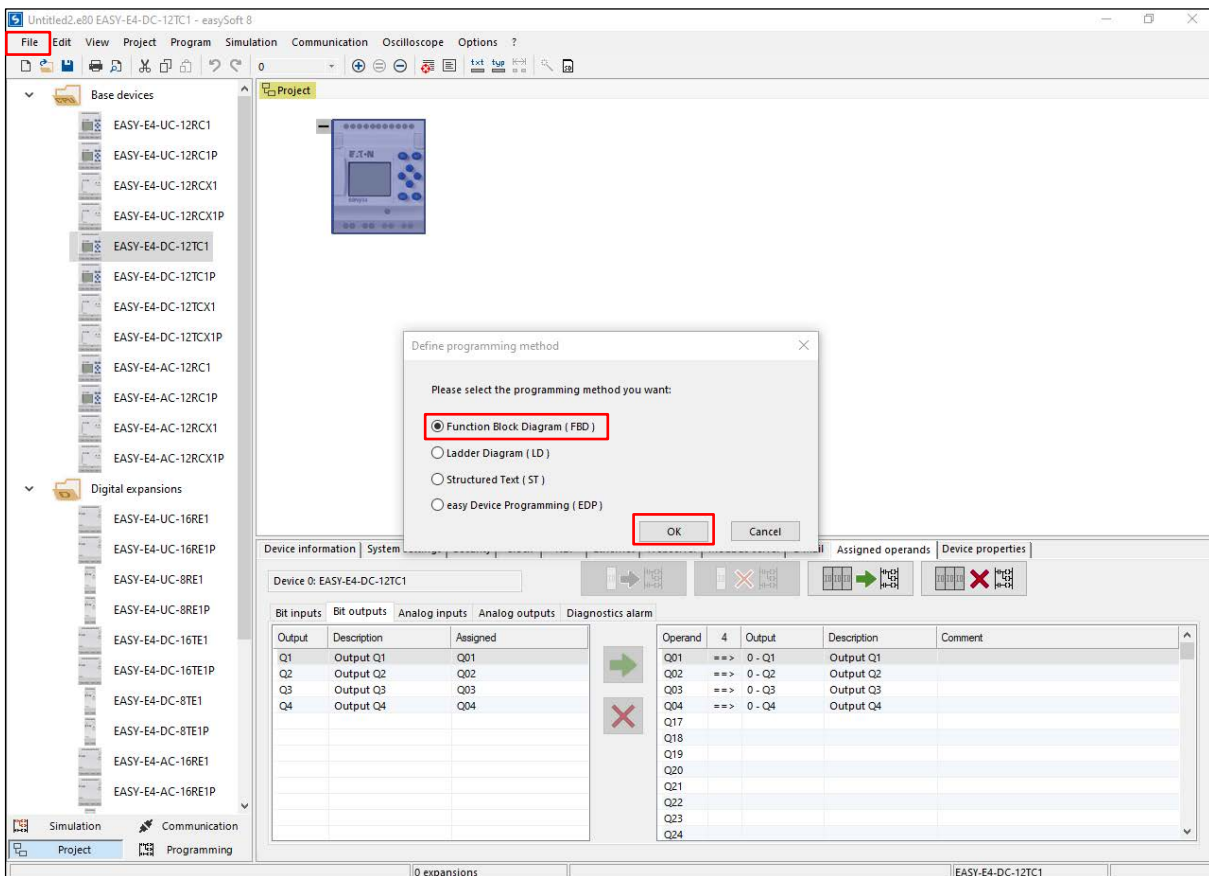


Fig. 78

- Open a new program.
- Go to "File" → "New" in the menu bar.

A new easySoft 8 program opens.

- Drag an EASY-E4-DC-12TC1 to the worksheet and select the desired FW version.
- Go to the "Programming" menu.
- Here, select the programming language "Function Block Diagram" and confirm this with "OK".

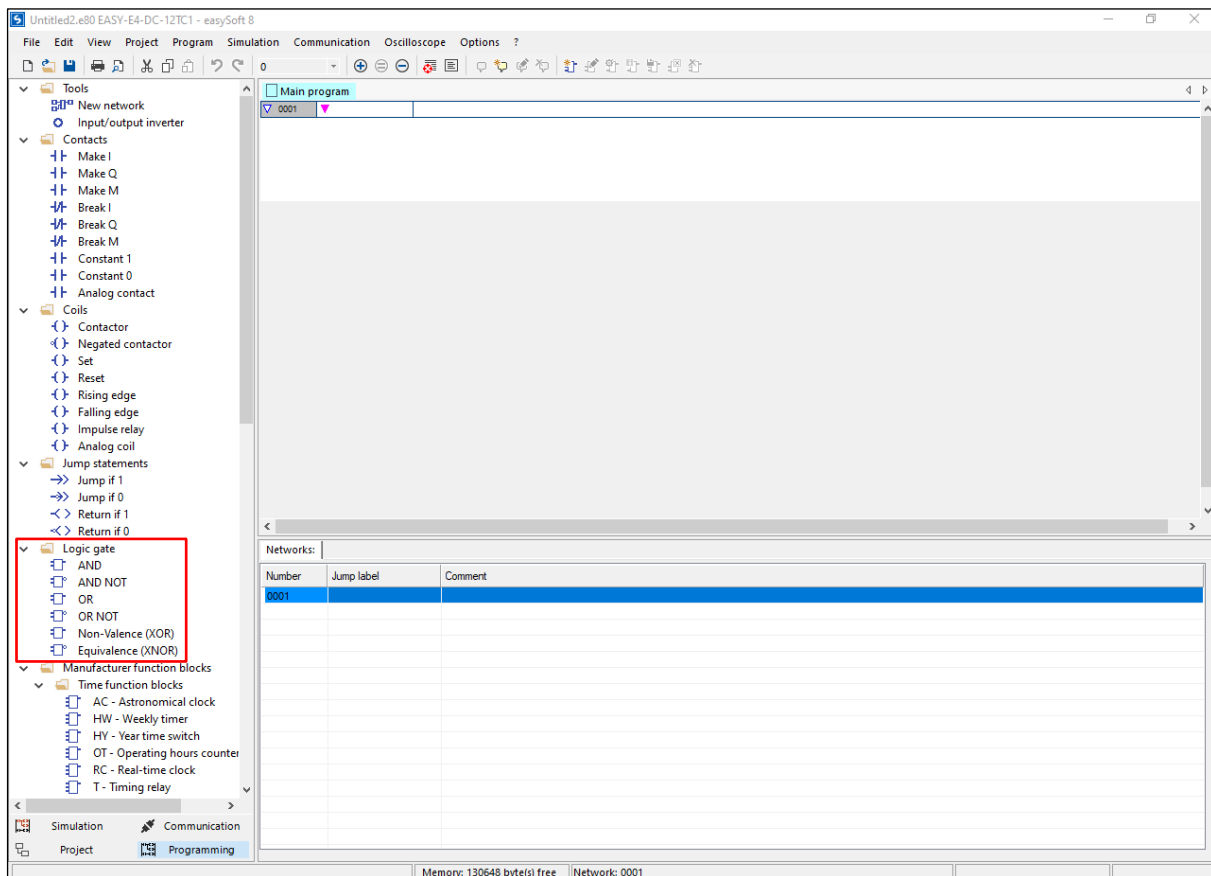


Fig. 79

The most striking thing is that logical gates have now become available in the toolbox.

- Save the program under the name "FBD\_Example\_01".

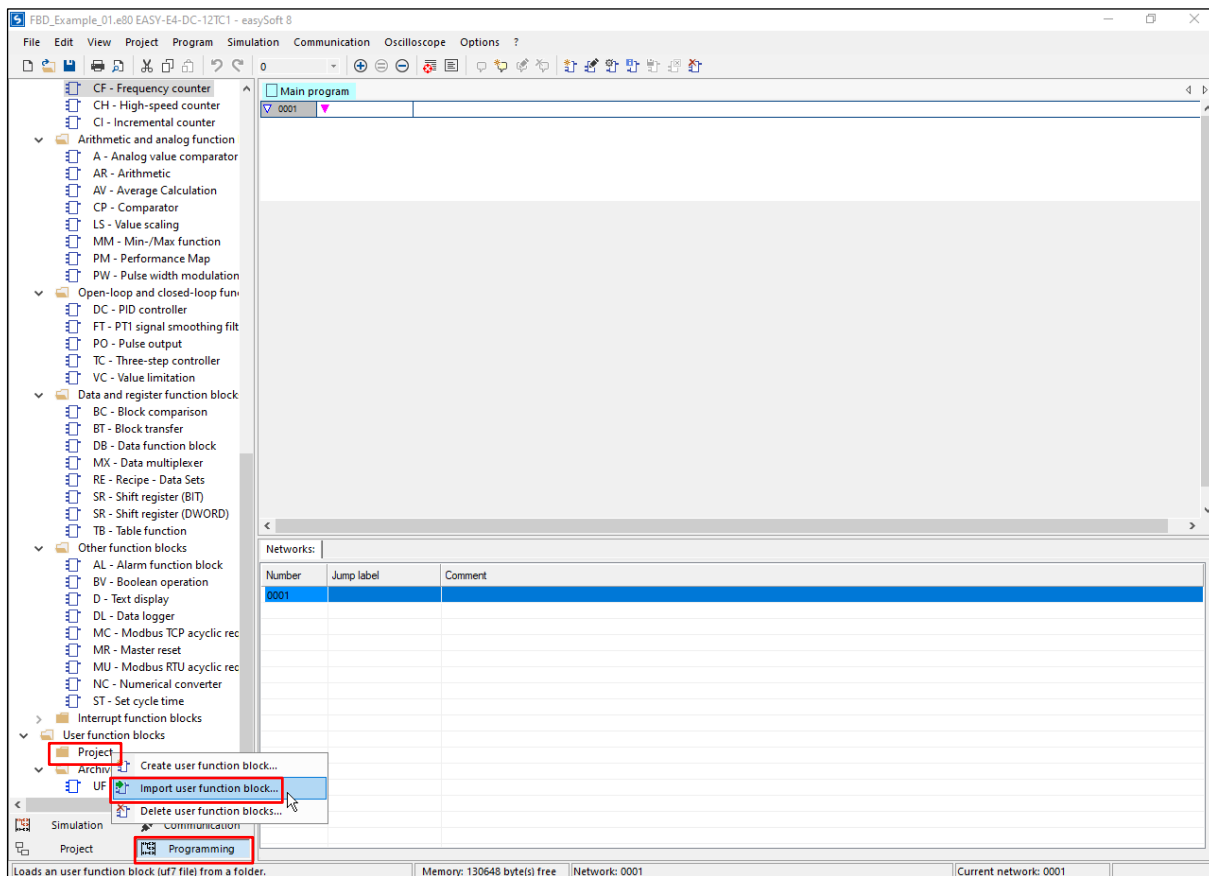


Fig. 80

- In "Programming" go to "User Function Blocks".
- Right-click on "Project" and select "Import user function block...".

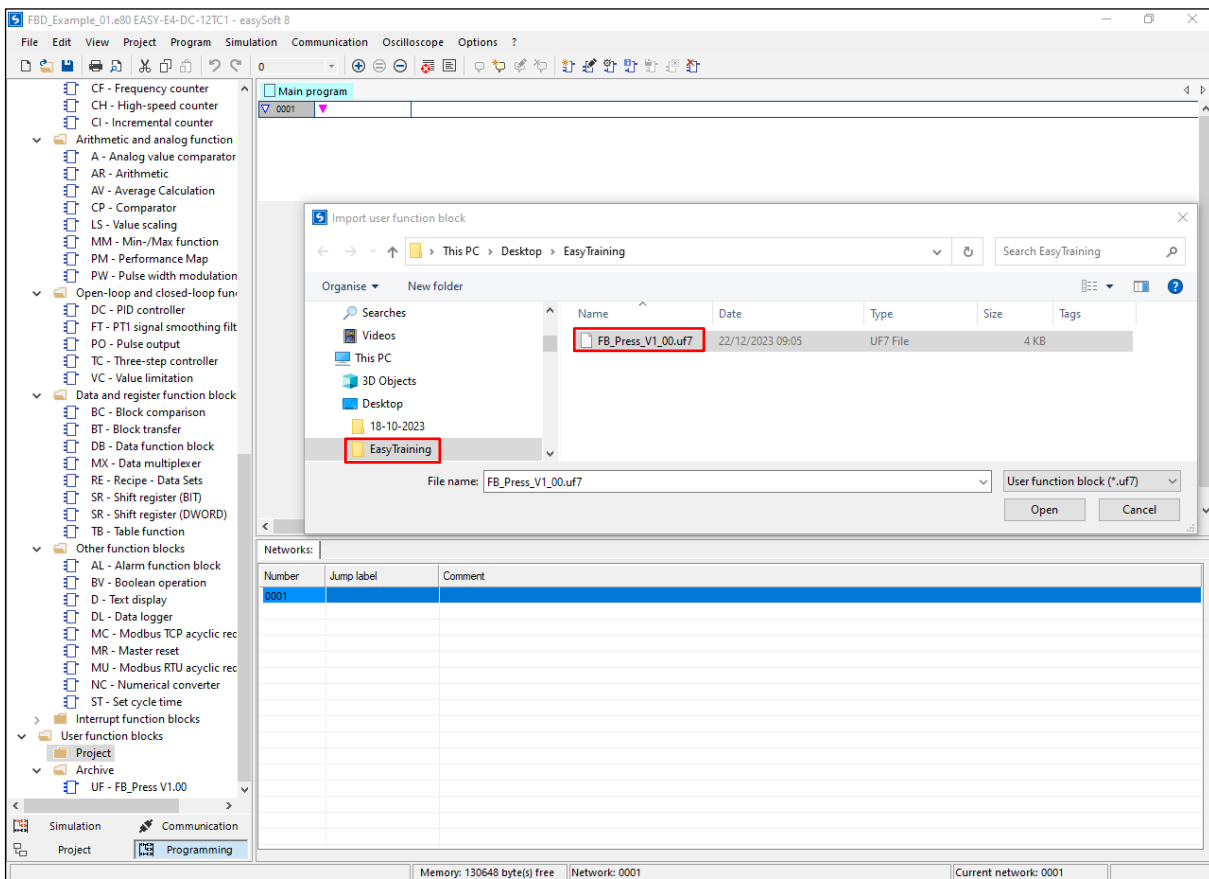


Fig. 81

- Go to the "EasyTraining" folder.
- Select the User Function Block with the name "FB\_Press\_v1\_00.uf7" here and click "Open".

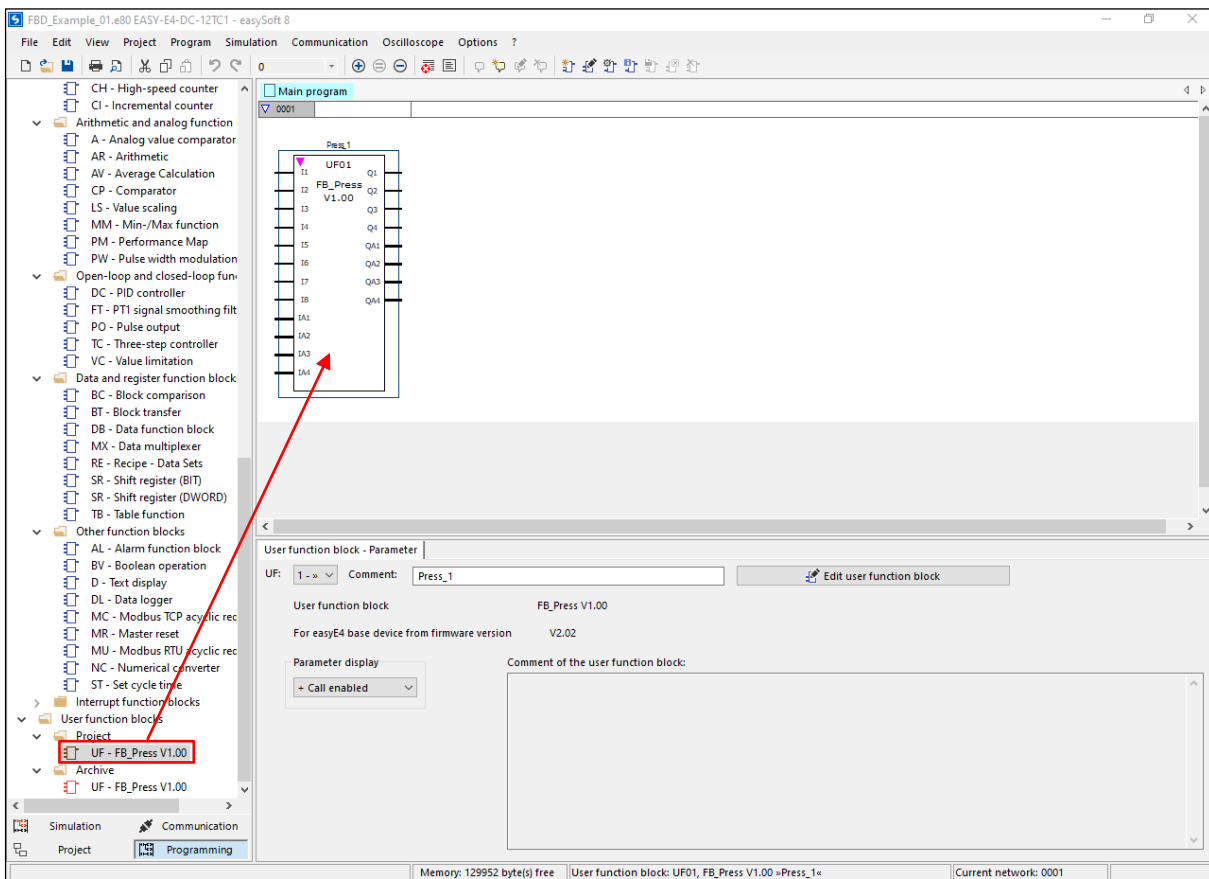


Fig. 82

- Drag the custom-made FB onto the workspace.
- Call this "Press\_1".

Provide the FB with the inputs and outputs as also described previously for programming in LD (Ladder).

The following I/O are to be used for the program.

#### Inputs

I1 Start\_1  
I2 Stop\_1  
I3 Reset\_1  
I4 ResetCounter\_1  
I5 CylinderUp\_1  
I6 ProductPlaced\_1  
I7 Error\_1

#### Outputs

Q1 CylinderPressing\_1 (monostable)  
Q2 PressErrorLight\_1

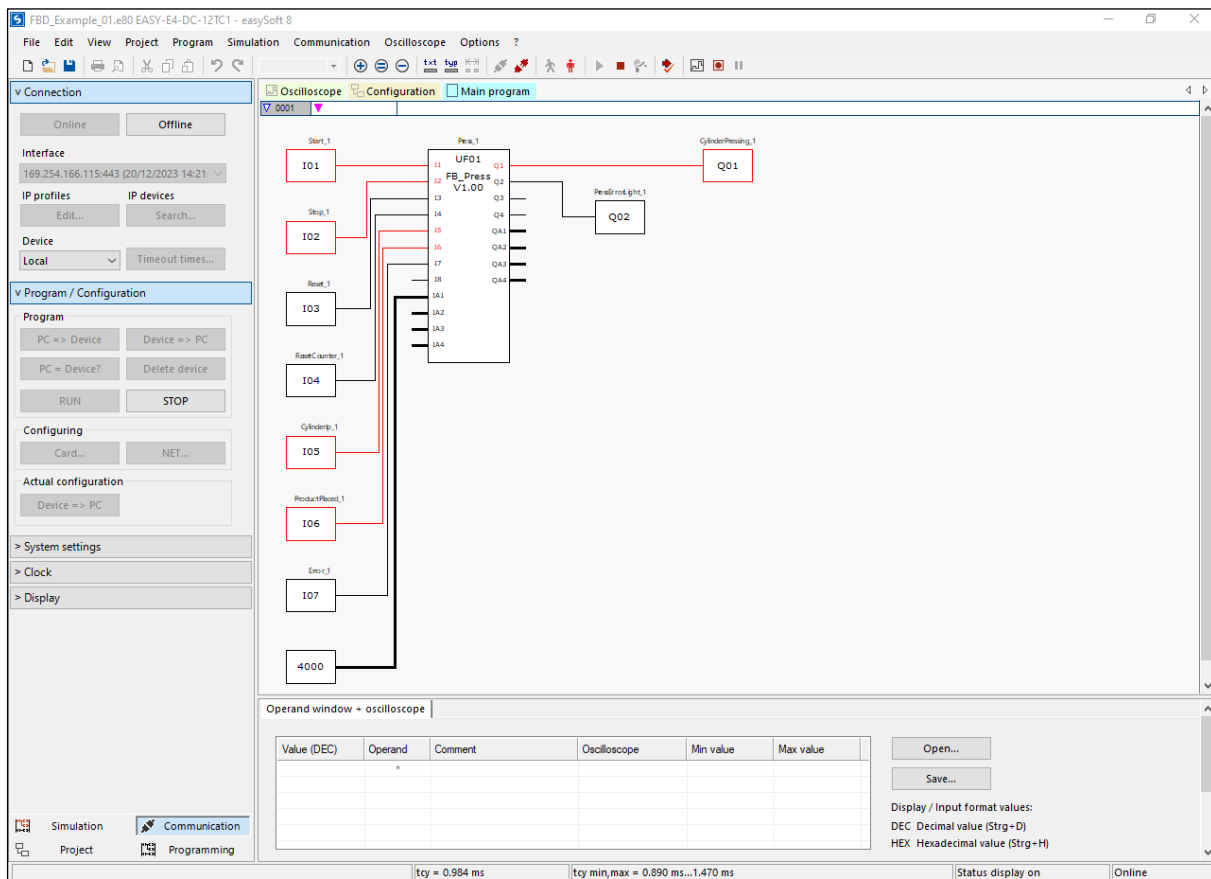


Fig. 83

- Send the program to the easy and test its operation.

This now works the same as the program programmed in LD.

## 2.13 Add I/O-module



Fig. 84

A second press must be controlled. For a second press, an I/O block must be added to the hardware to obtain sufficient inputs and outputs.

For the expansion, an EASY-E4-DC-16TE1 is used.

The extended I/O list is shown below. The I/O names have been adjusted to distinguish between starting press 1 and press 2.

### Press 1

|    |                 |    |                                 |
|----|-----------------|----|---------------------------------|
| I1 | Start_1         | Q1 | CylinderPressing_1 (monostable) |
| I2 | Stop_1          | Q2 | PressErrorLight_1               |
| I3 | Reset_1         |    |                                 |
| I4 | ResetCounter_1  |    |                                 |
| I5 | CylinderUp_1    |    |                                 |
| I6 | ProductPlaced_1 |    |                                 |
| I7 | Error_1         |    |                                 |

### Press 2

|    |                 |    |                                 |
|----|-----------------|----|---------------------------------|
| I1 | Start_2         | Q5 | CylinderPressing_2 (monostable) |
| I2 | Stop_2          | Q6 | PressErrorLight_2               |
| I3 | Reset_2         |    |                                 |
| I4 | ResetCounter_2  |    |                                 |
| I5 | CylinderUp_2    |    |                                 |
| I6 | ProductPlaced_2 |    |                                 |
| I7 | Error_2         |    |                                 |

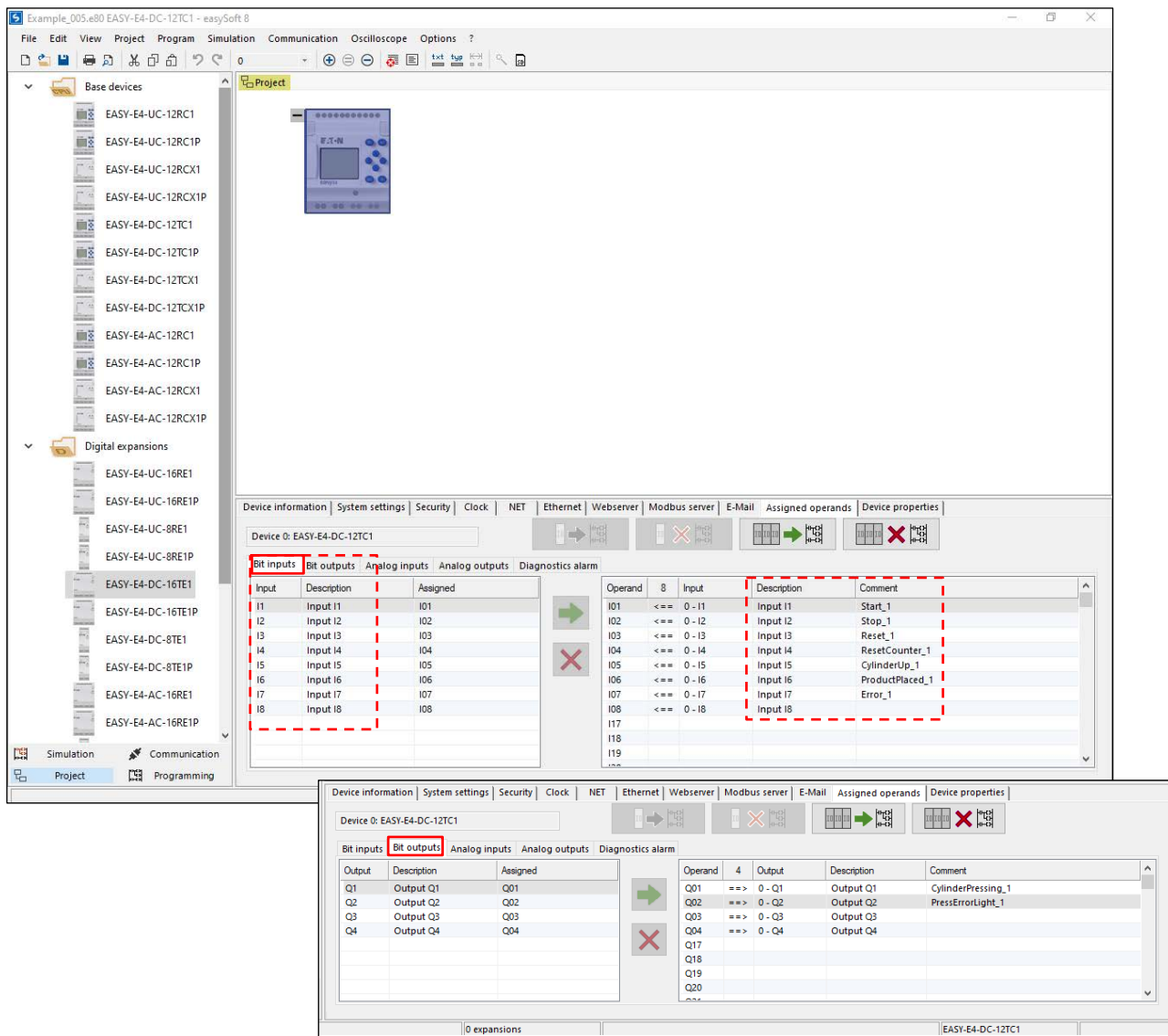


Fig. 85

- Click on the EASY-E4-DC-12TC1 and select "Assigned operands".

Here the names are shown that were assigned to the various I/O on the Easy-E4-DC-12TC1 during programming.

### Press 1

#### Inputs

I1 Start\_1  
I2 Stop\_1  
I3 Reset\_1  
I4 ResetCounter\_1  
I5 CylinderUp\_1  
I6 ProductPlaced\_1  
I7 Error\_1

#### Outputs

Q1 CylinderPressing\_1 (monostable)  
Q2 PressErrorLight\_1

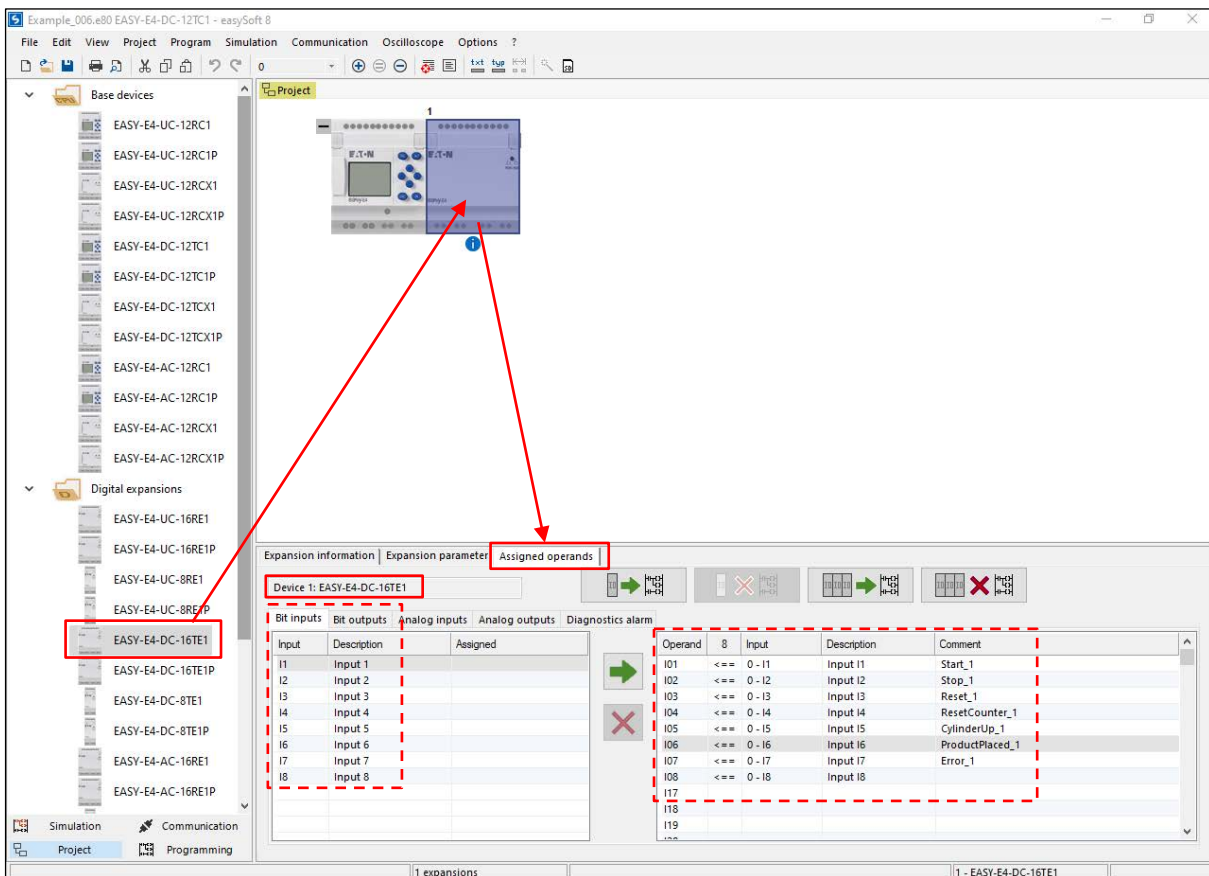


Fig. 86

Now the expansion must be added in the project itself.

- ▶ Save the program under a different name. For example "Example\_006".
- ▶ Go to the "Project" menu.
- ▶ In the "Toolbox", open the list of expansion modules by clicking on "Digital expansions".
- ▶ Drag an EASY-E4-DC-16TE1 onto the workspace next to the base unit.
- ▶ Select the expansion and in the properties field, go to "Assigned operands".

The inputs of the expansion are displayed on the left side at "Bit inputs". The inputs are named I0 ... I8. These are the names that are listed next to the terminals of the expansion device.

On the right side are the already assigned I/O of the base unit. The I/O of the expansion must also be assigned.

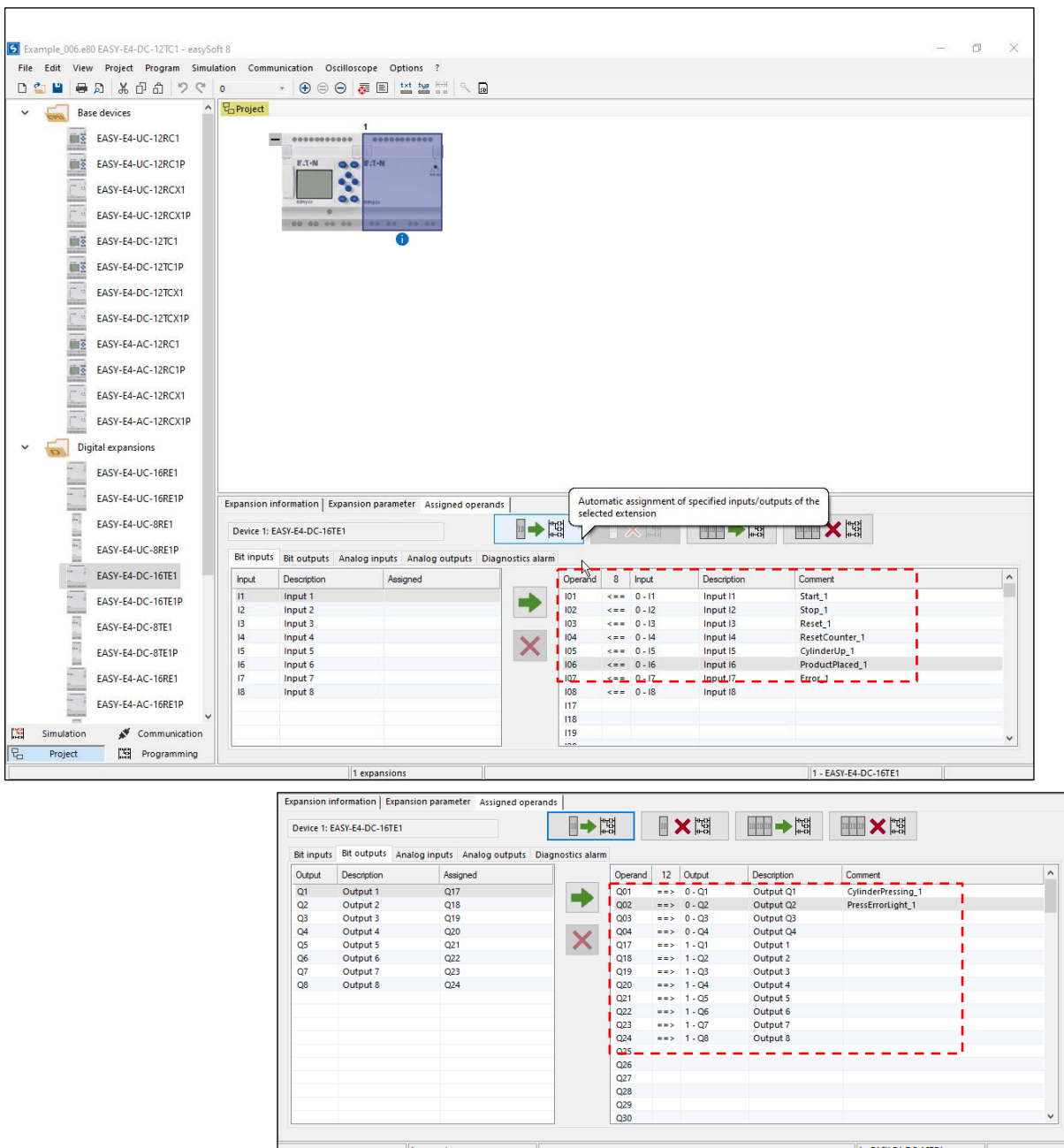


Fig. 87

The easiest way to assign the I/O of the expansion, is by clicking the "Automatic Assignment" button. All I/O of the selected expansion is assigned. Both the inputs and outputs.

The physical input I1 of the expansion is linked to I17 in the software, I2 to I18, and so forth. For the outputs, the physical Q1 is linked to Q17 in the software, and so forth.

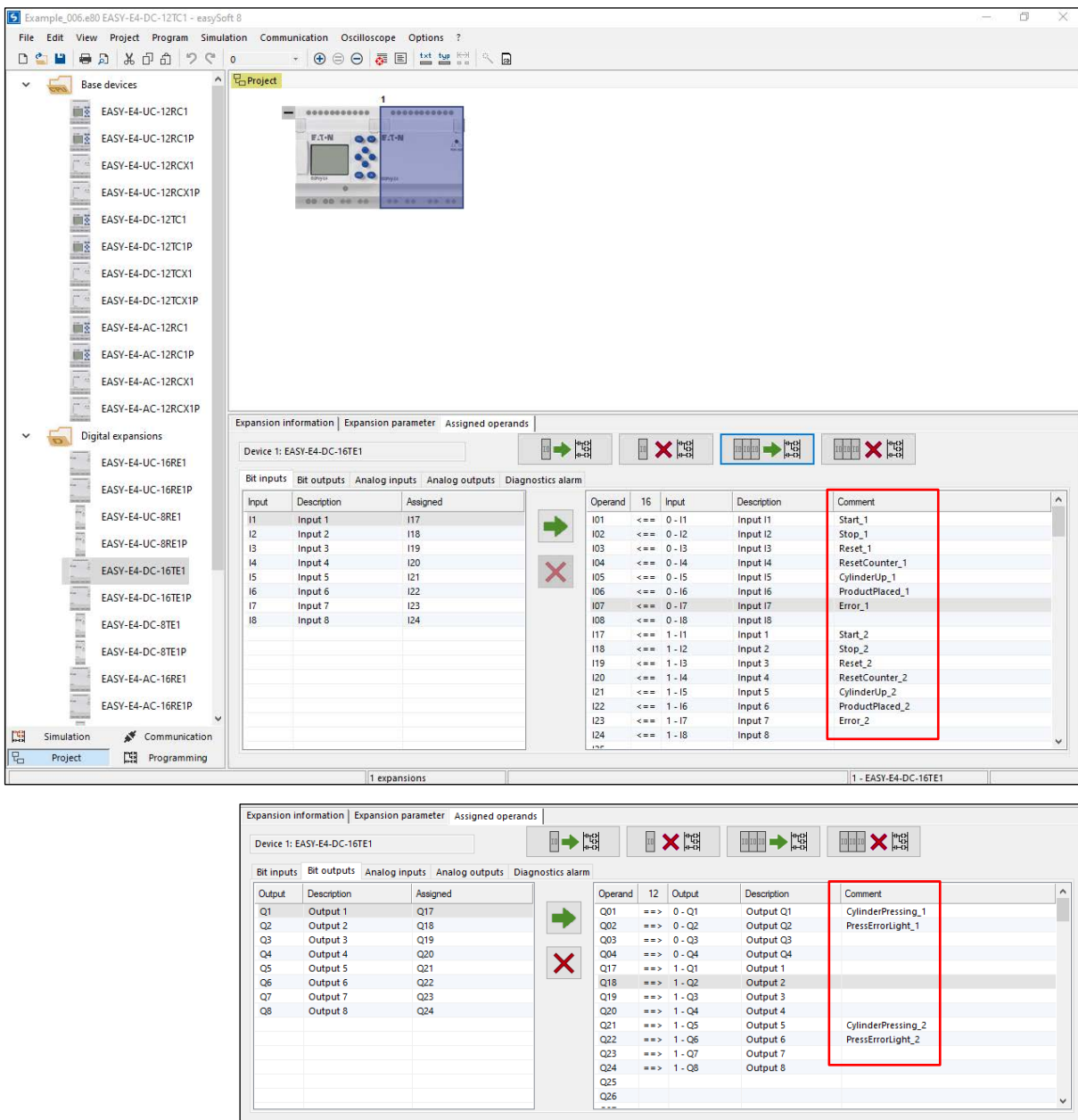


Fig. 88

Assign the desired designations to the I/O as shown in the list below. These designations can be added in the "Comment" column.

### Press 2

#### Inputs

I17 Start\_2  
I18 Stop\_2  
I19 Reset\_2  
I20 ResetCounter\_2  
I21 CylinderUp\_2  
I22 ProductPlaced\_2  
I23 Error\_2

#### Outputs

Q21 CylinderPressing\_2 (monostable)  
Q22 PressErrorLight\_2

## 2.14 Call function block again

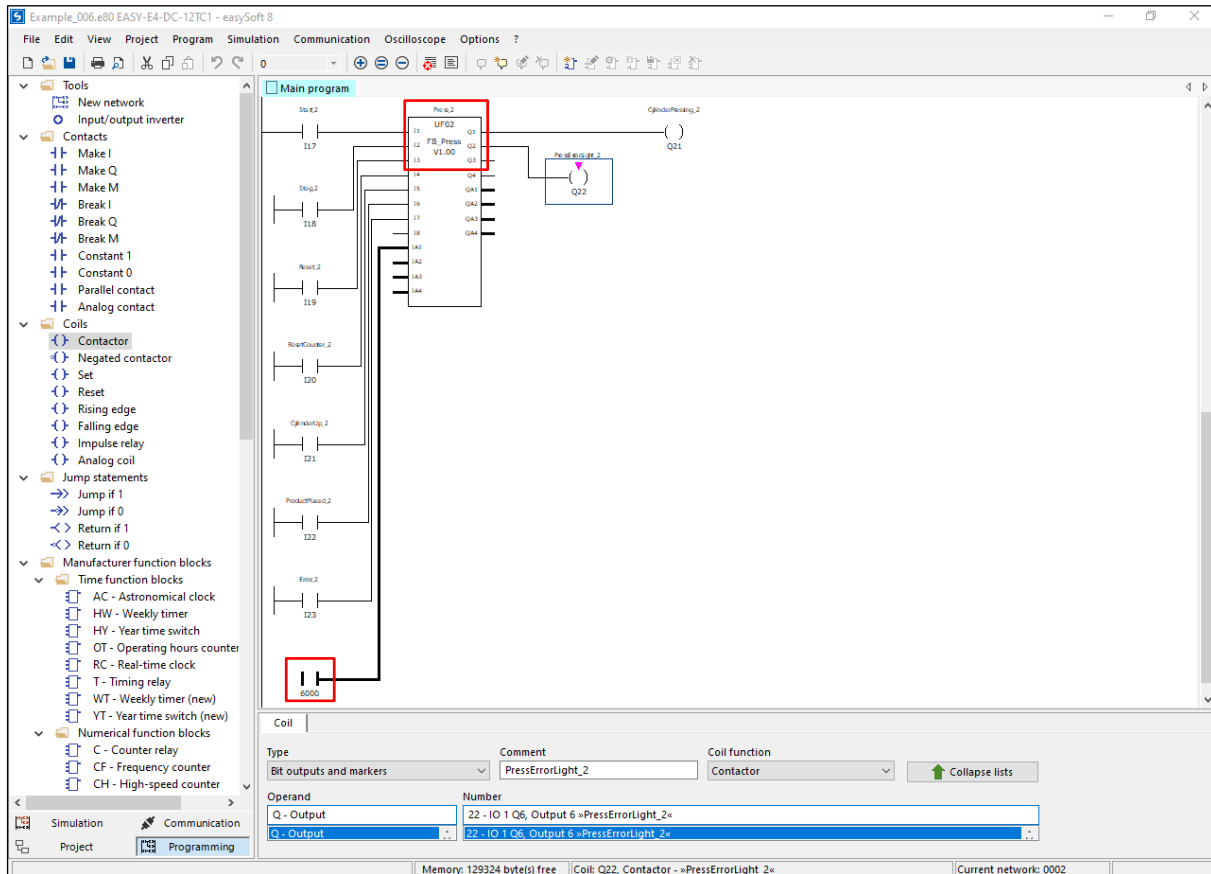


Fig. 89

- ▶ Go to "Programming" and add a new program rung in "Main program".
- ▶ Drag the "FB Press" FB to this rung. The FB now receives a sequence number UF02.
- ▶ Name the FB "Press\_2".
- ▶ Place the inputs and outputs belonging to press 2 on the FB. Make the time 6,000 ms. instead of 4,000 ms.

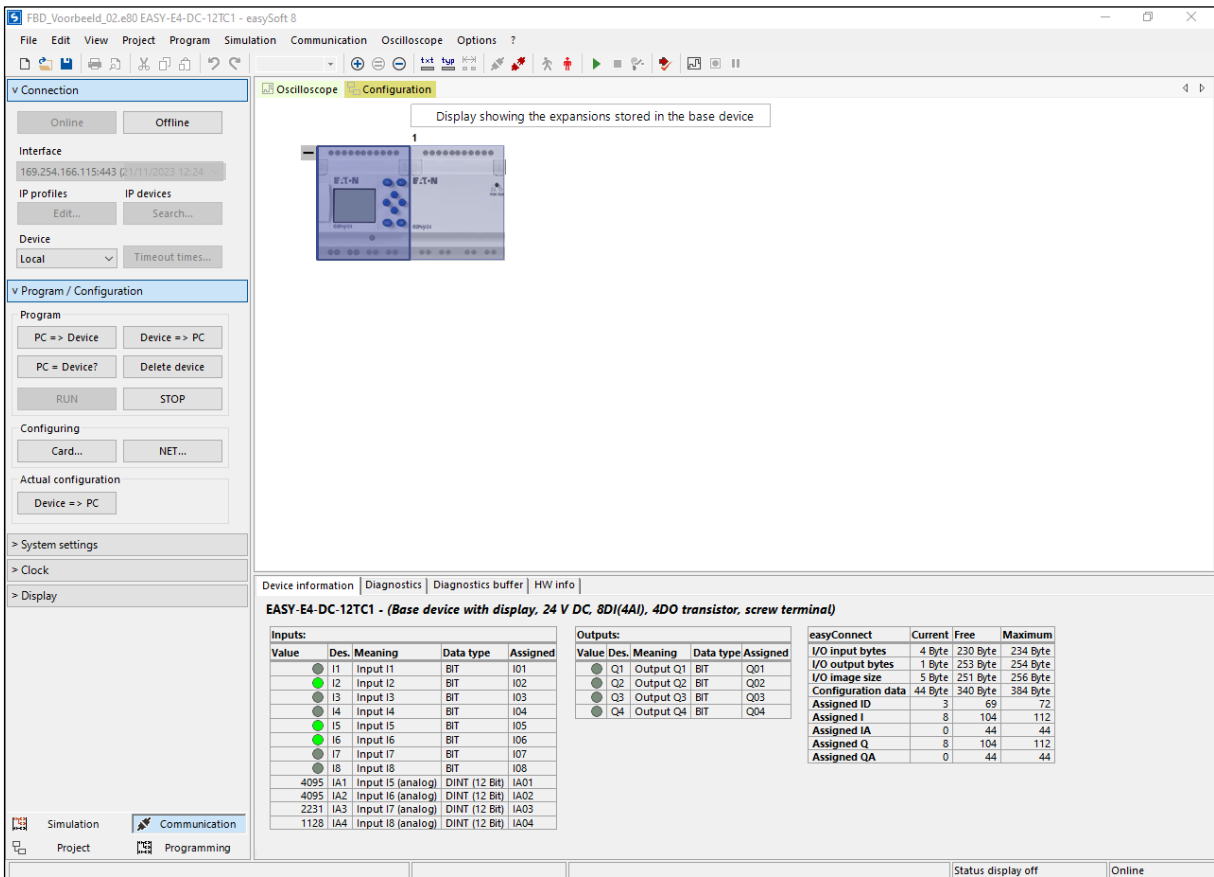


Fig. 90

► Go to the "Communication" menu and send the program to the easyE4.

As soon as the easyE4 is set to "RUN", the expansion is also shown.

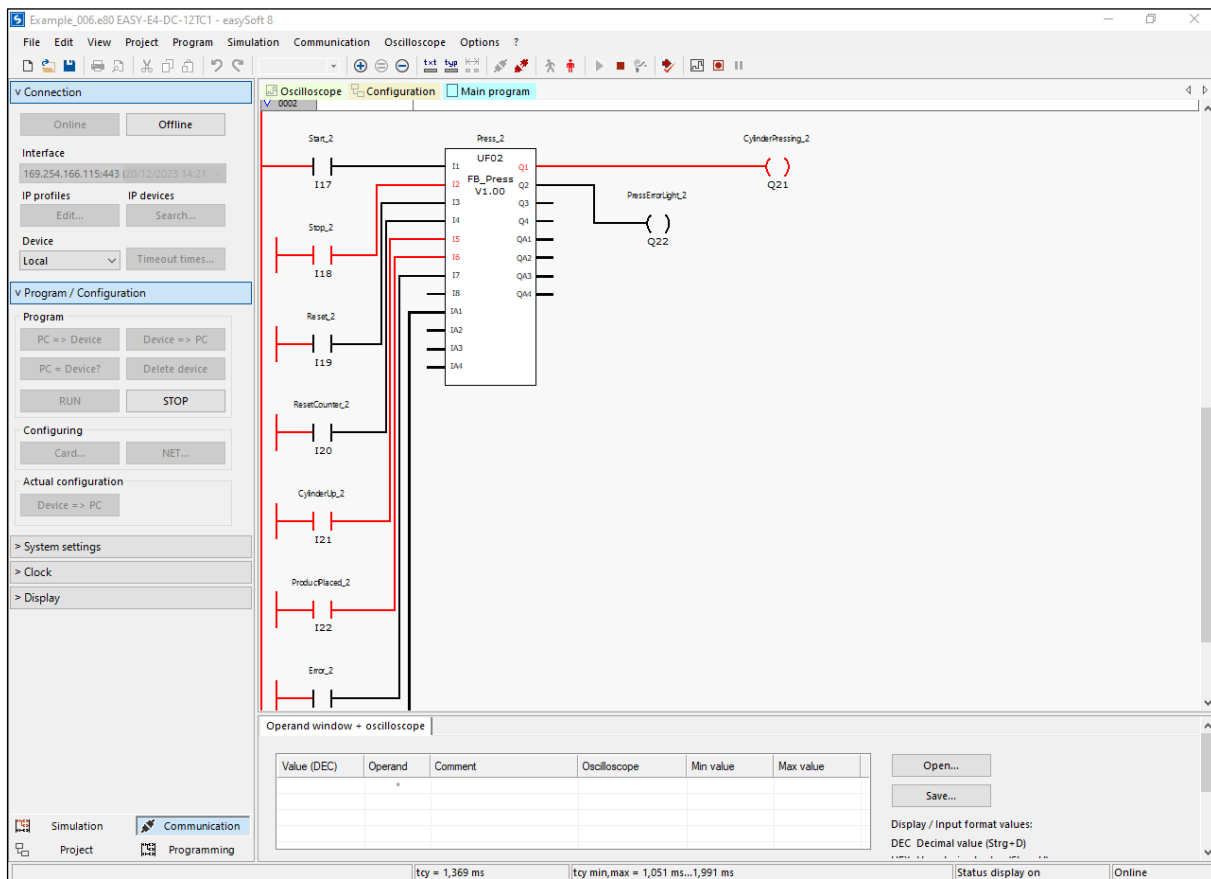


Fig. 91

View online and check the operation of the second press.

## 2.15 Visualization

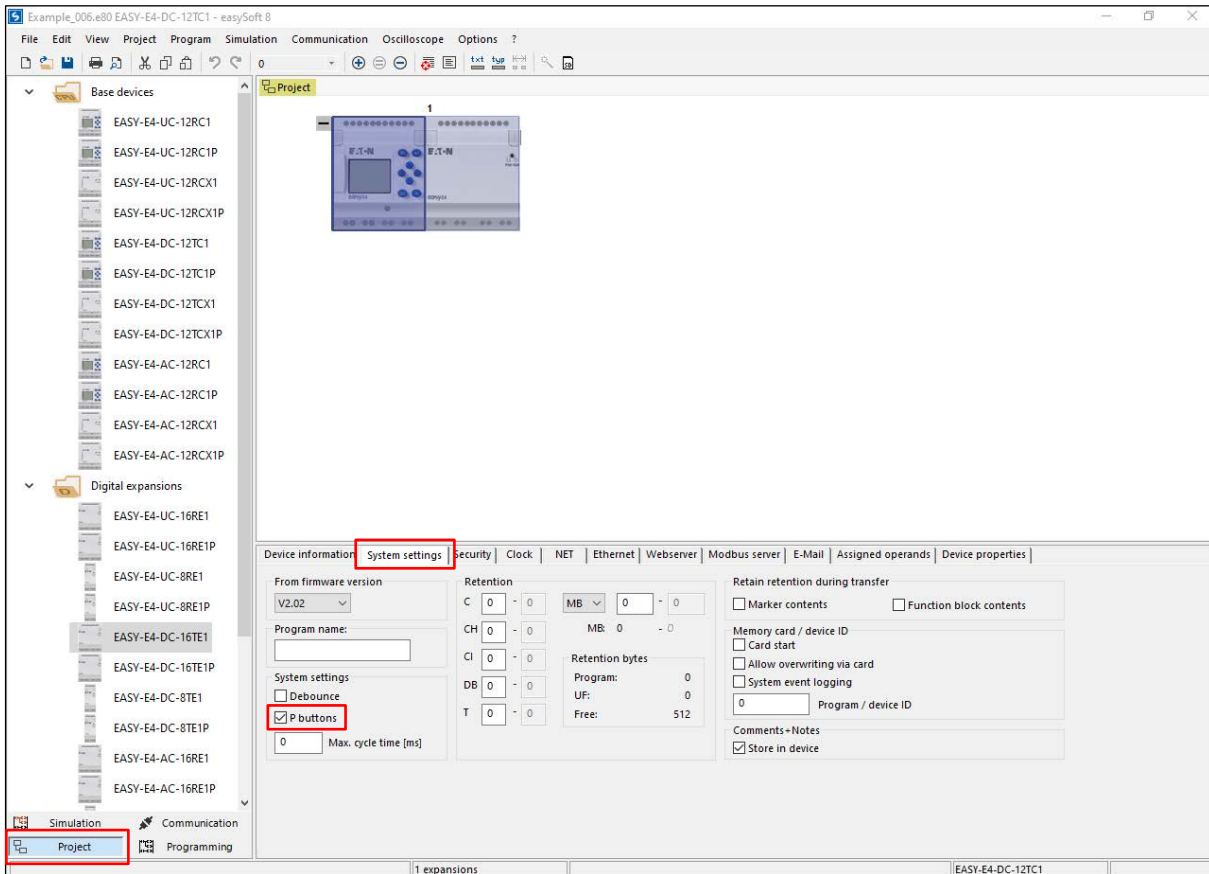


Fig. 92

Two screens must be shown on the display of the easyE4. The first screen relates to press 1 and the second to press 2. The screen must allow the pressing time to be set, and the number of pressings must be displayed.

The selection of the screens takes place using the cursor keys on the easyE4.

- ▶ Go to the "Project" menu, select the Easy-E4-DC-12TC1.
- ▶ Select the "System settings" tab and enable the cursor keys by checking the "P buttons".

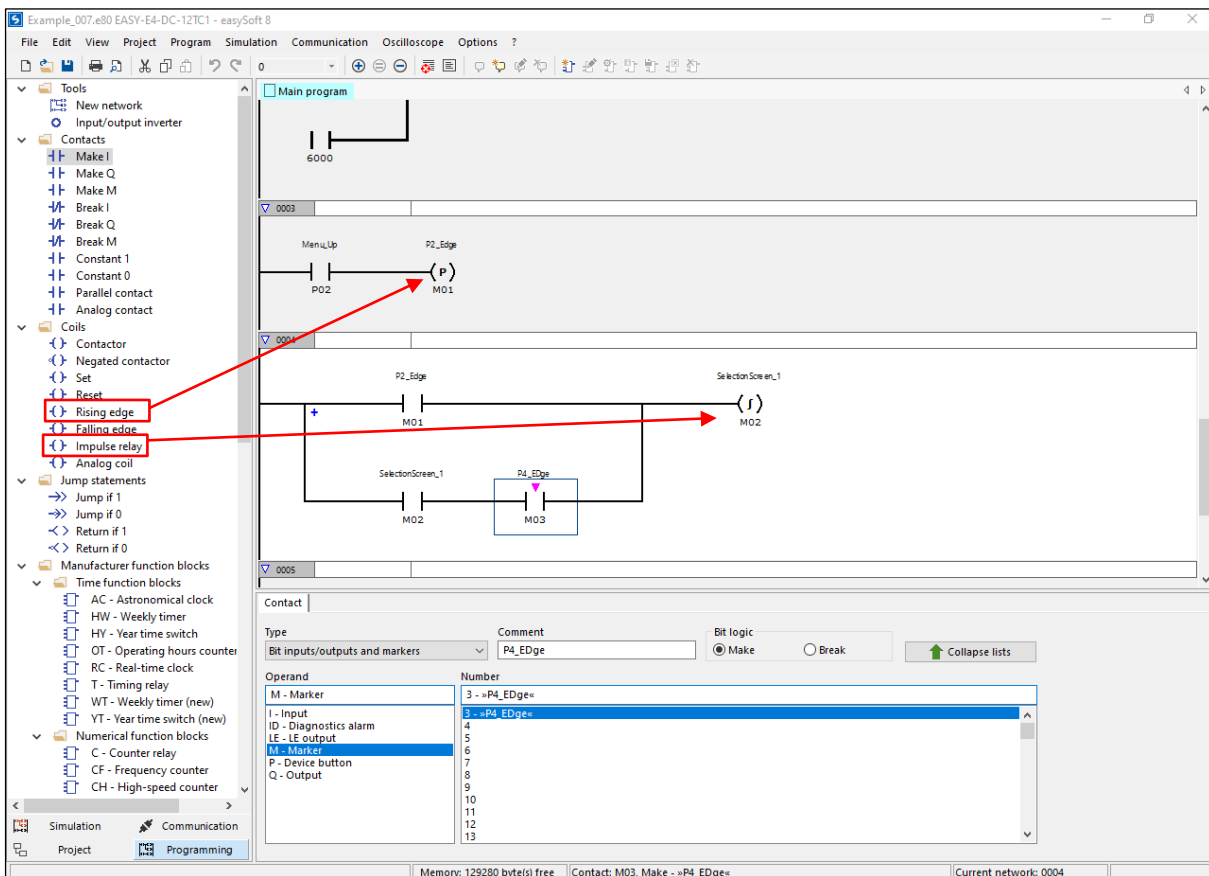


Fig. 93

For selecting the desired screens, markers are used.

- Add a new rung to the program.

The cursor key "P2" = "∧" = "Menu\_Up" is used to display the first screen.

P2 forwards Marker M1 with an up arrow and is named "P2\_Edge".

This rising edge is used on the next rung to control Marker M2. The M2 Marker is an impulse relay and is named "SelectionScreen\_1".

Marker M2 must also be triggered if M2 is high and there is an edge from cursor key P4 = "∨" = M3 with the name "P4\_Edge".

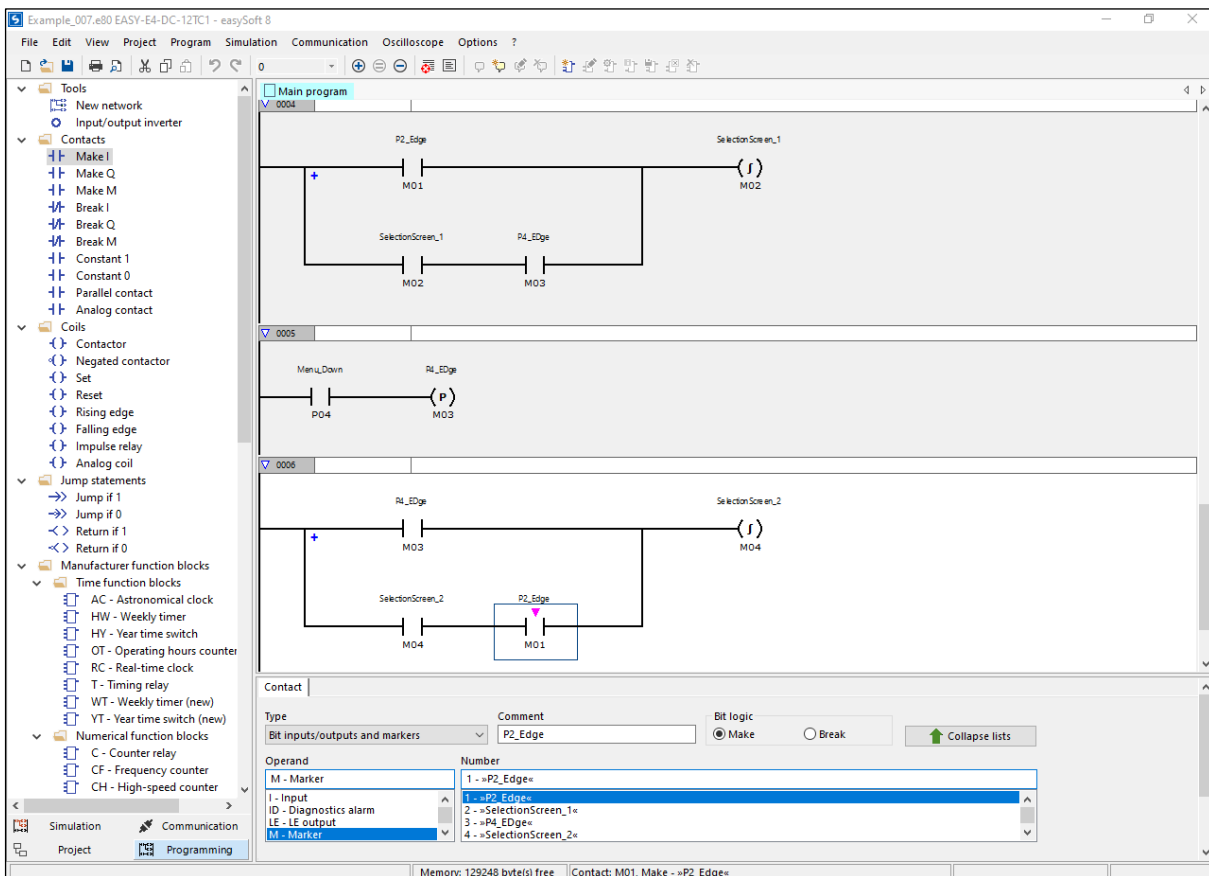


Fig. 94

The next step is to link the rising edge of P4 = "v" = "Menu\_Down" to Marker M3 = "P4\_Edge". Use this rising edge on the next rung to drive Marker M4. Marker M4 is an impulse relay and is given the name "SelectionScreen\_2".

This Marker M4 must also be triggered if M4 is high and there is an edge from cursor key P2 = "^" with the name "P2\_Edge".

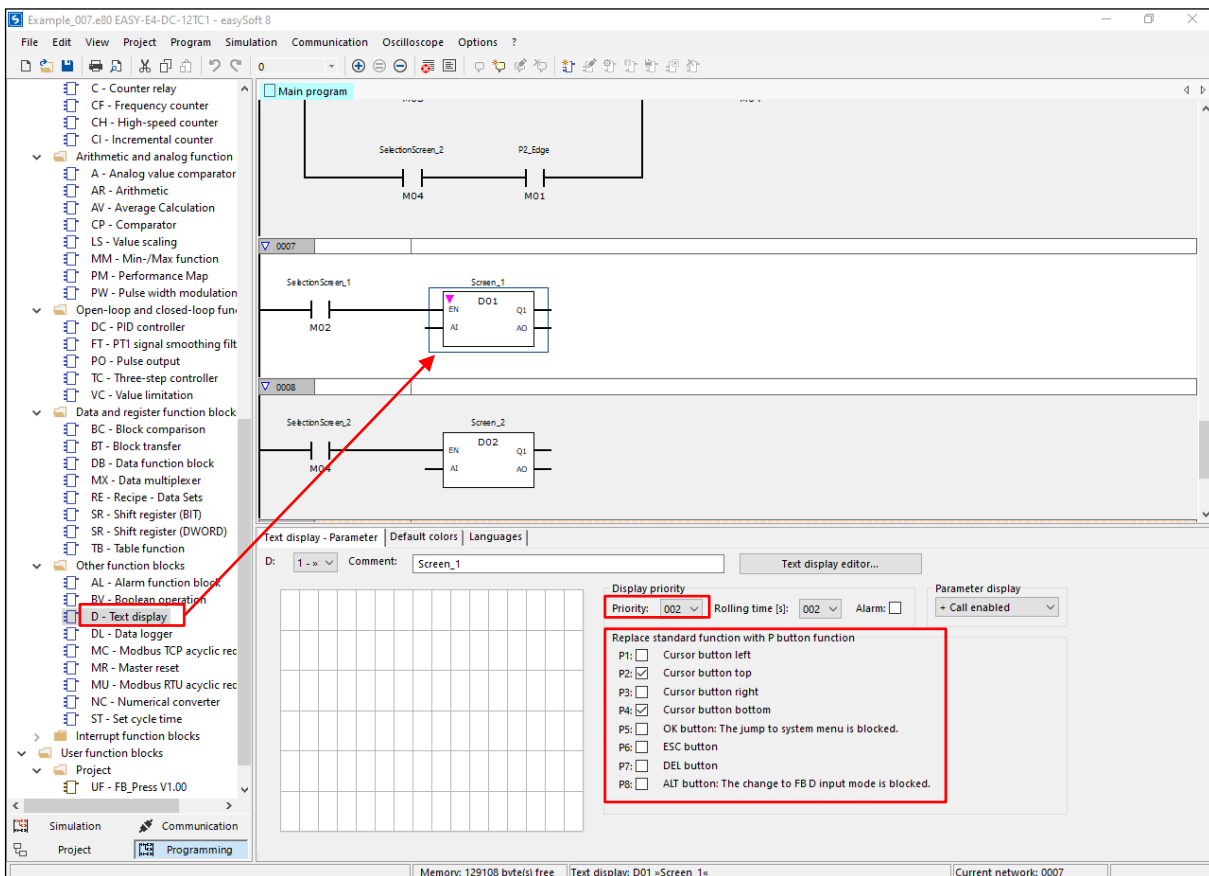


Fig. 95

- Link the marker M2 = "SelectionScreen\_1" to D01 = "D- Text display" and give it the name "Screen\_1".
- Link the marker M4 = "SelectionScreen\_2" to D02 = "D- Text display" and give it the name "Screen\_2".

The cursor keys had already been enabled for the program itself, but it must also be indicated which cursor keys should be active when a "D – Text display" is called up.

In this example the "Cursor button top" and "Curson button bottom" cursors must be activated.

- At "Display priority" assign the value 002 to the Priority.

This must be done for both screens.

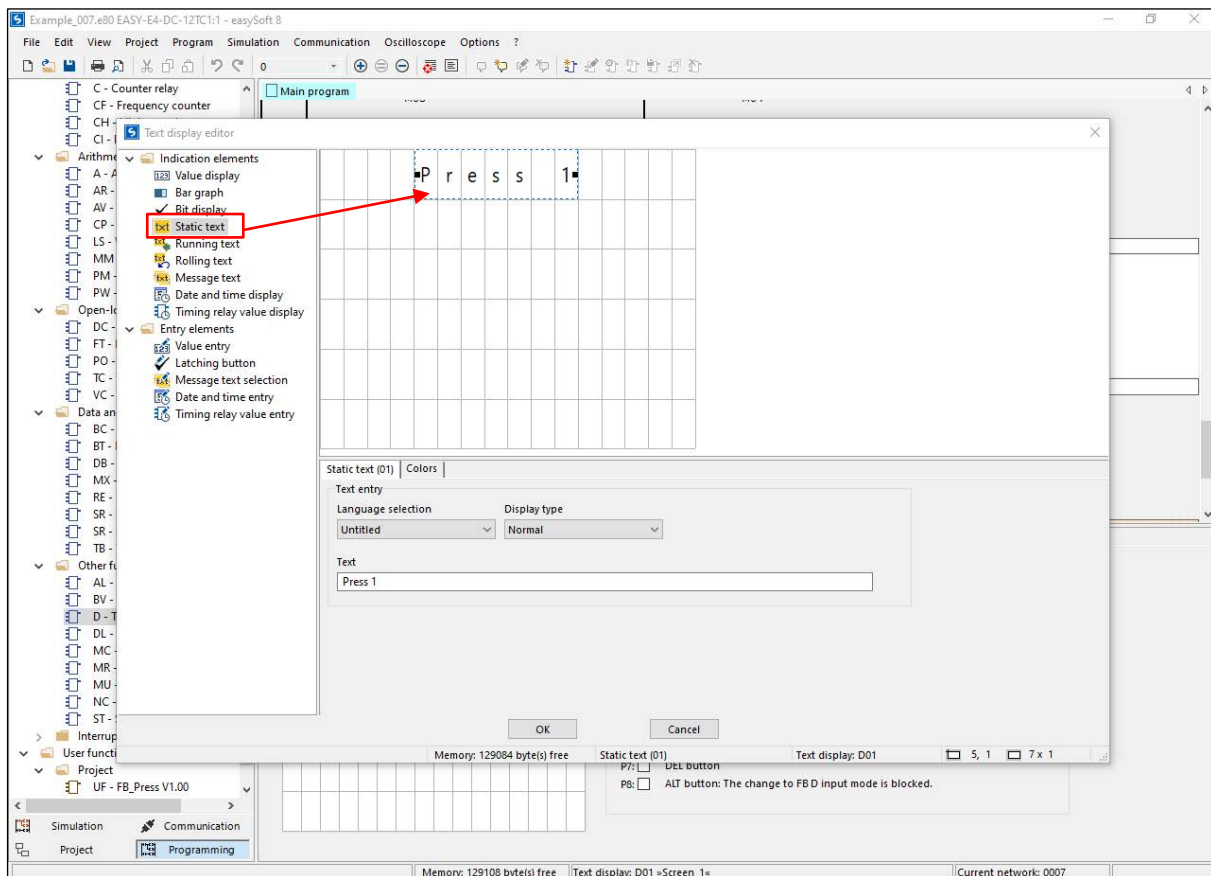


Fig. 96

Now, the visualization must be programmed on the screen.

- ▶ Select "Screen\_1" = D01 and click on "Text display editor ..."
- ▶ Drag a "Static text" element to the workspace and enter the text "Press 1".

If the text is not displayed completely, the box can be widened.

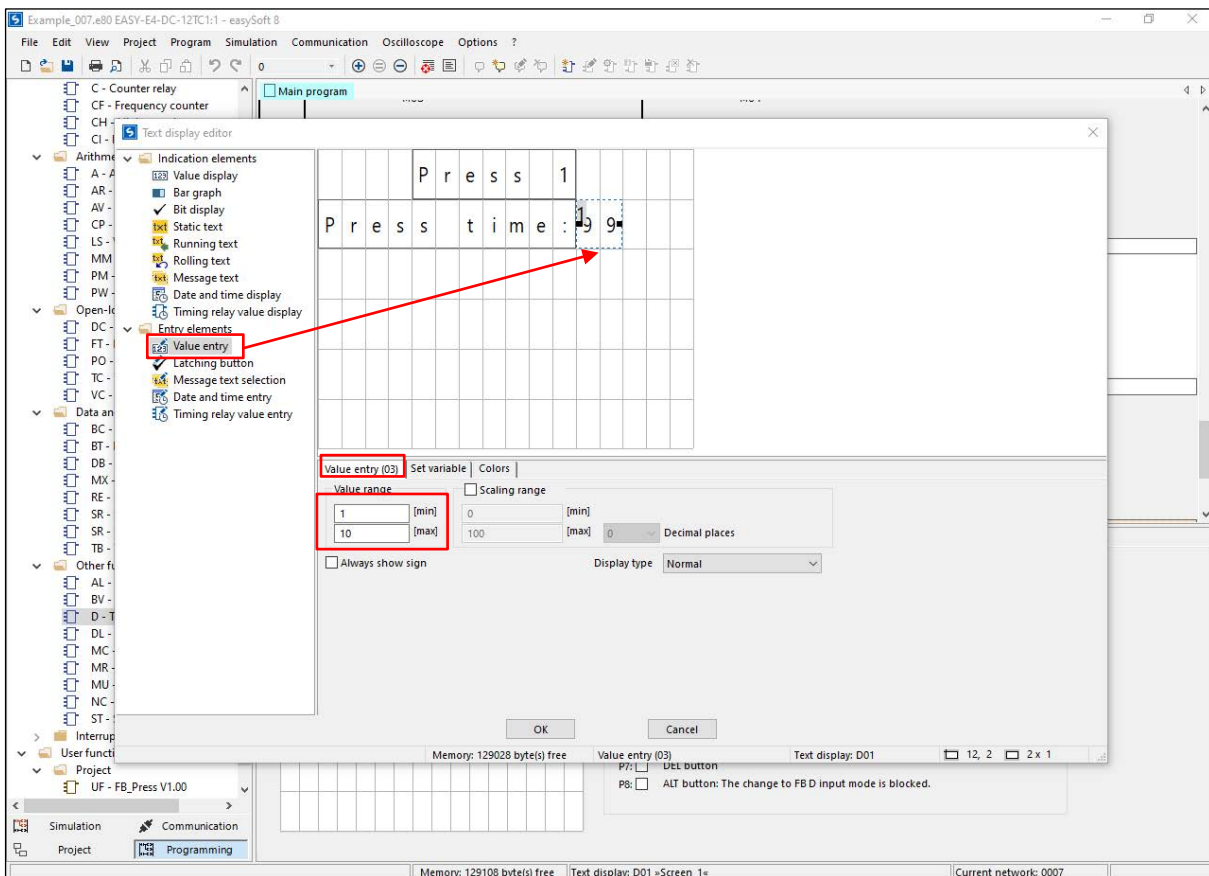


Fig. 97

- On the second rung, also insert a "Static text" field containing the text "Press time:".

There must be a "Value entry" field behind this text.

- Drag this to the workspace.

Under "Value entry", the "Value range" specifies what the minimum and maximum numerical values may be. In this example between 1 and 10.

- Adjust the format of the field to 2 digits.

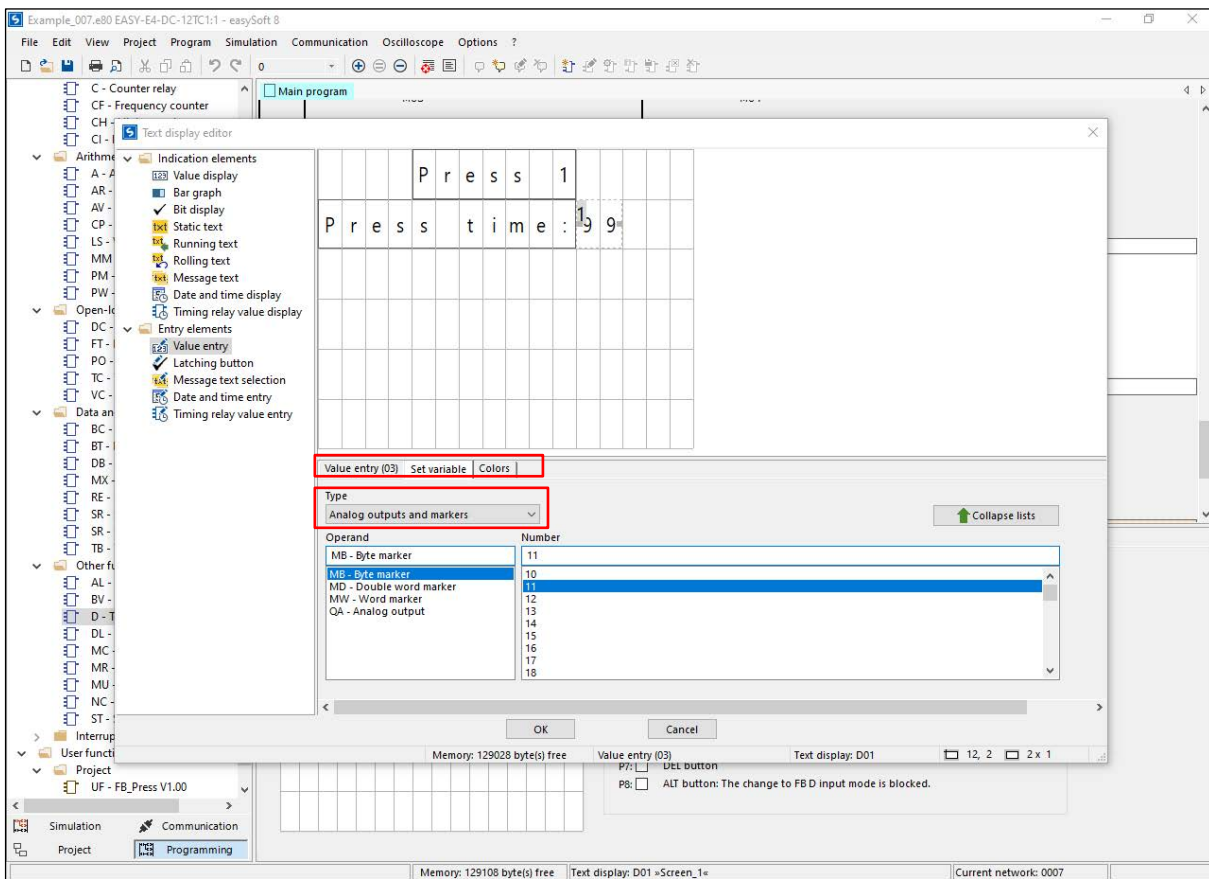


Fig. 98

- ▶ Go to the "Set variable" tab.
- ▶ Select "Analog outputs and markers" as type.
- ▶ Choose MB11 as input variable for the desired seconds of press 1.

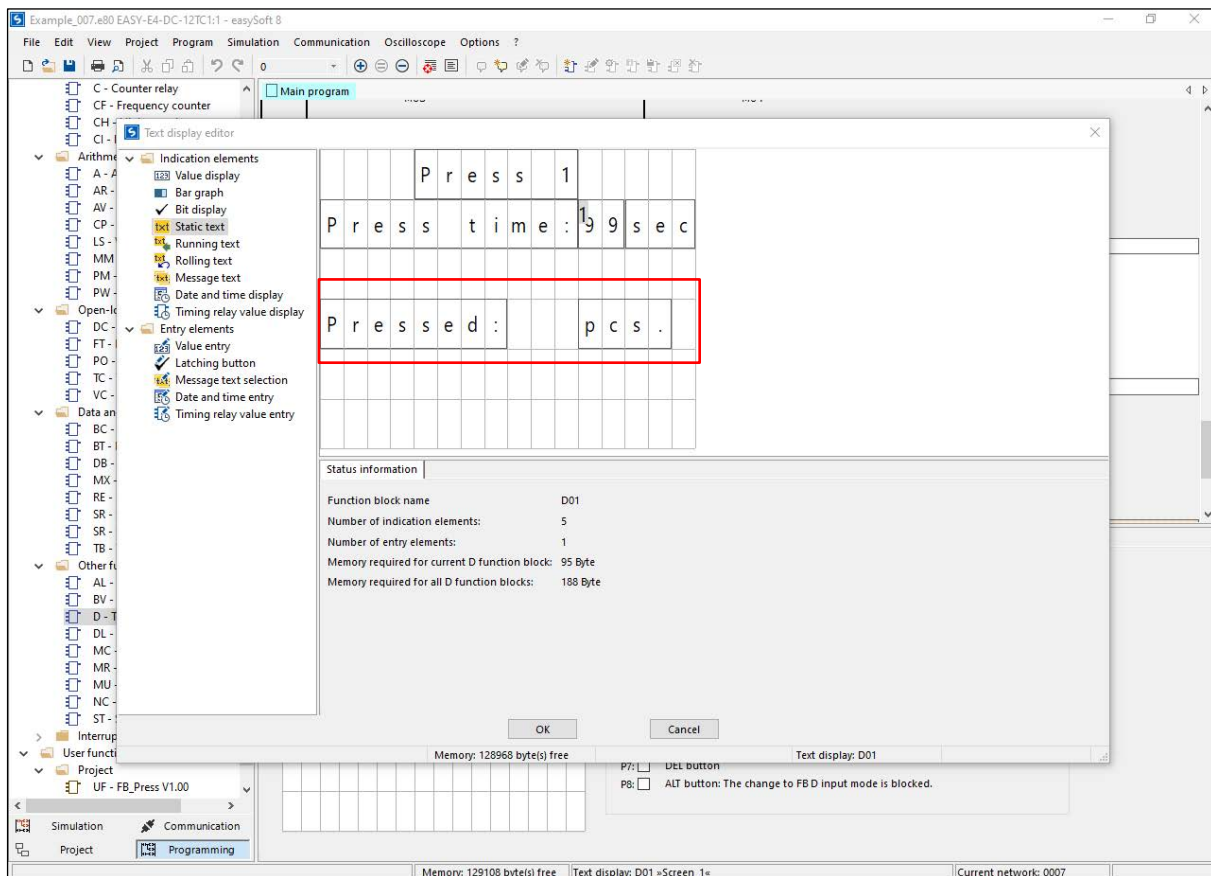


Fig. 99

- ▶ After the input field, insert a "Static text" field with the text "sec".
- ▶ Place two rungs below the texts "Pressed:" and "pcs".

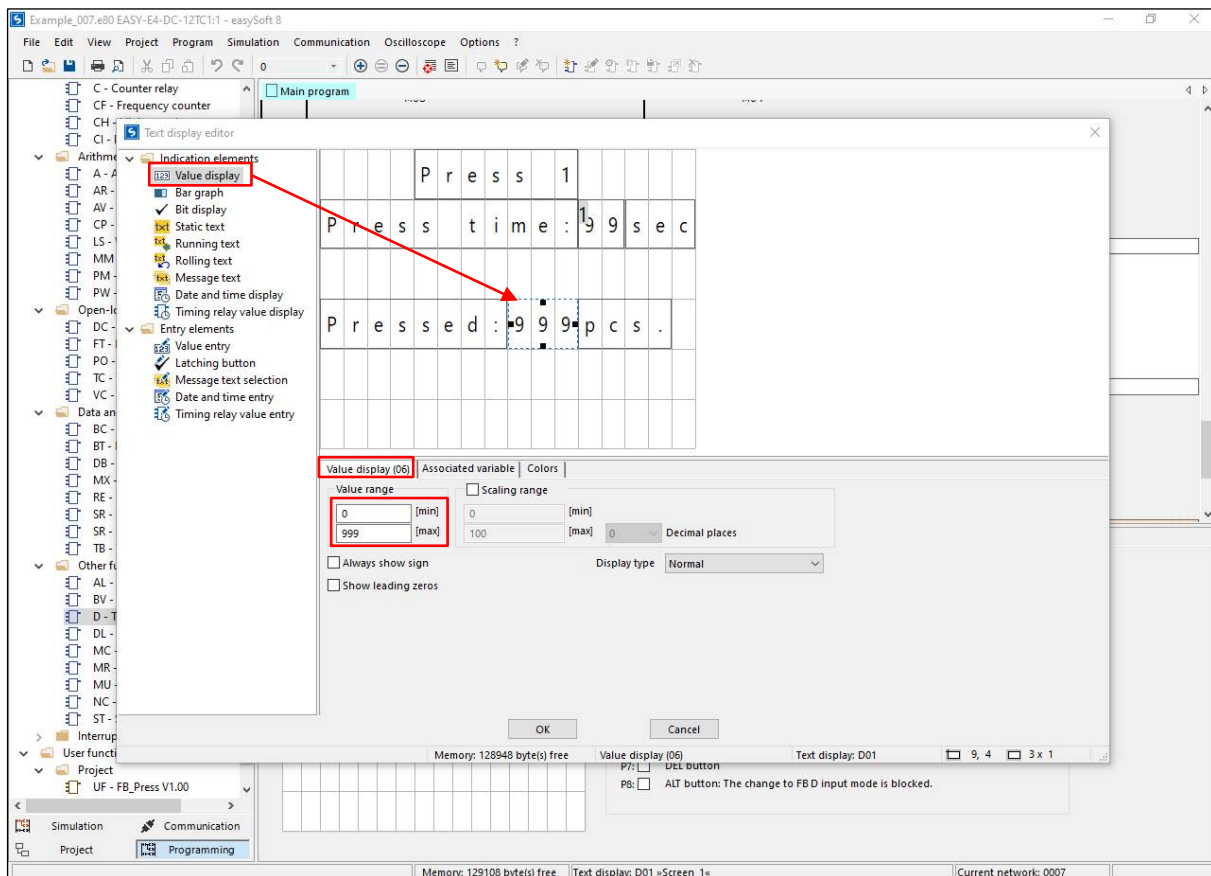


Fig. 100

The number of pressings must be displayed.

- ▶ Insert a "Value display" field on the workspace after "Pressed:".
- ▶ Go to the "Value display" tab.
- ▶ Enter the range of 0 - 999 at "Value range".

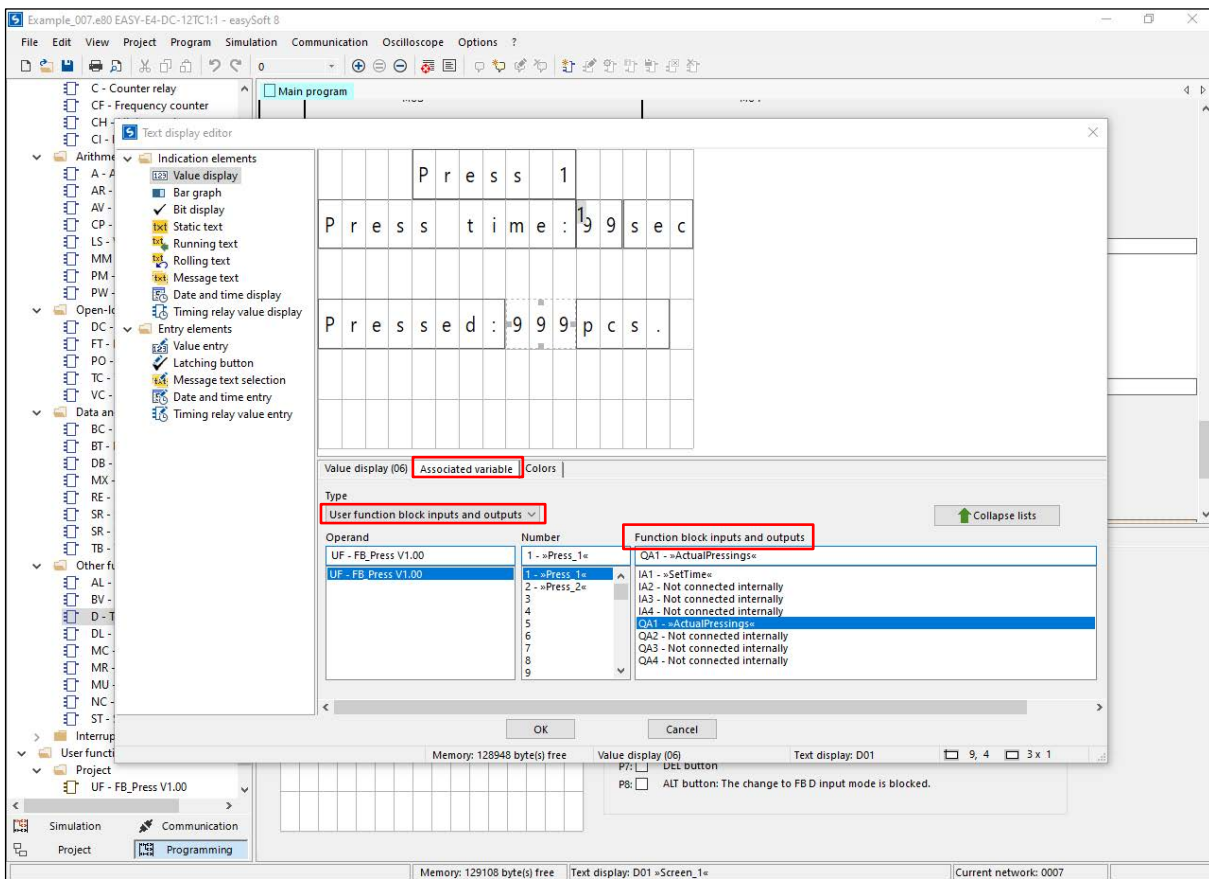


Fig. 101

- ▶ Go to the "Associated variable" tab. At "Type" → "User function block inputs and outputs" select the FB with the name "UF - FB\_Press V1.00".
- ▶ Under "number" select number 1 for the User function block with the name "Press\_1".

From this UF, the analog output "ActualPressings" (QA1) must be selected.

- ▶ Confirm all this with "OK".

The screen related to press 1 is now ready.

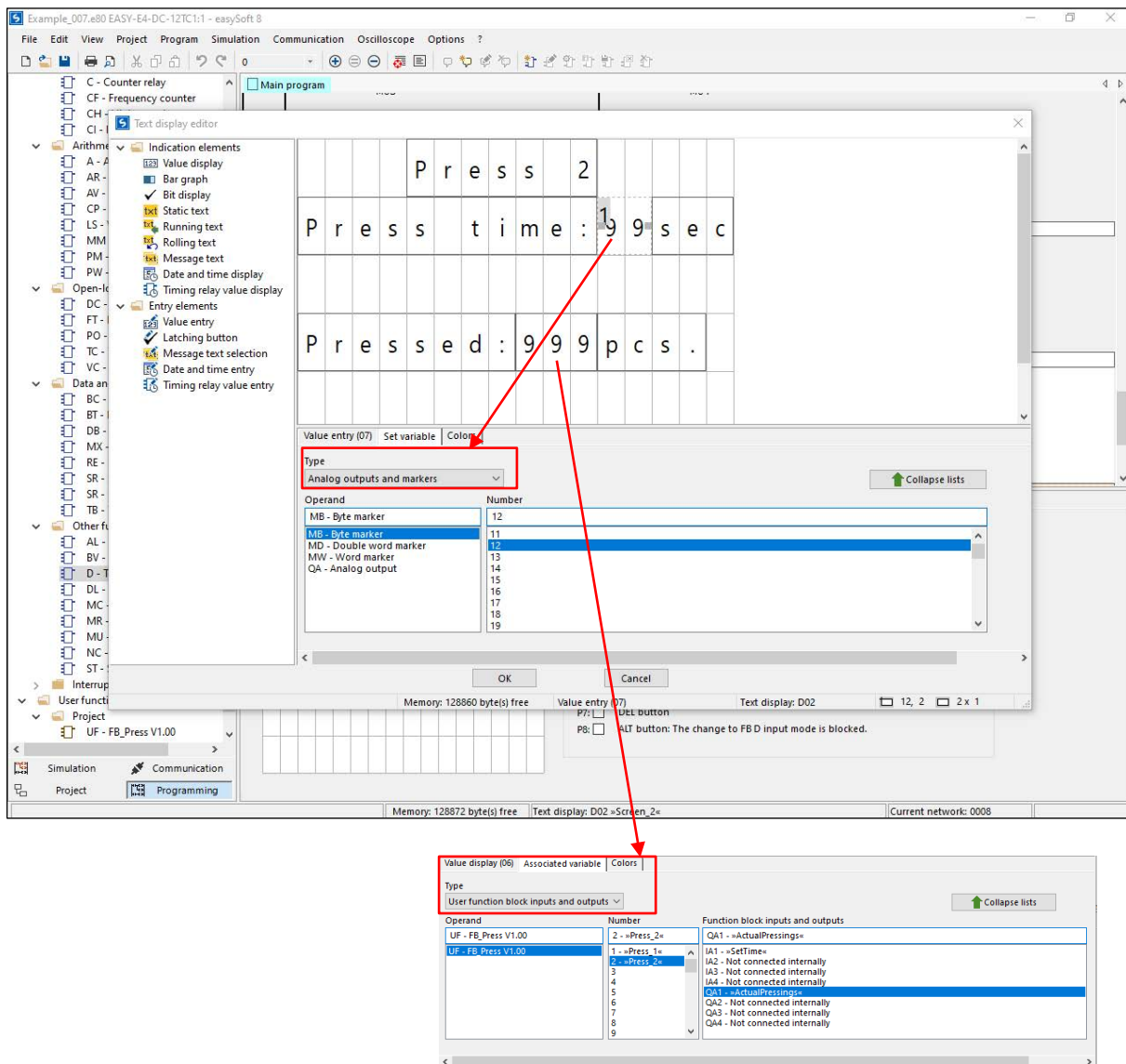


Fig. 102

Repeat this for the second screen. Here the input of the seconds are linked to MB12.

The number of counted pressings of press 2 is linked to the second call of the UF with the name "Press\_2".

- Confirm all this with "OK".

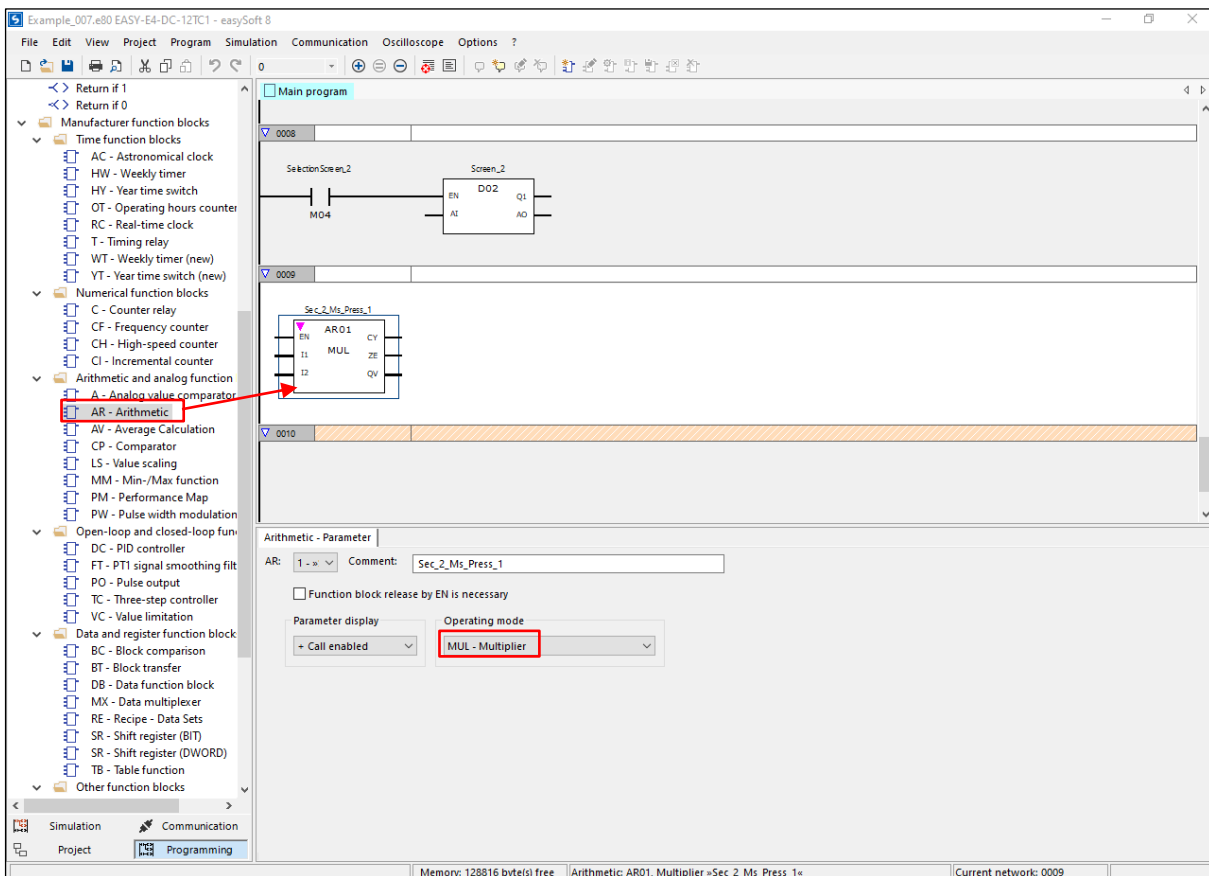


Fig. 103

The value entered on the screen is in seconds. However, the FB "FB-Press" expects milli seconds. This means that the entered value must be multiplied by 1000.

- Add an empty rung.
- Drag an arithmetic function block onto the workspace and give it the name "Sec\_2\_Ms\_Press\_1".

The "Operating mode" is set on "MUL - multiplier".

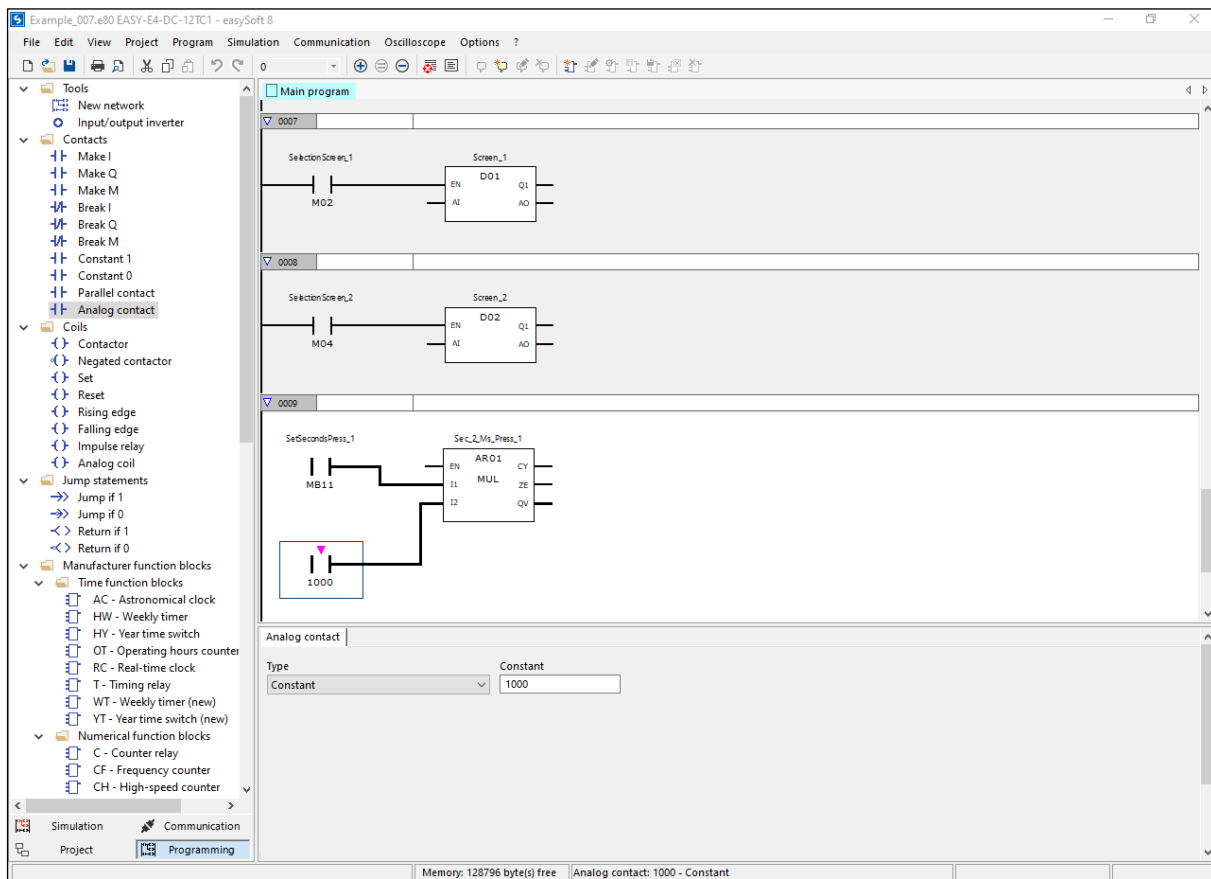


Fig. 104

The variable MB11 with the name "SetSecondsPress\_1" will be connected to I1 of the arithmetic function block. This MB11 is linked to the Press time input field on screen 1.

The fixed value 1000 must be entered at I2. This function block now multiplies the number of entered seconds by 1000.

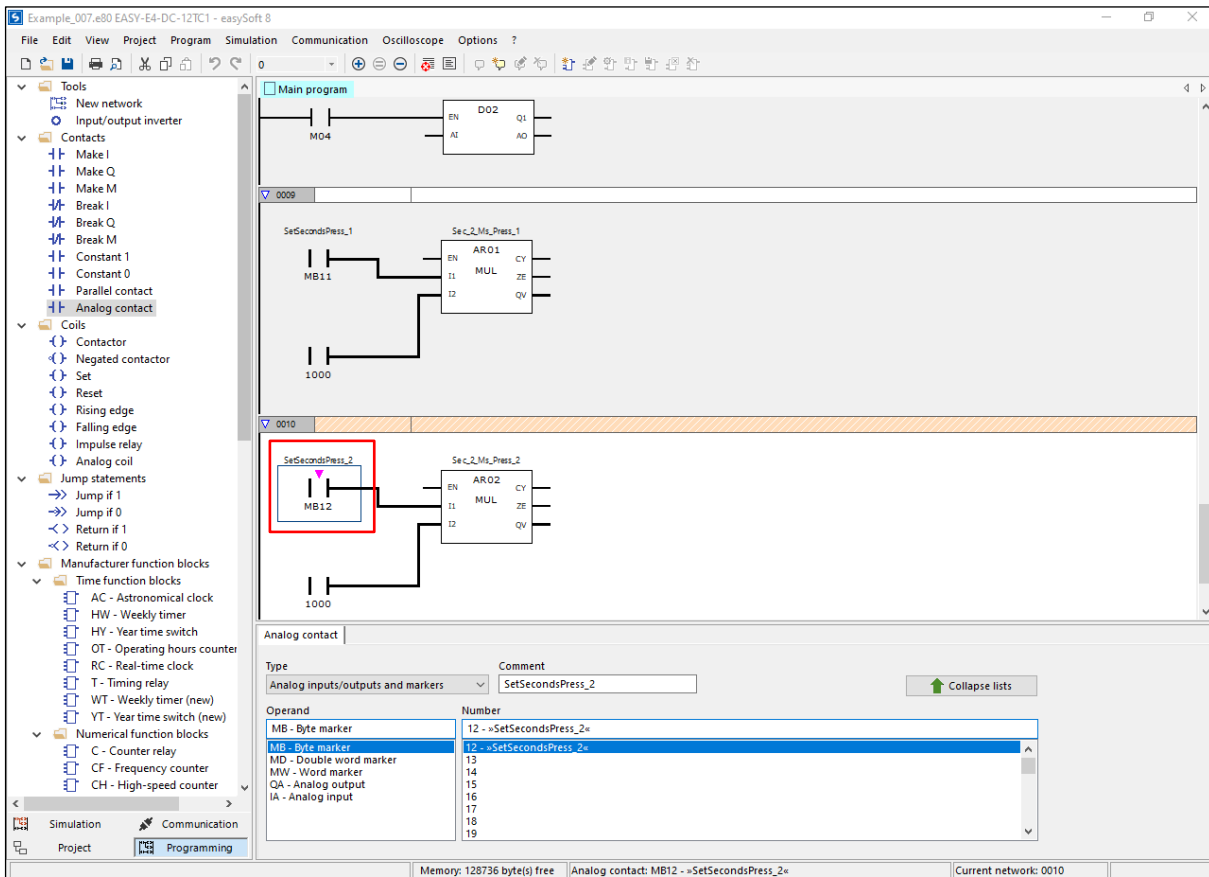


Fig. 105

- Add another empty rung and drag a second arithmetic function block onto the workspace.
- Name this "Sec\_2\_Ms\_Press\_2".

This must multiply the value of MB12 with the name "SetSecondsPress\_2" by 1000.

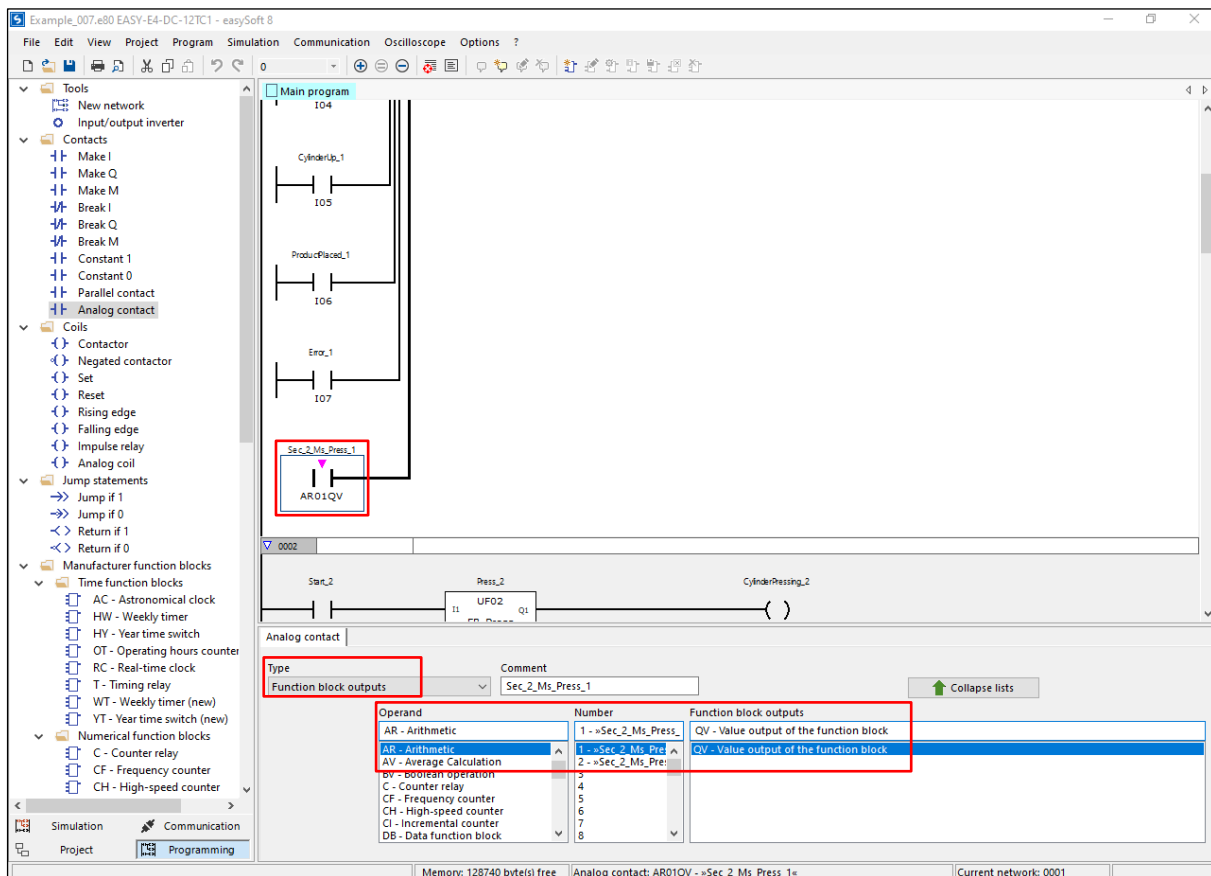


Fig. 106

The fixed value of 4000, which is on IA1 of "Press\_1", must be replaced by the result of arithmetic function block 1 with the name "Sec\_2\_Ms\_Press\_1".

- At "Type" select → "Function block outputs". At "Operand" → "AR- Arithmetic".

The first arithmetic function block must be selected under "Number".

- Select the output QV of "Sec\_2\_Ms\_Press\_1" here.

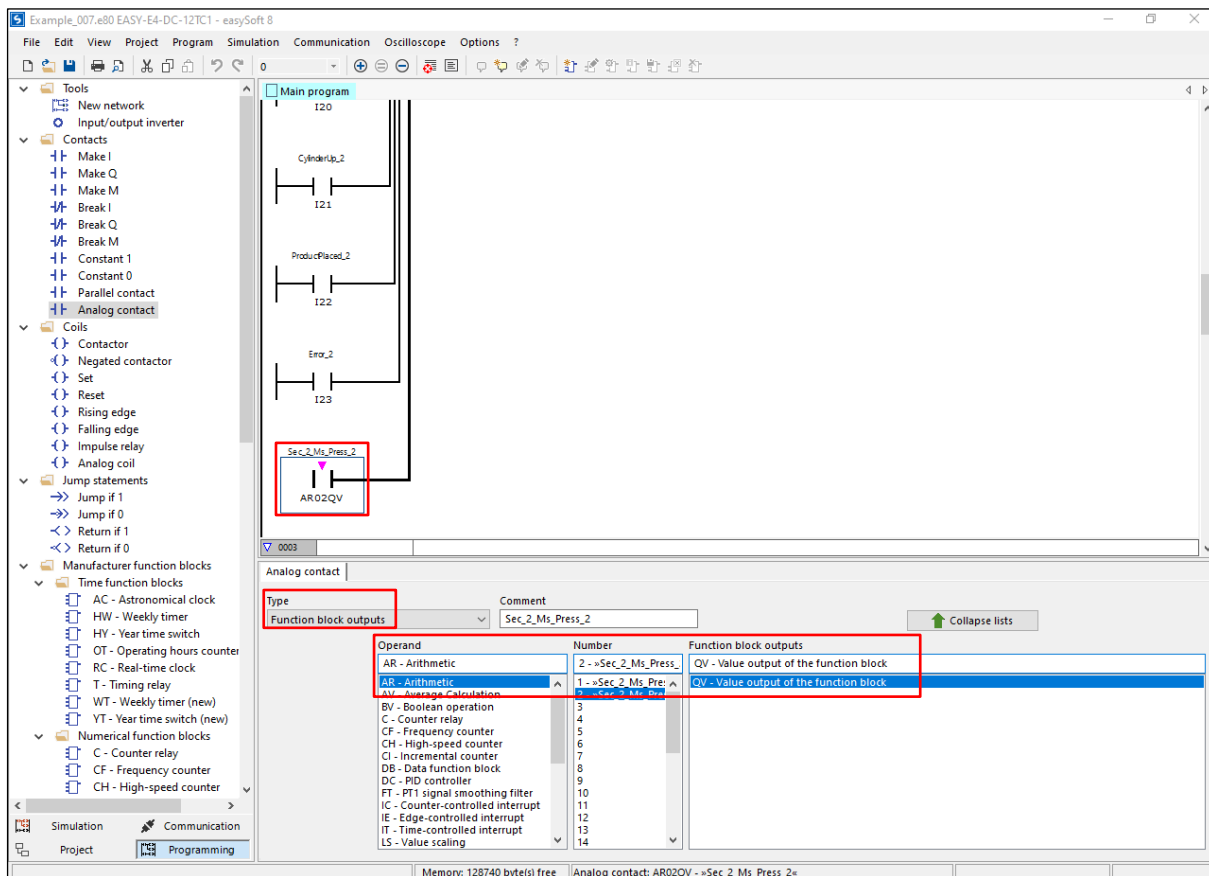


Fig. 107

Repeat this also for the second press for the UF "FB-Press".

The multiplied value of "SetSecondsPress\_2" (MB12) from the arithmetic function block "Sec\_2\_Ms\_Press\_2" is applied here.

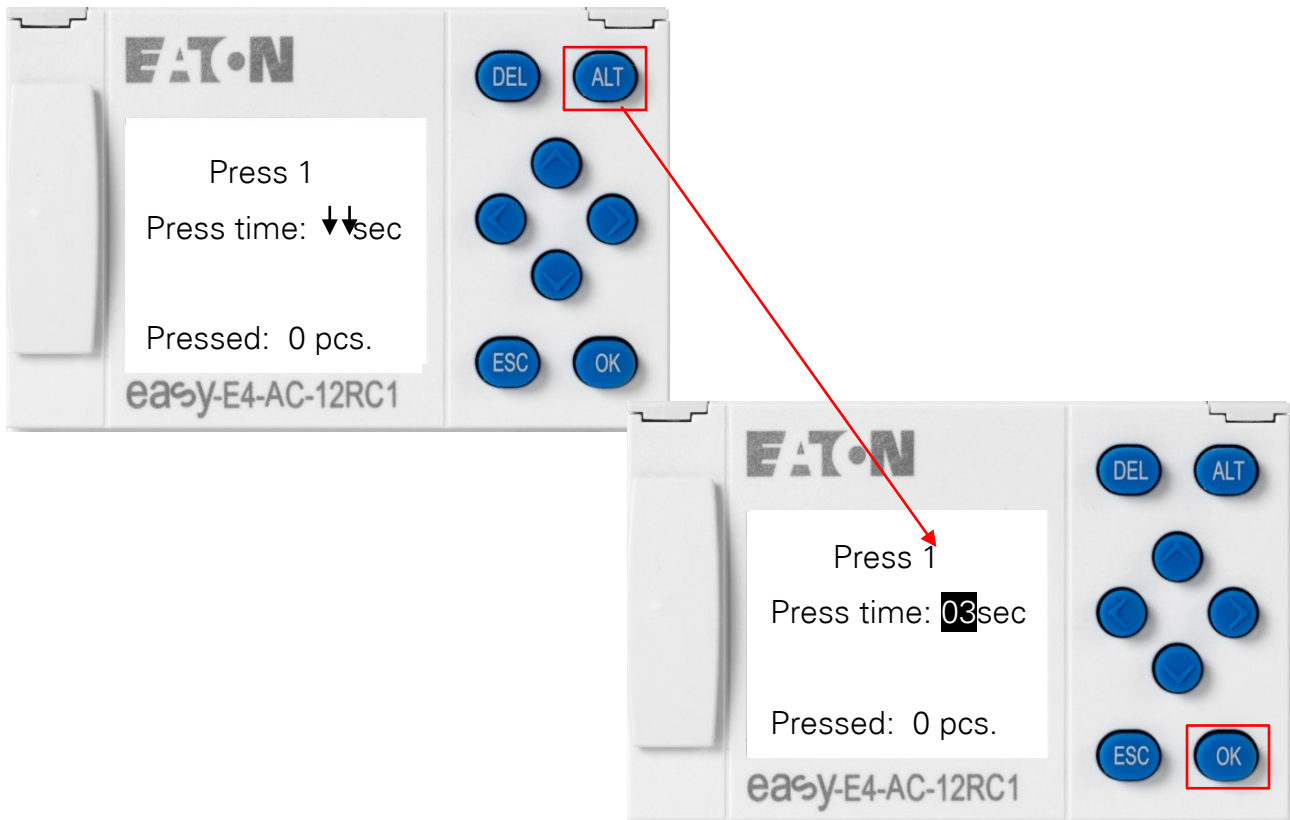


Fig. 108

Send the program to the easy and press on the “^” key.

The screen displays ↓↓ at the press time. This is because the value is at 0. Press the “Alt” key to enter a value. Use the “<” and “>” keys to select the digit that needs to be adjusted. The value can be increased or decreased using the “^” and “v” keys. The value is confirmed with the “OK” key.

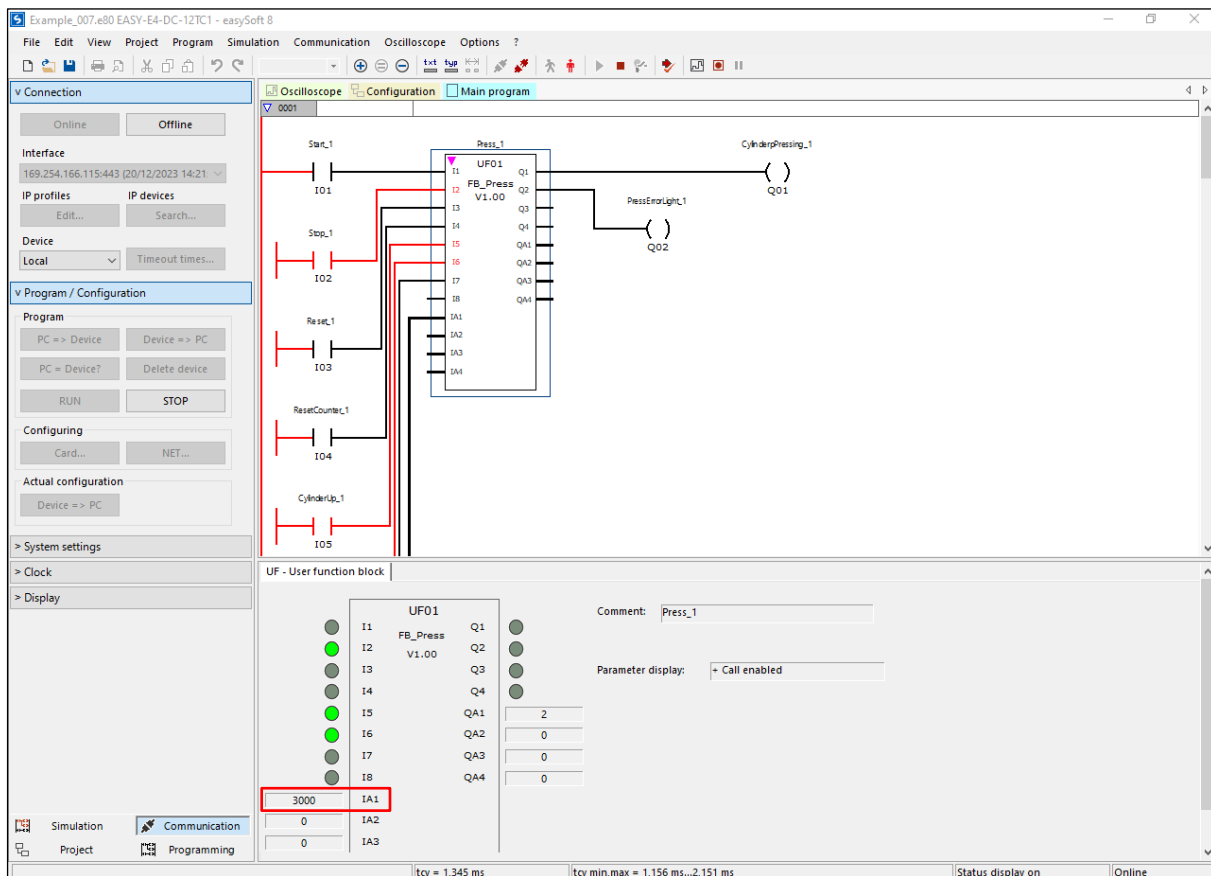


Fig. 109

The entered 3 seconds is multiplied by 1000 and placed on the input IA1 of the UF01 “FB-Press”.

## 2.16 Alarm screen

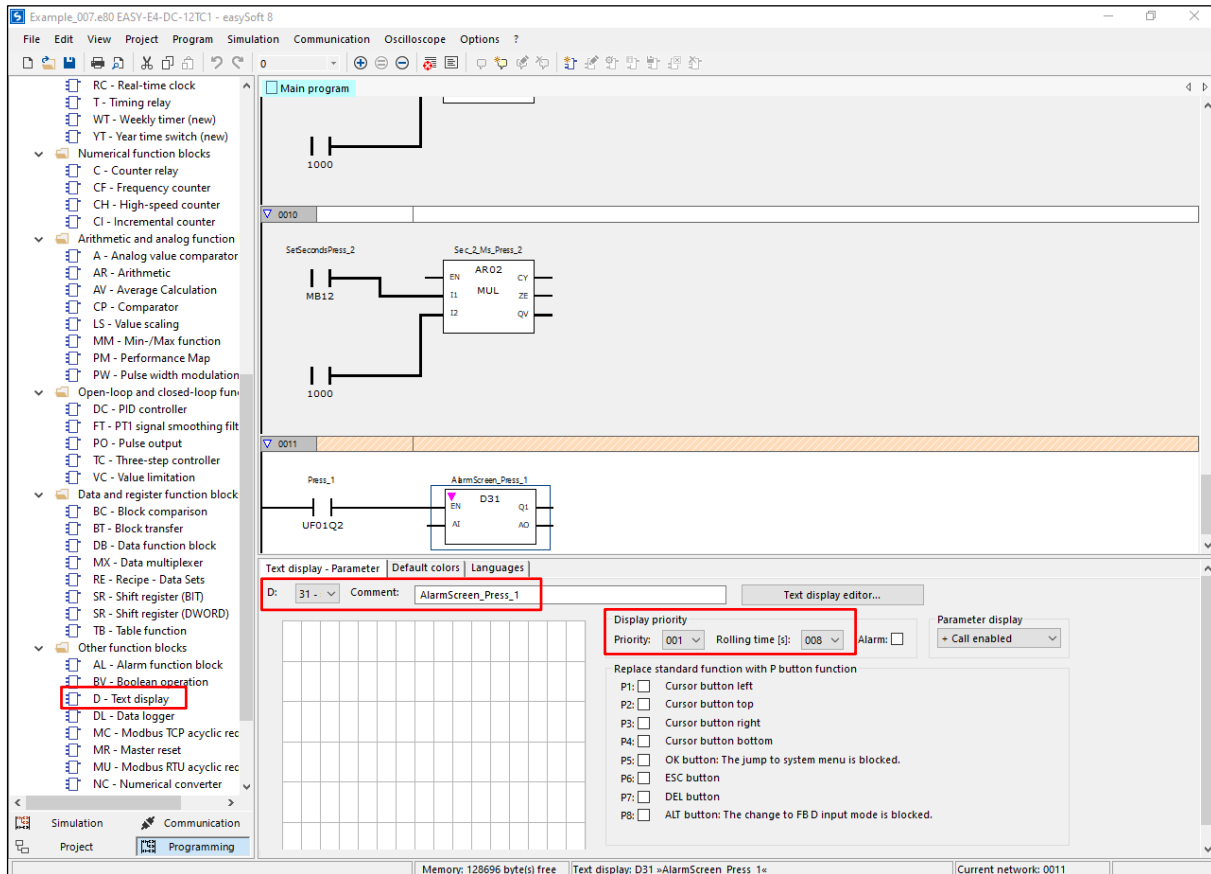


Fig. 110

The screens can also be used to display alarm notifications.

- ▶ Create a new rung and drag a "D-Text display" to the workspace.
- ▶ Give this number 31 and the name "AlarmScreen\_Press\_1".
- ▶ Select under "Display priority" → "Priority" 001 and for "Rolling time [s]" the value 8.
- ▶ Use the fault message "PressErrorLight" from Press\_1 (UF01Q2) to control the screen.

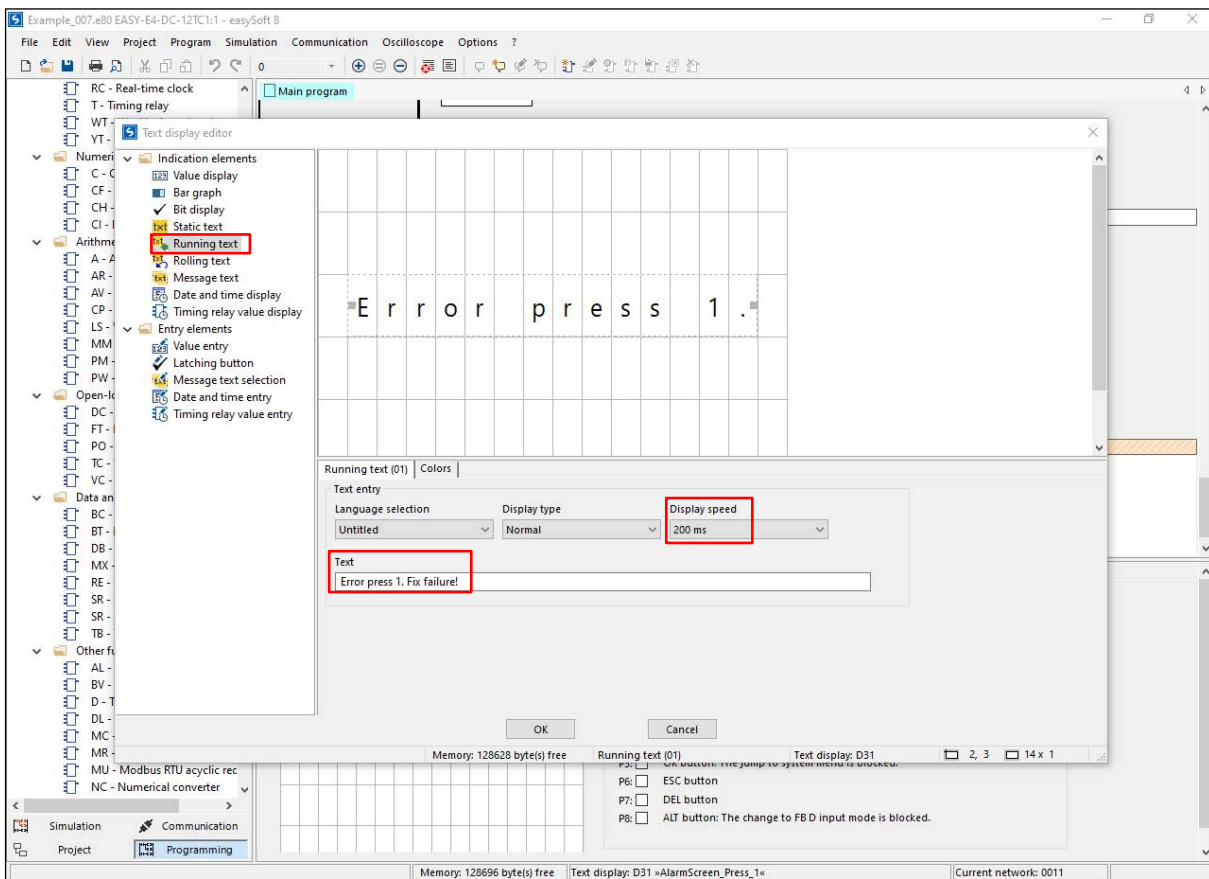


Fig. 111

- ▶ Go to "Text display editor".
- ▶ Drag a "Running text" to the screen.
- ▶ Enter the following text in "Text": "Error press 1. Fix failure!".
- ▶ At "Display speed" select the time of 200 ms.
- ▶ Confirm this with "OK".

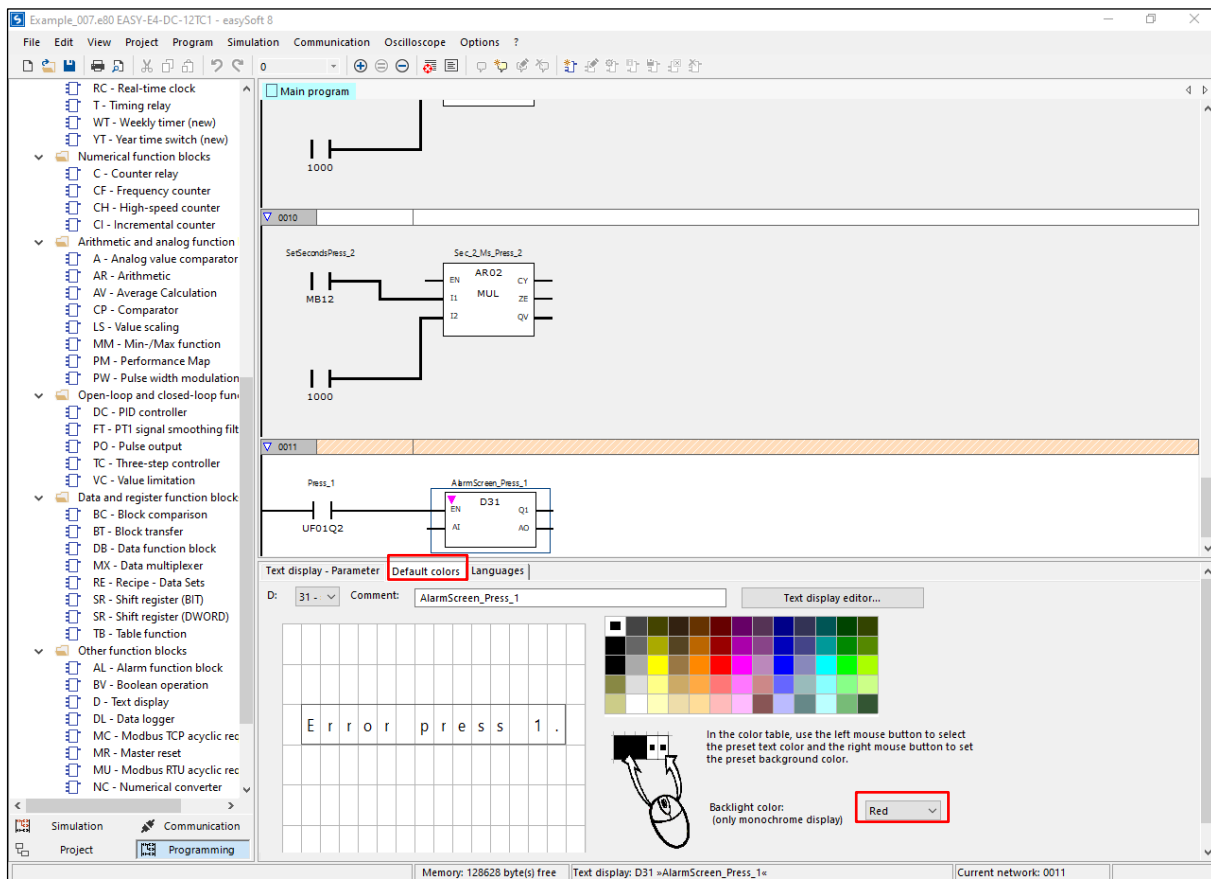


Fig. 112

- ▶ Select the "Default colors" tab.
- ▶ Select the colour "Red" at "Backlight color".

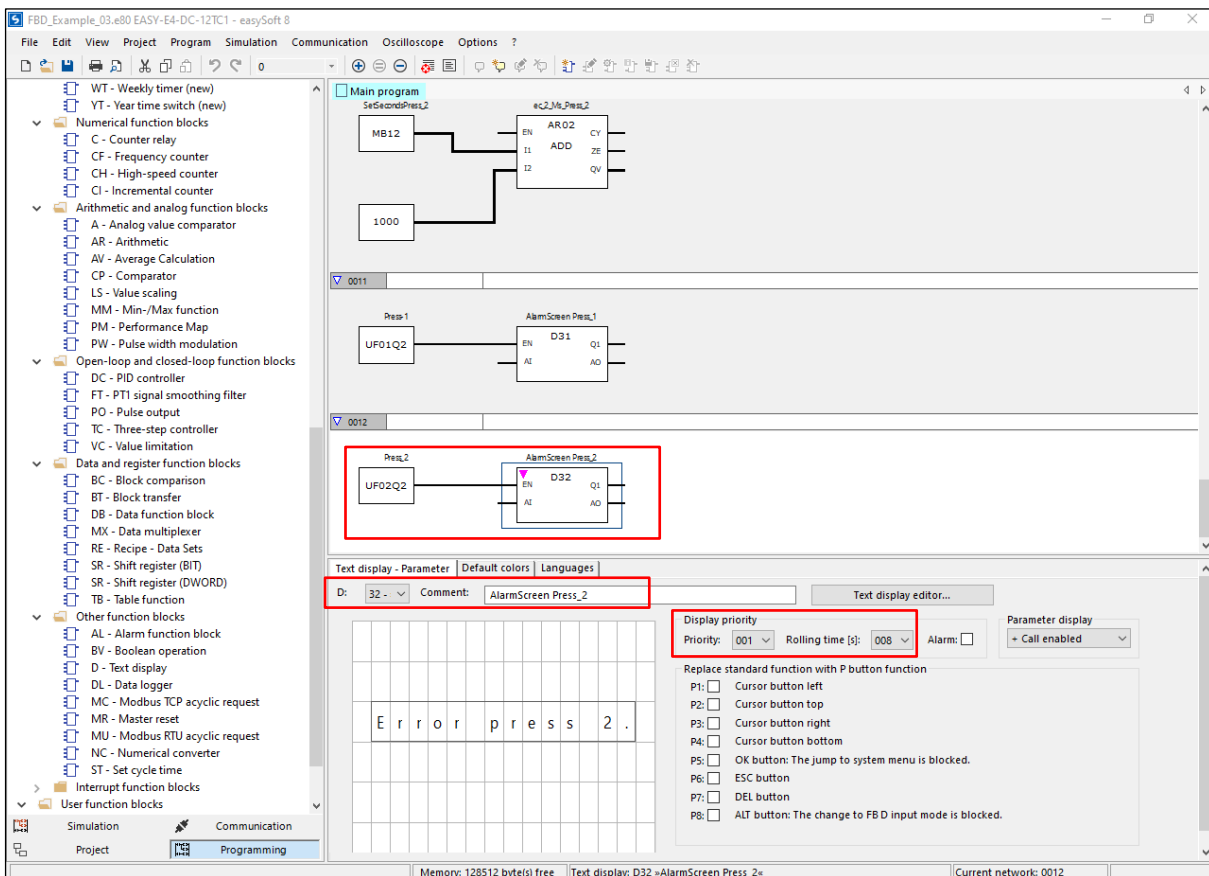


Fig. 113

Repeat this for the error of Press\_2.

- ▶ Create a "D-Text display" 32 here on the worksheet and give it the name "AlarmScreen\_Press\_2". Control this with the error message of Press\_2 (UF02Q2).
- ▶ Also create a "Running text" with the text "Error press 2. Fix failure!".
- ▶ Send the program to the easy and test its operation.

When a fault occurs, a red screen appears with a running text on it.

If both errors are high, the alarm screens are alternated.

The priority setting ensures that alarms have a higher priority than the standard input screens. Should an input screen be displayed, it is overwritten by the alarm screen.

## 2.17 Assign fixed IP-address

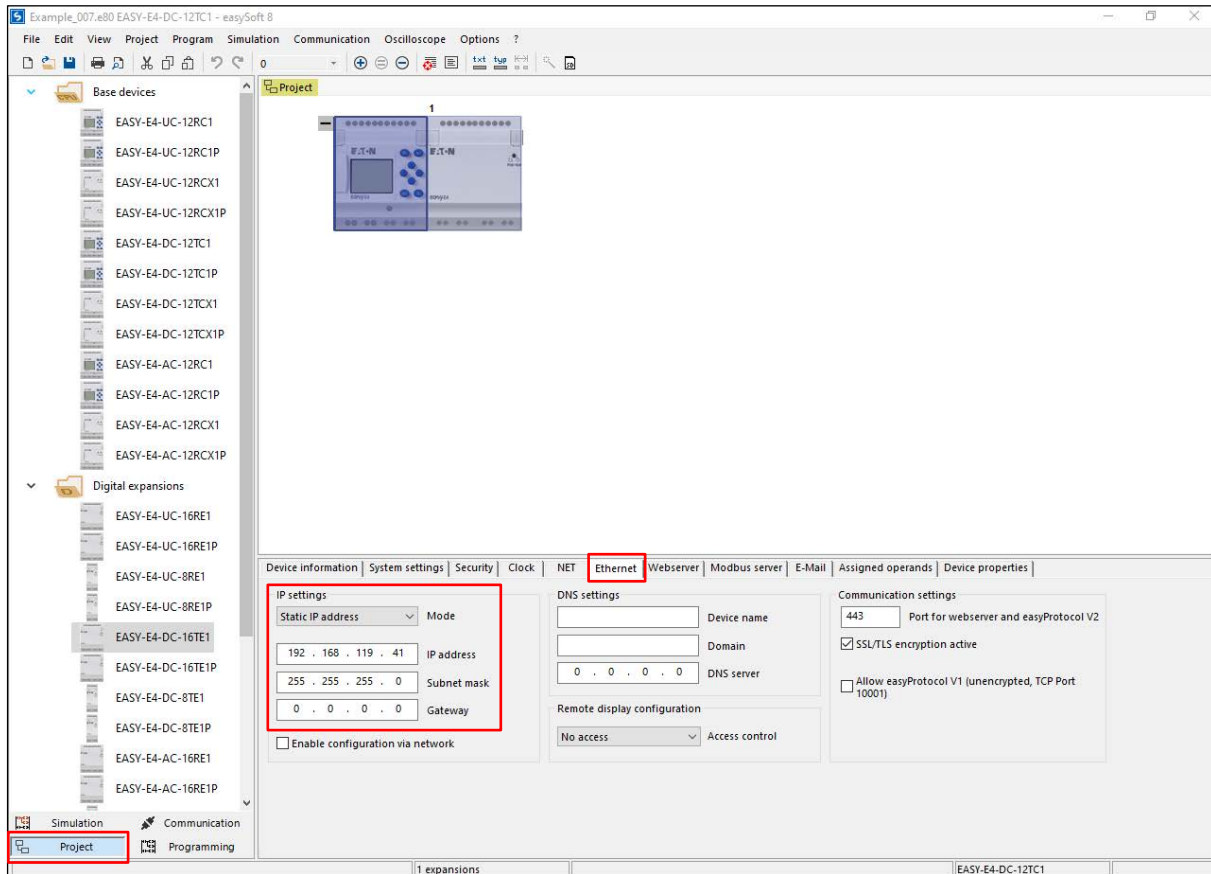


Fig. 114

Up to now, communication with the easy has been established via the automatically obtained IP address. It is also possible, however, to assign a fixed IP address to the easyE4. This is necessary if an easyE4 is to be included in an existing network, or if multiple easy devices must communicate with each other, or if communication with an external visualization system is required.

- ▶ Go to the "Project" menu and select the base device.
- ▶ Go to the "Ethernet" tab.

In "IP-settings" at "Mode" → "Static IP address" is selected.

The following address is entered:

IP address: 192.168.119.41  
Subnet mask: 255.255.255.0

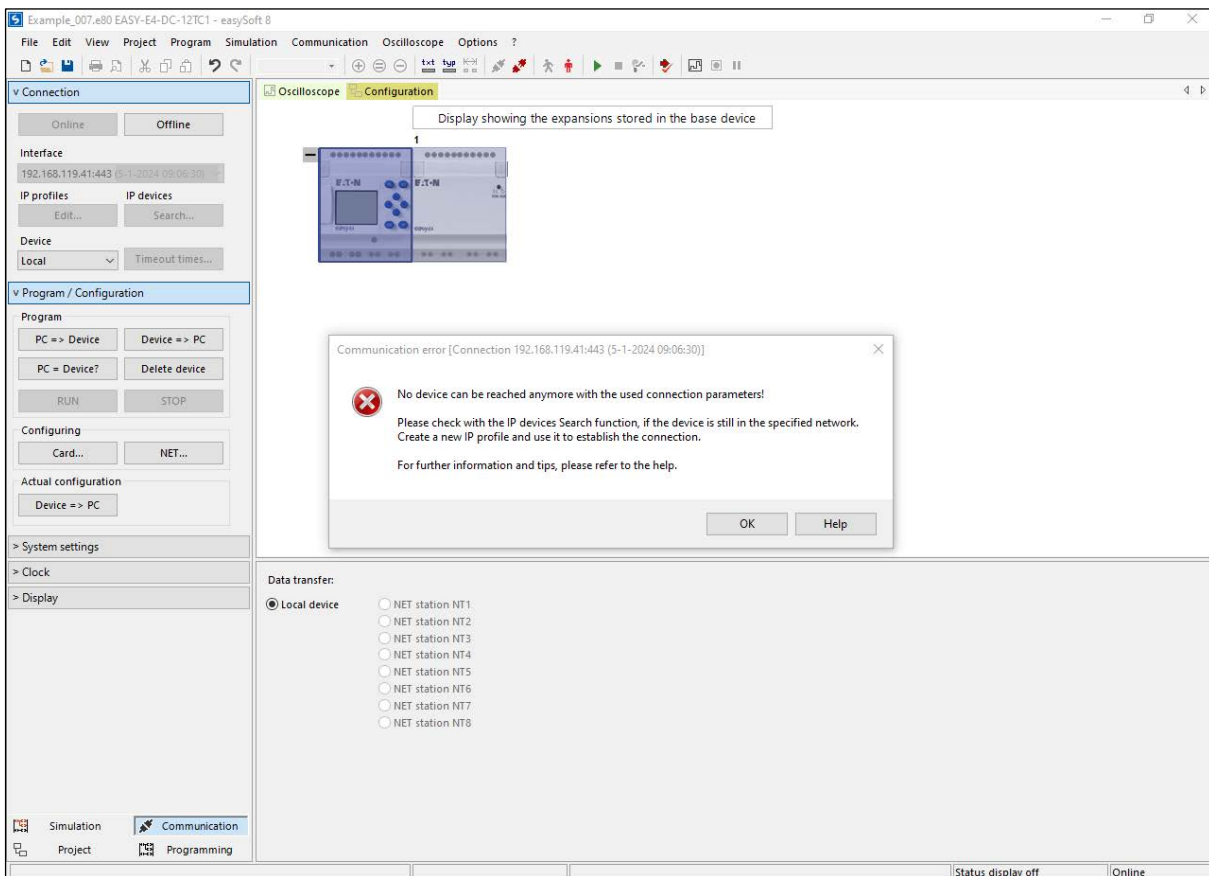


Fig. 115

- ▶ Go to the "Communication" menu.
- ▶ Send the program to the easyE4.

After the program has been transferred, a notification appears that the IP address can no longer be reached. This is correct. The program has been sent to the easyE4 using the old (Auto IP) IP address. However, due to the transfer, the IP address in the easyE4 has changed to the new address:

- ☒ IP address 192.168.119.41
- ☒ Subnet mask 255.255.255.0.

This new IP address can be read at the bottom of the easyE4 screen.

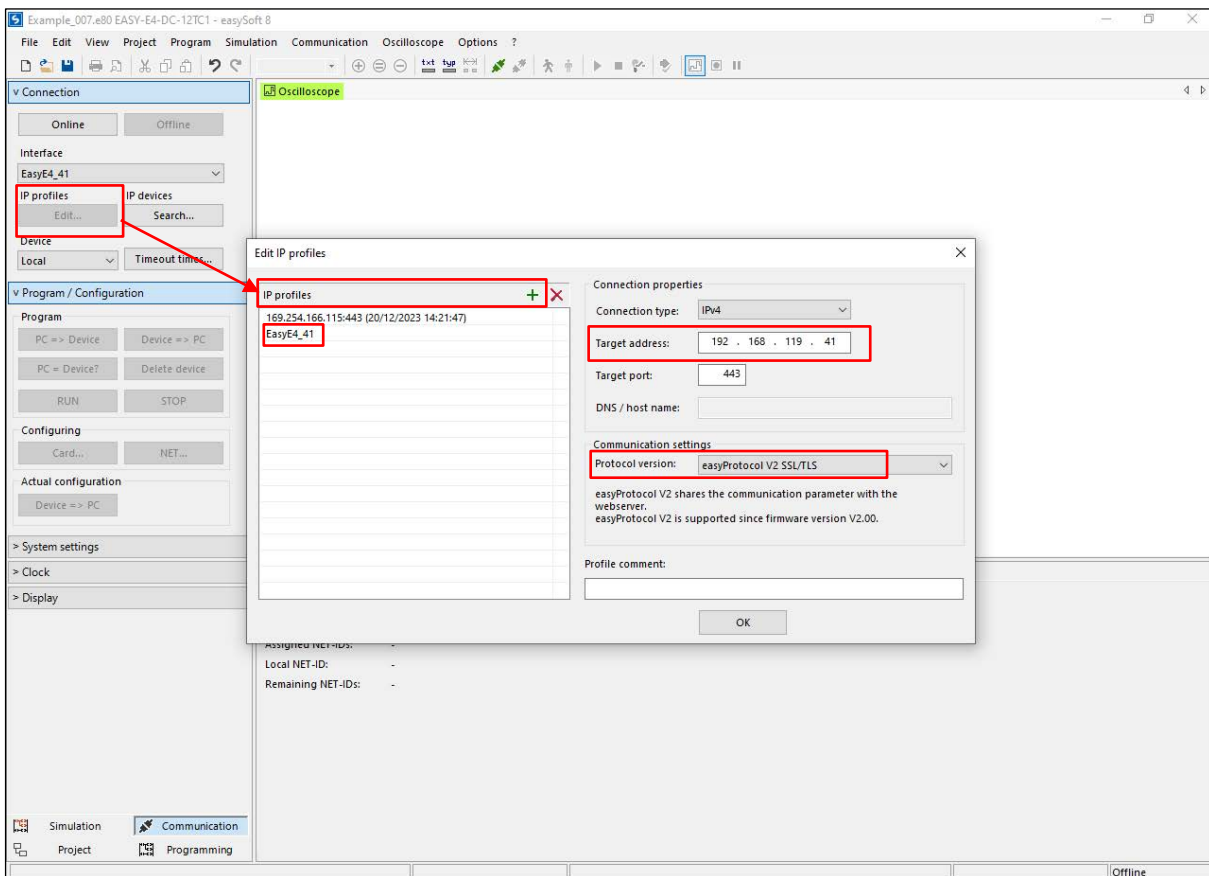


Fig. 116

In the chapter “Transferring the Program to the easyE4,” the IP address of the easyE4 was searched for and used for communication via “Search.”

A second possibility is to enter it manually in the easySoft.

- In "IP profiles", click on "Edit".

A new window is opened.

- Click on "+" at "IP profiles".
- Write a name for the profile on the empty line. In this example “EasyE4\_41”.
- Enter as “Target address” the IP-address 192.168.119.41.
- Select “easyProtocol V2 SSL/TLS” under “Communication settings” because secure communication has been chosen.
- Confirm all this with "OK".

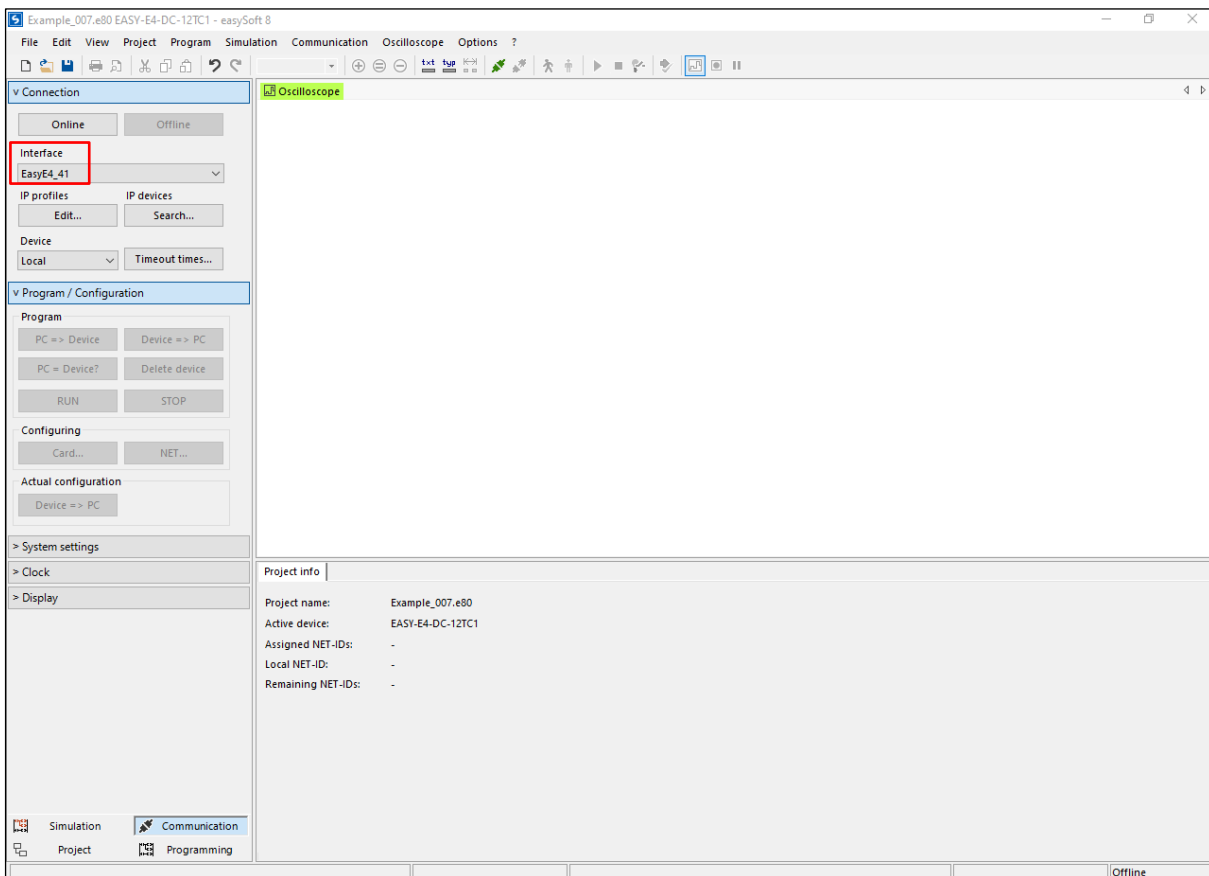


Fig. 117

“EasyE4\_41” can now be chosen at “Interface”. However, communication has not yet been established. The computer's Ethernet interface is still set to automatic address referencing. This must be adjusted.

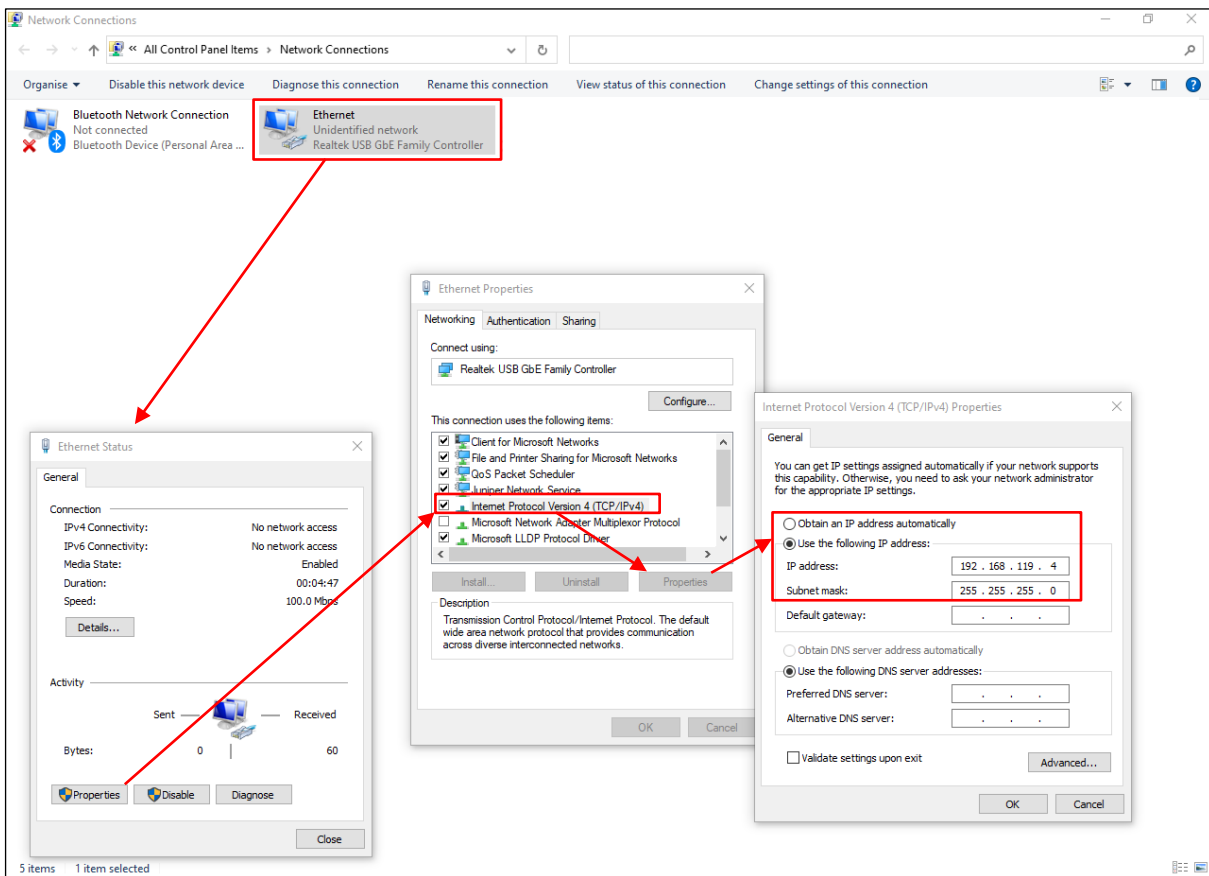


Fig. 118

In the chapter “Transferring the Program to the easyE4,” it is explained how the computer's IP address can be adjusted.

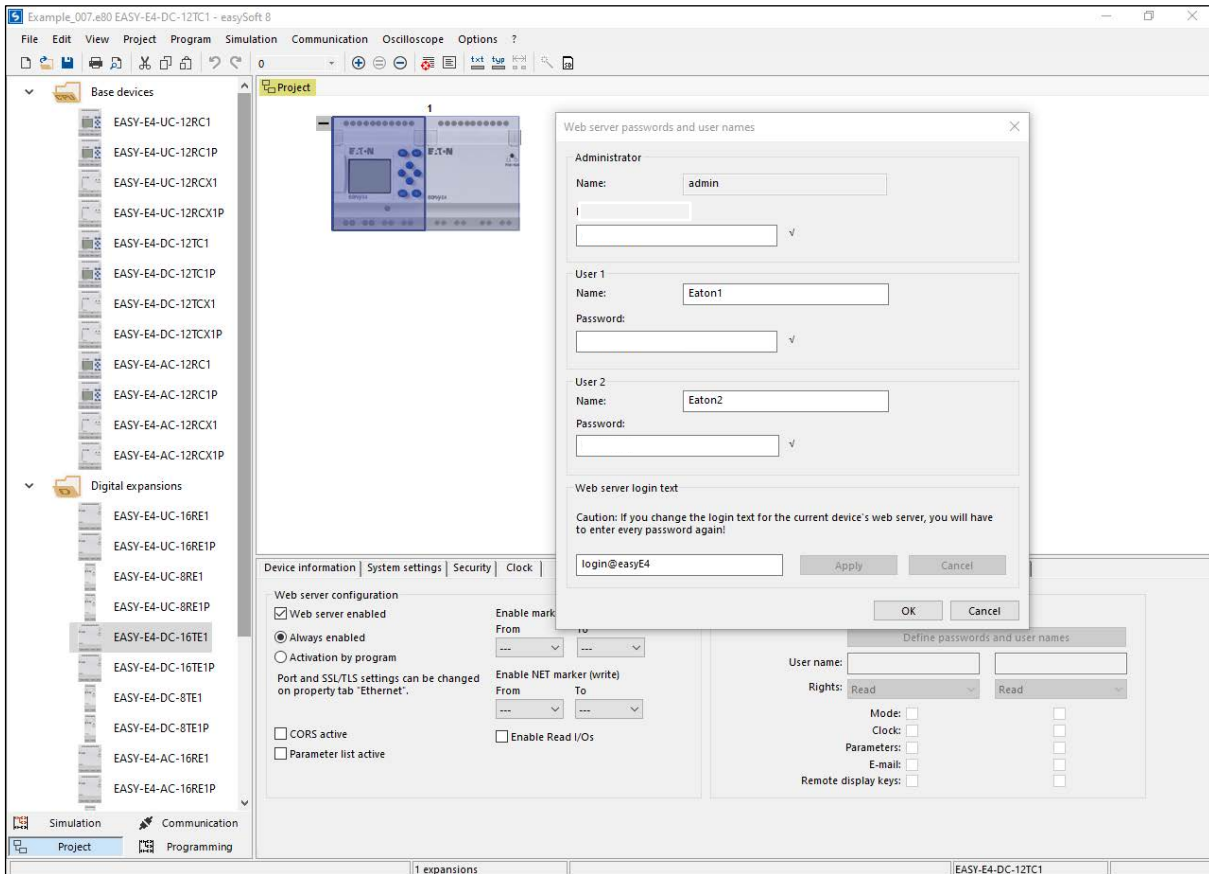
The computer's network adapter must also be assigned a fixed IP address that is within the range of the easyE4. The computer receives:

IP address: 192.168.119.4  
Subnet mask: 255.255.255.0.

- Confirm all this and close the settings adjustment.

The communication with the easyE4 is once again established.

## 2.18 Webserver



This figure deviates from the truth.

Fig. 119

The easyE4 also has a web server.

- ▶ Go to the "Project" menu and select the "Webserver" tab.
- ▶ Select "Web server enabled" here.

The screen immediately appears for entering the desired passwords.

In this example it is:

- ☒ Name = admin
    - Password = Admin\_01
  - ☒ User 1 = Eaton1
    - Password = Eaton\_01
  - ☒ User 2 = Eaton2
    - Password = Eaton\_02
- ▶ Confirm all this with "OK".

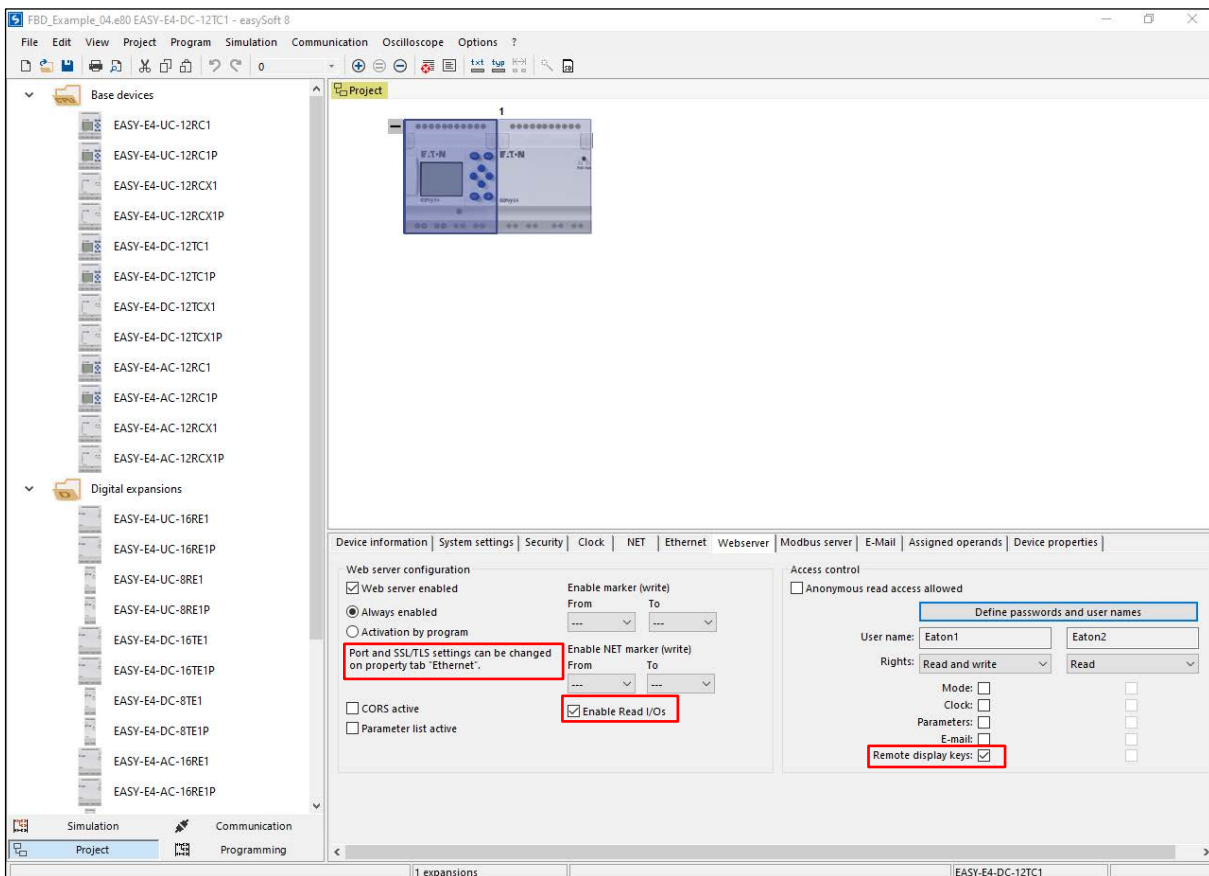


Fig. 120

Normally, the safe SSL/TLS connection is on. This can, when needed, be adjusted under the "Ethernet" menu.

By checking "Enable Read I/O's", it is possible to view the status of the various I/Os and Merkers via the web server.

- ▶ Give the user "Eaton1" the "Read and write" rights.
- ▶ Check the "Remote display keys" box here as well.

This makes it possible to use the displayed buttons of the easyE4 via the web browser.

The user "Eaton2" will only receive "Read" rights.

- ▶ Transfer the program to the easyE4 and set it back to RUN.

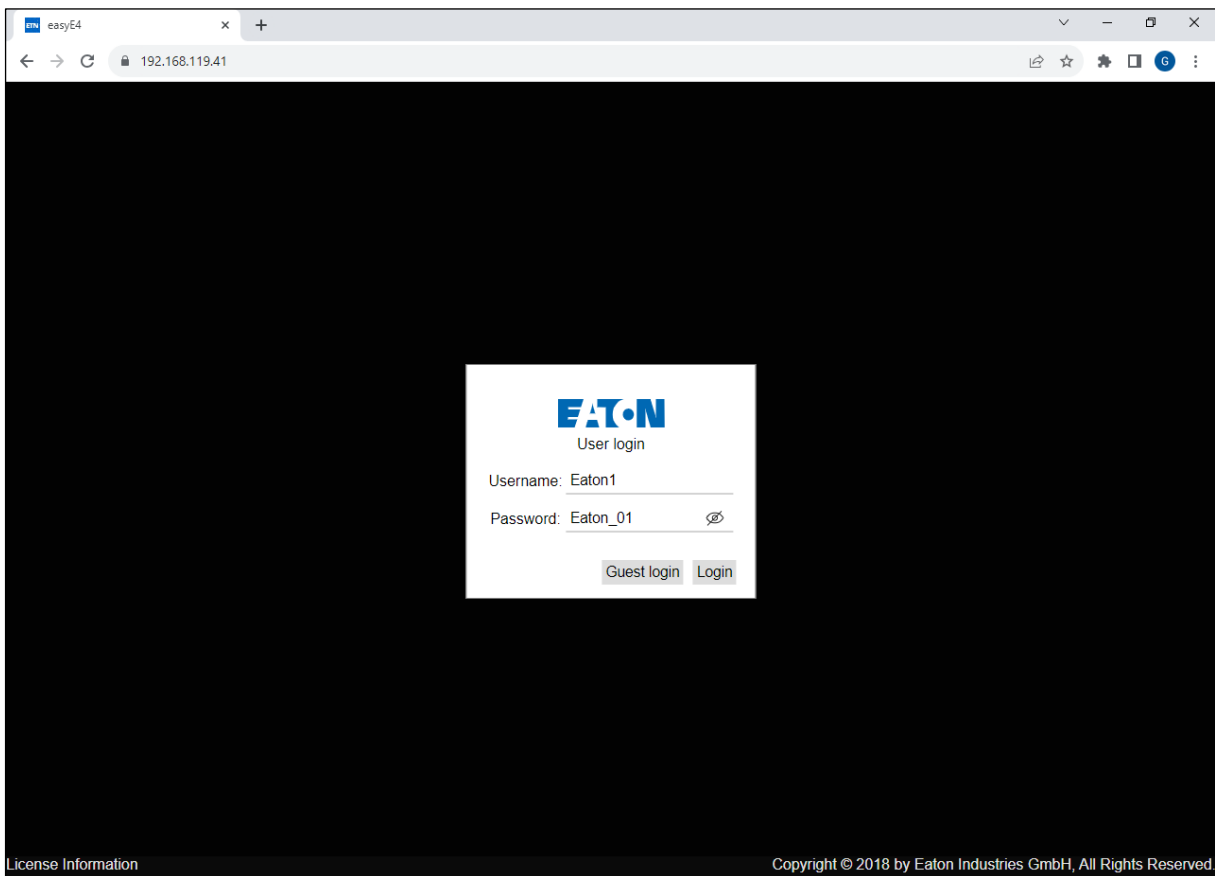


Fig. 121

Because a secure connection is going to be made, the IP address in the web browser must begin with [HTTPS://](https://).

Type the IP address: in the web browser on the computer. The above screen appears.

► Here, enter:

Username : Eaton1

Password : Eaton\_01

► Then click on "Login".

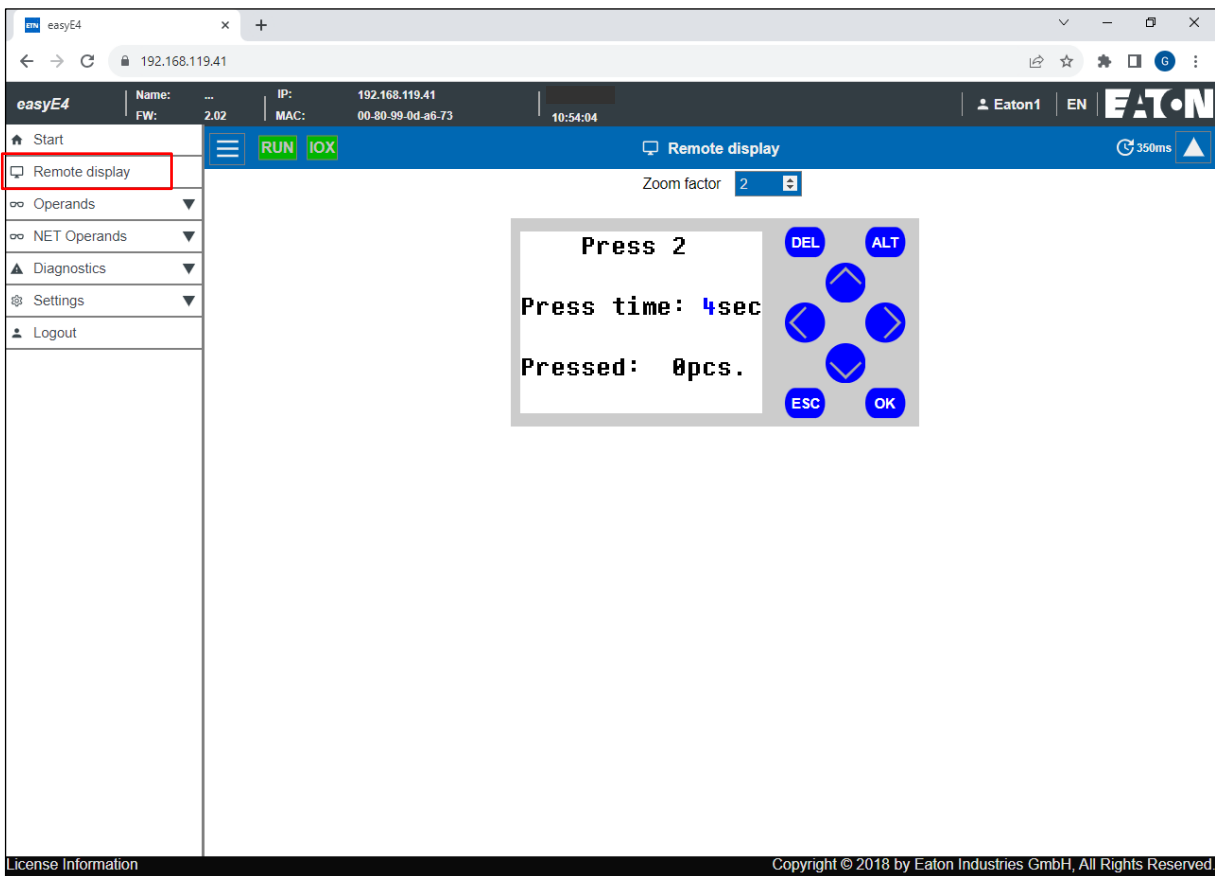


Fig. 122

- ▶ Select "Remote display".

The display of the easyE4 appears on the screen, including buttons. The operation of the easyE4 can now be done via the web browser.

- ▶ Click on the P4 = "✓".

The screen of press 2 is displayed.

- ▶ Adjust the value of the press time to 4 seconds using the "ALT" key.
- ▶ Operate the inputs of the easyE4.

The press time has now been adjusted to 4 seconds.

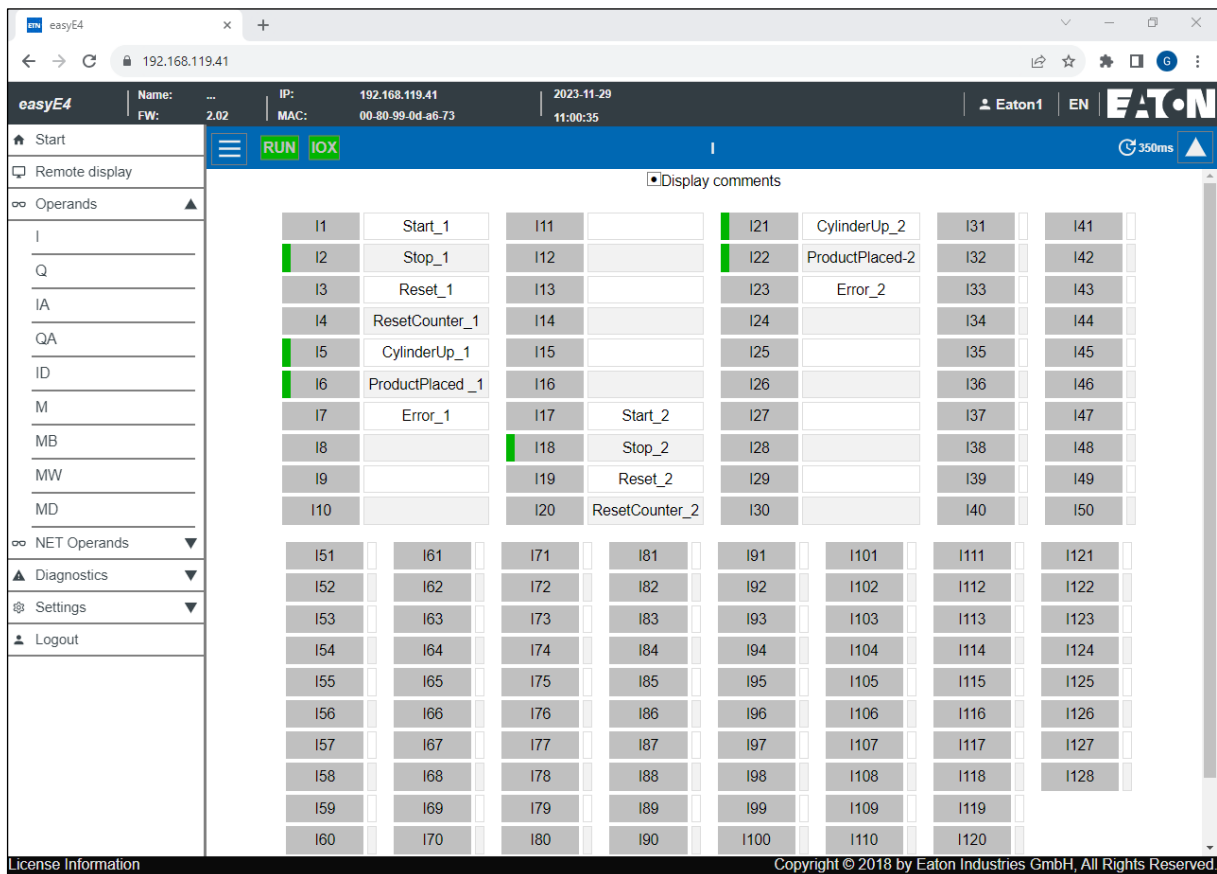


Fig. 123

It is also possible to view the status of the various I/Os.

- Click on "Operands" and select "I".

The inputs are now displayed.

- Select "Display comments" to also display the comments assigned to the inputs.

## Chapter 3 Example programs

### 3.1 Format SD-card

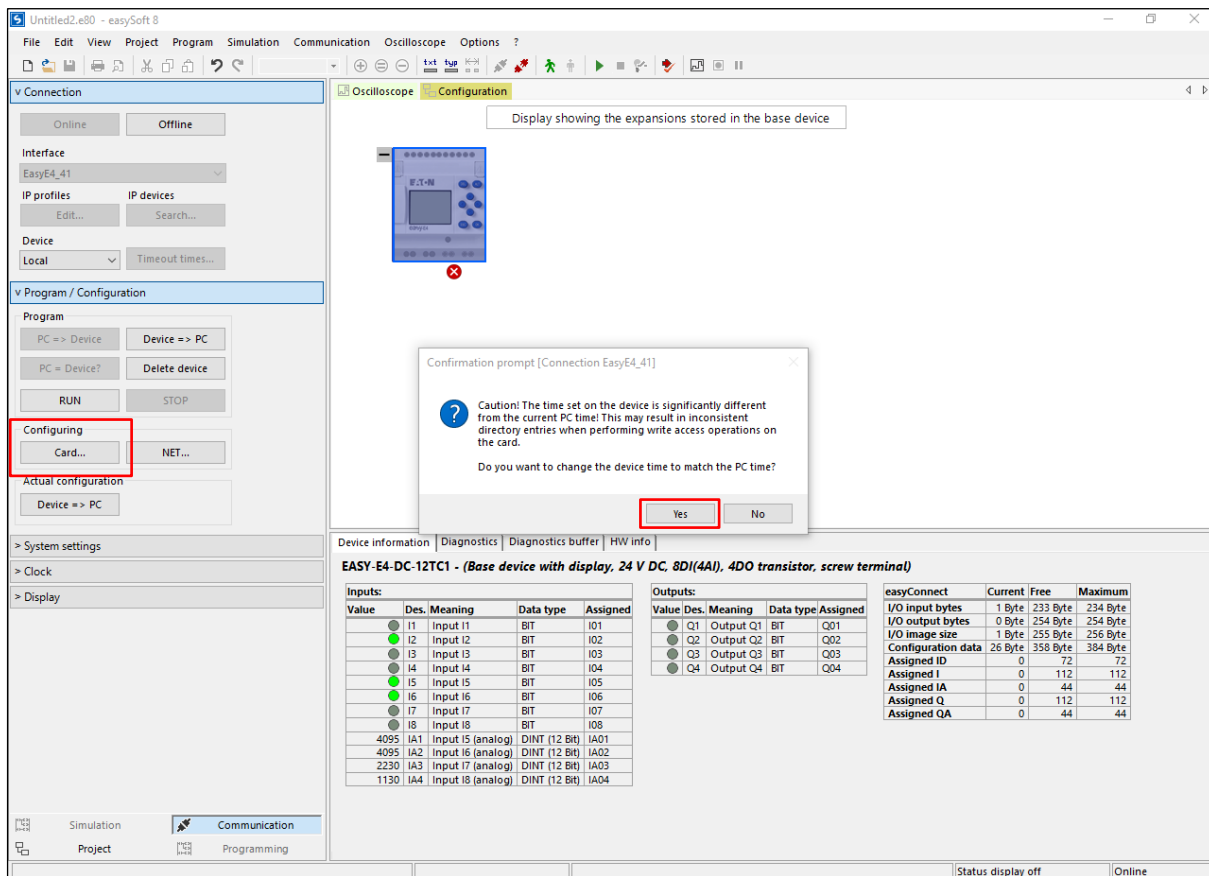


Fig. 124

Put a card in the SD-slot.



- Open a new program.
- Drag the desired easyE4 onto the worksheet.

A program is not required.

- Go to the "Communication" menu and connect with the easyE4.
- Under "Program / Configuration" select → "Configuring" → "Card".

In this example, a message appears indicating that the time in this easyE4 is not the same as that of the PC.

- Click on "Yes" to synchronize the time.

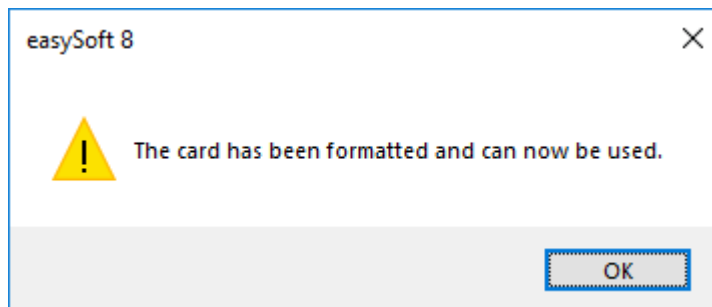
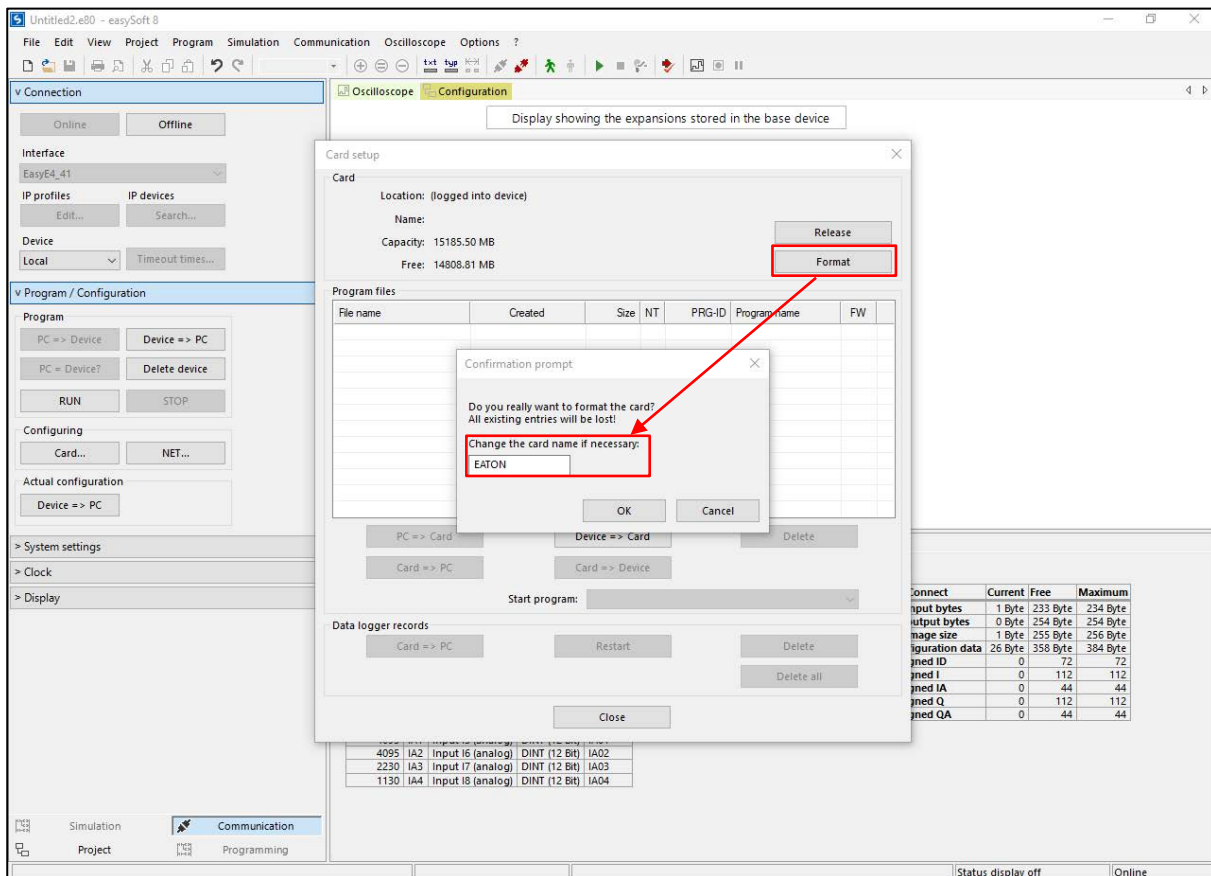


Fig. 125

- In the next window, select the "Format" button.

In the "Safety prompt" window, a name can be entered for the SD-card. In this example, the card is named "Eaton".

- Click on "OK" to confirm.

The software returns with the message that the card has been formatted.

- Confirm this with "OK".

## 3.2 Boot logo

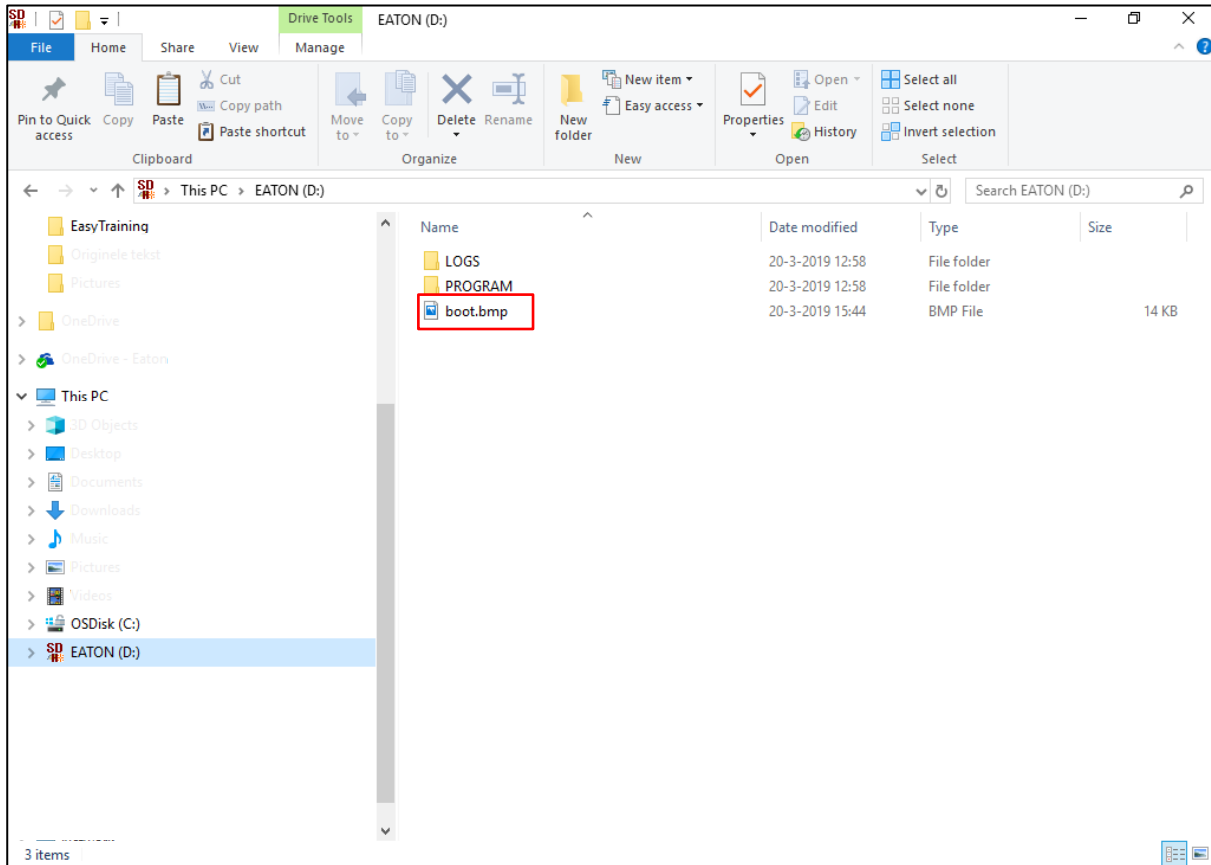


Fig. 126

When the easyE4 is started up, the Eaton logo appears on the screen. This can be adjusted to the logo of your choice. The image may be a maximum of 128 x 64 pixels (width x height) and monochrome. The file's name must always be "boot.bmp".

**TAKE NOTE!** The logo must be in the root of the SD card.

- ▶ Copy the logo created to the SD-card.
- ▶ Insert the SD-card in the easyE4 and connect the easyE4.

The new self-made logo appears.

## 3.2 Data storage

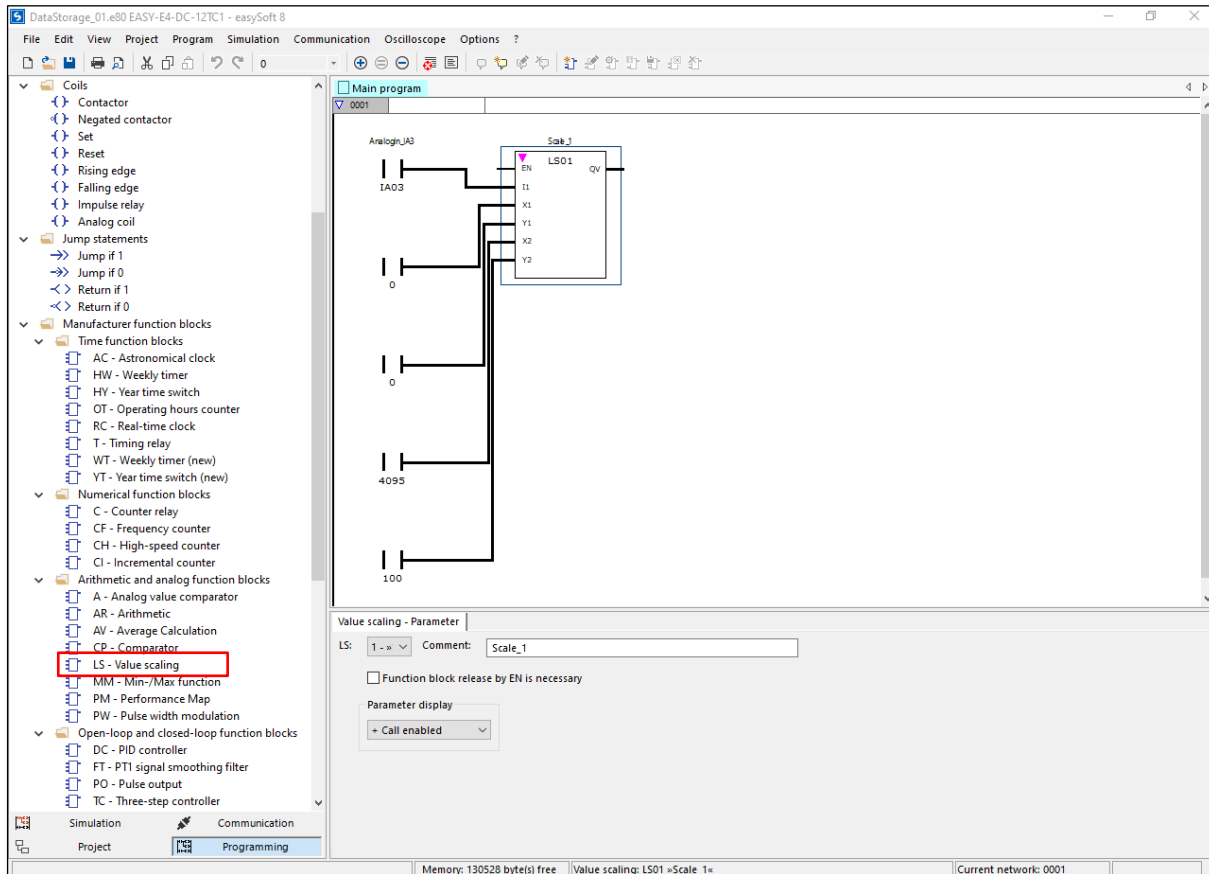


Fig. 127

If the SD card is formatted, it is also possible to save data on this card.

In the following example, an analog value is stored every second.

For this, an analog input is used, where the value 0-100 is generated via a scaling function block.

- ▶ Open a new program and select Ladder as the programming language.
- ▶ Drag an “LS - Value scaling” function block onto the worksheet and name it “Scale\_1”.
- ▶ Remove the check mark from “Function block release by EN is necessary”. This way the FB is always called in the program.

Connect the analog input IA3, named “AnalogIn\_IA3”, to “I1” of this FB. This is physically connected to I7 of the base unit.

The analog input is 12-bit and can therefore generate a value between 0–4095. This range is set to the inputs “X1” = 0 and “X2” = 4095.

The FB must output a number between 0–100. This range is set to the outputs “Y1” = 0 and “Y2” = 100.

A value of 0–100 can now be present at the output “QV”.

- ▶ Save the program under the name “DataStorage\_01”.

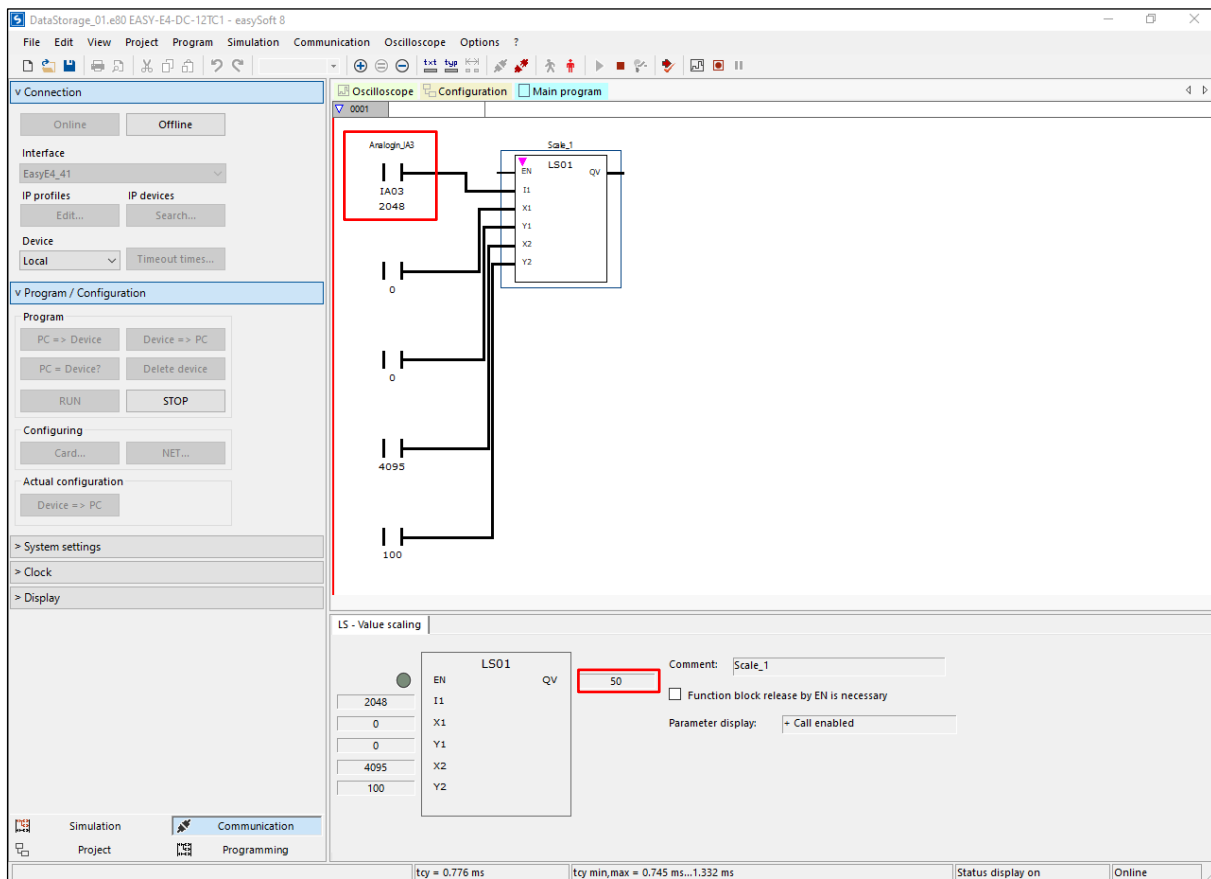


Fig. 128

- Upload the program to the easyE4 and test the function.

If the value 2048 (approximately 1/2 of 4095) is applied to the input, the output "QV" of the FB "Scale\_1" will have the value 50.

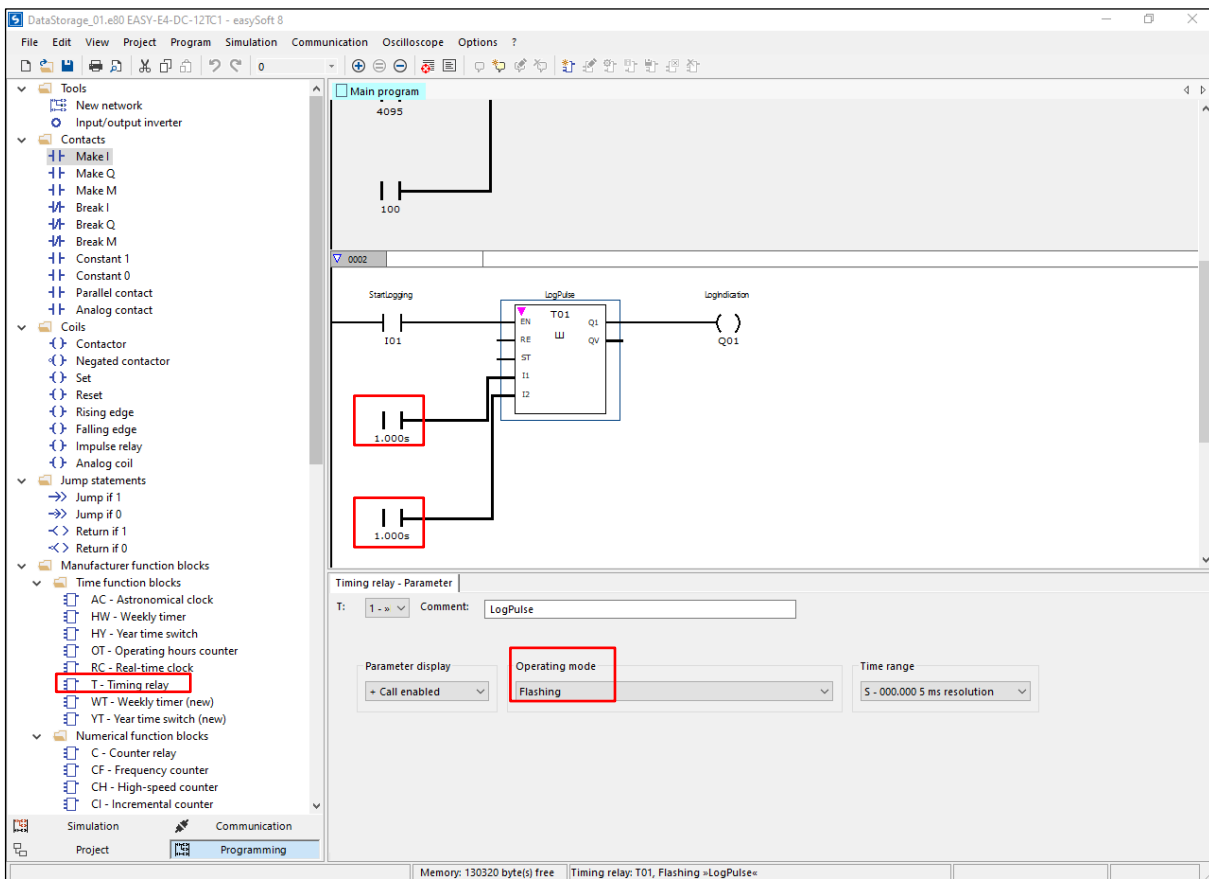


Fig. 129

Create a new rung.

- Drag a “T – Timing relay” onto the screen and name it “LogPulse”.

This timing relay will give a signal every second as soon as input I1 = “StartLogging” is high.

- Select a “Flashing” timer for this.

The timer's signal will be used to save the measured value to the SD card every second.

To be able to see whether the log pulse is being output, output Q1 of the FB “LogPulse” is connected to output Q1 “LogIndication” of the easyE4.

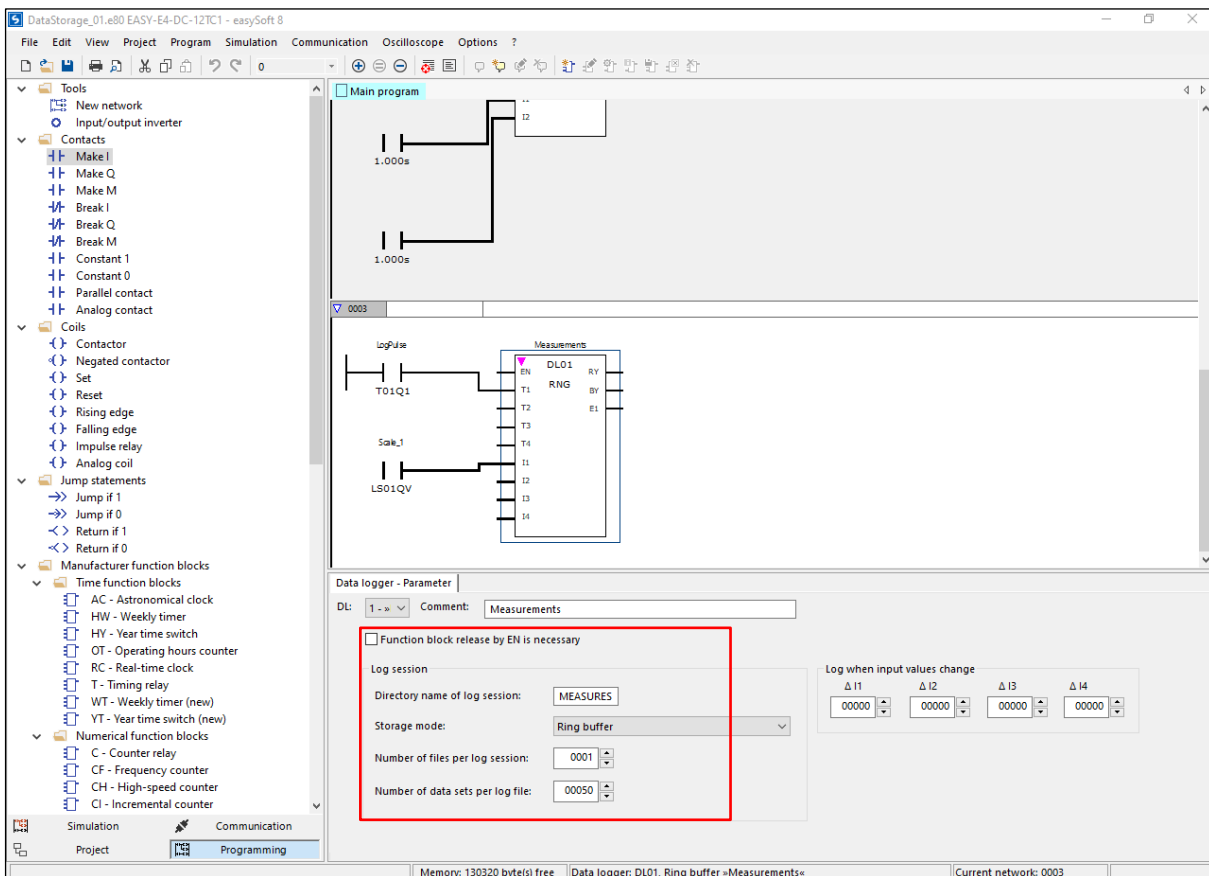


Fig. 130

- Create a new rung and drag a “DL – Data logger” function block onto the worksheet and name it “Measurements”.

The file that must be saved, must be named "Measures".

If “Storage mode” is chosen for a ring buffer.

For “Number of files per log session,” the number 1 is entered. For “Number of data sets per log file,” enter 50. If more than 50 measurements have now been saved, the first measurements are overwritten again.

The output of the “LogPulse” timer is connected to the “T1” input. The FB now receives a pulse every second to start logging.

The value from the scaling function block “Scale\_1” is placed at “I1”.

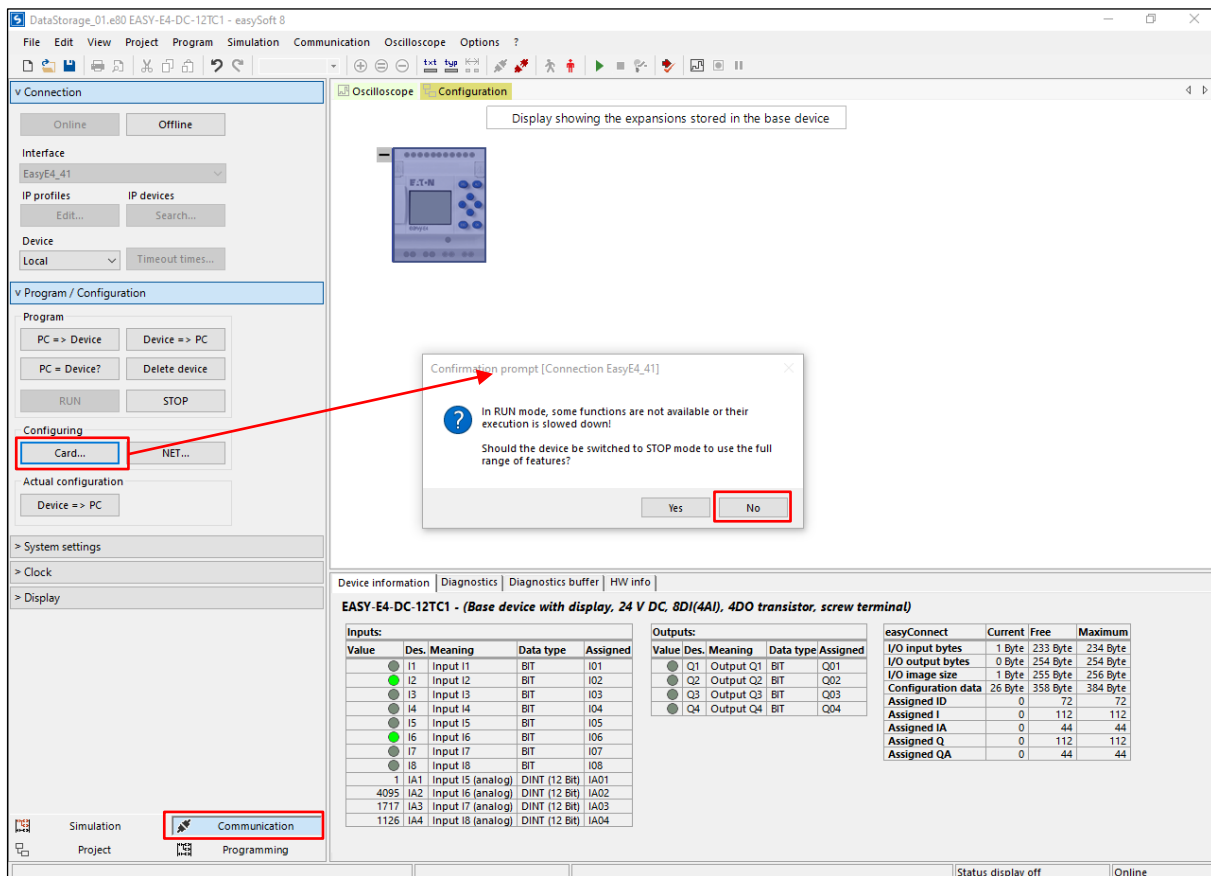


Fig. 131

- Upload the program to the easyE4 and test the function.
- Make the "StartLogging" input (I1) high.

The output "LogPulse" (Q1) will now flash. With every pulse, the value of "Scale\_1" is written on the SD card.

The data is now written to the card and can be saved on the PC.

- Go to the "Communication" menu.
- Select "Card" in the "Configuring" menu.

A message appears indicating that some functions are not available when the EasyE4 is in RUN mode.

- Click "No" here because the files can be retrieved from the card.

The following screen now appears on the easyE4 display.



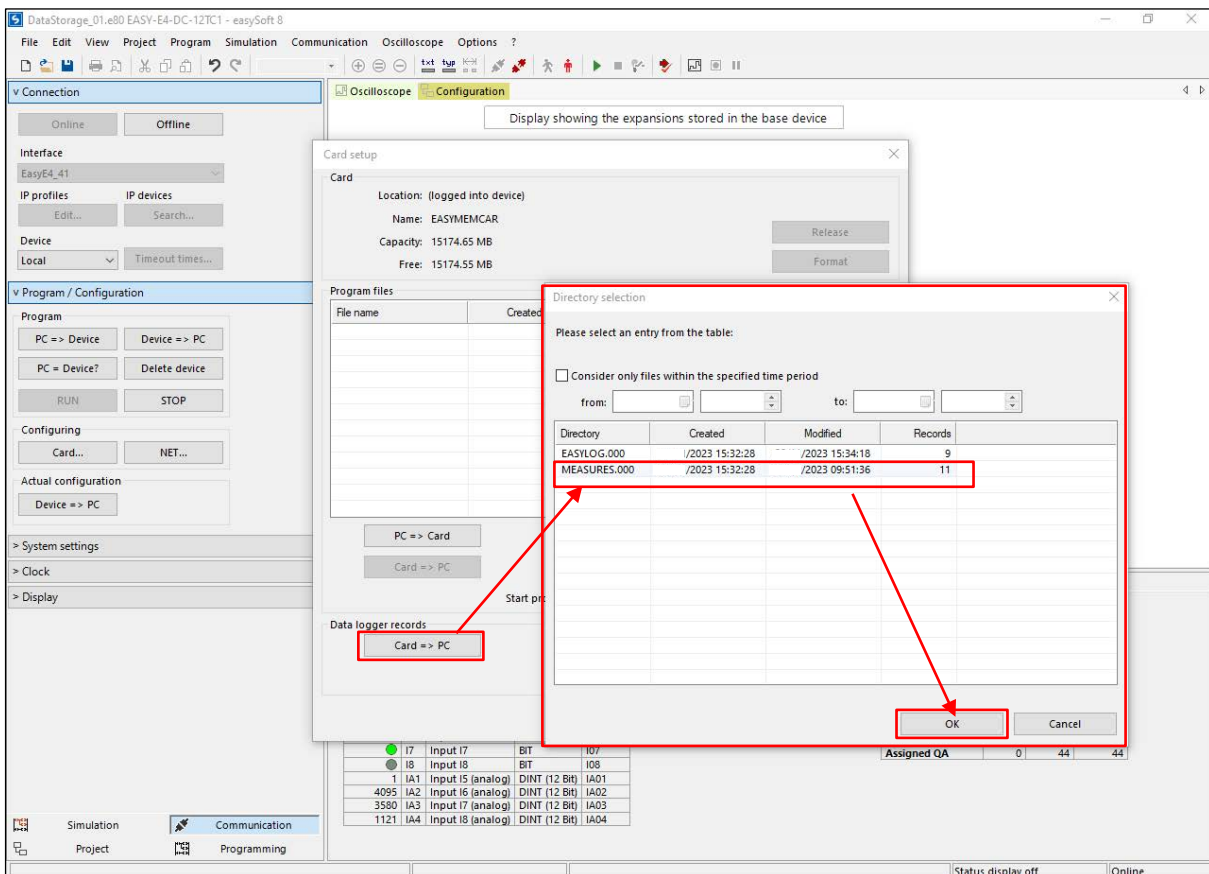


Fig. 132

The following window opens.

- Click on "Card => PC". A new window appears.

This contains the name "MEASURES.000" with a date and time following it.

- Select this file and click on "OK".

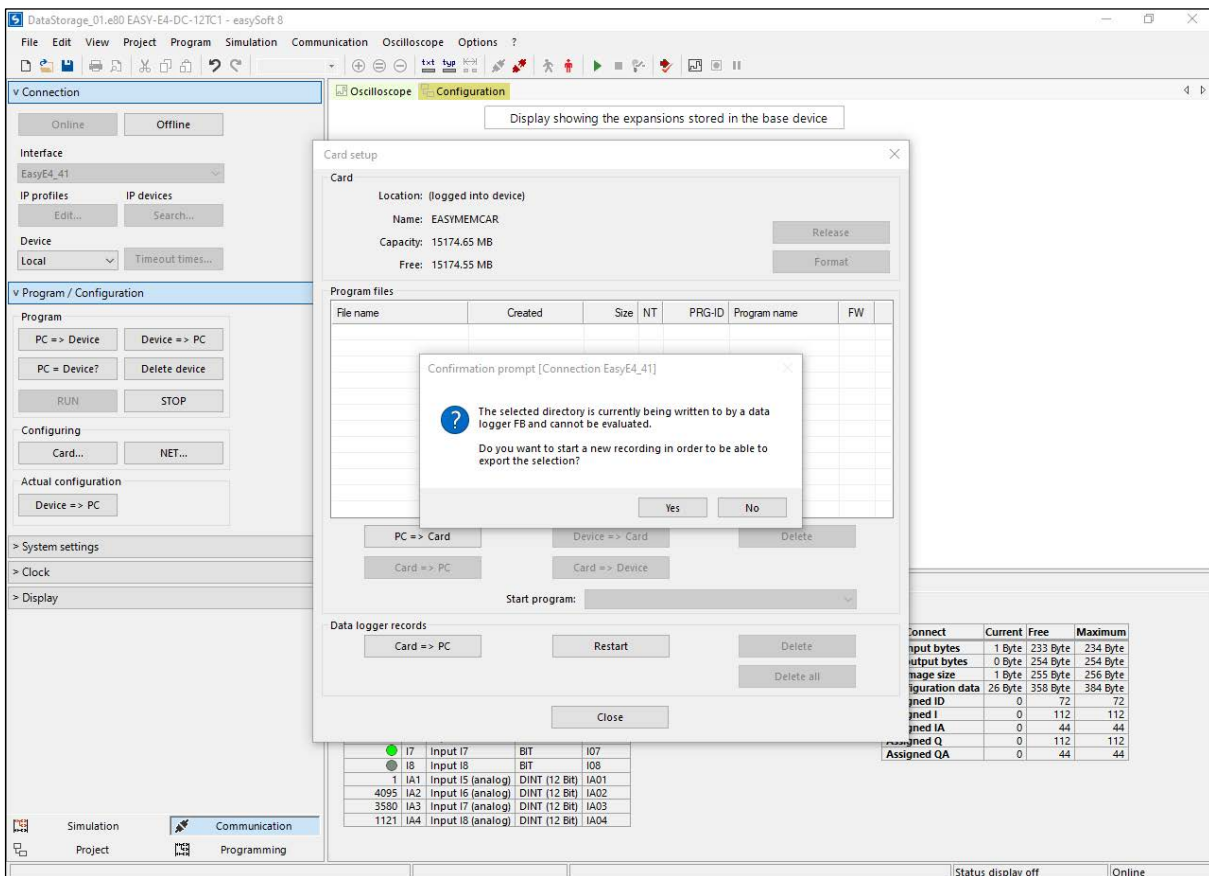


Fig. 133

A message appears that this file cannot be sent to the PC because the file is still in use.

- Answered question whether a new recording should be created. Confirm this with "Yes".

The new measurements are now saved under a different file name.

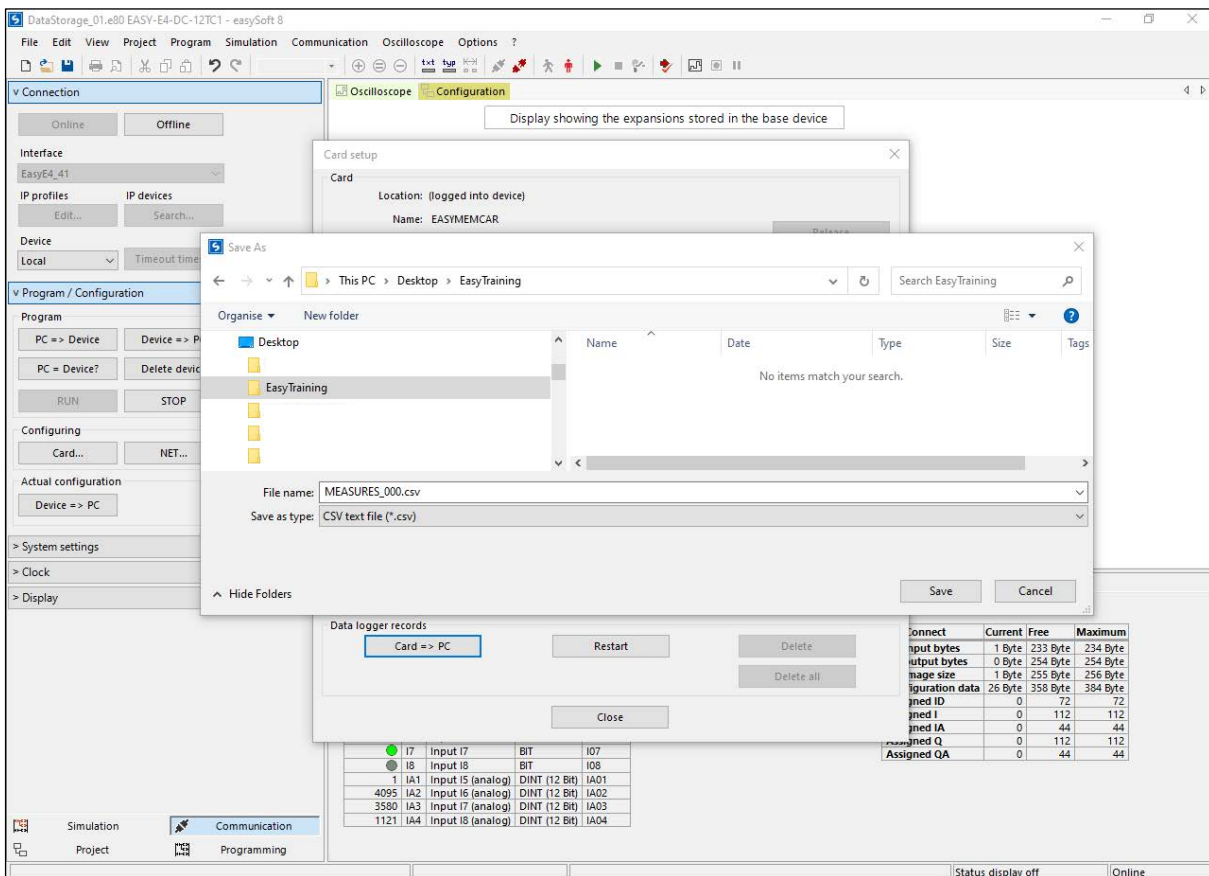


Fig. 134

- ▶ Select the folder "EasyTraining" and save the CSV file there by clicking on "Save".
- ▶ Close all these windows again.

The new measurements are saved in the file named "Measures.001".

|    | A       | B          | C        | D         | E      | F      | G      | H      | I      | J      | K      | L      | M | N | O | P | Q | R | S | T | U |
|----|---------|------------|----------|-----------|--------|--------|--------|--------|--------|--------|--------|--------|---|---|---|---|---|---|---|---|---|
| 1  | Counter | Date       | Time     | Time (ms) | DL01T1 | DL01T2 | DL01T3 | DL01T4 | DL01I1 | DL01I2 | DL01I3 | DL01I4 |   |   |   |   |   |   |   |   |   |
| 2  | 0       | 05/12/2023 | 09:50:34 | 303       | 1      | 0      | 0      | 0      | 91     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 3  | 1       | 05/12/2023 | 09:50:36 | 323       | 1      | 0      | 0      | 0      | 79     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 4  | 2       | 05/12/2023 | 09:50:38 | 343       | 1      | 0      | 0      | 0      | 28     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 5  | 3       | 05/12/2023 | 09:50:40 | 363       | 1      | 0      | 0      | 0      | 81     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 6  | 4       | 05/12/2023 | 09:50:42 | 273       | 1      | 0      | 0      | 0      | 1      | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 7  | 5       | 05/12/2023 | 09:50:44 | 293       | 1      | 0      | 0      | 0      | 47     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 8  | 6       | 05/12/2023 | 09:51:28 | 934       | 1      | 0      | 0      | 0      | 36     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 9  | 7       | 05/12/2023 | 09:51:30 | 954       | 1      | 0      | 0      | 0      | 36     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 10 | 8       | 05/12/2023 | 09:51:32 | 974       | 1      | 0      | 0      | 0      | 3      | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 11 | 9       | 05/12/2023 | 09:51:35 | 4         | 1      | 0      | 0      | 0      | 64     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |
| 12 | 10      | 05/12/2023 | 09:51:36 | 904       | 1      | 0      | 0      | 0      | 87     | 0      | 0      | 0      |   |   |   |   |   |   |   |   |   |

Fig. 135

Go to the folder on the PC where the file "MEASURES.000" is saved. Open the file with Excel. In column A, the numbering of the measurements can be seen.

In column 'I', the measured values that were set on input I1 of the data logger DL01 are noted.

## 3.3 Easy network

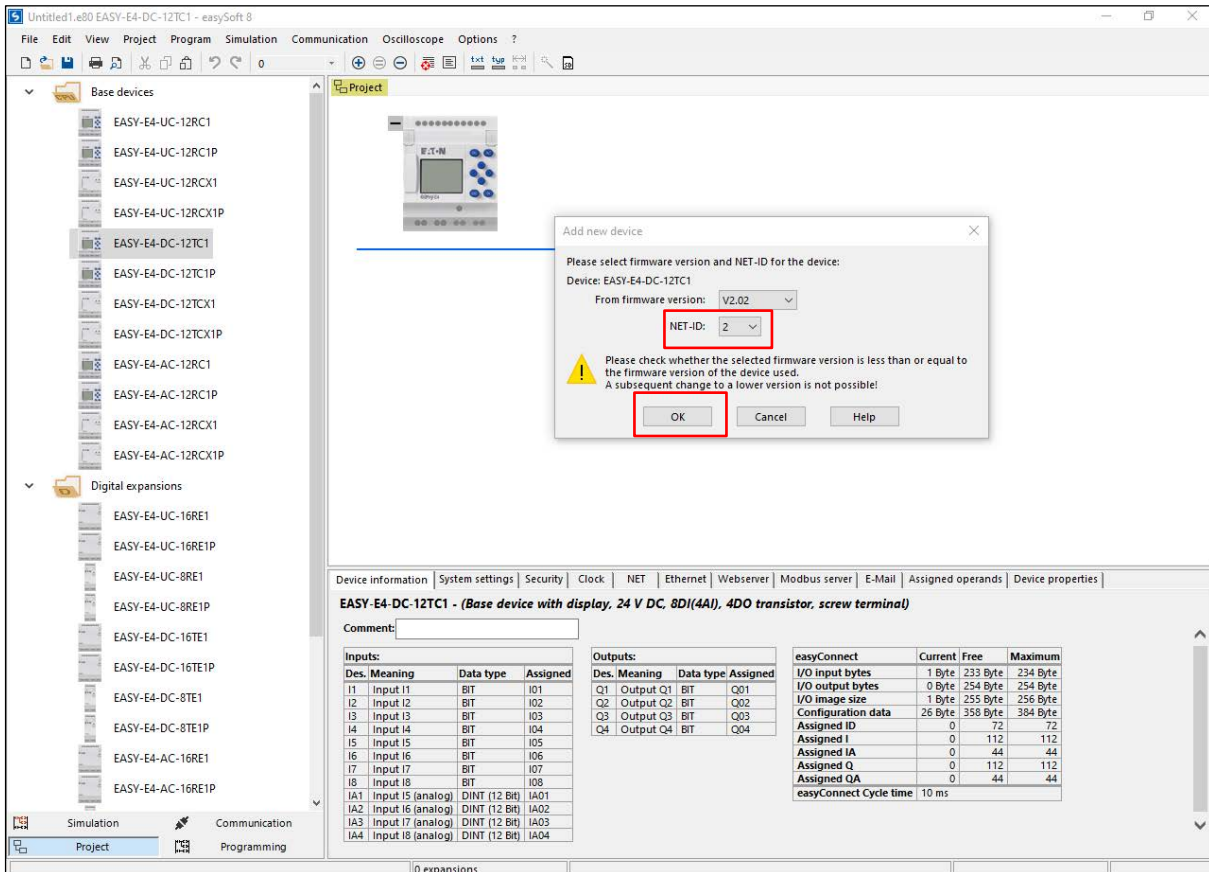


Fig. 136

A maximum of 8 easyE4 control relays can be included in an Ethernet network. In this example, two are placed in a network.

- Connect the easyE4s and the PC to an Ethernet switch.
- Drag a second easy onto the worksheet in easySoft.

A window opens with the question of which "NET-ID" this easyE4 should receive.

In this example "NET-ID" 2 was chosen.

- Confirm this with "OK".

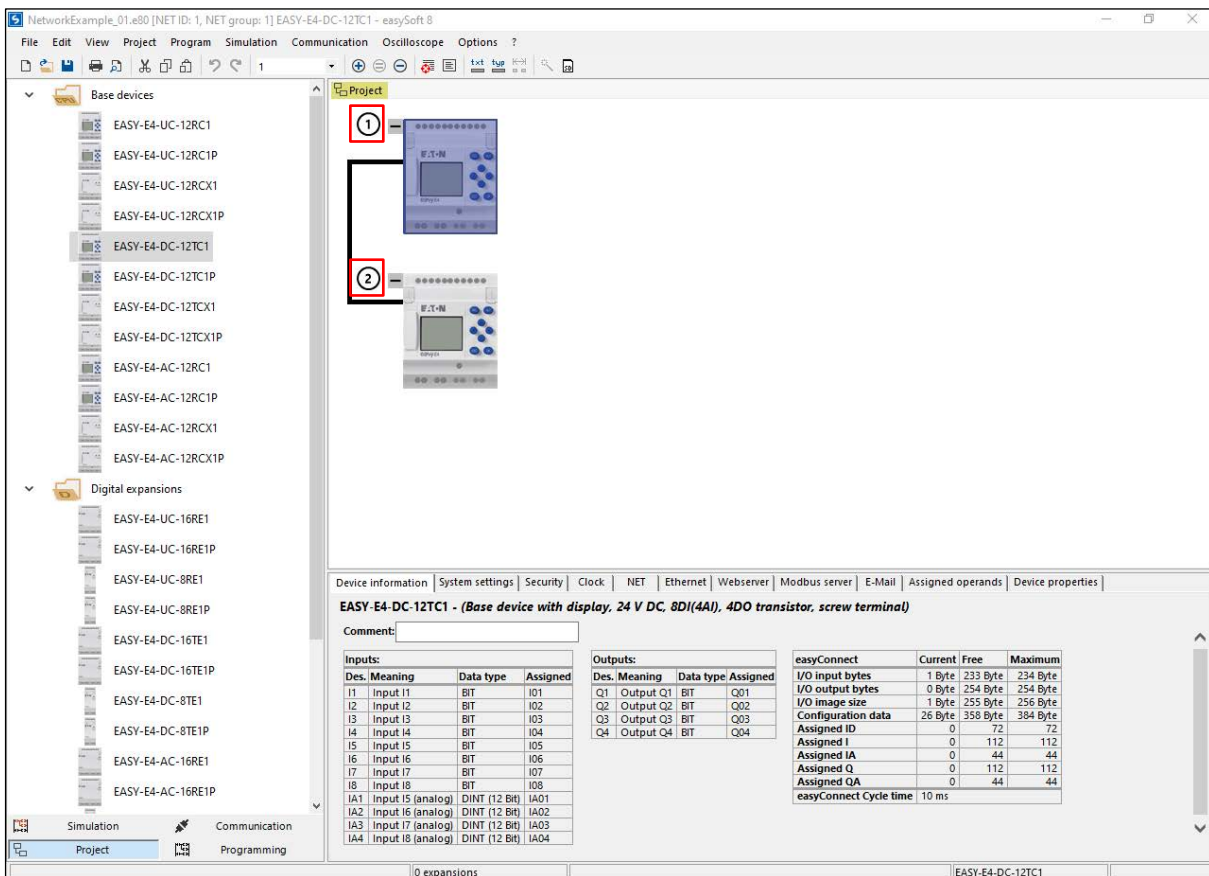


Fig. 137

The second easyE4 has now been placed. The numbers at the top left of the easyE4s indicate which "NET-ID" the easyE4 will receive.

- Save the project under the name "NetworkExample\_01".

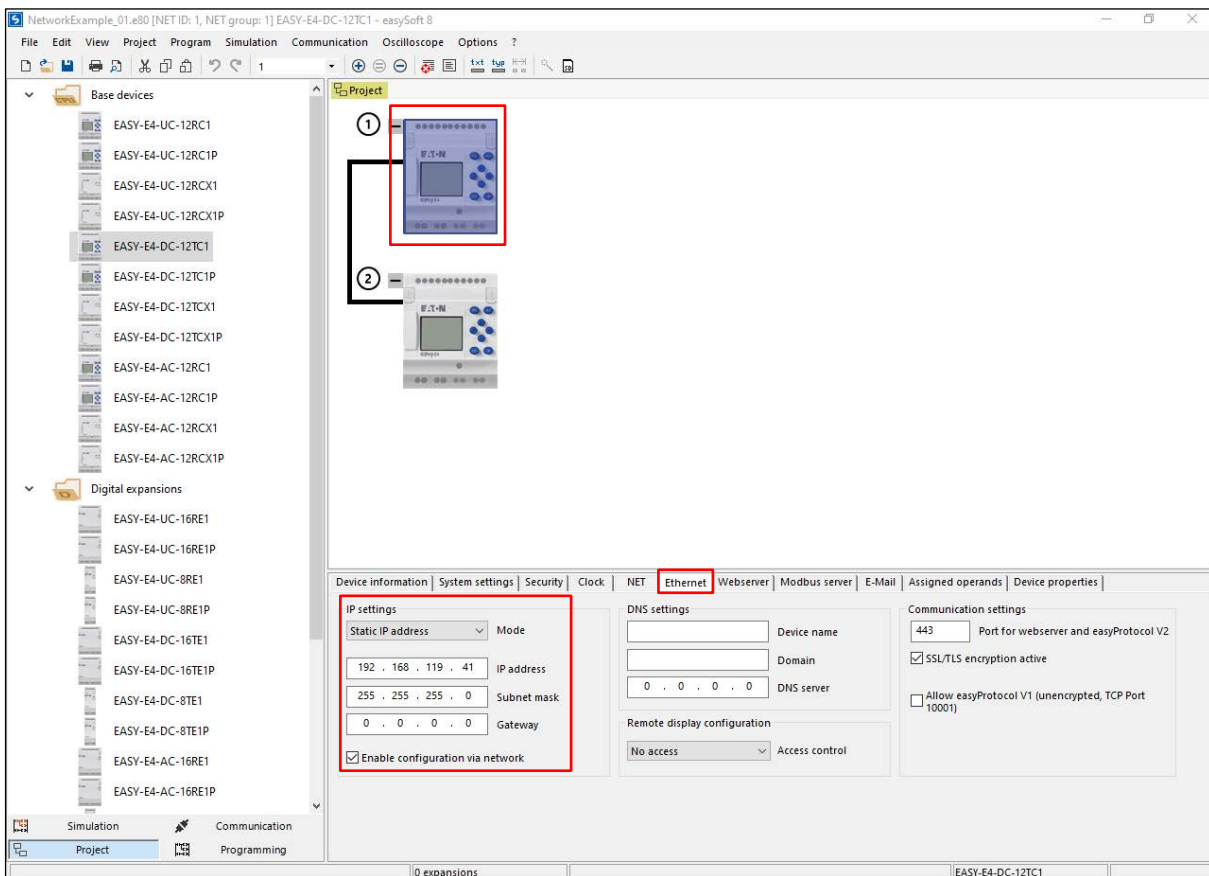


Fig. 138

- Select the first easyE4 and select the "Ethernet" tab.

Enter the following at "IP settings":

"Static IP address".

IP address : 192.168.119.41

Subnet mask: 255.255.255.0

- Tick "Enable configuration via network".

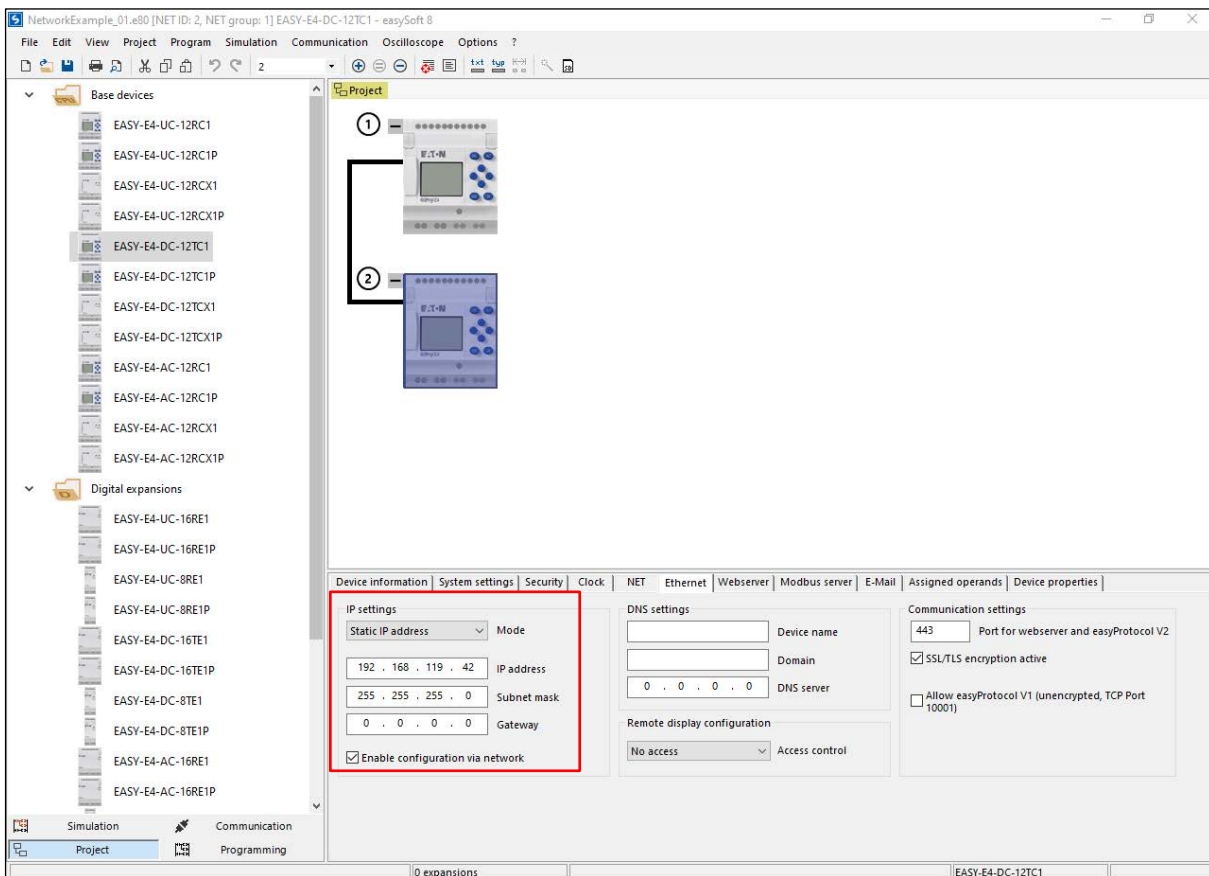


Fig. 139

Repeat this for the second easyE4. Enter the following at "IP settings":

Enter the following at "IP settings":

“Static IP address”.

IP address : 192.168.119.42

Subnet mask: 255.255.255.0

- Tick “Enable configuration via network”.

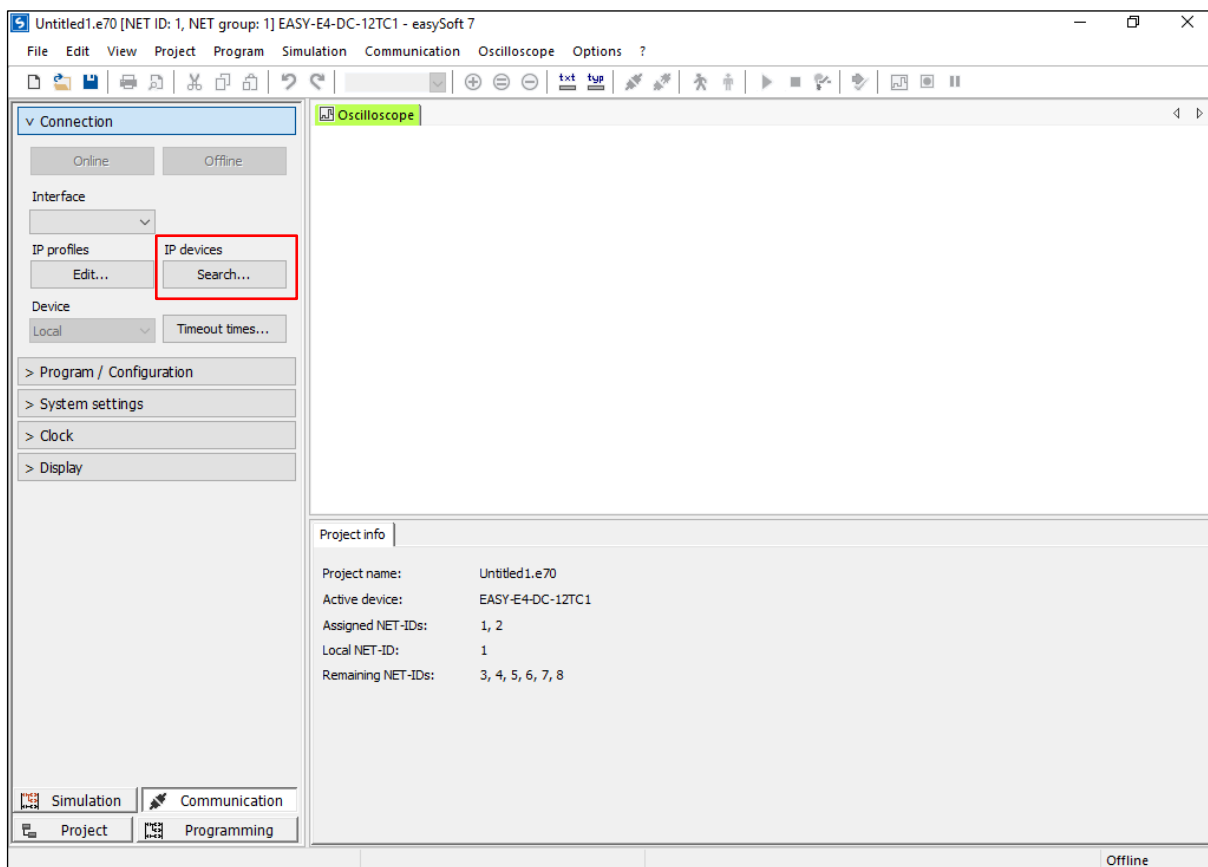


Fig. 140

- ▶ Go to the "Communication" menu.
- ▶ Go to "IP devices" → "Search".

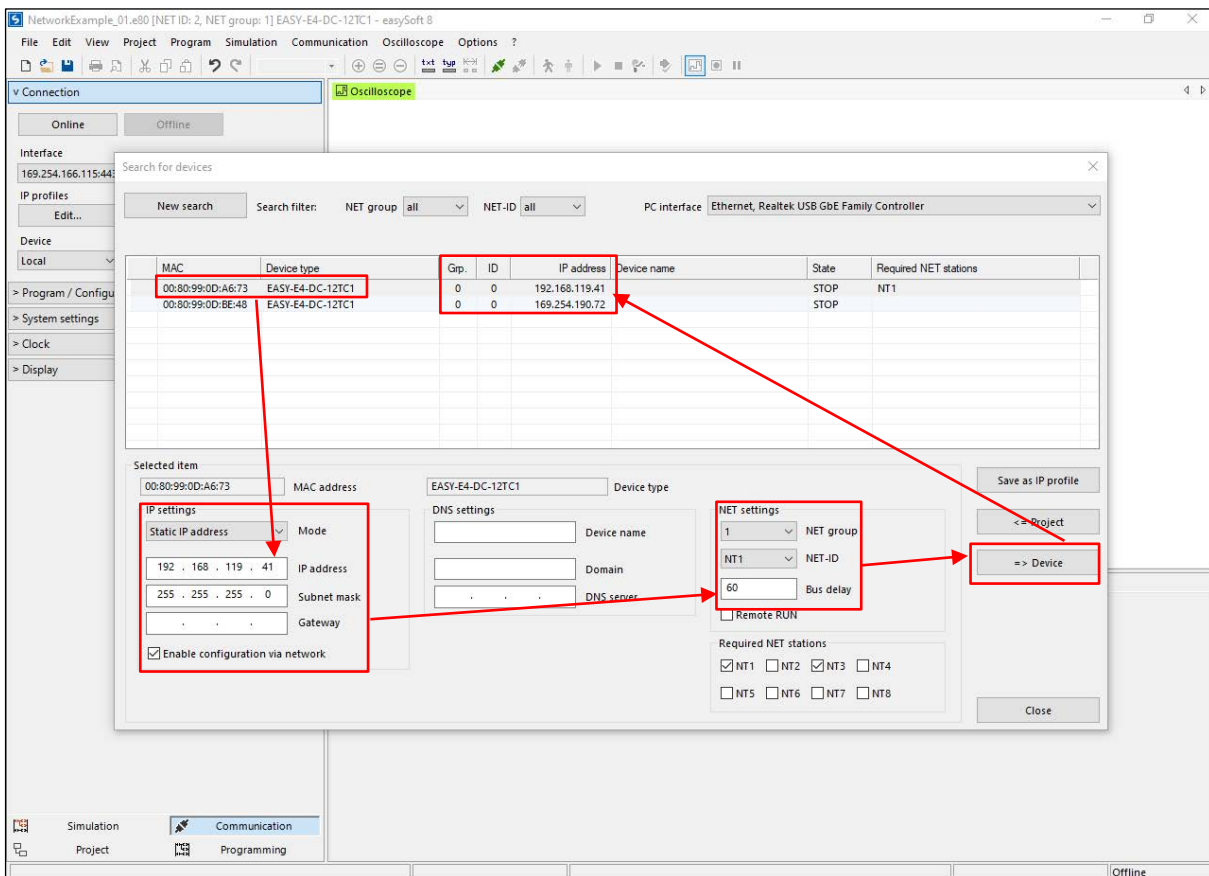


Fig. 141

- Here, click on "New search".

The software will now search for the connected easyE4s.

If the easyE4s need to communicate with each other, they must be within the same IP addressing scheme and subnet mask.

- Select the easyE4 that should receive the following IP address and subnet mask:

IP address: 192.168.119.41  
Subnet mask: 255.255.255.0

- Enter this data under "IP settings":

Mode: Static IP address  
IP address: 192.168.119.41  
Subnet mask: 255.255.255.0

Net settings:

NET group : 1

Net ID: NT1 (The NET-ID: 1 indicates that the network sees this easyE4 as the first easyE4.)

- Click on "=> Device" to write the network configuration to the respective easyE4.

The adjusted IP address can be seen on the easyE4 screen. The new IP address is also shown in the "IP address" column.

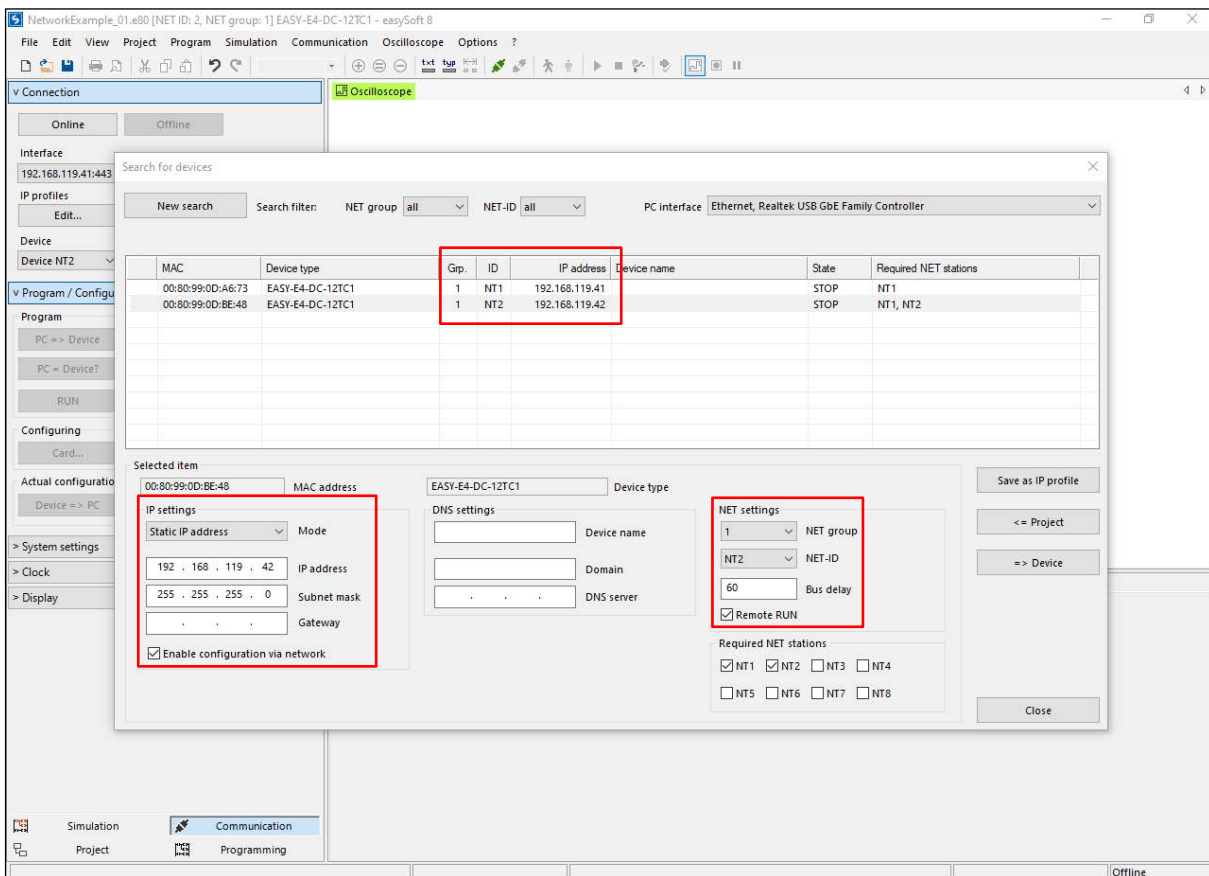


Fig. 142

- Repeat this for the second easyE4, but then with:

Mode: Static IP address

IP address: 192.168.119.42

Subnet mask: 255.255.255.0

- Tick: "Enable configuration via network"

Net settings:

NET group : 1

Net ID: NT2 (The NET-ID: 2 indicates that the network sees this easyE4 as the second easyE4.)

- Tick "RemoteRUN".
- Click on "=> Device" to write the network configuration to the respective easyE4.

The new addresses are displayed directly in the "IP address" column.

- Close the window with "Close".

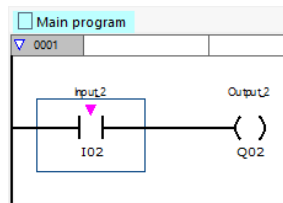
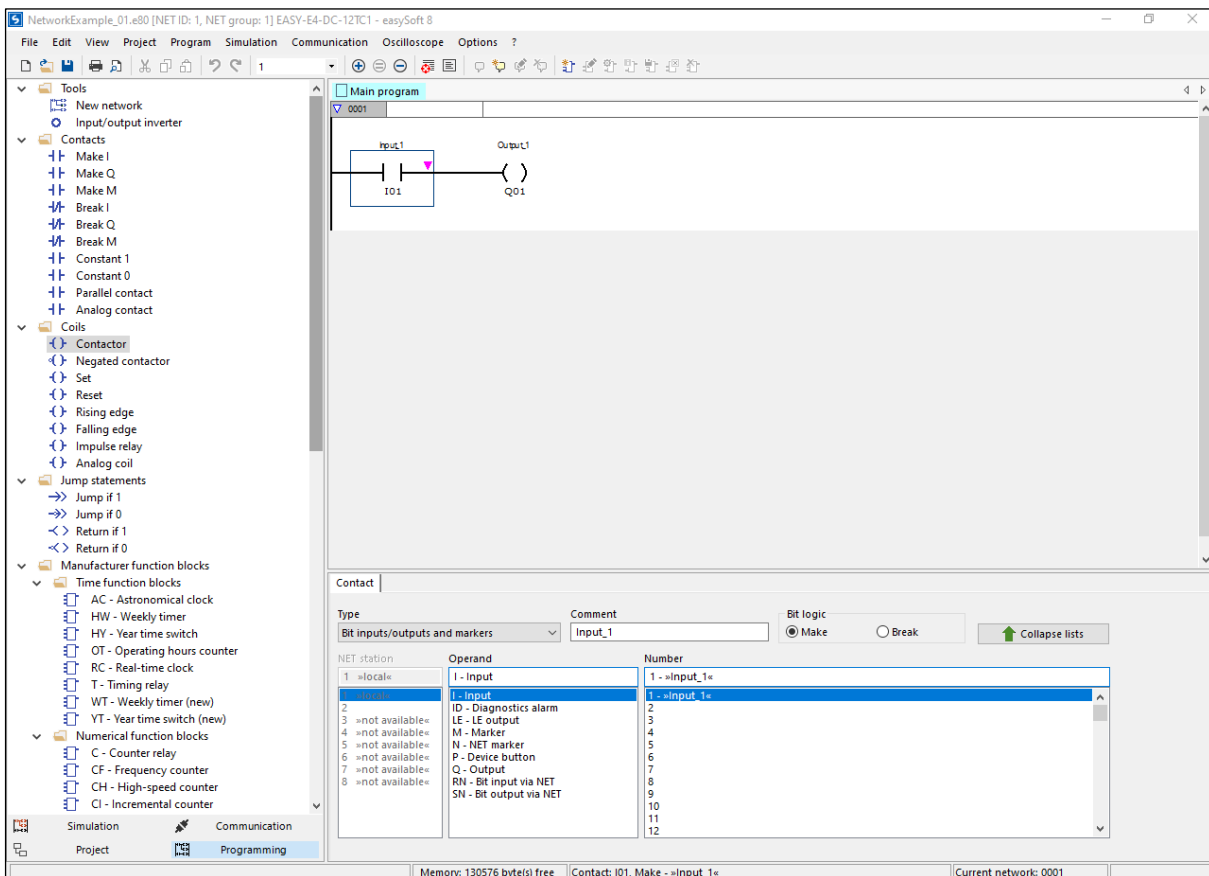


Fig. 143

- Go to the "Project" menu and select the first easyE4.
- Go to "Programming". And select the programming language "Ladder".

Write the program here that controls "Input\_1" (I1) "Output\_1" (Q1).

Repeat this for the program in the second easyE4, but with "Input\_2" (I2) and "Output\_2" (Q2).

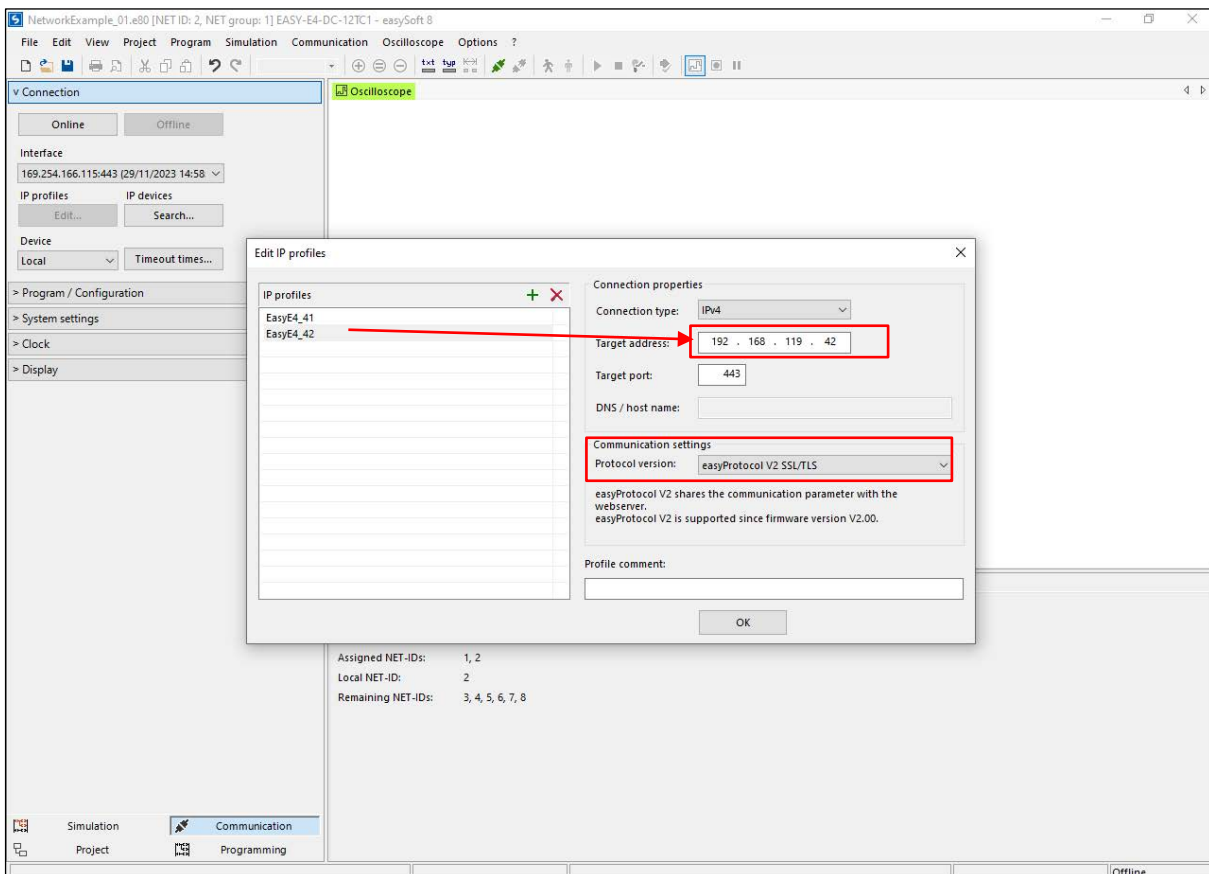


Fig. 144

- Go to the "Communication" menu.
- Select "Edit" under "IP profiles". Create two new profiles.

#### First easyE4:

- IP profiles: easyE4\_41
- Target address: 192.168.119.41
- Protocol version: easyProtocol V2 SSI/TLS

#### Second easyE4:

- IP profiles: easyE4\_42
- Target address: 192.168.119.42
- Protocol version: easyProtocol V2 SSI/TLS

- Confirm this with "OK".

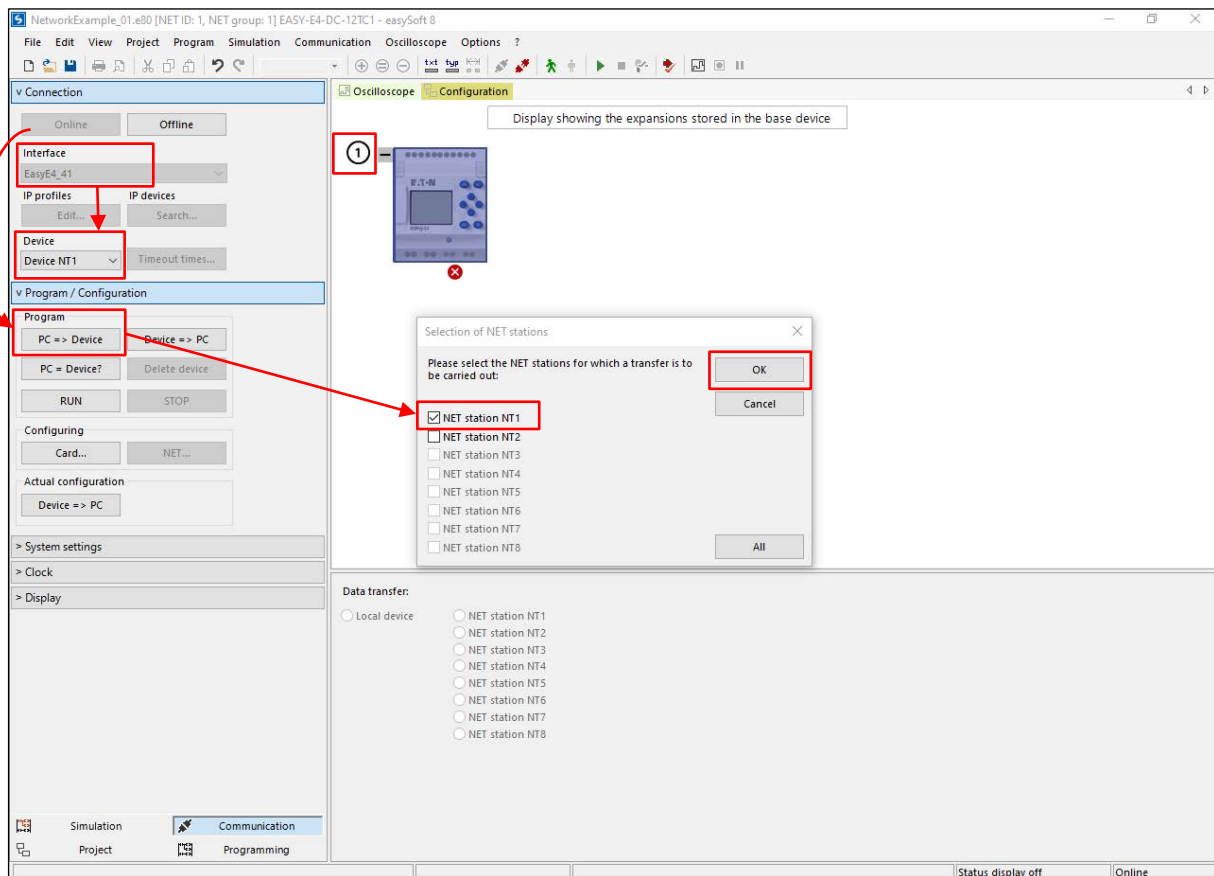


Fig. 145

Now the programs must be transferred to the easyE4s.

- ▶ At "Interface", select the profile "EasyE4\_41".
- ▶ At "Device", choose "Device NT 1".
- ▶ Click "Online".
- ▶ Click "PC => Device" under "Program".
- ▶ In the "Selection of NET stations" window, "NET station NT1" is checked.
- ▶ Click on "OK" and the program is transferred to the first easyE4.
- ▶ Set the easyE4 back to RUN.

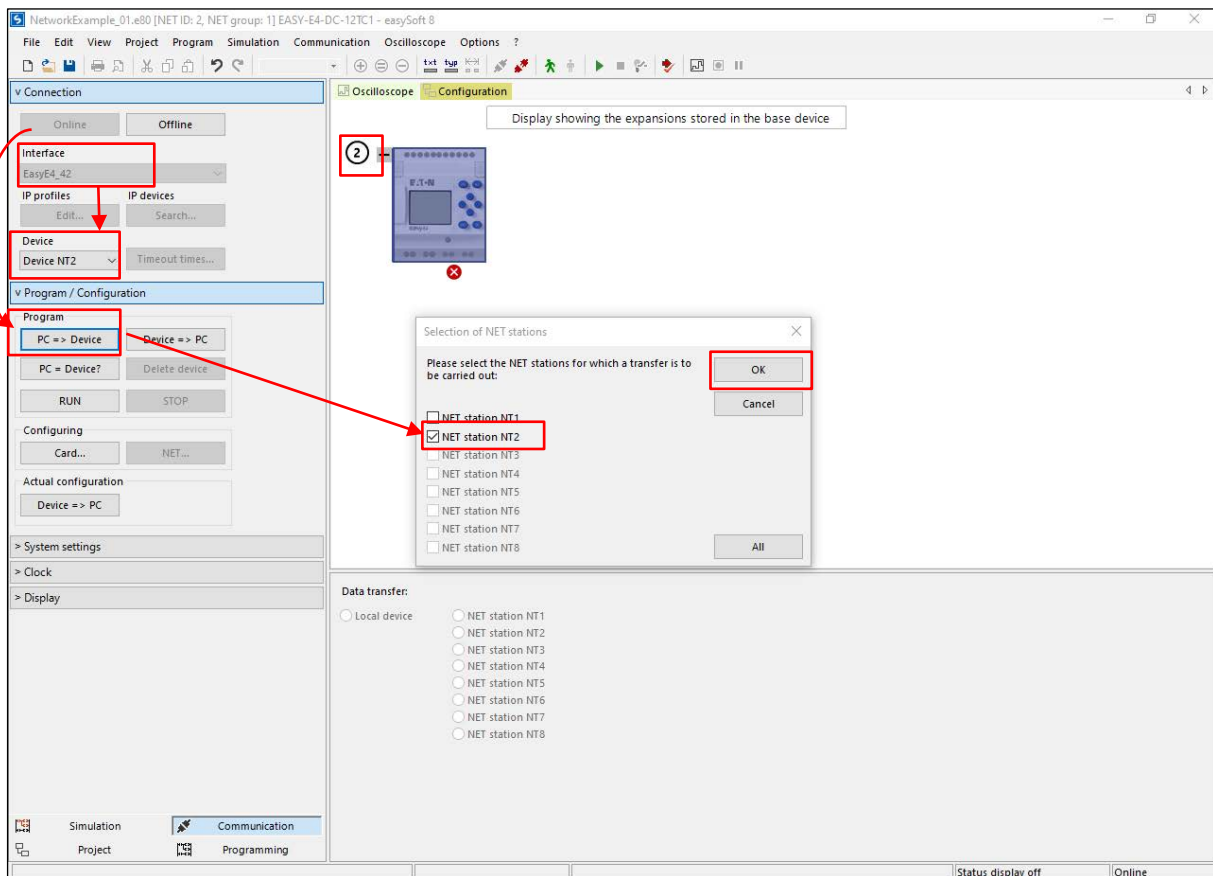


Fig. 146

The program for the second easyE4 must be transferred.

- ▶ At "Interface", select the profile "EasyE4\_21".
- ▶ At "Device", choose "Device NT 2".
- ▶ Click "Online".
- ▶ Click "PC => Device" under "Program".
- ▶ In the "Selection of NET stations" window, "NET station NT 2" is checked.
- ▶ Click on "OK" and the program is transferred to the second easyE4.
- ▶ Set the easyE4 back to RUN.
- ▶ Test the functioning of both easy's.

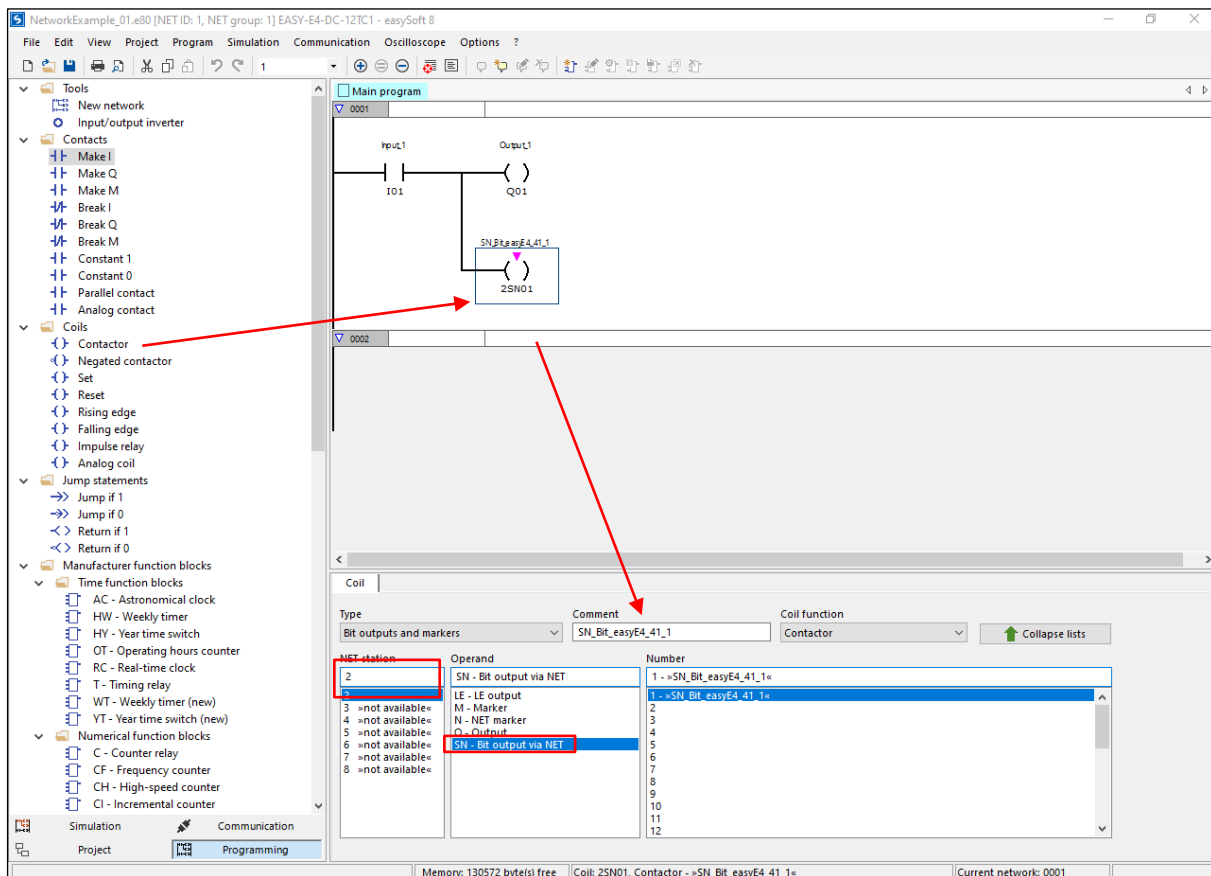


Fig. 147

It is always possible to send a bit of the one easyE4 to the other. The program is adjusted in the first easyE4.

"Input\_1" controls a "SN - Bit output via NET".

- ▶ Add a "Contractor" parallel to Q01.
- ▶ Select an "SN – Bit output via NET" for "Operand" and name it "SN\_Bit\_easyE4\_41\_1".

This bit is set on the network.

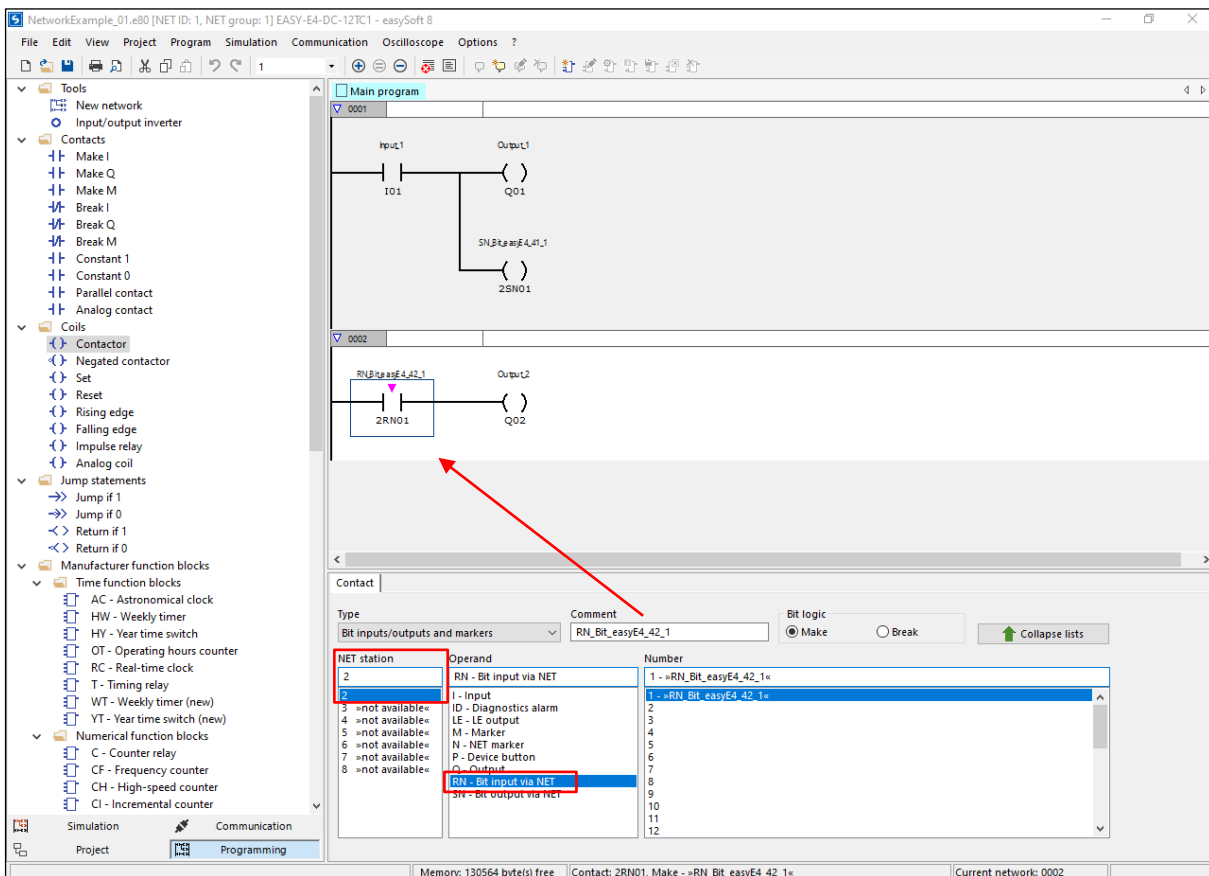


Fig. 148

A bit must be received from participant 2.

- Create a new rung.
- Drag a "Make" contact to the worksheet and change the operand in "RN - Bit input via NET".

Name this "RN\_Bit\_easyE4\_42\_1".

This bit is set on the network by the second easyE4.

"RN\_Bit\_easyE4\_42\_1" controls "Output\_2" (Q2).

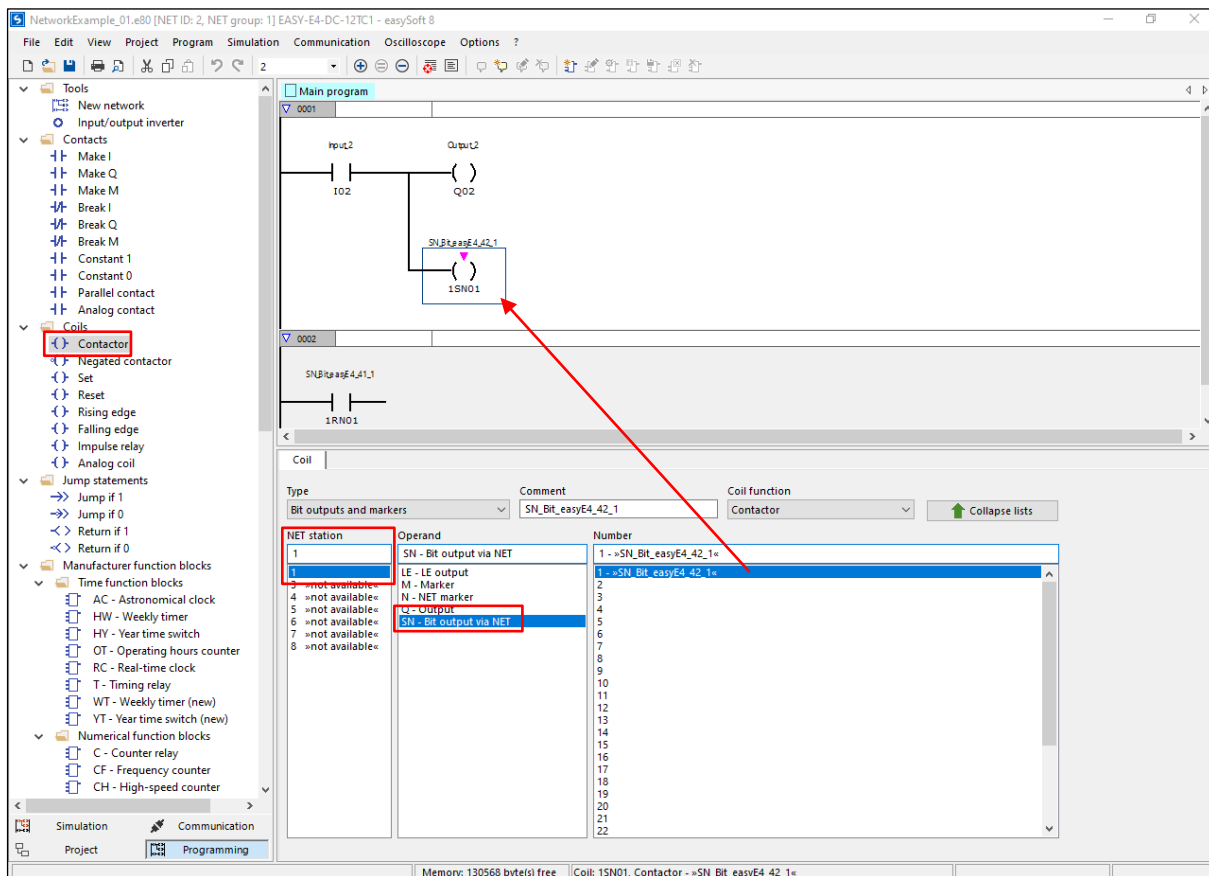


Fig. 149

The program is also adjusted in the second easyE4. A bit must be sent to the first easy.

"Input\_2" sets a "SN - Bit output via NET" on the network.

- ▶ Place a "Contactor" parallel to "Output\_2" (Q2) and change the operand to "SN – Bit output via NET".
- ▶ Name this "SN\_Bit\_easyE4\_42\_1". This is sent to "NET station" = 1.

A bit must be received from participant 1.

Create a new rung.

- ▶ Drag a "Make" contact to the worksheet and change the operand in "RN - Bit input via NET".
- ▶ Select "Net station" 1.
- ▶ Choose an "RN – Bit input via NET" here and then select the variable "SN\_Bit\_easyE4\_41\_1".

This bit comes from "NET station" = 1 and controls "Output\_1" (Q1) of the second easyE4.

- ▶ Send the adjusted programs to the easyE4s and test the function.

"Input\_1" of the first easyE4 now controls not only the "Output\_1" of the first easyE4 but also "Output\_1" (Q1) of the second easy.

Conversely, "Input\_2" (I2) of the second easyE4 controls not only "Output\_2" (Q2) of the second easyE4 but also "Output\_2" (Q2) of the first easy.

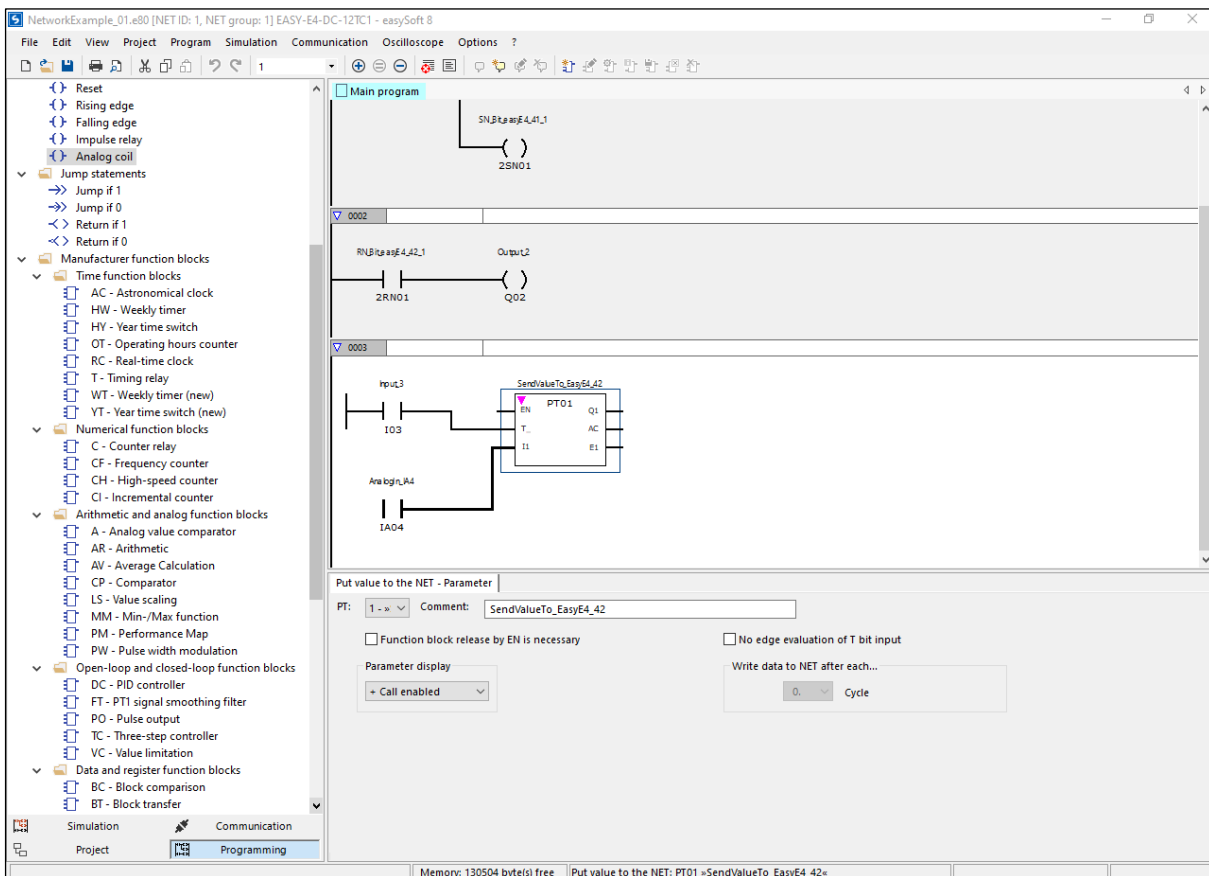


Fig. 150

With the function blocks (FB) “PT – Put value to the NET” and “GT – Get value from the NET,” it is possible to place bytes, words, and double words on the network. These values can then be retrieved by other easyE4s.

- Open the program of the first easyE4.

Create a new rung and drag the FB “PT - Put value to the NET” onto the worksheet and name it “SendValueTo\_EasyE4\_42”.

With “Input\_3” (I3), the value of the fourth analog input “AnalogIn\_IA4” (IA4) is placed on the network with every pulse.

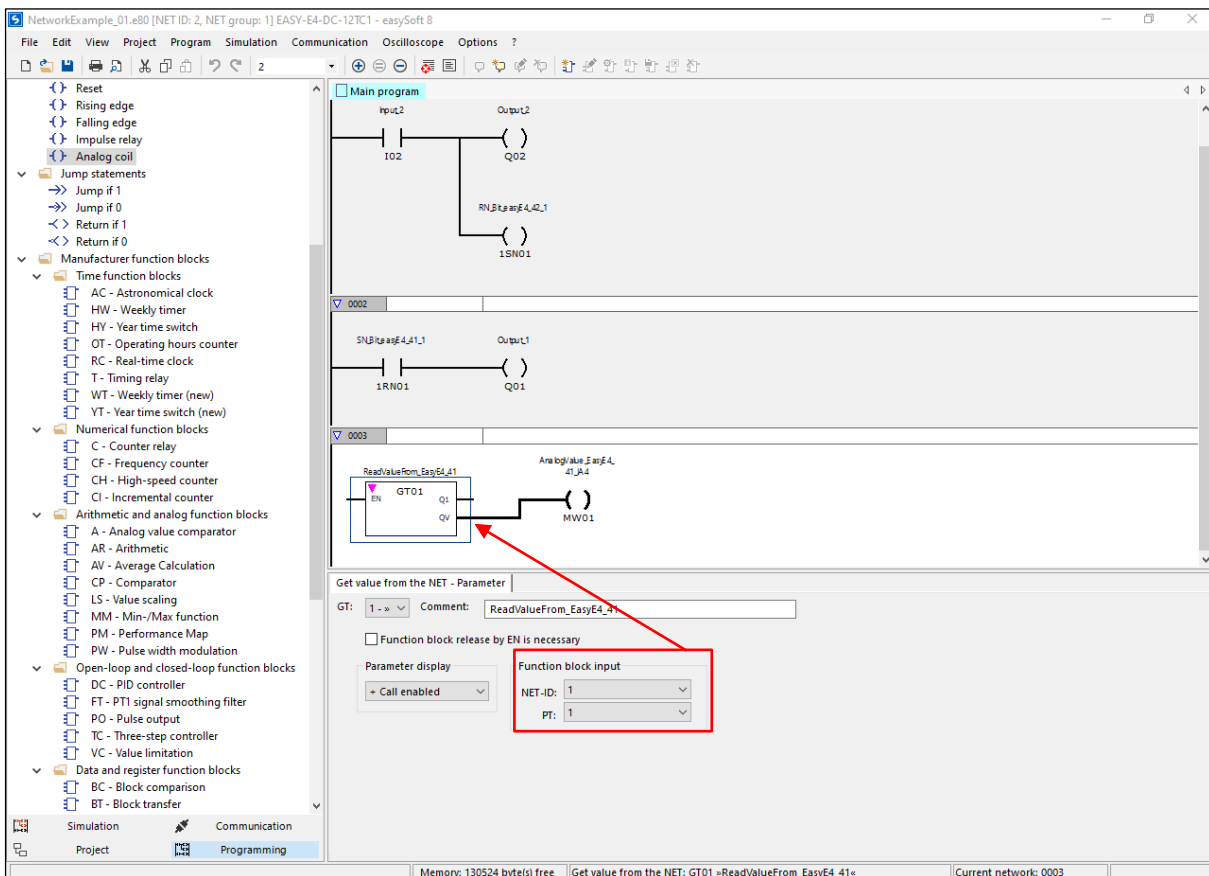


Fig. 151

- Open the program of the second easyE4.

Create a new rung and drag the FB "P=GT - Get value from the NET" onto the worksheet and name it "ReadValueFrom\_EasyE4\_41".

The FB retrieves the value from the easyE4 with NET-ID = 1 and from the first "PT – Put value to the NET".

The retrieved value is placed in "AnalogValue\_EasyE4\_41\_IA4" (MW1).

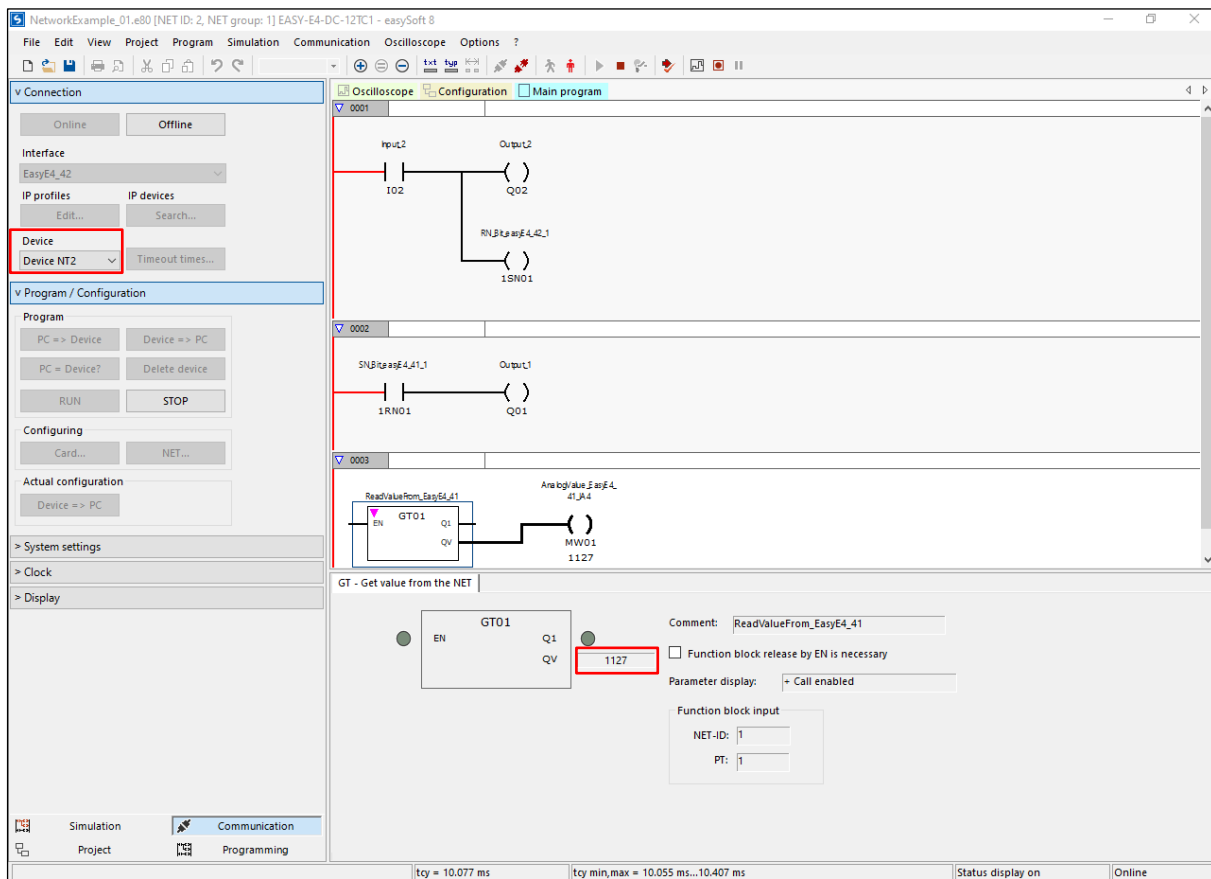


Fig. 152

- Upload the programs to the easyE4s and test the function.

Connect with the second easyE4. As seen above, the value is transferred from the first easyE4 to the second easyE4. The value is placed in "AnalogValue\_EasyE4\_41\_IA4" (MW1).

## 3.4 Default settings

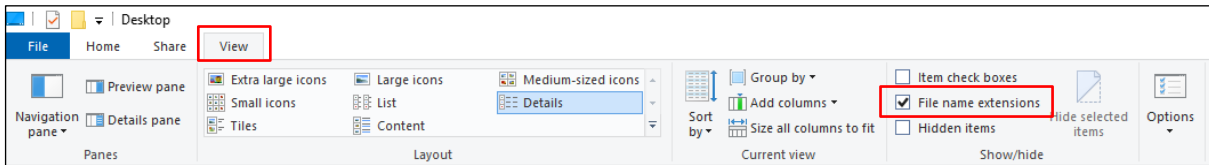


Fig. 153

It is possible to reset the easyE4 to the default settings.

Requirements:

- Micro SD Card, 128 MB to 32 GB (SD, SDHC, FAT12/16/32, Class 2 or 4).
- Card reader with SD-slot
- EasyE4

► Go to Windows explorer.

► Select the "View" tab and check "File name extension" if necessary.

1. Open a text editor. For example "Notepad" and create a file named "RESET\_MY\_E4.BIN". The file may not contain any text.
2. Copy the file to the MicroSD card.

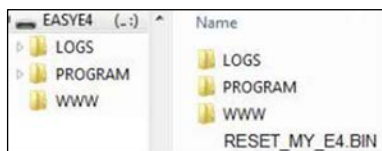


Fig. 154

3. Disconnect the easyE4.
4. Place the micro SD card in the easyE4.
5. Connect the easyE4.
6. Wait a few seconds.
7. Disconnect the easyE4.
8. Remove the micro SD card.
9. Connect the easyE4.

The easyE4 has now been reset to factory settings. The program, password and all other settings have been deleted. The ethernet port on the easyE4 is once again set on auto-IP.

## Notes



*Powering Business Worldwide*