



Communication SBOX-Q to CECC by telnet using CamaraControl.library

Example: Program selection by digital I/O and telnet

SBOC-Q-R3B-WB
(Tnr.:555841)

TitleCommunication SBOX-Q to CECC
Version 1.00
Document no. 12356
Original de
Author Bernd Schmid CM-SP

Last update 28.02.2018

Copyright Notice

This documentation is the intellectual property of Festo AG & Co. KG, which also has the exclusive copyright. Any modification of the content, duplication or reprinting of this documentation as well as distribution to third parties can only be made with the express consent of Festo AG & Co. KG.

Festo AG & Co KG reserves the right to make modifications to this document in whole or in part. All brand and product names are trademarks or registered trademarks of their respective owners.

Legal Notice

Hardware, software, operating systems and drivers may only be used for the applications described and only in conjunction with components recommended by Festo AG & Co. KG.

Festo AG & Co. KG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Defects resulting from the improper handling of devices and modules are excluded from the warranty.

The data and information specified in this document should not be used for the implementation of safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect. In other respects, the regulations with regard to liability from the terms and conditions of delivery, payment and use of software of Festo AG & Co. KG, which can be found at www.festo.com and can be supplied on request, shall apply.

All data contained in this document do not represent guaranteed specifications, particularly with regard to functionality, condition or quality, in the legal sense.

The information in this document serves only as basic information for the implementation of a specific, hypothetical application and is in no way intended as a substitute for the operating instructions of the respective manufacturers and the design and testing of the respective application by the user.

The operating instructions for Festo products can be found at www.festo.com.

Users of this document (application note) must verify that all functions described here also work correctly in the application. By reading this document and adhering to the specifications contained therein, users are also solely responsible for their own application.

Content

1	Hardware/Software	4
1.1	Manuals..... Fehler! Textmarke nicht definiert.	
	SBO...-Q: 548318; 548319.....	4
2	Description of the example.....	Fehler! Textmarke nicht definiert.
3	Configuration of SBO camera in CheckKon.....	7
4	Pinassignment CECC.....	Fehler! Textmarke nicht definiert.
4.1	Decription of Working sequence.....	Fehler! Textmarke nicht definiert.
5	Codesys.....	Fehler! Textmarke nicht definiert.
5.1	Prepare the demo file	11
5.2	Comissioning	14
5.3	Working with Visulisation	15
5.4	Test	16
6	Detailed description of the Code	19

1 Hardware/Software

Typ/Name	Version Software/Firmware	Herstellungsdatum
SBOC-Q-R3B-WB	3.5.3.58	
CECC-D TN 574415	Rev.03	
Codesys 3.5	SP 4 Patch 1	
Checkkon	4.3 rel 06	

Table 1.1: Hardware/Software

1.1 Available manuals

SBOC-Q-R3B-WB manual from Support Portal www.festo.com/sp

DE: 548318

EN: 548319

CECC › Support Portal

2 Description of the example

The intension of the example is to show, how to establish communication between a Codesys Controller CECC and SBOC-Q camera. The CECC will change the active inspection program on the SBO camera by telnet TCP/IP protocol. The communication is based on the CameraControl.library.

The example show the use of different function block out of the CC.library.

In addition this example could be taken to substitute an installation, where an I/O CP module at CAN bus on SBO camera, do an program change by I/O communication. (see fig. 2)

Due to the fact, that CAN bus is not available on SBOC-Q-R3B-WB. The CECC work as a gateway to translate the binary I/O signal of programno. to a telnet command with similar information.

So this is one possibility for a workaround to substitute the phase out camera model SBOC-Q-R1B with CAN bus.

Fig. 1 Installation topology: 1: ETH Hub, 2: PC, 3: SBOC-Q, 4: CECC

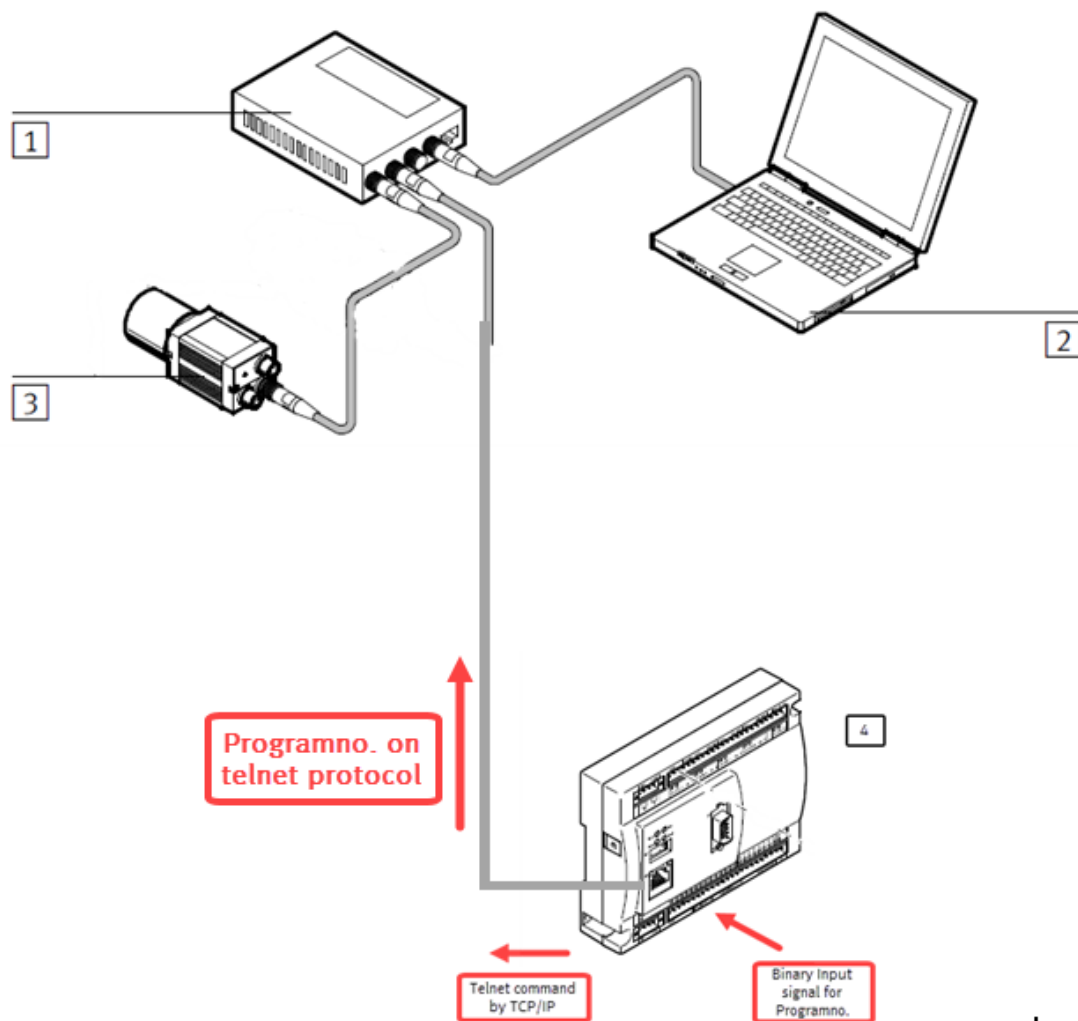
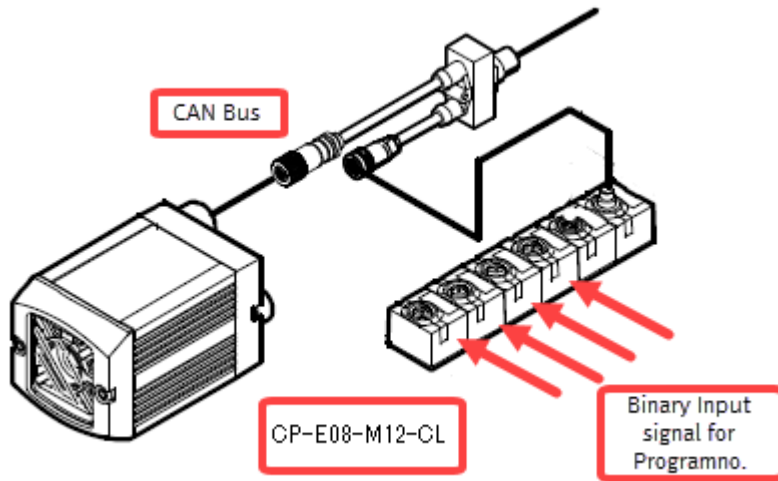


Fig.2 CAN bus installation for program selection by 24V signals



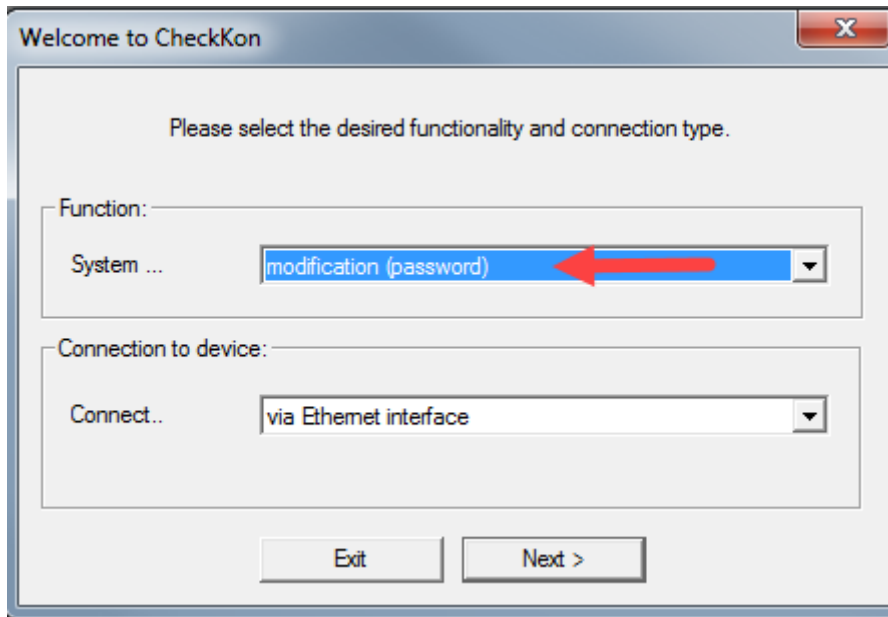
Note: This configuration with CAN bus is substituted by the Codesys example and program selection via CECC.

3 Configuration of SBO camera with software Checkkon

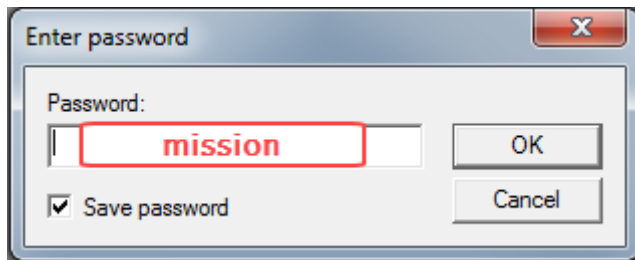
The camera is configured with the software „Checkkon“ The software is available on the support portal. Before you connect to Checkkon you should check you ETH adapter setting on your PC and the camera. The demo work with fix IP addresses for the particular components. So adapt your PC setting to a fix ip-address in the same subnet as the camera. The camera has on default an address 192.168.2.10 255.255.255.0

The IP-setting of the camera could be changed by the software FFT Festo Field Device Tool. > available on Support Portal.

Start Checkkon and establish the connection to camera. Select the access level: modification password

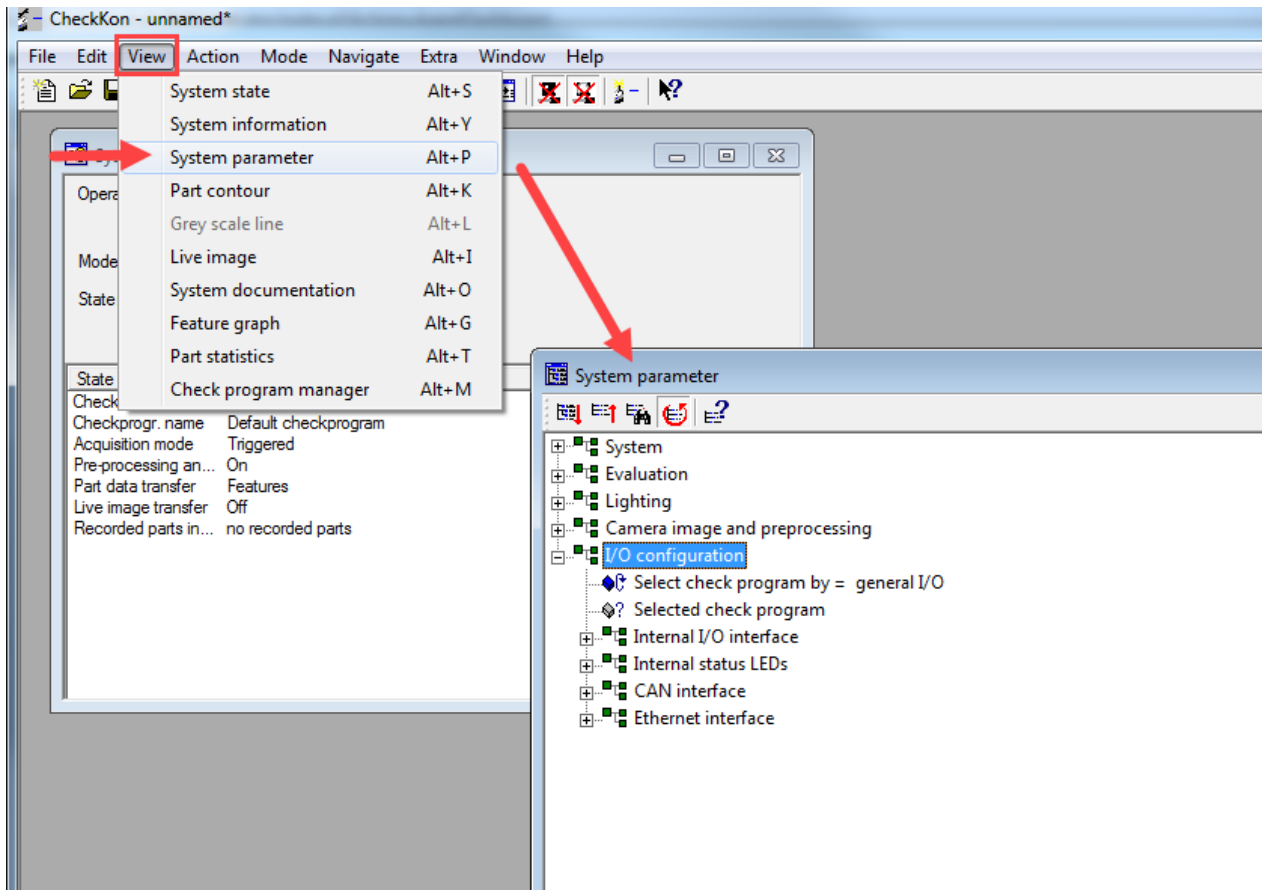


Enter the password:

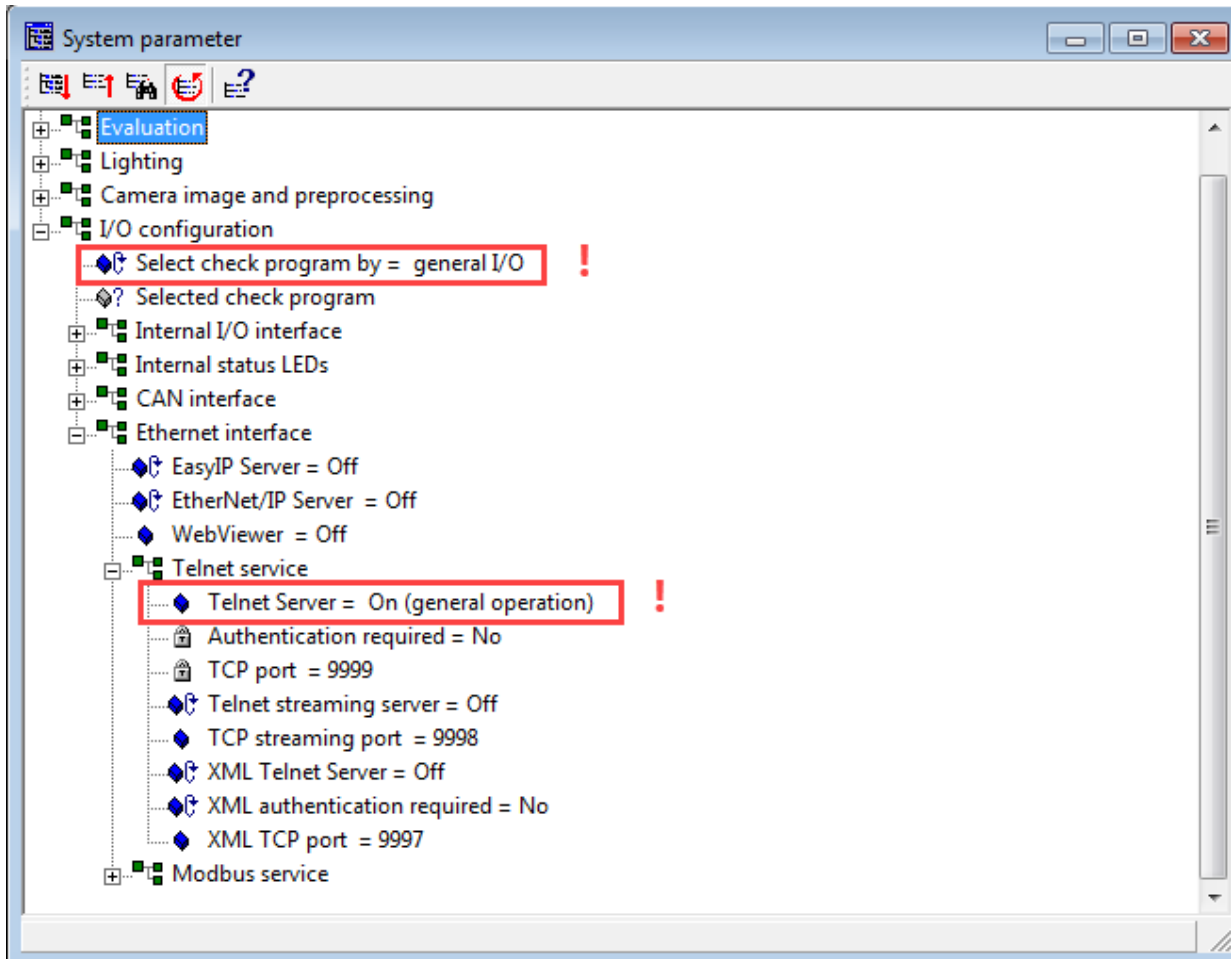


Open the window: „System parameter“

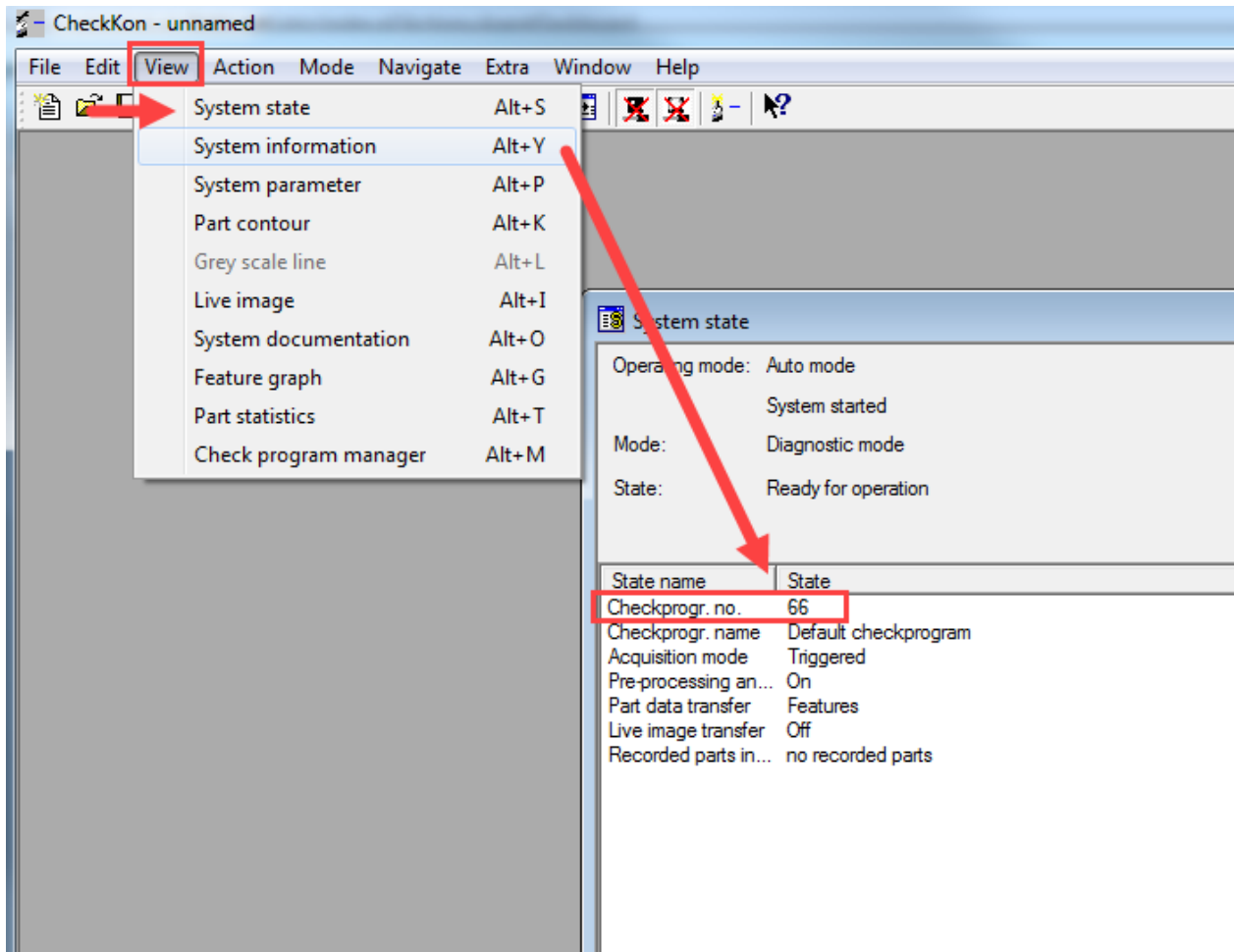
Configuration of SBO camera with software Checkkon



Open the I/O configuration and do following important setting:



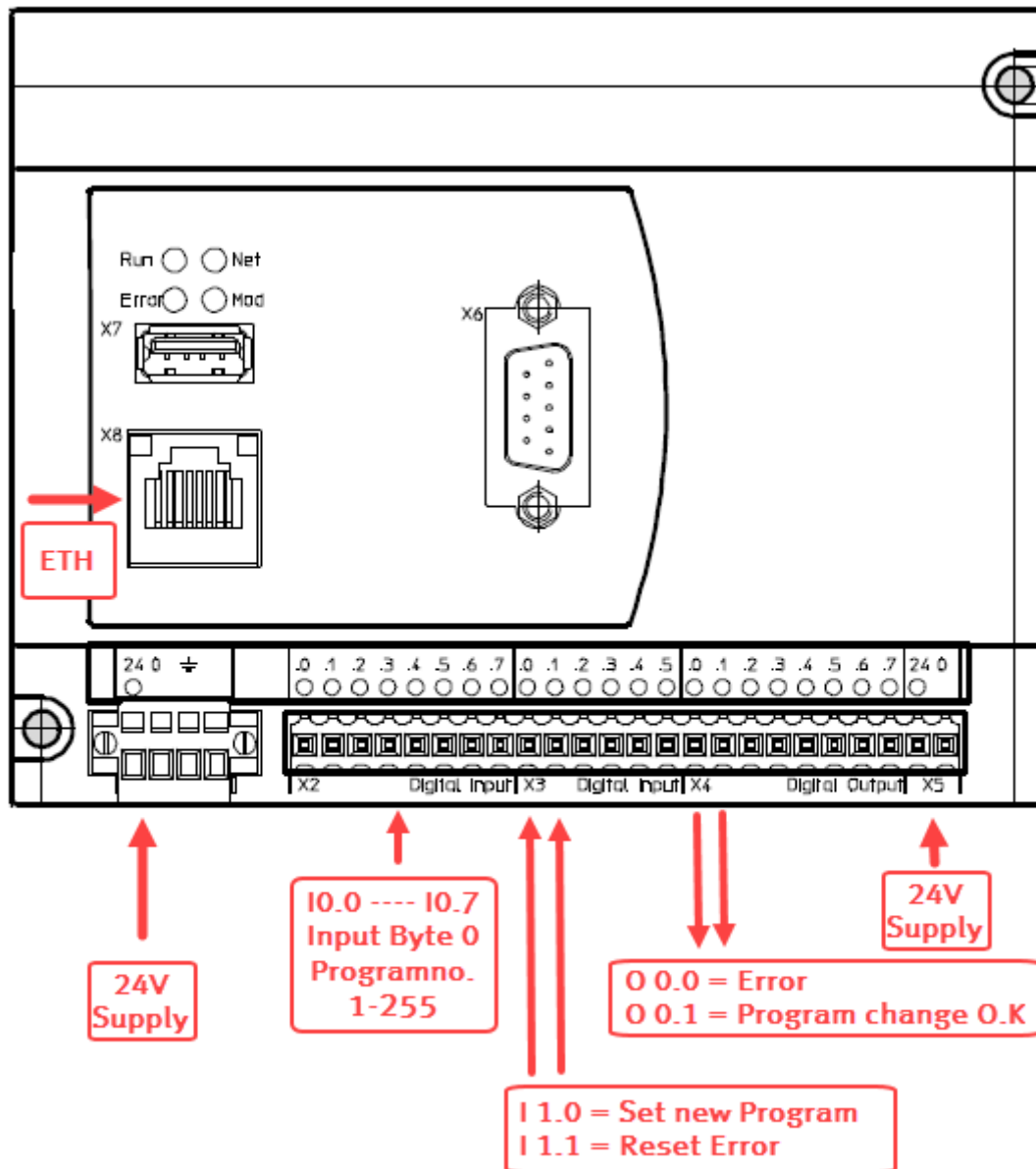
The active inspection program is shown in the window: „System state“



Note: Checkkon and CECC can be connected simultaneous to the camera. So it is possible to see the current programno. immediately.

4 CECC

4.1 Pin assignment for the demo use



4.2 Description and working sequence

24V supply for control and load voltage.

Input-byte 0 is the terminal where the binary coded programno. is applied. No signal (0V) keep the current program.

Any other bit combination will be transferred as new programno. to the camera, when a positive edge (24V) is applied to the Input 1.0.

The successful program change will be displayed as a High (24V) signal on the Output 0.0.

If the program switch failed, and the transfer was not successful the Output 0.1 will flash with a frequency of 0,5Hz.

The Error state could be reset by positive edge (24V) at Input 1.1. The Error output will be reset and the program wait for new Input signal.

4.3 ETH setting of CECC

The IP-address of the CECC must be adapted to the IP setting of camera and PC.

The IP-setting of the CECC could be changed by the software FFT Festo Field Device Tool. > available on Support Portal.

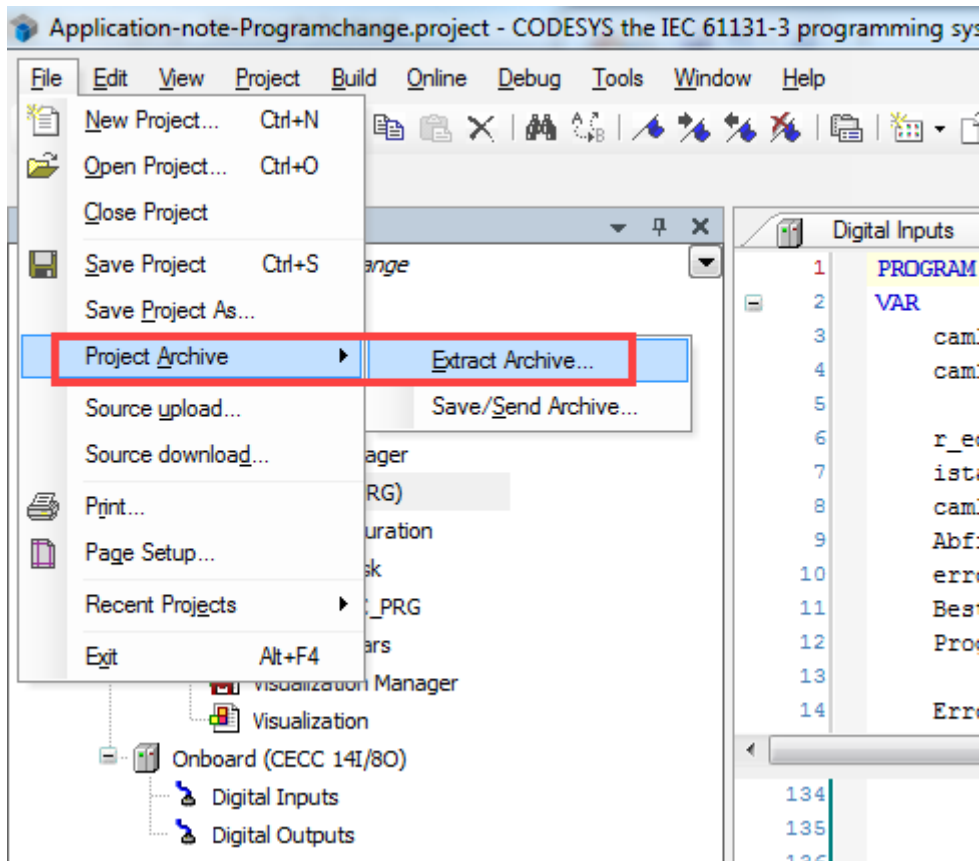
Example: 192.168.2.11 255.255.255.0

5 Codesys

5.1 Prepare the demo file

The demo file `Application-note-Programchange.projectarchive` is an project archive. That mean that all necessary library and system files were included.

So please extract the archive on your PC:



Then you find the hole program in the PLC_PRG and an Visualisation. The demo allows to work with the Visu and with the hw input and output of the CECC.

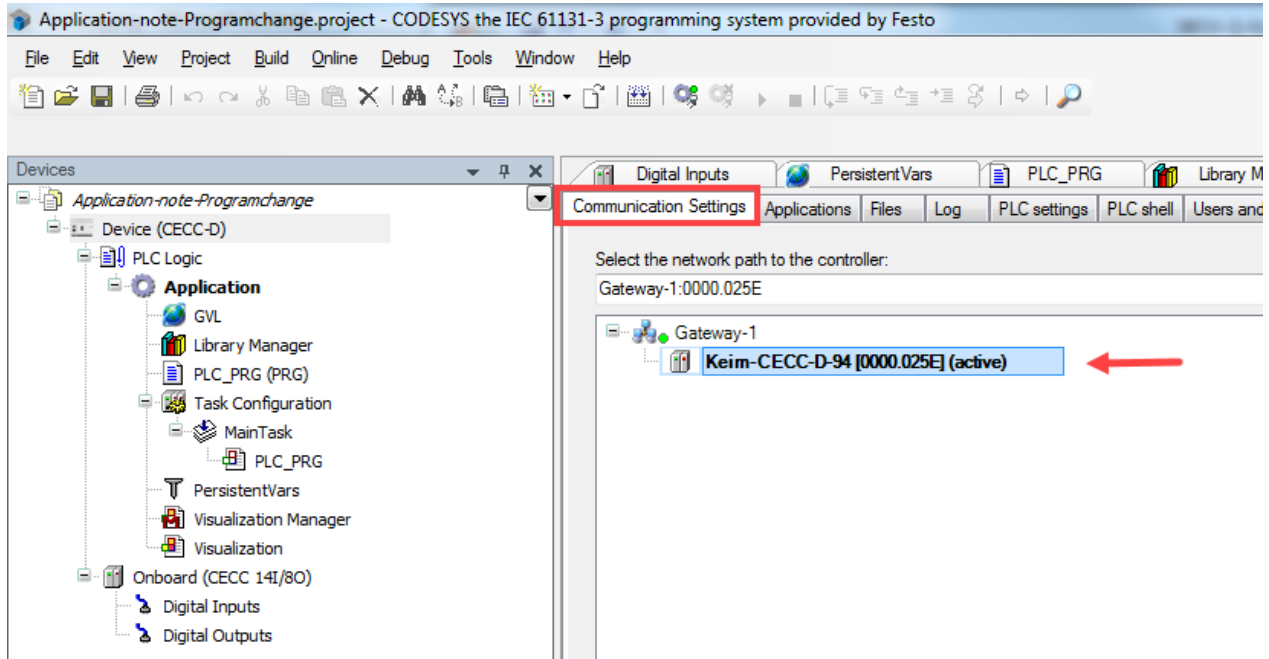
5.2 Comissioning

Wire all ETH connection as in fig.1 shown.

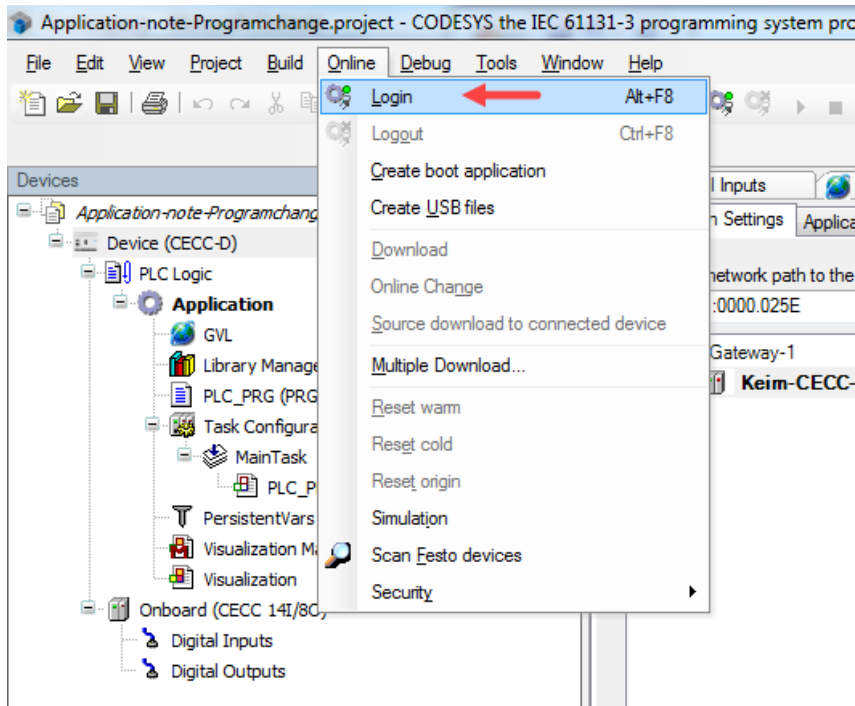
24V Power on for all devices.

Keep Checkkon connection and observe window. System State

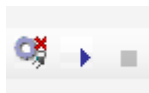
Setup connection to CECC



Set the connected controller to active device.



Login to CECC and start the controller:



Then the connection from CECC to camera should be established successfully.

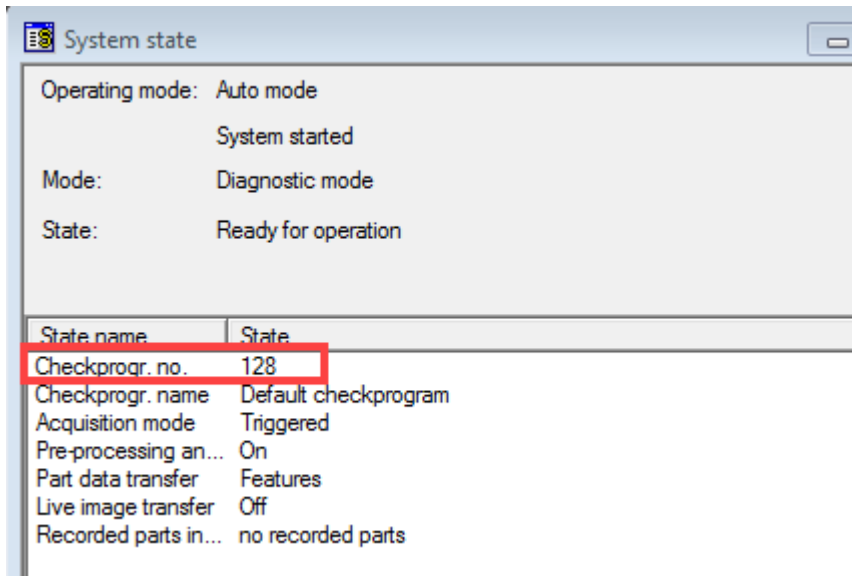
```
67 0: //Login to camera
68   camlogout.xExecute FALSE :=FALSE;
69   camlog.xExecute TRUE :=TRUE;
70
71   IF camlog.xDone TRUE THEN ←
72     istate 1 :=1;
73   ELSIF camlog.xError FALSE THEN
74     istate 1 :=99;
75
76   END_IF
77
78
79
80 1: // Wait for slope on Input or Visu to start program change process
81   IF r_edge.Q FALSE or visuI FALSE then
82     istate 1 :=2;
83     PchangeOK FALSE :=FALSE;
84
85   END_IF
86
```

Camera is connected and the state machine wait in step 1 for the input I1.0 to start program change.

Apply 24V input signal to Input byte 0.

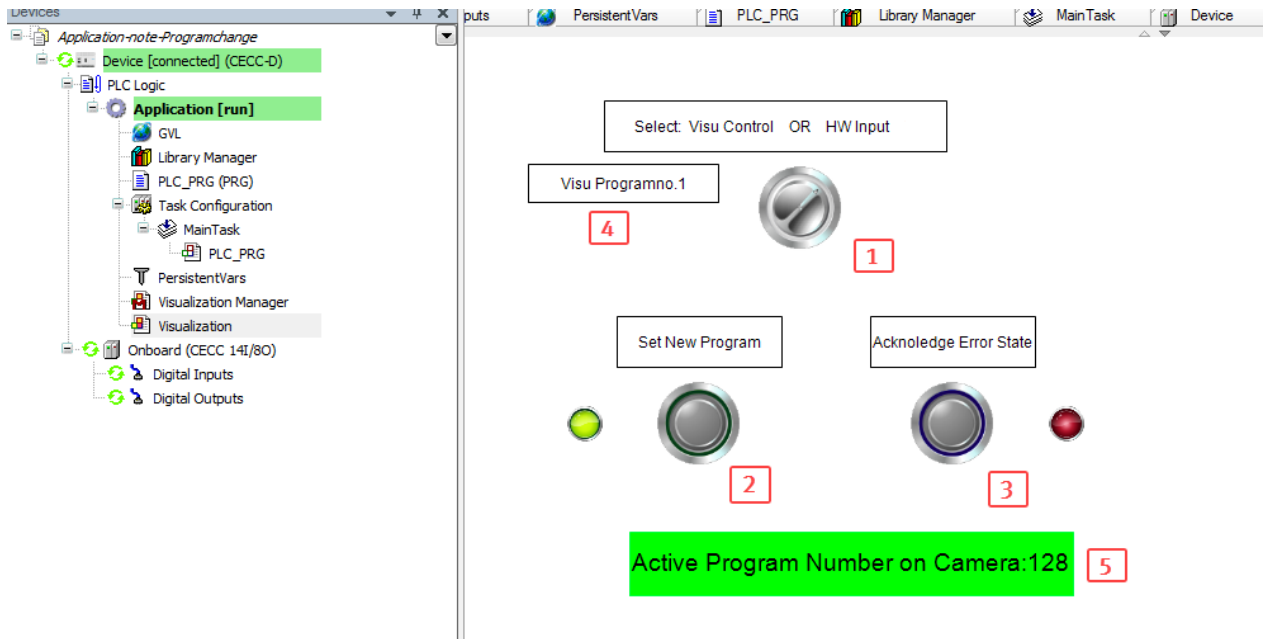
Then give 24V positive impulse to Input 1.0

Check if in Checkkon window System State the Checkprogramno. Has changed:



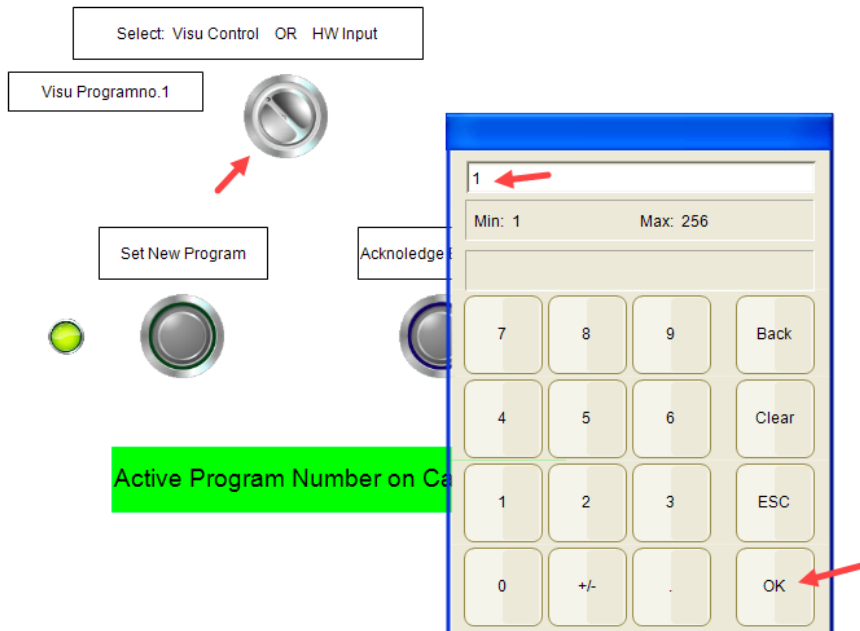
5.3 Working with Visualisation

For testing procedure it is possible to work with the integrated Codesys visualisation.



- 1: Selection switch between HW-Input control or Visu control.
- 2: Push button to start program change. Same function as Input 1.0
- 3: Reset Error state button.
- 4: Input box to select new programno. on mouse click > tic on field and a message box appears
- Green LED: lights on when transfer was successful
- Red LED: flashes when an Error occurred and transfer were not successful.
- 5: Display the active programno. from the camera

5.4 Test



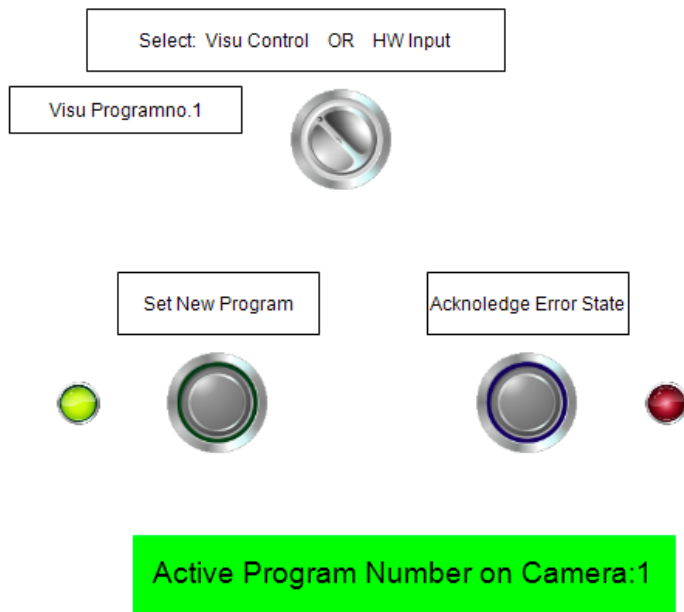
Select the Visu control with select button

Tic on Input Box, by mouse and edit the new programno. to message box

Confirm the no. by OK in the box.

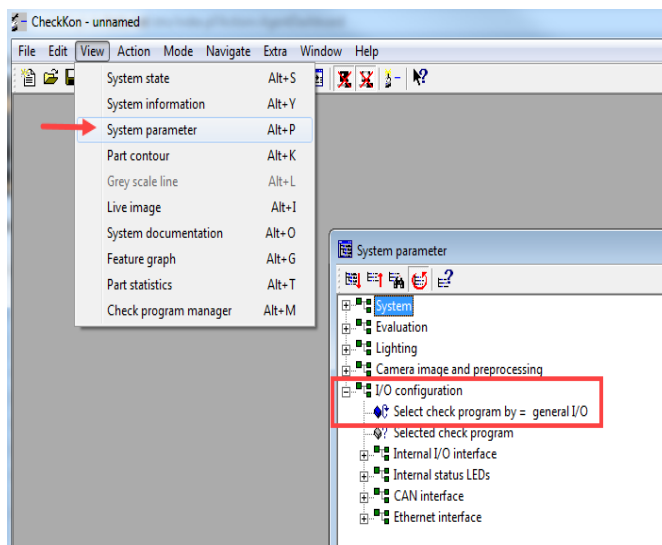
Tic on push button: Set New Program

- Green LED must highlight
- Active Program Number on Camera display must show the edited no.



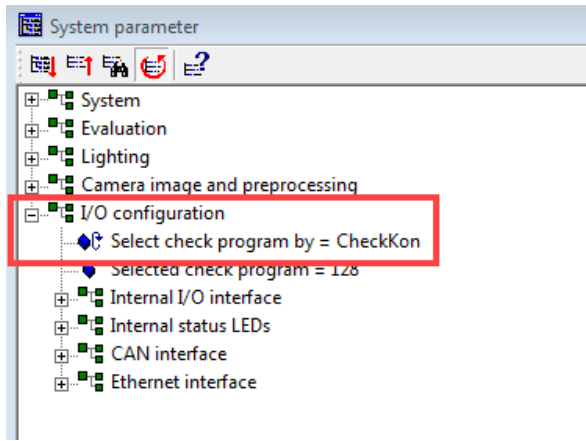
Test for Error

Change the following setting in Checkkon system parameter:

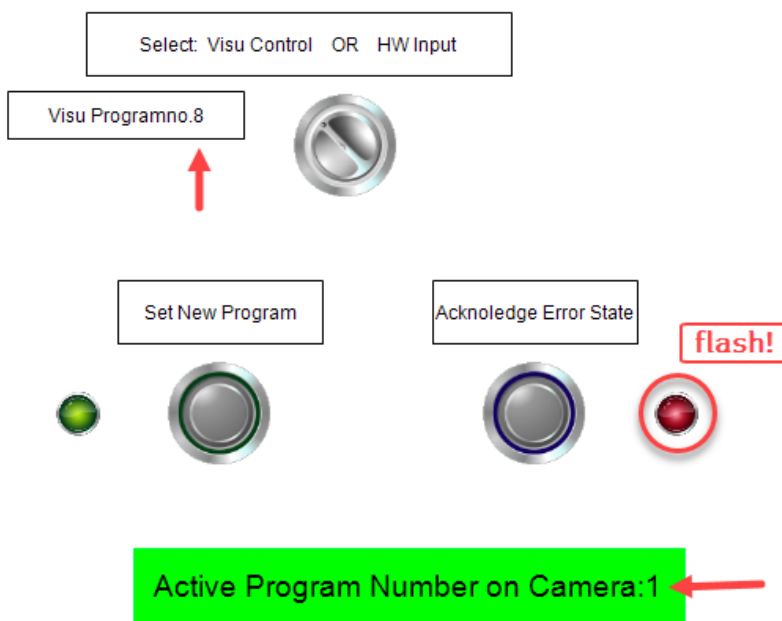


To this setting: Select check program by CheckKon.

With this setting the telnet command for changing the programno. on the camera is not accepted!



Then select a new programno. and tic on the Set New Program button.



Result:

The Red LED flash with 0,5 Hz.

The “old” program stay on the camera › see Display

Controller stay in this state until the button: Acknowledge Error State is pressed.

After that the Led is off and the controller wait for new tic on Set Program button.

Note! The Error will not occur when the preselection of the programno. is the same as the active program on the camera.

In this example that state will be simulated, when you preselect programno. 1 and transfer. The result is a green LED and the active programno. is 1.

6. Detail on Programming

Step1 is the “waiting” state. Here we wait for Input signal as described in chap. 5

Step 2: After the Input signal for program change is appeared the camera is checked if it is in ready state.

The ready state is given in flagword 16 on the camera. For this we use the fb to read flagword.

(camera control.library) If ready is 1 then we forward to the next step 3. If camera is not ready we wait and count the cycles. This counter evaluate if there were long processing times of the camera. Only for analysis demands.

```
2: // Check if camera is ready; Read fw 16
  visuI:=FALSE;
  Abfrage_bereit.uiFlagWord:=16;
  Abfrage_bereit.xExecute:=TRUE;

  IF Abfrage_bereit.xDone AND Abfrage_bereit.larResult=1 THEN
    ystate:=3;
    Abfrage_bereit.xExecute:=FALSE;
  ELSIF Abfrage_bereit.xDone AND Abfrage_bereit.larResult=0 THEN
    Abfrage_bereit.xExecute:=FALSE;
    errorcounter:=errorcounter+1;

  ELSIF Abfrage_bereit.xError THEN
    ystate:=99;
    Abfrage_bereit.xExecute:=FALSE;
  END_IF
```

Step 3: In Step 3 we archive the new programno. It will be set by Visu or Inputbyte of the CECC. The new no. is transferred to the camera by fb Change Program (camera control.library).

```
3: // Select which Programno. is used and start the change process
  IF visuPselect THEN
    Programmnr.uiCheckProgram:= visuPno;
  ELSE
    Programmnr.uiCheckProgram:= PIbyte;
  END_IF

  IF Programmnr.uiCheckProgram >0 THEN
    //Programmnr.uiCheckProgram:=PIbyte;
    Programmnr.xExecute:=TRUE;
  ELSE ystate:=1;

  END_IF

  IF Programmnr.xDone THEN

    ystate:=4;
  ELSIF Programmnr.xError THEN
    ystate:=99;
  END_IF
```

Step4: In Step 4 the output 0.1 for the successful download of the program is set to true. Then the next step to waiting state 1.

```
4: // set output for successful download
PchangeOK:=TRUE;
Programnr.xExecute:=FALSE;
istate:=1;
```

Step 99: Error Case. Every Error on any used fb will call the state 99. The blink sequence for the output 0.0 (Error LED) is started. Then the current programno. is read from the camera by fb read flagword 250. This will be displayed in the Visu. The program keeps in the state 99 until the acknowledge input 1.1 get an true signal. When the state 99 will be left, the telnet connection will be logged off. The next state is 0 with the log-on fb. This procedure will overcome failures on ETH disconnections.

```
99: // Errorcase; start blink and read current program from camera. Wait for acknowledge input
```

```
Errorblink05.ENABLE:=TRUE;
bBlinkON:= TRUE;
```

```
Abfrage_bereit.uiFlagWord:=250;
Abfrage_bereit.xExecute:=TRUE;
Programnr.uiCheckProgram:= LREAL_TO_UINT(Abfrage_bereit.larResult);
```

```
IF bLErrorAckn OR visuErrAkn //Error confirmation
THEN istate:=0;
Errorblink05.ENABLE:=FALSE;
Programnr.xExecute:=FALSE;
bBlinkON:= FALSE;
camlogout.xExecute:= TRUE;
camlog.xExecute:=FALSE;
Abfrage_bereit.xExecute:=false;
END_IF
```

```
END_CASE
```

End of document