Application Software for Industrial Vision Systems

NeuroCheck 6.1
Training Course
(Part I)

NEURO CHECK
Industrial Vision Systems

Trademarks and Imprint

NeuroCheck GmbH
Neckarstraße 76/1
D-71686 Remseck
Germany
Phone: +49 (0) 7146 8956-0
Fax: +49 (0) 7146 8956-29

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of NeuroCheck GmbH.

© 2014 NeuroCheck GmbH. All rights reserved.

www.neurocheck.com

Version 6.1 of 08.09.2014

NeuroCheck is a registered trademark of NeuroCheck GmbH.

Microsoft®, Windows®, Windows® Vista® are either registered trademarks or trademarks of Microsoft Corporation. FireWire® is a trademark of Apple Computer Inc.

Table Of Contents

The NeuroCheck Training Course consists of two manuals. This manual contains the first part of the training course with the following contents:

Welcome 5

1. Introduction 9

1.1 What is NeuroCheck? 10
1.2 Application areas 12
1.3 Modes of operation 15
1.4 Launching the software, intro screen 16
1.5 The NeuroCheck window in manual mode 18
1.6 Basic structure of a check routine 20

2. Working with Check Routines 21

2.1 The check routine wizard 22
2.2 Editing the check routine 30
2.3 Creating check routines manually 36
2.4 Outlook: Automatic mode 54

3. Configuration of Image Capture 67

3.1 Integrating digital cameras 68
3.2 Live image and camera configuration 76
3.3 Image capturing in the check routine 84
3.4 Exchange configurations of cameras 90
3.5 Camera reference images 96

4. Presence Verification 105

4.1 Object presence verification 106
4.2 Presence verification using hierarchical search 120
4.3 Presence verification using object properties 132
4.4 Presence verification using Template Matching 144
4.5 Optimizing Template Matching 156

5. Automated Inspection 163

5.1 Configuration of fieldbus for communication 164
5.2 Configuration of remote control 176
5.3 Execution of an automated inspection 184
5.4 Automated inspection with type change 196

In the following you'll find the contents of the second part of the training course:

6. Gauging 7

6.1 Gauging distances between two objects 8
6.2 Gauging with sub-pixel precision 26
6.3 Gauging parallel edges 30
6.4 Calibration to metric measurements 44

7. Configuration of Automatic Mode 51

7.1 Designing the process view 52
7.2 Configuration of check result 70
7.3 Creation of a control panel menu 72
7.4 Selection of a control panel 78

8. Position Adjustment 81

8.1 Compensation of offsets 82
8.2 Compensation of rotation 104

9. Data Input and Output 115

9.1 Write gauging results to file 116
9.2 Manual setting of target values in automatic mode 136

10. Troubleshooting 151

10.1 Configuration of error image storage 152
10.2 The event data base 160

Welcome

Welcome to NeuroCheck, the leading platform for machine vision on Microsoft Windows. With NeuroCheck you have chosen a professional software product that has stood the test in thousands of industrial vision applications.

The Training Course

This manual will acquaint you with the operation of the NeuroCheck software and present you the many fields of application in detail. It will guide you step by step through configuring automatic visual inspection applications. The Training Course manual is suitable for self study but also as a supporting document in a seminar.

To complete the Traning Course you need to install NeuroCheck 6.1 and a Professional License.

To conduct the examples without a camera, all image files used here are copied automatically to the project directory upon installation of NeuroCheck (the exact path depends on the operating system and language used):

`C:\Users\Public\Documents\NeuroCheck\6.1\Default\`

Furthermore, you will find the data of the project directory on the NeuroCheck installation DVD in the following folder:

`\Extras\Directories\NeuroCheck\6.1\Default\`.

The images can be found in the project directory sub-folder

`\Check Routines\Images\`.

In the project directory of the sub-folder

`\Check Routines\Examples\Training Course\`

you'll find the check routines that you will be creating according to the step-by-step instructions in this manual.

Each chapter of this Training Course is based upon the previous chapters: We will assume that the steps previously described and their effects are known and they will therefore not be repeated or only very briefly; on the other hand, the results of the previous chapters are re-used as the basis for examples.

Content

The NeuroCheck Training Course consists of two manuals. This manual contains the first part of the training course and is structured as follows:

- Chapter 1 *Introduction* explains the basic concepts and most important terms of the NeuroCheck software.
- Chapter 2 *Working with Check Routines* describes how to create and configure check routines as the central element of the visual inspection application.
- Chapter 3 *Configuration of Image Capture* shows how to integrate a digital camera into NeuroCheck and to transmit captured images to NeuroCheck.
- Chapter 4 *Presence Verification* illustrates how to verify the presence of objects using NeuroCheck.
- Chapter 5 *Automated Inspection* demonstrates how to control NeuroCheck in automatic mode externally.

You'll find the second part of the NeuroCheck Training Course in the manual "Training Course II". That manual is structured as follows:

- Chapter 6 *Gauging* describes how to measure the dimensions of objects and the distances between objects in NeuroCheck.
- Chapter 7 *Configuration of Automatic Mode* explains how to adjust the controls and the look of automatic mode.
- Chapter 8 *Position Correction* deals with the compensation of object position changes.
- Chapter 9 *Data Input and Output* shows how to output results measured with NeuroCheck and how to set target values dynamically for NeuroCheck.
- Chapter 10 *Troubleshooting* tells you how to diagnose problems when executing NeuroCheck.

Additional help resources

Getting Started:

This introduction to NeuroCheck will help you to install the system, will show you how to create your first image processing application using NeuroCheck, and how to get further information regarding working with NeuroCheck. The Getting Started manual is the ideal starting point for your entry into working with the NeuroCheck software when creating your first visual inspection application interactively.

Help System:

The help system describes all program functions of NeuroCheck 6.1 in detail and provides a comprehensive reference document for the NeuroCheck software. The help system provides you with the information you need and where you need it when working with the NeuroCheck software, whether in the lab or on the line. Using the NeuroCheck user interface in manual mode, open the help system by clicking on the menu item **‚?' ► Help Topics**. In addition you can access the appropriate information in the help system from most program dialogs.

NeuroCheck Software Support:

E-mail: support@neurocheck.com

We wish you fun reading the manual and success with your work with the NeuroCheck software!

1. Introduction

This chapter provides an overview about NeuroCheck and will introduce some important terms prior to the systematic step-by-step instructions for operating NeuroCheck detailed in the next chapter.

If you already have experience in working with NeuroCheck or if you have read the **Getting Started** manual, we suggest you skip this chapter and go directly to the step-by-step instructions in chapter 2 (*Working with Check Routines*).

1.1 What is NeuroCheck?

NeuroCheck is a universal machine vision software for all areas of automatic visual inspection in industrial manufacturing. NeuroCheck offers an integrated environment for the interactive configuration of visual inspection applications and their fully automated execution on the manufacturing line.

With its broad selection of image processing functions, NeuroCheck can be used in a wide range of application areas. Highly automated check routine configuration and an intuitive graphical user interface help to solve visual inspection problems. NeuroCheck reduces cost by easy set-up, reconfiguration and rapid turn-around.

The following simplified figure shows the basic setup of a visual inspection system:

The main features of NeuroCheck are:

Machine vision functions

The software encapsulates thousands of powerful and long-proven machine vision functions of the NeuroCheck image processing library. The check functions are logically grouped into categories such as image capturing, image pre-processing, machine vision analysis and measuring.

Camera integration

NeuroCheck supports a large number of modern digital cameras according to FireWire™ a/b and Gigabit Ethernet standards with various resolutions. The unique NeuroCheck device manager makes setup and configuration of cameras simple and easy.

Manual mode

Manual mode presents a development interface for creating an inspection solution with an interactive graphical approach. Here you can determine the logical structure of the check procedure and the parameters for executing the inspection process.

Automatic mode

In automatic mode, the previously configured check runs automatically. The system monitor visualizes status and inspection results in freely configurable windows. Process control is affected via PLC or master computer or manual intervention by an operator.

Process integration

For communicating with a supervisory control system, a number of modern standard interfaces are available. Execution of the inspection procedure can be affected dynamically by the process peripherals. NeuroCheck also sends the inspection results and measurement values to the peripherals.

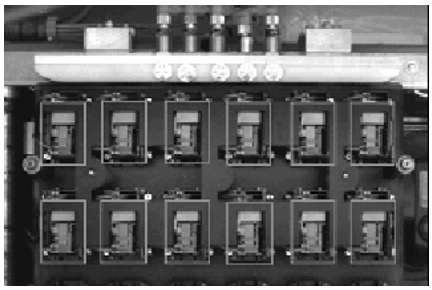
The NeuroCheck software is available in several licensing levels. These licensing levels differ in the features they provide.

1.2 Application areas

This section gives a brief introduction to typical visual inspection tasks NeuroCheck can be used for. You will find examples for all these application areas on the NeuroCheck Web site http://www.neurocheck.com and in the reference book **Industrial Image Processing** (Demant, Streicher-Abel, Garnica: Springer (Berlin, Heidelberg, New York), 2013, ISBN 978-3-642-33904-2), which presents solutions from all areas of industrial visual inspection created with NeuroCheck.

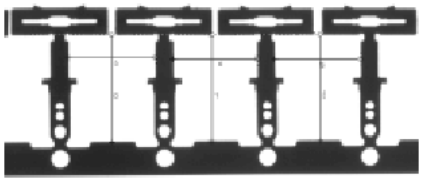
Presence verification / Completeness check

Use NeuroCheck to detect the presence of all required parts and components easily and reliably. The same functions can be applied to make sure that certain objects, for instance damaged parts, are not present.



Gauging / Dimensional inspection

You can use NeuroCheck to check every aspect of the geometry of a work piece, e.g. distances between objects, angles between edges, parallelism, concentricity.

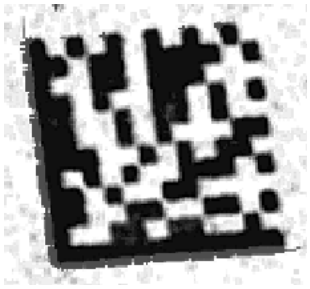


Character recognition (OCR, OCV)

NeuroCheck uses adaptive classifiers for character recognition. They can be trained to recognize characters in fonts created by marking technology used in industry, for example laser-engraving, stamping, printing…

Bar code and DataMatrix code identification

Different types of bar codes and the two-dimensional DataMatrix code are used for the identification of all kinds of products in packaging, logistics and manufacturing control. NeuroCheck's configurable code identification functions detect and decode such codes at any angle of rotation, in negative or positive print and under difficult surface conditions.

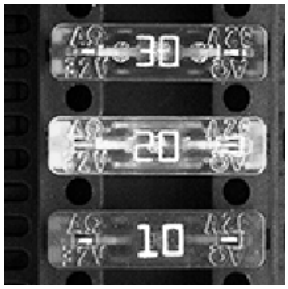


Pattern recognition

Parts can also be identified by readable letters or arbitrary symbols instead of bar codes. The neural networks used by NeuroCheck can learn from examples to recognize arbitrary patterns, enabling you to fine tune the recognition process to your particular application.

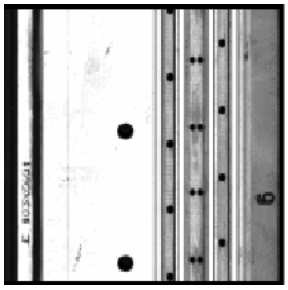
Color processing / Color recognition

All NeuroCheck image processing functions provide an extensive support for color images. Color processing is especially interesting if objects have to be distinguished based on their color or if segmentation of the objects is not possible in the gray scale range because they do not exhibit a brightness contrast.

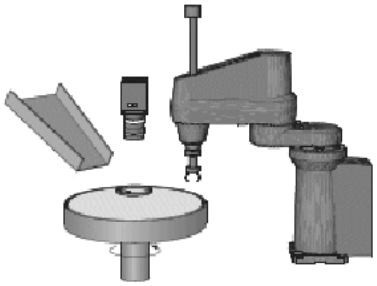


Surface inspection

Surface inspection applications frequently use line-scan cameras to scan the surface of large or cylindrical parts in order to check for scratches, holes, rough areas etc. Using linescan cameras in NeuroCheck is just as simple as using area-scan cameras. The same image processing functionality is available.



Position verification and Robot guidance



This topic summarizes all applications in which an image processing system is used to determine the position and orientation of an object and outputs this position in order to enable another machine to handle the object. A robot gripper can thus be enabled to grab work pieces from a conveyor belt. Another application is palletizing or depalletizing of crates containing assembly parts, packages or bottles.

Print inspection

The most typical application of image comparison methods is print quality inspection. In contrast to optical character recognition, the objective here is checking the quality of the print by evaluating the correspondence of the print to a reference image.

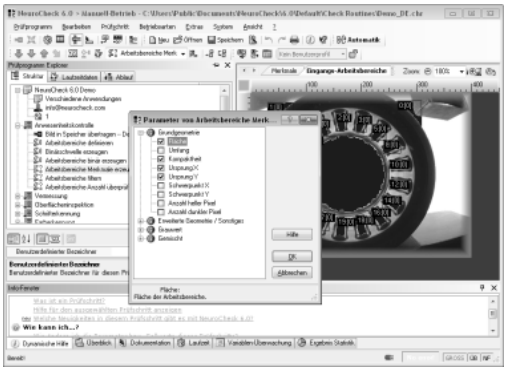
1.3 Modes of operation

NeuroCheck integrates both the development and runtime environment. On the one hand it offers a comfortable and powerful user interface for the interactive configuration of check routines, on the other hand it contains all features necessary to use the system in an automated production environment.

That's why there are two different modes of operation in NeuroCheck, each with its own particular user interface and with its own special purpose:

Manual mode

Manual mode is the "development environment" and provides a user interface for check routine editing. In Manual mode you construct check routines, set check function parameters and test the inspection step by step. Starting with chapter 2 (*Working with Check Routines*), each chapter of this Training Course will introduce a part of manual mode.



Automatic mode

Automatic mode is the "runtime environment" for executing the check routine, fully integrated into the manufacturing line. In Automatic mode, NeuroCheck is typically controlled from the outside, for example by a PLC via digital I/Os or Ethernet. The user interface of the NeuroCheck screen in Automatic mode is highly configurable. You'll find an introduction to automatic mode in chapter 2 (*Working with Check Routines*). The configuration of the user interface in automatic mode will be described in chapter 7 (*Configuration of Automatic Mode*).

To use the NeuroCheck software in **Manual mode** of operation, you need a license of the level **Professional** or **Premium**.

1.4 Launching the software, intro screen

After launching the NeuroCheck software, the intro screen will welcome you to the software. It contains clearly arranged status information and buttons for quick access to important software functions.

Buttons

Using the buttons on the left side of the window, the intro screen makes the following selections available to you:

The **Website** button will start a web browser directing you to the NeuroCheck website.

The **Help** button will open the Getting Started Manual as a help file.

The **Demo** button will install an extensive pre-configured check routine to demonstrate the scope and areas of application of the NeuroCheck software.

The **Wizard** button will start the Check Routine Wizard for the semi-automated creation of your first check routine.

The **Info**button will open a dialog with information about the current software version and license status.

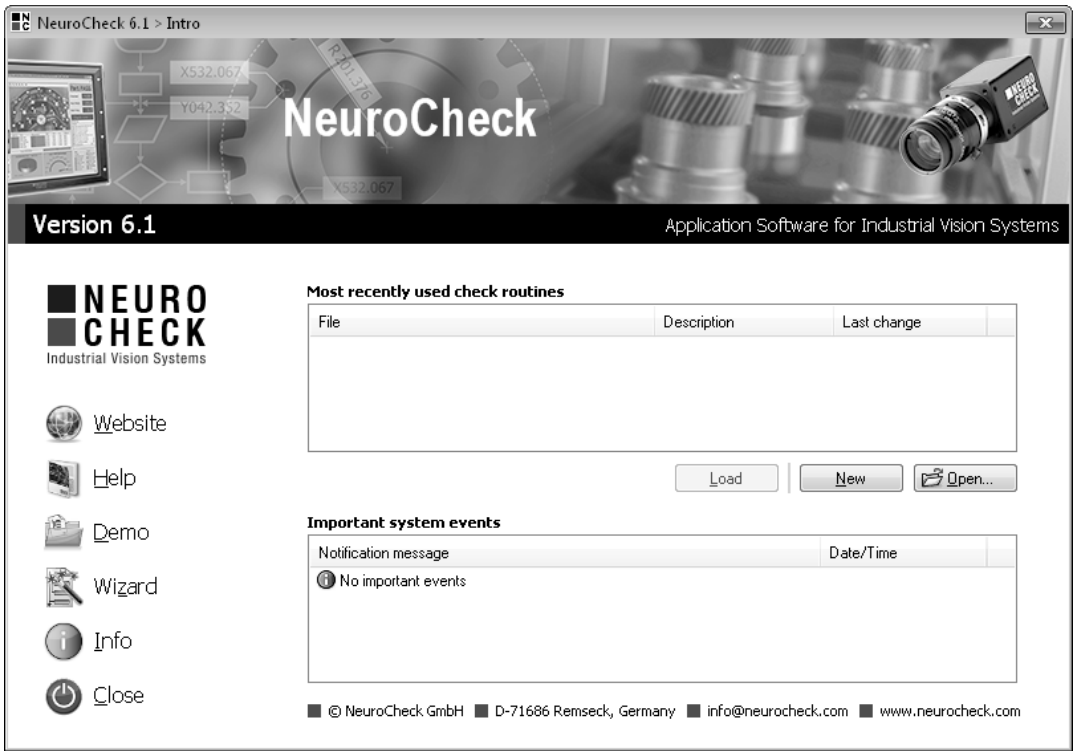
The **Close** button will close the software.

Status information

On the right hand side of the intro screen you see two list boxes. In the picture shown here, the list boxes already contain several entries. When you launch the software for the first time, both list boxes should be empty.

The **Most recently used check routines** list enables quick access to the check routines last edited. Using the **Load**, **New** or **Open** buttons, you can switch to manual mode and work with the check routine

The **Important system events** area list shows the most recent important system events. Logging system events helps with system diagnosis. Double-click one of the list items to open the **Event log viewer** showing more details for the event.



NeuroCheck intro screen.

1.5 The NeuroCheck window in manual mode

In manual mode the NeuroCheck window is the development environment for your check routine. The window comprises the following areas:

❶ **Main menus and tool bars**

The commands from the menu or tool bars are used to create and manage check routines, configure automatic mode, hardware etc. .

❷ **Check routine explorer**

Use the three tabs of the Check routine explorer to configure your check routine.

The **Structure** tab is the default view where the image processing steps are determined as a sequence of checks and check functions.

The **Runtime input** tab determines the kind of data to be processed by a check function.

On the **Flow** tab you can configure the execution of checks depending on the results of other checks.

❸ **Result view**

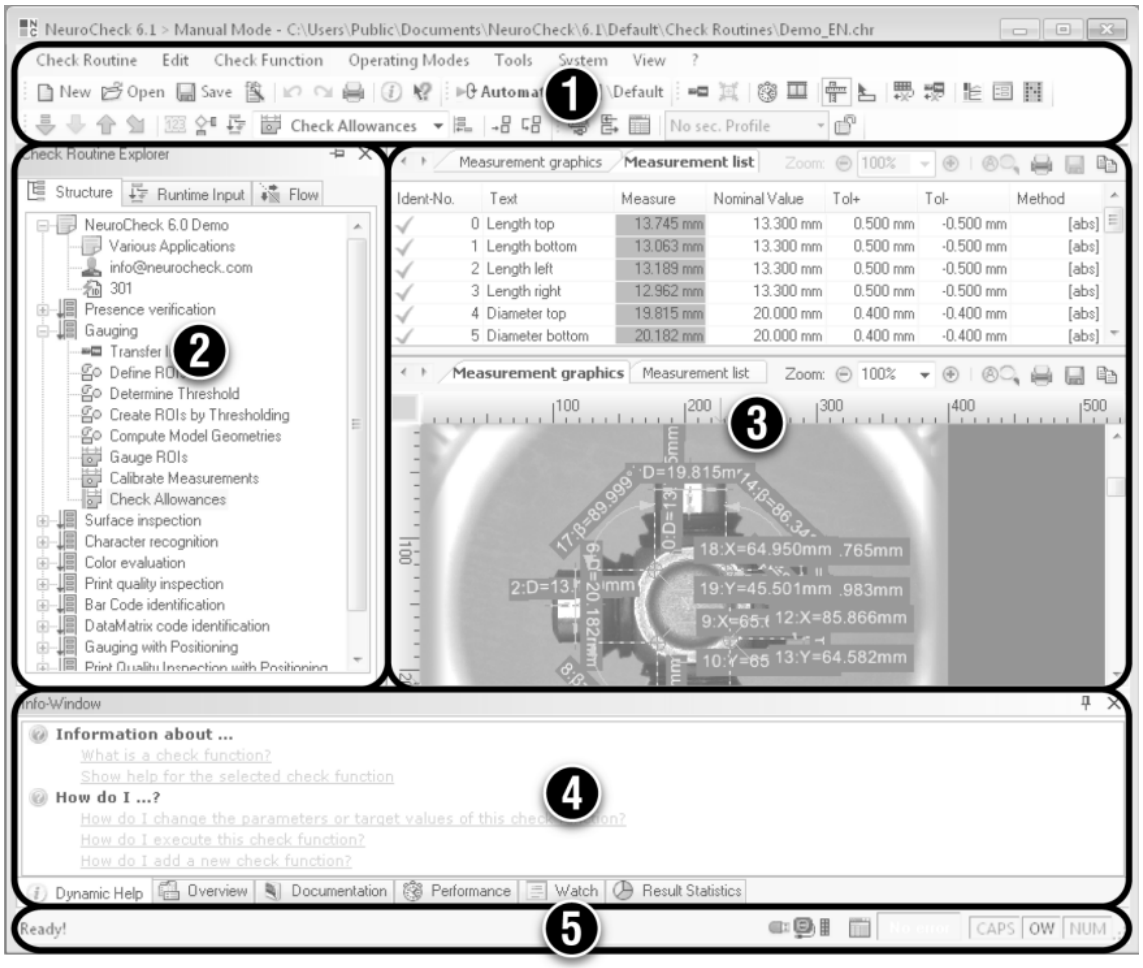
The result view shows the results of the last check function executed or various other types of status information. It can be split into two areas, for instance to view a result image and a result table simultaneously.

❹ **Check routine information window**

This window helps you to create a check routine. Here you'll find context sensitive dynamic help and detailed information about the currently selected object in the Check routine explorer.

❺ **Status bar**

Various symbols in the **Status bar** tell you whether required hardware components are currently available to the system and have been installed correctly.



The areas of the NeuroCheck window in manual mode.

1.6 Basic structure of a check routine

Check routine

The check routine is the central element of any vision application being implemented with NeuroCheck. It can be compared to a document in a word processing software or the source code in a development environment. It defines the structure and flow of the entire visual inspection of a test piece. Usually the check routine is made up of several checks.

Check

Checks inspect various aspects of a work piece. In multi-camera systems a check is often assigned to one camera.

Each check consists of several check functions. When the check routine is executed, all checks and their check functions are executed sequentially.

Check function

Check functions are image processing, analysis or system functions that can be inserted into a check to fulfill specific tasks. NeuroCheck offers check functions for a large variety of tasks such as filtering entire images, searching for individual objects or gauging them. All in all NeuroCheck has more than 80 different types of check functions.

After this initial overview of NeuroCheck, the systematic step-by-step instructions for using NeuroCheck begin in the next chapter.

2. Working with Check Routines

In this chapter you'll learn how to use NeuroCheck

- to create check routines and manage them as documents like documents in a word processing software
- to create a check routine for reading a bar code using the check routine wizard
- to use the Check routine explorer
- to prepare step-by-step a check routine for reading a bar code
- to execute a check routine in automatic mode.

2.1 The check routine wizard

Goal

You will create a short check routine to read a bar code.

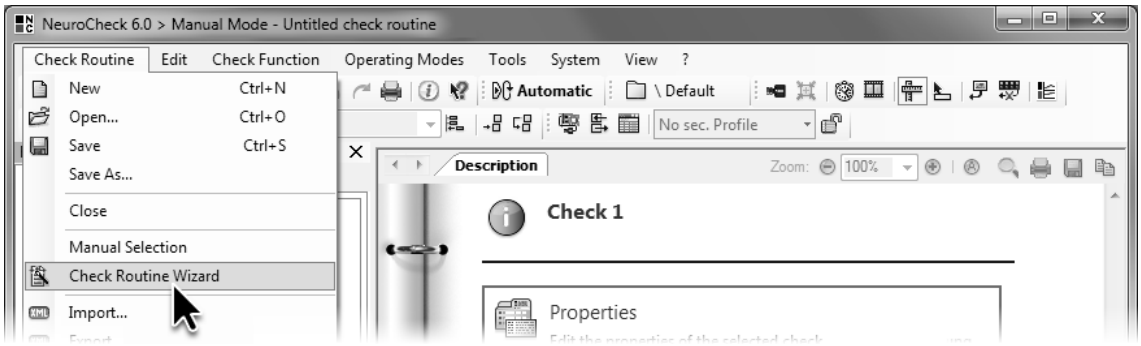
Result

You will be able to create and execute a simple vision application with the help of the check routine wizard.

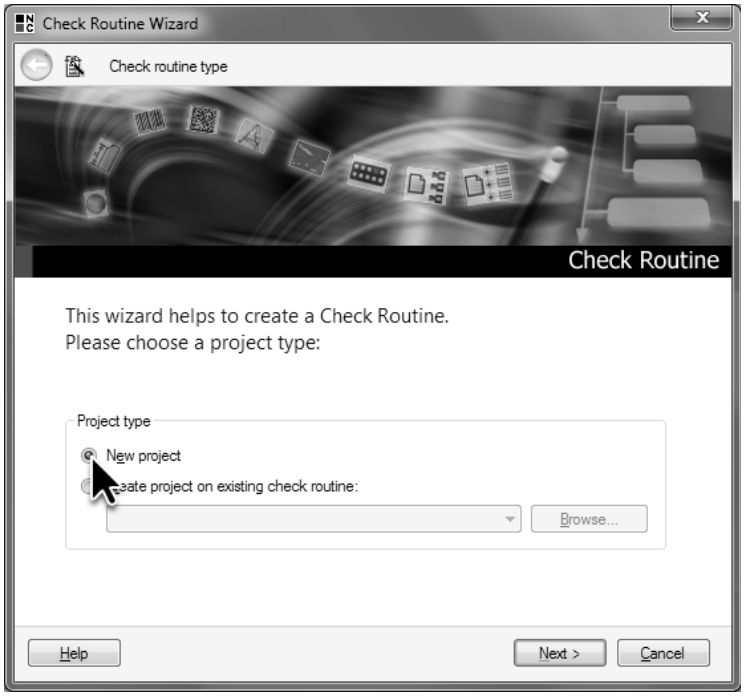
Solution

1. Select **Check Routine Wizard** from the **Check Routine** menu, click on this icon on the tool bar or click on the **Wizard** button on the intro screen.

2. The first page of the check routine wizard is displayed where you can select whether you want to create a new project or a project based on an existing check routine. Select **New project**. Click the **Next** button to switch to the second page.



1. Select the menu item **Check Routine ► Check Routine Wizard.**



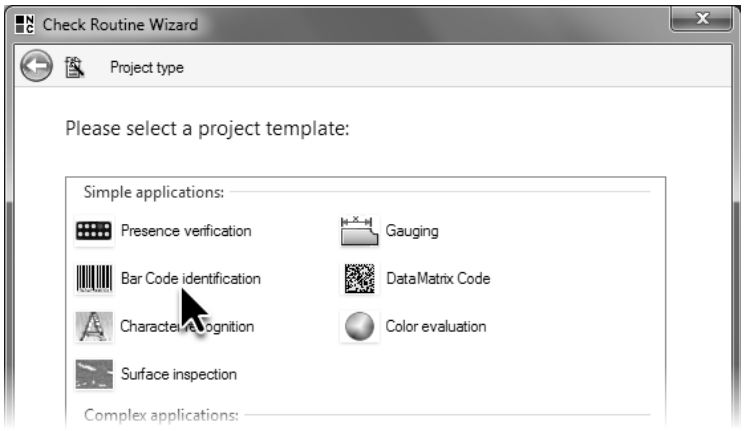
2. Select **New Project** to create a new check routine.

3. On the second page of the check routine wizard you can choose from various basic types of vision applications. Select the **Bar Code identification** type and go to the next page by clicking **Next**.

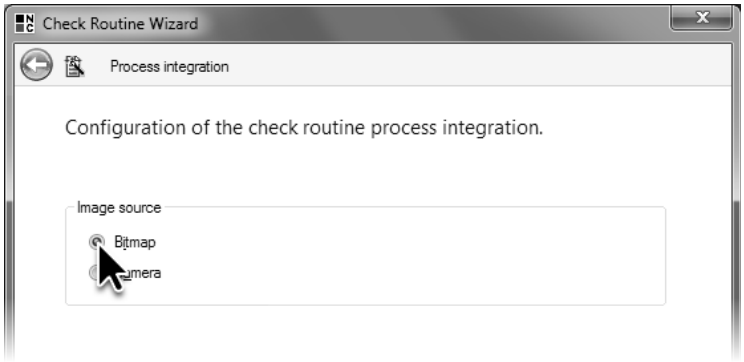
   The third page of the check routine wizard contains a number of options to comment the check routine, which we will ignore for the time being. Again, click **Next** to switch to the next page.

4. On the fourth page you can determine whether to use a file or a camera as the image source. Since we will deal with cameras in a later chapter, please enable **Bitmap**. Click **Next** to get to the last page of the check routine wizard where you will find an overview of the selected settings.

If you want to change one of the settings, click on this icon to switch to the previous wizard page. When everything is correct, click **Finish**.



❸ Select the **Bar Code identification** project template.



❹ On page four, select bitmap as the image source.

5

The check routine explorer on the left of the NeuroCheck window now displays several icons and lines of text. This represents the saved check routine created by the check routine wizard. One of the lines titled **Check 1** is marked with this icon. The lines below **Check 1** symbolize the check functions. Each check function fulfills a specific vision task.

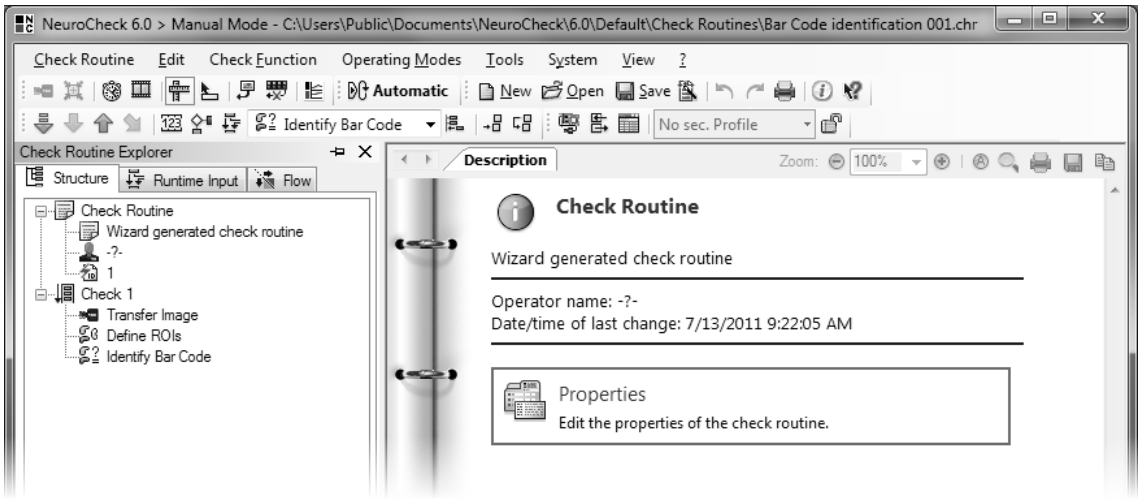
6

Left-click the first check function **Transfer image**. Select the menu item **Check function ► Execute** or click on this icon on the edit bar. The window on the right displays a gray-value image of a bar code.

If you click **Check function ► Execute** or the corresponding icon again, a different image of a bar code is displayed on the right. The wizard has set up a total of four images, each with one bar code to read. Every time you execute the check routine, a different image is used. You can see that one of the images shows a different bar code.

If the width of the right window is insufficient, you can move the separation between the two window areas. For this, move the mouse over the separating line until the mouse pointer changes into the depicted symbol. Press the left mouse button and hold it down. As long as you keep the button pressed you can change the separation by dragging the mouse.

You can also change the scale of the image by entering a different value into the list on the top frame of the display area, or by clicking one of the depicted icons.



❺ The check routine wizard creates a new check routine that will be displayed in the **Check Routine Explorer**.

❻ Result of the first check function **Transfer image**.

7

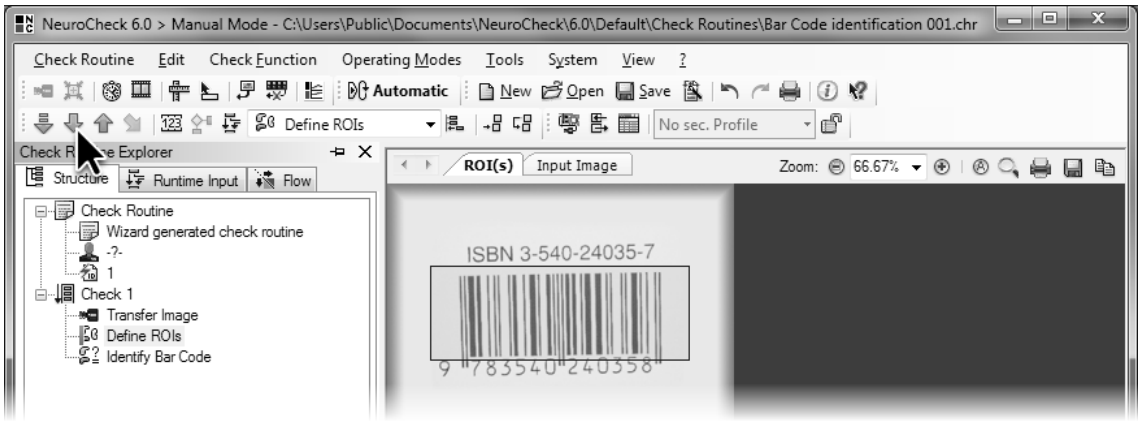
Click the depicted icon on the edit bar or select the menu item **Check function ► Next**. On the left of the check routine window the next check function **Define ROIs** will be highlighted. Also, the check function will use the image last transferred. In the **Result view** a blue rectangle appears around the bar code. This is the area where NeuroCheck will be looking for the bar code.

8

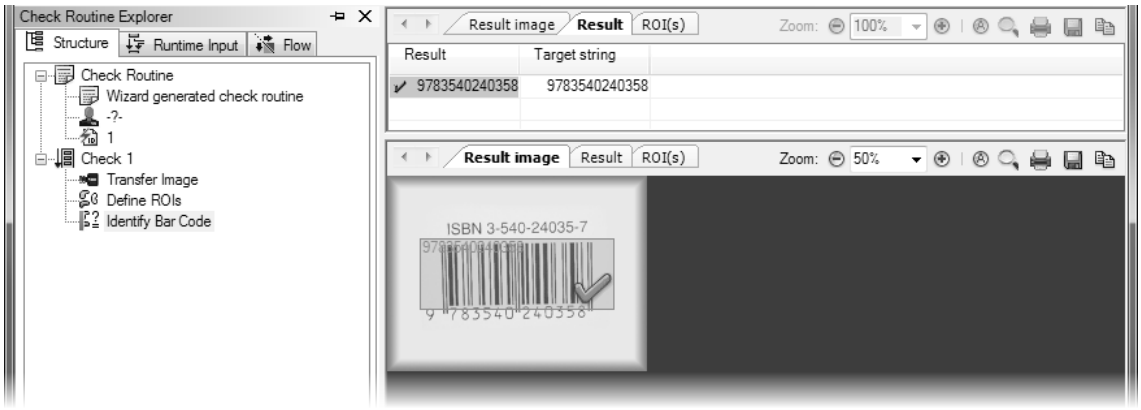
Execute the next check function **Identify bar code** in the same way. In the **Result view** the image of the bar code with ROI will be displayed including the bar code read in the upper left corner of the ROI.

Moving the mouse pointer to the window frame above the **Zoom** list box, it will change into a broad horizontal line and an up and down arrow. Keep the left mouse button pressed and drag the mouse downward. The window is split into an upper pane displaying a list with the bar code read and, a lower pane with the image of the bar code read.

Both windows have their own tabs to select the view. If the window displaying the image is active (i.e. clicked last), the buttons for changing the zoom level, enabling the magnifying glass, printing, saving and copying the image to the clipboard are available. Right-click on this window to open the context menu. The context menu gives access to the same functions as the buttons.



❼ Result of the second check function **Define ROIs**. NeuroCheck will later look for the bar code in the area displayed.



❽ In the split **Result view** NeuroCheck will display the search area on the bottom and the bar code read on top.

2.2 Editing the check routine

Goal

You can reuse the structure of an inspection several times by duplicating checks and adapting the properties.

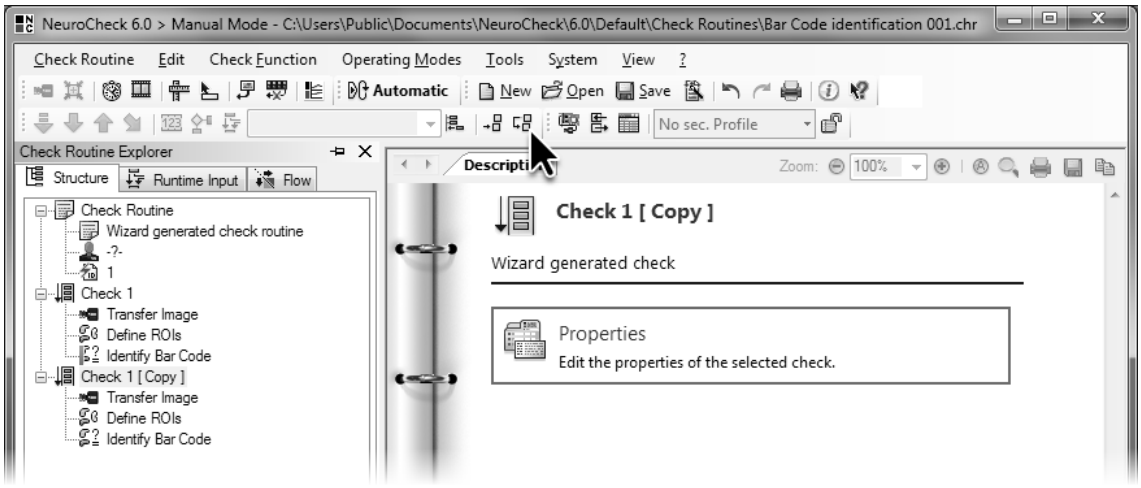
Result

You know how to handle the Check routine explorer, copy and delete checks.

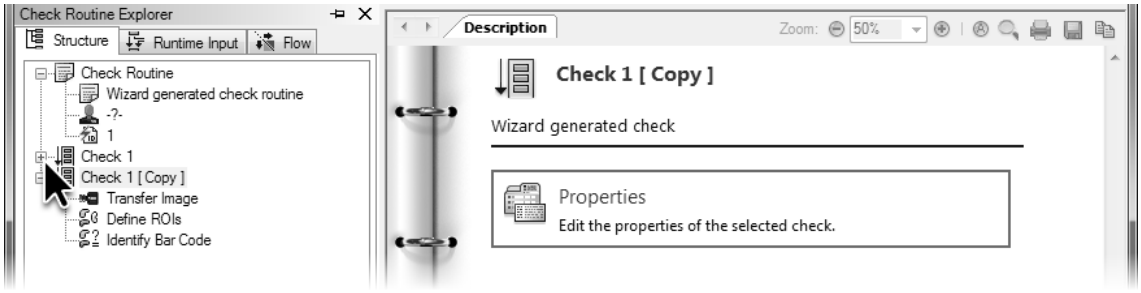
Solution

1. Left-click **Check 1** in the tree structure of the check routine. Click the depicted icon on the edit bar or select the menu item **Edit ► Copy Check**. Below the first check of the check routine the exact same sequence of check functions will appear. Thus you have created a second check as a basis for solving the second task, to which **[Copy]** has been added to the name.

2. Left-click the "minus" symbol in front of the first check. The check functions will be hidden leaving only the title line of the check. The "minus" symbol changes into "plus". If you click the "plus" symbol, the check functions of the first check will reappear. You can do the same by double-clicking the title row of the checks.



1 Check routine after copying the first check.



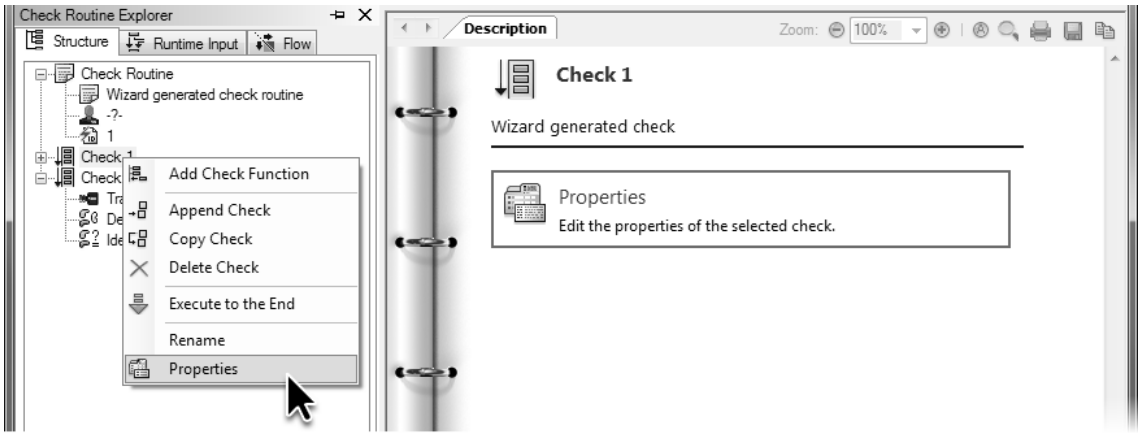
2 Check routine with collapsed first check.

3. Usually both checks should fulfill different tasks and have different names. Right-click the title row of the first check and select **Properties** from the context menu.

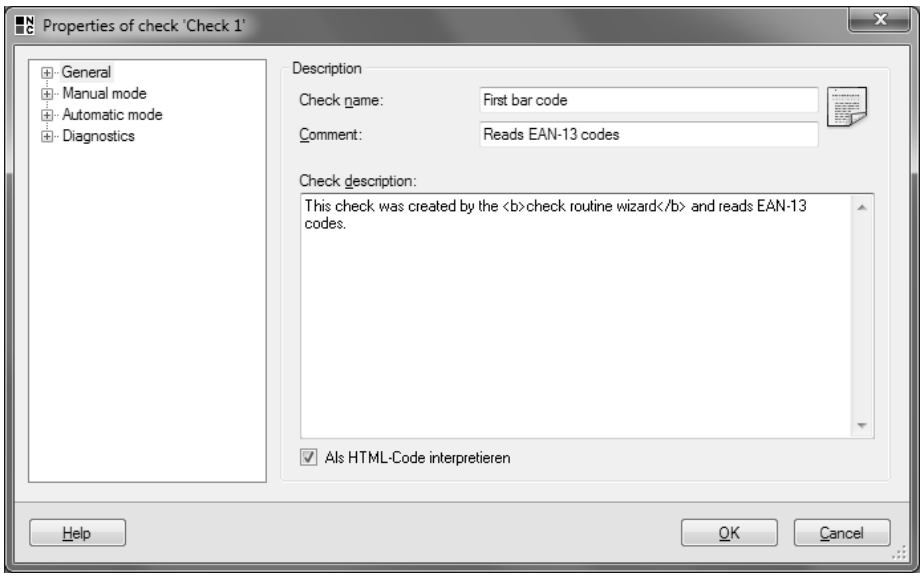
4. A dialog opens and you can enter a name, a comment and a detailed description of the check. You can use the description to describe the task, the procedure and the features of the check. This way you can document the check routine thus facilitating system maintenance.

   Enable **Interpret as HTML code** and enter the following text into **Check description**: "This check was created by the check routine wizard and reads EAN-13 codes." Both strings within angle brackets are HTML commands ensuring that "check routine wizard" is bold. You can use various HTML formatting such as fonts, lists and images for the description field.

   You can supply detailed description with HTML formatting in the property dialogs of check routines, check and all check functions.



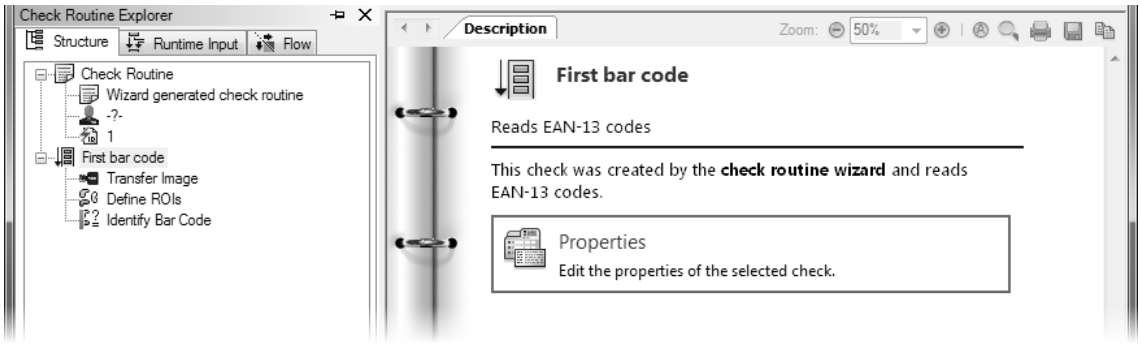
❸ Open the **Properties** dialog via the context menu of the check.



❹ Titles and descriptive texts for checks are useful to identify finished solutions later.

5. Exit the dialog by clicking **OK**. For the selected check these texts will be displayed with formatting in the right window and provide both the developer of the inspection task and the user with important information.

   We will deal with check routines featuring several checks later in more detail; for now, you can delete the new checks. Right-click the title row of the second check again and, this time, select **Delete Check**. Alternatively, left-click the check and press the Del button. This will reset the check routine to its original state.



❺ The result view will display the description of the check with HTML formatting.

2.3 Creating check routines manually

Goal

To create a check routine for reading a bar code without the help of the check routine wizard.

Result

You can create a check adding check function by check function, and set the parameters of each check function.

Solution

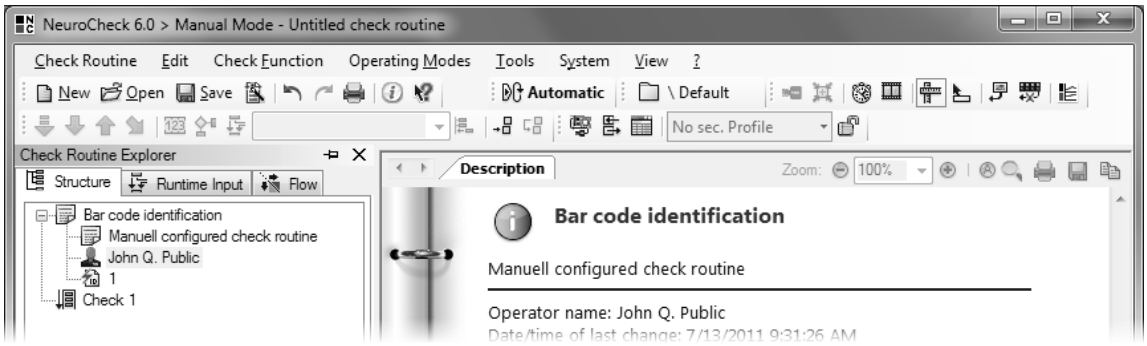
Before you start to solve completely new inspection tasks, you'll get to know the basic steps for creating a check routine in this section. We will use the check routine we have created with the help of the check routine wizard in the previous section as a model.

1. First of all, create a new check routine using **Check Routine ► New**. This creates a check routine with an empty check.

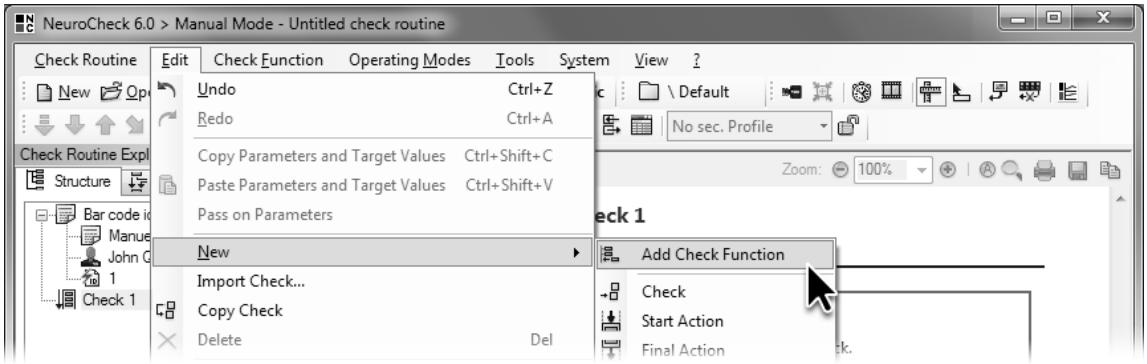
You can change the name of the check routine by in-place editing: left-click "Check Routine", then click again (make a pause to avoid double-clicking). Now enter "Bar code identification". Do the same with the comment row just below and enter "Manually configured check routine". Enter your name as the operator.

Save the check routine with the name "Bar code manually" to the folder suggested by NeuroCheck.

2. The first check function of the check automatically created by the check routine wizard was **Transfer Image**. This check function will now be added to the check as the first check function. Select from the **Edit ► New** menu or from the context menu of the title row of the check ("Check 1") the item **Add Check Function** or click the depicted icon on the edit bar. The dialog box **Select Check Function** will appear.



❶ Create a new check routine, enter title, comments and your name, and save the check routine.



❷ Select from the **Edit ► New** menu or from the context menu of the check the **Add check function** command to open the **Select Check Function** dialog.

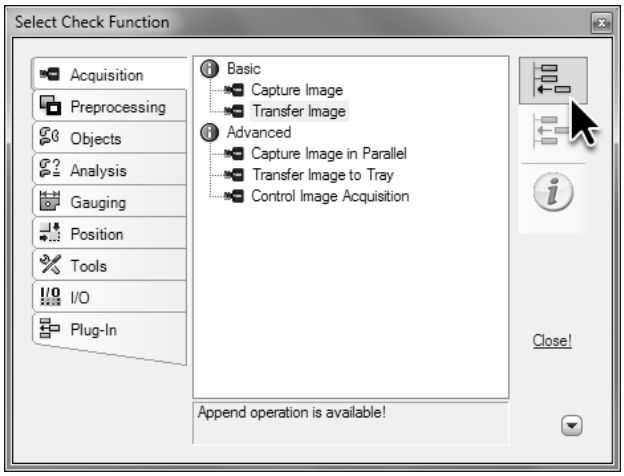
❸

In the **Select Check Function** dialog all check functions are divided into groups. You can select each group by pressing the buttons on the left side of the dialog. Click the button for the **Acquisition** group. You will then see all check functions for capturing and transferring image data.

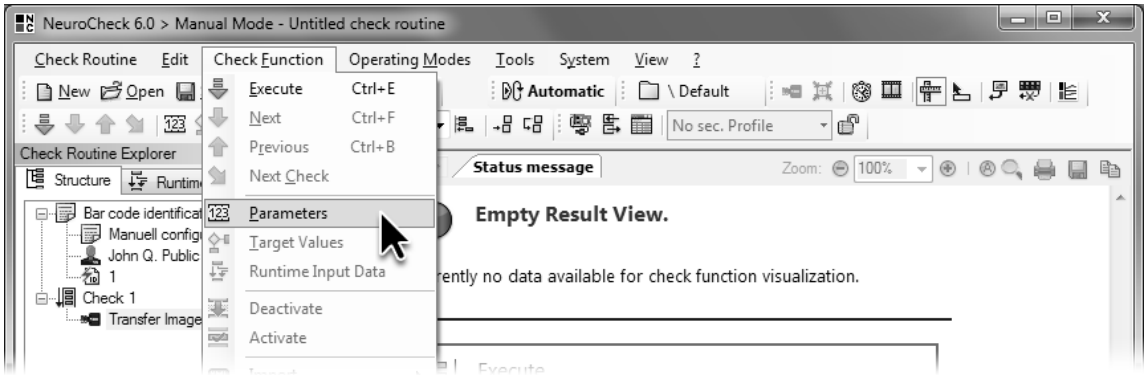
Select the check function **Transfer Image** by clicking, then click the depicted button, or double-click the check function **Transfer Image**. The first check function will appear in the Check routine explorer. The **Select Check Function** dialog will remain open. For useability, you can close it at any time and open it again later to add the next check function.

❹

Before adding additional check functions, enter the image data source for the check function already created. From the **Check function** menu select the item **Parameters**, or click the depicted icon on the edit bar. The parameter dialog **Transfer Image** appears.



❸ Select the check function **Transfer Image** by clicking and click the upper button, or double-click the check function **Transfer Image**.



❹ From the **Check Function** menu select the item **Parameters** to open the parameter dialog.

5. As before with the **Check Function Wizard** we are going to load images from files. As **Image source** you have to select **Bitmap file** which is the default setting when no camera is configured in NeuroCheck.

   To select images, click the **Browse…** button.

5 Select **Bitmap file** as image data source and click **Browse…** to select images.

6. To load several images sequentially, select the files `Demo60_gray.bmp`, `Demo60_Gray_001.bmp`, `Demo60_Gray_002.bmp` and `Demo60_Gray_003.bmp`. Upon opening the dialog for the first time, the `Check Routines\Images` folder from the NeuroCheck project directory - where the four images mentioned before can be found - is displayed automatically. Confirm your selection by clicking **Open**.

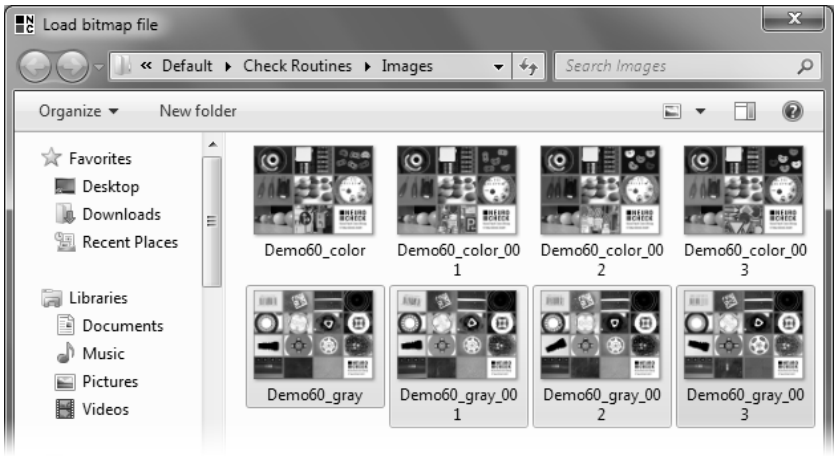
   The path of the NeuroCheck project directory depends on the operating system and language used. For an English Windows system the path for the project directory is:

   `C:\Users\Public\Documents\NeuroCheck\6.1\Default\`

   The project directory contains all files necessary for fully automated visual inspection with NeuroCheck.

7. Since only the top left field of the image is needed to read the bar code, we will reduce the image section. For this, disable **Always full image**.

   Left-click the blue image section. Holding down the left mouse button, drag the size boxes on the edges and corners of the image section to change the size. Please note that for a reliable bar code identification there must be space before and after the bar code. This space is called quiet zone. That's why the image section must not be too small. Exit the dialog by clicking **OK**.



6. Select the images `Demo60_gray`, `Demo60_Gray_001`, `Demo60_Gray_002` and `Demo60_Gray_003`.

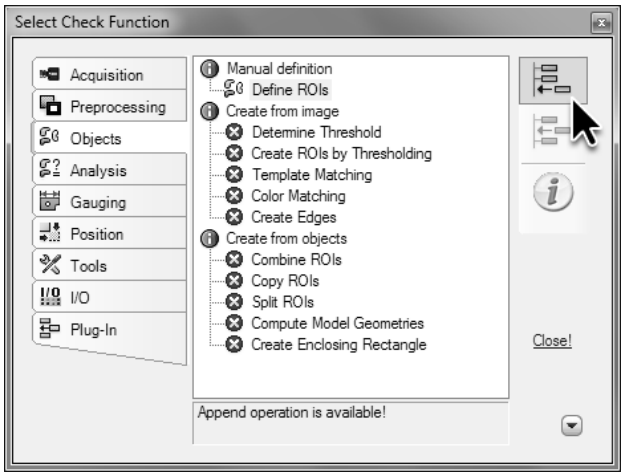
7. Adjust the image section with sufficient space before and after the bar code.

❽

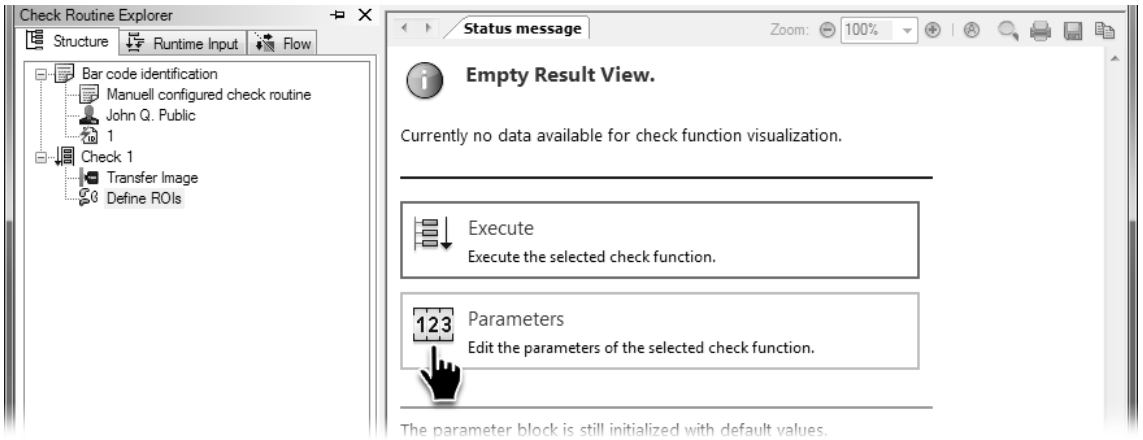
Add the check function **Define ROIs** to the check. For this, go to the **Select Check Function** dialog. If you have closed the dialog, follow the steps in ❷ to open it again. You'll find the check function **Define ROIs** by clicking the **Objects** button. In NeuroCheck all objects in images are described by regions of interest (ROIs).

❾

Once more, some parameters need to be set in the parameter dialog. Open the parameter dialog by clicking the **Parameters** button in the Result view, or as described in ❹.



8 Add the check function **Define ROIs** from the **Objects** group to the check.

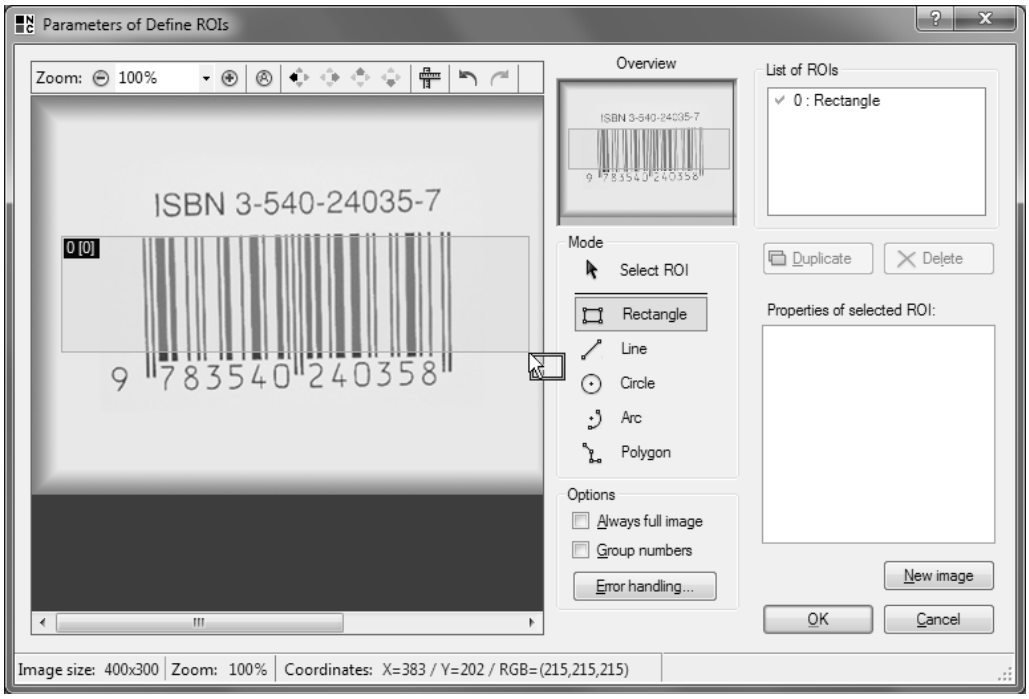


9 Click the **Parameters** button in the Result view to open the parameter dialog.

10 Select **Rectangle** mode for the ROI definition. Drag a rectangle horizontally through the bar code in the image area leaving sufficient search space on the left and right of the bar code.

If you cannot find the correct position and size for the rectangle right away, you can click the **Select ROI** button, then click the rectangle and modify it as described in 7 for the image section about to be loaded.

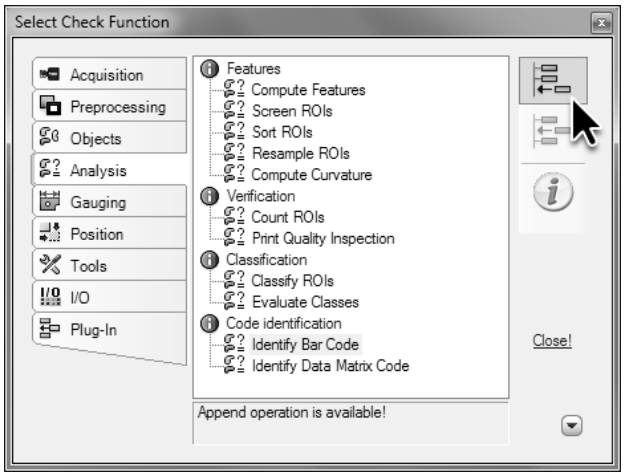
Exit the dialog by clicking **OK**.



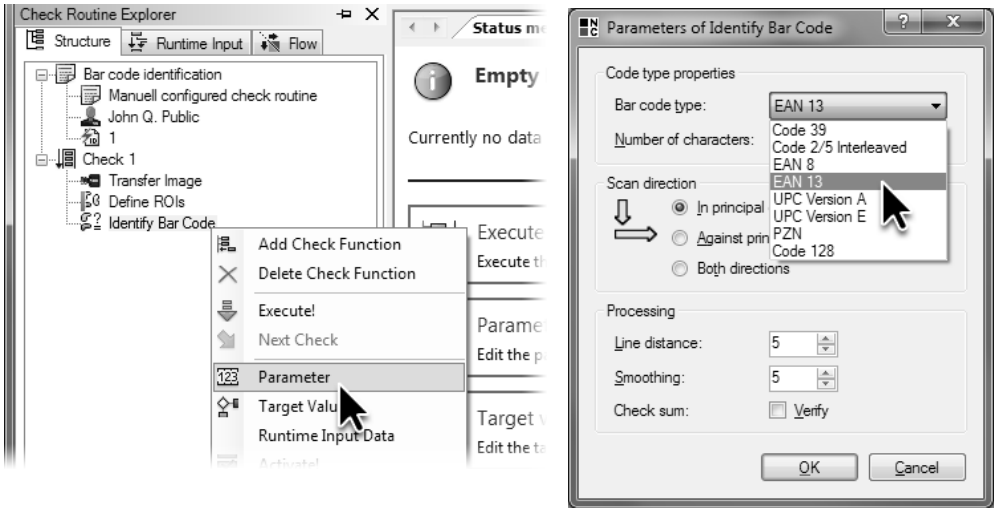
10 Drag a rectangle horizontally through the bar code in the image area leaving sufficient search space on the left and right of the bar code.

11 Add the check function **Identify Bar Code** to the check. You will find this check function in the **Analysis** group since it makes an identification and compares the result to a target value.

12 Open the context menu of the **Identify Bar Code** check function and select **Parameter** to open the parameter dialog. Enter the parameter dialog and in the field **Bar code type** select the **EAN 13** type and leave all other settings to default. Upon executing the check function, the bar code is read.



11 From the **Analysis** group, select **Identify Bar Code** and add it to the check.

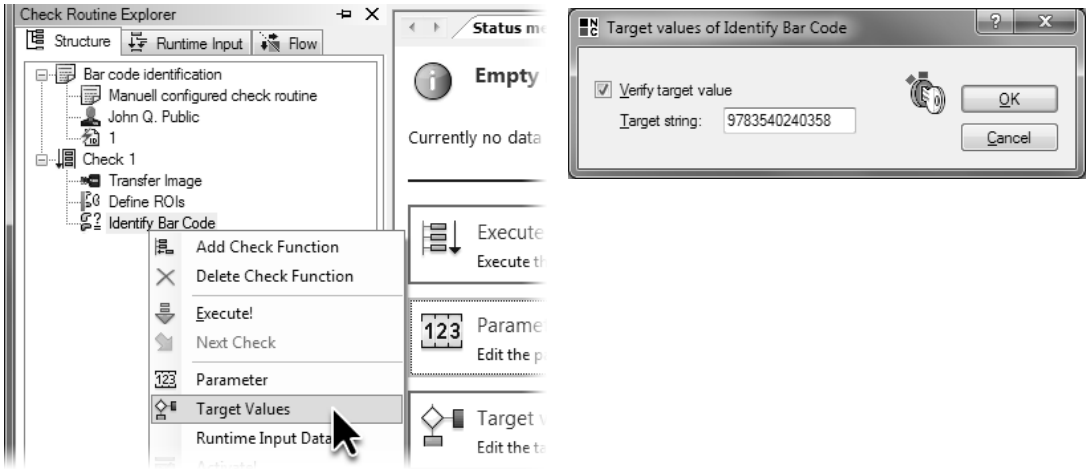


12 Select the **Bar code type EAN 13** from the parameters of the **Identify Bar Code** check function.

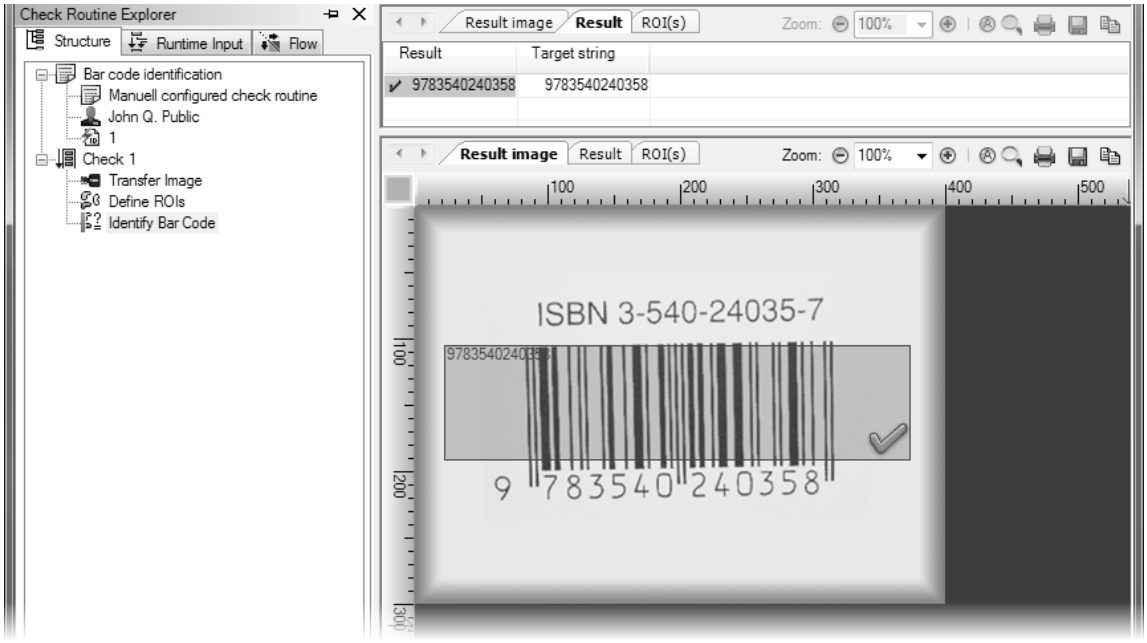
13

As a decision check function **Identify Bar Code** contains a target value dialog, which can be opened using the **Target Values** command from the **Check Function** menu, or from the context menu of the check function on the edit bar. The corresponding button symbolizes the decision being made by the check function between flawless and faulty parts. Enable **Verify target value** in the dialog and enter the string "9783540240358" as target string.

14 After exiting the dialog by clicking **OK** if you execute the check routine several times, it will yield *OK* for three images and *not OK* for one image.

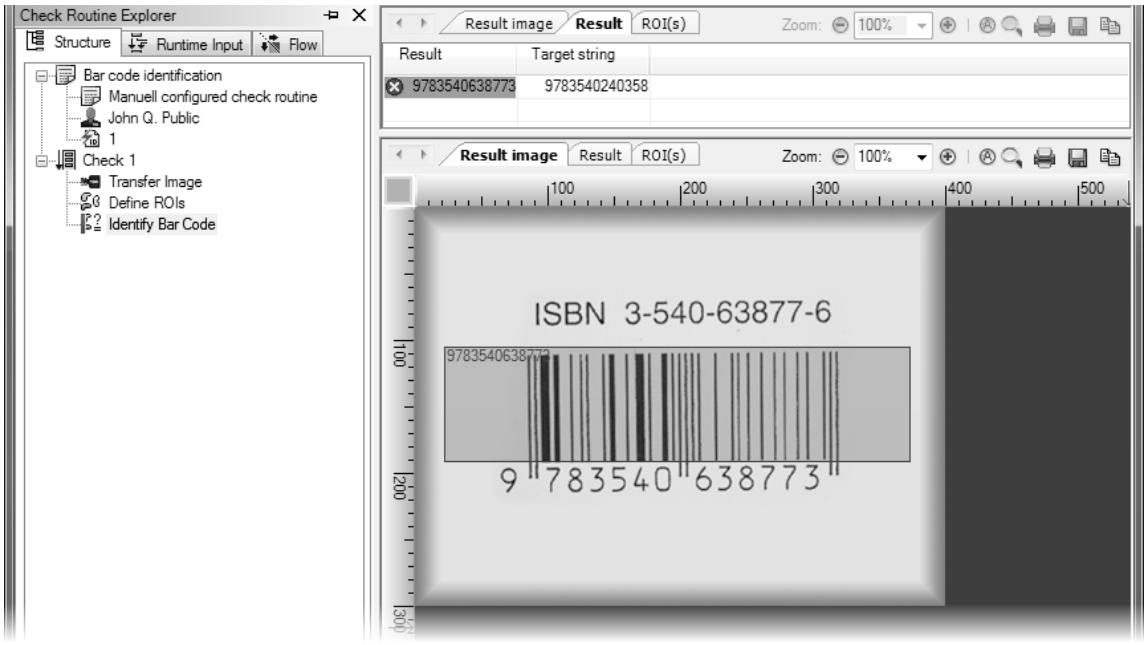


13 Open the check function's **Target Values** dialog and enable **Verify target value.**



14 The check routine will yield the same result as the one created with the **Check Routine Wizard** .

⓯ The fourth image shows a bar code with a different content, which is why the reading result differs from the pre-defined target value.



15 For bar codes with different content, NeuroCheck reports a target value error.

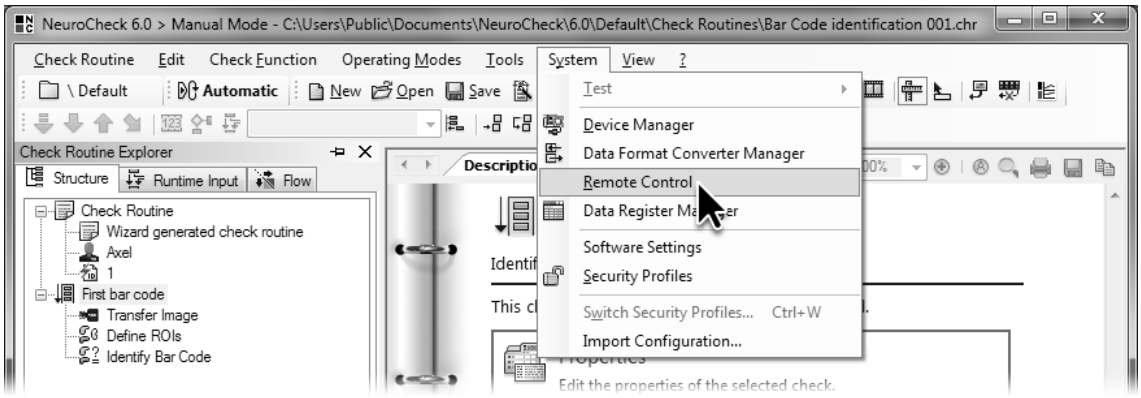
2.4 Outlook: Automatic mode

This section will give you a first impression of how simple it is to create a visual inspection with NeuroCheck that can be integrated into an automatic manufacturing process.

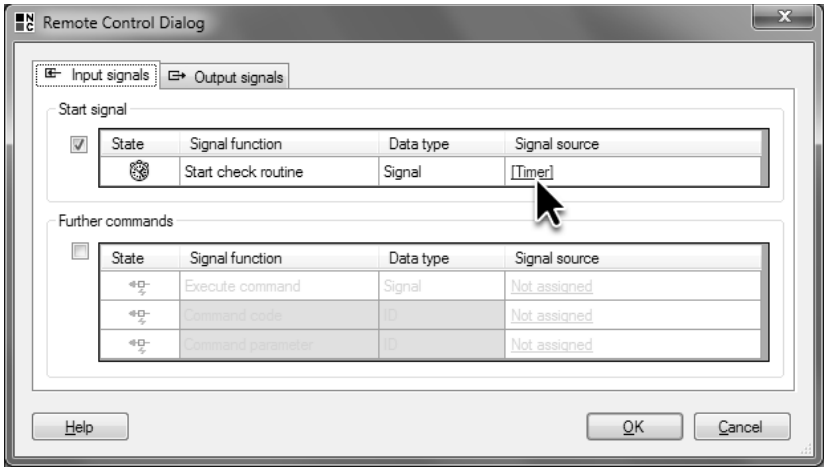
With a real application some more settings would be necessary to allow NeuroCheck to communicate with your PLC, your MES etc. These are mostly global settings you only have to set once. The automatic execution of a check routine is really as simple as it appears.

1. Select **Remote Control** from the **System** menu.

2. On the **Input signals** tab, enable the **Start check routine** signal by clicking the check box. Click the link in the **Signal source** column of the **Start check routine** row to select the signal source for the start signal. After installing NeuroCheck, this link is initially called **Timer** but after changing the source of the start signal, the link will change accordingly.



1. Select **Remote Control** from the **System** menu.



2. Enable the **Start check routine** signal.

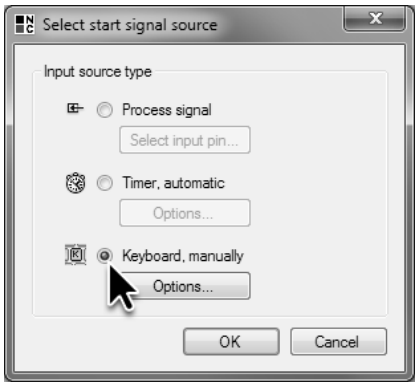
❸

In the now open **Select start signal source** dialog, select **Keyboard, manually** as the source for the **Start check routine** signal. With a real application, the start signal would be coming from an external control such as a PLC. The entire configuration necessary for this will be discussed in detail in a later chapter. Exit the dialog by clicking **OK**.

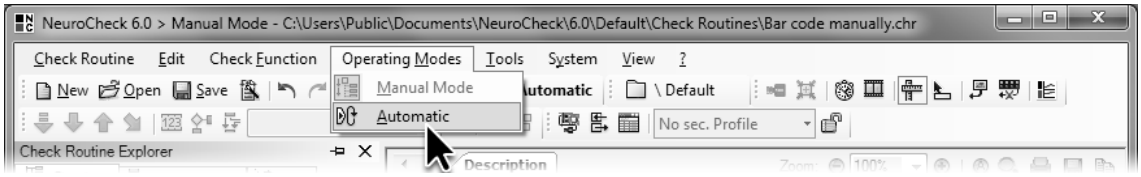
In the **Remote Control** dialog, you will now see in the right column of the **Select start signal source** row that the keyboard has been configured as the signal source. Exit the dialog by clicking **OK**.

❹

Select **Automatic** from the **Operating Modes** menu or click the depicted button.



❸ Select **Keyboard, manually** as start signal.



❹ Switch to automatic mode.

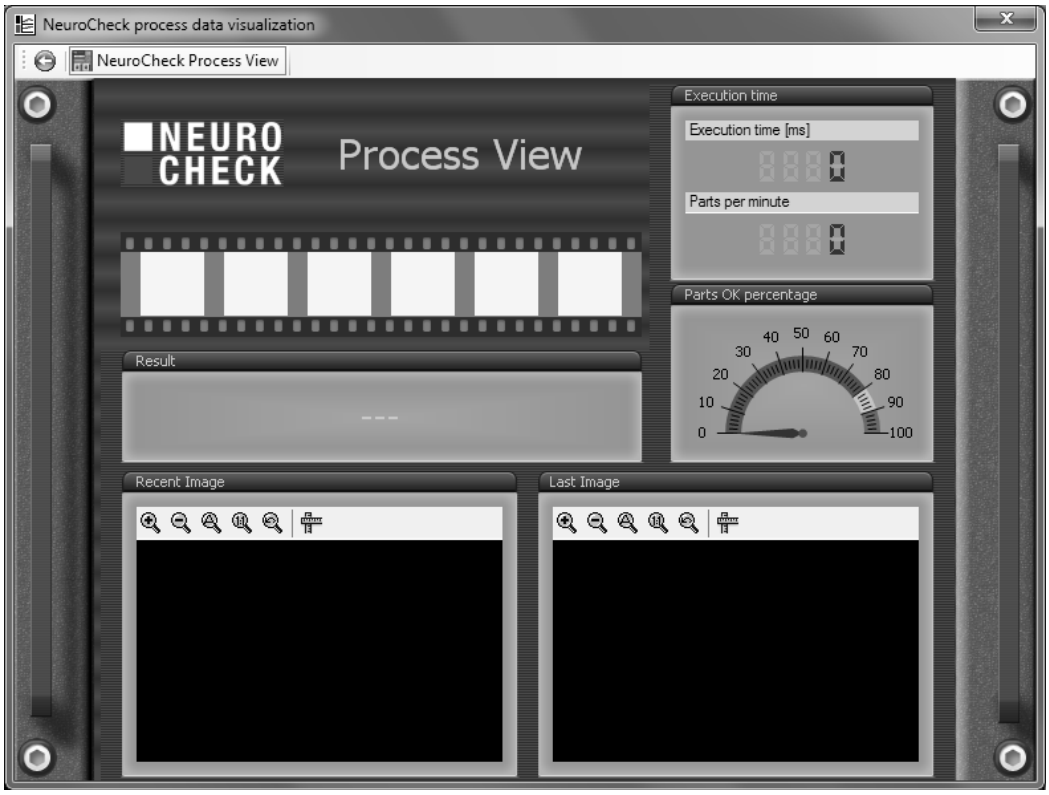
5 NeuroCheck switches to automatic mode and displays two windows. One window bears the title "Control Panel" on the right. The **Control Panel** is the NeuroCheck main window in automatic mode. Use the **Control Panel** to control NeuroCheck interactively in automatic mode. In addition, it provides important information on the system status.

❺ The **Control Panel** is used to control NeuroCheck in automatic mode.

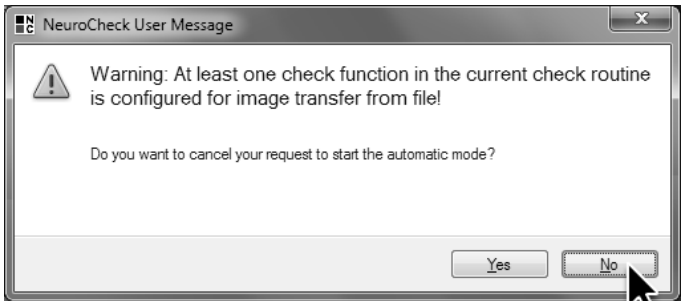
6. The other window displays the title "Process View". The **Process View** is the optional visualization of the current inspection results. The design options of **Control Panel** and **Process View** will be explained in detail in a later chapter.

7. Make the **Control Panel** the active window. Press the F3 key, the *F3* button or click *F3: Start Automatic*. A warning is displayed that the current images are loaded from file. In an automated visual inspection system that is part of a manufacturing line it would be a fatal mistake to inspect images from file instead of those captured by a camera, hence the warning. In this case, however, there's nothing wrong with using images from file. Click the **No** button since you do not wish to cancel automatic mode.

   The text *F3: Start Automatic* in the **LCD Matrix** of the **Control Panel** is now disabled. Automatic mode is now active and NeuroCheck is waiting for the start signal, this is also displayed in the window's title.

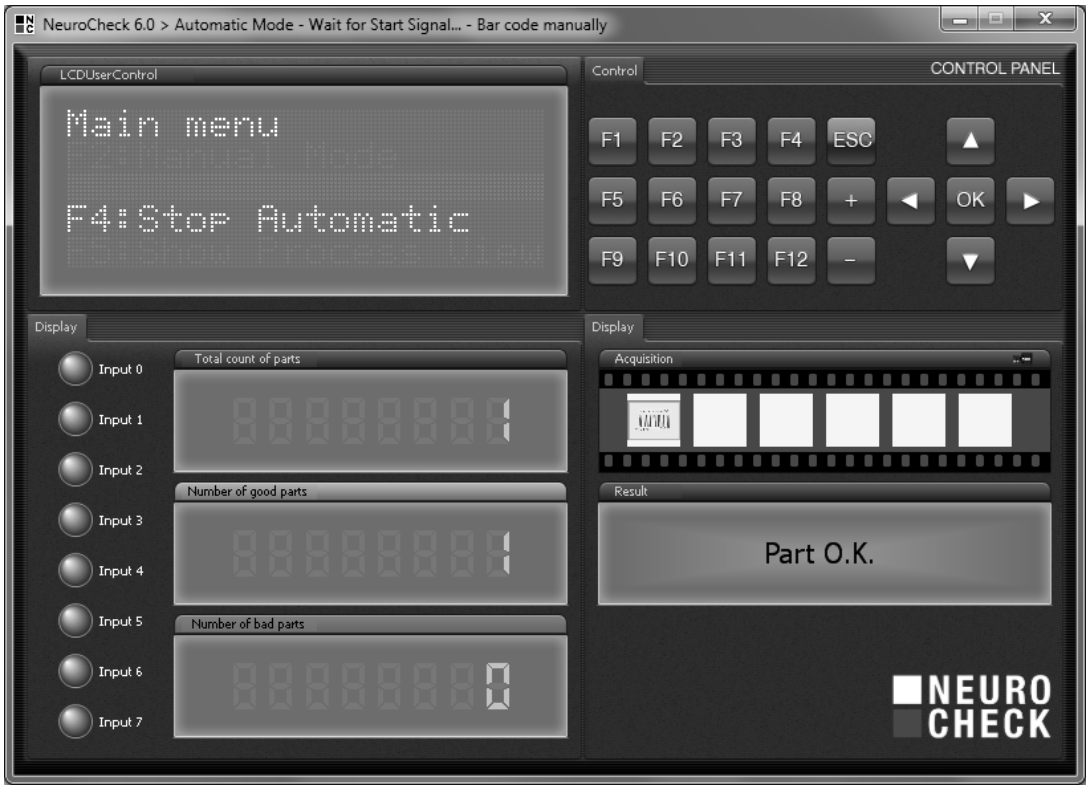


6 The **Process View** visualizes the inspection results in automatic mode.



7 Do not cancel automatic mode by pressing **No**.

8. Now press the + key on your numeric keypad. You have now executed an automatic visual inspection for the first time. The result of the inspection is displayed on the bottom right of the **Control Panel**: "Part OK" against a green background. The left side of the **Control Panel** displays "1" for the *Total count of parts* and the number of good parts. Above the result you can see the image from which the bar code was read in the film strip.

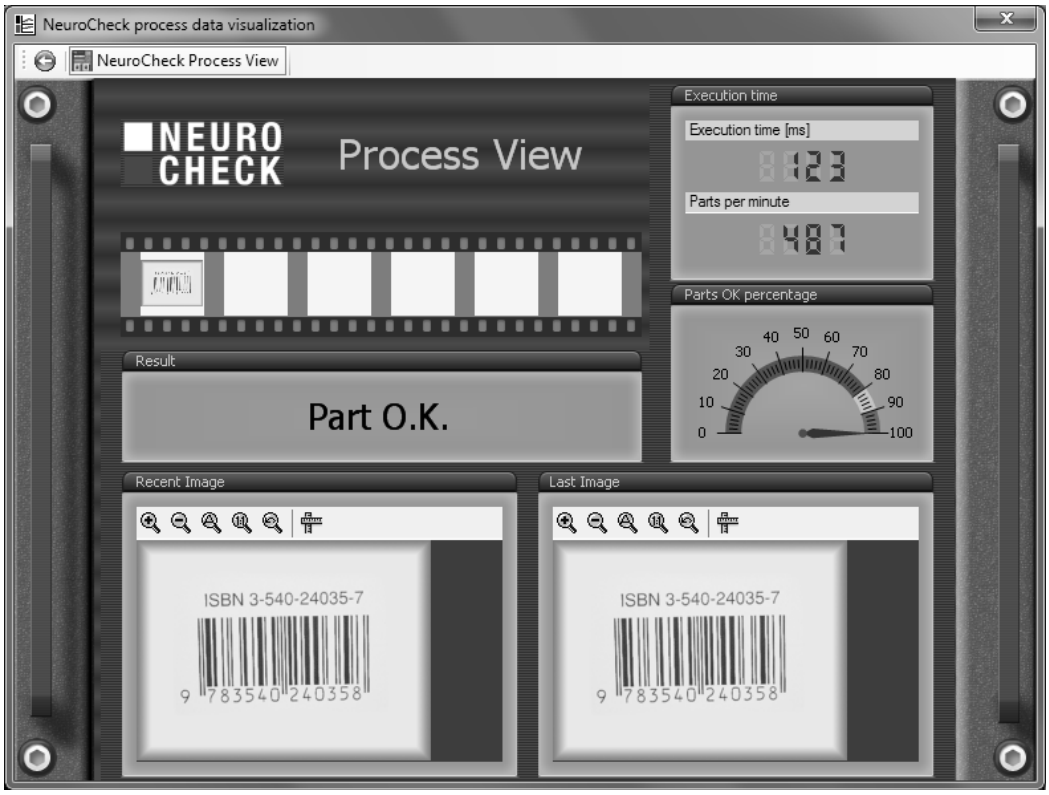


8 By pressing the [+] key on the numeric key pad, NeuroCheck is executed once.

❾ In **Process View** you can also see the result "Part O.K." and the image, from which the bar code was read. The digital display on the top right displays the time needed for execution and the scale of the pointer display shows the percentage of good parts with regard to the total number of parts.

In a real application the start signal would be given externally, e.g. by a PLC, whereto NeuroCheck would also send the result data, but apart from this there is no significant difference to the procedure as we have seen it just now.

Switch back to manual mode now. For this, make the **Control Panel** the active window. To switch from automatic to manual mode, automatic mode has to be stopped. For this press the F4 key or click the *F4* button or the text *F4: Stop Automatic* in the LCD matrix. Once automatic mode has been stopped, NeuroCheck does not respond to the start signal any longer. By pressing the F2 key or clicking the *F2* button or the text *F2: Manual mode* you can switch back to manual mode.



❾ Process view displays a visualization of the check routine result.

3. Configuration of Image Capture

In this chapter you'll learn how to

- integrate a digital camera into NeuroCheck
- display a camera's live image
- configure the camera settings
- capture an image from a check routine and access the image data in the check routine
- transfer one camera's configuration to another one of the same type
- create and use reference images.

3.1 Integrating digital cameras

Goal

You will capture camera images using NeuroCheck.

Result

You will be able to integrate digital cameras into NeuroCheck.

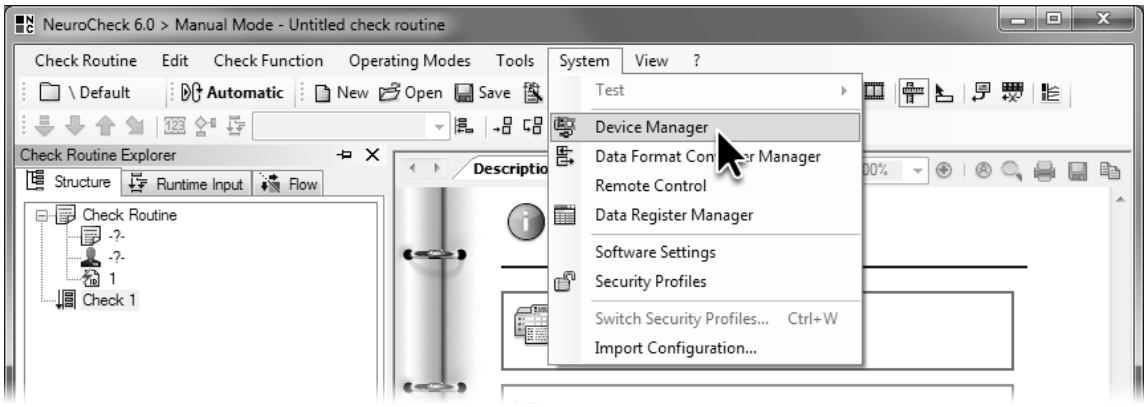
Solution

Before the camera can be integrated into NeuroCheck, it must be connected to the PC. If you are using IEEE1394 cameras, it may be necessary to insert an adapter card into the PC. In addition, Windows drivers for adapter cards need to be installed as well as Windows drivers for the digital cameras that are connected via bus systems such as IEEE1394 .

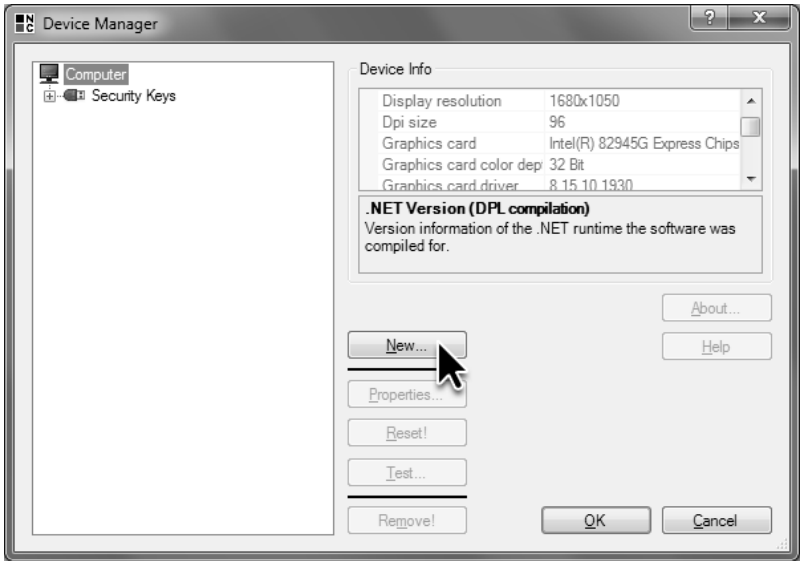
The Windows drivers are provided by the manufacturer either together with the hardware or as a download. For detailed information, please refer to the help file of the driver.

1. Having connected the camera and installed all Windows drivers, the new component must be integrated into NeuroCheck. For this select **Device Manager** from the **System** menu.

2. The **Device Manager** is the central place for configuring and registering new hardware for NeuroCheck. Select **New…** in the **Device Manager** to register new hardware.

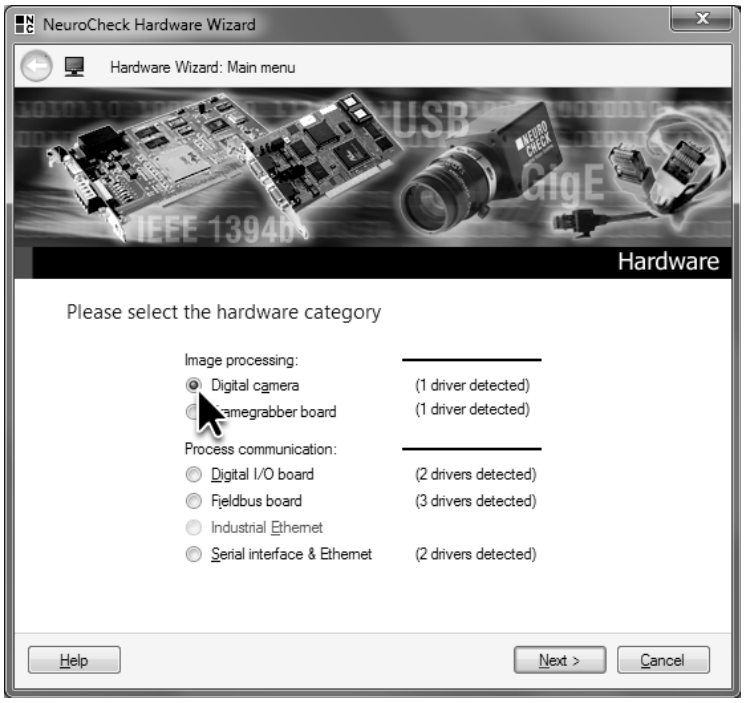


1. Select **Device Manager** from the **System** menu to integrate a new device.



2. Click **New…**

❸ The **Hardware Wizard** will guide you through the configuration process. On the first page, select **Digital camera** as new hardware. Clicking the **Next** button will take you to the next page.



❸ The **Hardware Wizard** will guide you through the process of installing new devices. Select the **Digital camera** group.

4. Expand the **Driver type** list and select the driver suitable for your camera manufacturer and type. Click **Next** to switch to the next page.

5. The list of available devices displays all cameras supported by the selected driver, connected to the PC and that have been detected by the driver. Finding the cameras can take a couple of moments which is why the list is compiled with slight delay.

If your camera is not listed, click **Rescan** or review the previous wizard page whether you have selected the correct driver. Click the arrow in the upper left corner of the wizard to switch to the previous page.

Enable the check box in front of the camera or cameras you wish to integrate. Exit the wizard by clicking **Finish**.

4 Select the driver suitable for your camera.

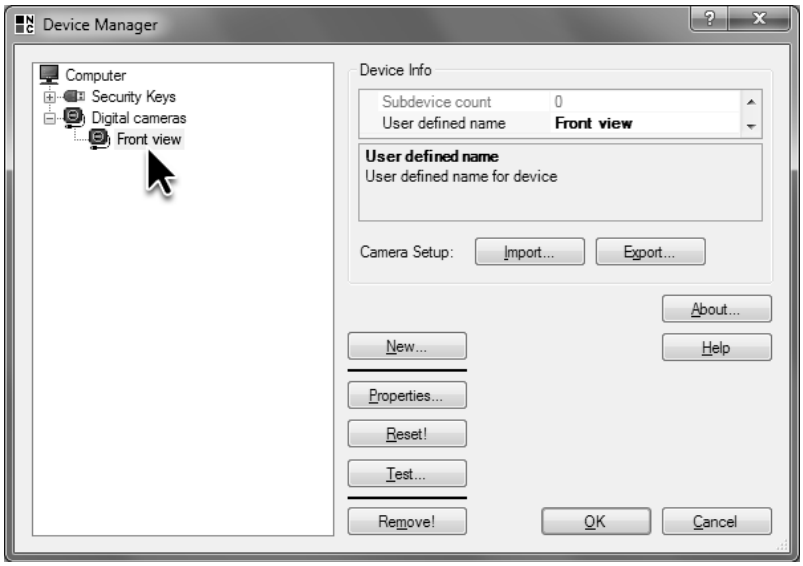
5 Select the camera(s) you wish to integrate.

6 The integrated cameras are automatically inserted into the tree view. In the tree view all integrated devices are displayed according to type.

The name of the entry in the tree view is also the name of the camera that will be used throughout the entire program. You can change the name by in-place editing in the tree view: Left-click the camera name twice (be sure not to double-click). Then enter a new name, e.g. "Top view". From now on NeuroCheck will refer to this camera by this name.

You can also change the camera name in the **Device Info** list. For this go to the entry **User defined name**. Click in the right column of this row for the cursor to appear.

Exit the device manager by clicking **OK**. The icon depicted will appear in the NeuroCheck status bar showing that a camera for image capturing is available.



6 The integrated camera is displayed in the tree view. You can change the name of the camera.

3.2 Live image and camera configuration

Goal

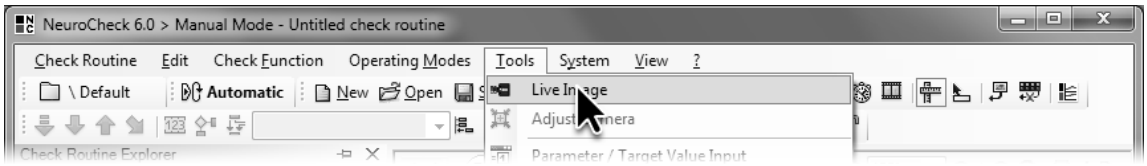
You want to review and configure lens and camera settings.

Result

You will know how to display live camera images and how to configure camera settings.

Solution

1. If one or more cameras are integrated into NeuroCheck, the **Live Image** dialog is available. To open the dialog, select **Live Image** from the **Tools** menu or click the depicted icon on the icon bar.



1. Select **Live Image** from the **Tools** menu to display the current camera image.

❷ The **Live Image** dialog contains an image window for displaying the current camera image. To display the images of other cameras, you can open additional image windows from the menu of the **Live Image** dialog. In the menu you'll also find options to display several image windows on the screen automatically.

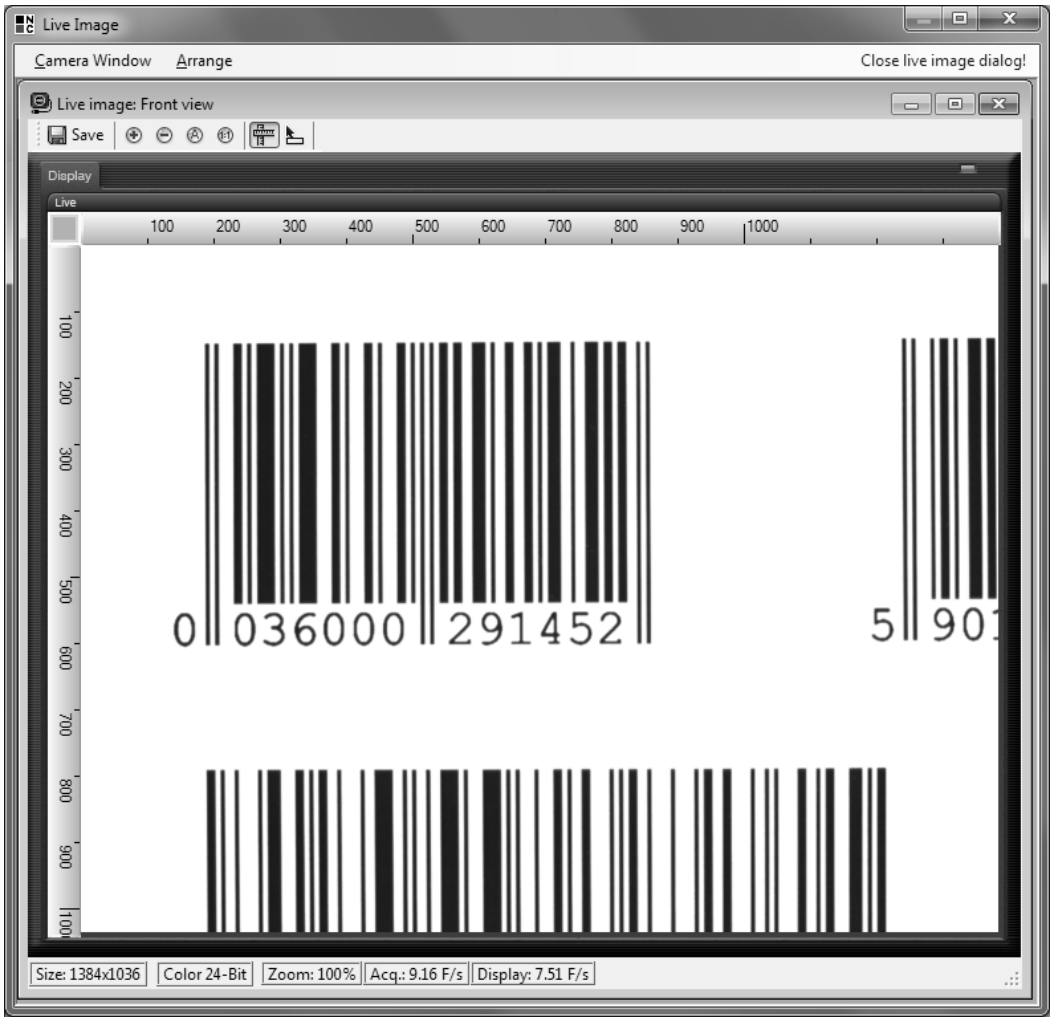
During the display of the camera image in the **Live Image** dialog you can check the camera position and lens settings such as aperture and focus.

Using the control elements on the top border of the image window, you can change the image's zoom factor. 100% zoom is best for setting the camera's focus but to get an overall impression, it is better to use smaller zoom factors especially with high-resolution cameras.

Maximize the **Live Image** dialog to use the entire screen for displaying the camera images. You can also maximize the image window to utilize the entire **Live Image** dialog for displaying the camera image.

As long as the **Live Image** dialog is open, the camera captures images continuously and transfers them to the PC. Depending on the system configuration, this may consume a considerable portion of computational power and, in the process, slow down other operations such as response to commands.

Use the **Save** button to save the current camera image, e.g. to serve as a basis for a check routine creation.



❷ The **Live Image** dialog displays the current camera images. Use the menu buttons to set the zoom and save the current image.

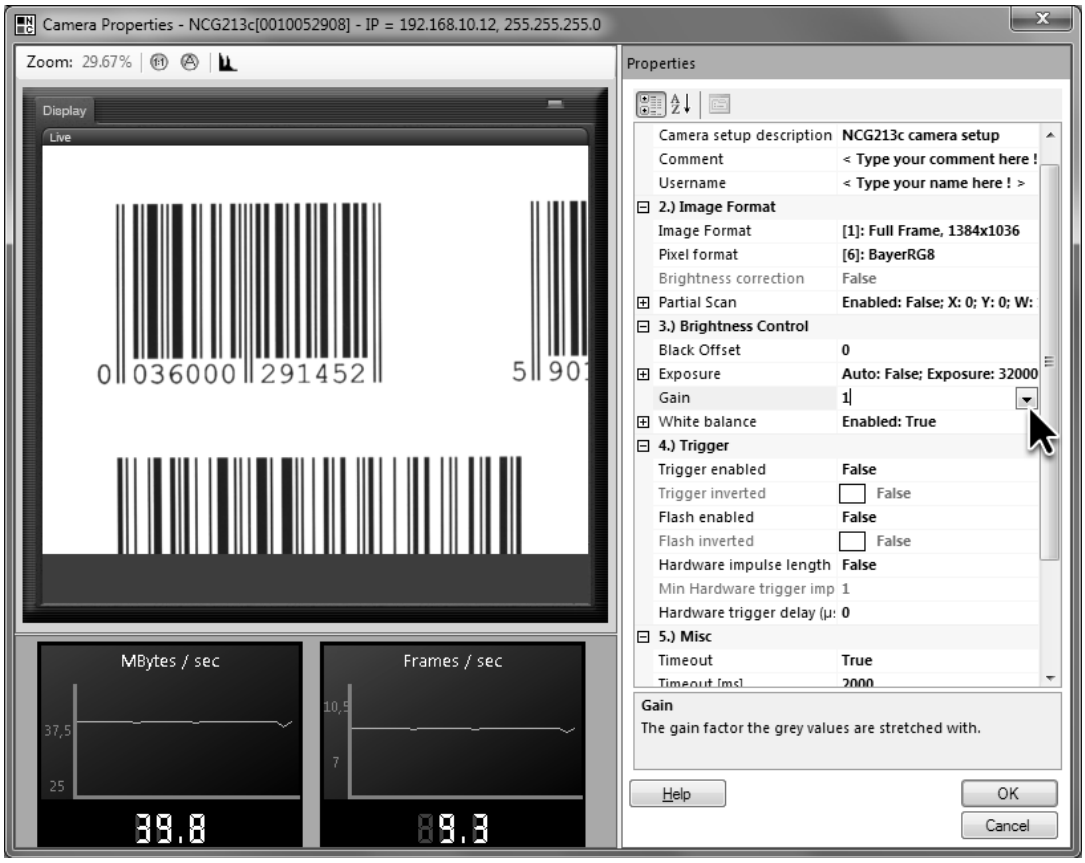
❸

Camera settings such as exposure time, image section or trigger source can be configured in the **Device Manager**. For this exit the **Live Image** dialog by clicking the **Close Live Image Dialog** button and re-open the **Device Manager** from the **System** menu. Select the camera whose properties you want to change and click **Properties**.

The camera's **Properties Dialog** is part of the camera driver. If you're not using NeuroCheck cameras with their drivers, the dialog displayed can vary significantly from the one described here for NeuroCheck cameras.

The **Properties Dialog** shows the current camera settings in a table on the right side. This is also where you can configure individual parameters. On the left side a live image is displayed so you can see the effects of the settings.

Simple parameters such as gain can be edited directly in the table. For this click the right column of the row **Gain**. Behind the current value the cursor will now be blinking and you can enter the new value. On the far right you will see the depicted icon. Click this icon to open a setup window.



❸ Click the right column of the row **Gain** once to change the setting. Enter the value directly or click the icon at the end of the row to open a setup window.

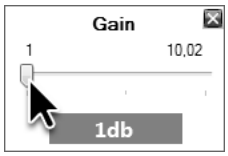
4

Use the slider in the setup box to configure the gain. Set the slider to 6. To apply the value, click once outside the window. To cancel, click on the depicted icon. Because you have increased the gain, the live image is now much brighter. Maybe you can see the noise which also increased. Reset gain to 1 to capture images with as little noise as possible.

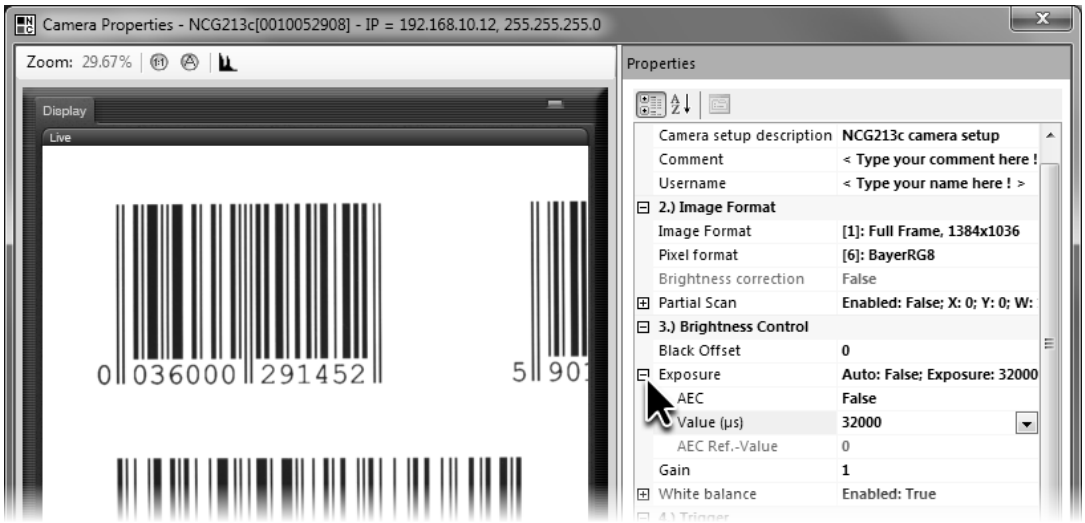
5

To set the exposure time you'll have to expand the **Exposure** parameter by clicking the "plus" icon. As with gain, In the row **Value (µs)** you can enter exposure time either directly or by using the slider in the setup window. Set the exposure to capture images with sufficient brightness.

Exit the camera's **Properties Dialog** by clicking **OK** and also exit the **Device manager** by clicking **OK**. The new camera settings are now saved to the NeuroCheck configuration. This ensures that the camera is started with exactly the same settings every time you start NeuroCheck.



4 You can adjust **Gain** by moving the slider in the setup window.



5 Click the ⊞ icon in front of **Exposure** to set exposure time.

3.3 Image capturing in the check routine

Goal

You want to read the bar code captured by a camera with a check routine.

Result

You will be able to configure checks that use a camera image.

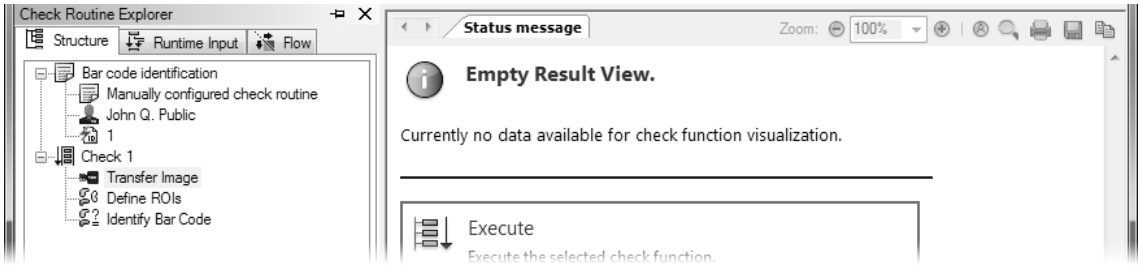
Solution

In the project subdirectory `\Check Routines\Images\` you'll find the image file `Barcodes.png`. This image can be used for image capturing in this section if you'll print it.

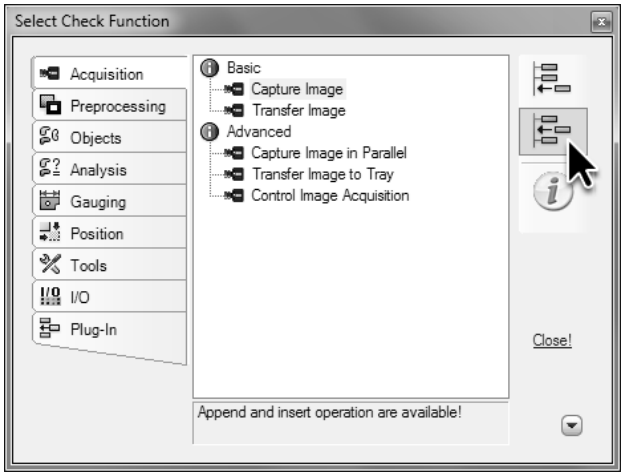
1. Open the check routine `Barcode manually` created in chapter 2 (*Working with Check Routines* and save it as `Barcode from camera`. Select the check function **Transfer Image**.

2. Open the **Select Check Function** dialog and insert the check function **Capture Image** from the **Acquisition** group by clicking the depicted symbol before the selected check function. The check function **Capture Image** triggers image capture by the camera.

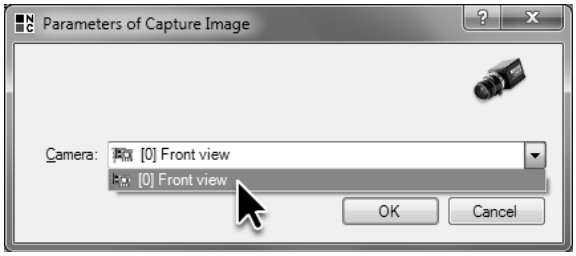
3. Open the parameter dialog of the check function **Capture Image** to select the camera you want to use for capturing the image.



1. Select the check function **Transfer Image**.



2. Insert the check function **Capture Image** prior to the selected check function.



3. Select the camera you want to use for capturing the image.

4 Open the parameter dialog of the check function **Transfer Image**. Enable the radio button **Camera** so the check function can access the images captured by the camera. Select the camera whose images you want to transfer. Please note, you'll have to execute **Capture Image** prior to executing **Transfer Image**.

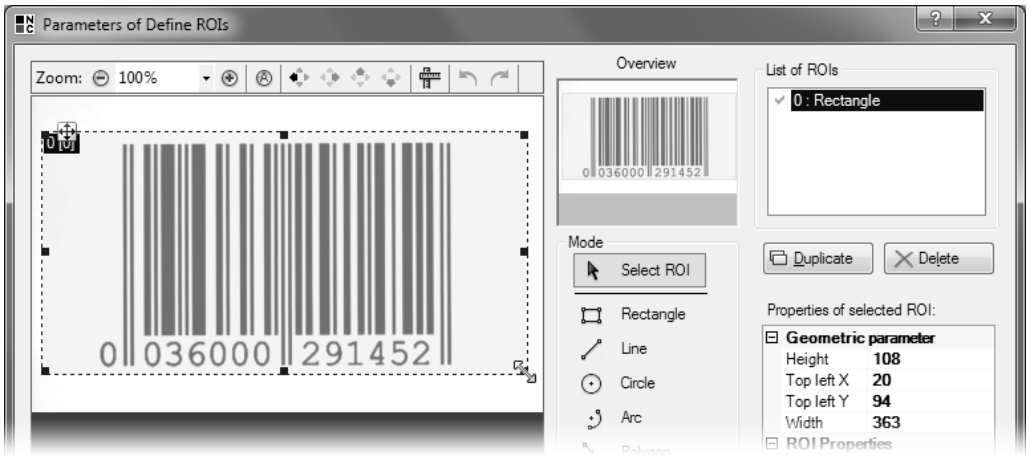
Use the **New Image** button to trigger capturing a new image from the selected camera. If you enable the **Live** check box, the camera will capture images continuously and they will be displayed continuously in this dialog.

You could use, for example, the images of bar codes printed on paper.

5 Since the size of the image section and the location of the bar code within the image section have probably changed when using the camera, the location of the ROI will have to be adjusted. Open the parameter dialog of the check function **Define ROIs**.

Select the ROI by clicking. If you drag one of the ROI's handles while keeping the left mouse button pressed, you can change the size and position of the ROI in such a way that the bar code is well covered.

4 Set the **Camera** as image source and adjust the image section accordingly.



5 Set the ROI in such a way that the bar code is completely covered and there are quiet zones before and after the bar code.

6 Execute the check. If you have used the bar code on these pictures printed on paper, NeuroCheck will read the code without further parameter changes. Of course the check function will report a target value failure since the bar code does not correspond to the target value set. For a different bar code you may have to adjust the check function **Identify Bar Code** to the code type.

6 Execute the check. NeuroCheck will read the bar code without further parameter changes.

3.4 Exchange configurations of cameras

Goal

You want to transfer the configuration of one camera to another camera of the same type, e.g. to parameterize two cameras of the same line in an identical way or to replace a faulty camera.

Result

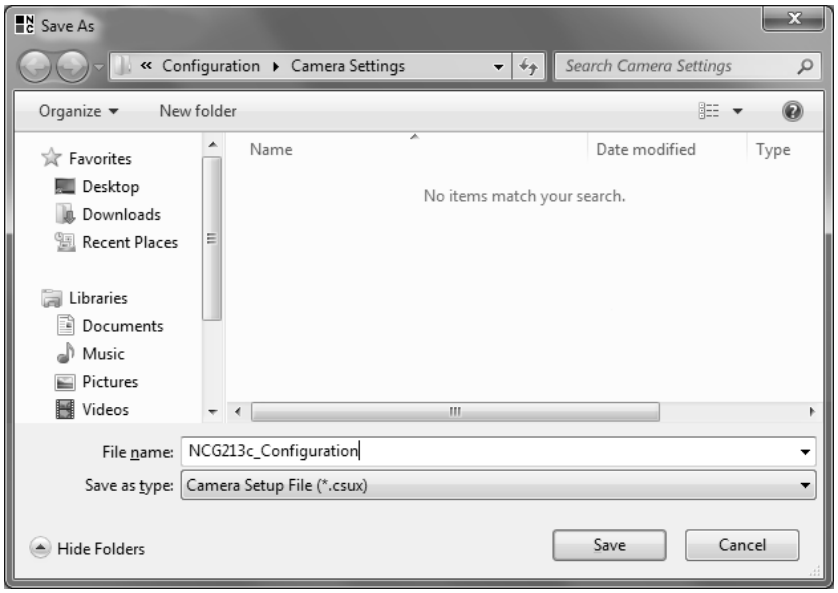
You will be able to export and import camera configurations.

To be able to export the camera configuration is a functionality of the camera driver. If you are not using NeuroCheck cameras, the functionality described here may be partially available or not at all.

Solution

1. Open the **Device Manager** and select the camera. Click the **Export…** button.
2. In the **Save as** dialog enter a suitable file name and save the configuration. NeuroCheck will propose the sub folder `\Configuration\Camera Settings\` of the current project directory.

1. Use the **Export...** button to save the current camera settings.

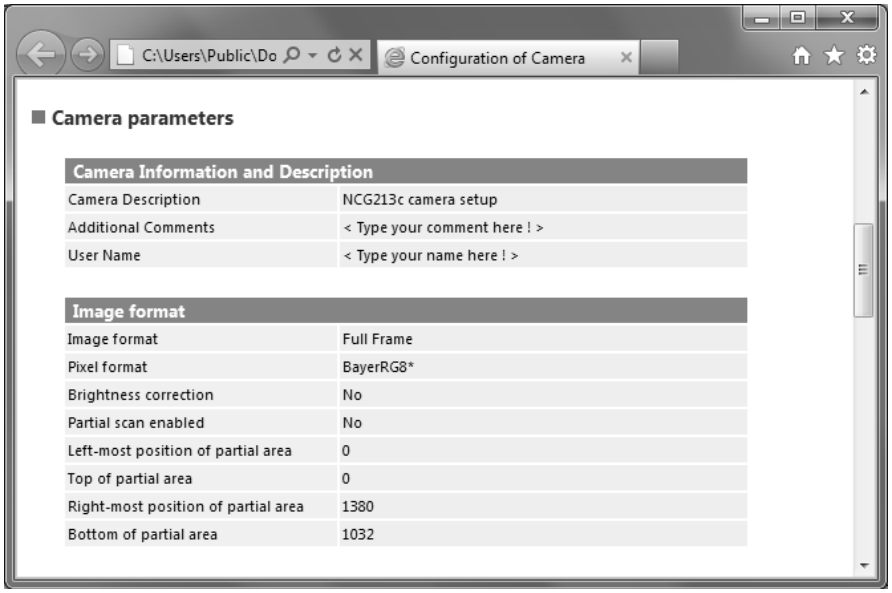


2. Select the directory and name where you want to save the camera settings to.

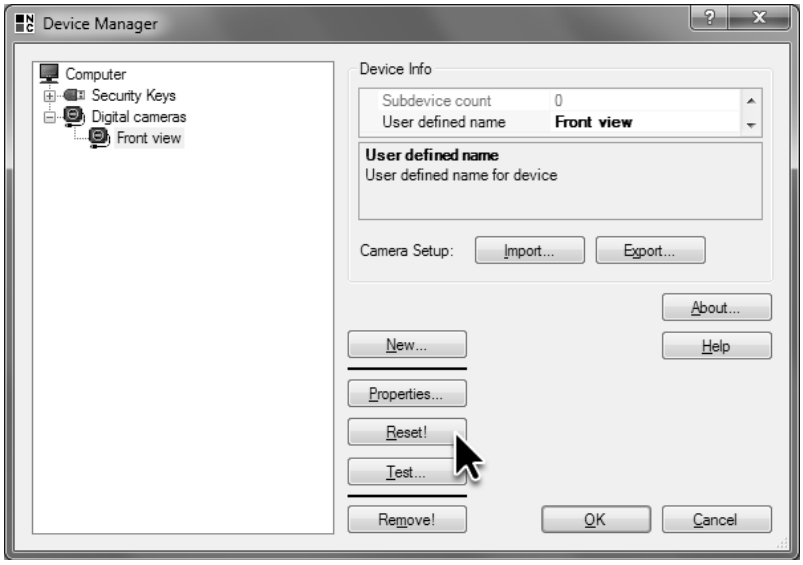
3. The driver of the NeuroCheck cameras will save the configuration file as a CSUX file (camera set up in XML format). Since this is an XML format, the file can be viewed with any browser , e.g. the Internet Explorer. Navigate with the Windows Explorer to the newly created CSUX file and open it by double-clicking.

   The file will then be displayed with your default browser. Exporting the configuration is thus an easy way to display it in a readable way. Please note that depending on your browser's settings, the display may vary.

4. To demonstrate the effects of exchanging the camera configuration, the camera will be reset to the default setting. For this go to the **Device Manager** and click **Reset!** Open the camera's properties dialog and make sure that the changes to your camera settings have been reversed. Then close the camera's properties dialog.



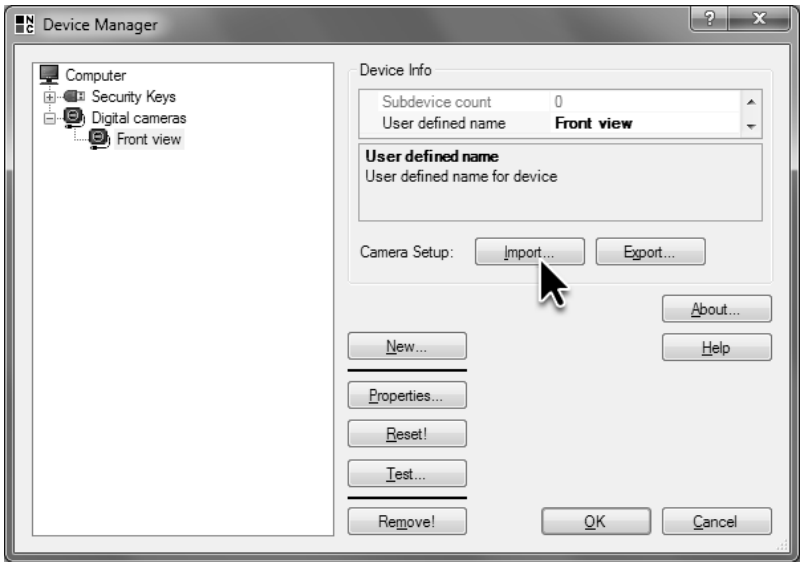
3 The camera's configuration can be displayed in a readable way in any browser.



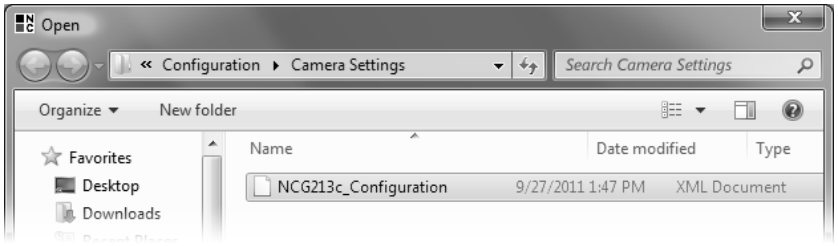
4 By clicking the **Reset!** button, the camera is reset to default.

5. To restore the camera settings, go to the **Device Manager** and click the **Import…** button.

6. In the **Open** dialog go to the configuration saved in step 2 and open it. The stored settings will automatically be transferred to the camera. Open the camera's properties dialog to make sure that the original configuration has been restored.



5 Use the **Import...** button to reload the saved camera setup.



6 Select the file containing the camera settings you want to use for the camera.

3.5 Camera reference images

Goal

After exchanging a faulty camera for a new one, you want to configure the new camera in the same way as the old one, or, if a camera's position was changed by accident, you want to restore the original setting.

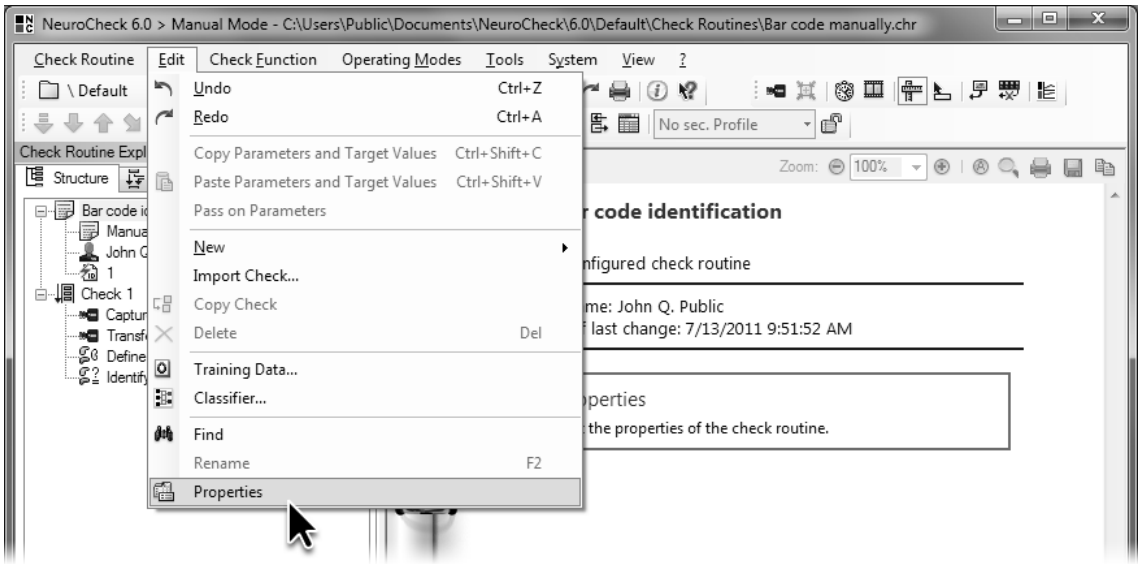
Result

You will be able to create, save and use reference images.

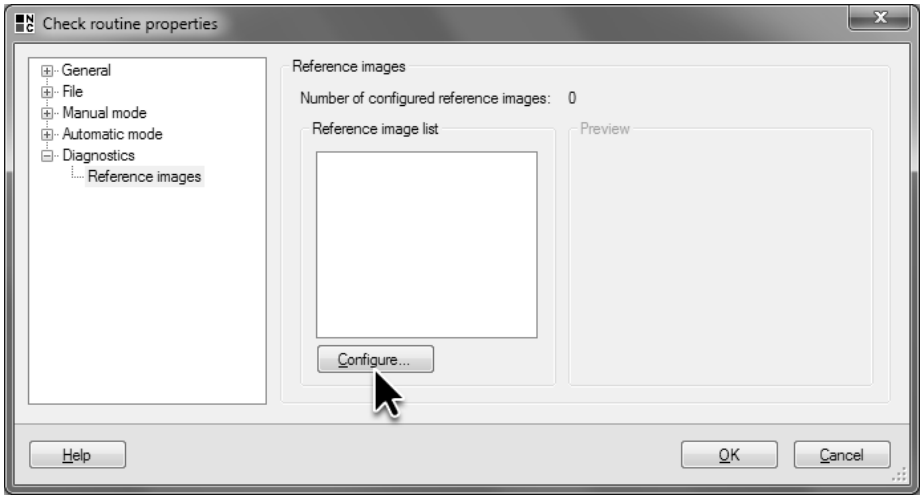
Solution

1. Select the uppermost row in the Check routine explorer. Click **Properties** in the **Edit** or context menu.

2. In the tree view select **Diagnostics ► Reference images**. This page shows a list of the reference images created for this check routine. Click the **Configure…** button to open the **Reference Image Editor** dialog.



❶ Select the topmost row in the **Check Routine Explorer** to open the check routine properties via the **Edit** menu.



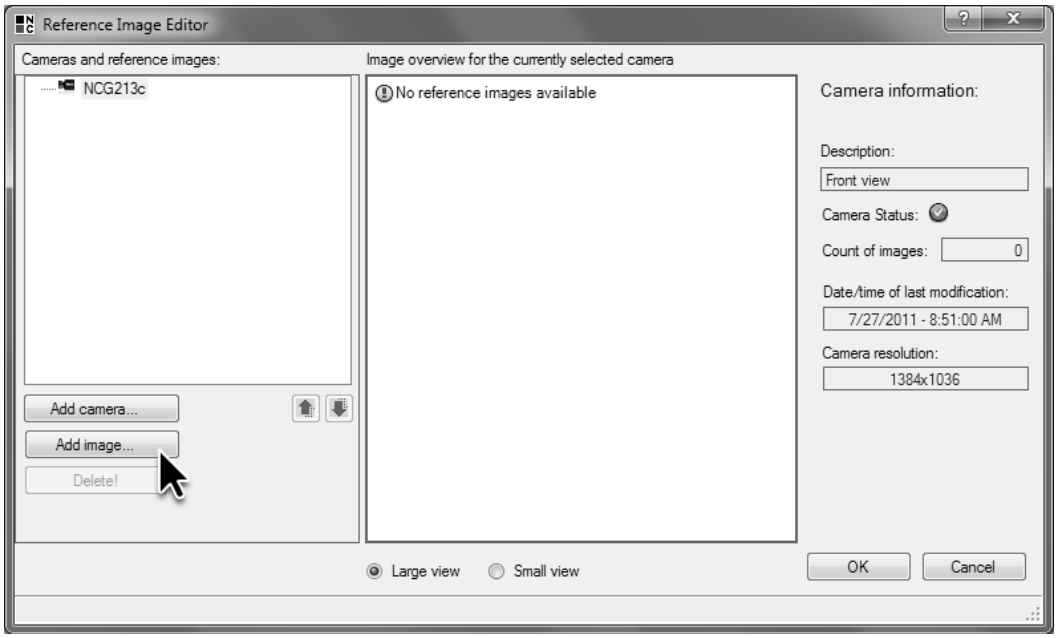
❷ The page **Diagnostics ► Reference images** displays a list of the saved reference images.

❸ The **Cameras and reference images** list in the **Reference Image Editor** dialog lists all cameras

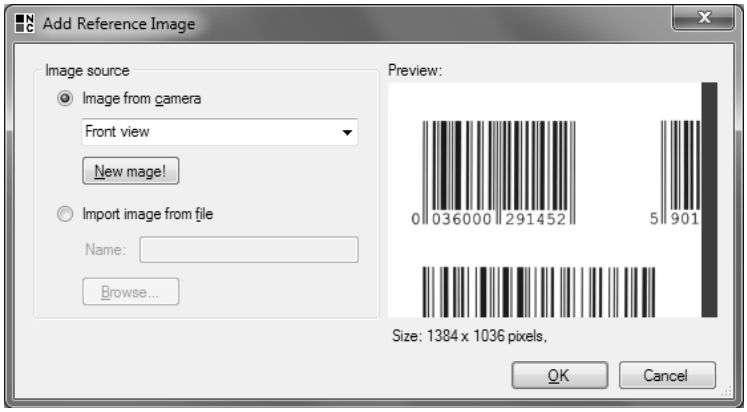
integrated in the system and their reference images. Select the camera you want to create a reference image for. Open the **Add Reference Image** dialog by clicking the **Add image…** button.

4. The **Add Reference Image** dialog shows the image captured by the selected camera when the dialog was opened. The selected camera is displayed in the list under the option **Image from camera**. Use the **New Image** button to trigger another image capture.

   Save the image displayed as reference image by closing the **Add Reference Image** dialog by clicking **OK**.



❸ The **Reference Image Editor** dialog displays the integrated cameras and their stored reference images.



❹ Click the **OK** button to save the displayed image as reference image.

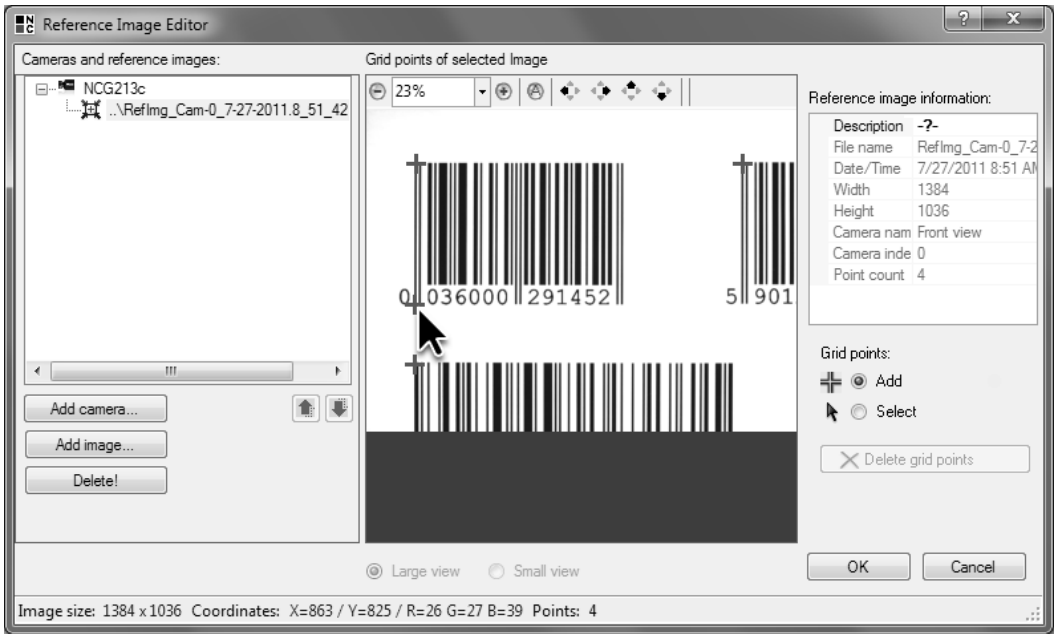
5. In the **Reference Image Editor** dialog you can now see the newly created reference image listed under the camera.

Select the newly created reference image to set optional markers in the image. Use the markers to pinpoint the position of distinctive features in the reference image. After exchanging cameras or changing camera positions, you can compare the position of distinctive image features using the markers in the reference image, thus getting information on how to adjust the camera position. Make sure that the **Add** option is enabled. Look for some distinctive features in the image. Mark these features by single clicks.

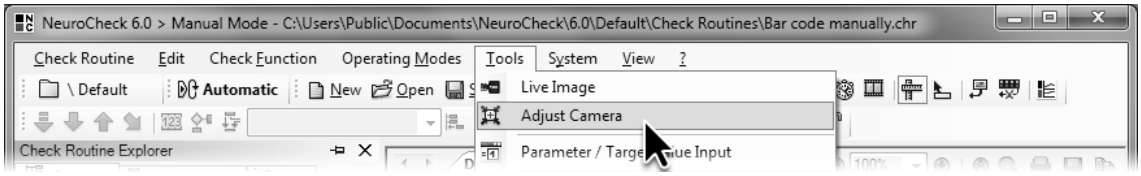
If you want to move one of the grid points, choose the **Select** option and drag the markers while keeping the mouse button pressed. To delete a marker, select it and click **Delete grid points**.

Exit the **Reference Image Editor** dialog and the properties dialog by clicking **OK**.

6. To simulate misalignment after an exchange of cameras, move the printout in front of the camera slightly. Then open the **Adjust Camera** dialog via **Adjust Camera** in the **Tools** menu to correct the misalignment.



5 Set the markers within the reference image that will help you to align the camera later on.



6 Click **Adjust Camera** in the **Tools** menu to open the **Adjust Camera** dialog.

7 In the **Adjust Camera** dialog you can see the reference image on the left side and the current camera live image on the right. Enlarge the dialog as much as possible and click on the depicted icon to view the reference and live image completely.

Both images show the markers. Take the reference image to see which marker corresponds to which position. In a real application you can adjust the camera to match the markers of the camera live image to the markers in the reference image. In this case we have to shift the print-out in front of the camera. When the markers in the reference and live image match, the camera is correctly adjusted.

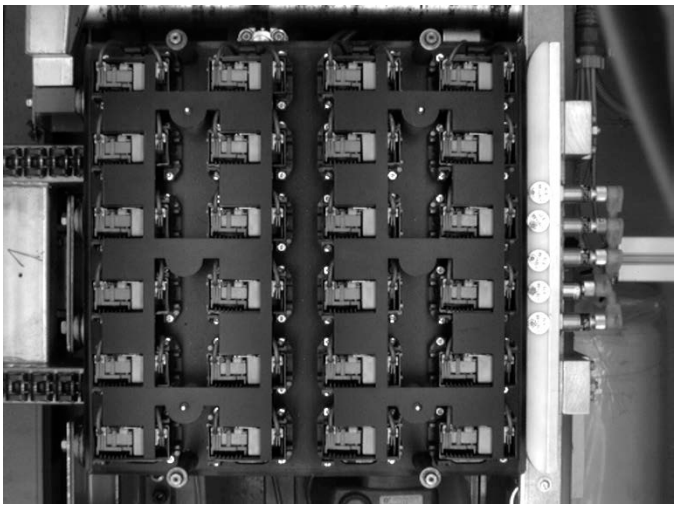
❼ Adjust the camera so that the markers in the camera live image and the reference image match.

4. Presence Verification

In many cases visual inspection has to answer the question whether certain parts of an inspection object are present in the correct number. To do a presence verification it is usually necessary to find all eligible objects in the image. These are then evaluated according to object characteristics to discern objects relevant for the counting.

In this chapter you'll learn how to use NeuroCheck

- to verify the presence of an object on the basis of light-dark-differences
- to create a hierarchic object search where objects within another object are searched
- to differentiate objects based on their features
- to localize objects using Template matching



4.1 Object presence verification

Goal

You want to make sure that no cover plate leaves your plant with too few punched holes.

Result

You will be able to use NeuroCheck to verify the presence of objects and you will know how to configure a preset number of objects.

Solution

1. Create a new check routine. As a description you might enter "Presence verification" and as an additional comment "Verifies the number of black holes". You can change the title and description of the check to "Cover plate Check 1" and "Check uses fixed rectangle" (in the next sub-section we will be coming back to this example with another check; then the title and description will make sense as well).

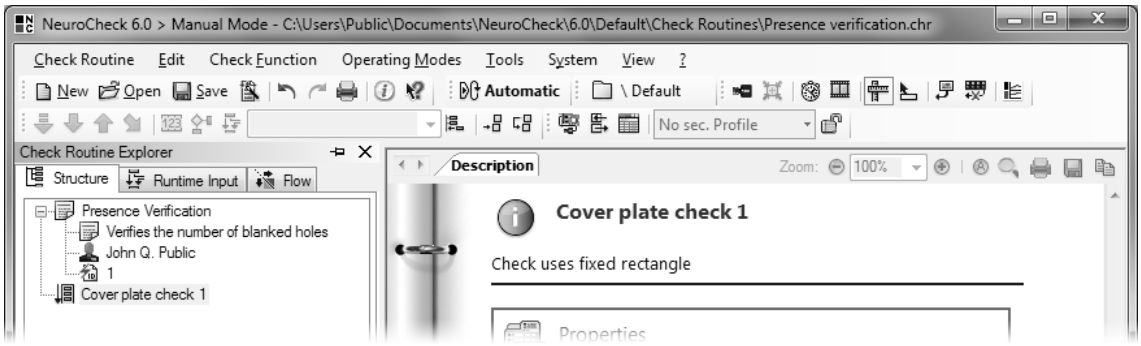
2. First, you have to add the check function **Transfer Image** to the check. In the parameter dialog of the check function, select the file `Cover_Plate_01_OK.bmp` and leave the check box **Always full image** enabled.

   If the selection dialog for the image does not open automatically to the directory into which this image - just like all other images used in this manual - was copied upon installation, you'll find the image in the project directory

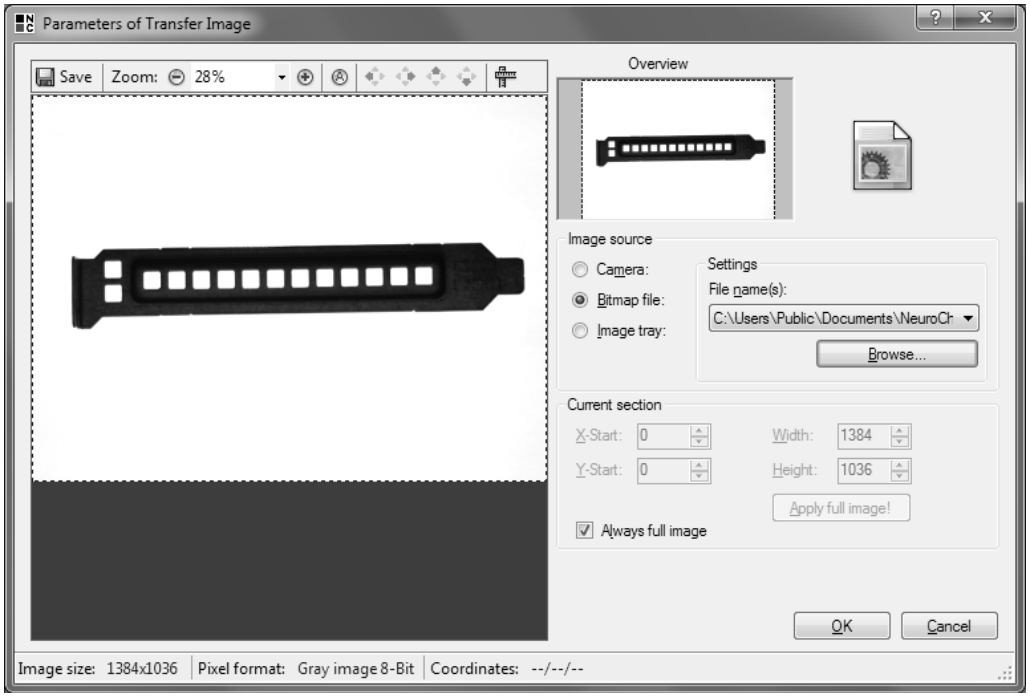
   `\Check Routines\Images\`.

   You'll also find this image and the other ones on the NeuroCheck installation DVD under

   `\Extras\Directories\NeuroCheck\6.1\Default\Check Routines\Images\`.



❶ Create a new check routine and enter a name and a description for the check routine and the check.

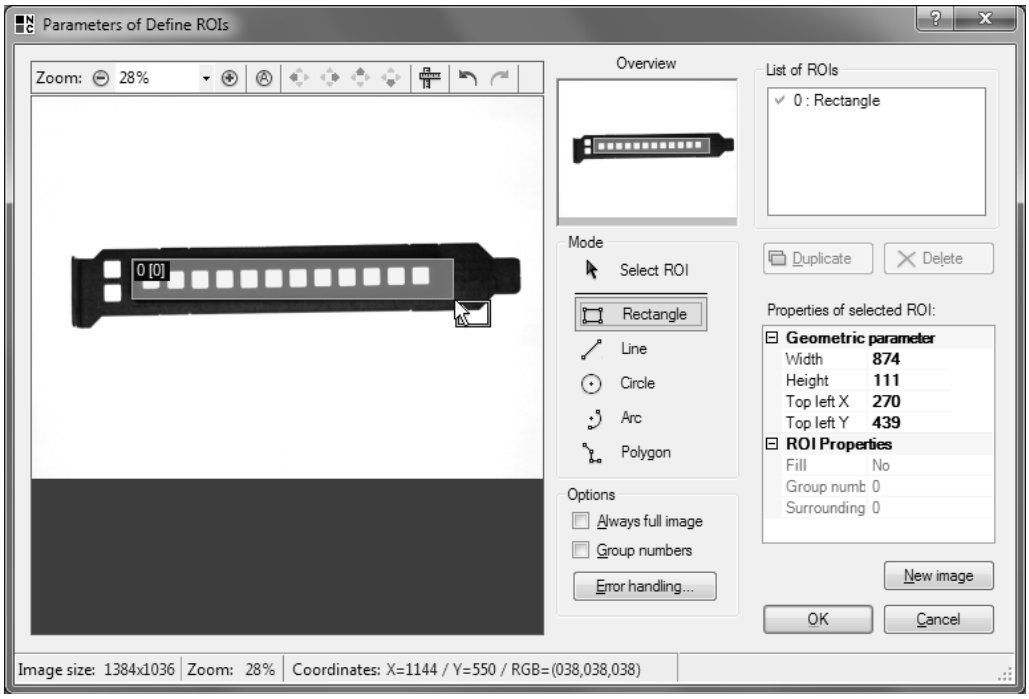


❷ Select the file `Cover_Plate_01_OK.bmp` as image.

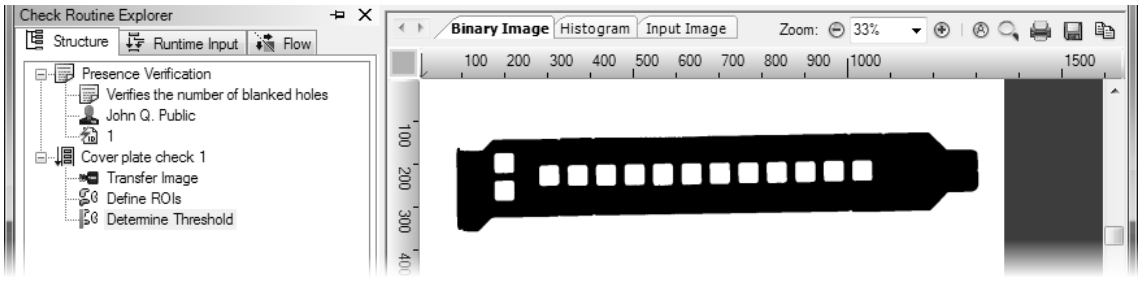
3. Add the check function **Define ROIs**. Open the parameter dialog and define a rectangular ROI enclosing all holes in the row without leaving the cover plate. The two holes on the left on top of each other we'll check later.

4. Add the check function **Determine Threshold**. This check function you'll find in the group **Objects**. This check function analyzes the brightness distribution in the defined ROI, and based upon this distribution the check function will determine the ideal threshold for differentiating object and background.

When you execute this check function, the image is depicted using black and white pixels, no gray values. This type of image is called a binary image.



❸ Define a rectangular ROI that is within the area of the cover plate and encloses all holes in a row.



❹ The check function **Determine Threshold** displays an image containing only black and white pixels.

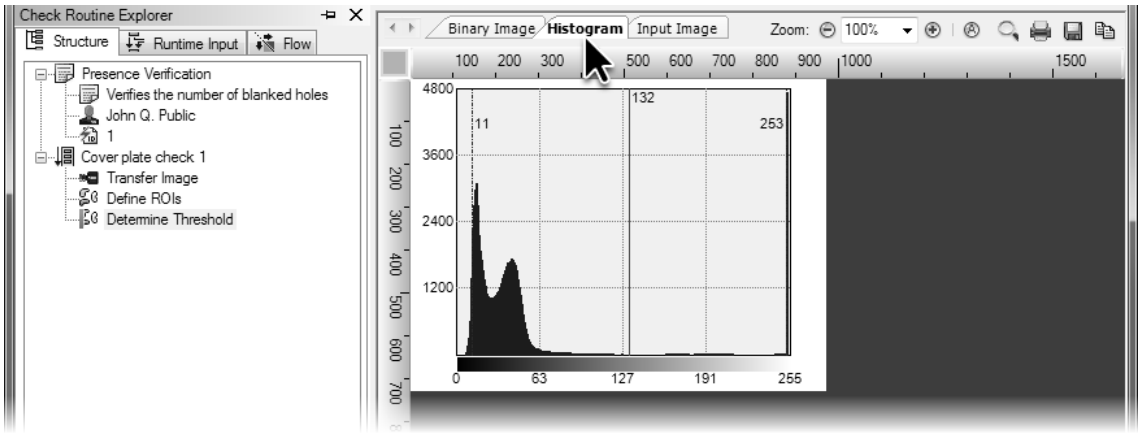
❺ You can see the gray value distribution in the **Histogram** tab. The image shows the gray value distribution of the ROI. The higher the bar for a gray value, the higher the number of pixels with this gray value. As you can see there are many gray values in the current ROI that are completely white with the gray value of 255. These are the pixels in the holes. Then there are many pixels with gray values between 10 and 60 making up the cover plate.

Three numbers are displayed in the histogram. The middle number represents the computed binary threshold. All pixels lighter than the binary threshold are depicted white in the binary image, all darker pixels are shown in black.

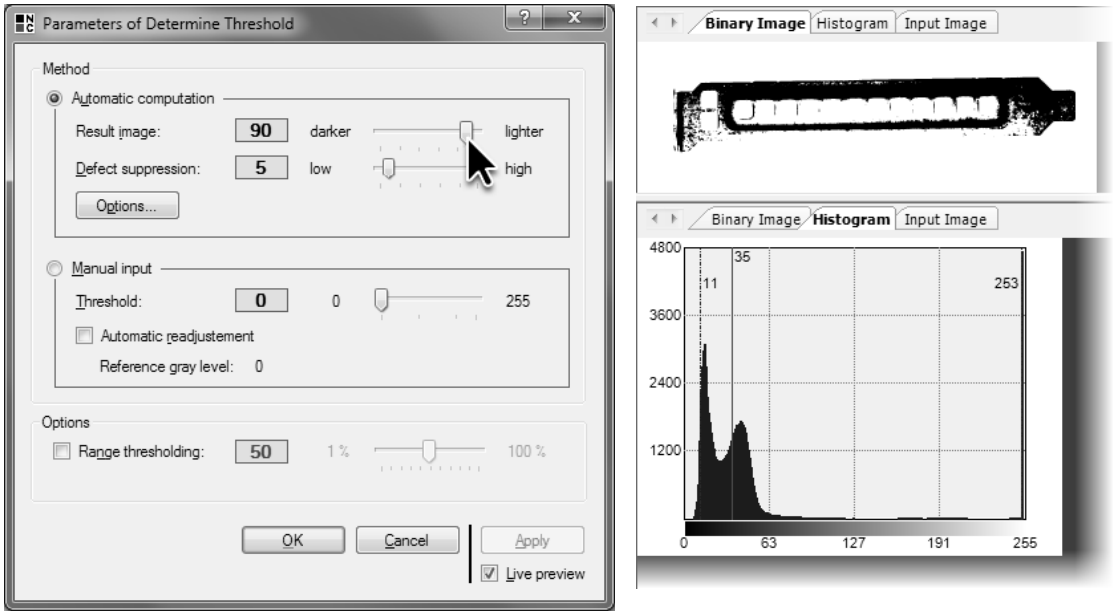
❻ As you can see in image ❹, the settings of this check function are ideally suited to compute a binary threshold for separating the holes and cover plate. To visualize the effect of the settings, we'll change them temporarily for the worse.

Split the Result view so you can see the results of **Binary image** and **Histogram** simultaneously. Adjust the zoom of the binary image so you can see the image completely. Open the **Determine Threshold** parameter dialog and enable the **Live preview** check box. Every time you change a parameter, the binary threshold and the histogram will now be updated accordingly.

The calculation of the binary threshold is based upon the gray value of the lightest and darkest pixel. The slider **Result Image** in the check function's parameter dialog sets at what percentage of the difference in gray value between the lightest and darkest pixel the binary threshold is to be placed. Therefore, a large value, as depicted, means that the binary threshold is lower and thus many pixels in the binary image are depicted white. In this example this has the effect that parts of the cover plate appear white rather than black.



5 The **Histogram** tab displays the gray value distribution within the ROI.

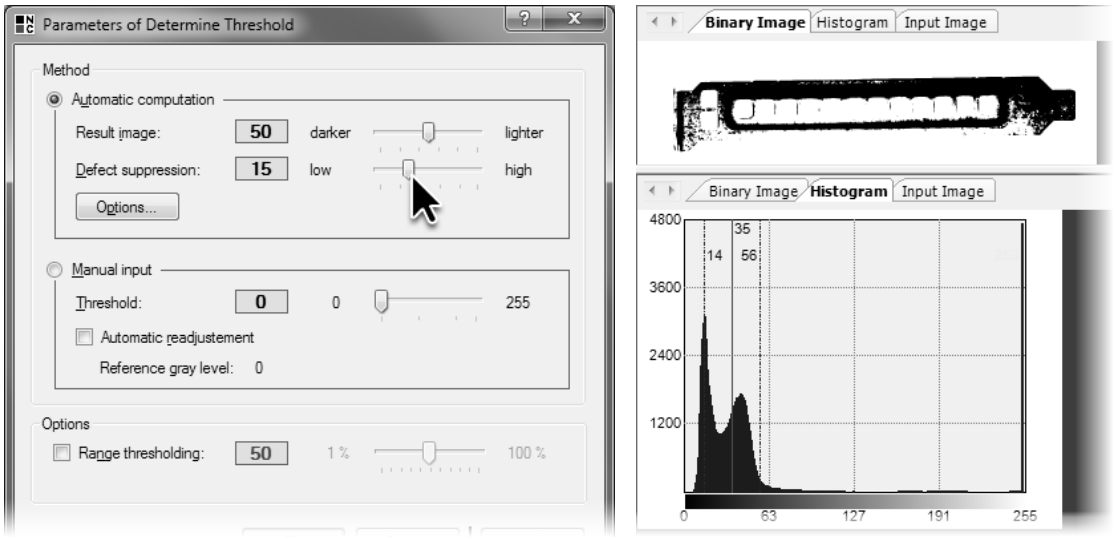


6 If the **Result Image** slider is set to a high value, many pixels in the binary image will be white.

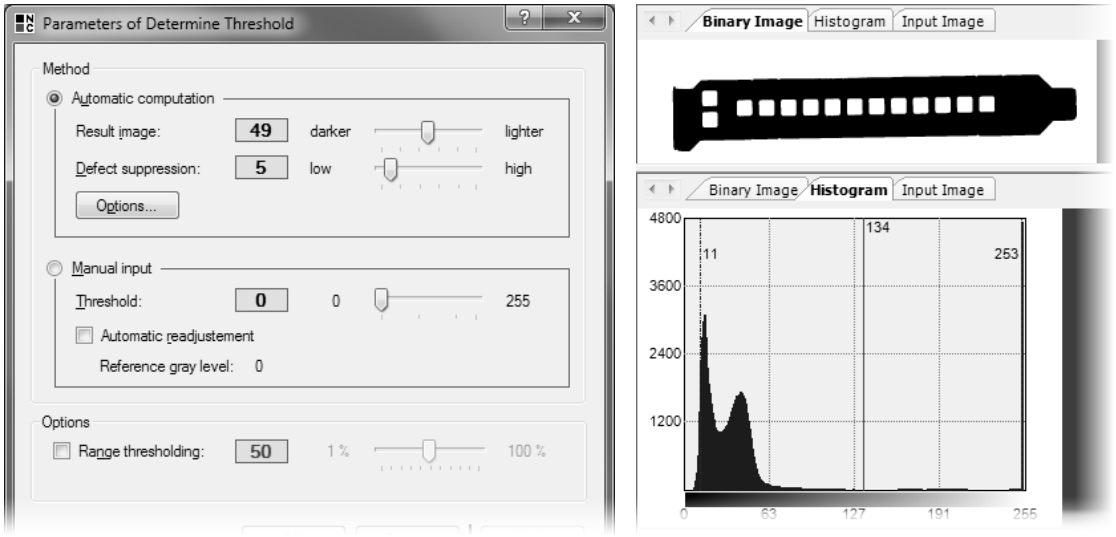
7. The **Defect suppression** slider determines how many extremely bright and extremely dark pixels are ignored when calculating the binary threshold. The default setting is 5 which means that the 5% brightest pixels and the 5% darkest pixels will be ignored for calculating the binary threshold. The gray value of the brightest pixel being used for calculating the binary threshold is depicted as the number on the right in the histogram. The gray value of the darkest pixel is shown accordingly on the left.

   Therefore, the effect of noise suppression on the binary threshold depends on the gray value distribution. In this case, increasing the value for the noise suppression will lower the binary threshold which will result in parts of the cover plate being depicted as white.

8. Set the **Result Image** and **Defect suppression** sliders so that the cover plate is black and the holes white in the binary image. Exit the parameter dialog by clicking **OK**.



❼ The **Defect suppression** sliders controls which pixels are used for calculating the binary threshold.



❽ The setting of the threshold is good when the entire cover plate is black in the binary image.

9 Add the **Create ROIs by Thresholding** check function from the **Objects** group and open the parameter dialog. The check function combines neighboring pixels of the same color in the binary image to objects. For objects to be created they need to be within the ROI and surrounded by the opposite binary color.

Accordingly, the most important setting of the check function is whether you are looking for white objects against a dark background or vice versa. In this case we are looking for holes that are white against a cover plate that is black. Please select **Object color Light** in the parameter dialog.

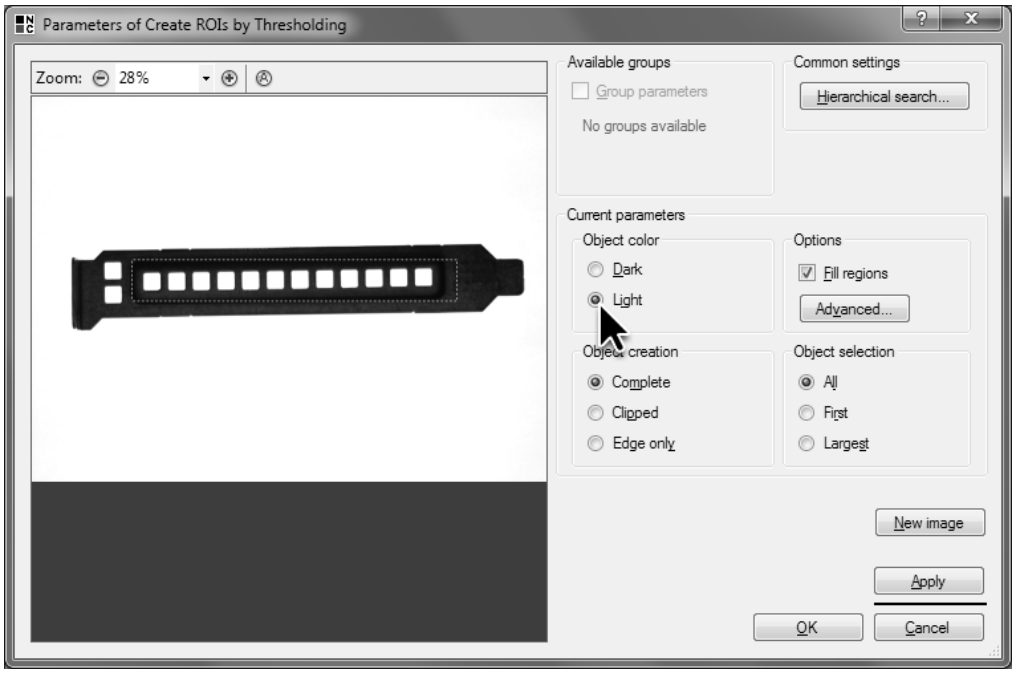
Exit the dialog by clicking **OK**.

10 When executing a check function you will notice that ROIs created by thresholding within the holes are now drawn in blue. If there are no ROIs being depicted, make sure that the **Thresholded ROI(s): Area** tab is displayed.

If there are still no ROIs, make sure that the ROI you configured in **Define ROIs** is completely within the cover plate. If not, the check function is looking for white objects against a white background which won't work. In this case change the ROI so it will be completely within the cover plate.

It is equally important for the holes to be within the ROI configured in **Define ROIs**. Since objects are only searched for within this ROI, no hole will be found if the holes are outside of the ROI.

Another reason for not finding any objects may be a not correctly computed binary threshold in **Determine Threshold**.

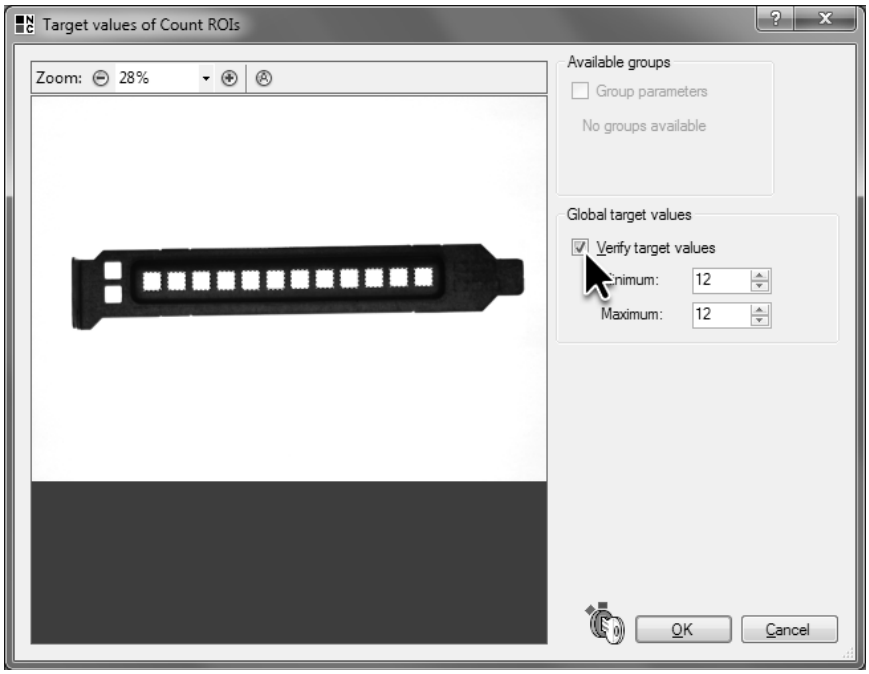


❾ In this case, the check function **Create ROIs by Thresholding** is looking for white objects, i.e. holes against a black background, i.e. the cover plate.

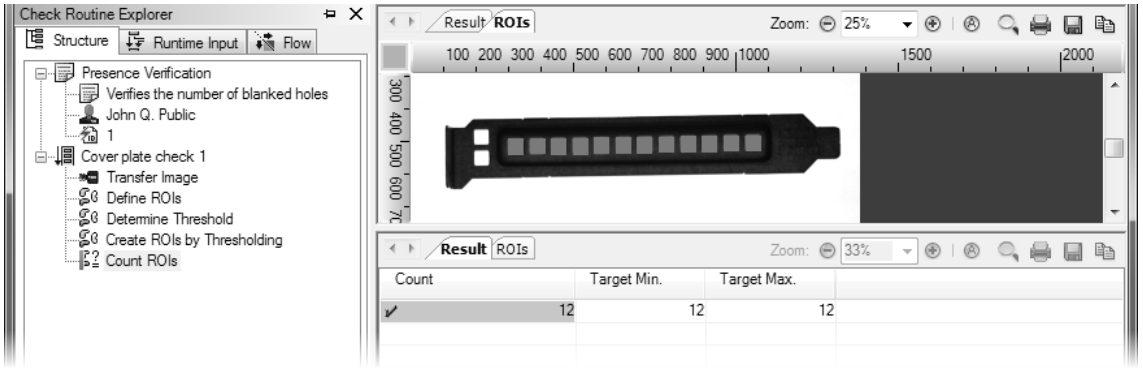
❿ The objects found are displayed in blue in the image.

11 Add the check function **Count ROIs** from the **Analysis** group to the check and open the **Target values** dialog. Enable the check box **Verify target values** and set the **Minimum** and **Maximum** to 12 each corresponding to the number of holes. Exit the dialog by clicking **OK**.

12 Now you can execute the check routine and split the Result View by dragging the window frame. In the **Result** tab the current count, the target minimum and target maximum are shown in a table. Since the current value is between minimum and maximum and therefore O.K. it is highlighted in green and bears a check mark. In the **ROIs** tabs the objects found are displayed in the image.

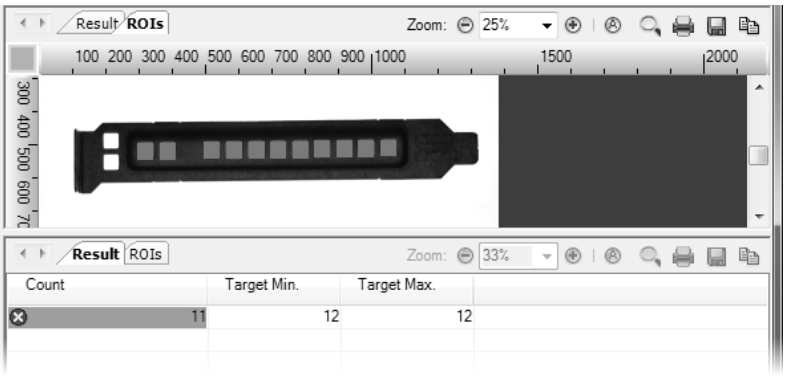


⓫ In the target value dialog of the check function **Count ROIs** you can define the minimum and maximum number of allowed objects, i.e. holes.



⓬ The Result View shows the current number of ROIs and the permissible minimum and maximum.

13 In the **Transfer Image** parameter dialog, if you select the image `Cover_Plate_02_NOK_Missing_Hole.bmp` and then execute the entire check again, the check function **Count ROIs** will report a target value failure since the cover plate in this image has only 11 holes. The current number in the Result View is highlighted in red and marked with a yellow cross.



13 For a Not OK part a target value failure will be reported since too few holes were found.

4.2 Presence verification using hierarchical search

Goal

You will be able to configure the inspection in such a way that variations in position of the cover plate will have no effects on the result.

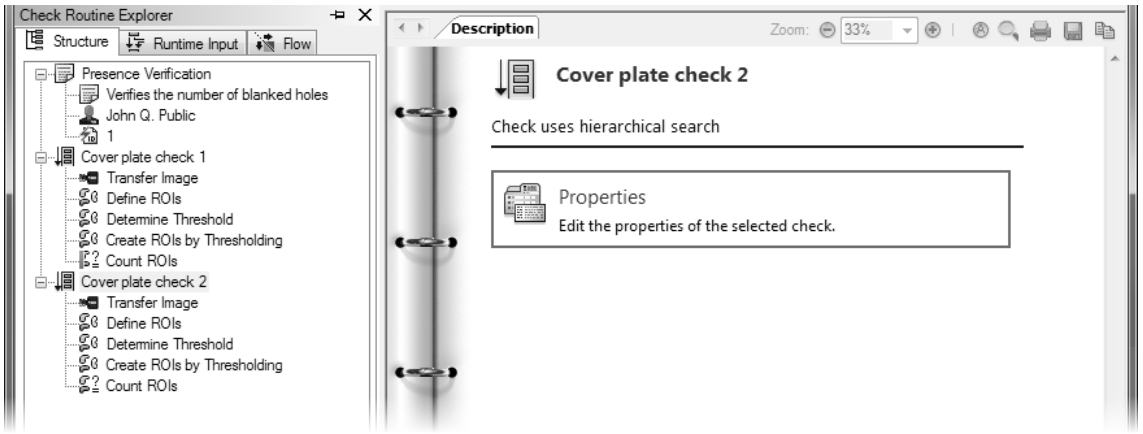
Result

You will know how to create a hierarchical object search in NeuroCheck.

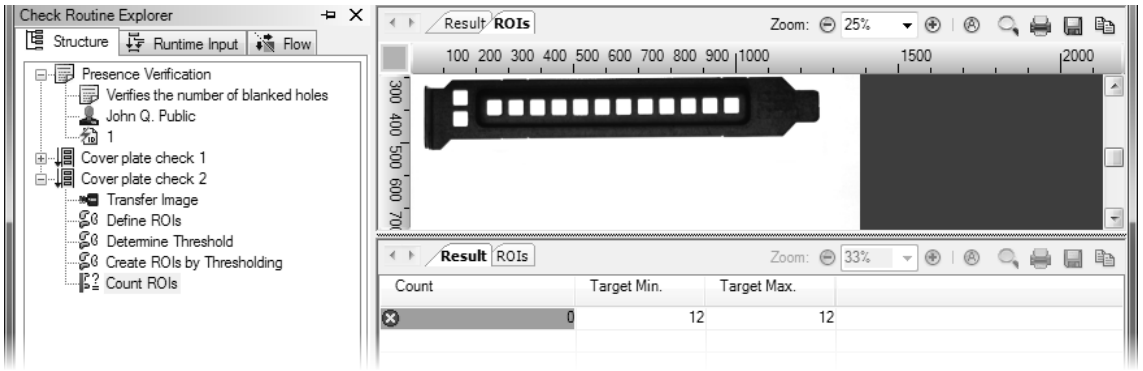
Solution

1.
The start of the procedure corresponds to the previous example. The easiest way therefore is to copy the check just created by clicking the depicted symbol from the NeuroCheck menu bar or select **Copy Check** from the **Edit** menu or context menu. Save the copied check as "Cover plate check 2" and change the description to "Check uses hierarchical search".

2.
Open the parameter dialog of the **Transfer Image** check function, click **Browse** and select the image `Cover_Plate_03_OK_Moved.bmp`. Execute the check function **Count ROIs**. Its result will be *not OK*. The check function **Determine Threshold** aborts with the message **Contrast fall below required minimum!** to begin with. So no ROI was found although the cover plate in the image has 12 holes in a row.



1. Copy the check and adjust the properties.



2. No holes are found in the image `Cover_Plate_03_OK_Moved.bmp`.

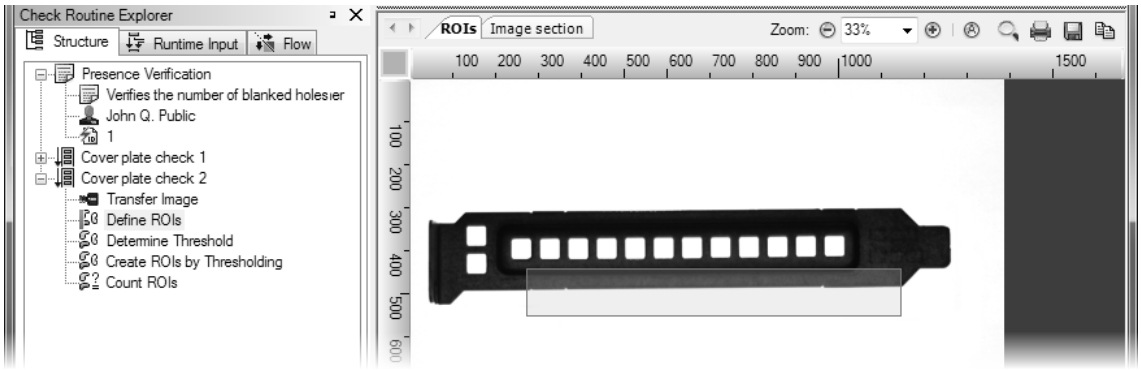
3 To see the cause for the not OK or error message, execute the check function **Define ROIs**. As you can see, the holes are outside the ROI and cannot be found. Of course you can adjust the ROIs manually, but that would not be sensible since you would have to do this again for every image. In the following steps we are going to look for a fundamental solution which is called hierarchical search.

Select the check function **Count ROIs** in the new check and delete it by pressing the Del key, or select **Delete** from the context menu or the **Edit** menu.

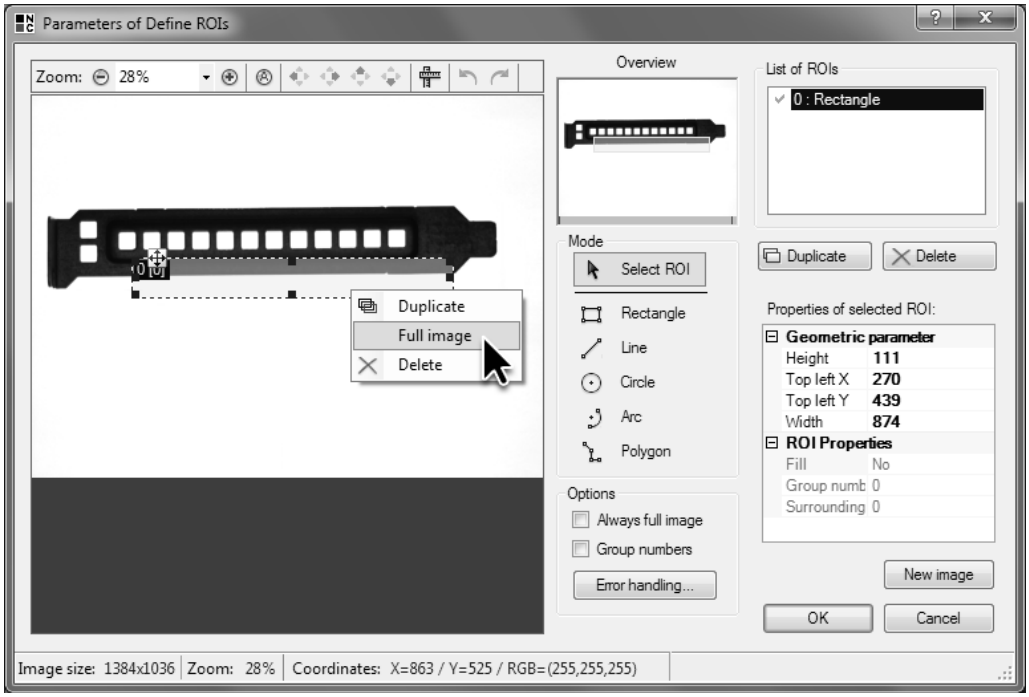
4 Open the parameter dialog of the **Define ROIs** check function.

123 To become independent of the cover plate's position in the image, we are going to look for the cover plate in the entire image first. In the second step we are going to look for all holes within the found cover plate.

For this the rectangular used for the first search must be enlarged to include the entire image. Therefore, select the rectangle and select **Full image** from the context menu of the rectangle. Exit the dialog by clicking **OK**.



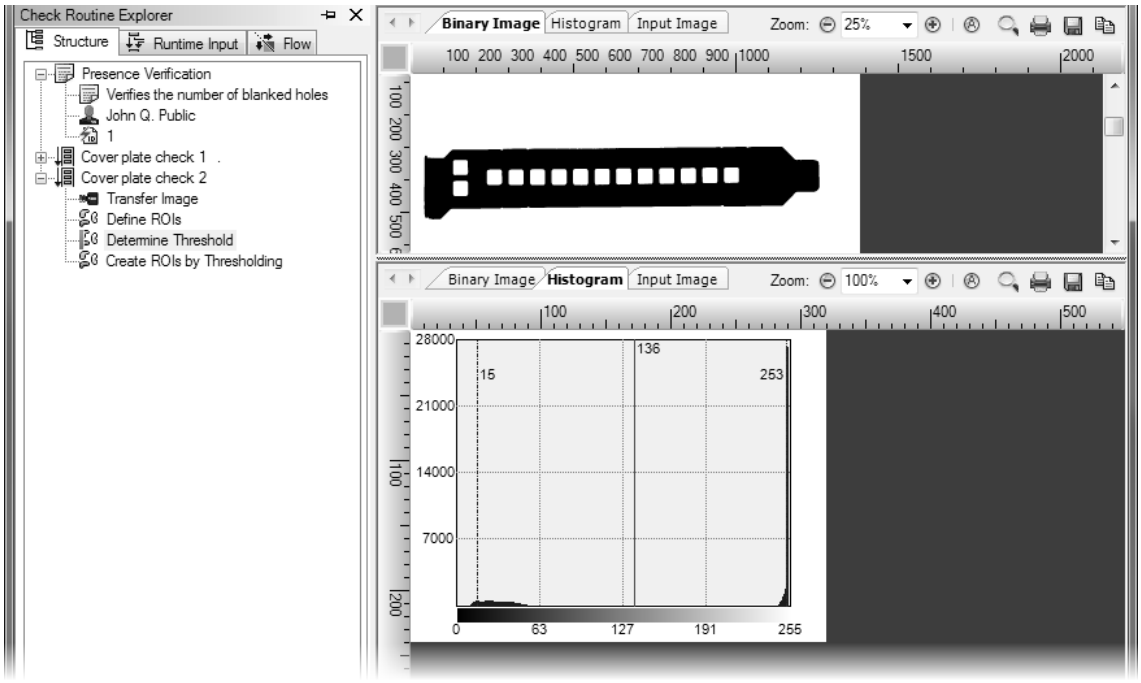
3 In this image the holes are not within the search area and can therefore not be created.



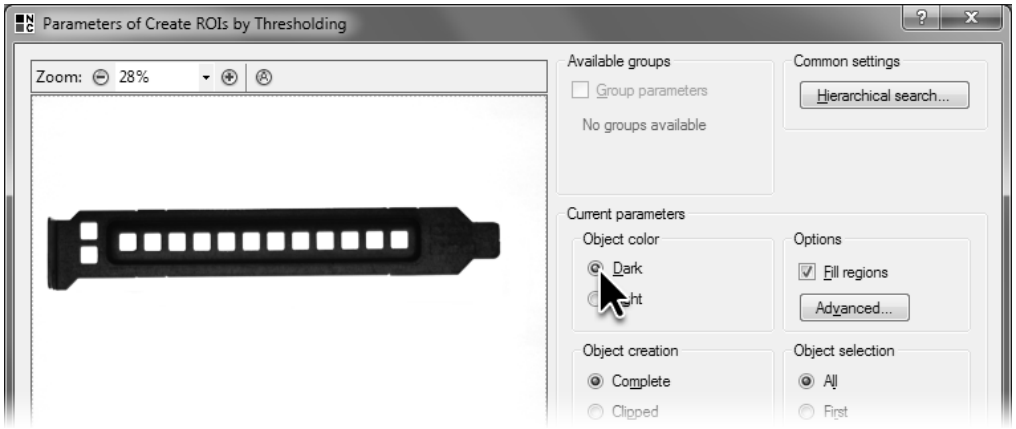
4 Adjust the search area to look for the cover plate in the entire image.

5 Execute the check function **Determine Threshold** and make sure that the computed binary threshold is suitable for looking for the cover plate. If not, open the parameter dialog and adjust the parameters.

6 The check function **Create ROIs by Thresholding** has been used to look for the white holes. But now we are looking for the black cover plate which is why we have to change the **Object color**. For this open the **Create ROIs by Thresholding** parameter dialog and set **Object color** to **Dark** to look for the black cover plate. Exit the dialog by clicking **OK**.



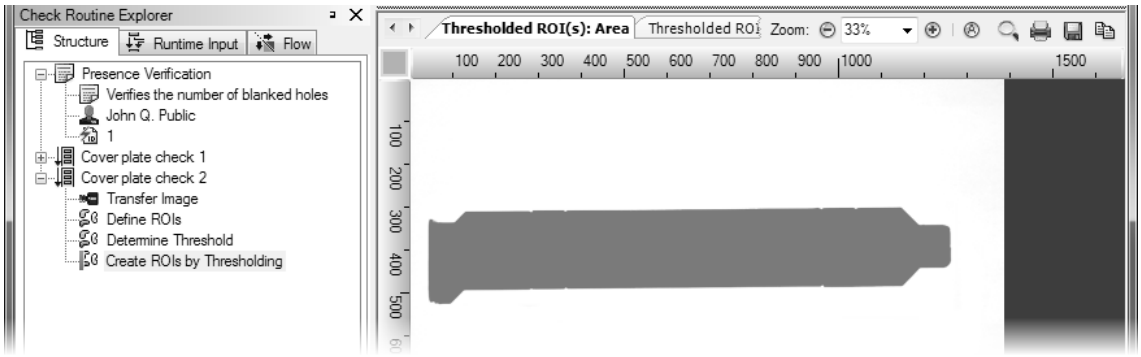
5 Make sure that the binary threshold separates the cover plate visibly from the background.



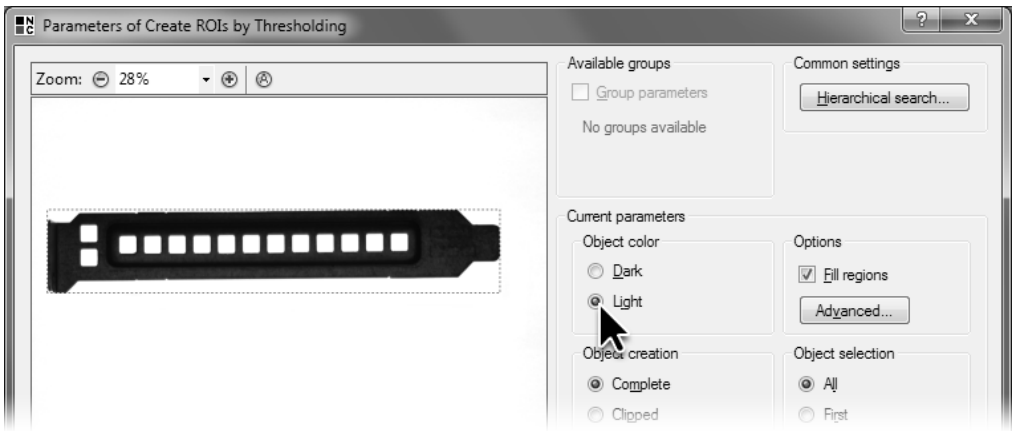
6 Configure the check function **Create ROIs by Thresholding** to look for the black cover plate against a white background.

7. Execute the check function **Create ROIs by Thresholding**. This check function will find the cover plate and will draw the created object in the Result View in blue.

8. In the next step we are going to look for the holes in the newly found object. For this, add the check function **Create ROIs by Thresholding** to the check a second time. Open the parameter dialog and set **Object color** to **Light** to look for the white holes against the black cover plate. Exit the dialog by clicking **OK**.



7 The cover plate is created as an object. The ROI created by thresholding can be used as search area for the holes.

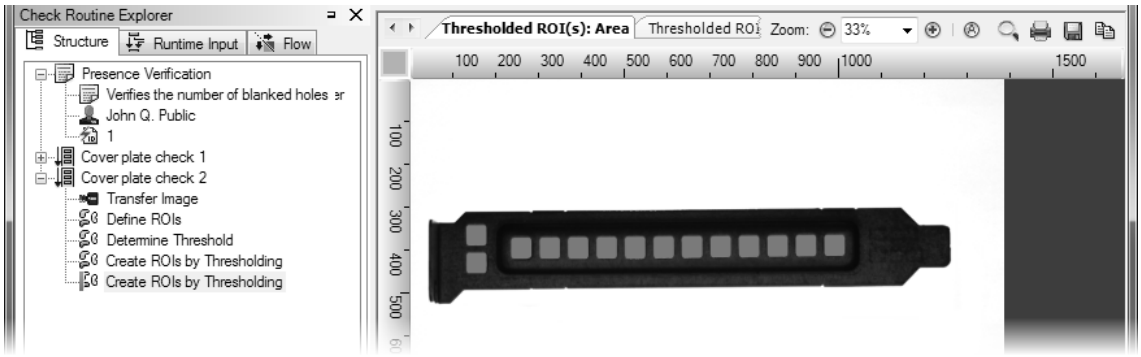


8 To look for the holes in the cover plate, white objects are created.

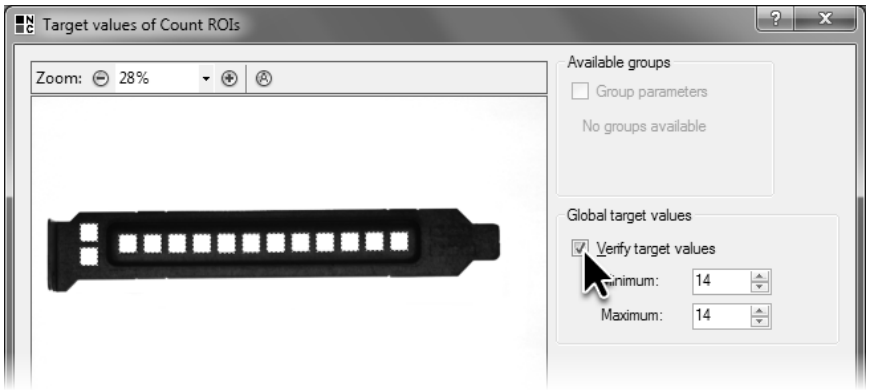
9 Execute the check function. This check function will create the holes as objects in the cover plate that are depicted in blue in the Result View. The check function will use the threshold created by **Determine Threshold** and as search area the ROI found by the previous check function **Create ROIs by Thresholding**. This makes sure that the holes will always be within the search area irrespective of the cover plate's position.

10 Add the check function **Count ROIs** to the check. Open the target value dialog, enable the **Verify target values** check box and enter 14 as **Minimum** and **Maximum**. Since the two holes on top of each other on the very left of the cover plate will be found now, we have 14 holes that are found in the cover plate.

Open the parameter dialog of **Transfer Image**, click **Browse** and select the images `Cover_Plate_01_OK.bmp`, `Cover_Plate_02_NOK_Missing_ Hole.bmp` and `Cover_Plate_03_OK_Moved.bmp`. Exit the parameter dialog by clicking **OK**.

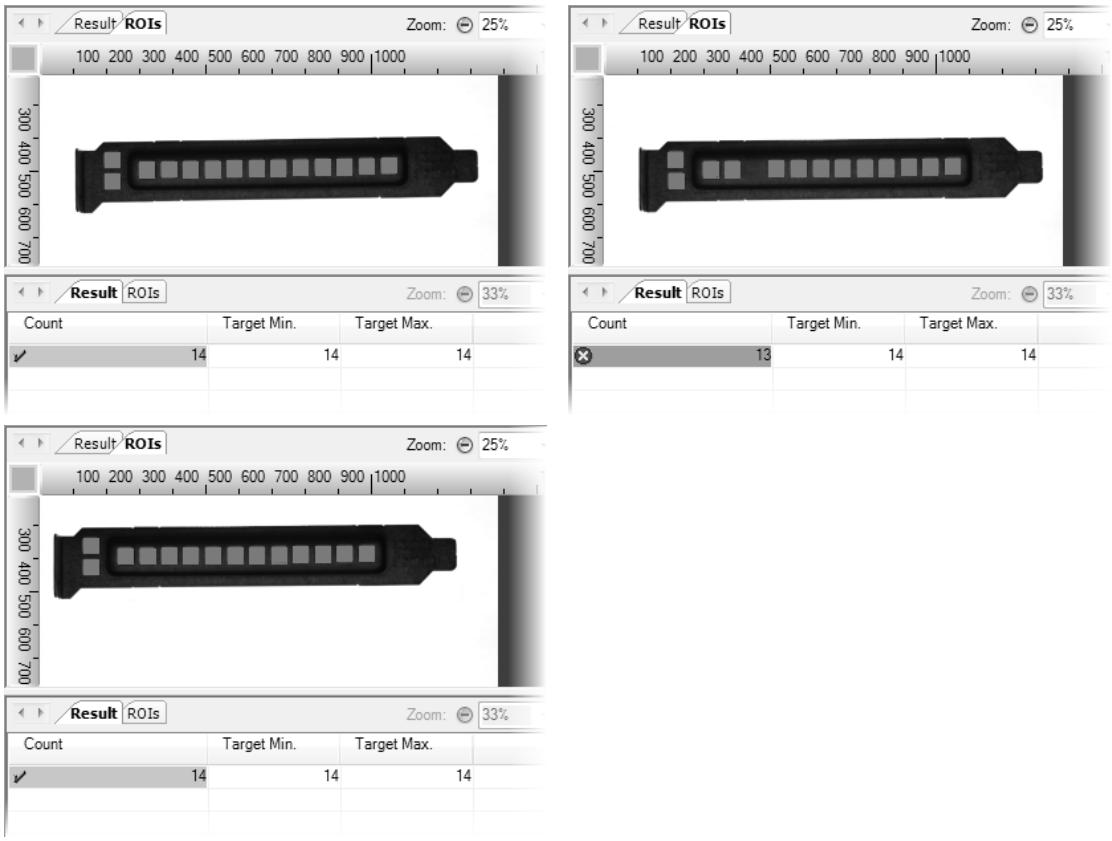


9 The holes are created as objects by thresholding.



10 Since the two holes on top of each other on the very left of the cover plate will be found now, there must be 14 ROIs.

11 Select the check function **Count ROIs** and execute the check several times. Irrespective of the cover plate's position in each image, all holes are found and the depicted cover plates are correctly classified as *OK* or *not OK*.



11 The inspection of the cover plates will yield correct results for all three images.

4.3 Presence verification using object properties

Goal

You want to configure the inspection in such a way that even disturbances such as reflexions are tolerated.

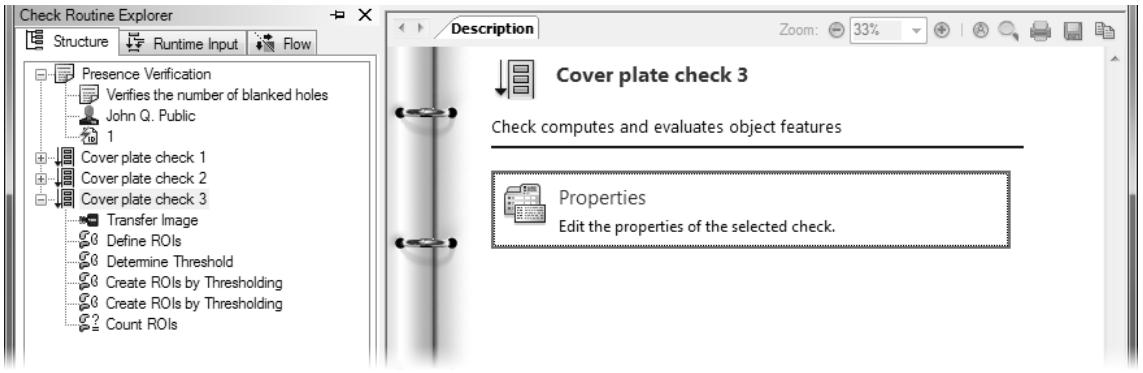
Result

You will know how to determine object properties and evaluate objects according to their properties.

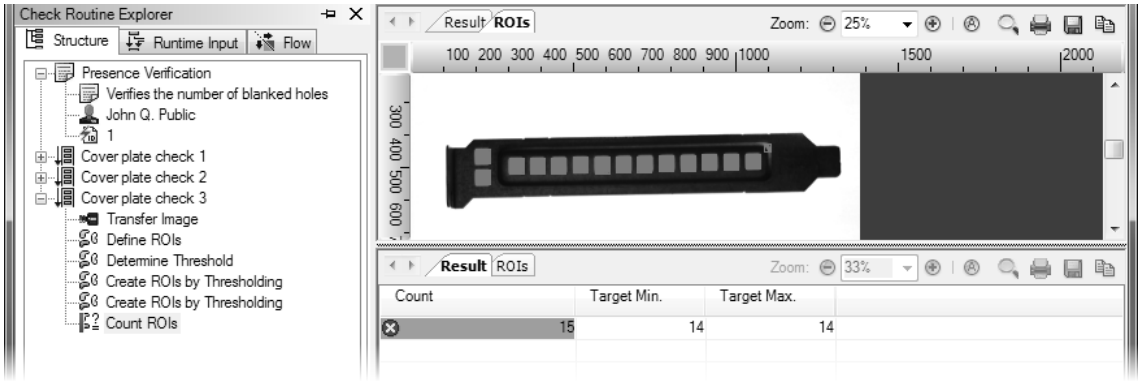
Solution

1.
   The procedure is identical to the previous example extending it with the determination and evaluation of object properties. Copy the previous check by clicking the depicted symbol from the NeuroCheck menu bar or select **Copy Check** from the **Edit** menu or context menu. Save the copied check as "Cover plate check 3" and change the description to "Check computes and evaluates object features".

2.
   Open the parameter dialog of the **Transfer Image** check function, click **Browse** and select the image `Cover_Plate_04_OK_Reflexion.bmp`. Execute the check function **Count ROIs**. The result is *not OK* since the reflection on the cover plate is found as the 15th object. In the following steps we are going to add differentiating between holes and reflections to the inspection.



❶ Copy the check and adjust the properties.



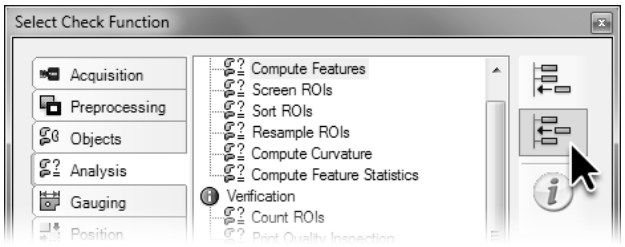
❷ In the image `Cover_Plate_04_OK_Reflexion.bmp` the reflection is taken for another object. Thus there seem to be too many holes.

❸

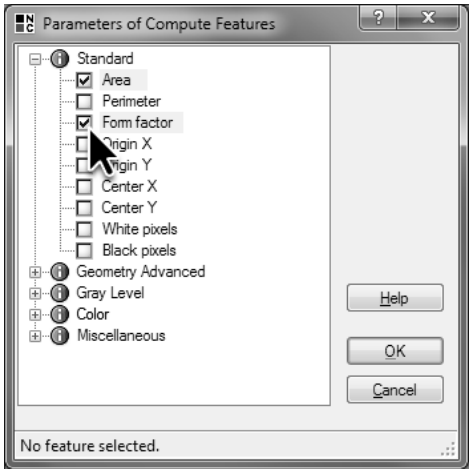
Select the check function **Count ROIs**. Open the **Select Check Function** dialog and select the check function **Compute Features** from the **Analysis** group and click on the depicted icon to insert the check function before the selected check function **Count ROIs**.

❹

Open the parameter dialog of **Compute Features**. You will see a list of features that can be computed for the ROIs. Disable the default features **Origin X** and **Origin Y**. Instead enable the features **Area** and **Form factor**.



❸ Prior to the check function **Count ROIs** insert the check function **Compute Features** from the **Analysis** group.



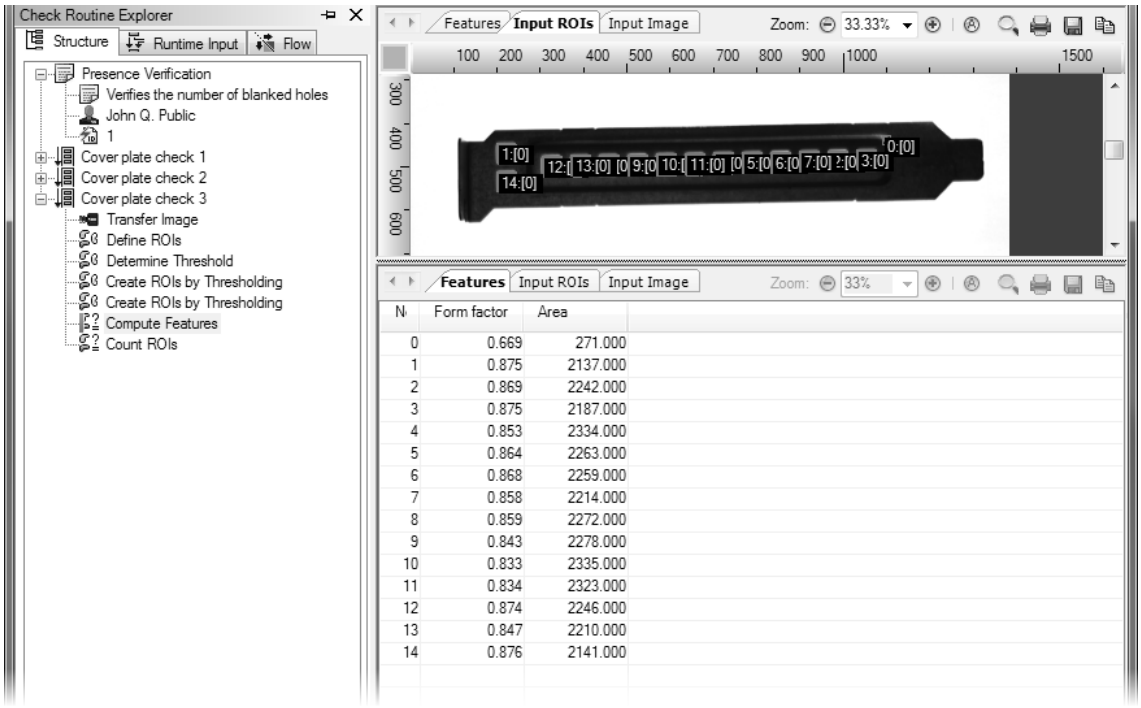
❹ Disable the default features and enable **Area** and **Form factor** to remove the reflection.

5

Execute the check function and split Result View so you can see both the **Features** and **Input ROIs** tab. Internally NeuroCheck numbers the ROIs consecutively. The **Input ROIs** tab shows the numbers depicted in yellow against a black background in the ROIs. The **Features** tab displays the table with the computed features of the ROIs. The table's first column displays the number of the ROI which also determines the sorting of the table.

The **Area** corresponds to the number of pixels in an object. The **Area** is a good feature to differentiate the reflection from the holes since the holes are significantly bigger, i.e. consist of more pixels, as can be seen from the computed values.

The **From factor** is an indicator for the object's form. The **From factor** is the ratio of an object's area and its perimeter, the value is between 0 and 1. Circles have a **Form factor** of 1 while differently shaped objects have lower values. The square-cut holes of the cover plate are more similar to a circle than the elongate reflection as can be seen from the computed values.



❺ The **From factor** and the **Area** are computed for all found objects and displayed in the **Features** tab. In the **Input ROIs** tab the ROIs and their corresponding indices are displayed.

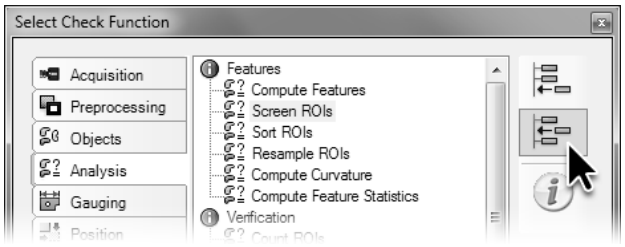
6

The next step is to remove those objects that are not holes using the computed features. For this select the check function **Count ROIs**, open the **Select Check Function** dialog and select the check function **Screen ROIs** from the **Analysis** group. Click the depicted icon to add the check function before the check function **Count ROIs**.

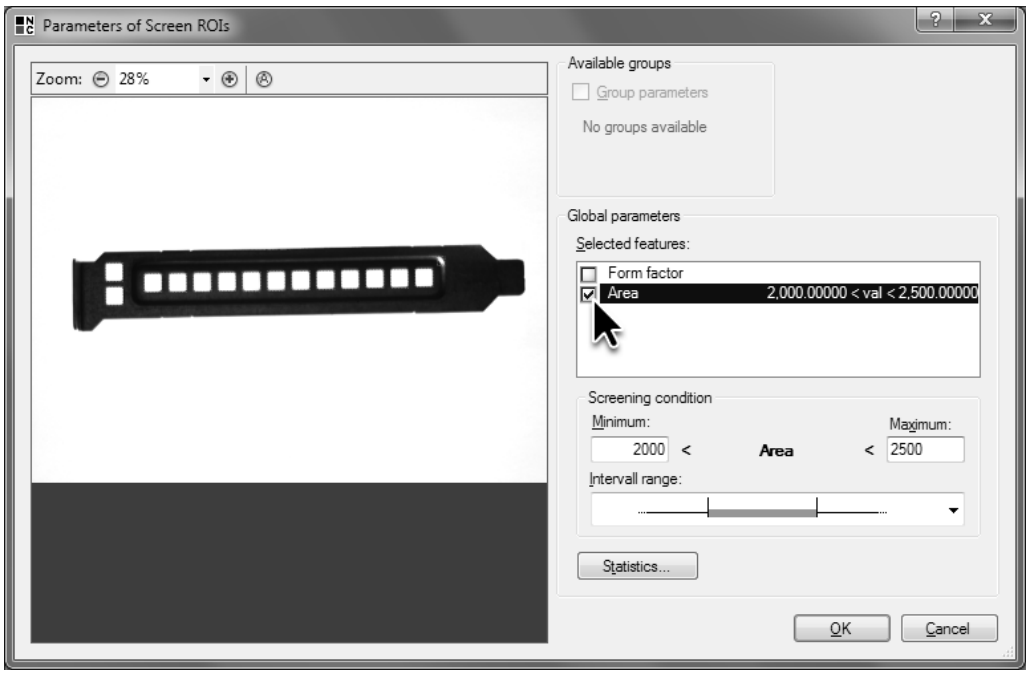
7

Open the parameter dialog of **Screen ROIs**. A list displays the features that were computed for the objects and which can also be used for screening.

First, enable the check box for screening the **Area**. For **Interval range** select the mode for which the green bar is between the two vertical lines. With this mode, the objects are retained whose features are between the **Minimum** and the **Maximum**. Enter 2000 for the **Minimum** and 2500 for the **Maximum** since the objects' **Area** is between these two values. If you used other parameters for computing the binary threshold, the holes can be smaller or bigger than the value stated here. In this case you have to adjust the values for **Minimum** and **Maximum** according to your object's size.



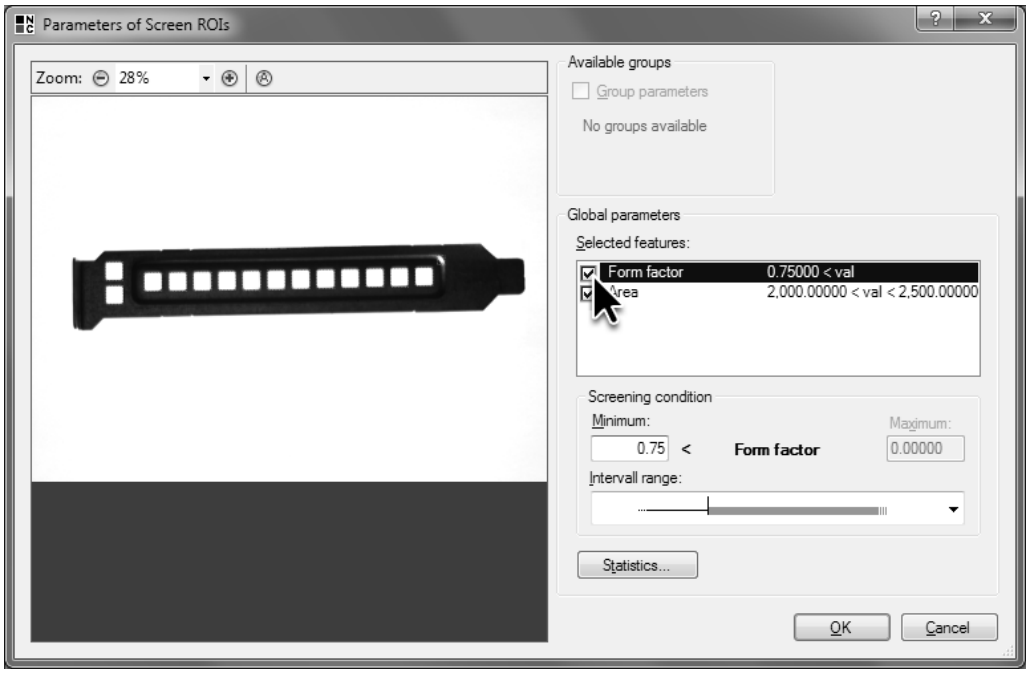
6 Prior to the check function **Count ROIs** insert the check function **Compute Features** from the **Analysis** group.



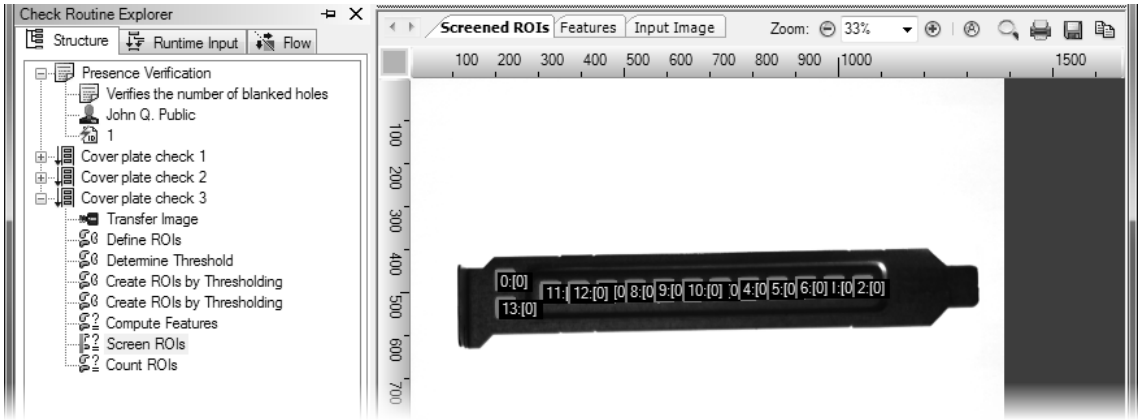
7 Enable the **Area** feature as a screening criterion. Select the screening mode with a valid range between **Minimum** and **Maximum**. Set the **Minimum** to 2000 pixels and the **Maximum** to 2500 pixels.

8. Then enable the check box for screening according to **Form factor**. For **Interval range** select the mode for which the green bar starts on the left with the vertical line and runs to the right. With this mode, the objects are retained whose feature value is above **Minimum**. Enter .75 for the **Minimum** since the holes' **Form factor** is above this value. If you used other parameters to compute the binary threshold, the holes' **Form factor** can be smaller so you have to select a smaller value as the **Minimum**.

9. Execute the check function **Screen ROIs** so the list of ROIs will only contain objects fulfilling both criteria. The reflection's **Area** and the **Form factor** do not fulfill the criteria, thus it is removed from the list of ROIs.

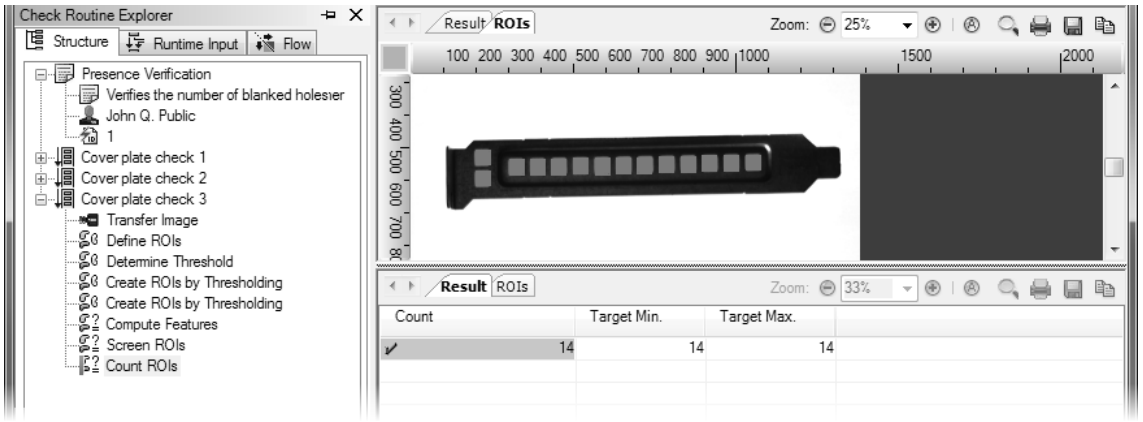


❽ Enable the **Form factor** feature as a screening criterion. Select the screening mode with a valid range above the **Minimum**. Set **Minimum** to .75.



❾ Since the reflection's **Area** and **Form factor** are outside the valid range, the corresponding ROI is removed.

10 The check function **Count ROIs** will now report *OK* for the image with the reflection since now only the 14 required ROIs are found.



10 After removing the reflection, **Count ROIs** will yield the result *OK* since the correct number of objects is present.

4.4 Presence verification using Template Matching

Goal

You want to verify the correct assembly of a plug on a circuit board.

Result

You will know how to look for objects in the image using **Template Matching**.

Solution

1. Create a new check routine and save it as "Template Matching". The automatically created check can be named "Presence verification plug".

Add the check function **Transfer Image** to the check and click **Browse** in the check function's parameter dialog. Select the image `Board_01_OK.bmp`.

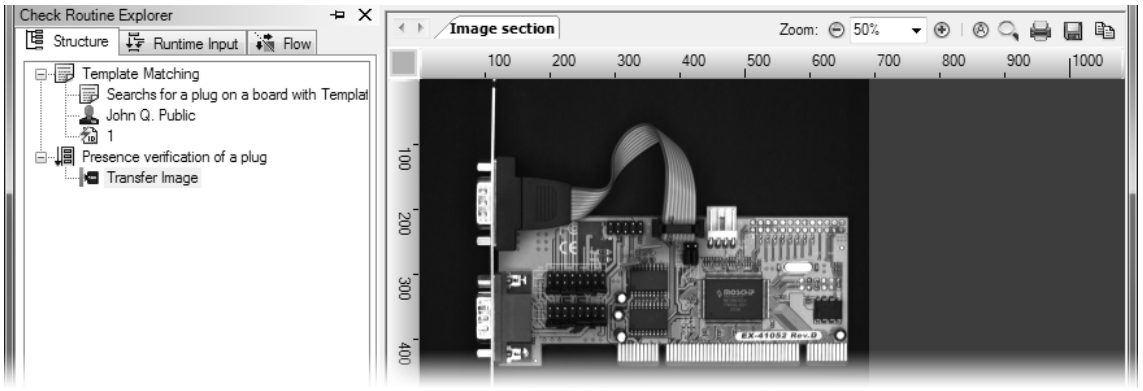
You can see a plug in the top middle of the board. The goal of this check is verifying the presence of the plug. Because of the shadow between the plug and the edge of the board, it's not possible to verify the plug's presence reliably using thresholding. Therefore, we are going to use a different segmentation method to look for the object called Template matching.

Add the check function **Define ROIs** to the check and define a rectangle in the parameter dialog enclosing the entire image.

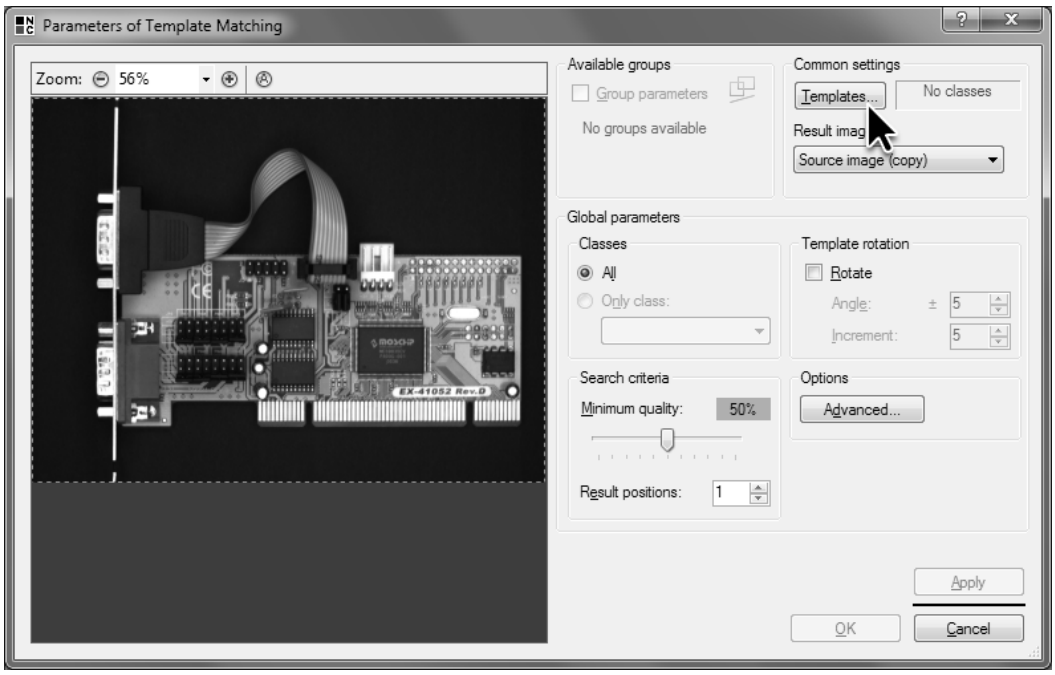
2. Add the check function **Template Matching** from the **Objects** group to the check and open the check function's parameter dialog. On the top right, the dialog indicates that no classes have been defined yet.

For template matching you have to define sample images of the object you are looking for during configuration, we call these “templates”. Upon executing this check function, these templates are moved step by step over the image and compared to each image section. In those positions where the template and the image are very similar, i.e. where the template matches the image section, an ROI is created.

In NeuroCheck templates are organized in classes. One class corresponds to one type of object. To execute the check function, it is necessary to define at least one class containing one template. Click the **Templates…** button.



1. Create a new check routine and load the image `Board_01_OK.bmp` into NeuroCheck.

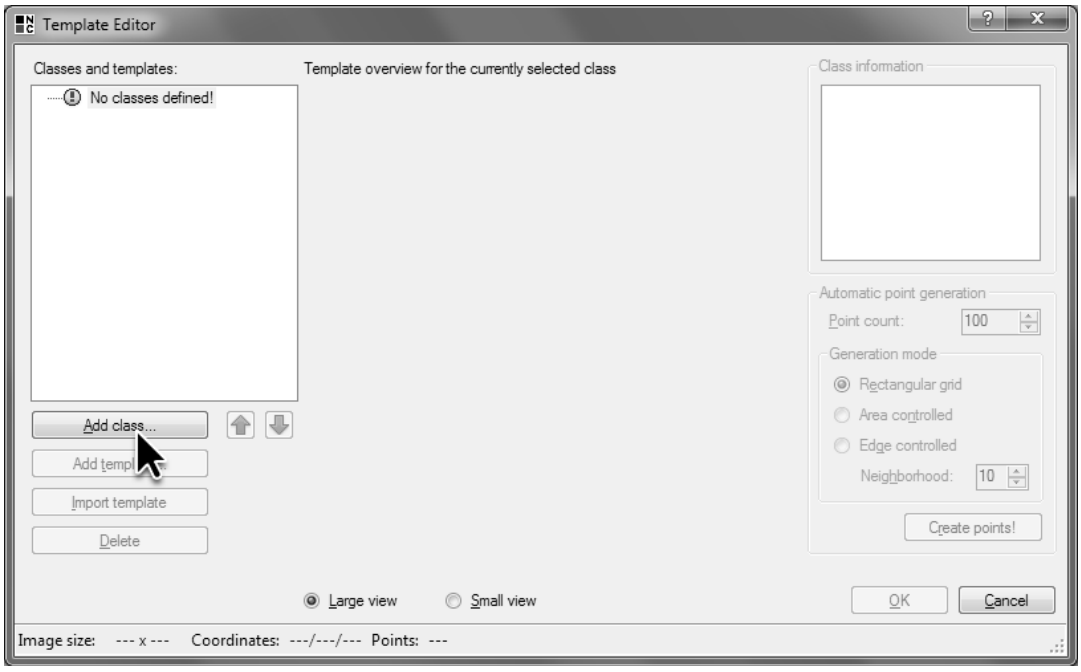


2. Open the parameter dialog of the check function **Template Matching** and there you open the **Template Editor**.

3. The **Template Editor**, a dialog for creating and editing templates, has been opened. Now you have to create a class. Click the **Add class…** button.

4. In the **Add class...** dialog, enter the name for the new class "Plug". The new class is created when you exit the dialog by clicking **OK**.

In the tree view of the template editor you can see the newly created class. So far the class contains no template. Therefore, click the **Add template…** button to define a new template for this class.



3. First, you need to define a class in the **Template Editor**. Click the **Add class** button.

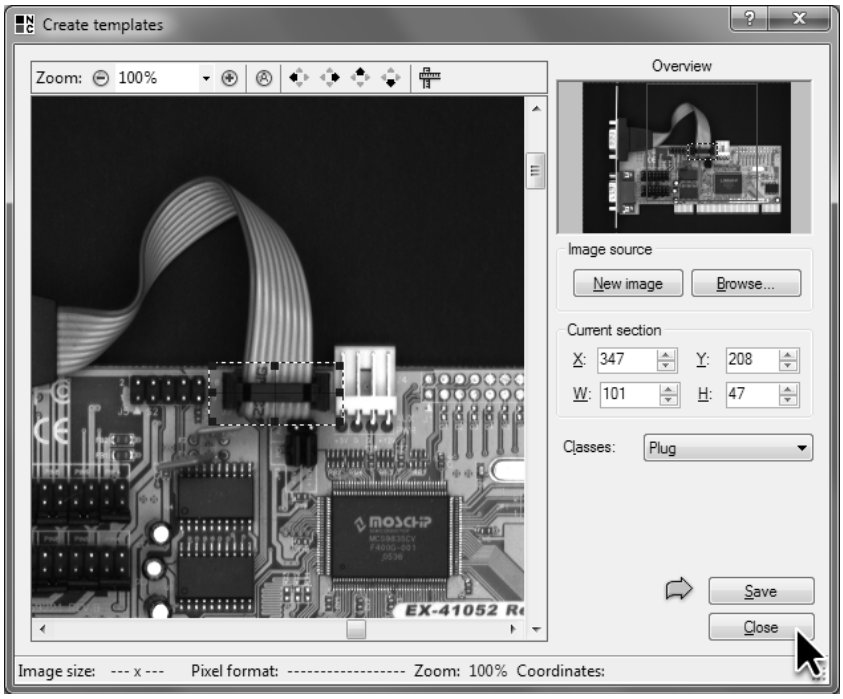
4. In the **Add class** dialog, give the class a name.

5 You can create new templates in the **Create templates** dialog. Click within the red frame of the **Overview** window. If you keep the mouse button pressed, you can move the red frame and also the visible image section. Move the frame so you can see the plug and its surroundings in the main window of the parameter dialog. Then let go of the mouse button.

Click the middle of the plug in the dialog's main window. Around the point where you clicked, a rectangle will be drawn. This rectangle is the image section for the definition of a template.

Position the image section so that the plug and some space around it will be within the image section. By clicking one of the blue squares and keeping the left mouse button pressed, you can adjust the size of the image section. By clicking within the image section and keeping the left mouse button pressed, you can move the image section. You can also enter the position and size of the image section pixel accurate using the input fields of the group frame **Current section**.

Set the class the template belongs to in **Class**, in this case the "Plug" class. Click the **Save** button to save the selected image section as template. Exit the dialog by clicking the **Close** button.



❺ Use the **Create templates** dialog to select image sections and save them as templates.

6

In the **Template Editor** dialog you will see the newly created template listed in the tree view **Classes and templates** in the "Plug" class. In **Grid points of selected template** view you will see the created template with a number of blue crosses. We will deal with the blue crosses in the next section. Close the **Template Editor** by clicking the **OK** button.

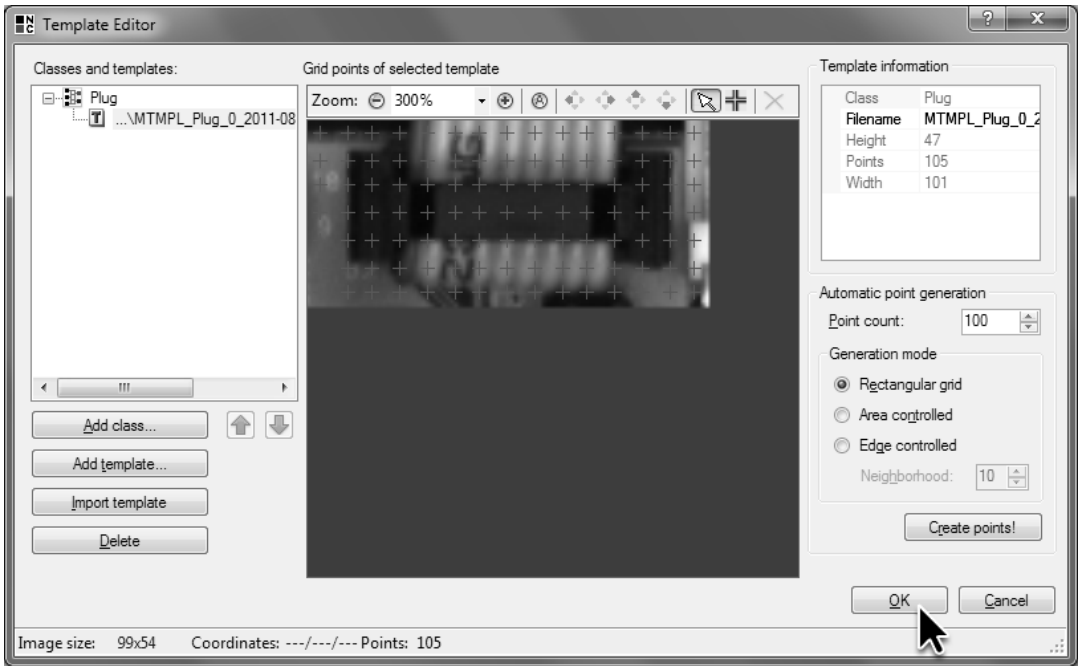
The input field **Result positions** of the parameter dialog displays the number of objects that are looked for. Since the plug is present just once, you can leave the default value 1 and exit the dialog by clicking **OK**.

7

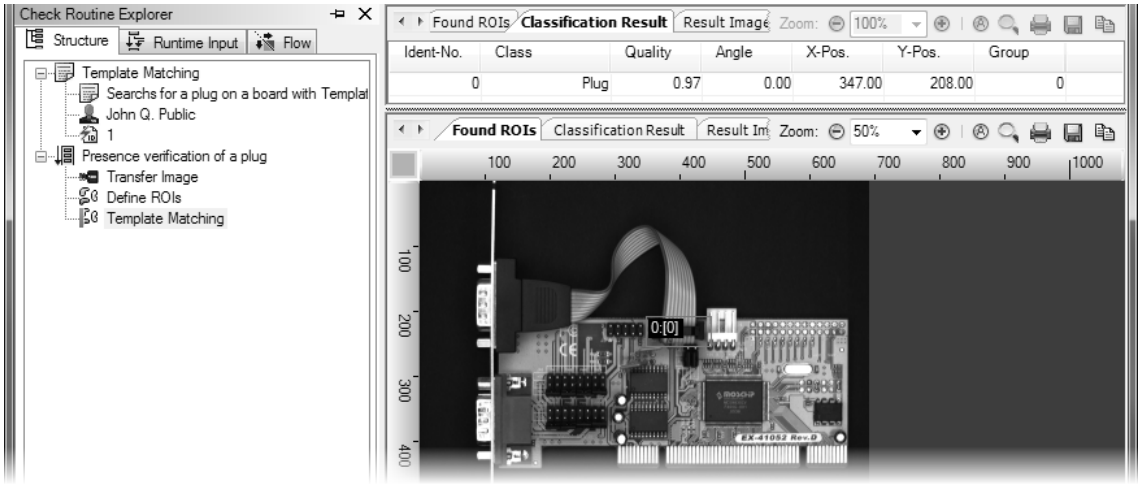
Execute the check function. Split the **Result view** to be able to see the **Found ROIs** and **Classification result** tabs.

In the **Found ROIs** tab, a rectangle the size of the template is drawn around the plug. This specifies the position of the best match between template and image.

The **Classification Result** tab displays the results of the object search. The third column displays the quality. Quality signifies the correspondence of template and image with values ranging between 0.0 and 1.0. A quality of 1.0 (i.e. 100%) is almost never achieved, not even in the image the template was taken from because of runtime optimizations.



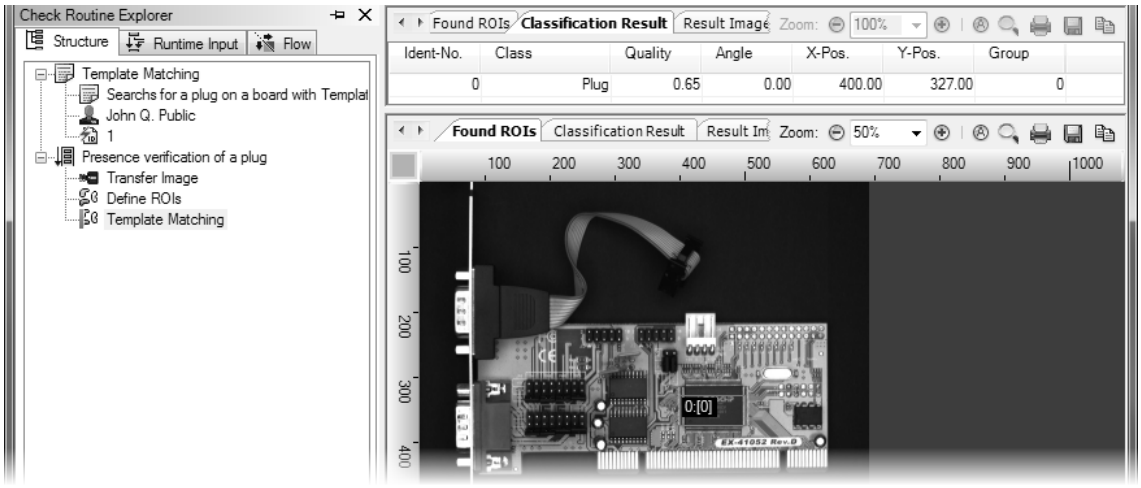
❻ The **Template Editor** dialog displays the classes and their corresponding templates.



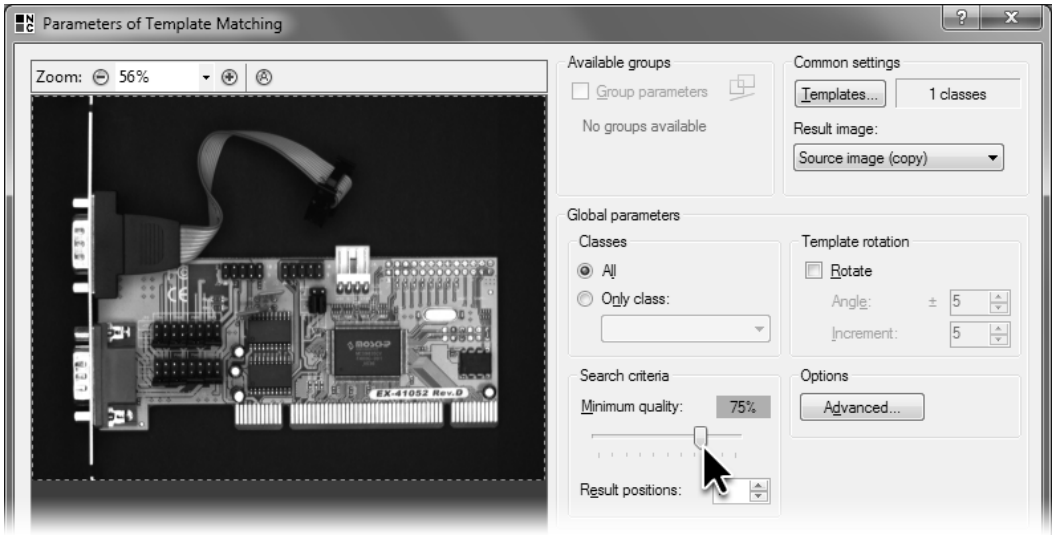
❼ The plug will be found with a quality of 0.97 (i.e. 97%). Because of runtime optimizations, a quality of 100% is almost never achieved.

8 Just as a quality of 0% is never achieved. In the check function **Transfer Image**, if you select the image `Board_02_NOK.bmp` and then execute **Template Matching**, a match will be drawn somewhere in the image. The exact position and quality depend on the template. In any case, there will be a similarity between any section of the image and the template and therefore there will be a correspondence. To prevent the template from being located in unsuitable positions, open the **Template Matching** parameter dialog.

9 Use the **Minimum quality** slider to set the minimum quality of a match to be accepted as an object. Set the slider to 75% to avoid pseudo findings.



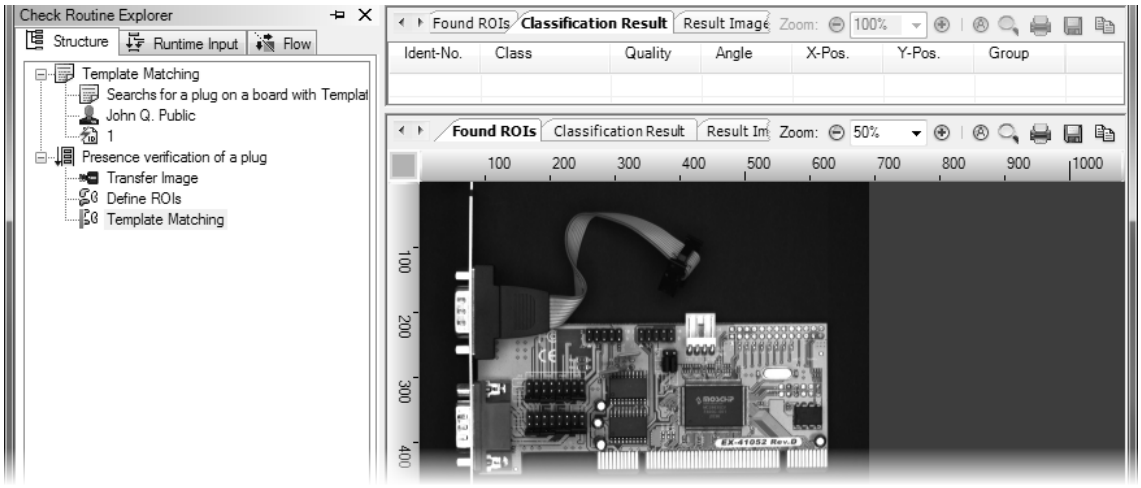
8 If the plug is not mounted, the chip will be found with the highest degree of quality of 65%.



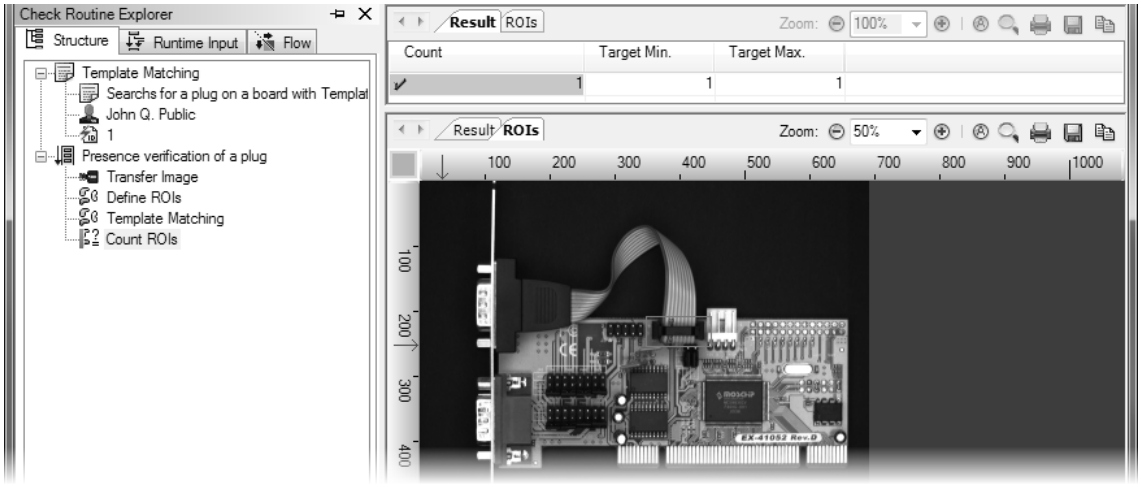
9 By setting the **Minimum quality**, matches with too little match between the image and the template can be suppressed.

10. Close the parameter dialog by clicking **OK** and execute the check function. In case of the picture with the unmounted plug no more matches are displayed since for all positions the quality of the match is below the configured 75%.

11. Use the check function **Count ROIs** to verify whether the plug has been found, i.e. correctly mounted or not. In the target value dialog, set **Minimum** and **Maximum** for the expected ROIs to 1.



⑩ If the plug is not correctly mounted, **Template Matching** will find no image section matching the template with the required quality.



⑪ Use the check function **Count ROIs** to ensure that the correctly mounted plug will be found in the image.

4.5 Optimizing Template Matching

Goal

You want to reduce the execution time for template matching.

Result

You will know how to reduce the number of correlation points that are used for template matching.

Solution

1. To measure the effect of the runtime optimization, click the depicted icon to display the **Execution time report**. Then execute the check. The **Execution time report** will display the time for the execution of each check function.

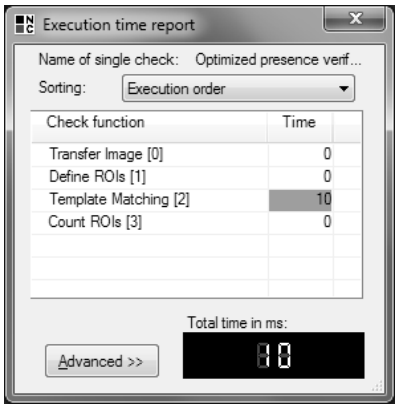
Copy the current check and rename the new check "Optimized presence verification plug".

2. In the new check, open the parameter dialog of the check function **Template Matching** and open the **Template Editor** by clicking the **Templates…** button.

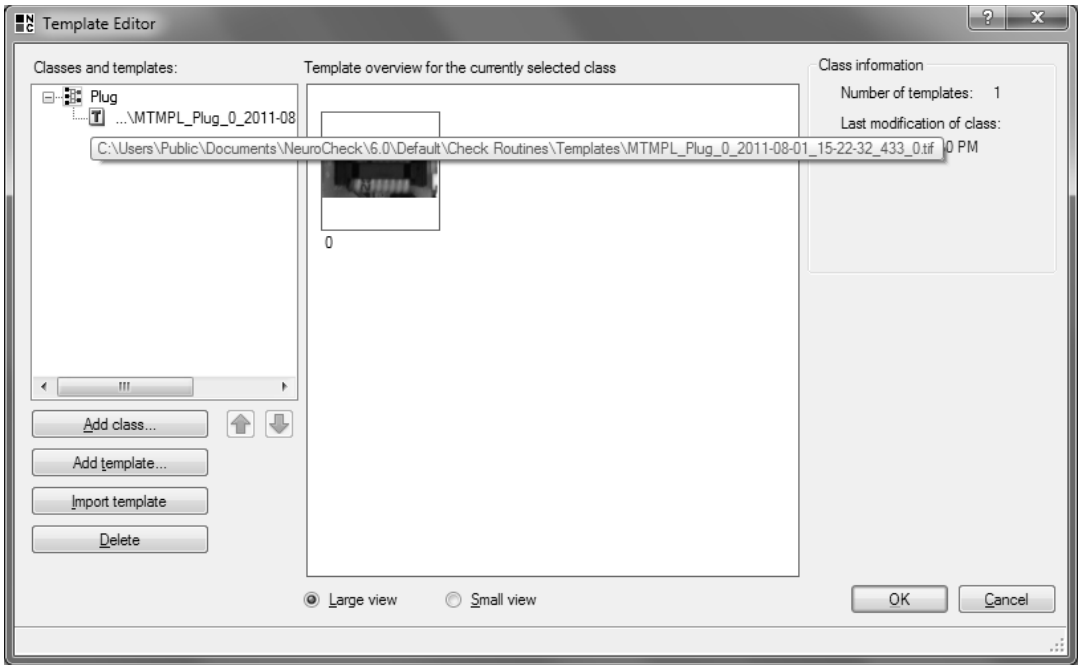
The templates are saved as separate image files. When copying the check, the reference to this separate template file was copied without the file itself being copied. The check functions **Template Matching** in both checks access the same template file. In order not to change the template from the original check, we will first copy the template.

For this write down the template's name. The template's name is displayed in a tool tip if the mouse pointer hovers over the template in the tree view **Classes and Templates**.

Delete the template reference by selecting it and clicking the **Delete** button or by pressing the Del key.



❶ The runtime log displays the runtime of each executed check function of the check.



❷ If the mouse pointer hovers of the template entry in the tree view **Classes and Templates**, the complete path to the template file is displayed.

3. Create a copy of the template by clicking the **Import Template** button. In the **Open** dialog, select the template file whose name you have written down. In the background, a copy of the template file is automatically created that is used by the new check function **Template Matching** as template for the "plug" class.

4. In **Grid points of selected template** view you will see the imported template. On a regular grid, blue crosses are drawn in the template's image. These crosses pinpoint the correlation points where the template is actually matched to the image. Although many points of the template are not used for comparison, the template is reliably found, as you have just seen.

   To reduce runtime, the number of correlation points can be further reduced. Set **Point count** to 50.

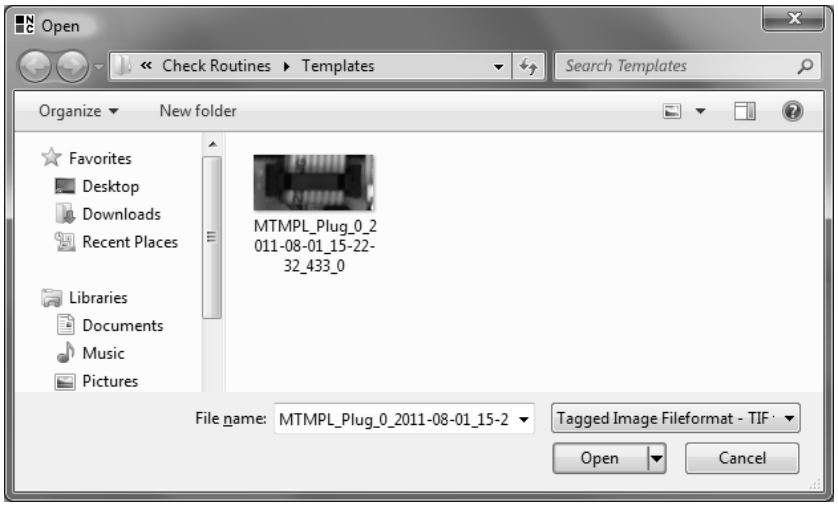
   To go on finding objects reliably, the correlation points should be arranged in a more intelligent manner. If you place two correlation points next to a dark-light border in such a way that one point is in the dark and one in the light area, you can check whether this border exists. Using this principle, objects can be found more reliably using few correlation points than with points arranged in a rectangular grid.

   Select the option **Edge controlled** to position points along the dark-light border, i.e. edges.

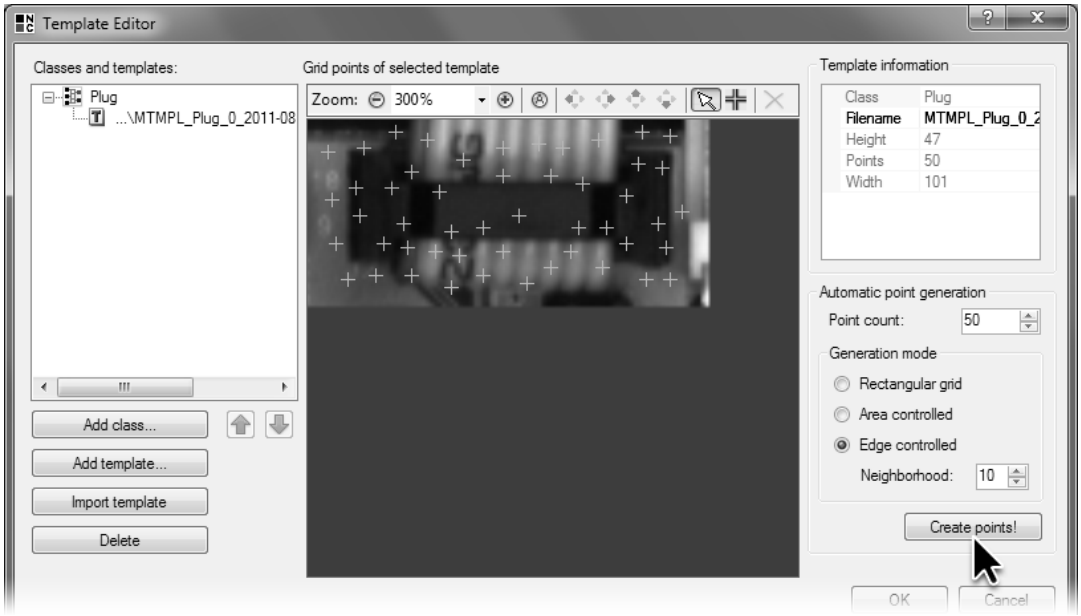
   Set the distance of the correlation points of the dark-light border in the input field **Neighborhood** to 10.

   Click the **Create points!** button to create correlation points according to the settings.

   Close the template editor and parameter dialog by clicking **OK**.



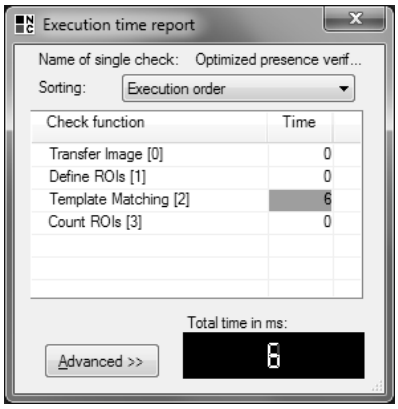
❸ When importing the template from the first check, a copy is automatically created.



❹ In **Edge controlled** mode, the correlation points are placed around the edges after clicking the **Create points!** button.

5. Execute the check. The time logged by the **Execution time report** is much shorter after optimization than before.

Please note that the runtime of the check function **Template Matching** increases linearly with the number of templates. Therefore, the necessary computing time increases when an application varies the appearance of a workpiece so much that several samples are necessary for a reliable detection.



5. After optimizing the correlation points, the runtime of the check function **Template Matching** is significantly shorter.

5. Automated Inspection

The main application area of NeuroCheck is automated visual inspection in the manufacturing process. For seamless production integration of the inspection, NeuroCheck is able to communicate with master computers or programmable logic controllers (PLC).

In this chapter you'll learn how to

- load the drivers necessary for this communication
- remote control NeuroCheck from a PLC or master computer
- conduct a check routine change

5.1 Configuration of fieldbus for communication

Goal

You want to establish communication via fieldbus between NeuroCheck and the control system of your manufacturing process (PLC, master computer).

Result

You will be able to register and test a fieldbus card in NeuroCheck. For this you can integrate and configure a suitable communication driver, a data format converter (converter). Furthermore, you will learn about the virtual fieldbus device of NeuroCheck, which you can use for the examples from the training manual if you do not have suitable hardware installed in your system.

Solution

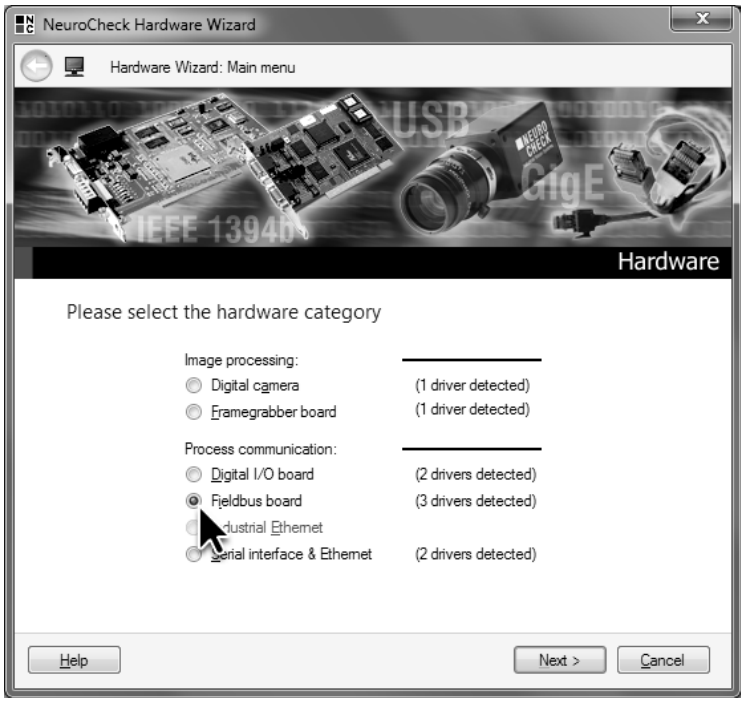
1. Open the device manager by clicking **Device manager** in the **System** menu or the depicted icon. Click the **New…** button to integrate a new fieldbus card. On the wizard's first page select the **Fieldbus board** option and click **Next**.

2. On the second page choose a suitable driver for your fieldbus card. If there's no fieldbus card in your computer, you can use NeuroCheck's "virtual fieldbus device". The virtual fieldbus consists of a NeuroCheck driver and a program acting as the remote station. The NeuroCheck driver of the virtual fieldbus can be integrated into NeuroCheck just like a fieldbus card. This makes it possible to configure all settings in NeuroCheck for the communication without a real fieldbus card. Use the remote station to set the inputs of the virtual fieldbus device interactively using your PC and to monitor the outputs.

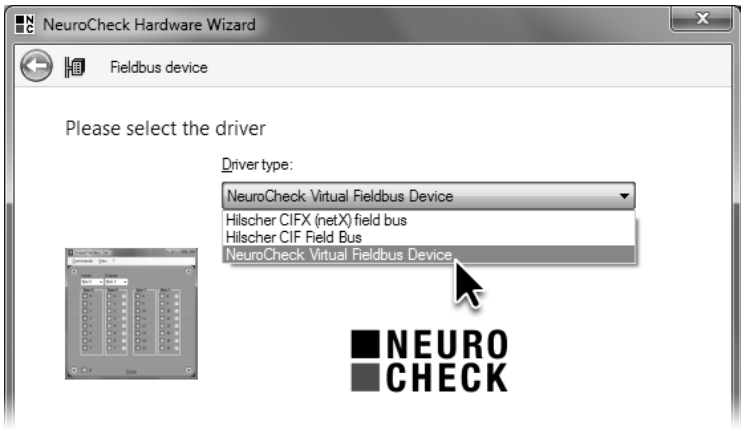
   If you have access to a fieldbus card at a later time, you just have to integrate the card and remove the virtual fieldbus from the device manager to enable communication via the actual fieldbus card.

   Let us suppose you're using a virtual fieldbus. All information is analogously valid for real fieldbus cards and their remote station on a control computer.

   You can either select the driver for the fieldbus card in your PC or *NeuroCheck Virtual Fieldbus Device* from the **Driver type** list. Switch to the next page by clicking **Next**.



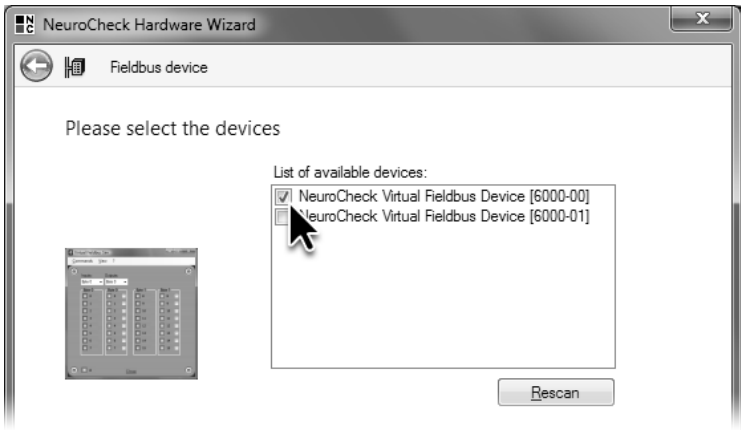
1. From the **NeuroCheck Hardware Wizard** select the **Fieldbus Card** option.



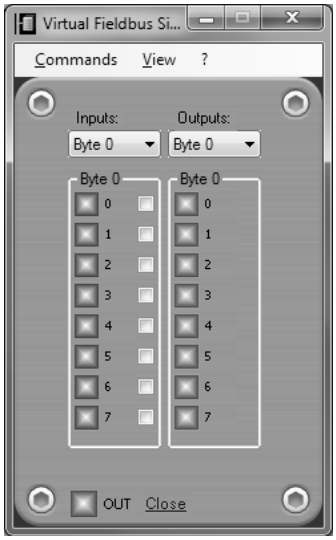
2. On the second page select the *NeuroCheck Virtual Fieldbus Device* driver.

3. On the wizard's last page select the virtual fieldbus device *NeuroCheck Virtual Fieldbus Device [6000-00]* and exit the wizard by clicking **Rescan**.

4. The remote station of the virtual fieldbus *Virtual Fieldbus Sim0* is opened as a separate application in a new program window. The remote station shows two rows of bits. The green column on the left shows the status of the remote station's outputs corresponding to the inputs of the driver in NeuroCheck. Use the selection fields to set the status of the outputs. The red column on the right shows the status of the remote station's inputs corresponding to the outputs of the driver in NeuroCheck.



3. On the third page select the *NeuroCheck Virtual Fieldbus Device [6000-00]*.



4. The program *Virtual Fieldbus Sim0* serves as remote station of the virtual fieldbus.

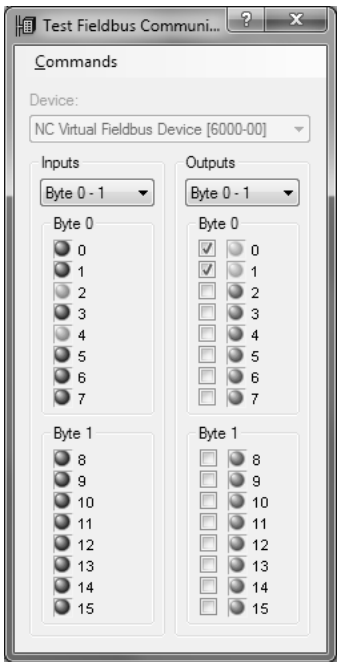
5 In the category **Fieldbus devices** of the device manager you'll find the newly integrated *NeuroCheck Virtual Fieldbus Device [6000-00]*. Select the device and click the **Test Communication...** button to open the **Test Fieldbus Communication** dialog.

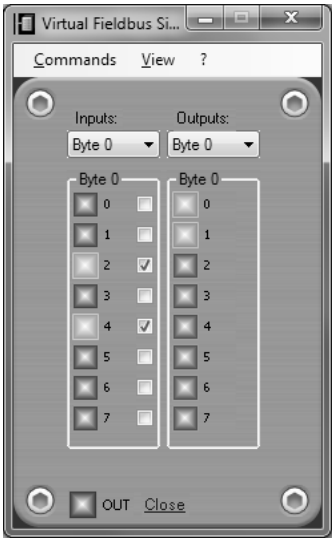
6 The **Test Fieldbus Communication** dialog is used to check whether the communication with the control system - here: the virtual remote station - is working. The blue column on the left of the dialog shows the status of the inputs of the fieldbus card integrated into NeuroCheck. The yellow column on the right shows the status of the fieldbus card's outputs. Use the check boxes to the right to change the status of the outputs for test purposes.

When you change the status of some outputs in the **Test Fieldbus Communication** dialog, the status of the corresponding inputs of the virtual remote station is changed in the same instant. You can also change the status of the outputs of the virtual remote station to see the changes of the fieldbus card's inputs in the **Test Fieldbus Communication** dialog.

Exit the **Test Fieldbus Communication** dialog by clicking the close button in the upper right corner of the window. Then exit the **Device Manager** by clicking **OK**.

5 Click **Test Communication** in the **Device Manager** to test communication with the remote station.





6 Set the outputs of the **Test Fieldbus Communication** dialog to check whether the signals of the corresponding inputs of the remote station are set.

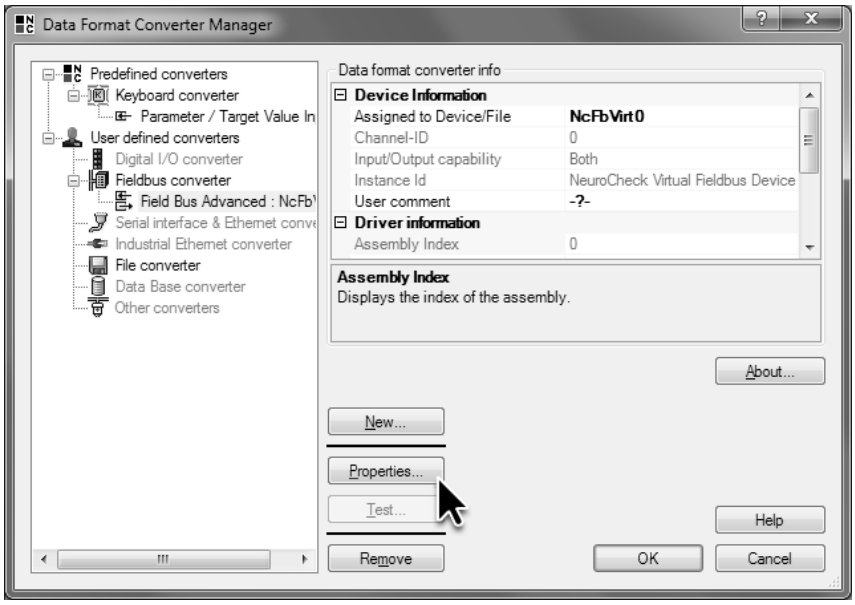
7. A window appears and you are asked whether you would like to integrate a data format converter automatically. It's the converter's job to convert the status of each bit of the fieldbus card into data that NeuroCheck can use. For example the converter can interpret the status pattern of several fieldbus inputs as binary representation of a number and transfer that number to NeuroCheck. Conversely, the results computed by NeuroCheck are converted into bit patterns according to which the outputs of the fieldbus card are set.

   Since NeuroCheck needs a converter to import and export data, it is mandatory to integrate a converter. Click the **Yes** button.

8. Open the **Data Format Converter Manager** to check the configuration of the automatically integrated Data Format Converter. Select the **Data Format Converter Manager** from the **System** menu or click on the depicted icon. The **Data Format Converter Manager** is the core dialog for managing converters. In the tree view on the left side of the **Data Format Converter Manager** dialog you can see in the sub-category **Fieldbus Converter** of the category **User defined converter** the entry *Fieldbus Advanced: NcFbVirt0*. This entry signifies the automatically integrated converter of the *Fieldbus Advanced* type that is connected to the fieldbus card *NcFbVirt0*.

   Select the *Fieldbus Advanced:* converter connected to the fieldbus card *NcFbVirt0* and click the **Properties** button to check the converter's configuration.

7. Click **Yes** to integrate a data format converter automatically.



8. Open the **Data Format Converter Manager**, select the converter *Fieldbus Advanced: NcFbVirt0* and click **Properties**.

9 The **Fieldbus Advanced – Data Format Converter Configuration** dialog has three pages that can be selected using the list on the left side of the dialog. The **Device** page shows the fieldbus card the converter is linked to and how many inputs and outputs the card has.

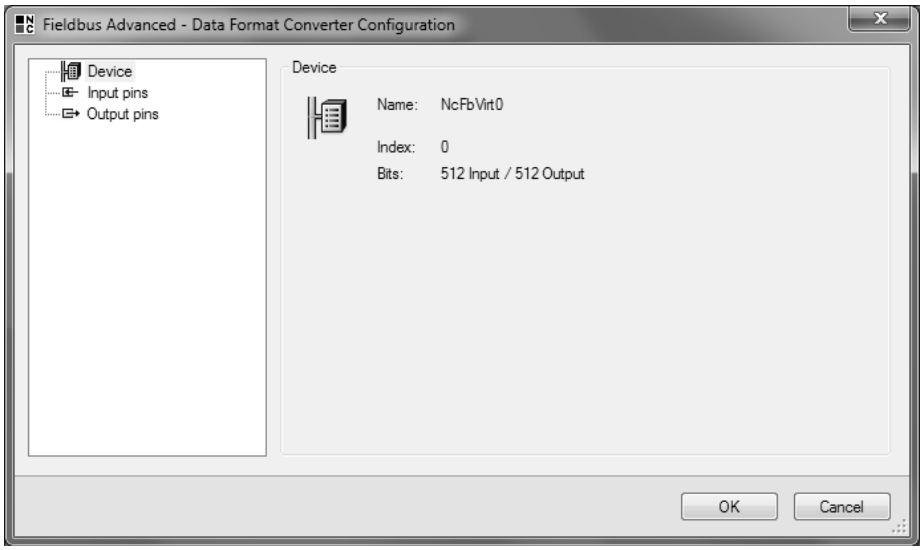
10 Switch to the **Input pins** page by clicking the list entry of the same name. It's the task of the data format converter to convert the status of the fieldbus card's inputs to data NeuroCheck can use, and to convert values computed by NeuroCheck to output status of the fieldbus card.

Data between NeuroCheck and the converter is transmitted via so-called pins. One pin corresponds to a value being transmitted between NeuroCheck and the converter. A pin can only transmit in input or output direction, i.e. either data is transmitted from the fieldbus card to NeuroCheck via the converter, then transmission takes place via input pins. Or the data are transmitted from NeuroCheck to the converter and then to the fieldbus card. The pins involved then are called output pins. The integrated converter can have both input and output pins to allow data transmission in both directions.

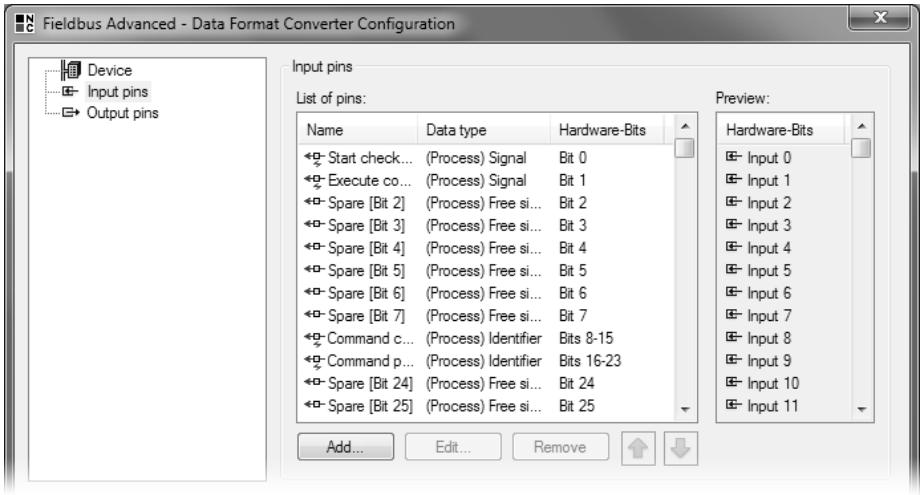
Upon integration of the converter into NeuroCheck, a number of input and output pins have been created. The input pins are displayed in the **List of pins** on the page of the same name of the **Fieldbus Advanced – Data Format Converter Configuration** dialog. The list shows the pre-defined pins and their properties. The left column of the list displays the pin's **Name** as shown in the selection dialogs of NeuroCheck. The name of the first pin in the list of pins is, for example, *Start check routine*.

The middle column displays the pin's **Data type**. Each pin is of a specific data type, for example **Boolean**, **Integer**, **Float** or **String**. Depending on the value's application in NeuroCheck, the pin needs to be of a suitable data type. For example, the data type of the first pin on the list is **(Process) Signal**, which corresponds to a Boolean value used in **Remote control**.

The right column of the list shows the inputs of the fieldbus card, i.e. **Hardware bits**, the pin's state depends on. The first pin on the list, for example, needs just one bit since it is of the type **(Process) Signal** and only signifies a Boolean value. In this case, this pin is linked to input 0 of the virtual fieldbus.



❾ On the **Device** page of the **Fieldbus Advanced – Data Format Converter Configuration** dialog, the properties of the fieldbus card are displayed.



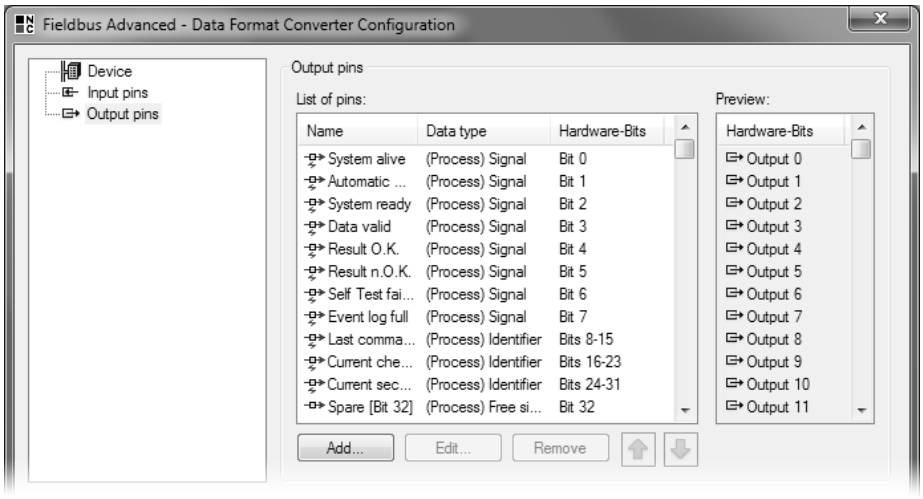
❿ On the **Input pins** page all pins are listed that are made available to NeuroCheck by the fieldbus card. The list also contains information to which hardware bits of the fieldbus card a pin is linked.

11

The **Output pins** page is structured like the **Input pins** page. However, the **List of pins** on the **Output pins** page contains pins NeuroCheck can use to transfer data via the fieldbus card's outputs to the control system.

For the following sections the output pins *System alive* (linked to output 0), *Automatic mode enabled* (linked to output 1), *System ready* (linked to output 2), *Result O.K.* (linked to output 4) and *Result not O.K.* (linked to output 5) are important.

Do not change the configuration of the data format converter and exit the **Fieldbus Advanced – Data Format Converter Configuration** dialog and then the **Data Format Converter Manager** by clicking the **Remove** button.



⓫ On the **Output pins** page all pins are listed that are used by NeuroCheck to transfer data to the fieldbus card. The list also contains information to which hardware bits of the fieldbus card a pin is linked.

5.2 Configuration of remote control

Goal

After the previous section has already laid the foundation for fieldbus communication between NeuroCheck and the remote station or control system, you want to configure NeuroCheck to remote control it from the remote station or control system.

Result

You will be able to configure the settings for starting an inspection and communicating the check routine result.

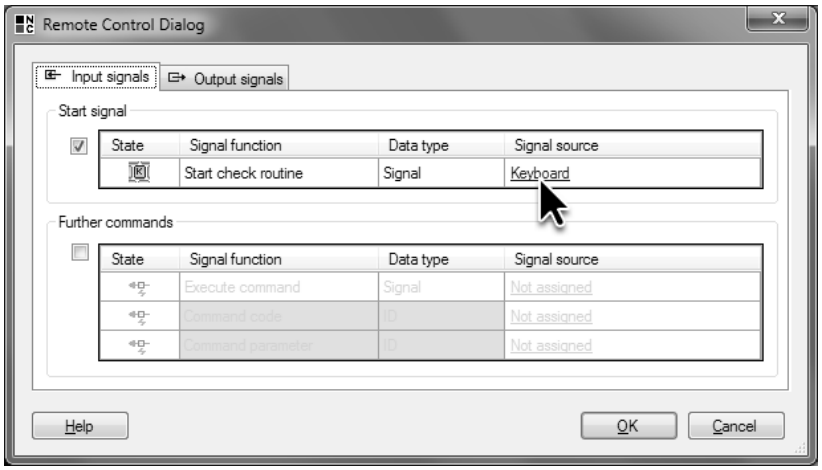
Solution

1. Select **Remote Control** from the **System** menu. **Remote Control** is the central dialog for configuring the input signals, i.e. the functionality for the automated control of NeuroCheck from a master computer. Furthermore, the output of status messages by NeuroCheck is also configured in the **Remote Control** dialog.

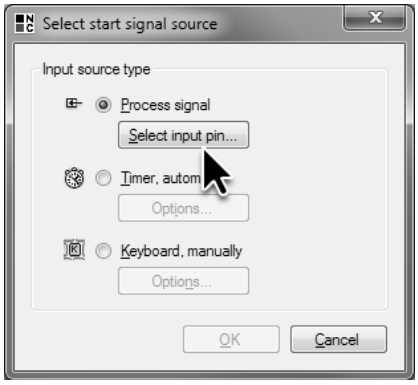
Switch to **Input signals** tab if it isn't displayed already. The most important command NeuroCheck has to process is the **Start signal**, the signal to start an inspection, i.e., to execute a check routine. The signal **Start check routine** is already activated. The signal source is configured as in chapter 2 (*Working with Check Routines*) as **Keyboard**. In real visual inspection stations in industrial manufacturing the start signal will be coming from the master computer or, for example, from a photoelectric barrier. To configure remote control so that the signal **Start check routine** can be received from the fieldbus card via the Data Format Converter, click the link **Keyboard** in the **Signal source** column.

2. The **Select start signal source** dialog is opened and you can choose from three kinds of start signals. To receive the start signal via an input pin from the virtual fieldbus device, select the option **Process signal**.

Now you need to configure which pin of the converter will be used by **Remote Control** to read the start signal. Click the **Select input pin** button.



1. Open **Remote Control** and click the link **Keyboard** to configure the **Start signal**.



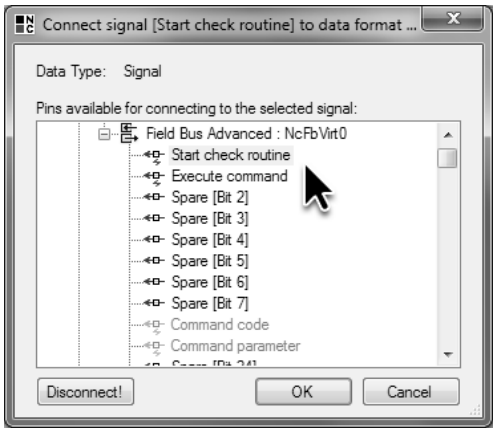
2. Select **Process signal** as start signal source.

❸ The **Connect signal [Start check routine] to data format converter pin** dialog opens and displays a list of available pins with the suitable data type and the correct direction of data exchange. In the **User-defined converters** group you'll find the sub-group **Fieldbus converter** and in that group the automatically integrated data format converter *Field Bus Advanced : NcFbVirt0* from the previous chapter. As sub-entries of this converter the available input process pins are listed. Since the input signal **Start check routine** is linked to an input pin for exchange of a Boolean value, the input process pins for exchanging numeric values are disabled.

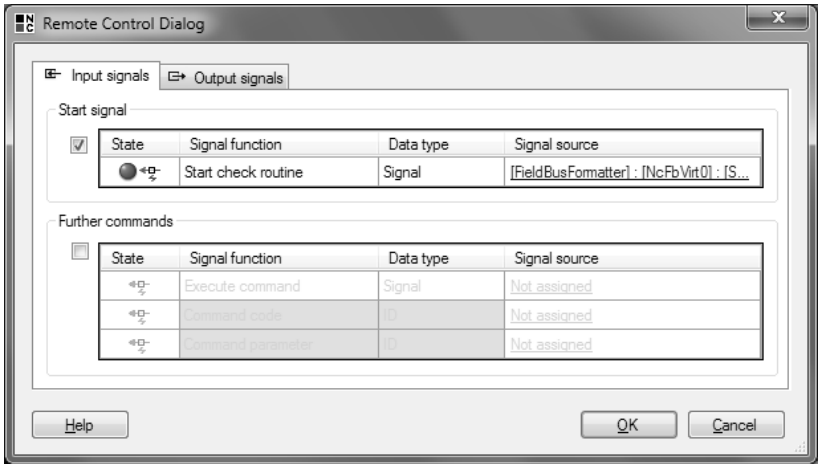
Click the entry *Start check routine* to link this input pin to the input signal **Start check routine**. Close the **Connect signal [Start check routine] to data format converter pin** dialog and the **Select start signal source** dialog by clicking **OK**.

❹ In the remote control dialog, next to the **Start check routine** entry on the **Input signals** page, in the **Signal source** column you'll now find the selected pin *Start check routine* that is part of a converter for fieldbus cards that is connected to the field bus card *NcFbVirt0*.

In automatic mode NeuroCheck will continuously check the state of the input pin *Start check routine*. As soon as the state of the pin is active, the current check routine is executed. Right after the execution of the check routine has ended, the status of the pin is cyclically checked until it is activated again, then the check routine is executed once again.



3. Select the process pin *Start check routine* as source for the input signal of the same name.

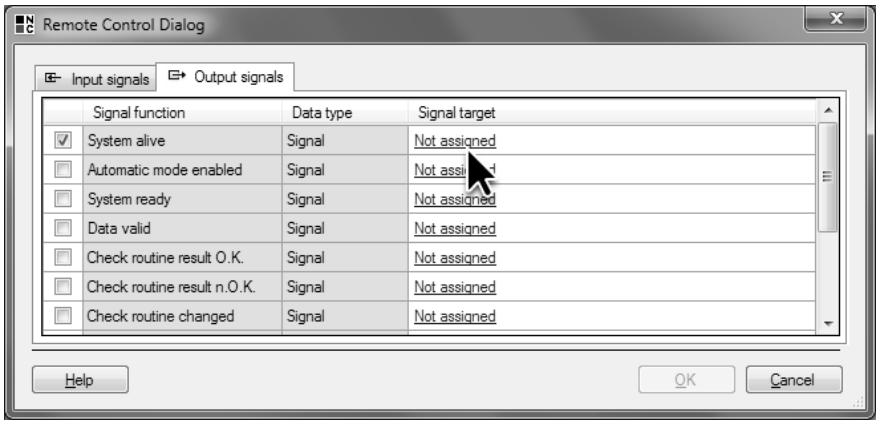


4. On the **Input signals** page of **Remote Control,** the new source of the **Start signal** is displayed.

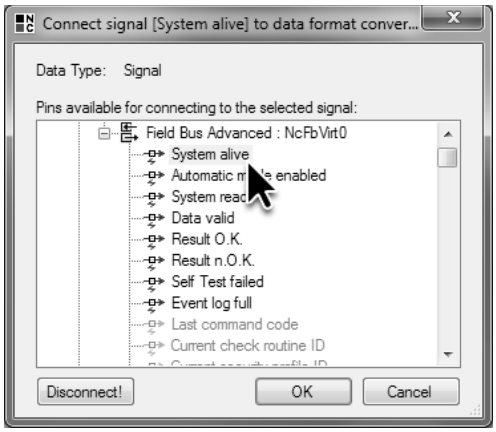
5 Switch to the **Output signals** page to configure the output of status information to the fieldbus card. Activate the output signal **System alive** to transmit whether the NeuroCheck software is running or not. Click the link **Not assigned** to select the signal target over which the output signal **System alive** is communicated.

6 You already know the dialog **Connect signal [System alive] to data format converter pin** from the selection of the input pin for the signal **System alive**. In this case the output process pins are displayed.

Select the output pin *System alive* as signal target for the output signal **System alive** and close the dialog by clicking **OK**.



5 Click the link **Not assigned** in the row of the output signal **System alive**.



6 Select the process pin *System alive* as source for the output signal of the same name.

❼ Likewise, activate the output signals **Automatic mode enabled**, **System ready**, **Check routine result O.K.** and **Check routine result not O.K.**, and link them to the respective output pins as signal targets.

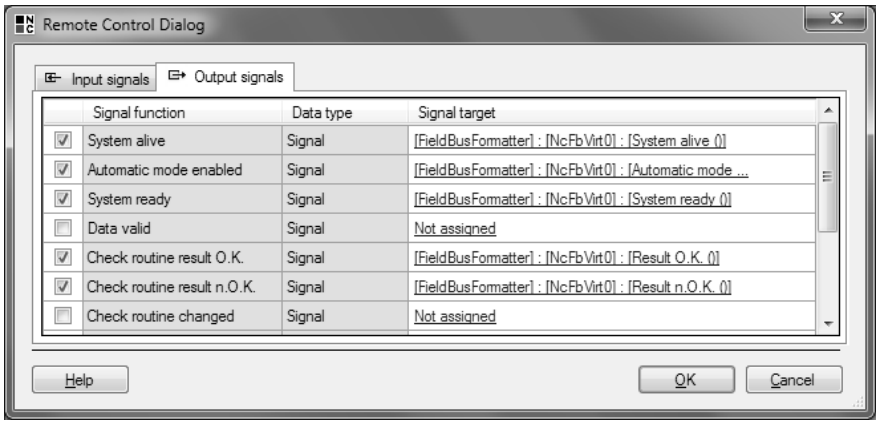
The output signal **Automatic mode enabled** indicates whether the automatic mode of NeuroCheck has been started. Only when automatic mode is enabled, NeuroCheck is able to respond to a start signal.

The output signal **System ready** indicates whether NeuroCheck is ready to execute a command like the start signal. For this, automatic mode must be enabled and no check routine must be executed at the time.

The output signals **Check routine result O.K.** and **Check routine result not O.K.** give the result of the current inspection. If the result of the inspection is *okay* (abbr. O.K.), **Check routine result O.K.** is set and **Check routine result not O.K.** is not set. If the result of the inspection is *not okay*, (abbr. not O.K.), **Check routine result O.K.** is not set, instead **Check routine result not O.K.** is set.

The result of the inspection is derived from the results of all checks in the check routine: If there is no other configuration, the result of the check routine is *okay* only when the results of all checks are *okay*. In other words: If just one check yields *not okay*, the result of the check routine is *not okay*.

Now the most important signals are configured to control NeuroCheck in automatic mode and to report state and results to the control system.



❼ On the **Output signals** page of **Remote Control** the targets for the activated output signals are displayed.

5.3 Execution of an automated inspection

Goal

You want to execute a check routine in automatic mode.

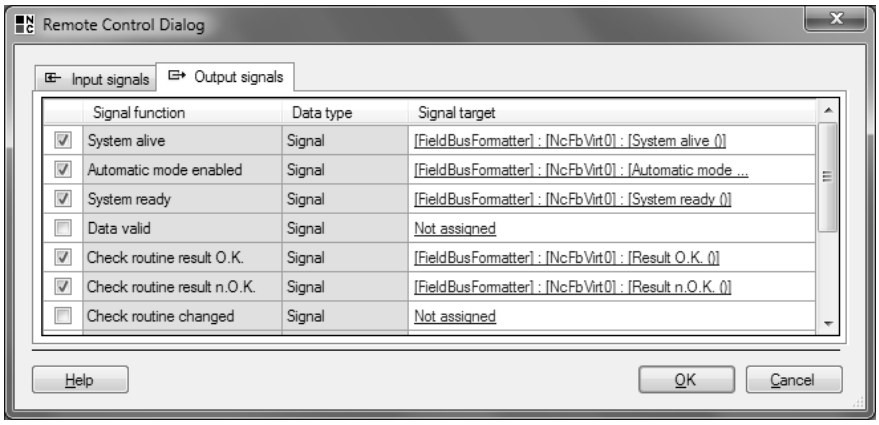
Result

You will get to know the status of the output signals in various modes of operation and the phases of a check routine execution. The sequence of signals and system states is irrespective of the start signal source and the target of the output signals (fieldbus, digital I/O, Ethernet, timer, keyboard...). We'll be using the virtual fieldbus and its remote station for process control and the display of output signals.

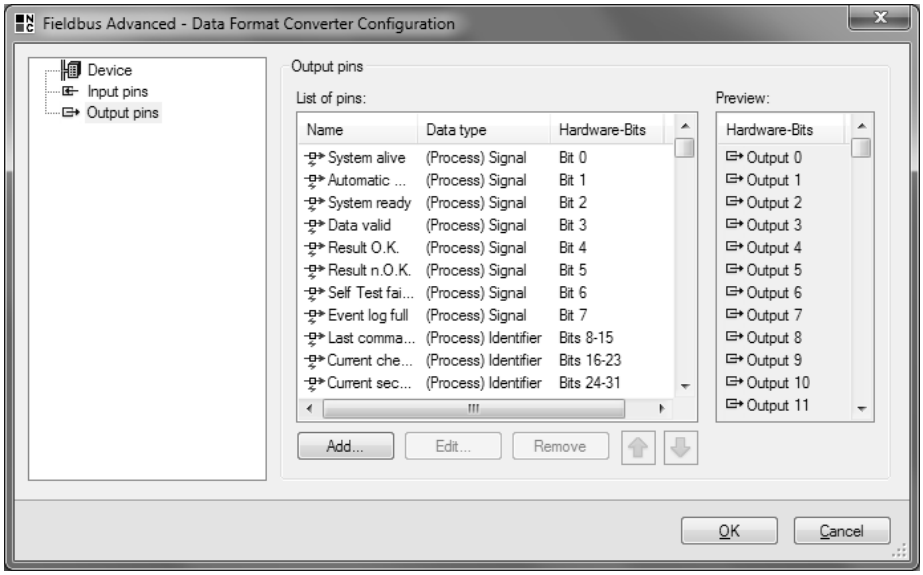
Solution

1. As you have seen in the previous sections, you can view the links between signals and pins from the **Output signals** page of the **Remote Control** dialog.

2. The links between pins and outputs of the fieldbus card are configured and displayed in the **Fieldbus Advanced– Data Format Converter Configuration** dialog.



❶ The assignment of output signals to output pins can be viewed on the **Output signals** page of the **Remote Control** dialog.



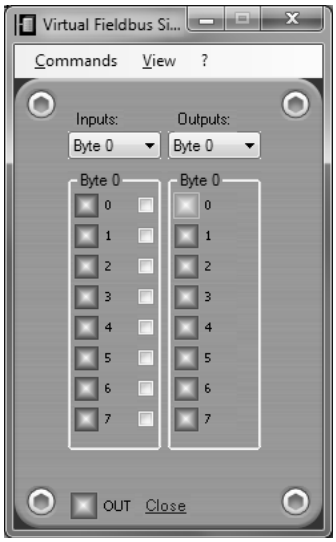
❷ The assignment of output pins to the outputs of the fieldbus card can be viewed on the **Output pins** page of the **Fieldbus Advanced– Data Format Converter Configuration** dialog.

❸ Initially, NeuroCheck is in manual mode. In this mode, only the output signal **System alive** is active, which is linked to the pin of the same name, which in turn is linked to the output with the index 0 of the virtual field bus. Therefore, from the inputs of the remote station of the virtual fieldbus only the red LED at bit 0 is glowing.

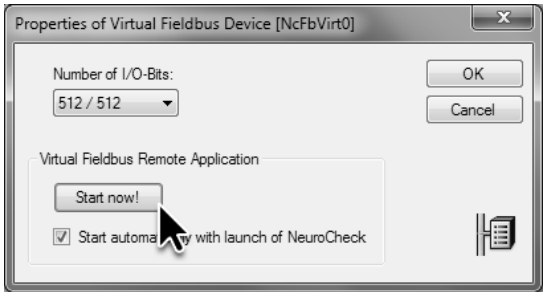
❹ If you have closed the virtual remote station application in the meantime, open the **Device manager**, select *NeuroCheck Virtual Fieldbus Device [6000-00]* and click **Properties**. In the **Properties of Virtual Fieldbus Device [NcFbVirt0]** dialog click the **Start now!** button to start the remote station again. Close the **Properties of Virtual Fieldbus Device [NcFbVirt0]** dialog and the **Device manager** by clicking **OK**.

Open the check routine `Template Matching.chr` and switch to automatic mode by selecting **Automatic mode** in the **Operating modes** menu or by clicking the depicted icon. The status of the fieldbus card's outputs doesn't change since NeuroCheck's automatic mode is still inactive.

Make the **Control Panel**, which is used for controlling NeuroCheck in automatic mode, the active window to call commands interactively.



❸ If NeuroCheck is running, this will be signaled to the remote station via output signal **System alive**.

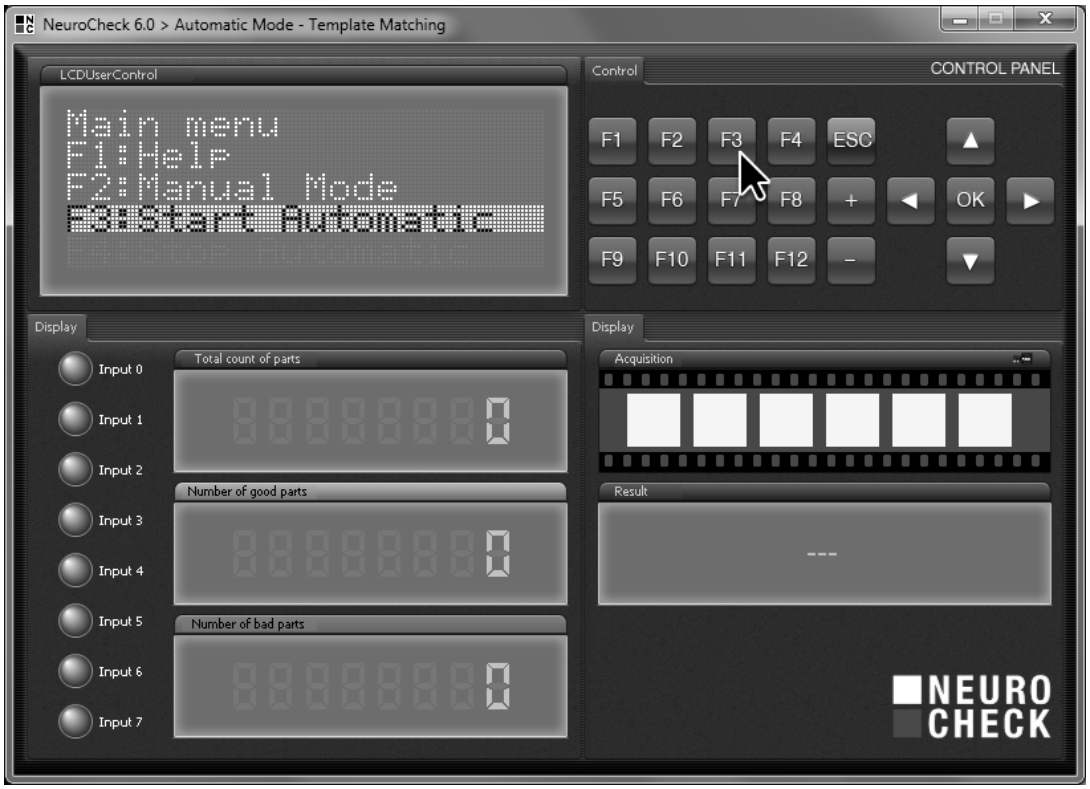


❹ If you have closed the remote station, you can restart it in the properties dialog.

5. To activate automatic mode, either click the *F3* button in the **Control Panel**, click *F3: Start Automatic* in the LCD display of the **Control Panel**, or press the F3 key.

Please note that both the entries of the LCD display and the assigned hot keys depend on the current configuration. The depicted entries and keyboard assignments are the default configuration after installing NeuroCheck.

Since the image in the current check routine is read from file, a warning is displayed upon activating automatic mode. When you are dealing with a real visual inspection station in an assembly line, you will but very rarely read images from file so this warning points to a faulty configuration. In this case reading the image from file is correct so you can exit the dialog by clicking **No**.



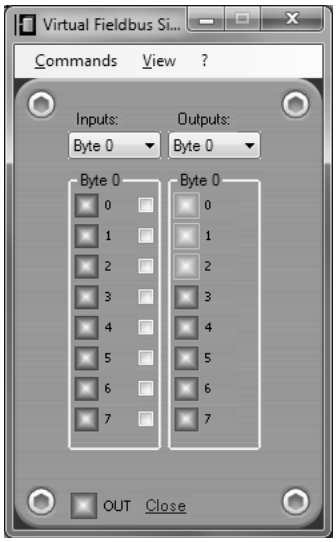
5 Click *F3* to enable automatic mode.

6. Since automatic mode is now active, the LED bit 1 of the virtual fieldbus' remote station is glowing because bit 1 is linked to the pin *Automatic mode enabled*, which is the signal target for the corresponding output signal.

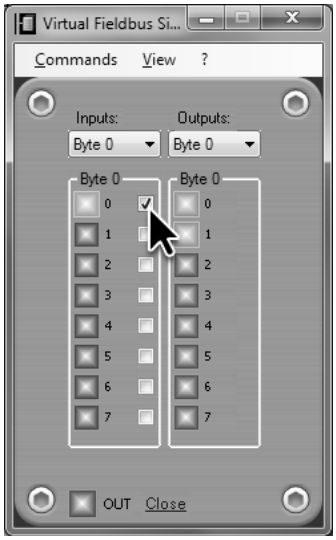
   When NeuroCheck's automatic mode is active, NeuroCheck is ready to respond to these input signals. Therefore, after activating automatic mode the output signal **System ready** is set so that the LED of bit 2 of the remote station is glowing as well.

7. In the remote station of the virtual fieldbus set the output 0 to set the input signal **Start check routine** for NeuroCheck. This output is mirrored on the input 0 of the virtual fieldbus device. In the first section of this chapter you saw that the pin *Start check routine* of the data format converter for fieldbus cards is linked to this input.

   Immediately after NeuroCheck has received the input signal **Start check routine**, the output signal **System ready** is reset so the LED for bit 2 isn't glowing any longer. Furthermore, the check routine is executed. At the same time, the windows of the **Control Panel** and **Process View** are updated.



6 At the remote station you can see that **Automatic mode enabled** and **System ready** are set.



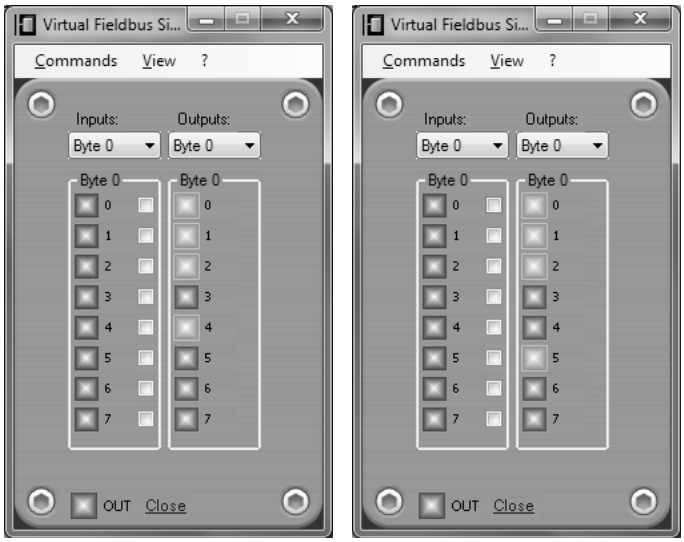
7 After setting the input signal **Start check routine**, the check routine is executed. During that time the system is not ready to receive another start signal.

❽ As soon as the check routine has completed and yielded a result, the output signal corresponding to the result is set. If the results of all checks of the check routine were *okay* and thus the overall result of the check routine is also *okay*, the output **Check routine result O.K.** is set so the LED for bit 4 of the remote station is glowing and the output **Check routine result not O.K.** is not set so the LED for bit 5 of the remote station does not glow. If one or more checks yield *not okay*, the result of the check routine is *not okay*. Correspondingly, the output signal **Check routine result not O.K.** is set and the output signal **Check routine result O.K.** is not set so that the LED of bit 4 does not glow, however, the LED for bit 5 does.

After setting the output signals for the check routine result, the output signal **System ready** is set. One inspection has completely run through and NeuroCheck is ready to receive the start signal for the next check routine.

Please note that checking the input signal **Start check routine** is level-based. If you leave the input signal **Start check routine** set at the remote station during the execution of the check routine then, after the end of the inspection, the next execution of the check routine is started immediately.

When receiving the next start signal, the output signal **System ready** is reset. Furthermore, the output signals **Check routine result O.K.** and **Check routine result not O.K.** of the previous inspection cycle are reset. The rest of the process is identical.



8 After executing the check routine, the system is ready and the result of the check routine is displayed (left: *okay*; right: *not okay*).

9. To stop automatic mode, either click the *F4* button in the Control Panel, click *F4: Stop Automatic* in the LCD display of the **Control Panel**, or press the F4 key. Automatic mode is only stopped after a possibly running check routine has run through so there can be a delay between clicking and stopping. Since automatic mode is not active anymore and thus the system no longer ready, the LEDs of bit 1 and bit 2 of the remote station of the virtual fieldbus are not glowing any more.

   Only when automatic mode is stopped it is possible to switch to manual mode. To switch to manual mode, either click the *F2* button in the Control Panel, click *F2: Manual mode* in the LCD display of the Control Panel, or press the F2 key.

9 Click *F4* to stop automatic mode and then *F2* to return to manual mode.

5.4 Automated inspection with type change

Goal

You want to use one system to execute various automated inspections, for example, to inspect various types of a work piece for which you need to initiate a type change from your master computer, i.e., a change of the inspection routine loaded in NeuroCheck in automatic mode.

Result

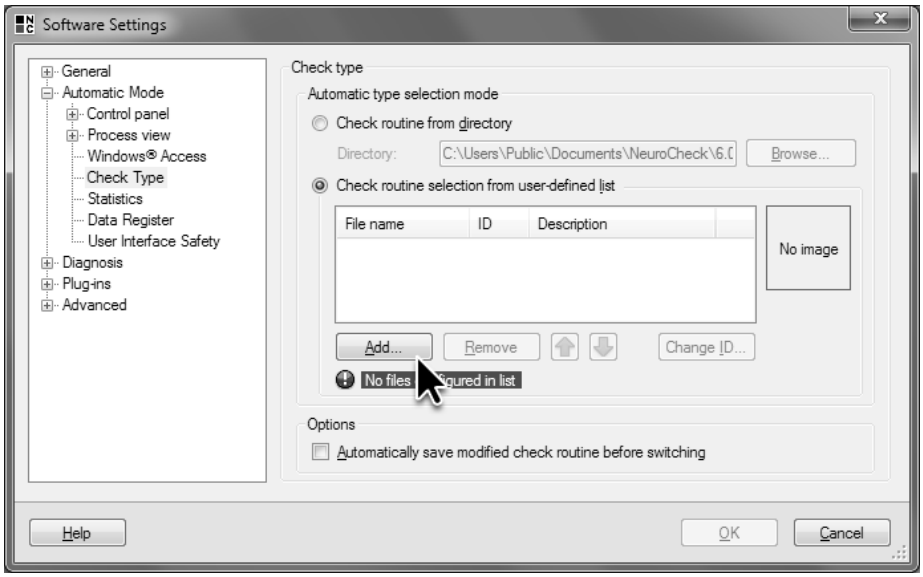
You will be able to configure check routines and remote control for a so-called type change and you will know the signals the master computer has to set for a type change.

Solution

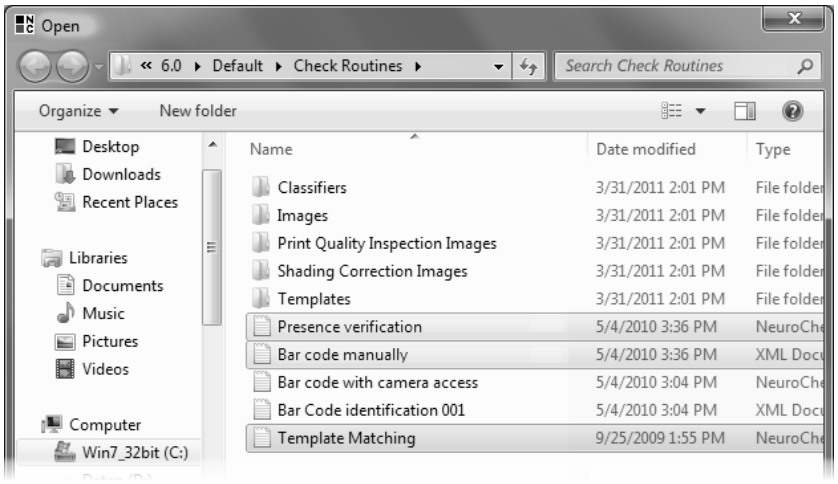
1. The first step is the selection of the check routines among which you want to switch. Select **Software Settings** from the **System** menu. The **Software Settings** dialog is the central dialog for configuring the operation and appearance of the NeuroCheck user interface. To select the check routines for the type change, in the tree view on the left side of the dialog click the "plus" next to **Automatic mode** so that the sub-entries are displayed. Click **Check Type**.

   Select the **Check routine selection from user-defined list** option so only the check routines you select manually will be loaded. Below the option you see the list of selected check routines that is currently empty. To add check routines to the list, click **Add…**.

2. In the **Open** dialog, select the check routines you created in the previous chapters `Bar code manually.chr`, `Presence verification.chr` and `Template Matching.chr`. By keeping the Ctrl key or ⇧ key pressed you can select several files simultaneously. Confirm your selection by clicking **Open**.



❶ Go to the **Check Routine Selection** page of **Software Settings** to select the check routines among which you can switch in automatic mode.

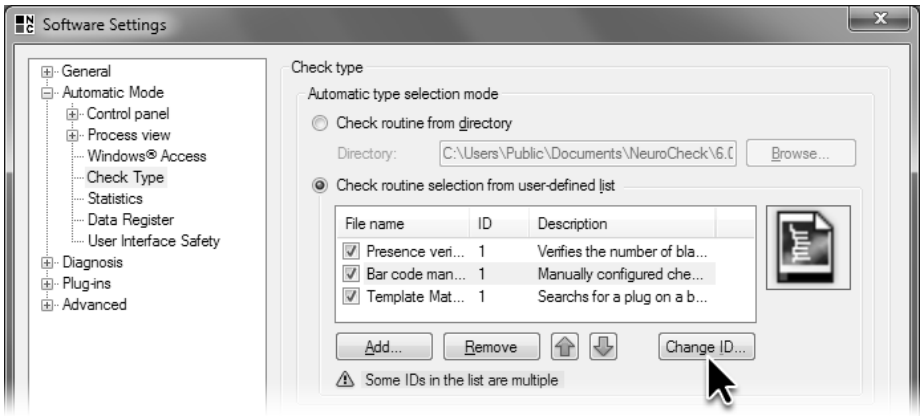


❷ Select the check routines among which you want to switch in automatic mode.

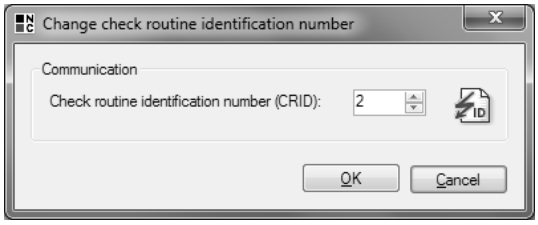
❸ The list will display the selected check routines. To identify the check routines when changing types, each check routine must have a unique check routine identification number **ID**. At first, the check routine identification number is 1. Since the selected check routines must have different identification numbers, select the check routine `Bar code manually.chr` and click the **Change ID…** button.

❹ In the dialog **Change check routine identification number** enter the new identification number 2 in the input field **Check routine identification number (CRID)**. Close the dialog by clicking **OK**. This saves the check routine file with the new identification number.

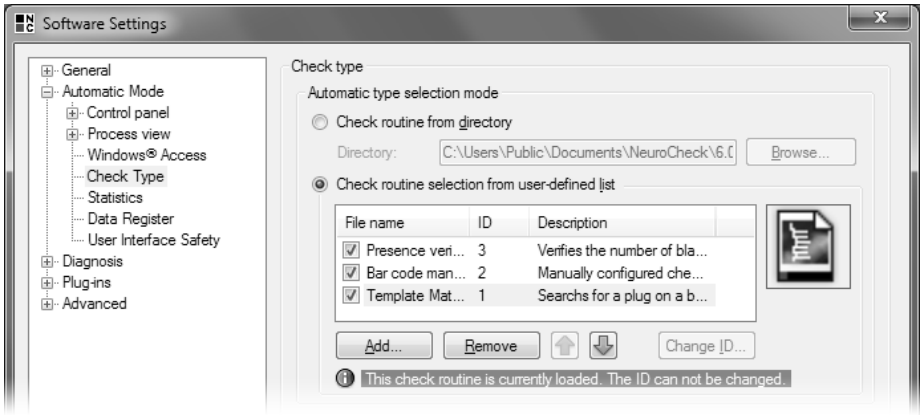
❺ Likewise change the identification number of the check routine `Presence verification.chr` to 3. If the check routine `Template Matching.chr` is still loaded, you cannot change the check routine identification number in the **Software Settings** dialog. Therefore, close the dialog by clicking **OK**. If the check routine `Template Matching.chr` is not loaded, open it.



❸ Click **Change ID…** to assign each check routine its own unique ID.



❹ Here you can enter the new ID of the check routine.



❺ The ID of the currently open check routine cannot be changed in the **Software Settings** dialog.

6 Change the identification number of the opened check routine `Template Matching .chr` in the **Check Routine Explorer** in the row with the depicted icon to 4. Save the check routine now. Now you have three check routines bearing the identification numbers 2, 3 and 4 available for type changes.

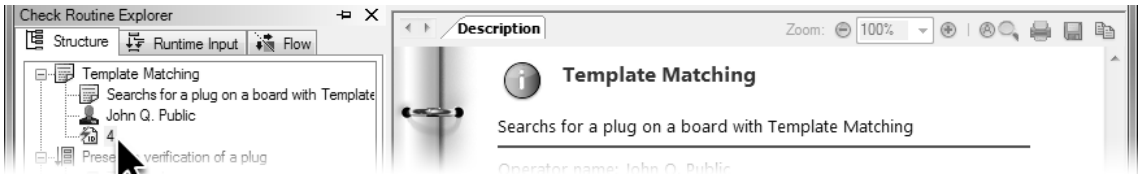
7 To change the inspection type, a command must be sent by the control system that can be received and processed by NeuroCheck. For this **Remote Control** has to be configured correspondingly. Select **Remote Control** in the **System** menu and select the **Input signals** tab.

The inspection type change is one of the functions that are available after you have configured the input signals for **Further commands**. For this you need to activate **Further commands**. To call this function you need a combination of three input signals.

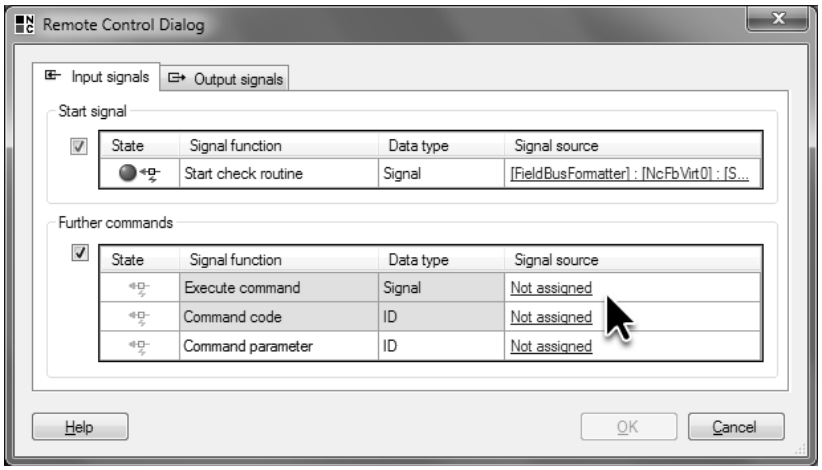
The first input signal is **Execute command**. It indicates that the other signals are set and NeuroCheck is to execute the configured function. Click the link **Not assigned** in the **Signal source** column to link the output signal via a pin to the virtual fieldbus.

8 In the **Connect signal [Execute command] to data format converter pin** dialog, select the process pin *Execute command* as source for the process signal of the same name. Exit the dialog by clicking **OK**.

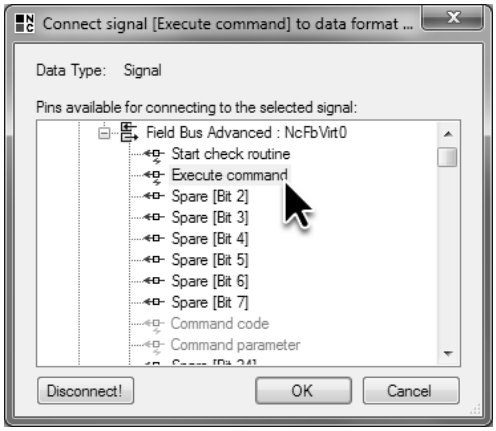
The second input signal is **Command code**. The input signal **Command code** indicates what function NeuroCheck will execute. Each addressable function is identified by a number, for example, the inspection type change is identified by code 1. For details concerning other functions and their corresponding codes, please refer to the manual or help function.



6 The ID of loaded check routines can be changed in the **Check Routine Explorer**.



7 Activate **Further Commands** on the **Input signals** tab of the **Remote Control** dialog.

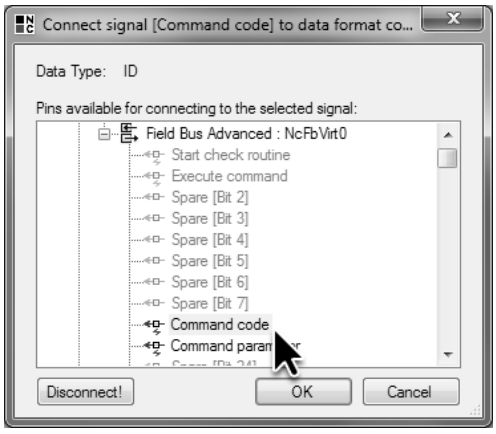


8 Link the input signal **Execute command** to the process pin of the same name.

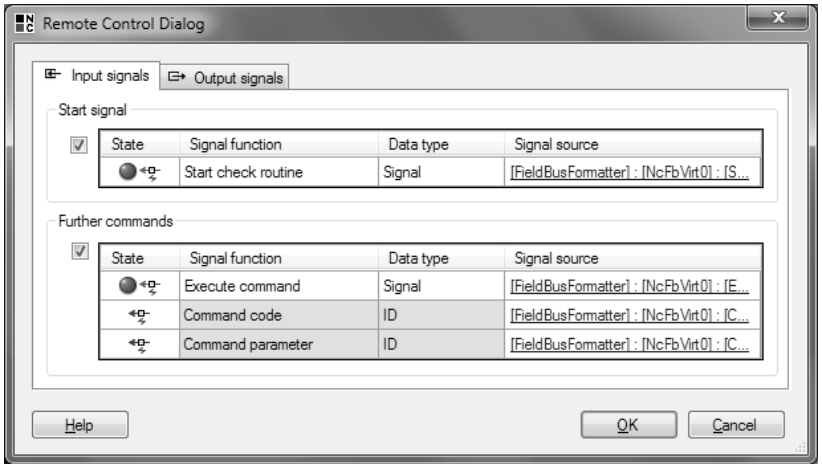
9 The input signal **Command code** must also be linked to a pin. For this click the link **Not assigned** in the **Signal source** column. The **Connect signal [Start check routine] to data format converter pin** dialog is opened. The first pins in the list of available pins are disabled since the input signal **Command Code** necessitates transmission of a code; however, the disabled pins can only transmit Boolean Values. The pins that can transmit a code are *Command Code* and *Command Parameter*. The data type of these pins is ID. Select *Command Code* as signal source for the input signal of the same name and close the dialog by clicking **OK**.

The third input signal is **Command Parameter** for transmitting additional information. In case of an inspection type change it indicates which check routine to open. The selection of the check routine is based upon the **ID** that was configured in the **Software Settings** dialog.

10 Link the input signal **Command parameter** to the pin of the same name, just as you have previously linked the input signal **Command Code**. After all three input signals have been assigned , exit the **Remote Control Dialog** by clicking **OK**.



9. Link the input signal **Command code** to the process pin of the same name.



10. After assigning all input signals for **Further Commands**, the inspection type change can take place.

⓫ If you want to view the links between the process pins *Execute command*, *Command Code* and *Command Parameter* and the respective inputs of the fieldbus card, select **Data Format Converter Manager** from the **System** menu or click the depicted icon. Click the **Properties** button and go to the **Input pins** page. On this page the links between pins and hardware inputs are listed. The input pin *Execute Command* is linked to input 1, the input pin *Command Code* is linked to inputs 8–15, and the input pin *Command Parameter* to inputs 16–23. Close the **Fieldbus Advanced– Data Format Converter Configuration** dialog and the **Data Format Converter Manager** dialog by clicking **Cancel**.

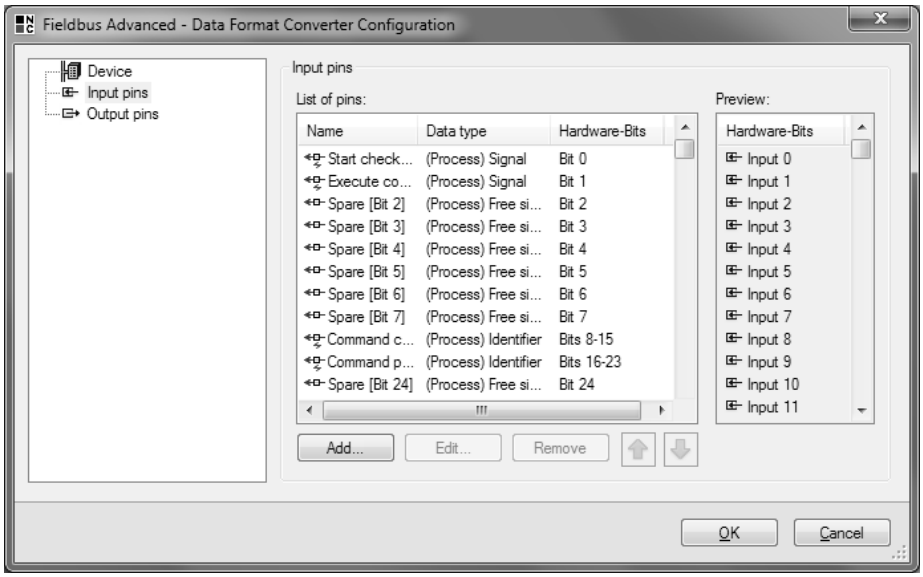
Switch to automatic mode and activate automatic by pressing the F3 key. If the output signal **System ready** is set, instead of the start signal a different function can also be executed, e.g. an inspection type change.

⓬ Drag the window of the remote station of the virtual fieldbus until it is large enough so you can see at least three bytes.

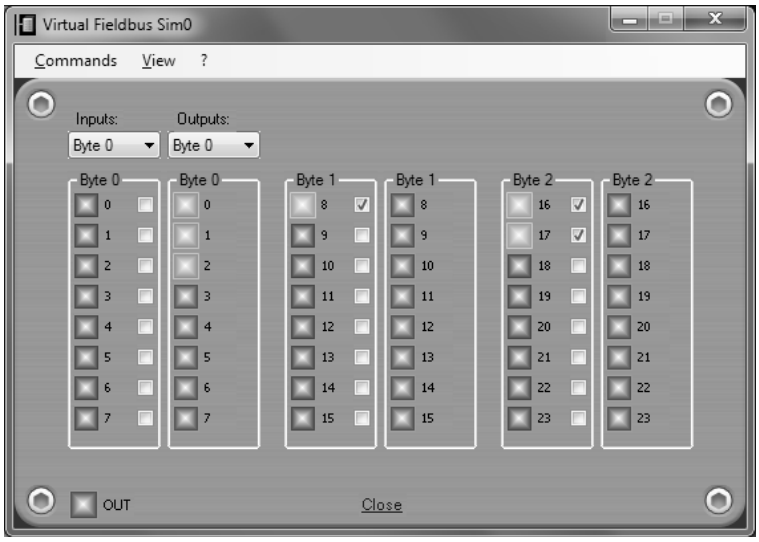
The IDs are transmitted in binary code. Thus the number 1 is the binary number 00000001, number 2 is 00000010, number 3 is 00000011, number 4 is 00000100, number 5 is 00000101 and so on.

To change a type the input signal **Command Code** must be set to 1. The inputs 8–15 of the virtual fieldbus are linked to the input signal **Command Code** via a process pin of the ID type. To set these inputs to 1, set output 8 of the remote station and reset all other outputs if they haven't already been reset.

The ID of the designated check routine must be set as input signal **Command Parameter**. Set 3 to select the check routine `Presence Verification.chr`. Set bits 16 and 17 and reset all other bits,



⓫ On the **Input pins** page of the **Fieldbus Advanced– Data Format Converter Configuration** dialog you can see, what inputs of the fieldbus card are connected to what pins.



⓬ At the remote station set the **Command code** (1) and the **Command parameter** (3) for an inspection type change.

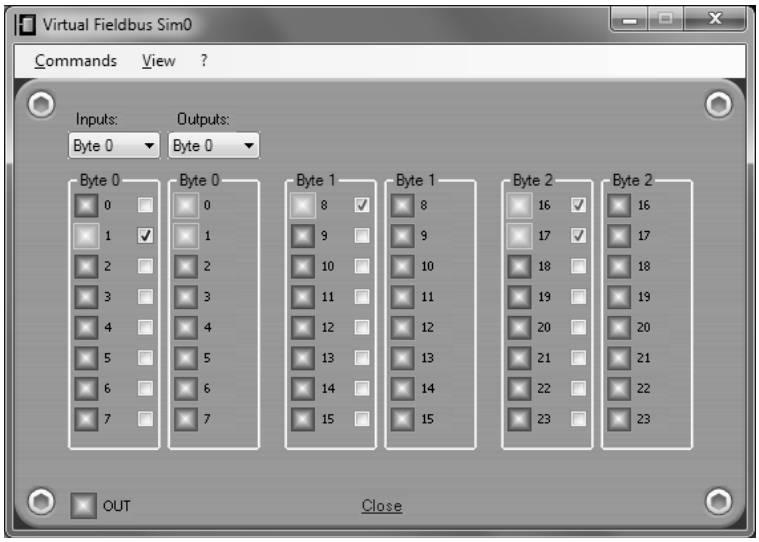
13. To trigger the type change, please set output 1 of the remote station. The corresponding input of the virtual fieldbus device is linked via an input pin to the input signal **Execute Command**.

    Please note that the signal **Execute Command** must not be set before **Command Code** and **Command Parameter** are set.

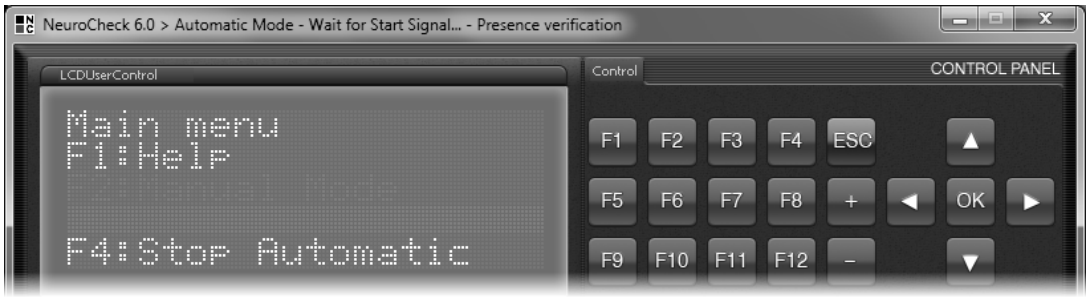
14. After setting this signal, NeuroCheck executes the configured function, i.e. it switches to the check routine `Presence verification.chr` as you can see from the window title of the **Control Panel**. Please note that the inspection type change can only take place if no inspection and no other command is being executed.

    During the inspection type change the output signal **System ready** is reset and the corresponding LED of input 2 of the remote station does not glow during type change.

    Reset the input signal **Execute Command** by resetting output 1 of the remote station. By setting the input signal **Start check routine** the inspection can be executed using the check routine `Presence verification.chr`.



13. Set the output to 1 to trigger the inspection type change. While the inspection type is changed, the output signal **System ready** is not set.

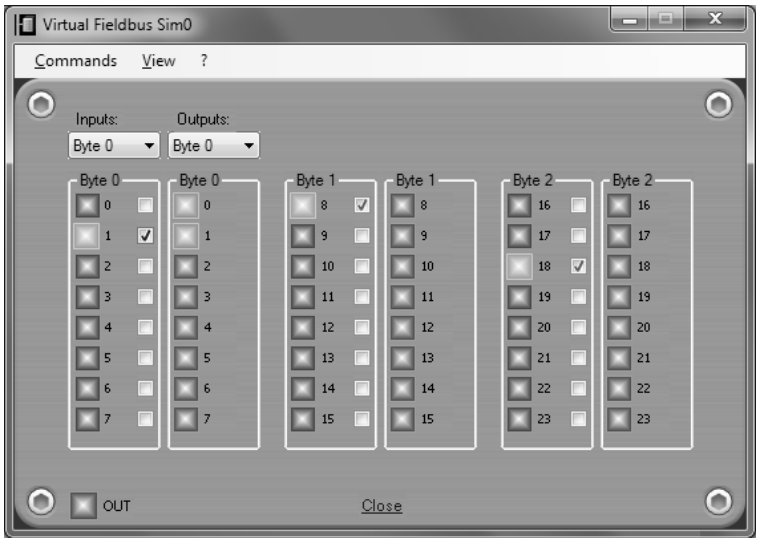


14. The title of the **Control Panel** displays the name of the new check routine.

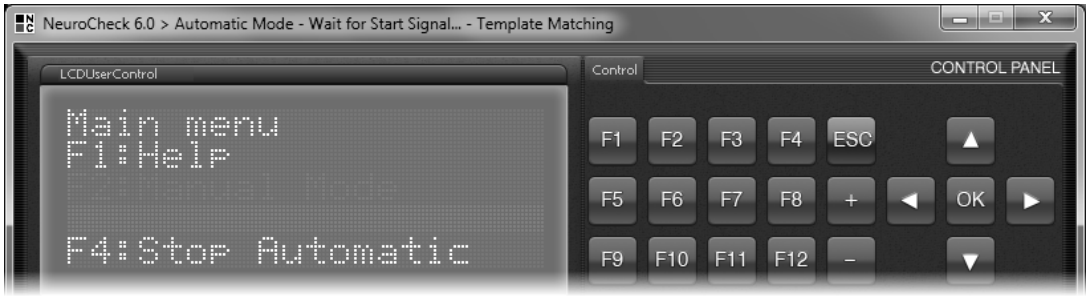
15. Set the input signal **Command Parameter** to 4 to load the check routine `Template Matching.chr`. Reset outputs 16 and 17, and set output 18, which is equivalent to binary 4.

16. If you set the input signal **Execute Command** again, the check routine `Template Matching.chr` is loaded. By setting the input signal **Start check routine** the inspection can be executed using the check routine `Template Matching.chr`.

    Stop automatic mode by pressing the F4 key and then change to manual mode by pressing the F2 key.



15 Set the outputs of the remote station to open the check routine with ID 4.



16 The title of the **Control Panel** displays the name of the new check routine.