NV4SCRIPT

Script option for NextView®4

Automate your NextView®4.

You want to automate and control measurement processes in NextView®4? Your measurement task is to be realized individually? A certain action needs to be set off in case of alarm? Very easy with the script option NV4SCRIPT for the data acquisition and analysis software NextView®4.

BASIC as basis. And much more.

The script language is based on the programming language BASIC to make the first steps easier. It has been supplemented by special commands and functions, however, to extend the standard commands and features of NextView®4 in an ideal way.

Event-oriented.

The programmed commands of a NextView®4 Script are executed when an event occurs, e.g. a button is clicked in NextView®4 or a trigger is set off. Events are either time-related or are provided by certain objects (e.g. slider, graph display).

Script editor.

Programming is done in the integrated script editor in NextView®4, which opens by double-clicking an event.

Help. For example.

Programming support is provided by an online help integrated in NextView®4 as well as by several programming examples in the documentation. Individual example scripts for bmcm measurement hardware can be downloaded at www.bmcm.de/us .

For all versions.

Whether Lite, Pro, or Analysis: NV4SCRIPT can be used with all versions of NextView®4. Installation is not necessary. To activate the script option, only the license number of NV4SCRIPT has to be entered in NextView®4.
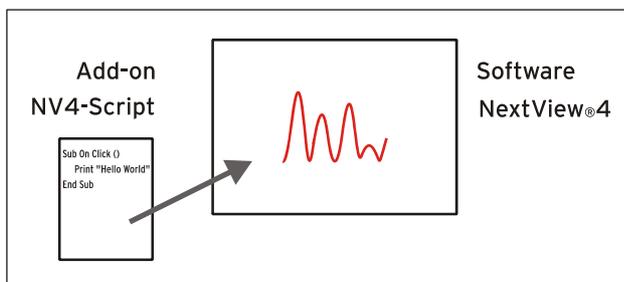
Functional diagram

1   Activation and licensing

The software NextView®4 (version: Lite, Professional, Analysis) must have been installed and licensed before activating NV4SCRIPT.

The add-on module NV4SCRIPT is activated directly in NextView®4.

Choosing the item "License" in the "Options" menu opens a window displaying the installed NextView®4 version as well as the corresponding license key.

To unlock the script option, click "Add".
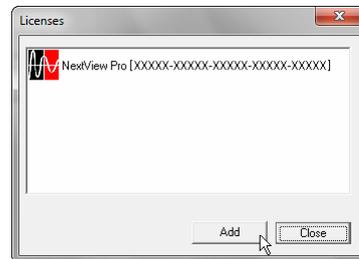
Figure 1

In the window "License Information", enter the serial number received at purchase of NV4SCRIPT to get a license key for this PC.

Now click the button "Request License".
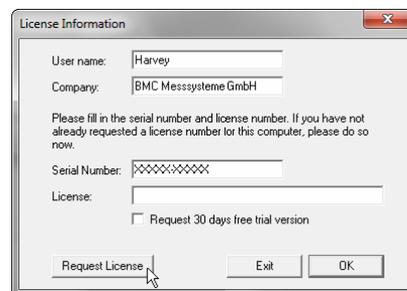
Figure 2

After accepting the licensing agreement, you can choose if you want to request the license number via e-mail or via our website (if necessary from a different PC).
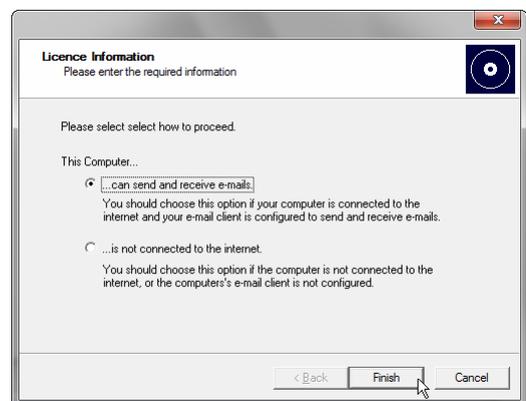
Figure 3

- **Request license number via e-mail:**

  If selecting this option, your e-mail program will automatically be opened. The displayed e-mail contains important data required for licensing and must not be changed. Send the e-mail (licence@bmcm.de) for the license number to be sent to your e-mail address.

- **Request license number via internet:**

  If the PC on which NextView®4 is installed has no connection to the internet, or if no e-mail program is configured, write down the serial number and the request code. Enter those on the webpage "NextView® Licensing" (accessible via: www.nextview.de) on any PC.

With an existing internet connection, choose the button "Open" instead. Your default browser will automatically show your licensing data on the webpage "NextView® Licensing".

Now enter the e-mail address to which you want the license number to be sent.
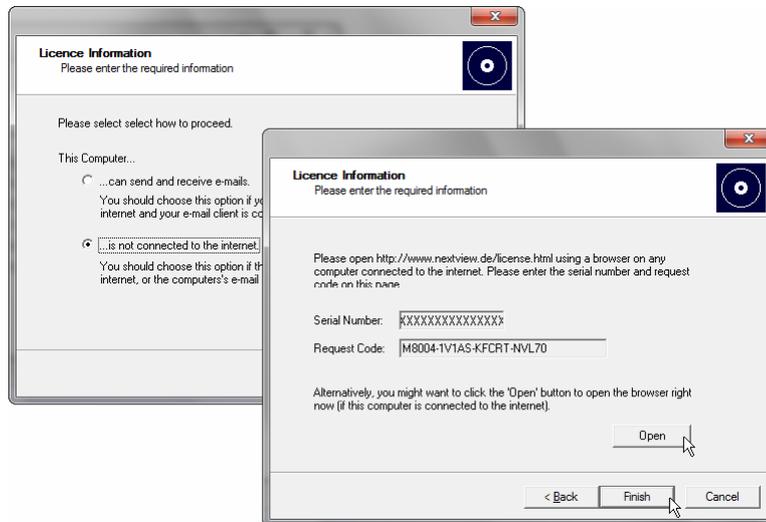
Figure 4

To complete the licensing process, enter the license number (25-digit) received by e-mail in the NextView®4.5 dialog "License Information" (see Figure 2) and press "OK".

After closing and reopening the "License" dialog in NextView®4, the item "NextView Script" will have been added to the list of licensed products (see Figure 1) and the extended script functionality is activated.

2 Create script

Script files have the file format **`*.nvs`**. They are always saved with the same file name and in the same directory as the NextView®4 project file (**`*.nvp`**) itself. Scripts can be used with any standard editor (not Word!).

The script editor integrated in NextView®4 is opened with <**STRG**>+**F12** or via the tab "Script" in the property bar on the left by double-clicking a provided NV4SCRIPT event (see chapter 3).
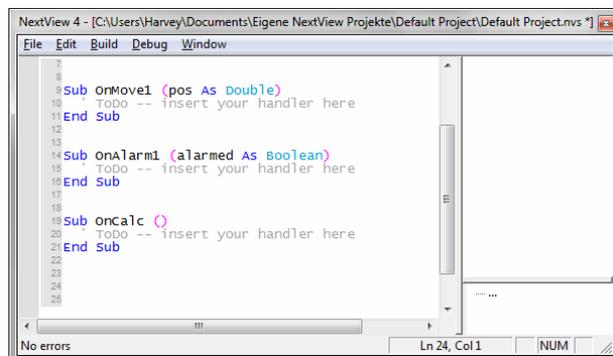
Figure 5

3 Available events

Script events in NextView®4 can be defined in relation to time or an object. If an event occurs, the corresponding commands of a NV4SCRIPT will be executed.

3.1 Time-based events

Time-controlled events are set off, for example, when loading, saving, or closing a project. A regularly occurring time event (event: **OnTimer**) is provided while NextView®4 is running to initialize DAQ systems or realize time-oriented sequential controls (e.g. rectangle signal at analog output).
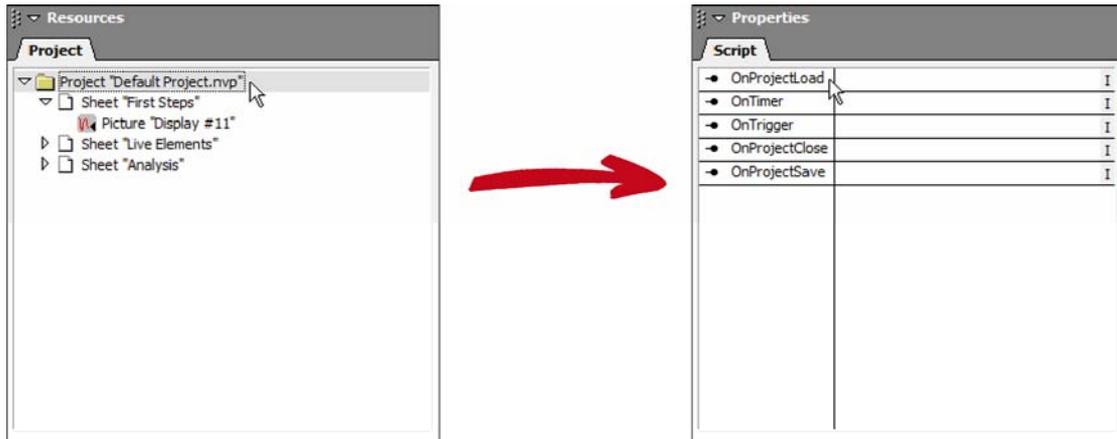
Figure 6

Time-based events can be viewed in tab "Script" of the property bar if you select the project ***.nvp** in the treeview of the "Project" tab (section "Ressources") first.

Time-based event	Description
OnProjectLoad	▶ Routine is executed when loading a project. Project-based initializations are one example in this connection.
OnTimer	▶ Routine is executed periodically in the defined interval. Time-controlled events can be processed in this connection.
OnTrigger	▶ Routine is executed after reaching the defined trigger.
OnProjectClose	▶ Routine is executed when closing a project.
OnProjectSave	▶ Routine is executed when saving a project.

3.2 Object-based events

With object-based events, it is possible to start user-controlled processes or to react to alarm conditions.

Simple controls can be realized by means of the objects switch and slider. For example, a relay could be turned on or off by a switch.

Alarm events can be realized by means of display elements, such as digital multimeters, text boxes, or level indicators. For instance, if a defined signal level is exceeded, an alarm will be set off and a certain reaction can be programmed, e.g. turning on a cooling system.

Figure 7

The tabs in the section "Properties" on the left side of the user interface show the current settings of the active object (click object to activate) on the sheet.

Objects which can trigger an event feature a "Script" tab in addition listing the programmable events of the object in the left column. The right column contains the corresponding name of the function to be executed if an event occurs (becomes editable by clicking it).

Object-based event	Object	
OnCalc	graph display	▶ Routine is executed when calculating key values of a signal (e.g.min, max) on tab "Cursor".
OnClick	switch	▶ Routine is executed when clicking a switch.
	slider	▶ Routine is executed when clicking or releasing the slider handle.
OnMove	slider	▶ Routine is executed when shifting the slider.
OnAlarm	static text, digital multimeter, level indicator	▶ Routine is executed when the alarm state changes.

4 Programming examples for NV4SCRIPT

The following programming examples demonstrate in an easy way how to use NV4SCRIPT.

4.1 Example "Switch"
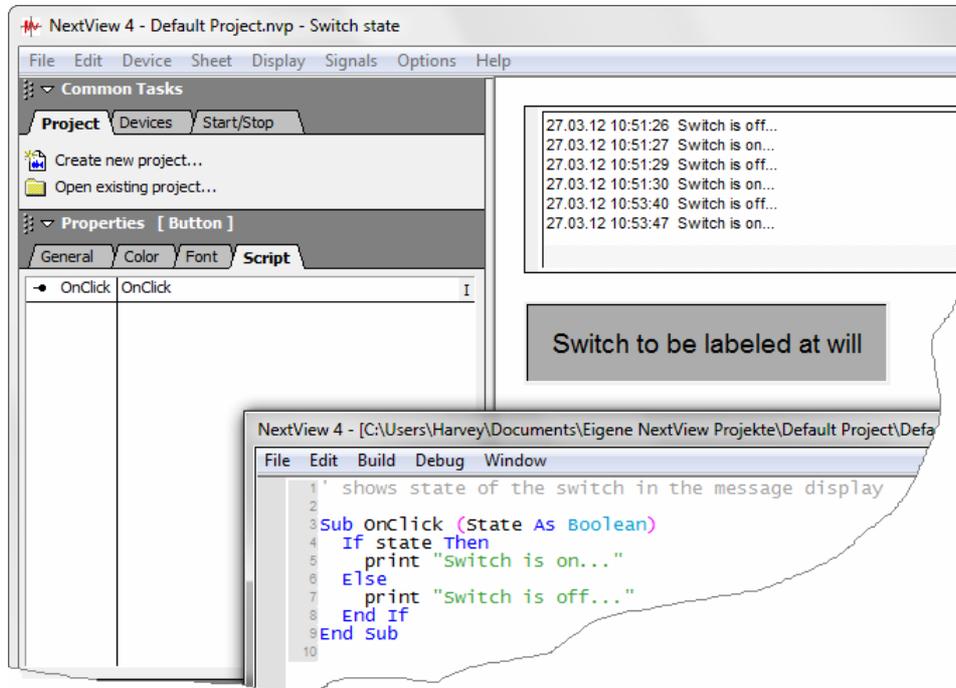
Figure 8

This example shows the current state of the switch (in or out) in the message display when being pressed.

Button	
| | ```
' shows state of the switch in the message display

Sub OnClick (State As Boolean)
  if state then
    print "Switch is on..."
  else
    print "Switch is off..."
  end if
End Sub
``` |

4.2 Example "Slider"

Figure 9

The message display indicates in this example when the slider handle has been clicked or released. If the defined limit of 70% (range 0..1; 0 = bottom, 1 = top) has been exceeded, a window opens containing a warning.

```
Slider          ' shows the slider position in the message display
                ' and opens a warning when exceeding a limit
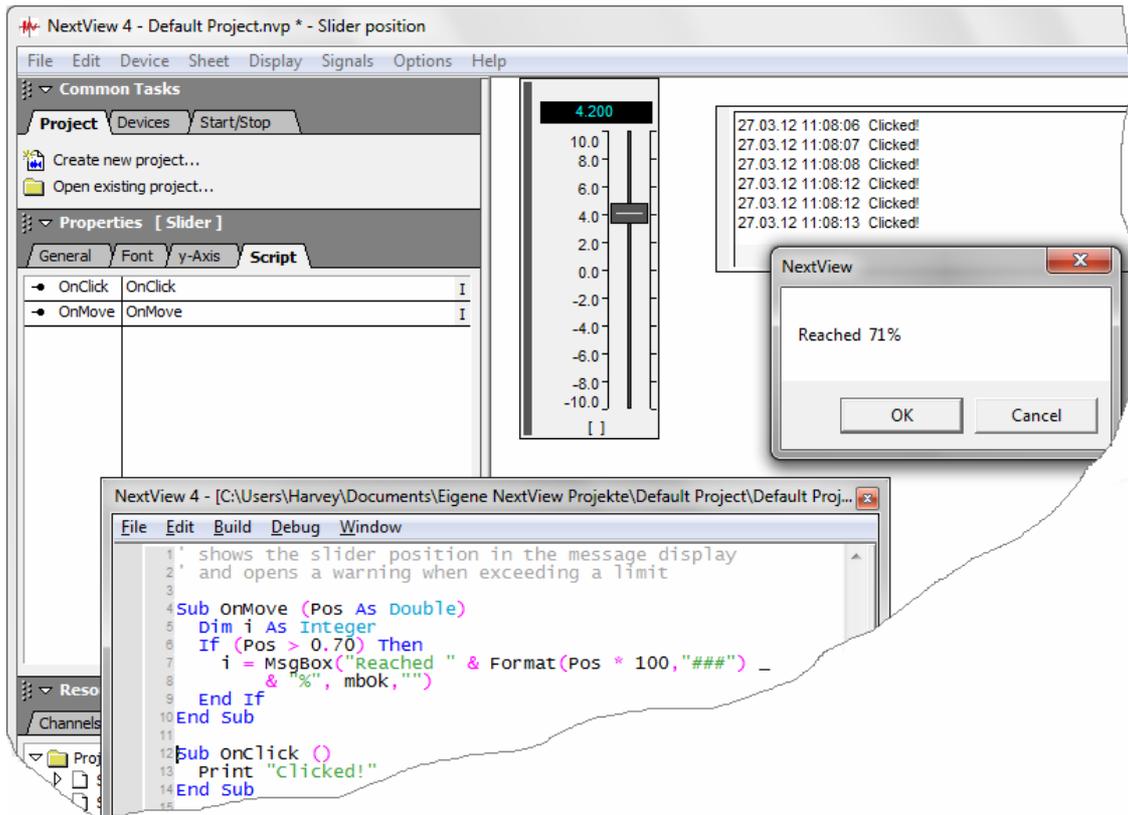
                Sub OnMove (Pos as Double)
                  Dim i as Integer
                  If (Pos > 0.70) then
                    i = MsgBox("Reached " & Format(Pos * 100,"###") _
                        & "%", mbOk,"")
                  End if
                End Sub
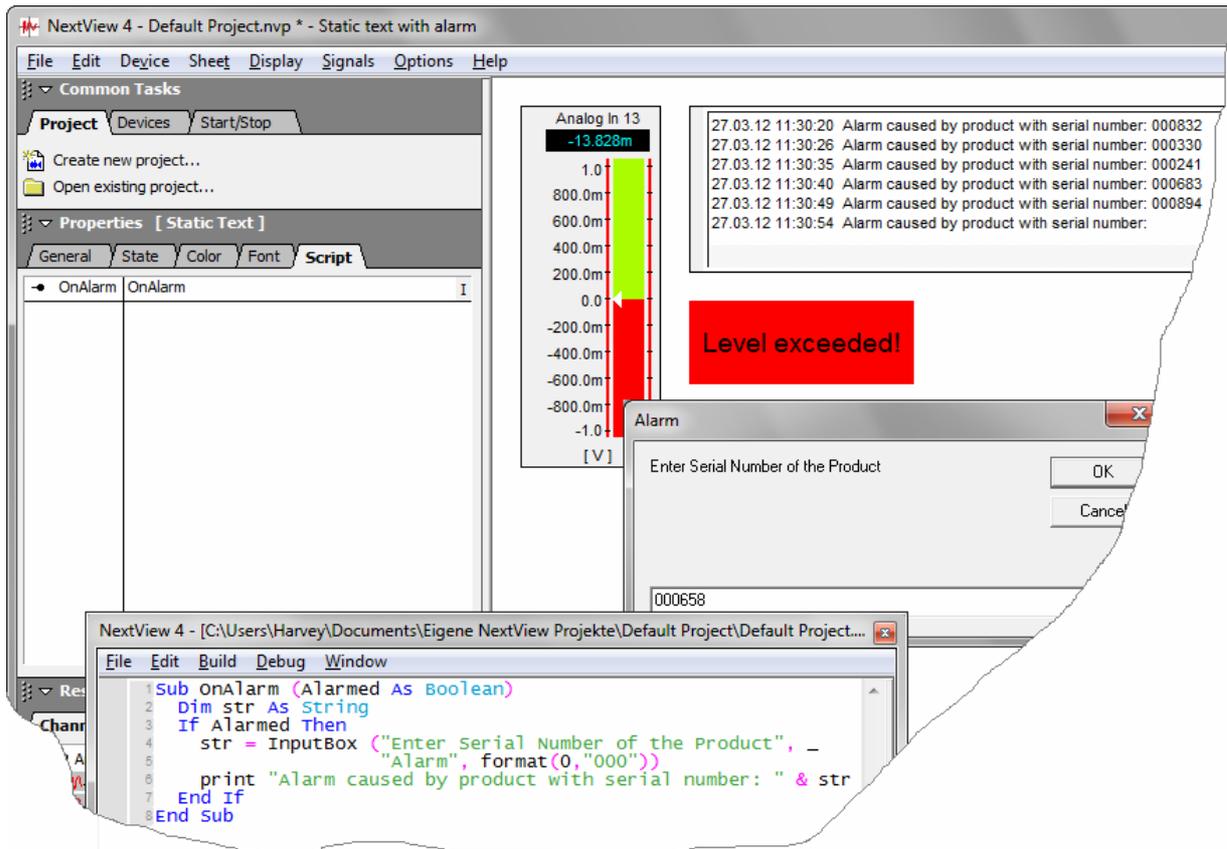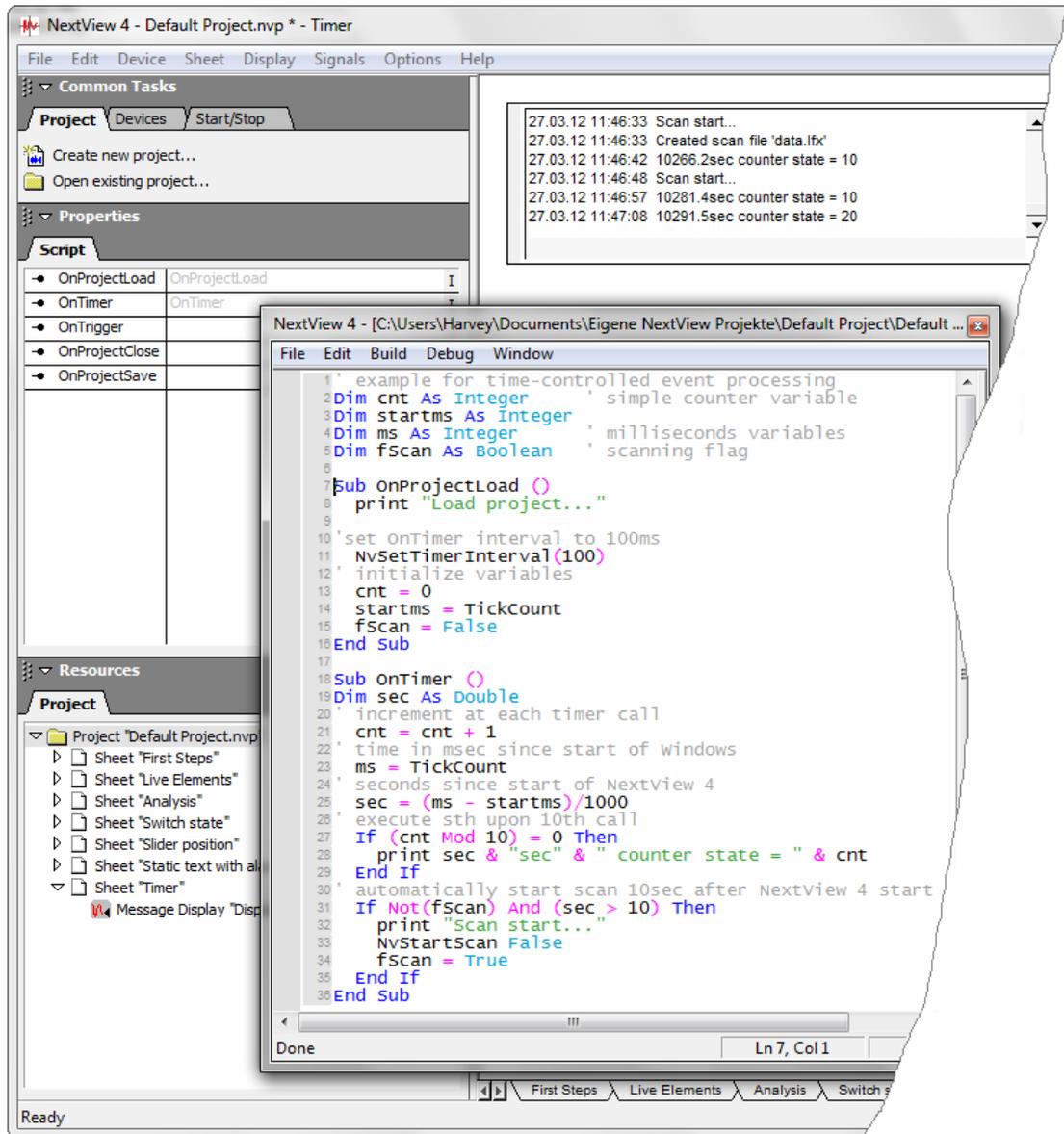
                Sub OnClick ()
                  Print "Clicked!"
                End Sub
```

4.3 Example "Static text"

Figure 10

As soon as the static text has reached the alarm state, a window opens prompting to enter the serial number of the device causing the alarm. The serial number will be displayed in the message display then.

| Static Text | ```
Sub OnAlarm (Alarmed As Boolean)
  Dim str As String
  If Alarmed Then
    str = InputBox ("Enter Serial Number of the Product", _
                    "Alarm", format(0,"000"))
    print "Alarm caused by product with serial number: " & str
  End if
End Sub
``` |
|---|---|

4.4 Example "OnProjectLoad" and "OnTimer"

Figure 11

The current time and counter state will be shown in the message display every 10th call of the **OnTimer** event. In addition, a scan starts about 10 seconds after opening NextView®4.

OnProjectLoad OnTimer	

	``
	' example for time-controlled event processing
	
	Dim cnt as Integer ' simple counter variable
	Dim startms as Integer
	Dim ms as Integer ' milliseconds variables
	Dim fScan as Boolean ' scanning flag
	
	Sub OnProjectLoad ()
	  print "Load project..."
	
	'set OnTimer interval to 100ms
	  NvSetTimerInterval(100)
	
	' initialize variables
	  cnt = 0
	  startms = TickCount
	  fScan = False
	End Sub
	
	Sub OnTimer ()
	Dim sec as Double
	
	' increment at each timer call
	  cnt = cnt + 1
	' time in msec since start of Windows
	  ms = TickCount
	' seconds since start of NextView 4
	  sec = (ms - startms)/1000
	
	' execute sth upon 10th call
	  if (cnt Mod 10) = 0 then
	    print sec & "sec" & " counter state = " & cnt
	  end if
	
	' automatically start scan 10sec after NextView 4 start
	  if Not(fScan) And (sec > 10) then
	    print "Scan start..."
	    NvStartScan False
	    fScan = True
	  end if
	End Sub
	``