

# NV4SCRIPT

## Skriptoption für NextView®4

### Automatisieren Sie Ihr NextView®4.

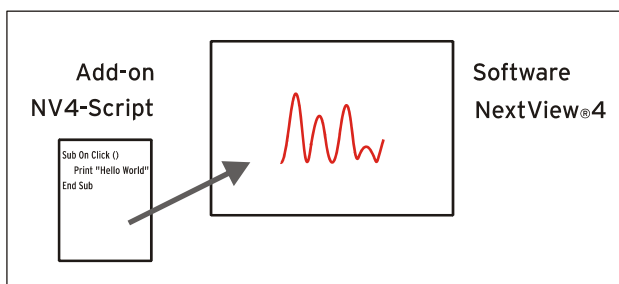
Sie wollen messtechnische Abläufe in NextView®4 automatisieren und steuern? Ihre Messaufgabe soll individuell realisiert werden? Im Alarmfall soll eine bestimmte Aktion ausgelöst werden? Mit der Skriptoption NV4SCRIPT für die Messdatenerfassungs- und Analysesoftware NextView®4 ganz einfach realisierbar.

### BASIC als Basis. Und noch mehr.

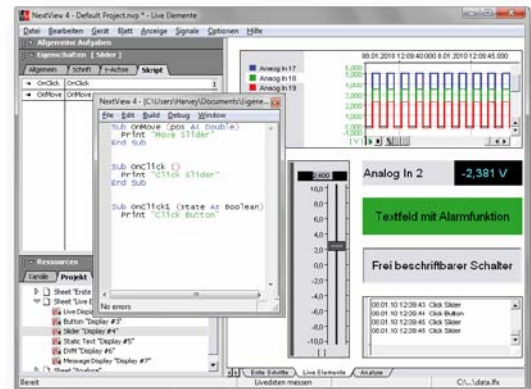
Um den Einstieg zu erleichtern, basiert die Skriptsprache auf der Programmiersprache BASIC. Diese wurde jedoch durch spezielle Befehle und Funktionen ergänzt, um die Standardbefehle und Features von NextView®4 auf ideale Weise zu erweitern.

### Ereignisorientiert.

Die programmierten Anweisungen in einem NextView®4 Script werden bei Eintreten eines Ereignisses ausgeführt, wenn zum Beispiel ein Schalter in NextView®4 gedrückt oder ein Trigger ausgelöst wird. Ereignisse sind entweder Zeit basiert oder werden durch bestimmte Objekte (z. B. Schieberegler, Signalanzeige) zur Verfügung gestellt.



Funktionsschaltbild



### Skripteditor.

Die Programmierung erfolgt im integrierten Skripteditor in NextView®4, der durch einen Doppelklick auf ein Ereignis geöffnet wird.

### Hilfe. Zum Beispiel.

Programmierunterstützung bietet die in NextView®4 integrierte Online-Hilfe, sowie verschiedene Programmierbeispiele in der Dokumentation. Einzelne Beispielskripte für die bmcm Messhardware stehen im Downloadbereich unter [www.bmcm.de](http://www.bmcm.de) zur Verfügung.

### Für alle Versionen.

Ob Lite, Pro oder Analyse: NV4SCRIPT kann mit allen Versionen von NextView®4 verwendet werden. Eine Installation ist nicht erforderlich. Um die Skriptoption freizuschalten, muss nur die Lizenznummer von NV4SCRIPT in NextView®4 eingegeben werden.

# 1 Freischaltung und Lizenzierung



**Vor der Freischaltung von NV4SCRIPT muss die Software NextView®4 (Version: Lite, Professional, Analyse) installiert und lizenziert worden sein.**

Die Freischaltung des Zusatzmoduls NV4SCRIPT erfolgt im Programm NextView®4 selbst.

Die Auswahl des Befehls "Lizenzen" im Menü "Optionen" öffnet ein Fenster, das die verwendete NextView®4 Version sowie die Lizenznummer anzeigt.

Um die Skriptoption freizuschalten, wählt man "Hinzufügen".

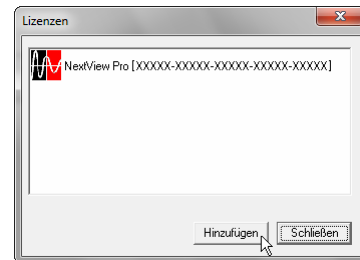


Abbildung 1

Im Dialog "Lizenzierung" wird nun die neue Seriennummer eingegeben, die Sie beim Erwerb von NV4SCRIPT erhalten haben, um die Lizenznummer für diesen PC anzufordern.

Klicken Sie nun auf die Schaltfläche "Lizenz anfordern".

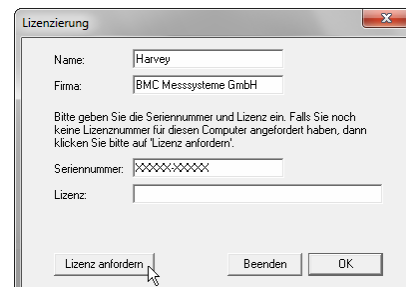


Abbildung 2

Nachdem Sie der Softwarevereinbarung zugestimmt haben, können Sie auswählen, ob Sie die Lizenznummer per E-Mail oder über unsere Website (ggf. auch von einem anderen PC aus) anfordern wollen.

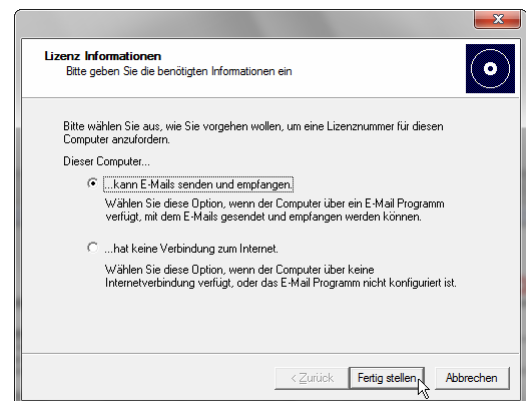


Abbildung 3

- **Lizenznummer per E-Mail anfordern:**

Bei Auswahl dieser Option wird automatisch Ihr E-Mail Programm geöffnet. Die angezeigte E-Mail enthält wichtige erforderliche Daten, die nicht verändert werden dürfen. Schicken Sie die E-Mail ab ([licence@bmcm.de](mailto:licence@bmcm.de)), damit die Lizenznummer an Ihre E-Mailadresse zurückgeschickt werden kann.

- **Lizenznummer per Internet anfordern:**

Besitzt der PC, auf dem NextView®4 installiert ist, keine Verbindung zum Internet oder ist kein E-Mail Programm eingerichtet, notieren Sie sich Seriennummer und Anforderungscode. Geben Sie diese auf der Seite "NextView® Lizenzierung" (erreichbar über: [www.nextview.de](http://www.nextview.de)) an einem beliebigen PC ein.

Bei einer bestehenden Internetverbindung wählen Sie stattdessen die Schaltfläche "Öffnen". Ihre Lizenzierungsdaten werden automatisch auf der Seite "NextView® Lizenzierung" in Ihrem Standardbrowser angezeigt.

Geben Sie nun die E-Mailadresse ein, an die die Lizenznummer geschickt werden soll.

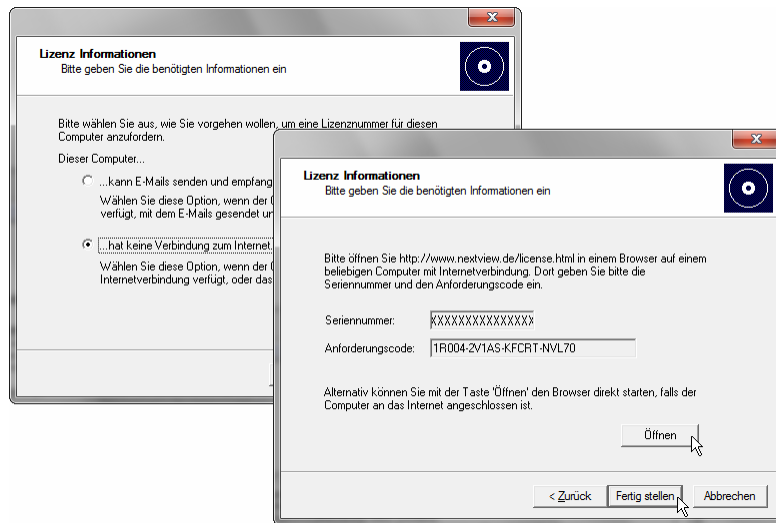


Abbildung 4

Um den Lizenzierungsvorgang abzuschließen, wird die per E-Mail erhaltene Lizenznummer (25-stellig) in den NextView®4 Dialog "Lizenzierung" (s. Abbildung 2) eingefügt und die Schaltfläche "OK" betätigt.

Nach Verlassen und erneutem Öffnen des Dialogs "Lizenzen" in NextView®4 befindet sich der Eintrag "NextView Script" in der Liste der lizenzierten Produkte (s. Abbildung 1) und die erweiterte Skriptfunktionalität ist freigeschaltet.

## 2 Skript erstellen



Skriptdateien besitzen das Format **\*.nvs** und werden immer unter dem gleichen Dateinamen und im gleichen Verzeichnis gespeichert, wie die NextView®4 Projektdatei (**\*.nvp**) selbst. Sie können mit jedem Standardeditor (kein Word!) bearbeitet werden.

Den in NextView®4 integrierten Skripteditor öffnet man mit **<STRG>+F12** oder über das TAB "Skript" in der Eigenschaftensleiste von NextView®4 durch Doppelklick auf ein verfügbares NV4SCRIPT Ereignis (s. Kap. 3).

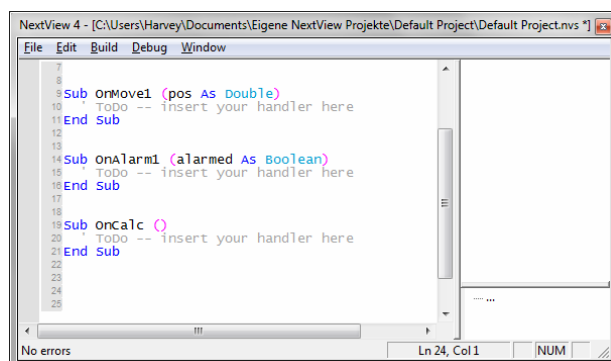


Abbildung 5

### 3 Verfügbare Ereignisse (Events)

Skript-Ereignisse (Events) in NextView®4 können Zeit basiert oder Objekt basiert definiert werden. Bei Eintreten eines Ereignisses werden die zugehörigen Anweisungen eines NV4SCRIPTs ausgeführt.

#### 3.1 Zeit basierte Ereignisse

Zeitgesteuerte Ereignisse werden z. B. beim Laden, Speichern oder Schließen eines Projekts ausgelöst. Ein regelmäßig auftretendes Zeitereignis (Event: **OnTimer**) steht während der Ausführung von NextView®4 zur Verfügung, mit dem sich auf einfache Weise Messsysteme initialisieren oder zeitliche Ablaufsteuerungen (z. B. Rechtecksignal an Analogausgang) realisieren lassen.

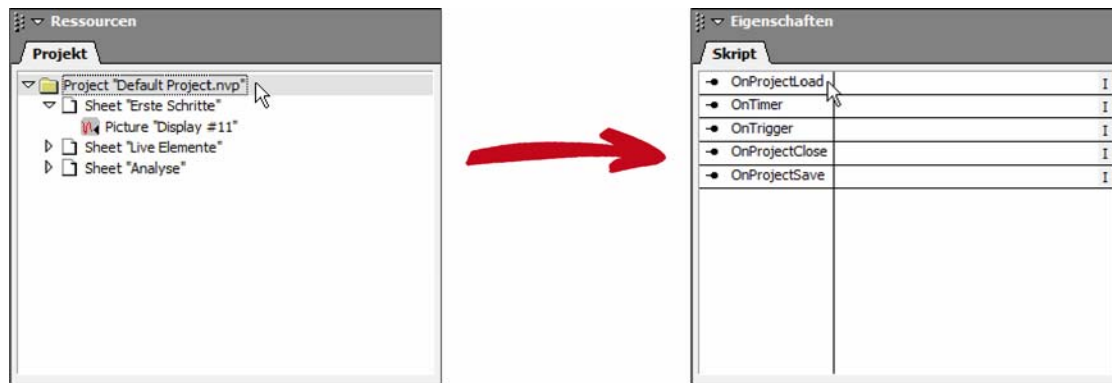


Abbildung 6

Zeitbasierte Events werden im TAB "Skript" der Eigenschaftensleiste angezeigt, wenn im TAB "Projekt" darunter (Abschnitt "Ressourcen") in der TreeView das Projekt **\*.nvp** selektiert wird.

Zeit basiertes Event	Beschreibung
<b>OnProjectLoad</b>	▶ Routine wird beim Laden eines Projekts ausgeführt. Projekt basierte Initialisierungen können hier erfolgen.
<b>OnTimer</b>	▶ Routine wird regelmäßig im eingestellten Intervall ausgeführt. Zeitgesteuerte Ereignisse können hier abgearbeitet werden.
<b>OnTrigger</b>	▶ Routine wird nach Erreichen des eingestellten Triggers ausgeführt.
<b>OnProjectClose</b>	▶ Routine wird beim Schließen eines Projekts ausgeführt.
<b>OnProjectSave</b>	▶ Routine wird beim Speichern eines Projekts ausgeführt.

### 3.2 Objekt basierte Ereignisse

Mit Objekt basierten Ereignissen kann man Benutzer gesteuerte Abläufe starten oder auf Alarmbedingungen reagieren.

Mit Hilfe eines Schalters oder Schiebereglers lassen sich einfache Steuerungen umsetzen. Zum Beispiel könnte ein Relais über einen Schalter ein- und ausgeschaltet werden.

Mit einem Digitalmultimeter, Textfeld oder Füllstandsanzeige sind Alarmereignisse realisierbar. So wird bei Überschreiten eines definierten Signalpegels beispielsweise ein Alarm ausgelöst und eine entsprechende Reaktion ist programmierbar, wie das Einschalten einer Kühlung.

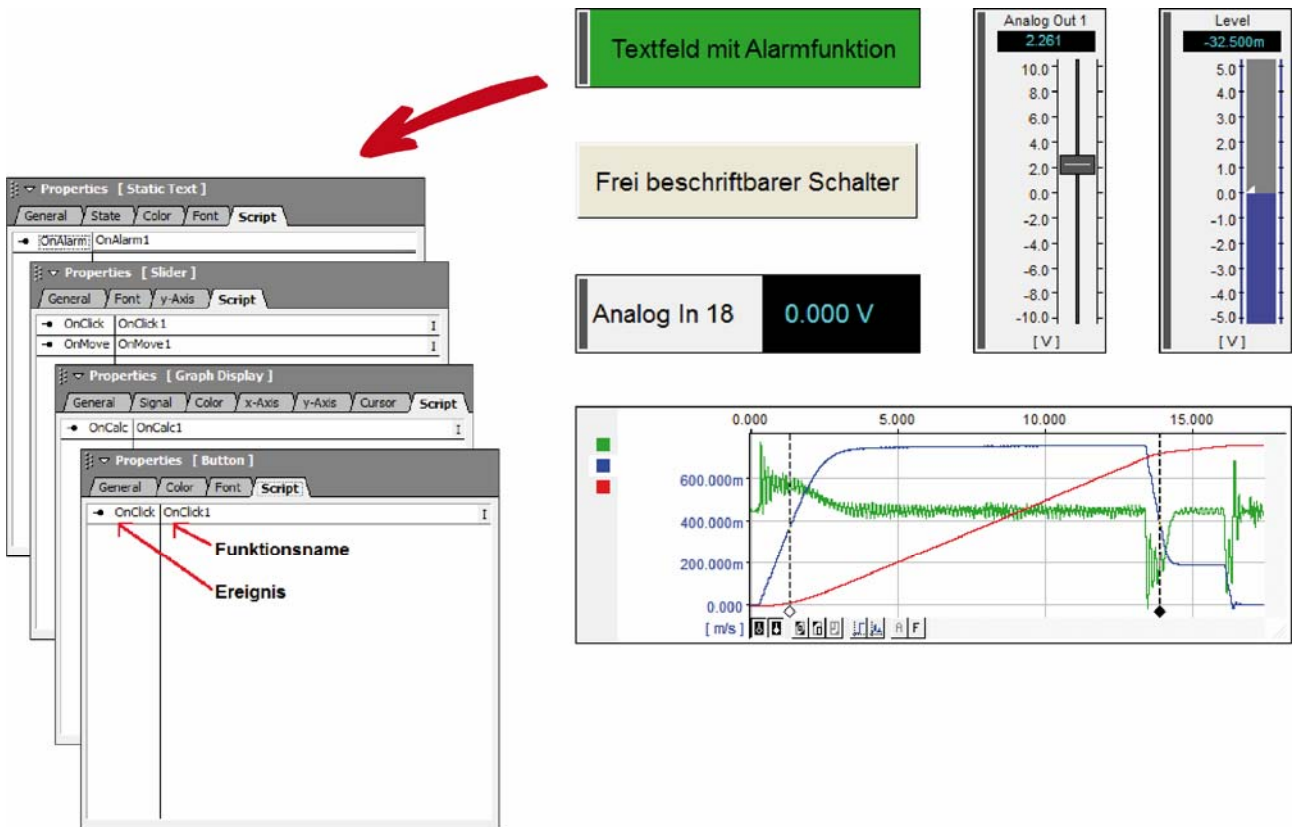


Abbildung 7

Die Eigenschaftenleiste von NextView®4 zeigt im Abschnitt "Eigenschaften" auf verschiedenen TABs die aktuellen Einstellungen des selektierten Objekts (Auswahl durch Anklicken) auf dem Blatt an.

Event auslösende Objekte enthalten zusätzlich das TAB "Skript", das in der linken Spalte die programmierbaren Ereignisse des Objekts auflistet. Die rechte Spalte enthält den zugehörigen Namen der Funktion, die bei Eintreten des Ereignisses ausgeführt werden soll (modifizierbar durch Anklicken).

Objekt basiertes Event	Objekt	
OnCalc	Signalanzeige	▶ Routine wird bei Berechnen von Signalkennwerten (z. B. Min, Max) im TAB "Cursor" ausgeführt.
OnClick	Schalter	▶ Routine wird bei Klicken eines Schalters ausgeführt.
	Schieberegler	▶ Routine wird bei Klicken u. Loslassen des Schiebereglers ausgeführt.
OnMove	Schieberegler	▶ Routine wird bei Verschieben des Schiebereglers ausgeführt.
OnAlarm	Textbox, Digitalmultimeter, Füllstandsanzeige	▶ Routine wird bei Änderung des Alarmstatus ausgeführt.

## 4 Programmierbeispiele für NV4SCRIPT

Die folgenden Programmierbeispiele zeigen auf einfache Weise die Anwendung von NV4SCRIPT.

### 4.1 Beispiel "Schalter"

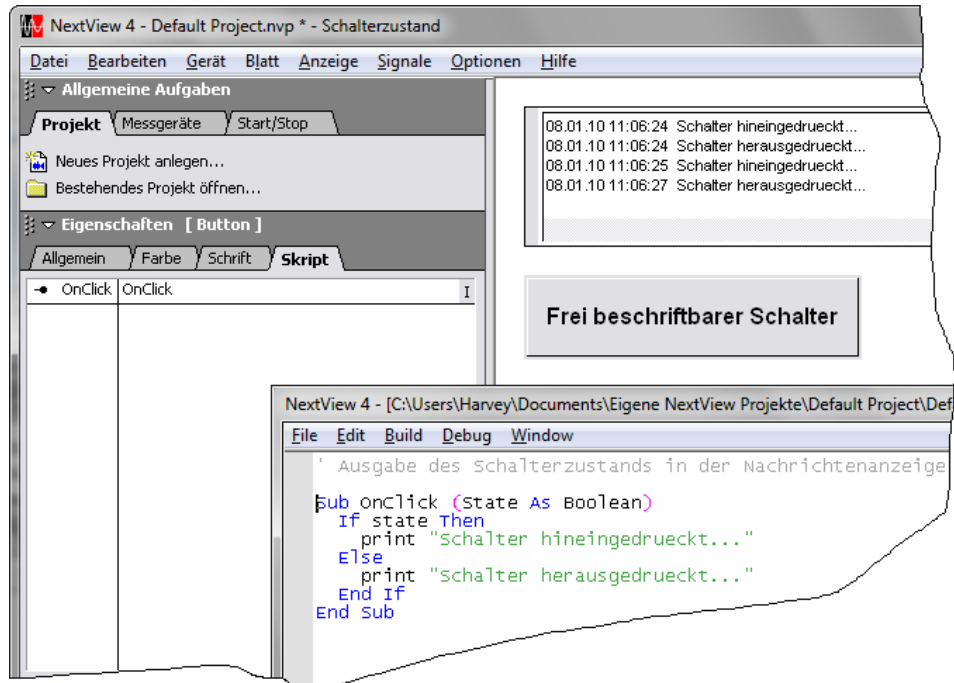


Abbildung 8

Dieses Beispiel zeigt in der Nachrichtenanzeige den aktuellen Zustand (hinein- bzw. herausgedrückt) des Schalters an, wenn dieser betätigt wurde.

<i>Button</i>	<pre>' Ausgabe des Schalterzustands in der Nachrichtenanzeige  Sub onClick (State As Boolean)   if state then     print "Schalter hineingedrueckt..."   else     print "Schalter herausgedrueckt..."   end if End Sub</pre>
---------------	---

## 4.2 Beispiel "Schieberegler"

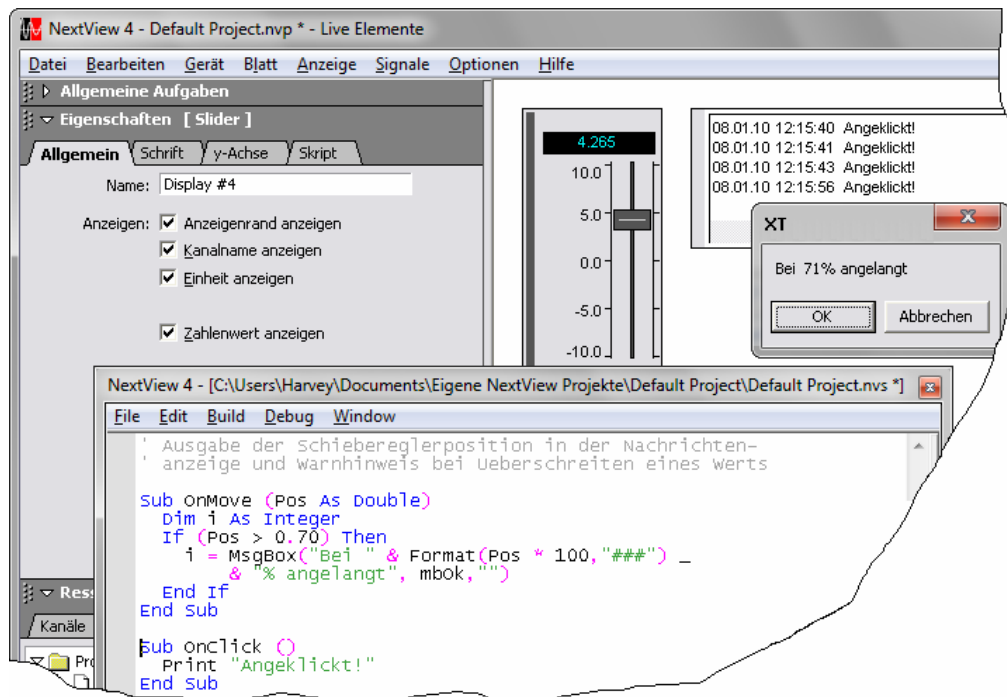


Abbildung 9

In diesem Beispiel wird in der Nachrichtenanzeige angezeigt, wenn die Schiebereglermarke angeklickt bzw. wieder losgelassen wurde. Bei Überschreiten des angegebenen Grenzwerts von 70% (Wertebereich 0..1; 0 = unten, 1 = oben) öffnet sich ein Fenster mit einem Warnhinweis.

*Slider*

' Ausgabe der Schiebereglerposition in der Nachrichtenanzeige  
' und Warnhinweis bei Ueberschreiten eines Werts

```

Sub OnMove (Pos as Double)
  Dim i as Integer
  If (Pos > 0.70) then
    i = MsgBox("Bei " & Format(Pos * 100,"###") _
      & "% angelangt", mbOk,"")
  End if
End Sub

Sub OnClick ()
  Print "Angeklickt!"
End Sub

```

### 4.3 Beispiel "Textfeld"

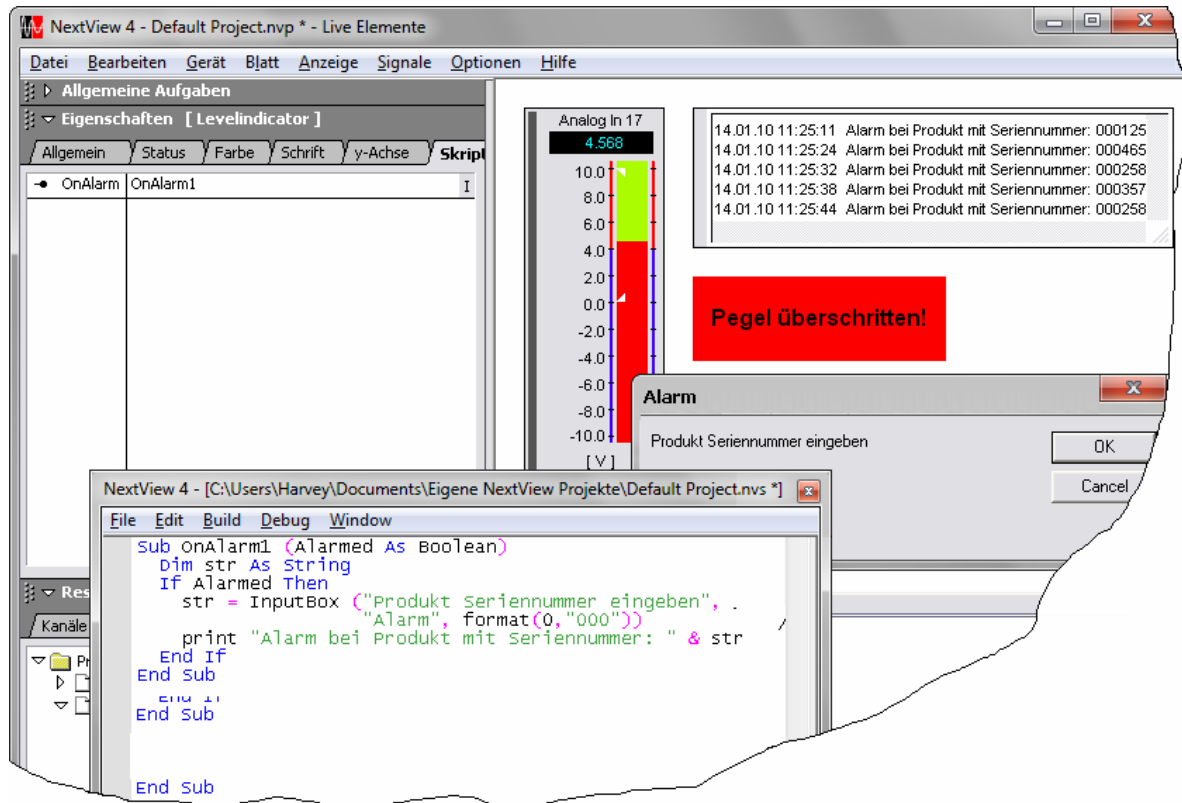


Abbildung 10

Sobald das Textfeld den Alarmzustand erreicht, öffnet ein Fenster, in dem die Seriennummer des Geräts, das den Alarm verursacht hat, einzugeben ist. Diese wird dann in der Nachrichtenanzeige ausgegeben.

<i>Static Text</i>	<pre> Sub OnAlarm (Alarmed As Boolean)   Dim str As String   If Alarmed Then     str = InputBox ("Produkt Seriennummer eingeben", _                   "Alarm", format(0,"000"))     print "Alarm bei Produkt mit Seriennummer: " &amp; str   End if End Sub </pre>
--------------------	--



### 4.4 Beispiel "OnProjectLoad" und "OnTimer"

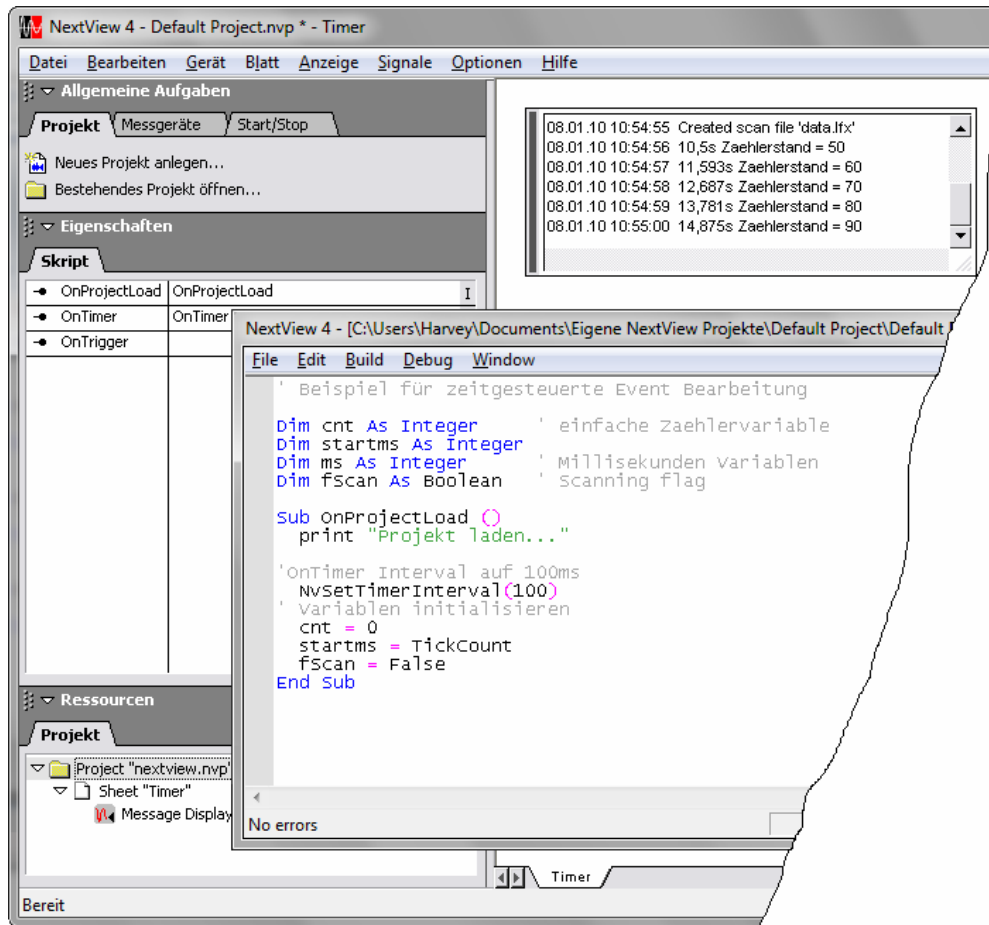


Abbildung 11

Hier werden bei jedem 10. Aufruf des **OnTimer** Events in der Nachrichtenanzeige die aktuelle Zeit und der Zählerstand ausgegeben. Etwa 10 Sekunden nach dem nächsten Öffnen von NextView®4 startet ein Scan.

*OnProjectLoad*  
*OnTimer*

```
' Beispiel für zeitgesteuerte Event Bearbeitung
Dim cnt as Integer      ' einfache Zaehlervariable
Dim startms as Integer
Dim ms as Integer      ' Millisekunden Variablen
Dim fScan as Boolean   ' Scanning flag

Sub OnProjectLoad ()
    print "Projekt laden..."

'OnTimer Interval auf 100ms
    NvSetTimerInterval(100)

' Variablen initialisieren
    cnt = 0
    startms = TickCount
    fScan = False
End Sub

Sub OnTimer ()
Dim sek as Double

' bei jedem Timeraufruf hochzaehlen
    cnt = cnt + 1
' Zeit in ms seit Start von Windows
    ms = TickCount
' Sekunden seit Start von NextView 4
    sek = (ms - startms)/1000

' beim 10. Aufruf etwas ausfuehren
    if (cnt Mod 10) = 0 then
        print sek & "s" & " Zaehlerstand = " & cnt
    end if

' 10s nach Start von NextView 4 Scan ohne Nachfrage starten
    if Not(fScan) And (sek > 10) then
        print "Scan Start..."
        NvStartScan False
        fScan = True
    end if
End Sub
```