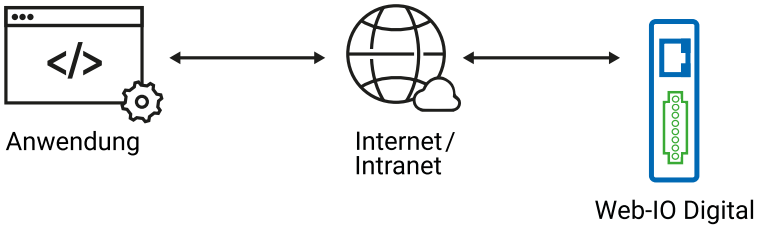
Applikation zum Web-IO Digital:

Automatisiertes Schalten mit VBScript und Batch-Jobs

Produktübersicht
Applikationsübersicht

Bei vielen kleineren Schaltaufgaben lohnt es nicht, eine spezielle Anwendungssoftware zu erstellen oder aufwändige Automatisierungstools zuzukaufen. Es reicht völlig aus, den gewünschten Schaltvorgang mit einem Klick auf ein Icon oder durch Eingabe eines Kommandos zu starten. Mit dem Web-IO Digital und kleinen VBScripten lassen sich solche Lösungen mit wenigen Befehlszeilen umsetzen.

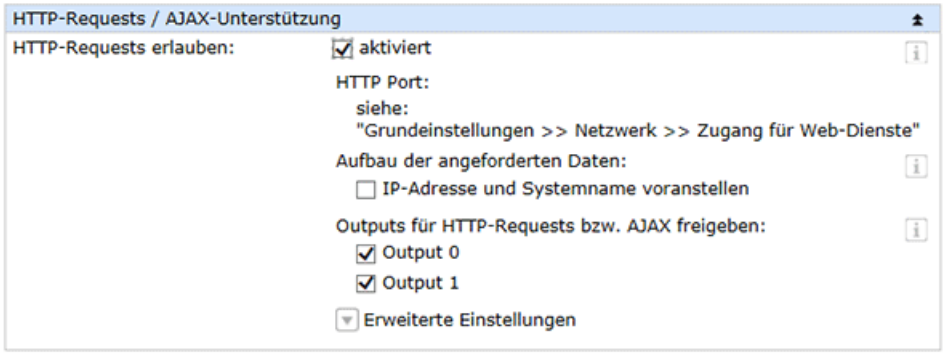


VBScript ist eine einfach zu programmierende Interpretersprache, die von den aktuellen Windows Versionen standardmäßig unterstützt wird. Wie der Name (Visual Basic Script) schon vermuten lässt, ist die Syntax von VBScript an die von Visual Basic angelehnt. Eine spezielle Entwicklungsumgebung ist für die Erstellung von VBScripten nicht erforderlich - es reicht ein Texteditor. Empfehlenswert ist der kostenlose Editor PSPAD (www.pspad.com), der eine Syntax-Hervorhebung für VBScript hat. Beim Abspeichern von VBScripten wird die Dateiendung ".vbs" verwendet.

Die im Folgenden beschriebenen Beispiele zeigen das Zusammenspiel von VBScript und Web-IO.

Vorbereitungen

- Web-IO mit Spannung versorgen und IOs verdrahten
- Web-IO mit dem Netzwerk verbinden
- IP-Adressen vergeben
- Beim Web-IO im Bereich *Kommunikationswege >> Web-API* den Punkt *HTTP-Request erlauben* aktivieren und die *Outputs zum Schalten freigeben*



1. Einen Output mit VBScript setzen

Das folgende Script setzt bei Aufruf Output 0 des Web-IO mit der IP-Adresse 10.40.22.101.

```
' VB Script Document
option explicit

Dim objHttp
Set objHttp = WScript.CreateObject("WinHttp.WinHttpRequest.5.1")
If objHttp Is Nothing Then Set objHttp = WScript.CreateObject("WinHttp.WinHttpRequest")
  objHttp.Option(4) = 256 + 512 + 4096 + 8192
  objHttp.SetTimeouts 0, 5000, 10000, 10000
  objHttp.Open "GET", "http://10.40.22.101/outputaccess0?PW=&State=ON&", FALSE
  objHttp.setRequestHeader "User-Agent", WScript.ScriptName
  objHttp.Send ""
If Not (objHttp.statusText = "OK") Then
  WScript.Echo "Error: " & objHttp.statusText
  WScript.Quit 1
Else
  WScript.Echo objHttp.ResponseText
End If
```

Kernstück dieses Scriptes ist das WinHttp-Objekt, welches benutzt wird, um Befehle an das Web-IO zu senden. Die eigentliche Befehlsübergabe findet mit *objHttp.Open* statt.

`http://10.40.22.101/outputaccess0?PW=&State=ON&` bestimmt, dass Output 0 von Web-IO 10.40.22.101 den Zustand ON annehmen soll.

Mit objHttp.ResponseText wird die Antwort des Web-IO eingelesen. Im Fall des Kommandos *outputaccess* gibt das Web-IO einen

String bestehend aus dem Wort *output*, einem Semikolon und dem Output-Status zurück.

Beispiel *output;1*

Eine ausführliche Beschreibung der möglichen Web-IO Kommandos ist in der Anleitung zum Web-IO ab Seite 102 zu finden.

2. VBScripte aus Batch-Jobs aufrufen

Batch-Jobs unter Windows sind im ursprünglichen Sinn dafür gedacht, Windows Kommandos automatisiert auszuführen. Ein klassisches Beispiel ist auch die Autoexec.bat in älteren DOS Umgebungen . In die Batch-Datei wird eine Liste mit Befehlen eingetragen, die bei Aufruf der Liste nacheinander abgearbeitet werden.

In Windows-Systemen können so auch Scripte und Programmaufrufe automatisiert werden. Sollen z.B. Outputs verschiedener Web-IOs mit einem Aufruf geschaltet werden, lässt sich auch das mittels eines Batch-Jobs realisieren.

Die entsprechenden Namen der VBScripte müssen dazu in der Batch-Datei untereinander geschrieben werden. Bei Scripten wie dem oben aufgeführten ist es nötig, für jede Schaltaufgabe ein eigenes Script zu erstellen.

Gerade in Batch-Jobs macht es daher mehr Sinn, ein universell benutzbares Script zu erstellen, dem bei Aufruf über Zusatzparameter übergeben wird, wie die Schaltaufgabe aussieht. Dem im Folgenden gezeigten VBScritp können bei Aufruf diese Parameter übergeben werden:

IP	IP-Adresse des Web-IO
PORT	TCP-Port des Web-IO *Dieser Parameter ist optional; wird er nicht übergeben, benutzt das Script Port 80*
PW	Administrator oder Operator Passwort des Web-IO *Dieser Parameter ist optional; wird er nicht übergeben, arbeitet das Script ohne Passwort*
MASK	gibt in hexadezimaler Schreibweise an, welche Outputs gesetzt werden sollen. *Dieser Parameter ist optional; wird er nicht übergeben, arbeitet das Script mit allen Outputs*
STATE	gibt in hexadezimaler Schreibweise an, in welchen Zustand die Outputs gesetzt werden sollen

Der Aufruf des Scriptes sieht so aus:

```
setoutput.vbs IP=<IP address> [PORT=<portno.>] [PASSWORD=<password>] [MASK=<hex value>] STATE=<hex value>
```

```
' VB Script Document
option explicit

Dim objArgs, strArg, strArgall
Dim IP, PORT, PASSWORD, MASK, STATE, URLStr
Dim objHttp
Set objArgs = WScript.Arguments

'# Make sure that script starts as console application (best way "cscript //h:cscript")
Dim WshShell : Set WshShell = WScript.CreateObject("WScript.Shell")
If Right(WScript.FullName, Len(WScript.Fullname) - Len(WScript.Path) -1) _ = "WScript.exe" Then
  For each strArg in objArgs
    strArgall = strArgall & " " & strArg
  next
  WshShell.Run "cmd /k cscript " & Chr(34) & WScript.ScriptFullName _ & Chr(34) & strArgall
  WScript.Quit
End If

'# Check if there are Parameters
If WScript.Arguments.count < 1 Then
  WScript.Echo "*********************************"
  WScript.Echo "* Not enough arguments *"
  WScript.Echo "*********************************"
  WScript.Echo ""
  WScript.Echo "Syntyx: setoutput.vbs
  IP=<IP address> [PORT=<portno.>] _
  [PASSWORD=<password>] [MASK=<hex value>] STATE=<hex value>"
  WScript.Quit
End If

'# Read the Parameters
for each strArg in objArgs
  If Left(strArg,3) = "IP=" then
    IP = Mid(strArg, 4, Len(strArg) - 3)
  End If

  If Left(strArg,5) = "PORT=" then
    PORT = Mid(strArg, 6, Len(strArg) - 5)
  End If

  If Left(strArg,9) = "PASSWORD=" then
    PASSWORD = Mid(strArg, 10, Len(strArg) - 9)
  End If

  If Left(strArg,5) = "MASK=" then
    MASK = Mid(strArg, 6, Len(strArg) - 5)
  End If

  If Left(strArg,6) = "STATE=" then
    STATE = Mid(strArg, 7, Len(strArg) - 6)
  End If
```

```
Next

'# Mount the command String
If IP <> "" then
  URLStr = "http://" & IP
else
  WScript.Echo "***********************************"
  WScript.Echo "* Not enough arguments : IP *"
  WScript.Echo "***********************************"
End If

If PORT <> "" then
  URLStr = URLStr & ":" & PORT
End If

URLStr = URLStr & "/outputaccess?PW=" & PASSWORD & "&"

If MASK <> "" then
  URLStr = URLStr & "Mask=" & MASK & "&"
End If

If STATE <> "" then
  URLStr = URLStr & "State=" & STATE & "&"
else
  WScript.Echo "***********************************"
  WScript.Echo "*  Not
  enough arguments : STATE   *"
  WScript.Echo "***********************************"
End If

'# Send the Command String via HTTP object
Set objHttp = WScript.CreateObject("WinHttp.WinHttpRequest.5.1")
If objHttp Is Nothing Then Set objHttp = WScript.CreateObject("WinHttp.WinHttpRequest")
objHttp.Option(4) = 256 + 512 + 4096 + 8192
objHttp.SetTimeouts 0, 5000, 10000, 10000
objHttp.Open "GET", URLStr, FALSE
objHttp.setRequestHeader "User-Agent", WScript.ScriptName
objHttp.Send ""
If Not (objHttp.statusText = "OK") Then
  WScript.Echo "Error: "& objHttp.statusText
  WScript.Quit 1
else
  WScript.Echo objHttp.ResponseText
End If
```

Nach Ausführung der gewählten Aktion gibt das Script den Status der Outputs zurück.

z.B. *output:0100*

Die Statusmeldung besteht aus dem Wort output, einem Semikolon und dem Output-Status in hexadezimaler Schreibweise.

TIPP: Für die Ausführung von VBScripten ist der Windows Scripting Host verantwortlich, der in zwei Varianten auf jedem Windows PC vorhanden ist. Die beiden Varianten WScript und CScript unterscheiden sich in erster Linie durch die Art der Datenausgabe. Wenn nicht anders konfiguriert, werden die Scripte von WScript verarbeitet und Textmeldungen in einer Windows Dialogbox ausgegeben. Das hat den Nachteil, dass jede Meldung vom User quittiert werden muss, damit das Script weiterläuft. Diese Art der Verarbeitung ist für den Einsatz in Batch-Jobs ungünstig.

Durch Eingabe des Kommandos *wscrit //H:cscript* wird CScript als Standard Script Host definiert. CScript gibt alle Meldungen in einer DOSBOX aus und wartet nicht auf Bestätigung.

Das oben gezeigte Script prüft, welcher Scripthost aktiv ist und bricht, wenn WSript aktiv ist, ab und startet erneut mit CScript.

Damit das Script schneller arbeiten kann, ist es dennoch zu empfehlen, generell auf CScript umzuschalten.

Weitere Programmbeispiele zur Socket-Programmierung finden Sie auf den Tool-Seiten zum Web-IO. Eine detaillierte Beschreibung zu den Kommandos der Web-IO Digital Modelle finden Sie im Referenzhandbuch.

Programmbeispiel herunterladen

Produkte

Web-IO 4.0 Digital
2xIn, 2xOut

Bei Bedarf auch über PoE zu versorgen

Web-IO 4.0 Digital
12xIn, 12xOut

12x Eingänge,
12x Ausgänge

Weitere Web-IOs

Alle W&T Web-IO Digital 24V

Wiesemann & Theis
GmbH
Porschestr. 12
42279 Wuppertal
Tel.: 0202/2680-110 (Mo-Fr. 8-17
Uhr)
Fax: 0202/2680-265
info@wut.de