

Handbuch Web-Server xx7

Controls Division

0	Inhalt	
0.1	Dokument-Historie	0-4
0.2	Warenzeichen	0-4
1	Einführung	
1.1	Mindestanforderungen	1-1
1.2	Eigenschaften	1-1
1.3	Web-Server ohne Zusatzbaugruppen	1-1
1.4	In der Firmware abgespeicherte HTML Seiten	1-2
2	Struktur und Funktion	
2.1	HTML-Server	2-1
2.2	Daten-Server	2-1
2.3	HTML-Seiten, Bilder, Grafiken etc.	2-2
2.4	Lokales Verzeichnis	2-2
2.5	Web-Builder	2-3
2.6	Web-Connect	2-3
2.6.1	Verbindungs-Möglichkeiten	2-4
2.7	Web-Browser	2-5
3	Installation und Einstellungen	
3.1	Web-Builder	3-1
3.2	Web-Connect	3-2
3.3	Web-Browser- und Netzwerk-Einstellungen	3-3
3.3.1	Proxy Server des PCs	3-3
3.3.2	Modem Verbindung	3-3
3.3.3	Cache Memory des PCs	3-4
3.4	Einstellungen und Tests zu TCP/IP	3-5
3.4.1	TCP/IP Einstellungen	3-5
3.4.2	IP-Adresse für den PC	3-6
3.4.3	PC Name	3-7
3.4.4	Test der IP Adresse und des TCP/IP Protokolls	3-7
3.4.5	IP Adresse finden	3-8
4	Konfiguration und Verwaltung	
4.1	Web-Builder	4-1
4.1.1	Aufbau des Web-Builders	4-1
4.1.2	Verwenden des Web-Builders	4-3
4.1.3	Einfügen der HTML-Seiten in das Anwenderprogramm	4-4
4.2	Web-Connect	4-6
4.3	Variablenseiten	4-7
4.3.1	Vor der Verwendung der Variablenliste	4-7
4.4	Beschreibung der PCD Daten-Punkte	4-8
4.4.1	Ändern von Einträgen	4-9
4.5	Zugriff auf PCD Daten von HTML Seiten	4-9
4.6	Konfiguration des Web-Servers mit dem Saia xx7 I/O-Builder	4-11
4.6.1	Definition der seriellen Schnittstelle	4-12
4.6.2	Definition Modem	4-12

4.6.3	Definition des Web-Servers	4-13
4.7	Konfiguration und Definition des Web-Servers im Anwenderprogramm	4-14
4.7.1	Definition der seriellen Schnittstelle	4-14
4.7.2	Definition Modem	4-14
4.7.3	Definition des Web-Servers	4-17
5.	Beispiel Web-Projekt	
5.1	Seitenstruktur	5-1
5.2	Aufbau der Seiten	5-3
5.2.1	PCD Datenpunkte in den HTML Seiten	5-4
5.2.2	Seite: input.htm	5-4
5.2.3	Seite: output.htm	5-5
5.3	Seiten verlinken	5-10
5.4	Generierung der DBs	5-12
5.5	Einfügen der HTML-Seiten in das Anwenderprogramm	5-13
5.6	Verwenden der Web-Connect SW	5-15
6	Problembhebung	
6.1	Variablen Liste (varlist).....	6-1
A	Anhang A	
A.1	HTML Seiten ohne Editor erstellen	A-1
A.2	CGI (Common Gateway Interface) des Web-Servers	A-1
A.2.1	Readval.exe	A-1
A.2.2	Writeval.exe	A-2
A.2.3	Ordervalues.exe	A-2
A.2.4	Readfile.exe	A-3
A.3	HTML Code	A-4
A.3.1	Werte in die PCD schreiben	A-4
A.3.2	Passwort eingeben.....	A-5
A.4	Dynamische Daten in den Html Seiten auffrischen.....	A-5
A.4.1	Konzept.....	A-6
A.4.2	Seite Starting.....	A-7
A.4.3	Seite Update	A-7
A.4.4	Seite Force.....	A-8
A.4.5	Seite Display	A-9
A.5	Allgemeine Grundlagen zur Web-Server Technologie	A-11
A.5.1	Allgemeine Hinweise zu Web-Servern	A-11
A.5.2	TCP/IP-Protokoll	A-11
A.5.3	IP Adressierung.....	A-12
A.5.4	Client-Server-Technologie.....	A-13
A.5.5	DNS - Domain Name Service	A-14
A.5.6	Routing und gateways.....	A-15
A.6	Allgemeine Grundlagen HTML.....	A-15
A.6.1	“Markup” Elemente.....	A-15
A.6.2	Verknüpfen mit Hyperlinks	A-16
A.6.3	Software-unabhängiger Klartext.....	A-16
A.6.4	Universelle Anwendung.....	A-17

A.6.5	Formatierung für HTML-Elemente	A-17
A.6.6	Eine Programmiersprache für WWW-Seiten.....	A-18
A.6.7	JavaScript und HTML.....	A-18
A.6.8	JavaScript, JScript, ECMA-262, Sprachversionen.....	A-19
A.6.9	HTML-Editoren.....	A-19
B	Anhang B	
B.1	Icons	B-1
B.2	Adresse.....	B-1



0.1 Dokument-Historie

Dokumentnr.	Ausgabe	Änderung	Publikation	Bemerkungen
26/775	D3	31.12.2003	29.02.2004	General revision
	D4	2008-06-26	2008-06-26	Kapitel 3.2 und 4.2 Hinweis auf 26/800 Kapitel 5.6 und 6.1 gelöscht
	DE05	2014-03-26		Logoänderung

0.2 Warenzeichen

- Saia PG5® und Saia PCD® sind eingetragene Warenzeichen der Saia-Burgess Controls AG.
- STEP®, SIMATIC®, S7-300®, S7-400® und Siemens® sind eingetragene Warenzeichen der Siemens AG.
- Windows 95/98/2000, Windows NT, Word, Excel, PowerPoint, FrontPage und Microsoft Internet Explorer sind eingetragene Warenzeichen der Microsoft Corporation.
- Netscape Navigator ist ein eingetragenes Warenzeichen der Netscape Communications Corporation.

Technische Anpassungen und Änderungen unterliegen dem Stand der Technik
Saia-Burgess Controls AG, 2003. © Alle Rechte vorbehalten.

Herausgegeben in der Schweiz

1 Einführung

1.1 Mindestanforderungen

1

Um die Web-Server Funktionalität voll nutzen zu können, müssen folgende Mindestanforderungen eingehalten werden:

PCD System	PCD2.M157, M177, M257, M487
PCD FW	V 2.200
PCD serieller Port	Jeder
PC	<ul style="list-style-type: none"> • IBM PC oder kompatibel • Minimum 133 MHz Pentium mit 32MB RAM • Microsoft Windows 95/98/2000/XP oder Windows NT 4.0. • Winsock 1.1 API compliant networking package • Telephony Application Programming Interface (TAPI) • Microsoft Internet Explorer 5.5 oder Netscape Navigator 6

1.2 Eigenschaften

Der in der Saia PCD integrierte Web-Server erlaubt dem Anwender, Funktionen fürs Bedienen & Beobachten (B&B) mit Hilfe von HTML-Seiten rasch auf einfache und effiziente Weise zu realisieren.

Zur Realisierung der B&B Funktionen werden Standard-Microsoft Programme wie z.B. Word, Excel, PowerPoint oder FrontPage eingesetzt.

Traditionelle Web-Server benötigen ein hohes Mass an Ressourcen, sei es Speicherkapazität oder Rechenleistung. Beides steht bei Steuerungsprozessoren nicht unbegrenzt zur Verfügung bzw. wird zur Maschinen- oder Prozesssteuerung genutzt. In der Steuerungstechnik kommen daher heute Web-Server nur in Form von zusätzlichen PCs oder teuren Zusatz-Modulen (Co-Prozessoren) zum Einsatz.

1.3 Web-Server ohne Zusatzbaugruppen

Ist ein Web-Server ohne Zusatzbaugruppen möglich? Ja, um die genaue Funktion solcher Zusatzbaugruppen zu bestimmen, muss der Aufbau eines Web-Servers näher betrachtet werden:

Die Aufgabe eines Web-Servers ist die Bereitstellung bzw. Verwaltung von Dateien (HTML-Seiten, Bild-Dateien, Java-Applets etc.), sowie der Versand dieser Dateien auf Anforderung. Während die Datei-Verwaltung sich einfach gestaltet, sieht es beim Versenden der Dateien anders aus – muss doch der gesamte Datenverkehr mit Standard-Browsern via TCP/IP-Protokoll erfolgen. TCP/IP stellt jedoch hohe Ansprüche an Speicherressourcen und CPU-Leistung.

Die Funktionen des Web-Servers der PCD xx7 Serie von SBC wurden daher aufgeteilt. Die Dateiverwaltung, befindet sich in der CPU der PCD. Der ressourcenintensive TCP/IP-Kommunikationsteil wird auf dem PC installiert, mit dem die B&B-Funktion aufgerufen wird. Die Kommunikation zwischen PCs und den PCDs kann so über ein einfaches und effizientes Protokoll erfolgen, welches wesentlich ressourcenschonender ist als TCP/IP. Das auf die wesentlichen Transferfunktionen reduzierte Protokoll gewährleistet ausserdem ein deterministisches Verhalten und beeinträchtigt so das Echtzeitverhalten der PCD nicht.

Standard-Browser wie Microsoft Internet Explorer oder Netscape Communicator fordern Dateien von Web-Servern unter Verwendung des TCP/IP-Protokolls an. Um mit dem Web-Server in der PCD kommunizieren zu können, muss auf dem PC das Treiberprogramm Web-Connect installiert sein, was in seiner Funktion dem Kommunikationsteil des Web-Servers entspricht. Web-Connect nimmt die Anforderungen vom Browser im TCP/IP-Protokoll entgegen und leitet diese mit einem effizienten Protokoll an die PCD weiter, je nachdem wie die Kommunikation konfiguriert ist. Dies kann direkt seriell sein oder über ein Modem. Daten von der PCD werden in TCP/IP konvertiert und an den Browser übermittelt.

Durch das Aufsplitten der Web-Server Aufgaben entfällt ein TCP/IP-Co-Prozessor oder TCP/IP-Zusatz-Modul für die PCDs. Es ist nun möglich den PCD Web-Server mit einer einfachen seriellen Punkt-zu-Punkt-Verbindung oder über ein Modem zu verbinden und zu nutzen. Ausserdem können die in einem lokalen Netzwerk (LAN) verbunden PC's über die Web-Connect SW auf den Web-Server der PCD zugreifen.

Vorteile des SBC Web-Servers:

- Preisgünstige Lösung für B&B Funktionen in der Automation.
- Für die Nutzung des Web-Servers ist keine Lizenz notwendig.
- Einfache Erstellung von B&B-Oberflächen mit bekannten Microsoft-Werkzeugen wie Word, Excel, PowerPoint, FrontPage etc.
- Bedienung der B&B-Oberfläche mit Standard-Browsern wie Microsoft Internet Explorer oder Netscape Communicator. Browser haben sich zu einem Standard in punkto Bedienung entwickelt. Die Schaltflächen «Vorwärts», «Rückwärts», «Startseite» oder «Aktualisieren» sind dem Anwender bekannt. Auch ungeschultes Personal kann sofort eine B&B-Oberfläche mit einem Browser bedienen. Die Navigation durch einfaches Anklicken von «Links» ist intuitiv und bereitet keine Schwierigkeiten.
- Keine aufwendige Infrastruktur wie Ethernet-Netzwerke, Internet-Provider, Interface-Baugruppen etc. notwendig, weder auf der PCD noch auf dem PC.
- Mit einem integrierten Modem ideal für Fernwartung von Maschinen und Prozessen über weite Distanzen.
- Mit den 4 integrierten Variablen-Seiten kann der Zustand jedes Prozessdatenpunktes beobachtet und bei Bedarf angepasst werden.
- Der Zugriff auf die B&B Funktionen und Prozessdatenpunkte kann durch 4 frei definierbare Passwörter geschützt werden.

1.4 In der Firmware abgespeicherte HTML Seiten

In der Firmware (FW) sind vom Werk aus 4 HTML-Seiten abgespeichert, welche mit jedem Standardbrowser angezeigt werden können. Dabei muss in Anwenderprogramm nichts spezielles programmiert werden.

Die 4 in der FW abgespeicherten Seiten:

start.htm	Standard Einstiegseite: Wenn im Web-Browser nur die IP-Adresse des PC's eingegeben wird, so erscheint als erstes immer die Seite start.htm. Diese Seite muss zwingend im Web-Projekt enthalten sein. Von dieser Seite aus, werden dann die Verbindungen zu den anderen HTML-Seiten gemacht.
pwdform.htm	Standard Passwort-Seite: In dieser Seite können die Passwörter eingegeben werden. Der Anwender kann eine eigene pwdform.htm-Seite definieren.
status.htm	Auf dieser Seite ist der Typ der PCD zu sehen, ihr Status und die Betriebssystem-Version (Firmware).
varlist.htm	Standard Variablen-Liste: Unter diesem Menüpunkt sind 4 Variablen-Seiten gespeichert, die der Anwender frei editieren kann. Diese Seite ermöglicht den Zugang zu allen PCD-Daten-Punkten (wenn durch den entsprechenden Passwort-Level erlaubt). Es können auch eigene varlist.htm-Seiten definiert werden.

1

Regeln bezüglich der 4, in der FW enthaltenen HTML-Seiten:

- Wenn im Anwenderprogramm HTML-Seiten mit den gleichen Namen (start.htm, user.htm oder pwdform.htm) abgespeichert sind, so werden beim Aufruf dieser Seiten durch den Web-Browser, die im Anwenderprogramm gespeicherten Seiten in den Web-Browser geladen.
- Falls die oben erwähnten HTML-Seiten im Web-Browser aufgerufen werden, ohne dass die HTML-Seiten im Anwenderprogramm vorhanden sind, so werden die Standard-HTML-Seiten aus der FW in den Web-Browser geladen.

Seite: **start.htm**

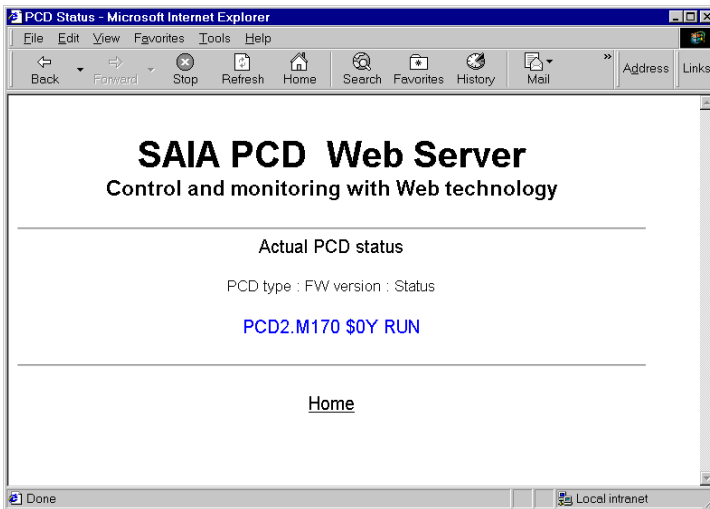


Seite: **pwdform.htm**

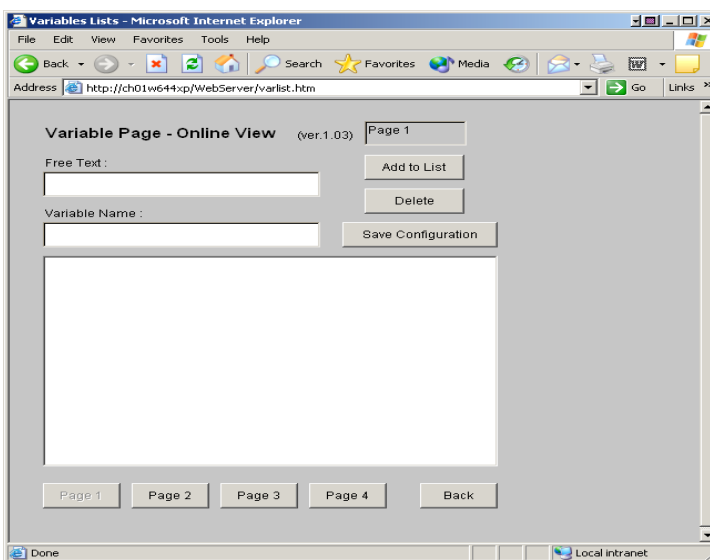


1

Seite: **status.htm**

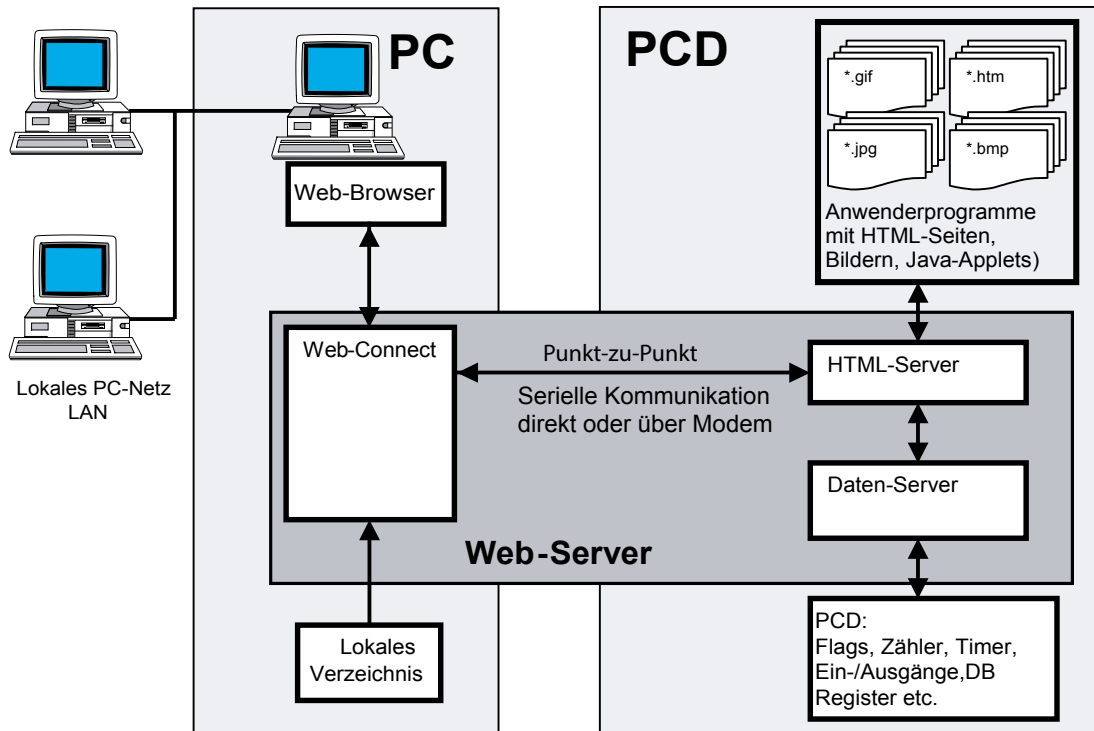


Seite: **varlist.htm**



2 Struktur und Funktion

Der in der PCD integrierte Web-Server besteht aus folgenden Einzelementen:



2

2.1 HTML-Server

Der HTML-Server ist der Kern des gesamten Web-Servers. Dieser sendet die vom Web-Browser angeforderten HTML-Seiten und die zugehörigen Bild-Dateien über die serielle Schnittstelle an den PC. Die HTML-Seiten, Bilder oder Dateien sind in DBs (Daten-Blocks) des Anwenderprogrammspeichers der PCD abgespeichert, da dies zum Anwenderprogramm gehört, wird es vom PG5 heruntergeladen. Im HTML-Server werden die HTML-Seiten, die zur Anzeige von PCD Ressourcen verwendet werden, zusätzlich noch auf die Prozessdatenpunkt-Kennung (PDP-Kennung) geprüft. Sobald der HTML-Server in der HTML-Seite eine PDP-Kennung gefunden hat, sendet er diese an den Daten-Server weiter.



Der HTML-Server prüft nur Dateien mit der Endung *.HTML und *.HTM auf eine mögliche PDP-Kennung und transferiert diese dann an den Daten-Server. Das heisst, dass alle Dateien welche eine PDP-Kennung enthalten, zwingend die Endung *.HTML oder *.HTM besitzen müssen.

2.2 Daten-Server

Der Daten-Server bearbeitet die vom HTML-Server erhaltenen PDP-Kennungen und transferiert die in der PDP-Kennung angeforderten PCD Daten direkt aus dem PCD-Speicher an den HTML-Server. Der Daten-Server kann auf DBs, Flags, Ein-/Ausgänge, Register, Timer und Zähler, sowie den Status der PCD zugreifen.

2.3 HTML-Seiten, Bilder, Grafiken etc.

Ein wesentlicher Vorteil des Web-Servers ist die Erstellung der B&B-Oberfläche mit Standard-Werkzeugen von Microsoft. Jedes HTML generierende Software-Werkzeug kann verwendet werden, um HTML-Seiten zu erstellen. Die einzelnen Seiten des B&B-Projekts können beispielsweise komfortabel mit Word in Form einer Text-Datei erstellt werden.

Es können alle von Web-Browsern darstellbare Tabellen, Grafiken, Diagramme, Fotos usw. eingefügt werden. Hierbei kann die gesamte Funktionalität der jeweiligen Microsoft-Anwendung genutzt werden. Ohne besondere Vorkenntnisse lassen sich beliebig viele Seiten zu komplexen Menüstrukturen verbinden.

Um aktuelle PCD-Daten in die B&B-Funktionen zu integrieren, werden einfache Textkommandos in Form von PDP-Kennungen in die Text-Datei eingefügt. Dadurch können PCD Media Daten während des Programmablaufs in verschiedenen Formaten angezeigt werden.

In der PCD werden alle HTML-Seiten, Bilder oder Dateien der B&B-Applikation in Datenbausteinen (DB's) des Anwenderprogrammes abgespeichert. Der Anwender kann jede beliebige Datei in jedem beliebigen Format und jeder beliebigen Endung (*.htm, *.html, *.bmp, *.giv, *.txt, *.zip oder andere) in der SPS abspeichern, solange die Datei maximal **32kByte** gross ist. Jede HTML-Seite, jedes Bild und jede Datei der B&B-Applikation wird in einem individuellen DB abgespeichert. Die B&B-Applikation kann bis zu **1022** einzelne HTML-Seiten, Bilder oder Dateien enthalten und kann den gesamten Anwenderprogrammspeicher belegen.



- Es gilt zu beachten, dass die Informationen der B&B-Applikation in den DB's des Anwenderprogrammspeichers abgespeichert werden und das Anwenderprogramm und die B&B-Applikation die selben DB's belegen. Es muss sichergestellt werden, dass im Anwenderprogramm nicht auf die DB's der B&B-Applikation zugegriffen wird.
- Der HTML-Server überprüft nur Dateien mit der Endung *.HTML oder *.HTM auf die PDP-Kennung.



Falls sehr grosse HTML-Seiten, Bilder oder Dateien in der B&B-Applikation dargestellt werden sollen, welche nicht im Anwenderprogrammspeicher der PCD Platz haben, können diese HTML-Seiten, Bilder oder Dateien auf der Harddisk des PCs abgespeichert werden, auf welchem die Web-Connect SW läuft. Dabei müssen aber die HTML-Seiten, Bilder oder Dateien im Web-Connect Verzeichnis abgespeichert werden.

2.4 Lokales Verzeichnis

Das Lokale Verzeichnis ist ein Verzeichnis, das sich auf dem PC befindet, auf dem die Web-Connect SW läuft. In der Voreinstellung ist der Name und Ort des Verzeichnisses «C:\WebPages», dies kann in den Web-Connect Optionen geändert werden, oder während der Software Installation.

Das Verzeichnis kann Dateien enthalten, welche nun «the local files» genannt werden.

Es besteht die Möglichkeit Dateien, welche nicht in der PCD gespeichert werden sol-

len, in diesem Verzeichnis abzulegen. Wenn die Dateien gross sind, könnte es relativ lange dauern, diese von der PCD zu laden. Um dies zu verhindern, kann man diese Dateien in dem lokalen Verzeichnis speichern. Der Vorgang ist dann folgender: die Datei wird von der PCD verlangt und wenn sie dort nicht existiert, wird die Web-Connect SW das lokale Verzeichnis überprüfen und sie zurücksenden. Dieser Vorgang ist transparent für den Anwender.

Ebenso speichern Sie hier die lokale Web-Seite, wie die default.html-Seite und alle dazugehörigen Dateien, dieses wird näher im Kapitel «Web-Connect» erläutert.

2.5 Web-Builder

Die PCD xx7 Steuerungen werden mit der STEP®7-Programmiersoftware von Siemens® programmiert. Die B&B-Dateien werden im Anwenderprogrammspeicher der PCD abgelegt. Die STEP®7-kompatiblen PCD-Steuerungen (STEP®7 Siemens®) sind mit bis zu 1024kByte Arbeitsspeicher ausgestattet. Damit lassen sich auch umfangreiche B&B-Funktionen in der PCD abspeichern.

Sämtliche Dateien, welche zur B&B-Funktion gehören (HTML-Seiten, Bilder, etc.), werden in Datenbausteinen gespeichert. Zuerst müssen die Web-Dateien in STEP®7-Quelldateien konvertiert werden. Die Konvertierung erfolgt durch das einfach zu handhabende Software-Werkzeug Web-Builder. Damit lassen sich die zu konvertierenden HTML-Seiten und Bilder auswählen sowie der Bereich der Datenbausteine festlegen, in welchem die HTML-Seiten abgespeichert werden sollen. Die STEP®7-Quelldatei muss dann nur noch in die STEP®7-Programmiersoftware von Siemens® importiert werden. Dies erfolgt wie gewohnt mit der Funktion «Einfügen, Externe Quelle». Danach wird mit der Funktion «Bearbeiten, Übersetzen» die Quelldatei übersetzt und die Datenbausteine zum STEP®7-Projekt hinzugefügt. Danach können diese in die PCD geladen werden.

Die Web-Builder SW ist in zwei Versionen verfügbar:

Demo-Version:

- Benötigt kein Passwort bei der Installation.
- Maximal 2 eigene HTML-Seiten, Bilder oder Dateien können in DBs konvertiert werden.
- Option VarList Sites kann nicht ausgewählt werden.

Voll-Version:

- Bei der Installation muss ein Passwort eingegeben werden.
- Keine Beschränkung in der Anzahl der zu konvertierenden HTML-Seiten, Bilder oder Dateien.

2.6 Web-Connect

Um die in der Steuerung abgespeicherten B&B Funktionen auf dem Web-Browser anzuzeigen, muss das PC-Treiberprogramm Web-Connect gestartet werden.

Diese SW empfängt die Anfragen welche vom Browser kommen und leitet diese zu der richtigen PCD Station weiter wie auch umgekehrt.

Diese SW erstellt einen Link zwischen dem PC-Browser und der Kommunikation der PCD.

Das Web-Connect Programm ermöglicht dem Benutzer:

- das Konfigurieren der Station, auf die von diesem PC zugegriffen werden soll
- das Konfigurieren der Kanäle, über die auf die Station zugegriffen werden soll
- das Weiterleiten von Anfragen vom PC zu den PCDs
- das Weiterleiten von Anfragen anderer PCs auf demselben LAN zu den PCDs

Es gibt mehrere Möglichkeiten auf eine PCD zuzugreifen:

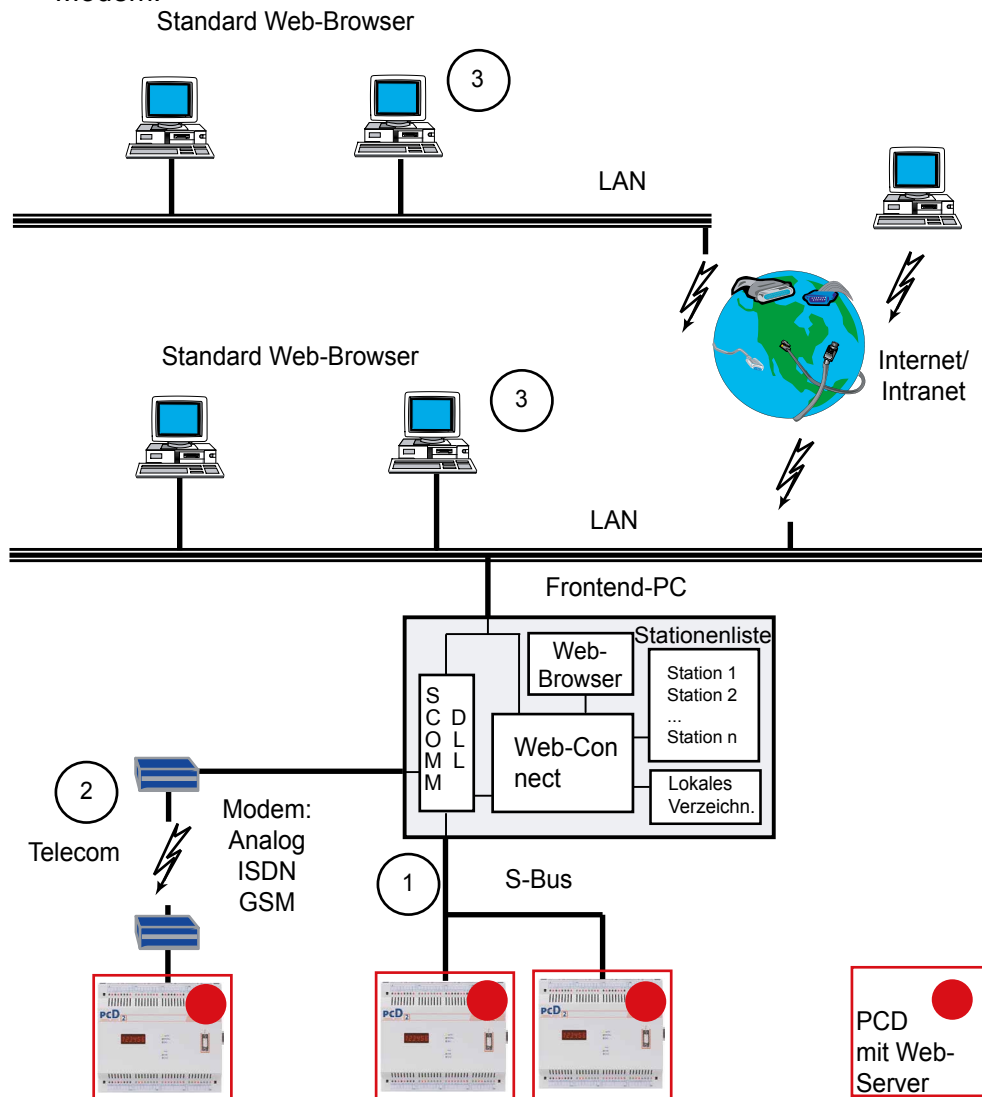
- der Direkte Zugriff und
- der Indirekte Zugriff.

2.6.1 Verbindungsmöglichkeiten

Direkter Zugriff:

Die Anfrage des Web-Browsers und die Web-Connect SW laufen beide auf dem selben PC. Je nach dem welche PCD die Anfrage betrifft, gibt es verschiedene Verbindungsarten, welche hier vorgestellt werden:

- ① Direkte Verbindung mit der seriellen Schnittstelle des PCs.
Die PCD ist direkt mit dem PC über eine serielle Schnittstelle oder über ein Gateway verbunden.
- ② Direkte Telecom-Netz Verbindung über Modems
Die PCD ist mit einem Modem ausgestattet und kommuniziert mit dem PC über Modem.



Indirekter Zugriff:

- ③ Indirekt über ein LAN Netzwerk:
Browser-PCs, die keine Web-Connect SW enthalten, können ebenfalls Anfragen an eine PCD mit Web-Server schicken. Dazu müssen sie einen indirekten Zugriff vornehmen. Dabei schickt der PC zuerst die Anfrage an den Frontend-PC, auf dem die Web-Connect SW läuft und dieser schickt die Anfrage weiter zur PCD, auf dem selben Weg erfolgt die Antwort.

2

Beide Vorgehensweisen sind möglich, da die Web-Connect SW nicht unterscheidet, ob die Anfrage von einem Lokalen Web-Browser kommt, oder von einem Web-Browser, der auf einem PC läuft, der an ein LAN angeschlossen ist.

2.7 Web-Browser

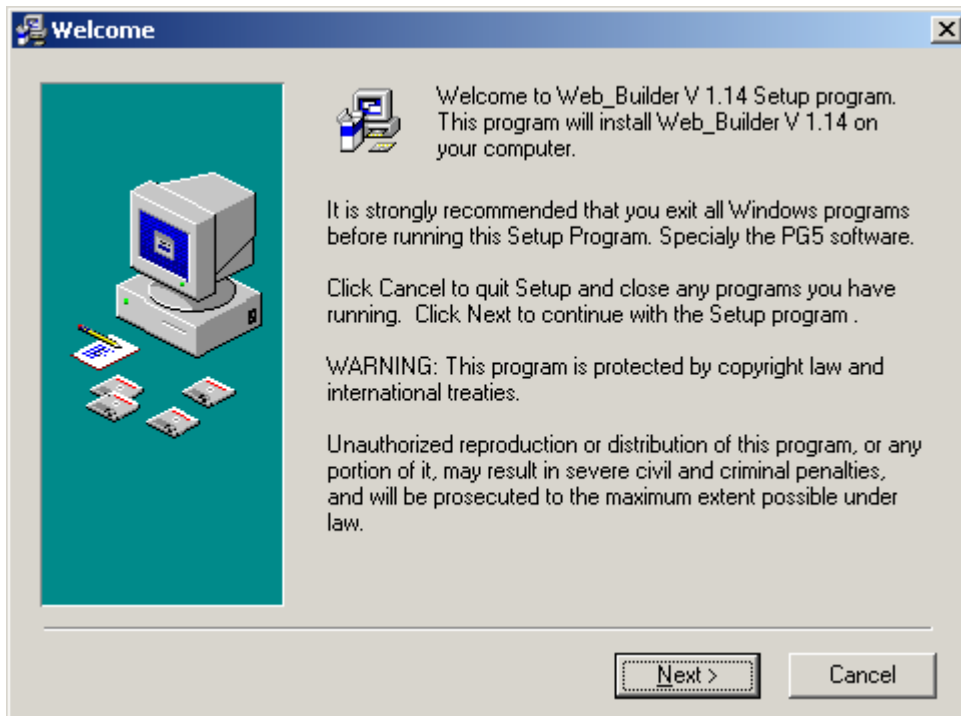
Wir empfehlen die Web-Browser Netscape Navigator oder Microsoft Internet Explorer um die HTML-Seiten, Bilder oder Dateien anzuzeigen, die in der PCD abgelegt sind und zur B&B-Anwendung gehören.

3 Installation und Einstellungen

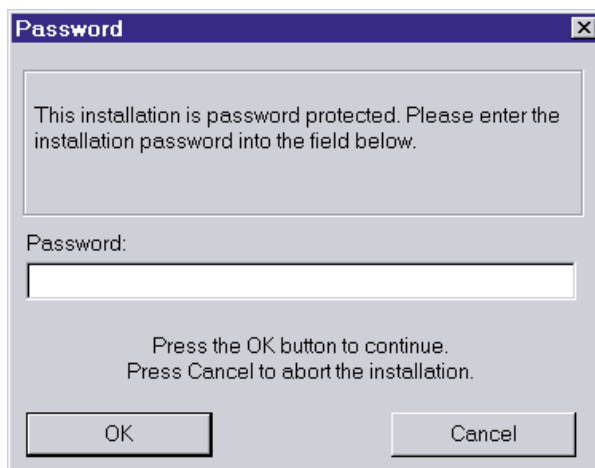
3.1 Web-Builder

Die Web-Builder SW wird mittels eines Setup Programms installiert: Wählen Sie unter Windows das Menü «Start», «Ausführen» und wählen Sie den Pfad für die Web-Builder SW. In diesem Pfad muss die Datei «Setup.exe» ausgewählt und gestartet werden. Danach wird die Web-Builder SW auf dem PC installiert.

3



In einem entsprechenden Menü muss das kundenspezifische Passwort eingegeben werden. (Nur bei der Voll-Version)



Nach der Eingabe des korrekten Passwortes wird die Web-Builder SW defaultmässig im Pfad D:\Programme\SBC\xx7\Web-Builder installiert und im Windows Programm Menü wurde folgender Programmaufruf für den Web-Builder unter Programme, SBC, xx7 installiert:



3.2 **Web-Connect**

Für Installation und Einstellung bitte Handbuch 26-800 Web-Connect verwenden.
Das Dokument befindet sich unter www.sbc-support.com unter Software.

3.3 Web-Browser- und Netzwerk-Einstellungen

Falls noch kein Web-Browser auf Ihrem PC installiert ist, muss dieser gemäss den Installationsangaben des jeweiligen Web-Browser installiert werden. Microsoft Virtual Machine muss installiert sein. Diese Installation erfolgt mit der Installation des Web-Browsers.

Die erforderliche Version von MS Internet Explorer oder Netscape Navigator entnehmen Sie **1.1 Mindestanforderungen**.

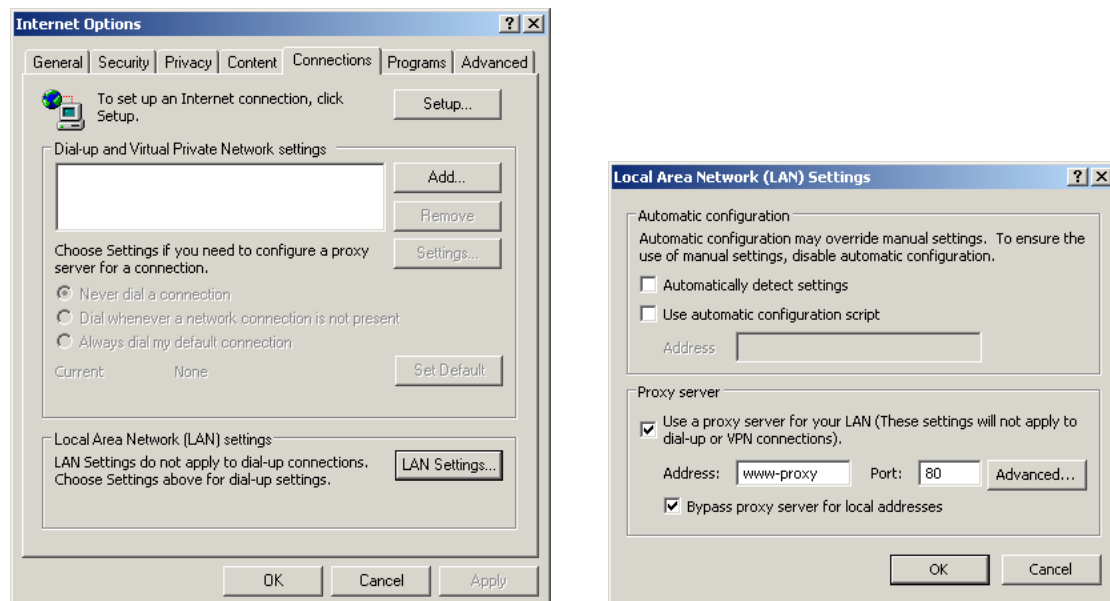
Web-Seiten von denen Web-Browser geladen werden können:

MS Internet Explorer in Deutsch:	http://www.eu.microsoft.com/germany/
MS Internet Explorer in Englisch:	http://www.microsoft.com/
Netscape Navigator in Deutsch:	http://home.de.netscape.com/de/
Netscape Navigator in Englisch:	http://home.netscape.com/

3.3.1 Proxy Server des PCs

Falls der PC über einem Proxy Server mit dem Internet verbunden ist, muss dieser Proxy Server für lokale Adressen umgangen werden.

Dies kann z.B. im Internet Explorer mit folgender Option eingestellt werden:



«Tools»→«Internet Options»..., Auswahl Tab «Connections» und drücken der Taste «LAN Settings» und anhängen der Option «Bypass proxy server for local addresses».

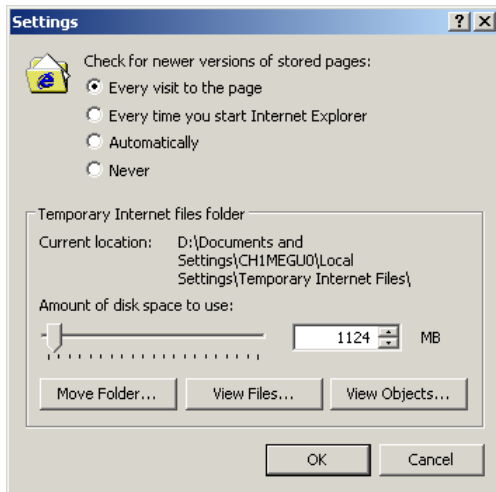
3.3.2 Modem Verbindung

Sollte der PC über eine Modemverbindung mit dem Internet kommunizieren, muss die Option «Never dial a connection» ausgewählt werden. Diese Option erscheint beim Internet Explorer unter «Tools» -> «Internet Options»..., Tab «Connection»

Dieser Modus ist zu wählen, um zu verhindern, dass sich der PC nicht jedesmal, wenn man auf die lokale Web-Connect SW zugreifen will, ins Internet einloggt.

3.3.3 Cache Memory des PCs

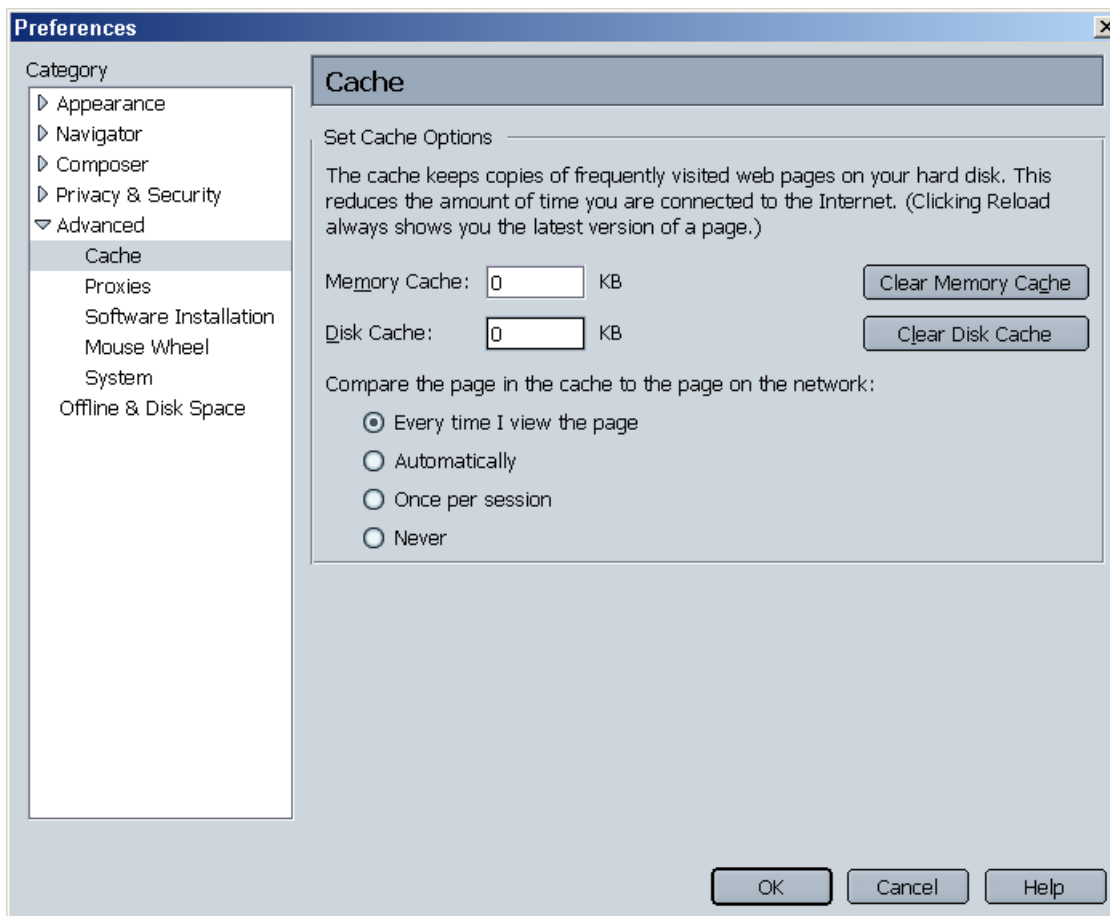
Damit die HTML Seiten periodisch aufgefrischt werden, muss das Cache Memory des Web-Browsers ausgeschaltet sein.



3

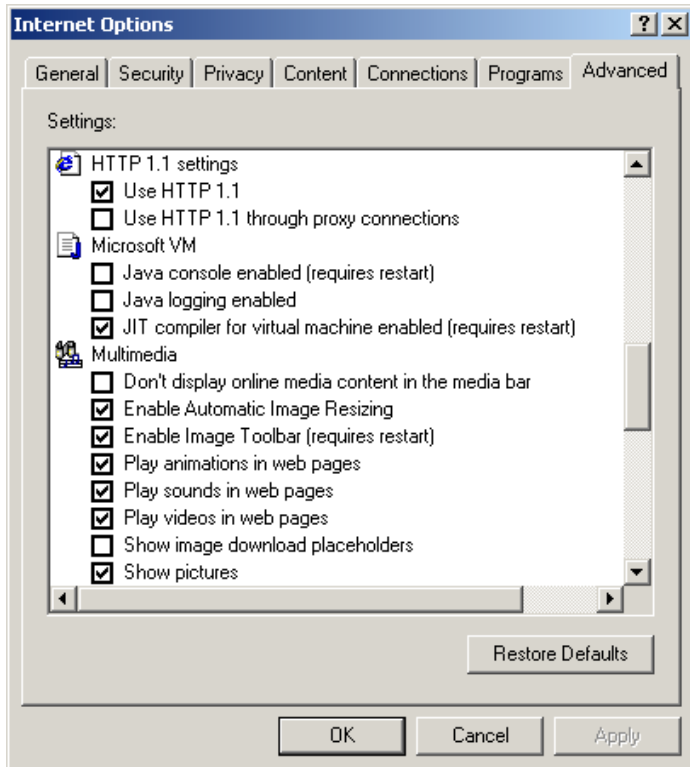
Dies kann man beim Internet Explorer durch «**Tools**»→«**Internet Options**»..., Tab «**General**», «**Settings**» auswählen, unter der Option «**Check for newer versions of stored pages**», «**Every visit to the page**».

Beim Netscape Communicator – Menu: «**Edit**»→«**Preferences**»:





Um die Ausführung von Java Applets unter Internet Explorer zu optimieren, muss die Option Java Konsole deaktiviert sein.



3

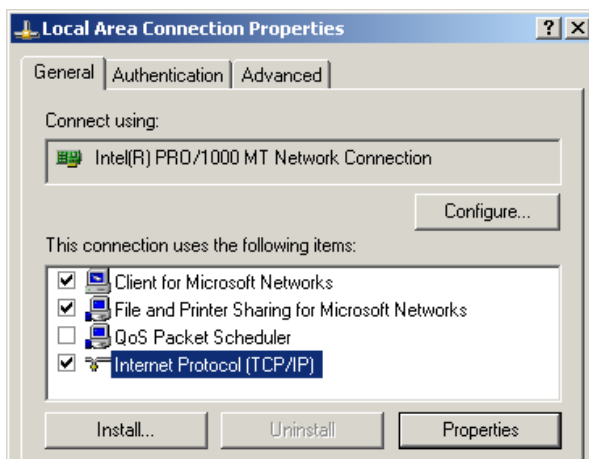
Um die Einstellungen der Java Konsole zu prüfen, muss unter Internet Explorer «Tools», «Internet Options», «Advanced», «Microsoft VM» (Virtual Machine) ausgewählt werden. Dabei darf die Option «Java console enabled» nicht angehakt sein.

3.4 Einstellungen und Tests zu TCP/IP

3.4.1 TCP/IP Einstellungen

Der PC auf welchem der Web-Browser die B&B-Anwendungen der PCD anzeigen soll, muss über ein TCP/IP Protokoll verfügen. Um zu testen, ob auf dem PC das TCP/IP Protokoll installiert ist, kann man über «Start», «Settings», «Control Panel», «Network Connections» z.B. «Local Area Connection» aufrufen.

Das TCP/IP Protokoll ist auf dem PC installiert, wenn unter dem Tab «General» ein TCP/IP Protokoll definiert ist.



3.4.2 IP-Adresse für den PC

Der Zugang zum PCD Web-Server erfolgt über einen Web-Browser.

Im Web Browser muss eine eindeutige Zuordnung zum PC, auf dem die Web-Connect SW läuft, erfolgen. Diese Zuordnung geschieht durch die URL in welcher die IP-Adresse oder der Name des PCs enthalten sein muss.

Wenn Sie nun nicht den PC Namen verwenden wollen, sondern die IP-Adresse bevorzugen, diese jedoch keine feste IP-Adresse ist, erfordert dies noch die Fixierung der IP-Adresse. Um die IP-Adresse zu fixieren müssen Sie wie folgt vorgehen. Zuerst müssen Sie eine gültige und eindeutige IP-Adresse haben. Eine eigene, eindeutige Adresse heisst, dass die Option **«Obtain an IP address automatically»** unter den TCP/IP Einstellungen des PCs **nicht** ausgewählt sein darf. Diese Option ist bei WIN 2000 unter **«Start», «Settings», «Control Panel», «Network Connections»** und z.B. **«Local Area Connection»** und dann **«General»** zu finden.

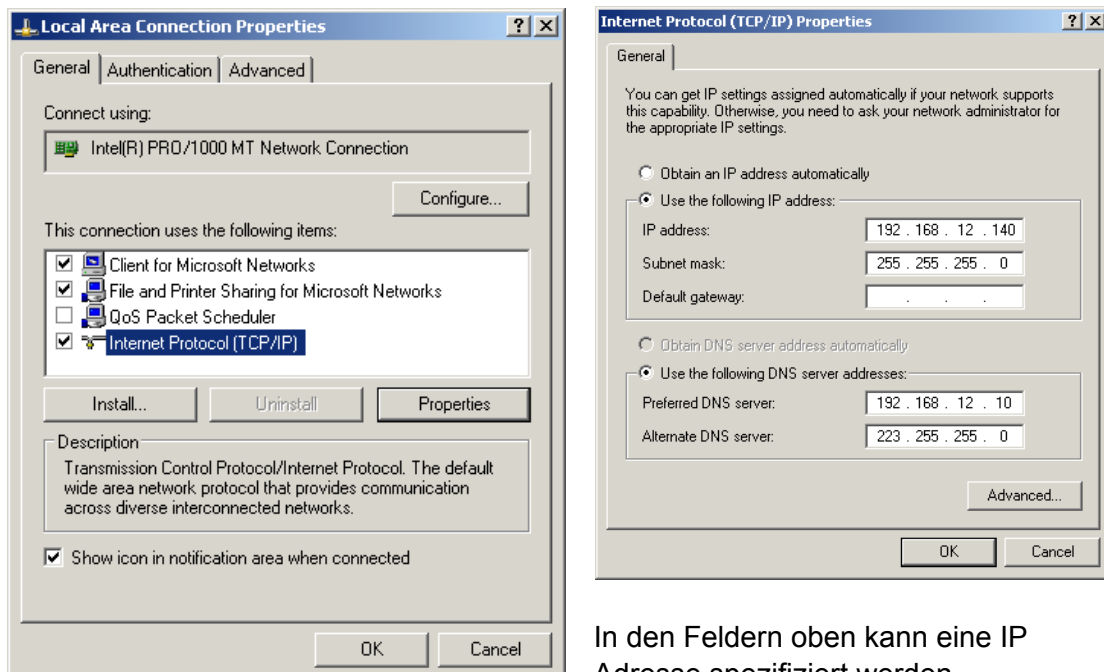
Unter demselben Menüpunkt kann auch unter **«Properties»** die IP-Adresse des PCs definiert werden.



Jeder PC hat von Haus aus die gleiche voreingestellte IP-Adresse. Diese ist: 127.0.0.1

D.h. falls keine IP-Adresse spezifiziert ist, so kann im Web-Browser diese Adresse benutzt werden, um auf die HTML-Seiten der PCD zuzugreifen.

Bei WIN 2000

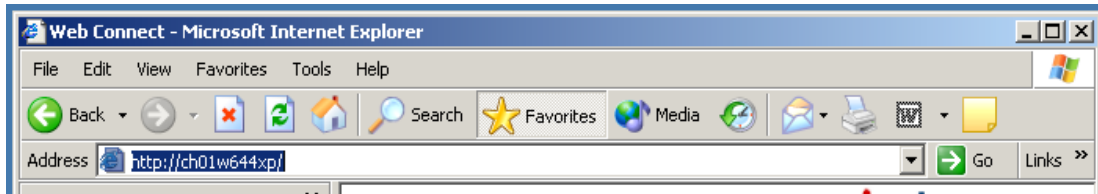


In den Feldern oben kann eine IP Adresse spezifiziert werden.

3.4.3 PC Name

Wie bereits erwähnt ist es auch möglich mit dem PC Namen zu arbeiten, statt der IP Adresse.

In der URL kann man localhost eintragen und es wird dann der PC Name angezeigt. In diesem Fall ist der Name ch01w644xp.

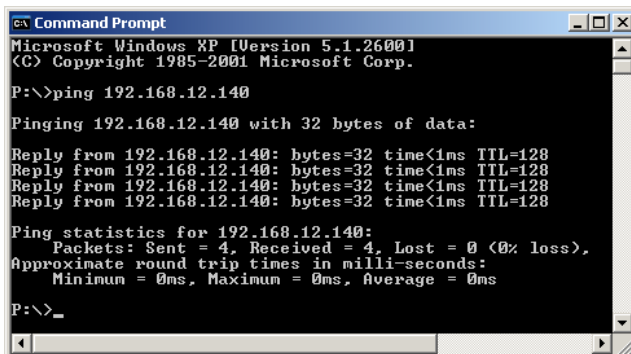


3.4.4 Test der IP Adresse und des TCP/IP Protokolls

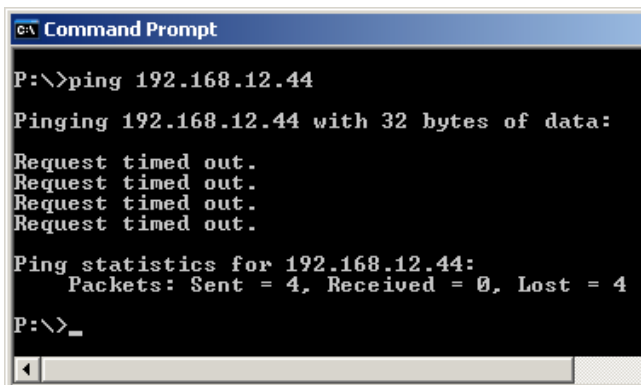
Um zu prüfen, ob auf dem PC ein TCP/IP Protokoll installiert ist, und ob die IP Adresse stimmt, kann folgender Test durchgeführt werden:



Start der DOS Konsole mit «Start», «Program», «MS-DOS Command Prompt».



Eingabe des Befehls Ping «IP Nummer des PCs». z.B. Ping 192.168.12.140. Wenn die IP-Adresse richtig ist, erscheint folgende Antwort.



Existiert die IP-Adresse nicht, erscheint folgende Antwort.

Falls die IP-Adresse des PC nicht bekannt ist, so kann man die Adresse 127.0.0.1 benutzen.

```

c:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

P:\>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
    
```

Mit der Ping Funktion «localhost» ist es ebenfalls möglich den PC_Namen zu überprüfen, den Sie verwenden,:

```

c:\ Command Prompt

P:\>ping localhost

Pinging CH01W644XP [127.0.0.1] with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

P:\>
    
```

«ping localhost»

In diesem Beispiel ist der PC_Name ch01w644xp.

3.4.5 IP Adresse finden

Ein weitere einfache Art zu überprüfen, was auf Ihrem PC konfiguriert ist, ist die Funktion «ipconfig».

```

c:\ Command Prompt

P:\>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . . : 
    IP Address . . . . . : 192.168.12.140
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.12.221

P:\>
    
```

4 Konfiguration und Verwaltung

4.1 Web-Builder

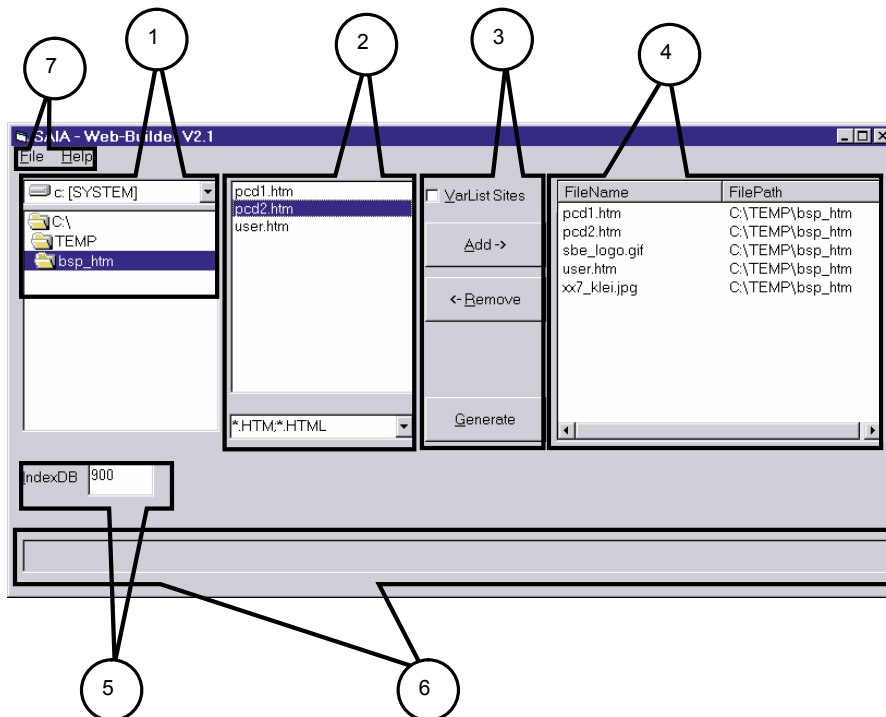
4.1.1 Aufbau des Web-Builders

Der Web-Builder kann aus der Task Liste aufgerufen werden, indem dieses Feld betätigt wird:

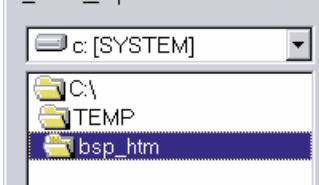
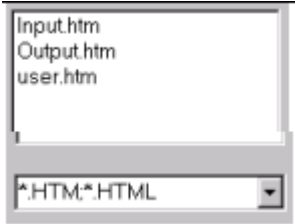


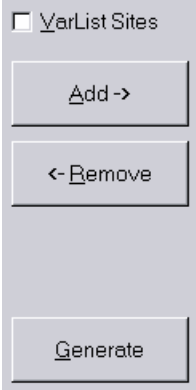

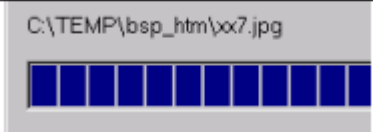
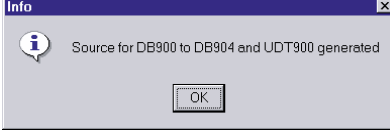
4

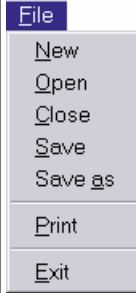
Danach ist folgender Bildschirm sichtbar:



Die einzelnen Komponenten auf dem Bildschirm sind:

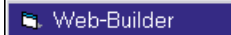
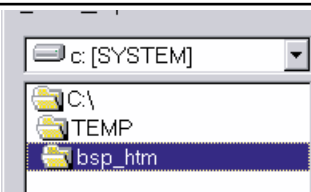

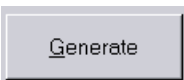
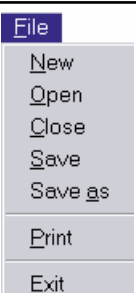
<p>1 Navigator mit welchem das gewünschte Laufwerk und der gewünschte Pfad ausgewählt werden kann in welchem die zu konvertierenden B&B Dateien und Bilder abgespeichert sind.</p>	
<p>2 Auswahlfenster: Nur die Dateien, die sich im Ordner «html» Ihrer CPU befinden, sind hier dargestellt. Folgende Anzeigen können ausgewählt werden: *.HTM, *.HTML: Anzeige HTML Dateien *.gif, *.jpg, *.bmp: Anzeige Grafik/Bild-Dateien *.*: Anzeige aller Dateien</p>	

<p>3</p>	<p>Funktionstasten mit welchen Dateien und Bilder aus dem Auswahlfenster in das Web-Server Inhaltsfenster eingefügt oder entfernt werden können.</p> <ul style="list-style-type: none"> • Add→ Die im Auswahlfenster markierten Dateien werden in das Inhaltsfenster kopiert. • ← Remove Die im Inhaltsfenster markierten Dateien werden gelöscht. • Generate Alle im Übersetzungsfenster eingetragenen Dateien und Bilder werden in DB's konvertiert. Die DB's werden in einer *.AWL Datei abgelegt. Nach Betätigung des Knopfes erscheint ein Fenster in welchem die *.AWL Datei spezifiziert werden kann in welche die DB's abgespeichert werden sollen. • VarListSites Wenn dies Option angewählt ist, werden im Übersetzungsfenster 4 Dateien generiert, welche es ermöglichen die Variablenseiten des Web-Servers abzuspeichern. Dadurch können die on-line editierten Einträge der Variablenseiten auf der SPS abgespeichert werden. Wenn diese Option nicht angewählt ist, so können die Variablenseiten zwar editiert werden, jedoch ist eine Speicherung der Variablenseiten auf der SPS nicht möglich. 											
<p>4</p>	<p>Übersetzungsfenster Alle zu übersetzende Dateien und Bilder müssen in diesem Fenster aufgeführt sein. Jede in diesem Fenster aufgeführte Datei wird in einen DB konvertiert.</p>	<table border="1"> <thead> <tr> <th>FileName</th> <th>FilePath</th> </tr> </thead> <tbody> <tr> <td>Input.htm</td> <td>C:\TEMP\bsj</td> </tr> <tr> <td>Output.htm</td> <td>C:\TEMP\bsj</td> </tr> <tr> <td>user.htm</td> <td>C:\TEMP\bsj</td> </tr> <tr> <td>xx7.jpg</td> <td>C:\TEMP\bsj</td> </tr> </tbody> </table>	FileName	FilePath	Input.htm	C:\TEMP\bsj	Output.htm	C:\TEMP\bsj	user.htm	C:\TEMP\bsj	xx7.jpg	C:\TEMP\bsj
FileName	FilePath											
Input.htm	C:\TEMP\bsj											
Output.htm	C:\TEMP\bsj											
user.htm	C:\TEMP\bsj											
xx7.jpg	C:\TEMP\bsj											
<p>5</p>	<p>IndexDB Definition des ersten DBs in welchen die Dateien und Bilder abgespeichert werden. Die Nummer des IndexDBs muss auch im CDB des Anwenderprogrammes definiert werden. Zusätzlich wird noch ein UDT mit der Index Nummer generiert. Die Anzahl der generierten DBs entspricht der Summe der im Übersetzungsfenster eingetragenen Dateien + 1.</p>											
<p>6</p>	<p>Progress Bar Während der Generierung der DBs erscheint der Name des aktuell übersetzten DBs und der Übersetzungsfortschritt wird grafisch dargestellt.</p>											
<p>7</p>	<p>Nach Beendigung der DB Generierung wird in diesem Fenster angezeigt, welche DBs generiert worden sind. Die Anzahl der generierten DBs entspricht der Summe der im Übersetzungsfenster eingetragenen Dateien + 1.</p>											

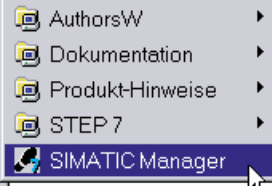

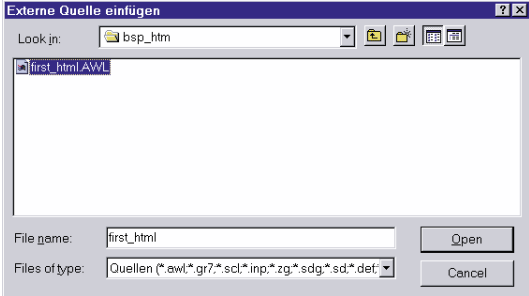
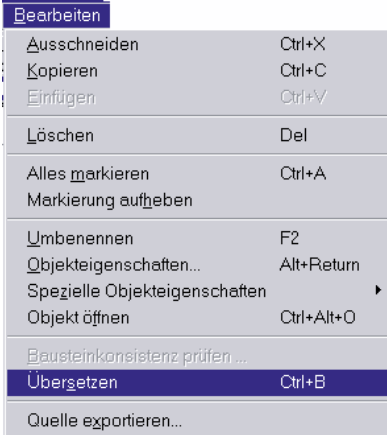
8	<p>Menüstruktur File</p> <ul style="list-style-type: none"> • New Öffnen eine neuen Fensters • Open Öffnen eines bestehenden Projektes welches die im Übersetzungsfenster eingetragenen Dateien und Bilder wie auch die dazugehörigen Pfade wiederherstellt. • Close Schliesst das Fenster • Save/Save as Die im Übersetzungsfenster eingetragenen Dateien und Bilder wie auch die dazugehörigen Pfade können unter diesem Punkt als Projekt gespeichert werden. • Print Druckt Informationen des aktuellen Projekts. • Exit Beendet den Web-Builder 	
---	---	--

4.1.2 Verwenden des Web-Builders

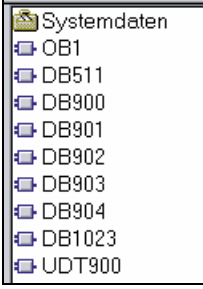
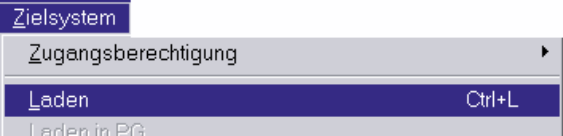
Generell müssen alle Dateien und Bilder für die B&B Funktionen welche in der PCD abgespeichert werden sollen in DBs (Datenbausteine) des Anwenderprogrammes abgespeichert werden. Die Konvertierung und Abspeicherung der Dateien und Bilder erfolgt mit dem Web-Builder. Dabei kann wie folgt vorgegangen werden:

1	Starten des Web-Builders									
2	Auswahl des Pfades in welchem die zuvor mit FrontPage Express entwickelten Dateien und Bilder abgespeichert wurden.									
3	Auswahl der Dateien user.htm, input.htm, output.htm etc.	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">FileName</th> <th style="width: 50%;">FilePath</th> </tr> </thead> <tbody> <tr> <td>Input.htm</td> <td>C:\TEMP\bsp_htm</td> </tr> <tr> <td>Output.htm</td> <td>C:\TEMP\bsp_htm</td> </tr> <tr> <td>user.htm</td> <td>C:\TEMP\bsp_htm</td> </tr> </tbody> </table>	FileName	FilePath	Input.htm	C:\TEMP\bsp_htm	Output.htm	C:\TEMP\bsp_htm	user.htm	C:\TEMP\bsp_htm
FileName	FilePath									
Input.htm	C:\TEMP\bsp_htm									
Output.htm	C:\TEMP\bsp_htm									
user.htm	C:\TEMP\bsp_htm									
4	Eingabe des IndexDB									
5	Konvertierung der Dateien und Bilder									
6	Falls gewünscht können die Informationen des Übersetzungsfenster (Dateien und Pfade) in einer Projektdatei abgespeichert werden. Diese Projektdatei kann zu einem späteren Zeitpunkt wieder geladen werden.									

4.1.3 Einfügen der HTML-Seiten in das Anwenderprogramm

1	Starten des Siemens® Simatic® Managers von Step®7.	
2	Einfügen der externen Quellen im Menü: Einfügen, Externe Quelle	
3	Auswahl der *.awl Datei, welche mit dem Web-Builder generiert wurde	
4	Übersetzen der neu eingeführten Quelldatei mit dem Menü: Bearbeiten, Übersetzen	

4

5	<p>Nach dem Übersetzen wurden die in der Quelldatei abgespeicherten DBs im Bausteinbehälter des Anwenderprogrammes abgelegt.</p> <p>Im Beispiel sind dies: DBs 900 bis 904 und der UDT900</p>																															
6	<p>Definition des Web-Servers im CDB</p> <p>In Beispiel ist dies der CD 1023</p> <p>Darin muss mindestens der IndexDB definiert werden. Wenn keine Defintion der Schnittstelle erfolgt, wird Default-mässig die Schnittstelle 1 mit 19'200 Baud, 8 Datenbits, ungerader Parität und 1 Stop Bit definiert</p>	<table border="1"> <thead> <tr> <th>Adresse</th> <th>Name</th> <th>Typ</th> <th>Anfangswert</th> <th>Kommentar</th> </tr> </thead> <tbody> <tr> <td>0.0</td> <td></td> <td>STRUCT</td> <td></td> <td></td> </tr> <tr> <td>+0.0</td> <td>Identificator</td> <td>STRING[12]</td> <td>'SAIA xx7 CDB'</td> <td>CDB Identification</td> </tr> <tr> <td>+14.0</td> <td>WebPara</td> <td>STRING[30]</td> <td>'COM1:PTP_MPI,RS232,38400,8,o,1'</td> <td>Parameter of serial Port</td> </tr> <tr> <td>+46.0</td> <td>IndexDB</td> <td>STRING[15]</td> <td>'WEB:INDEXDB=900'</td> <td>Index DB -> identical with Web-Bu.</td> </tr> <tr> <td>=64.0</td> <td></td> <td>END_STRUCT</td> <td></td> <td></td> </tr> </tbody> </table>	Adresse	Name	Typ	Anfangswert	Kommentar	0.0		STRUCT			+0.0	Identificator	STRING[12]	'SAIA xx7 CDB'	CDB Identification	+14.0	WebPara	STRING[30]	'COM1:PTP_MPI,RS232,38400,8,o,1'	Parameter of serial Port	+46.0	IndexDB	STRING[15]	'WEB:INDEXDB=900'	Index DB -> identical with Web-Bu.	=64.0		END_STRUCT		
Adresse	Name	Typ	Anfangswert	Kommentar																												
0.0		STRUCT																														
+0.0	Identificator	STRING[12]	'SAIA xx7 CDB'	CDB Identification																												
+14.0	WebPara	STRING[30]	'COM1:PTP_MPI,RS232,38400,8,o,1'	Parameter of serial Port																												
+46.0	IndexDB	STRING[15]	'WEB:INDEXDB=900'	Index DB -> identical with Web-Bu.																												
=64.0		END_STRUCT																														
7	<p>Laden des Anwenderprogrammes.</p> <p>Hinweis: Da im Beispiel auf Ein- und Ausgänge zugegriffen wird, müssen diese im Projekt konfiguriert sein.</p> <p>Falls der CDB das erste Mal geladen wurde, oder falls die Einträge im CDB geändert wurden, muss zur Aktivierung der CDB-Einträge die Speisepannung der Steuerung aus- und eingeschaltet werden</p>																															

4.2 **Web-Connect**

Für Details bitte Handbuch 26-800 Web-Connect verwenden.
Das Dokument befindet sich unter www.sbc-support.com unter Software.

4.3 Variablenseiten

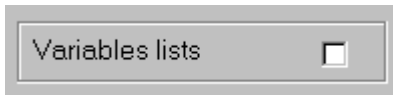
Im Web-Server sind 4 Variablenseiten untergebracht mit denen auf jeden PCD Datenpunkt lesend oder schreiben zugegriffen werden kann. Die Variablenseiten sind für Service und Debuggzwecke geeignet und entsprechen den VAT (Variablentabellen) unter STEP[®]7 von Siemens[®].

Jede Variablenseite kann bis zu 16 frei definierbare Prozessdatenpunkte enthalten. Jedem Prozessdatenpunkt kann auch Text zugeordnet werden. Lesen und Schreiben der Prozessdatenpunkte sowie das Speichern von Variablenseiten ist Passwort-geschützt.

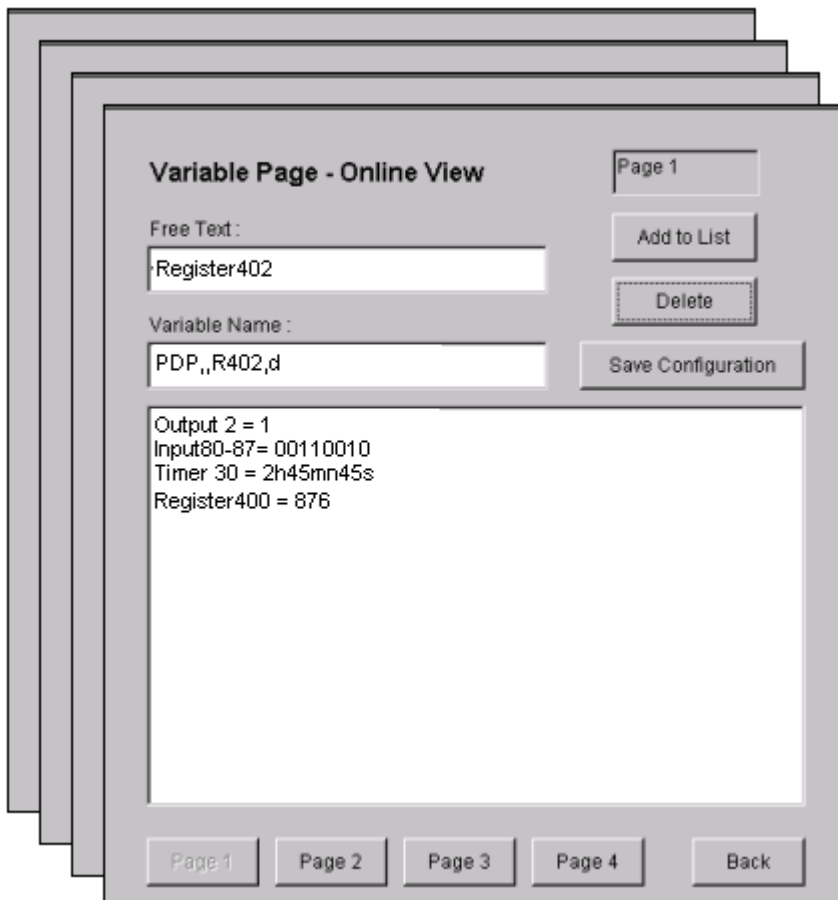
4

4.3.1 Vor der Verwendung der Variablenliste

Wenn Variablenseiten gespeichert sind, sind diese in Data Blocks des Anwenderprogramms gespeichert. Um also diese Funktion nutzen zu können, müssen Sie zuerst Data Blocks erstellen, dies wird in den Einstellungen des Web-Builder vorgenommen, durch anhängen des Kästchens «Variables lists»



Falls die Variablenseiten im Anwenderspeicher der PCD abgespeichert werden sollen so muss zuvor ein Speicherbereich reserviert worden sein. Dies wird im Web-Builder durch die Selektion der Option VarList Sites durchgeführt.



Free Text	Frei editierbarer Text zum Prozessdatenpunkt
Variable Name	Prozessdatenpunkt mit PDP-Kennung (ohne % Zeichen)
Add to List	Freier Text und PDP-Kennung werden in die Liste übernommen
Delete	Ausgewählter Listeneintrag wird gelöscht
Save Configuration	Aktuelle Einträge der Variablenseite werden in der PCD abgespeichert (Nur möglich mit Passwordebene 4 und wenn der entsprechende Speicherplatz in der PCD reserviert ist).
Page1...Page 4	Wechseln zu den anderen Variablenseiten
Back	Zurück zur Startseite.

4.4 Beschreibung der PCD Daten-Punkte

Die Einträge im Feld Variable Name haben folgende Syntax:

PDP„,<Adr>,<Format>

Adr Adresse des PCD Datenpunktes. Beschreibung untenstehend

Format Format des adressierten PCD Datenpunktes. Beschreibung untenstehend.

Adresse:

Bereich	Datablock Num	Grösse	Offset	Beispiel
I (Input) Eingang		B (Byte)	Ofs.BitNr	I10.3
		W (Word)	Ofs	IB20
		D (DWord)		
Q (Output) Ausgang		B (Byte)	Ofs.BitNr	Q30.4
		W (Word)	Ofs	QW15
		D (DWord)		
M (Merker)		B (Byte)	Ofs.BitNr	M40.5
		W (Word)	Ofs	MD33
		D (DWord)		
DB (Datablock)	Num.	DBX	Ofs.BitNr	DB20.DBX11.6
		DBB (Byte)	Ofs	DB21.DBB4
		DBW (Word)		
		DBD (DWord)		
T (Timer) Zeiten			TimerNum	T45
C (Counter) Zähler			CounterNum	C60

Format:

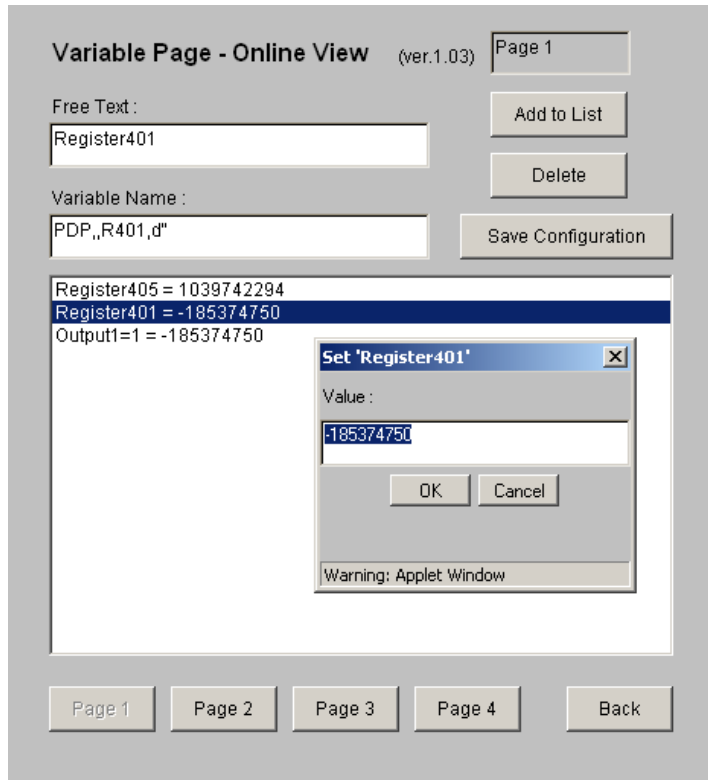
u	Dezimal ohne Vorzeichen	
d	Dezimal mit Vorzeichen	
b	Binär	bit, byte, word und dword
x,X	Hexadezimal (klein-, Grosschrift)	
o	Octal	
f	Fliesskomma	Nur für DWORD
s	Zeichenkette	Null terminierter String (C-Konvention)
n	S7 Zeichenkette	STEP7 String. Das erste Adressenbyte enthält die maximale Länge des Strings, das zweite Byte enthält die aktuelle Länge. Die eigentlichen Stringdaten sind ab dem 3 Byte abgespeichert.
t	Zeiten	Nur im Zusammenhang mit Zeiten.
D	Datum	Datum aus STEP7 DATE_AND_TIME (DT) Variablen. Dieses Format wird ab FW 2.300 unterstützt.
T	Zeit	Zeit aus STEP7 DATE_AND_TIME (DT) Variablen. Dieses Format wird ab FW 2.300 unterstützt.

Beispiel:

PDP,,M100.0,u
 PDP,,DB11.DBB40,n
 PDP,,MD33,f
 PDP,,QW3,X

4.4.1 Ändern von Einträgen

Prozessdatenpunkte in den Variablenseiten können geändert werden, indem man auf den spezifischen Prozessdatenpunkt mit Doppelklick zugreift.



4

4.5 Zugriff auf PCD Daten von HTML Seiten

Wenn Sie auf Prozessdatenpunkte von HTML-Seiten zugreifen möchten, müssen diese Prozessdatenpunkte mit einem speziellem Prozessdatenpunktschlüssel (PDP key) in den HTML-Seiten definiert sein. PDP-Schlüssel können in HTML-Seiten oder in Kontrollfunktionen definiert sein. Ein PDP-Schlüssel sieht gleich aus, wie im vorigen Kapitel beschrieben, jedoch ist er nun mit dem «%»-Zeichen umschlossen. Zwei Zeichen (%%) am Anfang und eines (%) am Ende.

Eine PDP-Kennung hat folgendes Format:

%%PDP,<NetAdr>,<Adr>,<Format>%

- NetAdr:** Reserviert, Wird zur Zeit nicht unterstützt
In diesem Feld darf kein Wert definiert werden
- Adr:** Adresse des PDP Datenpunktes. Beschreibung Kap. 4.4
- Format** Format des adressierten SPS Datenpunktes. Beschreibung Kap. 4.4



- In HTML Seiten müssen PDP-Schlüssel durch das Zeichen % abgegrenzt sein. In Kontrollfunktionen **werden keine Abgrenzungen (%) benötigt**.
- Ist der PDP-Schlüssel inkorrekt definiert, wird als Lese-Anweisung anstelle der Daten eine Fehlermeldung angezeigt. Bei einer Schreib-Anweisung, wird der Wert nicht zur PCD geschrieben.
- Der PDP-Schlüssel ist nur für Dateien mit der Endung *.html oder *.htm. Ist ein PDP Schlüssel in einem anderen Dateityp platziert, wird er ignoriert und kein Wert angegeben.
- Die Elementinformation im PDP Schlüssel ist nicht von der Schreibweise abhängig, der PDP selbst muss jedoch in Grossbuchstaben geschrieben sein.

Beispiel:

```
%%PDP,,M100.0,u%
%%PDP,,DB11.DBB40,n%
%%PDP,,MD33,f%
%%PDP,,QW3,X%
```

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Ident	STRING[12]	'SAIA xx7 CDB'	CDB identification
+14.0	ComParaSerial	STRING[40]	'COM1:PTP_MPI,RS232,19200,8,n,1'	Serial Interface or ...
+56.0	ComParaModem	STRING[40]	'// COM1:PTP_MPI,MODEM'	... Modem connection (now just as comment!)
+98.0	WebIndexDB	STRING[40]	'WEB:INDEXDB=900'	IndexDB -> Identical with WebBuilder!
+140.0	WebRamDisk	STRING[40]	'WEB:RAMDISK=16'	16k RamDisk

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Identificator	STRING[12]	'SAIA xx7 CDB'	CDB Identification
+14.0	Memory	STRING[8]	'MEM7:512'	Memoryrange of PCD
+24.0	Timeout	STRING[24]	'COM1:PTP_MPI,TIMEOUT=2'	Timeout of serial Port
+50.0	WebPara	STRING[30]	'COM1:PTP_MPI,RS232,38400,8,o,1'	Parameter of serial Port
+82.0	WebPassword	STRING[60]	'WEB:PASSWORD=,,, '	Password for Web-Server-> void= no password
+144.0	IndexDB	STRING[15]	'WEB:INDEXDB=900'	Index DB -> identical with Web-Builder
=162.0		END_STRUCT		

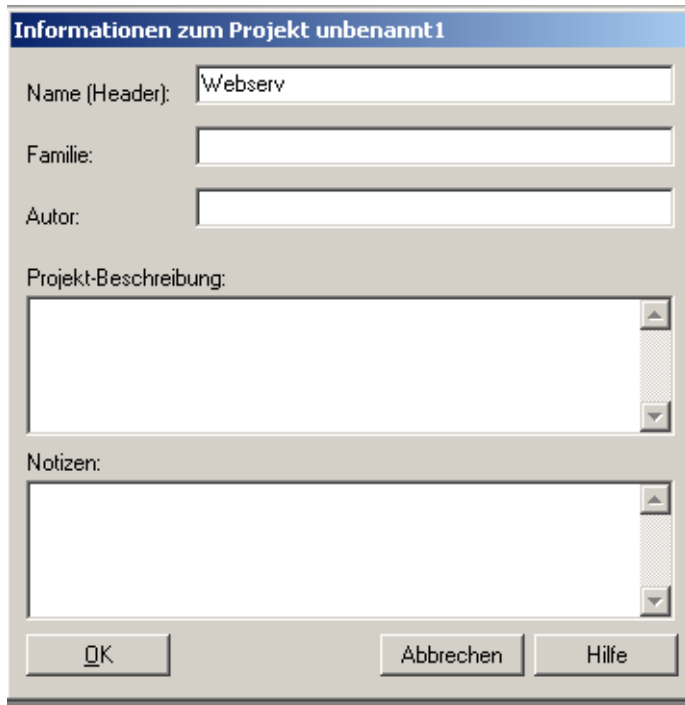
4.6 Konfiguration des Web-Servers mit dem SBC xx7 I/O-Builder

Die in Kapitel «4.7 Konfiguration und Definition des Web-Servers im Anwenderprogramm» beschriebenen Einstellungen können weitgehend und auf komfortablere Weise mit dem SBC xx7 I/O-Builder vorgenommen werden. Dazu unter «**Programme**», «**SBC**», «**xx7**», «**I/O-Builder**» dieses Werkzeug öffnen:

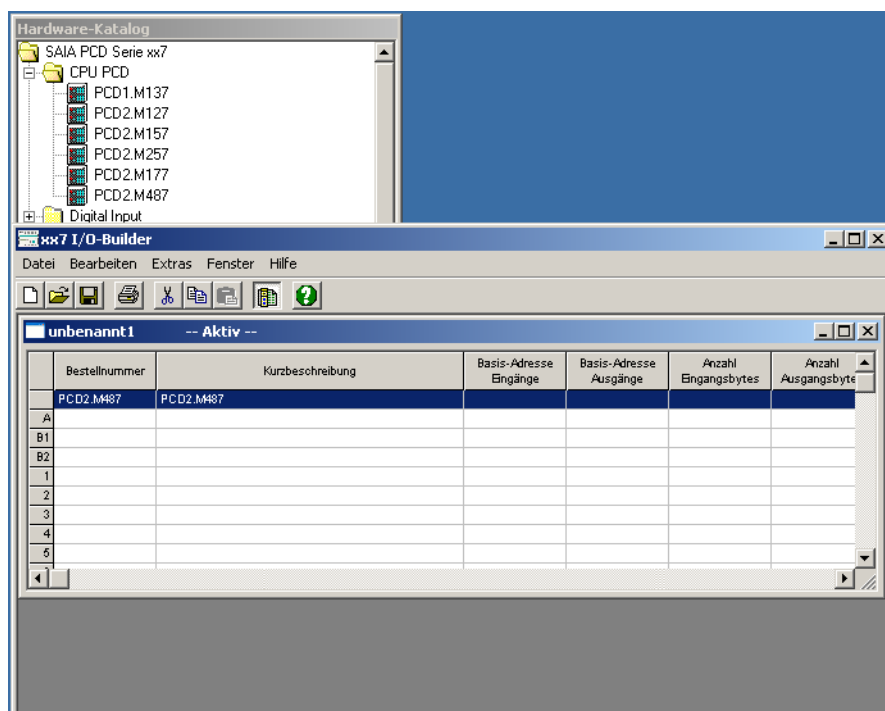


Neues Projekt anlegen

4

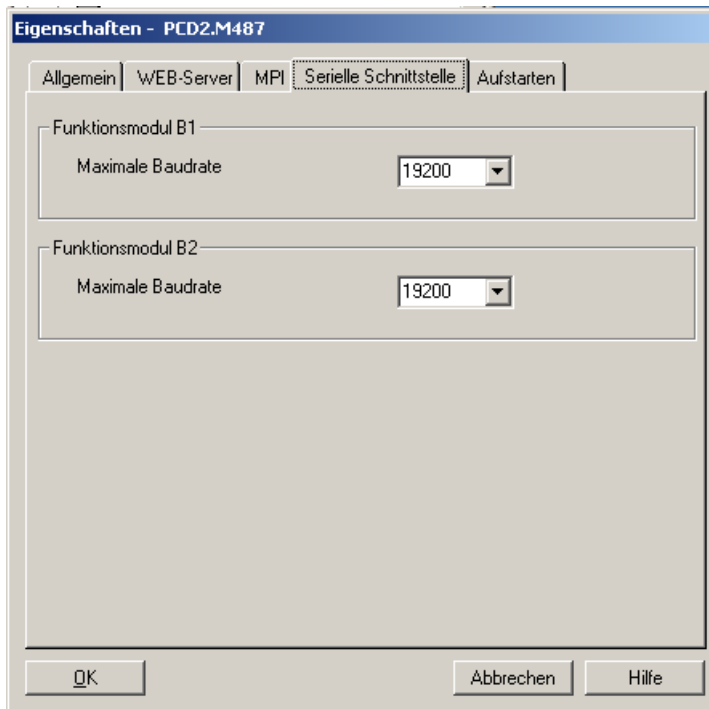


und eine PCD aus dem Hardware-Katalog auswählen auswählen.



4.6.1 Definition der seriellen Schnittstelle

Dazu auf die ausgewählte PCD doppelklicken und es öffnet sich folgendes Fenster. In diesem den Tab «Serielle Schnittstelle» wählen.

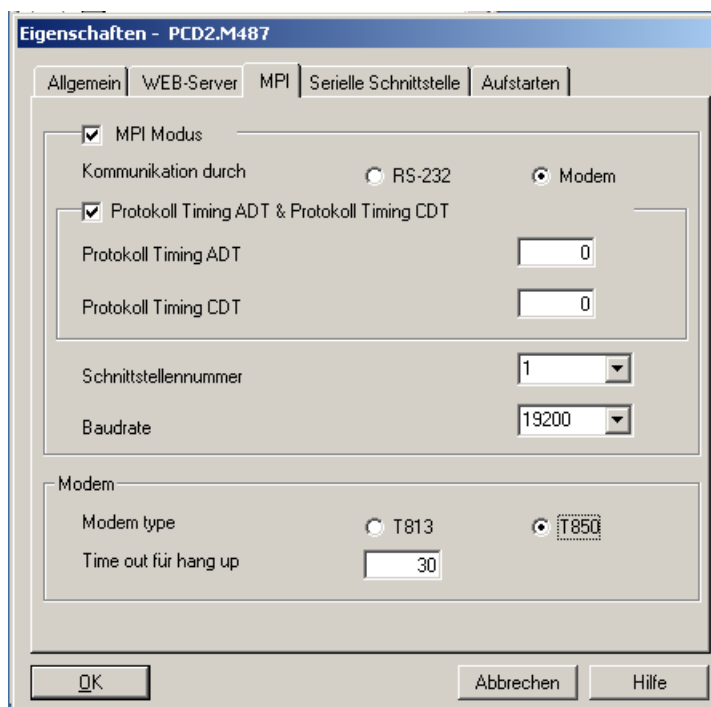


4

Sind weitere Einstellungen notwendig, können diese später im CDB des Anwender-Programms hinzugefügt werden.

4.6.2 Definition Modem

Hierzu den Tab «MPI» auswählen und MPI Modus anhaken.



Auswahl des Modems:

Für Analog: T813

Für ISDN: T850

Timeout für hang up:

Voreingestellt sind 30 Minuten

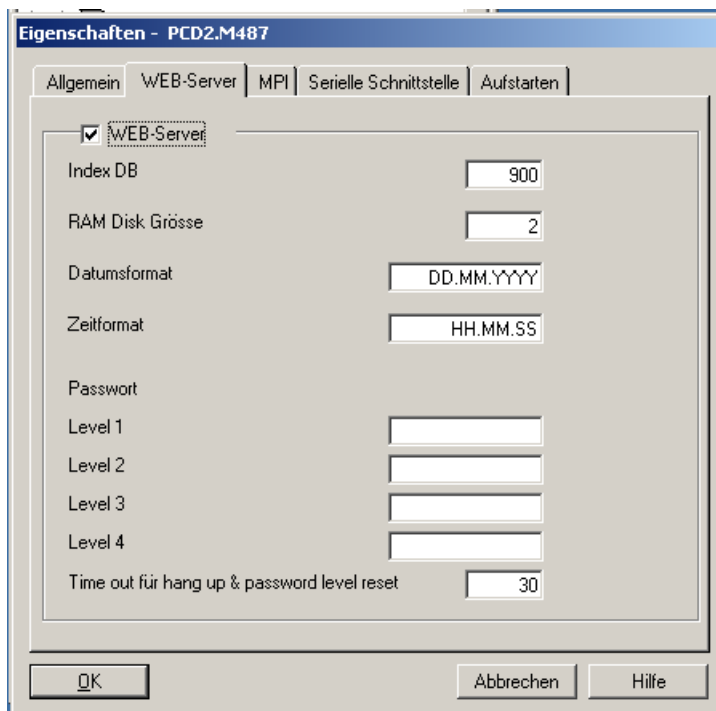
Protokollzeiten:

ADT 2000 ms bei direkter Verbindung
10000 ms mit Modem-Verbindung

CDT 220 ms bei direkter Verbindung
1000 ms mit Modem-Verbindung

4.6.3 Definition des Web-Servers

Dazu den Tab WEB-Server auswählen und WEB-Server anhaken.



4

Es können alle Einstellungen zur Definition des Web-Servers vorgenommen werden:

- Festlegen des Index DB
- Grösse der RAM Disk in kB
- Datumsformat
- Zeitformat
- Passwörter und Ebenen
- Timeout für hang up und Passwortlevel-Reset in Minuten (Voreinstellung 30 Minuten)

Mögliche Formate, Berechtigungen etc. sind im Kapitel 4.7.3 zu finden.

4.7 Konfiguration und Definition des Web-Servers im Anwenderprogramm

4.7.1 Definition der seriellen Schnittstelle

Konfiguration

Die serielle Schnittstelle wird konfiguriert, indem die entsprechenden Parameter in einen CDB (Configuration Data Block) geschrieben werden.

Kommunikations Parameter

4

Falls kein CDB oder keine entsprechenden Einträge im CDB vorhanden sind, so wird per Vorgabewerte die Schnittstelle 1 mit 19'200 Baud, 8 Datenbits, ungerader (odd) Parität und 1 Stopbit als MPI Port/Web-Server initialisiert. Dabei kann über diesen Port entweder eine Web-Server Applikation aufgerufen oder mit der STEP®7 Programmiersoftware kommuniziert werden.

Mit den folgenden Parametern kann die Schnittstelle eingestellt werden:

COM<n>: PTP_MPI,RS232,<baud>,<data>,<parity>,<stop>

n	Schnittstelle 1 oder 3		
baud	110...38'400 Baud. Hinweis: Port 2 unterstützt 38'400 Baud nicht		
bata	Datenbits (7 oder 8)		
parity	Parität. Mögliche Werte sind:		
	• E	Even	Gerade
	• O	Odd	Ungerade
	• N	None	Keine
	• L	Force Low	Fix auf 0
	• H	Force High	Fix auf 1
stop	Stopbits (1 oder 2)		



Um die Einstellungen zu aktivieren, muss die Speisespannung aus- und wieder eingeschaltet werden!

Beispiel:

```
COM1: PTP_MPI,RS232,9600,8,N,1
COM2: PTP_MPI,RS232,19200,8,E,2
COM1: PTP_MPI,RS232,38400,8,O,1
```

4.7.2 Definition Modem

An die Schnittstelle 1 der PCD kann ein Saia Modem angeschlossen werden. Damit die FW das Modem richtig initialisiert, muss folgender Befehl im CDB definiert werden:

Analog Modem T813

```
COM1: PTP_MPI,T813           → Ab FW 2.0.206
COM1: PTP_MPI,MODEM        → nur für Rückwärtscompatibilität mit FW V 2.0.200
```

ISDN Modem T850

COM1: PTP_MPI,T850[,<prot>[,<msn>]] → Ab FW 2.0.206

prot Protokoll. Möglich sind:

V.110
 HDLC_ASYNC
 HDLC_TRANSPARENT
 BYTE_TRANSPARENT
X.75-NL
 V.120
 X.31B
 X.31D

4



Wenn kein, oder kein gültiges Protokoll definiert wurde, wird das Modem per Vorgabe mit **X.75-NL** initialisiert.

msn Multiple subscriber number (Mehrfach-Teilnehmernummer)

- Bis zu 22 Ziffern (0...9; *; #)
- Wird keine oder eine ungültige Msn definiert, wird die Msn mit * definiert → d.h. Modem antwortet immer

Beispiel:

COM1: PTP_MPI,T850,V.110,21

COM1: PTP_MPI,T850,V.110



- Um die Einstellungen zu aktivieren, muss die Speisespannung aus- und wieder eingeschaltet werden!
- Modemkommunikation nicht unterbrechen. Status des DCD-Signals prüfen, bevor die Modemverbindung unterbrochen wird.
- Das Modem muss so konfiguriert sein, dass bei DTR=0 das Modem zurückgesetzt wird. (AT&D3)

Umschalten zwischen FW und Anwenderprogramm

Eine mit dem Befehl «PTP_MPI...» konfigurierte Schnittstelle (mit oder ohne Modem) kann bei Bedarf auch vom Anwenderprogramm genutzt werden, z.B. um aus dem Anwenderprogramm Texte auf die Schnittstelle zu senden. Dies erfolgt durch den Aufruf des SFC 200:

FW hat Kontrolle über die Schnittstelle Aufruf SFC 200, mit VKE = 0
 Warten bis VKE nach dem Aufruf auf 1 ist.

Anwenderprogramm hat Kontrolle über die Schnittstelle Aufruf SFC 200, mit VKE = 1
 Warten bis VKE nach dem Aufruf auf 1 ist.

Timeout

Der Datenaustausch auf der Schnittstelle kann mit einer Timeoutzeit gekoppelt werden. Dabei wird die definierte Timeoutzeit bei jedem empfangenen oder gesendeten Telegramm neu gestartet. Sollte während der Timeoutzeit kein Datenaustausch stattfinden, so wird nach Ablauf der Timeoutzeit das Passwort und eine bestehende Modemverbindung zurückgesetzt. Vorgabewert für die Timeoutzeit sind 30 Minuten.

COM<n>: PTP_MPI,TIMEOUT=<timeout>

n Schnittstelle 1 oder 3
 timeout Timeoutzeit in Minuten
 Durch setzen des Wertes auf 0 wird die Timeoutfunktion ausgeschaltet.



Um die Einstellungen zu aktivieren, muss die Speisespannung aus- und wieder eingeschaltet werden oder die CPU muss von Stop in Run geschaltet werden!

Beispiel:

COM1: PTP_MPI,TIMEOUT=5
 COM2: PTP_MPI,TIMEOUT=0

4

Protokollzeiten

Bei Bedarf kann eine Zeichenverzugszeit (character delay time CDT; Vorgabewert = 220ms bei direkter Verbindung, 1000ms bei Modemverbindung) oder eine Antwortverzugszeit (answer delay time ADT; Vorgabewert = 2000ms bei direkter Verbindung, 10000ms bei Modemverbindung) für die Punkt zu Punkt Verbindung zwischen dem PC und der PCD definiert werden. Durch die Anpassung dieser zwei Zeiten können Verbindungen über lange Distanzen oder über Modem realisiert werden. Es ist wichtig, das auf beiden Seiten (PC und SPS) die gleichen Werte eingestellt werden.

COM<n>: PTP_MPI,CDT=<cdt>
 COM<n>: PTP_MPI,ADT=<adt>

n: Schnittstelle 1 oder 3
 cdt, adt Timeoutzeiten in ms



Um die Einstellungen zu aktivieren, muss die Speisespannung aus- und wieder eingeschaltet werden!

Beispiel:

COM1: PTP_MPI,CDT=300
 COM2: PTP_MPI,ADT=2500

4.7.3 Definition des Web-Servers

Konfiguration

Analog zu den Kommunikationsparametern der Seriellen Schnittstelle wird der Web-Server im CDB konfiguriert

IndexDB

Um den Zugriff auf Ihre eigenen HTML Seiten zu ermöglichen, muss im CDB ein IndexDB definiert sein. Der IndexDB ist der erste, vom Konvertierungstool «Web-Builder» generierte DB.

4

WEB: INDEXDB = <num>

num Nummer des Index-Datenblockes



- Um die Einstellungen zu aktivieren, muss die Speisespannung aus- und wieder eingeschaltet werden oder die CPU muss von Stop in Run geschaltet werden!
- Wird vom Web-Browser eine Datei geladen so wird folgende Suchreihenfolge eingehalten:
 1. DBs (Dateien in DBs abgespeichert)
 2. Firmware (Dateien in der FW abgespeichert)
 3. PC (Dateien auf der Harddisk des PCs gespeichert)
- Auf Offset 2 (DBW2) des IndexDB kann die aktuelle Passwordebene des Web-Servers gelesen werden
- Auf Offset 4 (DBW4) des IndexDB kann ein Lebenszeichen des Web-Servers gelesen werden.
Dieses Wort wird bei jeder Anfrage des Web-Servers inkrementiert.

Beispiel:

WEB: INDEXDB=900

WEB: INDEXDB=256

Passwort

Es können bis zu 4 Passwörter definiert werden.

Jedes Passwort entspricht einer Passwordebene.

Dadurch können im Web-Server in vier verschiedenen Passwordebene angemeldet und die entsprechenden Funktionen ausgeführt werden. In der jeweiligen Passwordebene sind alle Rechte verfügbar ausser der Rechte der nächst höheren Ebenen.

Ebene 1	Lesen von HTML Seiten
Ebene 2	Lesen von PCD Datenpunkten
Ebene 3	Schreiben von PCD Datenpunkten
Ebene 4	Modifizieren / Speichern von Variablenlisten

Per Vorgabe ist kein Passwort definiert und die Passwordebene 4 ist aktiv.

Die Passwörter werden wie folgt definiert.

WEB: PASSWORD=<pw1>,<pw2>,<pw3>,<pw4>

pw1... Passwort für jede Ebene. Es ist nicht nötig alle 4 Passwörter zu
pw4 definieren.



- Um die Einstellungen zu aktivieren, muss die Speisespannung aus- und wieder eingeschaltet werden oder die CPU muss von Stop in Run geschaltet werden!
- Gross-/Kleinschreibung der Passwörter wird nicht unterstützt. SBC und Sbc bedeuten das gleiche.
- Es müssen nicht alle 4 Passwörter definiert werden. Wird z.B. nur das Passwort für die Passwortebene 4 definiert, so kann dies wie folgt geschehen: WEB: PASSWORD=,,,<pw4>

4

Beispiel:

WEB:PASSWORD=,sbc1,sbc2,sbc3,sbc4

WEB:PASSWORD=, , ,sbc4

WEB:PASSWORD=, ,sbc3,sbc4

RAM Disk

Der Web-Server hat in der FW einen RAM Disk Bereich von 2kB reserviert, in dem Daten zwischengespeichert werden können.

Bei Bedarf kann die RAM Disk wie folgt vergrößert werden:

WEB: RAMDISK=<size>

Size Grösse der RAM Disk in kB



- Um die Einstellungen zu aktivieren, muss die Speisespannung aus- und wieder eingeschaltet werden!
- Der Speicherbereich der RAM Disk wird vom Anwenderspeicher abgezogen. Vergewissern, dass noch genügend Anwenderspeicher frei ist, falls der RAM Disk Bereich vergrößert wird.

Beispiel:

WEB:RAMDISK=4

WEB:RAMDISK=16

Datumsformat von DATA_AND_TIME Variablen

Der Web-Server zeigt das Datum aus der STEP®7 DATA_AND_TIME (DT) Variablen per Vorgabe wie folgt an: DD.MM.JJJJ (Beispiel: 10.09.2001).

Dieses Format kann im CDB geändert werden:

WEB: DATEFORMAT=<F1><T1><F2><T2><F3>

F1,F2,F3	D	Anzeige Tag ohne Füllnull
	DD	Anzeige Tag mit Füllnull
	M	Anzeige Monat ohne Füllnull
	MM	Anzeige Monat mit Füllnull
	YY	Anzeige Jahr
	YYYY	Anzeige Jahr, 4-stellig
T1,T2		Gültiger Separator. Ein gültiger Separator muss in folgendem Bereich liegen: Dezimalwerte von ASCII-Zeichen zwischen 33 und 47 oder zwischen 58 und 64.



- Die Änderungen werden beim nächsten Stop zu RUN Wechsel übernommen
- Bei fehlerhaften Eingaben wird das default Format verwendet
- Der Befehl wird ab FW 2.300 unterstützt

Beispiel:

WEB:DATEFORMAT=D/M/YY

WEB:DATEFORMAT=YYYY.MM.DD

Zeitformat von DATA_AND_TIME Variablen

Der Web-Server zeigt die Zeit aus der STEP7 DATA_AND_TIME (DT) Variablen per Vorgabe wie folgt an: HH:MM:SS (Beispiel: 11:55:00).

4

Dieses Format kann im CDB geändert werden:

WEB: TIMEFORMAT=<F1><T1><F2><T2><F3>

F1,F2,F3	H	Anzeige Stunden ohne Füllnull
	HH	Anzeige Stunden mit Füllnull
	M	Anzeige Minuten ohne Füllnull
	MM	Anzeige Minuten mit Füllnull
	S	Anzeige Sekunden ohne Füllnull
	SS	Anzeige mit Füllnull
T1,T2	Gültiger Separator. Ein gültiger Separator muss in folgendem Bereich liegen: Dezimalwerte von ASCII-Zeichen zwischen 33 und 47 oder zwischen 58 und 64.	



- Die Änderungen werden beim nächsten Stop zu RUN Wechsel übernommen
- Bei fehlerhaften Eingaben wird das default Format verwendet
- Der Befehl wird ab FW 2.300 unterstützt

Beispiel:

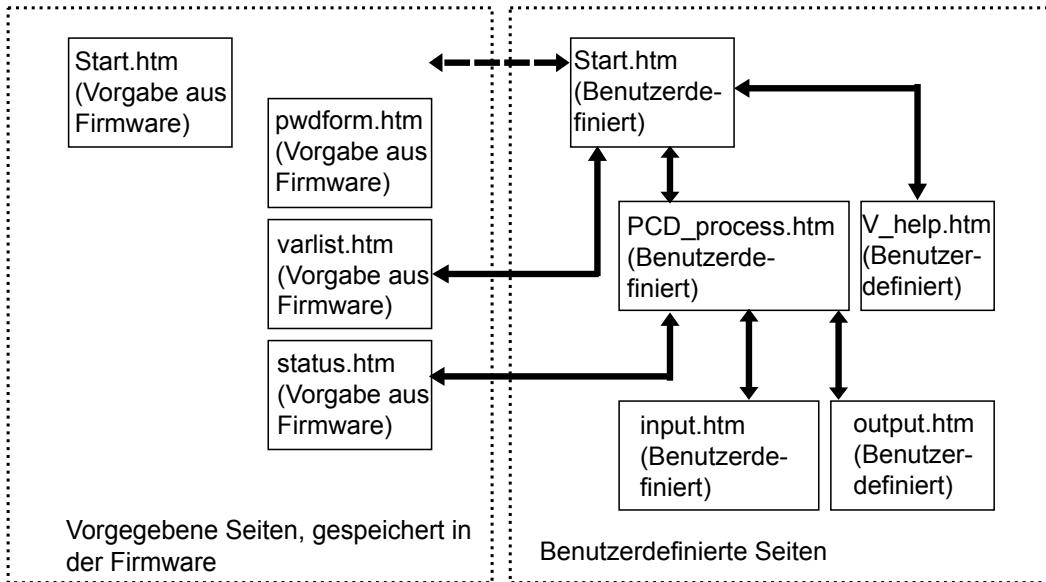
WEB: TIMEFORMAT=H/M/S

WEB: TIMEFORMAT=HH.MM.SS

5. Beispiel Web-Projekt

5.1 Seitenstruktur

Im folgenden Beispiel werden 4 HTML Seiten definiert und auf die PCD geladen. Die Struktur sieht folgendermassen aus:

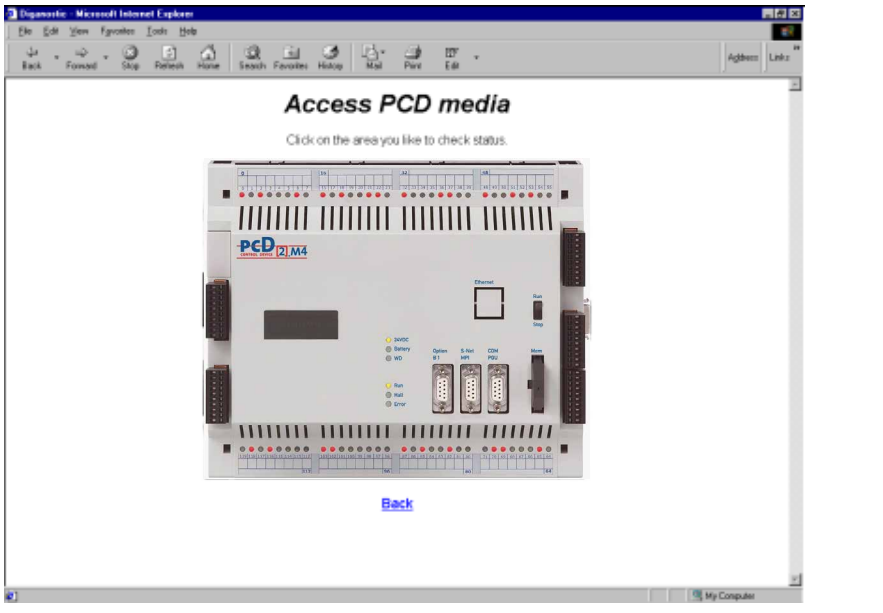
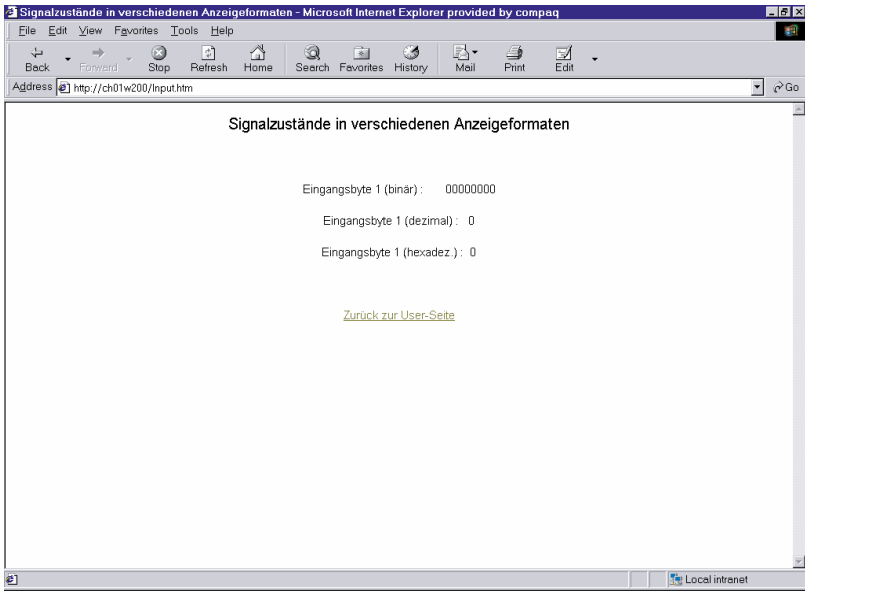
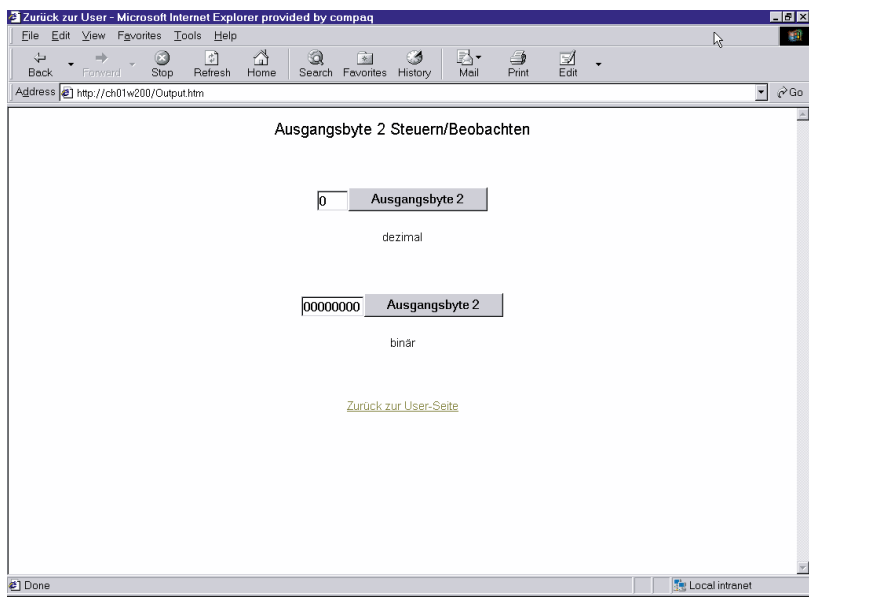


5

Wir werden nicht alle HTML Seiten behandeln, sondern nur die interessanten. Mit interessant ist gemeint, nur die Seiten welche PCD Ressourcen enthalten. Im Detail wird beschrieben, wie die input.html- und die output.html Seiten zu erstellen sind.

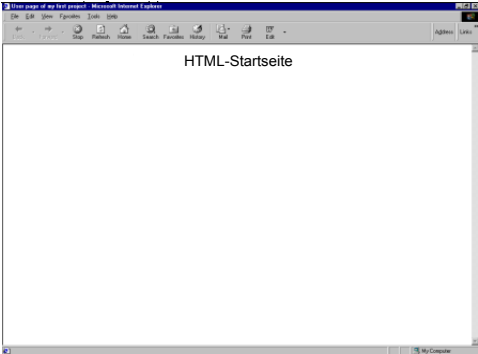
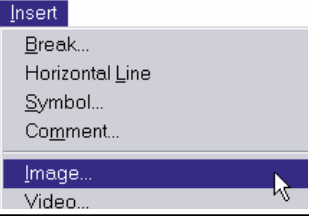
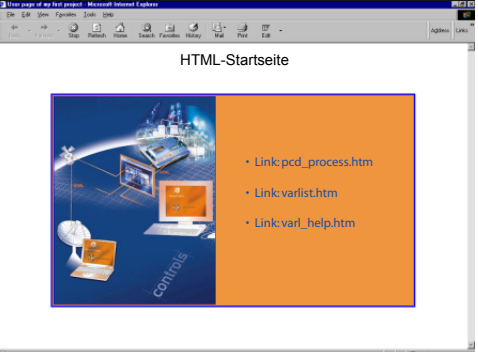
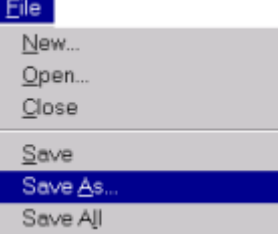
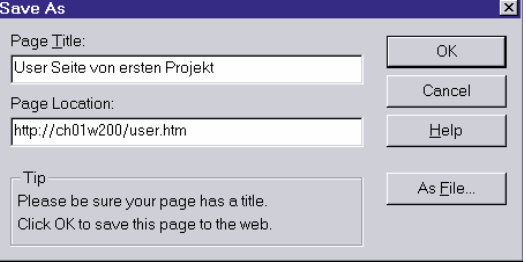
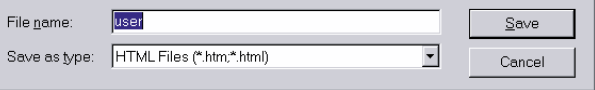
Die 4 im Anwender-Programm definierten Seiten erscheinen wie folgt im Web-Browser:

<p>Seite: start.htm</p> <p>Start-Seite, mit einem Bild und Hyperlinks zu den Seiten: varlist.htm pcd_process.htm varl_help.htm</p>	
--	--

<p>Seite: pcd_process.htm</p> <p>Anzeige und Zugriff auf die Seiten:</p> <p>input.htm output.htm</p>	
<p>Seite: input.html</p> <p>Seite input.htm</p> <p>Anzeige der Eingänge 0 bis 7</p>	
<p>Seite output.htm</p> <p>Anzeige und Modifikation der Ausgänge 0 bis 7</p>	

5.2 Aufbau der Seiten

HTML-Editor z.B: Frontpage Express starten und Startseite anlegen

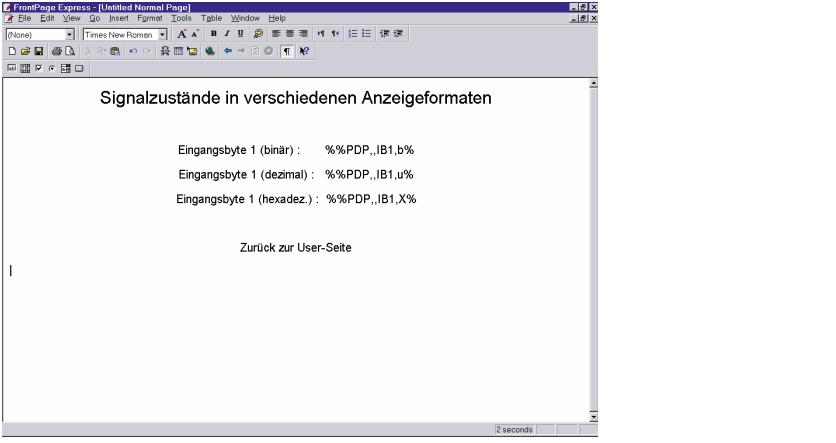
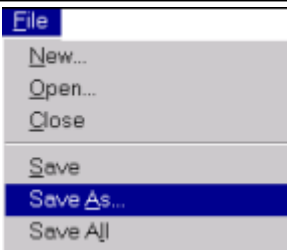
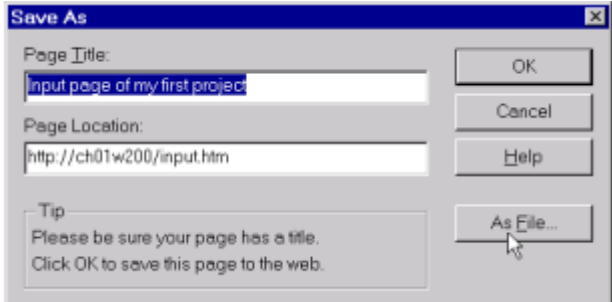
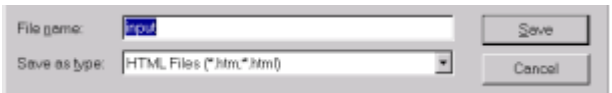
1	Text einfügen	
2	Einfügen des Bildes mit Insert, Image...	 
3	Speichern der Datei mit Save as...	
4	Seitentitel korrekt eingeben, As File... wählen	
5	File name: user eingeben, Datei nicht schliessen	

5.2.1 PCD Datenpunkte in den HTML Seiten

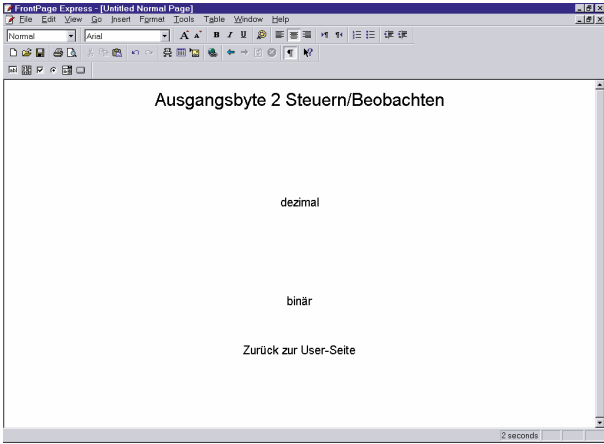
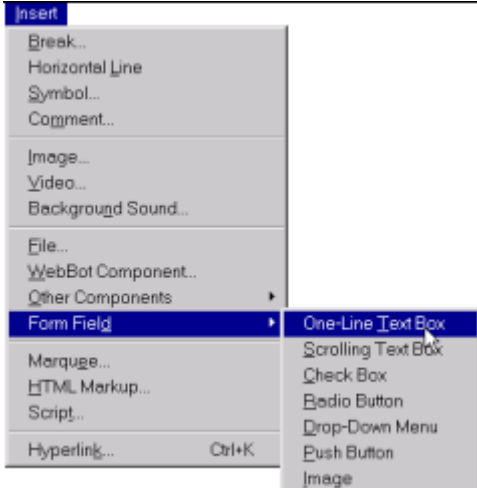
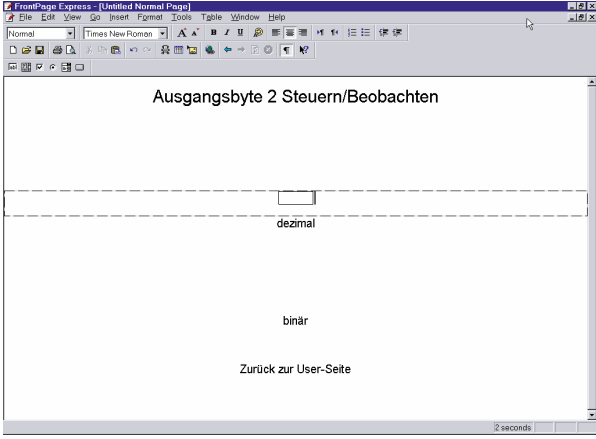
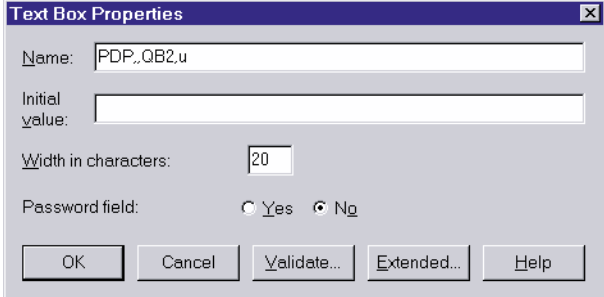
Hier wird mit dem FrontPage Express-Editor gezeigt, wie ein dynamischer Bereich, in dem Werte von PCD-Elementen angezeigt werden, auf Ihrer HTML Seite dargestellt werden kann. Sie können auch jeden anderen HTML-Editor verwenden.

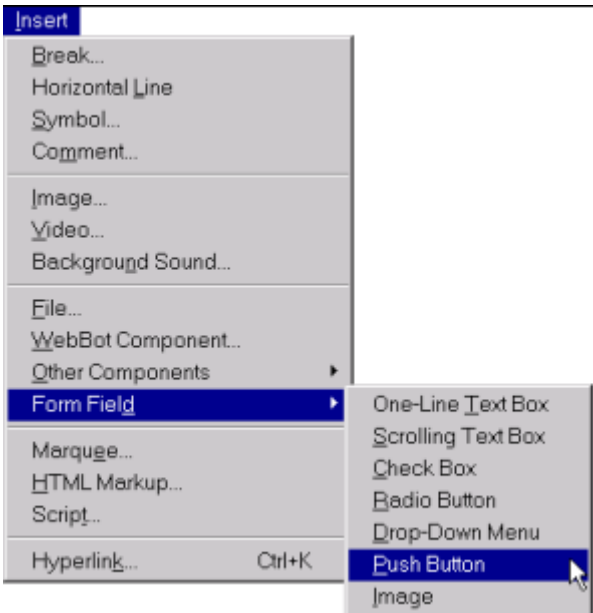
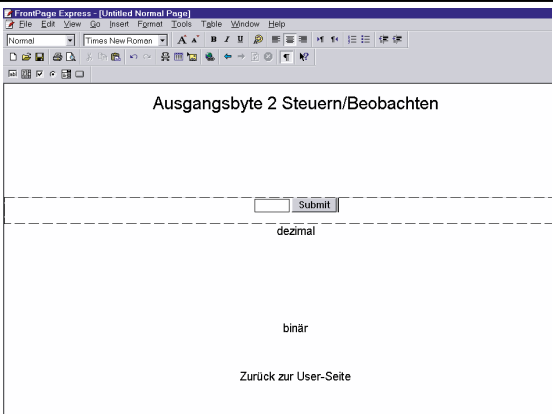
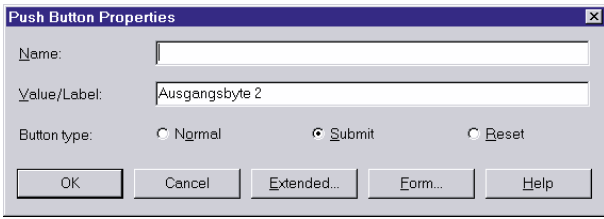
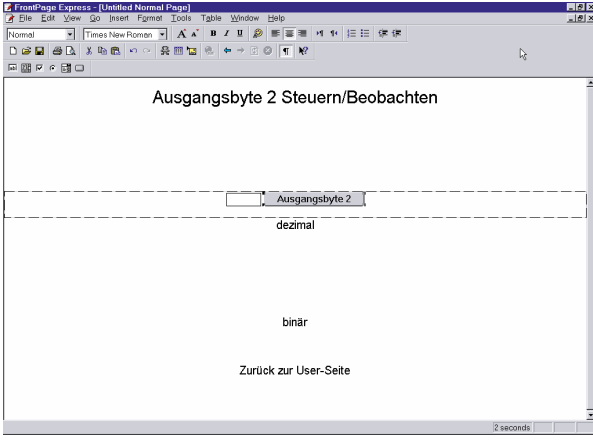
5.2.2 Seite: input.htm

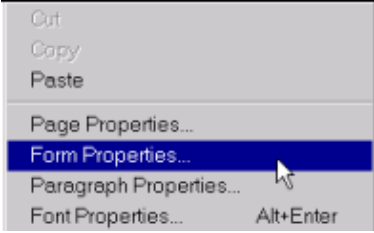
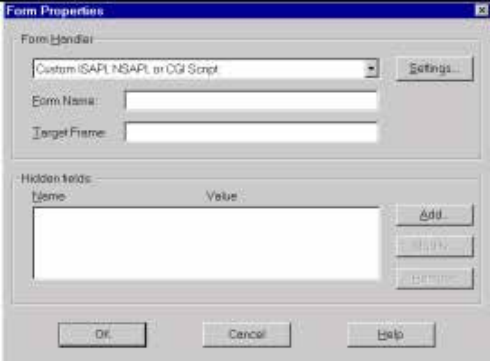
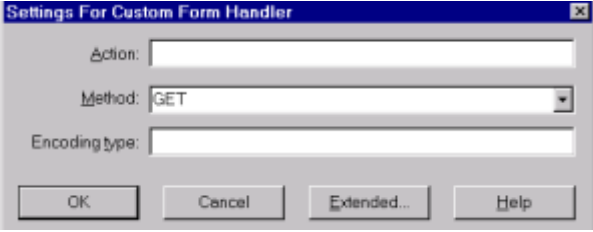
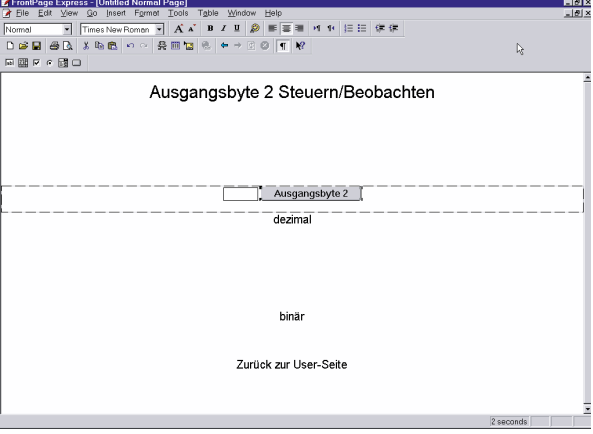
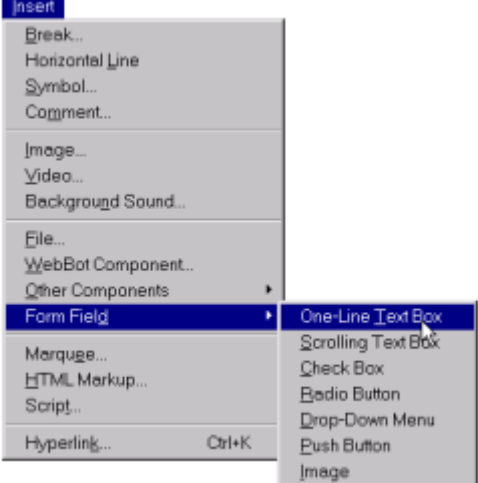
Auf diesen Seiten wird gezeigt, wie einfach es ist, ein Feld in einer Seite zu erstellen, in dem Werte angezeigt werden, wenn wir die HTML Seite von der PCD laden. Dies funktioniert für die hier gezeigten Eingänge, genauso wie für jeden anderen Bereich.

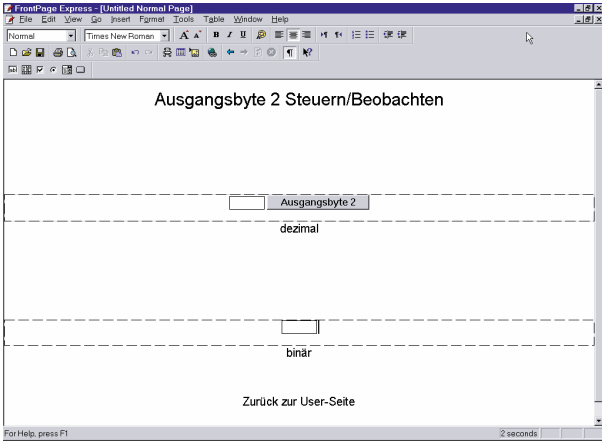
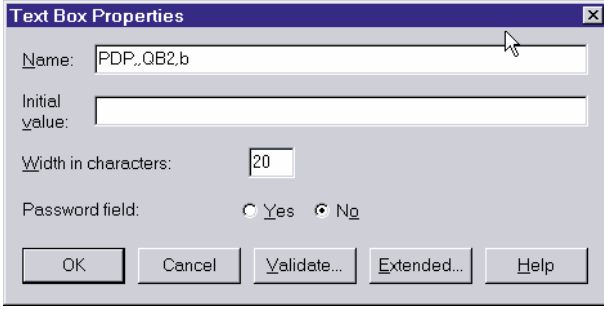
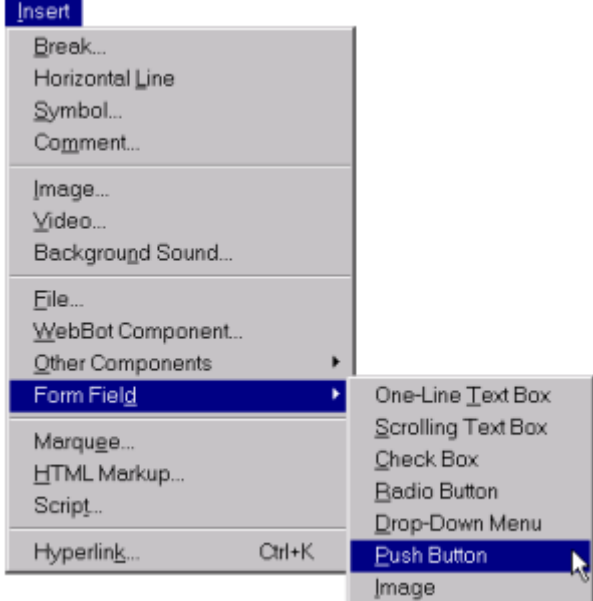
<p>1 Öffnen Sie eine neue, leere Seite.</p> <p>Fügen Sie den Text ein. Positionieren Sie den Text.</p> <p>Syntax für PDP Schlüssel: %%PDP,,IB1,b% %%PDP,,IB1,u% %%PDP,,IB1,X%</p>	
<p>2 Datei speichern mit «Save As»</p>	
<p>3 Diesen Titel korrekt eintragen.</p> <p>Wählen «As File»</p>	
<p>4 Dateiname einfügen: Input. Seite nicht schliessen.</p>	

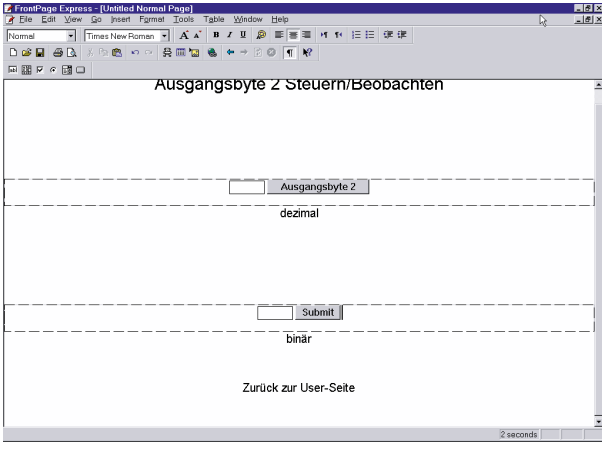
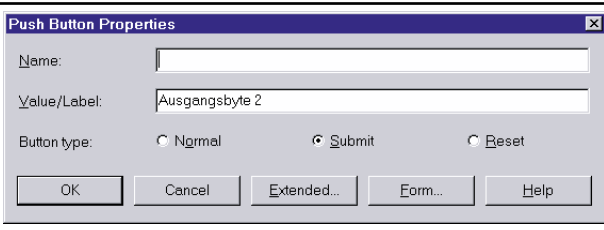
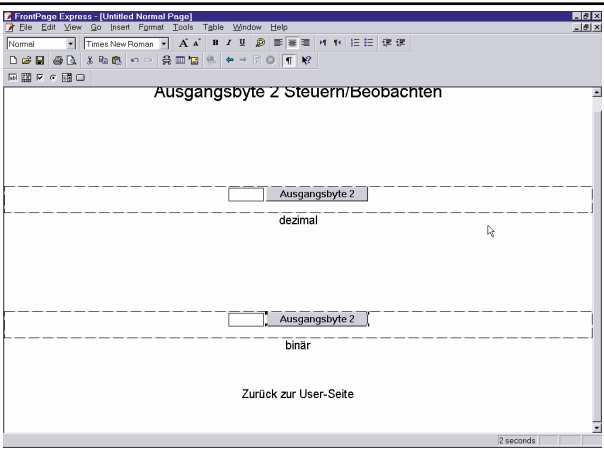
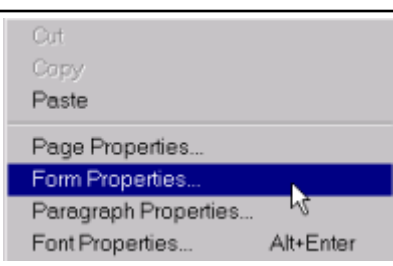
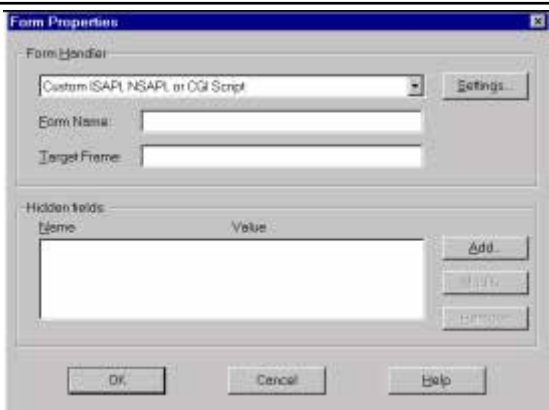
5.2.3 Seite: output.htm

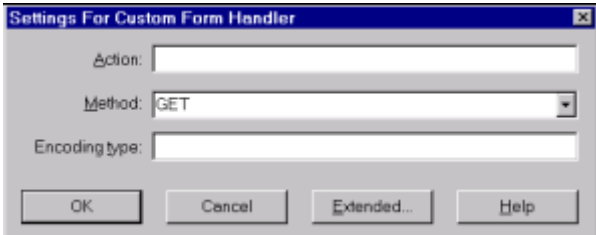
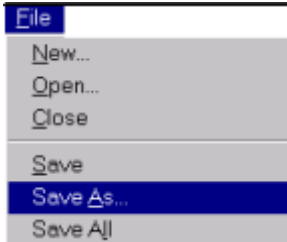
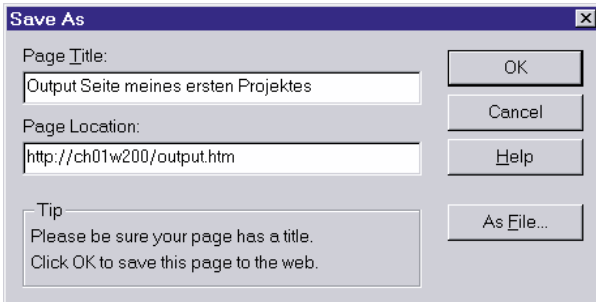
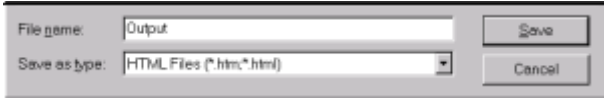
<p>1 Neue, leere Seite öffnen.</p> <p>Fügen Sie Text ein.</p> <p>Positionieren Sie den Text.</p>	
<p>2 Fügen Sie Input/Output-Felder ein: Positionieren Sie den Cursor über den Text «dezimal»'. Fügen Sie ein Feld ein mit: Insert, FormField, One-Line Text Box.</p>	
<p>3 Positionieren Sie das Feld.</p>	
<p>4 Doppelklick auf das Feld. Fügen Sie den PDP-Schlüssel ein.</p> <p>Beachten Sie die richtige Syntax: PDP,,QB2,u</p>	

<p>5</p>	<p>Fügen Sie einen «Push Button» ein. Positionieren Sie den Cursor in der Nähe des Eingabe Feldes, im selben Rahmen (Form Field). Dann: Insert, Form Field, Push Button</p>	
<p>6</p>	<p>Doppelklick auf den Button.</p>	
<p>7</p>	<p>Fügen Sie den Text ein, der auf dem Button erscheinen soll.</p>	
<p>8</p>	<p>Die Seite erscheint nun folgendermaßen:</p>	

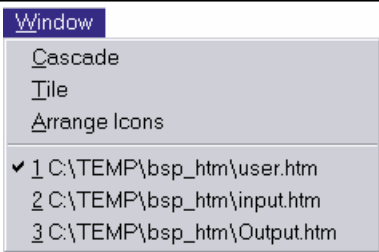
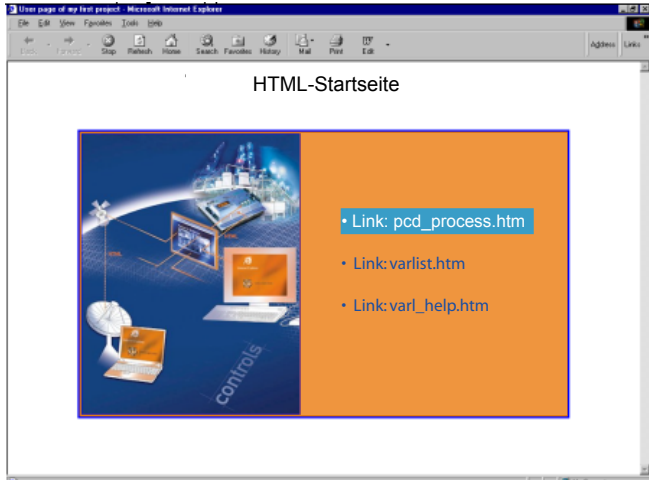
<p>9</p>	<p>Den Cursor rechts vom Button positionieren. Mit der rechten Maustaste klicken. «Form Properties» wählen.</p>	
<p>10</p>	<p>«Settings» klicken.</p>	
<p>11</p>	<p>Method: «GET» eingeben, Fenster schliessen</p>	
<p>12</p>	<p>Die Seite erscheint nun folgendermassen. Derselbe Vorgang wie in den Schritten 2 bis 11 beschrieben, folgt nun für das Binärwert Eingabefeld.</p>	
<p>13</p>	<p>Fügen Sie die Input/Output Felder ein: Den Cursor über den Text «binär» positionieren. Feld einfügen mit: Insert, FormField, One-Line Text Box.</p>	

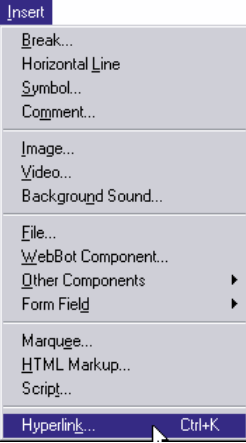
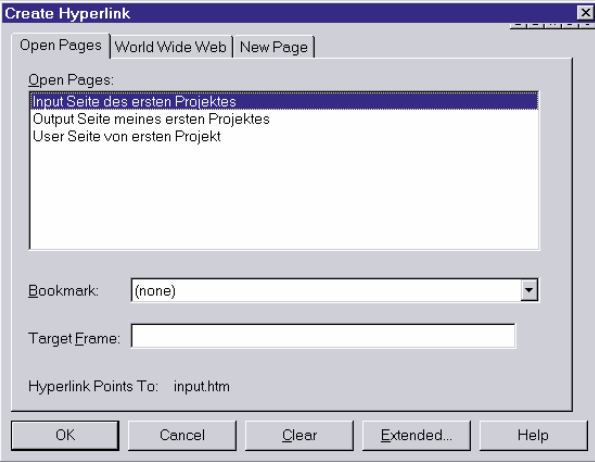
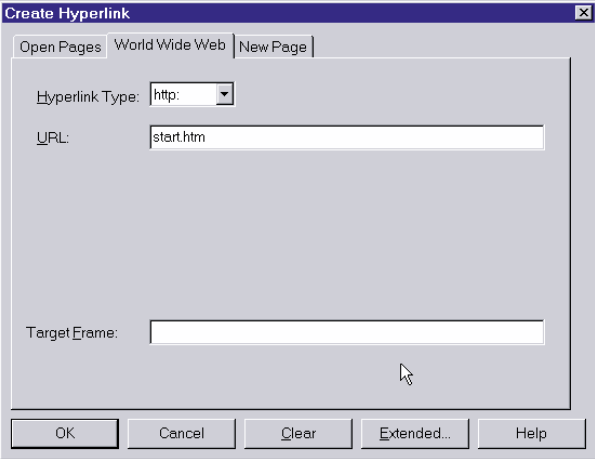
<p>14</p>	<p>Positionieren Sie das Feld.</p>	
<p>15</p>	<p>Doppelklick auf das Feld. Fügen Sie den PDP Schlüssel ein. Beachten Sie die korrekte Syntax: PDP,,QB2,b</p>	
<p>16</p>	<p>Fügen Sie einen «Push Button» ein. Positionieren Sie den Cursor in der Nähe des Eingabe- Feldes, im selben Rahmen (Form Field).Dann: Insert, Form Field, Push Button</p>	

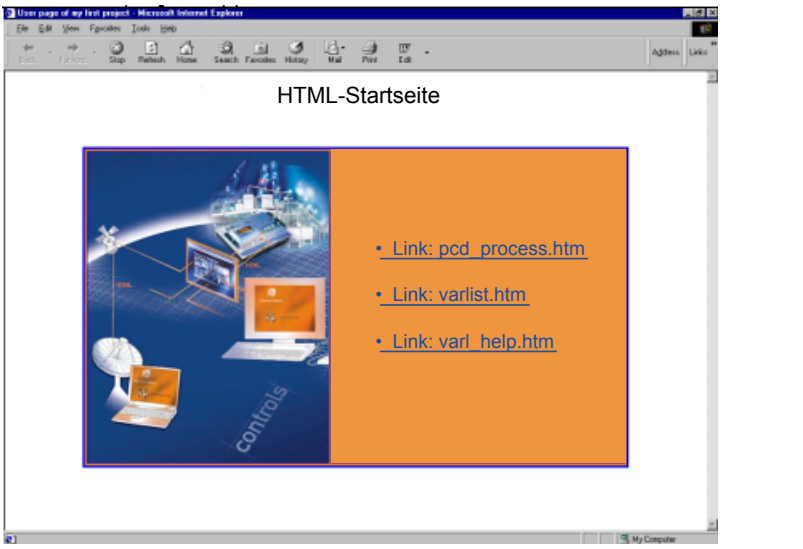
<p>17</p>	<p>Doppelklick auf den Button</p>	
<p>18</p>	<p>Fügen Sie den Text ein, der auf dem Button erscheinen soll.</p>	
<p>19</p>	<p>Die Seite erscheint nun folgedermassen.</p>	
<p>20</p>	<p>Den Cursor rechts vom Button positionieren. Mit der rechten Maustaste klicken. «Form Properties» wählen.</p>	
<p>21</p>	<p>«Settings» wählen.</p>	

<p>22 Method: GET eingeben. Fenster schließen.</p>	
<p>23 Datei speichern mit «Save As».</p>	
<p>24 Diesen Titel korrekt eintragen. «As File» wählen.</p>	
<p>25 Dateiname einfügen: Output Seite nicht schließen.</p>	

5.3 Seiten verlinken


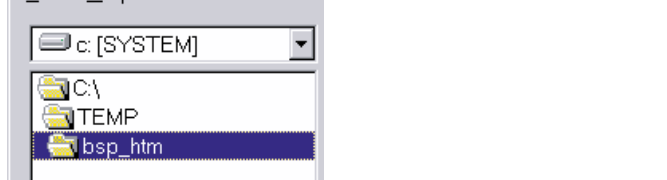


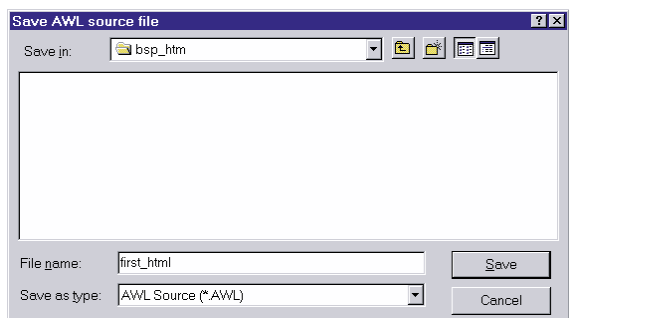
<p>1 Alle zu verlinkenden HTML Seiten müssen offen sein. Seite user.htm in Frontpage Express laden mit «Window»</p>	
<p>2 Markieren des zu verlinkenden Textes</p>	

<p>3</p>	<p>Hyperlink erstellen mit Insert, Hyperlink</p>	 <p>The screenshot shows the 'Insert' menu with the following options: Break..., Horizontal Line, Symbol..., Comment..., Image..., Video..., Background Sound..., File..., WebBot Component..., Other Components, Form Field, Marquee..., HTML Markup..., Script..., and Hyperlink... (highlighted with a mouse cursor). The keyboard shortcut 'Ctrl+K' is shown next to the Hyperlink... option.</p>
<p>4</p>	<p>Auswahl der zu verlinkenden Seite (Ordner Open pages) Die Links zu weiteren Seiten (z.B. Output.htm) sind analog zu erstellen</p>	 <p>The screenshot shows the 'Create Hyperlink' dialog box. It has tabs for 'Open Pages', 'World Wide Web', and 'New Page'. The 'Open Pages' tab is active, showing a list of files: 'Input Seite des ersten Projektes' (selected), 'Output Seite meines ersten Projektes', and 'User Seite von ersten Projekt'. Below the list, there are fields for 'Bookmark:' (set to '(none)'), 'Target Frame:', and 'Hyperlink Points To: input.htm'. Buttons for 'OK', 'Cancel', 'Clear', 'Extended...', and 'Help' are at the bottom.</p>
<p>5</p>	<p>Der Link zur Startseite erfolgt im World Wide Web. Als URL muss start.htm eingetragen werden</p>	 <p>The screenshot shows the 'Create Hyperlink' dialog box with the 'World Wide Web' tab selected. The 'Hyperlink Type:' dropdown is set to 'http:'. The 'URL:' field contains 'start.htm'. The 'Target Frame:' field is empty. Buttons for 'OK', 'Cancel', 'Clear', 'Extended...', and 'Help' are at the bottom.</p>

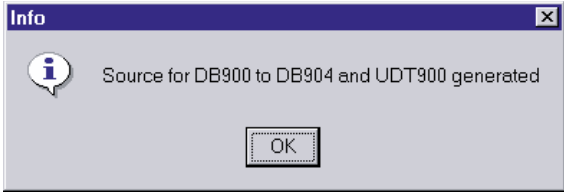
<p>6 Die fertige Seite sieht dann folgendermassen aus</p>	
---	--

5

5.4 Generierung der DBs

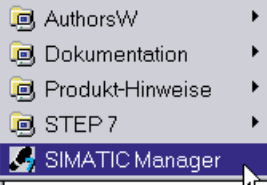

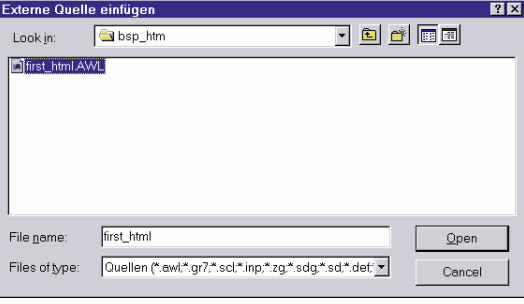
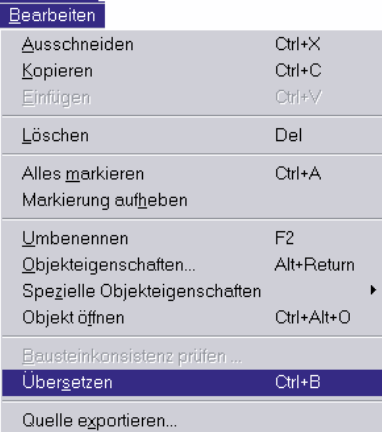
<p>1 Starten des Web-Builders</p>									
<p>2 Auswahl des Pfades in welchem die zuvor mit FrontPage Express entwickelten Dateien und Bilder abgespeichert wurden.</p>									
<p>3 Auswahl der Dateien user.htm, input.htm, output.htm etc.</p>	<table border="1" data-bbox="700 1155 1369 1283"> <thead> <tr> <th>FileName</th> <th>FilePath</th> </tr> </thead> <tbody> <tr> <td>Input.htm</td> <td>C:\TEMP\bsp_html</td> </tr> <tr> <td>Output.htm</td> <td>C:\TEMP\bsp_html</td> </tr> <tr> <td>user.htm</td> <td>C:\TEMP\bsp_html</td> </tr> </tbody> </table>	FileName	FilePath	Input.htm	C:\TEMP\bsp_html	Output.htm	C:\TEMP\bsp_html	user.htm	C:\TEMP\bsp_html
FileName	FilePath								
Input.htm	C:\TEMP\bsp_html								
Output.htm	C:\TEMP\bsp_html								
user.htm	C:\TEMP\bsp_html								
<p>4 Eingabe des IndexDB</p>									
<p>5 Konvertierung der Dateien und Bilder</p>									
<p>6 Abspeichern der DBs in der Datei first_html</p>									

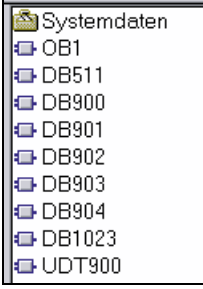
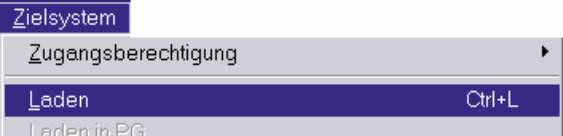
Einfügen der HTML-Seiten in das Anwenderprogramm

7	Bestätigen der Generierung	
---	----------------------------	--

5.5 Einfügen der HTML-Seiten in das Anwenderprogramm



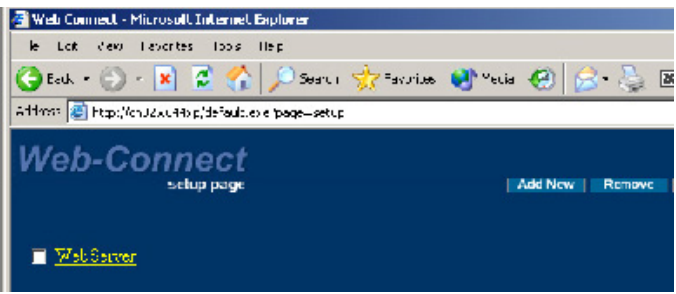

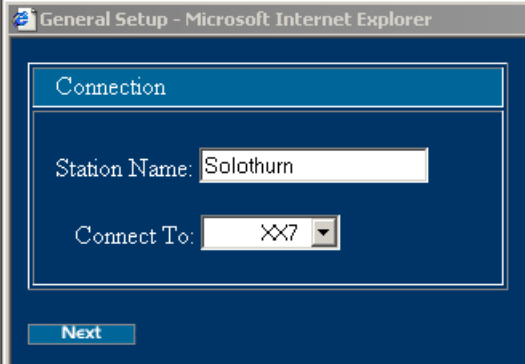
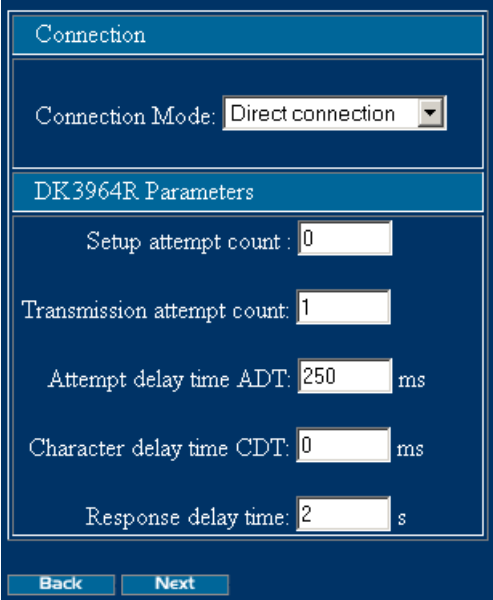
5

1	Starten des Siemens® Simatic® Managers von Step®7.	
2	Einfügen der externen Quellen im Menü: Einfügen, Externe Quelle	
3	Auswahl der *.awl Datei welche mit dem Web-Builder generiert wurde	
4	Übersetzen der neu eingeführten Quelldatei mit dem Menü: Bearbeiten, Übersetzen	

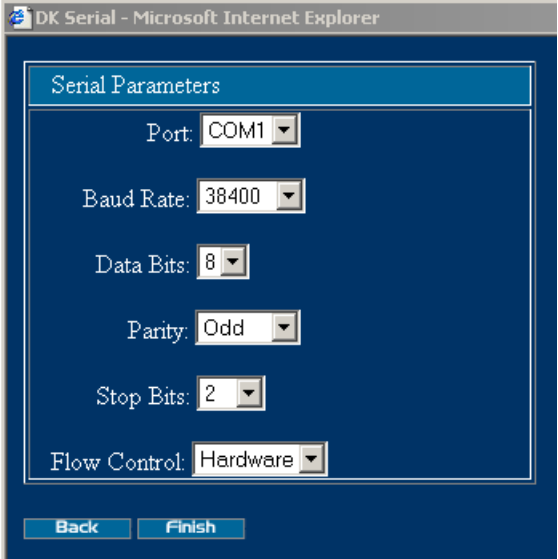
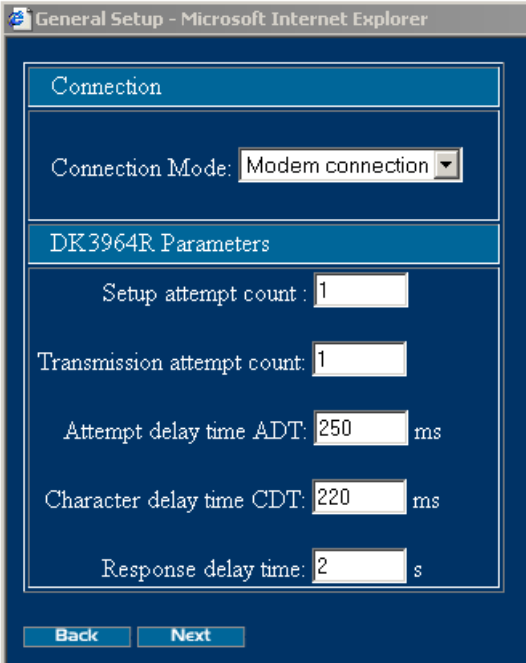
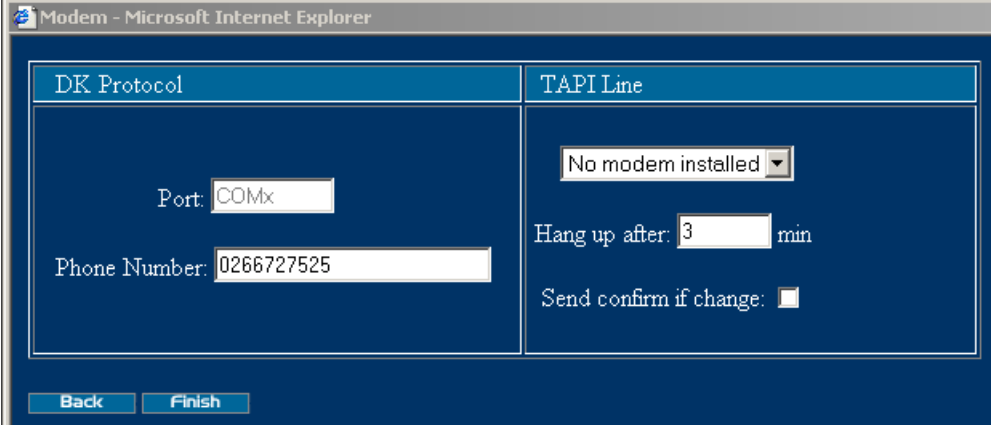
5	<p>Nach dem Übersetzen wurden die in der Quelldatei abgespeicherten DBs im Bausteinbehälter des Anwenderprogrammes abgelegt.</p> <p>Im Beispiel sind dies: DBs 900 bis 904 und der UDT900</p>																															
6	<p>Definition des Web-Servers im CDB</p> <p>In Beispiel ist dies der CD 1023</p> <p>Darin muss mindestens der IndexDB definiert werden. Wenn keine Defintion der Schnittstelle erfolgt, wird Default-mässig die Schnittstelle 1 mit 19'200 Baud, 8 Datenbits, ungerader Parität und 1 Stop Bit definiert</p>	<table border="1"> <thead> <tr> <th>Adresse</th> <th>Name</th> <th>Typ</th> <th>Anfangswert</th> <th>Kommentar</th> </tr> </thead> <tbody> <tr> <td>0.0</td> <td></td> <td>STRUCT</td> <td></td> <td></td> </tr> <tr> <td>+0.0</td> <td>Identificator</td> <td>STRING[12]</td> <td>'SAIA xx7 CDB'</td> <td>CDB Identification</td> </tr> <tr> <td>+14.0</td> <td>WebPara</td> <td>STRING[30]</td> <td>'COM1:PTP_MPI,RS232,38400,8,o,1</td> <td>Parameter of serial Port</td> </tr> <tr> <td>+46.0</td> <td>IndexDB</td> <td>STRING[15]</td> <td>'WEB:INDEXDB=900'</td> <td>Index DB -> identical with Web-Bu.</td> </tr> <tr> <td>=64.0</td> <td></td> <td>END_STRUCT</td> <td></td> <td></td> </tr> </tbody> </table>	Adresse	Name	Typ	Anfangswert	Kommentar	0.0		STRUCT			+0.0	Identificator	STRING[12]	'SAIA xx7 CDB'	CDB Identification	+14.0	WebPara	STRING[30]	'COM1:PTP_MPI,RS232,38400,8,o,1	Parameter of serial Port	+46.0	IndexDB	STRING[15]	'WEB:INDEXDB=900'	Index DB -> identical with Web-Bu.	=64.0		END_STRUCT		
Adresse	Name	Typ	Anfangswert	Kommentar																												
0.0		STRUCT																														
+0.0	Identificator	STRING[12]	'SAIA xx7 CDB'	CDB Identification																												
+14.0	WebPara	STRING[30]	'COM1:PTP_MPI,RS232,38400,8,o,1	Parameter of serial Port																												
+46.0	IndexDB	STRING[15]	'WEB:INDEXDB=900'	Index DB -> identical with Web-Bu.																												
=64.0		END_STRUCT																														
7	<p>Laden des Anwenderprogrammes.</p> <p>Hinweis: Da im Beispiel auf Ein- und Ausgänge zugegriffen wird, müssen diese im Projekt konfiguriert sein.</p> <p>Falls der CDB das erste Mal geladen wurde, oder falls die Einträge im CDB geändert wurden, muss zur Aktivierung der CDB-Einträge die Speisepannung der Steuerung aus- und eingeschaltet werden</p>																															

5

5.6 Verwenden der Web-Connect SW

1	Starten der Web-Connect SW.	
2	Starten des Web-Browsers.	
3	Fügen Sie für die URL ein: HTTP://localhost/setup Siehe auch 4.2.4 Konfiguration mit SBC xx7 Die Setup- Seite erscheint. Hier haben Sie die Möglichkeit eine neue Station hinzuzufügen (durch «Add New»)	
4	Klicken Sie auf «Add New»	
5	Es erscheint die Seite mit den Wahlmöglichkeiten: Saia PCD, xx7 oder RIO. Bei xx7 kommt die «General Setup» Seite mit diesen Feldern: <ul style="list-style-type: none"> • Connection: Hier haben Sie 2 Möglichkeiten (Direct connection und Modem connection) zur Auswahl, wie der PC und die PCD miteinander kommunizieren sollen. • DK3964R Parameter: Es wird empfohlen, die voreingestellten Werte zu belassen. Das nächste Fenster, das erscheint, ist abhängig vom Kommunikations-Modus (Connection Mode), den Sie ausgewählt haben. Bestätigen Sie zum Schluss mit «Next»..	 

5

<p>5.1</p>	<p>Direct connection: Wenn dieser Modus gewählt ist, müssen Sie folgendes eingeben: <u>Port</u> des verwendeten PCs, <u>Baud Rate</u> der Kommunikation.</p> <p>Wählen Sie Data Bits, Parity, Stop Bits und Flow control die die PCD in diesem Modus bei Verbindungen verwenden soll.</p> <p>Nachdem alle Einstellungen gesetzt sind, mit "Finish" bestätigen.</p>	
<p>5.2</p>	<p>Modem connection: In diesem Modus müssen Sie folgendes eingeben: <u>Port</u>: wird automatisch definiert, abhängig vom gewählten Modem. <u>Phone Number</u>: Telefonnummer.</p> <p>TAPI Line: Ein Modem auswählen. <u>Hang up after</u>: Wenn in der hier angegeben Zeitspanne keine Anfragen vom Browser eingehen, wird die Modemverbindung automatisch getrennt. <u>Send confirm change</u>: ist dieses Feld bestätigt, wird eine Bestätigung des Abbruchs der Modem Verbindung angefordert. Nachdem alle Einstellungen gesetzt sind, mit «Finish» bestätigen.</p>	
		

6 Problembhebung

6.1 Variablen Liste (varlist)

F: Wie kann ich auf die Varlist-Funktionen zugreifen ?

A: Um auf die varlist Funktionen zugreifen zu können, müssen Sie folgende URL eingeben : http://pc_name/station_name/varlist.htm

F: Varlist läuft nicht, nichts passiert !!!

A: Überprüfen Sie, ob die Java Einstellungen des Browsers korrekt sind und ein von uns empfohlener Browser verwendet wird. Weitere Informationen dazu im Kapitel: **3.3.3 Cache Memory des PCs.**

A Anhang A

A.1 HTML Seiten ohne Editor erstellen

Die folgenden Kapitel beschreiben das Erstellen von HTML Seiten ohne HTML Editor. Dies führt zu dynamischeren HTML Seiten, deren Inhalte automatisch aufgefrischt werden und deren Datenpunkte mit Java applets geändert werden können.

Gute Kenntnisse über HTML, JavaScript und Java Programmierung werden zum Verständnis der folgenden Kapitel vorausgesetzt.

Alle Beispiele wurden mit dem Browser Internet Explorer 6.0 erstellt und getestet.

A.2 CGI (Common Gateway Interface) des Web-Servers

Im Web-Server (in der PCD) bestehen bereits 4 (ausführbare) CGIs :

- readval.exe
- writeval.exe
- ordervalues.exe
- readfile.exe

Mit diesen CGIs können Werte der PCD mittels tags manipuliert werden.



- 1) Diese CGIs, haben die gleichen Einschränkungen wie die Web-Server Zugriffsebenen, d.h., wenn der Web-Server Zugriffsebene 2 besitzt, ist es nur möglich, Werte zu lesen und nicht zu schreiben, etc...
- 2) Alle diese Funktionen basieren auf der «GET Methode» des HTTP Protokolls.



A.2.1 Readval.exe

Read value wird gebraucht, um einzelne PCD Datenpunkte zu lesen.

Die Syntax ist:

`http://pc_name/station_name/cgi-bin/readVal.exe?TAG`

pc_name: der PC mit Web-Connect.
Station_name: Name der PCD, entsprechend ihrer Kommunikationseinstellungen
/cgi-bin/readVal.exe: das CGI
? : Trennung zwischen CGI und Parametern.
TAG : das zu lesende media tag.

Die erhaltene Antwort enthält die Werte der einzelnen Datenpunkte.

Beispiel:

Anfrage Merkerwort 100 lesen, Wert in dezimalem Format:

`http://pc_name/station_name/cgi-bin/readVal.exe?PDP,,MW100,u`

Antwort im HTTP Protokoll:

```
HTTP/1.0 200 OK[0x0A]Content-Type:text/plain[0x0A]Content-
Length:4[0x0A][0x0A]3400
```

Der Wert ist: 3400

Zugriffsebene der PCD

Auch die momentane Zugriffsebene der PCD kann gelesen werden, dies dient dazu einzelne Funktionalitäten während des Schreibens der eigenen Java Applets zu beschränken.

`http://pc_name/station_name/cgi-bin/readVal.exe? Password Level`

Als Antwort erhält man die momentan eingestellte Zugriffsebene des Web-Servers (Wert zwischen 0 und 4).

A.2.2 Writeval.exe

Dient dazu einzelne PCD Media zu schreiben.

Die Syntax ist:

`http://pc_name/station_name/cgi-bin/writeval.exe? TAG+xxx`

pc_name:	der PC mit Web-Connect.
Station_name:	Name der PCD, entsprechend ihrer Kommunikationseinstellungen
/cgi-bin/writeval.exe:	das CGI
?	Trennung zwischen CGI und Parametern.
TAG :	das zu schreibende media tag.
+	Abtrennung der Parameter
xxx:	der zu schreibende Wert, das Format ist bereits im Tag festgelegt!

Als Antwort erhält man:

```
HTTP/1.0 200 OK[0x0A]Content-Type: text/plain[0x0A]Content-Length:
3[0x0A][0x0A]xxx
```

xxx: ist der gesendete Wert

[0x0A]: ist ein hexadezimaler Wert, der nicht in ASCII Zeichen dargestellt werden kann.

Beispiel

Wert 2300 in das Merkerwort 100 in dezimalem Format schreiben.

`http://pc_name/station_name/cgi-bin/writeVal.exe?PDP,,MW100,u+2300`

Antwort im HTTP Protokoll:

```
HTTP/1.0 200 OK[0x0A]Content-Type:text/plain[0x0A]Content-
Length:4[0x0A] [0x0A]2300
```

Der eingeschriebene Wert ist: 2300

A.2.3 Ordervalues.exe

Ordervalues wird gebraucht, wenn gleichzeitig auf mehrere PCD Media zugegriffen wird. Es kann eine Liste der zu lesenden Media definiert werden. Dies definiert jedoch nur die Media-Liste, zum lesen ist «readfile.exe» zu benutzen, siehe nächstes Kapitel.

Die Syntax ist:

`http://pc_name/station_name/cgi-bin/OrderValues.exe?listname+dummy+ssss+TAG1+TAG2+TAG3+TAG4+TAG5+TAG6+TAG7`



pc_name:	der PC mit Web-Connect.
Station_name:	Name der PCD, entsprechend ihrer Kommunikationseinstellungen
/cgi-bin/ordervalues.exe:	das CGI
?:	Trennung zwischen CGI und Parametern.
Listname:	Name der Medialiste (Datei), die aufgerufen, aufgelistet, referenziert werden soll. Dieser Name wird gebraucht um die Media zu lesen, oder die Liste zu überschreiben. Man kann mehrere Listen im Web-Server anlegen, abhängig vom vorhandenen Speicherplatz auf dem Web-Server RAM Disk.
+:	Abtrennung der Parameter
dummy:	Dies ist ein Parameter, der zur Zeit noch nicht genutzt wird, allerdings muss er in der Anfrage enthalten sein, also Wort «dummy» in die Anfrage schreiben.
ssss:	Hier muss die notwendige Speichergrösse auf der RAM Diskette des Web-Servers angegeben werden, damit die aktuelle Medialiste abgespeichert werden kann und die Antworten aufbereitet werden können. Als Antwort wird die Medialiste inklusive der Werte ausgegeben. Siehe readfile.exe. Es muss also genügend Platz für die Liste und die Antworten festgelegt werden. Die Grösse muss in Byte angegeben werden, so bedeutet die Eingabe 1000, 1000 Bytes sind für die Liste festgelegt. Der Vorgabewert ist 2000 Bytes, es kann aber auch mehr Platz vorgesehen werden.
TAGx:	Media, auf das zugegriffen werden soll.

Als Antwort erhält man:

„HTTP/1.0 200 OK[0x0A]Content-Type: text/plain [0x0A][0x0A]done[0x0A]“, d.h. die Anfrage wurde korrekt empfangen, aber es heisst nicht, dass die Syntax der Tags korrekt sind.

[0x0A]: ist ein hexadezimaler Wert, der nicht in ASCII Zeichen dargestellt werden kann.

Beispiel:

Eine Liste mit Tags erstellen mit dem Namen «room2». Die Liste enthält 3 Tags, die sich auf die Media Merkerwort 100, Merkerwort 102 und Merker 104.0 beziehen sollen.

```
http://pc_name/station_name /cgi-bin/OrderValues.exe?room2+dummy+1000+PDP,,MW100,u+PDP,,MW102,u+PDP,,M104.0,b
```

Die Antwort ist:

```
HTTP/1.0 200 OK[0x0A]Content-Type:text/plain[0x0A][0x0A]Done.[0x0A]
```

A.2.4 Readfile.exe

Readfile dient dazu, die mit der «ordervalue» Function vordefinierten Media Werte zu lesen. Es wird eine List der Tags und deren Werte ausgegeben.

Die Syntax ist:

`http://pc_name/station_name/cgi-bin/ReadFile.exe?listname`

pc_name: der PC mit Web-Connect.
Station_name: Name der PCD, entsprechend ihrer Kommunikationseinstellungen
/cgi-bin/readfile.exe: das CGI
?: Trennung zwischen CGI und Parametern.
Listname: Name der Medialiste (datei), die aufgerufen werden soll (vorher festgelegt in «ordervalue»).

HTTP/1.0 200 OK.Content-Typecontent/unknown Content-Length:
 343[0x0A][0x0A]TAG1=20689[0x0A]TAG2=61377[0x0A]TAG3=543[0x0A]
 TAG4=543[0x0A] TAG5=543[0x0A]

Die festgelegten Tags und deren Werte werden ausgegeben.

[0x0A]: ist ein hexadezimaler Wert, der nicht in ASCII Zeichen dargestellt werden kann.

Beispiel

In diesem Beispiel lesen wir die Liste, die wir im Kapitel OrderValues angelegt haben. Der Listenname ist also «room2».

Anfrage:

`http://pc_name/station_name/cgi-bin/ReadFile.exe?room2`

Antwort:

HTTP/1.0 200 OK[0x0A]Content-Type: content/
 unknown[0x0A]Content-Length: 47[0x0A] [0x0A]PDP,,MW100,u=3450[0x0A]
 PDP,,MW102,u=320[0x0A] PDP,,M104.0,b=0[0x0A]

A.3 HTML Code

In diesem Kapitel wird erklärt wie man direkt mit HTML und dem Web-Server arbeitet, jedoch nicht wie man in HTML programmiert.

A.3.1 Werte in die PCD schreiben

Im diesem Handbuch ist bereits beschrieben, wie Werte in die PCD geschrieben werden, das Beispiel wurde mit dem FrontPage Express Editor erstellt. Dies wird nun direkt im HTML Code ausgeführt.

- 1) Zuerst wird ein Formular angelegt und Methode «GET» benutzt. In diesem Formular brauchen wir ein Texteingabefeld, um Werte einzuschreiben und eine Taste, um das Formular an den Server zu übergeben.
- 2) Das Texteingabefeld wird definiert mit dem Schlüsselwort «input» und dem Typ «text». Sehr wichtig ist der Name, denn dieser ist das Tag, das sich auf das PCD Media bezieht. Hier ist der Bezug Merkerwort 200, das Format ist binär. Vor und nach dem Tag kein % Zeichen eingeben, sonst wird das Tag durch den Wert ersetzt.
- 3) Das Eingabefeld vom Typ «submit» ist eine Taste mit der Funktion Formular an den Server übergeben.

```
<form method="GET">
  <p align="left"><input type="text" size="32"
name="PDP,,MW200,b">
  <input type="submit" value="Merkerwort 200"></p>
</form>
```

Damit wird ein Wert an den Server gesendet und ein Media geschrieben (force), es ist natürlich möglich gleichzeitig mehrere Media zu schreiben, dazu sind einfach soviel Eingabefelder (wie in Schritt 2 beschrieben) anzulegen, wie benötigt werden.

A.3.2 Passwort eingeben

Wie bereits beschrieben kann der PCD Web-Server mit Passwörtern geschützt werden. Zum Festlegen der Passwörter gibt es eine spezielle Seite in der PCD, diese heisst «pwdform.htm». Aus bestimmten Gründen möchten Sie aber eine eigene Seite mit einer anderen Darstellung anlegen. Hierzu der benötigte Code:

- 1) Wieder ein Formular anlegen, aber diesmal die Methode «POST» angeben.
- 2) Dann das Eingabefeld, in unserem Fall ist es vom Typ «PASSWORD», damit werden die eingegebenen Zeichen «verborgen», wenn das nicht erwünscht ist wählen Sie den Typ «text». **Wichtig ist der Name «pwd».**
- 3) Eine Taste wird benötigt, also ein Eingabefeld vom Typ «submit».
- 4) Die Rücksetz-Funktion ist optional, sie dient zum Rücksetzen des Formulars, falls bei der Eingabe des Passwortes ein Fehler gemacht wurde.

```
<FORM ACTION="login.cgi" METHOD="POST">
<P><INPUT TYPE="PASSWORD" NAME="pwd" SIZE="25"></P>
<INPUT TYPE="SUBMIT" VALUE="Submit">
<INPUT TYPE="RESET" VALUE="Reset">
</FORM>
```

A.4 Dynamische Daten in den Html Seiten auffrischen

Wie schon beschrieben, ist es möglich Werte eines PCD Media in einer HTML Seite darzustellen, wenn es mit einem Tag versehen ist. Dieses Tag wird durch den Media Wert ersetzt, sobald die Web-Server PCD die Datei zum Browser des PCs sendet.

Der Wert in der PCD ist nun geändert, um nun frische Werte zu bekommen, muss die Seite wieder hochgeladen werden. Dies genügt für manche Anwendungen, wenn jedoch Bilder/Grafiken oder umfangreiche Seiten jedesmal geladen werden müssen nur um die Werte wieder aufzufrischen, kann dies unerwünschte Effekte mit sich bringen: die Seite verschwindet bis die nächste angezeigt wird oder das Laden der ganzen Seite, eventuell mit Bildern, nimmt sehr viel Zeit in Anspruch.

Mit einem kleinen Trick können solche Probleme umgangen werden.

Dasselbe Problem kann auftreten, wenn mit einem Formular in einer HTML Seite Werte in der PCD überschrieben werden sollen. Wird das Formular übergeben, wird die HTML Seite neu geladen.

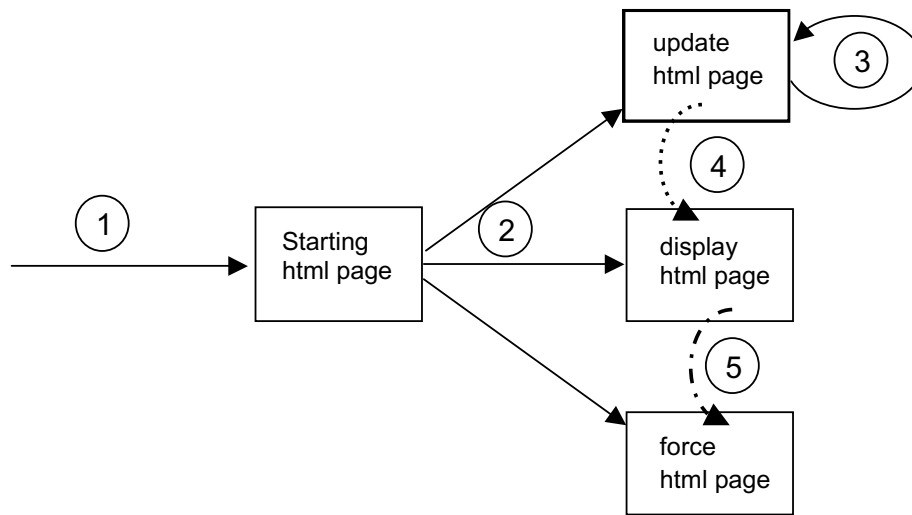


Dieser Trick zum automatischen Auffrischen der Werte ist kein offizielles Leistungsmerkmal des PCD Web-Servers, **er wird nur beispielhaft für die Möglichkeiten gezeigt, die HTML und das Internet bieten.** Dieses Beispiel läuft unter Internet Explorer Version 6. Zukünftige Versionen des Internet Explorers oder andere Browser können Anpassungen nötig machen. **Saia-Burgess Controls ist nicht verant-**

wortlich für jegliche Probleme, die dieses Beispiel in einer Anwendung verursachen könnte.

A.4.1 Konzept

Wir benötigen mehr als nur eine Seite, um all das ausführen zu können, was wir uns vorstellen. Tatsächlich wird dem Anwender nur eine Seite angezeigt, im Hintergrund haben wir jedoch eine weitere Seite, die sobald notwendig Werte auffrischt und in die PCD schreibt. Dies ermöglicht Java Script und die Kommunikation zwischen verschiedenen HTML Seiten. Eine Seite führt die Arbeit aus und kommuniziert das Ergebnis an eine andere Seite.



Damit dieses System mit all seinen Möglichkeiten (auffrischen und überschreiben von Werten) arbeitet benötigen wir 4 HTML Seiten, angezeigt wird jedoch nur eine.

Die Seite **Starting** wird mit **(1)** bezeichnet. Diese Seite macht nichts anderes als die drei anderen Seiten mittels sogenannter Rahmen aufzurufen. Ein Rahmen für jede Seite, der einzige sichtbare Rahmen ist jedoch nur der, der die Seite **display** enthält.

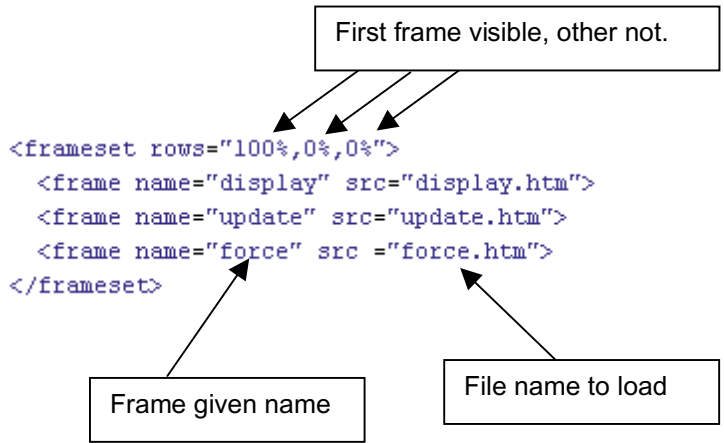
Die Seite **update** enthält alle Tags und wird regelmässig durch eine automatisierte Funktion und ein «Timeout» von der PCD hochgeladen **(3)**. Wenn die Seite geladen ist, ruft sie eine Funktion **(4)** in der Seite **display** auf, die signalisiert, dass neue Werte anstehen. Auf der Seite **display** überschreibt diese Funktion die Werte in jedem Feld, das aufgefrischt werden muss.

Die Seite **force** hat bis jetzt gewartet. Will der Anwender nun einen Wert der PCD ändern, schreibt er diesen Wert in ein Feld in der Seite **display** und/oder betätigt eine Taste. Dies ruft eine Funktion auf, die den Wert **(5)** in die Seite **force** überträgt. Die Werte werden dann in ein Formular in der Seite **force** eingefügt. Dies hat den Vorteil, dass nach dem Übertragen eines neuen Wertes nur die Seite **force** wieder geladen wird, die Seite **display** ist nicht betroffen.

Dies war das Konzept, nun ein konkretes Beispiel, der Projekt Name ist «**Dynamic_update**».

A.4.2 Seite Starting

Wie vorher beschrieben erzeugt diese Seite nur einen Rahmen und öffnet die drei anderen Seiten. Um den Überblick zu behalten, vergeben wir jedem Rahmen einen Namen.



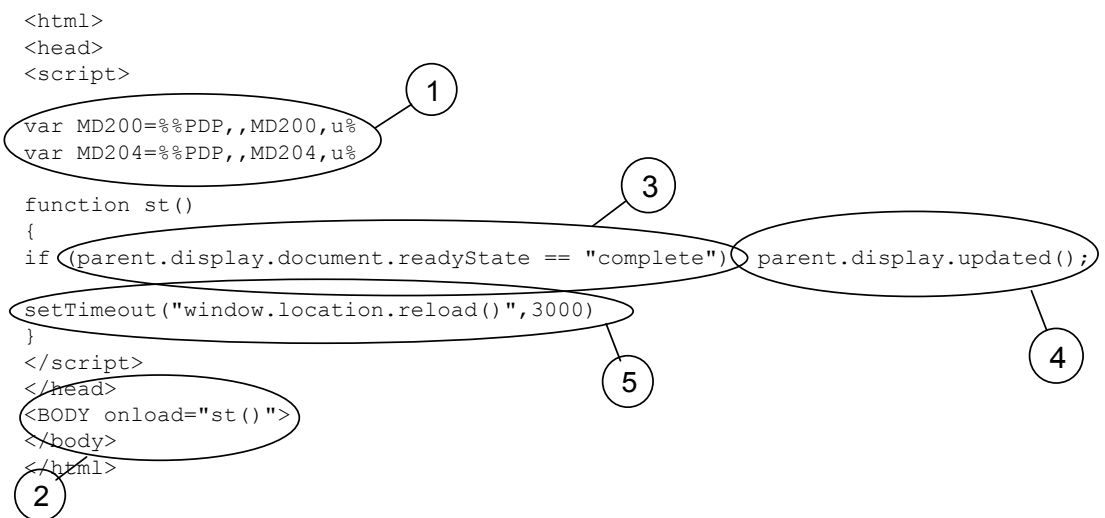
A.4.3 Seite Update



Diese Seite sammelt die PCD Media und frischt sie automatisch auf.

Die PCD Media werden durch ein einfaches Tag direkt (1) in ein Java Script Object gesetzt. So kann dieser Wert einfach verschiedenen Seiten oder Funktionen kommuniziert werden.

Der Wert und das Objekt stehen natürlich nur zur Verfügung, wenn die ganze Seite geladen ist. Wir benutzen das «onload» Ereignis (2), um die Funktion st() auszuführen.



Diese Funktion überprüft, ob die Seite **display** vorhanden ist **(3)**. Wenn diese Bedingung erfüllt ist, erfolgt die Funktion «updated()» **(4)** auf der Seite **display** (Kapitel Seite display). Ist die Funktion ausgeführt, startet ein Timer **(5)**, um die Seite wieder zu laden. Nach Ablauf des Timers, ist die Seite **upload** aus der PCD wieder geladen und alles beginnt von vorn **(1)**.

Variable definieren

Variable oder PCD Media, auf die in der PCD zugegriffen werden, müssen in dieser Seite im Teil **(1)** definiert werden.

Auffrischzeit

Wenn die Auffrischzeit neu definiert werden muss, dann nur hier in der Seite **update** mit der Funktion «setTimeout» im Teil **(5)**. In diesem Beispiel ist sie auf 3000 Millisekunden = 3 Sekunden gesetzt.



Diesen Wert nicht zu klein wählen, da sonst Probleme bei der Synchronisation der HTML Seiten auftreten.

A.4.4 Seite Force

Von dieser Seite werden die Werte zur PCD gesendet, dies um zu verhindern, dass die Seite **display** gerade nach der Übertragung eines Formulars geladen wird. Diese Seite enthält ein Formular **(1)**, dem wir den Namen «F1» geben. In diesem Formular haben wir ein Eingabefeld, dessen Name **(2)** mit dem Tag (PCD Media) den wir dem Wert zuschreiben wollen korrespondiert. Da dieses Eingabefeld nicht angezeigt wird (die Seite **force** ist nicht sichtbar), erklären wir dieses Eingabefeld als «verborgen».

```

<html>
<head>
<script>
function send(t)
{
  document.all.T1.value = t;
  document.F1.submit();
  parent.update.location.reload();
}
</script>
</head>
<BODY>
<form method="GET" name="F1">
<input type="hidden" name="PDP,,MD200,u" id="T1">
</form>
</body>
</html>

```

The code is annotated with circled numbers 1 through 5. Number 1 points to the opening <form> tag. Number 2 points to the name attribute of the hidden input field. Number 3 points to the function definition. Number 4 points to the assignment of the value to the input field. Number 5 points to the reload() call.



Die Seite **display** page ruft die Funktion «send ()» auf und ein Parameter ist der Wert, der zum PCD Media gesendet wird. So wird die Function «send ()» erst aktiv, **(3)**, wenn der empfangene Wert dieses Eingabefeldes zu dieser Seite gehört. Dann wird das Formular dieser Seite übertragen **(4)** und an die PCD gesendet. Damit nun die aufgefrischten neuen Werte am Bildschirm angezeigt werden, veranlassen wir das Laden **(5)** der Seite **update** (dieser Befehl ist optional). Die Funktion wird ausgeführt und die Seite wird automatisch vom Web-Server geladen, danach kann das nächste Überschreiben erfolgen.

A.4.5 Seite Display

Die Seite Display ist die wichtigste, da sie als einzige angezeigt wird. In dieser Seite oben sehen wir einige Java Script Funktionen und den HTML Körper, wo letztlich der Inhalt der Anzeige erstellt wird.

In unserem Beispiel zeigt der Browser folgendes:

- (1)** Ein Eingabefeld und zwei Tasten mit denen der Wert im Eingabefeld erhöht oder vermindert werden kann. Der Wert wird jedoch nur erhöht, wenn der Wert in der PCD ändert. So ist man sicher, dass der Wert erhöht oder vermindert wurde.
- (2)** Ein einfaches Textfeld, das einen Wert anzeigt und ständig aufgefrischt wird.

2 Functionen in Java Script:

- (3)** Die Funktion «change()» wird aufgerufen, wenn eine Taste betätigt wird und der Wert in der PCD geändert hat. Ein Parameter ist das Feld mit dem Wert, der andere Parameter ist der Wert, der addiert oder subtrahiert werden muss. Dann führt die Funktion die Kalkulation aus und ruft die Function in der Seite **force** zum übergeben des Wertes an die PCD auf.

A

Example of dynamic data update and force value

+

-

1

Here; display the value of the Marker DWord 204, update every 3 seconds

2

!!! Don't forget to have the PCD in RUN to see the value moving !!!

- (4)** Diese Funktion wird von der Seite **update** aufgerufen, wenn der neue Wert von der PCD kommt. Mit «SOE» wird ein Test durchgeführt, um zu sehen, ob der Anwender gerade das Eingabefeld editiert, denn in diesem Fall soll der eingegebene Wert nicht überschrieben werden. Danach wird er Wert von der Seite **update** und setzen ihn in das Eingabefeld «T1» und den zweiten Wert in «T2».

```
<script language="JavaScript"><!--
```

```
var SOE = 0
```

```
function change(t,i)
{
  i = Number(t.value) + i;
  parent.force.send(i);
}
```

3

```
function updated()
{
  if (SOE==0) document.all.T1.value = parent.update.MD200;
  document.all.T2.value = parent.update.MD204;
}
// --></script>
```

4

Nun werden in der Fläche der HTML Seite das Textfeld und die Tasten wie im untenstehenden Beispiel angelegt.

Das Eingabefeld **(5)** wird mit dem Schlüsselwort «input» und dem Typ «text» angelegt und dann mit der id versehen, id=«T1». In diesem Beispiel wird auch erkannt, wenn der Anwender den Cursor in das Feld führt und wenn der Cursor das Feld verlässt, diese Erkennung ist optional und erfolgt durch onFocus und onBlur. Die beiden Tasten sind einerseits zum erhöhen **(6)** und andererseits zum vermindern **(7)**. Das Schlüsselwort ist ebenfalls «input», aber der Typ ist «button», benannt sind sie mit «B1» und «B2». «onclick» wird generiert, wenn die Taste betätigt wird. In diesem Fall wird die vorher beschriebene Funktion ausgeführt und das Textfeld in dem der Wert geändert hat, wird ausgegeben sowie der Wert der addiert oder subtrahiert werden soll.

A

```
<td rowspan="2" width="50%"><input type="text" size="20" id="T1" onBlur="SOE=0" onFocus="SOE=1" >
```

5

```
<td width="50%"><input type="button" name="B1" value=" + " onclick="change(T1,+1)">
```

6

```
<td width="50%"><input type="button" name="B2" value=" - " onclick="change(T1,-1)"></td>
```

7

A.5 Allgemeine Grundlagen zur Web-Server Technologie

A.5.1 Allgemeine Hinweise zu Web-Servern

Die Web-Server-Technik basiert darauf, dass Web-Informationen (Web = Gesamtheit verschiedener, verbundener HTML-Seiten und dazugehöriger Dateien, so dass eine Menü-Struktur entsteht) mittels standardisierter SW-Tools (Web-Browsern) dargestellt werden.

Dabei können die darzustellenden Web-Informationen direkt auf dem PC gespeichert sein auf welchem sich der Web-Browser befindet.

Falls dagegen die Web-Informationen auf einem entfernt gelegenen PC gespeichert sind, so bezeichnet man diesen PC als Web-Server welcher nur zur Speicherung der Web-Informationen dient und die angeforderten Web-Informationen dem Web-Browser übermittelt.

Die Übermittlung der Web-Informationen zwischen dem Web-Server PC zum Web-Browser PC kann über eine beliebige PC-Schnittstelle erfolgen (Serielle Schnittstelle, Modem, LAN, Internet...) und normalerweise mit dem TCP/IP Protokoll.

Bedingt durch die beschränkten Übertragungskapazitäten z.B. im Internet und dem Wunsch nach schnellen Ladezeiten sind die Datenstrukturen dieser Web-Informationen sehr effizient aufgebaut und benötigen wenig Speicherplatz auf dem Web-Server.

Der Umfang einer üblichen HTML-Seite liegt im Bereich weniger kByte; Bild-Dateien werden mit leistungsfähigen Algorithmen komprimiert, ausführbare Programme wie z.B. Java-Applets werden interpretativ im Browser (auf dem PC) abgearbeitet.

Da auch in den Steuerungen industrieller Anwendungen der Speicherplatz limitiert ist, eignet sich die Web-Server-Technik ideal auch für diesen Bereich um z.B. Bilder und Dateien zu Bedienung- & Beobachtungsoberflächen direkt auf dem industriellen Steuergerät zu speichern.

Die kleinen Datenmengen belasten die CPU-Leistung einer PCD nur geringfügig und erlauben dadurch, dass der grösste Teil der CPU-Leistung für Steuerungszwecke verwendet werden kann.

Die B&B Informationen sind so immer dezentral, direkt auf dem Steuergerät abgespeichert. Alle Anwender, die nun auf diese B&B Informationen zugreifen, erhalten unabhängig vom Gerät, mit welchem sie arbeiten immer die gleichen, aktuellen Informationen.

Dadurch entfallen die aufwendigen, bei bisherigen auf dem PC abgespeicherten B&B Lösungen notwendigen, Synchronisationen der B&B Informationen bei einem allfälligen Änderungen.

Hinzu kommt, dass die Web-Informationen mit bekannten Microsoft-Werkzeugen wie Word oder FrontPage erstellt werden können.

Die Bedienung der Web-Informationen erfolgt mit Standard-Browsern, deren Handhabung allgemein bekannt und intuitiv ausgelegt ist.

A.5.2 TCP/IP-Protokoll

TCP/IP ist der kleinste gemeinsame Nenner des gesamten Datenverkehrs im Internet. Erst durch dieses Protokoll wurde, historisch gesehen, aus einem begrenzten Netz ein Netz der Netze. Egal, ob Sie WWW-Seiten aufrufen, E-Mails versenden,

FTP Dateien herunterladen oder mit Telnet auf einem entfernten Rechner arbeiten: stets werden die Daten auf gleiche Weise adressiert und transportiert. TCP bedeutet **Transmission Control Protocol** (Protokoll für Übertragungskontrolle), IP bedeutet **Internet Protokoll**.

Wenn Sie eine E-Mail verschicken oder eine HTML-Datei im WWW aufrufen, werden die Daten bei der Übertragung im Netz in kleine Pakete zerstückelt. Jedes Paket enthält eine Angabe dazu, an welche Adresse es geschickt werden soll und den Platz innerhalb der Übertragungsabfolge. Die Adressierung übernimmt das IP. Dazu existiert ein Adressierungsplan, mit den sogenannten IP-Adressen.

Dass die Datenpakete auch wirklich beim Empfänger ankommen, und zwar in der richtigen Reihenfolge, dafür sorgt das TCP. Das TCP verwendet Sequenznummern für die einzelnen Pakete einer Sendung. Erst wenn alle Pakete einer Sendung vollständig beim Empfänger angekommen sind, gilt die Übertragung der Daten als abgeschlossen.

Jeder Rechner, der am Internet teilnimmt, ist mit einer IP-Adresse im Netz angemeldet. Rechner, die ans Internet angeschlossen sind, werden als **Hosts** oder **Hostrechner** bezeichnet. Wenn Sie also mit Ihrem PC im WWW surfen oder neue E-Mails abholen, sind Sie mit einer IP-Adresse im Internet angemeldet. Ihr Zugangs-Provider, über dessen Hostrechner Sie sich einwählen, kann feste IP-Adressen für Sie einrichten. Grosse Zugangs-Provider, wie CompuServe oder AOL, vergeben unpersönliche, dynamische IP-Adressen für jede Internet-Einwahl. Damit ein Rechner am Internet teilnehmen kann, muss er über eine Software verfügen, die das TCP/IP-Protokoll unterstützt. Unter MS Windows ist dies beispielsweise die Datei **winsoc.dll** im Windows-Verzeichnis.

A

A.5.3 IP Adressierung

Eine typische IP-Adresse sieht in Dezimalschreibweise so aus: 149.174.211.5 - vier Zahlengruppen also, getrennt durch Punkte. Die Punkte haben die Aufgabe, über- und untergeordnete Netze anzusprechen. So wie zu einer Telefonnummer im weltweiten Telefonnetz eine Landeskennzahl, eine Ortsnetzkenzahl, eine Teilnehmer-Nummer und manchmal auch noch eine Durchwahlnummer gehört, gibt es auch im Internet eine Vorwahl - die Netzwerknummer, und eine Durchwahl - die Hostnummer.

Der erste Teil einer IP-Adresse ist die Netzwerknummer, der zweite Teil die Hostnummer. Wo die Grenze zwischen Netzwerk- und Hostnummer liegt, bestimmt ein Klassifizierungsschema für Netztypen. Die folgende Tabelle verdeutlicht dieses Schema. In den Spalten 'IP - Adressierung' und 'Typische IP-Adresse' ist die Netzwerknummer (der Vorwahlteil) fett dargestellt. Der Rest der IP-Adresse ist die Hostnummer eines Rechners innerhalb dieses Netzes.

Netztyp	IP - Adressierung	Typische IP - Adresse
Klasse - A- Netzwerk	xxx .xxx.xxx.xxx	103 .234.123.87
Klasse - B- Netzwerk	xxx.xxx .xxx.xxx	151.170 .102.15
Klasse - C- Netzwerk	xxx.xxx.xxx .xxx	196.23.155 .113

Klasse A Netzwerke stellen die höchste Stufe in der Netzhierarchie dar. Nur die erste Zifferngruppe wird zur Adressierung des Netzwerks benötigt. Die anderen Zifferngruppen sind Hostnummern innerhalb des Netzwerks. Bei Netzwerknummern solcher Netze sind Adressen zwischen 1 und 126 möglich, d.h. es gibt weltweit nur

126 Klasse-A-Netze geben. Eine IP-Adresse, die zu einem Klasse-A-Netz gehört, ist also daran erkennbar, dass die erste Zifferngruppe zwischen 1 und 126 liegt. Das amerikanische Militärnetz ist beispielsweise solch ein Klasse-A-Netz. Innerhalb eines Klasse-A-Netzes kann der entsprechende Netzbetreiber die zweite, dritte und vierte Zifferngruppen der einzelnen IP-Adressen seiner Netzteilnehmer frei vergeben. Da alle drei Zifferngruppen Werte von 0 bis 255 haben können, kann ein Klasse-A-Netzbetreiber also bis zu 16,7 Millionen IP-Adressen für Host-Rechner innerhalb seines Netzes vergeben.

Die zweithöchste Hierarchiestufe sind die **Klasse-B-Netze**. Die Netzwerknummer dieser Netze benötigt die beiden ersten Zifferngruppen der Adressierung. In der ersten Zifferngruppe können Klasse-B-Netze die Werte zwischen 128 und 192 benutzen. Eine IP-Adresse, die zu einem Klasse-B-Netz gehört, ist also daran zu erkennen, dass die erste Zifferngruppe zwischen 128 und 192 liegt. Bei der zweiten sind Werte zwischen 0 und 255 erlaubt. Dadurch sind etwa 16.000 Netze adressierbar. Da die Zifferngruppen drei und vier ebenfalls zwischen 0 und 255 liegen dürfen, können an jedes Klasse-B-Netz bis zu 65'000 Hostrechner angeschlossen werden. Klasse-B-Netze werden vor allem an grosse Firmen, Universitäten und Online-Dienste vergeben.

In der untersten Hierarchiestufe befinden sich die **Klasse-C-Netze**. Für die Adressierung werden die ersten drei Zifferngruppen benötigt, wobei die erste Zifferngruppe zwischen 192 und 223 liegen muss. Die Zifferngruppen zwei und drei gehören ebenfalls noch zur Netzwerknummer. Über zwei Millionen solcher Netze sind dadurch adressierbar. Vor allem an kleine und mittlere Unternehmen mit direkter Internet-Verbindung, aber auch an kleinere Internet-Provider, werden diese Adressen vergeben. Da nur noch eine Zifferngruppe mit Werten zwischen 0 und 255 übrig bleibt, können in einem C-Netz maximal 255 Host-Rechner adressiert werden.

Dieses Adressierungs-Schema wird den Anforderungen der Zukunft kaum noch gerecht. Es gibt jedoch bereits Ideen zur Neustrukturierung des World Wide Web.

A.5.4 Client-Server-Technologie

Für die einzelnen Internet-Dienste wie World Wide Web, Gopher, E-Mail, FTP usw. muss auf einem Hostrechner, der anderen Rechnern diese Dienste anbietet, eine entsprechende **Server**-Software laufen. Ein Hostrechner kann einen Internet-Dienst nur anbieten, wenn eine entsprechende Server-Software auf dem Rechner aktiv ist, und wenn mit dem Rechner eine Verbindung besteht.

Server sind Programme, die permanent darauf warten, dass eine Anfrage eintrifft, die ihren Dienst betreffen. So wartet etwa ein WWW-Server darauf, dass Anfragen eintreffen, die WWW-Seiten auf dem Server-Rechner abrufen wollen.

Clients sind dagegen Software-Programme, die typischerweise Daten von Servern anfordern. Ein WWW-Browser ist beispielsweise ein Client. Wenn man z.B. auf einen Verweis klickt, der zu einer HTTP-Adresse führt, startet der Browser, also der WWW-Client, eine Anfrage an den entsprechenden Server auf dem entfernten Hostrechner. Der Server wertet die Anfrage aus und sendet die gewünschten Daten. Um die Kommunikation zwischen Clients und Servern zu regeln, gibt es entsprechende Protokolle. Die Client-Server-Kommunikation im WWW regelt das HTTP-Protokoll. Dieses Protokoll läuft über dem TCP/IP-Protokoll.

Dass ein Client Daten anfordert und ein Server die Daten sendet, ist der Normalfall.



Es gibt jedoch auch «Ausnahmen». So kann ein Client nicht nur Daten anfordern, sondern auch Daten an einen Server schicken: zum Beispiel, wenn Sie per FTP eine Datei auf den Server-Rechner hochladen, wenn Sie eine E-Mail versenden oder im WWW ein Formular ausfüllen und abschicken. In diesen Fällen spricht man auch von **Client-Push** («Client drängt dem Server Daten auf»).

Eine andere Ausnahme ist, wenn der Server zuerst aktiv wird und dem Client etwas ohne dessen Anforderung zuschickt. Dies nennt man **Server-Push** («Server drängt dem Client Daten auf»). Neue Technologien wollen diese Ausnahme zur Regel erheben: die sogenannten **Push-Technologies**. Diese Technologien sollen ermöglichen, dass ein Client regelmässig Daten empfangen kann, ohne diese eigens anzufordern. Dadurch werden Broadcasting-Dienste wie aktuelle Nachrichten usw. möglich. Netscape Navigator und Microsoft Internet Explorer (beide ab Version 4.0) haben entsprechende Schnittstellen, um solche Dienste zu nutzen.

A.5.5 DNS - Domain Name Service

Computer können mit Zahlen besser umgehen, Menschen in der Regel besser mit Namen. Deshalb hat man ein System erdacht, das die numerischen IP-Adressen für die Endanwender in anschauliche Namensadressen übersetzt.

Dieses System ist ähnlich wie das IP-Adressensystem hierarchisch aufgebaut. Eine Namensadresse in diesem System gehört zur sogenannten **Top-Level-Domain** und innerhalb dieser zu einer **Sub-Level-Domain**. Jede Sub-Level-Domain kann nochmals untergeordnete Domains enthalten, muss aber nicht. Die einzelnen Teile solcher Namensadressen sind wie bei IP-Adressen durch Punkte voneinander getrennt. Eine solche Namensadresse ist beispielsweise **teamone.de**.

Top-Level-Domains stehen in einem Domain-Namen an letzter Stelle. Es handelt sich um einigermaßen sprechende Abkürzungen. Die Abkürzungen, die solche Top-Level-Domains kennzeichnen, sind entweder Landeskennungen oder Typenkennungen. Beispiele sind:

de = Deutschland
au = Österreich
ch = Schweiz
it = Italien
my = Malaysia
com = Kommerziell orientierte Namensinhaber
org = Organisationen
net = Allgemeines Netz
edu = amerikanische Hochschulen
gov = amerikanische Behörden
mil = amerikanische Militärbehörden

Jede dieser Top-Level-Domains stellt einen Verwaltungsbereich dar, für die es auch eine «Verwaltungsbehörde» gibt, die für die Namensvergabe von Sub-Level-Domains innerhalb ihres Verwaltungsbereichs zuständig ist. Wenn Sie beispielsweise einen Domain-Namen wie **MeineFirma.de** beantragen wollen, muss der Antrag dafür an das DENIC (Deutsches Network Information Center) gestellt werden. Kommerzielle Provider erledigen das für Sie, wenn Sie dort einen entsprechenden Service in Anspruch nehmen. Ihren Wunschnamen erhalten Sie jedoch nur, wenn diese Namensadresse noch nicht anderweitig vergeben wurde. Schlaufüchse sind daher auf die Idee gekommen, Namen grosser Firmen, die noch keine eigene Domain beantragt



haben, für sich zu reservieren, um sie dann, wenn diese Firma diesen Namen beantragt, teuer an diese weiterzuverkaufen. Mittlerweile sind solchen Machenschaften Riegel vorgeschoben. Dennoch kommt es immer wieder zu Rechtsstreitereien wegen attraktiven Domain-Namen. Wenn etwa zwei zufällig gleichnamige Firmen, die sonst nichts miteinander zu tun haben, den gleichen Domain-Namen reservieren lassen wollen, kann nur eine der Firmen den Zuschlag erhalten. Um Streitigkeiten dieser Art zu reduzieren, gibt es mittlerweile auch neue Top-Level-Endungen.

Inhaber von zweiteiligen Domain-Namen können nochmals Sub-Level-Domains vergeben. So gibt es beispielsweise eine Domain namens **seite.net**. Die Betreiber dieser Domain haben nochmals Sub-Domains vergeben, wodurch Domain-Adressen wie **java.seite.net** oder **javascript.seite.net** entstanden.

A.5.6 Routing und gateways

Im Internet ist es zunächst nur innerhalb des eigenen Sub-Netzes möglich, Daten von einer IP-Adresse zu einer anderen zu schicken. In allen anderen Fällen, wenn Daten an andere Netze geschickt werden sollen, treten Rechner auf den Plan, die den Verkehr zwischen den Netzen regeln. Solche Rechner werden als **Gateways** bezeichnet. Diese Rechner leiten Daten von Hostrechnern aus dem eigenen Sub-Netz an Gateways in anderen Sub-Netzen weiter und ankommende Daten von Gateways anderer Sub-Netze an die darin adressierten Host-Rechner im eigenen Sub-Netz. Ohne Gateways gäbe es kein Internet.

Das Weiterleiten von Daten zwischen Sub-Netzen wird als **Routing** bezeichnet. Die Beschreibung der möglichen Routen vom eigenen Netzwerk Gateway-Rechner zu anderen Netzwerken sind in **Routing-Tabellen** auf den Gateway-Rechnern festgehalten.

Zu den Aufgaben des Gateways gehört es auch, Alternativ-Routen zu finden, wenn die übliche Route nicht funktioniert, etwa, weil eine Störung oder ein Datenstau aufgetreten ist. Gateways senden sich ständig Testpakete zu, um das Funktionieren der Verbindung zu testen und für Datentransfers «verkehrsarme» Wege zu finden.

Wenn also im Internet ein Datentransfer stattfindet, ist keinesfalls von vorneherein klar, welchen Weg die Daten nehmen werden. Sogar einzelne Pakete einer einzigen Sendung können völlig unterschiedliche Wege nehmen. Wird beispielsweise von Deutschland aus eine WWW-Seite aufgerufen, die auf einem Rechner in den USA liegt, kann es sein, dass die Hälfte der Seite über den Atlantik kommt und die andere über den Pazifik, bevor der WWW-Browser diese anzeigen kann.

A.6 Allgemeine Grundlagen HTML

A.6.1 “Markup” Elemente

HTML bedeutet **HyperText Markup Language**. Es handelt sich dabei um eine Sprache, die mit Hilfe von SGML (Standard Generalized Markup Language) definiert wird. SGML ist als ISO-Norm 8879 festgeschrieben.

HTML ist eine sogenannte Auszeichnungssprache (Markup Language). Sie hat die Aufgabe, die logischen Bestandteile eines Dokuments zu beschreiben. Als Auszeichnungssprache enthält HTML daher Befehle zum Markieren typischer Elemente eines Dokuments, wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen.

Das Beschreibungsschema von HTML geht von einer hierarchischen Gliederung aus. HTML beschreibt Dokumente. Dokumente haben globale Eigenschaften wie zum Beispiel einen Titel oder eine Hintergrundfarbe. Der eigentliche Inhalt besteht aus Elementen, zum Beispiel einer Haupt-Überschrift. Einige dieser Elemente haben wiederum Unterelemente. So enthält ein Textabsatz zum Beispiel eine fett markierte Textstelle, eine Liste besteht aus einzelnen Listenpunkten, und eine Tabelle gliedert sich in einzelne Tabellenzellen.

Die meisten dieser Elemente haben einen fest definierten Erstreckungsraum. So geht eine Überschrift vom ersten bis zum letzten Zeichen, eine Liste vom ersten bis zum letzten Listenpunkt, oder eine Tabelle von der ersten bis zur letzten Zelle. Auszeichnungen markieren Anfang und Ende von Elementen. Um z.B. eine Überschrift auszuzeichnen, lautet das Schema:

[Überschrift] <i>Text der Überschrift</i> [Ende Überschrift]
--

Bei einem Element, das Unterelemente besitzt, etwa einer Liste, lässt sich das gleiche Schema anwenden:

[Liste] [Listenpunkt] <i>Text des Listenpunkts</i> [Ende Listenpunkt] [Listenpunkt] <i>Text des Listenpunkts</i> [Ende Listenpunkt] [Ende Liste]



WWW-Browser, die HTML-Dateien am Bildschirm anzeigen, lösen diese Auszeichnungsbefehle auf und stellen die Elemente dann in optisch gut erkennbarer Form am Bildschirm dar. Dabei ist die Bildschirmdarstellung aber nicht die einzige denkbare Ausgabeform. HTML kann beispielsweise genauso gut mit Hilfe synthetisch erzeugter Stimmen auf Audio-Systemen ausgegeben werden.

A.6.2 Verknüpfen mit Hyperlinks

Eine der wichtigsten Eigenschaften von HTML ist die Möglichkeit, Verweise zu definieren. Verweise («Hyperlinks») können zu anderen Stellen im gleichen Projekt führen, oder aber zu beliebigen anderen Adressen im World Wide Web und sogar zu Internet-Adressen, die nicht Teil des WWW sind.

Durch diese einfache Grundeigenschaft eröffnet HTML völlig neue Welten. Das Bewegen zwischen räumlich weit entfernten Rechnern wird bei modernen grafischen WWW-Browsern auf einen Mausklick reduziert. In den eigenen HTML-Dateien können Sie Verweise notieren und dadurch inhaltliche Verknüpfungen zwischen eigenen Inhalten und denen anderer Anbieter herstellen. Auf dieser Grundidee beruht letztlich das gesamte World Wide Web, und dieser Grundidee verdankt es seinen Namen.

A.6.3 Software-unabhängiger Klartext

HTML ist ein sogenanntes Klartext-Format. HTML-Dateien kann man mit jedem beliebigen Texteditor bearbeiten, der Daten als reine Textdateien speichern kann. Es gibt also keine bestimmte Software, die zum Erstellen von HTML-Dateien nötig ist. Zwar gibt es längst mächtige Programme, die auf das Editieren von HTML spezialisiert sind, doch das ändert nichts an der entscheidenden Eigenschaft: HTML ist nicht an irgendein bestimmtes, kommerzielles Software-Produkt gebunden. Diese wichtige, vielleicht sogar wichtigste Eigenschaft von HTML sollten Sie immer im Auge behalten, wenn man Ihnen erzählen will, dass Web-Publishing nur mit bestimmten Software-

Produkten möglich sei.

Die Klartext-Befehle von HTML sind für Maschinen **und** Menschen gedacht. Wer keine sinnlosen Vorurteile gegenüber sichtbaren Befehlen am Bildschirm hat, wird in HTML eine überraschend einfache Befehlssprache finden. Die Sprache ist Englisch, doch da die Anzahl der Befehle begrenzt ist, ist es auch ohne tiefere Kenntnisse der englischen Sprache möglich, sich in HTML einzuarbeiten.

Da HTML ein Klartextformat ist, lässt es sich auch hervorragend mit Hilfe von Programmen generieren. Von dieser Möglichkeit machen beispielsweise CGI-Programme Gebrauch. Wenn Sie im WWW zum Beispiel einen Suchdienst benutzen und nach einer Suchanfrage die Ergebnisse präsentiert bekommen, dann ist das, was Sie am Bildschirm sehen, HTML-Code, der von einem Programm generiert wurde.

A.6.4 Universelle Anwendung

HTML ist eigentlich als Auszeichnungssprache zum Erstellen von WWW-Seiten gedacht. HTML-Dateien funktionieren aber nicht nur im WWW. Es ist kein Problem, eine HTML-Datei lokal auf jedem Rechner mit einem WWW-Browser zu öffnen. HTML-Dateien sind deshalb auch ideal geeignet für lokale Dokumentationen, für CD-ROM-Oberflächen, für Readme-Dateien usw. Mit HTML und den ergänzenden Sprachen CSS und JavaScript, die ebenfalls lokal funktionieren, lassen sich auch anspruchsvolle Projekte realisieren, die nicht für den Einsatz im WWW gedacht sind. Egal ob ein Tagebuch fürs nächste Jahrtausend herzustellen ist, ob der nächsten SW-Version eine HTML-basierte Online-Hilfe beizugeben ist, oder ob eine informative CD produziert werden soll - HTML ist längst das verbreitetste Dateiformat der Welt. HTML-Dateien laufen auf jedem Rechner, auf dem ein WWW-Browser installiert ist - und Rechner, auf denen kein WWW-Browser verfügbar ist, dürfen mittlerweile als veraltet bezeichnet werden.

A

A.6.5 Formatierung für HTML-Elemente

HTML ist eine «logische» Sprache. HTML definiert den grundsätzlichen Aufbau von WWW-Seiten, Elemente, Strukturen, Verweise, verknüpfte Elemente wie Grafiken, Multimedia usw. HTML ist jedoch nicht dazu gedacht, anzugeben, wie ein Element genau aussehen soll. So kann man in HTML eine Überschrift definieren und dabei angeben, dass es sich um eine Überschrift 1. Ordnung handeln soll. Man kann aber innerhalb von HTML nicht bestimmen, wie gross, in welcher Schriftart usw. die Überschrift dargestellt werden soll. Das übernimmt der WWW-Browser beim Anzeigen. Er benutzt dabei eine Mischung aus Grundeinstellungen, die der Anwender vornehmen kann und fest programmierten Darstellungen einzelner HTML-Elemente.

In den Zeiten des ersten großen Erfolgs von HTML wurde die Sprache dann für allerlei physische Formatierungen «missbraucht». So tauchte plötzlich ein HTML-Befehl zur Darstellung von blinkendem Text auf und später wurde ein Befehl beliebt, mit dessen Hilfe sich die Schriftart, die Schriftgrösse (in 7 relativen Stufen) und die Schriftfarbe bestimmen liessen. Doch all diese Implementierungen in HTML sind letztendlich inkonsequent.

An diesem Punkt setzen die Cascading Style Sheets (CSS) auf. Es handelt sich dabei um eine Ergänzungssprache, die für HTML entwickelt wurde. Sie klinkt sich nahtlos in HTML ein und erlaubt das beliebige Formatieren einzelner HTML-Elemente. Mit Hilfe von CSS kann man beispielsweise festlegen, dass alle Überschriften 1. Ordnung 24 Punkt gross, in roter Helvetica-Schrift, mit einem Nachabstand von 16

Punkt und mit einem grünen doppelten Rahmen dargestellt werden sollen. Genau so gut kann man auch für einen beliebigen Text festlegen, dass nur dieser Text 3 Zentimeter gross sein soll und einen gelben Hintergrund erhält. Daneben enthält die CSS-Sprache auch Befehle zum punktgenauen Platzieren von Elementen am Bildschirm und für andere Ausgabemedien wie Druck oder Audio-Systeme. Durch CSS erfährt HTML einen gewaltigen Vorwärtsschub.

Die CSS-Sprache ist genauso wie HTML eine offiziell normierte Sprache. Sie wird wie HTML vom W3-Konsortium gepflegt und weiterentwickelt. Mittlerweile interpretieren die neueren Browser-Versionen von Netscape und Microsoft die Style-Sheets ebenfalls.

A.6.6 Eine Programmiersprache für WWW-Seiten

In HTML können unter anderem auch Formulare definiert werden. Diese Formulare können Eingabefelder, Listen, Tasten usw. enthalten. Ein Benutzer kann dieses Formular ausfüllen und über das WWW absenden. Dies ist für viele Zwecke eine nützliche Erfindung. HTML erlaubt jedoch dem Formularanbieter beispielsweise nicht, die Eingaben des Benutzers vor dem Absenden auf Vollständigkeit und Plausibilität zu prüfen.

Oder ein anderer Fall: Man kann in HTML zwar Dateien einbinden, weiss aber nicht, ob der Anwender einen Browser besitzt, der diese anzeigen kann. Es ist praktisch, den Befehl zum Einbinden der Datei davon abhängig zu machen, ob der Browser des Anwenders diese anzeigen kann oder nicht.

Für diese und unzählige andere nützliche, allerdings auch weniger nützliche Zwecke erfand Netscape eine Programmiersprache namens JavaScript. JavaScript-Befehle können direkt innerhalb von HTML-Dateien oder als separate Datei eingebunden werden.

JavaScript gewinnt auf modernen WWW-Seiten immer mehr an Bedeutung. Neuere Entwicklungen in JavaScript sind vor allem verantwortlich für das, was unter dem Schlagwort Dynamic HTML bekannt geworden ist. JavaScript erlaubt den nachträglichen Zugriff auf alle Elemente einer HTML-Datei. So werden faszinierende neue Effekte möglich, die normalerweise mit HTML nicht möglich sind. Ein Text kann etwa per Mausklick durch einen anderen ersetzt werden, Elemente können auf ein Anwenderereignis hin oder auch zeitgesteuert automatisch verschwinden, eingeblendet und am Bildschirm verschoben werden.

Beim Erstellen von anspruchsvollen WWW-Seiten, kommt man zumindest um einfache JavaScripts kaum mehr herum. Aus diesem Grund enthält SELFHTML eine ausführliche Dokumentation zu JavaScript. Mit den Grundlagen zu JavaScript sollte man sich neben HTML auf jeden Fall beschäftigen.

A.6.7 JavaScript und HTML

JavaScript ist kein Bestandteil von HTML, sondern eine eigene Programmiersprache. Diese Sprache wurde jedoch eigens zu dem Zweck geschaffen, HTML-Autoren ein Werkzeug in die Hand zu geben, mit dessen Hilfe sich WWW-Seiten optimieren lassen.

JavaScript-Programme werden wahlweise direkt in der HTML-Datei oder in separaten Dateien eingebettet. Sie werden nicht - wie Java-Programme - kompiliert, sondern als



Quelltext während Laufzeit interpretiert, also ähnlich wie Batchdateien bzw. Shell-scripts. Dazu besitzen moderne WWW-Browser wie Netscape Navigator oder Microsoft Internet Explorer entsprechende Interpreter-Software.

In einer Programmiersprache wie JavaScript gibt es für Anfänger viele verwirrende Elemente: Sonderzeichen, Variablen, Wenn-Dann-Anweisungen, Schleifen, Funktionen, Methoden, Parameter, Objekte, Eigenschaften und anderes mehr. Um mit diesen Elementen richtig umzugehen, muss man lernen sich vorzustellen, was im Computer passiert, wenn der Programmcode abläuft. Das ist ein Lernprozess, den Anfänger nur durch viel Übung bewältigen. JavaScript ist dazu allerdings hervorragend geeignet, da sie eine vergleichsweise einfache Sprache ist, ohne die vielen Arbeitsbereiche einer «grossen» Programmiersprache, wie z.B. Arbeitsspeicherverwaltung oder Dateioperationen. Ausserdem setzt JavaScript auf einer bestimmten Umgebung auf, nämlich auf einer anzuzeigenden oder einer angezeigten WWW-Seite.

A.6.8 JavaScript, JScript, ECMA-262, Sprachversionen

JavaScript ist eine Programmiersprache, die von Netscape eingeführt wurde und von Netscape lizenziert ist. Deshalb beherrschen die jeweils neuen Versionen des Netscape-Browsers auch immer etwas mehr JavaScript als die Konkurrenz, die abwarten muss, was Netscape an Neuerungen implementiert.

Netscape Navigator 2.0 interpretiert den seinerzeit eingeführten JavaScript-Sprachstandard 1.0. Der MS Internet Explorer versteht diesen Sprachstandard weitgehend erst seit der Version 3.0. Netscape Navigator interpretiert seit Version 3.0 den JavaScript-Standard 1.1. Der MS Internet Explorer interpretiert diesen erst seit der Version 4.0. Netscape Navigator 4.0 interpretiert den JavaScript-Standard 1.2. Einige Befehle dieser Version werden auch vom MS Internet Explorer 4.0 interpretiert.

Weiterhin unterscheiden sich Netscape Navigator und MS Internet Explorer in der Interpretation einzelner Objekteigenschaften. So funktionieren beispielsweise Cookies (ein bekanntes JavaScript-Feature) bei Netscape Navigator auch lokal, während der MS Internet Explorer, zumindest in der Version 3.x, eine HTTP-Verbindung voraussetzt, um Cookies zu unterstützen. Auch bei internen Abläufen, etwa bei vielfach zu durchlaufenden Schleifen, gibt es Unterschiede zwischen einzelnen Browsern und Browser-Versionen.

Der MS Internet Explorer 4.x interpretiert JavaScript, ausserdem aber vor allem die Microsoft-eigene Variante **JScript**, die spezielle Befehle für die Betriebssystemerweiterungen des MS Internet Explorers bereitstellt.

Sowohl Netscape als auch Microsoft betonen, dass sie sich mit ihren Sprachinterpretieren am Standard für Internet-Scriptsprachen, an **ECMA-262**, orientieren. Das ECMA-Komitee, dem verschiedene Software-Hersteller, unter anderem Microsoft und Netscape, angehören, ist bemüht, auch im Bereich der Scriptsprachen einen allgemeingültigen Standard zu definieren, so wie es bei HTML oder CSS der Fall war.

A.6.9 HTML-Editoren

HTML-Seiten können mit jedem beliebigen HTML-Editor entwickelt werden.

Grundsätzlich kann man bei HTML-Editoren zwischen textbasierten und WYSIWYG-Editoren (WYSIWYG = **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) unterscheiden. Textbasierten Editoren arbeiten direkt mit HTML-Befehlen. Diese Editoren verfügen in der Regel

über eine Werkzeugleiste und diverse Menübefehle, um die HTML-Tags in den Text einzufügen. Die Tags erscheinen sichtbar im Text. WYSIWYG-Editoren bieten ebenfalls Werkzeugleisten und Menübefehle, um das Setzen der HTML-Tags im Dialog zu unterstützen. Der Unterschied ist jedoch, dass die Tags nicht angezeigt werden (bzw. nur auf Wunsch). Stattdessen wird der Text bereits so am Bildschirm angezeigt wie er später in den Web-Browsern dargestellt wird.






Von den meisten der folgenden Adressen können Sie die betreffenden HTML-Editoren direkt herunterladen. Es handelt sich meist um Shareware-Produkte, d.h. Sie dürfen das Programm eine gewisse Zeit testen, müssen es aber kaufen und registrieren, um es endgültig nutzen zu können.

Diese Auswahl erhebt keinen Anspruch auf Vollständigkeit.

1-4-All	http://www.mmssoftware.com/14All/	HTML editor
4W Publisher	http://www.4w.com/4wpublisher/	Datenbanktool, das automatisch Reports in HTML-Form erstellen kann
AOL Press	http://www.aolpress.com/press/index.html	Wysiwyg HTML-Editor
Arachnophilia	http://www.arachnoid.com/arachnophilia/	HTML-Editor
Coffee Cup	http://www.coffeecup.com/editor/	HTML-Editor mit fertigen JavaScripts, Grafiken usw
Claris Homepage	http://www.clarishomepage.com/	Wysiwyg Editor
Frontpage	http://www.microsoft.com/frontpage/	Wysiwyg-HTML-Editor von Microsoft
Horizon Web Text	http://www.gcnet.com/bw/horizon/	32-Bit HTML-Editor
HTMLed (Pro)	http://www.ist.ca/	HTML Editor
HTML Assistant	http://www.exit0.com/ez1/products/pro2000.html	HTML Editor
SchemaText	http://www.schema.de/sitehtml/site-d/schemat5.htm	Profi-Autorensystem für große Web-Projekte
NetObjects	http://www.netobjects.com/	Mächtiges Authoring-Tool
SuperHTML	http://www.superhtml.de/	HTML Editor
Tarantula	http://www.indian-sites.com/nostrumindia/	Wysiwyg Editor
WebMedia Publisher	http://www.wbmedia.com/publisher/	WYSIWYG HTML-Publisher

B Anhang B

B.1 Icons

	Dieses Symbol verweist den Leser innerhalb eines Handbuches auf weiterführende Informationen in diesem oder einem anderen Handbuch, oder in technischen Informationsbroschüren. In der Regel besteht kein direkter Link zu diesen Dokumenten.
	Dieses Symbol warnt den Leser vor dem Risiko elektrischer Entladung durch Berühren. Empfehlung: Bevor Sie in Kontakt mit elektronischen Bauteilen kommen, sollten Sie zumindest vorher den Minuspol des Systems (Gehäuse der PGU-Buchse) berühren. Besser ist es, permanent mit einer Erdungslasche am Handgelenk mit dem Minuspol verbunden zu sein.
	Dieses Zeichen steht neben Anweisungen, die befolgt werden müssen.
	Erklärungen neben diesem Zeichen sind nur für die Saia PCD Classic Serie gültig.
	Erklärungen neben diesem Zeichen sind nur für die Saia PCD xx7 Serie gültig.

B.2 Adresse

Saia-Burgess Controls AG

Bahnhofstrasse 18
CH-3280 Murten | Schweiz

Telefon +41 26 / 672 72 72
Telefax +41 26 / 672 74 43

E-mail: support@saia-pcd.com
Homepage: www.saia-pcd.com
Technischer Support: www.sbc-support.com