

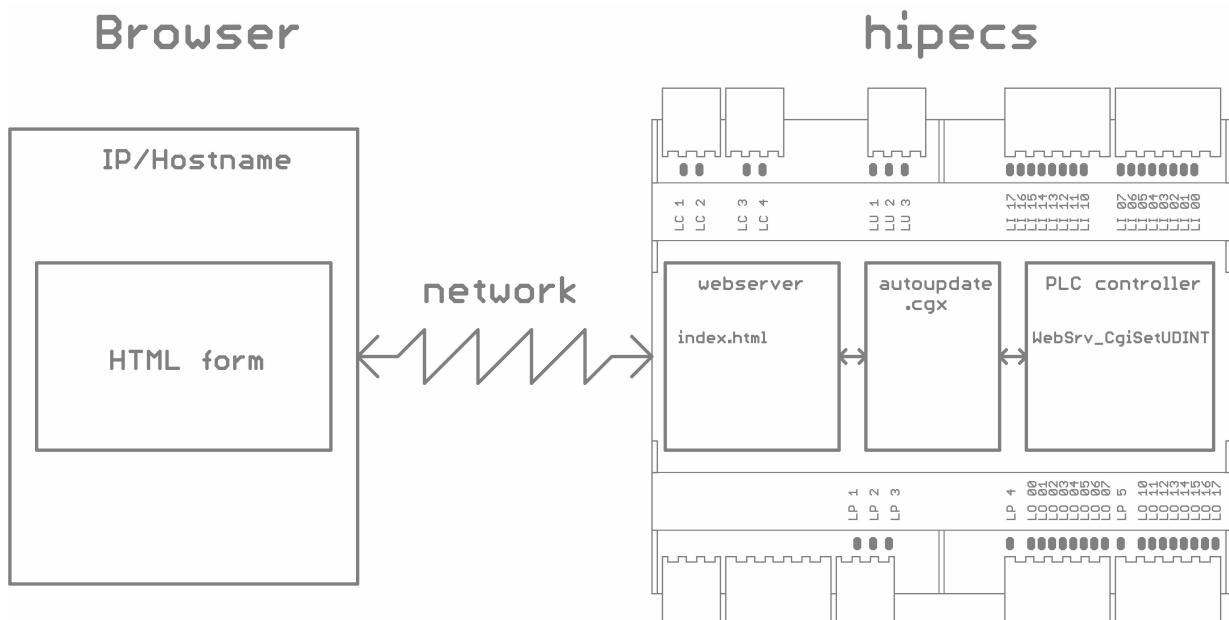
Web Visu Tutorial

Hipecs Web Visu

Die hipecs (high performance controller system) bietet die Möglichkeit einer sog. Web-Visualisierung über den integrierten Webserver. Hierfür wird im Standard Web-Verzeichnis der Steuerung eine index.htm Datei angelegt, welche dann bei Browserzugriff standardmäßig angezeigt wird. Um auch aktuelle Daten aus der SPS-Steuerung anzeigen zu können müssen jedoch einige Anpassungen und Einstellungen vorgenommen werden.

Übersicht

Der in die hipecs integrierte Webserver zeigt die HTML Dateien, welche im Standard-Webverzeichnis gespeichert sind an. Um Daten und Variablen aus der SPS auch im Browser zugänglich zu machen, wird ein sogenanntes HTML-Formular benötigt, über welches die Daten ausgetauscht werden. Zusätzlich wird auch ein Java-Script „xml_http.js“ benötigt, welches für ein selbstständiges aktualisieren der Werte im Web-Browser sorgt, damit immer aktuelle Werte aus der SPS im Browser angezeigt werden. Die Daten des HTML-Formulars werden dabei vom hipecs webserver verarbeitet. Um diese aber mit dem SPS-Laufzeitsystem auszutauschen ist noch ein weiteres CGI-Script nötig. In dieser .cgx Datei wird die Verknüpfung zwischen den Variablen des SPS-Programmes und derer, die im HTML-Code verwendet werden, hergestellt. Um die Daten des Webserver dann auch im CoDeSys zugänglich zu machen, müssen die entsprechenden Funktionen aus der Webserver-Bibliothek verwendet werden. Die Web-Adresse der hipecs ist entweder die entsprechende IP oder der vergebene Hostname, der in der Regel aus „hipecs+[letzte 6 Stellen d. Seriennummer]“ besteht.



Web Visu Tutorial

Einstellungen im CoDeSys-Programm der hipecs

Zur Zeit unterstützt der hipecs Webserver die Datentypen BOOL und UDINT. Um nun die entsprechenden Funktionen für das Handling dieser Daten zugänglich zu machen, muss die Bibliothek **FBESysNet.lib** ins CoDeSys-Projekt eingebunden werden.

- **SysNet_HttpSetDir('A:web')**: Diese Funktion definiert das Standard-Webverzeichnis der hipecs Steuerung. Die HTML Datei index.htm wird nun bei Zugriff auf den Webserver standardmäßig angezeigt. In diesem Beispiel ist „A:\web“ das Standardverzeichnis
- **SysNet_HttpRegCgiFunction(4, INDEXOF(WebSrv_CgiGetUDINT))**: Diese Funktion registriert eine weitere Funktion, anhand welcher die Daten über das CGI-Script mit dem Webserver ausgetauscht werden. Diese Funktion (hier: **WebSrv_CgiGetUDINT**) muss ebenfalls im CoDeSys angelegt werden, kann aber frei benannt werden. Die Funktion liefert einen Bool'schen Ausdruck als Rückgabewert, True für den Fall, dass die Registrierung erfolgreich war. Die Registrierung dieser Funktion ist nötig, damit in jedem SPS-Zyklus die Daten vom bzw. zum Webserver aktualisiert werden.

Um den Webserver zu konfigurieren, müssen diese Funktionen nur ein einziges mal im SPS-Programm aufgerufen werden.

Funktion zur Verarbeitung des CGI - Scriptes

Wie oben bereits erwähnt, wird eine Funktion benötigt, welche die Daten mit dem CGI-Script austauscht. Da die Variablen nicht mit ihrem ursprünglich im CoDeSys verwendeten Namen zum Webserver übertragen werden können, muss hier auf den Variablen Index zurückgegriffen werden können. Dieser kann aber für jede Variable frei festgelegt werden. Im oben beschriebenen Beispiel heißt die Funktion: '**WebSrv_cgiGetUDINT**' und sieht wie folgt aus:

```
CASE VarIndex OF
    100 : WebSrv_CgiGetUDINT:=DOUTB0;
    101 : WebSrv_CgiGetUDINT:=PLC_PRG.Outbyte2;
    ELSE
        WebSrv_CgiGetUDINT:=0;
END_CASE;
```

Innerhalb dieser CASE-Anweisung wird der entsprechende Wert der CoDeSys-Variable (grün) anhand der Funktion (blau) dem Variablen-Index (rot) zugewiesen. Anhand dieses Indexes wird dann über das CGI-Script der Bezug zu den Variablen im HTML-Code hergestellt.

Das Selbe gilt auch für Bool'sche Variablen und für das Empfangen von Daten vom Webserver. Diese müssen dann aber als separate Funktion wieder wie oben beschrieben einmal in CoDeSys mit der Funktion **SysNet_HttpRegCgiFunction()** registriert werden. Das bedeutet, dass jeweils für senden und empfangen und bool und uint separate Funktionen angelegt und registriert werden müssen.

Bool'sche Variablen senden	UDINT Variablen senden
<pre>CASE VarIndex OF 102 : WebSrv_CgiGetBOOL:=PLC_PRG.Visible; 103 : WebSrv_CgiGetBOOL:=PLC_PRG.Outbyte2.0; ELSE WebSrv_CgiGetBOOL:=FALSE; END_CASE;</pre>	<pre>CASE VarIndex OF 200 : variable1:=Value; WebSrv_CgiSetUDINT:= TRUE; 201 : variable2:=Value; WebSrv_CgiSetUDINT:= TRUE; END_CASE</pre>

VarIndex: Der Index der Variablen mit der im CGI-Script der Bezug zur HTML-Variablen hergestellt wird.

WebSrv_CgiGetUDINT: Dies ist die Funktion der CoDeSys Bibliothek, anhand welcher die Daten über das CGI-Script vom oder zum Webserver übertragen werden.

PLC_PRG.Visible: Name der Variablen in der CoDeSys Programmierumgebung.

Web Visu Tutorial

Vorbereiten des Standard-Webverzeichnisses

Damit die Web-Visualisierung reibungslos funktioniert, müssen sich gewisse Dateien im Standard-Webverzeichnis befinden:

- **index.htm**: Standard Webseite, welche vom Server angezeigt wird.
- **xml_http.js**: Das Java-Script, welches für die Aktualisierung der Formulardaten auf der HTML-Seite sorgt.
- **autoupdate.cgx**: In dieser Datei wird der Bezug zwischen den im CoDeSys vergebenen Variablen Indizes und den Variablennamen im HTML-Code hergestellt. (siehe Beschreibung unten)

Vorbereiten der anderen Dateien

Erstellen der autoupdate.cgx

In dieser Datei wird die Verbindung zwischen Web-Visualisierung und SPS-Steuerung hergestellt. Die Variablen des HTML-Codes werden mit dem VarIndex aus dem CoDeSys Programm verknüpft.

Vor dem VarIndex der CoDeSys-Variablen muss auch der Variablentyp mit angegeben werden. D.h. in diesem Fall entweder bool oder udint.

Beispiel einer autoupdate.cgx-Datei:

```
t <form>  
r udint100 value100  
r udint101 value101  
r bool102 testvariable1  
r bool103 testvariable2  
r bool104 value104  
r udint220 value220  
r udint221 value221  
r udint222 value222  
t </form>
```

Das 't', bzw. das 'r' am Zeilenanfang, sowie auch die <form>/</form> -Tags müssen eingehalten werden. Mit einem ',' wird die Datei abgeschlossen.

Zeilenanfang	Typ und VarIndex in CoDeSys	Name(ID) im HTML-Formular	Beschreibung
r	udint100	value100	Der UDINT-Wert der Variablen mit Index 100 im CoDeSys wird dem Wert der Variablen value100 im HTML zugewiesen.
r	bool102	testvariable1	Der Bool-Wert der Variablen mit Index 102 im CoDeSys wird dem Wert der Variablen testvariable1 im HTML zugewiesen.

Die Datei autoupdate.cgx muss im Standard-Webverzeichnis gespeichert werden.

Web Visu Tutorial

Die Datei index.htm

Java-Script einbinden

Das HTML-Formular samt Inhalt wird gewöhnlich nur einmal beim ersten Download vom Server geladen. Dies führt natürlich dazu, dass die Daten schon nach kurzer Zeit veraltet sind. Durch das Ausführen dies Java-Scriptes ist es möglich, die Formulardaten zyklisch neu zu laden, ohne dass dabei die ganze Seite neu geladen werden muss. Das Java-Script wird hierzu einfach in den Header der HTML Datei eingebunden:

```
<html>
.
.
<head>
<title>.....</title>
.
<script language=JavaScript type="text/javascript" src="xml_http.js"></script>
<script language=JavaScript type="text/javascript">
var formUpdate = new periodicObj("autoupdate.cgx", 500); <!-- 500 is the refresh time in ms -->
function periodicUpdate() {
    updateMultiple(formUpdate);
    periodicFormTime = setTimeout("periodicUpdate()", formUpdate.period);
}
</script>
.
.
</head>
```

Die Zahl 500 gibt hier die Zeit in ms an, in der die Daten aktualisiert werden.

HTML-Formular erstellen

Der Datenaustausch zwischen Browser und SPS wird über ein sog. HTML-Formular realisiert.

Anzeigen eine UDINT-Wertes aus der SPS im Browser:

```
<form action="index.htm" method="post" id="indexform" name="indexform">
  <input type="text" id="value100" value="?" readonly size="3" name="outbyte">
</form>
```

action="index.htm": Name der Seite zu der die Daten übertragen werden.

method="post": HTML-Übertragungsmodus. Hier immer post.

id="indexform": Name des HTML-Elements.

name="indexform": Name des Formularfeldes.

type="text": Angabe, ob es sich um einen Text oder ein Bild oder ein anderes HTML-Element handelt.

id="value100": Name der Variablen die angezeigt werden soll. Diese steht so in der autoupdate.cgx.

value="?": Dieser Wert wird angezeigt wenn die Variable nicht angezeigt werden kann.

readonly: Es kann nur angezeigt, nichts eingegeben werden.

size="3": Breite des Formularfeldes.

Web Visu Tutorial

Eingabe eines UDINT Wertes:

```
<form action="index.htm" method="post" id="form02" name="form02">  
    <input type="text" name="udint200" id="value200" size="6" value="0"> input value 1 <br>  
    <input type="submit" name="button2" value="send to hipecs!">  
</form>
```

Die Angabe des `<type="submit">`-Tags bewirkt das Anzeigen einer Schaltfläche, mit der die Formulardaten zum Webserver abgeschickt werden.