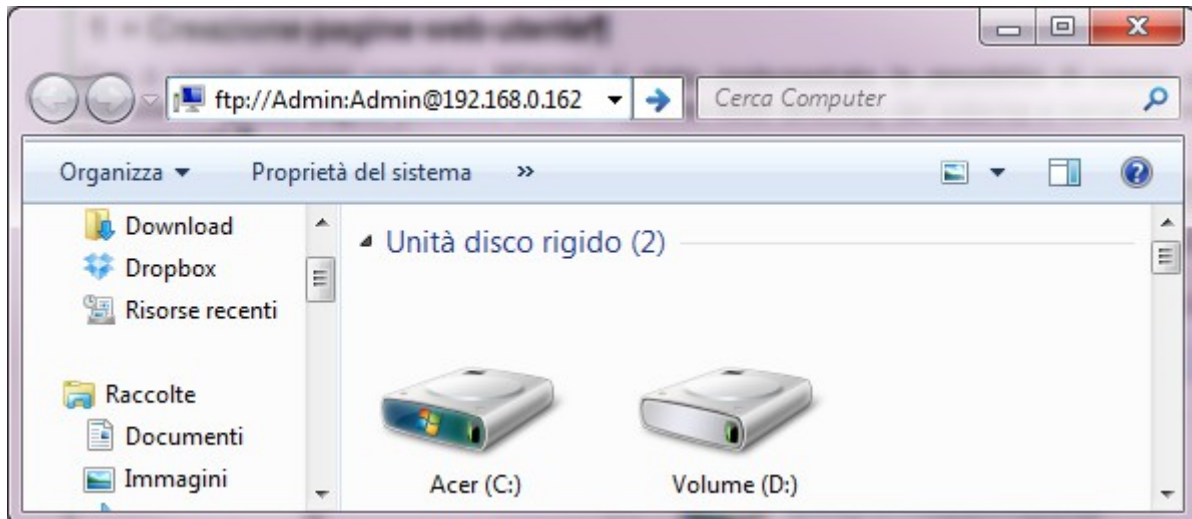


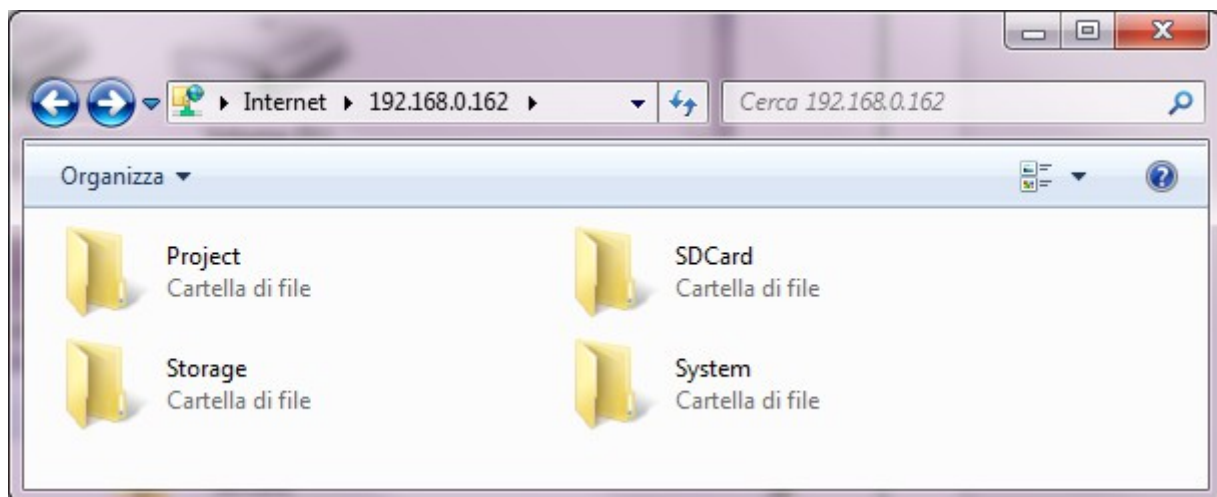
## 1 Creazione pagine web utente

Con il nuovo sistema operativo SFW184 è stata implementata la possibilità di creare pagine web direttamente dall'utente, queste pagine potranno essere trasferite nelle directory del sistema e verranno visualizzate accedendo da browser web.

Per trasferire le pagine web create dall'utente nel file system del sistema SlimLine occorre utilizzare un client Ftp (Esempio FileZilla) ma è possibile usare anche il semplice esplora risorse di Windows. Come si vede dalla figura sottostante, impostando nella barra indirizzo le credenziali di accesso al sistema ed il suo indirizzo IP **ftp://Admin:Admin@192.168.0.162**, è possibile connettersi e visualizzare il file system.



Ecco come si presenta la visualizzazione del file system alla connessione. Le cartelle **Project** e **System** sono riservate al sistema e si consiglia di **non modificarne il contenuto**. I file delle pagine utente possono essere trasferiti nelle cartelle **Storage** e **SDCard** (Se presente).



Quindi l'utente può creare le sue pagine web utilizzando un qualsiasi editor html ma anche semplicemente usando un semplice editor di testo come il blocco note, certo deve conoscere la sintassi del linguaggio html. Le pagine create saranno trasferite nella directory desiderata ed accedendo da un normale browser alla pagina la pagina sarà visualizzata.

## 1.1 Criteri per realizzazione pagina

Naturalmente il web server integrato nello SlimLine ha solo un ridotto set di funzioni e quindi nella creazione delle pagine web occorre sottostare a certe regole, vediamole:

- La pagina non può contenere inclusione di altre pagine (Esempio pagine di stile o di scripts).
- La pagina non può contenere inclusione di immagini (Esempio file gif o jpg), eventuali immagini possono essere embedded nella pagina stessa.

Vediamo ad esempio una semplice pagina che visualizza un messaggio di presentazione.

### Sorgente html pagina

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SlimLine - Simple page</title>
</head>
<body>
This page is served by the <b>SlimLine</b>
</body>
</html>
```

Salvando il testo riportato in un file, esempio **SPage.htm**, e trasferendo il file nella directory **Storage** di SlimLine, sarà possibile visualizzare la pagina web risultante semplicemente digitando nel proprio browser l'indirizzo della pagina.



Naturalmente la pagina può contenere links ad altre pagine, sarà così possibile realizzare una propria navigazione tra diverse pagine. Ecco lo stesso esempio di prima con incluso la definizione di uno stile.

### Sorgente html pagina

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SlimLine - Simple page</title>
<style type="text/css">
.Bolded {font-family: Arial, Helvetica, sans-serif; font-size: 20px; font-style: normal;font-weight: bold;}
</style>
</head>
<body>
This page is served by the <span class="Bolded">SlimLine</span>
</body>
</html>
```

## 1.2 Pagine dinamiche

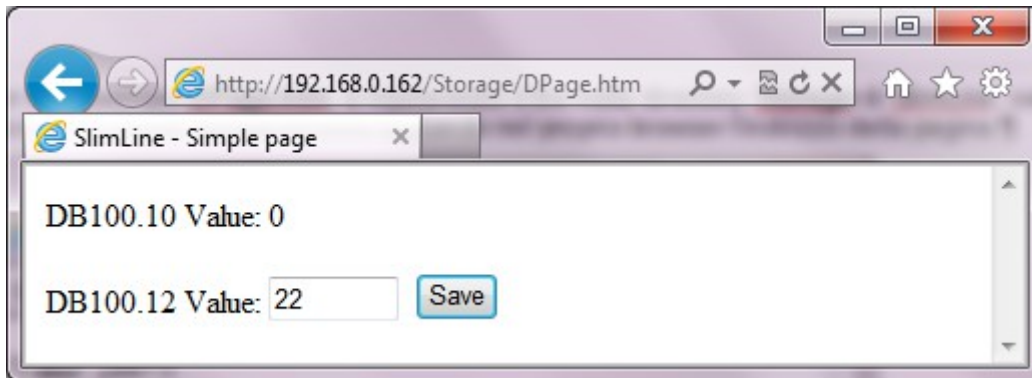
La caratteristica più importante del server web integrato in SlimLine è possibilità di gestire le **pagine dinamiche**. Una pagina web dinamica è una pagina il cui contenuto, in tutto o in parte, è generato sul momento dal server, potendo dunque essere diversa ogni volta che viene richiamata, consentendo quindi un'interattività con l'utente.

Ecco quindi che sarà possibile realizzare pagine che riportano valori di variabili PLC e permettono di modificare il valore di variabili PLC. Nell'esempio seguente riporto il sorgente html di una semplice pagine che visualizza il valore di una variabile PLC di tipo UINT allocata all'indirizzo DB100.10 e permette di impostare il valore di una variabile PLC di tipo UINT allocata all'indirizzo DB100.12.

### Sorgente html pagina

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SlimLine - Simple page</title>
<script language="javascript">
function Set(Field, Value) {document.MyForm[Field].value=Value;}
function SetValues() {Set("UINT12", "<!--["%d", UINT, 12]-->");}
</script>
</head>
<body onload="SetValues();">
DB100.10 Value:&nbsp;&nbsp;&nbsp;<!--["%d", UINT, 10]--></br>
<form id="MyForm" name="MyForm" method="post" action="DPage.htm">
DB100.12 Value:&nbsp;&nbsp;&nbsp;<input name="UINT 12" type="text" id="UINT12" size="5" maxlength="10">&nbsp;&nbsp;&nbsp;
<input name="MyButton" type="submit" id="MyButton" value="Save"/>
</form>
</body>
</html>
```

Salvando il testo riportato in un file, esempio **DPage.htm**, e trasferendo il file nella directory **Storage** di SlimLine, sarà possibile visualizzare la pagina web risultante semplicemente digitando nel proprio browser l'indirizzo della pagina.



Come si vede nella riga superiore viene visualizzato il valore della variabile PLC DB100.10 mentre impostando un valore nella casella di testo della riga inferiore e agendo sul tasto **Save** sarà possibile impostare il valore della variabile PLC DB100.12.

Naturalmente in una pagina web possono essere visualizzate e possono essere impostate tutte le variabili desiderate, si consiglia comunque di non esagerare con il numero di variabili, è preferibile suddividerle in più pagine.

## 1.3 Formato TAGs

Come si è visto in una pagina dinamica parte del contenuto viene generato sul momento dal server Http (Il modulo CPU SlimLine), vediamo quali sono i meccanismi per definire le **TAGs** da visualizzare. All'interno del sorgente della pagina Html è possibile definire dei campi di commento del tipo `<!--["%d", UINT, 10]-->`.

I campi sono interpretati come commenti e quindi sono gestibili da qualsiasi editor Html (Esempio Macromedia), ma il server Http nel momento in cui invia la pagina al client (Il browser che la visualizza) sostituisce al campo il valore della variabile indicata. Nell TAG sono riportate tutte le informazioni necessarie secondo la sintassi

**<!--[Format, Type, Address]-->**

### 1.3.1 Campo Format

La stringa di formattazione **Format**, può contenere elementi di due tipi, il primo consiste in caratteri che vengono ritornati nella pagina inalterati. Il secondo consiste in direttive di conversione che descrivono il modo in cui gli argomenti devono essere visualizzati. Le direttive di conversione iniziano con il simbolo % seguito dalle direttive secondo il formato:

**% [Flags] [Width] [.Precision] [Length] Conversion**

#### Flags

- +** La visualizzazione delle variabili con segno, inizierà sempre con il segno - o +.
- space** La visualizzazione delle variabili con segno, inizierà sempre con il segno - o con lo spazio.
- x** I valori diversi da 0 vengono prefissati con 0x.
- 0** Al valore visualizzato vengono aggiunti 0 fino al raggiungimento del numero di cifre desiderato (Per variabili di tipo d, i, o, u, x, X, e, E, f, g, G).

**Width:** Definisce il numero di cifre che devono essere visualizzate.

**Precision:** Definisce il numero di cifre decimali da visualizzare (Per variabili di tipo e, E, f).

#### Length

- h** Prima di (d, i, u, x, X, o) denota una variabile short int o unsigned short int.
- l (elle)** Prima di (d, i, u, x, X, o) denota una variabile long int o unsigned long int.
- L** Prima di (e, E, f, g, G) denota una variabile long double.

#### Conversion

- d** Valore decimale con segno.
- i** Valore decimale con segno.
- o** Valore ottale senza segno.
- u** Valore decimale senza segno.
- x** Valore esadecimale, viene visualizzato utilizzando lettere minuscole (Da 0 a 9, da a a f).
- X** Valore esadecimale, viene visualizzato utilizzando lettere maiuscole (Da 0 a 9, da A a F).
- e** Valore decimale in virgola mobile, visualizzato con indicazione dell'esponente (Esempio: [-]d.ddde+dd).
- E** Valore decimale in virgola mobile, visualizzato con indicazione dell'esponente (Esempio: [-]d.dddE+dd).
- f** Valore decimale in virgola mobile (Esempio: [-]d.ddd).
- c** Singolo carattere.
- s** Stringa.

### 1.3.2 Campo Type

Il campo **Type** indica il tipo di variabile che si vuole visualizzare, sono gestiti tutti i tipi definiti nella IEC61131.

### 1.3.3 Campo Address

Il campo Address indica l'indirizzo della variabile, ricordo che è possibile indicare solo variabili allocate nella DB 100.

### 1.3.4 Esempi di TAGs

Per meglio comprendere il formato di visualizzazione delle TAGs riporto alcuni esempi.

<!--["%d", UINT, 10]-->	Visualizza il valore della variabile UINT allocata all'indirizzo DB 100.10 con un numero di cifre intere variabili in base al valore.
<!--["%04d", UINT, 10]-->	Visualizza il valore della variabile UINT allocata all'indirizzo DB 100.10 sempre espresso con 4 cifre.
<!--["%3.0f", REAL, 32]-->	Visualizza il valore della variabile REAL allocata all'indirizzo DB 100.32 con 3 cifre intere e nessuna cifra decimale.
<!--["%4.2f", REAL, 50]-->	Visualizza il valore della variabile REAL allocata all'indirizzo DB 100.50 con 2 cifre intere e 2 cifre decimali.

## 2 Formato ARGs

La principale peculiarità del web dinamico è la possibilità di variare i contenuti delle pagine in base alle richieste degli utenti. Questa possibilità si materializza attraverso i meccanismi che permettono agli utenti, oltre che di richiedere una pagina ad un web server, anche di specificare determinati parametri da inviare al server web. Per impostare da pagina web valori di variabili PLC viene gestita una richiesta di tipo POST, il metodo è utilizzato con i moduli: quando una pagina Html contiene un tag **<form>**. I dati impostati nei vari oggetti contenuti nel **<form>** sono inviati in maniera da non essere direttamente visibili per l'utente, attraverso la richiesta HTTP che il browser invia al server.

Se ci riferiamo all'esempio precedente vediamo che la parte di pagina Html che permette l'impostazione della variabile PLC UINT allocata all'indirizzo DB 100.12 è la seguente.

### Sorgente html pagina

```
<form id="MyForm" name="MyForm" method="post" action="DPage.htm">
DB100.12 Value:&nbsp;  <input name="UINT 12" type="text" id="UINT12" size="5" maxlength="10">&nbsp;  
<input name="MyButton" type="submit" id="MyButton" value="Save"/>
</form>
```

In pratica un campo **<form>** con id **MyForm** contiene una casella di testo con id **UINT12** di dimensione 5 caratteri con un massimo impostabile di 10 caratteri. Nel form trova posto anche un pulsante di tipo submit la cui pressione esegue l'invio dell'intero modulo.

Definendo nel browser il valore della casella di testo ed agendo sul tasto Save, il dato inputato verrà inviato al server che visualizzerà la pagina **DPage.htm** e contemporaneamente provvederà a scrivere il valore definito nella variabile UINT DB100.12.

La funzione javascript **SetValues()**, che si trova in testa alla pagina Html ed eseguita automaticamente sul caricamento della pagina, serve a inizializzare la casella di testo con il valore reale della variabile UINT allocata a DB100.12.

### Sorgente html pagina

```
<script language="javascript">
function Set(Field, Value) {document.MyForm[Field].value=Value;}
function SetValues() {Set("UINT12", "<!--["%d", UINT, 12]-->");}
</script>
</head>
<body onLoad="SetValues();">
```

#### 2.1.1 ARG name

Il campo **name** dell'argomento è molto importante in quanto definisce il tipo di variabile PLC da impostare (Sono gestiti tutti i tipi definiti nella IEC61131) ed il suo indirizzo, i due campi devono essere separati da uno spazio.

Un nome del tipo UINT 12 indicherà una variabile UINT allocata ad indirizzo DB 100.12.

Un nome del tipo REAL 128 indicherà una variabile REAL allocata ad indirizzo DB 100.128.

Quindi un nome del tipo STRING 1000 indicherà una variabile STRING allocata ad indirizzo DB 100.1000.

#### 2.1.2 ARG id

Il campo **id** dell'argomento serve a referenziare l'oggetto all'interno del form in modo da poterlo valorizzare con la funzione **SetValues()**. La scelta di definirlo come UINT12 utilizzata nell'esempio è puramente indicativa, meglio sarebbe utilizzare una definizione che ne riprenda il significato (Esempio "SetPoint", "Preset", ecc.).

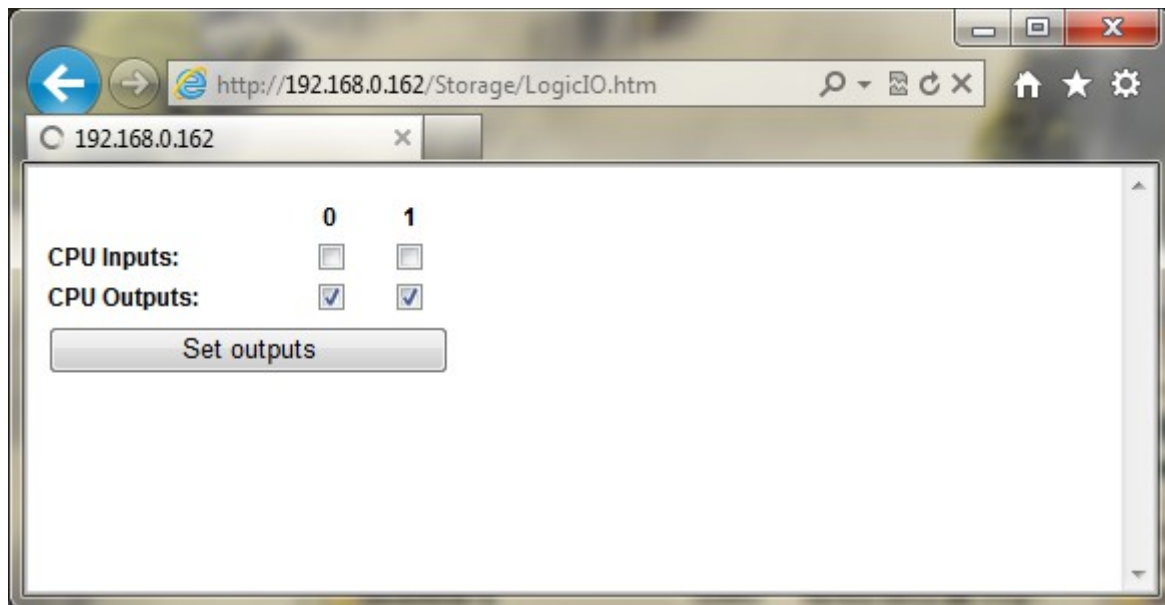
### 3 Alcuni esempi

Naturalmente le pagine web vanno create in base alle proprie esigenze inserendo gli oggetti desiderati. Per facilitare lo sviluppo delle proprie pagine viene fornito un programma dimostrativo PTP128\*000 che contiene una serie di programmi SlimLine e relative pagine web.

Per testare i vari programmi occorre trasferire sul modulo CPU il programma tramite LogicLab e tramite un client FTP trasferire la pagina **htm** nella directory **Storage**. Ora da un browser si digita l'indirizzo IP del modulo CPU seguito dalla directory e dal nome della pagina Esempio <http://192.168.0.122/Storage/Page.htm>.

### 3.1 LogicIO, gestione I/O logici

Ecco un esempio di gestione I/O logici da pagina web, per visualizzare lo stato degli ingressi e delle uscite sono stati utilizzati degli oggetti **checkbox**. Lo stato di attivo è indicato dalla presenza del tick, per attivare le uscite si pone il tick sulla uscita desiderata e si agisce sul tasto **Set outputs**.



Per visualizzare lo stato reale degli ingressi la pagina viene automaticamente rinfrescata ogni 10 secondi. Per ottenere l'aggiornamento della pagina dopo la direttiva **<head>** viene posta la dichiarazione:

```
<meta http-equiv="refresh" content="10">
```

Per la gestione della pagina sono utilizzate alcune funzioni javascript.

**Check(Field, Value)**, Imposta o rimuove il simbolo di tick sull'oggetto di tipo checkbox indicato in **Field** in base a **Value**.

**SetValues()**, Eseguita al caricamento della pagina esegue l'aggiornamento di tutti gli oggetti di tipo checkbox presenti.

**SubmitForm(Form)**, Eseguita su pressione del tasto Set outputs controlla se i checkbox di attivazione uscite sono settati ed aggiorna il valore dei campi hidden di scrittura variabili.

#### Sorgente funzioni javascript di pagina

```
<script language="javascript">
function Check(Field, Value) {document.MyForm[Field].checked=(Value != 0);}
function SetValues()
{
    Check("Inp00", '<!--["%d", BOOL, 0]-->');
    Check("Inp01", '<!--["%d", BOOL, 1]-->');
    Check("Out00", '<!--["%d", BOOL, 3]-->');
    Check("Out01", '<!--["%d", BOOL, 4]-->');
}

function SubmitForm(Form)
{
    if (document.getElementById('Out00').checked) document.getElementById('BOOL3').value="1";
    if (document.getElementById('Out01').checked) document.getElementById('BOOL4').value="1";
    document.forms[Form].submit();
}
</script>
```



### 3.2 COMPort, parametri comunicazione seriale

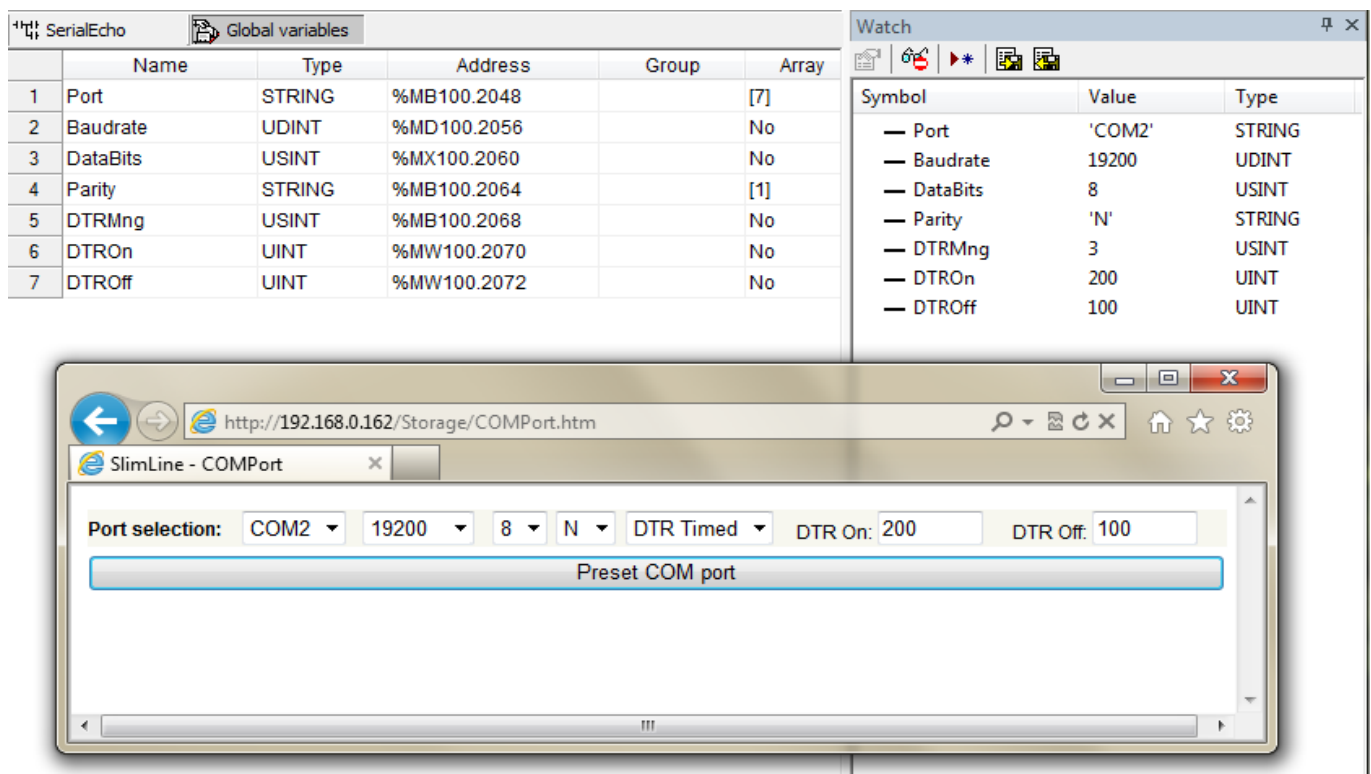
Ecco un esempio di come gestire la visualizzazione e l'impostazione dei parametri di comunicazione seriale da pagina web. Per la selezione della porta, baud rate, numero bit dato, tipo di parità e modo di gestione del segnale DTR si sono utilizzati oggetti **select**. L'oggetto select permette di selezionare il valore desiderato tramite una lista di valori.

Per le definizioni dei tempi di ritardo sul segnale DTR si sono utilizzati degli oggetti **text** che permettono di visualizzare ed impostare valori.

Richiamando la pagina vengono visualizzati i valori corrispondenti alle impostazioni definite nelle variabili dello SlimLine. I valori sono stati appoggiati su di un'area ritentiva per mantenerne il valore impostato anche allo spegnimento del sistema.

Definendo nuovi valori ed agendo sul tasto **Preset COM port** i valori saranno trasferiti nelle variabili dello SlimLine e la pagina verrà visualizzata con i nuovi valori definiti.

Ecco come si presenta la pagina web, è stata visualizzata sulla finestra del programma LogicLab che visualizza in debug i valori delle variabili.



The screenshot displays the LogicLab interface. At the top left, the 'Global variables' table lists the following parameters:

	Name	Type	Address	Group	Array
1	Port	STRING	%MB100.2048		[7]
2	Baudrate	UDINT	%MD100.2056		No
3	DataBits	USINT	%MX100.2060		No
4	Parity	STRING	%MB100.2064		[1]
5	DTRMng	USINT	%MB100.2068		No
6	DTROn	UINT	%MW100.2070		No
7	DTROff	UINT	%MW100.2072		No

To the right, the 'Watch' window shows the current values of these variables:

Symbol	Value	Type
Port	'COM2'	STRING
Baudrate	19200	UDINT
DataBits	8	USINT
Parity	'N'	STRING
DTRMng	3	USINT
DTROn	200	UINT
DTROff	100	UINT

Below these windows, a browser window titled 'SlimLine - COMPort' is shown at the URL <http://192.168.0.162/Storage/COMPort.htm>. The page contains the following configuration fields:

Port selection: COM2 | 19200 | 8 | N | DTR Timed | DTR On: 200 | DTR Off: 100

Below the fields is a large button labeled 'Preset COM port'.

### 3.2.1 Funzioni javascript

Per la gestione della pagina sono utilizzate alcune funzioni javascript.

**Set(Field, Value)**, Imposta il campo **text** di name **Field** con il valore **Value**.

**Choose(Field, Value)**, Seleziona l'oggetto **select** di name **Field** sull'opzione il cui valore corrisponde a **Value**.

**SetValues()**, Eseguita al caricamento della pagina esegue l'aggiornamento di tutti gli oggetti presenti.

#### Sorgente funzioni javascript di pagina

```
<script language="javascript">
function Set(Field, Value) {document.MyForm[Field].value=Value;}
function Choose(Field, Value)
{
    for (i=0; i<document.MyForm[Field].options.length; i++)
        if (document.MyForm[Field].options[i].value == Value) document.MyForm[Field].selectedIndex=i;
}

function SetValues()
{
    Choose("Port", '<!--["%s", STRING, 2048]-->');
    Choose("Baudrate", '<!--["%d", UDINT, 2056]-->');
    Choose("DataBits", '<!--["%d", USINT, 2060]-->');
    Choose("Parity", '<!--["%d", STRING, 2064]-->');
    Choose("DTRMng", '<!--["%d", USINT, 2068]-->');
    Set("DTROn", '<!--["%d", UINT, 2070]-->');
    Set("DTROff", '<!--["%d", UINT, 2072]-->');}
</script>
```