

### **Siemens Industry Online Support**

Document został zaczerpnięty ze stron supportu online firmy Siemens. Poniższy link prowadzi bezpośrednio do strony z której można go pobrać:

<http://support.automation.siemens.com/WW/view/en/58862931>

### **Uwaga**

Funkcje i rozwiązania opisane w tym artykule ograniczają się głównie do realizacji zadań automatyki. Ponadto proszę wziąć pod uwagę, że muszą być uwzględnione odpowiednie środki ochronne w kontekście bezpieczeństwa przemysłowego przy podłączaniu sterownika do istniejącego obiektu, sieci zakładowej lub Internetu. Więcej informacji na temat bezpieczeństwa można znaleźć tutaj:

<http://support.automation.siemens.com/WW/view/en/50203404>

Jeżeli podane informacje nie są wystarczające, zachęcamy do korzystania z forum internetowego na stronach Service & Support:

<http://www.siemens.com/forum-applications>

# Spis treści

<b>1</b>	<b>OPIS ZAGADNIENIA OMAWIANEGO W DOKUMENCIE .....</b>	<b>3</b>
<b>2</b>	<b>ROZWIĄZANIE W ZAKRESIE AUTOMATYKI .....</b>	<b>4</b>
2.1	OGÓLNY PRZEGLĄD ROZWIĄZANIA .....	4
2.2	OPIS ZAWARTOŚCI APLIKACJI .....	8
2.2.1	<i>Zawartość przykładowej aplikacji .....</i>	<i>8</i>
2.2.2	<i>Opis i przegląd interfejsu użytkownika .....</i>	<i>9</i>
2.3	UŻYTE KOMPONENTY SPRZĘTOWE I PROGRAMOWE .....	13
<b>3</b>	<b>PODSTAWY TWORZENIA STRON INTERNETOWYCH.....</b>	<b>15</b>
3.1	OGÓLNE ZASADY PROJEKTOWANIA STRON INTERNETOWYCH .....	15
3.1.1	<i>Zasady HTML.....</i>	<i>15</i>
3.1.2	<i>Używanie form.....</i>	<i>17</i>
3.1.3	<i>Podstawy stylu kaskadowego (CSS).....</i>	<i>17</i>
3.1.4	<i>Podstawy JavaScript.....</i>	<i>18</i>
3.1.5	<i>Automatyczne odświeżanie strony web .....</i>	<i>19</i>
3.2	PODSTAWY DOTYCZĄCE STANDARDOWEJ STRONY WEB .....	21
3.3	PODSTAWY DOTYCZĄCE STRON UŻYTKOWNIKA .....	23
3.3.1	<i>Tworzenie stron użytkownika .....</i>	<i>23</i>
3.3.2	<i>Wymagane bloki – WWW (SFC99).....</i>	<i>25</i>
3.4	WYŚWIETLANIE ZMIENNYCH CPU NA STRONIE INTERNETOWEJ .....	26
3.4.1	<i>Interakcje między przeglądarką internetową a CPU.....</i>	<i>26</i>
3.4.2	<i>Wymagania.....</i>	<i>27</i>
3.4.3	<i>Wymagane bloki – WWW (SFC99).....</i>	<i>27</i>
3.5	ZAPIS ZMIENNYCH CPU ZE STRONY INTERNETOWEJ .....	28
3.5.1	<i>Interakcje między przeglądarką internetową a CPU.....</i>	<i>28</i>
3.5.2	<i>Wymagania.....</i>	<i>29</i>
3.5.3	<i>Procedura .....</i>	<i>29</i>
3.6	POWIĄZANIE ZMIENNYCH Z TEKSTEM W PLIKU HTML.....	31
3.6.1	<i>Wymagania.....</i>	<i>32</i>
3.7	TWORZENIE OPTYMALNYCH CZASOWO STRON UŻYTKOWNIKA (OPCJA) .....	33
<b>4</b>	<b>MECHANIZMY FUNKCJONALNE APLIKACJI.....</b>	<b>35</b>
4.1	PODSTAWY FUNKCJONOWANIA PROGRAMU S7 .....	35
4.1.1	<i>Blok startowy (OB100) .....</i>	<i>36</i>
4.1.2	<i>Main (OB1).....</i>	<i>37</i>
4.1.3	<i>TankSimu (FB1) .....</i>	<i>39</i>
4.1.4	<i>DataString (FB2) .....</i>	<i>41</i>
4.2	PODSTAWY FUNKCJONOWANIA PLIKU HTML .....	42
4.2.1	<i>Komendy AWP .....</i>	<i>42</i>
4.2.2	<i>Informacje na temat doctype oraz nagłówka pliku HTML.....</i>	<i>43</i>
4.2.3	<i>Wyświetlanie obszarów .....</i>	<i>45</i>
4.2.4	<i>Wyświetlanie obrazów .....</i>	<i>47</i>
4.2.5	<i>Tworzenie tabel z tekstem.....</i>	<i>48</i>
4.2.6	<i>Wyświetlanie zmiennych CPU.....</i>	<i>49</i>
4.2.7	<i>Wyświetlanie tekstu poprzez wyliczenia.....</i>	<i>49</i>
4.2.8	<i>Ustawianie zmiennych w CPU za pomocą wartości i przycisku .....</i>	<i>50</i>
4.2.9	<i>Ustawianie zmiennych w CPU tylko za pomocą przycisku .....</i>	<i>51</i>
<b>5</b>	<b>KONFIGURACJA I USTAWIENIA .....</b>	<b>52</b>
5.1	PROCEDURA TWORZENIA STRONY INTERNETOWEJ .....	52
5.2	KONFIGURACJA S7-1200 (1214C DC/DC/DC) .....	53
5.3	TWORZENIE ZMIENNYCH W TABLICY ZMIENNYCH LUB BLOKU DB .....	54
5.4	TWORZENIE PLIKÓW HTML .....	55
5.5	USTAWIENIA WEB SERWERA I GENEROWANIE BŁOKÓW DANYCH .....	56
5.6	TWORZENIE PLIKÓW HTML .....	57
5.7	TWORZENIE, KOMPILACJA I ŁADOWANIE PROGRAMU S7 .....	58
5.8	WYWOŁANIE STRONY POPRZECZ PRZEGLĄDARKĄ INTERNETOWĄ.....	59

<b>6</b>	<b>INSTALACJA.....</b>	<b>61</b>
6.1	PROCEDURA TWORZENIA STRONY INTERNETOWEJ .....	61
6.2	INSTALACJA PRZYKŁADOWEJ APLIKACJI .....	62
<b>7</b>	<b>PRACA Z APLIKACJĄ .....</b>	<b>63</b>
<b>8</b>	<b>GLOSARIUSZ.....</b>	<b>67</b>
<b>9</b>	<b>LITERATURA.....</b>	<b>68</b>

# 1 Opis zagadnienia omawianego w dokumencie

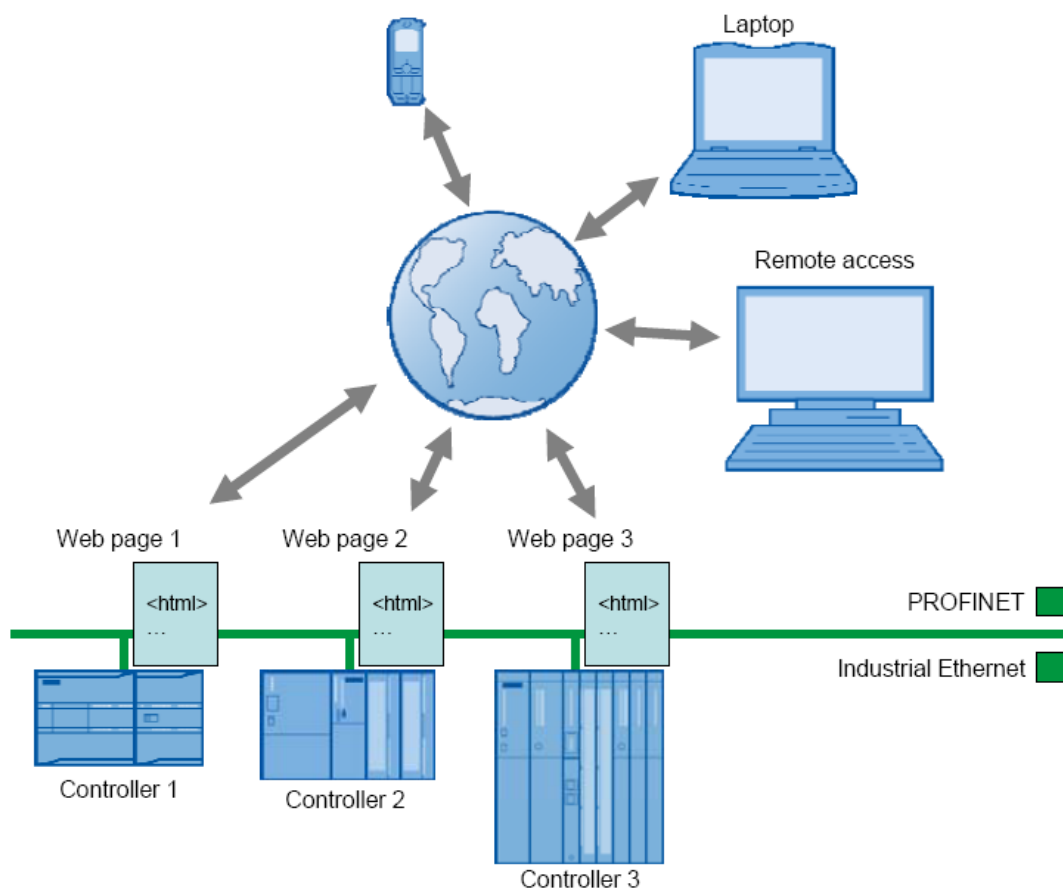
## Przegląd

### Przegląd zagadnienia omawianego w dokumencie

Nowoczesna technologia automatyki coraz bardziej integruje technologie internetowe, które – wraz z komunikacją opartą na Ethernetie – umożliwiają np. bezpośredni dostęp do systemu za pośrednictwem intranetu. W fazie uruchomienia i testów inżynier chce mieć elastyczny dostęp do CPU; indywidualne dane mają być wizualizowane w trakcie pracy do celów diagnostycznych.

Dla mechanizmów dostępu poprzez Internet lub intranet uzasadnione jest używanie już istniejących standardów, takich jak np. http, standardowe przeglądarki internetowe, oraz popularne języki programowania takie jak HTML (Hypertext Markup Language) lub JavaScript.

Rysunek 1-1



### Opis zagadnienia

Dostęp do CPU poprzez standardowy mechanizm web, jest możliwy po spełnieniu następujących warunków:

- Dostęp do CPU poprzez standardowy hardware oraz standardowe mechanizmy za pośrednictwem Industrial Ethernet. Nie jest potrzebny żaden dodatkowy sprzęt czy oprogramowanie.
- Indywidualny dostęp do CPU związanego z odpowiednim systemem (aplikacją). Każde CPU posiada swoją własną stronę zaszytą w wewnętrznej pamięci
- Personel obsługi z podstawową wiedzą z dziedziny automatyki.

## 2 Rozwiązanie w zakresie automatyki

### 2.1 Ogólny przegląd rozwiązania

#### Schemat

Sterowniki SIMATIC z interfejsem PROFINET dają możliwość dostępu do zmiennych w CPU za pomocą stron www dostarczanych przez system.

Dostęp do web serwera w CPU poprzez przeglądarkę internetową. Oprócz standardowego mechanizmów strony web, takich jak identyfikacja, bufor diagnostyczny, status modułów, komunikacja, status zmiennych oraz logi danych, możliwe jest również zaprojektowanie i wywołanie własnej strony Web odpowiedniej dla danej aplikacji.

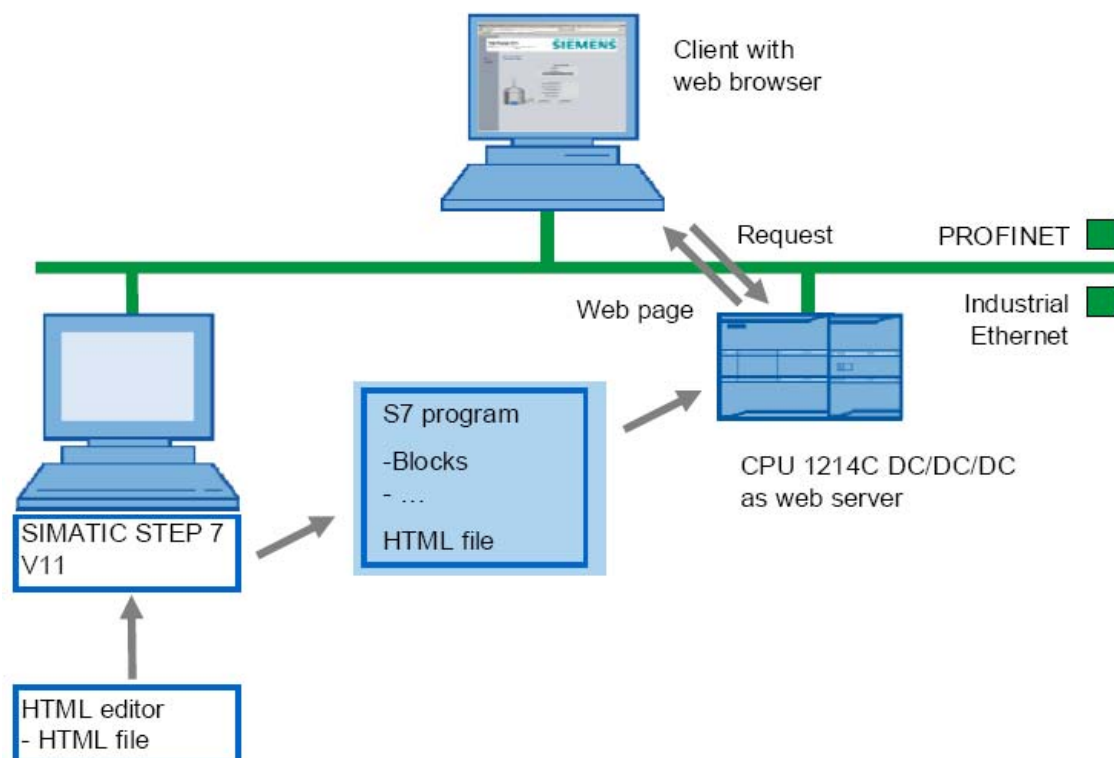
Web server ze stroną web jest standardowo zintegrowany w CPU.

Własną stronę web (user-defined web page), można stworzyć za pomocą dowolnych narzędzi takich jak Microsoft Frontpage, Notepad++, etc.. Można używać wszystkich opcji oferowanych przez HTML, CSS (Cascading Style Sheets) and JavaScript.

Dodatkowo jest zbiór specjalnych komend (AWP command) do bezpośredniej komunikacji z CPU.

Poniższy rysunek daje obraz rozwiązania.

Rysunek 2-1



**Zalety i możliwości aplikacyjne serwera www**

Dzięki opcji dostępu poprzez różne przeglądarki internetowe, zmienne mogą być odczytywane oraz w ograniczonym stopniu zmieniane z dowolnego komputera z dostępem do Internetu, bez instalowania dodatkowego oprogramowania.

Kolejną zaletą jest możliwość wykorzystania istniejącej infrastruktury sieci zakładowej bez żadnych dodatkowych komponentów sprzętowych. Oznacza to że dostęp do dowolnego sterownika może się odbyć z dowolnego miejsca w zakładzie w którym jest przyłącze do sieci.

Ocena, diagnostyka i sterowanie może odbywać się na duże odległości lub poprzez sieć WI-FI lub telefony z dostępem do Internetu.

Jednakże z powodu braku deterministycznego dostępu do web serwera, nie może być on pełnoprawnym zamiennikiem dla paneli operatorskich. Zadaniem tej aplikacji nie jest użycie takiej wizualizacji zamiast panelu HMI.

**WARNING**

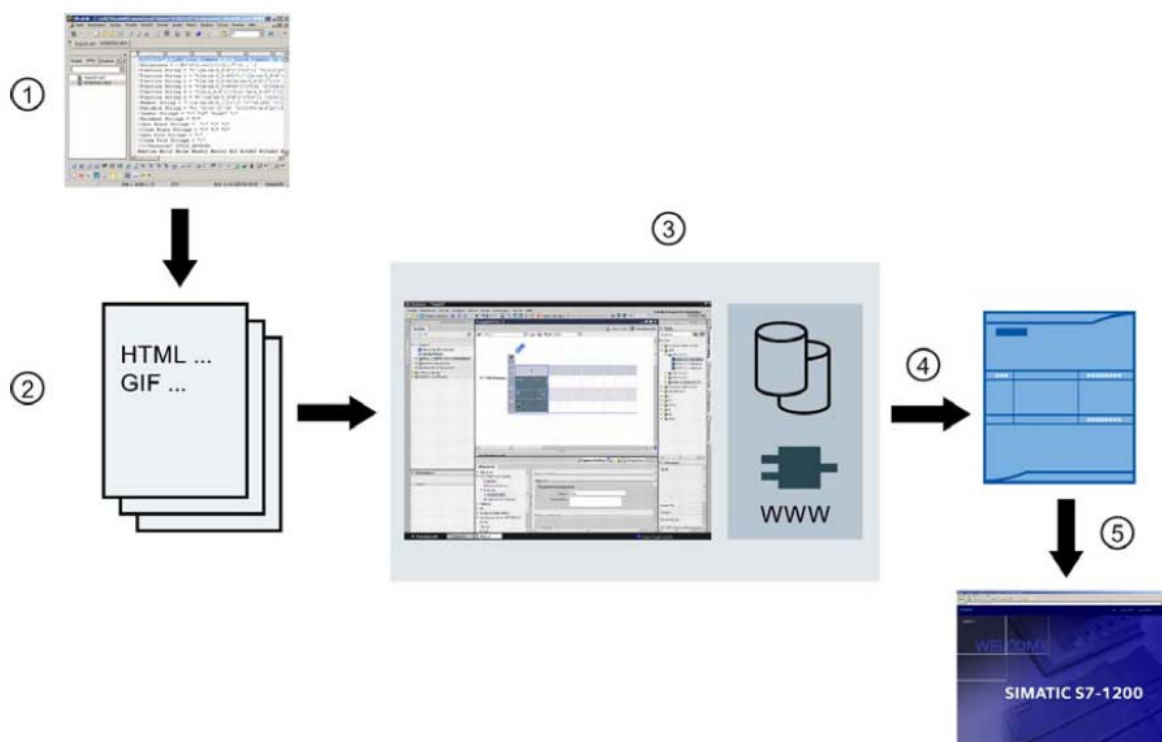
**Z powodu braku determinizmu, aplikacje oparte o web serwer nie mogą być używane do realizacji funkcji bezpieczeństwa SAFETY**

**UWAGA**

Ponadto należy pamiętać, że powinny być wzięte pod uwagę odpowiednie przemysłowe środki bezpieczeństwa, jeżeli system jest podpięty do innych części obiektu, sieci zakładowej lub Internetu. Aby uzyskać więcej informacji proszę odwiedzić stronę:  
<http://support.automation.siemens.com/WW/view/en/50203404>

**Procedura tworzenia stron internetowych w skrócie**

Rysunek 2-2



L.p.	Instrukcja
1	Utworzenie strony w edytorze HTML
2	Strona internetowa składa się z pojedynczych plików źródłowych np. *.html, *.gif, *.js, ...
3	Pliki HTML z obrazami itd. są przechowywane w bloku danych z SIMATIC STEP 7 V11 SP2. Wywołanie instrukcji WWW w programie S7
4	Transfer wszystkich bloków do CPU
5	Otwarcie strony web CPU poprzez przeglądarkę internetową. Dostęp do serwera web CPU jest niezależny od konfiguracji komputera. Każde urządzenie podłączone do portu PN CPU może wyświetlić stronę web.

Dokładny opis procedury tworzenia strony web oraz programowania w STEP 7 znajduje się w rozdziale 5 Konfiguracja i Ustawienia

**Struktura aplikacji**

Opisywana aplikacja zrealizowana była z CPU 1214C DC/DC/DC. PC jest podłączony poprzez interfejs PROFINET. PC służy tutaj do tworzenia programu S7, strony html, jak również wyświetlania strony w przeglądarce internetowej. Pokazane zostaną wszystkie kroki niezbędne do stworzenia strony web i wywołania jej w CPU

**Tematy nie ujęte w aplikacji**

Ta aplikacja jest wstępem do tworzenia stron użytkownika dla początkujących. Pokazane zostaną proste metody dostępu przez stronę web do CPU.

Aplikacja nie zawiera kompletnego opisu języka HTML. Aby uzyskać więcej informacji na temat HTML i JavaScript, należy zapoznać się z literaturą wyspecyfikowaną w rozdziale 9 Literatura.

**Wymagana wiedza**

Podstawowa znajomość sterowników S7 oraz programowania w STEP 7. Podstawowy wiedzy z zakresu HTML nie są wymagane.



## 2.2 Opis zawartości aplikacji

### 2.2.1 Zawartość przykładowej aplikacji

Przykładowa aplikacja dostarcza następującą treść:

- Konfiguracja web serwera dla CPU z interfejsem PN
- Utworzenie stronu użytkownika dla CPU z następującymi funkcjami:

Numery w tabeli nawiązują do numerów na rysunkach 2-3 do 2-5

Tabela 2-2

L.p.	Opis
1	Wyświetlanie zmiennych CPU
2	Graficzne przedstawienie zmiennych
3	Zmiana danych w CPU
4	Sprawdzanie wartości wejściowych z Javascript
5	Wyświetlanie tekstów połączonych ze zmiennymi w CPU
6	Wyświetlanie obrazów połączonych ze zmiennymi w CPU
7	Przechodzenie do strony web poprzez adres wpisany w pasku adresu
8	Cykliczne odświeżanie strony
9	Odświeżanie zmiennych zoptymalizowane czasowo

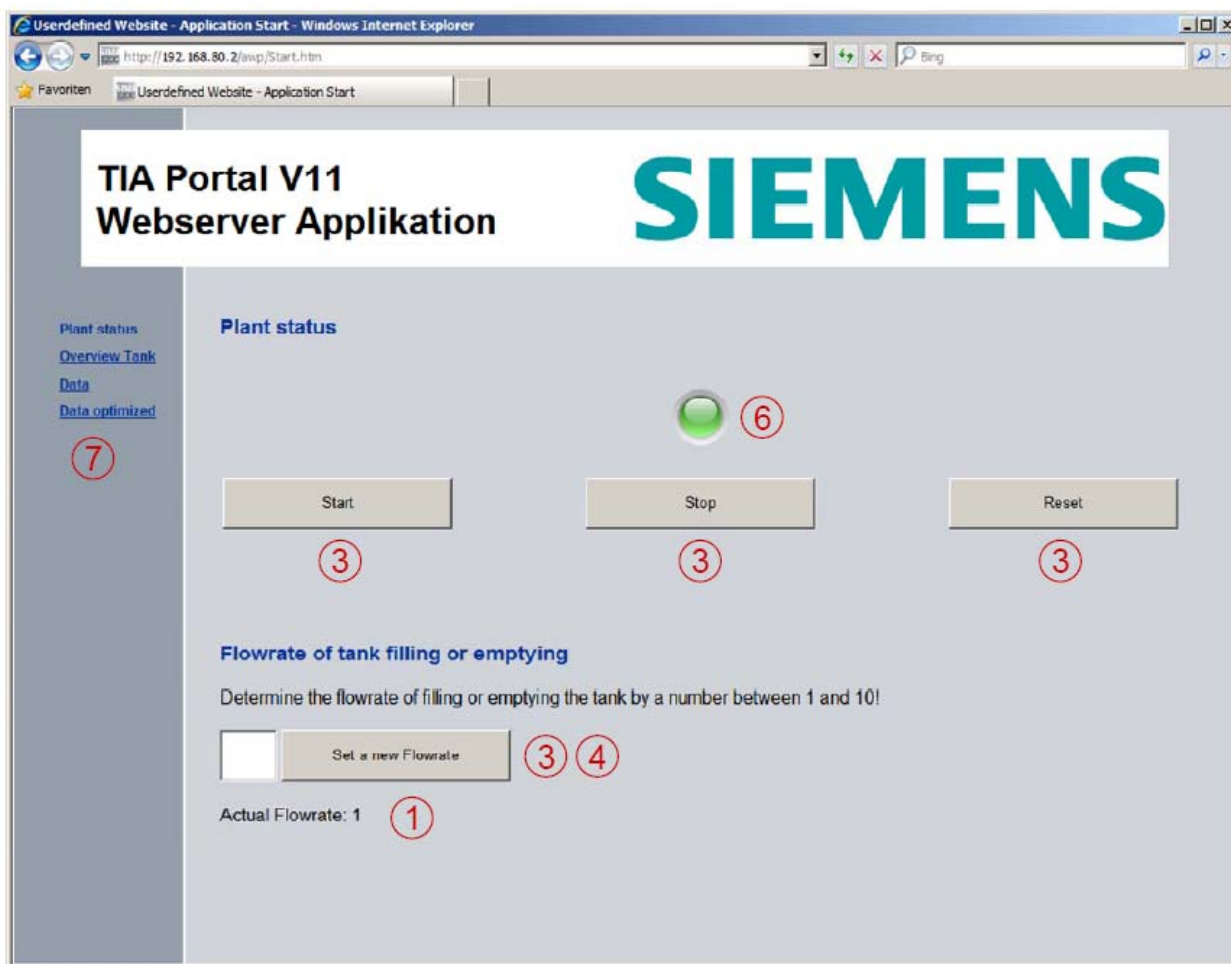
- Szczegóły tworzenia stron internetowych
  - dostarczanie zmiennych dla strony web
  - przetwarzanie zmiennych ze strony web w programie S7

## 2.2.2 Opis i przegląd interfejsu użytkownika

Poniższy rysunek przedstawia stronę internetową utworzoną dla tej aplikacji

### Strona startowa „Start”

Rysunek 2-3



Strona web pokazuje stronę startową aplikacji.

Aplikacja może zostać wystartowana przyciskiem „**Start**” i zatrzymana przyciskiem „**Stop**”. Przycisk „**Reset**” inicjalizuje dane aplikacji do stanu początkowego. Status pracy aplikacji jest wyświetlany przy użyciu diody LED.

Wartość przepływu z jaką zbiornik jest napełniany i opróżniany definiowany jest poprzez przycisk „**Set a new Flowrate**”. Domyślnie przepływ ustawiony jest na 5. Im wyższa wartość przepływu jest ustawiona tym szybciej zbiornik jest napełniany lub opróżniany.

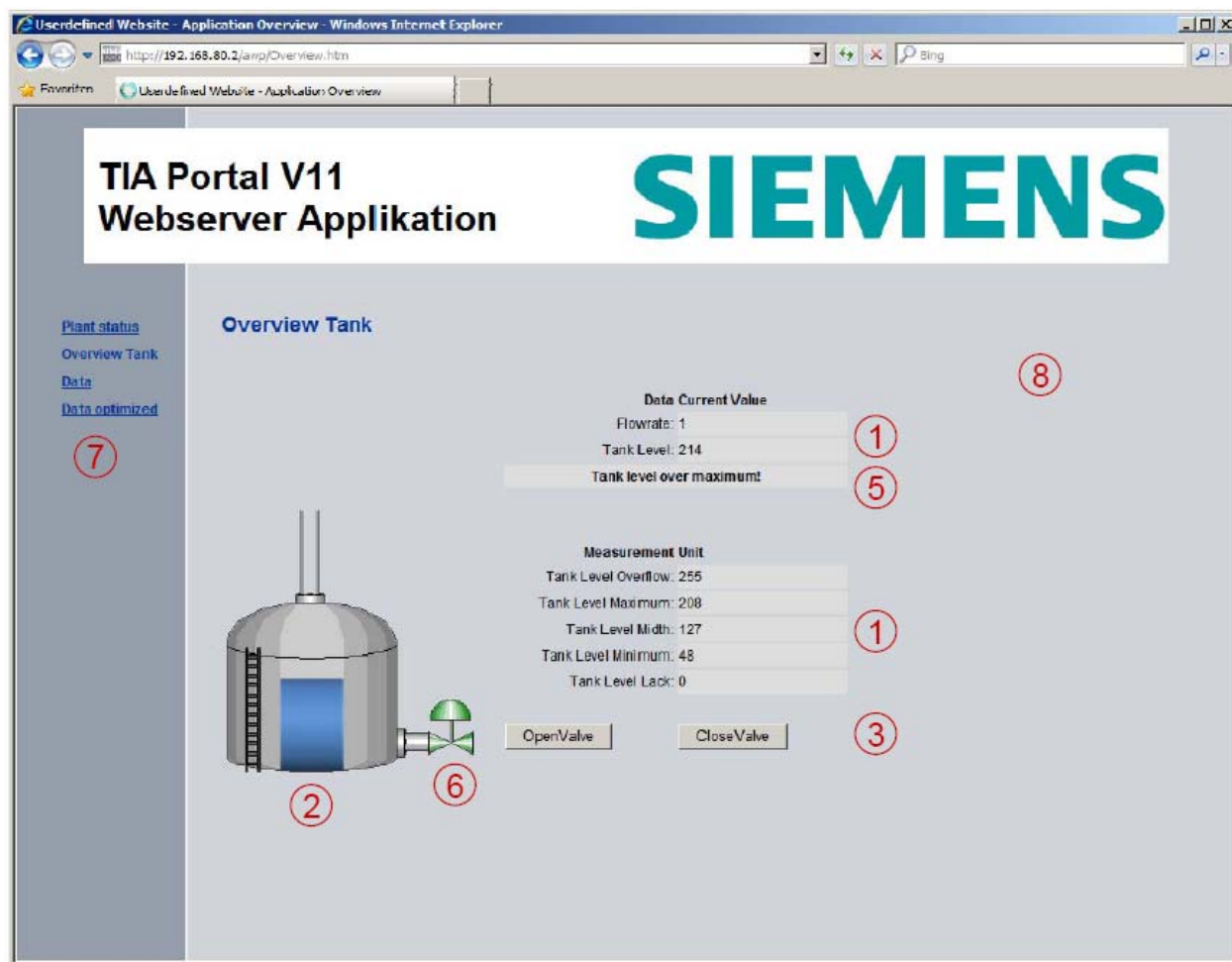
Wpisywane wartości przepływu jest monitorowane przez skrypt. Jeżeli wartość jest niepoprawna, wtedy wyświetlany jest komunikat.

Aktualna wartość przepływu wyświetlana jest w polu „Actual Flowrate”.

Na pasku nawigacyjnym z lewej strony znajdują się linki do pozostałych stron aplikacji.

## Strona „Overview Tank”

Rysunek 2-4



Strona web pokazuje zbiornik oraz jego poziom napełnienia „**Tank Level**”. Wartości graniczne napełnienia zbiornika pod „**Measurement**”.

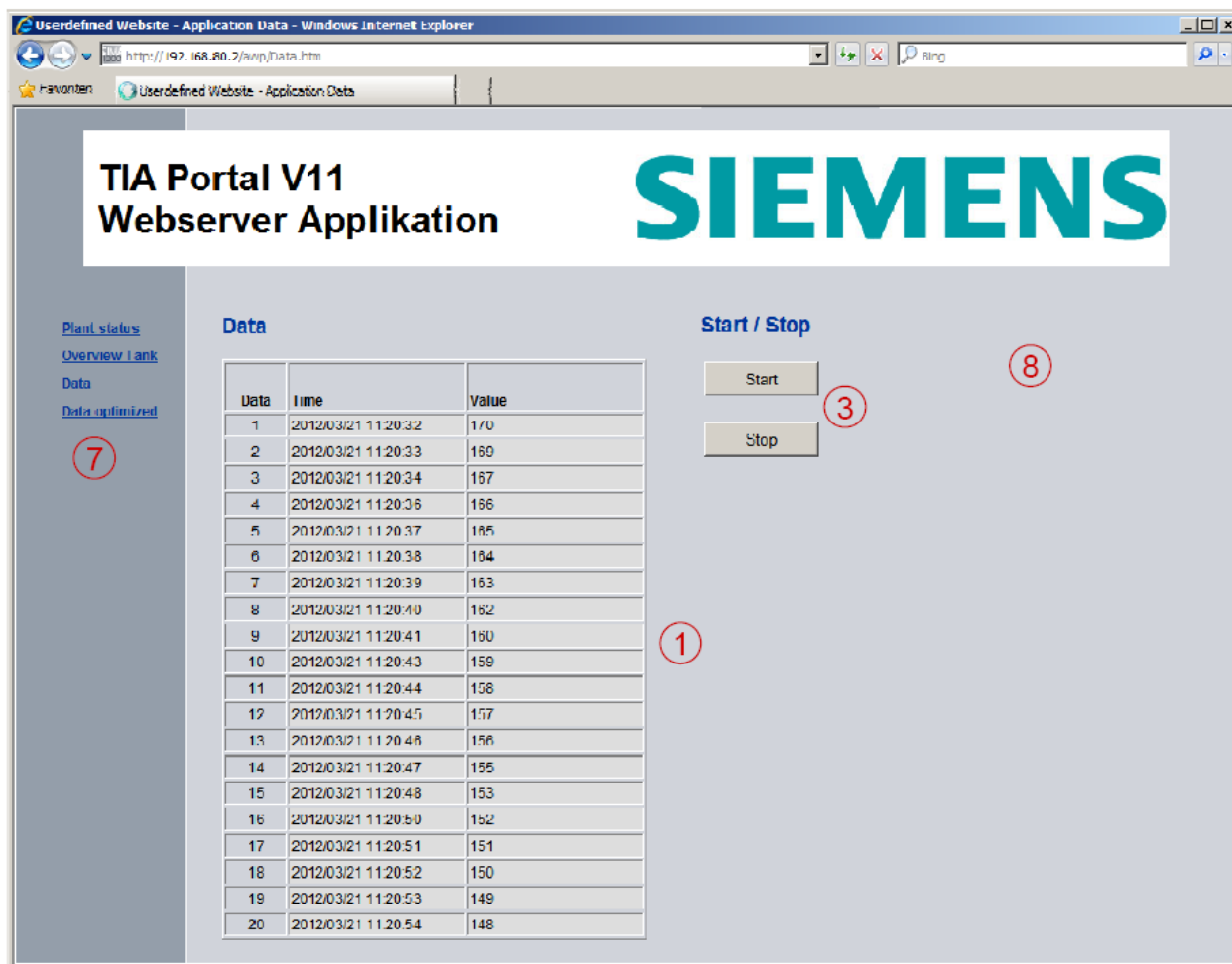
Poprzez przycisk „**OpenValve**” otwarty zostaje zawór zbiornika i następuje opróżnianie zbiornika. Po naciśnięciu przycisku „**CloseValve**” zawór zostaje zamknięty i następuje napełnianie zbiornika. W zależności od tego który przycisk został wciśnięty, stan zaworu jest przedstawiany poprzez zmianę jego koloru.

Strona jest automatycznie odświeżana.

Na pasku nawigacyjnym z lewej strony znajdują się linki do pozostałych stron aplikacji.

**Strona „Data”**

Rysunek 2-5



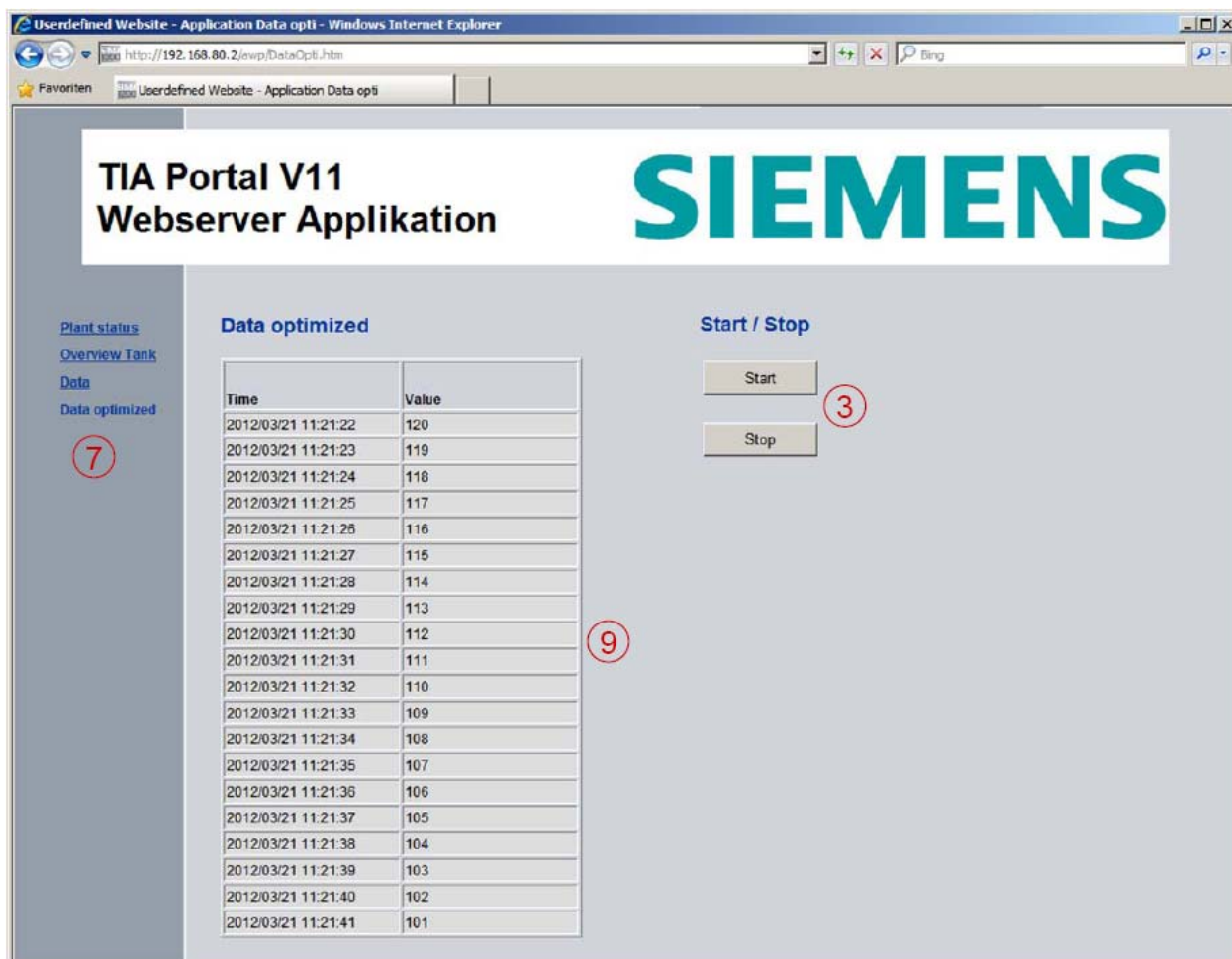
Strona web pokazuje bufor zapisanych ostatnio 20 poziomów zbiornika.

Strona jest automatycznie odświeżana.

Na pasku nawigacyjnym z lewej strony znajdują się linki do pozostałych stron aplikacji.

## Strona „Data Optimized”

Rysunek 2-6



Strona web jest podzielona na pliki HTML (strona główna) bez zmiennych oraz dodatkowo pliki HTML do czytania i zapisu danych. Dodatkowe pliki HTML są dołączone jako ramki do strony głównej.

Strona web pokazuje 20 ostatnich wpisów poziomu cieczy w zbiorniku. Dane które mają być wyświetlane nie są przesyłane indywidualnie do web serwera lecz jako 3 zmienne typu string. Skrypt Java dzieli wyjściowe stringi na pojedyncze zmienne i udostępnia je na stronie jako dane wyjściowe. Odświeżanie poszczególnych ramek jest realizowane z poziomu skryptu Java. Uruchomienie aplikacji następuje po naciśnięciu przycisku „**Start**”, a zatrzymanie „**Stop**”. Ze względu na to że przyciski znajdują się w niezależnej ramce, cała strona nie zostanie odświeżona podczas zapisu.

Na pasku nawigacyjnym z lewej strony znajdują się linki do pozostałych stron aplikacji.

## 2.3 Użyte komponenty sprzętowe i programowe

Aplikacja stworzona została w oparciu o następujące komponenty

### Sprzęt

### UWAGA

Dla tej aplikacji wymagany jest sterownik z aktualnym firmware'm. W zależności od typu CPU poniższy link zawiera linki do pobrania odpowiedniego firmware'u

<http://support.automation.siemens.com/WW/view/en/41886031/133100>

Komponent	ilość	Numer zamówieniowy	Uwagi
CPU 1214C DC/DC/DC Firmware V2.2	1	6ES7214-1AE30-0XB0	Zamiennie dowolne CPU 1200 z firmware'm 2.2 może być użyte
PG/PC with Ethernet interface	1	-	-
IE FC TP STANDARD CABLE	1	6XV1840-2AH10	Przewód ethernetowy, minimalna ilość zamówieniowa 20m
RJ45 plug-in connector	2	6GK1901-1BB10-2AA0	Mogą być dopasowane odpowiednie inne wtyczki

**Oprogramowanie**

Komponent	ilość	Numer zamówieniowy	Uwagi
SIMATIC STEP 7 V11 SP2	1	6ES7822-1AA01-0YC5	-
Oprogramowanie do tworzenia stron internetowych, np. Frontpage, Notepad++, ...	1	-	-
Przeglądarka internetowa np. Internet Explorer, Mozilla Firefox <sup>1)</sup>	1	-	Optymalnie – Internet explorer 8.0

<sup>1)</sup> Web serwer S7-1200 został sprawdzony z poniższymi przeglądarkami:

- Internet Explorer 8.0 lub wyższy
- Mozilla Firefox 3.0 lub wyższy
- Opera 11.0 lub wyższy

**UWAGA**

Przy używaniu innych przeglądarek może być wymagane dokonanie pewnych modyfikacji

**UWAGA**

Zauważono problemy przy próbie zapisu tagów w przeglądarce Internet Explorer 9.0. Nie jest to winą web serwera S7-1200 ani napisanej aplikacji. Aby uniknąć tego problemu zaleca się stosowanie przeglądarki Internet Explorer 8.0

**Przykładowe pliki i projekty**

Poniższa tabelka zawiera wszystkie pliki i projekty użyte w tej aplikacji

Komponent	Uwagi:
Example_S7-1200_Webserver.zip	Plik zip zawiera project STEP7 oraz plik HTML. Plik HTML wraz z powiązanymi plikami jest zlokalizowany w \html directory
58862931_S7-1200_Webserver_DOKU_V10_en.pdf	Ten dokument

## 3 Podstawy tworzenia stron internetowych

### Ogólne informacje

W kontekście projektowania stron internetowych, termin strona internetowa (strona web) służy do określania dokumentu w globalnej sieci Web (Word Wide Web), która może być wywoływana z serwera www za pomocą przeglądarki internetowej przez kreślenie adresu Uniform Resource Locator (URL).

Strona internetowa użytkownika (user-defined web page) jest rozumiana jako strona z dodatkowymi komendami takimi jak AWP commands, które mogą być używane w celu uzyskania dostępu do S7 CPU z interfejsem PN.

### 3.1 Ogólne zasady projektowania stron internetowych

Jeżeli ogólne zasady projektowania stron internetowych są znane, można pominąć punkt 3.1 i przejść od razu do punktu 3.2.

#### 3.1.1 Zasady HTML

Skrót HTML oznacza „Hypertext Markup Language” i jest oparty na znacznikach nagłówków, tekstów, list, tabel lub obrazów. Nie wykorzystuje natomiast deklaracji zmiennych oraz pętli tak więc nie jest językiem programowania.

##### Struktura

Dokument HTML składa się z trzech obszarów:

- Deklaracja typu dokumentu (doctype) na początku pliku, rozpoczyna dokument definicją typu (DTD) używając np. HTML 4.01 Transitional.
- Nagłówek HTML (head) – informacje które nie są wyświetlane w przeglądarce internetowej.
- Ciało HTML – informacje, które są wyświetlane w przeglądarce internetowej.

##### Elementy HTML (tagi)

Elementy używane do identyfikacji i struktury różnych części strony internetowej. Plik HTML zawiera elementy HTML „HTML elements”, które są oznaczane przez tagi (pary tagów). Prawie wszystkie elementy HTML są oznaczone przez tag wstępny i kończący.

Zawartość między tymi tagami to zakres stosowania danego elementu.

Przykład: Tekst oznaczony tagiem <p>. Tag kończący jest reprezentowany przez wstępny znak „</”.

<p>To jest tekst</p>

Tagi są kaskadowe i mogą być zagnieżdżane.



### Typowe tagi

Poniższa tabela daje przegląd najważniejszych tagów do tworzenia struktury informacji, są one użyte w tej przykładowej aplikacji.

Reprezentacja	Funkcja	Przykład
<!-- ... -->	Komentarz	<!-- This is a comment! -->
<a> ... </a>	Link	
<b> ... </b>	Pogrubienie	<b>This text is bold</b>.
<body> ...</body>	Zawartość jest wyświetlana w przeglądarce	
<div> ... </div>	Grupowanie elementów	
<form> ... </form>	Definicja formy	
<h1> ... </h1>	Nagłówek tekstu	
<head> ... </head>	Obszar nagłówka pliku HTML	
<html> ... </html>	Fundamentalny tag strony internetowej	
<iframe> ... </iframe>	Definiuje wbudowane okno	
<img>	Integruje obraz	
<input>	Tworzy element formy	
<link>	Tworzy logiczne powiązanie z innymi plikami	
<meta>	Definiuje dane meta	
<p> ... </p>	Paragraf	
<script> ... </script>	Definiuje obszar skryptu (np. JavaScript)	
<style> ... </style>	Definicja stylu formatowania	
<table> ... </table>	Tworzy tabelę w powiązaniu z <tr> oraz <td>	
<td> ... </td>	Kolumna tabeli	
<th> ... </th>	Nagłówek tabeli	
<tr> ... </tr>	Wiersz tabeli	

### 3.1.2 Używanie form

Formy używane są w celu umożliwienia stworzenia interakcji z użytkownikiem strony. Na przykład, użytkownik może wypełniać pole tekstowe a następnie przesłać ten tekst przez naciśnięcie przycisku. Zawartość formy jest wtedy przesyłana do web serwera. Metoda „POST” służy do transferu zawartości formy z przeglądarki do web serwera.

### 3.1.3 Podstawy stylu kaskadowego (CSS)

Formatowanie CSS jest językiem formatowania dla elementów HTML. Za pomocą arkuszy stylów są określane takie parametry jak np. czcionka, rozmiar czcionki, kolor, obramowanie, wysokość, szerokość itd.

Można zdefiniować centralne formatowanie dla np. wszystkich nagłówków, okien tabeli itd.

Struktura formatowania CSS jest następująca:

Selektor {właściwość: wartość}

Selektor może zawierać kilka deklaracji {właściwość: wartość}

#### Typowe właściwości CSS

Poniższa tabela daje przegląd najważniejszych właściwości formatowania elementów HTML, są one użyte w tej przykładowej aplikacji.

Wartość CSS	Funkcja	Przykład
position	Typ pozycji	static, relative, absolute, fixed
top left bottom right	Pozycja od góry Pozycja od lewej Pozycja od dołu Pozycja od prawej	10px, 2%
width height	Szerokość Wysokość	100px, 20%
direction	Kierunek	ltr, rtl
z-index	Pozycja w warstwach nakładających się	1, 2
font-family	Czcionka	Arial, Helvetica
font-style	Styl czcionki	italic, oblique, normal
font-size	Rozmiar czcionki	20px, 100%, small, medium, large
font-weight	Waga czcionki	bold, normal, bolder, lighter, 100 to 900
text-decoration	Dekoracja tekstu	underline, blink, none
text-transform	Transformacja tekstu	uppercase, lowercase
color	Kolor tekstu	rgb(51,102,170), #FFFFFF
vertical-align:	Pionowe ustawienie	top, middle, bottom
text-align	Poziome ustawienie	left, center, right, justify
margin margin-top margin-right margin-bottom margin-left	Margines/ogólny odstęp Margines/odstęp górny Margines/odstęp prawy Margines/odstęp dolny Margines/odstęp lewy	10px, 5%
padding padding-top padding-right padding-bottom padding-left		10px, 5%
border[-top, -right, -bottom, -left] border[-top, -right, -bottom, -left] border[-top, -right, -bottom, -left] border[-top, -right, -bottom, -left]	Obramowanie/ogólnie  Obramowanie/grubość  Obramowanie/kolor	2px solid white 2px, 1%, thin, medium, thick #FFFF00, white none, hidden, dotted, solid, dashed, double separate, collapse

border-collapse	Obramowanie/typ Obramowanie/model	
background background-color background-image background-repeat background-attachment background-position	Kolor i obraz tła Kolor tła Obraz tła Efekt przeładowania strony Efekt znaku wodnego Pozycja tła	Image.png no-repeat rgb(51,102,170), #FFFFFF Image.png repeat, no-repeat, repeat-x, repeat-y scroll, fixed 10px 10px, top, bottom, center, left, right
list-style-type	Lista typów stylu	none, square, circle, disc
empty-cells	Wyświetlanie pustych celek	show, hide

## Integracja stylów CSS w HTML

Istnieje kilka sposobów integrowania arkuszy stylów w pliku HTML:

- wewnątrz elementu HTML
- pomiędzy tagami <script> oraz </script>
- w zewnętrznym pliku CSS

Arkusze stylów definiowane są w oddzielnym pliku CSS, jeżeli ma być użyte jednolite wzornictwo w kilku plikach HTML. Ten plik CSS jest integrowany w pliku HTML. Składnia jest następująca:

```
<link rel="stylesheet" type="text/css" href="<Formats>.css">
```

Zdefiniowany arkusz stylów jest adresowany poprzez id oraz atrybut klasy tagów HTML. Plik CSS dostarcza rozbudowane opcje formatowania.

### 3.1.4 Podstawy JavaScript

JavaScript jest językiem programowania stworzonym w celu optymalizacji stron HTML..

Aplikacje JavaScript są wykonywane w przeglądarce internetowej i bezpośrednio przez nią interpretowane w trybie runtime.

JavaScript nie zastępuje HTML, lecz jest uzupełnieniem.

JavaScript jest całkiem inne od języka programowania Java. Podobieństwo nazewnictwa jest spowodowane jedynie celami marketingowymi.

Dzięki JavaScript można, oprócz innych rzeczy, rozszerzyć stronę HTML przy użyciu następujących funkcji:

- Przetwarzanie żądań z klawiatury
- Dynamiczna modyfikacja stron web

#### Integracja JavaScript w HTML

Istnieje kilka sposobów integracji JavaScript w HTML:

- pomiędzy tagami <script> oraz </script>
- poprzez referencję
- jako parametr Tagu HTML
- jako zewnętrzny plik JS

Użyteczne jest pisanie kodu JavaScript w oddzielnych plikach, jeżeli zachodzi potrzeba użycia tych samych funkcji JavaScript w kilku plikach HTML.

Składnia jest następująca:

```
<script src="<Script>.js" type="text/javascript"> </script>
```

### 3.1.5 Automatyczne odświeżanie strony web

#### Czas trwania przeładowania strony

Czas trwania przeładowania strony zależy od zawartości tej strony. Części statyczne oraz dynamiczne (zmiennne) muszą być odświeżone.

#### Czas transmisji zmiennych

Wewnętrzny czas przesyłania między CPU i wbudowanym web serwerem zależy od ilości przesyłanych zmiennych. Prędkość transmisji można zwiększyć w ustawieniach CPU, jako procent czasu cyklu programu.

Przegląd czasów transmisji w zależności od ilości użytych zmiennych i konfiguracji procentu czasu cyklu przeznaczonego na komunikację przedstawia tabela poniżej.

Tabela 3-3

Liczba zmiennych	Obciążenie komunikacji [%]	Czas odświeżania [s]
10	20	2.4
10	40	2.1
20	20	3.3
20	40	2.8
40	20	5.9
40	40	4.8

## UWAGA

Należy wykasować z pliku HTML wszelkie nieużywane tagi, aby zmniejszyć czas transmisji. Zaznaczenie zmiennych, jako komentarz jest niewystarczające

#### Opcje

Opcja automatycznego odświeżania działa tylko w przypadku standardowych stron web, nie działa w przypadku stron użytkownika.

Strona HTML jest statyczna i nie reaguje na zmianę zawartości.

Pożądane jest natomiast aby wartości zmiennych z programu S7 były widoczne w przeglądarce. Istnieje kilka sposobów na odświeżenie zawartości strony:

- Ręczne odświeżanie za pomocą przycisku „F5”
- Automatyczne odświeżanie za pomocą odpowiedniego polecenia w nagłówku kodu HTML
- Automatyczne odświeżanie za pomocą JavaScript umieszczonego w pliku HTML

W celu zapisania wartości ze strony do CPU, powinna być utworzona oddzielna strona HTML bez automatycznego odświeżania. Pozwoli to zapobiec wpisaniu do CPU niekompletnej wartości podczas automatycznego odświeżania.

#### Ręczne odświeżanie

Po wciśnięciu klawisza „F5” (Internet Explorer: „View > Refresh”), zawartość okna w przeglądarce zostaje odświeżona.

#### Odświeżanie przez HTML

Za pomocą poniższego kodu umieszczonego w nagłówku HTML, zawartość strony internetowej będzie odświeżana cyklicznie:

```
<meta http-equiv="refresh" content="10; URL=Example.htm">
```

Czas cyklu odświeżania podawany jest w sekundach. W przykładzie wpisano content="10, oznacza to, że strona będzie odświeżana co 10s. Rzeczywisty czas odświeżania strony zależy od złożoności strony internetowej. Im więcej zawartości na stronie tym dłuższy czas odświeżania. Można wpisać stronę poprzez „URL=". W przykładzie są to strony „Overview.html” oraz „Data.html”.

**Odświeżanie za pomocą JavaScript**

Poniższy kod umieszczony w „ciele” HTML pozwoli na odświeżania zawartości strony co 10 sekund:

```
<script type="text/javascript">  
setInterval("document.location.reload()",1000);  
</script>
```

## 3.2 Podstawy dotyczące standardowej strony web

### Wymagania

Opcja web server musi zostać zaznaczona we właściwościach PLC.

Jeżeli potrzebny jest bezpieczny dostęp do standardowej strony należy zaznaczyć opcję „Permit Access only with HTTPS”.

Automatyczne odświeżanie strony jest włączone. Czas odświeżania jest standardowo ustawiony na 10 sekund i nie może być zmieniony.

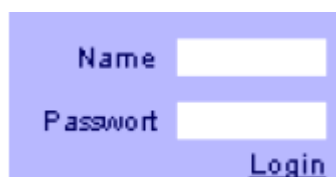
### Dostęp poprzez HTTP lub HTTPS

Za pomocą URL „http://ww.xx.yy.zz” lub „https://ww.xx.yy.zz” możliwy jest dostęp do standardowej strony web. „ww.xx.yy.zz” nawiązuje do adresu IP sterownika S7-1200.

HTTPS jest używane do szyfrowania i uwierzytelniania komunikacji między przeglądarką i serwerem web. Gdy „Permit access only with HTTPS” jest zaznaczone, wywołanie strony jest możliwe tylko poprzez HTTPS.

### Logowanie

Nie ma potrzeby logowania aby mieć dostęp do strony web. Wykonywanie akcji takich jak zmiana trybu pracy (RUN/STOP) lub zmiana wartości zmiennych w sterowniku wymaga od użytkownika zalogowania się jako „admin”. Pole do logowania znajduje się w lewym górnym rogu strony.



Aby zalogować się, jako administrator należy wpisać login „admin” oraz hasło.

Name: admin

Password: brak lub skonfigurowane hasło CPU (dla CPU zabezpieczonych hasłem).

Ustawianie hasła jest opisane w rozdziale 5.5. Ustawianie hasła CPU.

**Standardowe strony web sterownika SIMATIC S7-1200**

Web server w sterowniku S7-1200 oferuje wiele informacji o sterowniku poprzez zintegrowane standardowe strony web. Poszczególne standardowe strony wymienione są poniżej:

Oznaczenie	Zawartość
intro	Strona wprowadzająca
Start Page	Strona startowa oferuje przegląd ogólnych informacji o CPU, nazwie CPU, typie i podstawowych informacjach o CPU oraz o aktualnym trybie pracy
Identification	Wyświetla statyczne informacje identyfikujące sterownik takie jak numer seryjny, numer zamówieniowy oraz informację o wersji firmware
Diagnostic Buffet	Wyświetla bufor diagnostyczny z ostatnimi wpisami
Module Information	Wyświetla informacje na temat komponentów stacji, pokazuje błędy na modułach lub informacje o tym czy moduły są na swoim miejscu
Communication	Wyświetlanie informacji o połączeniach komunikacyjnych otwartej komunikacji, zasobach oraz adresach
Variable Status	Wyświetla statusy zmiennych w programie użytkownika, daje możliwość podglądu i zmiany ich wartości
Data Logs	Archiwum zmiennych w formacie CSV – możliwość zapisania na komputerze. Archiwum jest tworzone za pomocą instrukcji data log w programie użytkownika
User Pages	Dostęp do strony użytkownika

### 3.3 Podstawy dotyczące stron użytkownika

Poniższe rozdziały dostarczają podstawowej wiedzy na temat stron użytkownika w relacji do aplikacji.

Powiązane informacje są dostępne w helpie online SIMATIC STEP 7 V11 SP2 pod hasłem „WWW” (SFC 99).

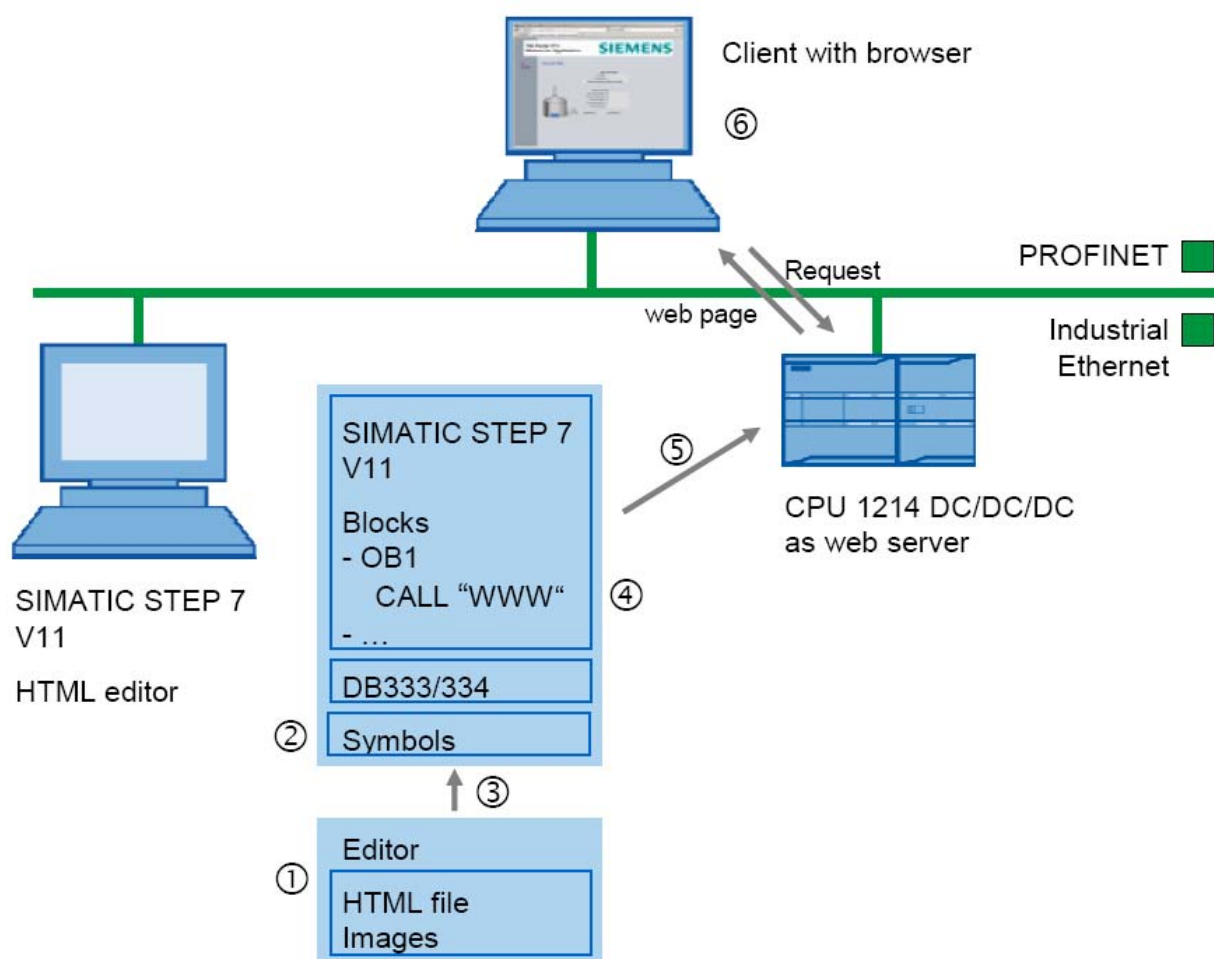
#### Korzyści

Możliwość tworzenia własnych stron internetowych na sterowniku jest korzystne, jeżeli nie jest wymagany rozbudowany stały system HMI, a informacje diagnostyczne i wizualizacja są potrzebne sporadycznie.

Rozwiązanie z użyciem AWP jest rozsądne przy tworzeniu prostych wizualizacji tworzonych indywidualnie według wymagań danej aplikacji.

#### 3.3.1 Tworzenie stron użytkownika

Rysunek 3-1





**Procedura**

Nr	Instrukcja
1	Utworzenie pliku HTML w edytorze HTML. Cała strona internetowa może składać się z kilku pojedynczych plików źródłowych: *.html, *.png, *.js, *.css, itd. Aby mieć dostęp do zmiennych w CPU należy wykorzystać komendy AWP
2	Przypisanie nazwy symbolicznej do zmiennej która ma zostać wykorzystana na stronie internetowej
3	Generacja bloków (Web Control DB oraz dodatkowe bloki DB) za pomocą STEP7 wykorzystując pliki źródłowe. Numery DB mogą być dowolne (domyślnie: DB333 oraz powyżej DB334). Bloki DB są przechowywane w „Program blocks > system blocks > Web Server” w drzewie projektu. Te bloki danych składają się z bloków które sterują wyświetlaniem strony oraz bloków które przechowują stronę internetową.
4	Program tworzony jest przy użyciu STEP7. Synchronizacja pomiędzy programem i web serwerem następuje poprzez wywołanie w programie instrukcji WWW (SFC99)
5	Transfer wszystkich bloków do CPU
6	Otwarcie przeglądarki internetowej wpisując adres IP sterownika „http://www.xx.yy.zz” lub „https://www.xx.yy.zz”

Dostęp do Web serwera w CPU jest możliwy niezależnie od konfiguracji komputera. Każde urządzenie z zainstalowaną przeglądarką internetową oraz interfejsem ethernetowym może wyświetlić stronę www.

Aby mieć możliwość zmiany wartości zmiennych w sterowniku, trzeba być zalogowanym jako administrator.

### 3.3.2 Wymagane bloki – WWW (SFC99)

Za pomocą instrukcji „WWW (SFC99) CPU może zinterpretować bloki danych i użyć ich do wyświetlenia strony internetowej.

#### Bloki DB sterujące oraz DB przechowujące stronę użytkownika

Podstawą własnej strony użytkownika jest plik HTML (lub kilka połączonych plików HTML z obrazami).

Sterujący blok danych DB (domyślnie DB333) zawiera:

- zmienne statusowe i sterujące strony internetowej
- status komunikacji
- informacje o błędach

Oprócz sterującego bloku danych są również bloki DB zawierające dane strony internetowej (domyślnie powyżej DB334). Te bloki danych zawierają zakodowane strony internetowej oraz dodatkowe pliki (np. obrazy).

Wszystkie sterujące bloki danych są zlokalizowane w „Program blocks > System blocks > Web server”.

Rozmiar pamięci potrzebnej do przechowania strony użytkownika determinuje rozmiar pamięci sterownika. Rozmiar pamięci jest ograniczony przez pamięć load.

## UWAGA

Jeżeli zachodzi potrzeba zredukowania przestrzeni pamięci przeznaczonej na stronę użytkownika, można usunąć zbędne grafiki

#### Typowe użycie zmiennych

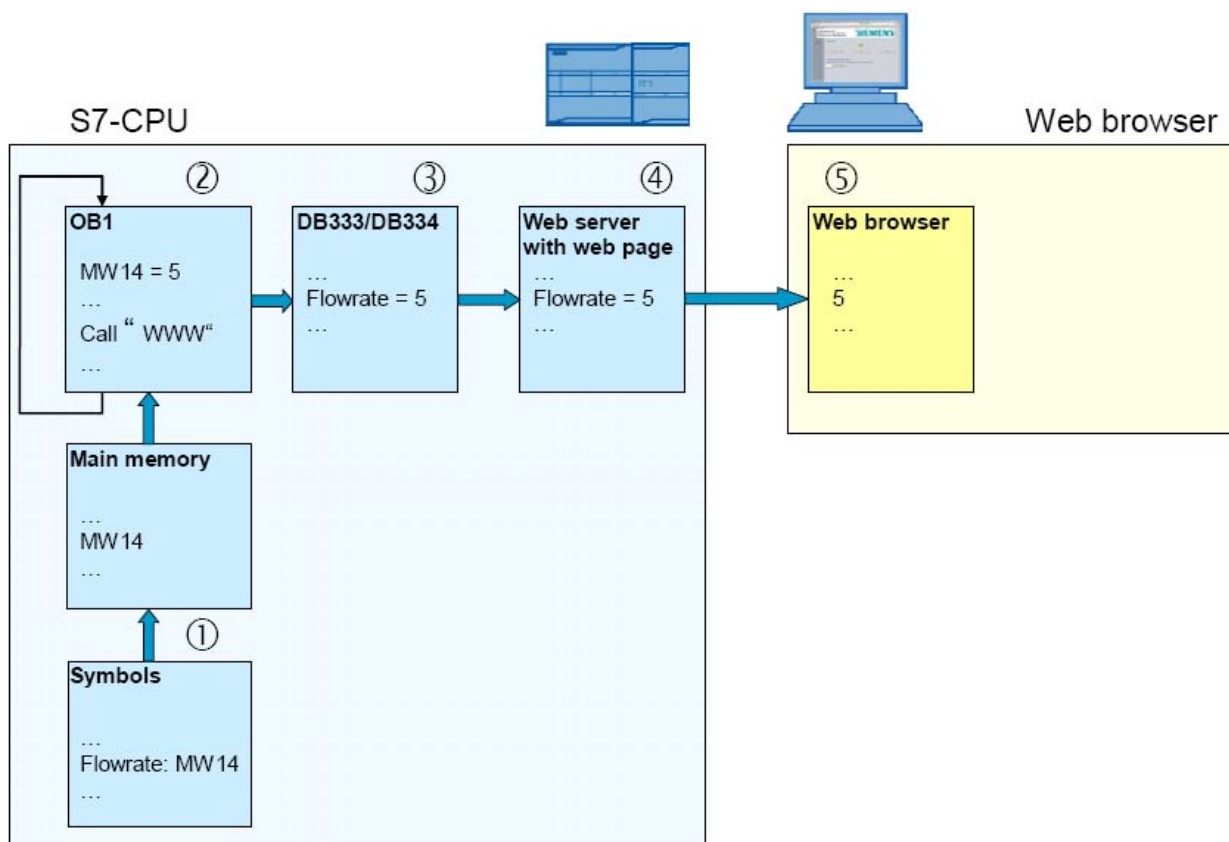
W tabeli poniżej znajduje się przegląd najczęściej używanych zmiennych:

Reprezentacja	Funkcja	Przykład	Informacje
<code>:= "&lt;Name&gt;"</code>	Wyświetla zmienną CPU	<code>:= "TankLevelMinimum"</code> :	Rozdz. 3.4
<code>&lt;!-- AWP_In_Variable Name = "&lt;Name&gt;" --&gt;</code>	Konfiguracja umożliwiająca zapis zmiennej w CPU poprzez oddzielne metody „POST”	<code>&lt;!-- AWP_In_Variable Name = "OpenValve" - --&gt;</code>	Rozdz. 3.5
<code>&lt;!-- AWP_Enum_Ref Name = "&lt;Name&gt;" Enum = "&lt;Variable&gt;" --&gt;</code>	Przypisanie tekstu do wartości zmiennej	<code>&lt;!-- AWP_Enum_Ref Name = "Alarm" Enum = "AlarmValue" -- &gt;</code>	Rozdz. 3.6

### 3.4 Wyświetlanie zmiennych CPU na stronie internetowej

#### 3.4.1 Interakcje między przeglądarką internetową a CPU

Rysunek 3-2



nr	Opis
1.	Zmienne które są odczytywane bądź zapisywane ze strony muszą mieć nazwę symboliczną. Na przykład zmienne w bloku danych „Nazwa_DB”.nazwa_zmiennej
2.	Instrukcja „WWW” (SFC99) musi być wywołana w programie S7
3.	Poprzez wywołanie funkcji „WWW” zostaje zainicjowany blok sterujący (domyślnie: DB333)
4.	Web serwer w CPU konwertuje dane za pomocą informacji zawartych w bloku DB sterującym (domyślnie: DB333) do formatu (=web page) który będzie możliwy do interpretacji przez przeglądarkę. Strona internetowa wywoływana jest w przeglądarce poprzez adres IP sterownika
5.	Przy każdym wywołaniu z przeglądarki, strona www jest odświeżana (ręcznie lub automatycznie). Informacje na temat odświeżania strony można znaleźć w rozdziale 3.1.5 Automatyczne odświeżanie strony. Wywołanie na stronie internetowej można wykonać również poprzez metodę „POST” podczas zapisu zmiennej do CPU. Po wysłaniu strony, cała zawartość zostaje odświeżona

### 3.4.2 Wymagania

Aby było możliwe wyświetlanie zmiennych CPU na stronie internetowej, muszą być spełnione następujące wymagania

Program S7	Plik HTML
<ul style="list-style-type: none"> <li>- Każda zmienna musi mieć przypisaną zmienną symboliczną. Zmienna może być wyświetlona na stronie internetowej wywołana tylko przez nazwę symboliczną.</li> <li>- Instrukcja „WWW” (SFC99) musi być wywołana w cyklu programu</li> <li>- Dozwolone są standardowe typy danych (dane typu DTL nie są dozwolone), typy danych użytkownika oraz struktury</li> </ul>	<ul style="list-style-type: none"> <li>- Nie jest konieczne deklarowanie zmiennych poprzez komendę AWP w pliku HTML</li> </ul>

### 3.4.3 Wymagane bloki – WWW (SFC99)

#### Program S7:

Nie ma potrzeby programowania czegokolwiek w programie S7

#### Plik HTML:

Zmienna może być wyświetlona w dowolnej pozycji w pliku HTML. Składnia jest następująca:  
`:= "<Variable>" :`

Przykład zmiennej „TankLevelMaximum”

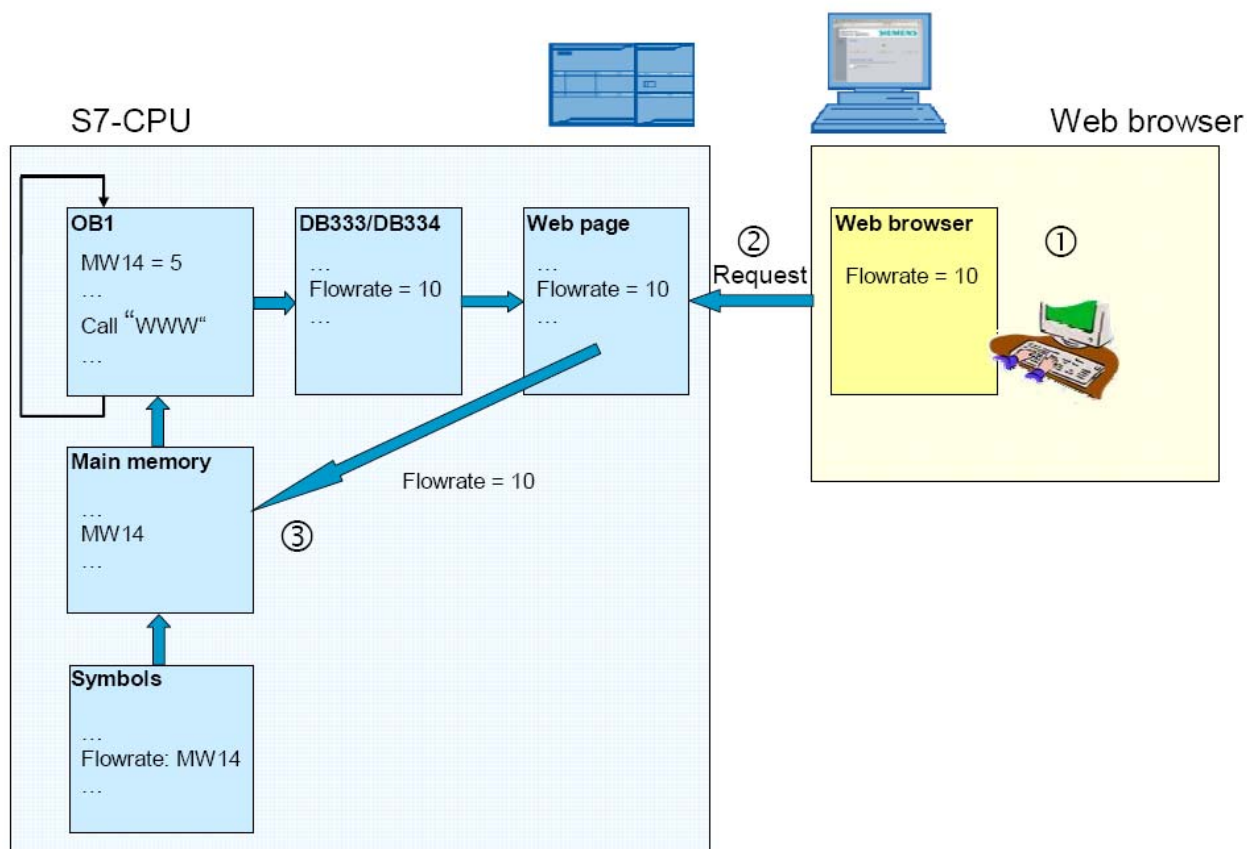
`<p>:= "TankLevelMaximum" : </p>`

Wyświetlanie zmiennej jest niezależne od jej typu. Odświeżanie zmiennych jest opisane w rozdziale 3.1.5.

### 3.5 Zapis zmiennych CPU ze strony internetowej

#### 3.5.1 Interakcje między przeglądarką internetową a CPU

Rysunek 3-3



Nr	Opis
1	Poprzez stronę internetową użytkownik zmienia wartość zmiennej przepływ „Flowrate” na „10”.
2	Przeglądarka internetowa raportuje zapytanie (metoda „POST”)
3	Program S7 akceptuje zmienioną wartość zmiennej „Flowrate”, wyświetlanie zmiennej w przeglądarce jest odświeżone i nowa wartość zostaje wyświetlona.

### 3.5.2 Wymagania

Aby zapisywanie zmiennych w CPU przez przeglądarkę internetową było możliwe muszą zostać spełnione następujące wymagania:

Program S7	Plik HTML
<ul style="list-style-type: none"> <li>- Każda zmienna musi mieć przypisaną zmienną symboliczną. Zmienna może być adresowana tylko przez nazwy symboliczne</li> <li>- Instrukcja „WWW” (SFC99) musi być wywołana w cyklu programu</li> <li>- Dozwolone są standardowe typy danych (dane typu DTL nie są dozwolone), typy danych użytkownika oraz struktury</li> </ul>	<ul style="list-style-type: none"> <li>- Zmienne muszą zostać zadeklarowane poprzez komendę AWP &lt;!-- AWP_In_Variable ...--&gt;</li> <li>- Zmienne muszą zostać transferowane do CPU (np. metoda „POST” w pliku HTML)</li> </ul>

### 3.5.3 Procedura

#### Program S7:

Instrukcja „WWW” (SFC99) musi być wywoływana cyklicznie

#### Plik HTML:

Komenda AWP dzięki której zmienna może być zapisana do CPU jest następująca:

```
<!-- AWP_In_Variable Name="Variable" -->
```

Przykład zapisania zmiennej Przepływ "Flowrate":

```
<!-- AWP_In_Variable Name="Flowrate" -->
```

Komenda AWP typowo umieszczana jest na początku pliku HTML.

**Transfer zmiennych z przeglądarki**

Transfer zmiennych między przeglądarką i web serwerem odbywa się za pomocą metody „POST”. Forma składa się z dwóch jednostek:

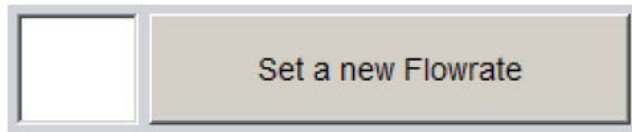
-Pole do wpisania wartości

Pole jest nazwane poprzez zmienną np.

```
<!-- AWP_In_Variable Name='Flowrate' -->.
```

-Przycisk za pomocą którego wpisanie wartości jest potwierdzane.

Przykład:

A screenshot of a web form. It consists of a small, empty text input field on the left, followed by a rectangular button with a light gray background and a thin border. The button contains the text "Set a new Flowrate" in a dark, sans-serif font.

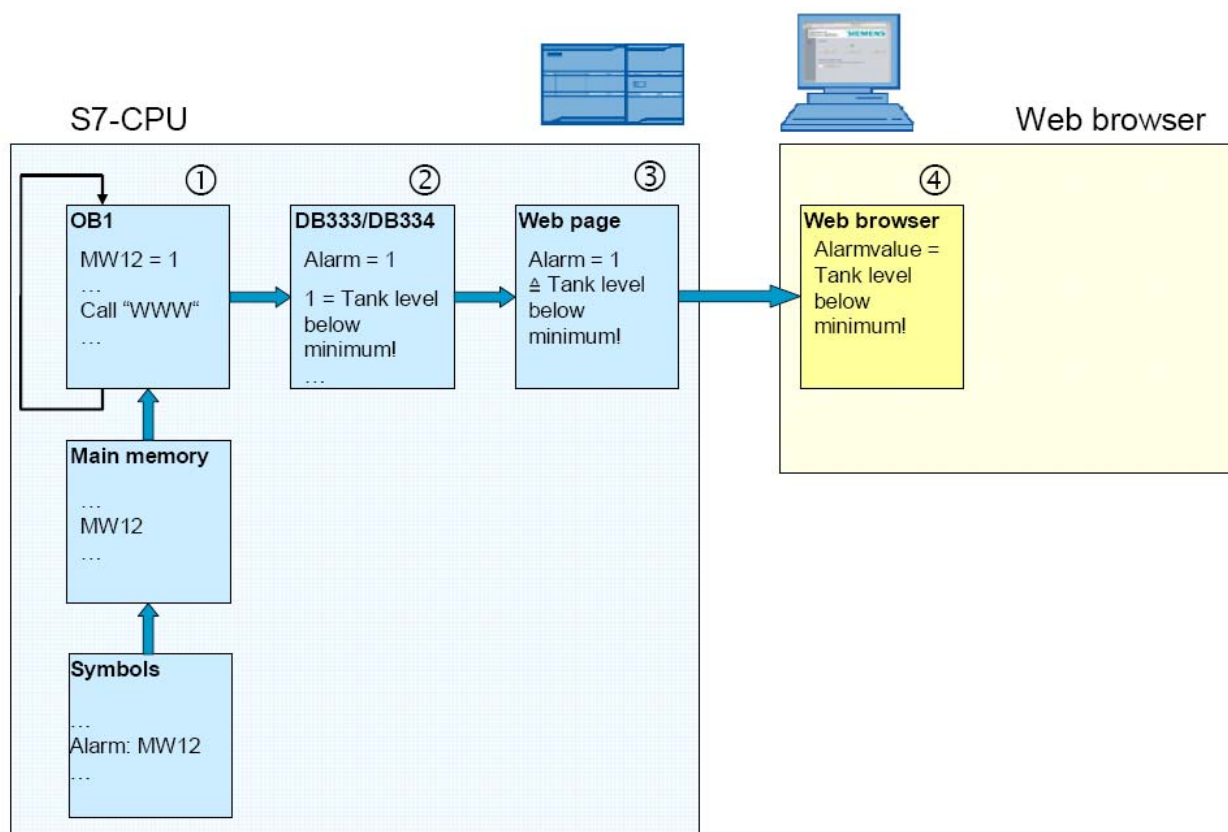
Kod:

```
<form method="post" action="" onsubmit="return check();" >
<input type="text" name="Flowrate" size="2">
<input type="submit" value="Set a new Flowrate">
</form>
```

### 3.6 Powiązanie zmiennych z tekstem w pliku HTML

Czasami jest uzasadnione aby informacje na stronie internetowej były wyświetlone bezpośrednio jako tekst. Jest możliwe połączenie między zmiennymi w sterowniku i tekstem. Tekst może być w kilku językach. Opisywana aplikacja zawiera tekst w jednym języku. Poniższy rysunek obrazuje interakcje między sterownikiem a przeglądarką.

Rysunek 3-4



Nr	Opis
1.	Program S7 wywołuje instrukcję „WWW” (SFC99) i ustawia wartość MW12 („Alarm”) na „1”.
2.	Poprzez cykliczne wywołanie instrukcji „WWW” (SFC99) zmienna „Alarm” w blokach DB333/DB334 są również odświeżane
3.	Web serwer tworzy powiązanie między zmienną „Alarm” i tekstem
4.	W przeglądarce wyświetlany jest tekst zamiast wartości zmiennej „Alarm”



### 3.6.1 Wymagania

Aby było możliwe wyświetlanie tekstu w miejsce zmiennej muszą być spełnione następujące wymagania:

Program S7	Plik HTML
<ul style="list-style-type: none"> <li>- Każda zmienna musi mieć przypisaną zmienną symboliczną. Zmienna może być adresowana tylko przez nazwy symboliczne</li> <li>- Instrukcja „WWW” (SFC99) musi być wywołana w cyklu programu</li> <li>- Dozwolone są standardowe typy danych (dane typu DTL nie są dozwolone), typy danych użytkownika oraz struktury</li> </ul>	<ul style="list-style-type: none"> <li>- Zmienne muszą zostać zadeklarowane poprzez komendę AWP &lt;!-- AWP_In_Variable ...--&gt;</li> <li>- Zmienne muszą zostać transferowane do CPU (np. metoda „POST” w pliku HTML)</li> </ul>

#### Tworzenie typów ENUM

Komenda AWP, poprzez którą definiowane są typy ENUM jest następująca:

```
<!-- AWP_Enum_Def Name= "<Name Enum type>"
Values='0:"<Text_1>", 1:"<Text_2>", ... , x:"<Text_x>"' -->
```

Przykład dla zmiennej „AlarmValue”:

```
<!-- AWP_Enum_Def Name="AlarmValue" Values='0:"Tank empty!",
1:"Tank level below minimum!", 2:"Tank level between minimum
and midth!", 3:"Tank level between midth and maximum!",
4:"Tank level over maximum!", 5:"Tank level overflow!"' -->
```

Typowo komenda AWP znajduje się na początku pliku HTML lub znajduje się w odrębnym pliku HTML.

#### Przypisywanie typów ENUM

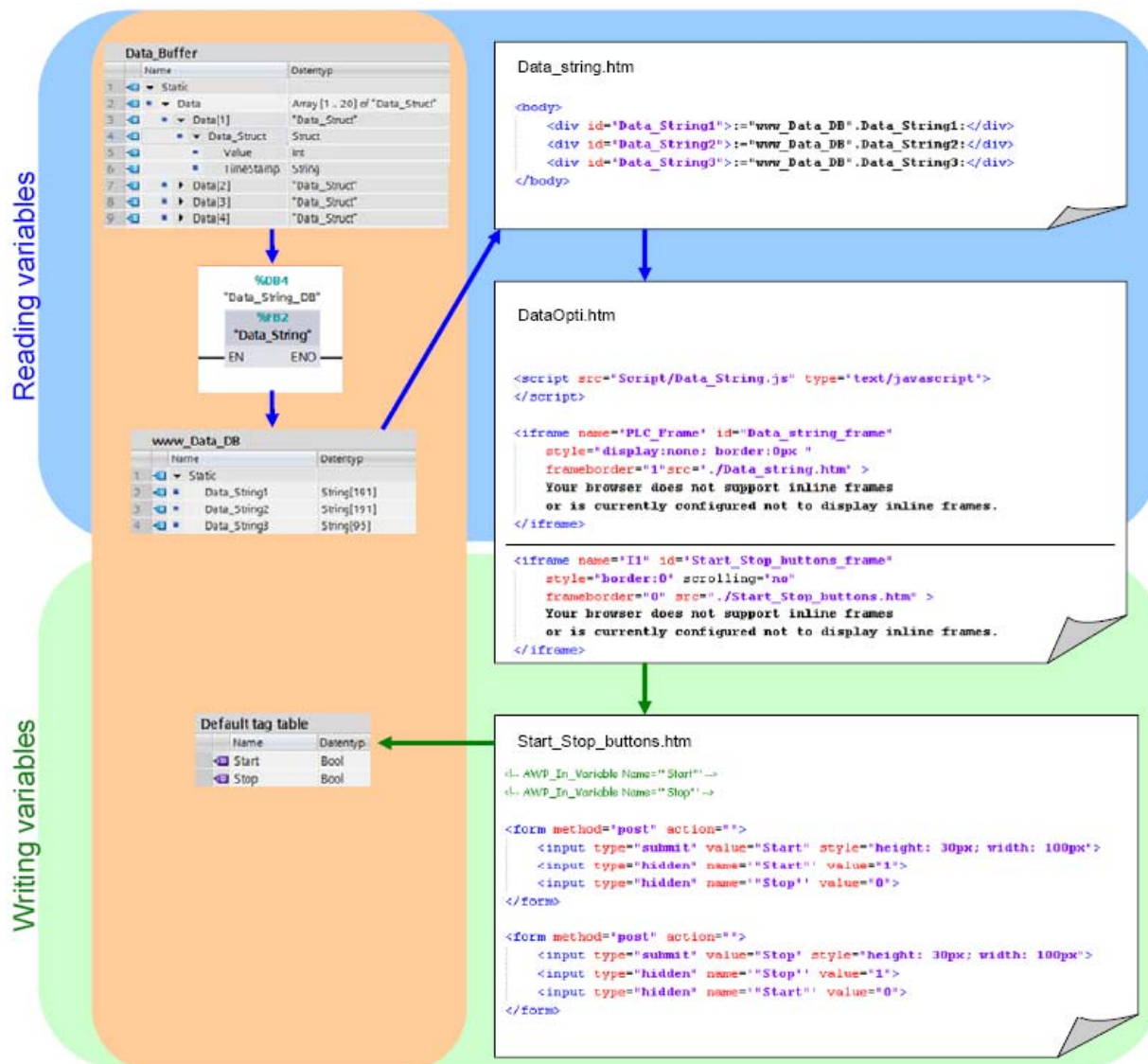
Składnia służąca wyświetlaniu tekstu zamiast wartości zmiennej jest następująca (na przykładzie zmiennej „Alarm”):

```
<!-- AWP_Enum_Ref Name=' "Alarm" ' Enum="AlarmValue" -->
:="Alarm":
```

### 3.7 Tworzenie optymalnych czasowo stron użytkownika (opcja)

Tworzenie optymalnych czasowo stron HTML nie jest opisane w manualu S7-1200. Założenie przedstawia poniższy rysunek:

Rysunek 3-5 Czytanie/zapisywanie zmiennych



Nr	Opis
1.	Opisywana jest strona użytkownika (poniżej nazywana stroną główną). W tej aplikacji jest to strona „DataOpti.htm”. Nie jest ona odświeżana toteż nie ma odpowiednich instrukcji w kodzie html.
2.	Strona główna nie zawiera zmiennych i dlatego nie posiada bezpośredniego dostępu do zmiennych sterujących. Wszystkie zmienne sterujące, które mają być czytane lub zapisywane są umieszczone w dodatkowych plikach HTML (tutaj „Data_string.htm” oraz „Start_Stop_buttons.htm”), które zawarte są na stronie głównej jako ramki (iframe) i z tego powodu nie ma potrzeby okresowego odświeżania strony internetowej. Tylko poszczególne ramki są odświeżane
3.	Kolejnym ważnym krokiem jest minimalizacja liczby zmiennych sterujących które mają być odczytywane. W programie użytkownika w sterowniku (tutaj FB „Data_String”) indywidualne zmienne sterujące są oddzielone przecinkami, zapisywane do ciągu wyjściowego (string) i dopiero przesyłane jako jedna

	zmienna do web serwera co pozwala zaoszczędzić bardzo dużo czasu. Ciąg jest umieszczany jako zmienna wyjściowa w oddzielnym pliku html (tutaj „Data_String.htm”), który wyświetla się jako wbudowana ramka (iframe, patrz punkt 2 tabeli) na stronie głównej. Tutaj ciąg znaków jest wyłącznie dostarczany do dalszej obróbki przez JavaScript (tutaj „Data_String.js”).
4.	Dla zmiennych wejściowych (zobacz rozdział 3.5) utworzone są formy które umieszczone są w oddzielnych plikach HTML (tutaj „Start_Stop_buttons.htm”). Pliki HTML są umieszczone na stronie głównej jako ramki (iframes, zobacz punkt 2 tabeli). Możliwe jest zapisanie kilku ramek na stronie HTML
5.	Strona główna umożliwia odniesienie do pliku JavaScript (tutaj „Data_String.js”), Który wykonuje poniższe zadania cyklicznie (w poniższym przykładzie, co 500ms): <ul style="list-style-type: none"> <li>- Dzielenie ciągu wyjściowego (zobacz pkt 3 tabeli) na poszczególne zmienne za pomocą metody Split („,”), które równolegle są przechowywane w tabeli wyjściowej jako ss_values</li> <li>- Dostarczenie elementów wyjściowych (tutaj: komórki tabeli), strony głównej ze zmiennymi sterującymi. Które są dostarczane jako elementy ss_values tabeli wyjściowej.</li> <li>- Odświeżanie ramek (zobacz pkt 3 tabeli) które zawierają ciąg wyjściowy za pomocą metody reload(0).</li> </ul>

Rysunek 3-6 Wyciągnięte ze skryptu „Data\_String.js”

```

setTimeout("init_plc()",1500);

function init_plc()
{
    setInterval("plc_update()",500);
}

function plc_update()
{
    var iframe = document.getElementById('Data_string_frame');
    if (iframe.readyState == 'complete')
    {
        var Data_string1 = window.frames['Data_string_frame'].document.getElementById('Data_String1').innerHTML;
        var ss_values = Data_string1.split(",");

        document.getElementById('Table1').rows[1].cells[0].innerHTML = ss_values[0];
        document.getElementById('Table1').rows[1].cells[1].innerHTML = ss_values[1];
        ...

        parent.frames['Data_string_frame'].window.location.reload();
    }
}

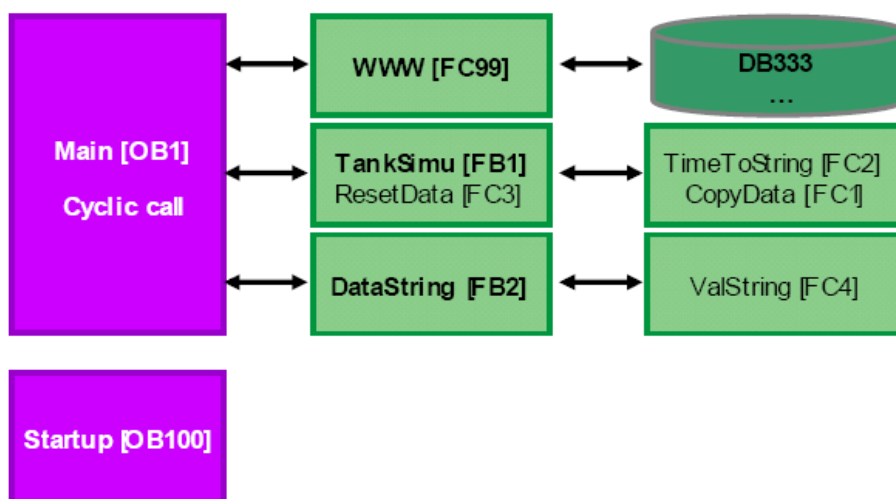
```

## 4 Mechanizmy funkcjonalne aplikacji

### 4.1 Podstawy funkcjonowania programu S7

Program S7 dla tej aplikacji służy wyłącznie do reprezentacji ogólnych zasad funkcji STEP7. Struktura wywołań w programie S7 jest następująca:

Rysunek 4-1



Następujące symbole i zmienne są w bloku danych „Data\_Buffer”:

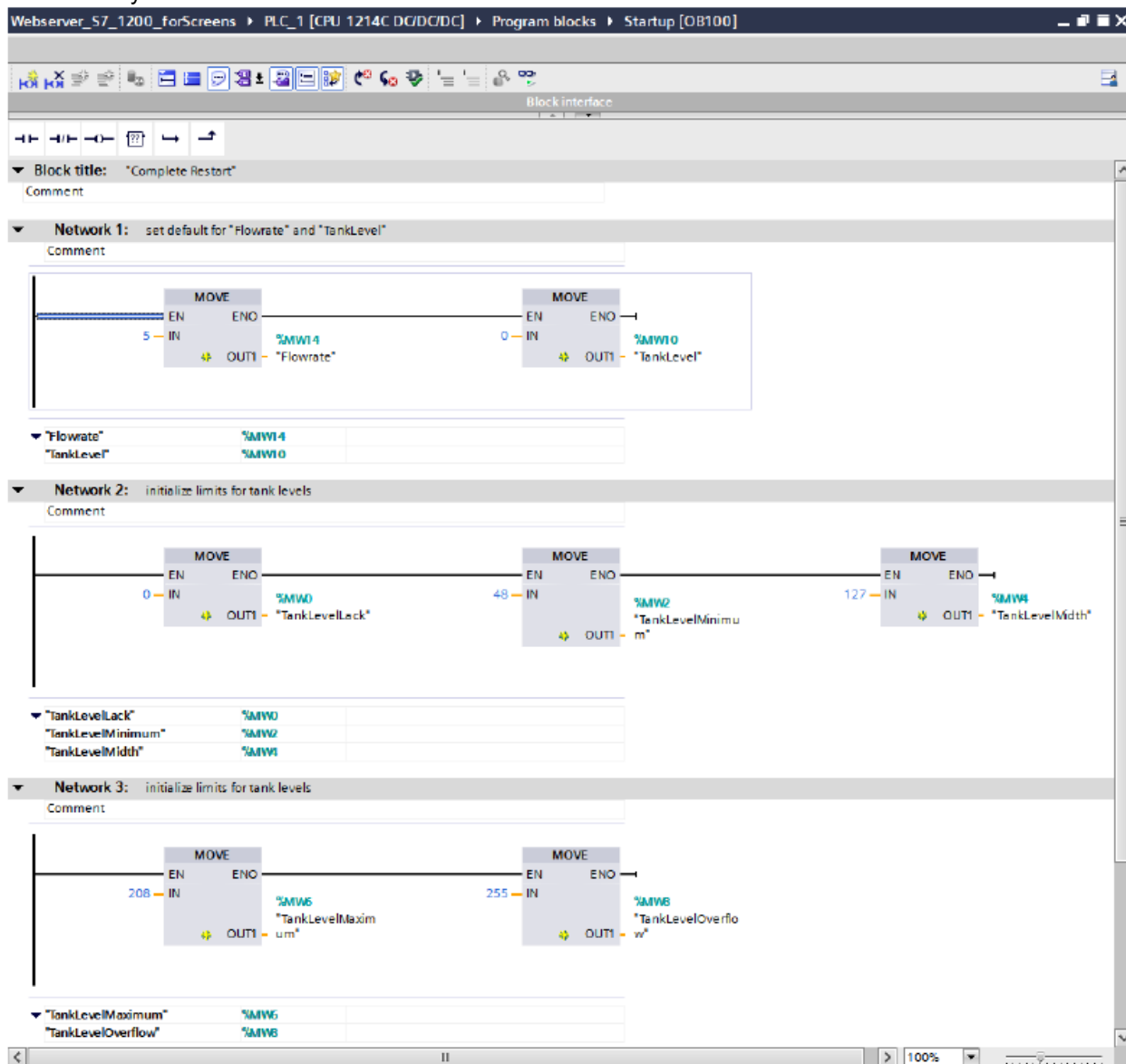
Rysunek 4-2

... DC/DC/DC > PLC tags > Default tag table [32]					
Tags User constants System constants					
Default tag table					
	Name	Data type	Address	Retain	
1	TankLevelLack	Int	%MW0	<input type="checkbox"/>	▲
2	TankLevelMinimum	Int	%MW2	<input type="checkbox"/>	☰
3	TankLevelMidth	Int	%MW4	<input type="checkbox"/>	
4	TankLevelMaximum	Int	%MW6	<input type="checkbox"/>	
5	TankLevelOverflow	Int	%MW8	<input type="checkbox"/>	
6	TankLevel	Int	%MW10	<input type="checkbox"/>	
7	Alarm	Int	%MW12	<input type="checkbox"/>	
8	Flowrate	Int	%MW14	<input type="checkbox"/>	
9	StatusValveCPU	Bool	%M16.0	<input type="checkbox"/>	
10	CloseValve	Bool	%M16.1	<input type="checkbox"/>	
11	OpenValve	Bool	%M16.2	<input type="checkbox"/>	
12	Start	Bool	%M17.0	<input type="checkbox"/>	
13	Stop	Bool	%M17.1	<input type="checkbox"/>	
14	Reset	Bool	%M17.2	<input type="checkbox"/>	
15	StartStop	Bool	%M17.3	<input type="checkbox"/>	
16	Status_String	Int	%MW18	<input type="checkbox"/>	
17	ActFlowrate	Int	%MW22	<input type="checkbox"/>	
18	TankLevelScal	Int	%MW24	<input type="checkbox"/>	▼

### 4.1.1 Blok startowy (OB100)

W bloku startowym „Startup” (OB100) ustawiana jest wartość przepływu „Flowrate” oraz wartości graniczne przechowywanych zmiennych.

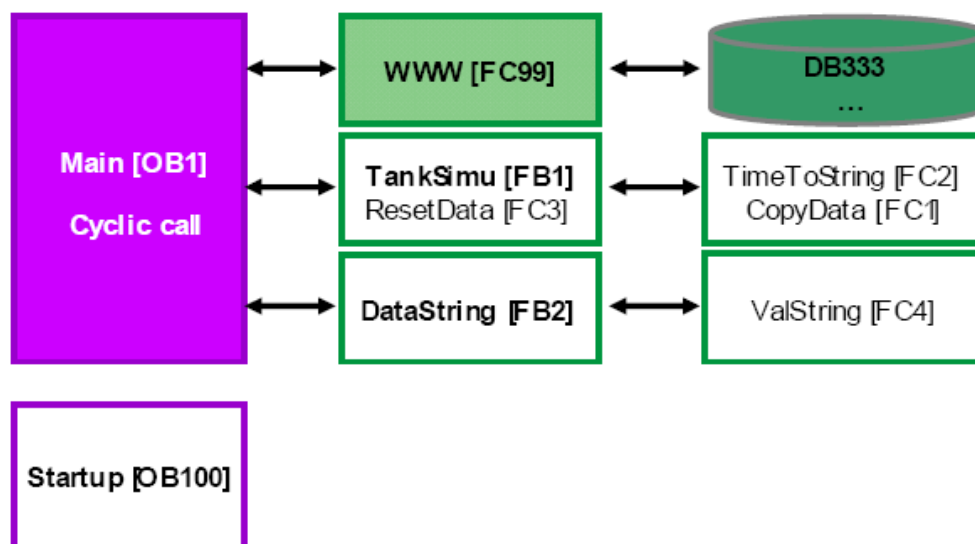
Rysunek 4-3



#### 4.1.2 Main (OB1)

W bloku głównym „Main” (OB1) status DB333 jest odpytywany cyklicznie aby było możliwe wychwycenie zapytania z web serwera. Powodem wysłania zapytania może być na przykład zmiana wartości zmiennej w przeglądarce internetowej która ma być wysłana do web serwera.

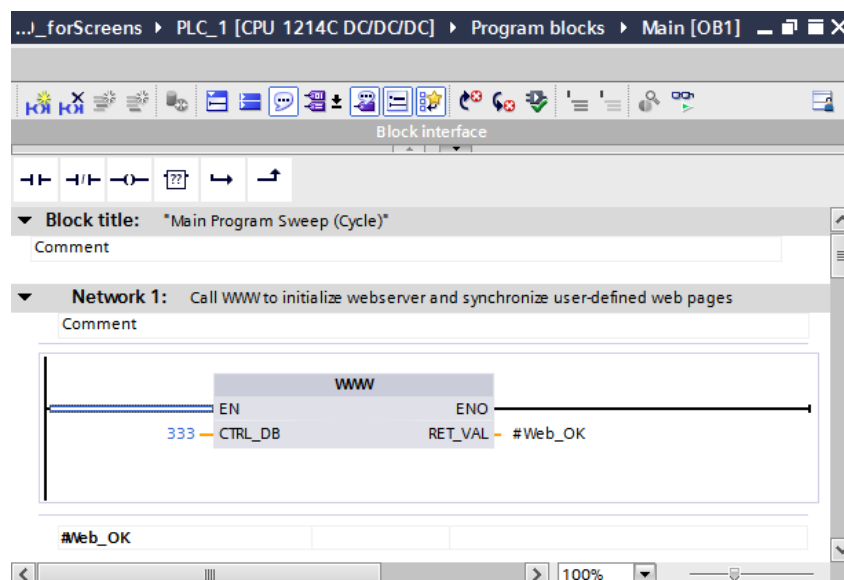
Rysunek 4-4



#### Synchronizacja stron użytkownika

Instrukcja „WWW” (SFC99) inicjalizuje web serwer w CPU. Informacja o błędach jest wystawiona przez “RET\_VAL”.

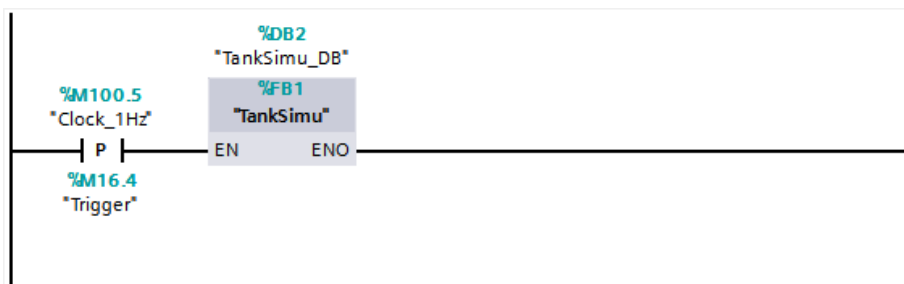
Rysunek 4-5



### Wywołanie symulacji zbiornika

Aby mieć pewność że napełnianie lub opróżnianie zbiornika nie następuje zbyt szybko, funkcja „TankSimu” jest wywołana w OB1 raz na sekundę.

Rysunek 4-6



### Odpytywanie przycisków „Start”, „Stop”, oraz „Reset”

Status przycisków „Start” oraz „Stop” jest sprawdzany przez stronę internetową. Jeżeli jeden z przycisków został wciśnięty, status jest zapisany w zmiennej PLC „StartStop”. Dodatkowo sprawdzany jest status przycisku „Reset”. Poprzez kliknięcie „Reset”, zdefiniowana wartość początkowa pozycji zostaje utworzona za pomocą funkcji „ResetData”.

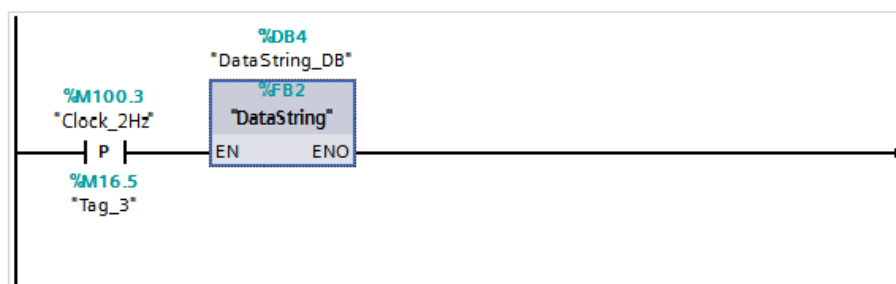
### Odpytywanie przycisków „OpenValve” oraz „CloseValve”

Dodatkowo odpytywany jest status przycisków „OpenValve” oraz „CloseValve”. Jeżeli jeden z nich zostanie wciśnięty, status zostaje zapisany w zmiennej „StatusValveCPU”.

### Wywoływanie bloku funkcyjnego „DataString”

Odświeżanie zmiennych typu string jest wywoływane dwa razy na sekundę. Odnosi się to do czasu odświeżania w Javascript.

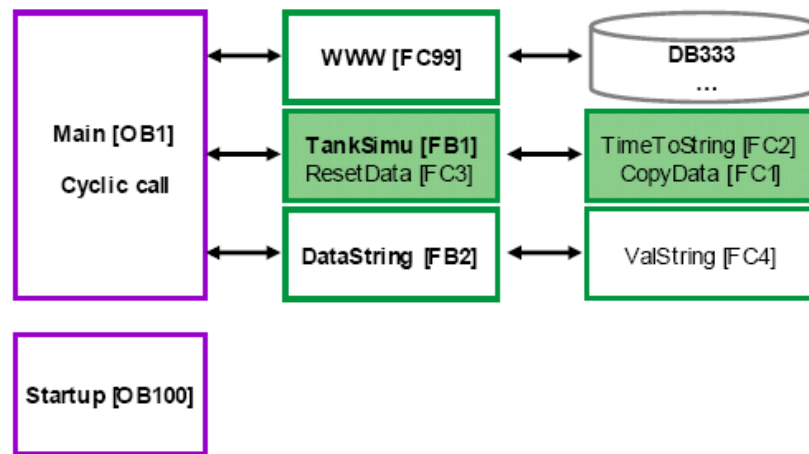
Rysunek 4-7



#### 4.1.3 TankSimu (FB1)

##### Funkcjonalność FB1

Rysunek 4-8



W FB1, symulowane jest napełnianie lub opróżnianie zbiornika, w zależności od ustawionego przepływu i pozycji zaworu.

Blok jest wywoływany tylko raz na sekundę.

Użytkownik może zdefiniować wartość przepływu poprzez zmienną „Flowrate” na stronie internetowej.

Poziom napełnienia zbiornika jest zwiększany lub zmniejszany poprzez wywołanie FB1.

Aktualna wartość poziomu napełnienia jest przechowywana w zmiennej PLC „TankLevel”.

Poprzez dwie zmienne PLC „OpenValve” i „CloseValve”, pozycja zaworu jest odczytywana, a następnie zapisywana w zmiennej PLC „StatusValveCPU”.

W zależności od poziomu napełnienia zbiornika wyświetlone zostaje:

- Zbiornik został całkowicie opróżniony (TankLevelLack)
- Minimalny poziom zbiornika (TankLevelMinimum)
- Zbiornik jest napełniony w połowie (TankLevelMidth)
- Zbiornik jest pełny (TankLevelMaximum)
- Przepełnienie zbiornika (TankLevelOverflow)

Poprzez zmienną „Alarm”, poziom napełnienia zbiornika wyświetlany jest jako tekst.

##### Status StartStop

Tylko kiedy zmienna „StartStop” jest ustawiona, poziom napełnienia zbiornika zmienia się i zmienna są zapisywane do bufora.

##### Status zaworu

Poprzez bit „StatusVentilCPU” zapamiętane jest ostatnie wciśnięcie przycisku (OpenValve lub Close Valve)

W zależności od stanu tego bitu, zbiornik jest napełniany lub opróżniany.

##### Napełnianie zbiornika

Napełnianie zbiornika rozpoczyna się od sprawdzenia czy jest napełniony.

Jeżeli zbiornik nie jest pełny, poziom napełnienia zbiornika jest zwiększany z ustawionym przepływem. Poziom napełnienia zbiornika ograniczony jest poprzez zmienną „TankLevelOverflow”



**Opróżnianie zbiornika**

Opróżnianie zbiornika jest analogiczne do napełniania. Poziom napełnienia zbiornika jest zmniejszany zgodnie z ustawionym przepływem i jest ograniczony wartością 0.

**Alarm status**

Następnie poziom napełnienia zbiornika porównywany jest z wartościami limitów napełnienia zbiornika.

W zależności od osiągnięcia poziomu napełnienia, wartości od 0 do 5 są zapisywane do zmiennej „Alarm”. Na podstawie przypisanej wartości do zmiennej „Alarm”, na stronie internetowej wyświetlany jest odpowiedni tekst alarmu.

**Zapisywanie danych w buforze**

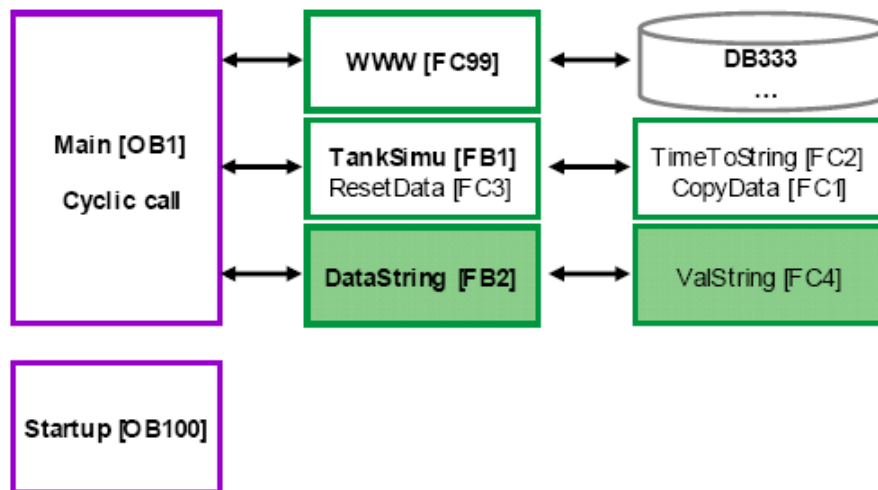
Aktualny poziom napełnienia zbiornika jest zapisywany ze znacznikiem czasowym (string) w buforze pierścieniowym. Przechowywanych jest 20 par wartości.

Za pomocą funkcji TimeToString, znacznik czasowy jest zapisywany do zmiennej typu string. Za pomocą funkcji „CopyData” pary wartości są kopiowane do bufora pierścieniowego.

#### 4.1.4 DataString (FB2)

##### Funkcjonalność FB2

Rysunek 4-9



W FB2 pary zmiennych (stempel czasowy, wartość) są zapisywane w buforze pierścieniowym, oddzielone przecinkami i zapisane jako zmienna typu string. Zmienna typu string może mieć długość 254 znaków, tak więc muszą być wykorzystane trzy zmienne typu string i przesłane do servera jako pojedyncze zmienne.

Stempel czasowy jest już zapisany jako string. Poziom zbiornika (wartość) jest przekonwertowana do typu string za pomocą funkcji „Valstring”.

Blok jest wywoływany tylko dwa razy na sekundę.

## 4.2 Podstawy funkcjonowania pliku HTML

Poniższy rozdział dostarcza szczegółowych wyjaśnień dotyczących poszczególnych sekcji pliku HTML. W celu stworzenia strony internetowej zostały użyte stałe wartości dla ustalenia pozycji i wymiarów elementów. Zapobiega to rozjeżdżaniu się i zachodzeniu na siebie elementów podczas zmiany wymiarów okna.

### 4.2.1 Komendy AWP

#### Podstawy

Komendy AWP umieszczone są w pliku HTML jako komentarze. Mogą być umieszczone w dowolnym miejscu pliku. Jednakże w celu zwiększenia przejrzystości zaleca się umieszczać je na początku pliku HTML.

Rysunek 4-10

```
<!-- AWP_In_Variable Name="Start" -->
<!-- AWP_In_Variable Name="Stop" -->
<!-- AWP_In_Variable Name="Reset" -->
<!-- AWP_In_Variable Name="Flowrate" -->
```

#### Wyjaśnienia

Kod	Wyjaśnienie
<pre>&lt;!-- AWP_In_Variable Name="Start" --&gt;</pre>	<p>Wszystkie zmienne transferowane do CPU muszą zidentyfikowane jako AWP_In_Variable.</p> <p>Uwaga: Proszę mieć na uwadze, że cudzysłów jest zagnieżdżony. Zmienna wpisana jest między znakami cudzysłowie i otoczone jest apostrofami.</p>
<pre>&lt;!-- AWP_Enum_Def Name="AlarmValue" Values='0:"Tank empty!", 1:"Tank level below minimum!", 2:"Tank level between minimum and midth!", 3:"Tank level between midth and maximum!", 4:"Tank level over maximum!", 5:"Tank level overflow!"' --&gt;</pre>	<p>Typy ENUM są definiowane poprzez AWP_Enum_Def</p>
<pre>&lt;!-- AWP_Enum_Ref Name="Alarm" Enum="AlarmValue" -- &gt;:="Alarm":</pre>	<p>Typy anum są przypisane do zmiennych z AWP_Enum_Ref.</p>

## 4.2.2 Informacje na temat doctype oraz nagłówka pliku HTML

### Podstawy

Poniższe informacje muszą być uwzględnione w każdym pliku HTML aby był zgodny z HTML. Jedynym wyjątkiem jest tag "<meta http-equiv='refresh' ...>": Jeżeli nie będzie używane automatyczne odświeżanie i zamiast tego używany będzie przycisk F5, można nie używać tego tagu.

Rysunek 4-11

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>

  <head>
    <title>Userdefined Website - Application Example</title>
    <meta http-equiv="Content-Language" content="en" >
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
    <meta http-equiv="Content-Script-Type" content="text/javascript" >
    <meta http-equiv="refresh" content="10; URL=Overview.htm">
    <link rel="stylesheet" type="text/css" href="Stylesheet/siemens_Stylesheet.css"/>
    <script src="Script/siemens_script.js" type="text/javascript"> </script>
  </head>

  <body>
    ...
  </body>
</html>
```

### Wyjaśnienia

Kod	Wyjaśnienie
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">	Specyfikacja dokumentu HTML: typ dokumentu HTML w wersji języka v4.1 w wariantie „transitional” Kod „EN” określa język Tagów, np. Angielski. Typ dokumentu zawsze jest określony przed tagiem "<html>"
<html> ... </html>	Zawiera treść HTML
<title>Userdefined Website - Application Example</title>	Tytuł strony internetowej, który zostanie później wyświetlony w nagłówku przeglądarki internetowej
<meta http-equiv="Content-Language" content="en" >	Język zawartości pliku
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" >	Za pomocą "content='text/html'" wyspecyfikowano typ MIME, po którym użyto zestawu znaków UTF-8
<meta http-equiv="refresh" content="10; URL=Overview.htm">	Opcjonalne dane meta: z tą komendą strona web jest odświeżana co 10 sekund. Szczególnie dla monitorowania procesu jest wymagane aby strona odświeżana była cyklicznie. Dla stron z polami wejściowymi, odświeżanie stron może być problematyczne. Więcej informacji znaleźć można w rozdziale 3.1.5

<code>&lt;link rel="stylesheet" type="text/css" href="Stylesheet/siemens_Stylesheet.css"&gt;</code>	Poprzez link <link...> zostaje utworzone powiązanie z plikiem CSS który zawiera wszystkie informacje dotyczące wyglądu strony internetowej np. białe tło, itd...
<code>&lt;script src="Script/siemens_script.js" type="text/javascript"&gt; &lt;/script&gt;</code>	Przestrzeń dla skryptów (np. JavaScript) definiowana jest między Tagami <script> oraz </script>.
<code>&lt;body&gt; ...&lt;/body&gt;</code>	Ciało dokumentu

### 4.2.3 Wyświetlanie obszarów

#### Podstawy

W pliku HTML zostały użyte trzy obszary:

- obszar nagłówka (header)
- pasek nawigacyjny (navi)
- obszar danych (page)

#### Wyjaśnienia

Rysunek poniżej przedstawia obszary w pliku HTML

```
<body>

<!-- Header Line -->
    <div id="header">
        ...
    </div>
<!-- Header Line End-->

<!-- Navigation -->
    <div id="navi">
        ...
    </div>
<!-- Navigation End-->

<!-- Data Area -->
    <div id="page">
        ...
    </div>
<!-- Data Area End-->
</body>
```

**Formatowanie obszarów jest definiowany centralnie w oddzielnym pliku CSS**

Rysunek 4-13

```

#header {
    POSITION: absolute;
    width: 950px;
    height: 110px;
    left: 60px;
    top: 20px;
    background-color: rgb(255,255,255);
    z-index: 2;
}

#navi {
    POSITION: absolute;
    left: 0;
    top: 0;
    width: 150px;
    height: 800px;
    padding-top: 180px;
    padding-left: 0px;
    text-align: left;
    border-color: white;
    border-style: solid;
    border-width: 1px;
    background-color: rgb(148,158,170);
    border-collapse : separate;
    z-index: 1;
}

#page {
    POSITION: absolute;
    left: 150px;
    top: 0;
    height: 800px;
    width: 920px;
    padding-top: 180px;
    padding-left: 30px;
    padding-right: 30px;
    text-align: left;
    border-color: white;
    border-style: solid;
    border-width: 1px;
    background-color: rgb(208,211,218);
    border-collapse : separate;
    z-index: 1;
}

```

Kod	Wyjaśnienie
<pre> #page { POSITION: absolute; left: 150px; top: 0; height: 800px; width: 920px; padding-top: 180px; padding-left: 30px; padding-right: 30px; text-align: left; border-color: white; border-style: solid; border-width: 1px; background-color: rgb(208,211,218); border-collapse : separate; z-index: 1; } </pre>	<p>Format CSS ma następującą strukturę:  Selektor {właściwość:wartość}  W tym przykładzie, „page” jest selektorem z  kilkoma deklaracjami (właściwość:wartość):  Więcej informacji na temat formatowania elementu  HTML może być znalezionych w rozdziale 3.1.3</p>

#### 4.2.4 Wyświetlanie obrazów

##### Podstawy

W pliku HTML jest kilka obrazów:

- Obrazy statyczne
- Obrazy tła
- Oraz ze zmienną wysokością
- Obraz dynamiczny który zmieniany jest w zależności od bitu statusowego CPU

##### Wyjaśnienia

Rysunek 4-14

```
<td width="300px"></td>

<td width="250px" height="200px" rowspan="14" valign="bottom" background="Images\TankExample.PNG"
style="background-repeat:no-repeat; background-position:bottom left">


</td>
```

Kod	Wyjaśnienie
	Obraz jest integrowany poprzez tag "img"
background="Images\TankExample.PNG" style="background-repeat:no-repeat; background-position:bottom left"	"background" specyfikuje obraz tła z właściwościami
alt="Level" width="56px" height=":=TankLevelScal:px " style="margin-left:48px; marginbottom:- 3px">	Dla obrazów ze zmienną wysokością takich jak na przykład, wyświetlanie poziomu, zmienna "TankLevelScal" jest określona z „:=” oraz jednostką miary np. „px”. zamiast wartości dla wysokości. W tym przykładzie, „margins” pozycjonuje wyświetlacz w obrazie tła
<img src="Images\Valve:=StatusValveCPU":. png" name="Valve" "StatusValveCPU = 0" id="StatusValveCPU" alt="Valve" style="margin-left:70px; marginright: 2px; margin-bottom:12px">	Ten obraz zależy od zmiennej „StatusValveCPU”. Ta zmienna może przyjąć wartość 0 lub 1. Przechowywane obrazy to Valve0.png (zawór zamknięty) oraz Valve1.png (zawór otwarty). Gdy zawór jest zamknięty „StatusValveCPU” ma wartość 0: wywołanie obrazu realizowane jest przez „Valve” + „0” + „.png”=Valve0.png. Z „alt” można określić tekst który będzie wyświetlony jeżeli obraz nie będzie mógł być wyświetlony. Za pomocą „margin” zawór jest pozycjonowany do obrazu tła



## 4.2.5 Tworzenie tabel z tekstem

### Podstawy

Używanie tabeli jest rekomendowane w celu uniknięcia przesunięcia tekstu w zależności od wielkości okna.

Oczywiście można również zdefiniować tabelę centralnie poprzez CSS.

### Wyjaśnienia

Na poniższym obrazku pokazano tylko nagłówek oraz dwie pierwsze linie tabeli w celu zwiększenia przejrzystości.

Rysunek 4-15

```
<table border='1' >
  <tr>
    <td class="static_field_headline_small">Data</td>
    <td class="static_field_headline_left">Time</td>
    <td class="static_field_headline_left">Value</td>
  </tr>

  <tr>
    <td class="static_field_small">1</td>
    <td class="output_field">:= "Data_Buffer".Data[1].Data_Struct.Timestamp:</td>
    <td class="output_field">:= "Data_Buffer".Data[1].Data_Struct.Value:</td>
  </tr>
  <tr>
    <td class="static_field_small">2</td>
    <td class="output_field">:= "Data_Buffer".Data[2].Data_Struct.Timestamp:</td>
    <td class="output_field">:= "Data_Buffer".Data[2].Data_Struct.Value:</td>
  </tr>
  ...
</table>
```

Kod	Wyjaśnienie
<code>&lt;table border="1"&gt;...&lt;/table&gt;</code>	Grubość ramki tabeli wynosi "1". Tabela bez ramki: <code>border="0"</code> .
<code>&lt;tr&gt;</code> <code>&lt;td</code> <code>class="static_field_headline_small"&gt;</code> <code>Data&lt;/td&gt;</code> <code>&lt;td</code> <code>class="static_field_headline_left"&gt;</code> <code>Time&lt;/td&gt;</code> <code>&lt;td</code> <code>class="static_field_headline_left"&gt;</code> <code>Value&lt;/td&gt;</code> <code>&lt;/tr&gt;</code>	<code>&lt;tr&gt;</code> - wiersz tabeli. Zawartość komórki tabeli znajduje się między <code>&lt;td&gt;</code> (table data) oraz <code>&lt;/td&gt;</code> . Format np. "static_field_headline_small" of dane tabeli definiowane w pliku CSS. "class="<name>" przypisuje format z pliku CSS do elementów w pliku HTML.

#### 4.2.6 Wyświetlanie zmiennych CPU

##### Wyjaśnienia

Zmienne w CPU są zawsze wyświetlane poprzez nazwę symboliczną

Rysunek 4-16

```
<td class="output_field">:= "TankLevel" :</td>
```

Zamiast "TankLevel", wyświetlana jest aktualna wartość zmiennej na stronie internetowej.

### UWAGA

Odkąd zmienna jest umieszczona w tabeli w tagu „<td> . </td>” dodatkowo jest tu wyświetlana. Centralny format „output\_field” jest zdefiniowany w pliku CSS.

#### 4.2.7 Wyświetlanie tekstu poprzez wyliczenia

##### Wyjaśnienia

Poprzez wyliczenia, teksty mogą być umieszczone w indywidualnych wartościach CPU

Rysunek 4-17

```
<td class="output_field_long" colspan="2">
  <b><!-- AWP_Enum_Ref Name="Alarm" Enum="AlarmValue" --> := "Alarm" :</b></td>
```

Zamiast indywidualnych wartości zmiennej "Alarm", uprzednio przypisany tekst w HTML jest wyświetlany. Teksty te są przechowywane w typie enum „AlarmValue”. Te teksty są przenoszone do strony internetowej poprzez DB333

### UWAGA

Odkąd wyliczenia są umieszczone w tabeli, tag „<td> . </td>” jest wyświetlany tutaj. Centralny format „output\_field\_long” jest definiowany w pliku CSS. Rezultat wyświetlany jest jako pogrubiony co jest określone przez tag „<b> . </b>”.

## 4.2.8 Ustawianie zmiennych w CPU za pomocą wartości i przycisku

### Podstawy

Aby było możliwe transferowanie zmiennych do CPU poprzez przeglądarkę internetową, należy użyć form i np. metody „POST”

### Wyjaśnienia

Rysunek 4-18

```
<form method="post" action="" onsubmit="return check();">
  <input type="text" id="wert1" name="Flowrate" size="2"
    style="height: 45px; width: 50px; font-size: 21px; text-align: center; padding: 8px;">
  <input type="submit" value="Set a new Flowrate"
    style="height: 45px; width: 200px">
</form>
```

Kod	Wyjaśnienie
<pre>&lt;form method="post" action="" onsubmit="return check();"&gt; &lt;input type="text" id="wert1" name="Flowrate" size="2" style="height: 45px; width: 50px; font-size: 21px; text-align: center; padding: 8px;"&gt; &lt;input type="submit" value="Set a new Flowrate" style="height: 45px; width: 200px"&gt; &lt;/form&gt;</pre>	<p>Wywołanie formy z metodą POST. Pod „action” nie trzeba określać żadnych detail ponieważ z „action” aktualna strona jest wywoływana domyślnie. Z wywołaniem „onsubmit”, funkcja „check()” jest wykonywana tak jak jest zdefiniowana w pliku JS. Poprzez kliknięcie na „submit”, funkcja sprawdza czy wejście jest w przedziale 1 do 10.</p> <p>Jeżeli warunek ten jest spełniony, funkcja „check()” raportuje TRUE w przeciwnym wypadku wartość wyjściowa to FALSE i wyświetlany jest dodatkowy komunikat.</p> <p>Z wejściowym type=„text”, pole wejściowe jest powiązane z zawartością wysyłana z web serwera CPU z „submit” (tylko gdy check()=TRUE). „submit” jest kontrolowany poprzez przycisk „Set a new Flowrate”.</p>

## 4.2.9 Ustawianie zmiennych w CPU tylko za pomocą przycisku

### Podstawy

Aby przypisać zmiennej w CPU wartość predefiniowaną, należy użyć formy „POST” z ukrytą wartością.

### Wyjaśnienia

Rysunek 4-19

```
<td width="144px" height="21px">
  <form method="post" action="">
    <input type="submit" value="OpenValve">
    <input type="hidden" name="OpenValve" size="20px" value="1">
    <input type="hidden" name="CloseValve" size="20px" value="0">
  </form>
</td>

<td width="144px" height="21px">
  <form method="post" action="">
    <input type="submit" value="CloseValve">
    <input type="hidden" name="CloseValve" size="20px" value="1">
    <input type="hidden" name="OpenValve" size="20px" value="0">
  </form>
</td>
```

### UWAGA

Odkąd przyciski są zlokalizowane w tabeli, można zobaczyć tu również tagi „<td> . </td>”

Kod	Wyjaśnienie
<pre>&lt;form method="post" action=""&gt; &lt;input type="submit" value="OpenValve"&gt; &lt;input type="hidden" name="OpenValve" size="20px" value="1"&gt; &lt;input type="hidden" name="CloseValve" size="20px" value="0"&gt; &lt;/form&gt;</pre>	<p>Wywołanie formy z metodą „post”. Pod „action” nie ma potrzeby określania detali ponieważ z „action” aktualna strona jest wywoływana domyślnie. Z wejściowym type=„hidden”, zmiennej „OpenValve” jest przypisana wartość 1, a zmiennej „CloseValve” wartość 0. Za pomocą przycisku „submit”, wartości zmiennicy są wysyłane do web serwera CPU.</p>
<pre>&lt;form method="post" action=""&gt; &lt;input type="submit" value="CloseValve"&gt; &lt;input type="hidden" name="CloseValve" size="20px" value="1"&gt; &lt;input type="hidden" name="OpenValve" size="20px" value="0"&gt; &lt;/form&gt;</pre>	<p>Akcja odwrotna do powyżej opisanej: Wywołanie formy aby przypisać wartość 1 do zmiennej „CloseValve” oraz wartości 0 do zmiennej „OpenValve”</p>

## 5 Konfiguracja i ustawienia

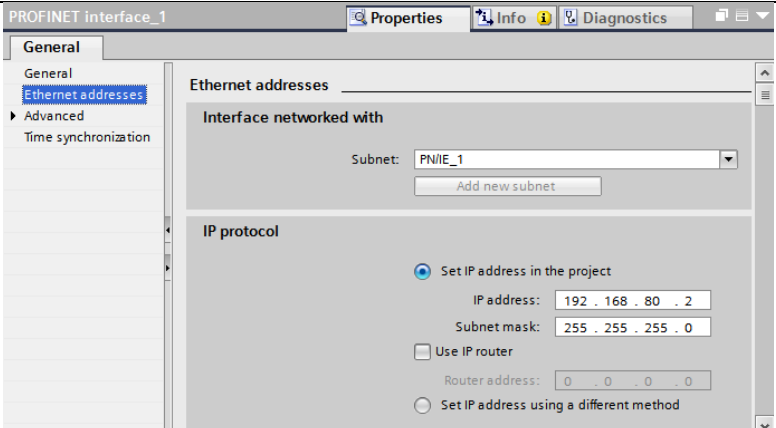
Rozdział zawiera wszystkie informacje na temat możliwości samodzielnego tworzenia i obsługi strony internetowej dla CPU z interfejsem PN. W tym przykładzie użyto CPU 1214C DC/DC/DC. Wszystkie poniższe kroki są kompletnymi przykładami. Aby uruchomić stronę internetową proszę przeczytać rozdział 6.

### 5.1 Procedura tworzenia strony internetowej

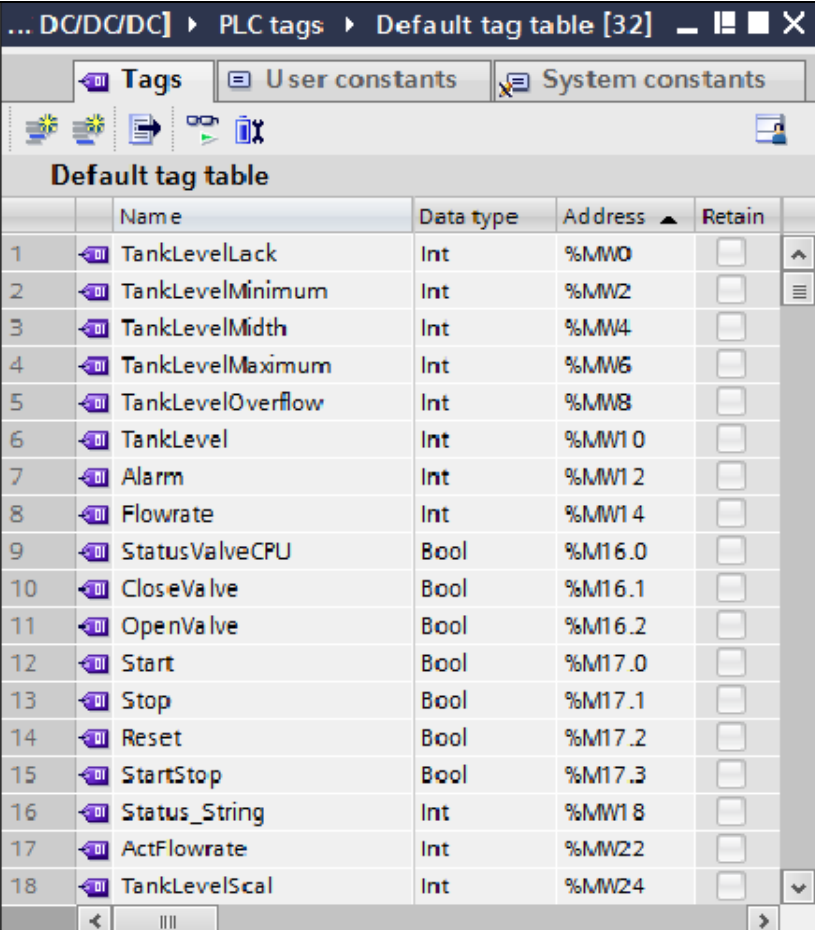
Konfiguracja i ustawienia w STEP 7 oraz pisanie pliku w HTML są ściśle powiązane. Poniższa procedura jest rekomendowana dla:

1. Konfiguracji S7-1200 (1214C DC/DC/DC)
2. Tworzenie zmiennych w tablicy zmiennych lub boku DB
3. Tworzenie plików HTML
4. Ustawienia Web serwera i generowanie bloków danych
5. Ustawianie hasła CPU
6. Tworzenie, kompilacja i ładowanie programu S7
7. Wywoływanie strony w przeglądarce internetowej

## 5.2 Konfiguracja S7-1200 (1214C DC/DC/DC)

Nr	Akcja	Komentarz
1.	Uruchomić STEP 7 i utworzyć nowy project o nazwie "Webserver_S7-1200" poprzez "Project > New...".	
2.	Umieścić sterownik S7-1200 poprzez "Add new device" >> "PLC > SIMATIC S7-1200 > CPU 1214C DC/DC/DC > 6ES7 214-1AE30-0XB0 v2.2". Automatycznie otwiera się widok ustawień sterownika „Device view”.	
3.	Kliknąć na przycisk "Add new subnet" we właściwościach interfejsu Ethernet.	
4.	Przypisać adres IP do interfejsu Ethernet. Poprzez ten adres IP będzie możliwe połączenie się ze sterownikiem poprzez przeglądarkę internetową.	

### 5.3 Tworzenie zmiennych w tablicy zmiennych lub bloku DB

Nr	Akcja	Komentarz
1.	Kliknąć dwa razy na "Default tag table" w folderze "PLC_1 > PLC tags"	
2.	Edycja zmiennych.	

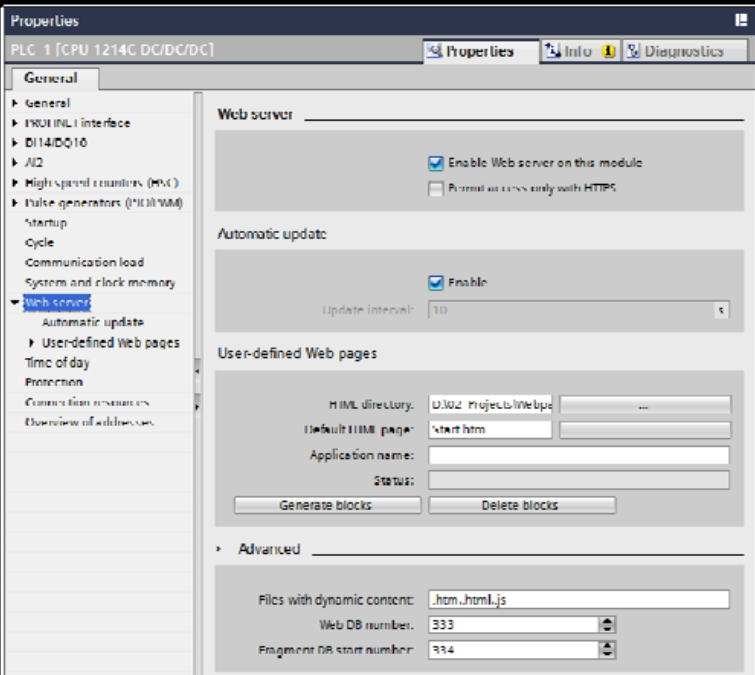
## 5.4 Tworzenie plików HTML

W celu stworzenie pliku HTML, potrzebna jest lista zmiennych z rozdziału 5.3. Tworzenie zmiennych w tablicy zmiennych lub bloku danych DB. W celu stworzenia strony internetowej polecane jest użycie edytora HTML takiego jak np. Microsoft FrontPage, który automatycznie tworzy tagi, zaznacza na czerwono błędnie wpisany kod, lub użycie prostego narzędzia takiego jak np. Notepad++.

Nr	Akcja	Komentarz
1.	Utworzyć plik HTML przy użyciu edytora. Zapisać plik HTML z wymaganymi obrazami, formatkami stylów i skryptami w ścieżce „\html”.	Szczegółowe informacje na temat tworzenia pliku HTML mogą być znalezione w rozdziale 3.3. Podstawy tworzenia stron użytkownika w rozdziale 4.2.

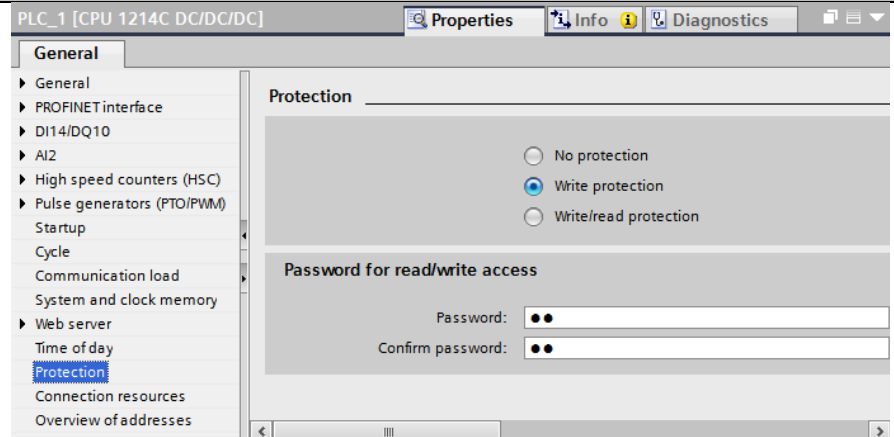


## 5.5 Ustawienia Web serwera i generowanie bloków danych

Nr	Akcja	Komentarz
1.	Kliknąć "Web server" we właściwościach PLC. Aktywować opcję "Enable Web server on this module" oraz "Automatic update". Jeżeli wymagany jest bezpieczny dostęp do standardowej strony opcja „Permit Access only with HTTPS” musi być zaznaczona.	
2.	Wpisać ścieżkę dostępu do pliku HTML i wybrać stronę startową.	
3.	Wygenerować blok sterujący (block Control, domyślnie: DB333) oraz blok z danymi strony (Fragment DB, domyślnie od DB334) klikając „Generate blocks. STEP 7 V11 sprawdza projekt w związku ze zmiennymi, łąduje poszczególne pliki, sprawdza odnośniki do obrazów itd., następnie zapisuje wszystko do DB333 i od DB334. Status generowania wyświetlany jest w niezależnym oknie lub oknie inspektora pod oknem info.	

## 5.6 Tworzenie plików HTML

W celu stworzenie pliku HTML, potrzebna jest lista zmiennych z rozdziału 5.3. Tworzenie zmiennych w tablicy zmiennych lub bloku danych DB. W celu stworzenia strony internetowej polecane jest

Nr	Akcja	Komentarz
1.	Kliknąć "Protection" we właściwościach PLC. Zaznaczyć „Write protection”.	
2.	Wpisać hasło i potwierdzić to hasło	

### UWAGA

Ze strony Web serwera nie ma znaczenia czy jest zaznaczone „Write protection” lub „Write/read protection”. Web serwer nie obsługuje ochrony przed zapisem.

## 5.7 Tworzenie, kompilacja i ładowanie programu S7

Przykładowy program S7 może być znaleziony jako załącznik do tego przykładu. Podczas tworzenia programu należy pamiętać o następujących aspektach:


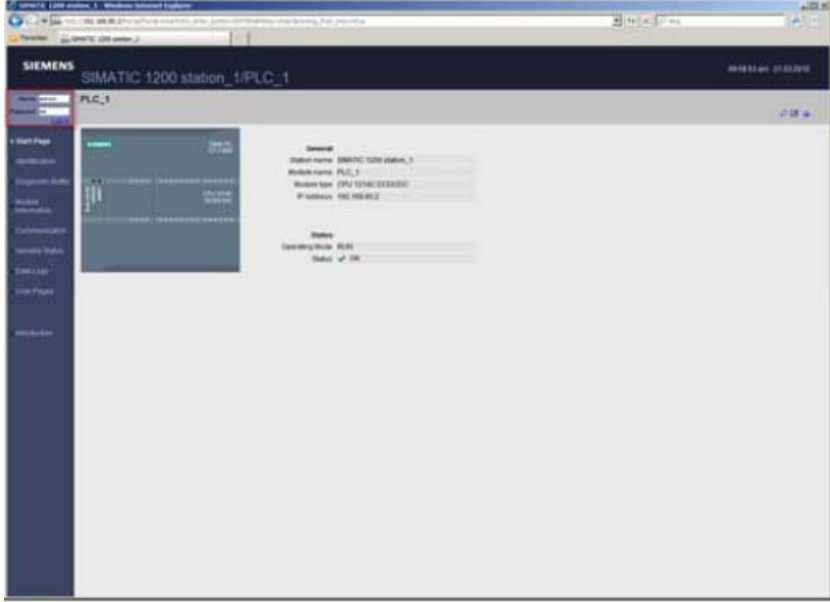
- Wywołać funkcję „WWW” (SFC99). Funkcja „WWW” inicjalizuje web serwer w CPU. Cykliczne wywołanie tej funkcji zapewnia wyświetlanie na stronie zmieniających się zmiennych. Cykliczne wywołanie SFC99 należy zrealizować w OB1.
- Wpisać numer bloku sterującego (Web Control DB: np. DB333) w miejscu parametru wejściowego CTRL\_DB funkcji „WWW”.


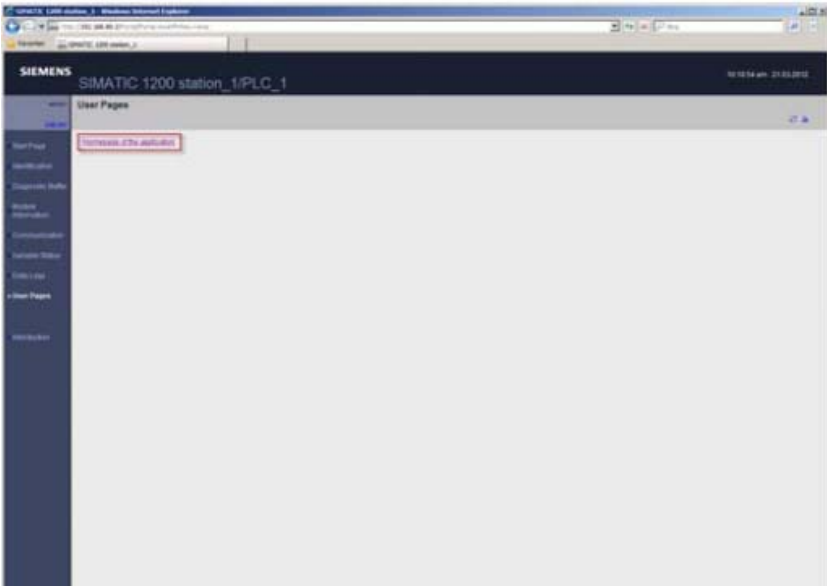
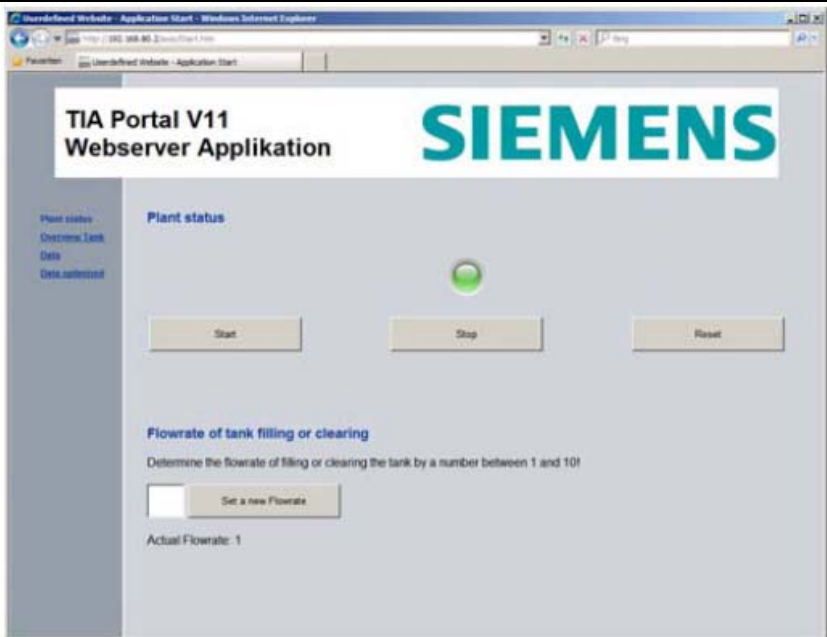
Nr	Akcja	Komentarz
1.	Kompilacja Kliknąć prawym na S7 1214C DC/DC/DC i wybrać “Compile >> All”.	
2.	Ładowanie projektu Kliknąć prawym przyciskiem myszy na 7 1214C DC/DC/DC i wybrać “Download to Device > All” Ustawić interfejs PG/PC w oknie dialogowym: -PN/IE -<network adapter (local) PN/IE Wybrać S7 1214C DC/DC/DC i następnie kliknąć „Load”.	Okno dialogowe do ustawienia interfejsu PG/PC jest pokazywane tylko podczas pierwszego ładowania.

### UWAGA

Jeżeli ma zostać użyte inne CPU, można je zmienić w „Devices & networks”. Po zmianie, należy pamiętać o ponownym ustawieniu opcji web serwer (patrz rozdział 5.5 )

## 5.8 Wywołanie strony poprzez przeglądarkę internetową

Nr	Akcja	Komentarz
1.	<p>Uruchomić przeglądarkę internetową np. Internet Explorer. Wpisać adres IP sterownika np.: <a href="http://192.168.80.20">http://192.168.80.20</a>.            Uwaga: komputer i sterownik muszą być w tej samej podsieci.            Ładuje się strona startowa sterownika.            - Kliknąć „ENTER”.</p>	
2.	<p>Wpisać nazwę „admin” i hasło „s7”. Kliknąć „Log in”.            Kompletna strona internetowa CPU jest otwarta.</p>	
<p>Uwaga: Jeżeli wystąpi problem z certyfikatem bezpieczeństwa strony internetowej należy przejść do „Introduction” i ściągnąć Siemens security certificate np. w Internet explorerze:</p> <ul style="list-style-type: none"> <li>- Kliknąć link „download certificate”. Wyświetla się komunikat „File download – Security Warning”</li> <li>- Kliknąć „Open” w oknie „File download – Security Warning” aby otworzyć plik. Otwiera się okno „Certificate”</li> <li>- Kliknąć przycisk „Install certificate...” w oknie „Certificate”, w celu otworzenia asystenta importu certyfikatu</li> <li>- Postępować zgodnie z pojawiającymi się informacjami w „Certificate import Wizard” w celu importu certyfikatu. „Trusted Root Certification Authorities”</li> </ul>		

3.	Aby otworzyć stronę użytkownika, kliknąć "User Pages"	
4.	Aby wystartować przykładową aplikację kliknąć "Homepage of the application". Otwiera się strona startowa "Start".	
5.	Szczegółowe omówienie działania przykładowej strony znajduje się w rozdziale 7	

## 6 Instalacja

### 6.1 Procedura tworzenia strony internetowej

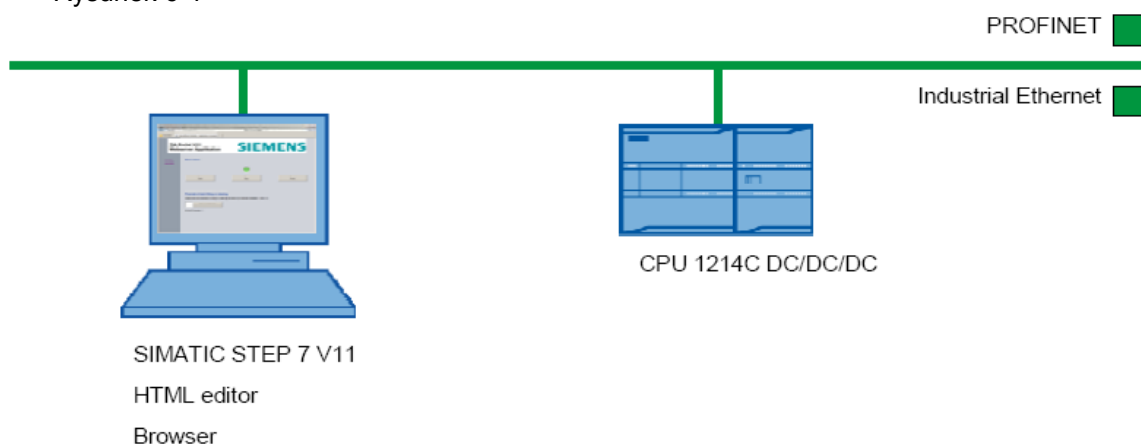
#### Instalacja sprzętu

Poniższy rysunek pokazuje strukturę sprzętu przykładowej aplikacji.

Komputer z zainstalowaną przeglądarką internetową musi być podłączony do CPU poprzez Industrial Ethernet, np.:

- bezpośrednio przez interfejs PN w CPU
- poprzez switch

Rysunek 6-1



#### UWAGA

Proszę skorzystać z przewodników instalacji poszczególnych elementów

#### Instalacja oprogramowania

Nr	Akcja	Komentarz
1.	Zainstalować STEP 7 V11 SP2	
2.	Zainstalować a komputerze narzędzie do tworzenia stron internetowych, np. MS frontpage lub Notepad++.	
3.	Zainstalować przeglądarkę internetową na komputerze np. Internet Explorer, Firefox	


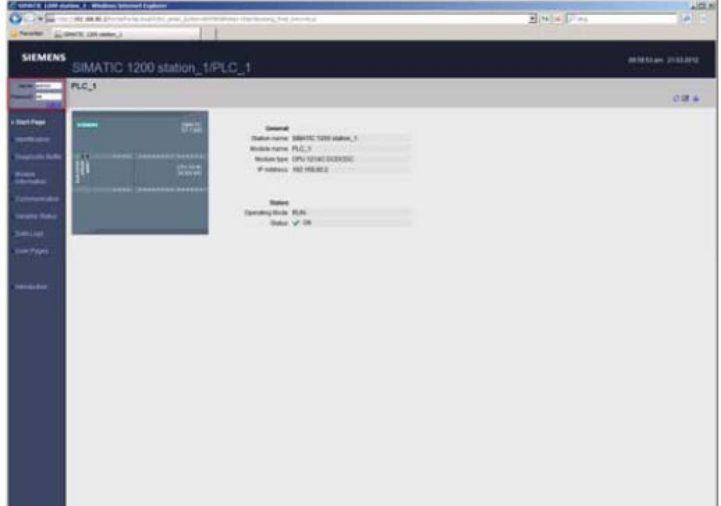
## 6.2 Instalacja przykładowej aplikacji

Nr	Akcja	Komentarz
1.	Rozpakować plik „Example_S7-1200_Webserver.zip” w folderze projektu	
2.	Uruchomić SIMATIC STEP 7 V11 SP2	
3.	Otworzyć projekt w SIMATIC STEP 7 SP2	
4.	Przejsć do Device view	
5.	Jeżeli posiadane CPU jest inne od tego w projekcie, należy zmienić CPU za pomocą opcji change device	
6.	We właściwościach CPU, przypisać adres IP do interfejsu sterownika	Więcej informacji w rozdziale 5.2
7.	Wybrać Sterownik S7-1200 i załadować cały projekt	
8.	Otworzyć przeglądarkę internetową i wywołać stronę internetową sterownika poprzez wpisanie jego adresu IP	Więcej informacji w rozdziale 5.8

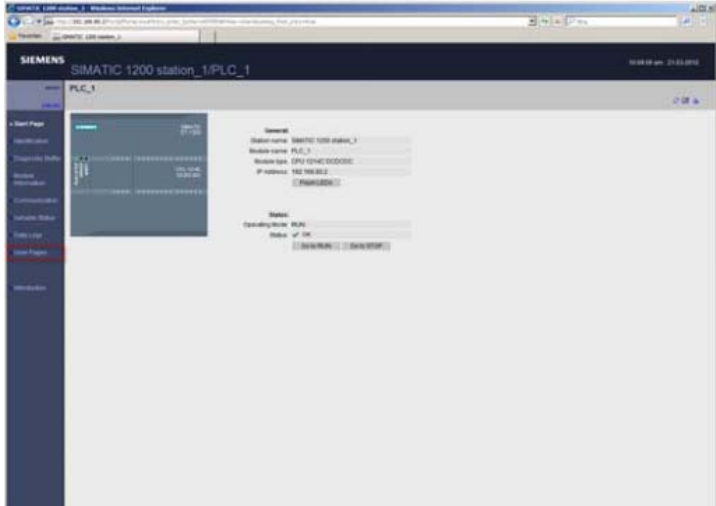
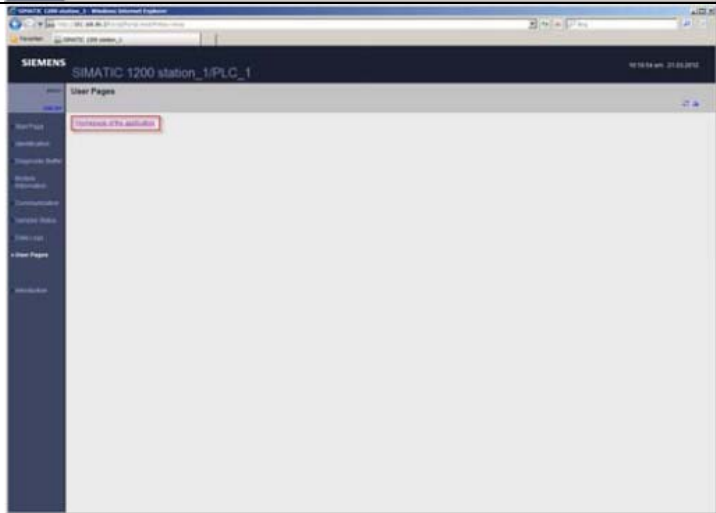
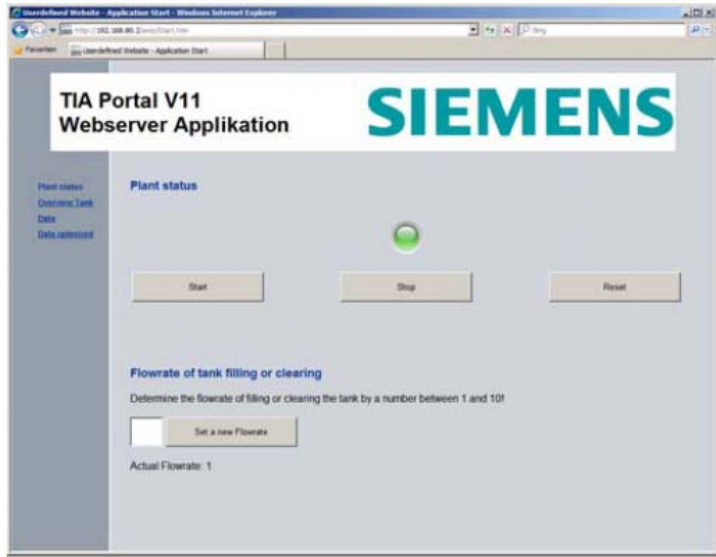
## 7 Praca z aplikacją


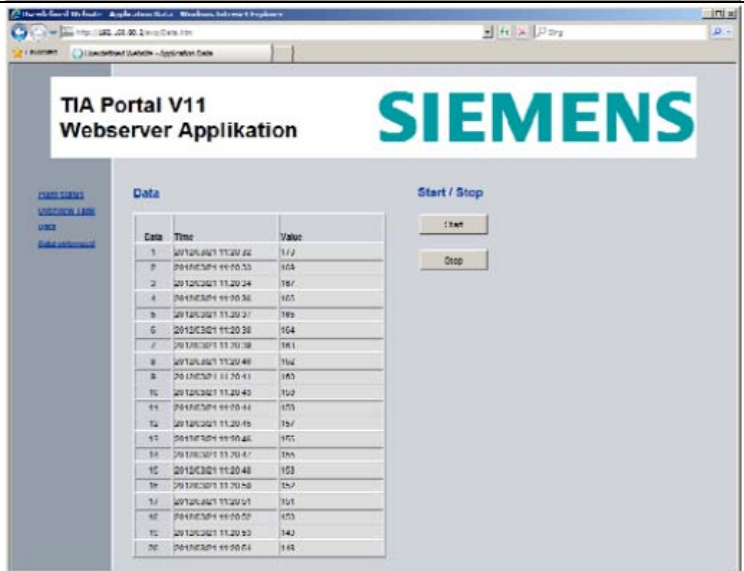
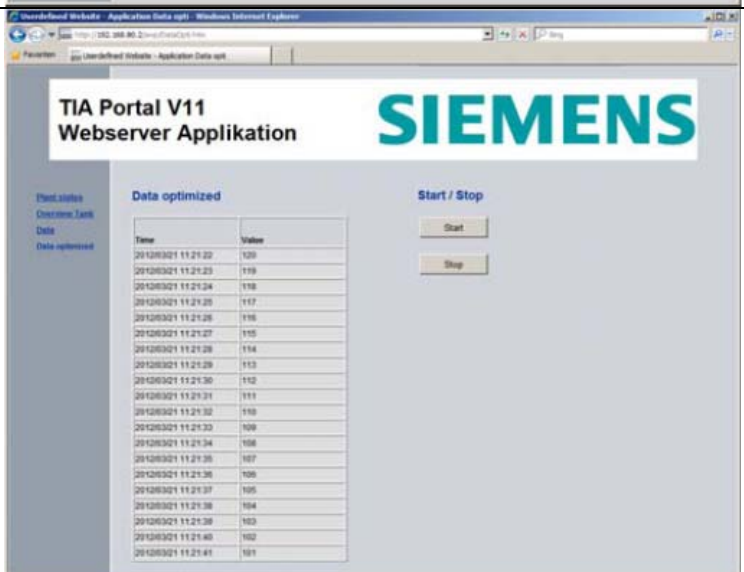
## W tym rozdziale

Ten rozdział dostarcza informacji na temat pracy z przykładową aplikacją

Nr	Akcja	Komentarz
1.	<p>-uruchomić przeglądarkę internetową np. Internet Explorer. Wpisać adres IP sterownika <a href="http://192.168.80.20">http://192.168.80.20</a>. Otwiera się strona startowa -wcisnąć ENTER</p>	
2.	<p>Wpisać nazwę „admin” i hasło „s7”. Kliknąć „Log in”. Kompletna strona internetowa CPU jest otwarta.</p>	
<p>Uwaga: Jeżeli wystąpi problem z certyfikatem bezpieczeństwa strony internetowej należy przejść do „Introduction” i ściągnąć Siemens security certificate np. w Internet explorerze:</p> <ul style="list-style-type: none"> <li>- Kliknąć link „download certificate”. Wyświetla się komunikat „File download – Security Warning”</li> <li>- Kliknąć „Open” w oknie “File download – Security Warning” aby otworzyć plik. Otwiera się okno „Certificate”</li> <li>- Kliknąć przycisk „Install certificate...” w oknie „Certificate”, w celu otworzenia asystenta importu certyfikatu</li> <li>- Postępować zgodnie z pojawiającymi się informacjami w „Certificate import Wizard” w celu importu certyfikatu. “Trusted Root Certification Authorities”</li> </ul>		



3.	Aby otworzyć stronę użytkownika, kliknąć "User Pages"	
4.	Aby wystartować przykładową aplikację kliknąć "Homepage of the application". Otwiera się strona startowa "Start".	
5.	<p>Poprzez strony internetowe jest bezpośredni dostęp do sterownika:</p> <p><b>Strona <i>Plant status</i>:</b></p> <ul style="list-style-type: none"> <li>- Uruchomić aplikację poprzez kliknięcie przycisku „Start”</li> <li>- Po kliknięciu „<b>Stop</b>” aplikacja zatrzymuje się</li> <li>- Kliknięcie przycisku „<b>Reset</b>” powoduje powrót aplikacji do stanu początkowego.</li> <li>- Status pracy aplikacji sygnalizowany jest przez LED (czerwony: wyłączona, zielony: włączona)</li> <li>- Wartość przepływu może być wpisana ręcznie. W programie S7, średnia wartość przepływu ustawiona jest na 5. Im wyższa wartość przepływu, tym szybszy przepływ. Wpisywanie wartości przepływu kontrolowane jest przez skrypt. Jeżeli pojawią się niewłaściwe wartości, zostaje wyświetlony komunikat. Poprzez linki w pasku nawigacyjnym można przełączać ekrany</li> </ul>	

6.	<p>Strona <b>Tank overview</b>:</p> <ul style="list-style-type: none"> <li>- Poprzez niebieski pasek na zbiorniku można podejrzeć aktualny poziom zbiornika. Dodatkowo poziom napełnienia wyświetlony jest w postaci tekstu.</li> <li>- po kliknięciu przycisku „OpenValve” następuje opróżnianie zbiornika.</li> <li>- po kliknięciu „CloseValve” następuje napełnianie zbiornika.</li> <li>-zawór zielony:otwarty</li> <li>- zawór czerwony: zamknięty</li> <li>- Strona jest automatycznie odświeżana.</li> </ul>	
7.	<p>Strona <b>Data</b></p> <ul style="list-style-type: none"> <li>-Ta strona pokazuje bufor pierścieniowy z dwudziestoma ostatnimi wpisami poziomu zbiornika.</li> <li>- strona jest automatycznie odświeżana</li> </ul>	
8.	<p>Strona <b>Data optimized</b>:</p> <ul style="list-style-type: none"> <li>-strona pokazuje dwadzieścia ostatnich poziomów zbiornika</li> <li>-wpisy są automatycznie odświeżane poprzez JavaScript</li> <li>-Uruchomić aplikację poprzez przycisk „Start”</li> <li>-Po kliknięciu „Stop” aplikacja zatrzymuje się</li> </ul>	



## 8 Glosariusz

**AWP**

Automation Web Programming

**AWP command (komenda AWP)**

Komenda AWP jest rozumiana jako specjalna składnia komendy określająca które dane są wymieniane między sterownikiem i plikiem HTML

**CSS**

CSS (Cascading Style Sheets) definiuje sposób wyświetlania zawartości strony internetowej (formatowanie).

**HTML file (Plik HTML)**

Pliki HTML są podstawą World Wide Web i są otwierane poprzez przeglądarkę internetową.

**HTTP**

Hypertext Transfer Protocol (HTTP) jest protokołem służącym do transferu danych poprzez sieć.

**HTTPS**

Hypertext Transfer Protocol Secure jest protokołem komunikacyjnym, który jest używany w WWW do wymiany bezpiecznych danych

**MIME type**

Multipurpose Internet Mail Extensions (MIME), informuje przeglądarkę internetową które dane są wysyłane poprzez web serwer, np. Czy to jest czysty tekst, plik HTML lub obraz PNG.

**UTF-8**

UTF-8 (8-bit UCS Transformation Format) jest najczęściej używanym kodem znaków Unicode.

**Web browser (przeglądarka internetowa)**

Przeglądarka internetowa jest programem wizualizacyjnym dla stron internetowych i komunikują się z web serwerami. Typowe przeglądarki to:

-Microsoft Internet Explorer

-Mozilla Firefox

**Web page (Strona Web)**

Patrz plik HTML.

**Web server**

Web serwer przechowuje i udostępnia stronę internetową. Web serwer jest programem który transferuje dokumenty za pomocą standardowych protokołów transmisji (http, HTTPS) do przeglądarki internetowej.

## 9 Literatura

Poniższa lista zawiera jedynie wybór pozycji uzupełniających:

Nr	Temat	Tytuł
/1/	STEP7 SIMATIC S7 -1200	Automating in STEP 7 with SIMATIC S7-1200 Author: Hans Berger Publicis Publishing ISBN: 978-3895783562
/2/	HTML	HTML und CSS, Praxisrezepte für Einsteiger Robert R. Agular mitp ISBN 978-3-8266-1779-9
/3/	HTML	HTML Handbuch Stefan Münz/Wolfgang Nefzger Franzis Verlag ISBN 3-7723-6654-6
/4/	Javascript	JavaScript und Ajax, Das umfassende Handbuch Christian Wenz Galileo Press ISBN 978-3-8362-1128-4