

SIMATIC w Internecie,

część 2

Tym razem zajmiemy się omówieniem programowania serwera WWW. Jak wspomniałem poprzednio, po otrzymaniu zapytania, serwer WWW uruchamia plik *index.html*, w którym znajdują się bezpośrednio wywołania stron wizualizacyjnych lub mechanizmy organizacji okien. Sposób definiowania poszczególnych elementów okien proponuję przestudiować edytując pliki. W menu pozioma, czteroelementowa tabela zawiera nazwy obrazów synoptycznych. Kliknięcie na nazwę powoduje pojawienie się w dolnym obszarze okna przeglądarki pożądanego obrazu synoptycznego.

Aplety

Każdy obraz widoczny w przeglądarce internetowej jest opisany jest przez plik tekstowy z rozszerzeniem *html* (oczywiście w naszym przypadku). Strony opisane w HTML są jednak tylko statyczne. Oznacza to, że obraz składa się z elementów, które po wywołaniu nie zmieniają się (brak animacji). Aby to zmienić w kod HTML wplata się tzw. aplety, które po wywołaniu strony można powiedzieć żyją własnym życiem. Aplet jest więc funkcją napisaną w języku wyższego poziomu (zwykle w JAVA), która realizuje zazwyczaj kilka zadań. W naszym przypadku wywołany aplet zajmuje się sprawdzeniem adresu w CPU i na podstawie stanu/wartości wykonuje animację obiektu graficznego. Animacja to dla zmiennych cyfrowych pokazanie/ukrycie bitmapy, dla zmiennych analogowych to wyświetlenie zmieniającej się wartości. Nowe procesory CP fabrycznie zawierają kilka apletów do swobodnego użycia. Można także wykonać własne aplety a następnie użyć w aplikacji. Aplety są zawarte w plikach *JBeansAPI* z rozszerzeniem *jar* i znajdują się w katalogu *applets* serwera FTP. Opis użycia „fabrycznych” apletów można znaleźć na stronie WWW: <http://support.automation.siemens.com/WW/view/en/21602415>.

Programowanie serwerów WWW z rodziny SIMATIC niczym się nie różni od programowania innych komputerów. W naszym projekcie korzystamy z kodu HTML, w który wplatamy funkcje czy też aplety napisane w Javie.

Aplety, które posiada CP, mają postać graficznych kontrolki, które przy odpowiedniej konfiguracji mogą mieć różną wizualną reprezentację. Patrząc jednak na tzw. gotowe rozwiązania można poczuć pewien niedosyt.

Dostępne są aplety:

- S7IdentApplet (tylko dla S7-300/400),
- S7StatusApplet (j.w.),
- S7GetApplet,
- S7PutApplet.

Jak widać, statusy oraz identyfikację można odczytywać jedynie z modeli wyższych S7-300/400. Pozostałe aplety są uniwersalne dla CP S7-200/300/400. Na stronach wizualizacji, do odzwierciedlenia procesu służą dwa ostatnie aplety. Narzucają nam one z góry postać graficzną animowanego obiektu. Aplet *Get* dla zmiennych analogowych może mieć postać tachometru, termometru lub pola z wartością liczbową. W przypadku zmiennej cyfrowej na podstawie wartości można wyświetlać jeden z dwóch zadeklarowanych ciągów znaków. Aplet *Put* ma postać pola z przyciskiem. W pole wpisujemy wartość, przyciskiem zatwierdzamy – przepisujemy

wartość do PLC. Wielkości poszczególnych elementów są niezmiennie.

Na bazie istniejących apletów podjęliśmy się przebudowania ich nieco. Wynikiem są dwie klasy *S7NewGet* oraz *S7NewPut*, które zawierają kilka typów apletów (kontrolki) *Get* i *Put*. Nowe kontrolki realizują:

- „KCzytajBitGr” (odczyt): sprawdzenie wartości cyfrowej (bitu) w PLC i wyświetlenie różnych grafik dla stanów 0 i 1,
- „KCzytajBitTx” – j.w. ale dla wartości 0 i 1 podajemy ciągi znaków,
- „KPoziomZb” (j.w.): odzwierciedla wartość analogową – pasek ze zdefiniowaną wielkością oraz z animowaną wysokością,
- „KDisplayAn” (j.w.): odzwierciedla wartość analogową – liczba,
- „KUstawBit” (zapis): ustawienie/zerowanie bitu,
- „KUstawAn” (j.w.): ustawienie wartości analogowej,
- „KWstawGr”: wstawienie miejsce o podanych współrzędnych statycznej grafiki – bez animacji,
- „KAppletGet”: kontrolka wskaźnika graficznego (tachograf) wartości analogowej.

Oczywiście używając kontrolki



Rys. 1. Główna strona projektu WWW (górną część to Menu, dolną to okno z obrazem start)

nowych w dalszym ciągu można korzystać z apletów „fabrycznych” nawet na jednej stronie. Dodatkowo został wykonany jeden, dość istotny zabieg. Każda kontrolka/aplet odzwierciedla jedną zmienną w PLC. Jeżeli na stronie mamy pokazać np. 10 zmiennych to należy użyć 10 apletów. Każdy aplet to kilka kB kodu co oznacza, że strona będzie zajmować dużą ilość pamięci na serwerze FTP, która jest przecież ograniczona. Najgorsze jest to, że taka strona będzie się ładować do przeglądarki WWW wielokrotnie dłużej i jeżeli zostanie przekroczony maksymalny czas dostępu do serwera to strona nie załaduje się w ogóle. Aby uniknąć takich ograniczeń została wykonana w JS klasa do zarządzania kontrolkami znajdującymi się w obrębie jednego okna/strony WWW. Klasa realizuje kolejkę kontrollek, które na swoim poziomie jako parametr mają zdefiniowany adres komórki w PLC, który odzwierciedlają. Klasa dynamicznie podstawia adresy z kontrollek do apletu *Get/Put*

a po zwróceniu wartości przekazuje ją z powrotem kontrolce, która wykonuje ostateczną animację obiektu graficznego bądź tekstowego.

Obrazy WWW

Każdy obraz synoptyczny to oddzielna strona WWW. Każda taka strona jest opisana w oddzielnym pliku *HTML*. Każdy plik składa się z części stałej (nagłówki *HTML*, odwołanie do IP CP), deklaracji nowej instancji klasy *Obraz* oraz wywołania poszczególnych kontrollek. Wywołanie kontrolki to utworzenie nowej instancji danej kontrolki, jej parametryzacja (parametry graficznej reprezentacji), wskazanie adresu pamięci PLC, który ma reprezentować oraz aktywacja kontrolki w kolejce odświeżania danych (kolejka dostępu do apletu *Get* odpytującego PLC) w klasie *Obraz*. Po deklaracji klasy *Obraz* musimy zadeklarować czas odświeżania kontrollek. Wartością zalecaną jest 500 ms. Oznacza to, że co 0,5 sekundy aplet *Get* będzie odpytował CP o kolejną komórkę da-

nych. Czas oczywiście można zwiększyć, ale zalecam nie zmniejszać go. Rzeczywista częstotliwość odświeżania danych na obrazie będzie zależała od jakości sieci poprzez którą nawiązuje połączenie przeglądarka (PC) z procesorem CP (PLC) oraz od liczby kontrollek w jednym oknie WWW. Zasada jest jedna: im jest ich mniej tym lepiej.

Nowe biblioteki

Na początek należy do katalogu *applets* serwera FTP wgrać nowe klasy bibliotek. Do istniejących plików należy dograć:

- *S7newGet.class*
- *S7newPut.class*
- *cp343.js*
- *pusty.png*

Pliki można pobrać ze strony polskiego wsparcia Siemens, dział *Aplikacje*.

Kod

Jak już wspomniałem strony WWW opisane są poprzez kod *HTML* (list. 1). Część górna to de-

List. 1. Kod HTML opisujący testową stronę WWW

```
<HTML><HEAD> (1)
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
<script type="text/javascript" src="/applets/cp343.js" language="JavaScript1.5"></script>
<style type="text/css"> <!--
@import url("main.css");
--> </style>
<TITLE>Demo wizualizacji dla modułu SIEMENS CP343-1</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<center>
<script type="text/javascript" language="JavaScript1.5">

    var Obraz1 = new Obraz(700,410,"","img/wizualizacja.png") (2)

    Obraz1.ip = „82.115.86.93:80” // IP+ nr portu
    Obraz1.zwloka_komunikacji = 500

    // Dla zbiornika nr 1 (3)
    var zbiornik1 = new Obraz1.KPoziomZb(„zbiornik1”, „blue”,0, 1)
    zbiornik1.ustaw_pozycje(100,70,46,100)
    //zbiornik1.ustaw_any_pointer(new Obraz1.AnyPointer(„BYTE”,1,“M”,0,0,0))
    zbiornik1.ustaw_symbol(„PoziomZ1(0/2)”)
    Obraz1.dodaj_kontrolke(zbiornik1)

    var zbiornik_wartosc1 = new Obraz1.KDisplayAn(„zbiornik_wartosc1”, „1px solid red”, „white”,
    „center”, „9px”, „- - -”,
    „<b>Poziom Z1<br>”, „[%]”, 1)
    zbiornik_wartosc1.ustaw_pozycje(50,100,50,40)
    //zbiornik_wartosc1.ustaw_any_pointer(new Obraz1.AnyPointer(„BYTE”,1,“M”,0,0,0))
    zbiornik_wartosc1.ustaw_symbol(„PoziomZ1(0/2)”)
    Obraz1.dodaj_kontrolke(zbiornik_wartosc1)

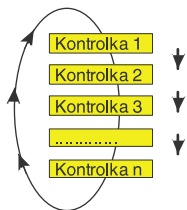
    var zawor1 = new Obraz1.KCzytajBitGr(„zawor1”, „img/zawor_c.png”, „img/zawor_z.png”,0)
    zawor1.ustaw_pozycje(81,239)
    zawor1.ustaw_any_pointer(new Obraz1.AnyPointer(„BOOL”,1,“DB”,5,60,0))
    Obraz1.dodaj_kontrolke(zawor1)

    var zawor_wartosc1 = new Obraz1.KCzytajBitTx(„zawor_wartosc1”, „1px solid red”, „white”,
    „center”, „9px”, „<b>Zamknięty”, „<b>Otwarty”,0)
    zawor_wartosc1.ustaw_pozycje(130,239,60,40)
    zawor_wartosc1.ustaw_any_pointer(new Obraz1.AnyPointer(„BOOL”,1,“DB”,5,60,0))
    Obraz1.dodaj_kontrolke(zawor_wartosc1)

    Obraz1.uruchom()

</script>
</center>

</BODY>
</HTML>
```

Rys. 2. Zasada działania klasy *Obraz*

inicja nagłówków (1), taka sama w każdym kolejnym pliku, potem jest deklaracja nowej instancji okna (2) a następnie (3) wywoływane są instancje kontrolki animujących obiekty. Na końcu znajduje się instrukcja uruchamiająca instancję okna oraz znaczniki HTML kończące (zamykające) plik. Na potrzeby projektu założony został na dysku FTP katalog *wizualizacja*. Znajdują

się w nim 3 pliki HTML opisujące obrazy wizualizacyjne. Katalog *img* zawiera bitmapy wykorzystywane podczas animacji. Np. wizualizacja pracy pompy polega na pokazywaniu symbolu w kolorze zielonym lub szarym. Wykonane grafiki symboli (w kolorze zielonym i szarym) pobierane są z katalogu do bieżącej animacji stanu pompy. Należy zaznaczyć, że obydwa symbole pompy mogą być wykorzystywane wielokrotnie. Żadna kontrolka nie blokuje grafik w *img*. Na potrzeby animacji wykonuje kopię wskazanego wzorca i umieszcza go na stronie WWW.

Uruchomienie na stronie WWW apletu wymaga podania nazwy oraz hasła użytkownika. Są to oczywi-

ście hasła, które zostały zdefiniowane podczas parametryzacji CP z poziomu STEP7. Każda kontrolka oraz aplet może odwoływać się do komórki pamięci poprzez adres bezwzględny (parametryzacja wskaźnika ANY) lub poprzez zdefiniowany na poziomie PLC (STEP7) symbol. Obydwa sposoby pokazane są w oknie kodu (rys. 2). Można je stosować zamiennie. Odwołanie symboliczne jest szczególnie wygodne w przypadku wykonywania aplikacji dla sterownika S7-200 z CP 243-1 IT. Dokładny opis parametryzacji poszczególnych kontrolki można pobrać ze strony www.siemens.pl (dział *Aplikacje*).

Andrzej Cyganik
Siemens Sp. z o.o.

R	E	K	L	A	M	A
<h1 style="color: yellow;">Prenumeruj</h1> <h2 style="color: red;">za darmo !</h2> <h2 style="color: red;">lub półdarmo !</h2> <h1 style="color: yellow;">start gratis</h1> <p style="text-align: center;">za pierwsze trzy miesiące prenumeraty nie musisz płacić!</p> <div style="display: flex; justify-content: space-between;"> <div style="text-align: center;">  </div> <div style="text-align: center;"> <h1 style="color: yellow;">kontynuacja do</h1> <h1 style="color: red;">50% taniej</h1> <h2 style="color: yellow;">super bonusy</h2> </div> <div style="text-align: center;"> <p>po roku prenumeraty dostaniesz 2 numery gratis, po dwóch – 3 numery... i w ten sposób po kilku latach nawet pół prenumeraty możesz mieć za darmo!</p> </div> </div> <p>tylko dla prenumeratorów: co miesiąc numer archiwalny, bez limitu – dostęp do specjalnego serwisu w sieci, Klub AVT, rabaty, przywileje...</p> <p>Najważniejsze! Prenumerator wydania papierowego dostaje za darmo e-wydanie</p> <div style="text-align: right; color: red; transform: rotate(-45deg); font-weight: bold;"> <p>SZCZEGÓŁY STRONA 144</p> </div>						