



Logo Soft Comfort v8

Wbudowany serwer Web oraz aplikacje pracujące w sieci

Tym, co najbardziej odróżnia nowe Logo! od wcześniejszych wersji jest możliwość pracy w sieci Ethernet. Owszem, w ten rodzaj interfejsu była wyposażona już wersja siódma sterownika, ale dopiero w ósmej wersji firma Siemens stanęła na wysokości zadania, bo po pierwsze, ułatwiono tworzenia programów, a po drugie, cena „ósemki” jest prawie taka sama, jak „szóstki”. Ba! W niektórych sklepach jest nawet o te kilka złotych niższa!

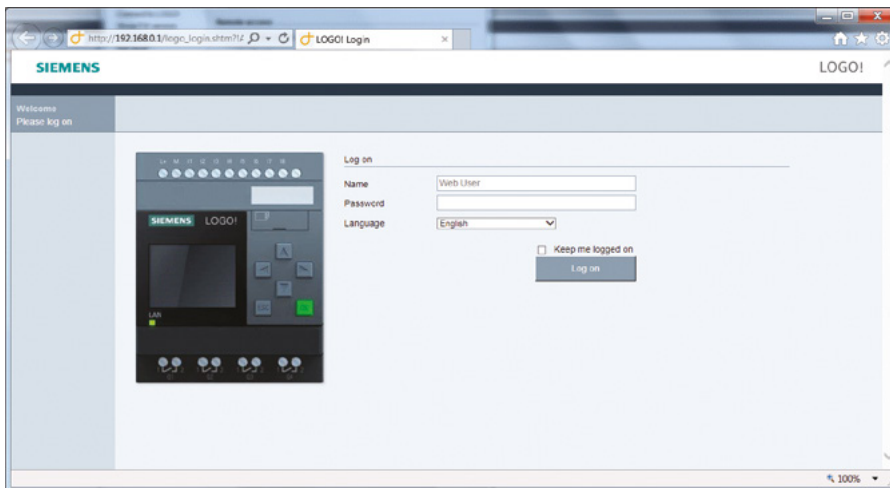
Interfejs łączący Logo! ze światem zewnętrznym był chyba jedną z najbardziej oczekiwanych przez użytkowników modyfikacji. Sygnałem, że producent słucha opinii użytkowników było wprowadzenie do sprzedaży wersji Logo! 7, ale przyznam się, że mnie przekonała dopiero wersja ósma. Po pierwsze, swoją ceną, a po drugie świetnym środowiskiem projektowym. Nie bez znaczenia był też dla mnie fakt, że producent zamiast wyposażyć nowy mikrosterownik jedynie w interfejs producent poszedł o krok dalej i dał użytkownikowi do dyspozycji serwer

Web, a za jego pomocą – wirtualne wyświetlacze i przyciski sterownika oraz panelu operatorskiego TDE. Co ciekawe, można z nich korzystać nawet wtedy, gdy nie mają one fizycznych odpowiedników – wystarczy sama deklaracja w programie. Daje to szerokie możliwości oraz wielce poprawia funkcjonalność Logo!

Wykonując aplikacje ze starszymi wersjami Logo! (v5 i v6) mocno „cierpiełem” z powodu braku możliwości łączenia sterowników i przesyłania pomiędzy nimi danych. Aby przekazać parametr z jednego

systemu do drugiego, dobudowywałem przystawki, wykonywałem coś w rodzaju szyny równoległej i uprawiałem różne inne „magiczne sztuczki”. Wiem, te możliwości mają większe sterowniki, ale cena środowiska programistycznego przeważającej większości z nich była dla mnie odstraszałą. Nie budowałem żadnych rozległych instalacji przemysłowych, ale nieskomplikowane urządzenia automatyki, realizujące pewne funkcje sterowania, przeważnie przeznaczone do własnego użytku. A wówczas, jeśli samodzielnie zbudowałem odpowiednią przystawkę, to koszty jej wykonania były znacznie niższe od ceny samego IDE. A przecież należało jeszcze dodać do tego droższy sterownik.

Interfejs Ethernet daje też inne możliwości. Do jego gniazda można dołączyć typowy, bezprzewodowy Access Point, połączyć go z komputerem lub siecią, z którą połączenie ma nasz komputer i już możemy bezprzewodowo monitorować pracę



Rysunek 1. Ekran główny serwera Web

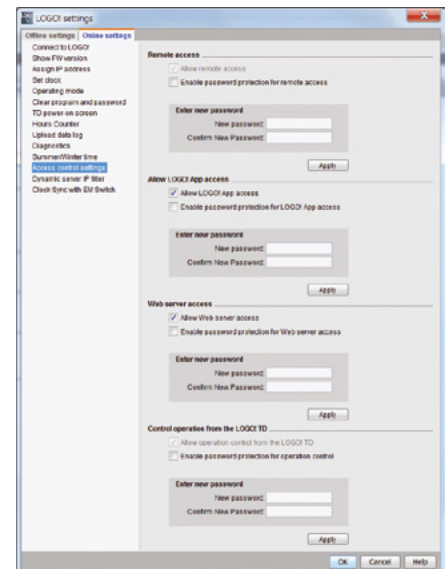
sterownika, modyfikować jego oprogramowanie, sterować procesami za pomocą wirtualnych przycisków dostępnych via serwer Web wbudowany w Logo! Oczywiście, można też „wystawić” interfejs Logo! na zewnątrz sieci i mieć do niego dostęp poprzez Internet. Dzięki dostępowi przez dowolną przeglądarkę internetową można obsługiwać Logo! za pomocą tabletu, smartfona i innych urządzeń przenośnych (dla Androida opracowano specjalną aplikację). Wreszcie – można mieć dostęp do naszej instalacji poprzez sieć telefonii komórkowej i być rzeczywiście mobilnym. Jedynymi ograniczeniami są zdrowy rozsądek i kwestia bezpieczeństwa naszej instalacji. Co by nie powiedzieć, jest to ogromne ułatwienie tak dla programisty, jak i dla osoby obsługującej instalację.

Niegdyś, aby zmienić oprogramowanie, trzeba było połączyć się z Logo! za pomocą specjalnego kabla programującego. Jego długość wynosząca około 2-3 metry była poważnym ograniczeniem, które nieraz narządziło serwisanta czy programistę

na pracę na drabinie, w hałasie hali produkcyjnej i innych warunkach niekoniecznie sprzyjających pracy. A teraz można usiąść spokojnie w biurze lub wręcz na drugim końcu świata, w ciepłej i zacisznej wykonać niezbędne czynności.

Nowy sterownik Logo!, to znaczy taki prosto ze sklepu, ma domyślnie ustawiony adres IP na wartość 192.168.0.1 oraz 24-bitową maskę sieci (255.255.255.0). Jeśli nie chcemy zmieniać adresu IP, to musimy zadbać, aby komputer PC używany przez nas do komunikacji z Logo! miał adres z puli adresów sterownika PLC. Następnie, należy połączyć oba urządzenia kablem Ethernet lub za pomocą sieci (dowolnego switcha sieciowego), uruchomić przeglądarkę internetową i na pasku adresu wpisać adres Logo! Pozostawiłem adres domyślny nadając interfejsowi Ethernet swojego komputera adres o 1 wyższy, więc u mnie będzie to `http://192.168.0.1/`. Po zatwierdzeniu adresu na ekranie ukaże się obraz jak na **rysunku 1**.

W zasadzie na tym możemy skończyć chyba, że ciekawość pcha nas dalej. Jednak



Rysunek 2. Okno umożliwiające skonfigurowanie sposobu dostępu do sterownika Logo!

nie mając żadnej aplikacji w pamięci sterownika po kliknięciu *Log On* będziemy mogli jedynie zobaczyć informacje na temat sterownika, ekran zawierający zmienne, wizualizację sterownika oraz panelu Logo! TD i to wszystko. Wydaje mi się, że szkoda na to czasu i jeżeli otworzył się nam strona, jak na rys. 1, to szkoda czasu na dalsze oglądanie tego, co tam jest dopóty, dopóki nie dowiemy się jak z tych nowych właściwości skorzystać. Zaczniemy więc od zaprzęgnięcia serwera web do pracy.

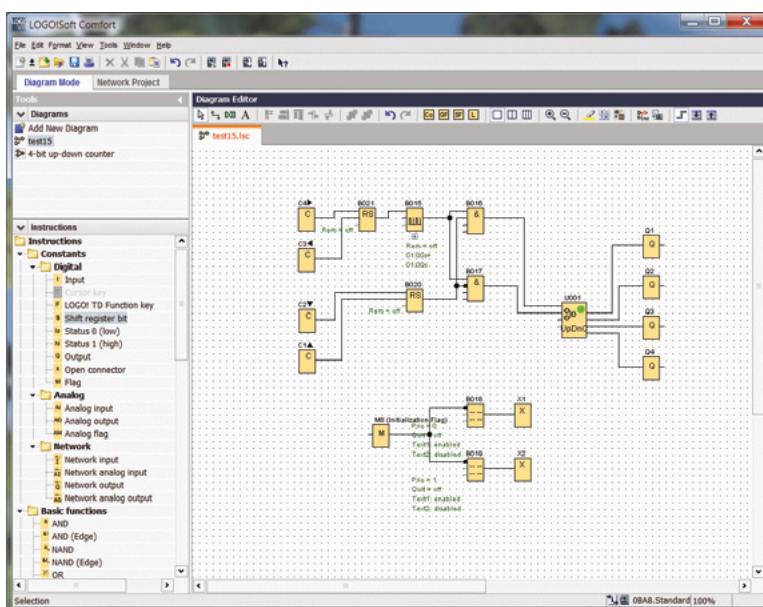
Serwer Web

Przed użyciem serwera web wymaga skonfigurowania. Nie należy się tego obawiać, gdyż tę czynność uproszczono do maksimum. W praktyce polega ona na zaznaczeniu pola wyboru i ewentualnym wpisaniu hasła dostępu. Nastawy serwera są wykonywane za pomocą programu Logo Soft Comfort. Po jego uruchomieniu należy wybrać z menu *Transfer* → *Access Control*, wskazać lub wpisać adres naszego sterownika i kliknąć na OK. Na ekranie zostanie wyświetlone okno, jak na **rysunku 2**.

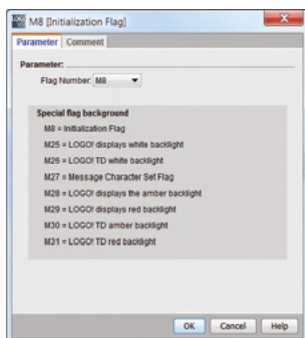
Pozycja *Allow remote access* w sekcji *Remote Access* jest oznaczona szarym kolorem, więc nie mamy do niej dostępu i nie można wyłączyć dostępu zdalnego. Jedyną możliwością jest zabezpieczenie tego dostępu za pomocą hasła, co można zrobić oznaczając pole wyboru *Enable password protection for remote access* i wpisując hasło.

Kolejna sekcja – *Allow LOGO! App Access* dotyczy dostępu do sterownika za pomocą aplikacji dla systemu Android. Jeśli nie zamierzamy z niej korzystać, to należy to pole wyboru odznaczyć. Jeśli tak, to dobrą praktyką jest wpisanie hasła.

Sekcja *Web server access* jest dla nas w tym momencie najbardziej interesującą,



Rysunek 3. Schemat licznika dwukierunkowego złożonego z bloków FBD



Rysunek 4. Okno z wykazem flag specjalnych dostępnych w programie

więc należy zaznaczyć pola wyboru *Allow Web server access* oraz *Enable password protection for Web server access* i w pustych liniach dwukrotnie wpisać hasło (*New password* oraz *Confirm New Password*) i kliknąć na *Apply*. Przyjmijmy, że na początek wpisujemy 0000.

Sekcja *Control Operation from LOGO! TD* – podobnie jak *Allow remote access* – jest oznaczona szarym kolorem, więc jedynie możemy zabezpieczyć korzystanie z panelu operatorskiego hasłem, ale nie możemy go wyłączyć. Hasło ma znaczenie, jeśli korzystamy z panelu fizycznego, dołączonego do sieci Logo! Dostęp do panelu wirtualnego za pomocą serwera Web jest i tak zabezpieczony hasłem, więc jeśli nie używamy urządzenia fizycznego, to nie musimy zaprzątać sobie tym głowy.

Zajmijmy się teraz wykonaniem jakiejś niekomplikowanej aplikacji, dzięki której będziemy mogli zapoznać się z pracą serwera Web. Posłużymy się licznikiem z poprzedniego przykładu. Niech przyciski „w górę” i „w dół” panelu TD i sterownika posłużą do zmiany kierunku liczenia, odpowiednio: w dół i w górę. Przycisk „w lewo” zatrzymuje generator, natomiast „w prawo”

– uruchamia. Na wyświetlaczu sterownika i panelu TD będą pokazywane zmienne robocze oraz stany wyjść. Niech będzie również możliwa zmiana częstotliwości, z jaką generator taktuje licznik.

Utworzony przeze mnie przykładowy program w postaci diagramu FBD pokazano na rysunku 3. Generator i licznik umieszczono w górnej części schematu FBD. Przyciski dołączone do wejść przerezutników RS (blok *Latching Relay*). Przerzutnik oznaczony B021 ma wyjście dołączone do wejścia *Enable* generatora astabilnego, natomiast B020 steruje kierunkiem zliczania sterując doprowadzaniem sygnału z generatora do wejść zliczania „w górę” i „w dół”. Wyjścia bloku UDF (*UpDnCnt*) są dołączone do wyjść sterownika – w związku z tym, że użyłem sterownika z wyjściami przekaźnikowymi, to poziom logiczny na wyjściu licznika będzie załączał lub wyłączał połączony z nim przekaźnik. Skupmy się teraz na tym, co znajduje się w dolnej części schematu.

Blok M8 to flaga ustawiana, gdy zostaną zakończone procedury inicjowania sterownika. Zwykle wykorzystuję ją do wyświetlania ekranu powitalnego lub uruchomienia jakichś procesów związanych z inicjowaniem wyjść, przerezutników itp.

Flagi to pewne specjalne bity sterownika, które mogą coś sygnalizować lub za pomocą, których można sterować pewnymi funkcjami, jak dla przykładu załączenie podświetlenia tła oraz jego kolor. Część z flag jest dostępna dla użytkownika, natomiast część jest zarezerwowana przez system i pełni pewne specjalne funkcje (jak na przykład użyta przeze mnie flaga *M8 = Initialization Flag*). Na rysunku 4 pokazano okno z wykazem flag specjalnych dostępnych w programie. Pojawia się ono po dwukrotnym kliknięciu na symbolu flagi ułożonej na schemacie FBD. Jak można się zorientować, jedynie M8 jest flagą specjalną, informacyjną, natomiast pozostałymi można sterować tym samym załączając pewne różne funkcje sterownika (na przykład – zmieniając M27 można zmienić

do wejścia *Enable* generatora dołączono M1, natomiast do wyjścia sterującego kierunkiem zliczania M2. Dzięki temu wyzerowanie flagi M2 oznacza kierunek zliczania „w górę”, natomiast ustawienie – „w dół”. Zajmijmy się teraz ekranami komunikatów. Na rysunku 5 pokazano zawartość bloku funkcyjnego B18. Ten blok nosi nazwę *Message* i służy do wyświetlania komunikatów na ekranie sterownika oraz panelu operatorskiego. Te bloki mają szereg możliwości, które pokrótce omówię.

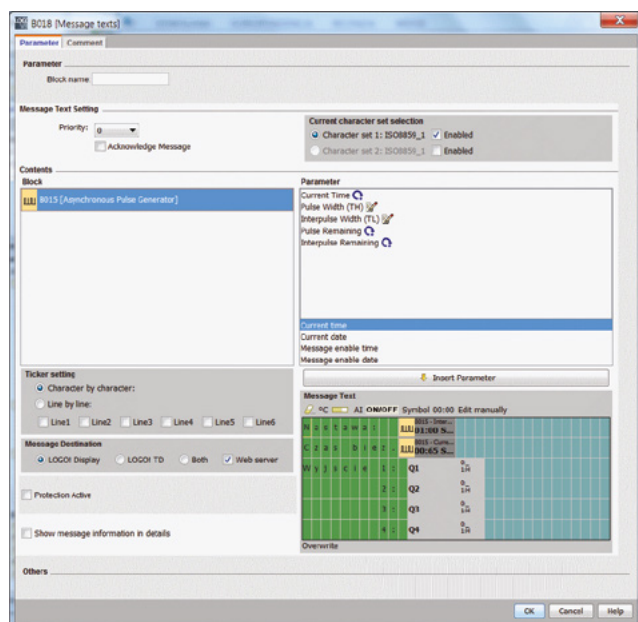
Sekcja *Parameter* umożliwia nazwanie bloku. Czasami, gdy program jest bardzo rozbudowany, nadawanie nazw jest pomocne. Jeśli składa się z kilku bloków, jak ten przykładowy, to możemy pozostawić pole puste.

Sekcja *Message Text Setting* pozwala na ustalenie priorytetu wyświetlania komunikatów (w praktyce – całych ekranów), jeśli jest ich wiele. Komunikaty o wyższym numerze są wyświetlane „na wierzchu”, a pomiędzy poszczególnymi ekranami można poruszać się za pomocą przycisków kursora. Ta opcja przyda się na przykład wtedy, gdy po załączeniu sterownika chcemy na przykład wyświetlić numer wersji programu i status urządzenia, a „pod spodem” ukryć ekrany zawierające informacje diagnostyczne. Oczywiście, przyda się nie tylko wtedy. Zaznaczenie pola *Acknowledge message* spowoduje, że sterownik po wyświetleniu komunikatu będzie oczekiwał na potwierdzenie jego przeczytania. Ta funkcja może przydać się zwłaszcza dla komunikatów związanych ze stanem pracy urządzenia. I ostatnia w tej sekcji – *Current character set selection* umożliwi nam wybranie aktywnego zestawu znaków dla komunikatów, który później może być przełączany ręcznie w programie lub za pomocą wspomnianej flagi M27.

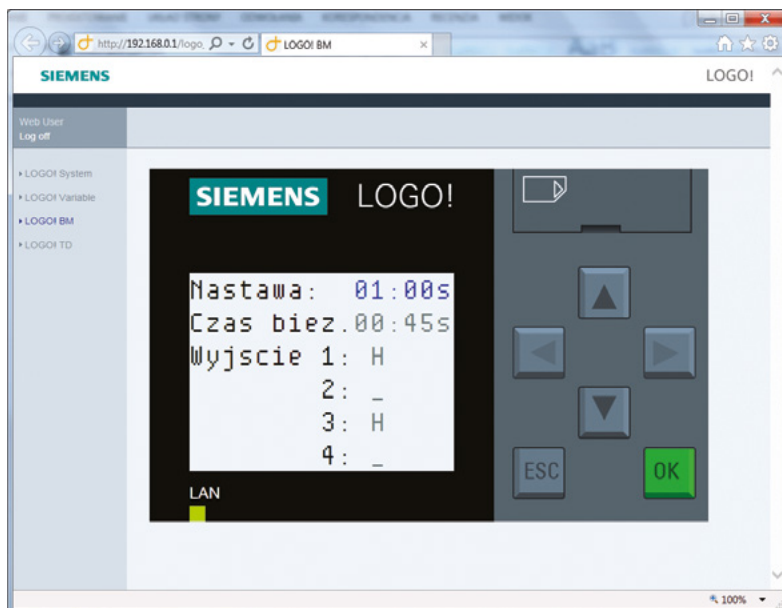
W sekcji *Content* można wskazać blok (*Blocks*), którego parametry (*Parameter*) będą wyświetlane.

Sekcja *Ticker setting* pozwala na ustalenie sposobu wyświetlania komunikatów, które nie mieszczą się na ekranie. Mogą one być wyświetlane (skrolowane) znak po znaku (*Character by character*) lub linia po linii (*Line by line*). W tym polu musimy też zaznaczyć, w której linii znajdzie się ten zbyt długi komunikat.

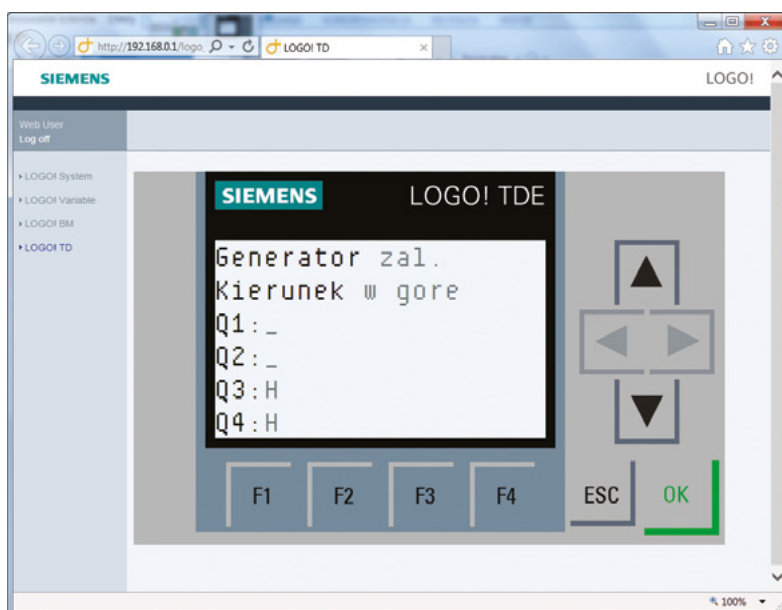
Wreszcie dotarliśmy do interesującej nas sekcji *Message destination*. Tu można wybrać miejsce, w którym będzie wyświetlany komunikat. Czy będzie to wyświetlacz sterownika Logo!, czy wyświetlacz panelu operatorskiego TD, czy oba i wreszcie, czy **ma być dostępny za pomocą serwera Web**. Co ciekawe, nawet jeśli nie mamy panelu operatorskiego fizycznie dołączonego do sterownika, to serwer Web również wyświetli go w przeglądarce. Nie ma również



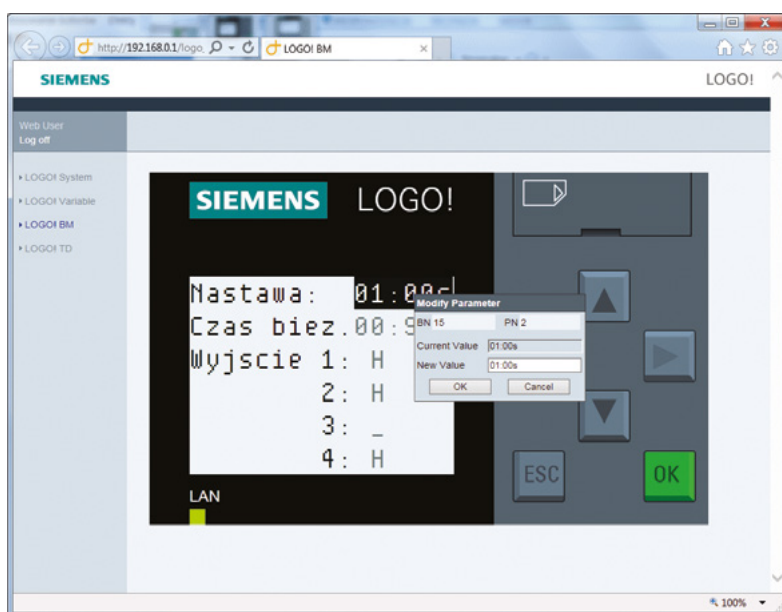
Rysunek 5. Zawartość bloku funkcyjnego B18



Rysunek 6. Wygląd strony Web, która jest odzwierciedleniem ekranu sterownika Logo!



Rysunek 7. Wirtualny panel operatorski Logo! TD



Rysunek 8. Edycja parametru za pomocą serwera Web

znaczenia, czy Logo! ma wbudowany wyświetlacz, czy też nie. W każdym wypadku serwer Web pracuje tak samo. Wystarczy zaznaczyć pole wyboru *Web server*.

Blok B019 jest taki sam, jak opisany wcześniej z tym, że dotyczy panelu operatorskiego TD i umieszczono na nim inne komunikaty. Jakże? Za chwilę pokażemy to za pomocą serwera Web.

Na **rysunku 6** i **rysunku 7** pokazano efekt pracy programy z tego przykładu. Pierwszy z rysunków przedstawia wygląd strony Web, która jest odzwierciedleniem ekranu sterownika Logo! Na drugim z rysunków zaprezentowano wirtualny panel operatorski Logo! TD. Co ciekawe, możemy klikać myszką na przyciski kursora i poprzez sieć zdalnie sterować naszym programem. Jeśli nie zablokujemy tej możliwości w programie, to dwukrotne kliknięcie na parametrze umożliwi jego edytowanie. Tu możemy zmienić odstęp pomiędzy impulsami (**rysunek 8**).

Zdaję sobie sprawę, że przedstawiony przykład nie wyczerpuje wszystkich aspektów i możliwości, a jedynie sygnalizuje niektóre. Stare przysłowie programistów mówi, że nie da się popsuć komputera z klawiatury. Podobnie jest ze sterownikiem Logo! Namawiam do samodzielnego eksperymentowania i poznawania możliwości tego ciekawego urządzenia.

Tworzenie aplikacji sieciowych

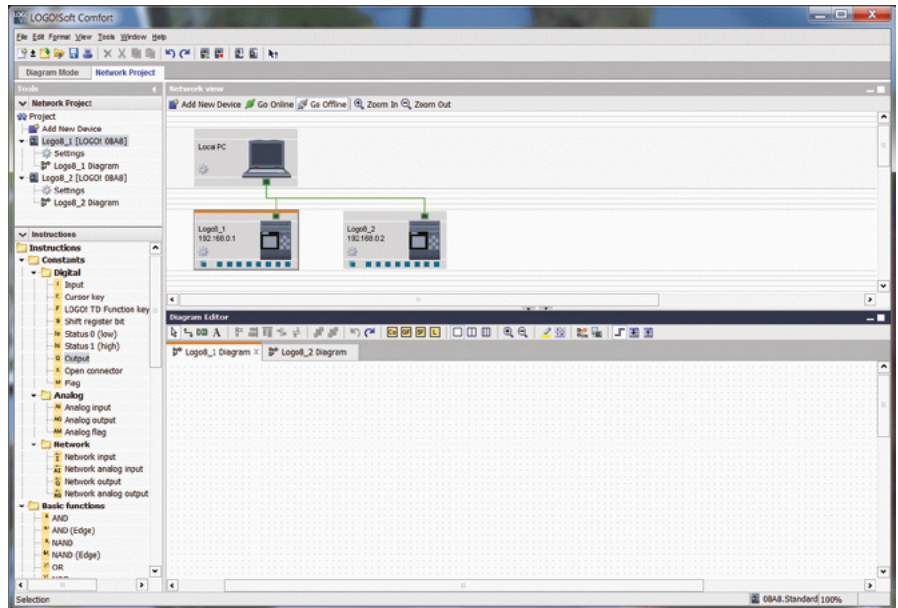
Sterownik Logo! 8 może pracować w sieci w trybie *master* lub *slave*. W trybie *slave* pełni on rolę jedynie urządzenia peryferyjnego, podczas gdy program jest wykonywany przez sterownik *master*. Ten tryb daje możliwość rozszerzenia dostępnej liczby wejść/wyjść oraz pozwala na rozproszenie urządzeń pobierających z jakiejś rozległej instalacji lub sterujących jej pracą. Jak wspomniano, w trybie *master* sterownik zarządza sterownikami pracującymi w trybie *slave* lub wymienia dane z innymi *masterami* realizującymi jakieś własne zadania. Spróbujmy prześledzić na przykładzie sposób, w jaki wykonuje się program sieciowy. Na początek wykonajmy jedynie jakiś nieskomplikowany program, który przekona nas, że komunikacja działa poprawnie i jednocześnie nauczy, jak z niej korzystać. Założmy, że w naszej sieci będą pracowały dwa sterowniki Logo! Pierwszy będzie dodawał kolejne naciśnięcia przycisku „w górę” i odejmował „w dół” od stanu licznika. Jeśli stan wewnętrznego licznika będzie zawierał się w przedziale 0...5, to kolor podświetlenia drugiego sterownika (dołączonego za pomocą sieci Ethernet) będzie biały. Jeśli w przedziale 6...10, to żółty, natomiast, jeśli więcej od 10, to czerwony.

Tworzenie programu należy rozpocząć od utworzenia nowego arkusza FBD (*File*

New Function block diagram). Następnie, na zakładce właśnie utworzonego arkusza, trzeba kliknąć na *Network Project*. Na pasku *Network View* klikamy na *Add New Device*. Wpisujemy adres IP pierwszego Logo – wybrałem domyślną nazwę proponowaną przez oprogramowanie oraz adres 192.168.0.1. Następnie ponownie klikamy na *Add New Device* i do projektu dodajemy jeszcze jedno Logo! Tu ponownie użyłem nazwy domyślnej oraz adresu o 1 większego, a więc 192.169.0.2. Przykładowy projekt powinien wyglądać, jak na **rysunku 9**. Można zauważyć, że zostały utworzone dwa arkusze, które nazywają się tak samo, sterowniki umieszczone na panelu widoku sieci. Na tych arkuszach będziemy teraz umieszczali odrębny program dla każdego ze sterowników.

Przykładowy program dla sterownika „1” wykonałem z użyciem dwóch bloków FBD *Up/Down Counter*: Blok licznika ma wejścia: sygnału taktującego, kierunku liczenia i zerujące. Wyjście licznika jest ustawiane, jeśli liczba impulsów zawiera się w przedziale ustawionym w oknie właściwości bloku FBD. Oprócz wymienionych liczników, dla realizacji zamierzonej funkcjonalności użyłem jeszcze bramek logicznych. Zadaniem jednej z nich jest wprowadzenie opóźnienia, natomiast pozostałych wypracowanie odpowiednich sygnałów wyjściowych. Dla nas ważniejsze jednak będzie to, że wyjścia bramek muszą sterować podświetleniem tła innego Logo!, dołączonego przez sieć Ethernet.

Jak pamiętamy, załączeniem podświetlenia oraz jego kolorem sterują odpowiednie flagi: za załączenie koloru białego odpowiada flaga M25, bursztynowego – M28, czerwonego – M29. Dla wygody podzieliłem ekran na dwie części. W tym celu należy kliknąć na pasku narzędziowym *Diagram Editor* na kwadratową ikonę, taką jakby miniaturową kartkę przedzieloną na dwie części. Do pustego okienka można, a nawet wręcz należy, przeciągnąć okienko drugiego programu – robimy to po prostu chwytając za zakładkę i przenosząc okno. Następnie, trzeba w nim umieścić wymienione wyżej flagi i po prostu przeciągnąć połączenia od flag do bramek lub od bramek do flag, z okna do okna (lub inaczej – z arkusza na arkusz) nie przejmując się tym, że połączenie jest przeciągane pomiędzy dwoma różnymi oknami. Oprogramowanie samo doda niezbędne konektory sieciowe oraz linię łączącą sterowniki w górnej części ekranu. Pokazano to na **rysunku 10**. Czy można było to zrobić łatwiej? Moim zdaniem nie. Co bardzo ważne, można uruchomić jednoczesną symulację pracy obu programów na obu arkuszach i przekonać się, że sieć faktycznie działa, a odpowiednie sygnały są przekazywane. A więc zanim zaprogramujemy lub kupimy sterowniki, możemy wykonać odpowiednie



Rysunek 9. Początek tworzenia projektu sieciowego

testy bez sprzętu, a jedynie z użyciem komputera PC.

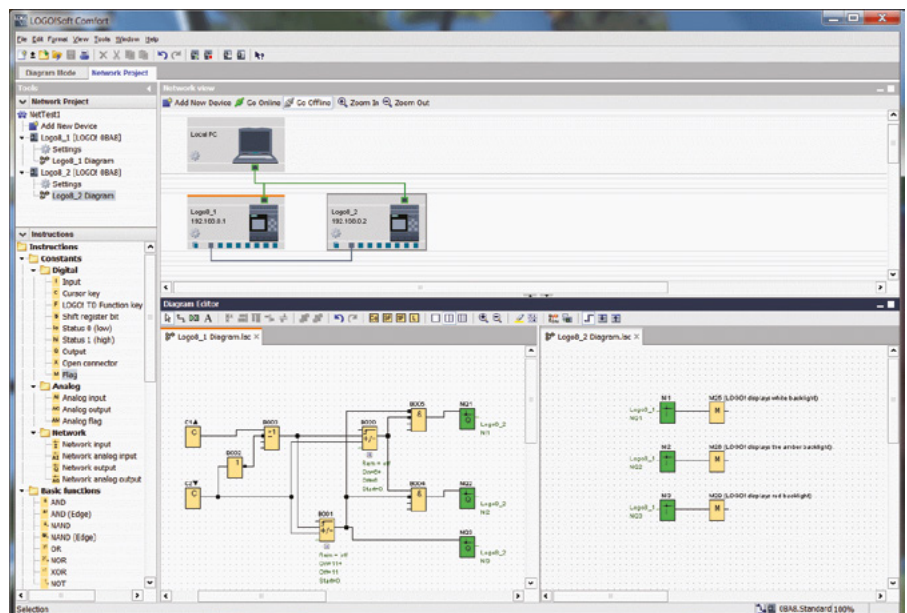
Oczywiście, można przy tym dla wygody można przy tym ukrywać lub wyświetlać panele, dzielić obszar roboczy na kilka części, umieszczać w nim kilka arkuszy, powiększać je, pomniejszać, dzielić. Myślę, że każdy wybierze sobie jakąś wygodną dla siebie i zależną od eksploatowanego sprzętu metodę pracy.

Podsumowanie

Każdy sterownik Logo! może współpracować z ośmioma innymi. Sterowniki wymieniane ze sobą dane są łączone na diagramie za pomocą pogrubionej linii. Współpraca sterowników w sieci przyda się nie tylko w sytuacji, w której urządzenia mają wymieniać dane lub wspólnie kontrolować jakiś proces, ale również wtedy, gdy potrzebujemy

„oddelegować” jakieś wejścia/wyjścia, rozszerzyć ich liczbę itp. Tworzenie sieci współpracujących ze sobą sterowników Logo! i co ważniejsze – oprogramowania dla niej – jest łatwiejsze, niż się tego spodziewałem! To, czego brakuje mi w momencie powstawania artykułu, to możliwość przekazywania poprzez sieć referencji (tych opisywanych w pierwszej części) oraz wysyłania danych do dowolnego urządzenia, innego niż S7 lub Logo!, a dołączonego pod danym adresem IP. Mógłby być to nawet rodzaj konwertera Ethernet/UART, do którego dałoby się przesyłać liczby lub łańcuchy znaków. Pytałem u źródeł – za jakis takie możliwości zostaną wprowadzone, a dotyczy to zwłaszcza przekazywania poprzez sieć parametrów (referencji) pomiędzy blokami.

Jacek Bogusz, EP



Rysunek 10. Gotowy program funkcjonujący w sieci – w dolnej części są widoczne konektory sieciowe