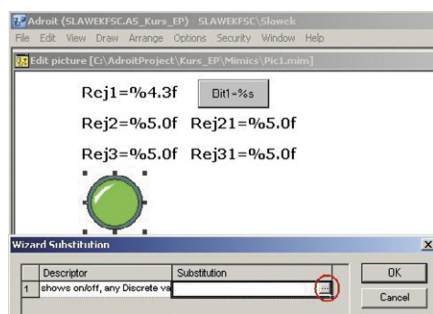


# Oprogramowanie SCADA w praktyce elektronika- automatyka, część 7

## Biblioteka elementów

W poprzednich częściach kursu zajmowaliśmy się konfigurowaniem agentów, driverów komunikacyjnych, tworzeniem prostych obiektów graficznych. Aby w pełni wykorzystać możliwości graficzne musimy sięgnąć do biblioteki elementów. Program posiada standardową bibliotekę elementów, która jest instalowana razem z oprogramowaniem. Znajdujące się w niej elementy mogą być wykorzystane w całości lub częściowo, tj. po dokonaniu przeróbek. Na początku możemy zobaczyć wszystkie elementy, które są umieszczone w bibliotece. Po uruchomieniu programu „Agent Server – timed licensed” oraz programu „Classic User Interface”, z menu wybieramy „File→Open” i otwieramy plik „\_WizardMenu.mim”, który jest standardowo umieszczony w katalogu „C:\Adroit\Wizards”. Klikając na poszczególne przyciski możemy oglądać kolejne elementy biblioteki. Przycisk „More Wizards” przechodzi do następnej strony biblioteki, natomiast przycisk „Shapes” wyświetla tylko elementy statyczne.

Aby wykorzystać dany element w synoptyce, należy go zaznaczyć (jest to równoznaczne z wykonaniem operacji na klawiaturze „Ctrl+C”), a następnie wkleić (skrót klawiaturowy „Ctrl+V”) na ekran synoptyki. Projekt synoptyki musi być

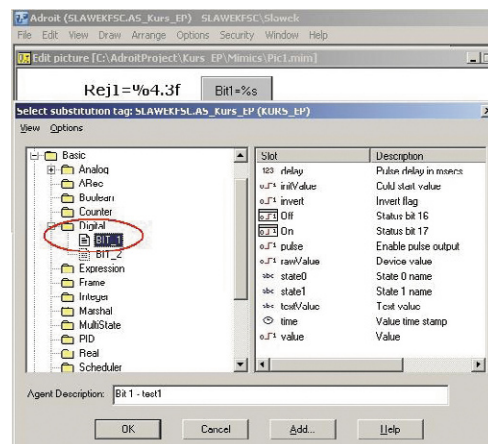


Rys. 1. Widok okna podczas wprowadzania nazwy zmiennej dla elementu z biblioteki

W poprzednich częściach kursu zostały wyjaśnione zasady tworzenia i konfigurowania agentów, sterowników komunikacyjnych i prostych obiektów graficznych. W tym odcinku zdobytą wiedzę rozszerzymy jeszcze o umiejętność korzystania z szablonów oraz biblioteki elementów.

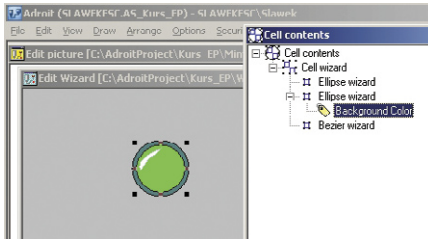
w tym czasie w trybie edycji. Po wykonaniu tej czynności pojawi się okno „Wizard substitution” (rys. 1), w którym definiuje się zmienną powiązaną z tym elementem. Można ją wpisać ręcznie lub wybrać z listy klikając na przycisk „...”. Zmienna musi być wcześniej zdefiniowana lub może być dodana w oknie wyboru „Select substitution tag” (rys. 2). Po zatwierdzeniu wyboru można sprawdzić, czy wszystko działa poprawnie poprzez przełączenie mimiki do trybu Runtime.

W tym przykładzie wykorzystaliśmy gotowy (przygotowany przez producenta oprogramowania) element biblioteki. Jeśli chcemy, możemy zmodyfikować element lub stworzyć całkiem nowy. Przed modyfikacją warto zrobić kopię elementu, a następnie zmieniać jego właściwości. Najpierw otwieramy nowe okno typu „Wizard”, a następnie w trybie edycji wstawiamy element skopiowany z biblioteki, który możemy dowolnie zmieniać. Większość elementów biblioteki składa się z kilku, czasami kilkudziesięciu elementów, które są zgrupowane (połączone) ze sobą. Jeśli jakiś element po zgrupowaniu zawiera właściwości dynamiczne (*behaviors*), to po „rozgrupowaniu” zostaną one utracone. Strukturę połączeń poszczególnych elementów w obiekcie można zobaczyć poprzez zaznaczenie myszką obiektu, wciśnięcie przycisku „Ctrl”, a następnie kliknięcie prawym lub lewym klawiszem myszy. Pojawi się takie okno jak na rys. 3. Możemy wówczas zmieniać dowolnie dany element obiektu. Część elementów są to statyczne elementy grafiki, niektóre



Rys. 2. Widok okna podczas dodawania zmiennej do elementu z biblioteki

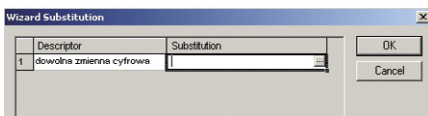
natomiast zawierają również dodatkowe właściwości. Klikając w tym oknie na „Background color” możemy zmienić właściwości tej funkcji. W tym przypadku mamy wpisane w polu „Descriptor: <shows on/off, any Discrete value>.value”, zmienimy to na tekst „<dowolna zmienna cyfrowa>.value”. Możemy również zmienić kolor, jaki ma się pojawić, jeśli zmienna ma wartość „1” i „0”. Po zatwierdzeniu wpisanych zmian zapisujemy bibliotekę w katalogu „Wizards” projektu i przechodzimy do trybu „Runtime”. Następnie wybieramy obiekt, klikając na niego myszką, przechodzimy do mimiki i wklejamy go przy użyciu skrótu klawiaturowego „Ctrl+V” lub z menu górnego „Edit→Paste”. Pojawi się okno „Wizard substitution” (rys. 4), w kolumnie „Descriptor” jest widoczny tekst „dowolna zmienna cyfrowa”, a w polu „Substitution” podajemy nazwę zmiennej. W nawiasach „< >” jest przekazywany parametr do obiektu. Użycie danego elementu



Rys. 3. Widok okna właściwości elementów obiektu

na synoptyce poprzez wprowadzenie nazwy agenta pozwala na przekazanie wartości tego agenta jako parametru. Takich parametrów może być więcej niż jeden i mogą być różnego typu. Do elementu typu „wizard” możemy również dodać inne dodatkowe właściwości.

Przykładem może być taka zmiana elementu, aby sygnalizował on poprawną pracę urządzenia oraz awarię. W tym celu należy przejść do edycji obiektu w bibliotece i kliknąć obiekt, trzymając jednocześnie klawisz Ctrl. W oknie „Cell contents” wybieramy właściwość „Background color” i zmieniamy tekst w nawiasach na „<Sygnal potw pracy>.value”, zatwierdzamy wprowadzone zmiany. Następnie klikamy na element „Ellipse wizard” doda-



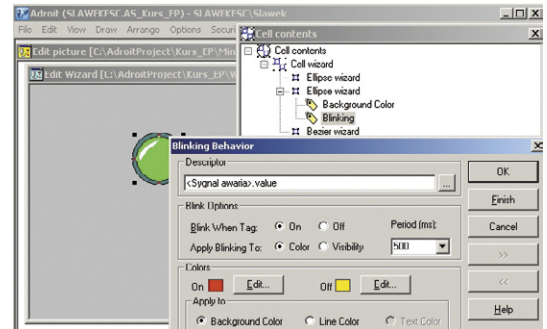
Rys. 4. Widok okna „Wizard Substitution” podczas wprowadzania nazwy agenta

jemy nową właściwość „Blinking” i wprowadzamy tekst „<Sygnal awaria>.value”. Możemy zmienić również kolor, typ i okres migania. Sposób wprowadzenia tej funkcji został przedstawiony na rys. 5.

Jeśli wprowadzimy zmiany w danym obiekcie biblioteki, to na ekranie synoptycznym istniejące obiekty zachowają swoje właściwości przed wprowadzonymi zmianami. Jeśli chcemy, aby elementy na synoptyce miały właściwości obiektów z biblioteki, to musimy je jeszcze raz wprowadzić. Warto pamiętać o tym, aby obiekty biblioteki stworzone dla danej aplikacji były umieszczone w folderze projektu. Dobrą zasadą jest również wprowadzanie komentarzy dla danego obiektu w postaci dodatkowego tekstu. W tym przykładzie w konfiguratorze agentów zostały ponadto stworzone zmienne typu „Digital” o nazwie „AWARIA” i „POTW\_PRACY”. Po wprowadzeniu obiektu z biblioteki wprowadzamy nazwy zmiennych w oknie „Wizard Substitution”. Widok okna po wprowadzeniu nazw zmiennych przedstawiono na rys. 6.

## Szablony

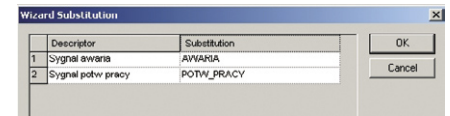
Kolejnym elementem oprogramowania są szablony (*mimic templates*). Są one zasobnikami obiektów graficznych umieszczanych na synoptyce, lub innych zdefiniowanych obiektów, i mogą być wielokrotnie używane w da-



Rys. 5. Widok okna podczas dodawania nowej właściwości do obiektu

nym projekcie. Zmiany wprowadzone w jednym szablonie znajdują odzwierciedlenie wszędzie tam, gdzie dany szablon został użyty. Jest to szczególnie użyteczne dla obiektów, które mają te same lub podobne właściwości i są wielokrotnie wykorzystywane w tym samym projekcie. Przykładem mogą być silniki, mieszała, pompy, zawory itp.

W oprogramowaniu Adroit szablony mogą być obrazy synoptyczne, okna wykresów, alarmów lub zdarzeń. Każda nowo utworzona synoptyka może być szablonem. Z kolei utworzona mimika, wykres lub okno alarmowe mogą być przerobione na



Rys. 6. Widok okna „Wizard Substitution” po wprowadzeniu nazw agentów



**AT Control System**  
ul. Nowiny 56B  
80-020 Gdańsk  
tel/fax (058)3065391

**NOWOŚĆ**





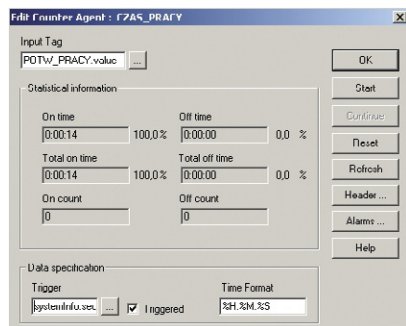
- 4 wejścia analogowe prądowe lub napięciowe,
- 2 wyjścia analogowe prądowe lub napięciowe,
- 12 wejść binarnych DC,
- 8 wyjść binarnych przekaźnikowych,
- pamięć programu 9K kroków (wbudowana),
- obsługa 76 punktów (moduły rozszerzeń),
- czas wykonywania instrukcji - kilka usek,
- 2 porty komunikacyjne RS232/485,
- wbudowany 1 szybki licznik 8 kHz,
- wyjście impulsowe 5 kHz,
- zegar RTC,
- 8 pętli regulacji PID,
- protokół użytkownika, Modbus RTU Master lub Slave w COM2;
- Modbus RTU Slave w COM1,
- darmowe oprogramowanie WinGPC

**Seria NX<sub>7</sub>**

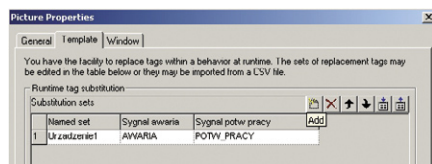
**Dystrybucja, doradztwo i serwis.**

• Automatyka kontrolno-pomiarowa   • Panele operatorskie   • Sterowniki PLC  
 • Wizualizacja SCADA   • Układy napędowe   • Zasilacze   • Radiomodemy

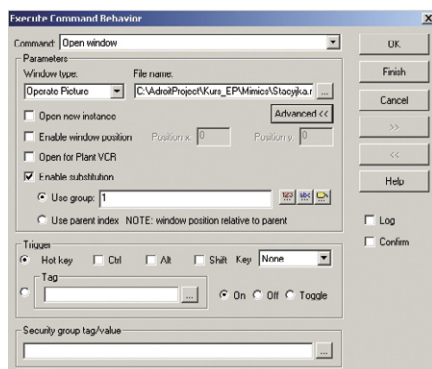
**www.atcontrol.pl**



Rys. 7. Widok okna konfiguracyjnego agenta typu „Counter”



Rys. 8. Widok właściwości zakładki Template



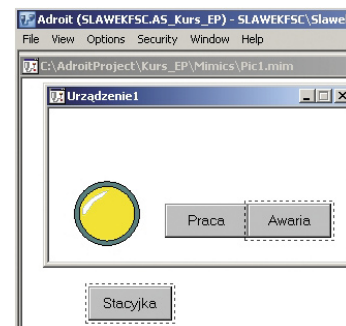
Rys. 9. Widok okna konfiguracyjnego funkcji „Execute Command”

szablon. Najczęściej są używane okna synoptyk do tworzenia stacyjek dla obiektów graficznych. Przedstawimy przykład, w którym po naciśnięciu przycisku pojawi się okno, w którym będzie przedstawiona zmienna analogowa oraz cyfrowa z możliwością interakcji ze strony użytkownika.

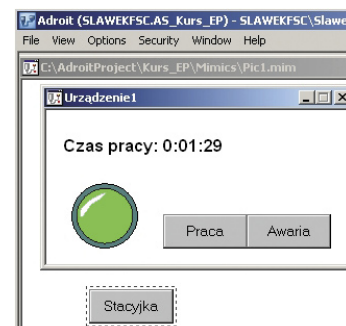
Najpierw tworzymy nowe okno, wybierając z menu górnego „File→New→Picture”. Zmniejszamy wielkość okna tak, aby nie zajmowało dużo miejsca. Wstawiamy dwa elementy typu przycisk „Button” (Ctrl+F8) oraz wykorzystujemy element typu lampka stworzony wcześniej w bibliotece elementów. Jeden przycisk będzie symulował potwierdzenie pracy urządzenia (zmienna typu „Digital” o nazwie „POTW\_PRACY”), a drugi będzie symulował awarię (zmienna typu „Digital” o nazwie „AWARIA”). Lampka będzie odzwierciedlała stan urządzenia. Dodatkowo

będziemy wyświetlać zmienną analogową typu „Counter”, która będzie liczyła czas pracy urządzenia. Aby to zrobić, należy stworzyć nową zmienną typu „Counter” o nazwie „CZAS\_PRACY”, a jako parametr wejściowy (Input Tag) podajemy zmienną typu „Digital” o nazwie „POTW\_PRACY”. W polu „Trigger” wybieramy zmienną „SystemInfo.second”, zaznaczamy pole „Triggered” i uruchamiamy agenta przyciskiem „Start”. Opcja „Triggered” określa, kiedy zmienna ma być uaktualniana na synoptyce. Na rys. 7 został przedstawiony widok okna z konfiguracją tej zmiennej. Zatwierdzamy wprowadzone zmiany i przechodzimy do szablonu. Jeden z przycisków nazywamy „Praca” i dodajemy mu właściwość „Operator Action”, w polu Tag wpisujemy „<Signal potw pracy>.value”, typ akcji (Action) ustawiamy na „Digital Toggle”. Drugi przycisk o nazwie „Awaria” konfigurujemy podobnie, z tym że w polu tag wpisujemy „<Signal awaria>.value”. Dla lampki w oknie „Wizard Substitution” wpisujemy odpowiednio „<Signal awaria>” oraz „<Signal potw pracy>”. Po wstawieniu tych elementów trzeba zapisać plik, np. pod nazwą *Stacyjka.mim*. Następnie klikamy na oknie w miejscu poza obiektami graficznymi, tak aby pokazały się właściwości tego obrazka lub wybieramy z menu „View→Properties”. Przechodzimy do zakładki „Template” i w kolumnie „Named set” wprowadzamy dowolną nazwę np. „Urządzenie1”. W następnej kolumnie „Signal awaria” wprowadzamy nazwę zmiennej, która będzie ten stan sygnalizować – w tym przykładzie będzie to „AWARIA”. W kolumnie „Signal potw pracy” wprowadzamy lub wybieramy z listy zmienną „POTW\_PRACY”. Jeśli będziemy więcej urządzeń, to dodajemy następne przyciskiem „Add”. Na rys. 8 przedstawiono widok okna podczas konfiguracji w zakładce „Template”.

Po zatwierdzeniu zmian należy przejść do głównego obrazu synoptyki i wstawić nowy przycisk, np. wprowadzając nazwę „Stacyjka” oraz dodać nową funkcję „Execute Command”. W polu „Command” wybieramy funkcję „Open window”, w parametrach podajemy jako typ okna „Operate picture” i polu „File name” wybieramy nasz plik o nazwie *Stacyjka.mim*. Klikamy przycisk „Advanced”, zaznaczamy opcję „Enable substitution” i wpisujemy w polu „Use group” wartość



Rys. 10. Widok okna szablonu synoptyk w trybie „Runtime”



Rys. 11. Widok okna szablonu z dodatkowym agentem typu „Counter”

1. Na rys. 9 został przedstawiony widok okna z konfiguracją funkcji „Execute command”. Zapisujemy zmiany i przełączamy do trybu „Runtime”. Po przyciśnięciu przycisku „Stacyjka” powinno pojawić się okno, takie jak na rys. 10.

Następnie dodamy do okna „Stacyjka” zmienną typu „Counter”, która będzie wyświetlać czas pracy urządzenia. Wstawiamy obiekt typu „Text” (Ctrl+F7), wpisujemy tekst „Czas pracy=%s”, dodajemy funkcję „Display value” i wprowadzamy w polu „Tag” zmienną z parametrem „<Czas pracy calc>.onTotalTime”. Jako wyświetlany slot został wybrany „onTotalTime”, który zawiera wartość czasu załączenia w formie ciągu znaków typu „String”. Po zatwierdzeniu przechodzimy do właściwości okna „View→Properties” i zakładki „Template”. Pojawiła nam się następna kolumna o nazwie „Czas pracy calc”, wprowadzamy nazwę zmiennej „CZAS\_PRACY”. Po zapisaniu zmian, przechodzimy do trybu „Runtime” i powinno pojawić się okno (rys. 11) z wyświetlaną wartością czasu pracy.

Naszą stacyjkę możemy powielać poprzez dodawanie następnych urządzeń w zakładce „Template” szablonu synoptyki „Stacyjki.mim”.

**Sławomir Kacprzak**  
**AT Control System**