

Ćwiczenie 4

Celem ćwiczenia jest poznanie składni języka ST stosowanego w automatyce budynków inteligentnych

Język ST

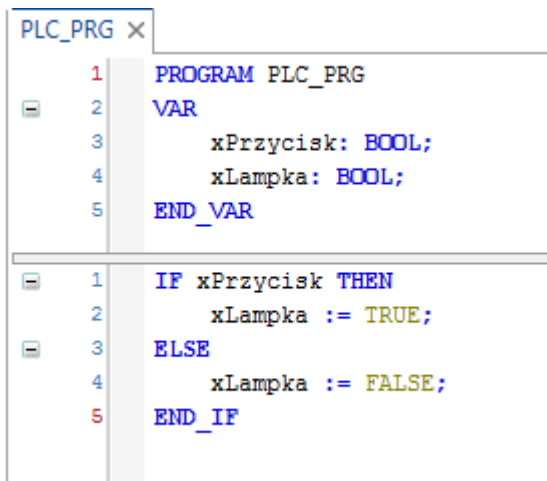
ST jest językiem programowania zdefiniowanym w trzeciej międzynarodowej normie IEC 61131-3, która opisuje graficzne i tekstowe języki programowania dla sterowników PLC. W odróżnieniu od graficznych języków programowania, bazuje on na tekście. Dzięki temu zajmuje dużo mniej miejsca niż graficzny język programowania, a logika programu jest łatwiejsza do rozszyfrowania i zrozumienia.

Najczęściej bloki funkcyjne są pisane w języku ST, a program główny w języku graficznym, np. FBD. Sterowniki WAGO także posiadają możliwość programowania w języku ST. Jak powszechnie wiadomo, najlepszą formą nauki jest praktyka. Dlatego WAGO posiada w ofercie zestawy startowe, które idealnie nadają się do praktykowania umiejętności programistycznych.

Języki wysokiego poziomu

Można zauważyć podobieństwa między nimi a ST. Składnia języka ST jest bardzo podobna do składni języków wysokiego poziomu i zawiera w sobie pętle, zmienne, instrukcje warunkowe i operatory.

Poniżej przedstawiono przykład programu PLC napisanego w ST.

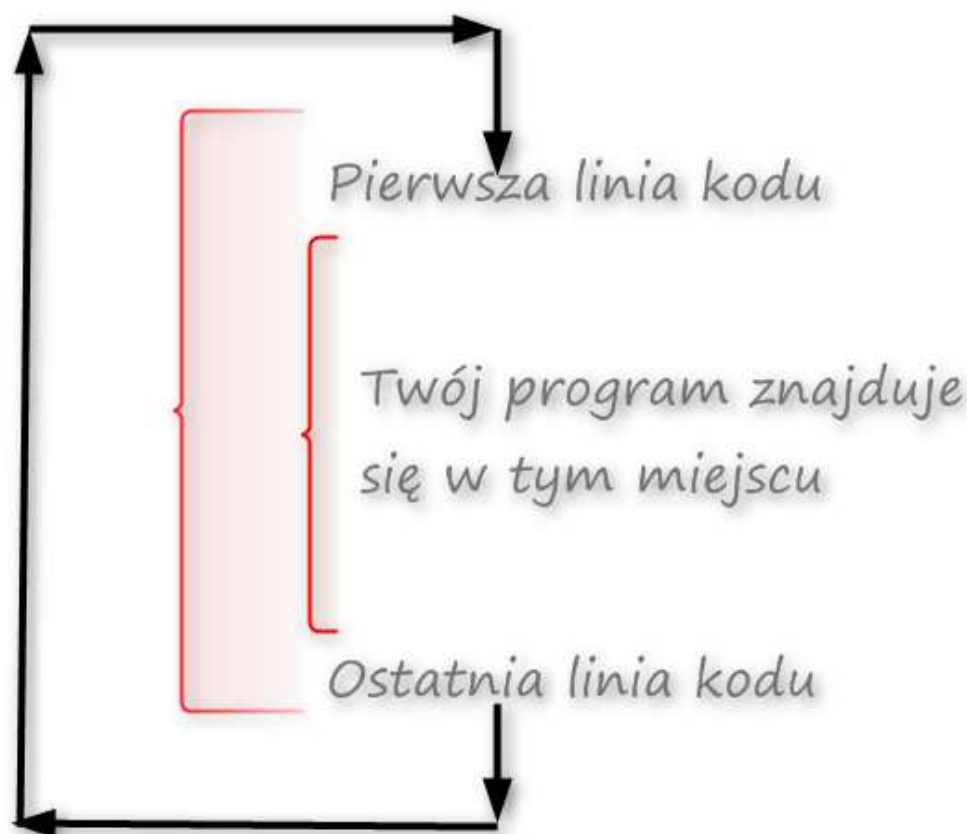


```
PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3      xPrzycisk: BOOL;
4      xLampka: BOOL;
5  END_VAR

1  IF xPrzycisk THEN
2      xLampka := TRUE;
3  ELSE
4      xLampka := FALSE;
5  END_IF
```

Program zawiera dwie zmienne. Sprawdzany jest stan przycisku i jeśli jest typu TRUE (prawda) to xLampka przyjmie wartość TRUE, w przeciwnym wypadku przyjmie wartość FALSE.

Przepływ programu



Jak widać program wykonywany jest w nieskończonej pętli.

Program analizowany jest od pierwszej linii kodu. Analiza kodu wykonywana jest linia po linii. Cała operacja odbywa się cyklicznie. Jeżeli sterownik osiągnie ostatnią linijkę, to wcale nie oznacza, że jest to koniec programu. Cykl sterownika rozpocznie się od nowa i program będzie wykonywany tak długo, jak sterownik będzie ustawiony w trybie RUN (wykonywanie programu przez PLC). Analogicznie wygląda to w języku drabinkowym i każdym innym języku programowania. W innym systemach celu zapętlenia wykonywania programu należy zaimplementować pętlę nieskończoną. Przy programowaniu sterowników PLC nie należy martwić się o implementację pętli nieskończonej dla wykonywania programu. Sterownik PLC zrobi to automatycznie.

Składnia języka ST

Składnia języka programowania to zbiór zasad określających jego budowę. W języku ST występuje wiele dwukropków, średników oraz innych symboli. Każdy symbol ma swoje znaczenie. Niektóre z nich to operatory, inne to funkcje lub zmienne.

W języku ST występuje kilka głównych zasad dotyczących składni. Nie trzeba uczyć się na pamięć wszystkich, ale warto zapamiętać najważniejsze:

- deklaracje zmiennych i komendy poleceń kończą się średnikami

- można używać małych oraz wielkich liter; kompilator nie widzi różnicy
- odstępy pomiędzy kolejnymi poleceniami nie pełnią żadnej funkcji, stosowane są w celu poprawy czytelności kodu

Należy także pamiętać, że komputer przetłumaczy program w języku ST na język zrozumiały dla sterownika PLC. Oprogramowania wyposażone są w kompilator, który tłumaczy kod na kod maszynowy zrozumiały dla sterownika PLC.

Kompilator analizuje składnię danego języka programowania. Jeżeli składnia będzie niepoprawna, to kompilator poinformuje o błędach.

Ważne! Jeżeli kompilator zauważy średnik, to oznacza dla niego koniec danego wyrażenia. Kompilator zawsze czyta wszystko, aż do momentu napotkania średnika i wtedy wykonuje dane wyrażenie.

Jak dodawać komentarze?

W języku ST występuje funkcja dodawania komentarzy.

```
1 // Komentarz dla pojedynczej linii
2
3 xZmienna := TRUE; (* Komentarz po wyrażeniu *)
4
5 (* Początek komentarza
6 .....
7 .....
8 Jest to komentarz dla wielu linii
9 .....
10 .....
11 Koniec komentarza*)
```

Komentarze są ważne, ponieważ objaśniają części składowe kodu programu. Wraz ze wzrostem umiejętności programistycznych można komentować mniej szczegółów, ale pozostać przy komentowaniu kluczowych części programu. Jednak najważniejsze jest to, żeby Twój kod programu był maksymalnie zrozumiały zarówno dla Ciebie, jak i innych użytkowników.

Wiemy już, czym charakteryzuje się język ST (Structured Text), na czym polega przepływ programu oraz główne zasady dotyczące składni i dodawania komentarzy. W tej części skupimy się przede wszystkim na zmiennych.

Tworzenie poleceń w języku ST

Język ST składa się z poleceń. Czym właściwie jest polecenie?

1 // Polecenie to wydanie rozkazu sterownikowi w jaki sposób PLC ma się zachować

Poniżej przedstawiono przykładowe polecenie jako przykład:

1 xPrzycisk := TRUE;

W powyższym poleceniu sterownik wie, że ma przypisać stan wysoki zmiennej „xPrzycisk”.

Kompilator zinterpretuje tę linię kodu jako jedno polecenie, ponieważ kiedy osiągnie średnik, to wie, że jest to koniec polecenia. Polecenia są oddzielane średnikami. Jest to główna zasada składni języka ST.

Używanie zmiennych w języku ST

Zmienne są definiowane między dwoma wyrazami VAR oraz END_VAR. Są to zarezerwowane słowa kluczowe. To znaczy, że nie można ich używać w żadnym innym miejscu w programie.

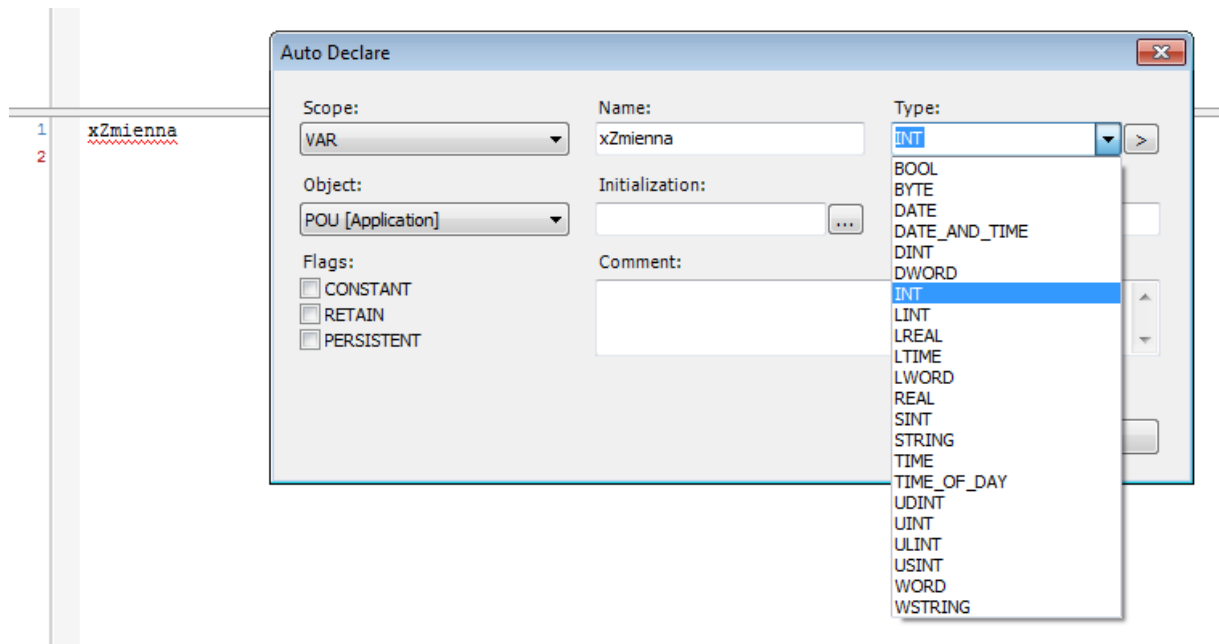
1 // Zmienna to miejsce gdzie można przechowywać dane

Zmienna zawsze przyjmuje określony typ danych. Przykładem jest zmienna typu BOOL, gdzie można przechowywać wartość TRUE lub FALSE (tylko dwa stany).

Zmienna zawsze ma przypisany odpowiedni typ danych i przechowuje wartość określonego typu danych. Programista może nadawać nazwy deklarowanym zmiennym.

```
1 PROGRAM POU
2 VAR
3     xZmienna : BOOL; // deklaracja zmiennej
4 END_VAR
```

Używając środowiska programistycznego CodeSys deklaruje się zmienne na dwa sposoby. Można wykonać operacje deklaracji zmiennych poprzez wywołanie okna deklaracji lub wpisanie „ręcznie” w polu deklaracji zmiennych.



Dodatkowo rozróżnia się podział zmiennych na zmienne globalne oraz zmienne lokalne.

1 // Zmienna lokalna – zdefiniowana i widoczna tylko w określonym bloku programu

1 // Zmienna globalna – widoczna we wszystkich miejscach w programie

Typy zmiennych używane w języku ST

Międzynarodowa norma IEC 61131-3 opisująca graficzne i tekstowe języki programowania dla sterowników PLC standardyzuje także typy danych na dwie kategorie: elementarne typy danych oraz pochodne typy danych.

Elementarne typy danych:

Integer

Float

Time

String

Pod każdym typem elementarnym definiuje się ich podtypy. I tak:

Dla zmiennych typu integer

Zadanie do wykonania

Stwórz w języku ST program, który w połączeniu z wizualizatorem umożliwi sterowanie zamkiem kodowym wyposażonym w 10 przycisków z cyframi od 0 do 9 oraz z przyciskiem klucz. Zamek kodowy ma zostać otwarty po wystukaniu prawidłowej sekwencji 4 cyfr szyfru w taki sposób aby w momencie wybrania 4 ostatniej cyfry kodu po naciśnięciu przycisku „klucz” załączyła się lampka zielona na 5 sekund. Gdy kod był nieprawidłowy powinna załączyć się lampka czerwona na 10 sekund. Dopiero po tym czasie możliwa będzie kolejna próba otwarcia zamka szyfrowego. Po 3 nieudanych próbach zamek blokuje się na 3 minuty.

Umieść komentarze w treści programu.