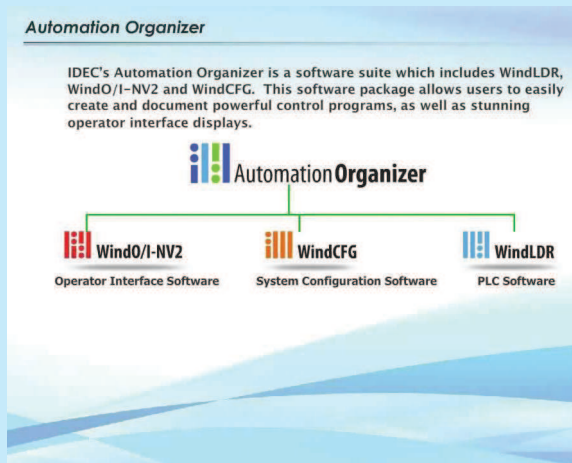




Fotografia 1. Sterownik IDEC SmartAXIS FT1A H40RSA



# Sterownik PLC SmartAXIS firmy IDEC (1)

**W ramach Klubu Aplikantów Próbek otrzymałem sterownik PLC SmartAXIS firmy IDEC. Stało się to przyczyną powstania kilku artykułów na jego temat. Na początek przedstawię ogólne możliwości i sposoby jego programowania. W kolejnych częściach przedstawię przykładowe rozwiązania układowe. A więc do dzieła...**

PLC (*Programmable Logic Controller*) to tłumacząc wprost – programowalny sterownik logiczny. Zgodnie z definicją podaną w Wikipedii, PLC to „uniwersalne urządzenie mikroprocesorowe przeznaczone do sterowania pracą maszyny lub urządzenia technologicznego”. Już po tym zdaniu widzimy, że sterowniki PLC są domeną automatyków i największe zastosowanie mają w automatyce przemysłowej. Ich szczegółowe możliwości przedstawię na podstawie sterownika, który otrzymałem w ramach KAP. Posłuży do tego celu IDEC SmartAXIS FT1A H40RSA. Jego wygląd pokazano na **fotografii 1**.

Sterownik jest wyposażony w:

- 18 wejść cyfrowych.
- 6 wejść analogowych.
- 4 wyjścia tranzystorowe PNP.
- 4 wyjścia przekaźnikowe o obciążalności do 10 A.
- 8 wyjść przekaźnikowych o obciążalności do 2 A.
- 200 timerów (1 sec, 100 ms, 10 ms, 1 ms).
- Szybki licznik mieszczący słowo 32-bitowe.
- Pamięć programu mieszcząca 47,4 kB kodu w języku drabinkowym lub 38 kB w języku FBD.
- 2 wyjścia PWM o maksymalnej częstotliwości 100 kHz.
- 2 wyjścia PWM o maksymalnej częstotliwości 5 kHz.
- 1 port USB do programowania, opcjonalny port Ethernet, opcjonalny adapter RS232/RS485.
- 1 złącze na karty SD.
- Monochromatyczny wyświetlacz LCD

Tak naprawdę sterownik PLC to procesor umieszczony w jednej obudowie z zasilaczem oraz układami elektronicznymi pozwalającemu pracować przy napięciu

i prądzie występującym w automatyce oraz wymieniać dane z innymi urządzeniami.

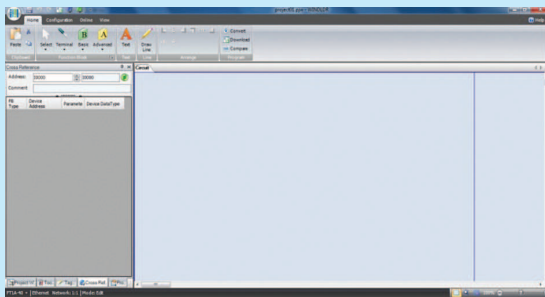
Sterownik PLC w wersji, którą otrzymałem jest zasilany napięciem 24 V DC. Pobiera z zasilania moc maksymalną 7,9 W. Wejścia analogowe mierzą napięcie z zakresu od 0 do 10 V DC. Napięcie jest mierzone z rozdzielczością 10-bitową. Wejścia cyfrowe akceptują napięcie maksymalnie 28,8 V DC. Sterownik ma wyjścia cyfrowe, tranzystorowe i przekaźnikowe.

W prawym górnym rogu obudowy umieszczono potrójne złącze śrubowe zasilania 24 V DC. Na prawo od jest cały rząd złączy śrubowych, na których wprowadzone są wejścia (najpierw 18 cyfrowych, a na końcu 6 analogowych). Natomiast cały rząd u dołu obudowy sterownika to wyjścia.

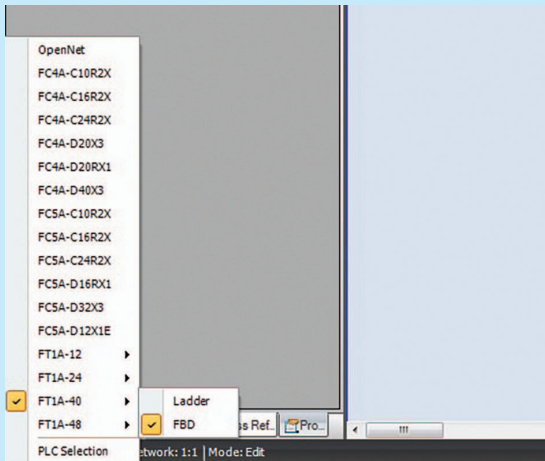
Na fotografii zaczerpniętej od producenta widać jeszcze zamontowane moduły interfejsów RS232 i RS485. Mój sterownik ich nie ma. Poszczególne peryferia omówię dokładnie podczas opisywania konkretnych rozwiązań układowych.

## Programowanie

Każdy producent sterowników wytwarza dla nich własne środowisko programistyczne. Większość jest płatna, ale zdarzają się też producenci oferujący je za darmo. Co oczywiste, dysponując środowiskiem programistycznym danego producenta sterowników możemy zaprogramować wyłącznie jego produkty. Zwykle jest tak, że środowiska są uniwersalne i możemy za ich pomocą zaprogramować wszystkie PLC oferowane przez tego producenta. Nie jest to jednak regułą.



Rysunek 2. Okno główne programu WindLDRv7



Rysunek 3. Wybór sterownika oraz języka programowania

Ważne jest to, że język programowania jest ustandaryzowany i niemal u każdego producenta programuje się identycznie. Współcześnie najbardziej popularne są 2 języki programowania: drabinkowy (LAD) oraz blokowy (FBD). Do tego dochodzi jeszcze możliwość umieszczania w obydwu językach skryptów realizujących bardziej zaawansowaną część sterowań.

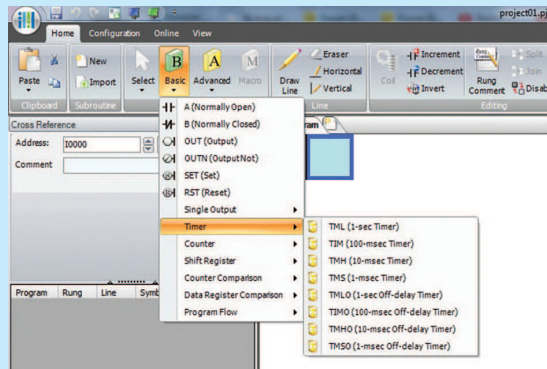
Producent sterownika, firma IDEC, dostarcza pakiet oprogramowania w formie 4 płytanych programów:

- *WindCFG* do konfigurowania systemów automatyki. Za pomocą tego programu można łatwo tworzyć schemat połączeń oraz dokumentację systemu (wykaz komponentów, kosztorys itp.).
- *WindLDR* do tworzenia oprogramowania do sterownika PLC.
- *WindO/I-NV3* do tworzenia oprogramowania dla paneli dotykowych.
- *WindO/I-NV2* –do oprogramowanie paneli dotykowych typu HG1F (4,6”), HG2G (5,7”), HG3F (10,4”) oraz z serii HG4F (12,4”).

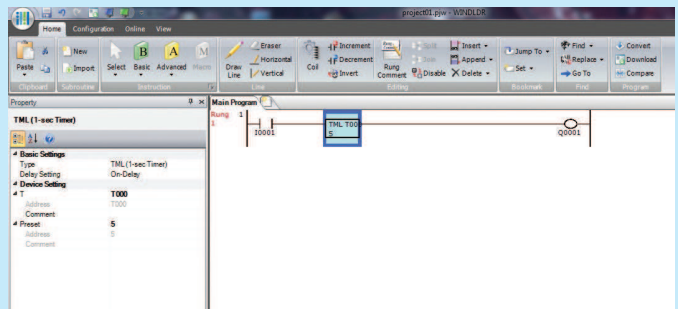
Do samego programowania sterownika wystarczy WindLDR. Ja korzystałem z wersji 7. Na **rysunku 2** pokazano ekran główny programu, który jest wyświetlany po uruchomieniu środowiska. Rozpoczynając nowy projekt musimy wybrać typ sterownika, dla którego będziemy tworzyli oprogramowanie oraz sposób, w jaki będziemy go programowali. W tym celu, w dolnym lewym rogu programu rozwijamy menu (**rysunek 3**) i wybieramy posiadany przez nas sterownik.

## LAD – język drabinkowy

Nazwa języka wzięła się stąd, że kod programu napisany w tym właśnie języku przypomina drabinę, a poszczególne instrukcje wykonywane są „szczebel po szczeblu”



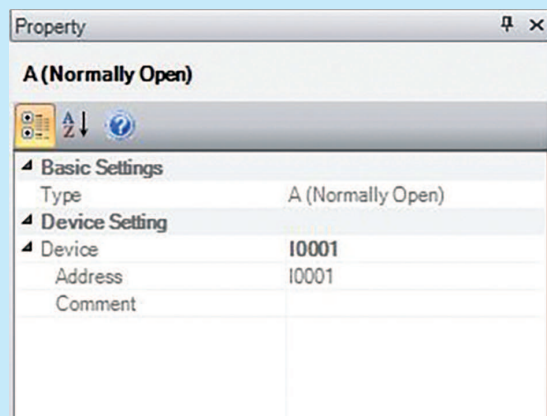
Rysunek 4. Dostępne elementy języka drabinkowego



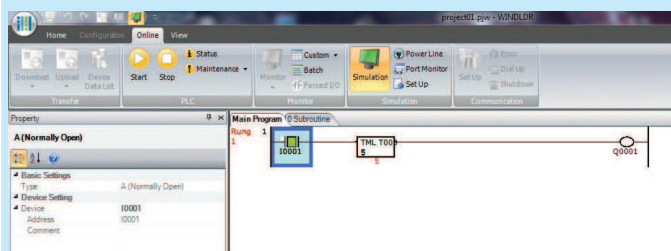
Rysunek 5. Załączenie wyjścia po 5 sekundach od wyzwolenia wejścia w języku drabinkowym

od góry do dołu, od strony lewej do prawej. Nie musimy uczyć się symboli na pamięć, gdyż program ma rozwijane menu z wszystkim dostępnymi elementami języka (**rysunek 4**).

W ramach demonstracji napiszmy kod, który po wyzwoleniu wejścia I0001 załączy wyjście Q0001 po czasie 5 sekund. Na **rysunku 5** pokazano taki przypadek. Widzimy również na nim, że każdy element ma swoje menu właściwości umieszczone z lewej strony. Na **rysunku 6** pokazano właściwości elementu licznika. Możemy tutaj wybrać licznik (mamy do dyspozycji 200 liczników), jak również jego interwał. Możemy również ustalić czy licznik po tym czasie ma swoje wyjścia włączyć lub wyłączyć. Możemy również wybrać rodzaj licznika (1-sekundowy, 100-milisekundowy, 10-milisekundowy, 1-milisekundowy) oraz dodać komentarz opisujący np. do czego służy w naszym programie ten licznik. Każdy element ma swoje własne właściwości. I tak wejścia mają pola numery wejścia, rodzaju wejścia (normalnie otwarte, normalnie zamknięte) oraz pole komentarza. Podobnie wyjścia z tą różnicą, że zamiast pola numeru wejścia, podajemy numer wyjścia.



Rysunek 6. Okno właściwości elementu licznika



Rysunek 7. Tryb symulacji

Muszę tutaj wyjaśnić pewną nieścisłość. W opisie języka drabinkowego użyłem pojęć „wejście” oraz „wyjście”. Tymczasem zamiast tych określeń w języku drabinkowym używa się nazw „styk” oraz „cewka”. Styk to wejście, a cewka to wyjście. To efekt zaszczości czasowej. Pierwsze sterowania w automatyce robiło się w ogromnych szafach wypełnionych przekaźnikami. Żeby łatwiej było automatom przyzwyczaić się do sterowników PLC, które dopiero co pojawiły się na rynku, zapożyczono nazwy wejść/wyjść od przekaźników. Ja, aby nie gmatwać opisu, użyłem typowych nazw, tj. wejście i wyjście.

Kiedy mamy napisany program, możemy go zasymulować. Wystarczy przejść do zakładki *Online* → *Simulation* i tam mamy ikonę służącą do przełączenia

się w tryb symulacji. Możemy również użyć skrótu klawiaturowego i wcisnąć F5. Na **rysunku 7** widzimy program w chwili rozpoczęcia symulacji. Po kliknięciu na wejście (styk) zmienia ono kolor na zielony, co oznacza, że zostało wyzwolone. Od tego momentu licznik odlicza 5 sekund, a po tym czasie zostaje wyzwolone wyjście (cewka).

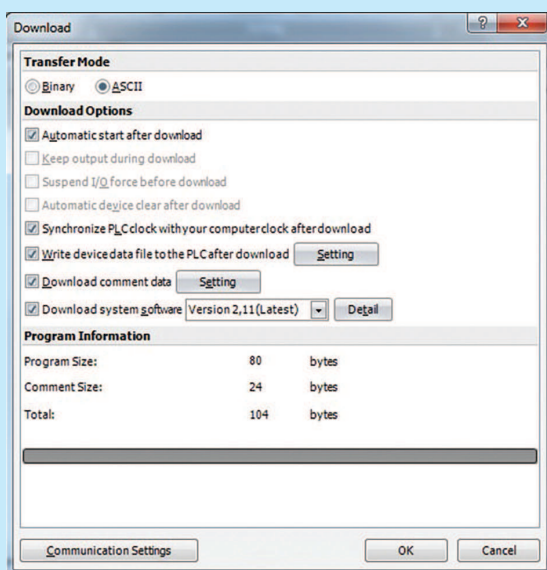
W celu wgrania oprogramowania do sterownika wybieramy skrót klawiaturowy Ctrl+F9. Program zostanie sprawdzony pod kątem poprawności i jeśli nie zostaną wykryte żadne błędy, to zostanie otwarte okno dialogowe z opcjami wgrывania oprogramowania do sterownika – pokazano to na **rysunku 8**. Aby wgrać program do pamięci sterownika należy wcisnąć F8. Następnie możemy przejść w tzw. tryb Monitora, podglądać i uruchamiać krok po kroku program w sterowniku i zarazem podglądać na komputerze stan sterownika oraz stan jego wejść/wyjść.

## FBD – język bloków funkcjonalnych

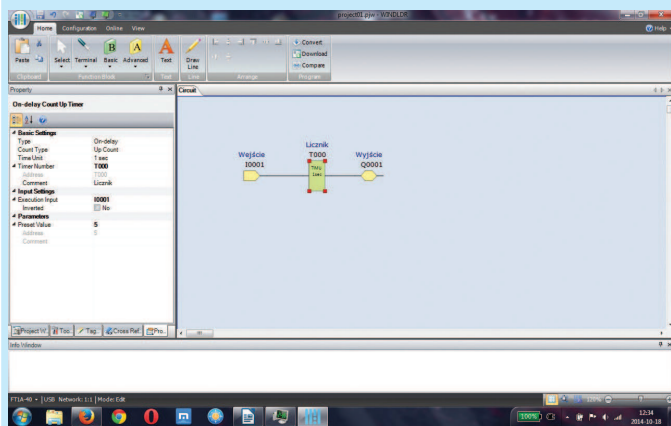
Aby programować w języku blokowym musimy ponownie się przełączyć, ale tym razem wybieramy FBD. I tym razem zrobimy prosty program, który po wyzwoleniu wejścia I0001 po 5 sekundach załączy wyjście Q0001. Widok takiego programu przedstawia **rysunek 9**. Jak widzimy, język FBD składa się z bloków funkcjonalnych, a programowanie w tym języku sprowadza się do połączenia odpowiednich bloczków ze sobą. Zaletą tego języka (oprócz przejrzystości programu) jest również to, że wybierając dany bloczek od razu widzimy, jakiego rodzaju sygnały musimy do niego doprowadzić.

Dostępne bloczki podzielone są 3 główne kategorie:

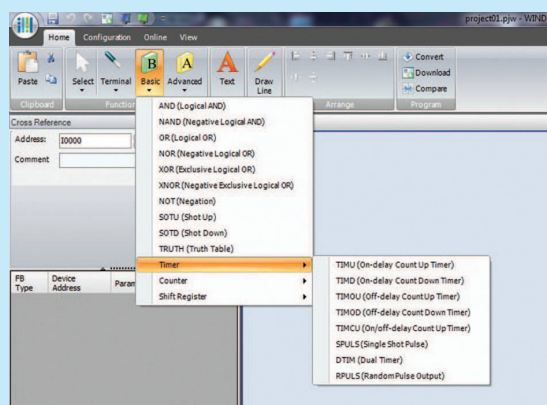
1. TERMINAL
  - I (Digital Input)
  - SM (Specjal Internal Relay)
  - A (Analog Input)
  - R (Shift Register)
  - Q (Digital Output)
  - M (Internal Relay)
2. BASIC (**rysunek 10**)
  - AND
  - NAND
  - OR
  - NOR
  - XOR



Rysunek 8. Okno dialogowe programowania sterownika



Rysunek 9. Załączanie wyjścia po 5 sekundach od wyzwolenia wejścia w języku blokowym FBD

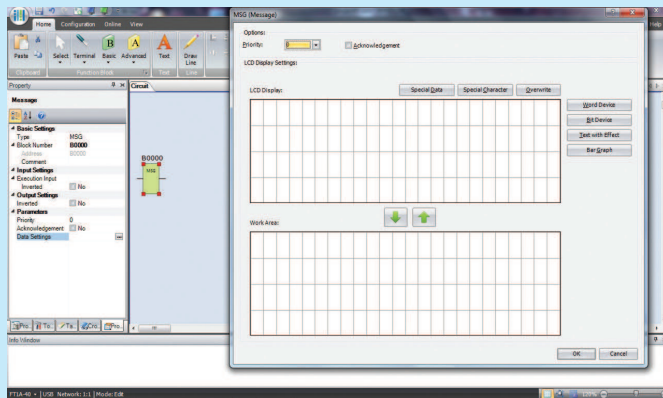


Rysunek 10. Elementy kategorii BASIC

- XNOR
  - NOT
  - SOTU (Shot up)
  - SOTD (Shot down)
  - TRUTH (Truth Table)
  - TIMER [TIMU, TIMD, ...]
  - COUNTER [CNT, CUD, HOUR]
  - SHIFT REGISTER [SFR]
3. ADVANCED
- DATA COMPARISON [CMP, STTG, RCMP]
  - DATA CONVERSION [ALT]
  - WEEK PROGRAMMER [WEEK, YEAR]
  - INTERFACE [MSG]
  - PULSE [PULS, PWM, RAMP, ZRN, ARAMP]
  - DATA LOGIN [DLOG, TRACE]
  - SCRIPT [SCRPT]
  - SPECIAL [HSC, RSFF]

Liczba dostępnych bloków jest duża, ale zarazem bardzo uproszczona. Na przykład, jeśli chcemy skorzystać z dostępnego w sterowniku wyświetlacza wystarczy skorzystać z bloczku ADVANCED → INTERFACE → MSG. Bloczek umieszczamy w naszym kodzie, a następnie w właściwościach klikamy na DATA SETTING, po czym wyświetli się nam specjalne okno, w którym będziemy mogli ustawić zmienne lub stałe wyświetlane na ekranie (rysunek 11).

Symulowanie, wgrywanie oraz monitorowanie oprogramowania w obydwu językach odbywa się identycznie.



Rysunek 11. Okno obsługi wbudowanego w sterownik PLC wyświetlacza monochromatycznego

## Podsumowanie

Jak widać programowanie sterownika PLC nie jest wcale trudną czynnością, zwłaszcza przy korzystaniu z języka FBD. W kolejnych częściach pokażę konkretne przykłady oprogramowania, oraz w jak prosty sposób za pomocą sterownika PLC stworzyć całkiem użyteczne urządzenie. Dla wszystkich chętnych którzy chcieliby spróbować swoich sił bez ponoszenia kosztów w związku z zakupem sterownika PLC, polecam symulatory PLC online, jak choćby ten: <http://www.plc-simulator.net/>.

Sławomir Kabat  
elektronika@sigaris.pl

REKLAMA

Fujitsu wireless  
modules

Bluetooth® low  
energy modules



[emea.fujitsu.com/components](http://emea.fujitsu.com/components)

FUJITSU

shaping tomorrow with you