

Programowanie Alpha2 (1)

Sprzęt i podstawy użytkowania IDE



Sterowniki PLC zwykle kojarzą się z drogimi urządzeniami do zastosowania w przemyśle. Oczywiście, są wśród nich i takie, jednak dzięki zdobyciom współczesnej elektroniki, ceny tych urządzeń są coraz niższe i częstokroć opłaca się stosować w aplikacjach, dla których dawniej tańsze było wykonanie specjalizowanego sterownika w oparciu o mikrokontroler.

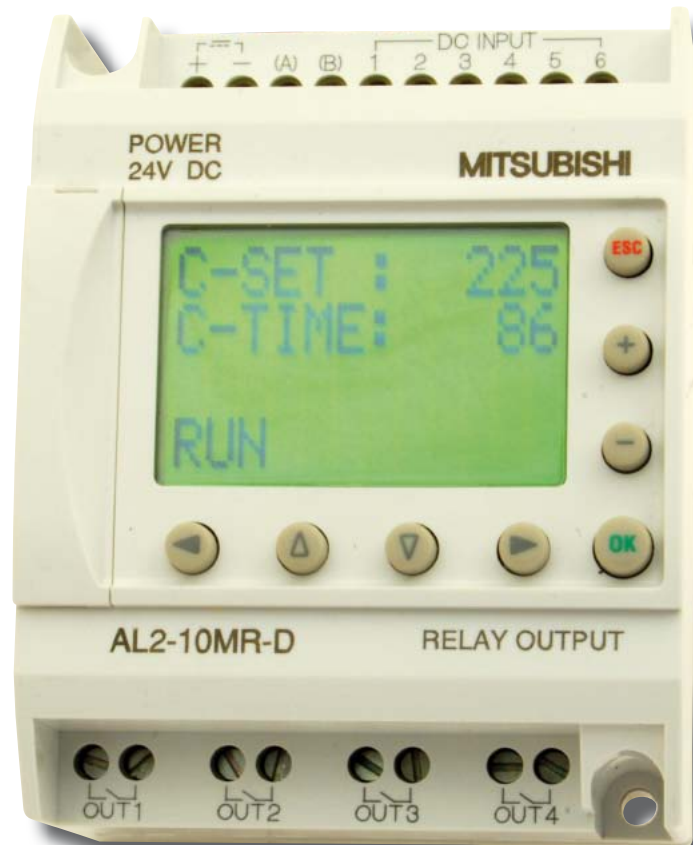
Sprzęt

Sterownik Alpha2 firmy Mitsubishi określany jest przez samego producenta jako *Simple Application Controller*. Wersja sterownika, na bazie której powstał niniejszy kurs programowania, ma 4 wyjścia przekaźnikowe, 6 wejść mogących pełnić funkcję analogowych lub cyfrowych. Jego wygląd przedstawiono na **fol. 1**.

Już na pierwszy rzut oka produkt sprawia bardzo dobre wrażenie: solidna, wykonana z ABS obudowa, zaciski śrubowe, wyświetlacz i prosta klawiatura umożliwiające poruszanie się po menu użytkownika. Pod zaślepką ukryte jest gniazdo służące do podłączenia adaptera (**fol. 2**), dzięki któremu można zaprogramować „Alfę”.

Sterownik produkowany jest w trzech wersjach, różniących się pomiędzy sobą liczbą wejść/wyjść cyfrowych oraz możliwością podłączenia dodatkowych modułów rozszerzeń i dostępnością portu szeregowego do zamontowania modemu. Jak wspomniano wcześniej, na potrzeby kursu wybraliśmy najprostszy i najtańszy sterownik AL2-10MR-D, jednak potencjalny użytkownik może również wybrać jedną z dwóch innych wersji: AL2-14MR (8 wejść/6 wyjść) lub AL2-24MR (15 wejść/9 wyjść). Każdy ze sterowników produkowany jest w wersji przystosowanej do zasilania napięciem 100...230 VAC (A) lub 24 VDC (D).

Aby uniknąć nieporozumień, omawiając sygnały wejścia/wyjścia, należy wspomnieć o jednej cesze sterowników Alpha 2. Otóż żaden ze sterowników zasilanych prądem przemiennym (z literą -A) nie ma wejść



Fot. 1. Wygląd sterownika Mitsubishi Alpha2 typu AL2-10MR-D

analogowych. Wersje zasilane prądem stałym (z literą -D) mają wejścia napięciowe 0...10 VDC o rozdzielczości 9 bitów (20 mV), z czasem konwersji A/C 8 ms. Wszystkie sterowniki mają wyjścia przekaźnikowe, a opiswany przez nas AL2-10MR-D może przełączać prąd o maksymalnym natężeniu 8 A, natomiast minimalna moc przełączana to 50 mW. Maksymalne przełączane napięcie to 250 VAC lub 30 VDC. Tylko moduł rozszerzeń o nazwie AL2-4ET, dołączany do wyższych modeli (AL2-14 i AL2-24), ma odizolowane galwanicznie wyjścia tranzystorowe, za pomocą których można sterować np. modułem falownika. Orientację w typach sterowników ułatwi **tab. 1**.

Jak wspomniano wcześniej, „Alfa” ma wyświetlacz LCD, którego można użyć do wyświetlania komunikatów przeznaczonych dla użytkownika. Mogą to być informacje o stanie bieżącym, o stanach alarmowych, wartości zadane i bieżące liczników czy timerów, wielkości analogowe. Ekran jest pod-

świetlany i ma 4 linie po 12 znaków. Jeśli dodamy do tego możliwości przewijania komunikatów, to LCD jest zupełnie wystarczający w prostych aplikacjach. Z ekranem LCD wiąże się ciekawa funkcjonalność, a mianowicie oprogramowanie sterownika może przesłać zawartość ekranu w postaci komunikatu SMS. I w tym miejscu jedna bardzo ważna uwaga: jeśli sterownik musi komunikować się z otoczeniem przez modem, to AL2-10MR nie nadaje się do tego celu, ponieważ nie ma portu służącego do podłączenia adaptera modemu.

Oprogramowanie AL-PCS/WIN-E

Do napisania programu dla sterownika służy IDE o bardzo ciekawych możliwościach. Jest to nie tylko rodzaj kompilatora, ale również kompletne środowisko umożliwiające tak symulację, jak i uruchomienie oraz monitoring programu w rzeczywistym środowisku jego pracy. Programista może obserwować pobudzenia, sygnały wewnętrzne,

Tab. 1. Zestawienie sterowników z serii Alpha2

Typ sterownika Funkcjonalność	AL2-10MR-A	AL2-10MR-D	AL2-14MR-A	AL2-14MR-D	AL2-24MR-A	AL2-24MR-D
Wejścia/Wyjścia	10	10	14	14	24	24
Wejścia dwustanowe	6	6	8	8	15	15
Wejścia analogowe	-	6	-	8	-	18
Wyjścia dwustanowe	4	4	6	6	9	9
Port adaptera modemu	-	-	Tak	Tak	Tak	Tak
Karty rozszerzeń	Tylko AL2-PT i AL2-TC	Tylko AL2-PT i AL2-TC	Tak	Tak	Tak	Tak
Zasilanie	100...230 VAC	24 VDC	100...230 VAC	24 VDC	100...230 VAC	24 VDC



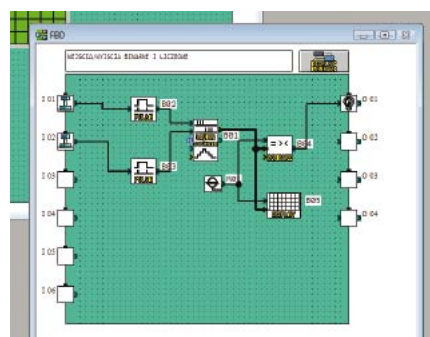
Fot. 2. Kabel AL-232CAB do programowania sterownika

pracę liczników oraz wszystkich układów wewnętrznych, jak również sygnały wyjściowe i wyświetlacz użytkownika. Niektórym użytkownikom, mającym pewne „stare nawyki programowania” (jak autor artykułu...), może przeszkadzać brak możliwości pisania programu w języku STL (albo innym tekstowym) lub programowania w postaci tzw. drabinek, ponieważ programy dla „Alfy” pisze się jedynie w postaci graficznej: zadaniem programisty jest odpowiednie ułożenie i połączenie ze sobą bloków funkcjonalnych. Tworzenie programu nieco przypomina zajęcie z podstaw techniki cyfrowej. Gdy utykałem na brak tych mechanizmów, to pewien kolega programista powiedział: „dinozaury też chciały być sobą i co?”. Faktycznie, świat dąży ku uproszczeniom i zapewne taki sposób programowania już wkrótce będzie obowiązujący. Ten trend widoczny jest zarówno w kompilatorach dla komputerów PC, jak i mikrokontrolerów.

Programista ma do dyspozycji szereg gotowych funkcji, dzięki czemu proste programy użytkowe mogą z powodzeniem być tworzone również przez osoby nieznające metod programowania. Oprócz tego, bardziej zaawansowani użytkownicy mogą tworzyć własne funkcje – dodawać swoje bloki do już istniejących. Dokumentacja producenta

podaje również, że funkcje można pisać np. w języku C (np. zaawansowane procedury obsługi komunikacji w sieci), jednak owa funkcjonalność, mimo iż bardzo ciekawa, przerasta chyba wymagania większości potencjalnych użytkowników.

Bloki mogą mieć wejścia i wyjścia sygnałów w postaci liczbowej (liczby *integer*, z zakresu -32767...+32768) lub bitowej. Wejścia umieszczane są po lewej stronie bloku, natomiast wyjścia po prawej. Dodatkowo symbolizują je strzałki skierowane, odpowiednio, „do” lub „od” bloku. Ani bloczków, ani funkcyjnych nie można obracać, więc opisana wyżej reguła jest zawsze prawdziwa. W symbolach



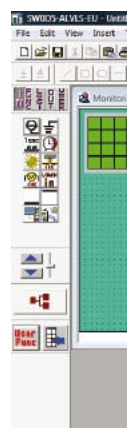
Rys. 3. Różnice w grubościach linii połączeń bitowych (cienkie) i liczbowych (grube)

wejść i wyjść bitowych znak „>” umieszczony jest na tle szaroniebieskim, natomiast w liczbowych – na zielonym. Różnicę widać również po połączeniu bloków: połączenia wiodące bity prowadzone są cienką linią, natomiast liczby – grubą (rys. 3). Czasami po lewej stronie bloku umieszczony jest na niebieskim tle napis „CL”. W ten sposób oznaczane jest wejście zerowania, np. licznika.

W związku z tym, że na potrzeby tego artykułu przyjęto zasadę nauki przez przykłady programów użytkowych, w dalszej części kursu zajmiemy się opisami poszczególnych bloków i ich cech użytkowych. Wyżej podano podstawowe informacje, które pozwalają tylko nieco orientować się w symbolice wprowadzonej przez producenta.

Po lewej stronie ekranu umieszczony jest pasek (rys. 4) narzędziowy umożliwiający narysowanie schematu. Wybranie przycisku w górnej części paska napisu: IN, FUNC, LOGIC, OUT, USER, powoduje wyświetlenie odpowiednio: symboli wejść, bloków funkcyjnych, funkcyjnych, symboli wyjść, funkcji użytkownika. Na ekranie wyświetlana jest tylko niewielka część listy dostępnych funkcji czy opcji, jej przesuwanie umożliwiają umieszczone poniżej symbole strzałek i suwak. Bezpośrednio pod nimi umieszczono przycisk (trzy brązowe, połączone ze sobą kwadraty), po którego naciśnięciu IDE przechodzi do trybu rysowania połączeń. Bloki można łączyć ze sobą na dwa sposoby: klikając na wejściu/wyjściu źródłowym i docelowym oraz techniką *drag & drop*. W przypadku niezgodności typu wejścia z wyjściem na ekranie natychmiast zostanie wyświetlony odpowiedni komunikat.

Osobom przyzwyczajonym do programów – edytorów schematów urządzeń elektronicznych mogą przeszkadzać skromne funkcje edycji połączeń oraz daleko posunięta automatyzacja. Czasami po prostu nie można odpowiednio czytelnie i estetycznie poprowadzić linii. Owszem, nie ma to żadnego znaczenia dla kompilatora



Rys. 4. Pasek narzędziowy z zasobnikiem funkcji



Rys. 5. Wybór modelu sterownika

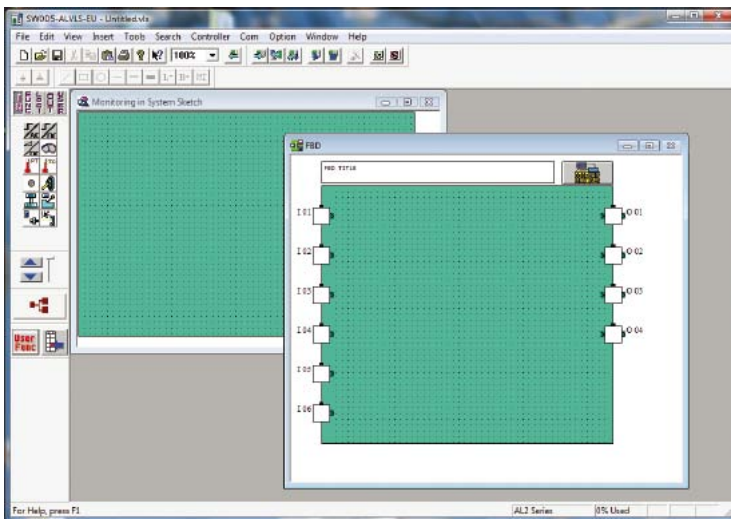
i pewnie w żaden sposób nie wpływa na optymalizację kodu wynikowego, ale wygląd i czytelność schematu są ważne dla człowieka – programisty, który musi orientować się w tym gąszczu połączeń.

Do przesłania skompilowanego programu „Alfy” służy specjalny adapter – przejściówka pomiędzy portem szeregowym RS232 komputera PC (fot. 2) a sterownikiem. Kabel nosi oznaczenie AL-232CAB i można go nabyć jako akcesorium dodatkowe.

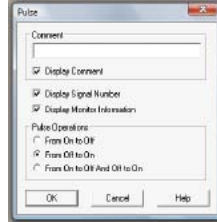
Mimo że ten adapter tak jakby dodaje do sterownika łącze szeregowo, to nie nadaje się on np. do podłączenia modemu. Do tego celu służy inny adapter AL2-GSM-CAB, za pomocą którego sterownik może również przysłać komunikaty do innych systemów, wyposażonych w interfejs RS232. Kabel AL-232CAB w połączeniu z komputerem PC ma bardzo szeroką funkcjonalność, jednak czasami, np. do skopiowania programu do innego sterownika, nie jest ona potrzebna. Wówczas można posłużyć się kasetą EEPROM, dzięki której nie jest konieczne taszczenie ze sobą komputera, kabli połączonych itp.

Program AL-PCS (podczas pisania artykułu była to wersja 2.60) pracuje pod kontrolą środowiska Windows, w tym również Windows Vista. Nie sprawia mu też żadnego problemu współpraca z adapterami USB/RS232. Numer portu COM, do którego podłączony jest adapter sterownika, ustawia się w konfiguracji (menu *Com* -> *Configuration*).

Program przesyłany jest do sterownika po wybraniu opcji *Controller* -> *Write to Controller*. Oprócz tego, menu *Controller*



Rys. 6. Obszar roboczy



Rys. 7. Okno właściwości generatora impulsu PULSE

umożliwia weryfikację i odczyt programu, uruchomienie i zatrzymanie jego realizacji, odczyt zajętości pamięci, diagnostykę „Alfy”, oraz symulację realizacji programu na komputerze PC i jego monitorowanie w rzeczywistym środowisku pracy.

Przykład 1: Opóźnione wyłączenie

W pierwszym z przykładów programowania sterownika Alpha 2 pokażemy, w jaki sposób zbudować wyłącznik czasowy o ustawianym opóźnieniu wyłączenia. Wyłącznik będzie reagował na opadające zbocze sygnału na wejściu 1 (zwarcie do masy), załączając przełącznik wyjściowy 1 (OUT1) na ustawiony czas opóźnienia. Niech czas opóźnienia będzie ustawiany klawiszami „+” i „-” w zakresie 1...300 sekund, a przedłużający się stan niski na wejściu 1 będzie ignorowany. Nastawy niech będą możliwe po naciśnięciu „ESC”, a powrót do trybu pracy i wyjście z trybu nastaw, po naciśnięciu „OK”. Odpowiednie stany pracy niech będą sygnalizowane na wyświetlaczu.

Na płycie CD-EP8/2009B w pliku o nazwie CZASÓWKA.VLS zapisano program funkcjonujący zgodnie z opisanymi wyżej założeniami. Spróbujmy pokrótce przeanalizować jego konstrukcję.

Pobudzenie

Tworzenie programu rozpoczyna się od wybrania z menu *File* opcji *New*. Na ekranie komputera pokaże się okienko (rys. 5), w którym należy dokonać wyboru modelu sterownika. Następnie ukaże się ekran jak na rys. 6. Po rozciągnięciu obszaru roboczego (FBD)

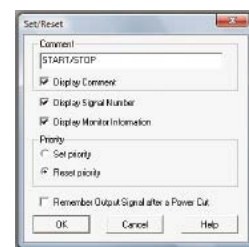
można przystąpić do układania poszczególnych komponentów.

Na wejściu I01 ułożono symbol przycisku, który będzie imitował pobudzenie podczas symulacji. Sygnał ten podano na wejście bramki AND (B17) blokującej wyzwalenie w trybie nastaw. Z wyjścia bramki pobudzenie dociera do bloku PULSE (B12), którego zadaniem jest generowanie impulsu wyzwalającego w odpowiedzi na zbocze sygnału. Dwukrotne kliknięcie na symbolu bloku otwiera okno jego właściwości (rys. 7), umożliwiające wybór zbrocza, na którym generowany będzie sygnał wyjściowy: *From On to Off* – opadające, *From Off to On* – narastające, *From On to Off And Off to On* – oba. Opcje te odnoszą się do logiki wejścia i zależą od konfiguracji wejść. Na potrzeby tego przykładu, posługującego się masą jako sygnałem aktywnym, należy wybrać opcję drugą, czyli *From Off to On* (masa jest aktywna, a więc jej brak to *Off*, a dołączenie to *On*).

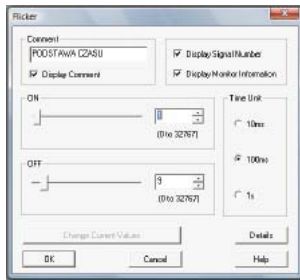
Sygnał z bloku PULSE podawany jest na wejście SET przerzutnika RS i powoduje ustawienie na jego wyjściu (B05) stanu wysokiego, który steruje pracą generatora podstawy czasu, bloku o nazwie FLICKER. Wróćmy jeszcze na chwilę do przerzutnika RS. Dwukrotne kliknięcie na jego symbolu otwiera okno właściwości (rys. 8), dzięki któremu możemy ustalić, w jakim stanie ma być ustawiony przerzutnik po załączeniu (*Set priority* – stan wysoki, *Reset priority* – stan niski) oraz czy stan przerzutnika ma być zapamiętany w przypadku awarii napięcia zasilania (*Remember Output Signal after a Power Cut*). W tym przykładzie, po załączeniu zasilania, wyjście przerzutnika ma być wyzerowane (*Reset priority*) oraz nie ma potrzeby zapamiętywania stanu w przypadku wyłączenia zasilania.

Sygnał B05 bramkuje generator FLICKER. Generuje on przebieg prostokątny, jeśli na jego wejściu binarnym panuje stan wysoki. Parametry generowanego przebiegu można ustawić po dwukrotnym kliknięciu na symbolu (rys. 9). Suwakami wybiera się czas wyłączenia i załączenia sygnału. W przykładzie ustawiono odpowiednio 100 ms i 900 ms – generowany jest przebieg o częstotliwości 1 Hz.

Sygnał ten (B06) steruje pracą licznika COUNTER, którego stan (B04) porównywany jest z nastawą (B02) przez moduł COMPARE. Moduł COUNTER zlicza do 300 (rys. 10), a następnie jest zerowany.



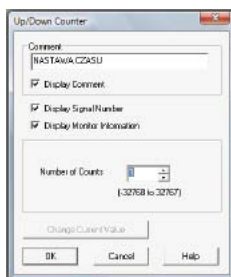
Rys. 8. Okno właściwości przerzutnika



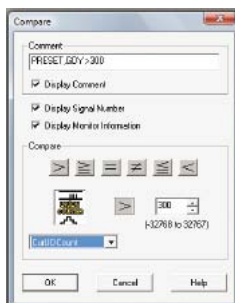
Rys. 9. Okno właściwości bloku PULSE



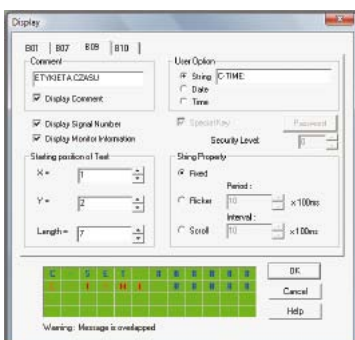
Rys. 10. Okno właściwości generatora FLICKER



Rys. 11. Okno właściwości licznika UP/DN COUNTER



Rys. 12. Okno właściwości komparatora COMPARE



Rys. 13. Okno właściwości bloku DISPLAY: wyświetlanie stałych

Nastawa czasu

Nastawa czasu wprowadzana jest przyciskami „+” (zwiększanie) i „-” (zmniejszanie). Naturalne wydaje się zastosowanie do wprowadzenia nastawy licznika mającego

możliwość zliczania dwukierunkowego i podłączenie przycisków do odpowiednich wejść powodujących inkrementację lub dekrementację zawartości. W programie wyłącznika czasowego posłużono się blokiem UP/DN COUNTER, do którego wejścia UP podłączono przycisk „+” (K03), a do wejścia DN przycisk „-” (K04). Sygnał z przycisków bramkowany jest przez dwie bramki AND (odpowiednio B16 i B18) i dopiero te sygnały podawane są na wejścia licznika. Bramkowanie zastosowano po to, aby nie było możliwe wprowadzanie nastaw w trybie odmierzania czasu. W okienku właściwości licznika (rys. 11) wprowadzono nastawę minimalną, tj. liczbę 1, od której rozpoczyna się nastawę czasu. Liczba 1 wpisywana jest również do rejestru licznika po sygnale preset.

Blok UP/DN COUNTER ma 5 wejść. Są to (od góry): bitowe wejście inkrementacji, bitowe wejście dekrementacji, wejście zerowania (CL), bitowe wejście sygnału preset oraz liczbowe wejście nastawy w postaci liczby typu integer (-32767...+32768). W programie wejście zerowania nie jest używane. Sygnał (B14) na wejściu preset pojawia się w następujących sytuacjach:

- Gdy wartość nastawy przekroczy 300. Odpowiada za to blok COMPARE porównujący stan licznika (B02) ze stałą liczbą 300, którą wpisano w oknie właściwości (rys. 12). Korzystając z niego, należy ustawić sygnał, który jest porównywany (lista rozwijana w dolnej części ekranu) oraz warunek porównania (przyciski z symbolami matematycznymi). Przy poprawnym ustawieniu, na wyjściu bloku COMPARE panuje stan niski, jeśli warunek $B02:CurUDCount > 300$ nie jest spełniony.
- Gdy zawartość licznika równa się 0, co może zdarzyć się przy pierwszym załączeniu zasilania sterownika lub podczas zmniejszania nastawy. Za wygenerowanie sygnału odpowiedzialny jest blok COMPARE (B27).

Wszystkie wymienione wyżej sygnały (B14 i B27) sumowane są przez bramkę OR (B13) i podawane na wejście preset licznika UP/DN COUNTER.

Wymuszenie stanu licznika można wykonać również w inny sposób. Wyjście binarne bloku UP/DN COUNTER ma stan wysoki, gdy wartość zliczana jest większa od wpisanej Number of counts. Jak łatwo się domyślić, to wyjście pełni podobną rolę, jak zastosowany blok COMPARE. Problem pojawia się jednak w momencie załączenia zasilania. Pierwotnie zastosowałem w programie blok PULSE, do wejścia którego podłączyłem sygnał ALWAYS ON, a wyjście podałem na bramkę OR (B13). Okazało się jednak, że w ten sposób wartość licznika nastaw nie była przechowywana przy zaniku napięcia zasilania, tylko powracała do nastawy minimalnej. Stąd

konieczność użycia bloku COMPARE (B27), którego zadaniem jest generowanie sygnału preset przy niedozwolonym stanie licznika.

Stan licznika B04 zliczającego impulsy podstawy czasu (FLICKER, B06) i licznika zawierającego nastawy (UP/DN COUNTER (B02) porównywany jest przez blok COMPARE (B03), generujący na swoim wyjściu stan wysoki, gdy spełniony jest warunek $B04:CurCount < B02:CurUDCount$, tzn. stan licznika impulsów podstawy czasu jest mniejszy od nastawy. Jego wyjście (B03) bezpośrednio steruje pracą przełącznika wyjściowego O01.

Opadające zbocze sygnału wyjściowego (B03) podane jest na wejście bloku PULSE generującego sygnał (B08) zerujący licznik podstawy czasu i przerzutnik RS bramkujący generator podstawy czasu.

Interfejs użytkownika

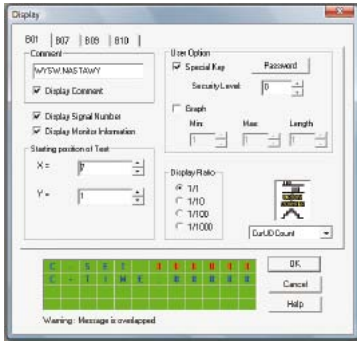
Jak to zwykle bywa przy pisaniu programów, funkcje interfejsu użytkownika zajmują całkiem sporo zasobów i komplikują program, jednak komunikaty prezentowane użytkownikowi są niezbędne po to, aby ten wiedział, że sterownik pracuje poprawnie.

Blok o nazwie DISPLAY odpowiada za wyświetlanie stałych napisów i zmiennych, tj. liczb całkowitych typu integer. Komunikat pojawia się na ekranie wtedy, gdy wejście binarne bloku jest w stanie wysokim.

Wejścia binarne bloków B09 i B10 podłączono na stałe do sygnału ALWAYS ON (M01). Bloki te powodują wyświetlanie komunikatu, odpowiednio, C-TIME (etykieta czasu bieżącego) i C-SET (etykieta czasu ustawionego). Przyjrzyjmy się bliżej właściwościom bloku DISPLAY.

Okienko to wygląda różnie w zależności od tego, czy wejście zmiennych liczbowych jest podłączone, czy też nie. Jeśli nie, to na wyświetlaczu można pokazać: dowolny napis (np. jak na rys. 13), datę lub godzinę. Wyboru dokonuje się przez zaznaczenie odpowiedniego pola w prawym, górnym rogu. Jeśli zostanie wybrane wyświetlanie stałej typu string, to na ekranie pojawi się właściwe pole umożliwiające wpisanie tekstu. Format daty i czasu określany jest w zmiennych globalnych i dlatego nie ma możliwości jego wyboru. Po lewej stronie ekranu, w środkowej części okna, programista ma możliwość wyboru współrzędnych X, Y, na których pojawi się napis. Dodatkowo można określić długość łańcucha – jeśli ta będzie większa od rzeczywistych potrzeb, to łańcuch zostanie dopełniony spacjami, co można zastosować do wymazywania wcześniejszych komunikatów. Następnie, zaznaczając odpowiednie pole wyboru po prawej stronie, można wybrać, czy komunikat ma być stały (Fixed), czy ma migotać i z jaką częstotliwością (Flicker), czy też ma być przewijany w poziomie w ramach swojej długości.

Jeśli do wejścia modułu DISPLAY doprowadzono zmienną, to okienko przyjmuje wy-



Rys. 14. Okno właściwości bloku DISPLAY: wyświetlanie zmiennych

gląd taki, jak na rys. 14. Długość wyświetlanej liczby jest stała (znak + 5-cyfrowa liczba lub 5 cyfr bez znaku), więc z okienka znika możliwość wyboru długości napisu, a zostaje tylko możliwość określenia współrzędnych X i Y. Liczby wyświetlane są od prawej do lewej strony w taki sposób, że najmniej znacząca cyfra umieszczana jest po prawej stronie i od niej, w kierunku w lewo, umieszczone są cyfry bardziej znaczące.

Po prawej stronie, w miejscu, gdzie poprzednio wyświetlane było pole wyboru *Fixed/Time/Date*, teraz umieszczone jest pole wyboru o nazwie *Graph*, a poniżej niego lista sygnałów dostępnych z danego źródła oraz pola wyboru *Display Ratio* umożliwiające wstawienie przecinka (np. przy *Display Ratio=1/10* liczba 3450 wyświetlana jest jako 345.0). Polem *Graph* zajmiemy się w kolejnych przykładach, kiedy potrzebny będzie sposób obrazowania liczb w postaci słupka graficznego.

Obraz projektowanego wyglądu wyświetlacza umieszczany jest na samym dole. Pojedyncze cyfry reprezentowane są przez znak #, natomiast etykiety wyświetlane są tak, jak wpisano w polu *string*. Na górze okienka umieszczone są zakładki, których wybranie umożliwia projektowanie wyglądu wyświetlanego komunikatu w jednym miejscu, bez konieczności „skakania” po symbolach na schemacie. Zakładki obejmują bloki *DISPLAY* podłączone do danego sygnału wy-



Fot. 16. Usunięcie zaślepki gniazda

boru załączenia wyświetlacza (np. *ALWAYS ON*, jak w tym przykładzie). Wygląd projektowanego komunikatu umieszczony jest na samym dole okna – aktualnie wybrana zakładka wyróżniana jest kolorem czerwonym. Wróćmy jednak do interfejsu użytkownika.

Bloki odpowiedzialne za wyświetlanie etykiet i czasów podłączone są do sygnału *ALWAYS ON* (M01) i wyświetlane zawsze. Zmienny jest komunikat na samym dole ekranu LCD, który sygnalizuje tryb pracy (B21, komunikat *RUN*) i tryb nastaw (B20, komunikat *SET*). Przelączaniem wyświetlania komunikatów steruje sygnał B15, pobierany z wyjścia przerzutnika RS (stan wysoki odpowiada trybowi *SET*, a stan niski *RUN*). Zmiany stanu tego przerzutnika dokonuje się, naciskając klawisz *ESC* lub *OK*. Odpowiada to wejściu w tryb nastaw lub tryb pracy. Funkcja klawisza *ESC* jest blokowana (B24 po zanegowaniu B08), gdy została wyzwolona funkcja odmierzenia czasu. Dodatkowo sygnał B15 uaktywnia klawisze „+” i „-” oraz blokuje wejście I01, a tym samym możliwość wyzwolenia odmierzenia czasu.



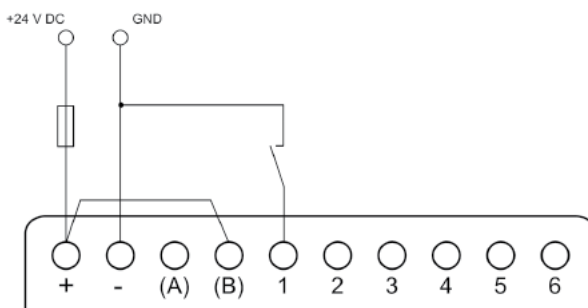
Fot. 17. Podłączenie kabla do programowania sterownika

Nie używane klawisze podłączone na wejście bramki *OR*. Jej wyjście (B25) steruje wyświetlaniem migoczącej gwiazdki w prawym, dolnym rogu wyświetlacza (B026), która sygnalizuje wciśnięcie, ale również nieaktywność klawisza. Gdyby nie używane klawisze pozostawić niepodłączone, to wciśnięcie dowolnego z nich powoduje migotanie wyświetlacza (funkcja związana z systemowym menu użytkownika), co mogłoby zdezorientować użytkownika.

Podłączenie i uruchomienie

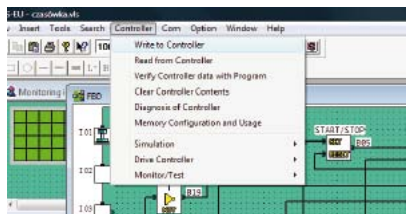
Sterownik należy podłączyć zgodnie ze schematem umieszczonym na rys. 15. Przy wyłączonym zasilaniu sterownika do komputera PC należy podłączyć kabel RS232 przeznaczony do przesłania programu, a następnie podłączyć go do gniazda sterownika zasłoniętego zaślepką. Tę należy delikatnie usunąć, odginając zaczep za pomocą cienkiego wkrętaka (fot. 16). Po podłączeniu przewodu (fot. 17) można włączyć zasilanie sterownika. Jest to postępowanie zalecane przez producenta.

Teraz, mając otwarty program *AL-PCS/WIN-E*, a w nim projekt *CZASÓWKA.VLS*, należy wybrać z menu *Controller* opcję *Write to Controller* (rys. 18). Program można przesłać wyłącznie do sterownika, który nie pracuje i jest w trybie oczekiwania na uruchomienie (menu *Controller* -> *Drive Controller* -> *Stop*). W okienku (rys. 19) można zaznaczyć obie opcje, co wprowadzi *IDE* w tryb monitorowania i uruchomi program na sterowniku.

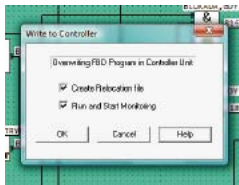


Rys. 15. Schemat połączeń sterownika

R E K L A M A

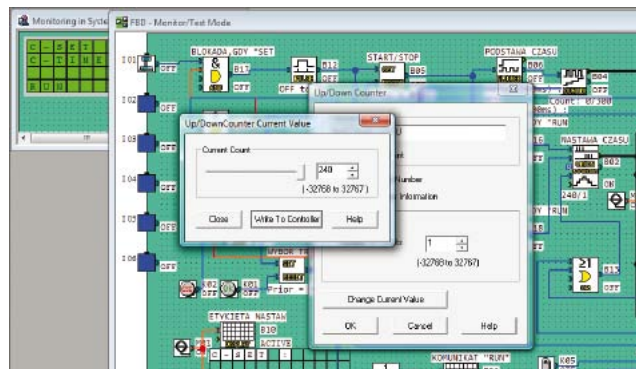


Rys. 18. Menu zapisu programu do sterownika



Rys. 19. Wybór opcji zapisu programu

Monitoring aplikacji pokazuje stany sygnałów wewnątrz sterownika, pracę liczników i komunikaty wyświetlane na LCD. Umożliwia również wymuszanie pobudzeń pochodzących od sygnałów zewnętrznych (w tej aplikacji będzie to możliwość podania sygnału wyzwalającego odmierzenie czasu) oraz zmiany stanów liczników. Przykład zmiany nastawy licznika UP/DN COUNTER przechowującego nastawę czasu pokazano na rys. 20. Aby otworzyć okienko umożliwiające modyfikację, należy dwukrotnie kliknąć na symbolu licznika, a następnie na



Rys. 20. Zmiana stanu licznika nastaw

przycisk *Change Current Value*. Po wpisaniu nowej zawartości, trzeba kliknąć na *Write to Controller*. Monitoring można przerwać, wybierając z menu opcję *Controller -> Monitor/Test -> Stop* lub wskazując klawisz z literą „M” na pasku skrótów.

Inną możliwością uruchomienia programu jest jego symulacja. Wystarczy wybrać z menu *Controller -> Simulation -> Start*. Program uruchamiany jest na komputerze PC i zachowuje się w taki sposób, jakby był uruchomiony na sterowniku w trybie monitorowania. Oczywiście w takiej sytuacji nie sposób jest symulować rzeczywiste przebiegi wejściowe, ale tryb ten doskonale nadaje się do uruchomienia i weryfikacji funkcjonowania prostych funkcji logicznych oraz sterowania.

Podsumowanie

W kolejnych odcinkach kursu programowania sterownika Alpha2 pokażemy, w jaki sposób korzystać z innych, nieopisanych wyżej, funkcji sterownika. Opisanymi bloków będziemy zajmować się przy okazji ich stosowania, ucząc przez przykłady i odbiegając tym samym od zwykle przyjętej formy kursu, w którym po kolei wymienia się wszystkie funkcje, aby następnie przejść do przykładów. Programowanie „Alfy” jest tak łatwe, że możemy sobie pozwolić na drobne odstępstwo od norm, prezentując Czytelnikom użyteczne przykłady programów.

Jacek Bogusz, EP
jacek.bogusz@ep.com.pl

R E K L A M A

www.sklep.avt.pl

WYKRYWACZ METALI Z WYŚWIETLACZEM LCD CS300 – 590 zł

Profesjonalny wykrywacz metali z wodoodporną cewką, identyfikacją rodzaju metalu, wskaźnikiem głębokości

RĘCZNY DETEKTOR METALI CS10MD – 180 zł

Wysokiej jakości przenośny wykrywacz metali. Doskonały dla służb porządkowych chroniących imprezy masowe, na lotniskach i w miejscach szczególnie narażonych na ataki terrorystyczne. Urządzenie pozwala bezdotykowo skontrolować czy badana osoba posiada metalowe przedmioty np: broń lub nóż

AMATORSKI WYKRYWACZ METALI CS100 – 210 zł

Wykrywacz metali średniej klasy, doskonały do poszukiwania monet, kosztowności, złota, srebra i innych 'skarbów'

WYKRYWACZ METALI Z DISKRYMINATOREM CS150 – 320 zł

Maksymalna głębokość wykrywania wynosi około 1m (duże przedmioty). Posiada funkcję dyskryminacji rozróżniania rodzaju metali znajdujących się pod ziemią