

# Programowanie Alpha2 (3)

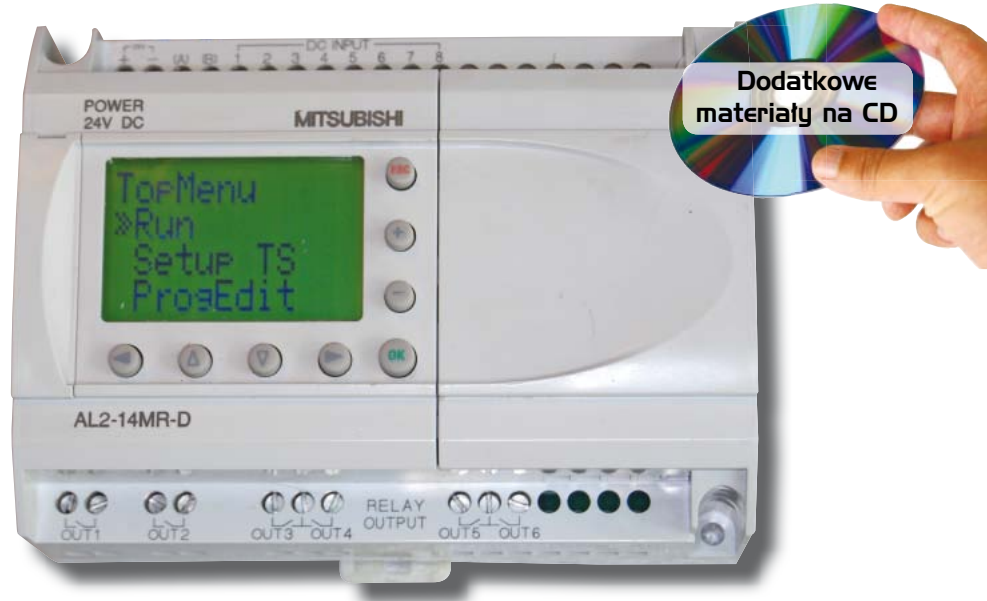
## Moduł wyjść analogowych

*Kontynuujemy kurs programowania sterownika Alpha2. Poprzednio opisywaliśmy sposób użycia wejść analogowych, a teraz zajmiemy się modułem wyjść analogowych.*

*Do tego celu musimy użyć „większej” Alfy AL2-14MR-D, ponieważ jest ona wyposażona przez producenta w gniazdo dla dodatkowych modułów rozszerzeń.*

Mimo swoich wad, sygnał analogowy stale używany jest w automatyce. Powód jest oczywisty – prostota wykonania interfejsu łączącego czasami całkowicie różne urządzenia. Wystarczy określić zakresy wartości napięć lub prądów i już można łączyć ze sobą układy. W razie jakichś problemów, bardzo łatwo jest też skalować lub filtrować taki sygnał. Techniki są bardzo dobrze znane, praktycznie od zarania dziejów elektrotechniki, a urządzenia automatyki sterowane sygnałem analogowym można mnożyć.

Sterownik Alfa w swojej podstawowej konfiguracji nie ma wyjść analogowych a jedynie przekaźnikowe. Trzeba do niego podłączyć dodatkowy moduł, a ten wymaga gniazda rozszerzenia mającego dostęp do szyny mikroprocesora, tak jak karty dodawane do komputera PC. Poprzez to gniazdo moduł bezpośrednio komunikuje się z CPU. Informacja ta jest o tyle istotna, że ograni-



Fot. 1. Sterownik Alfa AL2-14MR-D

cza zastosowanie modułu do tych wersji sterownika Alfa, które mają wzmiankowane gniazdo rozszerzenia. Należą do nich AL2-24MR oraz AL2-14MR. Najmniejszy, używany przez nas w poprzednich odcinkach kursy, AL2-10MR nie umożliwia podłączenia żadnego dodatkowego modułu za pośrednictwem gniazda. Trzeba o tym pamiętać wybierając sterownik, który np. ma komunikować się z otoczeniem za pośrednictwem RS232. Mimo, iż wszystkie Alfy programowane są z użyciem tego interfejsu, to nie nadaje się on do wykorzystania we własnej aplikacji. Do połączenia np. z modemem GSM trzeba wpiąć w gniazdo Alfy moduł, który instaluje dodatkowy port COM.

### Mitsubishi AL2-14MR-D

Do napisania tego odcinka kursu posłużyła nam Alfa AL2-14MR-D (fot. 1). Jest ona wyposażona w 8 wejść uniwersalnych oraz 6 wyjść przekaźnikowych. Dwa z nich są zupełnie niezależne, natomiast cztery mają wspólny zestyk. W związku z tym, budując aplikację trzeba ją zaprojektować w taki sposób, aby wspólny zestyk był np. dodatnim lub ujemnym, wspólnym biegunem zasilania.

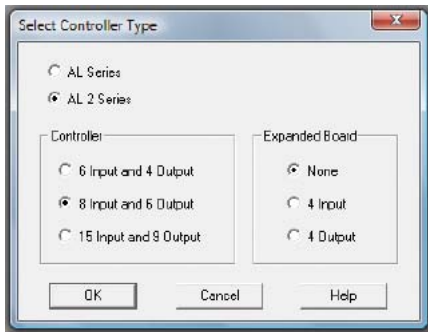
Alfa 14 jest szersza od używanej przez nas poprzednio Alfy 10. Ma więcej wejść i wyjść, natomiast pozostałe elementy nie ulegają zmianie. Identycznie podłącza się zasilanie oraz konfiguruje wejścia. Programy



Fot. 2. Sposób umieszczenia gniazda modułów rozszerzeń



Fot. 3. Sposób podłączenia dodatkowego modułu wyjść analogowych



Rys. 4. Okno wyboru sterownika i modułu rozszerzenia

napisane dla Alfę 10 funkcjonują na Alfie 14. Oprócz drobnych zmian kosmetycznych nie trzeba wykonywać żadnych poważniejszych przeróbek. Sterownik doskonale sprawdzi się więc w sytuacji, w której przez rozbudowę aplikacji potrzebna będzie większa liczba wejść lub/i wyjść. Śmiało można zaryzykować twierdzenie, że raz wykonane oprogramowanie bez żadnych przeszkód jest kompatybilne w górę i pod pewnymi warunkami w dół. To bardzo komfortowa sytuacja dla programisty.

Gniazdo do programowania umieszczono (tak samo jak w Alfie 10) pod klapką po lewej stronie. Gniazdo modułów rozszerzeń schowane jest po prawej stronie i dostępne po odkręceniu śruby blokującej oraz odsunięciu pokrywy (fot. 2). Moduł rozszerzenia wsuwa się na miejsce klapki osłony (fot. 3) i na wszelki wypadek blokuje śrubą, chociaż jego zatrzaski są mocne i pewne. Śruba jest integralną częścią tak modułu, jak i pokrywy. Nie ma więc obaw, że gdzieś się zapodzieje.

## Moduł wyjść analogowych AL2-2DA

Moduł wyjść analogowych ma oznaczenie AL2-2DA. Ciekawostką jest fakt, że jest to moduł uniwersalny, to znaczy można korzystać bądź z wyjścia napięciowego lub prądowego. W związku z tym, że moduł ma dwa wyjścia, to można np. jedno z nich wykorzystywać jako prądowe, a drugie jako napięciowe. Nic nie stoi na przeszkodzie, aby oba były prądowymi lub napięciowymi. Wszystko zależy od inwencji projektanta.

Zakres napięć wyjściowych rozciąga się w granicach 0...10 V, natomiast prądów 4...20 mA. Rozdzielczość oferowana przez moduł to odpowiednio 2,5 mV (4 tys. kroków) lub 8  $\mu$ A (2 tys. kroków). Czas konwersji jest nie gorszy niż 20 ms dla obu kanałów (10 ms dla pojedynczego kanału).

Moduł jest odizolowany od sterownika z użyciem transoptorów. Zasilanie (24 VDC) modułu podłączane jest z zewnątrz, co pozwala na zastosowanie dla niego odrębnego źródła zasilania. Umożliwia to całkowite, galwaniczne odseparowanie modułu od sterownika. Jest to fakt nie bez znaczenia dla bezpieczeństwa użytkownika i aplikacji.

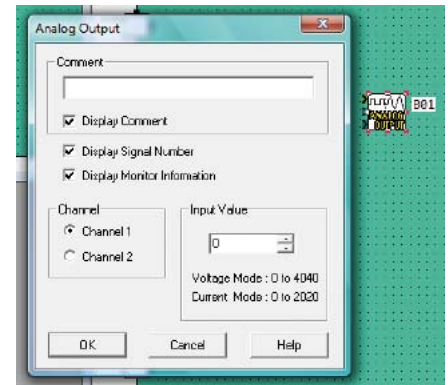
## Programowanie

Po podłączeniu modułu wyjść analogowych do Alfę, można napisać program obsługi. Za pierwszym razem, jak każdy typowy programista, który nie ma czasu na czytanie obszernej dokumentacji (sic!!!), już na samym początku pisania programu popełniłem błąd. Po wybraniu z menu *File*  $\rightarrow$  *New* pojawia się na ekranie okienko z pytaniem o typ sterownika oraz dołączone moduły rozszerzeń. Rzut oka wystarczył, aby poczuć się lekko zdezorientowanym. Po prawej stronie okienka (rys. 4) można dokonać wyboru dołączonej płytki rozszerzenia – pole *Expanded Board*. Do wyboru są trzy możliwości: *None*, *4 Input*, *4 Output*. No dobrze, a gdzie moduł wyjść analogowych? Na chybił trafił zaznaczyłem *4 Output*, zgodnie z zasadą „jakoś to będzie”. Program otworzył okno FBD, w którym na symbolu sterownika dodatkowo umieścił wyjścia oznaczone E 01...02. Powstał kolejny problem – rozumiałem, że wyjścia przekaźnikowe (O 01...06) nie mogły pełnić funkcji wyjść analogowych, ale w jaki sposób spowodować aby te dodatkowe, oznaczone literą „E” wyjścia, chciały przyjąć funkcję analogowych? Jaką ikonę z grupy *Output* ułożyć na symbolu wyjścia? Bo jak pamiętamy z poprzedniego odcinka kursu, to ikonki w pewien sposób determinują rolę wejścia, czyli – jak myślałem – zapewne też i wyjścia. Jak przypisać mu wartość liczbową (grube linie na schemacie), która zostanie zamieniona na analogową? Niestety, IDE w żaden sposób nie umożliwiło mi takiej operacji. Różne podejmowane próby skończyły się klęską. Otrzymałszy tego przysłowiowego „prztyczka w nos” wreszcie sięgnąłem do dokumentacji.

Po pierwsze, okazało się, że po podłączeniu rozszerzenia nie trzeba wybierać w menu żadnych dodatkowych modułów. Wśród bloczków w grupie *Func* można znaleźć blok o nazwie *Analog Output*, a umieszczenie go na schemacie jest równoważne z podłączeniem wyjścia analogowego. Wystarczy wybrać odpowiedni kanał, podłączyć sterowanie i po problemie. Dlaczego firma przyjęła tę koncepcję, to wyniknie z dalszej lektury.

Funkcja *Analog Output* używana jest do konwersji słowa (stałej lub zmiennej) uzyskiwanego jako wynik działania licznika lub wejścia analogowego (sterownika można np. użyć do skalowania analogowej wielkości wejściowej lub konwersji napięcia na adekwatną wartość prądu). Bloczek ma trzy wejścia i tak, odpowiednio, wskazują od góry, są to:

- Wejście słowa dwubajtowego, bezpośrednio sterujące wielkością napięcia lub prądu wyjściowego. Liczba 0...4000 zamieniana jest na napięcie 0...10 V, natomiast 0...2000 na prąd 4...20 mA.
- Wejście sterujące wyborem trybu pracy. Jeśli to doprowadzenie nie jest podłączo-



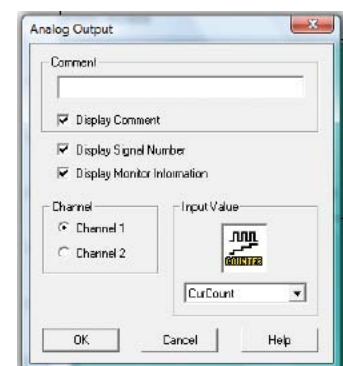
Rys. 5. Okno właściwości bloku *Analog Output*

ne lub zostanie podłączone do sygnału *Always Off*, lub logicznego „0”, to wyjście pracuje jako napięciowe, natomiast podłączenie sygnału *Always On* lub logicznego „1”, wymusza tryb prądowy.

- Wejście sterujące zezwoleniem na wprowadzenie sygnału wejściowego. Jeśli to wejście jest niepodłączone, podłączone do sygnału *Always On* lub do logicznego „1”, to wyjście na bieżąco reaguje na zmiany sygnału wejściowego. Podanie na wejście sygnału *Always Off* lub logicznego „0” zabrania zmianę sygnału wyjściowego.

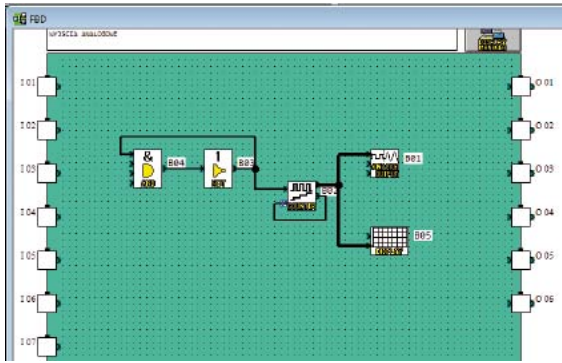
Oprogramowanie Alfę pozwala na stosowanie wielu bloków funkcyjnych *Analog Output* pracujących we wspólnym kanale. Dlatego też firma Mitsubishi przyjęła koncepcję bloku, który można włączyć lub wyłączyć, a nie wyjścia, do którego mogłaby być podłączona tylko pojedyncza wartość liczbowa zamieniana na adekwatne napięcie/prąd. Trzeba tylko pamiętać o tym, że w danym momencie, w obrębie danego kanału, wyłącznie jeden z nich może mieć zezwolenie na pracę oraz że wszystkie w obrębie tegoż kanału muszą pracować w tym samym trybie, to jest napięciowym lub prądowym.

Okno właściwości bloku *Analog Output* pokazano na rys. 5. Jego konstrukcja jest przejrzysta, podobna do okien właściwości innych bloków. W polu *Channel* dokonuje się wyboru numeru kanału, na którym wykonywane będą operacje. W polu *Input Value*,

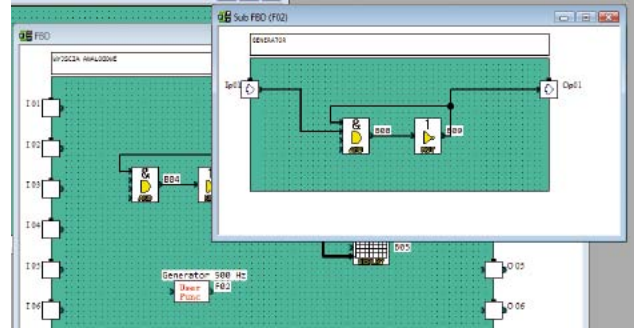


Rys. 6. Podłączenie wyjścia licznika do wejścia bloku *Analog Output*

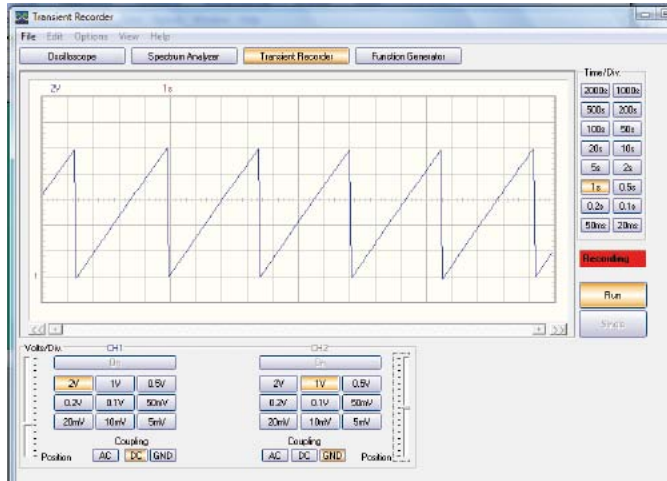




Rys. 7. Prosta aplikacja, której użyto do testów



Rys. 11. Okno podprogramu użytkownika



Rys. 8. Oscylogram przebiegu wyjściowego

jeśli wejście słowa nie jest podłączone, można wpisać stałą odpowiadającą pożądanemu napięciu. Jeśli wejście jest podłączone, to zamiast stałej można wybrać nazwę zmiennej sterującej (rys. 6).

Na rys. 7 pokazano bardzo prostą aplikację, która posłużyła do przetestowania pracy wyjść analogowych. Połączenie funktorów *And* i *Not* tworzy prosty generator. Owszem, oprogramowanie *Alfy* oferuje moduł *Flicker*, który jest generatorem astabilnym, jednak jego minimalny okres to 20 ms (50 Hz), przez co generowany przebieg schodkowy miał okres 40 s, natomiast generator prezentowany na ekranie umożliwiał uzyskanie okresu tego przebiegu równego około 4 s. To dziesięć razy mniej! W ten sposób można go było łatwo oglądać na ekranie oscyloskopu. Mówiąc szczerze, nie wierzyłem, że tak „kombinatoryka” zadziała, ale mimo tego świetnie spełniała ona swoje funkcje. Owszem, jego funkcjonowanie pozostaje w sprzeczności z podanymi wyżej parametrami przetwornika, ale i ten radzi sobie znakomicie, co widać na oscylogramie z rys. 8, co dobrze świadczy o sprzecz.

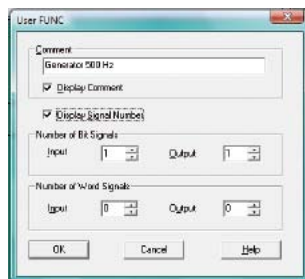


Rys. 9. Umieszczenie przycisku *User Func*

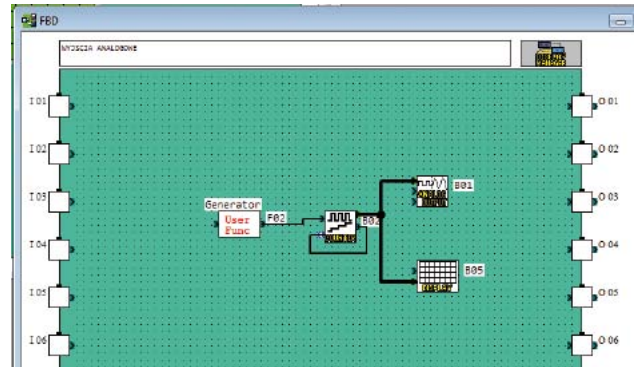
Wyjście sygnalizujące osiągnięcie przez blok *Counter* wartości zadanej (wartość tę wpisuje się w pole *Number of Counts*) podłączono do wejścia zerowania licznika. Dzięki temu liczy on do 4000, następnie samoczynnie zeruje się i cykl jest powtarzany. Wyjście liczbowe licznika podłączono do bloku wejścia sterującego przetwarzaniem bloku *Analog Output*, dla którego wybrano tryb napięciowy oraz do wejścia bloku wyświetlacza po to, aby użytkownik widział, że program działa.

**Funkcje użytkownika**

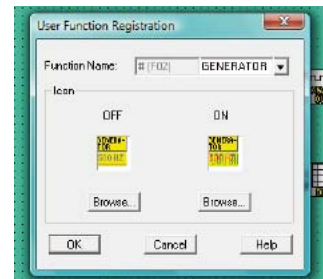
Raz opracowane i często stosowane bloki funkcjonalne można zapisać w postaci symboli, rodzaju podprogramów, tak aby można było używać ich bez potrzeby ponownego, mozolnego układania elementów. Aby wytłumaczyć w jaki sposób można utworzyć własny symbol i jak nimi zarządzać, posłużymy się prostym przykładem. Z „dziwnego generatora” zbudowanego na bramkach utworzymy symbol, chociaż w jego przy-



Rys. 10. Okno właściwości generatora z przykładu



Rys. 12. Podłączenie własnego bloku funkcyjnego



Rys. 13. Okno właściwości bloku wyświetlane po *Register*

padku nie trzeba zbyt wielkiego wysiłku, aby ponownie ułożyć komponenty. Sposób postępowania jest jednak identyczny, bez względu na stopień komplikacji tworzonego komponentu, więc nie ma sensu utrudniać sobie życia. Wszak to tylko ilustracja zasady.

Po lewej stronie ekranu należy kliknąć na przycisk *User Func* (rys. 9), następnie gdzieś w obrębie okna roboczego FBD. Pojawi się okno właściwości właśnie tworzonego podprogramu, w którym można wpisać komentarz, określić czy ma być wyświetlany, ustawić liczbę sygnałów bitowych (pole *Number of Bit Signals*), oraz liczbę liczbowych zmiennych wejściowych typu *Word* (pole *Number of Word Signals*). Dokumentacja podaje, że maksymalnie można zdefiniować do 8 wejść. Dla potrzeb definiowanego generatora wystarczy jed-



Rys. 14. Ikona podprogramu użytkownika

no wejście bitowe, jedno wyjście bitowe, bez wejść/wyjść typu liczbowego (rys. 10). Po kliknięciu OK na ekranie otworzy się okienko *Sub FBD*, w obszarze którego, stosując ogólne zasady rysowania, należy rozmieścić elementy funkcyjne (rys. 11). Jednocześnie na głównym schemacie pojawia się nowy bloczek funkcyjny, który można podłączyć w obrębie własnego schematu (rys. 12), opisany jako *User Func*. Na tym w zasadzie można zakończyć. Zyskuje się w ten sposób uproszczenie schematu, jednak to nie koniec możliwości.

Zdefiniowaną w ten sposób funkcję można dodać do paska menu zawierającego funkcje użytkownika. W tym celu należy wskazać bloczek na ekranie, kliknąć na nim prawym

klawiszem myszki i z menu kontekstowego wybrać *Register*. W okienku właściwości (rys. 13) wpisać nazwę tworzonego bloczku, wskazać ikonę dla stanu *Off* oraz dla stanu *On* i nacisnąć OK (ikonka powinna mieć wymiar 34×34 piksele i 16 lub 256 kolorów). Funkcja pojawi się na pasku po lewej stronie ekranu, z ikonką wskazaną dla stanu *Off* (rys. 14). Ikonka *On* będzie używana tylko w czasie symulacji i monitorowania, jeśli blok będzie aktywny.

W tym miejscu jedna uwaga związana z użytkowaniem IDE Alfy pod kontrolą środowiska Windows Vista. Aby móc dodawać do paska narzędzi bloczki musiałem ręcznie ustawić uprawnienia do katalogu, w którym zainstalowany jest program. Przy dodawaniu

nowych bloczków do paska program kopiuje ikony i funkcje do odpowiednich podkatalogów w ścieżce *Program Files\ALVLS\Library*. Zauważyłem, że domyślnie wyłączone jest uprawnienie *Create*, co uniemożliwia poprawne dodawanie symboli. Użytkownicy Windows XP i starszych systemów, zapewne nie napotkają tego problemu.

Warto jeszcze wspomnieć, że utworzonymi bloczkami można podzielić się z innym programistą lub przenieść je na inny komputer. Służą do tego opcje w menu *File: Export Registered User Func* (wysłanie) oraz *Import Registered User Func* (wgranie).

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

R E K L A M A

# ZESTAWY LUTOWNICZE

**GAS/SET**

## LUTOWNICA 10-60 W ZESTAW

cena: 136 zł

- \* przenośna lutownica, zasilana gazem do zapalniczek
- \* zapalarka wbudowana w osłonę grotu
- \* czas nagrzewania grotu <40 s
- \* zapas paliwa wystarcza na 90 min pracy
- \* temperatura grotu (max.): 450°C
- \* regulacja temperatury
- \* w zestawie lutownica, grot, pojemnik na topnik, etui



**Gaz do lutownic RONSON 300ml 8 zł**

**TERAZ TANIEJ O 15%**

**GAS/PROSET**

## LUTOWNICA 25-125 W ZESTAW

cena: 212,50 zł

- \* zapłonnik piezo
- \* możliwość łatwej zmiany grotów
- \* zbiornik paliwa ze stali
- \* zapas paliwa na 2 godziny pracy
- \* czas rozgrzewania grotu 30 s
- \* maksymalna temperatura grotu 580°C
- \* regulacja temperatury
- \* w zestawie grot, pojemnik na topnik, etui



**WWW.SKLEP.AVT.PL**