PRACE NAUKOWE Akademii im. Jana Długosza w Częstochowie
SERIA: Edukacja Techniczna i Informatyczna 2011 z. VI

Maciej Drabik
Akademia im. Jana Długosza w Częstochowie

ZASTOSOWANIE PROGRAMOWANIA GRAFICZNEGO W DYDAKTYCE AUTOMATYKI

Streszczenie

Artykuł omawia podstawowe funkcje logiczne, opisuje budowę, działanie i sposób programowania graficznego sterownika logicznego ALPHA oraz przedstawia przykład zastosowania tego sterownika w dydaktyce automatyki.

Słowa kluczowe: programowanie graficzne, automatyka, funkcje logiczne, sterownik programowalny

Wstęp

Dydaktyka każdego z przedmiotów technicznych ma swoją specyfikę. Dydaktyka powinna umożliwić studentowi zrozumienie przedmiotu, czyli nauczyć go umiejętności zastosowania zdobytej wiedzy w praktyce. W nauczaniu automatyki dużą rolę odgrywają: logika, funkcje logiczne oraz algebra funkcji logicznych - czyli algebra Boole'a. One to właśnie stanowią podstawę do projektowania logicznych układów sterowania najpowszechniej spotykanych w życiu codziennym.

Funkcje logiczne

Przypomnijmy podstawowe funkcje logiczne stosowane w automatyce takie jak: AND, OR, NOT, XOR, NAND, NOR, w których zarówno zmienna

wejściowa, jak i zmienna wyjściowa mogą przybierać tylko dwie wartości umownie określone jako „0” i”1”.

Funkcja AND (iloczyn logiczny)

Wyjście funkcji AND przybiera wartość „1”, gdy wszystkie wejścia mają wartość „1”. Dowolne wejście będące w stanie”0” ustawia wyjście na „0”.

Tab. 1. Tabela logiki funkcji AND

In	In	Out
1	1	1
1	0	0
0	1	0
0	0	0

Funkcja OR (suma logiczna)

Wyjście funkcji OR przyjmuje wartość „1”, gdy chociaż jedno wejście jest w stanie „1”. Gdy wszystkie wejścia są „0” wyjście jest także „0”.

Tab. 2. Tabela logiki funkcji OR

In	In	Out
1	1	1
1	0	1
0	1	1
0	0	0

Funkcja NOT (negacja)

Funkcja NOT zmienia stan logiczny sygnału wejściowego na przeciwny. Gdy wejście jest logiczną „1” na wyjściu otrzymujemy logiczne „0” i odwrotnie.

Tab. 3. Tabela logiki funkcji NOT

In	Out
1	0
0	1

Funkcja XOR (exclusive OR)

Funkcja XOR przyjmuje na wyjściu wartość „1” wtedy, gdy wejścia funkcji są w różnym stanie logicznym tzn. jedno jest w stanie „0”, a drugie jest w stanie „1”.

Tab. 4. Tabela logiki funkcji XOR

In	In	Out
1	1	0
1	0	1

Funkcja NAND (NOT AND, negacja iloczynu)

Funkcja ta przyjmuje na wyjściu wartość „1”, gdy co najmniej jedno wejście jest w stanie „0”. Jeżeli wszystkie wejścia są w stanie „1”, wyjście ustawia się na „0”.

Tab. 5. Tabela logiki funkcji NAND

In	In	Out
1	0	1
0	1	1
0	0	1
1	1	0

Funkcja NOR (NOT OR, negacja sumy)

Wyjście funkcji NOR przybiera wartość „1”, gdy wszystkie wejścia są w stanie „0”. Wyjście jest w stanie „0”, gdy co najmniej jedno wejście jest logiczną „1”.

Tab. 6. Tabela logiki funkcji NOR

In	In	Out
1	0	0
0	1	0
1	1	0
0	0	1

Zrozumienie samych funkcji logicznych nie jest trudne. Natomiast umiejętność łączenia tych funkcji w układy sterowania zapewniające pożądane zachowanie obiektów sterowania jest znacznie trudniejsze.

Programowanie graficzne

W nauczaniu automatyki zastosowanie komputera i programowania odgrywa bardzo znaczącą rolę. Klasyczny sposób wykorzystania komputera w automatyce polegał początkowo na opracowaniu od zera programu sterującego wraz z poleceniami do komunikacji komputera z urządzeniami wejścia i wyjścia. Używano do tego celu języków wysokiego poziomu (Basic, Pascal, C). Sposób ten umożliwiał odbiór informacji z czujników, ich przetwarzanie oraz wysyłanie poleceń – będąc jednak bardzo pracochłonnym. Drugim krokiem w kierunku uproszczenia projektowania i uruchamiania systemów pomiarowych było opracowanie przez wiodących producentów przyrządów pomiarowych standardu, który określał metody programowania przyrządów pomiarowych. W latach dziewięćdziesiątych powstał standard SCPI (Standard Commands for Programmable Instruments). Był to zestaw instrukcji, który pozwalał na pełne zaprogramowanie pracy przyrządu pomiarowego niezależnie od jego rodzaju. Kolejnym osiągnięciem w dziedzinie projektowania oprogramowania systemów pomiarowych było powstanie zintegrowanych środowisk pomiarowych. W początku lat dziewięćdziesiątych rozwój języków programowania wyższego poziomu (Pascal czy C) i graficzny system operacyjny Windows stworzyły warunki dla rozwoju tych środowisk. Na tej bazie różne firmy (National Instruments, Hewlett–Packard i inne) zaczęły budować pakiety programowe wspomagające projektowanie systemów pomiarowych o dużych możliwościach pomiarowych i jednocześnie łatwej obsłudze. Powstały środowiska tekstowo-graficzne (np. LabWindows/CVI) i typowo graficzne (LabVIEW). Środowiska tekstowo-graficzne są połączeniem projektowania graficznego z pisaniem kodu programu, zaś graficzne pomijają zupełnie pisanie kodu programu. W następnych latach nastąpił rozwój prac nad systemami wspomagania projektowania układów automatyki przemysłowej. Powstały takie systemy, jak BridgeVIEW i Lookout firmy National Instruments oras systemy innych firm (Keithley Instruments, Advantech, Hewlett–Packard). Okazało się wkrótce, że lepiej są odbierane środowiska czysto graficzne, a klasyczne języki programowania odchodzą do lamusa. Obecnie znakomita większość środowisk programowania systemów pomiarowych, to środowiska graficzne [1]. Wynika to z faktu, że użytkownikiem takich środowisk może być zwykły użytkownik komputera – nie znający żadnych języków programowania, gdyż układanie programu polega na umiejętnym doborze symboli (ikon) w polu programowania i ich prawidłowym łączeniu. Przyjęta terminologia, stosowane symbole graficz-

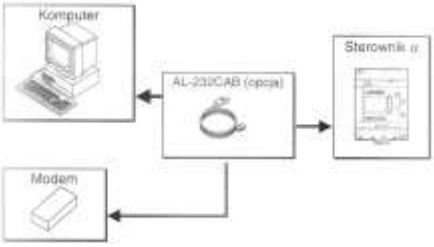
ne oraz blokowy sposób przedstawienia działania programu nie różnią się od notacji powszechnie używanej przez inżynierów. Ponadto „obrazkowy” charakter środowiska ułatwia graficzną prezentację danych [2].

Sterownik ALPHA

Przykładem zastosowania programowania graficznego jest sterownik ALPHA firmy Mitsubishi (rys. 1). Jest to urządzenie mikroprocesorowe, które może sterować różnymi urządzeniami. Posiada 6 wejść przekaźnikowych (przyjmujących wartości sygnału 0 i 1) i 4 wyjścia, a zatem może obsługiwać 6 sygnałów wejściowych i sterować jednocześnie czterema obiektami.

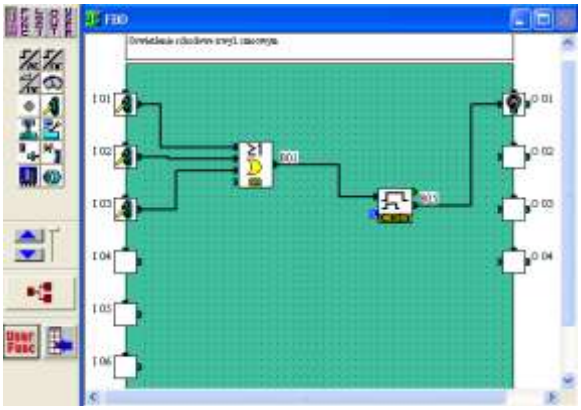
Rys. 1. Sterownik logiczny ALPHA model AL.-10MR-D

Sterownik programujemy przy użyciu bloków funkcyjnych tzn. bloków realizujących rozmaite funkcje. Oprócz omówionych wcześniej sześciu funkcji logicznych sterownik posiada jeszcze 16 bloków realizujących inne funkcje użyteczne w sterowaniu. Programowanie sterownika odbywa się na komputerze połączonym odpowiednim przewodem ze sterownikiem (rys. 2).



Rys. 2. Transfer programu z komputera do sterownika

Do programowania sterownika służy program graficzny AL.-PCS/WIN-E pracujący w środowisku Windows. Platformą do programowania sterownika jest tzw. Baza Diagramu Bloków Funkcyjnych (FDB). Jest to pole na ekranie komputera (rys. 3), które posiada na lewej krawędzi kwadraty odpowiadające wejściom sterownika i kwadraty na lewej krawędzi będące wyjściami. Na lewo od pola FDB są ikony urządzeń wejścia (IN), urządzeń wyjścia (OUT), funkcji logicznych (LOG), funkcji pozostałych (FUN) oraz ikona przewodów połączeniowych. Programując sterownik umieszczamy w wejściach i wyjściach pola FDB odpowiednie elementy (lampki, wyłączniki, grzałki, diody, tłoki itp. W polu FDB rozmieszczamy odpowiednie bloki funkcyjne. Łączymy wejścia z blokami funkcyjnymi, a bloki z wyjściami. Operacji tych dokonujemy znaną metodą „chwyć i upuść”- klikając na odpowiednią ikonę i zwalniając przycisk myszy w miejscu w którym chcemy umieścić ikonę. Projektowanie układu sterowania sprowadza się więc do układania „logicznych klocków” i łączenia ich między sobą.



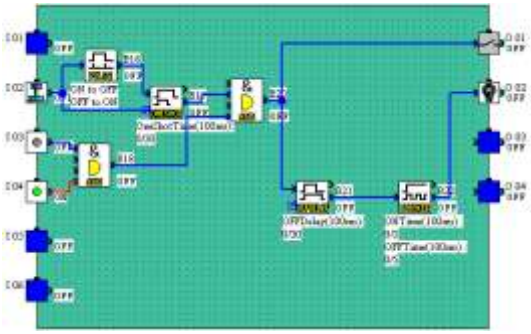
Rys. 3. Baza diagramu bloków funkcyjnych

Po zaprojektowaniu układu sprawdzamy jego działania, przechodząc w tryb symulacji. Klikając na wejścia zmieniamy wartość sygnału z 0 na 1

i odwrotnie. Zmieniają się kolory na ekranie. Tam gdzie sygnał ma wartość 1 pojawia się kolor czerwony. Wejścia, wyjścia i przewody w których jest brak sygnału pozostają niebieskie. Możemy śledzić przebieg sygnału przez cały układ sterowania- od wejścia do wyjścia układu.

Jeżeli zaprojektowany przez nas układ sterowania nie działa w trybie symulacji, tak jak powinien, przechodzimy ponownie do trybu projektowania. Dokonujemy zmian i ponownie sprawdzamy układ w trybie symulacji. Gdy działa prawidłowo wówczas przegrywamy program sterowania z komputera do sterownika i sprawdzamy jak sterownik steruje podłączonym do niego rzeczywistym obiektem.

Na rysunku 4 przedstawiono program sterowania szlabanem parkingowym. Szlaban ten jest uruchamiany i zatrzymywany sygnałem z pilota (wejście I01). Sygnał wyjściowy który powoduje ruch szlabanu jest podawany na wyjście O01. Jednocześnie ten sam sygnał doprowadzony jest poprzez człon opóźniający i generator impulsów do wyjścia O02 gdzie umieszczona jest lampa ostrzegawcza. Człon opóźniający i generator impulsów powodują, że lampa ostrzegawcza miga podczas ruchu szlabanu oraz jej działanie lampy jest opóźnione względem tego ruchu. Na wejściach I03 oraz I04 zaznaczone są symbolicznie fotokomórki, które umieszczone być powinny przed i za szlabanem. Wstrzymują one ruch szlabanu, gdy samochód znajdzie się w ich zasięgu. Zapobiega to uszkodzeniu samochodu przez szlaban. Sygnały z tych fotokomórek dochodzą do członu iloczynu logicznego, aby sygnał z co najmniej jednej mógł zablokować ruch szlabanu. Sygnał wynikowy z fotokomórek dochodzi do członu iloczynu logicznego wraz z sygnałem z pilota, gdyż tylko jednoczesny sygnał z pilota i obu fotokomórek może uruchomić szlaban. Sygnał z pilota dochodzi wcześniej do członu PULSE który wytwarza impuls jednostkowy przy zmianie stanu wejścia pilota (z ON na OFF i odwrotnie) oraz do członu ONE SHOT wytwarzającego impuls o zaprogramowanej długości trwania umożliwiający zamknięcie lub otwarcie bramy.



Rys. 4. Program sterowania szlabanem parkingowym

Podsumowanie

Zastosowanie programowania graficznego w dydaktyce automatyki ma kilka zalet:

- Umożliwia poznanie metody programowania graficznego,
- Pozwala na wizualizację przebiegu sygnałów sterowania i zrozumienie działania całego układu,
- Umożliwia poprawę i optymalizację zaprojektowanego układu,
- Pozwala na sprawdzenie działania układu sterowania w rzeczywistym obiekcie.

Maciej Drabik
Akademia im. Jana Długosza w Częstochowie

APPLICATION OF THE GRAPHICAL PROGRAMMINGIN THE TEACHING OF AUTOMATIZATION

Summary

This article presents possibilities of application of the graphical programming in the teaching of automatization in polish schools. It describes logical functions, structure and method of his programming. It describes advantages of this teaching, too.

Keywords: graphical programming, automatic, logical functions, programmable controller