

## Sterownik programowalny najprościej

### Dlaczego nie sięgamy po nowe rozwiązania?

Wydaje się, że wśród wielu powodów najważniejsze to: brak przekonania o takiej konieczności, niepewność czy nowe nie okaże się zbyt trudne i kłopotliwe, wątpliwości co do korzyści, no i brak okazji. Sprawa nie jest tak prosta, jak może się wydawać, tym bardziej że wiele produktów wyposażanych jest w cechy korzystne głównie ze względów marketingowych, w praktyce mało przydatnych.

Z pewnością do technicznych nowości, które zyskały już ugruntowaną pozycję należą sterowniki programowalne. Zostaje więc tylko problem, jak szybko poznać je praktycznie i ocenić, czy będą przydatne w zamierzonych zastosowaniach.

Być może tylko dla porządku przypomnijmy, że najprostszy sterownik programowalny zbudowany jest z:

- wejść wykrywających zmiany dwustanowe (jest sygnał albo go nie ma) lub analogowe (jaką wartość ma sygnał dołączony do wejścia)
- wewnętrznego mikroprocesorowego układu sterującego z programowalną pamięcią
- wyjść w formie styków przełącznika lub tranzystorów sterowanych układem mikroprocesorowym, zgodnie z wolą użytkownika zapisaną w programowalnej pamięci.

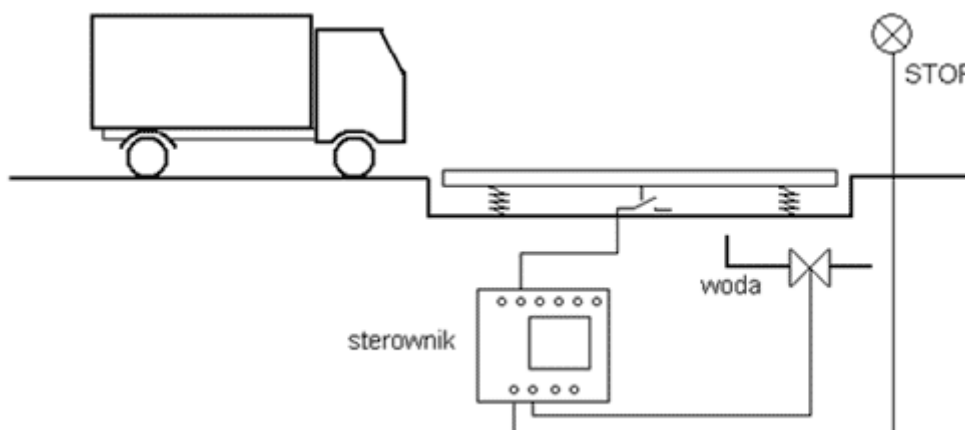
Dodatkowe układy to: wewnętrzny zasilacz dostosowujący sterownik do typowego zasilania np. 12-24 V lub 230 V, port do komunikacji z komputerem, czasami zegar czasu rzeczywistego RTC, wyświetlacz i klawiatura. Szczegóły techniczne podają instrukcje, dlatego poniżej proponujemy szybkie i praktyczne poznanie jednej z metod programowania sterowników, prawdopodobnie najłatwiejszej do prostych zastosowań. Metoda ta, to FBD (Function Block Diagram), czyli rysowanie na ekranie komputera schematu (diagramu) z użyciem dostępnych bloków funkcyjnych. Tworzony diagram podobny jest do schematu obwodów elektrycznych. Sygnały wejściowe całego schematu odpowiadają wejściom sterownika, a wyjściowe są wyjściami sterownika (np. przełączniki). Dostępne w metodzie FBD bloki i ich działanie, są bardzo zbliżone do elementów stosowanych w technice cyfrowej. Nie oznacza to, że osoby nie mające dotychczas doświadczenia z techniką cyfrową nie poradzą sobie z programowaniem FBD. Programy komputerowe przeznaczone do programowania sterowników wyposażane są w symulacje działania bloków i całego diagramu. Największą zaletą FBD w stosunku do metody drabinkowej LAD, czy wpisywania komend, jest możliwość łatwego śledzenia przebiegu sygnałów od wejścia do wyjścia sterownika w formie ciągłej, podświetlanej linii. Nie musimy więc uważać na ukryte skoki w linijkach programu czy szczeblach drabinki, bo w FBD wszystko widać.

Wydaje się, że ktoś kto widział skomplikowany schemat układu przełącznikowego (np. fragmentów dawnej centrali telefonicznej) nie ma wątpliwości, dlaczego technika cyfrowa wprowadziła swoje symbole logiczne, zamiast kontynuować pozornie prostsze stosowanie symboli styków.

### Ćwiczenie praktyczne dla osób początkujących

Zadaniem jest wykonanie prostego projektu metodą FBD i sprawdzenie działania poprzez symulację programową, a realizujące następujące funkcje:

Pojazd (np. śmieciarka opuszczająca wysypisko) wjeżdżający na platformę z czujnikiem obciążenia ma być spryskany wodą przez czas 20 sek. Włączenie natrysku należy opóźnić o 5 sek. od chwili wykrycia obciążenia pojazdem. W czasie oczyszczania pojazdu powinno zapalić się czerwone światło sygnalizujące. Poniżej przedstawiono rysunek obiektu do zadania.



Rys. Automatyka myjni podwozia

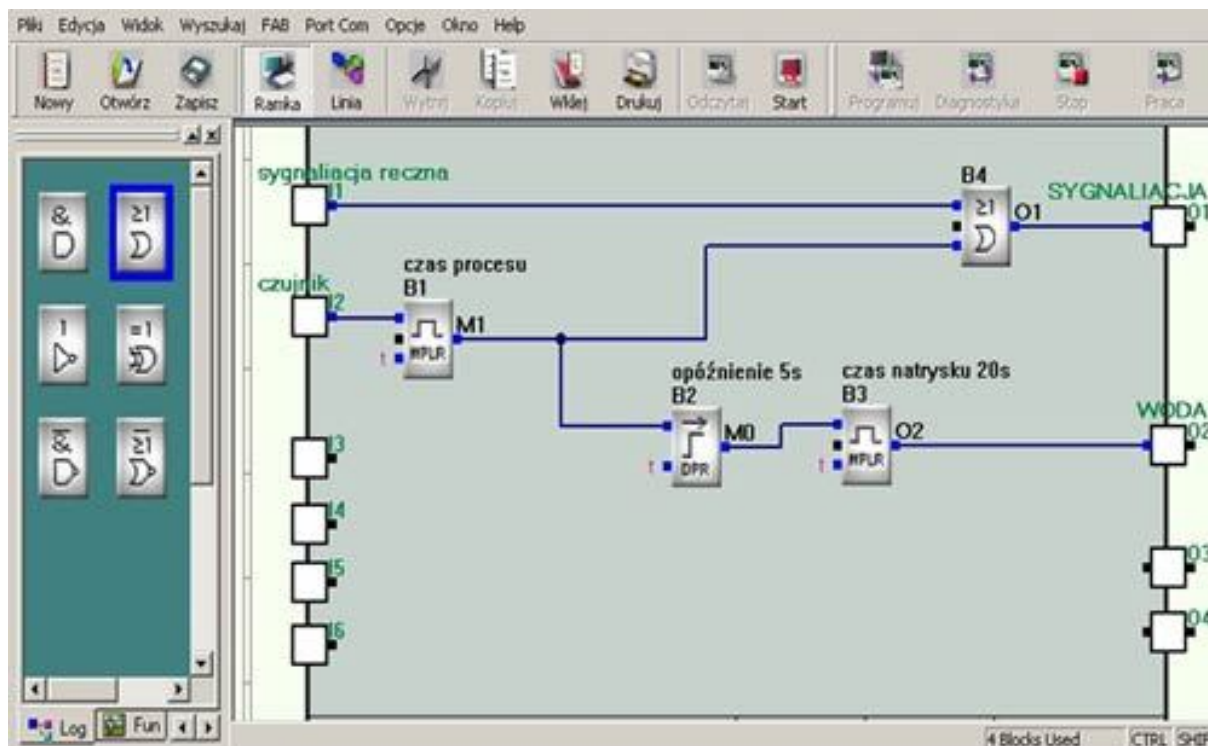
Wersja demonstracyjna nie wymaga instalacji ale nie można nią programować sterownika. Jednak wykonany, przetestowany i zapisany projekt jest pełnowartościowy. Uruchomienie programu QuickII.exe spowoduje pojawienie się pola z odpowiednimi narzędziami do rysowania diagramu lub otwierania gotowych już plików. Na początku należy więc zdecydować i wybrać polecenie "Otwórz" lub "Nowy" a następnie wykonywać czynności zgodnie z naniesioną na rysunku numeracją.



Rys. Wygląd uruchomionego programu QuickII

- 1 - decyzja o wykonaniu nowego projektu
- 2 - wybór wielkości sterownika (np. 6 wejść, 4 wyjścia) i akceptacja przyciskiem OK
- 3 - wybór grupy dostępnych bloków funkcyjnych (pole wyżej)
- 4 - aktualnie dostępne bloki funkcyjne, przenoszone na pole rysowania diagramu
- 5 - przycisk uruchamiający funkcję rysowania połączeń między blokami

W konkretnym, naszym zadaniu, wygląd gotowego diagramu może wyglądać jak na rysunku poniżej:

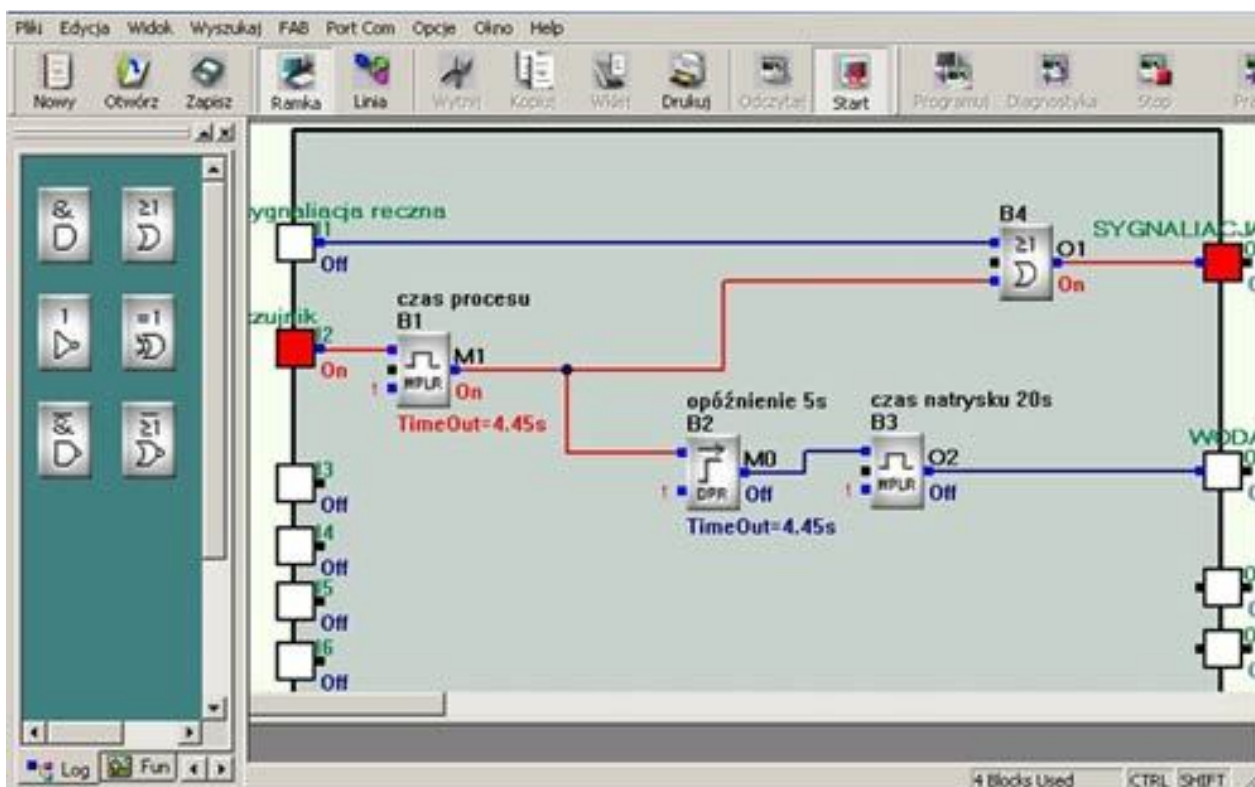


Rys. Przykład projektu działania sterownika myjni podwozia

Przykład jest bardzo skromny, ale pozwala poznać ważniejsze zasady. Z lewej strony symbolicznie zaznaczone są wejścia sterownika I1-I6, z prawej wyjścia Q1-Q4. Do wejść podawane jest napięcie jako sygnał sterujący. Wyjściami są niezależne styki przełączników, zamykające obwód sygnalizacji i elektrozasoru wody. Do realizacji użyto 4 bloków (licznik w prawym dolnym rogu ekranu), z możliwych do wykorzystania 127. Pokazuje to potencjalne możliwości nawet bardzo prostego sterownika.

Jak to działa? Po podaniu napięcia na wejście I2, czyli pojawieniu się stanu wysokiego, blok B1 na swoim wyjściu wygeneruje impuls o czasie trwania 30 sek (ustawiany we właściwościach bloku). Sygnał z wyjścia bloku B1 poprzez element sumujący sygnały (podobnie jak równolegle połączone styki) B4 przepisywany jest na wyjście Q1 sterownika (zapalenie sygnalizacji). Jak widać możliwe jest również ręczne zapalenie sygnalizacji, przez drugie wejście I1. Sygnał z wyjścia B1 podawany jest również do wejścia bloku B2 opóźniającego pojawienie się jego na wejściu B3 o 5 sekund. Blok B3 wygeneruje impuls o czasie 20 sek, czyli wymaganym czasie natrysku wody.

Działanie całego projektu, czy jeszcze wcześniej pojedynczych bloków, najłatwiej sprawdzić uruchamiając w programie Quick symulację działania sterownika (przyciskiem "Start"). Nasz komputer wykona narysowane zadanie, dodatkowo sygnalizując zmianą koloru stany wszystkich linii.



Rys. Widok programu Quick II z włączoną symulacją pracy sterownika

Obserwując zmieniające się stany wejść, wyjść, linii czy odliczane w blokach czasy, jako reakcje na wymuszenia od strony wejść sterownika, (przez kliknięcie myszką) możemy analizować działanie naszego projektu. Podczas symulacji innych diagramów również będziemy mogli obserwować zmieniające się stany liczników, wyjść bloków zegarowych, kalendarza czy efekty porównań sygnałów analogowych w komparatorach itp.

Po przesłaniu projektu do sterownika (przez port RS232, USB itp.), jego mikroprocesor będzie wykonywał ten sam program analizując w niekończącej się pętli blok po bloku. Jednak dla zapewnienia właściwej kolejności analizowania bloków przez sterownik, przez analogię z kierunkiem przepływem prądu od wejścia do wyjścia sterownika, pod koniec prac projektowych należy uporządkować numery bloków. Zasadą powinno być, aby blok bliżej źródła sygnału miał numer niższy (był przez to wcześniej analizowany) niż blok następujący po nim. Na rysunku bliżej wejścia (źródła sygnału) jest blok B1, po nim następują bloki z numerami B2 i B4. W programie Quick zmianę numeru wykonuje się "ręcznie" ale niekiedy realizowane jest to automatycznie np. w SuperCad.

## Uruchomienie obiektu

Właściwie po zapisaniu programu do sterownika i wykonaniu połączeń elektrycznych powinno już wszystko działać. A jeśli nie, to pomocna przy uruchomieniu będzie funkcja programu Quick II, nazwana "monitor". Tym razem podświetlane wejścia i wyjścia czy wewnętrzne linie diagramu odzwierciedlać będą faktyczne stany pracy sterownika (już nie symulacji). Jednak brak oczekiwanej zmiany sygnału z czujnika, czy prawidłowe wystrojenie przełącznika możemy zaobserwować również na wyświetlaczu LCD.