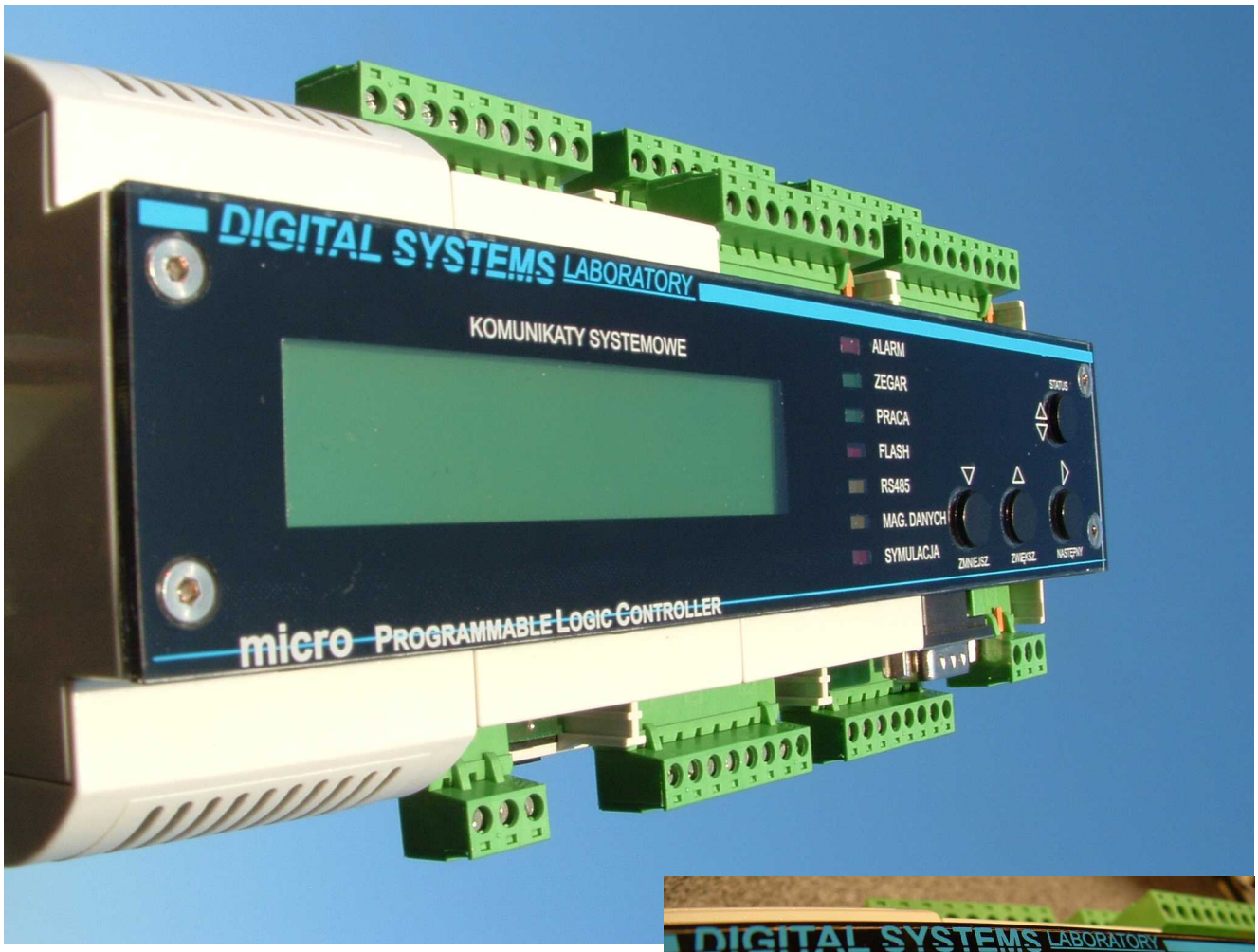
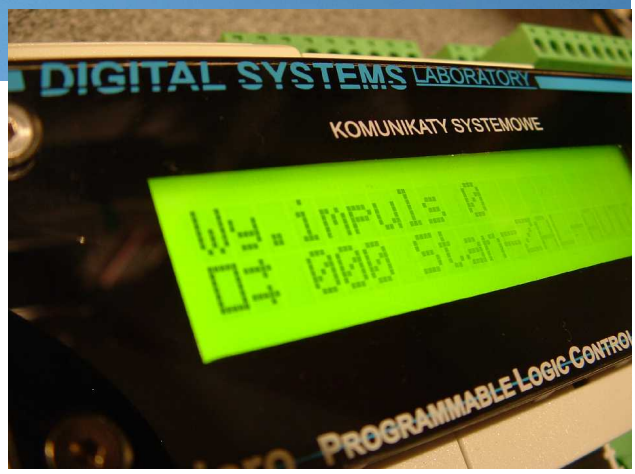
Opis funkcji sterownika swobodnie programowalnego





microPLC

V1.06 – 01.06.2009

1.WPROWADZENIE DO PROGRAMOWANIA microPLC 4

Zasady programowanie drabinkowego....9

2.FUNKCJE PODSTAWOWE 10

||--| |-- Rozpoczęcie operacji logicznej - proste....10
||--|/|--- Rozpoczęcie operacji logicznej - zanegowane....11
---( )--- Przekaźnik wyjściowy....13
---(S)--- Przekaźnik wyjściowy załączany stanem wysokim - SET....14
---(R)--- Przekaźnik wyjściowy wyłączany stanem wysokim - RESET....16
---| |--- Szeregowe połączenie styków – logiczne AND....17
---|/|--- Szeregowe połączenie styków zanegowane – logiczne NAND....19
‘--| |--‘ Równoległe połączenie styków – logiczne OR....20
‘--|/|--‘ Równoległe połączenie styków zanegowane – logiczne NOR....21
---/--- Negacja stanu gałęzi....22
---|DIF|--- Wyszukanie zbocza narastającego....23
---|DFN|--- Wyszukanie zbocza opadającego....24
---(TD)--- Przekaźnik czasowy – opóźnione załączenie (Time Delay)....25
---(SS)--- Przekaźnik czasowy – opóźnione wyłączenie (Single Shot)....26

3.RELACJE LOGICZNE 28

||---| A == B |--- Warunek równości....28
||---| A != B |--- Warunek nierówności....30
||---| A < B |--- Warunek mniejszości....31
||---| A <= B |--- Warunek mniejszy równy....32

4.FUNKCJE MATEMATYCZNE 33

---| D = S |--- Przypisanie....33
---[D = S1 + S2]--- Suma dwóch argumentów....34
---[D = S1 - S2]--- Różnica dwóch argumentów....35
---[D = S1 * S2]--- Iloczyn dwóch argumentów....37
---[D = S1 / S2]--- Iloraz dwóch argumentów....38
---[D = S1 OR S2]--- Suma logiczna dwóch argumentów....39
---[D = S1 AND S2]--- Iloczyn logiczny dwóch argumentów....40
---[D = S1 XOR S2]--- Alternatywa logiczna dwóch argumentów....41

5.FUNKCJE STERUJĄCE 43

----[JMP->] Skok warunkowy....43
----[MCS_]--- Sterownie blokiem programu - inicjacja....44
----[MCR_]--- Sterownie blokiem programu - zakończenie....44
----[MPS]--- Rozgałęzienie linii – Zapis stanu linii....44
----[MRD]--- Rozgałęzienie linii – Odczyt stanu linii....44
----[MPP]--- Rozgałęzienie linii – Ostatni odczyt stanu linii....45
||---[END]--- Koniec programu....46

6.FUNKCJE SPECJALNE 47

||---[STORE ANAL.IN] Rejestrator wejść analogowych....47
-----[LOG] Rejestracja dowolnych parametrów sterownika....47
----[POGODÓWKA] Krzywe grzewcze....48
----[PID] Regulator PID....52
||---[COPY CL/DT/IMP] Kopiowanie rejestrów systemowych....54
||---[LIGHT] Sterowanie scenami świetlnymi....57

----[MASTER WRITE] Zapis rejestru wewnętrznego w urządzeniu przyłączonym do sterownika 58
---[MASTER CALL] Inicjacja wymiany danych z innymi urządzeniami przyłączonymi do sterownika....59
----[MASTER RECEIVE] Odczyt odebranych danych z bufora magistrali szeregowej....60
||---[MODEM] Obsługa modemu telefonicznego....62
----[FILL] Wypełnienie obszaru jednakową wartością....62
||---[RANGE] Zmiana zakresu pomiarowego....63
||---[CONFIG] Konfiguracja sterownika....64
||---[TXT] Zapis tekstu do programu technologicznego....66

7.OBSŁUGA ZEGARA STEROWNIKA 67

8.ZESTAWIENIE REJESTRÓW STEROWNIKA 68

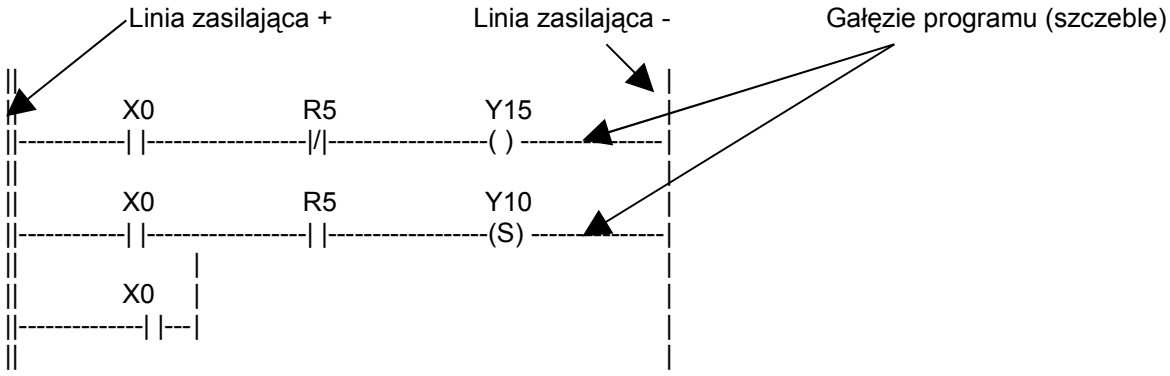
REJESTRY IMPULSOWE 68
REJESTRY ANALOGOWE 68
TABELA DOSTĘPNYCH REJESTRÓW....70

9.CZYM JEST PLC? 72

Program drabinkowy....73
Zasilanie....73
Moduły wejść/wyjść....73
Basic Operation of a PLC system....74
PLC Program....74
Basic Scan....74
Memory....75
Moduły wejść i wyjść w sterownikach PLC....75
Cyfrowe systemy I/O....75
Analogowe systemy I/O....75
Ilość punktów pomiarowych oraz rozmiar systemów PLC....76
Logika drabinkowa....76

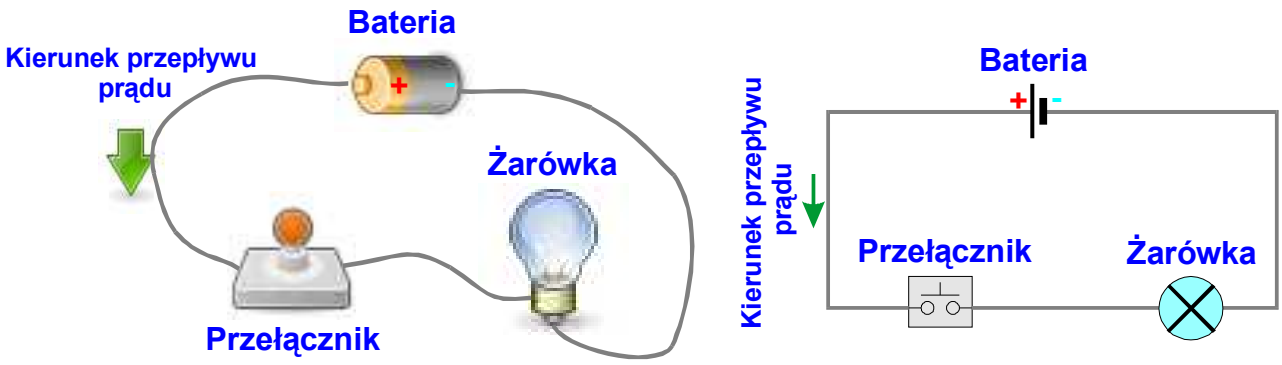
1.WPROWADZENIE DO PROGRAMOWANIA microPLC

Programowanie sterowników PLC odbywa się przy pomocy programu drabinkowego. Nazwa ”programowanie drabinkowe” wzięła się stąd, że program zapisywany do sterownika wygląda jak drabina. Po obydwu stronach programu znajdują się dwie pionowe linie traktowane jak bieguny zasilania, a pomiędzy nimi umieszczone są „szczeble”, popularnie zwane gałęziami, których zawartość reprezentuje wykonywany program.

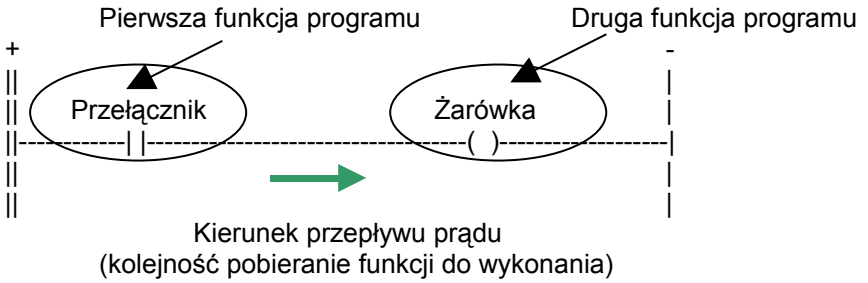


Teraz przejdźmy do wyjaśnienia sposobu programowania sterownika.
Aby wyjaśnić sposób konstruowania programu posłużymy się najprostszym przykładem. Będzie to połączenie źródła zasilania z żarówką załączaną przy pomocy przełącznika.

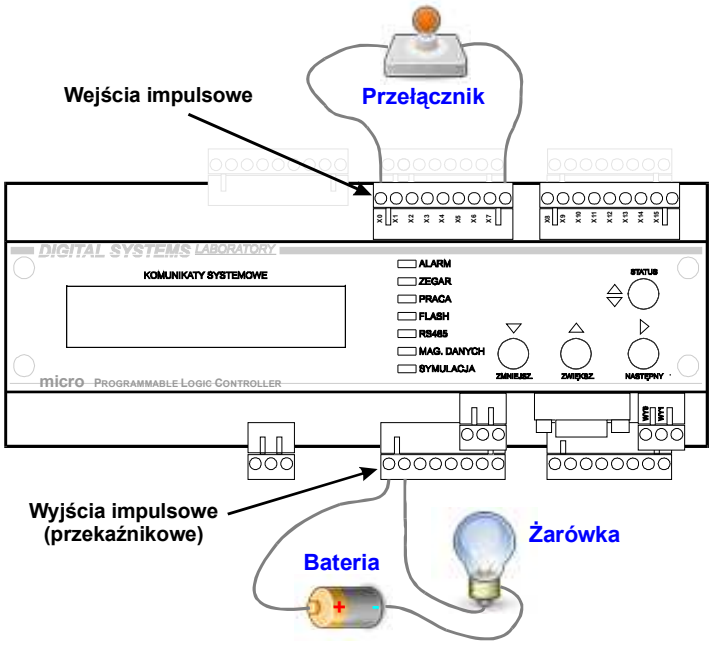
Oto schemat połączeń i odpowiadający im schemat elektryczny :

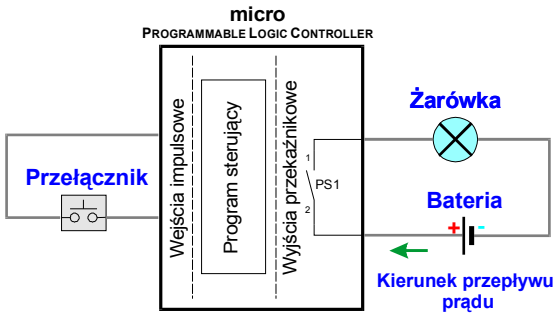


Przedstawiony układ jest zapewne znany Wam z lekcji fizyki. Jeżeli naciśniemy przełącznik - żarówka się zaświeci. Jeśli puścimy przełącznik - żarówka zgaśnie. Prąd w obwodzie płynie od dodatniego do ujemnego bieguna zasilania. Oto program drabinkowy realizujący to samo zadanie:



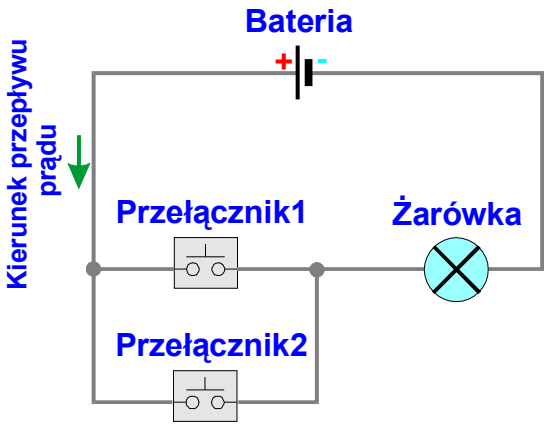
Zamknięcie styku przełącznika spowoduje podłączenie zasilania do żarówki. W pokazanym przykładzie programu kierunek przepływu prądu jest umowny. Tak naprawdę jest to kolejność, z jaką sterownik pobiera wykonywane funkcje do realizacji. Najpierw pobierany jest element z lewej strony, potem następny itd. Zobaczmy teraz jak wyglądałby równoważny schemat elektryczny podłączeń do samego sterownika:





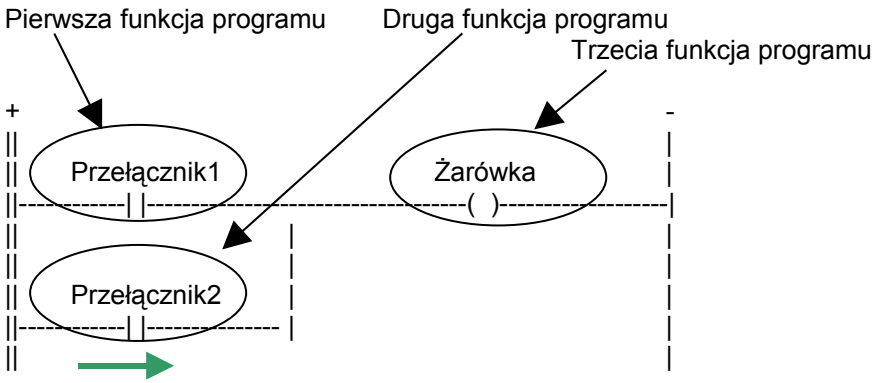
Po lewej stronie rysunku pokazano jak powinien wyglądać nasz układ podłączony do zacisków sterownika. Jak widać, przełącznik tak naprawdę nie styka się bezpośrednio z kablem do żarówki i baterii. Jego zaciski przyłączone są bezpośrednio do wejścia sterownika. Zaciski, do których dołączono przełącznik, nazywają się wejściami impulsowymi sterownika. Wejścia te służą do odczytu stanu przełączników, styków przekaźników itp. Od dołu do sterownika podłączono drugą część układu. Zaciski, do których włączona jest bateria i żarówka są podłączone do przekaźnika znajdującego się wewnątrz sterownika, który może być załączany lub wyłączany przez jego program. Zaciski te nazywane są wyjściami impulsowymi sterownika. Jak widać na schemacie elektrycznym, po prawej stronie rysunku przycisk i żarówka nie mają ze sobą nic wspólnego. Jest jednak coś co wiąże je ze sobą. Jest to program sterujący, który wie, że jeżeli naciśniemy przycisk, to ma on załączyć swój wewnętrzny przekaźnik zamykający obwód z żarówką. Ten program, jak już wspominaliśmy, nosi nazwę programu drabinkowego. W ten sposób, przy pomocy odpowiedniego programu, możemy przystosować pracę urządzenia do naszych indywidualnych potrzeb.

Teraz przetestujmy inny układ, w którym wykorzystujemy dwa przyciski połączone równolegle.



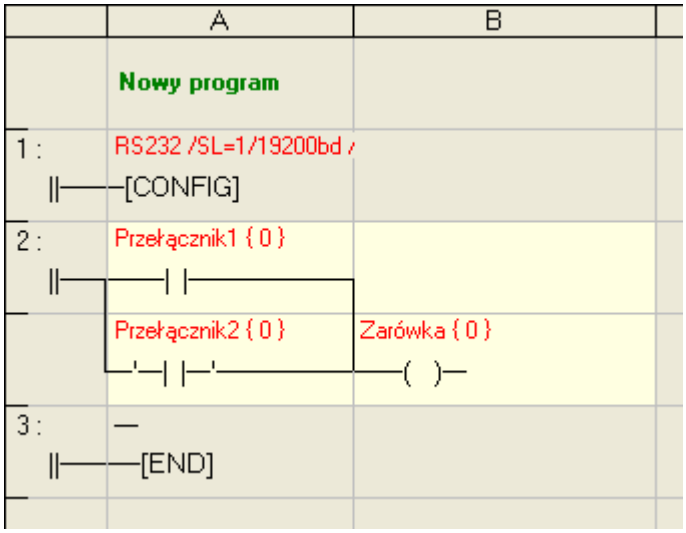
Jeżeli załączymy dowolny z przycisków, żarówka zaświeci się. Taki układ, w rozumieniu logiki drabinkowej realizuje funkcję logiczną OR czyli LUB.

Oto program drabinkowy odzwierciedlający powyższy układ:

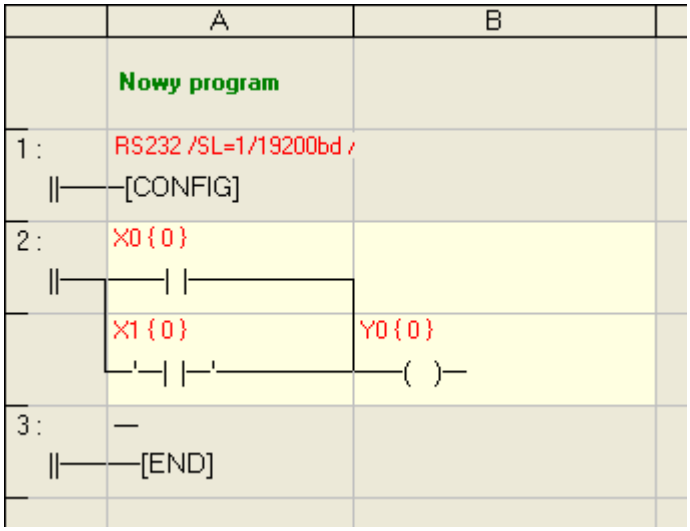


Kierunek przepływu prądu
(kolejność pobieranie funkcji do wykonania)

Żarówka zaświeci się jeżeli naciśniemy Przełącznik1 LUB Przełącznik2. Program realizowany jest następująco: najpierw sterownik sprawdza stan przełącznika pierwszego, potem sprawdza stan przełącznika drugiego i dopiero gdy zna stany obydwu przycisków, steruje przekaźnikiem wewnętrznym, który, jak już wspomniano, nazywa się przekaźnikiem wyjść impulsowych. Program wykonywany jest od lewej do prawej strony. Zobaczmy jak ten fragment programu wygląda by w edytorze dołączonym do sterownika:

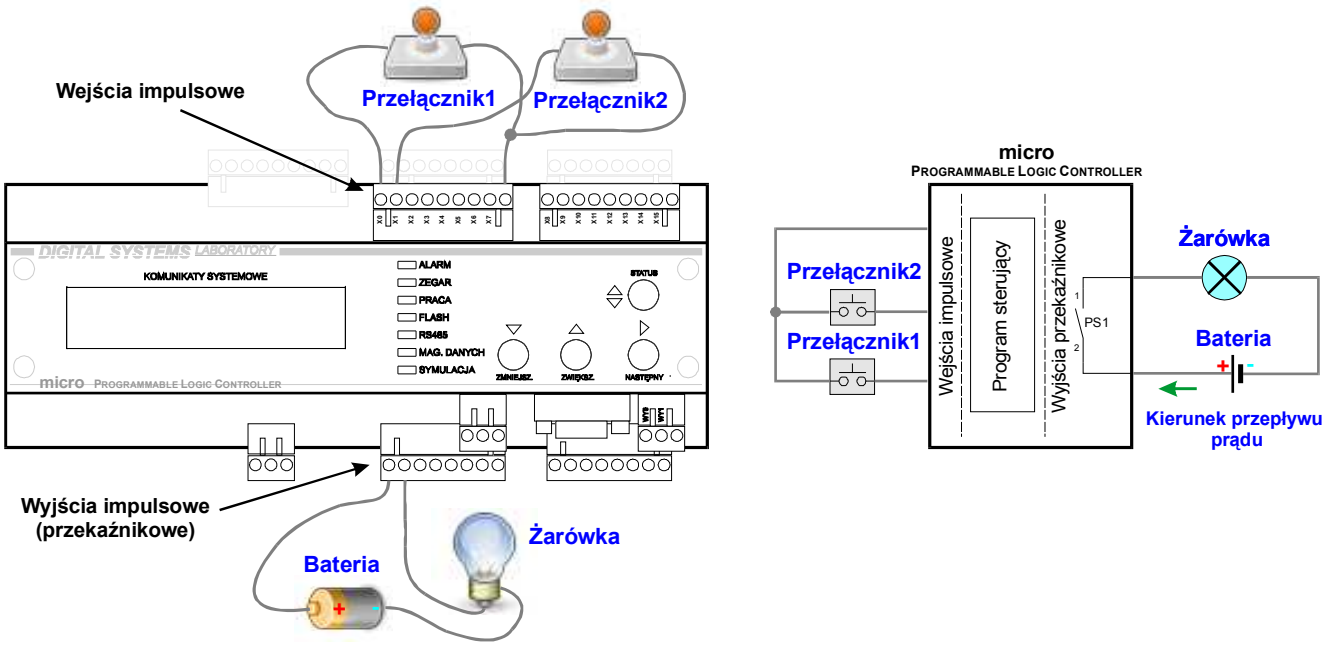


Na razie nie wyjaśniamy znaczenia linii pierwszej i trzeciej, ale muszą one występować w każdym programie. Każdy program musi zaczynać się od linii z funkcją konfiguracji [CONFIG] i kończyć funkcją [END] oznaczającą koniec programu sterującego. Jak widać w linii 2 znajduje się gałąź programu sterująca naszą żarówką. Z układu rodzaju użytych funkcji w tej gałęzi możemy wywnioskować jaką funkcję logiczną realizuje program. W programie nie będzie użyta nazwa OR lecz układ styków wejść impulsowych będzie realizował OR (sumę logiczną). Zobaczmy teraz jak wyglądałby program, który zamiast nazw opisujących przyłączone urządzenia zewnętrzne pokazywałby nazwy fizycznych wejść i wyjść impulsowych.



Jak widać, w nomenklaturze sterownika, przełączniki są podłączone do wejść impulsowych o adresach X0 i X1, a żarówka przyłączona jest do przekaźnika wyjść impulsowych o adresie Y0. Proszę zwrócić uwagę na sposób ułożenia funkcji w programie. Ma to ścisły związek z kolejnością ich wykonywania. Pierwsza funkcja do wykonania pobierana jest z komórki A1. Po wykonaniu tej funkcji sterownik sprawdza dalej czy w pierwszej linii nie znajduje się jeszcze jakaś funkcja i przechodzi do początku drugiej linii na pozycje A2. Po sprawdzeniu pierwszej pozycji w linii drugiej sprawdza kolejne kolumny i przechodzi do następnej linii. Funkcje są pobierane i wykonywane do chwili aż sterownik natknie się na funkcję END która kończy pisany przez nas program technologiczny. Teraz mamy gotowy program i gdy wgramy go do sterownika, będzie on wykonywał wcześniej opisane zadania.

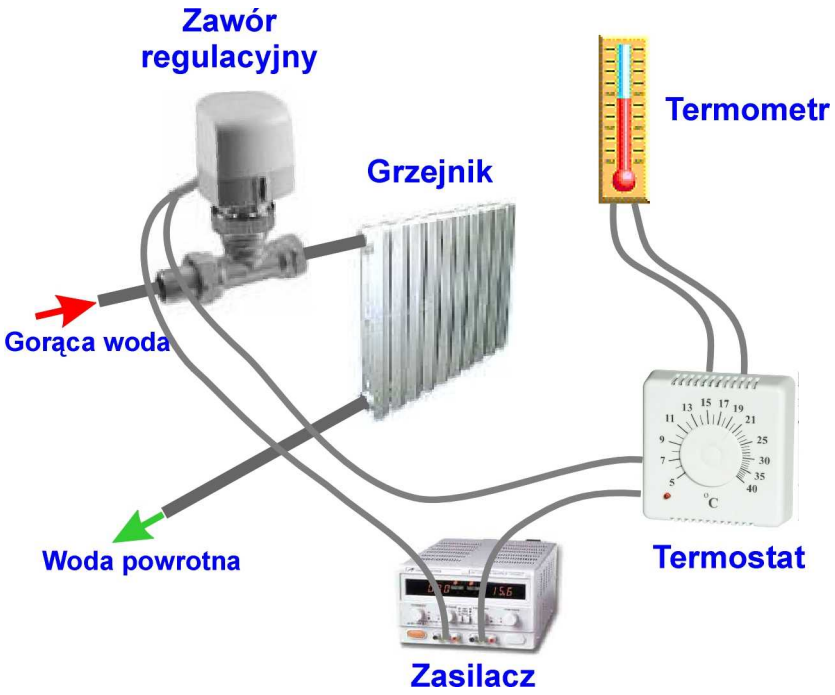
Zobaczmy teraz jak wyglądałby schemat elektryczny podłączeń do samego sterownika:



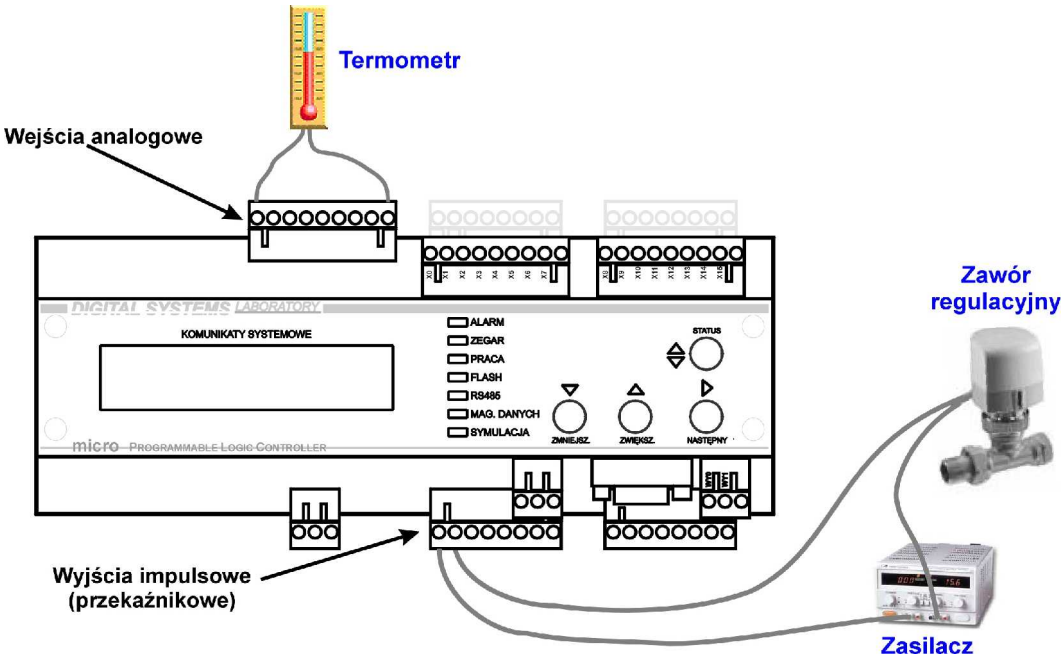
Podłączona bateria jest potrzebna do tego aby zasilić żarówkę. Wyjścia impulsowe nie mogą zasilać bezpośrednio żarówki, ponieważ podpięte są do nich przekaźniki i tylko za ich pośrednictwem możemy sterować urządzeniami zewnętrznymi takimi jak żarówka.

Sterownik poza wejściami i wyjściami impulsowymi, które są tylko odzwierciedleniem zamkniętego lub otwartego styku, czyli przyjmują stan 0 lub 1,posiada również wejścia i wyjścia analogowe, które mogą odczytywać i zapisywać wielkości analogowe o wartościach od 0 do 4095. Wartości te wynikają z budowy urządzenia i określają maksymalną pojemność rejestru w pamięci urządzenia. Służą one do mierzenia sygnałów posiadających konkretną wartość, takich jak np. temperatura, ciśnienie, prędkość. Do wyjść analogowych mogą być podłączone urządzenia które do swojej pracy też potrzebują sygnału sterującego o konkretnej wartości np. natężenie oświetlenia, prędkość obrotowa silnika itd.

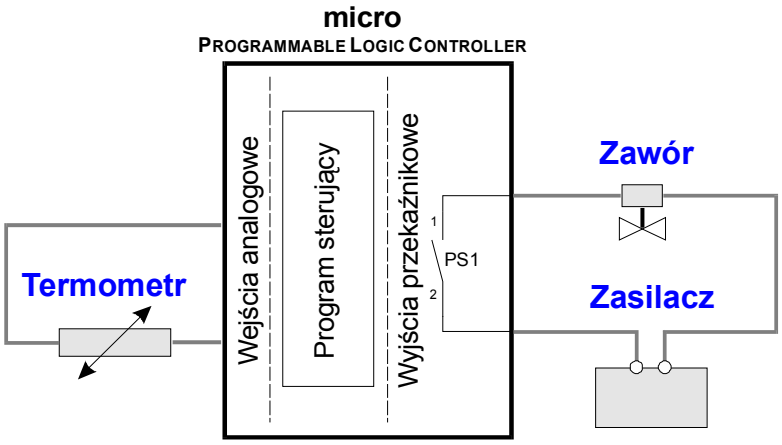
Zajmijmy się teraz nieco bardziej skomplikowanym układem wykorzystującym wejście analogowe. Będzie to regulacja temperatury w pomieszczeniu przy pomocy termostatu z przyłączonym czujnikiem temperatury. Schemat instalacji pokazano poniżej:



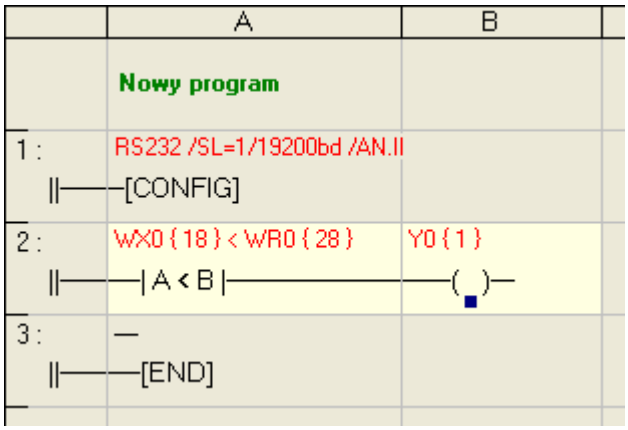
Jeżeli temperatura w pomieszczeniu obniży się do pewnego zadanego poziomu termostat wysyła sygnał do zaworu i otwiera go. W ten sposób gorąca woda może wpłynąć do grzejnika i rozgrzewając go podnieść temperaturę w pomieszczeniu. Jeżeli pomieszczenie dostatecznie się rozgrzeje zawór znowu zostanie zamknięty. Schłodzona woda z grzejnika wypływa w celu powtórnego podgrzania.
W pokazanym przykładzie wykorzystujemy dwa typy sygnałów. Pierwszy to analogowy pomiar temperatury który musi być przekazany do termostatu. Wartość tego sygnału waha się w zakresie od 0 do 100 °C. Kolejnym analogowym sygnałem jest pozycja w której ustawione jest pokrętło termostatu nastawiające temperaturę zadaną pomieszczenia. Wartość zadaną zmienić można w zakresie od 0 do 40 °C. I ostatni dobrze znany sygnał wyjścia impulsowego. Do układu musimy podłączyć zasilacz który zasili zawór sterujący.
Zobaczmy teraz jak wyglądałby schemat elektryczny podłączeń do samego sterownika:



Jak widać czujnik temperatury przyłączony jest do nowej listwy zaciskowej. Jest to listwa wejść analogowych. Pozwala ona na podłączenie do ośmiu czujników temperatury. Pokazany czujnik przyłączona do pierwszego wejścia o adresie WX0. Zawór i zasilacz jak poprzednio przyłączono do pierwszego wyjścia przekaźnikowego o adresie Y0. Zobaczmy jak wygląda schemat elektryczny tego układu:



Układ w stosunku do poprzednich aplikacji uległ niewielkiej korekcie. Zawór jest podłączony jak poprzednio do pierwszego przekaźnika wyjściowego. Termometr do wejścia analogowego. Wejścia impulsowe są nie podłączone. Z układu połączeń wynika że brakuje nam w układzie termostatu. Termostatem będzie sterownik ale najpierw trzeba napisać program który będzie realizował tą funkcję.
Zobaczmy jak powinien wyglądać program aby zaistniała regulacja temperatury.



Aby sterownik pełnił funkcję termostatu wystarczy napisać jedną linię programu. W linii 2 pierwsza funkcja sprawdza czy wartość zadana czyli „pokrętło termostatu" jest ustawione na temperaturę większą, niż odczytywana z czujnika temperatury. Na przykładzie czujnik temperatury pokazuje {18} °C a a termostat ustawiony jest na temperaturę { 28 } °C. Jeżeli temperatura zadana jest wyższa niż zmierzona nastąpi załączenie wyjścia Y0 które podłączone jest do zaworu odcinającego dopływ wody. Jeżeli temperatura wzrośnie powyżej 28 °C wyjście zostanie wyłączone i zawór zamknie się. Czy to nie jest proste ?.

Zasady programowanie drabinkowego.

- Program drabinkowy zawsze rozpoczynamy od funkcji ||---[CONFIG] w której następuje konfiguracja podstawowych parametrów sterownika.
- Program drabinkowy zawsze kończy funkcja ||---[END], po wykonaniu której sterownik przechodzi do zapisu wyjść sterownika, a następnie ponownie do początku programu.
- Sterownik odczytuje program z każdej linii od lewej do prawej strony. Po odczytaniu ostatniej funkcji w linii następuje odczyt kolejnej funkcji następnej linii
- Nie ma znaczenia kolejność obsługiwanych sygnałów wejść i wyjść.
- W programie można wielokrotnie zmieniać np. stan jednego wyjścia impulsowego lecz obowiązywać będzie tylko stan ostatnio zdefiniowany najbliżej końca programu.

Teraz prosimy zapoznać się z wszystkimi dostępnymi funkcjami sterownika.

2.FUNKCJE PODSTAWOWE

||--| |-- Rozpoczęcie operacji logicznej - proste

	||---| |--- kod 128 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	128	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

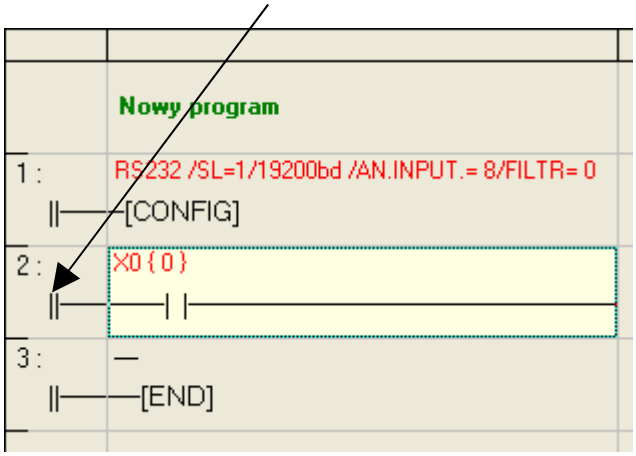
Rozkazem tym rozpoczynamy każdą linię w której będą występowały operacje logiczne. Funkcja ta symbolizuje styk który w chwili gdy jest rozwarty nie przewodzi prądu. Symbol ||-- oznacza początek gałęzi która dostarcza zasilania do pracy programu. W chwili zwarcia następuje przewodzenie prądu od strony lewej do prawej, to jest od początku gałęzi do dalszej części gałęzi oznaczonej „---„. Zapis „-----„ symbolizuje przewód elektryczny przez który następuje przekazanie sygnału do następnej funkcji. Załączenie tego styku może odbywać się za pomocą sygnału pochodzącego np. z wejścia impulsowego sterownika. W programie zapisany on będzie w następujący sposób:

X0
||--| |--- Styk otwarty

X0
||--|/| |--- Styk zamknięty

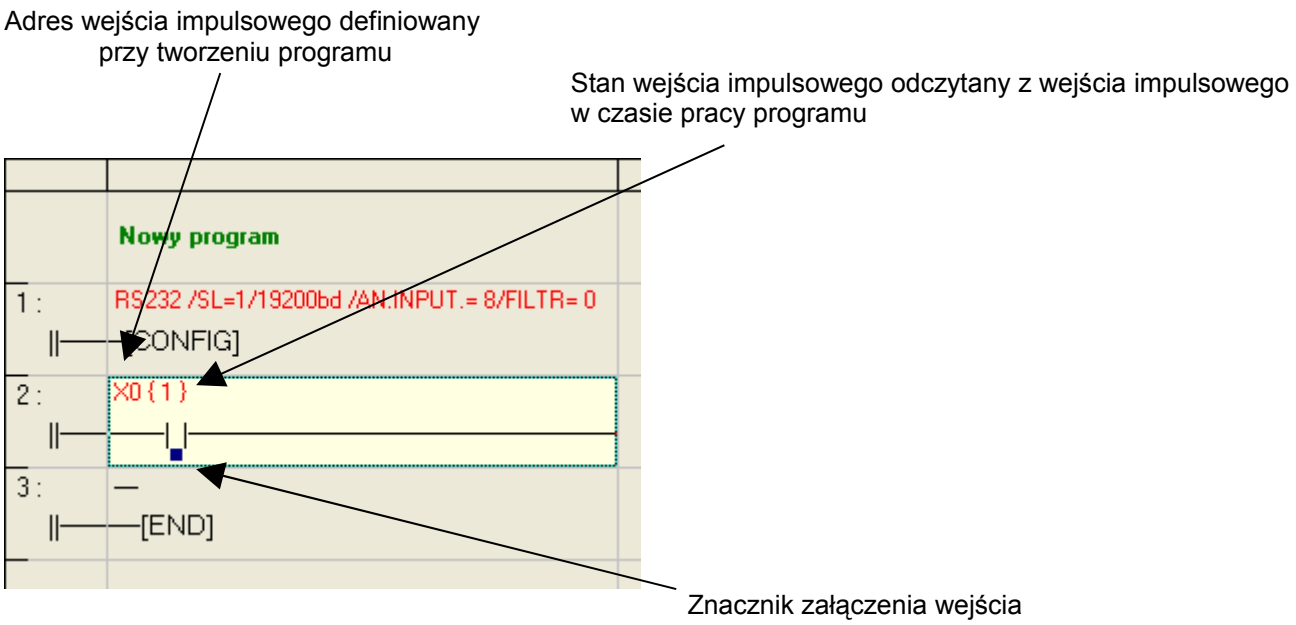
Styk otwarty symbolizuje znak pokazany wyżej, a styk zamknięty jest wtedy jeżeli poniżej pionowych kresek pojawi się mały kwadracik. Taki zapis oznacza że sygnał do pracy styku będzie pobrany z wejścia impulsowego o adresie X0. Oznacza to ze z pierwszego fizycznego wejścia sterownika będzie pobierany stan do pracy programu. Adresowanie wszystkich wejść zaczyna się od zera. W praktyce oznacza to że podanie napięcia +12V na 1 wejście impulsowe sterownika spowoduje przekazanie stanu =1 na prawą stronę naszej funkcji ---. Oto jak funkcja ta będzie przedstawiona w edytorze dołączonym do sterownika. Na ekranie widzimy okno w którym pokazane będą trzy linie.

Znak rozpoczęcia linii programu („podłączenie do + zasilania” , logiczne 1)

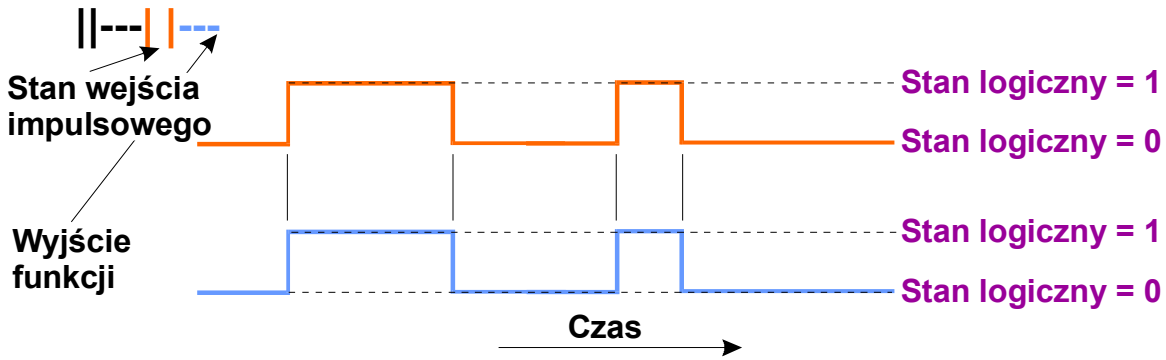


Pierwsza zawiera instrukcję która zawsze rozpoczyna każdy program i służy do konfiguracji sterownika. Druga lina jest naszą pierwszą funkcją. Trzecia linia występuje zawsze na końcu programu. Brak tej linii uniemożliwi zapis programu do sterownika. Każda funkcja rozpoczynająca linia programu zaczyna się od znaku ||-- jeżeli przy funkcji nie ma takiego znaku oznacza że funkcją nie można rozpoczynać linii.

W linii 2 umieszczono funkcje rozpoczęcia operacji logicznej która w tym momencie nie przekazuje sygnału z lewej do prawej części gałęzi.



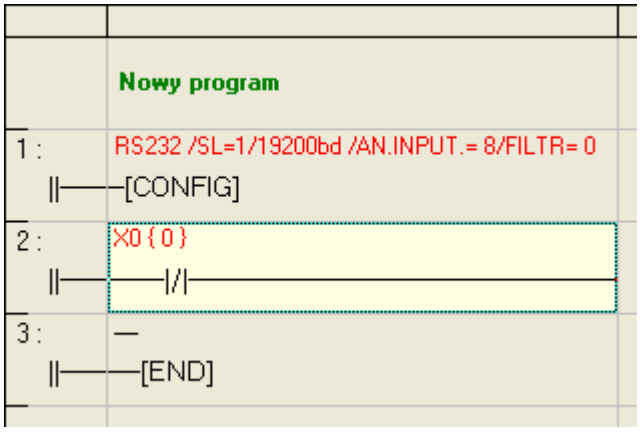
Teraz do wejścia impulsowego sterownika doprowadzono zasilanie na wskutek czego jego stan zmienił się z 0 na 1. Stan ten przedstawiono w postaci małego niebieskiego kwadracika pod funkcją --| |--. Dodatkowo jest on również uwidoczniony w nawiasach klamrowych obok adresu wejścia impulsowego X0 { 1 }. Pisząc program podajemy tylko adres wejścia impulsowego a stan pokazany w nawiasach { } jest odczytywany bezpośrednio z wejścia impulsowego.
Oto schemat działania funkcji:



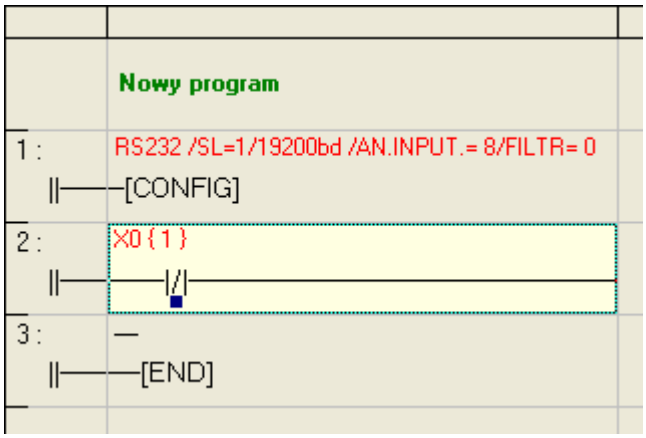
||--|/|--- Rozpoczęcie operacji logicznej - zanegowane

	||---|/|--- kod 129 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	129	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

Rozkazem tym jak poprzednio rozpoczynamy każdą linię w której będą występowały operacje logiczne. Funkcja ta jest identyczna z poprzednią z tą tylko różnicą że „przewodzi" wtedy gdy stan np. wejścia impulsowego na które wskazuje funkcja jest = 0. Oznacza to że sygnałem do złączania naszego umownego styku jest stan niski (=0). Stanem wysokim nazywamy stan logiczny =1.

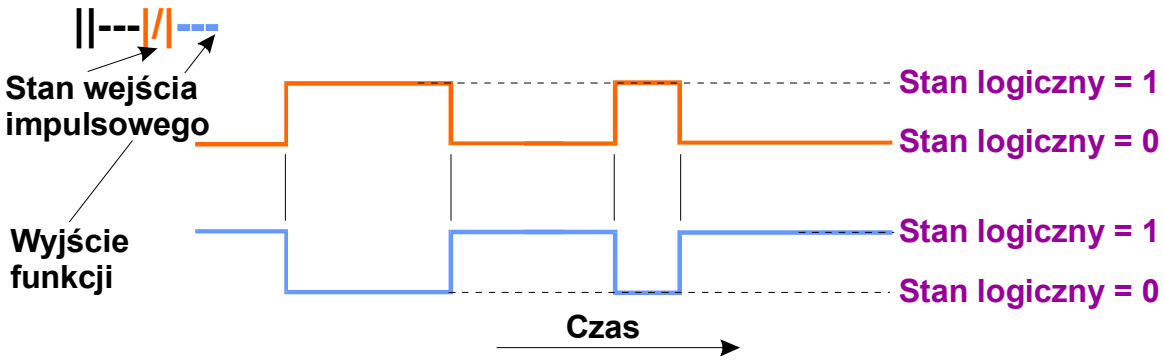


Na powyższym rysunku funkcja przy braku sygnału styk w linii 2 przewodzi sygnał z lewej części na prawą.



W chwili pojawienia się sygnału wyjście z funkcji ma wartość logiczną 0.

Oto schemat działania funkcji:

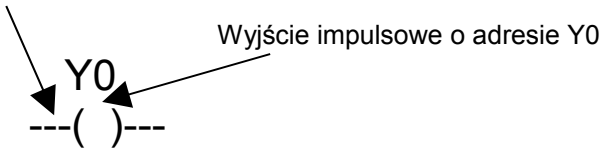


---( )--- Przekaźnik wyjściowy

	---( )--- kod 138 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	138	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

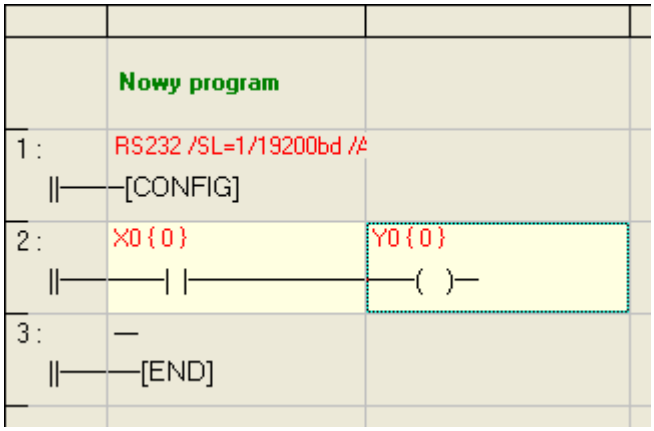
Rozkazem tym kończymy linię w której występowały operacje logiczne. Funkcja ta imituje pracę przekaźnika. Jeżeli z lewej strony dochodzi sygnał to przekaźnik się załącza. Jeżeli sygnał na wejściu przekaźnika ma stan = 0, przekaźnik jest wyłączony. W programie będzie on zapisany w następujący sposób:

Wejście do funkcji przekaźnika

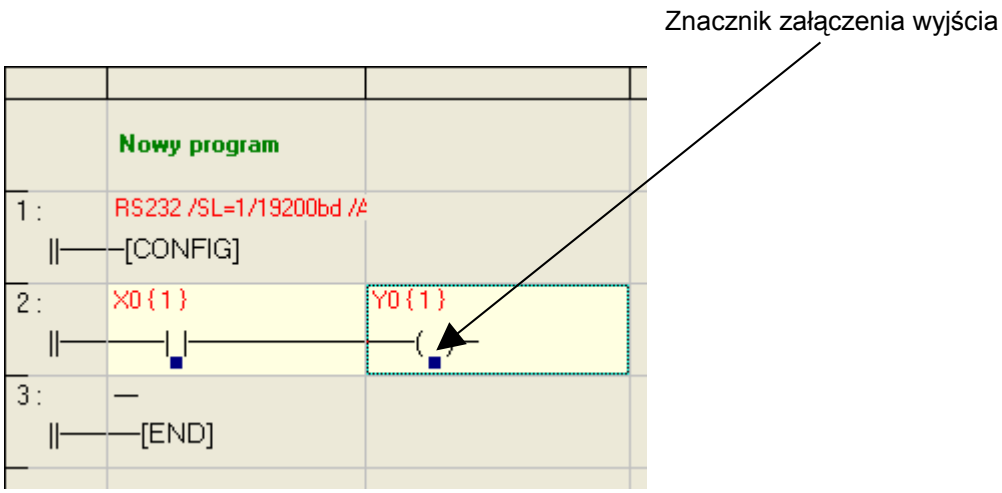


Taki zapis oznacza że sygnał który pojawił się na wejściu funkcji przekaźnika wyjściowego zostanie wysłany do wyjścia impulsowego o adresie Y0. W praktyce oznacza to że jeżeli wynik poprzedniej funkcji umieszczonej w gałęzi będzie = 1 to pierwszy przekaźnik sterownika zostanie załączony.

Poniżej pokazano przykład prostego programu:

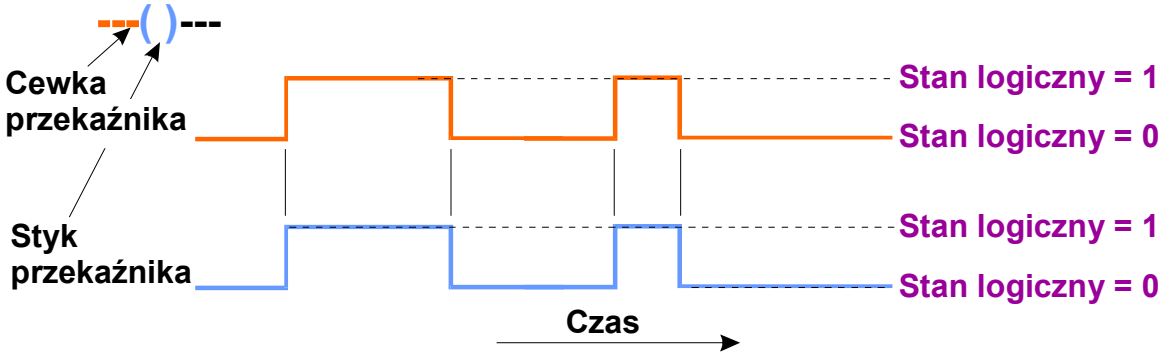


Program odczytuje stan z wejścia impulsowego o adresie X0 i wysyła go do przekaźnika wyjściowego o adresie Y0. Krótko mówiąc jeśli wejście zostanie załączone wyjście tez zostanie załączone. Powyższy rysunek pokazuje stan kiedy na wejściu impulsowym jest stan niski co nie powoduje wysterowania przekaźnika wyjściowego.



Teraz na wejściu impulsowym pojawił się sygnał co powoduje załączenie przekaźnika wyjściowego.

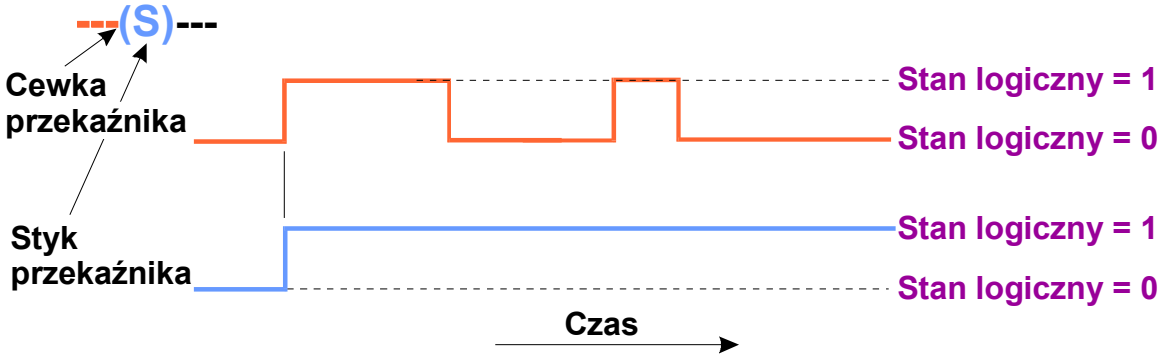
Oto schemat działania funkcji:



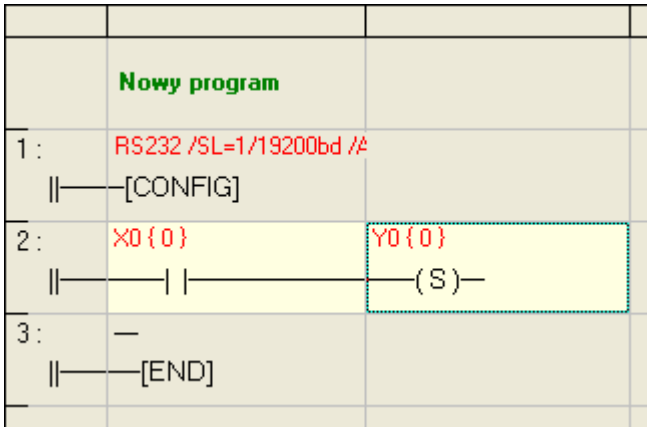
---(S)--- Przekaźnik wyjściowy załączany stanem wysokim - SET

	---(S)--- kod 139 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	139	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

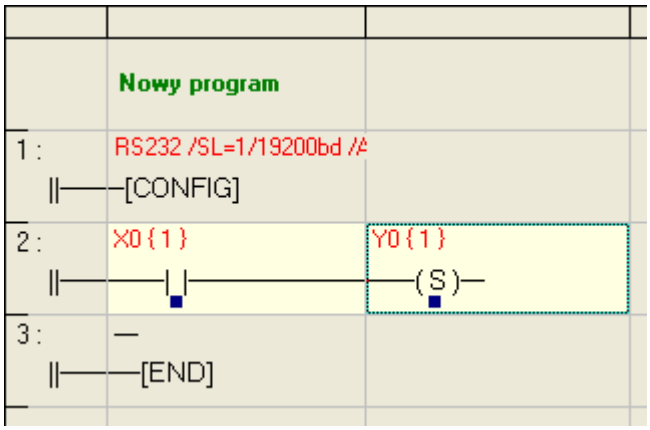
Rozkazem tym kończymy linię w której występowały operacje logiczne. Funkcja ta imituje pracę przekaźnika który pod wpływem zasilania załącza się i po odłączeniu go od zasilania jego styki nadal pozostają w stanie załączenia. Styki przekaźnika pozostają dotąd zwarte do czasu kiedy inna funkcja nie zmodyfikuje jego stanu. Z tego wynika że krótki impuls może załączyć taki przekaźnik. Wyłączenie go jest tylko wtedy możliwe jeżeli po impulsie załączającym dochodzącym do lewej strony funkcji następuje stan niski. Jeżeli na dojściu do przekaźnika mamy stan =0 to funkcja nie zmienia stanu wyjścia i omija go. Innymi słowy można powiedzieć że funkcja ta jest rodzajem działania warunkowego jeśli na wejściu do funkcji pojawia się 1 to przekaźnik jest załączany jeśli pojawia się 0 wykonywanie funkcji przekaźnika jest ignorowane.
Oto schemat działania funkcji:



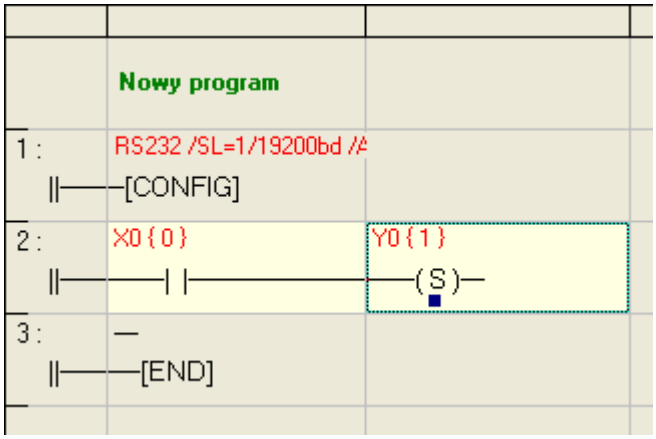
Oto fragment programu z funkcją przekaźnika załączanego stanem wysokim:



W linii drugiej użyto funkcji przekaźnika załączanego zboczem narastającym. Zbocze narastające jest to zmiana stanu z niskiego na wysoki. Pokazany stan jest stanem występującym bezpośrednio po resecie sterownika. Stan wyjścia jest =0 i przekaźnik wyjściowy jest wyłączony.



Teraz na wejściu impulsowym o adresie X0 pojawił się stan wysoki co spowodowało załączenie się pierwszego przekaźnika wyjściowego o adresie Y0.

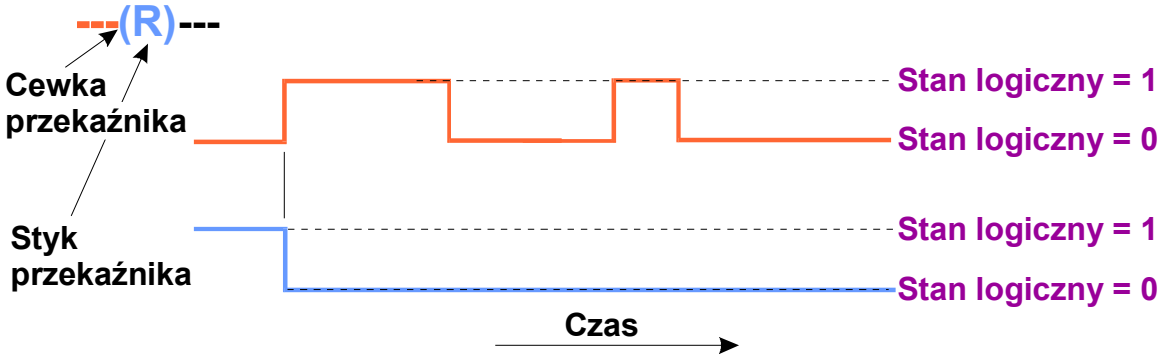


Po zaniku na wejściu impulsowym X0 sygnału wyjście impulsowe Y0 nadal pozostaje załączone. W tym programie stan taki utrzyma się do następnego resetu urządzenia np. przez wyłączenie zasilania sterownika lub modyfikacji przez funkcję –-(R)--- opisną poniżej

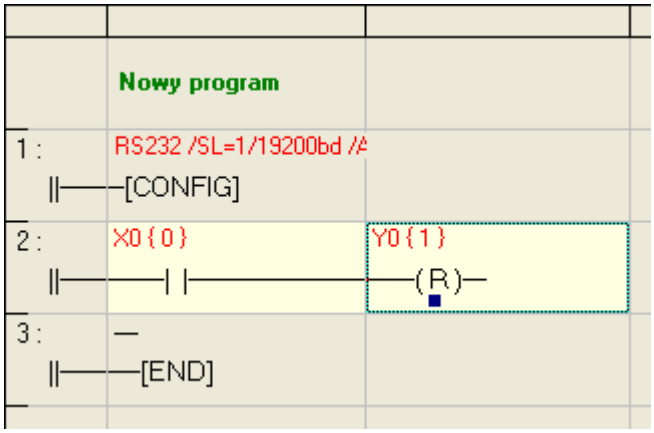
---(R)--- Przekaźnik wyjściowy wyłączany stanem wysokim - RESET

	---(R)--- kod 140 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	140	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

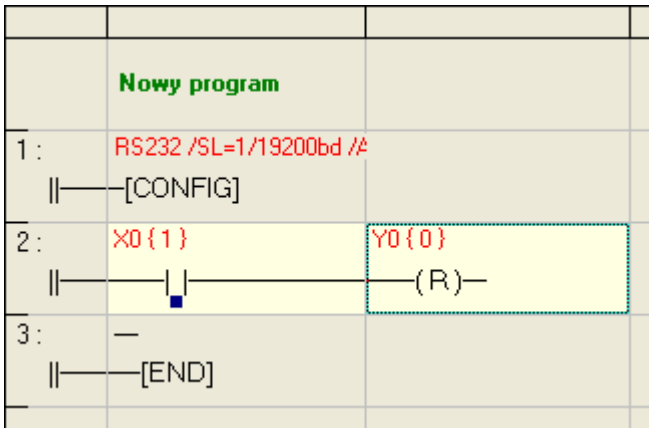
Rozkazem tym kończymy linię w której występowały operacje logiczne. Funkcja ma podobne działanie jak poprzednia funkcja z tą różnicą że stan wysoki sygnału dochodzącego do funkcji wyłącza przekaźnik jeżeli jest załączony. Styki przekaźnika pozostają rozwarte do czasu kiedy inna funkcja nie zmodyfikuje jego stanu. Funkcja ta występuje zwykle w parze z opisaną poprzednio funkcją –-(S)---, i powoduje wyłączenie konkretnego wyjścia lub zmianę stanu przekaźnika wewnętrznego.
Oto schemat działania funkcji:



Oto fragment programu z funkcją przekaźnika wyłączanego stanem wysokim:



Teraz pokazano przykład w którym ręcznie załączono przekaźnik wyjściowy o adresie Y0.



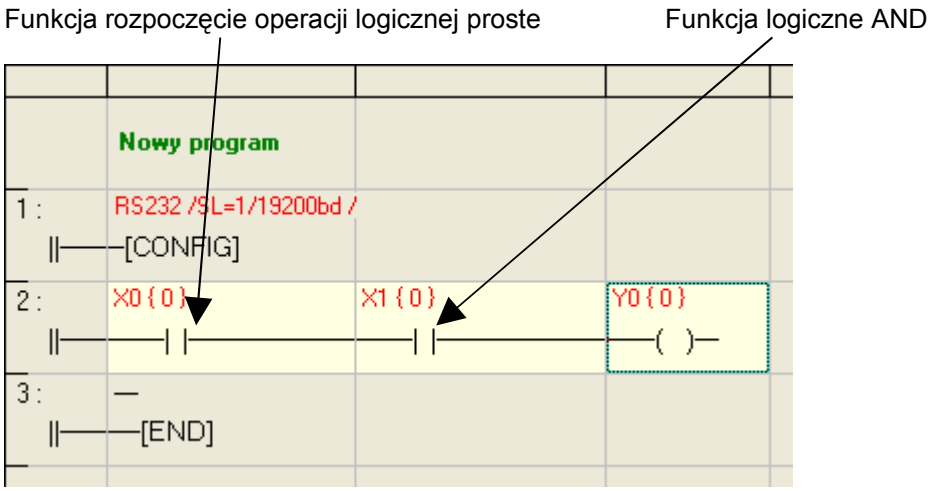
Po wysterowaniu wejścia impulsowego o adresie X0 następuje natychmiastowe wyłączenia przekaźnika Y0. W tym programie stan taki utrzyma się do czasu kiedy ktoś z klawiatury sterownika nie załączy pierwszego wyjścia impulsowego.

---| |--- Szeregowe połączenie styków – logiczne AND

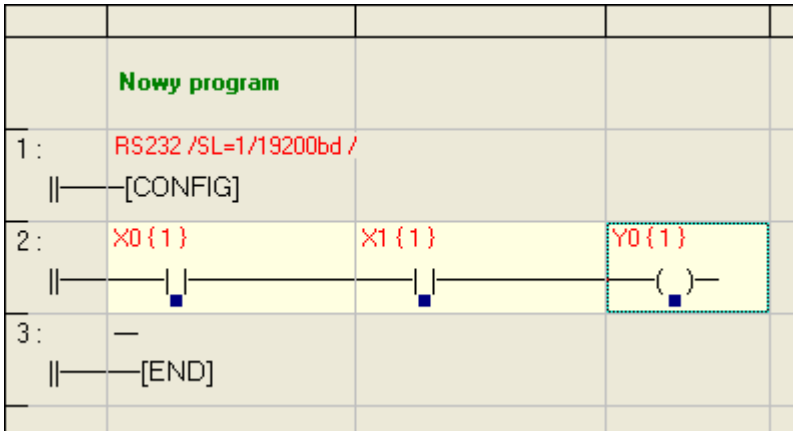
	---| |--- kod 131 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	131	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

Rozkazem tym nigdy nie rozpoczynamy linii w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna. Funkcja ta jest bardzo podobna do funkcji „rozpoczęcie operacji logicznej proste” oznaczonej jako ||--- | |--- która rozpoczyna nową linię. Nasza funkcja jest oznaczona jako ---| |--- i nie służy do rozpoczynania linii programu. Funkcja realizuje logiczną funkcję AND. Jeżeli jej „styk” jest zwarty to stan z wejścia przekazany jest na wyjście. Połączenie np. dwóch takich funkcji szeregowo powoduje powstanie warunku logicznego AND w którym żeby sygnał został przekazany z wejścia gałęzi na jej wyjście styk pierwszy i drugi i trzeci muszą być zwarte.

Oto fragment programu realizujący warunek AND:



Jeżeli wejście impulsowe o adresie X0 i wejście impulsowe o adresie X1 będzie miało stan niski to wyjście przekaźnikowe Y0 będzie wyłączone.



Aby wyjście przekaźnikowe Y0 zostało załączone to obydwa wejścia impulsowe X0 i X1 muszą mieć stan 1.

Uwaga: W związku z tym że funkcją ---| |--- nie możemy rozpoczynać nowej linii w przykładzie jako pierwsza umieszczona jest funkcja „Rozpoczęcie operacji logicznej proste ||---| |--- ” a dopiero następna funkcja logiczna AND.

Pracę funkcji można przedstawić w tabeli stanu:

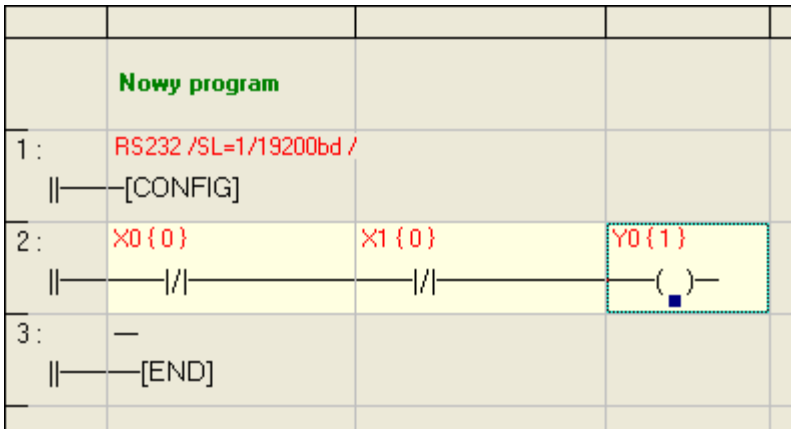
Stan na wejściu do funkcji	Stan na wyjściu funkcji	Uwagi
1	1	Jeśli funkcja odczytuje stan=1
0	0	Jeśli funkcja odczytuje stan=0

---|/|--- Szeregowe połączenie styków zanegowane – logiczne NAND

	---|/|--- kod 132 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	132	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

Rozkazem tym nigdy nie rozpoczynamy linii w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna. Funkcja realizuje logiczną funkcję NAND. Jeżeli jej „styk" jest rozwarty to stan z wejścia przekazany jest na wyjście. Połączenie np. dwóch takich funkcji szeregowo powoduje powstanie warunku logicznego NAND w którym żeby sygnał został przekazany z wejścia gałęzi na jej wyjście styk pierwszy i drugi i trzeci muszą być rozwarte.

Oto fragment programu realizujący warunek NAND:



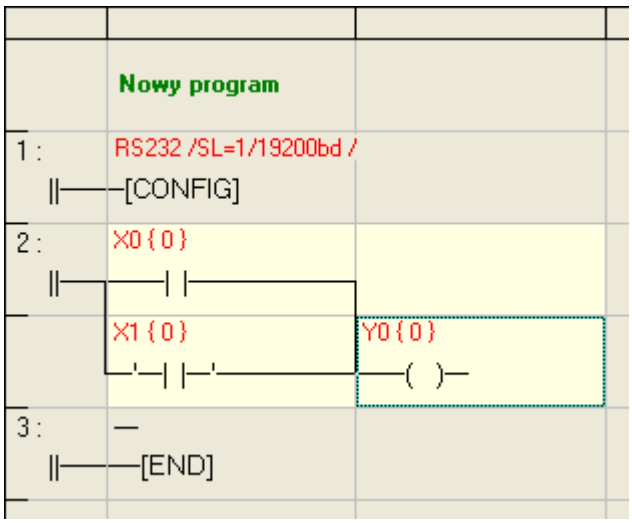
Jeżeli stany obydwu wejść impulsowych X0 i X1 mają stan niski to wyjście przekaźnikowe Y0 będzie wysterowane.
Uwaga w przykładzie jako pierwsza funkcja umieszczona jest funkcja „Rozpoczęcie operacji logicznej zanegowane" a dopiero następna funkcja logiczna NAND.
Pracę funkcji można przedstawić w tabeli stanu:

Stan na wejściu do funkcji	Stan na wyjściu funkcji	Uwagi
1	1	Jeśli funkcja odczytuje stan=0
1	0	Jeśli funkcja odczytuje stan=1

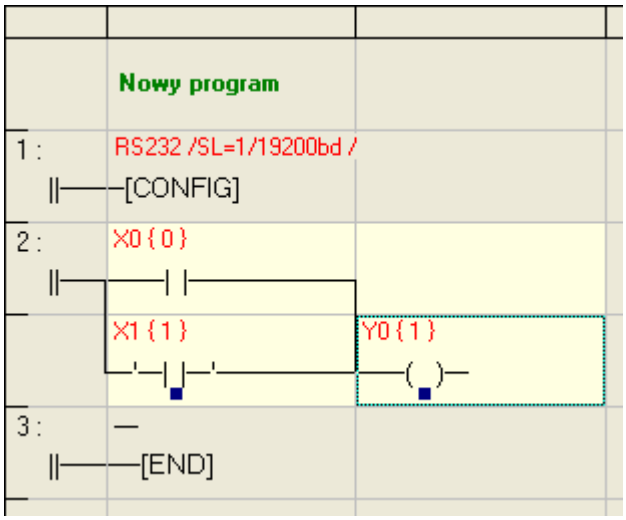
‘--| |--‘ Równoległe połączenie styków – logiczne OR

	‘--| |--‘ kod 133 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	133	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

Rozkazem tym nigdy nie rozpoczynamy linii w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna. Funkcja realizuje logiczną funkcję OR. Jeżeli jej „styk” jest zwarty to stan z wejścia przekazany jest na wyjście. Połączenie równoległe kilku takich funkcji realizuje sumę logiczną.
Oto fragment programu realizujący warunek OR:



Jeżeli na obydwóch wejściach impulsowych X0 i X1 jest stan niski to na wyjściu o adresie Y0 jest stan niski.



Jeżeli przynajmniej na jednym z wejść impulsowych X0 lub X1 pojawi się stan logiczny wysoki wyjście o adresie Y0 zostanie złączone

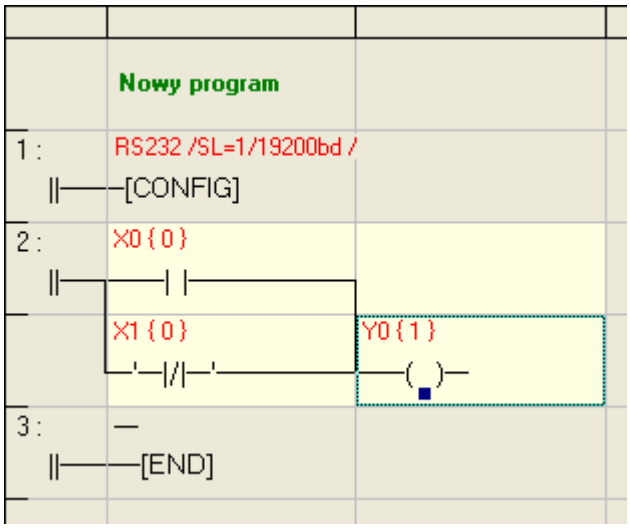
Pracę funkcji można przedstawić w tabeli stanu:

Stan na wejściu do funkcji	Stan na wyjściu funkcji	Uwagi
1	1	Jeśli funkcja odczytuje stan=1
1	0	Jeśli funkcja odczytuje stan=0

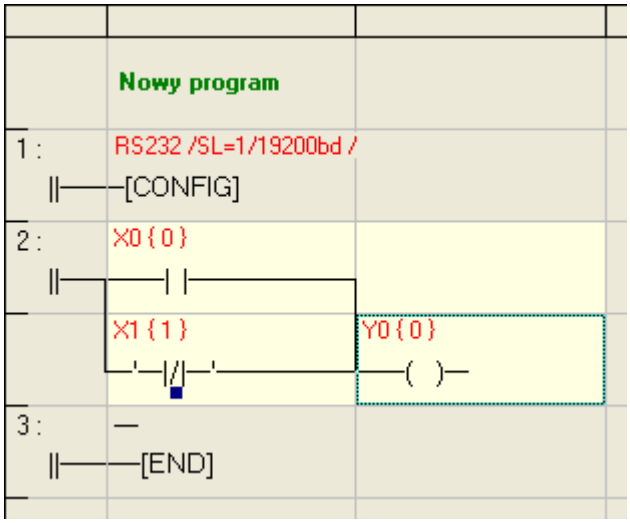
‘--|/|--‘ Równoległe połączenie styków zanegowane – logiczne NOR

	‘--|/|--‘ kod 134 , 4		
1	Długość funkcji w bajtach	4	
2	Kod funkcji	134	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru

Rozkazem tym nigdy nie rozpoczynamy linii w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna. Funkcja realizuje logiczną funkcję NOR. Jeżeli jej „styk” jest rozwarty to stan z wejścia przekazany jest na wyjście. Połączenie równoległe kilku takich funkcji realizuje zanegowaną sumę logiczną.
Oto fragment programu realizujący warunek NOR:



Na przykładzie linię rozpoczęto funkcją „Rozpoczęcie operacji logicznej proste” jako drugą użyto NOR. W efekcie przy nie wysterowanych wejściach impulsowych X0 i X1otrzymano na wyjściu Y0 stan wysoki, ponieważ stan X1 jest =0 a po zanegowaniu jest =1 co na wyjściu tej funkcji daje stan wysoki.



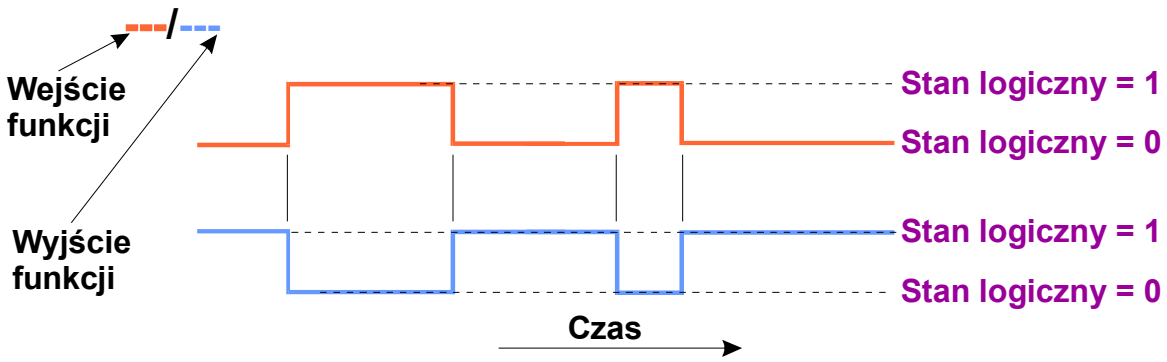
W chwili gdy wejście impulsowe =1 na wyjściu funkcji NOR pojawia się stan =0. Z funkcji umieszczonej na początku linii 2 nie otrzymujemy stanu wysokiego co w efekcie nie ustawia wyjścia przekaźnikowego Y0. Pracę funkcji można przedstawić w tabeli stanu:

Stan na wejściu do funkcji	Stan na wyjściu funkcji	Uwagi
1	1	Jeśli funkcja odczytuje stan=0
1	0	Jeśli funkcja odczytuje stan=1

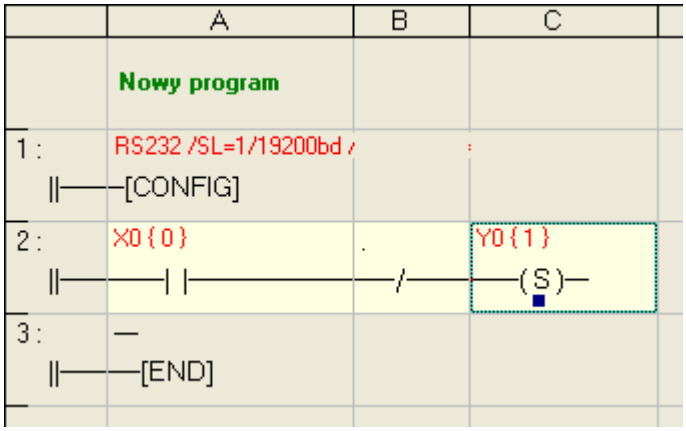
---/--- Negacja stanu gałęzi

	---/--- kod 135 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	135	
3	Nieużywane	0	

Rozkazem tym nigdy nie rozpoczynamy linii w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna funkcja gałęzi. Realizuje logiczną funkcję NOT. Funkcja neguje stan który pojawia się na wejściu i przesyła na prawą stronę funkcji. Funkcja nie korzysta z żadnych dodatkowych parametrów. Działanie funkcji można pokazać przy pomocy przebiegów:



Na wyjściu funkcji pojawia się negacja wejścia.
Poniżej zamieszczono przykładowy program wykorzystujący negację stanu gałęzi:

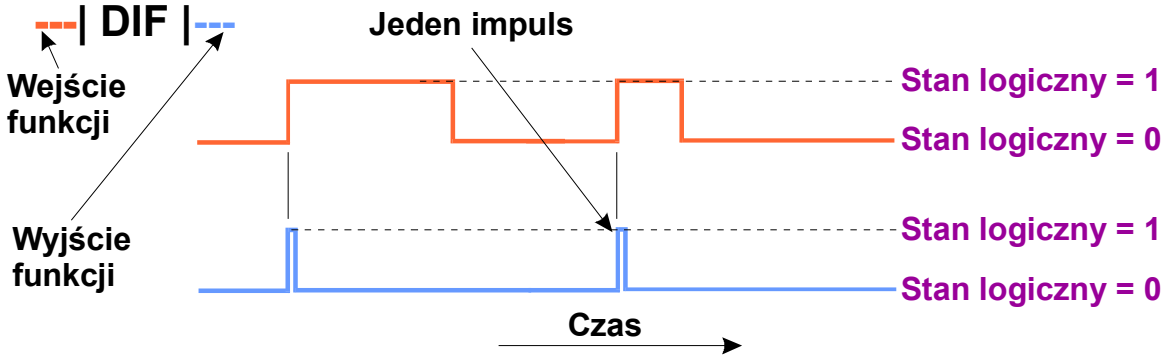


Powyżej pokazano modyfikację programu zamieszczonego wcześniej przy opisie przekaźnika załączanego stanem wysokim z tą różnicą że teraz przekaźnik Y0 załączany jest niskim stanem wejścia impulsowego X0.

---|DIF|--- Wyszukanie zbocza narastającego

	---|DIF|--- kod 136 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	136	
3	Nr. funkcji DIF adr. 0-63	0-63	Obszar wspólny dla DIF i DFN

Rozkazem tym nigdy nie rozpoczynamy linii w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna funkcja gałęzi. Zbocze narastające czyli zmiana stanu z niskiego na wysoki na wejściu do funkcji wytwarza na jej wyjściu pik trwający przez jedną pętlę programu technologicznego sterownika. Przy następnym przebiegu programu wyjście z funkcji ma wartość = 0. Taki stan utrzymuje się przez cały czas kiedy do funkcji nie dochodzi zmiana stanu logicznego. Uzyskanie kolejnego zbocza nastąpi po ponownej zmianie stanu z niskiego na wysoki. Funkcja do swojej pracy wykorzystuje dodatkowy parametr jakim jest jej numer kolejny. Funkcje ---| DIF |--- można numerować od 0 – 255. Należy dbać oto aby numeracja funkcji ---| DIF |--- i ---| DFN |--- się nie powtarzała. Powtórzenie tego samego numeru funkcji w programie spowoduje nieprawidłowe jego działanie.

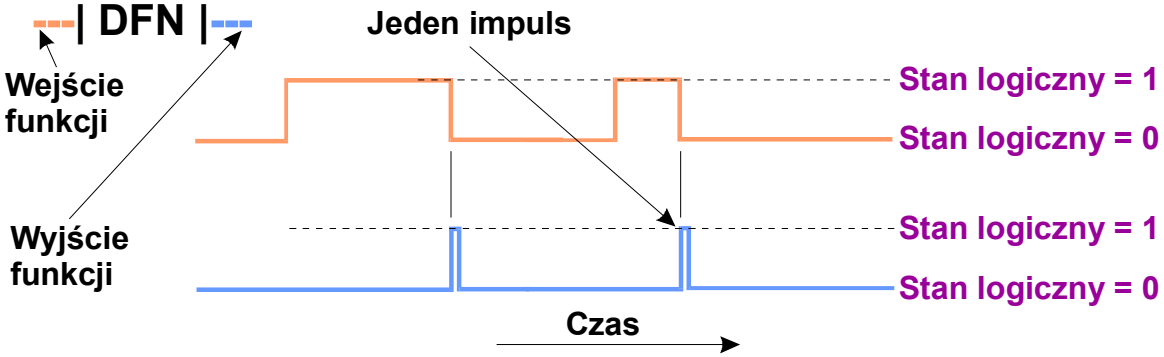


Impuls generowany jest tylko przy zmianie stanu na wejściu do funkcji z niskiego na wysoki.

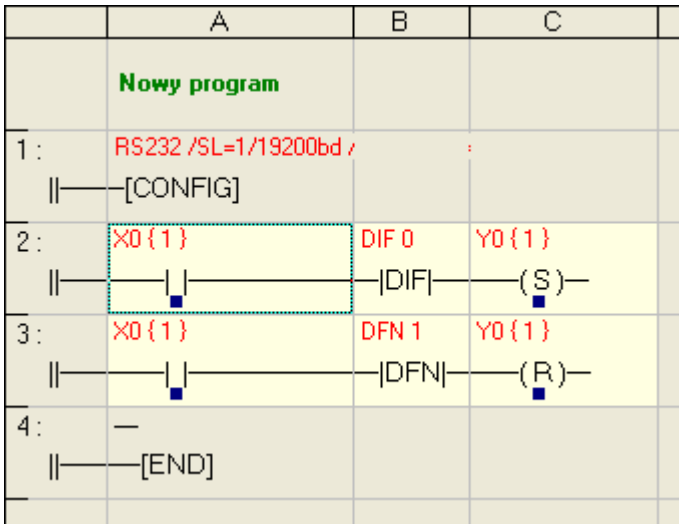
---|DFN|--- Wyszukanie zbocza opadającego

	---|DFN|--- kod 137 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	137	
3	Nr. funkcji DIF adr. 0-63	0-63	Obszar wspólny dla DIF i DFN

Rozkazem tym nigdy nie rozpoczynamy linii w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna funkcja gałęzi. Zbocze opadające czyli zmiana stanu z wysokiego na niski na wejściu do funkcji wytwarza na jej wyjściu pik trwający przez jedną pętlę programu technologicznego sterownika. Przy następnym przebiegu programu wyjście z funkcji bez względu na stan na wejściu ma wartość = 0. Taki stan utrzymuje się przez cały czas kiedy do funkcji nie dochodzi zmiana stanu logicznego. Uzyskanie kolejnego zbocza nastąpi po ponownej zmianie stanu z wysokiego na niski. Funkcja do swojej pracy wykorzystuje dodatkowy parametr jakim jest jej numer kolejny. Funkcje ---| DFN |--- można numerować od 0 – 255. Należy dbać oto aby numeracja funkcji ---| DIF |--- i ---| DFN |--- się nie powtarzała. Powtórzenie tego samego numeru funkcji w programie spowoduje nieprawidłowe jego działanie.



Impuls generowany jest tylko przy zmianie stanu na wejściu do funkcji z wysokiego na niski.

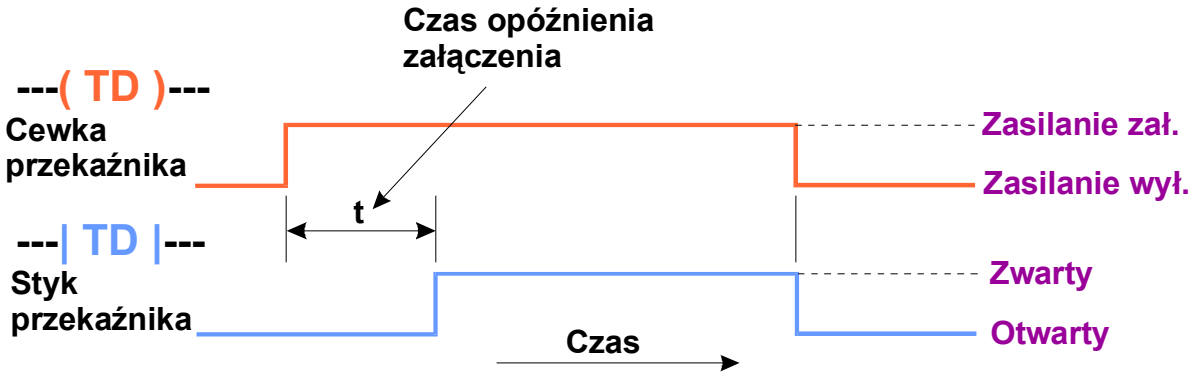


Powyżej pokazano przykład użycia funkcji DIF i DFN do sterowania wyjściem impulsowym Y0 tylko przy pomocy zboczy. Jeżeli na wejście impulsowe X0 podamy stan wysoki nastąpi załączenie wyjścia a jeżeli stan zmieni się na niski nastąpi jego wyłączenie. Należy zwrócić uwagę aby numery funkcji wykrywające zbocza nie powtarzały się. W czasie w którym nie ma zmian stanu wejścia impulsowego X0 można swobodnie manipulować stanem wyjścia impulsowego Y0.

---(TD)--- Przekaźnik czasowy – opóźnione załączenie (Time Delay)

	---(TD)--- kod 150 , 7		
1	Długość funkcji w bajtach	7	
2	Kod funkcji	150	
3	Numer zegara adr 0-63	0-63	Obszar wspólny dla TD i SS
4	Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
5	Nazwa rejestru zależna od wyboru z linii 4	XX	Adres użytego rejestru
6	Nazwa rejestru zależna od wyboru z linii 4	XX	Adres użytego rejestru
7	Rejestr taktujący R adr. 0 - 255	0-255	Adres użytego rejestru

Rozkazem tym kończymy linę w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja imituje pracę przekaźnika czasowego z opóźnionym załączeniem. Jeżeli cewka przekaźnika czasowego podłączona jest do zasilania to jego styki nie zewrą się od razu lecz po pewnym czasie. Dlatego przekaźnik nazywa się czasowym. Oczywiście czas po jakim nastąpi zwarcie styków jest regulowany w zakresie od sekund do godzin. Wyjaśnienie jak pracuje przekaźnik czasowy najlepiej pokazać w następujący sposób:

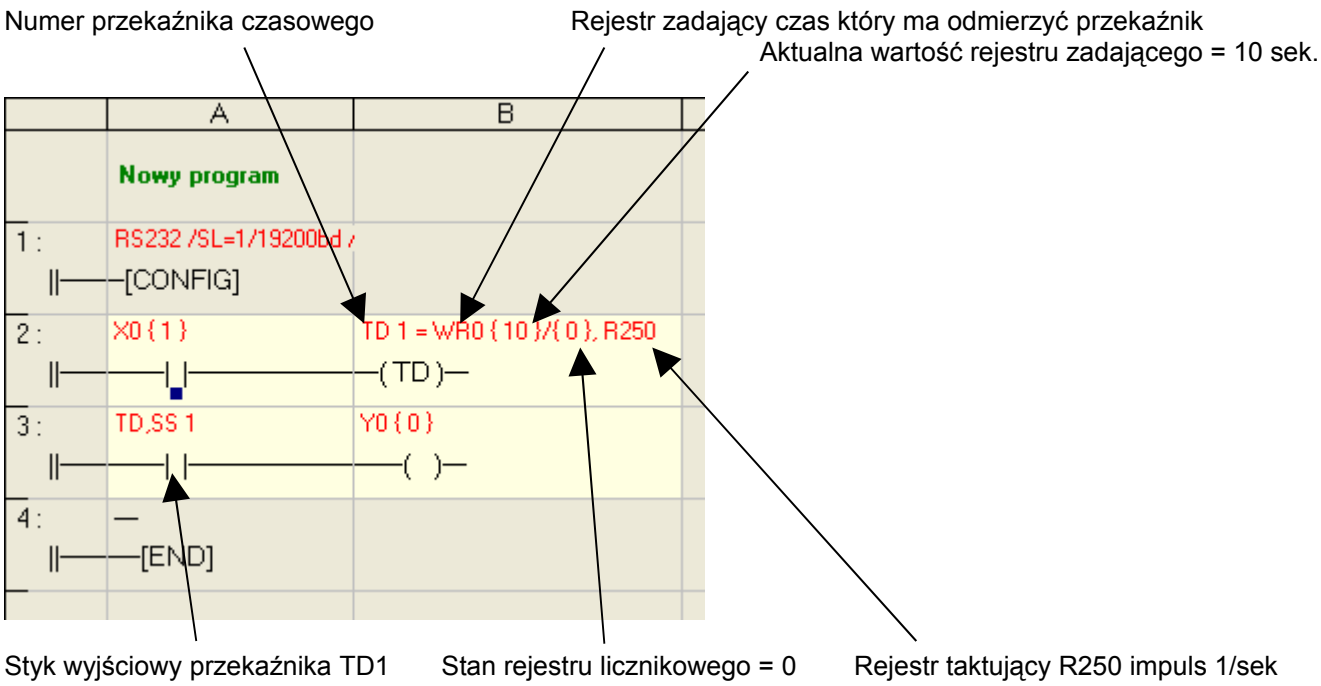


Funkcja ta do swojej pracy wykorzystuje trzy parametry. Pierwszy parametr jest numerem przekaźnika czasowego w zakresie od 0 do 63. Numery przekaźników TD i SS nie mogą się powtarzać. Drugim parametrem jest rejestr WR który określa po jakim czasie przekaźnik ma się załączyć. Trzeci parametr to impuls taktujący zegara. W zależności od tego jaki czas ma odmierzać przekaźnik tak dobrany jest drugi parametr. Parametr ten to przekaźnik wewnętrzny sterownika który działa podobnie do przekaźników wyjść impulsowych lecz nie ma do niego przyłączonego fizycznego przekaźnika. Jest to tylko rodzaj pamięci wewnętrznej. Przekaźniki wewnętrzne nazwane literką R (skrót od Relay) a ich ilość wynosi 256 o adresach od R0 – R255. Większość przekaźników można wykorzystywać do dowolnych celów. Jest jednak grupa przekaźników która ma z góry narzuconą funkcję. My będziemy wykorzystywać przekaźniki o adresach od R247 do R250. Każdy z nich może być użyty do generowania impulsu czasowego. Oto zestawienie i funkcja wykorzystywanych przez nas rejestrów:

R247 - Przekaźnik wewnętrzny generujący jeden impuls na dobę
R248 - Przekaźnik wewnętrzny generujący jeden impuls na godzinę
R249 - Przekaźnik wewnętrzny generujący jeden impuls na minutę
R250 - Przekaźnik wewnętrzny generujący jeden impuls na sekundę

Czas trwania impulsu wynosi jedną pętlę programową.

Oto przykład wykorzystania przekaźnika czasowego TD:

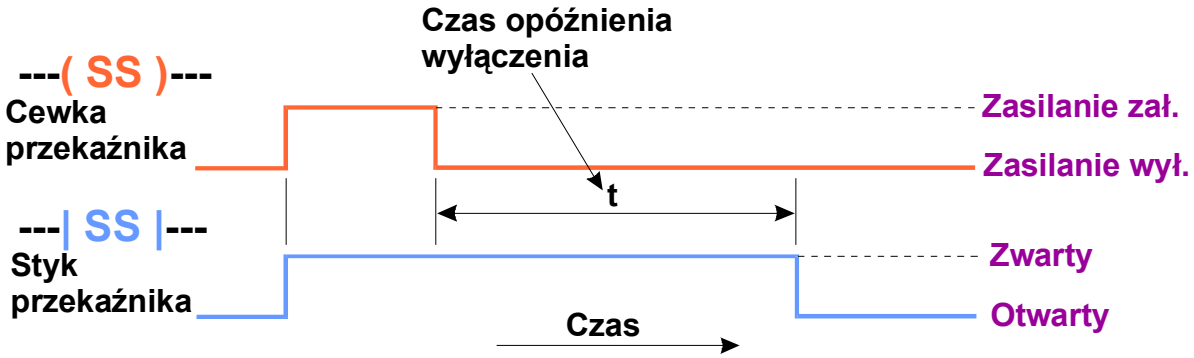


W przykładzie wykorzystano przekaźnik czasowy o adresie TD1. Pojawienie się wysokiego stanu na wejściu impulsowym X0 spowoduje uruchomienie przekaźnika czasowego TD1. Czas opóźnienia załączenia wyjścia pobierany jest z rejestru wartości zadanych o adresie WR0. Przekaźnik będzie odmierzał sekundy ponieważ rejestrem taktującym jest rejestr generujący jeden impuls na sekundę R250. Przy pojawieniu się kolejnych impulsów będzie następowało zmniejszenie rejestru odliczającego. Jeżeli rejestr odliczający osiągnie wartość równą zero wyjście przekaźnika załączy się. Wyjście zostanie tek długo załączone aż stan na wejściu impulsowym X0 nie zmieni się z wysokiego na niski. W linii 3: stan z wyjścia przekaźnika czasowego przekazany jest do wyjścia impulsowego o adresieY0. W pokazanym programie przekaźnik pracuje licznik odliczył już zadany czas ale jeszcze nie zdążył załączyć wyjścia przekaźnika czasowego.

---(SS)--- Przekaźnik czasowy – opóźnione wyłączenie (Single Shot)

	---(SS)--- kod 152 , 7		
1	Długość funkcji w bajtach	7	
2	Kod funkcji	152	
3	Numer zegara adr 0-63	0-63	Obszar wspólny dla TD i SS
4	Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
5	Nazwa rejestru zależna od wyboru z linii 4	XX	Adres użytego rejestru
6	Nazwa rejestru zależna od wyboru z linii 4	XX	Adres użytego rejestru
7	Rejestr taktujący R adr. 0 - 255	0-255	Adres użytego rejestru

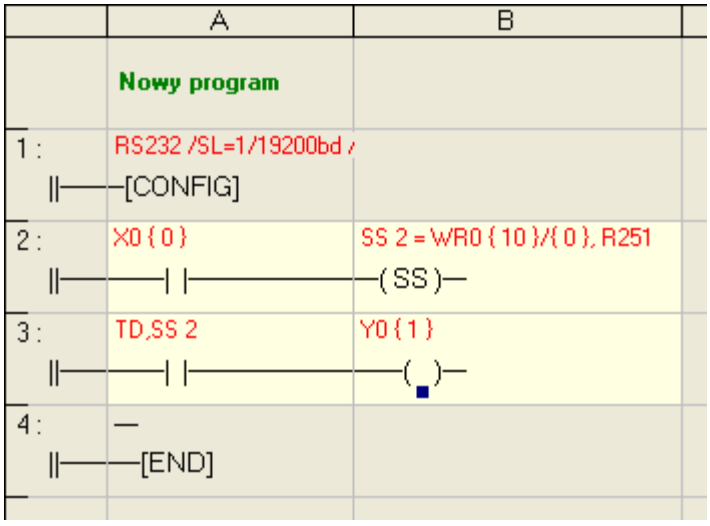
Rozkazem tym kończymy linę w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja imituje pracę przekaźnika czasowego z opóźnionym wyłączeniem. Jeżeli przekaźnik czasowy podłączymy do zasilania to jego styki natychmiast się zewrą. Po odłączeniu od zasilania musi upłynąć pewien czas po jakim styki otworzą się. Czas po jakim nastąpi rozwarcie zwarcie styków jest regulowany w zakresie od sekund do godzin. Wyjaśnienie jak pracuje przekaźnik czasowy z opóźnionym wyłączeniem najlepiej pokazać w następujący sposób:



Funkcja ta do swojej pracy wykorzystuje trzy parametry. Pierwszy parametr jest numerem przekaźnika czasowego w zakresie od 0 do 63. Numery przekaźników TD i SS nie mogą się powtarzać. Drugim parametrem jest rejestr WR który określa po jakim czasie przekaźnik ma się wyłączyć. Trzeci parametr to impuls taktujący zegara. W zależności od tego jaki czas ma odmierzać przekaźnik tak dobrany jest drugi parametr. Parametr ten to przekaźnik wewnętrzny sterownika który działa podobnie do przekaźników wyjść impulsowych lecz nie ma do niego przyłączonego fizycznego przekaźnika. Jest to tylko rodzaj pamięci wewnętrznej. Przekaźniki wewnętrzne nazwane literką R (Relay) a ich ilość wynosi 256 o adresach od R0 – R255. Większość przekaźników można wykorzystywać do dowolnych celów. Jest jednak grupa przekaźników która ma z góry narzuconą funkcję. My będziemy wykorzystywać przekaźniki o adresach od R247 do R250. Każdy z nich może być użyty do generowania impulsu czasowego. Oto zestawienie i funkcja wykorzystywanych przez nas rejestrów:

R247 - Przekaźnik wewnętrzny generujący jeden impuls na dobę
R248 - Przekaźnik wewnętrzny generujący jeden impuls na godzinę
R249 - Przekaźnik wewnętrzny generujący jeden impuls na minutę
R250 - Przekaźnik wewnętrzny generujący jeden impuls na sekundę

Czas trwania impulsu wynosi jedną pętlę programową.
Oto przykład wykorzystania przekaźnika czasowego SS:



W przykładzie wykorzystano przekaźnik czasowy o adresie SS2. Pojawienie się wysokiego stanu na wejściu impulsowym X0 spowodowało uruchomienie przekaźnika czasowego i załączenie jego wyjścia. Czas opóźnienia wyłączenia wyjścia pobierany jest z rejestru wartości zadanych o adresie WR0. Przekaźnik będzie odmierzał minuty ponieważ rejestrem taktującym jest rejestr generujący jeden impuls na minutę R251. Przy pojawieniu się kolejnych impulsów będzie następowało zmniejszenie rejestru odliczającego. Jeżeli rejestr odliczający osiągnie wartość równą zero wyjście przekaźnika wyłączy się. W linii 3: stan z wyjścia przekaźnika czasowego przekazany jest do wyjścia impulsowego o adresieY0. W pokazanym programie przekaźnik pracuje licznik odliczył już zadany czas ale jeszcze nie zdążył wyłączyć wyjścia przekaźnika czasowego.

3.RELACJE LOGICZNE

||---| A == B |--- Warunek równości

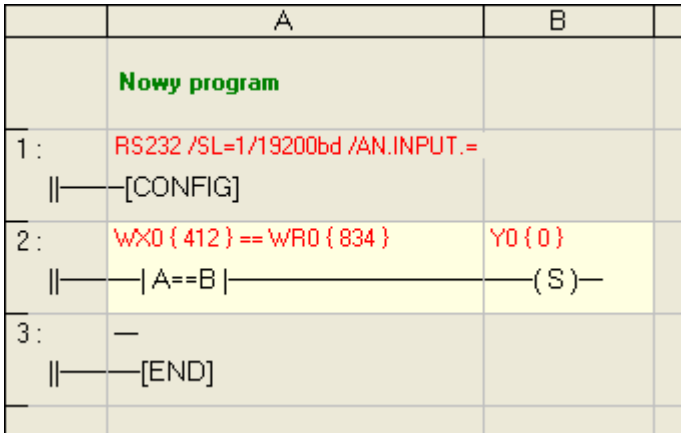
	||---| A == B |--- kod 160 , 8		
1	Długość funkcji w bajtach	8	
2	Kod funkcji	160	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
6	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
7	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru
8	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru

Rozkazem tym rozpoczynamy każdą linię, w której będą występowały operacje logiczne. Funkcja pozwala na wykrycie czy zmienna A i B są sobie równe. Innymi słowy, wynikiem działania tej funkcji jest przekazanie na wyjście z funkcji stanu logicznego 0 lub 1 /FAŁSZ lub PRAWDA/. Jeżeli warunek A==B jest spełniony, to na wyjście zostaje wysłana logiczna jedynka. Jeżeli warunek nie jest spełniony, funkcja wysyła wartość równą zero.

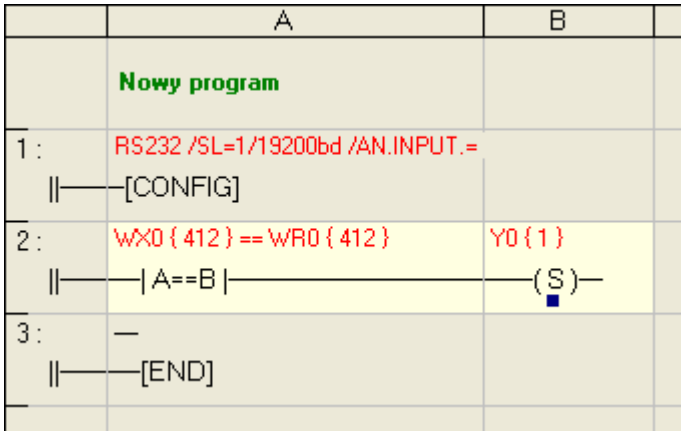
Porównywanymi zmiennymi mogą być :

Wejścia impulsowe	o adresach X0 do X127	- wartość 0 lub 1
Wyjścia impulsowe	o adresach Y0 do Y127	- wartość 0 lub 1
Przekaźniki wewnętrzne	o adresach R0 do R255	- wartość 0 lub 1
Wejścia analogowe	o adresach WX0 do WX63	- wartość od 0 do 1024
Wyjścia analogowe	o adresach WY0 do WY31	- wartość od 0 do 255
Rejestry wartości zadanych	o adresach WR0 do WR255	- wartość od 0 do 4095

Można porównywać ze sobą różne typy wyżej wymienionych zmiennych np. wejścia impulsowe, które mogą przyjmować stan 0 lub 1, rejestry wartości zadanych mogących przyjmować wartość od 0 do 4095 oraz stany odczytane z wejść analogowych o wartości od 0 - 1024.
Sprawdźmy jak działa ta funkcja w programie:



W powyższym programie porównywane są dwie wartości, WX0 reprezentująca stan odczytany z wejścia analogowego oraz WR0, wartość, która jest zapamiętana w rejestrze wartości zadanych. Na przykładzie pokazano, że aktualna wartość wejścia analogowego wynosi 412, a rejestru wartości zadanych jest równa 834. Porównywane wartości są różne, w związku z czym warunek nie został spełniony, a co zatem idzie, do gałęzi został przekazany stan 0, co w efekcie nie powoduje zmiany stanu wyjścia impulsowego o adresie Y0.

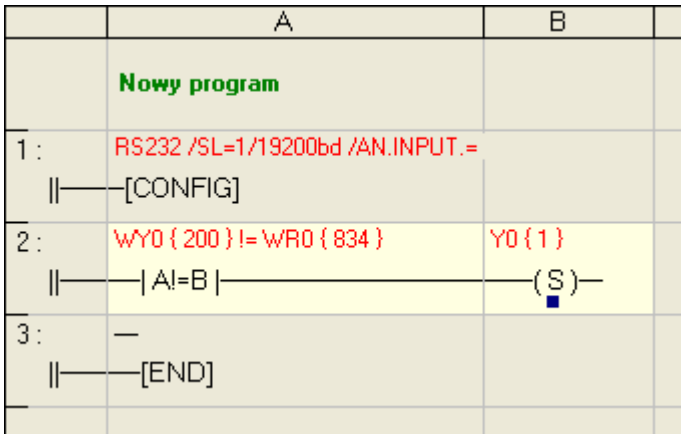


Jeśli wartości wejścia analogowego i wartość w rejestrze zrównają się, na wyjście z funkcji podany zostanie stan 1 co spowoduje załączenie na stałe wyjścia impulsowego Y0. Wyjście to zostanie załączone na stałe, ponieważ funkcja ---(S)--- może tylko załączyć wyjście (patrz opis funkcji „Przekaźnik wyjściowy załączany stanem wysokim”).

||---| A != B |--- Warunek nierówności

	||---| A != B |--- kod 161 , 8		
1	Długość funkcji w bajtach	8	
2	Kod funkcji	161	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
6	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
7	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru
8	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru

Rozkazem tym rozpoczynamy każdą linię, w której będą występowały operacje logiczne. Funkcja pozwala na wykrycie czy zmienne A i B są różne. Innymi słowy, wynikiem działania tej funkcji jest przekazanie na wyjście z funkcji stanu logicznego 0 lub 1. Jeżeli warunek A!=B jest spełniony, to na wyjście zostaje wysłana logiczna jedynka. Jeżeli warunek nie jest spełniony, funkcja wysyła wartość równą zero. Typy porównywanych zmiennych opisano w funkcji A==B.
Sprawdźmy jak działa ta funkcja w programie:

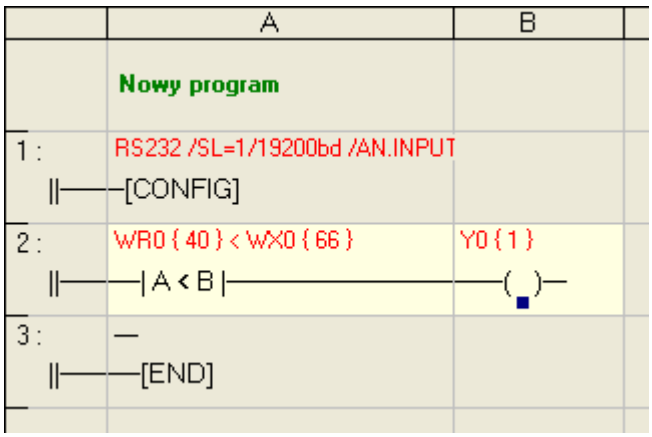


Jak widać, teraz porównywana jest zawartość rejestru wyjść analogowych WY0=200 oraz, jak poprzednio, rejestr WR0=834. Dwie odczytywane wartości różnią się od siebie {200} != {834}, dlatego nastąpiło załączenie na stałe impulsowego Y0.

||---| A < B |--- Warunek mniejszości

	||---| A < B |--- kod 162 , 8		
1	Długość funkcji w bajtach	8	
2	Kod funkcji	162	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
6	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
7	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru
8	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru

Rozkazem tym rozpoczynamy każdą linię, w której będą występowały operacje logiczne. Funkcja pozwala na wykrycie czy zmienna A jest mniejsza od B. Innymi słowy, wynikiem działania tej funkcji jest przekazanie na wyjście z funkcji stanu logicznego 0 lub 1. Jeżeli warunek A<B jest spełniony, to na wyjście zostaje wysłana logiczna jedynka. Jeżeli warunek nie jest spełniony, funkcja wysyła wartość równą zero. Typy porównywanych zmiennych opisano w funkcji A==B.

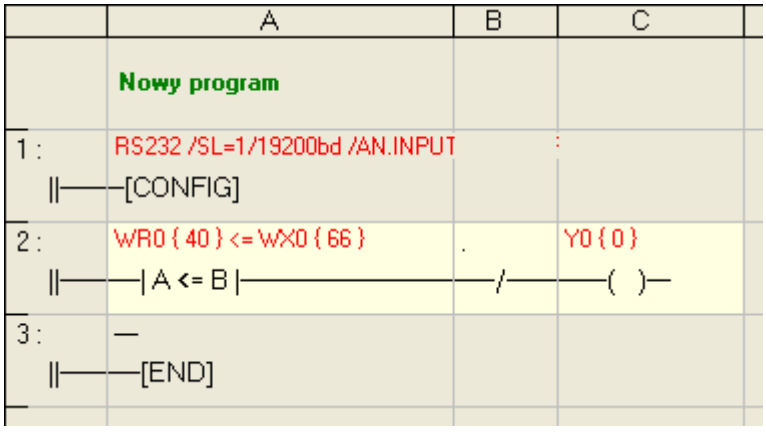


Powyższy program posiada już praktyczne zastosowanie i może służyć do kontroli temperatury. Wartość zadana WR0 ustawiona jest na 40 stopni Celsjusza, a do wejścia analogowego podłączony jest termometr mierzący temperaturę od 0 – 100 °C. Wyjście impulsowe Y0 załączy się za każdym razem, kiedy temperatura na wejściu WX0 przekroczy wartość zadaną = 40 °C. Kiedy temperatura spadnie poniżej wartości zadanej wyjście wyłączy się.

||---| A <= B |--- Warunek mniejszy równy

	||---| A <= B |--- kod 163 , 8		
1	Długość funkcji w bajtach	8	
2	Kod funkcji	163	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
6	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
7	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru
8	Nazwa rejestru zależna od wyboru z linii 6	XX	Adres użytego rejestru

Rozkazem tym rozpoczynamy każdą linię, w której będą występowały operacje logiczne. Funkcja jest połączeniem dwóch funkcji A==B oraz A < B. Pozwala sprawdzić czy zmienna jest A mniejsza lub równa od B. Innymi słowy, wynikiem działania tej funkcji jest przekazanie na wyjście z funkcji stanu logicznego 0 lub 1. Jeżeli warunek A<=B jest spełniony, to na wyjście zostaje wysłana logiczna jedynka. Jeżeli warunek nie jest spełniony, funkcja wysyła wartość równą zero. Typy porównywanych zmiennych opisano w funkcji A==B.
W poniższym przykładzie pokazano jak zmienić kierunek i rodzaj nierówności. Przy pomocy negacji gałęzi możemy zrobić z funkcji A <= B funkcję A > B. Sprawdźmy jak to działa:



Jak widać, wynik funkcji jest prawdziwy, lecz wyjście się nie załącza. Z warunku A <= B otrzymujemy logiczną jedynkę, ale zostaje ona zanegowana przez funkcję negacji ---/---. W efekcie, do wyjście impulsowego dochodzi stan niski i przekaźnik wyjściowy się nie załączy. W ten sposób zapisany program będzie działał prawidłowo, ale nie będzie zoptymalizowany pod kątem prędkości. Nie dość, że warunek <= nie należy do najszybciej wykonywanych funkcji, ponieważ sprawdzane są dwa warunki == i <, to jeszcze dochodzi negacja. Najlepiej korzystać z rozkazów przeznaczonych dla danych operacji.

4.FUNKCJE MATEMATYCZNE

---| D = S |--- Przypisanie

	---| D = S |--- kod 176 , 7		
1	Długość funkcji w bajtach	7	
2	Kod funkcji	176	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala przepisać wartość ze zmiennej występującej po prawej stronie równania, do zmiennej znajdującej się po jego lewej stronie. Funkcja działa tylko wtedy, gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe jest przypisywanie zmiennych różnych typów. Oczywiście przypisanie np. zmiennej impulsowej R wartości z rejestru WR spowoduje, że wartość rejestru WR zostanie ograniczona do stanu 0 lub 1. Jeżeli rejestr WR będzie miał wartość większą od zera, wtedy stan w rejestrze R będzie równy 1. Nie ma możliwości przypisania rejestrowi wejść analogowych żadnej wartości, ponieważ rejestr ten jest zapisywany tylko przez wartości pobierane z wejść analogowych. Po lewej stronie równania mogą znaleźć się tylko rejestry do których można dokonać zapisu WR, WY, R, Y.

---[D = S1 + S2]--- Suma dwóch argumentów

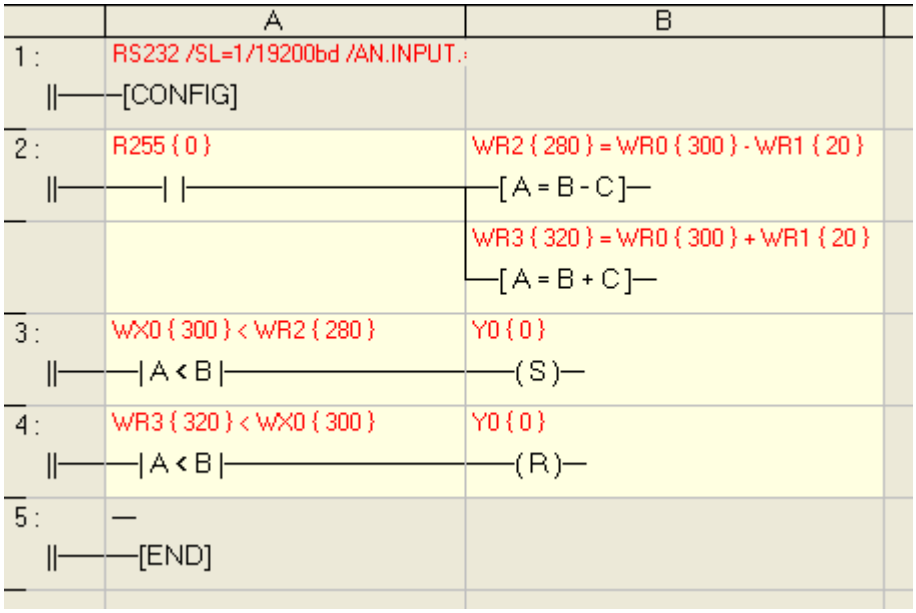
	---[D = S1 + S2]--- kod 177 , 10		
1	Długość funkcji w bajtach	10	
2	Kod funkcji	177	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
8	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
9	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru
10	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala na wyliczenie sumy dwóch argumentów i przepisanie wyniku do zmiennej znajdującej się po lewej stronie. Funkcja działa tylko wtedy, gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe są działania na zmiennych różnych typów. Aby działanie wykonane było prawidłowo, należy tak dobrać argumenty, aby nie została przekroczona pojemność zmiennej umieszczonej po lewej stronie działania. Jeżeli wartość z działania przekracza pojemność zmiennej, wynikiem będzie maksymalna wartość jaką może pomieścić wybrany rejestr i zostanie ustawiony wskaźnik przepełnienia, którym jest przekaźnik wewnętrzny R242. Dla rejestrów WR będzie to wartość 4095, a dla WY wartość 255.
Po lewej stronie równania mogą znaleźć się tylko rejestry, do których można dokonać zapisu WR, WY, R, Y.

---[D = S1 - S2]--- Różnica dwóch argumentów

	---[D = S1 - S2]--- kod 178 , 10		
1	Długość funkcji w bajtach	10	
2	Kod funkcji	178	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
8	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
9	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru
10	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala na wyliczenie różnicy dwóch argumentów i przepisaniu wyniku do zmiennej znajdującej się po lewej stronie równania. Funkcja działa tylko wtedy, gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe są działania na zmiennych różnych typów. Jeżeli wartość z działania będzie mniejsza od zera, zmienna, do której zapisywany jest wynik, przyjmie wartość równą zero.
Po lewej stronie równania mogą znaleźć się tylko rejestry, do których można dokonać zapisu WR, WY, R, Y. Najprostszym przykładem sprawdzenia sposobu działania funkcji sumy i różnicy dwóch argumentów, jest regulacja poziomu wody w zbiorniku. Aby można było regulować poziomem wody, na dopływie powinien być umieszczony zawór odcinający. Parametrami zadanymi powinien być poziom wody oraz histereza pracy zaworu. Poziom mierzony będzie za pośrednictwem sondy ultradźwiękowej, która podaje odczyt ciągły w zakresie od 0 dla zbiornika pustego, do 1000 dla zbiornika całkowicie napełnionego. Sonda podłączona będzie do wejścia analogowego o adresie WX0. Zawór podłączony będzie do wyjścia impulsowego o adresie Y0. Oto kompletny program, który pozwala sterować zaworem dopełniającym.



Sporządźmy teraz wykaz sygnałów i wartości zadanych, jakimi będziemy się posługiwać w programie:
Wartości zadane:
WR0 – zadany poziom wody w zbiorniku
WR1 – histereza pracy zaworu
WR2 – poziom, przy którym nastąpi załączenie zaworu
WR3 – poziom, przy którym nastąpi wyłączenie zaworu

Wejścia analogowe
WX0 – analogowy odczyt poziomu wody w zbiorniku

Wyjścia impulsowe
Y0 – przekaźnik sterujący elektromagnetycznym zaworem dopełniającym

Przekaźniki wewnętrzne
R255 – przekaźnik zawsze załączony przez system

Nasz program sterujący zaczyna się w linii drugiej. W linii tej, jako pierwsza, umieszczona jest funkcja, na wyjściu której zawsze będzie stan = 1. Funkcja musi być użyta, ponieważ funkcje matematyczne nie mogą rozpoczynać nowej linii. W kolumnie „2:B" znajdują się dwie funkcje wyliczające poziom, przy którym powinno nastąpić załączenie oraz wyłączenie uzupełniania wody w zbiorniku. Poziom, przy którym następuje załączenie zaworu wyliczany jest przez odjęcie od zadanego poziomu wody histerezy pracy zaworu. W ten sposób, jeżeli poziom wody będzie ustawiony na 300 cm, a histereza na 20 cm, zawór załączy się wtedy, gdy sonda poziomu wskaże poniżej 280 cm. Podobnie jest z wyłączaniem zaworu. Poziom, przy którym nastąpi wyłączenie zaworu wyliczany jest przez dodanie do zadanego poziomu wody histerezy pracy zaworu. W wyniku otrzymamy tym razem poziom 320 cm, oznacza to że zawór zostanie wyłączony gdy sonda wskaże odczyt większy do tego poziomu. W gałęzi drugiej dokonano tylko wyliczenia górnego i dolnego poziomu pracy. Właściwa regulacja zaworem odbywa się linii 3 i 4. Linia 3 załącza zawór, a następna go wyłącza. Użyto tutaj funkcji mniejszości i dlatego zawór będzie przełączał się jeżeli poziom przekroczy wartość wyliczoną. Jeżeli chcielibyśmy aby zawór przełączał się przy wartości wyliczonej, trzeba by użyć funkcji <=.
Prześledźmy teraz jaka jest dynamika procesu sterowania. Mamy pusty zbiornik. Odczyt z sondy poziomu jest równy zero. Warunek w linii 2 otwiera zawór. Zawór jest otwarty do czasu kiedy warunek z linii 3 nie wyłączy go, co oznacza, że poziom osiągnie wartość większą od 320 cm. Teraz następuje odbiór wody ze zbiornika i poziom obniża się. W chwili gdy spada on do wartości poniżej 280 cm, następuje ponowne załączenie zaworu, i cykl się powtarza. Jeżeli poziom wody znajduje się w zakresie pomiędzy 280 a 320 cm, program nie steruje zaworem i pozostaje on w pozycji do jakiej przełączył go ostatni spełniony warunek logiczny.
Nasuwa się pytanie- dlaczego sterownie zaworem przeprowadzane jest w tak skomplikowany sposób?. Czy nie wystarczyło by napisać program analogiczny do tego, jaki był pokazany przy opisie funkcji | A < B |. Niestety, w rzeczywistości nie mamy do czynienia z sygnałami, które zmieniają się w sposób płynny, a ich wartość zmienia się tylko w jednym kierunku. Wartości odczytywane przez przyrządy zamontowane na obiektach technologicznych są bardzo często zakłócane, co w efekcie może doprowadzić do uszkodzenia urządzeń sterowanych (zbyt częste załączenie i wyłączenie), dlatego potrzebna jest histereza, przy której następuje pewne przełączenie urządzenia wykonawczego.

---[D = S1 * S2]--- Iloczyn dwóch argumentów

	---[D = S1 * S2]--- kod 179 , 10		
1	Długość funkcji w bajtach	10	
2	Kod funkcji	179	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
8	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
9	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru
10	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala na wyliczenie iloczynu dwóch argumentów i przepisanie wyniku do zmiennej znajdującej się po lewej stronie. Funkcja działa tylko wtedy gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe są działania na zmiennych różnych typów. Aby działanie wykonane było prawidłowo, należy tak dobrać argumenty, aby nie została przekroczona pojemność zmiennej umieszczonej po lewej stronie działania. Jeżeli wartość z mnożenia przekracza pojemność zmiennej, wynikiem będzie maksymalna wartość jaką może pomieścić wybrany rejestr i zostanie ustawiony wskaźnik przepełnienia, którym jest przekaźnik wewnętrzny R242. Dla rejestrów WR będzie to wartość 4095, a dla WY wartość 255.
Po lewej stronie równania mogą znaleźć się tylko rejestry, do których można dokonać zapisu WR, WY, R, Y.

---[D = S1 / S2]--- Iloraz dwóch argumentów

	---[D = S1 / S2]--- kod 180 , 10		
1	Długość funkcji w bajtach	10	
2	Kod funkcji	180	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
8	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
9	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru
10	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala na wyliczenie ilorazu dwóch argumentów i przepisanie wyniku do zmiennej znajdującej się po lewej stronie równania. Funkcja działa tylko wtedy, gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe są działania na zmiennych różnych typów. Jeżeli wartość z dzielenia będzie mniejsza od zera, zmienna, do której zapisywany jest wynik, przyjmie wartość równą zero.
Po lewej stronie równania mogą znaleźć się tylko rejestry, do których można dokonać zapisu WR, WY, R, Y.

---[D = S1 OR S2]--- Suma logiczna dwóch argumentów

	---[D = S1 OR S2]--- kod 181 , 10		
1	Długość funkcji w bajtach	10	
2	Kod funkcji	181	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
8	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
9	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru
10	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala na wyliczenie sumy logicznej dwóch argumentów i przepisanie wyniku do zmiennej znajdującej się po lewej stronie równania. Funkcja działa tylko wtedy, gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe są działania na zmiennych różnych typów. Działanie wykonywane jest na poszczególnych bitach zdefiniowanych zmiennych. Dla przykładu, jeżeli zmienna S1 będzie równa 7 (binarnie 00000111) a zmienna S2 będzie równa 1 (binarnie 00000001) to wynik zapisany do zmiennej D będzie wynosił 7 (binarnie 00000111).

```
    00000111
    00000001
OR  ________
    00000111
```

Jeżeli zmienna S1 będzie równa 7 (binarnie 00000111) a zmienna S2 będzie równa 8 (binarnie 00001000) to wynik zapisany do zmiennej D będzie wynosił 15 (binarnie 00001111).

```
    00000111
    00001000
OR  ________
    00001111
```

Po lewej stronie równania mogą znaleźć się tylko rejestry, do których można dokonać zapisu WR, WY, R, Y.

---[D = S1 AND S2]--- Iloczyn logiczny dwóch argumentów

	---[D = S1 AND S2]--- kod 182 , 10		
1	Długość funkcji w bajtach	10	
2	Kod funkcji	182	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
8	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
9	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru
10	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala na wyliczenie iloczynu logicznego dwóch argumentów i przepisanie wyniku do zmiennej znajdującej się po lewej stronie równania. Funkcja działa tylko wtedy, gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe są działania na zmiennych różnych typów. Działanie wykonywane jest na poszczególnych bitach zdefiniowanych zmiennych.
Dla przykładu, jeżeli zmienna S1 będzie równa 7 (binarnie 00000111), a zmienna S2 będzie równa 1 (binarnie 00000001), to wynik zapisany do zmiennej D będzie wynosił 1 (binarnie 00000001).

```
        00000111
        00000001
AND _____________
        00000001
```

Jeżeli zmienna S1 będzie równa 7 (binarnie 00000111), a zmienna S2 będzie równa 8 (binarnie 00001000), to wynik zapisany do zmiennej D będzie wynosił 0 (binarnie 00000000).

```
        00000111
        00001000
AND _____________
        00000000
```

Po lewej stronie równania mogą znaleźć się tylko rejestry, do których można dokonać zapisu WR, WY, R, Y.

---[D = S1 XOR S2]--- Alternatywa logiczna dwóch argumentów

	---[D = S1 XOR S2]--- kod 183 , 10		
1	Długość funkcji w bajtach	10	
2	Kod funkcji	183	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Nazwa rejestru zależna od wyboru z linii 3	XX	Adres użytego rejestru
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
6	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
7	Nazwa rejestru zależna od wyboru z linii 5	XX	Adres użytego rejestru
8	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Liczba kod 7 0 - 4095 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
9	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru
10	Nazwa rejestru zależna od wyboru z linii 8	XX	Adres użytego rejestru

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcja pozwala na wyliczenie alternatywy logicznej dwóch argumentów i przepisanie wyniku do zmiennej znajdującej się po lewej stronie równania. Funkcja działa tylko wtedy, gdy na jej wejściu pojawi się stan wysoki. Jeżeli z poprzedniej funkcji przekazany będzie stan niski, funkcja nie będzie wykonywana. Możliwe są działania na zmiennych różnych typów. Działanie wykonywane jest na poszczególnych bitach zdefiniowanych zmiennych.
Dla przykładu, jeżeli zmienna S1 będzie równa 7 (binarnie 00000111), a zmienna S2 będzie równa 1 (binarnie 00000001), to wynik zapisany do zmiennej D będzie wynosił 6 (binarnie 00000110).

```
        00000111
        00000001
XOR _____________
        00000110
```

Jeżeli zmienna S1 będzie równa 7 (binarnie 00000111), a zmienna S2 będzie równa 8 (binarnie 00001000), to wynik zapisany do zmiennej D będzie wynosił 15 (binarnie 00001111).

```
        00000111
        00001000
XOR  ____________
        00001111
```

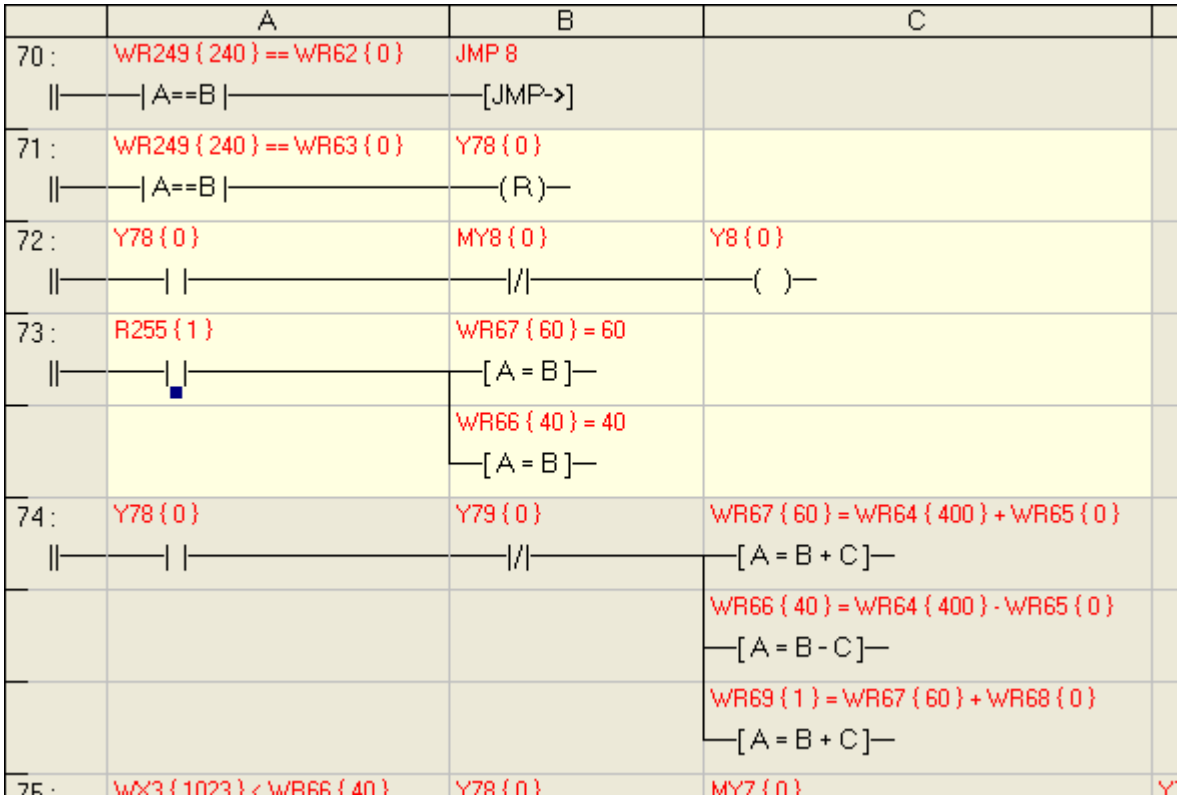
Po lewej stronie równania mogą znaleźć się tylko rejestry, do których można dokonać zapisu WR, WY, R, Y.

5.FUNKCJE STERUJĄCE

----[JMP->] Skok warunkowy

	---[JMP->] kod 48 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	48	
3	Ilość ominiętych funkcji ... 0 - 255	0-255	

Rozkazem tym kończymy linię, w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją w linii. Dzięki niej, możemy ominąć wykonanie do 255 funkcji znajdujących się w programie, następujących po funkcji JMP->. Skok wykonywany jest tylko do przodu. Warunkiem wykonania skoku jest podanie stanu wysokiego na wejściu do funkcji. Jeżeli do funkcji dochodzi stan niski, funkcja skoku jest ignorowana. Funkcją tą należy posługiwać się bardzo rozważnie, ponieważ wykonanie nieprawidłowego skoku może spowodować powstanie błędów w programie. Skoki wykonuje się z reguły do początków linii logicznych. „Wskoczenie" w drugą lub trzecią funkcję w gałęzi spowoduje wpisanie do niej stanu ostatnio wykonywanej funkcji przed skokiem (stanu = 1 który inicjuje skok) i może spowodować niestabilność programu. Katastrofalne w skutkach może być ominięcie funkcji końca programu [END]. Funkcja skoku posiada jeden parametr jakim jest ilość omijanych funkcji. Jeżeli jako parametr wpiszemy wartość = 3, to zostaną ominięte trzy funkcje licząc od pierwszej funkcji występującej za funkcją [JMP->]. Program zostanie wykonywany od 4 w kolejności funkcji. Podanie jako parametru wartości = 0 nie spowoduje skoku, a program będzie wykonywany tak jak by nie było funkcji JMP.
Zobaczmy jak to działa na przykładzie programu:



Funkcja skoku warunkowego znajduje się w linii 70:B. Program dokona skoku dopiero wtedy, gdy wartości w rejestrach WR249 i WR62 zrównają się. Jeśli wynik porównania będzie stanem niskim, skok nie zostanie wykonany i program będzie realizowany w całości. Parametrem dla funkcji JMP jest 8. Oznacza to, że zostanie ominiętych osiem funkcji. Policzmy, które funkcje zostaną ominięte. Dwie w linii 71, trzy w linii 72 i kolejne trzy w linii 73. Funkcje te znajdują się w jaśniejszych polach. Po wykonaniu skoku, program zostanie wznowiony od początku linii 74:A.

----[MCS_]--- Sterownie blokiem programu - inicjacja

Funkcja w przygotowaniu.

----[MCR_]--- Sterownie blokiem programu - zakończenie

Funkcja w przygotowaniu.

----[MPS]--- Rozgałęzienie linii – Zapis stanu linii

	----[MPS]--- kod 143 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	143	
3	Nieużywane	0	

Rozkazem tym nigdy nie rozpoczynamy ani nie kończymy linii, w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna funkcja gałęzi. Funkcja pozwala rozgałęzić linię logiczną na dwie lub więcej linii i rozpoczyna rozgałęzienie linii. Działanie funkcji polega na tym, że zapisany zostaje stan logiczny gałęzi w miejscu wystąpienia funkcji [MPS]. Stan ten przechowywany jest w specjalnym rejestrze i jest możliwy do odczytu w dowolnej chwili za pośrednictwem funkcji MRD lub MPP.

----[MRD]--- Rozgałęzienie linii – Odczyt stanu linii

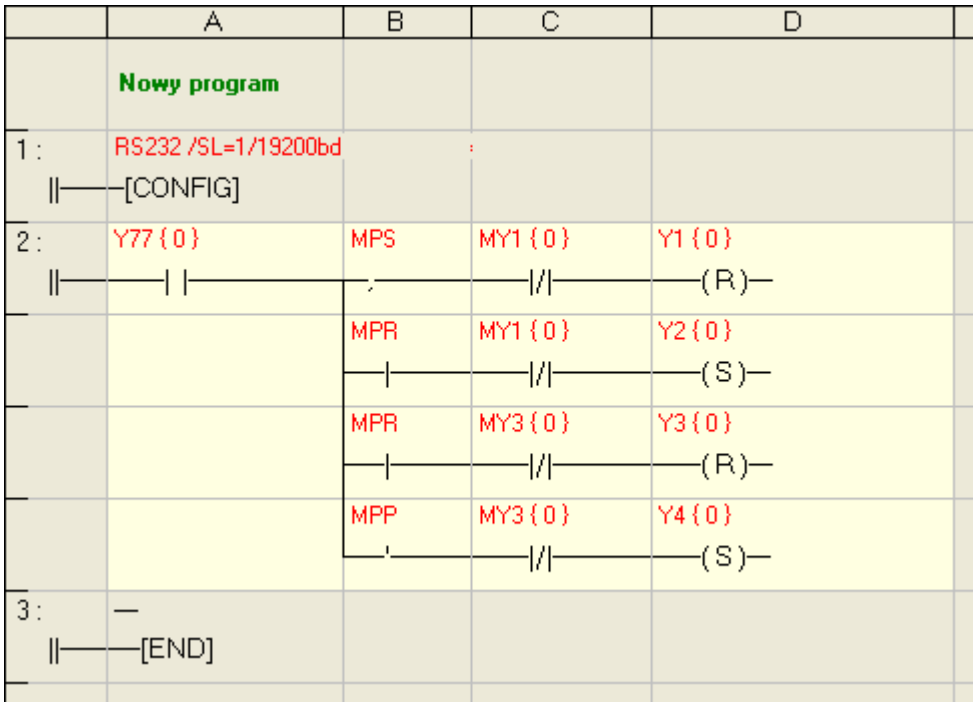
	----[MRD]--- kod 144 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	144	
3	Nieużywane	0	

Rozkazem tym nigdy nie rozpoczynamy ani nie kończymy linii, w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna funkcja gałęzi. Funkcja odczytuje stan linii w miejscu, w którym wystąpiła funkcja [MPS]. Dokładniej rzecz ujmując, funkcja odczytuje zawartość rejestru, do którego funkcja [MPS] zapisała wcześniej stan występujący w gałęzi programu Odczytany stan pojawia się na wyjściu funkcji [MRD].

----[MPP]--- Rozgałęzienie linii – Ostatni odczyt stanu linii

	----[MPP]--- kod 145 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	145	
3	Nieużywane	0	

Rozkazem tym nigdy nie rozpoczynamy ani nie kończymy linii, w której będą występowały operacje logiczne. Funkcja ta występuje w kolejności jako druga lub następna funkcja gałęzi. Funkcja odczytuje stan linii w miejscu, w którym wystąpiła funkcja [MPS]. Odczytany stan pojawia się na wyjściu funkcji [MPP] a zawartość rejestru, do którego funkcja [MPS] dokonała zapisu jest kasowana a rejestr zwalniany do kolejnego zapisu. Funkcja ta jest funkcją zamykającą rozgałęzienie linii.
Zobaczmy jak działają funkcje MPS, MRD, MPP:



Na przykładzie pokazano rozgałęzienie przy pomocy funkcji [MPS]. Funkcja ta odczytuje stan wyjścia impulsowego Y77 i zachowuje go w specjalnym rejestrze pomocniczym. Program jest realizowany dalej. Jeżeli program napotka funkcję [MPR] odczyta rejestr specjalny i jego zawartość wstawi do gałęzi na swoje wyjście dalsze wykonanie programu będzie wykonywane tak jakby poprzednią funkcją był odczyt wyjścia impulsowego Y77. Ostatnim rzędzie w linii 2 znajduje się funkcja [MRD], która jeszcze raz odczytuje rejestr specjalny i wstawia stan z niego do gałęzi. lecz równocześnie odblokowuje ten rejestr do ponownego użycia przez funkcję [MPS], która może być użyta jeszcze raz w dalszej części programu. Jeżeli rozgałęzienia nie zakończymy funkcją MPP, pojawią się błędy w wykonaniu programu

||---[END]--- Koniec programu

	||---[END]--- kod 51 , 3		
1	Długość funkcji w bajtach	3	
2	Kod funkcji	51	
3	Nieużywane	0	

Rozkaz ten występuje jako jedyny w linii. Znajduje się na końcu programu i informuje sterownik o tym, że w tym miejscu zakończył się program technologiczny. Następną wykonywaną funkcją będzie funkcja znajdująca się na początku programu. Brak tej funkcji w programie spowoduje wystąpienie poważnych błędów sterownika. Funkcja nie posiada żadnych dodatkowych parametrów.

6.FUNKCJE SPECJALNE

||---[STORE ANAL.IN] Rejestrator wejść analogowych

Rozkaz umieszczony jest jako jedyny w linii. Pozwala na rejestrację zadanej liczby kanałów analogowych. Jako parametry podawane są ilość zapamiętywanych kanałów oraz adres rejestru WR który określa co ile minut mają być zapamiętywane dane w sterowniku. Zapamiętywane dane są gromadzone a w pamięci która nawet przy braku zasilania nie gubi danych. Zapisane dane mogą być odczytane przez program do obsługi sterownika i automatycznie umieszczone na wykresach. Odczytane dane gromadzone są również na dysku w formacie z którego można je wczytać do innych programów np. Excel.

-----[LOG] Rejestracja dowolnych parametrów sterownika

	------[LOG] kod 5 , 7		
1	Długość funkcji w bajtach	7	
2	Kod funkcji	5	
3	Wejścia impulsowe X kod 0 adr. 0-127 Wyjścia impulsowe Y kod 1 adr. 0-127 Przek. wewnętrzny R kod 2 adr. 0-255 A/MAN impulsowy M kod 4 adr. 0-127 Impuls. zegara TD +SS kod 5 adr. 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Adres pierwszego zapisywanego rejestru	0-254	
5	Ilość zapamiętywanych rejestrów	1-255	
6	Numer banku do którego nastąpi zapis	1-15	
7	Informacja o zapisie ostatniego rekordu	0-255	Jeśli zapisano ostatni rekord w buforze wybrany rejestr R przyjmuje wartość = 1

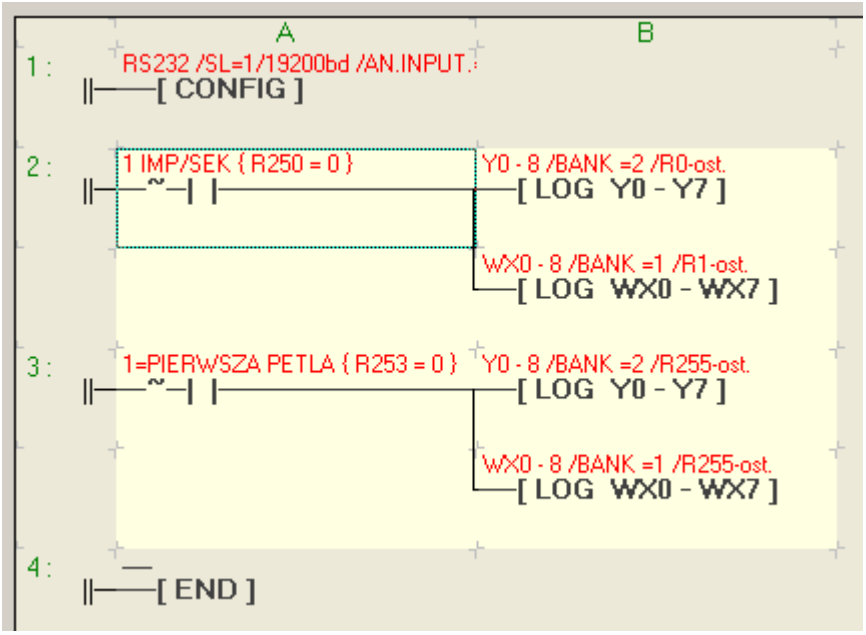
Rozkaz ten wykorzystywany jest do zapisywania stanów dowolnych rejestrów sterownika do pamięci z której można je potem odczytać. Możliwość gromadzenia danych jest bardzo przydatna szczególnie podczas uruchamiania instalacji technologicznych. Rozkazem tym możemy np. zapisać wielkości temperatur zmierzonych w czasie pracy instalacji, następnie przy pomocy komputera możemy je odczytać. Na tej podstawie można dokonać regulacji parametrów instalacji.

Dane mogą być gromadzone w piętnastu bankach pamięci. Pamięć ta w przypadku wyłączenia urządzenia jest zasilana z baterii dzięki czemu dane nawet po wyłączeniu urządzenia nie ulegają wymazaniu. W ramach jednego banku można rejestrować tylko jeden rodzaj danych. W innym banku można równocześnie zapamiętywać dane z innych rejestrów.

Każdy zapisany rekord danych rozpoczyna się od daty składającej się z miesiąca, dnia, godziny, minuty i sekundy. Z tego formatu wynika że dane mogą być zapisywane najczęściej co sekundę. Oczywiście można dane zapisywać częściej lecz nie będą one miały różnych znaczników czasu. Bufor z danymi może przechować dane w zależności od ich ilości. Jeśli rekordy danych są długie to w buforze zmieści się ich mniej. Poszczególne banki można ze sobą łączyć tworząc w ten sposób długi bufor który może zapamiętać większą ilość danych tego samego typu.

Funkcja realizująca zapis do pamięci składa się z pięciu parametrów. Pierwszy z nich definiuje dane z jakiego rejestru będą gromadzone w pamięci. Następny parametr określa adres pierwszego zapisywanego rejestru. Kolejny parametr określa ile ma być zapisanych kolejnych parametrów. Dalej podajemy numer banku do którego będą zapisywane dane. Ostatni parametr ma dwa przeznaczenia. W parametrze tym podajemy adres przekaźnika wewnętrznego R który jest ustawiany gdy sterownik dokonał ostatniego zapisu do wybranego banku pamięci. W ten sposób otrzymujemy informację że następny zapis będzie kasował poprzednio zapisane

dane. Drugie przeznaczenie tego parametru to możliwość formatowania bufora pamięci tak aby mogły być w nim gromadzone dane. Zapisanie funkcji z parametrem który posiada wartość 255 inicjuje bufor pamięci. Dane ze sterownika mogą być odczytywane za pośrednictwem programu PLCcommunicator. Aby dokonać odczytu najpierw należy wybrać bank pamięci który chcemy odczytać. Pogram automatycznie umieści odczytane wartości w tabeli oraz na wykresach.
Poniżej pokazano przykład użycia dwóch banków do zapisu danych. W linii 3 następuje inicjacja banków. Przekaźnik wewnętrzny R253 załącza się tylko raz przy pierwszej pętli programowej po załączeniu sterownika. Taki impuls wystarczy aby sformatować pamięć w której będą zapisywane dane. W linii tej zdefiniowano BANK 2 pamięci. Mają być w nim zapisywane stany wyjść impulsowych od Y0 do Y7. Zapis ostatniego rekordu sygnalizowane jest przez przekaźnik wewnętrzny R0. Równocześnie Zdefiniowano BANK 1. Mają w nim być zapisywane wartości wejść analogowych od WX0 do WX7. Zapis ostatniego rekordu sygnalizowane jest przez przekaźnik wewnętrzny R1. W obydwóch przypadkach formatowanie danych zainicjowane jest przez ostatni parametr funkcji który posiada adres R255.

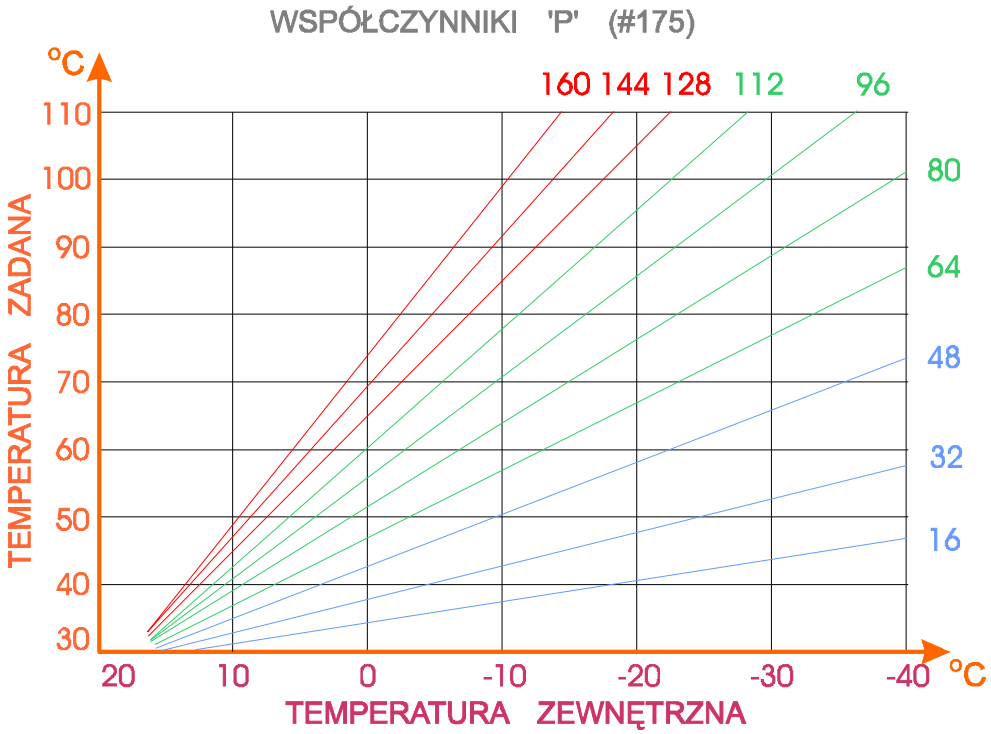


Linia 2 realizuje zapis do obydwóch banków co 1 sekundę. Posłużono się tu rejestrem systemowym R250 wytwarzającym pik co sekundę.

----[POGODÓWKA] Krzywe grzewcze

	----[POGODÓWKA] kod 11 , 5		
1	Długość funkcji w bajtach	7	
2	Kod funkcji	11	
3	Nr. kanału temp. zewn WX_	0-63	
4	Nastawa WR_ z nr. krzywej grzewczej	0-255	Wartość w zakresie od 2 - 160
5	Nastawa WR_ z temperaturą wynikową	0-255	Wartość wyliczana co 10 sekund

Rozkaz umieszczony jest jako drugi lub następny w linii. Warunkiem pracy regulatora jest podanie stanu wysokiego na jego wejście. Stan niski na wejściu do funkcji nie powoduje wyliczenia temperatury zadanej, a w rejestrze z temperaturą wynikową zostają wartości wyliczone ostatnim razem. Funkcja jest niezbędna do obsługi instalacji grzewczych. Pozwala przeliczyć temperaturę zewnętrzną na temperaturę zadaną źródła ciepła za pośrednictwem odpowiedniej krzywej grzewczej. Najczęściej używane krzywe grzewcze mieszczą się w zakresie 64 do 112. Oto zestaw krzywych grzewczych i temperatur obsługiwanych przez funkcję:



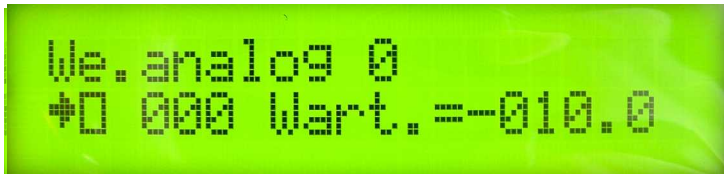
Temperatura zadana źródła ciepła cały czas jest dostosowywana do temperatury zewnętrznej. Przy niższych temperaturach zewnętrznych wzrasta; natomiast gdy temperatura zewnętrzna podnosi się, temperatura wymagana maleje. I tak na przykład dla temperatury zewnętrznej - 10 °C przy zadanym współczynniku ciepła 64 sterownik automatycznie będzie dążył do temperatury w równej wymaganej 60 °C. Jeżeli przy takiej samej temperaturze zewnętrznej zmienimy współczynnik ciepła na 128, to temperatura wymagana źródła ciepła będzie wynosić 90 °C. Taki system pracy pozwala na bardzo szybką reakcję i zabezpieczenie ogrzewanych obiektów przed zbytnim wychłodzeniem lub przegrzaniem. Wyliczenie temperatury wymaganej odbywa się co 10 sekund i wpisywane jest do rejestru WR. Ręczne zmienianie lub użycie tego rejestru do innych celów jest już nie możliwe.
Funkcja korzysta z trzech parametrów: wejścia analogowego WX do którego podłączony jest czujnik temperatury zewnętrznej, rejestru WR w którym zawarta jest krzywa grzewcza w zakresie od 2 do 160 oraz następnego rejestru WR w którym wyliczana będzie temperatura wymagana źródła ciepła.

Uwaga:
Mierzona temperatura zewnętrzna musi być skalowana w zakresie od –51,1 do 51,1; patrz funkcja ||---[RANGE]. Oznacza to że dla temperatury –51,1°C wartość w odczytana z wejścia analogowego będzie =0. Jeżeli odczytywana temperatura zewnętrzna będzie wynosić 0°C wartość na wejściu analogowym będzie = 511. Dla temperatury zewnętrznej 51,1°C odczytywana wartość =1024. Tylko dla takiego skalowania wykres krzywych przedstawiony wcześniej będzie prawidłowy.

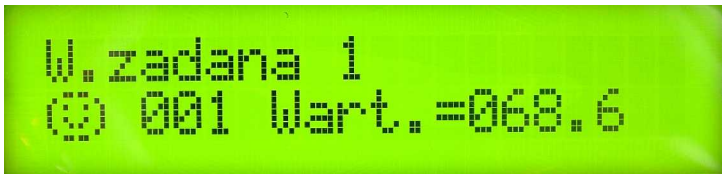
Zobaczmy jak to wygląda na przykładzie. W tym przypadku skalowanie dla pierwszego kanału wejść analogowych będzie znajdowało się w linii 3:A.

	A	B
1 :	RS232 /SL=1/57600bd ||———[CONFIG]	
2 :	Przek.wewn 255 { 1 } ||———| |———[POGODOWKA]	WX 0{ 411 }/WRk 0{ 90 }/WRtz 1{ 686 }
3 :	WX 0/ * 1/ : 1/-.. 130 ||———[RANGE]	
4 :	WR 1/ * 1/ : 1/-.. 2 ||———[RANGE]	
5 :	— ||———[END]	

Wyświetlacz dla wartości wejścia analogowego WX0 = 411 będzie pokazywał –10.0°C.



Aby skalowanie było pełne należy również przeliczyć temperaturę która wynika z krzywej. Rejestr z temperaturą wynikową WR1 będzie miał wartość 686 a na wyświetlaczu po przeskalowaniu które następuje w linii 4:A pojawi się wartość 68.6 °C.

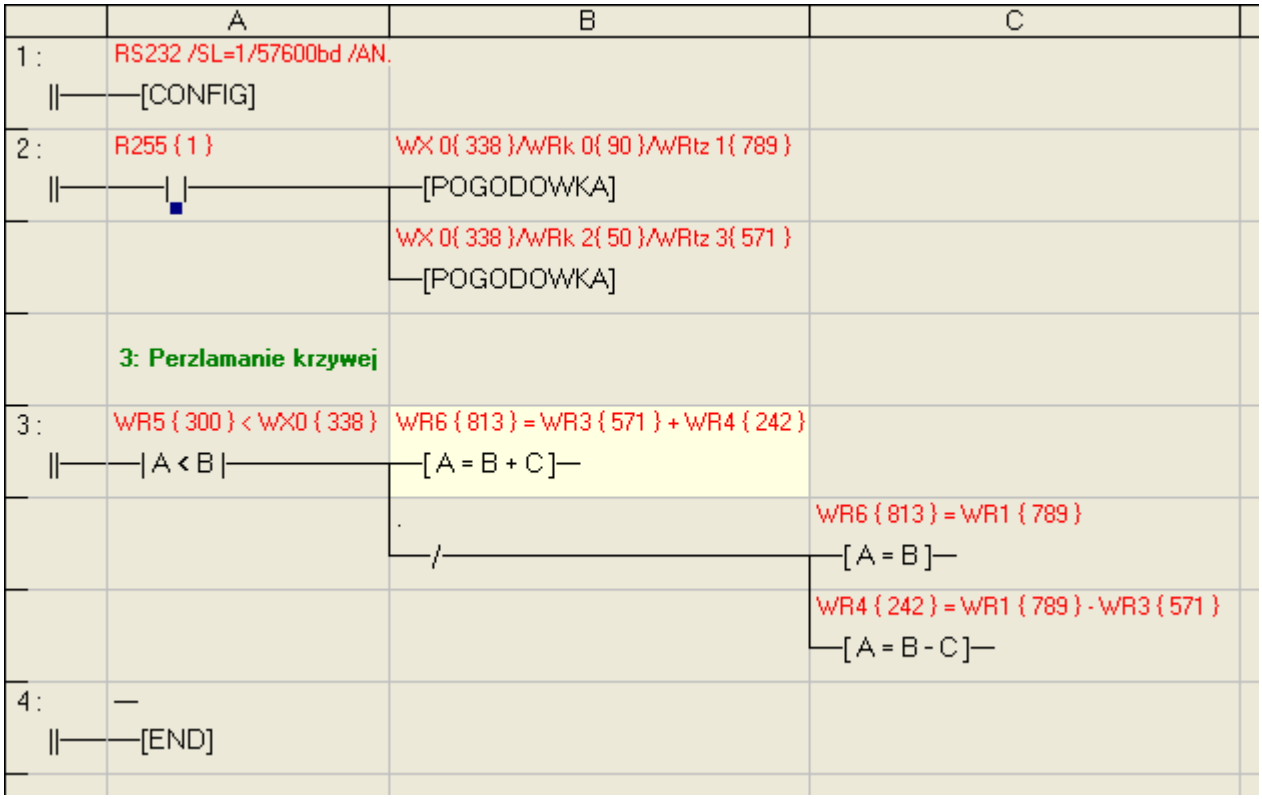


Z linii 2:A wynika ze przeliczenie będzie następowało zawsze ponieważ funkcja „---[POGODOWKA]" cały czas jest wykonywana na wskutek podania stanu wysokiego z funkcji „---| |---„ . Parametry funkcji pogodowej to wartość wejścia analogowego z czujnikiem temperatury zewnętrznej WX0 która ma wartość { 411 }. Następnie numer krzywej grzewczej znajdujący się w rejestrze WR0 (tutaj WRk) który ma wartość { 90 } i odpowiadająca mu wartość wynikowa umieszczona w rejestrze WR1 (tutaj WRtz) który ma wartość { 686 }.
Jeśli obwód czujnika temperatury zostanie rozwarty, to znaczy wartość temperatury odczytana w rejestrze wejść analogowych WX_ będzie równa 1024, funkcja dla bezpieczeństwa instalacji grzewczej jako temperaturę wynikową wstawi wartość z krzywej dla temperatury zewnętrznej równej zero stopni.
Dla mniejszego obciążenia procesora funkcja obliczana jest w cyklach co dziesięć sekund. Aby nastąpiło przynajmniej jedno wyliczenie wartości zadanej, podanie stanu wysokiego na wejście funkcji powinno trwać przynajmniej jedenaście sekund.

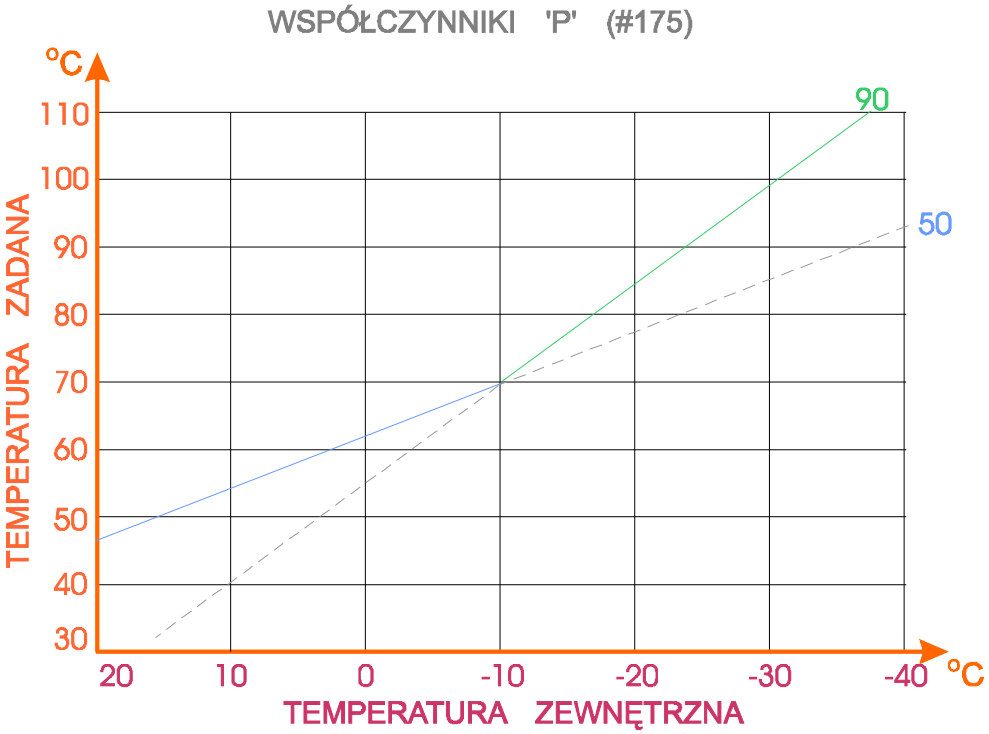
Opisywana funkcja reprezentuje zestaw krzywych które nadają się do większości obiektów. Czasem zachodzi konieczność użycia bardziej złożonych krzywych grzewczych. Na poniższym przykładzie pokazano jak wykonać krzywą grzewczą przełamaną w jednym miejscu. Program jest tak napisany że można zadać temperaturę zewnętrzna przy której nastąpi zmiana krzywych grzewczych. W instalacjach grzewczych z reguły pierwszą krzywą jest krzywa bardziej stroma. Przy wyższych temperaturach zewnętrznych pracują krzywe bardziej łagodne.

Wykorzystane rejestry:

WX0	temperatura zewnętrzna
WR0	numer krzywej grzewczej 1 bardziej stromej { 90 }
WR1	temperatura wyliczona z krzywej 1
WR2	numer krzywej grzewczej 2 łagodniejszej { 50 }
WR3	temperatura wyliczona z krzywej 2
WR4	rejestr pomocniczy różnica temperatur zadanych w miejscu przełamania
WR5	temperatura zewnętrzna przełamania krzywej { 300 } po przeliczeniu –21,1°C
WR6	temperatura zadana z krzywej przełamanej – brana do regulacji temperatury



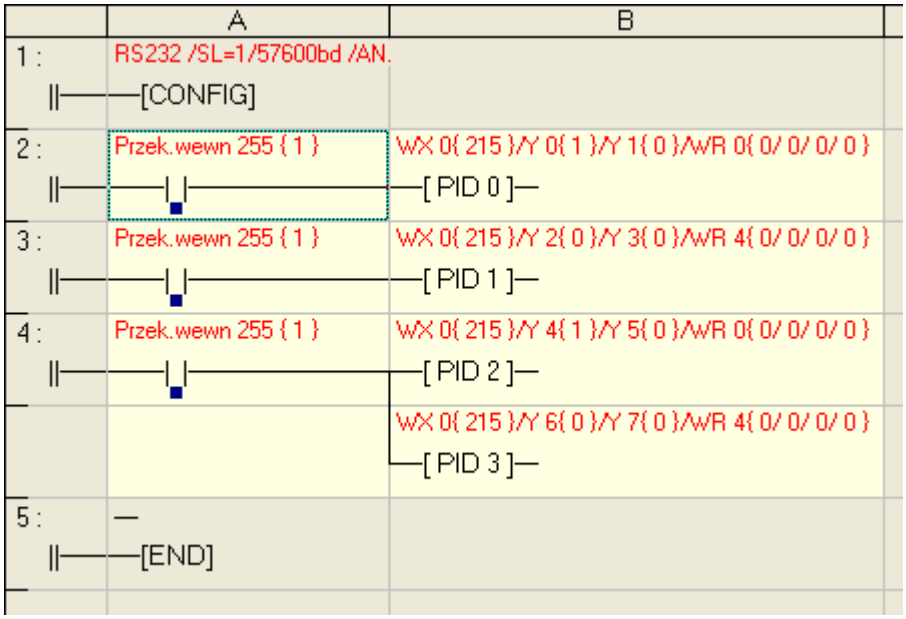
W linii drugiej następuje wyliczenie temperatur dwóch krzywych które zostaną ze sobą połączone. W linii 3 odbywa się realizacja przełamania i takiego połączenia krzywych aby nie nastąpił skok temperatury zadanej. Zaczynamy od warunku porównania temperatury zewnętrznej z wartością zadaną w której nastąpi przejście krzywej stromej na łagodniejszą. Jeżeli temperatura zadana WR5 jest mniejsza od zewnętrznej WX0 z relacji logicznej otrzymujemy stan niski i funkcja matematyczna umieszczona zaraz za relacją nie jest wykonywana. W następnej lini stan z relacji jest negowany co w efekcie daje nam logiczną jedynkę i w tym miejscu następuje wykonanie dwóch następnych funkcji, przypisania oraz różnicy argumentów. Przypisanie powoduje że do temperatury zadanej WR6 wpisywana jest wartość bezpośrednio z krzywej pierwszej WR1. Równocześnie cały czas wyliczana jest różnica pomiędzy temperaturą zadaną krzywej pierwszej i drugiej z równoczesnym zapisem do WR4. Różnica ta będzie wykorzystana w chwili przełączenia na krzywą łagodniejszą. Fragment tej gałęzi realizowany jest do chwili aż temperatura zewnętrzna WX0 nie przekroczy temperatury zadanej WR5. Wtedy będzie wykonywana tylko funkcja dodawania. Do temperatury zadanej z krzywej łagodniejszej WR3 będzie dodawana wyliczona wcześniej różnica z rejestru WR4. Od tej pory temperatura zadana WR6 będzie temperaturą z krzywej drugiej podniesioną o różnicę temperatur zadanych krzywych w punkcie przełamania. W ten sposób nastąpi łagodne przejście z jednej krzywej w drugą. Na przykładzie jest właśnie wykonywana podświetlona funkcja sumowania. Wygląd krzywej przy ustawieniach takich jak w programie i punkcie przełamania w temperaturze –10°C będzie wyglądał następująco:



----[PID] Regulator PID

	---[PID] kod 12 , 6		
1	Długość funkcji w bajtach	6	
2	Kod funkcji	12	
3	Nr. kanału temp. regulowanej WX_	0-63	
4	Nr. wy. Imp. Y_ ster. zaworem	0-127	
5	Nr. WR_ 4 kolejne nast. regulacyjne	0-252	
6	Nr. regulatora PID (0 - 13)	0-13	

Rozkaz umieszczony jest jako drugi lub następny w linii. Warunkiem pracy regulatora jest podanie stanu wysokiego na jego wejście. Można umieszczać funkcje PID jedna za drugą. Funkcje nie muszą być zawsze rozdzielone funkcją rozpoczynającą nową linię. Funkcja zawsze powinna być umieszczona na samym końcu i mogą występować za nią inne funkcje w gałęzi.



Na przykładzie podano użycie funkcji.

Mamy do dyspozycji 14 niezależnych regulatorów adresowanych od 0 do 13. Regulator PID jest niezbędny do obsługi technologii grzewczych i chłodniczych. Funkcja służy do regulacji wielkości analogowej (temperatura, przepływ, ciśnienie, poziom itp.) realizowanej przy pomocy sygnału dwustanowego otwierającego i zamykającego. Do procesu regulacji wykorzystywane jest pięć parametrów zawartych w kolejnych rejestrach WR:

- bieżąca wielkość regulowana (np. temperatura z WX0)
- wartość zadana (np. WR1)
- parametr trendu (np.WR2)
- współczynnik proporcjonalności (np.WR3)
- czas przerwy pomiędzy załączeniami (np.WR4 wartość od 1-255)

Podstawą pracy regulatora jest następujące równanie (przykład dla regulacji temperatury):

$$w=1/x*(Tz-Takt)+k*(Tpop-Takt)$$

gdzie:

Tz - temperatura zadana
Takt - temperatura aktualna
Tpop - temperatura z poprzedniego pomiaru
x - współczynnik proporcjonalności (wartość współczynnika może zmieniać się od 1 do 250)
k - parametr trendu (wartość współczynnika może zmieniać się od 0 do 50)
dla k = 0 trend wyłączony
w - stan wyjścia regulatora - wartość bezwzględna określa czas załączenia wyjścia
dla w>0 wyjście otwierające zostaje załączone
dla w=0 i w zadanej odchyłką histerezie wyjścia są wyłączone
dla w<0 wyjście zamykające zostaje załączone

Sposób regulacji polega na sprawdzeniu temperatury zadanej i zmierzonej (pierwsza część równania) oraz określenia znaku różnicy temperatury zadanej i mierzonej. Z odwrotności iloczynu współczynnika proporcjonalności i różnicy temperatury, wyliczana jest wielkość czasu załączenia lub wyłączenia sterowanego np. zaworu. Współczynnik proporcjonalności może się zmieniać od 0 do 250 co odpowiada zmianie wzmocnienia od największego do najmniejszego. Dla wartości współczynnika równego zero czas załączenia zawsze trwa 0.5 sekundy. Dla pozostałych wartości zmiana czasu załączenia mieści się w zakresie od 0.5 do 9.5 sekundy. Cykl regulacyjny jest dziesięcio sekundowy. Oznacza to że jeżeli czas załączenia wynosi np. 4 sekundy to czas przerwy wynosi 6 sekund. Zmieniając ten współczynnik należy pamiętać o tym że dobranie małej wartości współczynnika x spowoduje to że wzmocnienie układu będzie bardzo duże (odwrotność współczynnika).

Druga część równania odpowiedzialna jest za dynamikę procesu i kierunek zmian. Współczynnik k ustawiamy w zależności od inercji układu. Dla układów bardzo wolnych współczynnik powinien wahać się w granicach 0-2. Taka zasada regulacji pozwala reagować 'z wyprzedzeniem'. Na przykład dla odpowiednio dobranego współczynnika **k** regulator może reagować wcześniej, zanim temperatura mierzona przejdzie przez próg temperatury zadanej, przeciwdziałając w ten sposób zbyt dużemu przeregulowaniu obiektu. Regulator przeciwdziała zbyt gwałtownym zmianom np. temperatury (zwłaszcza rozruch) zabezpieczając układ przed wpadnięciem w oscylacje. Wtedy gdy nie chcemy korzystać ze współczynnika trendu wystarczy do nastawy wpisać wartość 0.
Ostatni parametr określa czas przerwy pomiędzy cyklami regulacji. może on się zmieniać od 10 sekund co stanowi podstawową wielkość cyklu regulacji do około 40 minut. co odpowiada wartościom nastawy równej 1 - 255. Nastawa określa ile cykli dziesięcio sekundowych należy odczekać do następnego otwarcia bądź zamknięcia np. zaworu. Ustawienie nastawy na wartość jeden powoduje pracę w cyklu 10 sek. Ustawienie nastawy na 2 powoduje odmierzenie 20 sekund czasu od pierwszego załączenia do następnego załączenia. W czasie przerwy pomiędzy załączeniami jest cały czas analizowany parametr trendu i obliczany za okres który upłynął od poprzedniego załączenia.
Temperaturę zadaną dla funkcji wpisujemy do pierwszej z czterech nastaw (np. WR0). Strefa nieczułości ustawiona jest na stałe i wynosi odpowiednio dla zakresów pomiarowych:

ZAKRES 0 DO 1000 °C	ZAKRES 0 DO 100.0 °C
±2 °C	±0.2 °C

Współczynnik **k** musimy zapisać do nastawy z numerem drugim (np.WR1), a **x** zapisujemy do nastawy trzeciej (np.WR2), czas opóźnienia załączenia ma być umieszczone w nastawie czwartej (npWR3).
Uwaga! Dla tego regulatora nastawy obsługiwane są cztery naraz (następują kolejno po sobie), dlatego w czasie programowania musimy zadbać aby nastawy z innych regulatorów i funkcji nie kolidowały z nimi. Podobna sytuacja jest z numerami regulatorów. Numery te nie mogą się powielać ponieważ regulatory będą się wzajemnie zakłócać

||---[COPY CL/DT/IMP] Kopiowanie rejestrów systemowych

Rozkaz umieszczony jest jako jedyny w linii. Funkcja służy do kopiowania wartości za pośrednictwem nastaw. Funkcję powinni wykorzystywać zawansowani programiści ponieważ przy jej pomocy można zmienić zawartość rejestrów systemowych które wpływają na pracę systemu operacyjnego sterownika. Format funkcji wygląda następująco:

NR.	FORMAT	NAZWA	UWAGI
1	00	BAJT POCZĄTKU DANYCH	
2	0Eh	KOD ROZKAZU	
3	XX	PAMIĘĆ ŹRÓDŁOWA	OPIS PONIŻEJ
4	XX	PAMIĘĆ DOCELOWA	OPIS PONIŻEJ
5	0/1/2/3/4/5/6/7	SPOSÓB PRACY	OPIS PONIŻEJ
6	FF(255)	BAJT KOŃCZĄCY	OPIS PONIŻEJ
7	FF(255)	BAJT KOŃCZĄCY	OPIS PONIŻEJ

Mamy do dyspozycji następujące możliwości przepisania danych:

KOPIOWANIE Z NASTAWY DO NASTAWY
- SPOSÓB PRACY **0**

3	XX - 0 DO 255	NASTAWA ŹRÓDŁOWA	ODCZYT 10 BIT
4	XX - 0 DO 255	NASTAWA DOCELOWA	ZAPIS 10 BIT
5	0	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie zawartości jednej nastawy do drugiej wraz z odchyłką.

KOPIOWANIE STANU WEJŚĆIA/WYJŚCIA ANALOGOWEGO DO NASTAWY
- SPOSÓB PRACY **1**

3	XX - 0 DO 63 LUB 64 DO 96	NUMER KANAŁU WEJŚCIOWEGO NUMER KANAŁU WYJŚCIOWEGO	ODCZYT 10 BIT ODCZYT 8 BIT
4	XX - 0 DO 255	NASTAWA DOCELOWA	ZAPIS 10 BIT
5	1	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie zawartości z wejścia lub wyjścia analogowego do wybranej nastawy. Wartość dla wejścia analogowego zawiera się od 0 - 1024. Wartość dla wyjścia analogowego zawiera się od 0 - 200.

KOPIOWANIE STANU WEJŚĆ/WYJŚĆ IMPULSOWYCH DO NASTAWY
- SPOSÓB PRACY **2**

3	XX - 0 DO 127 LUB 128 DO 255	NUMER PIERWSZEGO WEJŚCIA NUMER PIERWSZEGO WYJŚCIA	
4	XX - 0 DO 255	NASTAWA DOCELOWA	ZAPIS 10 BIT
5	2	SPOSÓB PRACY	
3	XX 1 - 8	ILIŚĆ KOPIOWANYCH WEJŚĆ/WYJŚĆ	

Wybranie tej opcji powoduje kopiowanie zawartości z wybranych wejść lub wyjść impulsowych do nastawy docelowej. Liczbę kopiowanych wejść lub wyjść podaje się w parametrze nr.5. Bity nastawy o wagach wyższych będą zerowane.

KOPIOWANIE NASTAWY DO OBSZARU WYJŚCIA ANALOGOWEGO
- SPOSÓB PRACY **3**

3	XX - 0 DO 255	NASTAWA ŹRÓDŁOWA	
4	XX -0 DO 32	NUMER ANALOGOWEGO KANAŁU WYJŚCIOWEGO	ZAPIS 8 BIT
5	3	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie zawartości z wybranej nastawy do wyjścia analogowego. Kopiowana wartość nie może przekraczać 200, co odpowiada 100% napięcia lub prądu wyjściowego. Kopiowanie wartości większej od 200 spowoduje obcięcie bardziej znaczących bitów.

KOPIOWANIE NASTAWY DO OBSZARU WYJŚĆ IMPULSOWYCH
- SPOSÓB PRACY **4**

3	XX - 0 DO 255	NASTAWA ŹRÓDŁOWA	
4	XX - 0 DO 127	NUMER PIERWSZEGO WYJŚCIA	
5	4	SPOSÓB PRACY	
3	XX 1 - 8	ILIŚĆ ZAPISYWANYCH WEJŚĆ/WYJŚĆ	

Wybranie tej opcji powoduje kopiowanie zawartości z nastawy do wybranego ciągu wyjść impulsowych. Jednorazowo można zapisać z nastawy od 1 do 8 wyjść impulsowych. Bardziej znaczące bity nastawy będą niewykorzystane.

KOPIOWANIE ZEGARA, DATY, LUB STATUSU DO WYBRANYCH NASTAW
- SPOSÓB PRACY **5**

3	XX - **0/1/2**	ODCZYT **0** - ZEGARA / **1** - DATY / **2** - STATUSU	
4	XX - 0 DO 253	NR.1 Z TRZECH KOLEJNYCH NASTAW	ZAPIS 3 BAJTY
5	5	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie czasu lub daty do wybranej nastawy. Wartości kopiowane są do trzech kolejnych nastaw. Format kopiowanych danych jest następujący:

dla zegara (0): XX - GODZINY, XX - MINUTY, XX - SEKUNDY,
dla daty (1): XX - DZIEŃ TYGODNIA, XX - DZIEŃ MIESIĄCA, XX - MIESIAC.

Liczby reprezentujące dzień tygodnia mają następujące znaczenie:

1 - poniedziałek
2 - wtorek
3 - środa
5 - piątek
6 - sobota
7 - niedziela

4 – czwartek

UWAGA: Aby odczyt zegara był możliwy w programie musi być wykonywana funkcja nr.82.

dla statusu (2):
Bity w słowie stanu mają następujące znaczenie:
bit 0 1 - magistrala w stanie transmisji/ 0 - magistrala jest wolna
bit 1 1 - zestawiona linia modemowa (gotowość do transmisji)/ 0 - modem nie połączony (zwarty JP12 na module procesora)
bit 2 1 - dane na magistrali szeregowej niezgodne z headerem
bit 3 nz
bit 4 nz
bit 5 nz
bit 6 nz
bit 7 impuls zegarowy 1 sekundowy (jedna pętla programowa, zwarty JP15)

ODCZYT WARTOŚCI STAŁEJ DO WYBRANEJ NASTAWY
- SPOSÓB PRACY **6**

3	XX	WARTOŚĆ OBOJĘTNA	
4	XX - 0 DO 255	NASTAWA DOCELOWA	ZAPIS 3 BAJTY
5	6	SPOSÓB PRACY	
6	XX - 0 do 15	GÓRNY BAJT WARTOŚCI STAŁEJ	XX*256
7	XX - 0 do 256	DOLNY BAJT WARTOŚCI STAŁEJ	

Dzięki tej opcji można zapisać wartość stałą do wybranej nastawy. Jako pierwszy podajemy górny bajt zapisywanej liczby (pozycja 05), jako drugi dolny bajt (pozycja 06). Do nastawy maksymalnie można zapisać wartość 4096 (uwzględniając odchyłkę). Wartości w nastawach wynosiłyby kolejno 15, i 256 (15*256+256=4096) wartość w nastawie wynosiła by 1023 i odchyłka nr.4.

ZAPIS WYBRANYCH NASTAW DO REJESTRÓW ZEGARA LUB DATY
- SPOSÓB PRACY **7**

3	XX - **0/1**	ZAPIS 0 - ZEGARA / 1 - DATY	
4	XX - 0 DO 125	NR.1 Z TRZECH KOLEJNYCH NASTAW	ZAPIS 3 BAJTY
5	7	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie wybranej nastawy do czasu lub daty zegara sterownika. Wartości kopiowane są z trzech kolejnych nastaw. Format kopiowanych danych jest taki jak dla sposobu pracy 5:

KOPIOWANIE KOMÓRKI STRONNICY ZEROWEJ DO NASTAWY
- SPOSÓB PRACY **8**

3	XX - 32 do 255	ADRES STRONNICY ZEROWEJ	8 BITÓW
4	XX - 0 DO 255	NASTAWA DOCELOWA	
5	8	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie określonej komórki stronnicy zerowej sterownika do nastawy. Funkcja ta pozwala na dokładne prześledzenie pracy sterownika oraz wykorzystanie parametrów zgromadzonych na stronnicy zerowej, do sterowania wykonywaniem programu technologicznego. Oczywiście należy posiadać wykaz przeznaczenia komórek systemowych rejestrów sterownika. Adres stronnicy zerowej rozpoczyna się od komórki 32 ponieważ poniżej znajduje się pamięć stosu procesora. Oto jedna z komórek stronnicy zerowej z której możemy odczytać aktualny stan klawiatury sterownika:

;adr.=32 - BITY BUFOR KLAWIATURY 32=255 ZADEN KLAWISZ NIE NACISNIETY

;BIT 00 = 0 NACISNIETY KLAWISZ SW1
;BIT 01 = 0 NACISNIETY KLAWISZ SW2
;BIT 02 = 0 NACISNIETY KLAWISZ SW3
;BIT 03 = 0 NACISNIETY KLAWISZ SW4 (1-SYSTEM COLUMBUS/ 0-micro PLC-ZWARTY SW8)
;BIT 04 = 0 NACISNIETY KLAWISZ SW5 (STATUS)
;BIT 05 = 0 NACISNIETY KLAWISZ SW6 (ZMNIEJSZANIE)
;BIT 06 = 0 NACISNIETY KLAWISZ SW7 (ZWIĘKSZANIE)
;BIT 07 = 0 NACISNIETY KLAWISZ SW8 (NASTĘPNY)

Naciśnięcie wybranego klawisza zeruje bit odpowiedzialny za jego stan. Klawisz nie naciśnięty powoduje że odpowiadający mu bit =1.

KOPIOWANIE NASTAWY KOMÓRKI STRONNICY ZEROWEJ
- SPOSÓB PRACY **9**

3	XX - 0 do 255	NASTAWA DOCELOWA	
4	XX - 20 DO 127	ADRES STRONNICY ZEROWEJ	8 BITÓW
5	9	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie określonej komórki nastawy do stronnicy zerowej sterownika. Funkcja ta pozwala na wykorzystanie systemu operacyjnego sterownika przy sterowaniu stanem w jakim znajduje urządzenie. Zalecamy aby opcji tej nie korzystali początkujący programiści ponieważ złe jej użycie może doprowadzić do niestabilności pracy sterownika.

ODCZYT WARTOŚCI Z NASTAWY O ADRESCIE RELATYWNYM
- SPOSÓB PRACY **10**

3	XX - 0 do 255	NASTAWA Z ADRESEM RELATYWNYM	
4	XX - 0 DO 255	NASTAWA DOCELOWA	8 BITÓW
5	10	SPOSÓB PRACY	

Wybranie tej opcji powoduje kopiowanie wartości z nastawy której adres jest umieszczony w NASTAWIE Z ADRESEM RELATYWNYM. Po odczytaniu adresu pobierana jest wartość z nastawy wskazanej przez adres i wpisywana do nastawy docelowej. Np. sekwencja :

...
3 = 10
4 = 200
5 = 10
...
spowoduje odczytanie wartości z nastawy nr. 10 i zapisanie jej do nastawy 200.

||---[LIGHT] Sterowanie scenami świetlnymi

Rozkaz umieszczony jest jako jedyny w linii. Funkcja służy do sterowania oświetleniem. Format funkcji wygląda następująco:

ADRES	FORMAT	NAZWA	UWAGI
00	00	BAJT POCZĄTKU DANYCH	
01	13h	KOD ROZKAZU	
02	XX - 0 DO 127 128 DO 255	ADRES 2 WEJŚĆ IMPULSOWYCH LUB 2 WYJŚĆ IMPULSOWYCH	1 ZAŁ/ZWIĘKSZ. 2 WYŁĄCZAJĄCE
03	XX -0 DO 127 128 - 255	ADRES POCZĄTKU WEJŚĆ / WYJŚĆ IMPULSOWYCH ZAŁACZANYCH	
04	XX - 0 DO 250	NASTAWA Z PIERWSZĄ SCENĄ	
05	**XX** - 1 DO 8 **00** - **16** + (1 DO 8) - **32** + (1 DO 8) -	ILOŚĆ SCEN I WYJŚĆ IMPULSOWYCH POGRAMOWO WYŁĄCZA WYJŚCIA PROGRAMOWO ZAŁ.AKTUALNĄ SCENĘ PROGRAMOWO ZAŁ.NASTĘPNĄSCENĘ	
06	XX - 0 DO 255	NASTAWA WSKAŻNIKOWA	POMOCNICZA

Do funkcji wprowadza się adresy wejść/wyjść impulsowych z których pierwsze załącza aktualną scenę. Przy powtórnym naciśnięciu zaświecana jest następna scena i tak do numeru sceny zadanej w parametrze 05. Potem znowu jest odczytywana scena pierwsza. Wyłączenie następuje poprzez kolejne wejście/wyjście impulsowe. Ilość scen jest jednocześnie ilością obsługiwanych wyjść impulsowych. Parametr 06 jest nastawą pomocniczą przechowującą aktualny adres sceny, stan załączenia i stan wejścia/wyjścia załączającego.
Funkcją można sterować modyfikując parametr 05 i wywołując funkcję. Jeżeli będzie on równy zero sterowane wyjścia będą wyłączone. Jeżeli do ilości scen dodamy wartość 16 możemy złączyć aktualną scenę. Jeżeli do ilości scen dodamy wartość 32 możemy załączyć następną scenę z pamięci.
Uwaga: Przy sterowaniu programowym należy wejść do funkcji tylko raz na przełączenie. Ciągłe wchodzenie w funkcję powoduje ciągłe wykonanie poleceń i stratę czasu.

----[MASTER WRITE] Zapis rejestru wewnętrznego w urządzeniu przyłączonym do sterownika

	----[MASTER WRITE] kod 20 , 9		
1	Długość funkcji w bajtach	9	
2	Kod funkcji	20	
3	Adres zapisywanego urządzenia MODBUS	1-255	
4	Rodzaj zapisywanych danych MODBUS	XX	05 – Zapis przekaźnika wyjściowego 06 – Zapis rejestru WR
5	Górny bajt adresu zapisu danych MODBUS	XX	
6	Dolny bajt adresu zapisu danych MODBUS	XX	
7	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
8	Górny bajt adresu danych sterownika	XX	
9	Dolny bajt adresu danych sterownika	XX	

Funkcja pozwala zapisać dowolny rejestr w urządzeniu zewnętrznym przyłączonym do sterownika przez magistralę szeregową. Przy pomocy tej funkcji można zapisać tylko jeden rejestr. Pierwszy parametr funkcji określa do którego urządzenia się odwołujemy. Zapis dokonywany jest za pośrednictwem protokołu MODBUS RTU. Prędkość i parametry transmisji należy wcześniej ustawić funkcją [CONFIG]. Przy zapisie możemy wykorzystać następujące rodzajów rejestrów:

05h	Wymuszenie pojedynczej cewki (zapis wyjścia cyfrowego)
06h	Zapis do pojedynczego rejestru

W następnych dwóch rejestrach podajemy adres pamięci urządzenia pod który następuje zapis.
Kolejny parametr określa z jakiego rejestru sterownika następuje przesłanie danych. Najpierw podajemy rodzaj rejestru źródłowego a następnie jego adres.
Jeśli urządzeniem zewnętrznym był by drugi sterownik microPLC to rejestry do których może nastąpić zapis wyglądają następująco:

```
;--- ROZKAZ 06 – ZAPIS REJESTRÓW

; ADRES 0x0000 - 0x00FF   - REJESTRY WEWNĘTRZNE WR
; ADRES 0x0100 - 0x013F   - REJESTRY TIMEROW TD
; ADRES 0x0200 - 0x021F   - WYJŚCIA ANALOGOWE WY
; ADRES 0x0300 - 0x033F   - WEJŚCIA ANALOGOWE WX

;--- ROZKAZ 05 – ZAPIS SYGNAŁÓW IMPULSOWYCH

; ADRES 0x0000 - 0x007F   - PRZEKAZNIKI WYJSCIOWE Y
; ADRES 0x0100 - 0x01FF   - PRZEKAZNIKI WEWNETRZNE R
; ADRES 0x0200 - 0x027F   - REJESTRY AUTO-MANUAL MX
; ADRES 0x0300 - 0x037F   - WEJŚCIA IMPULSOWE X
```

Jak widać istnieje możliwość zapisu rejestrów wejść analogowych oraz wejść impulsowych. Zapisu można dokonać pod adresy których nie obejmują fizyczne wejścia sterownika. Zapisu do wszystkich rejestrów wejść można dokonać gdy urządzenie jest przestawione w stan symulacji.

---[MASTER CALL] Inicjacja wymiany danych z innymi urządzeniami przyłączonymi do sterownika

	----[MASTER CALL] kod 21 , 7		
1	Długość funkcji w bajtach	7	
2	Kod funkcji	21	
3	Adres wywoływanego urządzenia MODBUS	1-255	
4	Rodzaj odczytywanych danych MODBUS	1 - 3	01 – Odczyt rejestrów wewnętrznych R 02 – Odczyt wejść impulsowych X 03 – Odczyt rejestrów WR
5	Górny bajt adresu danych MODBUS	XX	
6	Dolny bajt adresu danych MODBUS	XX	
7	Ilość odczytywanych danych	1-100	

Rozkaz umieszczony jest jako drugi lub następny w linii. Dzięki niemu za pośrednictwem magistrali szeregowej można odczytać dane z urządzeń podłączonych do sterownika. Wymiana danych odbywa się zgodnie z protokołem MODBUS RTU. Funkcja ta inicjuje wymianę danych z urządzeniem zewnętrznym za pośrednictwem ciągu bajtów wysyłanych przez magistralę szeregową. Aby urządzenie mogło odpowiedzieć na wywołanie musi posiadać adres zgodny z adresem umieszczonym w trzeciej linii funkcji „Adres wywoływanego urządzenia MODBUS”. Adres musi zawierać się w zakresie od 1 do 255. Następnym parametrem funkcji jest „Rodzaj odczytywanych danych MODBUS”. Parametr ten określa z jakich rejestrów urządzenia zewnętrznego chcemy odczytać dane. W standardzie MODBUS istnieje możliwość odczytu następujących standardowych typów rejestrów:

01	Odczyt stanu cewek (odczyt wyjść cyfrowych)
02	Odczyt wejść cyfrowych
03	Odczyt rejestrów wewnętrznych
04	Odczyt wejść analogowych
07	Odczyt statusu
17	Odczyt ID jednostki slave
20	Odczyt zawartości rejestrów w pamięci rozszerzonej
23	Odczyt/zapis grupy 4 rejestrów
24	Odczyt kolejki FIFO

Powyższa specyfikacja dotyczy protokołu uniwersalnego dla urządzeń dostarczonych przez dowolnego producenta. Zwykle producent urządzenia dokładnie specyfikują co znajduje się pod konkretnymi rodzajami i adresami rejestrów.

Poniżej pokazano rodzaje rejestrów jeśli wymiana danych następowała by z drugim sterownikiem microPLC.

```
;--- ROZKAZ 01 - ODCZYT PRZEKAZNIKOW

; ADRES 0x0000 - 0x007F   - Y - PRZEKAZNIKI WYJSCIOWE
; ADRES 0x0100 - 0x01FF   - R - PRZEKAZNIKI WEWNETRZNE
; ADRES 0x0200 - 0x027F   - MX- AUTO/MAN
```

; ADRES 0x0300 - 0x037F - MY- AUTO/MAN

;--- ROZKAZ 02 - ODCZYT WEJSC IMPULSOWYCH

; ADRES 0x0000 - 0x007F - X - WEJSCIA IMPULSOWE

;--- ROZKAZ 03 - ODCZYT REJESTROW WEWNETRZNYCH - 16 BIT

; ADRES 0x0000 - 0x00FF - WR - REJESTRY WEWNETRZNE
; ADRES 0x0100 - 0x013F - TD - REJESTRY TIMEROW
; ADRES 0x0200 - 0x021F - WY - WYJSCIA ANALOGOWE

;--- ROZKAZ 04 - ODCZYT WEJSC ANALOGOWYCH

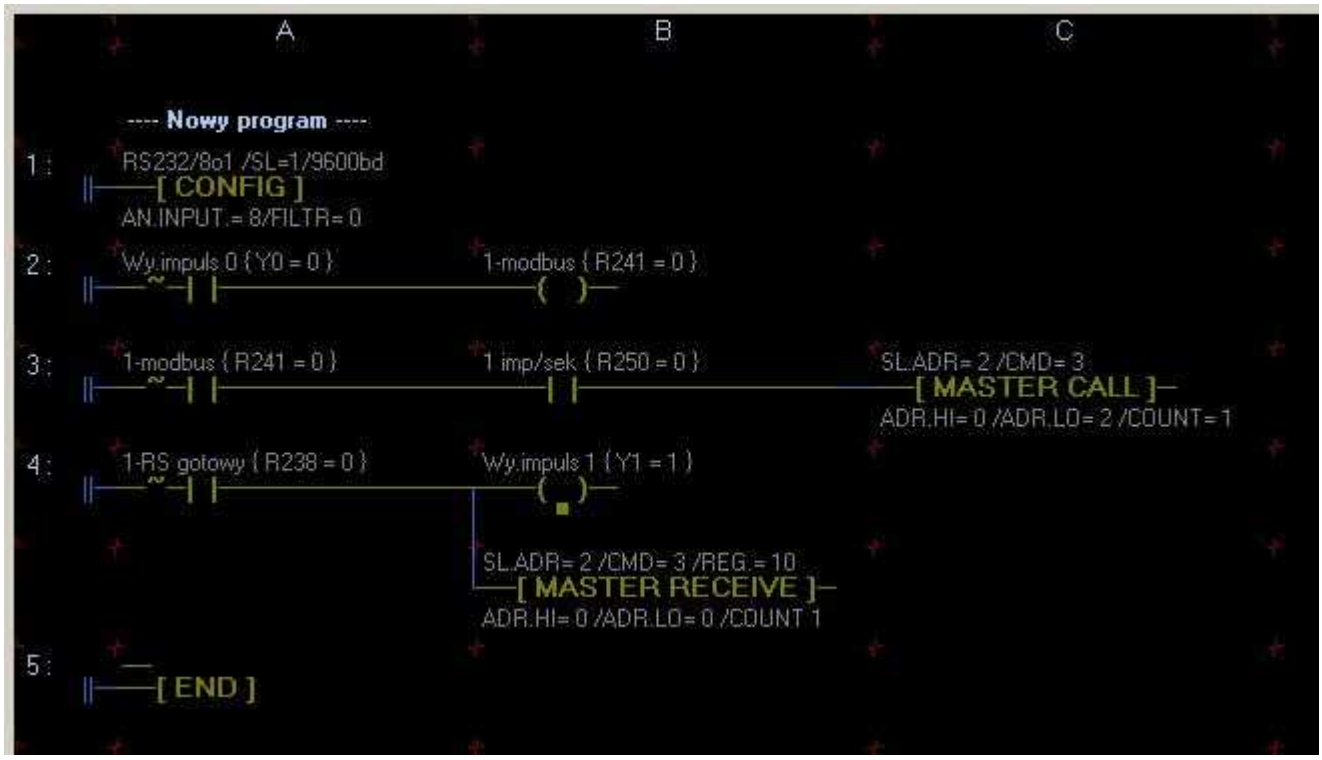
; ADRES 0x0000 - 0x003F - WX - WEJSCIA ANALOGOWE

Aby wywołanie było kompletne należy po za numerem urządzenia i typem rejestru podać jeszcze adres z którego chcemy odczytać wartość. Jak pokazano wyżej wybierając typ rejestru 01 możemy obok przekaźników wyjściowych Y odczytać również np. stan przekaźników wewnętrznych „R". Ich stan znajduje się w pamięci od adresu 0100 heksadecymalinie (256 dziesiętnie).
Na koniec musimy podać jeszcze ilość bajtów które chcemy jednorazowo odczytać. Parametr ten powinien zawierać się w zakresie od 1do 100.
Funkcja wysyła dane dopiero wtedy gdy na jej wejściu pojawi się stan wysoki. W tym momencie następuje wysłanie na zewnątrz ciągu danych. Pisząc program należy pamiętać o tym żeby wywołania nie następowały zbyt często ponieważ urządzenie zewnętrzne może nie zdążyć odpowiedzieć. Wysłanie ostatniego bajtu wywołania spowoduje ustawienie przekaźnika wewnętrznego R237. Przekaźnik ten może być tylko odczytywany a jego stan = 1 oznacza że nastąpiło wywołanie przez funkcje [MASTER CALL]. Teraz należy czekać aż wywoływane urządzenie odpowie. Jeśli odpowiedź przyjdzie to przekaźnik wewnętrzny R238 zmieni się ze stanu zero na jeden. Oznacza to że dane które wysłało urządzenie zewnętrzne są gotowe do odczytu. Odczyt przeprowadzamy za pośrednictwem następnej funkcji [MASTER RECEIVE]. Jeśli dane nie nadchodzą można co pewien czas ponawiać wywołanie. Po pewnym czasie jeśli urządzenie nie odpowiada można załączyć diodę LED na panelu przednim sterownika co poinformuje o braku łączności.

----[MASTER RECEIVE] Odczyt odebranych danych z bufora magistrali szeregowej

	----[MASTER RECEIVE] kod 22 , 8		
1	Długość funkcji w bajtach	8	
2	Kod funkcji	22	
3	Adres urządzenia zewnętrznego MODBUS	1-255	
4	Rodzaj pamięci urządzenia zewnętrznego MODBUS	XX	
5	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Impuls. zegara TD +SS kod 5 adr 0-63 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów sterownika możliwych do użycia i ich adresy
6	Górny bajt adresu danych sterownika	XX	
7	Dolny bajt adresu danych sterownika	XX	
8	Ilość zapisywanych danych do sterownika	1-100	

Rozkaz umieszczony jest jako drugi lub następny w linii. Jak wcześniej wspomniano dzięki niemu możemy pobrać dane które przesłało urządzenie zewnętrzne. Aby można było odczytać dane musimy podać numer urządzenia z jakiego zostały one przesłane oraz jaki jest ich typ. Następne parametry funkcji decydują o tym gdzie przesłane dane zostaną umieszczone w sterowniku. Parametr w linii piątej decyduje do jakiego rejestru wewnętrznego zostaną zapisane dane z bufora szeregowego magistrali szeregowej. Dzięki temu parametrowi można przesyłać między sobą dane różnych typów. Na przykład dane pobrane z wejść impulsowych można zapisać w rejestrach WR sterownika.
Parametr szósty i siódmy określa pod jaki adres w pamięci sterownika nastąpi zapis danych. Ostatni parametr mówi o tym ile odebranych danych należy zapisać do rejestrów sterownika. Należy pamiętać żeby ilość danych odebranych nie był większa od ilości danych przyjętych ponieważ spowoduje to błędne zapisy do rejestrów.
Znakiem do odczytu danych z magistrali szeregowej jest przekaźnik wewnętrzny R238 którego wysoki stan mówi o tym że przyjęto odpowiedź dla funkcji [MASTER RECEIVE] i dane w buforze są gotowe do odczytu.
Prześledźmy jak wygląda przykładowy program pozwalający odczytać dane z urządzenia zewnętrznego. Dla przykładu pokazano program odczytujący dane z termostatu elektronicznego firmy Keveno. Program odczytuje dane z rejestru przechowującego temperaturę zadaną pomieszczenia.
Program rozpoczyna się od funkcji konfiguracyjnej w której definiujemy nowe parametry transmisji dla magistrali szeregowej. Termostat posiada „na sztywno" zdefiniowane następujące parametry magistrali 9600/8/o/1. Funkcja konfiguracyjna definiuje takie właśnie parametry. W linii drugiej przestawiamy protokół sterownika na Modbus RTU. W tym momencie jesteśmy gotowi do uruchomienia wymiany danych. Inicjacja następuje w chwili załączenia z klawiatury sterownika wyjścia przekaźnikowego Y0. Linia trzecia wykorzystuje rejestr wewnętrzny R250 który co sekundę zmienia swój stan na jeden przez jeden cykl programu. Lina ta w efekcie co sekundę wysyła wywołanie do urządzenia zewnętrznego przy pomocy funkcji [MASTER CALL]. Wywołanie skierowane jest do urządzenia zewnętrznego o adresie = 2. Rejestr odczytywany jest przy pomocy rozkazu 03 – co definiuje pamięć rejestrów typu „holding register". Adres rejestru równy jest 2. Ilość odczytywanych danych jest równa 1.
Linia czwarta odpowiedzialna jest za odczyt danych które właśnie nadeszły. W chwili gdy termostat przesłał wszystkie dane następuje zmiana stanu przekaźnika wewnętrznego R238. Stan wysoki tego przekaźnika oznacza że wszystkie dane zostały odebrane bezbłędnie i są gotowe do odczytu. Jeśli przekaźnik ten nie zmieni stanu to znaczy że dane nie nadeszły lub są błędnie odebrane. Stan tego rejestru dodatkowo sygnalizowany jest przez przekaźnik wyjściowy sterownika o adresie Y2. Odbiór danych realizowany jest przy pomocy funkcji [MASTEWR RECEIVE]. W chwili kiedy program napotka funkcję odbioru danych kasuje stan przekaźnika wewnętrznego R238. Funkcja przyjmująca dane skieruje je do rejestru o kodzie = 10 to znaczy do WR, pod adres 0. W ten sposób co sekundę mamy uaktualnioną wartość temperatury zadanej pomieszczenia. Zatrzymanie transmisji danych kończymy zmieniając stan przekaźnika wyjściowego Y2.



||---[MODEM] Obsługa modemu telefonicznego

Rozkaz umieszczony jest jako jedyny w linii. Format funkcji wygląda następująco:

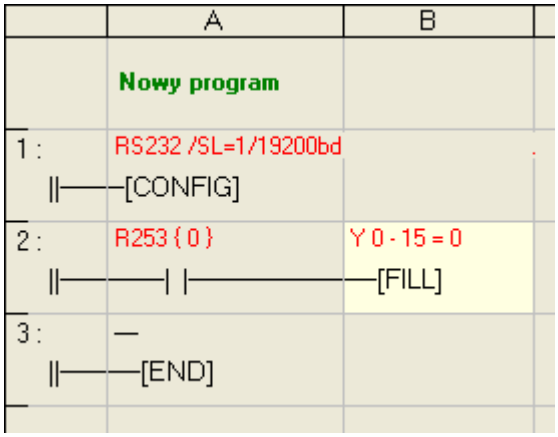
ADRES	FORMAT	NAZWA	UWAGI
00	00	BAJT POCZĄTKU DANYCH	
01	17h	KOD ROZKAZU	
02	XX - 0 DO 127	NUMER PIERWSZEJ NASTAWY	
03	XX - 0 DO 127	LICZBA WYSYŁANYCH Z NASTAW KODÓW ASCII	
04	XX - 0 DO 127 128 - 255	NUMER WYJŚCIA IMPULSOWEGO	ZAŁĄCZANE PO WYKONANIU FUNKCJI
05	0/1	SPOSÓB PRACY	**0**=BAJTY Z NASTAWY **1**=WYSŁANIE ROZKAZU **ATH**
06	FF(255)	BAJT KOŃCZĄCY	

Funkcja służy do komunikacji z modemem telefonicznym. Dzięki niej sterownik może samodzielnie wybrać wcześniej ustawiony numer telefonu i po połączeniu się z drugim modemem powiadomić np. o powstałym alarmie. Do prawidłowej pracy funkcji potrzebne jest podanie numeru nastawy od której zaczyna się ciąg znaków do wysłania. W kolejnych nastawach zapisywany ma być ciąg znaków który będzie transmitowany do modemu. Kolejny parametr to ilość znaków do wysłania. Określa on w ilu nastawach zawarte są znaki które będą wysyłane do modemu. Następny parametr to numer wyjścia impulsowego które przyjmuje stan wysoki po wykonaniu funkcji. Ostatni parametr określa czy znaki będą transmitowane z nastaw (wartość 0) czy będzie to rozkaz odłożenia słuchawki (wartość 1).

----[FILL] Wypełnienie obszaru jednakową wartością

	----[FILL] kod 24 , 6		
1	Długość funkcji w bajtach	6	
2	Kod funkcji	24	
3	Wejścia impulsowe X kod 0 adr 0-127 Wyjścia impulsowe Y kod 1 adr 0-127 Przek. wewnętrzny R kod 2 adr 0-255 A/MAN impulsowy M kod 4 adr 0-127 Wejścia analogowe WX kod 8 adr.0 – 63 Wyjścia analogowe WY kod 9 adr.0 – 32 Wartości zadane WR kod 10 adr.0 - 255		Rodzaje rejestrów możliwych do użycia i ich adresy
4	Adres pierwszej zmiennej	XX	Adres użytego rejestru
5	Zapisywana wartość	XX	Wartość zależna od użytego rejestru
6	Adres ostatniej zmiennej	XX	Adres użytego rejestru

Rozkaz umieszczony jest jako drugi lub następny w linii. Warunkiem pracy regulatora jest podanie stanu Rozkazem tym kończymy linę w której będą występowały operacje logiczne. Funkcja ta jest ostatnią funkcją gałęzi. Funkcją tą możemy zmienić zawartość pewnego obszaru wewnątrz sterownika. Zmienić możemy stan wyjść impulsowych Y, przekaźników wewnętrznych R, rejestrów wewnętrznych WR, oraz wyjść analogowych WY. Do swojej pracy funkcja potrzebuje czterech parametrów. Pierwszy to rodzaj modyfikowanego obszaru pamięci (wejścia , wyjścia lub rejestry wewnętrzne). Następnie adres początku zapisywanego obszaru, potem wartość jaką mamy zapisać i adres końca zapisywanego obszaru.
Zobaczmy praktyczne wykorzystanie funkcji



W przykładzie wykorzystano funkcję ---[FILL] do wyłączenia pierwszych 16 przekaźników wyjściowych sterownika przez zapisanie do wyjść impulsowych Y0 do Y15 stanu = 0. Wykonanie funkcji nastąpi tylko raz po załączeniu sterownika, ponieważ warunek wykonania tej funkcji poprzedzony jest odczytem rejestru R253 który przyjmuje stan 1, tylko wtedy gdy program technologiczny wykonywany jest po raz pierwszy. Po zakończeniu pierwszego przejścia przez program jego stan rejestru R253 jest zawsze niski.

||---[RANGE] Zmiana zakresu pomiarowego

Rozkaz umieszczony jest jako jedyny w linii. Format funkcji wygląda następująco:

NR	FORMAT	NAZWA	UWAGI
1	00	BAJT POCZĄTKU DANYCH	
2	50h	KOD ROZKAZU	
3	XX - 0 DO 63 LUB 0 DO 127	ADRES KANAŁU WEJŚCIOWEGO LUB NASTAWY	
4	XX	MNOŻNIK	WARTOŚĆ < 65
5	XX	DZIELNIK	WARTOŚĆ != 0
6	**M**XXX**KKKK**	STEROWANIE **M**INUSEM I **K**ROPKAMI	
7	0/1	ZMIANA ZAKRSU 0 = WEJŚCIA ANALOGOWE 1 =-WARTOŚCI ZADANE	

Funkcja służy do zmiany formatu wyświetlania zmierzonych wartości analogowych WX lub wartości parametrów zadanych WR na wyświetlaczu sterownika. Wartość wyświetlana może być zmieniona w wyniku pomnożenia jej przez wybraną wartość a następnie przedzielenie wcześniejszego wyniku. Przy przeliczaniu należy pamiętać aby wartość wyniku nie przekraczała liczby 65535, ponieważ do dyspozycji są tylko rejestry szesnasto-bitowe. Wyliczane liczby są wartościami całkowitymi. Sterowanie minusem i znakiem dziesiętnym odbywa się następująco:

STEROWANIE MINUSEM I KROPKAMI	ZAKRES	MIEJSCE KROPKI
00 dzies./ 00hex	0 do 1000	1000
01 dzies./ 01hex	0 do 1000	1000.
02 dzies./ 02hex	0 do 1000	100.0
03 dzies./ 04hex	0 do 1000	10.00
08 dzies./ 08hex	0 do 1000	1.000
129 dzies./ 81hex	-500 do 500	500.
130 dzies./ 82hex	-500 do 500	50.0
132 dzies./ 84hex	-500 do 500	5.00
136 dzies./ 88hex	-500 do 500	.500

Ustawienie zakresu pomiarowego wejścia analogowego 0 – 1,600 odbywa się przez wpisanie kolejnych wartości od nr.4: 16, 10, 08, 00. Zobaczmy jak wykonywane są przeliczenia:
Mierzona wartość 1000 ma odpowiadać zakresowi ciśnienia 1,600 Mpa
wyliczamy pierwszą wartość 1000 * 16 = 16 000

wyliczamy drugą wartość 16000 : 10 = 1600
do otrzymanej wartości dodajemy kropkę na 4 miejscu 1,600
zmieniany zakres dotyczy wejścia analogowego 00
Aby można było mierzyć ciśnienie należy zamówić sterownik z wejściem analogowym 4 –20 mA. Podłączenie do niego przetwornika ciśnienia o zakresie 0 – 1,6 Mpa da nam poprawny odczyt na wyświetlaczu sterownika.

Dla zakresu wartości zadanych 0 – 600,0 wpisujemy 06, 01, 02, 01.
Zadana wartość 1000 ma odpowiadać zakresowi temperatury 600 °C
wyliczamy pierwszą wartość 1000 * 6 = 6 000
wyliczamy drugą wartość 16000 : 1 = 600
do otrzymanej wartości dodajemy kropkę na 2 miejscu 600,0
zmieniany zakres dotyczy wartości zadanej WR 01
Aby można było mierzyć taki zakres temperatur należy zamówić sterownik z wejściem na zakres 0 - 600 °C. Pomiar może odbywać się czujnikiem temperatury PT1000. Pomiar takim czujnikiem nie wymaga żadnych dodatkowych urządzeń. Czujnik podłączamy bezpośrednio do wejścia analogowego sterownika.

Następny przykład pokazuje jak zmierzyć temperaturę w zakresie od –50,0 do 50,0 °C. Zamawiamy sterownik z typowymi wejściami czujników rezystancyjnych PT1000 na zakres od 0 do 100 °C. Do wejścia za pomocą którego chcemy dokonać pomiaru dołączamy szeregowo z czujnikiem rezystor dostarczony ze sterownikiem. Wartość tego rezystora wynosi 194 ohm co symuluje temperaturę 50,0 °C. W ten sposób sterownik przy temperaturze zmierzonej przez czujnik 0 °C pokazuje 50,0 °C. Teraz aby pomiar był właściwy musimy odpowiednio skalibrować wejście pomiarowe. kalibracja będzie wyglądają następująco 06, 01, 01, 130.
wyliczamy pierwszą wartość 1000 * 1 = 1000
wyliczamy drugą wartość 1000 : 1 = 1000
otrzymaną wartość przesuwamy w dół o 50,0 i dodajemy kropkę
na 2 miejscu tak jak we wcześniej przedstawionej tabeli 130
zmieniany zakres dotyczy wartości wejścia analogowego 01

W przypadku liczb ujemnych należy pamiętać że minus pojawia się zamiast pierwszej wyświetlanej cyfry.

||---[CONFIG] Konfiguracja sterownika

Rozkaz umieszczony jest na początku programu jako jedyny w linii. Format funkcji wygląda następująco:

	||---[CONFIG] kod 82 , 7		
1	Długość funkcji w bajtach	7	
2	Kod funkcji	82	
3	0=RS485/8n1, 1=RS232/8n1, 2=RS485/8o1, 3=RS232/8o1 6=RS485/8e1, 7=RS232/8e1	00000**xxx**	Wartość interpretowana bitowo
4	Numer sterownika na magistrali danych	1-255	Numer slave/rejestr WR z nr.slave
5	Prędkość transmisji danych/ numer rejestru WR z prędkością transmisji danych	0-255	
6	Ilość przetwarzanych kanałów analogowych	0-63 (+128)	Dodanie 128 powoduje załączenie cyfrowej filtracji wejść analogowych
7	Opóźnienie transmisji długich bloków	0-10	

Funkcja ta wykorzystywana do konfiguracji podstawowych parametrów sterownika. Pierwszym podstawowym parametrem funkcji jest konfiguracja rodzaju i parametrów magistrali szeregowej, używanej podczas wymiany danych ze sterownikiem. Parametr określa z jakiego typu magistrali korzystamy oraz określamy czy będziemy korzystać z bitów parzystości.

Bity w słowie sterującym mają następujące znaczenie:

bit 0 **A**; 1 – RS232/ 0 – RS485
bit 1 **B**; 1 – Transmisja z wykorzystaniem bitów parzystości / 0 – Transmisja bez bitów parzystości 8n1

bit 2 **C**; 1 – bit parzystości równy 1 (EVEN – parzysty) / 0 bit parzystości równy 0 (ODD – nieparzysty)
bit 3 nz
bit 4 nz
bit 5 nz
bit 6 nz
bit 7 nz

Drugi parametr definiuje jaki adres posiada nasz sterownik. Adres ten jest konieczny do tego aby dane wysyłane przez komputer były przyjmowane tylko przez nasze urządzenie. Przy programowaniu adresu slave i ustalaniu prędkości transmisji trzeba postępować bardzo ostrożnie ponieważ po wpisaniu nowego numeru sterownika lub prędkości transmisji tracimy łączność z urządzeniem. Aby uzyskać ponowną transmisję należy wywołać urządzenie z jego nowym numerem i na nowych parametrach transmisji. Numer sterownika określany jest przez wpisanie numeru od 1 do 16. Zapisanie numeru 0 spowoduje że sterownik nie będzie reagował na żadne dane przychodzące magistralą szeregową.
Jeżeli bit 0 (A) będzie ustawiony, sterownik przyłączy magistralę RS232 do procesora. Gdy ten bit będzie wymazany, transmisja będzie odbywała się po magistrali RS485. Bit 1 (B) określa czy podczas transmisji będzie wykorzystywany bit parzystości. Jeśli bit 1 ustawimy na jeden, przy wymianie danych będzie sprawdzany stan bitu parzystości. Jeśli jego stan jest równy zero wymiana danych będzie odbywać się bez sprawdzania bitu parzystości. Bit 2 (B) decyduje o typie bitu parzystości. Jeśli ustawimy jeden podczas transmisji jest sprawdzane czy bit parzystości ma stan wysoki(EVEN – parzysty). Dla stanu niskiego będzie sprawdzane czy bit parzystości będzie miał stan niski0 (ODD – nieparzysty).
Brak tej funkcji w programie powoduje że sterownik przestawi się na pracę RS485 a jego adres wewnętrzny będzie równy 1, magistrala szeregowa samoczynnie ustawi się na parametry 8N1/2400 bodów, przetwarzane będą 64 kanały analogowe.
Wybór prędkości transmisji następuje przez wpisanie odpowiedniej wartości na pozycję nr.5. Oto wartości które odpowiedzialne są za poszczególne prędkości transmisji:

255=57600 bodów (prędkość preferowana dla łączności internetowej dla łącza symetrycznego 8 MB),,
253=19200 bodów (prędkość preferowana dla łączności internetowej dla łącza symetrycznego 1 MB),
250=9600 bodów (prędkość preferowana dla łączności modemowej GSM),
232=2400 bodów.

W sterowniku zdefiniowano specjalny wskaźnik sygnalizujący zajętość magistrali danych. Jest nim przekaźnik wewnętrzny R239, który jest ustawiany wtedy gdy do sterownika dotrze dowolny bajt który będzie przesłany zgodnie z ustawionymi parametrami magistrali szeregowej. Wymazanie tego znacznika nastąpi wtedy gdy dane będą skierowane do naszego sterownika i obsłuży on transmisję.

Pozycja 6 tabeli odpowiedzialna jest za ilość odczytywanych kanałów analogowych. Odczyt kanałów analogowych absorbuje czas sterownika i jeżeli nie potrzebujemy mierzyć na wszystkich wejściach to nie powinniśmy inicjować tak dużej liczby kanałów wejściowych. Ilość dostępnych kanałów dla sterownika microPLC wynosi 8. Pozostałe kanały dostępne są w systemie Columbus. Nie używane kanały analogowe mogą być traktowane jako rejestry wewnętrzne. Odczyt kanałów analogowych może odbywać się na dwa sposoby. Pierwszy to przetwarzanie bezpośrednie. Wartość na wejściu analogowym jest mierzona i przetwarzana bezpośrednio do rejestrów wejść analogowych WX. Drugi sposób to przetwarzanie z cyfrową filtracją zakłóceń. Opcja ta przydaje się w większości przypadków realizowanych technologii. Rzeczywiste instalacje prawie zawsze obarczone są zakłóceniami które powodują nieprawidłową pracę całej instalacji. Filtracja cyfrowa polega na tym że kolejne cztery pomiary wczytywane są do pamięci sterownika a następnie ich suma jest dzielona przez cztery. Wynik zapisywany jest do rejestrów wejść analogowych. Aby uruchomić przetwarzanie z filtracją należy do ilości kanałów dodać wartość 128. Jeżeli chcemy przetwarzać osiem kanałów analogowych z załączoną filtracją należy wpisać wartość 8+128=136.

Ostatnia pozycja w tabeli odpowiada tylko za prędkość przesyłania danych zgromadzonych w buforach sterownika. Normalnie parametr ten jest równy zero. Dla takiej wartości odstępy pomiędzy wysyłanymi blokami danych są minimalne i dane wysyłane są z maksymalną prędkością. Wpisanie wartości równej jeden wytworzy opóźnienie które jest równe jednemu przebiegowi programu technologicznego, to znaczy że po wysłaniu pierwszego bloku danych nastąpi wykonanie jednej pętli programowej i nastąpi wysłanie kolejnego bloku. Dla różnych długości programów będzie ono różne. Parametr ten wykorzystywany jest głownie przy łączności internetowej przy użyciu łącza o niewielkiej przepustowości. Zwiększenie opóźnienia pomiędzy wysyłanymi blokami daje czas na przesłanie poprzedniego bloku danych. Parametr ten dobieramy eksperymentalnie w zależności od długości programu i rzeczywistych parametrów łącza. Zalecane wartości tego parametry zwykle nie przekraczają wartości równej 10. Zmiana tego parametry nie ma wpływu na prędkość zwykłej wymiany danych.

||---[TXT] Zapis tekstu do programu technologicznego

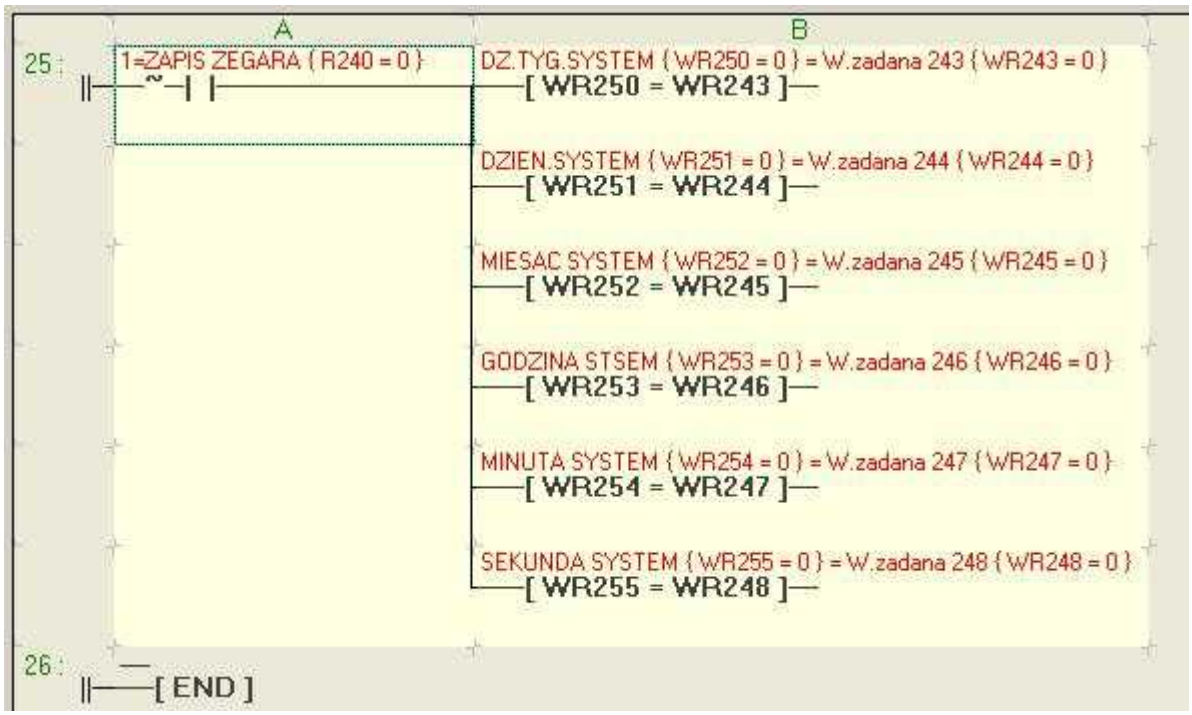
Rozkaz umieszczony jest na początku linii jako jedyny w linii. Funkcja pozwala zapisać tekst do kodu programu technologicznego. Tekstem można oznaczyć wersję programu. Dane zdefiniowane przez tą funkcję wgrywane są do sterownika razem z programem technologicznym. Na tej podstawie można bez programu źródłowego zorientować się jaką technologią mamy do czynienia. Funkcja znajduje się zazwyczaj na końcu programu technologicznego po funkcji [END].

	||---[TXT] kod 52 , 3...203		
1	Długość funkcji w bajtach	3..203	
2	Kod funkcji	52	
3	Tekst max 200 znaków	abcd	Tekst zapisywany do programu

Długość funkcji zostanie wyliczona automatycznie w zależności od długości zapisywanego tekstu. Należy pamiętać o tym że najdłuższy wprowadzany tekst nie może przekraczać 200 znaków. W tym przypadku długość funkcji będzie wynosiła 203 bajty.

7.OBSŁUGA ZEGARA STEROWNIKA

Sterownik microPLC jest wyposażony w zegar oraz kalendarz systemowy. Ustawianie go jest możliwe na dwa sposoby. Pierwszy to przy pomocy klawiatury urządzenia. Drugim sposobem jest możliwość ustawienia go z komputera za pośrednictwem programu technologicznego. Zapis jest dokonywany za pośrednictwem przekaźnika wewnętrznego R240. W czasie normalnej pracy zegara przekaźnik ma stan niski. Aktualna data i czas znajduje się w rejestrach WR250 – WR255. Zmiana daty i czasu z poziomu programu polega na wpisaniu do tych rejestrów nowych ustawień i zmianie stanu przekaźnika R240 na wysoki. Proces ten musi odbywać się tylko w jednym cyklu wykonania programu ponieważ po każdym cyklu rejestr R240 jest zerowany. Poniżej pokazano fragment programu który przestawia czas.



W programie do zmiany czasu wykorzystano rejestry pomocnicze od WR 243 do WR248. W rejestrach od WR234 do WR246 ma być zapisany nowa data, a w WR 246 do WR255 nowy czas. Teraz wystarczy zmienić stan przekaźnika wewnętrznego R240 na stan wysoki i zegar zostanie przestawiony. Teraz przekaźnik wewnętrzny R240 zostanie samoczynnie ustawiony na stan niski. Nową datę i godzinę można od razu odczytać z rejestrów czasu systemowego WR250 – WR255.

8.ZESTAWIENIE REJESTRÓW STEROWNIKA

Sterownik microPLC posiada dwa podstawowe typy rejestrów. Pierwszy rodzaj to rejestry przechowujące wartości dyskretne zero lub jeden. Rejestry te noszą nazwę impulsowych. Drugi rodzaj to rejestry przechowujące wartości analogowe reprezentujące liczby od 0 do 4096. Wszystkie operacje wykonywane są na tych właśnie rejestrach. Część z nich posiada konkretne przeznaczenie inne mogą być wykorzystywane w sposób dowolny.

REJESTRY IMPULSOWE

Rejestry wejść impulsowych X0 – X127

W sterowniku znajduje się taki fragment pamięci w którym zapisywany jest stan jaki panuje na zaciskach wejść impulsowych urządzenia. Wartość w tym rejestrze może przyjmować dwa stany 0 lub 1. Stan niski znajduje się w rejestrze wtedy gdy do wejścia impulsowego nic nie jest podłączone. Stan wysoki przekazywany jest wtedy gdy do wejścia impulsowego podłączymy napięcie stałe w zakresie 12 – 24V. Odczyt zacisków wejściowych sterownika dokonywany jest przed każdorazowym wykonaniem pętli programu technologicznego zapisanego w sterowniku.

Rejestry wyjść impulsowych Y0 – Y127

Inna część pamięci sterownika przechowuje stany jakie przekazywane są do przekaźników sterownika. Wartość w tym rejestrze może przyjmować dwa stany 0 lub 1. Jeżeli w rejestrze znajduje się stan niski to przekaźnik wyjściowy nie jest załączony. W pisanie stanu wysokiego do rejestru powoduje załączenie się przekaźnika. Po załączeniu urządzenia wszystkie przekaźniki zostają wyłączone. Załączenie ich może odbywać się za pośrednictwem klawiatury urządzenia lub poprzez program technologiczny zapisany do sterownika.

Rejestry przekaźników wewnętrznych R0 – R255

Kolejna część pamięci sterownika jest rejestrem stanów 0 lub 1. Rejestry te nazywają się dlatego przekaźnikami wewnętrznymi ponieważ nie są związane z żadnym fizycznym wejściem lub wyjściem sterownika. Są one formą pamięci w której przechowujemy stany wskaźnikowe które są wykorzystywane do pracy programu. Na przykład wynik porównania wartości dwóch rejestrów może być przechowywany w tym rejestrze. Dostęp do tych rejestrów odbywa się tylko za pośrednictwem programu PLCcomunicator. Istnieje pewna ilść rejestrów które mają z góry przydzielone funkcje. Niektóre z nich generują falę prostokątną. inne wysyłają impulsy w określonych odstępach czasu. Ich przeznaczenie zostanie wyjaśnione w tabeli na końcu tego rozdziału.

Rejestry auto/manual MY0-MY127

Rejestry te podobnie jak R są formą pamięci do przechowywania danych 0 lub 1. Różnica polega na tym że rejestry te są dedykowane do współpracy z rejestrami wyjść impulsowych sterownika. Dzięki nim można przeprowadzać przełączenie wyjścia w stan Manual lub Auto. Niski stan rejestru odpowiada trybowi pracy Auto stan wysoki odpowiada trybowi Manual. Dostęp do tych rejestrów jest możliwy z klawiatury sterownika.

REJESTRY ANALOGOWE

Rejestry wejść analogowych WX0 – WX63

W sterowniku znajduje się również taki fragment pamięci w którym zapisywane są wartości napięcia jakie dochodzą do zacisków wejść analogowych. W rejestrach tych mogą być gromadzone wartości w zakresie od 0 – 1023. Wartość ta wynika z rozdzielczości przetwornika analogowo – cyfrowego sterownika. Wartość ta zmienia sią proporcjonalnie w zależnością od wielkości napięcia na zaciskach wejścia urządzenia. Jeżeli do wejść sterownika nie doprowadzimy żadnego napięcia, wartość w rejestrze będzie = 0, jeżeli wartość napięcia będzie równa 6,7V w rejestrze zapisana zostanie wartość 670. Jak widać pomiar napięcia realizowany jest z dokładnością do 0,1%, dla zastosowań układów automatyki jest to wystarczające.

Rejestry wejść analogowych WY0 – WY31

Wyjścia analogowe reprezentuje inny fragment pamięci w którym zapisywane są wartości napięcia jakie wysyłane są przez sterownik. W rejestrach tych mogą być umieszczane wartości w zakresie od 0 – 255. Wartość ta wynika z rozdzielczości przetwornika cyfrowego - analogowego sterownika. Wartość napięcia wyjściowego zmienia się proporcjonalnie w zależnością od wielkości zapisanej w rejestrze wyjść analogowych. Jeżeli do rejestru zapiszemy wartość 0 to napięcie na zaciskach analogowych sterownika będzie równe 0V. Jeżeli do rejestru wpiszemy wartość 100 to na wyjściu pojawi się napięcie 5,0V. Z tego wynika że zmiana napięcia na wyjściu sterownika odbywa się z dokładnością 0,5%, co odpowiada ośmio-bitowej rozdzielczości przetwornika cyfrowo – analogowego.

Rejestry uniwersalne - wartości zadanych WR0 – WR255

Ostatnim typem rejestrów są rejestry uniwersalne zwane też rejestrami wartości zadanych. Rejestry te mogą przechowywać wartości w zakresie od 0 – 4095. Za ich pośrednictwem definiowane są wszystkie wartości zadane potrzebne dla pracy programu technologicznego. W rejestrach tych przechowywane są również wyniki obliczeń. Dostęp do tych rejestrów jest możliwy za pośrednictwem klawiatury sterownika. Ostatnie sześć rejestrów jest zdefiniowana na stałe. Znajduje się w nich kopia zegara systemowego sterownika.

TABELA DOSTĘPNYCH REJESTRÓW

Adres	Nazwa	Znaczenie	Zakres wartości	Status
X0 – X127	Wejścia impulsowe	0: brak sygnału na wejściu impulsowym sterownika 1: sygnał na wejściu impulsowym sterownika	0 lub 1	Odczyt
Y0 – Y127	Wyjścia impulsowe	0: przekaźnik wyjściowy sterownika wyłączony 1: przekaźnik wyjściowy sterownika załączony	0 lub 1	Odczyt/ zapis
R0 – R236	Przekaźniki wewnętrzne	0: przekaźnik wewnętrzny wyłączony 1: przekaźnik wewnętrzny załączony	0 lub 1	Odczyt/ zapis
R237	Przekaźnik wewnętrzny	0: funkcja [MASTER CALL] nie zainicjowała odczytu danych 1: oczekiwanie na dane dla funkcji [MASTER CALL]		
R238	Przekaźnik wewnętrzny	0: nie przyjęto odpowiedzi po magistrali RS 1: przyjęto odpowiedź po magistrali RS dla funkcji [MASTER RECEIVE]	0 lub 1	Odczyt
R239	Przekaźnik wewnętrzny	0: magistrala danych sterownika wolna 1: magistrala danych sterownika w trakcie transmisji	0 lub 1	Odczyt
R240	Przekaźnik wewnętrzny	0: zegar systemowy zapisuje swoją datę do rejestrów WR250, WR251, WR252, oraz czas w WR253, WR254, WR255 1: zegar systemowy jest ustawiany z rejestrów WR250, WR251, WR252 data i czas z WR253, WR254, WR255	0 lub 1	Odczyt/ zapis
R241	Przekaźnik wewnętrzny	0: sterownik pracuje ze standardową magistralą 1: sterownik pracuje w standardzie MODBUS RTU	0 lub 1	Odczyt/ zapis
R242	Przekaźnik wewnętrzny	0: ostatnia operacja dodawania lub mnożenia przebiegła poprawnie 1: podczas wykonywania ostatniej operacji dodawania lub mnożenia nastąpiło przekroczenie pojemności rejestru z wynikiem	0 lub 1	Odczyt
R243	Przekaźnik wewnętrzny	0: podświetlanie wyświetlacza LCD wyłączone 1: podświetlanie wyświetlacza LCD załączone	0 lub 1	Odczyt/ zapis
R244	Przekaźnik wewnętrzny	0: wyłączenie czerwonego LED „ALARM” 1: załączenie czerwonego LED „ALARM”	0 lub 1	Odczyt/ zapis
R245	Przekaźnik wewnętrzny	0: wyłączenie sygnału akustycznego wewnątrz urządzenia 1: załączenie sygnału akustycznego wewnątrz urządzenia	0 lub 1	Odczyt/ zapis
R246	Przekaźnik wewnętrzny	0: przyłączona magistrala RS485 do procesora 1: przyłączona magistrala RS232 do procesora	0 lub 1	Odczyt
R247	Przekaźnik wewnętrzny	Jeden impuls na dobę	0 lub 1	Odczyt
R248	Przekaźnik wewnętrzny	Jeden impuls na godzinę	0 lub 1	Odczyt
R249	Przekaźnik wewnętrzny	Jeden impuls na minutę	0 lub 1	Odczyt
R250	Przekaźnik wewnętrzny	Jeden impuls na sekundę	0 lub 1	Odczyt
R251	Przekaźnik wewnętrzny	Generator przebiegu prostokątnego o okresie 1 minuta	0 lub 1	Odczyt
R252	Przekaźnik	Generator przebiegu prostokątnego o okresie 1 sekunda	0 lub 1	Odczyt

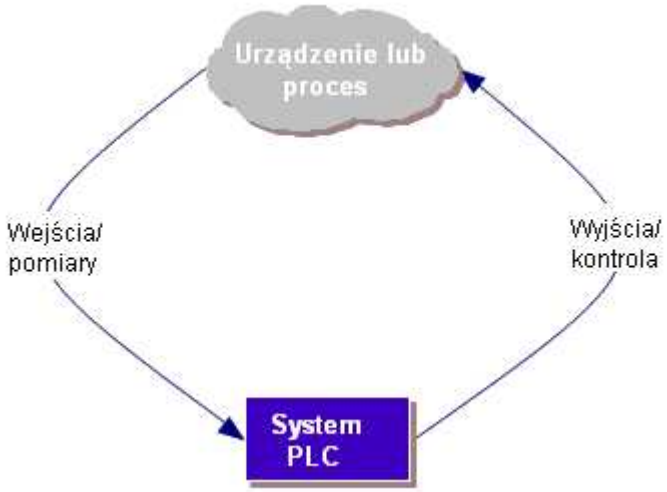
	wewnętrzny			
R253	Przekaźnik wewnętrzny	0: pierwsza pętla programu została wykonana 1: aktualnie wykonywana pierwsza pętla programu	0 lub 1	Odczyt
R254	Przekaźnik wewnętrzny	Zawsze zapisany stan = 0	0	Odczyt
R255	Przekaźnik wewnętrzny	Zawsze zapisany stan = 1	1	Odczyt
WX0 – WX63	Wejścia analogowe	Pomiar sygnału na zaciskach wejść analogowych sterownika	0 - 1023	Odczyt
WY0 – WY31	Wyjścia analogowe	Wysłanie napięcia 0 – 10V na zaciski analogowe sterownika	0 - 255	Odczyt/ zapis
WR0 – WR249	Rejestry uniwersalne	Rejestry wykorzystywane do przechowywania wartości wynikowych i zadanych	0 - 4095	Odczyt/ zapis
WR250	Rejestr uniwersalny	Dzień tygodnia zegara systemowego	1-7	Odczyt/ zapis
WR251	Rejestr uniwersalny	Dzień miesiąca zegara systemowego	1-31	Odczyt/ zapis
WR252	Rejestr uniwersalny	Miesiąc zegara systemowego	1-12	Odczyt/ zapis
WR253	Rejestr uniwersalny	Godziny zegara systemowego	0-23	Odczyt/ zapis
WR254	Rejestr uniwersalny	Minuty zegara systemowego	0-59	Odczyt/ zapis
WR255	Rejestr uniwersalny	Sekundu zegara systemowego	0-59	Odczyt/ zapis

9.CZYM JEST PLC?

NEMA, the National Electrical Manufacturers Association (Amerykańska Narodowa Organizacja Wytwórców elektroniki), definiuje programowalne sterowniki logiczne (PLC - a programmable logic controller) jako:

> elektroniczne urządzenie obsługiwane cyfrowo, które wykorzystuje programowalną pamięć do przechowywania instrukcji służących przeprowadzaniu specyficznych funkcji takich jak operacje logiczne, operacje sekwencyjne, mierzenie czasu, liczenie i wykonywanie działań arytmetycznych aby za pośrednictwem wejść i wyjść analogowych oraz cyfrowych sterować różnorodnymi typami maszyn lub procesów.

Definicja ta jak wszystkie definicje nie jest za bardzo zrozumiała. Spróbujmy wyjaśnić o co tu chodzi. Urządzenia PLC są wyspecjalizowanymi komputerami przemysłowymi. Współpracują z wejściami do których przyłączone są przełączniki i czujniki (systemu mierzenia bądź wykrywania). Ich stan analizowany jest przez program logiczny który odpowiednio zmienia stan wyjść w taki sposób aby właściwie sterować maszyną lub procesem.

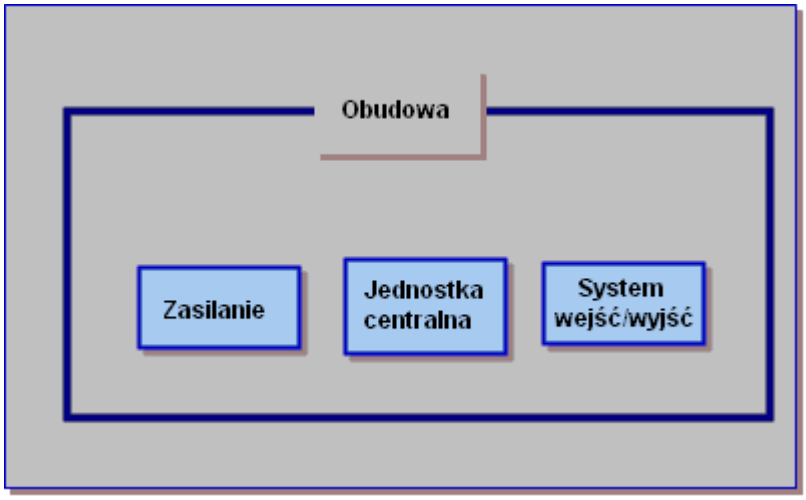


Początkowo systemy PLC były wykorzystywane aby zastąpić tradycyjne układy przekaźnikowe które realizowały funkcje logiczne. Ten system pozwala nam znacznie rozszerzyć funkcjonalność oraz daje możliwość kompleksowego wykorzystania w wielu aplikacjach. Systemy PLC są wykorzystywane w wielu układach przemysłowych gdzie sterowanie wymaga kompleksowości i ciągłej regulacji bądź tam gdzie niezbędna jest wysoka niezawodność.

Podstawowe składniki systemów PLC.

Szybki rozwój systemów PLC spowodowany był łatwością zastosowania go w aplikacjach przemysłowych ze względu na proste okablowanie oraz łatwość ich wymiany na nowe urządzenia. Obecnie produkowane urządzenia wielkością mogą być dostosowane do wymogów technologii. Bez względu na wielkość posiadają pięć podstawowych elementów:

- procesor centralny (CPU)
- obudowa urządzenia
- moduły wyjść
- zasilacz
- urządzenie programujące (komputer)



Zaczniemy wyjaśnianie od fizycznych składników, które widzimy gdy popatrzymy na sterownik. Potem objaśnimy to, co zawiera każda część i jak są one ze sobą powiązane.

Program drabinkowy

Większość średnich i dużych sterowników zbudowana jest z oddzielnych elementów takich jak – CPU, moduły wejścia/wyjścia i zasilacz. Są to moduły które umieszczone są wewnątrz obudowy lub na mocującej je płycie bazowej. W mniejszych systemach PLC wszystkie te komponenty mogą znajdować się w jednej obudowie.

Zasilanie

Zasilanie urządzenia zapewnia zasilacz, dostarczający odpowiednich napięć. Napięcie do zasilacza zwykle podaje się na specjalne zaciski wejściowe. Zasilanie sterownika odbywa się za pomocą wtyku na którym pojawiają się wymagane do pracy procesora i modułów wejścia wyjścia napięcia stałe. Jest to możliwe dzięki Najpowszechniej używanymi są napięcia zasilania 24V DC bądź 230V AC.

Procesor (CPU)

CPU (central processing unit – jednostka centralna) jest „mózgiem” sterownika. Jego wielkość i typ określa – dostępność funkcji programowalnych, wielkość aplikacji, dostępna ilość pamięci i prędkość sterowanych procesów.

Moduły wejść/wyjść

Wejścia przenoszą sygnał z urządzeń zewnętrznych do sterownika. Urządzeniami zewnętrznymi mogą być przełączniki, mierniki ciśnienia, wejścia operacyjne itp. Można porównać je z ludzkimi zmysłami.

Wyjścia są elementami, które system PLC wykorzystuje aby przesłać zmiany „w świat”. Są wykorzystywane aby sterownik mógł zmienić opcje, dostosować się aby kontrolować dany system np. wyłączyć bądź włączyć urządzenia takie jak silniki, światła, pompy etc.

Do systemów PLC może być podłączone wiele typów wejść i wyjść. Ogólnie możemy podzielić je na dwie duże grupy – wejścia i wyjścia analogowe oraz cyfrowe. Cyfrowe to takie, które pracują na sygnałach dwu stanowych – załącz/wyłącz, tak/nie. Wejścia analogowe wykorzystywane są do pomiarów ciągłej zmiany różnorodnych wartości np. ciśnienia, temperatury, pozycji potencjometrów.

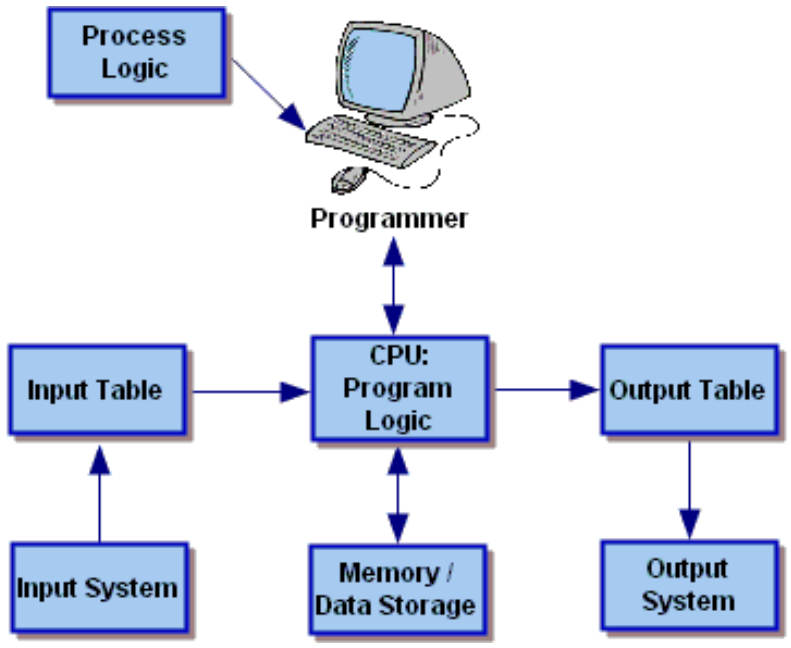
Programatory sterowników

Systemy PLC są skonstruowane do współpracy ze specjalistycznymi programami komputerowymi, które pozwalają załadować i zmienić ich wewnętrzną logikę. Najnowocześniejsze urządzenia są przystosowane do użycia oprogramowania na komputerze stacjonarnym lub laptopie. Starsze systemy wykorzystują specjalistyczne programatory.

Podstawowe operacje systemów PLC

Basic Operation of a PLC system

The operation of the PLC system is simple and straightforward. The Process or CPU completes three processes: (1) scans, or reads, from the input devices (2) executes or "solves" the program logic, and (3) updates, or writes, to the output devices.



PLC Program

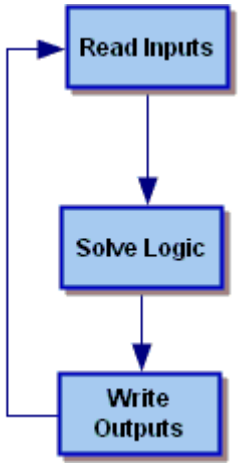
For the PLC to be useful, it must first have a Program or Logic for the CPU to execute. A system engineer or PLC programmer will first create the program logic in a programming device (these days it is usually software running on a personal computer). This logic can be written in Ladder Logic, Instruction List, Sequential Function Charts, or any of the IEC languages.

The programmer will then download the program to the PLC. This is usually done by temporarily connecting the programmer to the PLC. Once the program is installed or downloaded to the CPU - it is usually not necessary for the PC to remain connected.

Basic Scan

Once the program is in the CPU - the PLC is then set to "run", and the PLC executes the application program repeatedly. In addition to executing the program, the CPU regularly reads the status of the input devices, and sends data to the output devices. The Input system senses the status of the real world inputs (a switch, a level, etc.), translates them to values that can be used by the CPU, and writes those values to the Input table. The application program is executed, and writes values to the Output table. The Output system then converts the output value to a real world change (motor turns on, valve opens, etc.)

This process of reading inputs, executing logic, and writing outputs is called the PLC Scan or Sweep.



The CPU continuously Reads Inputs, Solves Logic, and Writes to the outputs (there are other tasks the CPU does - which will be discussed later). It is important to understand the scan because it may dictate how a programmer structures logic.

Memory

The control program or application program is stored in memory. As the PLC executes logic, it may also read and store values to memory. The values may also be used and refernced by the application program.

Moduły wejść i wyjść w sterownikach PLC

Używane poniżej określenie I/O odnosi się do terminu wyjścia/wejścia (ang. outputs/inputs). I/O reprezentuje dane otrzymywane z czujników oraz sygnały do zasilania urządzeń wykonawczych. System I/O to zbiór fizycznych elementów systemu sterowania, z których są pobierane dane do dalszej ich analizy oraz wysyłane do sterowania urządzeniami. Wyróżniamy dwa główne typy wejść i wyjść:

- Cyfrowe – sygnały dwustanowe, które mogą rozróżniać tylko dwa stany – on i off (załącz i wyłącz)
- Analogowe – sygnały reprezentujące pomiar ciągły – odczytują i odpowiadają na zmiany wartości

Cyfrowe systemy I/O

Systemy te mogą przyjmować tylko dwa stany – załączony lub wyłączony. Nie mogą natomiast przyjmować innych wartości. Na przykład impulsowe czujniki zblizeniowe nie odczytają w jakiej odległości od nich jest dany obiekt. Mogą tylko stwierdzić czy obiekt znajduje się w zasięgu czujnika. Do wejść cyfrowych można podpiąć grupę urządzeń takich jak przyciski, przełączniki krańcowe oraz fotokomórki. Wyjścia obejmują urządzenia takie jak: przekaźniki, styczniki i napędy zaworów.

Analogowe systemy I/O

Analogowe systemy I/O odczytują wartości ciągłe. Informacja przez nie odczytywana jest w postaci zbioru różnych wartości, a nie tylko stanu załączenia lub wyłączenia. Wejścia te mogą mierzyć wartości takie jak temperatura, ciśnienie, prędkość itd. Wyjścia analogowe wysyłają sygnały ciągłe o takim samym charakterze jak z wejścia analogowe. Sygnały takie pokazują np. ciśnienie powietrza, prędkość silnika czy pozycję zaworu

Moduły I/O łączą grupę urządzeń technologicznych ze sterownikiem. Przetwarzają impulsy elektryczne wysyłane przez urządzenia na sygnały elektroniczne mogące być wykorzystane w systemie sterowania które następnie mogą być zapisane w pamięci urządzenia.

Moduły I/O komunikują się z CPU sterownika na 3 sposoby:

- jednomodułowe– moduły I/O mogą być umieszczone w tej samej obudowie, a komunikacja między nimi odbywa się bezpośrednio na tej samej płytce

- wielomodułowe – moduły rozszerzające pozwalają modułom I/O znajdować się w obudowie bezpośrednio sąsiadującej z obudową sterownika

- sieć rozproszona – moduły mogą komunikować się ze sterownikiem poprzez sieć. Sieci przemysłowe używane są do wewnętrznego komunikowania się urządzeń mierzących parametry procesu z kontrolerami. Przykładowymi sieciami I/O są FieldBus, ProfiBus, DeviceNet.

Ilość punktów pomiarowych oraz rozmiar systemów PLC

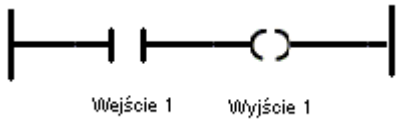
Liczba urządzeń I/O użyta wewnątrz systemu kontrolującego nosi nazwę punktu pomiarowego. Dane z urządzeń analogowych znacznie więcej obciążają system niż dane cyfrowe. Analogowe i cyfrowe punkty pomiarowe zwykle zliczane są oddzielnie. Liczba wszystkich cyfrowych i analogowych sygnałów pozwala ustalić wielkość urządzenia sterującego.

Logika drabinkowa

Logika drabinkowa (bądź diagramy drabinkowe) są najczęściej wykorzystywanym językiem programowania sterowników PLC.

Logika drabinkowa była jedną z pierwszych dostępnych metod programowania w systemach PLC, ponieważ zapożyczała dużo ze schematów elektrycznych, które powszechnie były znane w branży elektrycznej.

Symbole użyte w logice drabinkowej składają się gałęzi zasilającej z lewej, drugiej gałęzi zasilającej z prawej i poszczególnych elementów łączących obie gałęzie. Logika poszczególnych elementów rozwiązywana jest od strony lewej do prawej. Symbole tych diagramów wyglądają jak drabina – z dwoma bokami, a środkowe elementy przypominają szczeble.



Na powyższym obrazku widoczny jest jeden szczebel drabiny.

- jeśli wejście 1 ma wartość 1 (binarnie czyli on) – w wyniku pracy gałęzi wyjście przyjmuje wartość 1

-jeśli wejście 1 ma wartość 0 (off) – w wyniku działania gałęzi wyjście przyjmuje wartość 0

Jest wiele symboli dostępnych w logice drabinkowej np. czasomierze, liczniki, wyliczenia matematyczne – wszystkie z tych elementów mogą zostać wykorzystane w logice drabinkowej. Kilka najbardziej popularnych symboli to – styk otwarty/ zamknięty, przekaźnik otwarty/zamknięty, czasomierz, licznik.