

Laboratorium Systemów Komunikacyjnych w Przemysłowych Układach Automatyki PLC

Sieć przemysłowa AS-Interface



Instrukcja laboratoryjna powstała na podstawie pracy dyplomowej mgr inż. Łukasza Odrobiny

1. Wstęp

W ćwiczeniu zaprezentowano, możliwości jakie daje zastosowanie sterownika PLC w sterowaniu przykładowym procesem technologicznym, sposoby komunikacji z aparaturą detekcyjną (czujniki fotoelektryczne, indukcyjne, wyłączniki krańcowe), języki programowania sterowników (list instrukcji, schematów drabinkowych, Grafcet), a także sposoby oddziaływanie na elementy wykonawcze takie jak przemiennik częstotliwości regulujący prędkość silnika, urządzenia sygnalizacyjne itp.

Praca została wykonana na makiecie, którą przekazała Politechnice Warszawskiej firma Schneider Electric.

2. Budowa stanowiska

2.1 Schemat połączeń układu

Makieta, jaką otrzymała do dyspozycji Politechnika Warszawska, została zbudowana przez inżynierów z firmy Schneider Electric na potrzeby targów Automaticon i jest wyposażona w aparaturę wykorzystywaną obecnie w przemyśle. Przedstawia ona zminiaturyzowaną linię technologiczną, symulującą pewien proces.



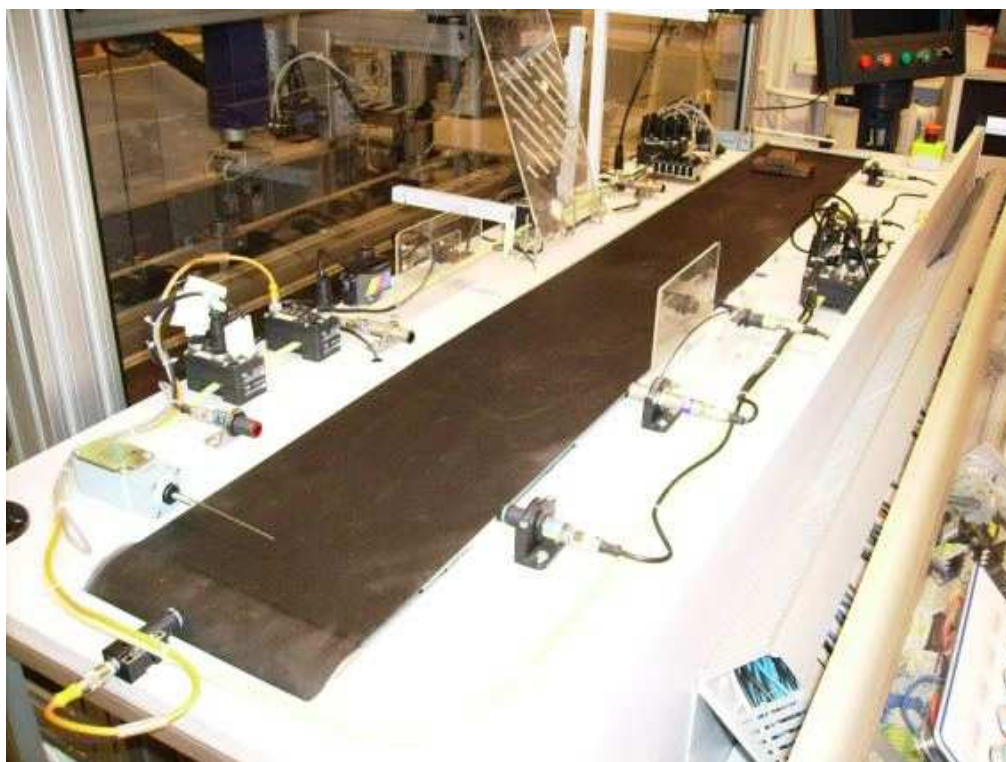
Rys.2.1.1. Model zminiaturyzowanej linii technologicznej

Jej sercem jest sterownik programowalny PLC, który za pomocą specjalnych przystawek komunikacyjnych pracujących w magistrali AS-i, zbiera dane z licznych czujników fotoelektrycznych, indukcyjnych itp., a następnie po przetworzeniu sygnałów wejściowych, poprzez przemiennik częstotliwości, steruje pracą silnika indukcyjnego napędzającego taśmociąg.

a)



b)



Rys. 2.1.2. Aparatura wchodząca w skład układu: a) zasilacz magistrali AS-i oraz sterownik; b) czujniki i wyłączniki krańcowe.

Układ składa się z czterech części. Pierwszą z nich stanowi obwód sterownika PLC, wraz z dołączonymi do niego dwoma zewnętrznymi modułami złączy wejść i wyjść dwustanowych. Druga część to obwód przetwornicy napięcia 230V AC / 24V DC zasilającej wspomniane wcześniej zewnętrzne moduły złączy wejść i wyjść oraz dołączone do nich czujniki i elementy wykonawcze. Trzecia, to magistrala As-i z dołączonymi do niej czujnikami, komunikująca się ze sterownikiem za pomocą specjalnego dodatkowego modułu. Czwarta natomiast, to obwód zasilania przemiennika częstotliwości.

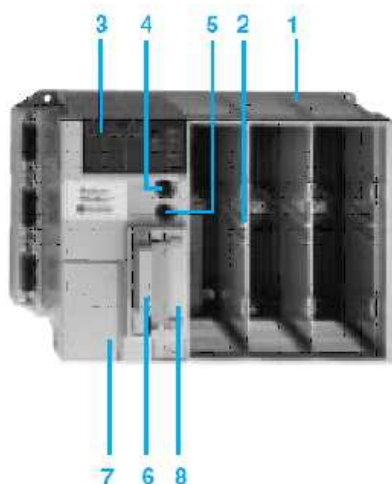
Po zamknięciu rozłącznika Q10 oraz wyłącznika głównego Q1, a następnie kolejnych wyłączników Q2, Q3, Q4, Q5, Q6 i Q7 zabezpieczających wymienione wcześniej obwody, doprowadzone zostanie do całego układu napięcie zasilania.

W obwodzie przemiennika częstotliwości, po przetworzeniu napięcia z 230V AC na 24V DC za pomocą transformatora T1, podane zostaje napięcie na zaciski A1, A2 cewki stycznika KM1, zamykającego za pomocą styków głównych 1-2 / 3-4 obwód zasilania przemiennik częstotliwości. Układ jest gotowy do pracy.

Regulacja pracy silnika odbywa się za pośrednictwem przemiennika częstotliwości na podstawie sygnałów sterujących doprowadzonych do niego ze sterownika. Wciśnięcie przycisków awaryjnych włączonych szeregowo w obwód pomocniczy poprzez styki NC, powoduje ich otwarcie i przerwanie obwodu. Zanik prądu w cewce stycznika KM1 i rozłączenie jego styków głównych 1-2 / 3-4 przerywa obwód zasilania przemiennika i zatrzymuje pracę silnika.

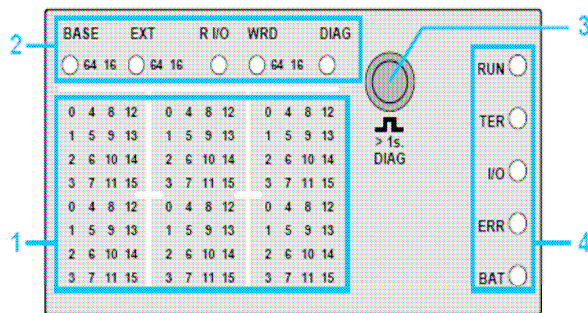
2.2 Sterownik PLC

Sterownikiem zastosowanym w układzie jest Modicon TSX Micro 3721 v 4.0. Jest on zasilany napięciem przemiennym 100-240V i posiada wbudowane zabezpieczenie wyjść przeciwko zwarciom i przeciążeniom. Może on zostać wyposażony w 5 modułów we/wy dwustanowych 28/32 kanałowych lub 3 moduły 64 kanałowe, co pozwala na obsługę 140/160 urządzeń zewnętrznych (czujniki/elementy wykonawcze) lub do 248 w przypadku zastosowania specjalnego złącza HE10. Pozwala on również na zastosowanie 4 modułów analogowych (32 we lub 16 wy), 4 modułów zliczających, 8 modułów zabezpieczeniowych gamy PREVENTA, oraz jednego modułu komunikacyjnego dla magistrali AS-i (340 we, wy lub we/wy). Posiada zintegrowany port dla komunikacji RS485 (UNI-TELWAY lub tryb znakowy), zegar czasu rzeczywistego i wewnętrzną pamięć RAM o pojemności 20k słów z możliwością rozszerzenia za pomocą karty PCIMCIA do 84k. Czas wykonania jednej instrukcji od 0.13us do 4.50us.



- 1 3- gniazdowa kaseta podstawowa
- 2 gniazdo na moduł o formacie standardowym lub półówkowym,
- 3 centralny blok wyświetlacza,
- 4 złącze terminala (do programowania),
- 5 złącze interfejsu operatorskiego,
- 6 gniazdo karty rozszerzenia pamięci,
- 7 osłona dostępnych zacisków zasilania,
- 8 gniazdo modułu komunikacji,

Rys.2.2.1. Sterownik TSX Micro 3721 z kasetą rozszerzeń



1 trzy bloki 32 diod LED reprezentujące gniazda, w których zostały zainstalowane moduły w kasecie podstawowej i rozszerzenia,

2 linia informacyjna składająca się z diod LED pokazujących wyświetlane stany pracy,

3 przycisk rozkazów, który umożliwia dostęp do wyświetlania różnorodnych stanów pracy,

4 pięć diod LED:

- RUN, PLC start/stop,
- TER, złącze terminala pracuje (odbywa się komunikacja),

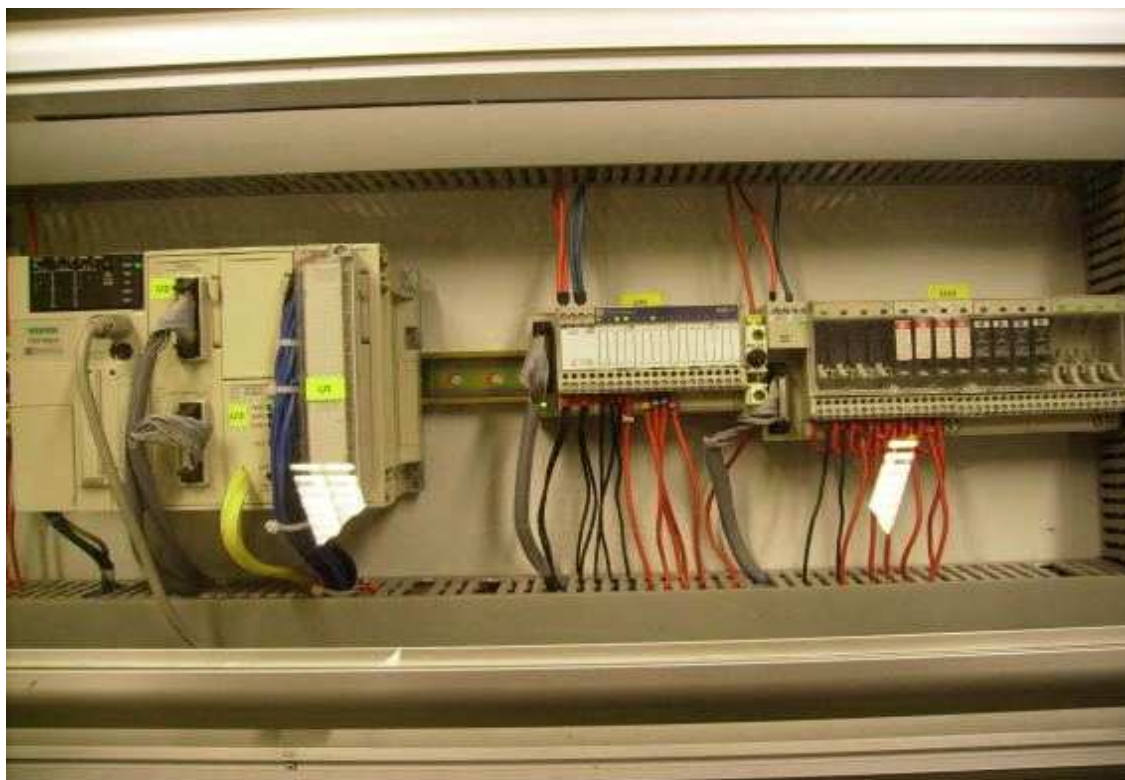
- I/O, błąd we/wy,

- ERR błąd procesora lub aplikacji,

- BAT, uszkodzenie lub jej brak

Rys.2.2.2. Centralny blok wyświetlacza

Sterownik ten wyposażony został w dwa moduły 16wejść/12wyjść dwustanowych 24VDC (jeden z listwą przyłączeniową, drugi komunikujący się za pomocą złącza HE10 z zewnętrznymi modułami Telefast2), oraz moduł komunikacyjny dla magistrali AS-i.



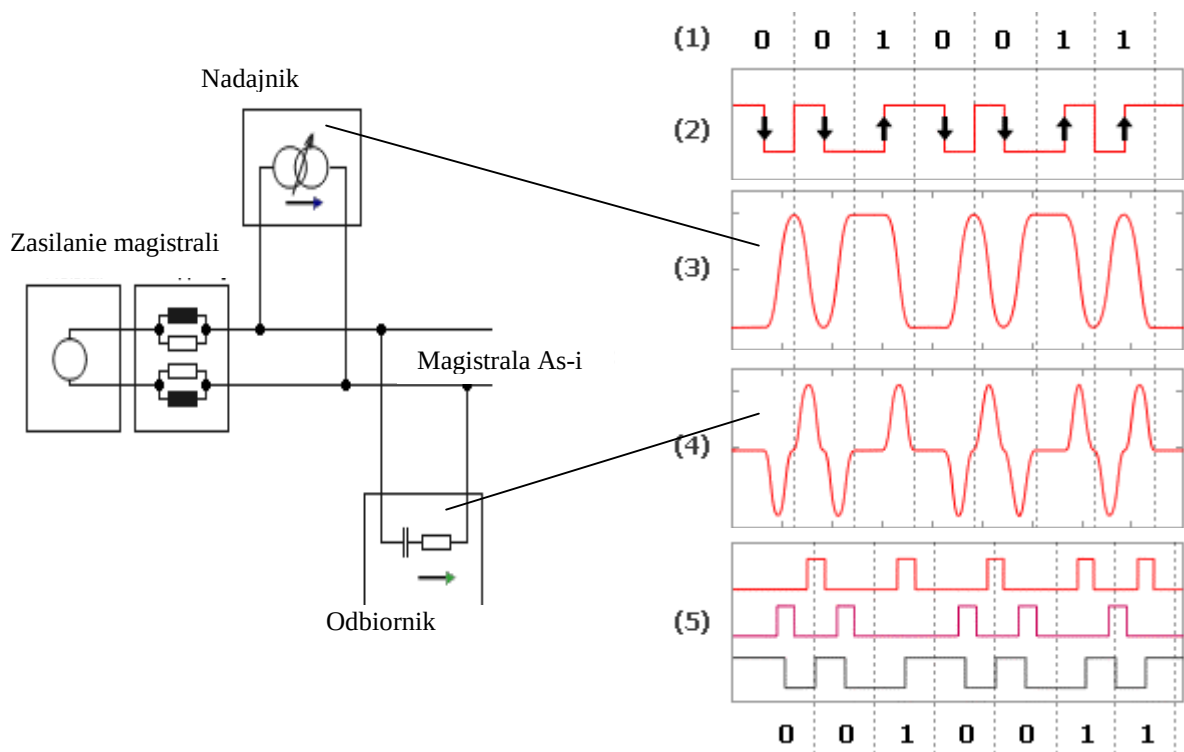
Rys. 2.2.3. Sterownik TSX Micro z zainstalowanymi modułami we/wy oraz modułem AS-i (żółty przewód) , przystawki Telefast2.

2.3 Magistrala As-i

AS-i (Actuator Sensor Interface) jest otwartym standardem przemysłowym magistrali obejmującej czujniki i elementy wykonawcze, charakteryzującej się bardzo krótkim czasem odpowiedzi pozwalającym na bardzo szybką reakcję elementów wykonawczych.

Standard ten wykorzystuje do transmisji danych kod PE (Phase Encodage), zwany też kodem Manchester. Reguła kodowania jest następująca: '1' kodowana jest jako przejście w środku bitu, od poziomu niższego do wyższego, '0' - odwrotnie. Zaletą takiego kodowania jest co najmniej jedno przejście dla każdego bitu. Słowo nadawane, w naturalnym kodzie binarnym (1), kodowane jest według reguł kodu PE (2) a następnie podawane na filtr

dołnoprzepustowy. Przebieg ten o ograniczonym paśmie wchodzi na konwerter U/I, na którego wyjściu otrzymujemy prąd I z przedziału (0...60) mA. Takie zmiany prądu są forsowane na magistrali (3). Oddziaływanie między wyjściem nadajnika a specjalnym zasilaczem z odpowiednio dobraną wartością indukcyjności wyjściowej prowadzi do uzyskania naprzemiennej modulacji impulsowej (APM - Alternate Pulse Modulation) napięcia zasilającego(4). Modulacja APM odpowiada różniczkowaniu przebiegu (3).



- (1) Sekwencja przesyłanych bitów
- (2) Kodowanie Manchester
- (3) Nadawany sygnał prądowy
- (4) Napięcie wykrywane przez odbiornik
- (5) Sekwencja odebranych bitów

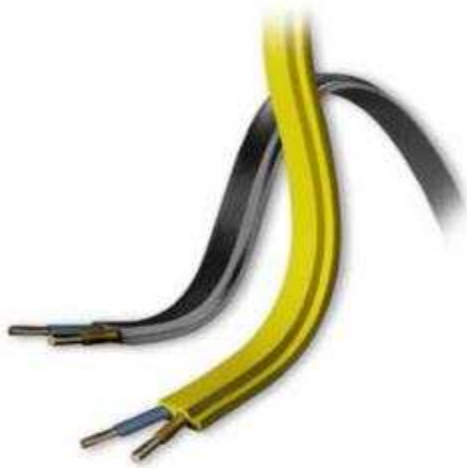
Swoją uniwersalność magistrali AS-i zawdzięcza elektronicznym układom scalonym (ASIC-Application Specific Integrated Circuit), które mogą być

zintegrowane bezpośrednio z czujnikiem lub elementem wykonawczym (określanym jako element komunikacyjny) lub blokiem specjalnego rozdzielacza, który umożliwia przyłączenie do 4 urządzeń.



Rys.2.3.1 Rozdzielacz z modułem ASIC

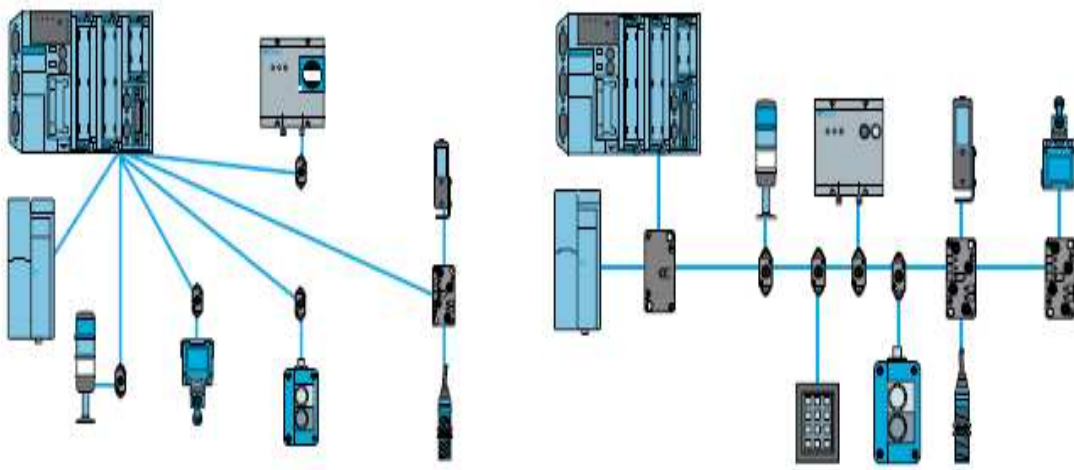
Urządzenie to zarządza wszystkimi funkcjami komunikacyjnymi czujników w celu przesyłania do mastera AS-i (sterownika) informacji o stanach czujników. Dane są transmitowane przez kable zawierające parę nieskręconych, nieekranowanych żył o przekrojach 1,5 do 2,5 mm² które jednocześnie doprowadzają zasilania do czujników i członów wykonawczych.



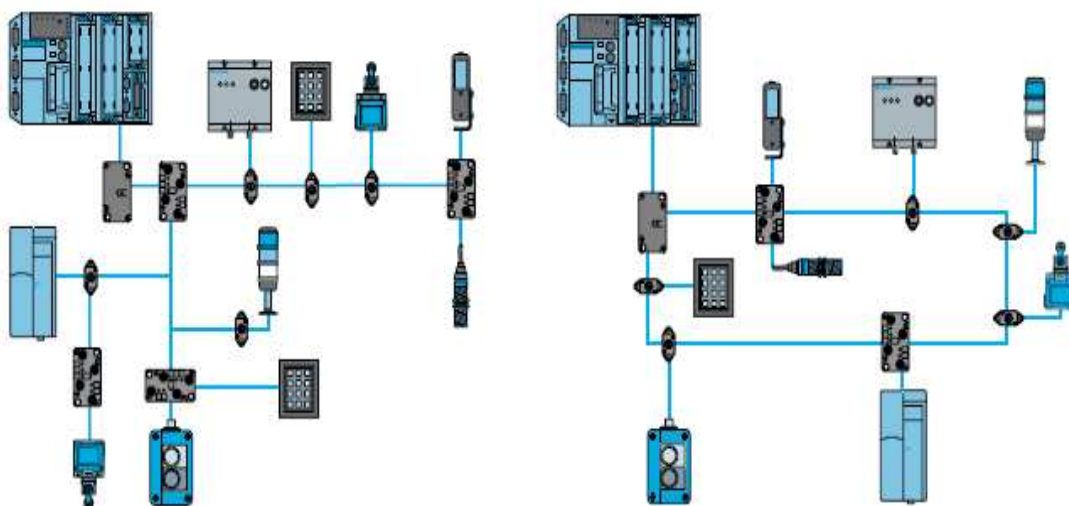
Rys.2.3.2 Przewód 2 żyłowy wykorzystywany do transmisji danych As-i

Maksymalna odległość, na jaką mogą być transmitowane sygnały to 100m, lub 200 w przypadku zastosowania specjalnego wzmacniacza.

Topologia magistrali AS-i jest otwarta. Nowe połączenia mogą być wykonane w dowolnym punkcie, bez potrzeby użycia adaptera.



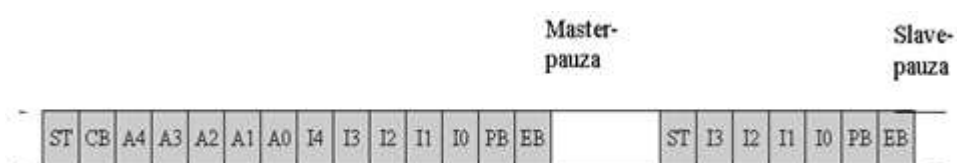
Rys. 2.3.3 Topologia As-i: punkt-punkt oraz linia



Rys. 2.3.4 Topologia As-i: drzewo oraz pierścień

AS-i jest magistralą typu master-slave. Dialog jest zawsze inicjowany przez pojedynczego mastera zarządzającego magistralą. Wysyła on telegram zaopatrzony w adres modułu slave i odpowiada wyłącznie wskazany moduł.

Format ramki pakietu przesyłanych bitów przedstawia się w sposób następujący:



- ST** – bit startu zawsze ST=0
- CB** – bit sterujący, 0-przesyłanie danych/parametry, 1-instrukcja sterująca
- A0...A4** – adres modułu slave, 01H...1FH
- I0...I4** – bity informacyjne zależne od typu wywołania
- PB** – bit parzystości, testowane są bity bez ST i PB, 0-parzysta liczba '1'
- EB** – bit stopu, zawsze EB=1

Rys. 2.3.5 Format ramki w transmisji AS-i

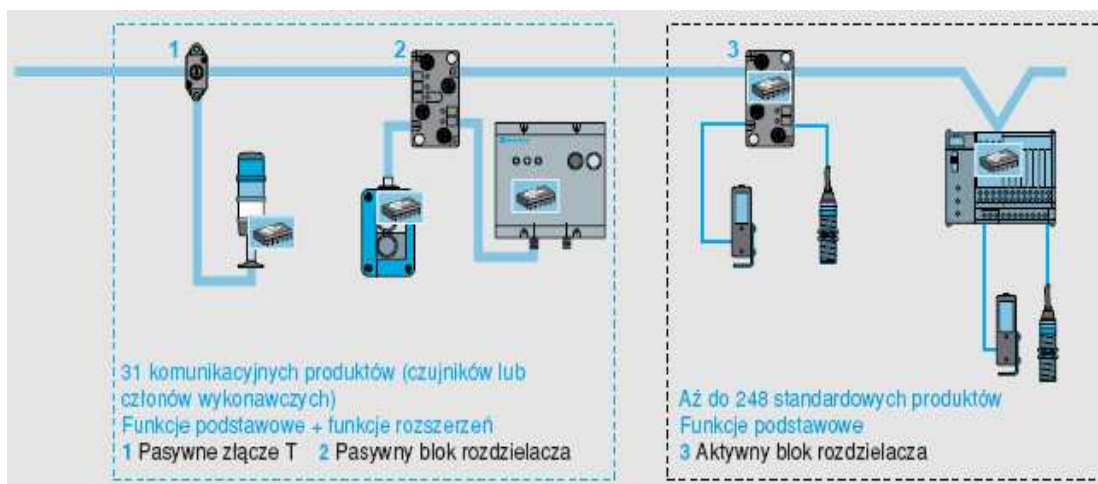
Master-pauza wynosi 3-10 odcinków jednostkowych. Jeśli slave jest zsynchronizowany, wtedy już po trzech odcinkach bitowych przełącza się na nadawanie odpowiedzi. Przy braku synchronizacji wymagane są dwa dodatkowe bity. Jeśli master nie otrzyma bitu startowego odpowiedzi w ciągu 10 bitów pauzy, przechodzi do następnego wywołania. Slave-pauza trwa tylko jeden odcinek czasowy. Czas cyklu dostępu w sieci AS-i zależy od liczby modułów podrzędnych; przy pełnej obsadzie 31 modułów wynosi on około 5ms.

Magistrala AS-i akceptuje do 31 slave'ów (komponenty zawierające inteligentne moduły ASIC) każdy z 4 bitami wejścia i 4 bitami wyjścia do cyklicznej wymiany danych z masterem oraz 4 bitami ustawiania parametrów pozwalającymi na rozszerzenie funkcji (konfiguracja, diagnostyka itp.).

Każdy slave ma swój własny adres i jest definiowany przez profil (kod wyszczególniający typ jednostki slave). Adres ten jest przydzielany poszczególnym urządzeniom za pomocą specjalnego terminala. Każde urządzenie, któremu adres taki nie został jeszcze przypisany, ma nadany adres 0 (zabroniony dla terminala), który po wykryciu przez mastera jest automatycznie zmieniany na tymczasowy „pierwszy wolny”. Dzięki temu urządzenia bez fizycznie przypisanego adresu mogą być podłączane w dowolnym momencie w dowolnym punkcie magistrali.

Czujniki komunikacyjne lub człony wykonawcze (zawierające ASIC) są bezpośrednio dołączone do magistrali za pośrednictwem pasywnego bloku rozdzielacza lub złącz T. Tradycyjne dwustanowe czujniki lub człony wykonawcze (bez ASIC) są natomiast dołączane do magistrali z wykorzystaniem aktywnych bloków rozdzielacza lub specjalnych interfejsów.

Maksymalna liczba czujników komunikacyjnych lub członów wykonawczych (z ASIC), które mogą być połączone wynosi 31, a dla tradycyjnych czujników lub członów wykonawczych liczba ta wynosi 248. Możliwa jest kombinacja standardowych i komunikacyjnych czujników/elementów wykonawczych.

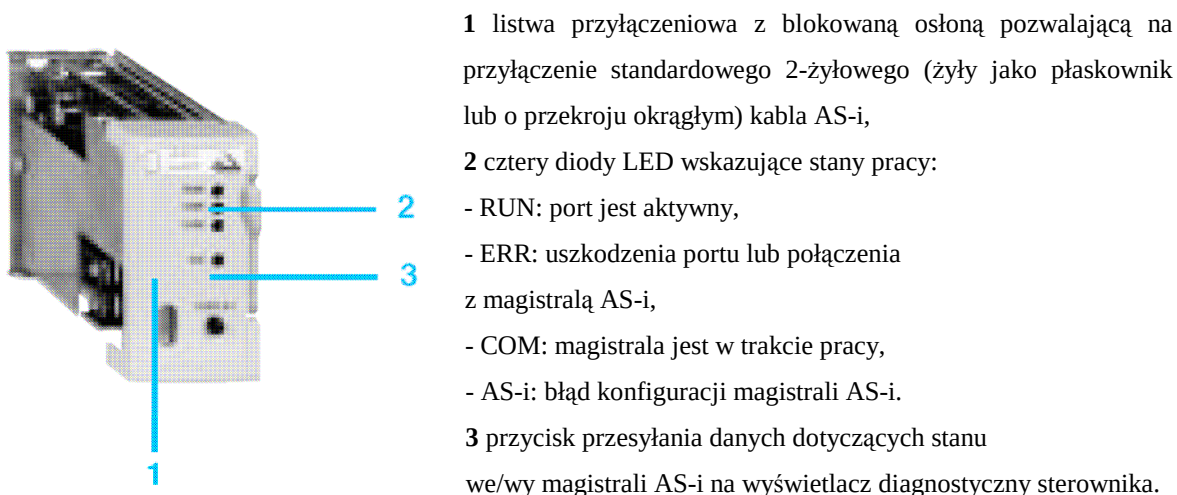


Rys. 2.3.6 Sposoby przyłączania urządzeń typu Slave: z modułem ASIC, oraz bez modułu



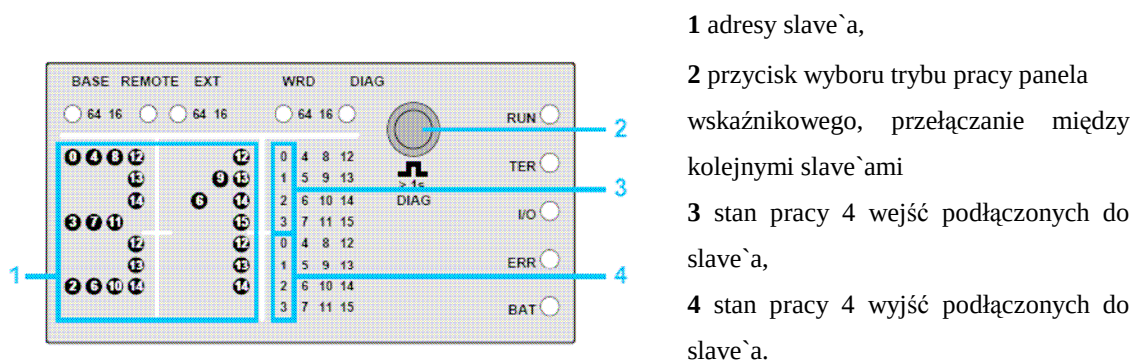
Rys. 2.3.6 Sposoby przyłączania urządzeń typu Slave: czujniki oraz aktywne bloki rozdzielaczy

Aby sterownik TSX Micro mógł pełnić funkcję mastera dla magistrali As-i, musi być on wyposażony w specjalny moduł TSX SAZ 10 (instalowanym zawsze w „gnieździe 4”)z dołączonym do niego bezpośrednio przewodem magistrali. Moduł ten cyklicznie komunikuje się z każdym z urządzeń uaktualniając dane we/wy każdego z nich i przesyła je do programu aplikacji sterownika.



Rys. 2.3.7 Moduł TSX SAZ 10

Moduł TSX SAZ 10 umożliwia również diagnozowanie magistrali. Po wciśnięciu przycisku 3 modułu, na centralny wyświetlacz sterownika TSX Micro wyświetlany zostaje stan wszystkich we/wy kanałów a także dodatkowe informacje o urządzeniach podłączonych do magistrali (obecność, nieobecność, uszkodzenie, niezgodność z konfiguracją.):



Rys.2.3.8 Tryb diagnozowania na wyświetlaczu sterownika

Magistrala AS-i jest konfigurowana przez sterownik z użyciem pakietu oprogramowania PL7 Micro lub PL7 Junior z zaimplementowaną biblioteką urządzeń typu slave. Oprogramowanie to umożliwia ponadto:

- zarządzanie profilami i parametrami slave`ów,
- określanie adresów slave`ów: każdy węzłowy slave AS-i wymaga adresu (pomiędzy 1 i 31). Operacja przypisania może być przeprowadzona automatycznie,
- każde wejście i wyjście AS-i będzie traktowane jako konwencjonalne we/wy w programie aplikacji (np. %I\4.0\16.2 oznaczać będzie wejście 3 węzła magistrali AS-i z adresem 16).

2.4 Przekształtnik częstotliwości Altivar 58

Taśmociąg makiety napędzany jest za pomocą trójfazowego silnika klatkowego o mocy 120 W z reduktorem i rezystorem rozładowczym. Jak w każdym silniku indukcyjnym tego typu, regulacja prędkości odbywa się poprzez zmianę częstotliwości oraz wartości napięcia zasilającego.

Przekształtnikiem zastosowanym w układzie jest przekształtnik Altivar58.



Rys.2.4.3 Przekształtnik częstotliwości Altivar 58.

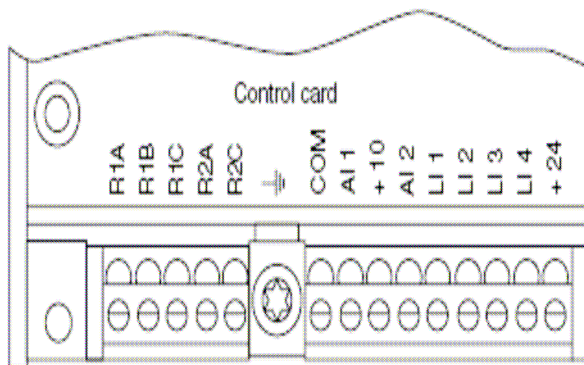
Przekształtnik ten, oprócz tego że może być stosowany we wszystkich rodzajach napędów (charakterystyki maszyn roboczych: stałomomentowe, liniowe, wentylatorowe) do regulacji prędkości, rozruchów oraz hamowania o zadanych rampach (stromości narastania lub opadania przebiegu prędkości w czasie), posiada także wiele innych przydatnych funkcji. Należą do nich między innymi:

- funkcja regulatora PI,

- realizacji pętli sprzężenia zwrotnego regulacji prędkości z wykorzystaniem prądniczki tachometrycznej lub przetwornika obrotowo-impulsowego,
- automatycznego przechwytywania wirującego obciążenia z pomiarem prędkości („lotny start”)
- zabezpieczenia zwarciovowego oraz przeciążeniowego silnika.

Ponadto, przemienniki te wyposażone są w filtry EMC zapewniające kompatybilność elektromagnetyczną a także złącze szeregowo RS 485.

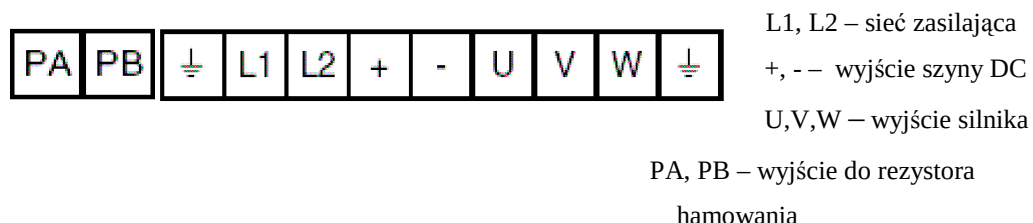
Programowanie przemiennika odbywa się za pomocą dodatkowego panelu operatorskiego z wyświetlaczem LCD lub komputera PC z zainstalowanym oprogramowaniem PowerSuite, natomiast sterowanie za pomocą sygnałów dostarczanych do zacisków listwy obwodów sterowania.



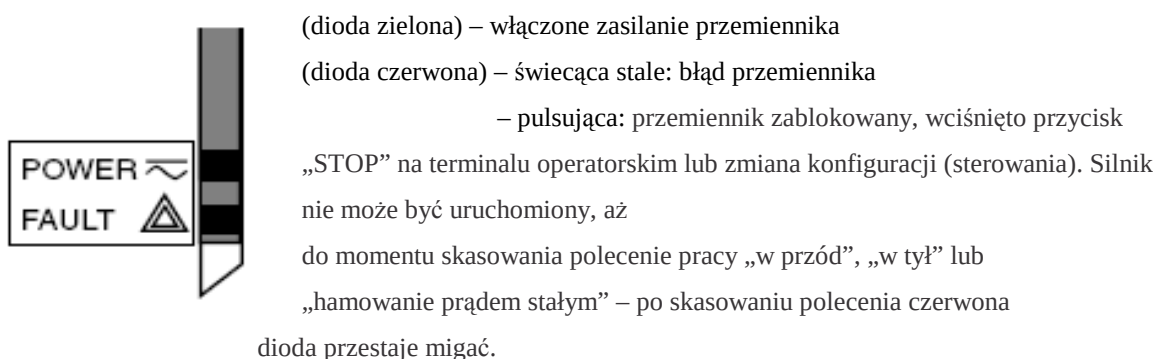
R1A...C – styki przekaźnika sygnalizującego stan błędu
 R2A, R2C – styki przekaźnika programowalnego
 COM – zacisk wspólny dla wejściowych sygnałów analogowych i logicznych
 AI1, AI2 – wejścia analogowe dla sygnałów: napięciowego i prądowego
 LI1...LI4 – wejścia logiczne
 +24, +10 – źródła zasilania dla sygnałów

analogowych i logicznych

Rys. 2.4.4. Listwa zacisków obwodów sterowania



Rys. 2.4.5. Listwa zacisków obwodów silnoprądowych



Rys. 2.4.6. Sygnalizacja na płycie czołowej sterownika

Po doprowadzeniu napięcia zasilania do zacisków L1, L2 przemiennika, zapali się dioda zielona, a styki przekaźnika R1A, R1C zamkną się sygnalizując gotowość do pracy. W tym momencie za pomocą sygnałów logicznych doprowadzanych do zacisków LI1-LI3 możliwe jest sterowanie pracą przemiennika, w oparciu o zaprogramowane parametry: „Low Speed” – niższa wartość prędkości, „High Speed” – wyższa wartość prędkości, czas narastania prędkości do wartości zadanej oraz jej opadania, itp.

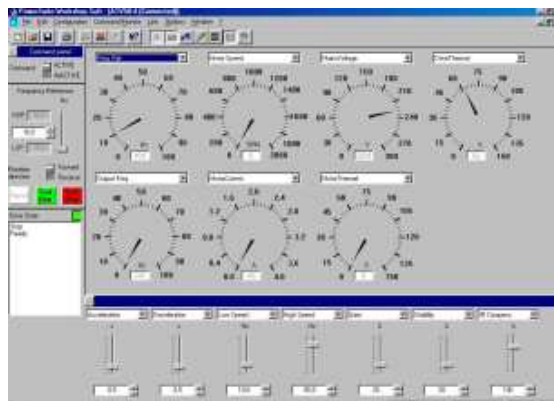
Funkcje przypisane kolejnym sygnałom logicznym:

-LI1 - obroty silnika w prawo „Low Speed”

-LI2 - obroty silnika w lewo „Low Speed”

-LI3 - „High Speed”

Zadawanie wartości wszystkich parametrów jak również monitorowanie pracy przemiennika możliwe jest dzięki oprogramowaniu PowerSuite. Za jego pomocą generowana jest konfiguracja pracy urządzenia, która następnie zostaje zapisana w jego pamięci. Podczas pracy, możliwe jest zadawanie sygnałów „START”, „STOP”, „LEWO”, „PRAWO” oraz zmiana zadawanych wartości prędkości „HIGH SPEED”, „LOW SPEED” oraz wartości przyspieszenia i hamowania. Program PowerSuite pozwala również na monitorowanie parametrów pracy przemiennika.



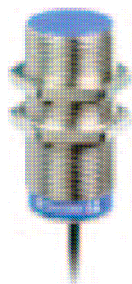
Rys. 2.4.7 Oprogramowanie PowerSuite

2.5. Detekcja – czujniki fotoelektryczne, indukcyjne, wyłączniki krańcowe

Do określenia położenia obiektu na taśmie, a tym samym etapu na jakim znajduje się wykonywany proces, wykorzystane zostały czujniki o dwustanowym wyjściu: indukcyjne (P11) oraz fotoelektryczne (P1, P3, P4, P5, P8 i P9).

Ich zasada działania polega na mierzeniu w sposób pośredni przetwarzanej wielkości fizycznej i generowaniu pod wpływem jej zmian sygnału zamykającego wewnętrzny obwód zasilania cewki wbudowanego przekaźnika. W wyniku tego, zmieniony zostaje stan ich styków, które otwierają się, w przypadku czujników ze stykiem NC lub otwierają w przypadku styków NO.

Czujniki indukcyjne.



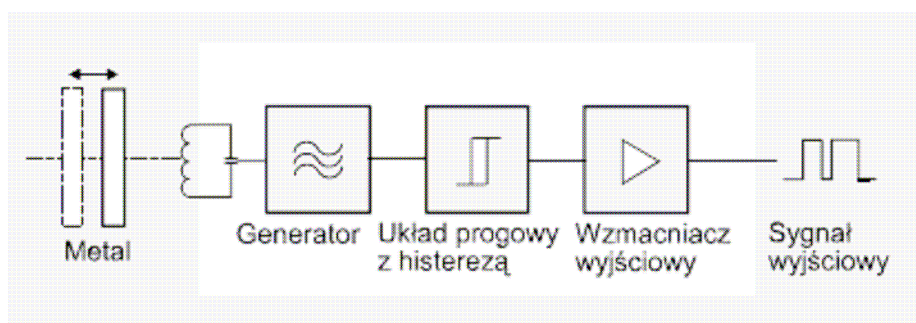
Rys.2.5.1 Czujnik indukcyjny

Indukcyjne czujniki zbliżeniowe są powszechnie stosowane w układach automatyki przemysłowej i sterowania do kontroli położenia, przemieszczeń i ruchu mechanizmów związanych ze sterowanymi urządzeniami. Reagują one na wprowadzanie metalu w strefę czułości, charakteryzując się dużą pewnością działania i niezawodnością.

Dwustanowy sygnał wyjściowy czujników umożliwia ich współpracę z programowalnymi sterownikami PLC lub bezpośrednie sterowanie pracą przekaźników, elektrozaworów i innych elementów wykonawczych.

Parametry techniczne i konstrukcja czujników umożliwia ich pracę w większości gałęzi przemysłu w układach sterowania prądu stałego lub prądu przemiennego.

Działanie czujników indukcyjnych polega na indukowaniu prądów wirowych w metalu zbliżanym do pola czułości czujnika. Indukowanie prądów wirowych wpływa na pole elektromagnetyczne, wytworzone wokół czoła czujnika, przez obwód indukcyjny generatora LC wysokiej częstotliwości. Układ progowy kontroluje amplitudę generowanego napięcia, która maleje wraz ze zbliżaniem metalu. Przełączenie wyjścia czujnika następuje po zbliżeniu metalu na odległość odpowiadającą punktom włączenia i wyłączenia.



Rys. 2.5.2 Czujnik indukcyjny – zasada działania.

Czujniki wykorzystane w układzie, charakteryzują się 10 mm nominalną strefą działania.

Czujniki fotoelektryczne



Rys. 2.5.3 Czujniki fotoelektryczne

Odbiorniki czujników fotoelektrycznych reagują na zanik wiązki emitowanej przez nadajnik, wywołany przemieszczaniem się wykrywanego obiektu.

Zaletą czujników optycznych są duże zasięgi działania uzyskiwane dla małych obudów czujników. Wykorzystuje się je m.in. do kontroli położenia ruchomych części maszyn, do identyfikacji obiektów znajdujących się w

zasięgu działania, np. przesuwających się na taśmach transportowych, do określania poziomu cieczy i materiałów sypkich, itp.

Duże znaczenie w czujnikach optycznych odgrywa długość fali świetlnej emitowanej przez nadajnik. W najczęściej wykorzystywane jest modulowane światło z zakresu bliskiej podczerwieni, dzięki czemu osiąga się małą wrażliwość czujników na widzialne światło z otoczenia. Dodatkowo poprzez wzajemną synchronizację nadajnika i odbiornika gwarantowana jest duża odporność czujników na zakłócenia i możliwość pracy w warunkach zanieczyszczenia powietrza i przy zabrudzeniu układu optycznego czujnika. Wytworzony w nadajniku silny impuls świetlny nawet osłabiony rozproszeniem dociera do odbiornika, jest wzmocniony i analizowany zapewniając poprawne działanie czujnika.

Zanieczyszczenie powietrza i zabrudzenie układu optycznego skraca jednak jego strefę działania.

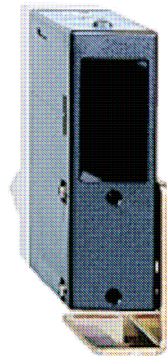
Czujniki optyczne są wyposażone w wysokiej jakości systemy soczewek optycznych, które dokładnie ukierunkowują promień świetlny w nadajniku i odbiorniku, umożliwiając realizację różnych funkcji zależnie od wykonania i przeznaczenia czujników.

Zastosowane w układzie czujniki pracują w systemie „bariera”. System ten polega na umieszczeniu nadajnika i odbiornika w oddzielnych obudowach, naprzeciw siebie i przeprowadzeniu między nimi wiązki światła. Przesłonięcie wiązki promieni świetlnych przez obiekt powoduje przerwanie transmisji i przełączenie obwodu wyjściowego czujnika. Ten typ systemu stosuje się dla identyfikacji obiektów nieprzezroczystych a także odbijających promienie świetlne.

Czujniki emitują wiązki laserowe lub z zakresu podczerwieni, a nominalne strefy działania wynoszą odpowiednio do 15m lub 100m.

W przypadku czujnika laserowego, po odpowiednim jego zogniskowaniu, istnieje możliwość wykrywania obiektów o rozmiarach 0,7mm.

Czujnikiem, wykorzystanym do identyfikacji położenia obiektu, jest również pojedynczy czujnik P5, działający w systemie odbiciowym.



Rys.2.5.4 Czujnik XUJK703538

Emituje on wiązkę światła z zakresu promieniowania podczerwieni, która odbijając się od umieszczonego na jej drodze obiektu ponownie trafiają do czujnika. Sygnał ten służy następnie do wygenerowania sygnału przełączającego wyjście czujnika.

Jedną z zalet tego czujnika, jest możliwość przetwarzania odległości (od 0 do 80 cm) pomiędzy czujnikiem a obiektem odbijającym wiązkę, na sygnał analogowy, zarówno prądowy (4-10mA) jak i napięciowy (0-10 V).

Programowalny czujnik P7, współpracujący z dwiema głowicami światłowodowymi, w zależności od układu w jakim są one ustawione, pozwala natomiast na rozróżnianie 3 zapamiętanych barw, w różnym odcieniu z ustawianą tolerancją dla obiektu znajdującego się do 3 cm (system odbiciowy), lub do rozróżniania 3 zapamiętanych odcieni przezroczystości, do 25 cm między głowicami (system „bariera”).



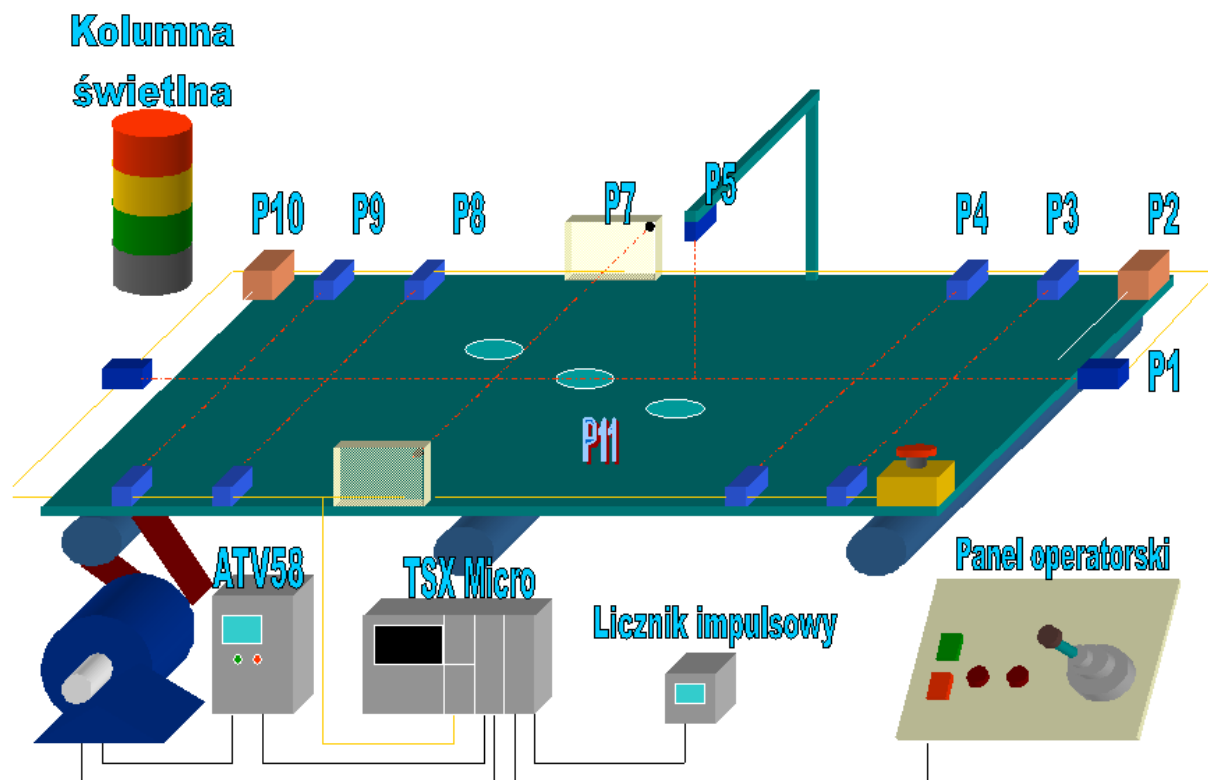
Wyłączniki krańcowe

Urządzeniami, uniemożliwiającymi w naszym układzie osiągnięcie przez poruszający się wraz z taśmociągami obiekt niedozwolonego położenia są wyłączniki krańcowe. Ich zasada działania polega na przełączaniu styków na skutek przemieszczenia sprzężonego z nimi mechanicznie elementu ruchomego wyłącznika. W zależności od ich rodzaju, na przełączanie może mieć wpływ zarówno wartość przemieszczenia jak również jego kierunek, siła, itp. Wyłączniki tego typu są powszechnie stosowane w obrabiarkach, dźwigach, windach, suwnicach, itp.



Rys. 2.5.6 Wyłącznik krańcowy

3. Algorytm sterowania pracą układu



Rys. 3.1 Algorytm sterowania pracą układu

Układ pracować może w dwóch trybach, trybie automatycznym oraz ręcznym. Wyboru trybu dokonujemy za pomocą przycisków „start” (tryb automatyczny), oraz „stop” (tryb ręczny) na panelu operatorskim.

Warunkiem koniecznym do uruchomienia procesu w trybie automatycznym jest pojawienie się obiektu na taśmociągu w miejscu przecięcia się wiązek laserowych par czujników P1 oraz P3.

Po wciśnięciu przycisku zasilania panelu operatorskiego (przycisk ON/OFF, zielony, oznaczony symbolem „I”), a następnie wciśnięciu przycisku „start” przemiennik zainicjuje pracę silnika, którego wirnik zacznie obracać się w lewo

z prędkością „Low Speed” aż do chwili wykrycia obiektu przez kolejną parę czujników P4, kiedy to prędkość obrotowa silnika zwiększona zostanie do wartości „High Speed”. Kiedy przesuwający się wraz z taśmociągami obiekt znajdzie się w pozycjach odpowiadających punktom rozmieszczenia trzech czujników indukcyjnych P11, silnik trzykrotnie zatrzyma się na 1 sekundę. Czujnik P7 wykryje kolor obiektu, natomiast sygnał wywołany zanikiem wiązki czujników P8 ponownie zmieni prędkość wirowania silnika do wartości „Low Speed”.

Sekwencja przesuwania się taśmociągu w lewo zostanie zakończona w chwili osiągnięcia przez obiekt położenia sygnalizowanego przez parę czujników P9, po czym układ natychmiast przejdzie do realizacji sekwencji powrotu. Zmieniony zostanie kierunek obrotów na prawy, oraz zwiększona prędkość obrotów silnika między pozycjami czujników P8-P4, aż do zatrzymania silnika i zakończenia całego cyklu po ponownym znalezieniu się obiektu w położeniu początkowym (para czujników P3). W sekwencji tej, na podstawie podanej wcześniej przez czujnik P7 informacji o kolorze obiektu, w przypadku kiedy był to kolor „A”, za pośrednictwem czujnika P5 przesłany zostanie sygnał do licznika impulsowego, zliczającego ilość cykli wykonanych z obiektami o tym kolorze. W przypadku koloru „B” cykle te nie będą zliczane, natomiast wykrycie koloru „C” przez czujnik P7 jeszcze w pierwszej fazie całego procesu (kierunek przesuwania w lewo) spowoduje natychmiastowy powrót do pozycji początkowej.

(Uwaga: praca w trybie automatycznym jest możliwa wyłącznie przy zasłoniętej wiązce czujników P1. W przypadku choćby krótkotrwałego wykrycia wiązki przez czujnik, cykl zostaje przerwany, a jego wznowienie jest możliwe dopiero po ponownym umieszczeniu obiektu, w pozycji początkowej.)

Aby przejść do trybu ręcznego, wystarczy w dowolnym momencie trybu automatycznego nacisnąć przycisk stop (panel operatorski musi być załączony – przyciski ON/OFF).



Rys. 3.2 Panel operatorski

Obsługa trybu ręcznego za pomocą panelu operatorskiego odbywa się przy użyciu manipulatora „RĘCZNE” umożliwiającego sterowanie obrotami silnika w prawo lub lewo. Przełącznik wyboru prędkości „R” musi wówczas znajdować się w pozycji lewej („High Speed”) lub prawej („Low Speed”). W pozycji środkowej sterowanie jest bowiem zablokowane.

(Uwaga: Po załączeniu zasilania, układ znajduje się w trybie ręcznym)

O rodzaju trybu, w jakim obecnie pracuje układ, informują nas lampki sygnalizacyjne, podświetlenia przycisków „start”, „stop”, oraz kolumna świetlna. Kolor czerwony sygnalizuje tryb ręczny, natomiast kolor zielony tryb automatyczny.

W przypadku wciśnięcia jednego z dwóch przycisków awaryjnych odłączone zostaje natychmiast zasilanie obwodu sterowania silnikiem, oraz włączony zostaje sygnał alarmowy. Na kolumnie świetlnej zapala się wówczas migające

światło żółte wraz z czerwonym lub tylko żółte, sygnalizując tryb podczas którego awaria ta nastąpiła (ręczny czy automatyczny).

Wybór trybów jest możliwy również z pozycji panelu znajdującego się na wysięgniku. Przełączenia między trybem ręcznym oraz automatycznym dokonujemy za pomocą przycisków „O” oraz „I” po wcześniejszym załączeniu panelu ustawiając piórko przełącznikiem dwupozycyjnego w pozycji górnej.

(Uwaga: obsługa z pozycji panelu znajdującego się na wysięgniku ma wyższy priorytet niż obsługa z pozycji panelu operatorskiego pokazanego na rys. 3.2. Oznacza to, iż w momencie gdy panel znajdujący się na wysięgniku jest załączony – piórko przełącznika dwupołożeniowego znajduje się w pozycji górnej – panel operatorski z rys. 3.2 jest nieaktywny. Sygnalizuje on jedynie tryb pracy, za pomocą lampek sygnalizacyjnych).

4. Programowanie sterownika za pomocą PL7 Micro

4.1. Struktury jednozadaniowe i wielozadaniowe

Do programowania sterownika wykorzystane zostało oprogramowanie PL7 Micro umożliwiające wykorzystanie następujących języków:

- graficznego Ladder (LD), służącego do definiowania diagramów drabinkowych, przeznaczonych szczególnie do przetwarzania kombinacyjnego, wykorzystującego „styki”, „cewki” oraz bloki,
- logicznego List (IL), stanowiącego kod maszynowy dla zapisywania logicznych i numerycznych instrukcji przetwarzania,
- graficznego Grafcet, służącego do graficznego i strukturalnego przedstawiania sekwencji systemu sterowania.

Języki te zawierają zdefiniowane wstępnie bloki funkcyjne (zegary, liczniki, itp.) i wszystkim obiektom umożliwiają przypisywanie nazw własnych lub symboli.

Struktura programu może posiadać jedną z dwóch form: jednozadaniową lub wielozadaniową.

Struktura jednozadaniowa jest strukturą domyślną. Program zawiera jedno zadanie - główne, które może być wykonywane cyklicznie (ustawienie domyślne) lub okresowo. W przypadku działania cyklicznego kolejne "przebiegi" zadania są łączone jeden z drugim, bez przerw. Przy działaniu okresowym "przebiegi" zadania następują po sobie w zdefiniowanym przez użytkownika okresie.

Struktura wielozadaniowa pozwala natomiast na zastosowanie aplikacji pracujących w czasie rzeczywistym, w których każdej aplikacji przypisany jest specjalny program. Każdy z tych programów jest zarządzany przez zadanie. Zadania te są w stosunku do siebie niezależne i są wykonywane

równolegle przez główny procesor, który decyduje zarówno o ich priorytecie jak i o ich wykonaniu. Zaletami tego typu struktury są: optymalne wykorzystanie mocy procesora, uproszczenie projektowania (każde zadanie zapisuje się i poprawia niezależnie od pozostałych), nadanie aplikacji określonej struktury (każde zadanie pełni unikatową funkcję), optymalizacja dostępności.

System wielozadaniowy może składać się z zadania głównego MASTER, szybkiego FAST oraz 8 - 64 (w zależności od rodzaju procesora) zadań wyzwalanych zdarzeniami.

Zadanie szybkie (opcjonalne) jest zadaniem okresowym służącym do realizacji szybkich operacji przetwarzania z wyższym priorytetem w stosunku do zadania głównego MASTER.

Zaprogramowane zadanie jest automatycznie uruchamiane przez system podczas inicjacji, a następnie może być zatrzymywane oraz uruchamiane za pośrednictwem odpowiedniego bitu systemowego.

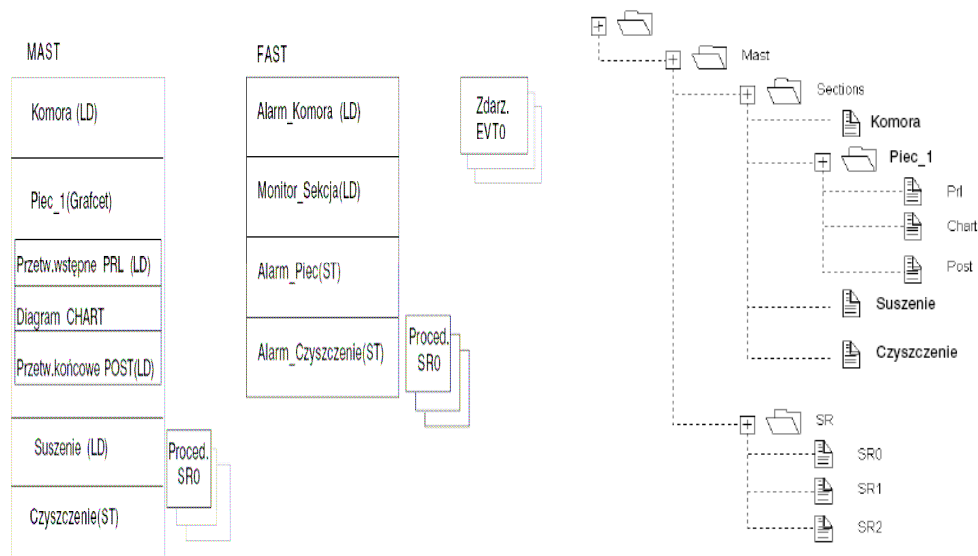
Zadania wyzwalane zdarzeniami w odróżnieniu od zadania opisanego powyżej nie są okresowe, lecz wyzwalane (uruchamiane) przez pewne moduły. Mają one najwyższy priorytet, a ich przetwarzanie odbywa się na tyle szybko, by nie zakłócić wykonywania innych zadań.



Rys. 4.1.1 Zadania główne, szybkie, wyzwalane zdarzeniami oraz ich priorytet.

Każde z zadań może składać się z kilku sekcji, przy czym każda z nich może być napisana w innym, najlepiej odpowiadającym jej języku. Taki podział

oznacza, że możliwy jest do stworzenia program strukturalny oraz można z łatwością generować lub też dołączać moduły programu. Struktura taka umożliwia również tworzenie procedur (podprogramów), które mogą być wywoływane z dowolnej sekcji zadania do którego należą lub z innych procedur tego samego zadania.



Rys. 4.1.2 Przykład wielozadaniowego programu strukturalnego. Każdemu z 3 zadań przyporządkowane są sekcje napisane w różnych językach wraz z odnoszącymi się do nich procedurami.

4.2 Adresowanie

Głównymi obiektami logicznymi, do których możemy odwoływać się podczas programowania sterownika są:

- bity I/O stanowiące logiczne odwzorowanie elektrycznych stanów wejść i wyjść

- bity wewnętrzne %Mi służące do zapamiętywania stanów pośrednich w trakcie wykonywania programu
- bity systemowe %S0 do %S127 umożliwiające monitorowanie poprawności działania sterownika oraz przebiegu realizacji programu,
- bity bloków funkcyjnych których wartości odpowiadają stanom wyjść bloków,
- bity wydzielone ze słów
 - bity statusu kroków języka Grafset.

(Poniżej przedstawione zostaną sposoby adresowania obiektów do których odwoływano się podczas pisania programu w aplikacji).

Adresy podstawowych słów oraz bitów obiektów dla modułów I/O (We/Wy) definiuje się w następujący sposób:

%Rodzaj_obiektuFormatPozycja.Nr_kanału

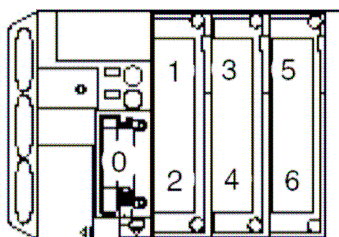
Gdzie:

Rodzaj_obiektu – (I lub Q): Fizyczne wejścia i wyjścia modułu dokonującego wymiany w sposób ukryty (niejawny) przy każdym wykonaniu zadania, do którego są one podłączone.

Format –(X, W lub D):X (bajt), W (słowo), D (słowo podwójnej precyzji).

Dla obiektów zapisanych w formacie logicznym, symbol X można pomijać.

Pozycja i Nr_kanału – Sterowniki TSX` Micro 37 mają budowę modułową 2-adresową. Pozycja każdego z modułów sterownika pokazano poniżej:



Rys. 4.2.1 Numery pozycji poszczególnych modułów sterownika

Moduły standardowe adresuje się jak dwa moduły pół formatowe.

(np. moduł zawierający 64 We/Wy jest widziany jak dwa moduły: 32-wejściowy moduł umieszczony w slotie 5 oraz 32-wyjściowy, umieszczony w slotie 6.)

Adresowanie słów używanych w języku PL7 (z wyjątkiem słów sieciowych oraz słów bloków funkcyjnych) przedstawia się w podobny sposób:

%Rodzaj_obiektuFormatNr

Rodzaj obiektu – (M, K,S)

M - słowa wewnętrzne, czyli te, w których zapisywane są wartości w trakcie wykonywania programu. Słowa te są zapamiętywane w strefie danych pamięci.

K - słowa stałe służące do przechowywania wartości stałych i komunikatów alfanumerycznych. Ich zawartość może być zapisywana i modyfikowana jedynie przez terminal. Słowa te są przechowywane w tym samym miejscu co program, i mogą one wykorzystywać również pamięć FLASH EPROM.

S - słowa systemowe pełnią różne funkcje. Część z nich zawiera informacje o statusie, czasie działania systemu bądź aplikacji, itp. a pozostałe mogą służyć do wykonywania odpowiednich operacji.

Format – (S, B, W, D, F)

B - bajt: ten format jest używany wyłącznie do działania na łańcuchach słów.

W - słowo pojedynczej precyzji: te 16-bitowe słowa mogą zawierać wartości algebraiczne z przedziału od -32 768 do 32 767.

D – słowa podwójnej precyzji: te 32-bitowe słowa mogą zawierać wartości algebraiczne z przedziału od - 2 147 483 648 do 2 147 483 647. Są one

zapisywane do pamięci w dwu następujących po sobie słowach pojedynczej precyzji.

F – słowo zmiennoprzecinkowe.

4.3 Języki programowania

4.3.1 Język Ladder

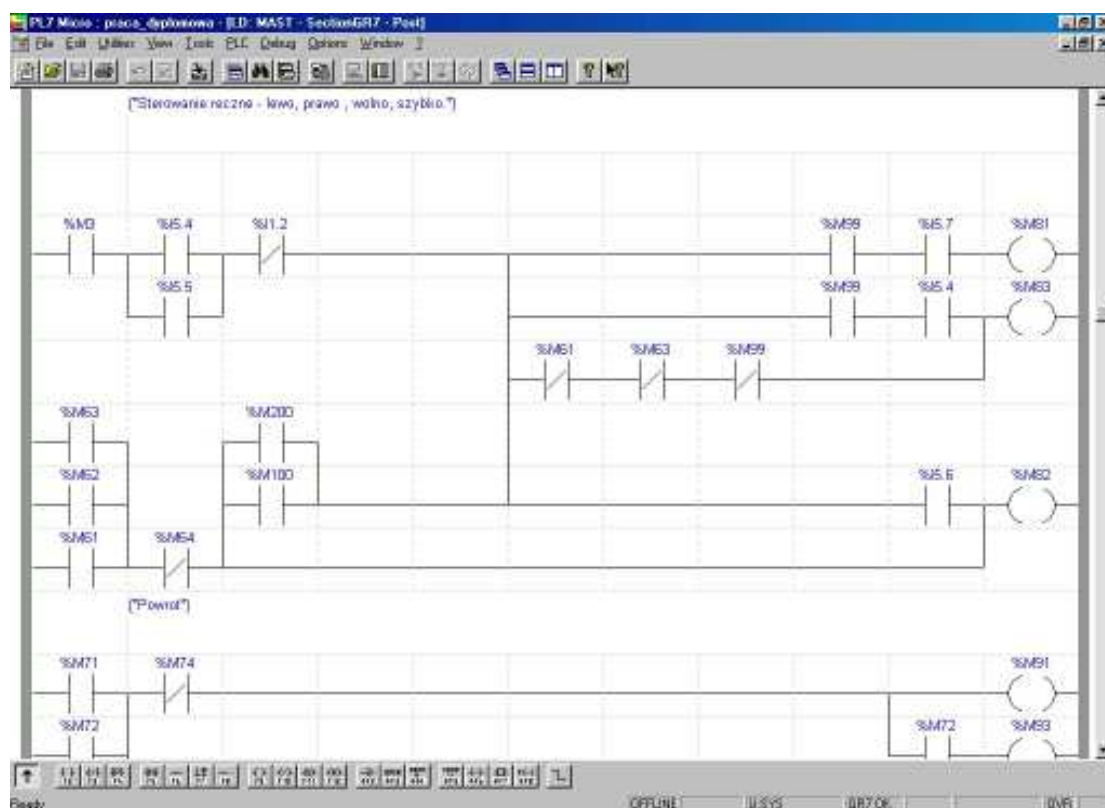
Programy napisane w języku Ladder składają się z szeregu segmentów oraz podsieci (labeled) wykonywanych sekwencyjnie przez sterownik.

Labelkę tworzą elementy graficzne ograniczone z dwóch stron liniami bazowymi (w notacji elektrycznej odpowiadają one zasilaniu).

W zależności od tego jakie przypisane zostaną do nich adresy, mogą one odwzorowywać:

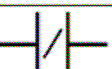
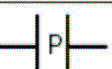
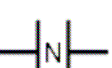
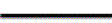
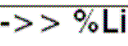
- wejścia i wyjścia sterownika (przyciski, czujniki, przekaźniki, kontrolki, itp.),
- standardowe funkcje systemu sterującego (zegary, liczniki, itp.),
- operacje arytmetyczne, logiczne i specjalne,
- zmienne wewnętrzne sterownika.

Elementy graficzne łączone są ze sobą za pomocą poziomych i pionowych linii. Każda labelka może składać się maksymalnie z 7 linii i 11 kolumn podzielonych na dwie strefy: strefę warunków, definiującą warunki niezbędne dla wykonania określonej akcji, oraz strefę operacji w której zdefiniowane są operacje, wykonywane po przetestowaniu warunków.



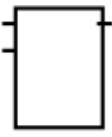
Rys. 4.3.1 Podsieć (labelka) w języku Ladder

Elementy wykorzystywane przy programowaniu w języku Ladder, zestawiono w poniższej tabeli:

Grupa elementów		Symbol	Funkcja
Elementy warunków	• Styk normalnie otwarty		Styk się zamyka, gdy kontrolujący go bit obiektowy ma wartość 1.
	• Styk normalnie zamknięty		Styk jest zamknięty, gdy kontrolujący go bit obiektowy ma wartość 0.
	• Styki reagujące na zmianę sygnału		Zbocze narastające: styk zamknięty, gdy bit obiektowy zmienia stan z 0 na 1.
			Zbocze opadające: styk zamknięty, gdy bit obiektowy zmienia stan z 1 na 0.
Elementy łączące	• Łącznik poziomy		Służy do szeregowego łączenia elementów graficznych pomiędzy liniami bazowymi.
	• Łącznik pionowy		Służy do łączenia równoległego elementów graficznych.
Elementy strefy akcji	• Sprzężenie proste		Nadaje bitowi sprzężonemu z obiektem, wartość wynikającą ze strefy warunków.
	• Sprzężenie odwrotne		Nadaje bitowi odwrotną wartość w stosunku do wartości ze strefy warunków.
	• Przelącznik prosty		Nadaje sprzężonemu bitowi wartość 1, gdy wynikiem testowania warunków jest 1.
	• Przelącznik odwrotny		Kasuje do 0 bit sprzężony z obiektem, gdy wynikiem testowania warunków jest 1.
	• Skok (<i>JUMP</i>) warunkowy do innej labelki		Umożliwia połączenie z labelką o znanej etykiecie, zarówno do przodu, jak i do tyłu. Skoki dopuszczalne są tylko w obrębie jednolitej części programu (program główny, procedura, itd.).
			Wykonanie skoku powoduje: • przerwanie realizacji aktualnej labelki, • wykonanie wywołanej labelki, • część programu między labelką ze skokiem a labelką docelową skoku nie jest wykonywana.
	• Sprzężenie warunkowe		Z języka Grafcet, używane gdy określa się warunki związane z przejściem bramki (<i>transition</i>). Umożliwia przejście do kolejnego kroku.

Grupa elementów		Symbol	Funkcja
Elementy strefy akcji (c.d.)	• Wywołanie procedury (CALL)		Powoduje przejście do wykonania procedury, gdy wynikiem testowania warunków akcji jest wartość 1. Wywołanie procedury powoduje: • przerwanie wykonywania bieżącej labelki, • wykonanie procedury, • powrót do wykonywania labelki.
	• Powrót do modułu	<RETURN>	Zarezerwowany dla procedur (SR). Umożliwia powrót do modułu, z którego procedura została wywołana, gdy wynikiem testu jest 1.
	• Zatrzymanie programu	<HALT>	Zatrzymuje wykonywanie programu, gdy warunek przyjmuje wartość 1.

Standardowe bloki funkcyjne

Grupy elementów		Symbol	Function
Elementy strefy warunków	• Bloki: Zegar Licznik Przerzutnik Rejestr Bęben		Każdy ze standardowych bloków posiada wejście lub wyjście, umożliwiające połączenie go z innymi elementami graficznymi. Funkcje poszczególnych bloków są opisane w części B.

4.3.2 Język tekstowy List

Program napisany w języku LIST składa się z szeregu instrukcji wykonywanych sekwencyjnie przez sterownik.

Przykładowa instrukcja wygląda w sposób następujący:

LD %I1.0

/ \

Kod instrukcji Argument

Każda instrukcja składa się z kodu instrukcji i argumentu.

Instrukcje te oddziałują na:

- obiekty I/O sterownika (przyciski, czujniki, przekaźniki, kontrolki, itd.),
- standardowe funkcje sterujące (zegary, liczniki, itd.),
- operacje arytmetyczne i logiczne oraz operacje związane z transferem danych,
- zmienne wewnętrzne sterownika.

Można wyróżnić dwa rodzaje instrukcji:

- instrukcje definiujące warunki, których spełnienie warunkuje wykonanie akcji, np. LD, AND, OR, itd.
- instrukcje definiujące akcje jakie ma wykonać sterownik w razie spełnienia zdefiniowanych warunków, np. ST, STN, R, itd.

Podobnie, jak w języku Ladder instrukcje zestawiane są w ciągu instrukcji (odpowiadające stopniom labelek) zwane sekwencjami. Każda z sekwencji zawiera co najmniej jedną instrukcję definiowania warunków, która jest przyporządkowywana (a w zasadzie jej wynik) co najmniej jednej instrukcji definiowania operacji. Każda z instrukcji zajmuje jedną linię.

Sekwencja rozpoczyna się wykrzyknikiem (generowanym automatycznie) i mogą być rozbudowane o komentarz lub etykietę, umożliwiające jej identyfikację.

Przykładowa sekwencja instrukcji wraz z opisem i etykietą:

! (*Oczekiwanie na sygnał wejściowy*)

%L2:

LD %I0.1

AND %M10

ST %Q2.5

Instrukcje podstawowe wykorzystywane przy programowaniu w języku List, oraz ich odpowiedniki w języku Ladder zestawiono w poniższej tabeli:

Grupa	Instrukcje	Funkcje równoważne
Instrukcje warunków	• LD, LDN, LDR, LDF	
	• AND, ANDN, ANDR, ANDF	
	• OR, ORN, ORR, ORF	
	• AND(, OR((8 poziomów nawiasów)	
	• XOR, XORN, XORR, XORF nierównoważność (<i>exclusive OR</i>)	
	• MPS MRD MPP	
	• N	Negacja
Instrukcje akcji	• ST, STN, S, R	
	• JMP, JMPC, JMPCN	Realizują skok (bezwarunkowy, warunkowy, gdy wynikiem jest 0 lub warunkowy, gdy wynikiem jest 1) do instrukcji o podanej etykiecie zarówno do przodu, jak i do tyłu programu.
	• SRn RET, RETC, RETCN	Umożliwia wywołanie wybranej procedury (SR). Powrót z procedury do programu (bezwarunkowy, warunkowy, gdy wynikiem jest 1 i warunkowy, gdy wynikiem jest 0).
	• END, ENDC, ENDCN HALT	Koniec programu (bezwarunkowy, warunkowy, gdy wynikiem jest 1 i warunkowy, gdy wynikiem jest 0). Zatrzymanie wykonywania programu.

Instrukcje związane z blokami funkcyjnymi:

Instrukcje związane z blokami funkcyjnymi

Grupa	Instrukcje	Funkcje
Elementy warunków	• Bloki: Zegar <i>Timer</i> Licznik <i>Counter</i> Przerzutnik <i>Monostable</i> Rejestr <i>Register</i> Bęben <i>Drum controller</i>	Są to instrukcje sterujące działaniem wszystkich standardowych bloków funkcyjnych. Wejścia, wyjścia, I/O bloków funkcyjnych są podłączane bezpośrednio w formie ustrukturyzowanej.

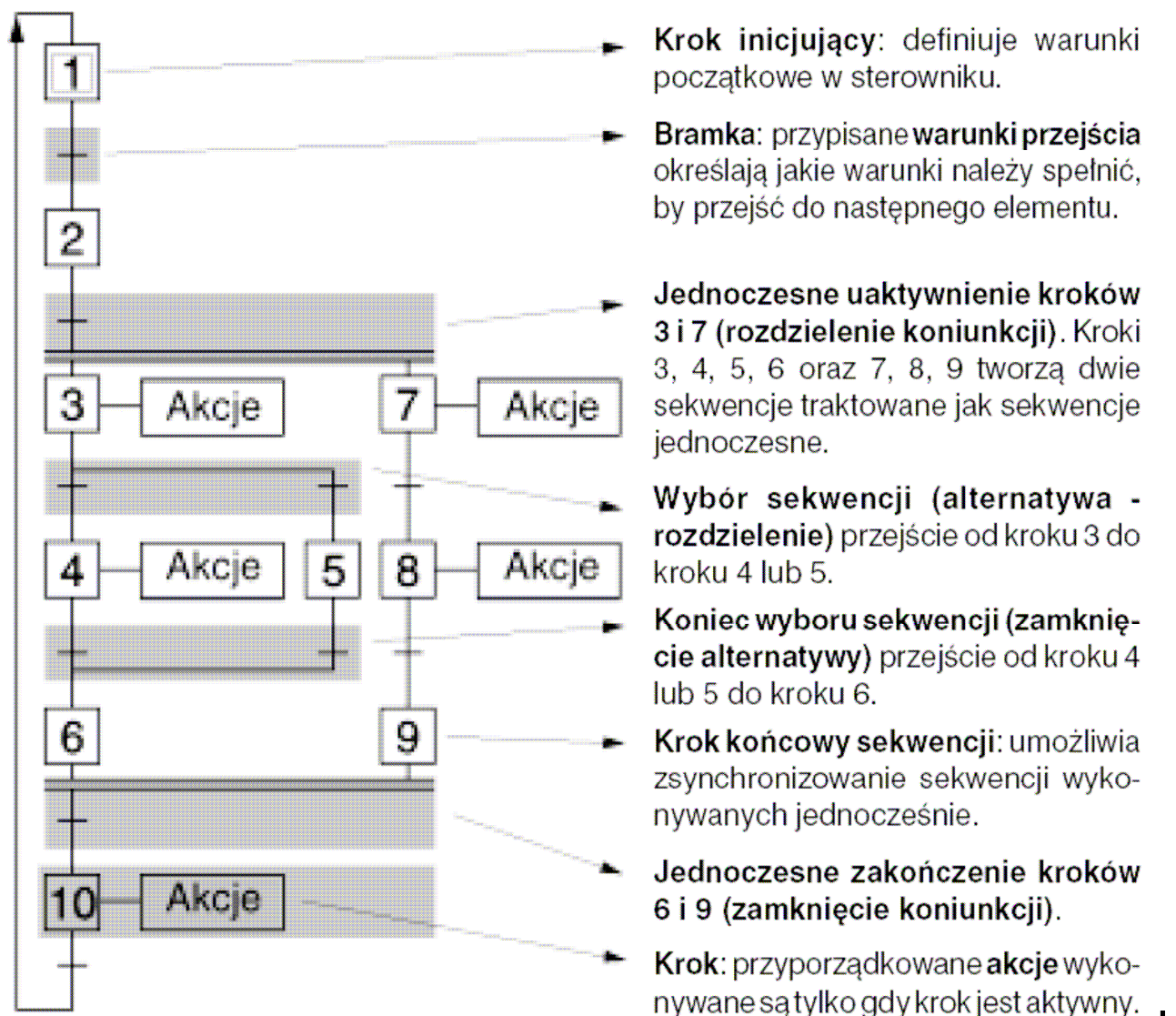
Instrukcje numeryczne:

Grupa	Instrukcje	Funkcje
Elementy warunków	• LD[.....] AND[.....] OR[.....] Przykład: LD[%MW10<1000]	Służy do porównywania dwóch argumentów. Po dokonaniu porównania i spełnieniu warunku wyjście zmienia stan na 1. Gdy %MW10<1000, to wyjście zmienia stan na 1.
Elementy akcji	• [.....] Przykład: [%MW10:=%MW0+100]	Wykonanie operacji arytmetycznych, logicznych. Składnia strukturalnego języka ST (patrz cz. B). Wynik operacji %MW0+100 zapisywany jest w słowie wewnętrznym %MW10.

4.3.3 Język Grafcet

Język Grafcet stosowany jest do odwzorowywania działania sekwencyjnego systemu sterującego w formie graficznej.

Opis graficzny działania systemu oraz występujących w trakcie sterowania sytuacji realizuje się za pomocą prostych symboli graficznych:



Bramki i łączniki reprezentują w formie symbolicznej dopuszczalne drogi realizacji **aktywnych kroków**.

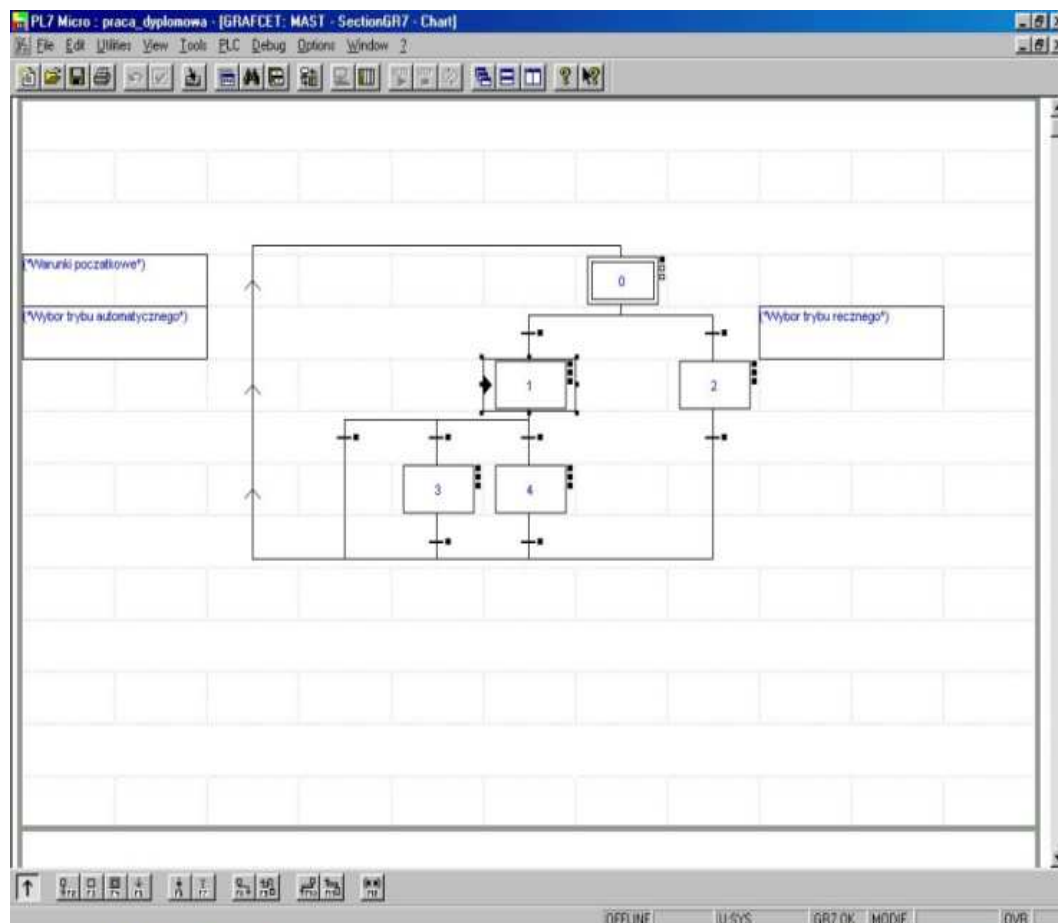
Akcje przyporządkowane krokom odzwierciedlają instrukcje wykonywane gdy krok zostanie uaktywniony. W szczególności opisują one rozkazy, jakie mają być wysłane do części wykonawczej systemu (wykonującego zautomatyzowany proces) lub innego zautomatyzowanego systemu. Zdefiniowanie, w dowolnym momencie, kroków aktywnych rzutuje na sytuację całego diagramu Grafcet.

Każdemu krokowi przyporządkowuje się akcje (operacje) zaprogramowane w języku Ladder lub języku instrukcji LIST. Akcje te są wykonywane tylko wtedy, kiedy sprzężony z nimi krok jest aktywny. PL7 posiada trzy rodzaje akcji:

- **akcje na skutek uaktywnienia:** wykonywane w momencie, gdy następuje uaktywnienie kroku,
- **akcje na skutek dezaktywacji:** wykonywane w momencie dezaktywacji kroku,
- **akcje ciągle:** wykonywane w sposób ciągły przez cały czas aktywności sprzężonego z nimi kroku.

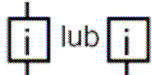
Akcje tych trzech rodzajów można stosować dla każdego kroku.

Pojedyncza akcja składa się z kilku elementów programowych (sekwencje, wyrażenia, labelki).

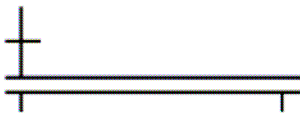


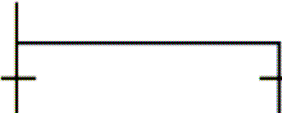
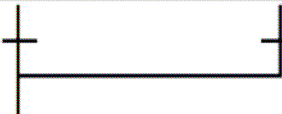
Rys. 4.3.2 Diagram Grafcet

Symbole graficzne wykorzystywane przy tworzeniu diagramów Grafcet zestawiono w poniższej tabeli:

Oznaczenie	Symbol	Funkcja
Krok inicjujący		Oznacza kroki aktywne na początku cyklu po inicjacji lub zimnym starcie.
Krok pojedynczy		Oznacza, że system sterujący jest stabilny. Maksymalna liczba kroków jakie można skonfigurować: - 1 do 96 dla TSX 37-10, - 1 do 128 dla TSX 37-20, - 1 do 250 dla TSX 57. Maksymalna liczba jednocześnie aktywnych kroków jest konfigurowana.

Bramka		Bramka realizuje przejście od jednego kroku, do drugiego. Związany z bramką warunek służy do określania warunku logicznego niezbędnego do przejścia do następnego kroku. Maksymalna liczba bramek wynosi 1024. Liczba ta nie podlega konfigurowaniu. Konfiguruje się natomiast maksymalną liczbę bramek jednocześnie otwartych.
--------	---	--

Oznaczenie	Symbol	Funkcja
Początek koniunkcji		Jest to przejście od jednego kroku do kilku kroków. Umożliwia jednocześnie uaktywnienie maksymalnie 11 kroków.
Koniec koniunkcji		Przejście od kilku jednocześnie otwartych kroków do jednego kroku. Umożliwia jednocześnie zamknięcie maksymalnie 11 kroków.

Początek alternatywy		Przejście od jednego kroku do kilku innych. Umożliwia wybór maksymalnie 11 sekwencji kroków.
Koniec alternatywy		Przejście od kilku kroków do jednego. Zamyka jedną z maksymalnie 11 sekwencji kroków.
Łącznik źródłowy		'n' jest numerem kroku, od którego przychodzi połączenie (krok źródłowy).
Łącznik docelowy		'n' numerem kroku, do którego ma nastąpić przejście (krok docelowy).
Łączniki bezpośrednie: • do góry • w dół • w lewo lub w prawo	  	Te łączniki stosowane są do wyboru sekwencji, do przeskoczenia jednego lub kilku kroków, do powtórzenia kroków (sekwencji).

Program zapisany w języku Grafcet można podzielić na trzy kolejne sekcje:

- przetwarzanie wstępne *Prl*,
- przetwarzanie sekwencyjne *Chart*,
- przetwarzanie końcowe *Post*.

Przetwarzanie wstępne wykonywane jest przed przetwarzaniem sekwencyjnym i przetwarzaniem końcowym i służy do przetwarzania wszystkich zdarzeń mogących mieć wpływ na zarządzanie systemem w razie zaniku zasilania oraz podczas reinicjacji, a także kasowaniu lub wstępnym zadawaniu parametrów diagramów Grafcet.

Przetwarzanie sekwencyjne stosowane jest natomiast do programowania sekwencyjnej struktury programu. Składa się ono z diagramu głównego

zapisanego na 8 stronach, w którym można zamieścić kilka autonomicznych diagramów, wykonywanych jednocześnie.

Podczas przetwarzania sekwencyjnego aktualizowane są dwie dedykowane tablice zawierające informacje o aktywności kroków i odblokowaniu bramek:

- **tablica aktywności kroków**, która przechowuje aktualne (dla bieżącego "przejścia" programu) informacje o krokach aktywnych, krokach które mają być uaktywnione oraz o krokach które mają być dezaktywowane,

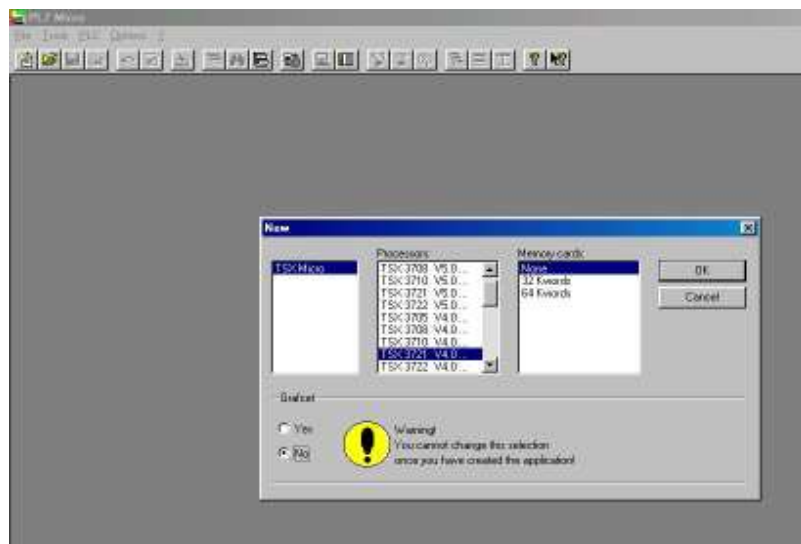
- **tablica odblokowania bramek**, która przechowuje aktualne (dla bieżącego "przejścia" programu) informacje o bramkach następujących po krokach uwzględnionych w poprzedniej tablicy.

Przetwarzanie końcowe jest przetwarzaniem wykonywanym na samym końcu cyklu, tuż przed zaktualizowaniem stanów wyjść. Służy do programowania ich logiki oraz wykonania akcji wygenerowanych w czasie przetwarzania sekwencyjnego.

4.4 Konfiguracja sterownika

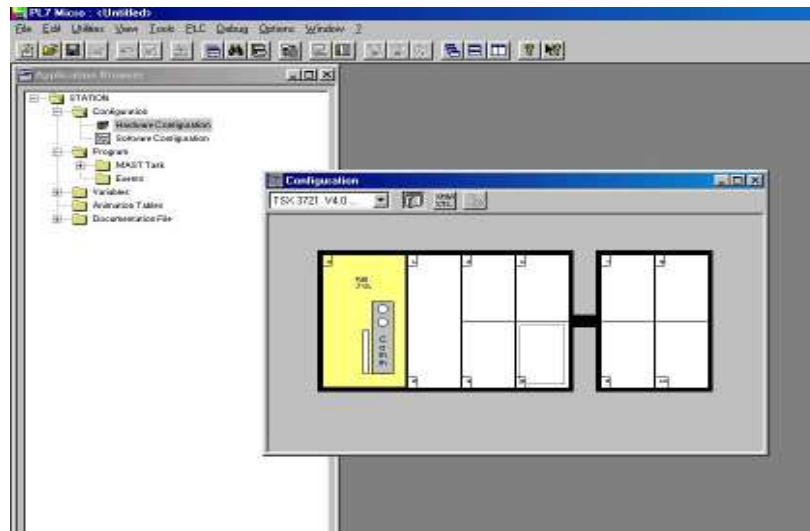
Aby skonfigurować sterownik, po uruchomieniu programu PL7 Micro należy utworzyć nową aplikację wybierając z menu górnego *File / New...*

W nowo otwartym oknie, należy wybrać rodzaj sterownika (w naszym wypadku jest to sterownik o procesorze TSX 3721 V4.0), pojemność karty pamięci jeśli sterownik jest w nią wyposażony, oraz czy podczas pisania nowego programu będziemy korzystać z języka Grafcet, czy też nie.

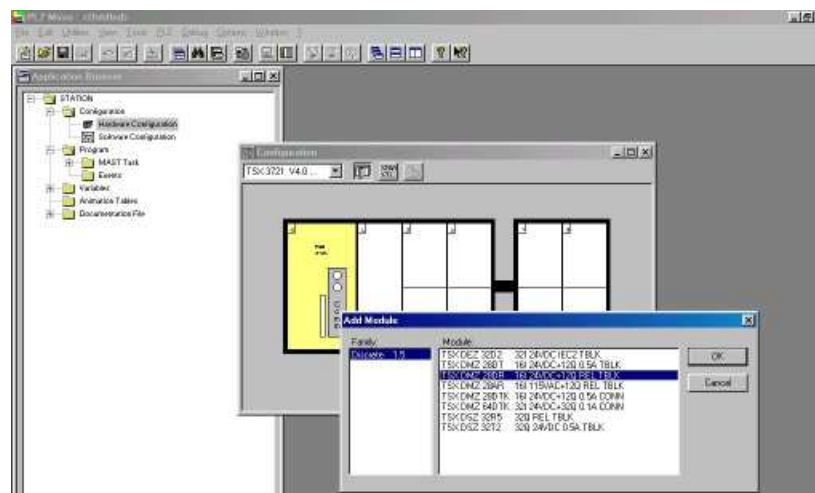


Rys. 4.4.1 Definiowanie sterownika

Po utworzeniu nowej aplikacji, w oknie *Application Browser* należy po rozwinięciu zakładki *Station/Configuration*, dwukrotnie kliknąć *Hardware Configuration*.



Rys. 4.4.2 Konfiguracja



Rys. 4.4.3 Przypisywanie modułów do poszczególnych gniazd sterownika

Klikając dwukrotnie lewym przyciskiem myszy na poszczególnych gniazdach, możemy dokonać wyboru modułów które zostały tam umieszczone. (Typ modułu można odczytać bezpośrednio z płyty czołowej każdego z nich.)

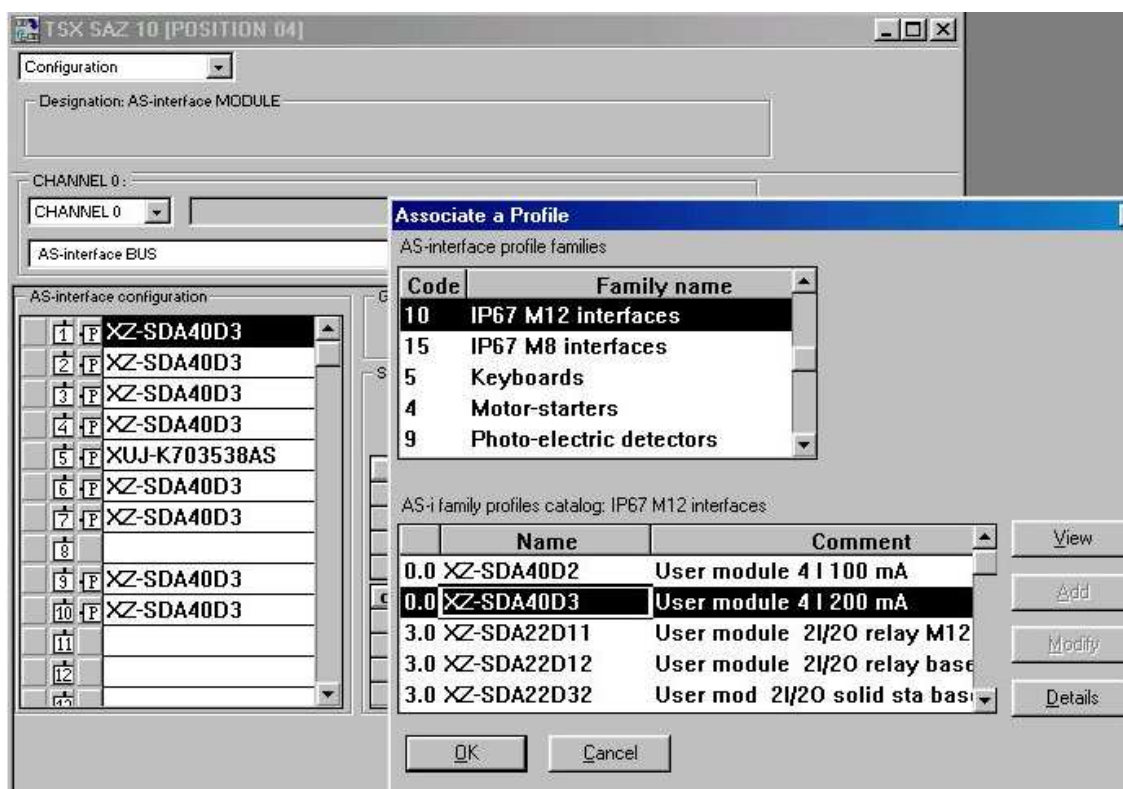
W przypadku zastosowania modułu komunikacyjnego TSX SAZ10 (magistrala As-i), po skonfigurowaniu gniazda 4, należy dodatkowo dokonać przypisania adresów Slave do poszczególnych urządzeń podłączonych do magistrali. Dzięki dołączonej do oprogramowania PL7 Micro biblioteki

produktów firmy Schneider, oraz biblioteki umożliwiającej zaadresowanie wszystkich innych urządzeń o standardowych parametrach As-i, adresowanie to jest bardzo proste.

Aby to uczynić należy dwukrotnie kliknąć na gniazdo 4 w którym skonfigurowany został wcześniej moduł komunikacyjny TSX SAZ10, a następnie w nowo otwartym oknie, ponownie za pomocą dwukrotnego kliknięcia lewym przyciskiem myszy, poszczególnym adresom (1-31) przypisać odpowiednie urządzenia, wybierając ich rodzaj i typ z listy.

(Moduły rozdzielaczy XZSDA40D3 możemy odnaleźć w rodzinie produktów *IP67 M12 interfaces*, natomiast czujnik XUJK703538AS *Photo-electric detectors*).

W przypadku omawianego układu, konfiguracja ta powinna wyglądać w sposób następujący:



Rys. 4.4.4 Przypisywanie adresów urządzeniom podłączonym do magistrali As-i

Oprócz konfiguracji magistrali, Program PL7 umożliwia nam również jej diagnostykę:



- 1 stan pracy modułu TSXSAZ10 (RUN, ERR, stan we/wy),
- 2 stan pracy kanału AS-i dołączonego do modułu,
- 3 uszkodzeniu slave'a,
- 4 danych związanych z dowolnie wybranym slave'm (profil, bity parametrów, wymuszenia itp.).

Rys. 4.4.5 Diagnostyka magistrali As-i

W przypadku uszkodzenia modułu AS-i lub uszkodzenia linii, jako druga może być dostępna diagnostyka „Window”, dająca wyjaśnienie prawdopodobnej przyczyny uszkodzenia.



Rys. 4.4. Diagnostyka „Window” w przypadku uszkodzenia jednego z urządzeń