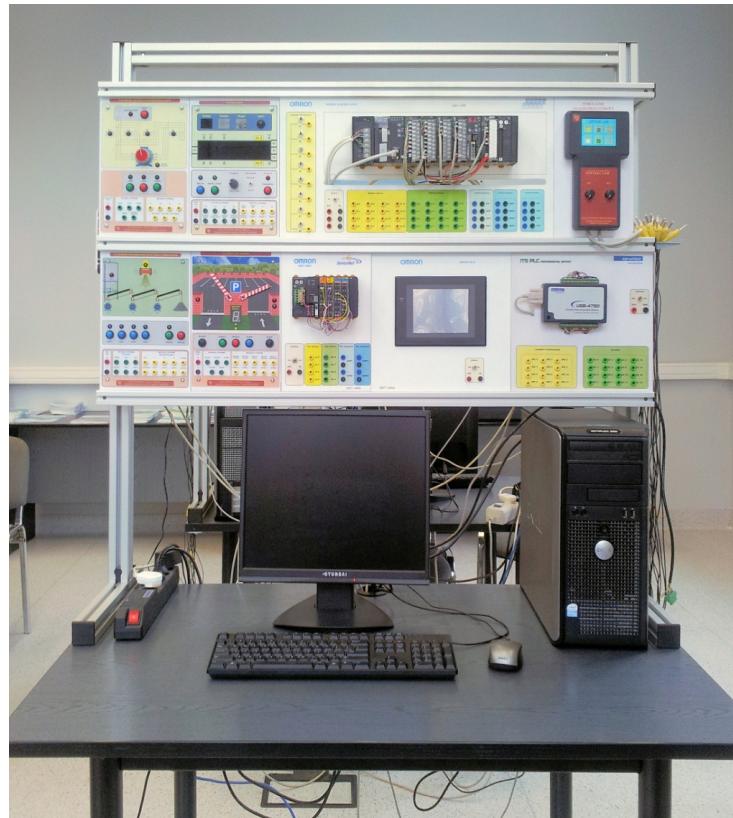




Politechnika Wrocławska

Instytut Maszyn, Napędów i Pomiarów Elektrycznych

Instrukcja stanowiskowa



LABORATORIUM STEROWANIA URZĄDZENIAMI I NAPĘDAMI PRZEMYSŁOWYMI

Spis treści

1. Laboratorium Sterowania Urządzeniami i Napędami Przemysłowymi	5
1.1. Wyposażenie laboratorium.....	5
1.2. Budowa stanowiska laboratoryjnego	6
2. Sterowniki programowalne OMRON z rodziny CJ	9
2.1. Rodzina sterowników OMRON	9
2.2. Sterownik CJ1M	10
2.2.1. Budowa i parametry sterownika	10
2.2.2. Jednostka centralna CJ1M CPU12-ETN	12
2.2.3. Mapa pamięci sterownika CJ1M	14
2.2.4. Alokacja modułów rozszerzeń.....	17
2.3. Wybrane moduły rozszerzeń sterowników serii CJ.....	19
2.3.1. Moduł wejść cyfrowych ID211	19
2.3.2. Moduł wyjść cyfrowych OD212.....	21
2.3.3. Moduł wejść analogowych AD041	23
2.3.4. Moduł wyjść analogowych DA041	25
3. Oprogramowanie narzędziowe dla sterowników OMRON.....	28
3.1. Pakiet programowy CX-One.....	28
3.2. CX-Programmer	31
3.2.1. Główne funkcje programu.....	31
3.2.2. Zakładanie i konfiguracja nowego projektu	37
3.2.3. Edycja programu w języku drabinkowym.....	40
3.2.4. Uruchamianie i testowanie programów.....	42
4. Wybrane instrukcje i funkcje języka drabinkowego dla sterowników OMRON ...	46
4.1. Instrukcje detekcji zboczy	46
4.1.1. Instrukcje DIFU i DIFD.....	46
4.1.2. Instrukcje stykowe różniczkujące	47
4.2. Elementy bistabilne	47
4.2.1. Zastosowanie instrukcji stykowych do realizacji funkcji przerzutników.....	48
4.2.2. Instrukcje SET i RSET.....	48
4.2.3. Instrukcja KEEP	49
4.3. Czasomierze	49
4.3.1. Instrukcje TIM, TIMH i TMHH.....	49
4.3.2. Czasomierz akumulujący TTIM	51
4.3.3. Zmiana formatu danych czasomierzy	52
4.3.4. Programowa realizacja funkcji czasowych typu TP i TOF	52
4.4. Liczniki	54
4.4.1. Licznik CNT	54

4.4.2. Licznik reweryjny CNTR.....	54
4.5. Komparatory	56
4.5.1. Komparatory CMP i CPS	56
4.5.2. Bezpośrednie instrukcje porównania.....	58
4.6. Generatory taktujące	59
4.6.1. Systemowe generatory taktujące.....	59
4.6.2. Programowa realizacja generatora taktującego	60
4.7. Operacje na danych	61
4.7.1. Instrukcje transferu danych	61
4.7.2. Operacje przesuwania bitów.....	64
4.7.3. Operacje logiczne	65
4.7.4. Operacje arytmetyczne	66
4.8. Instrukcje sterujące przebiegiem wykonania programu.....	68
4.8.1. Instrukcje skoków.....	68
4.8.2. Pętla FOR-NEXT	69
4.8.3. Realizacja podprogramów.....	70
5. Programowanie paneli operatorskich serii NS za pomocą oprogramowania CX-Designer.....	72
5.1. Zasady adresowania urządzeń w laboratorium	72
5.2. CX-Designer – konfiguracja i programowanie.....	72
6. Konfiguracja sieci przemysłowych w pakiecie CX-ONE.....	79
6.1. Konfiguracja sieci PROFIBUS	79
6.1.1. Przygotowanie tabeli routingu.	79
6.1.2. Konfiguracja modułów komunikacyjnych Profibus	81
6.1.3. Przykład sterowania zdalnymi modułami wejść/wyjścia.....	88
6.2. Konfiguracja sieci DeviceNet w pakiecie Cx-One.	89
6.2.1. Konfiguracja tabeli routingu.....	89
6.2.2. Konfiguracja modułu komunikacyjnego DeviceNet.....	92
6.2.3. Przykład sterowania zdalnymi modułami wejść/wyjścia.....	98

1. Laboratorium Sterowania Urządzeniami i Napędami Przemysłowymi

Laboratorium Sterowania Urządzeniami i Napędami Przemysłowymi, znajdujące się w Instytucie Maszyn, Napędów i Pomiarów Elektrycznych powstało w 2010 roku na podstawie porozumienia pomiędzy Politechniką Wrocławską a firmą OMRON Electronics sp. z o.o. W ramach współpracy obu firm, laboratorium zostało wyposażone w najnowszy sprzęt z branży automatyki przemysłowej, produkowany przez firmę OMRON w ostatnim czasie. Głównym celem współpracy było utworzenie branżowego centrum szkoleniowego dla specjalistów automatyków oraz podniesienie jakości kształcenia na Wydziale Elektrycznym Politechniki Wrocławskiej, na kierunku Automatyka i Robotyka.

1.1. Wypożyczenie laboratorium

W laboratorium znajduje się 6 niezależnych stanowisk szkoleniowych, wyposażonych w nowoczesne sterowniki programowalne typu CJ1M CPU-12-ETN. Sterowniki te charakteryzują się modułową konstrukcją, która zapewnia elastyczną konfigurację sprzętową, w zależności od stawianych wymagań funkcjonalnych. Wszystkie sterowniki posiadają odpowiednie moduły wejść-wyjść cyfrowych oraz analogowych, do których można dołączyć zewnętrzne czujniki i urządzenia wykonawcze. Każde stanowisko laboratoryjne wyposażone jest dodatkowo w specjalne makiety dydaktyczne, które pozwalają na zasymulowanie działania różnych urządzeń oraz procesów przemysłowych, oddając ich realistyczny charakter. Dzięki temu możliwa jest nauka programowania rzeczywistych systemów automatyki przemysłowej oraz przetestowanie poprawności działania aplikacji w stanach awaryjnych.

Wszystkie sterowniki PLC wyposażone zostały w moduły komunikacyjne, które pozwalają na wymianę danych z różnymi urządzeniami zewnętrznymi, do których można zaliczyć znajdujące się na wyposażeniu laboratorium - stacje rozproszonych wejść-wyjść. Elastyczna budowa stanowisk laboratoryjnych umożliwia realizację rozmaitych konfiguracji połączeń sieciowych, pozwalając na zamodelowanie rozproszonych systemów sterowania, wykorzystujących najpopularniejsze standardy sieci przemysłowych, takie jak: PROFIBUS DP, DeviceNET, Ethernet.

Na wszystkich stanowiskach szkoleniowych znajdują się ponadto zaawansowane panele operatorskie typu NS5 i NS8, wyposażone w kolorowe ekrany

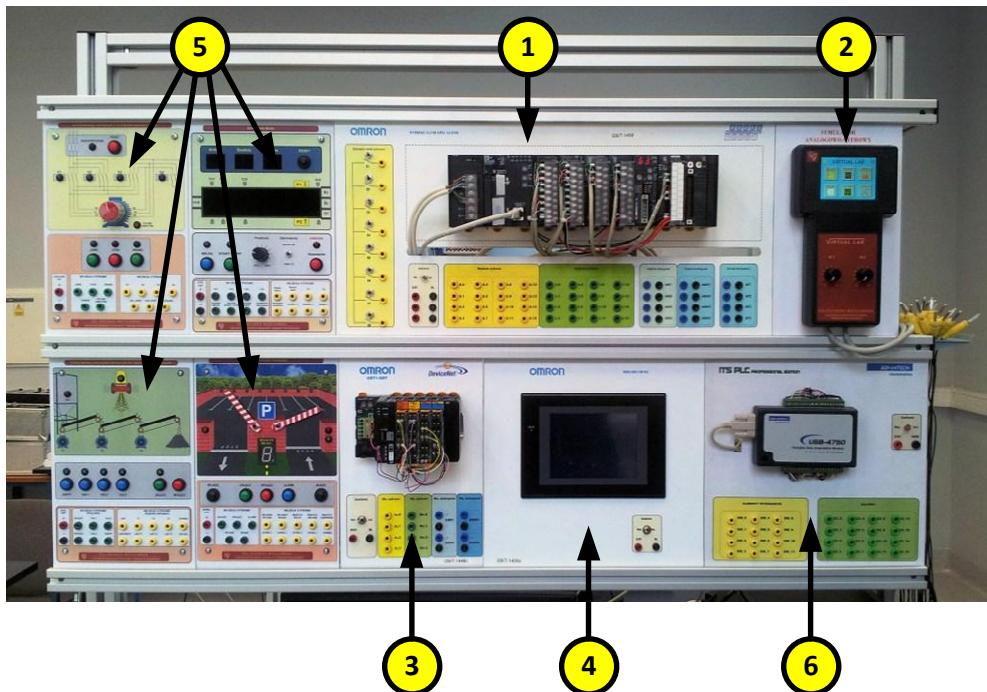
dotykowe. Panele te reprezentują grupę urządzeń klasy HMI (ang. *Human - Machine Interface*) i przeznaczone są do tworzenia graficznych interfejsów użytkownika i wizualizacji procesów przemysłowych.

Integralną częścią każdego stanowiska laboratoryjnego jest komputer PC z zainstalowanym pakietem oprogramowania inżynierskiego, który służy m.in. do: programowania, testowania, konfiguracji oraz diagnostyki podłączonych sterowników PLC. Ponadto, na każdym komputerze został zainstalowany program CX-Supervisor, który reprezentuje grupę programów typu SCADA (ang. *Supervisory Control and Data Acquisition*) i służy do projektowania i obsługi nadzorzących systemów sterowania i wizualizacji za pomocą komputera.

Aktualnie w Laboratorium Sterowania Urządzeniami i Napędami Przemysłowymi odbywają się regularnie specjalistyczne szkolenia tematyczne z zakresu programowania i obsługi sterowników programowalnych firmy OMRON na poziomie podstawowym i zaawansowanym.

1.2. Budowa stanowiska laboratoryjnego

Na rysunku 1.1 przedstawiono przykładową konfigurację jednego z sześciu stanowisk laboratoryjnych.



Rys.1.1. Przykładowa konfiguracja stanowiska laboratoryjnego

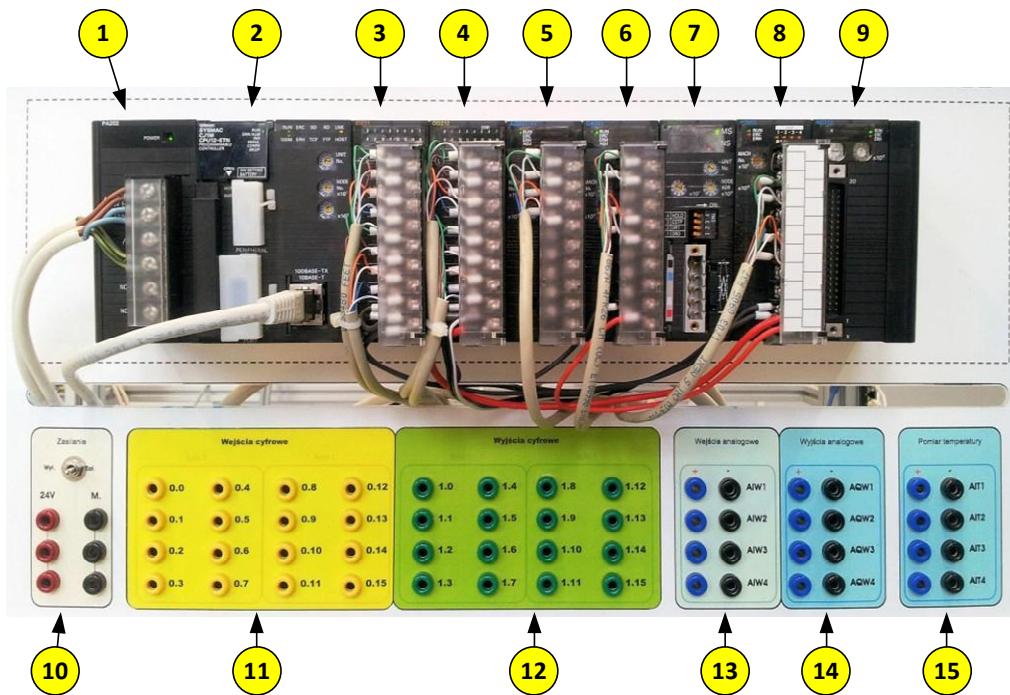
Konstrukcja mechaniczna każdego stanowiska laboratoryjnego pozwala na dowolną konfigurację sprzętową, poprzez wymianę poszczególnych paneli funkcjonalnych. Przykładowa konfiguracja z rys. 1.1 zawiera następujące komponenty:

1. Panel główny ze sterownikiem PLC o budowie modułowej;
2. Wielofunkcyjny symulator wejść/wyjść cyfrowych i analogowych;
3. Rozproszona stacja I/O – system OMRON SmartSlice;
4. Panel operatorski NS-5 z ekranem dotykowym
5. Zestaw wybranych modeli urządzeń i procesów przemysłowych;
6. Interfejs I/O wirtualnego symulatora procesów przemysłowych ITS PLC

Na rysunku 1.2 przedstawiono budowę panelu głównego ze sterownikiem CJ1M, na którym wyróżnić można następujące elementy:

1. Zasilacz PA202, dostarczający niezbędnych napięć dla sterownika PLC (+5V. +24V). Napięcie +24V zostało dodatkowo wyprowadzone na panelu zasilania (10), co umożliwia podłączenie zewnętrznych urządzeń małej mocy, m.in. modeli dydaktycznych.
2. Jednostka centralna sterownika CJ1M CPU12-ETN, zintegrowana z modułem komunikacyjnym w standardzie Ethernet.
3. Moduł ID211 – 16 wejść cyfrowych stałoprądowych o napięciu znamionowym +24V, skonfigurowanych do współpracy z sygnałami logicznymi w polaryzacji „PNP”
4. Moduł OD212 – 16 wyjść cyfrowych stałoprądowych o napięciu znamionowym +24V w konfiguracji PNP.
5. Moduł AD041 – 4 wejścia analogowe, skonfigurowane do pomiaru napięć w zakresie $\pm 10V$.
6. Moduł DA041 – 4 wyjścia analogowe napięciowe, o zakresie wyjściowym $\pm 10V$.
7. Moduł komunikacyjny DRM21 sieci DeviceNet. Na różnych stanowiskach laboratoryjnych występuje zamiennie z modułem PRM21 sieci PROFIBUS.
8. 4-kanałowy moduł PTS52 do pomiaru temperatury, współpracujący z czujnikami rezystancyjnymi typu Pt100.
9. Moduł NC113, zawierający wysokoczęstotliwościowe wyjścia impulsowe, przystosowany do sterowania napędów z silnikami krokowymi.

10. Panel zasilania – zawiera zaciski do podłączenia zewnętrznych urządzeń małej mocy o napięciu znamionowym +24V. Wyposażony jest w wyłącznik, który pozwala na szybkie rozłączenie obwodu zasilania.
11. Panel z gniazdami wejść cyfrowych – podłączenie sygnału o napięciu +24V powoduje wysterowanie wejścia w stan wysoki.
12. Panel z gniazdami wyjść cyfrowych – wysoki stan logiczny powoduje pojawienie się napięcia +24V na wyjściu.
13. Panel z gniazdami wejść analogowych (akceptuje sygnały napięciowe w zakresie $\pm 10V$).
14. Panel z gniazdami wyjść analogowych (generuje sygnały napięciowe w zakresie $\pm 10V$)
15. Panel z gniazdami dla 4 niezależnych czujników temperatury typu Pt100. Umożliwia pomiar temperatury w zakresie (-200÷650°C)

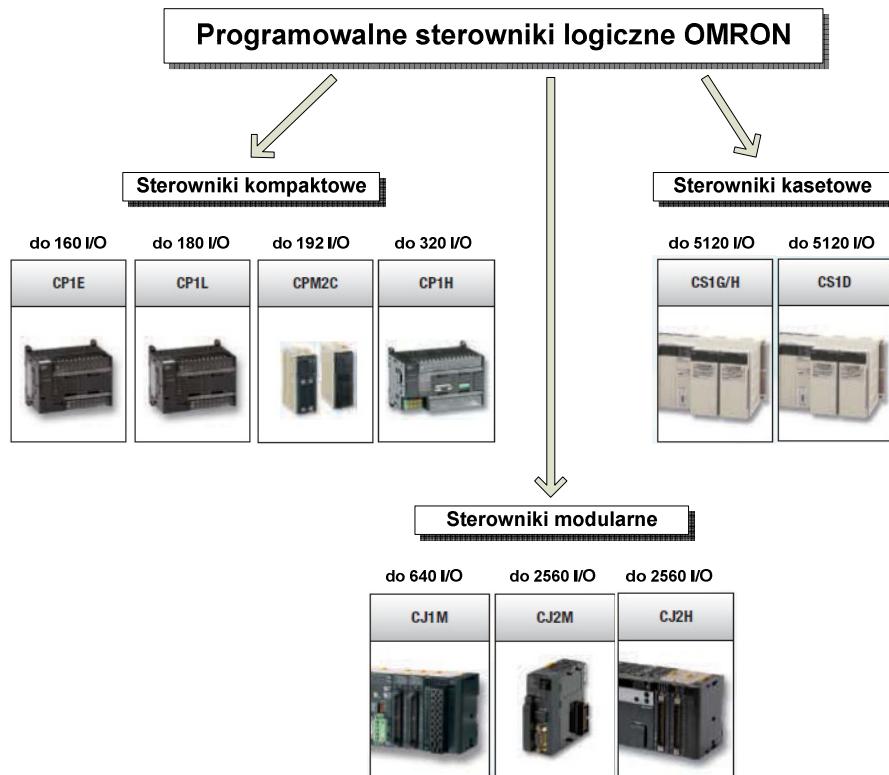


Rys.1.2. Fotografia głównego panelu ze sterownikiem PLC

2. Sterowniki programowalne OMRON z rodziny CJ

2.1. Rodzina sterowników OMRON

Na rysunku 2.1 przedstawiono klasyfikację aktualnie produkowanych sterowników OMRON.



Rys.2.1. Klasyfikacja sterowników PLC firmy OMRON

Firma OMRON obecnie jest jednym z największych producentów systemów i urządzeń automatyki przemysłowej na świecie. W swojej ofercie posiada kilkanaście modeli programowalnych sterowników logicznych, uszeregowanych w trzech grupach, w zależności od ich wielkości i możliwości funkcjonalnych:

- **sterowniki kompaktowe**, które posiadają zintegrowany zasilacz oraz wbudowane wejścia i wyjścia. Do grupy tej zalicza się zarówno proste, nierozszerzalne sterowniki 10-punktowe typu CPM1A, jak i zaawansowane sterowniki CP1H.
- **sterowniki modułowe serii CJ**, obsługujące do 2500 punktów I/O – seria sterowników elastycznie konfigurowalnych, posiadających szeroką gamę wymiennych modułów CPU, zestawów wejść-wyjść oraz modułów komunikacyjnych.
- **zaawansowane sterowniki serii CS1**, instalowane na płytach montażowych (kasetach), obsługujące do 5000 zmiennych procesowych, umożliwiające redundancję jednostek centralnych, znajdujące zastosowanie w przemysłowych systemach sterowania automatyki procesowej.

2.2. Sterownik CJ1M

2.2.1. BUDOWA I PARAMETRY STEROWNIKA

Sterownik programowalny CJ1M jest przedstawicielem grupy sterowników modularnych, należących do wysokowydajnej rodziny CJ1. W grupie tej oferowanych jest kilka modeli jednostek centralnych CPU, różniących się między sobą parametrami i możliwościami funkcjijnymi. Jednostki CPU2x wyposażone są w zestawy szybkich wejść i wyjść cyfrowych, dzięki czemu mogą być wykorzystane w aplikacjach sterujących napędami elektrycznymi. Z kolei jednostki centralne typu CPUxx-ETN posiadają zintegrowany moduł komunikacyjny sieci Ethernet, który umożliwia wymianę danych pomiędzy innymi sterownikami, panelami operatorskimi oraz komputerami z oprogramowaniem typu SCADA. W tabeli 2.1 przedstawiono wybrane parametry, którymi różnią się poszczególne jednostki centralne sterowników serii CJ1M.

Mozliwości funkcjonalne i zasoby sprzętowe sterowników serii CJ1M zależą od doboru modułów rozszerzeń, do których należą:

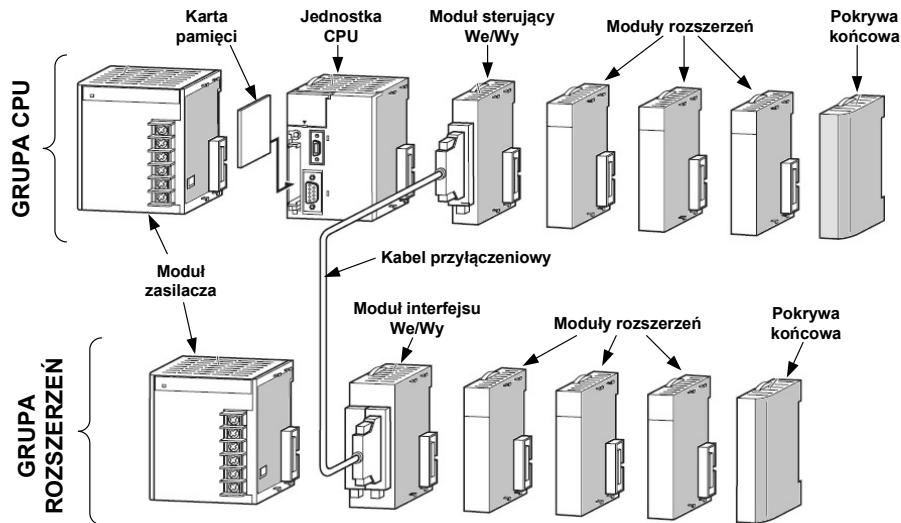
- moduł zasilacza – dostarcza odpowiednich napięć do zasilania jednostki centralnej oraz poszczególnych modułów rozszerzeń;
- podstawowe moduły wejść/wyjść – stanowią interfejs cyfrowy sterownika i zawierają: moduły wejść dyskretnych (16, 32, 64) oraz moduły wyjść dyskretnych (8, 16, 32, 64);
- specjalne moduły wejść/wyjść, do których zalicza się: moduły z wejściami/wyjściami analogowymi, regulatory temperatury, moduły z szybkimi licznikami oraz moduły pozycjonujące;

- moduły komunikacyjne, wśród których występują m.in.: moduły komunikacji szeregowej, moduły sieci Ethernet, moduły sieci DeviceNet, moduły sieci Profibus DP;
- moduły sterujące, które umożliwiają dołączenie dodatkowych grup rozszerzeń do sterownika.

Tab.2.1. Wybrane parametry jednostek centralnych sterowników rodziny CJ1M

Jednostka centralna	Maksymalna liczba cyfrowych wejść/wyjść	Pojemność programu [kroków]	Maksymalna liczba modułów rozszerzeń	Funkcje wbudowane
CPU11	160	5.000	10	
CPU11-ETN			9	Port 100 base-Tx Ethernet
CPU12	320	10.000	10	
CPU12-ETN			9	Port 100 base-Tx Ethernet
CPU13	640	20.000	20	
CPU13-ETN			19	Port 100 base-Tx Ethernet
CPU21	160	5.000	10	2 wejścia enkodera (100kHz) 2 wyjścia impulsowe (100kHz) 4 wejścia przerwaniowe / wejścia licznika
CPU22	320	10.000	10	
CPU23	640	20.000	20	

Na rysunku 2.2 przedstawiono przykładową konfigurację sprzętową sterownika z rodziny CJ1M.



Rys.2.2. Przykładowa konfiguracja sprzętowa sterownika CJ1M

Podstawowe jednostki centralne CPUx1 oraz CPUx2 umożliwiają dołączenie maksymalnie 10 modułów rozszerzeń. Minimalna konfiguracja sprzętowa sterownika wymaga obecności zasilacza, jednostki centralnej oraz pokrywy krańcowej, która niezbędna jest do prawidłowego działania magistrali rozszerzeń sterownika. Jednostki centralne typu CPUx3 umożliwiają dołączenie dodatkowej grupy modułów rozszerzeń, za pomocą specjalnego modułu sterującego We/Wy. Wówczas maksymalna liczba modułów obsługiwanych przez sterownik wzrasta do 20.

2.2.2. Jednostka centralna CJ1M CPU12-ETN

Jednostka centralna CJ1M CPU12-ETN jest odmianą bardzo popularnej, wysokowydajnej jednostki CPU sterowników rodziny CJ, zawierającej w jednej obudowie zintegrowany moduł komunikacyjny sieci Ethernet. Oprócz możliwości komunikacyjnych, rozwiązywanie to umożliwia zaprogramowanie sterownika za pomocą dowolnego komputera PC wyposażonego w standardową kartę sieciową. W tabeli 2.2 zestawiono podstawowe parametry jednostki sterownika CJ1M z jednostką centralną CPU12-ETN.

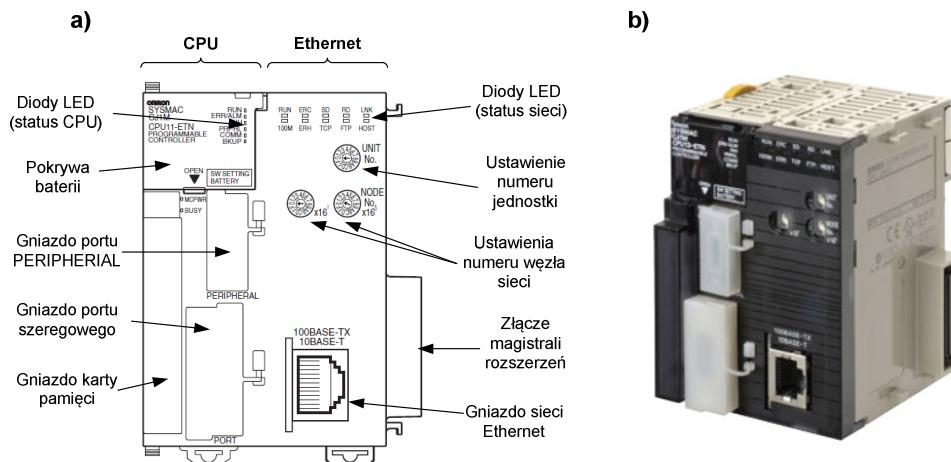
Tab.2.2. Wybrane parametry sterownika OMRON CJ1M CPU12-ETN

Obsługiwana liczba wejść/wyjść (<i>I/O points</i>)		320
Pamięć programu użytkownika		10 tys. kroków
Maksymalna liczba modułów rozszerzeń		9
Pamięć danych		32 tys. słów
Liczba wejść przerwaniowych		1
Maksymalna liczba podprogramów		256
Maksymalna liczba instrukcji skoku (JMP)		256
Bloki funkcyjne (FB)	maksymalna liczba definicji	128
	maksymalna liczba wystąpień	256
Pamięć Flash	Pamięć dla bloków funkcyjnych	256kB
	Obszar komentarzy	64kB
	Obszar <i>Program Index File</i>	64kB
	Symboli zmiennej	64kB
Czas wykonania instrukcji	Instrukcje podstawowe	100ns
	Instrukcje specjalne	300ns
Czas przetwarzania		0,5ms
Maksymalna liczba zadań (<i>Tasks</i>)	Zadania przerwaniowe	256
	Zadania cykliczne	32
Zakres ustawień minimalnego czasu trwania cyklu		1–32ms

Oprócz wbudowanego modułu sieci Ethernet, sterownik ten wyposażony jest w dwa złącza komunikacji szeregowej: zwykły port szeregowy zgodny ze standardem RS-232C oraz dodatkowy port „Peripheral”, który jest głównie wykorzystywany do

komunikacji sieciowej między sterownikami OMRON. Główny port szeregowy umożliwia podłączenie dowolnego urządzenia zewnętrznego obsługującego standard RS-232C, przy czym użytkownik może zdefiniować własny protokół wymiany danych. Ponadto oba porty obsługują komunikację szeregową przy wykorzystaniu protokołów *Host Link*, *NT Links* czy *Serial Gateway (CompoWay/F master)*. Przy podłączeniu zewnętrznego urządzenia należy pamiętać o zastosowaniu specjalnego kabla do transmisji szeregowej, ponieważ rozmieszczenie sygnałów w złączu DSUB9 jest niestandardowe.

Na rysunku 2.3 przedstawiono rozmieszczenie elementów funkcjonalnych na panelu sterownika oraz jego fotografię. W górnej części obudowy znajdują się diody LED, które sygnalizują status sterownika (napięcie zasilania, stan pracy CPU, błędy sprzętowe i programowe). Pod pokrywą znajduje się bateria podtrzymująca napięcie zasilania oraz zestaw przełączników typu *DIP-switch* służących do zaawansowanych ustawień sprzętowych sterownika (ochrona zapisu do pamięci, transfer programu do/z karty pamięci, wybór nastaw domyślnych portów szeregowych).



Rys.2.3. Jednostka centralna CJ1M CPU12-ETN: a) – rozmieszczenie elementów funkcjonalnych , b) – fotografia modułu.

Poniżej, po lewej stronie gniazd portów szeregowych znajduje się gniazdo karty pamięci typu Compact Flash. Karta ta jest opcjonalna i może być wykorzystana do przechowywania kopii zapasowej programu oraz do przenoszenia programu pomiędzy sterownikami. W przypadku wykonania kopii zapasowej programu (*ang. program backup*), na kartę flash zostanie skopiowany cały program użytkownika, bloki funkcyjne, tabele symboli (zmiennych) globalnych i lokalnych oraz wszystkie komentarze umieszczone w programie. Należy przy tym pamiętać, że sterownik obsługuje tylko karty CF o pojemnościach 15, 30 i 60MB.

Prawa część panelu sterownika dotyczy ustawień modułu sieci Ethernet. W górnej części obudowy umieszczone są diody LED informujące o statusie połączenia sieciowego. Poniżej znajdują się nastawniki numeru jednostki CPU (Unit Number) oraz nastawniki numeru węzła sieci (Node Number). Wszystkie moduły komunikacyjne pracujące w jednej sieci powinny posiadać swój własny, unikatowy numer węzła. W przypadku modułów ethernetowych zaleca się, żeby numer węzła był zgodny z ostatnim oktetem adresu IP (np. dla modułu o adresie IP: 192.168.115.5 należy wprowadzić numer węzła 5).

2.2.3. MAPA PAMIĘCI STEROWNIKA CJ1M

Pamięć danych sterownika CJ1M posiada strukturę 16-bitową, w związku z tym adres komórki pamięci określa numer słowa, a nie bajtu, jak to ma miejsce w innych sterownikach (np. rodziny SIMATIC S7). Pamięć ta została podzielona na kilka odrębnych obszarów, w których przechowane są dane pełniące określone funkcje. Dostęp do poszczególnych komórek pamięci uzyskuje się poprzez podanie prefiku obszaru pamięci (np. W, D, T...) oraz określenie adresu słowa i ewentualnie numeru bitu (np. W100.15). Wyjątkiem od tej reguły jest dostęp do obszaru wejść/wyjść sterownika (ang. *CIO – Common Input-Output area*), gdzie podaje się jedynie adres, bez prefiksu (np. 500.05).

W adresowaniu obszarów pamięci, które pozwalają na bezpośredni dostęp do bitów stosuje się dwie metody notacji:

- **z kropką** – w której podaje się adres słowa, a po kropce (separatorem) umieszcza się numer bitu (np. W15.01 lub W15.1 ; 100.15);
- **bez kropki** – gdzie nie podaje się separatora bitowego, a dwie ostatnie cyfry określają numer bitu (np. W10010 odpowiada zapisowi W100.10). W takim przypadku podanie adresu w postaci „10” będzie odnosiło się do 10-tego bitu komórki z obszaru CIO o adresie 0 (czyli 0.10).

Jeżeli adresowanie dotyczy całego słowa (16 bitów), wówczas adres składa się tylko z sekcji określającej numer komórki (np. 10, W30, D1000).

W tabeli 2.3 przedstawiono podział pamięci sterownika serii CJ1 na poszczególne obszary funkcyjne.

W obszarze **CIO** przechowywane są dane określające stany wejść i wyjść cyfrowych sterownika, dane modułów komunikacyjnych, modułów specjalnych oraz bity robocze ogólnego przeznaczenia. Wewnątrz pamięci *CIO* wydzielono kilka specyficznych obszarów, pełniących określone funkcje, dla których na stałe przypisano odpowiednie zakresy adresowe.

Do przechowywania danych użytkownika sterujących przebiegiem programu (w szczególności bitów roboczych – tzw. markerów) służy obszar **W** (ang. *Working area*) o pojemności 512 słów. Dostęp do tego obszaru może odbywać się zarówno w formie

adresowania bitów, jak i całych słów. Zawartość pamięci w tym obszarze nie jest podtrzymywana po zaniku napięcia zasilającego, ani po zresetowaniu sterownika. W przypadku, gdy wielkość pamięci typu W okaże się niewystarczająca, użytkownik może umieszczać dane (bity robocze) w wolnym obszarze CIO .

Obszar danych D (ang. *Data area*) posiada pojemność 32 kilosłów i może służyć do przechowywania ogólnych danych liczbowych użytkownika. W tym obszarze nie ma możliwości zaadresowania konkretnego bitu, możliwy jest tylko dostęp do całych komórek 16-bitowych. Zawartość całej pamięci obszaru D jest podtrzymywana po zaniku zasilania i restarcie sterownika. W obszarze D znajdują się wydzielone komórki, które służą do wymiany danych pomiędzy sterownikiem a specjalnymi modułami rozszerzeń (D20000-D29599) oraz komórki, które przechowują dane transferowane przez moduły komunikacyjne (D30000-D31599).

Obszar pamięci H (ang. *Holding area*) służy do przechowywania różnych danych w postaci słów albo bitów. Dane umieszczone w tym obszarze są podtrzymywane po zaniku zasilania sterownika. Oprócz danych ogólnego przeznaczenia, w obszarze H umieszczane są dane generowane przez program, w przypadku użycia bloków funkcyjnych oraz innych instrukcji wymagających zapamiętania stanów.

Obszar pomocniczy A (ang. *Auxiliary area*) zawiera flagi i bity sterujące, które mogą służyć do monitorowania i sterowania pracą sterownika PLC. Ogólnie obszar ten dzieli się na dwie części: komórki tylko do odczytu (A000-A447) oraz komórki do zapisu i odczytu (A448-A959). W obszarze pomocniczym przechowywane są między innymi: kody błędów sterownika, flagi systemowe, bieżący czas i data, czas działania sterownika, znaczniki czasowe ostatnich wyłączeń i zaników zasilania sterownika, dane i parametry związane z obsługą szybkich liczników i wiele innych pożytecznych informacji.

Obszar czasomierzy T (ang. *Timer area*) służy do przechowywania bieżących wartości PV (ang. *Preset Value*) oraz flag wypełnienia (ang. *Completion Flag*) poszczególnych czasomierzy, o numerach z zakresu 0–4095. W zależności od typu zastosowanej instrukcji czasomierza (TIM, TIMX), wartości PV mogą posiadać format binarny albo BCD. Z kolei flagi wypełnienia czasomierzy mają format bitowy i w programie występują w postaci styków normalnie otwartych (NO) lub zamkniętych (NC). Połączeniu zasilania sterownika wszystkie wartości PV i znaczniki wypełnienia zostają wyzerowane.

W Obszarze liczników C (ang. *Counter area*) przechowywane są wartości bieżące PV oraz znaczniki wypełnienia (flagi) poszczególnych liczników, o numerach z zakresu 0–4095. Podobnie jak w przypadku czasomierzy, format wartości bieżących liczników PV zależy od typu użytych instrukcji licznikowych i może przyjąć postać binarną albo BCD. Po wyłączeniu zasilania i powtórnym uruchomieniu sterownika, wartości PV i znaczniki wypełnienia dla wszystkich liczników zostają zachowane.

Tab.2.3. Obszary pamięci sterownika serii CJ1

Obszar	Zakres	Opis
Obszar Wejść – Wyjścia (CIO)	Obszar We/Wy	CIO (0000 – 0159) Obszar przydzielany automatycznie do podstawowych modułów We/Wy. Ustawienia domyślne dla pierwszej grupy modułów (0000) można zmieniać w zakresie (0000 – 0999)
	Obszar wbudowanych We/Wy	CIO (2960 – 2961) Dla wbudowanych wejść w sterownikach CJ1M-CPU22/23
	Obszar wspólny	CIO (1000 – 1199) Bity wspólne są używane jako dane wspólne przydzielane do modułów Controller Link
	Obszar modułów komunikacyjnych	CIO (1500 – 1899) W obszarze tym przechowywany jest stan pracy modułów komunikacyjnych (25 słów na moduł, maksymalnie 16 modułów)
	Obszar specjalnych modułów We/Wy	CIO (2000 – 2959) Obszar przydzielany do specjalnych modułów rozszerzeń, np. analogowych We/Wy (10 słów na moduł, maksymalnie 96 modułów)
	Obszar połączenia szeregowego sterowników	CIO (3100 – 3189) Do przekazywania danych przy połączeniach szeregowych sterowników PLC – tylko dla CJ1M.
	Obszar sieci DeviceNet	CIO (3200 – 3799) Obszar przydzielany dla urządzeń sieci DeviceNet realizujących komunikację ze zdalnymi stacjami We/Wy dla stałych alokacji w module typu master.
	Obszar We/Wy wewnętrznych (bitы robocze)	CIO (1200 – 1499) (3800 – 6143) Bity używane jako standardowe bity robocze. Nie można ich wykorzystywać do adresowania We/Wy zewnętrznych
	Obszar roboczy	W (000 – 511) Obszar przeznaczony na dane użytkownika. Komórki pamięci w tym obszarze mogą być adresowane bitowo (np. W100.15).
	Obszar podrzymywania	H (000 – 511) Dane umieszczone w tym obszarze zostają zachowane po wyłączeniu zasilania oraz po zmianie trybu działania sterownika.
	Obszar pomocniczy	A (000 – 959) Przechowywane są tu bity sterujące, flagi systemowe, znaczniki błędów oraz bity pomocnicze, pełniące funkcje specjalne.
	Obszar tymczasowy	TR (00 – 15) 16 bitów tymczasowych służy do przechowywania informacji o punktach rozgałęzienia programów.
	Obszar czasomierzy	T (0000 – 4095) Przechowuje wartości bieżące czasomierzy oraz ich znaczniki wypełnienia.
	Obszar liczników	C (0000 – 4095) Przechowuje wartości bieżące liczników oraz ich znaczniki wypełnienia.

Obszar danych	D (00000 – 32767)	Obszar przeznaczony dla danych użytkownika. Odczyt i zapis danych z tego obszaru możliwy jest tylko dla całych słów. Nie ma możliwości bezpośredniego dostępu do pojedynczych bitów.
Rejestry indeksowe	IR (00 – 15)	W obszarze tym przechowywane są adresy komórek pamięci, przy adresowaniu pośrednim – indeksowym. Każdy rejestr indeksowy ma pojemność 32-bitów (dwóch słów).
Znaczniki zadań	TK (00 – 31)	Znaczniki zadań są to bity, które przyjmują wartość logiczną 1, wtedy, gdy odpowiadające im zadanie cykliczne jest wykonywalne.

2.2.4. ALOKACJA MODUŁÓW ROZSZERZEŃ

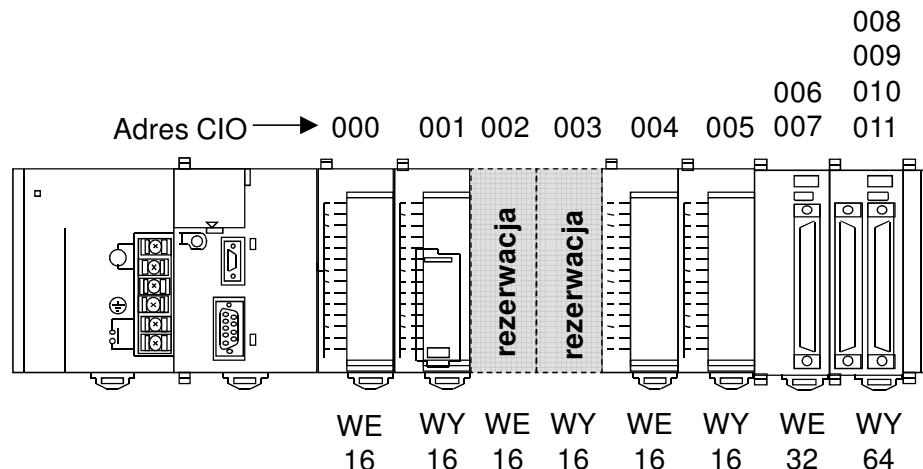
Sterowniki programowalne serii CJ umożliwiają wykonanie automatycznej alokacji adresów wszystkich dołączonych podstawowych modułów I/O, zaraz połączeniu zasilania sterownika. Proces ten polega na zarezerwowaniu dla każdego modułu odpowiedniej przestrzeni pamięci oraz przypisanie mu unikalnego adresu w obszarze CIO.

Dla modułów podstawowych (*Basic I/O Units*) adresy przypisywane są po kolei, rozpoczynając od modułu znajdującego się najbliżej jednostki centralnej CPU. W konfiguracji domyślnej alokacja rozpoczyna się od adresu CIO 0000, jednak użytkownik za pomocą oprogramowania *CX-Programmer* może początkowy adres ustawić w zakresie CIO 0000–0999. Dla każdego modułu alokowane jest tyle słów pamięci, ile jest wymagane. Przykładowo, dla modułów obsługujących 16 wejść cyfrowych będzie przydzielone tylko jedno słwo pamięci (16 bitów), zaś moduły 64-wejściowe otrzymają 4 słowa (4x16 bitów).

Jeżeli przewidywane są w przyszłości zmiany w konfiguracji sprzętowej sterownika, np. planowane jest zainstalowanie dodatkowych modułów I/O, wówczas użytkownik może zawsze zarezerwować pewien obszar pamięci dla tych modułów, tak aby później nie trzeba było zmieniać adresów w programie sterującym. Na rysunku 2.4 przedstawiono przykładową alokację adresową dla zestawu podstawowych modułów I/O. W tej konfiguracji, zarezerwowano 2 słowa pamięci (002 i 003) przeznaczone dla przyszłych modułów, które będą umieszczone w 3 i 4 slocie.

Alokacja specjalnych modułów wejść/wyjść (ang. *SIOU – Specjal Input-Output Unit*) polega na przypisaniu dla każdego modułu 10 kolejnych słów pamięci w obszarze CIO 2000–2959. Adres początkowy zależy od numeru modułu (ang. *Unit Number*), który jest ustawiany za pomocą odpowiednich nastawników znajdujących się na obudowie modułu. Każdy moduł SIOU musi posiadać swój niepowtarzalny

numer w zakresie 0–95. W tabeli 2.4 przedstawiono zakresy adresowe modułów specjalnych, w zależności od ich numerów.



Rys.2.4. Przykładowa alokacja pamięci modułów podstawowych I/O

Tab.2.4. Alokacja pamięci specjalnych modułów wejść/wyjść

Numer modułu	Alokacja pamięci
0	CIO 2000 ÷ 2009
1	CIO 2010 ÷ 2019
2	CIO 2020 ÷ 2029
50	CIO 2050 ÷ 2059
N	CIO $(2000+N*10) \div (2009+N*10)$
95	CIO 2950 ÷ 2959

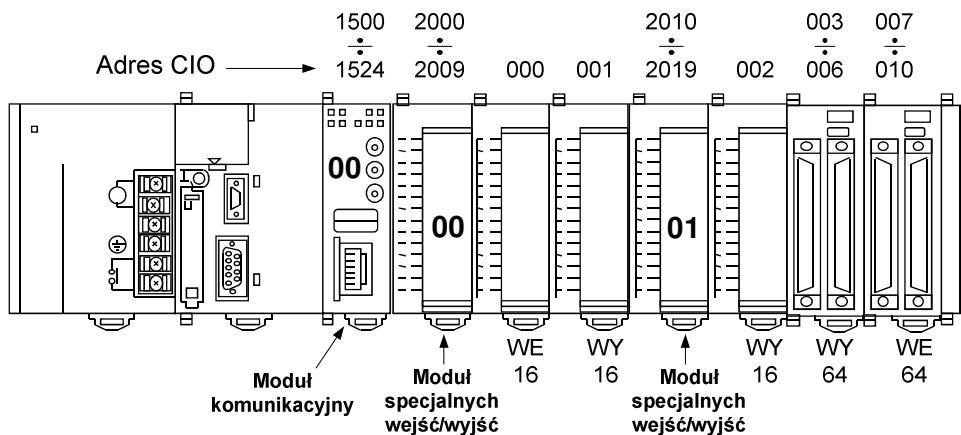
Dla każdego modułu komunikacyjnego zostaje alokowanych 25 słów pamięci w obszarze CIO 1500–1899. Podobnie jak w przypadku modułów SIOU, adres początkowy modułu komunikacyjnego zależy od jego numeru, ustawionego za pomocą nastawników obrotowych znajdujących się na jego obudowie. Każdy moduł komunikacyjny musi posiadać unikalny numer w zakresie 0–15. W tabeli 2.5 przedstawiono zasady alokacji pamięci dla modułów komunikacyjnych.

Tab.2.5. Alokacja pamięci modułów komunikacyjnych

Numer modułu	Alokacja pamięci
0	CIO 1500 ÷ 1524
1	CIO 1525 ÷ 1549
2	CIO 1550 ÷ 1574
N	CIO $(1500+N*25) \div (1524+N*25)$
15	CIO 1875 ÷ 1899

Zasada unikalności modułów dotyczy tylko urządzeń jednego typu. To znaczy, że w jednej konfiguracji sprzętowej może znajdować się moduł z grupy SIOU o tym samym numerze, co moduł komunikacyjny czy też moduł z grupy podstawowych wejść/wyjść.

Na rysunku 2.5 przedstawiono przykładową alokację pamięci modułów w konfiguracji mieszanej, w której jednocześnie występują moduły podstawowe, specjalne i komunikacyjne.



Rys.2.5. Przykładowa alokacja pamięci modułów mieszanych

2.3. Wybrane moduły rozszerzeń sterowników serii CJ

2.3.1. MODUŁ WEJŚĆ CYFROWYCH ID211

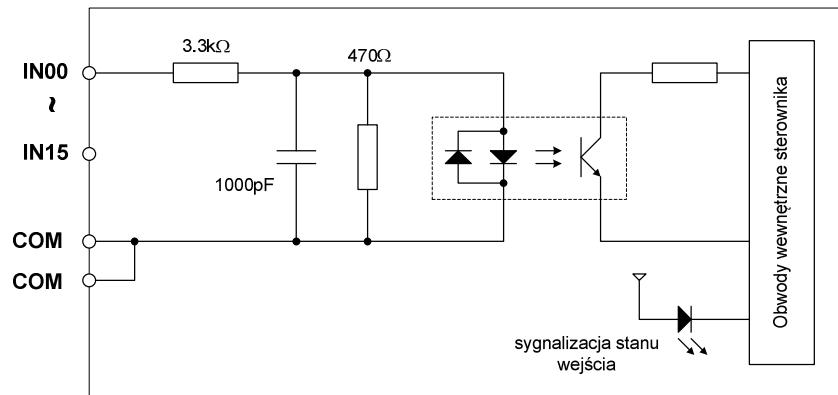
Moduł CJ1W-ID211 jest 16-kanałowym modułem wejść dyskretnych o napięciu znamionowym 24V_{DC}. W przestrzeni adresowej pamięci CIO zajmuje jedno słowo, w którym kolejne bity odpowiadają wartościom logicznym występującym na poszczególnych wejściach. Przy kodowaniu sygnałów wejściowych przyjęto logikę dodatnią, co oznacza, że obecność napięcia +24V na wejściu fizycznym modułu będzie powodować ustawienie odpowiadającego mu bitu na wartość logiczną „1”. W tabeli 2.7 zestawiono podstawowe parametry modułu ID211.

Tab.2.7. Podstawowe parametry modułu ID211

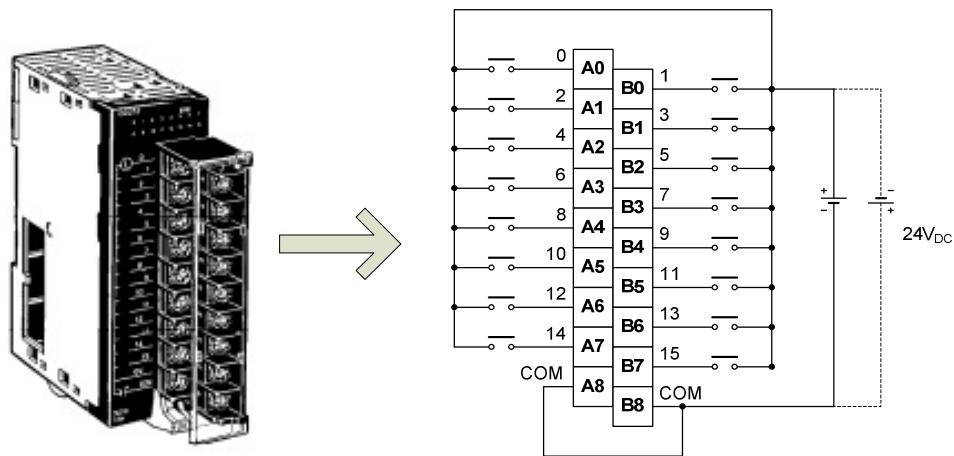
Napięcie znamionowe	24 V DC
Zakres napięcia wejściowego	20,4–26,4 V DC
Impedancja wejściowa	3,3 kΩ
Prąd wejścia	7mA (przy 24V)
Napięcie załączenia (ON)	14,4 V DC min./3mA min.
Napięcie wyłączenia (OFF)	5 V DC max./1mA max.
Czas załączenia (ON)	8ms max. (możliwość ustawienia od 0 do 32ms w PLC Setup)
Czas wyłączenia (OFF)	8ms max. (możliwość ustawienia od 0 do 32ms w PLC Setup)
Rezystancja izolacji	20 MΩ (pomiędzy zaciskami zewnętrznymi a zaciskiem GR)
izolacja napięciowa	1000 V AC
Pobór prądu modułu	5 V DC: 80mA max.

Na rysunku 2.6 przedstawiono schemat ideowy obwodu wejściowego dla pojedynczego wejścia modułu ID211. Wejściowe elementy pasywne R i C tworzą dzielnicę napięciową oraz prosty układ filtra dolnoprzepustowego, którego celem jest tłumienie zakłóceń wejściowych o charakterze impulsowym. Sygnał napięcia wejściowego zostaje doprowadzony do diody LED wewnętrznego transortora, którego zadaniem jest przekazanie sygnału sterującego doewnętrznych obwodów sterownika PLC przy jednoczesnym zachowaniu izolacji galwanicznej. Dla takiej konfiguracji obwodów wejściowych, wejściowy prąd sterujący osiąga wartość ok. 7mA przy napięciu 24V.

W module ID211 zastosowano transortory bipolarne z wejściem AC, które mogą pracować przy dowolnej polaryzacji napięcia sterującego, w tym również przy napięciach przemiennych. Dzięki temu użytkownik może dowolnie skonfigurować połączenia zewnętrznych sygnałów sterujących, tak aby podawać na wejścia modułu poziomy wysokie lub niskie napięcia (układ z logiką dodatnią lub ujemną). Sposób dołączenia sygnałów wejściowych do zacisków modułu przedstawiono na rysunku 2.7.



Rys.2.6. Schemat ideowy obwodów wejściowych modułu ID211



Rys.2.7. Sposób dołączenia sygnałów sterujących do modułu ID211

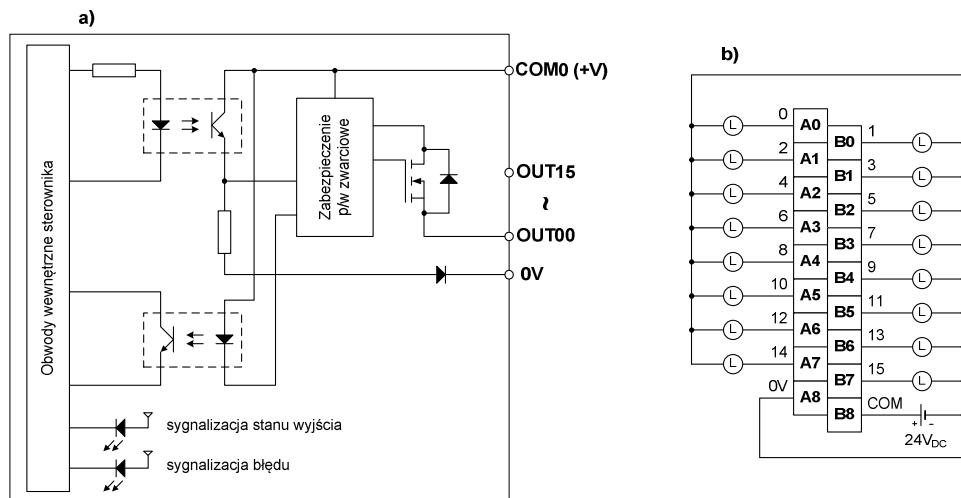
2.3.2. MODUŁ WYJŚĆ CYFROWYCH OD212

Moduł CJ1W-OD212 jest 16-kanałowym modułem półprzewodnikowych wyjść dyskretnych o napięciu znamionowym 24V_{DC}. Podobnie jak w module wejściowym, stan poszczególnych wyjść cyfrowych jest reprezentowany w przestrzeni adresowej pamięci CIO przez jedno słowo (16 bitów). W module zastosowano wyjścia tranzystorowe o logice dodatniej, co oznacza, że w stanie aktywnym (ON) na zaciskach wyjściowych pojawia się napięcie +24V DC. W tabeli 2.8 przedstawiono podstawowe parametry modułu OD212.

Tab.2.8. Podstawowe parametry modułu OD212

Napięcie znamionowe	24 V DC
Zakres zmian napięcia wyjściowego	20,4–26,4 V DC
Maksymalny prąd obciążenia	0,5 A / kanał, 5 A / moduł
Zabezpieczenie przeciążeniowe	Sygnalizacja przeciążenia w zakresie prądów 0,7–2,5A
Czas załączenia (ON)	0,5 ms max.
Czas wyłączenia (OFF)	1,0 ms max.
Rezystancja izolacji	20 MΩ (pomiędzy zaciskami zewnętrznymi a zaciskiem GR)
izolacja napięciowa	1000 V AC
Zewnętrzne napięcie zasilające	20,4–26,4 V DC, 40 mA min.
Pobór prądu modułu	5 V DC: 100 mA max.

Na rysunku 2.8a przedstawiono schemat ideowy wewnętrznego obwodu sterowania wyjścia cyfrowego.



Rys.2.8. Schemat obwodów wyjściowych modułu OD212 (a) oraz schemat podłączenia odbiorników zewnętrznych do zacisków wyjściowych (b).

W roli wyjściowych kluczy półprzewodnikowych zostały zastosowane tranzystory MOSFET wyposażone w dodatkowy układ zabezpieczenia nadprądowego, który wykrywa nadmierny wzrost prądu obciążenia i sygnalizuje go za pomocą czerwonej diody LED umieszczonej na panelu modułu. Dodatkowo, stan przeciążenia powoduje

uaktywnienie flagi błędu w obszarze pamięci pomocniczej, w odpowiednich komórkach z zakresu A050–A069. Rysunek 2.8b przedstawia sposób podłączenia odbiorników i zewnętrznego źródła zasilania do zacisków modułu OD212. Należy zwrócić uwagę na właściwą polaryzację źródła zasilania, gdyż odwrotne podłączenie może spowodować załączenie odbiorników wskutek przepływu prądu przez diody zwrotne wbudowane w wyjściowe tranzystory MOSFET.

2.3.3. MODUŁ WEJŚĆ ANALOGOWYCH AD041

Moduł specjalny AD041 posiada 4 wejścia analogowe, które mogą być skonfigurowane jako wejścia napięciowe lub jako wejścia prądowe. Oprócz standardowych funkcji pomiarowych, moduł analogowy oferuje dodatkowe funkcje:

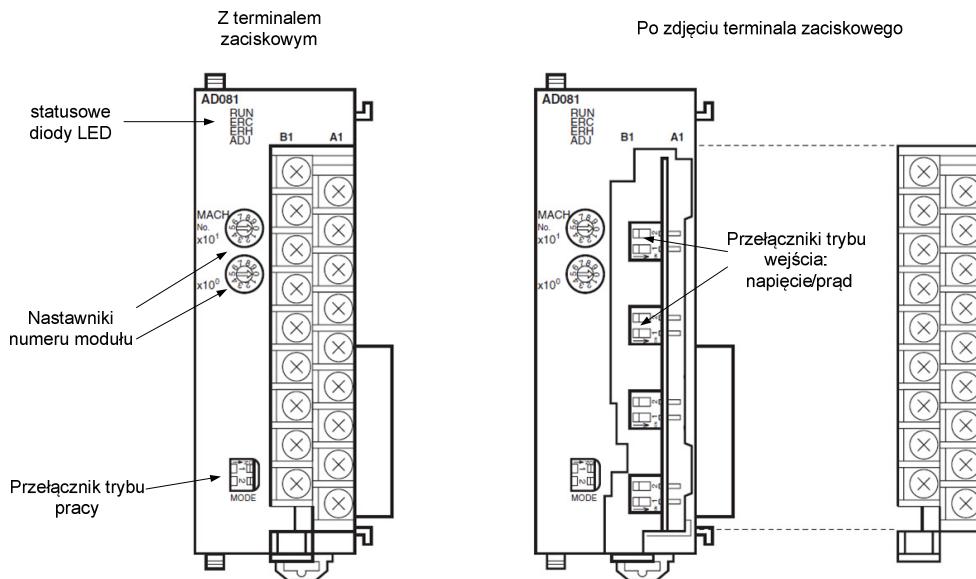
- możliwość wykrycia stanu rozłączenia obwodu pomiarowego (tylko przy konfiguracji zakresów wejściowych: 1–5V lub 4–20mA),
- obliczanie wartości średniej sygnału wejściowego, na podstawie konfigurowalnej liczby buforowanych próbek (2, 4, 8, 16, 32 i 64),
- detekcja wartości szczytowych mierzonych sygnałów,
- możliwość kalibracji (regulacja wzmacnienia i składowej stałej dla wzmacniaczy wejściowych).

W tabeli 2.9 przedstawiono podstawowe parametry modułu AD041.

Tab.2.9. Wybrane parametry modułu wejść analogowych AD041

Liczba wejść analogowych	4	
Zakresy sygnałów wejściowych	1–5V 0–5V 0–10V -10–10V 4–20mA	
Maksymalne wartości napięć i prądów wejściowych		
Impedancja wejściowa		Wejście napięciowe: $1M\Omega$ min. Wejście prądowe: 250Ω
Rozdzielcość		4.000/8.000
Format danych wyjściowych	16-bitowy binarny	
Dokładność przetwarzania	23±2°C	Wejście napięciowe: ±0,2% zakresu Wejście prądowe: ±0,4% zakresu
	0–50°C	Wejście napięciowe: ±0,4% zakresu Wejście prądowe: ±0,6% zakresu
Czas konwersji	1ms/250μs	

Na rysunku 2.9 przedstawiono rozmieszczenie elementów funkcyjnych na panelu modułu AD041. Przełączniki trybu pracy wejścia (napięcie/prąd) znajdują się pod terminalem zaciskowym. W górnej części panelu znajdują się diody LED sygnalizujące stan pracy modułu. W tabeli 2.10 przedstawiono znaczenie poszczególnych sygnałów diagnostycznych diod LED.

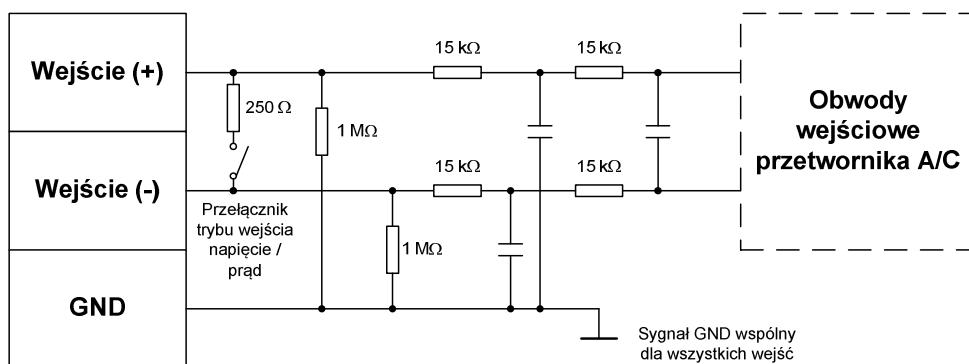


Rys.2.9. Rozmieszczenie elementów funkcyjnych na przednim panelu modułu AD041/AD081

Tab.2.10. Znaczenie sygnałów diagnostycznych wyświetlanych przez diody LED

LED	funkcja	Stan	Opis
RUN (zielona)	Tryb pracy	świeci	Działanie normalne
		nie świeci	Moduł został zatrzymany podczas wymiany danych z CPU
ERC (czerwona)	Błąd wykryty przez moduł	świeci	Pojawił się stan alarmowy lub wprowadzono nieprawidłowe ustawienia początkowe
		nie świeci	Działanie normalne
ERH (czerwona)	Błąd w module CPU	świeci	Wystąpił błąd podczas wymiany danych z CPU
		nie świeci	Działanie normalne
ADJ (żółta)	Tryb ustaleń	świeci	Działa w trybie kalibracji wzmacnienia lub offsetu
		nie świeci	żaden z powyższych

Rysunek 2.10 przedstawia schemat ideowy obwodu wejściowego dla jednego kanału pomiarowego. Na wejściu znajduje się przełącznik rodzaju pracy, który przy wyborze trybu pomiaru prądu dołącza obwód z rezystorem wejściowym o wartości 250Ω . W dalszej części obwodu znajdują się filtry RC, które redukują zakłócenia wysokoczęstotliwościowe. Poszczególne wejścia analogowe znajdują się na tym samym potencjale – posiadają wspólny punkt masy sygnałowej GND. Wewnętrzne obwody wejściowe modułu są odseparowane galwanicznie od magistrali sterownika PLC za pomocą transceptorów.



Rys.2.10. Schemat ideowy obwodu wejściowego dla jednego kanału modułu AD041

Wymiana danych pomiędzy modułem analogowym a sterownikiem odbywa się poprzez komórki pamięci w obszarze CIO (10 słów na każdy moduł) w zakresie adresowym 2000–2959. Oprócz tego, ustawienia i parametry modułu przechowywane są w obszarze pamięci D zakresie adresowym 20000–29599 (100 słów dla każdego modułu).

2.3.4. MODUŁ WYJŚĆ ANALOGOWYCH DA041

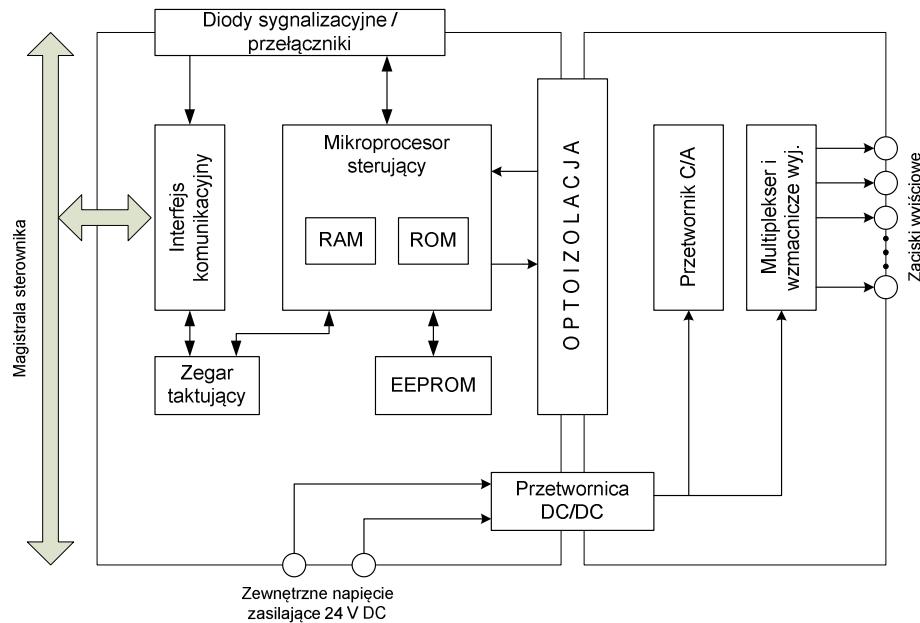
Moduł wyjść analogowych CJ1W-DA041 służy do wyrowadzenia ze sterownika sygnałów analogowych, w postaci napięcia lub prądu. Dla każdego z czterech wyjść analogowych można wprowadzić niezależne ustawienia parametrów sygnałów wyjściowych, wybierając tryb pracy (napięcie/prąd) oraz dobierając zakres przetwarzania w zależności od wymagań aplikacji użytkownika. W tabeli 2.11 przedstawiono podstawowe parametry modułu DA041.

Rozmieszczenie elementów funkcjnych na panelu modułu DA041 oraz znaczenie sygnałów diagnostycznych diod LED jest identyczne jak dla modułu wejść analogowych AD041.

Tab.2.11. Wybrane parametry modułu wejść analogowych DA041

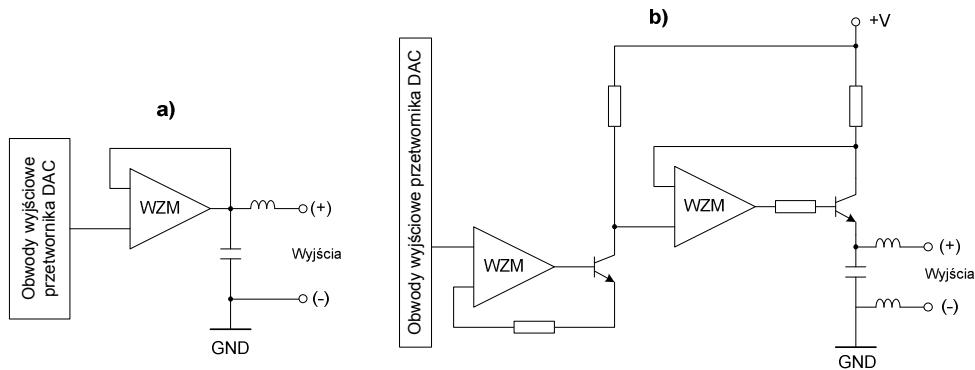
Liczba wyjść analogowych	4
Zakresy sygnałów wyjściowych	1–5V 0–5V 0–10V -10–10V 4–20mA
Impedancja wyjściowa	0,5 Ω max. (dla wyjścia napięciowego)
Maksymalny prąd wyjścia	12 mA (dla wyjścia napięciowego)
Maksymalna rezystancja obciążenia	600 Ω max. (dla wyjścia prądowego)
Rozdzielcość	4.000
Format danych wejściowych	16-bitowy binarny
Dokładność przetwarzania	23±2°C Wyjście napięciowe: ±0,3% zakresu Wejście prądowe: ±0,5% zakresu 0–50°C Wejście napięciowe: ±0,5% zakresu Wejście prądowe: ±0,8% zakresu
Czas konwersji	1ms max. / wyjście

Na rysunku 2.11 przedstawiono schemat blokowy wewnętrznych obwodów modułu DA041.



Rys.2.11. Schemat blokowy modułu wejść analogowych DA041

Głównym elementem jest mikroprocesor sterujący, który zarządza konfiguracją sprzętową i parametryczną modułu, wymienia dane z jednostką centralną sterownika oraz nadzoruje poprawność jego pracy. Wyjściowa sekcja analogowa modułu jest odseparowana galwanicznie od obwodów wewnętrznych sterownika PLC, dzięki czemu możliwe jest dołączenie do obwodów zewnętrznych, znajdujących się na innym potencjale niż sterownik. Obwody wyjściowe modułu DA041 zawierają multiplekser analogowy oraz cztery niezależne sekcje wzmacniaczy wyjściowych, po jednej dla każdego kanału wyjściowego. Wzmacniacze wyjściowe pracują w jednej z dwóch konfiguracji, przełączanych za pomocą przełączników trybu napięcie/prąd. Na rysunku 2.12 przedstawiono schematy ideowe obwodów wyjściowych wzmacniaczy, dla trybu napięciowego (a) i prądowego (b).



Rys.2.12. Schematy ideowe obwodów wyjściowych modułu DA041
w konfiguracji wyjścia napięciowego (a) oraz prądowego (b)

3. Oprogramowanie narzędziowe dla sterowników OMRON

3.1. Pakiet programowy CX-One

Wszystkie obecnie produkowane sterowniki programowalne firmy OMRON oraz inne produkty z grupy automatyki przemysłowej mogą być programowane, konfigurowane i uruchamiane za pomocą jednego, wspólnego pakietu programowego o nazwie CX-ONE. Rozwiążanie to znacznie upraszcza procedury projektowe, gdyż użytkownik korzysta tylko z jednego oprogramowania, z jedną instalacją i pojedynczym numerem licencji.

Pakiet CX-One zapewnia obsługę następujących urządzeń:

- wszystkie sterowniki PLC serii CP, CJ i CS;
- wszystkie terminaly operatorskie serii NS;
- sterowniki Trajexia;
- serwonapędy i falowniki OMRON Yaskawa;
- systemy i urządzenia komunikacyjne sieci Profibus, Profinet i CompoNet;
- regulatory temperatury;
- czujniki.

Programy narzędziowe pakietu CX-One zostały pogrupowane w zależności od pełnionej funkcji:

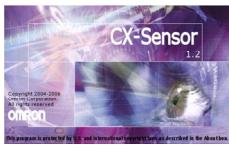
- narzędzia do programowania;
- narzędzia obsługi i konfiguracji sieci;
- programy do obsługi napędów;
- narzędzia do regulacji i przełączania;
- programy obsługi czujników.

W tabeli 3.1 przedstawiono podział programów narzędziowych należących do pakietu CX-One.

Tab.3.1. Programy narzędziowe pakietu CX-One

Program	Opis
Programowanie	
CX Programmer 	CX-Programmer zapewnia wspólną platformę oprogramowania sterowników PLC dla wszystkich urządzeń tego typu produkowanych przez firmę Omron. Umożliwia łatwą konwersję i ponowne wykorzystanie kodu w sterownikach PLC różnych typów, a także korzystanie (w pełnym zakresie) z programów kontrolnych utworzonych za pomocą aplikacji do programowania sterowników PLC starszych generacji.
CX Simulator 	CX-Simulator to zintegrowane środowisko debugowania odpowiadające rzeczywistemu sterownikowi PLC, które umożliwia ocenę działania programu, sprawdzenie długości cyklu i skrócenie czasu debugowania przed oddaniem urządzenia do użytku.
CX Designer 	Program CX-Designer służy do tworzenia danych ekranowych dla potrzeb programowalnych terminali z serii NS. Za jego pomocą można również sprawdzać na komputerze działanie utworzonych ekranów. CX-Designer zawiera około tysiąca standardowych obiektów funkcyjnych wraz z grafikami i zaawansowanymi funkcjami.
Obsługa sieci	
CX-Integrator 	CX-Integrator to główny składnik konfiguracyjny pakietu CX-One. Umożliwia on łatwe wykonywanie wielu operacji, takich jak monitorowanie stanu połączeń różnych sieci, ustawianie parametrów i diagnozowanie sieci.
CX-ConfiguratorFDT 	Oparty na technologii FDT/DTM program CX-ConfiguratorFDT służy do konfiguracji urządzeń dowolnego producenta podłączonych do sieci PROFIBUS. Koncepcja ta zostanie w przyszłości rozwinięta w celu obsługi innych sieci z użyciem technologii FDT/DTM.
CX-Protocol 	Służy do tworzenia procedur przesyłania danych (makro protokołów) w celu wymiany danych między standardowymi urządzeniami szeregowymi a sterownikami PLC za pośrednictwem modułu lub karty do obsługi transmisji szeregowej.
Network Configurator 	Network Configurator for EtherNet/IP umożliwia konfigurację jednostek EtherNet/IP oraz łączy danych „sterownik-sterownik”.

Sterowniki i Napędy	
	Program CX-Motion służy do tworzenia, edycji i drukowania rozmaitych parametrów, danych pozycji oraz programów sterowania ruchem (w języku G code) niezbędnych do działania sterowników ruchu, przesyłania danych do modułów sterowników ruchu oraz działania monitorów tychże modułów.
	Dzięki CX-Motion Pro programowanie sterowania ruchem jest proste, przejrzyste i szybkie.
	Program CX-Drive umożliwia pełen dostęp do parametrów wszystkich produkowanych obecnie falowników i serwonapędów Omron Yaskawa na jednym z trzech wybranych przez operatora poziomów. Program pozwala również w prosty sposób przeglądać parametry, filtrować wartości różne od domyślnych, różne od wartości falownika oraz nieprawidłowe ustawienia. Prezentacje graficzne służą pomocą przy konfiguracji bardziej szczegółowych parametrów, takich jak częstotliwość przeskoków, profile U/f i parametry nastawników analogowych.
	Program CX-Position upraszcza wszystkie aspekty kontroli pozycjonowania, od tworzenia i edycji danych wykorzystywanych przez moduły pozycjonujące (moduły NC) po komunikację online i monitoring. Program zawiera funkcje poprawiające wydajność, takie jak automatyczne tworzenie danych projektu i korzystanie z istniejących danych.
	Ten zintegrowany program konfiguracyjny umożliwia szybkie tworzenie rozwiązań w zakresie ruchu z punktu do punktu dla systemów opartych na sieci MECHATROLINK-II.
	Umożliwia programowanie zaawansowanych aplikacji sterowania ruchem za pomocą modułów sterowników ruchu CS1W-MCH71 i CJ1W-MCH71, ustawianie parametrów systemowych, danych pozycji, programów zadań ruchu i danych procesu CAM.
Regulacja i przełączanie	
	CX-Thermo to program do konfiguracji i monitorowania regulatorów temperatury z serii E5CN i E5ZN. Umożliwia łatwą konfigurację, rejestrowanie danych online i monitorowanie w czasie rzeczywistym. Użytkownicy mogą w prosty sposób tworzyć i edytować parametry oraz pobierać je w plikach z komputera PC, co zmniejsza nakład pracy potrzebnej do ich ustawienia. Można monitorować dane nawet z 31 regulatorów temperatury jednocześnie.

CX-Process 	Program CX-Process upraszcza sterowanie pętli, od tworzenia i przenoszenia bloków funkcyjnych po uruchamianie funkcji kart i modułów oraz debugowanie (tuning parametrów PID itp.). Istnieje możliwość łatwego tworzenia programów z bloków funkcyjnych przez wklejanie bloków do okna i tworzenie połączeń programowych za pomocą myszy.
Czujniki	
CX-Sensor 	CX-Sensor umożliwia konfigurację i monitorowanie czujników firmy Omron z serii ZX za pomocą łatwych w użyciu wskaźników. Graficzne okno dialogowe pozwala na wyświetlanie i porównywanie danych z kilkunastu czujników jednocześnie, dzięki czemu można konfigurować złożone procesy. Oprogramowanie obejmuje również sterownik umożliwiający dostęp do danych czujników za pośrednictwem modułów szeregowych SCU firmy Omron i innych aplikacji firmy, takich jak CX-Supervisor. W połączeniu z programem CX-Server OPC można nawet monitorować dane czujników w czasie rzeczywistym, korzystając w tym celu z programu Microsoft Excel.

3.2. CX-Programmer

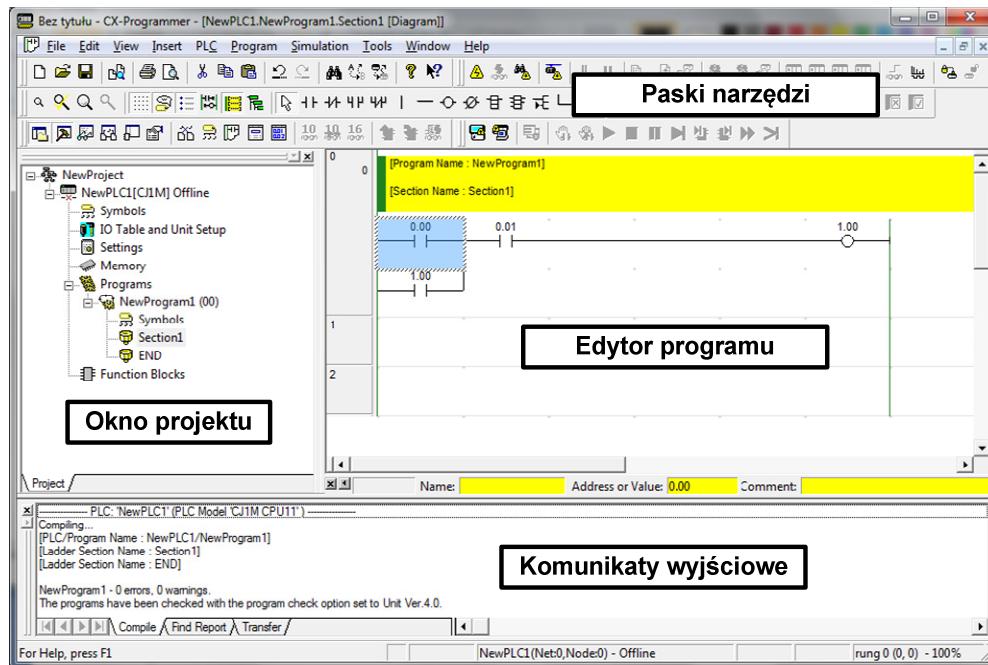
3.2.1. GŁÓWNE FUNKCJE PROGRAMU

CX-Programmer jest podstawowym oprogramowaniem narzędziowym do tworzenia, testowania i uruchamiania programów dla sterowników PLC OMRON z rodziny CP/CJ/CS. Edycja programu użytkownika możliwa jest w języku drabinkowym (LD), w języku ST oraz za pomocą grafu sekwencji SFC. CX-Programmer udostępnia bogatą bibliotekę skompilowanych bloków funkcyjnych, w których użytkownik znajdzie m.in.: gotowe funkcje do obsługi specjalnych modułów rozszerzeń, procedury komunikacyjne czy też zaawansowane funkcje do obsługi urządzeń peryferyjnych. Oprócz tego, oprogramowanie narzędziowe pozwala na tworzenie własnych bloków funkcyjnych, które mogą być umieszczone w bibliotece użytkownika, w celu ich wielokrotnego wykorzystania.

Na rysunku 3.1 przedstawiono wygląd głównego okna środowiska CX-Programmer. Głównie okno programu zawiera kilka charakterystycznych obszarów funkcyjnych, z których najważniejsze to:

- **okno projektu** – przedstawia drzewo plików projektu i umożliwia wyświetlenie i wprowadzenie ustawień sprzętowych i parametrów sterownika, podgląd i edycję tabeli symboli globalnych i lokalnych oraz otwieranie i przełączanie się pomiędzy programami (sekcjami) aplikacji użytkownika;

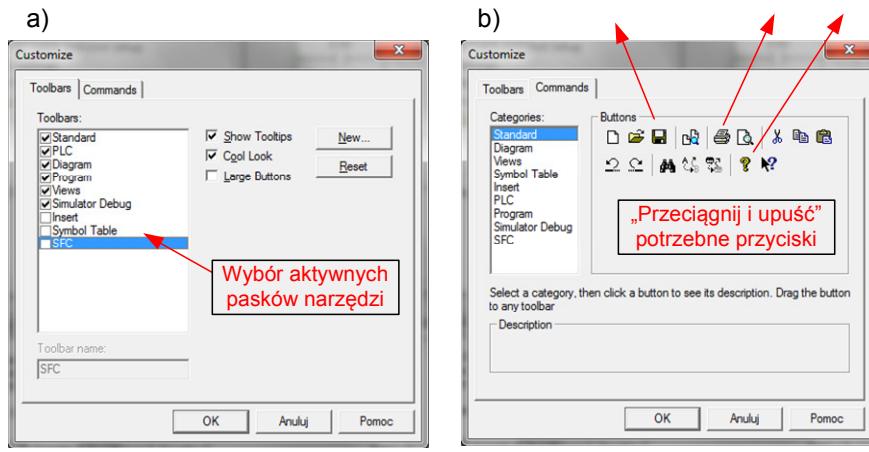
- **okno edycji programu** – przedstawia w postaci tekstowej lub graficznej program użytkownika i umożliwia jego edycję oraz analizę podczas fazy debugowania;
- **okno komunikatów wyjściowych** – wyświetla komunikaty i błędy pojawiające się podczas komplikacji ładowania programu do sterownika;
- **paski narzędzi** – zawierają przyciski, za pomocą których programista uzyskuje szybki dostęp do większości funkcji programu CX-Programmer. Zawartość palety przycisków może być dowolnie konfigurowana przez użytkownika.



Rys.3.1. Główne okno programu CX-Programmer

Po zainstalowaniu pakietu CX-One, środowisko projektowe CX-Programmer uruchamia się z domyślnymi ustawieniami. Efektem tego jest pojawienie się okna jak na rysunku 3.1, w którym większość z dostępnych pasków narzędzi jest od razu aktywna, a oczom użytkownika ukazuje się paleta z zawartością blisko stu przycisków funkcyjnych. Przyciski te na poszczególnych paskach narzędzi zostały pogrupowane tematycznie, co pozwala dość łatwo odnaleźć żądaną funkcję spośród „gąszcza” dostępnych opcji. Jednak konfiguracja taka nie każdemu musi odpowiadać. Po pierwsze nadmiar eksponowanych przycisków pogarsza czytelność menu oraz zajmuje sporo miejsca na ekranie, zwłaszcza na małych monitorach o ograniczonej rozdzielczości. Po drugie, większość z dostępnych funkcji jest uruchamiana

sporadycznie, a jeśli już zajdzie potrzeba ich uruchomienia, to użytkownik i tak bez problemu znajdzie ją w menu głównym programu. Z tego względu, przed przystąpieniem do pracy z programem CX-Programmer zaleca się wykonanie „personalizacji” pasków narzędzi, tak aby usunąć zbędne przyciski i dostosować wygląd programu do własnych preferencji. W tym celu należy uruchomić polecenie z menu View/Toolbars, czego efektem będzie pojawienie się okna „Customize” – jak na rysunku 3.2a. W oknie tym wyszczególnione są wszystkie aktywne i nieaktywne standardowe paski narzędzi, które użytkownik może dowolnie włączać i wyłączać. Po wybraniu zakładki „Commands” (Rys.3.2b) możliwa jest dalsza personalizacja palety narzędzi, polegająca na dodaniu nowych lub usunięciu niepotrzebnych przycisków metodą „przeciągnij i upuść”.



Rys.3.2. Personalizacja palety przycisków w programie CX-Programmer

Poniżej przedstawiono znaczenie poszczególnych przycisków na wybranych paskach narzędzi.

Pasek narzędzi „Standard”



- 1 – Tworzy nowy projekt
- 2 – Otwiera zapisany projekt

- 3 – Zapisuje projekt
- 4 – Porównuje aktualny program z innym programem, zapisanym na dysku
- 5 – Drukuję bieżący projekt
- 6 – Otwiera podgląd wydruku
- 7 – Wytnij
- 8 – Kopiuj
- 9 – Wklej
- 10 – Cofnij ostatnią operację
- 11 – Ponów cofniętą operację
- 12 – Znajdź
- 13 – Zamień
- 14 – Zamień wszystko
- 15 – Wyświetla okno „o programie” (About...)
- 16 – Pomoc kontekstowa

Pasek narzędzi „Widok”



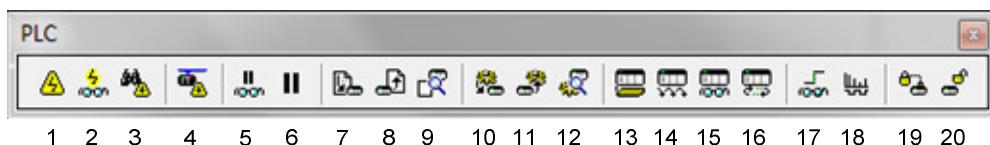
- 1 – Otwórz/zamknij okno projektu
- 2 – Otwórz/zamknij okno komunikatów wyjściowych
- 3 – Otwórz/zamknij okno śledzenia zmiennych (*Watch window*)
- 4 – Otwórz/zamknij okno adresów źródłowych
- 5 – Otwórz/zamknij okno podglądu bloku funkcyjnego
- 6 – Właściwości obiektu
- 7 – Okno użycia zmiennych „*Cross Reference*”
- 8 – Otwórz tabelę zmiennych lokalnych
- 9 – Widok schematu drabinkowego
- 10 – Reprezentacja programu w postaci listy instrukcji
- 11 – Okno komentarzy użytych zmiennych
- 12 – Wyświetla monitorowane wartości w postaci dziesiętnej (*Decimal*)
- 13 – Wyświetla monitorowane wartości w postaci dziesiętnej ze znakiem (*Signed Decimal*)
- 14 – Wyświetla monitorowane wartości w postaci szesnastkowej
- 15 – Pokaż wyższy poziom (program wywołujący blok funkcyjny)
- 16 – Pokaż niższy poziom (zawartość bloku funkcyjnego)
- 17 – Włącz monitorowanie bloku funkcyjnego

Pasek narzędzi „Diagram”



- 1 – Pomniejsz
- 2 – Dopasuj powiększenie do okna
- 3 – Powiększ
- 4 – Resetuj powiększenie
- 5 – Włącz siatkę
- 6 – Włącz/wyłącz opis zmiennych
- 7 – Włącz/wyłącz komentarze nagłówkowe dla instrukcji
- 8 – Zawijaj drabinki schematu (jeśli są za długie)
- 9 – Włącz/wyłącz komentarze sekcji
- 10 – Pokaż strukturę zagnieżdżeń
- 11 – Kursor normalny (strzałka)
- 12 – Wstawia styk NO
- 13 – Wstawia styk NC
- 14 – Połączenie równoległe ze stykiem NO
- 15 – Połączenie równoległe ze stykiem NC
- 16 – Połączenie pionowe
- 17 – Połączenie poziome
- 18 – Cewka (wyjście) normalna
- 19 – Cewka negująca
- 20 – Instrukcja języka drabinkowego (funkcja)
- 21 – Wywołanie bloku funkcyjnego
- 22 – Edycja parametru (pola) bloku funkcyjnego
- 23 – Włącz/wyłącz tryb rysowania połączeń
- 24 – Włącz/wyłącz tryb wymazywania połączeń („gumka”)

Pasek narzędzi „PLC”



- 1 – Praca w trybie *On-line*
- 2 – Przełączenie trybu monitorowania

- 3 – Bezpośrednie połączenie ze sterownikiem
- 4 – Tryb *On-line* poprzez sieć Ethernet
- 5 – Wstrzymuje monitorowanie po wystąpieniu warunku wyzwalającego
- 6 – Wstrzymuje monitorowanie
- 7 – Ładowanie programu do PLC
- 8 – Odczyt programu z PLC
- 9 – Porównanie bieżącego programu z programem zawartym w PLC
- 10 – Ładowanie zadania do PLC
- 11 – Odczyt zadania z PLC
- 12 – Porównanie bieżącego zadania z zadaniem zawartym w PLC
- 13 – Tryb programowania
- 14 – Tryb debugowania
- 15 – Tryb Monitor
- 16 – Tryb RUN
- 17 – Monitorowanie zbocza sygnału
- 18 – Narzędzie Data Trace
- 19 – Ustawienie hasła zabezpieczającego program
- 20 – Usunięcie hasła zabezpieczającego

Pasek narzędzi „Program”



- 1 – Przełączka okno monitorowania
- 2 – kompiluje bieżący program
- 3 – kompiluje wszystkie programy projektu
- 4 – Edycja w trybie *On-line*
- 5 – Koniec edycji w trybie *On-line*
- 6 – Akceptacja zmian w trybie *On-line*
- 7 – Przejście do edytowanej linii
- 8 – Uruchomienie menadżera sekcji/linii
- 9 – Włączenie trybu tylko do odczytu
- 10 – Rozpoczyna edycję w trybie tylko do odczytu
- 11 – Rezygnacja z wprowadzonych zmian
- 12 – Akceptacja wprowadzonych zmian

Pasek narzędzi „Simulator Debug”



- 1 – Uruchamia symulator zintegrowany z symulatorem paneli operatorskich
- 2 – Uruchamia symulator programowy
- 3 – Uruchamia symulator błędów sterownika
- 4 – Ustawia/usuwa punkt wstrzymania
- 5 – Usuwa wszystkie punkty wstrzymania
- 6 – Uruchamia program (w trybie Monitor)
- 7 – Zatrzymuje i resetuje program (tryb programowania)
- 8 – Wstrzymuje działanie programu (pauza)
- 9 – Wykonuje jeden krok programu
- 10 – Wykonuje jeden krok programu (wchodzi w głąb obiektu zagnieżdzonego, np. FB)
- 11 – Opuszcza obiekt zagnieżdżony (FB)
- 12 – Wykonuje program krok po kroku
- 13 – Wykonuje program przez jeden cykl przetwarzania sterownika

3.2.2. ZAKŁADANIE I KONFIGURACJA NOWEGO PROJEKTU

W niniejszym rozdziale zostaną przedstawione instrukcje dotyczące zakładania i konfiguracji nowego projektu, na przykładzie sterowników CJ1M znajdujących się na stanowiskach ćwiczeniowych w Laboratorium Sterowania Urządzeń i Napędów Przemysłowych, w Instytucie Maszyn Napędów i Pomiarów Elektrycznych w Politechnice Wrocławskiej.

Na rysunku 3.3 przedstawiono drzewo typowego projektu. Każdy nowotworzony projekt posiada charakterystyczną strukturę, w której można wyróżnić następujące elementy składowe:

- *Nazwa projektu* – dowolny identyfikator projektu, który powinien kojarzyć się z realizowanym zadaniem;
- *Nazwa sterownika* – w przypadku, gdy projekt składa się z kilku sterowników, warto im nadawać odpowiednio dobrane nazwy, które ułatwią ich identyfikację;
- *Symbols* – tablica zawierająca nazwy wszystkich symboli i zmiennych globalnych, dostępnych w obrębie całego projektu;
- *IO Table and Unit Setup* – tabela zawierająca listę wszystkich dołączonych modułów rozszerzeń oraz ich ustawienia;

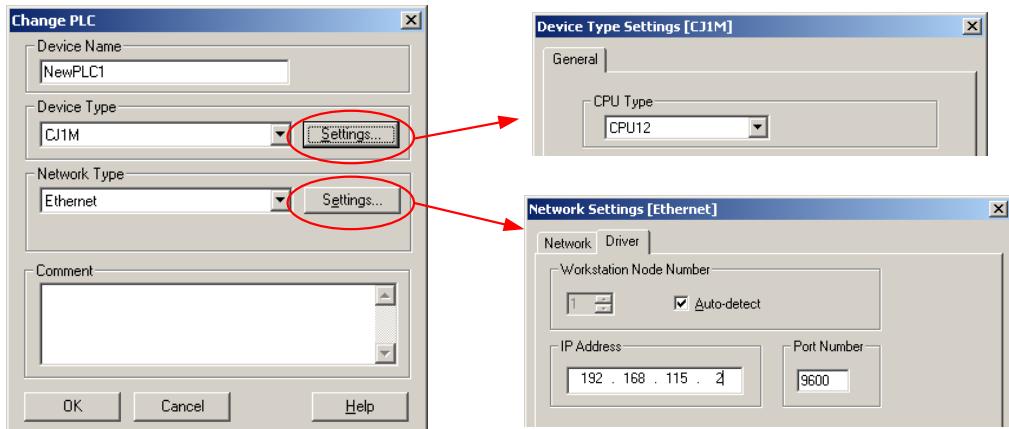
- *Memory card* – narzędzie dostępu do karty pamięci *Compact flash* (dostęp w trybie *online*);
- *Error log* – dziennik błędów sterownika (dostęp w trybie *online*);
- *PLC Clock* – narzędzie ustawień bieżącej daty i czasu sterownika (dostęp w trybie *online*);
- *Settings* – przechowuje ustawienia parametrów sterownika;
- *Memory* – narzędzie dostępu do wszystkich obszarów pamięci sterownika, które mogą być odczytywane i modyfikowane w trybie *online*;
- *Programs* – zawiera programy użytkownika, które są przypisane do zadań (*Tasks*). Poszczególne zadania mogą być wykonywane cyklicznie, lub zdarzeniowo – w postaci przerwań sprzętowych lub programowych. Do programów mogą być dołączone tabele zmiennych lokalnych, które nie są widoczne dla innych programów;
- *Program główny* – wykonywany jest w każdym cyku przetwarzania sterownika;
- *Sekcje* – poszczególne programy mogą być podzielone na sekcje, które są wykonywane jedna po drugiej, w kolejności ich występowania;
- *Function Blocks* – bloki funkcyjne, które mogą być utworzone przez użytkownika lub wczytane z biblioteki.



Rys.3.3. Elementy składowe projektu

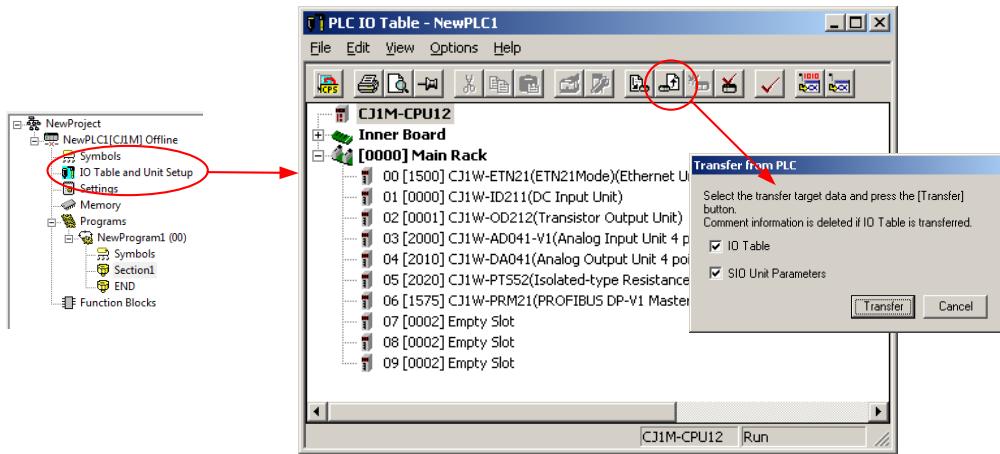
W celu założenia nowego projektu w programie CX-Programmer należy wybrać polecenie *New*, znajdujące się w menu *File*. W wyniku tego pojawi się okno dialogowe „Change PLC”, dotyczące ustawień sterownika PLC (Rys. 3.4). W polu *Device Name* należy nadać dowolną nazwę dla sterownika a następnie wybrać jego

typ, rozwijając listę *Device Type*. Dla wybranego typu sterownika należy określić rodzaj posiadanej jednostki centralnej (w tym przypadku CPU12), w oknie dialogowym, wywoływanym po naciśnięciu przycisku *Settings*. W kolejnej kolejności, w polu *Network Type* należy określić rodzaj interfejsu komunikacyjnego, używanego do zaprogramowania sterownika. Dla sterowników znajdujących się w laboratorium będzie to sieć Ethernet, której parametry trzeba zdefiniować w oknie *Network Settings* (Rys. 3.4), dostępnym po naciśnięciu przycisku *Settings*. W zakładce *Network* nie trzeba wprowadzać żadnych ustawień, natomiast w zakładce *Driver* należy wprowadzić adres IP sterownika programowanego, w zakresie podsieci 192.168.115.xxx. Po zatwierdzeniu wprowadzonych ustawień zostanie otwarte główne okno projektowe, takie jak na rysunku 3.1.



Rys.5.4. Wybór i ustawienia sterownika w programie CX-Programmer

Po założeniu nowego projektu należy utworzyć tabelę wejść/wyjść sterownika (*IO Table*), przedstawiającą konfigurację adresową i ustawienia wszystkich modułów rozszerzeń. Można to zrobić na dwa sposoby. Pierwszy sposób polega na ręcznej edycji tabeli I/O, w której dodaje się kolejne moduły rozszerzeń i wprowadza ich ustawienia, zgodnie z ustawieniami fizycznymi sterownika. Drugi sposób, znacznie prostszy, polega na wczytaniu istniejącej już tabeli I/O, bezpośrednio z pamięci sterownika. W tym celu trzeba połączyć się ze sterownikiem, wybierając polecenie *Work Online* dostępne w menu *PLC* lub na pasku narzędzi. W dalszej kolejności należy otworzyć tabelę I/O, klikając dwukrotnie na pozycji *IO Table and Unit Setup*, znajdującej się w oknie projektu po lewej stronie. Następnie trzeba przetransferować ze sterownika zawartość tej tabeli oraz parametry ustawień modułów specjalnych. Szczegóły tej operacji przedstawiono na rysunku 3.5. Po tych czynnościach można przystąpić do edycji programu użytkownika.



Rys.3.5. Konfiguracja tabeli wejść/wyjść sterownika

3.2.3. EDYCJA PROGRAMU W JĘZYKU DRABINKOWYM

W środowisku CX-Programmer programy mogą być opracowywane w języku drabinkowym, ST oraz SFC. Z racji, że język drabinkowy jest najpopularniejszy, poniżej zostanie omówiony sposób edycji programu użytkownika właśnie w tym języku. Struktura projektu umożliwia podział programu na mniejsze fragmenty, zwane sekcjami. Poszczególne sekcje są wykonywane w każdym cyklu, kolejno jedna po drugiej. Kolejność wykonywania sekcji zależy od ich umiejscowienia w drzewie projektu. Obowiązuje tu reguła, że sekcja położona najwyższej jest wykonywana jako pierwsza.

Po założeniu nowego projektu, automatycznie zostają utworzone dwie sekcje, o nazwach domyślnych *Section1* oraz *END*. W sekcji *END* znajduje się tylko jedna instrukcja – *END*, która jest instrukcją kończącą każdy program i jest wymagana przez kompilator. Użytkownik może umieścić instrukcję *END* w dowolnym miejscu programu, jednak należy pamiętać, że wszystkie instrukcje programu znajdujące się poza nią nie będą wykonywane przez sterownik, a przy próbie komplikacji zostanie wyświetlane ostrzeżenie „*Unreachable code at rung...*”.

Główny program użytkownika jest tworzony w sekcji *Section1*, której nazwę można dowolnie zmienić. W przypadku bardzo rozbudowanych programów, zaleca się podzielenie ich na mniejsze fragmenty, które będą umieszczone w kilku kolejnych sekcjach. Zabieg ten spowoduje zwiększenie czytelności programu oraz ułatwi jego analizę pod kątem znalezienia ewentualnych błędów. Aby dodać nową sekcję do programu należy kliknąć prawym przyciskiem na nazwie programu w drzewie projektu (domyślnie *NewProgram1*) i wybrać polecenie „*Insert new section*”. W

wyniku tej operacji, w strukturze projektu pojawi się dodatkowa sekcja o nazwie domyślnej *Section2*.

Edycja programu w języku drabinkowym polega na umieszczeniu w oknie edytora schematów, odpowiednich instrukcji stykowych i funkcji specjalnych. W trakcie opracowywania programu, CX-Programmer na bieżąco kontroluje poprawność jego składni i podświetla w kolorze czerwonym wszelkie nieprawidłowości, takie jak: niewłaściwe użycie instrukcji, błędna struktura schematu czy nieprawidłowy typ danych. Pełna komplikacja programu możliwa jest po usunięciu wszystkich błędów semantycznych.

Początkujący programiści zazwyczaj tworzą schematy drabinkowe przy użyciu myszy oraz przycisków dostępnych na pasku narzędzi „Diagram”. Z kolei bardziej doświadczeni użytkownicy najczęściej edytują swoje programy bez użycia myszki, wyłącznie za pomocą klawiatury komputera. Metoda ta przy odrobinie wprawy przynosi najlepsze efekty, ponieważ jest wygodniejsza i zdecydowanie szybsza. W środowisku CX-Programmer można włączyć okno informacji (*Ctrl+Shift+I*), które jest umieszczone zawsze na wierzchu i przedstawia najważniejsze predefiniowane skróty klawiszowe, wykorzystywane do szybkiej edycji i uruchamiania programu. Rysunek 3.6 przedstawia wygląd okienka informacji. Oprócz tego, użytkownik może zdefiniować własne skróty klawiszowe dla większości poleceń CX-Programmera, wybierając w menu *Tools* polecenie *Keyboard Mapping*.

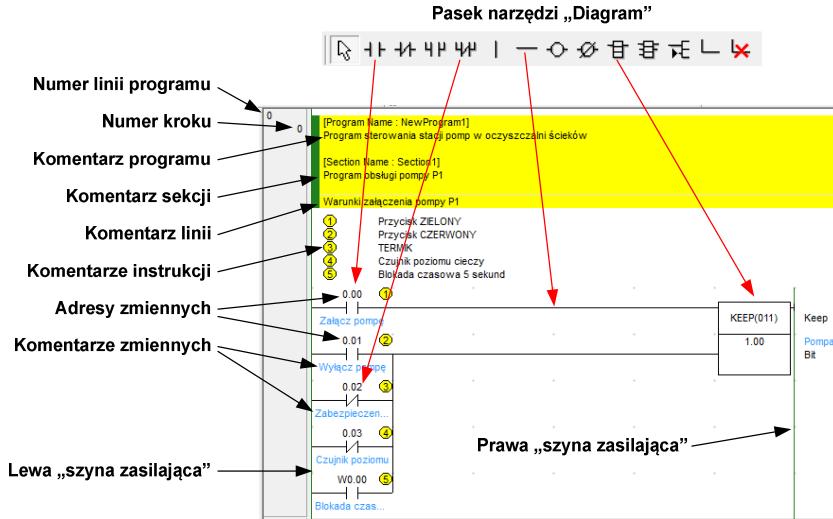


Rys.5.6. Okno informacji o skrótach klawiszowych programu CX-Programmer

Na rysunku 3.7 przedstawiono fragment okna edytora schematów, na którym widać jedną linię przykładowego programu do sterowania pompą. W programie tym wyróżnić można kilka charakterystycznych elementów, do których należą:

- instrukcje stykowe – elementy reprezentujące stan zmiennych boolowskich w postaci styków normalnie otwartych i normalnie zamkniętych. Możliwe są połączenia równoległe i szeregowe styków oraz kombinacje mieszane tych połączeń;
- instrukcje funkcyjne – oznaczane na schemacie w postaci prostokąta. Większość spotykanych instrukcji funkcyjnych to instrukcje kończące linię programu. Oznacza to, że z prawej strony instrukcji nie można podłączyć już żadnego elementu. Na schemacie przykładowym znajduje się instrukcja KEEP, realizująca funkcję przerzutnika RS;

- komentarze programu, sekcji, linii, zmiennych oraz instrukcji – przedstawiają opis tekstowy tych elementów, znacznie ułatwiający późniejszą analizę programu i identyfikację jego fragmentów.



Rys.3.7. Okno edytora schematów – podstawowe elementy programu.

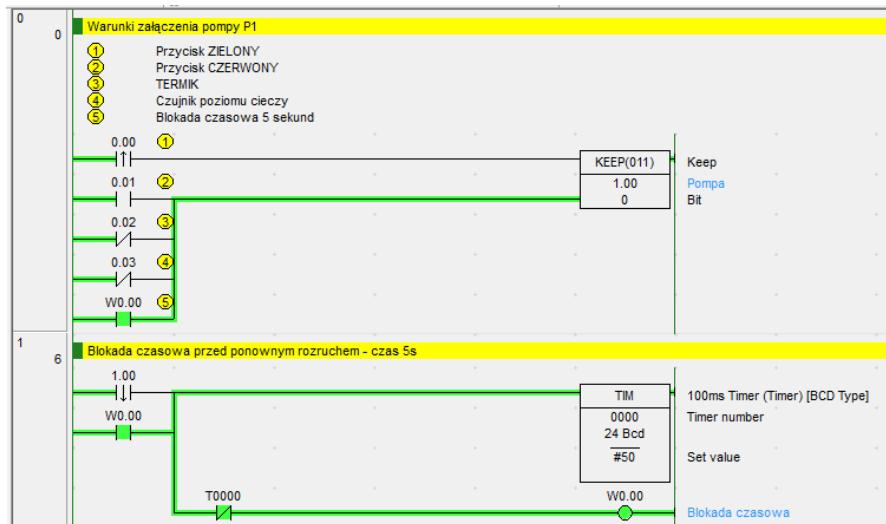
Chociaż umieszczanie komentarzy w programie jest dobrowolne, dobrym zwyczajem jest stosowanie ich możliwie często. Dobrze napisany program, oprócz poprawnego zapisu algorytmu działania, powinien posiadać przejrzystą strukturę i właściwie dobrane komentarze, objaśniające działanie zastosowanych funkcji i algorytmów oraz opisujące znaczenie użytych zmiennych. Z funkcji komentarzy warto często korzystać szczególnie w sterownikach firmy OMRON, ponieważ są one zapisywane w pamięci sterownika równolegle z programem i listą zmiennych. Jest to bardzo istotne z punktu widzenia konieczności wprowadzenia późniejszych zmian w programie, zwłaszcza przez osoby inne niż autor oprogramowania. Wyświetlanie komentarzy w oknie programu można w dowolnej chwili wyłączyć, używając przycisków na pasku narzędzi „Diagram”, skrótów klawiszowych lub poleceń w menu *View*.

3.2.4. URUCHAMIANIE I TESTOWANIE PROGRAMÓW

Przed uruchomieniem programu warto sprawdzić jego poprawność pod kątem występowania wszelkich błędów składniowych i strukturalnych. W tym celu należy użyć polecenia komplikacji, które jest dostępne na pasku narzędzi „Program”, w menu głównym (*PLC / Compile All PLC Programs*) lub poprzez naciśnięcie klawisza F7. W

przypadku, gdy w programie znajdą się jakieś nieprawidłowości, kompilator umieści odpowiednie komunikaty o błędach i ostrzeżeniach w oknie wyjściowym (*Output Window*). W celu odnalezienia zgłoszanego błędu w programie wystarczy dwukrotnie kliknąć na wiersz, w którym znajduje się komunikat o błędzie (*ERROR ...*), wówczas w programie zostanie podświetlona nieprawidłowa instrukcja.

Po usunięciu wszystkich błędów w programie można przystąpić do załadowania programu do sterownika. W tym celu należy połączyć się ze sterownikiem, wybierając polecenie *Work OnLine*. W trybie połączenia *OnLine* tło okna edytora schematu zmienia się z koloru białego na kolor szary. Następnie należy załadować program(y) do sterownika, używając polecenia *Transfer To PLC*. W pojawiającym się oknie dialogowym należy zaznaczyć te składniki, które powinny zostać wysłane do sterownika (*Programs, Settings, Symbols, Comments...*). Po wykonaniu tych czynności sterownik jest gotowy do pracy i może zostać ustawiony w trybie wykonywania programu (*RUN Mode*) lub w trybie monitorowania (*Monitor mode*). W fazie uruchamiania programów prototypowych szczególnie przydatny jest tryb monitorowania, gdyż użytkownik uzyskuje dostęp do wszystkich zasobów pamięciowych sterownika, co daje mu szerokie możliwości testowania działającego programu.

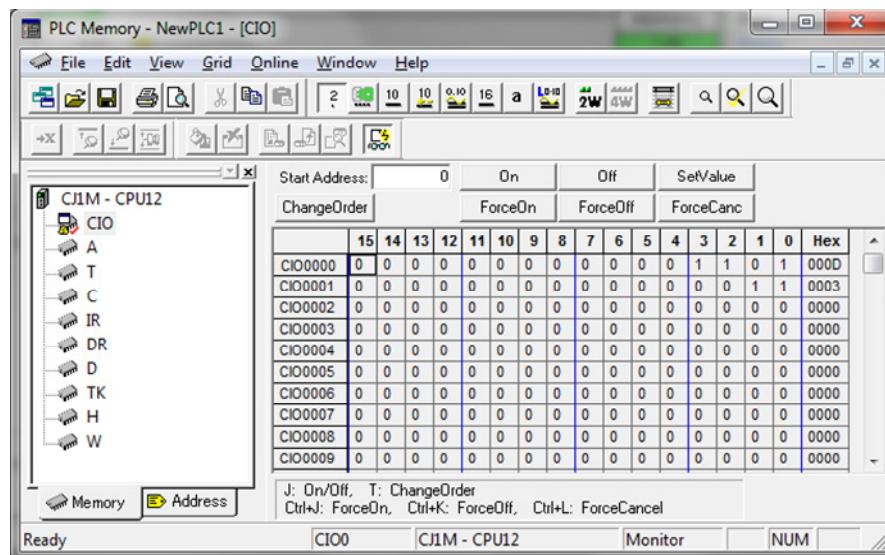


Rys.3.8. Przykładowy program w trybie monitorowania

Na rysunku 3.8 przedstawiono okno podglądu działania przykładowego programu, uruchomionego w sterowniku w trybie monitorowania. Użytkownik na bieżąco może obserwować występowanie wysokich stanów logicznych na poszczególnych elementach i liniach sygnałowych, które są zaznaczone kolorem zielonym. Oprócz

tego, tryb monitorowania daje możliwość zmiany wartości większości zmiennych logicznych i modyfikacji zawartości innych komórek pamięci (np. zmiana wartości liczników, parametrów wejściowych czasomierzy, itp.).

Na rysunku 3.9 przedstawiono okno narzędzia „PLC Memory”, które pozwala na dostęp do wszystkich obszarów pamięci sterownika i umożliwia podgląd komórek pamięci oraz edycję ich wartości. Po lewej stronie znajduje się okno wyboru typu pamięci. W prawej części z kolei znajduje się wyświetlana zawartość wybranego obszaru. Po włączeniu trybu monitorowania można na bieżąco podglądać i modyfikować zawartość poszczególnych komórek. Wszystkie wprowadzane zmiany są na bieżąco aktualizowane w fizycznej pamięci sterownika, dzięki czemu można w czasie rzeczywistym zmieniać parametry i bezpośrednio oddziaływać na przebieg sterowanego procesu.



Rys.3.9. Okno programu PLC Memory w trybie monitorowania pamięci CIO

Oprócz przedstawionych podstawowych możliwości testowania programów użytkownika w czasie rzeczywistym, środowisko CX-Programmer oferuje znacznie więcej funkcji ułatwiających debugowanie i śledzenie przebiegu działania programów. Do najciekawszych narzędzi zaliczyć można:

- **Differential Monitor** – służy do wykrywania i zliczania krótkotrwałych stanów logicznych (zboczy) i sygnalizacji ich wystąpienia w postaci graficznej i akustycznej;
- **Pause Upon Trigger** – funkcja, która zatrzymuje tryb monitorowania po wystąpieniu określonego zdarzenia wyzwalającego;

- **Data Trace** oraz **Time Chart Monitor** – programy do monitorowania i wyświetlania zmian sygnałów w funkcji czasu. Ich działanie przypomina pracę wielokanałowego oscyloskopu, który rejestruje sygnały w takt zadanej podstawy czasu, po wystąpieniu określonych warunków wyzwalania;
- **Switch Box Utility** – narzędzie do wygodnej zmiany wartości bitów i słów wybranych obszarów pamięci.

Przedstawione wyżej narzędzia współpracują również z symulatorem programowym, zintegrowanym w środowisku CX-Programmer. Symulator ten umożliwia przetestowanie działania programów użytkownika wirtualnie, bez konieczności posiadania sterownika rzeczywistego. Narzędzie to jest niezwykle przydatne dla początkujących programistów, którzy mają możliwość przetestowania większości instrukcji i funkcji języka drabinkowego, dostępnych dla różnych sterowników firmy OMRON.

Aby uruchomić symulator należy wybrać polecenie *Work Online Simulator* znajdujące się w menu *Simulation* lub nacisnąć odpowiedni przycisk na pasku narzędzi *Simulator Debug*. W tym momencie testowany program zostanie uruchomiony w wirtualnym sterowniku, w trybie monitorowania, dając użytkownikowi możliwość podglądu wartości poszczególnych zmiennych i stanów logicznych na połączeniach sygnałowych. W trybie monitorowania można również modyfikować wartości różnych zmiennych oraz zmieniać stany logiczne instrukcji stykowych. W celu zmiany wartości wybranej komórki lub styku, należy umieścić na niej kursor i nacisnąć klawisz *Enter*. W pojawiającym się oknie dialogowym należy wprowadzić nową wartość w polu *Value* i zatwierdzić jej zmianę klawiszem *OK*. Wprowadzone zmiany dają efekt natychmiastowy, powodując odpowiednią reakcję programu użytkownika.

4. Wybrane instrukcje i funkcje języka drabinkowego dla sterowników OMRON

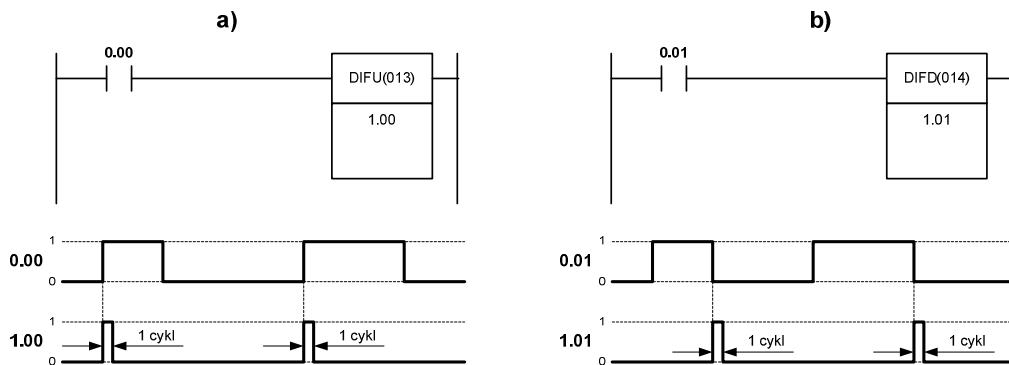
4.1. Instrukcje detekcji zboczy

Detektory zboczy należą do podstawowych instrukcji bitowych, które wykrywają dynamiczne zmiany sygnałów cyfrowych. Wyróżnia się dwa rodzaje zboczy:

- zbocze narastające – zmiana wartości logicznej ze stanu niskiego na wysoki ($0 \rightarrow 1$);
- zbocze opadające – zmiana wartości logicznej ze stanu wysokiego na niski ($1 \rightarrow 0$);

4.1.1. INSTRUKCJE DIFU I DIFD

Do wykrywania zbocza narastającego stosuje się instrukcję **DIFU** (*Differentiate Up*). Instrukcja ta posiada jedno wejście sygnałowe oraz jeden operand wyjściowy – dowolny adres bitowy z obszarów pamięci CIO, W, H, A, IR. Jeżeli wartość na wejściu sygnałowym zmieni się ze stanu 0 na 1, operand wyjściowy zostanie ustawiony na wartość logiczną 1 na czas trwania jednego cyklu przetwarzania sterownika.



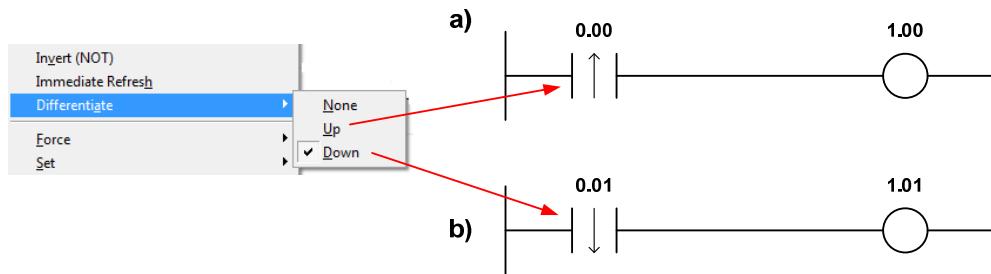
Rys.4.1. Działanie instrukcji DIFU (a) i DIFD (b)

Do wykrywania zbocza opadającego służy instrukcja **DIFD** (*Differentiate Down*), która działa analogicznie do **DIFU**, z tą różnicą, że reaguje na zmianę stanu sygnału wejściowego z wartości 1 na 0.

Na rysunku 4.1 przedstawiono przykładowe programy wykorzystujące instrukcje DIFU i DIFD oraz wykresy czasowe, objaśniające działanie tych instrukcji.

4.1.2. INSTRUKCJE STYKOWE RÓŻNICZKUJĄCE

W programie CX-Programmer dostępne są również różniczkujące instrukcje stykowe, reagujące na zmiany stanów logicznych skojarzonych z nimi zmiennych. W celu wprowadzenia warunku różniczkowania do instrukcji stykowej, należy kliknąć na niej prawym przyciskiem myszy i wybrać opcję *Differentiate>Up* lub *Differentiate>Down*. Wówczas na symbolu instrukcji stykowej pojawi się strzałka skierowana do góry (dla detektora zbocza narastającego) lub w dół (dla detektora zbocza opadającego). Przykład wykorzystania tych instrukcji przedstawiono na rysunku 4.2.



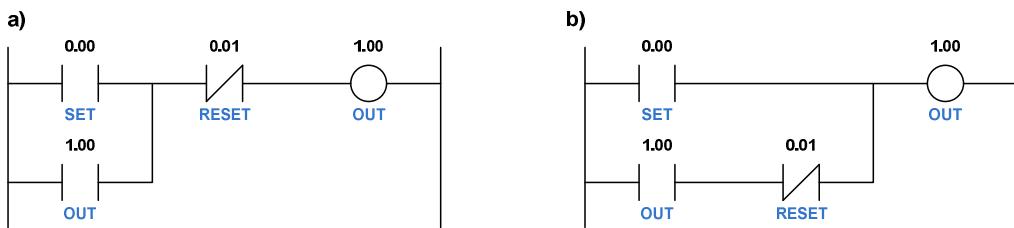
Rys.4.2. Zastosowanie styków wykrywających zbocze narastające (a) i opadające (b)

4.2. Elementy bistabilne

Podstawowymi elementami bistabilnymi są przerzutniki RS i SR, zaliczane do najprostszych elementów pamięciowych. Elementy te posiadają dwa wejścia i jedno wyjście. Wejście ustawiające (SET) powoduje trwałe ustawienie stanu wysokiego na wyjściu przerzutnika, natomiast wejście kasujące (RESET) powoduje zapamiętanie stanu niskiego na wyjściu. Różnica pomiędzy przerzutnikami RS i SR polega na odmiennym priorytecie sygnałów wejściowych. W przypadku, gdy na oba wejścia zostanie jednocześnie podany stan wysoki, wyjście przerzutnika RS przyjmie niski, zaś wyjście przerzutnika SR ustawi się w stan wysoki.

4.2.1. ZASTOSOWANIE INSTRUKCJI STYKOWYCH DO REALIZACJI FUNKCJI PRZERZUTNIKÓW

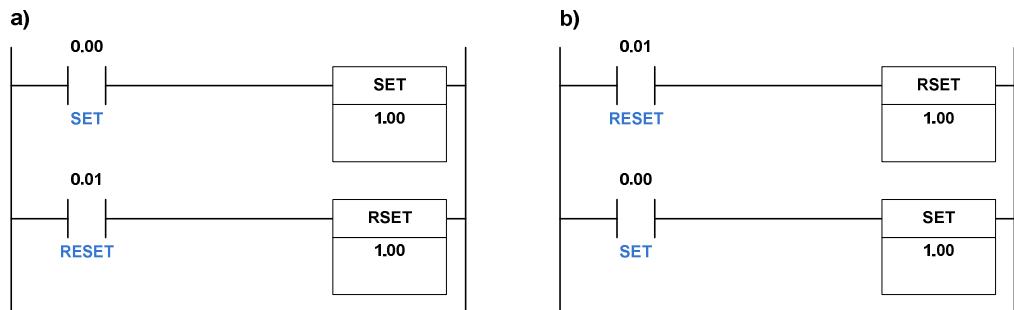
W programie CX-Programmer, w języku drabinkowym istnieje kilka możliwości realizacji funkcji przerzutników RS i SR. Jedną z nich jest wykorzystanie zwykłych instrukcji stykowych, połączonych w układzie sprzężenia zwrotnego od wyjścia (tzw. układy z podtrzymaniem). Przykład wykorzystania instrukcji stykowych do realizacji obu typów przerzutników przedstawiono na rysunku 4.3.



Rys.4.3. Zastosowanie instrukcji stykowych do realizacji funkcji przerzutników RS (a) i SR (b).

4.2.2. INSTRUKCJE SET I RSET

Inną metodą realizacji przerzutników RS i SR jest wykorzystanie instrukcji **SET** i **RSET**, które posiadają po jednym wejściu i jednym operandzie wyjściowym. Instrukcje te zazwyczaj występują w parze (dla tego samego operandu wyjściowego), choć nie koniecznie muszą być użyte w jednym miejscu programu. Podanie stanu wysokiego na wejście instrukcji SET powoduje trwałe przypisanie jedynki logicznej do operandu wyjściowego. Konsekwentnie, w instrukcji RSET, podanie na wejście stanu wysokiego spowoduje wyzerowanie operandu wyjściowego.

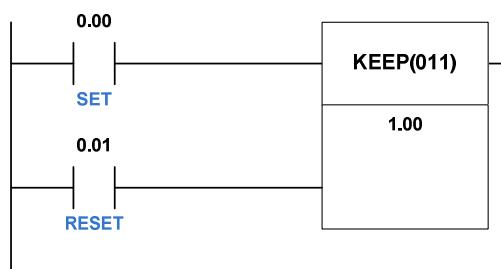


Rys.4.4. Zastosowanie instrukcji SET i RSET do realizacji przerzutników RS (a) i SR (b).

Ponieważ instrukcje SET i RSET występują w programie osobno, kolejność ich występowania na schemacie drabinkowym będzie jednoznacznie określać priorytet działania tak powstałego przerzutnika, przy jednoczesnym podaniu stanów wysokich na wejścia tych instrukcji. Dominującą instrukcją w tym przypadku będzie ta, która w programie zostanie wykonana jako ostatnia. Na rysunku 4.4 przedstawiono przykład realizacji przerzutników RS i SR przy wykorzystaniu instrukcji SET i RSET.

4.2.3. INSTRUKCJA KEEP

Oprócz przedstawionych wyżej metod realizacji funkcji przerzutników RS i SR, w programie CX-Programmer dostępna jest instrukcja **KEEP**, będąca gotowym odpowiednikiem klasycznego przerzutnika RS. Instrukcja ta posiada dwa wejścia (SET i RESET) oraz jeden operand wyjściowy, typu boolowskiego. Sposób wykorzystania tej instrukcji przedstawiono na rysunku 4.5.



Rys.4.5. Zastosowanie instrukcji KEEP realizującej funkcję przerzutnika RS.

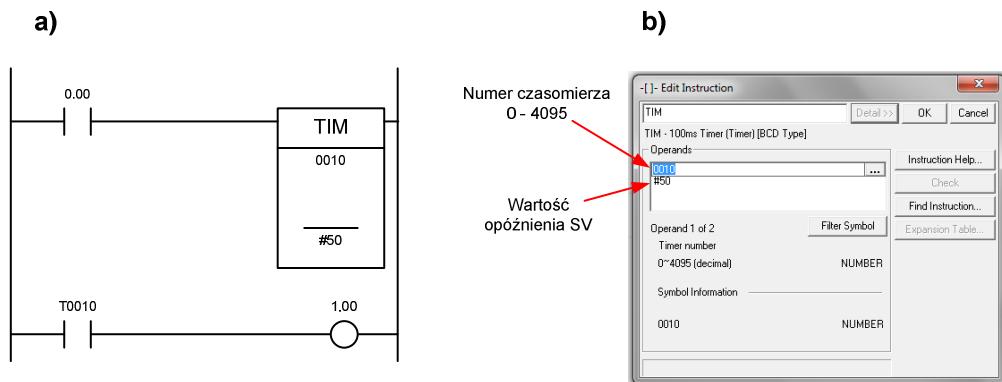
4.3. Czasomierze

W sterownikach programowalnych OMRON rodziny CJ występuje kilka instrukcji realizujących funkcje czasowe. Wszystkie czasomierze wykorzystują wspólny obszar pamięci typu T, do przechowywania aktualnych wartości opóźnienia oraz stanów logicznych ich flag wypełnienia. W sterownikach serii CJ można wykorzystać maksymalnie 4096 czasomierzy.

4.3.1. INSTRUKCJE TIM, TIMH I TMHH

Podstawową instrukcją realizującą funkcję czasomierza jest instrukcja **TIM**. Czasomierz ten umożliwia odmierzanie czasu w zakresie 0-999,9s z rozdzielcością 100ms. Wartość zadaną opóźnienia wprowadza się w postaci operandu SV (ang. *Set Value*), reprezentującego liczbę całkowitą w kodzie BCD, która wyraża krotność

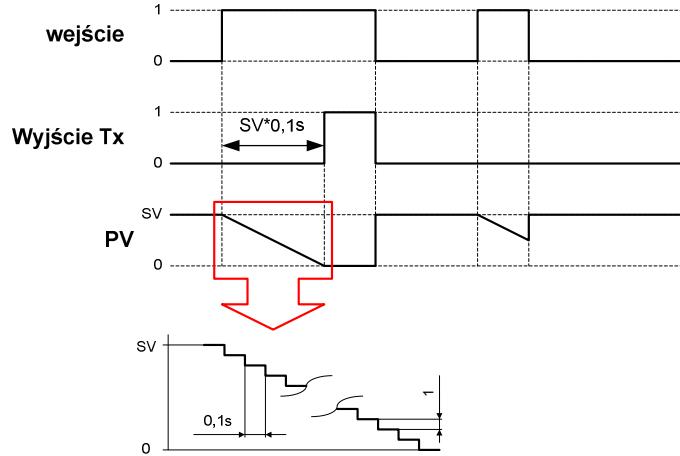
jednostkowego opóźnienia o wartości 0,1s. Przykładowo, chcąc uzyskać czas opóźnienia równy 5s, należy wprowadzić wartość wejściową #50 ($50 \times 0,1\text{s} = 5\text{s}$). Każdy użyty czasomierz TIM musi posiadać swój unikalny numer z zakresu 0-4095, który jednoznacznie identyfikuje go w obszarze pamięci T. Ponadto, numer ten związany jest bezpośrednio z tzw. flagą wypełnienia czasomierza, która zostaje ustawiona po odmierzeniu ustalonego czasu. Podobnie jak większość instrukcji w języku drabinkowym sterowników OMRON, instrukcja TIM jest instrukcją kończącą linię programu. Na rysunku 4.6 przedstawiono przykład zastosowania czasomierza oraz sposób parametryzacji instrukcji TIM.



Rys.4.6. Zastosowanie instrukcji TIM – program przykładowy (a) i edycja parametrów czasomierza (b)

W programie przykładowym przedstawionym na rysunku 6.6a, czasomierz TIM (o numerze T0010) działa w układzie opóźniającym załączenie wyjścia sterownika. Po ustawieniu stanu wysokiego na wejściu 0.00, czasomierz rozpoczyna odliczanie czasu opóźnienia, odejmując co 100ms wartość 1, od wartości początkowej 50 do zera. Z chwilą wyzerowania czasomierza (po upływie 5s) następuje ustawienie znacznika wypełnienia T10, który powoduje ustawienie stanu wysokiego na wyjściu 1.00 sterownika. Stan taki trwa do momentu, kiedy na wejściu 0.00 pojawi się niski stan logiczny. Wykres czasowy objaśniający działanie czasomierza TIM przedstawiono na rysunku 4.7.

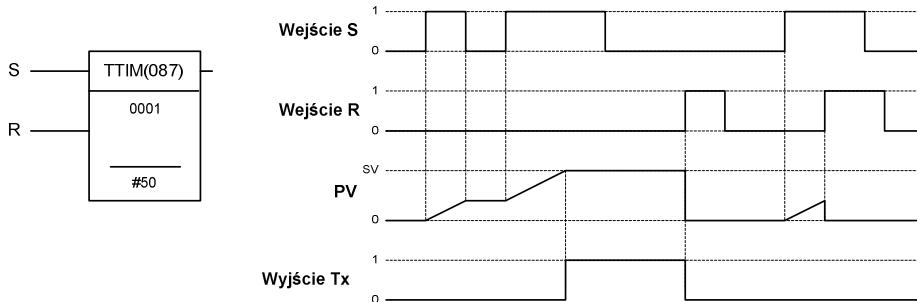
W sterownikach CJ1 czasomierze TIM występują dodatkowo w dwóch „szybszych” wersjach: **TIMH (High-Speed Timer)** oraz **TTMH (Ultra-High-Speed Timer)**. Zasada działania tych czasomierzy jest taka sama jak czasomierza TIM, z tą różnicą, że wartość zadana SV jest dekrementowana z krokiem 1/100 s w czasomierzu TIMH oraz z krokiem 1/1000 s w czasomierzu TTMH. Przy stosowaniu ultraszybkiego czasomierza typu TTMH należy pamiętać, aby jego numer był wybierany z przedziału 0–15. W przeciwnym wypadku czasomierz może działać niedokładnie.



Rys.4.7. Wykres czasowy czasomierza TIM

4.3.2. CZASOMIERZ AKUMULUJĄCY TTIM

Oprócz standardowego czasomierza, realizującego funkcję opóźnionego załączenia, w programie CX-Programmer dostępny jest tzw. czasomierz akumulujący, uruchamiany za pomocą instrukcji **TTIM** (ang. *Totalising Timer*). Czasomierz ten posiada dwa wejścia: **S** (*SET*) – wejście wyzwalające oraz **R** (*RESET*) – wejście kasujące.



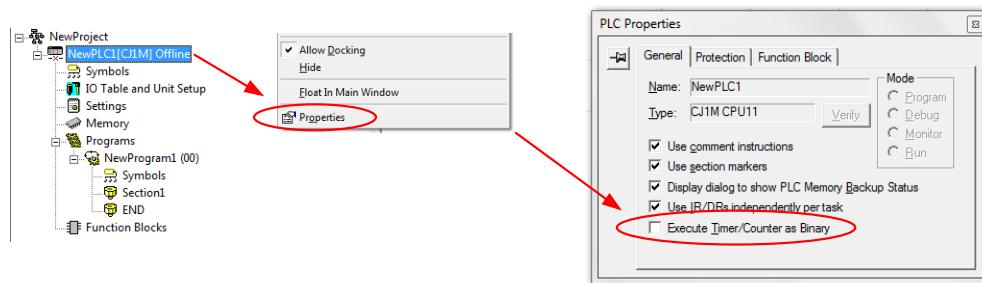
Rys.4.8. Zasada działania czasomierza TTIM

Podanie na wejściu S wysokiego stanu logicznego powoduje inkrementację wartości PV czasomierza z krokiem 100ms. Po wyłączeniu sygnału wyzwalającego, czasomierz przestaje zwiększać wartość PV, a jej ostatnia wartość zostaje zapamiętana. Ponowne podanie stanu wysokiego na wejściu S powoduje dalszą inkrementację wartości PV (akumulacja). Flaga wypełnienia czasomierza zostaje ustawiona z chwilą, gdy wartość PV osiągnie wartość zadaną, wprowadzoną jako

operand SV. Wejście R służy do wyzerowania wartości PV czasomierza i skasowania flagi wypełnienia. Na rysunku 4.8 przedstawiono wykres czasowy obrazujący działanie czasomierza TTIM.

4.3.3. ZMIANA FORMATU DANYCH CZASOMIERZY

Wszystkie przedstawione czasomierze (TIM, TIMH, TTIM, TTIMH) domyślnie operują na liczbach całkowitych w formacie BCD. W sterownikach serii CJ istnieje możliwość zamiany typu danych w czasomierzach i licznikach, z formatu BCD na format binarny. Po tej zamianie, wartości opóźnienia SV mogą być wprowadzane jako liczby dziesiętne z zakresu 0–65535, lub w zapisie szesnastkowym w przedziale 0–FFFF. Po zamianie typu na binarny, wszystkie mnemoniki instrukcji czasomierzy i liczników otrzymują w nazwie przyrostek X ($TIM \rightarrow TIMX$, $TTIM \rightarrow TTIMX$, ...). Należy jednak pamiętać, że w jednym projekcie można używać tylko jednego formatu danych – niedopuszczalne jest jednoczesne użycie czasomierzy TIM i TIMX. Dlatego też wybór typu danych należy dokonać na etapie zakładania projektu. Ustawienia te można zmienić klikając prawym klawiszem myszy na nazwie sterownika znajdującej się w drzewie projektu (NewPLC1), a następnie wybraniu opcji *Properties* i zaznaczeniu pola „*Execute Timer/Counter as Binary*”. Szczegóły tej zmiany przedstawia rysunek 4.9.



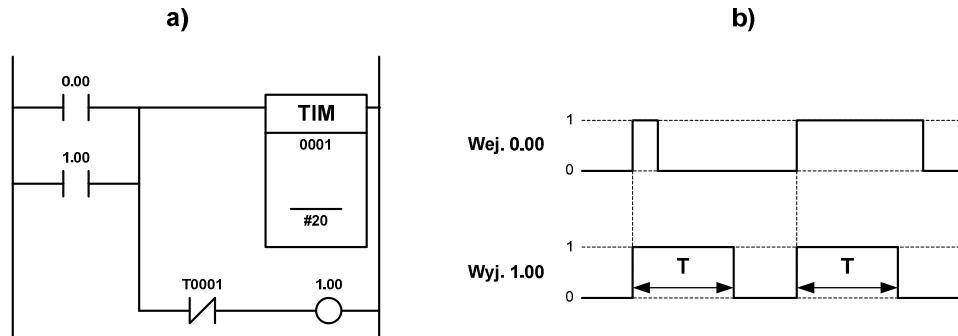
Rys.4.9. Wybór typu danych czasomierzy i liczników

4.3.4. PROGRAMOWA REALIZACJA FUNKCJI CZASOWYCH TYPU TP I TOF

W normie IEC 61131 zdefiniowano trzy typy standardowych czasomierzy: TON, TOF i TP. Analizując działanie czasomierza TIM można stwierdzić, że jest to dokładna implementacja funkcji czasomierza **TON** (*On-Delayed Timer*). Jednak w środowisku CX-Programmer użytkownik nie znajdzie gotowych instrukcji, które realizowałyby funkcje czasomierza impulsowego **TP** (*Pulse Timer*) i czasomierza z opóźnionym wyłączeniem **TOF** (*Off-Delayed Timer*). Ponieważ są to dość często

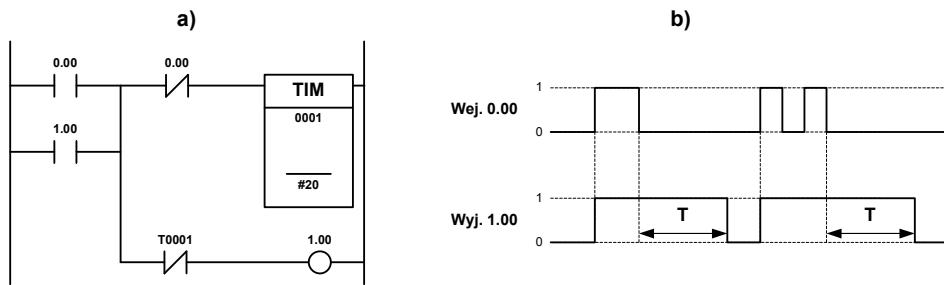
stosowane struktury czasowe, poniżej zostaną przedstawione możliwości programowej realizacji funkcji tych czasomierzy.

Na rysunku 4.10 przedstawiono prosty program realizujący funkcję czasomierza impulsowego TP oraz jego wykres czasowy. Podanie stanu wysokiego na wejście 0.00 powoduje wyzwolenie czasomierza TIM. W tej samej chwili, zanegowana flaga T0001 powoduje ustawienie stanu wysokiego na wyjściu 1.00. Wyjście to zostało użyte w roli bitu podtrzymywania, który równolegle połączony z wejściem sterującym 0.00, zapewnia ciągłość działania czasomierza TIM po ustawienia stanu niskiego na wyjściu 0.00.



Rys.4.10. Realizacja funkcji czasomierza TP: a) – program w języku drabinkowym, b) – wykres czasowy.

Rysunek 4.11 przedstawia przykładową implementację funkcji czasomierza TOF. Załączenie wejścia 0.00 powoduje przepływ sygnału przez szeregowo połączoną, zanegowaną flagę T0001 i ustawienie stanu wysokiego wyjściu 1.00. W chwili wyłączenia wejścia 0.00 następuje wyzwolenie czasomierza TIM, który po upływie nastawionego czasu spowoduje ustawienie flagi T0001 i jednocześnie wyłączenie wyjścia 1.00. Jeżeli przed upływem nastawionego czasu wejście 0.00 ponownie zostanie przełączone w stan wysoki, czasomierz TIM zostanie zresetowany, a cała akcja powtórzy się po zniknięciu sygnału wyzwalającego na wejściu 0.00.



Rys.4.11. Realizacja czasomierza TOF: a) – program w języku drabinkowym, b) – wykres czasowy.

4.4. Liczniki

W sterownikach serii CJ1 wszystkie liczniki wykorzystują wspólny obszar pamięci typu C, o zakresie adresowym 0–4095. W jednym projekcie można zastosować maksymalnie 4096 liczników, z których każdy zajmuje w pamięci 1 słowo i musi posiadać swój unikalny numer, odpowiadający adresowi komórki z obszaru C.

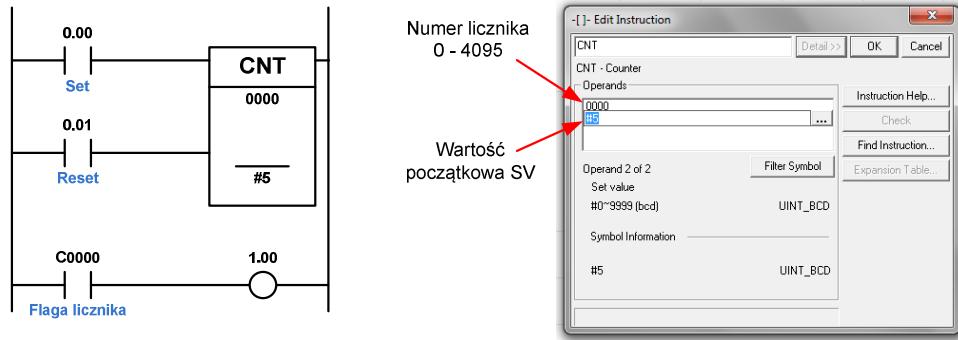
4.4.1. LICZNIK CNT

Instrukcja **CNT** (*Counter*) pełni funkcję podstawowego licznika, którego zadaniem jest zliczanie impulsów pojawiających się na jego wejściu S. W tym przypadku, impuls rozumiany jest jako zmiana stanu logicznego z wartości 0 na wartość 1 (zbocze narastające). Licznik CNT jest licznikiem zliczającym w dół, co oznacza, że z każdym impulsem wejściowym następuje dekrementacja wartości PV licznika, począwszy od wartości początkowej, zadeklarowanej jako parametr SV (*Set Value*). Z chwilą osiągnięcia wartości zerowej zostaje ustawiona flaga wypełnienia licznika – bit Cx, gdzie indeks x odpowiada numerowi licznika. Drugie wejście licznika R służy do skasowania flagi wypełnienia i wyzerowania licznika, polegającego na przypisaniu do PV wartości początkowej SV. Na rysunku 4.12 przedstawiono przykład zastosowania instrukcji CNT. Z kolei rysunek 4.13 przedstawia wykres czasowy obrazujący działanie licznika.

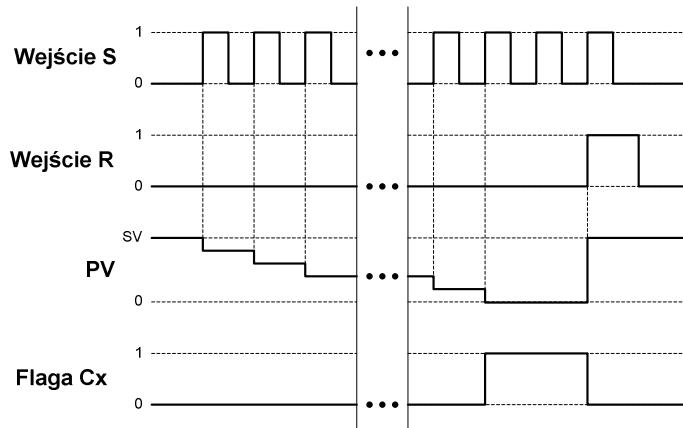
4.4.2. LICZNIK REWERSYJNY CNTR

Instrukcja **CNTR** realizuje funkcję dwukierunkowego licznika, posiadającego dwa wejścia liczące: UP (w górę) i DN (w dół) oraz wejście kasujące R. Licznik CNTR jest licznikiem cyklicznym (okrężnym), co oznacza, że po osiągnięciu wartości maksymalnej proces zliczania w górę kontynuowany jest od wartości zero. Podobna zasada obowiązuje w kierunku zliczania w dół. Maksymalna wartość, do której licznik zlicza impulsy jest określona przez użytkownika jako parametr SV. Po osiągnięciu wartości maksymalnej, kolejny impuls pojawiający się na wejściu UP powoduje wyzerowanie licznika (PV=0) i ustawienie znacznika wypełnienia Cx. Analogicznie, w chwili gdy licznik posiada wartość zerową a na wejściu DN pojawi się impuls, wartość licznika zostanie ustawiona na maksymalną (PV=SV) i również w tym przypadku nastąpi ustawienie flagi wypełnienia Cx. Każdy impuls na wejściach UP lub DN niepowodujący przepełnienia licznika kasuje znacznik wypełnienia. Wejście R służy do natychmiastowego wyzerowania wartości licznika i skasowania flagi wypełnienia.

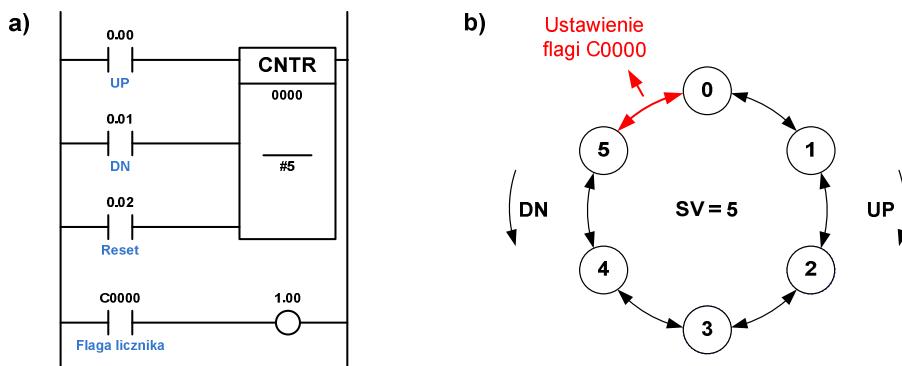
Na rysunku 4.14 przedstawiono przykład wykorzystania licznika CNTR oraz zasadę pracy cyklicznej, a na rysunku 4.15 jego wykres czasowy.



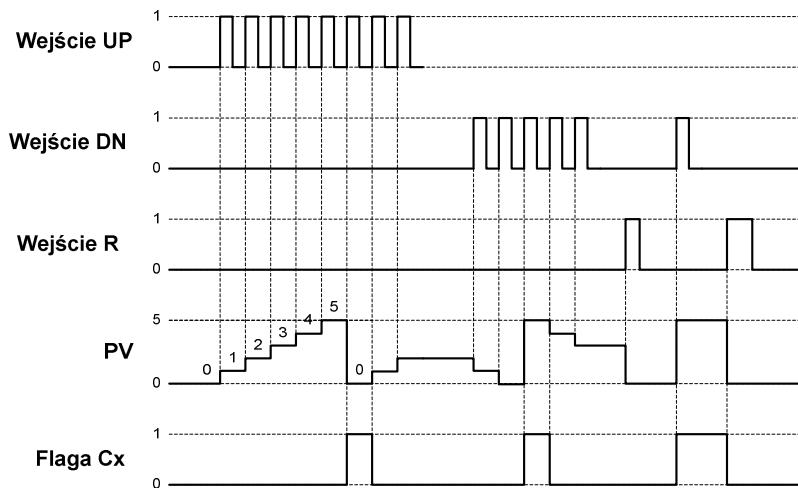
Rys.4.12. Przykład zastosowania instrukcji CNT.



Rys.4.13. Wykres czasowy obrazujący zasadę działania licznika CNT.



Rys.4.14. Przykład użycia licznika CNTR (a) oraz zasada jego pracy cyklicznej (b).



Rys.4.15. Wykres czasowy obrazujący zasadę działania licznika rewersyjnego CNTR

4.5. Komparatory

Komparatory są to elementy funkcyjne, które wykonują operację porównania ze sobą dwóch wartości liczbowych, najczęściej typu całkowitego lub rzeczywistego. Jeżeli zadany warunek porównania jest spełniony, zostają ustawione odpowiednie flagi.

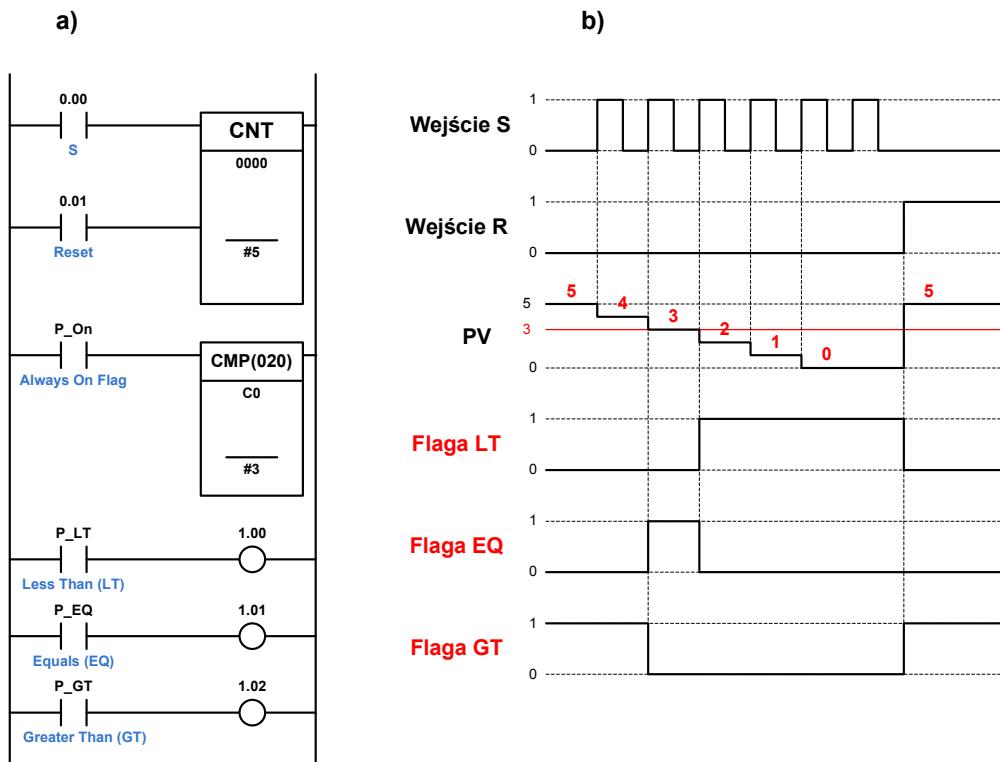
4.5.1. KOMPARATORY CMP I CPS

Instrukcja **CMP** realizuje funkcję komparatora, którego argumentami wejściowymi są liczby całkowite bez znaku, o długości jednego słowa (UINT). W wyniku porównania zostają ustawione systemowe flagi porównania, które określają relację pomiędzy porównywanyymi wartościami. Instrukcją pokrewną, działającą na liczbach całkowitych ze znakiem (INT) jest instrukcja **CPS**. W tabeli 4.1 zestawiono wszystkie systemowe flagi porównania oraz odpowiadające im operatory relacji.

Na rysunku 4.16a przedstawiono przykładowy program zawierający licznik CNT i komparator CMP, który porównuje bieżącą wartość licznika z wartością #3. W zależności od wyniku tego porównania zostają odpowiednio ustawione flagi *P_LT*, *P_EQ* i *P_GT*, które dołączone są do kolejnych wyjść sterownika (1.00–1.02). Przedstawiony wykres czasowy (rys.4.16b) objaśnia zasadę działania komparatora CMP, w połączeniu z systemowymi flagami porównania.

Tab.4.1. Systemowe flagi porównania

Symbol	Operator relacji	Nazwa flagi
P_LT	<	Less Than Flag
P_LE	<=	Less Than or Equal Flag
P_EQ	=	Equal Flag
P_GE	>=	Greater Than or Equal Flag
P_GT	>	Greater Than Flag
P_NE	\diamond	Not Equal Flag

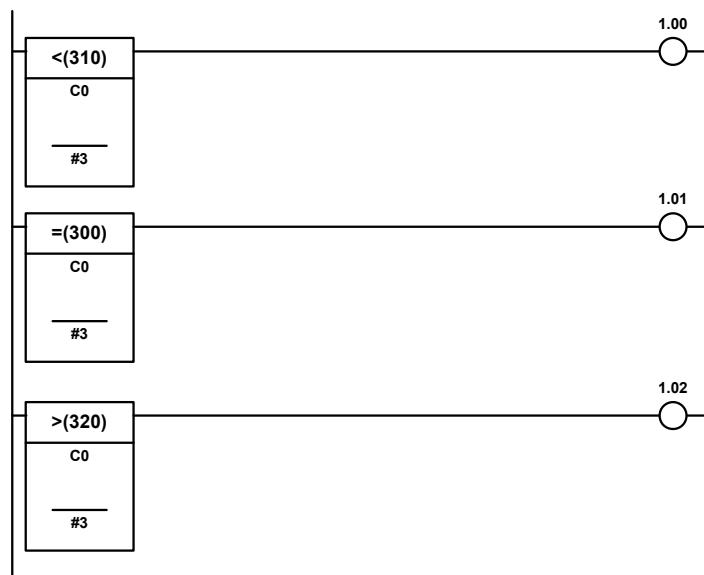


Rys.4.16. Przykładowy program z użyciem komparatora CMP (a) oraz wykres czasowy obrazujący jego działanie (b).

4.5.2. BEZPOŚREDNIE INSTRUKCJE PORÓWNANIA

W sterownikach serii CJ możliwe jest zastosowanie bezpośrednich instrukcji porównania, które mogą być wstawione w dowolnym miejscu schematu drabinkowego. Instrukcje te posiadają jedno wejście boolowskie (warunek wykonywalności) oraz jedno wyjście, na którym pojawia wysoki stan logiczny, jeżeli warunek porównania jest spełniony. Nazwa instrukcji (mnemonik) wskazuje bezpośrednio na wykonywaną przez nią operację porównania (np. $=$, $<$, \geq). Na rysunku 4.17 przedstawiono fragment przykładowego programu, w którym wykorzystano bezpośrednie instrukcje porównania. W programie tym instrukcje te porównują aktualną wartość licznika $C0$ z wartością liczbową #3 i bezpośrednio sterują wyjściami sterownika, bez potrzeby użycia flag porównania. Na wyjściu 1.00 pojawi się wysoki stan logiczny, wówczas, kiedy wartość licznika $C0$ będzie mniejsza od 3. Analogicznie, wyjście 1.01 zostanie ustawione w stan wysoki, gdy licznik $C0$ osiągnie wartość 3, zaś po jej przekroczeniu zostanie ustawione wyjście 1.02.

Bezpośrednie instrukcje porównania w podstawowej formie wykonują działania na liczbach całkowitych bez znaku, o długości jednego słowa (*UINT*). Możliwe jest stosowanie dodatkowych modyfikatorów tych instrukcji, które pozwalają na działanie na liczbach o długości 32 bitów (*Double Length*) i/lub na liczbach całkowitych ze znakiem (*Signed*). W tabeli 4.2 przedstawiono wykaz tych modyfikatorów oraz opis ich działania.



Rys.4.17. Przykładowy program z użyciem bezpośrednich instrukcji porównania

Tab.4.2. Modyfikatory bezpośrednich instrukcji porównania

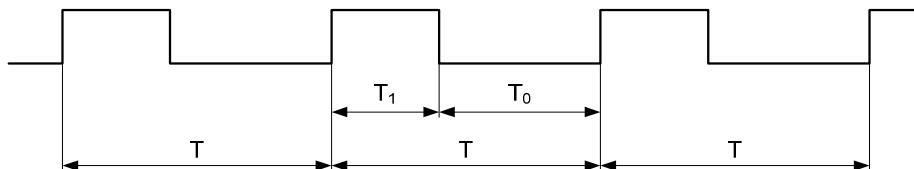
Modyfikator	Typ danych	Przykład zastosowania
brak	liczby całkowite bez znaku w formacie binarnym lub BCD (<i>Unsigned Data</i>)	=, >, <, >=, <>
*L	liczby całkowite bez znaku podwójnej długości (<i>Unsigned Double Length</i>)	=L, <L, >=L
*S	liczby całkowite ze znakiem w formacie binarnym (<i>Signed Data</i>)	=S, >S, <S
*SL	liczby całkowite ze znakiem podwójnej długości (<i>Signed Double Length</i>)	=SL, >=SL, <SL

4.6. Generatory taktujące

Generator taktujący jest elementem, który wytwarza sygnał prostokątny o zadanych parametrach czasowych. Do podstawowych parametrów charakteryzujących przebiegi prostokątne zalicza się:

- częstotliwość f ,
- okres sygnału T (odwrotność częstotliwości),
- czas trwania wysokiego stanu logicznego T_1 ,
- czas trwania niskiego stanu logicznego T_0 ,
- współczynnik wypełnienia $d=T_1/T$.

Na rysunku 4.18 przedstawiono przykładowy przebieg sygnału prostokątnego, na którym zaznaczono jego podstawowe parametry czasowe.



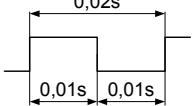
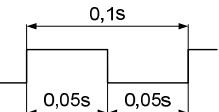
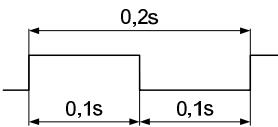
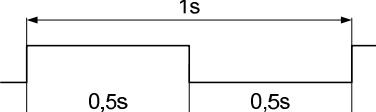
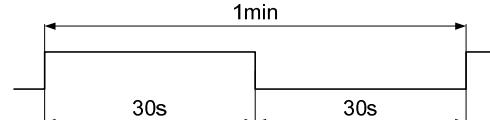
Rys.4.18. Podstawowe parametry czasowe przebiegu prostokątnego

4.6.1. SYSTEMOWE GENERATORY TAKTUJĄCE

W sterownikach programowalnych OMRON dostępne są specjalne flagi systemowe, reprezentujące wyjścia wewnętrznych generatorów sygnałów prostokątnych, o różnych częstotliwościach i stałym współczynniku wypełnienia, wynoszącym 0,5 (50%). Zastosowanie tych sygnałów w programie użytkownika

polega na przypisaniu odpowiedniej flagi generatora do instrukcji stykowej, typu boolowskiego. W tabeli 4.3 przedstawiono wykaz dostępnych flag generatorów taktujących oraz ich parametry czasowe.

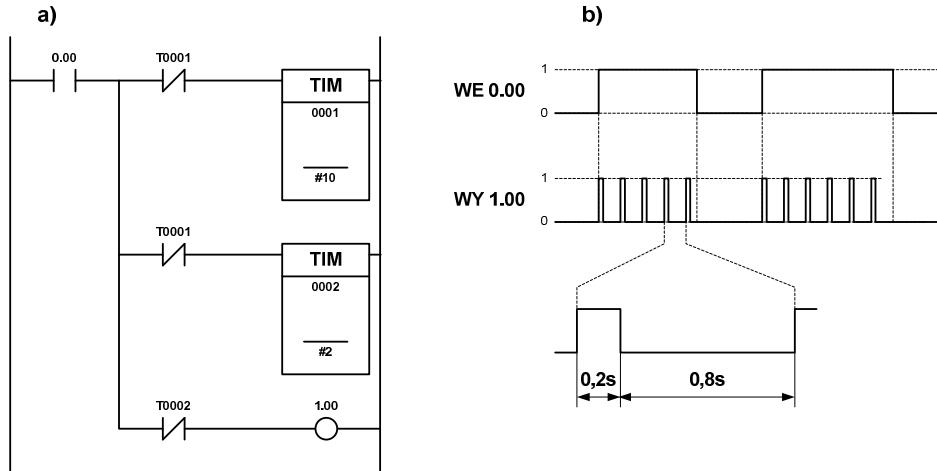
Tab.4.3. Flagi generatorów sygnałów prostokątnych

Flaga generatora	Przebieg czasowy	Częstotliwość
P_0_02s		f = 50 Hz
P_0_1s		f = 10 Hz
P_0_2s		f = 5 Hz
P_1s		f = 1 Hz
P_1min		f = 1/60 Hz

4.6.2. PROGRAMOWA REALIZACJA GENERATORA TAKTUJĄCEGO

W przypadku, gdy program użytkownika wymaga użycia generatora taktującego o nietypowych parametrach czasowych, wówczas niezbędna jest jego programowa realizacja. Generator taki można wykonać na wiele sposobów. Jedną z częściej stosowanych struktur programowych jest wykorzystanie dwóch czasomierzy TIM, które umożliwiają wytworzenie przebiegów prostokątnych praktycznie o dowolnej częstotliwości i dowolnym współczynniku wypełnienia. Na rysunku 4.19a przedstawiono przykładowy program, realizujący funkcję sterowanego generatora

sygnałów prostokątnych, o częstotliwości 1Hz i współczynniku wypełnienia 20%. W programie tym wykorzystano dwa czasomierze TIM, z których pierwszy (T1) pracuje w układzie generatora podstawy czasu (1Hz), a drugi (T2) odmierza czas trwania wysokiego stanu logicznego (200ms) na wyjściu 1.00. Generator ten posiada wejście sterujące 0.00, które musi być ustawione w stan wysoki, aby na wyjściu układu pojawił się sygnał prostokątny. Na rysunku 4.19b przedstawiono wykres czasowy obrazujący działanie generatora.

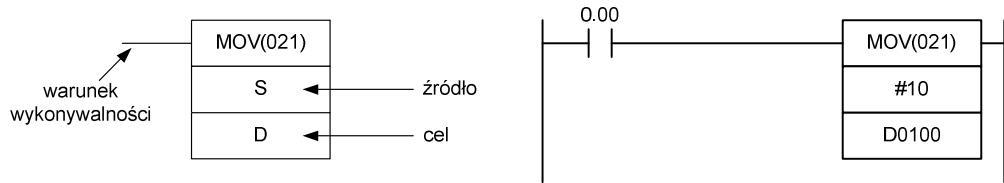


Rys.4.19. Przykład programowej realizacji generatora przebiegu prostokątnego (a) i wykres czasowy obrazujący jego działanie (b)

4.7. Operacje na danych

4.7.1. INSTRUKCJE TRANSFERU DANYCH

Podstawową instrukcją używaną do przenoszenia (kopiowania) danych jest instrukcja **MOV**. Instrukcja ta posiada jedno wejście zezwalające, na które wprowadza się logiczny warunek wykonywalności. Zastosowanie instrukcji MOV wymaga wprowadzenia dwóch operandów, z których pierwszy określa komórkę źródłową albo wartość liczbową, zaś drugi określa adres komórki docelowej, do której ma być zapisana dana. Na rysunku 4.20 przedstawiono sposób użycia instrukcji MOV, która w programie przykładowym wpisuje wartość #10 do komórki pamięci z obszaru D o adresie 100. W tym przypadku, warunkiem wykonywalności tej instrukcji jest wysoki stan logiczny na wejściu 0.00.

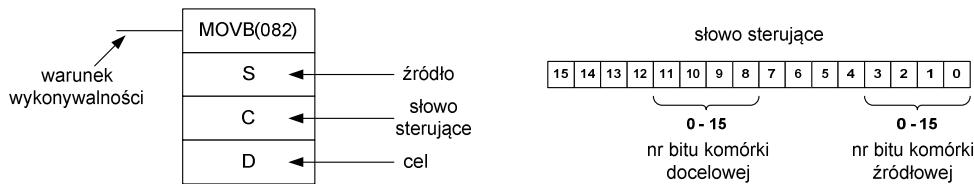


Rys.4.20. Przykład zastosowania instrukcji MOV do wprowadzania danych liczbowych

Instrukcja **MOV** występuje również w postaci różniczkowej, która polega na jednokrotnym wykonaniu tej instrukcji z chwilą, gdy warunek wykonywalności zmieni się z wartości 0 na wartość 1. Różniczkową postać instrukcji, która reaguje na zbocze narastające oznacza się dodatkowym prefiksem „@”, dodawanym przed mnemonikiem instrukcji (@MOV).

Instrukcja **MVN** (*Move Not*) jest odmianą instrukcji MOV, która pobiera daną z komórki źródłowej, dokonuje inwersji wszystkich jej bitów (negacja słowa) a następnie umieszcza ją w komórce docelowej. Instrukcja MVN podlega takim samym regułom wykonania, jak instrukcja MOV.

Instrukcja **MOVB** służy do skopiowania wartości jednego określonego bitu z komórki źródłowej do wybranego bitu komórki docelowej. Instrukcja ta posiada trzy operandy, które kolejno określają: adres komórki źródłowej, słowo sterujące i adres komórki docelowej. Słowo sterujące zawiera informacje, który bit z komórki źródłowej ma być skopiowany do którego bitu komórki docelowej. Na rysunku 4.21 przedstawiono sposób użycia instrukcji MOVB oraz znaczenie bitów słowa sterującego.

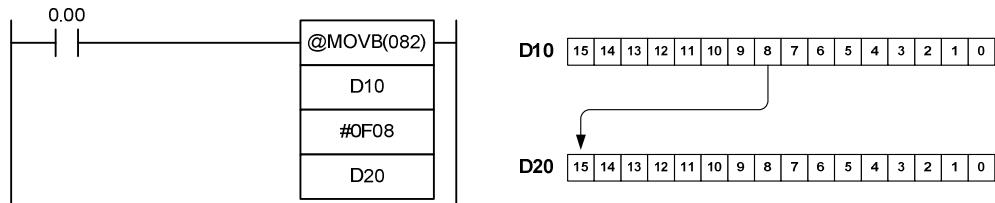


Rys.4.21. Sposób parametryzacji instrukcji MOVB

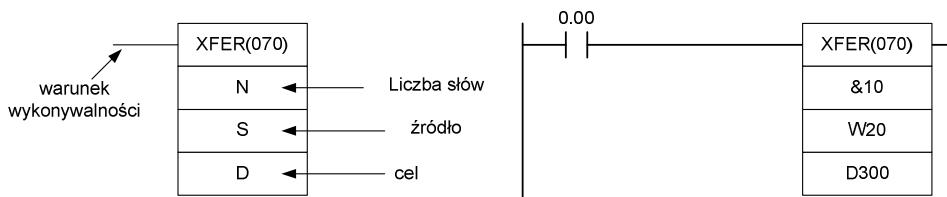
Na rysunku 4.22 przedstawiono przykładowy program wykorzystujący instrukcję MOVB. W programie tym, kopiwana jest wartość ósmego bitu z komórki pamięci D10 do piętnastego bitu komórki D20. Obecność prefiksu „@” przed mnemonikiem instrukcji oznacza, że instrukcja zostanie wykonana jednokrotnie, w chwili zmiany stanu bitu 0.00 z wartości 0 na wartość 1.

Instrukcja **XFER** służy do kopiowania bloków danych. Posiada 3 operandy i jedno wejście zezwolenia wykonania. Pierwszy operand określa liczbę kopiowanych

komórek, drugi wskazuje na adres początkowy źródłowego obszaru pamięci, natomiast trzeci operand określa adres docelowego obszaru pamięci. Na rysunku 4.23 przedstawiono sposób użycia instrukcji XFER, która w programie przykładowym kopiuje 10 kolejnych słów z obszaru pamięci W o adresie początkowym 20, do pamięci z obszaru D o adresie początkowym 300.

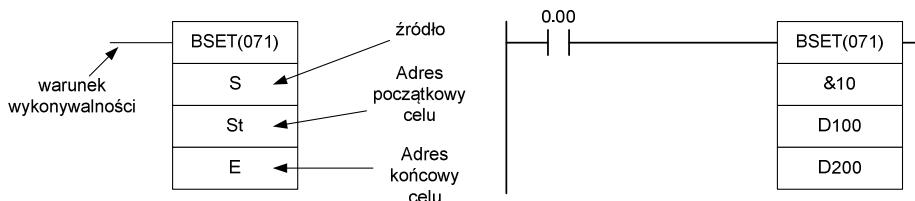


Rys.4.22. Przykładowy program wykorzystujący instrukcję MOVB



Rys.4.23. Przykład zastosowania instrukcji XFER do kopiowania bloków danych

Instrukcja **BSET** umożliwia wpisanie tej samej danej do kilku kolejnych komórek pamięci naraz. Instrukcja posiada 3 operandy i wejście zezwolenia. Pierwszy operand wskazuje na adres źródła kopiowanej danej lub określa kopowaną wartość liczbową. Pozostałe dwa operandy określają adres początkowy i końcowy docelowego obszaru pamięci. Instrukcja ta najczęściej jest stosowana do zerowania lub ustawiania wartości początkowej jednocześnie w wielu komórkach pamięci. Na rysunku 4.24 przedstawiono sposób użycia instrukcji BSET, która w programie przykładowym wpisuje wartość 10 do wszystkich komórek pamięci D, mieściących się w zakresie adresowym D100–D200.



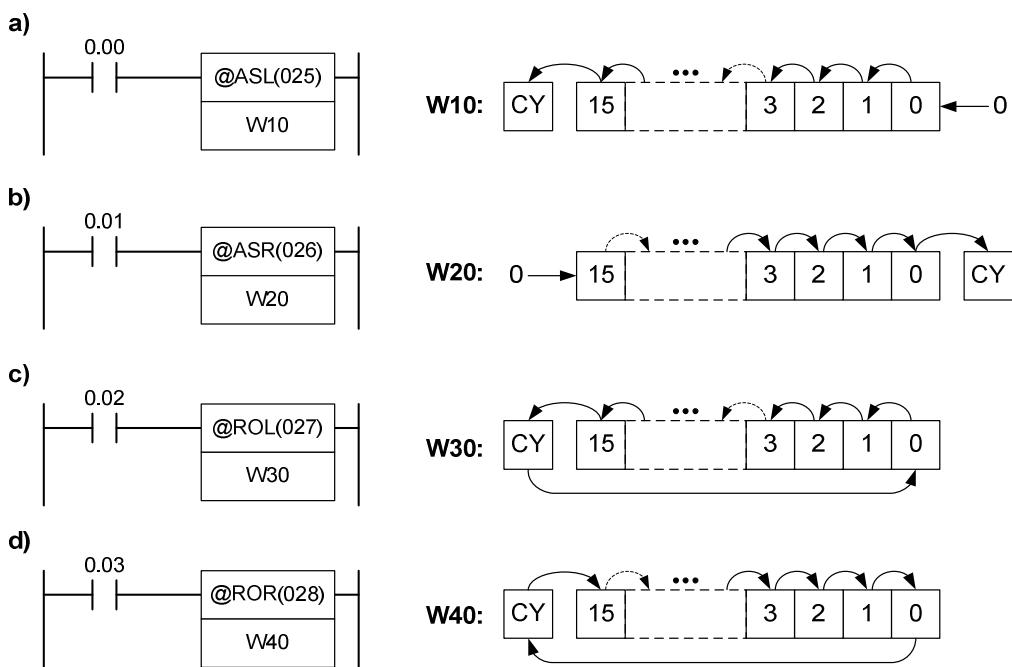
Rys.4.24. Przykład zastosowania instrukcji BSET do wpisywania stałej wartości do bloku danych

4.7.2. OPERACJE PRZESUWANIA BITÓW

Instrukcje **ASL** i **ASR** realizują funkcję arytmetycznego przesunięcia bitów w słowie, odpowiednio w lewo lub w prawo. Instrukcje te posiadają wejście wyzwalające i jeden operand, który wskazuje na adres przesuwanej danej. W wyniku wykonania operacji przesunięcia danej w lewo, najmłodszy jej bit otrzymuje wartość zerową, zaś wartość najstarszego bitu zostaje wpisana do znacznika CY. Analogicznie, po przesunięciu w prawo najstarszy bit słowa zostaje wyzerowany, a do znacznika CY zostaje wpisana wartość najmłodszego bitu.

Instrukcje **ROL** i **ROR** realizują funkcję rotacji – cyklicznego przesunięcia bitów. W wyniku wykonania operacji ROL, poszczególne bity słowa zostają przesunięte o jedną pozycję w lewo, przy czym w miejsce najmłodszego bitu zostaje podstawiona wartość znacznika CY, natomiast wartość najstarszego bitu zostaje wpisana do znacznika CY. Instrukcja ROR działa analogicznie, z tą różnicą, że przesuwa bity w prawo, również z uwzględnieniem flagi przeniesienia CY.

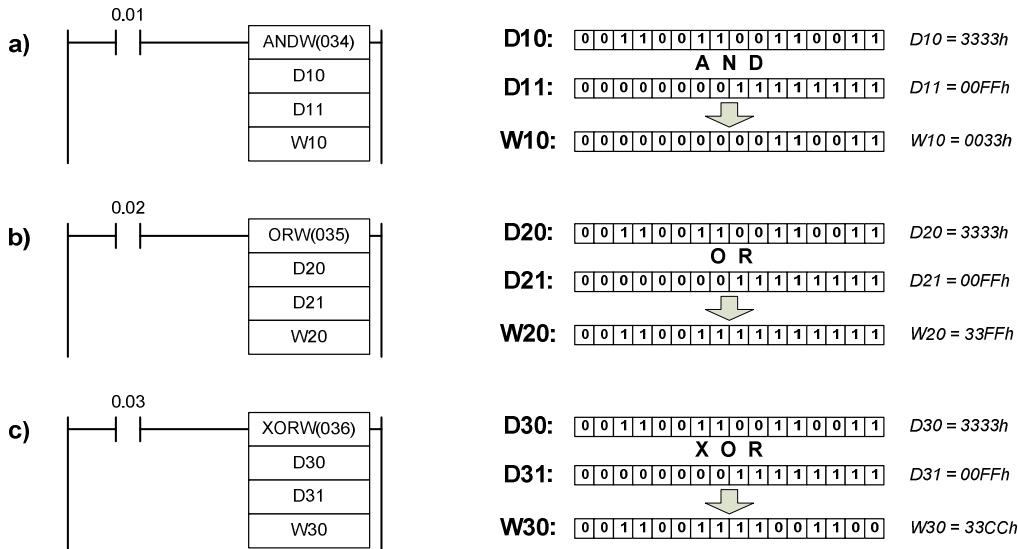
Na rysunku 4.25 przedstawiono zasadę działania poszczególnych instrukcji przesuwania bitów.



Rys.4.25. Zasada działania instrukcji ASL, ASR, ROL i ROR

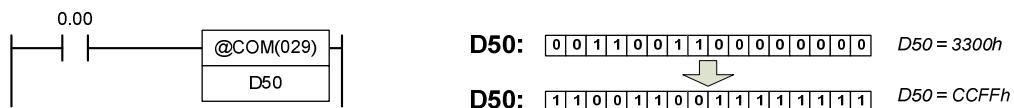
4.7.3. OPERACJE LOGICZNE

Instrukcja **ANDW** realizuje funkcję iloczynu logicznego (AND) na poszczególnych bitach dwóch argumentów wejściowych. Posiada jedno wejście zezwolenia oraz trzy operandy. Pierwsze dwa operandy zawierają dane wejściowe o szerokości jednego słowa. Trzeci operand wskazuje na adres komórki, w której umieszczony będzie wynik operacji logicznej. Identyczną strukturę posiadają instrukcje logiczne **ORW** i **XORW**, realizujące funkcje sumy logicznej (OR) oraz alternatywy rozłącznej (XOR), pomiędzy poszczególnymi bitami dwóch słów wyjściowych. Na rysunku 4.26 przedstawiono zasadę działania wymienionych funkcji logicznych na przykładowych wartościach liczbowych.



Rys.4.26. Zasada działania instrukcji logicznych ANDW, ORW i XORW.

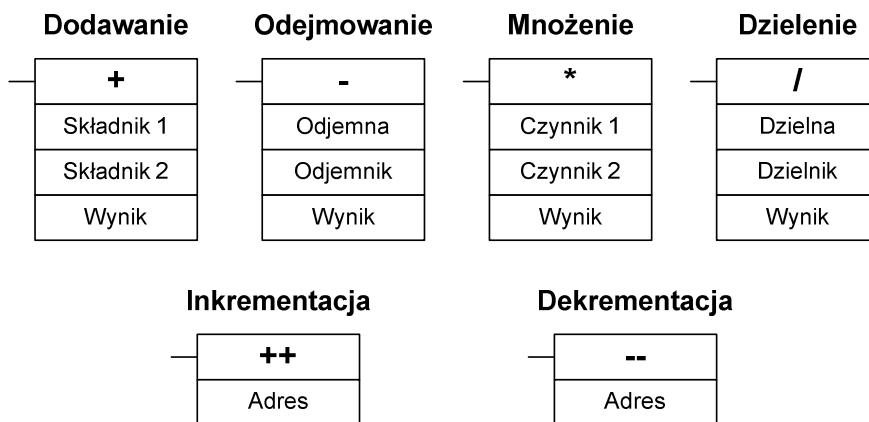
Instrukcja **COM** realizuje funkcję inwersji (negacji) wszystkich bitów argumentu wejściowego o długości 1 słowa. Na rysunku 4.27 przedstawiono zasadę działania tej instrukcji.



Rys.4.27. Zasada działania instrukcji COM.

4.7.4. OPERACJE ARYTMETYCZNE

W sterownikach programowalnych OMRON instrukcje arytmetyczne występują w postaci operatorów, które jednoznacznie określają wykonywaną operację. Instrukcje te mogą wykonywać działania na liczbach wprowadzonych w formacie binarnym lub BCD. Na rysunku 4.28 przedstawiono ogólną postać podstawowych instrukcji arytmetycznych.



Rys.4.28. Ogólna postać podstawowych instrukcji arytmetycznych.

Mnemonik instrukcji arytmetycznej, oprócz operatora, może posiadać dodatkowe oznaczenia literowe, które specyfikują typ danych i szczegółowo określają działanie instrukcji. Wśród tych oznaczeń literowych można wyróżnić:

- **C** – instrukcja wykonuje operację arytmetyczną z uwzględnieniem bitu przeniesienia CY, np. +C, -C;
- **B** – obliczenia wykonywane są na liczbach w formacie BCD, np. +B, *B, ++B;
- **L** – operandy mają długość podwójnego słowa – 32 bity, np. ++L, -L, /L;
- **U** – operacje dzielenia i mnożenia wykonywane są na danych bez znaku (*Unsigned*), np. *U, /U;
- **F** – operacje na liczbach zmiennoprzecinkowych 32-bitowych (*Float*), np. +F, *F, /F;
- **D** – operacje na liczbach zmiennoprzecinkowych 64-bitowych (*Double*), np. +D, /D.

W tabeli 4.4 przedstawiono symbole oraz opis działania wybranych instrukcji arytmetycznych.

Tab.4.4. Wybrane instrukcje arytmetyczne

Symbol instrukcji	Opis działania
+	Dodawanie liczb całkowitych ze znakiem bez uwzględnienia flagi CY
+C	Dodawanie liczb całkowitych ze znakiem z uwzględnieniem flagi CY
+L	Dodawanie liczb całkowitych ze znakiem podwójnej długości, bez uwzględnienia flagi CY
+CL	Dodawanie liczb całkowitych ze znakiem podwójnej długości, z uwzględnieniem flagi CY
+F	Dodawanie liczb zmiennoprzecinkowych o długości 32 bitów
+B	Dodawanie liczb całkowitych w formacie BCD bez uwzględnienia flagi CY
-	Odejmowanie liczb całkowitych ze znakiem bez uwzględnienia flagi CY
-F	Odejmowanie liczb zmiennoprzecinkowych
-C	Odejmowanie liczb całkowitych ze znakiem z uwzględnieniem flagi CY
*	Mnożenie liczb całkowitych ze znakiem
*U	Mnożenie liczb całkowitych bez znaku
*BL	Mnożenie liczb całkowitych w formacie BCD o podwójnej długości
/	Dzielenie liczb całkowitych ze znakiem
/D	Dzielenie liczb zmiennoprzecinkowych o podwójnej długości
/B	Dzielenie liczb całkowitych w formacie BCD
++	Inkrementacja liczby binarnej
++B	Inkrementacja liczby w formacie BCD
--	Dekrementacja liczby binarnej
--L	Dekrementacja liczby binarnej o podwójnej długości

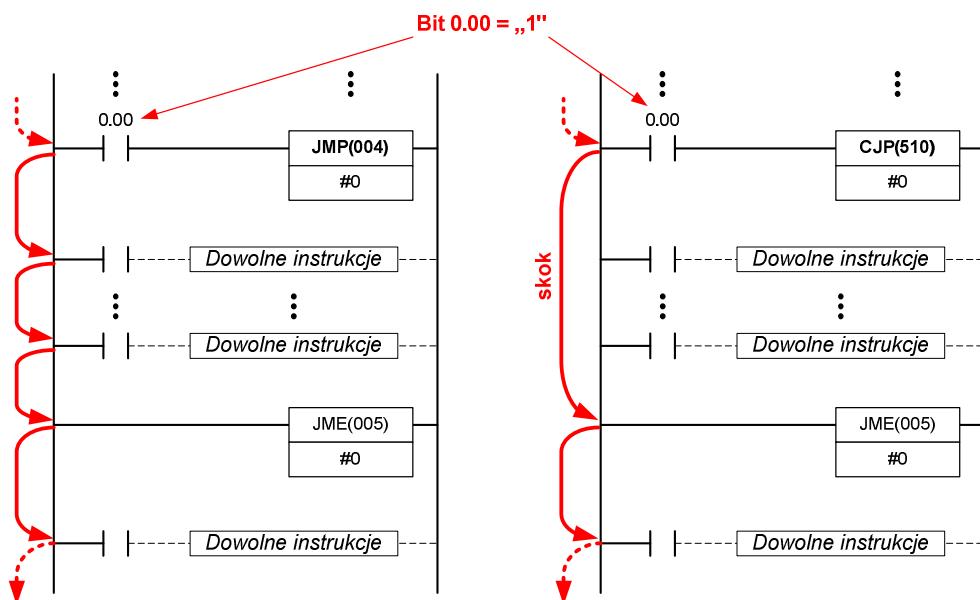
4.8. Instrukcje sterujące przebiegiem wykonania programu

4.8.1. INSTRUKCJE SKOKÓW

Instrukcja **JMP** występuje w parze zawsze z instrukcją **JME** i realizuje funkcję skoku bezwarunkowego do miejsca w programie, w którym jest umieszczona skojarzona z nią instrukcja JME. Obie instrukcje posiadają jeden operand, na który wprowadza się numer instrukcji skoku (dla obu ten sam) w zakresie 0–1023. Dodatkowo, instrukcja JMP posiada jedno wejście zezwolenia, które steruje wykonaniem skoku. W tym miejscu należy zaznaczyć, że działanie to jest nieco mylące, gdyż instrukcja JMP wykona skok jeśli warunek wykonywalności **nie będzie spełniony**. W przeciwnym wypadku, jeśli na wejściu sterującym pojawi się wysoki stan logiczny, instrukcja skoku będzie ignorowana i program wykona po kolei wszystkie instrukcje występujące po instrukcji JMP.

Instrukcja **CJP** realizuje funkcję skoku warunkowego i posiada identyczną postać wywołania jak instrukcja JMP. Różnica w działaniu tych instrukcji polega na tym, że w instrukcji CJP skok zostanie wykonany, jeżeli warunek wykonywalności będzie spełniony.

Na rysunku 4.29 przedstawiono sposób użycia oraz zobrazowano zasadę działania instrukcji JMP i CJP na programach przykładowych.



Rys.4.29. Zasada działania instrukcji JMP i CJP.

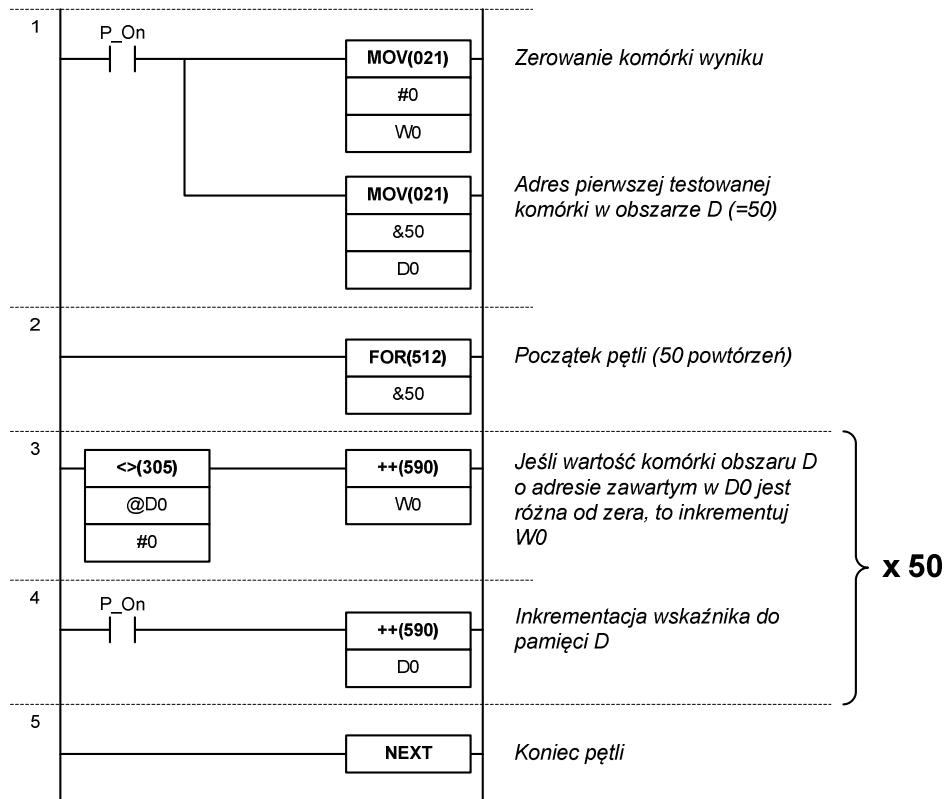
Jeżeli bit zezwolenia 0.00 będzie znajdował się w stanie wysokim, to w programie z lewej strony skok nie zostanie wykonany, w związku z tym wszystkie instrukcje występujące pomiędzy instrukcjami JMP i JME zostaną kolejno wykonane. Z kolei w programie znajdującym się po prawej stronie, skok warunkowy wywoływanego instrukcją CJP zostanie wykonany, a instrukcje znajdujące się pomiędzy instrukcjami CJP i JME zostaną ominięte.

4.8.2. PĘTLA FOR-NEXT

Instrukcja **FOR** w połączeniu z instrukcją **NEXT** umożliwia realizację pętli programowych, które powtarzają wykonanie fragmentu programu zadeklarowaną liczbę razy pod rząd. Instrukcja FOR posiada jeden operand wejściowy, na który wprowadza się wartość liczbową z przedziału 0–65535, określającą liczbę powtórzeń pętli. Należy zauważyć, że instrukcja FOR jest wykonywana bezwarunkowo, co oznacza, że nie posiada ona wejścia zezwolenia i jest pierwszą a zarazem ostatnią instrukcją w linii programu.

Instrukcja **NEXT** jest bezparametrową instrukcją zamkującą powtarzany fragment programu. W przypadku, gdy występuje konieczność wcześniejszego zakończenia wykonywania pętli, można zastosować instrukcję **BREAK**, która spowoduje, że zamiast instrukcji następujących po niej zostaną wykonane instrukcje puste (NOP) i pętla zostanie zakończona.

Na rysunku 4.30 przedstawiono przykładowy program, który sprawdza ile komórek pamięci z obszaru D o adresach od 50 do 99 posiada niezerową wartość. W pierwszej linii programu następuje wyzerowanie komórki W0, która będzie zawierała wynik działania. W tej samej linii zostaje wpisana wartość początkowa 50 do komórki D0, będącej wskaźnikiem do przeszukiwanego obszaru pamięci (D50–D99). W drugiej linii programu znajduje się instrukcja FOR, która definiuje liczbę iteracji (50 cykli) i jednocześnie stanowi początek powtarzanego fragmentu programu. W trzeciej linii programu znajduje się instrukcja komparatora ($<>$), która sprawdza czy wartość komórki o adresie wskazywanym przez D0 jest różna od zera. Jeśli tak, wówczas zostaje wykonana instrukcja inkrementacji (++) , która powiększa wartość komórki W0, stanowiącej końcowy wynik działania programu. Czwarta linia programu inkrementuje wartość wskaźnika D0, tak aby w kolejnej iteracji program sprawdził następną komórkę z przeszukiwanego obszaru. Instrukcja NEXT powoduje powtórne wykonanie trzeciej i czwartej linii programu, i tak do momentu, aż zostanie wyzerowany licznik powtórzeń. Po wykonaniu pięćdziesięciu iteracji, program przechodzi do realizacji kolejnych linii programu, a w komórce W0 znajduje się wartość, która określa liczbę niezerowych komórek pamięci obszaru D w zakresie adresowym 50–99.



Rys.4.30. Przykładowy program wykorzystujący strukturę iteracyjną FOR-NEXT.

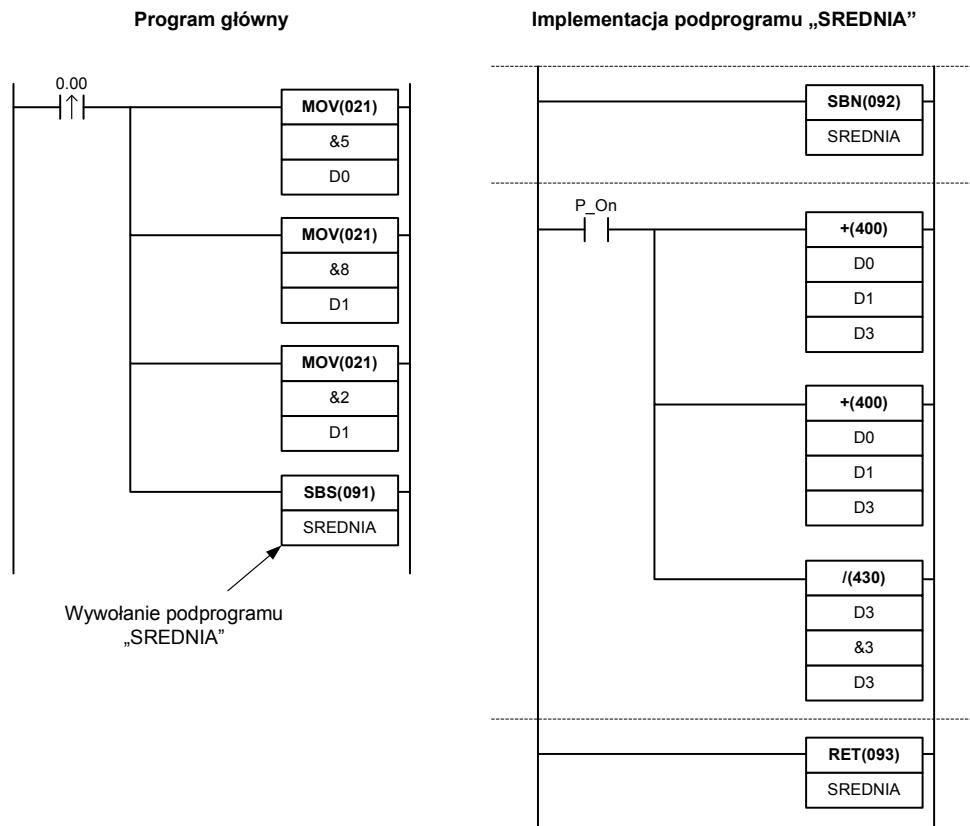
4.8.3. REALIZACJA PODPROGRAMÓW

Podprogramy są wydzielonymi fragmentami programu głównego, które zawierają zestaw instrukcji realizujących pewną funkcję, która może być wywoływana wielokrotnie, w obrębie działania programu głównego. W sterownikach OMRON serii CJ istnieje możliwość definiowania tzw. podprogramów globalnych, które są dostępne dla wszystkich programów, należących do jednego projektu.

Definicja podprogramu rozpoczyna się bezwarunkową instrukcją **SBN**, która posiada jeden operand wejściowy, oznaczający numer podprogramu (0–1023). W kolejnych liniach umieszcza się instrukcje realizujące zamierzoną funkcję podprogramu. Obszar podprogramu musi kończyć się bezargumentową instrukcją **RET**, która oznacza powrót do punktu, z którego wywołany był podprogram. W celu zachowania przejrzystości w programie głównym, implementacja podprogramów może być zrealizowana w osobnych sekcjach.

Wywołanie podprogramu polega na wykonaniu instrukcji **SBS**, której parametrem jest numer podprogramu. Po wykonaniu wszystkich instrukcji zawartych w podprogramie, następuje powrót do programu głównego i realizacja kolejnych jego instrukcji.

Na rysunku 4.31 przedstawiono przykładowy program, który wywołuje podprogram o nazwie *SREDNIA*. Podprogram ten oblicza wartość średnią z trzech komórek: D0, D1 i D2 i umieszcza wynik w komórce D3. Nazwa podprogramu *SREDNIA* została zdefiniowana jako symbol lokalny typu liczbowego (*Number*), do którego przypisano wartość 0 (numer podprogramu). Używanie nazw symbolicznych w podprogramach, znacznie ułatwia ich identyfikację i poprawia czytelność całego programu.



Rys.4.31. Przykładowy program wywołujący podprogram „SREDNIA” i jego implementacja.

5. Programowanie paneli operatorskich serii NS za pomocą oprogramowania CX-Designer

5.1. Zasady adresowania urządzeń w laboratorium

W Laboratorium Sterowania Urządzeniami i Napędami Przemysłowymi obowiązuje spójna zasada adresowania poszczególnych urządzeń, w ramach jednej podsieci Ethernet, według następujących zależności:

Konfiguracja adresów IP w laboratorium:

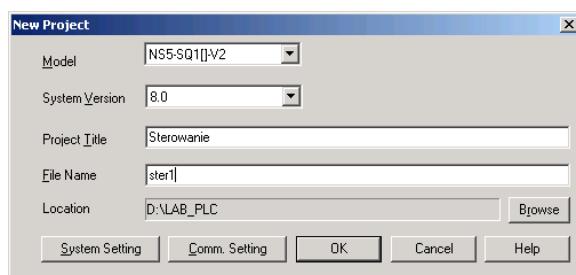
Komputer:	192.168.115.x	gdzie x:	(111÷116)
Sterownik:	192.168.115.y	gdzie y:	(1÷6)
Panel OP:	192.168.115.z	gdzie z:	(11÷16)

Konfiguracja węzłów sieci Ethernet (*Node Number*) dla sterownika i panelu operatorskiego powinna być identyczna z ostatnim oktetem ich adresu IP, wg poniższego przykładu:

Dla sterownika o adresie IP: **192.168.115.4** *Node Number* powinien być ustawiony na wartość **4** (również na nastawniku obrotowym znajdującym się na obudowie sterownika). Z kolei panel operatorski na tym stanowisku powinien posiadać adres IP: **192.168.115.14**, zaś jego *Node Number* należy ustawić na wartość **14**.

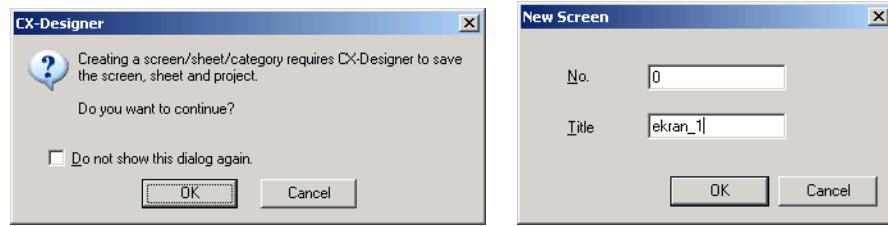
5.2. CX-Designer – konfiguracja i programowanie

Po uruchomieniu programu CX-Designer należy założyć nowy projekt, podając jego nazwę i tytuł (rys.5.1).



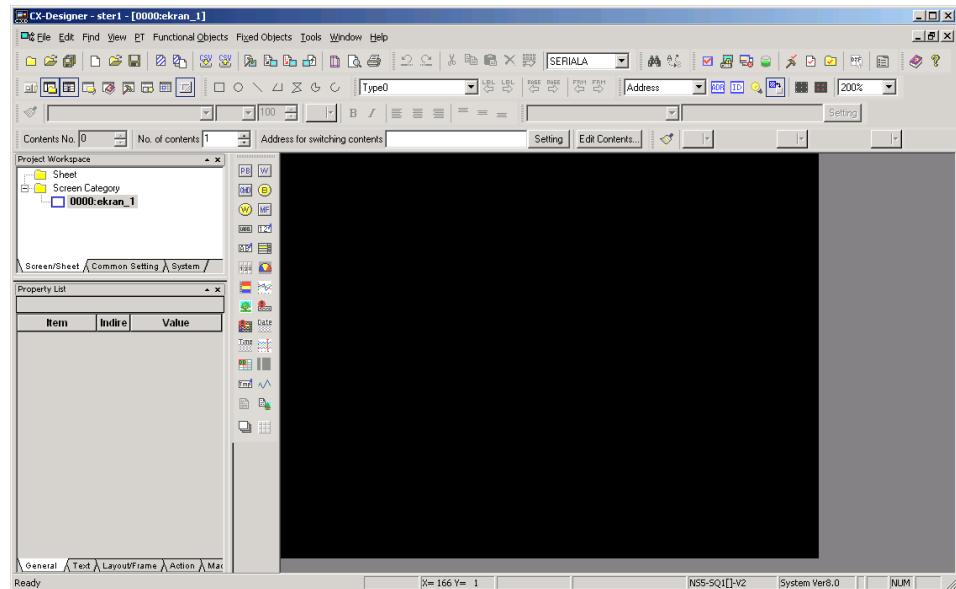
Rys.5.1. Założenie nowego projektu

Po chwili program przypomni, że należy utworzyć i zapisać ekran startowy (Rys.5.2).



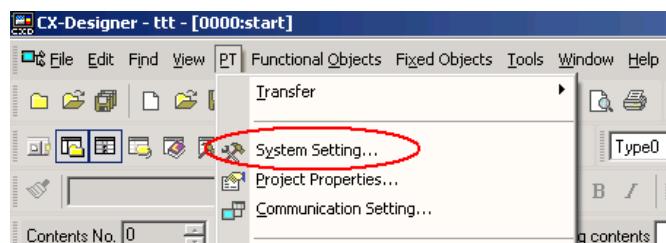
Rys.5.2. Utworzenie nowego ekranu wizualizacji

Na rysunku 5.3 przedstawiono wygląd głównego okna programu CX-Designer.



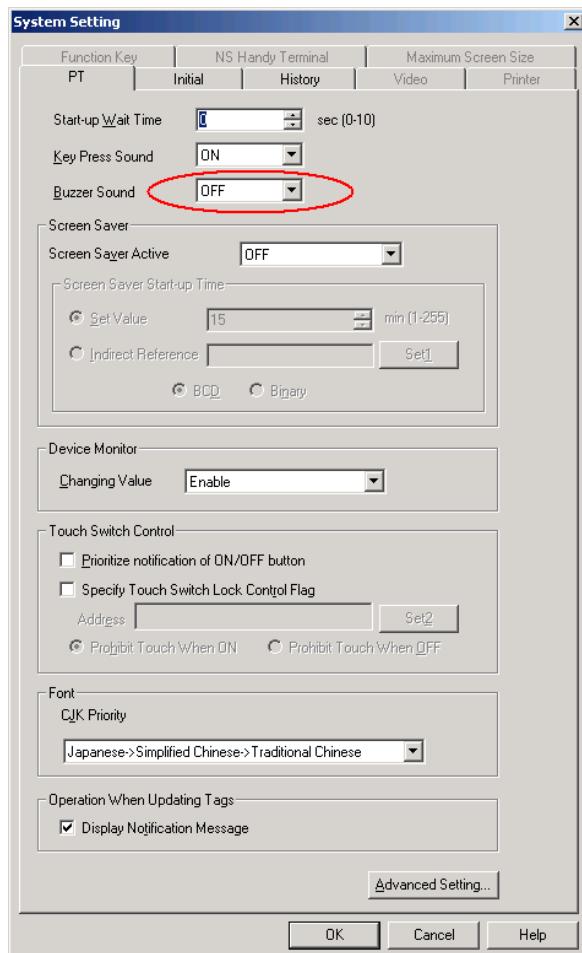
Rys.5.3. Główne okno programu CX-Designer

W następnej kolejności należy wprowadzić ustawienia systemowe, korzystając z menu jak na rysunku 5.4.



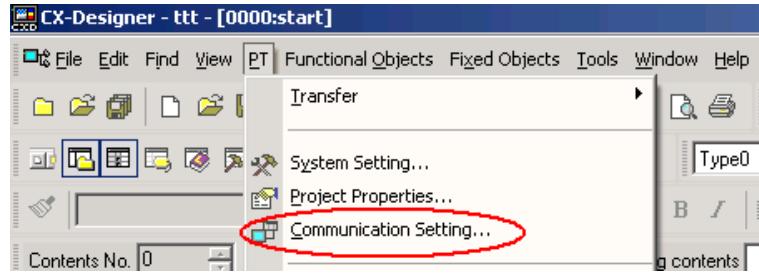
Rys.5.4. Wybór ustawień systemowych panelu operatorskiego

W oknie jak na rysunku 5.5 zaleca się wyłączyć dźwięki ostrzeżeń panelu (Buzzer Sound ustawić na OFF) – zapobiegnie to generowaniu bardzo głośnych dźwięków podczas wyświetlania komunikatów o błędach i ostrzeżeniach. Można również wyłączyć dźwięki klawiszy, występujące przy wszystkich operacjach na panelu dotykowym (Key Press Sound).



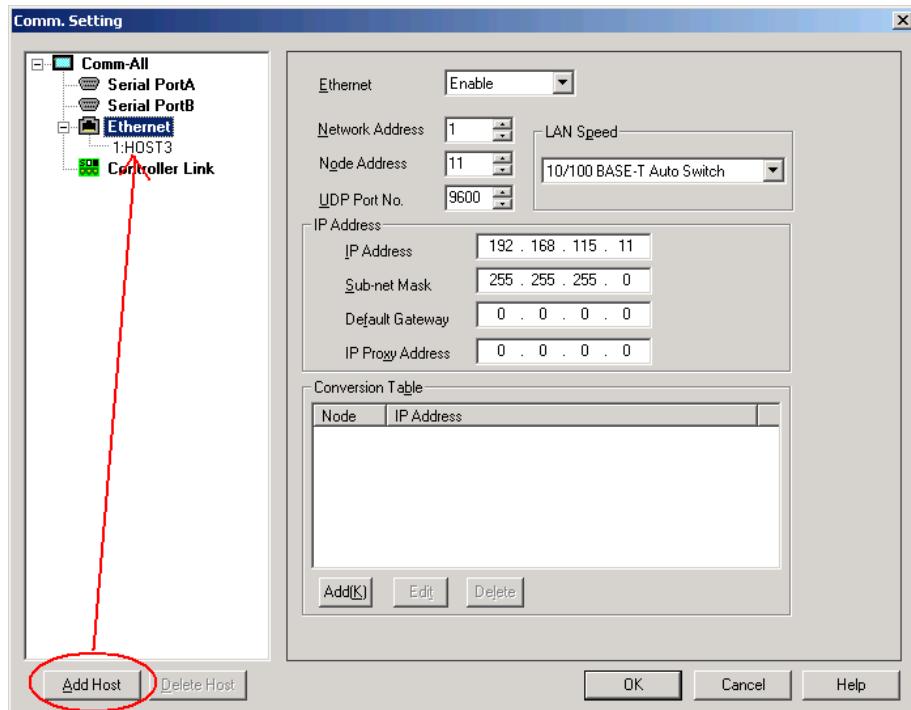
Rys.5.5. Ustawienia systemowe panelu operatorskiego

W dalszej części należy skonfigurować parametry połączeń komunikacyjnych pomiędzy komputerem, panelem operatorskim i sterownikiem PLC. W tym celu należy wybrać opcję „Communication Settings”, korzystając z menu wg rysunku 5.6.



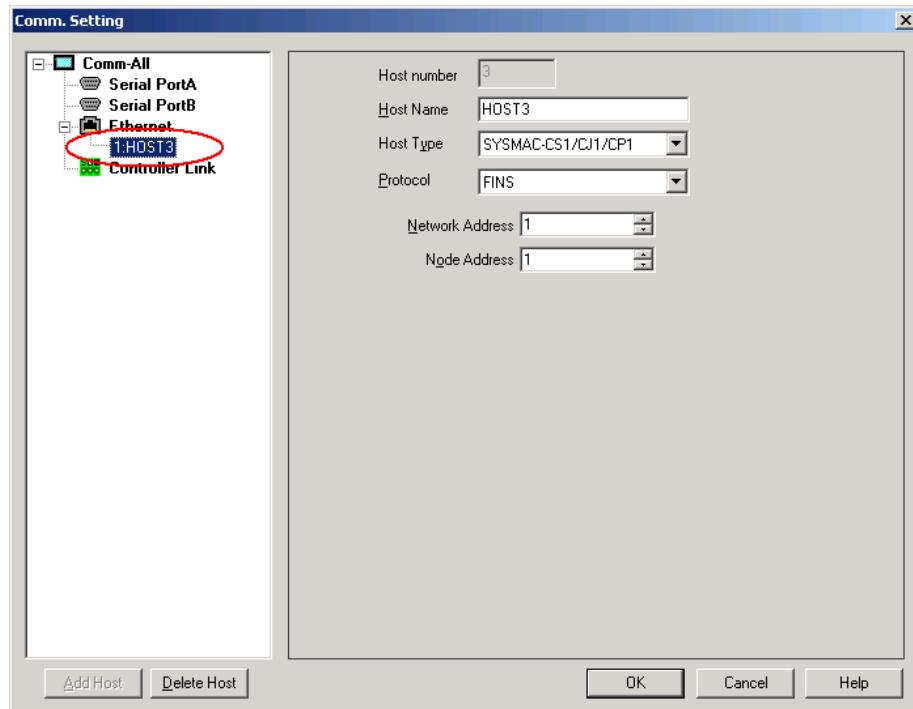
Rys.5.6. Wybór menu „Communication Settings”

W oknie, jak na rysunku 5.7, należy usunąć domyślne połączenie szeregowe „Serial PortA” a następnie aktywować połączenie ethernetowe (Ethernet Enable) i skonfigurować parametry tego połączenia. Przedstawione na rysunku parametry połączenia są przykładowe i dotyczą stanowiska laboratoryjnego ze sterownikiem PLC o adresie IP: 192.168.115.1. W następnej kolejności należy dodać kanał komunikacyjny pomiędzy panelem operatorskim a sterownikiem PLC wybierając opcję „Add Host”.



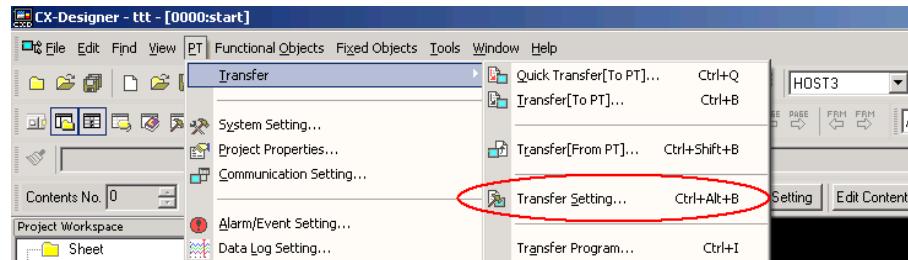
Rys.5.7. Ustawienie połączeń komunikacyjnych

Przykładowe ustawienia komunikacji pomiędzy panelem operatorskim a sterownikiem o nazwie HOST3 o adresie IP: 192.168.115.1 przedstawiono na rysunku 5.8.



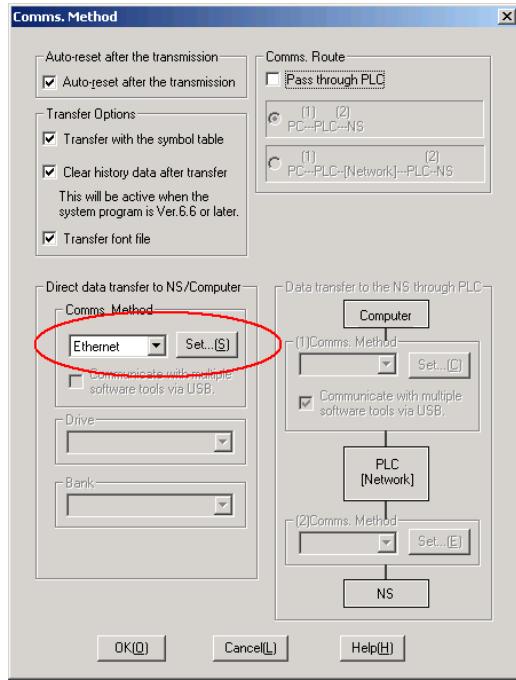
Rys.5.8. Ustawienia komunikacji dla sterownika „HOST3”.

Kolejny etap konfiguracji to ustawienia parametrów transferu danych pomiędzy komputerem PC a sterownikiem PLC – rysunek 5.9.



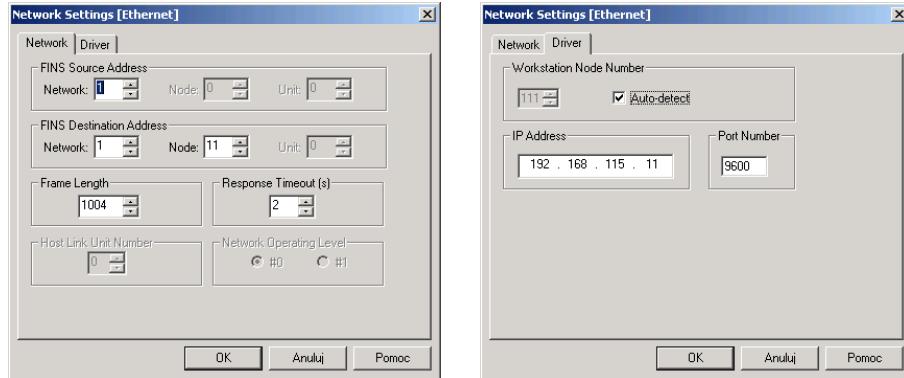
Rys.5.9. Otwarcie okna „Transfer Setting”

W tym celu, w oknie jak na rysunku 5.10, należy ustawić metodę komunikacji „Ethernet” i wprowadzić ustawienia szczegółowe połączenia, naciskając przycisk „Set...”.



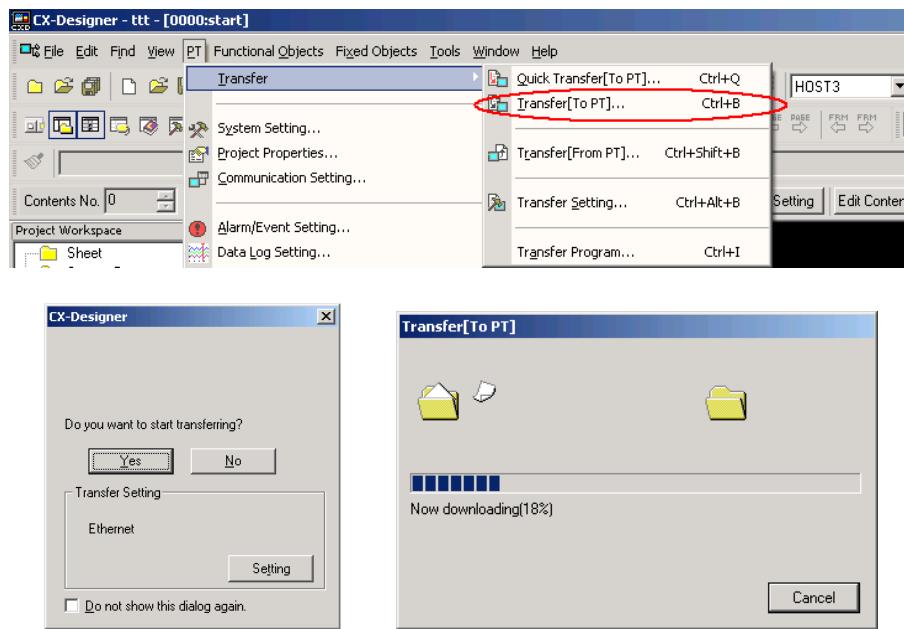
Rys.5.10. Wybór metody komunikacji pomiędzy komputerem a sterownikiem

Następnie należy skonfigurować parametry połączenia sieci Ethernet, zgodnie z ustawieniami przykładowymi, widocznymi na rysunku 5.11.



Rys.5.11. Przykładowa konfiguracja sieci Ethernet dla połączenia pomiędzy komputerem a sterownikiem PLC.

Po skonfigurowaniu parametrów komunikacyjnych można przystąpić do edycji projektu użytkownika. Na zakończenie pracy, cały projekt wraz z jego ustawieniami należy przesyłać do pamięci panelu operatorskiego, korzystając z polecenia „Transfer[To PT]”, zgodnie z rysunkiem 5.12.



Rys.12. Transfer projektu i jego ustawień do panelu operatorskiego.

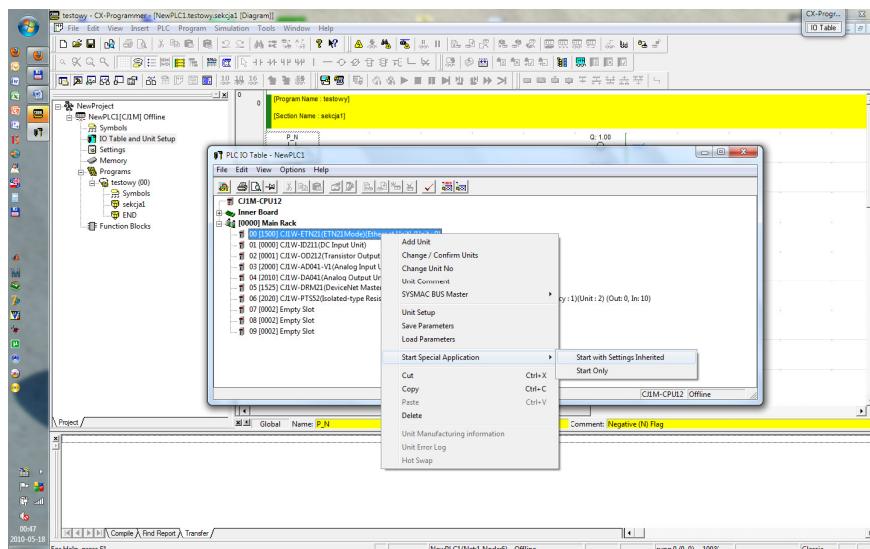
6. Konfiguracja sieci przemysłowych w pakiecie CX-ONE

6.1. Konfiguracja sieci PROFIBUS

6.1.1. PRZYGOTOWANIE TABELI ROUTINGU.

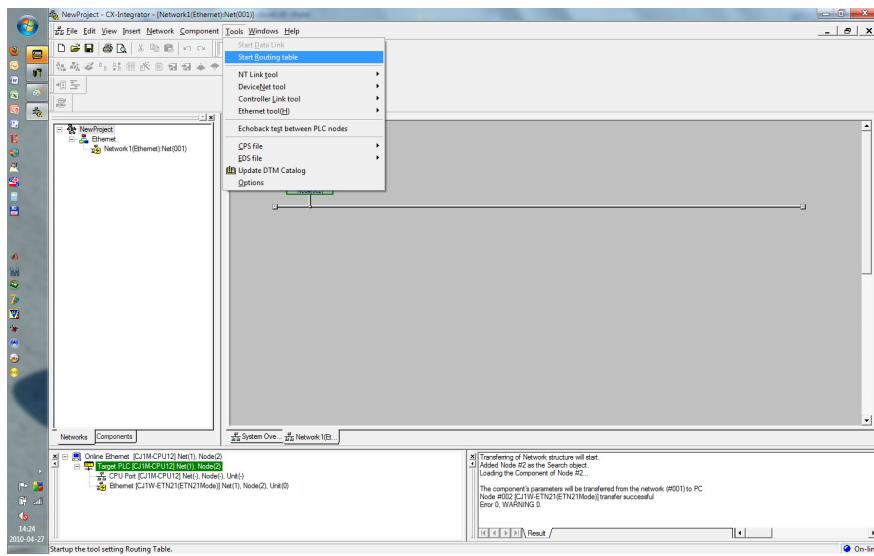
W celu poprawnej pracy modułu Profibus, należy odpowiednio przygotować tabelę Routingu. W tym celu należy wykonać następujące czynności:

W oprogramowaniu Cx-Programmer otworzyć do edycji tabelę wejść/wyjść. Następnie na opisie modułu komunikacyjnego Ethernet ETN21 należy kliknąć prawym klawiszem myszy i wybrać opcję Start Special Application/Start With Settings Inherited, tak jak to pokazano na rys.6.1.



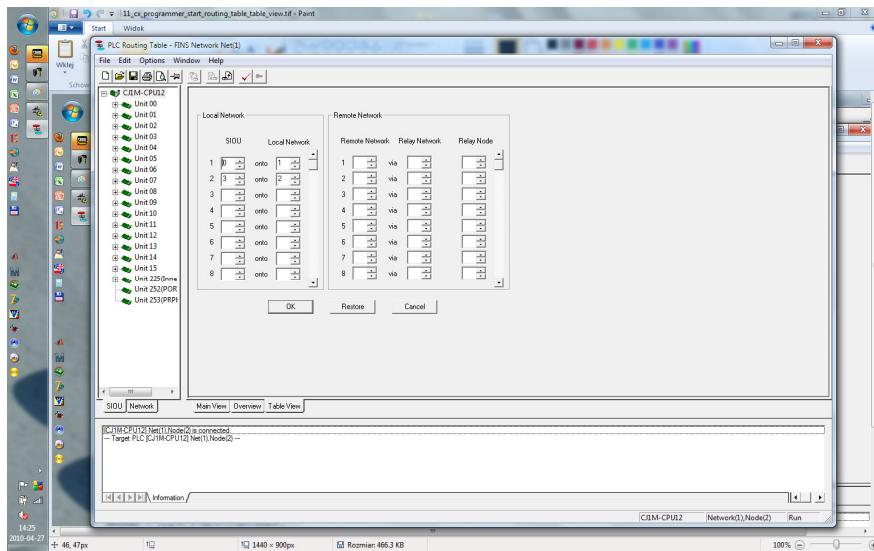
Rys.6.1. Uruchamianie programu Cx-Integrator

Zostanie uruchomiony program Cx-Integrator. Następnie w menu głównym należy wybrać opcję Tools/Start Routing Table (rys.6.2.).



Rys.6.2. Rozpoczynanie edycji tabeli routingu

Zostanie wyświetlone okno jak na rys. 6.3.



Rys.6.3. Widok tabeli routingu

W zakładce Table View należy uzupełnić wpisy w zależności od ustawień sprzętowych modułów komunikacyjnych zgodnie z następującymi zasadami:

- SIOU : numer urządzenia komunikacyjnego ustawiony na panelu czołowym tego urządzenia za pomocą przełącznika oznaczonego Unit Number,
- Local Network : numer lokalnej sieci komunikacyjnej, kolejno od 1 do n.

W przykładzie widocznym na rysunku powyżej przyjęto następującą konfigurację sieci:

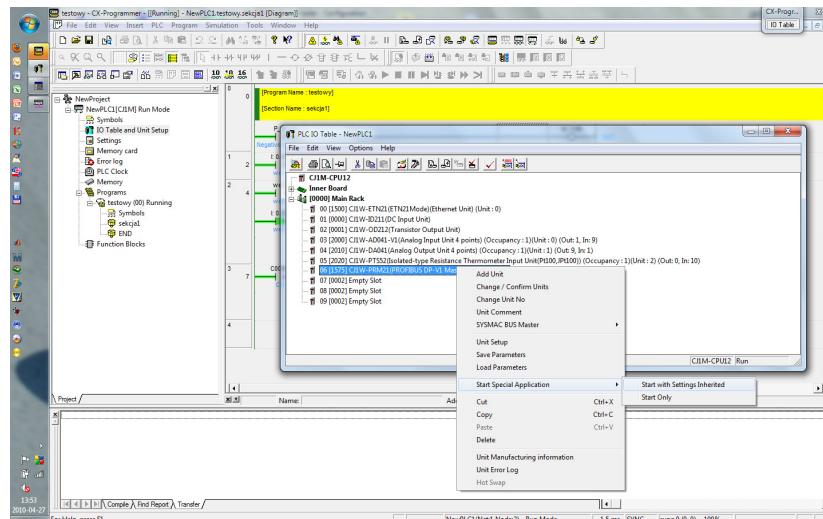
- | | |
|----------------|-------------------|
| sieć Ethernet: | SIOU – 0 |
| | Local Network – 1 |
| Sieć Profibus: | SIOU – 1 |
| | Local Network – 2 |

Po ustaleniu konfiguracji sieci należy ją zatwierdzić klikając na przycisk OK, a następnie wysłać do sterownika PLC przez użycie menu głównego Options/Transfer To PLC. Sterownik musi być w stanie Online.

Po wykonaniu tych czynności można zapisać tablicę routingu na dysku oraz zamknąć program Cx-Integrator.

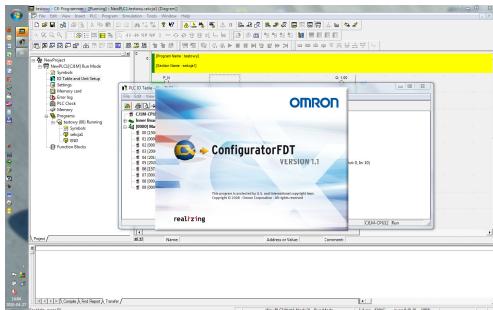
6.1.2. KONFIGURACJA MODUŁÓW KOMUNIKACYJNYCH PROFIBUS

Do konfiguracji Profibus służy program Cx-ConfiguratorFDT, który uruchamia się w sposób następujący: w programie Cx-Programmer należy otworzyć do edycji tabelę wejść/wyjść. Następnie na opisie modułu Profibus należy kliknąć prawym klawiszem myszy i wybrać opcję Start Special Application/Start With Settings Inherited, jak to pokazano na rys.6.4.

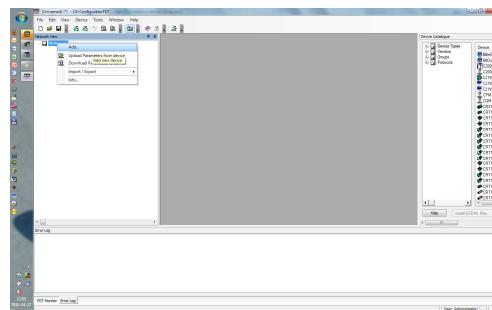


Rys.6.4. Tabela wejść/wyjść z zaznaczonym modułem komunikacyjnym Profibus

Następuje uruchomienie programu Cx-ConfiguratorFDT, którego ekran startowy pokazano na rys. 6.5. Po uruchomieniu programu należy kliknąć prawym klawiszem myszy na nazwie sieci My Network i wybrać opcję ADD, jak pokazano na rys. 6.6.

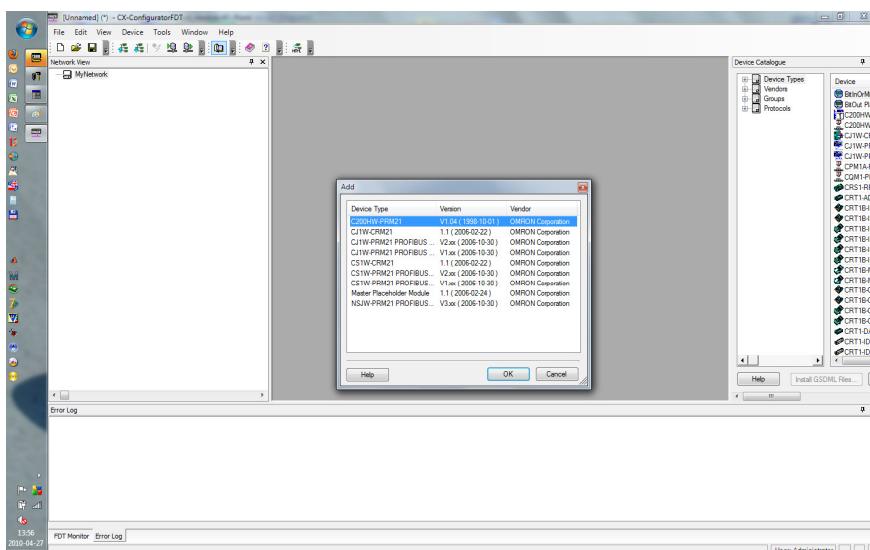


Rys. 6.5. Start programu Cx-ConfiguratorFDT

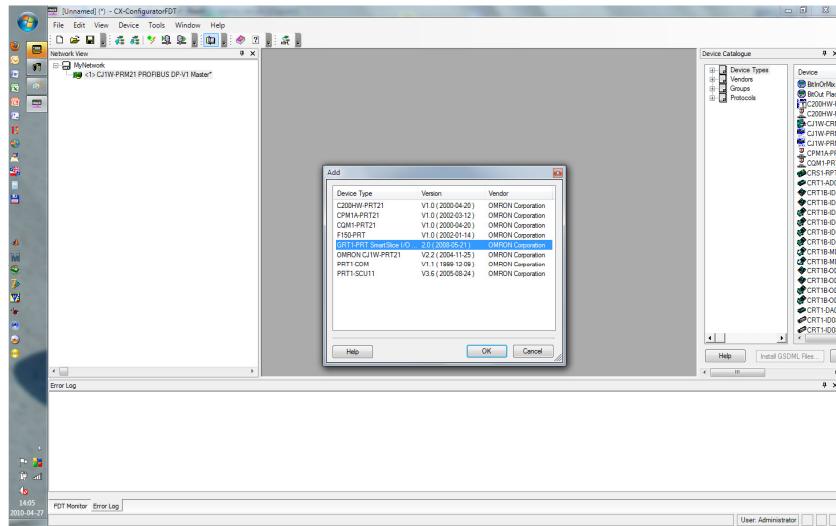


Rys. 6.6. Dodawanie modułów sieciowych

Otworzy się okno pokazane na rys. 6.7. Z dostępnych na liście modułów komunikacyjnych należy wybrać moduł, który jest aktualnie używany (zamontowany na magistrali sterownika PLC). W przypadku laboratorium jest to moduł o oznaczeniu **CJ1W-PRM21 Profibus DP-V1 Master ver 2.x**. Następnie klikając prawym klawiszem myszy na wybranym module komunikacyjnym należy dodać posiadany moduł we/wyjścia rozproszonych, tak pokazano na rys. 6.8. W konfiguracji laboratorium jest to moduł **GRT1-PRT Smart Slice I/O Coupler**.



Rys.6.7. Dodawanie modułu master sieci Profibus



Rys. 6.8. Dodawanie modułu we/wy rozproszonych

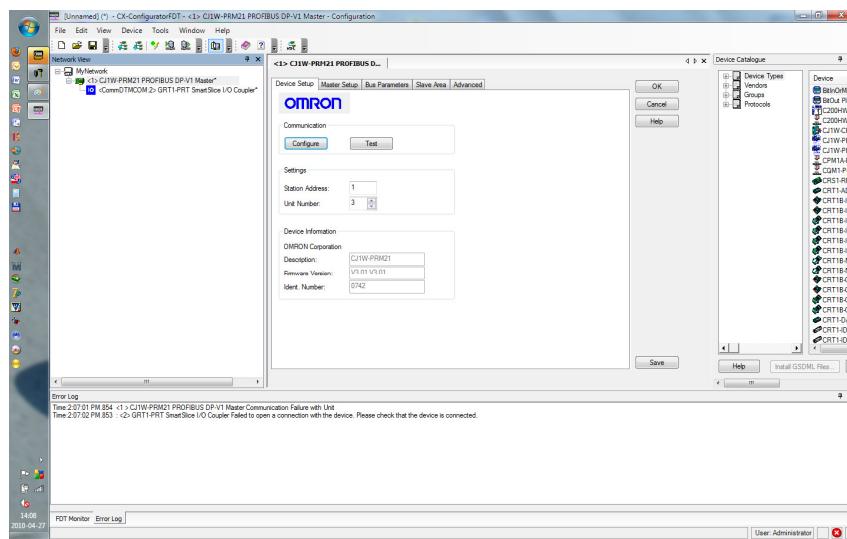
Następnie klikając dwukrotnie lewym klawiszem myszy na nazwie modułu komunikacyjnego Profibus Master **CJ1W-PRM21** należy otworzyć kartę z właściwościami modułu komunikacyjnego, pokazaną na rys. 6.9. W zakładce Device Setup należy ustawić parametry komunikacji przez kliknięcie na przycisk Communication/Configure. Parametry komunikacji muszą być zgodne z ustawionymi w programie Cx-Programmer: należy wybrać odpowiedni interfejs połączenia i jego parametry, takie jak adres sterownika, prędkość transmisji itp. Następnie należy uzupełnić pola związane bezpośrednio z modułem master:

- Station Address: zgodny z adresem stacji,
- Unit number: zgodny z numerem ustawionym za pomocą mikroprzełącznika na panelu czołowym modułu.

Wpisane parametry należy koniecznie zatwierdzić przyciskiem OK., w przeciwnym razie nie zostaną zapamiętane i nie będzie możliwe ustanowienie prawidłowej komunikacji między sterownikiem PLC a modułem Profibus Master CJ1W-PRM21.

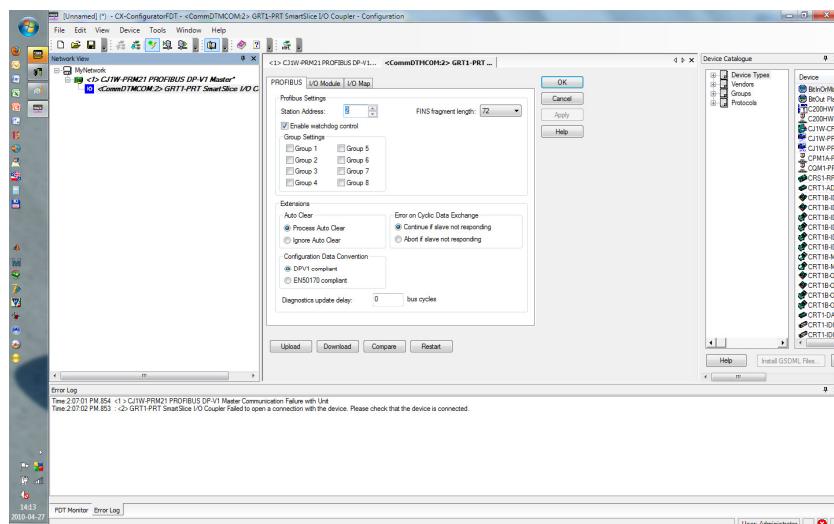
Po ustawieniu wymienionych parametrów należy kliknąć lewym klawiszem myszy na przycisk Test. Jeśli wszystkie parametry zostały prawidłowo ustawione, w polach Description, Firmware Version oraz Ident. Number pojawią się informacje związane z posiadany modułem. Jeśli to nie nastąpi, należy sprawdzić wpisane parametry i przeprowadzić ponowny test komunikacji. Tylko pozytywny wynik testu świadczy o prawidłowej komunikacji modułu Profibus Master CJ1W-PRM21 ze sterownikiem.

Zawartość pozostałych zakładek karty Właściwości opisana jest szczegółowo w dokumentacji technicznej modułu i nie jest przedmiotem niniejszego opracowania.



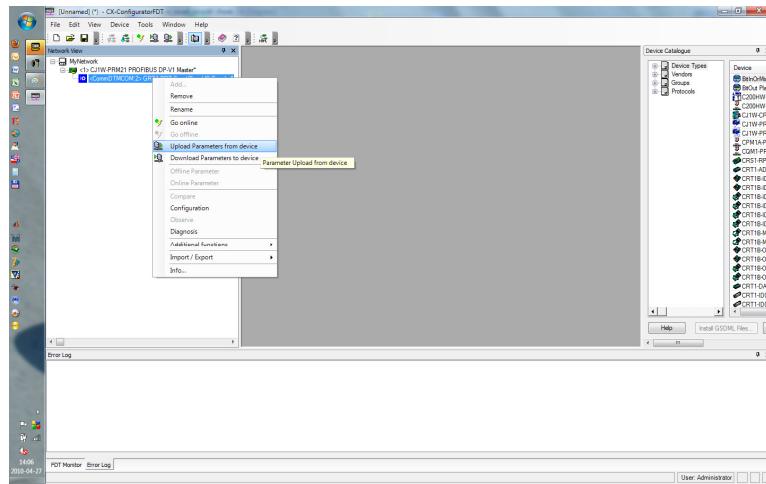
Rys.6.9. Karta właściwości modułu komunikacyjnego Profibus Master

Następnie należy ustawić adres komunikacyjny zdalnego modułu wejść wyjść rozproszonych Profibus. W tym celu należy dwa razy kliknąć lewym klawiszem myszy na nazwie modułu zdalnego GRT1-PRT i w karcie właściwości modułu należy wybrać zakładkę Profibus, tak jak pokazano na rys. 6.10. W polu Station Address należy wpisać adres modułu taki sam, jak ustawiony mikroprzełącznikiem na panelu czołowym zdalnego modułu.



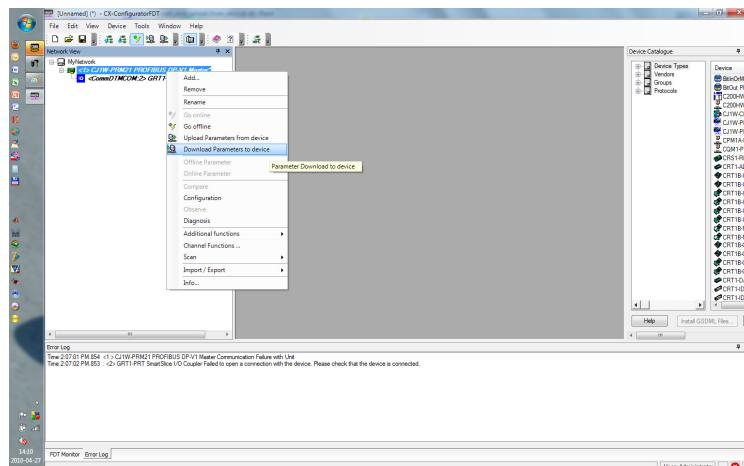
Rys. 6.10. Karta właściwości zdalnego modułu wejść/wyjść rozproszonych Profibus

Po wpisaniu parametrów komunikacji należy odczytać konfigurację sprzętową zdalnego modułu wejścia/wyjścia rozproszonych Profibus. W tym celu należy kliknąć prawym klawiszem myszy na nazwie zdalnego modułu Profibus GRT1-PRT i z menu kontekstowego wybrać opcję Upload Parameters from Device (jak na rys. 6.11).



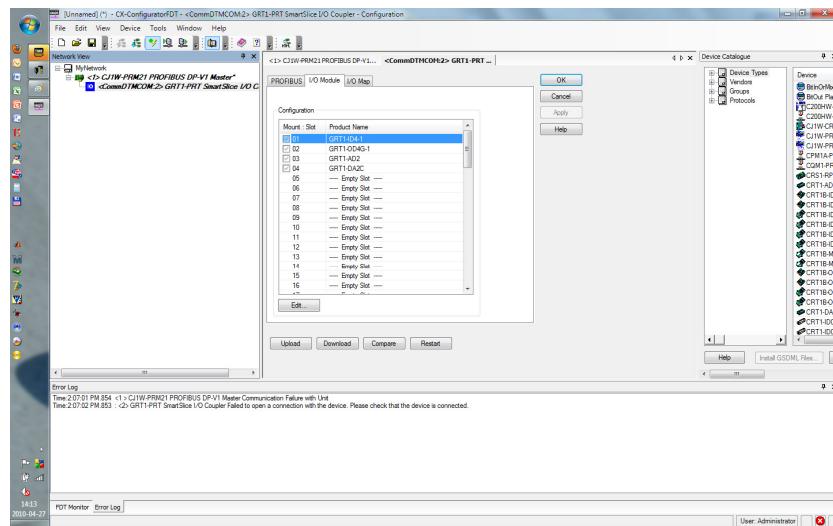
Rys.6.11. Odczytywanie konfiguracji sprzętowej zdalnego modułu Profibus

Po odczytaniu konfiguracji sprzętowej zdalnego modułu Profibus, należy ostateczną konfigurację sieci Profibus ponownie wysłać do modułu komunikacyjnego Profibus Master **CJ1W-PRM21** klikając na nazwie modułu prawym klawiszem myszy i wybierając opcję Download Parameters to Device, jak pokazano to na rys. 6.12. Jest to ostatni etap konfiguracji sieci Profibus.



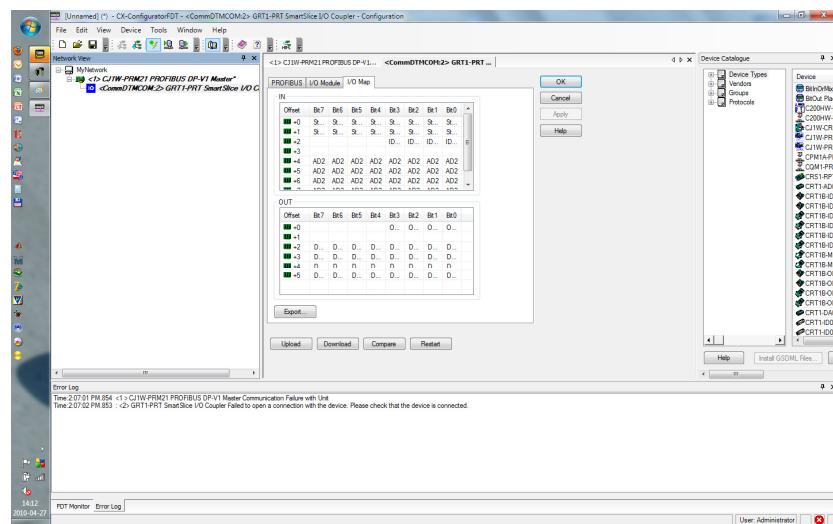
Rys.6.12. Przesyłanie konfiguracji do modułu Profibus Master

Poprawność konfiguracji należy sprawdzić klikając dwa razy na nazwie modułu zdalnego GRT1-PRT. Pojawi się karta właściwości modułu. Po wybraniu karty I/O module wyświetlna zostanie sprzętowa konfiguracja zdalnego modułu wejść/wyjść Profibus pokazana na rys.6.13.



Rys.6.13. Karta z listą zainstalowanych zdalnych modułów wejścia/wyjścia sieci Profibus

Położenie poszczególnych bitów odpowiadających określonym wejściom lub wyjściom zdalnego modułu Profibus GRT1-PRT należy odczytać w zakładce I/O Map pokazanej na rys. 6.14.



Rys.6.14. Zakładka I/O Map karty właściwości zdalnego modułu Profibus

Parametr „Offset” widoczny w polach opisujących wejścia (IN) i wyjścia (OUT) oznacza przesunięcie danego bajtu względem adresu bazowego. W związku z tym, konfiguracja zdalnych wejść i wyjść pokazana na rys.6.14 jest następująca:

Wejścia cyfrowe (IN):

Adres bazowy modułu wejść +0 : bity statusowe bit7-bit0

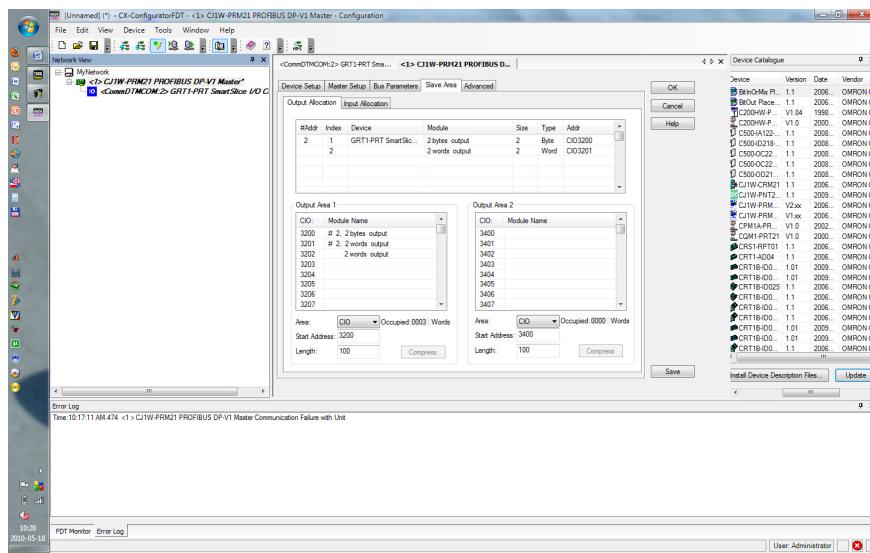
Adres bazowy modułu wejść +1 : bity statusowe bit7-bit0

Adres bazowy modułu wejść +2 : bity danych IN3-IN0

Wyjścia cyfrowe (OUT):

Adres bazowy modułu wyjść +0 : bity danych OUT3-OUT0

Adresy bazowe poszczególnych modułów wejścia/wyjścia umieszczone są w zakładce konfiguracyjnej modułu Profibus Master **CJ1W-PRM21**. Aby odczytać adresy, należy kliknąć dwa razy lewym klawiszem myszy na nazwie modułu komunikacyjnego Profibus Master **CJ1W-PRM21**. Zostanie wyświetlona okno właściwości modułu, w którym należy wybrać zakładkę Slave Area jak to pokazano na rys.6.15.



Rys.6.15. Mapa pamięci rozproszonych modułów wejścia/wyjścia w sieci Profibus.

Następnie należy odczytać adresy odpowiednich modułów wejścia/wyjścia wybierając zakładki Output Allocation lub Input Allocation. Na rys.6.15 widoczne są adresy wyjść cyfrowych (aktywna zakładka Output Allocation).

W rozpatrywanym przykładzie adresy bazowe są następujące:

- dla wejść adres bazowy 3300,
- dla wyjść adres bazowy 3200.

Należy zwrócić uwagę, że poszczególne moduły adresowane są za pomocą 16-bitowych słów, w związku z tym adresy poszczególnych bitów wejścia/wyjścia są takie, jak w tabeli 6.1.

Tab.6.1. Adresy poszczególnych bitów wejść/wyjść

Wejścia		Wyjścia	
Numer wejścia	Adres	Numer wyjścia	Adres
IN0	3301.0	OUT0	3200.0
IN1	3301.1	OUT1	3200.1
IN2	3301.2	OUT2	3200.2
IN3	3301.3	OUT3	3200.3

Po wykonanej konfiguracji można zapisać konfigurację sieci na dysku komputera oraz zamknąć program konfiguracyjny sieci Profibus Cx-ConfiguratorFDT.

6.1.3. PRZYKŁAD STEROWANIA ZDALNYMI MODUŁAMI WEJŚĆ/WYJŚĆ

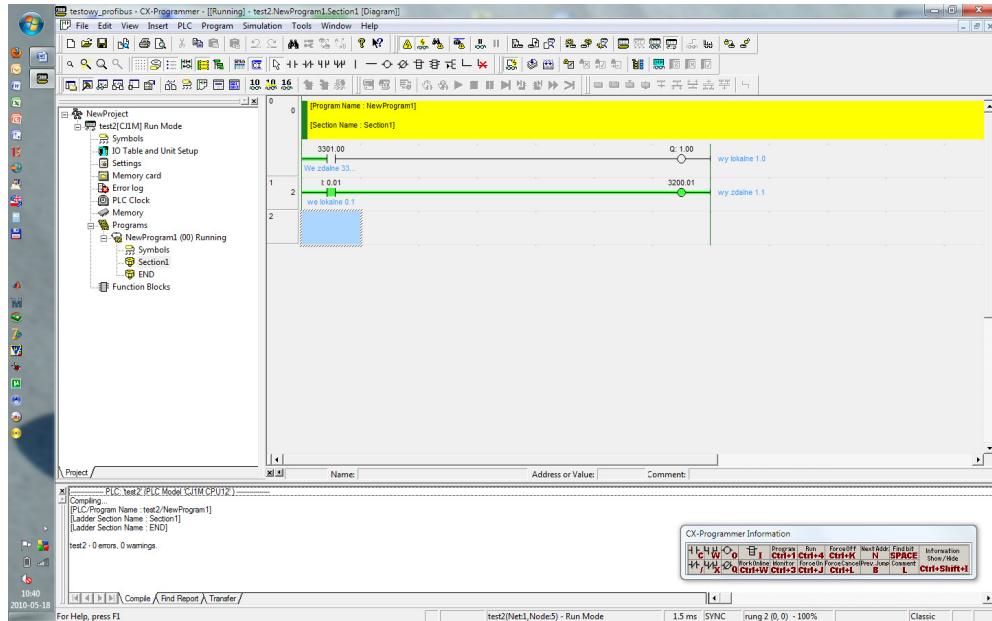
Poniżej pokazano sposób odczytu i zapisu informacji z wykorzystaniem rozproszonych wejść/wyjść.

W rozpatrywanym przykładzie odczytywany jest stan bitu na zdalnym wejściu I0.0 i przekazywany na wyjście sterownika O1.0. Następnie stan wejścia lokalnego sterownika I0.1 przepisywany jest na zdalne wyjście O1.1.

Zgodnie z zasadami opisanymi w poprzednich podrozdziałach, należy odnaleźć adresy poszczególnych zdalnych wejść i wyjść. W opisywanym przykładzie są to odpowiednio:

- adres zdalnego wejścia I0.0: 3301.0,
- adres zdalnego wyjścia O1.1: 3200.1.

Na rys. 6.16 pokazano program realizujący opisane zadanie.



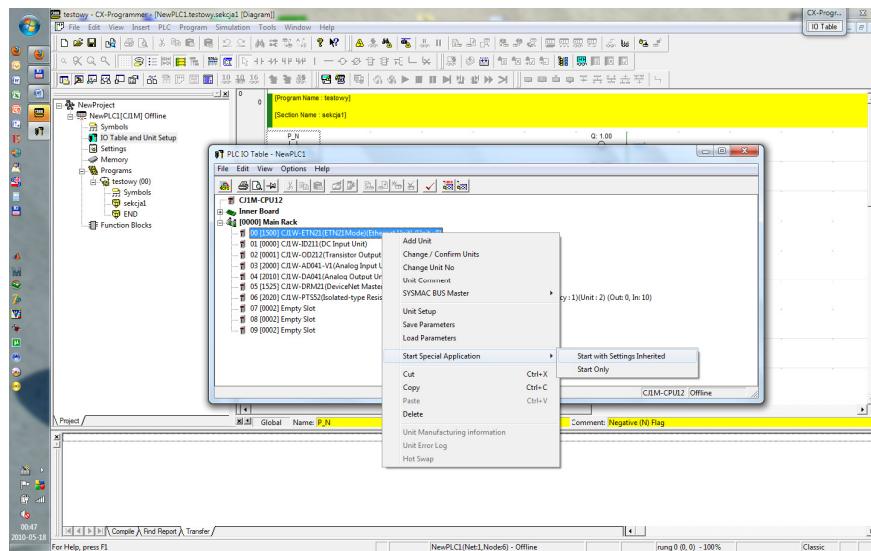
Rys.6.16. Przykładowy program sterujący zdalnymi wejściami/wyjściami

6.2. Konfiguracja sieci DeviceNet w pakiecie Cx-One.

6.2.1. KONFIGURACJA TABELI ROUTINGU

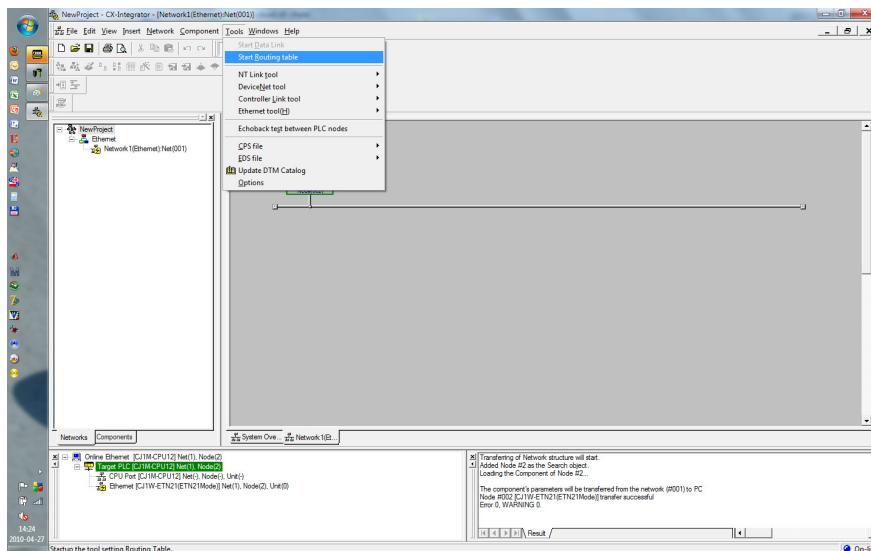
Do prawidłowej pracy modułu komunikacyjnego DeviceNet wymagana jest prawidłowa konfiguracja tabeli routingu. W tym celu należy wykonać następujące czynności:

W oprogramowaniu Cx-Programmer otworzyć do edycji tabelę wejść/wyjść. Następnie na opisie modułu komunikacyjnego Ethernet ETN21 należy kliknąć prawym klawiszem myszy i wybrać opcję Start Special Application/Start With Settings Inherited, tak jak to pokazano na rys.6.17.



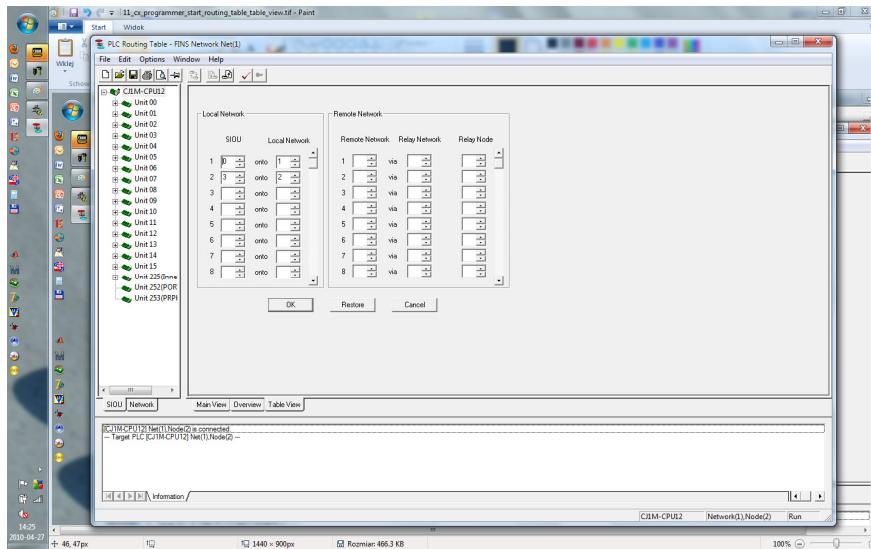
Rys.6.17. Uruchamianie programu Cx-Integrator

Zostanie uruchomiony program Cx-Integrator. Następnie w menu głównym należy wybrać opcję Tools/Start Routing Table (rys.6.18).



Rys.6.18. Rozpoczynanie edycji tabeli routingu

Zostanie wyświetcone okno jak na rys. 6.19.



Rys.6.19. Widok tabeli routingu

W zakładce Table View należy uzupełnić wpisy w zależności od ustawień sprzętowych modułów komunikacyjnych zgodnie z następującymi zasadami:

- SIOU : numer urządzenia komunikacyjnego ustalony na panelu czołowym tego urządzenia za pomocą przełącznika oznaczonego Unit Number,
- Local Network : numer lokalnej sieci komunikacyjnej, kolejno od 1 do n.

W przykładzie widocznym na rysunku powyżej przyjęto następującą konfigurację sieci:

- sieć Ethernet: SIOU – 0
 Local Network – 1
- Sieć Profibus: SIOU – 1
 Local Network – 2

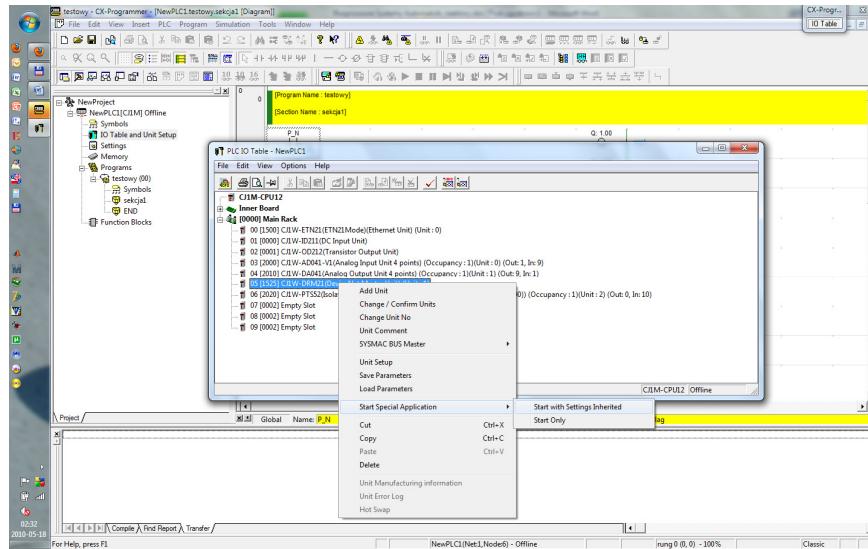
Po ustaleniu konfiguracji sieci należy ją zatwierdzić klikając na przycisk OK, a następnie wysłać do sterownika PLC przez użycie menu głównego Options/Transfer To PLC. Sterownik musi być w stanie Online.

Po wykonaniu tych czynności można zapisać tablicę routingu na dysku oraz zamknąć program Cx-Integrator.

6.2.2. KONFIGURACJA MODUŁU KOMUNIKACYJNEGO DEVICENET

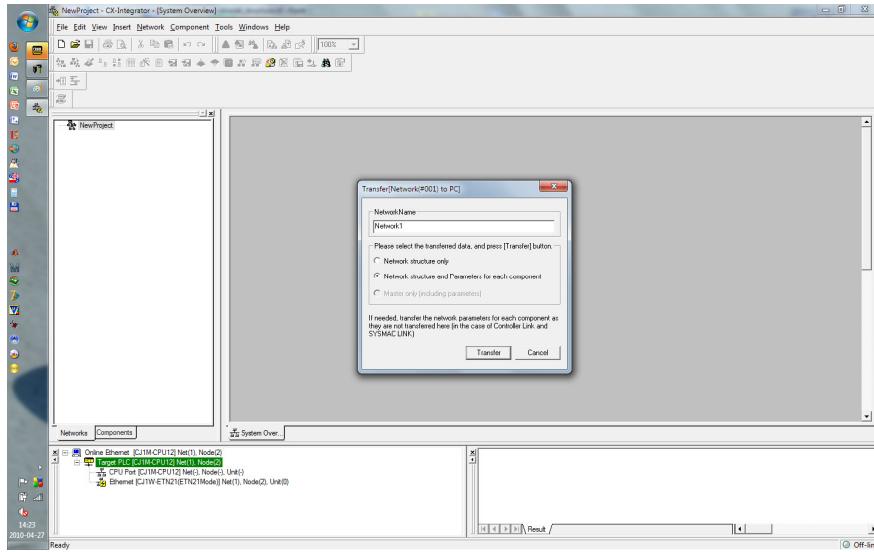
Do konfiguracji modułów i sieci DeviceNet służy program Cx-Integrator, który uruchamia się w sposób następujący:

W programie Cx-Programmer należy otworzyć do edycji tabelę wejścia/wyjścia. Następnie na opisie modułu DeviceNet należy kliknąć prawym klawiszem myszy i wybrać opcję Start Special Application/Start With Settings Inherited, jak to pokazano na rys.6.20.



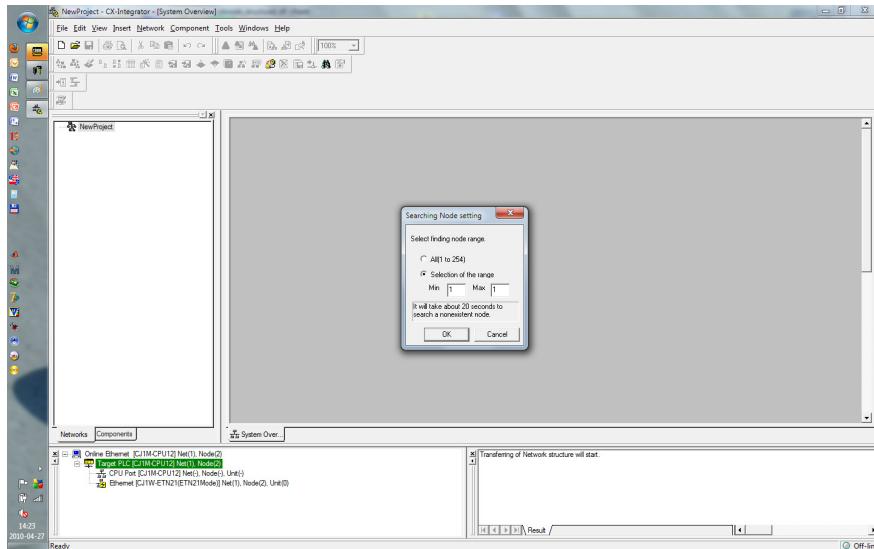
Rys.6.20. Uruchomienie programu Cx-Integrator dla modułu DeviceNet

Następuje uruchomienie programu Cx-Integrator. Po uruchomieniu program zaproponuje automatyczną konfigurację sieci. Aby to wykonać, należy nacisnąć przycisk Transfer, tak jak to pokazano na rys. 6.21.



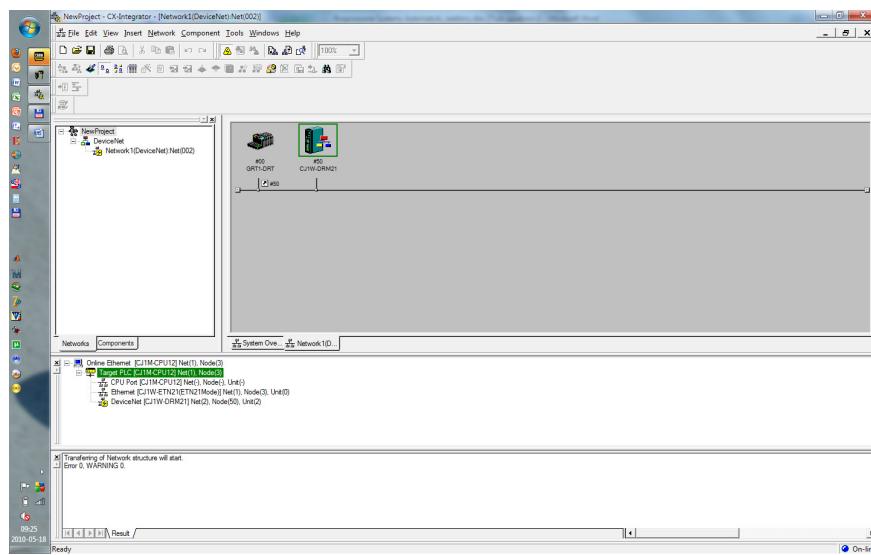
Rys.6.21. Okno rozpoczęcia automatycznego transferu parametrów sieci DeviceNet

Po uruchomieniu transferu oprogramowanie automatycznie wykryje strukturę sieci DeviceNet i zaproponuje odczyt parametrów zdalnych modułów sieciowych dołączonych do modułu Master Devicenet, co pokazano na rys.6.22.



Rys. 6.22. Okno transferu parametrów elementów sieci DeviceNet

Po wykonanym transferze struktury i parametrów modułów sieci deviceNet zostanie wyświetcone okno z konfiguracją sieci, pokazane na rys.6.23.



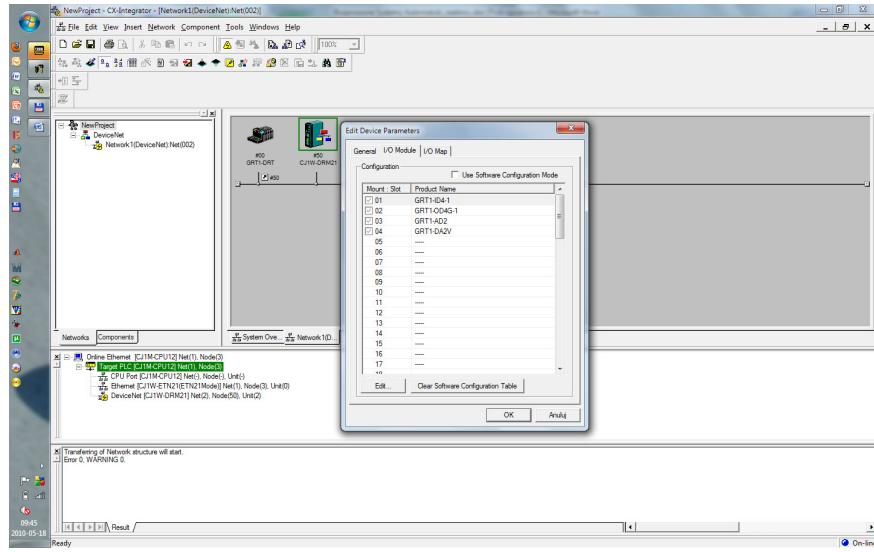
Rys.6.23. Okno konfiguracji sieci

W omawianym przykładzie w sieci DeviceNet znajdują się moduły:

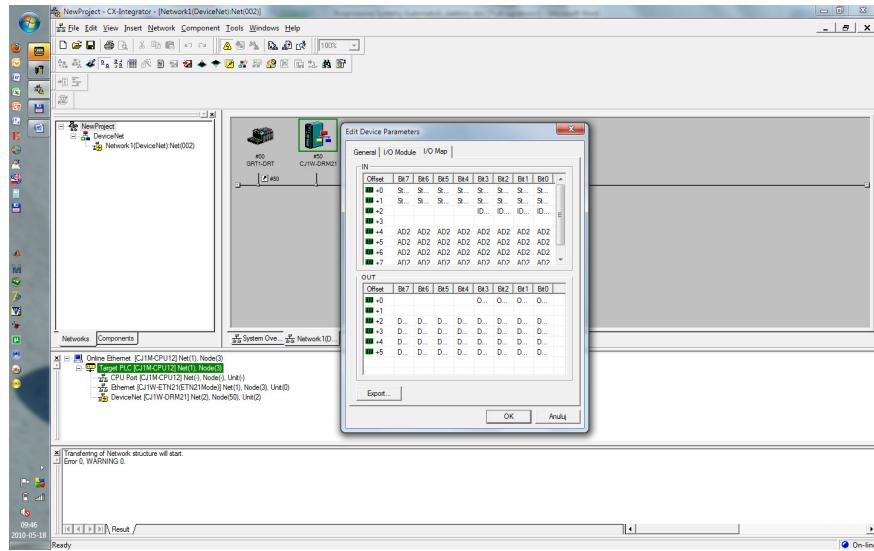
- DeviceNet Master typ CJ1M-DRM21,
- DeviceNet Slave typ GRT1-DRT.

Następnie należy określić adresy poszczególnych bitów zdalnych modułów wejścia/wyjścia. W tym celu należy określić adres bazowy każdego z modułów, a także zawartość poszczególnych słów w mapie pamięci modułu.

W celu ustalenia adresów bitów zdalnych modułów wejścia/wyjścia, należy kliknąć dwa razy lewym klawiszem myszy na obrazie zdalnego modułu DeviceNet GRT1-DRT. Wyświetli się okno konfiguracji modułu z widocznymi trzema zakładkami konfiguracyjnymi. W zakładce General wyświetlane są podstawowe informacje o zdalnym module, zakładka I/O module zawiera dokładną informację o wykrytych w sieci zdalnych modułach wejścia/wyjścia w kolejności montażu na magistrali zdalnej (rys.6.24), natomiast zakładka I/O Map zawiera szczegółowe informacje o adresach poszczególnych bajtów wejść/wyjść rozproszonych (rys.6.25).



Rys.6.24. Lista wykrytych zdalnych modułów wejścia/wyjścia sieci DeviceNet



Rys.6.25. Mapa pamięci zdalnych modułów wejścia/wyjścia sieci DeviceNet

W omawianym przykładzie mapa pamięci dla poszczególnych modułów wejścia/wyjścia jest następująca:

Wejścia cyfrowe (IN):

Adres bazowy modułu wejść +0 : bity statusowe bit7-bit0

Adres bazowy modułu wejść +1 : bity statusowe bit7-bit0

Adres bazowy modułu wejść +2 : bity danych IN3-IN0

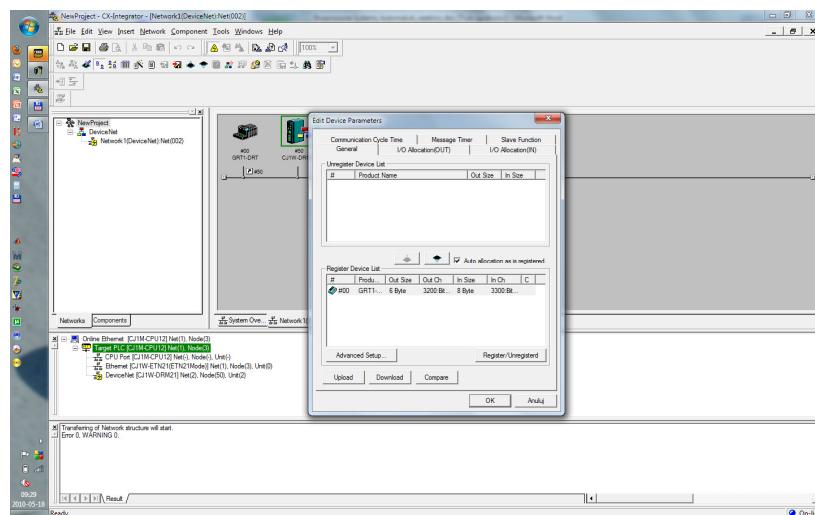
Wyjścia cyfrowe (OUT):

Adres bazowy modułu wyjść +0 : bity danych OUT3-OUT0

Adresy bazowe poszczególnych modułów wejścia/wyjścia umieszczone są w zakładce konfiguracyjnej modułu komunikacyjnego DeviceNet Master CJ1M-DRM21 i należy je odpowiednio odczytać. W tym celu należy kliknąć dwa razy lewym klawiszem myszy na obrazie modułu komunikacyjnego DeviceNet Master. W zakładce General wyświetli się konfiguracja wykrytych zdalnych modułów, jak to pokazano na rys.6.26.

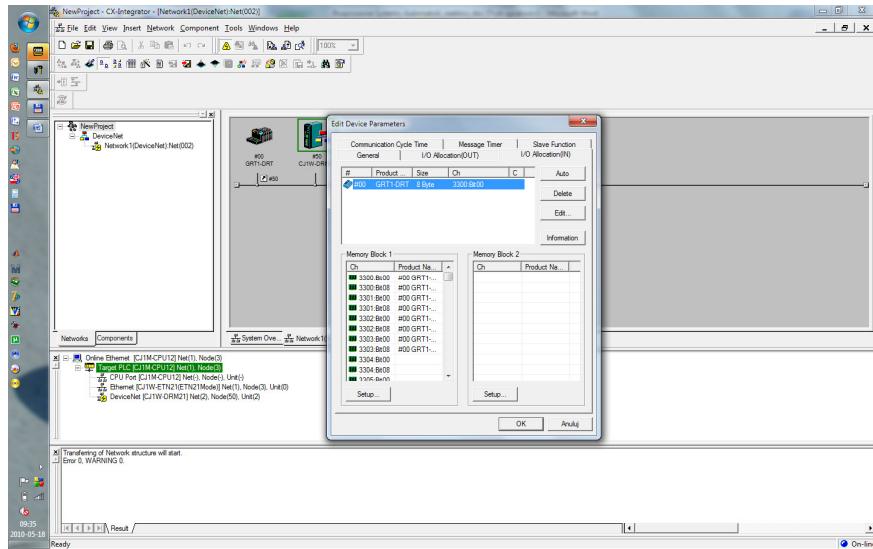
W omawianym przykładzie charakterystyka wykrytego modułu (Register Device) jest następująca:

- #: 00,
- Product: GRT1-DRT,
- Out Size: 6 byte,
- Out Ch: 3200: bit 00 – adres bazowy wyjść cyfrowych !,
- In Size: 8 byte,
- In Ch: 3300:bit 00 - adres bazowy wejść cyfrowych !



Rys.6.26. Przykładowa konfiguracja wykrytej sieci deviceNet

Dokładne położenie bajtów adresowych ujawnione jest w zakładkach I/O Allocation (OUT) oraz I/O Allocation (IN). W omawianym przykładzie położenie bajtów wejściowych pokazano na rys.6.27.



Rys.6.27. Położenie bajtów adresowych dla modułu wejść cyfrowych

W analogiczny sposób należy sprawdzić położenie bajtów adresowych dla pozostałych modułów wykrytych w sieci DeviceNet.

Należy zwrócić uwagę, że poszczególne moduły adresowane są za pomocą 16-bitowych słów, w związku z tym adresy poszczególnych bitów wejścia/wyjścia są takie jak w tabeli 6.2.

Tab.6.2. Adresy poszczególnych bitów wejścia/wyjścia

Wejścia		Wyjścia	
Numer wejścia	Adres	Numer wyjścia	Adres
IN0	3301.0	OUT0	3200.0
IN1	3301.1	OUT1	3200.1
IN2	3301.2	OUT2	3200.2
IN3	3301.3	OUT3	3200.3

Po wykonanej konfiguracji można zapisać konfigurację sieci na dysku oraz zamknąć program konfiguracyjny sieci DeviceNet Cx-Integrator.

6.2.3. PRZYKŁAD STEROWANIA ZDALNYMI MODUŁAMI WEJŚĆ/WYJŚĆ

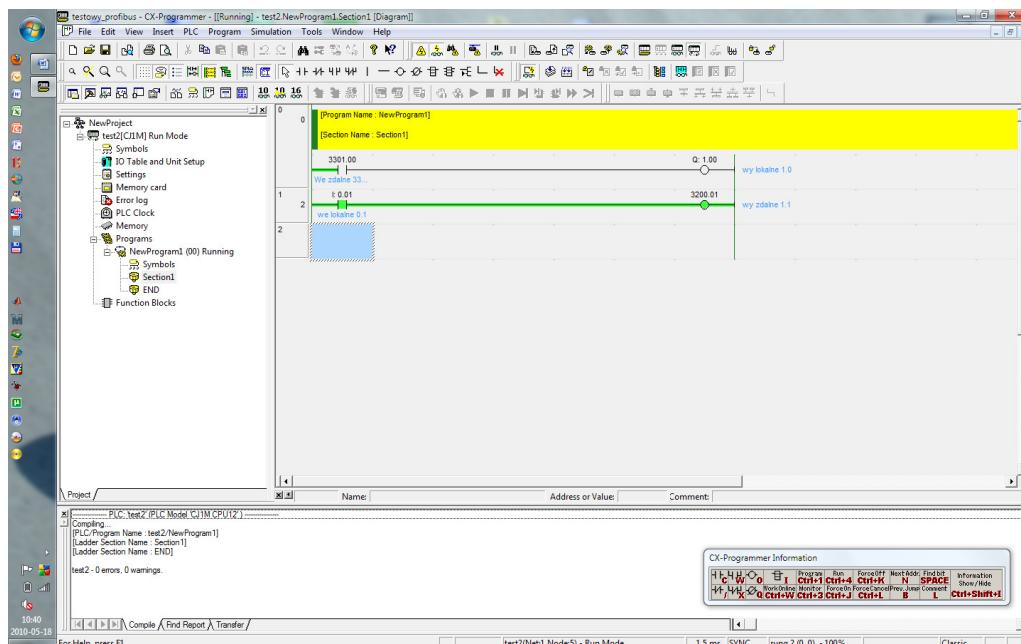
Poniżej pokazano sposób odczytu i zapisu informacji z wykorzystaniem rozproszonych wejść/wyjść.

W rozpatrywanym przykładzie odczytywany jest stan bitu na zdalnym wejściu I0.0 i przekazywany na wyjście sterownika O1.0. Następnie stan wejścia lokalnego sterownika I0.1 przepisywany jest na zdalne wyjście O1.1.

Zgodnie z zasadami opisanymi w poprzednich podrozdziałach, należy odnaleźć adresy poszczególnych zdalnych wejść i wyjść. W opisywanym przykładzie są to odpowiednio:

- adres zdalnego wejścia I0.0: 3301.0,
- adres zdalnego wyjścia O1.1: 3200.1.

Na rys. 6.28 pokazano program realizujący opisane zadanie.



Rys.6.28. Przykładowy program sterujący zdalnymi wejściami/wyjściami

