

Uniwersytet Mikołaja Kopernika
Wydział Fizyki, Astronomii i Informatyki Stosowanej



kierunek: Fizyka Techniczna
specjalność: Miernictwo Komputerowe

Katarzyna Ostrowska

Stanowisko dydaktyczne z programatorem
Vision 260

Praca inżynierska
wykonana w Zakładzie Fizyki Technicznej
opiekun: **dr inż. Kazimierz Karwowski**

Toruń 2007

*Dziękuję
dr. inż. K. Karwowskiemu i mgr M. Gahblerowi
za okazana pomoc i cenne wskazówki.*

UMK zastrzega sobie prawo własności niniejszej pracy magisterskiej (licencjackiej, inżynierskiej) w celu udostępniania dla potrzeb działalności naukowo-badawczej lub dydaktycznej

Spis treści

	Strona
1. Wstęp	5
2. Sterowniki PLC.....	6
3. Programowanie PLC.....	8
4. Sterowniki Vision.....	9
4.1. Sterowniki z rodziny Vision.....	9
4.2. Budowa sterownika Vision 260 i modułu Snap V200-18-E2B.....	10
4.2.1. Warunki pracy	11
4.2.2. Wyświetlacz graficzny i klawiatura.....	11
4.2.3. Porty komunikacyjne.....	15
4.2.4. Budowa modułu Snap V200-18-E2B.....	15
4.3. Oprogramowanie.....	20
4.3.1. Pierwsze uruchomienie VisiLogic.....	21
4.3.2. Aplikacja Ladder.....	22
4.3.3. Aplikacja HMI.....	40
5. Stanowisko dydaktyczne ze sterownikiem Vision 260.....	45
5.1. Budowa stanowiska.....	45
5.2. Oprogramowanie stanowiska.....	49
6. Podsumowanie.....	52
7. Literatura.....	53

Załączniki:

Załącznik 1. Podręcznik użytkownika Vision 230, 260, 280

Załącznik 2. V200-18-E2B Snap-in I/O Module

Załącznik 3. Instrukcja Obsługi SMC 81

Załącznik 4. Kod programu stanowiska ze sterownikiem Vision 260

Załącznik 5. Instrukcja obsługi stanowiska dydaktycznego z programatorem Vision 260

1. Wstęp

W ostatnich latach sterowniki PLC przeszły burzliwy rozwój, przez co stały się bardziej wydajne i łatwiejsze w obsłudze. Możliwość połączenia sterowników w sieci umożliwia obsługę bardzo dużej ilości urządzeń naraz, bez potrzeby tworzenia dużych i niewygodnych w użyciu układów przekaźnikowych. Ich zalety sprawiły że znalazły one szerokie zastosowanie w wielu zakładach przemysłowych.

Niniejsza praca została poświęcona opracowaniu stanowiska dydaktycznego z programatorem Vision 260. Przy jego budowie starano się wykorzystać jak najwięcej możliwości jakie posiada sterownik. W skład stanowiska wchodzi dwa silniki krokowe przeznaczone do testowania wyjść typu high-speed, układ przełączników symulujący stany na wejściach cyfrowych, diody sygnalizujące stan wyjść przekaźnikowych, oraz układ potencjometri współpracujący z wejściami analogowymi.

Rozdział drugi przedstawia historię rozwoju sterowników, ich zastosowanie i ogólną budowę. Znajduje się tu także opis normy **IEC 1131**, która standaryzuje osprzęt, określa wymagania testowe, oraz ujednolica sposób programowania w językach tekstowych i graficznych.

Rozdział trzeci opisuje sposób programowania sterowników, a także wszystkie dostępne obecnie języki programowania PLC.

Czwarty rozdział to szczegółowy opis sterowników serii Vision OPLC z wyszczególnieniem sterownika Vision 260. Znajdują się tam informacje na temat budowy wejść i wyjść, opis środowiska programistycznego VisiLogic i dostępnych w nim funkcji oraz zmiennych.

Głównym celem tego rozdziału jest przedstawienie zasady działania sterownika Vision, po to, aby każdy użytkownik mógł korzystać ze stanowiska w bezpieczny sposób i w pełni wykorzystywać jego możliwości.

Piąty rozdział to opis budowy stanowiska i jego oprogramowania. Zamieszczono w nim schemat elektryczny stanowiska wraz z opisem wyjaśniającym działanie poszczególnych urządzeń. Znajduje się tu także schemat blokowy programu sterownika Vision 260.

Ostatni rozdział stanowi podsumowanie całej pracy. Przedstawiono w nim uwagi na temat ograniczeń sterownika Vision 260, a także możliwości których w tej pracy nie zdołano wykorzystać.

2. Sterowniki PLC

Sterowniki programowalne PLC (Programmable Logic Controllers) rozwinęły się na początku lat siedemdziesiątych XX wieku. Zastąpiły niewygodne w użyciu układy przekaźnikowe. Szybko znalazły szerokie zastosowanie w przemyśle, głównie dla tego, że są łatwe w programowaniu, umożliwiając kontrolę programową wejść i wyjść, a także pozwalają na dobranie instrukcji w taki sposób, aby można było uwzględnić warunki pracy danego urządzenia.

Pierwsze modele sterowników PLC nie były ze sobą kompatybilne i nie było możliwości zwiększania ilości obsługiwanych wejść i wyjść. Dzisiaj producenci oferują całe rodziny sterowników o podobnej budowie, które można programować w tym samym języku wykorzystując te same biblioteki i zmienne. Dostępne są także różnego rodzaju moduły rozszerzeń wejść i wyjść dzięki którym można zwiększyć liczbę obsługiwanych urządzeń.

W 1993 r. Międzynarodowa Komisja Elektroniki (International Electronical Commission IEC) wydała normę **IEC 1131** „Programmable Controllers”. Norma ta standaryzuje osprzęt, określa wymagania testowe, oraz ujednolica sposób programowania w językach tekstowych i graficznych. Znajdują się tam wytyczne dla użytkownika, a także model komunikacji.

Obecnie sterowniki programowalne PLC to komputery przemysłowe oparte na systemie operacyjnym czasu rzeczywistego. W skład architektury sterownika wchodzi:

- **moduły wejść i wyjść** połączone z jednostką CPU magistralą. Wyróżniamy moduły cyfrowe, analogowe, specjalizowane np. Liczniki czy moduły przeznaczone do pracy w sieci
- **jednostki centralnej CPU** (układ wieloprocesorowy) z pamięcią (EPROM lub FLASH służącą do przechowywania systemu operacyjnego, a także EEPROM lub FLASH w których przechowywany jest program i dane)
- **zasilacza**
- **plyty łączeniowej** (CPU baseplate, rack) która pełni rolę elementu konstrukcyjnego lub magistrali.

Zadaniem sterowników PLC jest:

- zbieranie pomiarów z różnych czujników i urządzeń pomiarowych, podłączonych do modułów wejść/wyjść (modułów I/O)
- przetwarzanie zebranych danych

- wykonanie programu wprowadzonego przez użytkownika w oparciu o stan kontrolowanych wejść i wyjść
- wygenerowanie sygnału sterującego dla danego urządzenia.

Działanie sterowników to przede wszystkim przetwarzanie sygnałów wejściowych, pochodzących z podłączonych do sterownika urządzeń peryferyjnych, na sygnały logiczne zrozumiałe dla jednostki logicznej CPU. Dalej opierając się na instrukcjach zawartych w pamięci jednostka CPU wykonuje funkcje sterowania. Funkcje sterowania przetwarzane są przez moduły wyjściowe na sygnały analogowe lub cyfrowe. Informacje o realizowanym procesie możemy obejrzeć na panelu operatorskim.

3. Programowanie PLC

Obecnie do programowania sterowników PLC wykorzystuje się programatory lub komputery PC z zainstalowanym odpowiednim środowiskiem programistycznym. Sterownik programujemy stosując języki tekstowe bądź graficzne. Do pierwszej grupy należą:

Język listy instrukcji (IL – Instruction List) – jak sama nazwa wskazuje składa się z szeregu instrukcji. Każda z nich powinna zaczynać się od nowej linii. Pojedyncza instrukcja zawiera:

- Operator – to wykonywana funkcja, może zawierać modyfikatory. Każdemu operatorowi przypisany jest co najmniej jeden operand.
- Operand – są to stałe lub zmienne przypisane do odpowiedniego operatora

Na początku instrukcji można znaleźć etykietę zakończoną dwukropkiem, natomiast na końcu instrukcji umieszczamy komentarz. Między instrukcjami mogą znajdować się puste linie.

Język strukturalny (ST – Structured Text) – język ten zawiera wyrażenia i polecenia w postaci tekstowej. Wyrażenie składa się z operatorów (funkcji) i operandów (zmienna, stała, wywołana funkcja itd.). Polecenia to przede wszystkim pętle takie jak: if, case, for, while, repeat. Znajdziemy tu także polecenie przypisania, wywołanie bloku funkcyjnego czy wyjście (EXIT). Wywołanie funkcji ST polega na wpisaniu do wyrażenia jej nazwy i listy argumentów podobnie wywołujemy bloki funkcyjne.

Do grupy języków graficznych należą:

Język schematów drabinkowych (LD- Ladder Diagram) – jest to język którego struktura przypomina schemat elektryczny. Programując wykorzystujemy szereg standaryzowanych symboli graficznych. Program napisany w LD składa się z wzajemnie połączonych elementów zwanych obwodami. Każdy obwód w LD ograniczony jest szynami prądowymi: lewą i prawą (prawa nie zawsze jest rysowana). Stan lewej szyny jest zawsze wysoki natomiast dla prawej szyny stanu się nie definiuje. Wykonywanie programu w LD polega na przepływie prądu z lewej strony do prawej poprzez poszczególne elementy.

Język schematów bloków funkcyjnych (FBD – Function Block Diagram) - język ten polega na przepływie sygnału z wyjścia z prawej strony funkcji (bloku funkcyjnego) do wejścia z lewej strony następnej funkcji (bloku funkcyjnego). Program konstruujemy ze zdefiniowanych w oprogramowaniu funkcji. Zbiór połączonych ze sobą funkcji tworzy obwód.

4. Sterowniki Vision

4.1. Sterowniki z rodziny Vision

Firma Unitronics opracowała serię sterowników Vision OPLC:

- Vision 120
- Vision 230
- Vision 260
- Vision 280
- Vision 290
- Vision 530
- Vision 570

Są to sterowniki należące do grupy sterowników programowalnych. Każdy sterownik serii Vision OPLC posiada zintegrowany panel operatorski, dwa porty komunikacyjne RS232, port CANbus a także RTC, czyli zegar czasu rzeczywistego.

Część sterowników Vision tak jak Vision 260 posiada graficzny wyświetlacz LCD z klawiaturą, inne np. Vision 280 wyposażone są w panel graficzny dotykowy. Panel graficzny znacząco ułatwia pracę ze sterownikiem. Wyświetlane są na nim instrukcje, obrazy graficzne, zmienne systemowe, alarmy a także czas. Panel operatorski umożliwia nam kontrolę zachodzącego procesu a także pozwala na modyfikację danych w sterowniku.

Klawiatura sterownika Vision posiada od kilkunastu do kilkudziesięciu programowalnych klawiszy (w zależności od modelu), które umożliwiają wprowadzenie danych oraz sprawne poruszanie się po programie zawartym w sterowniku (Rys.1.).



Rys.1. Panel operatorski sterownika Vision 230

Każdy ze sterowników Vision może współpracować z modułami wejść i wyjść: Snap I/O lub modułami rozszerzeń I/O.

Moduł Snap to moduł wejść i wyjść dołączanych do sterownika. Obecnie może on być podłączony tylko do serii Vision 200 i Vision 500 (można podłączyć tylko jeden Snap do jednego panelu operatorskiego). Moduł Snap jest umieszczany z tyłu sterownika.

Obecnie dostępne są następujące moduły Snap:

- V200- 18- E1B
- V200- 18- E2B
- V200- 18- E3XB
- V200- 18- E4XB
- V200- 18- E5B

W zależności od modelu moduł Snap zawiera:

- wejścia /wyjścia dyskretne (dwustanowe)
- wejścia/wyjścia analogowe
- wyjścia tranzystorowe
- wyjścia przekaźnikowe
- wyjścia high-speed (szybkie liczniki do 50 kHz)
- wejścia enkoderowe do 10 kHz

Do rozbudowy sterowników o kolejne porty wejścia/wyjścia służą moduły rozszerzeń I/O. Rozbudowę umożliwia nam port rozszerzeń wbudowany w sterowniki Vision. Do jednego sterownika możemy podłączyć do ośmiu modułów. Moduły rozszerzeń w większości wymagają stosowania adaptera EX-A1 który łączy je ze sterownikiem Vision.

4.2. Budowa sterownika Vision 260 i modułu Snap V200-18-E2B

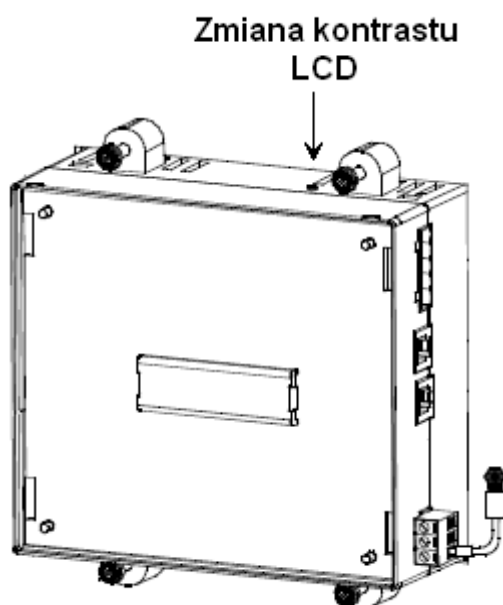
Sterownik Vision 260 należy do grupy sterowników Vision OPLC. Posiada zintegrowany panel operatorski z trzydziestoma trzema programowalnymi klawiszami. Ma wbudowane dwa porty komunikacyjne RS232 oraz CANbus. Posiada zegar czasu rzeczywistego (RTC) i port rozszerzeń.

4.2.1. Warunki pracy

Sterownik Vision 260 wymaga zewnętrznego zasilania napięciem 12 VDC lub 24 VDC dopuszczalna wartość napięcia zasilania to 10,2 VDC do 28,8 VDC z wahaniami nie przekraczającymi 10%. Sterownik ten może pracować w temperaturze od 0 do 50°C przy wilgotności względnej od 5% do 95% bez skraplania. Prąd zasilania sterownika przy napięciu zasilania 12 VDC nie przekracza 460 mA, dla zasilania 24 VDC nie przekracza 220 mA. Moc pobierana przez sterownik to 4,2 W.

4.2.2. Wyświetlacz graficzny i klawiatura

Vision 260 posiada wyświetlacz typu Negative blue STN LCD (Rys.2.). Do podświetlenia użyto lamp fluorescencyjnych CCFL. Rozdzielczość ekranu to 240x64 pixele. W górnej części sterownika za pomocą płaskiego śrubokręta możemy ustawić kontrast ekranu (Rys.2.). Program obsługujący panel graficzny tworzymy w środowisku programistycznym VisiLogic przy użyciu aplikacji HMI. Na panelu można umieścić instrukcje dla użytkownika, oraz różnego rodzaju zmienne systemowe zarówno w postaci tekstowej jak i graficznej. W aplikacji HMI dla Sterownika Vision 260 można również tworzyć animacje, wyświetlać stan wejść i wyjść, a także przypisywać klawiszom określone funkcje.



Rys.2. Sterownik Vision – widok z tyłu¹

¹ Rysunek zaczerpnięty z dokumentacji do sterownika Vision 260

Klawisze w sterowniku Vision 260 to przyciski membranowe, tworzą je metalowe kopuły. Opis klawiszy to zadrukowana odpowiednimi symbolami folia. Klawiatura sterownika składa się z trzydziestu trzech klawiszy (Rys.3.) z czego dziesięć to klawisze alfanumeryczne a pozostałe to klawisze funkcyjne: strzałka w górę, w dół, w lewo, w prawo, klawisze ENTER, ESC, plus/minus oraz klawisz inf.(i), a także klawisze od F1 do F15.

Klawisze umożliwiają użytkownikowi sprawne poruszanie się po umieszczonym w sterowniku programie, wprowadzenie niezbędnych danych, a także kontrolowanie zachodzącego procesu.



Rys.3. Panel operatorski sterownika Vision 260

Sterownik Vision 260 posiada tzw. tryb informacji jest on uruchamiany za pomocą klawisza inf. (Rys.3.). Tryb ten daje dostęp do wszystkich zmiennych i ustawień sterownika. Można uruchomić go w dowolnej chwili nawet podczas trwania programu, nie ma to wpływu na jego pracę. W trybie informacji klawisze mają przypisane funkcje, które umożliwiają poruszanie się po kategoriach. Klawisze wracają do swojego pierwotnego przeznaczenia z chwili opuszczenia trybu informacji.

Aby wejść do trybu informacji użytkownik musi przytrzymać klawisz inf. (i) przez kilka sekund. Na ekranie ukaże się pole hasła. Po poprawnym wpisaniu hasła uzyskuje się dostęp do poszczególnych kategorii. Strzałki góra i dół służą do poruszania

się po kategoriach. Klawiszem ENTER zatwierdzamy wybór kategorii. Aby cofnąć się o jeden poziom wciskamy klawisz ESC, służy on także do wyjścia z trybu informacji.

Struktura trybu informacji :

Tab.1.Struktura trybu informacji cz.1

Kategoria	Temat	Opis
Typy Danych (Data Types)	Bity Pamięci (Memory Bits)	- umożliwia sprawdzenie stanu bitów.
	Bity Systemowe (System Bits)	- możliwość zmiany stanu bitów (wysoki/niski).
	Memory Integers System Integers Memory Longs System Longs Memory Double Words System Double Words	- można uzyskać podgląd zmiennych zarówno w formie decymalnej jak i heksadecymalnej oraz zmienić wartość wybranej zmiennej.
	Wejścia (Inputs)	- otrzymujemy dostęp zarówno do podglądu stanu wejść jak i do zmiany ich stanu (z 0 na 1 i z 1 na 0).
	Wyjścia (Outputs)	- możliwy jest dostęp do podglądu stanu wyjść oraz zmiany ich stanu (1/0).
	Liczniki (Timers)	- umożliwia sprawdzenie aktualnej wartości licznika a także ustawienie innej.

Tab.2.Struktura trybu informacji cz.2

Kategoria	Temat	Opis
System	Model i wersja systemu operacyjnego (Model & O/S Ver)	- umożliwia sprawdzenie modelu sterownika i wersję systemu operacyjnego, a także sprawdzenie czy sterownik działa w danej chwili w trybie Run czy Stop.
	Tryb pracy (Working Mode)	-tutaj mamy możliwość: * zresetowania sterownika: następuje restart programu przy czym otrzymujemy power-up we wszystkich typach danych (nie dotyczy danych chronionych przed podtrzymaniem pamięci) * inicjalizację sterownika, podczas której wszystkie typy danych przyjmują wartość 0 * sprawdzenia czy sterownik znajduje się w trybie Run czy Stop
	Czas i Data (Time & Date)	- podgląd i możliwość zmiany daty i czasu.
	Numer ID (Unit ID)	- pozwala sprawdzić oraz zmienić numer ID, który identyfikuje sterownik w sieci.
	Serial port 1 Serial port 2	- możliwy jest podgląd ustawień komunikacyjnych a także ich zmiana
	CANbus Baud Rate	- umożliwia ustawienie prędkości CANbus.
Blok Funkcji (Function Block)	FBs in use	- tutaj można obejrzeć listę wszystkich bloków funkcyjnych jakie były wgrywane do sterownika.
Hasło (Password)	Nowe (New)	- możliwość zmiany hasła dostępu do trybu informacji.
(Hardware Configuration)		- możliwość sprawdzenia czy moduły I/O są podłączone, a także czy są zwarte.

UWAGA!!!

Zmiana jakichkolwiek danych w trybie informacji może wpłynąć negatywnie na pracę zainstalowanego programu !!!

4.2.3.Porty komunikacyjne

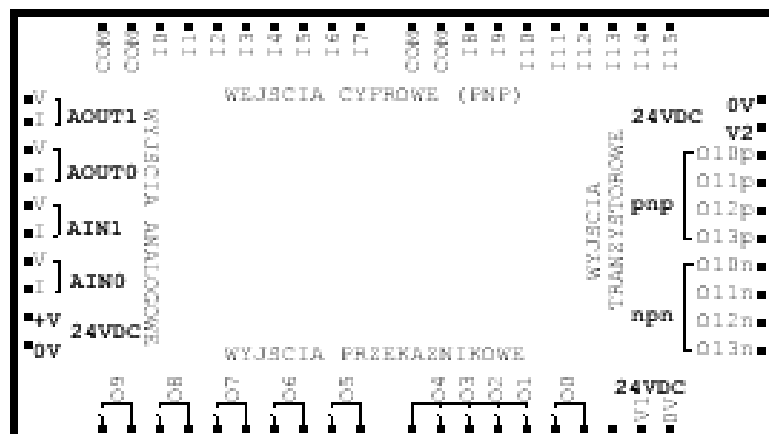
Sterownik Vision 260 posiada dwa porty komunikacyjne RS232 (nie są optoizolowane). Do ich zadań należy: przesyłanie programu z komputera do sterownika, a także komunikacja w sieci, czy też oglądanie i modyfikowania danych. Zakończeniem portu RS232 w sterowniku Vision 260 jest gniazdo RJ-11. Opis wyprowadzeń portów RS232 znajduje się w załączniku nr 1 na stronie 16.

Kolejnym portem komunikacyjnym jest CANbus służy on do tworzenia sieci sterowników. Może się w niej znajdować do 63 sterowników przy czym możliwa jest komunikacja z komputerem. Budowę złącza magistrali CANbus oraz specyfikacje można odnaleźć w załączniku 1 na stronie 18-19.

4.2.4.Budowa modułu Snap V200-18-E2B

Moduł Snap V200- 180-E2B jest umieszczany z tyłu sterownika i łączony z nim poprzez złącze I/O connector cap. Rozbudowuje on sterownik o osiemnaście wejść i dwadzieścia wyjść które podlegają całkowitej kontroli sterownika .Snap posiada (Rys.4.):

- **szesnaście izolowanych wejść cyfrowych** które mogą pracować jako npn lub pnp, z czego dwa z nich mogą pracować jako wejścia high-speed.
- **dziesięć izolowanych wyjść przekaźnikowych**
- **osiem wyjść tranzystorowych** z czego cztery pracują jako pnp a pozostałe jako npn, dwa z tych wyjść mogą działać jako wyjścia high-speed
- **dwa wejścia analogowe oraz dwa wyjścia analogowe**

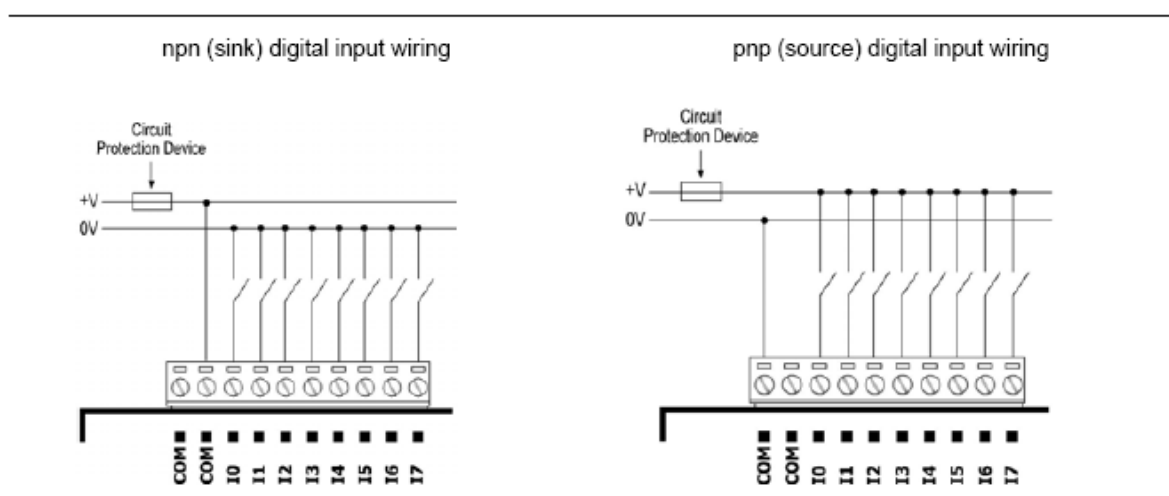


Rys.4. Snap V200-18-E2B – rozkład wejść i wyjść

Snap wymaga zewnętrznego zasilania 24 VDC. Nie pobiera on zasilania ze sterownika!!! Do każdej grupy wejść/wyjść napięcie doprowadzane jest osobno w miejscach oznaczonych „ 24VDC ”. Moduł Snap V200-18-E2B pracuje w temperaturze od 0°C do 60°C przy wilgotności od 5% do 95% (bez skraplania).

Budowa wejść/wyjść modułu Snap V200- 180-E2B:

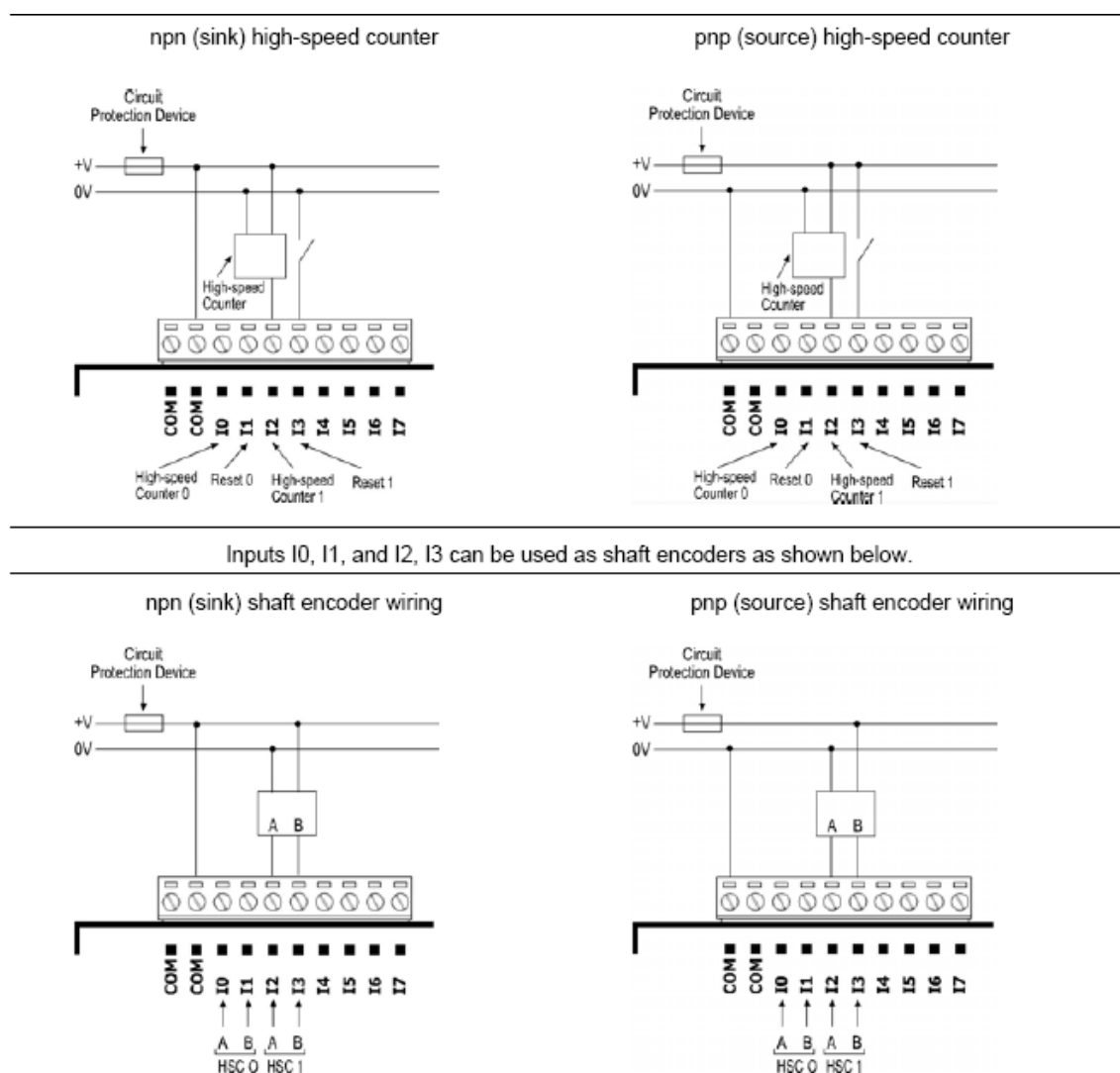
Wejścia cyfrowe są izolowane, tworzą je dwie grupy liczące po osiem wejść. Pierwszą z nich tworzą wejścia od I0 do I7, drugą grupę tworzą wejścia I8 do I15 . Każda z grup posiada dodatkowe dwa wejścia (com), które regulują prace wejść I0- I7 oraz I8- I15 jako pnp lub npn. Gdy wyjścia pracują jako pnp logiczne 0 otrzymujemy w przedziale napięć 0-5 VDC, natomiast logiczna 1 w przedziale 17-28.8 VDC. W przypadku wejść npn logiczne 0 to 17-28.8 VDC, a logiczna 1 to 0-5 VDC. Czas reakcji wejść wynosi 10 ms. Sposób podłączenia przedstawiono na Rys.5.



Rys. 5. Podłączenie wejść cyfrowych¹

¹ Rysunek został zaczerpnięty z dokumentacji modułu Snap V200-18-E2B ze str.3

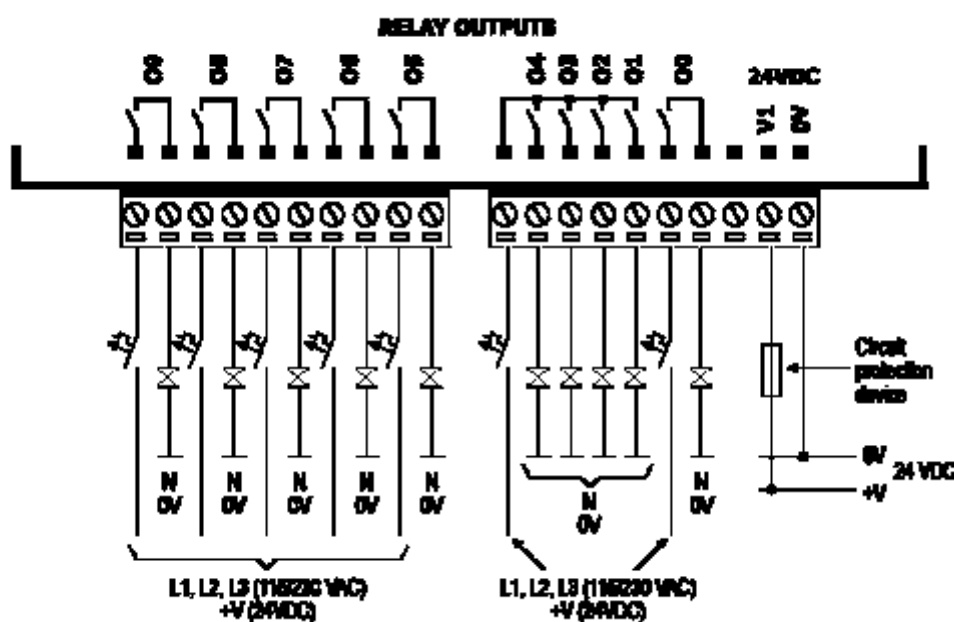
Wejścia I0 i I2 mogą pracować jako normalne wejścia cyfrowe, wejścia typu szybki licznik (high-speed counters), lub jako wejścia przetwornika obrotowego-impulsowego przyrostowego (shaft encoder wiring). Wejścia I1 oraz I3 także mogą pracować jako normalne wejścia cyfrowe, wejścia typu szybki licznik zerujący (high-speed counters), czy też jako wejścia przetwornika obrotowego-impulsowego przyrostowego (shaft encoder wiring). Częstotliwość pracy wejść w trybie high-speed dochodzi do 10 kHz, a minimalna wartość szerokości impulsu to 40 us. Sposób podłączenia tych wejść w zależności od przeznaczenia pokazano na rysunku : Rys.5. oraz Rys.6.



Rys. 6. Podłączenie wejść cyfrowych¹

¹ Rysunek został zaczerpnięty z dokumentacji modułu Snap V200-18-E2B ze str.3

Wyjścia przekaźnikowe są izolowane galwanicznie. Wyjścia od O5 do O9 oraz O0 mają osobne styki pozostałe posiadają jedną linię wspólną styków. Prąd obciążenia rezystancyjnego styków to maksymalnie: 5 A na wyjściach O5-O9 i O0, oraz 8 A na wyjściach O4- O1. W przypadku obciążenia indukcyjnego odpowiednio 1 A i 4 A. Maksymalna częstotliwość to 10 Hz. Nominalna wartość zasilania wynosi 24 VDC. Sposób podłączenia urządzeń do wyjść pokazany jest na Rys.7.

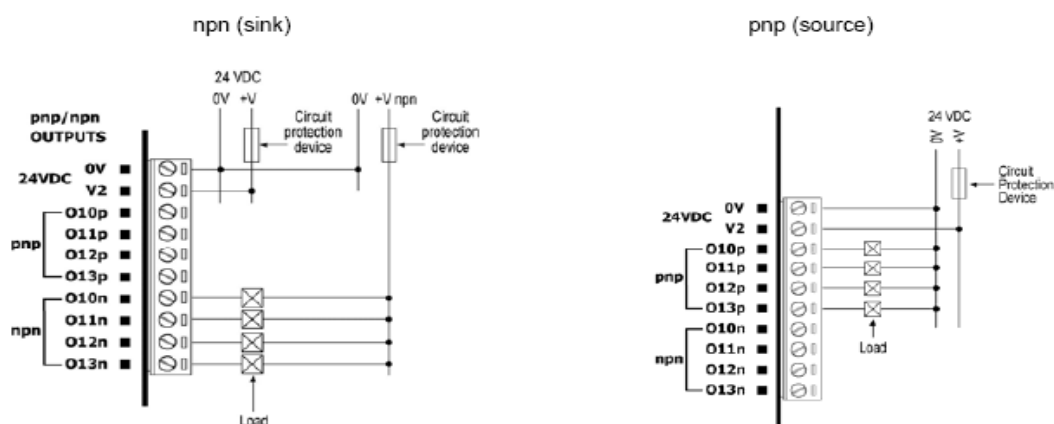


Rys. 7. Podłączenie wyjść przekaźnikowych¹

W celu zabezpieczenia wyjść przekaźnikowych w przypadku obciążenia indukcyjnego należy podłączyć równolegle do obciążenia diodę przy napięciu stałym lub układ RC przy napięciu zmiennym.

Wyjścia tranzystorowe są izolowane galwanicznie, dzielą się na cztery wyjścia typu pnp i cztery wyjścia typu npn, z czego O12 i O13 mogą pracować w trybie high-speed. Nominalna wartość zasilania to 24 VDC. Maksymalny prąd obciążenia dla tych wyjść to odpowiednio 0.5 A i 50 mA. Maksymalna częstotliwość przy obciążeniu rezystancyjnym to dla pnp 20 Hz a dla npn 0.5 Hz. Dla wyjść typu high-speed maksymalna częstotliwość wynosi: pnp - 2 kHz, npn - 50 kHz. Podłączenie urządzeń do wyjść pokazano na Rys.8.

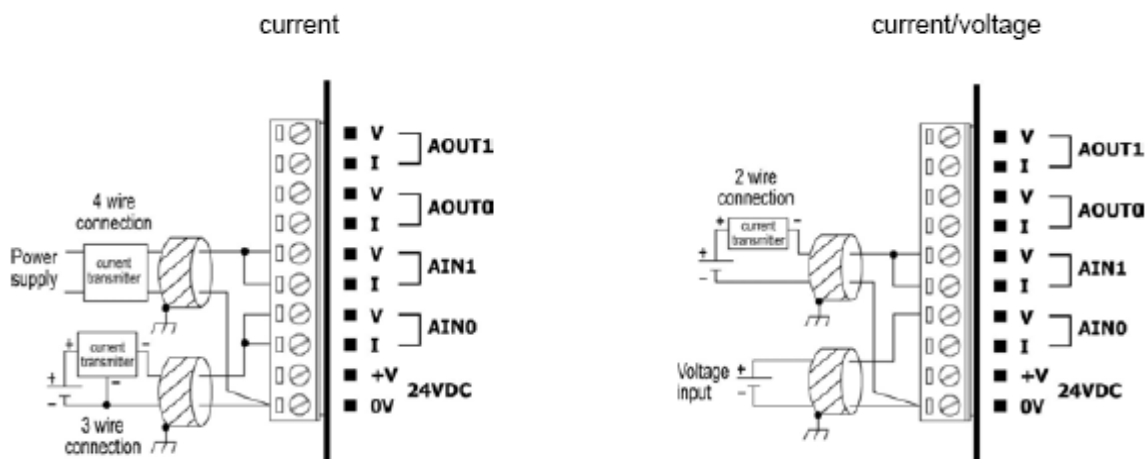
¹ Rysunek został zaczerpnięty z dokumentacji modułu Snap V200-18-E2B ze str.4



Rys. 8. Podłączenie wyjść tranzystorowych¹

Wejścia analogowe to dwa nieizolowane wejścia typu single-ended to znaczy, że mamy osobne wejście do pomiaru prądu i napięcia (Rys.9.). Zakresy pracy wejść to: 0-10 V, 0-20 mA, 4-20 mA. Do konwersji danych otrzymanych w pomiarze używa się sukcesywnej aproksymacji.

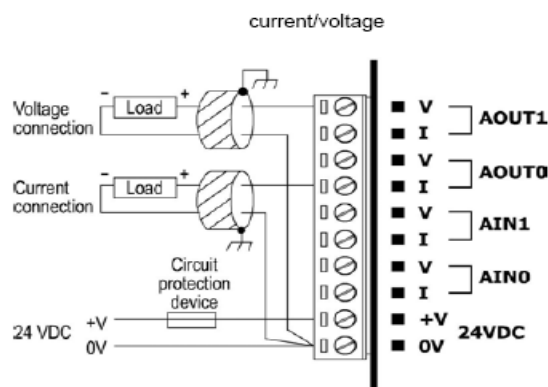
Impedancja wejściowa to ponad 100 k Ω dla wejść napięciowych i 500 Ω dla prądowych. Maksymalne wartości dla wejść napięciowych to ± 15 V, a dla prądowych ± 30 mA.



Rys. 9. Podłączenie wejść analogowych²

1 Rysunek został zaczerpnięty z dokumentacji modułu Snap V200-18-E2B ze str.5
2 Rysunek został zaczerpnięty z dokumentacji modułu Snap V200-18-E2B ze str.5

Wyjścia analogowe to dwa nieizolowane wyjścia i tak jak wejścia analogowe są typu single-ended (Rys.10.). Ich zakresy pracy to: 0-10 V, 0-20 mA, 4-20 mA. Impedancja obciążenia to minimalnie 1 k Ω dla wejść napięciowych i maksymalnie 500 Ω dla prądowych.



Rys. 10. Podłączenie wyjść analogowych¹

4.3. Oprogramowanie

Vision 260 pozwala na zapisanie 192 k kodu drabinki programu. Do tworzenia programu służy oprogramowanie VisiLogic dołączone do sterownika lub pobrane bezpłatnie ze strony www.support.elmark.com.pl/unitronics/. Minimalne wymagania sprzętowe do uruchomienia oprogramowania to :


- Windows ME/ 2000/ XP
- komputer PC z procesorem 300 MHz
- 250 MB pamięci RAM oraz 230 MB wolnego miejsca na twardym dysku
- minimalna rozdzielczość monitora to 800x600

Podczas instalacji oprogramowania postępujemy zgodnie z zaleceniami producenta.

Oprogramowanie VisiLogic służy do tworzenia kodu programu, weryfikowania błędów składni, a także do programowania dowolnego sterownika z serii Vision. Składa się z trzech edytorów: Ladder, HMI Display oraz variable. Każdy z nich odpowiada za inną część kodu. Ladder służy do tworzenia kodu drabinki, w HMI Display definiujemy zmienne, które chcemy umieścić na ekranie LCD natomiast variable odpowiada za wszystkie tworzone przez nas zmienne.

¹ Rysunek został zaczerpnięty z dokumentacji modułu Snap V200-18-E2B ze str.6

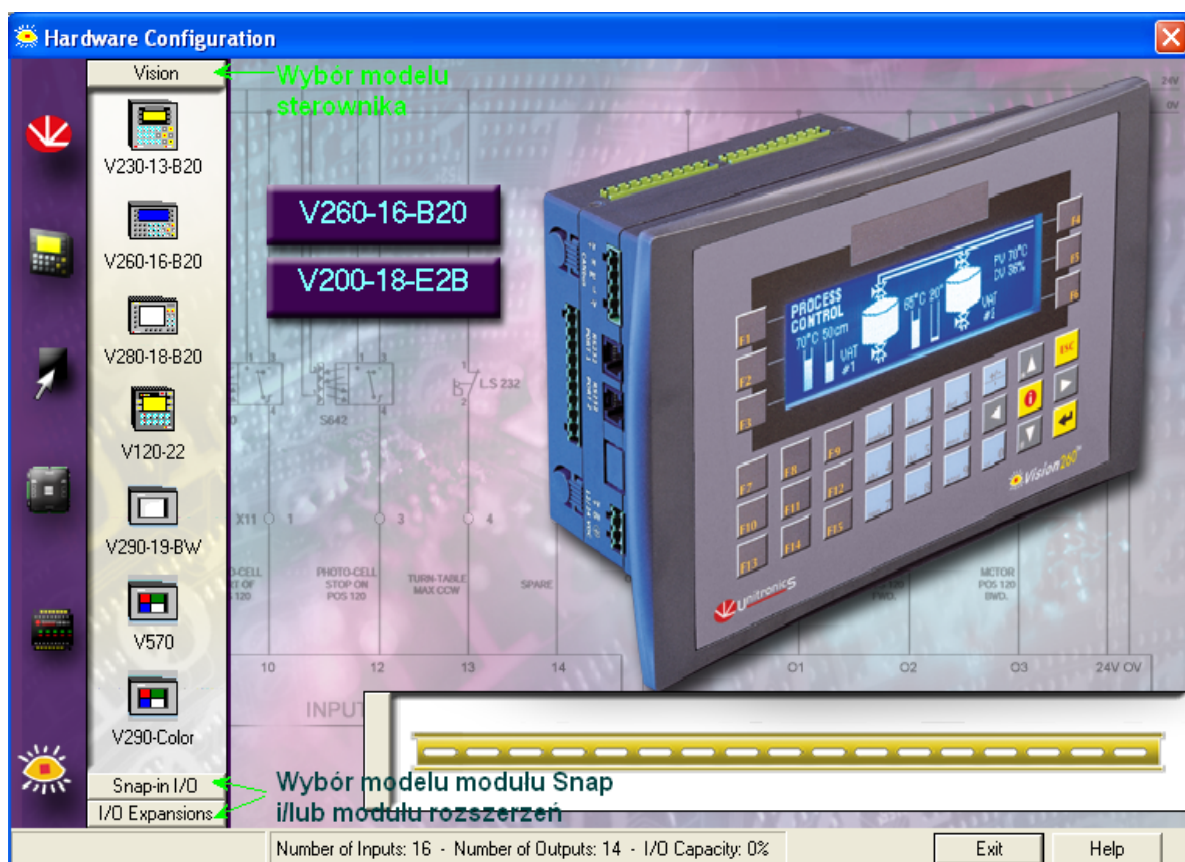
4.3.1. Pierwsze uruchomienie VisiLogic

Podczas pierwszego uruchomienia należy skonfigurować osprzęt (Rys.11.). Okno pojawia się automatycznie, ale w każdej chwili można wrócić do konfiguracji poprzez wciśnięcie klawisza:  - Konfiguracja sterownika.

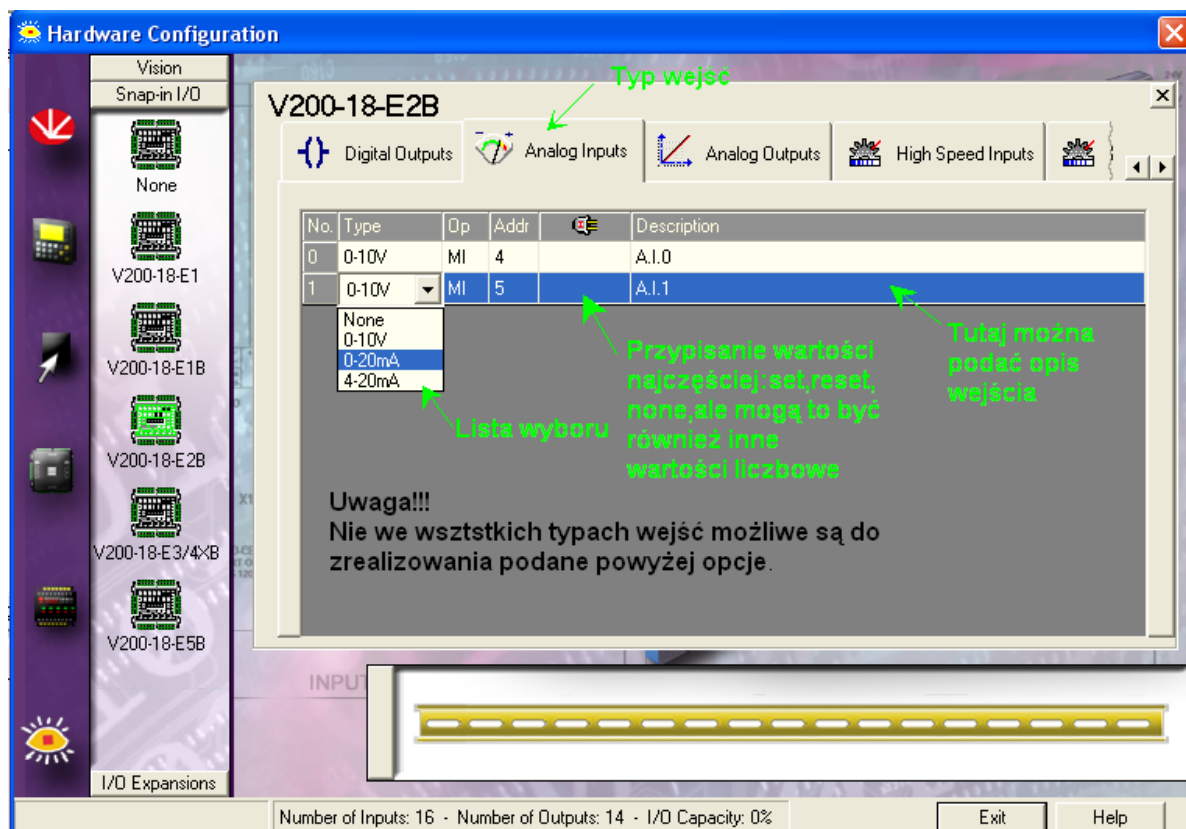
Na rysunkach 11, 12 przedstawiono okno konfiguracji osprzętu. Z lewej strony widnieją trzy zakładki: Vision, Snap-in I/O, I/O Expansion. Każdą z zakładek należy rozwinąć aby wybrać odpowiedni model urządzenia. Wyboru dokonuje się poprzez naciśnięcie lewym przyciskiem myszy na ikonę reprezentującą wybrany model. Tutaj także można ustawić tryb pracy wejść i wyjść oraz nadać im nazwy i przypisać określone wartości (Rys.12.). Wyjścia z panelu dokonujemy poprzez wciśnięcie przycisku Exit znajdujący się w prawym dolnym rogu. Zapisanie wybranych ustawień następuje automatycznie.

UWAGA!!!

Jeśli nie dokonano konfiguracji osprzętu nie będzie można załadować programu do sterownika.



Rys.11. Konfiguracja osprzętu













Rys. 12. Konfiguracja wejść/wyjść

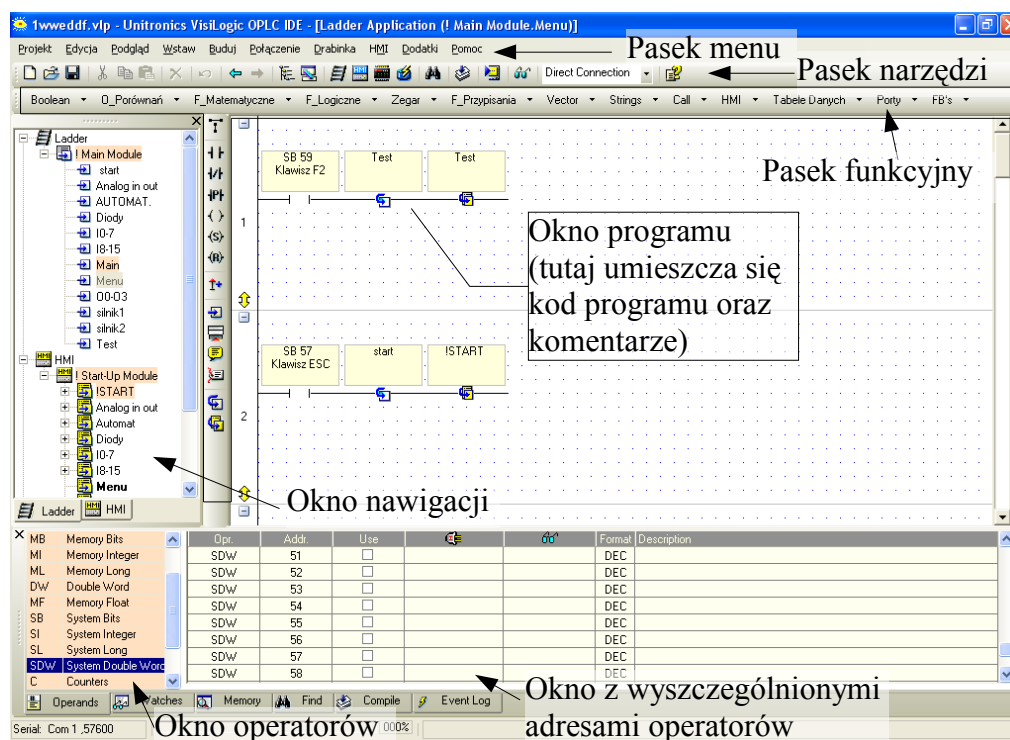
4.3.2. Aplikacja Ladder

Ladder to miejsce, gdzie tworzymy kod drabinki. Okno aplikacji Ladder pojawia się po uruchomieniu programu VisiLogic (Rys.13.).

Na pasku menu znajdziemy standardowe zakładki pomagające w konfiguracji środowiska programistycznego, zapisaniu projektu czy też uzyskaniu pomocy. Pasek narzędzi zawiera najbardziej potrzebne funkcje takie jak :

-  - Drabinka- powoduje przejście do aplikacji ladder
-  - HMI- powoduje przejście do aplikacji HMI
-  - Konfiguracja sterownika- umożliwia konfigurację osprzętu
-  - Tabela danych- służy do tworzenia tabel z danymi oraz ich konfiguracji
-  - Znajdź- pozwala znaleźć: zmienną, stałą, połączenia sieciowe, skoki, podprogramy lub ekrany.
-  - Kompiluj- służy do kompilacji stworzonego programu
-  - Prześlij program- powoduje załadowanie programu do sterownika
-  - On-Line Test- pozwala przetestować program, prześledzić jego działanie
-  - Połączenia- pozwala wybrać rodzaj połączeń dla sieci sterowników

-  - Pomoc- tutaj znajdują się szczegółowe informacje o VisiLogic i wszystkich dostępnych w oprogramowaniu opcjach



Rys. 13. Aplikacja Ladder

Okno nawigacji pokazuje nam poszczególne części programu tzn. podprogramy (Subroutine) oraz przyporządkowane im ekrany (Display). Zakładki w dolnej części okna służą do przechodzenia między Ladder a HMI.

Okno w dolnej części ekranu posiada szereg zakładek, które pozwalają na:

- przeglądanie zmiennych (zakładka Operands)
- wyników wyszukiwania (zakładka Watcher)
- sprawdzenie ile bitów zajmuje wybrana zmienna (zakładka Memory)
- podgląd wyników wyszukiwania (zakładka Find)
- przegląd raportów z kompilacji (zakładka Compile)
- sprawdzenie czasu kompilacji i wgrywania do sterownika programu (zakładka Event Log)

Pasek funkcyjny zawiera wszystkie dostępne w oprogramowaniu VisiLogic funkcje i elementy niezbędne do programowania w języku drabinkowym (Tab.3.- Tab.19.).

Tab.3.Operandy pamięci dostępne w VisiLogic

Symbol	Nazwa	Opis	Liczba	Adresy
MB	Memory Bits	Są to operandy, które przyjmują wartość 0 lub 1	4096	MB 0- MB 4095
I	Inputs	To operandy reprezentujące wejścia, przyjmują wartość 0 lub 1	Ich liczba zależy od podłączonych modułów I/O	np. V200- 180-E2B: I0- I15
O	Outputs	operandy reprezentujące wyjścia, przyjmują wartość 0 lub 1	Ich liczba zależy od podłączonych modułów I/O	np. V200- 180-E2B: O0- O13
MI	Memory Integers	Jest 16-bitową liczbą całkowitą z lub bez znaku. Jej zakres to -32768 do +32767	2048	MI0- MI2047
ML	Memory Long Integer	Jest 32-bitową liczbą całkowitą z lub bez znaku. Jej zakres: -2147483648 do +2147483647	256	ML0- ML255
DW	Double Word	Jest to 32-bitowa liczba całkowita bez znaku. Jej maksymalna wartość to: 4294967296	64	DW0- DW63
MF	Memory Floating Point Integer	Jest 32-bitową liczbą całkowitą z lub bez znaku. Jej zakres: -3,402823E38 do -1,401298E-45 dla liczb ujemnych i + 1,401298E do +3,402823E38 dla liczb dodatnich	24	MF0- MF23

Tab.4.Operandy systemowe dostępne w VisiLogic

Symbol	Nazwa	Opis		Liczba	Adresy
SI	System Integer	Wiele z tych operandów posiada opcje które są wykorzystywane przez układ sterujący sterownika. Ich lista i dokładny opis znajduje się w pomocy oprogramowania VisiLogic	Jest 32-bitową liczbą całkowitą z lub bez znaku. Jej zakres: -32768 do +32767	512	SI 0- SI511
SB	System Bits		Operator przyjmuje wartość 0 lub 1	512	SB0- SB511
SL	System Long		Jest 16-bitową liczbą całkowitą z lub bez znaku.	56	SB0- SB55
SDW	System Double Word		Jest to 32-bitowa liczba całkowita bez znaku. Jej maksymalna wartość to: 4294967296	64	SDW0- SDW63

Tab.5.operandy sieci dostępne w VisiLogic










Symbol	Nazwa	Opis	Liczba	Adresy
NSB	Network System Bit	Są to zmienne umożliwiające komunikację sterowników w sieci	8	SB200-SB207
NI	Network Input	tzn.: wysyłanie i odbieranie instrukcji czy definiowanie nr ID sterownika.	17	I0-I16
NSI	Network System Integer	Szczegóły znajdują się w pomocy oprogramowania VisiLogic	2	SI200-SI201

W VisiLogic dostępne są także stałe oznaczone przez symbol #.

Tab.6.Liczniki dostępne w VisiLogic








Symbol	Nazwa	Opis	Liczba	Adresy
T	Timers	<p>Są trzy rodzaje liczników czasowych:</p> <p>-TD- Licznik opóźniający. Zlicza zawsze od zadanej wartości do 0. Podczas zliczania na wyjściu podaje stan niski, po skończonej pracy na wyjściu pojawia się stan wysoki</p> <p>- TA- Licznik sumaryczny. Zlicza od zadanej wartości do zera, jednak gdy zostanie wstrzymany podczas zliczania a później ponownie uruchomiony odlicza od wartości, na której się zatrzymał. Podczas zliczania na wyjściu podaje stan niski, po skończonej pracy na wyjściu pojawia się stan wysoki</p> <p>- TE- Licznik o przedłużonym impulsie. Zlicza od zadanej wartości do zera, jednak gdy zostanie wstrzymany podczas zliczania a później ponownie uruchomiony dodaje wartość, która mu pozostała z poprzedniego odliczania do wartości początkowej i ponownie zaczyna odliczać. Podczas zliczania na wyjściu podaje stan wysoki, po skończonej pracy na wyjściu pojawia się stan niski</p> <p>Liczniki inicjalizowane są stanem niskim.</p> <p>Każdy z nich posiada trzy zmienne:</p> <ul style="list-style-type: none"> - wartość bitu licznika - bieżąca wartość licznika (od której zaczyna się zliczanie w dół) - aktualna wartość stanu licznika 	192	T0- T191
C	Counters	<p>Jest to licznik cyfrowy w połączeniu z funkcją inkrementacji lub dekrementacji może zliczać w dół od zadanej wartości do zera lub w górę od zera do zadanej wartości. Podczas zliczania na wyjściu podaje stan wysoki, po skończonej pracy na wyjściu pojawia się stan niski.</p>	24	C0-C23

Tab.7.Funkcje Boolean¹

Ikona	Nazwa	Opis	Dostępne operandy
	Styk otwarty	Jest to styk normalnie otwarty (ON). Obsługuje jeden operand.	MS, SB, N, NSB, O, T
	Styk zamknięty	Jest to styk normalnie zamknięty (OFF). Obsługuje jeden operand.	
	Styk o zboczu opadającym	Styk daje stan niski przy jednorazowym impulsie kiedy operand przechodzi ze stanu 1 do 0, a wysoki przy przejściu operandu ze stanu 1 do 0.	
	Styk o zboczu narastającym	Styk daje stan wysoki przy jednorazowym impulsie kiedy operand przechodzi ze stanu 0 do 1, a niski przy przejściu operandu ze stanu 1 do 0.	MB, SB, O, T, C
	Cewka otwarta	Cewka przyjmuje stan 1 gdy poprzedzająca ją sieć wysyła do niej stan 1, a 0 gdy sieć ma stan 0.	MB, SB, O, T
	Cewka zamknięta	Cewka zanegowana przyjmuje stan 0 gdy poprzedzająca ją sieć wysyła do niej stan 1, a 1 gdy sieć ma stan 0.	
	Cewka set	Cewka set po otrzymaniu sygnału 1 z poprzedzającej ją sieci ustawia się w stan 1 i pozostaje w tym stanie dopóki nie zostanie zresetowana poprzez podanie sygnału 1 na cewkę reset o takim samym adresie.	MB, SB, O
	Cewka reset	Cewka reset po otrzymaniu sygnału 1 z poprzedzającej ją sieci ustawia się w stan 0 i pozostaje w tym stanie dopóki nie zostanie ustawiona w stan 1 poprzez podanie sygnału 1 na cewkę set o takim samym adresie.	MB, SB, O, T
	Przełącznik	Cewka ta ustawia się w stan 1 po jednorazowym otrzymaniu 1, w stan niski przechodzi gdy zostanie ponownie podany stan wysoki na wejście cewki	MB, O













¹ Funkcje zawarte w tabeli obsługują po jednym operandzie naraz.

Tab.8.Funkcje Porównań¹

Ikona	Nazwa	Opis	Dostępne operandy
	wiekszy	Funkcja porównuje ze sobą dwie zmienne wejściowe A i B. Jeżeli A będzie mniejsze lub równe B to na wyjściu wystąpi stan niski. Jeżeli warunek będzie spełniony to na wyjściu pojawi się stan wysoki.	MI, ML, DW, SI, SL, SDW, NSI, #, C
	wiekszy równy	Funkcja porównuje ze sobą dwie zmienne wejściowe A i B. Jeżeli A będzie mniejsze niż B to na wyjściu wystąpi stan niski. Jeżeli warunek będzie spełniony to na wyjściu pojawi się stan wysoki.	
	równy	Funkcja porównuje ze sobą dwie zmienne wejściowe A i B. Jeżeli A będzie równe B to na wyjściu wystąpi stan wysoki, w przeciwnym wypadku pojawi się stan niski.	
	różny	Funkcja porównuje ze sobą dwie zmienne wejściowe A i B. Jeżeli A będzie różne B to na wyjściu wystąpi stan wysoki, w przeciwnym wypadku pojawi się stan niski.	
	mniejszy równy	Funkcja porównuje ze sobą dwie zmienne wejściowe A i B. Jeżeli A będzie większe niż B to na wyjściu wystąpi stan niski. Jeżeli warunek będzie spełniony to na wyjściu pojawi się stan wysoki.	
	mniejszy	Funkcja porównuje ze sobą dwie zmienne wejściowe A i B. Jeżeli A będzie większe lub równe B to na wyjściu wystąpi stan niski. Jeżeli warunek będzie spełniony to na wyjściu pojawi się stan wysoki.	
	przedział	Funkcja sprawdza czy badana przez nas zmienna znajduje się w zadanym przedziale. Jeśli tak jest to na wyjściu pojawi się stan 1 w przeciwnym wypadku będzie to stan 0.	

¹ Każda z funkcji inicjalizowana jest stanem wysokim podanym na jej wejście.

Tab.9.Funkcje Matematyczne¹








Ikona	Nazwa	Opis	Dostępne operandy
	Inkrementacja	Funkcja inkrementuje zadaną wartość o 1.	MI, SI, ML, SL, C, DW, SDW
	Dekrementacja	Funkcja dekrementuje zadaną wartość o 1.	
	Wzór	Funkcja pozwala na zdefiniowanie przez użytkownika dowolnej formuły matematycznej.	MI, SI, ML, SL, MF
	Linearyzacja	Funkcja konwertuje wartości np. odczyt z wejścia analogowego na °C.	MI, ML, DW, SI, SL, SDW
	Dodawanie	Blok pozwala na dodanie do siebie od dwóch do ośmiu zmiennych. Wynik wpisywany jest do osobnego adresu.	MI, ML, DW, SI, SDW, NSI, #
	Odejmowanie	Blok pozwala na odejmowanie od siebie od dwóch do ośmiu zmiennych. Wynik wpisywany jest do osobnego adresu.	
	Mnożenie	Funkcja mnoży ze sobą od dwóch do ośmiu zmiennych. Wynik wpisywany jest do osobnego adresu.	
	Dzielenie	Funkcja daje możliwość podzielenie przez siebie od dwóch do ośmiu zmiennych. Wynik wpisywany jest do osobnego adresu.	
	Mod	Blok wykonuje funkcje modulo na dwóch zmiennych. Wynik wpisywany jest do osobnego adresu.	
	Pierwiastek kwadratowy	Funkcja liczy pierwiastek kwadratowy z podanej zmiennej. Wynik wpisywany jest do osobnego adresu.	
	Potęga	Funkcja pozwala podnosić zmienne do dowolnej potęgi, używa dwóch zmiennych A jako podstawy potęgi i B jako wykładnika. Wynik wpisywany jest do osobnego adresu.	
	Współczynnik	Funkcja wykorzystuje trzy zmienne: A, B, C..Dzieli A przez B i wynik dzielenia mnoży przez C. Wynik zapisywany jest do osobnego adresu.	

¹ Każda z funkcji inicjalizowana jest stanem wysokim podanym na jej wejście.









Tab.10.Funkcje Matematyczne – zmienne typu Float

Ikona	Nazwa	Opis
Float	Float	<p>Zmienne typu float są to 32-bitowe liczby zmiennoprzecinkowe, mogą występować z lub bez znaku. Ich zakres to: -3.402823E38 do -1.401298E-45 dla liczb ujemnych i +1.401298E-45 do +3.402823E38 dla liczb dodatnich. W VisiLogic dostępne są 24 adresy przeznaczone na typ float: MF 0 – MF23. Są to szczególne zmienne, dlatego występują osobne funkcje obsługujące ten typ danych. Należą do nich:</p> <ul style="list-style-type: none"> ● Funkcje podstawowe (Basic) czyli: przeniesienie bezpośrednio (tzn. przepisanie wartości z jednego adresu do drugiego np. z MF1 do MF5), dodawanie, odejmowanie, mnożenie, dzielenie, wartość bezwzględna (Abs) ● Funkcje rozszerzone (Extended): pierwiastek, potęgowanie, eksponent, logarytm, logarytm dziesiętny, oraz wyrażenie $A \cdot (10^B)$ ● Funkcje trygonometryczne (Trig): sinus, cosinus, tangens, arcus cosinus, arcus sinus, arcus tangens, stopnie Celsjusza (zamiana wartości float na °C), radiany (zamiana wartości float na radiany) ● Funkcje porównujące (Compare): większe, mniejsze, większe równe, mniejsze równe, równe , różne, przedział ● Funkcje konwertujące (Convert): A+B/n- konwertuje dwie zmienne (nie będące typu float) na typ float, INV(A+B/n) rozбивa zmienną float na dwie zmienne nie będące typu float.

Tab.11.Zegar










Ikona	Nazwa	Opis
<p>Funkcje zegara: bezpośrednie/ pośrednie</p> <p>a) </p> <p>b) </p> <p>c) </p> <p>d) </p> <p>e) </p>	<p>a) godzina</p> <p>b) dzień tygodnia</p> <p>c) dzień miesiąca</p> <p>d) miesiąc</p> <p>e) rok</p>	<p>Funkcje pozwalają ustawić przedział czasu, w którym na wyjściu ma pojawiać się stan wysoki. Przedziałem mogą być: godziny, dni tygodnia, miesiące a nawet lata.</p> <p>Bezpośrednie funkcje zegara to funkcje, w których czas i data ustawiane są wewnątrz tych funkcji, natomiast pośrednie funkcje zegara wymagają podpięcia rejestrów (MI, ML, SI, SL) i wpisania do nich wartości poprzez program użytkownika lub z klawiatury sterownika.</p>
<p>UTC:</p> <p>a) </p> <p>b) </p>	<p>a)UTC to RTC</p> <p>b)RTC to UTC</p>	<p>a) Funkcja ta przekształca 32-bitową liczbę całkowitą bez znaku (DW) na czas rzeczywisty(RTC). Wartość ta zostaje przesłana do rejestru SI30 ustawiając automatycznie RTC sterownika i nadpisując rejestry SI30- SI34.</p> <p>b) Funkcja pozwala przekształcić wartość RTC na 32-bitową liczbę całkowitą bez znaku (DW) poprzez wykorzystanie rejestru SI30.</p>

Tab.12.Funkcje Logiczne




Ikona	Nazwa	Opis	Dostępne operandy																													
	Test Bit	Funkcja pozwala na wybranie bitu z wektora rejestru i zapisanie jego pozycji w MB.	MI, SI, ML, SL, DW, SDW, NSI, #																													
	Ustaw Bit	Pozwala wybrać bit z wektora rejestru i ustawić go na 1.																														
	Resetuj Bit	Pozwala wybrać bit z wektora rejestru i ustawić go na 0.																														
	AND	Funkcja AND bada stan logiczny dwóch zmiennych. Wynik ma wartość 1 gdy obie zmienne są prawdziwe, w innym przypadku otrzymujemy 0.																														
	OR	Funkcja OR bada stan logiczny dwóch zmiennych aby zobaczyć czy którakolwiek z nich ma stan wysoki. Jeśli choć jedna zmienna ma stan wysoki to otrzymujemy 1.																														
	XOR	Funkcja XOR bada stan logiczny dwóch zmiennych aby zobaczyć czy są one sobie równe. Jeżeli zmienne mają takie same stany to otrzymujemy 0.																														
Shift	Shift	Funkcja pozwala na przesunięcie wybranego bitu w lewo lub w prawo, o zadany wektor. Uwaga: Każdy przesunięty bit nie może przywrócony do poprzedniego położenia.																														
Obrót	Obrót	Funkcja przesuwa określoną liczbę bitów rejestru w lewo lub w prawo.																														
	RS Flip Flop	To funkcje działające jak przerzutniki. Badają one stan logiczny dwóch zmiennych i postępują zgodnie z poniższymi tabelami ¹ :																														
	SR Flip Flop																															
		<table><tr><th colspan="3">RS</th><th colspan="3">SR</th></tr><tr><th>R (A)</th><th>S (B)</th><th>Q</th><th>S (A)</th><th>R (B)</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>No change</td><td>0</td><td>0</td><td>No change</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	RS			SR			R (A)	S (B)	Q	S (A)	R (B)	Q	0	0	No change	0	0	No change	0	1	1	0	1	0	1	0	0	1	1	1
RS			SR																													
R (A)	S (B)	Q	S (A)	R (B)	Q																											
0	0	No change	0	0	No change																											
0	1	1	0	1	0																											
1	0	0	1	1	1																											
RLO to Bit	RLO to Bit	RLO to Bit bierze stan RLO (Result of Logical Operation) i gromadzi go w bicie danego rejestru z zadany przesunięciem.																														

¹ Tabele zostały zaczerpnięte z pomocy programu VisiLogic















Tab.13.Funkcje przypisania

Ikona	Nazwa	Opis	Dostępne operandy
	Przypisanie bezpośrednie	Pozwala na przepisanie wartości argumentu (lub stałą) do innego argumentu.	MI, SI, ML, SL, DW, SDW, #
	Przypisanie pośrednie	Pozwala na przepisanie wartości zawartej w pewnych typach argumentów do innego argumentu poprzez adresowanie pośrednie.	
	Pobranie pośrednie	Pozwala na pobranie wartości argumentu źródłowego i załadowanie jej do innego argumentu poprzez adresowanie pośrednie.	
	Przypisz do licznika czasowego/cyfrowego bieżącą wartość	Pozwala na wpisanie wartości jakiegoś argumentu (nie będącego C lub T) do licznika czasowego lub cyfrowego.	MI, SI, ML, SL, DW, SDW, #, C, T
	Pobierz z licznika czasowego/cyfrowego bieżącą wartość	Pozwala na wpisanie do argumentu (nie będącego C lub T) bieżącej wartości licznika czasowego/cyfrowego.	
	Przypisz do licznika czasowego/cyfrowego aktualną wartość	Pozwala na wpisanie aktualnej wartości jakiegoś argumentu (nie będącego C lub T) do licznika czasowego/cyfrowego.	
	Pobierz z licznika czasowego/cyfrowego aktualną wartość	Pozwala na wpisanie do argumentu (nie będącego C lub T) aktualnej wartości licznika czasowego/cyfrowego.	
	Num to BCD	Pozwala na zamianę wartości numerycznej na BCD.	MI, SI, ML, SL, DW, SDW, #
	BCD to Num	Pozwala na zamianę BCD na wartość numeryczną.	








Tab.14.Call

Ikona	Nazwa	Opis
	Skok do etykiety	Wskazuje miejsce, w którym program ma przeskoczyć do wybranej etykiety
	Wykonaj podprogram	Wymusza uruchomienie wybranego podprogramu w zadanym punkcie drabinki
	Podprogram: powrót	Wskazuje miejsce, w którym podprogram kończy pracę i następuje skok do początku poprzedzającego go podprogramu










Tab.15.Funkcje wektorowe (vector)

Ikona	Nazwa	Opis	Dostępne operandy
	Load	Pozwala wziąć wartość (z określonego przez użytkownika zakresu bitów) operandu źródłowego i umieścić ją w wybranym argumencie.	MB,SB, MI, SI, ML, SL, I, O
	Find	Pozwala na wyszukiwanie wartości.	
	Store	Pozwala wziąć wartość (z określonego przez użytkownika zakresu bitów) operandu źródłowego i przechować ją w wybranym argumencie.	MB, MI, ML, O, T, C, DW
	Fill	Funkcja wypełnienia pozwala na wpisanie wybranej wartości do każdego operatora znajdującego się w obrębie podanego wektora.	MI, ML, DW
	Copy	Funkcja kopiowania umożliwia: kopiowanie wartości lub pozycji każdego operatora znajdującego się w zadanym wektorze oraz zapisanie tych wartości lub pozycji do odpowiedniego wektora operandów tej samej długości.	MB,SB,M I, SI, ML, SL, I, O
	Compare	Pozwala na porównanie wartości lub pozycji bitu dwóch operandów w zadanym przedziale.	
	Shift	Pozwala na przesunięcie bitów operatora o zadany wektor.	MI, ML, DW, #
	Swap bytes	Pozwala przestawiać bajty operatora w obrębie zadanego wektora.	
Use Offset	Use Offset	W skład tej zakładki wchodzi funkcje: fill, copy, compare i copy memory rozbudowane o przesunięcie.	MB,SB,M I, SI, ML, SL, I, O,
	Bits to numeric	Pozwala na zamianę zbioru bitów na wartość numeryczna.	
	Numeric to bits	Pozwala na zamianę wartości numerycznej na zbiór bitów.	
	Get min	Funkcja szuka najmniejszej wartości w operatorze.	MI, SI, ML, SL, DW, SDW
	Get max	Funkcja szuka największej wartości w operatorze.	
	Sort	Porządkuje bity operandu malejąco lub rosnąco.	MB, MI, ML, MF, I, O, DW
	Struct	Pozwala pobrać dane z wektora pamięci i umieścić je w zadanych komórkach pamięci i na odwrót.	















Tab.16.Strings

Ikona	Nazwa	Opis	Dostępne operandy
	Num to ASCII	Pozwana zapisać wartość numeryczną w kodzie ASCII.	SI, MI, ML, SL, MF, DW, SDW, #
	ASCII to num	Pozwana zamienić kod ASCII na wartość numeryczną.	
	RTC to ASCII	Pozwala zamienić RTC na stringu ASCII.	MI, ML, DW, #
	Time to ASCII	Pozwala zamienić wartość licznika na stringu ASCII .	SI, MI, ML, SL, MF, DW, SDW, #
	IP to ASCII	Pozwala zamienić adres IP na stringu ASCII .	MI, ML, DW,
	Mac address to ASCII	Umożliwia zapisanie adresu MAC do pamięci w postaci ASCII string.	
	Transpose	Umożliwia “kompresję” operatora MI do mniej znaczących bajtów wybranego operatora MI lub “rozszerzenie” bajtów do wybranego operatora MI.	




















Tab.17.HMI

Ikona	Nazwa	Opis	Dostępne operandy
	Załaduj ekran HMI	Wskazuje miejsce, w którym ma być uruchomiony wybrany ekran HMI.	
	Załaduj ekran HMI	Ekran HMI ładowany jest gdy przypisany mu operand przyjmuje wartość 1 (ON).	MB
	Załaduj ostatni ekran HMI	Funkcja w wybranym miejscu drabinki spowoduje załadowanie ostatniego ekranu HMI.	
	Rysuj pixel	Daje możliwość umieszczenia zabarwionego pixela na ekranie LCD w punkcie (X,Y).	MI, ML, DW, SB, MB, #
	Rysuj linię	Daje możliwość umieszczenia linii o różnej grubości i formacie na ekranie LCD.	
	Czyść obszar (Clear Rectangle)	Daje możliwość wyczyszczenia prostokątnego obszaru na ekranie LCD z poziomu drabinki.	SI, MI, ML, SL, MF, DW, SDW, #
	Inverse vars	Funkcja odwraca kolory wybranej zmiennej.	
	Hide vars	Funkcja powoduje ukrycie wybranej zmiennej.	
	Previous entry variable	Funkcja ta powoduje przywołanie poprzednich zmiennych poprzez naciśnięcie klawisza z klawiatury sterownika	

Tab.18.Tablice danych

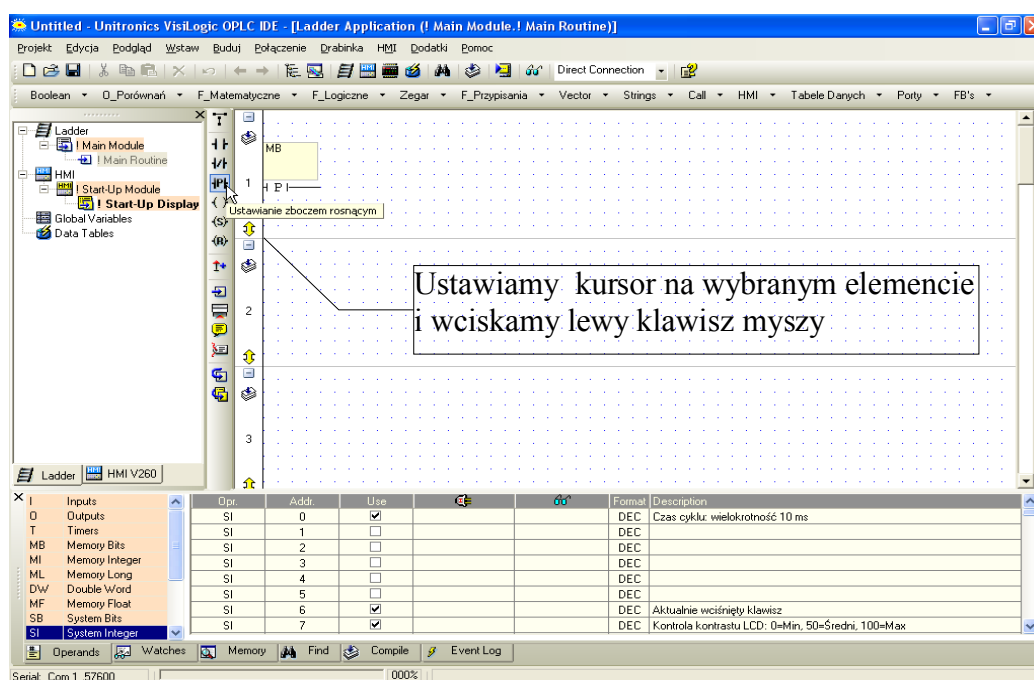
Ikona	Nazwa	Opis	Dostępne operandy
	Data table	Pozwala na umieszczenie danych w tablicy.	Wszystkie dostępne w VisiLogic operandy
	Data table read row	Pozwala wczytać wartość z określonego wiersza tabeli do wybranego operandu.	
	Data table write row	Pozwala wpisać dane z wybranego operandu do określonego wiersza tablicy danych	
	Data table read column	Pozwala wczytać wartość z określonej kolumny tabeli do wybranego operandu.	
	Data table write column	Pozwala wpisać dane z wybranego operandu do określonej kolumny tablicy danych.	
	Data table: clear table	Usuwa wszystkie dane z wybranej tablicy.	
	Data table: clear row	Usuwa wszystkie dane z wybranego wiersza tablicy danych.	ML, MI, DW, #
	Data table: clear column	Usuwa wszystkie dane z wybranej kolumny tablicy danych.	
	Data table: find	Porównuje wielkość wejściową z danymi zawartymi w tablicy danych. Jeśli wielkość zostanie znaleziona, to numer rzędu zawierającego tą wartość zostanie wpisany do wyjścia bloku funkcyjnego.	ML, MI, DW, #
	Data table: find (Extended)	Porównuje kilka wielkości wejściowych z danymi zawartymi w tablicy danych. Jeśli wielkości zostaną znalezione to numer rzędu w którym są zawarte zostanie wpisany do wyjścia bloku funkcyjnego.	
	Data table: copy row	Funkcja pozwala na kopiowanie danych z wybranego wiersza wskazanej przez użytkownika tabeli danych do innego wiersza w tej samej tabeli lub do określonego wiersza innej tabeli danych.	
	Data table: copy column	Funkcja pozwala na kopiowanie danych z wybranej kolumny wskazanej przez użytkownika tabeli danych do innej kolumny w tej samej tabeli lub do określonej kolumny innej tabeli danych.	
	Direct: read	Kopiuje dane z tablicy danych do wektora rejestrów w sterowniku	MI, SI, ML, SL, DW, SDW, #
	Direct: write	Kopiuje dane z wektora rejestrów do tablicy danych	

Tab.19.FB's

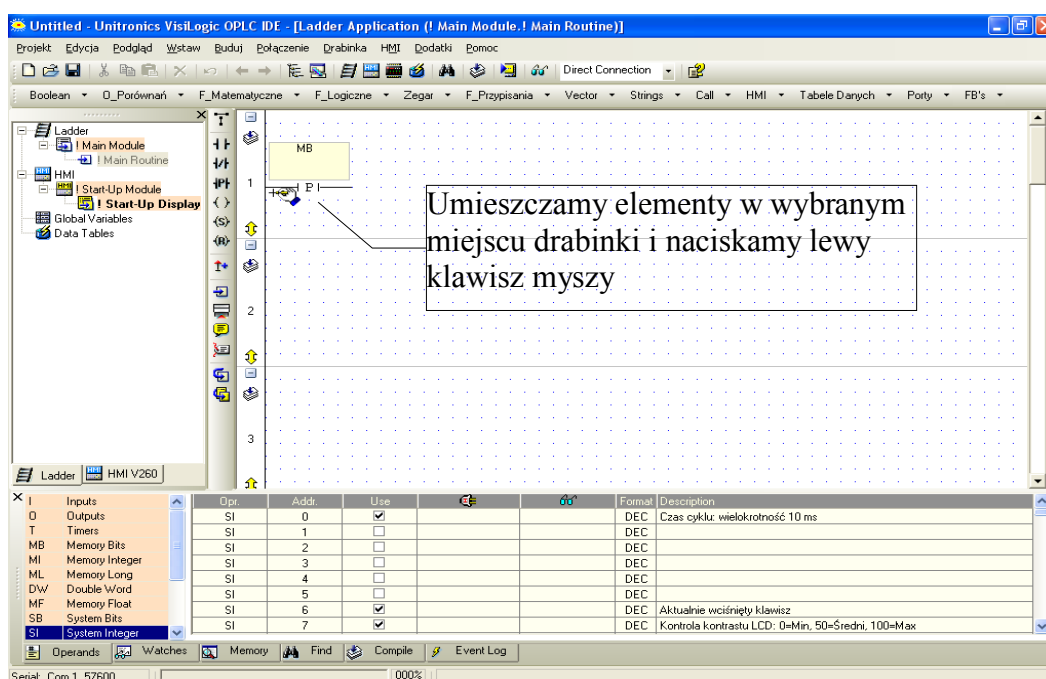
Ikona	Nazwa	Opis	Operandy
	Events	Monitoruje zmiany stanów operandu MB z 0 na 1. Funkcja może być użyta do monitorowania stanu bitów w tablicy alarmów.	MB, MI
	Trend	Umożliwia pokazanie wektora dynamicznie zmieniającej się wielkości na ekranie sterownika. Wielkości wejściowe mogą pochodzić z tablicy danych lub z liczników	ML, MI, DW, #, MB
	Draw Axis	Pozwala na wstawienie osi x i y na ekranie sterownika	ML, MI, DW, #
	PDI Auto-Tune	Funkcja umożliwia użycie sprzężenia zwrotnego by kontrolować dynamiczny proces. Funkcja używa Auto-tune (automatycznej regulacji).	MI, MB
	PDI	Funkcja umożliwia użycie sprzężenia zwrotnego by kontrolować dynamiczny proces. Funkcja nie używa Auto-tune.	
	Accelerate	Powoduje inkrementację/dekrementację wartości operandu	ML, MI, DW, #
	PWM	Reguluje wypełnienie impulsu sygnału taktującego	ML, MI, DW, MB
	Loadcell	Funkcja pozwala wykryć podłączone do sterownika moduły rozszerzeń I/O i skonfigurować je.	
	Fast Response: Expansion Module	Funkcja używana jest gdy mamy moduły rozszerzeń I/O. Pozwala ona na szybka reakcję modułu na zmianę stanu jego wejść/wyjść.	
	GPRS	Umożliwia konfigurację połączenia PLC z modemem GPRS.	
	BAS	Umożliwia wymianę danych między sterownikiem Vision a CSI.	
	Filter	Umożliwia obliczenie średniej z czterech do szesnastu wartości.	
	Remote PLC DataCom	Uaktywnia UDP podczas komunikacji poprzez sieć Ethernet	
	Drum Sequencer	Umożliwia sekwencyjną pracę operandów.	Wszystkie operandy
	MOD BUS	Pozwala nawiązać komunikację między sterownikami PLC lub między sterownikiem a urządzeniem wspierającym MODBUS. Urządzenia muszą być połączone poprzez port COM.	
	MOD BUS IP	Pozwala nawiązać komunikację między sterownikiem a urządzeniem wspierającym MODBUS. Urządzenia muszą być połączone poprzez sieć Ethernet.	
	SMS	Umożliwia odbieranie i wysyłanie wiadomości SMS. Sterownik musi być podłączony do modemu.	
	Protocol	Umożliwia wymianę danych z urządzeniami zewnętrznymi poprzez port COM.	
	Protocol TCP/IP	Umożliwia wymianę danych (z TCP/IP) z urządzeniami zewnętrznymi poprzez sieć Ethernet.	

Więcej informacji na temat zmiennych i funkcji zawartych w VisiLogic można znaleźć w pomocy oprogramowania.

Elementy znajdujące się na pasku funkcyjnym wstawiamy do drabinki poprzez naciśnięcie na ikonkę lewym przyciskiem myszy. Następnie należy wskazać lewym przyciskiem myszy (w obszarze okna drabinki) miejsce w którym ma się znaleźć wybrana funkcja (Rys.13. i Rys.14.)

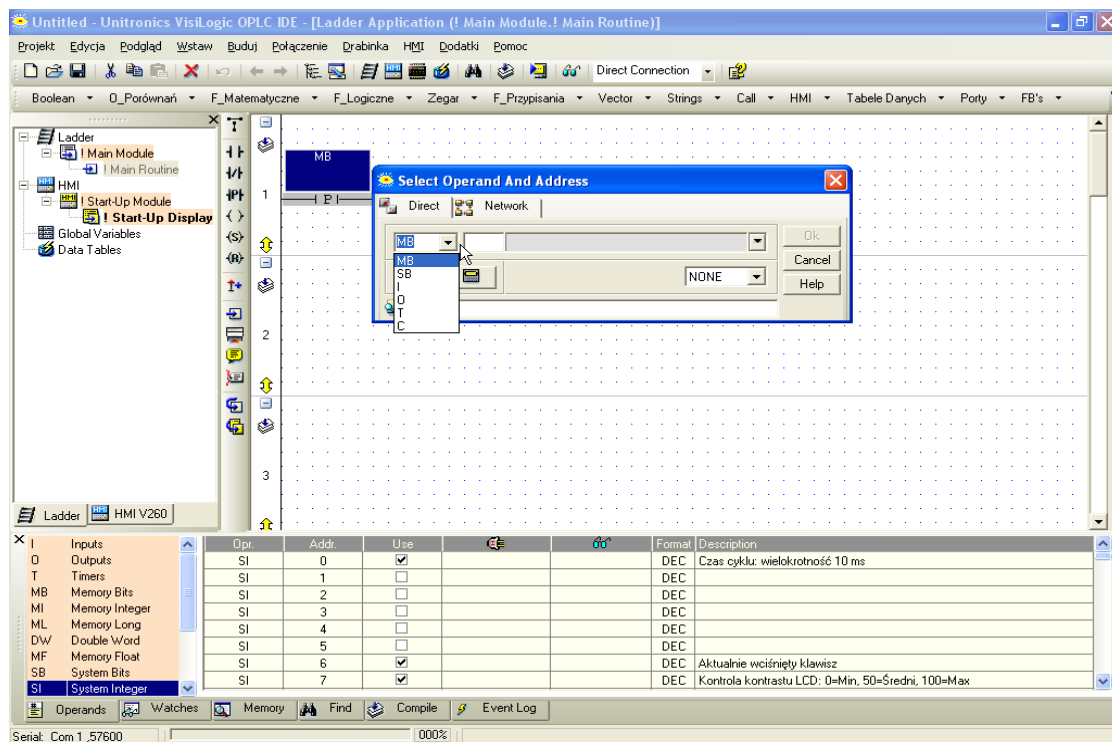


Rys.13.Wstawianie elementów cz.1

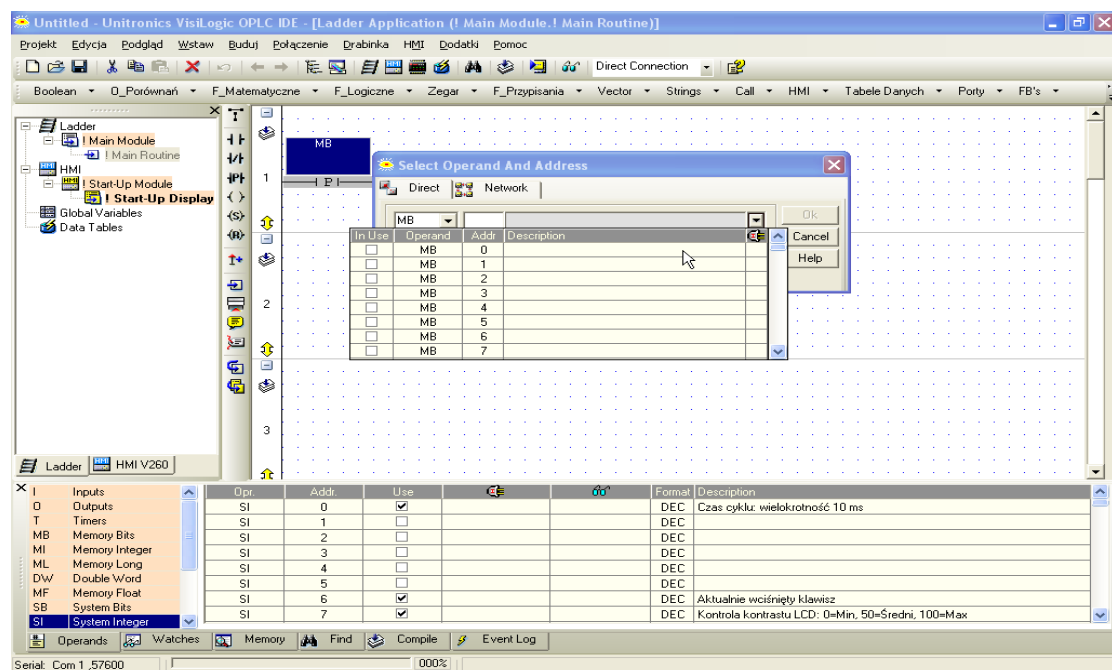


Rys.14.Wstawianie elementów cz.2

Po wstawieniu elementu pojawi się okno wyboru zmiennych (Rys.15). W celu wybrania operandu należy rozwinąć listę i nacisnąć lewym przyciskiem myszy. Podobnie należy postąpić przy wyborze adresu operatora (Rys.16)



Rys.15.Wybór operatora.



Rys.16.Wybór adresu operatora.

Elementy drabinki możemy:

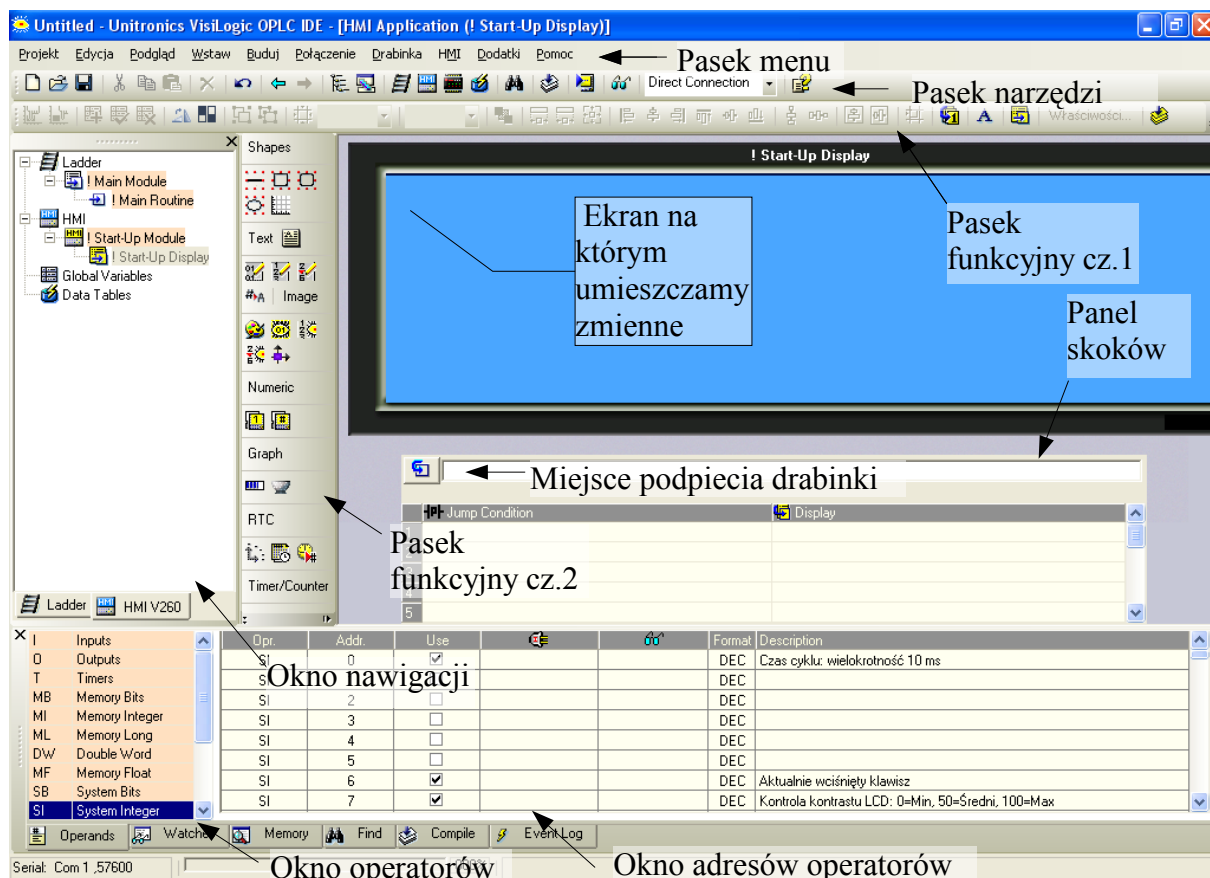
- usuwać poprzez zaznaczenie ich i wciśnięcie klawisza Del lub wybranie opcji usuń z menu podręcznego
- kopiować poprzez zaznaczenie ich i wybranie opcji kopiuj z menu podręcznego
- wycinać poprzez zaznaczenie ich i wybranie opcji wtnij z menu podręcznego
- zamieniać na elementy o podobnych właściwościach funkcyjnych poprzez zaznaczenie ich i wybranie opcji replace Ladder element z menu podręcznego

4.3.3. Aplikacja HMI

Aplikacja HMI służy do tworzenia ekranów wyświetlanych na LCD (Rys.17.). W górnej części okna znajduje się pasek menu i pasek narzędzi. Pełnią one takie same funkcje jak w aplikacji Ladder. Pasek funkcyjny podzielony został na dwie części. W pierwszej umieszczone są:

- funkcje pozwalające regulować położenie zmiennych na ekranie
- funkcja umożliwiająca kompilację ekranu (Kompilacja ekranu)
- funkcja ustawiająca kolejność zmiennych wpisywanych z klawiatury (Variable Tab Order)
- funkcja pozwalająca na zmianę ustawienia czcionek (Ustawienia czcionek)
- Funkcja pozwalającą na dodanie nowego ekranu (Dodaj nowy ekran)

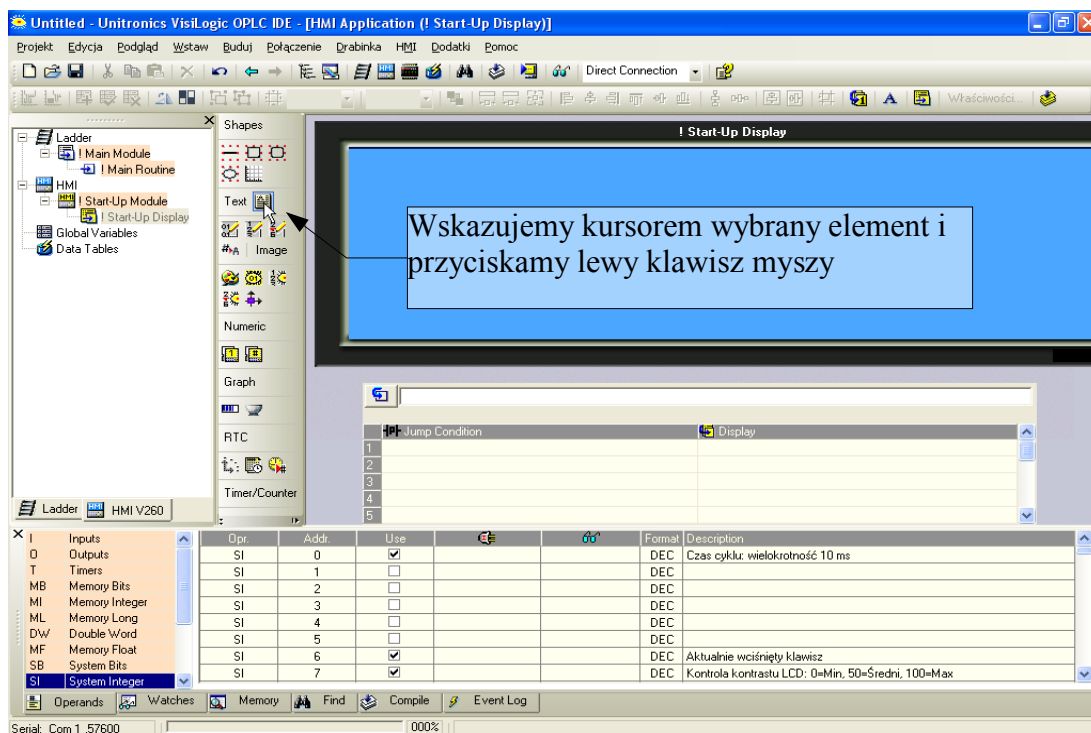
Druga część zawiera wszystkie dostępne funkcje HMI (Tab.20. - Tab.26.). Okna operandów i ich adresów znajdują się w dolnej części aplikacji i pełnią takie same funkcje jak w aplikacji Ladder. W panelu skoków definiujemy miejsca do których ma skoczyć program gdy przypisana do skoku zmienna przyjmie wartość 1 np. skok do panelu x po naciśnięciu klawisza F5.



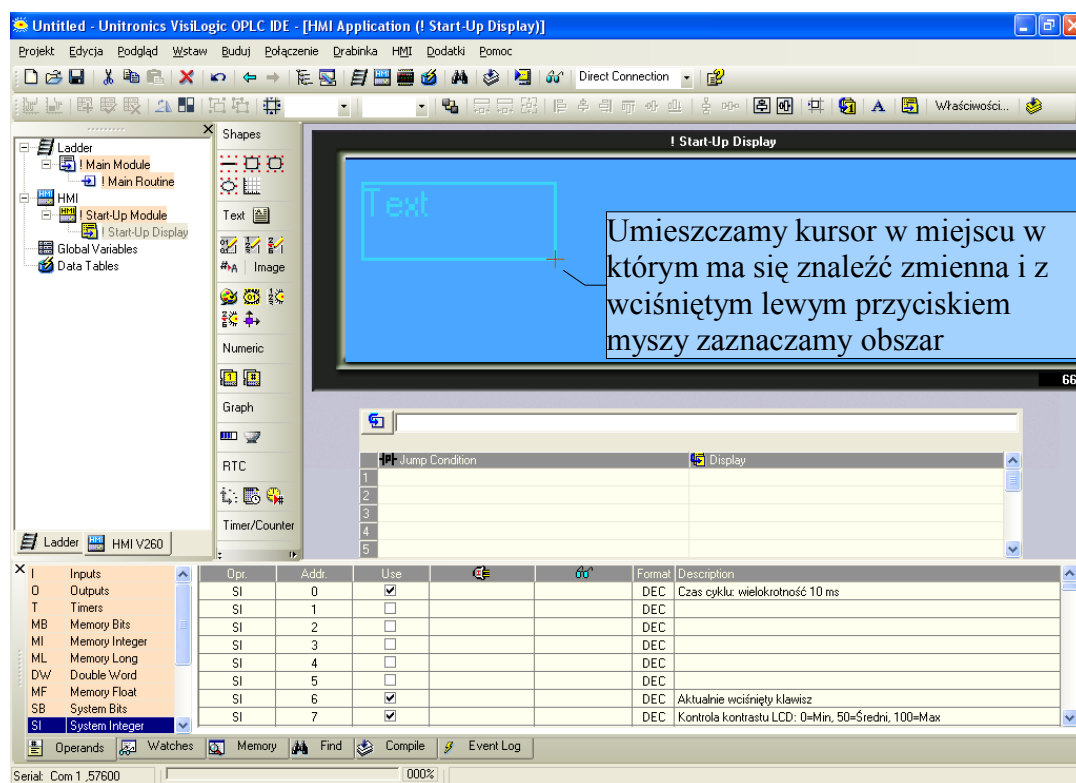
Rys.17. Aplikacja HMI.

Wstawianie zmiennych na ekranie odbywa się w podobny sposób jak umieszczanie elementów w drabince. Na początku należy najechać kursorem na wybraną funkcję i przycisnąć lewy klawisz myszy (Rys.18). Następnie lewym przyciskiem myszy należy zaznaczyć obszar, w którym ma znajdować się zmienna (Rys.19). Niektóre funkcje wymagają podania operandów i ich adresów, czynność tą wykonuje się identycznie jak w przypadku funkcji LD. Inne wymagają wpisania tekstu lub podpięcia bitmapy w wybranego przez użytkownika folderu. Wszystkie zmienne możemy usuwać, kopiować, wycinać, czy też przypisywać je do zmiennych globalnych¹ poprzez wybranie odpowiedniej funkcji z menu podręcznego.

¹ Zmienna globalna to zmienna odnosząca się do wszystkich stworzonych ekranów HMI.








Rys.18. Wstawianie elementów cz.1








Rys.19. Wstawianie elementów cz.2






Tab.20.Shapes

Ikona	Nazwa	Opis
	Linia	Funkcja umożliwia rysowanie linii na ekranie HMI.
	Prostokąt	Funkcja umożliwia rysowanie prostokątów na ekranie HMI.
	Zaokrąglony prostokąt	Funkcja umożliwia rysowanie prostokątów o zaokrąglonych krawędziach na ekranie HMI.
	Elipsa	Funkcja umożliwia rysowanie elipsy na ekranie HMI.
	Create static Axis	Funkcja umożliwia wstawieniu osi dla wykresów na ekranie HMI.



Tab.21.Text

Ikona	Nazwa	Opis	Dostępne operandy
	Tekst	Pozwala na wstawienie tekstu na ekran HMI.	
	Binary Text	Pozwala na wyświetlanie dwóch różnych tekstów w zależności od wartości bitu podpiętej zmiennej.	MB, SB, I, O, T, C
	List of text: by pointer	Pozwala na wyświetlanie kilku tekstów w zależności od wartości zmiennej.	MI, SI, ML, SL
	List of text: by range	Pozwala na wyświetlanie kilku tekstów w określonych przez użytkownika przedziałach wartości danej zmiennej zmiennej.	MI, #
	Display ASCII string	Pozwala na wyświetlenie stringu ASCII.	MI, ML, DW, MB



Tab.22.Image

Ikona	Nazwa	Opis	Dostępne operandy
	Bitmapa	Pozwala wstawić dowolną bitmapę na ekran HMI.	
	Binary images	Pozwala na wyświetlanie dwóch różnych bitmap w zależności od wartości bitu podpiętej zmiennej.	MB, SB, I, O, T, C
	List of images: by pointer	Pozwala na wyświetlanie kilku bitmap w zależności od wartości zmiennej.	MI, SI, ML, SL
	List of images: by range	Pozwala na wyświetlanie kilku tekstów w określonych przez użytkownika przedziałach wartości danej zmiennej zmiennej.	SI, ML, SL, MI, #
	Moving Image	Pozwala na wprowadzenie ruchu wybranej bitmapy w poziomie lub pionie. Ruch odbywa się wraz z inkrementacją bądź dekrementacją wartości przypisanej zmiennej.	MI, SI, ML, SL




Tab.23.Numeric

Ikona	Nazwa	Opis	Dostępne operandy
	Number	Pozwala wyświetlić: - wartość danej zmiennej - wpisać wartość do zmiennej z klawiatury (i określić jej wartość minimalną i maksymalną - dokonać linearyzacji wartości	MI, SI, ML, SL, DW, SDW
	Password	Pozwala na umieszczenie pola przeznaczonego na wpisanie hasła.	DW, #



Tab.24.Graph

Ikona	Nazwa	Opis	Dostępne operandy
	Bar Graph	Pozwala na umieszczenie wykresu słupkowego w celu badania zmiany wartości wybranej zmiennej.	MI, SI, ML, SL, DW, SDW
	Shape Graph	Pozwala na umieszczenie wykresu o wybranym kształcie w celu badania zmiany wartości wybranej zmiennej.	

Tab.25.RTC

Ikona	Nazwa	Opis	Dostępne operandy
	Real Time Clock	Pozwala wyświetlić na ekranie aktualny czas i datę.	
	Time function	Wyświetla przedział czasu, w którym zadana zmienna ma być aktywna (posiada stan 1).	MI, SI, ML, SL
	UTC	Pokazuje wartość podłączonego operandu jako RTC.	MI, SI, DW

Tab.26.Timer/Counter

Ikona	Nazwa	Opis	Dostępne operandy
	Timer	Pozwala wyświetlić wartość wybranego licznika czasowego, lub wprowadzić do niego wartość z klawiatury sterownika.	T
	Counter	Pozwala wyświetlić wartość wybranego licznika cyfrowego, lub wprowadzić do niego wartość z klawiatury sterownika.	C

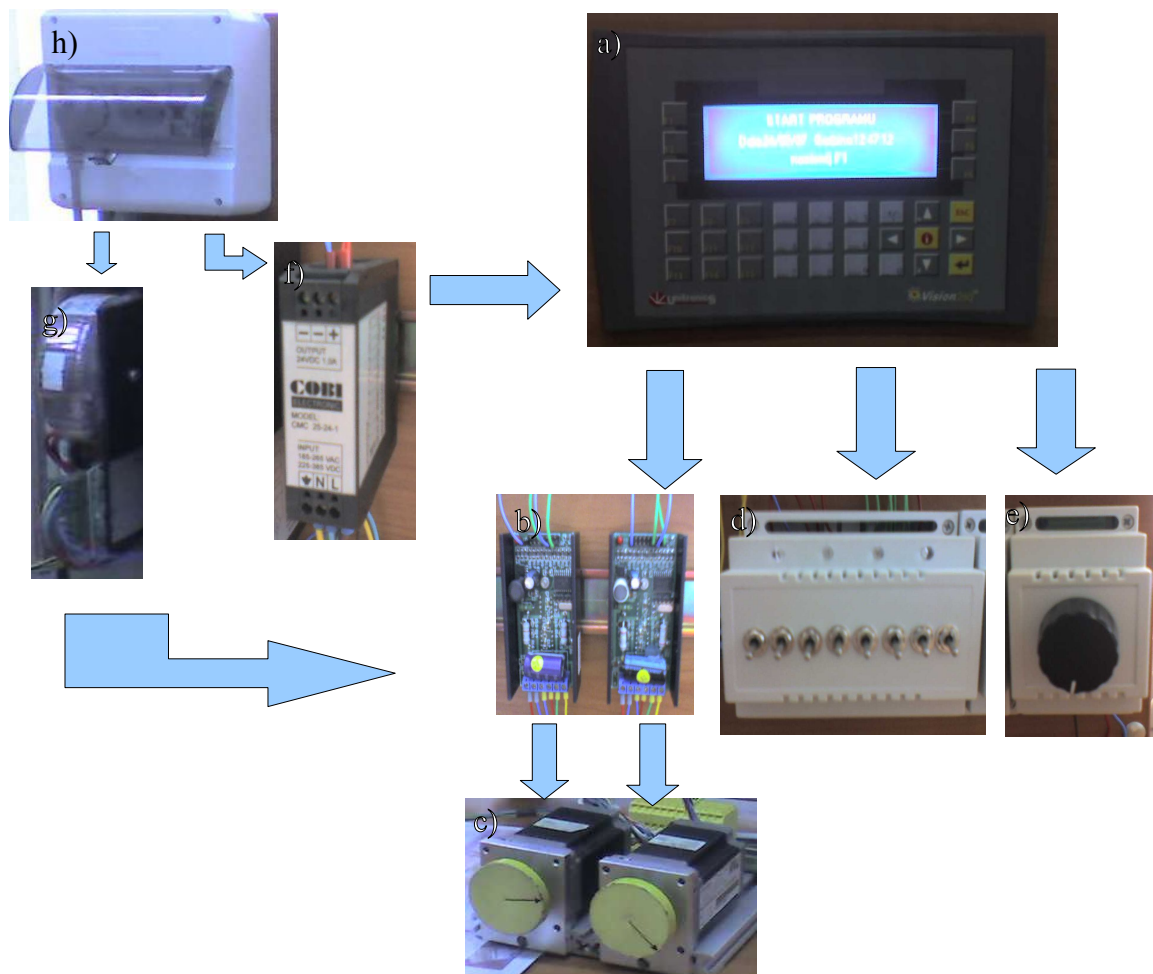
5.Stanowisko dydaktyczne ze sterownikiem Vision 260

5.1.Budowa stanowiska

Stanowisko (Rys.21.) przeznaczone jest do celów dydaktycznych. Zbudowane jest z następujących elementów (Rys.20.):

- a) Sterownika Vision 260
- b) sterowników SMC81
- c) dwufazowych silników krokowych
- d) puszek z wbudowanymi diodami i przełącznikami
- e) układu potencjometru
- f) zasilacza 24VDC COBI
- g) zasilacza 36VDC
- h) dwóch gniazd sieciowych

Elementy te przedstawiono na Rys.20.

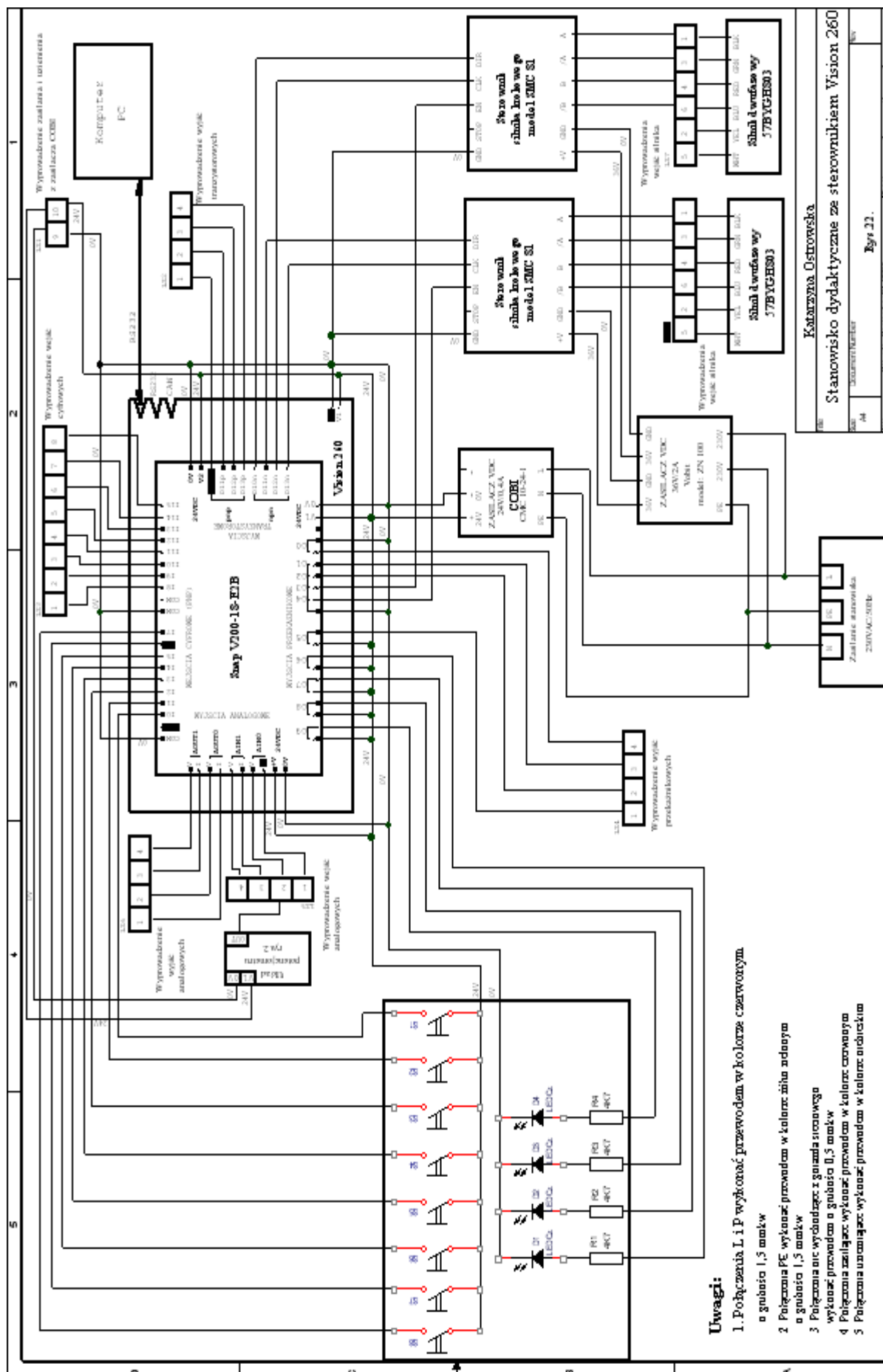


Rys.20.Elementy stanowiska dydaktycznego ze sterownikiem Vision



Rys.21.Stanowisko dydaktyczne z programatorem Vision 260

Stanowisko zasilane jest napięciem 230VAC/50Hz. Posiada ono dwa gniazda sieciowe, bezpiecznik 10A S310, oraz włącznik 20A LP351 uruchamiający stanowisko. Napięcie 230 VAC jest doprowadzane do zasilacza COBI oraz zasilacza Vobit ZN 100, gdzie transformowane jest odpowiednio na 24 VDC i 36 VDC. Zasilacz Vobit ZN 100 podłączono do sterowników SMC81 obsługujących silniki dwufazowe. COBI natomiast zasila sterownik Vision 260, dołączony do niego snap V200-18-E2B, układ czterech diod Led i układ ośmiu przełączników, a także układ potencjometru. Diody i przełączniki podłączone są do sterownika. Ich zadaniem jest symulacja stanów na wyjściach przekaźnikowych sterownika i wejściach cyfrowych. Potencjometr linowy przeznaczony jest do pracy z wejściami i wyjściami analogowymi sterownika Vision. Silniki krokowe sterowane są z poziomu programu stanowiska poprzez impulsy wysyłane z wyjść high-speed sterownika. Stanowisko wyposażone jest w komputer klasy PC połączony ze sterownikiem Vision 260 przez port RS232. Zainstalowano na nim oprogramowanie VisiLogic 5.02., które posłużyło do tworzenia programu i przesłania go do sterownika. Schemat stanowiska przedstawiono na Rys.22. i Rys.23.



5.2.Oprogramowanie stanowiska

Program stanowi integralną część stanowiska. Został napisany w oprogramowaniu VisiLogic przy pomocy języka drabinkowego (LD). Program został podzielony na część testową i automatyczną. Pierwsza część umożliwia testowanie pracy dwóch silników krokowych, oraz poszczególnych wejść i wyjść sterownika Vision 260. Druga część to demonstracja pracy obu silników naraz.

Po programie użytkownik porusza się za pomocą klawiatury sterownika stosując się do poleceń wyświetlanych na ekranie LCD. Wszystkie zmienne niezbędne do pracy wejść i wyjść wprowadzane są za pomocą klawiszy alfanumerycznych znajdujących się na panelu sterownika (wpisana zmienną należy zatwierdzić wciskając klawisz enter).

W celu uruchomienia programu należy załączyć zasilanie stanowiska. Następnie wcisnąć klawisz F1 w celu przejścia z ekranu startowego do Menu. Z poziomu Menu można przejść do podprogramu automatycznego (klawisz F3) lub testowego (klawisz F2), a także powrócić do ekranu startowego (klawisz ESC).

Podprogram automatyczny demonstruje prace silników krokowych w trzech fazach:

- praca ciągła
- dekrementacja częstotliwości obrotów silników
- inkrementacja częstotliwości obrotów silników

W podprogramie tym użytkownik zadaje częstotliwość dla obu silników przy użyciu klawiszy numerycznych i ustawia kierunek ich obrotu (w lewo lub w prawo) za pomocą strzałek z panelu sterownika. W podprogramie tym istnieje możliwość bezpośredniego przejścia do Menu poprzez wciśnięcie klawisza F1.

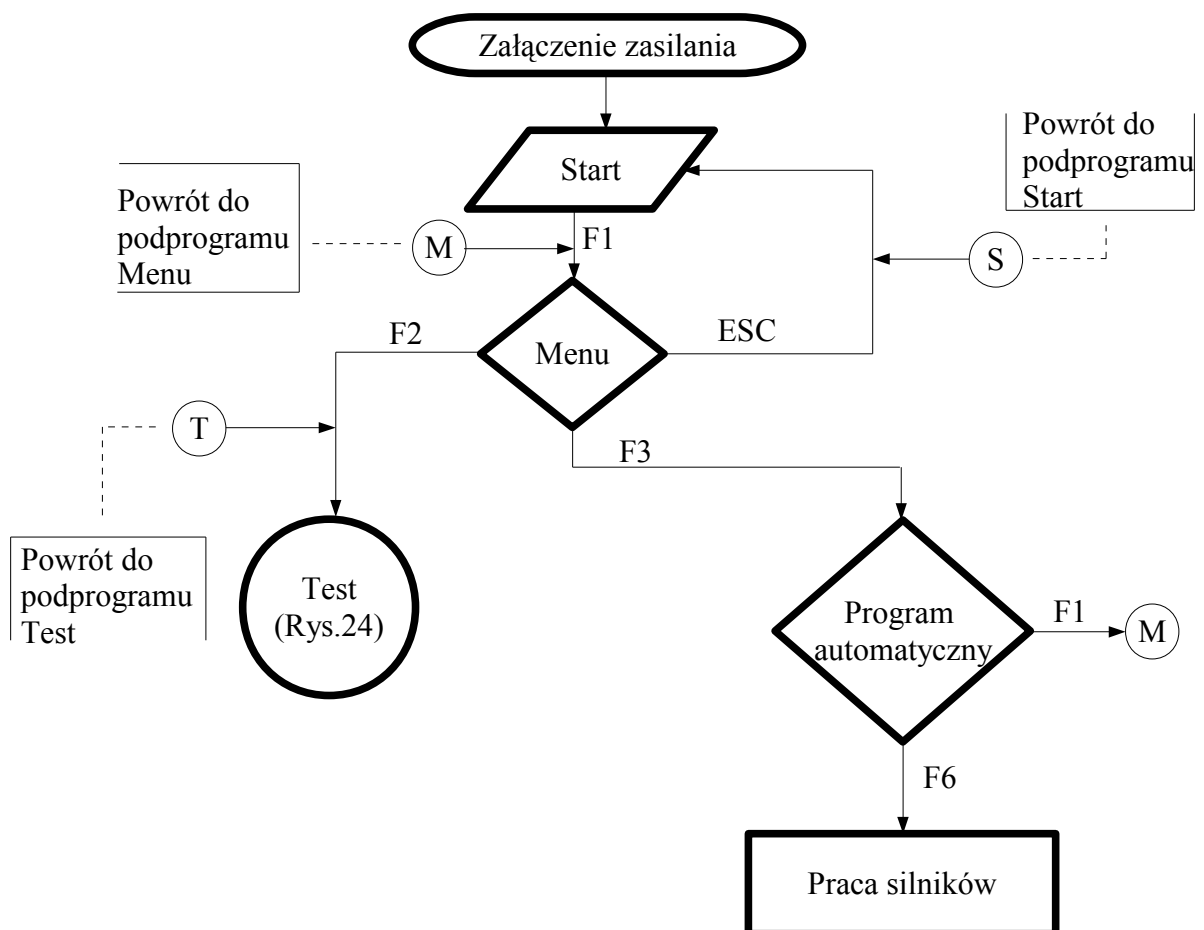
Z części testowej użytkownik może przejść do podprogramów testujących pracę wybranych wejść i wyjść poprzez wciśnięcie odpowiedniego klawisza z klawiatury sterownika Vision 260:

- test wejść cyfrowych (klawisz F7) – stan wysoki/niski danego wejścia cyfrowego ustawiany jest poprzez zmianę położenia odpowiadającego mu przełącznika (stan wysoki sygnalizowana jest pojawieniem się jedynki na wyświetlaczu pod odpowiednim numerem wejścia, a niski pojawieniem się zera)
- test wyjść przekaźnikowych (klawisz F8) – użytkownik poprzez wciśnięcie odpowiedniego klawisza z panelu sterownika może ustawić stan wysoki/niski na wyjściu (stan wysoki sygnalizuje zapalona dioda i jedynka pojawiająca się na ekranie sterownika pod nazwą wyjścia).

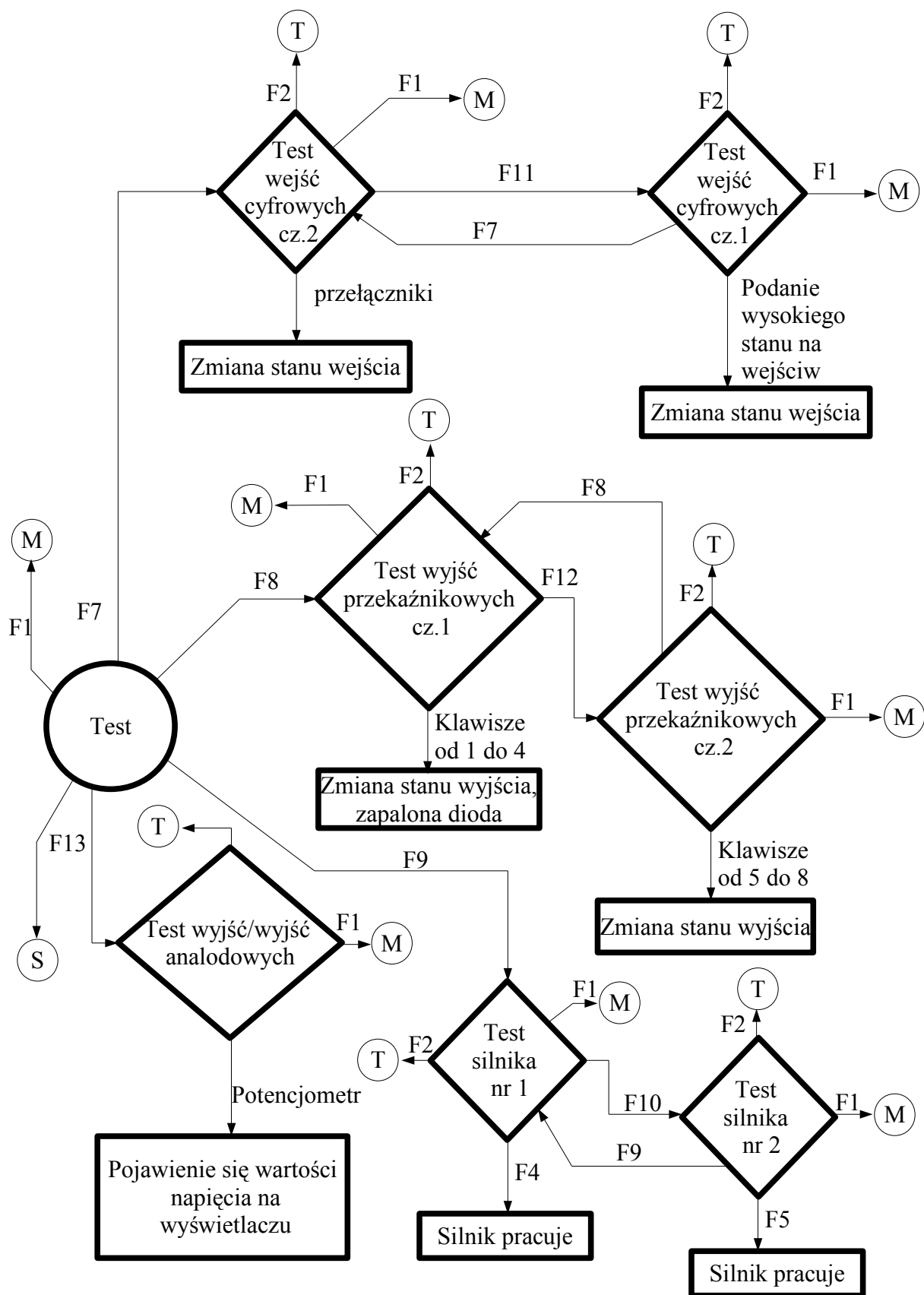
- test silników krokowych (klawisz F9) – pozwala na sprawdzenie pracy ciągłej każdego z silników. Użytkownik podaje częstotliwość pracy, drogę jaką ma przebyć silnik, oraz kierunek obrotu.
- test wejść/wyjść analogowych (klawisz F13) – umożliwia prześledzenie zmian napięcia sygnał wejściowego podawanego z potencjometru. Sygnał z wejść analogowych przekazywany jest do wyjść analogowych.

Każdy z powyższych podprogramów daje możliwość przejścia bezpośrednio do podprogramu Test za pomocą klawisza F2 i do Menu poprzez przyciśnięcie klawisza F1.

Sieć działania programu przedstawiono na : Rys.23. i Rys.24. Kod programu można znaleźć w załączniku nr 4



Rys.23.Schemat blokowy programu (cz.1)



Rys.24.Schemat blokowy programu (cz.2)

6. Podsumowanie

Podczas pracy nad stanowiskiem natrafiono na szereg problemów. Wynikały one głównie z niedokładności instrukcji dołączonej do sterownika Vision 260 i modułu snap V200-18-E2B. Inną przeszkodą okazała się niewielka ilość pamięci w sterowniku przez co nie użyto wielu zdefiniowanych w VisiLogic funkcji i zastąpiono je własnymi. Należy zaznaczyć, że dodatkowym utrudnieniem w programowaniu sterownika była niezgodność oprogramowania VisiLogic z jego pomocą np. często funkcje nie zgadzały się z opisem lub nie były opisane w ogóle.

Podczas budowy stanowiska nie wykorzystano wszystkich możliwości sterownika Vision 260. Dlatego warto pomyśleć o podłączeniu go do sieci sterowników lub dodaniu kilku czujników np. PT100.

W obecnej postaci stanowisko jest w pełni przygotowane do pracy dydaktycznej.

7. Literatura

- [1] T. Legierski, J.Kasprzyk, J.Wyrwał i J.Hajda, *Programowanie sterowników PLC*, Wydawnictwo Pracowni Komputerowej Jacka SKALMIERSKIEGO, Gliwice 1998.
- [2] S. Flaga, *Programowanie sterowników PLC w języku drabinkowym na przykładzie rodziny CPM1x/CPM2x*, ResNet S.C, Skawnia 2005.
- [3] <http://support.elmark.com.pl/unitronics/> - Oficjalna strona firmy UNITRONICS
- [4] <http://www.elmark.com.pl/> – Oficjalna strona firmy ELMARK.
- [5] *V200- 18- E2B Snap-in I/O Module*, dokumentacja modułu snap V200- 18 – E2B dołączona do sterownika Vision 260
- [6] *Vision OPLC Podręcznik Użytkownika*, dokumentacja sterownika Vision 260
- [7] *Instrukcja Obsługi SMC 81*, dokumentacja sterownika SMC 81.
- [8] *VisiLogic Software Manual - Ladder*, dokumentacja oprogramowania VisiLogic pobrana ze strony <http://support.elmark.com.pl/unitronics/>

Załączniki

Załącznik 1. Podręcznik użytkownika Vision 230, 260, 280

Załącznik 2. V200-18-E2B Snap-in I/O Module

Załącznik 3. Instrukcja Obsługi SMC 81

Załącznik 4. Kod programu stanowiska ze sterownikiem Vision 260

Załącznik 5. Instrukcja obsługi stanowiska dydaktycznego z programatorem Vision 260