

Uniwersytet Mikołaja Kopernika
Wydział Fizyki, Astronomii i Informatyki Stosowanej
Instytut Fizyki

Rafał Falkowski
nr albumu: 199547

Praca magisterska
na kierunku Fizyka Techniczna

Wizualizacja procesów i akwizycja danych w rozproszonych systemach sterowania – systemy SCADA

Opiekun pracy dyplomowej:
Dr inż. Kazimierz Karwowski
Zakład Fizyki Technicznej
i Zastosowań Fizyki

Toruń 2010

Pragnę podziękować Panu dr inż.
Kazimierzowi Karwowskiemu,
opiekunowi mojej pracy, za poświęcony
czas i zaangażowanie. Dziękuję również
Panu mgr inż. Krystianowi Erwińskiemu
za wskazówki i pomoc w wykonaniu
części praktycznej pracy magisterskiej.

*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. Rozproszone systemy sterowania	6
2.1. Wprowadzenie	6
2.2. Przemysłowe sterowniki programowalne	7
2.2.1. Sterowniki programowalne PLC	7
2.2.2. Sterowniki programowalne automatyki PAC	13
2.3. Programowanie sterowników przemysłowych	15
2.3.1. Języki programowania sterowników	15
2.3.2. Sekwencyjny schemat funkcjonalny SFC	22
2.4. Sieci przemysłowe	23
2.5. Systemy SCADA	24
2.5.1. Interfejs człowiek-maszyna (HMI)	25
2.5.2. Oprogramowanie i protokoły komunikacyjne	26
2.5.3. Główne cechy systemów SCADA	28
3. Wybrane oprogramowania SCADA	31
3.1. Wonderware InTouch	31
3.1.1. Narzędzia projektowe	31
3.1.2. System komunikacji	34
3.1.3. Cechy oprogramowania Wonderware	37
3.2. Rockwell FactoryTalk View Site Edition	41
3.2.1. Narzędzia projektowe	41
3.2.2. System komunikacji	43
3.2.3. FactoryTalk Services Platform	44
4. Stanowisko dydaktyczne ze sterownikiem CompactLogix i oprogramowaniem Wonderware InTouch	46
4.1. Budowa stanowiska	46
4.2. Oprogramowanie sterownika CompactLogix	47
4.3. Aplikacja InTouch	48
4.3.1. Zakładanie nowego projektu aplikacji	50

4.3.2.	Konfiguracja komunikacji ze sterownikiem	52
4.3.3.	Konfiguracja okien synoptycznych	55
4.3.4.	Definiowanie zmiennych	56
4.3.5.	Projektowanie i konfiguracja symboli graficznych	59
4.3.6.	Definiowanie skryptów	63
4.3.7.	Konfiguracja użytkowników i publikowanie aplikacji do katalogu	66
5.	Podsumowanie	70
6.	Literatura	71
7.	Załączniki	72

1. Wstęp

Obszar zastosowania rozproszonych systemów sterowania jest bardzo szeroki: poczynając od przedsiębiorstw infrastruktury komunalnej (wodociągi, przedsiębiorstwa elektroenergetyczne, ciepłownicze, gazownicze), przez zakłady przemysłowe (mechaniczne, chemiczne, spożywcze, itp.), kończąc na przedsiębiorstwach transportowych i instytucjach zajmujących się sterowaniem ruchem drogowym. Stąd należy wnioskować, że rozwój systemów pomiarowych i sterowniczych jest nieodzownie związany z rozwojem przedsiębiorstw.

Celem niniejszej pracy magisterskiej jest omówienie rozproszonych systemów sterowania, przegląd wybranego oprogramowania SCADA (Supervisory Control And Data Acquisition) oraz wykonanie przykładowej aplikacji demonstracyjnej (wizualizacji) przy pomocy tego oprogramowania.

Główne wytyczne, jakie należało uwzględnić podczas wykonywania pracy, to budowa stanowiska dydaktycznego, na którym zainstalowano oprogramowanie SCADA. Stanowisko dydaktyczne składa się z przetwornicy zasilanej z sieci trójfazowej, służącej do sterowania silnikiem indukcyjnym. Znajduje się również na nim komputer PC, który połączony jest ze sterownikiem programowalnym przez magistralę EtherNet/IP. Do wejść i wyjść sterownika podłączone są przełączniki i diody sygnalizujące, a do sterownika został wgrany program sterujący silnikiem, który symuluje pracę linii produkcyjnej. Za pomocą oprogramowania SCADA została wykonana wizualizacja tego procesu.

Pierwsza część pracy zawiera podstawy teoretyczne na temat rozproszonych systemów sterowania. Opisane zostały przemysłowe sterowniki programowalne – budowa i metody ich programowania. W dalszej kolejności przedstawiono podstawowe informacje na temat sieci przemysłowych i systemów SCADA.

Kolejny rozdział dotyczy wybranego oprogramowania SCADA: InTouch firmy Wonderware oraz FactoryTalk View Site Edition firmy Rockwell. Dla każdego z tych oprogramowań przedstawiono narzędzia projektowe, budowę systemu komunikacji oraz cechy charakterystyczne oprogramowania.

W rozdziale czwartym, ostatnim, opisano wykonanie części praktycznej – aplikacji demonstracyjnej. Aplikacja ta została wykonana za pomocą oprogramowania InTouch. Jest to wizualizacja procesu produkcyjnego wraz z możliwością sterowania tym procesem. Na tym przykładzie przedstawiono krok po kroku, jak należy projektować aplikacje.

2. Rozproszone systemy sterowania

2.1. Wprowadzenie

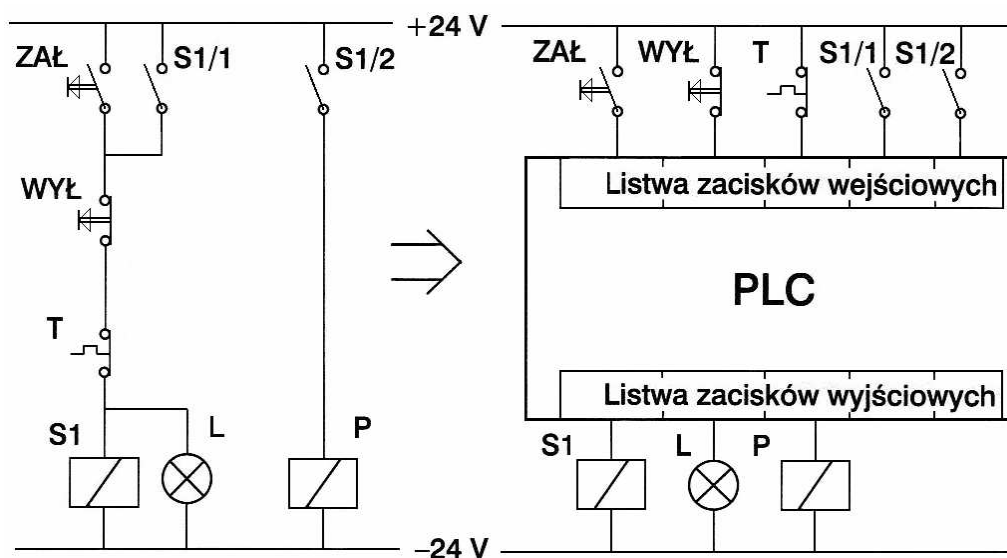
Jednym z kierunków rozwoju przedsiębiorstw jest rozwój techniki pomiarowej i sterowniczej poprzez wdrażanie nowych systemów. W wielu przedsiębiorstwach, w których urządzenia są rozmieszczone w znacznych odległościach od siebie, konieczne jest zainstalowanie rozproszonych systemów pomiarowo-kontrolnych. Kryterium podziału systemów na rozproszone i nierozproszone nie jest precyzyjne. Umownie za rozproszone systemy pomiarowo-kontrolne uważa się takie systemy, które mają urządzenia rozmieszczone w większej liczbie pomieszczeń niż jedno, lub w których odległość między przyrządami jest większa niż długość przewodu interfejsowego. Długość typowego przewodu interfejsowego wynosi od 1,5 m do 4 m.

W ostatnich 20 latach, wielkie zmiany, które zaszły w każdym elemencie systemu sterowania, nie ominęły także urządzeń do wizualizacji procesów. Osobom zwiedzającym zakład produkcyjny imponował widok pulpitu z dziesiątkami mierników w dyspozytorii. Z punktu widzenia ergonomii nie był to dobry sposób prezentacji wyników, ponieważ uważne śledzenie wskazań wielu podobnych przyrządów nie jest możliwe. Obecnie informacje o kontrolowanych procesach są podawane syntetycznie na monitorze komputera dzięki specjalnemu oprogramowaniu [1].

Systemem automatyki nazywamy zbiór elementów automatyki powiązanych ze sobą w określony sposób, stanowiący całość o określonym przeznaczeniu i scharakteryzowany pewną liczbą wielkości zwanych zmiennymi stanu. Stan systemu jest opisywany jako minimalna liczba danych wymaganych do dokładnego sprecyzowania jak zachowa się system przy danym sterowaniu. W praktyce system składa się z dwóch podsystemów: systemu sterowanego (obiektu sterowania) i systemu sterującego (komputera z urządzeniami wejściowo-wyjściowymi). Jeżeli stany systemu sterującego nie zależą od stanów systemu sterowanego, to mówimy, że system taki jest sterowny w torze otwartym. Natomiast, jeżeli stany systemu sterującego zależą od stanów systemu sterowanego, to system jest sterowany przy użyciu sprzężenia zwrotnego i nazywany jest systemem regulacji [2].

2.2. Przemysłowe sterowniki programowalne

Do końca lat osiemdziesiątych stosowano „drutowane” systemy sterowania. Polegały one na połączeniu styczników i przekaźników z elementami wykonawczymi przy użyciu przewodów według danego schematu. Systemy te zostały zastąpione przez sterowniki programowalne PLC (Rys. 1.). Sterownik PLC (Programmable Logic Controllers) odczytuje stany sygnałów z cyfrowych i analogowych elementów automatyki (przyciski, czujniki), które doprowadzone są do niego poprzez zaciski wejściowe. Następnie przetwarza te sygnały zgodnie z programem umieszczonym w sterowniku i przekazuje wyniki do zacisków wyjściowych, do których podłączone są styczniki, przekaźniki, silniki, itp. [2].



Rys. 1. Przykład zastosowania sterowników PLC [2]

2.2.1. Sterowniki programowalne PLC

Sterowniki programowalne PLC są to przemysłowe komputery umożliwiające sterowanie pracą maszyn i urządzeń. Mają one architekturę klasycznego komputera, na którą składają się [2]:

- jednostka centralna CPU wraz z pamięcią,
- moduły wejściowe i wyjściowe służące do wymiany danych pomiędzy otoczeniem a komputerem.

Jeden z podziałów sterowników to podział na sterowniki o budowie kompaktowej i modułowej. Sterowniki kompaktowe należą do klasy małych sterowników o sztywnej architekturze. W jednej obudowie znajduje się zasilacz, jednostka centralna oraz niewielka liczba wejść i wyjść. Przeznaczone są głównie do sterowania pojedynczymi maszynami [3]. W konstrukcji modułowej (Rys. 2.) można wyróżnić cztery trendy. Pierwszy polega na wsuwaniu modułów do firmowej kasety z gniazdami. W drugim, moduły zatrzaskuje się na standardowej szynie typu DIN i dociska do siebie w celu połączenia modułów. Trzeci trend dotyczy rozproszonych systemów sterowania i wtedy odległości między modułami sterownika sięgają setek metrów. Natomiast czwarty trend polega na zastosowaniu sterowania programowego – jednostką centralną jest przemysłowy PC-et pracujący z systemem operacyjnym (przeważnie Windows XP), na którym obok programu do obsługi modułów sterownika istnieje oprogramowanie PC umożliwiające pełną integrację z pakietami pracującymi w środowisku Windows. Pociąga to za sobą takie możliwości jak: stosowanie bloków zdefiniowanych przez użytkownika i napisanych np. w języku C++, Java, i wewnętrzne połączenie z aplikacjami SCADA. Podobne do czwartego trendu właściwości programowe ma grupa sterowników, która nosi nazwę programowalnych sterowników automatyki PAC [2].



Rys. 2. Budowa modułowa sterownika PLC firmy GE Fanuc [12]

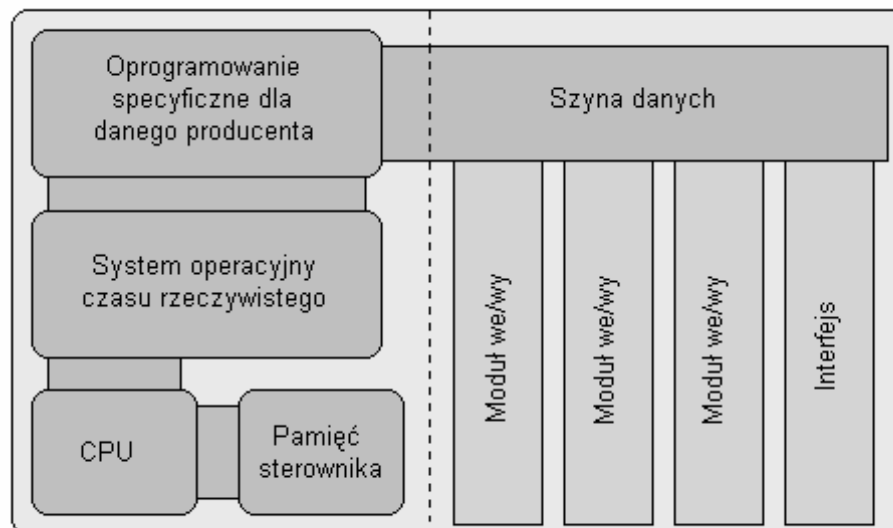
Sterowniki programowalne PLC są bardzo często spotykane w przemyśle. Wiele linii produkcyjnych modernizuje się, zastępując istniejące modele nowszymi. Istotnymi parametrami, stanowiącymi o przydatności sterowników w aplikacjach sterowania, są: niezawodność, wydajność, powtarzalność oraz przewidywalność. Zwracanie uwagi na te aspekty działania PLC zwiększa zapotrzebowanie na proste i odporne systemy, mogące

pracować w niebezpiecznych warunkach przemysłowych. Jednakże sterowniki PLC w ostatnich latach dość mocno ewoluowały. Wprowadzono do ich konstrukcji wiele standardowych interfejsów komunikacyjnych. Dodatkowo, coraz większe są możliwości oprogramowania zintegrowanego w ramach jednego rozwiązania systemu automatyki. Nazwa PLC jednoznacznie określa możliwości współczesnych systemów sterowania [4].

Architektura sprzętowa sterownika

Typowa budowa sprzętowa sterownika PLC (Rys. 3.) obejmuje pięć podstawowych komponentów [4]:

- mikroprocesor – jest tym komponentem sterownika, który wykonuje algorytm sterowania (zwykle 16 bitowy w sterownikach małych oraz 32 bitowy w dużych).
- pamięć – służy do przechowywania informacji w postaci cyfrowej. Procesor wykorzystuje pamięć do tymczasowego przechowywania danych potrzebnych przy wykonywaniu kodu programu sterownika. Pamięć ROM jest przeznaczona do przechowywania systemu operacyjnego, jednak coraz częściej oprogramowanie to znajduje się w pamięci typu FLASH. W pamięci RAM przechowywany jest program aplikacji użytkownika oraz dane użytkownika i systemowe [3].
- zasilacz i baterie – zasilacz zapewnia wewnętrzne zasilanie jednostki centralnej oraz modułów wejść/wyjść. Natomiast dzięki bateriom program sterownika oraz dane przechowywane są w przypadku zaniku zasilania.
- szyna danych – jest to system, dzięki któremu obsługiwane są moduły wejść/wyjść, czy moduły komunikacji. Służy ona procesorowi do wymiany informacji z pozostałymi komponentami sterownika. Podstawowymi modułami stanowiącymi interfejs między sterownikiem a otoczeniem są: moduły wejść i wyjść cyfrowych oraz moduły wejść i wyjść analogowych. Moduły we/wy cyfrowych zwane są modułami we/wy dyskretnych i na ogół dotyczą sygnałów dwustanowych [3].
- obudowa sterownika – dzięki niej wszystkie komponenty mogą funkcjonować w ramach jednego urządzenia. Wśród rozwiązań znaleźć można obudowy, w których montuje się wszystkie komponenty (rozwiązanie kompaktowe) oraz częściej stosowane, konstrukcje bez wspólnej obudowy, w których moduły łączone są ze sobą.



Rys. 3. Architektura sprzętowa sterownika PLC [4]

Komponenty architektury sprzętowej komputera PC niewiele różnią się od tych spotykanych w sterownikach PLC. Stałymi elementami są: procesor, pamięć danych, przestrzeń dyskowa, elementy do przechowywania i wykonywania programu, system obsługi wejść i wyjść oraz urządzeń peryferyjnych, i obudowa, w której to wszystko jest umieszczone. Sterownik PLC jest przystosowany do pracy w warunkach przemysłowych, natomiast komputer PC nie jest zaprojektowany do pracy w środowisku agresywnym, tylko do zastosowań ogólnych, głównie biurowych. Z drugiej strony, ważną zaletą komputerów w porównaniu do PLC jest prostota ich rozbudowy sprzętowej. Komputery bazują na dostępnych na rynku komponentach, które można dowolnie modyfikować, np. w celu osiągnięcia większej wydajności, rozbudowy możliwości komunikacyjnych, powiększenia rozmiaru pamięci czy też przestrzeni dyskowej do przechowywania danych. Komponenty, służące do przechowywania danych, są łatwiej dostępne konstrukcyjnie w przypadku komputerów PC, dzięki temu bardziej od sterowników PLC nadają się do przechowywania dużych programów. Ważne jest też, że elementy takie jak wirujące elementy dysków twardych nie nadają się do zastosowania w warunkach przemysłowych, w których z reguły działają sterowniki programowalne [4].

Architektura programowa sterownika

Podstawowym komponentem architektury oprogramowania sterownika PLC jest jego system operacyjny. Systemy operacyjne czasu rzeczywistego zapewniają sterownikom stabilność działania, jak i niezawodność. Dodatkowo, w ramach systemu operacyjnego wykonywany jest program, napisany w języku typowym dla sterowników PLC. Sterowniki PLC posiadają strukturę oprogramowania, przystosowaną do wykonywania pętli regulacji [4]. Cykl pracy (Rys. 4.) zawiera fazy: odczytu danych wejściowych (skanowanie stanu wejść), wykonania zaprojektowanego przez użytkownika programu sterującego, wysłania danych wyjściowych, obsługi procedur komunikacji i diagnostyki (sprawdzenie konfiguracji oraz poprawności kolejnego fragmentu programu użytkownika). W standardowym trybie pracy każdy cykl jest wykonywany tak szybko jak to możliwe, w konsekwencji czas trwania pojedynczych cykli może się zmieniać. Istnieje jednak możliwość ustawienia sterownika na pracę ze stałym czasem trwania cyklu. Jednym z powodów dla zastosowania tego trybu może być potrzeba zapewnienia uaktualnienia stanu wejść sterownika w stałych odstępach czasu. Innym powodem jest wprowadzenie pewnego odstępu czasowego pomiędzy fazą obsługi wyjść a fazą obsługi wejść sterownika w następnym cyklu, co umożliwia ustalenie się stanu wejść po otrzymaniu danych wyjściowych z programu [5].

Sterowniki PLC nie mają własnego interfejsu człowiek-urządzenie, stąd powinny mieć specjalny port, do którego można podłączyć komputer PC lub panel operatorski z odpowiednim oprogramowaniem. Łącze takie najczęściej jest wykonane w standardzie RS-232C lub RS-485 i umieszczone na module CPU. Transmisja na takim łączu odbywa się z użyciem danego protokołu, który opracowany jest przez producenta sterownika, np. Modbus. Dodatkowo można zastosować szereg modułów komunikacyjnych, np. przeznaczonych do sieci Ethernet z protokołem TCP/IP. Możliwość podłączenia sterowników PLC do różnych standardowych tzw. sieci polowych (Fieldbus) przy wykorzystaniu standardowych protokołów komunikacyjnych sprawia, że są one idealne do zastosowania w rozproszonych systemach sterowania, w których mogą być używane urządzenia różnych producentów. Dzięki zaawansowanej diagnostyce sieciowej jest zapewniona kontrola statusów wszystkich elementów sieci oraz obwodów wejściowo-wyjściowych [3].



Rys. 4. Cykl pracy sterownika [5]

Programista zajmuje się jedynie tworzeniem programu sterowania, ponieważ funkcje, takie jak skanowanie wejść/wyjść oraz komunikacja i diagnostyka wykonywane są bez kontroli użytkownika. Dzięki takiej architekturze możliwe jest sprawne i efektywne tworzenie systemów sterowania. Sztywność tego typu systemów uwalnia inżyniera od konieczności poznania operacji niższego rzędu w sterownikach PLC. Jednakże taka architektura oprogramowania systemowego sterownika PLC, będąca z jednej strony zaletą, sprawia że czasami staje się on mało elastyczny. Typowe rozwiązania sterowników programowalnych nie posiadają bardziej złożonych, szybkich wejść/wyjść, czy procesorów, dzięki którym np. może być prowadzone sterowanie ruchem. Wynika z tego, że system taki wymaga osobnych urządzeń, które z reguły posiadają inne oprogramowania, niż to służące do tworzenia aplikacji dla sterownika PLC. Odrębne oprogramowanie i sprzęt znacznie utrudnia osiągnięcie wymaganego poziomu integracji systemu. Większość producentów sterowników PLC rozszerza swoją ofertę o sterowniki PAC oraz rozbudowuje istniejące oprogramowanie dla PLC o nowe funkcje [4].

Podsumowując, o niezawodności sterowników PLC świadczą dwa podstawowe czynniki: odporność konstrukcji mechanicznej, która przystosowuje je do pracy w warunkach przemysłowych oraz sztywna architektura oprogramowania, w której jedyna ingerencja użytkownika polega na wprowadzeniu do systemu programu sterowania.

2.2.2. Sterowniki programowalne automatyki PAC

Obecnie coraz częściej stosowane są rozwiązania będące połączeniem sterownika PLC (jego wytrzymałości mechanicznej, jakości wykonania i odporności na zakłócenia przemysłowe) z funkcjonalnością komputerów osobistych PC. Rozwiązania te nazywane są programowalnymi sterownikami automatyki PAC (Programmable Automation Controller). Cechą charakterystyczną tych rozwiązań jest obecność w nich systemu operacyjnego czasu rzeczywistego, zarządzającego pracą urządzenia, przydzielaniem zadań sterowania i wizualizacji w sposób hierarchiczny. Rozwiązania te oferują też bardzo często możliwość ustalenia stałego czasu cyklu pracy, co jest niezbędne w przypadku realizacji dowolnego algorytmu cyfrowej regulacji procesami ciągłymi. Poza sterowaniem procesami dyskretnymi, sterowniki typu PAC oferują użytkownikom niezliczoną ilość funkcji dedykowanych sterowaniu procesami ciągłymi. Stanowią hybrydowe połączenie regulatora procesów ciągłych i sterownika procesów dyskretnych.

Sterowniki PAC przeznaczone są dla celów integracji systemów automatyki, stawiając producentów oryginalnych urządzeń oraz użytkowników końcowych przed faktem konieczności rozszerzenia swej wiedzy na temat systemów automatyki. Skutkiem tego są niższe koszty utrzymania oraz niższy całkowity koszt posiadania urządzeń. Producenci ciągle poszukują mniejszych i inteligentniejszych komponentów do stosowania w swoich produktach. Powodem tego jest pojawienie się na rynku coraz bardziej elastycznych urządzeń automatyki. Sterowniki PAC pokrywają w pełni to zapotrzebowanie dzięki: niezawodności platformy podobnej do sterowników PLC, funkcjonalności komputerów PC, wykorzystaniu standardów przemysłowych i prostocie wymiany informacji pomiędzy produktami różnych producentów.

Sterowniki PAC są na tyle elastyczne, że każdy użytkownik może je odpowiednio skonfigurować do wymogów aplikacji, zarówno w przypadku sterowania pojedynczą maszyną, jak i całą fabryką. Wszystkie części systemu zaprojektowane są tak, aby poziom integracji sprzętu i oprogramowania był możliwie najwyższy. W przypadku sterowników PAC wprowadza się takie podejście, że użytkownik korzysta z jednego oprogramowania

i jednego zestawu narzędzi dla całego systemu. Oprogramowanie sterownika PAC zapewnia stabilność i niezawodność systemu operacyjnego czasu rzeczywistego, która jest istotna podczas obsługi wejść/wyjść, ustalania zależności czasowych w systemie, priorytetów wykonywania zadań, takich jak np. pętle sterowania. Udostępnia także wystarczającą liczbę funkcji sterujących i analitycznych oraz umożliwia zarówno wykonywanie zadań sterowania logicznego jak i regulacji PID.

Dzięki sterownikom PAC osiągnięto sytuację, kiedy nie ma potrzeby stosowania nietypowego sprzętu. Pomimo tego, że sterowniki PAC są obecnie tym, co najnowsze w dziedzinie programowalnych sterowników, to wydaje się, że ich przyszłość jest związana z technologią zagnieżdżania. Jednym z przykładów może być konfiguracja architektury sprzętu za pomocą oprogramowania. Układy programowalne FPGA są komponentami elektronicznymi wykorzystywanymi między innymi w celu swobodnego budowania dowolnych procesorów. Wykorzystując funkcjonalność konfigurowalnych bloków logicznych wraz z modułami wejść/wyjść, inżynierowie mogą zaprojektować dowolny procesor bez ponoszenia kosztów związanych z ewentualną budową prototypu. Moduły FPGA można porównać do komputera, w którym dokonano zmiany połączeń tak, aby całe urządzenie wykonywało zaprogramowane przez użytkownika funkcje.

Programowalne sterowniki automatyki odgrywają znaczącą rolę w automatyzacji procesów, z uwagi na swoją zgodność ze standardami przemysłowymi oraz funkcjonalność przy różnych możliwościach programowania – od języków dla sterowników PLC po języki wyższego poziomu, jak np. ANSI C. Wszystkie elementy konstrukcji sterownika PAC muszą być tak zaprojektowane, aby poziom dopasowania sprzętu i oprogramowania był maksymalny. Cechą szczególną jest wykorzystanie jednego oprogramowania dla celów programowania i konfigurowania systemu jako całości złożonej z różnych elementów [4].

Podsumowując, można powiedzieć, że nowa generacja sterowników programowalnych powstała na skutek coraz większych wymagań inżynierów, dla których z jednej strony istotna jest odporność mechaniczna, stabilność, odporność na zakłócenia i niezawodność systemu typowa dla sterowników PLC, a z drugiej bardzo ważna jest wydajność i szeroko rozumiana elastyczność – typowa dla komputerów PC.

2.3. Programowanie sterowników przemysłowych

Możliwość programowania sterownika PLC jest dla użytkownika najbardziej interesującym elementem systemu sterowania realizowanego za ich pomocą, ponieważ w ten sposób wprowadza się do systemu odpowiedni algorytm sterowania. Przez cały okres rozwoju sterowników programowalnych producenci wprowadzali różne metody programowania. Poszczególne rozwiązania proponowane przez różnych producentów różniły się między sobą, co dla projektantów takich systemów stanowiło dużą niedogodność. Doświadczenie zebrane przez producentów i użytkowników systemów sterowania zaowocowały wydaniem normy IEC 61131 przez Międzynarodową Komisję Elektroniki (International Electrotechnical Commission). Wprowadzenie tej normy miało na celu ograniczenie dowolności istniejącej na poziomie sprzętowym oraz programowym. Trzecia część normy dotyczy języków programowania i z punktu widzenia użytkownika stanowi jej najważniejszą część. Dzięki niej ujednolicono koncepcję programowania sterowników PLC tak, aby użytkownik korzystając z wprowadzonych reguł, był w stanie programować bez większych trudności różne systemy sterowania [3].

2.3.1. Języki programowania sterowników

W trzeciej części normy IEC 61131 zdefiniowano pojęcia podstawowe i zasady ogólne związane z językami programowania i oprogramowaniem sterowników PLC oraz model programowy i model wymiany danych między elementami oprogramowania. Norma definiuje także podstawowe typy i struktury danych. Ustalono również zasady ułatwiające komunikację między sterownikami PLC i innymi elementami systemów automatyki [6].

W normie tej (IEC 61131-3) określono dwie grupy języków programowania:

- języki graficzne:
 - schemat drabinkowy LD (Ladder Diagram),
 - funkcjonalny schemat blokowy FBD (Function Block Diagram),
- języki tekstowe:
 - lista instrukcji IL (Instruction List),
 - tekst strukturalny ST (Structured Text).

Język LD

Język LD należy do grupy języków graficznych i umożliwia realizację zadania sterowania za pomocą standaryzowanych symboli graficznych. Fundament do opracowania tego języka stanowił język schematów przekaźnikowych, który używany jest do opisu klasycznych układów sterowania.

Obwód, który stanowi schemat drabinkowy jest zdefiniowany jako zbiór połączonych ze sobą elementów graficznych, ograniczonych z lewej i prawej strony przez szyny prądowe, przy czym nie są one elementami obwodu. Z obwodem może być skojarzona etykieta, której przypisano nazwę lub liczbę dziesiętną. Stosowanie nazwy etykiety jest lokalne w zakresie elementu organizacyjnego oprogramowania, w którym występuje [6].

Program napisany w języku LD reprezentuje sekwencje obwodów, które ma wykonać sterownik PLC. Podczas realizacji programu obowiązują następujące zasady [6]:

- nie można określić wartości żadnego sygnału wyjściowego obwodu, dopóki nie zostanie określony stan wszystkich sygnałów wejściowych,
- określenie wartości wyjściowych elementu obwodu nie może być zakończone, dopóki nie zostało zakończone określenie wartości wszystkich jego sygnałów wyjściowych,
- wykonywanie programu użytkownika nie może być zakończone, dopóki nie zostały ustawione wyjścia wszystkich elementów obwodu,
- program jest wykonywany od góry do dołu, zgodnie z kolejnością występowania szczebli w schemacie drabinkowym, z wyjątkiem gdy w schemacie drabinkowym pojawiają się elementy skoku lub powrotu warunkowego.

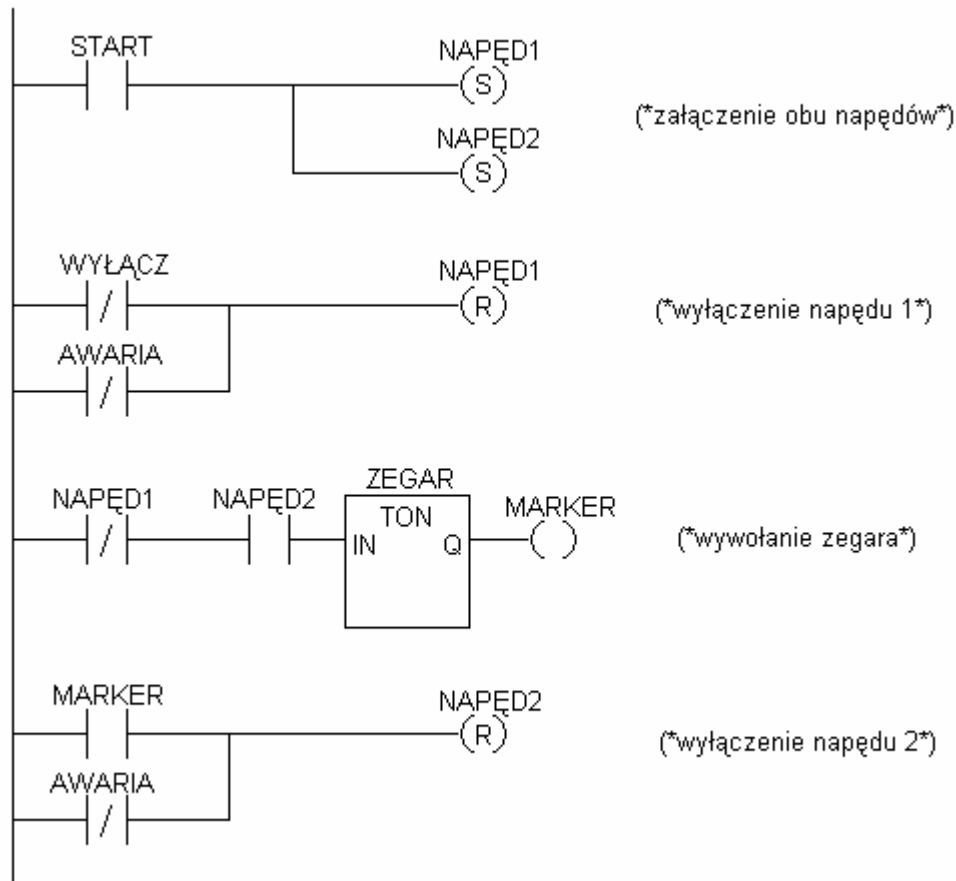
Elementy łączące mogą znajdować się w jednym z dwóch stanów ON lub OFF, co umownie odpowiada wartością algebry Boole'a 0 i 1. Lewa szyna schematu drabinkowego znajduje się zawsze w stanie ON, natomiast stan prawej strony szyny jest nieokreślony.

Połączenia poziome są przedstawiane za pomocą linii poziomych. Poziomy element łączący przekazuje stan elementu znajdującego się po jego lewej stronie do elementu umiejscowionego po stronie prawej. Połączenia pionowe przedstawia się liniami pionowymi, które przecinają się z każdej strony z jednym lub większą liczbą połączeń poziomych. Stan pionowego elementu łączącego odpowiada sumie logicznej stanów poziomych elementów łączących znajdujących się po jego lewej stronie. Oznacza to, że: element pionowy jest w stanie OFF, gdy stany wszystkich elementów poziomych są OFF oraz element pionowy jest

w stanie ON, gdy co najmniej jeden element poziomy jest w stanie ON. Stan połączenia pionowego powinien być przenoszony na wszystkie połączenia poziome znajdujące się po jego prawej stronie, natomiast nie może być przekazywany do jakiegokolwiek połączenia poziomego umiejscowionego po jego lewej stronie [6].

Podstawowymi elementami obwodów w języku LD są styki i cewki. Styk jest elementem przekazującym do połączenia poziomego po prawej stronie styku stan będący wynikiem mnożenia boolowskiego AND stanu połączenia po lewej stronie styku oraz wartości przypisanej stykowi zmiennej logicznej. Cewka przekazuje stan połączeń z lewej strony na prawą bez zmian. Jednocześnie cewka zapamiętuje stan elementu z lewej strony przez przyporządkowaną jej zmienną logiczną [3]. Symbole graficzne styków i cewek pokazano w Tab. 1.

Język LD stanowi najprostsze narzędzie programowania wyrażeń logicznych. Oprócz typowych elementów języka służących sterowaniu, w języku tym istnieje także możliwość realizacji operacji bardziej złożonych, np. arytmetycznych, przez wywołanie odpowiednich funkcji lub bloków funkcjonalnych (Rys. 5.).



Rys. 5. Przykład programu w języku LD [3]

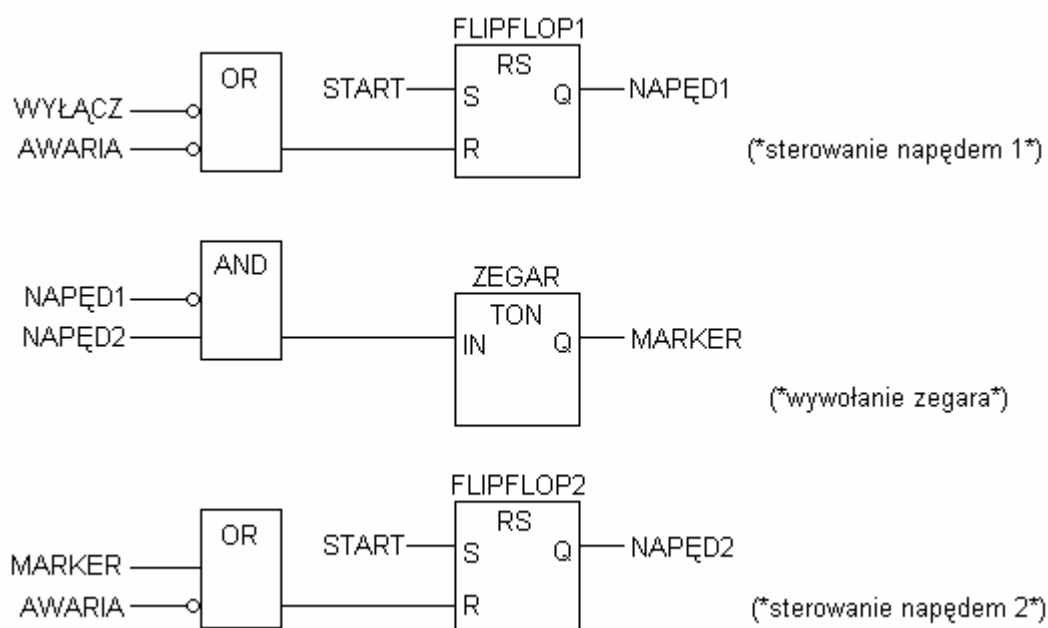
Tab. 1. Symbole styków i cewek [3]

Symbol	Opis
-- --	Styk zwierny (normalnie otwarty) Sygnał z lewej strony styku jest przenoszony na prawą stronę, jeżeli przypisana zmienna logiczna ma wartość 1, w przeciwnym przypadku połączenie jest w stanie OFF.
-- / --	Styk rozwierny (normalnie zamknięty) Sygnał z lewej strony styku jest przenoszony na prawą stronę, jeżeli przypisana zmienna logiczna ma wartość 0, w przeciwnym przypadku połączenie jest w stanie OFF.
-- P --	Styk reagujący na zbocze narastające Stan styku jest w stanie ON w czasie jednego cyklu, jeśli lewa strona jest w stanie ON, a przypisana zmienna logiczna zmieniła wartość z 0 na 1. Poza tym stan styku jest OFF.
-- N --	Styk reagujący na zbocze opadające Stan styku jest w stanie ON w czasie jednego cyklu, jeśli lewa strona jest w stanie ON, a przypisana zmienna logiczna zmieniła wartość z 1 na 0. Poza tym stan styku jest OFF.
--()--	Cewka Stan lewej strony cewki jest przenoszony na prawą stronę i zapamiętywany w przypisanej zmiennej logicznej.
--(/)--	Cewka zanegowana Stan lewej strony cewki jest przenoszony na prawą stronę, a jego odwrotność jest zapamiętywana w przypisanej zmiennej logicznej.
--(S)--	Zapis cewki Przypisana zmienna logiczna przyjmuje wartość 1, jeżeli stan połączenia z lewej strony jest ON. Zmianę stanu cewki ustawiającej powoduje cewka kasująca.
--(R)--	Kasowanie cewki Przypisana zmienna logiczna przyjmuje wartość 0, jeżeli stan połączenia z lewej strony jest ON. Zmianę stanu cewki kasującej powoduje cewka ustawiająca.
--(M)--	Cewka z zapamiętaniem stanu
--(SM)--	Zapis cewki z zapamiętaniem stanu
--(RM)--	Kasowanie cewki z zapamiętaniem stanu
--(P)--	Cewka reagująca na zbocze narastające Przypisana zmienna logiczna przyjmuje wartość 1 w czasie jednego cyklu, jeśli stan lewej strony cewki zmienił się z OFF na ON. Stan lewej strony jest zawsze przenoszony na prawą.
--(N)--	Cewka reagująca na zbocze opadające Przypisana zmienna logiczna przyjmuje wartość 1 w czasie jednego cyklu, jeśli stan lewej strony cewki zmienił się z ON na OFF. Stan lewej strony jest zawsze przenoszony na prawą.

Język FBD

Język FBD umożliwia zapisanie programu użytkownika za pomocą bloków operacji logicznych. W języku tym mogą być również używane inne bloki funkcjonalne, np. liczniki. Obwód, podobnie jak w języku LD, jest zdefiniowany jako zbiór połączonych ze sobą elementów graficznych języka. Z obwodem lub zbiorem obwodów może być skojarzona etykieta, której przypisano nazwę lub liczbę dziesiętną. Używanie nazwy ma zasięg lokalny w obrębie elementu organizacyjnego oprogramowania, w którym występuje. Podczas wykonywania programu użytkownika, obowiązują takie same zasady jak w przypadku realizacji programu zapisanego w języku LD. Również elementy powinny być łączone za pomocą linii przepływu sygnałów zgodnie z zasadami podanymi wcześniej.

W obrębie obwodu mogą występować sprzężenia zwrotne wtedy, gdy wyjście bloku funkcjonalnego stanowi wejście bloku, który go poprzedza w obwodzie. Wykonywanie obwodów jest zgodne z kolejnością ich występowania w programie, przy czym ta kolejność może być zmieniona, gdy zostaną użyte elementy języka służące do sterowania wykonywaniem programu. Wyjścia bloków funkcjonalnych nie powinny być ze sobą łączone, zwłaszcza jest niedopuszczalna realizacja sumy logicznej OR za pomocą równoległego łączenia elementów w węzeł, tak jak jest to możliwe w języku LD [6]. Na Rys. 6. pokazano analogiczny program jak na Rys. 5., ale zapisany w języku FBD.



Rys. 6. Przykład programu w języku FBD [3]

Język IL

Język IL jest klasycznym tekstowym językiem programowania sterowników, zbliżony składnią do języków typu assembler [3]. Program użytkownika reprezentuje sekwencję instrukcji, którą ma wykonywać sterownik PLC. Każda instrukcja powinna zaczynać się od nowej linii, zawierać nazwę operatora z ewentualnymi modyfikatorami, i jeżeli jest to niezbędne dla określonej operacji, powinna zawierać jeden lub więcej operandów (Tab. 2.). Operandami mogą być stałe lub zmienne. Instrukcję może poprzedzać etykieta [6].

Tab. 2. Przykład sekwencji instrukcji [6]

Etykieta	Operator	Operand	Komentarz
START:	LD	%IX1	(*wciśnij przycisk*)
	ANDN	%MX5	(*nie wstrzymany*)
	ST	%QX2	(*załącz*)

Działania wykonywane przez poszczególne operatory są realizowane w następujący sposób: wartość wyrażenia jest wyznaczana jako wynik działania operatora na aktualną wartość wyrażenia, z uwzględnieniem wartości operandu. Przykładowo instrukcję AND %MX5 należy interpretować jako: wynik:= wynik AND %MX5. Modyfikator N określa negację operandu, np. ANDN %MX5 oznacza, że wynik:= wynik AND NOT %MX5. Modyfikator C oznacza, że instrukcja powinna być wykonana wtedy, gdy aktualnie wyznaczony wynik ma wartość logiczną 1 lub 0, gdy operator jest połączony z modyfikatorem N.

Wynik porównania realizowanego przez operatory należy interpretować jako efekt porównania wyniku bieżącego, znajdującego się z lewej strony operatora, z operandem znajdującym się z jego prawej strony. Funkcje można wywoływać, umieszczając nazwę funkcji w polu operatora. Wynik bieżący stanowi pierwszy argument funkcji. Jeżeli jest wymaganych więcej argumentów, to należy je zadeklarować w polu operandu. Wartość określona przez funkcję, w wyniku pomyślnego wykonania instrukcji RET lub po osiągnięciu końca funkcji, staje się wartością bieżącą [6].

Język ST

Język ST jest językiem tekstowym wyższego poziomu, ponieważ nie używa się w nim operatorów, ale poleceń opisujących w skondensowany sposób działanie sterownika [3]. Podstawowymi elementami tego języka są wyrażenia i instrukcje. Wyrażenie stanowi konstrukcję złożoną z operatorów i argumentów, które po jego obliczeniu daje wartość odpowiadającą jednemu z typów danych. Argumentem może być stała, zmienna, wywoływana funkcja lub inne wyrażenie [6].

Następnymi elementami są instrukcje. Instrukcja przypisania powoduje zastąpienie wartości bieżącej zmiennej przez wynik obliczenia wyrażenia. Wywołanie funkcji umożliwia wprowadzenie do wyrażenia jej nazwy i listy argumentów umieszczonych w nawiasach. Opuśczenie funkcji następuje w wyniku dojścia do ich końca, natomiast wcześniejsze wyjście umożliwia instrukcja RETURN, która jest zwykle wynikiem wykonania instrukcji IF.

Instrukcja warunkowa IF umożliwia wykonanie jednej lub grupy instrukcji, pod warunkiem spełnienia określonego warunku, natomiast instrukcja wyboru CASE umożliwia wybór instrukcji do wykonania spośród wielu opcji.

Instrukcje iteracji umożliwiają cykliczne wykonywanie sekwencji instrukcji. W przypadku, gdy liczba iteracji jest określona, to stosuje się instrukcję FOR, w przeciwnym wypadku należy użyć instrukcji WHILE lub REPEAT. Wcześniejsze zakończenie (następujące przed spełnieniem warunku określającego zakończenie cyklicznej realizacji sekwencji instrukcji) wykonywania sekwencji instrukcji w pętli umożliwia instrukcja EXIT (Rys. 7.). Jeżeli instrukcja EXIT jest umieszczona w pętlach zagnieżdżonych, to wyjście może nastąpić tylko z tej pętli, w której jest zawarta instrukcja EXIT. Natomiast sterowanie jest przekazywane do instrukcji znajdującej się bezpośrednio po pierwszej instrukcji określającej zakończenie wykonywania pętli, która występuje po instrukcji EXIT [6].

SUM:=0	
FOR I:=1 TO 3 DO	(*początek pierwszej pętli*)
FOR J:=1 TO 2 DO	(*początek pętli zagnieżdżonej*)
IF FLAG THEN EXIT; END_IF;	(*wyjście z pętli zagnieżdżonej*)
SUM:=SUM+J;	
END_FOR;	(*koniec pętli zagnieżdżonej*)
SUM:=SUM+I;	
END_FOR;	(*koniec pętli pierwszej*)

Rys. 7. Przykład użycia instrukcji EXIT

2.3.2. Sekwencyjny schemat funkcjonalny SFC

W normie IEC 61131-3 oprócz dwóch języków graficznych i dwóch języków tekstowych, zdefiniowano język SFC. Język ten opisuje zadania sterowania sekwencyjnego za pomocą sieci zawierających kroki i przejścia między tymi krokami. Ten sposób programowania służy do tworzenia odpowiedniej struktury wewnętrznej programu użytkownika i może być wykorzystany do programowania w jednym z omówionych wcześniej języków. Metoda SFC nadaje się szczególnie do programowania algorytmu sterowania procesów, w których występują pewne sekwencje działań, krok po kroku [3].

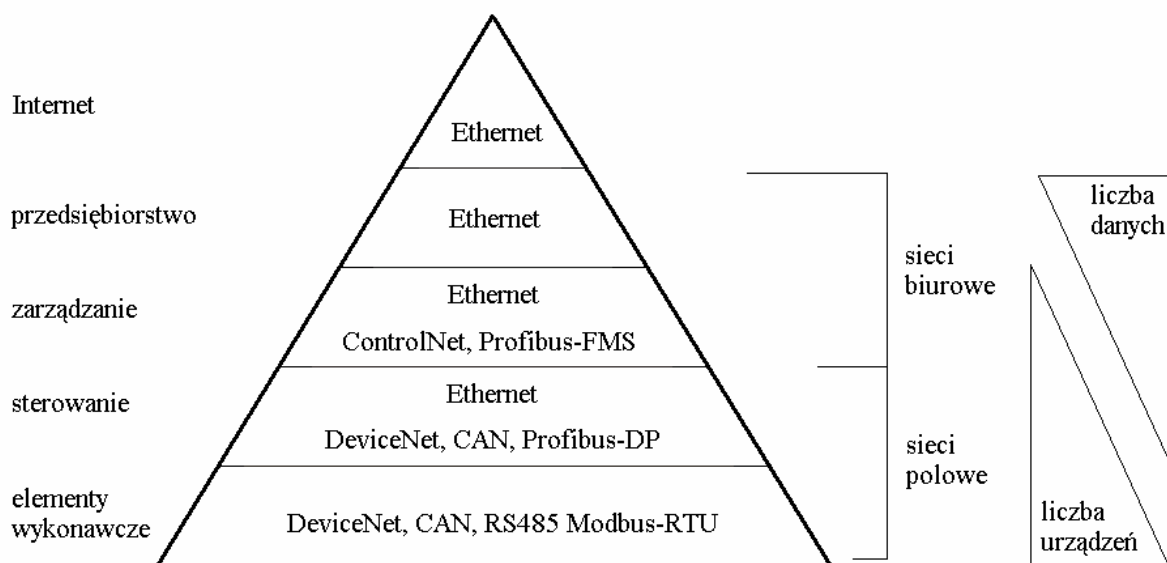
Elementy SFC – kroki i przejścia – są wzajemnie powiązane za pomocą połączeń bezpośrednich. Tworzą one sieć, która stanowi podstawową strukturę w schemacie SFC, podobnie jak obwody w językach graficznych. Z każdym krokiem jest skojarzony odpowiedni zbiór instrukcji, które nazywa się akcjami, a każdemu przejściu między krokami towarzyszy warunek przejścia. Wymienione elementy wymagają przechowywania informacji o ich stanie, dlatego do realizacji oprogramowania zorganizowanego w postaci sekwencyjnego schematu funkcjonalnego są używane tylko bloki funkcjonalne i programy.

Krok przedstawia zachowanie elementu oprogramowania dotyczące stanów jego wejść i wyjść, które określa akcja związana z krokiem. Natomiast przejścia reprezentują warunki logiczne określające stan aktywności kroków. Spełnienie określonych warunków logicznych jest podstawą kontroli przechodzenia od jednego kroku poprzedzającego przejście do drugiego następującego po przejściu. Zatem spełnienie warunku opisującego przejście prowadzi do dezaktywacji kroku poprzedniego i aktywacji kroku następnego. Z każdym krokiem mogą być skojarzone akcje, które mogą stanowić np. zestawy rozkazów lub instrukcji napisanych za pomocą języków programowania PLC. Przyjmuje się, że krok, z którym nie są skojarzone żadne akcje, realizuje funkcję WAIT, co oznacza oczekiwanie na spełnienie warunku przejścia do następnego kroku. Akcje przyporządkowane krokom wyrażają to, co powinno być zrealizowane, gdy następuje uaktywnienie kroku. W szczególności przedstawiają one rozkazy, jakie powinny być wysłane do części wykonawczej sterownika, związane z ustawieniem stanu wyjść [6].

2.4. Sieci przemysłowe

Siecią przemysłową nazywamy zbiór zasobów sprzętowo-programowych rozproszonych na danym obszarze i umożliwiający transmisję danych. Cechą tej sieci jest odporność na zakłócenia przemysłowe, praca w czasie rzeczywistym i hierarchiczność przepływu informacji.

Hierarchiczność przepływu informacji polega na wymianie danych między różnymi poziomami (Rys. 8.). Na najniższym poziomie znajduje się warstwa komponentów, czyli elementy wykonawcze. Następny poziom to warstwa sterowania – sterowniki i sterowanie nadrzędne. Stosowane na tych poziomach sieci pracują w czasie rzeczywistym i noszą nazwę sieci polowych (Fieldbus). W wyższych warstwach (zarządzania produkcją i zarządzania przedsiębiorstwem) stosuje się sieci biurowe. Sieci biurowe zapewniają wymianę informacji między częścią produkcyjną a dyrekcyjną. Połączenia te nie mają czasów krytycznych, natomiast cechuje je duża ilość danych o procesie. Ponadto istnieje możliwość połączenia się z Internetem przy użyciu protokołu TCP/IP [2].



Rys. 8. Hierarchizacja sieci przemysłowej [16]

Można zauważyć, że sieci Ethernet można znaleźć prawie na każdym piętrze piramidy sterowania przemysłowego. Ethernet zawsze znajdował się w górnych warstwach zastosowań, w których występują funkcje zarządzania. Ostatnio przesuwają się on także do dolnej sfery interfejsów Fieldbus, do tej pory zdominowanej przez kilka interfejsów szeregowych jak Modbus czy Profibus. W przemyśle oraz poza nim funkcjonuje szereg odmian Ethernetu.

Wynika to m.in. z kompromisów pomiędzy standardowym Ethernetem a wersjami rozszerzonymi umożliwiającymi pracę w czasie rzeczywistym. Stąd właśnie Ethernet jest często łączony z sieciami CAN, Modbus i Profibus, które mogą działać niezależnie w czasie rzeczywistym. Do głównych rodzajów Ethernetu przemysłowego zalicza się Modbus-TCP, ProfiNet, Ethernet Powerlink, Sercos oraz EtherNet/IP. Każda z odmian Ethernetu ma inne charakterystyczne właściwości i różne mogą być powody jej wyboru w danej aplikacji.

Z Ethernetem łączy się również sieci przemysłowe takie jak DeviceNet i ControlNet opracowane przez firmę Rockwell Automation. DeviceNet działa w ramach sieci CAN, podczas gdy ControlNet ma własną platformę do łączności w czasie rzeczywistym. Wszystko to wiąże razem wspólny protokół przemysłowy CIP (Common Industrial Protocol). Udostępnia on uzgodnioną infrastrukturę powiadamiania. Usługi czasu rzeczywistego zwykle mieszczą się w protokole DeviceNet lub ControlNet, a Ethernet tworzy szkielet infrastruktury. Natomiast Modbus-TCP, podobnie jak CIP zapewnia współdziałanie, ale nie reagowanie ściśle w czasie rzeczywistym. Musi to zostać zrównoważone z szybkością Ethernetu, który dla tego rozwiązania jest szybszy od większości innych wdrożeń. Zatem przydatność Ethernetu będzie tutaj zależała od aplikacji.

W wielu aplikacjach używanie wyłącznie Ethernetu jest niewykonalne. Ale parametry i wydajność mikrokontrolerów z ethernetowym wyposażeniem nieustannie wzrasta, co umożliwia łatwe włączanie również rozszerzonych protokołów Ethernetu jak CIP. Interfejsy fieldbusowe, takie jak CAN, będą wystarczające dla potrzeb sterowania przemysłowego u dołu piramidy. Ethernet może istnieć od góry do dołu całej hierarchii [15].

2.5. Systemy SCADA

Wraz z zastosowaniem komputerów osobistych do programowania sterowników PLC znacznie rozwinęły się możliwości programowe i komunikacyjne sterowników pracujących często w systemach rozproszonych. Przyczynił się do tego dynamiczny rozwój sprzętu i oprogramowania komputerowego, a w szczególności oprogramowanie do celów sterowania nadrzędnego i zbierania danych SCADA (Supervisory Control And Data Acquisition). Systemy SCADA, które zapewniają komunikację ze sprzętem PLC oraz możliwość zmian oprogramowania sterowników w systemach sieciowych spowodowały, że znacznie wzrósł zakres zastosowań i możliwości systemów sterowania [3].

Systemy SCADA składają się z [2]:

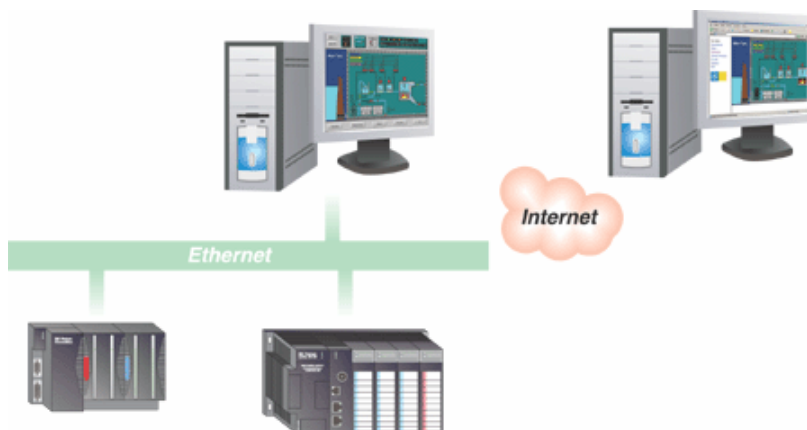
- części sprzętowej (komputery PC, sterowniki PLC/PAC, panele operatorskie, aparatura pomiarowa, elementy wykonawcze, urządzenia sieciowe),
- części programowej (system operacyjny, oprogramowanie narzędziowe do tworzenia aplikacji typu SCADA, programy komunikacyjne).

2.5.1. Interfejs człowiek-maszyna (HMI)

Układ składający się ze sterownika PLC sterującego dowolnym obiektem lub procesem przemysłowym, może działać poprawnie, lecz obsługa bez podłączenia dodatkowego urządzenia, np. programatora lub komputera, nie ma możliwości poglądu ani wpływu na proces poprzez zmianę nastaw, wartości zadanych, itp. W celu umożliwienia obsłudze prowadzenia kontroli procesu i ewentualnie jego korygowanie, stosuje się monitoring i wizualizację procesu. Takie możliwości daje system SCADA. Zastosowanie monitoringu wymaga połączenia sterownika PLC z komputerem PC, na którym zainstalowane zostało odpowiednie oprogramowanie, zapewniające możliwość współpracy ze sterownikami programowalnymi (Rys. 9.).

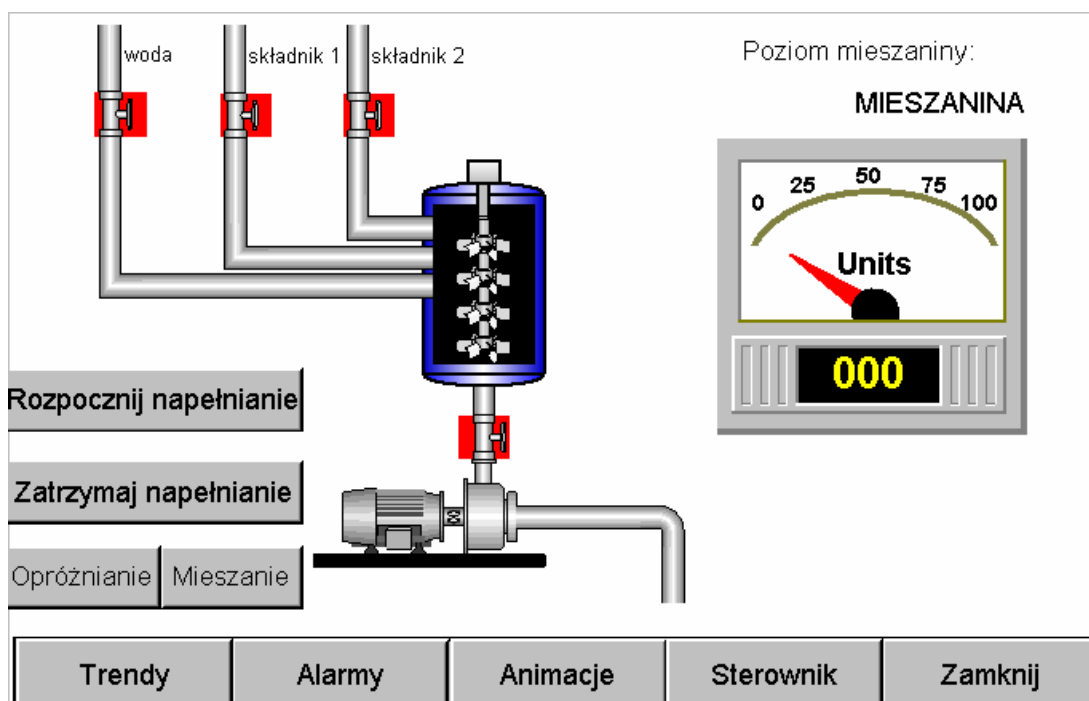
Do najważniejszych zadań systemu monitoringu i wizualizacji należy zaliczyć [5]:

- skupienie informacji w określonym miejscu (dyspozytorni),
- prezentacja informacji w sposób zgodny z potrzebami dyspozytora,
- możliwość oddziaływania na proces (nadrzędne sterowanie).



Rys. 9. Przykład wizualizacji i monitoringu za pomocą systemu SCADA [12]

Prezentacja danych może odbywać się za pomocą różnych narzędzi. Najczęściej są to animowane ekrany (Rys. 10.), wykresy oraz raporty. Obecnie do monitoringu używa się komputerów PC, często połączonych w układy sieciowe, z zainstalowanym oprogramowaniem SCADA. Takie stacje operatorskie przewyższają wygodą obsługi różnorodne wyświetlacze i panele operatorskie. W przypadku systemów SCADA panele takie nie muszą być stosowane, ewentualnie mogą stanowić pomoc dla stacji operatorskich jako interfejsy człowiek-maszyna HMI (Human-Machine Interface). Systemy SCADA zastąpiły konwencjonalne metody wizualizacji [5].



Rys. 10. Przykład animowanego ekranu sterowniczego [12]

2.5.2. Oprogramowanie i protokoły komunikacyjne

Oprogramowanie SCADA funkcjonuje na określonym zestawie zmiennych, które tworzone są i edytowane przy użyciu programu zwanego konfiguratorem. Każda zmienna połączona jest ze slotem, w którym znajduje się opis do zmiennej. Pisanie oprogramowania zaczynamy od utworzenia ekranów synoptycznych, obrazów procesów technologicznych składających się z obiektów graficznych, których zachowanie jest wcześniej zdefiniowane przez programistę. Stosuje się obiekty graficzne bierne i czynne. Obiekty czynne są połączone ze zmiennymi, aby np. poruszały się na ekranie synoptycznym.

Bardzo ważnym elementem oprogramowania typu SCADA są protokoły komunikacyjne w nich stosowane. Rozwój systemów automatyki przemysłowej nie byłby możliwy bez opracowania standardowych modułów programowych do wykorzystania w wielu systemach pochodzących od różnych producentów.

Na przełomie lat 80 i 90-tych powstała technologia DDE (Dynamic Data Exchange), która za pośrednictwem standardowych interfejsów (RS232) ułatwiała wymianę danych między aplikacjami w systemie Windows, takich jak Excel, Word oraz oprogramowanie SCADA. Protokół ten jest wykorzystywany w implementacji architektury klient/serwer między dwoma aplikacjami. Serwerem jest aplikacja, która udostępnia swoje dane innym aplikacjom – klientom. Klientem jest aplikacja, która otrzymuje dane w ramach protokołu DDE. Aplikacja może równocześnie funkcjonować jako serwer oraz jako klient [2].

Powstały dwie odmiany protokołu DDE. FastDDE umożliwia upakowanie wielu komunikatów w pojedynczy komunikat, co zwiększa efektywność oraz szybkość komunikacji poprzez zredukowanie ilości danych wymienianych za pomocą mechanizmu DDE pomiędzy klientem a serwerem. Druga odmiana NetDDE zwiększa funkcjonalność standardowego mechanizmu DDE pozwalając na komunikowanie się za pomocą sieci lokalnych i złączy szeregowych. Pozwala to na wymianę danych za pośrednictwem mechanizmu DDE pomiędzy aplikacjami pracującymi na różnych komputerach, połączonych siecią lokalną, lub też za pomocą modemów [10].

W latach 90-tych wprowadzono mechanizm łączenia i osadzania obiektów z różnych aplikacji OLE (Object Linking and Embedding). Opracowano standard komunikacji OPC (OLE for Process Control) pomiędzy oprogramowaniem przemysłowym a programami komunikacyjnymi do sterowników PLC. Specyfikacja standardu wykorzystuje mechanizm OLE, który korzysta z technologii COM (Component Object Model) definiowania i tworzenia interfejsów na poziomie binarnym dla komponentów programowych, którą opracowano w firmie Microsoft. Standard COM definiuje komponenty programowe niezależnie od języka programowania, co pozwala na włączenie do tworzenia aplikacji elementów należących do innych programów i wymianę danych między obiektami za pomocą wspomnianych interfejsów. Podzbiorem komponentów typu COM są kontrolki ActiveX. Stosowane są w programach pisanych za pomocą takich narzędzi jak Visual Basic, C++, Java i wielu innych. Technologia ActiveX jest modularna i umożliwia przekazywanie danych pomiędzy różnymi aplikacjami działającymi pod systemem operacyjnym Windows.

Standard OPC opiera się na architekturze klient/serwer. Klient OPC może połączyć się z serwerem OPC dostarczonym przez różnych producentów. Ponadto może pracować jako

moduł, który pozwala aplikacjom korzystać z danych OPC. Klient OPC może być zainstalowany na tym samym komputerze, co serwer lub może być zainstalowany na oddzielnych komputerach.

Obok wymienionych popularnych protokołów pozwalających na wymianę danych między aplikacjami, tj. protokoły z rodziny DDE oraz standardu komunikacyjnego OPC z interfejsami COM, firmy mają swoje autorskie protokoły [2].

2.5.3. Główne cechy systemów SCADA

Do głównych cech systemów SCADA należą: sterowanie procesami, akwizycja danych w czasie rzeczywistym, współpraca z rozproszonymi systemami sterowania, wizualizacja stanu procesu, raportowanie, sygnalizacja pojawiających się stanów alarmowych, skalowalność, otwartość systemu, zabezpieczenia dostępu, redundancja.

Sterowanie procesami i akwizycja danych w czasie rzeczywistym

Dane procesowe muszą być przesyłane pomiędzy sterownikiem a serwerem SCADA w możliwie krótkim czasie. W przypadku istotnych informacji czas ten nie powinien przekraczać dziesiątych części sekundy. Jednocześnie system powinien zapewnić możliwość zróżnicowania sposobu odczytywania i uaktualniania danych. Przetwarzanie danych może być cykliczne lub asynchroniczne. Cyklicznie są przetwarzane analogowe i cyfrowe sygnały pomiarowe, natomiast asynchronicznie zmienne wprowadzane przez obsługę. Przetwarzanie cykliczne wykonuje się z różnymi deklarowanymi okresami, które zależą od przetwarzanego sygnału (inny jest okres przetwarzania sygnałów z przetworników pomiarowych przepływu i inny dla sygnałów pomiarowych temperatury) [5].

Współpraca z rozproszonymi systemami sterowania

Systemy SCADA umożliwiają stworzenie zdecentralizowanej, przestrzennie rozproszonej struktury systemu nadzorowania i wizualizacji procesu. Dane procesowe mogą być archiwizowane na dowolnym serwerze SCADA. Ilość danych, które mogą być gromadzone w systemie zależy od liczby serwerów, pojemności pamięci masowych i sposobu kompresji danych. W systemie sieciowym wszystkie dane z dowolnej stacji systemu są dostępne dla wszystkich innych stacji [5].

Wizualizacja stanu procesu i raportowanie

Jednym z najważniejszych zadań systemów SCADA jest wizualizacja procesu przy użyciu animowanych ekranów synoptycznych, które zapewniają wysoki poziom grafiki (najczęściej jest to grafika wektorowa lub rastrowa). Dane procesowe mogą być prezentowane w formie wykresów tworzonych na bieżąco, pokazujących aktualny przebieg danych lub wykresów historycznych, pokazujących przebieg danych w przeszłości. Każdy proces ma swoją specyfikę i wymaga opracowania dla niego odpowiednich raportów. Istnieje możliwość inicjacji raportów na żądanie, cyklicznie lub po wystąpieniu określonego zdarzenia [5].

Sygnalizacja pojawiających się stanów alarmowych

Ważnym zadaniem systemów SCADA jest sygnalizacja wykrytych alarmów. Generowane mogą być różne ich rodzaje: systemowe o wykrytych uszkodzeniach sprzętowych, błędach programowych, błędach transmisji oraz ostrzeżenia technologiczne informujące o przekroczeniach dopuszczalnych granic, szybkości zmian zmiennych procesowych lub nieprawidłowych stanach zmiennych cyfrowych. Alarmy sygnalizowane są na ekranach synoptycznych. System SCADA zapewnia filtrację alarmów. Dla każdego użytkownika powinna istnieć możliwość ograniczenia strumienia sygnalizowanych alarmów, aby uzyskiwał on tylko te informacje alarmowe, które go bezpośrednio dotyczą. Zabezpiecza to obsługę przed przeciążeniem spowodowanym nadmiernym nagromadzeniem informacji. Systemy SCADA posiadają także możliwość archiwizacji zgłaszanych alarmów oraz działań operatora. Pozwala to na analizę ewentualnych przyczyn sytuacji krytycznych w układzie [5].

Skalowalność i otwartość systemu

Oprogramowanie SCADA cechuje się skalowalnością, czyli możliwością rozbudowy sprzętowej i programowej bez konieczności dokonywania istotnych zmian w istniejącej strukturze urządzeń i programu.

Systemy SCADA współpracują ze sterownikami i innymi urządzeniami automatyki produkowanymi przez różnych producentów. Dają możliwość indywidualnego pisania oprogramowania, współpracy z arkuszami kalkulacyjnymi i bazami danych, możliwość wymiany danych z innymi systemami, wykorzystywanie standardowego systemu

operacyjnego i oprogramowania sieciowego, zastosowanie standardowych sieci lokalnych oraz współpraca ze standardowymi sieciami polowymi [5].

Zabezpieczenia dostępu

Systemy SCADA mają rozbudowane możliwości zabezpieczenia określonych funkcji przed dostępem niepowołanych osób. Zapewniają możliwość przydzielenia poszczególnym użytkownikom różnych uprawnień do administracji systemem, sterowania procesem, modyfikacji parametrów, dostępu do danych. Każdy użytkownik powinien na początku pracy zalogować się do systemu. Inne uprawnienia muszą mieć: projektant systemu, inżynier, operator, itd. [5].

Redundancja – dodatkowy stopień niezawodności

Dla podwyższenia niezawodności systemów automatyki wprowadzono sprzętowe i programowe rozwiązania redundancyjne. Polegają one na podwojeniu elementów systemu oraz na zapewnieniu komunikacji pomiędzy tymi elementami. Również producenci systemów SCADA przewidują takie rozwiązania, wprowadzając redundancje stacji operatorskich, serwerów oraz redundancje połączeń sieciowych pomiędzy stacjami systemu oraz stacjami operatorskimi i sterownikami programowalnymi. Układy takie nazywane są Hot Standby Redundancy („gorąca rezerwa”) [5].

Omówione główne cechy systemów SCADA występują w różnym stopniu w konkretnych rozwiązaniach. Zależy to od projektu, stopnia zaawansowania systemu, jak i od cech użytkowych. Systemy te podlegają stałemu rozwojowi, co ma wpływ na zmianę ich własności użytkowych [5]. Do najpopularniejszych pakietów SCADA/HMI w Polsce należy zaliczyć [7]: InTouch firmy Wonderware, WinCC – Siemens, iFIX – GE Fanuc, Asix – Askom, Vijeo Citect – Schneider Electric, ControlMaestro – Elutions, FT View SE – Rockwell Automation i inne. Natomiast obszary najczęstszego wykorzystania oprogramowania SCADA/HMI przez polskich klientów to: przemysł spożywczy, energetyka, przemysł chemiczny, sektor wodno-kanalizacyjny, przemysł maszynowy, automatyka budynkowa, ochrona środowiska oraz inne [7].

3. Wybrane oprogramowanie SCADA

Wiodącymi markami, utrzymującymi prym na światowym rynku systemów w zakresie nadrzędnej wizualizacji procesów, są dwie amerykańskie firmy: Wonderware oraz Rockwell Automation. Oprogramowanie do projektowania aplikacji typu SCADA udostępnione przez te firmy to odpowiednio: InTouch i FactoryTalk View Site Edition.

3.1. Wonderware InTouch

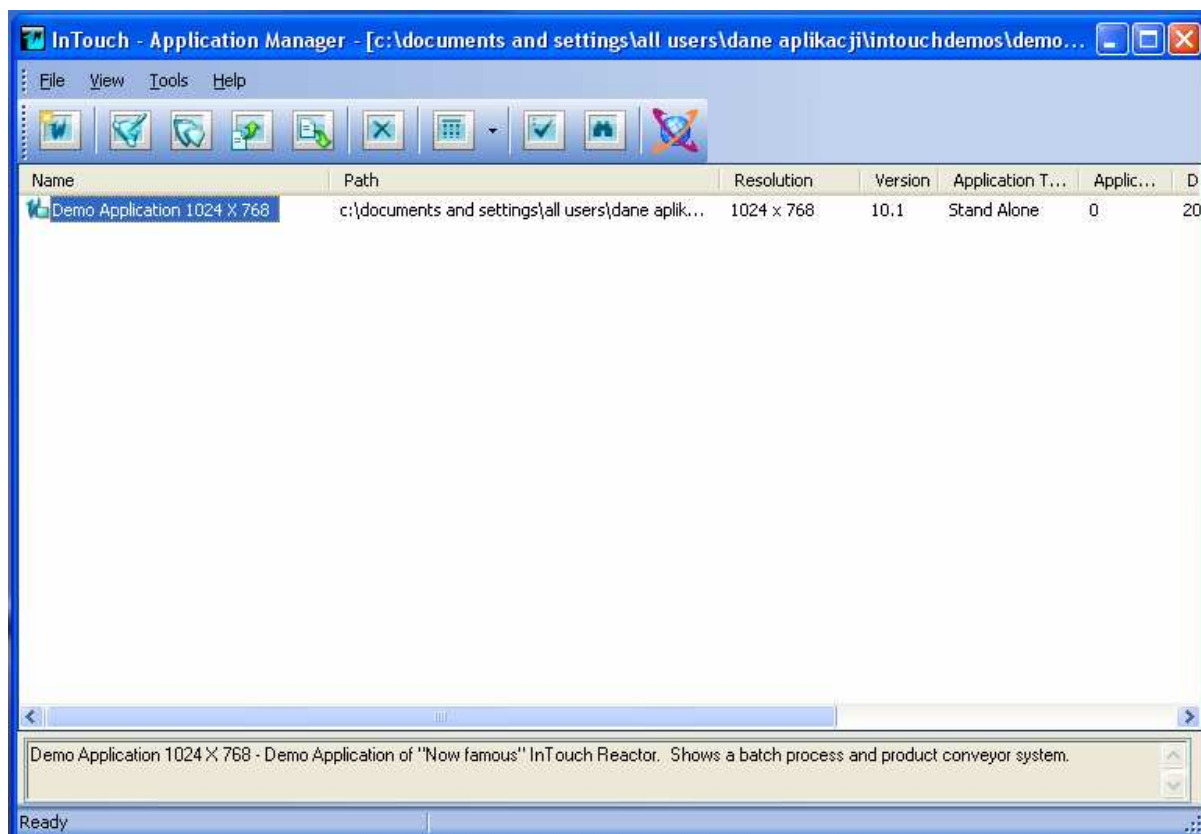
Oprogramowanie InTouch to najpopularniejszy na świecie pakiet wizualizacyjny, który jest używany w niemal każdej gałęzi przemysłu. InTouch sprawia, że inżynierowie mogą stać się bardziej efektywni i pomysłowi, ponieważ ułatwia budowanie intuicyjnych, bezpiecznych i łatwych w obsłudze aplikacji HMI oraz SCADA, jednocześnie umożliwiając obniżenie kosztów wdrożenia [13].

3.1.1. Narzędzia projektowe

InTouch to przemysłowe oprogramowanie zaprojektowane do wizualizacji oraz kontroli procesów produkcyjnych. Jest komponentem pakietu Wonderware Development Studio wykorzystującym nowatorską technologię ArchestrA. Może być zintegrowane z Platformą Systemową Wonderware, umożliwiając użytkownikom oprogramowania korzystanie z najnowszych rozwiązań w dziedzinie aplikacji przemysłowych, zachowując przy tym korzyści z inwestycji już poczynionych. Oferuje łatwe w użyciu środowisko do projektowania aplikacji oraz rozległą funkcjonalność umożliwiającą szybkie projektowanie, testowanie oraz wdrażanie systemów udostępniających użytkownikom dane bezpośrednio z systemów sterownia i produkcji. InTouch to także otwarte i elastyczne oprogramowanie, umożliwiające dostosowanie aplikacji do aktualnych potrzeb przy zachowaniu szerokiego wachlarza połączeń z urządzeniami oraz systemami spotykanymi w przemyśle [13].

Tworzenie aplikacji w InTouch'u od wersji 10.0 możliwe jest na dwa sposoby. Starszy sposób prowadzi przez Menedżer Aplikacji (Rys. 11.). Menedżer służy do zarządzania aplikacjami. Daje możliwość tworzenia nowych aplikacji, otwierania istniejących aplikacji w programach WindowMaker i WindowViewer, uruchamiania programów narzędziowych dla

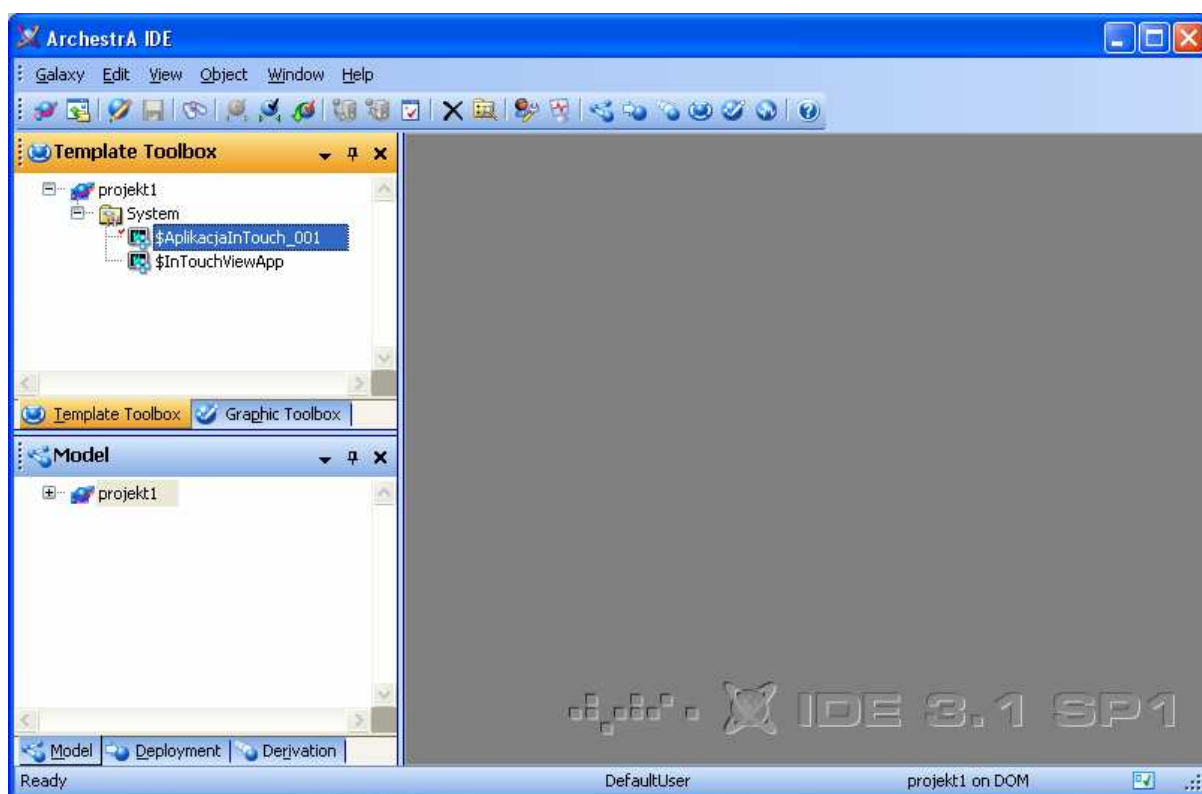
listy zmiennych (DBDump oraz DBLoad), tworzenia aplikacji sieciowych NAD (Network Application Development) zarówno dla architektury klienta i serwera, oraz konfigurowania dynamicznej konwersji rozdzielczości pozwalającej wszystkim stacją na niezależny podgląd aplikacji, nawet jeżeli pracują one przy innych rozdzielczościach ekranu [10].



Rys. 11. Menedżer aplikacji

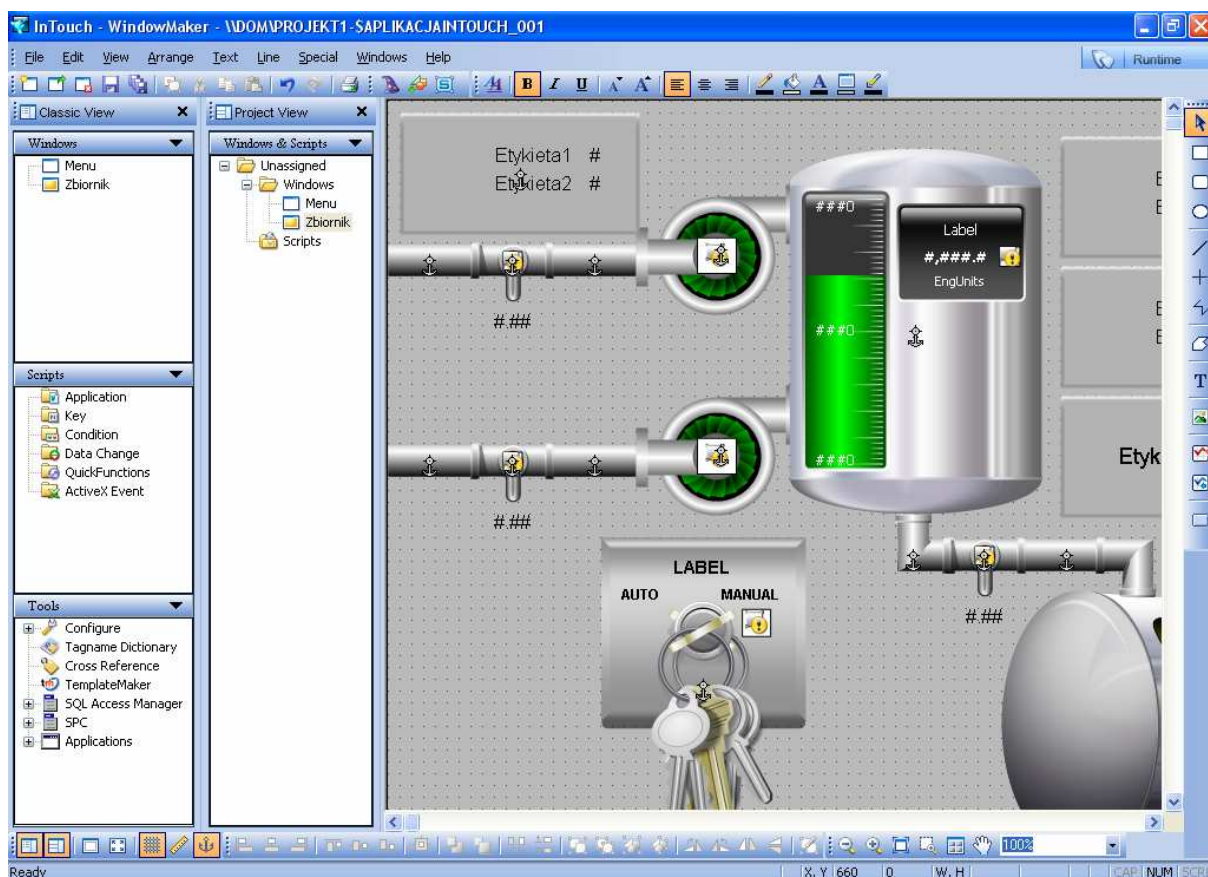
Poza ścieżką prowadzącą przez Menedżer Aplikacji, możliwe jest także tworzenie aplikacji z poziomu zintegrowanego środowiska projektowego (Rys. 12.) ArchestrA IDE (Integrated Development Environment). Aplikacja w tym przypadku, zamiast w plikach, zostaje założona w centralnej bazie aplikacji (strukturze), tzw. Galaxy. Zapis w takiej strukturze umożliwia łatwiejsze zarządzanie aplikacją oraz zapisywanie informacji o edycji i wprowadzonych przez projektanta zmianach. Poprzez łatwiejsze zarządzanie i dystrybucję na inne komputery w sieci, aplikacje tworzone w IDE określa się mianem „aplikacji zarządzalnych” [8].

Niezależnie od sposobu tworzenia aplikacji, narzędziem do konfiguracji ekranów z obiektowo zorientowaną grafiką oraz logiki działania jest WindowMaker (Rys. 13.). Ekrany te mogą być połączone z przemysłowymi systemami wejścia/wyjścia, jak również z innymi aplikacjami Microsoft Windows. W celu przyspieszenia procesu tworzenia okien wizualizacyjnych można używać gotowych kontroltek ActiveX lub korzystać z biblioteki symboli Symbol Faktory oraz ArchestrA Graphic (dostępne od wersji InTouch 10.0 z poziomu ArchestrA IDE). ArchestrA Graphic to biblioteka zawierająca ponad 500 zaawansowanych obiektów graficznych powszechnie używanych w przemyśle wraz z narzędziem do ich edycji oraz tworzenia własnych szablonów obiektów. Nowy silnik graficzny pozwala tworzyć grafiki dokładnie oddające rzeczywistą postać fizycznych układów. Każdy obiekt można ożywić poprzez sekwencję animacyjną elementów, z których jest złożony. Sekwencje tworzy się wykorzystując rozbudowane połączenia animacyjne i skrypty [8]. Eksplorator aplikacji programu WindowMaker to efektywna metoda przechodzenia pomiędzy elementami aplikacji InTouch oraz konfigurowania ich. Zapewnia on łatwy dostęp do skryptów oraz okien pogrupowanych w logiczne obszary [10].



Rys. 12. Środowisko projektowe IDE

Program WindowViewer jest środowiskiem, w którym uruchamiane są aplikacje stworzone wcześniej za pomocą programu WindowMaker. Program wykonuje skrypty, gromadzi dane archiwalne, przetwarza sygnały o alarmach, tworzy sprawozdania, może pracować jako klient lub jako serwer, zarówno dla protokołu komunikacyjnego DDE, jaki i dla protokołu SuiteLink [10].

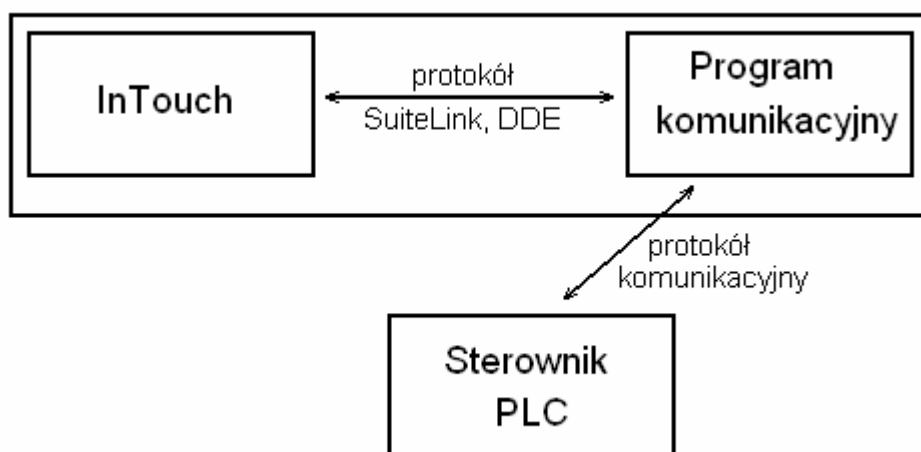


Rys. 13. WindowMaker

3.1.2. System komunikacji

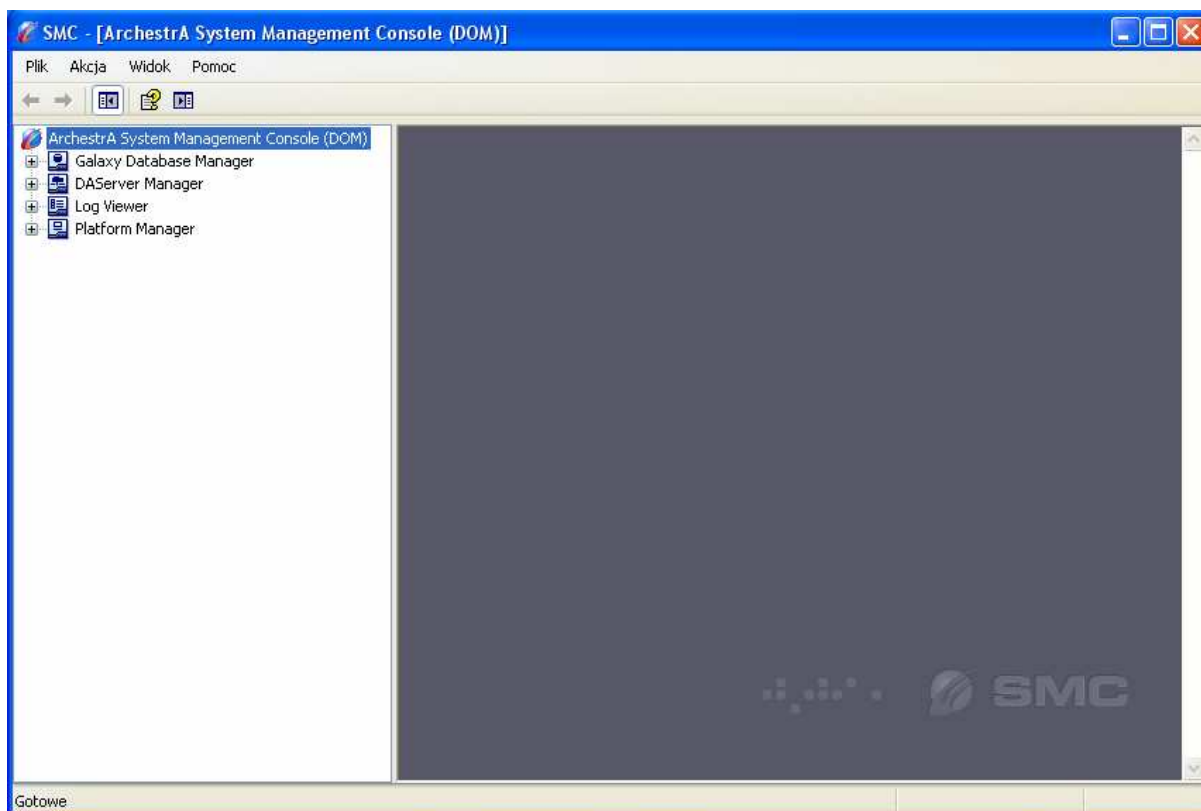
Wonderware wykorzystując standardy komunikacyjne połączył je z rozwojową technologią Microsoft i w efekcie daje użytkownikom bardziej otwarte narzędzia do tworzenia aplikacji. InTouch od wersji 10.0 jest zgodny z systemem Microsoft Windows Vista oraz współpracuje z oprogramowaniem Microsoft Office 2007. Obsługuje wiele standardów przemysłowych i umożliwia stosowanie własnych standardów każdego przedsiębiorstwa, zapewniając spójność aplikacji [13].

InTouch umożliwia połączenie z niemal każdym rodzajem urządzeń automatyki za pośrednictwem programów komunikacyjnych i serwerów OPC (Rys. 14.). Programy komunikacyjne zapewniają dostęp do aplikacji za pośrednictwem: standardu OPC, protokołu Wonderware SuiteLink oraz protokołu DDE. Firma Wonderware wychodząc naprzeciw potrzebom użytkowników dostarcza nową generację programów komunikacyjnych o nazwie DAServers (Data Access Server), które zaprojektowano przy użyciu technologii ArchestrA, wprowadzając nową jakość do tworzenia aplikacji wymagających zarządzania komunikacją z urządzeniami sterującymi. Środowisko konfiguracyjne ArchestrA System Management Console (Rys. 15.) umożliwia lokalne i zdalne zarządzanie programami komunikacyjnymi z jednego miejsca w firmie we wszystkich aplikacjach używających tych programów. W ten sposób jedno, wspólne środowisko konfiguracyjne ułatwia i przyspiesza zarządzanie i administrację programami DAServers, nawet jeśli zostały zainstalowane na różnych komputerach [9].



Rys. 14. Schemat komunikacji

Każdy program komunikacyjny DAServer posiada możliwość mapowania adresów, a więc zamiast odwoływać się do sterownika, stosując notację obsługiwaną przez protokół komunikacyjny można zastosować dowolne nazewnictwo wygodne dla osoby konfigurującej program. Jest to bardzo przydatne, ponieważ w przypadku konieczności zmiany odczytywanych adresów z systemu sterowania nie trzeba wprowadzać zmian w zaprojektowanej aplikacji. W takim przypadku wystarczy tylko zmienić ustawienia w samym programie komunikacyjnym [9].



Rys. 15. Środowisko konfiguracyjne programów DAServers

SuiteLink jest protokołem komunikacyjnym stworzonym przez firmę Wonderware. Ma on zapewnić wydajny i stabilny mechanizm komunikacji zoptymalizowany na przesyłanie dużej ilości danych szybkozmiennych dla Windows NT. Jest zgodny z protokołem OPC jeśli chodzi o format przesyłanych danych, tzn. ramka zawiera informacje na temat wartości zmiennej, czasu zmiany tej wartości oraz wiarygodności informacji zależnej od jakości łącza – w skrócie VTQ (Value, Time, Quality). Jednocześnie SuiteLink używa tego samego nazewnictwa w zakresie konfiguracji komunikacji co DDE, co pozwala na uaktualnianie istniejących aplikacji korzystających z DDE do protokołu SuiteLink bez żadnych zmian w tych aplikacjach. SuiteLink opiera się bezpośrednio na protokole TCP/IP, umożliwiając komunikację w systemie Windows NT poprzez stworzenie połączenia typu P2P (peer-to-peer) między komponentami programów Wonderware, przez co jest protokołem bardziej wydajnym niż OPC (wykorzystujący OLE zaprojektowany w pierwszym rzędzie do przesyłania skomplikowanych obiektów jak arkusze kalkulacyjne, dokumenty tekstowe, itd.) [2].

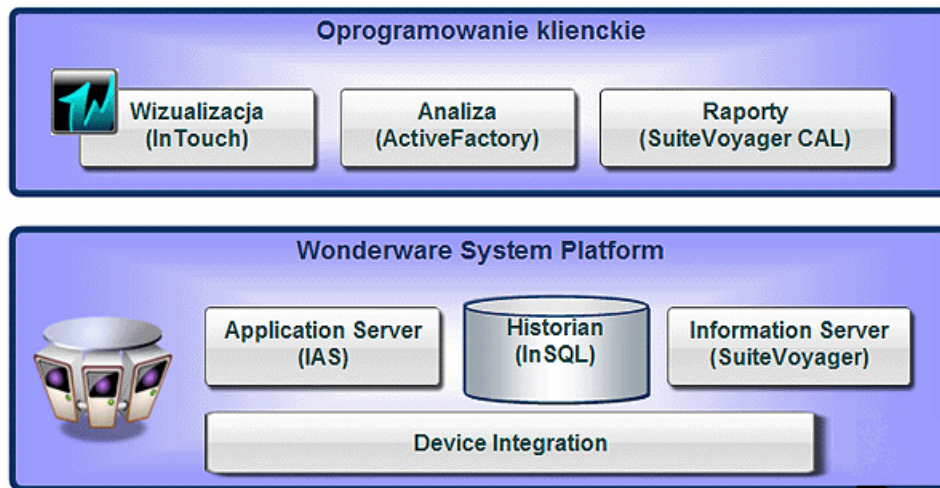
3.1.3. Cechy oprogramowania Wonderware

Platforma Systemowa Wonderware oraz technologia ArchestrA

Integracja oprogramowania InTouch z platformą przemysłową ArchestrA, opartą na technologii .NET firmy Microsoft umożliwia tworzenie bardzo rozbudowanych aplikacji wizualizacyjnych pracujących w systemie rozproszonym bez konieczności dublowania logiki działania systemu na wszystkich stacjach operatorskich. Odciąża to projektantów od ewentualnych modyfikacji każdej z aplikacji pracujących na stacjach roboczych na rzecz modyfikacji systemu na serwerze aplikacyjnym Platformy Systemowej Wonderware.

Platforma Systemowa Wonderware (Rys. 16.) to jednolita platforma składająca się z zestawu usług oraz aplikacji opartych o technologię ArchestrA, która pozwala w efektywny sposób tworzyć oraz rozwijać rozproszone aplikacje przemysłowe na wielu poziomach zarządzania informacją. System ten obejmuje zarówno komunikację z urządzeniami produkcyjnymi, warstwę pośrednie zarządzania informacją z produkcji jak i warstwę integracji z systemami służącymi wspomaganie zarządzania przedsiębiorstwem. W skład Platformy Systemowej wchodzi: Serwer Aplikacji (Wonderware Application Server), przemysłowa baza danych Wonderware Historian oraz przemysłowy serwer informacyjny Wonderware Information Server. Dodatkowo Platforma Systemowa Wonderware posiada zestaw narzędzi klienckich pozwalających na tworzenie aplikacji umożliwiających wizualizację, analizę oraz tworzenie raportów [13]. Wonderware Application Server jest silnikiem platformy, którego cechą charakterystyczną jest możliwość rozproszenia aplikacji na wiele stanowisk komputerowych lub serwerów przy jednoczesnym utrzymaniu jednolitości aplikacji [2].

ArchestrA jest zaawansowaną architekturą z zakresu automatyki przemysłowej, zaprojektowaną w celu przedłużenia cyklu życia systemów, istniejących od lat w przedsiębiorstwie. Możliwe jest to dzięki zastosowaniu najnowszych technologii informatycznych i otwartości na inne, działające już w przedsiębiorstwie systemy automatyki. ArchestrA jest sposobem na integrację aplikacji przemysłowych działających na platformach Microsoftu, umożliwiającym skorzystanie ze wszystkich najnowszych technologii, dostępnych pod systemami firmy Microsoft [13].



Rys. 16. Elementy składowe Platformy Systemowej Wonderware[13]

Elastyczna architektura i skalowalność

Aplikacje InTouch'a mogą działać jako stacje pojedyncze, co jest użyteczne dla firm gdzie jeden komputer nadzoruje jeden proces produkcyjny. Każda stacja jest w pełni niezależna, choć możliwe jest połączenie tych stacji bez ponoszenia dodatkowych kosztów. Aplikacje mogą także działać jako integralna część rozbudowanego systemu informatycznego, opartego na architekturze ArchestrA i Platformie Systemowej.

Jeden lub kilka komputerów może działać jako Tag Server (serwer zmiennych) posiadający listę wszystkich zmiennych używanych w aplikacjach, zarządzający ich logowaniem historycznym, uruchamianiem skryptów, alarmami oraz połączeniami z urządzeniami zewnętrznymi. Aplikacje uruchomione na stacjach klienckich łączą się z serwerem zmiennych i udostępniają dane operatorom. Natomiast dynamiczna konfiguracja sieciowa NAD ułatwia centralne zarządzanie aplikacją przez jeden serwer sieciowy. Każda stacja kliencka tworzy lokalną kopię aplikacji serwera. Daje to możliwość tworzenia systemów redundantnych pracujących także bez włączonego serwera. Inną ważną cechą NAD jest uaktualnianie aplikacji klienckich bez ich restartowania. System może jedynie powiadomić operatora o dokonywanych zmianach i podmienić wyłącznie części aplikacji, które uległy zmianie.

Dostępna jest również wersja terminalowa (Terminal Services) oprogramowania InTouch, umożliwiająca uruchomienie wielu sesji terminalowych wizualizacji pod kontrolą jednego serwera terminalowego.

InTouch jest w pełni skalowalny. Możesz rozpocząć pracę od zaprojektowania prostej aplikacji, którą potem łatwo można rozbudować nawet do dużego i skomplikowanego systemu rozproszonego [13].

Stany alarmowe i bezpieczeństwo

Użytkownicy systemu używają listy alarmów do śledzenia zdarzeń zaistniałych podczas całego procesu produkcji. Natychmiastowe prezentowanie alarmów oraz umożliwienie ich szybkiego potwierdzenia pozwala skrócić czasy przestojów oraz przyspieszyć reagowanie na sytuacje krytyczne. InTouch oferuje kilka różnych sposobów wglądu w system alarmowy. Każdy z nich może być użyty osobno bądź wszystkie razem.

Do wyświetlania alarmów można wykorzystać kontrolki ActiveX. Kontrolka Alarm Database View Control prezentuje alarmy historyczne zgromadzone w bazie danych i umożliwia ich łatwe i szybkie filtrowanie. Alarm Viewer Control pozwala na pokazywanie alarmów bieżących. Przez łatwą konfigurację kontrolki można szybko zmienić sposoby prezentacji oraz sortowanie informacji o alarmach. Alarm TreeView Control wyświetla aktywne alarmy w postaci przejrzystej struktury drzewa. Natomiast Distributed Alarm Display pokazuje alarmy bieżące z aplikacji rozproszonej.

Alarmy mogą być uaktywniane lub dezaktywowane bezpośrednio lub pośrednio przy użyciu zmiennych aplikacji. Sposób wyświetlania alarmów może być uzależniony od typu alarmu lub zmiennej. Operator ma trzy sposoby potwierdzania alarmów: warunkowy, zdarzeniowy - wymagający potwierdzenia najnowszych zdarzeń, rozszerzony - zezwalający na potwierdzanie każdej transakcji związanej z systemem alarmowym [13].

W oprogramowaniu InTouch możliwe jest stosowanie trzech poziomów dostępu. Aplikacyjny poziom dostępu decyduje o tym, jakie informacje są widoczne lub które parametry mogą być modyfikowane przez użytkowników posiadających własne hasła. Systemowy poziom dostępu pozwala przydzielać dostęp na poziomie kontrolera domeny lub lokalnego komputera bazując na identyfikatorze użytkownika lub grupy użytkowników. Ten system bezpieczeństwa ułatwia zarządzanie prawami dostępu użytkowników i administrację hasłami. Natomiast użytkownicy wykorzystujący Platformę Systemową mogą w aplikacjach w pełni korzystać z rozbudowanego systemu bezpieczeństwa [13].

Skrypty

W InTouch'u szeroko wykorzystywana jest technologia .NET firmy Microsoft. Oznacza to między innymi możliwość zagnieżdżenia kontrolek .NET w aplikacjach, a także wykorzystania funkcji .NET z poziomu języka skryptowego [13].

Jednym z najbardziej istotnych elementów funkcjonalności aplikacji InTouch jest możliwość korzystania ze skryptów. Skrypty dają możliwość wykonywania poleceń, oraz operacji logicznych, w zależności od spełnienia określonych kryteriów. Funkcje skryptowe QuickScript są skryptami, które można wywoływać z poziomu innych skryptów i połączeń animacyjnych. Kod źródłowy takiej funkcji pamiętany jest w jednym miejscu, tak więc jej edycja umożliwia wprowadzenie zmian we wszystkich aplikacjach, w których funkcja taka jest wywoływana. Wszystkie skrypty są sterowane zdarzeniami. Kolejność przetwarzania zależna jest od aplikacji. Podstawowe rodzaje skryptów to: aplikacji (obejmujące swoim zasięgiem całą aplikację), okien (powiązane z konkretnym oknem), klawiszowe (połączone z określonym klawiszem lub kombinacją klawiszy na klawiaturze), warunkowe (połączone ze zmienną dyskretną), zamiany wartości (związane tylko ze zmienną), przycisku (powiązany z obiektem, dla którego zdefiniowano połączenie dotykowe – przycisk) [10].

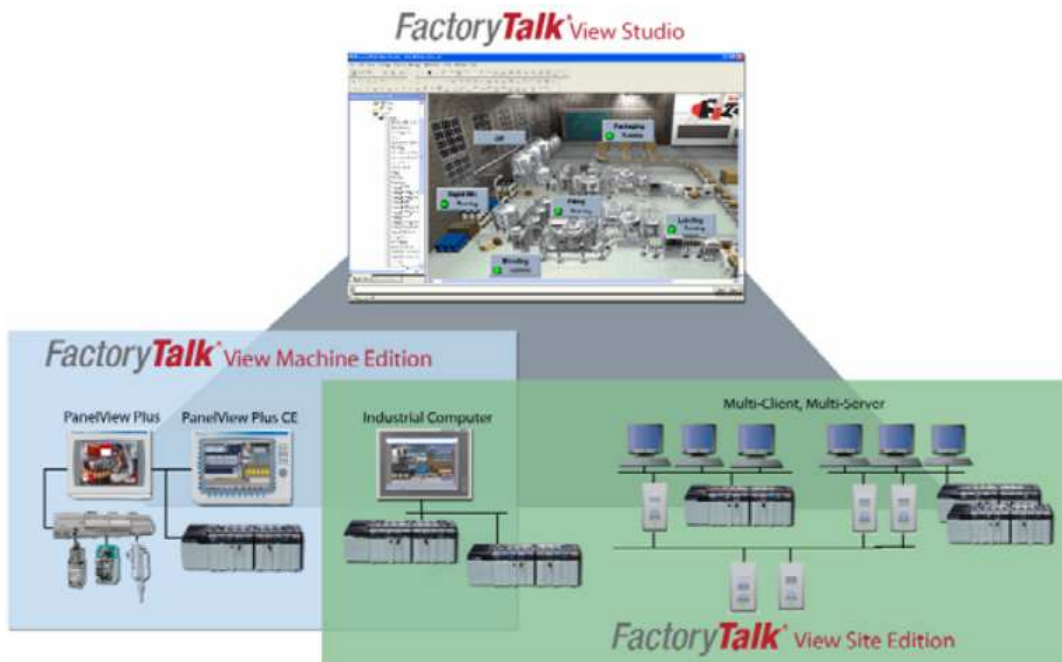
Użyteczny i intuicyjny edytor skryptów (QuickScript Editor), umożliwia poszerzenie funkcjonalności aplikacji oraz lepsze jej dopasowanie do potrzeb użytkownika, czyniąc z pakietu Wonderware InTouch jeden z najbardziej elastycznych systemów klasy SCADA. Jest on tak zaprojektowany, że osoby nie będące programistami mogą korzystać z jego funkcjonalności. Posiada wbudowany mechanizm diagnostyki definiowanych skryptów, czyli sprawdzania poprawności składni skryptów jeszcze przed uruchomieniem aplikacji. Skrypty można konfigurować na różne sposoby tak, aby były wykonywane w odpowiedzi na różne zdarzenia i warunki. Użytkownicy mogą również rozwijać bibliotekę skryptów, funkcji (QuickFunctions), które mogą być wielokrotnie wykorzystywane, co ułatwia projektowanie aplikacji i jej późniejsze modyfikowanie [13].

3.2. Rockwell FactoryTalk View Site Edition

FactoryTalk View Site Edition (FT View SE) to przeznaczone dla przedsiębiorstw oprogramowanie nadzorujące typu SCADA oferujące skalowalność oraz architekturę elastycznie dopasowaną do potrzeb, która funkcjonuje zarówno w klasycznej konfiguracji systemów jednostanowiskowych, jak i w postaci systemów rozproszonych [14].

3.2.1. Narzędzia projektowe

FactoryTalk View Site Edition jest komponentem pakietu oprogramowania FactoryTalk View (dawniej RSVIEW Enterprise), w skład którego wchodzi jeszcze FactoryTalk View Machine Edition (oprogramowanie przeznaczone głównie dla paneli operatorskich i niewielkich procesów – jednej maszyny) oraz FactoryTalk View Studio (wspólne środowisko projektowe do tworzenia aplikacji). Pakiet ten (Rys. 17.) zaprojektowany został z myślą o wspólnym wyglądzie, stylu i nawigacji, aby przyspieszyć rozwój aplikacji SCADA/HMI i czas szkolenia [14].

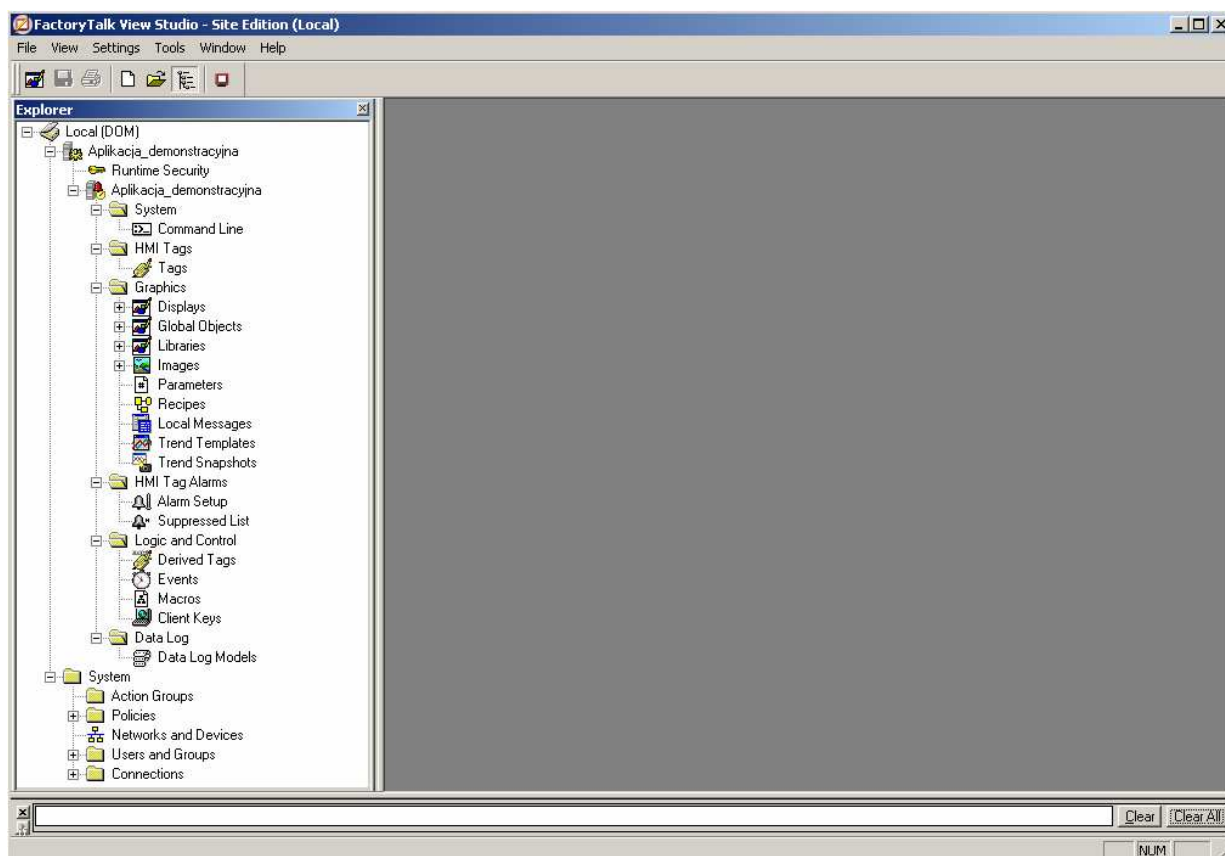


Rys. 17. FactoryTalk View [14]

FactoryTalk View Studio (Rys. 18.) to oprogramowanie konfiguracyjne, zawierające edytory do tworzenia kompletnych aplikacji oraz oprogramowanie klienta i serwera do ich testowania i uruchamiania. FT View SE Client służy do przeglądania aplikacji lokalnych i sieciowych w trybie wykonywania. Natomiast FT View SE Server przechowuje elementy HMI (np. obiekty graficzne) i udostępnia je klientowi. Serwer zawiera także listę zmiennych oraz obsługuje wykrywanie alarmów i logowanie historyczne. FT View SE Server nie ma swojego interfejsu użytkownika. Raz zainstalowany, działa jako usługa Windows i przekazuje informacje do klientów, jak się o nie zwrócić [11].

Kiedy zaczynamy projektować nową aplikację, po uruchomieniu programu FT View Studio, otwarte zostaje okno dialogowe do wyboru typu projektowanej aplikacji [11]:

- Site Edition (Network) – aplikacja rozproszona,
- Site Edition (Local) – aplikacja lokalna,
- Machine Edition – aplikacja dla paneli operatorskich.



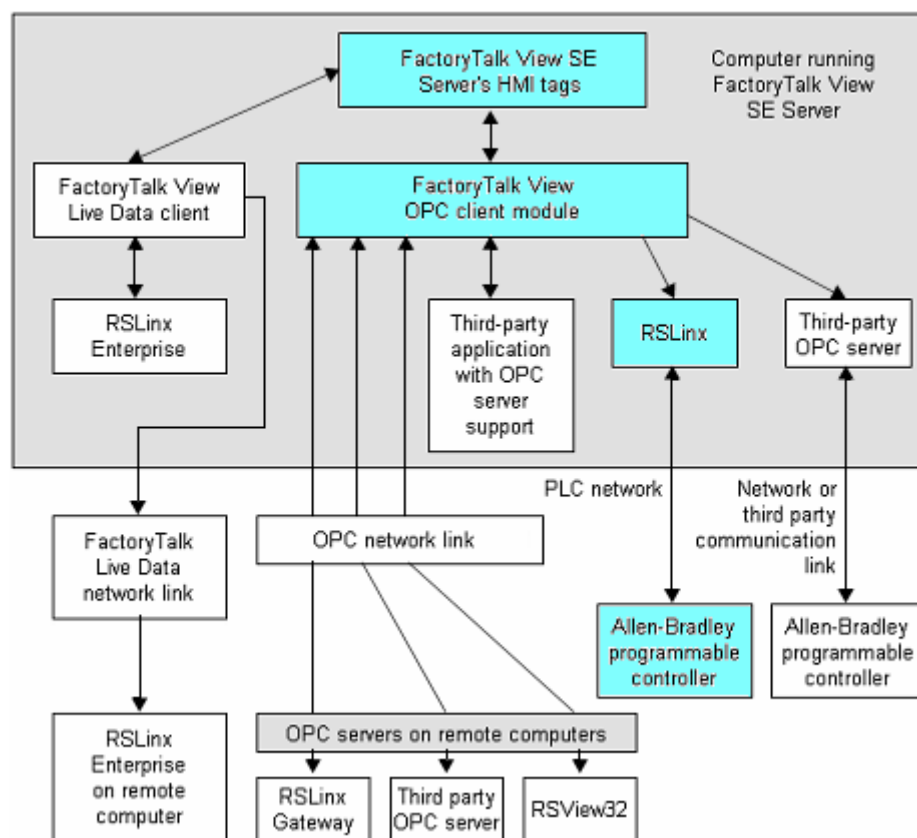
Rys. 18. FT View Studio

3.2.2. System komunikacji

Serwer danych stanowi drogę do fizycznego urządzenia w sieci, umożliwiając monitorowanie i kontrolę urządzeń za pomocą aplikacji. Serwery danych mogą łączyć klientów aplikacji ze zmiennymi sterowników programowalnych, z tagami OPC (ich wartościami lub informacjami o stanie) lub zmiennymi w programie RSLogix5000. FT View SE obsługuje następujące typy serwerów danych [11]:

- Rockwell Automation Device Servers (RSLinx Enterprise) – głównie dla aplikacji FT View,
- OPC data servers (RSLinx Classic) – wsparcie dowolnego serwera danych, który jest zgodny z OPC.

Aby skonfigurować połączenie między aplikacją a sterownikiem programowalnym, należy zainstalować jeden z programów: RSLinx Enterprise albo RSLinx Classic.



Rys. 19. Schemat komunikacji [11]

Usługa FactoryTalk Live Data zarządza połączeniami między serwerami danych i klientami, takimi jak FT View SE (Rys. 19.). FactoryTalk Live Data czyta i zapisuje wartości na serwerach danych i przedstawia dane dla klientów [11].

RSLinx dostarcza komplet programów do obsługi wszystkich sieci firmy Allen-Bradley (firma wchodząca w skład Rockwell Automation produkująca urządzenia sterujące). Za pomocą RSLinx użytkownik może mieć podgląd, w jednym oknie, wszystkich działających w systemie sterowania sieci. Może również jednocześnie uruchomić dowolną kombinację aplikacji, wykorzystując różne interfejsy komunikacyjne. RSLinx spełnia wymagania specyfikacji OPC Data Access. Ponadto obsługuje kilka różnych formatów DDE dla takich klientów, jak np. Microsoft Office. RSLinx pomaga użytkownikom w obsłudze systemu i usuwaniu usterek w sieciach, dostarczając zestawu narzędzi diagnostycznych dla sieci stacji lokalnych i połączeń OPC/DDE [14].

3.2.3. FactoryTalk Services Platform

FactoryTalk Services Platform oferuje zestaw podstawowych usług, między innymi komunikaty diagnostyczne, usługi monitorowania stanu i dostępu do danych w czasie rzeczywistym, w odniesieniu do produktów i aplikacji w systemie FactoryTalk. Korzystając z usług, produkty FactoryTalk mogą dzielić się i uzyskiwać jednoczesny dostęp do zasobów, takich jak zmienne i obiekty graficzne, także wystarczy raz zdefiniować je w systemie. FactoryTalk Services Platform musi być zainstalowana wszędzie tam, gdzie produkty, takie jak FT View SE mają działać. Do podstawowych usług platformy należą [11]:

- FactoryTalk Directory centralizuje dostęp do zasobów i składników. Działa jak elektroniczna książka adresowa, zapewniając wyszukiwanie dla części aplikacji tak, aby mogły one znaleźć siebie na jednym komputerze lub przez sieć. Komponenty aplikacji mogą być przechowywane w swoim oryginalnym środowisku i mogą być dostępne dla wszystkich klientów, bez konieczności ich powielania. Aplikacje FT View SE mogą korzystać z dwóch rodzajów FactoryTalk Directory. Local Directory – zarządza lokalnymi aplikacjami. Wszystkie składniki aplikacji, z wyjątkiem danych serwerów OPC, muszą znajdować się na tym samym komputerze. Network Directory – zarządza aplikacjami sieciowymi (rozproszonymi). Aplikacje sieciowe mogą składać się z wielu klientów i serwerów, podzielonych na wiele komputerów połączonych w sieci.
- FactoryTalk Security oferuje szeroką gamę usług ochrony, które są zintegrowane z FactoryTalk Directory. Daje nam istotną warstwę zabezpieczeń automatyki, przez ograniczenie dostępu i korzystania z oprogramowania. FactoryTalk Security zapewnia scentralizowane uwierzytelnianie i kontrolę dostępu, sprawdzając tożsamość każdego

użytkownika, który próbuje uzyskać dostęp do systemu, a następnie udzielanie lub odmawianie dostępu użytkownikowi żądającemu wykonania poszczególnych działań w systemie.

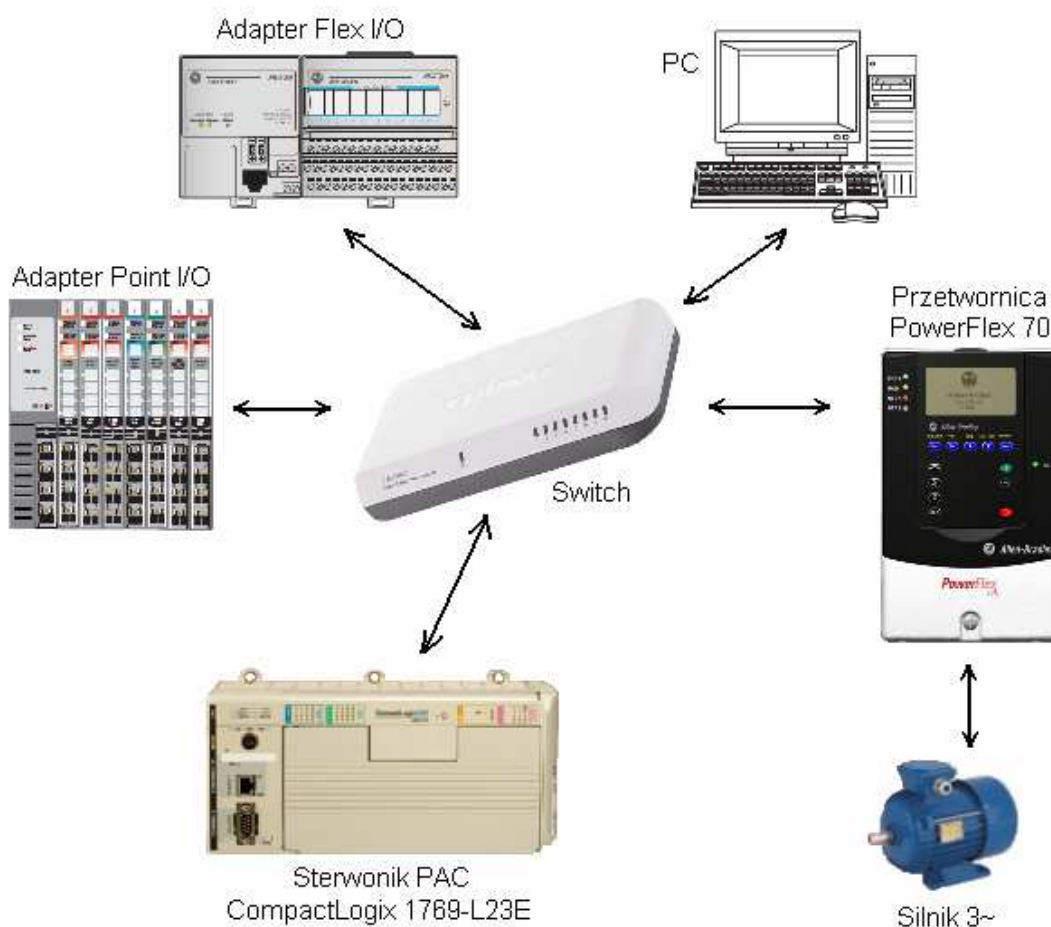
- FactoryTalk Live Data umożliwia komunikację w czasie rzeczywistym między sprzętem i oprogramowaniem, a także osobą trzecią na połączenie z serwerem danych OPC.
- FactoryTalk Diagnostics gromadzi, przechowuje i zapewnia dostęp do aktywności, statusu, ostrzeżeń i komunikatów o błędach generowanych przez produkty we wszystkich fazach projektu, włączając instalację, konfigurację i obsługę. Informacje te mogą być oglądane na liście Diagnostics List lub w przeglądarce Diagnostics Viewer, archiwizowane dla przyszłego przetwarzania lub analizy, eksportowane do formatu ODBC w trybie online. Eksport do formatu ODBC pozwala analizować dane za pomocą innych narzędzi, takich jak, np. Microsoft Excel.

FactoryTalk Services Platform zapewnia jakość poprzez ponowne wykorzystanie i dzielenie wspólnych usług w szerokim zakresie oprogramowania. Podział funkcji pozwala na efektywną pracę pomiędzy aplikacjami i powoduje zmniejszenie technicznych, operacyjnych i szkoleniowych kosztów. Innymi słowy, produkty FactoryTalk, które zostały zbudowane na FactoryTalk Services Platform są lepiej zintegrowane, łatwiejsze w konfiguracji, mają identyczny wygląd i są łatwiejsze w utrzymaniu niż produkty, które nie mają wspólnych cech [14].

4. Stanowisko dydaktyczne ze sterownikiem CompactLogix i oprogramowaniem Wonderware InTouch

4.1. Budowa stanowiska

Stanowisko dydaktyczne (Rys. 20.) składa się z przetwornicy PowerFlex 70 firmy Allen-Bradley, silnika indukcyjnego, komputera PC oraz sterownika PAC CompactLogix 1769-L23E (także firmy Allen-Bradley). Przetwornica zasilana z sieci trójfazowej służy do sterowania silnikiem (schemat elektryczny – patrz Załącznik 1.). Komputer PC połączony jest ze sterownikiem PAC przez magistralę EtherNet/IP. Do wejść/wyjść sterownika podłączone są przełączniki i diody sygnalizujące. Dodatkowo sterownik PAC został rozbudowany o dwa moduły wejść/wyjść Flex I/O oraz Point I/O. Całość jest połączona ze sobą po magistrali EtherNet/IP poprzez switch. Do konfiguracji sieci służy program RSLinx Classic.



Rys. 20. Budowa stanowiska dydaktycznego

4.2. Oprogramowanie sterownika CompactLogix

Narzędziem przeznaczonym do programowania sterowników CompactLogix jest oprogramowanie RSLogix 5000. RSLogix 5000 pozwala na tworzenie zarówno programów sekwencyjnych, sterowania procesami, napędami, jak i zaawansowanych programów sterowania ruchem. Oferuje łatwy w użyciu, zgodny z normą IEC-61131-3 interfejs użytkownika, programowanie symboliczne, struktury i tablice oraz instrukcje dedykowane do wielu różnych typów aplikacji. Środowisko RSLogix 5000 przeznaczone jest do programowania całej platformy Allen-Bradley Logix [14].

Program demonstracyjny, który był wcześniej stworzony na potrzeby tego stanowiska (symulacja linii produkcyjnej) został zmodyfikowany pod aplikację SCADA (patrz Załącznik 2.). Program został napisany za pomocą schematu drabinkowego i umożliwia sterowanie linią produkcyjną napojów, która składa się z następujących etapów: pojawienie się butelki, nalewanie napoju, kapslowanie, pakowanie i powrót na początek linii. Symulowana linia produkcyjna posiada czujniki, które informują użytkownika o aktualnie wykonywanym procesie, poprzez zapalenie odpowiednich wyjść sterownika. Załączenie zasilania, rozpoczęcie oraz zakończenie procesu produkcyjnego umożliwiające jest przez wykorzystanie przełączników, które podają odpowiedni sygnał na wejście sterownika.

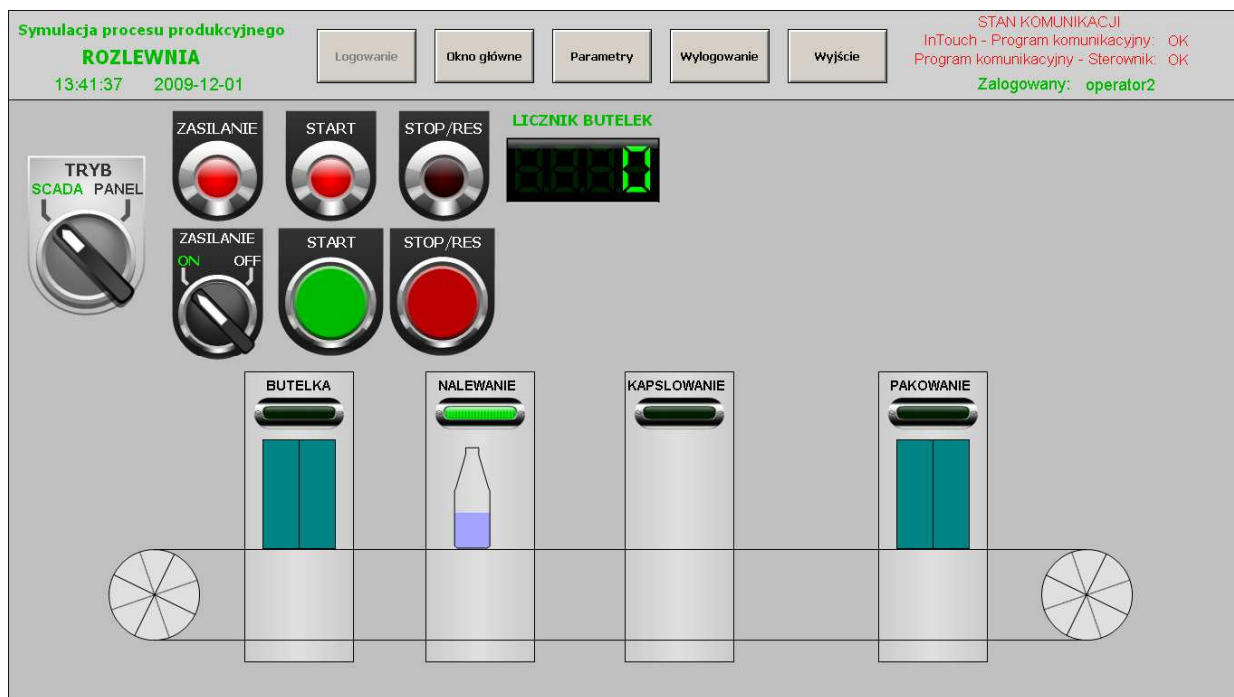
W programie demonstracyjnym zostały wykorzystane następujące przełączniki oraz wyjścia sterownika CompactLogix:

- przełącznik 1 – załączenie zasilania,
- przełącznik 2 – rozpoczęcie (start) procesu produkcyjnego,
- przełącznik 3 – zakończenie (stop) procesu produkcyjnego,
- przełącznik 5 – symulacja awarii silnika,
- wyjście 0 – sygnalizacja załączenia zasilania,
- wyjście 1 – sygnalizacja rozpoczęcia procesu produkcyjnego,
- wyjście 2 – sygnalizacja zakończenia procesu produkcyjnego,
- wyjście 5 – sygnalizacja I etapu produkcyjnego (pojawienie się butelki),
- wyjście 6 – sygnalizacja II etapu produkcyjnego (nalewanie napoju),
- wyjście 7 – sygnalizacja III etapu produkcyjnego (kapslowanie),
- wyjście 8 – sygnalizacja IV etapu produkcyjnego (pakowanie),
- wyjście 9 – sygnalizacja V etapu produkcyjnego (powrót).

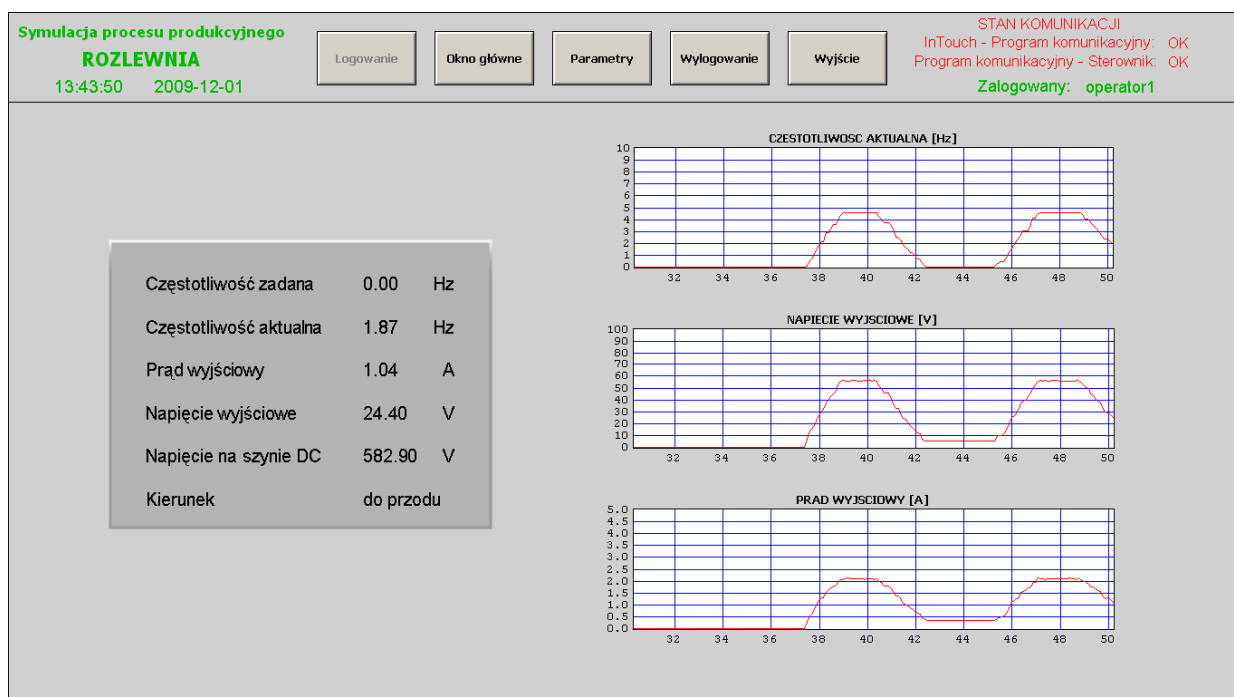
Po wgraniu programu należy pamiętać, aby skasować błędy pojawiające się w przetwornicy PowerFlex 70 przy wykorzystaniu przełącznika oznaczonego literą S. Przetwornica rozpocznie pracę, jeżeli wcześniej zostaną użyte następujące przełączniki: E – Enable, S – Stop, które spowodują zainicjowanie przetwornicy. Linia produkcyjna, po załączeniu zasilania (przełącznik 1) oraz uruchomieniu silnika (przełącznik 2), wykonuje automatycznie kolejne czynności produkcyjne, dopóki nie zostanie użyty przycisk stopu (przełącznik 3). Wykonywanie poszczególnych etapów powoduje zapalenie kolejnych wyjść sterownika (wyjście 5 – 9). Podczas ostatniego z etapów, silnik zmienia kierunek obrotów, powracając na początek procesu produkcyjnego. Proces produkcji napojów rozpoczyna się od początku. W programie zostały wykorzystane bloki funkcyjne (Timery), które odliczają czas wykonania poszczególnych czynności produkcyjnych. Użyto także jeden licznik (Counter) do zliczania butelek napojów podczas pakowania [16].

4.3. Aplikacja InTouch

Aplikacja demonstracyjna napisana w programie InTouch oparta jest na wcześniej opisanym programie drabinkowym, który należy wgrać do sterownika PAC przed uruchomieniem aplikacji. Aplikacja SCADA służy do wizualizacji i sterowania procesem produkcyjnym (linia produkcyjna napojów) za pomocą komputera PC. Po uruchomieniu aplikacji na monitorze pojawia się okno „menu”, w którym mamy do dyspozycji przyciski: Logowanie, Okno główne, Parametry, Wylogowanie, Wyjście. Podczas logowania mamy do wyboru dwóch użytkowników z różnymi uprawnieniami. Użytkownik o loginie Operator1 z hasłem 0001 ma dostęp tylko do wizualizacji procesu bez możliwości sterowania przez komputer. Sterowanie może odbywać się tylko za pomocą panelu sterowniczego z przełącznikami mechanicznymi. Natomiast Operator2 z hasłem 0002 ma możliwość sterowania procesem za pomocą komputera albo panelu sterowniczego, w zależności od wyboru trybu pracy w aplikacji (Rys. 21.). Dodatkowo każdy z użytkowników może podejrzeć parametry silnika napędzającego taśmociąg oraz trendy bieżące (wykresy w czasie) dla prędkości, napięcia i prądu silnika (Rys. 22.).



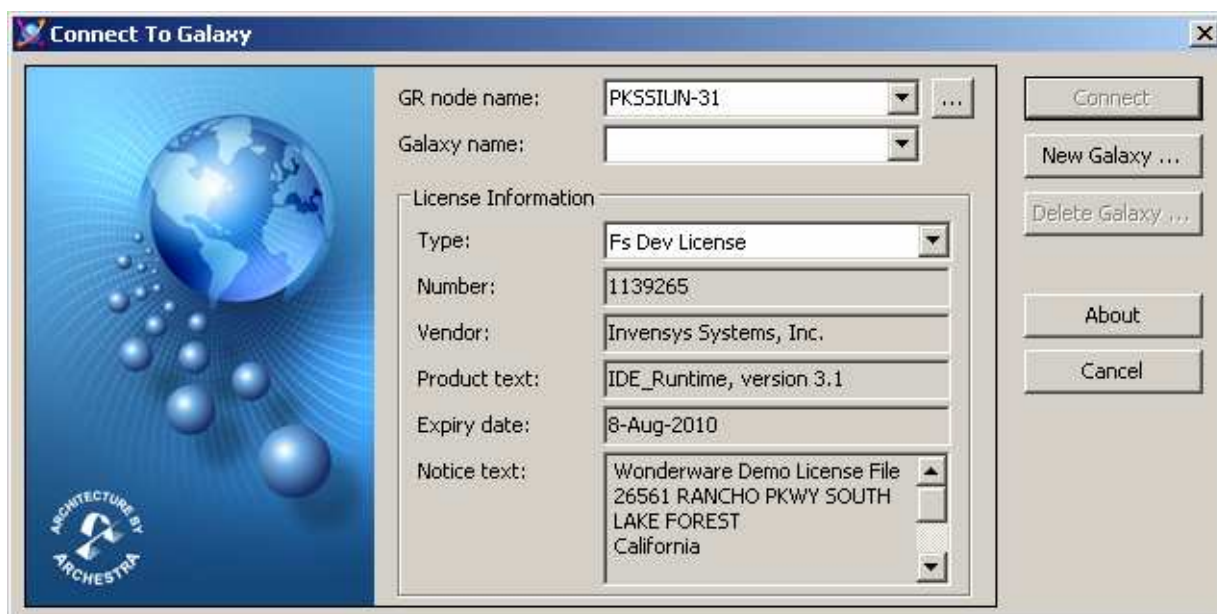
Rys. 21. Aplikacja demonstracyjna – Okno główne



Rys. 22. Aplikacja demonstracyjna - Parametry

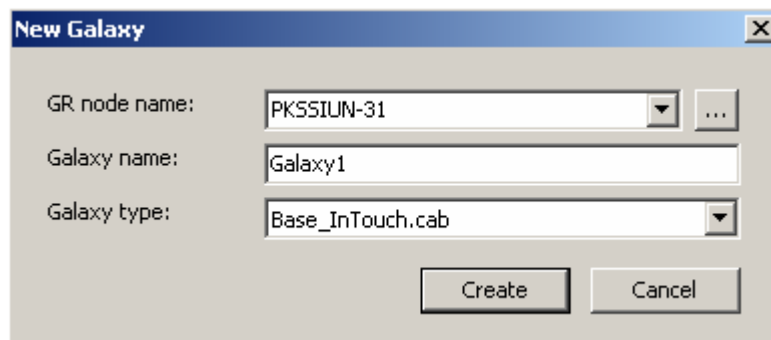
4.3.1. Zakładanie nowego projektu aplikacji

Aby utworzyć nowy projekt należy z menu Start uruchomić program Archestra IDE. Pojawi się okno Connect To Galaxy (Rys. 23.), w którym wskazujemy nazwę komputera, przy którym pracujemy i klikamy przycisk New Galaxy.



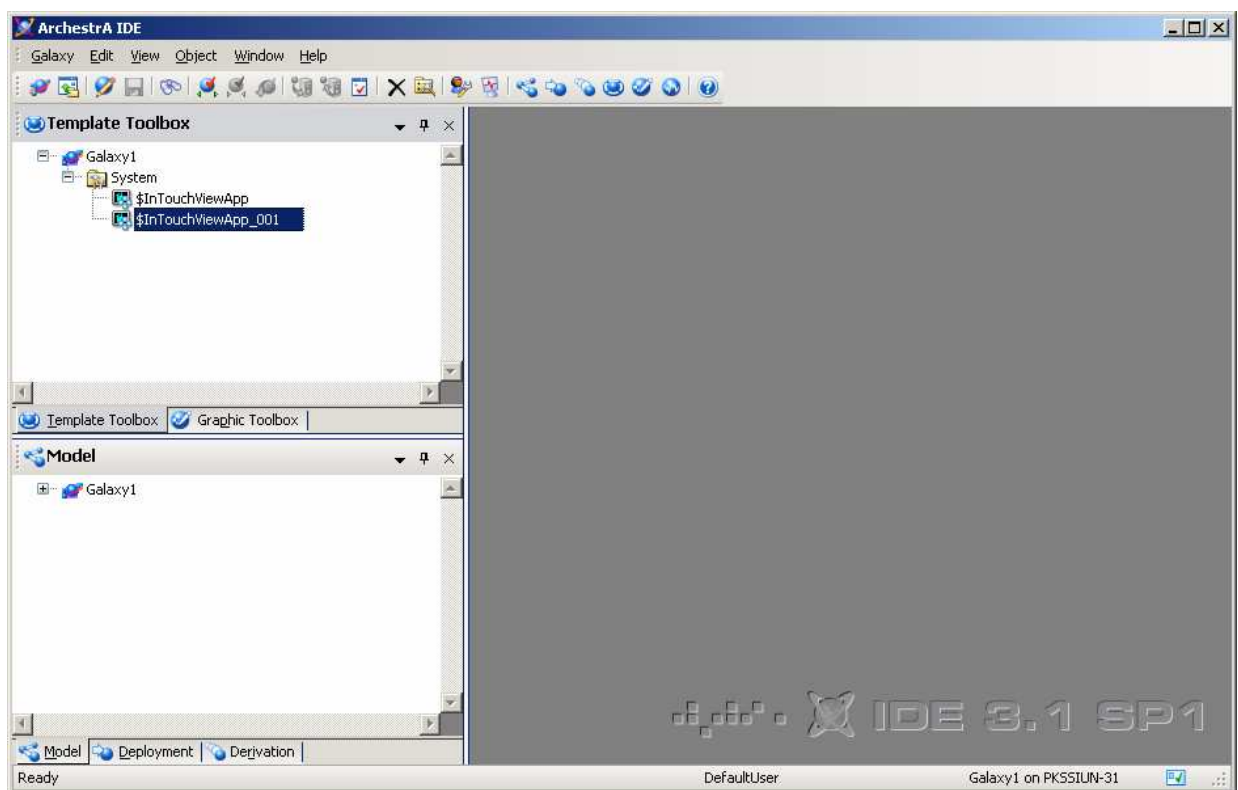
Rys. 23. Connect To Galaxy

W oknie New Galaxy (Rys. 24.) w polu Galaxy name należy wpisać nazwę nowego projektu, który zostanie utworzony, np. Galaxy1. W polu Galaxy type wybieramy Base_InTouch.cab, aby utworzyć projekt dla aplikacji wizualizacyjnej InTouch. Następnie klikamy Create. Pojawi się zestaw komunikatów informujących o postępie w tworzeniu aplikacji. W oknie Connect To Galaxy pokaże się nasza nazwa aplikacji i klikamy przycisk Connect, aby zalogować się do utworzonego projektu.

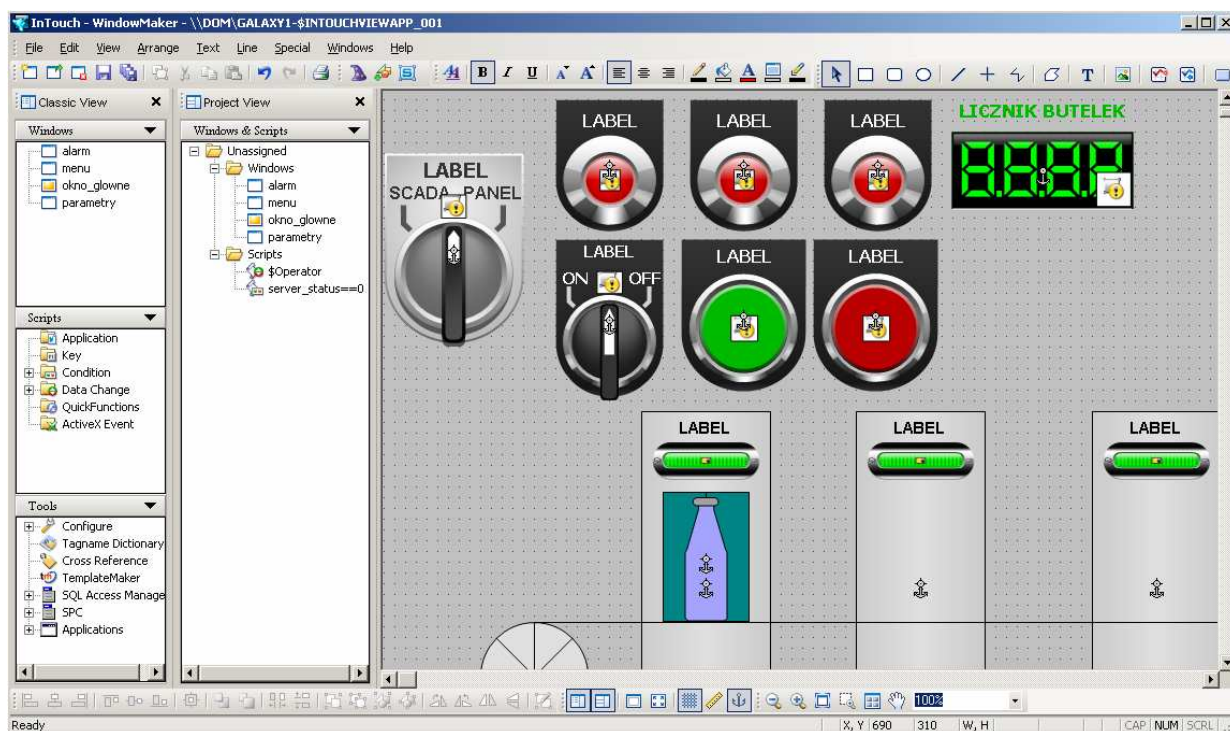


Rys. 24. New Galaxy

Po zalogowaniu się do utworzonego projektu aplikacji w środowisku projektowym ArchestrA IDE (Rys. 25.), najważniejsze w procesie projektowania aplikacji InTouch będą dwie zakładki: Template Toolbox (biblioteka aplikacji InTouch) i Graphic Toolbox (biblioteka szablonów graficznych ArchestrA). W zakładce Template Toolbox rozwijamy drzewo obiektów i klikamy prawym przyciskiem myszki na \$InTouchViewApp i wybieramy New\Derived Template. Zostanie utworzony nowy obiekt, który reprezentuje w środowisku ArchestrA IDE aplikację wizualizacyjną InTouch'a. Można nadać mu dowolną nazwę. Po utworzeniu nowego obiektu klikamy na niego dwukrotnie i w otwartym oknie zaznaczamy opcję Create New InTouch application i przyciskiem Next przechodzimy do następnego okna. W kolejnym oknie wpisujemy nazwę aplikacji InTouch i jej opis. Naciskamy przycisk Next i zostaje uruchomiony program WindowMaker (Rys. 26.), w którym będziemy projektować aplikację wizualizacyjną.



Rys. 25. ArchestrA IDE



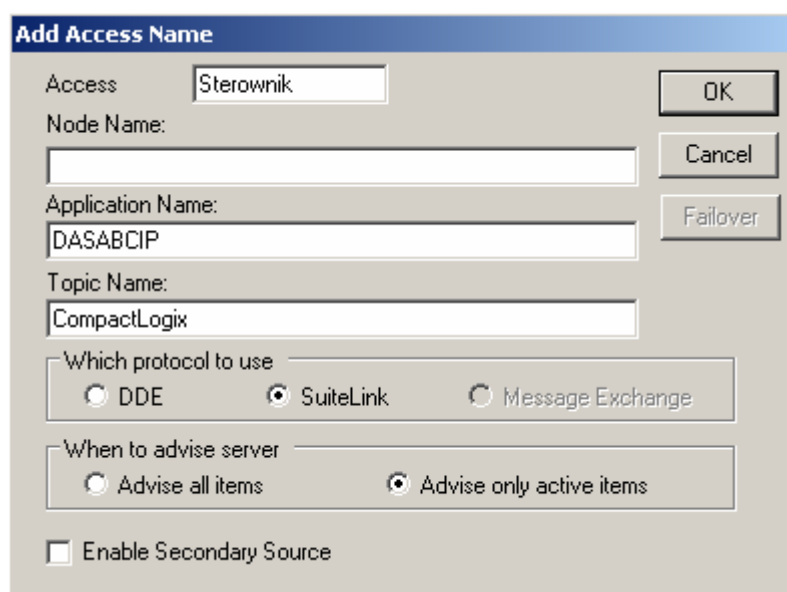
Rys. 26. WindowMaker

4.3.2. Konfiguracja komunikacji ze sterownikiem

W celu skonfigurowania komunikacji ze sterownikiem programowalnym należy zainstalować odpowiedni program komunikacyjny. Wonderware dostarcza nową generację programów komunikacyjnych o nazwie DAServers, które zaprojektowano przy użyciu technologii Archestra. Odpowiednim programem komunikacyjnym dla naszego stanowiska będzie DASABCIP, który umożliwia komunikację ze sterownikami Allen-Bradley przy użyciu protokołu CIP przez EtherNet/IP. Następnie z menu Start uruchamiamy środowisko konfiguracyjne System Management Console (SMC) służące do zarządzania programami komunikacyjnymi. W drzewie po lewej stronie (Rys. 27.) rozwijamy gałąź DAServer Manager i na poziomie Configuration klikamy prawym klawiszem myszki i wybieramy Add PORT_CIP Object. Następnie znowu klikamy i dodajemy ENB_CPLX obiekt, który jest logiczną reprezentacją zintegrowanego portu EtherNet/IP dla rodziny sterowników CompactLogix. W tym oknie w polu Host Name należy wpisać adres IP sterownika, w tym przypadku 192.168.1.103. Dodajemy jeszcze do gałęzi BACKPLANE_CPLX i LOGIX_CPLX.

Dalej zmieniamy tryb uruchamiania programu komunikacyjnego na automatyczny. W tym celu w drzewie klikamy prawym przyciskiem myszy na ArchestrA.DASABCIP.3 i z menu wybieramy Configure As Service\Auto Service. Teraz program komunikacyjny będzie uruchamiany automatycznie przy stracie systemu operacyjnego. Aby uruchomić program komunikacyjny klikamy prawym przyciskiem myszy na ArchestrA.DASABCIP.3 i wybieramy Activate Server. Możemy zamknąć konsolę SMC.

Następnym krokiem jest konfiguracja komunikacji aplikacji InTouch z programem komunikacyjnym. W programie WindowMaker z menu Special wybieramy Access Names. W otwartym oknie klikamy Add, aby skonfigurować nowe połączenie z programem komunikacyjnym. W oknie Add Access Name (Rys. 29.) w polu Access wpisujemy dowolną nazwę, np. Sterownik. Pole Node Name służy do wpisania nazwy komputera, na którym znajduje się program komunikacyjny. Jeśli program komunikacyjny znajduje się na tym samym komputerze, co aplikacja InTouch, to wtedy pole to zostawiamy puste. W Application Name należy podać nazwę programu komunikacyjnego, czyli DASABCIP. W Topic Name trzeba wpisać taką samą nazwę kanału komunikacji, jaka została skonfigurowana w konsoli SMC w zakładce Device Group, czyli CompactLogix. Zalecany przez Wonderware protokołem używanym do komunikacji pomiędzy aplikacją InTouch a programem komunikacyjnym jest SuiteLink. Natomiast zaznaczenie Advise only active items spowoduje, że ze sterownika będą odpytywane tylko zmienne aktywne, co jest metodą optymalną. Klikamy przycisk OK. Po skonfigurowaniu komunikacji możemy zabrać się za projektowanie aplikacji wizualizacyjnej.



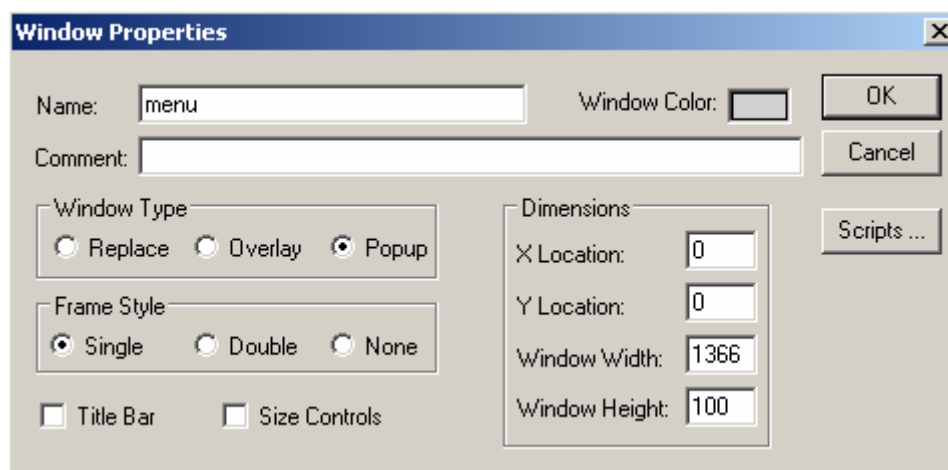
Rys. 29. Access Names

4.3.3. Konfiguracja okien synoptycznych

Konfigurowanie okien synoptycznych pokazane zostanie na przykładzie okna „menu” zdefiniowanego w aplikacji demonstracyjnej. W programie WindowMaker z menu File wybieramy New Window. W otwartym oknie Window Properties (Rys. 30.) w polu Name wpisujemy nazwę okna. Window Color służy do wyboru koloru tła okna. W grupie Dimensions podajemy położenie okna i jego wymiary w pikselach. Z kolei grupa Window Type służy do wyboru rodzaju okna:

- Replace – okno pojawiając się na ekranie zamyka inne okna, które pokrywa,
- Overlay – okno umieszczane jest na aktualnie wyświetlanych oknach. Kiedy okno tego typu zostanie zamknięte, inne okna wcześniej przykryte pojawiają się ponownie. Kliknięcie na widocznej części okna leżącego pod oknem tego typu, powoduje wyświetlenie tego okna i przeniesienie go na pierwszy plan jako okno aktywne.
- Popup – okno jest podobne do okna typu Overlay za wyjątkiem tego, że zawsze przykrywa wszystkie otwarte okna.

Oczywiście w przypadku okna „menu” zaznaczamy opcję Popup, ponieważ okno to musi być zawsze dostępne.



Rys. 30. Ustawienia nowego okna

Ostatnia grupa Frame Style daje możliwość wyboru stylu ramki okna:

- Single – 1-wymiarowe okno z krawędziami, posiadające pasek tytułowy oraz możliwość zmiany rozmiaru,
- Double – 3-wymiarowe okno z krawędziami, nieposiadające paska tytułowego, z możliwością zmiany rozmiarów,

- None – okno bez krawędzi, oraz z możliwością zmiany rozmiarów wyłącznie za pomocą przycisków zmiany rozmiaru.

Opcje Title Bar i Size Controls służą odpowiednio do włączenia paska tytułowego okna i zmiany rozmiaru okna. W tym przypadku wybieramy styl ramki Single bez paska tytułowego i kontroli rozmiaru okna.

Aby nasze okno miało przyciski, z paska Drawing (Rys. 31.) za pomocą ikony Button, osadzamy w oknie obiekty w postaci przycisków. Zaznaczamy przycisk i naciskając skrót klawiszowy Ctrl+L pojawi się okno, w którym możemy zmienić opis obiektu. W menu File klikamy opcję Save Window, aby zapisać zaprojektowane okno. Naciskamy napis Runtime znajdujący się w prawym górnym rogu programu WindowMaker. Zostanie uruchomiony program WindowViewer, w którym możemy podejrzeć efekty naszej pracy.



Rys. 31. Pasek Drawing

4.3.4. Definiowanie zmiennych

Definiowanie zmiennych może odbywać się na dwa sposoby. Pierwszy polega na dodawaniu zmiennych w programie WindowMaker odwołując się do nazw zmiennych w sterowniku. Kiedy zajdzie potrzeba zmodyfikowania zmiennej, to trzeba wprowadzić zmiany w aplikacji. Drugi sposób jest wygodniejszy. Każdy program komunikacyjny DAServer umożliwia tworzenie aliasów dla nazw zmiennych. Zamiast odwoływać się do sterownika, można zastosować dowolne nazewnictwo wygodne dla osoby konfigurującej program. Jest to bardzo przydatne, ponieważ w przypadku konieczności zmiany odczytywanych adresów z systemu sterowania nie trzeba wprowadzać zmian w zaprojektowanej aplikacji, wystarczy tylko zmienić ustawienia w samym programie komunikacyjnym. W tym przypadku dodajemy zmienne do aplikacji odwołując się do aliasów stworzonych w zakładce Device Items w konsoli SMC (Rys. 32.). W obu sposobach, aby zdefiniować zmienne w aplikacji, w programie WindowMaker wybieramy z menu Special opcję Tagname Dictionary (lista zmiennych).

Istnieją cztery rodzaje zmiennych [10]:

- Discrete – zmienna dyskretna o wartości 0 lub 1,
- Integer – 32-bitowa zmienna całkowita ze znakiem o wartościach z zakresu od -2147483648 do 2147483648,
- Real – zmienna typu rzeczywistego; może przyjmować wartość od $-3,4e^{38}$ do $3,4e^{38}$,
- Message – zmienna typu tekstowego o długości do 131 znaków.

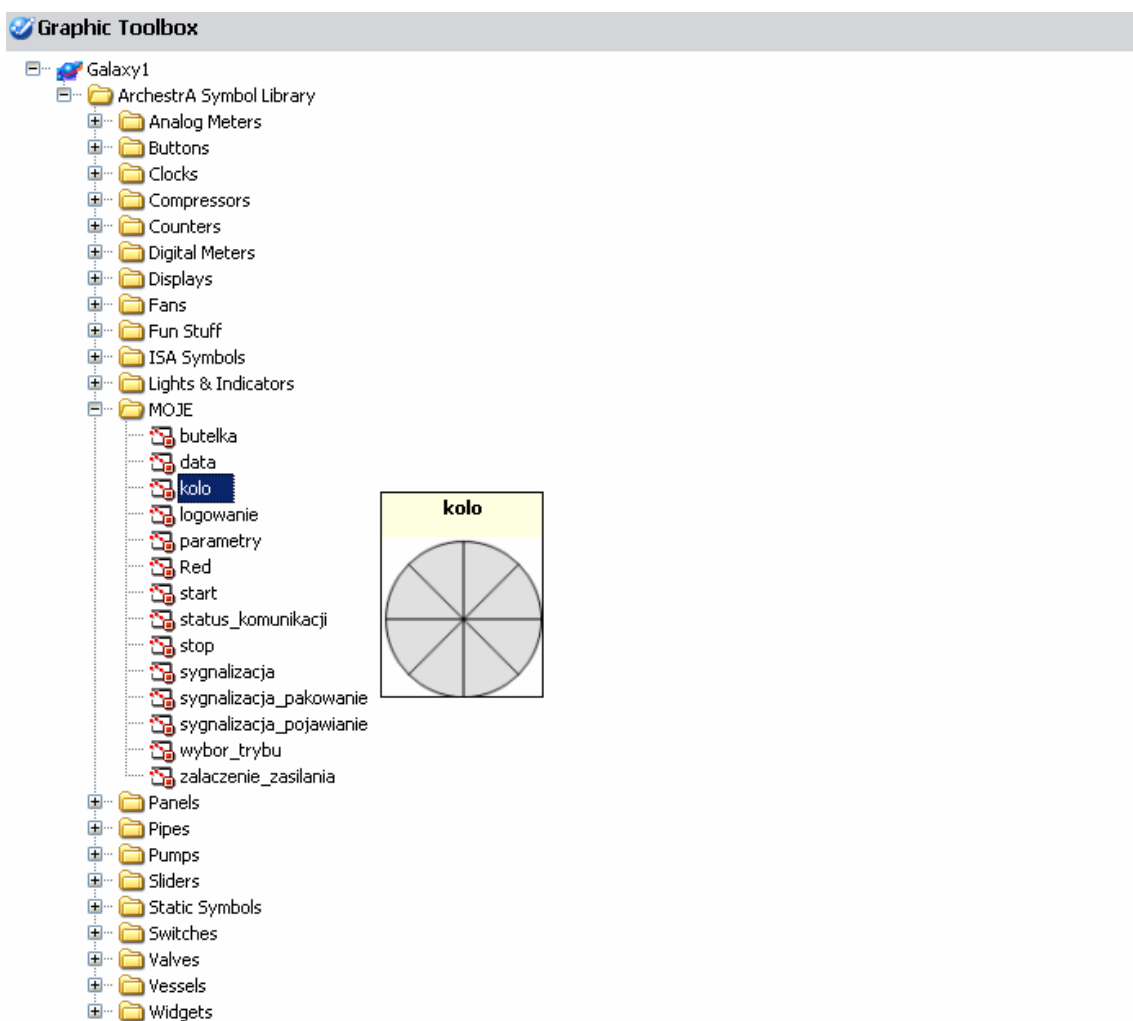
W oknie Tagname Dictionary (Rys. 33.) naciskając przycisk New można zacząć definiować nową zmienną. Na początek klikamy przycisk Type, aby wybrać typ zmiennej, np. I/O Discrete. W polu Item wpisujemy nazwę zmiennej, do której się odwołujemy (lub alias z konsoli SMC), np. awaria_silnika (jest to alias odwołujący się do zmiennej sterownika Local:2:O.Data.4). Zaznaczamy opcję Use Tagname as Item Name, aby nazwa zmiennej w programie WindowMaker była taka sama, jak nazwa zmiennej, do której się odwołujemy. Musimy jeszcze pamiętać, żeby przyporządkować zmienną do odpowiedniej nazwy dostępu. W tym celu klikamy na przycisk Access Name i wybieramy nazwę Sterownik. Po zakończeniu definicji zmiennej należy nacisnąć przycisk Save i naciskając ponownie New można przejść do definiowania kolejnej zmiennej. Z kolei przycisk Select pozwala na dostęp do przeglądarki zmiennych.

The screenshot shows the 'Tagname Dictionary' window with the 'Details' tab active. The 'New' button is highlighted. The 'Tagname' field contains 'awaria_silnika'. The 'Type' dropdown is set to 'I/O Discrete'. The 'Group' is '\$System'. The 'Comment' is 'AccessLevel'. The 'Initial Value' is 'Off'. The 'Input Conversion' is 'Direct'. The 'On Msg' and 'Off Msg' fields are empty. The 'Access Name' is 'Sterownik'. The 'Item' field contains 'awaria_silnika'. The 'Use Tagname as Item Name' checkbox is checked.

Rys. 33. Lista zmiennych

4.3.5. Projektowanie i konfiguracja symboli graficznych

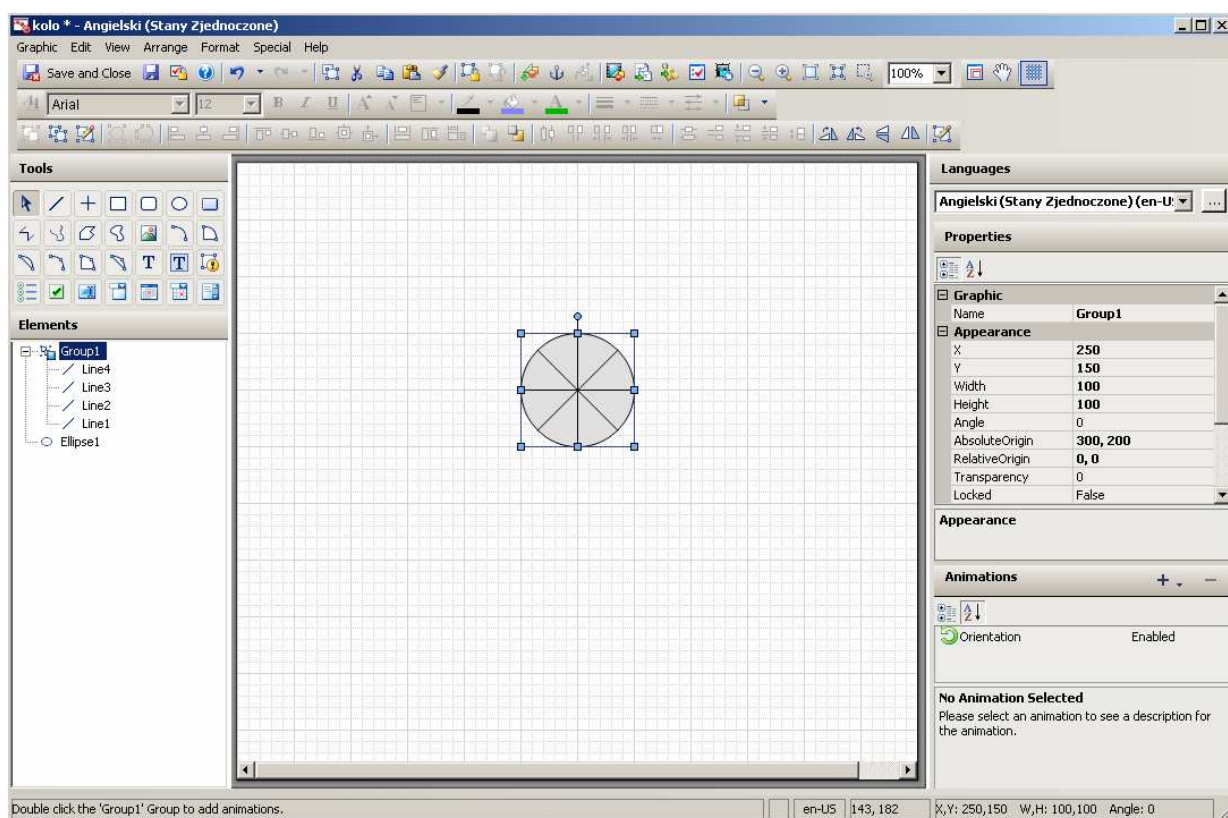
W celu utworzenia nowego symbolu graficznego w środowisku Archestra IDE należy przejść do zakładki Graphic Toolbox (Rys. 34.). Klikamy prawym przyciskiem myszy na bibliotekę Archestra Symbol Library i wybieramy opcję New\Graphic Toolset, aby stworzyć nową grupę, w której będą umieszczona nowe symbole graficzne. Możemy nadać dowolną nazwę grupie. Jeżeli chcemy modyfikować istniejące już w bibliotece symbole, zalecane jest robienie kopii standardowego symbolu, aby wprowadzane zmiany były wykonywane na kopiach. Aby stworzyć kopię standardowego symbolu, klikamy prawym przyciskiem myszy na symbol i wybieramy Duplicate. Następnie powielony symbol przenosimy do nowo utworzonej grupy. Teraz stworzymy nowy symbol w nowej grupie. Klikamy prawym klawiszem myszy na nową grupę i wybieramy New\Symbol. Nadajemy nazwę symbolowi. Po zdefiniowaniu nowego symbolu należy kliknąć na niego dwukrotnie. Pojawi się edytor graficzny symbolu.



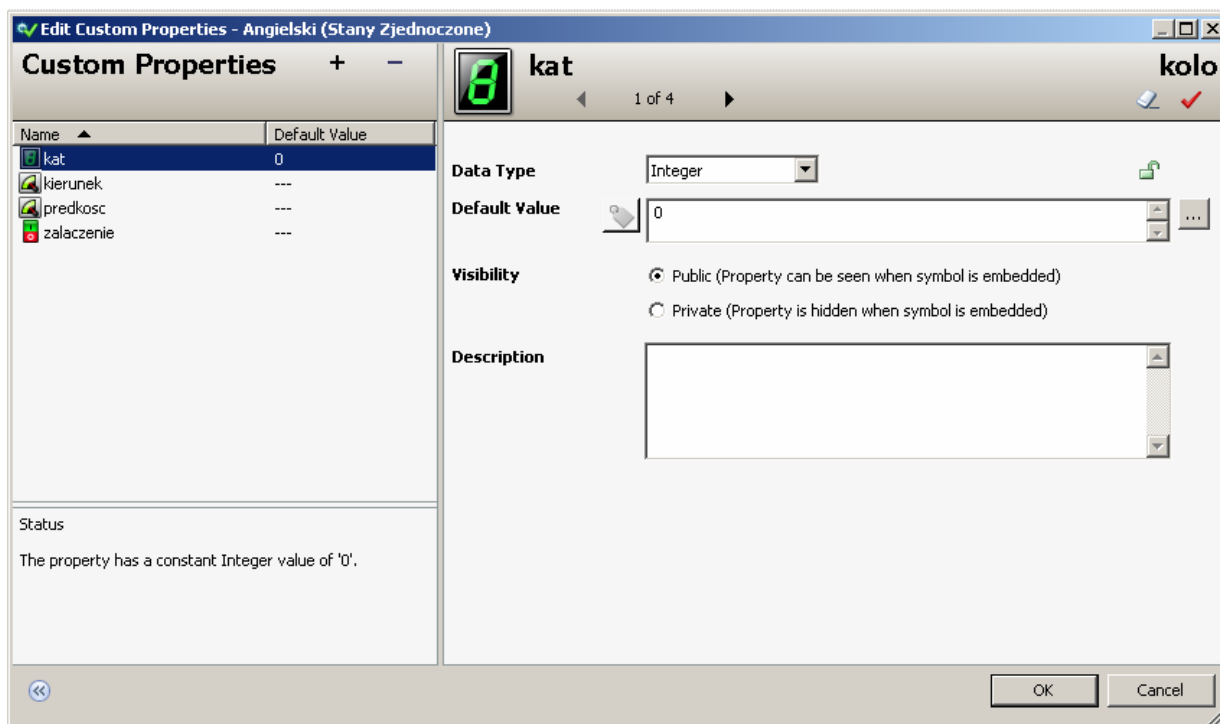
Rys. 34. Graphic Toolbox

Projektowanie symboli graficznych zostanie pokazane na przykładzie symbolu rolki taśmociągu w aplikacji demonstracyjnej. W tym celu w edytorze graficznym przy pomocy narzędzi z paska Tools rysujemy koło i cztery szprychy (Rys. 35.). Zaznaczamy wszystkie szprychy i z menu Arrange\Grouping wybieramy Group, aby zgrupować elementy. Klikamy w dowolnym miejscu poza obiektami, aby obiekty zostały odznaczone i wybieramy z menu Special opcję Custom Properties, czyli definiowanie właściwości (wewnętrznych zmiennych) symbolu graficznego ArchestraA.

W oknie Edit Custom Properties (Rys. 36.) naciskamy ikonę z plusem, aby dodać nową właściwość o nazwie „załączenie”, a po prawej stronie w polu Data Type wybieramy typ danych Boolean. Właściwość ta będzie przechowywać dane typu Boolean. W taki sam sposób dodajemy właściwości „prędkość” z typem danych Float, „kierunek” z typem danych Float i „kąt” z typem danych Integer. W polu Default Value dla każdej właściwości oprócz „kąt” wpisujemy ---. Dla właściwości „kąt” należy wpisać wartość 0. Przyciskiem OK zamykamy okno.

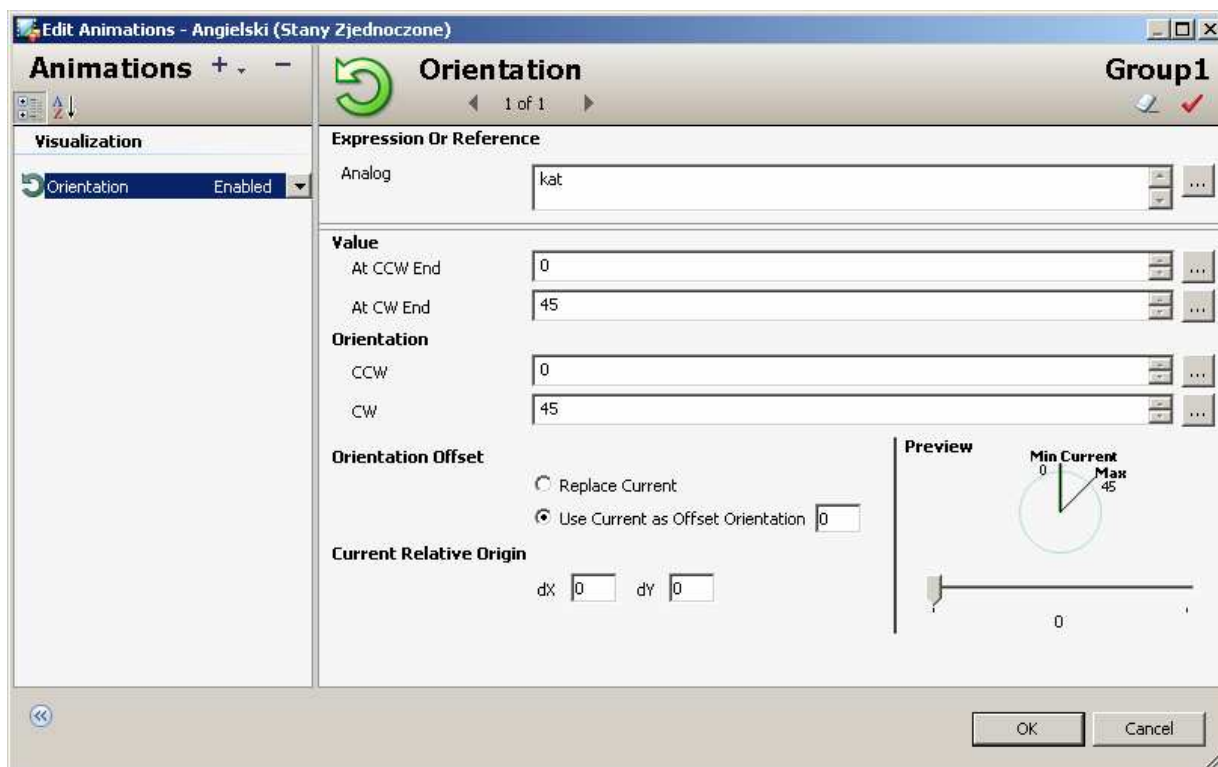


Rys. 35. Edytor graficzny



Rys. 36. Edit Custom Properties

Następnym krokiem przy projektowaniu symboli graficznych jest dodanie animacji. W tym celu klikamy prawym klawiszem myszy na grupę, w której znajdują się szprychy i wybieramy Edit Animations. Dodamy teraz animację obrotu szprych. W oknie Edit Animations (Rys. 37.) naciskamy ikonę z plusem. Zostanie pokazana lista wszystkich możliwych typów animacji do zdefiniowania. Wybieramy opcję Orientation. Po prawej stronie pola Expression or Reference należy nacisnąć ikonę z trzema kropkami. Pojawi się przeglądarka obiektów, gdzie przechodzimy do zakładki Element Browser i zaznaczamy na liście Elements nasz symbol. Wtedy po prawej stronie należy zaznaczyć właściwość „ką” i nacisnąć OK. W polach Value wpisujemy wartość zmiennej lub wartość wyrażenia, dla której obiekt ma być obrócony o maksymalny kąt, w kierunku przeciwnym do ruchu wskazówek zegara (CCW) i w kierunku zgodnym z ruchem wskazówek zegara (CW). W polu Orientation CCW należy wpisać kąt, o który obiekt ma zostać obrócony w kierunku przeciwnym do ruchu wskazówek zegara, w przypadku, gdy wartość zmiennej albo wyrażenia będzie równa wartości podanej w polu Value at CCW. Natomiast w polu Orientation CW wpisujemy kąt, o który obiekt ma zostać obrócony w kierunku zgodnym z ruchem wskazówek zegara, w przypadku, gdy wartość zmiennej albo wyrażenia będzie równa wartości podanej w polu Value at CW.



Rys. 37. Edit Animations

Przyciskiem OK zamykamy okno. W edytorze graficznym w lewym górnym rogu klikamy ikonę Save and Close, aby zapisać symbol i zamknąć edytor. Pojawi się okno Check In pozwalające na potwierdzenie wprowadzonych zmian oraz wpisanie komentarza.

Po zakończeniu projektowania nowego symbolu, należy skonfigurować go w programie WindowMaker. Tak samo postępuje się przy konfiguracji standardowych symboli. Na początek umieszczamy symbol w oknie synoptycznym. W tym celu na pasku Wizards klikamy ikonę Embed ArchestrA Graphic (Rys. 38.), wybieramy interesujący nas symbol i osadzamy w oknie.

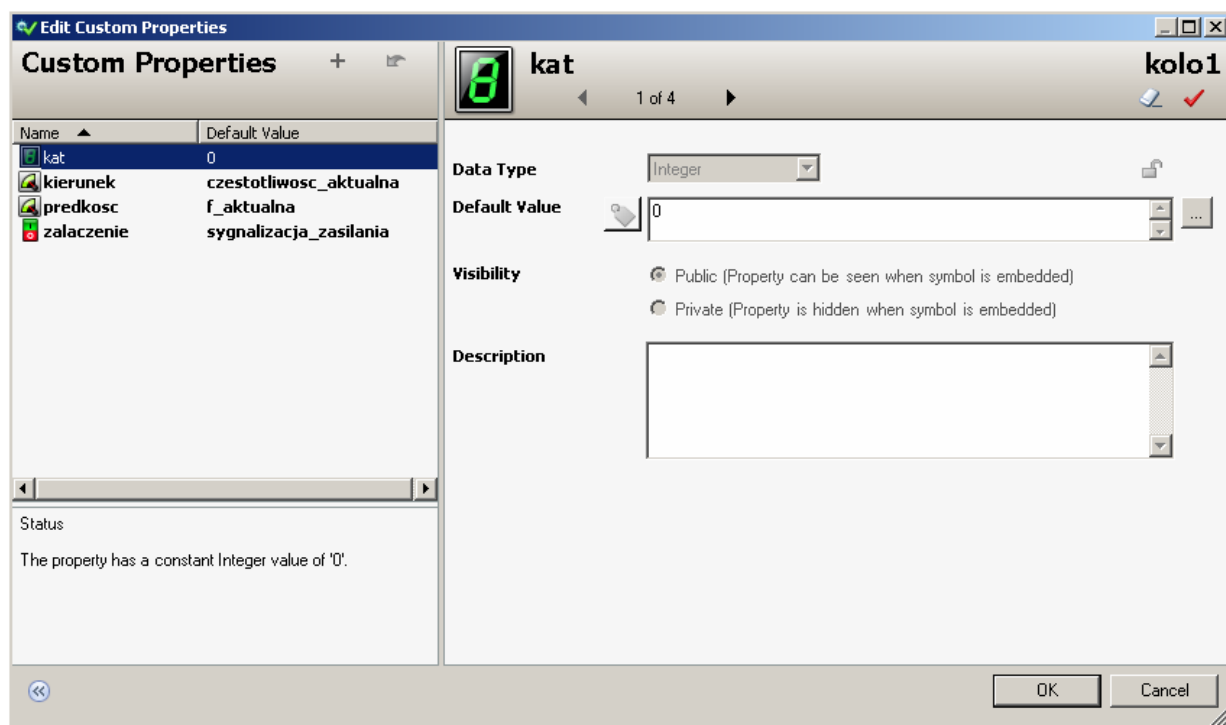


Rys. 38. Pasek Wizards

Następnie klikamy dwukrotnie na symbol i w otwartym oknie (Rys. 39.) przypisujemy do jego właściwości zmienne zdefiniowane wcześniej w programie.

Tak skonfigurowana rolka taśmociągu nie będzie spełniać swojej roli, czyli nie będzie pokazywać ruchu taśmociągu. Wartość przypisana do właściwości kąt jest równa zero, więc rolka nie będzie obracać się. W takiej sytuacji do właściwości kąt należy przypisać zmienną,

która podaje położenie rolki lub napisać skrypt, który będzie odzwierciedlać ruch rolki (korzystając z innych zmiennych, takich jak kierunek i prędkość).

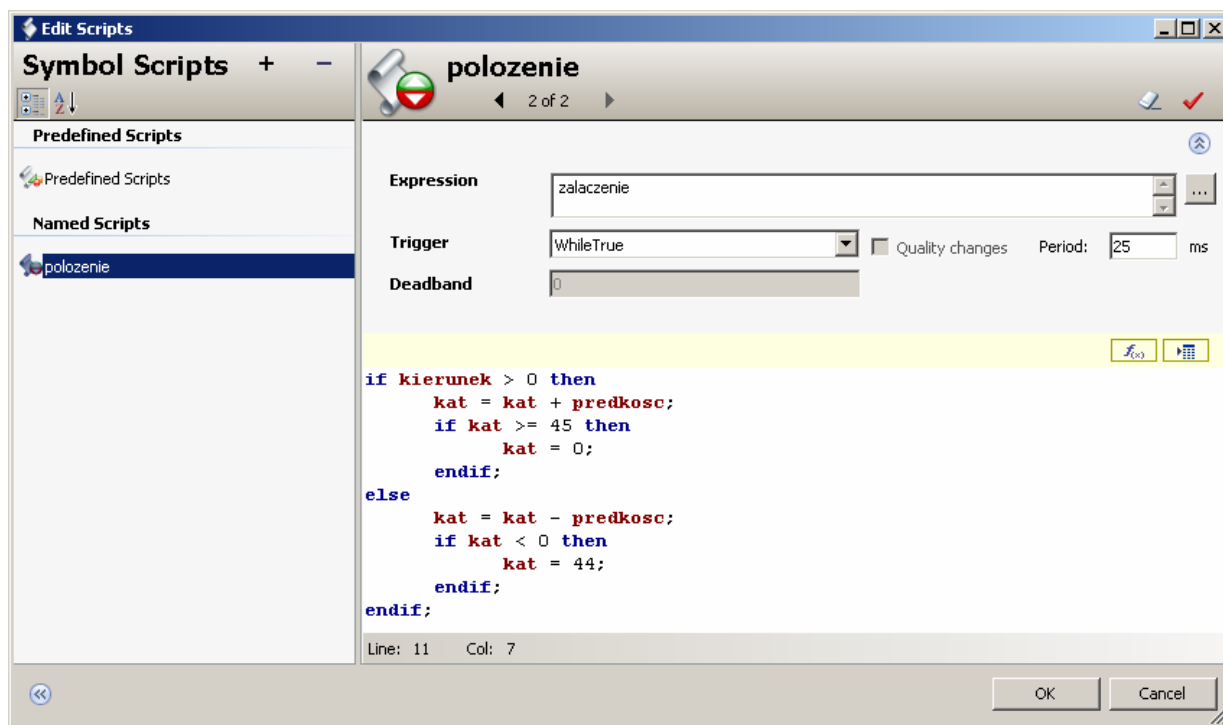


Rys. 39. Konfiguracja symbolu

4.3.6. Definiowanie skryptów

Definiowanie skryptów dla symboli graficznych zostanie pokazane na przykładzie symbolu rolki taśmociągu. Po otwarciu symbolu w edytorze graficznym z menu Special wybieramy Scripts. W otwartym oknie Edit Scripts (Rys. 40.), naciskając ikonę z plusem, możemy zdefiniować nowy skrypt. Wpisujemy nazwę skryptu, np. położenie. W polu Expression wybieramy właściwość „załączenie”, a parametr Trigger ustawiamy na WhileTrue. Oznacza to, że jeżeli zostanie załączony taśmociąg, to zostanie wykonany skrypt. W polu Period możemy wpisać jak często ma być wykonywany skrypt, gdy powyższy warunek jest spełniony. Poniżej jest miejsce, gdzie możemy napisać skrypt odpowiedzialny za wizualizację obrotu roli taśmociągu. Aby symulować ruch rolki taśmociągu, wystarczy do właściwości kąt dodawać właściwość prędkość (w zależności od kierunku). Dodawanie wystarczy wykonywać w zakresie 45° , ponieważ szprychy w symbolu rolki są ustawione pod takim kątem. W momencie, gdy kąt jest równy lub większy 45° , nastąpi szybki przeskok na 0° . Cały skrypt jest widoczny na Rys. 38. Przyciskiem OK zamykamy okno i zapisujemy

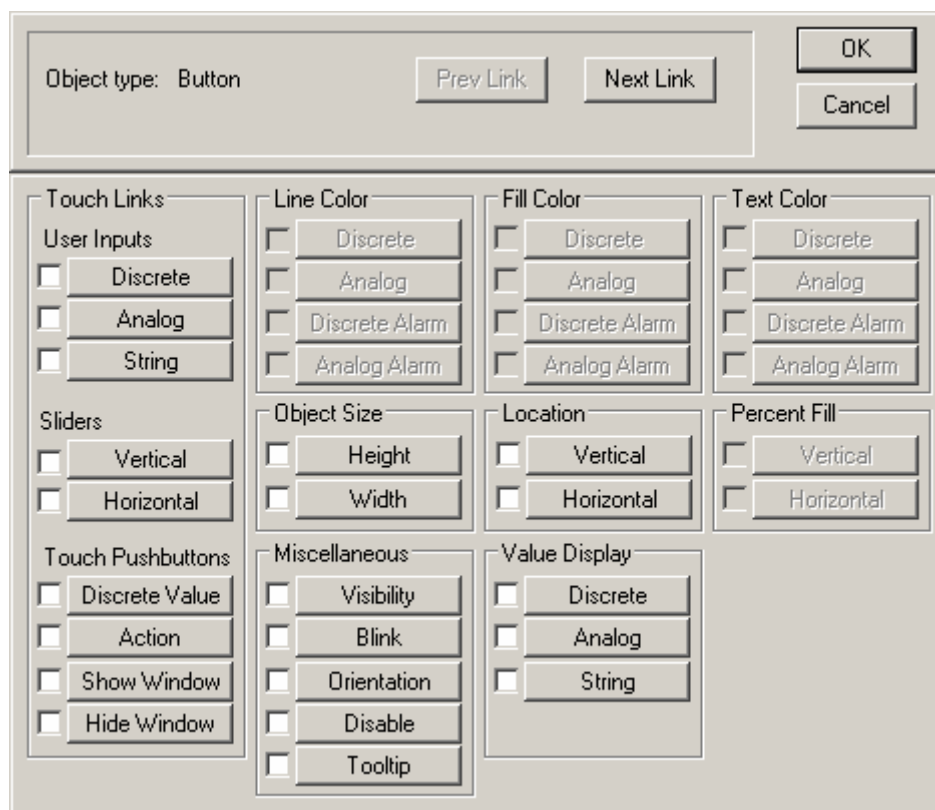
zmiany. Możemy przejść do programu WindowMaker. Jeżeli edytowany symbol był już wcześniej osadzony w oknie, naciskamy ikonę na pasku po prawej stronie na dole, aby zmiany symbolu przeniosły się do programu WindowMaker. Naciskając Runtime, możemy sprawdzić działanie naszego symbolu.



Rys. 40. Edit Scripts

Wracając do utworzonego wcześniej okna „menu”, będąc w programie WindowMaker, dodamy połączenia animacyjne do przycisków w tym oknie. Klikamy dwukrotnie myszką na dowolny przycisk. Pojawi się nowe okno dialogowe do wyboru połączeń animacyjnych (Rys. 41.). Przykładem będzie przycisk „Parametry”, który po naciśnięciu wyświetla okno synoptyczne z parametrami silnika napędzającego taśmociąg. Aby przycisk otwierał inne okno synoptyczne, w grupie Touch Pushbuttons klikamy Show Window i zaznaczamy nazwę okna, które ma być otwierane. Do tego przycisku dodamy jeszcze połączenie animacyjne wyłączenia przycisku (nieaktywny), gdy żaden z użytkowników nie jest zalogowany. W tym celu w grupie Miscellaneous klikamy Disable, wpisujemy \$AccessLevel>0 i zaznaczamy opcję Disabled State na Off. Innym przykładem jest przycisk „Wyjście”, który służy do zamykania aplikacji. Aby zdefiniować takie połączenie animacyjne, z grupy Touch Pushbuttons wybieramy Action i wpisujemy funkcję WWControl(InfoAppTitle("view"),"Close") odpowiedzialną za zamykanie aplikacji. W polu

Condition Type wybieramy On Left Click/Key Down, aby aktywować funkcję przy kliknięciu lewym przyciskiem myszki. W podobny sposób można skonfigurować pozostałe przyciski w oknie „menu”.



Rys. 41. Połączenia animacyjne

Oprócz skryptów związanych z obiektami graficznymi, InTouch daje nam możliwość definiowania skryptów (QuickScripts) związanych np. z całą aplikacją, z konkretnym oknem synoptycznym, ze zmienną, z klawiszem, itp. Aby utworzyć nowy skrypt, należy w programie WindowMaker z menu Special wybrać polecenie Scripts, a następnie typ skryptu, który chcemy utworzyć. Zostanie otwarty edytor skryptów, w którym możemy definiować skrypty. Przy użyciu QuickScripts możemy np. przeliczać zmienne na odpowiednie jednostki. Przykładem może być częstotliwość aktualna silnika, która jest podawana z przetwornicy w postaci cyfrowej. Wiedząc, że rozdzielczość wynosi 16 bitów (jeden bit przeznaczony jest na znak), możemy obliczyć poziom kwantyzacji: $2^{15} - 1 = 32767$. Jeżeli skala pomiaru jest od 0 do 50 Hz, to najmniejsza odczytana wartość częstotliwości w przybliżeniu wynosi: $50/32767 = 0,0015$ Hz. Ten wynik może być użyty w skrypcie do przeliczania wartości cyfrowej na wartość w hercach, np. częstotliwość aktualna w postaci cyfrowej wynosi 3000, to po przeliczeniu na jednostki wynosi 4,58 Hz.

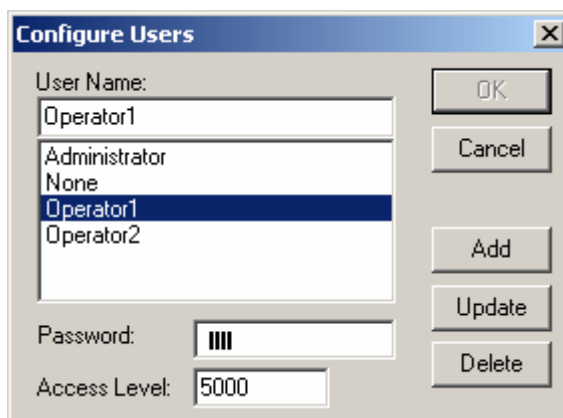
4.3.7. Konfiguracja użytkowników i publikowanie aplikacji do katalogu

Wprowadzanie zabezpieczeń do aplikacji jest opcjonalne. Jednak, korzystanie z tych mechanizmów w aplikacji, pozwala na określanie, które funkcje będą dostępne dla operatora poprzez powiązanie tych funkcji z wewnętrznymi zmiennymi. Dodatkowo, po wprowadzeniu zabezpieczeń do aplikacji, można powiązać operatora z alarmami, które wystąpiły w momencie, gdy był on zalogowany do systemu. Mechanizm zabezpieczania oparty jest na logowaniu się operatora do aplikacji przez wprowadzanie nazwy i hasła. Z tego powodu, każdy operator musi mieć przyporządkowaną nazwę, hasło, oraz poziom dostępu. Operator może zalogować się do systemu poprzez wybranie Security z menu Special, a następnie kliknięcie Log On (jeżeli menu Special jest wyświetlane), lub można utworzyć własne okna do logowania z obiektami reagującymi na naciskanie, powiązane ze zmiennymi zabezpieczającymi [10]. W przypadku aplikacji demonstracyjnej zdefiniowano przyciski do logowania w oknie „menu”.

Aby wprowadzić zabezpieczenia w naszej aplikacji, w programie WindowMaker z menu Special\Security wybieramy Select Security Type i zmieniamy typ bezpieczeństwa z None na InTouch. Teraz skonfigurujemy nowych użytkowników. Z menu Special\Security wybieramy Log On. Logujemy się na konto administrator z hasłem wonderware. Po zalogowaniu na konto administratora wybieramy z menu Special\Security opcję Configure Users. W otwartym oknie (Rys. 42.) definiujemy dwóch użytkowników:

- Operator1 z hasłem 0001 i poziomem dostępu (Access Level) 5000,
- Operator2 z hasłem 0002 i poziomem dostępu 9001.

Przyciskiem OK zamykamy okno.



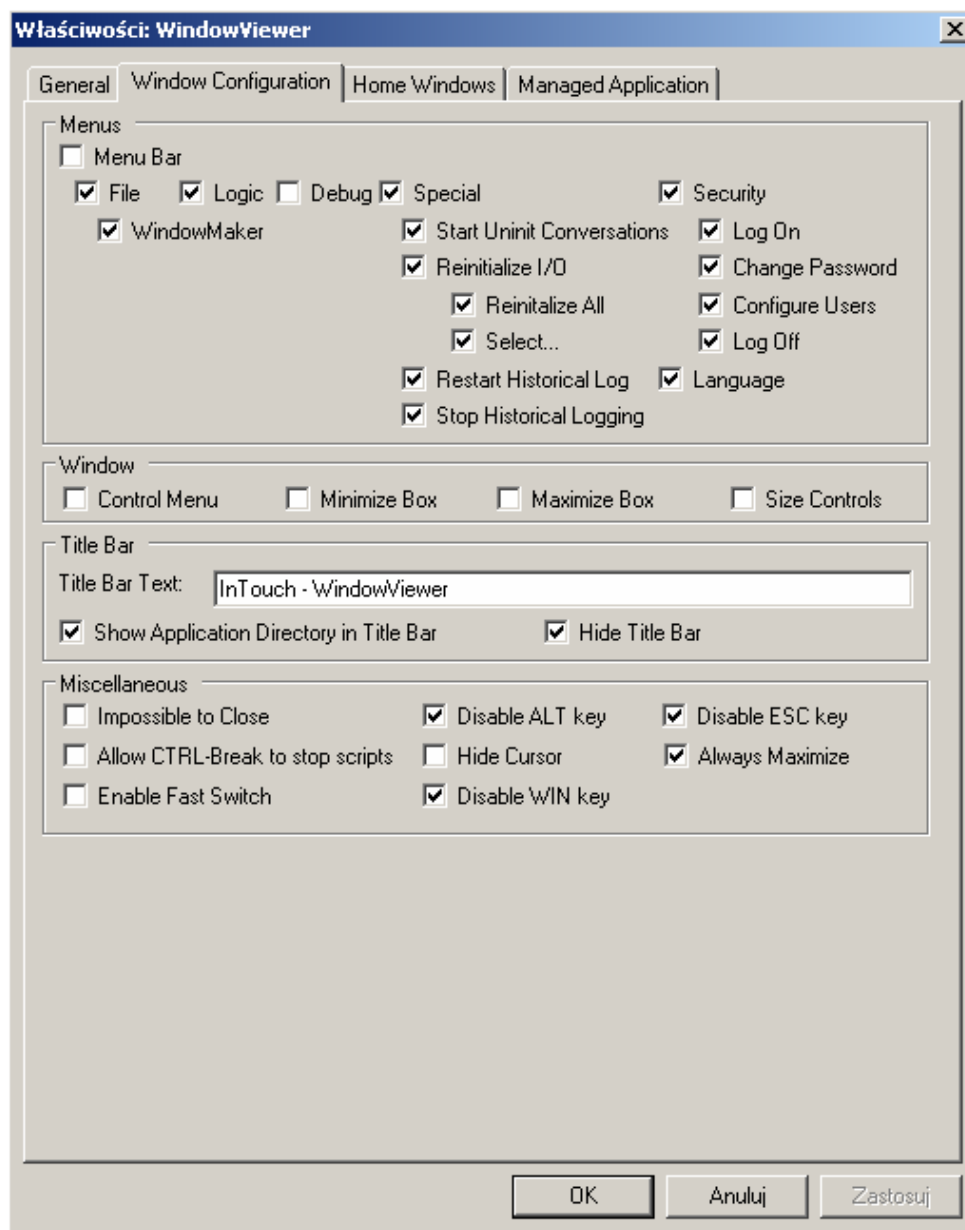
Rys. 42. Konfiguracja użytkowników

Następnym krokiem jest konfiguracja przycisków do logowania. W oknie „menu” klikamy dwukrotnie myszką na przycisk Logowanie. W oknie połączeń animacyjnych wybieramy z grupy Touch Pushbuttons opcję Action i wpisujemy skrypt: `DIM x; x = PostLogonDialog();`. Skrypt ten wyświetla okno dialogowe logowania. Natomiast dla przycisku Wylogowanie wpisujemy skrypt: `$OperatorEntered="None"; $PasswordEntered=""`; Skrypt umożliwia wylogowanie i przełączenie na użytkownika None. Musimy jeszcze pamiętać o dodaniu skryptu do aplikacji, który po wylogowaniu będzie zamykał wszystkie okna synoptyczne, oprócz okna „menu”. Wykorzystując różne poziomy dostępu dla użytkowników, można także ustawić różne role dla użytkowników, czyli np. jakie okna mają być dostępne dla danego użytkownika, itp.

W celu uruchomienia zaprojektowanej aplikacji należy najpierw skonfigurować okno startowe. W programie WindowMaker w menu Special\Configure wybieramy opcję WindowViewer. W otwartym oknie (Rys. 43.) przechodzimy na zakładkę Window Configuration. Domyślnie, w czasie pracy programu WindowViewer, pasek menu jest wyświetlany, natomiast chcemy żeby podczas uruchamiania było dostępne tylko okno „menu”. Należy zatem usunąć zaznaczenie opcji Menu Bar. To samo dotyczy opcji zamknięcia programu WindowViewer, ponieważ zakończenie aplikacji będzie możliwe po naciśnięciu przycisku w oknie „menu”. Odnaszamy wszystkie opcje w grupie Window i zaznaczamy opcję Hide Title Bar w celu ukrycia paska tytułowego. Następnie należy zaznaczyć opcję Disable ALT key, jeżeli operator ma być pozbawiony możliwości używania klawisza ALT i wywoływania poleceń z menu za pomocą skrótów klawiszowych. Tak samo postępujemy z klawiszami ESC i WIN, które również blokujemy, aby nie można było zamykać aplikacji, przełączać się między nimi czy też używać menu Start. Zaznaczamy jeszcze opcję Always Maximize, aby program WindowViewer był przez cały czas wyświetlany na pełnym ekranie. Przechodzimy na zakładkę Home Windows i zaznaczamy na liście okno „menu”. Będzie to pierwsze okno, które pojawi się po uruchomieniu programu WindowViewer. Naciskamy przycisk Zastosuj, a potem OK i zamykamy program WindowMaker.

Kolejnym krokiem jest publikacja zaprojektowanej aplikacji do katalogu. W środowisku Archestra IDE, w zakładce Template Toolbox, klikamy prawym przyciskiem myszy na obiekt reprezentujący w środowisku naszą aplikację i wybieramy opcję Publish InTouch Application. W przeglądarce folderów należy wskazać katalog lub utworzyć nowy, do którego zostanie opublikowana aplikacja. Zostanie pokazane okno, w którym pojawi się

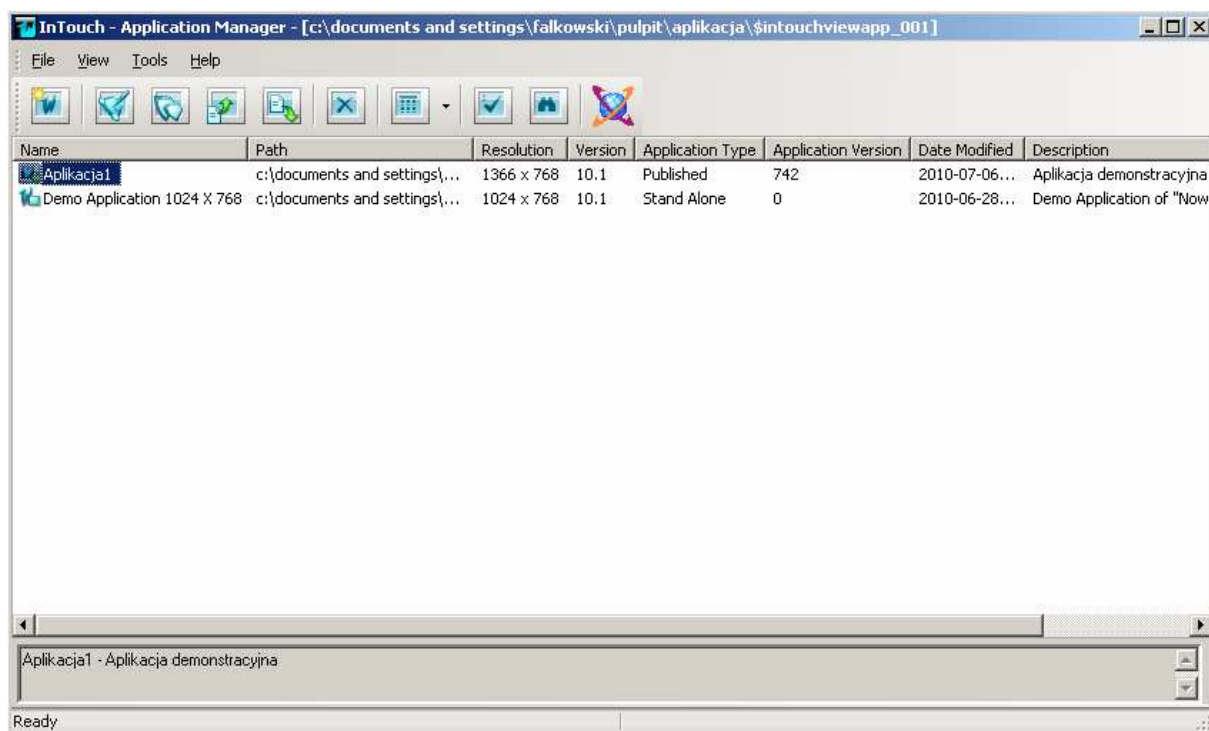
szereg komunikatów, a wśród nich m.in. informacja o skopiowaniu aplikacji do katalogu. Przyciskiem Close zamykamy okno. Katalog z aplikacją może zostać skopiowany na dowolny komputer. Aplikację można również opublikować zdalnie przez sieć na innym komputerze.



Rys. 43. Konfiguracja WindowViewer

W celu uruchomienia opublikowanej aplikacji należy z menu Start uruchomić Menedżer Aplikacji InTouch, czyli program InTouch (Rys. 44.). W programie wybieramy opcję Tools, a następnie Find Application. W wyświetlonym oknie należy wskazać katalog z opublikowaną aplikacją i nacisnąć przycisk OK. Aplikacja pojawi się na liście. Klikamy na nią prawym klawiszem myszy i wybieramy WindowViewer. Zostanie uruchomiona nasza aplikacja. Jeżeli przejdziemy do programu WindowMaker z poziomu Menedżera Aplikacji, to

nie będzie możliwa edycja symboli ArchestrA. Bez konfiguracji aplikacji dostępnej w ArchestrA IDE nie można edytować symboli ArchestrA.



Rys. 44. Uruchamianie aplikacji InTouch

5. Podsumowanie

Biorąc pod uwagę zapowiedzi nowych pakietów do programowania, opartych na nowych systemach operacyjnych firmy Windows, należy spodziewać się, że nastąpią zmiany w sposobie programowania. Zwiększy się jeszcze bardziej wykorzystanie komputerów osobistych klasy PC do sterowania procesami przemysłowymi. Połączone będą w jednym systemie możliwości sterowników i pakietów SCADA/HMI, co jest wyzwaniem dla producentów sterowników. Odpowiedzią są sterowniki o dużych pamięciach i pracujące na silnych platformach, np. sterowniki PAC.

Przed przystąpieniem do opracowywania pracy magisterskiej zostały postawione trzy główne cele: omówienie rozproszonych systemów sterowania, przegląd wybranego oprogramowania SCADA oraz wykonanie aplikacji demonstracyjnej w wybranym oprogramowaniu. W niniejszej dokumentacji przedstawiono realizację wszystkich wyżej wymienionych celów.

Opisano dwa oprogramowania SCADA czołowych firm (Wonderware i Rockwell Automation), utrzymujących pozycję liderów na światowym rynku systemów w zakresie nadrzędnej wizualizacji procesów, tj. InTouch i FactoryTalk View Site Edition. Przy pomocy oprogramowania InTouch została zaprojektowana aplikacja demonstracyjna służąca do wizualizacji i sterowania linią produkcyjną napojów przy użyciu komputera PC. Opierając się na tym przykładzie, została wyjaśniona cała procedura projektowania aplikacji, poczynając od konfiguracji komunikacji, przez projektowanie obiektów graficznych, aż po proste skrypty.

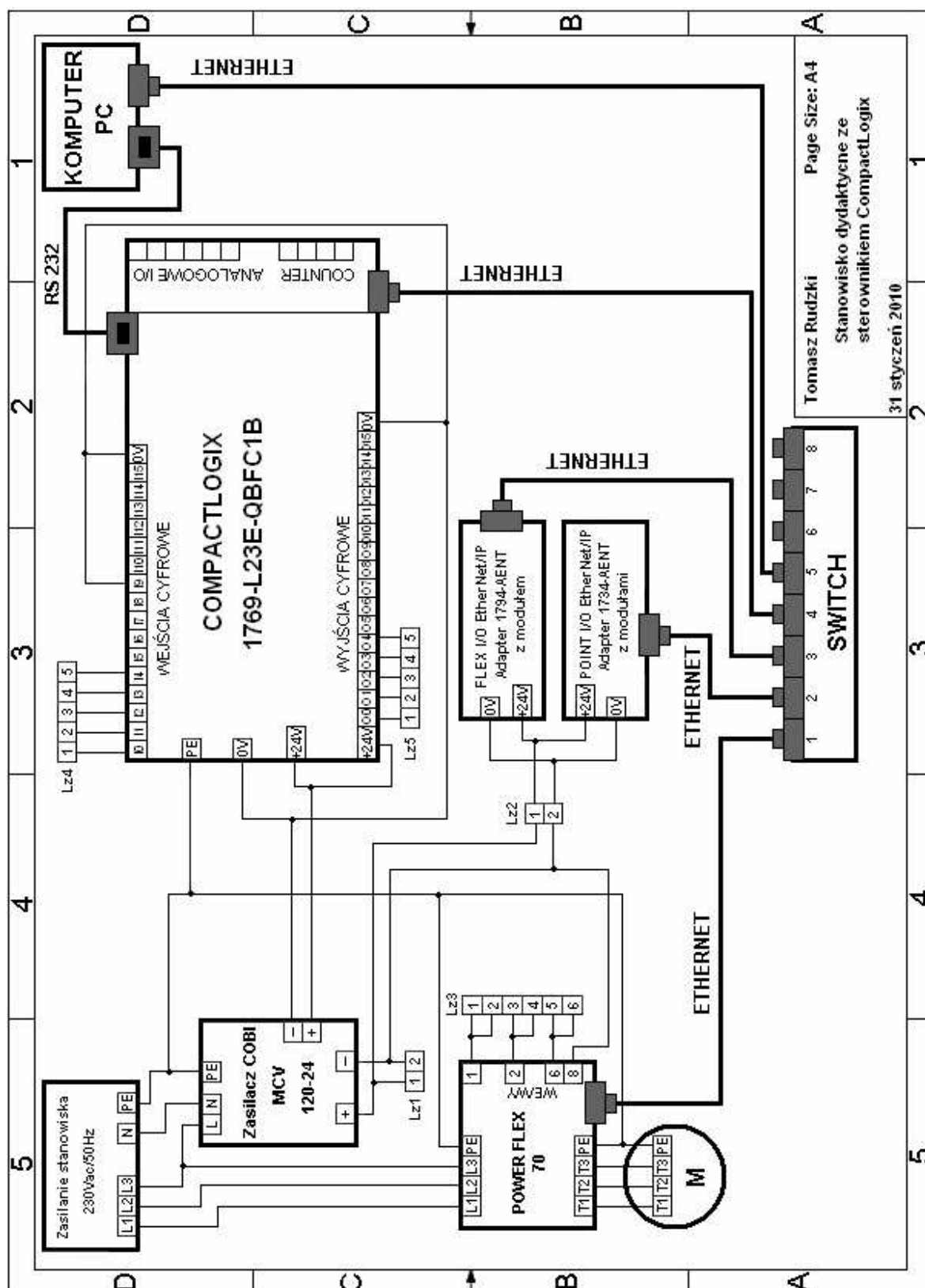
Po wykonaniu całej pracy można stwierdzić, że wszystkie założone cele zostały osiągnięte. Ciekawość tematyki była bardzo duża, co dodatkowo motywowało do dobrej pracy i pogłębiania zdolności w rozwiązywaniu problemów. Ponadto temat pracy magisterskiej obejmuje duży obszar z dziedziny automatyki, dzięki czemu dostarcza niezbędnej i jakże cennej wiedzy.

6. Literatura

- [1] W. Nawrocki, *Rozproszone systemy pomiarowe*, Wydawnictwa Komunikacji i Łączności, Warszawa 2006.
- [2] J. Kwaśniewski, *Sterowniki PLC w praktyce inżynierskiej*, Wydawnictwo BTC, Legionowo 2008.
- [3] J. Kasprzyk, *Programowanie sterowników przemysłowych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2006.
- [4] K. Pietrusewicz, P. Dworak, *Programowalne sterowniki automatyki PAC*, Wydawnictwo Nakom, Poznań 2007.
- [5] A. Maczyński, *Sterowniki programowalne PLC. Budowa systemu i podstawy programowania*, Wydawnictwo Astor Sp. z o.o., Kraków 2001.
- [6] T. Mikulczyński, *Automatyzacja procesów produkcyjnych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2006.
- [7] *Automatyka, Podzespoły, Aplikacje nr 11/2009*, Wydawnictwo AVT Korporacja Sp. z o.o., Warszawa 2009.
- [8] *Biuletyn Automatyki nr 4/2007*, Wydawnictwo Astor Sp. z o. o., Kraków 2007.
- [9] *Biuletyn Automatyki nr 2/2008*, Wydawnictwo Astor Sp. z o. o., Kraków 2008.
- [10] *InTouch 9.5, Podręcznik użytkownika*, Invensys Systems, Inc., 2005.
- [11] *FactoryTalk View Site Edition, User's Guide*, Rockwell Automation, 2009.
- [12] <http://www.astor.com.pl/> – strona internetowa ASTOR – dystrybutor oprogramowania SCADA.
- [13] <http://www.wonderware.com.pl/> – strona internetowa Wonderware – producent oprogramowania InTouch.
- [14] <http://www.rockwellautomation.pl/> – strona internetowa Rockwell – producent oprogramowania FT View SE.
- [15] <http://elektronikab2b.pl/technika/10304-ethernet-zdobywa-sieci-przemysowe>.
- [16] T. Rudzki, *Układy sterowania rozproszonego – budowa i oprogramowanie*, Toruń 2010 – praca inżynierska.

7. Załączniki

Załącznik 1. Schemat elektryczny połączenia elementów stanowiska dydaktycznego [16]



Załącznik 2. Program demonstracyjny sterownika PAC [16]

MainRoutine - Ladder Diagram

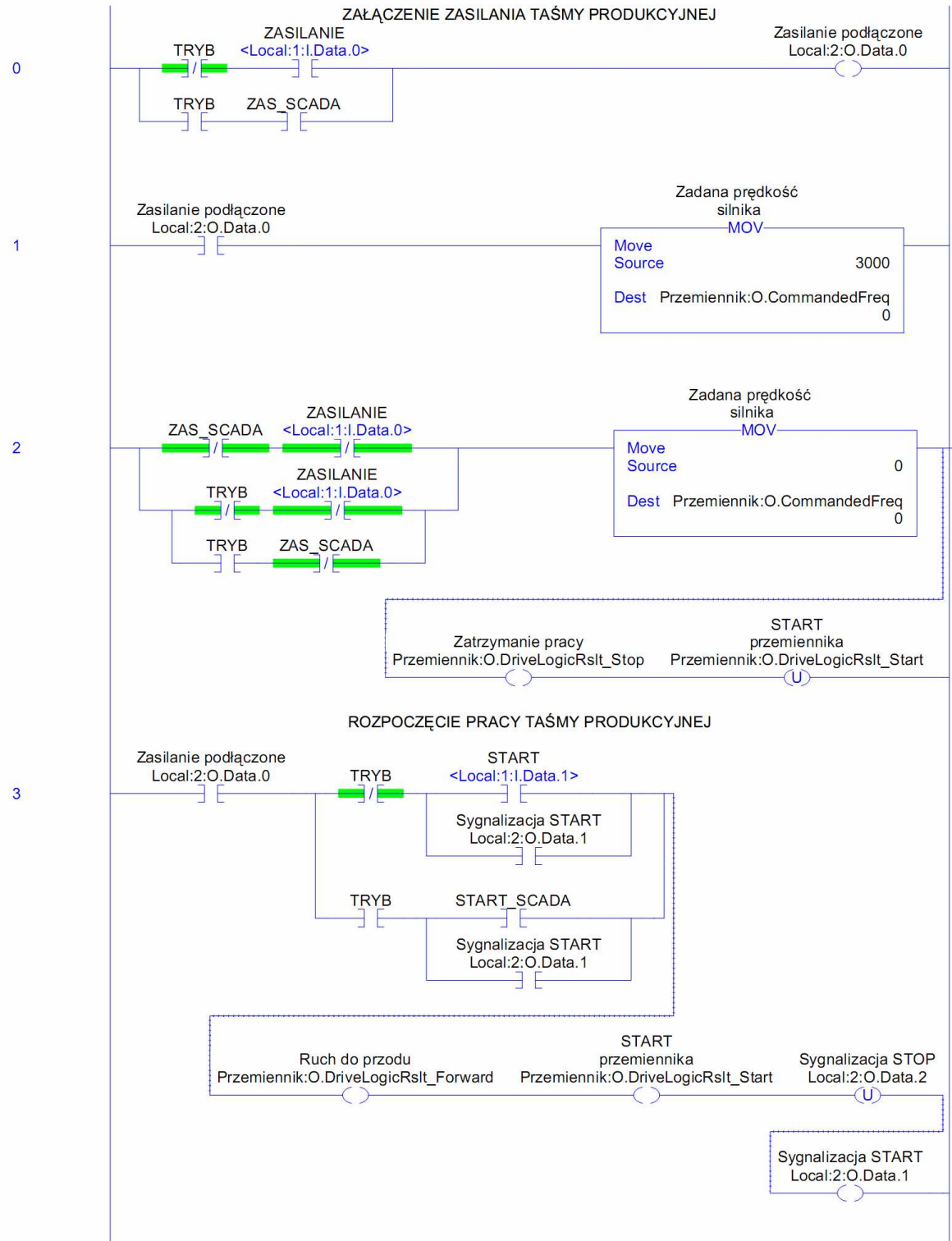
Projekt:MainTask:MainProgram

Total number of rungs in routine: 21

Page 1

2009-12-01 10:05:57

...acy\Pulpit\Program drabinkowy\ProgramDemonstracyjnySCADA.ACD



RSLogix 5000

