

INSTYTUT AUTOMATYKI I ROBOTYKI
WYDZIAŁ MECHATRONIKI
POLITECHNIKA WARSZAWSKA



Cezary Szachewicz
(nr albumu: PZ/63813)

**Projekt i wykonanie systemu wizualizacji i
sterowania stanowiska laboratoryjnego układu
regulacji poziomu cieczy w zbiornikach otwartych**
(Praca dyplomowa inżynierska)

**Design and performance visualisation and control
system of the laboratory stand designated to control
the level of the liquid in open tanks**

Promotor: dr inż. Danuta Holejko

Warszawa, grudzień 2008

Streszczenie

Niniejsza praca jest przedstawieniem wyników projektowania oraz wykonania systemu wizualizacji i sterowania stanowiska laboratoryjnego układu regulacji poziomu cieczy w zbiornikach otwartych. System wykonano za pomocą oprogramowania Proficy HMI/SCADA iFIX4.0. Urządzeniem sterującym zastosowanym w układzie regulacji, jest regulator mikroprocesorowy LB-600 firmy LAB-EL. System umożliwia pracę w strukturze jednoobwodowej lub kaskadowej. Oprócz wizualizacji procesu, system umożliwia rejestrację oraz archiwizację przebiegów układu. Dodatkowo, na podstawie wyników badań oraz analizy porównawczej algorytmów zaimplementowanych w regulatorze LB-600 oraz w oprogramowaniu iFIX4.0, przedstawiono problematykę związaną z cyfrowymi postaciami algorytmu PID.

Summary

The thesis presents the development and performance process results of the visualization and control system of laboratory stand that is designated to control the level of the liquid in open tanks. The system was created using the Proficy HMI/SCADA iFIX4.0 software. Hardware control device used in the above control system was LB-600 microprocessor PID-controller produced by LAB-EL Company. The system enables work in single-loop or cascade control system. Except process visualization the system enable registration and archiving all process signal flow of programmed structure of closed-loop system. Additionally, on the basis of the results of the experiments and comparative analysis of implemented algorithm in the LB-600 controller and in the iFIX4.0 system process data base, issues connected with digital form of PID algorithm was performed.

Spis treści

1	CEL I ZAKRES PRACY	5
2	OPIS STANOWISKA LABORATORYJNEGO	6
3	REGULATOR LB600	8
3.1	BUDOWA REGULATORA LB600	8
3.2	STRUKTURA FUNKCJONALNA REGULATORA	10
3.2.1	<i>Budowa funkora</i>	11
3.2.2	<i>Funkcje warstw</i>	12
3.3	WSPÓLPRACA W SYSTEMACH KOMPUTEROWYCH	14
4	PROTOKÓŁ KOMUNIKACYJNY MODBUS	17
4.1	PODSTAWOWE INFORMACJE	17
4.2	TRANSMISJA SZEREGOWA W SIECIACH MODBUS	18
4.2.1	<i>Tryb ASCII</i>	19
4.2.2	<i>Tryb RTU</i>	19
4.3	RAMKI KOMUNIKATÓW	20
4.3.1	<i>Ramka w trybie ASCII</i>	20
4.3.2	<i>Ramka w trybie RTU</i>	20
4.3.3	<i>Ramka znaku</i>	21
4.4	METODY SPRAWDZANIA BŁĘDÓW.....	22
4.4.1	<i>Sprawdzanie parzystości (parity checking)</i>	22
4.4.2	<i>Sprawdzanie LRC (Longitudinal Redundancy Check)</i>	23
4.4.3	<i>Sprawdzanie CRC (Cyclic Redundancy Check)</i>	23
4.5	KODY FUNKCJI W PROTOKOLE MODBUS	24
5	SYSTEM SCADA	25
5.1	KONCEPCJA SYSTEMU SCADA	25
5.2	HUMAN-MACHINE INTERFACE (HMI).....	27
5.3	REMOTE TERMINAL UNIT (RTU).....	28
6	PROFICY HMI/SCADA I FIX	30
6.1	PODSTAWOWE FUNKCJE HMI/SCADA	30
6.2	ARCHITEKTURA I FIX	32
6.3	TECHNOLOGIE WYKORZYSTYWANE W I FIX.....	33
6.4	DRAJWER WE/WY 7.X.....	36
6.5	PROCESOWA BAZA DANYCH	42
6.6	WORKSPACE	51
6.7	TWORZENIE APLIKACJI HMI/SCADA	53
7	SYSTEM WIZUALIZACJI I STEROWANIA UKŁADU REGULACJI POZIOMU CIECZY W ZBIORNIKACH OTWARTYCH	55
7.1	ANALIZA PROJEKTU	55
7.1.1	<i>Informacje o urządzeniach obiektowych</i>	55
7.1.2	<i>Szczegółowa analiza procesu</i>	56
7.1.2.1	<i>Jednoobwodowy układ automatycznej regulacji</i>	57
7.1.2.2	<i>Kaskadowy układ automatycznej regulacji</i>	58
7.1.2.3	<i>Jednoobwodowy układ automatycznej regulacji (Soft Control)</i>	59
7.1.3	<i>Wybór odpowiednich bloków</i>	60
7.1.4	<i>Schemat przepływu sygnałów w procesowej bazie danych</i>	61
7.2	PROJEKT SYSTEMU.....	63
7.2.1	<i>Wymagania funkcjonalne aplikacji</i>	63
7.2.2	<i>Podział funkcjonalny ekranów synoptycznych</i>	63
7.2.3	<i>Projekt graficzny ekranów synoptycznych</i>	67
7.2.3.1	<i>Listwa górna</i>	68
7.2.3.2	<i>Listwa dolna</i>	68
7.2.3.3	<i>Ekran kategorii: obiekt</i>	69

7.2.3.4	Ekran kategorii: wykres.....	72
7.2.3.5	Ekran kategorii: schemat funkcjonalny	72
7.2.3.6	Ekran kategorii: ustawienia	73
7.2.4	<i>Analiza wykonywalności</i>	74
7.2.4.1	Wykonanie rysunków	74
7.2.4.2	Algorytmy inkrementacji oraz dekrementacji wartości bazy danych.....	75
7.2.4.3	Opracowanie metody sterowania ręcznego	77
7.2.4.3.1	Sterowanie ręczne zrealizowane za pomocą adresów warstwy 9	78
7.2.4.3.2	Sterowanie ręczne zrealizowane za pomocą adresów warstwy 8.	79
7.2.4.3.3	Sterowanie ręczne zrealizowane za pomocą bloku PID dodatkowego kanału.....	80
7.2.4.4	Opracowanie algorytmu obsługi samostrojzenia.....	81
7.3	IMPLEMENTACJA	85
7.3.1	<i>Połączenie obiektów animowanych na ekranach z bazą danych</i>	86
8	ANALIZA PORÓWNAWCZA ALGORYTMÓW PID REALIZOWANYCH PRZEZ REGULATOR LB-600 ORAZ SYSTEM IFIX.....	89
8.1	IDEALNY ORAZ RZECZYWISTY ALGORYTM PID	89
8.2	CYFROWE POSTACIE ALGORYTMU PID.....	90
8.3	ALGORYTM PID, REGULATORA LB-600	91
8.4	ALGORYTM PID, SYSTEMU IFIX4.0.....	92
8.5	ODPOWIEDZI SKOKOWE REGULATORÓW LB-600 ORAZ SOFT CONTROL.....	93
8.5.1	<i>Algorytm P</i>	93
8.5.2	<i>Algorytm PI</i>	95
8.5.3	<i>Algorytm PID</i>	96
8.6	PRZEBIEGI PRZEJŚCIOWE UAR Z REGULATORAMI LB-600 ORAZ SOFT CONTROL.....	102
8.6.1	<i>Identyfikacja obiektu regulacji oraz dobór nastaw regulatora, metodą cyklu granicznego (Zieglera-Nicholsa)</i>	102
8.6.2	<i>Identyfikacja obiektu z użyciem procedur samostrojzenia regulatora LB-600</i>	104
8.6.3	<i>Przebiegi przejściowe obiektu z regulatorem LB-600 oraz Soft Control</i>	109
9	WNIOSKI	116
10	LITERATURA	118
11	ZAŁĄCZNIKI.....	119
11.1	ZAŁĄCZNIK A – WYKAZ ZMIENNYCH I ADRESÓW	120
11.2	ZAŁĄCZNIK B - INSTALACJA I KONFIGURACJA	130
11.2.1	<i>Instalacja Proficy HMI/SCADA iFIX 4.0</i>	130
11.2.2	<i>Instalacja drajwera komunikacyjnego MB1</i>	132
11.2.3	<i>Konfiguracja drajwera MB1</i>	135
11.2.4	<i>Konfiguracja iFIX4.0</i>	138
11.2.4.1	Dodanie drajwera MB1	138
11.2.4.2	Import bazy danych	142
11.2.4.3	Dodanie oraz konfiguracja zmiennych globalnych.....	148
11.2.4.4	Konfiguracja danych archiwalnych	150
11.2.4.5	Konfiguracja trybu wykonywalnego programu WorkSpace	154
11.2.4.6	Konfiguracja zabezpieczeń.....	155

1 Cel i zakres pracy

Celem pracy jest zaprojektowanie oraz wykonanie interfejsu HMI/SCADA regulatora LB600, pracującego w układzie automatycznej regulacji o strukturze jednoobwodowej oraz kaskadowej.

Aplikacja została zrealizowana za pomocą oprogramowania Proficy HMI/SCADA iFix4.0. Z uwagi na duże możliwości pakietu, autor postanowił dodatkowo zaimplementować w aplikacji sterowanie typu Soft Control, gdzie sygnał sterujący jest generowany przez aplikację oraz przekazywany za pomocą regulatora do elementu wykonawczego. Zrealizowana aplikacja posiada następujące cechy:

1. Przejrzysta forma prezentacji danych obiektowych za pomocą
 - a. Schematu obiektu
 - b. Schematu blokowego układu regulacji
 - c. Schematu funkcjonalnego regulatora
2. Możliwość obsługi za pomocą panelu dotykowego
3. Zintegrowany system bezpieczeństwa, uniemożliwiający dostęp do trybu konfiguracyjnego osób niepowołanych
4. Możliwość programowania w regulatorze następujących struktur:
 - a. Jednoobwodowej
 - b. Kaskadowej
 - c. Jednoobwodowej z sygnałem sterującym generowanym poprzez komputer
5. Archiwizacja danych procesowych
6. Prezentacja danych na wykresach
7. Eksport danych historycznych do pliku CSV

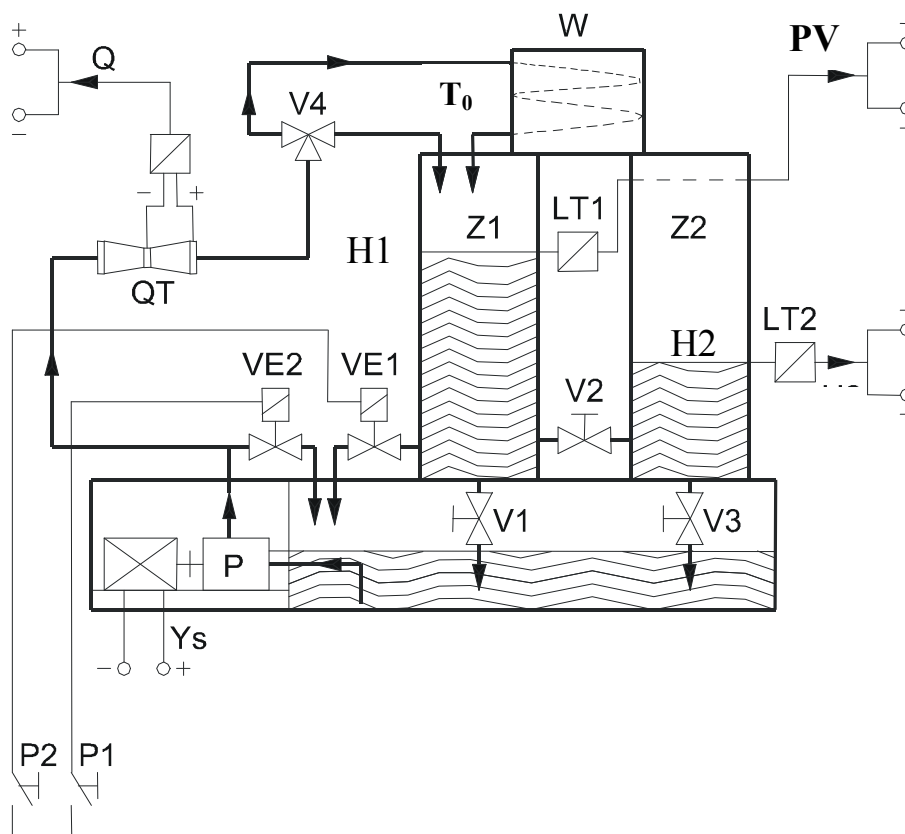
Autor pracy, w oparciu o własne doświadczenia, podjął próbę usystematyzowania procesu tworzenia aplikacji w środowiskach HMI/SCADA. W tym celu w rozdziale 6, pogrupowano oraz uporządkowano czynności towarzyszące procesowi projektowania aplikacji. Rozdział 7 jest przedstawieniem wyniku prac związanych z projektowaniem oraz wykonaniem aplikacji według zaproponowanego schematu.

2 Opis stanowiska laboratoryjnego.

Obiektem regulacji jest proces zmiany poziomu cieczy w odpowiednio połączonych zbiornikach Z1, Z2 (Rys. 2.1). Zbiorniki połączone są za pomocą zaworu odcinającego. Wielkościami regulowanymi mogą być:

- wysokość słupa cieczy H1 w zbiorniku Z1
- wysokość słupa cieczy H2 w zbiorniku Z2

Celem układu regulacji jest utrzymanie wielkości regulowanej na zadanym poziomie, niezależnie od działających zakłóceń. Zakłócenia mogą być generowane poprzez, otwarcie zaworu VE1, VE2.



Rys. 2.1 Schemat ogólny części obiektowej.

Oznaczenia:

Z1, Z2 – zbiorniki,

V1, V3 – zawory odpływowe,

V2 – zawór łączący zbiorniki Z1 i Z2,

V4 – zawór trójdrożny,

W – wężownica, T_0 – opóźnienie transportowe,

VE1 – zawór odcinający elektromagnetyczny odpływu ze zbiornika Z1,

VE2 – zawór odcinający elektromagnetyczny zrzutu wody na tłoczeniu z pompy,

H1, H2, - wysokość słupa cieczy w zbiorniku Z1, Z2,

PV – sygnały wyjściowe z przetworników poziomu cieczy LT1, LT2,

Q – wyjście z przetwornika QT przepływu (przepływomierz zwężkowy),

P – pompa o zmiennej wydajności,

Y_s – sygnał sterujący pompą (wyjściowy z regulatora mikroprocesorowego)

P1, P2 – przełączniki zaworów elektromagnetycznych.

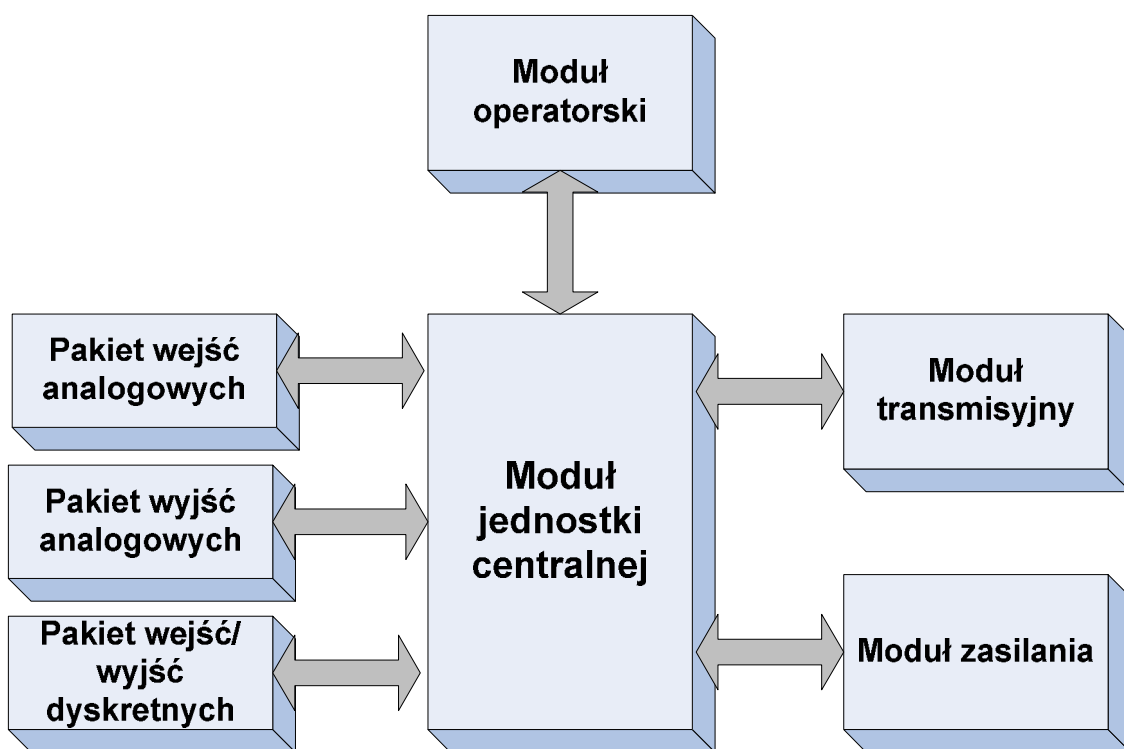
W zależności od konfiguracji stanów zaworów **V2** i **V4** oraz zastosowaniu na dopływie specjalnie skonstruowanej wężownicy **W**, można realizować różne właściwości obiektu regulacji. Mogą to być właściwości typu:

- inercyjnego I rzędu (zawór **V2** zamknięty, napływ cieczy bezpośredni, wielkość regulowana – **H1**),
- inercyjnego I rzędu z opóźnieniem (zawór **V2** zamknięty, zawór **V4** w położeniu napływ przez wężownice, wielkość regulowana **H1**),
- inercyjnego II rzędu (zawór **V2** otwarty, napływ cieczy bezpośredni, wielkość regulowana – **H2**),
- inercyjnego II rzędu z opóźnieniem (zawór **V2** otwarty, zawór **V4** w położeniu napływ przez wężownice, wielkość regulowana – **H2**).

3 Regulator LB600

Regulator LB600, jest specjalizowanym modułowym urządzeniem elektronicznym, zaprojektowanym do przetwarzania sygnałów pochodzących z obiektu w celu wygenerowania odpowiedniego sygnału sterującego. Poniżej zostanie omówiona budowa urządzenia, opis struktury funkcjonalnej, oraz przykłady współpracy przyrządu w systemach komputerowych.

3.1 Budowa regulatora LB600



Rys. 3.1 Schemat blokowy regulatora LB-600

Moduł jednostki centralnej, odpowiedzialny jest za przetwarzanie sygnałów uzyskanych z pakietów wejść oraz generowanie sygnału sterującego i realizację wszystkich funkcji

regulatora. Procesor zastosowany w jednostce centralnej charakteryzuje się między innymi następującymi cechami:

- Zegar 40MHz
- 50ns cykl przetwarzania
- 4 kanały PWM
- watchdog

Moduł operatorski, odpowiada za kontakt regulatora z użytkownikiem. Zapewnia obsługę przycisków (odciążając z tego zadania, procesor główny) oraz elementów wyświetlających dane.

Moduł transmisyjny, odpowiedzialny jest za komunikację z komputerem.

Pakiet wejść analogowych, może występować w dwóch wykonaniach:

- Pakiet zbudowany z 2 przetworników SIGMA-DELTA charakteryzujący się dokładnymi lecz niezbyt szybkimi pomiarami. W zależności od konfiguracji podłączenia sygnałów, może obsłużyć do 10 wejść analogowych, a możliwość umieszczenia do trzech takich pakietów w przyrządzie daje obsługę do 30 wejść. Pakiet w tym wykonaniu może realizować pomiary unipolarne i /lub bipolarne. Przetwarzanie analogowo-cyfrowe wymaga autokalibracji, filtracji, itp. Dlatego całkowity czas konwersji sygnału może w skrajnych przypadkach wynosić kilkadziesiąt ms na jedno wejście. Pakiet ten posiada pełną separację galwaniczną która, zrealizowana jest poprzez umieszczenie optoseparatorów pomiędzy przetwornikiem A/C a procesorem.
- Drugi rodzaj pakietu to układ prostszy polegający na multipleksowaniu sygnałów analogowych wejściowych, porównaniu z odpowiednim poziomem odniesienia i przesyłanych na jedno z wejść przetwornika a/c procesora. Pakiet tego typu przetwarza sygnał znacznie szybciej tzn. kilka μ s na jedno wejście, pomiar jest jednak mniej dokładny. Pakiet ten nie posiada separacji galwanicznej.

Pakiet wyjść analogowych, jeden dla obu wyjść służący do uzyskania dwu w pełni niezależnych, odseparowanych galwanicznie sygnałów wejściowych napięciowych lub/i prądowych regulatora. Pakiet wyjściowy wykonany jest w oparciu o 16-bitowe przetworniki

cyfrowo-analogowe. Nie są te jednak układy identyczne. Pierwszy z nich umożliwia generację sygnału 0/4...20mA lub 0/1...5V, drugi natomiast generuje sygnały niezerowe czyli 4...20mA lub 1...5V.

Pakiet wejść/wyjść dyskretnych, jeden dla 8 wejść i 6 wyjść.

Wejścia dyskretnie mogą być:

- Napięciowe 16...25Vdc, 3...5Vdc (pełna separacja galwaniczna)
- Zwarciove (brak separacji)

Wyjścia dyskretnie mogą być:

- Przekąźnikowe (zestyk przełączny) dla obciążenia 24Vdc (ac), 200mA
- Przekąźnikowe wysokiej mocy(zestyk zwierny) dla obciążeń 220Vac, 1A/fazę (w tym wykonaniu brak wejść dyskretnych)
- Triakowe (optotriak) dla obciążenia 220Vac, 1A/fazę.

Moduł zasilania, posiada następujące zabezpieczenia:

- Zwarciove
- Przeciążeniowe
- Nadnapięciowe

Dodatkowo generuje sygnały Reset oraz Power Fail obsługiwane przez diagnostykę w module jednostki centralnej. Może występować w wykonaniu 230Vac, 50Hz, oraz 24Vdc.

3.2 Struktura funkcjonalna regulatora

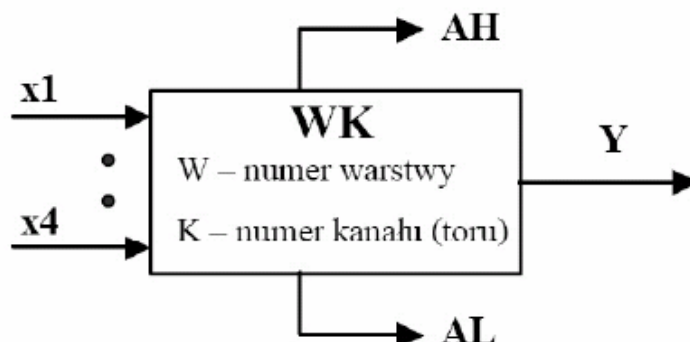
Logiczna struktura regulatora, składa się z pojedynczych bloków zwanych funktoami. Każdy funkto w zależności od pełnionej roli posiada określoną ilość wejść, wyjść oraz przyporządkowaną funkcję, którą może realizować. Realizowane funkcje są zgrupowane w jednostkach logicznych zwanych „warstwami”. W obrębie warstwy może występować kilka identycznych bloków, dlatego też w celu bardziej jednoznacznego określenia położenia poszczególnych funktoów została wprowadzona jednostka logiczna „tor” (kanał). Warstwy i kanały tworzą więc pewnego rodzaju macierz funktoów. Pozycję funktoora można jednoznacznie określić poprzez podanie adresu składającego się numeru warstwy i toru.

3.2.1 Budowa funkтора

Każdy funktor posiada parametry które należy określić w celu jego prawidłowego działania. Parametryzacja funkтора pozwala na:

- Wskazanie pochodzenia sygnału wejściowego,
- Określenia sposobu przetwarzania sygnału wejściowego przez blok
- Zachowanie bloku w zależności od stanu innych funktorów

Poniżej przedstawiono przykładowy funktor:



Rys. 3.2 Przykład funkтора regulatora LB-600.

Większość funktorów dostępnych w regulatorze posiada możliwość informowania o wartości sygnału przez niego przetwarzanego. Informacja jest przekazywana za pomocą rejestru stanów binarnych (RSB). Każdy funktor posiadający tę własność ma z góry określony adres w RSB. Przeważnie za pomocą RSB jest przekazywana informacja o wystąpieniu alarmu danego funkтора.

Programując strukturę regulatora, zazwyczaj posługujemy się funktorami leżącymi w jednym torze poprzez łączenie funktorów warstw początkowych z funktorami warstw wyższych. Nie oznacza to jednak iż musimy wykorzystać wszystkie bloki w danym torze oraz, że muszą one być połączone między sobą w kolejności występowania warstw. Ze względu na swobodną możliwość określania pochodzenia sygnału wejściowego, istnieje możliwość łączenia funktorów leżących w różnych warstwach i kanałach w celu realizacji dowolnych funkcji. Poniżej przedstawiono zestawienie wszystkich funktorów dostępnych w regulatorze.

		WARSTWA											
		0	1	2	3	4	5	6	7	8	9	A	B
KANAL (TOR)	Parametry generalne	1	11	21	31	41	51	61	71	81	91	A1	Skalowanie wejść analogowych
		2	12	22	32	42	52	62	72	82	92	A2	
		3	13	23	33	43	53	63	73	83		A3	
		4	14	24	34	44	54	64	74	84		A4	
		5	15	25	35	45	55	65	75	85		A5	
		6	16	26	36	46	56	66	76	86		A6	
		7	17	27	37	47	57	67	77	87			
		8	18	28	38	48	58	68	78	88			
		9	19		39	49	59	69					
		A	1A		3A	4A	5A	6A					
		B			3B	4B	5B	6B					
		C			3C	4C	5C	6C					
		D			3D	4D	5D	6D					
		E			3E	4E	5E	6E					
		F			3F	4F	5F	6F					

Rys. 3.3 Zestawienie funktorów regulatora LB-600, w poszczególnych warstwach programowych.

3.2.2 Funkcje warstw

Jak już wspomniano wcześniej, warstwy grupują funktry, według realizowanych przez nie funkcji. W regulatorze LB600 istnieją dodatkowe dwie warstwy nie zawierające funktorów jest to warstwa 0 oraz B. Poniżej w dużym skrócie opisano wszystkie warstwy regulatora.

Warstwa 0 – umożliwia ustawienie parametrów, dotyczących całego regulatora oraz pozostałych warstw min. Data i Czas, alarmy uaktywniane ustawioną datą, parametry transmisji, deklaracja urządzeń zewnętrznych firmy LAB-EL, ustawienia hasła głównego oraz pomocniczego itp.

Warstwa 1 – obsługuje wejścia analogowe oraz umożliwia ustawienia parametrów przetwarzania sygnału wejściowego takich jak:

- Ustawienie jednostek fizycznych
- Deklaracja progów alarmowych

- Deklaracja rodzaju sygnału wejściowego
- Ustawienie algorytmu przetwarzania sygnału
- Ustawienie współczynników przetwarzania sygnału
- Ustawienie filtracji sygnału wejściowego
- Inne

Warstwa 2 – obsługuje wejścia binarne

Warstwa 3, 4, 5, 6 – funktry tych warstw dostarczają przede wszystkim dodatkowych funkcji arytmetycznych, czasowych i logicznych sygnałów analogowych oraz binarnych takich jak:

- Timery
- Całkowanie
- Liniowy przyrost sygnału (Ramp)
- Klucz binarny
- Średnia arytmetyczna
- Funkcje logiczne (OR, AND, NOT, XOR, NAND itp.)

Warstwa 7 – dostarcza funktry odpowiedzialne za główną funkcje regulatora oraz umożliwia ich parametryzację w zakresie:

- Trybu pracy, z możliwością wyboru:
 - Regulatora o wyjściu ciągłym
 - Regulatora 2P
 - Regulatora 3P, ze sprzężeniem wewnętrznym lub zewnętrznym
- Algorytmu regulacji, z możliwością wyboru m.in.:
 - PID
 - P z nastawianym punktem pracy
 - PID Ratio
 - PID Auto Ratio
 - PID Auto Bias
 - PID NL1 (wzmocnienie w funkcji odchyłki)
 - PID NL2 (czas zdwojenia w funkcji odchyłki)
 - PID GAP

- Samostrojzenia
- Wyboru funkcji dodatkowych takich jak:
 - Fuzzy Logic
 - Feed Forward
 - PID strefowe
 - Backup
 - Programator wielokrokowy
 - Programator dwustrefowy

Warstwa 8 – zapewnia dodatkowy zbiór funkcji przetwarzania sygnału analogowego.

Warstwa 9 – dostarcza obsługę pakietów wyjść analogowych.

Warstwa A – dostarcza obsługę pakietów wyjść dyskretnych.

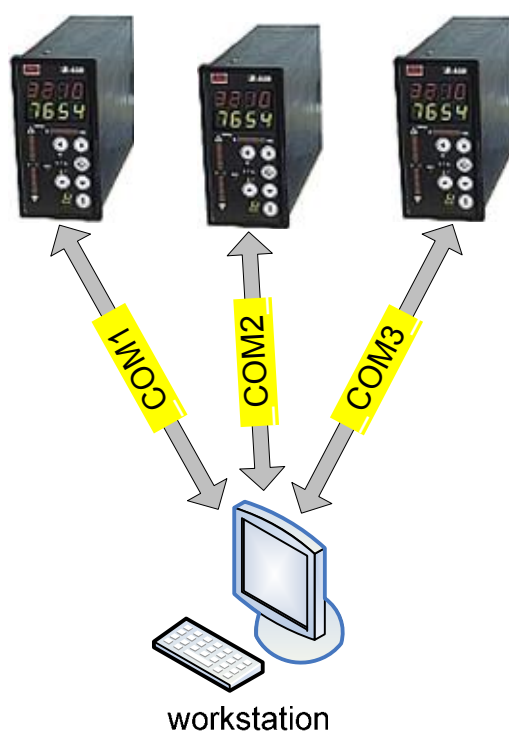
Warstwa B – umożliwia skalowanie wejść oraz wyjść analogowych

3.3 Współpraca w systemach komputerowych

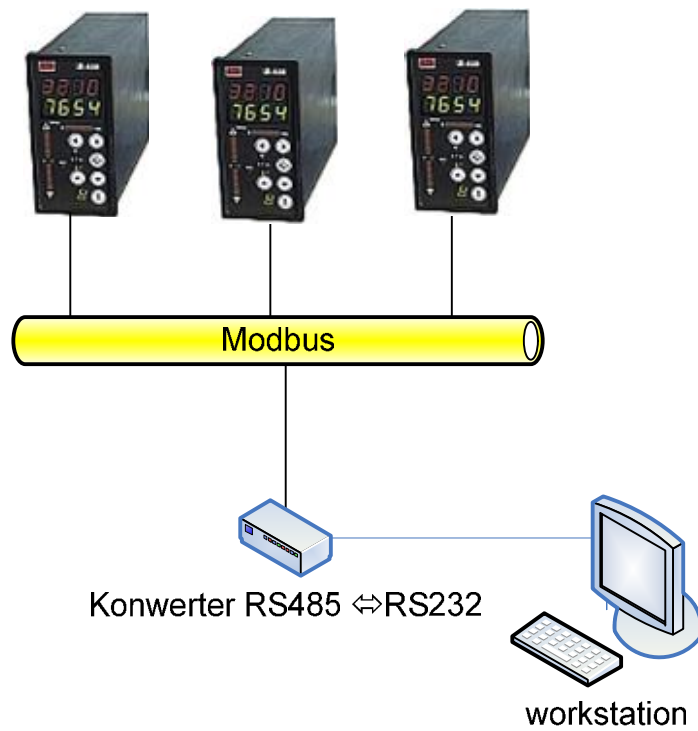
Regulator LB600, poprzez moduł komunikacyjny, umożliwia współpracę z komputerem. Komunikacja pomiędzy tymi urządzeniami odbywa się za pośrednictwem szeregowego połączenia, z wykorzystaniem protokołu Modbus. Komputer „widzi” regulator jako zbiór rejestrów, na których może przeprowadzać operacje odczytu oraz zapisu. Regulator udostępnia ponad 3000 adresów, co stanowi niebagatelną ilość, w porównaniu do innych urządzeń tej klasy. Dzięki udostępnieniu tak dużej liczby rejestrów, poprzez komputer można uzyskać dostęp do większości parametrów i funkcji oferowanych przez regulator. A dzięki temu zapewnić zdalną kontrolę urządzenia z dowolnego miejsca w zakładzie bądź nawet z innej miejscowości.

Duża skalowalność, pozwala nie tylko na pełną akwizycję danych procesowych, lecz wręcz daje swobodę tworzenia specjalizowanego oprogramowania pełniącego różnorodne funkcje, które normalnie nie są zaimplementowane w regulatorze.

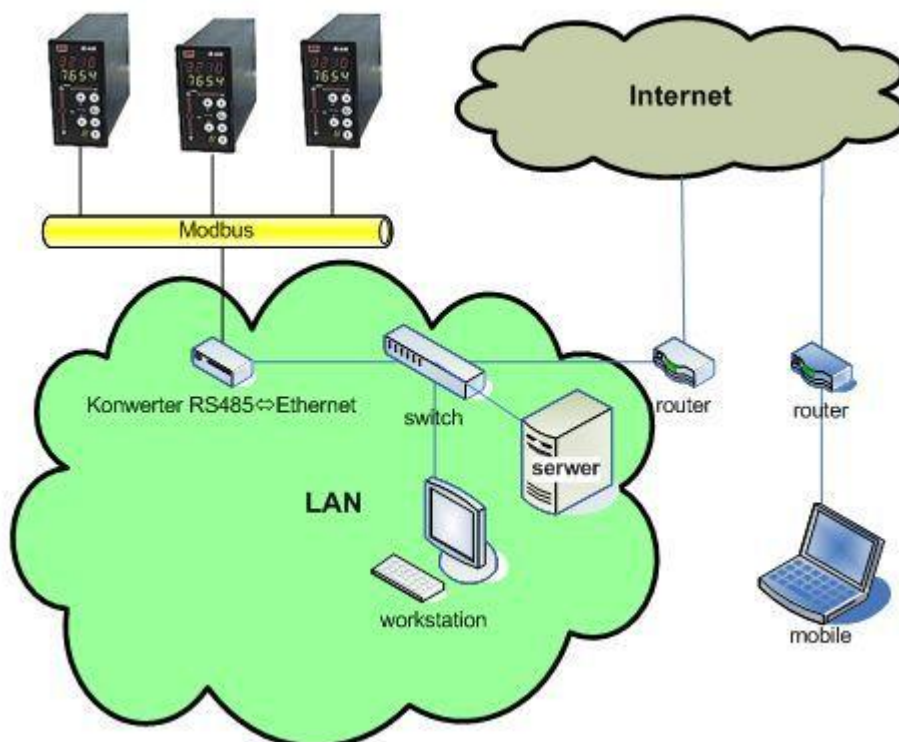
Dzięki obecnym na rynku, konwerterom pomiędzy różnymi interfejsami, możliwe jest połączenie kilku regulatorów w zależności od potrzeb do jednego komputera lub też do sieci komputerowej. Na poniższych rysunkach przedstawiono przykładowe sposoby realizacji komunikacji pomiędzy regulatorem a komputerem.



Rys. 3.4 Przykład komunikacji z wykorzystaniem wieloportowej karty COM.



Rys. 3.5 Przykład komunikacji z wykorzystaniem konwertera RS484/RS232.



Rys. 3.6 Przykład komunikacji z wykorzystaniem konwertera RS485/Ethernet.

4 Protokół komunikacyjny Modbus

4.1 Podstawowe informacje

Protokół Modbus został opracowany w 1979 roku przez firmę Modicon. Komunikacja za jego pośrednictwem, polega na wymianie komunikatów pomiędzy jednostkami Master-Slave (nadrzędny-podrzędny). Komunikat może zawierać żądanie wykonania polecenia (przesyłane przez jednostkę master) albo odpowiedź na żądanie (przesyłane przez jednostkę slave), para żądanie-odpowieź nazywana jest transakcją. Modbus jest w pełni otwartym standardem i jednym z najczęściej stosowanych protokołów sieciowych używanych w przemyśle.

Urządzenia wyposażone w możliwość komunikacji za pośrednictwem protokołu Modbus, używają techniki master-slave, w którym, tylko jedno urządzenie (master) inicjuje przesyłanie komunikatów (żądań). Inne urządzenia (slaves) odpowiadają, wysyłając żądane dane do urządzenia nadrzędnego albo podejmują działania określone w odebranych komunikacie. Urządzeniem typu slave może być każde urządzenie zewnętrzne np. zawór, przetwornik pomiarowy, sterownik PLC, które przetwarza żądania i wysyła na nie odpowiedzi.

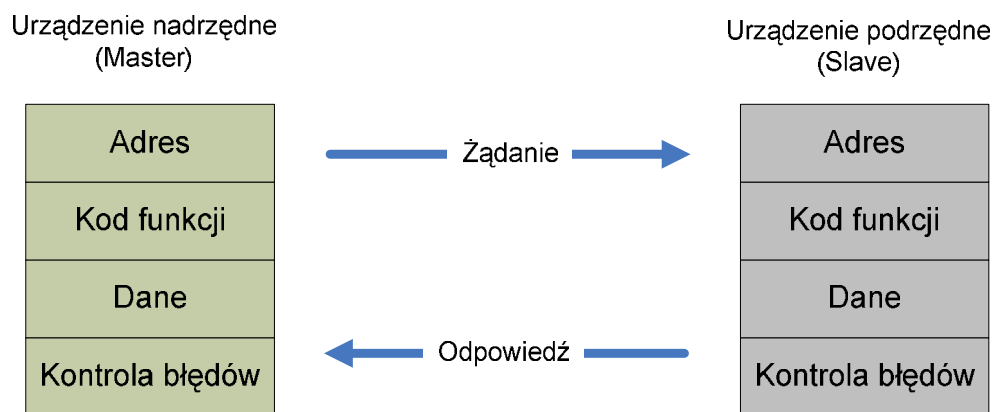
Jednostka master może wysłać komunikaty indywidualnie do każdej zaadresowanej jednostki slave bądź też, wysłać komunikaty w trybie rozgłoszenia (broadcast) do wszystkich jednostek slave. Urządzenie podrzędne przesyła odpowiedź na żądanie tylko jeżeli zostało indywidualnie zaadresowane. W żądaniu wysłanym przez jednostkę master są umieszczone następujące pola:

- adres jednostki slave,
- kod funkcji, którą ma wykonać jednostka slave np. zapis, odczyt wartości rejestru,
- dane niezbędne do wykonania żądanej funkcji, np. wartość którą należy wprowadzić do odpowiedniego rejestru, numer rejestru początkowego z którego należy odczytać wartość,
- informacje o kontroli błędów pozwalające jednostce slave sprawdzić poprawność odebranych danych.

Komunikat odpowiedzi jednostki slave zawiera następujące pola

- adres jednostki slave,
- kod funkcji która miała być wykonana,
- dane zawierające rezultat wykonanej funkcji,
- informacje o kontroli błędów pozwalające sprawdzić jednostce master, poprawność, odebranych danych.

Jeżeli urządzenie podrzędne odpowiada poprawnie, wtedy dwa pierwsze pola są odbiciem pól wysłanych przez urządzenie nadrzędne. Jeżeli powstanie błąd wtedy urządzenie slave zmienia kod funkcji, w celu powiadomienia o wystąpieniu błędu i w polu danych zwraca opisujące go informacje.



Rys. 4.1 Wymiana komunikatów pomiędzy jednostkami Master-Slave, cykl żądanie-odpowiedź.

Urządzenie z zaimplementowanym protokołem Modbus, zawierają zazwyczaj mapę rejestrów (register map), na której są wykonywane różne operacje w celu monitoringu, konfiguracji albo sterowania. Dzięki takiej konstrukcji każde urządzenie jest widziane jako zbiór rejestrów.

4.2 Transmisja szeregową w sieciach Modbus

Transmisja szeregową w sieciach Modbus może odbywać się w dwóch trybach:

- Tryb ASCII
- Tryb RTU

4.2.1 Tryb ASCII

Kiedy urządzenie jest ustawione do komunikacji w trybie ASCII (American Standard Code for Information Interchange), każdy 8-bitowy bajt, wiadomości jest przesyłany jako dwa znaki ASCII. Główną zaletą tego trybu jest to, iż pozwala on na maksymalnie jedno sekundowe odstępy czasowe pomiędzy przesyłanymi znakami, bez generowania błędu.

Format bajtu w trybie ASCII jest następujący:

Kodowanie:	szesnastkowe, jeden znak szesnastkowy zawarty w każdym znaku komunikatu ASCII
Bity w bajcie:	1 bit startu 7 bitów danych 1 bit parzystości 1 bit stopu
Sprawdzanie błędów:	LRC

4.2.2 Tryb RTU

Kiedy urządzenie korzysta z trybu RTU (Remote Terminal Unit), każdy 8-bitowy bajt w komunikacie zawiera dwa 4-bitowe znaki szesnastkowe. Główną zaletą trybu RTU jest lepsza przepustowość w stosunku do trybu ASCII przy tej samej prędkości łącza. Każdy komunikat musi być przesłany ciągłym strumieniem.

Format bajtu w trybie RTU jest następujący:

Kodowanie:	dwa znaki szesnastkowe zawarte w każdym 8-bitowym polu komunikatu
Bity w bajcie:	1 bit startu 8 bitów danych 1 bit parzystości 1 bit stopu
Sprawdzanie błędów:	CRC

W celu zaznaczenia początku oraz końca wiadomości każdy komunikat przesyłany za pośrednictwem protokołu Modbus jest zawarty w tzw. ramce wiadomości (message frame). Dzięki ramce urządzenie jest w stanie określić czy komunikat jest kierowany do niego, oraz czy przesłana wiadomość jest kompletna. Każda ramka komunikatu jest umieszczona przez urządzenie nadające w ramce danych (data frame) do której jest dodawany bit startu, stopu oraz kontroli parzystości. W zależności od trybu transmisji urządzenie nadające wysyła różne ramki.

4.3 Ramki komunikatów

4.3.1 Ramka w trybie ASCII

Komunikat w trybie ASCII rozpoczyna się znakiem dwukropka (:), a kończy znakiem Enter (CRLF). Dozwolone znaki w innych polach ramki są transmitowane w kodzie szesnastkowym. Urządzenia w sieci monitorują wiadomości pod kątem znaku początku i kiedy taki odbiorą dekodują pole adres. Jeżeli pole adres zgadza się z adresem urządzenia wtedy następuje dekodowanie reszty wiadomości. Typowa ramka wiadomości w trybie ASCII przedstawiona jest poniżej.

Start	Adres	Funkcja	dane	Kontrola LRC	Koniec
Znak :	2 znaki	2 znaki	N-znaków	2 znaki	2 znaki (CR i LF)

Rys. 4.2 Pola ramki komunikatu w trybie ASCII.

4.3.2 Ramka w trybie RTU

W trybie RTU komunikat rozpoczyna się „ciszą na łączu” trwającą co najmniej $3,5 \cdot$ (czas trwania pojedynczego znaku). Po upływie tego czasu (oznaczenie na rysunku T1-T2-T3-T4) rozpoczyna się transmisja pola zawierającego adres. Wszystkie dozwolone znaki transmitowane są w kodzie szesnastkowym. Urządzenia sieciowe nasłuchują przesyłane komunikaty (łącznie z odstępami czasowymi). Kiedy odbiorą pole adresu, każde z nich

rozkodowuje informacje w nim zawarte. Jeżeli adres przesłany w komunikacie zgadza się z adresem urządzenia to rozpoczyna się dekodowanie pozostałych pól. Ostatnie pole (cisza na łączu) jest zarazem pierwszym polem następnego komunikatu. Cała wiadomość musi być przesłana ciągłym strumieniem. Jeżeli przed odebraniem kompletnej ramki komunikatu, cisza na łączu będzie dłuższa od $1,5 \cdot$ (czas trwania pojedynczego znaku) urządzenie odbierające czyści bufor danych i spodziewa się, początku następnego komunikatu. Jeżeli cisza na łączu będzie krótsza niż $3,5 \cdot$ (czas trwania pojedynczego znaku), to urządzenie odbiorcze zinterpretuje to jako kontynuację poprzedniej ramki co będzie skutkowało błędem CRC. Na poniższym rysunku przedstawiono typową budowę ramki w trybie RTU.

Start	Adres	Funkcja	dane	Kontrola CRC	Koniec
T1-T2-T3-T4	8 bitów	8 bitów	N x 8 bitów	16 bitów	T1-T2-T3-T4

Rys. 4.3 Pola ramki komunikatu w trybie RTU.

4.3.3 Ramka znaku

W standardzie Modbus, znaki są przesyłane szeregowo (bitowo) od najmłodszego (LSB) do najstarszego bitu (MSB). Ramki pojedynczych znaków z włączoną/wyłączoną opcją parzystości są przedstawione poniżej:

Parzystość włączona

Start	1	2	3	4	5	6	7	PAR	Stop
-------	---	---	---	---	---	---	---	-----	------

Parzystość wyłączona

Start	1	2	3	4	5	6	7	Stop	Stop
-------	---	---	---	---	---	---	---	------	------

Rys. 4.4 Pola ramki pojedynczego znaku, komunikatu w trybie ASCII.

Parzystość włączona

Start	1	2	3	4	5	6	7	8	PAR	Stop
-------	---	---	---	---	---	---	---	---	-----	------

Parzystość wyłączona

Start	1	2	3	4	5	6	7	8	Stop	Stop
-------	---	---	---	---	---	---	---	---	------	------

Rys. 4.5 Pola ramki pojedynczego znaku, komunikatu w trybie RTU.

4.4 Metody sprawdzania błędów

Standard Modbus w sieciach opartych na transmisji szeregowej zapewnia następujące metody sprawdzania błędów transmisji:

- sprawdzanie parzystości (parity checking) – opcjonalnie stosowane do każdego znaku
- sprawdzanie ramki (LRC, CRC) - stosowane do każdej ramki wiadomości

Obie metody są inicjowane przez urządzenie nadrzędne i zastosowane przed rozpoczęciem nadawania transmisji. Urządzenie podrzędne, przed przyjęciem całego komunikatu, sprawdza czy nie występuje błąd.

4.4.1 Sprawdzanie parzystości (parity checking)

Sprawdzanie parzystości, kiedy jest włączone, posiada dwie konfigurowalne opcje:

- parzystość (Even)
- nieparzystość (Odd parity)

Jeżeli parzystość jest włączona, wtedy bit parzystości będzie obliczany dla każdego znaku komunikatu i ustawiony tak aby wszystkie bity były ustawione zgodnie z wcześniej wybraną opcją. Przykład, rozważmy 8 bitowy znak zawarty w ramce znaku RTU

1100 0101

Ramka zawiera cztery jedyne. Jeżeli jest włączona **parzystość** wtedy bit parzystości będzie ustawiony na 0 (suma jedynek jest parzysta). Jeżeli jest włączona opcja **nieparzystość**, wtedy bit parzystości będzie ustawiony na 1 (suma jedynek nieparzysta). Gdy sprawdzanie parzystości jest włączone (opcja: „parity”) to bit parzystości jest dołączany do każdego znaku transmitowanego komunikatu. Urządzenie odbiorcze oblicza ilość jedynek i wykrywa błąd w przypadku gdy ich suma nie zgadza się z bitem parzystości ustawionym przez urządzenie nadawcze.

Jeżeli sprawdzanie parzystości jest wyłączone (no parity) wtedy bit parzystości nie jest obliczany, a zamiast niego w ramce znaku znajduje się dodatkowy bit stopu.

4.4.2 Sprawdzanie LRC (Longitudinal Redundancy Check)

Pole LRC, ramki komunikatu składa się z jednego 8-bitowego bajtu. Wartość LRC jest obliczana przez urządzenie nadawcze. Urządzenie odbiorcze przelicza ponownie LRC, i porównuje wartość otrzymaną w ramce komunikatu z wartością wyliczoną przez siebie. Jeżeli obie wartości różnią się, to oznacza, że wystąpił błąd.

Obliczanie LRC polega na sumowaniu kolejnych 8-bitowych bajtów komunikatu, odrzuceniu przeniesień i na koniec wyznaczeniu uzupełnienia dwójkowego. Sumowanie obejmuje całą wiadomość za wyjątkiem znaczników początku i końca ramki.

4.4.3 Sprawdzanie CRC (Cyclic Redundancy Check)

Pole CRC, ramki komunikatu składa się z 16-bitowego rejestru podzielonego na dwie części. Wartość CRC jest obliczana przez transmitujące urządzenie i dodawana do komunikatu. Urządzenie odbiorcze oblicza ponownie wartość CRC i porównuje, ją do wartości odebranej (obliczonej przez urządzenie nadawcze). Jeżeli obliczone wartości nie są równe wtedy powstaje błąd.

W momencie rozpoczęcia obliczania sumy CRC, rejestr wypełniany jest jedynekami FFFF(hex). Następnie pobierany jest 8-bitowy znak danych komunikatu (bity startu, stopu, parzystości nie biorą udziału w generowaniu CRC) na którym, wraz z dotychczasową

wartością rejestru przeprowadzana jest operacja logiczna XOR. Wynik operacji XOR jest umieszczany w rejestrze CRC, następnie wartość rejestru jest przesuwana o jedną pozycję w stronę najmniej znaczącego bitu (LSB). Miejsce po dotychczasowym najbardziej znaczącym bicie jest wypełniane zerem. Następnym krokiem jest sprawdzenie wartości LSB, jeżeli wynosi 1 wtedy przeprowadzana jest kolejna operacja XOR, wartości przesuniętego rejestru z wcześniej zdefiniowaną stałą, w przeciwnym razie następuje kolejne przesunięcie. Powyższy proces jest powtarzany dopóki nie zostanie przesunięty ostatni bit (8-bitowego) znaku danych. Kiedy obliczenie CRC dla pojedynczego znaku danych zostanie zakończone, następuje pobranie następnego 8-bitowego znaku (dwa znaki w kodzie szesnastkowym), na którym wraz z wcześniej przetworzoną wartością rejestru jest wykonywana operacja XOR, następnie jest wykonywane przesunięcie wartości rejestru itd. Ostateczną wartością CRC jest wynik obliczeń przeprowadzony dla wszystkich pól transmitowanej ramki komunikatu.

4.5 Kody funkcji w protokole Modbus

W zależności od rodzaju implementacji protokołu Modbus w urządzeniu, dostępne są różne funkcje przez niego realizowane. Poniżej przedstawiono najpopularniejsze funkcje implementowane w urządzeniach (kody podane dziesiętnie):

Tabela 1 Przykładowe kody funkcji w protokole Modbus.

Kod funkcji	Nazwa
1	odczyt wyjścia dwustanowego
2	odczyt wejścia dwustanowego
3	odczyt rejestru wyjściowego
4	odczyt rejestru wejściowego
5	ustawienie pojedynczego wyjścia dwustanowego
6	zapis pojedynczego rejestru
7	odczyt statusu urządzenia slave
8	test diagnostyczny

5 System SCADA

Terminem SCADA (Supervisory Control And Data Acquisition) z reguły określamy przemysłowy komputerowy system sterowania i monitorowania procesu, który swoim zasięgiem obejmuje:

- procesy wytwórcze o charakterze ciągłym, dyskretnym, programowym lub też wsadowym.
- procesy transportowe, np. energii elektrycznej, gazu, ropy, wody
- procesy monitoringu i obsługi budynków, np. sterowanie oświetleniem, klimatyzacją, monitorowanie poboru energii elektrycznej.

System SCADA z reguły zawiera poniższe podsystemy:

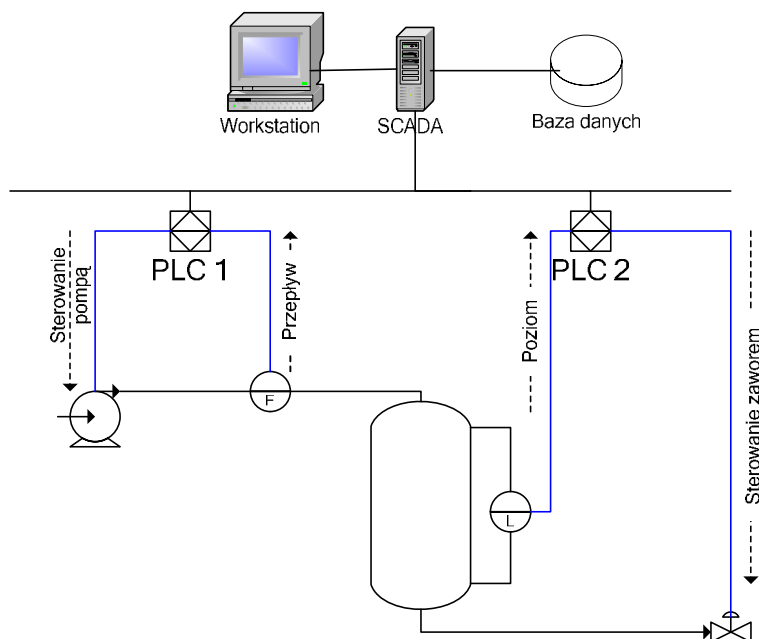
- HMI (Human-Machine Interface) interfejs człowiek-maszyna, przedstawia dane procesowe operatorowi, który na ich podstawie jest w stanie monitorować oraz kontrolować proces.
- System nadrzędny, który gromadzi dane procesowe oraz steruje procesem.
- RTU (Remote Terminal Unit) interfejs pomiędzy sprzętem fizycznym a systemem nadrzędnym, jego zadaniem jest konwersja danych obiektowych na dane cyfrowe i przesłanie ich do systemu nadrzędnego, lub odwrotnie.
- Komunikacyjny, który zapewnia łączność pomiędzy RTU a systemem nadrzędnym.

5.1 *Koncepcja systemu SCADA*

Jednym z najważniejszych celów automatyzacji jest lepsze wykorzystanie zasobów przedsiębiorstwa. Tradycyjne obiekty były monitorowane i sterowane za pomocą tablic synoptycznych w sterowniach, system SCADA może udoskonalić i zastąpić wiele z tradycyjnych funkcji, w nich realizowanych.

SCADA jest przeważnie centralnym systemem zapewniającym monitoring oraz sterowanie procesem w całym przedsiębiorstwie produkcyjnym, albo kompleksowym

systemem rozproszonym na dużej przestrzeni np. na obszarze miasta. Większość działań sterujących jest przygotowywana automatycznie przez RTU bądź też sterowniki programowalne (PLC). Funkcje sterujące systemu SCADA są zazwyczaj ograniczone do podstawowych przesterowań elementów wykonawczych albo odpowiednich reakcji na sygnał alarmowy (np. otwieranie zaworu w przypadku wysokiej temperatury). Na rysunku 5.1 przedstawiono przykład. System SCADA odczytuje wartości poziomu i przepływu oraz wysyła odpowiednie wartości punktów pracy do PLC. PLC1 porównuje wartość przepływu z wartością zadaną i steruje obrotami pompy tak aby wartość przepływu odpowiadała zadanemu punktowi pracy.



Rys. 5.1 Przykład systemu SCADA.

Gromadzenie danych procesowych rozpoczyna się na poziomie RTU albo PLC. Kolekcjonowane dane mogą zawierać:

1. wartości wielkości mierzonych,
2. informacje o stanie urządzenia,
3. informacje o komunikacji ze SCADA.

Dane są obrabiane i formatowane na jednostki fizyczne, których w sterowni używa operator, tak aby na ich podstawie można było podjąć decyzje odnośnie dostrojenia procesu. Dane mogą być również przesłane do zewnętrznych baz danych w celu archiwizacji i analizy przez inne aplikacje.

W systemie SCADA przeważnie jest zaimplementowana procesowa baza danych, której poszczególne rekordy są nazywane znacznikami (tag's), punktami (point's) bądź blokami. Blok reprezentuje pojedyncze wejście albo wyjście monitorowanej wartości albo sterowanego urządzenia. Bloki mogą dotyczyć sprzętu bądź operacji matematycznej. Blok sprzętu reprezentuje aktualne wejście albo wyjście urządzenia, natomiast blok operacji matematycznej przedstawia aktualną wartość wyliczoną na podstawie innych bloków. Wartości bloków są z reguły przechowywane parami: wartość-czas, seria takich par jest historią danego bloku.

5.2 Human-Machine Interface (HMI)

HMI jest mechanizmem, który prezentuje dane procesowe operatorowi i przez, który operator kontroluje proces. HMI przeważnie posiada interfejs do baz danych systemu SCADA i programów dostarczających wykresy, dane diagnostyczne oraz informacje służące do zarządzania takie jak: harmonogramy, procedury, receptury, schematy. Informacja dostarczana operatorowi jest zazwyczaj w postaci graficznej za pomocą schematów, reprezentujących uproszczony model fizycznego obiektu, np. rysunek pompy połączonej z rurą może pokazywać operatorowi czy pompa pracuje i ile cieczy w danej chwili jest pompowanej przez rury. Kiedy operator wyłączy pompę, HMI w czasie rzeczywistym pokazuje zmniejszony przepływ cieczy. Schematy mogą zawierać:

1. elementy geometryczne takie jak: linie, prostokąty,
2. schematyczne symbole reprezentujące aparaturę obiektową,
3. inne elementy graficzne np. animowane zdjęcia aparatury obiektowej.

HMI dla systemów SCADA zawiera program pozwalający na interakcję operatora z obiektami graficznymi. Dzięki temu programowi operator klikając np. na symbol pompy włącza lub wyłącza urządzenie.

Ważną częścią każdego systemu SCADA są alarmy, które przedstawiają logiczny stan pola alarmowego danego bloku bazy danych, wartość 0 reprezentuje status NORMAL, 1-ALARM. Alarmy są skonstruowane w taki sposób, aby włączały się wtedy, gdy spełnione będą określone warunki np. przekroczenie wcześniej zdefiniowanych limitów. Poprzez alarmy, udostępniana jest informacja o tej części systemu, która w danym momencie wymaga uwagi. Informacja o stanie alarmowym pojawia się dopóki stan pola alarmowego danego bloku nie przyjmie wartości NORMAL.

5.3 Remote Terminal Unit (RTU)

RTU jest mikroprocesorem, sterującym elektroniczne urządzenie, które łączy fizyczny sprzęt z systemem SCADA. Jego zadaniem jest konwertowanie sygnałów elektrycznych pochodzących od urządzeń obiektowych, na postać cyfrową. Konwertowane sygnały mogą mieć postać binarną, wyrażającą stany otwarty/zamknięty np. przełączników, lub zmiennoprzecinkową wyrażającą pomiary takie jak ciśnienie, przepływ, napięcie, natężenie prądu. Konwersja sygnałów odbywa się od i do urządzenia, dzięki temu poprzez RTU możemy np. otwierać/zamykać zawory lub ustawiać prędkość pompy.

RTU może zostać wyposażony w różne standardy komunikacji:

- RS232
- RS485
- RS422
- Ethernet
- Radio
- inne

Oraz może wspierać standardowe protokoły komunikacji:

- Modbus
- IEC 60870-5-101/103/104

- DNP3
- ICCP
- Inne

Dzięki powyższym cechom, można go zaimplementować w różnym oprogramowaniu. RTU może monitorować wejścia analogowe o sygnałach prądowych i napięciowych takich jak 4 - 20mA, 0 - 10V, -2,5 - 2,5V, następnie przed przedstawieniem danych operatorowi poprzez interfejs HMI, konwertować je na odpowiednie jednostki lub pozostawić tą rolę systemowi SCADA.

6 Proficy HMI/SCADA iFix

Do najpopularniejszych rozwiązań na rynku oprogramowania HMI/SCADA należą:

1. InTouch firmy Wonderware - 21,2 % udziału w rynku światowym
2. iFix firmy Ge Fanuc - 20,8% udziału w rynku światowym
3. WinCC firmy Siemens - 16,5% udziału w rynku światowym*

* dane za 2005 rok, „Rynek oprogramowania HMI/SCADA w Polsce i na świecie” - xtech.pl

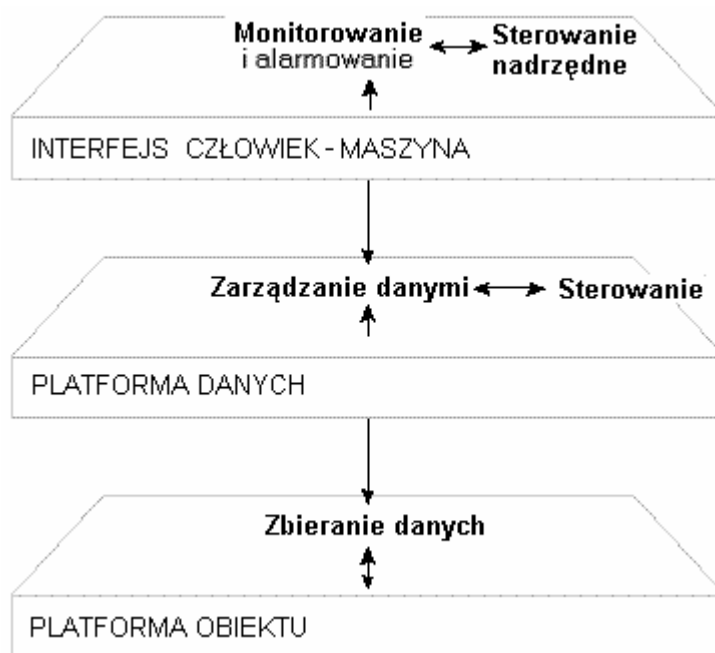
Wspólnym mianownikiem dla powyższego oprogramowania jest łatwość tworzenia aplikacji oraz potężne możliwości. Jednak jedynie iFix wyróżnia się na tle konkurencji następującymi cechami:

1. Możliwość sterowania procesem ciągłym i dyskretnym
2. Pełna integracja z Visual Basic for Application
3. Bloki realizujące specjalne funkcje min. sterowanie statystyczne, analiza Pareto, obliczenia statystyczne, histogram, liniowy przyrost sygnału.

6.1 Podstawowe funkcje HMI/SCADA

System iFix może udoskonalić i zastąpić wiele z tradycyjnych funkcji realizowanych w sterowaniach, a w szczególności:

- Monitorowanie
- Sterowanie nadrzędne
- Alarmowanie
- Sterowanie



Rys. 6.1 Funkcje systemu iFIX.

Monitorowanie jest zdolnością do gromadzenia i wyświetlania zainteresowanemu personelowi związanych z obiektem danych w czasie rzeczywistym. Dostępne są różnorodne numeryczne, tekstowe i graficzne formaty wyświetlania danych, w celu uzyskania lepszej czytelności, łatwiejszego dostępu do danych i ich zrozumienia.

Sterowanie nadrzędne jest zdolnością do monitorowania danych w czasie rzeczywistym połączoną z możliwością zmiany przez operatora wartości zadanych, jak również innych kluczowych parametrów procesu, bezpośrednio za pomocą komputera.

Dzięki systemowi iFix można łatwo określić kto ma prawo dostępu do tych danych i jakie dane mogą być zmieniane przez poszczególnych użytkowników systemu.

Alarmowanie jest zdolnością do rozpoznawania zdarzeń wyjątkowych w procesie użytkownika i do natychmiastowego raportowania tych zdarzeń odpowiedniemu personelowi. Alarmy są generowane przy przekroczeniu limitów (ograniczeń) ustawianych przez użytkownika i mogą być raportowane na wiele sposobów

Sterowanie jest zdolnością do automatycznego wykorzystania algorytmów w celu stabilizacji zmiennych procesowych, a więc do utrzymywania wartości tych zmiennych w zadanych

limitach (granicach). Sterowanie jest posunięte o krok do przodu w stosunku do sterowania nadrzędnego poprzez usunięcie konieczności bieżącego współdziałania z człowiekiem.

System iFix oferuje mechanizmy do sterowania ciągłego, sterowania procesami wsadowymi i sterowania statystycznego. Może zostać wykorzystany do sterowania całym procesem, lub też jego częścią.

6.2 Architektura iFix

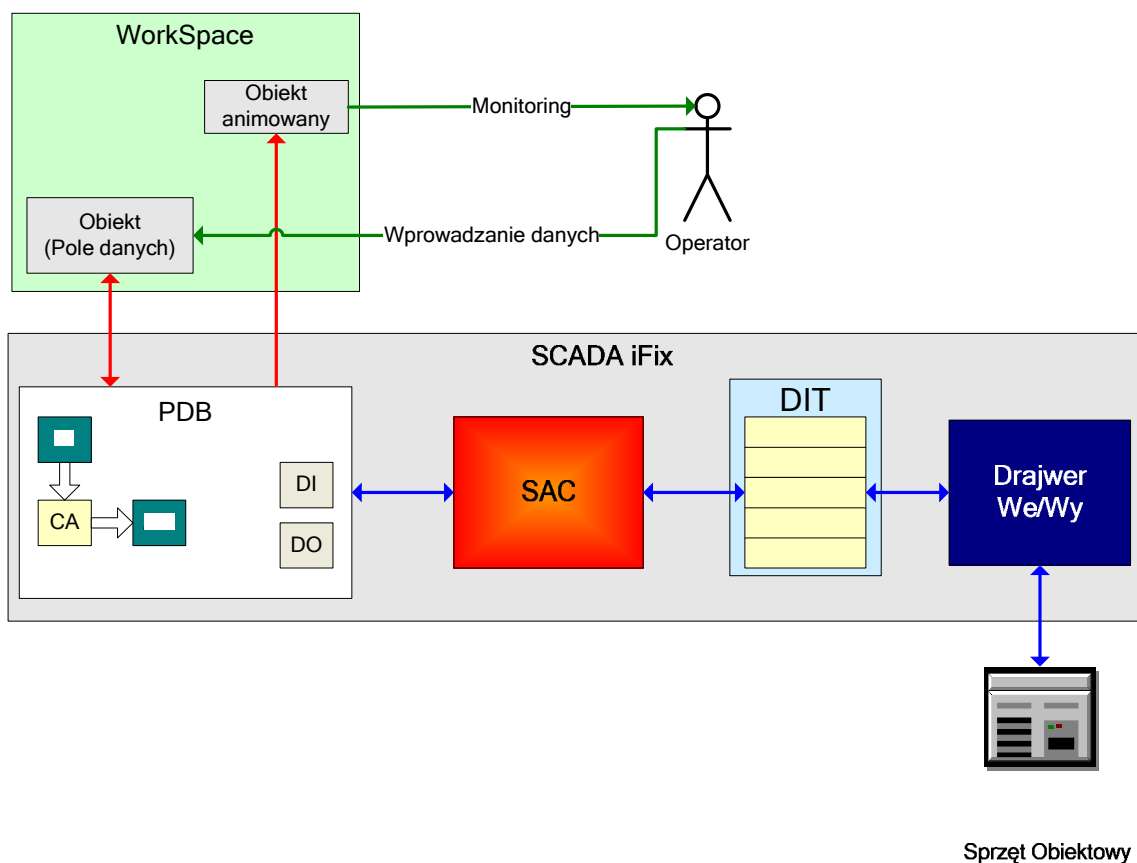
Do poprawnego działania systemu SCADA niezbędna jest odpowiednia komunikacja pomiędzy operatorem, a sprzętem obiektowym. Aby zapewnić wysokiej jakości komunikację, w systemie iFix rozdzielono to zadanie pomiędzy integralne elementy takie jak:

- Drajwer We/Wy
- DIT (Driver Image Table) – tablica obrazu drajwera
- SAC (Scan, Alarm and Control) – program skanowanie, alarmowanie i sterowanie
- PDB (Proces Data Base) – procesowa (sygnałowa) baza danych

Przepływ danych od strony sprzętu do operatora odbywa się w następujący sposób:

- Drajwer We/Wy, komunikuje się bezpośrednio ze sprzętem obiektowym i odpytuje urządzenie w zdefiniowanych czasach (poll times)
- DIT, rezerwuje w pamięci miejsce dla nieprzetworzonych danych i organizuje je w grupy (poll records)
- SAC, odczytuje, przetwarza dane z DIT a następnie skaluje je do jednostek fizycznych (inżynierskich) określonych w indywidualnych blokach PDB. Porównuje wartości poszczególnych bloków z ich zdefiniowanymi polami alarmowymi. Zarządza zapisem danych pochodzących z serwera SCADA do sprzętu obiektowego.
- PDB przechowuje przetworzone dane, reprezentujące bieżący stan procesu, w pamięci komputera oraz udostępnia je innym aplikacjom np. Workspace.

Proces przepływu danych zobrazowano na poniższym rysunku.



Rys. 6.2 Przepływ danych pomiędzy urządzeniem obiektowym a operatorem, w systemie iFIX.

6.3 Technologie wykorzystywane w iFIX

OPC (OLE for Process Control) – Specyfikacja OCP definiuje standardowe typy obiektów, ich zachowania oraz właściwości umożliwiające realizację współpracy w czasie rzeczywistym, aplikacji sterowania procesami przemysłowymi ze sprzętem automatyki. Wymagania te określają: standardowe sposoby dostępu do danych w sprzęcie lub systemach automatyki procesowej, sposoby efektywnego transferu danych pomiędzy sprzętem a aplikacją, możliwości jednoczesnej współpracy oprogramowania klienta z wieloma serwerami danych oraz sposób konfigurowania serwera danych.

OLE (Object Linking and Embedding) – Jest to technologia dołączania i wstawiania obiektów wykorzystywana w celu integracji funkcjonalnej różnych aplikacji i zapewniająca wysoki stopień ich kompatybilności nawet w przypadku różnorodnych typów informacji. Technologia OLE wykorzystuje specyfikację COM i umożliwia tworzenie uniwersalnych,

dających się wielokrotnie wykorzystywać obiektów typu „plug and play”, mogących pracować w różnorodnych aplikacjach. OLE stanowi również podstawę koncepcji tworzenia oprogramowania typu obiektowego, w której poszczególne obiekty mogą być tworzone w dowolnych językach programowania pochodzących od dowolnego dostawcy.

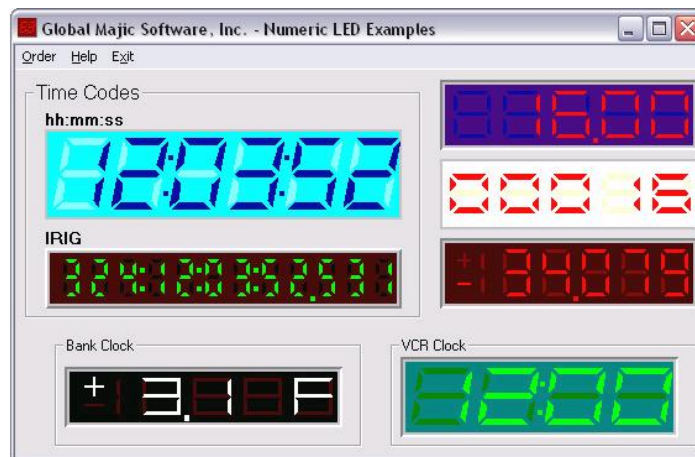
DDE (Dynamic Data Exchange) – Poprzednik technologii OLE. Metoda przekazywania danych pomiędzy aplikacjami na zasadzie wymiany komunikatów wysyłanych między aplikacją pełniącą rolę klienta, a aplikacją serwerową.

COM (Component Object Model) – Model komponentów obiektowych opracowany przez Microsoft. COM jest specyfikacją, określającą postać podstawowego obiektu, niezależną od języka programowania. Standaryzacja między obiektowego interfejsu komunikacyjnego umożliwia aplikacjom korzystanie z funkcji obiektów innych aplikacji (lub systemu operacyjnego) oraz pozwala na modyfikację komponentów bez wpływu na pracę całości systemu.

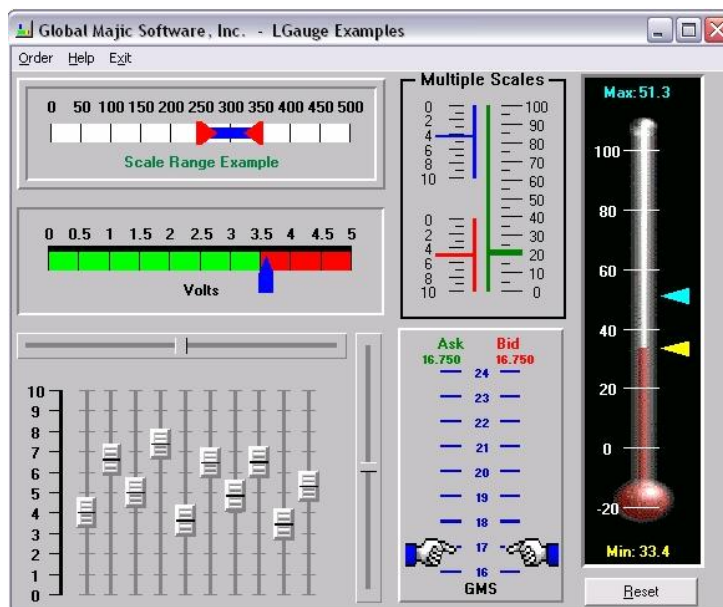
DCOM (Distributed Component Object Model) – Rozszerzenie specyfikacji COM na obiekty zdalne, czyli nie rezydujące w jednej stacji roboczej. DCOM jest zoptymalizowanym protokołem komunikacyjnym, dzięki któremu obiekty oddalone zachowują się jak lokalne. Specyfikacją DCOM została pierwszy raz zastosowana w systemie Windows z rodziny 98 oraz NT 4.0 w 1996 roku.

ActiveX – zestaw technologii programowania stworzonej przez Microsoft, umożliwia tworzenie komponentów w różnych językach, które komunikują się między sobą w jednej workstacji bądź też w środowisku sieciowym. Jest rozwinięciem technologii OLE bazuje na modelu COM, wspiera również DCOM

ActiveX Control (OCX) – kontrolka będąca obiektem wpierana przez OLE, manipulacja przez użytkownika przygotowuje odpowiednie działania wykonywane poprzez VBA albo animacje. W systemie iFIX kontrolki OCX są przechowywane w bezpiecznym kontenerze w celu zabezpieczenia środowiska pracy w przypadku ich niepożądanego działania. Poniżej przykłady kontrolki firmy Global Majic Software



Rys. 6.3 Przykłady kontroltek OCX firmy GMS.



Rys. 6.4 Przykłady kontroltek OCX firmy GMS.

Użycie kontroltek OCX przynosi dla tworzonej aplikacji następujące korzyści:

- większa przejrzystość,
- bardziej intuicyjna obsługa,
- większa atrakcyjność,
- profesjonalny wygląd.

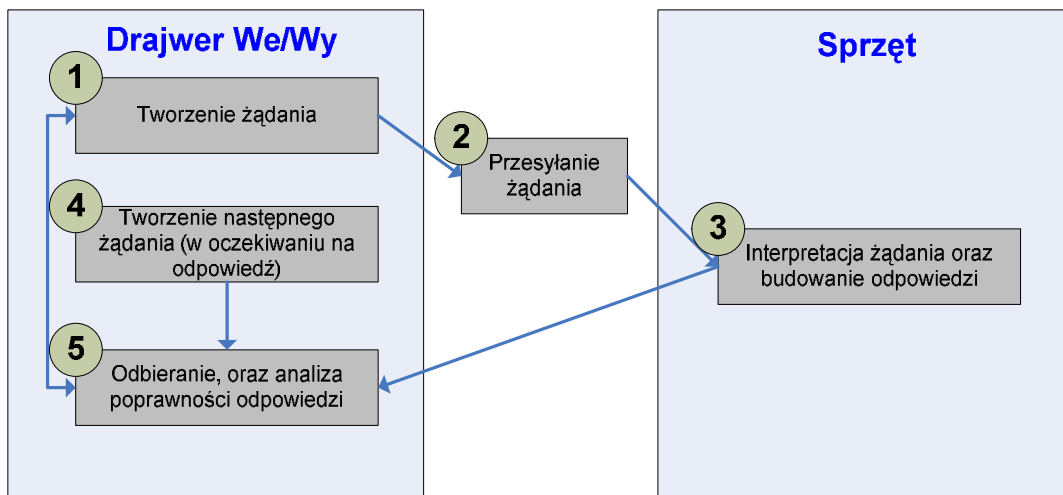
6.4 Drajwer We/Wy 7.x

Zbieranie danych z automatyzowanego procesu w przemyśle jest realizowane przez prawie każde oprogramowanie do automatyzacji. Sterowniki PLC zbierają dane z czujników i sterują elementami wykonawczymi. System iFIX dysponuje interfejsami dla sterowników PLC umożliwiającymi odczytywanie i zapis danych, do odpowiednich rejestrów sterowników PLC. Oprogramowanie to, nazywane jest drajwerem We/Wy [1]. Dalsze wyjaśnienie tak ważnej kwestii jaką jest komunikacja z urządzeniem obiektowym, zostanie przeprowadzone na przykładzie MBI I/O driver.

Drajwer zapewnia interfejs oraz protokół komunikacyjny pomiędzy sprzętem obiektowym zgodnym z Modicon Modbus I, a oprogramowaniem sterującym. W celu zapewnienia elastyczności oraz łatwości użycia został wyposażony w następujące właściwości[7]:

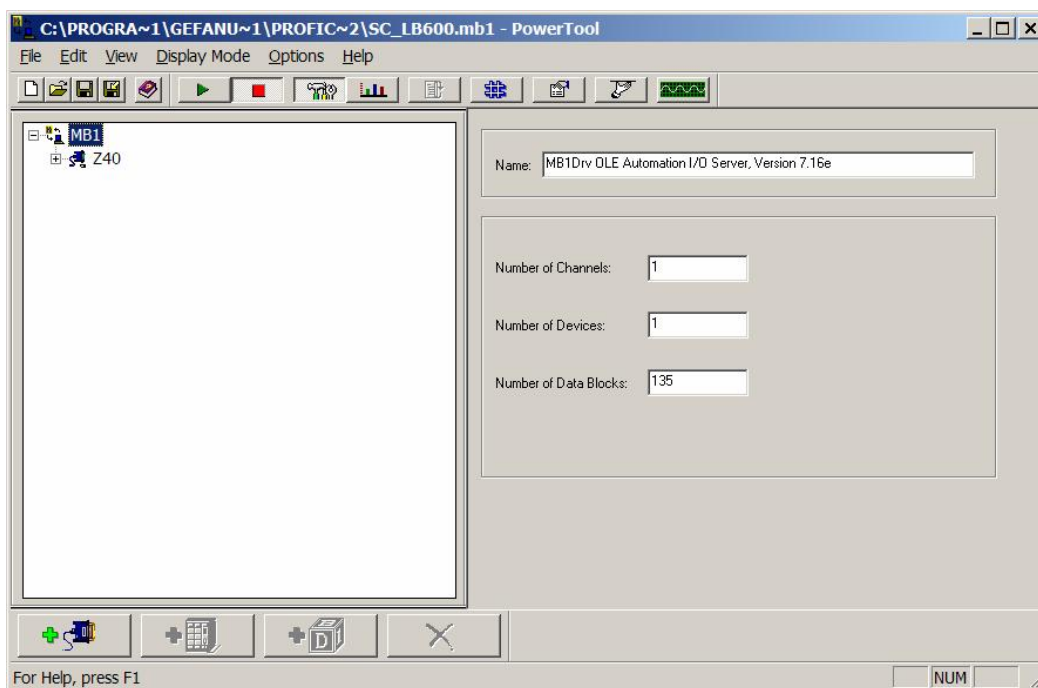
- OLE automation technology,
- integrację z FIX/iFIX,
- sterowanie zdarzeniowe,
- OLE for Process Control.

Komunikacja ze sprzętem obiektowym odbywa się na zasadzie wymiany komunikatów (message) o charakterze żądań i odpowiedzi. Drajwer tworzy żądanie (query), które jest przesyłane do urządzenia, następnie urządzenie po przetworzeniu żądania wysyła odpowiedź (response). W czasie oczekiwania na odpowiedź drajwer przygotowuje następne żądanie. Odebrana od urządzenia odpowiedź jest sprawdzana pod kątem poprawności (porównanie danych zawartych w żądaniu i odpowiedzi), a następnie udostępniania dla innych aplikacji. Jeżeli odpowiedź zawiera błąd, drajwer ponownie wysyła żądanie.



Rys. 6.5 Mechanizm budowania komunikatu.

Do konfigurowania, oraz diagnostyki drajwera służy program MB1 Power Toll.



Rys. 6.6 Okno konfiguracyjne drajwera MB1.

Power Tool umożliwia następujące czynności:

- dodanie kanału,
- dodanie urządzenia,
- dodanie bloków danych,
- sprawdzenie poprawności działania poprzez śledzenie statystyk.

Aby konfiguracja drajwera była poprawna, musi on zawierać trzy podstawowe rzeczy:

1. przynajmniej jeden kanał,
2. przynajmniej jedno urządzenie na każdy kanał,
3. przynajmniej jeden blok danych na każde urządzenie.

Na poniższych rysunkach przedstawiono konfiguracje drajwera oraz wyjaśniono ważniejsze pola konfiguracyjne na poszczególnych etapach konfiguracji.

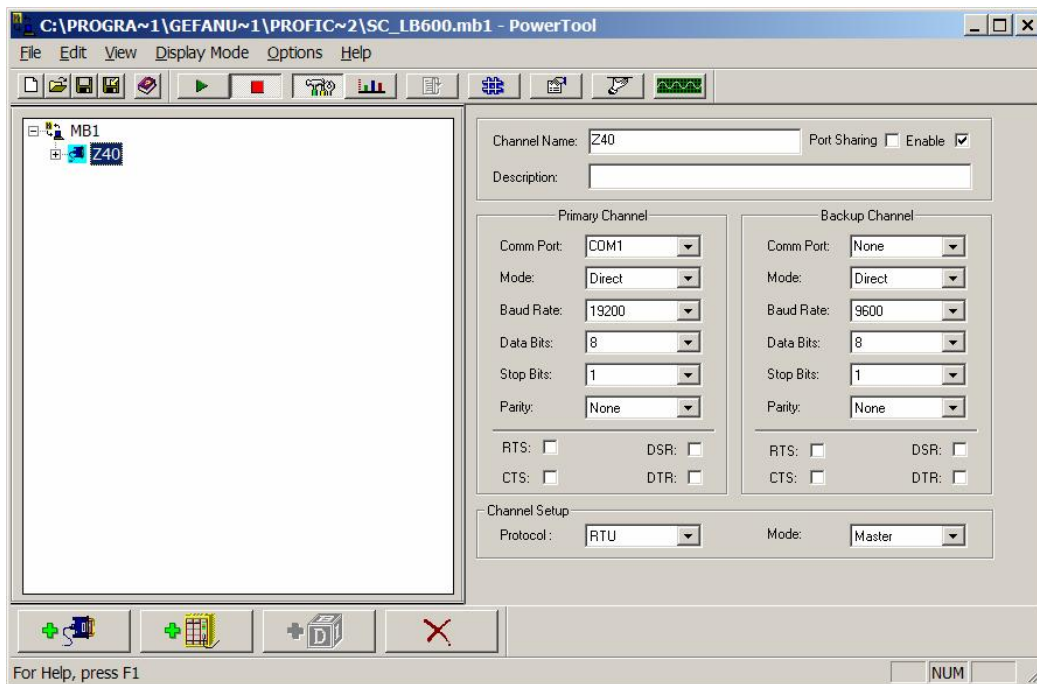
W pierwszej kolejności należy utworzyć kanał (Channel), poprzez wybranie w dolnej części programu przycisku po lewej stronie. Kanał odpowiada fizycznemu portowi COM, jeżeli port jest współdzielony z inną aplikacją należy zaznaczyć opcję „*Port Sharing*” . W dalszej części należy uzupełnić pola w sekcji „*Primary Chanel*” , która odpowiada za komunikację. W polu „*Comm Port*” należy określić do jakiego fizycznego portu COM w komputerze jest podłączone urządzenie, z którym chcemy się komunikować. Pole „*Mode*” określa medium służące do komunikacji, mamy do wyboru następujące opcje:

- Direct – bezpośrednie połączenie kablowe,
- Radio – połączenie za pomocą transmisji radiowej,
- Telephone – połączenie poprzez modem kablowy przy pomocy Microsoft Windows’s Telephony API (TAPI). TAPI pozwala drajwerowi kontrolować funkcje modemu takie jak inicjalizacja połączenia, udzielanie odpowiedzi, przerywanie połączenia.

Pole „*Baud Rate*” określa ilość zmian sygnału w czasie 1 sekundy, wyrażonych w bodach, jeżeli nie zastosowano modulacji sygnału to prędkość w bodach jest równoważna prędkości w bitach na sekundę. Prędkość transmisji drajwera musi odpowiadać, prędkości transmisji urządzenia, w przeciwnym razie komunikacja nie zostanie nawiązana. Podobna sprawa dotyczy reszty pól w tej sekcji, dlatego też przed konfiguracją pozostałych, zaleca się weryfikację możliwości komunikacyjnych urządzenia. Jeżeli planujemy zapewnić awaryjne połączenie należy w podobny sposób skonfigurować pola znajdujące się w sekcji „*Backup Channel*”.

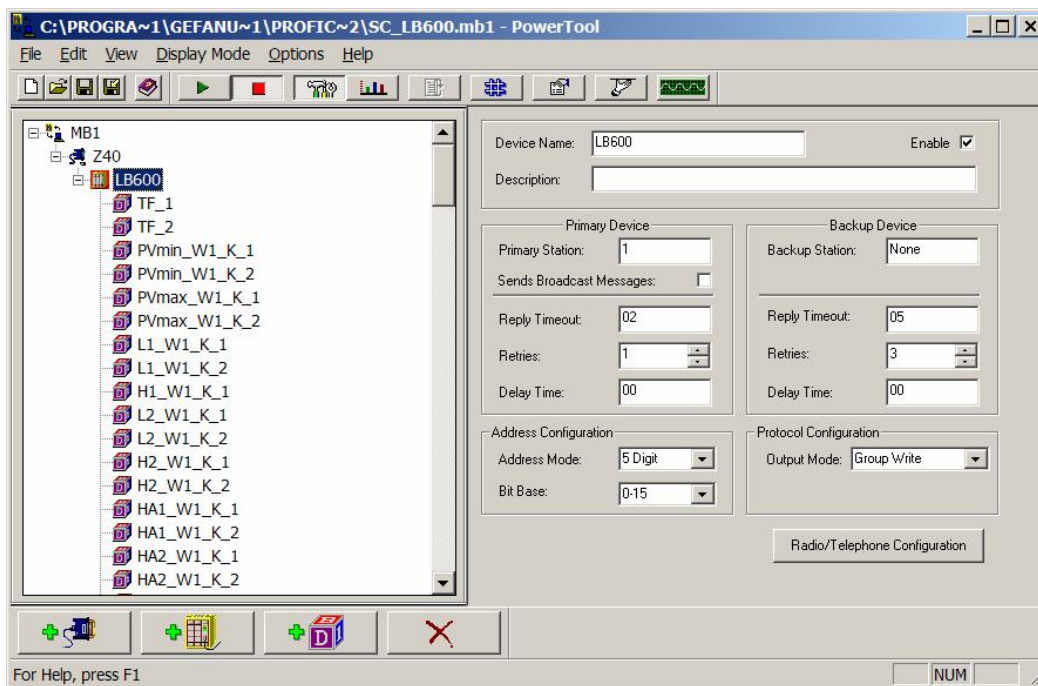
W sekcji „*Channel Setup*” należy wybrać protokół RTU (szybszy) bądź ASCII (tekstowy - wolniejszy) w polu „*Mode*” należy określić czy konfigurowany drajwer ma

pracować w trybie nadrzędnym (Master) czy też podrzędnym (Slave). Na koniec konfiguracji należy ustawić znacznik przy opcji „Enable”.



Rys. 6.7 Okno konfiguracji kanału.

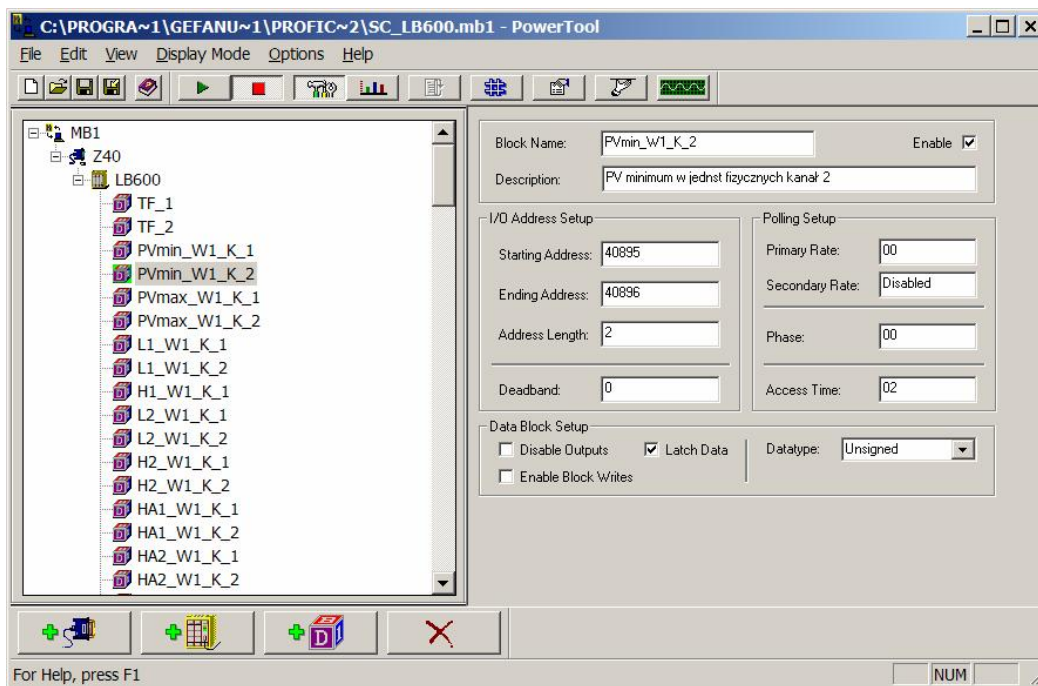
Następnym krokiem jest utworzenie i konfiguracja urządzenia, która jest analogiczna do wyżej wymienionych czynności. Pola „Primary Station”, „Adress Mode”, „bit base” oraz „Output Mode”, powinny być skonfigurowane zgodnie z dokumentacją urządzenia. Pole „Reply Timeout”, określa po jakim czasie, w przypadku braku odpowiedzi od urządzenia, drajwer ma ponowić transmisję bloków danych. Pole „Retries”, określa ile razy drajwer będzie powtarzał transmisję bloków danych, a w polu „Delay Time” należy podać czas po jakim drajwer ma ponowić transmisję w przypadku, gdyby urządzenie nie odpowiedziało na ostatnią powtórzoną transmisję. Po skonfigurowaniu wymaganych pól należy zaznaczyć opcję „Enable”.



Rys. 6.8 Okno konfiguracji urządzenia.

Ostatnim i najbardziej pracochłonnym krokiem jest utworzenie bloków danych, odwołujących się do pożądaných rejestrów konfigurowanego urządzenia. Po wprowadzeniu nazwy, należy uzupełnić sekcję „*I/O Address Setup*”, gdzie określamy adres początkowy oraz jego długość, pole adres końcowy zostanie automatycznie uzupełnione. Pole „*DeadBand*” oznacza strefę martwą. W sekcji „*Polling Setup*”, pola określają konfigurację odpytywania urządzenia przez drajwer, znaczenie pól jest następujące:

- **Primary Rate** – okres odpytywania bloku danych, wartość „disable” oznacza brak odpytywania, wartość „0” wymusza największą wydajność,
- **Secondary Rate** – okres odpytywania bloku danych, po czasie „*Access Time*”, wartość „disabled” oznacza, że odpytywanie jest wyłączone,
- **Phase** – przesunięcie w czasie odczytu bloku,
- **Access Time** – czas odpytywania bloku z okresem (prędkością) określoną w polu „*Primary Rate*”.



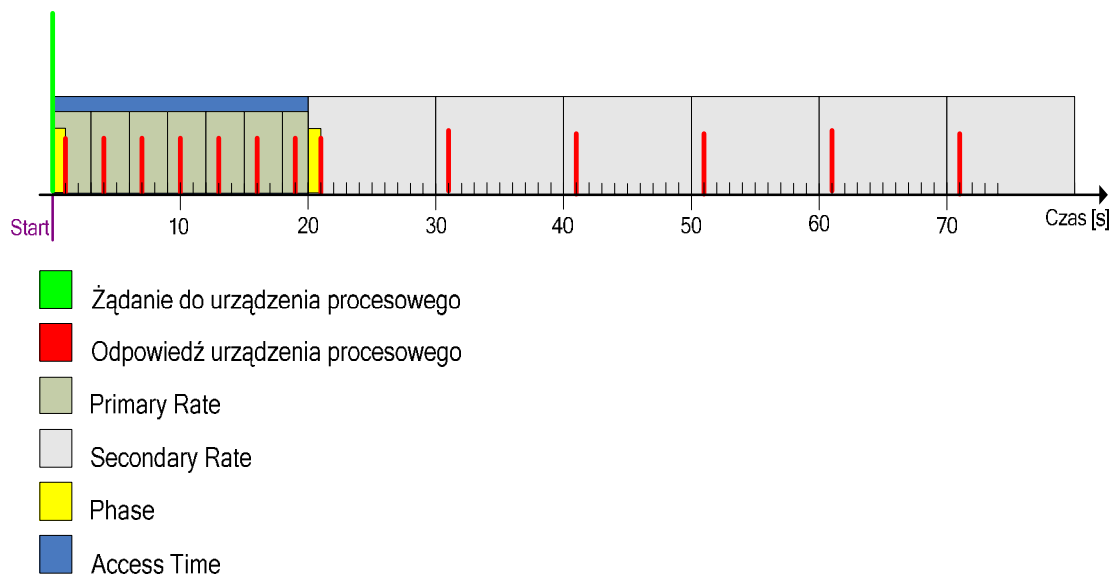
Rys. 6.9 Okno konfiguracji bloku danych.

W sekcji „Data Block Setup”, należy według potrzeby skonfigurować następujące opcje:

- **Disable Outputs** – ustawienie wyłącza możliwość zapisu danych,
- **Enable Block Writes** – włączenie zapisu blokowego (wszystkie dane z danego bloku danych są uaktualniane podczas zapisu nawet tylko jednej wartości),
- **Latch Data** – przekazywanie aplikacji ostatnich poprawnych danych w przypadku utraty komunikacji z urządzeniem,
- **Datatype** – typ danych, konfigurowanego bloku, poniżej wymieniono podstawowe typy danych:
 - *Unsigned* – 16-bitowa liczba całkowita dodatnia,
 - *Signed* – 16-bitowa liczba całkowita, mogąca przyjmować wartości dodatnie i ujemne,
 - *Float* – 32-bitowa liczba zmiennoprzecinkowa, mogąca przyjmować wartości dodatnie i ujemne, jest przechowywana w dwóch kolejnych rejestrach 16-bitowych,
 - *Digital* – wartość pojedynczego bitu.

Aby uaktywnić blok należy zaznaczyć opcję „Enabled”.

Poprawna konfiguracja odpytywania jest dość ważnym i trudnym zagadnieniem, ponieważ od niej zależy jak szybko, na ekranach synoptycznych będą odświeżały się dane. Definicję ustawień odpytywania, w celu lepszego zrozumienia, przedstawiono na poniższym rysunku.



Rys. 6.10 Parametry odpytywania urządzenia przez drajwer MB1.

W tym miejscu warto również wspomnieć, iż jeżeli pojawi się ponowne żądanie to drajwer zeruje czas określony w polu „Access Time” powodując tym samym, że jest on liczony od nowa. Niepoprawna konfiguracja czasów odpytywania może doprowadzić do sytuacji, że na ekranach synoptycznych będziemy widzieli nieaktualne dane. W ekstremalnych przypadkach taki stan rzeczy może trwać nawet kilka minut!

6.5 Procesowa baza danych

„Sercem” systemu iFix jest procesowa baza danych (PDB), która zawiera rekordy nazywane blokami. Bloki mogą reprezentować wejścia lub wyjścia do urządzeń obiektowych oraz operacje matematyczne. Do zarządzania blokami służy aplikacja Database Manager. Jest to prosty program, który umożliwia wykonywanie na blokach bazy, następujących operacji:

- dodawanie nowych bloków,
- modyfikowanie,
- usuwanie,
- generowanie grupami,
- podglądanie tylko do odczytu.

	Nazwa bloku	Typ	Opis	Następny blok	Dolny	Górny	Adr We/Wy	Urz	Wartość bież.
1	AI_K1	AI	Poziom (skala znormalizowa	AI1_S	0,000	1,000	LB600:40878	MB1	0,000
2	AI_K1_PID	AI	wartości do bloku PID	PID_KOMP	0,000	1,000	LB600:40878	MB1	0,000
3	AI_K2	AI	AI w skali znormalizowanej	AI2_S	0,000	1,000	LB600:40918	MB1	0,000
4	SP_K1	AI	SP 1 (wartości historyczne)	WARTOŚĆ_ZAD	0,0000	100,0000	LB600:44929	MB1	0,0000
5	SP_K2	AI	SP 2 (wartości historyczne)	WARTOŚĆ_ZAD	0,0000	100,0000	LB600:45029	MB1	0,0000
6	YS_K1	AI	CV z AO1 (warstwa 9) skala 0	CV	0	1	LB600:46012	MB1	0
7	ZBIORNIK2_POZIOM	AI	zbiornik połączony z SIM		0,00	100,00	RB	SIM	10,70
8	W8_WSP_A	AO	wartość CV z bloku PID KOM		0,00	1,00	LB600:45809	MB1	1,00
9	AH_REG1	AR	alarm regulatora od odchyłki		0	100	LB600:44937	MB1	????
10	AH_REG2	AR	alarm regulatora od odchyłki		0	100	LB600:45037	MB1	????
11	AKCEPATACJA_NASTAW	AR	akceptacja po zakończeniu s		0	1	LB600:44979	MB1	????
12	AKTW_BLOKU_AI1	AR	aktywnosc bloku AI1		0	525	LB600:40877	MB1	????
13	AKTW_BLOKU_AI2	AR	aktywnosc bloku AI1		0	525	LB600:40917	MB1	????
14	AKTW_BLOKU_AO1	AR	Aktywnosc bloku AO1		0	525	LB600:46011	MB1	????
15	AKTW_BLOKU_W8	AR	aktywnosc bloku w warstwie		0	525	LB600:45817	MB1	????
16	AKTW_DI1	AR	samostrojzenie		0	525	LB600:42684	MB1	????
17	AKTW_DI2	AR	samostrojzenie		0	525	LB600:42694	MB1	????

Rys. 6.11 Widok programu Database Manager.

W poniższej tabeli przedstawiono wszystkie bloki, wraz z opisami, wspierane przez Database Manager.

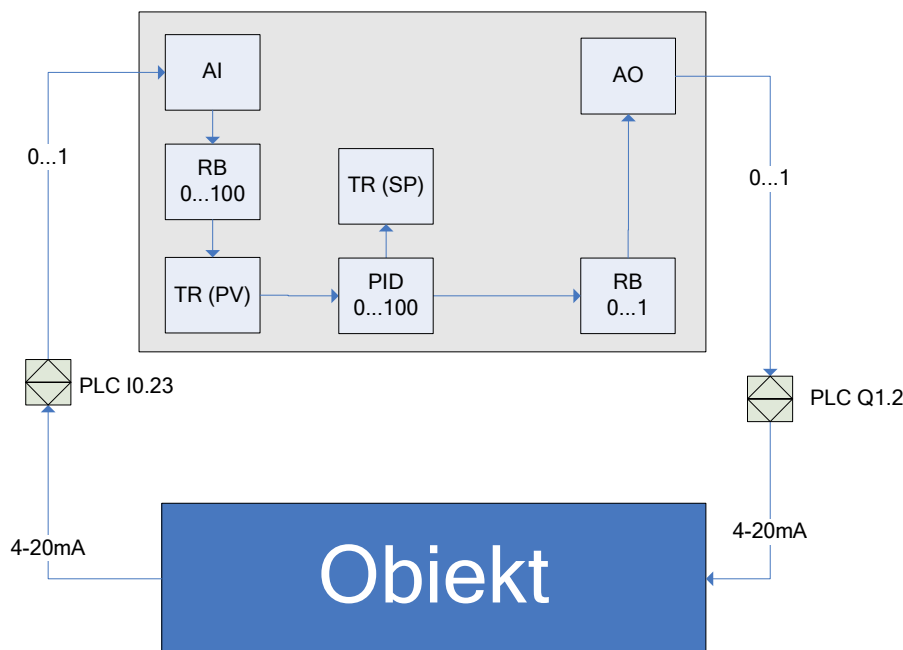
Tabela 2 Zestawienie funkcji wszystkich bloków dostępnych w procesowej bazie danych systemu iFIX.

Blok:	Funkcja:	Rodzaj bloku
Blok opóźniający (DT)	Opóźnia do 255 sekund transfer wartości wejściowej do następnego bloku w łańcuchu. Może przechowywać do 60 wartości i wysyła wartości w oparciu o zasadę pierwszy na wejściu pierwszy na wyjściu.	Sterujący
Blok przesunięcia fazowego (LL)	Umożliwia symulowanie dynamiki procesu i zawiera dyskretną aproksymację równań eksponentjalnych dla bloku przesunięcia fazowego. Blok ten jest użyteczny w strategiach ze sprzężeniem wprzód.	Sterujący
PID (PID)	Porównuje analogowe wejścia ze zdefiniowaną przez użytkownika wartościąadaną i wysyła inkrementowane sygnały sterujące, by zbliżyć wartość zmiennej procesowej do wartości zadanej.	Sterujący
Blok sterowania dwustanowego (BB)	Otrzymuje wartości analogowe i wysyła wartości dwustanowe.	Sterujący

Blok typu RAMP (RM)	Zwiększa lub zmniejsza wartości do wartości docelowej z określoną szybkością. Wartości docelowe mogą być wprowadzane przez użytkownika lub pobierane z innych bloków. Można zdefiniować trzy fazy dla procesu narastania.	Sterujący
Blok skalowania (RB)	Umożliwia zmianę sygnału wejściowego przez dodanie wartości stałej lub przez pomnożenie go przez wartość stałą po odjęciu offsetu od sygnału. Blok ten rezerwuje mniejszą ilość pamięci i wykonuje się szybciej niż blok obliczeniowy.	Sterujący
Histogram (HS)	Wyświetla częstotliwość występowania wartości w formie wykresu słupkowego.	statystyczne sterowanie procesem
Pareto (PA)	Akceptuje, oblicza i sortuje częstotliwość występowania maksymalnie do ośmiu wartości. Wartości te mogą być wyświetlone na rysunkach w postaci wykresu słupkowego.	statystyczne sterowanie procesem
Sterowanie statystyczne (SC)	Steruje on zmienną procesową na podstawie obliczania offsetu średniego i szybkości zmian odchylenia od wartości zadanej, XBARBAR. Blok ten jest uaktywniany, gdy blok Dane statystyczne generuje alarm.	statystyczne sterowanie procesem
Dane statystyczne (SD)	Śledzi dane wprowadzane przez operatora lub inne bloki i wykonuje obliczenia statystyczne. Blok ten generuje alarmy na podstawie standardowej techniki Statystycznego sterowania procesem.	statystyczne sterowanie procesem
Blok sterowania urządzeniami (DC)	Koordynuje otwieranie i zamykanie styków urządzeń dwustanowych w halach produkcyjnych na podstawie zdefiniowanych przez użytkownika warunków.	Wsadowy
Blok programowy (PG)	Wykonuje krótkie programy w celu wykonania operacji wsadowych lub zwiększenia stopnia automatyzacji w aplikacji.	Wsadowy
Dane SQL	Określa dane, które należy przemieścić z bazy danych procesu do relacyjnej bazy danych lub na odwrót z relacyjnej bazy danych do bazy danych procesu.	SQL
Inicjator operacji SQL	Inicjuje wykonanie instrukcji SQL oraz definiuje sposób komunikowania się relacyjnej bazy danych z bazą danych procesu.	SQL
Alarm analogowy (AA)	Umożliwia odczyt/zapis danych analogowych oraz generowanie i potwierdzanie alarmów.	Pierwotny
Wejście analogowe (AI)	Umożliwia odczyt/zapis danych analogowych oraz ustawienie progów alarmowych.	Pierwotny
Wyjście analogowe (AO)	Wysyła wartość analogową do drajwera We/Wy lub serwera OPC, gdy poprzedni blok lub gdy operator, blok programowy lub program Łatwy dostęp do bazy danych (ang. Easy Database Access (EDA)) ustawia wartość tego bloku.	Pierwotny
Rejestr analogowy (AR)	Umożliwia odczyt/zapis danej analogowej tylko wtedy, gdy połączenie do tego bloku wyświetla daną na rysunku.	Pierwotny
Blok Boolowski (logiczny) (BL)	Wykonuje obliczenia logiczne na podstawie maksymalnie ośmiu wejść.	Pierwotny
Alarm dwustanowy (DA)	Umożliwia odczyt/zapis danych dwustanowych oraz generowanie i potwierdzanie alarmów.	Pierwotny
Wejście dwustanowe (DI)	Umożliwia odczyt/zapis danych dwustanowych oraz ustawienie progów alarmowych.	Pierwotny
Wyjście dwustanowe (DO)	Wysyła wartość dwustanową do drajwera We/Wy lub serwera OPC, gdy poprzedni blok lub gdy operator, blok programowy, skrypt lub program Łatwy dostęp do bazy danych (ang. Easy Database Access (EDA)) ustawia wartość tego bloku.	Pierwotny
Rejestr dwustanowy (DR)	Umożliwia odczyt/zapis danych dwustanowych tylko wtedy, gdy połączenie do tego bloku wyświetla daną na rysunku.	Pierwotny

Wielokrotne wejścia dwustanowe (MDI)	Pobiera dane dwustanowe - do trzech wejść z drajwera We/Wy lub serwera OPC, wylicza wartość na podstawie wejść i umożliwia ustawienie progów alarmowych.	Pierwotny
Blok tekstowy (TX)	Umożliwia odczyt/zapis informacji tekstowej z/do urządzenia zewnętrznego.	Pierwotny
Blok obliczeniowy (CA)	Wykonuje obliczenia matematyczne stosując wartości z poprzedzającego bloku i maksymalnie do siedmiu innych stałych lub wartości bloków.	Wtórny
Blok reakcji na zdarzenie (EV)	Monitoruje wartości lub warunki alarmowe poprzedzającego bloku i wykonuje operacje oparte na wartościach wyjściowych bloku poprzedzającego.	Wtórny
Rozszerzony bufor wykresu (ETR)	Pozwala zebrać maksymalnie do 600 bieżących wartości z bloku poprzedzającego. Użytkownik może wyświetlić te wartości za pomocą obiektu Wykres na rysunkach.	Wtórny
Blok powielający (FN)	Przesyła wartość otrzymaną z bloku poprzedzającego do maksymalnie czterech dodatkowych bloków.	Wtórny
Selektor (SS)	Próbkuje maksymalnie do sześciu wejść, wykonując operacje na sygnałach wejściowych zgodnie z wybranym trybem przez użytkownika i wysyła wartość do następnego bloku.	Wtórny
Blok typu Timer (TM)	Działa jak licznik czasu, zwiększając lub zmniejszając swoją wartość.	Wtórny
Licznik (TT)	Sumuje wartości przesyłane z bloków poprzedzających w postaci wartości zmiennoprzecinkowych. Blok ten przesyła do innych bloków wartość z dokładnością do sześciu cyfr znaczących. W programie WorkSpace może wyświetlać wartości z dokładnością do 15 cyfr znaczących.	Wtórny
Bufor wykresu (TR)	Pozwala zebrać maksymalnie do 80 bieżących wartości z bloku poprzedzającego. Użytkownik może wyświetlić te wartości za pomocą obiektu Wykres na rysunkach.	Wtórny

DataBase Manager umożliwia łączenie ze sobą bloków. Połączone bloki są nazywane łańcuchami. Za pomocą łańcuchów możemy tworzyć odpowiedni przepływ sygnału, tak aby osiągnąć założony cel, np. wykonanie operacji matematycznych na wybranych sygnałach. Poniżej przedstawiono przykład połączenia ze sobą bloków w celu stworzenia układu regulacji z wykorzystaniem bloku sterującego PID.



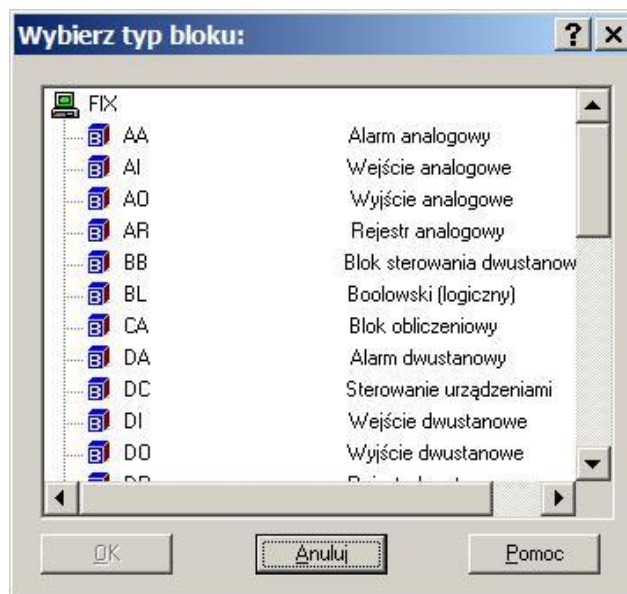
Rys. 6.12 Przykładowy łańcuch bloków w bazie danych systemu iFIX.

Fizycznie z obiektem komunikuje się PLC, wartość wejścia I0.23 w skali znormalizowanej 0 . . . 1 jest automatycznie wpisywana do bloku AI, następnym blokiem jest RB (blok skalowania), który zamienia sygnał znormalizowany, na sygnał w jednostkach fizycznych 0 . . . 100. Następnie wartość PV jest wprowadzana do bloku TR (bufor wykresu); w celu prezentacji wartości operatorowi, a ostatecznie trafia na wejście bloku PID. Wartość SP, jest wprowadzana do PID poprzez, jego odpowiednie pole danych, które jest połączone z blokiem TR. Wartość wyjściowa bloku PID jest połączona z RB w celu przeliczenia wartości sygnału na skalę znormalizowaną i ostatecznie trafia do bloku AO gdzie ten wpisuje ją do wyjścia PLC pod adres Q1.2.

Oczywiście powyższy łańcuch bloków ma sens w sytuacji, kiedy korzystamy z funkcji SCADA systemu iFix, w przypadku gdy PLC samodzielnie realizuje sterowanie, ograniczamy się do funkcji HMI oferowanej przez iFix i jedynie monitorujemy wartości PLC I0.23, Q1.2 oraz ustawiamy nowe wartości punktów pracy.

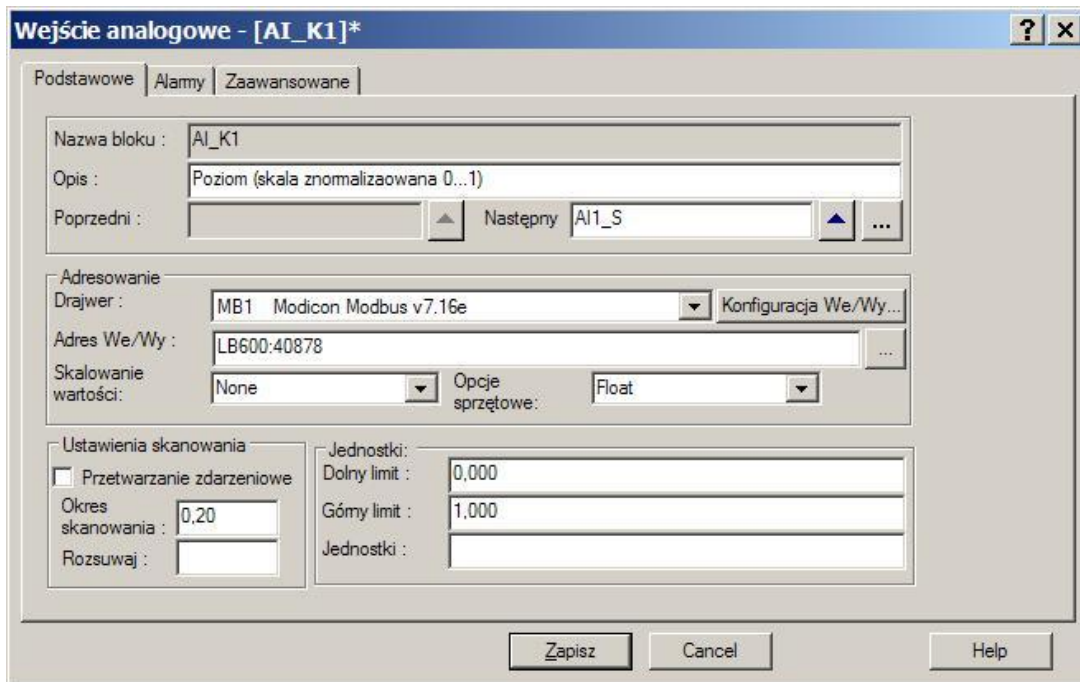
Wszystkie wspomniane powiązania możemy wykonać podczas dodawania nowego, bądź modyfikacji już istniejącego bloku. Prześledźmy teraz utworzenie w bazie danych nowego bloku AI.

Aby dodać nowy blok do bazy, należy z górnego menu wybrać opcję bloki a następnie dodaj. W nowym oknie prezentującym listę typów dostępnych bloków, należy zaznaczyć blok AI i wybrać przycisk OK.



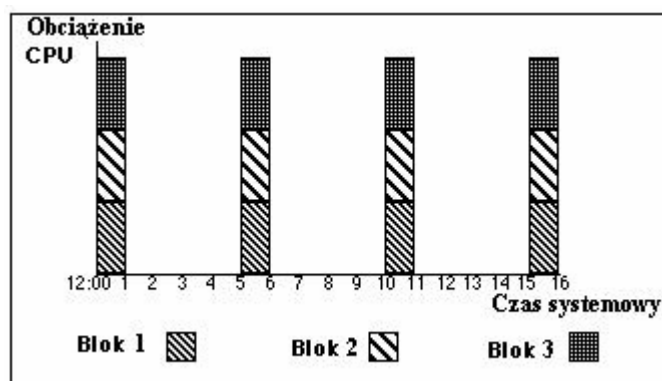
Rys. 6.13 Okno wyboru rodzaju bloków, przed dodaniem do procesowej bazy danych.

Po wyświetleniu formatki bloku AI, możemy przystąpić do jego konfiguracji, formatka domyślnie otwiera się na karcie „Podstawowe”. W polu „Nazwa bloku” wprowadzamy unikalną nazwę, która w całym systemie będzie reprezentowała ten blok. Jeżeli chcemy aby blok przekazywał swoją wartość do następnego bloku, to uzupełniamy pole „Następny”. Sekcja „Adresowanie”, służy do skojarzenia bloku z drajwerem We/Wy, tak aby poprzez niego, nowo tworzony blok AI, mógł otrzymywać dane z urządzenia obiektowego np. regulatora. Pole „Skalowanie wartości”, służy do przetworzenia sygnału z drajwera w przypadku, gdyby urządzenie obiektowe wysyłało dane pierwotne (Raw data). Pole „Opcje sprzętowe” determinuje typ danych przesyłanych do AI (np. wybranie opcji Float oznacza iż przesyłane dane będą miały postać liczby 32-bitowej zmiennoprzecinkowej, Unsigned - liczby 16-bitowej dodatniej, całkowitej, itp).



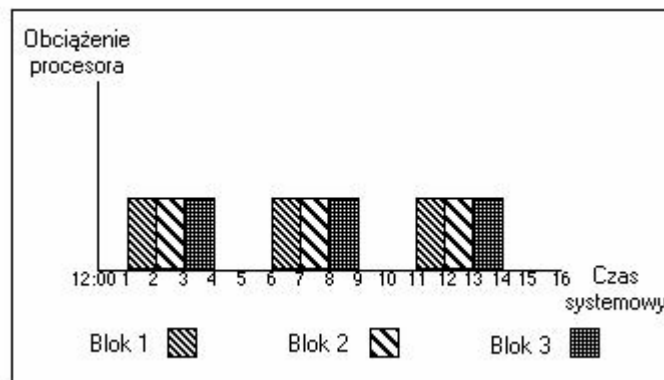
Rys. 6.14 Ustawienia konfiguracyjne dla bloku AI – karta podstawowe.

Następną dosyć ważną sekcją jest „Ustawienia skanowania”, która zawiera dwa pola „Okres skanowania” oraz „Rozsuwaj”. Pierwsze pole służy do określenia interwału czasowego z jakim blok AI będzie pobierał dane z drajwera We/Wy. Innymi słowami, tutaj można określić jak szybko dane mają być odświeżane. Drugie pole służy do określania fazy, w której sygnał będzie odbierany. Określenie fazy jest bardzo pomocne w przypadku gdy w bazie danych jest dużo wejść typu AI, skanowanych w tym samym czasie. Dzięki temu zabiegowi możemy dość znacząco odciążyć procesor komputera na którym baza jest przetwarzana. Aby lepiej zrozumieć pojęcie fazowania (rozsuwania) rozważmy następujący przykład. Trzy bloki AI są skanowane (pobierają dane z drajwera We/Wy) co 5 sekund



Rys. 6.15 Obciążenie procesora w trakcie wysyłania żądania do drajwera, bez skonfigurowanych faz dla bloków danych.

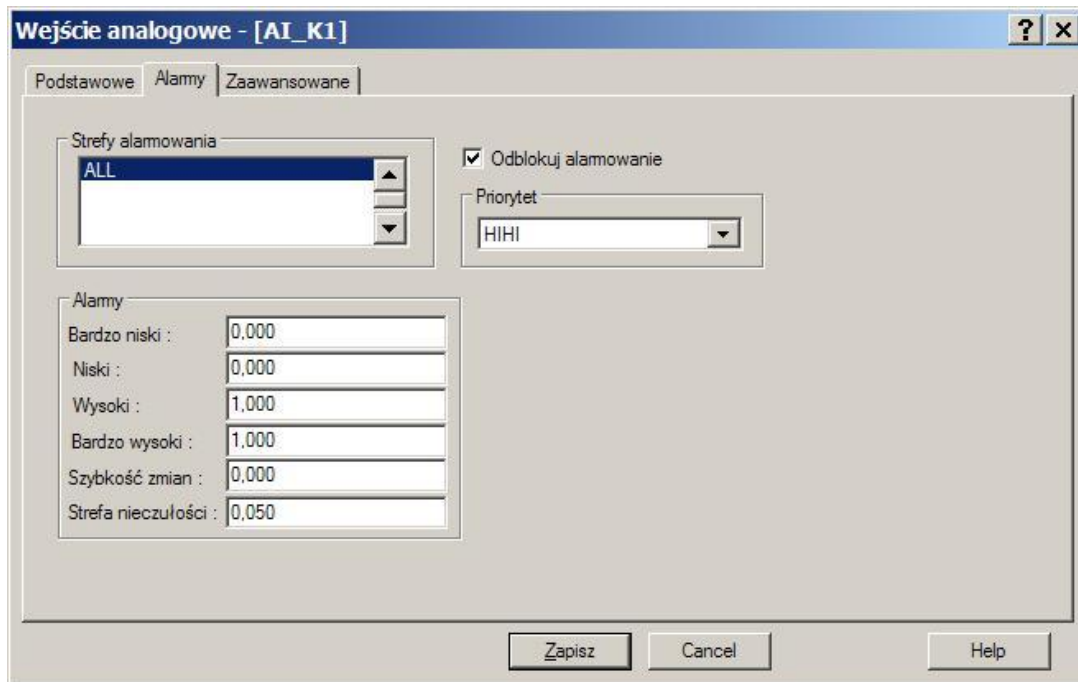
Jak widać program SAC (Scan And Control) odpowiadający za dostarczanie danych do bazy procesowej wysyła żądania do drajwera We/Wy, w tym samym czasie. Jeżeli dla pierwszego bloku ustawimy fazę 1 sekundową, dla drugiego – 2 sekundową oraz dla trzeciego bloku 3 sekundową. Zmniejszyśmy znacząco obciążenie procesora. Podczas tej optymalizacji, należy pamiętać o zasadzie: aby największa wartość fazy była mniejsza od okresu skanowania.



Rys. 6.16 Obciążenie procesora w trakcie wysyłania żądań do drajwera, po skonfigurowaniu fazowania.

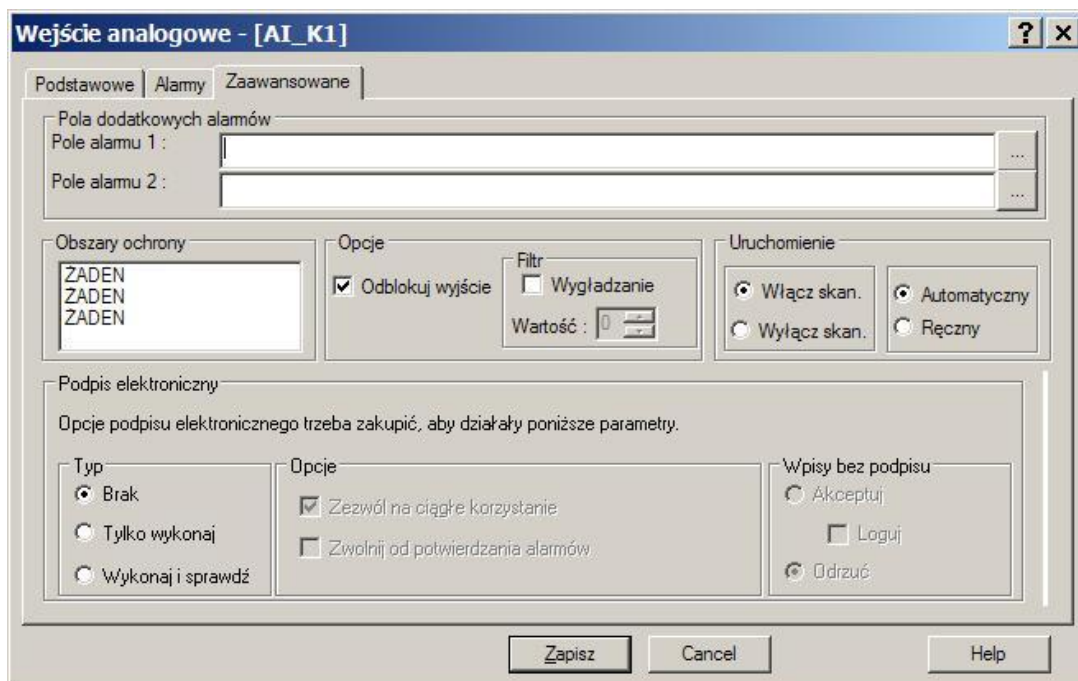
Ostatnią sekcją, którą należy uzupełnić na karcie „Podstawowe” jest obszar „Jednostki” , gdzie definiujemy granice, w których powinna znaleźć się wartość reprezentowana przez ten blok. Granice są wyrażone w jednostkach fizycznych, tutaj należy pamiętać, iż system iFix, uniemożliwi wprowadzenie przez operatora wartości wykraczających poza ustalone w tych polach granice.

Karta „Alarmy”, służy do konfiguracji alarmów, które obsługują dany typ bloku. Dla naszego bloku możemy skonfigurować opcje przedstawione na poniższym rysunku. Jeżeli chcemy aby dany blok nie wysyłał alarmów, to należy usunąć znacznik przy opcji „Odblokuj alarmowanie”.



Rys. 6.17 Ustawienia konfiguracyjne dla bloku AI – karta Alarmy.

Omówmy ostatnią kartę „Zaawansowane”.



Rys. 6.18 Ustawienia konfiguracyjne dla bloku AI – karta Zaawansowane.

Sekcja „*Obszar ochrony*” umożliwia przypisanie danego rysunku do konkretnego obszaru ochrony. Tą cechę możemy wykorzystać, jeżeli w tworzonej aplikacji zaimplementowaliśmy system ochrony i chcemy ograniczyć dostęp do tego rysunku poprzez nieuprawnionych użytkowników (wyłączając wybrany obszar ochrony z ich uprawnień). W sekcji „*Opcje*”, znacznik „*Odblokuj wyjście*” służy do umożliwienia wysłania danych do urządzenia. W podsekcji „*Filtr*” mamy możliwość włączenia cyfrowego filtra, którego zasadę obrazuje poniższa zależność:

$$\text{Wyjście} = (S/16)X1 + ((16 - S)/16)X2$$

Gdzie: S – wartość wygładzana wprowadzona w polu wartość,

X1 – początkowa wartość lub poprzednie wejście,

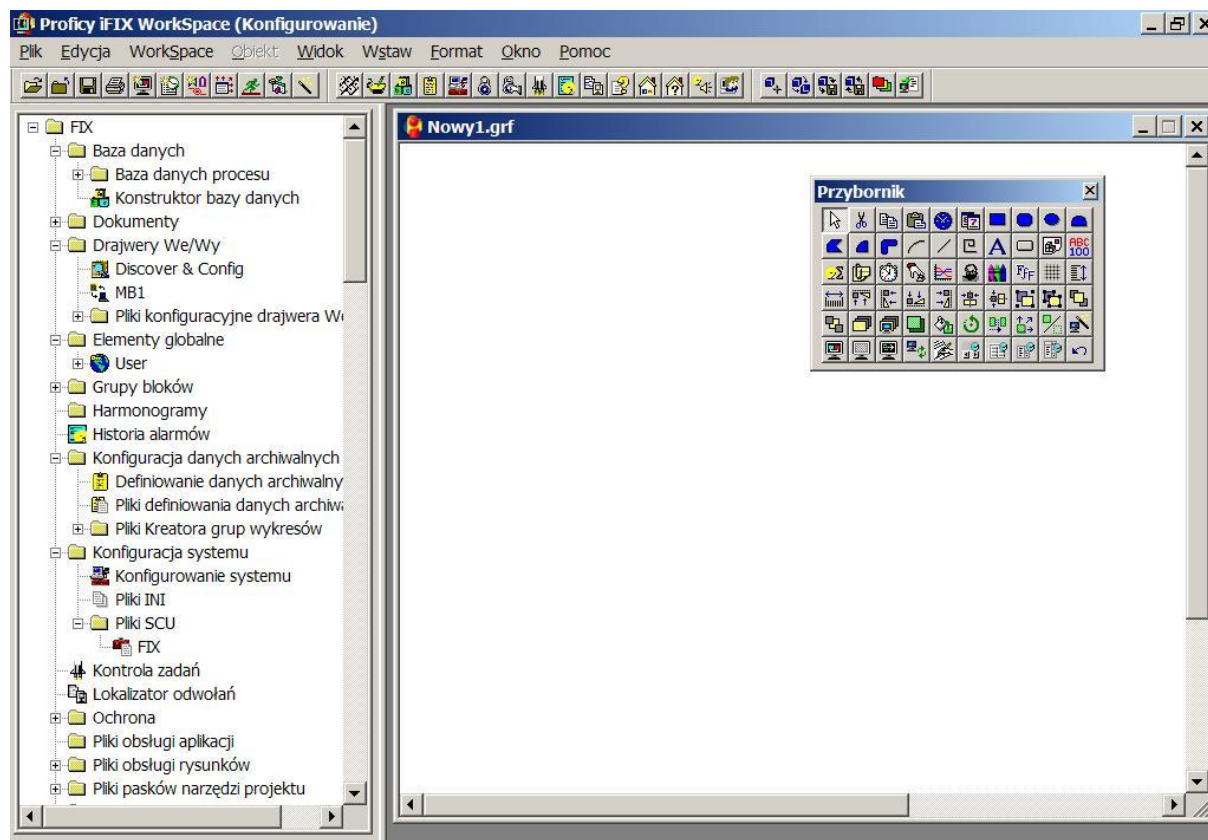
X2 nowa wartość wejścia z drajwera We/Wy.

Sekcja „*Uruchomienie*” zawiera opcje Wł/Wył skanowanie oraz Automatyczny - Ręczny. Pierwsza opcja określa czy program SAC ma przetwarzać dany blok (wysłać do bloku wartości, pochodzące z drajwera We/Wy zgodnie z ustawieniami skanowania). Druga opcja określa czy blok otrzymuje dane z drajwera We/Wy (opcja Automatyczny) czy też od operatora bądź też skryptu VBA (Ręczny).

Po skonfigurowaniu wszystkich opcji, należy wybrać przycisk „*Zapisz*” aby zachować nowy blok w bazie danych.

6.6 Workspace

W systemie iFix środowiskiem projektowym, do tworzenia rysunków, jest program Workspace uruchomiony w trybie konfiguracyjnym. Workspace dysponuje wieloma narzędziami graficznymi, tekstowymi, animacyjnymi, dostępu do procesowej bazy zmiennych i do tworzenia wykresów, tak by projektowane rysunki były atrakcyjne, dobrze zorganizowane, łatwe w obsłudze i zrozumieniu. Po utworzeniu rysunków można je przeglądać przy pomocy programu Workspace uruchomionego w trybie wykonywania. Pomiedzy dwoma trybami, przechodzi się przy pomocy skrótu klawiszowego Ctrl – W.



Rys. 6.19 Widok programu WorkSpace w trybie konfiguracyjnym.

WorkSpace składa się z następujących elementów:

- drzewo systemowe,
- obszar roboczy,
- paski narzędziowe,
- paski menu.

Drzewo systemowe jest umieszczone po lewej stronie okna WorkSpace jego głównym zadaniem jest organizacja pracy podczas projektowania aplikacji, oraz zapewnienie dostępu do konfiguracji wszystkich niezbędnych elementów systemu iFIX min:

- procesowej bazy danych,
- konfiguracji drajwera We/Wy,
- rysunków,
- konfiguracji danych archiwalnych,
- konfiguracji systemu,
- konfiguracji ochrony.

Drzewo jest również źródłem informacji o lokalizacji ważniejszych plików systemu iFix.

Obszar roboczy wyświetla aktualnie projektowany rysunek, a paski narzędziowe i menu, służą do szybkiego wywoływania pożądanych narzędzi projektowych, np. ekspert widzialności, pole danych, zdarzenie itp.

6.7 Tworzenie aplikacji HMI/SCADA

Tworzenie nowej aplikacji bez uprzedniego przygotowania podstawowych informacji jest procesem żmudnym, czasochłonnym i w większości przypadków końcowe rezultaty nie są wystarczająco zadowalające. Aby proces tworzenia nowej aplikacji przebiegł szybko, a końcowe efekty przyniosły oczekiwany rezultat, praca powinna być odpowiednio zorganizowana i podzielona na następujące etapy:

- **ETAP 1 – Analiza**
 - Analiza działania procesu
 - Wyróżnienie i zapisanie zadań, które należy zautomatyzować
 - Zebranie szczegółowych informacji na temat działania urządzeń procesowych
 - Szczegółowa analiza procesu
 - Określenie alarmów i ich progów
 - Stworzenie algorytmu automatyzacji
 - Określenie typów bloków, które będą użyte do zaimplementowania algorytmu automatyzacji
 - Opracowanie schematu przepływu sygnałów w procesowej bazie danych
 - Porównanie analizy działania procesu z przepływem sygnałów, z uwzględnieniem stanów alarmowych
 - Poprawki.
- **ETAP 2 – Projekt**
 - Zebranie wymagań dotyczących funkcjonowania aplikacji
 - Podział funkcjonalny ekranów synoptycznych
 - Opracowanie projektu graficznego ekranów synoptycznych
 - Analiza wykonywalności
 - Poprawki

- **ETAP 3 – Implementacja**
 - Stworzenie bazy danych na podstawie schematu przepływu sygnałów z etapu 1 oraz wstępna konfiguracja drajwera We/Wy
 - Wykonanie ekranów synoptycznych na podstawie projektu graficznego
 - Podłączenie obiektów animowanych na ekranach z bazą danych
 - Połączenie ekranów synoptycznych (jeżeli jest takie wymaganie)
 - Uzupełnienie funkcjonalności na podstawie wymagań funkcjonalnych z etapu 2
 - Optymalizacja drajwera We/Wy oraz czasów skanowania bloków w PDB
 - Wstępne testy aplikacji
 - Poprawki
- **ETAP 4 – Testy**
 - Przygotowanie scenariuszy testowych w oparciu o Etap 1 oraz 2
 - Przeprowadzenie testów głównych
 - Dokumentacja błędów i poprawek
 - Przeprowadzenie skróconych testów końcowych głównie pod kątem wykrytych błędów
- **ETAP 5 - Przygotowanie dokumentacji**
 - Dokumentacja techniczna
 - Instrukcje obsługi
 - Ewidencja błędów wykrytych podczas użytkowania
 - Ewidencja poprawek błędów

7 System wizualizacji i sterowania układu regulacji poziomu cieczy w zbiornikach otwartych.

Poniższy rozdział jest próbą udokumentowania procesu projektowania systemu wizualizacji i sterowania w oparciu o wskazówki projektowe ujęte w rozdziale 6.

7.1 Analiza projektu

Celem analizy jest przygotowanie niezbędnych informacji odnośnie urządzeń oraz oprogramowania. Analiza ma dostarczyć odpowiedzi na pytania:

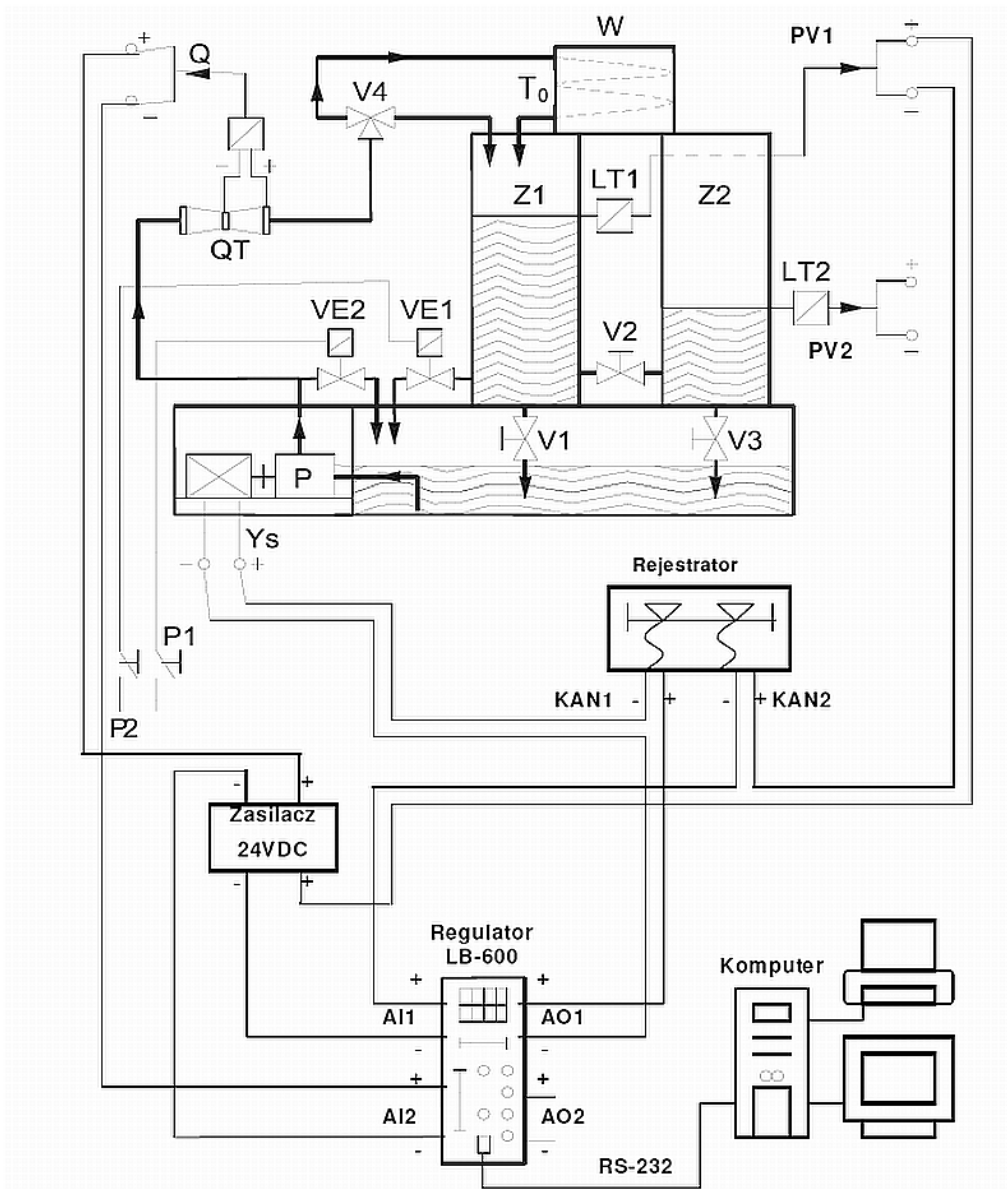
1. Co będziemy wykonywać?
2. Jakie zasoby mamy do dyspozycji?
3. W jaki sposób zostaną wykonane poszczególne rzeczy?

7.1.1 Informacje o urządzeniach obiektowych

Obiekt składa się z następujących urządzeń:

- Przetwornik poziomu, zakres 0-500 mmH₂O, sygnał wyjściowy: 4-20mA,
- Przetwornik przepływu, zakres:0-7 l/min , sygnał wyjściowy: 4-20mA,
- Pompa o zmiennej wydajności, zakres: 0-7 l/min, sygnał sterujący: 4-20mA,
- Regulator mikroprocesorowy LB600 wersja oprogramowania: 2.21.11, DTR 19.11.2007, sygnały dostępne w skali znormalizowanej 0...1, komunikacja Modbus/RTU, prędkość komunikacji – konfigurowalna.

Na poniższym rysunku przedstawiono schemat połączeń urządzeń obiektowych z regulatorem



Rys. 7.1 Schemat połączeń urządzeń obiektowych z regulatorem.

7.1.2 Szczegółowa analiza procesu

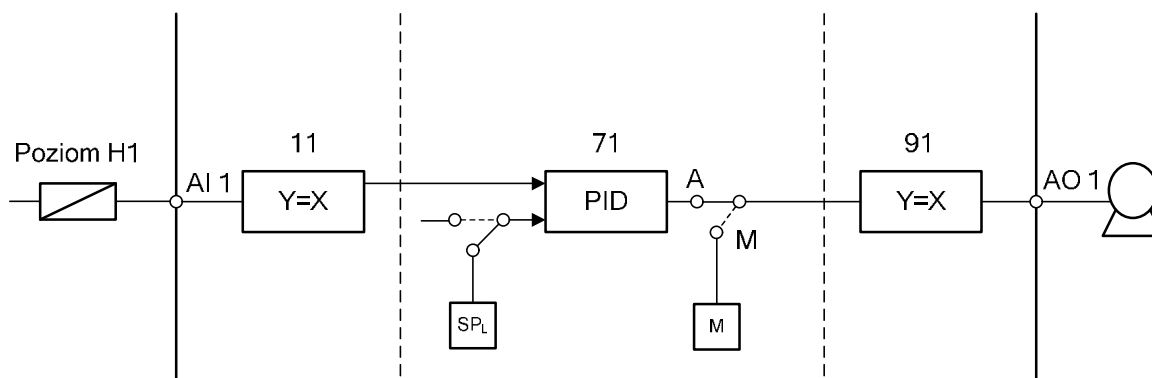
Komunikacja z regulatorem odbywa się za pośrednictwem protokołu Modbus/RTU, z tego powodu aplikacja zbiera i przesyła dane za pomocą drajwera MB1. Ze względu iż komunikujemy się z rejestrami regulatora, ostateczny adres, który należy wprowadzić do drajwera, będzie miał postać

Adres drajwera = 40001 + numer rejestru regulatora

Proces monitoringu oraz regulacji poziomu cieczy i przepływu zakłada generowanie sygnału sterującego przez regulator PID, w celu utrzymania poziomu cieczy w zbiorniku na zadanym poziomie, oraz bieżący monitoring zmiennych procesowych takich jak poziom, przepływ. Ponieważ przetworniki pomiarowe oraz urządzenia wykonawcze są połączone z regulatorem, zatem wszystkie niezbędne informacje można uzyskać z odpowiednich rejestrów regulatora. Dodatkowo istnieje potrzeba zmiany parametrów pracy regulatora oraz realizacja układu regulacji w strukturach jednoobwodowej oraz kaskadowej. O wyborze struktury układu regulacji decyduje użytkownik. Szczegółowe zestawienie parametrów niezbędnych do poprawnej pracy przedstawiono w **załączniku A**.

7.1.2.1 Jednoobwodowy układ automatycznej regulacji

Poniższy rysunek przedstawia strukturę funkcjonalną regulatora w jednoobwodowym układzie regulacji.



Rys. 7.2 Schemat funkcjonalny regulatora LB-600 w strukturze jednoobwodowej.

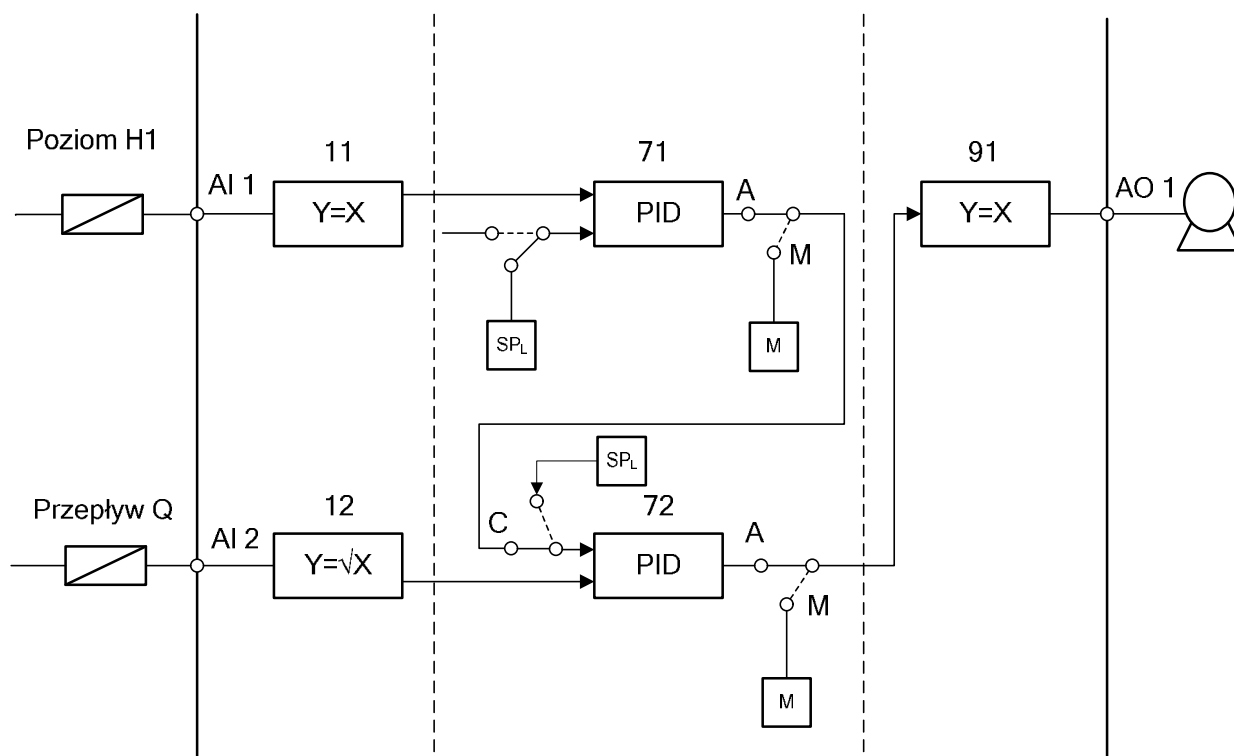
W poniższej tabeli zestawiono programowane parametry regulatora, oraz odpowiadające im adresy rejestrów, niezbędne w celu pracy w jednoobwodowym układzie regulacji.

Tabela 3 Zestawienie adresów, regulatora LB-600, realizujących jednoobwodowy układ regulacji.

Adres	Wartość	Adres	Wartość	Adres	Wartość	Adres	Wartość
1104	2	2204	0	7159	0	7462	100
1105	0	7102	1	7161	0	7465	1
1106	100	7103	1	7162	100	8106	0
1107	0	7107	0	7165	1	8111	8
1108	100	7108	100	7265	0	9102	7
1109	0	7109	0	7405	7	9102	7
1110	100	7110	1	7406	1	9103	1
1111	5	7123	1	7407	0	9103	1
1112	5	7124	100	7408	100	9111	1
1114	0	7125	100	7409	0		
1119	1	7126	10	7410	1		
1219	0	7127	1	7413	1		
2102	0	7142	328	7423	1		
2104	0	7143	329	7458	1		
2202	0	7158	0	7461	0		

7.1.2.2 Kaskadowy układ automatycznej regulacji

Struktura funkcjonalna regulatora przedstawiona jest na poniższym rysunku 7.3.



Rys. 7.3 Schemat funkcjonalny regulatora LB-600 w strukturze kaskadowej.

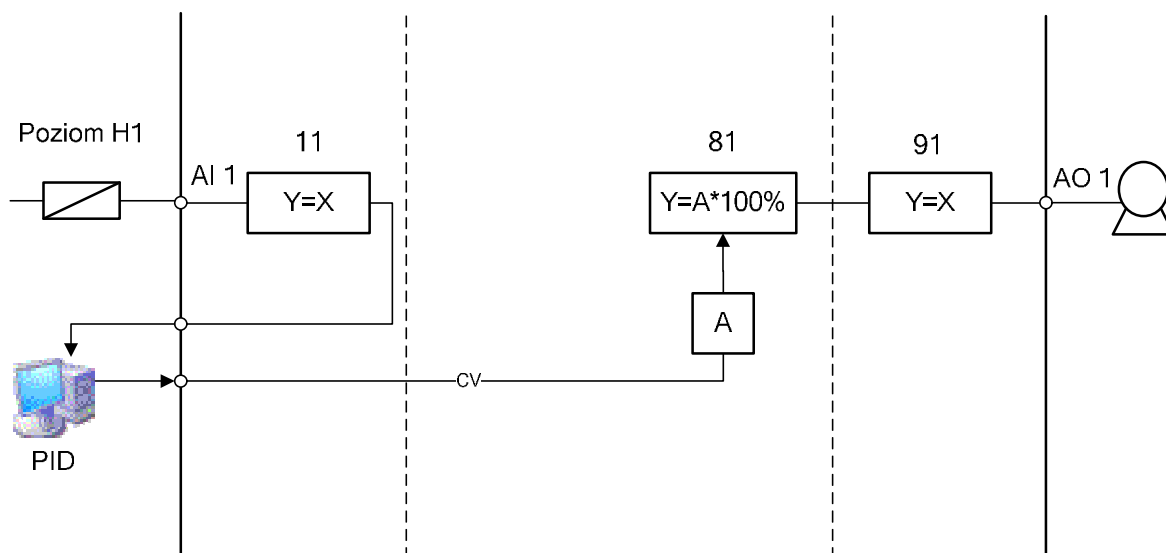
Zestawienie programowanych parametrów regulatora jest przedstawione w tabeli 4.

Tabela 4 Zestawienie adresów, regulatora LB-600, realizujących kaskadowy układ regulacji.

Adres	Wartość	Adres	Wartość	Adres	Wartość	Adres	Wartość
1104	2	2104	0	7207	0	7361	0
1105	0	2202	0	7208	100	7362	100
1106	100	2204	0	7209	0	7365	1
1107	0	7102	1	7210	1	7365	1
1108	100	7103	1	7223	1	7405	7
1109	0	7107	0	7224	100	7406	1
1110	100	7108	100	7225	100	7407	0
1111	5	7109	0	7226	10	7408	100
1112	5	7110	1	7227	1	7409	0
1114	0	7123	1	7258	0	7410	1
1119	1	7124	100	7261	0	7413	1
1204	2	7125	100	7262	100	7423	1
1205	0	7126	10	7265	1	7458	1
1206	100	7127	1	7305	7	7461	0
1207	0	7142	328	7306	2	7462	100
1208	100	7143	329	7307	0	8106	0
1209	0	7158	0	7308	100	8111	8
1210	100	7159	0	7309	0	9102	7
1211	5	7165	1	7310	1	9102	7
1212	5	7202	1	7313	1	9103	1
1214	0	7203	2	7323	1	9103	2
1219	1	7205	7	7358	1	9111	1
2102	0	7206	1	7359	0		

7.1.2.3 Jednoobwodowy układ automatycznej regulacji (Soft Control)

Na poniższym rysunku przedstawiono strukturę funkcjonalną regulatora, realizującą sterowanie Soft Control.



Rys. 7.4 Schemat funkcjonalny regulatora LB-600, realizującego Soft Control

W tabeli 5 zestawione są programowane parametry w celu realizacji jednoobwodowego układu regulacji z sygnałem CV generowanym przez komputer.

Tabela 5 Zestawienie adresów regulatora LB-600, ustawianych w celu realizacji Soft Control.

Adres	Wartość
1219	0
7165	0
7265	0
7365	0
7465	0
8106	8
8111	1
9102	8
9103	1
9111	1

7.1.3 Wybór odpowiednich bloków.

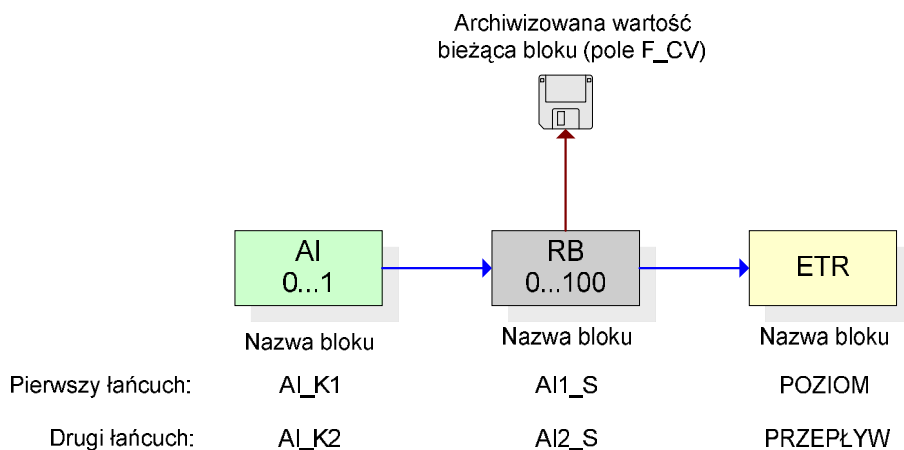
Poniżej przedstawiono bloki, użyte do utworzenia procesowej bazy danych:

- AI – blok analog input, pobiera dane z regulatora przez cały czas działania aplikacji, w ustawialnym interwale czasowym (0.05 – 23:59:59 [s]).

- AR – blok analog register, pobiera dane z regulatora z częstotliwością 1 [s], tylko jeżeli jest aktywny rysunek odwołujący się do niego. Dzięki tej właściwości można uaktualniać te dane, które w danym momencie są potrzebne.
- AO – blok analog output, zapisuje w regulatorze dane, za każdym razem gdy otrzyma nową wartość od bloku poprzedzającego go w łańcuchu.
- RB – blok ratio bias umożliwia skalowanie wartości uzyskanej od bloku poprzedzającego go w łańcuchu. Oprócz współczynnika skalującego, dodatkowo można skonfigurować przesunięcie.
- ETR – blok Extended Trend Block, buforuje maksymalnie 600 wartości, bloku poprzedzającego go w łańcuchu, oraz umożliwia wyświetlenie danych procesowych na wykresach, w postaci binarnej.
- PID – blok PID, działa w zamkniętej pętli, oraz generuje sygnał sterujący w celu eliminacji odchyłki wartości PV od SP.

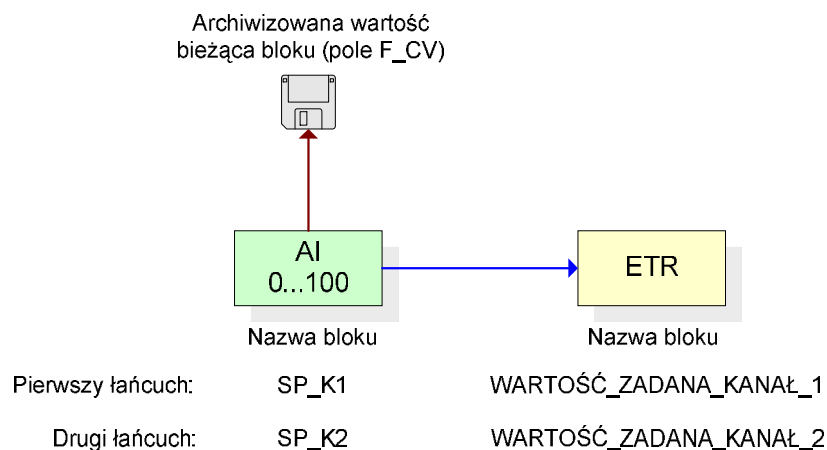
7.1.4 Schemat przepływu sygnałów w procesowej bazie danych

Przepływ sygnału w celu monitoringu oraz archiwizowania wartości PV przedstawiony jest na rysunku 7.5. Drajwer MB1, odczytuje w skali znormalizowanej 0...1, z regulatora określoną wartość PV. Następnie wartość ta jest przekazywana do bloku AI, który przesyła sygnał bloku skalowania RB, w celu przeliczenia go do skali 0...100. Wartość bloku skalowania jest archiwizowana oraz przesyłana do bloku ETR w celu udostępnienia danych na wykresie.



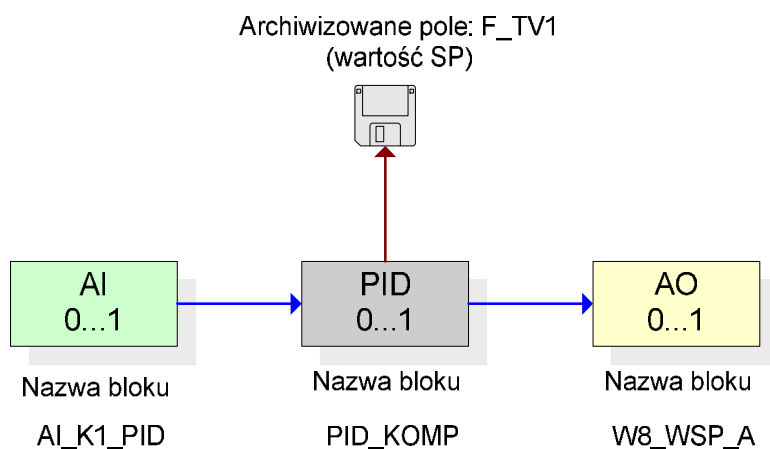
Rys. 7.5 Łańcuch bloków realizujący archiwizację oraz wyświetlanie na wykresie wartości PV.

Przepływ sygnału w celu monitoringu oraz archiwizacji wartości SP przedstawiony jest na rysunku 7.6. Ponieważ wartość SP jest wyrażona w skali 0...100, zatem po przekazaniu jej przez drajwer, system, natychmiastowo ją archiwizuje. Następnie wartość bloku AI jest przekazywana do bloku ETR w celu prezentacji danych na wykresie.



Rys. 7.6 Łańcuch bloków realizujący archiwizację oraz wyświetlanie na wykresie wartości SP.

Przepływ sygnału w procesowej bazie danych w celu realizacji Soft Control jest przedstawiony na rysunku 7.7. Wartość PV zapisana w bloku AI jest przekazywana do bloku PID, blok PID po obliczeniu sygnału CV, na podstawie wartości zadanej oraz wartości bloku AI, przesyła go do bloku AO, skąd wartość sygnału CV, poprzez drajwer MB1 wpisywana jest do odpowiedniego rejestru regulatora. Ponieważ wartość bloku AI (PV) jest archiwizowana w innym łańcuchu, dlatego postanowiono archiwizować jedynie wartość zadaną bloku PID, poprzez archiwizację wartości pola F_TV1.



Rys. 7.7 Łańcuch bloków realizujący Soft Control.

7.2 Projekt systemu

7.2.1 Wymagania funkcjonalne aplikacji

System wizualizacji i sterowania powinien posiadać następujące właściwości:

- przejrzystość,
- intuicyjna obsługa.

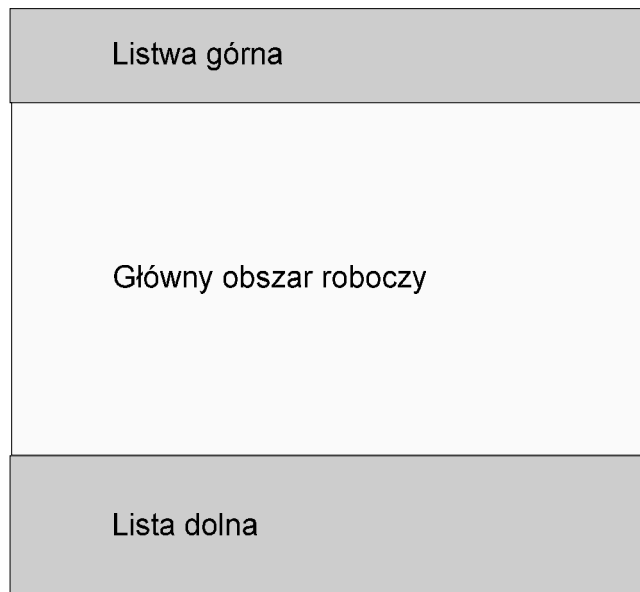
System wizualizacji i sterowania powinien zapewnić obsługę następujących funkcji:

- nadzór nad parametrami układu regulacji,
- prezentacja danych procesowych na wykresach,
- prezentacja i konfiguracja alarmów,
- zmiana parametrów pracy regulatora,
- zmiany nastaw algorytmu PID,
- obsługa samostrojenia,
- sterowanie ręczne,
- eksport wartości historycznych do pliku.
- wybór struktury układu regulacji,

oraz udostępniać mechanizm bezpieczeństwa, w celu ochrony dostępu, przez osoby niepowołane, do trybu konfiguracyjnego systemu.

7.2.2 Podział funkcjonalny ekranów synoptycznych

W celu przejrzystej prezentacji danych oraz ułatwienia obsługi systemu, przyjęto następujący podział obszaru roboczego komputera.



Rys. 7.8 Podział obszaru ekranowego monitora.

Listwa dolna oraz górna jest dostępna przez cały czas działania systemu, główny obszar roboczy jest miejscem prezentacji ekranów spełniających podstawowe funkcje.

Mając zaprojektowany podział logiczny aplikacji możemy przystąpić do przypisywania funkcji do poszczególnych części składowych obszaru roboczego komputera, oraz ustaleniu liczby ekranów głównych. Przyjęto iż listwa górna będzie spełniała następujące rolę:

- nawigacja pomiędzy głównymi ekranami,
- możliwość wyłączenia aplikacji,
- wyświetlanie daty i czasu,

a listwa dolna:

- wyświetlanie alarmów,
- wyświetlanie informacji o zalogowanym użytkowniku,
- umożliwienie uruchomienia programu służącego do logowania oraz wylogowywania użytkowników.

Przyjęto iż funkcja nadzoru nad parametrami układu regulacji będzie prezentowana w następujących postaciach:

- schemat blokowy,
- schemat funkcjonalny regulatora,
- schemat instalacji obiektowej.

W związku powyższym zdecydowano się na następujące kategorie tematyczne, głównych ekranów synoptycznych:

- obiekt,
- wykresy,
- schemat funkcjonalny,
- ustawienia.

Do każdej kategorii tematycznej, ostatecznie przypisano funkcję, które mają być spełniane przez ekrany w niej się znajdujące. Tak więc, kategoria obiekt pełni następujące funkcje:

- nadzór nad parametrami układu regulacji,
 - prezentacja w postaci instalacji obiektowej,
 - prezentacja danych na schemacie blokowym,
- zmiana parametrów pracy regulatora/ów,
- zmiana nastaw algorytmu PID,
- sterowanie ręczne,
- obsługa samostrojzenia.

Z racji pełnionych funkcji, kategoria obiekt jest najważniejsza.

Kategoria wykresy prezentuje dane procesowe w postaci wykresów.

Kategoria schemat funkcjonalny:

- nadzór nad parametrami układu,
 - prezentacja danych w postaci schematu funkcjonalnego regulatora
- zmiana nastaw algorytmu PID,
- sterowanie ręczne

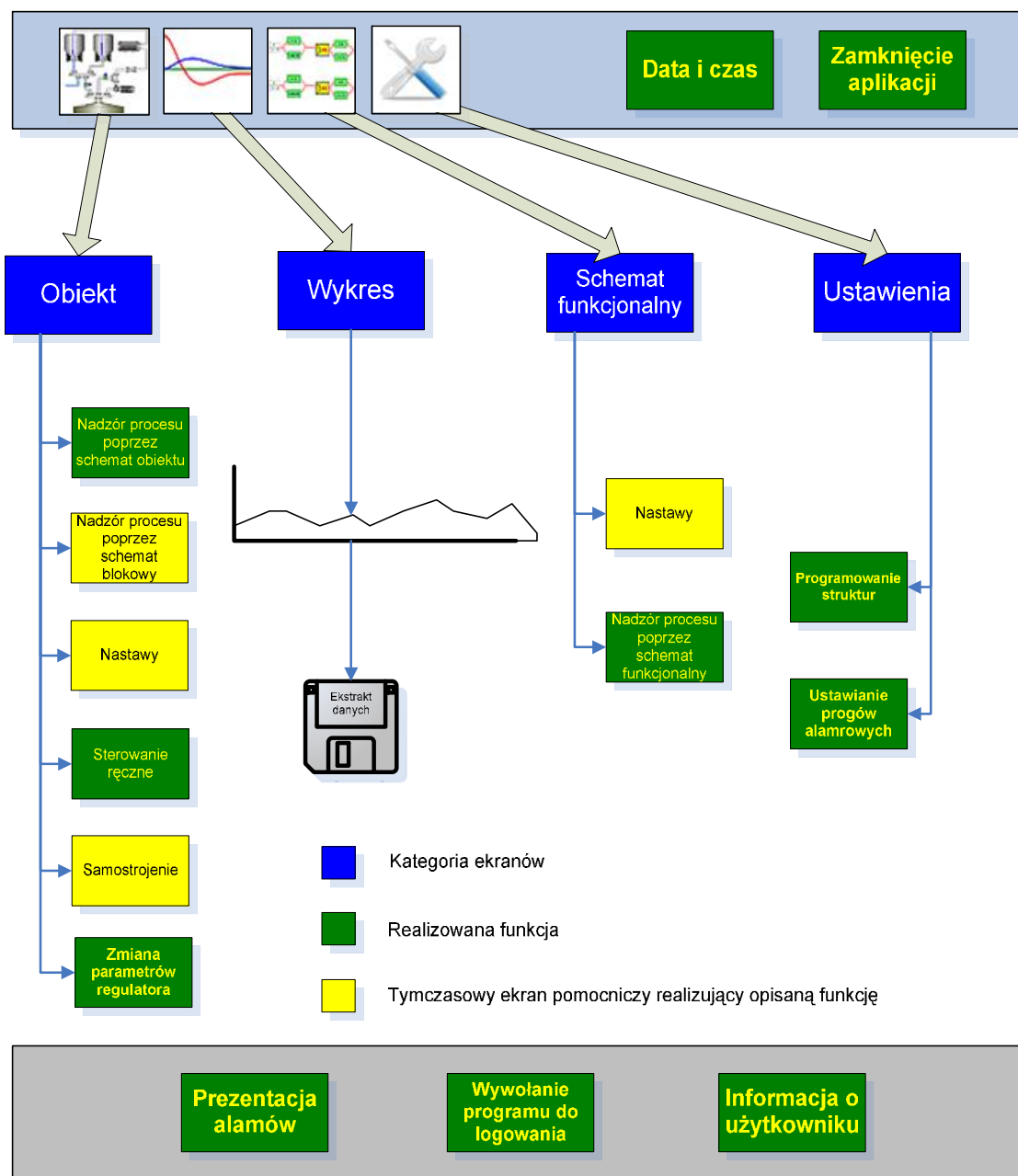
Ostatnia kategoria ustawienia, udostępnia następujące funkcje:

- programowanie wybranych struktur, układu regulacji,
- ustawianie progów alarmowych.

Ponieważ system umożliwia obsługę dwóch struktur układów regulacji, ze sterowaniem realizowanym poprzez regulator oraz jednej struktury ze sterowaniem typu Soft Control, ostatecznie w kategoriach tematycznych, należy zaprojektować następujące ekrany:

- **Obiekt**
 - *ekran obiektu w strukturach sterowanych regulatorem,*
 - nastawy regulatora głównego,
 - nastawy regulatora pomocniczego,
 - obsługa samostrojzenia,
 - schemat blokowy struktury jednoobwodowej,
 - schemat blokowy struktury kaskadowej,
 - *ekran obiektu w strukturach Soft Control,*
 - nastawy bloku regulatora (Soft Control),
 - ustawienia współczynników algorytmu PID,
 - schemat blokowy struktury jednoobwodowej.
- **Wykres**
 - *ekran wykresów.*
- **Struktura funkcjonalna**
 - *struktura funkcjonalna układu jednoobwodowego sterowana regulatorem,*
 - *struktura funkcjonalna układu kaskadowego sterowana regulatorem,*
 - *struktura funkcjonalna układu jednoobwodowego Soft Control.*
- **Ustawienia**
 - *ekran Ustawień.*

W celu lepszego zrozumienia powyższej funkcjonalności projektowanego systemu, zobrazowano ją na poniższym rysunku:



Rys. 7.9 Funkcjonalność systemu wizualizacji i sterowania.

7.2.3 Projekt graficzny ekranów synoptycznych

Podczas projektowania graficznego ekranów synoptycznych należy wykonać następujące rzeczy:

- opracować spójną szatę graficzną dla wszystkich ekranów,
- podzielić ekran synoptyczny na logiczne części,

- zaprojektować elementy realizujące wybraną funkcję oraz rozmieścić je na wybranych częściach ekranu synoptycznego,
- opracować wstępne warunki widzialności/animacji poszczególnych elementów.

7.2.3.1 Listwa górna

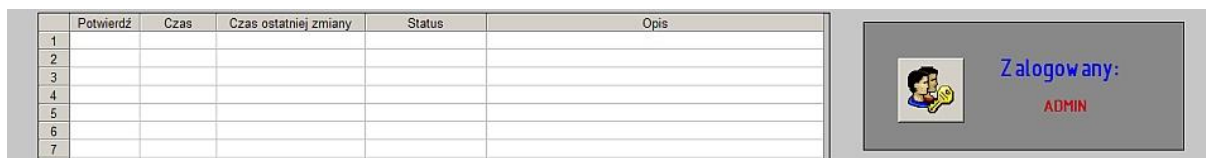
Do realizacji nawigacji pomiędzy ekranami głównymi, użyto wymownych ikon (rys. 7.10), które działają na zasadzie przycisku dwustanowego. Sygnalizacja otwartego rysunku odbywa się poprzez wyświetlanie efektu wklęsłości oraz zmiany koloru tła ikony. Otwieranie oraz zamykanie klawiatury wirtualnej oraz zamykanie aplikacji odbywa się poprzez wybór odpowiednich przycisków.



Rys. 7.10 Listwa górna.

7.2.3.2 Listwa dolna

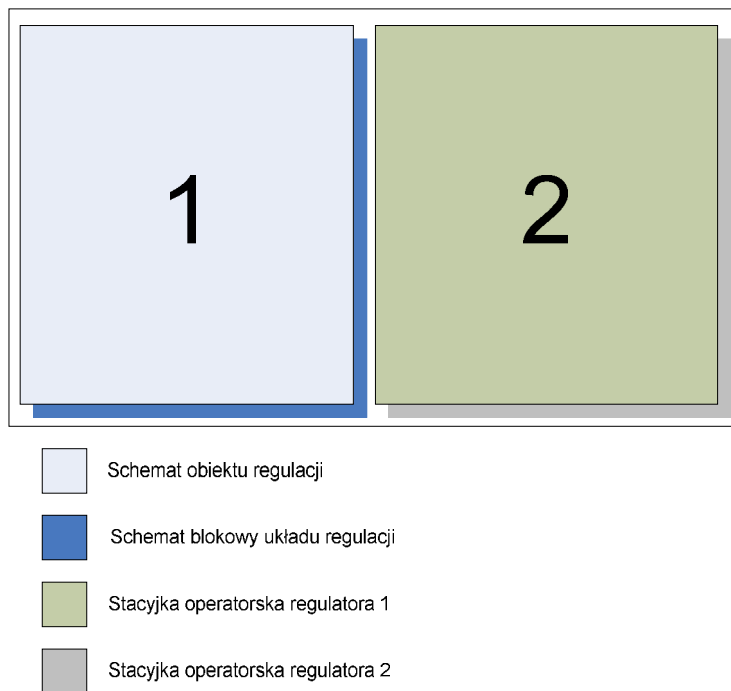
Lista prezentuje zestawienie alarmów, skróconą nazwę zalogowanego obecnie użytkownika oraz poprzez wybranie odpowiedniego przycisku umożliwia wywołanie programu obsługującego logowanie operatorów.



Rys. 7.11 Listwa dolna.

7.2.3.3 Ekran kategorii: obiekt

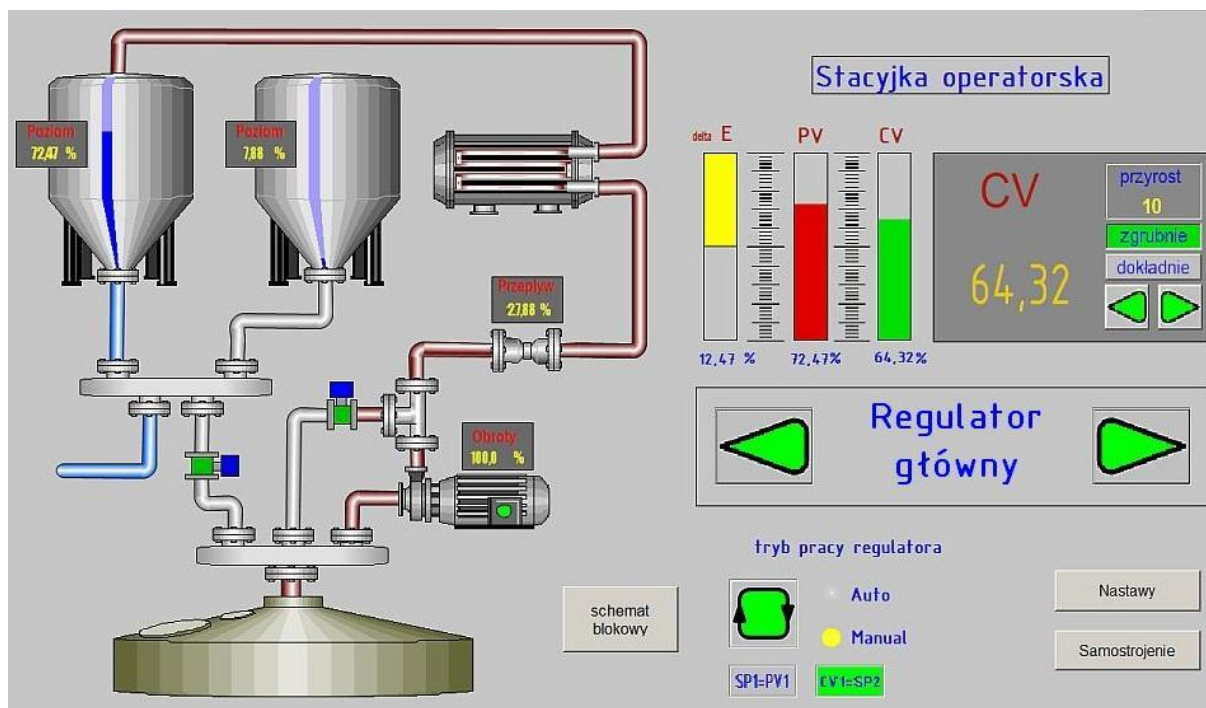
Ekran obiekt został logicznie podzielony na dwie części. Część pierwsza pełni rolę informacyjną, a część druga umożliwia zmianę parametrów pracy regulatora. Każda z części zawiera po dwa obiekty co przedstawiono na rysunku 7.12.



Rys. 7.12 Podział logiczny ekranu synoptycznego – obiekt.

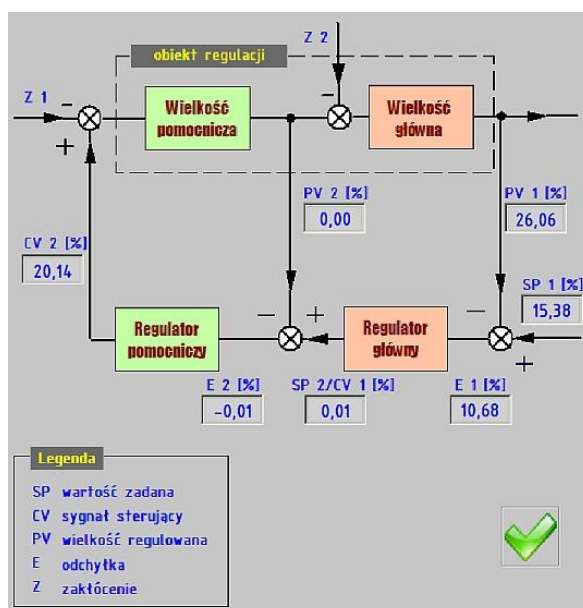
Wszystkie obiekty tworzą całość, prezentacja poszczególnych obiektów odbywa się poprzez wybranie odpowiednich przycisków. Domyślnie podczas inicjalizacji ekranu wyświetlają się obiekty:

- schemat obiektu regulacji,
- stacyjka operatorska regulatora (w przypadku układu kaskadowego – stacyjka regulatora głównego).



Rys. 7.13 Ekran synoptyczny – Obiekt w strukturze kaskadowej.

W zależności od wybranego układu regulacji, po naciśnięciu przycisku *schemat blokowy*, aplikacja wyświetla odpowiedni rysunek (np. rysunek 7.14). Znaki „+” oraz „-”, przy węzłach sumacyjnych są powiązane z działaniem na wyjściu regulatora (Normal albo Rewers).



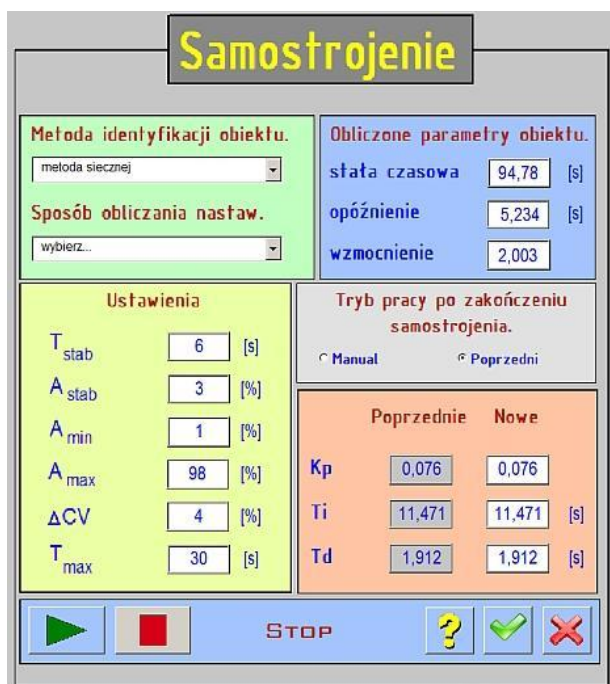
Rys. 7.14 Ekran synoptyczny – schemat blokowy w strukturze kaskadowej.

Po wybraniu przycisku *nastawy*, system wyświetla obiekt umożliwiający zmianę nastaw wybranego regulatora. Informacja o wybranym działaniu regulatora (Normal albo Rewers) jest podświetlona na zielono.



Rys. 7.15 Ekran synoptyczny – nastawy.

Po wybraniu przycisku *samostrojenie* aplikacja wyświetla obiekt umożliwiający obsługę samostrojenia. Przycisk jest dostępny jedynie dla struktur w, których sygnał sterujący jest generowany przez regulator.

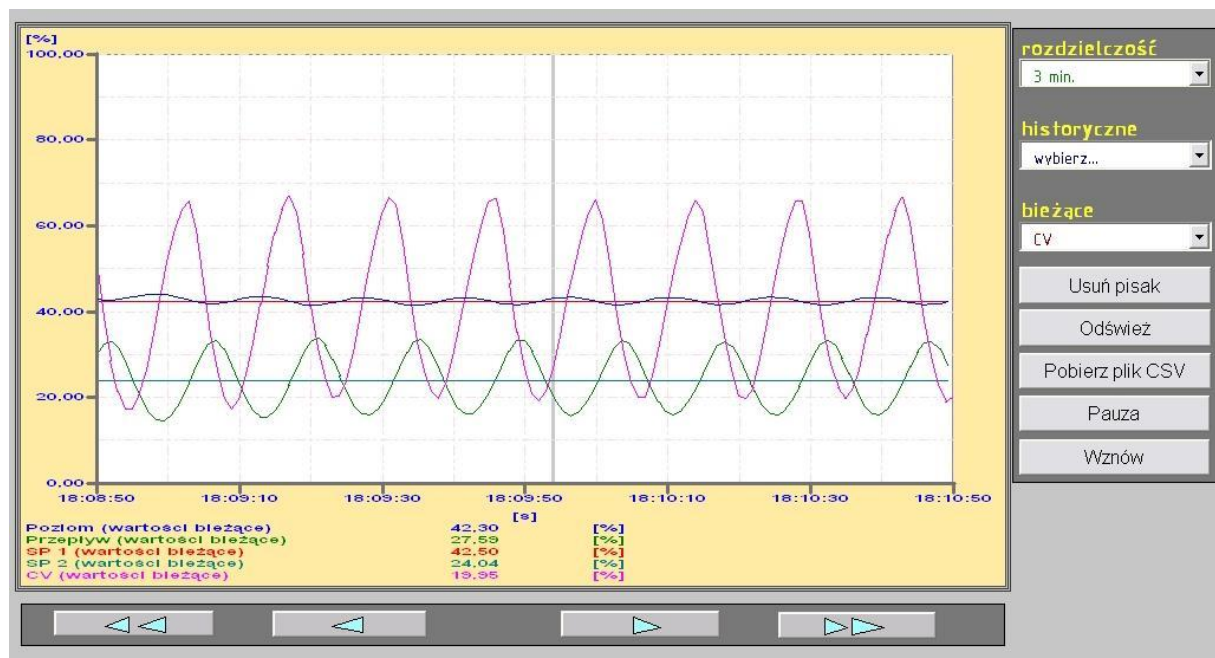


Rys. 7.16 Ekran synoptyczny – samostrojenie.

7.2.3.4 Ekran kategorii: wykres

Dla wszystkich struktur układu regulacji jest dostępny jeden wykres, wyróżnikiem, który informuje o wybranej strukturze (jednoobwodowa, kaskadowa, jednoobwodowa Soft Control) są różne wartości wyświetlające się, w rozwijanych listach. Dodatkowo ekran zawiera przyciski umożliwiające:

- usunięcie pisaków zgodnie z zasadą FIFO,
- odświeżenie danych na wykresie,
- eksport danych historycznych do pliku *.CSV,
- przewijanie danych historycznych.

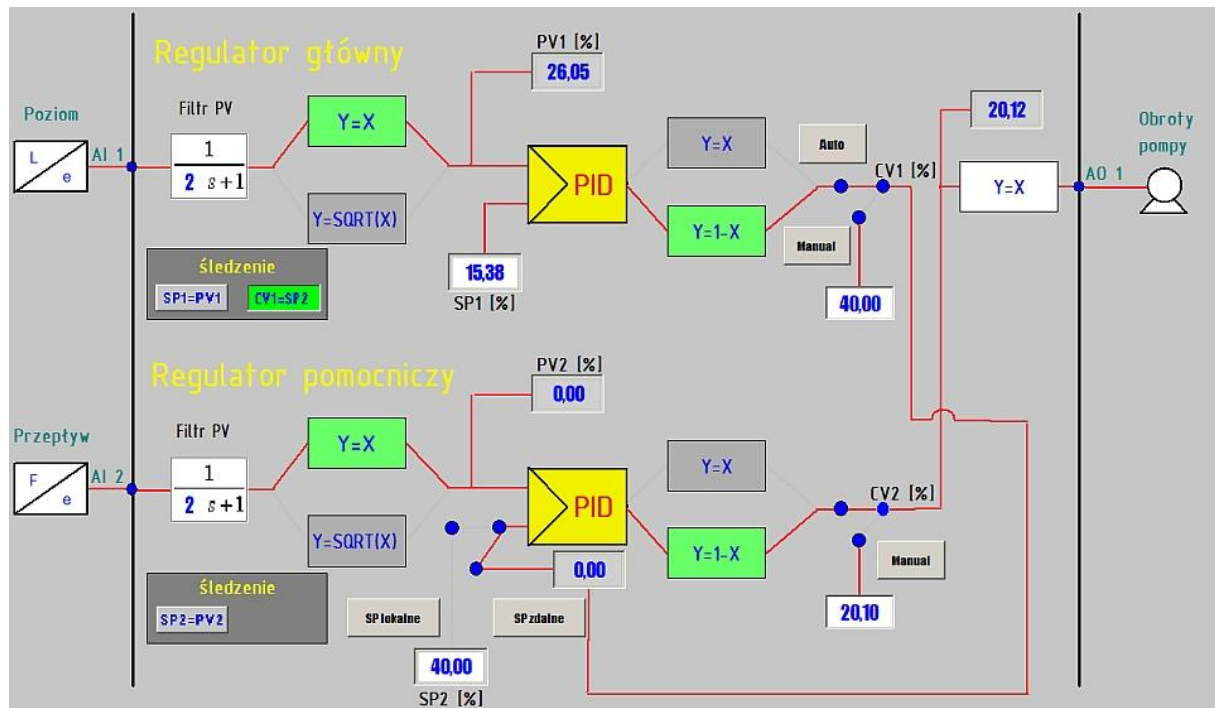


Rys. 7.17 Ekran synoptyczny – wykres.

7.2.3.5 Ekran kategorii: schemat funkcjonalny

Dostępne są trzy rysunki, kategorii schemat funkcjonalny, wyświetlenie odpowiedniego, jest uwarunkowane zaprogramowaną strukturą regulatora. Aktualny przepływ sygnałów, w wybranych funkktorach regulatora jest wyświetlany liniami w kolorze czerwonym. Ekran umożliwia zmianę algorytmu przetwarzania sygnału warstwy 1, działanie

wyjściowe z bloku regulatora, wybranie śledzenia, oraz zmianę stałej czasowej filtracji. Dodatkowo wybierając blok PID możemy wywołać ekran umożliwiający zmianę nastaw.



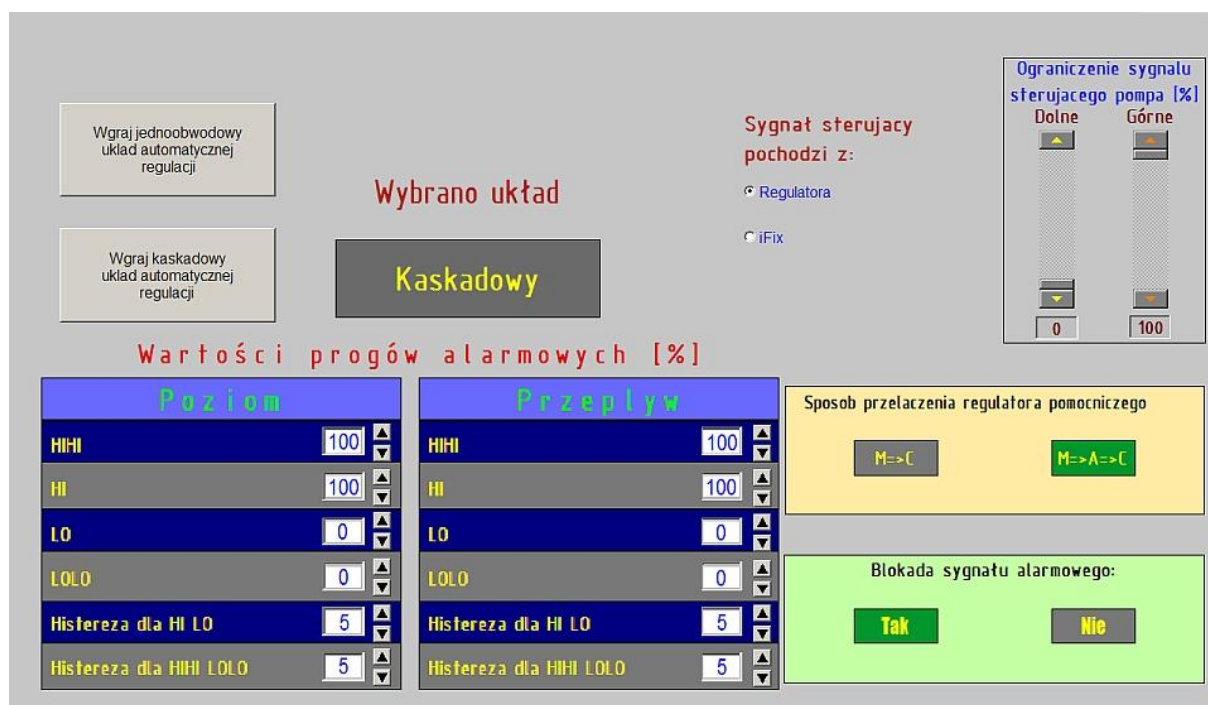
Rys. 7.18 Ekran synoptyczny – struktura funkcjonalna, w strukturze kaskadowej.

7.2.3.6 Ekran kategorii: ustawienia

Głównymi funkcjami ekranu ustawienia są:

- programowanie struktur, układów regulacji,
- informacja o wybranej strukturze,
- ograniczenie sygnału sterującego pompą (warstwa 9 - regulatora),
- blokada przełączania trybu AUTO w pomocniczym regulatorze,
- blokada dźwiękowego sygnału alarmowego regulatora,
- wyświetlanie oraz zmiana alarmów AI1 oraz AI2.

Zmiana opcji „*Sygnal sterujący pochodzi z*” powoduje ukrycie informacji o wybranym układzie regulacji oraz wymusza na użytkowniku ponowne wgranie struktury. Jeżeli wybrana jest opcja Soft Control, wtedy aplikacja uniemożliwia wgranie kaskadowego układu regulacji.



Rys. 7.19 Ekran synoptyczny – ustawienia.

7.2.4 Analiza wykonywalności

Głównym celem tego etapu jest ustalenie, w jaki sposób należy zrealizować funkcjonalność opisaną w poprzednich dwóch podrozdziałach, oraz czy technologie dostępne w środowisku projektowym iFIX są do tego celu wystarczające.

7.2.4.1 Wykonanie rysunków

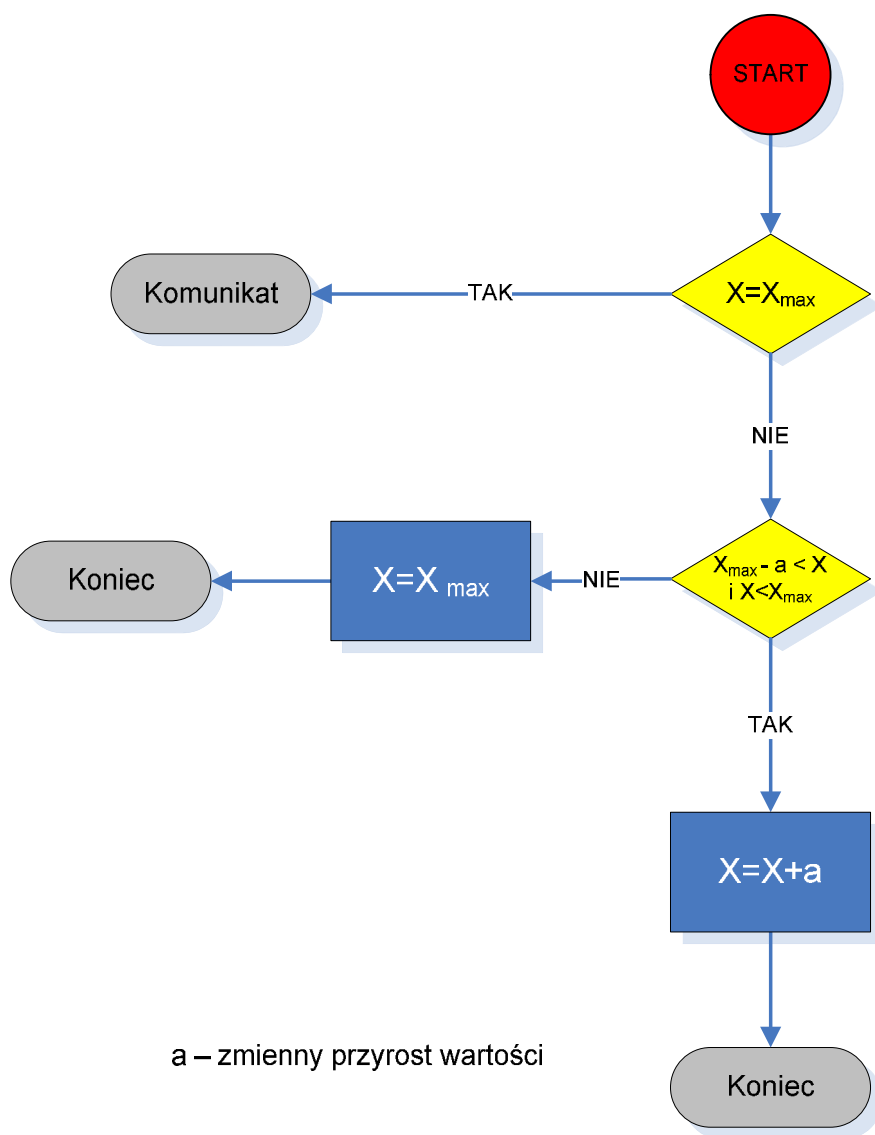
Do wykonania rysunków zostaną użyte standardowe narzędzia graficzne dostępne w programie WorkSpace (tryb konfiguracyjny). Do realizacji poszczególnych funkcji należy wykorzystać funkcje zestawione w tabeli 6.

Tabela 6 Zestawienie narzędzi systemu iFIX wykorzystanych w celu realizacji opisanych funkcji.

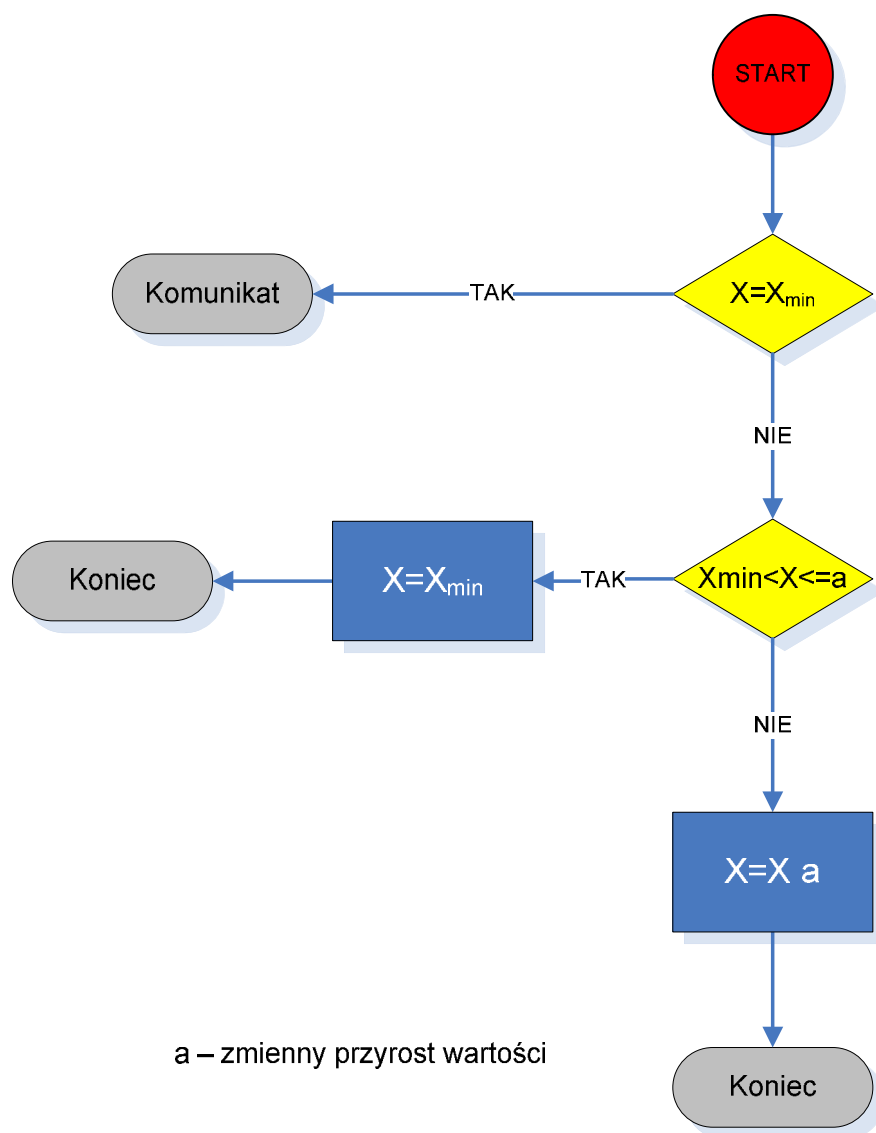
Funkcja	realizacja za pomocą
Animacja wypełniania	Ekspert wypełniania
Widzialność obiektów	Ekspert widzialności, zdarzenie, skrypt VBA
Zmiany kolorów obiektów	Ekspert koloru pierwszoplanowego
Wyświetlanie danych	Pole danych, kontrolka OCX TextBox
Wprowadzanie danych poprzez dwukrotne kliknięcie	Ekspert wprowadzania danych
Wprowadzanie danych inne	Kontrolka OCX, ComboBox, OptionBox, ScrollBar, CommandButton, skrypt VBA
Otwieranie rysunków	Przycisk + skrypt VBA opcjonalnie ekspert otwierania rysunków
Zamykanie rysunków	Przycisk + skrypt VBA, opcjonalnie ekspert zamykania rysunków
Obsługa logowania	Program Login.exe + skrypt VBA
Zestawienie alarmów	Kontrolka OCX (zestawienie alarmów)
Dodatkowa grafika (przyciski)	Bitmap
Walidacja danych	Skrypt VBA
Skalowanie rysunków	Skrypt VBA
Klawiatura ekranowa	Program MYTSOFT.EXE
Wyświetlanie daty i czasu	Narzędzie data i czas
Zamykanie aplikacji	Przycisk + Skrypt VBA

7.2.4.2 Algorytmy inkrementacji oraz dekrementacji wartości bazy danych

Zmiana wartości rekordu bazy danych, w przypadku gdy będziemy próbowali zwiększyć bądź zmniejszyć wartość poza granice jakie może przyjmować, poprzez skrypt VBA niesie za sobą niebezpieczeństwo wystąpienia błędu. Aby nie dopuścić do przerwania działania aplikacji, w opisanej sytuacji, skrypt VBA musi „wiedzieć” jakie są limity wartości, oraz nie pozwolić na dalsze zmiany, w przypadku ich przekroczenia. Poniżej przedstawiono algorytmy inkrementacji oraz dekrementacji wartości rekordów bazy danych w celu ochrony przed przerwaniem działania aplikacji.



Rys. 7.20 Algorytm inkrementacji danych zaimplementowany w aplikacji Regulacja poziomu.



Rys. 7.21 Algorytm dekrementacji, zaimplementowany w aplikacji Regulacja poziomą.

7.2.4.3 Opracowanie metody sterowania ręcznego

W regulatorze LB600, zadawanie wartości sygnału CV w trybie ręcznym, nie jest możliwe do uzyskania w sposób jawny (brak odpowiedniego rejestru). Taki stan rzeczy, w przypadku gdy dyspozytornia jest oddalona od fizycznego miejsca zainstalowania regulatora lub też instalacja jest zarządzana poprzez Internet za pośrednictwem przeglądarki WWW, powoduje, iż ręczne przesterowanie elementu wykonawczego, wymaga znacznie dłuższego czasu, bądź też jest niemożliwe.

W celu zapewnienia pełnej funkcjonalności systemu, podjęto próbę implementacji sterowania ręcznego. Po dokładnym przestudiowaniu wszystkich dostępnych funkcyj zauważono, iż pożądaną funkcję możemy osiągnąć w trojaki sposób:

- poprzez bloki warstwy 9,
- poprzez bloki warstwy 8, 3, 4, 5, 6,
- poprzez wykorzystanie bloku PID innego kanału.

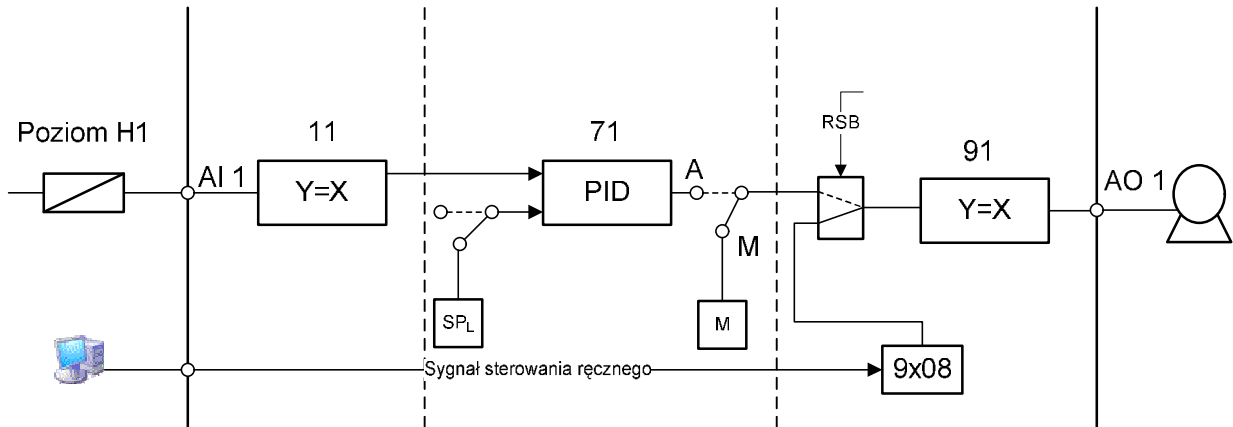
Poniżej dokładniej opisano możliwe sposoby implementacji sterowania ręcznego.

7.2.4.3.1 Sterowanie ręczne zrealizowane za pomocą adresów warstwy 9

W tym rozwiązaniu sterowanie ręczne odbywa się przy pomocy adresów 9x08 i 9x09. Zasada działania jest następująca jeżeli numer RSB określony pod adresem 9x09 przyjmie stan logiczny 1, wtedy na wyjściu, regulator ustawia, wartość określoną pod adresem 9x08. Wspomnianą wartość można zmieniać swobodnie w trakcie gdy określony RSB ma wartość 1. Niewątpliwymi plusami tego rozwiązania jest prostota wykonania oraz możliwość szybkiej implementacji w systemie SCADA. Rozwiązanie posiada dwie wady:

- Możliwość wprowadzania sygnału z dokładnością 1 % (w wielu przypadkach może być wystarczająca)
- Skokowa zmiana sygnału CV przy przełączaniu M => A

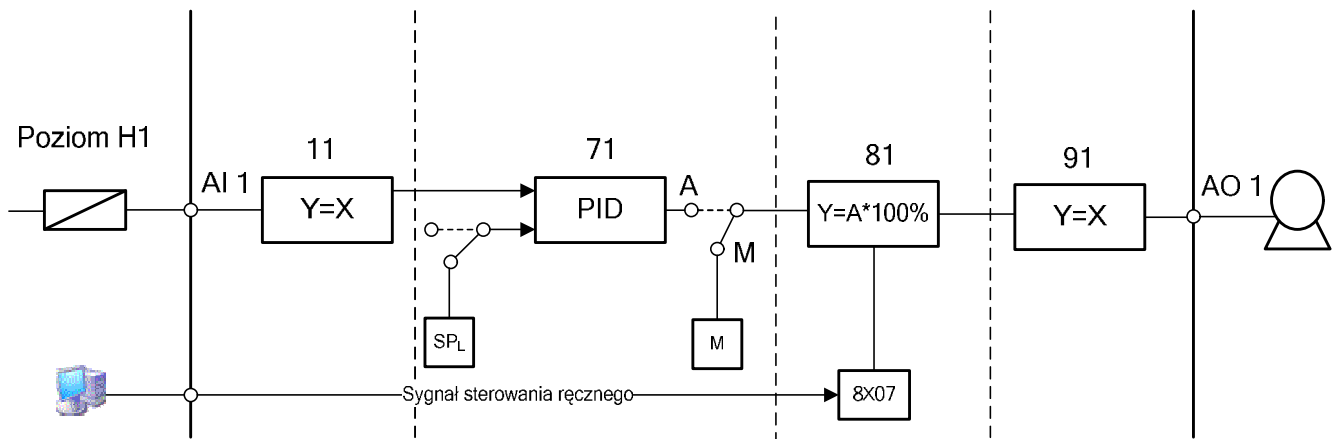
Poniżej przedstawiono realizację sterowania na schemacie funkcjonalnym regulatora.



Rys. 7.22 Przykład realizacji sterowania ręcznego przez komputer z wykorzystaniem warstwy 9.

7.2.4.3.2 Sterowanie ręczne zrealizowane za pomocą adresów warstwy 8.

Następnym sposobem implementacji sterowania ręcznego jest użycie bloków warstwy 8, bądź też 3,4,5,6. Rozwiązanie to bazuje na wykorzystaniu parametru 8x06 określającego algorytm funkcji przetwarzania sygnału. W czasie pracy w trybie automat wykorzystuje się algorytm $Y = X$, w czasie pracy w trybie ręcznym, $Y = A * 100 [\%]$. Wartość współczynnika A pod adresem 8x07 powinna być wprowadzana w skali znormalizowanej 0...1. Implementacja tego rozwiązania zajmuje więcej czasu niż, poprzednio opisanego, ale dzięki temu, zyskujemy możliwość zadawania sygnału CV z dowolną dokładnością. Sporym minusem jest tak samo jak w poprzednim rozwiązaniu, brak śledzenia, generowanego sygnału, przez wyjście regulatora przy przełączeniu $M \Rightarrow A$ (przełączenie uderzeniowe), co w niektórych przypadkach, może doprowadzić do utraty stabilności układu. Schemat funkcjonalny regulatora z proponowanym rozwiązaniem przedstawiono na poniższym rysunku.



Rys. 7.23 Przykład realizacji sterowania ręcznego przez komputer z wykorzystaniem warstwy 8.

7.2.4.3.3 Sterowanie ręczne zrealizowane za pomocą bloku PID dodatkowego kanału.

Ostatnim możliwym sposobem implementacji sterowania ręcznego jest użycie dodatkowego funkora warstwy 7, leżącego w innym torze. Rozwiązanie to, bazuje na możliwości śledzenia wartości SP regulatora pomocniczego przez wyjście regulatora głównego w trybie kaskady. W czasie pracy w trybie automat, regulator pomocniczego kanału jest nieaktywny. W czasie przełączenia regulatora głównego w tryb ręczny, regulator pomocniczy jest aktywowany i za pomocą zmiany jego wartości SP jesteśmy w stanie generować pożądany sygnał, który następnie jest przekazywany do warstwy 9. Do plusów tego rozwiązania możemy zaliczyć:

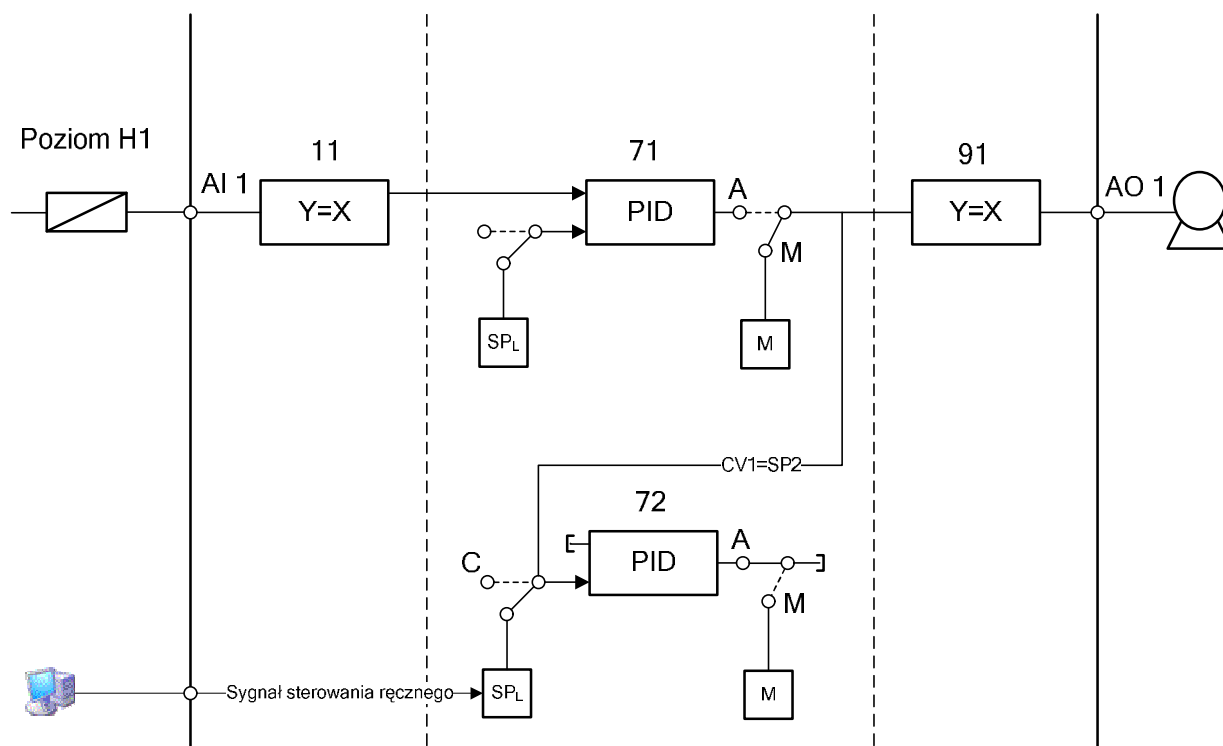
- generowanie sygnału z dowolną dokładnością,
- bezuderzeniowe przełączanie $M \Rightarrow A$.

Natomiast do minusów:

- duże nakłady pracy związane z implementacją,
- brak możliwości zbudowania pętli regulacyjnych w użytych kanałach.

Ze względu na fakt iż w projektowanym systemie, będą wykorzystane maksymalnie dwa kanały regulatora oraz na możliwość śledzenia sygnału przy przełączaniu $M \Rightarrow A$, zdecydowano się zaimplementować to rozwiązanie.

Schemat funkcjonalny regulatora z opisanym rozwiązaniem przedstawiono na poniższym rysunku.



Rys. 7.24 Przykład realizacji sterowania ręcznego przez komputer z wykorzystaniem dodatkowego bloku PID warstwy 7.

7.2.4.4 Opracowanie algorytmu obsługi samostrojzenia

Możliwość realizacji procedur samostrojzenia przez urządzeniu sterujące może być dla większości użytkowników kluczową cechą decydującą o jego zakupie. Regulator LB – 600, w tym zakresie oferuje dwie metody identyfikacji obiektu oraz pięć sposobów obliczania nastaw.

Niestety, w przeciwieństwie do większości funkcji regulatora, konstruktorzy urządzenia nie zaimplementowali odpowiednich rejestrów w celu obsługi samostrojzenia z poziomu systemu SCADA. Za pośrednictwem komputera jesteśmy w stanie jedynie zainicjować start oraz stop samostrojzenia. Do pełnej obsługi wspomnianej funkcji, poprzez system SCADA potrzebne są następujące rzeczy:

- informacja o starcie oraz zakończeniu procedury samostrojzenia,

- informacja o błędzie w przypadku niepomyślnego zakończenia procedury,
- informacja o wartości obliczonych nastawach ,
- akceptacja/odrzucanie nastaw,
- możliwość kasowania błędów.

W celu wyeliminowania powyższych trudności w projektowanym systemie, postanowiono wprowadzić dodatkowy nastawiany parametr T_{\max} , który określa maksymalny czas trwania procedury samostrojenia. Obsługa samostrojenia obejmuje trzy fazy:

1. walidacja wprowadzonych danych,
2. sprawdzenie czy procedura została uruchomiona,
3. sprawdzenie czy procedura została zakończona,

W pierwszej fazie Aplikacja sprawdza czy parametry wprowadzono zgodnie z poniższymi zasadami:

- Wybrano sposób obliczania nastaw,
- $PV_{\min} < A_{\max}$,
- $PV_{\min} \neq A_{\max}$,
- $A_{\text{stab}} > 0$,
- $\Delta CV > 0$,
- $T_{\max} \geq 60\text{s}$

W drugiej fazie aplikacja porównuje wartość sygnału CV po rozpoczęciu samostrojenia, z wartością sygnału sterującego po czasie stabilizacji. Jeżeli wartości nie różnią się, oznacza to, iż nie został wygenerowany skok CV, a samostrojenie nie zostało uruchomione w regulatorze.

W ostatniej fazie aplikacja sprawdza czy samostrojenie zostało zakończone. Realizacja tego zadania polega na tym iż, aplikacja porównuje zapamiętane podczas startu wartości nastaw z nastawami obliczonymi przez regulator. Różne nastawy są informacją, że samostrojenie zostało zakończone pomyślnie. Jeżeli regulator z jakiś powodów nie będzie mógł obliczyć nowych nastaw do czasu określonego parametrem T_{\max} wtedy aplikacja automatycznie przerwie samostrojenie i poinformuje operatora o jego niepowodzeniu.

Mimo wysiłku włożonego w przystosowanie samostrojenia do obsługi w projektowanym systemie, nie udało się rozwiązać następujących rzeczy:

- informacja o błędzie podczas samostrojania
- kasowanie obliczonych nastaw oraz błędu z pamięci regulatora

W przypadku niepowodzenia samostrojania, zaimplementowano w regulatorze prostą obsługę błędów, polegającą na wyświetlaniu wartości błędu, na panelu operatorskim. Niestety informacja o wartości błędu nie jest udostępniana poprzez rejestry.

Uruchomienie procedury samostrojania jest możliwe wtedy gdy z pamięci regulatora wykasowano błąd mogący pojawić w poprzednim rezultacie samostrojania i wykasowano (zatwierdzono) nastawy obliczone przy poprzednim samostrojeniu. Nie istnieje możliwość obsługi wspomnianych rzeczy za pośrednictwem komputera.

Następnym minusem jest brak odpowiednich rejestrów, które przechowywały by wartości obliczonych nastaw, tak jak jest to zrealizowane w przypadku parametrów obiektu. Tą niedogodność w przypadku aplikacji iFIX rozwiązano posługując się zmiennymi globalnymi dostępnymi dla całej aplikacji, jednak w przypadku użycia mniej zaawansowanego środowiska HMI/SCADA funkcjonalność porównywania nastaw poprzednich z obliczonymi byłaby niemożliwa.

Ostatnim minusem, jest skomplikowane uruchamianie procedur samostrojania, konieczność startu poprzez RSB, zmusza do zaalokowania części pakietu wejść/wyjść dyskretnych lub w przypadku jego braku, manipulacja stanami alarmowymi w innych warstwach, co też mało zaawansowanemu użytkownikowi sprawi duży problem.

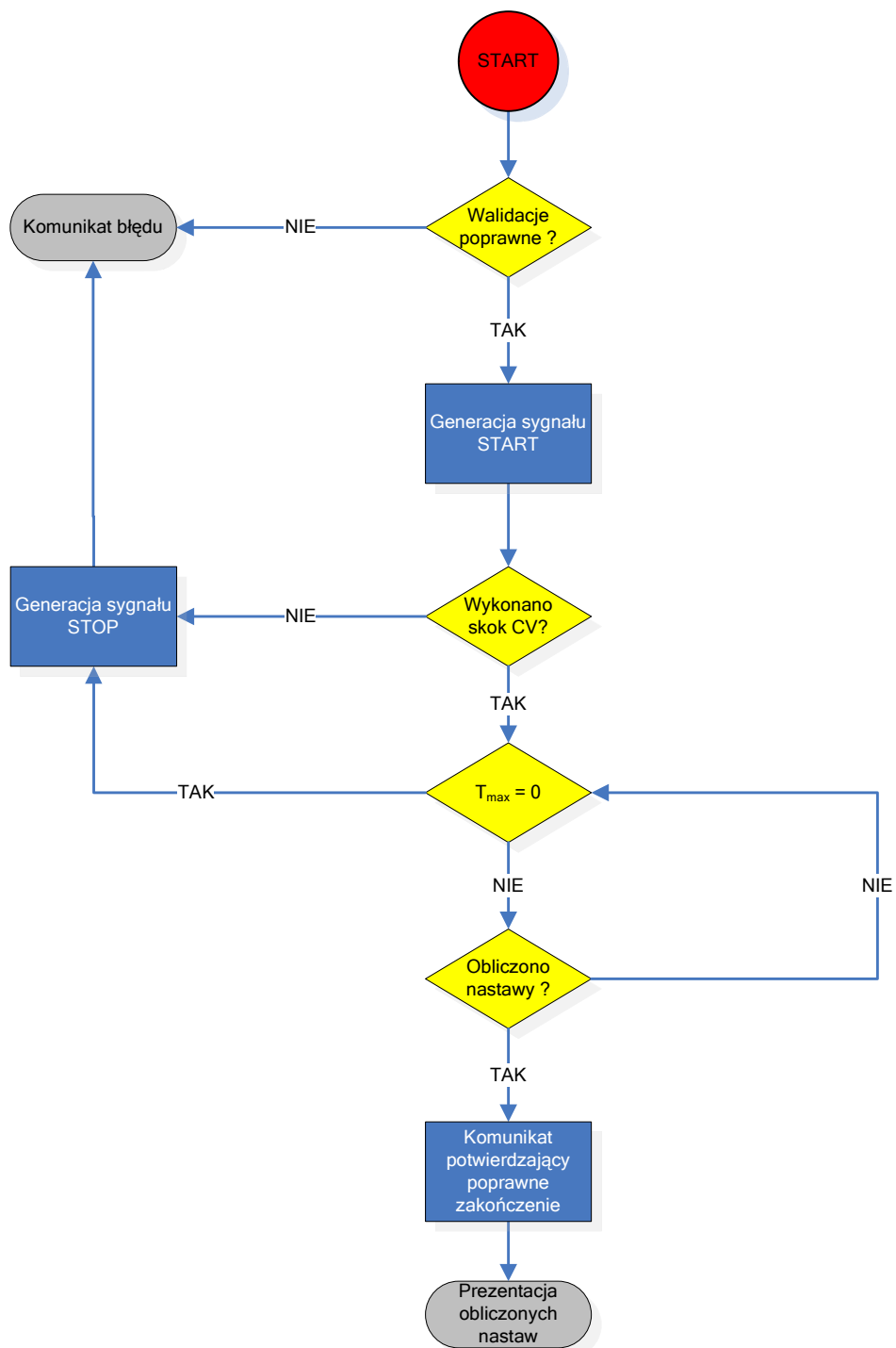
Podsumowując podczas uruchamiania samostrojania konieczna jest obecność operatora przy urządzeniu, co skutecznie uniemożliwia całkowitą zdalną kontrolę nad tą funkcją regulatora.

Do obsługi samostrojania w aplikacji, postanowiono zaimplementować komunikaty informujące operatora o aktualnym stanie procesu. Komunikaty wyświetlają się w dolnej środkowej części okna i mają następujące znaczenie:

- stabilizacja – informuje że trwa stabilizacja procesu,
- obliczam – dokonano skoku CV, trwa identyfikacja obiektu oraz obliczanie nastaw,
- OK – samostrojenie zakończone powodzeniem, dostępne nowe nastawy,
- Error – samostrojenie zakończone niepowodzeniem.
- STOP – ręcznie przerwano procedurę samostrojania.

Wymieniona wyżej obsługa samostrojzenia została zrealizowana z wykorzystaniem skryptu VBA.

Schemat algorytmu obsługi samostrojzenia uruchamianego z ręcznego trybu pracy przedstawiono na rysunku 7.25.



Rys. 7.25 Algorytm obsługi samostrojzenia zaimplementowany w aplikacji Regulacja pozioma.

7.3 Implementacja

Podczas tego etapu zostały przygotowane elementy składowe systemu takie jak:

- Procesowa Baza Danych,
- plik konfiguracyjny drajwera MB1,
- pliki rysunków.

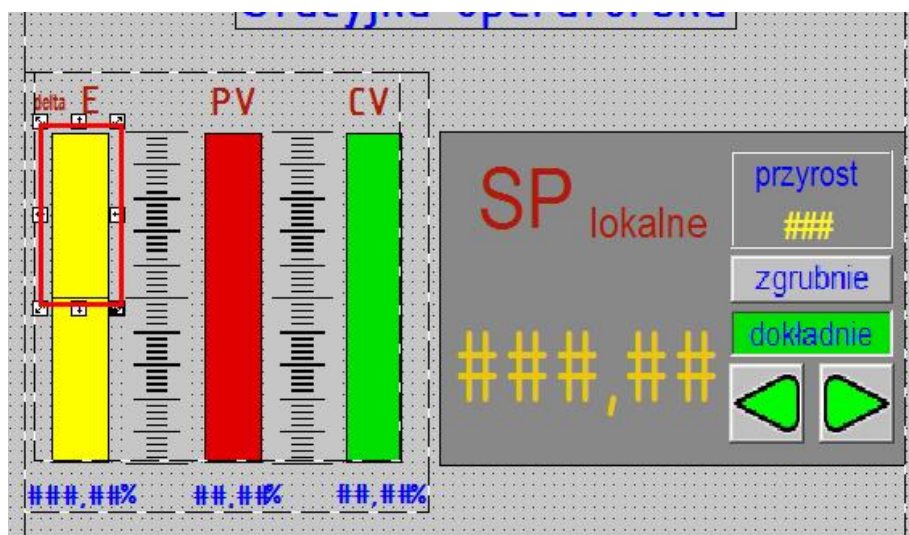
Poniższa tabela przedstawia nazwy utworzonych plików

Tabela 7 Opis plików konfiguracyjnych oraz obrazów synoptycznych.

Nazwa pliku	Opis
RegPoz.csv	Plik konfiguracyjny procesowej bazy danych w formacie CSV
MB1_LB600.csv	Plik konfiguracyjny drajwera MB1 w formacie CSV
A_Objekt.grf	Rysunek obiektu (LB600)
A_Objekt_komp.grf	Rysunek obiektu (iFIX)
A_schemBlok_jedno.grf	Rysunek schematu blokowego jednoobwodowego UAR (LB600)
A_schemBlok_jedno_komp.grf	Rysunek schematu blokowego jednoobwodowego UAR (iFIX)
A_schemBlok_kask.grf	Rysunek schematu blokowego kaskadowego UAR (LB600)
A_auto_strojenie.grf	Rysunek obsługujący samostrojenie
A_nastawy1.grf	Rysunek umożliwiający zmianę nastaw regulatora 1
A_nastawy2.grf	Rysunek umożliwiający zmianę nastaw regulatora 2
A_nastawy_komp.grf	Rysunek umożliwiający zmianę nastaw bloku regulatora w iFIX
A_realPrzyrz_jedno.grf	Rysunek schematu funkcjonalnego regulatora w jednoobwodowym UAR (LB600)
A_realprzyrzJednoKomp.grf	Rysunek schematu funkcjonalnego regulatora w jednoobwodowym UAR (iFIX)
A_realPrzyrz_kas.grf	Rysunek schematu funkcjonalnego regulatora w kaskadowym UAR (LB600)
A_trendy.grf	Rysunek wyświetlający wykresy
A_ust.grf	Rysunek umożliwiający zmianę ustawień
A_WspPID.grf	Rysunek umożliwiający zmianę parametrów algorytmu PID realizowanego przez iFIX
A_Help.grf	Rysunek przedstawiający pomoc do samostrojenia
A_listwa_dolna.grf	Rysunek Listy dolnej
A_listwa_gorna.grf	Rysunek Listy górnej

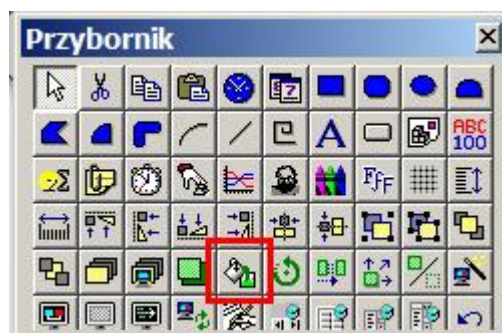
7.3.1 Połączenie obiektów animowanych na ekranach z bazą danych

Łączenie obiektów animowanych, takich jak: rurociągi, wskaźniki poszczególnych wartości, sygnalizacja włączenia funkcji, z procesową bazą danych. Odbywa się w trybie konfiguracyjnym programu WorkSpace, z wykorzystaniem dostępnych ekspertów. Wykonanie animacji może być przeprowadzone również z wykorzystaniem skryptu VBA, jednak ten sposób, wymaga dość dobrej znajomości programowania. Poniżej omówiono sposób połączenia obiektu za pomocą eksperta wypełnienia, na przykładzie części wskaźnika procentowego, wyświetlającego odchyłkę (obiekt zaznaczony kolorem czerwonym).



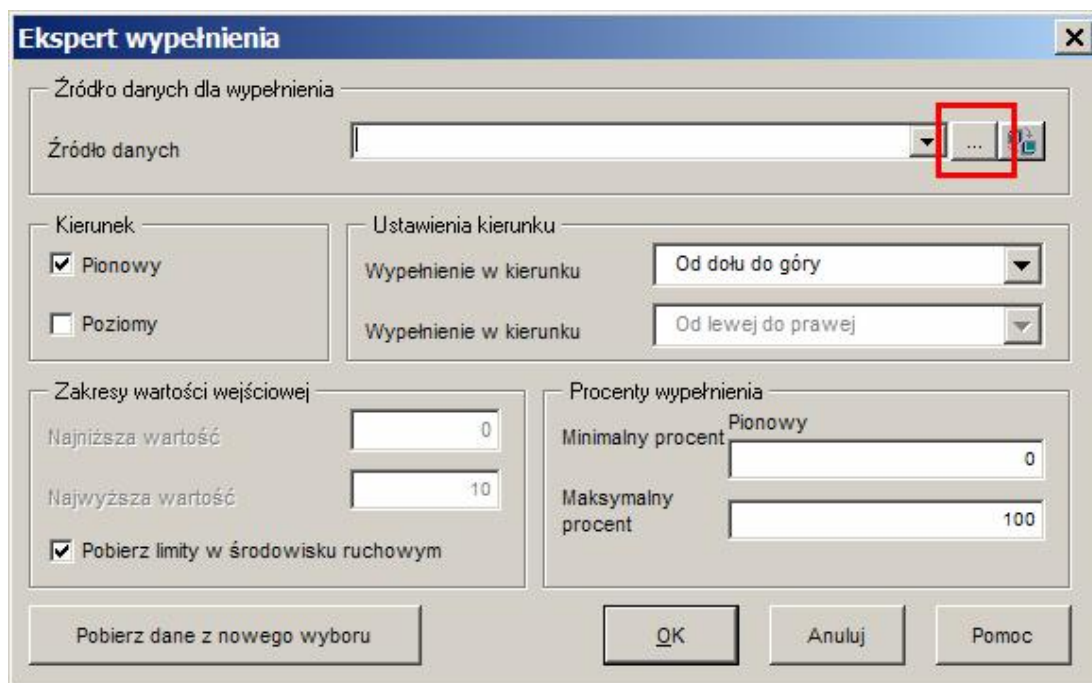
Rys. 7.26 Część animowanego barografu w stacyjce operatorskiej.

W celu dokonania animacji należy, wybrać animowany obiekt, a następnie z przybornika wybrać eksperta wypełnienia procentowego.



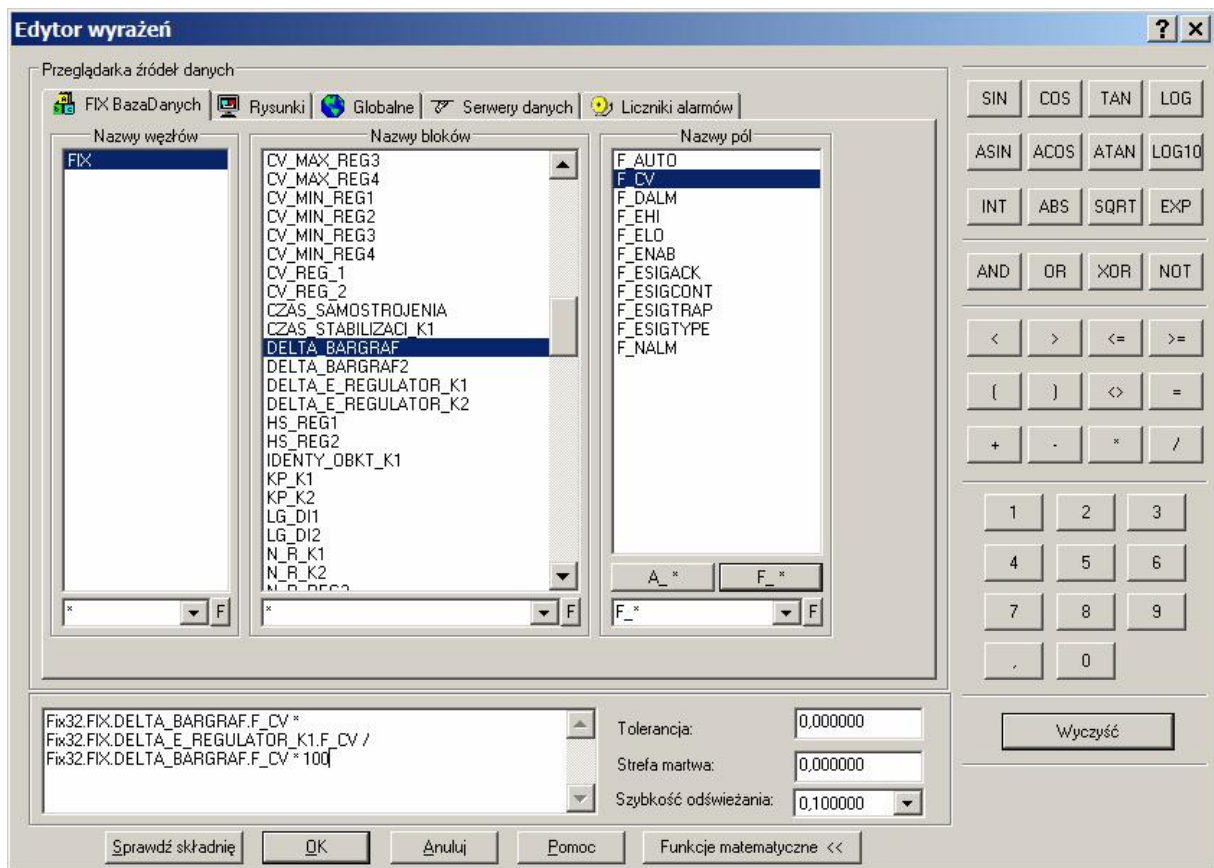
Rys. 7.27 Przybornik programu WorkSpace

Następnie na formatce przedstawiającej eksperta wypełnienia należy wskazać źródło w bazie danych, odpowiadające za animacje, poprzez wybranie przycisku zaznaczonego na rysunku 7.28.



Rys. 7.28 Formatka konfiguracyjna narzędzia – ekspert wypełnienia.

Program WorkSpace wyświetla „Edytora wyrażeń” rys 7.29, w którym w najprostszym przypadku należy wskazać rekord bazy danych, oraz jego pole, które ma być animowane za pomocą obiektu, w trybie wykonywania. Edytor wyrażeń oprócz „linkowania” obiektu z bazą danych, umożliwia wyliczenie wartości, która za pośrednictwem obiektu, ma być animowana. Dzięki temu narzędziu, możemy przedstawiać dane w pożądanym zakresie, bez angażowania w to zadanie bloków skalujących i obliczeniowych w bazie danych, a przez to znacznie uprościć proces tworzenia aplikacji. Na poniższym rysunku, przedstawione jest wyrażenie obliczające dodatnią część odchyłki. Warto zauważyć, iż animacja wyświetlana na rysunku jest uzależniona od dwóch rekordów w bazie danych.



Rys. 7.29 Widok edytora wyrażeń.

Po wybraniu rekordu bazy danych i/lub zbudowaniu odpowiedniego wyrażenia, należy wybrać przycisk OK, zapisujący dane oraz zamykający edytora wyrażeń. Program Workspace powróci do formatki eksperta wypełnienia, w którym według uznania należy wypełnić pola w sekcjach: „kierunek”, „ustawienia kierunku”, „Zakresy wartości wejściowej”, „Procent wypełnienia”. Ostatecznie w celu zapisania parametrów animacji należy wybrać przycisk OK znajdujący się na formatce eksperta wypełnienia.

8 Analiza porównawcza algorytmów PID realizowanych przez regulator LB-600 oraz system iFIX.

Najbardziej popularnym algorytmem realizującym sterowanie w oparciu o sprzężenie zwrotne jest algorytm PID (rów. 8.1). Algorytm ten, zapewnia znakomitą jakość sterowania niezależnie od dynamicznej charakterystyki procesu. Algorytm PID składa się z trzech części:

1. P – część proporcjonalna,
2. I – część całkująca,
3. D – część różniczkująca

Dostosowanie algorytmu do danego obiektu odbywa się poprzez wprowadzenie tzw. nastaw, które określają intensywność działania jego poszczególnych części. Tak więc nastawami dla części P, I, oraz D są odpowiednio: wzmocnienie proporcjonalne (K_p), czas zdwojenia (T_i) oraz czas wyprzedzenia (T_d). Przez lata badań oraz stosowania w technice automatycznej regulacji, zostało wypracowanych wiele metod pozwalających na prawidłowe dopasowanie nastaw algorytmu PID do obiektu regulacji.

8.1 Idealny oraz rzeczywisty algorytm PID

Sygnał wyjściowy klasycznego regulatora PID z idealnym różniczkowaniem, jest określony równaniem:

$$G(s) = K_p \cdot \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (8.1)$$

W wyniku idealnego działania różniczkującego powstaje tzw. impuls Diraca, który w rzeczywistości nie jest przydatny w technice, ponieważ żadne urządzenie wykonawcze nie jest w stanie na niego zareagować. Z tego też powodu, część różniczkująca w rzeczywistym regulatorze PID zawiera człon inercyjny I rzędu, a wartość sygnału wyjściowego regulatora jest opisana równaniem:

$$G(s) = K_p \cdot \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\tau s + 1}\right) \quad (8.2)$$

gdzie: $\tau = \frac{T_d}{k_d}$ – stała czasowa członu inercyjnego,

k_d – współczynnik wzmocnienia dynamicznego (stała przyrzędu)

8.2 Cyfrowe postacie algorytmu PID.

Algorytm PID, dotychczas był realizowany przez regulatory analogowe. Wraz z nadejściem techniki cyfrowej, powstała potrzeba przełożenia, postaci ciągłej algorytmu PID na równoważną postać cyfrową, która za pomocą obliczeń numerycznych, pozwala obliczać wartość sygnału sterującego.

Sygnał wyjściowy idealnego regulatora PID, w postaci równania ruchu jest opisany równaniem 8.3

$$CV(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (8.3)$$

gdzie: e – odchyłka regulacji (PV-SP)

Stosując proste aproksymacje

$$\frac{de}{dt} = \frac{(e_n - e_{n-1})}{T} \quad \text{oraz} \quad \int edt = \sum_{k=0}^n e_k T \quad (8.4)$$

gdzie: n chwila próbkowania,

k=0,1,2,...n,

T= okres próbkowania,

Wzór 8.3 możemy przekształcić do następującej postaci cyfrowej

$$CV = K_p \left[e_k + \frac{T}{T_i} s_k + \frac{T_d}{T} (e_k + e_{k-1}) \right] \quad (8.5)$$

gdzie: $s_k = s_{k-1} + e_k$ (sumowanie całkujące)

Ostatecznie mnożąc wszystkie wyrażenia w nawiasie kwadratowym przez wzmocnienie proporcjonalne, otrzymujemy tzw. **pozycyjny algorytm PID**. Pozycyjny algorytm PID ma tę wadę, iż wymaga podania punktu pracy (dodanie do sygnału wyjściowego odpowiedniej

wartości ustawionego punktu pracy). Swobodną zmianę wartości zadanej oraz łagodne przełączanie trybu pracy zapewnia **prędkościowy algorytm PID** (przyrostowy), który oblicza zmianę wartości wyjściowej dla każdej próbki [6]. Algorytm przyrostowy otrzymujemy poprzez zróżniczkowanie równania 8.3.

$$\frac{dCV(t)}{dt} = K_p \left[\frac{de(t)}{dt} + \frac{T}{T_i} e(t) + \frac{T_d}{T} \cdot \frac{d^2e(t)}{dt^2} \right] \quad (8.6)$$

Ostatecznie przyrost sygnału sterowania, w postaci cyfrowej jest wyrażony następująco:

$$\Delta CV = CV_n - CV_{n-1} = K_p \left[(e_n - e_{n-1}) + \frac{T}{T_i} e_n + \frac{T_d}{T} (e_n - 2e_{n-1} + e_{n-2}) \right] \quad (8.7)$$

Powyższa postać algorytmu, powoduje iż, wraz ze zmianą wartości zadanej SP, generowany jest gwałtowny skok, sygnału sterującego CV. W celu wyeliminowania tej niedogodności, algorytm prędkościowy realizuje się na sygnale wartości mierzonej PV (równanie 8.8). Ta modyfikacja zapewnia „łagodną” zmianę punktu pracy układu regulacji.

$$\Delta CV = K_p \left[(PV_{n-1} - PV_n) + \frac{T}{T_i} (SP - PV_n) + \frac{T_d}{T} (2PV_{n-1} - PV_{n-2} - PV_n) \right] \quad (8.8)$$

Należy zauważyć, iż wartość zadana występuje tylko w członie całkującym, dlatego należy zadbać o to aby spełnione były warunki $T_i > 0$ oraz $K_p > 0$.

8.3 Algorytm PID, regulatora LB-600

Poszczególne składowe algorytmu PID w regulatorze LB-600 zostały zrealizowane na sygnale odchyłki. Postać cyfrowa algorytmu jest przedstawiona poniżej:

$$\Delta CV = K_p \left[(e_n - e_{n-1}) + \frac{T}{T_i} e_n + D_n \right] \quad (8.9)$$

gdzie: ΔCV – przyrost sygnału sterującego,

T_i – czas zdwojenia,

e – odchyłka regulacji PV-SP,

D_n – wartość liczbowa części różniczkującej w n-tym okresie próbkowania

$$D_n = \frac{T_d}{T + \alpha T_d} [(e_n - 2e_{n-1} + e_{n-2}) + \alpha D_{n-1}] \quad (8.10)$$

gdzie: α – odwrotność współczynnika wzmocnienia dynamicznego k_d

W przypadku regulatora LB-600, $k_d = 8$.

8.4 Algorytm PID, systemu iFIX4.0

Algorytm bloku PID, systemu iFIX4.0 przedstawia się następująco:

$$\Delta CV = K_p \cdot \beta (E_n - E_{n-1}) + \frac{T}{T_i} (F_n - CV_{n-1}) + \frac{TK_p E_n}{T_i} + \frac{K_p \cdot T_D \cdot \gamma}{6 \cdot T \cdot (T_d \cdot \alpha + 1)} (E_n + 3E_{n-1} - 3E_{n-2} - E_{n-3}) \quad (8.11)$$

gdzie: F_n – wartość sygnału wewnętrznego sprzężenia zwrotnego,

α – stała czasowa członu inercyjnego, działania różniczkującego (0 – 0,125),

β – współczynnik udziału działania proporcjonalnego (0 – 1),

γ – współczynnik udziału działania różniczkującego (0 – 1).

Podobnie jak w algorytmie zaimplementowanym w regulatorze LB-600, algorytm PID systemu iFIX jest zrealizowany na sygnale odchyłki. Pierwszą widoczną różnicą jest to iż pochodna odchyłki jest obliczana metodą czteropunktowych różnic centralnych:

$$\frac{de}{dt} = \frac{\Delta e}{T} = \frac{1}{6T} [e_n + 3e_{n-1} - 3e_{n-2} - e_{n-3}] \quad (8.12)$$

Zastosowanie uśredniania wartości z kilku próbek, pozwala ograniczyć wpływ zakłóceń oraz błędów [6]. Dodatkowo w algorytmie, występuje opcjonalne sprzężenie zwrotne które, powoduje dodanie wartości obliczonej zgodnie ze wzorem 8.13, w przypadku gdy sygnał wyjściowy regulatora nie jest równy sygnałowi urządzenia wykonawczego. Zabezpieczenie ma na celu nie dopuścić do nasycenia sygnału sterującego w przypadku uszkodzenia lub braku komunikacji z urządzeniem wykonawczym.

$$\frac{T}{T_i}(F_n - CV_{n-1}) \quad (8.13)$$

8.5 Odpowiedzi skokowe regulatorów LB-600 oraz Soft Control

Analiza porównawcza algorytmów PID, została przeprowadzona na podstawie zarejestrowanych odpowiedzi na wymuszenie skokowe (skok wartości zadanej SP). Głównym celem badania było sprawdzenie poprawności odpowiedzi algorytmów: P, PI oraz PID. Podczas badania korzystano ze wspólnych nastaw przedstawionych tabeli 8. Dla regulatora Soft Control czas próbkowania wynosił 100ms.

Tabela 8 Nastawy regulatorów LB-600 oraz Soft Control podczas wyznaczania odpowiedzi na wymuszenie skokowe

	LB-600	Soft Control
K_p	2	2
T_i	60 [s]	1 [min]
T_d	30 [s]	0,5 [min]

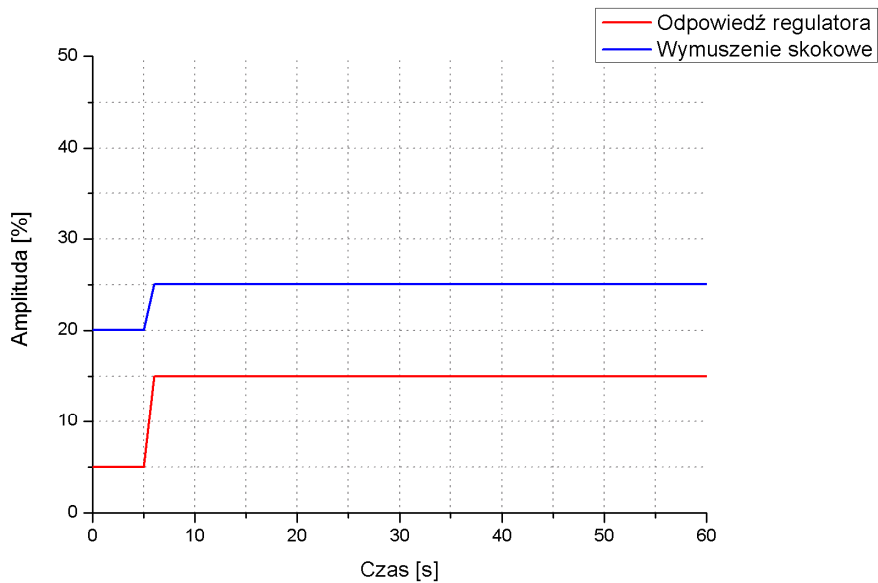
8.5.1 Algorytm P

Algorytm P należy do tzw. algorytmów statycznych tzn. w stanie ustalonym istnieje jednoznaczny związek między sygnałem wyjściowym u regulatora a odchyłką regulacji e . Sygnał wyjściowy regulatora P opisany jest równaniem 8.14

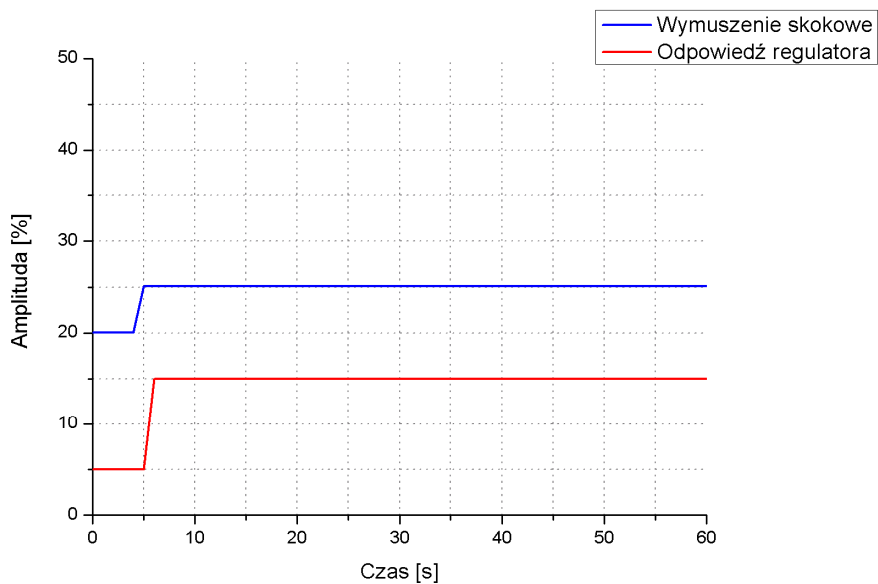
$$u(t) = K_p e(t) + u(0) \quad (8.14)$$

gdzie: $u(0)$ – wartość sygnału wyjściowego dla $e = 0$

Jak widać na poniższych rysunkach, odpowiedzi obu regulatorów są poprawne.



Rys. 8.1 Odpowiedź regulatora LB-600, algorytm P; na wymuszenie skokowe wartości zadanej SP o amplitudzie 5%.



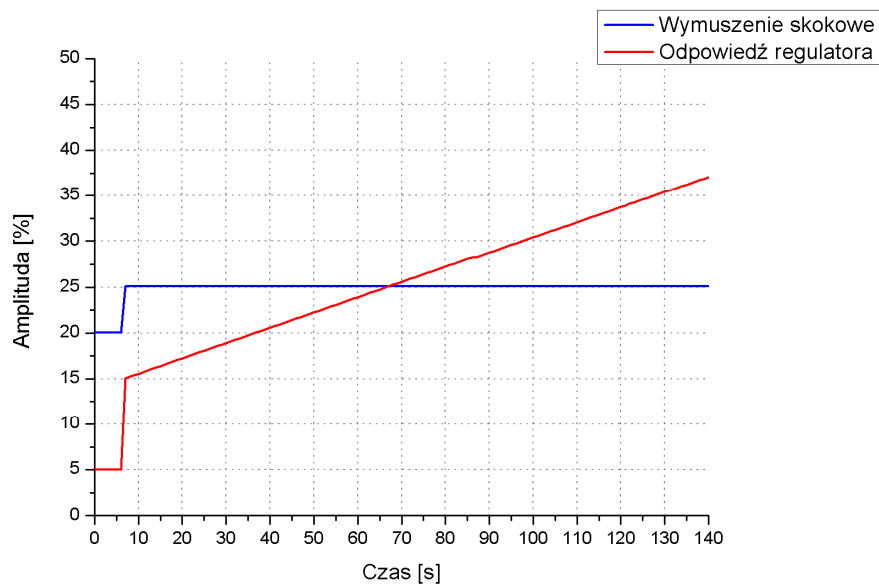
Rys. 8.2 Odpowiedź regulatora Soft Control, algorytm P; na wymuszenie skokowe wartości zadanej SP o amplitudzie 5%.

8.5.2 Algorytm PI

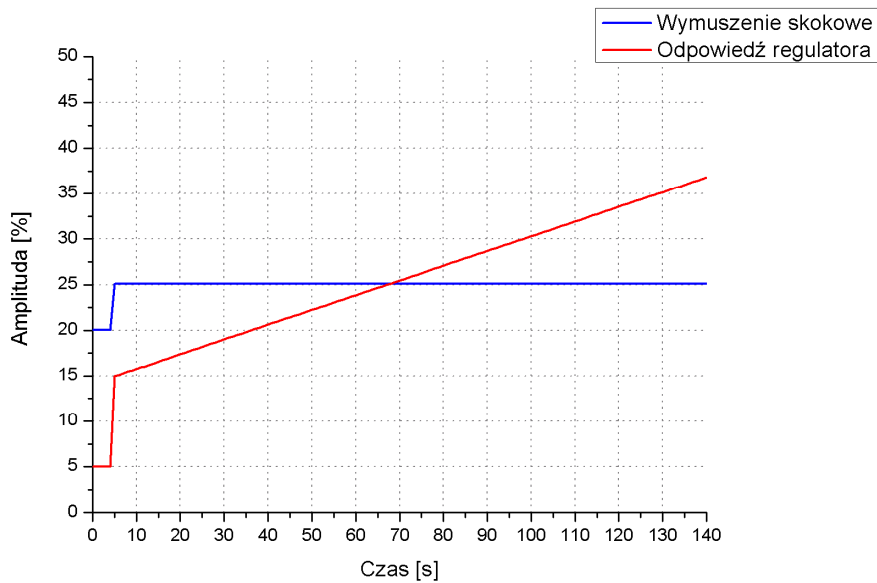
Sygnal wyjściowy regulatora o algorytmie PI opisuje równanie:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + u(0) \quad (8.15)$$

Sygnal wyjściowy regulatora o algorytmie PI jest więc sumą, odpowiedzi działania proporcjonalnego **P** oraz całkującego **I**. Składowa całkowa wraz z upływem czasu rośnie od wartości początkowej (równej zero dla $t=0$), a po czasie $t=T_i$ składowa ta osiąga wartość równą składowej proporcjonalnej, co w tym przypadku oznacza podwojenie wartości sygnału wyjściowego w stosunku do składowej proporcjonalnej. Jak widać na poniższych rysunkach odpowiedzi regulatorów LB-600 oraz Soft Control są poprawne.



Rys. 8.3 Odpowiedź regulatora LB-600, algorytm PI; na wymuszenie skokowe wartości zadanej SP o amplitudzie 5%.



Rys. 8.4 Odpowiedź regulatora Soft Control, algorytm PI; na wymuszenie skokowe wartości zadanej SP o amplitudzie 5%.

8.5.3 Algorytm PID

Odpowiedź skokowa rzeczywistego regulatora PID dla zmian odchyłki regulacji $e(t) = e_0 1(t)$ opisana jest równaniem 8.16.

$$u(t) = K_p e_0 + \frac{K_p}{T_i} e_0 t + K_p e_0 \frac{T_d}{\tau} e^{-\frac{t}{\tau}} \quad (8.16)$$

gdzie: e_0 – odchyłka w chwili wystąpienia zakłócenia skokowego,

τ - stała czasowa, inercji działania różniczkującego, (wartość zmienna),

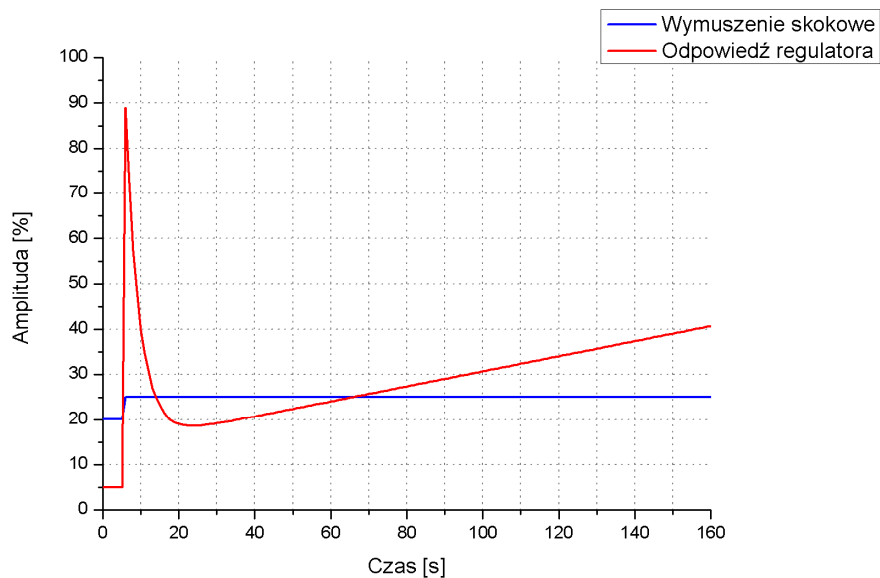
$\frac{T_d}{\tau}$ - współczynnik wzmocnienia dynamicznego oznaczany jako k_d (wartość stała).

Sygnal sterujący CV, regulatora o algorytmie PID w chwili wystąpienia zakłócenia osiąga wartość wyrażoną zależnością 8.17.

$$K_p e_0 (1 + k_d) + CV(0) \quad (8.17)$$

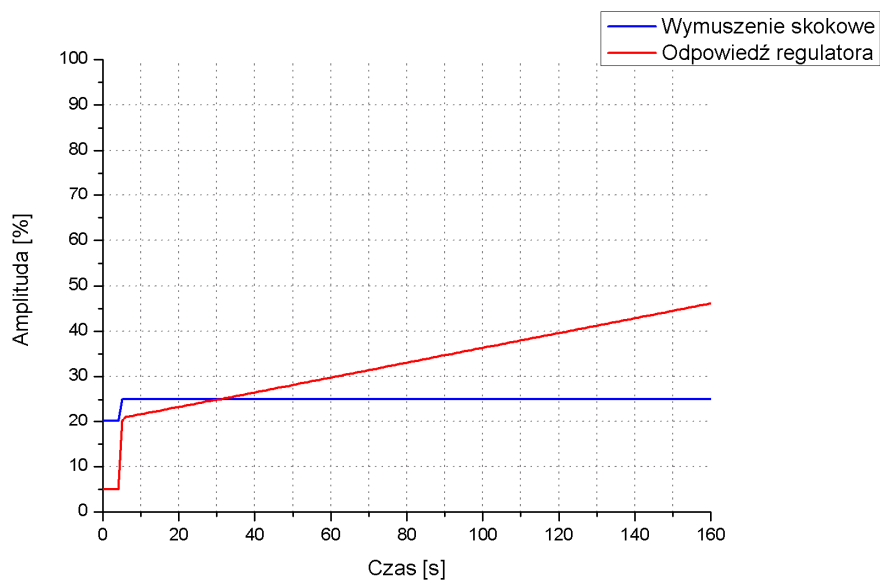
gdzie: $CV(0)$ wartość sygnału sterującego, w stanie ustalonym $SP=PV$,

która, w przypadku przeprowadzonego badania, powinna wynosić 95%. Poniższe rysunki przedstawiają odpowiedzi regulatorów PID uzyskanych podczas badania.



Rys. 8.5 Odpowiedź regulatora LB-600, algorytm PID; na wymuszenie skokowe wartości zadanej SP o amplitudzie 5%

Jak można zauważyć na powyższym wykresie, wartość amplitudy odpowiedzi regulatora LB-600 o algorytmie PID, w chwili wystąpienia zakłócenia jest ok. 5% niższa od spodziewanej.

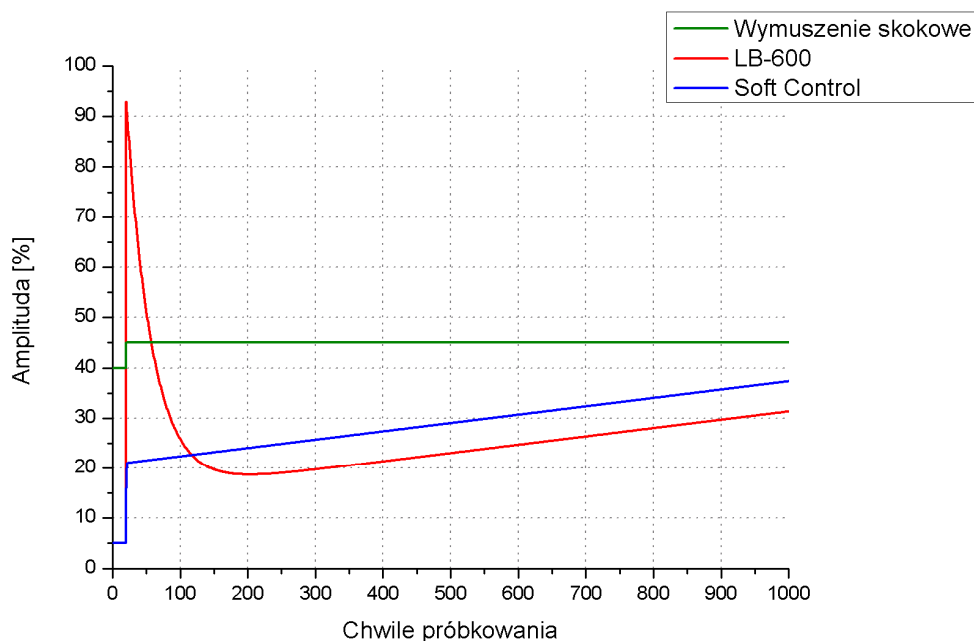


Rys. 8.6 Odpowiedź regulatora Soft Control, algorytm PID; na wymuszenie skokowe wartości zadanej SP o amplitudzie 5%

Odpowiedź regulatora Soft Control o algorytmie PID, przy współczynniku $\gamma = 1$, miała charakter odpowiedzi skokowej regulatora P z dużym wzmocnieniem ($K_p > 20$). Dlatego postanowiono zmniejszyć udział działania różniczkującego, przyjmując $\gamma = 0,002$. Dla nowych parametrów, przebieg odpowiedzi jest zbliżony do przebiegu odpowiedzi regulatora o algorytmie PI.

Ponieważ odpowiedź na wymuszenie skokowe regulatora Soft Control nie była satysfakcjonująca, podejrzewano, że człon inercyjny części różniczkującej, ujęty w równaniu 8.11 nie spełnia swojej roli, a oprogramowanie rejestrujące nie jest w stanie uchwycić maksymalnej odpowiedzi regulatora.

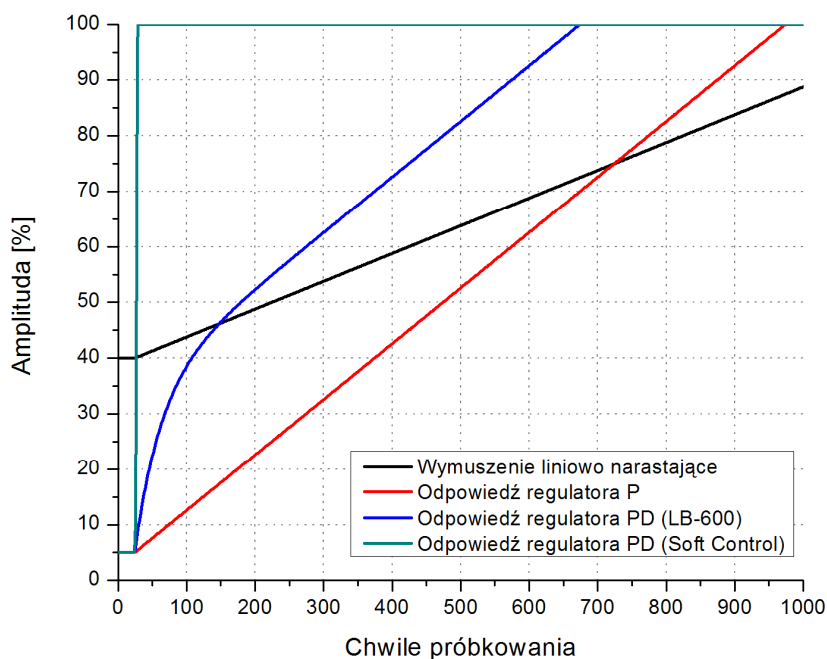
W celu dokładnego zbadania części różniczkującej obu regulatorów, w programie Ms Excel, przeprowadzono obliczenia bazujące na równaniach 8.9 i 8.11. Do obliczeń przyjęto czas próbkowania wynoszący 1/10 sekundy. Obliczone odpowiedzi obu regulatorów przedstawione są na rysunku 8.7.



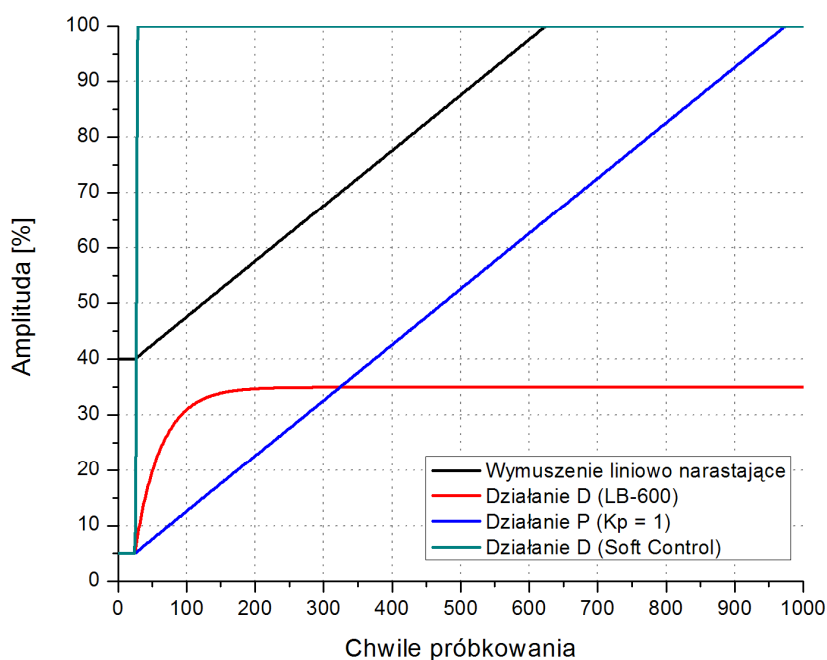
Rys. 8.7 Obliczone odpowiedzi regulatorów LB-600 oraz Soft Control na wymuszenie skokowe

Jak widać na powyższym rysunku, zarejestrowane wcześniej odpowiedzi były poprawne. Analizując odpowiedzi algorytmu Soft Control stwierdzono iż część różniczkująca nie działa poprawnie.

Dalsze badanie członu różniczkującego, polegało na zweryfikowaniu, czy czas T_d , wyrażający intensywność działania różniczkującego, jest zgodny z definicją przedstawioną w [8].

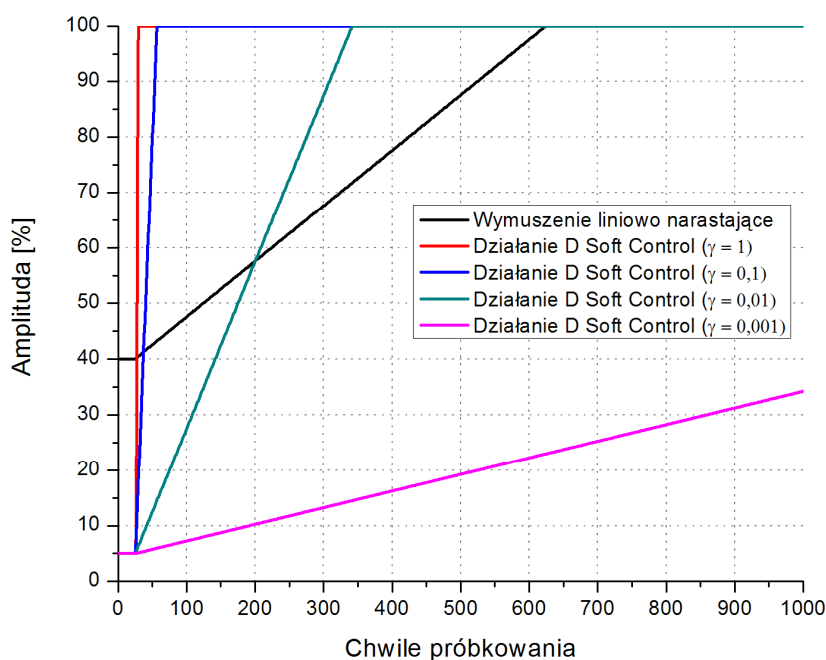


Rys. 8.8 Obliczone odpowiedzi regulatorów LB-600 oraz Soft Control o algorytmie PD, na wymuszenie liniowo narastające



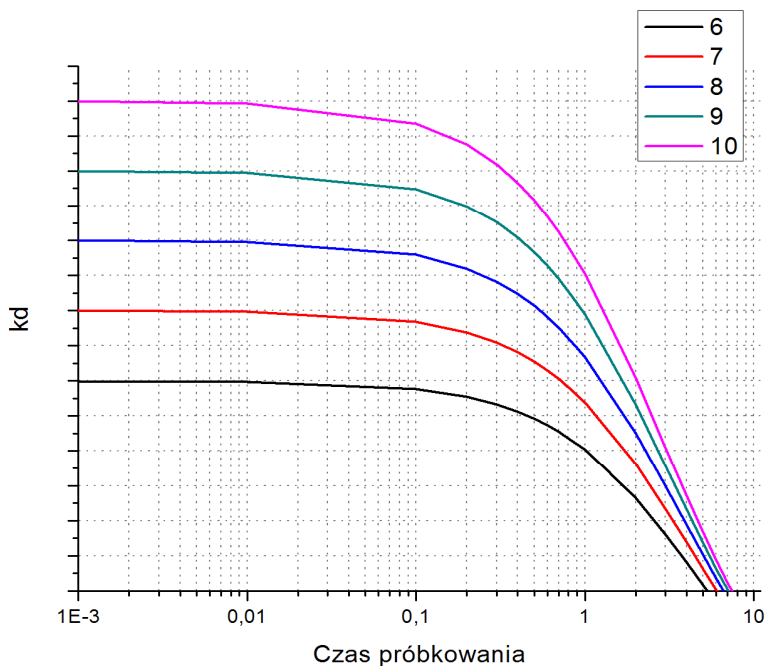
Rys. 8.9 Obliczone odpowiedzi składowych D, regulatorów LB-600 oraz Soft Control, na wymuszenie liniowo narastające

Jak widać na powyższych rysunkach, odpowiedź członu różniczkującego regulatora LB-600 jest poprawna - $T_d = 30s$ (Okres próbkowania * 300). W regulatorze Soft Control nie zaobserwowano żadnego działania różniczkującego. Zmiana wartości współczynnika α , ujętego w równaniu 8.11 nie wpływa w żaden znaczący sposób na przebieg odpowiedzi regulatora. Natomiast wpływ współczynnika γ , na przebieg odpowiedzi jest pokazany poniżej.



Rys. 8.10 Obliczona odpowiedź składowej D, regulatora Soft Control, na wymuszenie liniowo narastające, dla różnych współczynników γ ,

Podczas przeprowadzonych obliczeń stwierdzono, iż przebieg odpowiedzi części różniczkującej jest mocno związany z czasem próbkowania. Z przedstawionej na rysunku 8.11, zależności wynika, że minimalny czas próbkowania dla danego współczynnika wzmocnienia dynamicznego, zapewniający satysfakcjonującą odpowiedź członu różniczkującego wynosi $1/100$ s. Czas próbkowania wynoszący 10ms, w wielu przypadkach, z przyczyn technicznych może być nieosiągalny. W celu ograniczenia negatywnego (zbyt długiego) czasu próbkowania, na działanie różniczkujące, można odpowiednio dobrać współczynnik k_d . Przykład, zakładamy iż dobraliśmy współczynnik $k_d = 6$, oraz to, iż powodu przepustowości łącza danych nie jest możliwe osiągnięcie czasu próbkowania 10ms. Rozwiązaniem może być zastosowanie $k_d=7$ oraz czasu próbkowania wynoszącego 0,7s. Patrząc odwrotnie, jeżeli oszacujemy iż, maksymalny stabilny okres próbkowania będzie wynosił 0,3s wtedy możemy, względem czasu próbkowania obliczyć nową wartość k_d , w naszym przykładzie wynosząca 6,4.

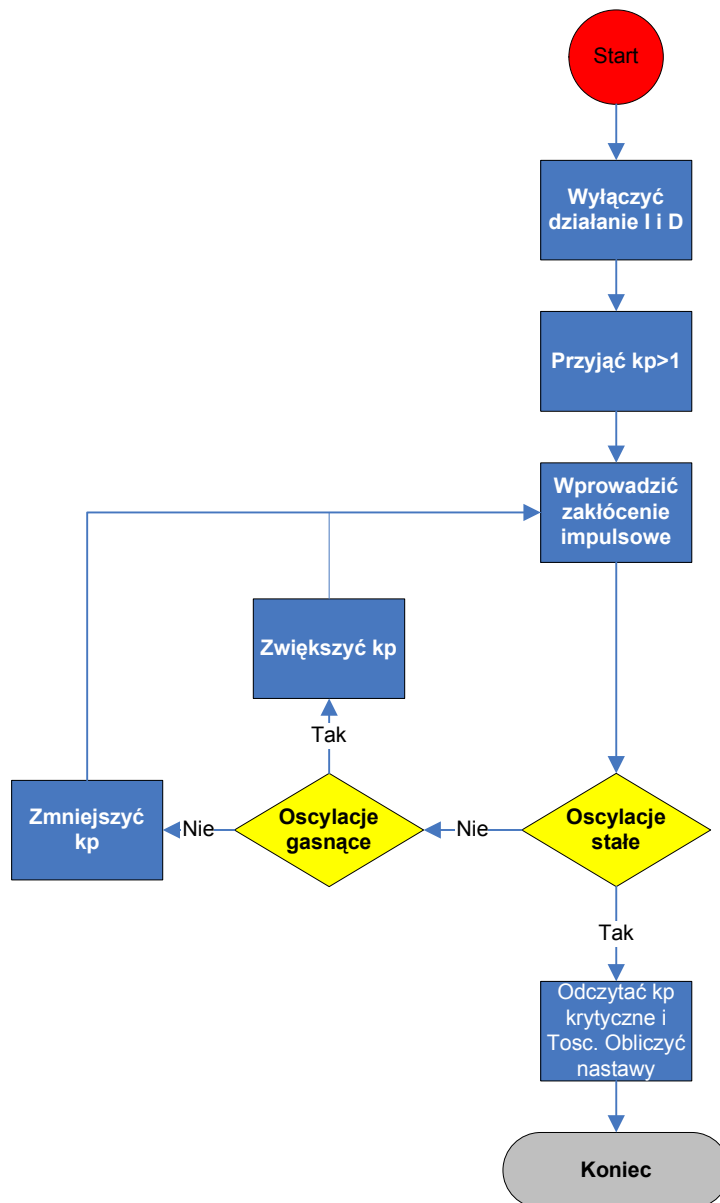


Rys. 8.11 Zależność współczynnika k_d od czasu próbkowania, cyfrowego algorytmu PID

8.6 Przebiegi przejściowe UAR z regulatorami LB-600 oraz Soft Control.

8.6.1 Identyfikacja obiektu regulacji oraz dobór nastaw regulatora, metodą cyklu granicznego (Zieglera-Nicholsa)

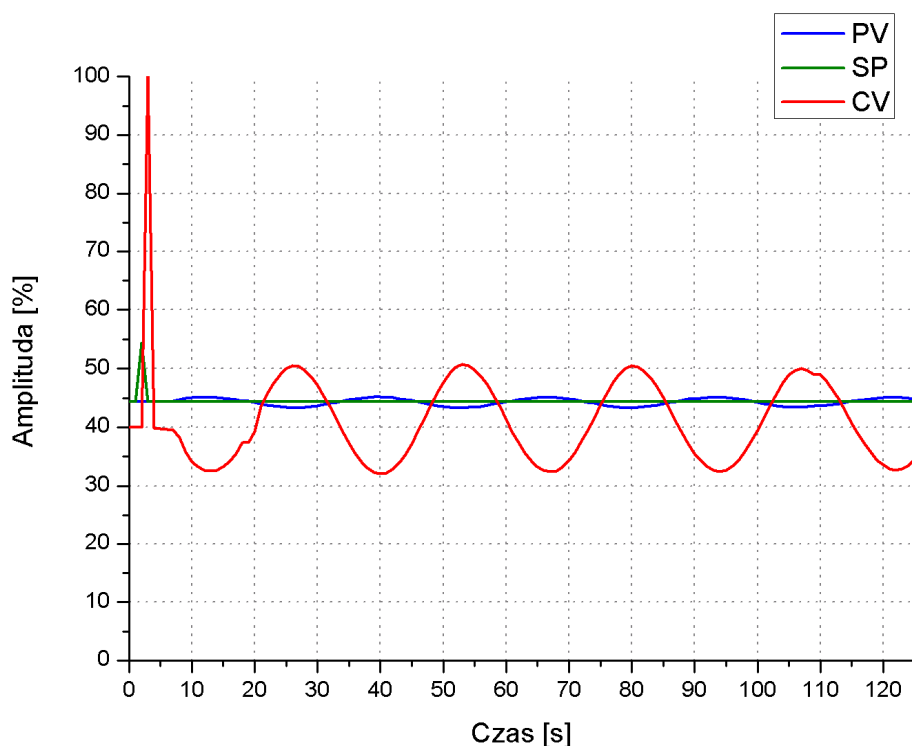
Identyfikacja obiektu regulacji metodą cyklu granicznego, jest metodą eksperymentalną. Polega na znalezieniu wzmocnienia krytycznego, przy którym obiekt regulacji znajduje się na granicy stabilności, oraz przebieg wielkości regulowanej i sygnału sterującego ma charakter stałych oscylacji. Na podstawie określonego wzmocnienia krytycznego oraz wzorów Zieglera-Nicholsa, przedstawionych w tabeli 9, oblicza się nastawy regulatora. Na poniższym rysunku przedstawiono procedurę wyznaczania wzmocnienia krytycznego.



Rys. 8.12 Algorytm wyznaczania wzmocnienia krytycznego w metodzie cyklu granicznego

Tabela 9 Wzory Zieglera-Nicholsa

	Kp	Ti	Td
P	$0,5K_{pkr}$	-	-
PI	$0,45 K_{pkr}$	$0,85 T_{osc}$	-
PID	$0,6 K_{pkr}$	$0,5 T_{osc}$	$0,12 T_{osc}$



Rys. 8.13 Odpowiedź układu automatycznej regulacji ze wzmocnieniem krytycznym

W wyniku eksperymentu, którego rezultat przedstawiono na rysunku 8.13 uzyskano $K_{pkr} = 10,2$ oraz $T_{osc} = 27s$. Obliczone nastawy według wzorów Zieglera-Nicholsa zestawiono w tabeli 10.

Tabela 10 Nastawy algorytmu PID, uzyskane za pomocą metody cyklu granicznego

	Kp	Ti	Td
P	5,1	-	-
PI	4,59	22,95	-
PID	6,12	13,5	3,24

8.6.2 Identyfikacja obiektu z użyciem procedur samostrojzenia regulatora LB-600

Regulator LB-600, wyposażony jest w procedury samostrojzenia, które bazują na dwóch metodach identyfikacji obiektu:

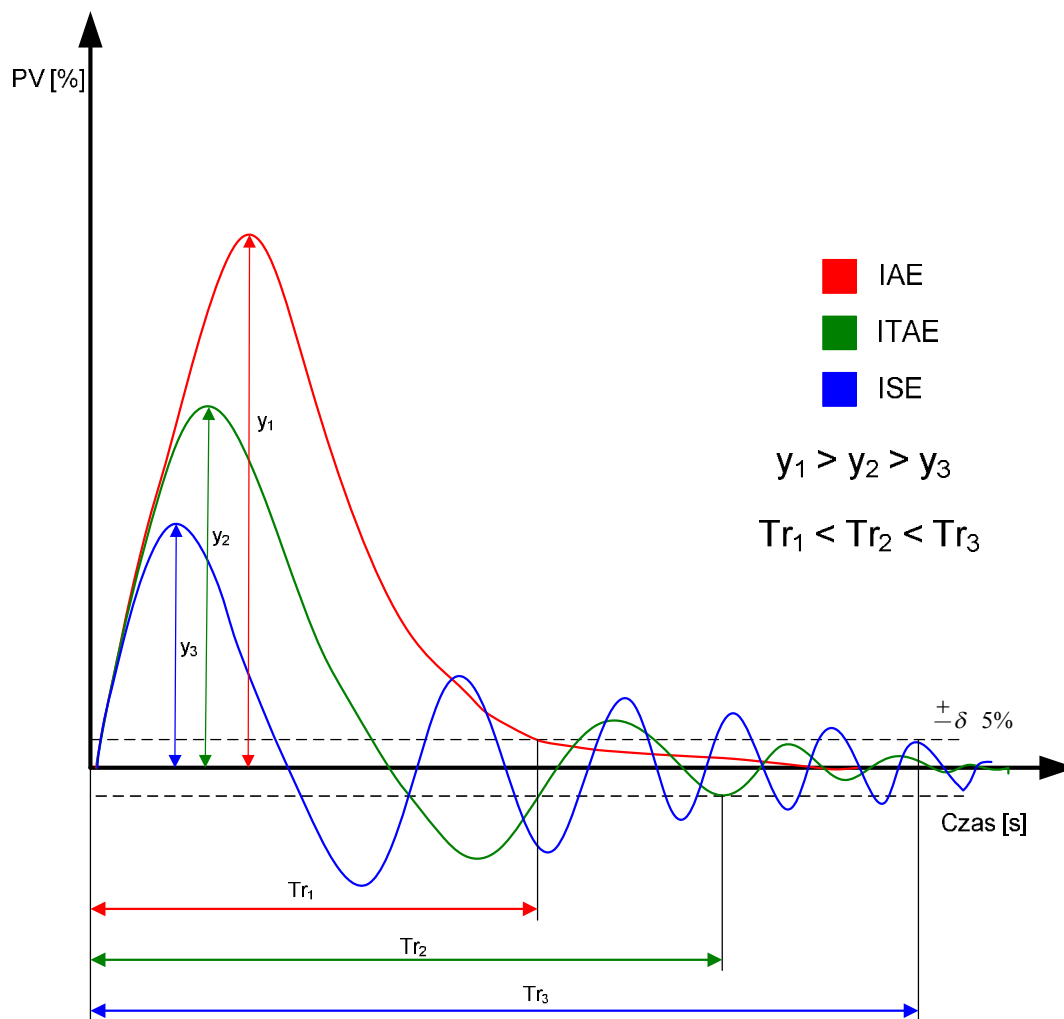
- a. metoda siecznej
- b. metoda stycznej

W wyniku otrzymanych wartości, urządzenie oblicza parametry obiektu takie jak, T_0 - czas martwy, T - stała czasowa, oraz K_{ob} - wzmacnienie. Następnie w zależności od wybranego algorytmu: P, PI lub PID; regulator oblicza nastawy zgodnie z wybranym sposobem.

Oprócz obliczania nastaw według zmodyfikowanego eksperymentu Zieglera-Nicholsa i Cohena-Coona regulator umożliwia obliczenie nastaw według następujących cech przebiegu przejściowego:

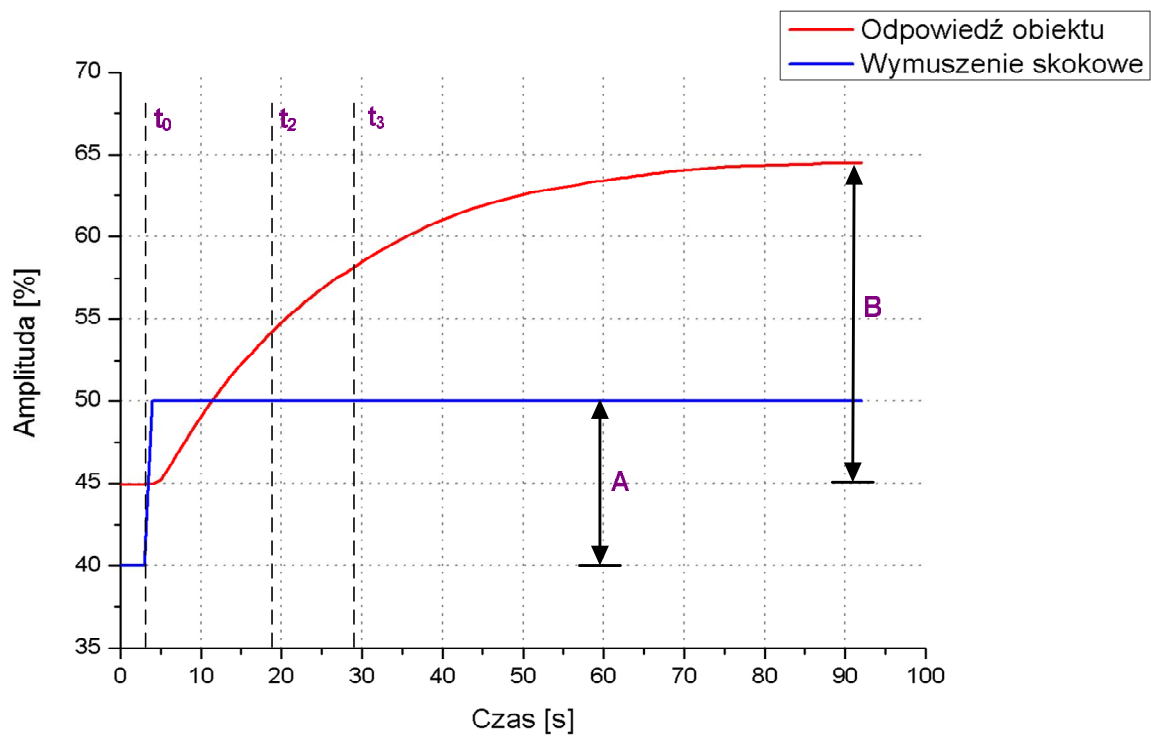
- Przeregulowanie 0% (**I**ntegral of the **A**bsolute value of **E**rror) zapewnia minimalne pole odchyłki regulacji obliczane wg. $\int_0^{\infty} |e(t)| dt$
- Przeregulowanie 45% (**I**ntegral of **S**quare of the **E**rror) zapewniające minimalne pole kwadratu odchyłki regulacji obliczane wg. $\int_0^{\infty} e^2(t) dt$
- Przeregulowanie 20% (**I**ntegral of the **T**ime weighted **A**bsolute **E**rror) będące kompromisem pomiędzy powyższymi kryteriami, obliczane wg. $\int_0^{\infty} t|e(t)| dt$

Różnicę pomiędzy powyższymi kryteriami została przedstawiona na rysunku 8.14.



Rys. 8.14 Zależność pomiędzy kryteriami według przebiegu przejściowego

Identyfikacja obiektu za pomocą metody sieciowej polega na zarejestrowaniu odpowiedzi na wymuszenie skokowe oraz odczytanie czasów w których, odpowiedź obiektu przyjmuje wartość 50% oraz 63,2% amplitudy w stanie ustalonym. Opisywana metoda identyfikacji została przedstawiona na poniższym rysunku.



Rys. 8.15 Wyznaczanie parametrów obiektu metodą sieciowej

Po odczytaniu czasów t_0 , t_2 oraz t_3 , regulator oblicza parametry obiektu za pomocą następujących wzorów:

$$t_1 = \frac{[t_2 - \ln(2)t_3]}{[1 - \ln(2)]} \quad (8.19)$$

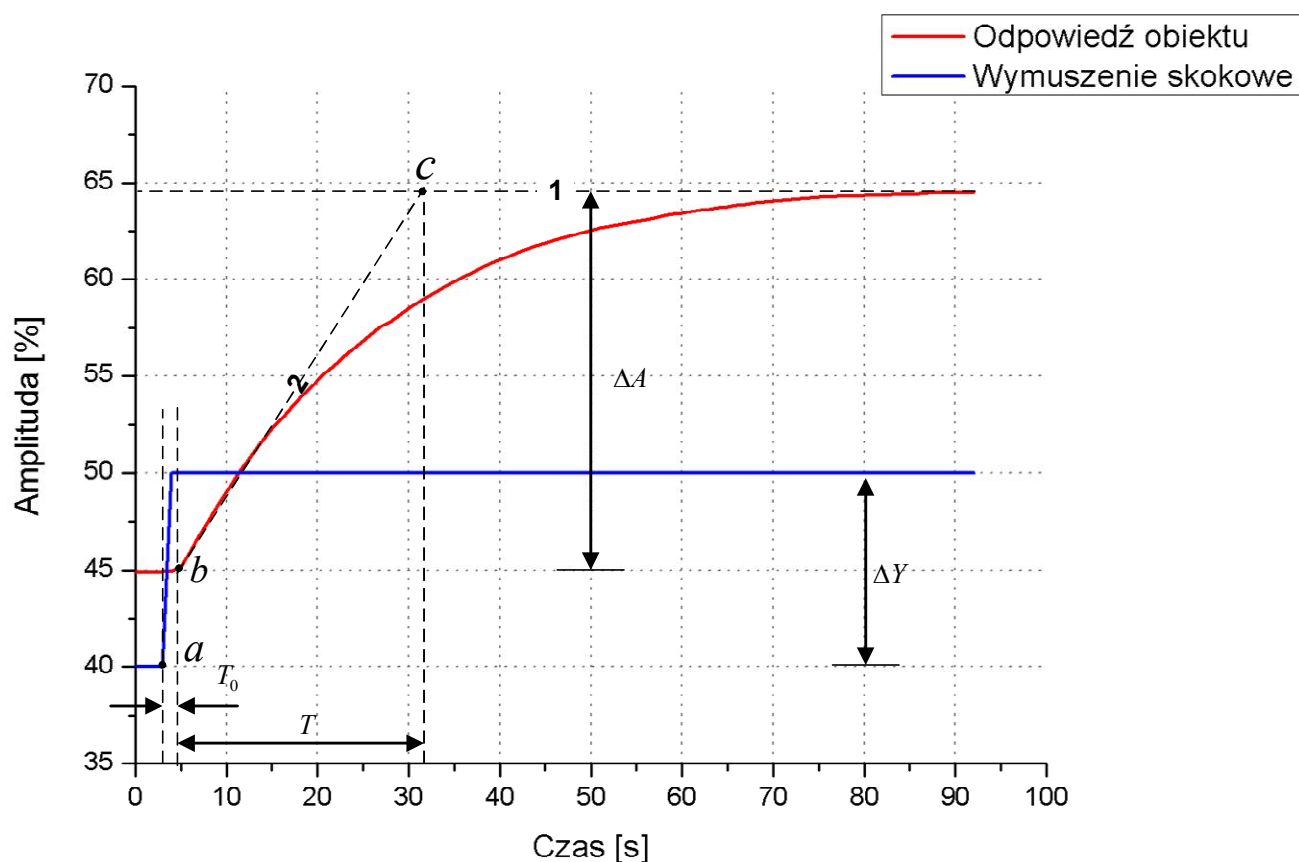
$$T = t_3 - t_1 \quad (8.20)$$

$$T_0 = t_1 - t_0 \quad (8.21)$$

$$K_{ob} = \frac{B}{A} \quad (8.22)$$

Drugą metodą identyfikacji obiektu jest, metoda stycznej (rysunek 8.16), jest ona znana i chętnie stosowana przez inżynierów. Odczytanie parametrów obiektu poprzedza wykonanie poniższych czynności:

1. wyznaczenie asymptoty (1), do której zmierza amplituda obiektu w stanie ustalonym,
2. wyznaczenie stycznej (2) w punkcie przegięcia odpowiedzi obiektu,
3. rzutowanie na oś czasu, punktów charakterystycznych a, b i c,
4. odczytanie parametrów obiektu.



Rys. 8.16 Wyznaczanie parametrów obiektu metodą stycznej

Nieprawidłowe określenie punktu przegięcia w metodzie stycznej, skutkuje automatycznym odczytaniem błędnych parametrów obiektu. Ograniczeń tego typu nie posiada metoda siecznej.

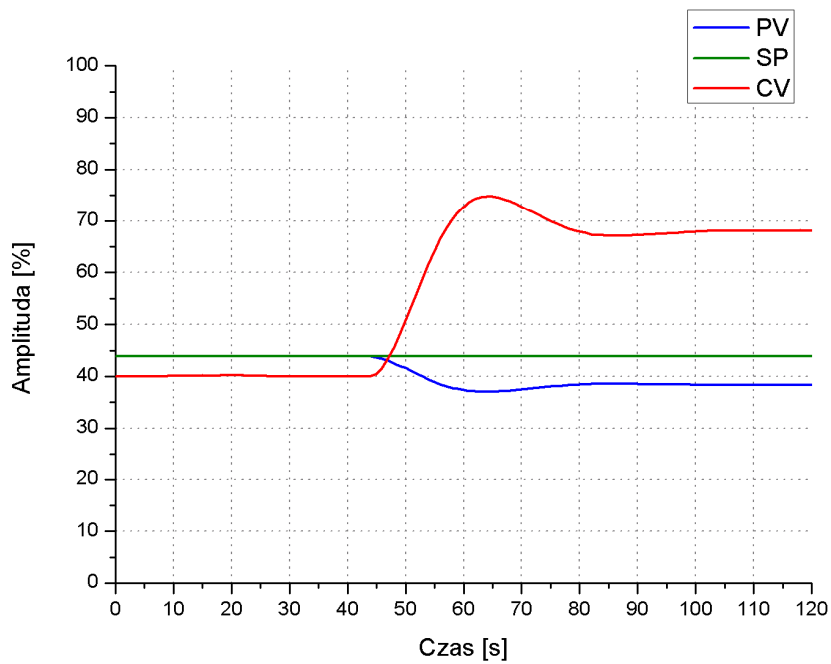
Na potrzeby badań, zostały przeprowadzone procedury samostrojzenia w celu, obliczenia nastaw według Zieglera-Nicholsa oraz Cohena-Coona. Obliczone nastawy przedstawia tabela 11.

Tabela 11 Nastawy algorytmu PID uzyskane za pomocą procedur samostrojzenia regulatora LB-600

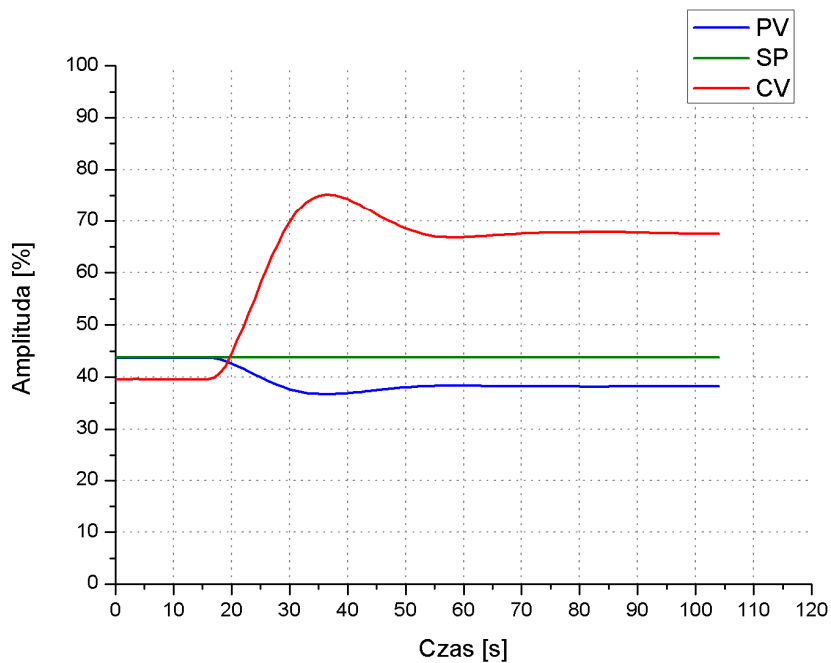
	algorytm	Kp	Ti	Td
wg Zieglera-Nicholsa	P	7,85	-	-
	PI	1,23	21,4	-
	PID	8,95	13,3	3,3
wg. Cohena-Coona	P	2,3	-	-
	PI	1,9	28,7	-
	PID	2,95	31,6	5

8.6.3 Przebiegi przejściowe obiektu z regulatorem LB-600 oraz Soft Control.

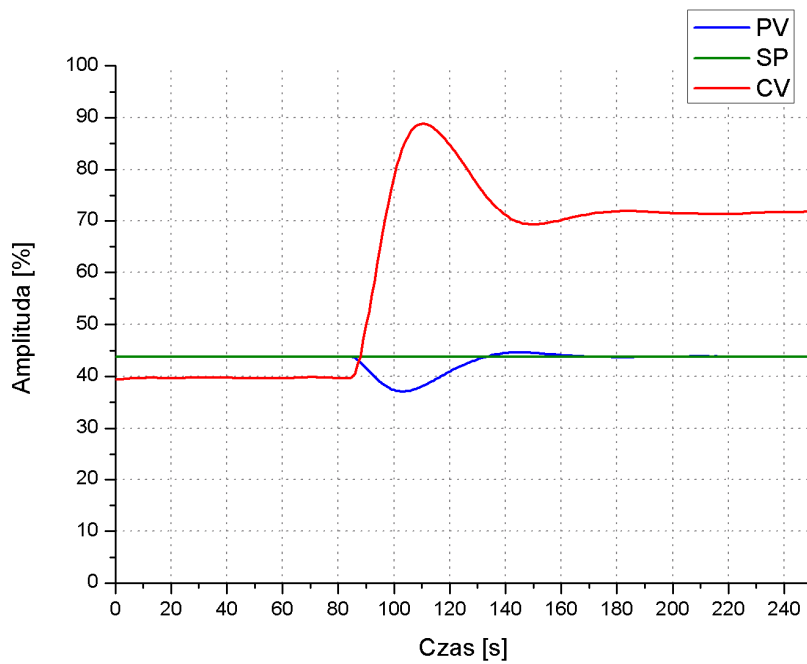
Na poniższych rysunkach przedstawiono przebiegi przejściowe układu regulacji poziomu z regulatorem LB-600 oraz Soft Control. Przebiegi zostały wyznaczone na podstawie nastaw obliczonych za pomocą metody cyklu granicznego. Dla regulatora Soft Control ograniczono działanie różniczkujące przyjmując współczynnik $\gamma = 0,003$. Ze względu na ograniczoną przepustowość łącza danych pomiędzy komputerem oraz regulatorem (przekazującym sygnały pomiędzy obiektem a komputerem), czas próbkowania ustawiono na 0,1 s. Zakłócenie zostało wywołane otwarciem zaworu VE2 (zrzut pompy).



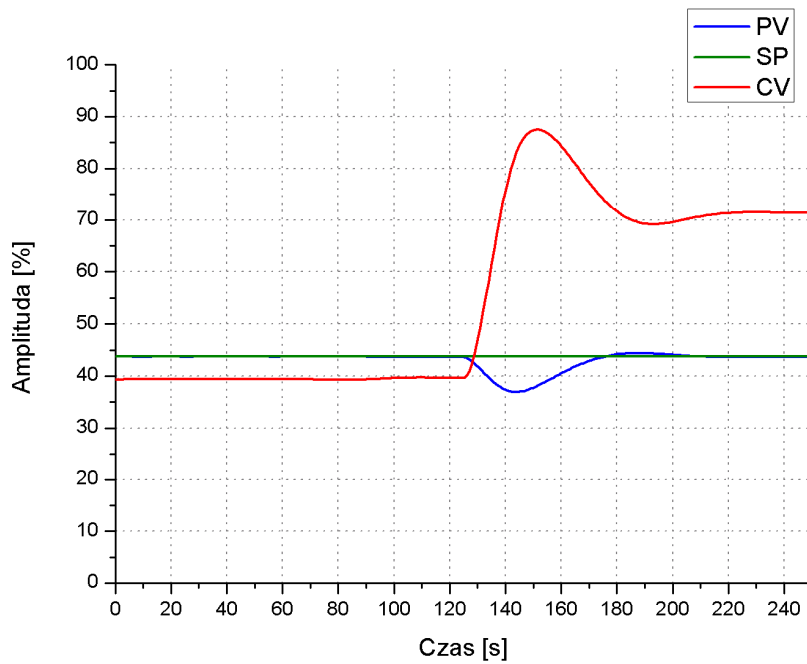
Rys. 8.17 Przebieg przejściowy w układzie regulacji, z regulatorem LB-600 o algorytmie P, wywołany otwarciem zaworu VE2



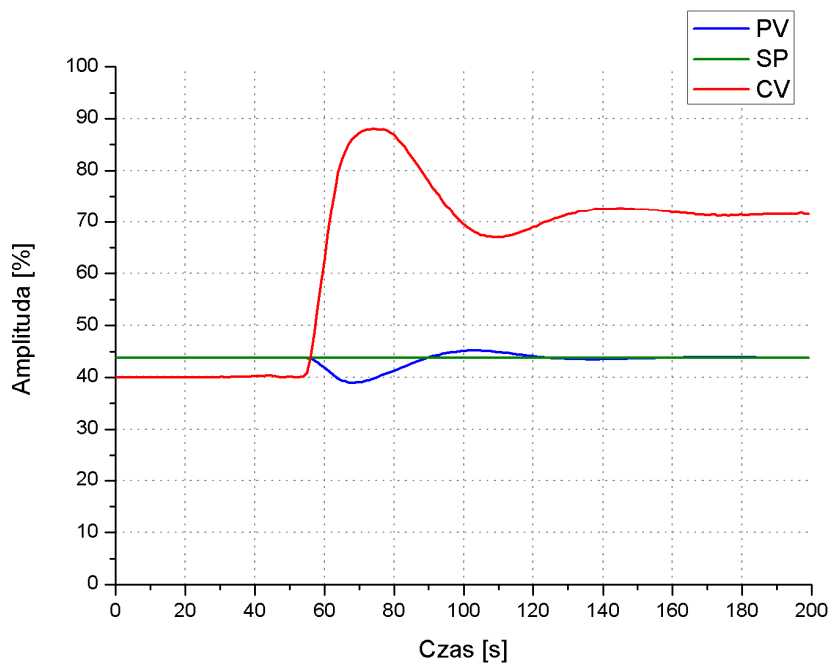
Rys. 8.18 Przebieg przejściowy w układzie regulacji, z regulatorem Soft Control o algorytmie P, wywołany otwarciem zaworu VE2



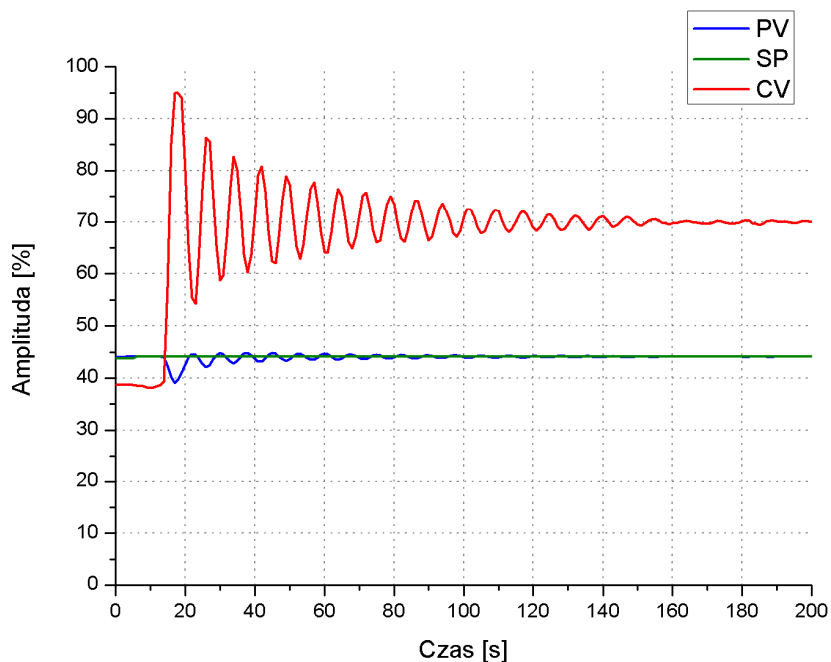
Rys. 8.19 Przebieg przejściowy w układzie regulacji, z regulatorem LB-600 o algorytmie PI, wywołany otwarciem zaworu VE2



Rys. 8.20 Przebieg przejściowy w układzie regulacji, z regulatorem Soft Control o algorytmie PI, wywołany otwarciem zaworu VE2



Rys. 8.21 Przebieg przejściowy w układzie regulacji, z regulatorem LB-600 o algorytmie PID, wywołany otwarciem zaworu VE2



Rys. 8.22 Przebieg przejściowy w układzie regulacji, z regulatorem Soft Control o algorytmie PID, wywołany otwarciem zaworu VE2

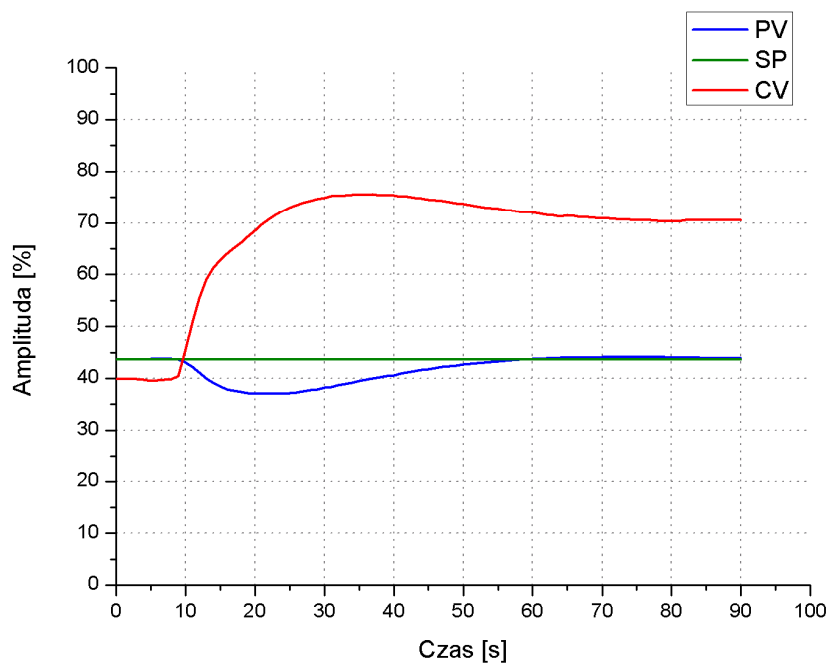
Odczytane wskaźniki regulacji, powyższych przebiegów, zestawiono w tabeli 12.

Tabela 12 Wskaźniki jakości regulacji dla przebiegów regulatorów LB-600 oraz Soft Control uzyskane na podstawie nastaw wyznaczonych metodą cyklu granicznego

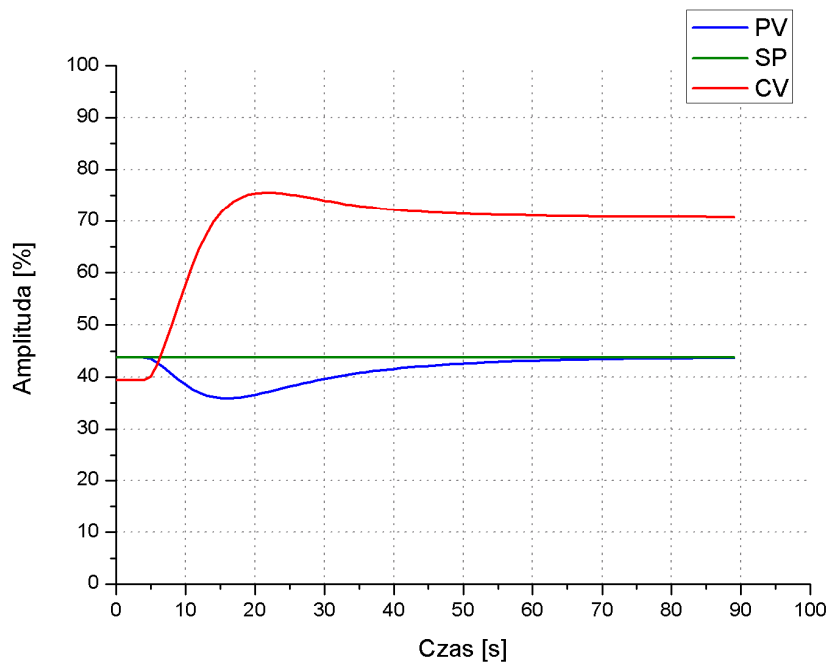
	LB-600				Soft Control			
	E_{st}	E_m	Tr [s]	χ [%]	Est	Em	Tr [s]	χ [%]
P	5,51	6,83	47	20	5,61	7,05	51	25
PI	0	6,69	43	12	0	6,76	47	10
PID	0	4,8	57	29	0	5,03	52	12

Przyczyną różnicy pomiędzy wskaźnikami jakości dynamicznej, algorytmów P oraz PI jest czas skanowania bloków bazy danych systemu iFIX4.0. Prędkość z jaką drajwer We/Wy odczytywał dane z regulatora wahał się w granicach 0,07 – 0,09[s], natomiast blok AI odpowiadający za przekazywanie wartości mierzonej do bloku PID był uaktualniany co 0,1 s. Z taką samą prędkością poprzez blok AO odbywało się również przekazywanie obliczonego sygnału sterującego do regulatora. Tak więc ostatecznie reakcja na zmianę sygnału wartości mierzonej mogła w skrajnych przypadkach, być opóźniona o czas ok. 200 ms.

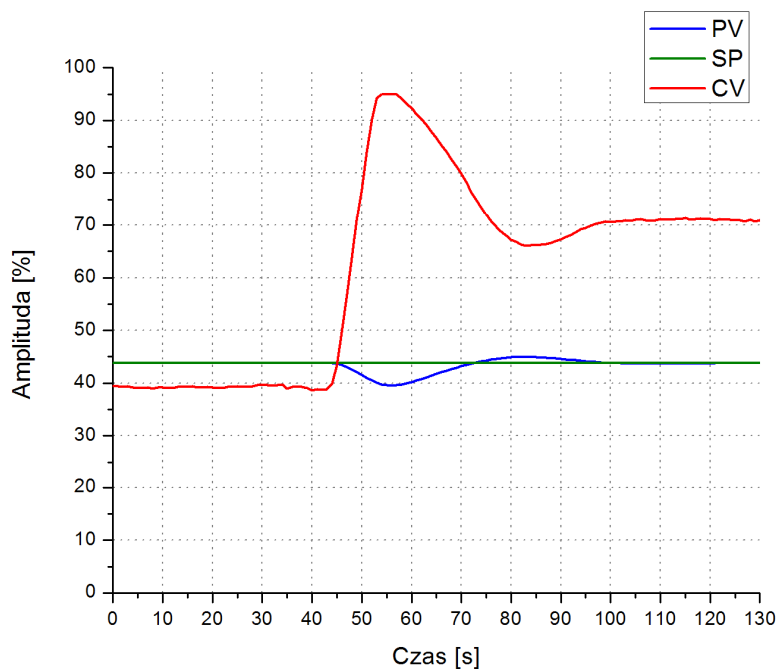
Pomimo dość obiecujących wskaźników jakości regulacji algorytmu PID Soft Control, należy zwrócić uwagę na agresywny przebieg sygnału sterującego. Różniczkowanie w regulatorach o algorytmie PID, pozwala ustawić większe wzmocnienie i krótszy czas T_i . W przypadku bloku PID systemu iFIX, zwiększenie K_p oraz T_i skutkuje niepoprawną pracą układu. Aby ograniczyć negatywny wpływ różniczkowania producent systemu wprowadził współczynnik , który de facto maskuje błąd algorytmu przy „łagodnych” nastawach rysunki 8.23 i 8.24. W przypadku „agresywnych” nastaw, które wykorzystują korzyści składowej D, już samo przełączenie, do pracy automatycznej przy znikomej odchyłce, powoduje niestabilność układu rysunek 8.26.



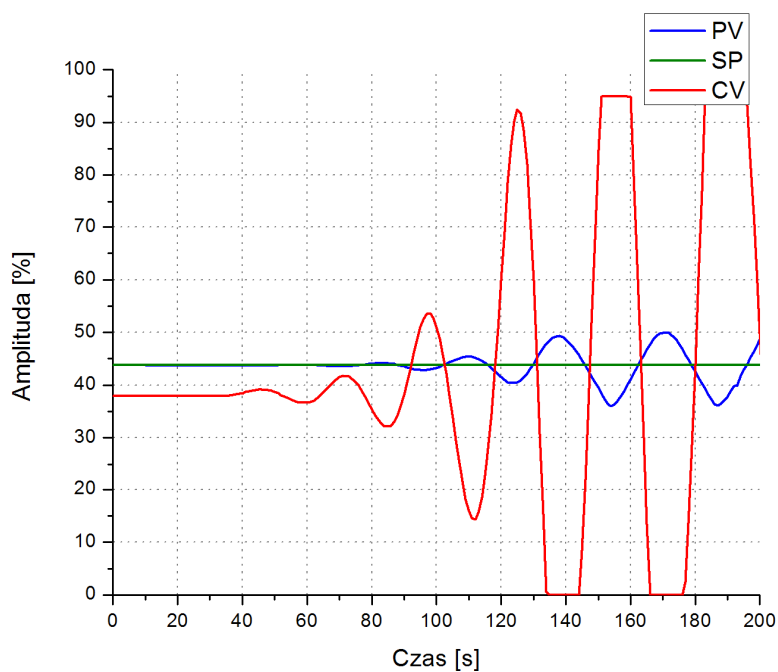
Rys. 8.23 Przebieg przejściowy w układzie regulacji, z regulatorem LB-600, wywołany otwarciem zaworu VE2 (algorytm PID, nastawy wg Cohena-Coona)



Rys. 8.24 Przebieg przejściowy w układzie regulacji, z regulatorem Soft Control, wywołany otwarciem zaworu VE2 (algorytm PID, nastawy wg Cohena-Coona)



Rys. 8.25 Przebieg przejściowy w układzie regulacji, z regulatorem LB-600, wywołany otwarciem zaworu VE2 (algorytm PID, nastawy wg Zieglera-Nicholsa)



Rys. 8.26 Reakcja algorytmu Soft Control, w chwili przełączenia na pracę automatyczną przy zerowej odchyłce, algorytm PID, nastawy wg Zieglera-Nicholsa

9 Wnioski

Oprogramowanie Proficy HMI/SCADA iFIX4.0, zapewniło wszelkie niezbędne narzędzia, potrzebne w procesie programowania, zaprojektowanych funkcji systemu wizualizacji. Dużą zaletą w tworzonych projektach jest możliwość użycia języka programowania: C++ oraz VBA. Wspomniana funkcjonalność umożliwia praktyczną realizację rozwiązań, od zaawansowanych animacji, poprzez komunikację z innymi systemami a skończywszy na skomplikowanych obliczeniach. Szeroka gama dostępnych drajwerów komunikacyjnych oraz możliwość tworzenia własnych sprawia iż, systemy tworzone za pomocą tego oprogramowania mogą komunikować się z dowolnym sprzętem. Proficy iFIX4.0 udostępnia również szeroką gamę specjalizowanych bloków, ułatwiających tworzenie różnorodnych systemów wizualizacji i sterowania.

Powyższe cechy sprawiają iż omawiane oprogramowanie jest w stanie spełnić najbardziej wygórowane wymagania przyszłego użytkownika.

Regulator mikroprocesorowy LB-600, spełnił większość oczekiwań odnośnie integracji z systemem wizualizacji. Mnogość obsługiwanych parametrów za pośrednictwem rejestrów jest imponująca. Możliwość dowolnego konfigurowania wejść poszczególnych bloków funkcyjnych, oraz duża różnorodność funkcji przez nie realizowanych sprawia iż, możemy zaprogramować praktycznie dowolną strukturę regulacji.

Mimo powyższych zalet regulatora, pełna zdalna obsługa funkcjonalna urządzenia jest niemożliwa. Wspomniana niedogodność dotyczy przede wszystkim braku rejestrów odpowiedzialnych za obsługę procedury samostrojzenia, które na tle dostępnej liczby rejestrów (np. ponad 1000 rejestrów obsługujących regulację tablicową) wydaje się być niedopatrzeniem. Następną funkcją początkowo sprawiająca wrażenie niemożliwej do zrealizowania, była zmiana sygnału sterującego CV w trybie Manual, za pośrednictwem komputera. Wszakże autor pracy uporał się z tą niegodnością, jednak zadowalające rozwiązanie, jest czasochłonne i zmniejszające zasoby regulatora oraz w rzeczywistości mogące wpłynąć na końcowy koszt systemu (zakup licencji na większą liczbę zmiennych, zakup dodatkowego regulatora w przypadku użycia więcej niż 4 kanałów PID).

Niewątpliwą zaletą regulatora LB-600 jest algorytm PID z rzeczywistym różniczkowaniem, którego cyfrowa implementacja jest dość trudna i jak wykazano w pracy może sprawić problem nawet dużym firmom wyspecjalizowanych w rozwiązaniach dla

automatyki. Pomimo prawidłowego różniczkowania, poprzez wprowadzenie nie-nastawialnego wzmocnienia dynamicznego, użytkownik regulatora jest pozbawiony możliwości kształtowania właściwości dynamicznych członu różniczkującego.

Zaprogramowany system wizualizacji i sterowania, prawidłowo realizuje założone funkcje. Wykryte, podczas testowania błędy zostały poprawione. Ze względu na błąd algorytmu bloku PID, sterowanie Soft Control jest ograniczone do algorytmów P oraz PI. Dzięki intuicyjnej formie prezentacji, oraz możliwości wgrywania odpowiednich struktur regulacji, system eliminuje konieczność zapoznania się z dokumentacją techniczno ruchową regulatora, oraz bezpośrednio jego programowanie. Umożliwiając przy skupieniu uwagi studentów, na głównym celu przeprowadzanych ćwiczeń laboratoryjnych. Kolejnym udogodnieniem dla osób wykonujących ćwiczenia jest komputerowa rejestracja oraz prezentacja badanych układów regulacji w trzech formach. Pierwsza cecha pomaga opracować dokładniejsze wyniki badań oraz pomaga odtworzyć dane uzyskane podczas badań, natomiast druga cecha pozwala lepiej zrozumieć zasadę działania badanych układów regulacji.

Ponieważ implementacja cyfrowego algorytmu PID jest uzależniona od projektanta urządzenia bądź programisty sterownika PLC, może on przyjmować różne formy. Z wyników badań przedstawionych w tej pracy można zauważyć, że nie każdy algorytm PID spełnia stawiane mu wymagania. Z tego też względu podczas pracy z urządzeniami w których zaimplementowany jest cyfrowy algorytm PID należy zachować szczególną ostrożność oraz przede wszystkim zbadać jego charakterystyki na wymuszenie skokowe oraz liniowo narastające. Brak wiedzy odnośnie charakterystyk oraz definicji poszczególnych nastaw algorytmu może doprowadzić do niestabilnej pracy obiektu regulacji.

10 Literatura

- [1] R. Januszewski „Programowanie systemów SCADA”, Wydawnictwo pracowni komputerowej Jacka Skalmierskiego, Gliwice 2006 ISBN 83-607-16-07-2
- [2] "Modicon Modbus Protocol Reference Guide" źródło <http://www.modbus.org/>
- [3] „Understanding iFIX” – Podręcznik elektroniczny systemu iFIX4.0
- [4] „Building SCADA System” - Podręcznik elektroniczny systemu iFIX4.0
- [5] Z. Komor, A.Łobzowski, W. Szkolnikowski – „Technika Regulacji Regulator LB-600” Agenda wydawnicza PAK-u, Warszawa 2003 ISBN 83-87982-11-3
- [6] J. Kwaśniewski „Programowalne sterowniki przemysłowe w systemach sterowania” Kraków 1999 ISBN 83-86320-45-1
- [7] MB1 I/O driver help – elektroniczna dokumentacja drajwera MB1
- [8] D. Holejko „Automatyka procesów ciągłych” – Preskrypt, IAIr Warszawa 2006
- [9] „Regulator – sterownik EFTRONIK LB-600” – instrukcja eksploatacyjna 19.11.2007

11 Załączniki

W załączniku A zestawiono, wszystkie zmienne występujące w procesowej bazie danych, oraz przyporządkowano do nich numery rejestrów drajwera komunikacyjnego MB1 i odpowiadające im adresy regulatora LB600.

Załącznik B, służy jako pomoc podczas instalacji i konfiguracji aplikacji: „Regulacja poziomu”. Mimo dołączenia do pracy wszelkich niezbędnych plików konfiguracyjnych, proces ten jest dość skomplikowany. Z tego też powodu wymieniony załącznik ma formę obrazkowej instrukcji w, której wszystkie czynności opisane są krok po kroku.

11.1 Załącznik A – wykaz zmiennych i adresów

Lp.	Nazwa zmiennej w bazie danych.	Adres regulatora	Adres driver'a MB1	Opis
1	AH_REG1	7125	44937	Alarm górny od odchyłki dla regulatora kanał 1, warstwa 7
2	AH_REG2	7225	45037	Alarm górny od odchyłki dla regulatora kanał 2, warstwa 7
3	AI_K1	1120	40878	PV wartość wejściowa do regulatora, kanał 1 warstwa 7 (poziom)
4	AI_K1_PID	1120	40878	Wartość wejściowa PV bloku PID
5	AI_K2	1220	40918	PV wartość wejściowa do regulatora kanał 2, warstwa 7 (przepływ)
6	AI1_S	-	-	Blok skalujący AI1
7	AI2_S	-	-	Blok skalujący AI2
8	AKCEPATAcja_NASTAW	7156	44979	akceptacja po zakończeniu samostrojzenia dla regulatora kanał1
9	AKTW_BLOKU_AI1	1119	40877	Aktywność bloku AI kanał 1 warstwa 1
10	AKTW_BLOKU_AI2	1219	40917	Aktywność bloku AI kanał 2 warstwa1
11	AKTW_BLOKU_AO1	9111	46011	Aktywność bloku przetwarzania kanał 1 warstwa 8
12	AKTW_BLOKU_W8	8111	45817	aktywność bloku w warstwie 8
13	AKTW_DI1	2104	42684	Aktywność bloku DI kanał 1 warstwa 2 (RSB bloku wykorzystywane przy samostrojzeniu)

14	AKTW_DI2	2204	42694	Aktywność bloku DI kanał 2 warstwa 2 (RSB bloku wykorzystywane przy samostrojeniu)
15	AKTW_REG1	7165	44992	Aktywność bloku regulatora kanał 1 warstwa 7
16	AKTW_REG2	7265	45092	Aktywność bloku regulatora kanał 2 warstwa 7
17	AKTW_REG3	7365	45192	Aktywność bloku regulatora kanał 3 warstwa 7 (realizacja sterowania ręcznego dla regulatora 2)
18	AKTW_REG4	7465	45292	Aktywność bloku regulatora kanał 4 warstwa 7 (realizacja sterowania ręcznego dla regulatora 1)
19	AL_HI_K1	1108	40861	Alarm wysoki, kanał 1
20	AL_HI_K2	1208	40901	Alarm wysoki, kanał2
21	AL_HIHI_K1	1110	40865	Alarm bardzo wysoki dla wejścia AI kanał 1
22	AL_HIHI_K2	1210	40905	Alarm bardzo wysoki dla wejścia AI kanał 2
23	AL_HILOHI_K1	1112	40868	Histerza alarmów bardzo wysokich/niskich kanał 1
24	AL_HILOHI_K2	1212	40908	Histerza alarmów bardzo wysokich/niskich kanał 2
25	AL_HILOLO_K1	1111	40867	Histerza alarmów wysokich/niskich, kanał 1
26	AL_HILOLO_K2	1211	40907	Histerza alarmów wysokich/niskich, kanał 2
27	AL_LO_K1	1107	40859	Alarm niski, kanał1
28	AL_LO_K2	1207	40899	Alarm niski, kanał 2

29	AL_LOLO_K1	1109	40863	Alarm bardzo niski, kanał 1
30	AL_LOLO_K2	1209	40903	Alarm bardzo niski, kanał 2
31	AL_REG1	7124	44936	Alarm niski od odchyłki regulatora, kanał 1 warstwa 7
32	AL_REG2	7224	45036	Alarm niski od odchyłki regulatora, kanał warstwa 7
33	ALGR_PRZTW_W8	8106	45808	Sposób przetwarzania sygnału dla bloku kanał 1 warstwa 8 (realizacja CV dla bloku PID)
34	ALGRM_REG1	7110	44912	Algorytm regulatora kanał 1
35	ALGRM_REG2	7210	45012	Algorytm regulatora kanał 2
36	ALGRM_REG3	7310	45112	Algorytm regulatora kanał 3
37	ALGRM_REG4	7410	45212	Algorytm regulatora kanał 4
38	AMPL_PV_K1	7150	44969	Amplituda PV podczas stabilizacji (samostrojenie), kanał 1 warstwa 7
39	AO_1_LIMIT_MAX	9105	46005	Górny limit wartości sygnału, kanał 1 warstwa 9
40	AO_1_LIMIT_MIN	9104	46004	Dolny limit wartości sygnału, kanał 1 warstwa 9
41	BLOCK_ALARM	0103	40004	Blokada sygnału alarmowego regulatora, warstwa 0
42	CV	-	-	Blok skalujący AO1
43	CV_MAX_REG1	7162	44987	Górny limit wartości CV, kanał 1 warstwa 7

44	CV_MAX_REG2	7262	45087	Górny limit wartości CV, kanał 2 warstwa 7
45	CV_MAX_REG3	7362	45187	Górny limit wartości CV, kanał 3 warstwa 7
46	CV_MAX_REG4	7462	45287	Górny limit wartości CV, kanał 4 warstwa 7
47	CV_MIN_REG1	7161	44985	Dolny limit wartości CV, kanał 1 warstwa 7
48	CV_MIN_REG2	7261	45085	Dolny limit wartości CV, kanał 2 warstwa 7
49	CV_MIN_REG3	7361	45185	Dolny limit wartości CV, kanał 3 warstwa 7
50	CV_MIN_REG4	7461	45285	Dolny limit wartości CV, kanał 4 warstwa 7
51	CV_REG_1	7166	44993	Wartość CV regulatora, kanał 1 warstwa 7 (wartość zadana regulatora 2 przy kaskadowym UAR)
52	CV_REG_2	7266	45093	Wartość CV regulatora, kanał 1 warstwa 7
53	CZAS_SAMOSTROJENIA	0124	40025	Okres przełączenia kanałów na wyświetlaczu regulatora (czas maksymalny przy samostrojeniu)
54	CZAS_STABILIZACI_K1	7149	44968	Czas stabilizacji procesu (samostrojenie) kanał 1 warstwa 7
55	DELTA_BARGRAF	7126	44938	Zakres wskazań odchyłki na bargrafie , kanał 1 warstwa 7
56	DELTA_BARGRAF2	7226	45038	Zakres wskazań odchyłki na bargrafie , kanał 2 warstwa 7
57	DELTA_E_REGULATOR_K1	7164	44990	Wartość odchyłki regulatora, kanał 1 warstwa 7
58	DELTA_E_REGULATOR_K2	7264	45090	Wartość odchyłki regulatora, kanał 2 warstwa 7

59	HS_REG1	7127	44939	Histeresa alarmów od odchyłki regulatora, kanał 1 warstwa 7
60	HS_REG2	7227	45039	Histeresa alarmów od odchyłki regulatora, kanał 2 warstwa 7
61	IDENTY_OBKT_K1	7148	44967	Metoda identyfikacji obiektu (samostrojzenie), kanał 1 warstwa 7
62	KP_K1	7114	44917	Wzmocnienie proporcjonalne regulatora, kanał 1 warstwa 7
63	KP_K2	7214	45017	Wzmocnienie proporcjonalne regulatora, kanał 2 warstwa 7
64	LG_DI1	2102	42682	Logika wejścia DI, kanał 1 warstwa 2
65	LG_DI2	2202	42692	Logika wejścia DI, kanał 2 warstwa 2
66	N_R_K1	7123	44935	Odwracanie sygnału CV regulatora, kanał 1 warstwa 7
67	N_R_K2	7223	45035	Odwracania sygnału CV regulatora, kanał 2 warstwa 7
68	N_R_REG3	7323	45135	Odwracania sygnału CV regulatora, kanał 3 warstwa 7
69	N_R_REG4	7423	45235	Odwracanie sygnału CV regulatora, kanał 4 warstwa 7
70	OBL_NASTAW_K1	7145	44962	Algorytm obliczania nastaw (samostrojzenie), kanał 1 warstwa 7
71	OPOZNIENIE_OBIEKTU_K1	7154	44975	Opóźnione obiektu, parametr obliczony podczas samostrojzenia, kanał 1 warstwa 7
72	PID_KOMP	-	-	Blok sterujący PID (iFix)
73	POZIOM	-	-	Bufor wykresu AI 1

74	PRZEPLÝW	-	-	Bufor wykresu AI 2
75	PV_MAX_K1	1106	40857	Górne ograniczenie sygnału z bloku AI wyrażone w jednostkach fizycznych, kanał 1 warstwa 1
76	PV_MAX_K2	1206	40897	Górne ograniczenie sygnału z bloku AI wyrażone w jednostkach fizycznych, kanał 2 warstwa 1
77	PV_MAX_REG1	7108	44909	Górne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 1 warstwa 7
78	PV_MAX_REG2	7208	45009	Górne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 2 warstwa 7
79	PV_MAX_REG3	7308	45109	Górne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 3 warstwa 7
80	PV_MAX_REG4	7408	45209	Górne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 4 warstwa 7
81	PV_MIN_K1	1105	40855	Dołne ograniczenie wartości bloku AI wyrażone w jednostkach fizycznych, kanał 1 warstwa 1
82	PV_MIN_K2	1205	40895	Dołne ograniczenie wartości bloku AI wyrażone w jednostkach fizycznych, kanał 2 warstwa 1
83	PV_MIN_REG1	7107	44907	Dołne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 1 warstwa 7
84	PV_MIN_REG2	7207	45007	Dołne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 2 warstwa 7
85	PV_MIN_REG3	7307	45107	Dołne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 3 warstwa 7
86	PV_MIN_REG4	7407	45207	Dołne ograniczenie CV regulatora wyrażone w jednostkach fizycznych, kanał 4 warstwa 7
87	RSB_START	7142	44959	RSB alarmu DI, kanał 1 warstwa 2
88	RSB_STOP	7143	44960	RSB alarmu DI, kanał 2 warstwa 2

89	SKOK_CV_K1	7151	44970	Wartość amplitudy skoku (samostrojenie), kanał 1 warstwa 7
90	SLEDZENIE_CAS_REG1	7113	44916	Równoważenie kaskady CV1 = SP2
91	SLEDZENIE_CAS_REG2	7213	45016	śledzenie przy kaskadzie
92	SLEDZENIE_REG1	7112	44915	Równoważenie SP=PV, kanał 1 warstwa 7
93	SLEDZENIE_REG2	7212	45015	Równoważenie SP=PV, kanał 2 warstwa 7
94	SLEDZENIE_REG3	7313	45116	Równoważenie kaskady CV1=PV2, kanał 3 warstwa 7
95	SLEDZENIE_REG4	7413	45216	Równoważenie kaskady CV1=PV2, kanał 4 warstwa 7
96	SP_K1	7120	44929	SP 1 (wartości historyczne)
97	SP_K2	7220	45029	SP 2 (wartości historyczne)
98	SP_REG3	7320	45129	Wartość zadana regulatora, kanał 3 warstwa 7 (realizacja sterowania ręcznego dla regulatora, kanał 2)
99	SP_REG4	7420	45229	Wartość zadana regulatora, kanał 4 warstwa 7 (realizacja sterowania ręcznego dla regulatora, kanał 1)
100	STALA_CZASOWA_OBIEKTU_K1	7153	44973	Stała czasowa obiektu, parametr obliczony podczas samostrojenia, kanał 1 warstwa 7
101	SWITCH_A_K2	7259	45082	Blokada trybu A podczas pracy w kaskadzie, kanał 2 warstwa 7
102	SWITCH_A_REG3	7359	45182	Blokada trybu A podczas pracy w kaskadzie, kanał 3 warstwa 7
103	SWITCH_A_REG4	7459	45282	Blokada trybu A podczas pracy w kaskadzie, kanał 4 warstwa 7

104	TD_K1	7116	44921	Czas wyprzedzenia, kanał 1 warstwa 7
105	TD_K2	7216	45021	Czas wyprzedzenia, kanał 2 warstwa 7
106	TF_1	1104	40854	Stała czasowa filtru wartości PV, kanał 1 warstwa 1
107	TF_2	1204	40894	Stałą czasowa filtru wartości PV, kanał 2 warstwa 1
108	TI_K1	7115	44919	Czas zdwojenia, kanał 1 warstwa 7
109	TI_K2	7215	45019	Czas zdwojenia, kanał 2 warstwa 7
110	TOR_SKAD_CV_AO1	9103	46003	Kanał z którego pochodzi sygnał wejściowy AO, kanał 1 warstwa 8
111	TOR_SKAD_PV_REG1	7103	44903	Kanał z którego pochodzi sygnał PV regulatora, kanał 1 warstwa 7
112	TOR_SKAD_PV_REG2	7203	45003	Kanał z którego pochodzi sygnał PV regulatora, kanał 2 warstwa 7
113	TOR_SKAD_SP_REG2	7206	45006	Kanał z którego pochodzi sygnał PV regulatora, kanał 1 warstwa 7
114	TOR_SKAD_SP_REG3	7306	45106	Kanał z którego pochodzi sygnał PV regulatora, kanał 1 warstwa 7
115	TOR_SKAD_SP_REG4	7406	45206	Kanał z którego pochodzi sygnał PV regulatora, kanał 1 warstwa 7
116	TRYB_PRACY_PO_SAMOSTROJENIU	7157	44980	Tryb pracy po zakończeniu samostrojania, kanał 1 warstwa 7
117	TRYB_PRACY_REG_K1	7158	44981	Przełączanie trybu pracy regulatora, kanał 1 warstwa 7
118	TRYB_PRACY_REG_K2	7258	45081	Przełączanie trybu pracy regulatora, kanał 2 warstwa 7

119	TRYB_PRACY_REG_K3	7358	45181	Przelączenie trybu pracy regulatora, kanał 3 warstwa 7
120	TRYB_PRACY_REG_K4	7458	45281	Przelączenie trybu pracy regulatora, kanał 4 warstwa 7
121	TYP_REG1	7109	44911	Typ regulatora, kanał 1 warstwa 7
122	TYP_REG2	7209	45011	Typ regulatora, kanał 2 warstwa 7
123	TYP_REG3	7309	45111	Typ regulatora, kanał 3 warstwa 7
124	TYP_REG4	7409	45211	Typ regulatora, kanał 4 warstwa 7
125	W8_WSP_A	8107	45809	Współczynnik przetwarzania A dla warstwy 8 (wartość wyjściowa bloku PID)
126	WARSTWA_SKAD_CV_AO1	9102	46002	Warstwa z której pochodzi sygnał wejściowy AO, kanał 1 warstwa 9
127	WARSTWA_SKAD_PV_REG1	7102	44902	Warstwa z której pochodzi sygnał PV regulatora, kanał 1 warstwa 7
128	WARSTWA_SKAD_PV_REG2	7202	45002	Warstwa z której pochodzi sygnał PV regulatora, kanał 2 warstwa 7
129	WARSTWA_SKAD_SP_REG2	7205	45005	Warstwa z której pochodzi sygnał PV regulatora, kanał 3 warstwa 7
130	WARSTWA_SKAD_SP_REG3	7305	45105	Warstwa z której pochodzi sygnał PV regulatora, kanał 3 warstwa 7
131	WARSTWA_SKAD_SP_REG4	7405	45205	Warstwa z której pochodzi sygnał PV regulatora, kanał 4 warstwa 7
132	WARTOŚĆ_ZADANA_KANAŁ_1	-	-	Bufor wykresu SP, kanał 2
133	WARTOŚĆ_ZADANA_KANAŁ_2	-	-	Bufor wykresu SP, kanał 1

134	WSP_PRZETW_AI1	1114	40870	Przetwarzania sygnału PV, kanał 1 warstwa 7
135	WSP_PRZETW_AI2	1214	40910	Przetwarzania sygnału PV, kanał 2 warstwa 7
136	WSP_PRZTW_K1_AI1	1115	40871	Wartość współczynnika przetwarzania, kanał 1 warstwa 1
137	WSP_PRZTW_K1_AI2	1215	40911	Wartość współczynnika przetwarzania, kanał 2 warstwa 1
138	WSP_SKAL_K1	7136	44952	wsp dla sterowania ręcznego
139	WSP_SKAL_K2	7236	45052	wsp dla sterowania ręcznego
140	WZMOCNIENIE_OBIEKTU	7155	44977	wzmocnienie_objektu
141	Y_MAX_K1	7147	44965	maksymalna odpowiedz (SAMOSTROJENIE)
142	Y_MIN_K1	7146	44963	minimalna odpowiedz (SAMOSTROJENIE)
143	YS_K1	9112	46012	CV z AO (warstwa 9)
144	ZBIORNIK2_POZIOM	-	RB	Symulacja poziomu w zbiorniku 2

11.2 Załącznik B - instalacja i konfiguracja

Dołączone na płycie CD pliki konfiguracyjne systemu wizualizacji należy importować do systemu iFIX w wersji 4.0 lub wyższej. Oprogramowanie MsOffice powinno być zainstalowane przed instalacją iFIX (MsOffice nie jest wymagany do działania aplikacji).

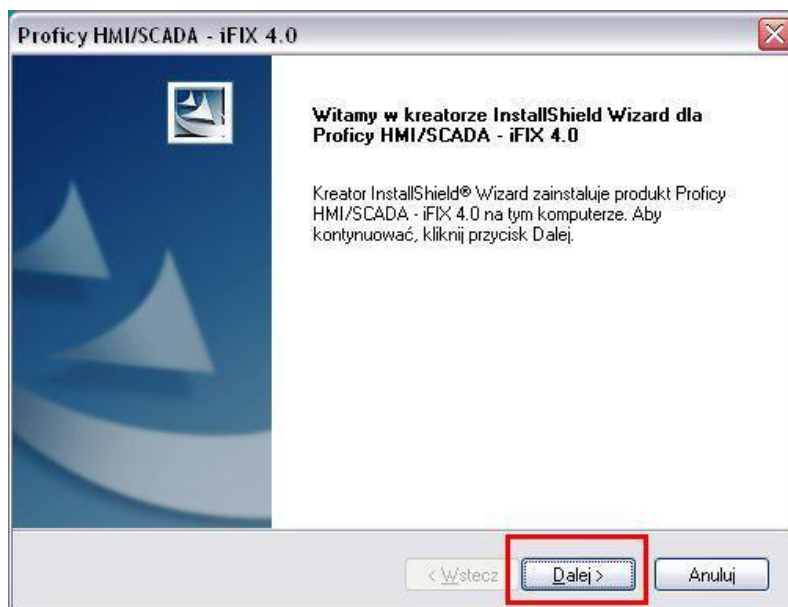
Przed uruchomieniem skonfigurowanej aplikacji należy skonfigurować regulator LB600 według poniższych wskazówek:

- pod adresem 0105 należy wprowadzić wartość 1
- pod adresem 0130 należy wprowadzić wartość 8

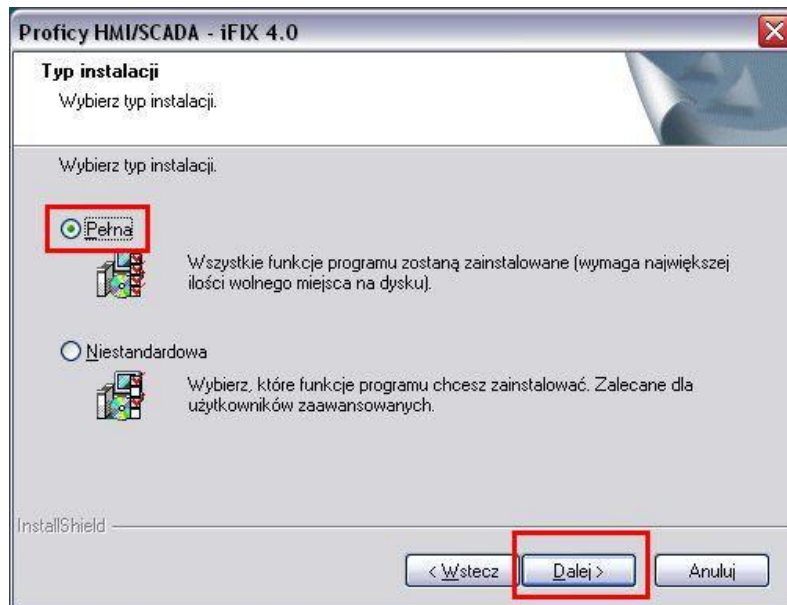
Oraz sprawdzić w systemie Windows, który port COM jest wykorzystywany do komunikacji z regulatorem. W przypadku gdy wykorzystywany port jest inny niż COM1, należy dokonać odpowiedniej modyfikacji na etapie konfiguracji drajwera MB1.

11.2.1 Instalacja Proficy HMI/SCADA iFIX 4.0

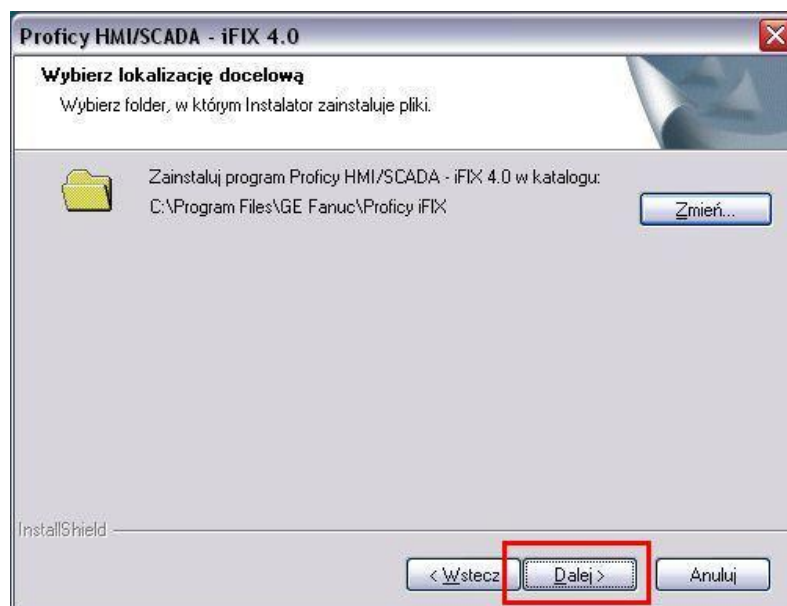
1. Wyświetlić zawartość płyty CD, na której dostarczony został iFIX4.0
2. Wyświetlić zawartość katalogu z wersją instalacyjną iFIX4.0
3. Wybrać **SETUP.EXE**, następnie postępować według wskazówek instalatora.



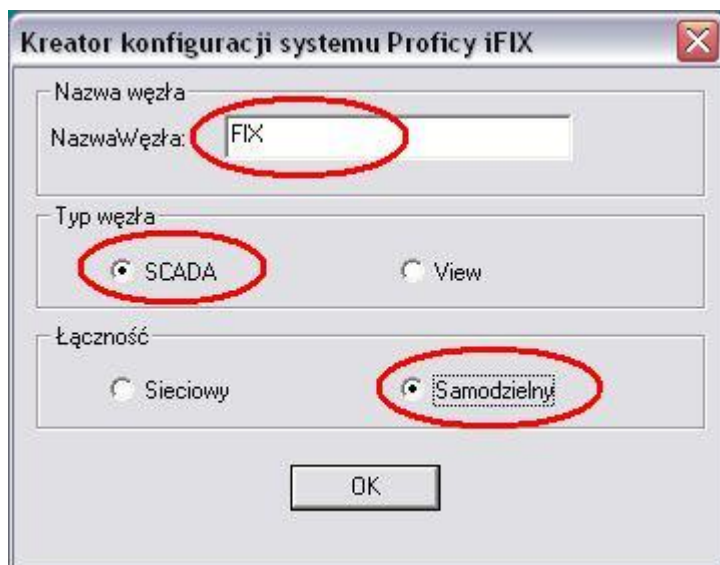
4. Zaznaczyć pełną instalację, następnie wybrać przycisk „Dalej”



5. Wybrać docelowy katalog w którym ma być zainstalowany iFIX 4.0 lub też zaakceptować proponowany, oraz wybrać przycisk „Dalej”



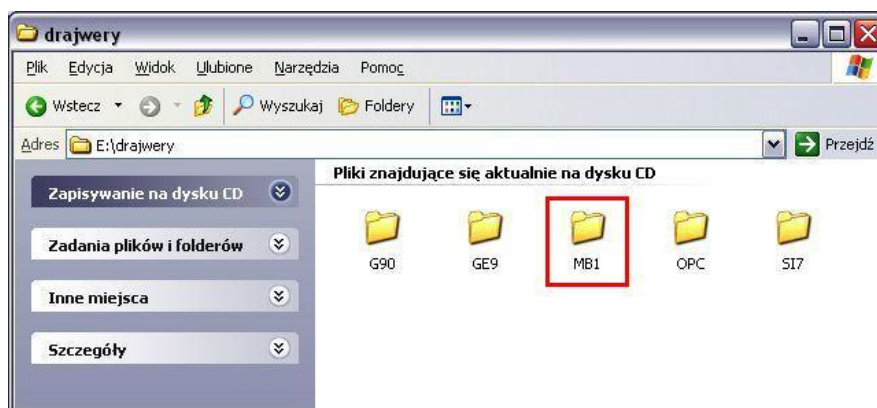
6. W oknie przedstawionym poniżej, wprowadzić nazwę węzła „FIX” oraz zaznaczyć opcję, zgodnie z rysunkiem poniżej i wybrać przycisk „OK.”



7. Zaczekać do końca instalacji.

11.2.2 Instalacja drajwera komunikacyjnego MB1

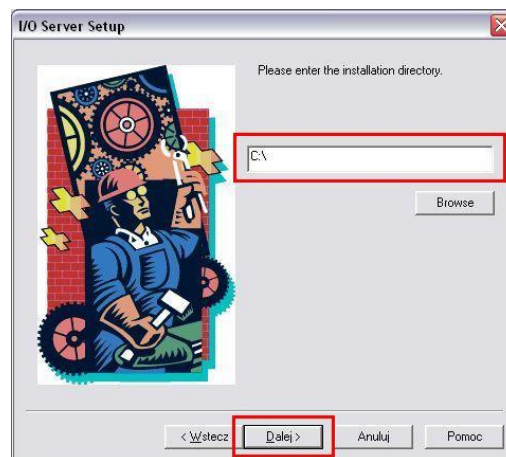
1. Otworzyć katalog „*drajwery*” następnie wybrać katalog „*MB1*”, oraz **SETUP.EXE**,



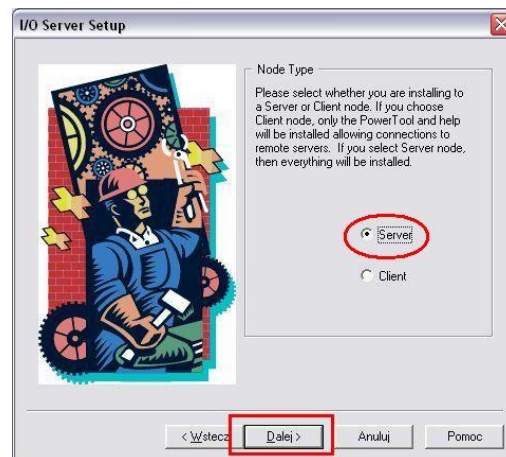
2. Wybrać przycisk „Next”



3. Zaakceptować proponowaną ścieżkę instalacji albo wprowadzić inną, wybrać przycisk „Dalej”



4. Wybrać opcję „Server” i nacisnąć przycisk „Dalej”



5. Określić nazwę grupy w menu „programy” lub zaakceptować proponowaną i wybrać przycisk „Zakończ”

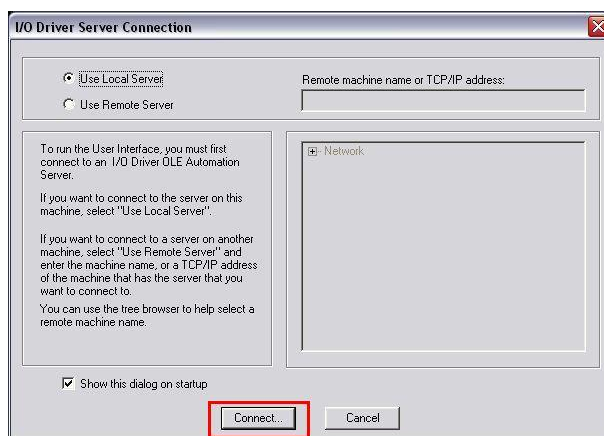


6. Wybrać przycisk „Done”

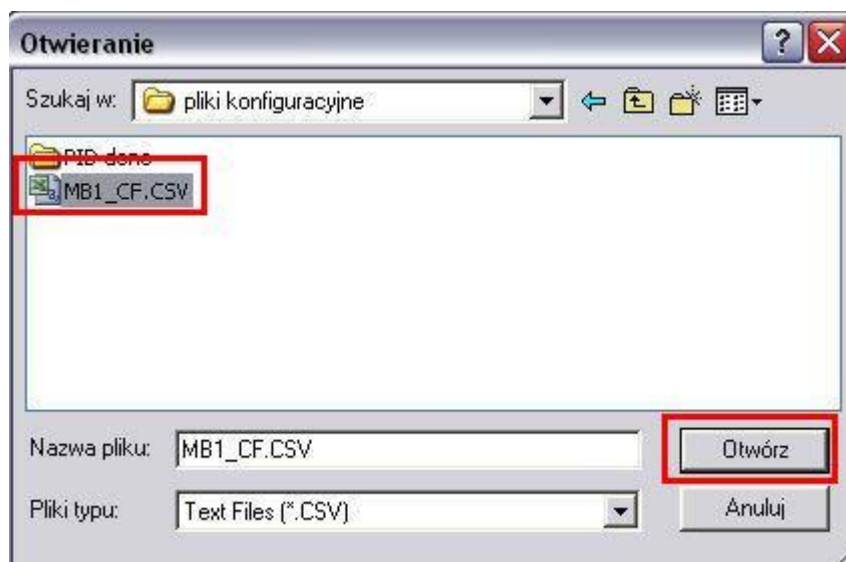


11.2.3 Konfiguracja drajwera MB1

1. Z menu start, uruchomić program „*MB1 Power Tool*” znajdujący się w grupie określonej podczas instalacji drajwera.
2. Zaznaczyć opcję „*Use Local Serwer*” oraz wybrać przycisk „**Connect**”



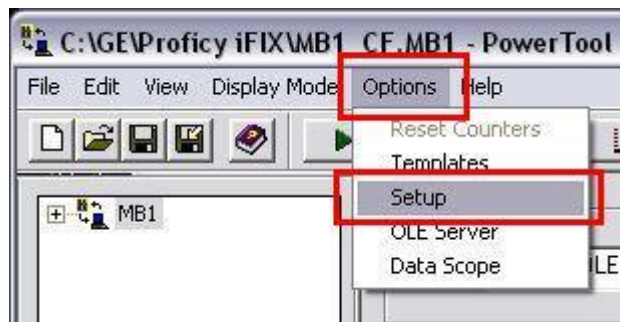
3. Wybrać menu „*File*” a następnie „*Open*”
4. Wskazać dołączony katalog z plikami konfiguracyjnymi, następnie wybrać *typ pliku CSV*, zaznaczyć plik konfiguracyjny drajwera „*MB1_CF.CSV*” oraz wybrać przycisk „**Otwórz**”



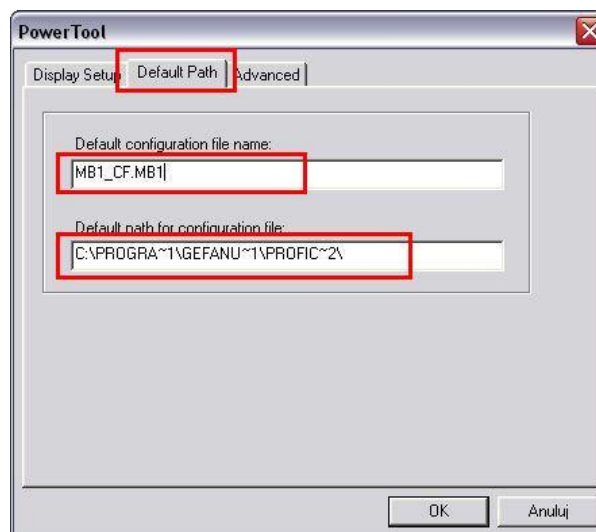
5. Zaimportowaną konfigurację zapisać jako plik MB1, w katalogu, w którym zainstalowano program iFIX4.0.



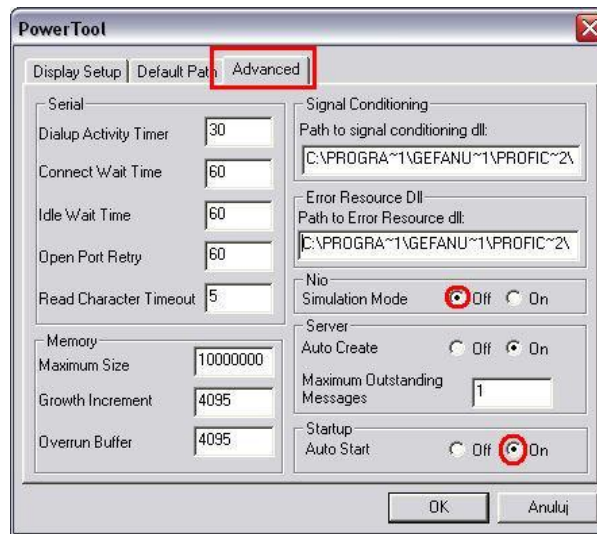
6. Wybrać menu „*Options*” następnie „*Setup*”



7. Wybrać zakładkę „*Default Path*” oraz w pierwszym polu, wprowadzić nazwę zapisanego w punkcie 5, pliku MB1



8. Wybrać zakładkę „**Advanced**” i upewnić się czy zaznaczone są opcje przedstawione na poniższym rysunku

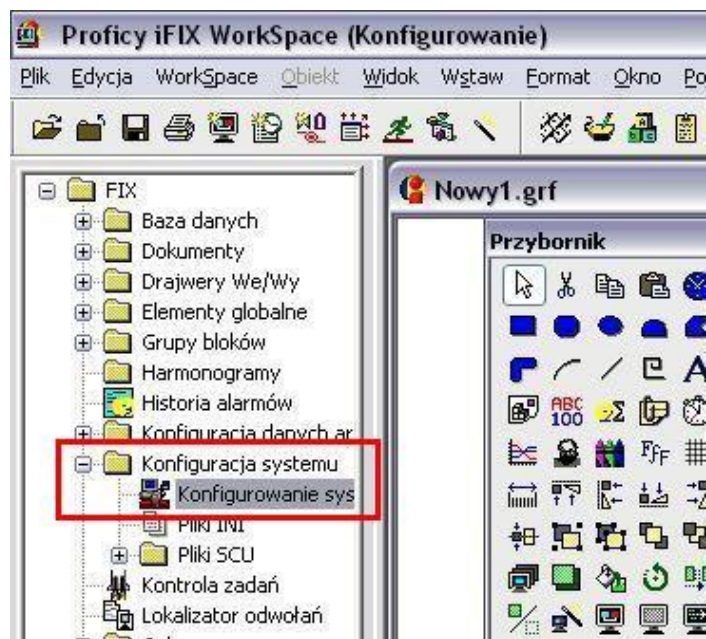


9. Wybrać przycisk „**OK.**” w celu potwierdzenia zmian
10. Wybrać menu „**File**” a następnie „**Save**” oraz zamknąć program „*MB1-Power Tool*”

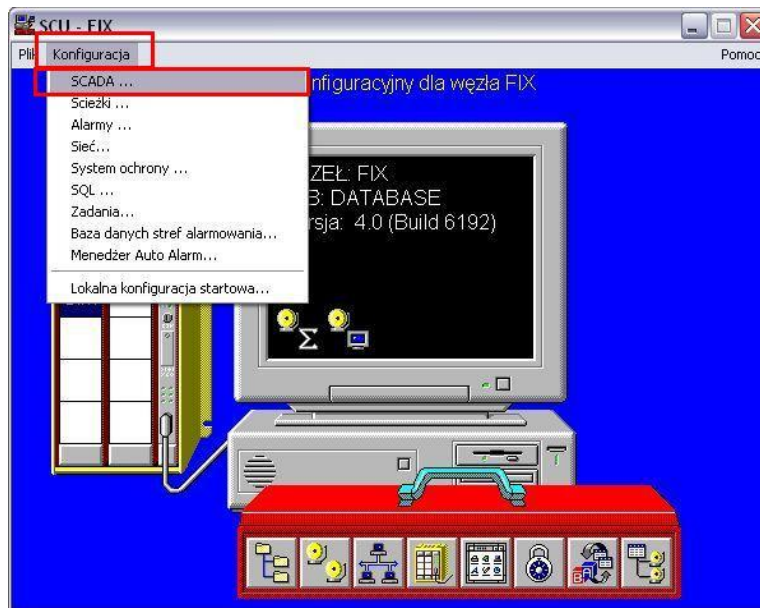
11.2.4 Konfiguracja iFIX4.0

11.2.4.1 Dodanie drajwera MB1

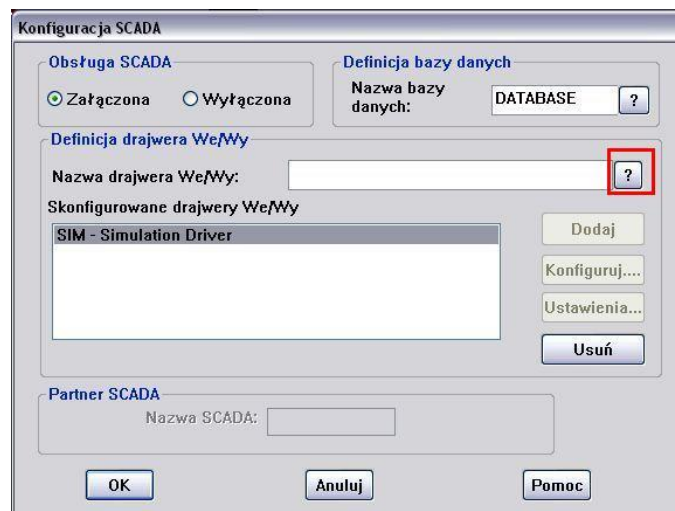
1. W głównym drzewie programu WorkSpace, wybrać „*Konfiguracja systemu*” następnie „*Konfigurowanie sytemu*”



2. W otwartym programie konfiguracji systemu, wybrać menu „*Konfiguracja*” oraz opcję „*SCADA*”



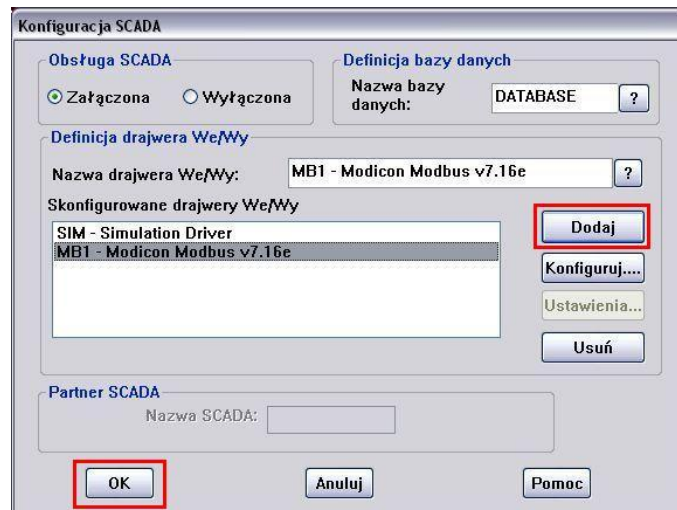
3. W formatce „*Konfiguracja SCADA*” wybrać zaznaczony przycisk



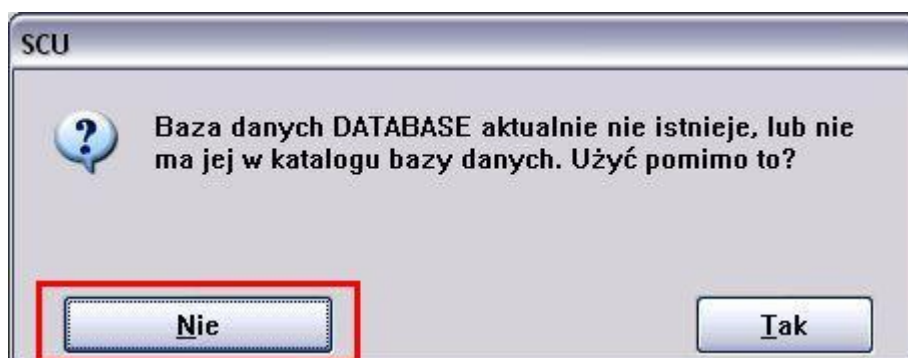
4. W formatce „*Dostępne drajwery*” wskazać drajwer MB1 oraz wybrać przycisk **OK**.



5. Wybrać przycisk „*Dodaj*”, a następnie *OK*.



6. W oknie informacyjnym wybrać przycisk „*NIE*” następnie na formatce „*Konfiguracja SCADA*”, wybrać przycisk „*Anuluj*”



7. Sprawdzić w programie konfiguracji systemu czy został dodany drajwer MB1, zgodnie z poniższym rysunkiem.



8. Z menu „*Plik*” wybrać opcję „*Zapisz*” a następnie „*Zakończ*”.

11.2.4.2 Import bazy danych

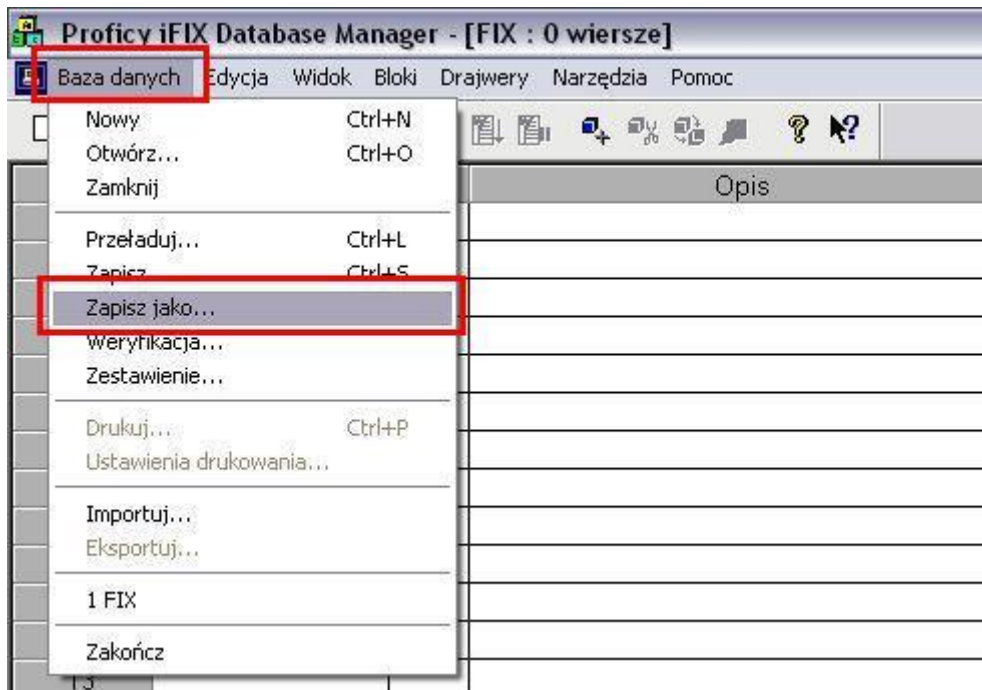
1. W głównym drzewie programu Workspace rozwinąć „*Baza danych*” oraz wybrać program *Konstruktor bazy danych*.



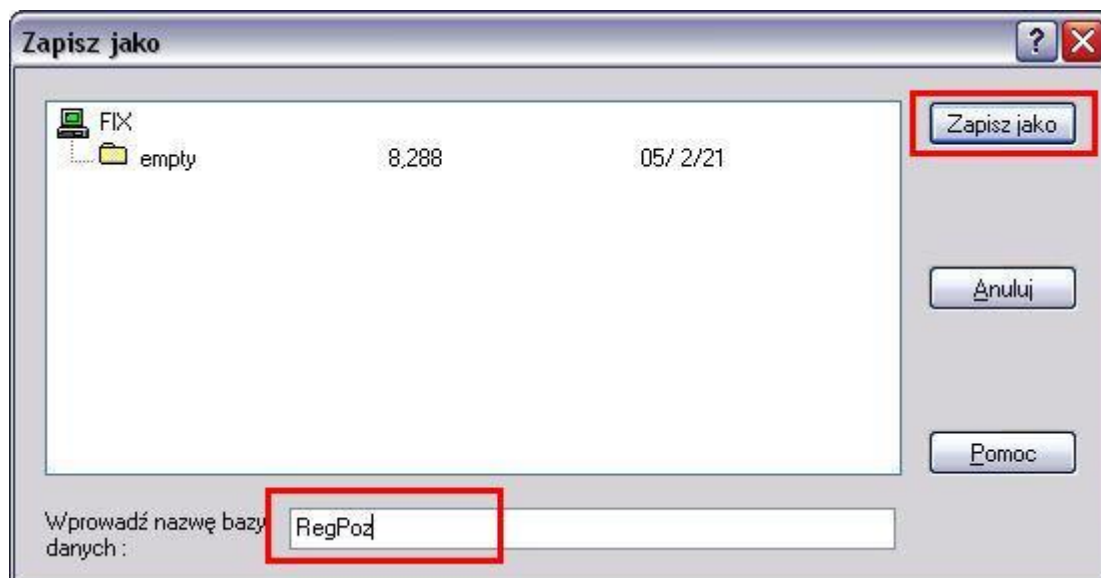
2. W oknie powitalnym zaznaczyć opcję „*Otwórz węzeł lokalny*” oraz wybrać przycisk *OK*.



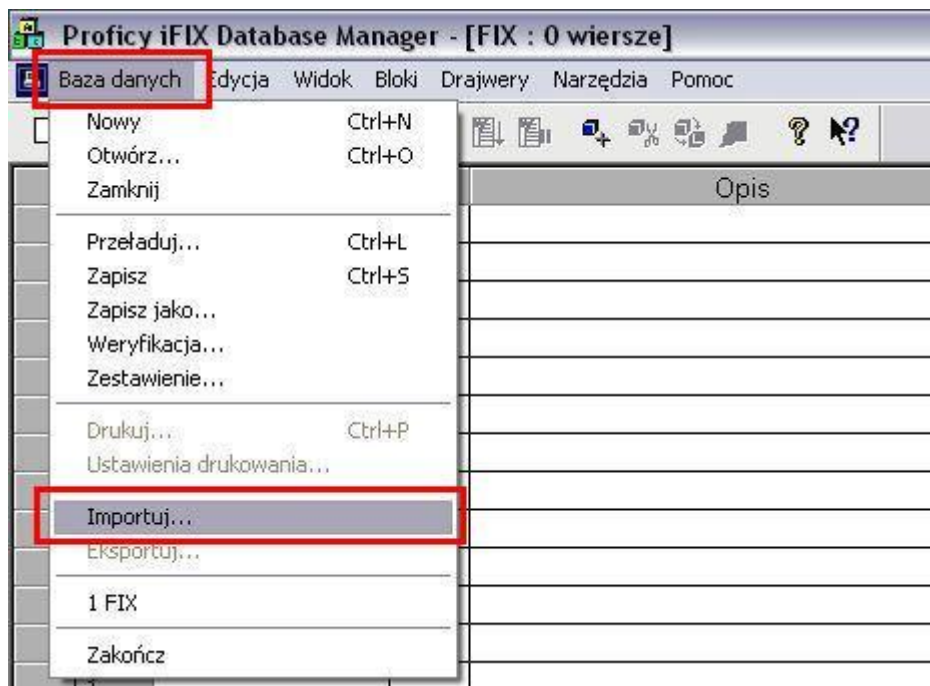
3. W programie *Database Manager*, wybrać menu „*Baza danych*”, następnie opcję „*Zapisz jako*”



4. Wprowadzić nazwę bazy danych oraz wybrać przycisk „Zapisz jako”



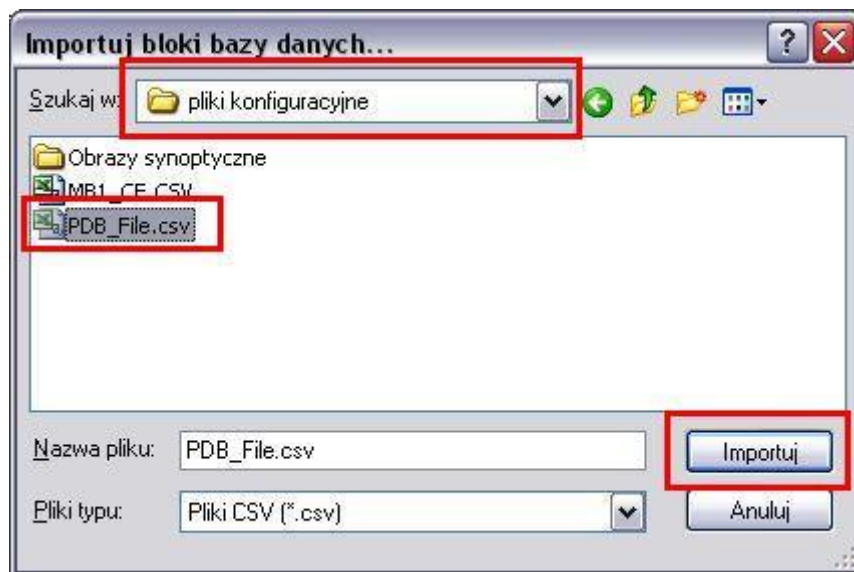
5. W programie *Database Manager*, wybrać menu „*Baza danych*”, następnie opcję „*Importuj...*”



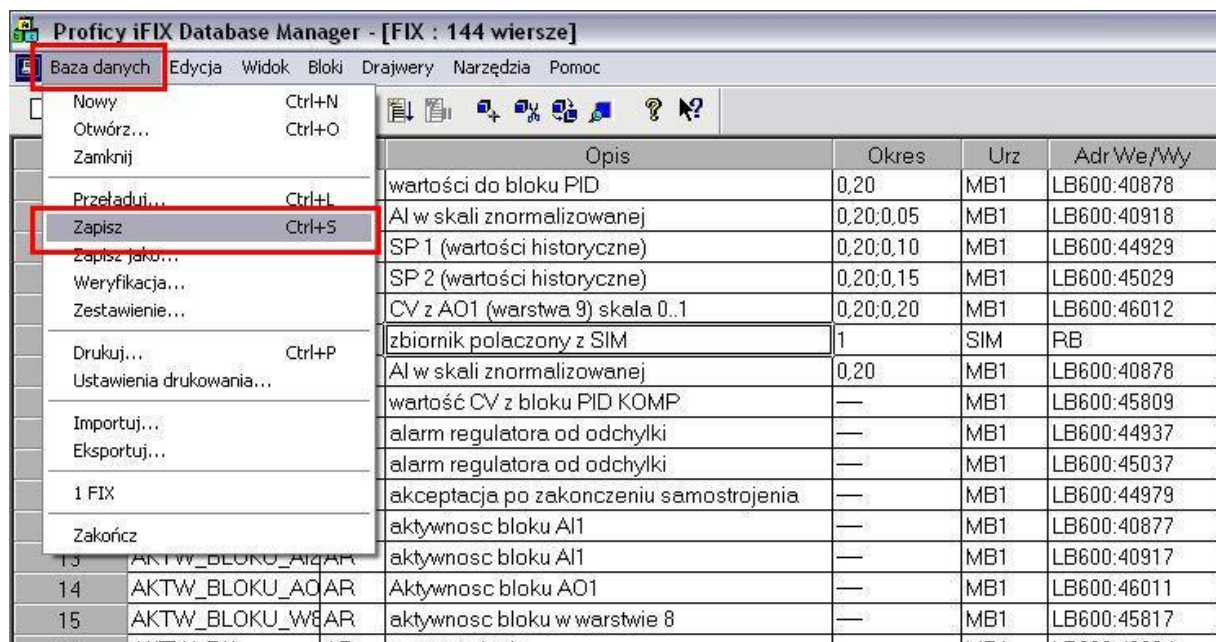
6. Potwierdzić komunikat informacyjny.



7. Wskazać katalog z plikami konfiguracyjnymi, wybrać typ pliku CSV, zaznaczyć plik PDB_File.CSV, wybrać „**Importuj**”.



8. W programie *Database Manager*, wybrać menu „**Baza danych**”, następnie opcję „**Zapisz**” oraz „**Zakończ**”

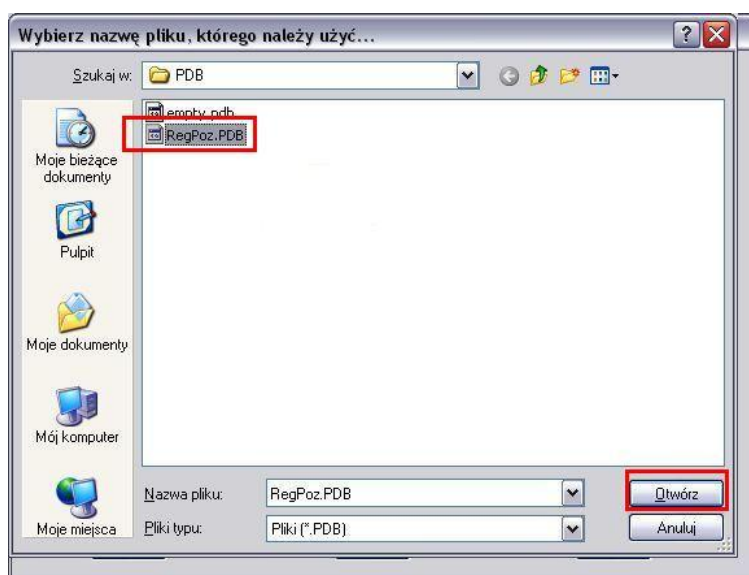


9. W głównym drzewie programu WorkSpace rozwinąć *konfigurację systemu* następnie wybrać program „**Konfiguracja systemu**”

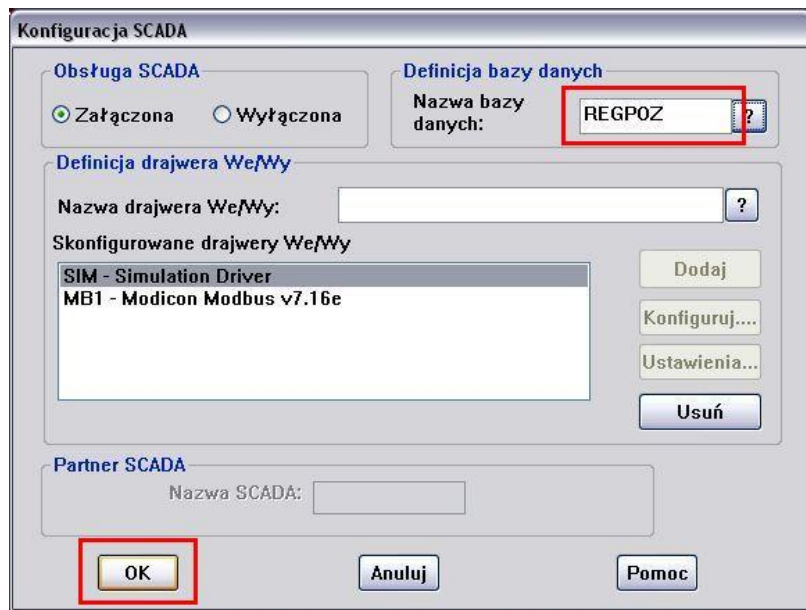
10. Wybrać przycisk zaznaczony na rysunku



11. Wskazać plik PDB z utworzoną w punkcie 4, oraz wybrać przycisk „*Otwórz*”



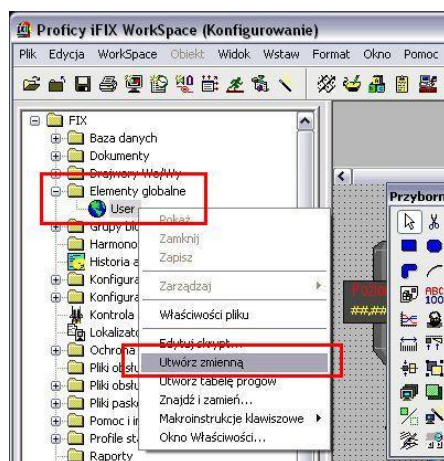
12. Upewnić się czy program wskazuje poprawny plik, następnie wybrać przycisk OK.



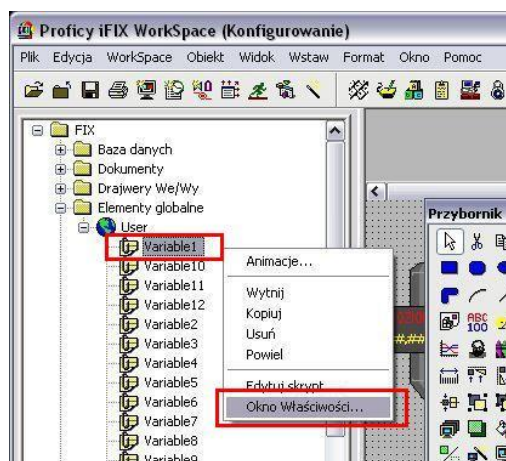
13. Zamknąć program „*Konfiguracja systemu*”

11.2.4.3 Dodanie oraz konfiguracja zmiennych globalnych

1. W głównym drzewie programu WorkSpace, wybrać „*Elementy globalne*”, następnie kliknąć prawym klawiszem myszy na „*User*” i z menu kontekstowego wybrać opcję „*Utwórz zmienną*”

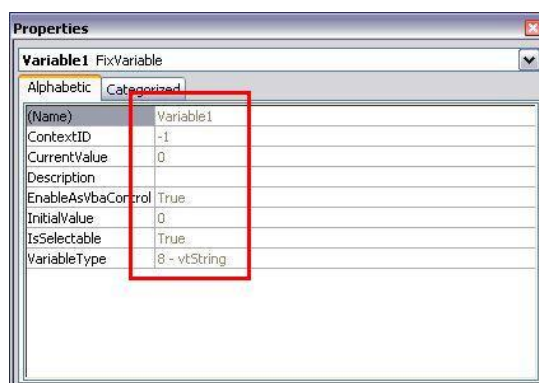


2. Utworzyć 12 zmiennych
3. Zaznaczyć pierwszą zmienną, i z menu kontekstowego wybrać opcję „*Okno właściwości*”

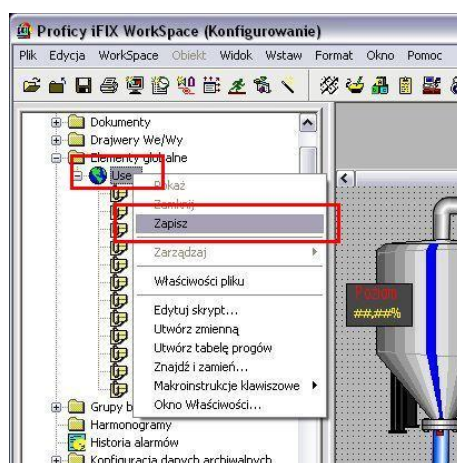


4. W oknie właściwości skonfigurować pola, zmiennych , według poniższej tabeli.

Lp.	Name	Enable As Vba Control	Initial Value	Variable type
1	AktywnyRys	True	0	8
2	AutoReg	True	0	2
3	AutoTuningStop	True	True	11
4	BlockFnctButton	True	False	11
5	Struktura	True	0	2
6	CvRegK1	True	0	4
7	Kp	True	0	4
8	ManualReg1	True	0	4
9	ManualReg2	True	0	4
10	SamostrojenieAktywne	True	False	11
11	Td	True	0	4
12	Ti	True	0	4

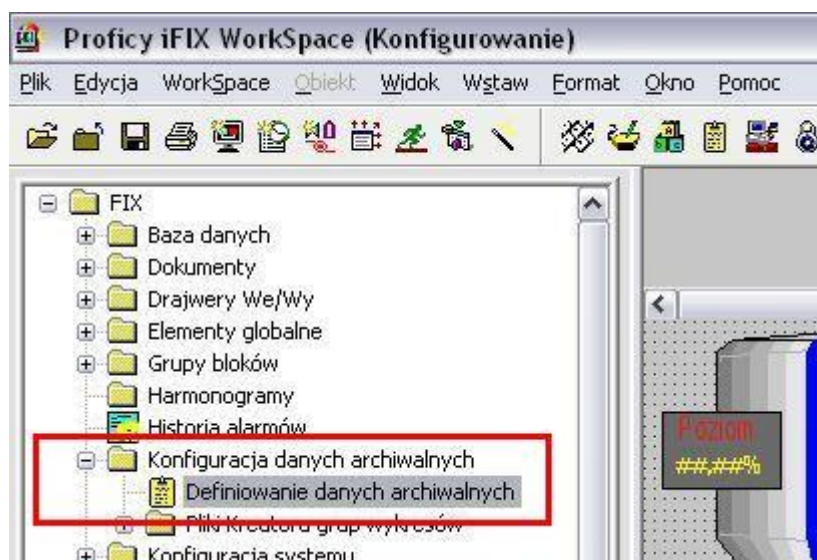


5. Wybrać menu kontekstowe „User” a następnie wybrać opcję „Zapisz”

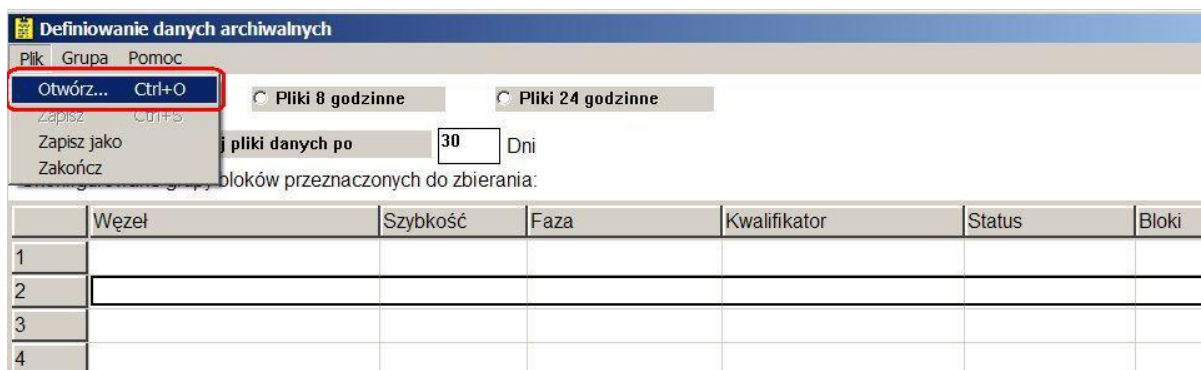


11.2.4.4 Konfiguracja danych archiwalnych

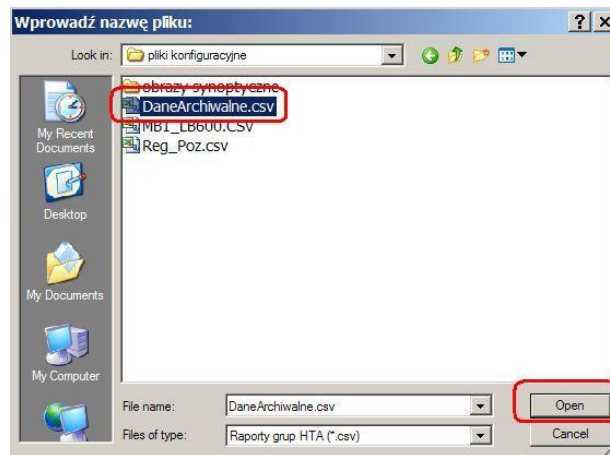
1. W głównym drzewie programu WorkSpace wybrać „*Konfiguracja danych archiwalnych*” a następnie „*Definiowanie danych archiwalnych*”



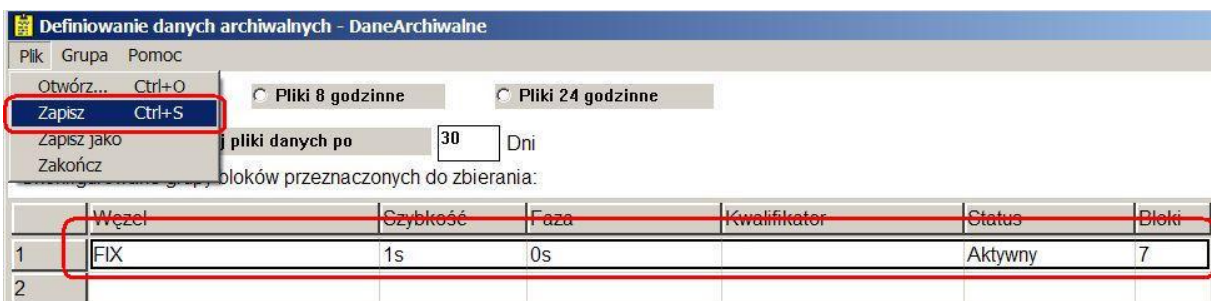
2. W otwartym oknie wybrać menu „*Plik*” a następnie „*Otwórz*”



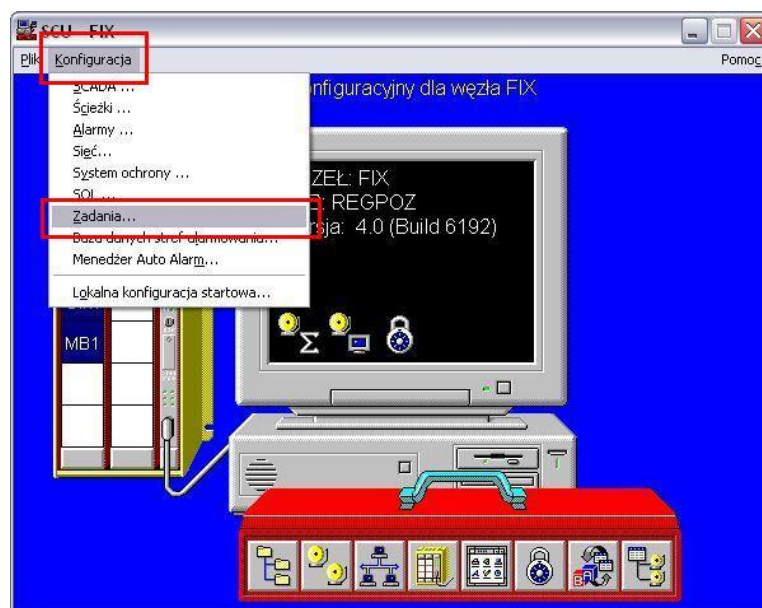
3. Wskazać katalog z plikami konfiguracyjnymi, następnie zaznaczyć plik DaneArchiwalne.csv i wybrać przycisk „*Otwórz*”



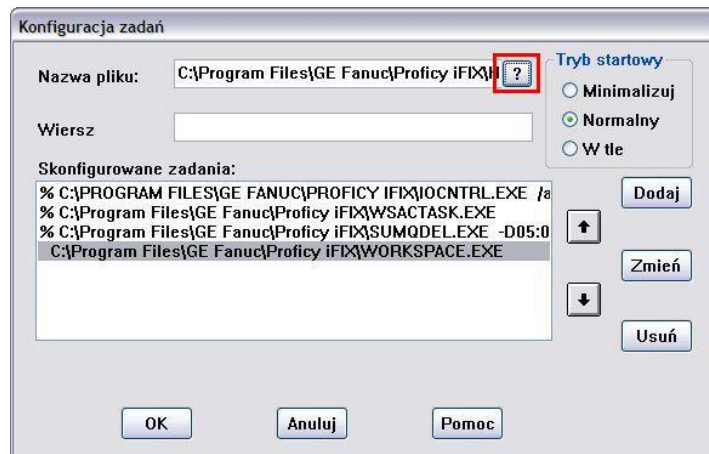
4. Sprawdzić czy dane zostały zaimportowane, następnie wybrać menu „*Plik*” i przycisk „*Zapisz*”



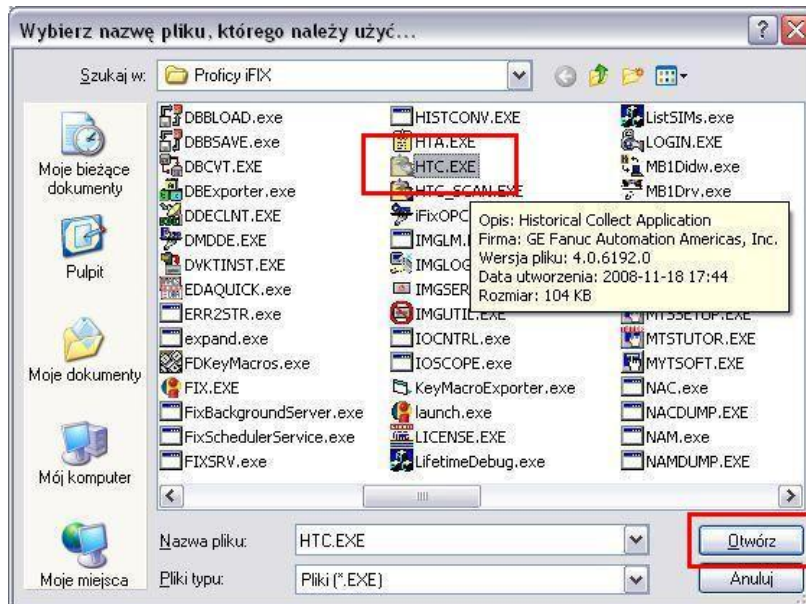
5. W głównym drzewie programu Workspace, wybrać konfigurację systemu
6. W programie konfiguracyjnym wybrać menu „*Konfiguracja*” oraz opcję „*Zadania*”



7. Wybrać przycisk zaznaczony na rysunku



8. Wskazać program **HTC.EXE** oraz wybrać przycisk „*Otwórz*”



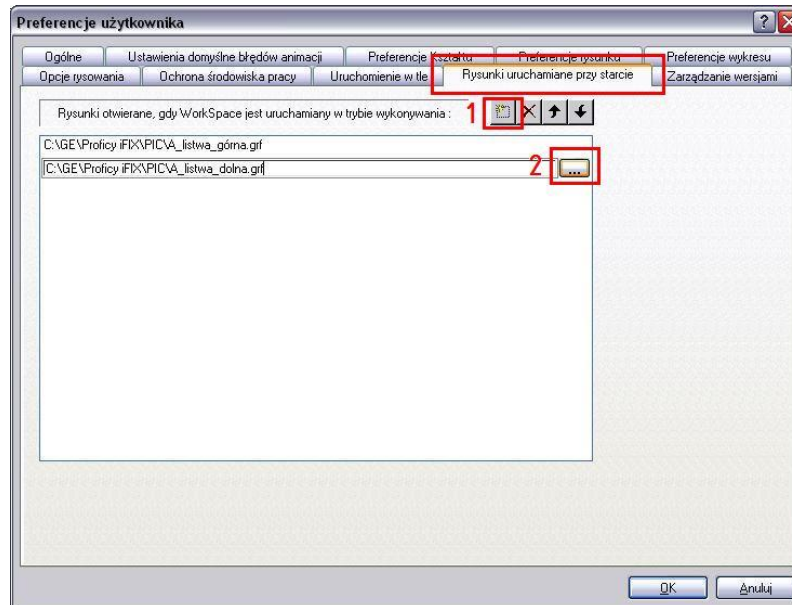
9. Zaznaczyć opcje „*W tle*”, wybrać przycisk „*Dodaj*” oraz zatwierdzić wprowadzone zmiany przyciskiem **OK**.



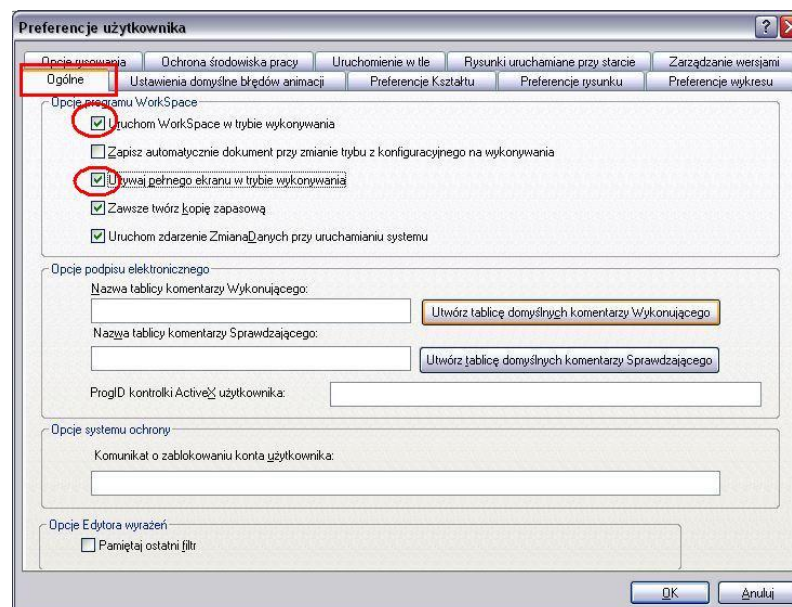
10. W programie konfiguracyjnym wybrać menu „*Plik*” następnie opcję „*Zapisz*” oraz „*Zakończ*”

11.2.4.5 Konfiguracja trybu wykonywalnego programu WorkSpace

1. W programie WorkSpace w trybie konfiguracyjnym wybrać menu „*WorkSpace*”, a następnie opcję „*Preferencje*”
2. Wybrać zakładkę „*Rysunki uruchamiane przy starcie*”, oraz za pomocą przycisków (1) i (2) wybrać rysunki przedstawione poniżej.

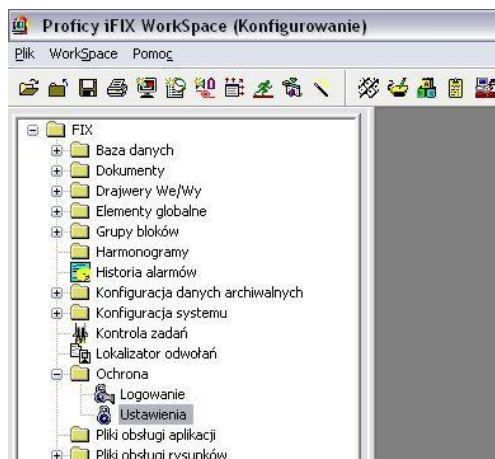


3. Wybrać zakładkę ogólne i zaznaczyć opcje pokazane poniżej

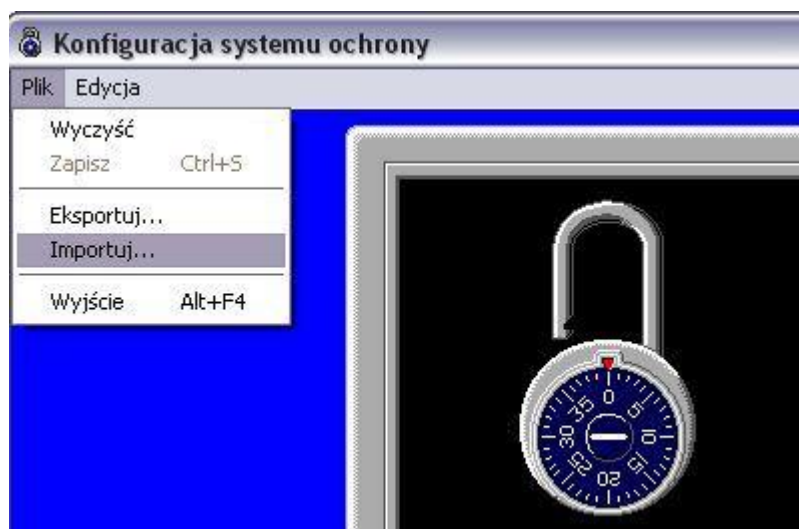


11.2.4.6 Konfiguracja zabezpieczeń

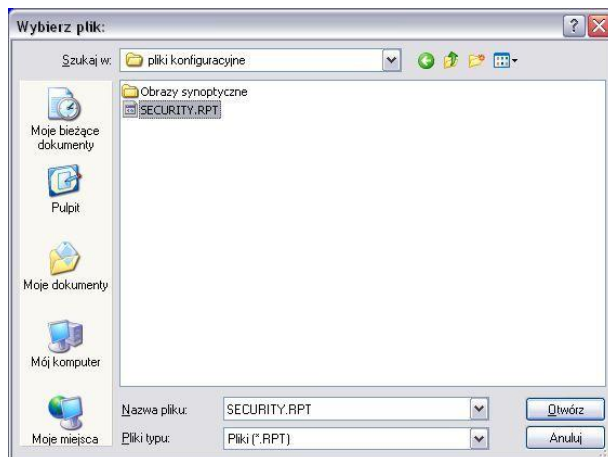
1. W drzewie programu WorkSpace, wskazać „*Ochrona*”, następnie „*Ustawienia*”



2. W programie „*Konfiguracja systemu ochrony*” wybrać menu „*Plik*” a następnie opcję „*Importuj...*”



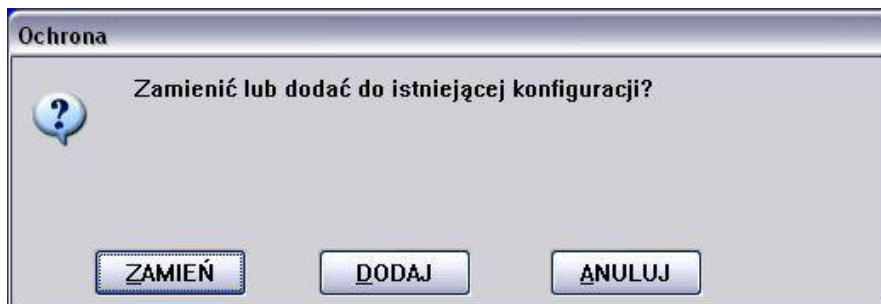
3. Wskazać plik *SECURITY.RPT* znajdujący się w dostarczonych plikach konfiguracyjnych



4. Zatwierdzić przyciskiem „*Tak*” poniższy komunikat



5. Na poniższym oknie wybrać przycisk „*Zmień*”



6. Wybrać menu „*Plik*” następnie „*Zapisz*” oraz „*Wyjście*”. Na poniższym rysunku przedstawiono sytuację, w której system ochrony jest prawidłowo skonfigurowany.



Ważne !

Podczas powyższej konfiguracji są importowane następujące konta użytkowników:

Administrator systemu (login: Admin, hasło: zac@hicz)

User (login: user, hasło 123456)

Gość (login: Gość, hasło 123456)

Technical Support (login: tchsup, hasło: lucky), **nie należy zmieniać hasła.**