

PODRĘCZNIK UŻYTKOWNIKA

# TRACE MODE 6

SOFTLOGIC SCADA/HMI MES EAM HRM

Zintegrowana platforma obejmująca wszystkie poziomy automatyzacji produkcji

SZYBKI START

**AdAstrA Research Group, Ltd.**

Autoryzowany dealer:



ul. Sulimy 1, 82-300 Elbląg  
tel. +48 512-468-634, +48 501-403-220  
[www.test-automatyka.pl](http://www.test-automatyka.pl)  
e-mail: [scada@test-automatyka.pl](mailto:scada@test-automatyka.pl)

## Znaki zastrzeżone

**IBM i IBM Personal Computer** są zarejestrowanymi znakami handlowymi International Business Machines Corporation.

**Microsoft, Windows, Windows NT, Windows CE i MS-DOS** są zarejestrowanymi znakami handlowymi Microsoft Corporation.

**Intel** jest zarejestrowanym znakiem handlowym Intel Corporation.

**Micro PC** jest zarejestrowanym znakiem handlowym OCTAGON SYSTEMS.

**ADAM i MIC-2000** są zarejestrowanymi znakami handlowymi Advantech.

**TRACE MODE** jest zarejestrowanym znakiem handlowym Adastra Research Group, Ltd.

**T-FACTORY.exe** jest zarejestrowanym znakiem handlowym Adastra Research Group, Ltd.

**ADASTRA** jest zarejestrowanym znakiem handlowym Adastra Research Group, Ltd.

**Autobuilding** jest zarejestrowanym znakiem handlowym Adastra Research Group, Ltd.

Wszystkie nazwy i znaki handlowe innych firm są zarejestrowane; należą do prawowitych właścicieli.

Prawa autorskie © 2005 AdAstrA Research Group, Ltd.

Wszelkie prawa zastrzeżone.

# Spis treści

<b>WSTĘP .....</b>	<b>5</b>
Krótki opis.....	5
Wprowadzenie .....	5
Zastosowanie.....	5
Struktura.....	6
Podstawy systemu .....	6
Instrukcja instalacji .....	7
Wymagania systemowe.....	7
Kolejność instalacji .....	7
<b>Szybki start .....</b>	<b>8</b>
Tworzenie najprostszego projektu .....	8
Dodawanie węzła stacji roboczej .....	8
Projektowanie ekranu graficznego HMI.....	9
Autobuilding kanałów .....	14
Dodawanie generatora sinusoidy i łączenie go z kanałem.....	15
Uruchomienie projektu.....	17
Dodawanie funkcji sterowania .....	17
Edycja ekranu graficznego .....	17
Łączenie argumentów ekranu z kanałem.....	21
Trendy .....	22
Uruchomienie projektu.....	25
Uprozczone przetwarzanie danych.....	26
Dopracowanie ekranu graficznego .....	26
Pisanie programu w Techno ST .....	27
Łączenie argumentów programu .....	29
Uruchomienie projektu.....	30
Dynamic Data Exchange (DDE) na przykładzie MS Excel .....	31
RTM jako serwer DDE .....	31
RTM jako klient DDE.....	32
Podłączanie rozproszonych modułów I/O.....	33
Dodawanie elementów źródłowych do wprowadzania danych z I-7011 .....	34
Dodanie oraz konfiguracja portu COM .....	35
Łączenie kanału ze źródłem danych.....	37
Uruchomienie projektu.....	37
Human Machine Interface: Monitoring i sterowanie.....	38
Dodawanie ekranów HMI .....	40
Tworzenie programu.....	64
Węzły projektu oraz kanały.....	69
Archiwum oraz alarmy .....	72
Podłączanie PLC do stacji roboczej .....	79
Konfiguracja portu COM .....	79
Dodawanie Sources/Receivers do wymiany danych przez protokół ModBus RTU.....	80
Łączenie Sources/Receivers z kanałami .....	82

Dodawanie kanałów do kontrolera PAC .....	84
Dodawanie Sources/Receivers do kontrolera PAC .....	86
Łączenie Sources/Receivers z kanałami .....	87
Konfigurowanie kanałów, ustawianie początkowych warunków sterowania .....	88
Parametry pracy w sieci i konfiguracja charakterystyk dynamicznych węzła .....	88
Praca w sieci .....	89
Konfiguracja wymiany danych w sieci .....	91
Edycja kanałów .....	93
Wyświetlanie czasu w HMI .....	93
Zdarzenia .....	94
Komunikacja z MS Access .....	96
Przetwarzanie danych z lokalnego archiwum .....	105
Dodawanie użytkowników .....	108
Symulatory .....	109
Przygotowania .....	109
Tworzenie programów symulujących i dodawanie ich do projektu .....	110
Pliki projektu .....	115
Debuggowanie .....	116
Konfiguracja protokołu sieciowego .....	117
Przygotowanie folderu projektu do debuggowania .....	118
Debuggowanie .....	119
Wnioski .....	122

# WSTĘP

## Krótki opis

### Wprowadzenie

TRACE MODE 6 development system jest zaprojektowany do automatyzacji przedsiębiorstw produkcyjnych, elektrowni, inteligentnych budynków, systemów transportu, itp. System sterowania zrealizowany w TRACE MODE 6 może mieć dowolny rozmiar – od pojedynczych sterowników i HMI do rozproszonego sterowania procesem lub zdalnych systemów sterowania, zarządzania środkami trwałymi i MES. TRACE MODE zapewnia całkowitą automatyzację procesów zarówno technologicznych jak i biznesowych (moduły T-Factory.exe), pozwalając uzyskać wysoką wydajność kosztową oraz szybki zwrot inwestycji.

### Zastosowanie

TRACE MODE 6 posiada wbudowane drivery, które pozwalają komunikować się z ponad 2000 urządzeń I/O - PLC, zdalne urządzenia I/O, karty I/O oraz sieci przemysłowe.

TRACE MODE 6 może być wykorzystany do tworzenia różnych systemów automatyki, do monitoringu i sterownia, w postaci różnych architektur - scentralizowanej, rozproszonej i mieszanej. Szczególną uwagę zwrócono na systemy zbudowane przy użyciu kontrolerów PAC, ponieważ dla projektów tego typu TRACE MODE 6 posiada *jednolity zestaw narzędzi* do tworzenia HMI operatora oraz programowania sterownika. Technologia **Autobuilding** oraz projektowanie rozproszonych systemów sterowania w ramach **integrated development environment** znacznie zwiększają efektywność projektanta, ograniczają ilość żmudnych, manualnych operacji, minimalizując liczbę błędów nieuniknionych w dużych projektach.

Wymiana danych w sieci automatycznych systemów sterowania, zrealizowanych przy użyciu TRACE MODE 6 odbywa się przy użyciu protokołu *I-Net* przez TCP/IP. Archiwizacja parametrów realizowana jest przez zaawansowany system zarządzania bazą danych – SIAD/SQL 6. Wbudowany w TRACE MODE Integrated Development Environment system redundancji jest dostępny dla wszystkich elementów rozproszonego systemu sterowania (PLC, HMI, EAM, loggery itp.). Elastyczność i niezawodność systemów sterowania zbudowanych w oparciu o TRACE MODE 6 pozwalają na zastosowanie ich w takich gałęziach przemysłu jak energetyka jądrowa, przemysł petrochemiczny, hutnictwo, energetyka, inżynieria mechaniczna, inteligentne budynki, przemysł spożywczy, transport jak również badania naukowe.

## Struktura


TRACE MODE 6 development system składa się z:

- integrated development environment - **tmdevenv.exe**;
- debugger z obsługą ekranów graficznych – **rtc.exe**;
- debugger bez obsługi ekranów graficznych – **rtmg32.exe**.

## Podstawy systemu

Do realizacji projektów automatycznych systemów sterowania wykorzystuje się jednolity TRACE MODE 6 *integrated development environment* (IDE), pracujący na systemie operacyjnym MS Windows. Tworzenie elementów projektu, edycja oraz łączenie odbywa się w *Nawigatorze Projektu*. Projekt jest przedstawiony jako *drzewo komponentów*, a budowanie takiego drzewa jest znacznie uproszczone dzięki technologii *autobuilding*.

Gdy tworzony jest szablon programu, ekranu, raportu lub innego elementu użytkownika, uruchamia się odpowiedni edytor w TRACE MODE 6 *integrated development environment*. Utworzone szablony mogą zostać ponownie wykorzystane w obecnym projekcie lub w kolejnych projektach, pod warunkiem, że są zapisane w bibliotece komponentów użytkownika.

Uruchamianie TRACE MODE 6 *integrated development environment* (**tmdevenv.exe**) odbywa się przez dwukrotne kliknięcie lewym przyciskiem myszy na ikonę  znajdującą się na pulpicie Windows lub przez menu Windows «**START/Wszystkie programy/Trace Mode 6/TRACE MODE IDE 6**».

TRACE MODE 6 development system generuje zestaw plików, które są wykorzystywane przez automatyczne systemy sterowania w modułach *real time monitor* (RTM) na stacjach roboczych HMI oraz w sterownikach. W poniższych przykładach zostaną użyte debuggery z obsługą ekranów graficznych - **rtc.exe** oraz bez obsługi ekranów graficznych **rtmg32.exe** (oba te pliki wykonywalne znajdują się w folderze TRACE MODE 6). Debugger pozwala uruchomić jeden węzeł projektu na komputerze z zainstalowanym TRACE MODE 6 development system. Projekty mogą być uruchamiane w debuggerach TRACE MODE 6 na różnych komputerach połączonych ze sobą siecią LAN. Takie podejście pozwala debuggować rozproszone systemy sterowania.

W przypadku korzystania z TRACE MODE 6 development system w wersji base, czas ciągłej pracy debuggera z węzłem komunikującym się z urządzeniami I/O jest ograniczony do *jednej* godziny oraz do *24 godzin* bez urządzeń I/O. Wersja professional pozwala na pracę odpowiednio *3 godziny* i *3 dni*. Po upływie tego czasu projekt może być ponownie uruchomiony dowolną ilość razy.

Pierwszym krokiem do rozpoczęcia pracy z TRACE MODE 6 jest zainstalowanie development system na komputerze.

---

## Instrukcja instalacji

### Wymagania systemowe

Do komfortowej pracy z TRACE MODE 6 development system zalecana jest następująca konfiguracja:

- System operacyjny – Windows XP;
- Procesor – PENTIUM 4;
- RAM – 512 MB;
- Wolne miejsce na dysku – 1.5 GB;
- Rozdzielczość ekranu – 1280x1024;
- Jakość kolorów – True Color;
- Obsługa OpenGL v.1.1;
- napęd CD-ROM, mysz, port równoległy lub USB.

### Kolejność instalacji

W zależności od typu TRACE MODE 6 development system, dostępne są następujące opcje instalacji:

- Dla wersji professional – należy włożyć elektroniczny klucz zabezpieczający do portu równoległego lub portu USB w komputerze i postępować zgodnie z poleceniami **Instrukcji instalacji**;
- Dla wersji basic – niewymagany jest klucz elektroniczny.

Instalacja wersji basic:

- Zamknij wszystkie działające aplikacje;
- Włóż CD z TRACE MODE 6 development system do napędu CD i postępuj zgodnie z instrukcjami na ekranie.

---

Prosimy zapoznać się z aktualnymi zaleceniami instalacyjnymi zawartymi w pliku **readme.txt**, który można znaleźć w głównym folderze TRACE MODE 6 development system na oryginalnej płycie CD.

---



## Szybki start

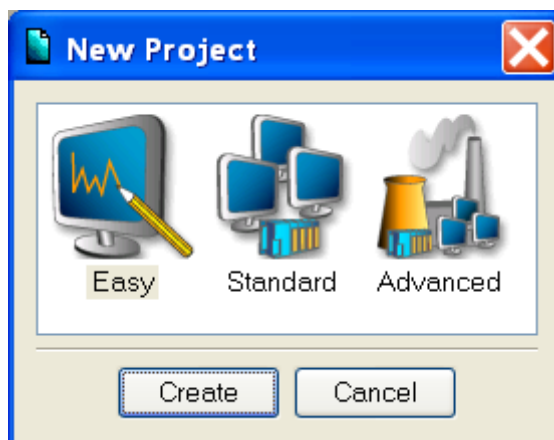
Ta sekcja pozwoli rozpocząć pracę w integrated development environment bez konieczności szczegółowego studiowania całej dokumentacji technicznej. W sekcji SZYBKI START znajduje się kilka projektów przewidzianych do nauki – możesz rozpocząć wstępne zapoznanie się z TRACE MODE 6 przy użyciu dowolnego z nich.

### Tworzenie najprostszego projektu

Celem tego ćwiczenia jest stworzenie HMI (human-machine interface) systemu monitoringu, zawierającego jedną stację roboczą. Wykorzystana do tego zostanie technologia *autobuilding*. Następnie wprowadzone zostaną funkcje sterowania, współpracy z aplikacjami Windows przy użyciu protokołu wymiany danych DDE, a na zakończenie dowiesz się jak dołączyć sygnał analogowy ze zdalnego modułu.

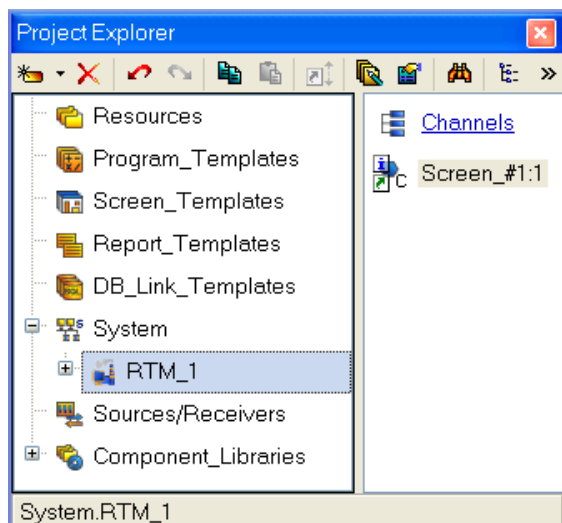
### Dodawanie węzła stacji roboczej

Uruchom development system klikając dwukrotnie na ikonę , znajdującą się na pulpicie Windows i stwórz nowy projekt klikając na ikonę .



W wyświetlonym oknie dialogowym wybierz styl **Easy**. Po kliknięciu lewym przyciskiem myszy na przycisk **Create**, pojawi się drzewo projektu z utworzonym węzłem stacji roboczej (**RTM\_1**). Otwórz węzeł **RTM\_1** klikając go dwukrotnie. Zawartość węzła zostanie wyświetlona po prawej stronie okna Explorer; jest to pusta grupa **Channels** i kanał **CALL**, **Screen#1**, przeznaczony do wyświetlania ekranu graficznego w węźle stacji roboczej;






## Projektowanie ekranu graficznego HMI

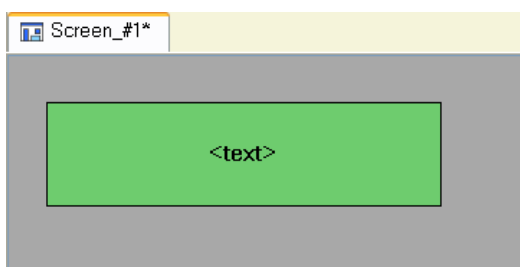
Kliknij dwukrotnie na **Screen#1**, aby otworzyć edytor graficzny.


### *Dodawanie statycznego tekstu*

Umieść statyczny tekst – **Parameter Value** w lewym, górnym rogu ekranu.

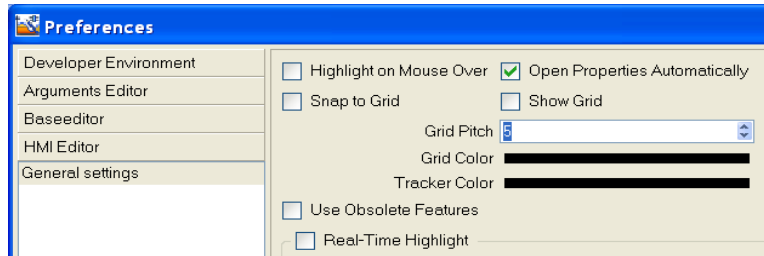
Aby to zrobić:

- Wybierz z paska narzędzi ikonę elementu graficznego (GE) – ;
- umieść prostokątny GE w arkuszu edytora – aby to zrobić:
- ustaw “punkt zaczepienia” klikając lewym przyciskiem myszy;
- ustaw rozmiar prostokąta przeciągając kursorem do wymaganej wielkości;

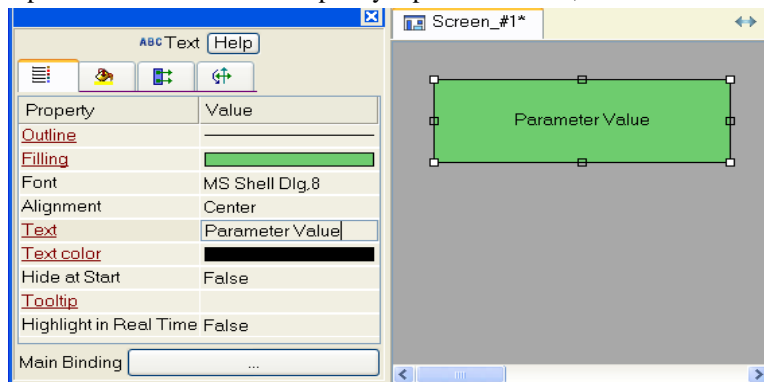



- Przejdź do edycji elementu wybierając ikonę  z paska narzędzi i klikając dwukrotnie na nim otwórz okno Properties (właściwości);

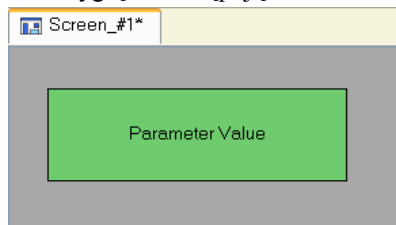
Automatyczne wyświetlanie okna Properties elementu graficznego po jego umiejscowieniu wymaga włączenia opcji **Open Properties Automatically** w sekcji **HMI/General**.



- wpisz **Parameter Value** w prawym polu linii **Text**;



- klikając na ikonę  zamknij okno Properties. Element graficzny będzie wyglądał następująco:




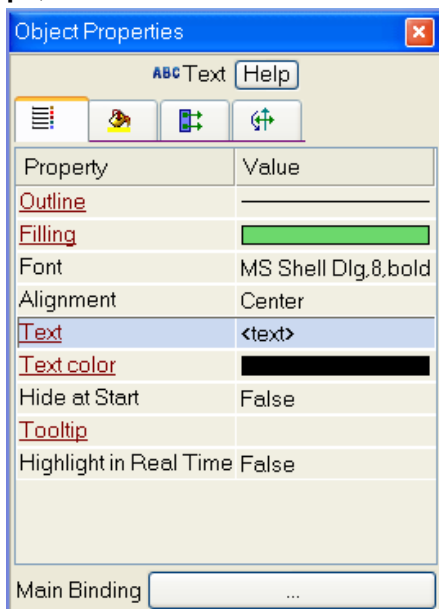
Jeżeli wpisany tekst nie pasuje do prostokąta, zaznacz go i powiększ do odpowiedniego rozmiaru przeciągając kursorem.

### ***Dodawanie dynamicznego tekstu, tworzenie argumentu ekranu podczas ustawiania dynamicznego tekstu***

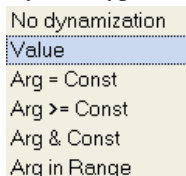
Dodaj wyświetlanie dynamicznego tekstu, aby pokazać liczbową wartość sygnału (wewnętrzny lub zewnętrzny) przy pomocy ustawienia dynamizacji elementu graficznego. Zdefiniuj przypisanie argumentu szablonu ekranu. Wy-

konaj to w następujący sposób:

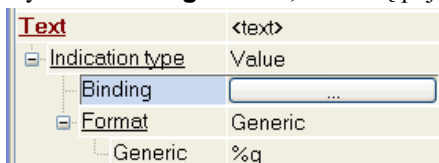
- Dodaj nowy element graficzny tekst (  ) i umieść go po prawej stronie od GE **Parameter Value**;
- kliknij dwukrotnie na linię **Text**, aby wyświetlić menu **Indication Type**;




- kliknij w prawym polu linii, aby wyświetlić listę dostępnych typów dynamizacji;
- wybierz typ **Value**;




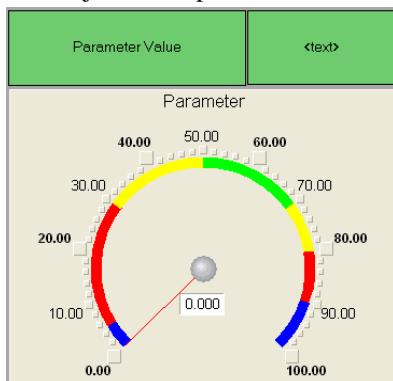
- wybierz **Binding** w menu, które się pojawi;




- wyświetli się okno **Binding Setup**. Dodaj argument ekranu klikając przycisk  na jego pasku narzędzi;



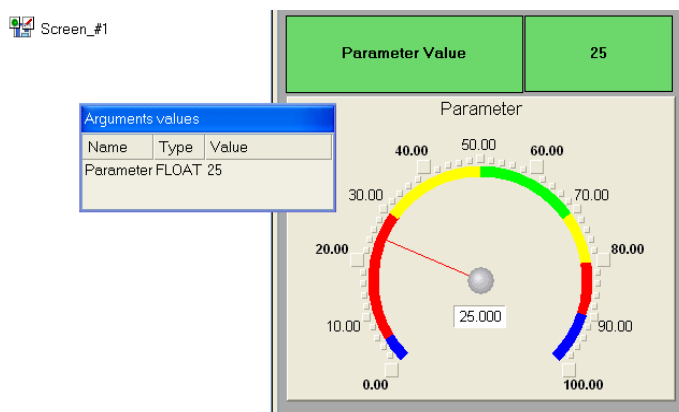
- kliknij dwukrotnie na właściwość **Caption** i wpisz słowo **Parameter** w linii **Text**;
- zamknij okno Properties GE  :



Powiązanie argumentów ekranu z GE można sprawdzić przy pomocy trybu emulacji. Uruchomienie trybu emulacji odbywa się przy pomocy ikony  na pasku narzędzi. Po kliknięciu pojawi się okno, w którym można ustawić wartość argumentu w odpowiednim polu:

Arguments values		
Name	Type	Value
Parameter	FLOAT	0

Po wprowadzeniu wartości 25 zobaczysz:



W tym przypadku oba elementy graficzne wyświetlają wprowadzoną wartość, więc powiązanie zostało wykonane prawidłowo. Kliknij jeszcze raz na przycisk

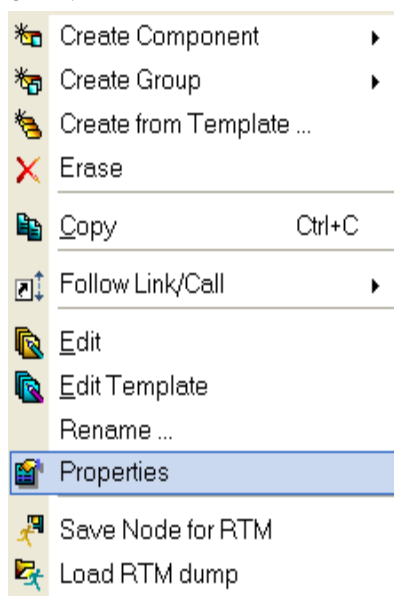



, aby wyjść z trybu emulacji.

## Autobuilding kanałów

Wykorzystaj procedurę *autobuilding*, aby przez szablon ekranu utworzyć kanał w węźle projektu:

- otwórz węzeł **RTM\_1** w powłoce **SYSTEM**;
- kliknij prawym przyciskiem myszy, aby wyświetlić menu kontekstowe i lewym przyciskiem, aby otworzyć okno Properties dla elementu **Screen#1**:



- wybierz zakładkę **Arguments**;
- wybierz argument **Parameter** klikając na nim lewym przyciskiem myszy i utwórz kanał **Float** typu **Input** o nazwie **Parameter** przy użyciu ikony :

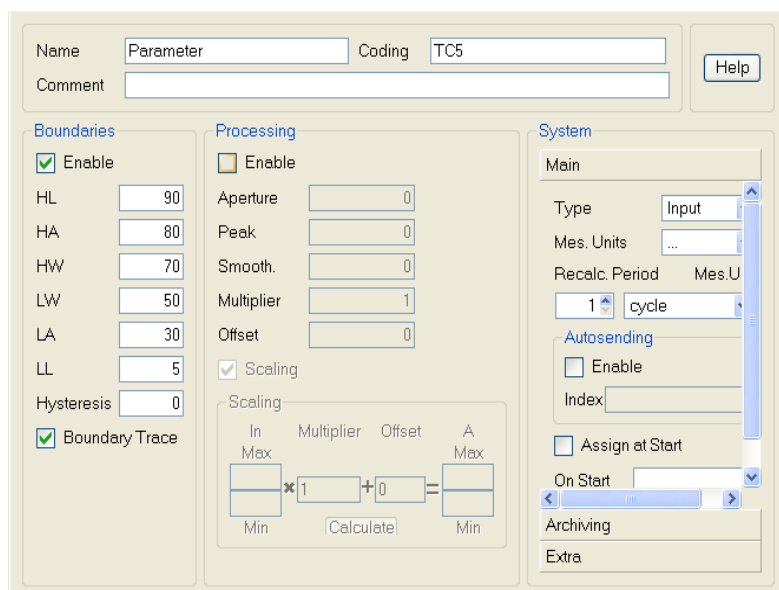
 Channels

 Screen\_#1:1

 Parameter

## Ustawianie progów

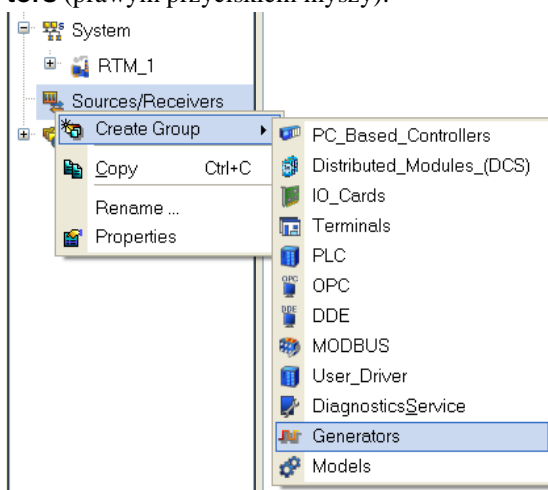
Kliknij dwukrotnie na kanał **Parameter**, aby otworzyć okno edycji atrybutów i wprowadź następujące dane do sekcji **Boundries**:



## Dodawanie generatora sinusoidy i łączenie go z kanałem

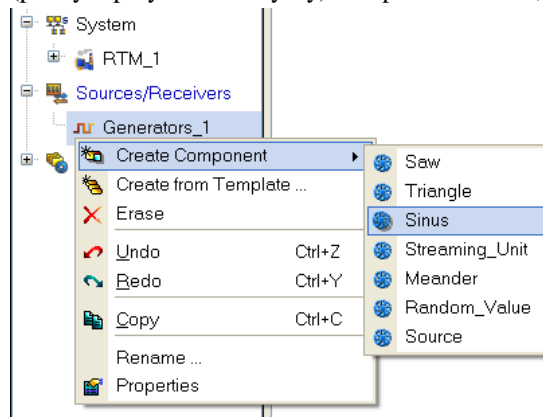
Dodaj do projektu źródło sygnału – wewnętrzny generator sinusoidy, połącz z utworzonym kanałem i sprawdź poprawność wyświetlania. Realizacja tych zadań wymaga wykonania następujących kroków:

- otwórz powłokę **Sources/Receivers** i utwórz w niej grupę **Generators** (prawy przycisk myszy):

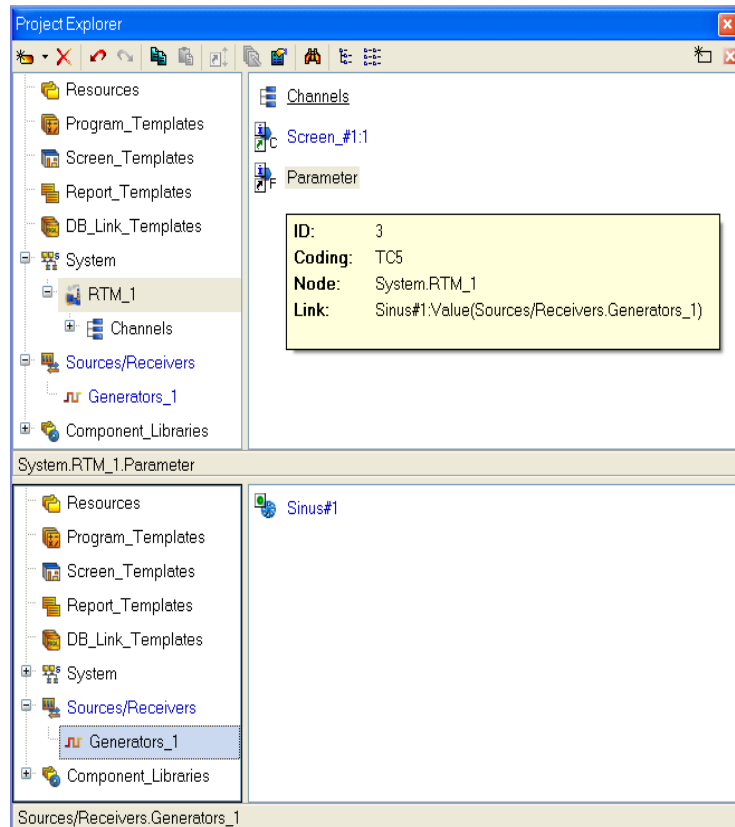


- otwórz grupę **Generators** podwójnym kliknięciem i utwórz w niej

(prawym przyciskiem myszy) komponent **Sinus**;






- chwycić utworzone źródło lewym przyciskiem myszy i przeciągnij (ciągle trzymając lewy przycisk myszy) na kanał **Parameter** znajdujący się w oknie komponentów węzła RTM\_1, powłoka **System**. Puść przycisk.

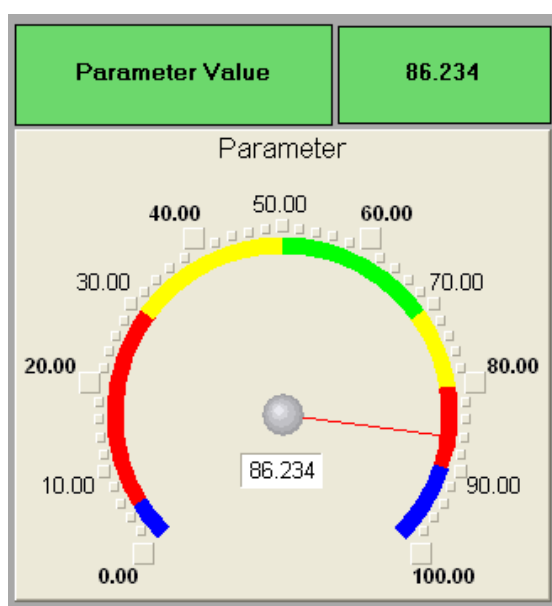




## Uruchomienie projektu

- Zapisz projekt przy pomocy ikony  nadając mu nazwę **QS\_Lesson\_1.prj**.
- Z paska narzędzi wybierz polecenie  i skompiluj projekt do uruchomienia w czasie rzeczywistym.
- Zaznacz węzeł **RTM\_1** w powłoce **System** i kliknij ikonę  z paska narzędzi, aby wejść w tryb uruchomieniowy.



Zmieniająca się wartość sygnału sinusoidalnego będzie widoczna w elemencie graficznym znajdującym się na prawo od elementu z napisem **Parameter Value**. Wskaźnik pointer gage także będzie wyświetlał tę wartość:

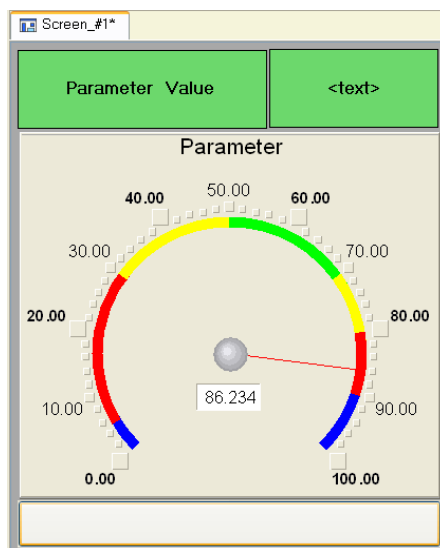




## Dodawanie funkcji sterowania

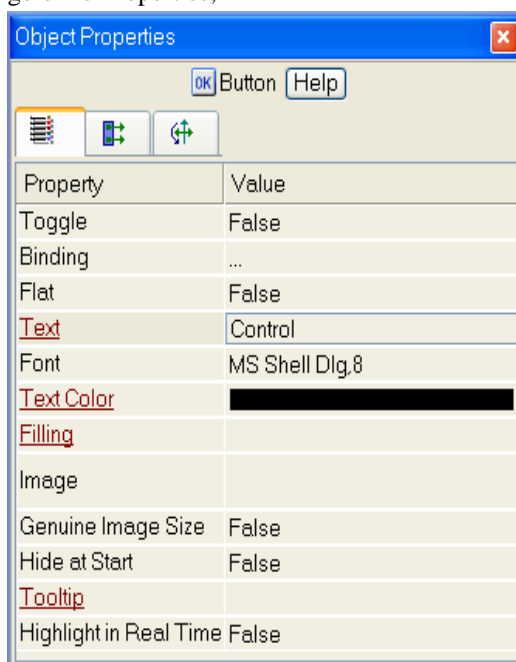
Dodaj do ekranu narzędzie, które pozwoli wprowadzać wartości liczbowe z klawiatury. Stwórz nowy szablon ekranu do odbierania tych wartości.


## Edycja ekranu graficznego

- Otwórz ekran do edycji;
- Wybierz ikonę  z paska narzędzi edytora graficznego i przy pomocy myszy umieść ją na ekranie, poniżej elementu .

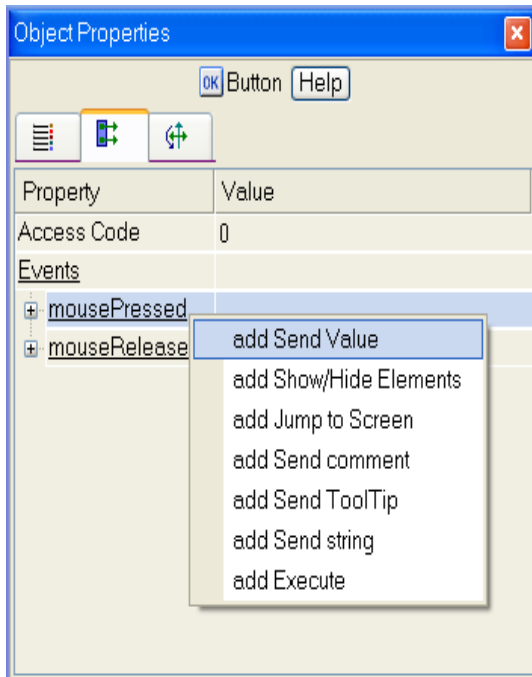


- Przejdź do trybu edycji (  ), wybierz GE **Button** (  ) i otwórz jego okno Properties;

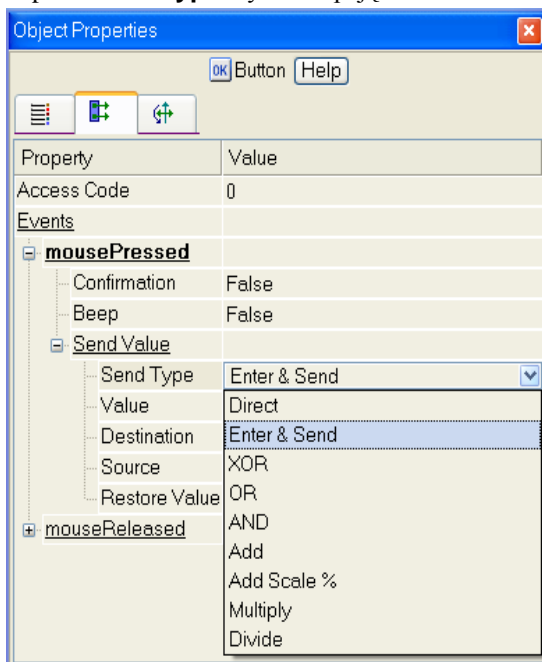


- W polu **Text** wpisz **Control**.
- Otwórz zakładkę **Events** (  ), a w niej menu **Pressed** (klikając prawym przyciskiem myszy na opcji **Pressed**);

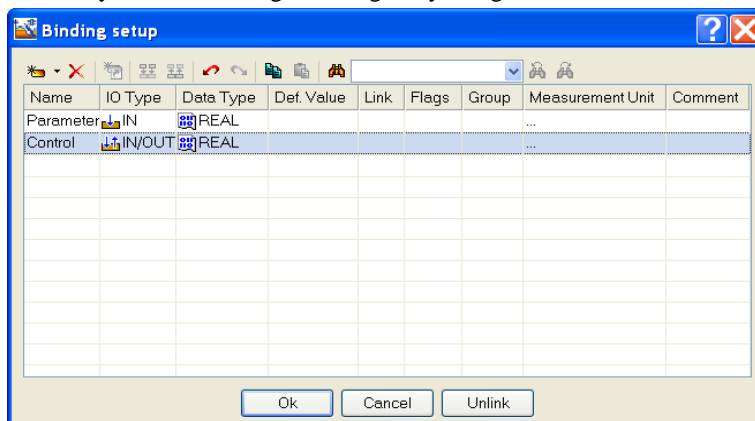
- Wybierz z listy polecenie **Add Send Value**:



- w polu **Send Type** wybierz opcję **Enter & Send**:



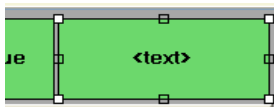
- Kliknij lewym przyciskiem na polu **Destination**, aby przejść do okna edytora argumentów;
- Dodaj argument o nazwie **Control**;
- Zmień typ argumentu na **IN/OUT** i kliknij **OK**, aby potwierdzić powiązanie atrybutu elementu graficznego z tym argumentem:



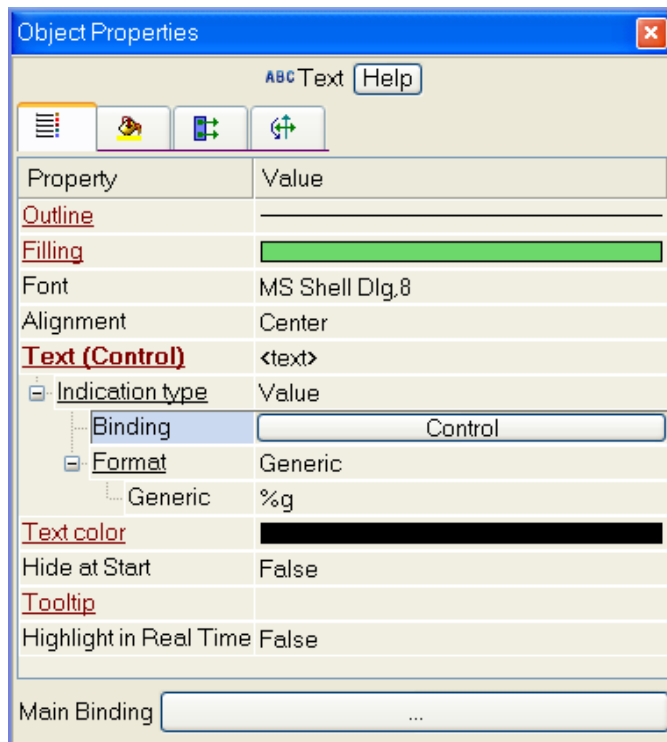
- zamknij okno Properties GE klikając na ikonę

Umieść teraz GE **Text** do wyświetlania wpisanej wartości. W tym celu skopiuj GE, który jest już na ekranie w następujący sposób:

- wybierz (lewym przyciskiem myszy) **Text** GE, wykorzystywany do wyświetlania zmiennej **Parameter**:



- skopiuj zaznaczony GE do schowka przy pomocy ikony z paska narzędzi lub kombinacji klawiszy **Ctrl+C**;
- następnie wklej element na ekran graficzny (przy użyciu ikony lub klawiszy **Ctrl+V**);
- przeciągnij kopię, aby umieścić ją po prawej stronie od GE **Button**;
- kliknij dwukrotnie na nowym elemencie **Text**, aby otworzyć jego okno Properties;
- kliknij dwukrotnie na opcję **Text** (zakładka ), aby przejść do ustawiania dynamizacji atrybutu elementu graficznego;
- kliknij lewym przyciskiem myszy w prawej części linii **Binding**, aby otworzyć edytor argumentów szablonu ekranu;
- wybierz z listy (lewym przyciskiem myszy) argument **Control** i kliknij **OK**, aby potwierdzić powiązanie atrybutu **Text** do argumentu ekranu;

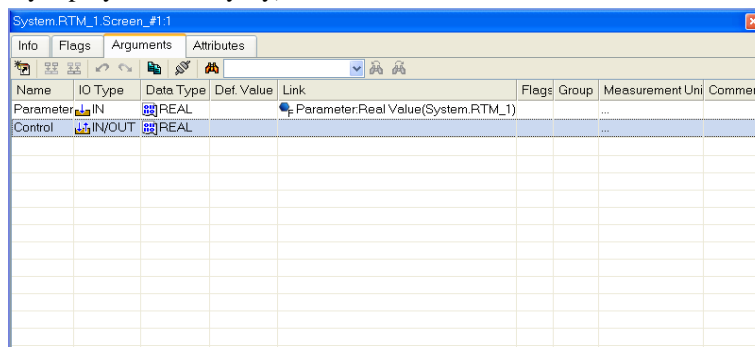


- zamknij edytor graficzny.


## Łączenie argumentów ekranu z kanałem

Dodaj nowy kanał dla argumentu **Control** i edytuj jego połączenie.

- Otwórz węzeł RTM\_1 w powłoce **System**;
- otwórz okno Properties dla elementu **Screen#1** (klikając na nim prawym przyciskiem myszy):




- wybierz zakładkę **Arguments**, zaznacz argument **Control** i utwórz


nowy kanał przy użyciu ikony .

- w wyniku tego w węźle **RTM\_1** automatycznie zostanie utworzony kanał **Control**:

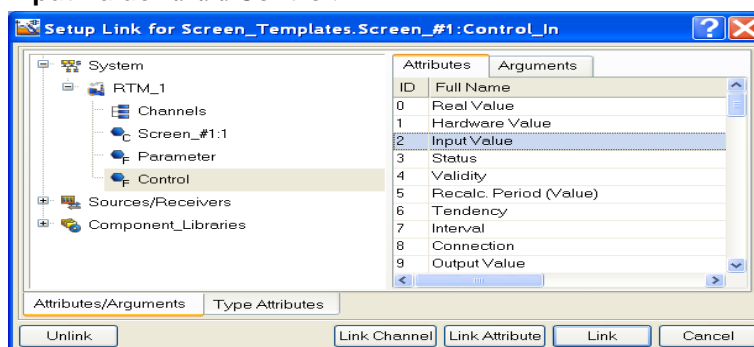
 Channels

 Screen\_#1:1

 Parameter

 Control



- Kliknij dwukrotnie w polu **Link** argumentu **Control**, aby otworzyć okno Binding Setup, wybierz w nim atrybut **Input Value** i kliknij przycisk **Link**, aby potwierdzić połączenie argumentu **Control** z atrybutem **Input Value** kanału **Control**:

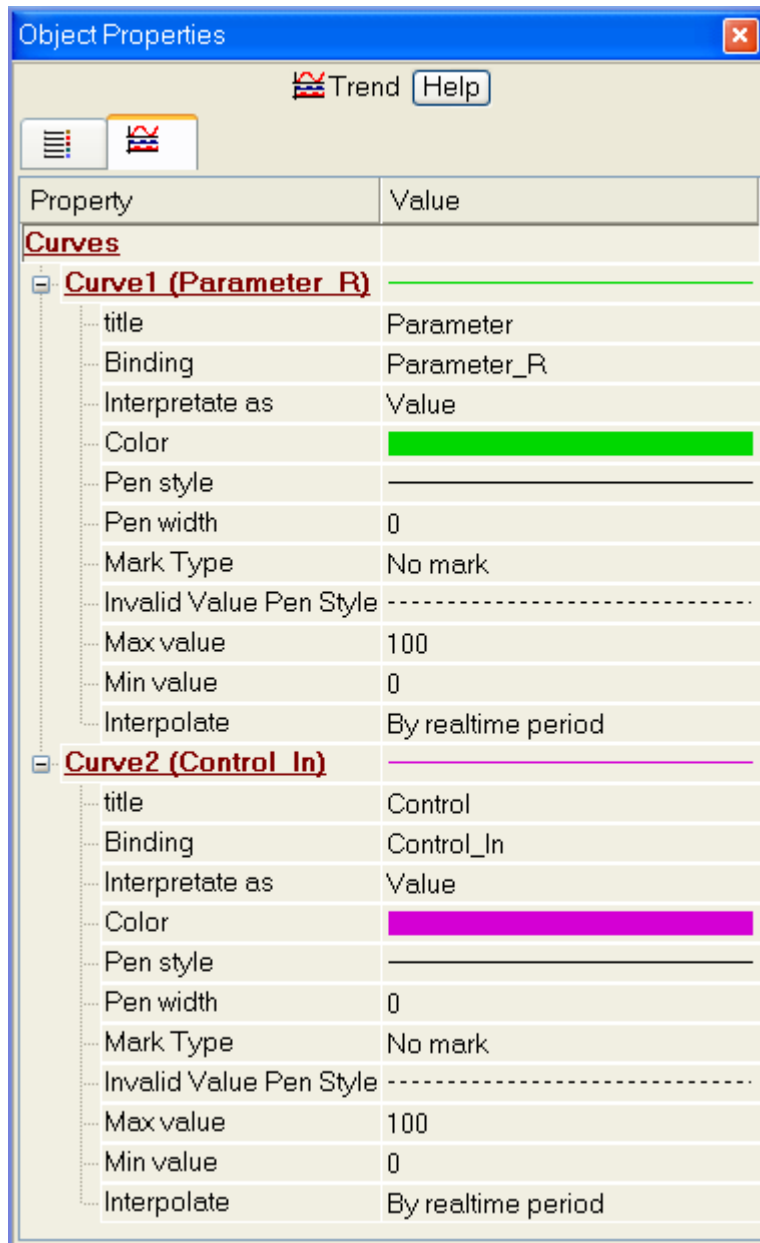


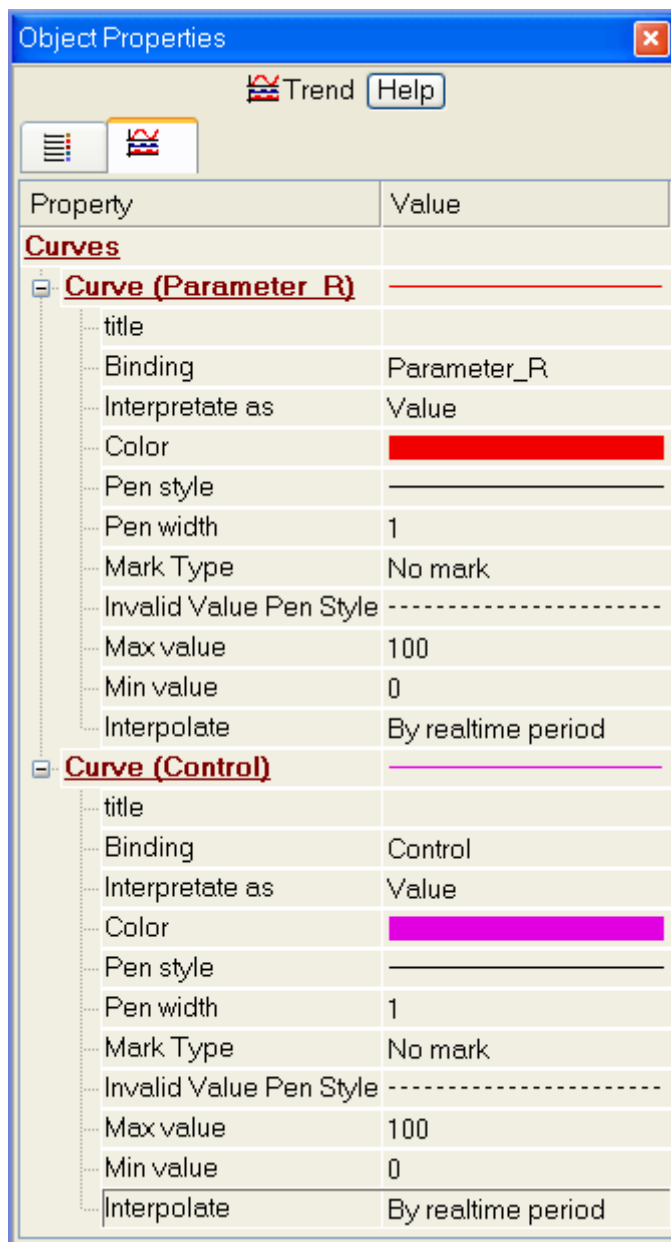
Zamknij okno Properties **Screen#1**.

## Trendy

Dodaj teraz nowy element graficzny do ekranu. Będzie to GE **Trend** umożliwiający śledzenie aktualnych wartości kanałów oraz wartości historycznych.

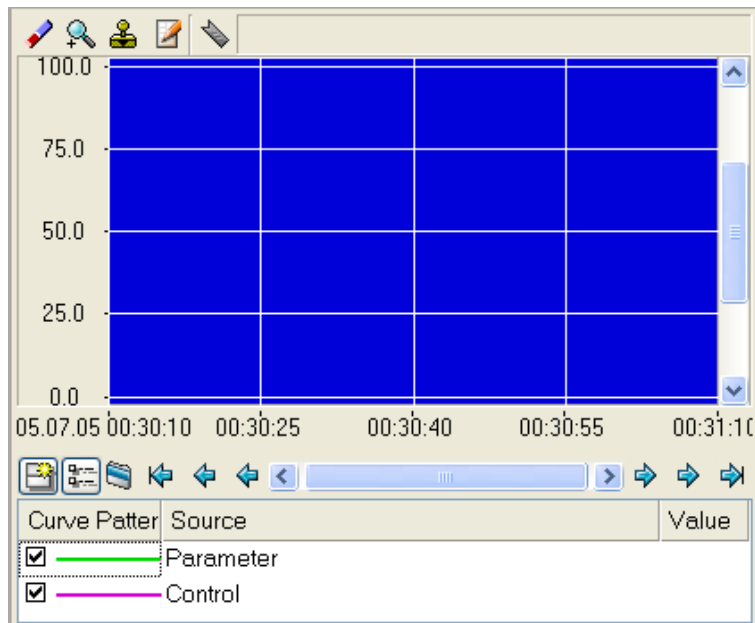
Umieść element **Trend** () w prawej części ekranu, aby wyświetlać wartości **Parameter** i **Control**. Kliknij dwukrotnie na tym elemencie, aby otworzyć jego okno Properties, wybierz zakładkę , kliknij prawym przyciskiem na linii **Curves** i dodaj dwie nowe krzywe. Ustaw teraz ich połączenia z argumentami jak również szerokość linii oraz kolor:








GE będzie wyglądał następująco:

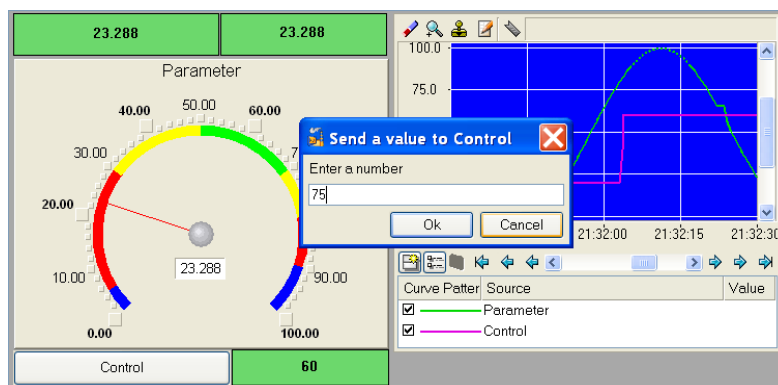




## Uruchomienie projektu

- Zapisz projekt klikając na ikonę  ;
- Kliknij ikonę  z paska narzędzi, aby skompilować projekt do uruchomienia w czasie rzeczywistym.
- Klikając na ikonę  z paska narzędzi wejdź w tryb uruchomieniowy.

Wprowadź teraz wartość **control action** przy pomocy przycisku **Control** i obserwuj zmianę na wskaźniku oraz trendzie:

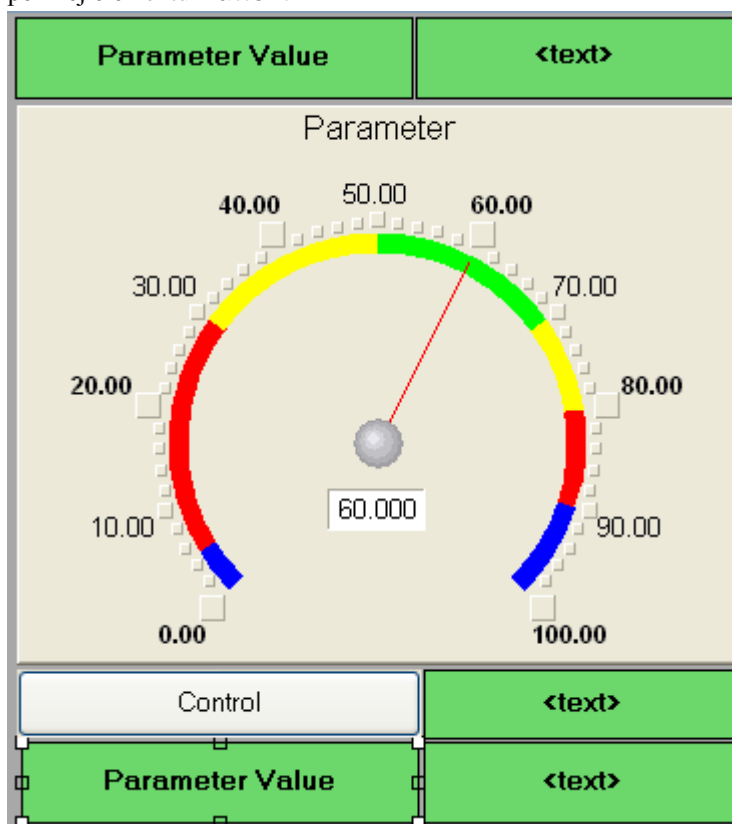


## Uprozczone przetwarzanie danych

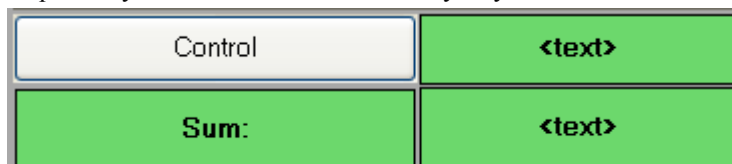
Wykorzystując teraz nowy komponent projektu, szablon programu, połącz dwa istniejące kanały ze sobą przy pomocy operacji dodawania. Sumowane będą aktualne wartości kanałów – **Parameter** i **Control**, a rezultat będzie przesyłany do nowo utworzonego argumentu ekranu **Sum** (wyświetlanie przy pomocy elementów graficznych **Text** i **Trend**) bez konieczności tworzenia dodatkowego kanału w projekcie.

### Dopracowanie ekranu graficznego

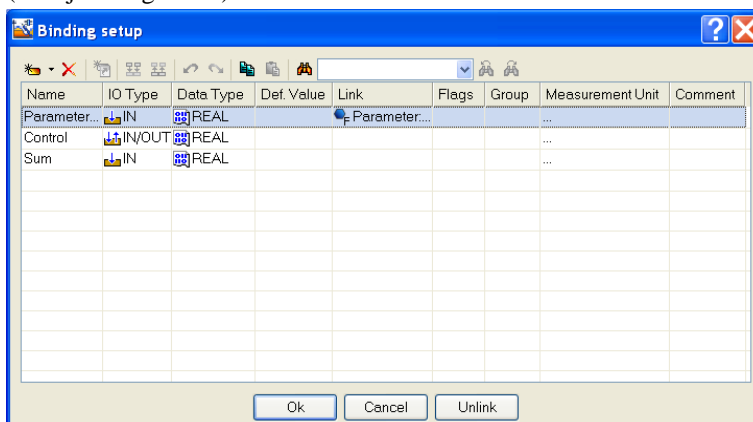
- Skopiuj elementy graficzne **Parameter Value** oraz **Text** i umieść je poniżej elementu **Button**.



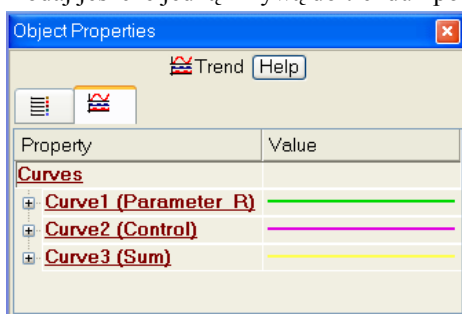
- W pierwszym elemencie zmień tekst statyczny na **Sum**.



- Połącz drugi GE do trzeciego argumentu ekranu (typ **IN**) o nazwie **Sum** (dodaj ten argument).



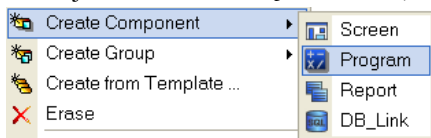
- Dodaj jeszcze jedną krzywą do trendu i połącz ją z argumentem **Sum**.



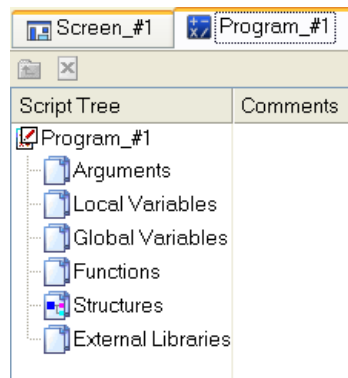
## Pisanie programu w Techno ST


Dodaj program, który sumuje dwa argumenty połączone z atrybutami **Real Value** kanałów **Parameter** i **Control** i przesyła sumę do trzeciego argumentu o nazwie **Sum**. Następnie skorzystaj z możliwości połączenia argumentów programu z argumentami ekranu, bez konieczności tworzenia dodatkowych kanałów.

- Kliknij dwukrotnie na węzeł RTM\_1, aby go otworzyć:



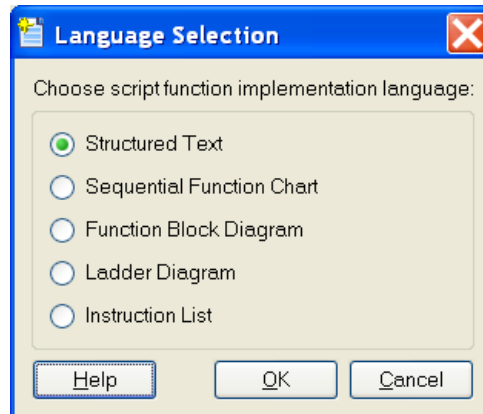
- dodaj do węzła komponent **Program**.
- kliknij dwukrotnie lewym przyciskiem myszy na komponent **Program#1**, aby przejść do jego edycji.



- kliknij lewym przyciskiem myszy na pole **Arguments** (w drzewie szablону **Program#1**), aby otworzyć edytor argumentów.
- przy użyciu ikony  dodaj trzy argumenty o nazwach **Parameter**, **Control** i **Sum**. Dwa pierwsze powinny być typu **IN**, podczas gdy trzeci – typu **OUT**.

Name	IO Type	Data Type	Def. Value
Parameter	↓ IN	REAL	
Control	↓ IN	REAL	
Sum	↑ OUT	REAL	

- kliknij lewym przyciskiem myszy na pole **Program#1** w drzewie szablону, a w oknie które się pojawi wybierz opcję **Structure Text**:



- kliknij **OK** i wpisz następujący kod w oknie edytora programu, które się pojawi:



```

PROGRAM
  VAR_INPUT Parameter : REAL; END_VAR
  VAR_INPUT Control : REAL; END_VAR
  VAR_OUTPUT Sum : REAL; END_VAR

Sum=Parameter+Control;

END_PROGRAM

```

- skompiluj program przy użyciu ikony  z paska narzędzi lub przy pomocy klawisza **F7**. Kliknij ikonę , aby uzyskać potwierdzenie o kompilacji zakończonej sukcesem (w oknie **Output**):

TRACE MODE Script Compiler v1.0

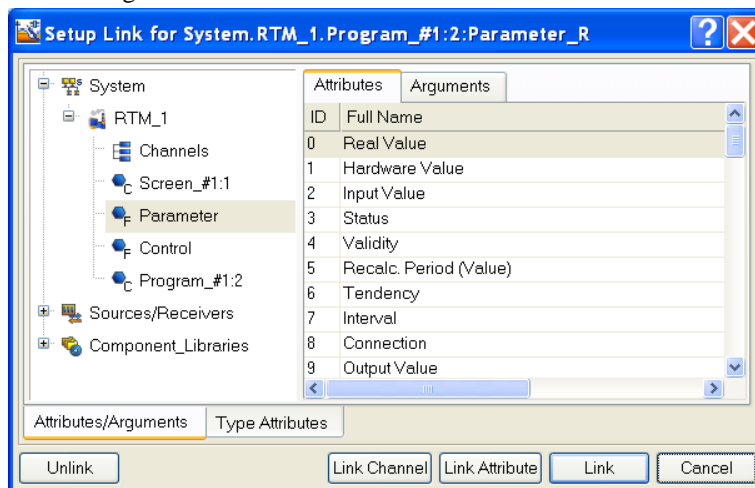
Program\_#1.tms compiled successfully

Build Debug

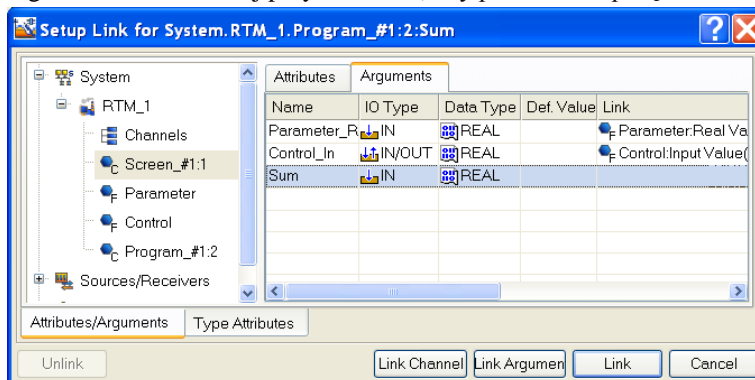
## Łączenie argumentów programu

Połącz teraz argumenty programu z atrybutami kanałów.

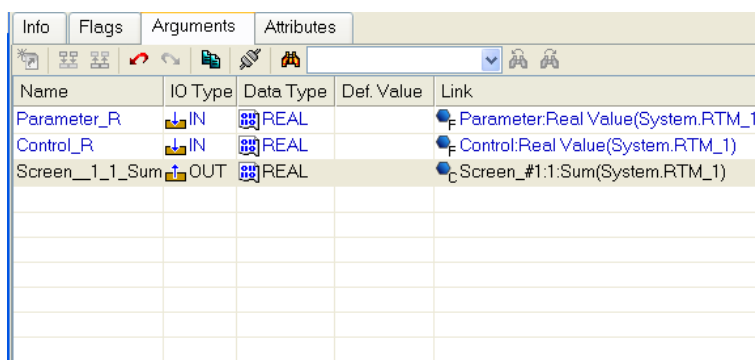
- Otwórz okno Properties komponentu **Program#1** (przy użyciu menu kontekstowego);
- wybierz zakładkę **Arguments**;
- kliknij dwukrotnie w polu **Link**, aby połączyć argumenty programu z atrybutami kanałów – argument **Parameter** z real value kanału **Parameter** i argument **Control** z real value kanału **Control**.



- kliknij dwukrotnie w polu **Link** argumentu **Sum**, aby otworzyć okno konfiguracji połączenia. Wybierz w lewym oknie kanał **Call Screen#1**, wybierz w prawym oknie zakładkę **Arguments** i zaznacz w niej argument **Sum**. Kliknij przycisk **Link**, aby potwierdzić połączenie:






Rezultat będzie następujący:

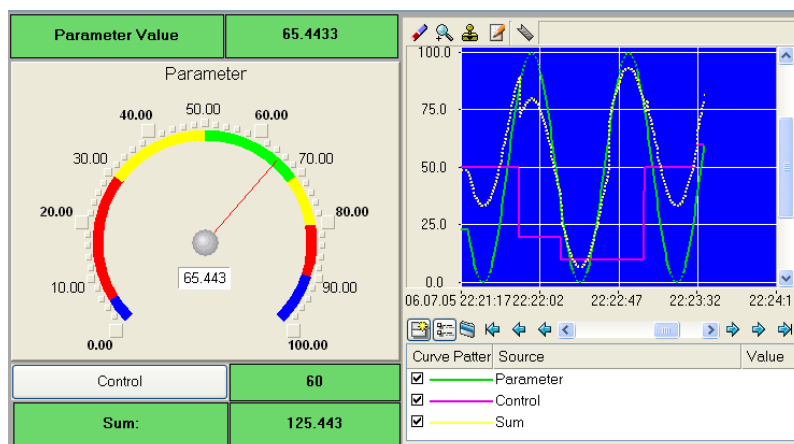


- zamknij okno Properties **Program #1**.

## Uruchomienie projektu

- Zapisz projekt klikając na ikonę  ;
- Kliknij ikonę  z paska narzędzi, aby skompilować projekt do uruchomienia w czasie rzeczywistym.
- Klikając na ikonę  z paska narzędzi wejdź w tryb uruchomieniowy.

Wpisz wartość **control action** przy użyciu przycisku **Control** i zaobserwuj odpowiednią zmianę aktualnej wartości kanału **Control** i zmiennej **Sum** z programu:



## Dynamic Data Exchange (DDE) na przykładzie MS Excel

### RTM jako serwer DDE

Stwórz teraz zapytanie generowane przez MS Excel (wersja 2007) o aktualne wartości kanałów węzła w następujący sposób:

- uruchom MS Excel;
- wpisz zapytania o dane w dwóch pierwszych komórkach pierwszej kolumny:

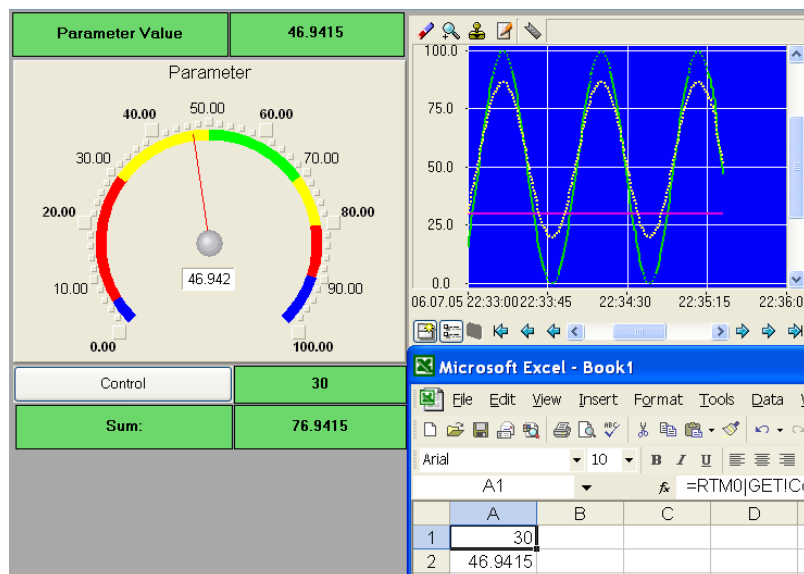
**=RTM0|GET!Parameter**

**=RTM0|GET!Control**

gdzie **0** jest indywidualnym numerem projektu;

- uruchom węzeł **RTM\_1**;
- otwórz menu **Dane** w MS Excel, wybierz polecenie **Edytuj łącza**, zaznacz wszystkie trzy parametry i kliknij przycisk **Aktualizuj wartości**; następnie kliknij **OK** aby zamknąć okno;

upewnij się, że wartości w arkuszu Excel są aktualizowane zgodnie z odpowiednimi wartościami kanału (wartość kanału **Parameter** jest ciągle aktualizowana, natomiast wartość kanału **Control** jest aktualizowana po wprowadzeniu nowej wartości przy pomocy GE **Button**):



### RTM jako klient DDE

W przypadku, gdy potrzebne jest odbieranie danych przy użyciu protokołu DDE z zewnętrznych aplikacji, TRACE MODE 6 RTM musi być klientem DDE. Jeżeli potrzebujesz np. wprowadzać dane do nowego kanału *From\_Table* należącego do węzła **RTM\_1** (do jego atrybutu **Input Value**) z komórki **R3C3** arkusza MS Excel, dodaj nową grupę **DDE** w powłoce **Sources/Receivers**, a w niej element **DDE#1** i edytuj go w następujący sposób:

The screenshot shows the configuration dialog for a DDE source. The 'Main' section includes fields for Name (Excel.Sheet1.R3C3), Coding (TW0), and Comment. The 'Parameters' section includes Mode (ADVISE), A (0), B (0), Direction (Input), and Format (Analog). An 'Extra' field is also present at the bottom.

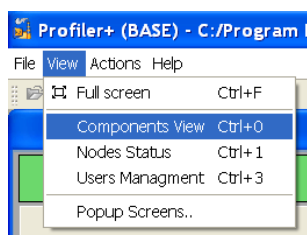


w atrybucie **Name**:

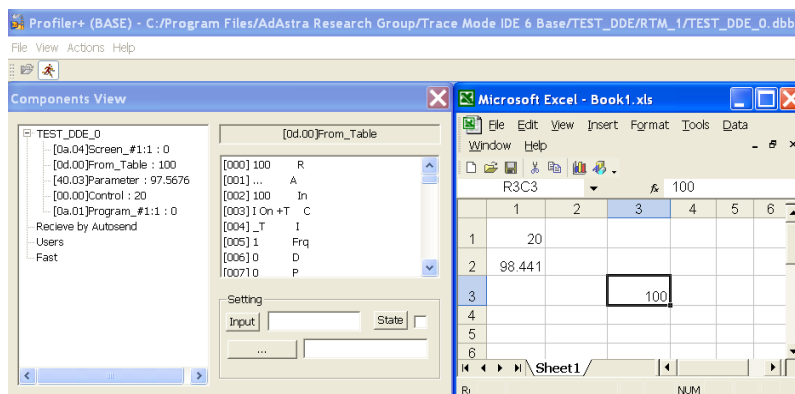
- Excel – nazwa aplikacji;
- Arkusz1 –nazwa arkusza Excel;
- W3K3 – adres komórki.

**ADVISE** – tryb, wysłanie do klienta wartości po każdej aktualizacji.

następnie dodaj kanał **Float** (typ **Input**) o nazwie *From\_Table* i połącz źródło **Excel.Arkusz1.W3K3** z tym kanałem (przy użyciu metody **przeciągnij-upuść**). Zapisz projekt i skompiluj go do pracy w czasie rzeczywistym. Uruchom Excel, a następnie uruchom węzeł **RTM\_1**. Po wprowadzeniu dowolnych wartości do komórki **W3K3** możesz obserwować je w atrybutach kanału *From\_Table*, przy użyciu okna Components View (możesz otworzyć je przy użyciu głównego menu):



W ten sposób otrzymasz następujący rezultat:



## Podłączanie rozproszonych modułów I/O

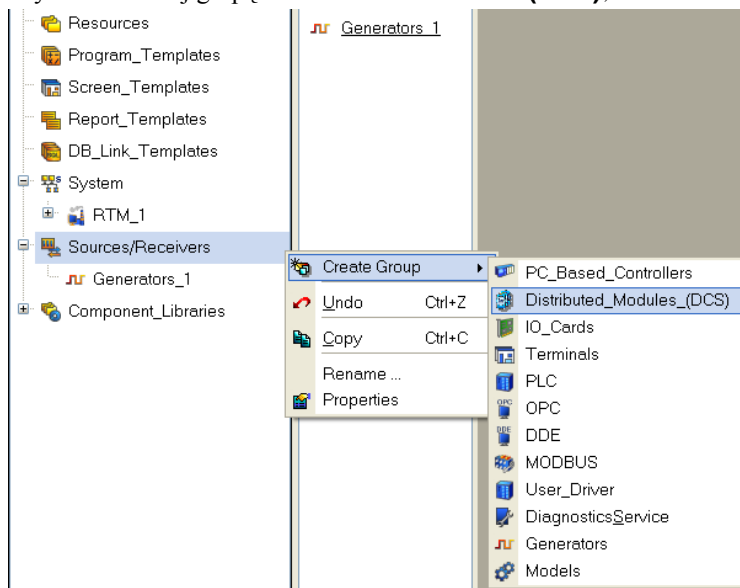
Dodaj teraz do projektu rozproszony moduł wejściowy **I-7011** z dołączonym czujnikiem. Jest nim jest termopara typu TXA (chromel-alumel) z charakterystykami kalibracyjnymi **Typu K**. Po pierwsze należy dopasować moduł do określonej charakterystyki kalibracyjnej przy użyciu narzędzi dostarczonych wraz z modulem. Ustaw **'inżynierski'** format danych, przypisz numer w sieci **RS-485**, który jest równy **1** i ustaw format wymiany danych na **9600,n,8,1** bez sumy kontrolnej. Następnie podłącz moduł do portu **COM1** przez automa-

tyczny konwerter interfejsów **I-7520** oraz zasilanie do obu modułów.

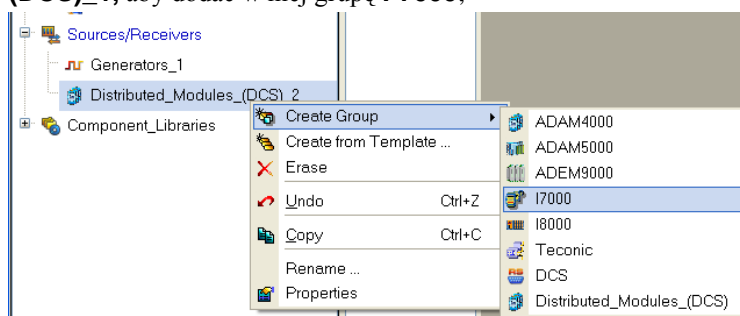
### Dodawanie elementów źródłowych do wprowadzania danych z I-7011

Dodaj elementy źródłowe połączone z wejściami/wyjściami sprzętu i ustaw ich atrybuty.

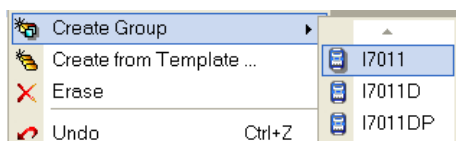
- Kliknij prawym przyciskiem myszy na powłokę **Sources/Receivers**, aby dodać w niej grupę **Distributed Modules (DCS)**;



- kliknij prawym przyciskiem myszy na grupę **Distributed Modules (DCS)\_1**, aby dodać w niej grupę **I-7000**;



- kliknij prawym przyciskiem myszy na grupę **I-7000\_1**, aby dodać w niej grupę **I-7000#1**.



- otwórz dodaną grupę **I-7000#1** klikając ją dwukrotnie i przejdź do utworzonych elementów, opisujących podłączony do stacji roboczej moduł I-7011:

AI#1

CJC#2

- kliknij dwukrotnie na element **AI#1**, aby przejść do edycji atrybutów;

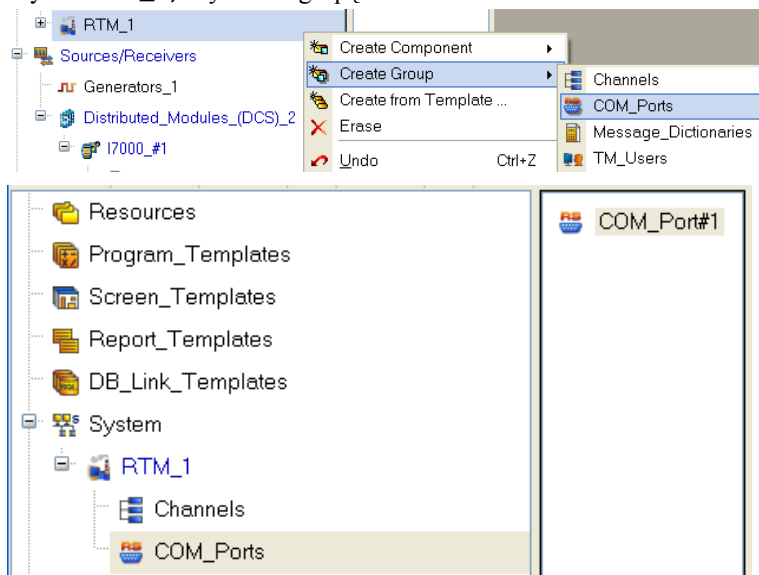
- w większości głównych atrybutów pozostaw wartości domyślne: **Port Number** 0 odpowiada portowi COM1, **Address** to 1, atrybuty **Channel** i **Slot** nie są ustawiane dla modułu, brak **Checksum**, **Signal Type** to Input. Zmień **Signal Type** przy użyciu listy pop-up zgodnie z typem czujnika, wpisz opis w polu **Comment**.

## Dodanie oraz konfiguracja portu COM

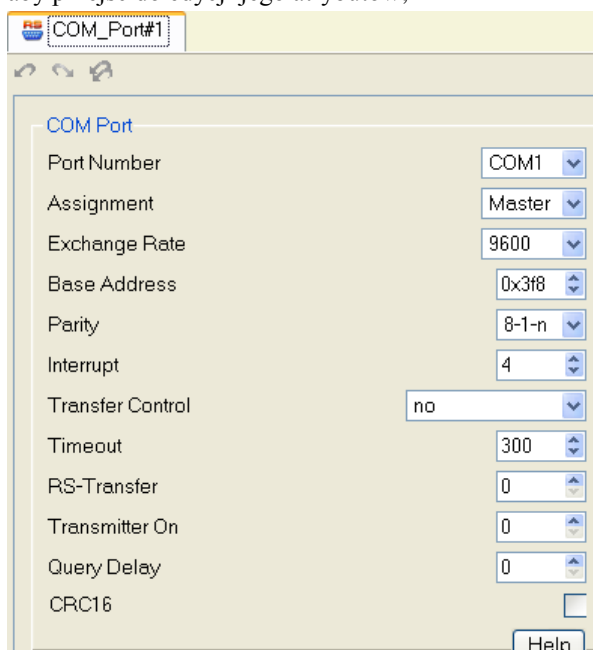
Z racji tego, że moduł I-7011 jest podłączony do węzła przez port szeregowy,


należy dodać i skonfigurować port szeregowy dla węzła RTM\_1 w następujący sposób:

- otwórz powłokę **System/RTM\_1** oraz kliknij prawym przyciskiem myszy na **RTM\_1**, aby dodać grupę **Com-Ports**:




- kliknij dwukrotnie lewym przyciskiem myszy na element **Com-Port#1**, aby przejść do edycji jego atrybutów;

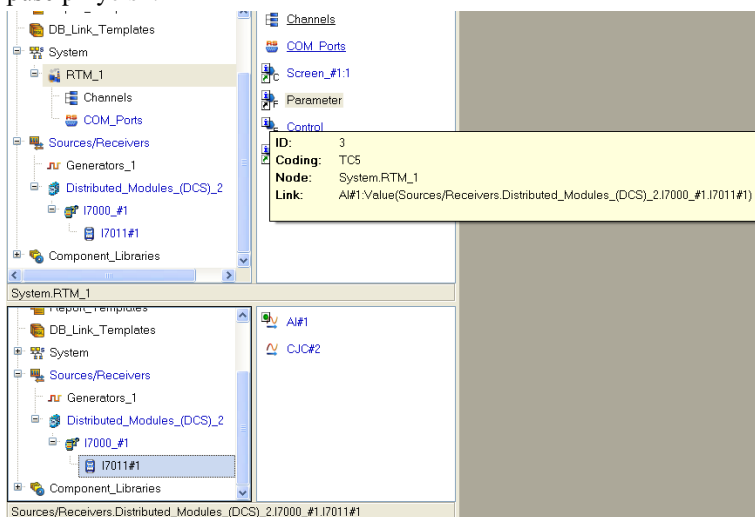


- w tym szczególnym przypadku pozostaw wszystkie wartości domyślne;
- zamknij okno klikając na przycisk .

## Łączenie kanału ze źródłem danych


Zmień połączenie kanału **Parameter**: z generatora sinusoidy na rzeczywisty, analogowy sygnał wejściowy.

- Przy pomocy przycisku () skorzystaj z opcji **Duplicate**, aby otworzyć jeszcze jedno okno Project Explorer. Otwórz powłokę **System/RTM\_1**, wybierz w niej (lewym przyciskiem myszy) analogowy sygnał wejściowy modułu **AI#1** i przeciągnij na kanał **Parameter**, który jest widoczny w dodatkowym oknie Project Explorer;
- puść przycisk.

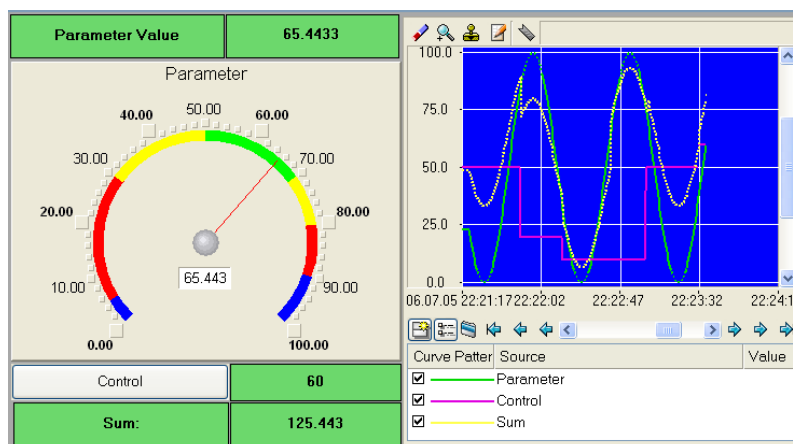


Zapisz projekt i przygotuj go do uruchomienia.

## Uruchomienie projektu

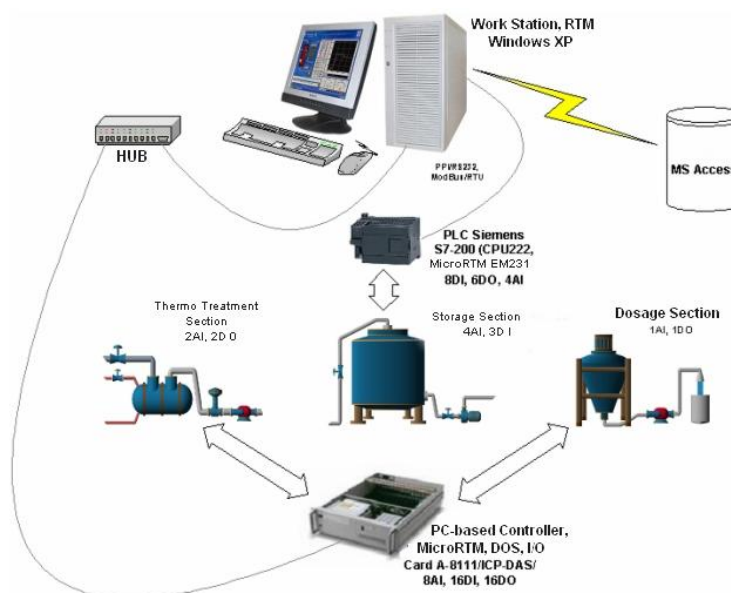
Wybierz węzeł **RTM\_1** (klikając na nim lewym przyciskiem myszy) w oknie Project Explorer i kliknij ikonę  na pasku narzędzi, aby wejść w tryb uruchomieniowy.

W czasie gdy czujnik podłączony do modułu jest podgrzewany lub chłodzony możesz zaobserwować zmianę sygnału na trendzie. Jeżeli z jakiegoś powodu nie ma czujnika, możesz połączyć element **CJC#2** z kanałem **Parameter** i obserwować fluktuacje temperatury termozłącza, które są mierzone przez moduł.



## Human Machine Interface: Monitoring i sterowanie

Analizowany proces produkcyjny składa się z trzech sekcji: *Thermotreatment (obróbki cieplnej)*, *Storage (magazynowania)* i *Dosage (dozowania)*. Celem jest zbudowanie systemu sterowania i monitoringu.



Sterowanie w sekcji obróbki cieplnej odbywa się przy użyciu kontrolera PAC (wykorzystywane są dwa analogowe sygnały wejściowe - 2AI oraz dwa binarne sygnały wyjściowe - 2DO). Celem jest utrzymanie stałej temperatury w urządzeniu, które podgrzewa surowiec (funkcja sterowania). Regulacja tempe-

ratury odbywa się przez zmianę ilości ciepła dostarczanego do urządzenia. Algorytm sterowania jest typu PDD; mechanizm aktuatora jest sterowany przy pomocy PWM. Sygnały analogowe z czujników parametrów technologicznych podawane są przez konwertery normalizujące do sterownika w postaci komputera PC, gdzie przy użyciu 12-bitowego przetwornika analogowo-cyfrowego (ADC) zamieniane są na liczby (0 – 4095). Załóżmy, że dla czujnika temperatury powyższy zakres liczbowy odpowiada zakresowi temperatur (0 – 100) C° oraz poborowi ciepła – (0 – 10) m<sup>3</sup>/h. Stacja robocza operatora monitoruje parametry technologiczne dzięki czujnikom podłączonym do kontrolera PAC (funkcja monitorowania) i zadaje parametry pętli PID (funkcja sterowania).

Sterowanie w sekcji magazynowania realizowane jest na tradycyjnym sterowniku PLC (z trzema wejściami binarnymi - 3DI oraz czterema wejściami analogowymi - 4AI). Stacja robocza jest wykorzystywana tylko do monitorowania. Sterownik PLC SIEMENS S7-200 posiada procesor centralny CPU222 (8DI, 6DO) oraz moduł EM231 (4AI). Program utrzymujący zadany poziom w sekcji oraz umożliwiający komunikację ze stacją roboczą przy użyciu dowolnie konfigurowalnego interfejsu komunikacyjnego PPI (przez standardowy protokół wymiany danych Modbus RTU), wbudowanego w moduł procesora, został zrealizowany przy użyciu oprogramowania STEP7. Młodsze bity bajtu z komórki o adresie 0x0 zawierają informacje z czujnika umieszczonego przy drzwiach do sekcji magazynowania (0 – drzwi zamknięte, 1 – otwarte), przy wentylatorze (0 – nie pracuje, 1 – pracuje) oraz czujnika przeciwpożarowego (0 – brak zadymienia, 1 – zadymienie). Dwubajtowe komórki o adresach 0x1, 0x3, 0x5 i 0x7 zawierają dane odpowiadające takim parametrom jak poziom wypełnienia, temperatura magazynowania, ciśnienie powietrza oraz wilgotność. ADC zaimplementowany w EM231 posiada 12-bitową rozdzielczość, co daje zakres liczbowy (0 – 4095), podczas gdy odpowiednie wartości sygnałów mają zakres (0 – 5) m, (0 – 100) C°, (0 – 5) atm i (0 – 100) %. Szeregowy port komunikacyjny PPI CPU222 posiada następujące parametry: 19200,n,8,1 (szybkość transmisji, bity danych, parzystość, bity stopu, kontrola przepływu)

Sekcja dozowania jest monitorowana i sterowana przy użyciu kontrolera PAC (wykorzystywany jest jeden analogowy sygnał wejściowy - 1AI, oraz jeden binarny sygnał wyjściowy - 1 DO). Dozowanie produktu finalnego zależne jest wartości parametrów zadanych w stacji roboczej. Sprzęt sterujący jest przedstawiony w postaci pomp dozujących. Włączanie pompy odbywa się przez wysłanie z kontrolera wartości logicznej «1» do modułu, który uruchamia aktuator pompy; logiczne «0» powoduje wyłączenie pompy.

Kontroler PAC jest połączony ze stacją roboczą przez hub sieciowy (protokół TCP/IP). Kontrolerem może być dowolny komputer PC. Działa on pod systemem operacyjnym DOS OS i wykorzystuje kartę akwizycji danych I/O A-8111 – ISA z 8 analogowymi i 16 binarnymi wejściami (8AI, 16DI) oraz 16 wyjściami (16DO /ICP DAS).


W czasie pracy systemu parametry magazynowania (poziom, temperatura, ciśnienie, wilgotność) powinny być co 5 minut zapisywane w bazie danych MS ACCESS.

System powinien umożliwiać jednoczesną pracę dwóm użytkownikom – operatorowi i projektantowi. W przeciwieństwie do projektanta, operator nie może wprowadzać żadnych zmian w strukturze systemu.

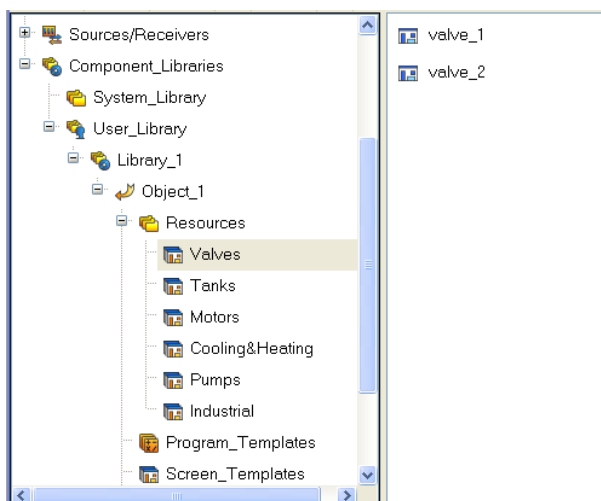
## Dodawanie ekranów HMI

Przedstawione teraz zostanie tworzenie systemu sterowania w sposób ‘zaczynając od szablonów’, tj. tworzenie podstaw projektu (kanały) bazując na argumentach szablonów ekranów i programów, główne podejście będzie uzupełnione procedurą autobuilding i łączeniem kanałów w węzłach projektu.

Wykorzystaj bibliotekę komponentów użytkownika. Skopiuj w tym celu plik **tmdevenv.tmul** z podkatalogu **%TRACE MODE%\Lib** do katalogu **%TRACE MODE%**.

Otwórz Integrated Development Environment i kliknij ikonę , aby utworzyć nowy projekt. Wybierz poziom zaawansowania **Standard**.

Otwórz powłokę **Component Libraries** i jej sekcję **User Library**. Otwórz następnie **Library\_1**. Obiekt biblioteki **Object\_1** zawiera w powłoce **Resources** zestaw obiektów graficznych, które zostaną wykorzystane w tym projekcie (te obiekty to zawory, zbiorniki, silnik, itp.).



---

Liczba dostępnych w bibliotece obiektów graficznych zależy od używanej wersji Integrated Development Environment (wersja base lub professional).

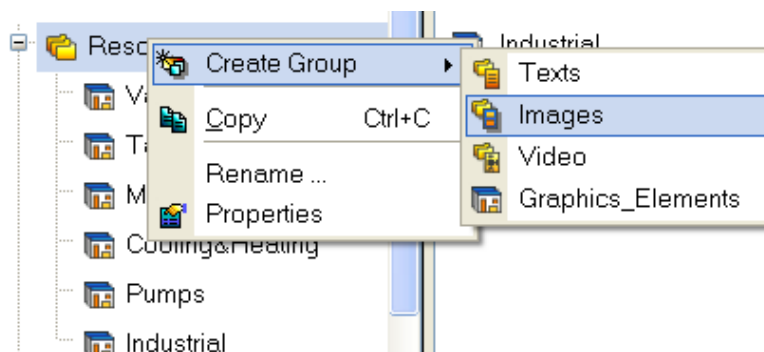
---

Przeciągnij odpowiednie grupy do powłoki **Resources** obecnego projektu przy użyciu metody **przeciągnij-upuść**.

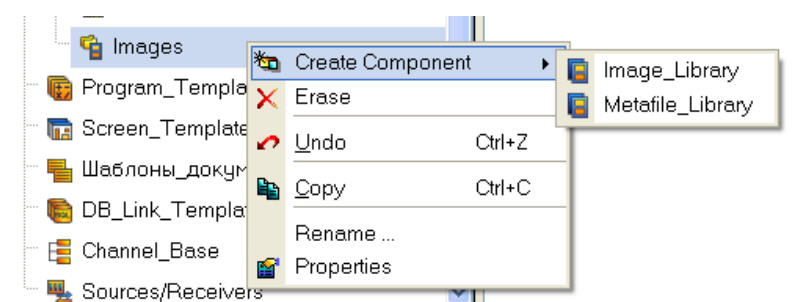
W tej samej powłoce **Resources** dodaj grupę **Images**, aby dodać do niej tek-




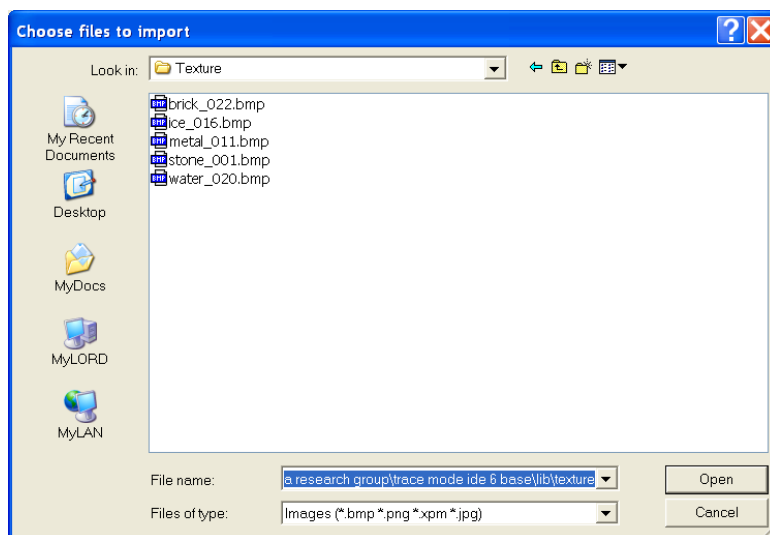
stury, które będą wykorzystywane w projekcie ekranów graficznych.



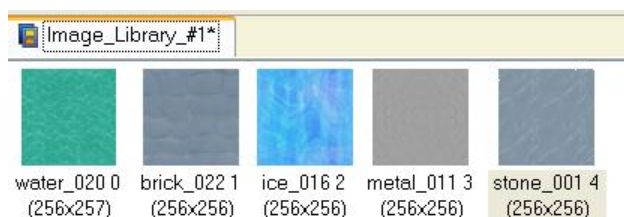
Dodaj do grupy **Images** nowy element – **Images\_Library#1**.



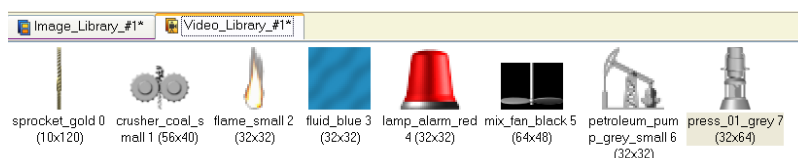
Edytuj dodaną bibliotekę klikając ją dwukrotnie. Wykorzystaj ikonę  z paska narzędzi, aby dodać do niej elementy. W oknie, które się wyświetli wybierz **...\LibTexture**, zaznacz wszystkie pliki i naciśnij przycisk **Open**.




Zawartość **Images\_Library#1** będzie następująca:



W ten sam sposób dodaj grupę **Animation** w powłoce **Resources**, a w niej **Videos\_Library #1**. Dodaj do niej zawartość **...\Lib\Animation**.



Ze wszystkich animacji obecnych w bibliotece wykorzystamy tylko **fluid\_blue**, pozostałe można skasować przy użyciu ikony  z paska narzędzi (zaznacz animację i kliknij przycisk)


---

Liczba dostępnych w bibliotece tekstur i animacji zależy od używanej wersji Integrated Development Environment (wersja base lub professional)..

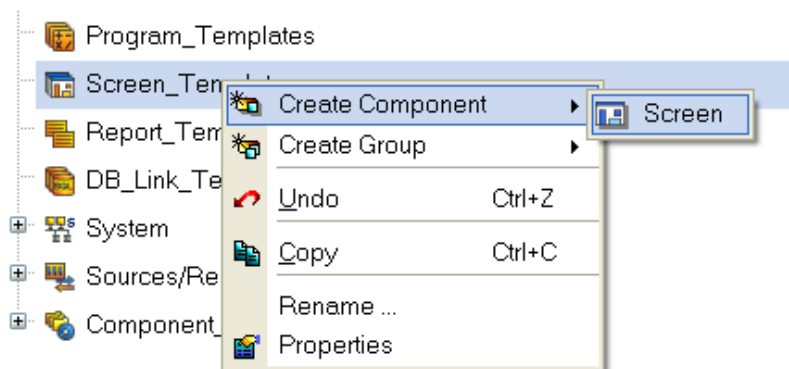
---

Obecnie dowolne pliki **avi** lub **mng** mogą zostać wykorzystane jako animacje.

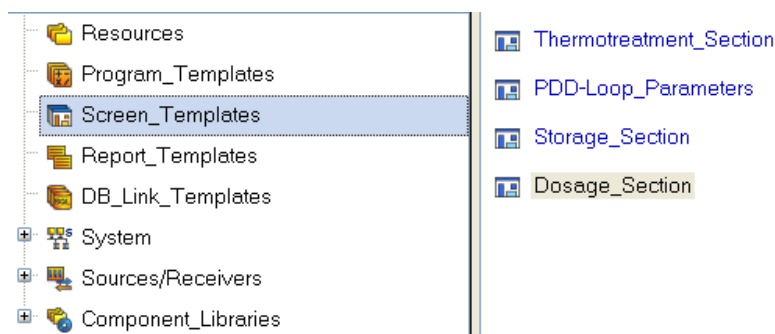
---


Zapisz projekt klikając lewym przyciskiem myszy na ikonę  i nadaj mu nazwę **QS\_Lesson\_2.prj**.

Przejdź do powłoki **Screen templates** i dodaj w niej element **Screen#1**.




Dodany ekran będzie wyświetlał parametry technologiczne sekcji obróbki cieplnej oraz będzie wykorzystany do sterowania. Zmień jego nazwę na **Thermotreatment\_Section**. Dodaj także ekrany do zadawania parametrów pętli **PDD-loop-Parameters**, **Storage\_Section** oraz **Dosage\_Section** i odpowiednio je nazwij:

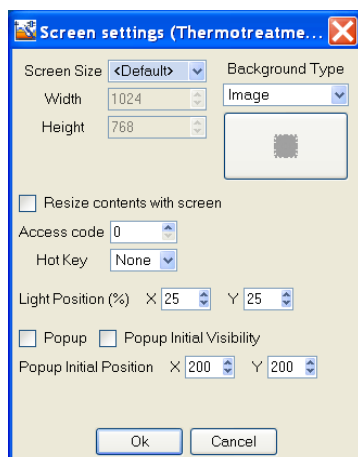


Otwórz okno Properties elementu **Thermotreatment\_Section** (kliknij prawym przyciskiem myszy i wybierz opcję **Properties**) i zaznacz zakładkę **Arguments**. Możesz dodać w niej niezbędne argumenty (przy użyciu ikony ) i nadać im nazwy, typ, typ danych, wartość domyślną, połączenia, flagi, itp.

Name	IO Type	Data Type	Def. Value	Link	Flags
Heat_Carrier_Consumption	IN	REAL			
Operation_Temperature	IN	REAL			
Temperature_Setting	OUT	REAL			
Date_Time	IN	DATE_AND_TIME			NP


Argumenty, których wartości będą wyświetlane na ekranie są typu **IN**, podczas gdy te zadawane z klawiatury, wyświetlane na ekranie i wysyłane do kontrolera PAC są typu **OUT**. W trakcie procedury autobuilding zostaną automatycznie wykonane połączenia argumentów z atrybutami Real oraz **Input Value** odpowiednich kanałów.

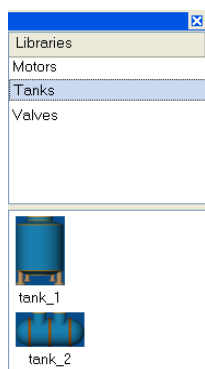
Kliknij ikonę , aby zamknąć okno Properties. Jako tło ekranu ustaw teksturę **metal\_011** - wybierz opcję **Tools** z głównego menu, a w niej **Screen Settings**. W oknie, które się pojawi ustaw **Image** jako typ tła oraz **metal\_011** jako teksturę.





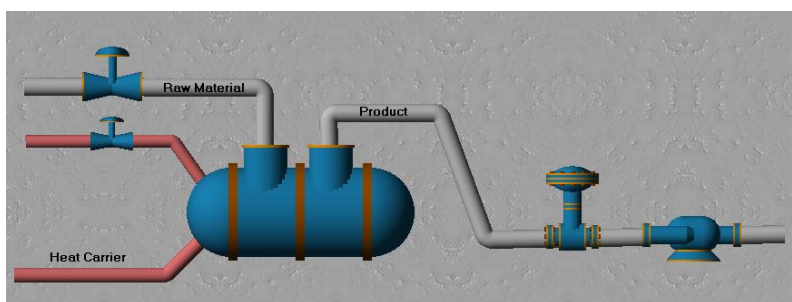
Po kliknięciu przycisku **OK** zmieni się tło ekranu graficznego.

Wykorzystując obiekty graficzne (GO), które są zapisane w bibliotece zasobów

i mogą zostać pobrane przy pomocy ikony  z paska narzędzi,




jak również elementy graficzne (GE) rura 3-D () oraz tekst () stwórz statyczną część ekranu. Przykładowy ekran jest zaprezentowany poniżej.

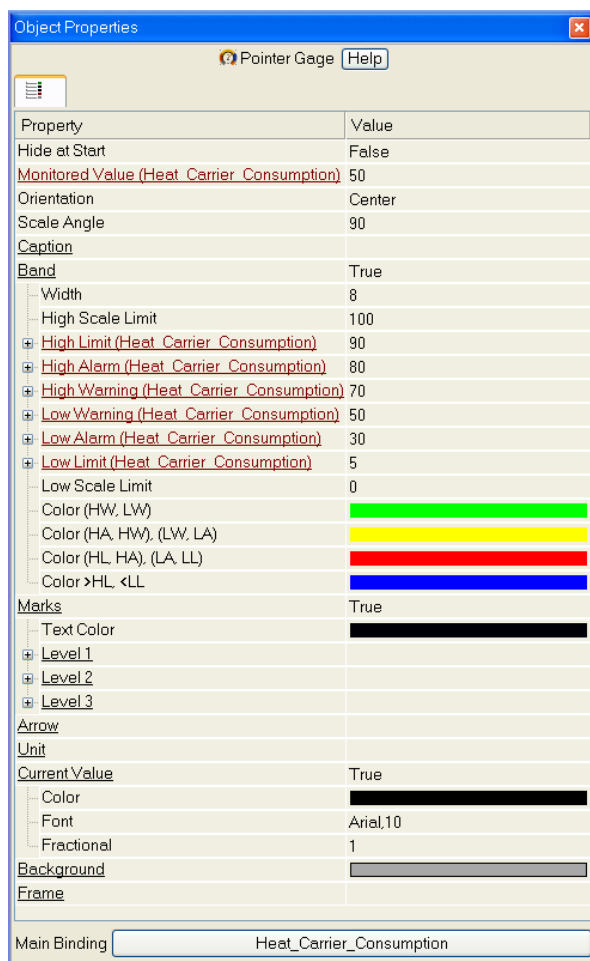


Obiekty graficzne są umieszczane przy pomocy metody **przeciągnij-upuść** i

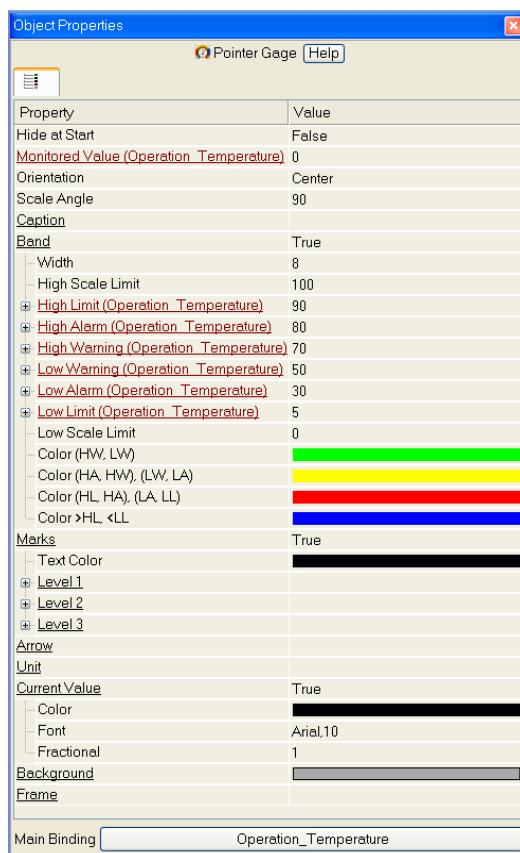
mogą zostać przeskalowane. Aby zmienić rozmiar obiektu zaznacz go lewym przyciskiem myszy, umieść kursor na jednym z punktów i przeciągnij w odpowiednim kierunku:




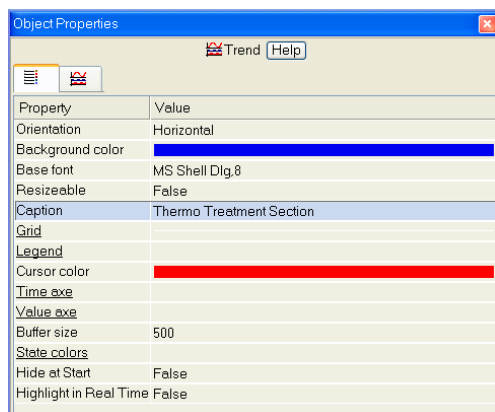
Wyświetl zużycie ciepła i temperaturę pracy przy użyciu wskaźników Pointer gage (  ). Umieść je na ekranie i kliknij dwukrotnie na lewym z nich, aby otworzyć jego okno Properties:



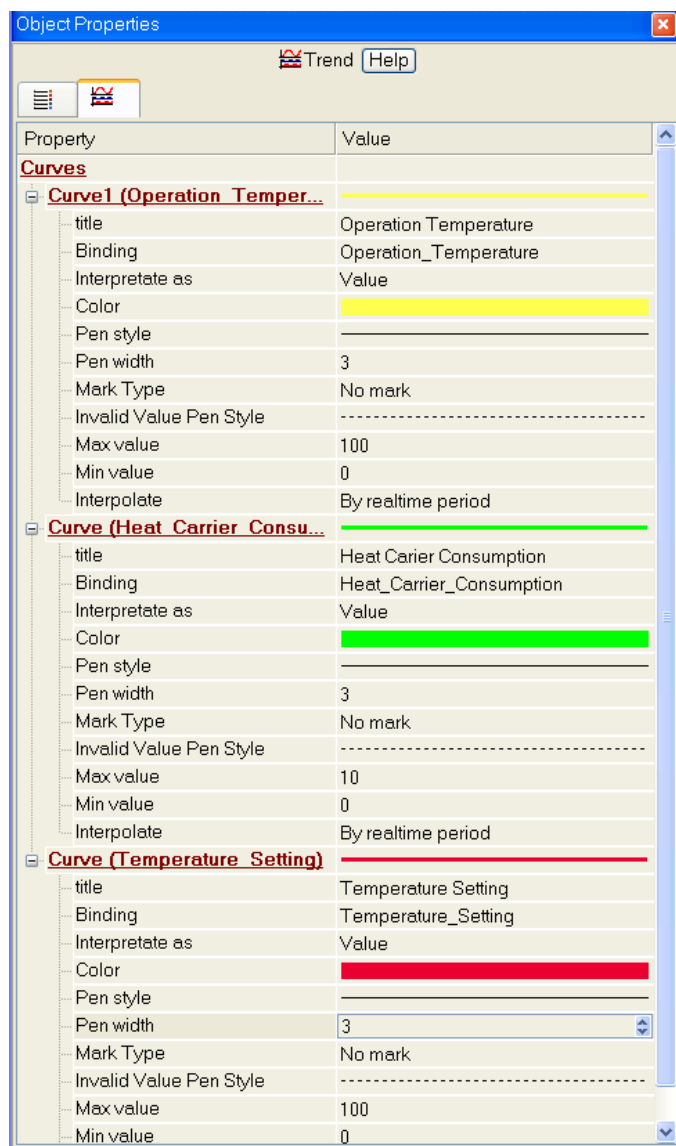
Następnie ustaw właściwości prawego Pointer Gage:






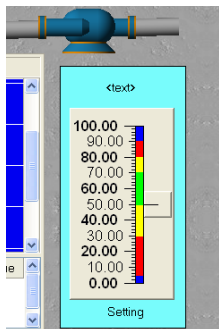
Przy użyciu ikony  umieść element 'Trend' w dolnej części ekranu, aby wyświetlać następujące parametry: **Operation Temperature**, **Heat Carrier Consumption** i **Temperature Setting**. Pozostaw główne właściwości elementu jako domyślne i dodaj nagłówek **Thermotreatment Section**.




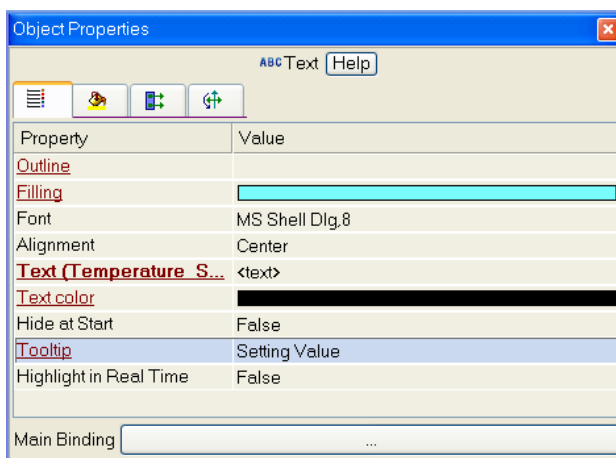
Dodaj trzy krzywe do wyświetlania na trendzie, połącz je z odpowiednimi argumentami kanałów i ustaw kolor, grubość linii oraz zakres wyświetlanych wartości.



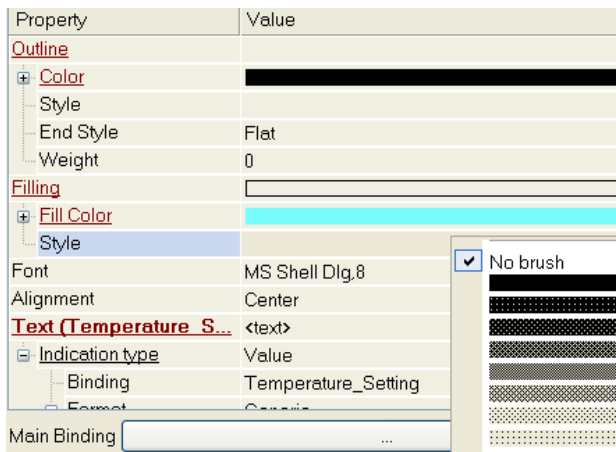
Do zadawania parametrów pętli PDD umieść element Rectangle (  ) po prawej stronie od trendu. Będzie to podstawa pod element Slider (  ), przy pomocy którego będą wyświetlane oraz ustawiane wartości. Zadana wartość będzie wyświetlana w górnej części prostokąta przy użyciu elementu Text (  ).



GE  będzie miał następujące właściwości:

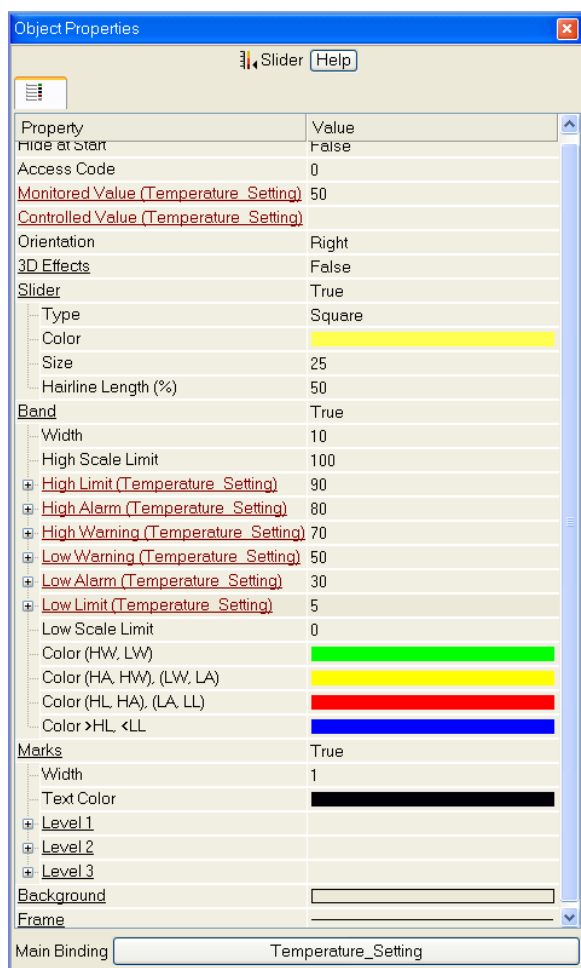


Dla tego elementu nie będzie używane obramowanie oraz wypełnienie, ustaw następujące właściwości:






Właściwości elementu  powinny być następujące:



Obramowanie i wypełnienie nie jest używane.


Umieść nagłówek – **Thermotreatment Section** – w prawym górnym rogu ekranu.

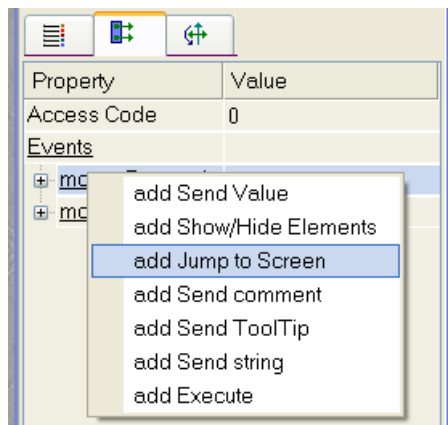
Z racji tego, że do stacji roboczej zostaną dodane jeszcze dwa ekrany (**Storage** i **Dosage**), należy zapewnić możliwość przełączania między nimi. Można do

tego celu wykorzystać element graficzny .

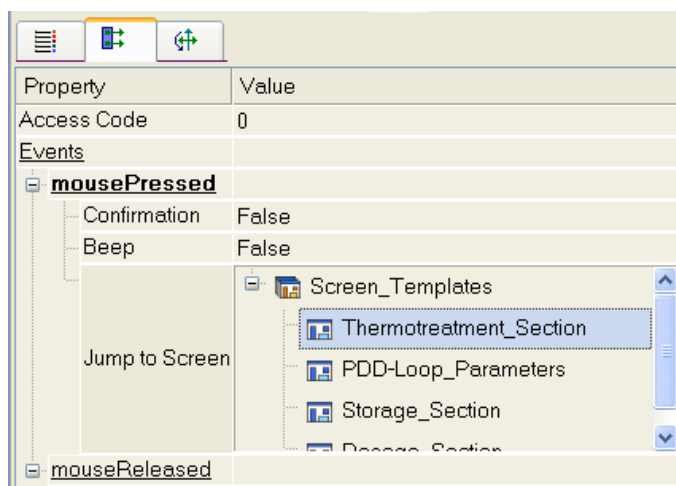


Kliknij dwukrotnie na element **Storage**, aby otworzyć jego okno Properties.


Wybierz zakładkę Events (  ) i kliknij dwukrotnie opcję **pressed** (zdarzenie wywołane kliknięciem lewym przyciskiem myszy na element). Kliknij prawym przyciskiem myszy, aby wyświetlić listę pop-up. Wybierz **Jump to Screen**:




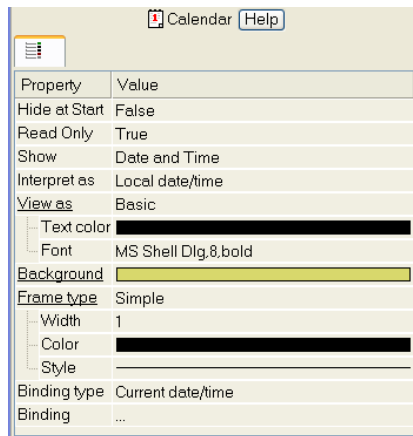
Kliknij lewym przyciskiem myszy na **Jump to Screen** i wybierz szablon ekranu **Storage Section**:



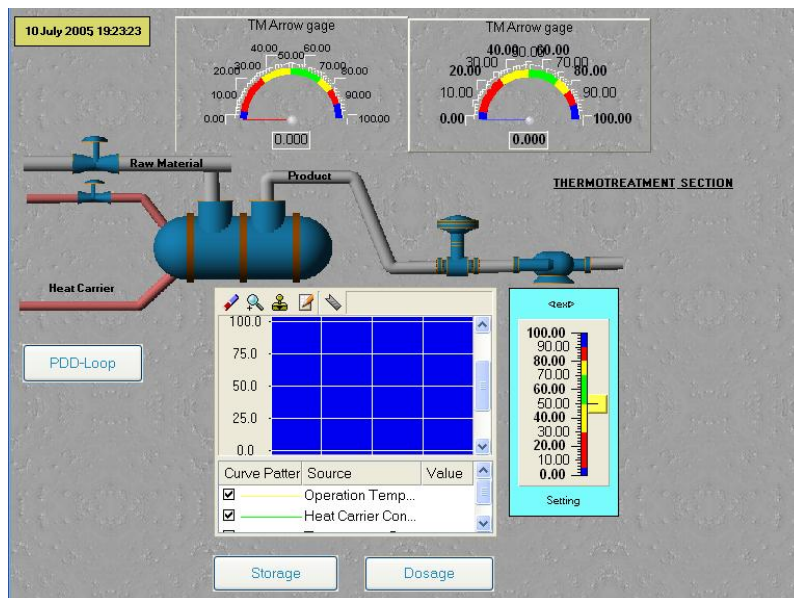
W ten sam sposób utwórz przełączanie do ekranu **Dosage Section**.

Ustawianie parametrów pętli PID (Kp, Kd, Kdd oraz Dead Zone) będzie odbywało się przy użyciu okna pop-up. Okno będzie otwierane przyciskiem , który należy umieścić w lewej części ekranu. Przechodzenie do okna powinno być zrealizowane w sposób pokazany wcześniej.

Do wyświetlania daty i czasu w lewym, górnym rogu ekranu wykorzystaj element **Calendar** (  ). Jego ustawienia powinny być następujące:



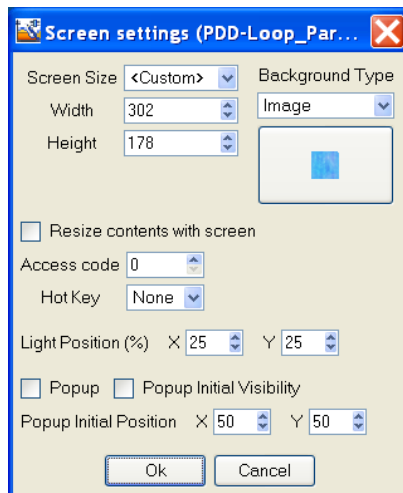
Po wykonaniu powyższych czynności ekran **Thermotreatment\_Section** jest już gotowy i może wyglądać w następujący sposób:








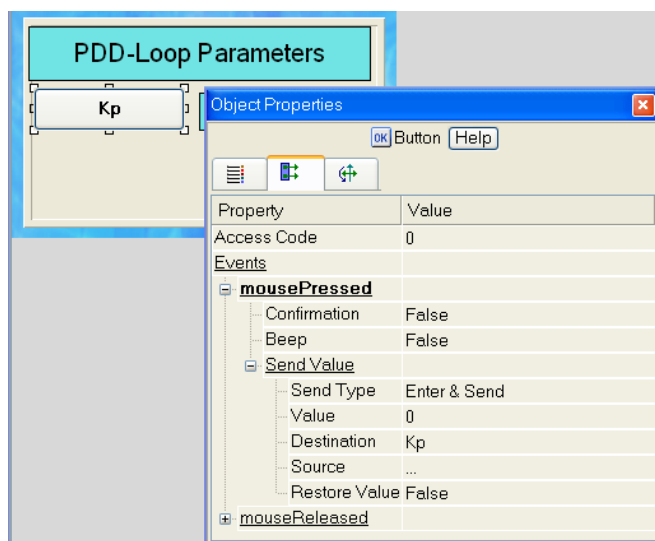
Ustaw następujące argumenty ekranu **PDD-Loop**:

Name	IO Type	Data Type	Def. Value	Link	Flags	Group
Kp	OUT	REAL				
Kd	OUT	REAL				
Kdd	OUT	REAL				
Dead_Zone	OUT	REAL				

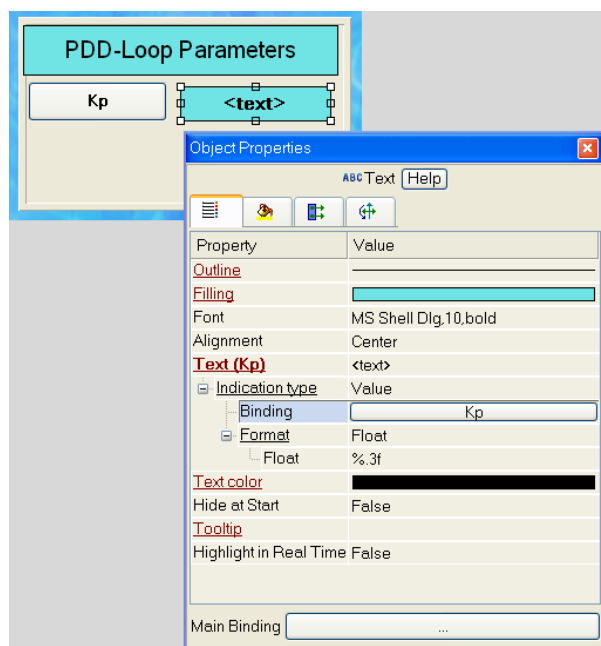
Otwórz ekran do edycji. Do ustawienia właściwości okna pop-up wybierz opcję **Tools** z głównego menu, a w niej opcję **Screen Parameters**. W oknie, które się wyświetli ustaw rozmiar ekranu, typ tła, ekran jako pop-up oraz początkową pozycję okna pop-up:




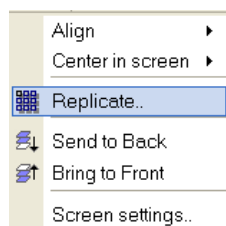
Wstaw element Frame () na ekranie i umieść go w tle klikając na ikonę  na pasku narzędzi. Umieść nagłówek ekranu na górze korzystając z GE Text () . Nagłówek to **PDD-Loop\_Parameters**. Następnie dodaj GE Button () , aby wysyłać wartości oraz element Text () na lewo od przycisku, aby je wyświetlać. Połącz elementy graficzne z argumentami ekranów:



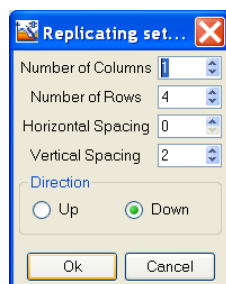
oraz




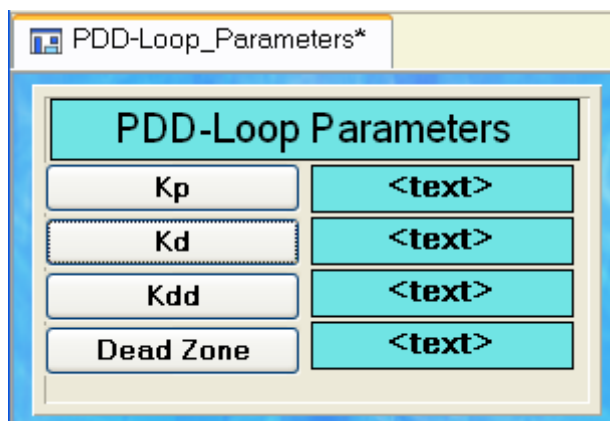
Wybierz GE **Button** () i skorzystaj z narzędzia replikacji:



Ustaw parametry w oknie, które się pojawi:




Edytuj nazwy i połączenia utworzonych elementów graficznych, następnie zrób to samo dla elementów Text (). Rezultat będzie następujący:



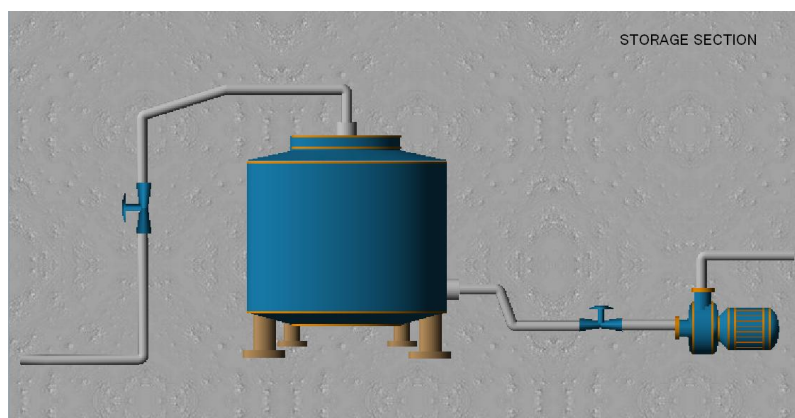
Argumenty dla szablonu ekranu **Storage\_Section** powinny być następujące:


Name	IO Type	Data Type	Def. Value	Link	Flags
Level	↓ IN	REAL			
Temperature	↓ IN	REAL			
Pressure	↓ IN	REAL			
Humidity	↓ IN	REAL			
Door	↓ IN	USINT			
Ventilation	↓ IN	USINT			
Fire_Alarm	↓ IN	USINT			
Event_Door	↕ IN/OUT	REAL			NP
Event_Vent	↕ IN/OUT	REAL			NP
Event_Fire	↕ IN/OUT	REAL			NP
Date_Time	↓ IN	DATE_AND_TIME			NP
Main_Layer	↓ IN	USINT			NP
Trend_Layer	↓ IN	USINT			NP

Argumenty **Event\_Door**, **Event\_Vent** oraz **Event\_Fire** są przeznaczone do wyświetlania i zatwierdzania zdarzeń przy użyciu GE Event (🔔). Argumenty **Main\_Layer** i **Trend\_Layer** będą wykorzystane do zarządzania wyświetlaniem powłok. Okno synoptyczne będzie wyświetlane w pierwszej powłoce, wartości parametrów magazynowania będą wyświetlane na trendzie w drugiej powłoce. Flaga **NP** ustawiona dla argumentów wyłączy tworzenie odpowiednich kanałów w trakcie procedury autobuilding.

Ustaw jedną z tekstur dostępnych w bibliotece jako tło ekranu. Do utworzenia statycznych części ekranu wykorzystaj obiekty graficzne  oraz elementy

graficzne  i  :




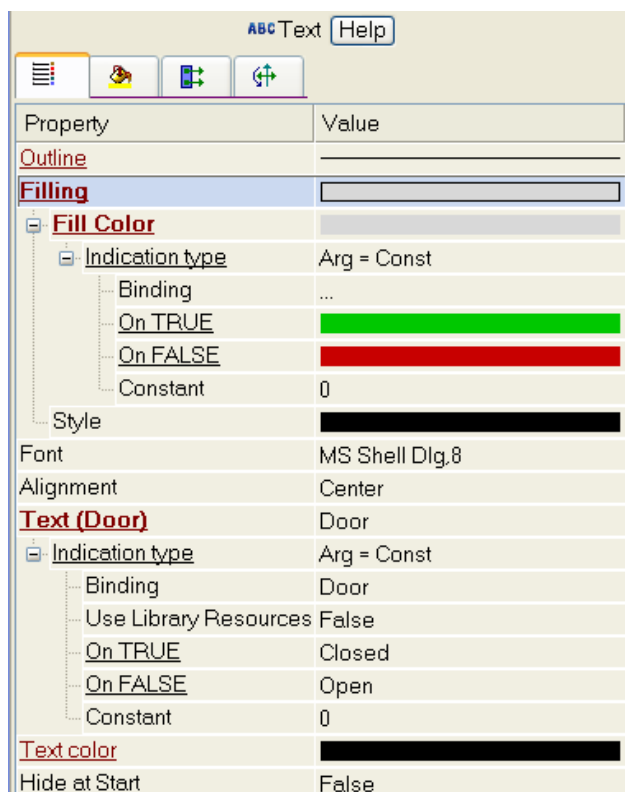
Wyświetl parametry magazynowania przy użyciu GE Text ():

Storage Parameters	
Level , m	<text>
Temperature "C	<text>
Pressure	<text>
Humidity	<text>
Door	
Ventilation	
Fire Alarm	

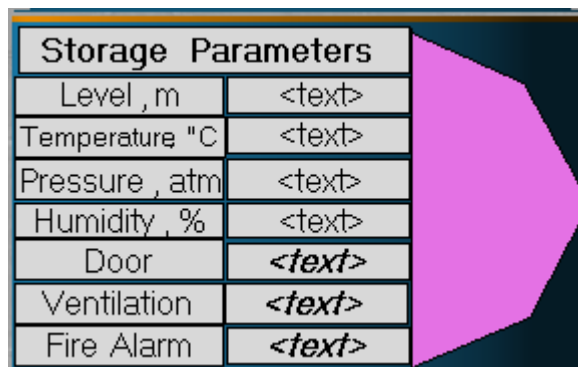
Połącz elementy graficzne z argumentami szablonu ekranu, ustaw format wyświetlania, jak np. dla argumentu **Level**:

<b>Text (Level)</b>	Level, m
Indication type	Value
Binding	Level
Format	Float
Float	%.3f

Wykorzystaj kolor i tekst zdefiniowany dla GE Text () , aby wyświetlać stan sygnałów binarnych otwarcia/zamknięcia drzwi w sekcji magazynowania, załączenia/wyłączenia wentylacji oraz uruchomienia alarmu przeciwpożarowego. Ustaw następujące właściwości, aby wyświetlać stan drzwi w sekcji magazynowania:

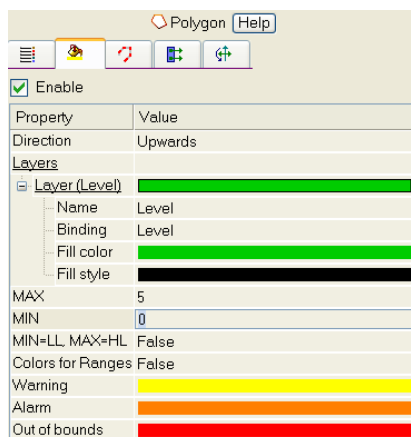




Ustaw właściwości pozostałych elementów graficznych w ten sam sposób. Poziom produkt w zbiorniku będzie wyświetlany przy pomocy dowolnego wykresu słupkowego, utworzonego np. przy pomocy GE Polygon ().




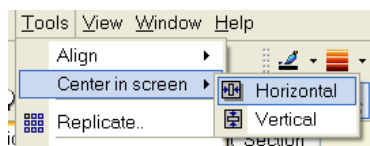
Ustaw dynamiczne wypełnienie dla tego elementu graficznego i połącz go z odpowiednim argumentem szablonu ekranu; ustaw tło, kolory wypełnienia oraz określ granice:




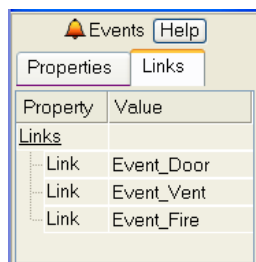


Następnie skopiuj (z szablonu ekranu **Thermotreatment\_Section** do szablonu **Storage\_Section**) elementy graficzne do wyświetlania aktualnego czasu oraz przyciski przechodzenia do innych ekranów. Aby to zrobić zaznacz odpowiedni GE (możesz zaznaczyć kilka elementów graficznych na raz trzymając klawisz **Ctrl** lub zaznaczając obszar myszką trzymając jej lewy przycisk) i kliknij  z paska narzędzi. Wklej GE przy pomocy ikony . Możesz użyć następujących, standardowych kombinacji klawiszy do kopiowania i wklejania: **Ctrl-C** i **Ctrl-V**. Po wklejeniu elementów musisz połączyć je z argumentami aktualnego szablonu ekranu.

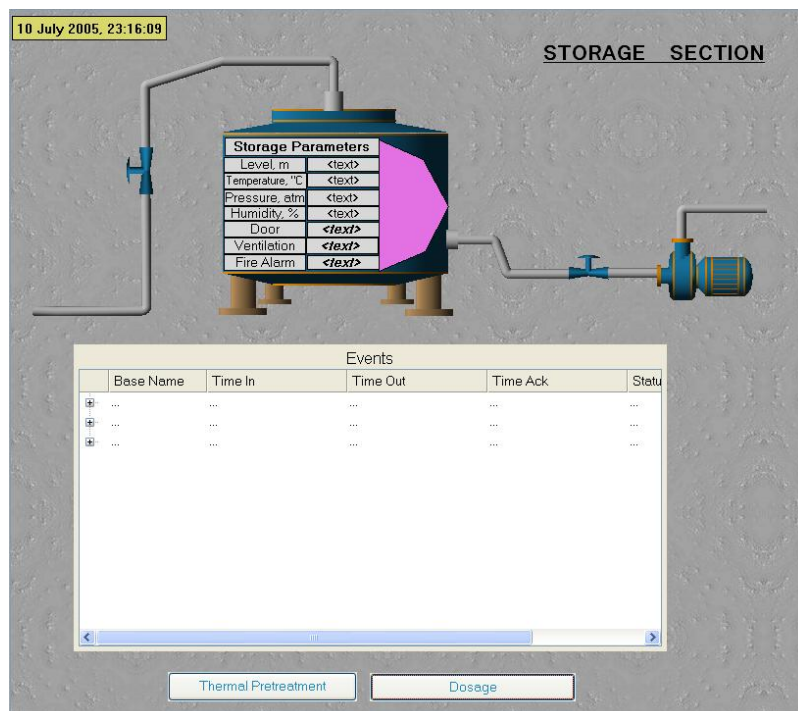
Umieść GE Events () w dolnej części ekranu, zaznacz go lewym przyciskiem myszy i wyrównaj poziomo na ekranie przy użyciu odpowiednich opcji menu:



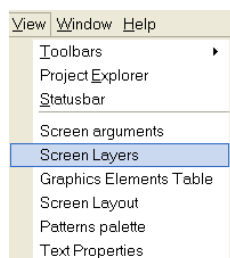
Zostaw główne funkcje GE  bez zmian i ustaw trzy połączenia z argumentami szablonu ekranu w zakładce **Links**:




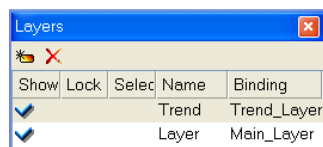
W ten sposób ekran będzie wyglądał następująco:



Do wyświetlania aktualnych wartości parametrów magazynowania na trendzie stwórz dodatkową powłokę w szablonie ekranu **Storage\_Section**. Otwórz okno Screen Layers przy użyciu opcji **View** z głównego menu:

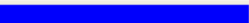



W oknie Layers zmień nazwę istniejącej powłoki na **Layer** i połącz ją z argumentem **Main\_Layer**. Kliknij na ikonę , aby utworzyć nową powłokę, zmień jej nazwę na **Trend** i połącz z argumentem **Trend\_Layer**.


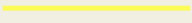
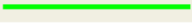
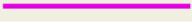
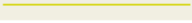
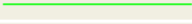
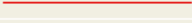


Umieść w powłoce (aby to zrobić trzeba ją zaznaczyć w oknie Layers) elemen-


ty graficzne Trend (  ) i Button (  ). Ustaw następujące właściwości dla GE Trend:

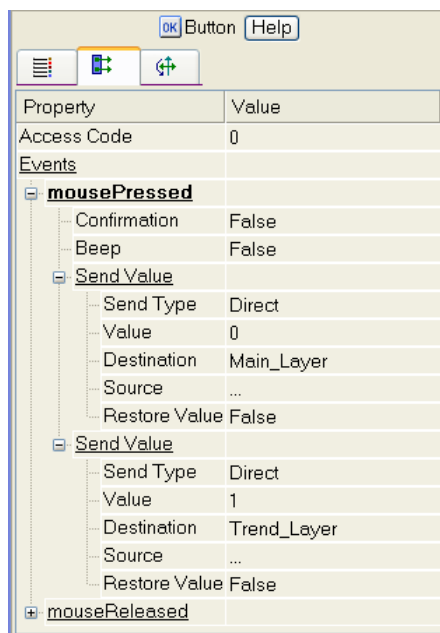
Property	Value
Orientation	Horizontal
Background color	
Base font	MS Shell Dlg.8
Resizeable	False
Caption	Storage Parameters
<u>Grid</u>	
<u>Legend</u>	
Cursor color	
<u>Time axe</u>	
<u>Value axe</u>	
Buffer size	500
<u>State colors</u>	
Hide at Start	False
Highlight in Real Time	False

Dodaj siedem krzywych:

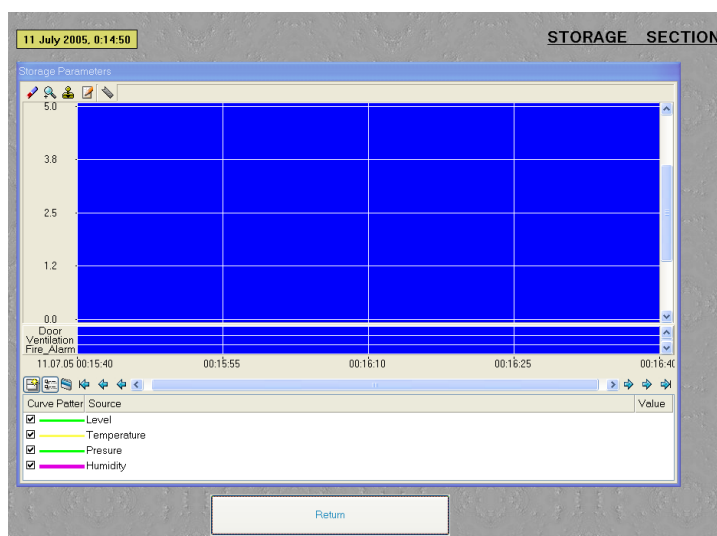
Property	Value
<b>Curves</b>	
+ Curve1 (Level)	
+ Curve2 (Temperature)	
+ Curve3 (Pressure)	
+ Curve4 (Humidity)	
+ Curve5 (Door)	
+ Curve6 (Ventilation)	
+ Curve7 (Fire Alarm)	

Ustaw następujące właściwości dla krzywych: połączenia z odpowiednimi argumentami, kolor i szerokość linii, granice oraz nazwy, dzięki którym można będzie zidentyfikować je na ekranie. Zdefiniuj właściwość **Events** dla GE

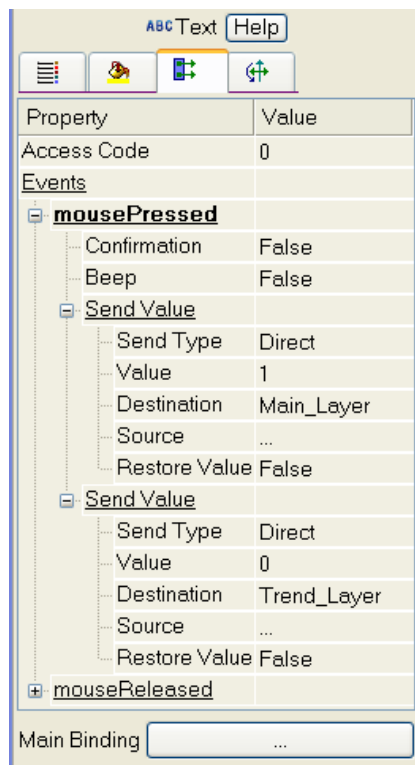
Button (  ) w następujący sposób: po naciśnięciu przycisku nastąpi bezpośrednio wysłanie wartości do argumentów ekranu, dla których określone są połączenia z powłokami ekranu. Wartości wysyłane do argumentów sterują wyświetlaniem powłok: 0 – powłoka jest wyświetlana, 1 (dowolna wartość różna od 0) – powłoka jest niewidoczna:



Elementy graficzne Text (ABC) do wyświetlania aktualnego czasu i nazwy sekcji należy skopiować do powłoki **Trend** z powłoki **Layer**. Powłoka trendu szablonu ekranu **Storage\_Section** będzie wyglądała w następujący sposób:



Ustaw następujące właściwości tekstowego elementu graficznego **Storage Parameters** (znajduje się w powłoce **Layer**), aby przejść do przeglądania wartości magazynowania na trendzie, umieszczonym w powłoce **Trend**:

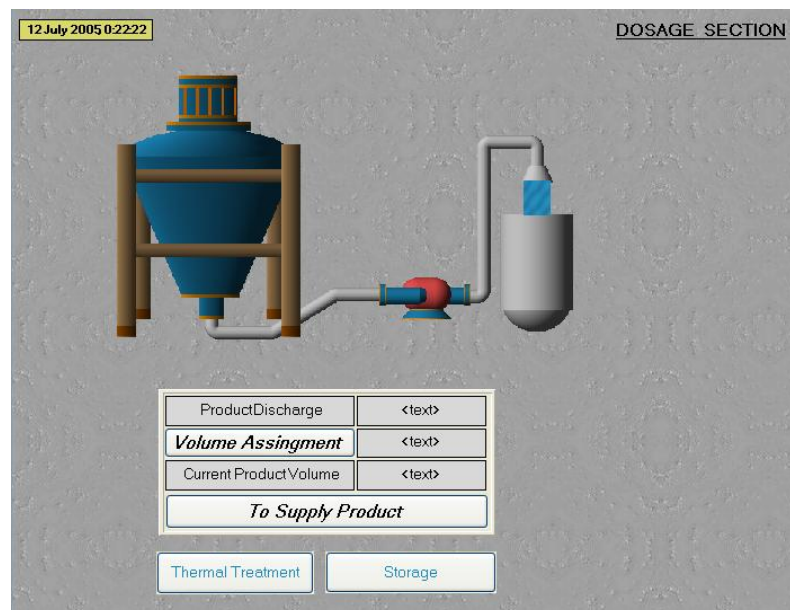


Aby umożliwić poprawną pracę powłok w trakcie działania projektu w czasie rzeczywistym, upewnij się przed zapisaniem projektu na HDD, że wszystkie powłoki są zaznaczone w kolumnie Show, w lewej części okna Screen Layers.


Ustaw argumenty dla ekranu **Dosage\_Section** następująco:


Name	IO Type	Data Type	Def. Value	Link	Flags
Product_Consumption	IN	REAL			
Volume_Setting	OUT	REAL			
ON_OFF	OUT	USINT			
Current_Product_Volume	IN	REAL			
Pump	IN	USINT			
Volume_Reset	OUT	USINT			NP
Date_Time	IN	DATE_AND_TIME			NP

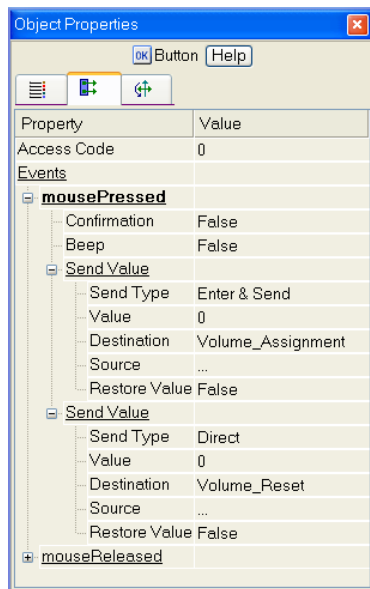
Korzystając z metod opisanych wcześniej umieść elementy graficzne i połącz je z argumentami ekranu w taki sposób, aby uzyskać następujący rezultat:



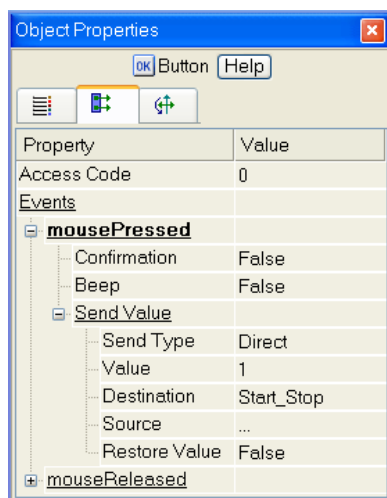
Animacja przedstawiająca przepływ produktu jest połączona z argumentem ekranu w następujący sposób:


Property	Value
<u>Video Clip</u>	
Binding	Pump
<u>Pause</u>	0
Show on Stop	False
Run Continuously	True
Hide at Start	False
<u>Tooltip</u>	Product Stream
Highlight in Real Time	False

Element graficzny **Setting Volume** () wymaga pewnego wyjaśnienia. Kliknięcie go lewym przyciskiem myszy spowoduje wysłanie wartości do dwóch argumentów ekranu:



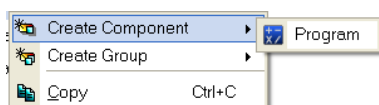
Pierwsza określa ilość dozowanego produktu, podczas gdy druga resetuje licznik ilości produktu. Po kliknięciu GE **PRODUCT SUPPLY** lewym przyciskiem myszy odpowiednie argumenty ekranu otrzymają wartość "1". Gdy zadana objętość produktu zostanie dostarczona, sterownik ustawi atrybut **Input Value** kanału na "0" (w stacji roboczej), który jest połączony z argumentem ekranu **Start\_Stop**.



Zapisz projekt klikając ikonę . W ten sposób część graficzna projektu węzła stacji roboczej jest gotowa, należy więc przejść do pisania programu.

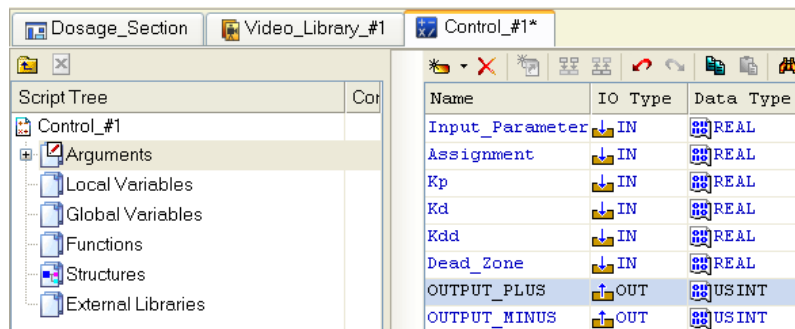
## Tworzenie programu

W rozdziale tym zostanie pokazane jak opracować programy, które będą pełniły funkcje sterujące – utrzymywanie temperatury oraz nawilżanie produktu, jak również inne, przeznaczone do pracy z sygnałami binarnymi. Wybierz powłokę **Program Templates** w lewym oknie Project Explorer i kliknij prawym przyciskiem na niej, aby dodać komponent **Program #1**:

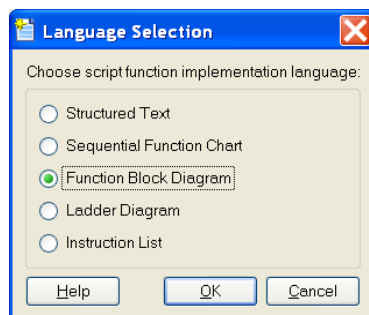


Zaznacz dodany komponent i zmień jego nazwę na **Control#1**, program będzie wgrany do kontrolera PAC z zadaniem utrzymywania odpowiedniej temperatury w urządzeniu znajdującym się w sekcji obróbki cieplnej.

Kliknij dwukrotnie na komponent Control #1, aby otworzyć edytor szablonu programu i wybierz opcję **Arguments**, aby otworzyć tabelę edytora argumentów. Dodaj argumenty do programu, które będą wchodziły w skład regulacji PID oraz PWM.




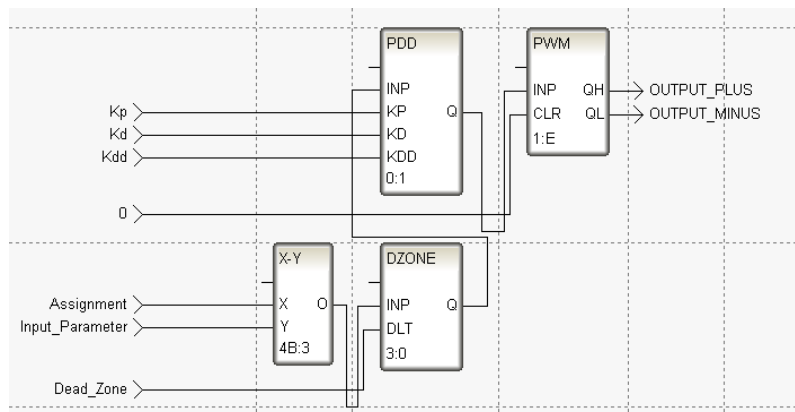
Po określeniu typu argumentów możesz przejść do realizacji programu. Kliknij lewym przyciskiem myszy na nazwę szablonu i w oknie, które się pojawi wybierz opcję **Function Block Diagram**.





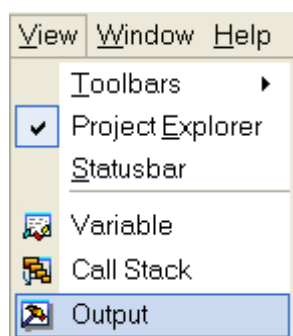
W oknie edytora programu, które się pojawi kliknij lewym przyciskiem myszy



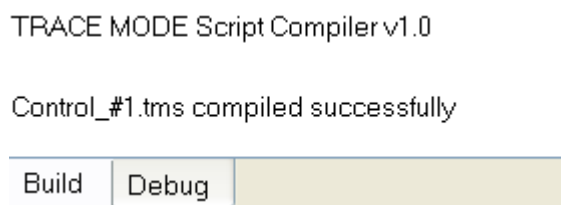
na ikonę , która zapewnia dostęp do biblioteki bloków funkcyjnych. Dodawaj odpowiednie bloczki przez zaznaczenie ich i przeciągnięcie do okna roboczego edytora. Ustaw połączenia pomiędzy wejściami i wyjściami bloków oraz przypisz połączenia do argumentów. Gotowy program powinien wyglądać następująco:



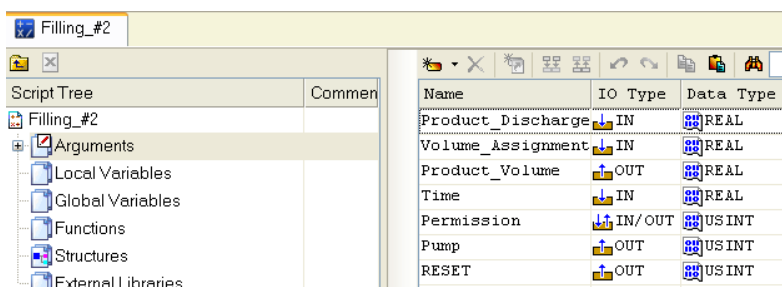
Uruchomienie debuggowania wymaga kompilacji programu. Możesz to zrobić przy użyciu ikony  z paska narzędzi lub przy użyciu klawisza **F7**. Wynik kompilacji pokazany jest w oknie **Output**, które można otworzyć z głównego menu IDE lub przez kliknięcie na ikonę  z paska narzędzi:



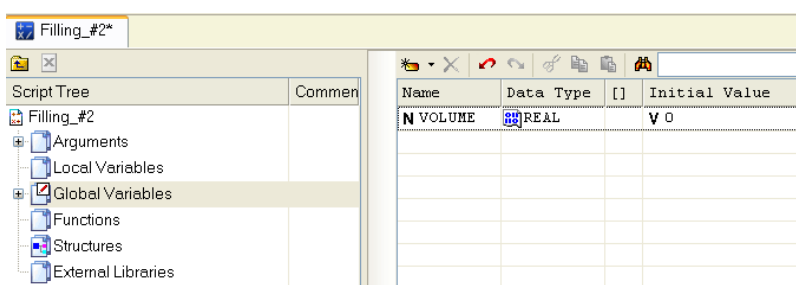
W tym przypadku okno zawiera informację o kompilacji zakończonej sukcesem:



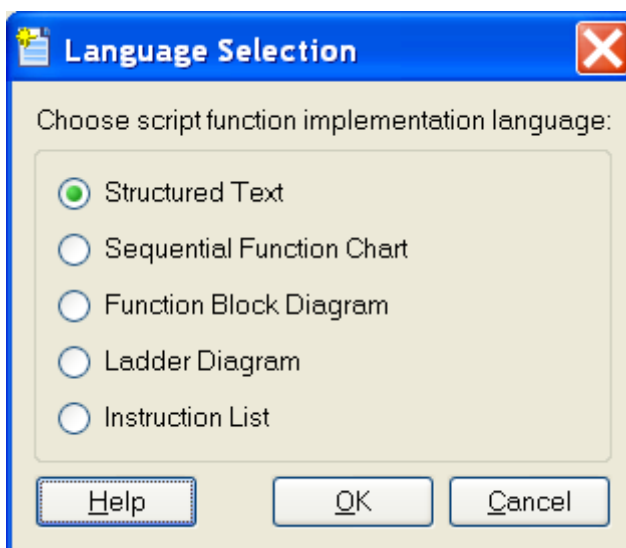
Dodaj program **Pouring#1** do sterowania nawilżaniem produktu i określ w nim następujące argumenty:



Dodaj jedną zmienną globalną, która będzie przechowywała wartość ilości dozowanego produktu:



Do utworzenia programu przy użyciu języka **Techo ST** wybierz w oknie Language Selection opcję **Structured Text**.



Kod programu może mieć następującą postać:

```

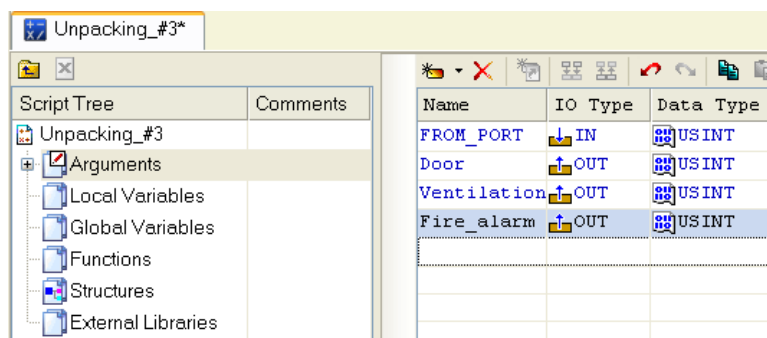
PROGRAM
VAR_INPUT Product_Discharge : REAL; END_VAR
VAR_INPUT Volume_Assignment : REAL; END_VAR
VAR_OUTPUT Product_Volume : REAL; END_VAR
VAR_INPUT Time : REAL; END_VAR
VAR_INOUT Permission : USINT; END_VAR
VAR_OUTPUT Pump : USINT; END_VAR
VAR_OUTPUT RESET : USINT; END_VAR

//Process preparation steps
if Permission==0 then
    //Keep pump ON
    Pump=0;
end_if;
//Verification of permission from workstation (Start_Stop=1 ?)
if Permission==1 then
    //Switching the pump ON and retention of Start_Stop argument
    //at the workstation
    Pump=1;
    RESET=1;
    //Calculation of accumulated product volume during each cycle
    //(time is converted from ms to min)
    //ADC codes from flowmeter are converted to physical value (l/min)
    //accumulated product value is assigned to global variable - accumulator
    VOLUME=VOLUME+Product_Discharge*0.02442*(Time/1000.0/60.0);
    Product_Volume=VOLUME;
end_if;
//Check filling condition, ignore inaccuracy
if Permission==1 && Product_Volume>=Volume_Assignment then
    //Pump switch off and reset Start_Stop argument to zero on the WS screen
    Pump=0;
    RESET=0;
    //No further volume calculation
    Permission=0;
    //Accumulator reset
    VOLUME=0;
end_if;
END_PROGRAM

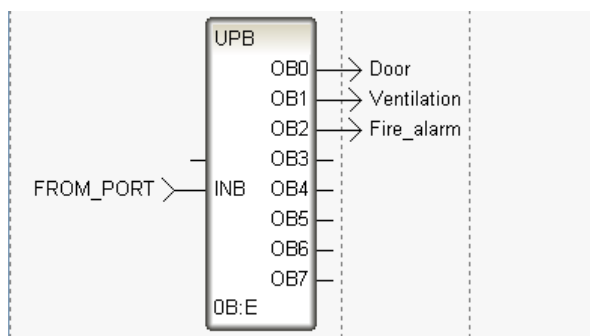
```

Skompiluj program klawiszem **F7**.

Do pracy z wejściami i wyjściami binarnymi dodaj program w postaci Techno **FBD**, o nazwie **Unpack#3**, do rozpakowywania bajtów otrzymanych ze sterownika PLC. Dodaj także program **Pack#4** służący do pakowania bitów w bajty, które są wysyłane do kontrolera PAC. Pamiętaj, że już nie trzeba tworzyć programów pakujących i rozpakowujących bity, ponieważ kanały **HEX16** zawierają atrybuty **Bit1**, **Bit2**... **Bit16** zapewniające dostęp do poszczególnych bitów. Użyj jednak tego sposobu, aby zapewnić ciągłość opisu metody 'od szablonu'. Dodaj zatem argumenty do szablonu programu Unpack#3:



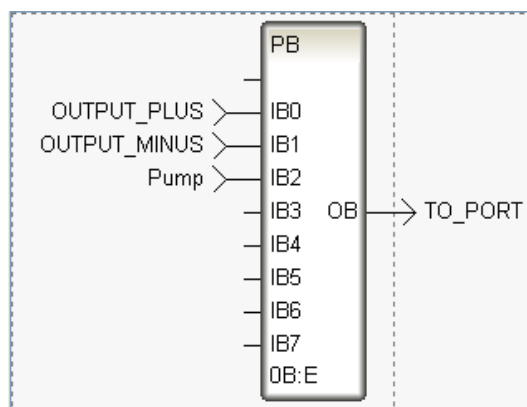
Program powinien wyglądać następująco:




Argumenty programu **Pack#4** oraz sam kod powinny wyglądać następująco:

Name	IO Type	Data Type
OUTPUT_PLUS	IN	US INT
OUTPUT_MINUS	IN	US INT
Pump	IN	US INT
TO_PORT	OUT	US INT

oraz

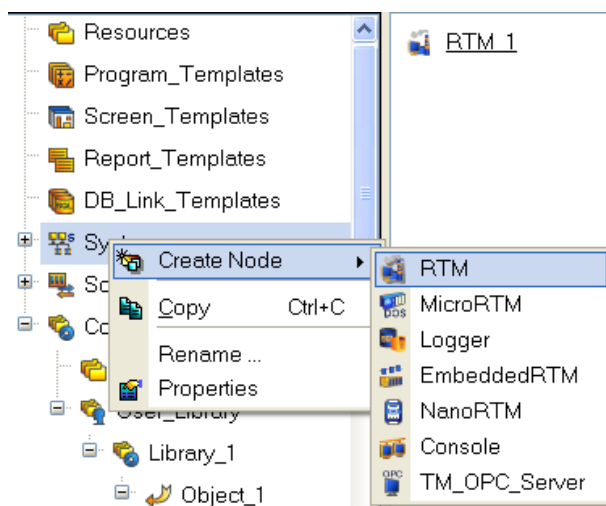


Skompiluj program klawiszem **F7**.

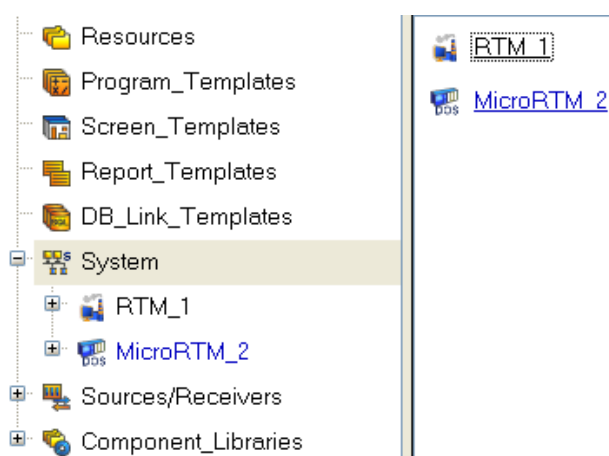
Zapisz projekt klikając na ikonę . Po dodaniu grafiki stacji roboczej i programu dla kontrolera PAC, możesz dodać węzły projektu oraz kanały w nich.


## Węzły projektu oraz kanały

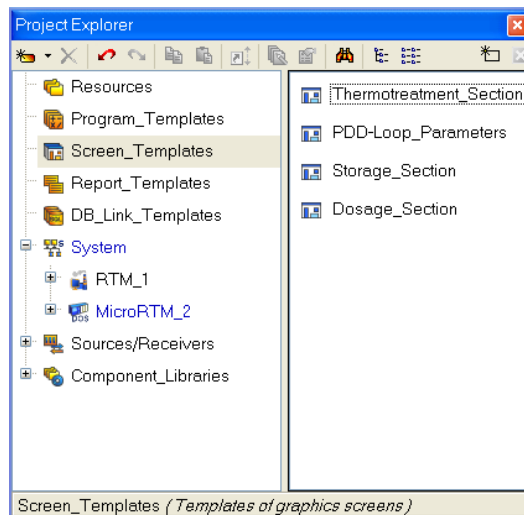
Dodaj teraz węzły projektu – stacja robocza i kontroler PAC, dla których kanały zostaną utworzone przy użyciu procedury autobuilding. Wybierz powłokę **System** klikając na niej lewym przyciskiem myszy. Następnie kliknij prawym przyciskiem myszy, aby dodać węzły (po kolei) RTM dla stacji roboczej oraz MicroRTM dla kontrolera PAC.



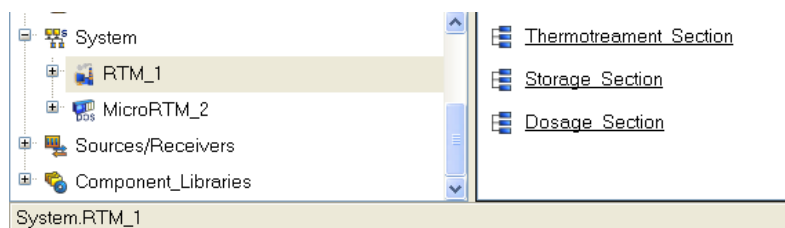
W wyniku tego zostaną dodane dwa węzły w powłoce **System**:



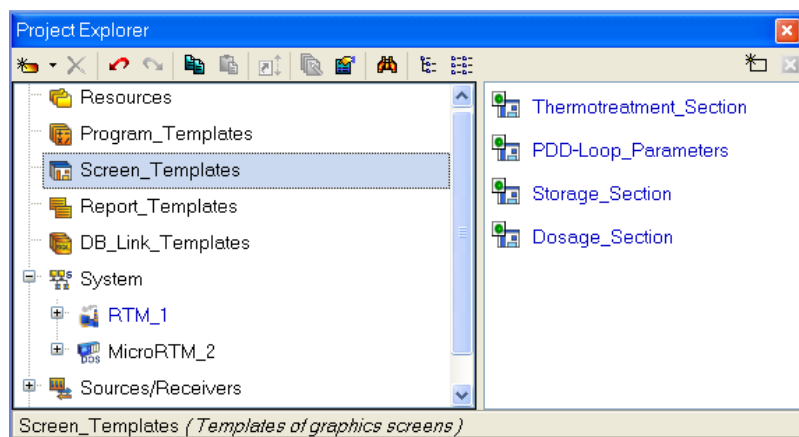
Kliknij na ikonę , aby zduplikować okno Project Explorer, otwórz powłokę **Screen Templates** w górnym oknie oraz grupę komponentów **Channels** (znajdującą się w dodanym wcześniej węźle **RTM\_1**) w oknie poniżej.

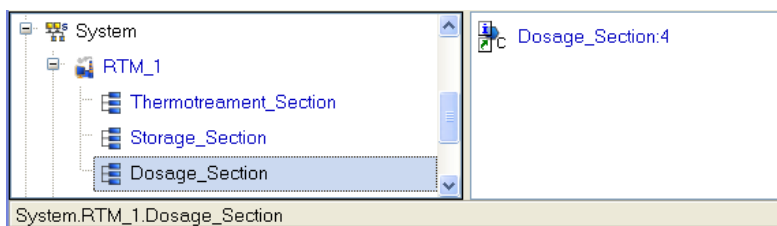


W węźle **RTM\_1** zmień nazwę grupy Channels na **Thermotreatment\_Section** i dodaj grupy z nazwami **Storage\_Section** i **Dosage\_Section**.

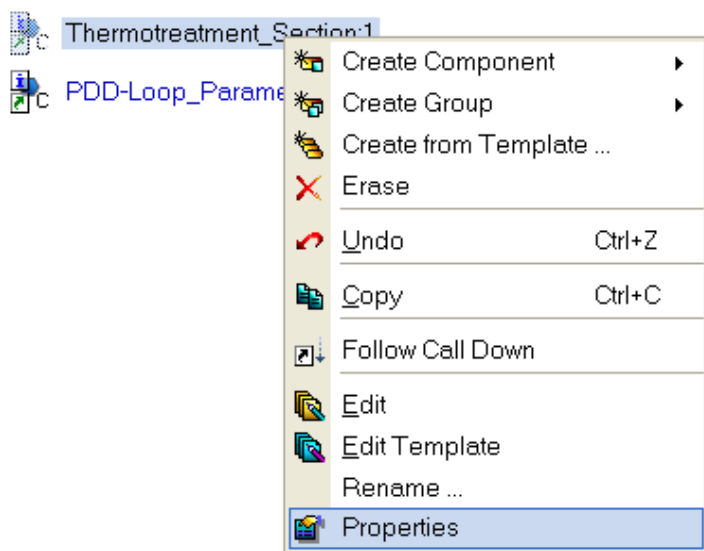



Następnie przeciągnij szablony ekranów do grup węzła **RTM\_1**.

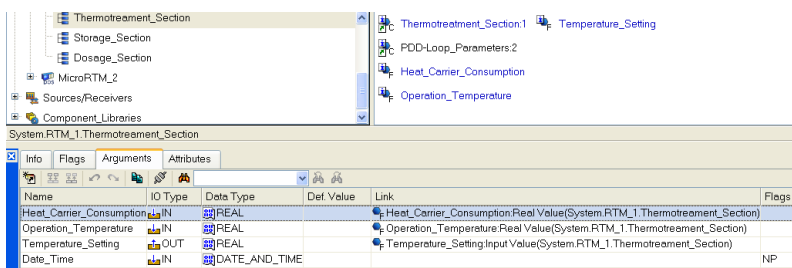


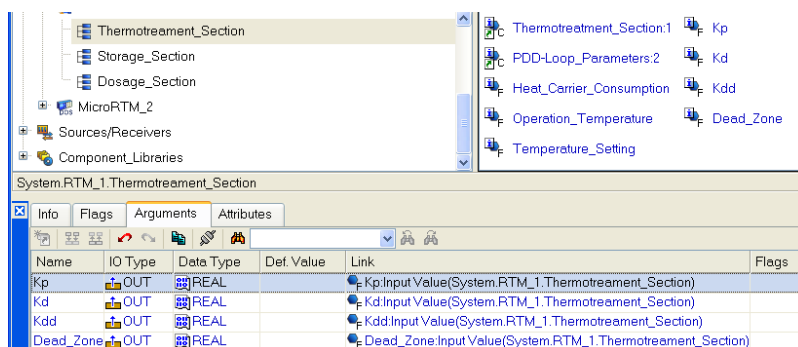


Kolejnym krokiem jest utworzenie kanałów w oparciu o argumenty szablonów ekranów, które zostały wykonane. Zaznacz grupę **Thermotreatment\_Section** (węzeł **RTM\_1**) i otwórz okno Properties kanału typu CALL o nazwie **Thermotreatment\_Section:1**.



Przejdź do zakładki **Arguments**, zaznacz (lewym przyciskiem myszy) pierwszy argument i kliknij ikonę , aby utworzyć kanały w grupie oraz automatycznie połączyć ich atrybuty z argumentami szablonu ekranu:





Dodaj kanały w **Storage\_Section** i **Dosage\_Section** w ten sam sposób. Najpierw jednak ustaw domyślne wartości argumentów **Main\_Layer** oraz **Trend\_Layer (Storage\_Section)** odpowiednio 0 i 1. Po ukończonej procedurze autobuilding otrzymasz:

Name	IO Type	Data Type	Def. Value	Link	Flags
Level	IN	REAL		Level.Real Value(System.RTM_1.Storage_Section)	
Temperature	IN	REAL		Temperature.Real Value(System.RTM_1.Storage_Section)	
Pressure	IN	REAL		Pressure.Real Value(System.RTM_1.Storage_Section)	
Humidity	IN	REAL		Humidity.Real Value(System.RTM_1.Storage_Section)	
Door	IN	USINT		Door.Real Value(System.RTM_1.Storage_Section)	
Ventilation	IN	USINT		Ventilation.Real Value(System.RTM_1.Storage_Section)	
Fire_Alarm	IN	USINT		Fire_Alarm.Real Value(System.RTM_1.Storage_Section)	
Event_Door	IN/OUT	REAL			NP
Event_Vent	IN/OUT	REAL			NP
Event_Fire	IN/OUT	REAL			NP
Date_Time	IN	DATE_AND_TIME			NP
Main_Layer	IN	USINT			NP
Trend_Layer	IN	USINT			NP

Dla **Dosage\_Section**:

Name	IO Type	Data Type	Def. Value	Link	Flags
Product_Consumption	IN	REAL		Product_Consumption.Real Value(System.RTM_1.Dosage_Section)	
Volume_Assignment	OUT	REAL		Volume_Assignment.Input Value(System.RTM_1.Dosage_Section)	
Start_Stop	OUT	USINT		Start_Stop.Input Value(System.RTM_1.Dosage_Section)	
Current_Product_Volume	IN	REAL		Current_Product_Volume.Real Value(System.RTM_1.Dosage_Section)	
Pump	IN	USINT		Pump.Real Value(System.RTM_1.Dosage_Section)	
Volume_Reset	OUT	USINT			NP
Date_Time	IN	DATE_AND_TIME			NP

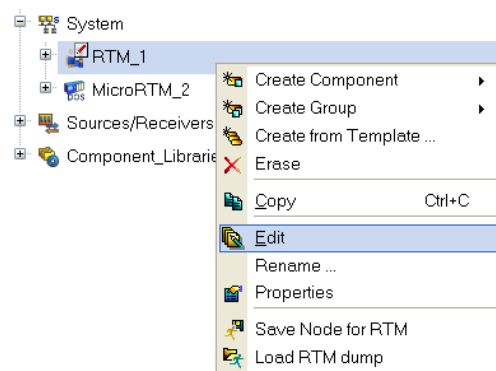
Kolejnym krokiem będzie ustawienie parametrów archiwum oraz Alarm Report (raport alarmów).

## Archiwum oraz alarmy

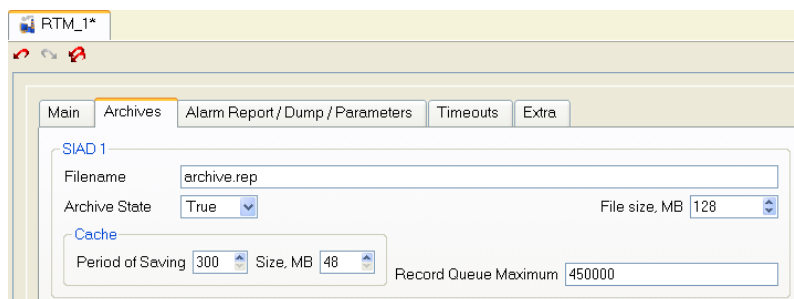
W celu archiwizowania wartości kanałów w bazie danych SIAD/SQL 6 i zapisywania informacji o zdarzeniach takich jak przekroczenie zadanego progu do raportu alarmów, należy skonfigurować kilka parametrów węzła, a następnie ustawić (indywidualnie lub grupowo) kanałom odpowiednie atrybuty **SIAD** i/lub **Alarm Report**.



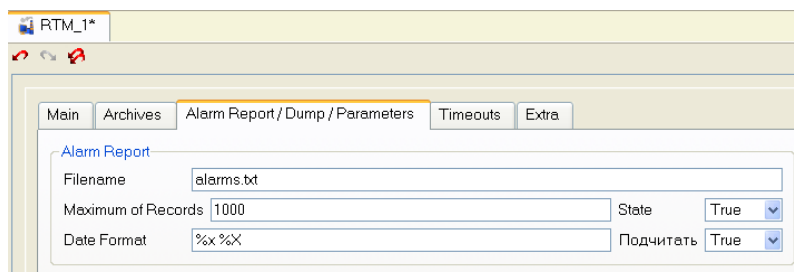
Kliknij prawym przyciskiem myszy na RTM\_1 i wybierz opcję Edit (edycja). Wyświetli się okno, w którym możesz określić ustawienia wymiany danych.



Przejdź do zakładki **Archives** i ustaw parametry techniczne archiwizowania dla pierwszego archiwum (**Type1**).



Ustaw następujące parametry w zakładce **Alarm Report/Dump/Parameters**:

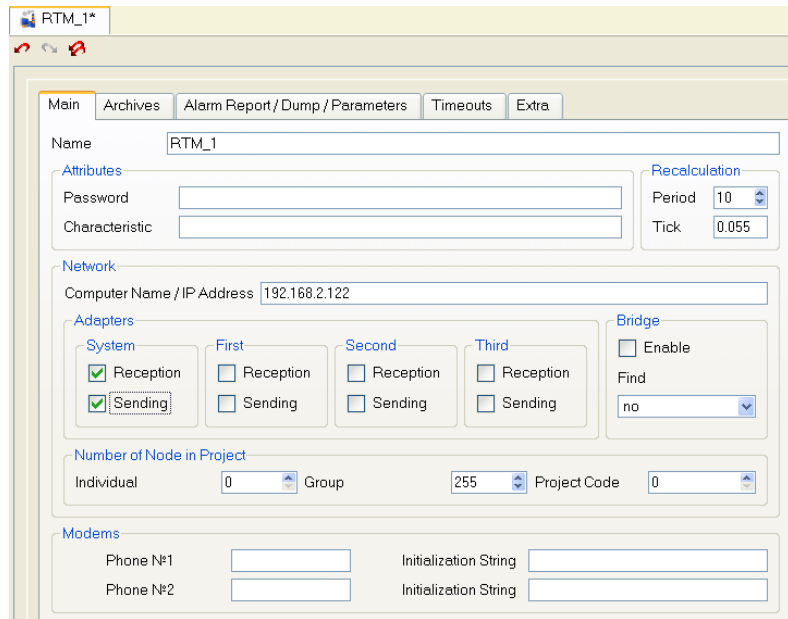


W zakładce **Main** ustaw adres IP stacji roboczej i kierunek przepływu danych przez sieć.

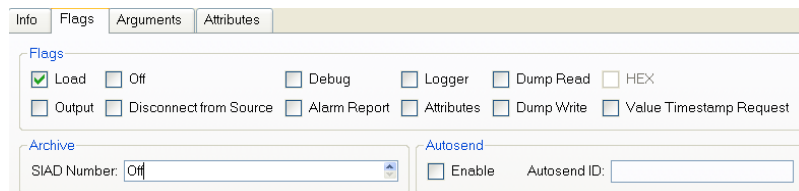
---

Wprowadzanie adresu IP nie jest obowiązkowe, ponieważ RTM identyfikuje ten parametr na starcie i wykorzystuje go podczas wymiany danych w sieci.

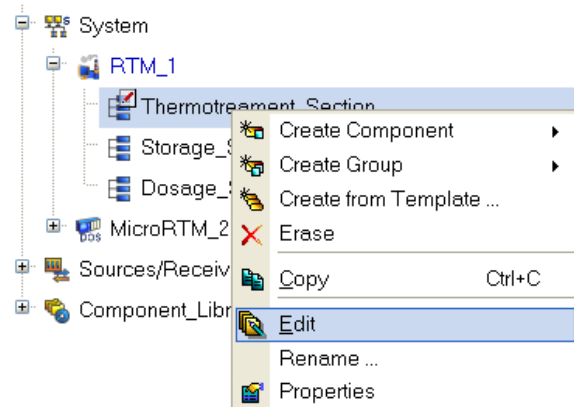
---



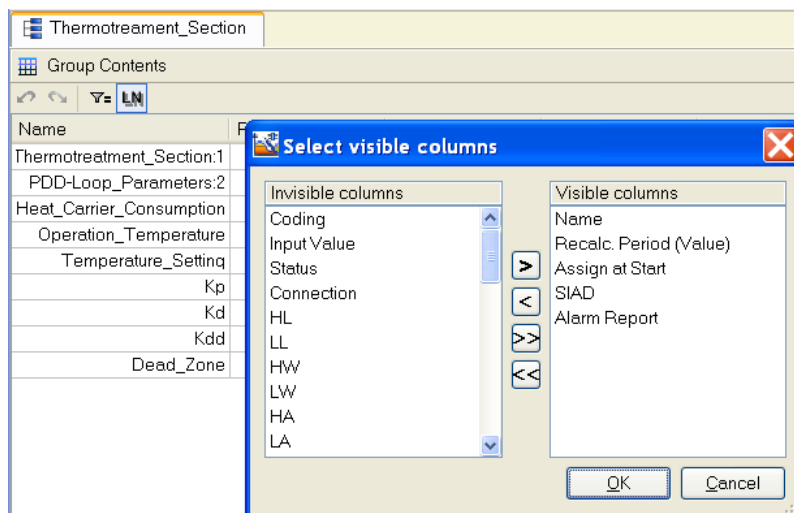
Zamknij okno edycji **RTM\_1**. Otwórz okno właściwości grupy **Thermothreatment\_Section** i ustaw flagę **Load Group** w zakładce **Flags** (kanały w profilerze zostaną pogrupowane). Zrób to samo dla pozostałych grup.



Dodaj kanały grupy **Thermothreatment\_Section** do archiwizowania oraz do raportu alarmów w następujący sposób:



W oknie, które się pojawi kliknij prawym przyciskiem myszy w linii atrybutów kanałów, aby otworzyć okno **Select visible columns**. W prawym oknie utwórz listę atrybutów, które będą wyświetlane w tabeli.



Zaznacz grupę kanałów lewym przyciskiem myszy (trzymając go) oraz trzymając klawisz **Ctrl**, kliknij dwukrotnie na wybranych kanałach w kolumnie SIAD i ustaw wartość 1.

Name	Recalc. Period (Value)	SIAD	Alarm Report
Thermotreatment_Section:1	1	No	0
PDD-Loop_Parameters:2	1	No	0
Heat_Carrier_Consumption	1	No	0
Operation_Temperature	1	No	0
Temperature_Setting	1	No	0
Kp	1	1	0
Kd	1	2	0
Kdd	1	3	0
Dead_Zone	1	No	0

W efekcie aktualne wartości wybranych kanałów będą zapisywane do archiwum nr 1. Poprawne atrybuty grupy **Thermothreatment\_Section**:

Name	Recalc. Period (Value)	Assign at Start	SIAD	Alarm Report
Thermotreatment_Section:1	1	0	No	0
PDD-Loop_Parameters:2	1	0	No	0
Heat_Carrier_Consumption	1	0	1	0
Operation_Temperature	1	0	1	0
Temperature_Setting	1	0	1	0
Kp	1	1	1	0
Kd	1	1	1	0
Kdd	1	1	1	0
Dead_Zone	1	1	1	0

Dla **Storage\_Section**:

Storage_Section					
Group Contents					
Name	Recalc. Period (Va)	Assign at Start	SIAD	Alarm Report	
Storage_Section:3	1	0	No	0	
Level	1	0	1	1	
Temperature	1	0	1	1	
Pressure	1	0	1	1	
Humidity	1	0	1	1	
Door	1	0	1	1	
Ventillation	1	0	1	1	
Fire_Alarm	1	0	1	1	

Dla **Dosage\_Section**:

Dosage_Section					
Group Contents					
Name	Recalc. Period (Va)	Assign at Start	SIAD	Alarm Report	
Dosage_Section:4	1	0	No	0	
Product_Consumption	1	0	1	0	
Volume_Assignment	1	0	1	0	
Start_Stop	1	0	1	0	
Current_Product_Volume	1	0	1	0	
Pump	1	0	1	0	

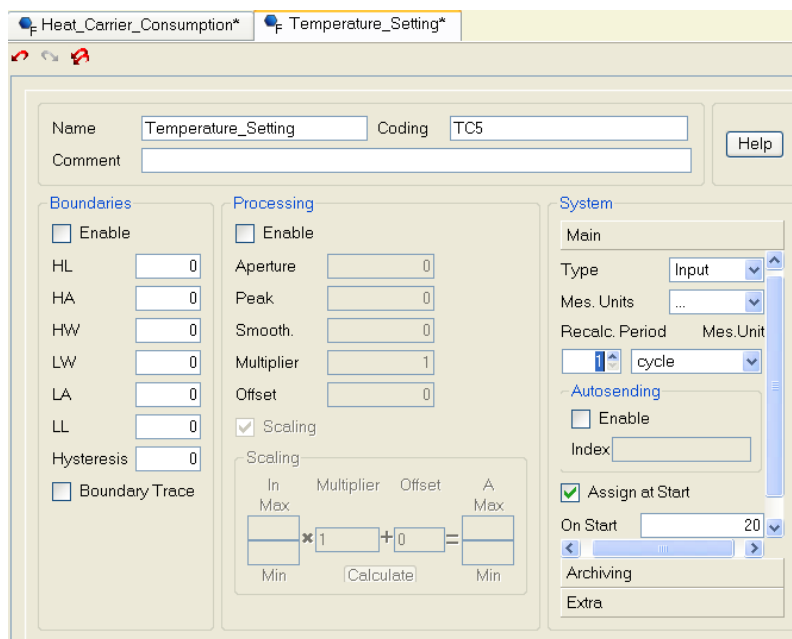
Ustaw teraz progi oraz skalowanie wielkości fizycznych dla parametrów analogowych. Pamiętaj, że wszystkie dane pochodzą z 12-bitowego przetwornika ADC, tj. mają zakres 0-4095. Poniżej jest przykład dla parametru **Heat\_Carrier\_Consumption** w grupie **Thermotreatment\_Section**:

The screenshot shows the configuration window for 'Heat\_Carrier\_Consumption\*'. The window is divided into several sections:

- Name:** Heat\_Carrier\_Consumption, **Coding:** TC5
- Boundaries:**
  - Enable
  - HL: 10, HA: 0, HW: 0, LW: 0, LA: 0, LL: 0, Hysteresis: 0
  - Boundary Trace
- Processing:**
  - Enable
  - Aperture: 0, Peak: 0, Smooth.: 0, Multiplier: 0.002442, Offset: 0
  - Scaling
  - Scaling:**
    - In Max: 4095, In Min: 0, Multiplier: 0.002442, Offset: 0, A Max: 10, A Min: 0
    - Buttons: Calculate
- System:**
  - Main:** Type: Input, Mes. Units: ..., Recalc. Period: 1, Mes. Unit: cycle
  - Autosending:**  Enable
  - Assign at Start:**  Assign at Start
  - On Start:** 0
  - Archiving:** Extra: ...

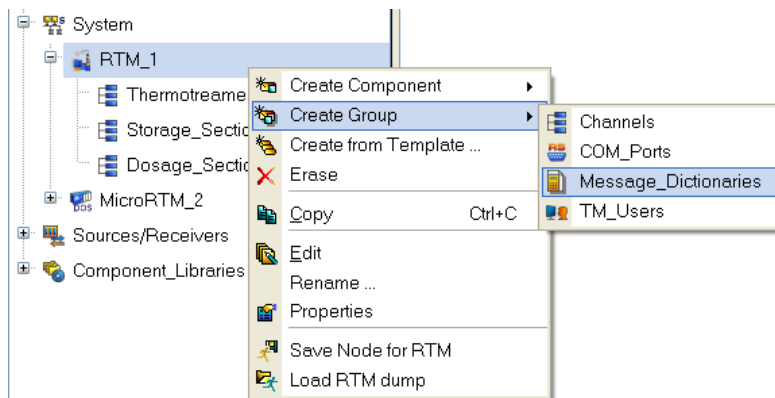
Ustaw wartości początkowe dla kanałów wysyłających polecenia do kontrolera

PAC. Dla parametru **Temperature\_Setting** może to wyglądać następująco:

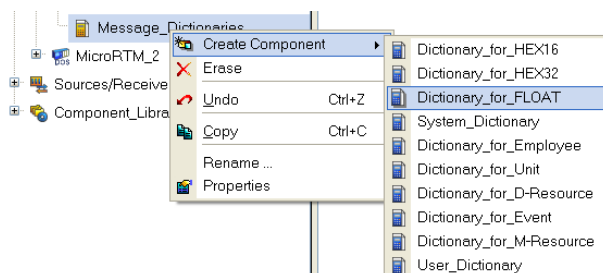


Dla innych parametrów ustaw następujące wartości:  $K_p=1$ ,  $K_d=0.1$ ,  $K_{dd}=0.01$ ,  $Dead\_Zone=0.5$  i  $Volume\_Setting=50$ .

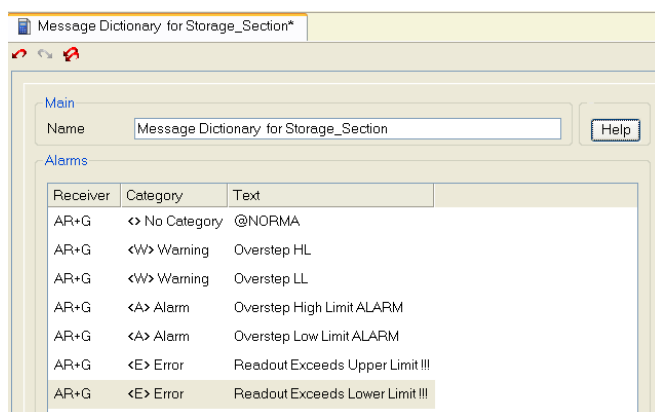
Ustaw teraz komunikaty raportu alarmów dla parametrów analogowych grupy kanałów **Storage\_Section**. W tym celu dodaj grupę **Message Dictionaries** w węźle **RTM\_1**



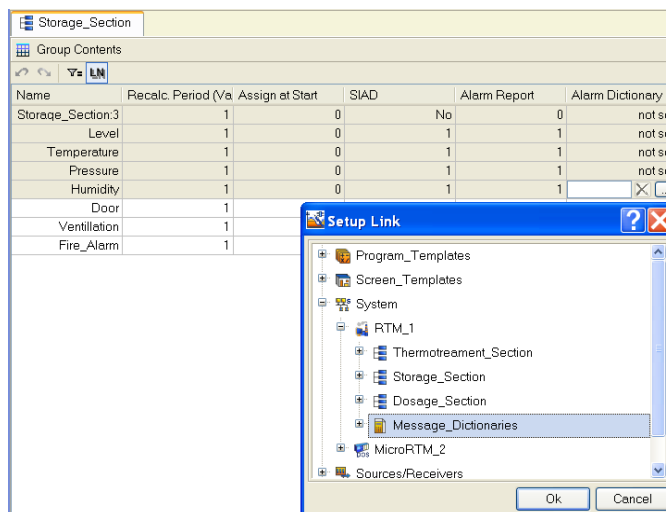
Dodaj w niej nowy element – **Message Dictionary** dla kanału **FLOAT**.



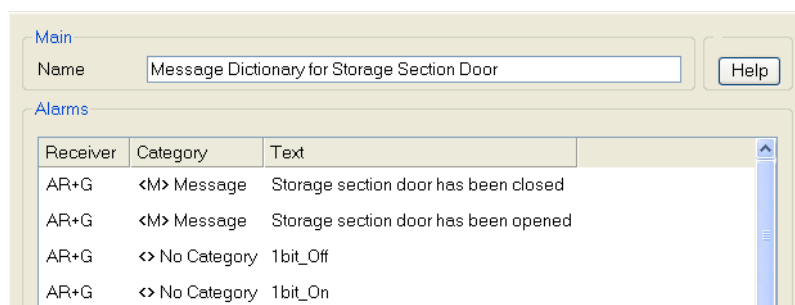
Kliknij dwukrotnie lewym przyciskiem myszy na tym elemencie, aby go edytować i dodać komunikaty raportu alarmów.



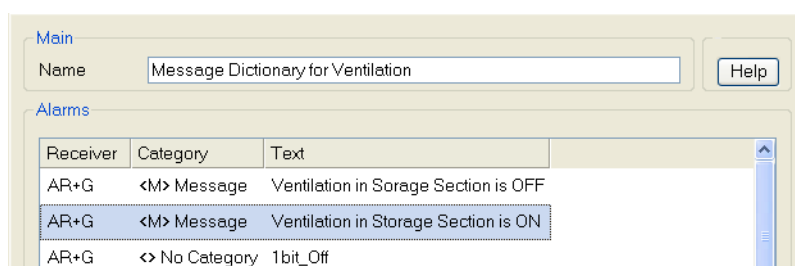
Połącz słownik z kanałami **FLOAT** grupy **Storage\_Section** przy pomocy jednego z następujących sposobów: edytuj po kolei kanały i ustawiaj dla każdego z nich w zakładce **Archiving**, w polu **Alarm Dictionary ID** odpowiedni słownik; lub skorzystaj z edytora grupowego:



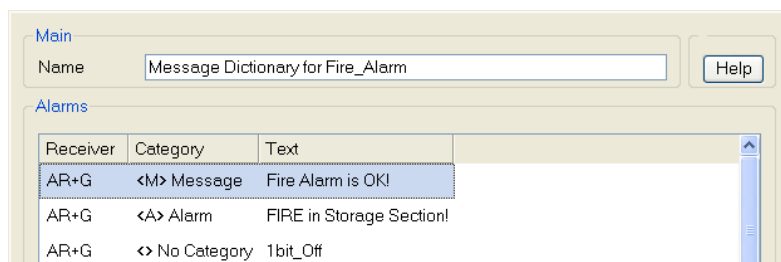
Dodaj teraz słownik komunikatów dla kanałów **HEX16**. Określ w nim komunikaty dla bitu nr 0:




Dla **Ventilation**:



Dla **Fire Alarm**:

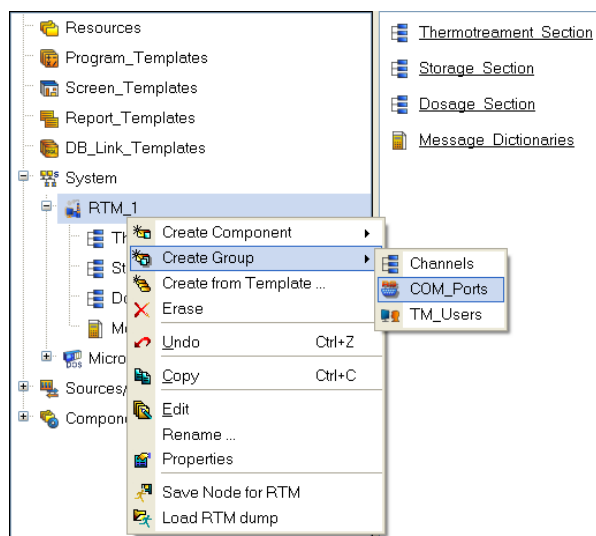


Dołącz utworzone słowniki do kanałów w ten sam sposób jak wcześniej. Zapisz projekt (  ).

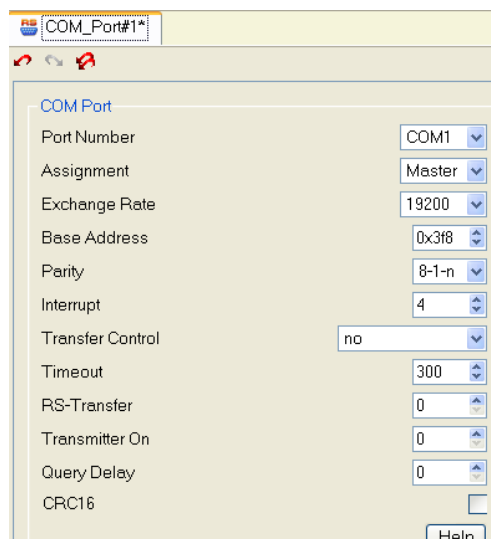
## Podłączanie PLC do stacji roboczej

### Konfiguracja portu COM

Z racji tego, że PLC obsługujące sekcję magazynowania jest podłączone do stacji roboczej przez interfejs szeregowy, należy dodać i skonfigurować port szeregowy dla stacji roboczej. Dodaj nową grupę – **COM-ports** – dla węzła **RTM\_1** i otwórz ją do edycji.



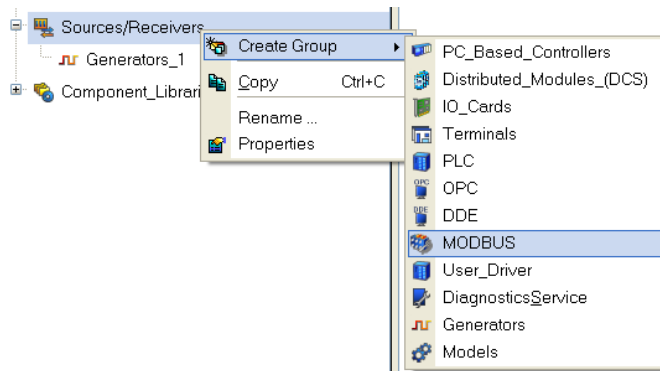
Ustaw odpowiednie parametry portu szeregowego.



### Dodawanie Sources/Receivers do wymiany danych przez protokół ModBus RTU

PLC obsługuje wymianę danych przez protokół Modbus RTU, więc dodaj grupę elementów **MODBUS\_1**, w powłoce **Sources/Receivers**, aby komunikować się z PLC.

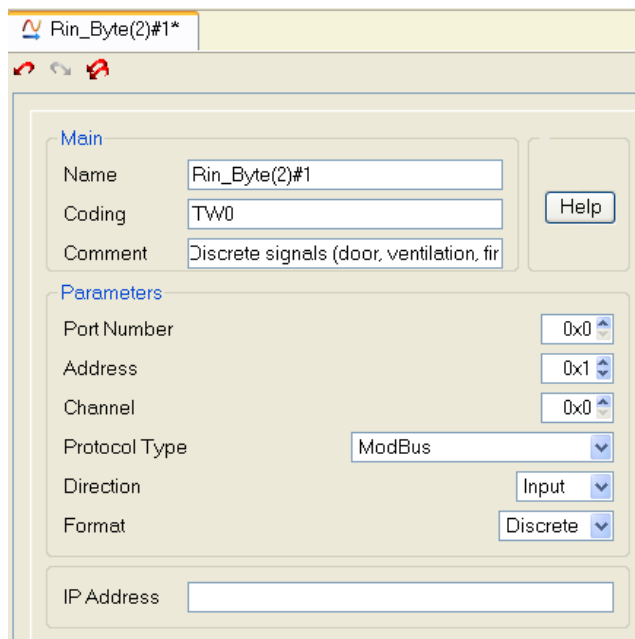




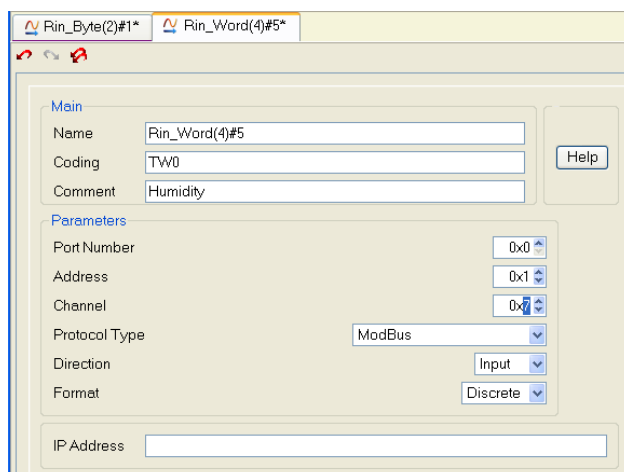
Dodaj elementy w grupie:

-  Rin\_Byte(2)#1
-  Rin\_Word(4)#2
-  Rin\_Word(4)#3
-  Rin\_Word(4)#4
-  Rin\_Word(4)#5

Edytuj elementy pamiętając, aby ustawić odpowiedni adres PLC oraz rejestru.

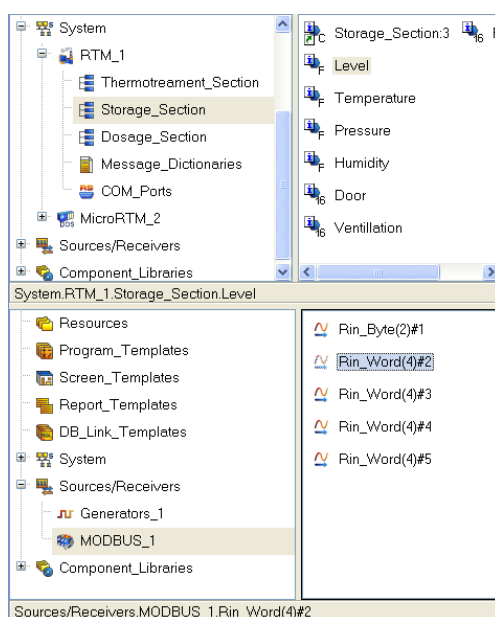


Ostatni parametr analogowy – **Humidity**:



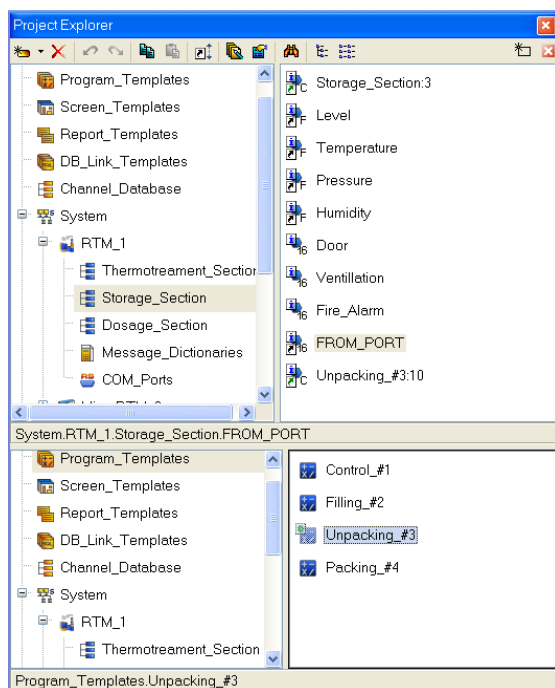
## Łączenie Sources/Receivers z kanałami

Po dodaniu elementów w powłoce Sources/Receivers należy połączyć je z atrybutami kanałów, zwykle z atrybutem **Input Value**. Wykonaj połączenia z kanałami parametrów analogowych grupy **Storage\_Section** w następujący sposób. Zdublikuj okno Project Explorer i wybierz grupę **Storage\_Section** w węźle **RTM\_1** w jednym oknie oraz grupę **MODBUS\_1** w drugim oknie. Zaznacz lewym przyciskiem myszy element **Rin Word(4)#2** i trzymając go przeciągnij na kanał **Level**.



Sprawdź poprawność połączenia. Połącz kanały **Temperature**, **Pressure** i


**Humidity** w ten sam sposób. Do odczytywania sygnałów binarnych należy dodać kanał **HEX16** o nazwie **FROM\_PORT** w grupie **Storage\_Section** i połączyć z elementem **Rin Byte(2)#1**. Po utworzeniu programu **Unpacking#3** do rozpakowywania bitów, otwórz powłokę **Program Templates** w dolnym oknie Project Explorer i przeciągnij szablon **Unpacking#3** do grupy **Storage\_Section**.



Do połączenia argumentów programu z atrybutami grupy kanałów **Storage\_Section** otwórz okno Properties kanału CALL **Unpacking#3** i kliknij dwukrotnie w polu **Link**, a następnie zaznacz atrybuty odpowiednich kanałów lub skorzystaj z metody **przeciągnij-upuść** – przeciągaj kanały do odpowiednich linii w zakładce **Arguments**.

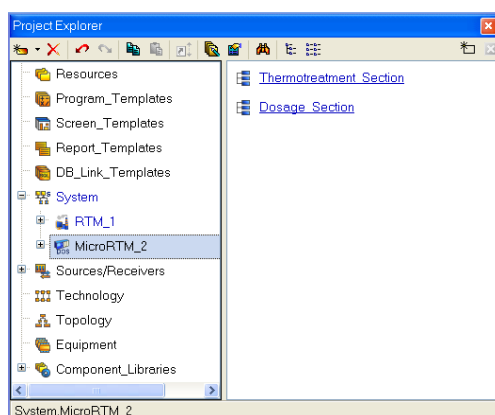
Name	IO Type	Data Type	Def. Value	Link
FROM_PORT	IN	USINT		FROM_PORT:Real Value(System.RTM_1.Storage_Section)
Door	OUT	USINT		Door:Input Value(System.RTM_1.Storage_Section)
Ventilation	OUT	USINT		Ventilation:Input Value(System.RTM_1.Storage_Section)
Fire_Alarm	OUT	USINT		Fire_Alarm:Input Value(System.RTM_1.Storage_Section)


Jeżeli w **Preferences** w zakładce **Arguments Editor** jest ustawiona flaga **Rename argument when link**, wówczas nazwy argumentów będą zmieniane na nazwy składające się z nazw kanałów połączonych z nimi, podkreślnika oraz skrótu atrybutu kanału.

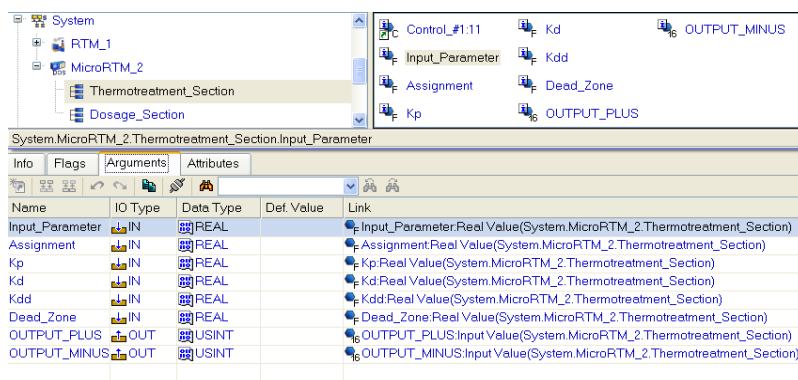
W ten sposób sygnały binarne odebrane z PLC będą przypisane do odpowiednich kanałów stacji roboczej przy użyciu programu. Zapisz projekt (  ).

## Dodawanie kanałów do kontrolera PAC

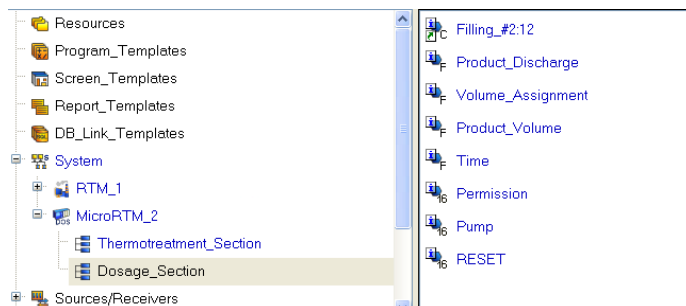
Dodawanie kanałów do kontrolera będzie bardzo podobne jak w przypadku węzła RTM (ten przypadek został opisany wcześniej). Otwórz w oknie Project Explorer węzeł kontrolera obsługującego sekcje Thermotreatment i Dosage. Utwórz w węźle **MicroRTM\_2** grupy kanałów **Thermotreatment\_Section** i **Dosage\_Section**.



Otwórz dodatkowe okno Project Explorer, a w nim powłokę **Program Templates**. Przeciągnij szablon programu **Control#1** do grupy **Thermotreatment\_Section** węzła **MicroRTM\_2**. W ten sposób zostanie utworzony kanał CALL programu **Control#1** w węźle. Wybierz zakładkę **Arguments** kanału **Control#1**, aby utworzyć odpowiednie kanały w grupie **Thermotreatment\_Section** i połączyć ich atrybuty z argumentami programu. Kliknij lewym przyciskiem myszy na ikonę , aby otrzymać następujący rezultat .



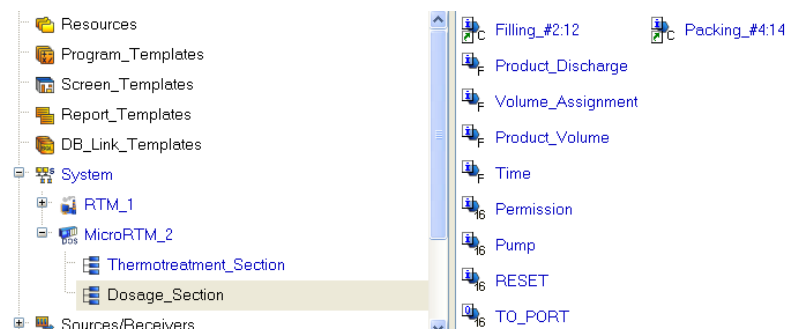
Przeciagnij szablon programu **Filling#2** do grupy **Dosage\_Section** (program **Filling#2** zapewnia dozowanie produktu zgodnie z wartościami zadanymi w stacji roboczej). Powtórz powyższe czynności. Efektem będzie następująca zawartość grupy **Dosage\_Section**:



Połączenia argumentów z atrybutami utworzonych kanałów powinny być następujące:

Name	IO Type	Data Type	Def. Value	Link
Product_Discharge	IN	REAL		Product_Discharge:Real Value(System.MicroRTM_2.Dosage_Section)
Volume_Assignment	IN	REAL		Volume_Assignment:Real Value(System.MicroRTM_2.Dosage_Section)
Product_Volume	OUT	REAL		Product_Volume:Input Value(System.MicroRTM_2.Dosage_Section)
Time	IN	REAL		Time:Real Value(System.MicroRTM_2.Dosage_Section)
Permission	IN/OUT	USINT		Permission:Real Value(System.MicroRTM_2.Dosage_Section)
Pump	OUT	USINT		Pump:Input Value(System.MicroRTM_2.Dosage_Section)
RESET	OUT	USINT		RESET:Input Value(System.MicroRTM_2.Dosage_Section)

Do pakowania sygnałów binarnych w bajty dodaj kanał **TO\_PORT HEX16** typu **OUTPUT** i przeciagnij go na szablon programu **Control#4**.

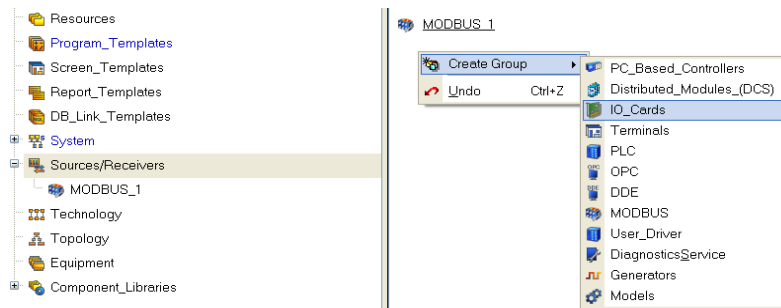


Połącz argumenty programu z atrybutami kanałów w zakładce **Arguments** okna Properties kanału **Control#4**.

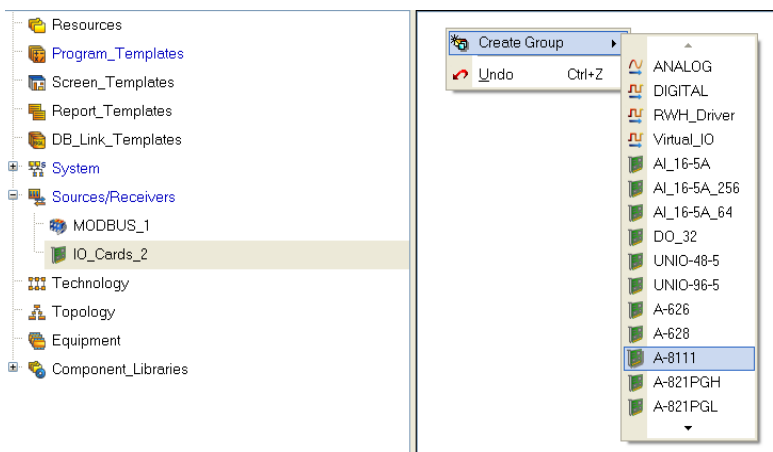
Name	IO Type	Data Type	Def. Value	Link
OUTPUT_PLUS_R	IN	USINT		OUTPUT_PLUS:Real Value(System.Mi)
OUTPUT_MINUS_R	IN	USINT		OUTPUT_MINUS:Real Value(System.l)
Pump_R	IN	USINT		Pump:Real Value(System.MicroRTM_2)
TO_PORT_In	OUT	USINT		TO_PORT:Input Value(System.MicroRT)

## Dodawanie Sources/Receivers do kontrolera PAC





Otwórz powłokę **Sources/Receivers** i zdefiniuj sprzęt kontrolera jako karta I/O typu **A-8111**. Dodaj element sources/receivers o nazwie **IO\_Cards\_2**.



Dodaj w nim podgrupę **A-8111#1**.



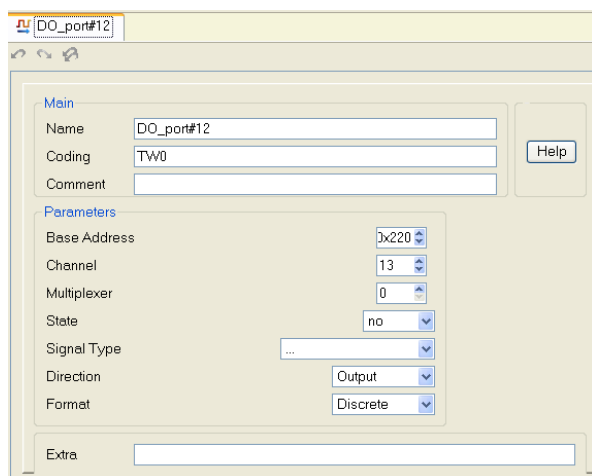
Otwórz do edycji nowo utworzoną grupę elementów karty **A-8111** i zostaw trzy wejścia analogowe i jedno wyjście binarne.

-  PCL\_channel#1
-  PCL\_channel#2
-  PCL\_channel#3
-  DO\_port#12

---

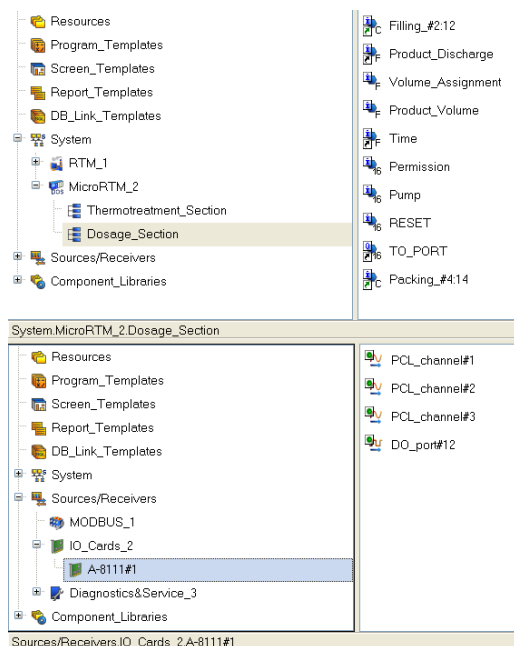
Domyślnie dla karty A-8111 tworzone są wejścia binarne, dlatego należy otworzyć ten komponent do edycji, zmienić jego parametry **State**, **Direction** oraz **Name**. Zmień piktogram elementu w oknie Properties.

---



## Łączenie Sources/Receivers z kanałami

Połącz elementy karty **A-8111** z kanałami węzła **MicroRTM\_2**: **PCL\_channel#1** z kanałem **Input\_Parameter**; **PCL\_channel#2** z kanałem **FLOAT Heat\_Carrier\_Consumption** (nowo utworzonym w grupie **Thermotreatment\_Section**), **PCL\_channel#3** z kanałem **Product\_Discharge** (grupa **Dosage\_Section**) oraz **DO\_port#12** z kanałem **TO\_PORT**.



W powłoce **Sources/Receivers** dodaj grupę **Diagnostics\_&\_Service\_3**, w niej podgrupę **Diagnostics\_1**, a w niej element **@Calc\_Loop\_1**. Będzie on sterował czasem przetwarzania kanałów węzła (w milisekundach). Połącz element z kanałem **Time** (grupa **Dosage\_Section** węzła **MicroRTM\_2**). Należy pamiętać, że aby zapewnić bardziej precyzyjne obliczanie ilości dozowanego produktu należy połączyć atrybut **Period Between Calls** kanału **CALL Filling#2** z argumentem **Time** tego samego kanału.

## Konfigurowanie kanałów, ustawianie początkowych warunków sterowania

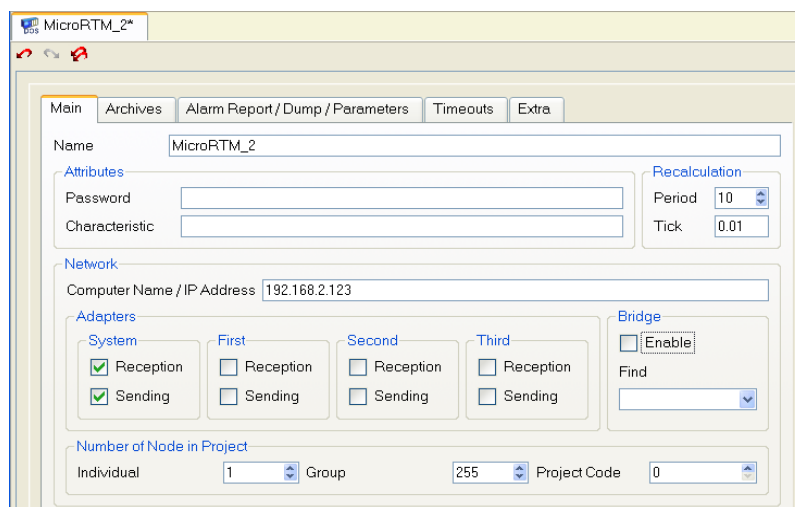
Ustaw wartości początkowe dla kanałów powiązanych ze sterowaniem urządzeniami w polu **On Start** oraz ustaw flagę **Assign at Start** w głównej zakładce edycji.

W ten sposób ustaw 1.000 dla kanału **Kp**, 0.1 dla kanału **Kd**, 0.01 dla kanału **Kdd**, 0.5 dla **Dead\_Zone**, 50 dla **Volume\_Setting**, 0 dla **Resolution**.

## Parametry pracy w sieci i konfiguracja charakterystyk dynamicznych węzła

Przed rozpoczęciem konfiguracji połączeń pomiędzy kanałami stacji roboczej i kontrolera PAC, otwórz węzeł **MicroRTM\_2** do edycji i ustaw kierunek wymiany danych w sieci, adres IP oraz Recalculation period (okres przetwarzania kanałów).



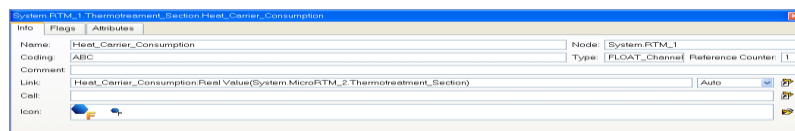


## Praca w sieci

Gdy wymiana danych pomiędzy węzłami jest skonfigurowana, należy określić w każdym przypadku źródło danych oraz odbiorcę danych, a także skonfigurować przepływ informacji.

Otwórz dwa okna Project Explorer; górne będzie wyświetlało węzeł stacji roboczej **RTM\_1**, a dolne będzie wyświetlało węzeł **MicroRTM\_2**.

Edytuj połączenia ręcznie przeciągając źródła danych na odbiorców przy pomocy metody **przeciągnij-upuść**, np. kanał **Heat\_Carrier\_Consumption** węzła **RTM\_1**, połączony z kanałem **Heat\_Carrier\_Discharge** węzła **MicroRTM\_1**, który z kolei otrzymuje dane z flow gauge, zainstalowanego w sekcji Dosage, wygląda to następująco w zakładce **Info** okna properties:



W tym szczególnym przypadku parametr **Auto** określa, że kanały zlokalizowane w różnych węzłach komunikują się przez sieć lokalną. Po połączeniu wszystkich kanałów pomiędzy węzłami stacji roboczej i sterownika otrzymamy następujący rezultat:

The image displays two screenshots of a software interface, likely a control system configuration tool, showing hierarchical tree views of system components.

**Top Screenshot:** The tree view shows the following structure:

- Resources
- Program\_Templates
- Screen\_Templates
- Report\_Templates
- DB\_Link\_Templates
- System
  - RTM\_1
    - Thermotreatment\_Section (Selected)
    - Storage\_Section
    - Dosage\_Section
    - Message\_Dictionaries
    - COM\_Ports
  - MicroRTM\_2
  - Sources/Receivers
  - Component\_Libraries

The right pane shows the following items:

- Thermotreatment\_Section:1
- PDD-Loop\_Parameters:2
- Heat\_Carrier\_Consumption
- Operation\_Temperature
- Temperature\_Setting
- Kp
- Kd
- Kdd
- Dead\_Zone

**Bottom Screenshot:** The tree view shows the following structure:

- Resources
- Program\_Templates
- Screen\_Templates
- Report\_Templates
- DB\_Link\_Templates
- System
  - RTM\_1
  - MicroRTM\_2
    - Thermotreatment\_Section (Selected)
    - Dosage\_Section
  - Sources/Receivers
  - Component\_Libraries

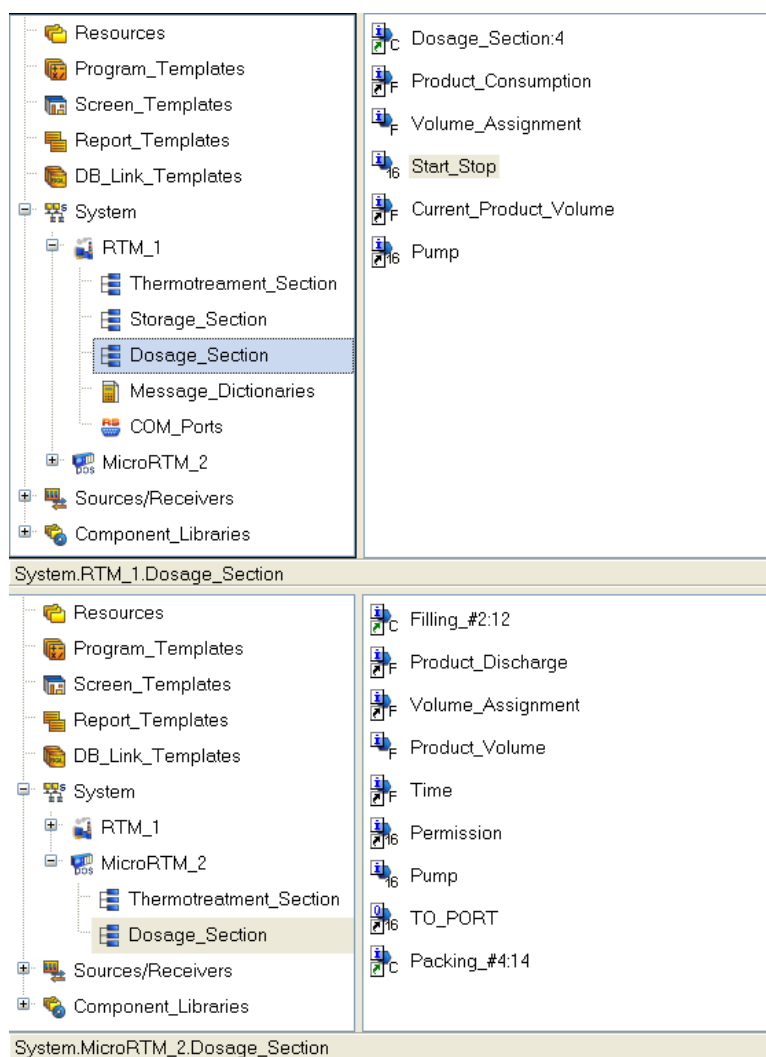
The right pane shows the following items:

- Control\_#1:11
- Input\_Parameter
- Assignment
- Kp
- Kd
- Kdd
- Dead\_Zone
- OUTPUT\_PLUS
- OUTPUT\_MINUS
- Heat\_Carrier\_Consumption

The status bars at the bottom of each screenshot indicate the selected path:

- Top: System.RTM\_1.Thermotreatment\_Section.PDD-Loop\_Parameters:2
- Bottom: System.MicroRTM\_2.Thermotreatment\_Section

oraz



## Konfiguracja wymiany danych w sieci

Ponieważ w projekcie zdefiniowane są tylko dwa węzły, a ilość danych, które będą wymieniały między sobą nie jest duża, sensowne jest zastosowanie metody **'point-to point'** do konfiguracji przepływu informacji. W celu zoptymalizowania ruchu w sieci ustaw flagę **Autosending** dla kanałów źródłowych w węzłach. Dzięki temu dane będą wysyłane ze źródła do odbiorcy tylko przy zmianie wartości kanału. Przez dopasowanie atrybutu **Aperture** dla kanałów źródłowych możesz także ustawić 'czułość' – aby odfiltrować fluktuacje wartości kanału, spowodowane np. szumami w torze pomiarowym.

Otwórz grupę **Thermotreatment\_Section** w węźle **MicroRTM\_2** i ustaw

flagę **To Network** w następujących kanałach:

Thermotreatment_Section			
Group Contents			
Name	To Network	Autosending ID	
Control_#1:11		0	not set
Input_Parameter	True		1
Assignment	False		not set
Kp	True		not set
Kd	0		not set
Kdd	0		not set
Dead_Zone	0		not set
OUTPUT_PLUS	0		not set
OUTPUT_MINUS	0		not set
Heat_Carrier_Consumption	1		2

Dla grupy **Dosage\_Section**:

Dosage_Section			
Group Contents			
Name	To Network	Autosending ID	
Filling_#2:12		0	not set
Product_Discharge	True		3
Volume_Assignment	False		not set
Product_Volume	True		4
Time	0		not set
Permission	0		not set
Pump	1		5
TO_PORT	0		not set
Packing_#4:14	0		not set

Wykonaj te same operacje dla grupy **Thermotreatment\_Section** (węzeł **RTM\_1**):

Thermotreatment_Section			
Group Contents			
Name	To Network	Autosending ID	
Thermotreatment_Section:1		0	not set
PDD-Loop_Parameters:2		0	not set
Heat_Carrier_Consumption		0	not set
Operation_Temperature		0	not set
Temperature_Setting		1	7
Kp	True		8
Kd	False		9
Kdd	True		10
Dead_Zone	1		11

i **Dosage\_Section**:

Thermotreatment_Section Dosage_Section		
Group Contents		
Name	To Network	Autosending ID
Dosage_Section:4	0	not set
Product_Consumption	0	not set
Volume_Assignment	True	12
Start_Stop	False	13
Current_Product_Volume	True	not set
Pump	0	not set

## Edycja kanałów


Kanał **RESET** utworzony przez procedurę autobuilding przez szablon programu **Filling#2** nie będzie wykorzystywany w sterowniku i może zostać usunięty, lecz w pierwszej kolejności edytuj kanał **Call Filling#2** w węźle **MicroRTM\_2** w następujący sposób:

Name	IO Type	Data Type	Def. Value	Link
Product_Discharge	IN	REAL		Product_Discharge:Real Value(System.MicroRTM_2.Dosage_Section)
Volume_Assignment	IN	REAL		Volume_Assignment:Real Value(System.MicroRTM_2.Dosage_Section)
Product_Volume	OUT	REAL		Product_Volume:Input Value(System.MicroRTM_2.Dosage_Section)
Time_CC	IN	REAL		Time:A time between calls(System.MicroRTM_2.Dosage_Section)
Permission	IN/OUT	USINT		Permission:Real Value(System.MicroRTM_2.Dosage_Section)
Pump	OUT	USINT		Pump:Input Value(System.MicroRTM_2.Dosage_Section)
RESET	OUT	USINT		Start_Stop:Real Value(System.RTM_1.Dosage_Section)

Edytuj także kanał **Call Dosage\_Section** (węzeł **RTM\_1**) w następujący sposób:

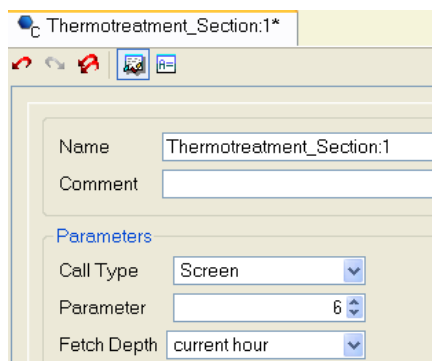
Name	IO Type	Data Type	Def. Value	Link
Product_Consumption	IN	REAL		Product_Consumption:Real Value(System.RTM_1.Dosage_Section)
Volume_Assignment	OUT	REAL		Volume_Assignment:Input Value(System.RTM_1.Dosage_Section)
Start_Stop	OUT	USINT		Start_Stop:Input Value(System.RTM_1.Dosage_Section)
Current_Product_Volume	IN	REAL		Current_Product_Volume:Real Value(System.RTM_1.Dosage_Section)
Pump	IN	USINT		Pump:Real Value(System.RTM_1.Dosage_Section)
Product_Volume_In	OUT	USINT		Product_Volume:Input Value(System.MicroRTM_2.Dosage_Section)

To w szczególności przedstawia sposób wykorzystania atrybutów kanałów zdalnego węzła jako argumentów szablonu. W celu uniknięcia tworzenia kanałów z argumentów szablonu, dostępne jest w edytorze argumentów pole **Flags**, w którym należy ustawić flagę **NP**.

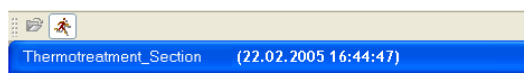
Kasowanie kanałów odbywa się przy użyciu klawisza **Del** lub ikony  z paska narzędzi.

## Wyświetlanie czasu w HMI

Jest kilka sposobów wyświetlania czasu na ekranie. Najprostszym z nich jest ustawienie atrybutu **Parameter** dla kanału **CALL**, który jest wykorzystywany do wywoływania ekranu graficznego.



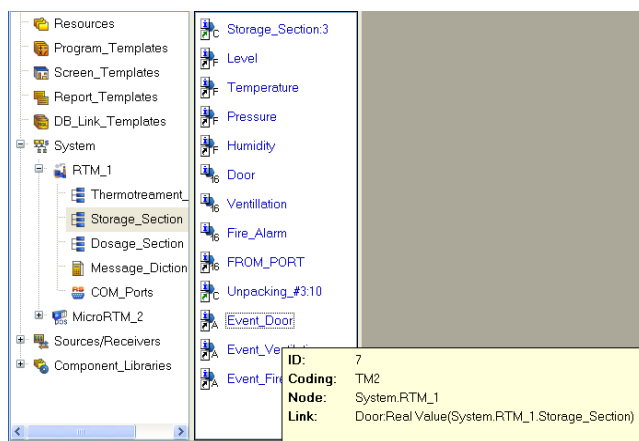
Jeżeli atrybut **Parameter** ma wartość **6** to następujący tekst zostanie wyświetlony w nagłówku okna:



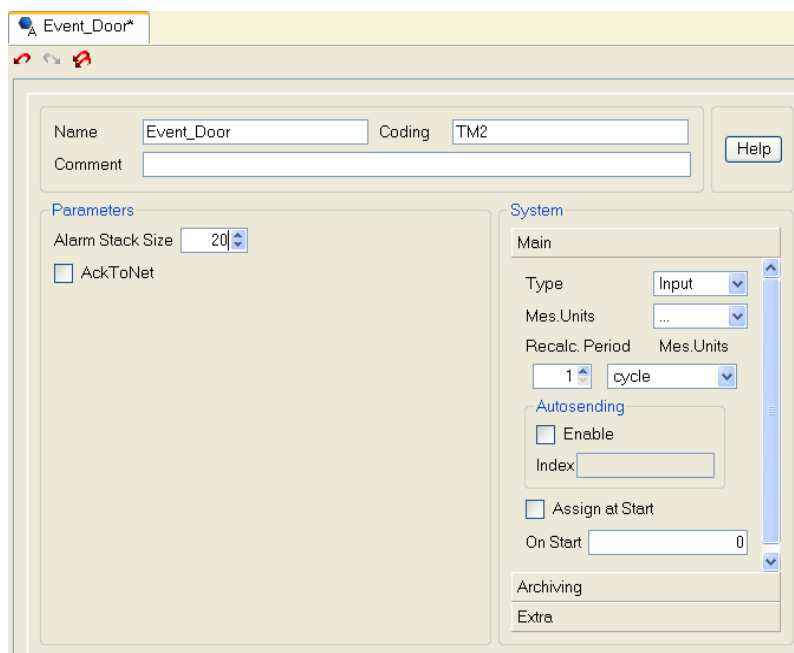
Innym sposobem jest dodanie zmiennych systemowych **@t\_Seconds**, **@t\_Minutes** i **@t\_Hours** w podgrupie elementów **System** grupy **Diagnositics&Service** w powłoce **Sources/Receivers** i połączenie ich z argumentami szablonu ekranu, które wyświetlają czas systemowy (godziny, minuty, sekundy) przy pomocy trzech elementów graficznych umieszczonych na ekranie.


## Zdarzenia

Do obsługi zdarzeń powiązanych ze zmianą sygnałów binarnych sekcji Storage (otwarte/zamknięte drzwi, stan wentylacji oraz systemów przeciwpożarowych), wyświetlania ich oraz zatwierdzania, dodaj trzy kanały **Event** w grupie **Storage\_Section** (węzeł stacji roboczej). Zmień ich nazwy, otwórz dodatkowe okno Project Explorer dla tej samej grupy i połącz nowo utworzone kanały z odpowiednimi kanałami **HEX16: Door, Ventilation, Fire Alarm**.





Otwórz do edycji kanał **Event\_Door** i ustaw wartość atrybutu **Alarm Stack Size** na 20. Powtórz czynność dla pozostałych dwóch kanałów.



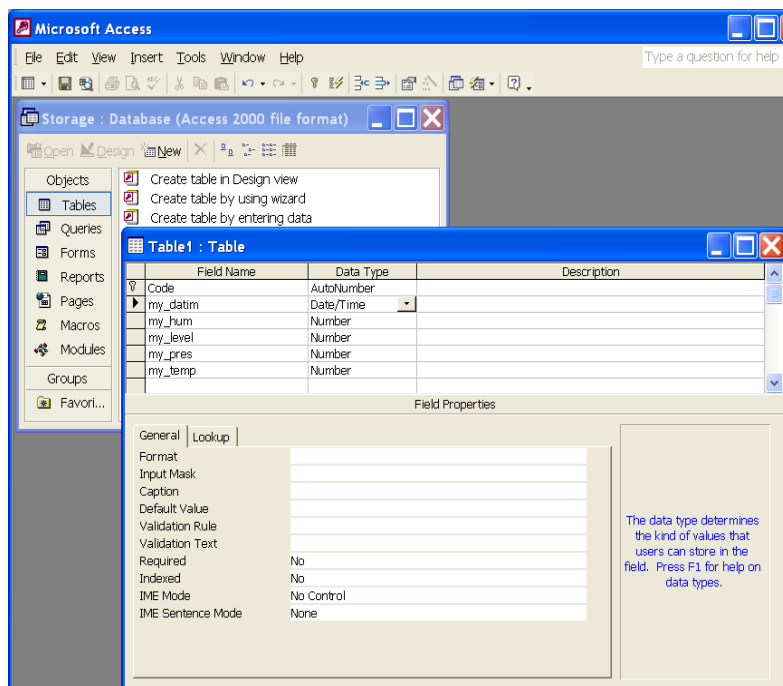
Ponieważ element graficzny Events () został dodany już wcześniej do szablonu ekranu **Storage\_Section**, połącz atrybuty dodanych kanałów **Event** z argumentami kanału **Call Storage\_Section**:

Name	IO Type	Data Type	Def. Value	Link	Flags
Level	IN	REAL		Level:Real Value(System.RTM_1.Storage_Section)	
Temperature	IN	REAL		Temperature:Real Value(System.RTM_1.Storage_Section)	
Pressure	IN	REAL		Pressure:Real Value(System.RTM_1.Storage_Section)	
Humidity	IN	REAL		Humidity:Real Value(System.RTM_1.Storage_Section)	
Door	IN	USINT		Door:Real Value(System.RTM_1.Storage_Section)	
Ventilation	IN	USINT		Ventilation:Real Value(System.RTM_1.Storage_Section)	
Fire_Alarm	IN	USINT		Fire_Alarm:Real Value(System.RTM_1.Storage_Section)	
Event_Door_R	IN/OUT	REAL		Event_Door:Real Value(System.RTM_1.Storage_Section)	NP
Event_Ventilation_R	IN/OUT	REAL		Event_Ventilation:Real Value(System.RTM_1.Storage_Section)	NP
Event_Fire_Alarm_R	IN/OUT	REAL		Event_Fire_Alarm:Real Value(System.RTM_1.Storage_Section)	NP
Date_Time	IN	DATE_AND_TIME			NP
Main_Layer	IN	USINT	0		NP
Trend_Layer	IN	USINT	1		NP

Ustaw wartości domyślne dla argumentów **Main\_Layer** i **Trend\_Layer** (odpowiednio **0** i **1**), dzięki czemu powłoka zawierająca GE Trend () nie będzie od razu wyświetlana. Zapisz projekt klikając na ikonę .

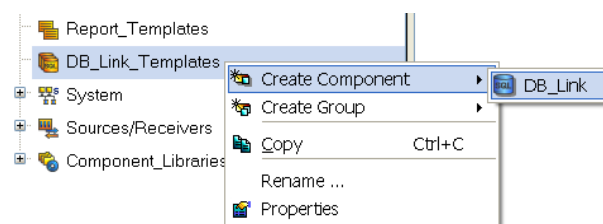
## Komunikacja z MS Access

Zgodnie ze specyfikacją przedstawioną wcześniej wymagane jest zapisywanie parametrów sekcji Storage, odebranych z PLC do relacyjnej bazy danych. Baza danych MS Access, **Storage.mdb**, została zaprojektowana do przechowywania parametrów technologicznych, odbieranych w czasie rzeczywistym ze stacji roboczej. Struktura tabeli **Storage1**, należącej do **Storage.mdb**, wygląda następująco:



Należy pamiętać, aby ustawić **kropkę** jako symbol dziesiętny. Można to zrobić w **panelu sterowania MS Windows – Opcje regionalne i językowe – zakładka Opcje regionalne**.

Dodaj teraz obsługę bazy danych w projekcie. Otwórz powłokę **DB-Link Templates** w oknie Project Explorer i dodaj nowy element – **Database #1**.

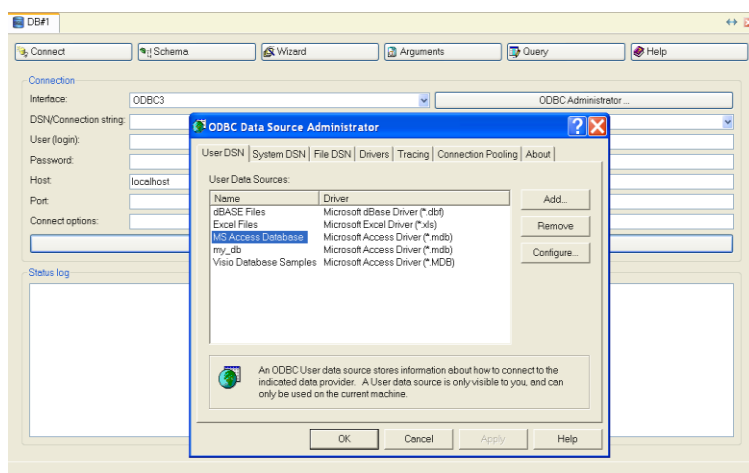




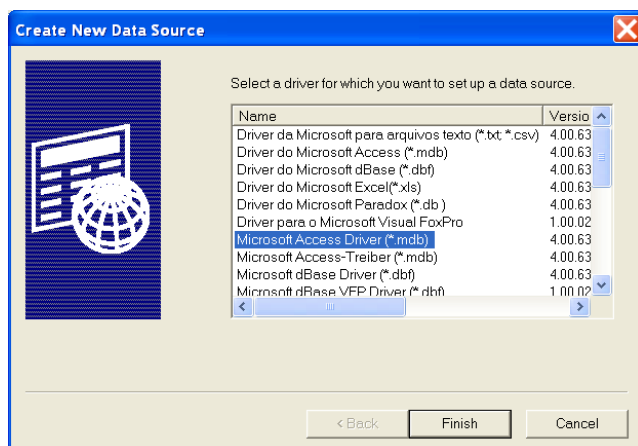
Otwórz okno Properties i dodaj argumenty:

Name	IO Type	Data Type	Def. Value	Link	Flags
Humidity	IN	REAL			
Level	IN	REAL			
Time_stamp	IN	REAL			
Pressure	IN	REAL			
Temperature	IN	REAL			

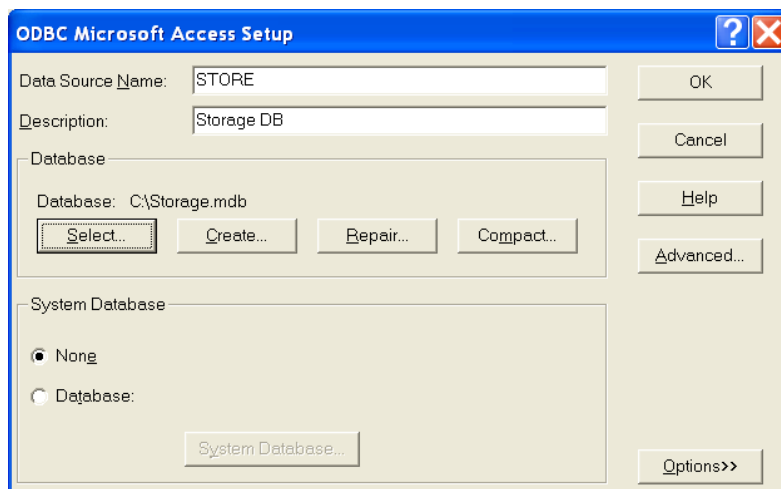
Zamknij okno Properties i kliknij dwukrotnie na szablon, aby przejść do edytora. Przed zapisywaniem danych z sekcji Storage do pliku C:\Storage.mdb, ustaw go jako plik danych Source/receiver. Kliknięcie lewym przyciskiem myszy na przycisk **ODBC Administrator** spowoduje wyświetlenie następującego okna:



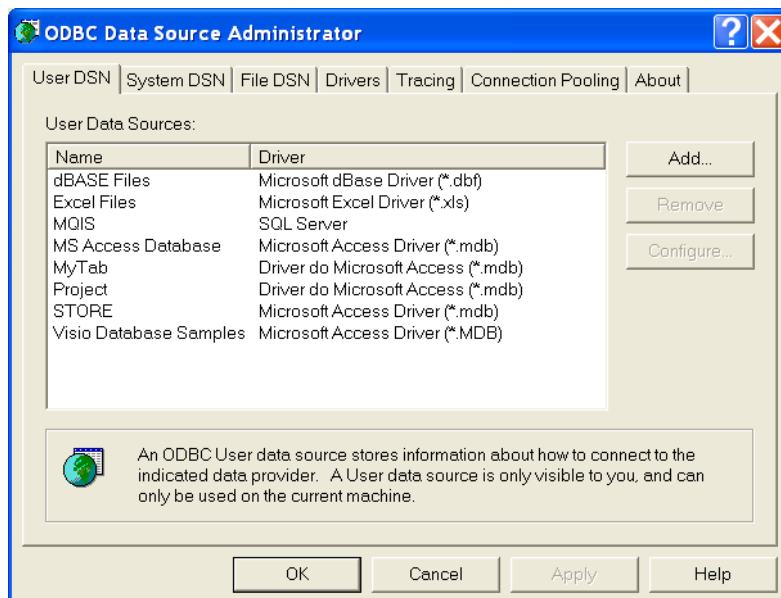
Przyciskiem **Add** wybierz nowe źródło danych - **STORE**.



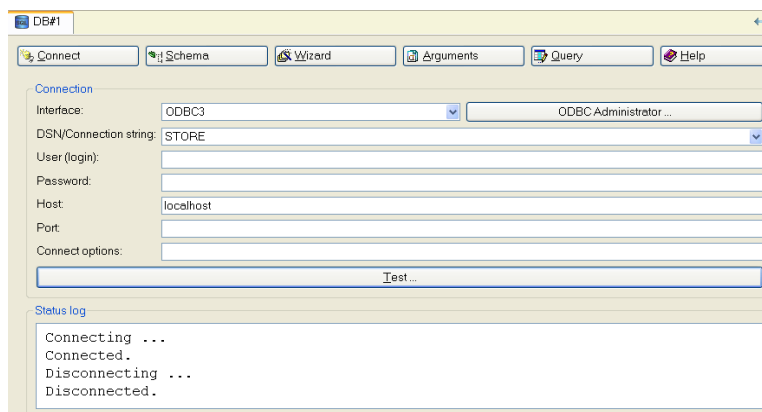
Skonfiguruj je w następujący sposób:



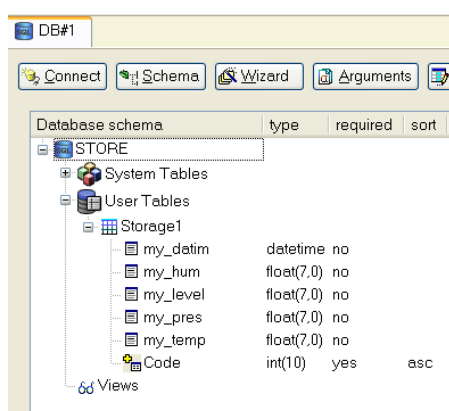
Kliknij **OK**. W ten sposób pojawi się nowy **DSN – STORE**.



Kliknij **OK** i kontynuuj pracę w edytorze szablonu **Data\_Base#1**. Określ utworzony **DSN** w **DSN/Connection String** (w naszym przypadku jest to **STORE**) i sprawdź (korzystając z przycisku **Test**) prawidłowość połączenia z bazą danych w oknie **Status Log**:



Przycisk **Schema** pozwala przejrzeć strukturę bazy danych:

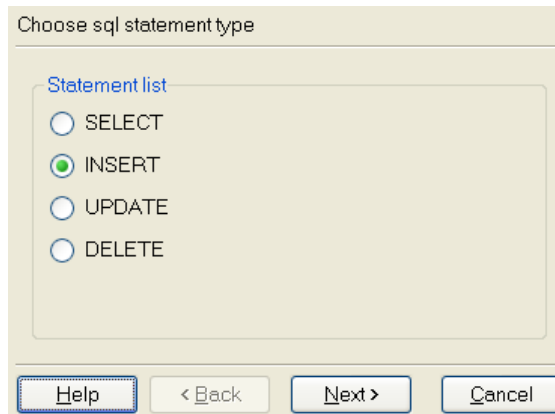


Następnie korzystając z **SQL Statement Wizard** lub bezpośrednio w zakładce **Query** utwórz zapytanie (w języku **SQL**), które dodaje jako rekordy atrybuty kanału, połączone z argumentami szablonu, do tabeli **Storage1** pliku **Storage.mdb**:

```
INSERT INTO Storage1
(
    my_datim,
    my_hum,
    my_level,
    my_pres,
    my_temp
)
VALUES
(
    '#Time_stamp#',
    '#Humidity#',
    '#Level#',
    '#Pressure#',
    '#Temperature#'
)
```

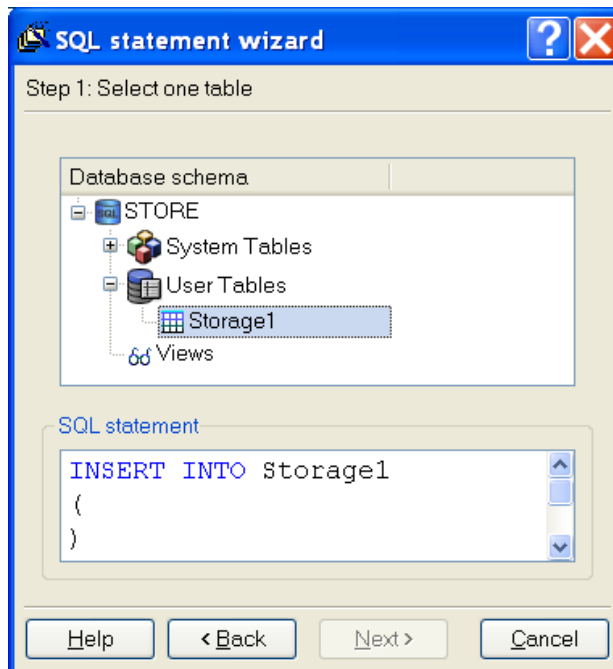
Kliknij lewym przyciskiem myszy na przycisk **Wizard**, aby korzystać z **SQL**

**Statement Wizard**, a następnie wybierz **INSERT**:



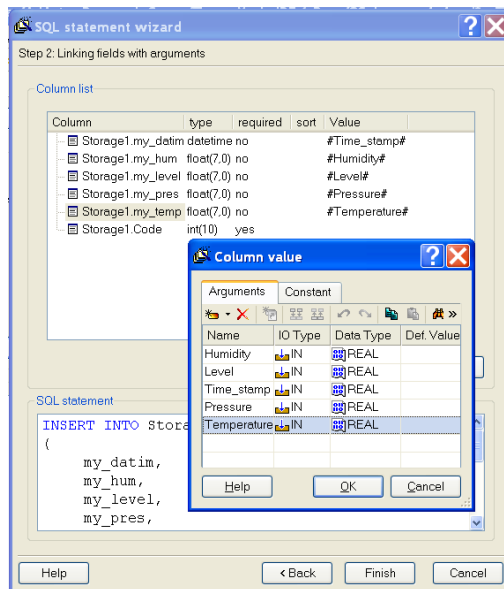
kliknij **Next**;

zaznacz tabelę **Storage1**;

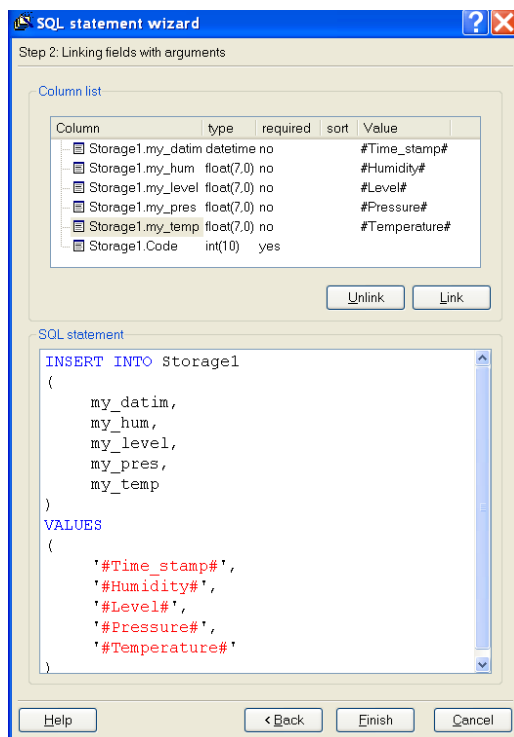


Kliknij **Next**, aby przejść do następnego kroku.


Połącz odpowiednie argumenty szablonu z polami bazy danych. Po wybraniu (podwójnym kliknięciem) pola bazy danych z listy i kliknięciu przycisku **Link**, argument, który ma być połączony jest określony w oknie pop-up **Column value**:

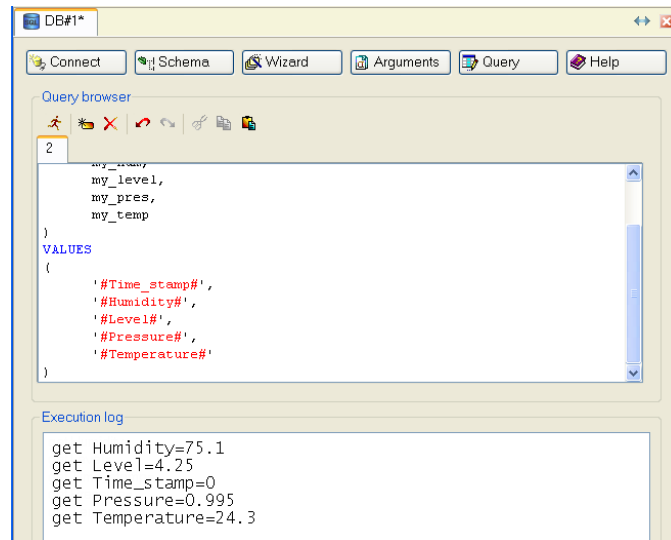


W ten sposób zapytanie zostanie wygenerowane w dolnym oknie **SQL Statement Wizard**:

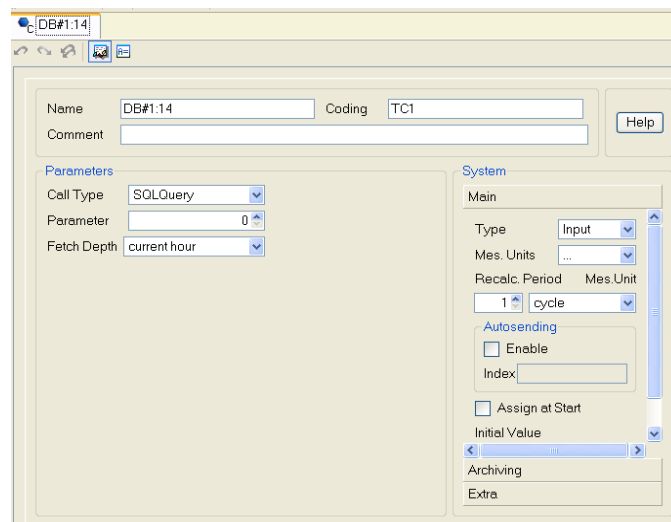


Zakończ tworzenie zapytania przyciskiem **Finish**.

Możesz sprawdzić zapytanie w zakładce **Query**. Debuggowanie odbywa się przez ustawianie wartości domyślnych w odpowiednich kolumnach tabeli edytora argumentów. Uruchom zapytanie w IDE klikając przycisk . Przedtem jednak ustaw domyślne wartości argumentów w odpowiedniej kolumnie edytora argumentów. Wykonywanie zapytania jest wyświetlane w oknie **Execution Log**:



Otwórz dodatkowe okno Project Explorer i zaznacz w nim grupę **Storage\_Section**. Zaznacz lewym przyciskiem myszy szablon **Database#1** i przeciągnij go do grupy **Storage\_Section**. Utworzony w niej zostanie kanał **Call Database#1**. Edytuj go klikając dwukrotnie:



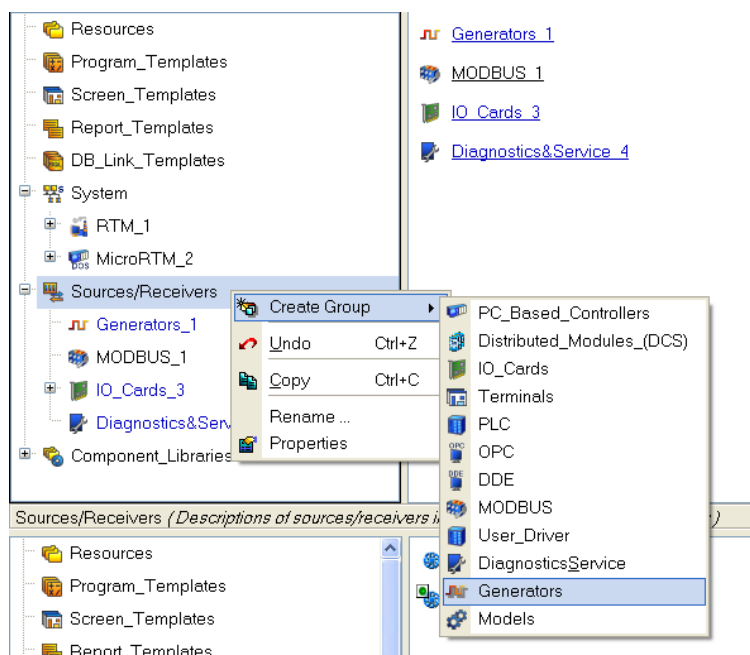
Połącz argumenty utworzonego zapytania z atrybutami kanałów: kliknij ikonę



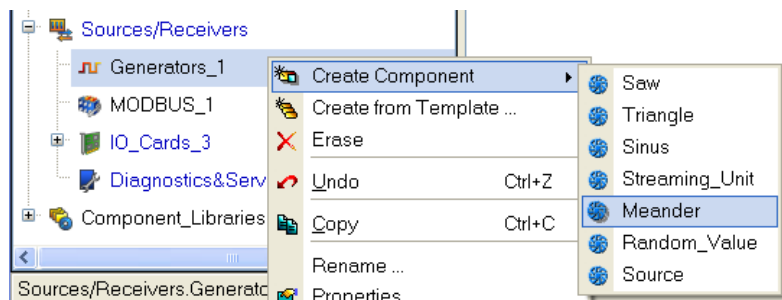
, aby otworzyć edytor argumentów i wykonaj następujące połączenia:

Name	IO Type	Data Type	Def. Value	Link	Flags	G
Humidity_R	IN	REAL		Humidity.Real Value(System.RTM_1.Storage_Section)		
Level_R	IN	REAL		Level.Real Value(System.RTM_1.Storage_Section)		
Meander_1_T	IN	REAL		Meander#1.Change Time(System.RTM_1.Storage_Section)		
Pressure_R	IN	REAL		Pressure.Real Value(System.RTM_1.Storage_Section)		
Temperature_R	IN	REAL		Temperature.Real Value(System.RTM_1.Storage_Section)		

Do wygenerowania daty i czasu wykorzystaj atrybut **Change Time** kanału **Meander#1**, utworzonego w grupie **Storage\_Section** przez procedurę autobuilding. Bazuje on na elemencie generator w powłoce **Sources/Receivers**:

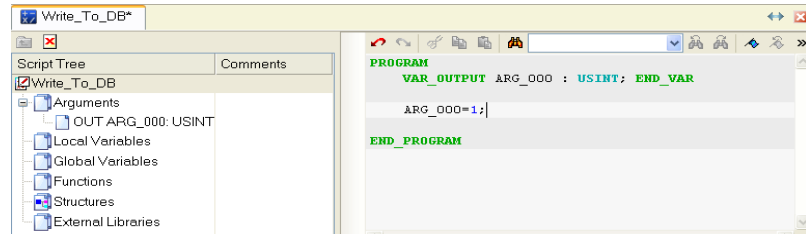


gdzie

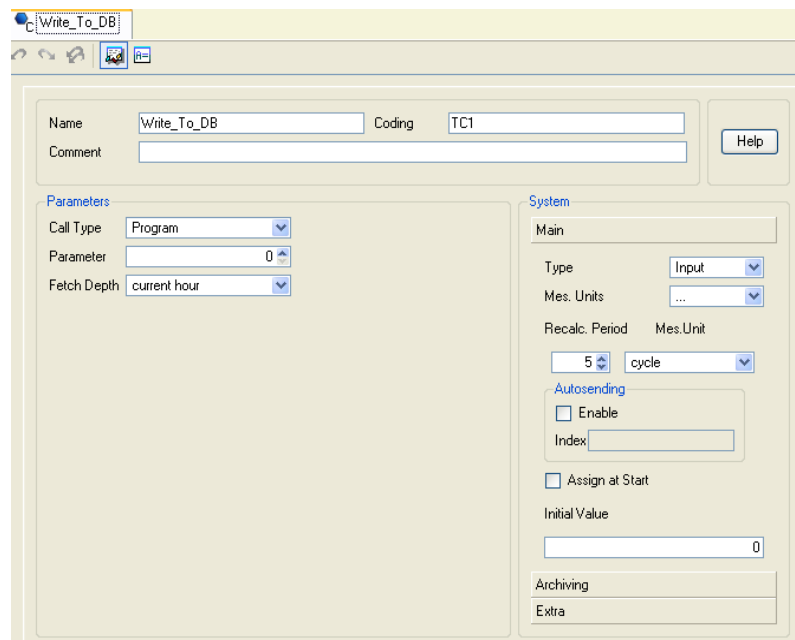



Przeciagnij utworzony element do grupy **Storage\_Section**. Z racji tego, że wartość kanału **Meander#1** zmienia się przy każdym cyklu przetwarzania kanału, aktualna wartość daty i czasu będzie zapisywana do pliku DB.

Zapisywanie danych sekcji Storage do pliku **Storage.mdb** w trakcie pracy stacji roboczej RTM, będzie się odbywało po wysłaniu numeru zapytania (w tym przypadku jest to **1**) do input value kanału **Call**. Można to zrobić przy użyciu programu lub GE **Button**. Dodaj więc nowy program, **Write\_To\_DB** w powłoce **Program Templates**:

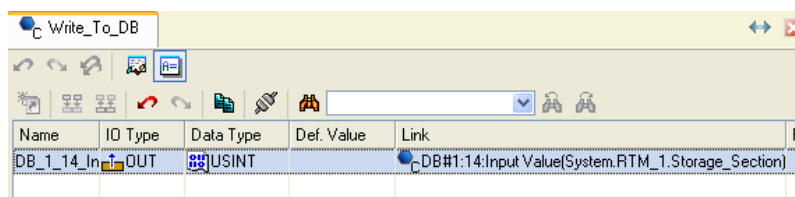


Skompiluj program i przeciągnij go do grupy **Storage\_Section** węzła **RTM\_1**. Otwórz kanał **Call Write\_To DB** do edycji i ustaw w jego zakładce Main wartość atrybutu **Period** równą **5 cycle**.




Kliknij zakładkę , aby przejść do argumentów kanału i ustawić następujące połączenia:



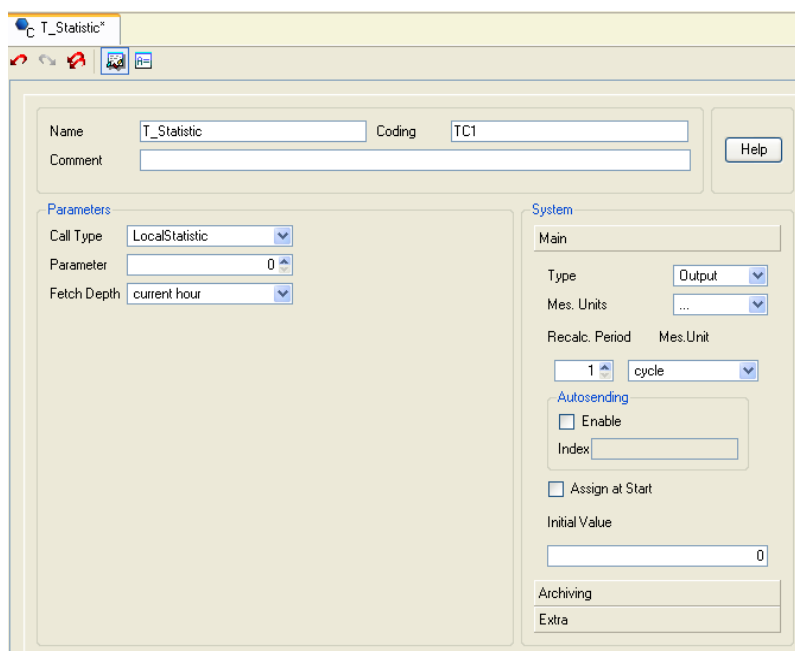


Name	IO Type	Data Type	Def. Value	Link
DB_1_14_In	OUT	USINT		DB#1:14:Input Value(System.RTM_1.Storage_Section)

Zmieniając okres przetwarzania kanału możesz regulować częstotliwość zapisu parametrów sekcji Storage do pliku bazy danych MS Access. Zapisz projekt klikając na ikonę .

## Przetwarzanie danych z lokalnego archiwum

Odkąd wszystkie parametry technologiczne są zapisywane w bazie danych **SIAD/SQL 6**, do uzyskania, np. charakterystyk statystycznych takiego parametru jak **Operational Temperature** (sekcja Thermotreatment) możesz wykorzystać procedurę przetwarzania danych zapisanych na dysku. Dodaj w grupie **Thermotreatment\_Section** węzła **RTM\_1** kanał **Call** i kliknij go dwukrotnie, aby przejść do jego edycji. Zmień atrybuty – **Name**, **Type** i **Call Type**:



Name: T\_Statistic    Coding: TC1    Help

Comment: \_\_\_\_\_

**Parameters**

Call Type: LocalStatistic

Parameter: 0

Fetch Depth: current hour

**System**

Main

Type: Output

Mes. Units: ...

Recalc. Period: 1    Mes.Unit: cycle

Autosending

Enable

Index: \_\_\_\_\_


Assign at Start

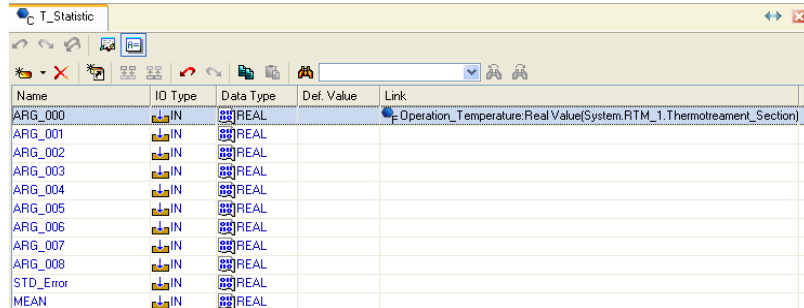
Initial Value: 0

Archiving

Extra



Wybrany typ call **LocalStatistic** został zaprojektowany do statystycznego przetwarzania danych z lokalnego archiwum. Ponieważ proces obróbki cieplnej trwa kilka godzin, **Fetch Depth** ma domyślną wartość current hour, która pozwala oszacować zmiany temperatury w czasie bieżącej godziny. Ustaw typ

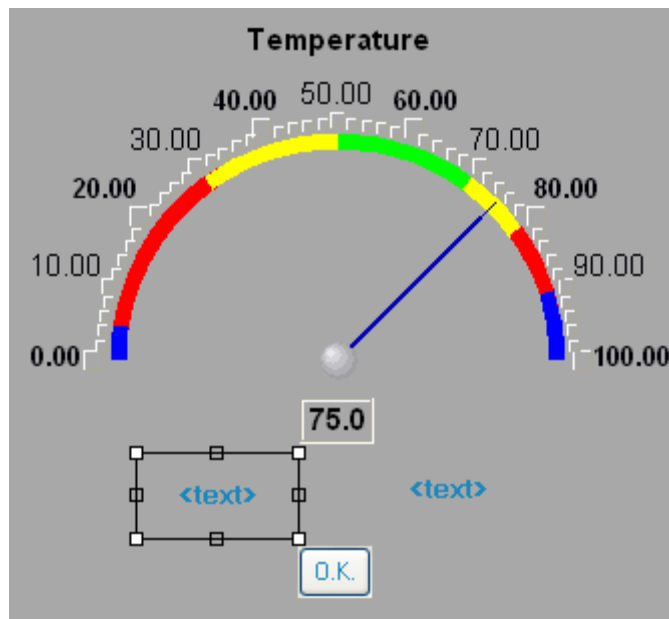
**OUTPUT** dla kanału **T\_Statistic**. Kliknij na ikonę , aby przejść do tabeli edytora argumentów. Dodaj 12 argumentów, zmień nazwę dwóch ostatnich i połącz argument **ARG\_000** z kanałem **Operational\_Temperature**.



Name	IO Type	Data Type	Def. Value	Link
ARG_000	IN	REAL		Operation_Temperature:Real Value(System.RTM_1.Thermotreatment_Section)
ARG_001	IN	REAL		
ARG_002	IN	REAL		
ARG_003	IN	REAL		
ARG_004	IN	REAL		
ARG_005	IN	REAL		
ARG_006	IN	REAL		
ARG_007	IN	REAL		
ARG_008	IN	REAL		
STD_Error	IN	REAL		
MEAN	IN	REAL		

Odchylenie standardowe będzie zapisywane do argumentu **STD\_Error**, a wartość średnia kanału określonego przez **ARG\_000** będzie zapisywana do argumentu **MEAN**.

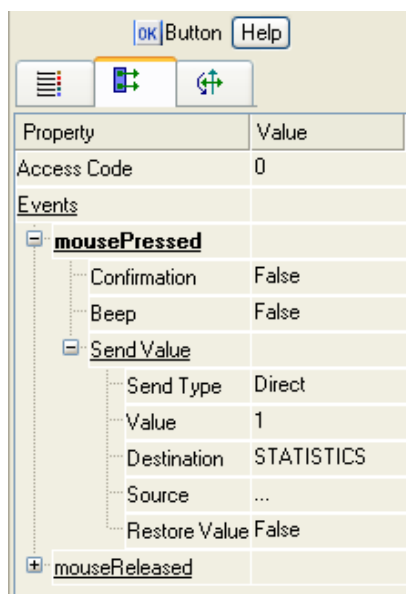
Wyświetlanie wartości średniej i odchylenia standardowego na ekranie: otwórz w powłoce **Screen Templates** szablon **Thermotreatment\_Section** i umieść w nim trzy elementy graficzne **Text** () oraz jeden element **Button** () , który będzie uruchamiał przetwarzanie kanału **T\_Statistic**.



Wyłącz obramowanie i wypełnienie dla elementów **Text**. Dodaj trzy nowe argumenty do szablonu ekranu i połącz je z elementami graficznymi:

Name	IO Type	Data Type	Def. Value
Heat_Carrier_Consumption	↓ IN	REAL	
Operation_Temperature	↓ IN	REAL	
Temperature_Setting	↑ OUT	REAL	
STD_Err	↓ IN	REAL	
Mean	↓ IN	REAL	
STATISTICS	↓↑ IN/OUT	REAL	

Dla GE Button ustaw bezpośrednie wysyłanie wartości 1 po kliknięciu lewym przyciskiem myszy:



Otwórz okno Properties kanału **Call Thermotreatment\_Section** węzła **RTM\_1** i połącz nowo dodane argumenty:

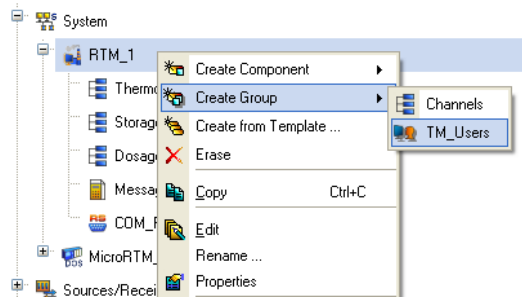
Name	IO Type	Data Type	Def. Value	Link
Heat_Carrier_Consumption	↓ IN	REAL		Heat_Carrier_Consumption:Real Value(System.RTM_1.Thermotreatment_Section)
Operation_Temperature	↓ IN	REAL		Operation_Temperature:Real Value(System.RTM_1.Thermotreatment_Section)
Temperature_Setting	↑ OUT	REAL		Temperature_Setting:Input Value(System.RTM_1.Thermotreatment_Section)
STD_Error	↓ IN	REAL		T_Statistic:STD_Error(System.RTM_1.Thermotreatment_Section)
MEAN	↓ IN	REAL		T_Statistic:MEAN(System.RTM_1.Thermotreatment_Section)
STATISTICS	↓↑ IN/OUT	REAL		T_Statistic:Input Value(System.RTM_1.Thermotreatment_Section)

Argumenty **STD\_Err** i **Mean** szablonu ekranu będą połączone z odpowiednimi argumentami kanału **Call T\_Statistic**, podczas gdy argument **STATISTICS** będzie połączony z argumentem **Input Value** tego samego kanału. W ten sposób, po kliknięciu lewym przyciskiem myszy na elemencie Button, zostanie wykonane przetwarzanie statystyczne wartości kanału **Operational\_Temperature** zgromadzonych w archiwum w trakcie bieżącej godziny, a odchylenie standardowe oraz wartość średnia zostaną wyświetlone na ekranie.

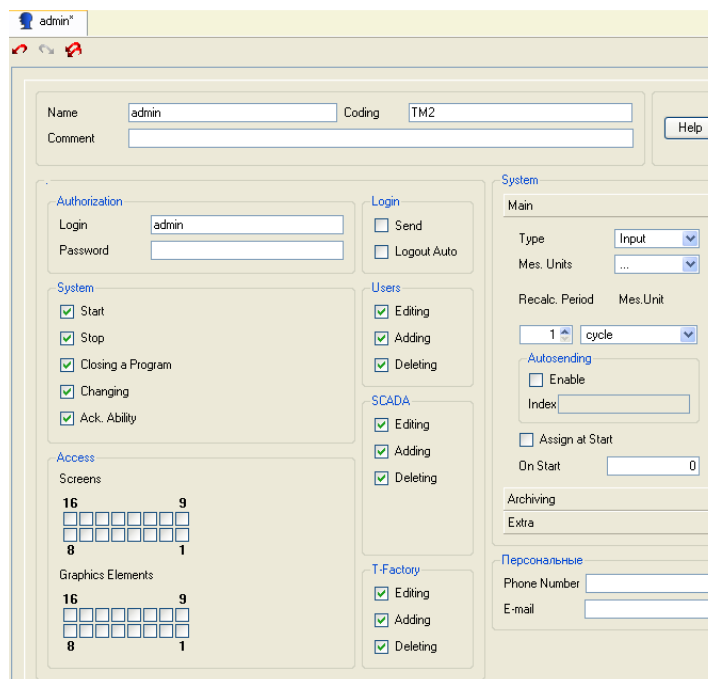
Kliknij na ikonę , aby zapisać projekt.

## Dodawanie użytkowników

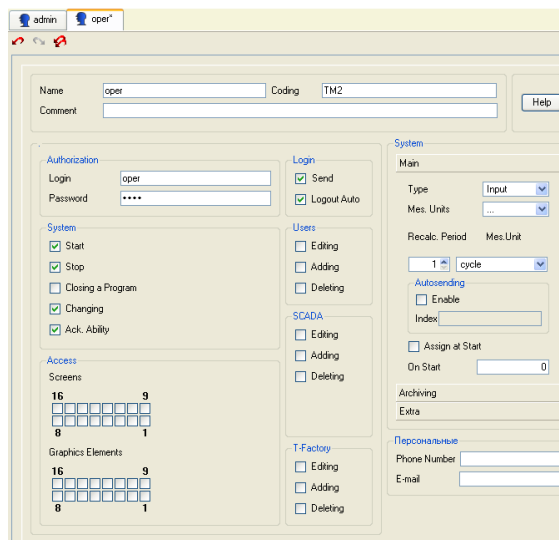
Zabezpiecz system sterowania przed nieautoryzowanym dostępem określając użytkowników tego systemu, nadając im prawa dostępu oraz loginy i hasła. W tym celu dodaj w węźle RTM nową grupę, **TM\_Users**.





Otwórz ją i dodaj dwa nowe komponenty – **TM\_User**. Otwórz kolejno każdy z nich do edycji, aby zmienić nazwę oraz ustawić login i hasło. Przydziel maksymalne prawa użytkownikowi **admin** (projektant), włącznie z prawami do dokonywania zmian, natomiast użytkownikowi **oper** przydziel ograniczone prawa (tylko do pracy z systemem). Dla użytkownika **admin** edytor powinien wyglądać następująco:



Dla użytkownika **oper**:



Zapisz projekt z nazwą **QS\_Lesson\_2.prj** (korzystając z ikony ) , a następnie kliknij lewym przyciskiem myszy na ikonę  , aby wygenerować pliki, które zostaną przegrane na docelowe platformy do debuggowania projektu.

## Symulatory

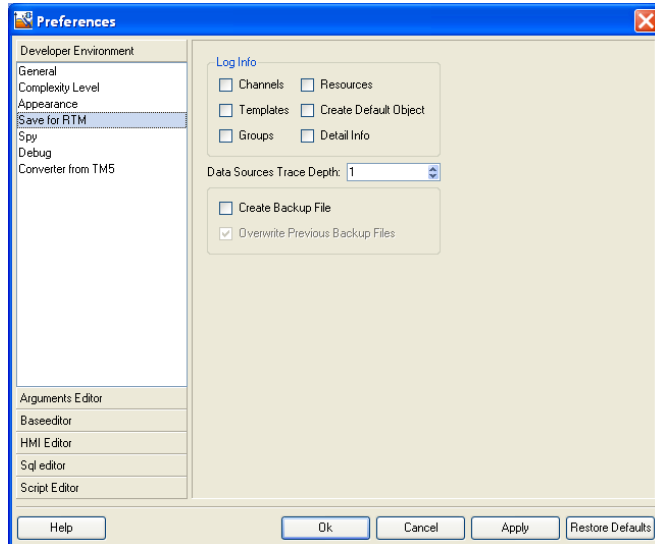
Przykłady analizowane w pierwszym rozdziale wykorzystywały konkretne urządzenia, jednak nie każdy użytkownik je posiada. Do sprawdzenia działania projektu zastosuj symulatory zarówno do obiektów technologicznych sekcji Thermotreatment Section jak i sygnałów wysyłanych z PLC sekcji Storage.

## Przygotowania

W pierwszej kolejności wykonaj pewne czynności przygotowawcze. Otwórz projekt **QS\_Lesson\_2.prj**, otwórz powłokę **Sources/Receivers** i skasuj grupy **MODBUS\_1** oraz **IO\_Cards\_2** przy użyciu klawisza **Del**. W ten sposób kasujesz połączenia kanałów ze sprzętem. W grupie **Storage\_Section** węzła **RTM\_1** zaznacz kanał **Call Unpacking#3** i otwórz jego okno Properties. Wybierz zakładkę **Arguments** i skasuj połączenia argumentów z atrybutami kanałów w następujący sposób:

Name	ID	Type	Data Type	Def. Value	Link
FROM_PORT	IN		USINT		FROM_PORT.Real Value[System.RTM_1.Storage_Section]
Door	OUT		USINT		
Ventilation	OUT		USINT		
Fire_Alarm	OUT		USINT		

W praktyce takiego podejścia się nie stosuje. Gdy potrzebne są sygnały do debugowania projektu można zastąpić te pochodzące ze sprzętu generatorami sygnałów lub innym sprzętem. W takich przypadkach najbardziej efektywnym rozwiązaniem jest zastosowanie podwójnego połączenia ze źródłem sygnału, przez ustawienie **Data Sources Trace Depth** w głównych ustawieniach IDE, przed wykonaniem procedury Saving For RTM:



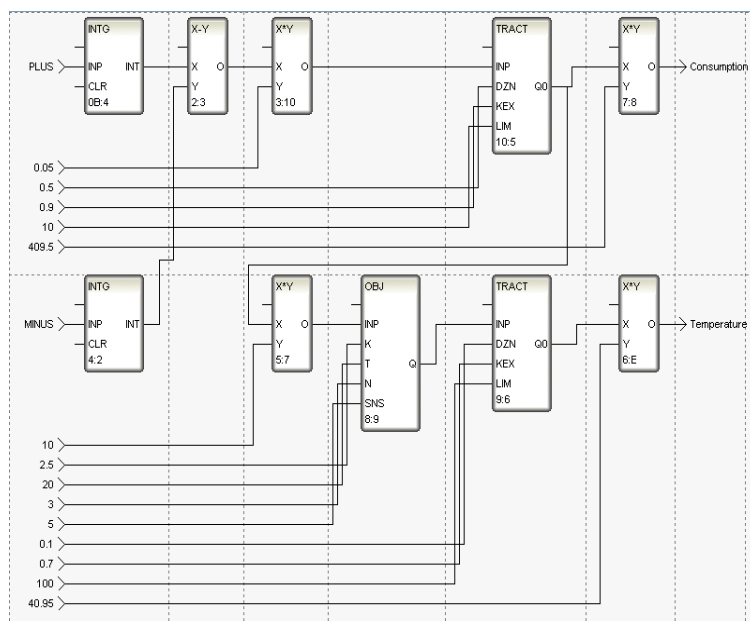
Do zamiany np. źródła **PCL\_channel#1** – wejście ADC na karcie I/O **A-8111** na generator sinusoidy wystarczy dodać źródło **Sinus#1** w grupie **Generators** w powłoce **Sources/Receivers** i przy użyciu metody **przeciągnij-upuść** połączyć z istniejącym źródłem **PCL\_channel#1**. Przez ustawienie **Data Sources Trace Depth** w głównych ustawieniach IDE na wartość równą **2** stosujesz zamianę źródeł. Zmiana ustawienia Data Source Trace Depth możliwa jest także w czasie pracy.

## Tworzenie programów symulujących i dodawanie ich do projektu

Dodaj symulator obiektu w sekcji Thermotreatment. Dodaj program **Thermotreatment\_Simulator** w powłoce **Program Templates** i ustaw w nim następujące argumenty:

Name	IO Type	Data Type	Def. Value	Link	Flags	Group	Measurement Unit
PLUS	IN	USINT					...
MINUS	IN	USINT					...
Consumption	OUT	REAL					...
Temperature	OUT	REAL					...

Argumenty wejściowe **PLUS** i **MINUS** są sygnałami z pętli PDD z PWM, podczas gdy argumenty wyjściowe **CONSUMPTION** i **TEMPERATURE** przedstawiają zużycie ciepła i temperaturę w urządzeniu. Stwórz matematyczny model urządzenia zakładając, że jego działanie jest opisane przez element inercyjny pierwszego stopnia z opóźnieniem. Ustaw następujące parametry obiektu: wzmacnienie 2.5, stała czasowa 20 (w cyklach przetwarzania), opóźnienie 3 (w cyklach przetwarzania). Sygnał wyjściowy zostanie dodatkowo zaszumiony. Jako język programowania wykorzystaj **Techno FBD**.



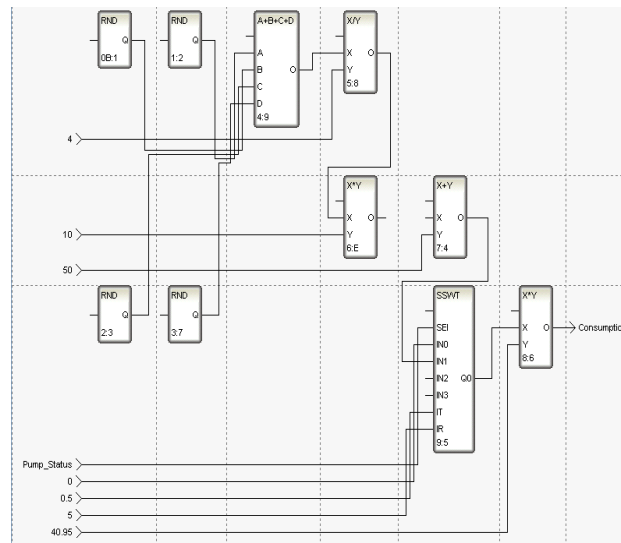
Bloki funkcyjne **TRACT** są wykorzystane do wygładzania i ograniczania generowanych parametrów wyjściowych, podczas gdy bloki funkcyjne mnożenia **X\*Y 8 i 9** są wykorzystywane do dopasowania wartości wyjściowych do zakresu 12 bitowego ADC. Skompiluj utworzony program. Otwórz dodatkowe okno Project Explorer, a w nim grupę **Thermotreatment\_Section** węzła **MicroRTM\_2** i przeciągnij do niej szablony programu. W utworzonym kanale **Thermotreatment\_Simulator** połącz atrybuty kanału z argumentami.

Name	IO Type	Data Type	Def. Value	Link
OUTPUT_PLUS	IN	USINT		OUTPUT_PLUS:Real Value(System.MicroRTM_2.Thermotreatment_section)
OUTPUT_MINUS	IN	USINT		OUTPUT_MINUS:Input Value(System.MicroRTM_2.Thermotreatment_section)
Consumtion	OUT	REAL		Heat_Carrier_Consumtion:Input Value(System.MicroRTM_2.Thermotreatment_section)
Temperature_In	OUT	REAL		Input_Parameter:Input Value(System.MicroRTM_2.Thermotreatment_section)

Do symulacji zużycia produktu dodaj szablon programu **Product\_Discharge\_Simulator** i wykorzystaj w nim język programowania **Techno FBD**. Argumenty programu powinny być określone w następujący sposób:

Info		Arguments	
Name	IO Type	Data Type	Def. Value
Pump_Status	IN	USINT	
Consumption	OUT	REAL	

Ustaw wskaźnik zużycia na **50** i zastosuj bloki funkcyjne przełączania z dynamicznym bilansowaniem, **SSWT**, dla złagodzenia zmian w momencie załączenia/wyłączenia pompy dozującej. Nie zapomnij o dostosowaniu wartości wyjściowej do zakresu 12-bitowego ADC, przy użyciu bloku **X\*Y**.



Skompiluj program. Zaznacz **Dosage\_Section** (węzeł **MicroRTM\_2**) w dodatkowym oknie Project Explorer i przeciągnij do niego szablon programu. W utworzonym kanale **Call Product\_Discharge\_Simulator** połącz atrybuty kanału z argumentami.

Info		Flags		Arguments		Attributes	
Name	IO Type	Data Type	Def. Value	Link			
Pump_Status	IN	USINT		Pump.Real Value[System.MicroRTM_2.Dosage_Section]			
CONSUMPTION	OUT	REAL		Product_Consumtion:Input Value[System.MicroRTM_2.Dosage_Section]			

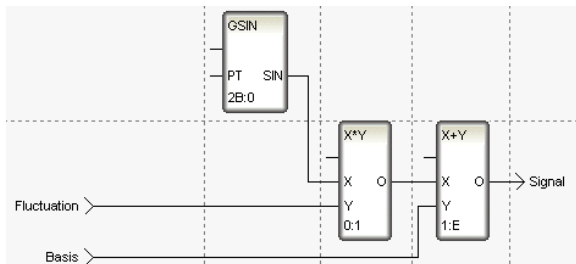
Ponieważ ze sterownika S7-200 odczytywane są zarówno sygnały analogowe jak i binarne, należy dodać dwa odpowiednie programy do symulacji danych w stacji roboczej. Ustawianie parametrów będzie odbywało się przy pomocy stałych, znajdujących się w polu **Default Value** tabeli edytora argumentów.

Dodaj nowy program, **YX\_Analogue\_Simulator** w powłoce **Program Templates**. Ustaw jego argumenty:



Name	IO Type	Data Type	Def. Value
Signal	OUT	REAL	
Fluctuation	IN	REAL	
Basis	IN	REAL	

Jako język programowania wybierz **Techno FBD**.



Przeskaluj element 'losowy' zmieniający się sinusoidalnie w zakresie (-1,1), ustawiając argument **Fluctuation** oraz dodając parametr **Basis**, który określa wartość stałą. Skompiluj program przyciskiem **F7**.

Zaznacz grupę **Storage\_Section** w węzle **RTM\_1**, aby połączyć utworzony program. Zaznacz kanał **Level** i otwórz jego okno Properties. W zakładce **Info** zdefiniuj kanał jako **Call** przez zaznaczenie w polu Call szablonu programu **Simulator\_YX\_Analogue**.

Info	Flags	Arguments	Attributes
Name:	Level	Node:	System.RTM_1
Coding:	TC5	Type:	FLOAT_Channel Reference Counter: 1
Comment:			
Link:			
Call:	Profiler_Analog_SS(Program_Templates)		
Icon:			

Przejdź do zakładki **Arguments**, połącz argumenty programu z atrybutami kanału i ustaw wartości stałych w kolumnie **Default Values**.

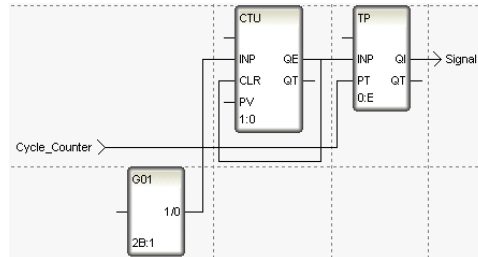
Name	IO Type	Data Type	Def. Value	Link
Level_R	OUT	REAL		Level.Real Value(System.RTM_1.Storage_Section)
Fluctuation	IN	REAL	0.25	
Basis	IN	REAL	3.5	

Wykonaj te same operacje dla kanałów **Temperature**, **Pressure** i **Humidity** oraz ustaw parametry **Fluctuation** i **Basis** równe odpowiednio (0.25,24), (0.025,0.98) i (0.2,70).

Do symulacji sygnałów binarnych utwórz szablon programu **Simulator\_YX\_Digital** z następującymi argumentami:

Info Arguments				
Name	ID Type	Data Type	Def. Value	Link
Signal	OUT	USINT		
Cycle_Counter	IN	REAL		

Program w języku **Techno FBD** będzie wyglądał następująco:



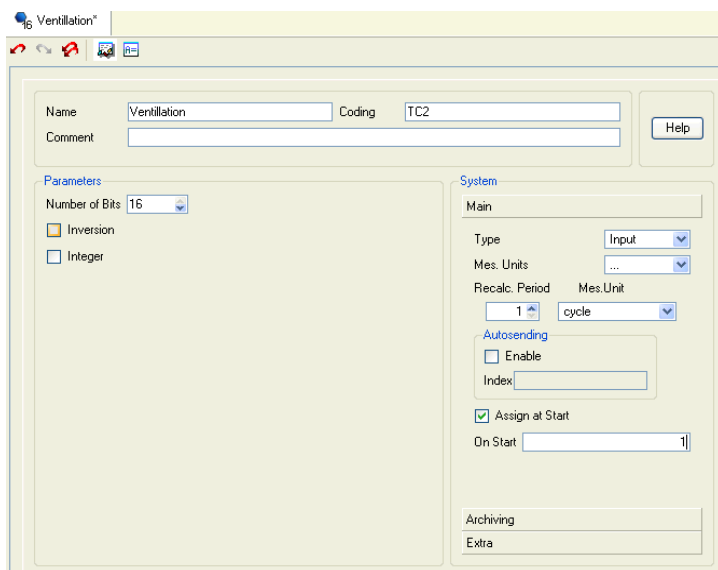
Program ten będzie cyklicznie generował sygnał czujnika. Czas trwania sygnału zależy od argumentu wejściowego **Cycle\_Counter**. Jak tylko wartość ta zostanie osiągnięta na wyjściu na pewien czas pojawi się **1**. Po kompilacji programu wybierz kanał **Door** w grupie **Storage\_Section** (węzeł **RTM\_1**) i ustaw go jako **Call** w zakładce **Info**:

Info	Flags	Arguments	Attributes
Name:	Door	Node:	System.RTM_1
Coding:	TC2	Type:	HEX:16_Channel Reference Counter: 2
Comment:			
Link:			
Call:	Profiler_Discrete_SS(Program_Templates)		
Icon:			


W zakładce **Arguments** połącz argumenty programu z atrybutami **Real Value** aktualnego kanału i ustaw domyślną długość trwania impulsu:

Info Arguments Attributes				
Name	ID Type	Data Type	Def. Value	Link
Door	OUT	USINT		16_Door.Real Value(System.RTM_1.Storage_Section)
Cycle_Counter	IN	REAL	250	

Wykonaj te same operacje dla kanału **Fire\_Alarm**, ale długość trwania impulsu ustaw na 1000. Dla kanału **Ventilation** ustaw następujące parametry:

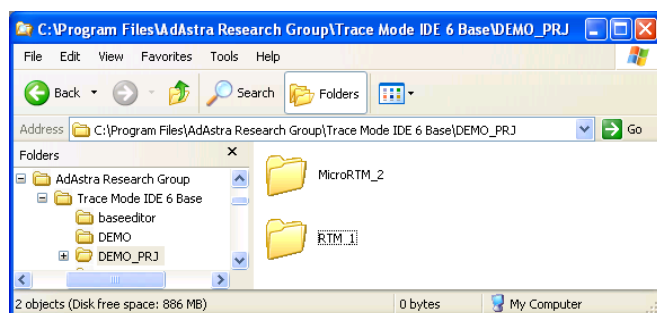


W ten sposób projekt systemu sterowania wykonany w poprzednim rozdziale jest dostępny do uruchomienia i debugowania, wszystkie zewnętrzne urządzenia zostały zastąpione przez symulatory.

Zapisz projekt z nazwą **DEMO\_PRJ.prj** i kliknij na ikonę , aby utworzyć pliki uruchomieniowe dla poszczególnych elementów systemu.

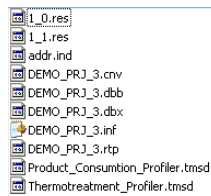
## Pliki projektu

Rezultatem poprzednich działań jest utworzenie folderu z taką samą nazwą jak nazwa projektu (bez rozszerzenia) w folderze gdzie zainstalowany jest **TRACE MODE 6** (domyślnie **C:\Program Files\AdAstra Research Group\Trace Mode IDE 6 Base\DEMO\_PRJ**):

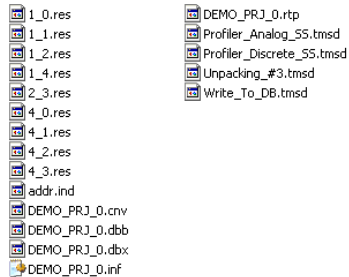


Folder zawiera podkatalogi z nazwami węzłów projektu, zawierające pliki potrzebne do ich uruchomienia. Folder węzła posiada nazwę **<nazwa>\_<l.p. + 1>**, gdzie **nazwa** jest nazwą węzła nadaną w trakcie jego konfiguracji w po-

włóce **System**. Podkatalog **MicroRTM\_2** zawiera następujące elementy:



RTM\_1:



W trakcie eksportu węzłów projektu system generuje następujące pliki:

- **<nazwa prj>\_<l.p.>.cnv** – plik tekstowy, zawierający informacje kontrolne;
- **< nazwa prj>\_< l.p.>.dbb** – plik binarny węzła (wczytywany przez RTM);
- **< nazwa prj>\_< l.p.>.dbx** – plik binarny węzła (wykorzystywany przy konwersji węzła projektu do węzła RTM);
- **< nazwa prj>\_< l.p.>.rtp** – opcjonalny plik tekstowy węzła (wczytywany przez RTM);
- **addr.ind** – opis parametrów węzła (plik jest generowany w każdym folderze);
- **<nazwa programu>.tmsd** – program wywoływany przez kanały węzła (skrypt do debuggowania w IDE);
- **< l.p. + 1>\_<numer porządkowy w węźle >.res** – pliki zawierające kod wykonywalny dla RTM, numeracja pliku zaczyna się od 0;
- **t\_factory.txt** – informacje o kanałach **User** i **T-FACTORY**.

## Debuggowanie

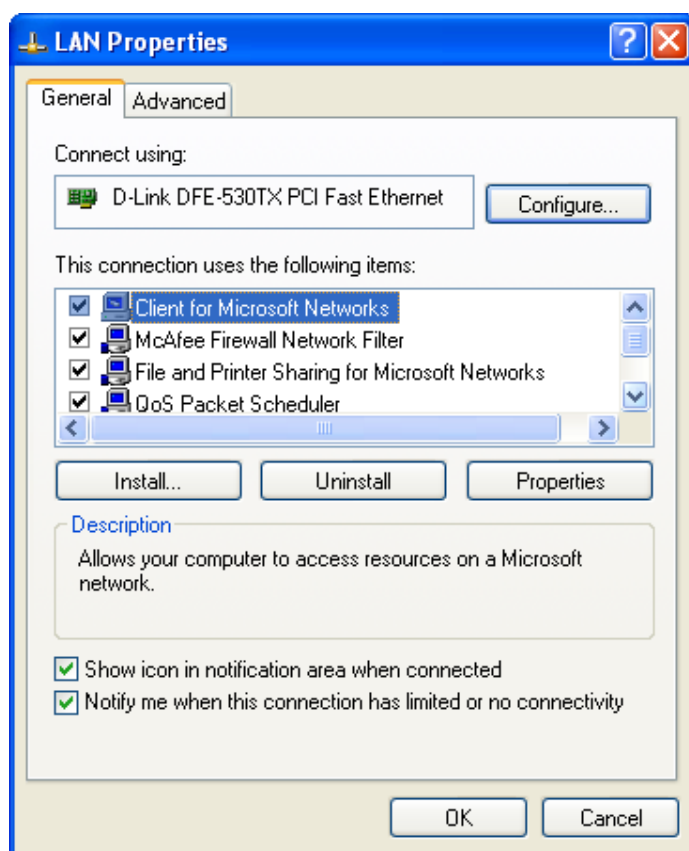
Projekt zawierający wiele węzłów można debuggować bez wykorzystywania urządzeń I/O w następujący sposób:

- przeciągnij węzły projektu do powłoki **Technology**;
- dodaj nowy węzeł w powłóce **System**, np. **RTM\_Common**;

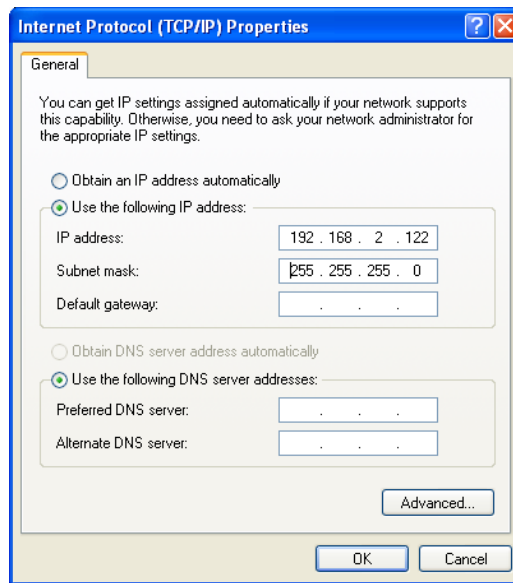
- przeciągnij powłokę **Technology** do nowo utworzonego węzła w powłoce **System**;
- wykonaj dla niego polecenie **Save for RTM** z menu kontekstowego;
- uruchom węzeł **RTM\_Common**, do zmiany atrybutów kanałów wykorzystaj funkcje debuggera z obsługą/bez obsługi ekranów graficznych;
- po zakończeniu debuggowania i dokonaniu niezbędnych zmian usuń węzeł **RTM\_Common**.

W tym przypadku do uruchomienia projektu w trybie demo wymagane są następujące elementy: dwa połączone siecią komputery z systemem operacyjnym **MS Windows XP**, posiadające zainstalowane i poprawnie skonfigurowane połączenie TCP/IP.

## Konfiguracja protokołu sieciowego

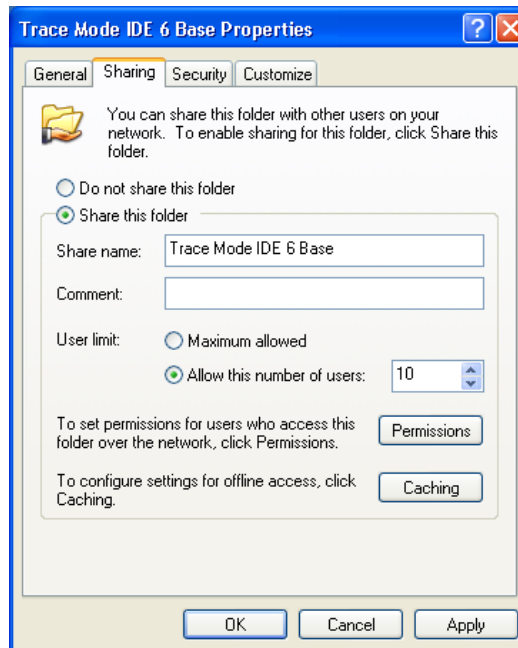


Przy konfiguracji **TCP/IP** wymagane jest ustawienie rzeczywistych adresów IP komputerów połączonych siecią, np. adres IP komputera, na którym będzie uruchomiony węzeł **RTM\_1** to 192.168.2.122, natomiast adres IP dla węzła **MicroRTM\_2** to 192.168.2.123:




## Przygotowanie folderu projektu do debuggowania

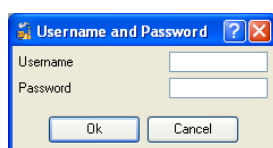
W celu udostępnienia innym komputerom plików utworzonych dla węzłów projektu, zalecamy udostępnić w sieci folder zawierający development system **TRACE MODE 6**:





i zezwolić użytkownikom sieci na dostęp do niego. Pozwoli to uruchomić debugowanie na zdalnych komputerach bez konieczności instalowania **TRACE MODE 6** na nich – **rtc.exe** (debugger z obsługą ekranów graficznych) oraz **rtmg32.exe** (debugger bez obsługi ekranów graficznych) – np. dla węzła **MicroRTM\_1**).

## Debugowanie

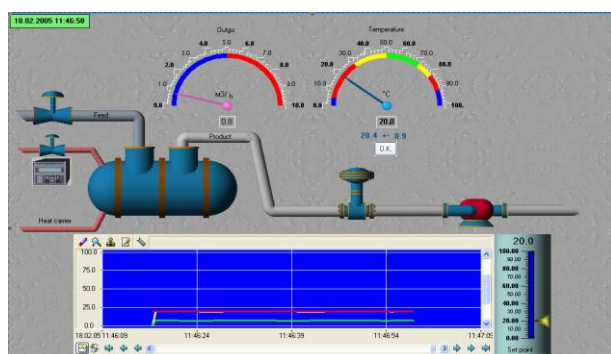
Na komputerze z zainstalowanym systemem **TRACE MODE 6** uruchom debugera z obsługą ekranów graficznych klikając na ikonę  znajdującą się na pulpicie lub przy użyciu pliku znajdującego się w folderze – **C:\TM6**. Wybierz opcję **File** w oknie które się pojawi, następnie – opcję **Open** i zaznacz plik **DEMO\_PRJ\_0.dbb** z podkatalogu **\RTM\_1**, aby wczytać węzeł stacji roboczej. Po zakończeniu wczytywania pojawi się następujące okno:



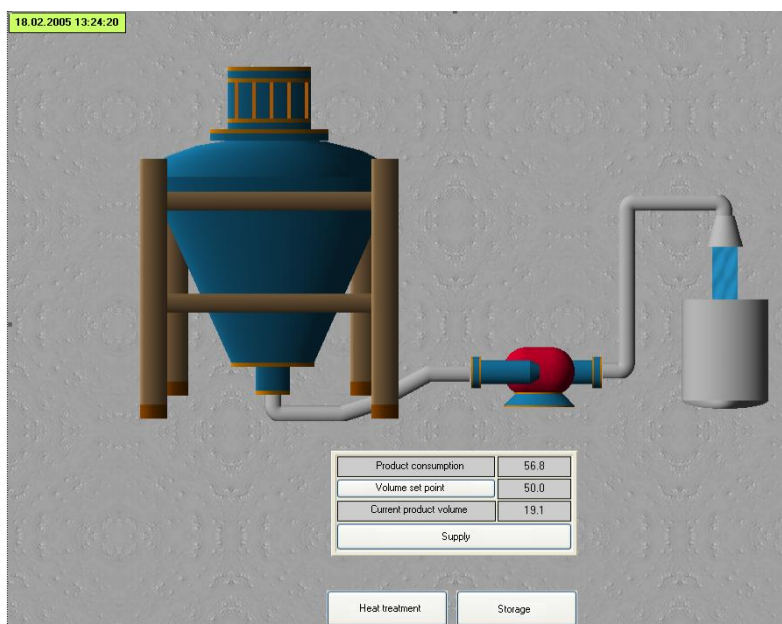
podaj w nim **Login** i **Hasło** użytkownika projektu – **admin** lub **oper**. Naciśnij **Ctrl+R**, aby uruchomić węzeł lub kliknij na ikonę  z paska narzędzi. Możesz także uruchomić węzeł z poziomu IDE zaznaczając węzeł **RTM\_1** w powłoce **System** i klikając ikonę  z paska narzędzi.

Do uruchomienia węzła z kontrolerem PAC na zdalnym komputerze otwórz folder **\\Nazwa\_komputera\_z\_zainstalowanym\_IDE\TM6** i uruchom plik **rtmg32.exe**, w oknie **Open** wybierz plik **DEMO\_PRJ\_1.dbb** z podkatalogu **\MicroRTM\_2**. Kliknij na ikonę , aby uruchomić projekt.

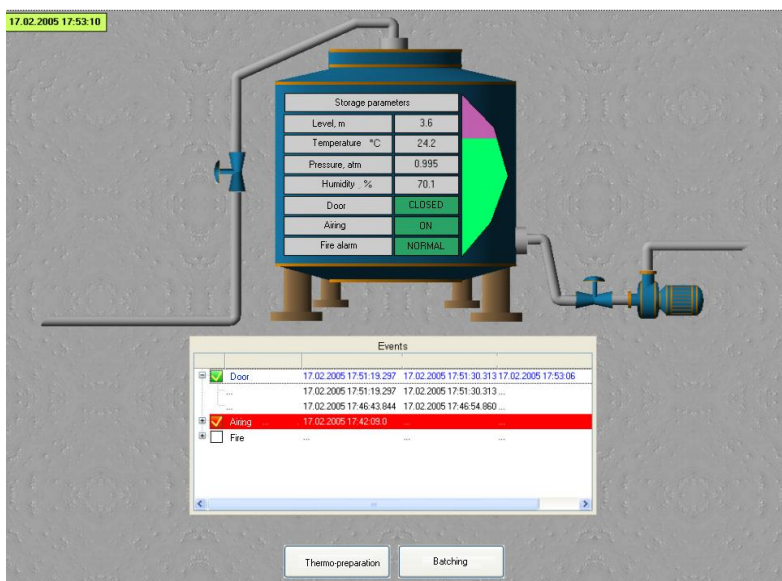
Jeżeli ustawienia sieciowe zostały poprawnie skonfigurowane, na ekranie graficznym węzła stacji roboczej, wyświetlającym sekcję **Thermotreatment**, będzie widać zmieniające się wartości. Po zmianie nastaw regulatora odpowiedź obiektu będzie widoczna na wykresie:



Przechodząc pomiędzy ekranami możesz monitorować sekcję Storage, dozowanie produktu,



zatwierdzać zdarzenia, które miały miejsce, itp.




Dane, które zostały zapisane do raportu alarmów węzła stacji roboczej (plik **alarms.txt**), są dostępne do przeglądania przy użyciu np. **Notatnika**:




17.02.2005 17:42:02.0	admin	Login	
17.02.2005 17:42:07.0	DEMO_PRJ_0	START	
17.02.2005 17:46:43.8	M Door	TC2	Door opened
17.02.2005 17:46:54.8	M Door	TC2	Door closed
17.02.2005 17:51:19.2	M Door	TC2	Door opened
17.02.2005 17:51:30.3	M Door	TC2	Door closed

Parametry sekcji Storage zapisywane do bazy danych także są dostępne do przeglądania w ‘czasie rzeczywistym’:

Code	my_datim	my_hum	my_level	my_pres	my_temp
824	17.02.2005 13:55:55	70.1176	3.64695	0.994695	24.194
825	17.02.2005 13:55:58	69.8824	3.35305	0.965305	23.806
826	17.02.2005 13:56:01	70.1902	3.73776	1.00378	24.3138
827	17.02.2005 13:56:04	69.8098	3.26224	0.956224	23.6862
828	17.02.2005 13:56:06	70.1902	3.73776	1.00378	24.3138
829	17.02.2005 13:56:09	69.8098	3.26224	0.956224	23.6862
830	17.02.2005 13:56:12	70.1176	3.64695	0.994695	24.194
831	17.02.2005 13:56:15	69.8824	3.35305	0.965305	23.806
832	17.02.2005 13:56:17	70	3.5	0.98	24
833	17.02.2005 13:56:20	70	3.5	0.98	24
834	17.02.2005 13:56:23	70	3.5	0.98	24
835	17.02.2005 13:56:26	70.1176	3.64695	0.994695	24.194
836	17.02.2005 13:56:28	69.8824	3.35305	0.965305	23.806
837	17.02.2005 13:56:31	70.1902	3.73776	1.00378	24.3138
838	17.02.2005 13:56:34	69.8098	3.26224	0.956224	23.6862
839	17.02.2005 13:56:37	70.1902	3.73776	1.00378	24.3138

Kliknij na ikonę , aby zakończyć pracę węzła.

Ikona  z paska narzędziowego dostarcza informacji potrzebnych do określenia wymaganej licencji RTM (liczba kanałów). Kliknięcie na ikonę powoduje wyświetlenie okna zawierającego informacje o całkowitej liczbie elementów w projekcie oraz ich rozmieszczeniu w węzłach:

Object Type	Maximum	Used	Remaining
Screen	16384	4	16380
Document	16384	0	16384
Database	16384	1	16383
Channel	65536	49	65487
Emitter/Receiver	32768	2	32766
RTM_1	65535	35(35)	65500(65500)
MicroRTM_2	65535	8(8)	65527(65527)

## Wnioski

Powyższy materiał ukazuje tylko jedno z podejść do zagadnienia projektowania systemów sterowania przy użyciu **TRACE MODE 6** i został stworzony do wprowadzenia użytkownika w nowy produkt. Mamy nadzieję, że informacje zawarte w tym materiale okażą się efektywne, a **TRACE MODE 6** w niedługim czasie stanie się Twoim ulubionym narzędziem – zapewniającym najlepsze rozwiązania w dziedzinie sterowania.

**AdAstrA Research Group, Ltd**  
PO Box 38, Moscow 107076, Russia  
Tel: +7 (495) 771-71-74 or  
Fax: +7 (495) 518-9846  
Email: [adastra@adastra.ru](mailto:adastra@adastra.ru)  
<http://www.adastra.ru>