ASTOR

gdzie technologia spotyka człowieka

Tworzenie aplikacji w Astraada HMI CFG

Pierwsze kroki

SPIS TREŚCI

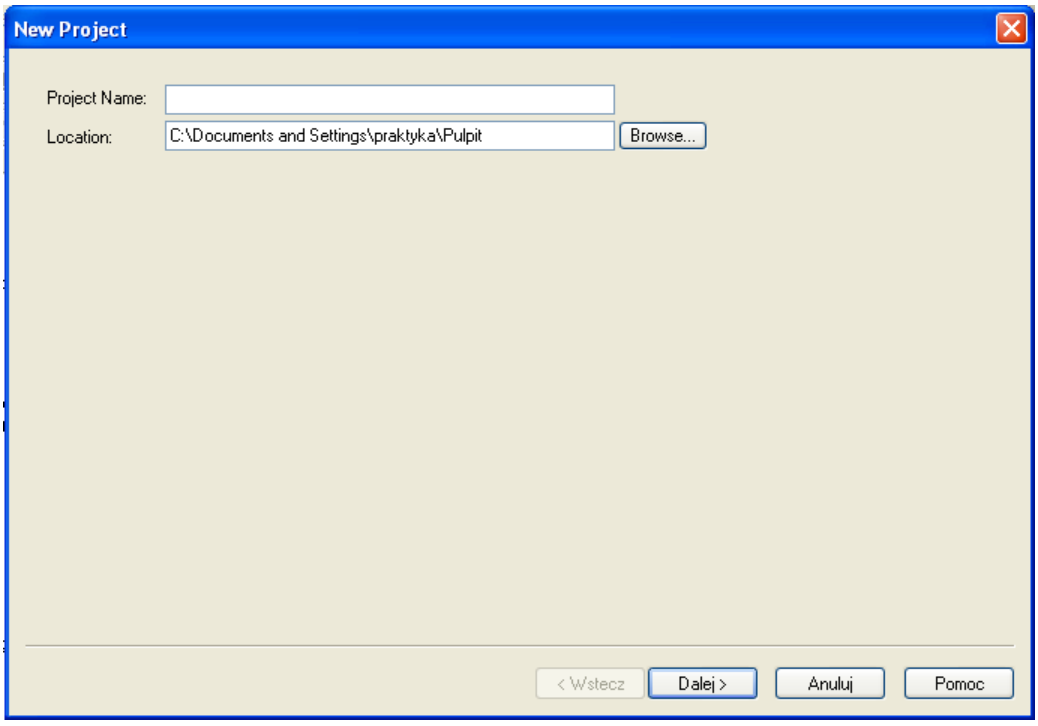
Astraada HMI CFG jest programem służącym do tworzenia aplikacji na panele operatorskie HMI firmy Astraada.

Jako przykładową aplikację stworzymy symulację sterowania elektrownią wodną. Wodę będziemy pobierać z zalewu i dostarczać do turbin poprzez cztery zawory. Będzie można kontrolować stopień otwarcia każdego z tych zaworów jak i pobierać i archiwizować dane dotyczące pracy elektrowni. O sytuacjach krytycznych powiadamiać będą nas alarmy.

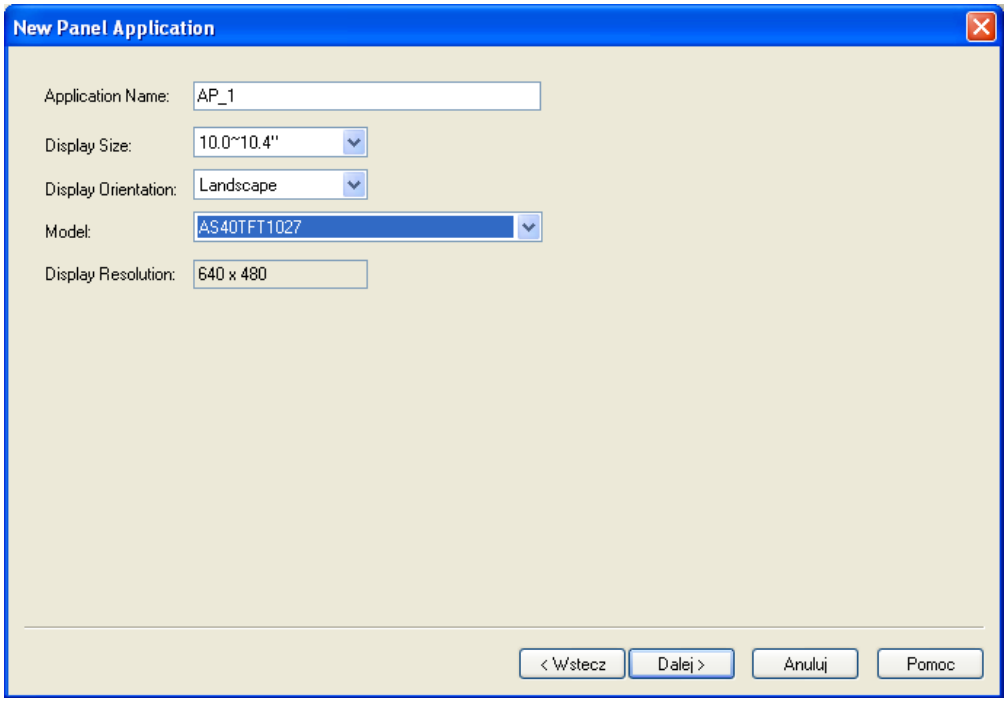
TWORZENIE NOWEGO PROJEKTU

Klikamy *File -> New*

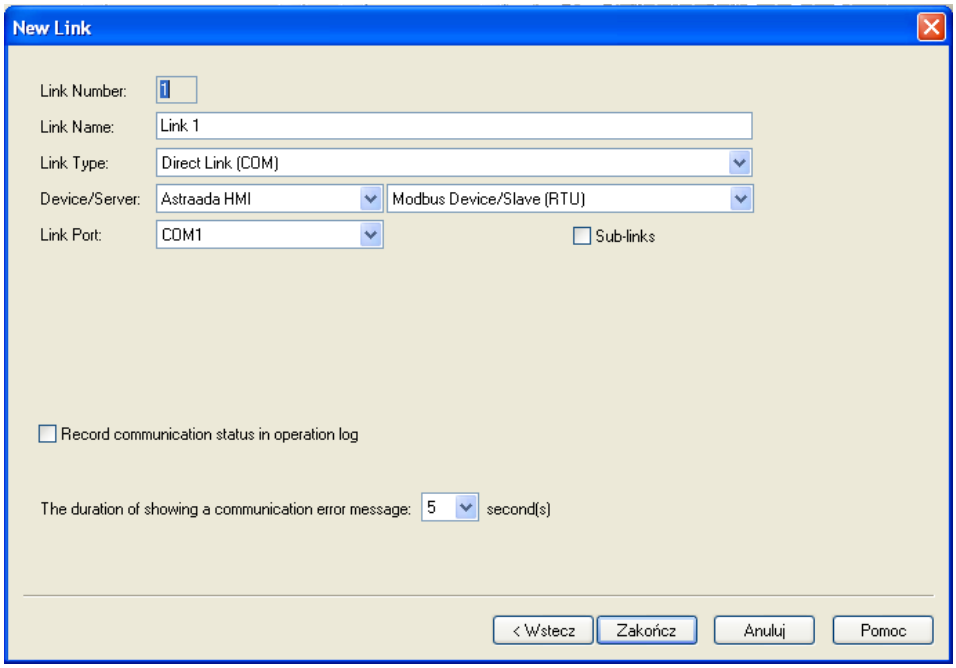
Pokazuje się okienko, w którym wpisujemy nazwę projektu oraz miejsce jego zapisania. Klikamy *Dalej*.



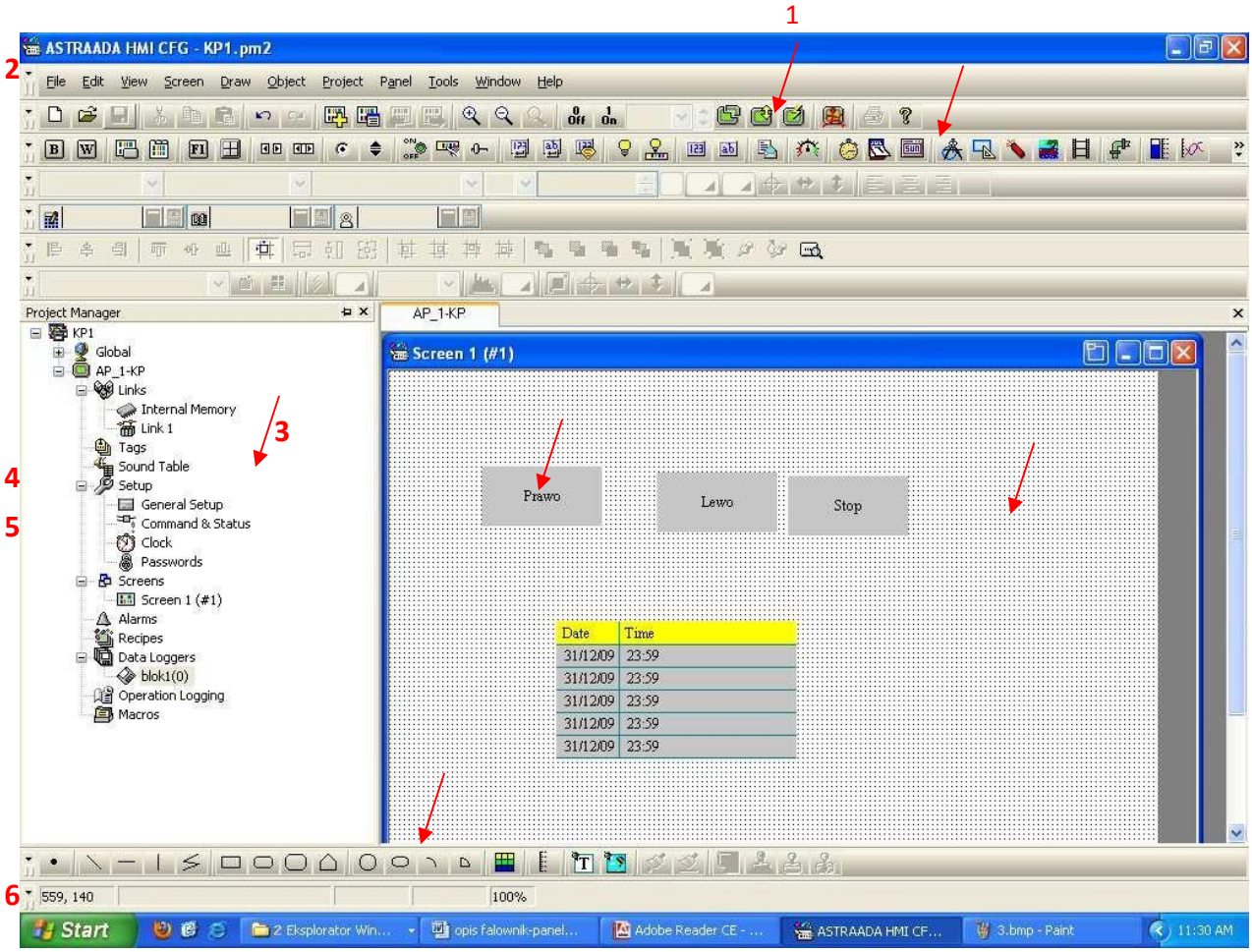
Wpisujemy nazwę tworzonej aplikacji. Wybieramy wielkość ekranu naszego panelu HMI. Następnie w oknie Model wybieramy model, na którym przyszło nam pracować. Klikamy *dalej*.



W następnym oknie wpisujemy rodzaj łącza (pomiędzy HMI a urządzeniem docelowym), oraz rodzaj i port komunikacji. Klikamy *Zakończ*.
W naszym przypadku nie ma to znaczenia, gdyż aplikacja jest symulacją wyłącznie na urządzenie HMI.



Otwiera nam się okno projektu:



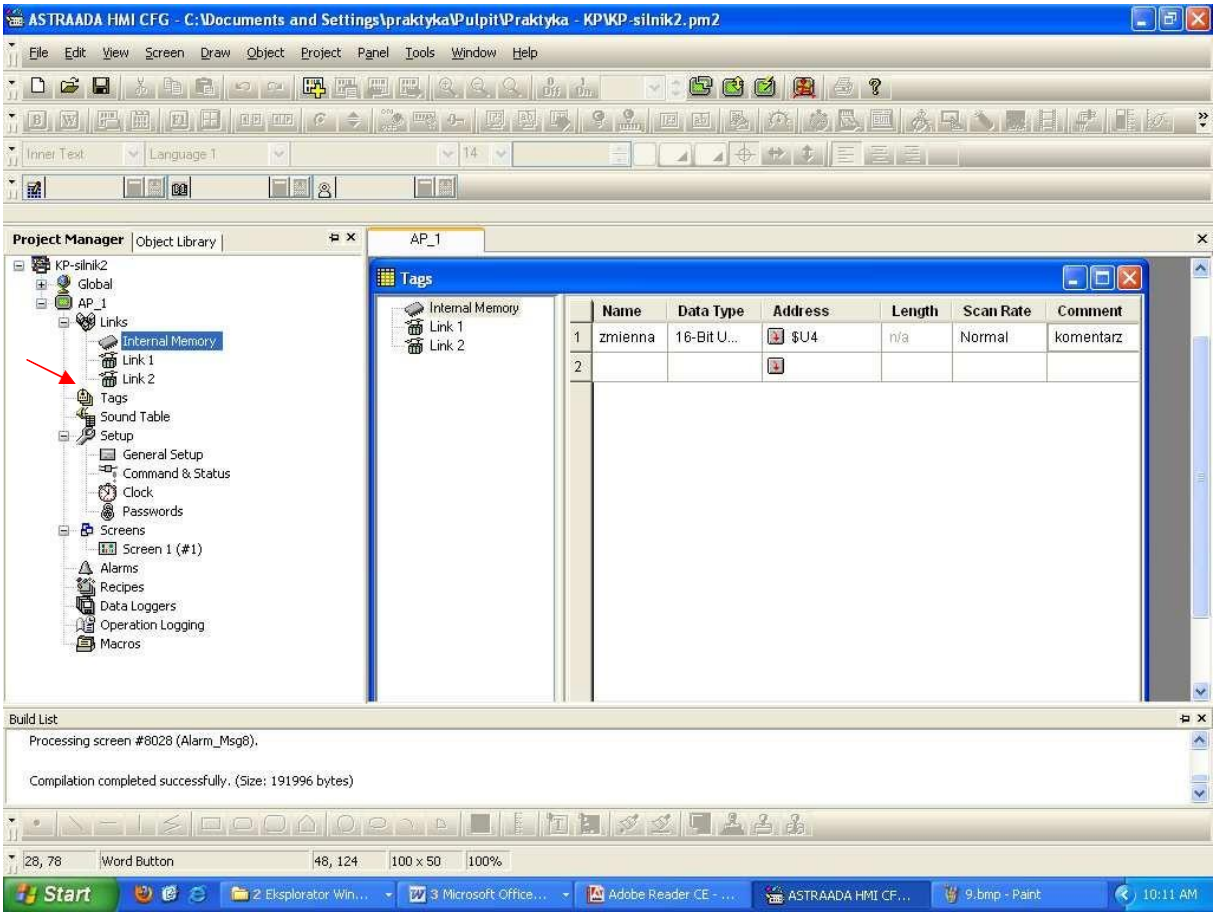
Okno z kilkoma przykładowymi elementami już przeniesionymi do obszaru roboczego.

1 Ikonki służące do kompilacji, ładowania, szybkiego ładowania, oraz symulacji programu.
2 Pasek z ikonkami obiektów możliwych do umieszczenia na panelu operatorskim.
3 Drzewko ze wszystkimi elementami naszego projektu.
4 Przykładowy obiekt panelu.
5 Widok projektowanego ekranu.
6 Pasek z narzędziami do rysowania.

Z dostępnych elementów składamy naszą aplikacją. Można użyć narzędzi rysowania, aby nasze dzieło stało się bardziej przejrzyste, gdyż ergonomia i intuicyjność to jedne z najważniejszych cech projektu często pomijane przez inżynierów.

DEKLARACJA ZMIENNYCH

Aby zadeklarować zmienne klikamy dwukrotnie na **Tags** w Project Manager. Wyświetla nam się tabela, w której wpisujemy nazwy zmiennych, ich typ oraz adres. Do każdej z nich możemy dodać komentarz.



Zmienne użyte w przykładzie:

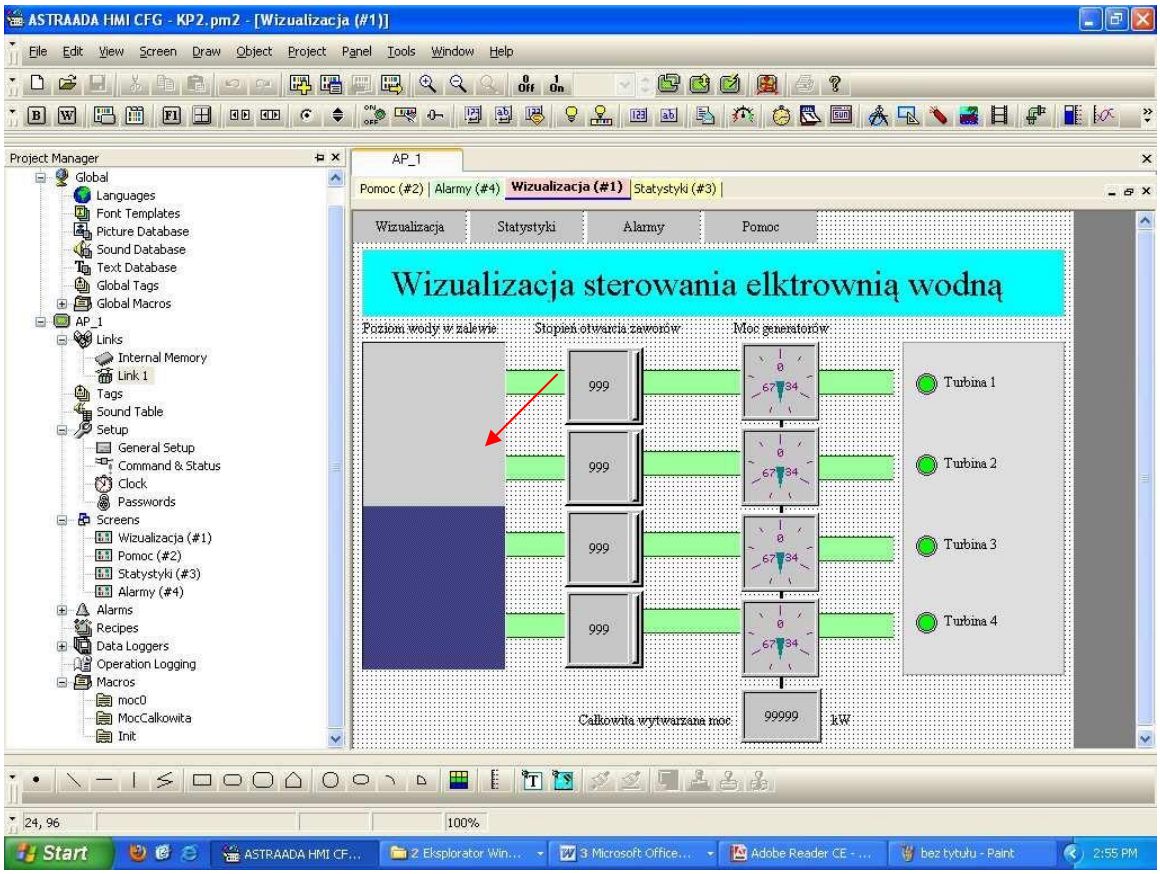
	Name	Data Type	Address	Length	Scan Rate	Comment
1	zawor1	16-Bit Si...	$U0	n/a	Normal	
2	zawor2	16-Bit U...	$U1	n/a	Normal	
3	zawor3	16-Bit U...	$U2	n/a	Normal	
4	zawor4	16-Bit U...	$U3	n/a	Normal	
5	turbina1	16-Bit U...	$U4	n/a	Normal	
6	turbina2	16-Bit U...	$U5	n/a	Normal	
7	turbina3	16-Bit U...	$U6	n/a	Normal	
8	turbina4	16-Bit U...	$U7	n/a	Normal	
9	Pcalkowita	16-Bit U...	$U8	n/a	Normal	
10	StanWody	16-Bit U...	$U9	n/a	Normal	
11	StanTurbin	16-Bit U...	$U10	n/a	Normal	
12	przelew	16-Bit U...	$U11	n/a	Normal	
13	NiskaWoda	16-Bit U...	$U12	n/a	Normal	
14	WysokaWoda	16-Bit U...	$U13	n/a	Normal	
15	BrakWody	16-Bit U...	$U14	n/a	Normal	
16						

Pracuję na zmiennych lokalnych ponieważ nasz projekt składa się tylko z jednej aplikacji.

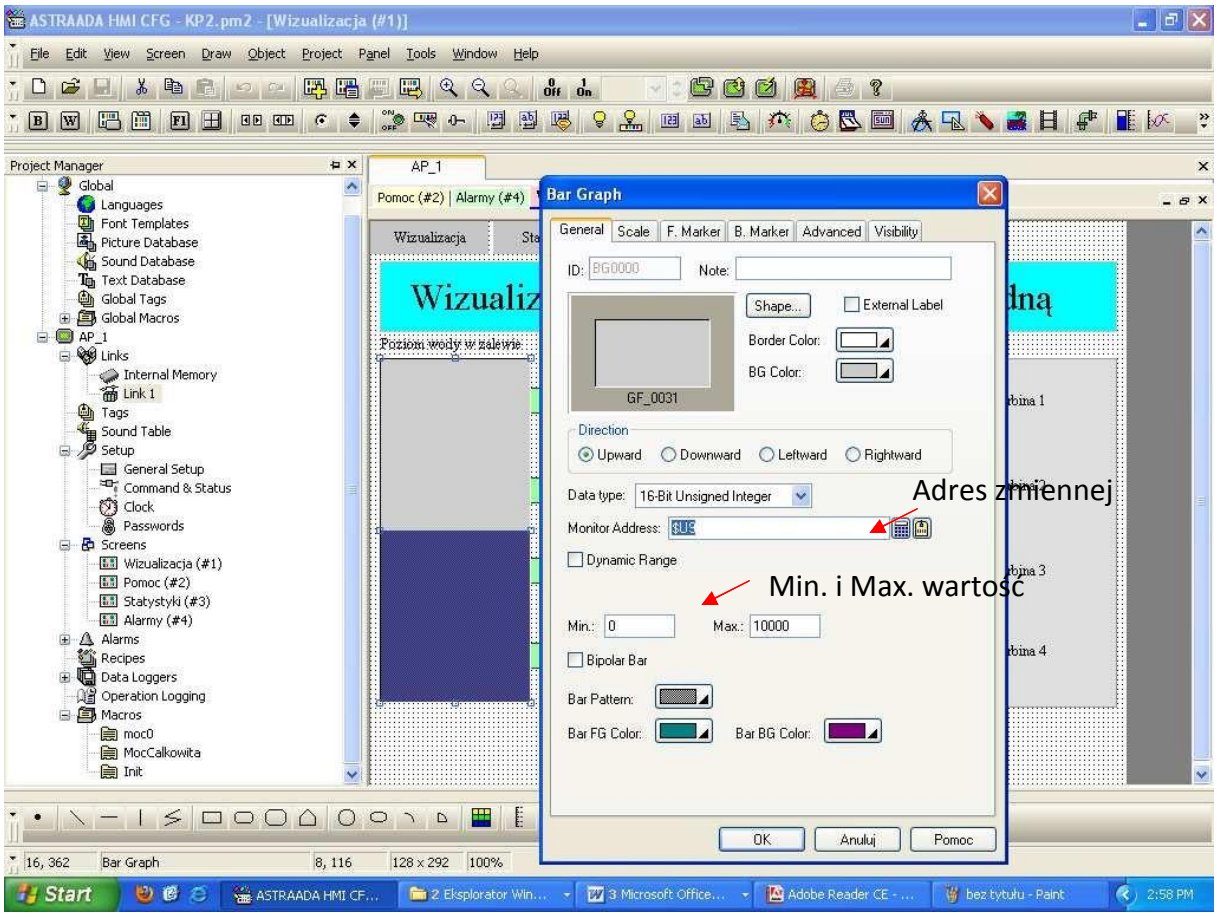
TWORZENIE EKRANÓW

Bar Graph

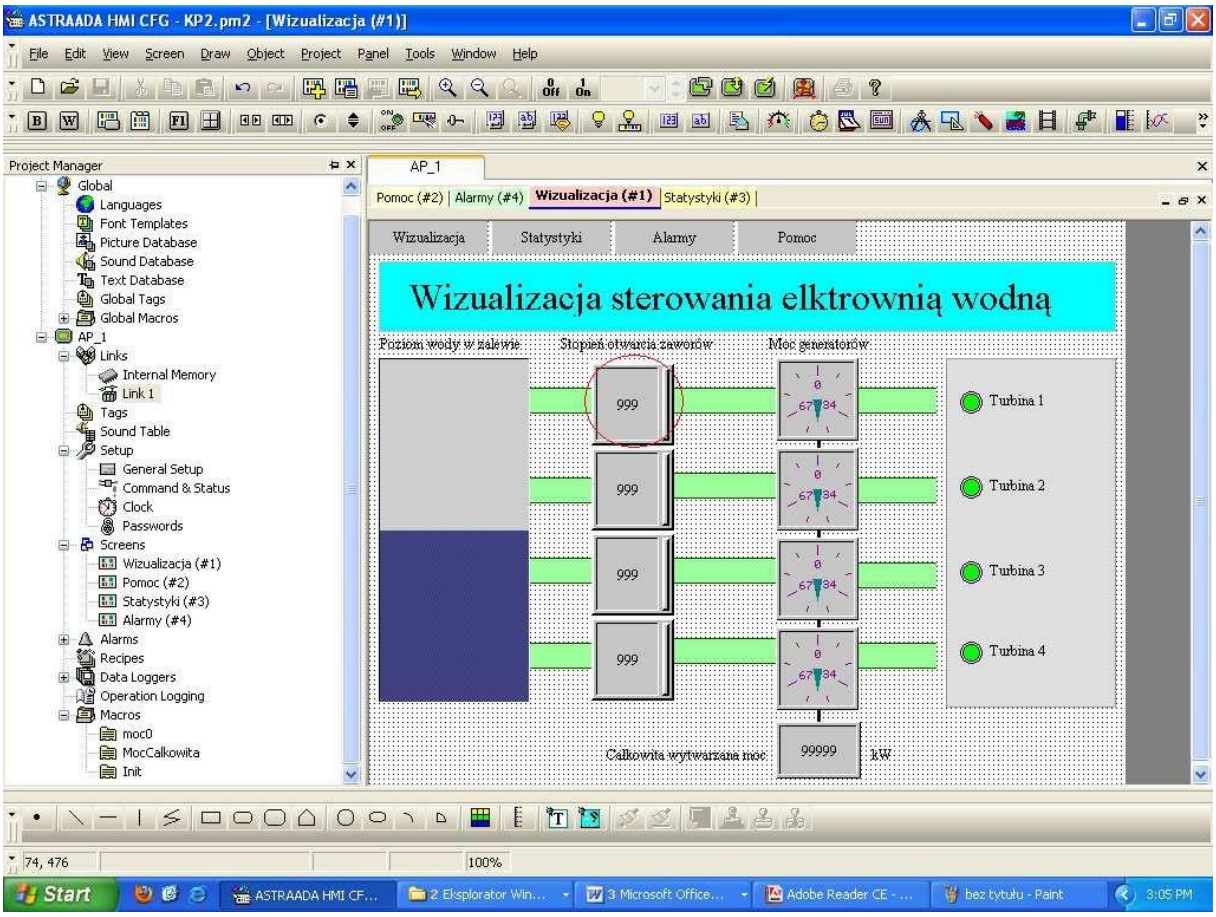
 Z paska narzędzi wybieramy obiekt **Bar Graph**, a następnie przeciągamy go do obszaru roboczego. Ten obiekt będzie symbolizował nam poziom wody w zalewie przed elektrownią wodną.



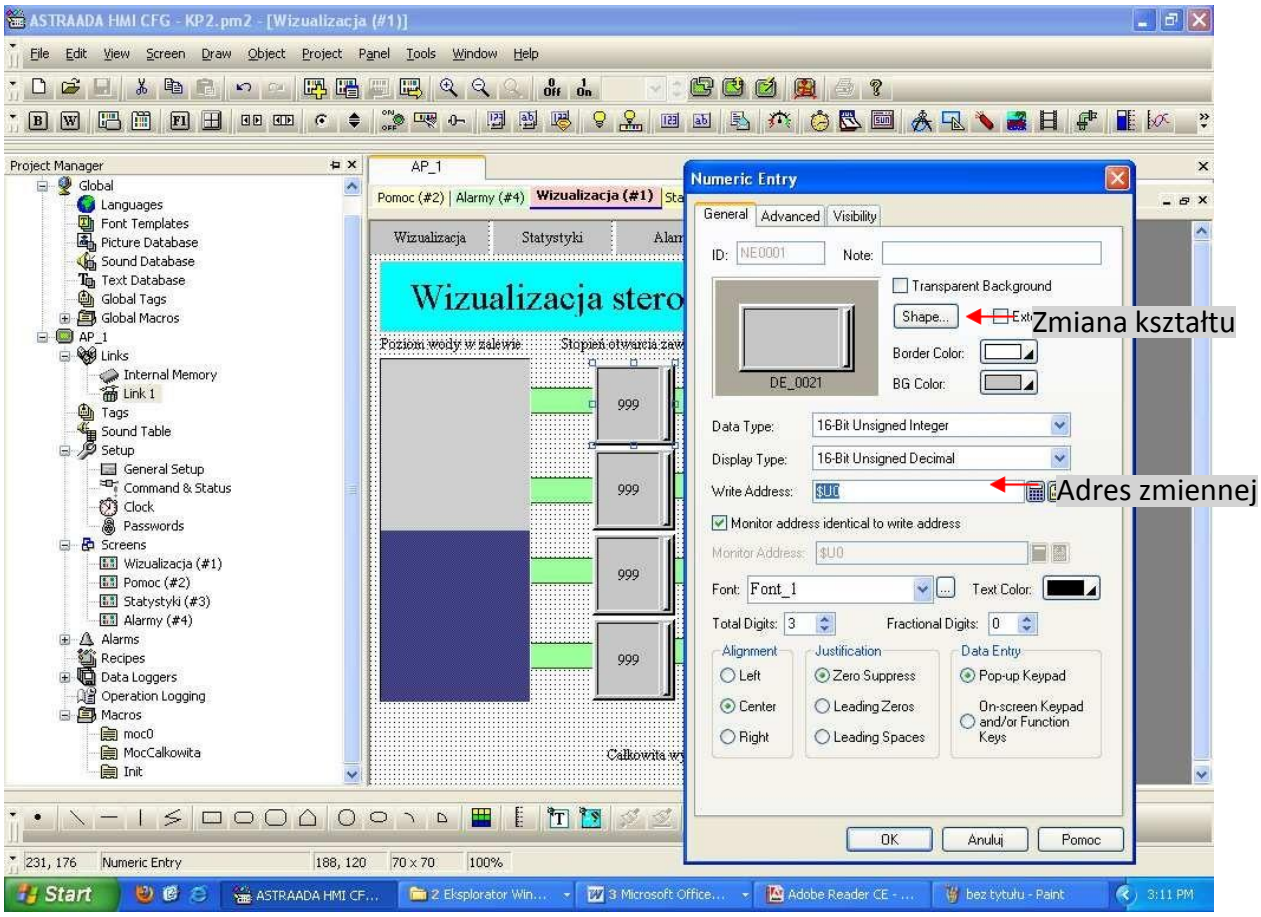
W oknie właściwości (dwukrotne kliknięcie na obiekt) ustawiamy jego „pojemność” czyli minimalną i maksymalną wartość, przypisujemy zmienną przechowującą wartość poziomu zbiornika oraz opcje graficzne.



Następnie wybieramy: Obiekt do zadawania wartości liczbowej **Numeric Entry**.



Po kliknięciu na niego (w trybie pracy na HMI, bądź symulatorze) otwiera nam się okienko z klawiaturą, na której wybieramy pożądaną przez nas wartość. Po jej zatwierdzeniu jest ona przypisana zmiennej skonfigurowanej w oknie właściwości.

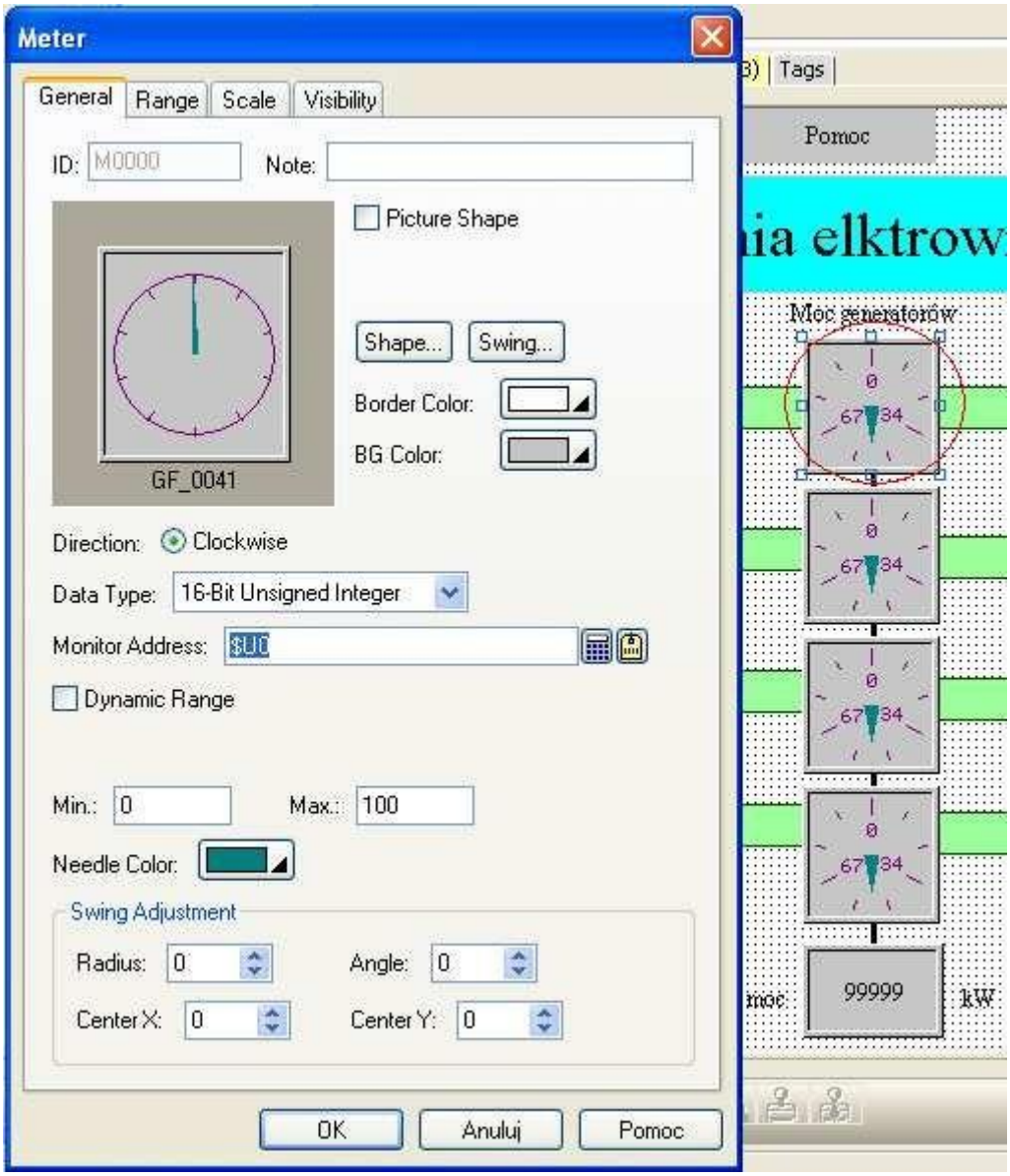


Będzie to przycisk odpowiadający z procent otwarcia zaworów turbin generatora. Ponieważ nasza elektrownia składać się będzie z czterech turbin kopiujemy jeszcze trzy razy ten element (zaznaczamy obiekt, Ctrl+C, Ctrl+V). Warto zauważyć, że znacznie wygodniej kopiować element po ustawieniu jego parametrów. Dzięki temu jego kopie będą miały identyczne właściwości i pozostanie nam jedynie zmiana adresów zmiennych przypisanych do kolejnych kopii obiektu.

Teraz dodajemy wskaźniki wytwarzanej mocy w generatorach.

Meter

Z paska narzędzi wybieramy **Meter**.
Klikamy na obszar roboczy i pojawia się nam analogowy wskaźnik. Klikamy na niego dwukrotnie i ustawiamy jego parametry.

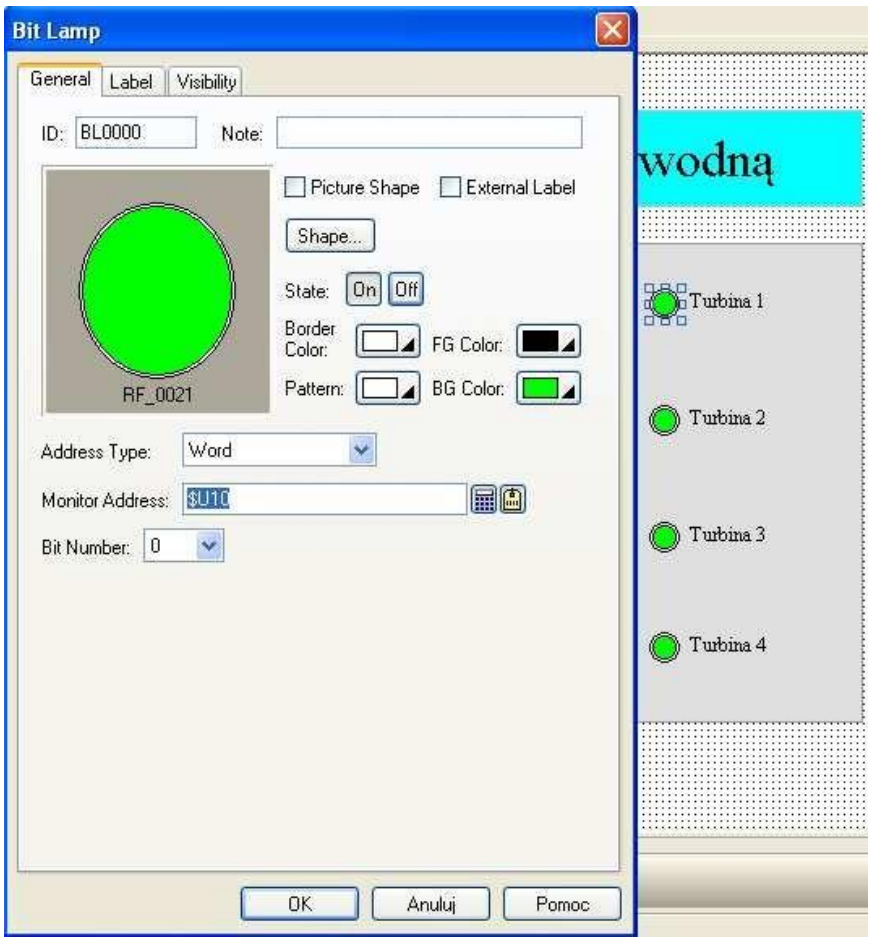


W Monitor Address wpisujemy adres zmiennej, którą będziemy monitorować.
Pola Min. i Max. służą do zadania przedziału wartości mierzonych.
Można również ustawić wiele więcej parametrów, głównie służących polepszeniu czytelności wskaźnika.

Aby wszystko wyglądało jak w prawdziwej elektrowni dodajemy rury ikonką  (w takim układzie jak na poprzednich obrazkach).

Bit Lamp

Służy do monitorowania wartości bitowych. W naszym projekcie potrzebujemy cztery takie obiekty jako wskaźniki pracy każdej z turbin. Konfigurację przeprowadzamy jak zawsze w oknie właściwości.

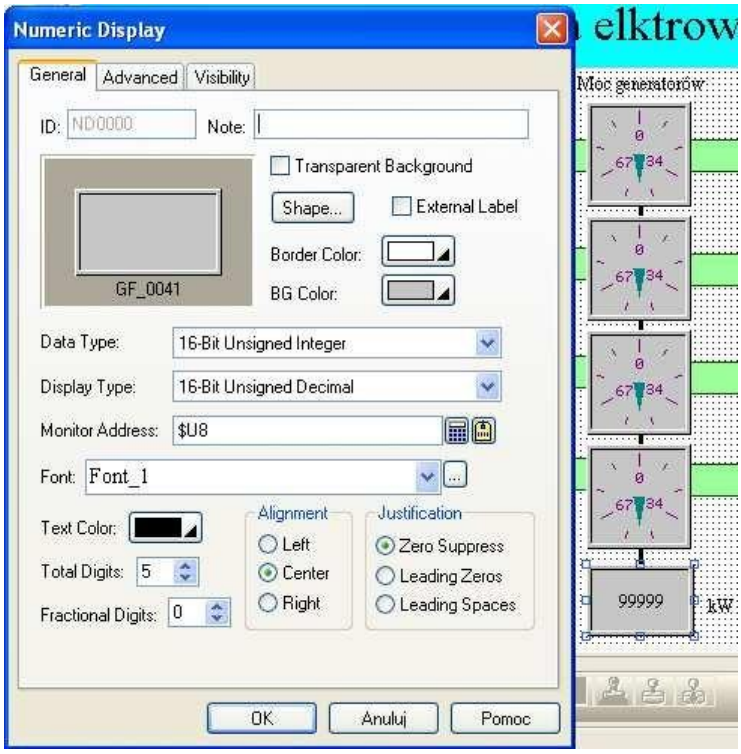


W Monitor Address wpisujemy adres zmiennej, którą będziemy obserwowali. U nas będzie jedna zmienna typu integer odpowiedzialna za stan wszystkich turbin (dlatego wybieramy Word w Address Type). Potrzebujemy tylko informację binarną więc kolejne bity tej zmiennej będą odpowiadały pracy kolejnych turbin (‘0’ gdy nie pracują, ‘1’ podczas pracy). W zakładce Label kasujemy teksty gdyż potrzebujemy tylko graficznej kontrolki. Tam również ustawiamy kolory w stanie ‘0’ i ‘1’.

Numeric Display

Obiekt do odczytywania wartości liczbowej.

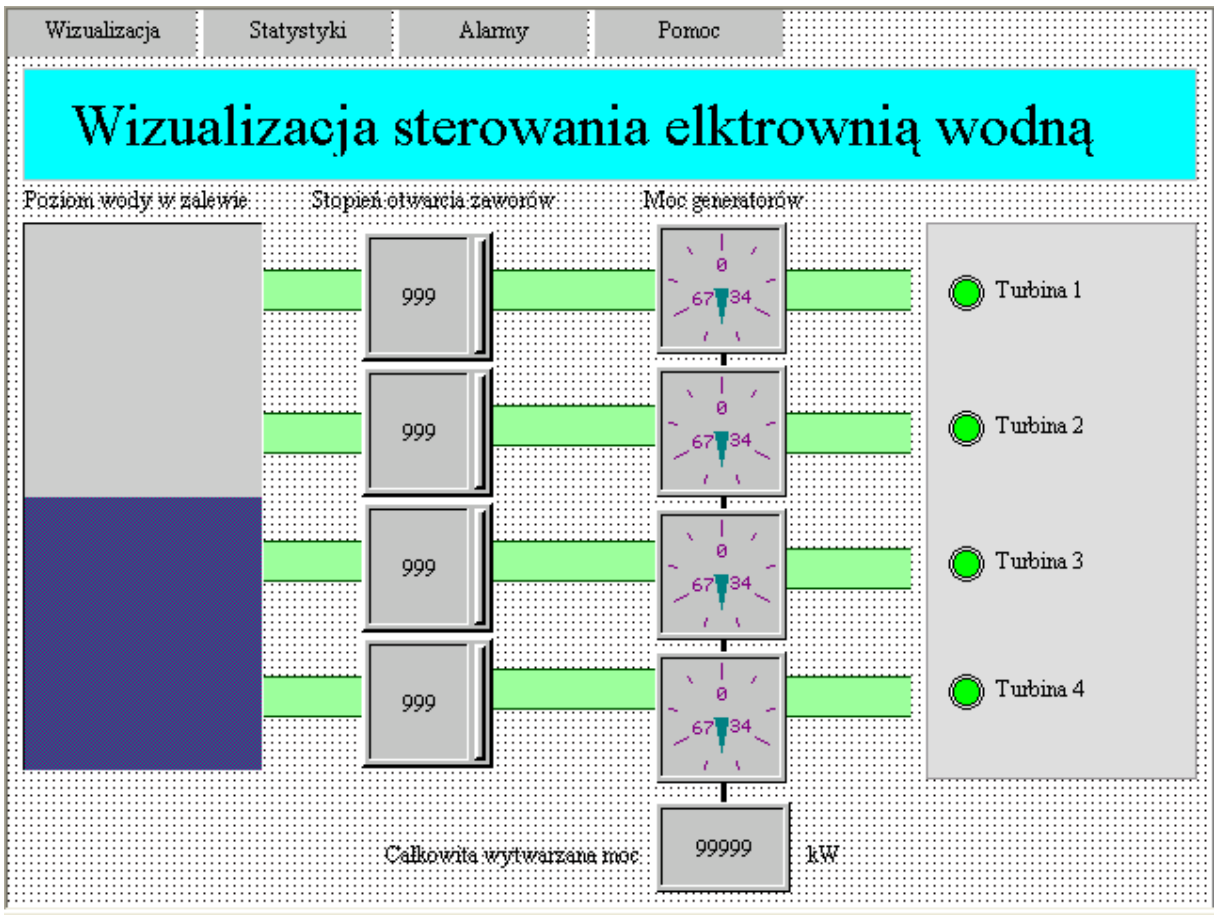
Będzie nam potrzebny taki jeden. Wyświetlać się będzie na nim całkowita generowana moc w turbinach elektrowni.



Podpinamy zmienną w polu Monitor Address, która będzie przechowywała aktualną informacją na temat generowanej mocy. Funkcja przypisująca tą wartość będzie podana w dalszej części, przy omawianiu makr.

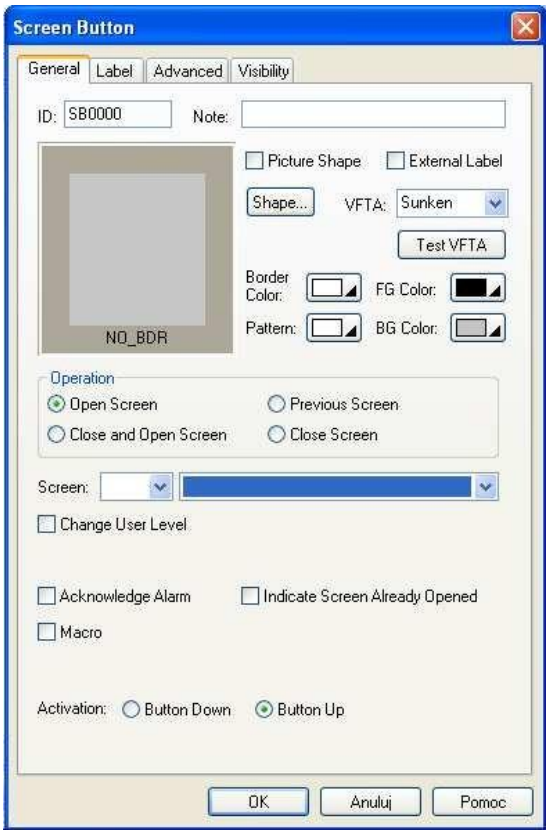
Pozostaje nam dodać odpowiednie opisy za pomocą ikonki do dodawania tekstu . Oraz prostokątne tła dla zwiększenia estetyki interfejsu (ikonka). Wskaźniki z wyświetlaczem mocy całkowitej możemy połączyć pionową linią (ikonka). Pojawi się ona na tle wszystkich elementów. Aby umieścić ją pod nimi klikamy na nią prawym klawiszem myszy i wybieramy *Send to Bottom*.

Ekran wizualizacji powinien ostatecznie tak wyglądać:



Screen Button


Przycisk służący do przełączania ekranów.

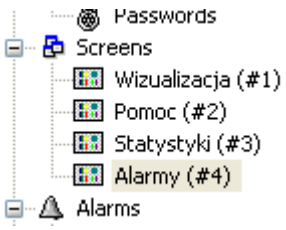


Gdy nie mieszczą nam się obiekty na ekranie, chcemy zadbać o czytelność projektu lub po prostu chcemy wyodrębnić pewne funkcje tworzymy kilka ekranów, które możemy zmieniać tymi przyciskami.

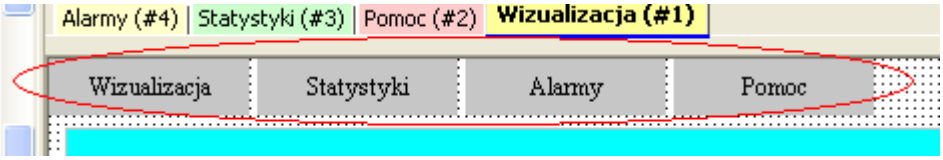
W oknie właściwości najważniejszym polem jest Screen, w którym podajemy jaki ekran ma się wyświetlić po naciśnięciu przycisku.
Oczywiście można też ustawić inną akcję typu zamknięcie ekranu lub inne wyszczególnione w grupie kontrolek Operation.

UWAGA Przy wybraniu opcji Close and Open Screen lub Close Screen nasz ekran musi być typu Window Screen. MoŜna to ustawić klikając prawym klawiszem na nasz ekran w Project Manager i wybierając Properties.

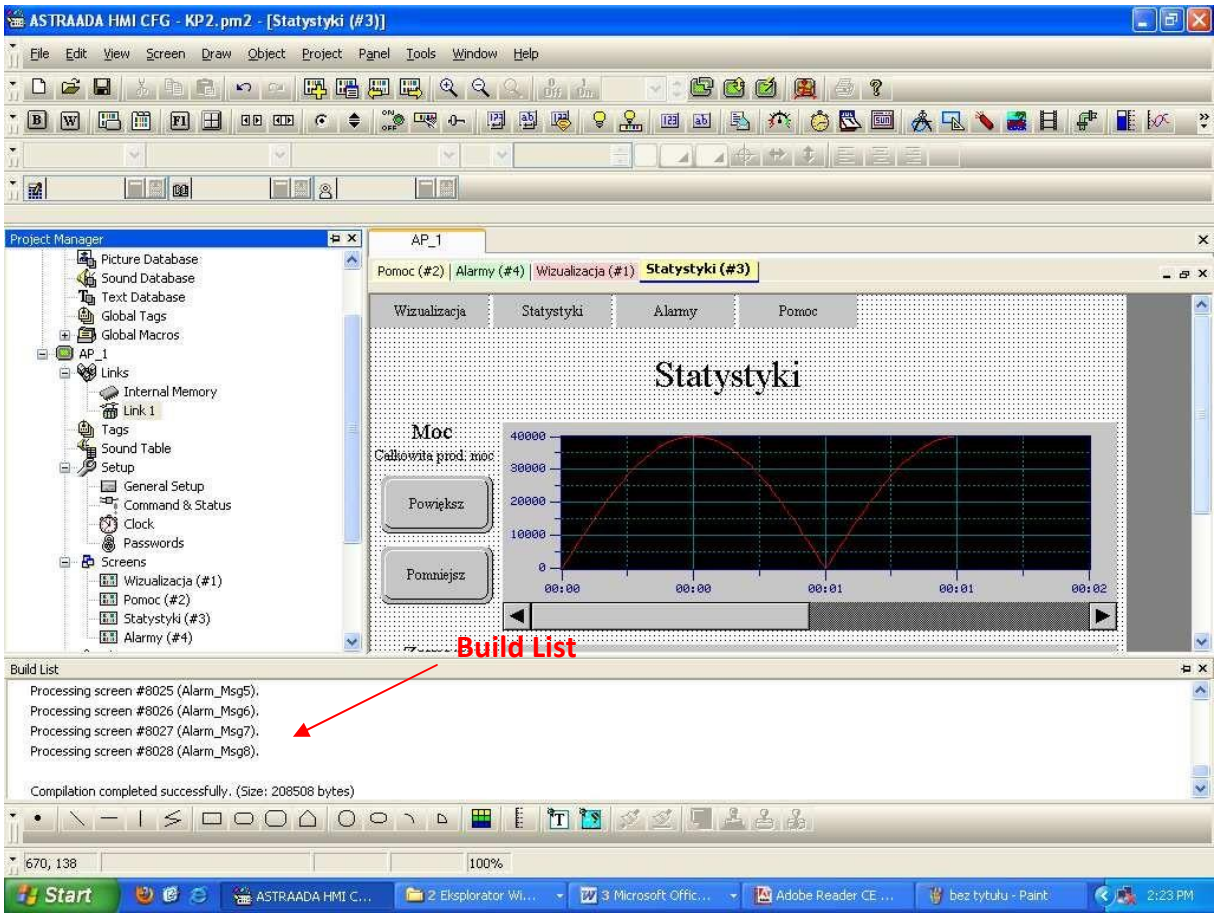
Nowe ekrany tworzymy klikając prawym przyciskiem w Project Manager a następnie New Screen…W ten sposób tworzymy dodatkowe trzy ekrany.



Na każdym z nich umieszczamy cztery przyciski odsyłające do każdego z ekranów.

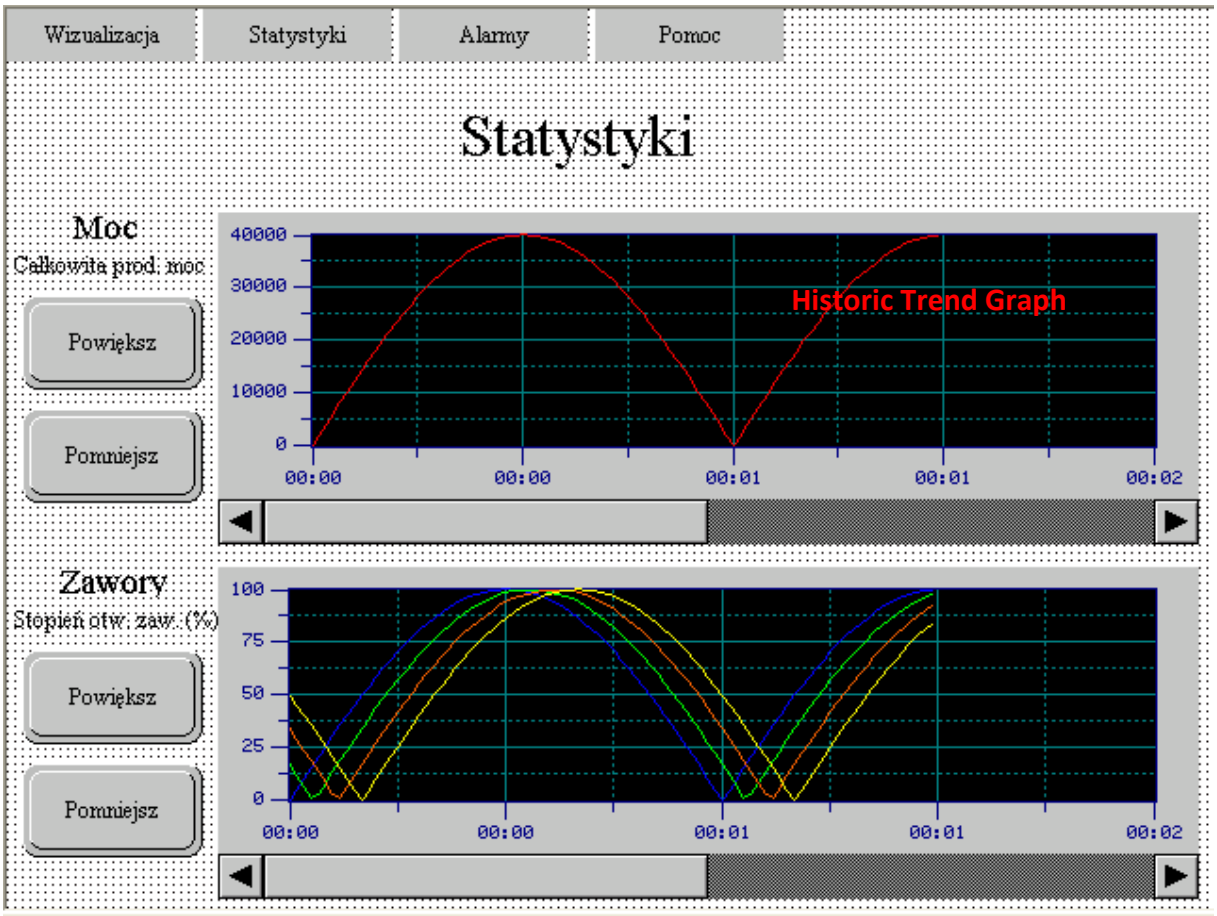


Przy dolnej krawędzi ekranu ukryte jest pole Build List. Po rozciągnięciu myszką będziemy w stanie zobaczyć wszystkie błędy, które pojawiły się po konfiguracji. Przed ich usunięciem nie będziemy w stanie załadować programu do naszego HMI, co zresztą wydaje się oczywiste.


ASTRAADA HMI CFG - KP2.pm2 - [Statystyki (#3)]
File Edit View Screen Draw Object Project Panel Tools Window Help
Project Manager
Picture Database
Sound Database
Text Database
Global Tags
Global Macros
AP_1
Links
Internal Memory
Link 1
Tags
Sound Table
Setup
General Setup
Command & Status
Clock
Passwords
Screens
Wizualizacja (#1)
Pomoc (#2)
Statystyki (#3)
Alarmy (#4)
AP_1
Pomoc (#2) | Alarmy (#4) | Wizualizacja (#1) | Statystyki (#3)
Wizualizacja
Statystyki
Alarmy
Pomoc
Statystyki
Moc
Całkowita prod. moc
Powiększ
Pomniejsz
40000
30000
20000
10000
0
00:00
00:00
00:01
00:01
00:02
Build List
Processing screen #8025 (Alarm_Msg5).
Processing screen #8026 (Alarm_Msg6).
Processing screen #8027 (Alarm_Msg7).
Processing screen #8028 (Alarm_Msg8).
Compilation completed successfully. (Size: 208508 bytes)
Build List
670, 138
100%
Start
ASTRAADA HMI C...
2 Eksplorator Wi...
3 Microsoft Offic...
Adobe Reader CE ...
bez tytułu - Paint
2:23 PM

Historic Trend Graph

Na drugim ekranie naszej aplikacji **Statystyki** będą wyświetlane przebiegi czasowe generowanej mocy oraz stopnia otwarcia zaworów. Stworzymy je za pomocą **Historic Trend Graph** (ikonka).
Utwórzmy dwa wykresy i z każdym z nich powiążmy **pasek przewijania** (ikonka) poprzez wpisanie w Associated Object ID numeru ID (możemy je sprawdzić we właściwościach wykresu) należącego do właściwego *Historic Trend Graph* (oczywiście robimy to w oknie właściwości paska).

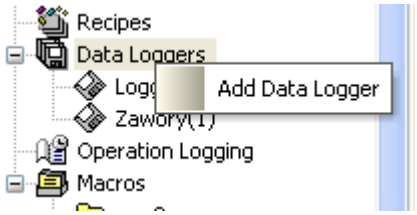


Konfiguracja wykresu Historic Trend Graph

DATA LOGGER

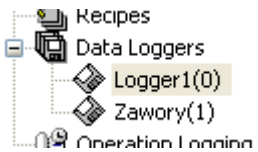
Służy do gromadzenia i przechowywania wartości bloku zmiennych. Można stworzyć do 16 na aplikację. Istnieje możliwość ustawienia częstotliwości próbkowania, typu pamięci do przechowywania informacji i sposobu ich zapisywania.

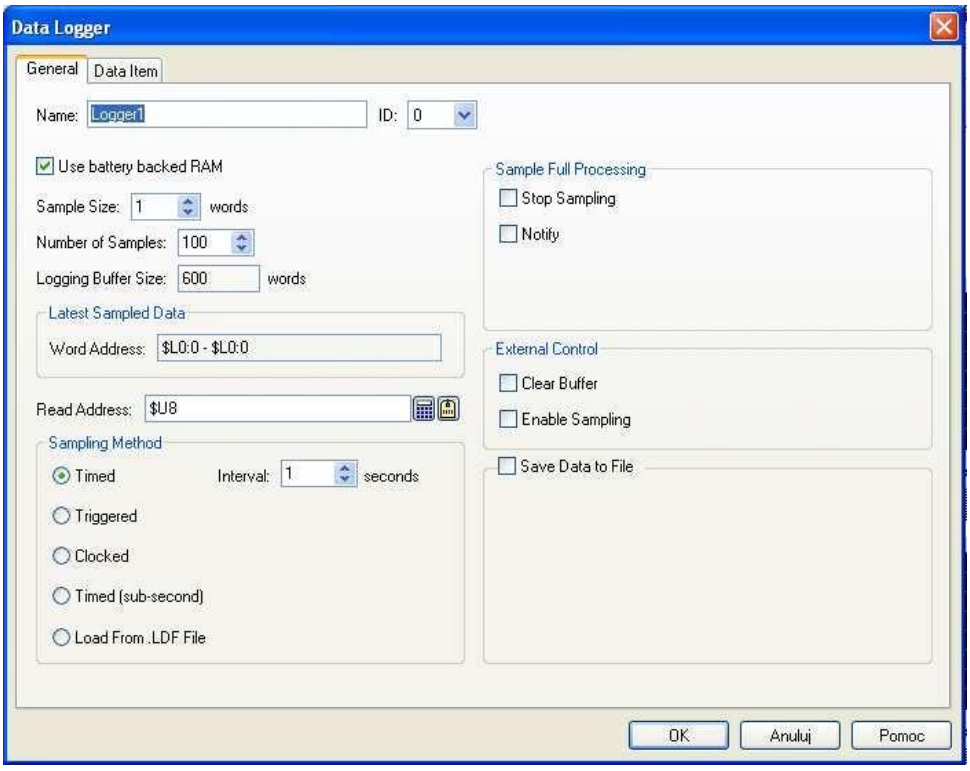
Aby stworzyć Data Logger klikamy prawym klawiszem myszy w Project Manager na Data Loggers.

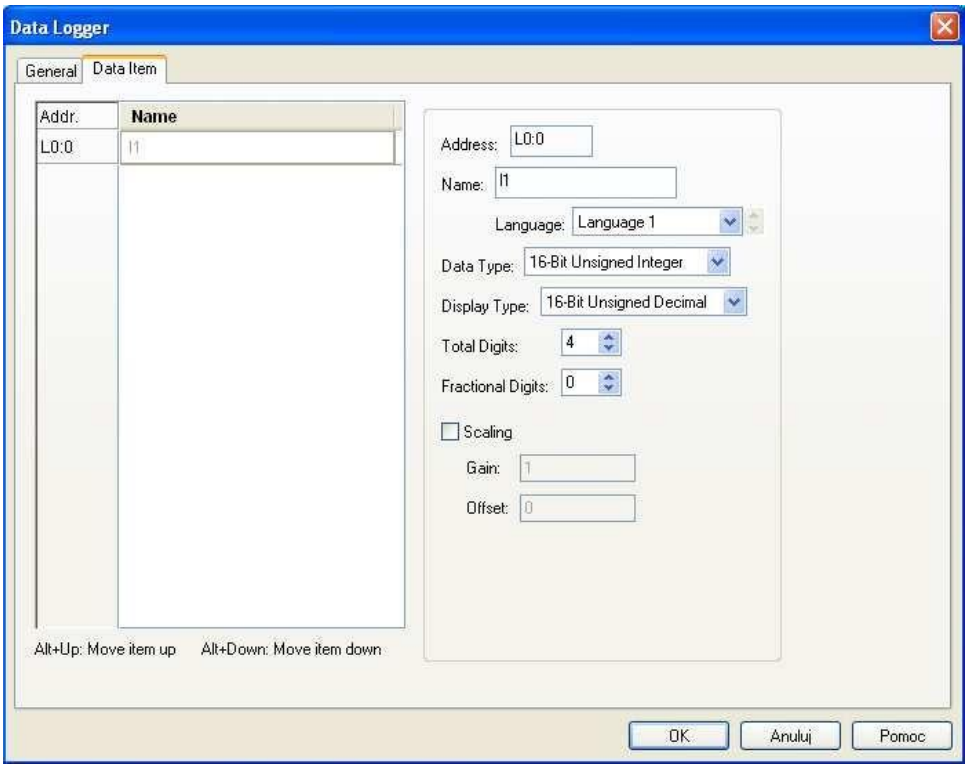


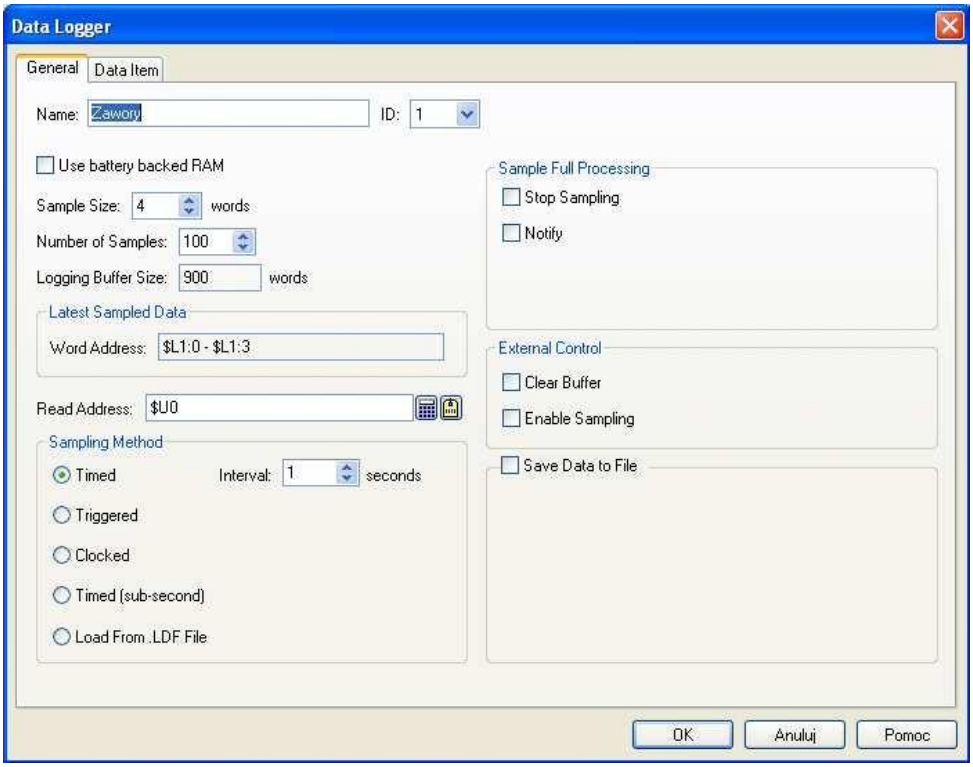
Następnie wybieramy *Add Data Logger*.

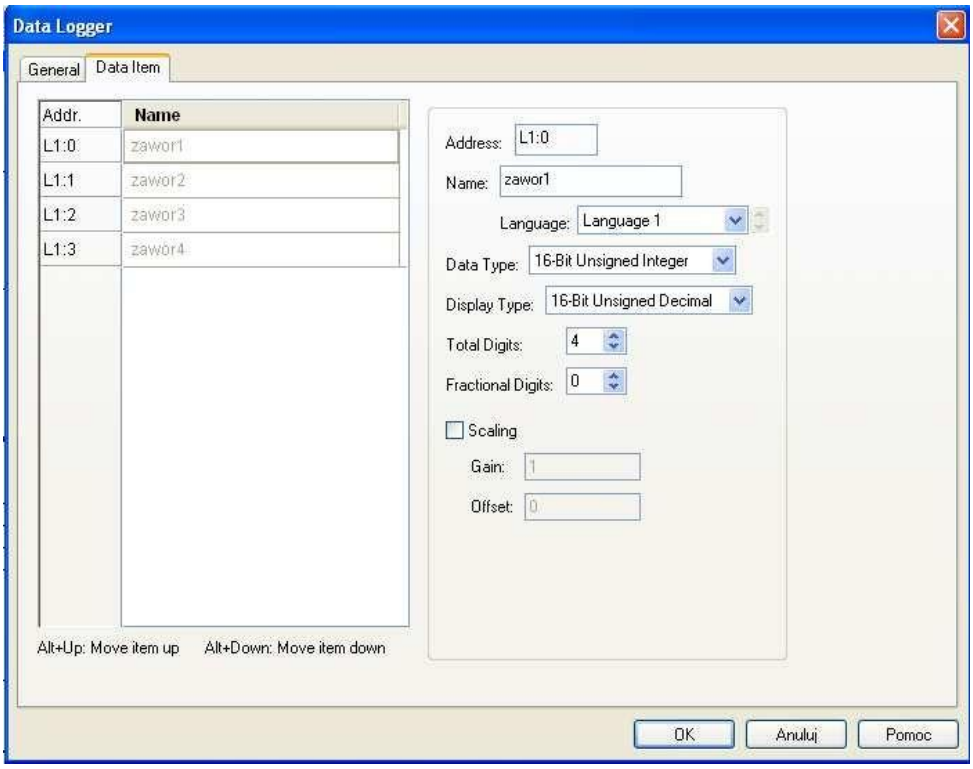
Potrzebne będą nam dwa takie elementy, które konfigurujemy następująco (podwójne kliknięcie na nazwę Data Logger'a w Project Manager):






Data Logger
General
Data Item
Addr.
Name
L0:0
I1
Address: L0:0
Name: I1
Language: Language 1
Data Type: 16-Bit Unsigned Integer
Display Type: 16-Bit Unsigned Decimal
Total Digits: 4
Fractional Digits: 0
Scaling
Gain: 1
Offset: 0
Alt+Up: Move item up
Alt+Down: Move item down
OK
Anuluj
Pomoc

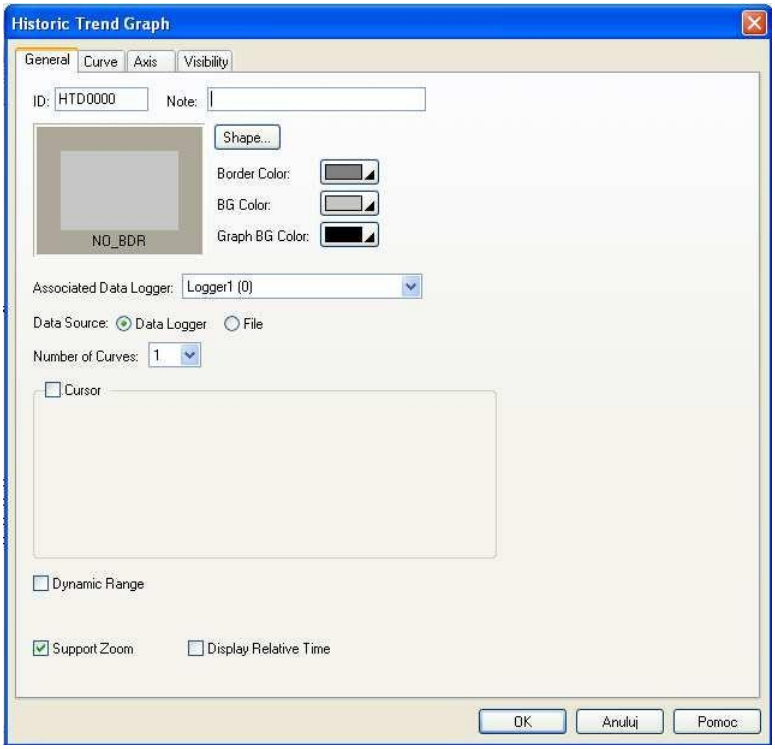

Data Logger
General
Data Item
Name: Zawory
ID: 1
Use battery backed RAM
Sample Size: 4 words
Number of Samples: 100
Logging Buffer Size: 900 words
Latest Sampled Data
Word Address: $L1:0 - $L1:3
Read Address: $U0
Sampling Method
Timed
Interval: 1 seconds
Triggered
Clocked
Timed (sub-second)
Load From .LDF File
Sample Full Processing
Stop Sampling
Notify
External Control
Clear Buffer
Enable Sampling
Save Data to File
OK
Anuluj
Pomoc



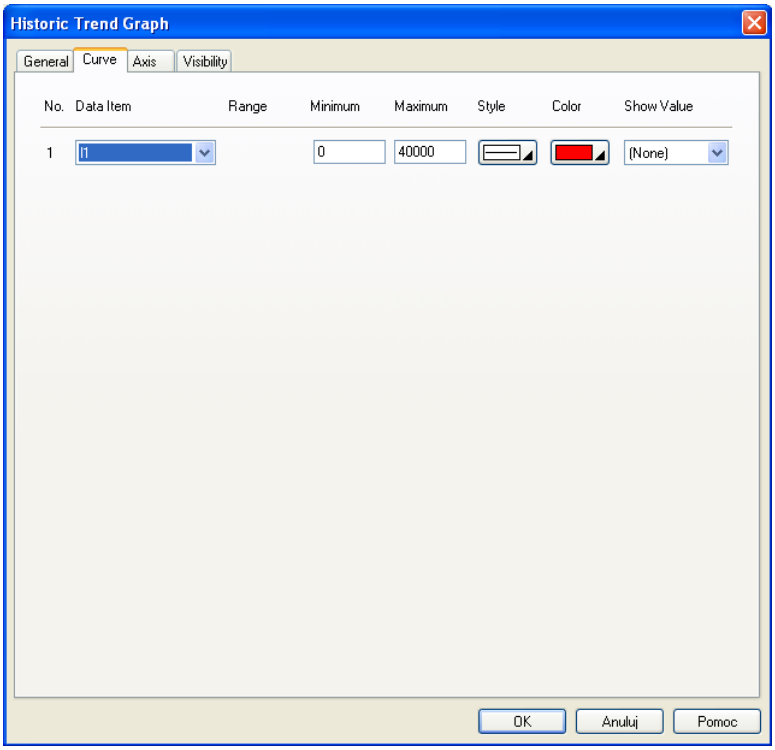
Teraz przechodzimy do okna właściwości Historic Trend Graph.

W polu *Associated Data Logger* wpisujemy Data Logger, w którym gromadzone są interesujące nas informacje. W polu Number of Curves wybieramy ile lini trendu (wykresów) chcemy mieć w tym obiekcie.

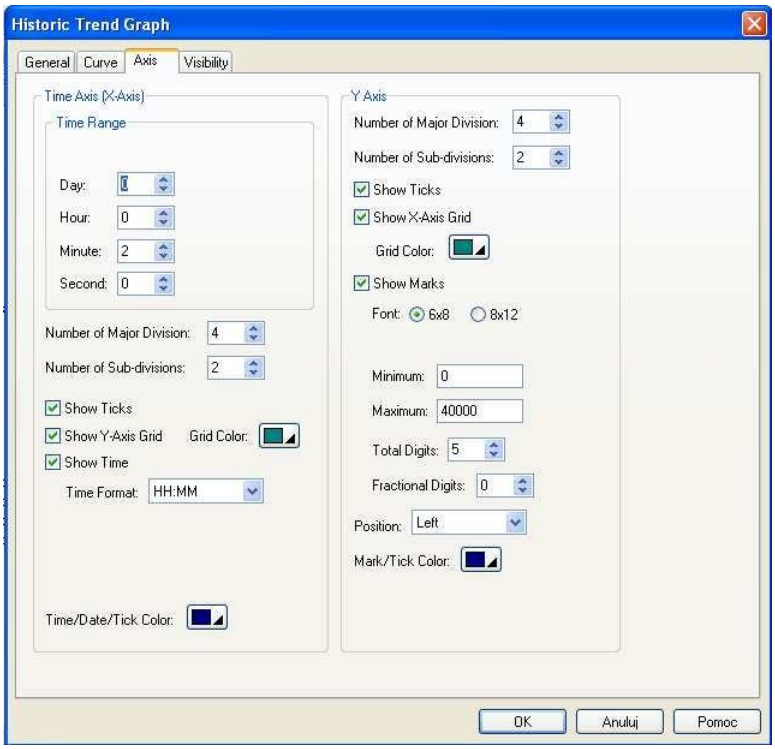
Konfiguracja dla pierwszego wykresu:



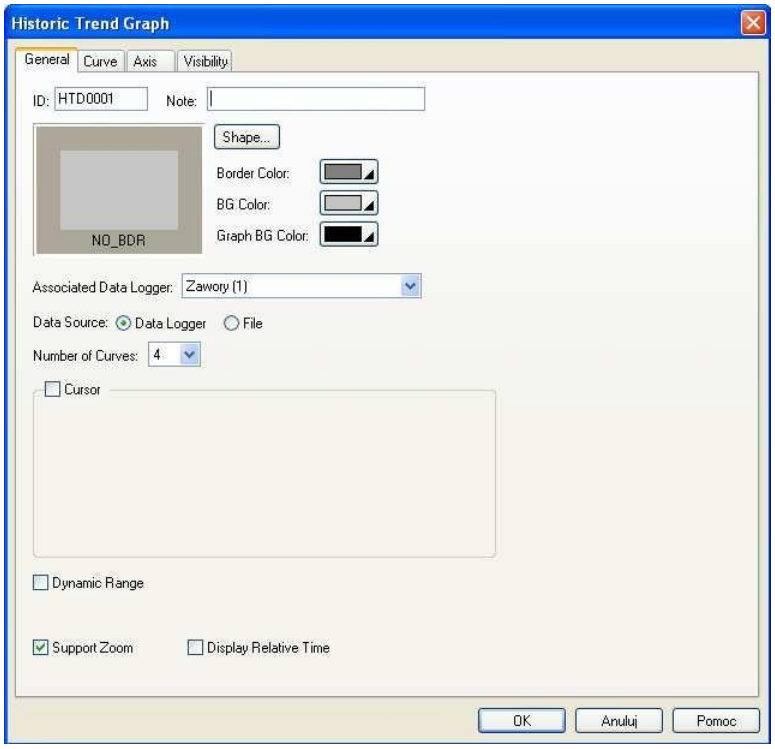
Konfiguracja linii:

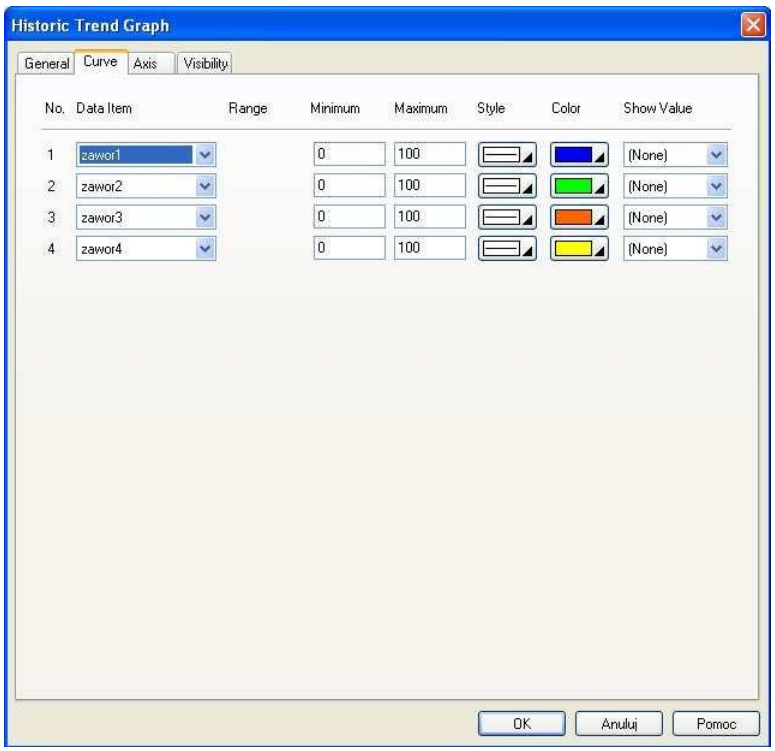


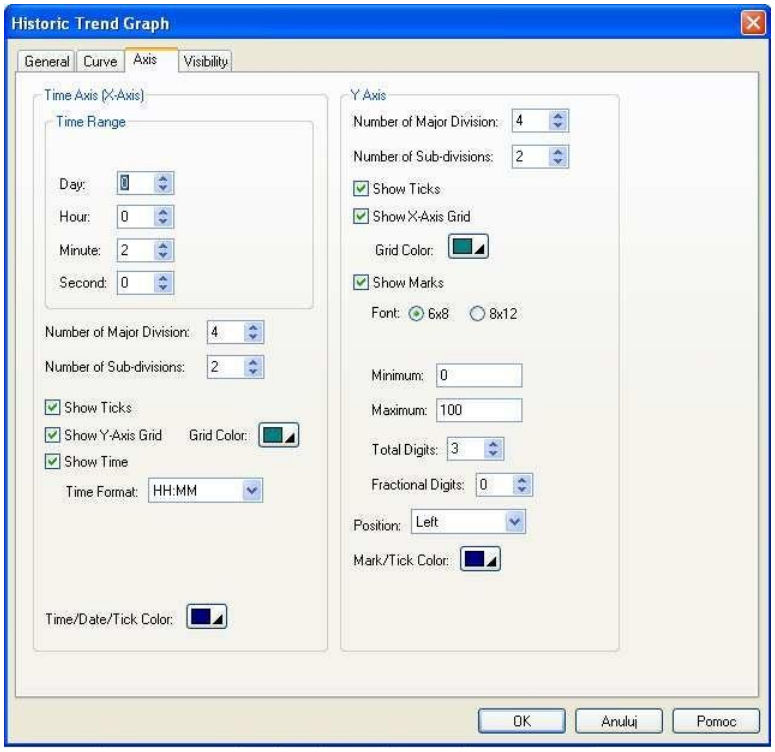
Przedział czasowy widziany w oknie wykresu ustawiamy w *Time Range*:



Konfiguracja dla drugiego wykresu:

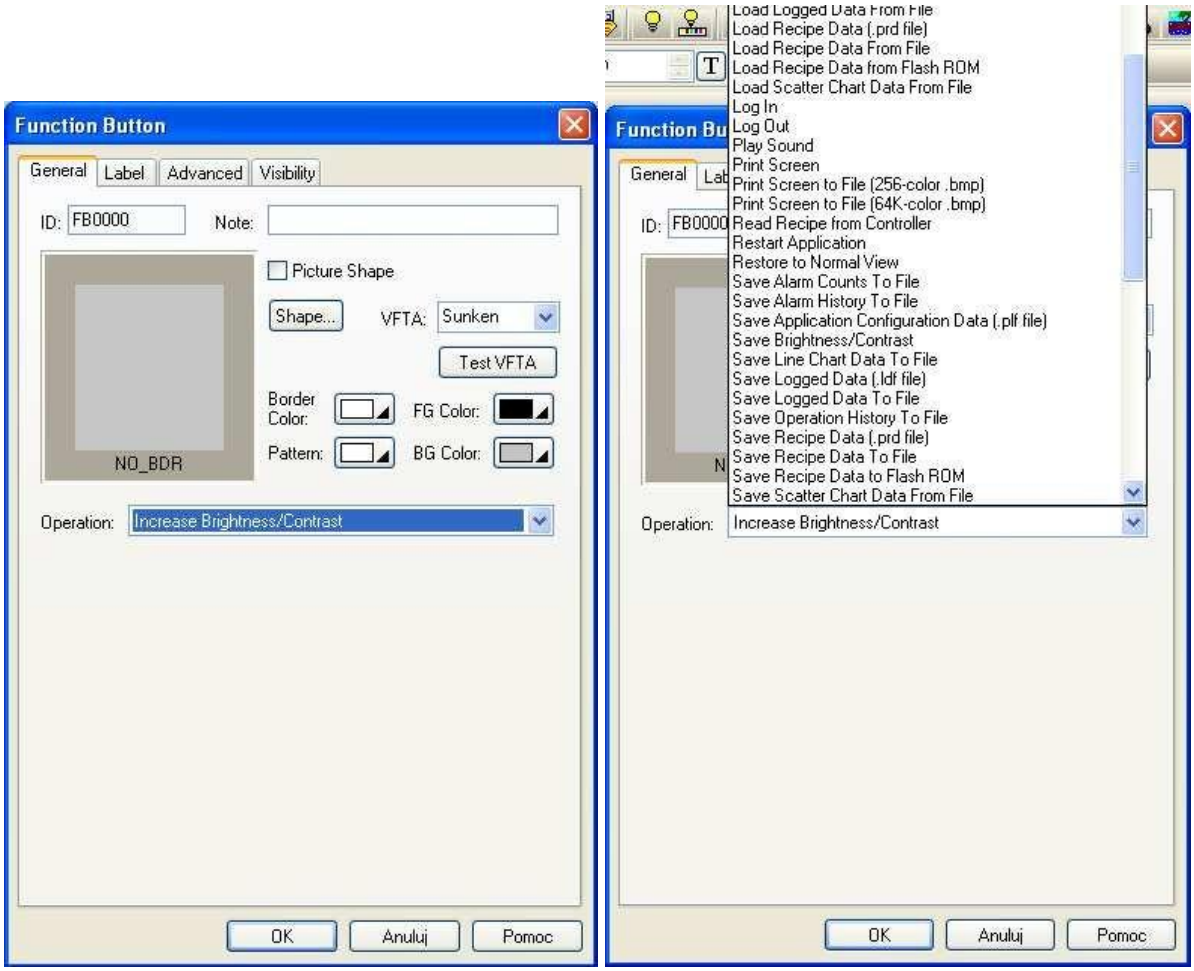



Historic Trend Graph
General
Curve
Axis
Visibility
No.
Data Item
Range
Minimum
Maximum
Style
Color
Show Value
1
zawor1
0
100
(None)
2
zawor2
0
100
(None)
3
zawor3
0
100
(None)
4
zawor4
0
100
(None)
OK
Anuluj
Pomoc


Historic Trend Graph
General
Curve
Axis
Visibility
Time Axis (X-Axis)
Time Range
Day:
Hour:
Minute:
Second:
Number of Major Division:
Number of Sub-divisions:
Show Ticks
Show Y-Axis Grid
Grid Color:
Show Time
Time Format:
HH:MM
Time/Date/Tick Color:
Y Axis
Number of Major Division:
Number of Sub-divisions:
Show Ticks
Show X-Axis Grid
Grid Color:
Show Marks
Font:
6x8
8x12
Minimum:
Maximum:
Total Digits:
Fractional Digits:
Position:
Left
Mark/Tick Color:
OK
Anuluj
Pomoc

Następnie tworzymy przyciski pozwalające nam powiększać i pomniejszać widok wykresu.

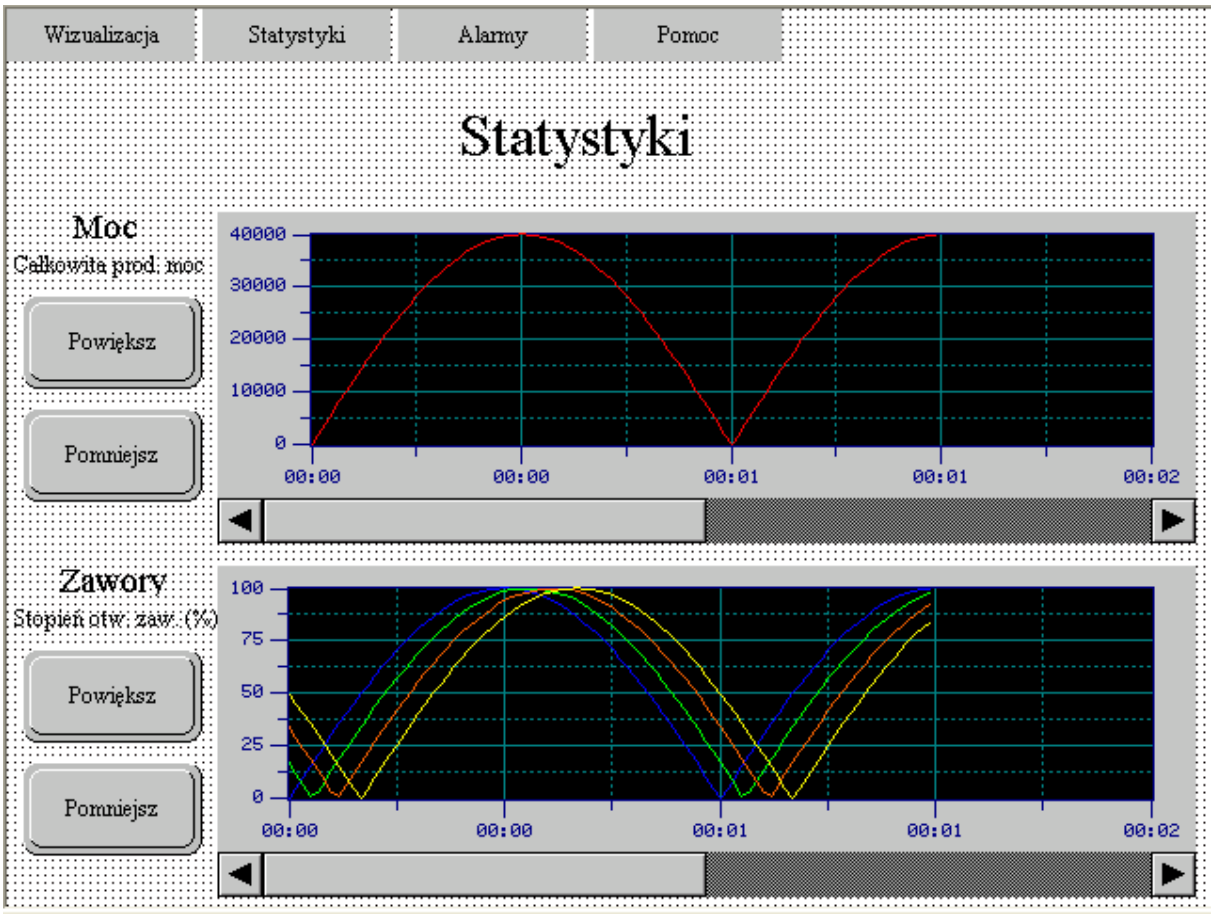
F1 Przycisk po kliknięciu którego zostaje wykonana wcześniej ustawiona w polu *Operation* funkcja.



Dla jednego z nich wybieramy *Zoom In*, a dla drugiego *Zoom Out*. W *Associated Object ID* wpisujemy ID odpowiedniego wykresu (możemy je sprawdzić we właściwościach wykresu). Tworzymy dwa komplety takich przycisków (po jednym dla każdego wykresu.

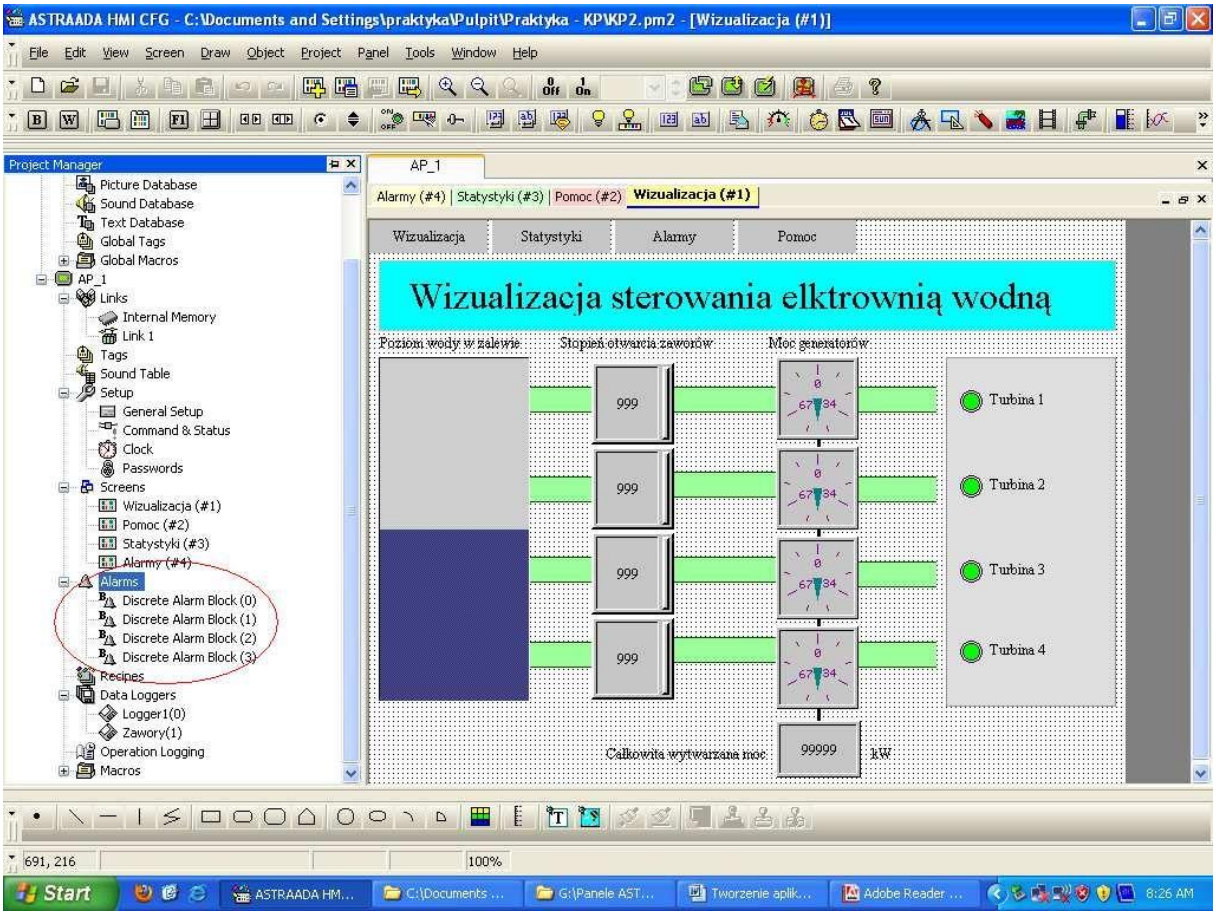
Na koniec dodajemy pola tekstowe z odpowiednimi opisami i wklejamy wcześniej przygotowane przyciski zmiany ekranu.

Tak ostatecznie ma wyglądać nasz ekran:

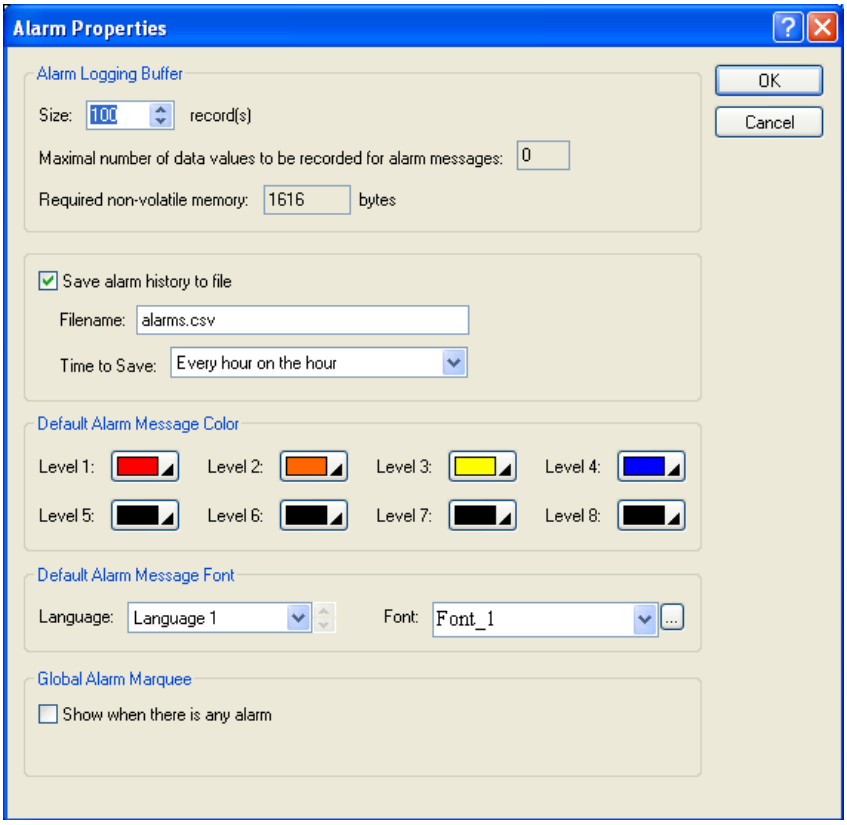


ALARMY

Aby móc używać alarmów najpierw trzeba ustawić *Alarm Properties*, a później zdefiniować *Alarm Block*.



Dwukrotne kliknięcie na *Alarms* przenosi nas do okna ustawień alarmów.

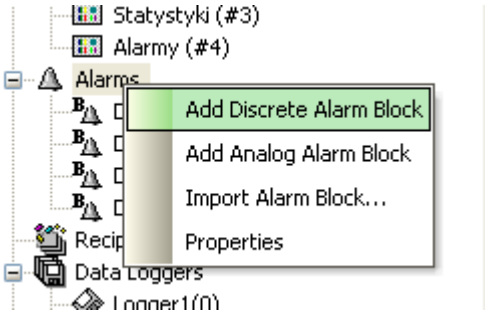


W oknie Size możemy ustawić ilość pamiętanych alarmów. Na przykład wartość 100 oznacza, że 101. alarm będzie zapisany w miejscu pierwszego.

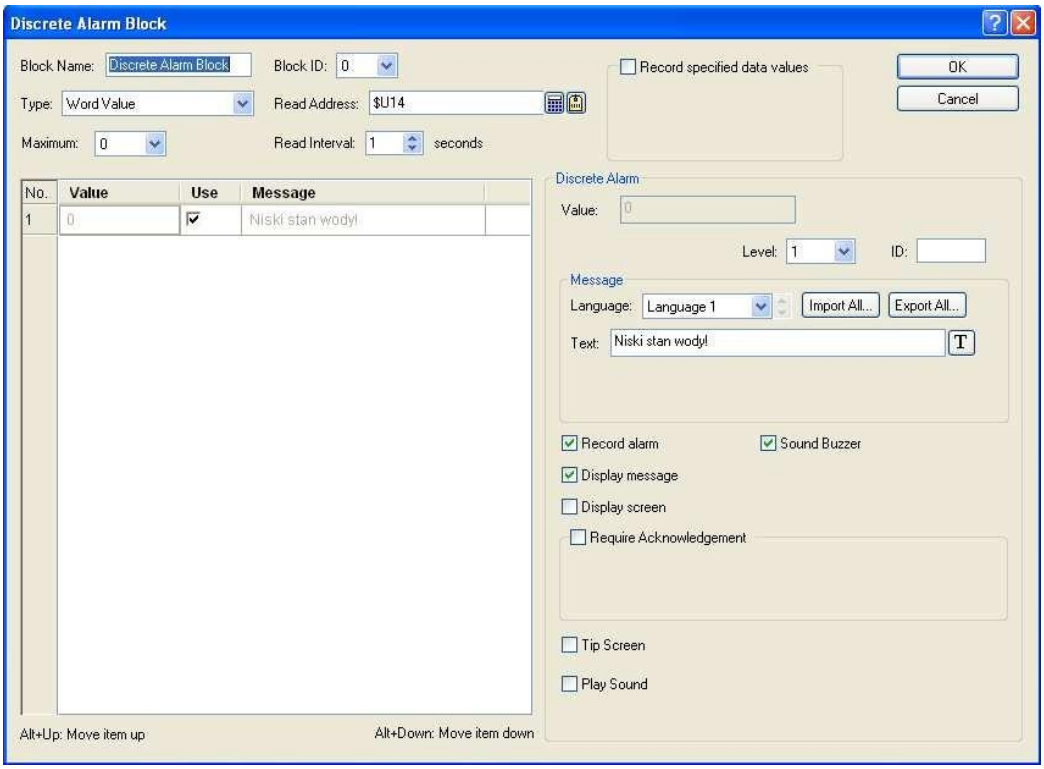
Zaznaczając opcję Save alarm history to file sprawimy, że informacje o alarmach będą zapisywane w pliku, którego nazwę podamy w polu Filename. Pod spodem możemy ustawić co jaki czas dane o alarmach będą zapisywane.
Poniżej ustawiamy kolor w jakim będą się wyświetlały alarmy określonych poziomów.

Do naszej symulacji potrzebujemy czterech dyskretnych bloków alarmowych. Będą odpowiadały za informację o niskim i wysokim stanie wody oraz o awaryjnym otwarciu i zamknięciu zaworów. Przyjmijmy, że zamykać będziemy zawory równocześnie z informacją o niskim stanie wody, a otwierać wraz z informacją o wysokim stanie wody w zbiorniku.

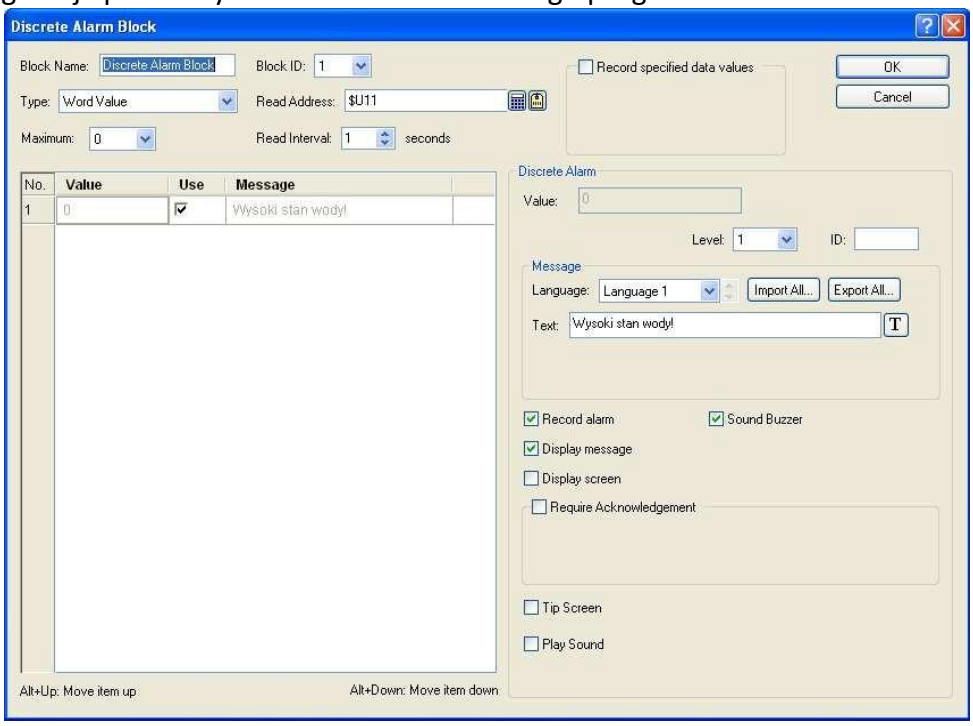
Klikamy prawym przyciskiem na Alarms i wybieramy opcję Add Discrete Alarm Block.

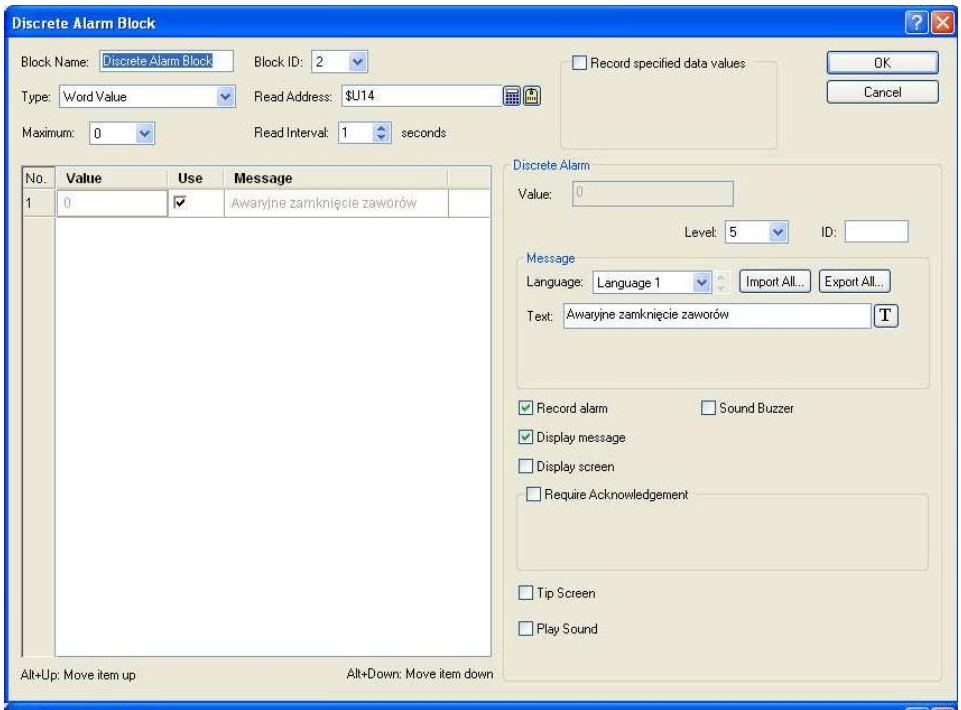


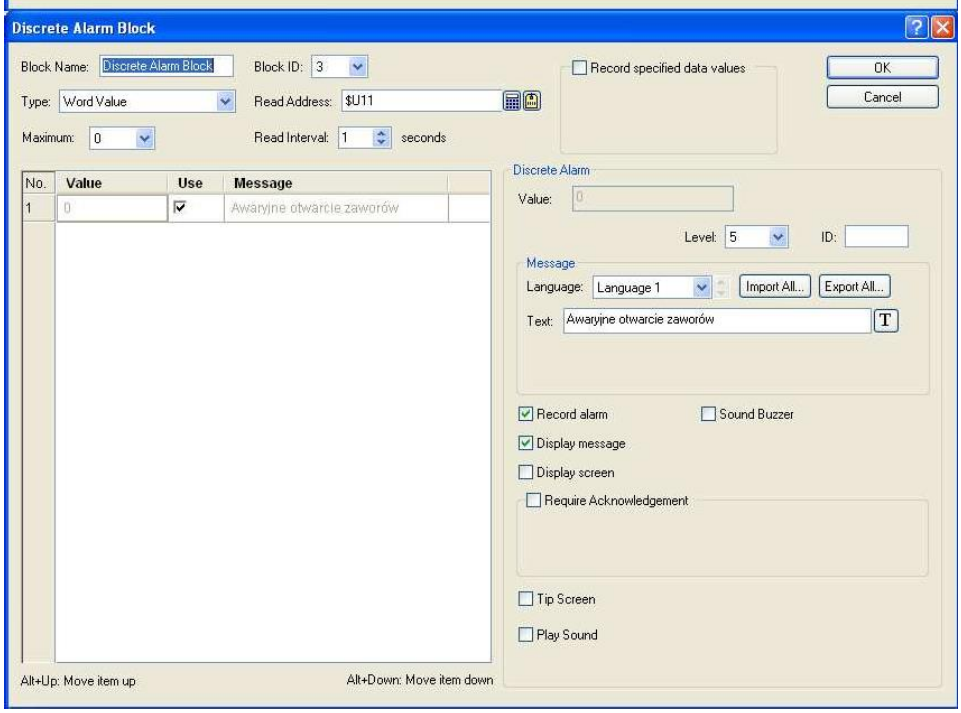
Po kliknięciu na stworzony przez nas alarm pokazuje się okno:



W polu Type wybieramy typ monitorowanej przez nas zmiennej (w naszym przypadku Word Value). W polu Read Address umieszczamy adres tej zmiennej. W polu Text wpisujemy treść wyświetlanego alarmu. Alarm będzie uaktywniony zawsze, gdy zmienna przyjmie wartość ‘0’. Oto konfiguracja pozostałych trzech alarmów naszego programu:



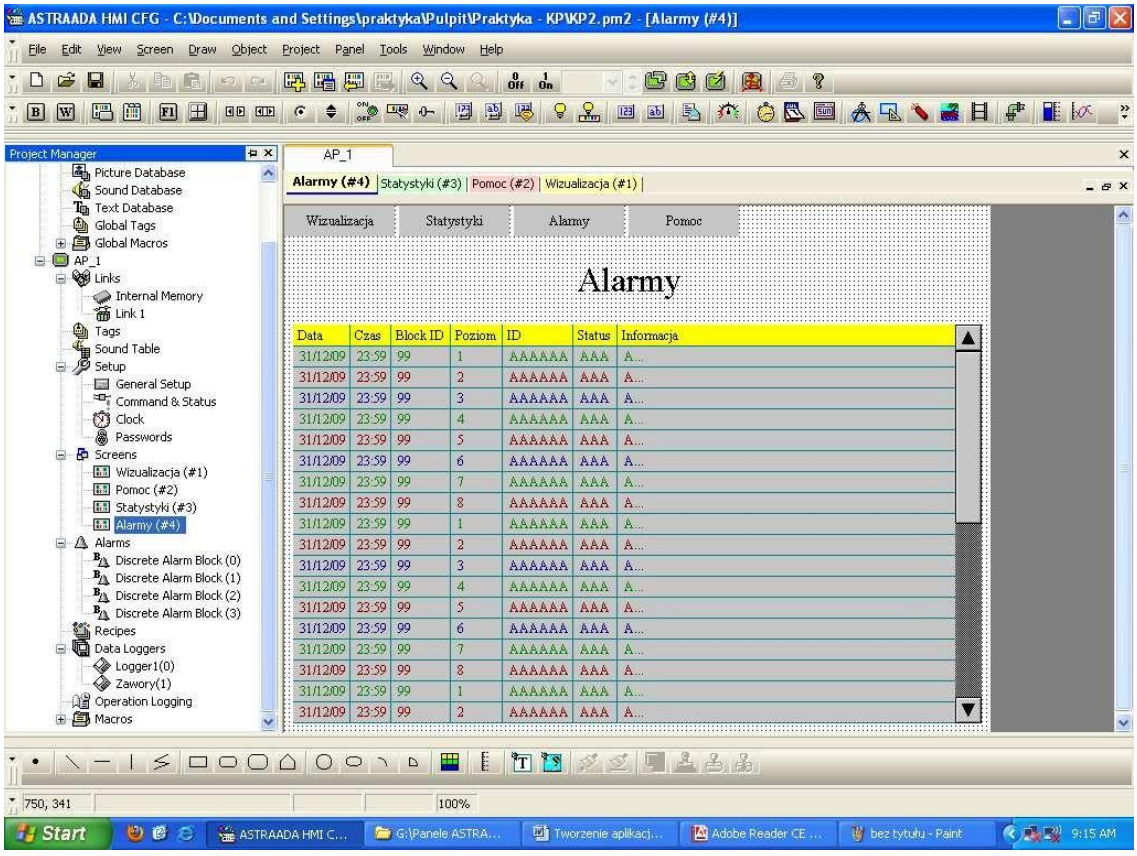

Discrete Alarm Block
Block Name: Discrete Alarm Block
Block ID: 2
Type: Word Value
Read Address: $U14
Maximum: 0
Read Interval: 1 seconds
Record specified data values
OK
Cancel
No. Value Use Message
1 0 Awaryjne zamknięcie zaworów
Discrete Alarm
Value: 0
Level: 5
ID:
Message
Language: Language 1
Import All...
Export All...
Text: Awaryjne zamknięcie zaworów
Record alarm
Sound Buzzer
Display message
Display screen
Require Acknowledgement
Tip Screen
Play Sound
Alt+Up: Move item up
Alt+Down: Move item down


Discrete Alarm Block
Block Name: Discrete Alarm Block
Block ID: 3
Type: Word Value
Read Address: $U11
Maximum: 0
Read Interval: 1 seconds
Record specified data values
OK
Cancel
No. Value Use Message
1 0 Awaryjne otwarcie zaworów
Discrete Alarm
Value: 0
Level: 5
ID:
Message
Language: Language 1
Import All...
Export All...
Text: Awaryjne otwarcie zaworów
Record alarm
Sound Buzzer
Display message
Display screen
Require Acknowledgement
Tip Screen
Play Sound
Alt+Up: Move item up
Alt+Down: Move item down

ALARMY HISTORYCZNE

Aby wyświetlać **historię alarmów** w naszym panelu musimy odnaleźć blok  . Umieszczamy do w oddzielnym ekranie (klikamy prawym klawiszem na Screens w Project Manager, wybieramy New Screen). Następnie wybieramy *pasek przewijania*  i powiązujemy go z wyświetlaczem alarmów w oknie jego właściwości.

Dodajemy opis ekranu ikonką  oraz przyciski zmiany ekranów skopiowane z poprzednich ekranów.

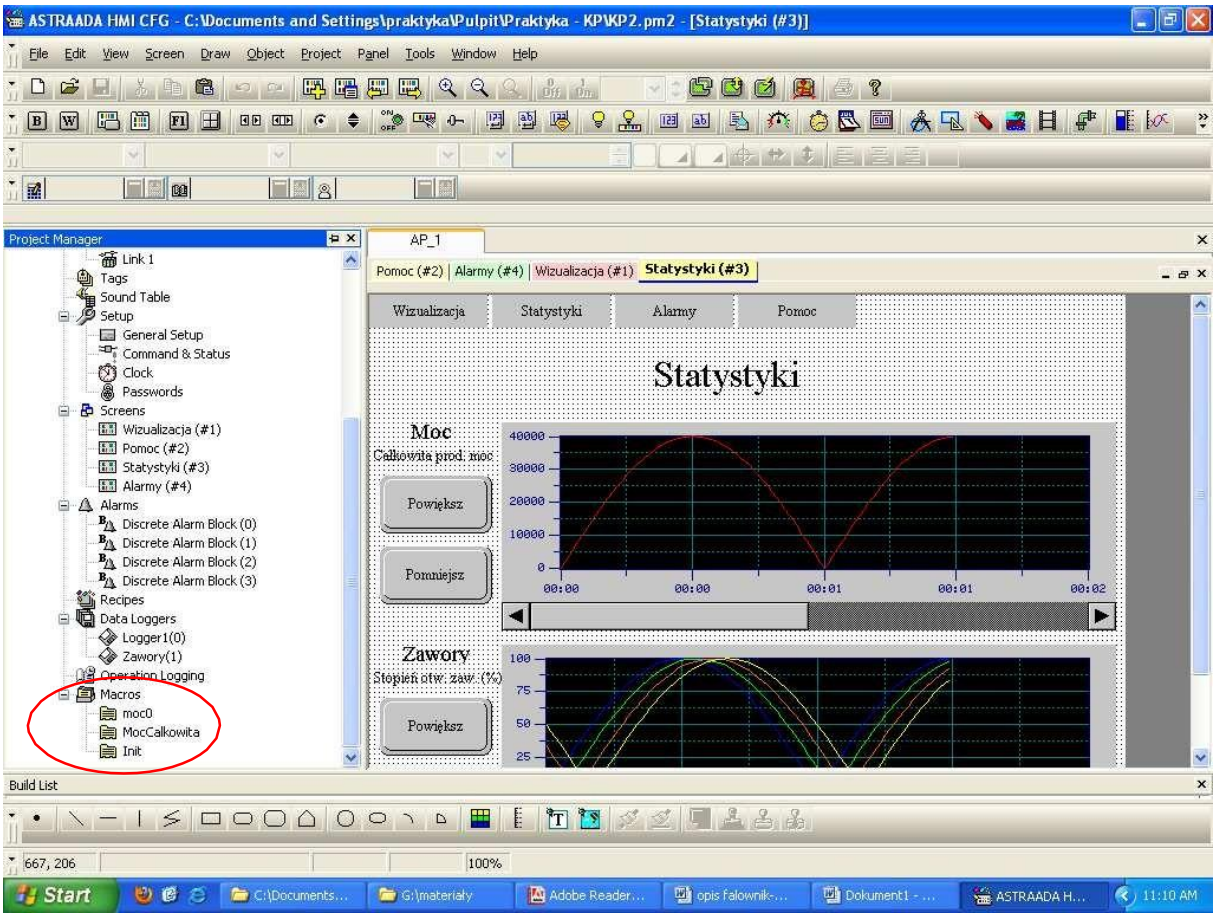


POMOC

Ostatni ekran to u nas ekran pomocy. Wpisujemy w nim w formie tekstowej informacje potrzebne do eksploatacji naszej aplikacji.

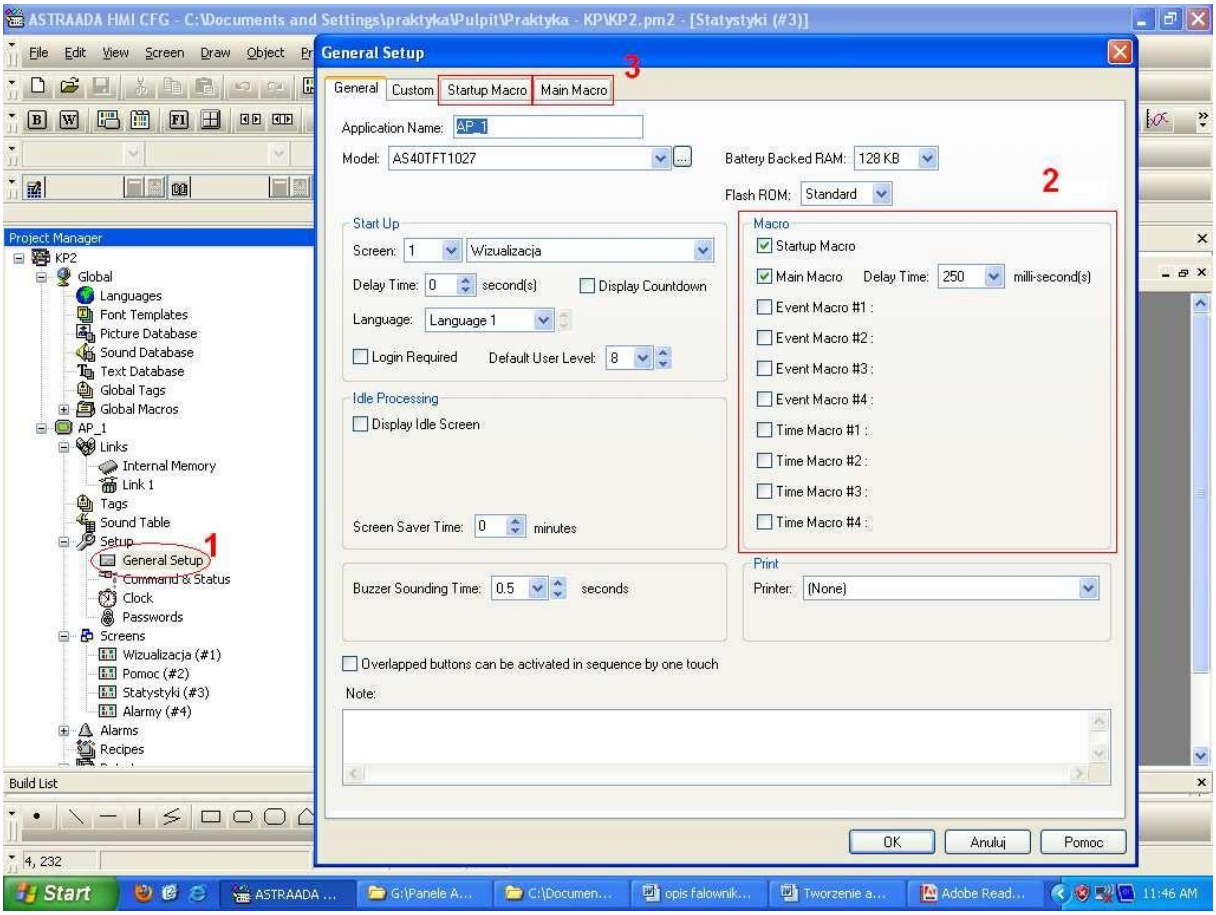
MAKRA

Skrypty można pisać w makrach. Są bardzo użyteczne, gdy chcemy stworzyć własną funkcję. Do dyspozycji mamy kilka rodzajów makr. Wyszczególnię i opiszę najważniejsze z nich.



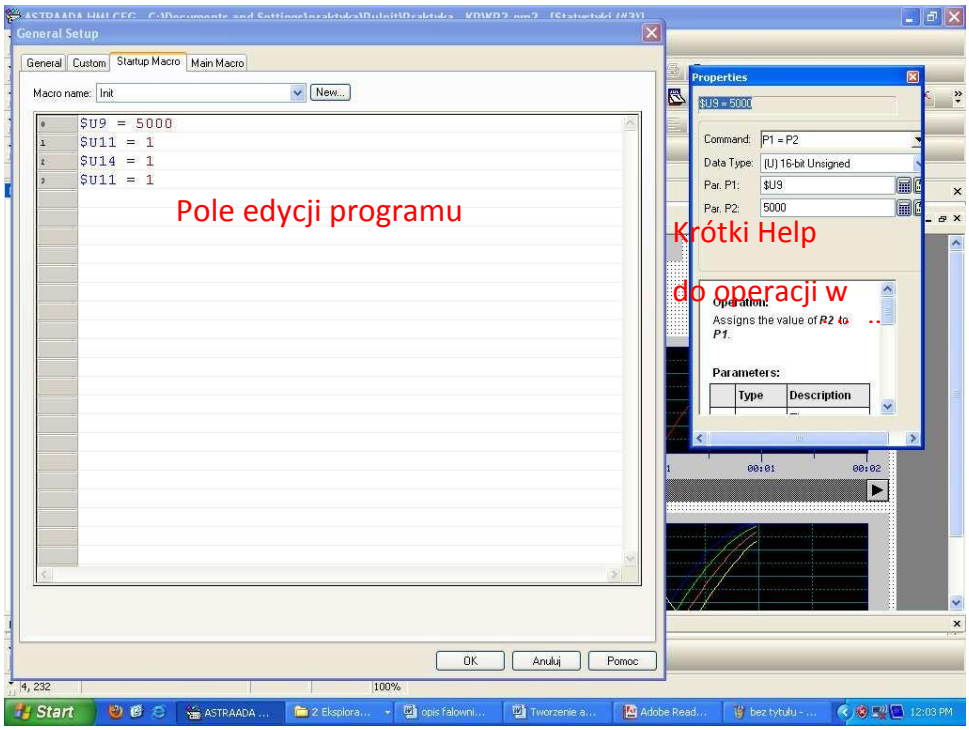
Sturtup Macro, Main Macro, Event Macro, Time Macro

Ustawiać i edytować te makra możemy w *General Setup* (**1**). Po dwukrotnym kliknięciu otwiera nam się okno, w którym po prawej stronie (**2**) musimy zaznaczyć, które makra będziemy tworzyć. Gdy już zaznaczymy interesujące nas makra pojawiają się nam nowe zakładki (**3**), w których możemy napisać interesujące nas funkcje.



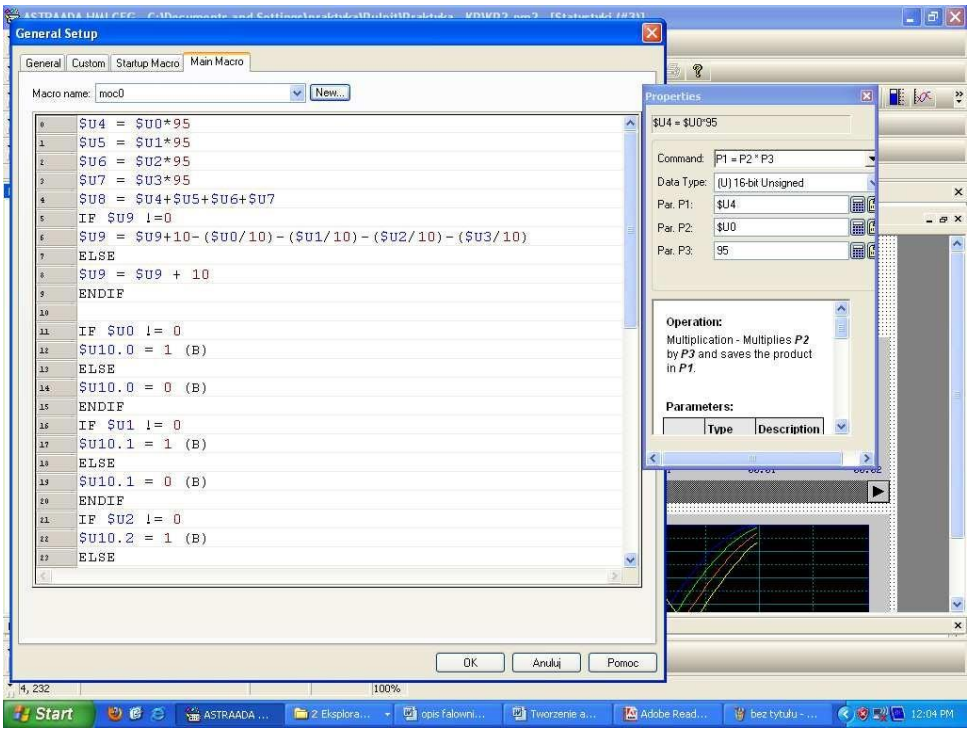
Startup Macro

Wykonuje się wyłącznie przy starcie aplikacji. Przydaje się do inicjalizacji zmiennych.



Main Macro

Wykonuje się przez cały czas działania aplikacji. Jest wykonywane cyklicznie od pierwszej instrukcji. Cykl kończy się na ostatniej instrukcji, bądź poleceniu END.



Event Macro

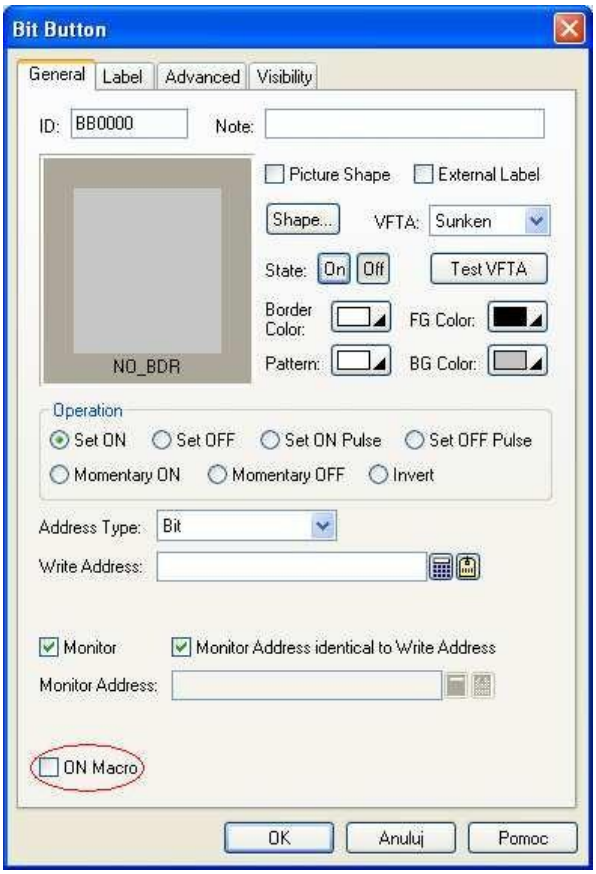
Jest uruchamiane gdy związany z nim bit (trigger bit czyli bit wyzwolenia) zmienia się z ‘0’ na ‘1’. Aplikacja może mieć do czterech Event Macro. Funkcje tworzymy również w oknie General Setup.

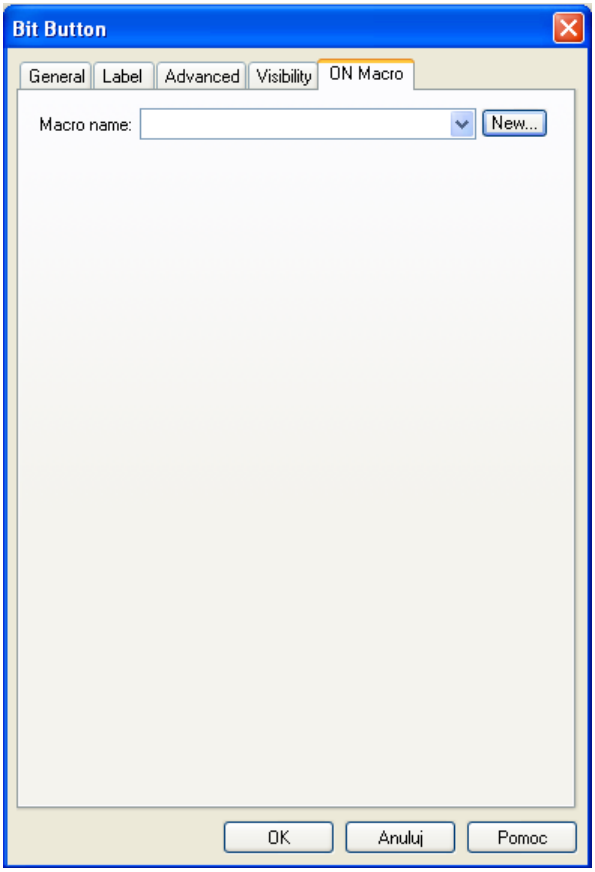
Time Macro

Jest uruchamiane periodycznie, zgodnie z ustawionym interwałem czasowym. Aplikacja może mieć do czterech Time Macro. Funkcje tworzymy również w oknie General Setup.

Object Macro

Jest uruchamiane (jedno wywołanie) gdy jest aktywowany obiekt powiązany z tym makrem. Object Macro może być używane w: Screen Button, Function Button, Keypad Button. Napisać makro możemy w polu konfiguracji odpowiedniego przycisku.





Po zaznaczeniu pola ON Macro pojawia się zakładka ON Macro (na rysunku powyżej). Można w niej podpiąć do przycisku istniejące już makro lub zdefiniować nowe.
Gdy to zrobimy pojawia się pole edycji makra (nie musimy się do niego dostawać przez Project Manager).

Na koniec warto dodać, że możemy tworzyć zarówno *globalne* jak i *lokalne makra*.

JĘZYK PROGRAMOWANIA

Język programowania nie zaskakuje niczym nowym. Przedstawiam tutaj podstawowe informacje na temat tworzenia programu zawarte w angielskiej dokumentacji technicznej.

Menu Command	Key Combination	Description
Cut	CTRL+X	Removes selected text from the active macro editor window.
Copy	CTRL+C	Duplicates selected text in the active macro editor window.
Paste	CTRL+V	Pastes cut or copied text into an active macro editor window.
	DELETE	Deletes text without copying it to the Clipboard.
Undo	CTRL+Z	Reverses the last editing action.
Redo	CTRL+Y	Reapplies the prior editing that have been undone.
	CTRL+A	Selects all texts in the active macro editor

Operators	Name or Meaning	Grouping	Used for
( )	Parentheses	Left to right	AE/CE
*	Multiplication	Left to right	AE
/	Division	Left to right	
%	Modulus	Left to right	
+	Addition	Left to right	
-	Subtraction	Left to right	
<<	Left shift	Left to right	
>>	Right shift	Left to right	
<	Less than	Left to right	CE
>	Greater than	Left to right	
<=	Less than or equal to	Left to right	
>=	Greater than or equal to	Left to right	
==	Equality	Left to right	
!=	Inequality	Left to right	
&	Bitwise AND	Left to right	AE
^	Bitwise exclusive OR	Left to right	
|	Bitwise inclusive OR	Left to right	
&&	Logical AND	Left to right	CE
||	Logical OR	Left to right	CE
=	Assignment	Right to left	AE/CE

Example 1

```
IF $U10 == 0
    JMP SKIP /* Skip the command "$U20 = $U10 / 2" */
ENDIF
$U20 = $U10 / 2
SKIP:
$U10 = 1
```

Example 1	Example: $U1 = 10 $U2 = 12 FOR $U1 $U100 = $U100 + 1 /* This command will be executed 10 times. */ FOR $U2 $U200 = $U200 + 1 /* This command will be executed 120 times. */ NEXT NEXT

PROGRAMY WYKORZYSTYWANE PRZEZ APLIKACJĘ

Strartup Macro

Tutaj inicjalizujemy zmienne.

```
$U9 = 5000
$U11 = 1
$U14 = 1
$U11 = 1
```

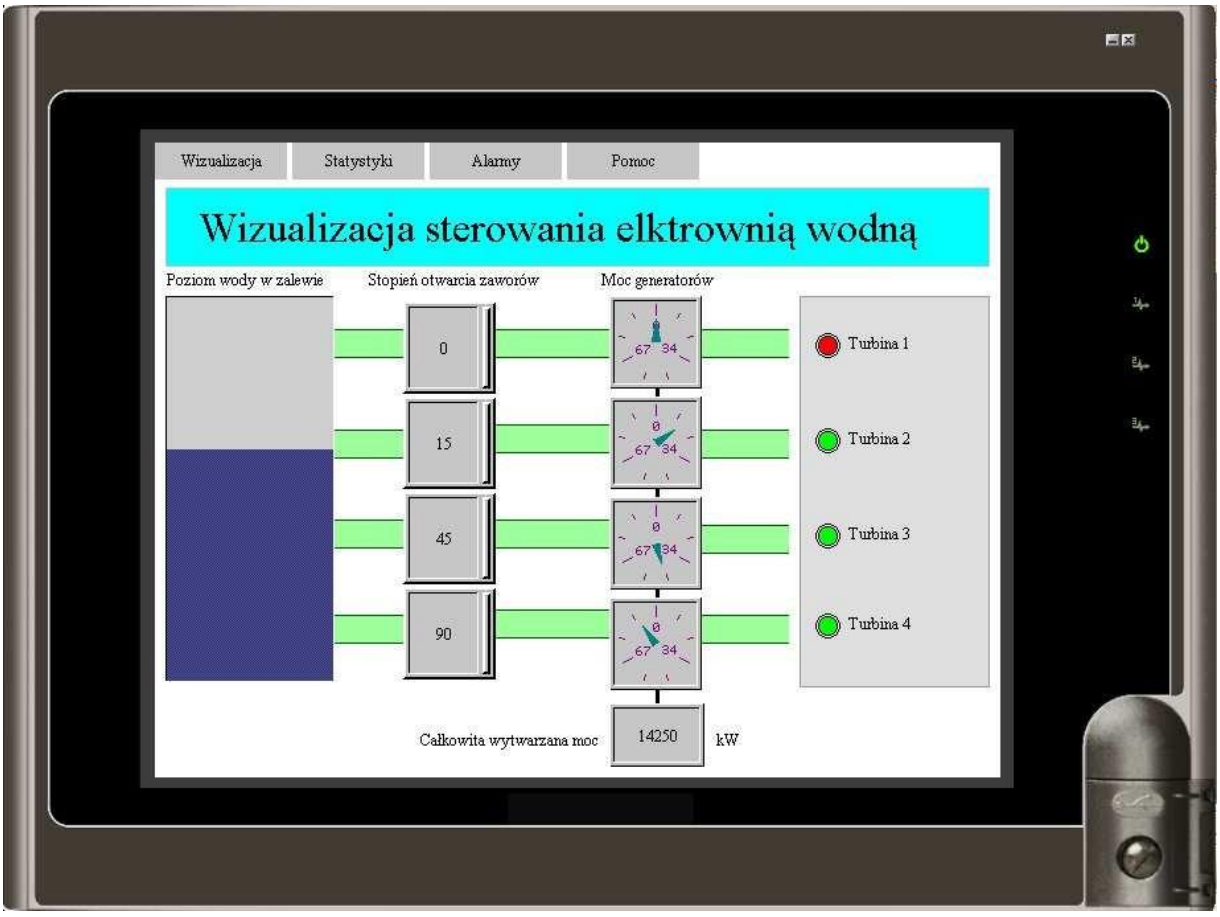
Main Macro

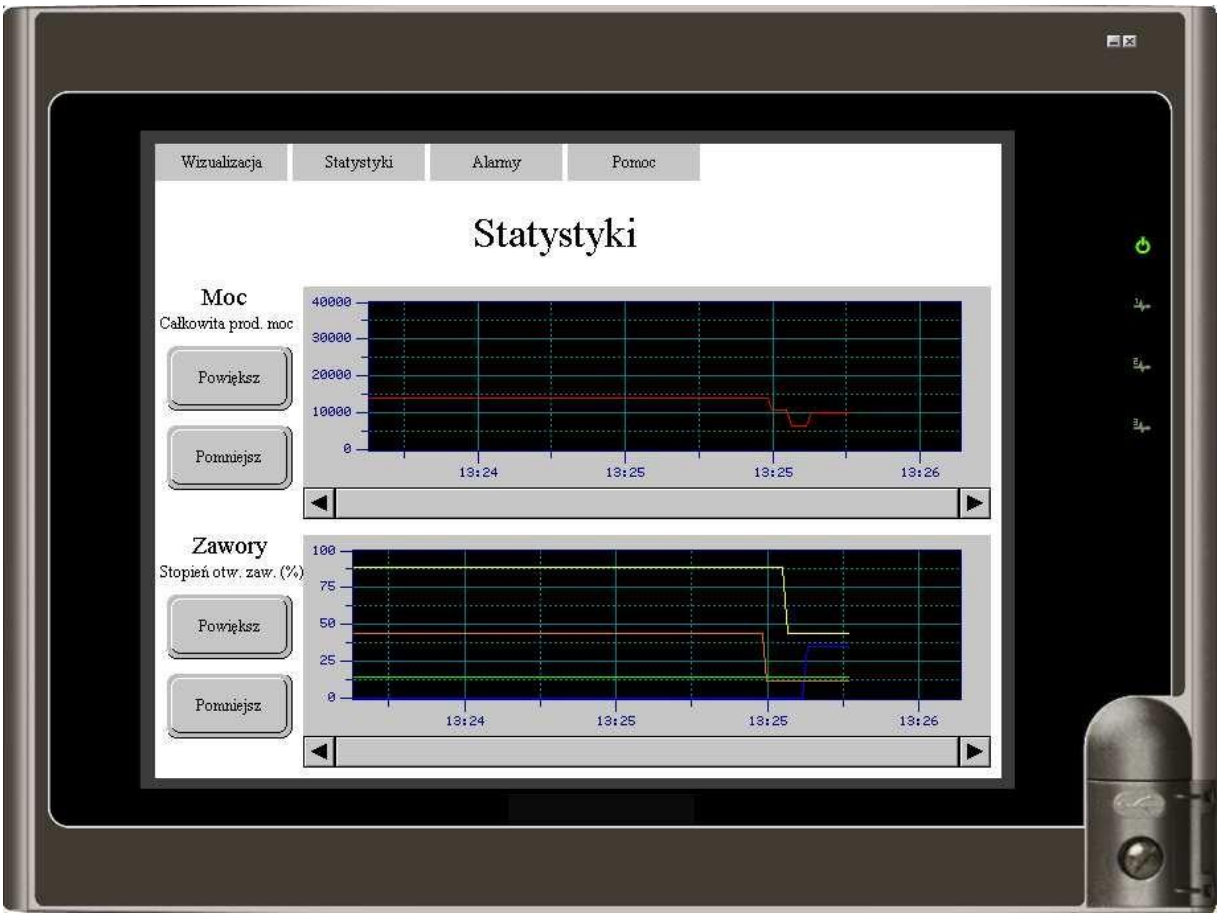
Ten program wykonuje się cały czas w ciągu działania aplikacji

```
$U4 = $U0*95  // umowne przeliczenie % otwarcia zaworu na generowaną moc
$U5 = $U1*95
$U6 = $U2*95
$U7 = $U3*95
$U8 = $U4+$U5+$U6+$U7 // całkowita generowana moc
IF $U9 !=0 // poziom wody nie może spadać poniżej zera
$U9 = $U9+10-($U0/10)-($U1/10)-($U2/10)-($U3/10)
ELSE
$U9 = $U9 + 10
ENDIF

IF $U0 != 0   // Ustawianie kontrolnych bitów pracy turbin
$U10.0 = 1 (B)
ELSE
$U10.0 = 0 (B)
ENDIF
IF $U1 != 0
$U10.1 = 1 (B)
ELSE
$U10.1 = 0 (B)
ENDIF
```

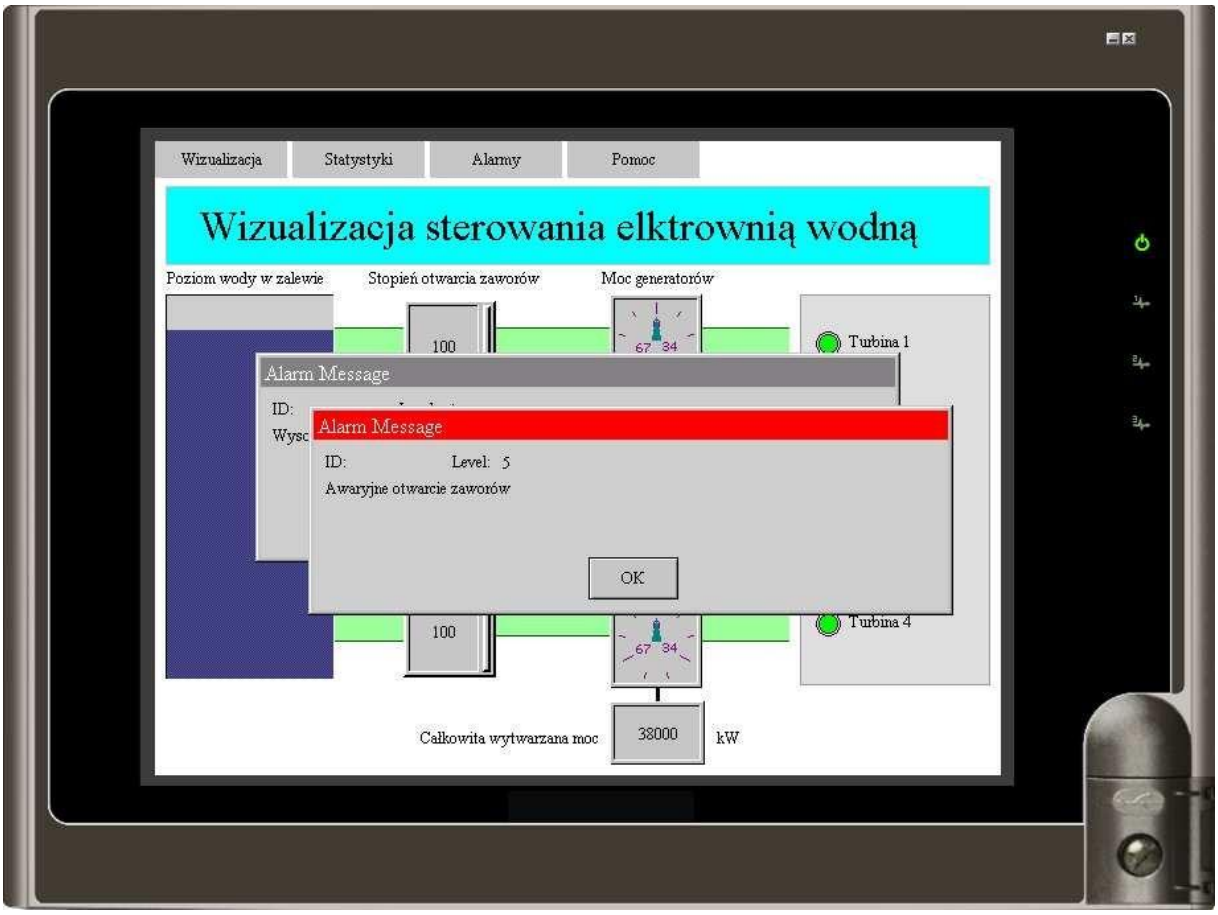
```
IF $U2 != 0
$U10.2 = 1 (B)
ELSE
$U10.2 = 0 (B)
ENDIF
IF $U3 != 0
$U10.3 = 1 (B)
ELSE
$U10.3 = 0 (B)
ENDIF

IF $U9 >= 9500   // awaryjne otwieranie zaworów
$U11 = 0
$U0 = 100
$U1 = 100
$U2 = 100
$U3 = 100
ELSE
$U14 = 1
ENDIF
IF $U9 <= 401  // awaryjne zamykanie zaworów
$U14 = 0
$U0 = 0
$U1 = 0
$U2 = 0
$U3 = 0
ELSE
$U14 = 1
ENDIF
```

EKRANY UKOŃCZONEJ I DZIAŁAJĄCEJ APLIKACJI




Wizualizacja
Statystyki
Alarmy
Pomoc
Statystyki
Moc
Całkowita prod. moc
Powiększ
Pomniejsz
40000
30000
20000
10000
0
13:24
13:25
13:25
13:26
Zawory
Stopień otw. zaw. (%)
Powiększ
Pomniejsz
100
75
50
25
0
13:24
13:25
13:25
13:26


Wizualizacja
Statystyki
Alarmy
Pomoc
Pomoc
Wizualizacja przedstawia schemat elektrowni wodnej.
Po lewej stronie znajduje się ikonka symbolizująca stan wody w zalewie. Gdy spadnie do zera nie jest możliwa dalsza praca elektrowni.
Stopień otwarcia zaworow to pole w którym mozna zadać wielkość przepływu (od 0 do 100 %), która bezpośrednio przekłada sie na generowaną moc.



Wyskakujące komunikaty alarmowe

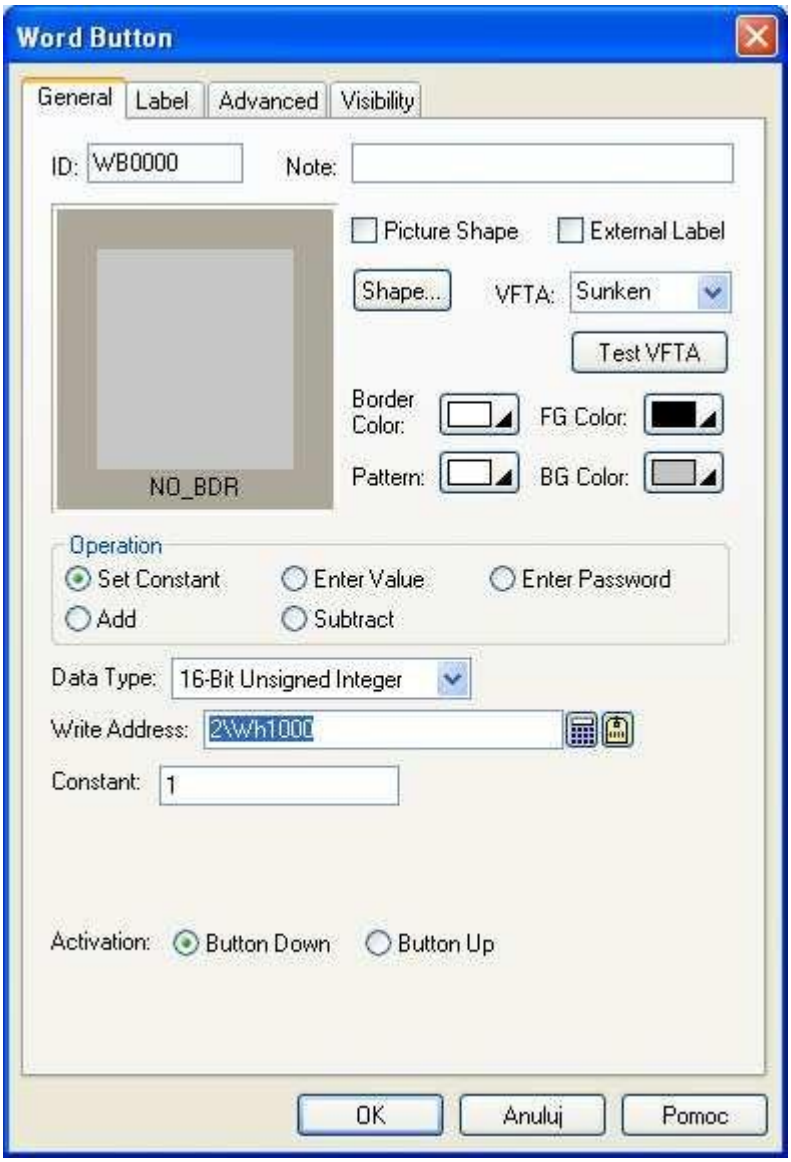

Wizualizacja
Statystyki
Alarmy
Pomoc
Alarmy
Data Czas Block ID Poziom ID Status Informacja
05/08/10 13:31 3 5 A Awaryjne otwarcie zaworów
05/08/10 13:31 1 1 A Wysoki stan wody!


Wizualizacja
Statystyki
Alarmy
Pomoc
Alarmy
Data Czas Block ID Poziom ID Status Informacja
05/08/10 13:33 2 5 C Awaryjne zamknięcie zaworów
05/08/10 13:33 0 1 C Niski stan wody!
05/08/10 13:33 2 5 A Awaryjne zamknięcie zaworów
05/08/10 13:33 0 1 A Niski stan wody!
05/08/10 13:31 3 5 A Awaryjne otwarcie zaworów
05/08/10 13:31 1 1 A Wysoki stan wody!

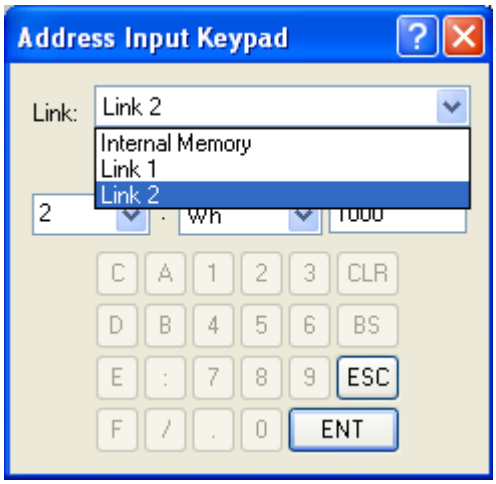
DODATEK

Word Button

Przycisk służący do zadawania stałej wartości. Po dwukrotnym kliknięciu otwierają się właściwości.
W polu Write Address wpisujemy adres zmiennej. Po kliknięciu na ikonkę obok tego pola otwiera nam się pomocne okienko Address Input Keypad. UWAGA Po wybraniu opcji



Internal Memory możemy adresować jedynie zmienne wewnątrz programu. Gdy chcemy zmieniać parametry urządzenia sterowanego i odnosimy się do jego własnych adresów musimy wybrać połączenie z tym urządzeniem wcześniej przez nas skonfigurowane.



W zakładce Label możemy zmienić napis na naszym przycisku oraz kilka podstawowych parametrów graficznych.
W polu Constant podajemy wartość jaka zostanie przypisana zmiennej wskazanej przez pole Write Address po naciśnięciu przycisku