

PANELE OPERATORSKIE WOP-2000

WebOP Designer 2.0

Instrukcja obsługi



ELMARK
Automatyka Sp. z o.o.

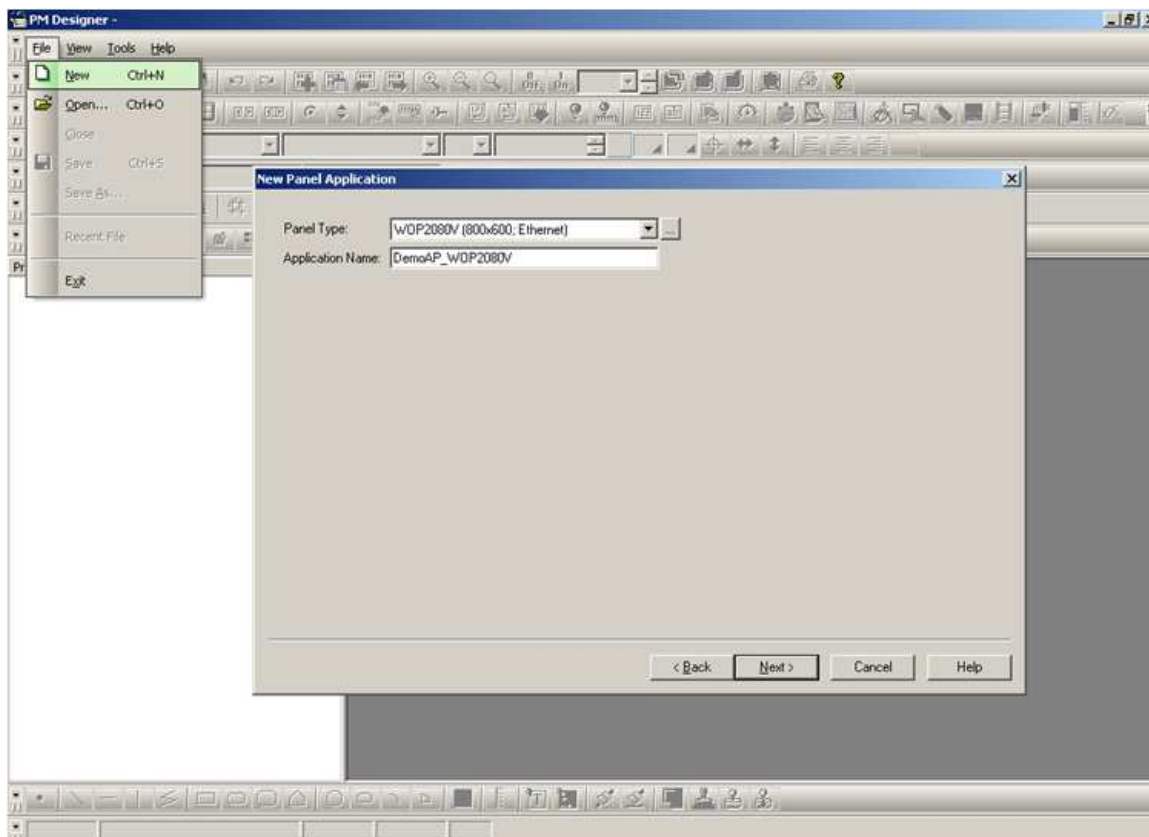
Spis treści

1. Tworzenie projektu i konfiguracja połączeń	- 4 -
1.1 Tworzenie projektu	- 4 -
1.2 Port Ethernet.....	- 5 -
1.3 Transmisja Transparent Serwer.....	- 7 -
2. Odczyt i zapis wartości z/do rejestrów.....	- 8 -
2.1 Odczyt i zapis wartości rejestru.....	- 8 -
2.1.1 Utworzenie połączenia z modułem ADAM-4055 i interfejsem RS-485.....	- 8 -
2.1.2 Odczyt stanów binarnych z urządzenia na przykładzie modułu ADAM-4055	- 9 -
2.1.3 Odczyt przy pomocy makra	- 10 -
2.1.4 Zapis stanów logicznych na przykładzie modułu ADAM-4055	- 13 -
2.1.4.1 Obsługa kontrolki Bit Button	- 14 -
2.1.4.2 Obsługa kontrolki Toogle Switch	- 15 -
2.1.5 Odczyt wejść analogowych na przykładzie modułu ADAM-4017+.....	- 16 -
2.1.5.1 Odczyt bezpośredni rejestru	- 16 -
2.1.5.2 Odczyt rejestru przez makro	- 18 -
3. Elementy graficzne	- 18 -
3.1 Blok Bar Graph.....	- 18 -
3.2 Powiązanie bloku wskaźnika <i>Meter</i> z suwakiem typu <i>Slider</i>	- 20 -
3.3 Powiązanie bloku graficznego <i>Picture Display</i> z blokiem <i>Bar Graph</i> i suwakiem <i>Slider</i>	- 21 -
4. Dodawanie Tagów	- 23 -
5. Receptury	- 24 -
6. Logowanie danych.....	- 27 -
6.1 Dodawanie i konfiguracja	- 27 -
6.2 Tabele i wykresy wykorzystujące Data Logger	- 30 -
6.2.1 Historic Data Table	- 30 -

6.2.2 Historic Trend Graph	- 31 -
6.2.3 Zapis danych typu Data Logger do pliku tekstowego.....	- 31 -
7. Uruchamianie i testowanie.....	- 32 -
7.1 Kompilacja i wgrywanie projektu do sterownika	- 32 -
7.2 Testowanie i symulacja.....	- 33 -
8. Poziomy dostęp do aplikacji panelu.....	- 35 -
9. Languages	- 37 -

1. Tworzenie projektu i konfiguracja połączeń

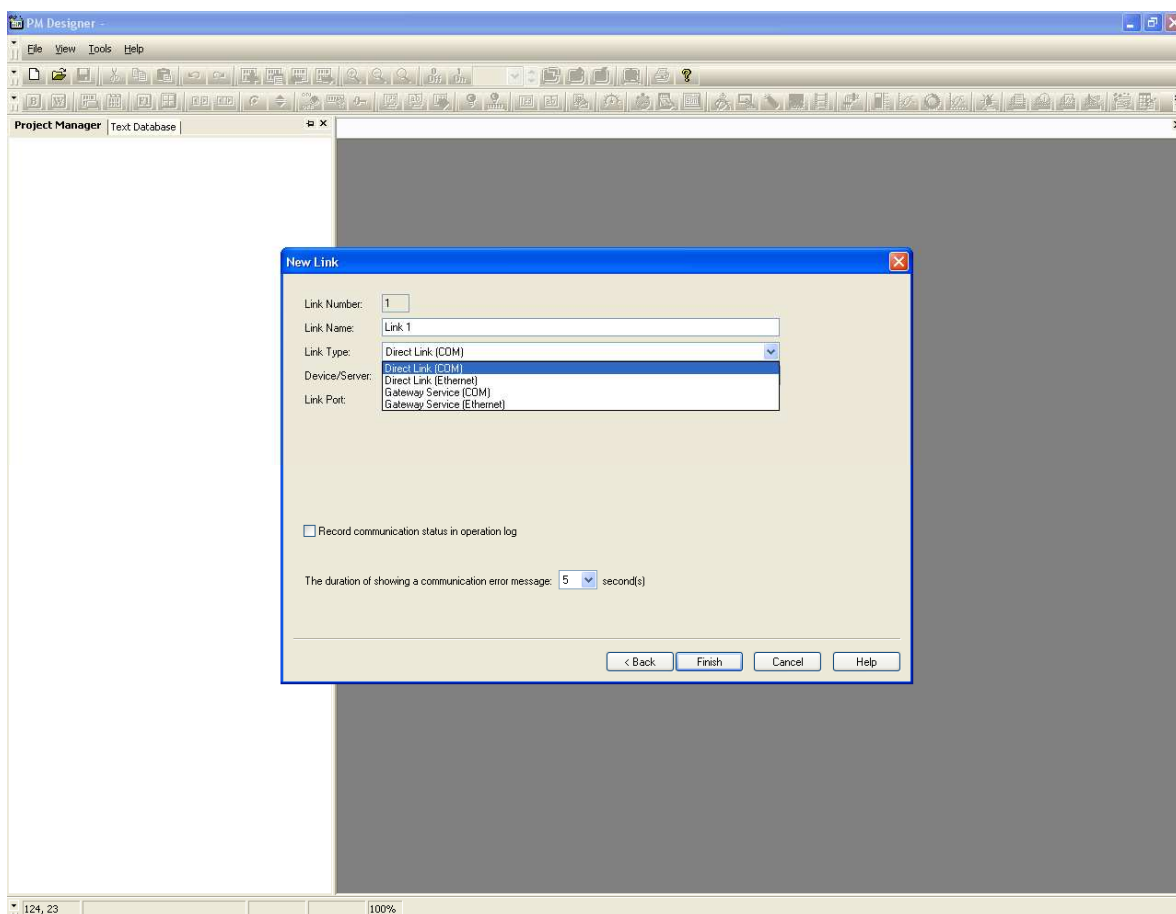
1.1 Tworzenie projektu



Rysunek 1. Tworzenie projektu

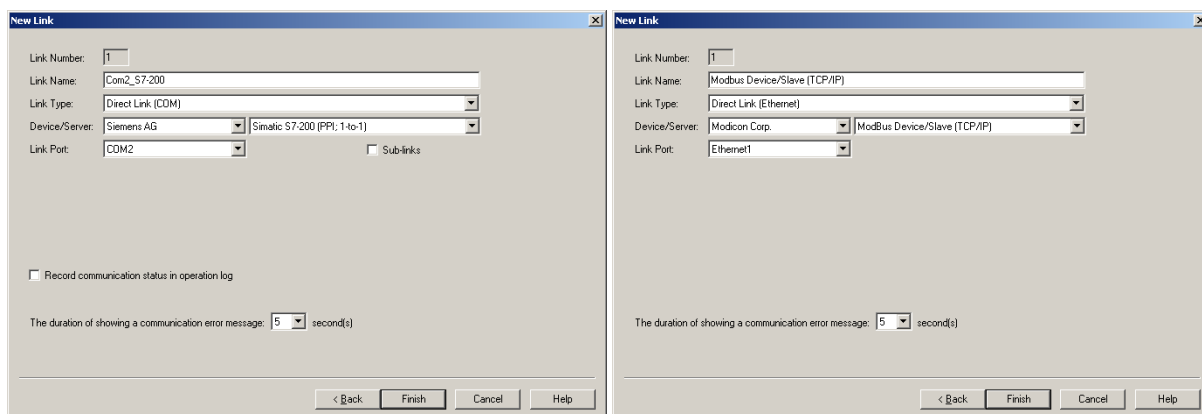
W pierwszej kolejności wchodzimy w menu projektu i wybieramy pole **New**. Następnie wybieramy model panelu operatorskiego w polu **Panel Type** oraz nadajemy nazwę projektu. Kolejno definiujemy połączenia z urządzeniami współpracującymi z panelem. W tym celu wybieramy interfejs oraz typ połączenia w polu „**Link Type**”:

- „**Direct Link (COM)**” umożliwia komunikację poprzez port szeregowy (w przypadku RS-485 nie jest definiowany adres tak więc w późniejszym etapie do połączenia można przypisać kilka urządzeń np. przy protokole ModbusRTU).
- „**Direct Link (Ethernet)**” zapewnia komunikację poprzez sieć Ethernet ze zdefiniowanym adresem IP.
- „**Gateway Service (COM)**” jest to brama RS.
- „**Gateway Service (Ethernet)**” jest to brama Ethernet.



Rysunek 2. Wybór interfejsu w połączenia Link

W kolejnym kroku należy wybrać protokół komunikacyjny w polu „**Device/Server**” oraz numer interfejsu w polu „**Link Port**”.

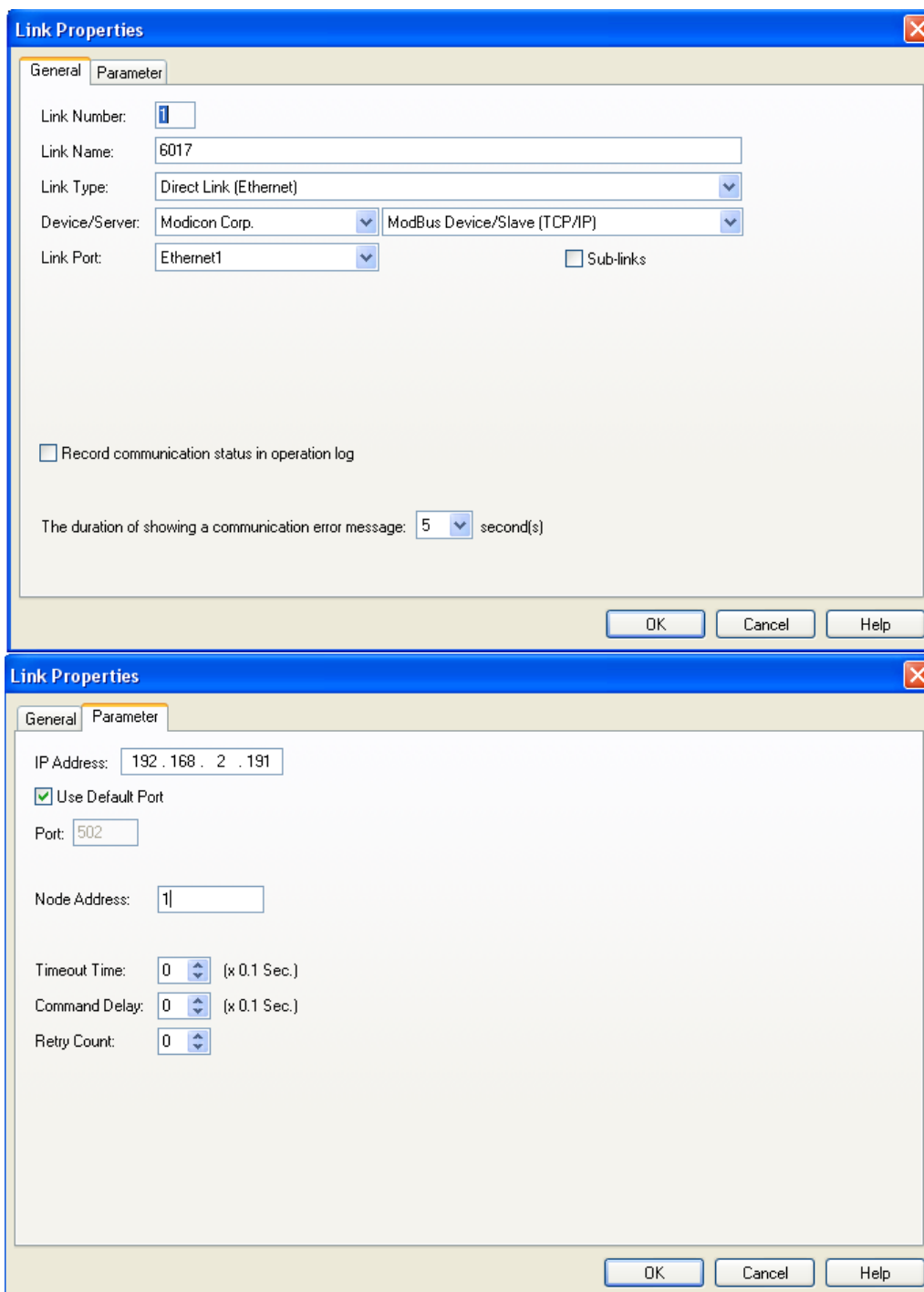


Rysunek 3. Definiowanie połączenia Link

1.2 Port Ethernet

Aby skorzystać z interfejsu Ethernet, na etapie konfiguracji połączenia Link, w polu **Link Type** należy określić parametr **Direct Link (Ethernet)**. Przy tym interfejsie w polu **Device/Server** mamy bogaty wybór protokołów, które

mogą posłużyć do komunikacji z urządzeniami wielu producentów. W naszym przykładzie skorzystamy z protokołu Modbus TCP niezbędnego do komunikacji z modułem wejść analogowych ADAM-6017 firmy Advantech. W tym celu wybieramy protokół **Modicon Corp.** oraz **Modbus Device/Slave (TCP/IP)**. We właściwościach połączenia Link podajemy adres IP modułu, port komunikacyjny (domyślnie dla połączeń Modbus TCP wynosi 502) oraz **Node Address**, który dla serii ADAM-6000 zawsze wynosi 1.

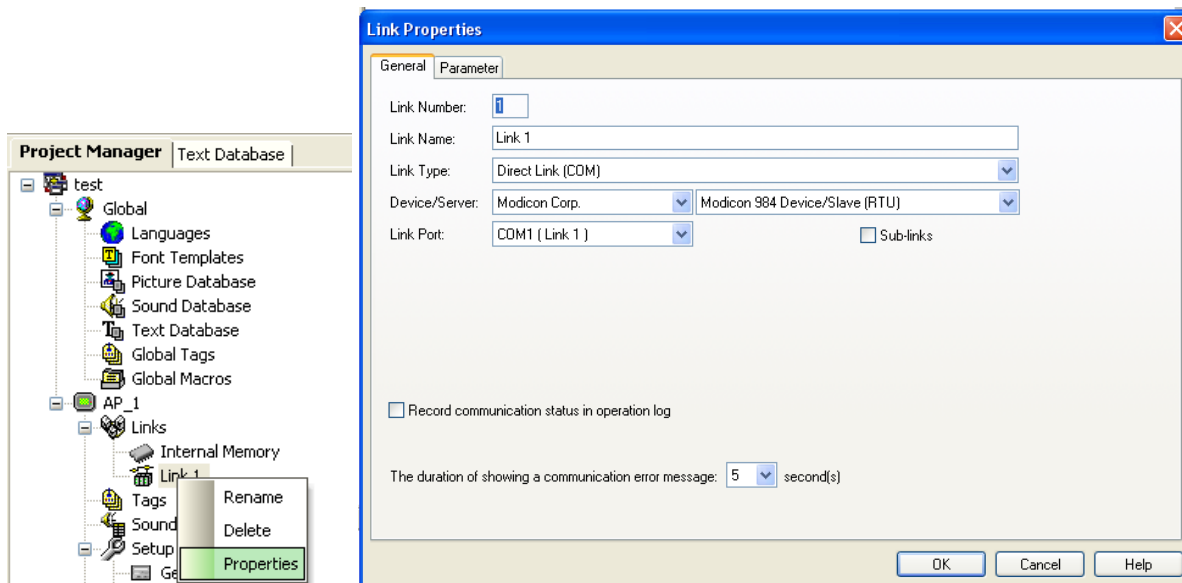


Rysunek 4. Dodawanie połączenia Link z portem Ethernet przy pomocy protokołu Modbus TCP

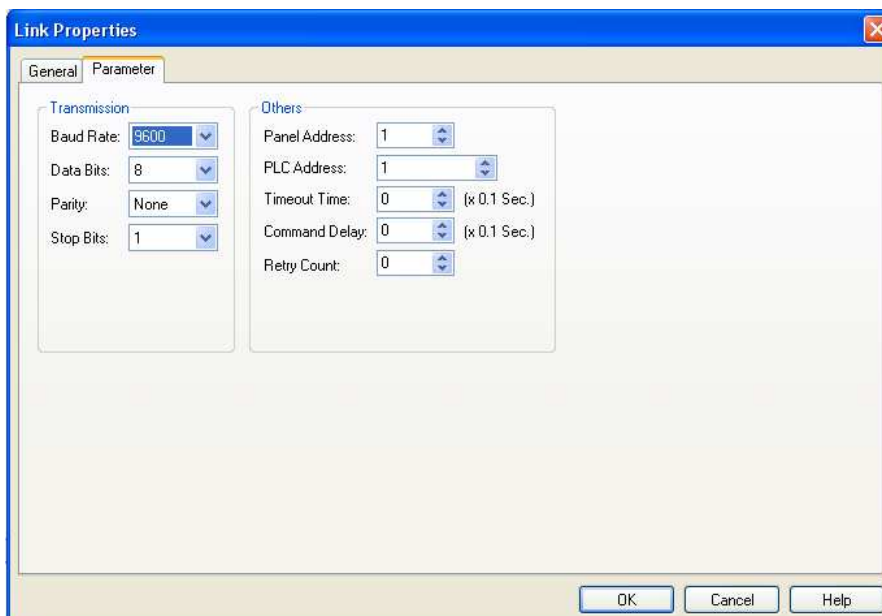
1.3 Transmisja Transparent Serwer

Tryb pracy Transparent Serwer umożliwia utworzenie lustrzanego, przezroczystego połączenia portu RS za pośrednictwem innego portu szeregowego. W takiej konfiguracji dane mogą być odczytywane lub zapisywane do urządzeń podłączonych do panela operatorskiego za pośrednictwem narzędzi zewnętrznych, takich jak PC z zainstalowanym oprogramowaniem typu Scada.

W pierwszej kolejności konfigurujemy połączenie główne w panelu operatorskim „Link 1”. Do transmisji wybieramy protokół Modbus RTU na porcie COM1.

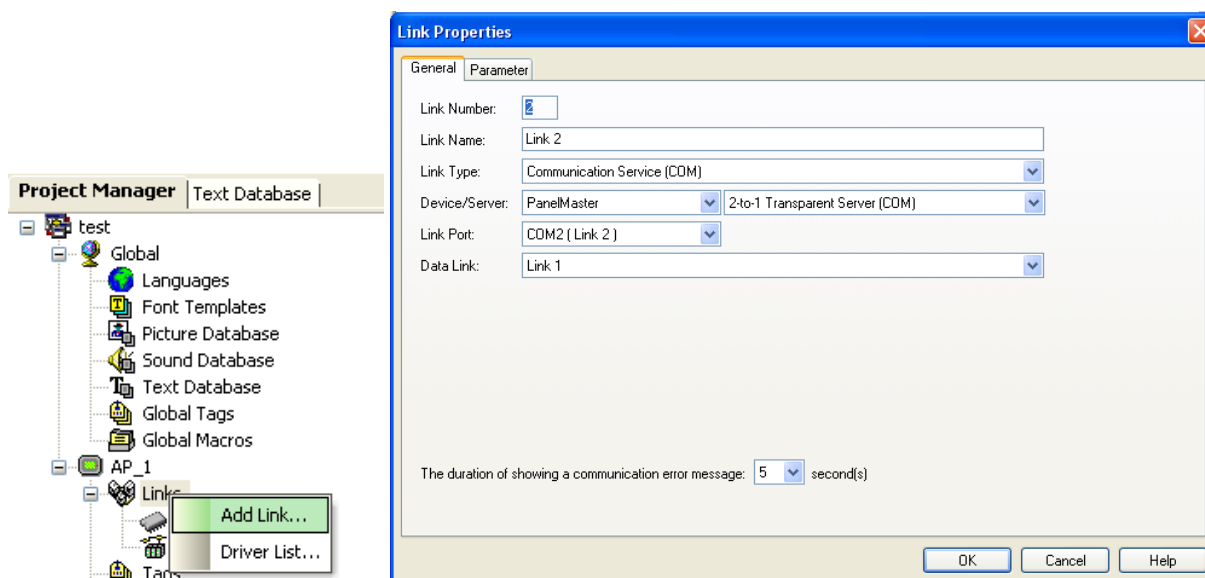


Rysunek 5. Zakładka General w ustawieniach połączenia Link 1



Rysunek 6. Zakładka Parameter w połączeniu Link 1

Następnie dodajemy nowe połączenie „Link 2”, które ustawiamy w trybie „2-to-1 Transparent Server (COM)”. W polu „Data Link” wybieramy port mapowany a w polu „Link Port” port na którym będą dostępne mapowane rejestry.



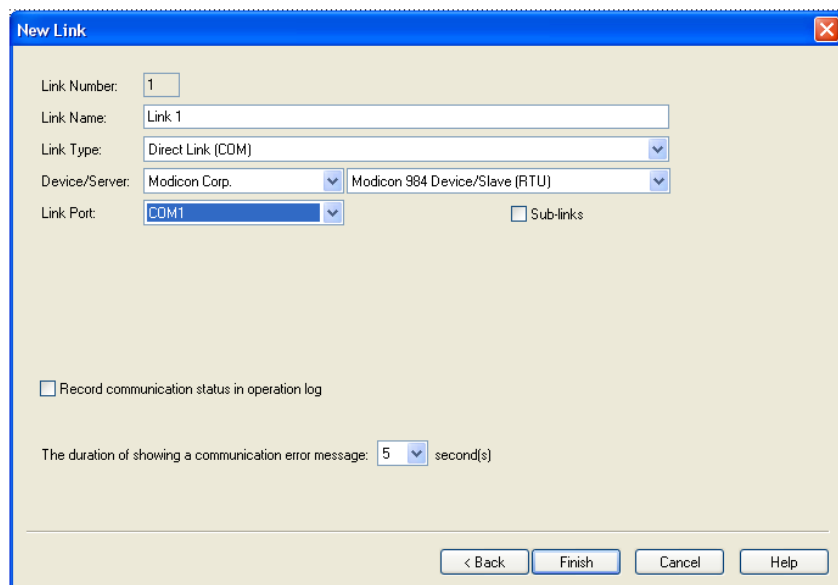
Rysunek 7. Ustawienia w połączeniu Link 2

2. Odczyt i zapis rejestrów

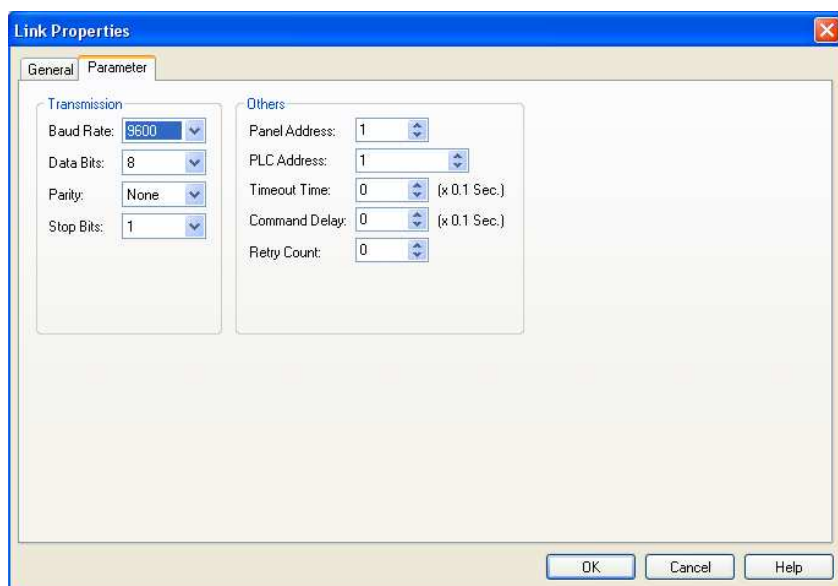
2.1 Odczyt i zapis wartości rejestru

2.1.1 Utworzenie połączenia z modułem ADAM-4055 z interfejsem RS-485

Klikamy prawym klawiszem na drzewie projektu na **Link** i wybieramy **Add Link**. Następnie podajemy nazwę połączenia i wybieramy odpowiedni port - tutaj **Direct Link (COM)**. Z listy obsługiwanych protokołów wybieramy **Modicorn Corp.** oraz **Modicon 984 Device/Slave (RTU)**, który odpowiada protokołowi Modbus RTU i zatwierdzamy. Po tym, klikamy prawym klawiszem myszy na utworzone połączenie i wchodzimy w ustawienia. W zakładce **Parameter** ustawiamy parzystość **Parity** jako **None** i zatwierdzamy.



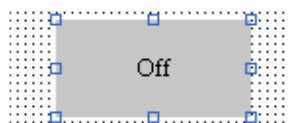
Rysunek 8. Dodawanie połączenia Link



Rysunek 9. Dodawanie połączenia Link

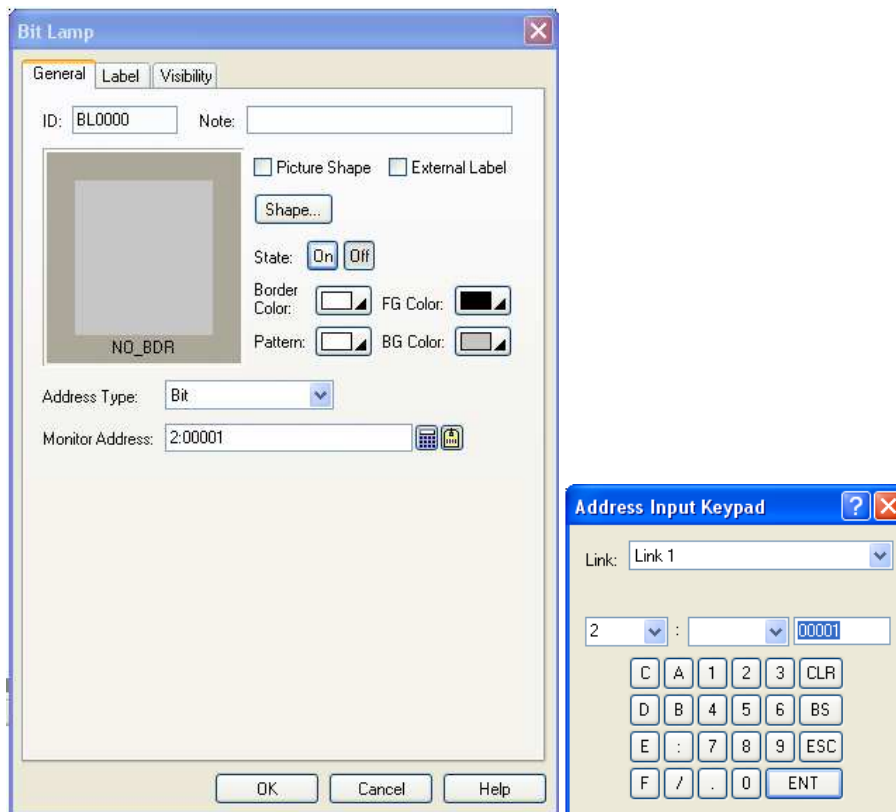
2.1.2 Odczyt stanów binarnych z urządzenia na przykładzie modułu ADAM-4055

Do wyświetlania stanów binarnych w programie WebOP Designer służy kontrolka **Bit Lamp**. W celu dodania kontrolki należy kliknąć lewym klawiszem myszy na ikonie  na pasku narzędziowym lub wybrać ją z górnego menu **Object->Bit Lamp**, i umieścić ją na polu ekranu.




Rysunek 10. Kontrolka Bit Lamp w obszarze roboczym projektu

Dwukrotne kliknięcie na dodanej kontrolce otwiera okno konfiguracji:



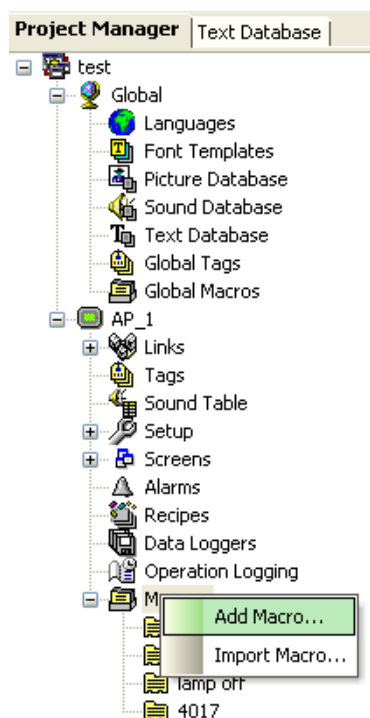
Rysunek 11. Właściwości kontrolki Bit Lamp

W przypadku modułu ADAM-4055 dane o stanie wejść binarnych pobierane są z rejestrów 00001-00008 kolejno dla kanałów DI0-DI7. W celu dodania monitorowanego bitu należy w polu **Address Type** wybrać wartość **Bit**. Następnie kliknąć klawisz  a po otwarciu się okna **Address Input Keypad** w polu **Link** wybierać Internal Memory lub inne zdefiniowane połączenie np. domyślnie Link 1. Teraz należy podać adres urządzenia, typ zmiennej oraz numer rejestru np. 00001. Po poprawnym wprowadzeniu danych należy kliknąć przycisk **ENT**.

2.1.3 Odczyt przy pomocy makra

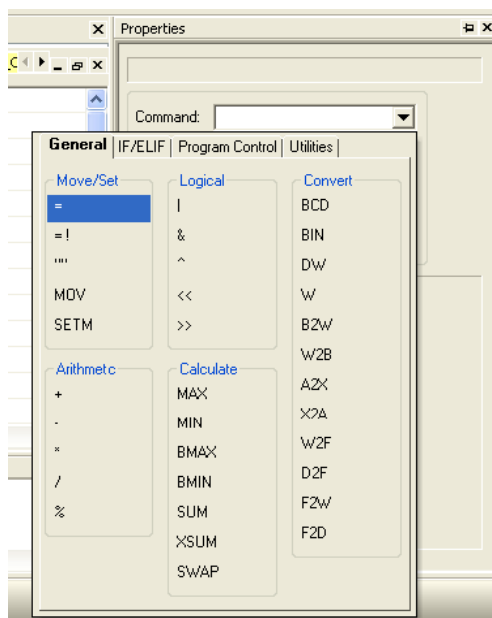
Inną możliwością odczytu stanów rejestrów jest wykorzystanie makra czasowego wywoływanego np. co dwie sekundy. W takim przypadku generowany ruch w sieci jest mniejszy a dane uaktualniane są regularnie co z góry ustalony przedział czasowy. Dodatkowa dane mogą być przetrzymywane w pamięci panelu operatorskiego, co umożliwi ich zastosowanie w wielu procesach jednocześnie i zmniejsza obciążenie urządzenia.

a) Aby dodać makra czasowe w drzewie projektu klikamy prawym przyciskiem myszy na **Macros** i wybrać opcję **Add Macro**.



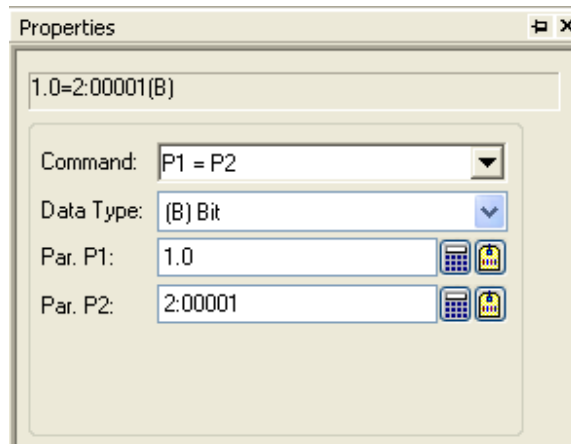
Rysunek 12. Dodawanie makra do projektu

W otwartym oknie **New Macro** podajemy nazwę makra i zatwierdzamy OK. Po tej czynności w drzewie Macro pojawi się nowe makro, które możemy edytować poprzez dwukrotne kliknięcie myszy. W edytorze makr, w polu **Command** wybieramy pozycję o określonym stanie logicznym - w naszym przypadku „=” z kolumny **Move/Set**.



Rysunek 13. Dodawanie komendy do makra

Dalej w ustawieniach **Propertis** definiujemy zmienne:



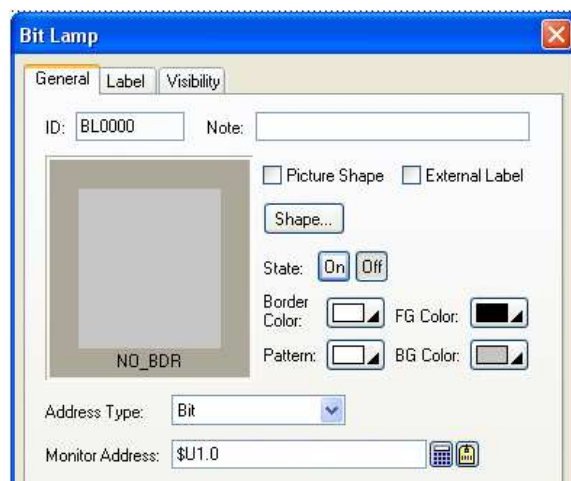
Rysunek 14. Wybór parametrów dla komendy makra

W polu **Data Type** wybieramy typ zmiennej jako **(B) Bit**. Do pola **Par. P1** wpisujemy zmienną lokalną np. \$U1.0 do której będzie zapisywany stan rejestru z urządzenia. Parametr **Par. P2** odpowiada za adres rejestru w urządzeniu. Adres należy dobrać analogicznie do poprzedniego przykładu (np. 00001 dla urządzenia o adresie 2 i rejestrze 00001).

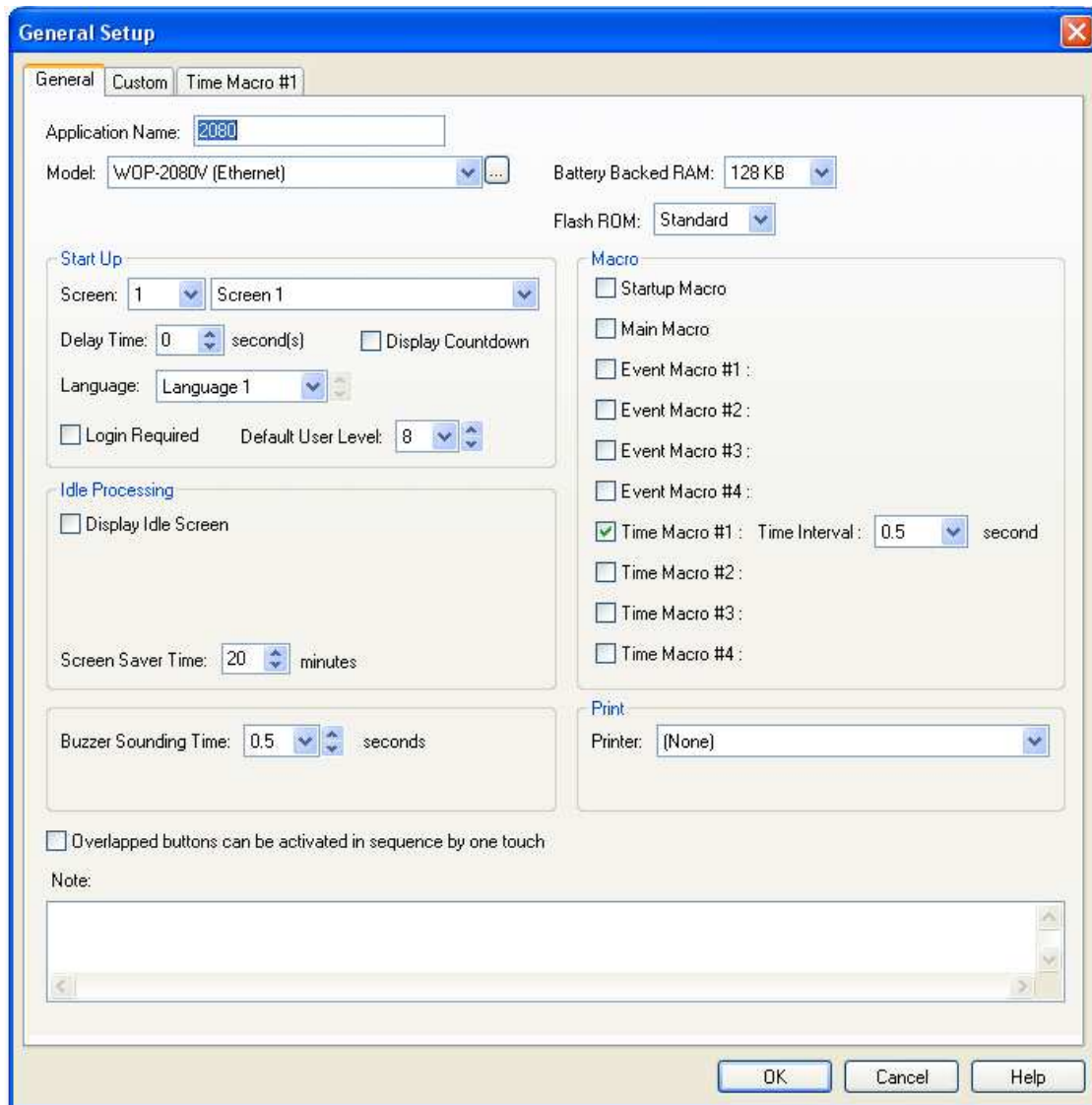
W edytorze makr pojawi się linia:
U\$1.0=2:00001(B)

b) Teraz pozostaje nam skonfigurowanie cykli w jakich będzie uruchamiane makro. W drzewie projektu dwukrotnie klikamy na nazwę projektu i wchodzimy w okno **General**. Zaznaczamy pozycję **Time Macro #1** z interwałem czasowym **Time Interval** np. 0.5 sekundy. W nowej zakładce **Time Macro #1** przypisujemy wcześniej zdefiniowane makro do pola **Macro name**. W takiej konfiguracji makro będzie uruchamiane cyklicznie co 0.5 sekundy.

c) Lampkę **Bit Lamp** dodajemy analogicznie jak w punkcie 2.1.2 z tymże do pola **Monitor Address** wpisujemy adres zmiennej lokalnej z pamięci Internal Memory czyli \$U1.0 jak w powyższym przykładzie.





Rysunek 15. Blok Bit Lamp

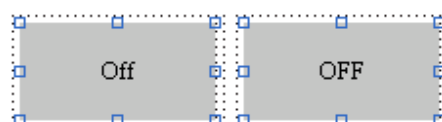


Rysunek 16. Konfiguracja podstawowych parametrów panelu

2.1.4 Zapis stanów logicznych na przykładzie modułu ADAM-4055


W przypadku modułu ADAM-4055 do zapisu/odczytu stanu wyjść cyfrowych DO0-DO7 wykorzystywane są rejestry z zakresu od 00017 do 00024.

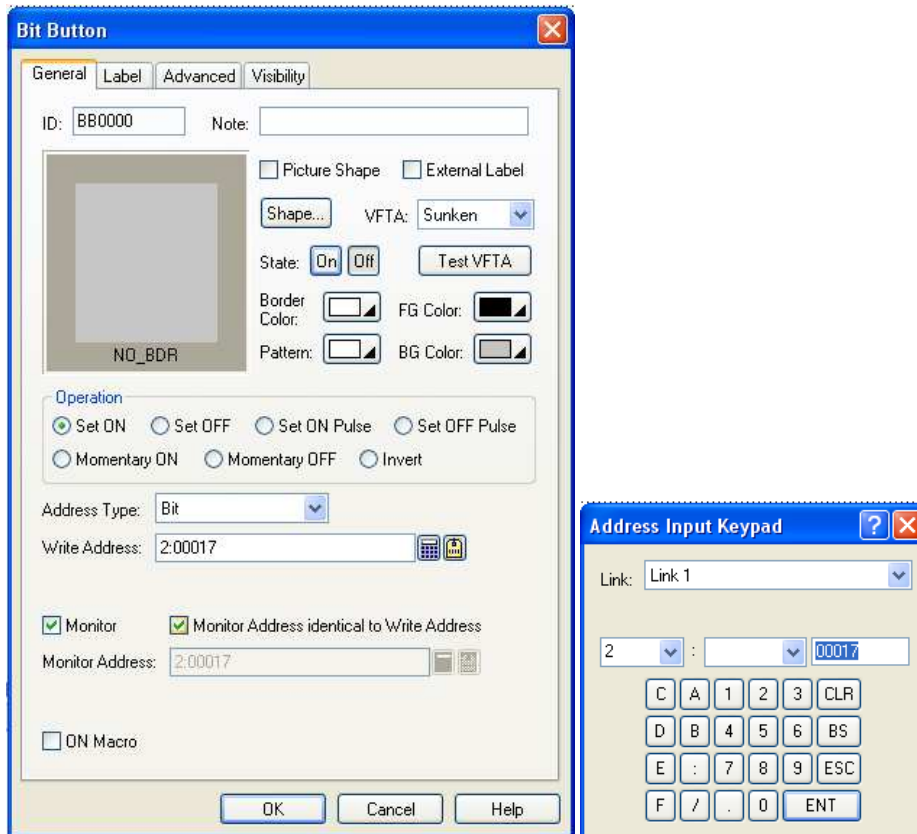
Do zapisu danych binarnych posłużyć mogą dwie kontrolki: **Bit Button**  oraz **Toggle Switch** . Wstawianie i otwieranie okna konfiguracyjnego jest analogiczne do punktu 2.1.2



Rysunek 17. Widok kontrolki Bit Button i Toggle Switch

2.1.4.1 Obsługa kontrolki Bit Button

Umieszczamy przycisk **Bit Button**  na polu ekranu i wchodzimy w jego ustawienia (klikamy dwukrotnie myszką).



Rysunek 18. Konfiguracja parametrów kontrolki Bit Button

W polu **Operation** wybieramy sposób działania przycisku:

Set ON, Set ON Pulse – ustawia bit **Write Address** na 1.

Set OFF, Set OFF pulse – ustawia bit **Write Address** na 0.

Momentary ON – ustawia bit **Write Address** na 1 przez czas trzymania przycisku, po puszczeniu przycisku bit przyjmuje wartość 0.

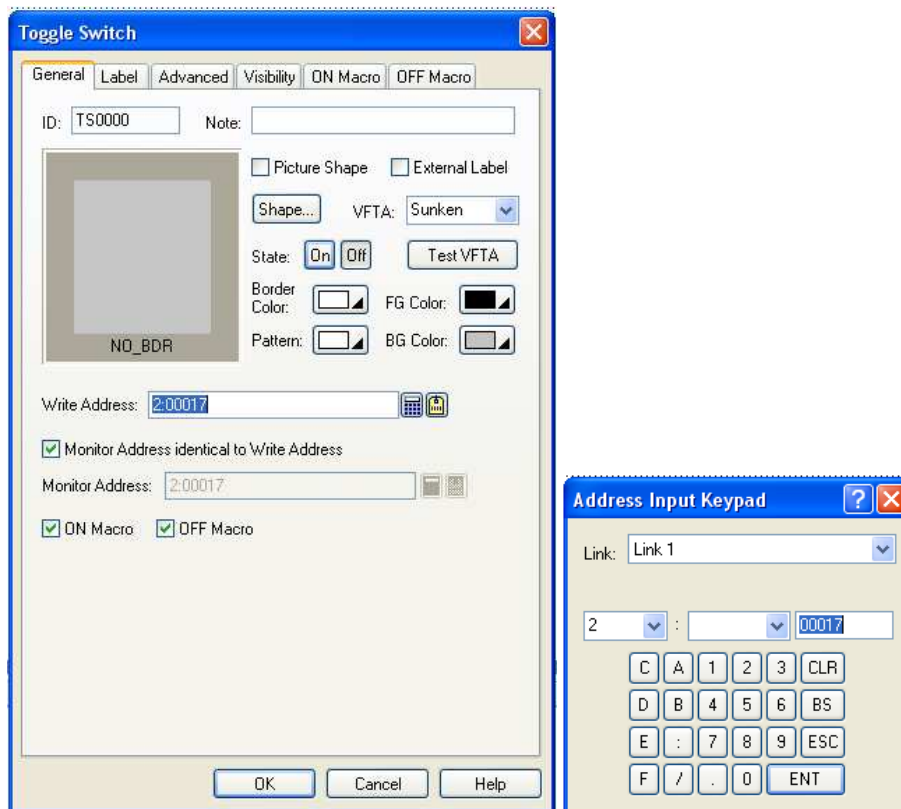
Momentary OFF – ustawia bit **Write Address** na 0 przez czas trzymania przycisku, po puszczeniu przycisku bit przyjmuje wartość 1.

Invert – ustawia bit **Write Address** na 1 gdy przycisk jest wyłączony (stan 0) i na odwrot.

W polu **Write Address** należy podać adres do którego będzie przypisany stan bitowy przycisku. W przypadku ADAM-4055 wartość 2:00017 oznacza że zostanieysterowane wyjście 0 o adresie 00017 (urządzenie posiada adres 2).

2.1.4.2 Obsługa kontrolki Toggle Switch

Kontrolka **Toggle Switch** umożliwia pracę w systemie dwustanowym w którym ON odpowiada stanowi wysokiemu „załączony” oraz OFF niskiemu „Wyłączony”. W poniższym przykładzie do kontrolki przypisany został rejestr 00017 powiązany z wyjściem cyfrowym DO0 w module ADAM-4055.

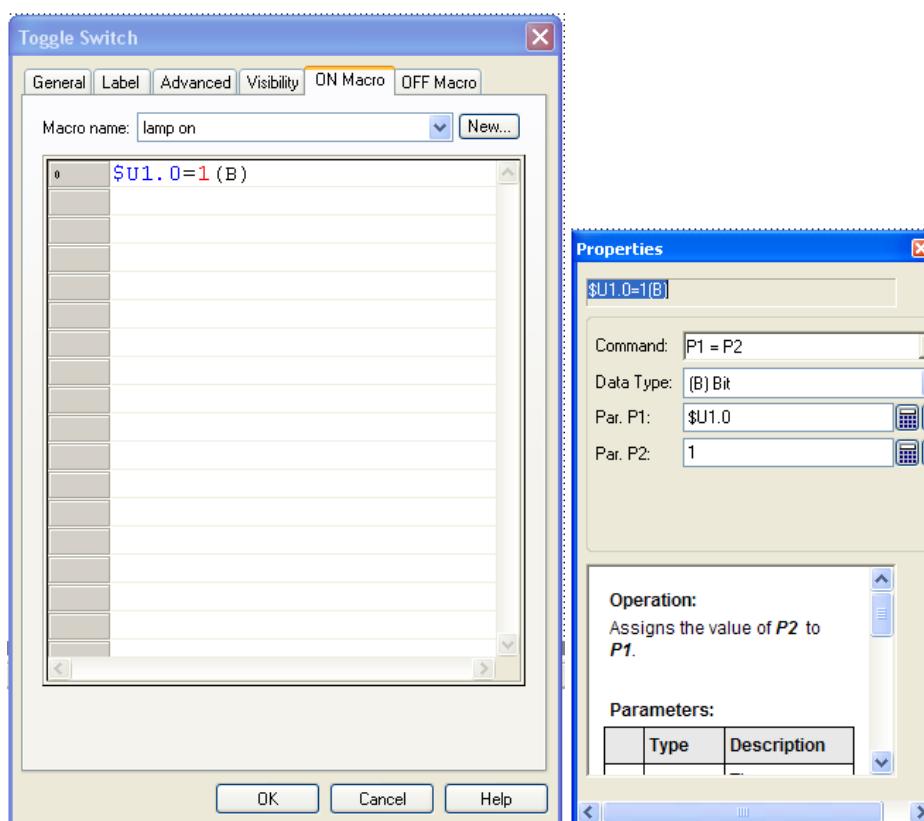


Rysunek 19. Konfiguracja parametrów kontrolki Toggle Switch

W przykładzie aktywowano również opcje ON i OFF Macro, pozwalające na wykonanie zaimplementowanych w makrach operacji przy każdej zmianie stanu przycisku. Zaznaczenie pola ON lub OFF Macro uaktywnia zakładkę o tej samej nazwie, w której należy dodać odpowiednie makro w polu Macro name. W naszym przykładzie przycisk został powiązany z wcześniej dodaną lampką Bit Lamp ze zdefiniowanym adresem lokalnym \$U1.0. I tak kolejno, w przypadku zmiany stanu (ON na OFF lub odwrotnie) zmienia się jednocześnie stan naszej lampki. W takiej sytuacji nie ma już dalszej potrzeby korzystania z makra czasowego ponieważ makro jest wywoływane tylko w momentach zmiany stanu, a nie cyklicznie co zadany w makrze interwał czasowy.

ON Macro - \$U1.0 = 1(B)

OFF Macro - \$U1.0 = 0(B)




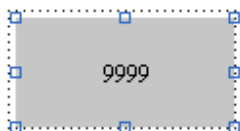
Rysunek 20. Konfiguracja ON Macro

2.1.5 Odczyt wejść analogowych na przykładzie modułu ADAM-4017+

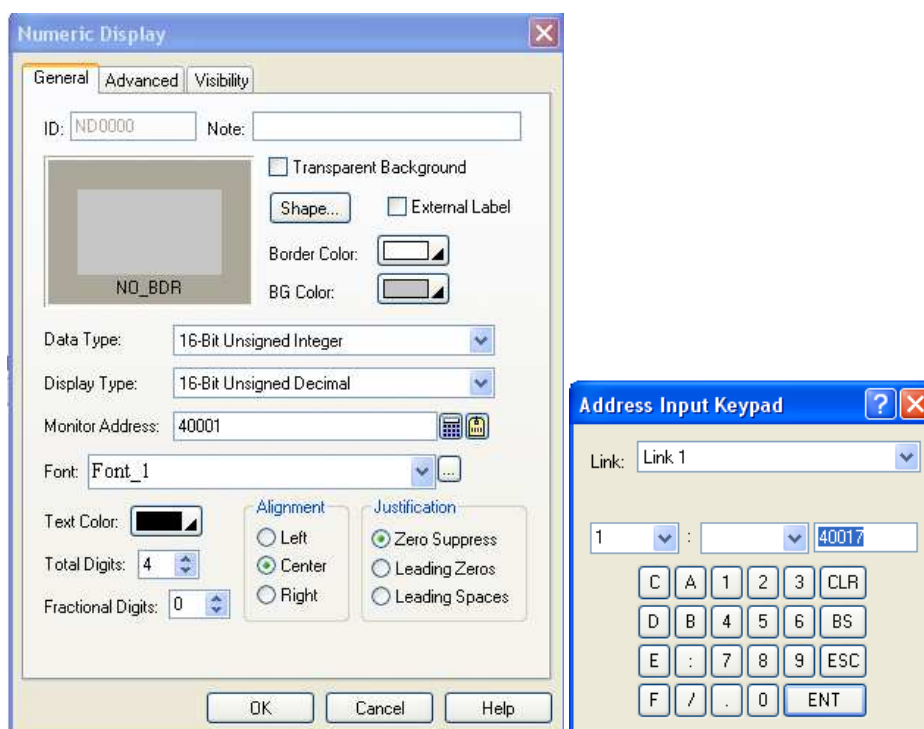
ADAM-4017+ to moduł wejść analogowych wyposażony w 8 niezależnych kanałów z wejściami napięciowymi i prądowymi o rozdzielczości 16 bitów. Urządzenie wyposażone zostało w interfejs RS-485 i współpracuje z protokołem Modbus RTU. Dane przechowywane są w rejestrach od 40001(kanał A10) do 40008(kanał A17).

2.1.5.1 Odczyt bezpośredni rejestru

a) W celu odczytania wartości rejestru bezpośrednio z urządzenia można skorzystać z przykładowej kontrolki **Numeric Display** .



Rysunek 21. Widok kontrolki Numeric Display



Rysunek 22. Konfiguracja parametrów kontrolki Numeric Display

Zakładka **General**:

Shape - klikając na ten przycisk można zdefiniować wygląd pola **Numeric Display**.

External Label - zaznaczenie pola pozwala na dodanie etykiety tekstowej przed polem **Numeric Display**. Parametry etykiety definiuje się w zakładce **External Label** pojawiającej się po zaznaczeniu tego pola.

Border Color i **BG Color** - opcje pozwalają określić kolor ramki oraz pola **Numeric Display**.

Text Color - opcja pozwala wybrać kolor tekstu.

Total Digits - opcja pozwala ustawić liczbę wyświetlanych cyfr.

Fractional Digits – opcja pozwala ustawić wyświetlaną liczbę cyfr po przecinku.

Alignment oraz **Justification** – opcje pozwalają ustawić sposób wyświetlania wartości w polu **Numeric Display**, takie jak wyrównanie tekstu lub niewyświetlanie nieznaczących zer.

b) W zakładce **Advanced** można ustawić opcję dodatkowe, jak np. zmiana koloru czcionki lub tła po przekroczeniu ustalonych poziomów wartości.

c) W zakładce **Visibility** można ustawić opcje dodatkowe, jak np. widoczność pola w zależności od poziomu zalogowanego użytkownika.

d) W zakładce **General** w polu **Data type** i **Display Type** wybieramy 16-Bit Unsigned Integer.

e) W polu **Monitor Address** wpisujemy odpowiadający kanałowi AI0 rejestr 40001 przypisany do urządzenia ADAM-4017+ o adresie 1.



2.1.5.2 Odczyt rejestru przez makro

a) W celu odczytu wartości rejestru przez makro skorzystać należy z makra czasowego (sposób tworzenia i konfiguracji podany w podpunkcie 2.1.3).

b) W makrze wpisujemy odpowiednią linijkę np.:

```
$U10=40001
```


W tym przypadku wartość z rejestru 40017 z urządzenia o adresie 1 jest zapisywana do zmiennej lokalnej o adresie \$U10. Typ zmiennej należy ustalić jako Unsigned Integer.

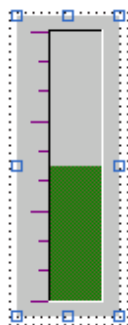
c) W polu **Monitor Address** kontrolki **Numeric Display** wpisać należy adres zmiennej, która przechowuje odczytaną wartość, w tym przypadku \$U10. Można również skorzystać z kontrolki innych typów (np. **Bar Graph** , **Meter** ) postępowanie jest analogiczne do wyżej opisanych przypadków.

d) Odpowiednio skonfigurować makra czasowe w drzewie projektu – punkt 2.1.3 c).

3. Elementy graficzne

3.1 Blok Bar Graph

Z paska narzędziowego dodajemy bloczek **Bar Graph**  i dodajemy go na ekran.



Rysunek 23. Widok bloku Bar Graph

Klikamy dwukrotnie prawym klawiszem myszy i wchodzimy w ustawienia bloczku.

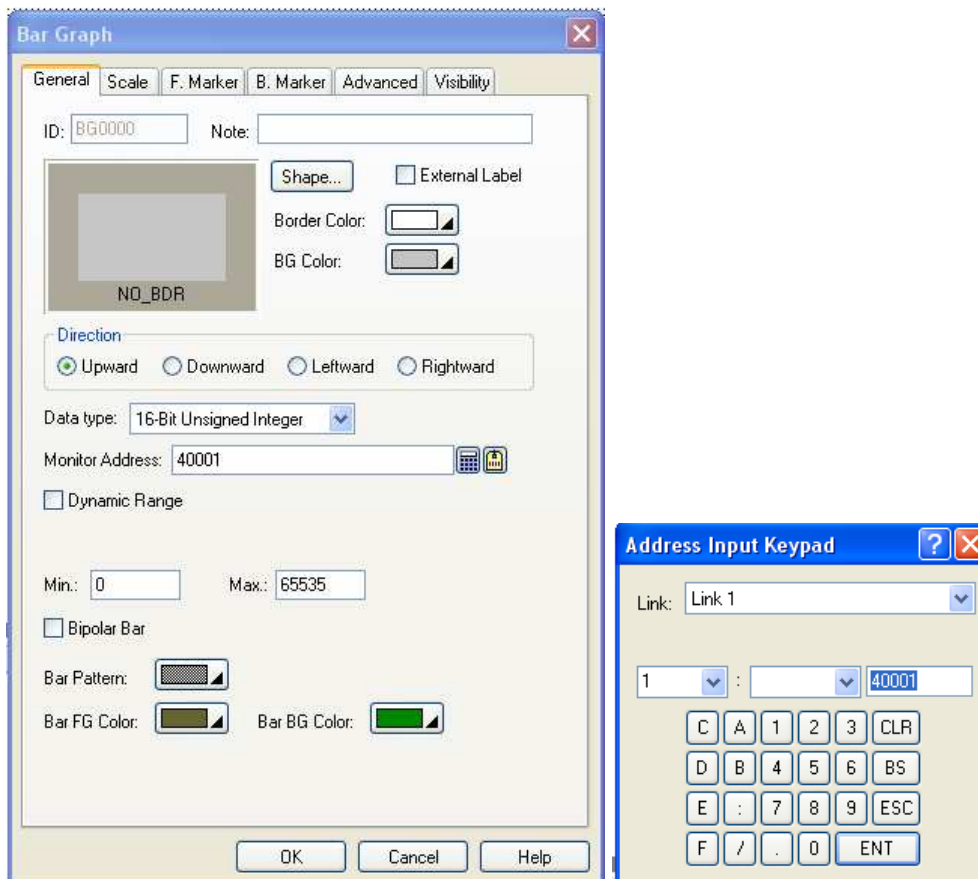
a) W zakładce głównej **General** wyróżniamy pola:

Shape - pozwala na zmianę wyglądu bloczki.

External label – pozwala na dodanie etykiety tekstowej w bloczku. W momencie aktywacji pojawia się odpowiednia zakładka umożliwiająca dalszą konfigurację.

Border Color i BG Color – odpowiadają za kolor ramki oraz pola Bar Graph.

Direction – określa kierunek zmiany wypełnienia.



Rysunek 24. Konfiguracja i dodawanie rejestru do bloku Bar Graph

Data Typ – definiuje typ zmiennej.

Monitor Address – w tym polu należy wpisać adres urządzenia i rejestru.

Min, Max – określają minimalną i maksymalną wartość słupka bargraphu. Zaznaczenie pola **Dynamic Range** umożliwia zdefiniowanie tych wartości poprzez przypisaną zmienną.

Bipolar Bar - pozwala na ustawienie skali zawierającej wartości ujemne i dodatnie, **Middle point** określa środek skali.


Bar Pattern, Bar FG Color, Bar BG Color – pozwalają zdefiniować wygląd słupka bargraphu.

b) Zakładka **Scale** umożliwia zdefiniowanie parametrów skali wyświetlanej obok słupka.


c) Zakładki **F. Marker, B. Marker** definiują markery w postaci strzałek, które są dodawane do przy osi słupka bargraph. We właściwościach można zdefiniować do 24 strzałek oraz określić ich kształt oraz kolor zmiany w przypadku przekroczenia przypisanej wartości.

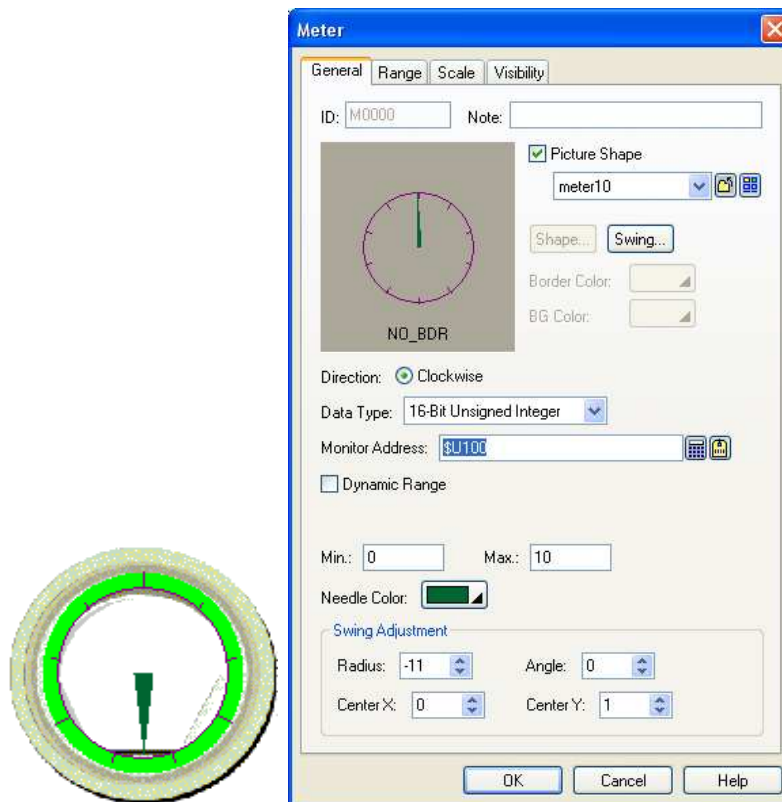
d) W zakładce **Advanced** można uaktywnić oraz zmienić zakres wyświetlanych danych. Pole **Percentage Display** uaktywnia procentowy wskaźnik wypełniania bloczku.

e) W zakładce **Visibility** można ustalić takie parametry jak położenie, uaktywnić widoczność w zależności od stanu przypisanego rejestru oraz określić dostęp użytkowników.


W podobny sposób należy skonfigurować bloczki typu **Meter** 

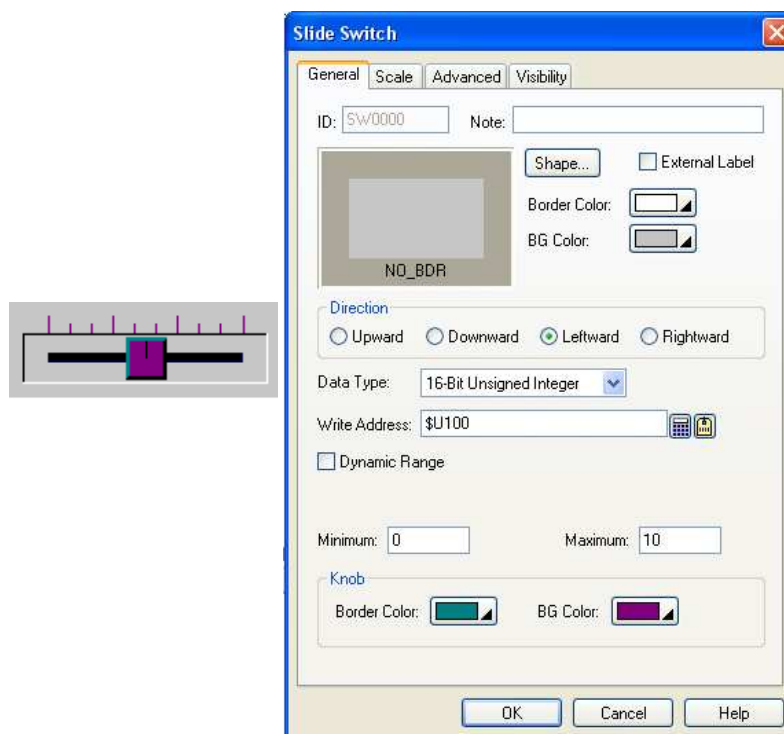
3.2 Powiązanie bloku wskaźnika *Meter* z suwakiem typu *Slider*

Na ekranie dodajemy bloczek typu **Meter** , którego konfiguracja odbywa się w podobny sposób jak w bloku Bar Graph w punkcie 3.1. W polu Monitor Address wstawiamy zmienną wewnętrzną Internal Memory \$U100, która powiązana zostanie ze zmienną z bloku **Slider**. Poprzez pole **Picture Shape** do bloku przypisujemy obraz graficzny, pobierany z dostępnej bazy programu lub z własnych zasobów (aby dodać poniższy rysunek należy zaznaczyć Picture Shape, nacisnąć przycisk Import from Library, wybrać Picture Library file jako Meters i zaznaczyć meter10). Położenie rysunku względem zarysu można ustalić w grupie pól **Swing Adjustment**.



Rysunek 25. Widok i konfiguracja bloku Meter

Następnie na ekranie wstawiamy blok **Slider Switch**  i w polu **Monitor Adres** dodajemy zmienną wewnętrzną \$U100. Ustawiamy **Direction** jako Rightward oraz podajemy zakres zmiennej od 0 do 10. W zakładce **Scale** zaznaczamy pole Display. Po skompilowaniu i uruchomieniu wskaźnik Meter będzie liniowo reagował na zmiany położenia suwaka Slider.



Rysunek 26. Konfiguracja bloku Slider Switch

3.3 Powiązanie bloku graficznego *Picture Display* z blokiem *Bar Graph* i suwakiem *Slider*

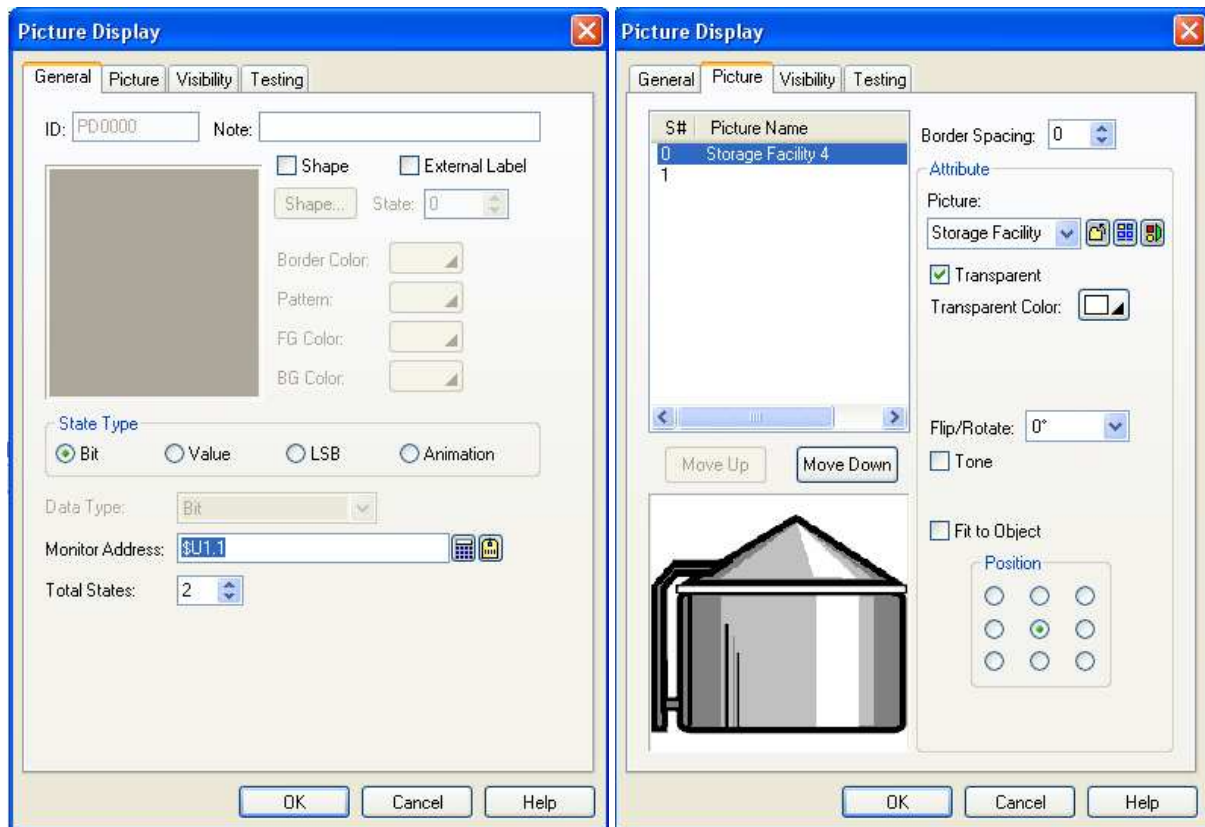
Na ekranie wstawiamy blok **Picture Display** , **Slider Switch**  oraz **Bar Graph** . Klikamy dwukrotnie na blok Picture Display i konfigurujemy:

State Type - typ zmiennych określających widoczność bloku.

Monitor Address - zmienna kontrolująca widoczność bloku (powiązana z polem State Type).

Picture - pole umożliwia wybór elementów graficznych z biblioteki programu lub z własnych zasobów zdjęć z rozszerzeniem bmp, gif lub wfm. W naszym przykładzie wybieramy element z rodziny Tanks o nazwie Storage Facility 4.

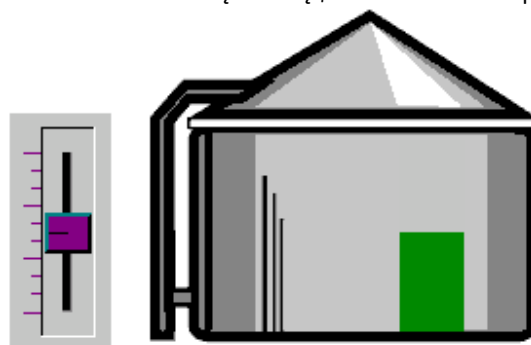
Transparent - jest opcja pozwalająca na wybranie koloru, który będzie traktowany jako przezroczysty – w naszym przypadku wybieramy biały.



Rysunek 27. Konfiguracja bloku Picture Display

Następnie w utworzonym bloku **Bar Graph** wybieramy **Direction** jako **Upward** i podajemy zmienną lokalną \$U100, odpowiadającą za stan wypełnienia. Zakres ustalamy od 0 do 10. Tak skonfigurowany blok przesuwamy na blok Picture Display i dostosowujemy jego wymiary. Jeżeli chcemy przesunąć któryś z bloków na wierzch należy kliknąć prawym klawiszem myszy na bloku i wybrać z menu kontekstowego opcję **Bing to Top** lub **Bing to Forward**.

Teraz pozostaje już tylko dodać blok Slider ze zmienną lokalną \$U100 i uruchomić program.

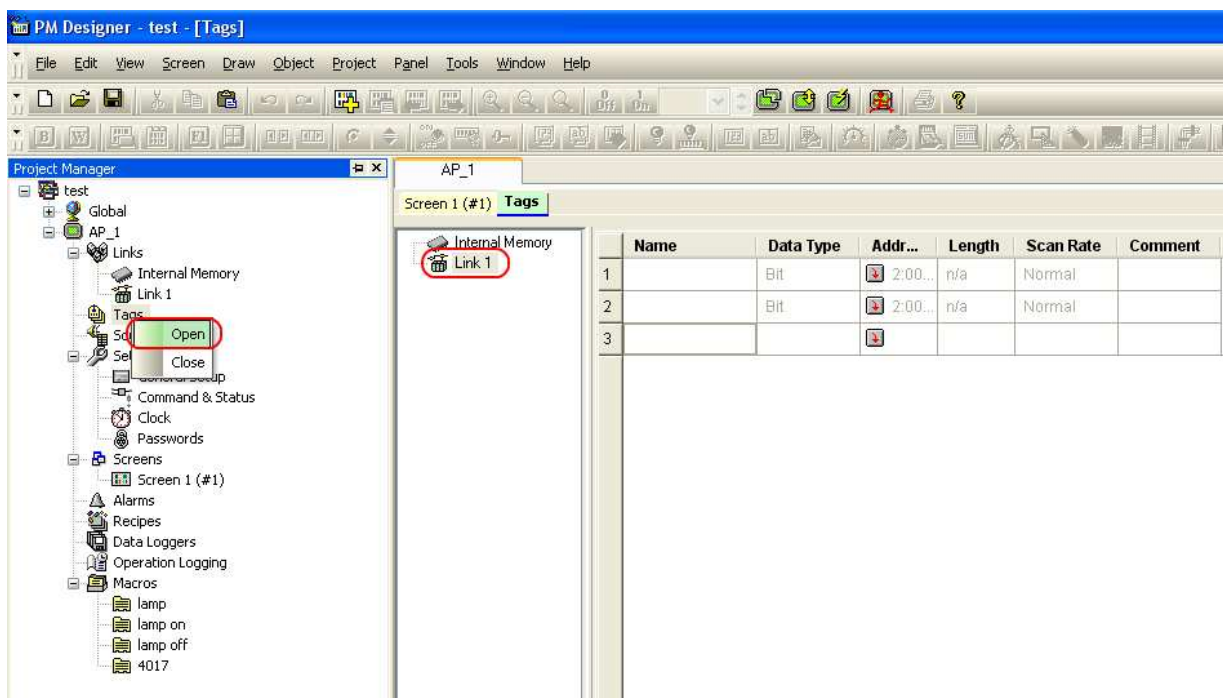


Rysunek 28. Widok połączonych bloków Picture Display z Bar Graph

4. Dodawanie Tagów

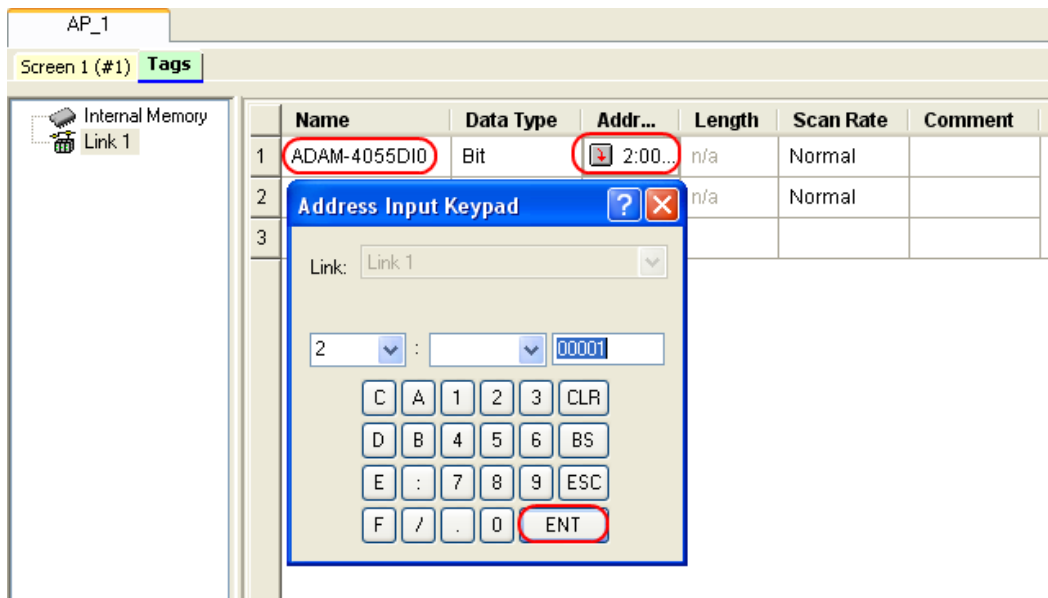
Innym sposobem adresacji urządzeń są wbudowane w programie znaczniki Tag. Są one o tyle korzystne że zdefiniowany Link oraz adres urządzenia kojarzone są bezpośrednio z przypisaną programowo zmienną. W dalszym procesie tworzenia aplikacji zmienna ta może być wykorzystywana wielokrotnie bez konieczności każdorazowego definiowania adresu urządzenia.

W pierwszym kroku w drzewie projektu klikamy prawym klawiszem myszy na znacznik **Tags** i otwieramy listę dostępnych znaczników. Dwukrotnie klikamy na interesujące nas połączenie (u nas Link 1) i w polu **Name** wprowadzamy nazwę zmiennej. W polu **Data Type** określamy typ zmiennej a w **Address** podajemy odpowiedni adres urządzenia.



Rysunek 29. Widok tabeli Tagów

W poniższym przykładzie dodajemy dwie zmienne połączone z wejściem i wyjściem modułu ADAM-4055 o adresie 2. Zmienna ADAM-4055DI0 powiązana zostanie z wejściem DI0 o rejestrze 00001 a zmienna ADAM-4055DO0 z wyjściem DO0 o rejestrze 00017.

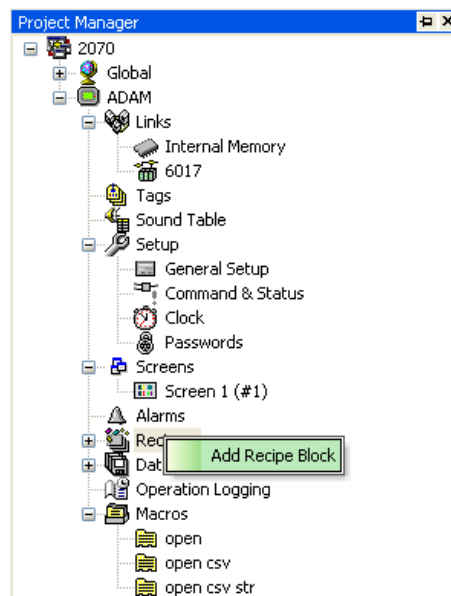


Rysunek 30. Definiowanie adresu zmiennej Tags

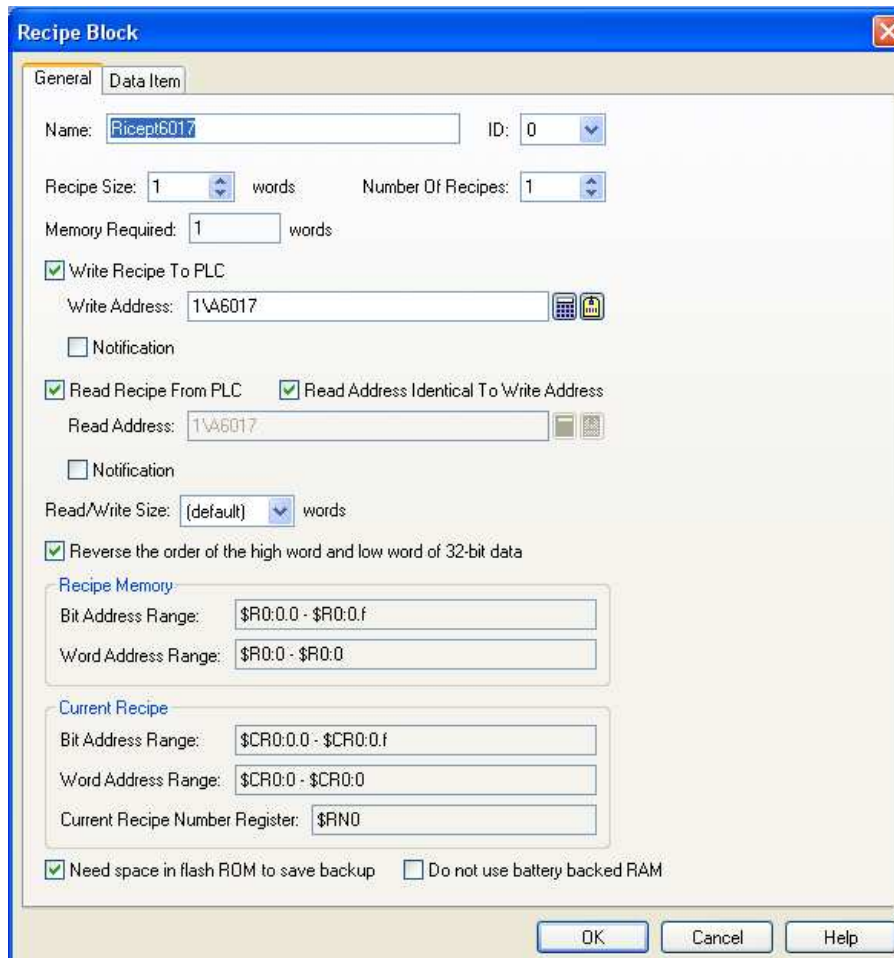
5. Receptury

Receptury pozwalają na przechowywanie danych w zarezerwowanym obszarze pamięci panelu operatorskiego. Odczyt oraz zapis danych dokonywany jest ze wskazanych na etapie konfiguracji adresów i może być rozszerzany o dowolne urządzenia podłączone do WOPa np. sterowniki PLC lub moduły ADAM.

W drzewie projektu klikamy prawym klawiszem myszy na Recipes i dodajemy nową recepturę. Następnie dwukrotnie klikamy na powstałą zmienną i wchodzimy jej ustawienia **General**. W poniższym przykładzie do receptury przypisano zmienną Tags A6017 powiązaną z modułem wejść analogowych ADAM-6017.



Rysunek 31. Położenie Receptur w drzewie projektu



Rysunek 32. Zakładka General w Recipe Block

W zakładce General wyróżniamy pola:

Name - nazwa recepty.

ID - identyfikator recepty.

Recipe size - wielkość recepty (0 - 1023).

Number of recipes - liczba recept (0 - 65535).

Memory Required - automatycznie przypisana pamięć do recepty.

Write Recipe To PLC - adres rejestru sterownika PLC do którego ma być zapisana recepta.

Read Recipe From PLC - adres rejestru z którego ma być odczytana receptura (będzie taki sam co rejestr zapisywany w przypadku zaznaczenia pola **Read Address Identical To Write Address**)

Read Address - adres rejestru sterownika PLC, z którego ma być odczytana recepta.

Recipe Memory - obszar pamięci zarezerwowany dla receptury.

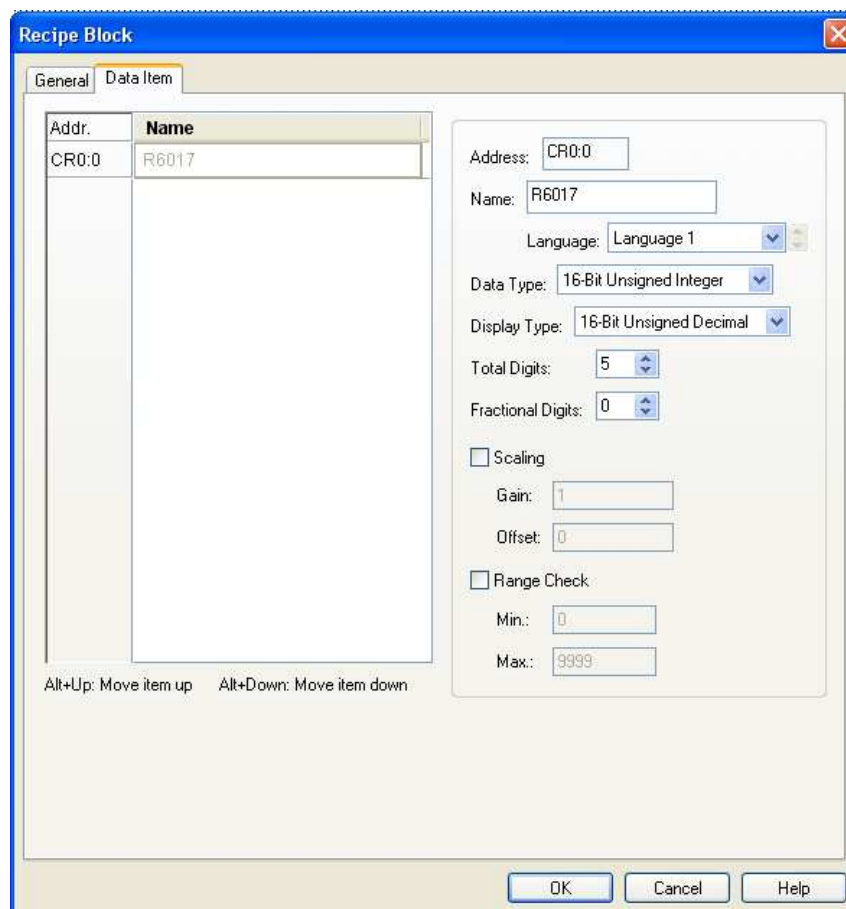
Current Recipe - obszar pamięci aktualnej receptury.

Need space in flash ROM to save backup – opcja ta umożliwia zapis danych z receptury do pamięci zewnętrznej poprzez port USB

Zapis receptury może zostać zrealizowany w połączeniu z blokiem funkcyjnym **Function Button** **F1** z wybraną funkcją **Save Recipe Data to Flash ROM** lub **Save Recipe Data to File** (zapis do zdefiniowanego pliku csv lub txt w pamięci zewnętrznej na USB np. pendrive).

Odczyt receptury odbywa się przy pomocy bloku funkcyjnego z funkcją **Read Recipe from Controller** (każdorazowy odczyt ze wszystkich receptur po kliknięciu przycisku).

W zakładce **Data Item** pojawiają się kolejne przebiegi uzależnione od wielkości recepty Rcept Size.



Rysunek 33. Zakładka Data Item w Recipe Block

W zakładce Data Item wyróżniamy pola:

Name - nazwa przebiegu.

Language - wybór języka.

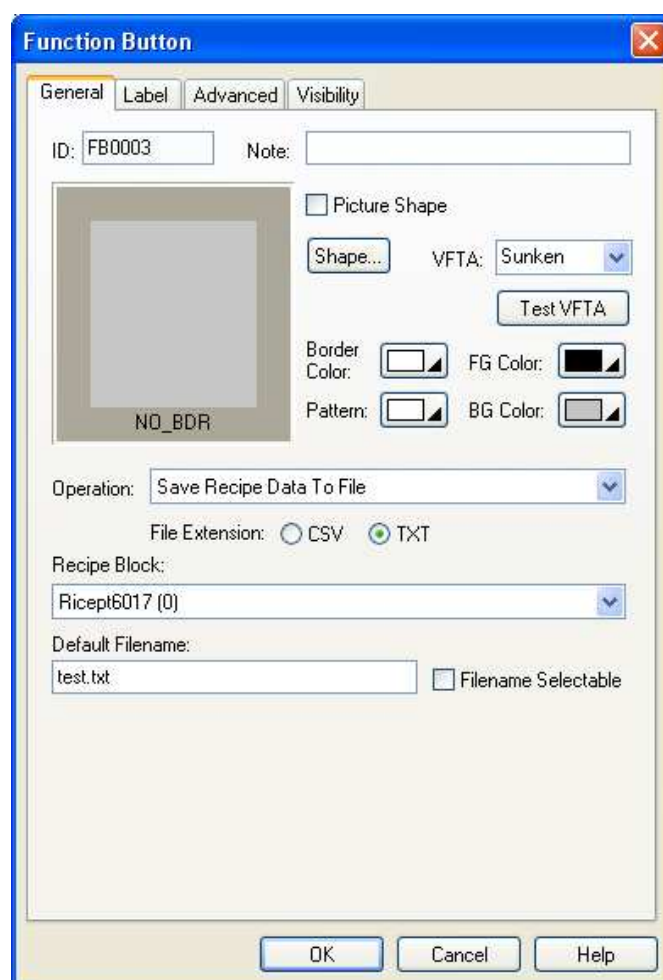
Data Type - typ zmiennej.

Display Type - typ zmiennej wyświetlanej.

Total Digits - liczba wyświetlanych cyfr.

Fractional Digits - Liczba wyświetlanych cyfr po przecinku.

Scaling - skalowanie według równania $Y=aX+b$ gdzie **Y** to wartość wyświetlana, **X** to wartość odczytana, **a** to wzmocnienie, **b** to Offset.

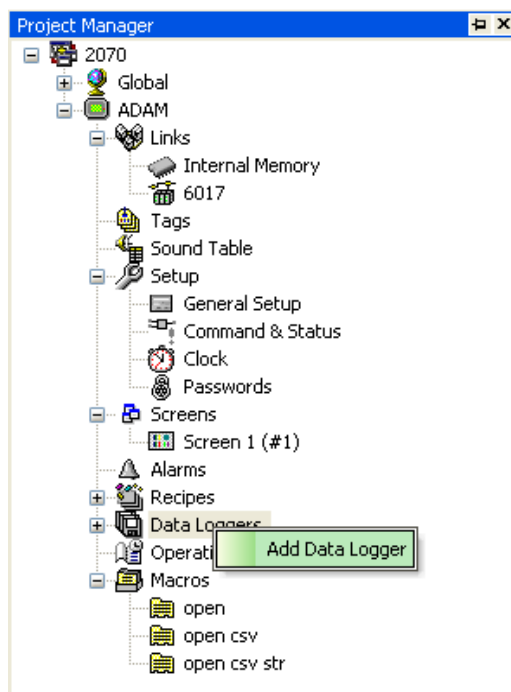


Rysunek 34. Wykorzystanie Bloku Function Button umożliwiające zapis receptury do zdefiniowanego pliku.

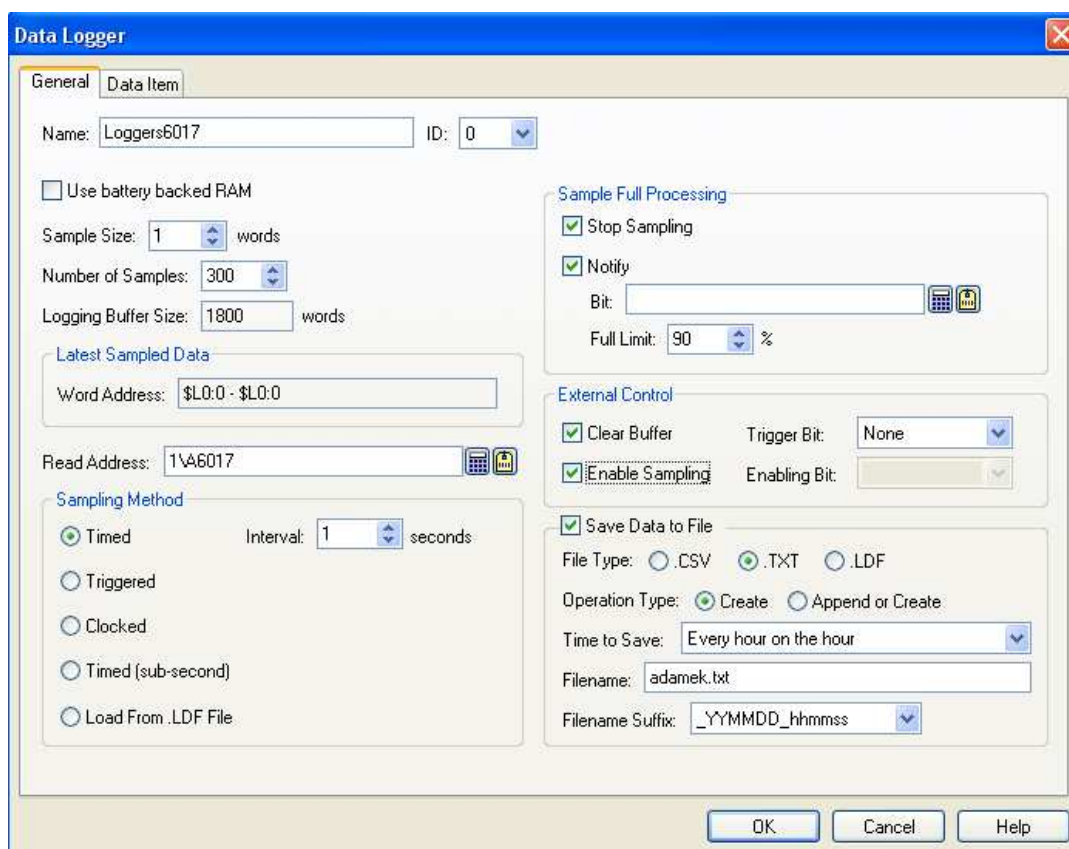
6. Logowanie danych

6.1 Dodawanie i konfiguracja

Moduł **Data Logger** umożliwia wprowadzanie danych do pamięci wewnętrznej panelu operatorskiego. Aby skorzystać z tej funkcji w pierwszej kolejności w drzewie projektu wyszukujemy pole **Data Logger**. Klikając prawym klawiszem myszy dodajemy moduł do archiwizacji danych do projektu – wybieramy **Add Data Logger**. Następnie dwukrotnie klikając myszą wchodzimy do ustawień modułu. W poniższym przykładzie do modułu przypisano zmienną Tags A6017 powiązaną z modułem wejść analogowych ADAM-6017.



Rysunek 35. Dodawanie modułu Data Loggera do projektu



Rysunek 36. Ustawienia modułu Data Logger w zakładce General

W zakładce General wyróżniamy pola:

Name - nazwa modułu.

ID - identyfikator.

Use battery backed RAM - zapis danych do pamięci z podtrzymaniem baterijnym.

Sample Size - wielkość próbki (jedna próbka ma 16 bitów).

Number of Samples - ilość próbek.

Logging Buffer Size - ilość pamięci potrzebnej dla danych archiwalnych (obliczana automatycznie).

Word Address - obszar pamięci przydzielany automatycznie.

Read address - adres pamięci z którego będą pobierane dane.

Sampling Method:

Timed - próbkowanie w określonych odstępach czasu (w sekundach od 0 do 65535).

Triggered - próbkowanie wyzwalane przez bit **Trigger Bit** zawarty w module **Command & Status**.

Clocked - wyzwalanie przez zegar systemowy (w minutach od 1 do 60).

Timed - próbkowanie z określonym interwałem czasowym (w sekundach od 0.1 do 0.9s).

Sample Full Processing:

Stop Sampling - zatrzymanie próbkowania w przypadku przepełnienia bufora danych.

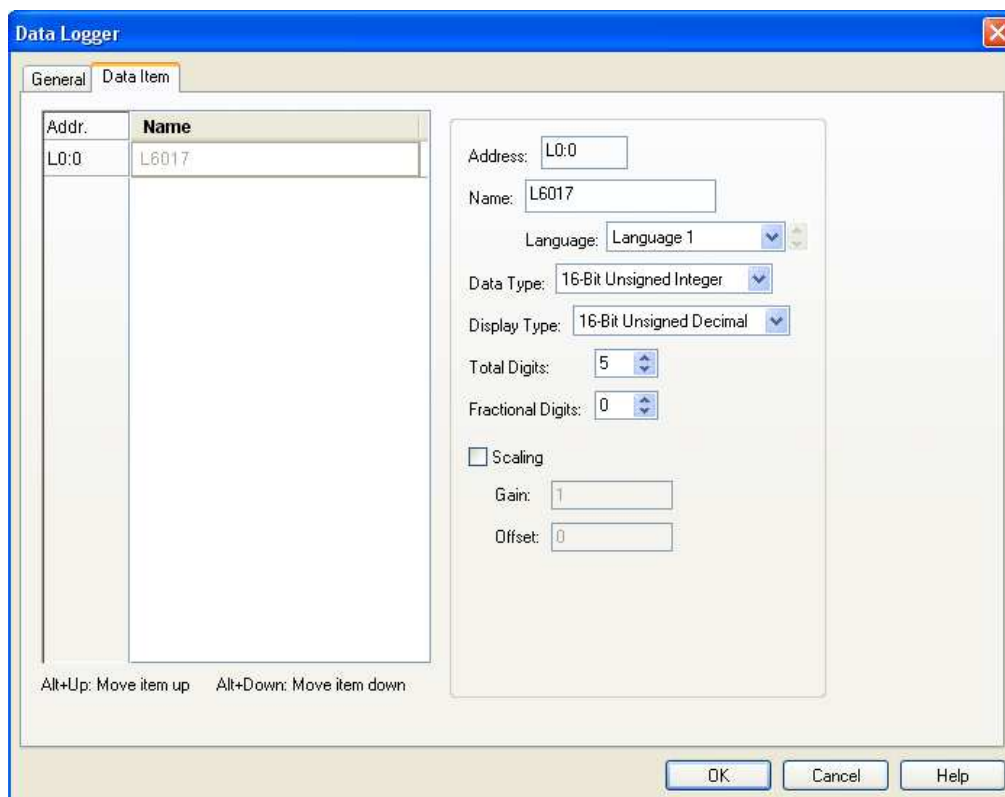
Notify - w przypadku przepełnienia bufora zmienia się stan w określonym bicie lub zmiennej.

External Control:

Clear Buffer - czyszczenie bufora danych wyzwalane przez bit **Trigger Bit** zawarty w module **Command & Status**.

Enable sampling - wyzwolenie próbkowania przez bit **Enabling Bit** zawarty w module **Command & Status**.

Save to file - zapisuje dane do wybranego pliku w zdefiniowanych cyklach czasowych.



Rysunek 37. Ustawienia modułu Data Logger w zakładce Data Item

W zakładce Data Item wyróżniamy pola:

Name - nazwa przebiegu.

Language - wybór języka.

Data Type - typ zmiennej.

Display Type - typ zmiennej wyświetlanej.

Total Digits - liczba wyświetlanych cyfr.


Fractional Digits - Liczba wyświetlanych cyfr po przecinku.

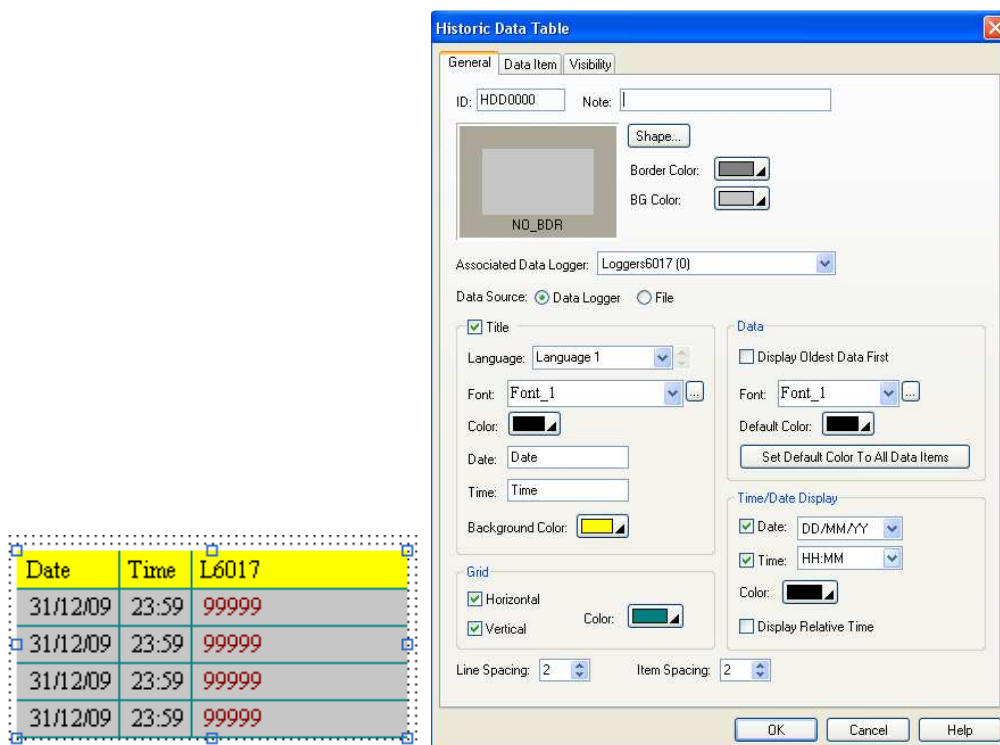
Scaling - skalowanie według równania $Y=aX+b$ gdzie **Y** to wartość wyświetlana, **X** to wartość odczytana, **a** to wzmacnienie, **b** to Offset.

6.2 Tabele i wykresy wykorzystujące dane typu Data Logger

Do obsługi modułu Data Logger można wykorzystać bloki **Historic Data Table**  oraz **Historic Trend Graph** .


6.2.1 Historic Data Table

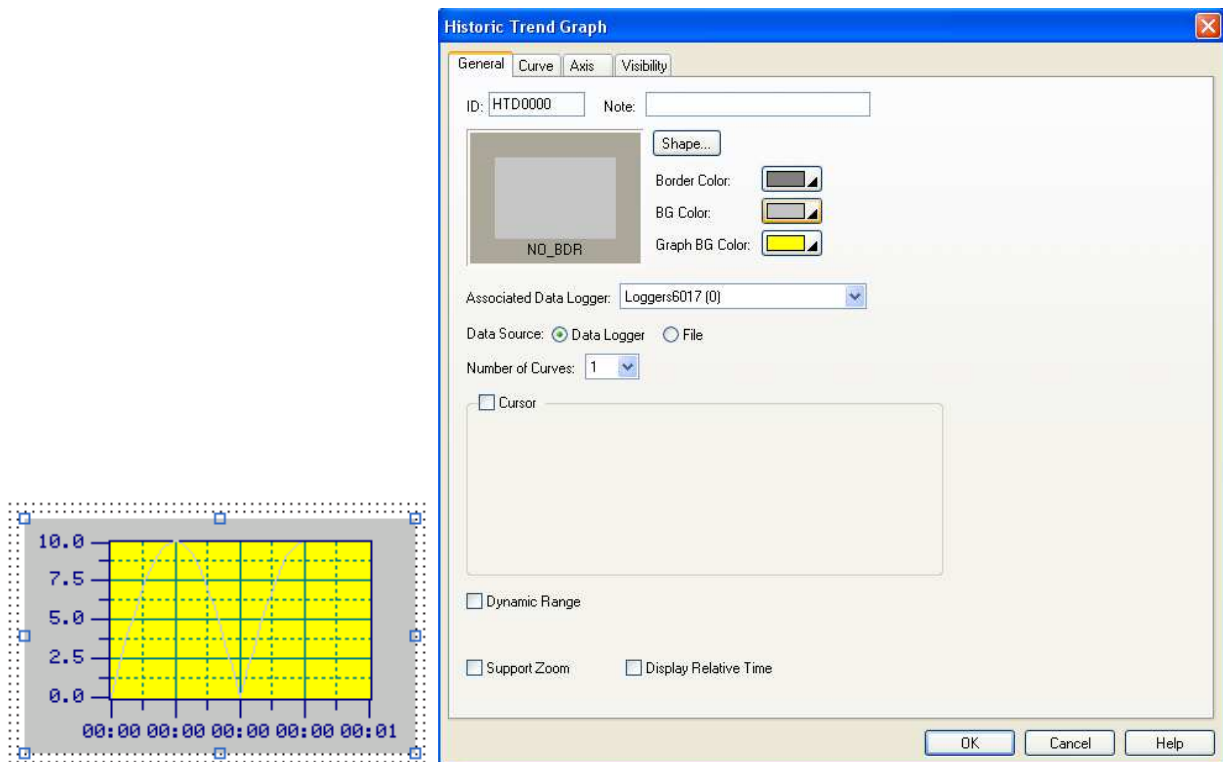
Blok **Historic Data Table** pozwala na wyświetlanie danych w tabeli ze zdefiniowanego modułu Data Loggers z określonym interwałem czasowym. W pierwszej kolejności wstawiamy blok  do naszej aplikacji i wchodzimy w ustawienia poprzez dwukrotne kliknięcie myszą. W polu **Associated Data Loggers** wybieramy odpowiedni moduł Data Loggers. Po uruchomieniu na ekranie uzyskamy cykliczny odczyt z wejścia analogowego AIO z modułu ADAM-6017, przy pomocy protokołu Modbus TCP i sieci Ethernet.



Rysunek 38. Ustawienia bloku Historic Data Table


6.2.2 Historic Trend Graph

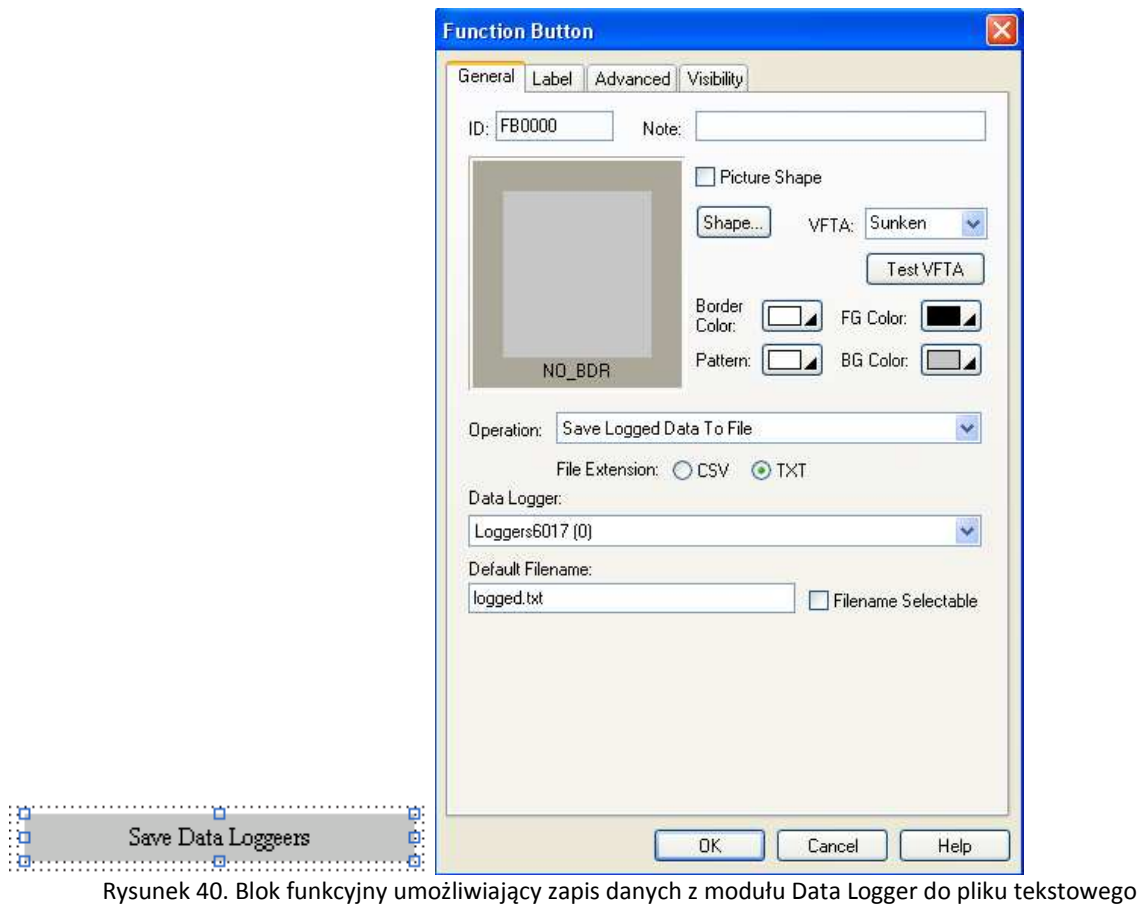
Blok Historic Trend Graph pozwala na wyświetlanie danych w postaci przebiegu ze zdefiniowanego moduły Data Loggers. W pierwszej kolejności wstawiamy blok  do naszej aplikacji i wchodzimy w ustawienia poprzez dwukrotne kliknięcie myszą. W polu Associated Data Loggers wybieramy odpowiedni moduł Data Loggers. W zakładkach **Curve** i **Axis** możemy zdefiniować zakres odczytu danych oraz wygląd przebiegu. Po uruchomieniu na ekranie uzyskamy cykliczny odczyt z wejścia analogowego AIO z modułu ADAM-6017 przy wykorzystaniu protokołu Modbus TCP i sieci Ethernet.



Rysunek 39. Ustawienia bloku Historic Trend Graph

6.2.3 Zapis danych typu Data Logger do pliku tekstowego



Aby zapisać dane typu Data Logger należy skorzystać z bloku funkcyjnego **Function Button**  i wybrać funkcję **Save Logged Data To File**. W ustawieniach należy określić odpowiedni Data Logger, typ pliku tekstowego oraz podać jego nazwę. Dane będą zapisywane do pliku w pamięci zewnętrznej (np. Pendrive) przy każdorazowym aktywowaniu przycisku Function Button.



Rysunek 40. Blok funkcyjny umożliwiający zapis danych z modułu Data Logger do pliku tekstowego

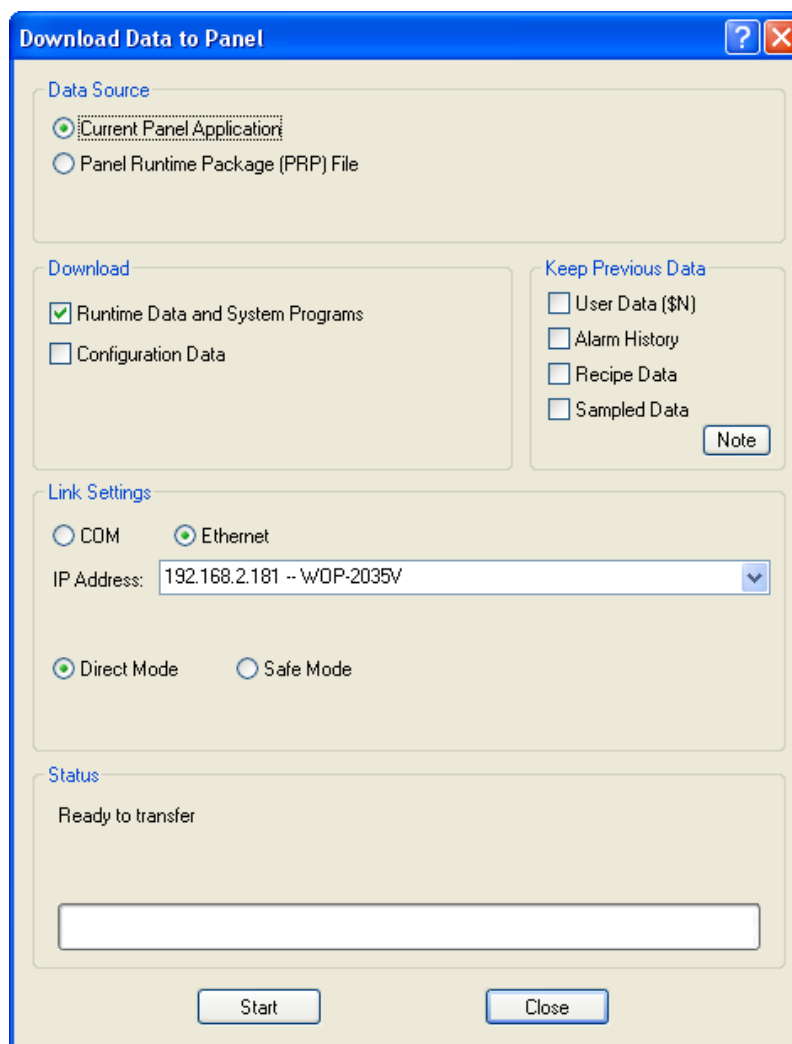
7. Uruchamianie i testowanie

7.1 Kompilacja i wgrywanie projektu do sterownika

Procesu kompilacji można dokonać z poziomu paska skrótów naciskając guzik **Compile**  lub z poziomu menu projektu wchodząc w **Panel->Compile**. W przypadku wystąpienia błędów w czasie kompilacji odpowiednie adnotacje pojawią się w polu **Build List**. Po prawidłowym skompilowaniu należy wgrać projekt do sterownika naciskając guzik **Download**  lub wybrać z menu projektu **Panel->Download**. Tutaj w ustawieniach **Link Settings** wybieramy port COM, w przypadku transmisji po porcie szeregowym, lub port Ethernet podając adres IP panelu operatorskiego.




Rysunek 41. Zakładka Toolbar

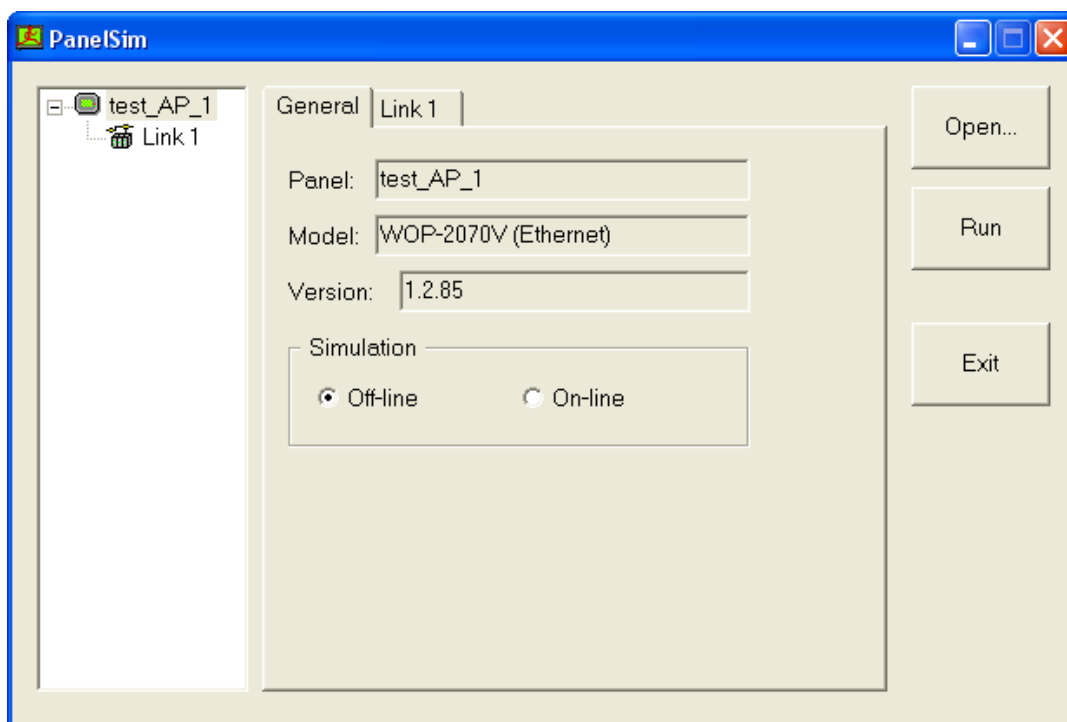


Rysunek 42. Download projektu

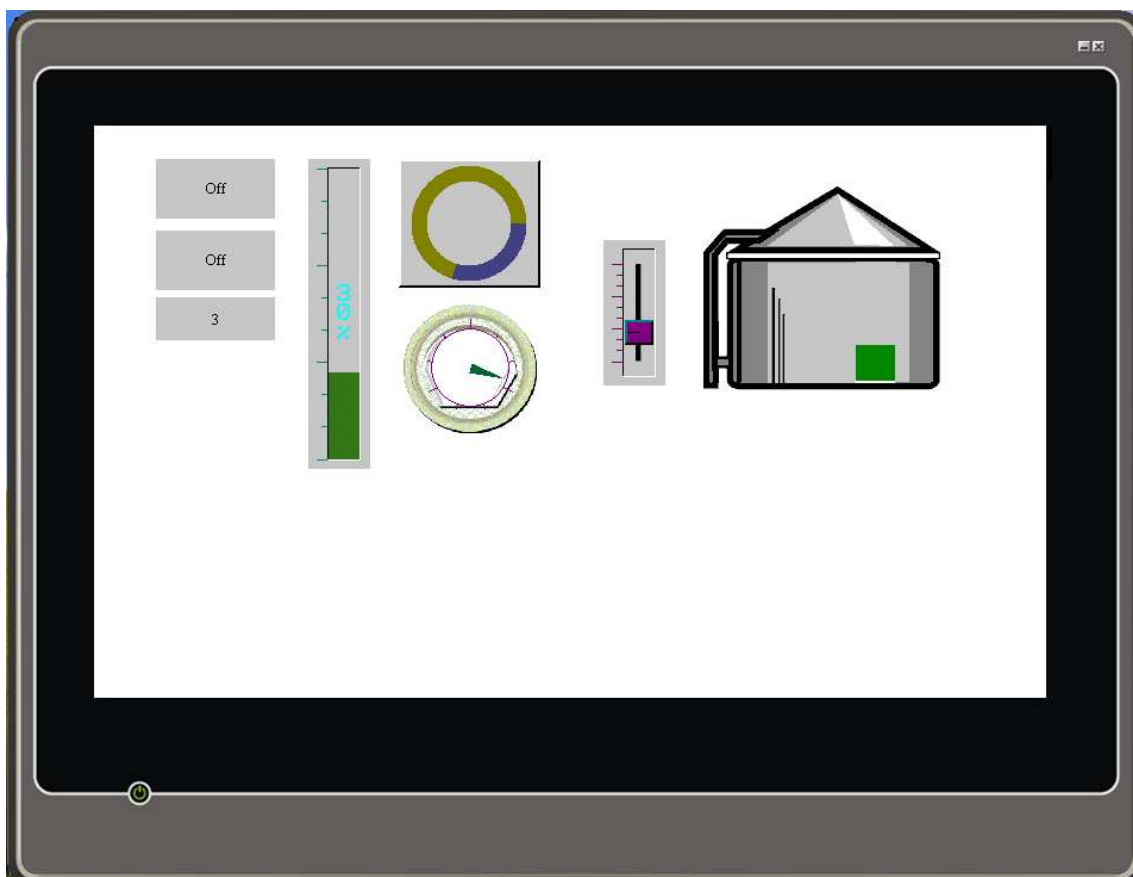
7.2 Testowanie i symulacja

Środowisko WebOP Designer udostępnia również bardzo wygodne funkcje symulacyjne, które mogą być użyte przed załadowaniem projektu do paneli z serii WOP-2000. Wyróżniamy tu dwa tryby symulacyjne Online i Offline.

Aby wejść w tryb symulacyjny **Offline** należy uaktywnić przycisk  lub wybrać z menu programu **Tool->Run Offline Symulation**. W tym trybie symulacja będzie odbywała się bez komunikacji ze zdefiniowanymi połączeniami typu Link. W przypadku wyboru symulacji **Online** (należy wybrać z menu **Tool->Run Online Symulation**) możliwe jest również przetestowanie połączeń z urządzeniami zewnętrznymi - pełna symulacja pracy tak jak na panelu operatorskim.



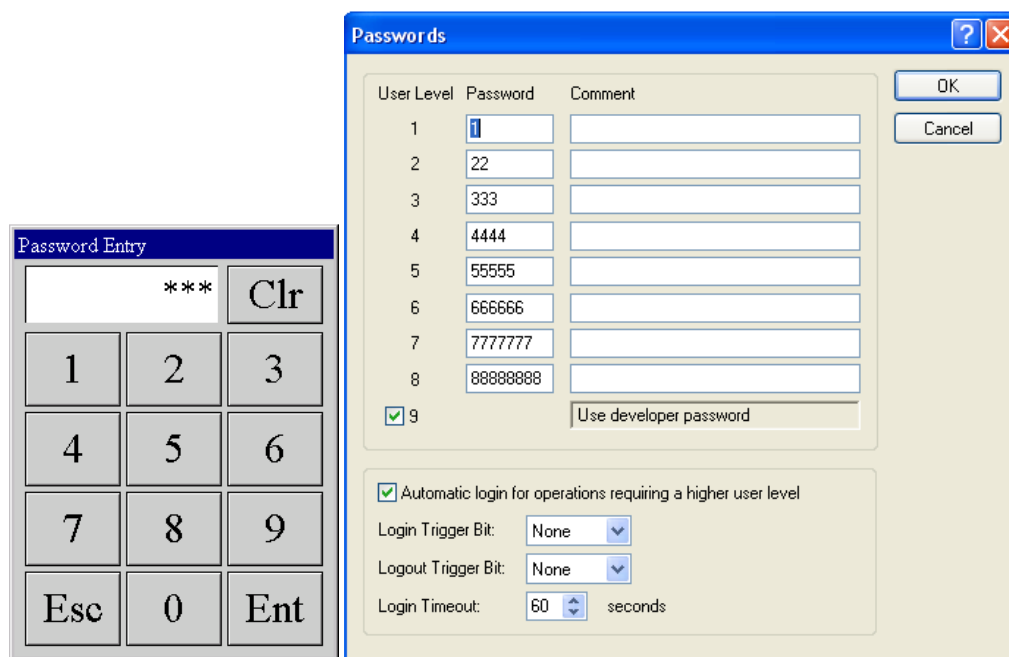
Rysunek 43. Symulacja w trybie Offline



Rysunek 44. Widok aplikacji w trybie symulacyjnym

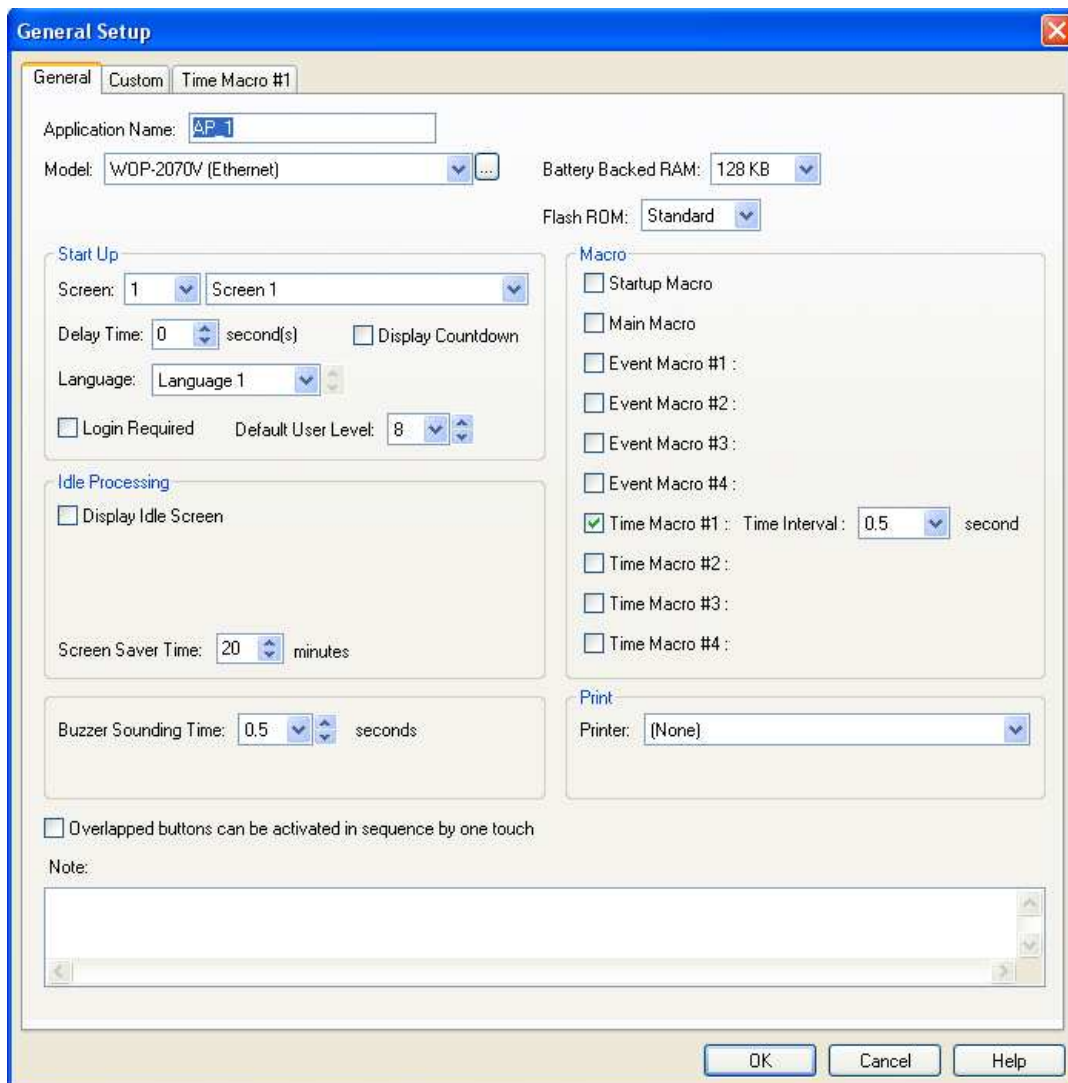
8. Poziomy dostęp do aplikacji panelu

WebOP Designer definiuje 9 (0-8) poziomów dostępu użytkowników, zróżnicowanych pod względem przypisanych uprawnień takich jak dostęp do bloków czy ekranów. Do każdego z poziomów przypisywane jest unikatowe hasło o różnym rozmiarze (poziom 0 nie ma zdefiniowanego hasła i ma najmniejsze możliwości kontroli uprawnień). Aby zmienić domyślnie zdefiniowane hasło należy wejść w menu programu **Panel->Passwords**. Zaznaczenie pola **Automatic login for operation requiring a higher user level** spowoduje że uaktywnienie bloku o wyższym poziomie dostępu automatycznie uruchomi okno autoryzacji **Password Entry**.



Rysunek 45. Hasła przypisane do poszczególnych poziomów użytkownika oraz pole Password Entry

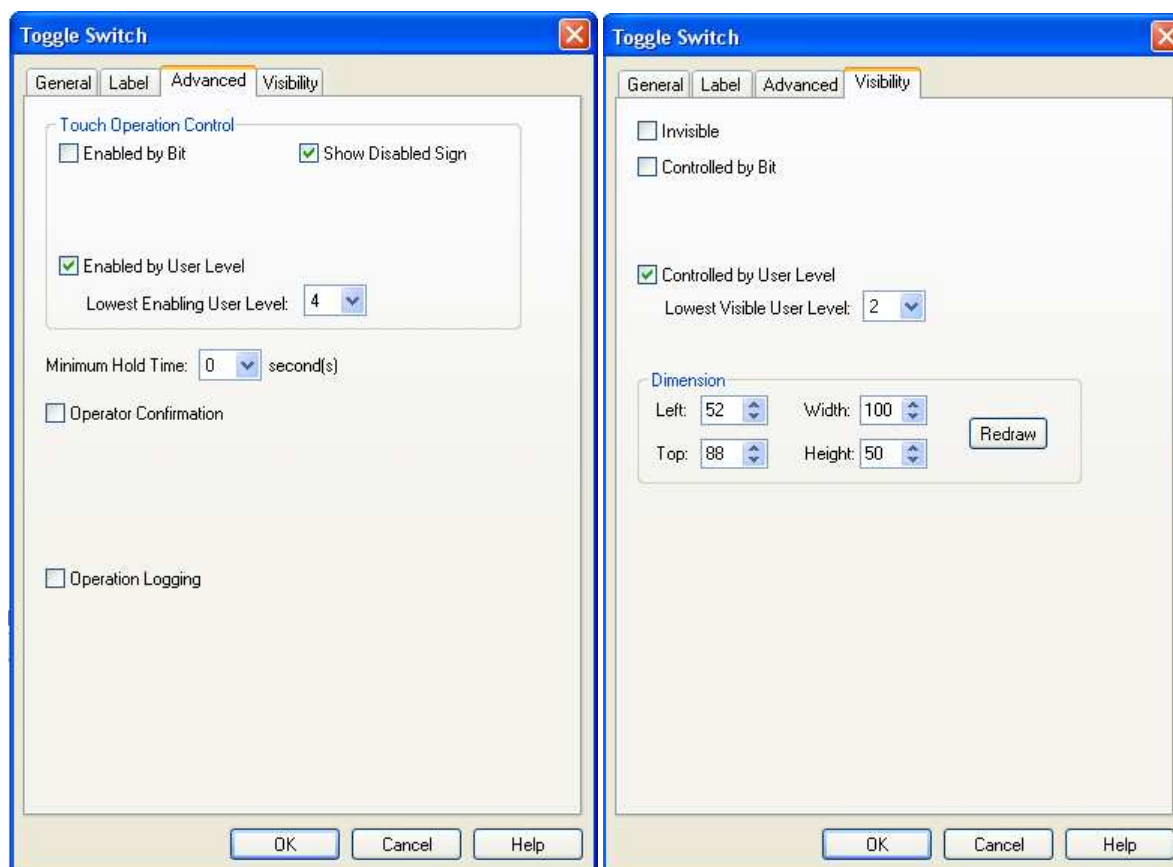
Ustawienia oraz aktywację poziomów dokonuje się z poziomu menu projektu **Panel->General Setup**. Zaznaczając pole **Default User Level** ustalamy domyślnie poziom projektu. W przypadku zaznaczenia pola **Login Required** przy starcie aplikacji wymagana będzie autoryzacja poprzez podanie przypisanego danemu poziomowi hasła.



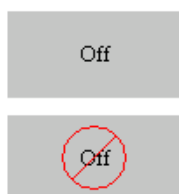
Rysunek 46. Konfiguracja poziomu użytkownika w projekcie

Jeżeli chcemy zróżnicować uprawnienia danym użytkownikom należy odpowiednio wcześniej skonfigurować poszczególne bloki w ich ustawieniach w zakładce **Advanced** (o ile jest możliwa) oraz **Visibility**. Aby zmienić prawa dostępu do danego bloku wchodzimy w ustawienia **Advanced** i zaznaczamy opcję **Enable by User Level**, a w polu **Lowest Available User Level** podajemy poziom dostępu do bloku. W przypadku, gdy zalogowany użytkownik ma niższe prawa niż zdefiniowany blok, po jego uaktywnieniu konieczne będzie podanie ponownie hasła o wyższym priorytecie. W każdej chwili można zmienić poziom przy pomocy bloku funkcyjnego z funkcją **Log In**.

W każdym bloku możliwe jest ustalenie widoczności w zależności od poziomu użytkownika. Konfiguracja odbywa się w ustawieniach bloku w zakładce **Visibility**. Poprzez zaznaczenie pola **Controlled by User Level** i wybranie liczby w polu **Lowest Available User Level** ustalamy najniższy poziom zezwalający na widoczność bloku.



Rysunek 47. Konfiguracja dostępu w bloku Toggle Switch

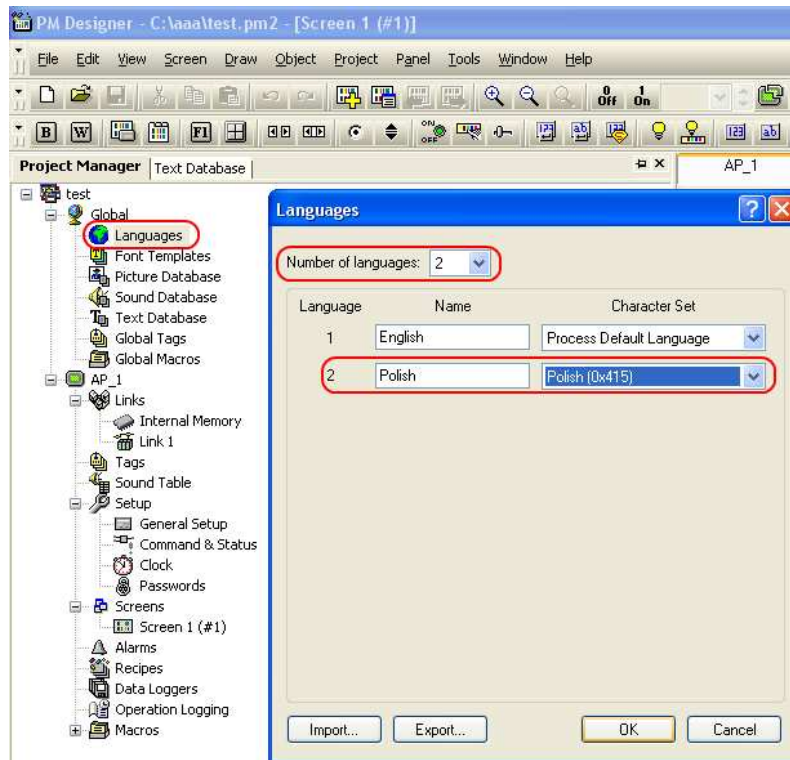


Rysunek 48. Widok zablokowanego bloku

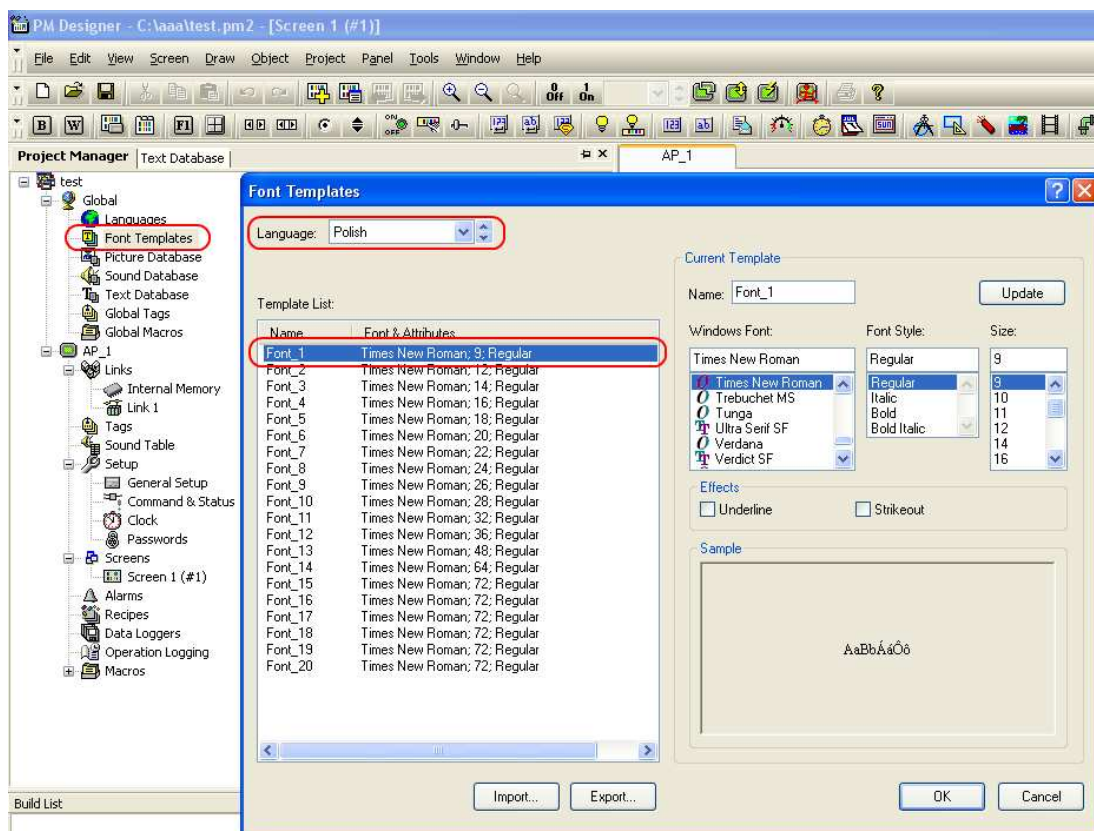
9. Languages

Jedną z podstawowych funkcji WebOP Designer jest możliwość definiowania różnych języków w ramach jednego projektu. Jest to o tyle wygodne, że pozwala na szybkie przetłumaczenie wszystkich słów występujących w danej aplikacji, przy pomocy jednego przycisku funkcyjnego. I tak w jednym projekcie można utworzyć do 10 różnych języków z przypisaną do niego unikalną bazą słów.

Po utworzeniu projektu domyślnie przypisany zostaje tylko język angielski. Aby dodać kolejne języki należy w drzewie projektu wybrać **Languages**, podać ilość języków w polu **Number of languages** i zdefiniować je w polu **Character Set**. Aby ustawić style danego języka należy wejść w **Font Templates** w drzewie projektu.

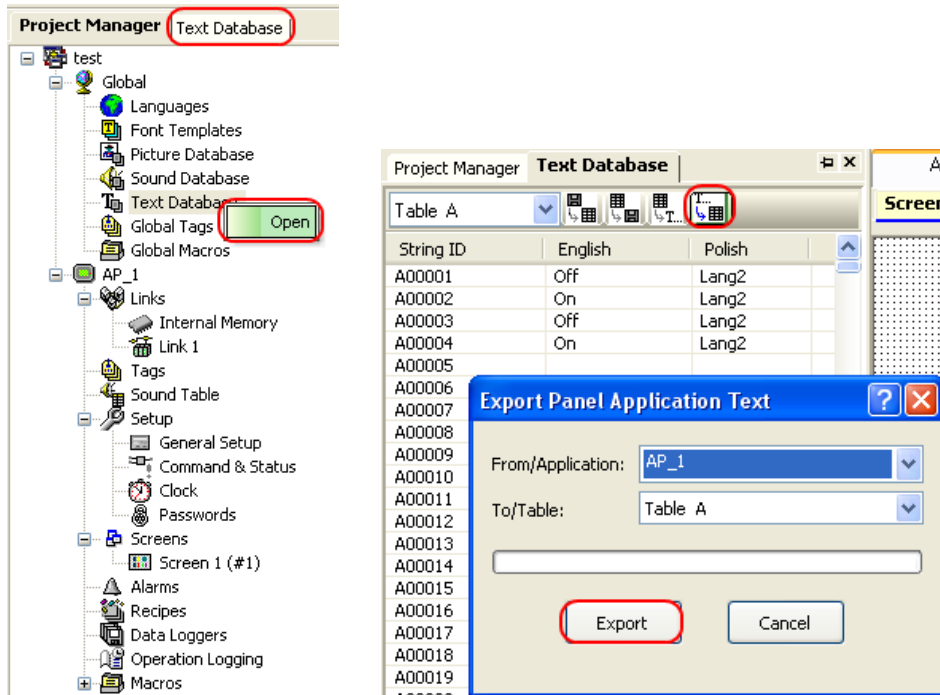


Rysunek 49. Konfiguracja dostępnych języków w projekcie.

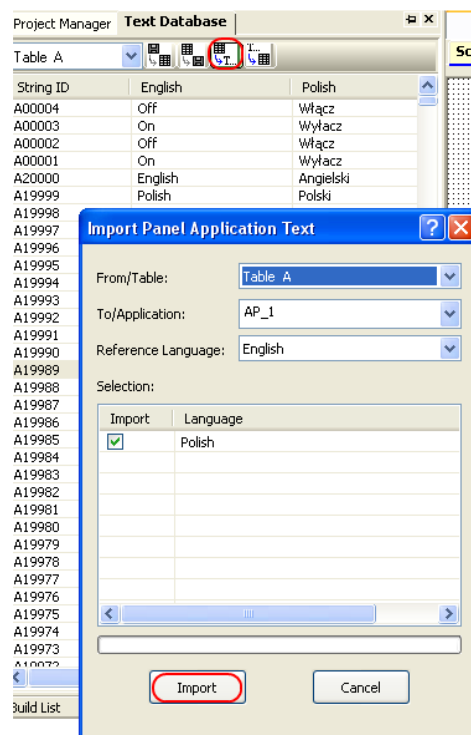


Rysunek 50. Przypisanie stylu do danego języka.



Tłumaczenie słów odbywa się z poziomu **Text Database** umieszczonym w drzewie projektu. W ustawieniach klikamy **Export Panel Application Text** i wprowadzamy kolejne tłumaczenia słów występujących w naszym projekcie. Po przetłumaczeniu klikamy **Import Panel Application Text** co spowoduje aktualizację językowe.




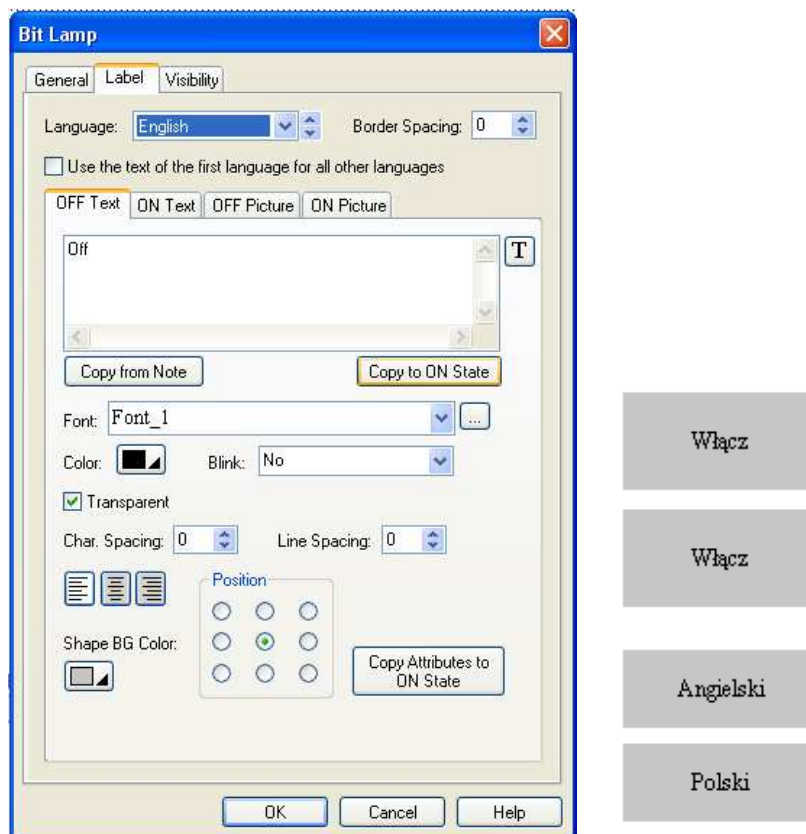
Rysunek 51. Tłumaczenie słów występujących w projekcie na język Polski



Rysunek 52. Import tłumaczeń do bloków

W naszym przykładzie skorzystamy już z wcześniej dodanych bloków Bit Lamp  i Toggle Switch . Klikamy dwukrotnie na blok Bit Lamp i wchodzimy w zakładkę **Label**. Tam możemy wprowadzić odpowiedni tekst pojawiający się na bloku w zależności stanu. W zależności od wybranego w polu **Language** języka możemy wprowadzić odpowiednie tłumaczenia oraz styl języka (my zrobiliśmy to już wcześniej z poziomu **Text Database**).

Podobnie postępujemy w przypadku bloku Toggle Switch. Następnie dodajemy dwa bloki funkcyjne  i w polu **Operation** wybieramy **Change language**. W jednym ustawiamy zdefiniowany język Language jako Polish a w drugim English. Po skompilowaniu i uruchomieniu w naszej aplikacji będą zmieniały się tłumaczenia w zależności od wybranej wersji językowej przycisków funkcyjnych.



Rysunek 53. Konfiguracja językowa w bloku Bit Lamp