

Opracował:

Rafał Józefowski
Politechnika Częstochowska
Wydział Inżynierii Mechanicznej i Informatyki
15.08.2001

Strona zadań

Program ten pozwala na budowanie bloków bez konieczności konwencjonalnego programowania. Pasek zadań zawiera bloki, które nanosi się na stronę zadań. Następnie należy dwukrotnie kliknąć na dany blok. Otwiera się wtedy strona ustawień charakterystyczna dla każdego bloku. Każdy z nich posiada charakterystyczne dla niego funkcje.

Metka (TAG block)

Ten blok jest używany do łączenia danego bloku zadań z blokiem wyświetlającym dane lub z wirtualną metką.



Dane mogą być podawane przez ten blok i używane z innymi blokami zadań. Dla wirtualnej metki nie trzeba łączyć jej bezpośrednio lecz tylko skonfigurować połączenie do niej na stronie ustawień. Przy takim połączeniu dane przenoszone przez wirtualną metkę mogą być stosowane z innymi blokami zadań.

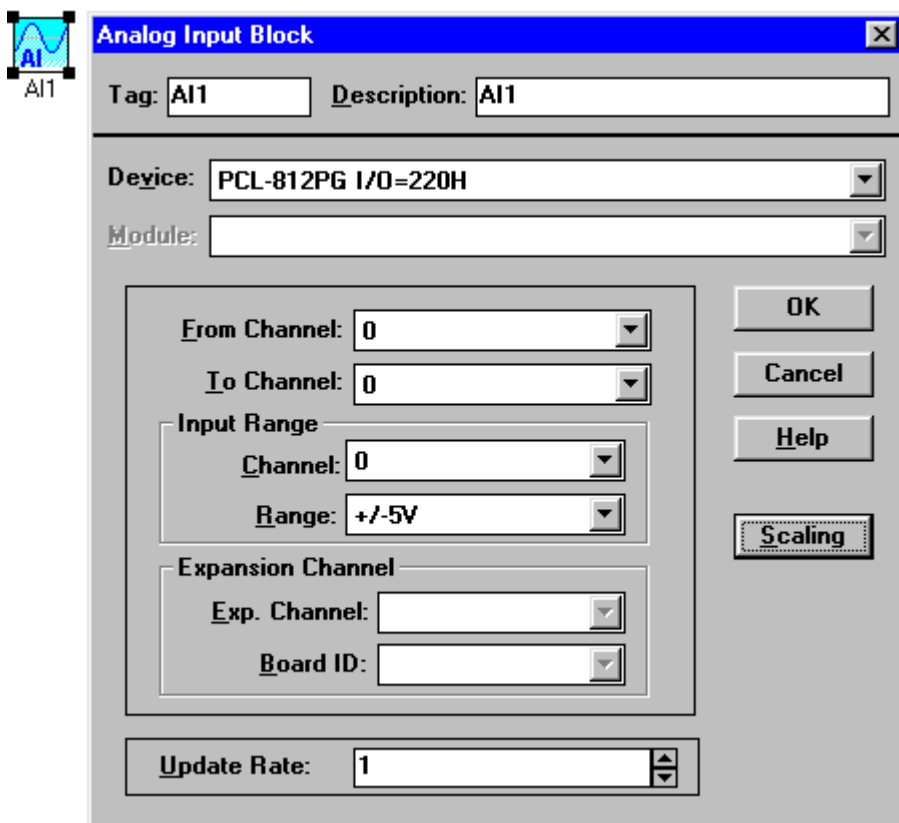
Wejścia. Wejść nie ma.

Wyjścia. Blok ten jedynie przekazuje wartości do podłączonego bloku.

Blok wejścia analogowego (Analog Input Block AI)

Ten blok służy do dostarczania sygnałów analogowych do innych bloków posiadających wejście analogowe.

Po dwukrotnym kliknięciu bloku wyświetla się strona ustawień.



Pole urządzeń (Device field)

Po kliknięciu strzałki rozwija się pasek z aktualnie zainstalowanymi urządzeniami. Po wybraniu jednego z nich zostanie on wyświetlony w tym polu.

Pole kanałów (Channel field)

Po wybraniu urządzenia lub modułu należy wybrać kanał, z którego będzie pobierany sygnał. Blok ten ma możliwość podłączenia do 16 kanałów zależnie od wybranego urządzenia.

W tym polu kanałów należy ustawić:

Z kanału (From channel) – zaznacza numer początkowego kanału, z którego będzie pobierany sygnał. Numer ten może mieć wartość od zera do maksymalnej ilości kanałów w wybranym wcześniej sterowniku.

Do kanału (To channel) – zaznacza numer końcowego kanału, z którego ma być pobierany sygnał. Numer ten musi być równy lub większy od numeru „Z kanału”

Wartość wejściowa (Input range) – Zakres wartości wejściowej dla danego urządzenia, np. +/-5V. Jeżeli jest ono wielokanałowe, należy wybrać zakres napięcia dla każdego z kanałów.

Częstotliwość próbkowania (Update rate field)

Zmniejszenie częstotliwości próbkowania, np. jeżeli standardowo jest to 100 Hz a potrzebna jest częstotliwość 20 Hz, wystarczy wpisać w to pole 5. Jednak w dalszym ciągu można uzyskiwać sygnał o częstotliwości próbkowania 100Hz, jeżeli połączy się blok z blokiem „Log file”, lub z blokiem wizualizacyjnym, jednak tylko jedna z pięciu próbek będzie „prawdziwa”, pozostałe będą tylko kopiami poprzednich. Liczba ta może przyjąć wartość od 0 do 32767.

Klawisz „Skalowanie” (Scaling)

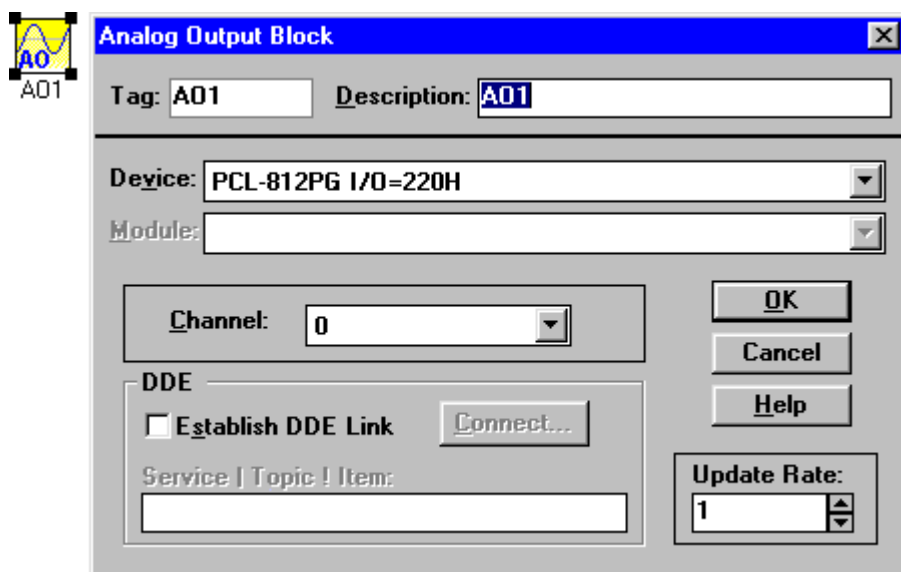
Funkcja ta umożliwi przeskalowanie zakresu sygnału z zakresu wejściowego do pożądanego zakresu. Np. jeżeli zakres wejściowy jest +/-5V, można go przeskalować na 0-10V.

Wejścia. Wejść nie ma. VisiDAQ wyświetli komunikat „Cannot accept input”

Wyjścia. Wszystkie wybrane kanały muszą być połączone z odpowiednimi blokami. Maksymalna ilość kanałów wyjściowych 16.

Blok wyjścia analogowego (Analog output block AO)

Blok ten posiada wejścia umożliwiające wprowadzenie analogowego sygnału z innego bloku, a następnie przekazania go do analogowej sekcji wyjściowej urządzenia.



Pole urządzeń (Device field)

Po wybraniu tego pola wszystkie aktualnie zainstalowane urządzenia zostaną wyświetlone. Po wybraniu jednego z nich zostaną wyświetlone dostępne dla niego kanały. Można tu wybrać wyjście analogowe [hardware], do którego zostanie podłączony blok wyjścia analogowego (jeden blok do jednego kanału urządzenia).

Pole kanałów (Channel field)

Zaznacza się tu kanał urządzenia, do którego będzie wysyłany sygnał z AO. Ilość kanałów jest zależna od dostępnych w danym urządzeniu.

Częstotliwość próbkowania (Update rate field)

Pole to przeznaczone jest do zmniejszenia częstotliwości próbkowania. Np. jeśli cała aplikacja pracuje przy częstotliwości 100 Hz, natomiast potrzebna jest częstotliwość 20 Hz, należy w polu tym wpisać 5.

Pole DDE – dynamiczne przetwarzanie danych (DDE field – Dynamic Data Exchange)

Dynamiczne przetwarzanie danych pozwala na dostosowanie danych do komunikacji między aplikacjami VisiDAQ a innymi aplikacjami Windows. Pozwala to na bezpośrednie połączenie przez to pole z dowolnym urządzeniem.

Wejścia. Blok AO pobiera dane z innego bloku i przekazuje je do podłączonego urządzenia. Blok posiada możliwość podłączenia tylko jednego źródła sygnału. Jeśli źródło zostało wcześniej podłączone, zostanie wyświetlony komunikat „The input already connected”.

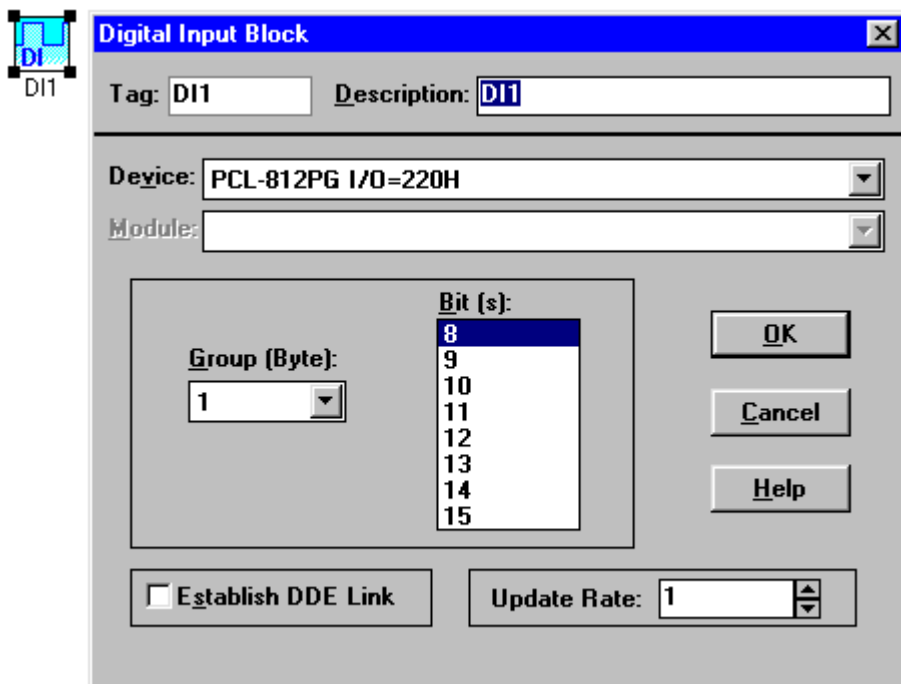
Wyjścia. Blok Przekazuje przyjęte dane bezpośrednio do podłączonego bloku.

Blok wejścia cyfrowego (Digital input block)

Blok ten posiada zdolność dostarczania do bloków z cyfrowym wejściem sygnału z sekcji cyfrowej urządzenia.

Pole urządzenia (Device field)

Po wybraniu tego pola wszystkie aktualnie zainstalowane urządzenia o wyjściach cyfrowych zostaną automatycznie wyświetlone. Po wybraniu jednego z nich zostaną wyświetlone dostępne dla danego urządzenia kanały (byte); należy wybrać jeden z nich, a następnie dostępny dla danego kanału bit lub bity (można ich wybrać więcej, niż jeden)



Częstotliwość próbkowania (Update rate field).

Pole to służy do zmiany (zmniejszenia) częstotliwości próbkowania. Żądaną wartość częstotliwości jest wynikiem dzielenia aktualnej częstotliwości (100 Hz) przez wpisaną w to pole wartość.

Połączenie DDE (Establish DDE link field)

Blok DI również posiada zdolność do dostosowywania danych z aplikacji VisiDAQ do komunikacji z innymi aplikacjami Windows

Wejścia. Nie ma wejść. VisiDAQ wyświetli komunikat „Cannot accept input”

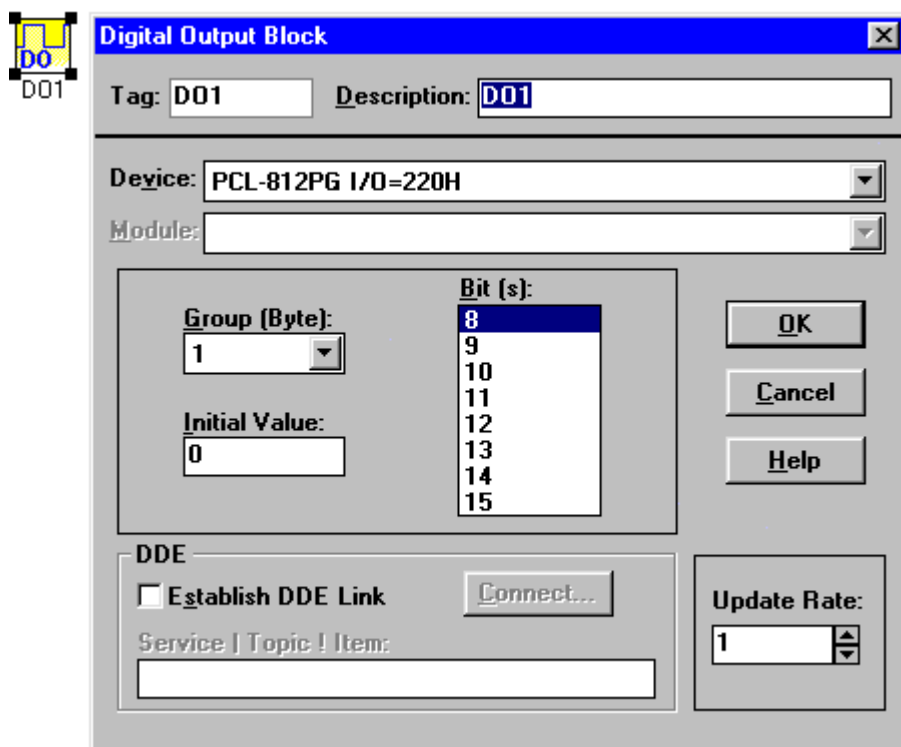
Wyjścia. Przekazuje dane ze sterownika do podłączonego bloku.

Blok wyjścia cyfrowego (Digital output block DO)

Blok ten pozwala na pobieranie danych cyfrowych z innego bloku i przekazywanie ich do sekcji wyjścia cyfrowego wybranego urządzenia.

Pole urządzenia (Device field)

Po wybraniu tego pola wszystkie aktualnie zainstalowane urządzenia o wyjściach cyfrowych są wyświetlane w rozwijanym pasku. Po wybraniu jednego z nich wyświetlane są dostępne dla danego urządzenia kanały (bajty), a następnie dla danego kanału bity. Można również ustawić wartość początkową dla danego kanału (initial value), np, 0.



Częstotliwość próbkowania (Update rate field)

Pole to służy do ustawiania (zmniejszania) częstotliwości próbkowania. Np. jeśli potrzebna jest częstotliwość 20 Hz a aktualna dla danej aplikacji jest częstotliwość 100 Hz, wystarczy wpisać lub wybrać 5.

Połączenie DDE (Establish DDE link field)

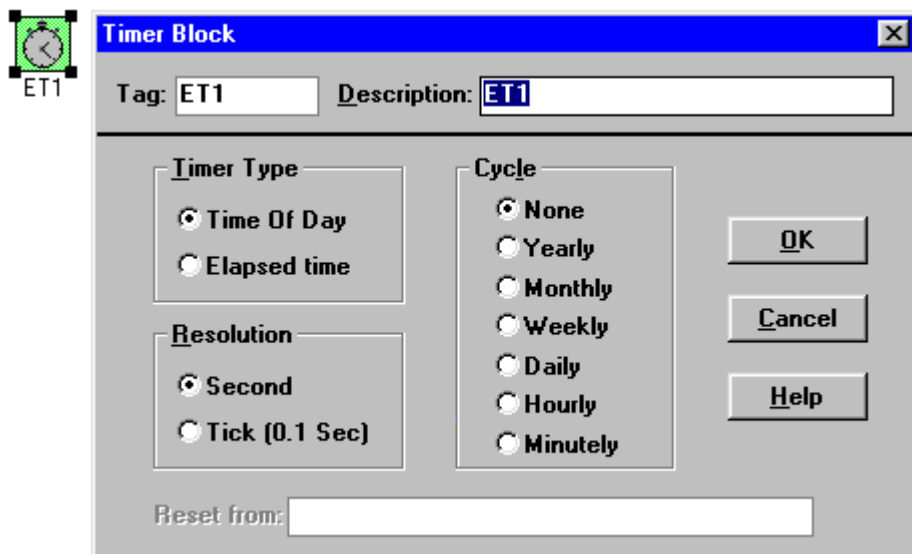
Blok DO jest również wyposażony w połączenie DDE (Dynamic Data Exchange). Służy ono do dostosowywania sygnału cyfrowego do komunikacji VisiDAQ z innymi aplikacjami Windows.

Wejścia. Blok ten bezpośrednio pobiera sygnał cyfrowy z innego bloku i przekazuje go do sekcji cyfrowej wyjścia wybranego urządzenia. Wejście jest tylko jedno. W przypadku próby podłączenia większej ilości bloków VisiDAQ wyświetli komunikat „The input already connected”

Wyjścia. Łączą bezpośrednio blok DO z wyjściem cyfrowym wybranego urządzenia.

Blok stopera (Timer block)

Blok ten posiada wejścia, jednak tylko do resetowania. Czas może być przez niego podawany jako aktualna godzina lub przyrostowo, natomiast zmiany mogą następować co sekundę lub co 0,1 sekundy. Zakres cyklu może być ustawiony od co 1 minuty do co 1 roku. Sygnał wyjściowy może być przekazywany do innego bloku.



Wyjście bloku stopera jest to ciąg cyfr od 0 do 4294967295. Blok ten jest bardzo przydatny w aplikacjach, w których czas jest zasadniczym elementem.

Jednostką wartości wyjściowej stopera jest sekunda lub 0,1 sekundy (tick), zależnie od wybranej opcji. Z powodu ograniczeń środowiska Windows trudno jest uzyskać lepsze rezultaty, niż te, które umożliwia VisiDAQ.

Wartość wyjściowa bloku stopera jest cykliczna tylko wtedy, gdy taka opcja zostanie wybrana. Np. wartość wyjściowa dla odliczania przyrostowego (Elapsed time), jednostki sekundy, oraz dla zakresu cyklu co 1 minuta będzie ona przyjmowała wartości od 0 do 59 i wróci do wartości zerowej. Ten sam blok z zakresem godzinnym odmierzy czas od 0 do 3599 i powróci do 0.

Blok stopera może być resetowany z pozycji innego bloku. Wartość 1 wysłana z niego do bloku stopera resetuje sygnał wyjściowy stopera.

Dla bloku stopera i zakresu cyklu dziennego wartość wyjściowa nie będzie zaczynała się od 0 wraz z uruchomieniem aplikacji. Będzie on podawał do granicy cyklu co 1 sekundę lub co 0,1 sekundy sygnał zgodnie z ustawieniem zegara. Granice dla poszczególnych ustawień stopera wynoszą:

- Yearly (corocznie) – rozpoczyna odliczanie od 0 o godzinie 00:00:00 pierwszego stycznia każdego roku,
- Monthly (comiesięcznie) – rozpoczyna odliczanie od 0 o godzinie 00:00:00 pierwszego dnia każdego miesiąca,
- Weekly (cotygodniowo) – rozpoczyna odliczanie od 0 o godzinie 00:00:00 w każdą niedzielę,
- Daily (codziennie) - rozpoczyna odliczanie od 0 o godzinie 00:00:00 każdego dnia,
- Hourly (co godzinę) - rozpoczyna odliczanie od 0 o każdej pełnej godzinie,
- Minutely (co minutę) - rozpoczyna odliczanie od 0 wraz z upłynięciem każdej minuty.

Wyliczenie aktualnego czasu z podawanego przez blok stopera sygnału nie przedstawia dużej trudności. Np. cotygodniowy stoper podaje wartość 86400 o godzinie 00:00 w poniedziałek, natomiast 518400 o godzinie 24:00 w piątek. Można użyć takiej wartości w bloku User Program Block, żeby zaprogramować wyłączenie jakiegoś urządzenia na weekend i powtórnego włączenia w dzień roboczy. Można także zastosować większą liczbę bloków stopera do sterowania bardziej skomplikowanymi czynnościami.

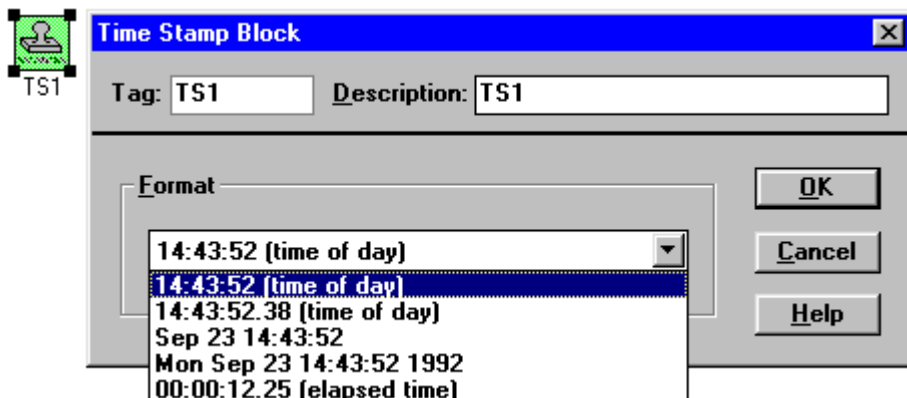
Proszę obejrzeć plik TIMER.GNI w katalogu VisiDAQ/Strategy. Są tam umieszczone przykłady, jak można budować i używać bloku stopera. Ta aplikacja generuje potrójny sygnał w 0-wej, 2-giej i 4-tej sekundzie każdej minuty przez pierwsze 5 minut każdej godziny od poniedziałku do piątku.

Wejścia. Blok posiada tylko wejście 0/1 służące do resetowania stopera (1=reset)

Wyjścia. Blok ten wyprowadza dane typu „long integer” czasu ubiegłego od uruchomienia aplikacji lub czasu absolutnego (sygnał pobrany z zegara komputera) do podłączonego bloku.

Datownik (Time Stamp Block)

Blok ten posiada tylko wyjście. Aktualny czas może być podawany na wyświetlacz lub do pliku typu „log file” przez połączenie go z wyjściem bloku datownika albo wyświetlaczem danych typu „numeric/string”. Innego typu dane są dopuszczalne, jednak wyjścia tego bloku są w postaci danych typu „string”, czyli ciągu znaków.



Do wyboru są różne opcje podawania czasu:

- 12:34:56 - aktualna godzina,
- 12:34:56.78 - aktualna godzina, jednak z setnymi częściami sekundy,
- Jun 22 12:34:56 – aktualna data i godzina,
- Fri Jun 22 12:34:56 2001 – dzień tygodnia, roku, aktualna godzina i rok,
- 0:01:06.64 – czas, jaki upłynął od uruchomienia aplikacji (z setnymi częściami sekundy)

Wejścia. Blok nie posiada wejść. W razie próby podłączenia sygnału wejściowego VisiDAQ wyświetli komunikat „Cannot accept input”.

Wyjścia. Blok podaje aktualny czas do podłączonego bloku.

Blok licznika (Event Counter Block)

Blok ten posiada wejścia i wyjścia cyfrowe. Oprogramowanie licznika zlicza każdą cyfrową, wzrastającą wartość (najwyższy cyfrowy odczyt) pobieraną z innego bloku z cyfrowym wyjściem. Wartość wyjściowa bloku licznika może być podana do innego bloku. Zliczanie jest wykonywane przy każdym skanowaniu, dlatego jego prędkość jest równa prędkości próbkowania systemu. Przez podłączenie wysokiej cyfrowej wartości do wejścia resetującego „reset from” bloku można sterować nim tak, że po podaniu sygnału licznik jest resetowany, a zliczanie zatrzymane. Niska wartość cyfrowego sygnału umożliwia zliczanie. Jeżeli do tego wejścia nie jest podłączony żaden sygnał, jest to traktowane jako podanie jego niskiej wartości. Przy zastosowaniu wysokiej wartości sygnału cyfrowego na wejściu zatrzymującym „hold from” zliczanie jest wstrzymywane, natomiast podanie niskiej wartości powoduje wznowienie zliczania od zatrzymanej wartości.

1234
CNT1

Event Counter Block

Tag: Description:

Start Count at: OK

Stop Count at: Cancel

Increment/Decrement by: Help

Automatically reset to start count upon reaching stop count.

Input from:

Reset from:

Hold from:

Wartość początkowa (Start value).

Liczba całkowita „integer”, od której licznik zaczyna zliczać impulsy. Jej maksymalna wartość może wynosić 65535.

Wartość końcowa (Stop value).

Liczba całkowita „integer”, na której zliczanie ma się zakończyć. Jej wartość może być wyższa lub niższa od wartości początkowej, jednak nie większa, niż 65535.

Jednostka zliczania (Increment/Decrement by).

Można tu ustawić, co ile impulsów będą one zliczane (standardowo co 1 impuls), niezależnie, czy odbywa się ono przyrostowo, czy malejąco.

Pole sygnału wejściowego (Input field).

Jeżeli na stronie zadań z blokiem licznika za pomocą „nici” zostanie połączony blok dostarczający sygnał, w tym polu zostanie wyświetlona jego nazwa.

Pole wejścia resetującego (Reset field).

Wysoki lub niski sygnał podawany do tej sekcji bloku powoduje analogicznie resetowanie licznika i zatrzymanie zliczania lub jego ponowne uruchomienie od wartości początkowej.

Nie doprowadzenie do tego wejścia żadnego sygnału jest traktowane przez system jako niska wartość sygnału i aplikacja pracuje normalnie.

Pole pauzy (Hold field)

Wysoki lub niski sygnał podawany do tej sekcji bloku powoduje analogicznie zatrzymanie zliczania impulsów lub ponowne uruchomienie od zliczonej wartości w momencie zatrzymania.

Nie doprowadzenie sygnału do tego wejścia jest traktowane przez system jako podanie niskiej wartości sygnału.

Wejścia. Blok licznika analizuje dane wejściowe jako zmieniające się wzrostowo wartości sygnału, posiada również wejścia cyfrowe (reagujące na sygnał 0/1) resetujące i wstrzymujące.

Wyjścia. Blok licznika podaje aktualna wartość zliczenia do, podłączonego bloku.

Blok licznika, częstotliwości pomiarów i generator impulsów (Hardware Event Counter / Frequency Measurement / Pulse Output Block)

Blok ten posiada wejścia i wyjścia. Podaje on do podłączonego bloku wynik zliczania, ustawioną częstotliwość pomiarów lub wygenerowane impulsy z sekcji licznika/stopera urządzenia.

Pole urządzeń/kanałów (Device/Channel field)

Po dwukrotnym kliknięciu tego pola wszystkie aktualnie zainstalowane urządzenia posiadające dostępne kanały licznika/stopera zostaną wyświetlone we wspólnym oknie dialogowym. Po wybraniu jednego z nich wszystkie dostępne kanały licznika/stopera zostaną wyświetlone poniżej. Wybiera się kanał sterownika, który chcemy podłączyć do bloku licznika, częstotliwości pomiarów lub generatora impulsów przez kliknięcie na nim.

Opcje licznika/częstotliwości pomiarów (Counter Input/Frequency Measurement option)

[Hardware'owa] kość licznika/stopera sterownika jest tu używana jako [hardware'owy] licznik zliczający cyfrowe wzrastające lub najwyższe wartości z zewnętrznego źródła dostarczającego cyfrowy sygnał do tego bloku (TTL lub inny poziomu zerojedynkowego).

Blok ten może być używany do kontroli wyjścia kości licznika/stopera (jeżeli jest załączony do sterownika), żeby być użytym do generowania impulsów. Sygnał wyjściowy z tego bloku może być wysyłany do innego bloku, np. wyświetlającego dane.

Wszystkie wartości wyjściowe tego bloku, tzn. licznika i generatora impulsów są generowane przez kartę zerojedynkową kości licznika/stopera i nie są one zależne od częstotliwości próbkowania charakterystycznej dla aplikacji. Blok ten jest zaprogramowany jako licznik odliczający od ustawionej wartości minimalnej do ustawionej wartości maksymalnej. Przy połączeniu go, jednak z blokiem programowym (User Program Block) lub z blokiem kalkulacji (Single Operation Calculation) blok może odejmować wynik zliczania od wartości maksymalnej i w ten sposób otrzymywać odliczanie wsteczne.

Jeżeli wejście start/stop jest połączone z blokiem posiadającym wyjście zerojedynkowe licznik oraz generator impulsów może być uruchamiany i zatrzymywany za jego pomocą podczas działania aplikacji. Jeżeli do tego wejścia nie będzie podłączony żaden sygnał zliczanie będzie się rozpoczynało w momencie uruchomienia aplikacji, natomiast zatrzymywało w chwili zatrzymania działania aplikacji. Jeżeli zostanie do tego wejścia podłączony opadający sygnał (od wartości maksymalnej do minimalnej) również może nastąpić zatrzymanie zliczania na aktualnej wartości.

Przy doprowadzeniu rosnącego (od wartości minimalnej do maksymalnej) sygnału do wejścia resetującego wynik zliczania zostanie wyzerowany, a samo zliczanie zostanie rozpoczęte od początku. Opadająca

wartość sygnału doprowadzona do wejścia resetującego nie wywołuje żadnych reakcji aplikacji. Brak doprowadzenia sygnału, natomiast jest traktowane przez system jako niska wartość sygnału.

Wyjście generatora impulsów (Pulse Output Option)

Przez ten blok można generować impulsy z zerojedynkowej sekcji licznika/stopera sterownika. Aby to zrobić, należy najpierw określić całkowity okres (Total Period) oraz czas pierwszej 1/2 cyklu (w sekundach), lub podłączyć inny blok do odpowiednich wejść o danych typu „floating point”.

Bramka zewnętrzna (Gate Mode)

Jeżeli dany sterownik posiada możliwość podłączenia zewnętrznej bramki, za pomocą której można doprowadzić inną, niż generowana w sterowniku wartość sterującą, można ją tu skonfigurować.

Częstotliwość próbkowania (Update Rate Field)

Pole to pozwala na uzyskanie innej, niż standardowa dla aplikacji częstotliwości próbkowania. Np. jeżeli standardowo ustawiona jest częstotliwość 20 Hz, natomiast potrzebna jest częstotliwość 4 Hz, wystarczy ustawić wartość w tym polu na 5, a aktualna wartość zostanie podzielona przez ustawioną. Jeżeli dane wyjściowe wyśle się do bloku zapisu (Log File) lub bloku wyświetlającego, dalej będzie to częstotliwość 20 Hz, jednak tylko jedna z pięciu próbek będzie „prawdziwa”; reszta będzie jedynie „kopia” poprzednich. Wartości wpisywane w polu częstotliwości próbkowania muszą się mieścić w granicach od 1 do 32767. Uwaga. Dla różnych liczników/stoperów dostarczających sygnał do tego bloku muszą być tworzone oddzielne bloki ze względu na ich różne specyfikacje hardware'owe.

Wejścia. Blok posiada cztery wejścia:

- wejście uruchamiająco-zatrzymujące (Start/Stop Input),
- wejście zerujące (Reset input),
- wejście określenia całkowitego impulsu (Total Period Input),
- wejście określenia pierwszej 1/2 cyklu (First 1/2 Cycle Input).

Wyjścia. Blok wysyła wynik zliczania, częstotliwość pomiarów lub impulsy, czyli wygenerowane lub przetworzone dane do podłączonego bloku.

Blok regulatora PID (PID Control Block)

Blok ten posiada zarówno wejścia, jak i wyjścia. Wejście składa się ze sprzężonych sygnałów wartości mierzonej i wartości żądanej (setpoint) ustawianej liczbowo (statycznie) lub doprowadzanej jako sygnał z innego bloku (dynamicznie). Wyjście bloku jest sygnałem kontrolującym proces.

Blok umożliwia pobieranie sygnału z czujnika (zwykle temperatury) podczas procesu i regulowanie wartości wejściowej tego procesu zgodnie z ustalonymi lub doprowadzonymi do niego danymi sterującymi.

Blok kontroli PID jest chętnie stosowanym do nadzoru procesów blokiem ze względu na możliwość regulacji kontroli procesu dla specyficznych w danym momencie żądań w stosunku do stanu procesu. Właśnie te cechy są powodem częstego stosowania tego bloku do kontroli procesów o przypadkowo, ale nieznacznie zmieniających się parametrach. Kontrola procesu polega na pobraniu zmierzonej wartości, porównaniu jej z wprowadzoną liczbowo lub doprowadzaną sygnałowo z innego bloku wartością żadaną, obliczeniu różnicy między nimi (błędu) oraz wyeliminowaniu tego błędu poprzez podanie na wejście procesu sygnału zwrotnego o określonych wcześniej właściwościach korekcyjnych. Jest to bardzo sprawny sposób kontroli procesu.

Jest możliwe kontrolowanie zmieniającej się „pozycji” czujnika lub kontrola prędkości.

Funkcje kontroli bloku PID mogą być różne w zależności od potrzeby aplikacji. Do dyspozycji użytkownika są trzy tryby kontroli procesu.

P – proporcjonalny.

Jest to najprostszy z tych trybów. Podczas jego działania na wejście procesu wywiera wpływ sprzężona z wartością żadaną wartość sygnału o odwrotnej względem niej wartości i proporcjonalnej do wpisanej w pole „P Value” liczby (również ujemnej i z przedziału $0 < P < 1$). Standardowo jest ustawiona wartość $P=1$. Powoduje to podawanie sygnału zwrotnego o wartości dokładnie symetrycznej do wartości żądanej. Nastawienie wartości $P > 1$ powoduje symetryczny względem wartości żądanej ale również zwielokrotniony sygnał, natomiast dla $P < 1$ sygnał jest odpowiednio proporcjonalnie pomniejszony. Jeżeli podamy jako wartość P liczbę ujemną uzyskamy sygnał pokrywający się z wprowadzanym, proporcjonalnie powiększony lub pomniejszony. Dodatkowo, tak, jak wartość kontrolowana, czy wartość żądana procesu można podawać wartość P doprowadzając ją jako sygnał z innego bloku.

P/I – proporcjonalno – całkujący.

Ta opcja pozwala na dodatkowy sposób kontroli procesu zależący od nagromadzonego w czasie błędu (całka). Suma błędów i ciągle kontrolujący proces sygnał tak wpływają na jego przebieg, że błąd zaczyna

się zmniejszać, a sumowanie błędów kończy się wraz z pokryciem się skorygowanej wartości sygnału z wartością żadaną. Ta opcja działa na zasadzie zdążania pochodnej wyniku do zera.

P/I/D – proporcjonalno – całkująco – różniczkujący.

Ta opcja bloku posiada dodatkową funkcję (różniczkującą), która redukuje tendencje do rozmijania się sygnału wejściowego z wartością żadaną. Jest to uzyskiwane przez przewidywanie zbliżenia się wartości błędu do zera i inicjuje sygnał zwrotny zanim wartość ta (zwykle mierzona temperatura) osiągnie je. Właściwe użycie tej funkcji może się przyczynić do znacznego przyspieszenia realizacji sprowadzania aktualnej wartości kontrolowanego procesu do wartości żadanej.

P – wartość parametru trybu proporcjonalnego kontroli procesu,

I – wartość parametru trybu całkującego kontroli procesu,

D – wartość parametru trybu różniczkującego kontroli procesu.

Dolna kłamra (Low Clamp) – ograniczenie sygnału wyjściowego od dołu. Zapobiega przekroczeniu przez wyprowadzany do systemu sygnał wartości przez niego dopuszczalnych (zwykle przetwornika analogowo – cyfrowego).

Górna kłamra (High Clamp) – ograniczenie sygnału wyjściowego od góry. Zapobiega przekroczeniu przez wyprowadzany do systemu sygnał wartości przez niego dopuszczalnych.

Wartość żadana (Setpoint) – nastawna wartość, względem której jest kontrolowany proces. Jest ona doprowadzana z innego bloku w postaci wartości, do której ma dążyć wartość kontrolowana procesu (dynamic setpoint). Jeśli wartość ta ma być stała, należy ją wpisać liczbowo w stosowne pole.

Wejścia. Blok kontroli PID przyjmuje sygnał jako sprzężenie (Feedback), wartość żadaną (Setpoint), oraz jako parametry: proporcjonalny (P), całkujący (I) i różniczkujący (D) a następnie używa ich do konfiguracji kontroli procesu.

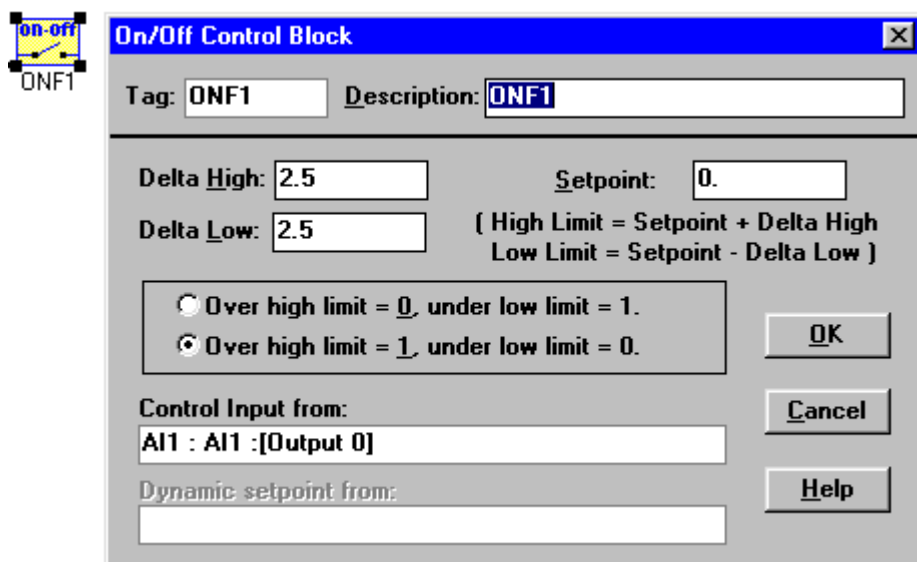
Wyjścia. Blok wyprowadza gotowy sygnał, który może być podawany do innego bloku, który z kolei poda go do wejścia aktualnie kontrolowanego procesu.

Blok włączająco – wyłączający (On/Off Control Block)

Blok ten posiada wejścia i wyjścia. Wejście składa się z mierzonej wartości (sprzężenia) kontrolowanej przez określony przez użytkownika zakres tolerancji (wpisany jako wartość stała lub doprowadzany jako sygnał z innego bloku. Wyjściem jest wysokiej lub niskiej wartości sygnał (zerojedynekowy), zależnie od ustawienia.

Jest to blok nieproporcjonalny, przyjmujący wartości wyjściowe tylko wysokie lub tylko niskie, zależnie od tego, czy mierzona wartość znajduje się poniżej, czy powyżej zadanych wartości.

Wartość odniesienia (Setpoint) – wartość, względem której są odnoszone dopuszczalne, również nastawne wartości przełączające, przy przekroczeniu których następuje przełączenie wyjścia w stan wysoki lub niski, w zależności od ustawienia. Wartość ta może być nastawiana za pomocą doprowadzanego sygnału (dynamic setpoint). Jeżeli wartość ta ma być stała, należy ją wpisać w pole „Setpoint”.



Górna granica błędu (Delta High) – nastawna wartość będąca składową sygnału sterującego wyjściem bloku. Sygnał ten jest równy sumie wartości odniesienia i wartości górnej granicy błędu.

Dolna granica błędu (Delta Low) – nastawna wartość będąca składową sygnału sterującego wyjściem bloku. Sygnał ten jest równy różnicy wartości odniesienia i wartości dolnej granicy błędu.

Dodatkowo jest do dyspozycji, przy przekroczeniu którego z limitów ma nastąpić przełączenie wyjścia bloku. Są dwie możliwości: opcja, przy której przełączenie na „0” następuje po przekroczeniu górnego limitu, natomiast na „1” po przekroczeniu dolnego, oraz opcja, przy której przełączenie na „1” następuje po przekroczeniu górnego limitu, natomiast na „0” po przekroczeniu dolnego.

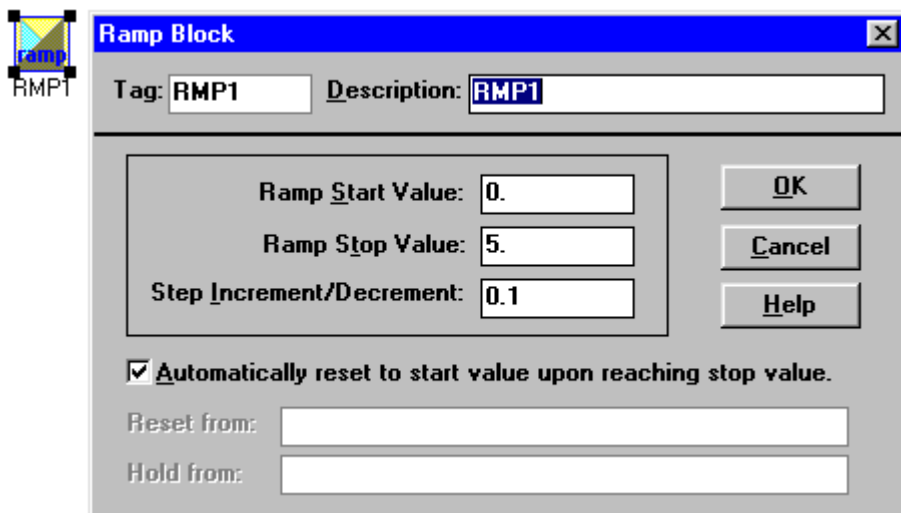
Jeśli przedział błędu nie jest potrzebny można nastawić górną i dolną granicę błędu na zero. W takim wypadku blok ten może być używany jako blok alarmu o przekroczeniu na stałe nastawionej wartości (statycznej).

Wejścia. Blok ten przyjmuje dane wejściowe do analizy oraz opcjonalnie dynamicznie doprowadzany sygnał sterujący wartością odniesienia.

Wyjścia. Blok wyprowadza wyłącznie cyfrowy wysoki lub niski sygnał, zależny od przekroczenia przez sygnał wejściowy nastawionej wartości dopuszczalnego błędu.

Blok [„rampujący”]

Blok ten posiada wejścia i wyjścia. Za jego pomocą można generować [„rampę”] wartości typu „floating point”. Przy doprowadzeniu do wejścia resetującego wysokiej wartości sygnału cyfrowego [„rampowanie”] zostanie zatrzymane, a sygnał wyjściowy powróci do wartości początkowej. Natomiast przy doprowadzeniu do wejścia pauzującego wysokiej wartości sygnału cyfrowego nastąpi chwilowe wstrzymanie [„rampowania”] na aktualnej wartości. Tempo wzrostu/spadku [„rampowania”] jest zależne od ustawionej częstotliwości próbkowania (okresu skanowania).



Wartość początkowa (Start Value) – wartość typu „floating point”, od której ma się rozpocząć [„rampowanie”].

Wartość końcowa (Stop Value) – wartość typu „floating point”, na której [„rampowanie”] ma się zakończyć. Może ona być wyższa lub niższa od wartości początkowej.

Wzrost/spadek (Increment/Decrement) – wartość, o jaką wzrastać / spadać będzie wartość wyjściowa w każdym kroku [„rampowania”].

Wejście resetujące (Reset from Block) – pozwala na podłączenie dowolnego bloku cyfrowego wyprowadzającego wysoki lub niski sygnał sterujący [„rampowaniem”]. Jego wysoka wartość resetuje oraz wstrzymuje działanie bloku, natomiast niska powoduje uruchomienie go od wartości początkowej. Brak doprowadzenia sygnału resetującego do tego wejścia jest traktowane przez system jako doprowadzenie do niego cyfrowego sygnału o niskiej wartości.

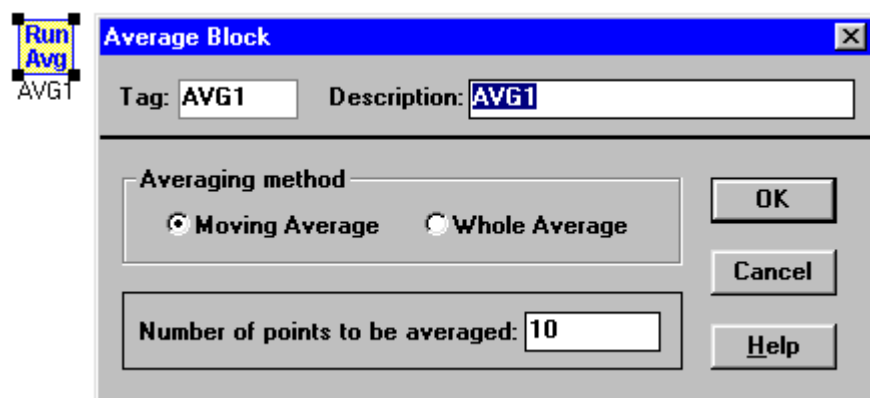
Wejście pauzujące (Hold from Block) – pozwala na podłączenie dowolnego bloku cyfrowego wyprowadzającego wysoki lub niski sygnał sterujący [„rampowaniem”]. Jego wysoka wartość wstrzymuje działanie bloku w jego aktualnym stanie, natomiast niska wznowia je od wartości, w której zostało ono wstrzymane. Brak doprowadzenia sygnału wstrzymującego jest traktowany przez system jako doprowadzenie do niego cyfrowego sygnału o niskiej wartości.

Wejścia. Blok [„rampujący”] posiada tylko dwa wejścia: resetujące i pauzujące przyjmujące dane z bloków cyfrowych i wykorzystuje je do sterowania wartością wyjściową.

Wyjścia. Blok wyprowadza odpowiednio skonfigurowany wygenerowany sygnał analogowy do innego bloku.

Blok uśredniający (Average Block)

Blok ten posiada wejścia i wyjście. Użytkownik ma do dyspozycji dwie metody uśredniania doprowadzanego sygnału.



Opcja pierwsza (Moving Average) to obliczanie średniej z ustalonej liczby ostatnio pobranych próbek, np., jeśli wybranych zostanie 10 próbek, program przy każdym następnym kroku będzie obliczał i wyprowadzał sygnał równy średniej z aktualnego i poprzedzających go 9-ciu próbek. Druga opcja (Whole Average) pozwala na obliczanie, a następnie wyprowadzanie sygnału równego średniej ze wszystkich próbek wykonanych od momentu uruchomienia aplikacji. Po dłuższym czasie, nawet przy dość zróżnicowanych wartościach doprowadzanego sygnału otrzymujemy prawie stały średni sygnał wyjściowy. Te właściwości bloku mogą mieć zastosowanie do wyciszania szumów i sygnałów ubocznych. Blok ten (z ustawieniem na określoną liczbę próbek) powinien być umieszczony pomiędzy blokiem próbkującym, a następnym blokiem logicznym, np. blokiem wyświetlającym.

Wejścia. Blok uśredniający przyjmuje sygnał jako zbiór próbek, a następnie dokonuje obliczeń.

Wyjścia. Blok wyprowadza sygnał będący wynikiem obliczeń na pobranych próbkach do innego bloku.

Blok pobierania danych z pliku (Data File Block)

Blok ten posiada wyłącznie wyjście. Za jego pomocą można odzyskiwać zapisane wcześniej dane z pliku. Dane są odzyskiwane po jednej linii we właściwym dla systemu tempie próbkowania. Są to dane liczbowe mimo, że system żąda pliku, z którego ma odczytywać dane, z rozszerzeniem „.txt”. Dane te muszą być w pliku typu ASCII uszeregowane w jednej kolumnie jako dane typu „integer” lub „floating point”. W momencie, gdy cały zasób danych z pliku zostanie wyczerpany, automatycznie jest on sczytywany od początku (pętla). Po dwukrotnym kliknięciu na ikonie bloku w wyświetlonym oknie dialogowym należy wpisać lub wyszukać w zawartości dysku nazwę pliku, z którego dane mają być pobierane.



Wejścia. Blok nie posiada wejść. W przypadku próby jego podłączenia VisiDAQ wyświetli komunikat „Cannot accept input”.

Wyjścia. Blok ten przekazuje dane pobrane z wybranego na dysku pliku do podłączonego bloku.

Blok zapisu danych (Log File Block)

Blok ten posiada tylko wejścia (maksymalnie 8). Przyjmuje on dane z kanałów wejściowych i loguje je w formacie wielokolumnowym. Każdemu blokowi przyporządkowana jest jedna kolumna, w której są szeregowane dane z niego pochodzące.

Do wyboru są rodzaje formatów zapisu danych. Logowanie danych może odbywać się za pomocą: kodu ASCII (American Standard Code for Information Interchange), systemu dwójkowego (binary float), bajtów (byte), liczby całkowitej (integer) lub ciągu znaków (long integer).

Dane mogą być zapisywane na dwa sposoby. Mogą one być dopisywane do uprzednio zapisanych lub w chwili wprowadzenia pierwszej danej kasowane są wszystkie wcześniej zapisane odnoszące się do tego wejścia.

Wybrać można również sposób separacji danych w poszczególnych kolumnach. Może to być odstęp (space), przecinek (comma) lub tabulator (tab).

Na początku każdej kolumny można umieścić nagłówek z komentarzem, głównie w celu identyfikacji, która z kolumn odnosi się do którego wejścia. Dodatkowo można określić szerokość kolumny (w ilości znaków) (Column Width) oraz dokładność (ilość zer po przecinku) (Format), z jaką będą sczytywane dane. Każdemu wejściu są przyporządkowywane numery kolumn w kolejności ich podłączenia. Jeżeli należy tę kolejność zmienić, wystarczy wszystkie, wyświetlone, a zarazem podłączone bloki, których kolumn umiejscowienie zmieniamy, dwukrotnie kliknąć. Obok każdego z nich ukaze się wtedy zamiast numeru kolumny myślnik (--). Należy wtedy w żądanej kolejności ponownie dwukrotnie kliknąć na każdym z nich.

Wejście włączająco – wyłączające (Log On/Off input)

Służy do kontrolowania zapisu danych. Doprowadzenie do niego niezerowego sygnału pozwala na logowanie danych bez ograniczeń, natomiast sygnał o wartości zerowej powoduje zawieszenie ich sczytywania. Z pomocą tej funkcji bloku można selekcjonować ręcznie dane, które chcemy, aby zostały zapisane, lub do zapisu danych np. z co n-tego pomiaru.

Wejście otwierająco – zamykające (Open/Close Input)

Służy do kontroli aktywności zapisu. Ta opcja współpracuje tylko z plikiem o nazwie typu „wildcard”. Od momentu otwarcia pliku blok ten będzie wywoływany, aby użyć innego pliku typu „wildcard”.

Zaawansowane funkcje tej opcji są dostępne pod klawiszem „Advanced”. Jest tam dostępnych łącznie 16 kombinacji otwierania i zamykania pliku (4 dla otwierania i 4 dla jego zamykania).

Opcje otwierające:

- A. Otwieranie wraz z uruchomieniem aplikacji (Open at the beginning of the run),
- B. Otwieranie po określonej liczbie próbek (Open after N scans),
- C. Otwieranie w przypadku, gdy sygnał kontrolujący doprowadzany z innego bloku wynosi „1” (Open if "File Control" is 1). Plik jest automatycznie zamykany, gdy sygnał ten przyjmie wartość „0”.
- D. Otwieranie po określonej liczbie minut od północy (Open at N minutes from midnight). Funkcja przydatna, gdy potrzebny jest zapis danych codziennie o tej samej porze. Następne pliki są obsługiwane przez opcje zamykania pliku.

Opcje zamykające:

- A. Zamykanie bezpośrednio po zatrzymaniu aplikacji (Close at the end of the run),
- B. Zamykanie po pobraniu określonej liczby próbek (Close after N records),
- C. Zamykanie w przypadku, gdy sygnał kontrolujący doprowadzany z innego bloku wynosi „0” (Close if "File Control" is 0). Następny plik jest automatycznie otwierany, gdy sygnał sterujący przyjmie znów wartość „1”.
- D. Zamykanie po określonej liczbie minut od momentu otwarcia pierwszego pliku (Close every N minutes). Następny plik jest automatycznie otwierany w chwili zamknięcia poprzedniego, jeżeli sobie tego życzymy.

Format nazw plików typu „wildcard” pozwala systemowi na automatyczne tworzenie plików o nazwach z kolejnymi ich numerami. Jest to realizowane przez niektóre z opcji otwierająco – zamykających pliki do zapisu danych. Tylko ten format pliku pozwala na użycie w nazwie znaku krzyżyka (#). Znak ten może się pojawiać na dowolnej pozycji standardowej nazwy pliku w systemie DOS w miejscu o charakterze regularnym. Ma on tu zastosowanie jako znak zastępujący cyfrę (0 – 9) w celu stworzenia standardowej nazwy pliku. Aby utworzyć więcej, niż 10 plików należy użyć więcej znaków # obok siebie. Plik typu „wildcard” w formie „FILE###.LOG” może wygenerować 1000 plików o nazwach począwszy od „FILE000.LOG” po „FILE999.LOG”. Przed utworzeniem takiego pliku należy się upewnić, czy ilość możliwych do wygenerowania nazw plików jest wystarczająca dla układanej przez nas aplikacji. Należy również pamiętać, aby wszystkie znaki znajdowały się obok siebie.

Zachowywanie i odczytywanie zawartości plików już zamkniętych jest możliwe tylko po zatrzymaniu aplikacji. Po jej ponownym uruchomieniu zostanie utworzony plik o nazwie kolejnej względem ostatnio utworzonego. Ta funkcja jest możliwa tylko jeżeli użyto nazwy pliku typu „wildcard”. Należy zaznaczyć, że plik zostanie zapisany tylko, jeśli sesja aplikacji zostanie zamknięta prawidłowo. Użycie kombinacji klawiszy Ctrl+Alt+Del w celu zresetowania komputera lub wyłączenie jego zasilania bez uprzedniego zatrzymania sesji spowoduje bezpowrotną utratę danych zapisanych w ostatnio używanym, a nie zamkniętym pliku.

Przykłady prawidłowych nazw plików typu „wildcard”:

```
MYFILE##.LOG
MY###R.LOG
MYFILE.###
MYFILE.L##
#####.TXT
#RECORD.TXT
```

Przykłady nieprawidłowych nazw plików typu „wildcard”:

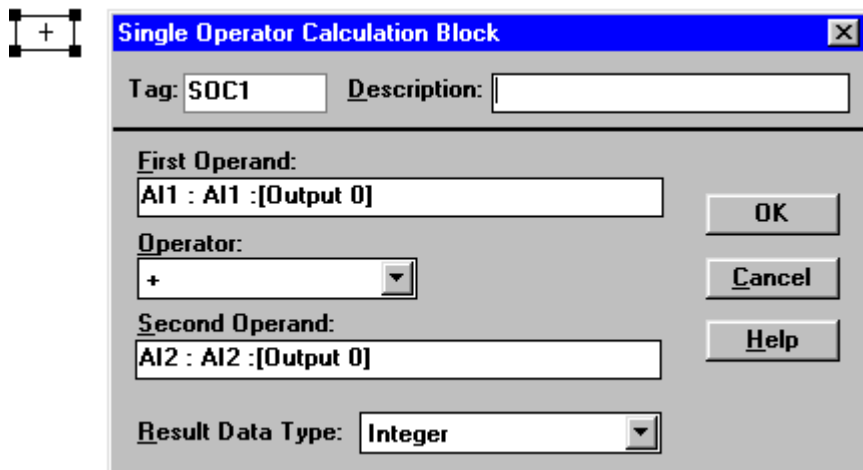
```
MY#FILE#.LOG – znaki # nie mogą być rozdzielone innymi znakami,
FILE##.##T – znaki # nie mogą być rozdzielone kropką,
MYFILE###.TXT – nazwa pliku jest za długa, musi być w formacie 8.3 znaków.
```

Wejścia. Blok zapisu danych posiada wejścia w postaci danych do zapisu, sygnału kontrolującego zapis w pliku oraz sygnału kontrolującego otwieranie i zamykanie tworzonych plików. Wszystkie podłączone wejścia są wyświetlane w oknie dialogowym.

Wyjścia. Nie ma dostępnych wyjść.

Blok kalkulacyjny (Single Operator Calculation Block (SOC))

Blok ten wykonuje pojedyncze operacje matematyczne, takie, jak dodawanie, odejmowanie, mnożenie itp. Co najmniej jeden blok musi być podłączony do wejścia jako pierwszy operator. Drugim operatorem może być inny blok lub stała wprowadzona cyfrowo do okna dialogowego.



Zamiana uprzednio wybranych operatorów miejscami może być dokonana za pomocą klawisza „Swap Operands”. Jest to szczególnie ważne, gdy nie jest obojętne, który z nich jest pierwszy, np. dzielenie, odejmowanie, itd. Pierwszy blok podłączony do bloku kalkulacyjnego automatycznie jest traktowany, jako pierwszy operator. Jeśli kolejność ta ma być zmieniona, wystarczy użyć klawisza „Swap Operands”.

Wartość wyjściowa bloku może mieć postać liczby rzeczywistej (Floating Point (Real)) lub całkowitej (Integer).

Działania możliwe do wykonania w bloku:

Działanie	Wynik
nop	sygnał wyjściowy równy 0
+	OP1 + OP2
-	OP1 - OP2
x	OP1 x OP2
/	OP1 / OP2
pow	OP1 do potęgi OP2
*mod	reszta z dzielenia OP1 przez OP2
*and	funkcja logiczna „i” dla OP1 i OP2
*or	funkcja logiczna „lub” dla OP1 i OP2
*xor	funkcja logiczna „x lub” dla OP1 i OP2
max	podawany sygnał z OP1 lub OP2 o wyższej wartości
min	podawany sygnał z OP1 lub OP2 o niższej wartości
>=	sygnał wyjściowy równy „1” jeśli OP1 >= OP2, inaczej sygnał równy „0”
<=	sygnał wyjściowy równy „1” jeśli OP1 <= OP2, inaczej sygnał równy „0”
>	sygnał wyjściowy równy „1” jeśli OP1 > OP2, inaczej sygnał równy „0”
<	sygnał wyjściowy równy „1” jeśli OP1 < OP2, inaczej sygnał równy „0”
equ	sygnał wyjściowy równy „1” jeśli OP1 = OP2, inaczej sygnał równy „0”
neq	sygnał wyjściowy równy „1” jeśli OP1 nie jest równy OP2, inaczej sygnał równy „0”
abs	wartość bezwzględna sygnału operatora OP1
*not	funkcja logiczna „nie” odnosząca się do OP1
inv	odwrotność wartości sygnału z OP1 (1 / OP1)
sqrt	pierwiastek kwadratowy wartości sygnału z OP1
log	logarytm dziesiętny wartości sygnału z OP1
ln	logarytm naturalny wartości sygnału z OP1
exp	eksponenta (e do potęgi OP1)
jct	blok węzłowy (czytaj poniżej)

Działania oznaczone gwiazdką (*) mogą być wykonywane tylko na operatorach wysyłających dane typu „integer” (liczba całkowita). Niektóre z wyżej wymienionych działań potrzebują tylko jednego operatora, inne dwóch. Działania logiczne (and, or, xor) wymagają dwóch operatorów o danych w postaci liczb całkowitych. Inne (abs, not, inv, sqrt, log, ln, exp, jct) potrzebują jednego operatora o formacie danych w postaci liczby całkowitej lub rzeczywistej (floating point).

Należy tu również zwrócić uwagę na wartość sygnału dostarczanego do wejść kalkulacyjnych. Np. jako operator OP2 do działania „/” lub „mod” nie może być podłączony sygnał przyjmujący wartości zerowe (nie można dzielić przez zero). Z tego samego powodu wartość sygnału operatora OP1 przy działaniach „sqrt”, „log”, „ln” musi być dodatnia. System będzie wykazywał błędy, jeżeli te zasady nie będą zachowane.

Działanie „jct” ma specjalną funkcję łatwego wyprowadzania sygnału wejściowego. Jest to użyteczne podczas łączenia klawisza wyświetlającego z większą ilością innych bloków. W opcjach tego wyświetlacza można podłączyć tylko jeden blok. Użycie funkcji „jct” jako rozgałęzienie pozwala na podłączenie większej ich ilości. Można to zrobić w następujący sposób. Skonfigurować klawisz wyświetlający, aby sterował wejściem bloku kalkulacyjnego, a blok wyświetlania danych połączyć z blokiem kalkulacyjnym „nicia”. Potem skonfigurować blok kalkulacyjny na funkcję „jct”. Teraz można podłączyć dowolną ilość bloków do wejścia tego bloku i w ten sposób uzyskać kontrolę nad wszystkimi tymi blokami za pomocą jednego klawisza.

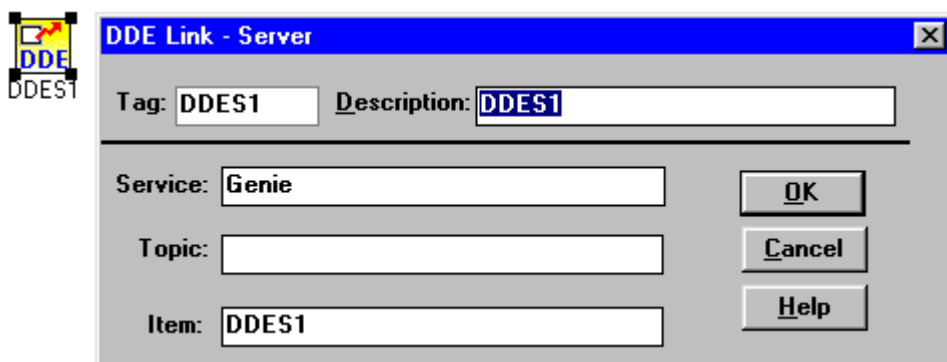
Wejścia. Blok przyjmuje dane jako operatory, na których wykonuje działania.

Wyjścia. Blok wyprowadza przeliczone dane do podłączonego bloku.

Serwer DDE (Dynamic Data Exchange Server)

Blok ten dostarcza dane z aplikacji VisiDAQ do innych aplikacji Windows. DDE wykorzystuje komunikację typu transmisyjnego i wyprowadza uniwersalny sygnał. Inne aplikacje są odpowiedzialne za odnalezienie go i użycie wg upodobania. Jeżeli inny blok jest podłączony do serwera DDE, jego „link” jest już ustalony. Po dwukrotnym kliknięciu na serwer pokazuje się okno dialogowe i informacja o nazwie serwisu (Service), temacie (Topic) i nazwie punktu emisji (Item). Ta informacja jest potrzebna innej aplikacji, aby mogła zlokalizować „link” stworzony w VisiDAQ.

Nazwą serwisu w naszym przypadku jest Nazwa aplikacji w VisiDAQ, czyli Genie. Jest to dla systemu Windows nazwa programu, z którego są dostarczane dane do innej aplikacji.



Tematem jest nazwa pliku, z którego są dostarczane dane. W VisiDAQ jest to nazwa aplikacji, w której wykorzystano ten blok (o rozszerzeniu *.GNI). Rozszerzenie to nie jest tutaj potrzebne, podaje się samą nazwę aplikacji. Np. jeśli nazwa aplikacji brzmi DDEDEMO.GNI, wyświetlona zostanie nazwa DDEDEMO. Należy zwrócić na to uwagę, ponieważ aplikacja bez tematu nie będzie zdolna do przesyłania danych przez połączenie DDE. Musi ona mieć nieużywaną w systemie nazwę i być zachowana w pamięci.

Punkt emisji (Item) jest to nazwa etykiety, pod jaką występuje dany serwer DDE, z którego są wyprowadzane dane (np. DDES1, DDES2, itd.). Inne prawidłowe nazwy etykiet to np. AI1, DO3 (pierwszy analogowy blok wejściowy, trzeci cyfrowy blok wyjściowy), itd. Te bloki o charakterze wejściowo –

wyjściowym mają wbudowaną funkcję DDE w ich oknach dialogowych. Znaczy to tyle, że etykieta jest tylko nazwą nadaną przez system VisiDAQ poszczególnym blokom którą można dowolnie zmieniać. Te trzy pola są swoistym adresem, pod którym inne aplikacje będą szukały danych. Zwykle większość innych aplikacji separuje nazwy z tych pól w następujący sposób:

Service|Topic!Item

Nie jest to jednak standardowy format adresu a jedynie zwykle praktykowany. Należy skorzystać z dokumentacji poszczególnych aplikacji, do których chcemy wysyłać sygnał za pomocą DDE, aby dowiedzieć się, jak format DDE jest wprowadzany do tego programu.

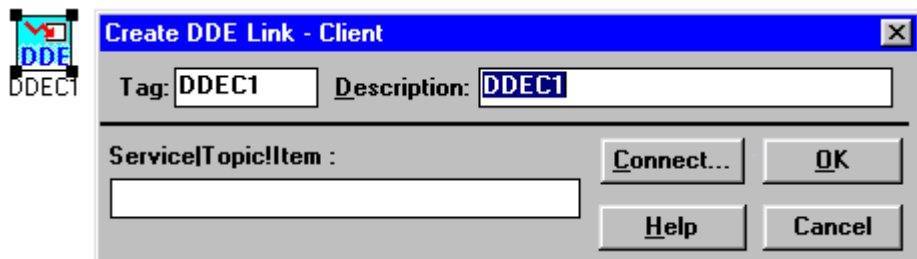
Wejścia. Blok przyjmuje dane i przekazuje je do innego odbiornika Windows za pomocą formatu DDE. Dostępne jest tylko jedno wejście. W przypadku próby podłączenia większej ich ilości zostanie wyświetlona informacja „The input already connected”.

Wyjścia. Blok przekazuje dane wejściowe bezpośrednio do podłączonego bloku.

Odbiornik DDE (DDE Client Block (DDEC))

Blok ten pobiera dane z innej aplikacji Windows. Potrzebuje on informacji na temat adresu, z którego będzie pobierał dane do aplikacji VisiDAQ. Informacje te należy również (tak, jak w przypadku serwera DDE) podać w formacie serwis (Service), temat (Topic) i punkt emisji (Item). Tu także jest konieczne skorzystanie sterownie z dokumentacji danej aplikacji w celu ustalenia ustawień emisji danych.

Blok odbiornika DDE będzie pobierał dane z innej aplikacji po podłączeniu sygnału do jego wejścia przez przewód łączący. Każda liczba bloków może zostać podłączona do odbiornika DDE i wszystkie one mogą korzystać z pobieranych przez niego z serwera innej aplikacji.



Aby przystąpić do ustawień odbiornika DDE należy aplikację, z której mają być pobierane dane uruchomić w tle VisiDAQ. Po wyprowadzeniu bloku na stronę zadań i podłączeniu bloków, które mają korzystać z importowanych z innej aplikacji danych należy dwukrotnie kliknąć ikonę bloku. Wywoła to okno dialogowe odbiornika DDE, w którym należy kliknąć klawisz „Connect...”. Wywoła to z kolei inne okno dialogowe o nazwie „Create DDE Link. Wśród wyświetlonych (co oznacza aktualna gotowość i zdolność do emisji danych w formacie DDE) nazw należy znaleźć aplikację, z której będziemy pobierać dane i zaznaczyć ją. W oknie obok (Topic) zostaną wyświetlone dostępne pliki-tematy. Po wybraniu jednego zostaną wyświetlone dostępne punkty emisji danych (Item). Po wybraniu żądanego należy potwierdzić ścieżkę dostępu klikając „OK”. To pozwoli na zapisanie jej w formacie

Service|Topic!Item

w oknie dialogowym „Create DDE Link – Client”. Połączenie zostało nawiązane. Od tej chwili bloki VisiDAQ mogą korzystać z danych importowanych z innej aplikacji Windows.

Jeżeli nazwa aplikacji, której chcemy użyć nie została wyszczególniona na liście dostępnych, trzeba ręcznie wprowadzić jej nazwę, plik i punkt emisji danych bezpośrednio do okna dialogowego „DDE Link – Client”.

Link DDE może być znów używany po zapisaniu konfiguracji aplikacji i późniejszym ponownym uruchomieniu. Jeżeli aplikacja źródłowa nie będzie uruchomiona podczas ponownego uruchamiania VisiDAQ, system zapyta, czy tę aplikację uruchomić. Jednak, aby to zrobić serwer aplikacji musi mieć wyraźną wzmiankę w wykazie PATH= w pliku AUTOEXEC.BAT. Jeśli nie, VisiDAQ nie będzie w stanie uruchomić potrzebnej aplikacji źródłowej.

Wejścia. Wejść nie ma. W przypadku próby podłączenia jakiegoś VisiDAQ wyświetli komunikat „Cannot accept input”.

Wyjścia. Blok przekazuje importowane z innej aplikacji dane do podłączonego bloku.

Wejście sieciowe (Network Input Block)

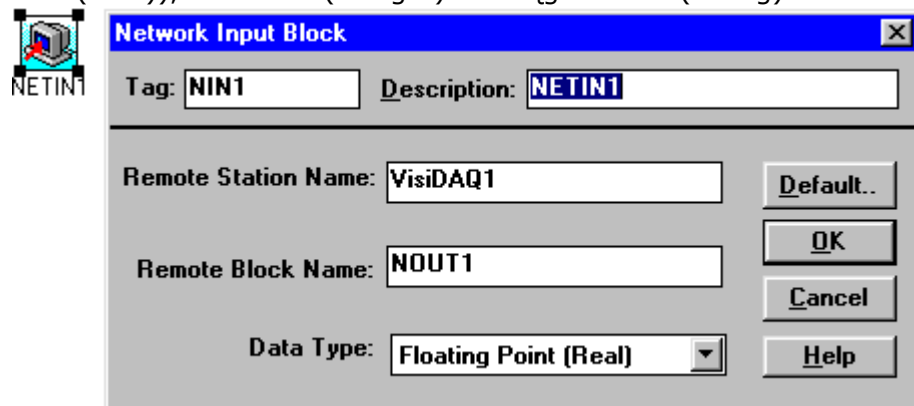
Do podłączenia tego wejścia potrzebny jest w sieci protokół IPX. Blok ten posiada jedynie wyjścia (maksymalnie 8). Znaczący to, że do ośmiu sygnałów może być pobieranych z odległej stacji pracującej również w danym momencie z programem VisiDAQ. Sygnały te w formatach typu liczba rzeczywista (floating point), liczba całkowita (integer) lub ciąg znaków (string) mogą być wykorzystane przez naszą aplikację, wyświetlone lub przesłane dalej do innej poprzez wyjście sieciowe. Blok ten może używać dla wszystkich 8 kanałów tylko jednego typu danych. Jeżeli potrzebne jest przesłanie sygnału o jeszcze innym typie danych należy użyć dodatkowej pary bloków z ustawionym żądanym typem. Jedna para wejścia i wyjścia w połączonych siecią aplikacjach może przekazywać dane tylko jednego typu. Jeśli tak nie będzie, dane będą błędnie odczytane po stronie odbiornika. Po dwukrotnym kliknięciu ikony pokazuje się okno dialogowe, w którym ustala się połączenie z inną aplikacją VisiDAQ. Status połączenia zostanie wyświetlony w poszczególnych jego częściach.

Aplikacja docelowa (Remote Station Name)
Nazwa aplikacji, z której mają być pobierane dane.

Blok docelowy (Remote Block Name)
bloku w aplikacji, z którego mają być pobierane dane.

Typ danych (Data Type)

Format danych, za pomocą którego będą one przekazywane. Są to: liczba rzeczywista (Floating Point(Real)), całkowita (Integer) lub ciąg znaków (String).



Użytkownik może zdefiniować wartość domyślną dla każdego z ośmiu kanałów bloku wejścia sieciowego. Oznacza to, że blok ten będzie podawał do podłączonego innego bloku sygnał o nastawionej wartości w wypadku, gdy nie będzie otrzymywał sygnału z innej aplikacji VisiDAQ. Można użyć tego sygnału do określenia, czy połączenie między nimi jest sprawne.

Wyjście sieciowe (Network Output Block)

Do korzystania z bloku wyjścia sieciowego potrzebny jest protokół IPX. Blok posiada jedynie wejścia (maksymalnie 8). Oznacza to, że ma możliwość wysyłania do ośmiu sygnałów do innej aplikacji VisiDAQ połączonej z tą za pomocą sieci. Sygnały te mogą być wysyłane w różnym formacie. Może to być liczba rzeczywista (Floating Point (Real)), całkowita (Integer) lub ciąg znaków (String). Jeden blok może jednak przysyłać tylko jeden typ danych. Jeżeli zachodzi potrzeba przesłania większej ich ilości, należy użyć kilku par wejść – wyjść sieciowych, każda dla danego typu danych. Jeżeli nie dopełni się tego warunku dane będą po stronie odbierającej sygnał odczytywane nieprawidłowo. Po dwukrotnym kliknięciu pokaże się

okno dialogowe z poszczególnymi częściami adresu, pod który mają być wysyłane dane. Status połączenia będzie wyświetlany podczas działania aplikacji.

Aplikacja docelowa (Remote Station Name)

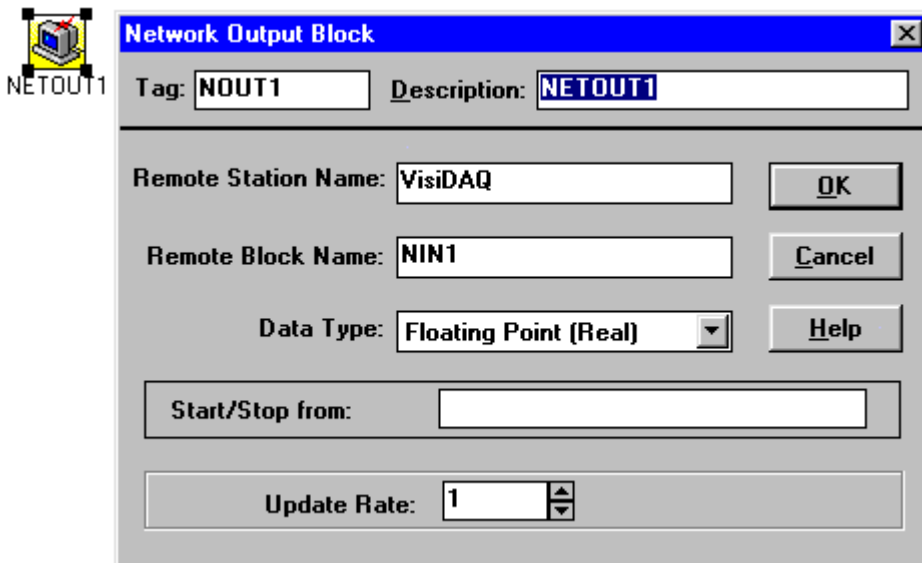
Nazwa aplikacji, do której dane będą wysyłane.

Blok docelowy (Remote Block Name)

Nazwa bloku w aplikacji, do którego będą wysyłane dane.

Typ danych (Data Type)

Format danych, w którym będą one wysyłane. Do wyboru są: liczba rzeczywista (Floating Point), całkowita (Integer) lub ciąg znaków (String).



Kontrola emisji danych (Start / Stop from)

Blok wyjścia sieciowego posiada jedno wejście cyfrowe do kontroli jego działania. Aby uruchomić emisję danych należy wysłać do tego wejścia sygnał cyfrowy o wartości „1” z dowolnego bloku sterującego lub po prostu z klawisza umieszczonego na stronie prezentacji. Aby wstrzymać emisję wystarczy w ten sam sposób wysłać do tego wejścia sygnał cyfrowy o wartości „0”. Jeśli nie zostanie tu doprowadzony żaden sygnał, blok będzie wysyłał dane od chwili uruchomienia aplikacji do jej zamknięcia. Należy pamiętać, że blok odbierający będzie dostarczał do aplikacji ustawioną wcześniej wartość domyślną, jeżeli nie otrzyma sygnału od naszej aplikacji.

Częstotliwość próbkowania (Update Rate)

Funkcja ta jest praktycznie dzielnikiem i pozwala na zmianę (zmniejszenie) efektywnej częstotliwości próbkowania na inną od tej, w której pracuje reszta aplikacji. Jeżeli np. aplikacja pracuje z częstotliwością 100Hz, natomiast potrzebna częstotliwość emisji danych wynosi 20Hz. Wystarczy wtedy ustawić w tym polu wartość dzielnika na 5. W takim wypadku w dalszym ciągu będziemy wysyłać sygnał z częstotliwością 100Hz jeżeli będziemy go wysyłać do bloku zapisującego lub wyświetlającego, jednak tylko jedna z pięciu próbek będzie „prawdziwa”. Pozostałe będą jedynie jej kopiami. Dzielnik ten może przyjmować wartości z zakresu od 1 do 32767.

Alarm (Alarm Log Block)

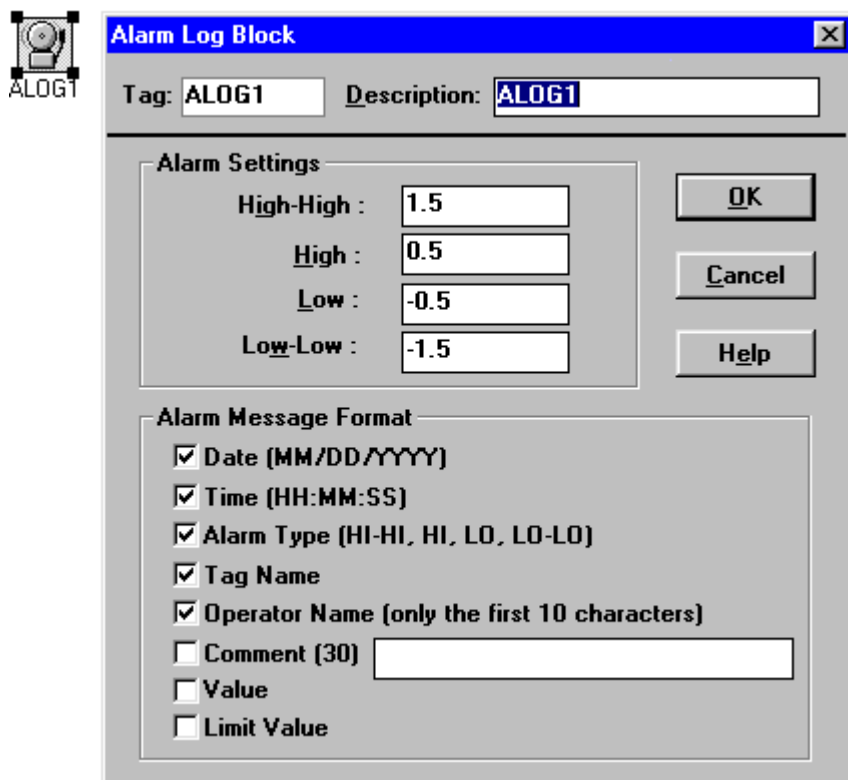
Blok ten posiada wejście i wyjście. Analizuje on sygnał wejściowy, a informacje o przekroczonych wartościach przekazuje do pliku z rozszerzeniem typu *.ELF.

Informacja ta może być wyświetlona podczas pracy aplikacji (Event Log Viewer). Nastąpi to w chwili przekroczenia jednej z ustawionych przez użytkownika granic:

- 1) jeżeli sygnał znajdzie się powyżej wartości maksymalnej
- 2) jeżeli sygnał znajdzie się pomiędzy wartością maksymalną a wysoką

- 3) jeżeli sygnał znajdzie się pomiędzy wartością wysoką a niską
- 4) jeżeli sygnał znajdzie się pomiędzy wartością niską a minimalną
- 5) jeżeli sygnał znajdzie się poniżej wartości minimalnej

Blok ten wyprowadza również sygnał o wartości zależnej od tego, w którym zakresie wartości znalazł się sygnał wejściowy. Może on być wykorzystany do sterowania blokiem wyświetlającym komunikaty, które z kolei mogą posłużyć do poinformowania użytkownika o przekroczeniu ustawionych zakresów.



- 1) Blok wyprowadza sygnał o wartości cztery (4)
- 2) Blok wyprowadza sygnał o wartości dwa (2)
- 3) Blok wyprowadza sygnał o wartości zero (0)
- 4) Blok wyprowadza sygnał o wartości jeden (1)
- 5) Blok wyprowadza sygnał o wartości trzy (3)

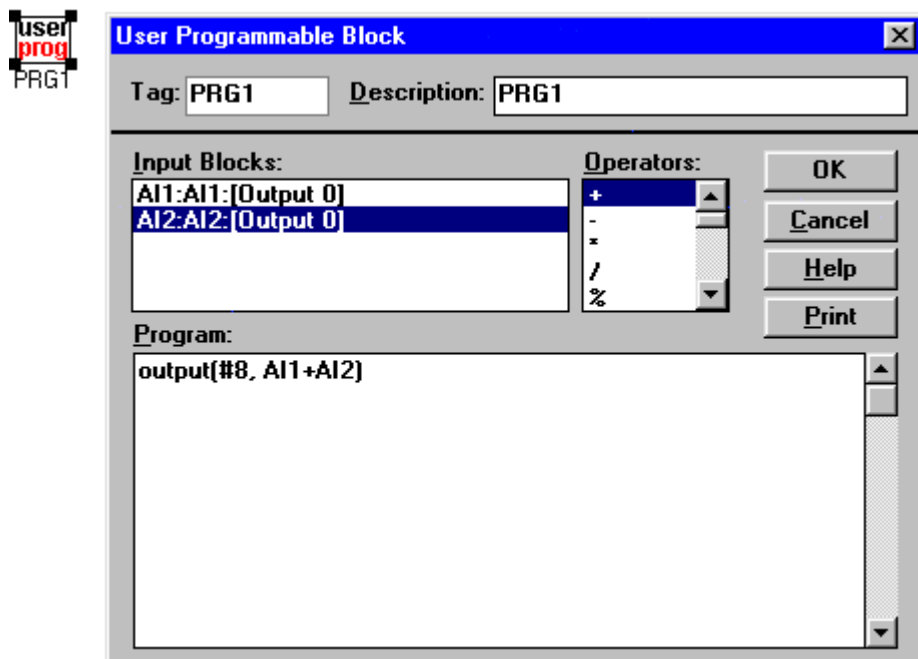
Podczas pracy aplikacji komunikat alarmu może być przekazywany do pliku jeśli zostanie dwukrotnie kliknięta ikona „Alarm Log Event” (ALOG1 HI, itd.). Podczas przekazywania alarm zostanie wyświetlony w kolorze czerwonym. Aby okno dialogowe bloku „Event Log” ukazywało się podczas pracy aplikacji należy to umożliwić przez ustawienia w menu „View”.

Wejścia. Blok pobiera dane, analizuje je i zapisuje w pliku typu „*.elf” za pomocą bloku VisiDAQ Event Log File. Dane te mogą również być wyświetlane podczas pracy aplikacji poprzez blok Event Log Viewer.

Wyjścia. Blok przekazuje dane do podłączonego bloku.

Blok programowy (User Programmable Block)

Posiada on wejścia (maksymalnie 8) oraz wyjścia (maksymalnie 8). Blok umożliwia wprowadzenie sygnałów i wykonywanie na nich operacji matematycznych lub logicznych. Można go również użyć do generowania zmiennej dostarczanej z kolei do innego bloku jako jedna z wielu innych wartości wejściowych. Blok ten może obsługiwać jednocześnie maksymalnie 8 kanałów wejściowych i 8 wyjściowych.



Blok ten został zaprojektowany tak, aby zapewnić użytkownikowi pełną elastyczność programowania. Można dzięki niemu dokonywać porównań, obliczeń, generować komunikaty zależne od wartości jednego z sygnałów wejściowych oraz tworzyć rozgałęzienia i pętle programowe. Do programowania jest tu używany język C, nieco tylko uproszczony (dostępne nie wszystkie komendy). Jednakże jest to jedynie tłumacz, nie kompilator. Tłumaczy to fakt, że szybkość funkcjonowania bloku jest spowodowana prostotą programu. Sam program może obsługiwać wiele wejść (bez ograniczeń) dostarczających sygnał z dowolnego typu bloków. Natomiast wartości wyjściowe mogą być przekazywane do dowolnej ilości bloków docelowych. Blok umożliwia napisanie programu służącego do manipulowania sygnałami wejściowymi, dokonywania obliczeń wartości wyjściowych bazowanych na wartościach wejściowych lub po prostu przeskoczyć do następnego skanu (próbki) nie wyprowadzając żadnego wyniku aktualnego pomiaru. W takim wypadku nie zostanie on po prostu w danym momencie przeanalizowany przez blok, do którego jest wyprowadzony sygnał z tego wyjścia. Używając tego bloku można również wstrzymać pracę całej aplikacji.

Blok programowy w swoim oknie dialogowym zawiera trzy podokna: okno bloków wejściowych (the input blocks box), okno operacji (the operators box) oraz okno edycyjne, w którym pisze się program (the program editor box). W pierwszym oknie jest wyświetlana lista aktualnie podłączonych bloków dostarczających sygnał wejściowy. Drugie okno zawiera wykaz dostępnych operacji możliwych do wykonywania na wartościach tych sygnałów (objaśnienia poniżej). Trzecie okno jest przeznaczone do projektowania programu. Kliknięcie na blok sygnału wejściowego powoduje wstawienie jego nazwy do programu w oknie edycyjnym. Taki sam efekt daje kliknięcie na konkretną operację w drugim oknie.

Program

Składa się on z jednego lub kilku poleceń o poprawnej składni.

[Polecenia]

Składają się ze słów kluczowych, wyrażeń oraz innych [poleceń]. [Polecenie], które jest częścią innego jest traktowane jako bazę (body) zawierającego go [polecenia]. „Powrót karetki” lub znak wysuwu wiersza w edytorze tekstu w programie nie ma znaczenia, dlatego polecenie może zarówno mieścić się w jednej linii, jak i być rozłożone na kilka. Blok programowy rozpoznaje słowa kluczowe:

- 1) wyrażenie
- 2) if (jeżeli)
- 3) if (jeżeli), else (w przeciwnym wypadku)
- 4) while (podczas)
- 5) output (wyjście)
- 6) skip (przeskok)
- 7) stop (zatrzymanie)
- 8) display (wyświetlenie)
- 9) [substr]

- 10) [sprintf]
- 11) kombinacje nazw bloków (np. AI1, DO2, itd.)
- 12) kombinacje liter i cyfr od a - z, oraz od a1 - z1 do a9 - z9.

Słowo kluczowe „output” pozwala na wysłanie wyrażenia do innego podłączonego do wyjścia tego bloku. Dane wyjściowe z bloku programowego mogą być wyprowadzane jako liczba rzeczywista (floating point value), całkowita (integer) lub ciąg cyfr (string). Ponieważ blok ten ma możliwość wyprowadzania 8 różnych wartości danych prawidłową formą słowa kluczowego wyjścia jest wyrażenie „output”. Wyrażenie to zawiera dwa parametry. Pierwszy to numer kanału wyjściowego (#1 - #8), natomiast drugi to wartość sygnału, która ma być wyprowadzona tym wyjściem. Jeżeli dana ta ma być w formie ciągu cyfr, słowo „string” musi znaleźć się w cudzysłowie („String”). Jeśli w wyrażeniu nie zostanie wymieniony numer kanału wyjściowego, sygnał ten zostanie podany do kanału domyślnego (pierwszego, tzn. #1). Przykłady zastosowań:

- 1) output(#2, „This String”); // wyprowadza ciąg cyfr do kanału 2
- 2) output „Another String”; // wyprowadza ciąg cyfr do kanału 1 (domyślny numer kanału)
- 3) output(#8, „String” + a4); // wyprowadza ciąg cyfr + a4, gdzie a4 = „Another String” do kanału #8
- 4) output(#1, AI1 + AI2); // wyprowadza nazwy bloków AI1 + AI2 do kanału #1
- 5) output(#8, 1); // wyprowadza sygnał o wartości 1 do kanału #8

Kiedy wyjście tego bloku jest podłączone do wejścia następnego, pojawi się okno dialogowe, w którym należy wskazać numer żądanego kanału wyjściowego (#1 - #8). Jeżeli program nie przydzieli wartości do konkretnego kanału określonego w danym momencie, wartość wyjściowa tego kanału będzie równa zero dopóki wyjście jest poprawne.

Słowo kluczowe „skip” pozwala na przeskoczenie (przeczekanie) do następnego skanu lub próbki przed jej przeanalizowaniem.

Słowo kluczowe „stop” pozwala na natychmiastowe wstrzymanie aplikacji.

Słowo kluczowe „display” pozwala na zmianę stron wyświetleń w czasie pracy aplikacji. Prawidłowa składnia słowa kluczowego „display” powinna zawierać wyrażenie display oraz numer strony wyświetleń w nawiasach (N). Na przykład:
display(2); // zmienia stronę wyświetleń na stronę nr 2

Słowo kluczowe „substr” pozwala na [parse] ciągu cyfr na [sub-string] podczas pracy aplikacji. Prawidłowa składnia dla słowa kluczowego „display” składa się z wyrażenia „substr” oraz trzech parametrów. Pierwszy z nich jest to ciąg cyfr lub zmienny ciąg cyfr przeznaczony do [parse]. Drugi parametr zawiera numer cyfry w ciągu, od którego ma zacząć się [parsing] (pierwsza cyfra w ciągu jest oznaczona numerem jeden (1)). Trzeci parametr odnosi się do długości [sub-string] lub do ilości cyfr w ciągu, które mają zostać [parse]. Na przykład:

- ```
d5 = „This string”;
```
- 1) substr(“abcdef”, 5, 2); // zaczyna od piątego znaku w ciągu „e”, [parse out] dwa znaki „ef”
  - 2) substr(d5, 6, 6); // zaczyna od szóstego znaku „s”, [parse out] sześć znaków „string”

Słowo kluczowe „sprintf” pozwala na sformatowanie i zgromadzenie serii znaków i wartości w zmiennym buforze. Prawidłowa składnia słowa kluczowego „sprintf” zawiera wyrażenie „sprintf” oraz trzy parametry. Pierwszy parametr jest to ciąg znaków lub zmienny ciąg znaków, który ma zostać sformatowany w buforze a następnie zapisany. Drugi parametr zawiera typ danych argumentu. Trzeci parametr zawiera argument który ma być formatowany. Argument jest konwertowany (formatowany) i przesyłany do bufora zgodnie z zaznaczonym odpowiednim formatem. Format zawiera standardowe znaki i zachowuje tę samą formę i funkcjonalność, jak format argumentu w systemie ANSI (American National Standards Institute – urząd normalizacyjny) C funkcji „sprintf()”. Jednakże tylko jeden argument może być formatowany przy użyciu tej wersji funkcji „sprintf”. Na przykład:

// Ten program wykorzystuje funkcję „sprintf” do formatowania zmiennej

// danej i umieszczenia jej w ciągu znaków o nazwie „b”.

```
s = „komputer”;
c = „1”;
i = 35;
f = 1.7320534;
//Formatowanie zmiennej danej:
sprintf(b, „\tString: %s\n”, s)
sprintf(b, “\tInteger: %3d\n”, i);
sprintf(b, “\tReal: %6.3f\n”, f);
```

Słowo kluczowe „statement” (operacja) pisane kursywą oznacza pojedynczą lub złożoną operację. Na złożoną operację składa się jedna lub więcej operacji zamkniętych w nawiasie klamrowym { }. Należy zauważyć, że złożonej operacji nie ma potrzeby zakańczać znakiem średnika (;), co trzeba czynić w przypadku pojedynczej operacji. Dlatego nie trzeba go umieszczać za zamkniętą nawiasem złożonej operacji. Nawias ten jest tylko sposobem zgrupowania pojedynczych operacji.

Aby załączyć komentarze wewnątrz programu należy użyć dwu ukośników (//) (forward slash marks) do rozpoczęcia każdej linii komentarza.

### Wyrażenia

Wyrażeniami są tu sekwencje [operantów] (zmiennych i liczb) oraz operatorów (funkcji matematycznych), według których są przeliczane wartości sygnałów wejściowych bloku programowego. Po przeliczeniu otrzymuje się sygnał wyjściowy w postaci liczby lub wyniku logicznego prawda / fałsz. Najprostszym wyrażeniem może być stała (operant) bez żadnego operatora.

Przykłady prawidłowych wyrażeń:

```
3
AI1+AI2
(abs(AI1)*2.5) + 1
AI1 > 0 && AI2 < 0
```

Ostatni przykład zawiera wyrażenie logiczne. Niektóre z operatorów mają wyższy stopień pierwszeństwa, niż inne. Operator z wyższym stopniem jest zawsze obrabiany w pierwszej kolejności, natomiast te z niższym później. Poniższa tabela (Tabela 1) zawiera operatory, ich objaśnienia i kierunek ich działania w kolejności od najwyższego do najniższego stopnia pierwszeństwa.

Tabela 1. Operatory, objaśnienia i kierunek ich działania

| Symbol    | Typ operacji               | Kierunek działania |
|-----------|----------------------------|--------------------|
| ( )       | wyrażenie                  | od lewej do prawej |
| -         | [unary]minus               | od prawej do lewej |
| !         | [unary]zaprzeczenie        |                    |
| *         | mnożenie                   | od lewej do prawej |
| /         | dzielenie                  |                    |
| %         | [reszta]                   |                    |
| + -       | [dodatek]                  | od lewej do prawej |
| < > <= >= | relacja                    | od lewej do prawej |
| == !=     | jest równy, nie jest równy | od lewej do prawej |
| &         | [bitwise]„i”               | od lewej do prawej |
|           | [bitwise] „lub”            | od lewej do prawej |
| ~         | [bitwise] „nie”            | od prawej do lewej |
| ^         | [bitwise] „x lub”          | od lewej do prawej |
| <<        | przesunięcie w lewo        | od lewej do prawej |
| >>        | przesunięcie w prawo       |                    |
| &&        | funkcja logiczna „i”       | od lewej do prawej |
|           | funkcja logiczna „lub”     | od lewej do prawej |
| =         | zadanie                    | od prawej do lewej |

Tabela 2. Funkcje matematyczne

| Funkcja  | Argument                                      | Wynik                                                                  |
|----------|-----------------------------------------------|------------------------------------------------------------------------|
| abs()    | liczba całkowita<br>lub rzeczywista           | wartość bezwzględna                                                    |
| cos()    | liczba rzeczywista lub<br>całkowita (radiany) | cosinus argumentu                                                      |
| int()    | liczba rzeczywista                            | zamienia argument z liczby<br>rzeczywistej na całkowitą                |
| float()  | ciąg cyfr lub liczba całkowita                | zamienia argument z ciągu cyfr lub<br>liczby całkowitej na rzeczywistą |
| rnd()    | liczba całkowita                              | losowo wybrana, zaczynając od<br>argumentu, liczba                     |
| sin()    | liczba rzeczywista lub<br>całkowita (radiany) | sinus argumentu                                                        |
| sqr()    | liczba rzeczywista<br>lub całkowita           | pierwiastek kwadratowy argumentu                                       |
| ln()     | liczba rzeczywista<br>lub całkowita           | logarytm naturalny argumentu                                           |
| log10()  | liczba rzeczywista<br>lub całkowita           | logarytm dziesiętny argumentu                                          |
| exp()    | liczba rzeczywista<br>lub całkowita           | stała „e” do potęgi „x” argumentu                                      |
| pow(x,y) | liczba rzeczywista<br>lub całkowita           | argument „x” do potęgi „y”                                             |
| tan()    | liczba rzeczywista lub<br>całkowita (radiany) | tangens argumentu                                                      |

Funkcja „int()” zasługuje tu na szczególną uwagę. Jeżeli sygnał jest dostarczany do bloku programowego jako liczba rzeczywista, natomiast jest ustawiona, aby była przyjmowana przez blok jako całkowita, jej poprzednia wartość może zostać przywrócona przez funkcję „int()”. W niektórych przypadkach może zająć potrzeba zamiany z odzyskanej liczby rzeczywistej na całkowitą. To również jest możliwe za pomocą tej funkcji. Zamiana danych z liczby rzeczywistej na całkowitą odbywa się przez tzw. „zaokrąglanie”. Na przykład, jeżeli pierwotna wartość wynosi 1.49, po zamianie będzie wynosiła 1, natomiast dla wartości 1.51 zmieni się na 2. Funkcja ta, więc zmienia liczbę rzeczywistą na najbliższą całkowitą.

### Zmienne

Mogą być one (a – z oraz a1 – z1 do a9 – z9) użyte do różnych celów. Należy pamiętać, że jeżeli konkretna zmienna jest użyta w jednym bloku programowym, jej wartość będzie odczytywana tak samo w innych blokach programowych używających tej samej zmiennej. Ten efekt może być w niektórych przypadkach pożądanym, jednak praktycznym jest używanie w różnych blokach programowych innych zmiennych. Wyświetlane zmienne bloków wejściowych (nazwy AIN, DIN, itd.) mogą być również używane jako operandy.

Na przykład:

```

if (AI1 > AI2)
{
 a = AI1 * 0.117; // przykład, jak
 output a; // użyć komentarza
}

```

Powyższy program sprawdza, czy sygnał analogowy 1 (AI1) jest wartościowo większy od sygnału analogowego 2 (AI2). Jeżeli tak jest, zmienna „a” jest tu użyta do reprezentowania wyniku AI1 i stałej 0.117 (współczynnik skalowania). Wynik ten jest wtedy wysyłany do podłączonego do wyjścia bloku programowego innego bloku. Wszystkie operacje zawarte wewnątrz nawiasu klamrowego { } są wykonywane tylko wtedy, gdy AI1 > AI2.

Wejścia. Blok posiada 8 kanałów wejściowych. Za ich pomocą można wprowadzić dane i dokonywać na nich operacje matematyczne.

Wyjście. Blok posiada 8 kanałów wyjściowych, za pomocą których można wyprowadzać przeliczone sygnały do podłączonych bloków.

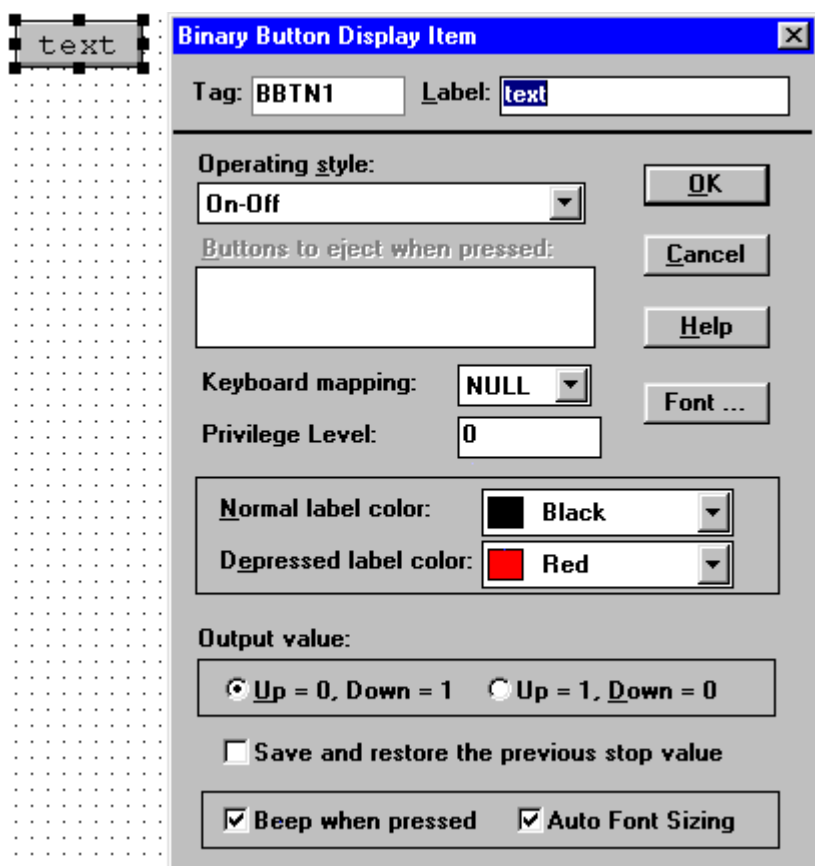
### Strona wyświetleń (Display Designer)

Poniższa część opisuje funkcje i możliwości strony wyświetleń. Strona ta jest narzędziem pozwalającym na utworzenie „panelu wyświetlającego – sterującego będącego dynamicznym interfejsem podczas pracy aplikacji. Można stworzyć panel służący do testowania sprzętu, ale również może być on zastosowany do Profesjonalnego monitorowania procesu przemysłowego. Można również w celu zilustrowania analizowanego procesu zastosować jako tło do odczytywanych i wizualizowanych danych bitmapy (grafiki o rozszerzeniu \*.bmp). Panel taki pozwala na monitorowanie, nadzorowanie, oraz kontrolowanie analizowanego procesu podczas pracy aplikacji. Przy pełniejszym wykorzystaniu jego funkcji ekran może być pełen liczników, słupków, wskaźników, wykresów, klawiszy, dynamicznych odczytów tekstowych i wielokolorowych kontrolek.

### Przycisk dwupołożeniowy (Binary Button Control Display Item)

Przycisk ten może być naniesiony na stronę wyświetleń i różnorodnie formowany w zakresie możliwości aplikacji. Automatycznie jest mu też nadawana nazwa (Tag). Po naniesieniu możliwa jest w każdej chwili (oprócz pracy aplikacji) dowolna zmiana jego wielkości, kształtu, oraz położenia. Ten moduł jest sygnałem wyjściowym ze strony wyświetleń. Klawisz ten jest naciskany za pomocą myszy. Moduł klawisza wysyła w zależności od ustawienia sygnał cyfrowy „0” lub „1” w położeniu wciśniętym lub zwolnionym. Sygnał ten jest przesyłany do podłączonego bloku. Klawisz ten może być również przełączany za pomocą klawisza „enter” z klawiatury, pod warunkiem, że „focus” jest aktualnie ustawiony na niego („focus” – standardowy termin w systemie Windows – oznacza klawisz, który aktualnie jest kontrolowany za pomocą klawiatury). Focus może być przenoszony z jednego przycisku na drugi za pomocą klawisza tabulatora (Tab). Dodatkowo przyciski mogą być obsługiwane z klawiatury bez zmiany focusa za pomocą innych klawiszy. Taki efekt można uzyskać zmieniając klawisz aktywny w oknie dialogowym (Keyboard Mapping).

Rozmiar i własności czcionki można ustawiać w oknie dialogowym przy użyciu przycisku „Font...”.



#### Nazwa modułu (Tag)

W tym polu znajduje się nazwa systemowa (bez możliwości zmiany) modułu). Reprezentuje ona wartość wyjściową danego przycisku przy jego wyborze jako sygnał wejściowy do bloku na stronie zadań. W oknie dialogowym danego bloku należy wówczas znaleźć tę właśnie nazwę systemową, dodatkowo uzupełnioną dla ułatwienia w opis (etykietkę). Sygnał wyjściowy każdego przycisku jest dostępny dla dowolnej liczby bloków. Jest to przydatne, gdy zachodzi potrzeba kontrolowania funkcji kilku bloków jednym przyciskiem.

#### Etykietka (Label)

W tym polu należy wpisać tekst informujący o funkcji, jaką ma pełnić przycisk. Tekst ten zostanie wyświetlony na przycisku. Jego maksymalna długość to 30 znaków.

#### Sposób działania (Operating style)

W tym polu są do wyboru trzy tryby pracy przycisku: „On-Off”, „Momentary” oraz „Radio Button”.

- Wł.-Wył. (On-Off) – przycisk działa, jak przełącznik dwupołożeniowy. Każde użycie powoduje zmianę jego położenia. Każdemu z nich jest przyporządkowany sygnał „0” lub „1” wysyłany do podłączonego bloku.

- Działanie impulsowe (Momentary) – każde użycie powoduje chwilową zmianę jego położenia. Zaraz potem przycisk powraca do poprzedniego.

- Przycisk sprzężony (Radio Button) – działa, jak przy opcji pierwszej, jednak posiada dodatkową funkcję. Polega ona na tym, że w chwili wciśnięcia zostają zwolnione wszystkie sprzężone z nim inne przyciski w danym momencie wciśnięte. Aby uruchomić tę funkcję, należy ją wybrać, a następnie w polu „Buttons to eject when pressed” wprowadzić nazwy przycisków, które mają być sprzężone. Sygnały wyjściowe tej opcji są takie same, jak przy opcji „Wł.-Wył.”.

#### Przyporządkowanie klawiszy (Keyboard Mapping)

Każdy przycisk może posiadać przyporządkowany mu klawisz, za pomocą którego będzie zmieniane jego położenie. W tym polu są do wyboru klawisze funkcyjne od F2 do F8 oraz wszystkie litery od A do Z.

#### Zapis ostatniego położenia przycisku (Save and restore the previous stop value)

Po zaznaczeniu tej funkcji ostatnie położenie przycisku zostanie zapamiętane i po ponownym uruchomieniu aplikacji jego położenie będzie takie samo, jak przed jej zamknięciem. Należy zwrócić jednak uwagę, że funkcja ta będzie działać tylko, jeżeli komputer zostanie wyłączony prawidłowo. Jeżeli zostanie użyta kombinacja klawiszy „Ctrl + Alt + Del” lub zostanie on wyłączony zanim zostanie zamknięta aplikacja, ostatnie położenie przycisku nie zostanie zapamiętane.

#### Stopień uprzywilejowania (Privilege level)

Ta funkcja jest używana do ochrony systemu kontroli przed nieuprzywilejowanymi użytkownikami. Może on przyjmować wartości od 0 do 255, z tym, że im większa liczba, tym wyższy stopień. Na przykład, jeśli jest on ustawiony na 100, to stopień uprzywilejowania użytkownika musi być większy lub równy 100, aby mógł użyć tego przycisku.

#### Kolor etykiетки (Label Color)

W tym polu można wybrać, w jakim kolorze ma pokazywać się na przycisku tekst etykiетки. Można wybrać spośród 16 kolorów dla obu położań przycisku (zwolnionego „Normal label color” oraz wciśniętego „Depressed label color”).

#### Wartość wyjściowa (Output value)

W polu tym można wybrać, jaką wartość w zależności od położenia przycisku ma on wysyłać. Do wyboru są dwie opcje:

0 – przycisk zwolniony, 1 – wciśnięty „Up=0, Down=1” lub odwrotnie „Up=1, Down=0”

#### Funkcje opcjonalne.

Sygnał dźwiękowy (Beep when pressed). Naciśnięcie jest sygnalizowane krótkim dźwiękiem.

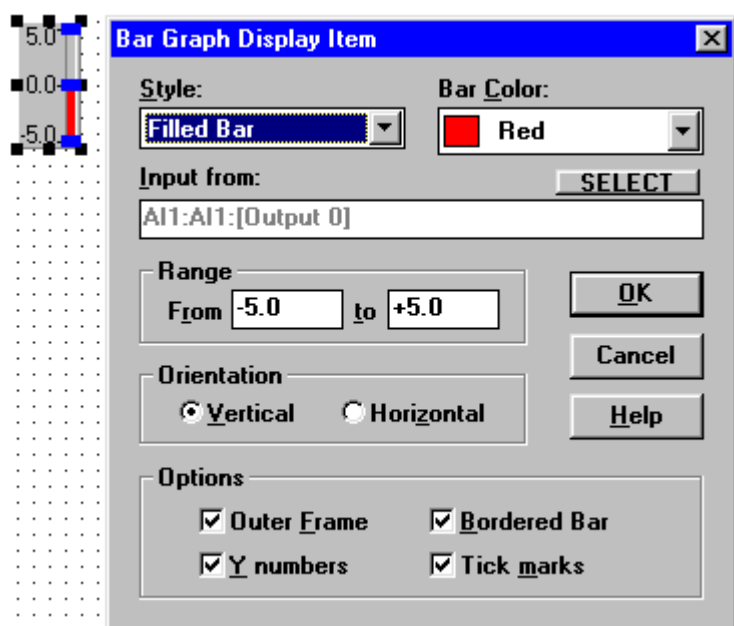
Dostosowanie formatu czcionki (Auto Font Sizing). Wielkość czcionki jest automatycznie dostosowywana do rozmiaru przycisku.

## Wskaźnik poziomego sygnału (Bar Graph Display Item)

Wskaźnik ten pozwala na graficzne przedstawienie wartości sygnału do niego doprowadzonego. Jego wielkość, kolor oraz dodatkowe własności mogą być dowolnie wybierane a później zmieniane. Po dwukrotnym kliknięciu na wskaźnik mamy do dyspozycji okno dialogowe. Są w nim wszystkie opcje, jakie można konfigurować w tym module.

### Źródło sygnału (Input from)

Pole to informuje, z jakiego źródła aktualnie są pobierane dane. Na początku jest ono puste. Aby dokonać w nim zapisu lub zmiany, należy użyć przycisku „Select”. Nastąpi wyświetlenie okna o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Sygnał ten będzie dynamicznie interpretowany i wyświetlany.



### Rodzaj wskaźnika (Style)

Może to być realizowane w postaci rosnącego i malejącego słupka (Filed Bar) lub przesuwanego się wskaźnika (Moving Mark).

### Kolor wskaźnika (Bar Color)

Zarówno kolor słupka, jak i wskaźnika można zmieniać w zakresie 16 kolorów.

### Zakres analizy (Range)

Dodatkowo można określić jego zakres wpisując w pola „From” zakres dolny, zaś w pole „to” górny.

### Orientacja (Orientation)

Do wyboru są dwa ułożenia a jednocześnie kierunki ruchu wskaźnika: pionowy lub poziomy (Vertical or Horizontal).

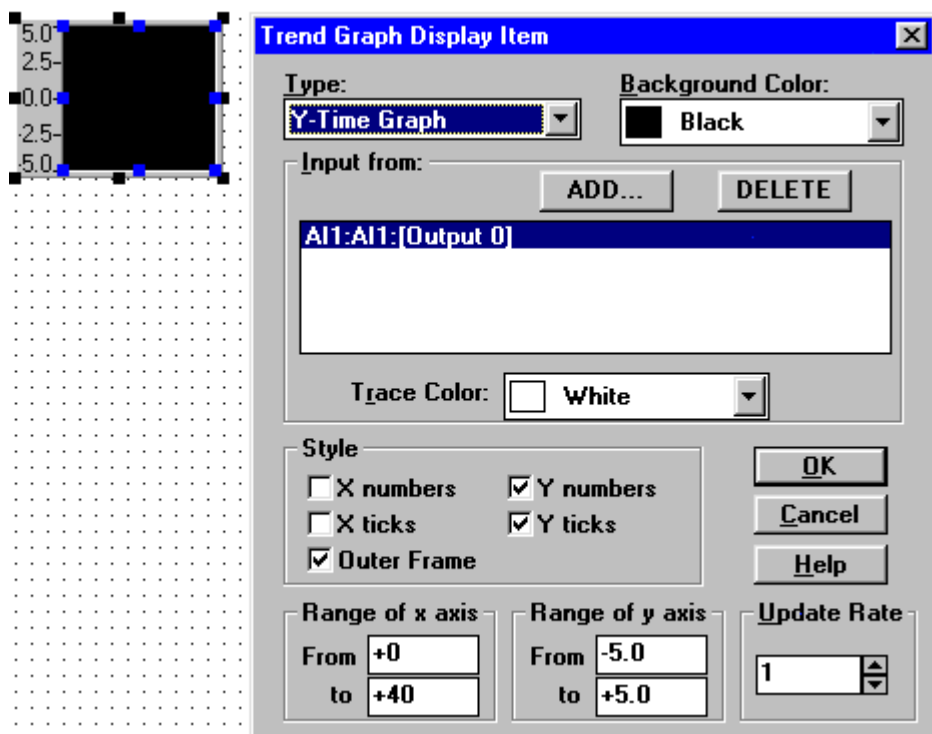
### Ogólne własności (Options)

Należą do nich: obwódka zewnętrzna (Outer Frame), obwódka samego wskaźnika (Bordered Bar), wartości na osi Y (Y numbers) oraz znaczniki skali na osi Y (Tick marks). Mogą one być włączane lub wyłączane w dowolnych kombinacjach.

## Wykres zmiennej Y w czasie (YT Graph Display Item)

Wykres ten może być rysowany przez jedną lub kilka (maksymalnie 8) zmiennych pobieranych z różnych bloków strony zadań lub modułów strony wyświetleń za pomocą przyporządkowanych im nazw. Kolor i

rozmiar wykresu może być ustawiony i dowolnie zmieniany. Analizowana dana jest wartością na osi Y natomiast na osi X odkładany jest czas.



#### Kolor tła (Background Color)

Do wyboru jest 16 kolorów tła wykresu. Należy go jednak wybrać tak, aby późniejszy wybrany kolor linii wykresu był na tym tle dobrze widoczny.

#### Źródło sygnału (Input from)

W tym polu znajdują się dwa okna dialogowe. Jedno wyświetla wybrane źródła, natomiast drugie przyporządkowuje każdemu z nich kolor linii wykresu (Trace Color). Przyciskiem „Add” wywołuje się okno o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Wszystkie nazwy wybranych bloków zostaną wyświetlone w tym oknie. Sygnały te będą dynamicznie interpretowane i wyświetlane przy pomocy linii wykresu, których kolor należy wybrać dla każdej z nich. Aktualnie jest ustawiany kolor dla podświetlonej w oknie listy połączeń linii.

#### Wygląd wykresu (Style)

Opcjonalnie można wybrać dodatkowe cechy wyświetlanego wykresu. Są to: wartości na osi X (X numbers), znaczniki skali na osi X (X ticks) (standardowo obydwa wyłączone), wartości na osi Y (Y numbers), znaczniki skali na osi Y (Y ticks) oraz obwódka zewnętrzna.

W zależności od tego, czy chcemy obserwować cały zakres analizowanego sygnału, czy też dokładniej, ale tylko jego część, należy ustawić zakres dla osi X (czasu) i osi Y (sygnału). Dolną granicę „From” i górną „to” wpisuje się z klawiatury. Należy jednak pamiętać, że im więcej linii na wykresie oraz im większy zakres zostanie wybrany, tym więcej pamięci zostanie zajęte. Jest to spowodowane ilością wyświetlanych punktów każdej z linii wykresu.

#### Częstotliwość skanowania (Update Rate)

W polu tym należy ustawić dzielnik, przez który zostanie podzielona standardowa częstotliwość. Np. jeśli jest ona równa 100 Hz a potrzebna jest 20 Hz, należy w polu tym wpisać lub ustawić za pomocą strzałek liczbę 5. Powoduje to, że tylko jedna z pięciu próbek jest wyświetlana, pozostałe są ignorowane. Funkcja ta jest przydatna w przypadku migotania zdarzającego się przy wysokiej częstotliwości próbkowania.

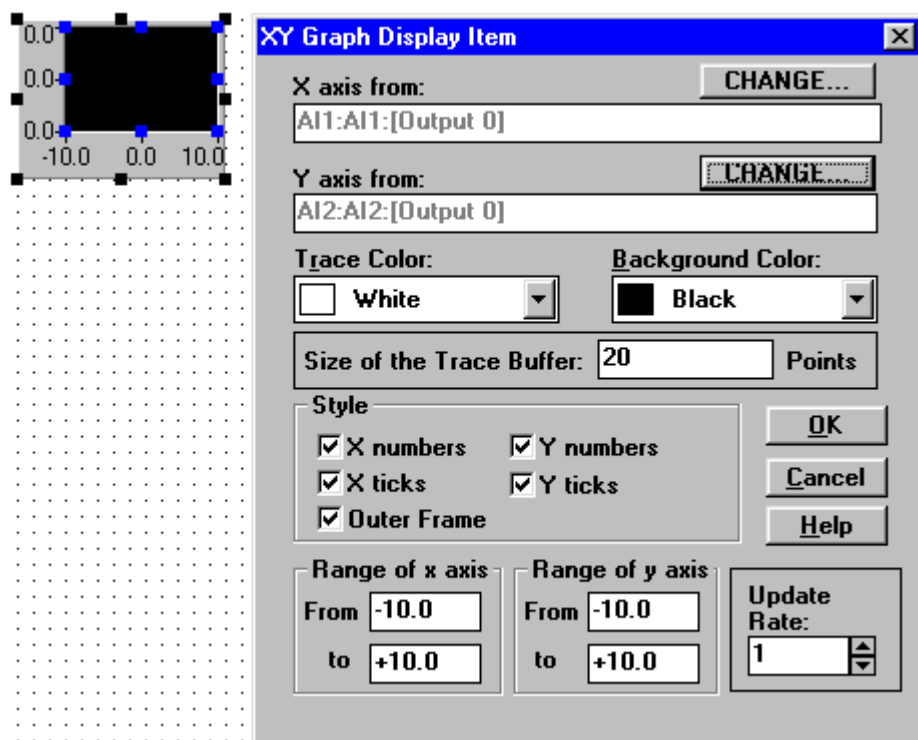
## Wykres zależności dwu sygnałów (XY Graph)

Pozwala on na sprzężenie dwu sygnałów z oddzielnych bloków przez odłożenie ich wartości na osi X i osi Y. Wygląd wykresu może być dowolnie ustawiony, a potem zmieniany. Do wyboru jest kolor tła (Background Color) i linii wykresu (Trace Color), oraz jego długość (potrzebne w wypadku, gdy wykres po pewnym czasie biegnie tak samo). Wskazane jest jednak używanie mniejszej liczby punktów linii wykresu (Size of the Trace Buffer), gdyż zajmuje to mniej pamięci operacyjnej systemu, a samo wyświetlanie przebiega sprawniej. Użycie dłuższej linii wykresu wiąże się z tym, że każdy wyświetlany punkt ma pewną określoną, stałą pojemność, jaką zajmuje w pamięci komputera; im jest ich więcej, tym więcej pamięci operacyjnej jest potrzebne do funkcjonowania programu

### Wygląd wykresu (Style)

Opcjonalnie można wybrać dodatkowe cechy wyświetlanego wykresu. Są to: wartości na osi X (X numbers), znaczniki skali na osi X (X ticks) (standardowo obydwa wyłączone), wartości na osi Y (Y numbers), znaczniki skali na osi Y (Y ticks) oraz obwódka zewnętrzna.

Dodatkowo można ustawiać zakresy analizowanych przedziałów sygnałów podłączonych do poszczególnych osi. Pozwala to na stworzenie nie tylko wykresu o dowolnej wielkości, ale i skali. W zależności od tego, czy chcemy obserwować cały zakres analizowanego sygnału, czy też dokładniej, ale tylko jego część, można ustawić zakres dla osi X i osi Y. Dolną granicę „From” i górną „to” wpisuje się z klawiatury.



### Źródło sygnału (X / Y axis from)

W polu tym znajdują się dwa okna dialogowe, które informuje o wybranym źródle sygnału (jeśli nie zostało jeszcze wybrane, pola te są puste). Przyciskiem „Change...” wywołuje się okno o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Na wykresie będzie wyświetlana dwuwymiarowa zależność między wybranymi dwoma sygnałami.

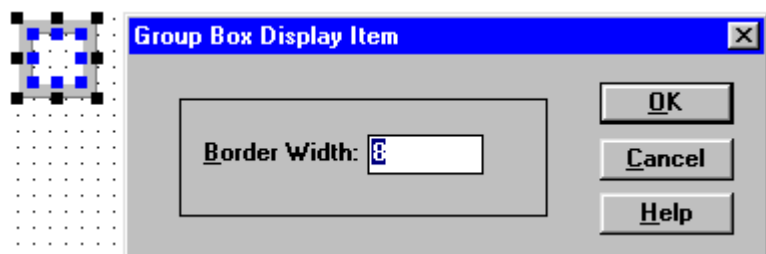
### Częstotliwość skanowania (Update Rate)

W polu tym należy ustawić dzielnik, przez który zostanie podzielona standardowa częstotliwość. Np. jeśli jest ona równa 100 Hz a potrzebna jest 20 Hz, należy w polu tym wpisać lub ustawić za pomocą strzałek liczbę 5. Powoduje to, że tylko jedna z pięciu próbek jest wyświetlana, pozostałe są ignorowane. Funkcja ta jest przydatna w przypadku migotania zdarzającego się przy wysokiej częstotliwości próbkowania.



## Obramowanie grup wyświetlaczy (Group Box Display Item)

Ramka może być rysowana wokół wybranej grupy wyświetlaczy, aby zaznaczyć np. ich wspólne źródło sygnału, lub wokół jednego, aby go wyróżnić. Może ona być dowolnie formowana, również jej wewnętrzna i zewnętrzna krawędź, nawet niezależnie od siebie. Może ona mieć jedynie szary kolor.

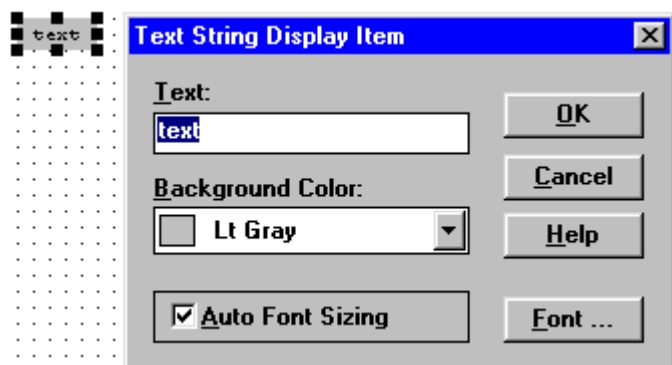


### Szerokość ramki (Border Width)

Pole to służy do ustawiania grubości ramki (odległości pomiędzy jej krawędziami). Po wpisaniu wartości do okna dialogowego wszystkie boki ramki wyrównują się do ustawionej wartości, jednak potem można je dowolnie zmieniać. W pole to można wpisać liczbę od 1 do 255.

## Statyczny wyświetlacz tekstu (Text String Display Item)

Wyświetlacz ten może być używany tylko jako stały moduł na stronie wyświetleń. Nie posiada on również nazwy systemowej, aby można było podłączyć do niego źródło sygnału. Można wybrać czcionkę, kolor, rozmiar oraz właściwości wyświetlanego tekstu po użyciu przycisku „Font...”. Są tam dostępne wszystkie znajdujące się w komputerze czcionki, 16 kolorów, 16 rozmiarów oraz styl (zwykły, pogrubiony, pochyły lub oba naraz) i efekty takie, jak przekreślenie lub podkreślenie.



### Pole edycyjne (Text)

Tu należy wpisać żądany tekst do wyświetlenia.

### Kolor tła (Background Color)

Do wyboru jest 16 kolorów tła. W tym kolorze ukaże się całe pole tekstu. Aby ustawić kolor samego tekstu należy wejść w ustawienia czcionki (przycisk „Font...”).

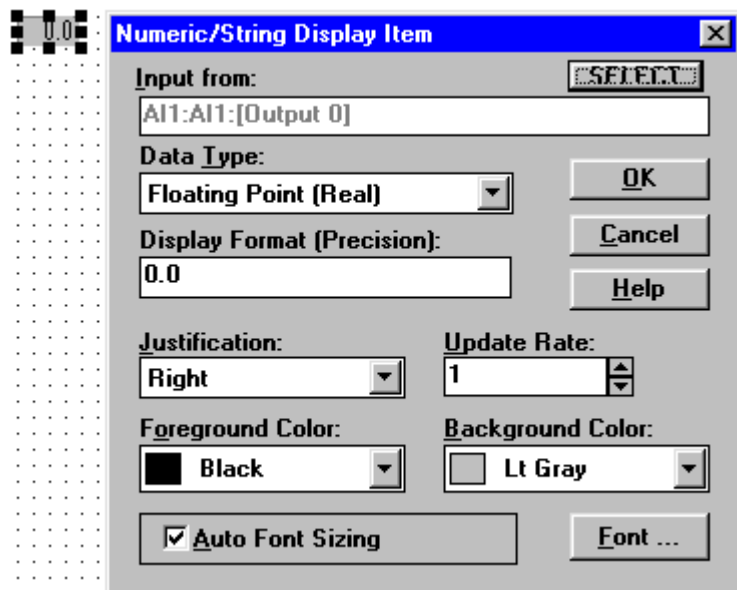
### Dostosowanie rozmiaru czcionki (Auto Font Sizing)

Jeśli ta opcja jest zaznaczona, każdy tekst zostanie, po zaakceptowaniu, automatycznie dostosowany do wielkości pola tekstowego.

## Dynamiczny wyświetlacz cyfrowy (Numeric/String Display Item)

Wyświetlacz ten pobiera sygnał z podłączonego bloku ze strony zadań. Jest on rysowany na stronie wyświetleń w dowolnym rozmiarze i proporcjach. Należy, jednak pamiętać, że wraz z podawanym sygnałem, wyświetlany ciąg znaków może ulegać zmianie (inna szerokość tekstu, minus przed liczbą),

dlatego trzeba tę ewentualność uwzględnić przy wyborze długości pola wyświetlacza. W razie niewystarczającej, można ją dowolnie zmieniać (tylko, jeśli aplikacja nie jest uruchomiona). Zmieniać można również dowolnie wysokość i położenie wyświetlacza.



#### Źródło sygnału (Input from)

Pole to informuje, z którego bloku jest pobierany sygnał do wyświetlenia. Jeżeli żaden nie został wybrany, pole jest puste. Przyciskiem „Select” wywołuje się okno o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Sygnał zostanie zinterpretowany i wyświetlony jako liczba lub ciąg znaków.

#### Typ danych (Data Type)

W tym polu należy wybrać typ danych, w jakim będą one dostarczane do wyświetlacza. Do wyboru są: liczba rzeczywista (Floating point (Real)), całkowita (Integer) lub ciąg znaków (String). Dodatkowo dla liczby rzeczywistej można ustawić dokładność, z jaką będzie odczytywana przez wyświetlacz. Standardowo ustawiona jest na dokładność do dziesiątych części (0.0). Aby ją zwiększyć należy wpisać w pole „Display Format (Precision)” 0.00, 0.000, itd., w zależności od żądanej ilości miejsc po przecinku.

#### Umiejscowienie wyświetlenia (Justification)

Liczba lub ciąg znaków może być automatycznie wyrównywana w zależności od ustawienia w tym polu do prawej strony (Right), do lewej (Left) lub centrowana (Center).

#### Kolor czcionki / kolor tła (Foreground Color / Background Color)

Do wyboru jest 16 kolorów dla czcionki i tła. Należy je dobrać tak, aby ze sobą kontrastowały, a wyświetlenie był czytelne. Dodatkowych ustawień czcionki można dokonać używając przycisku „Font...”. Jest tam dostępny wybór rodzaju, stylu i rozmiaru czcionki.

#### Dostosowanie rozmiaru czcionki (Auto Font Sizing)

Jeśli ta opcja jest zaznaczona, każde wyświetlenie zostanie, po zaakceptowaniu, automatycznie dostosowane do wielkości wyświetlacza.

#### Częstotliwość skanowania (Update Rate)

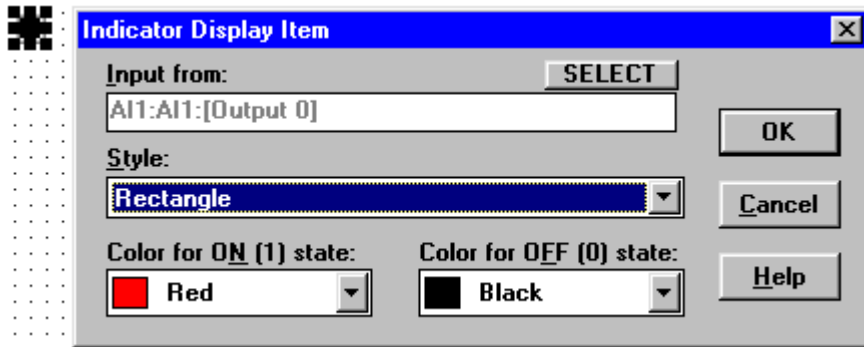
W polu tym należy ustawić dzielnik, przez który zostanie podzielona standardowa częstotliwość. Np. jeśli jest ona równa 100 Hz a potrzebna jest 20 Hz, należy w polu tym wpisać lub ustawić za pomocą strzałek liczbę 5. Powoduje to, że tylko jedna z pięciu próbek jest wyświetlana, pozostałe są ignorowane. Funkcja ta jest przydatna w przypadku migotania zdarzającego się przy wysokiej częstotliwości próbkowania.

## Kontrolka (Indicator Display Item)

Moduł ten służy do sygnalizowania stanu wyjścia bloku cyfrowego. Cyfrowy sygnał równy „1” włącza (On) kontrolkę, natomiast równy „0” wyłącza ją.

### Źródło sygnału (Input from)

Pole to informuje, z którego bloku jest pobierany sygnał do wyświetlenia. Jeżeli żaden nie został wybrany, pole jest puste. Przyciskiem „Select” wywołuje się okno o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Doprowadzony sygnał powoduje dostosowanie do jego aktualnej wartości koloru kontrolki.



### Wygląd kontrolki (Style)

Do wyboru jest prostokąt o wybranym kolorze otoczony ramką (Rectangle) lub elipsa (Round or Ellipse) wpisana w prostokąt (kolor zmienia elipsa).

### Kolor kontrolki w położeniu włączonym i wyłączonym (Color for ON (1) / OFF (0) state)

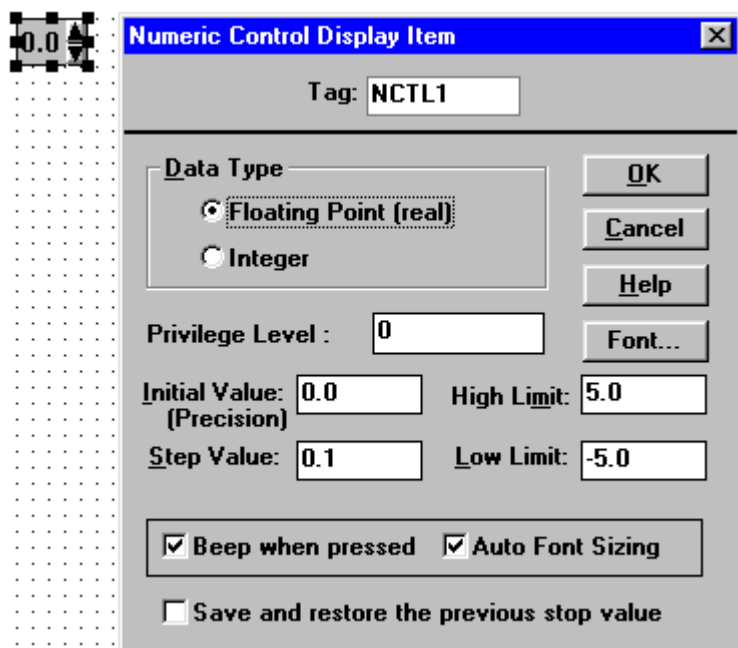
Do wyboru jest 16 kolorów dla poszczególnych położenia kontrolki. Należy uważać, aby nie ustawić ich jednakowych, bo zmiana sygnału źródłowego zostanie zasygnalizowana, ale nie będzie widoczna.

## Potencjometr cyfrowy (Numeric Control Display Item)

Służy on do kontroli bloków strony zadań za pomocą generowanego w tym module sygnału. Posiada on strzałki do zmiany wartości generowanego sygnału oraz wyświetlacz, na którym jest widoczna aktualna jego wartość. Wielkość potencjometru może być dowolna, jak również po narysowaniu dowolnie zmieniana. Potencjometr ten jest używany jako moduł generujący nastawny sygnał obsługiwany za pomocą myszy lub klawiatury. Sygnał ten jest dalej przekazywany do strony zadań do podłączonego bloku. W ten sposób można ręcznie kontrolować wejścia bloków i sterować procesem. Dane mogą zostać ustawione w formacie liczby rzeczywistej (real) z dodatkowo nastawną dokładnością (ilością miejsc po przecinku), lub liczby całkowitej (integer). Wielkość i rodzaj czcionki może być również ustawiony przy użyciu przycisku „Font...” w oknie dialogowym.

### Typ danych (Data Type)

W tym polu można wybrać typ, w jakim dane będą podawane na wyświetlaczu. Do wyboru jest format liczby rzeczywistej (real) lub całkowitej (integer).



#### Zapis ostatniego położenia potencjometru (Save and restore the previous stop value)

Po zaznaczeniu tej funkcji ostatnie położenie potencjometru zostanie zapamiętane i po ponownym uruchomieniu aplikacji jego położenie będzie takie samo, jak przed jej zamknięciem. Należy zwrócić jednak uwagę, że funkcja ta będzie działać tylko, jeżeli komputer zostanie wyłączony prawidłowo. Jeżeli zostanie użyta kombinacja klawiszy „Ctrl + Alt + Del” lub zostanie on wyłączony zanim zostanie zamknięta aplikacja, ostatnie położenie potencjometru nie zostanie zapamiętane.

#### Stopień uprzywilejowania (Privilege level)

Ta funkcja jest używana do ochrony systemu kontroli przed nieuprzywilejowanymi użytkownikami. Może on przyjmować wartości od 0 do 255, z tym, że im większa liczba, tym wyższy stopień. Na przykład, jeśli jest on ustawiony na 100, to stopień uprzywilejowania użytkownika musi być większy lub równy 100, aby mógł użyć tego potencjometru.

#### Wartość początkowa (Initial Value)

W tym polu należy wpisać wartość położenia, jaką ma mieć potencjometr w chwili uruchomienia aplikacji. Po każdym jej uruchomieniu będzie on przyjmował tę wartość, o ile nie została uruchomiona funkcja zapisu ostatniego położenia potencjometru.

#### Wartość skoku (Step Value)

Pole to jest używane do nastawiania skoku, z jakim zmieni się wartość położenia potencjometru po jednokrotnym kliknięciu na jedną ze strzałek.

#### Górna / dolna granica (High limit / Low limit)

W polach tych można określić zakres, w jakim będzie możliwa zmiana położenia potencjometru.

### **Potencjometr obrotowy (Knob Control Display Item)**

Służy on do kontroli bloków strony zadań za pomocą generowanego w tym module sygnału. Posiada on gałkę potencjometryczną z zaznaczonymi wkoło wartościami oraz wyświetlacz podający aktualną wartość położenia potencjometru. Wielkość potencjometru może być dowolna, jak również po narysowaniu dowolnie zmieniana. Potencjometr ten jest używany jako moduł generujący nastawny sygnał obsługiwany za pomocą myszy lub klawiatury. Sygnał ten jest dalej przekazywany do strony zadań do podłączonego bloku. W ten sposób można ręcznie kontrolować wejścia bloków i sterować procesem. Dane mogą zostać ustawione w formacie liczby rzeczywistej (real) z dodatkowo nastawną dokładnością (ilością miejsc po przecinku), lub liczby całkowitej (integer). Pokrętko może być również przemieszczane za pomocą strzałek na klawiaturze, o ile focus jest aktualnie na nim („focus” standardowe określenie Windows określa, który z modułów może być w danej chwili kontrolowany za pomocą klawiatury). Jeżeli nie jest, można to zmienić

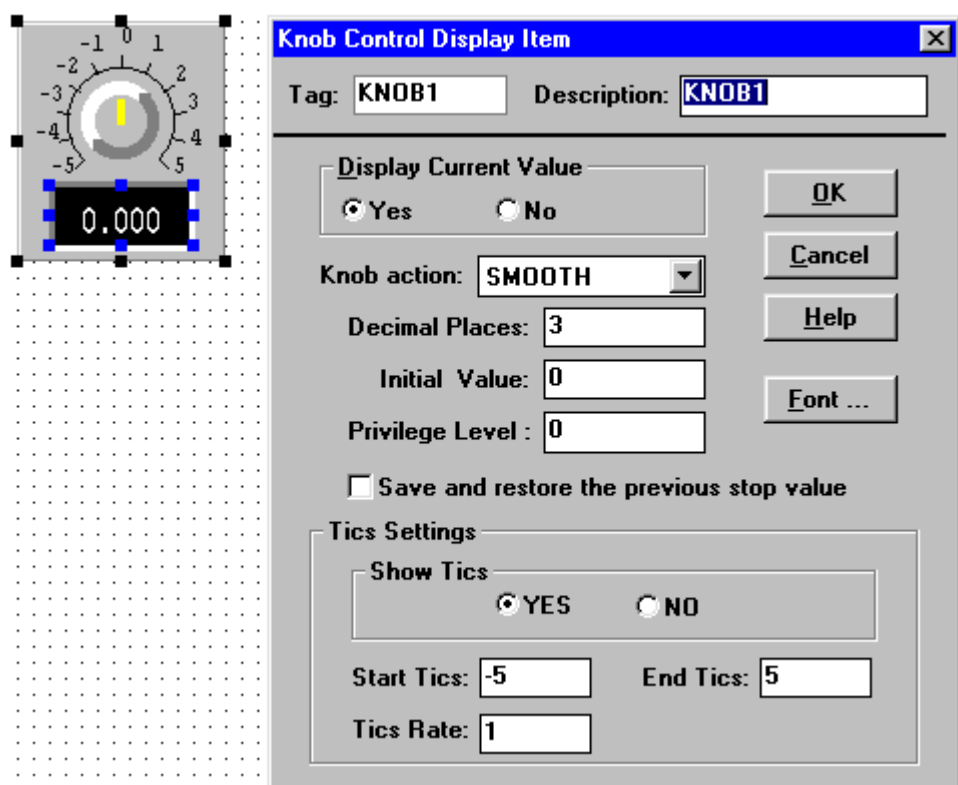
używając klawisza tabulatora „Tab”. Jego użycie powoduje zmianę focusa po kolei na każdy z umożliwiających kontrolę przez klawiaturę modułów. Sygnał uzyskiwany z tego potencjometru można wysyłać w formacie liczby rzeczywistej (floating point) z nastawną dokładnością (ilość miejsc po przecinku). Rozmiar i rodzaj czcionki wyświetlanej wartości położenia potencjometru może być dowolnie ustawiany po użyciu przycisku „Font...” w jego oknie dialogowym.

#### Wyświetlanie wartości położenia potencjometru (Display Current Value)

Można tu wybrać, czy wyświetlacz numeryczny umieszczony pod gałką potencjometru ma być aktywny (widoczny), czy nie.

#### Sposób działania potencjometru (Knob Action)

Do wyboru są dwa tryby jego działania: płynny (Smooth) – wartość zmienia się z dokładnością odczytu wyświetlacza, oraz skokowy (Incremental) – wartość zmienia się skokowo w nastawnych odstępach.



#### Dokładność odczytu (Decimal places)

Wpisana liczba oznacza ilość miejsc po przecinku wyświetlanej liczby.

#### Wartość początkowa (Initial Value)

W tym polu należy wpisać wartość położenia, jaką ma mieć potencjometr w chwili uruchomienia aplikacji. Po każdym jej uruchomieniu będzie on przyjmował tę wartość, o ile nie została uruchomiona funkcja zapisu ostatniego położenia potencjometru.

#### Zapis ostatniego położenia potencjometru (Save and restore the previous stop value)

Po zaznaczeniu tej funkcji ostatnie położenie potencjometru zostanie zapamiętane i po ponownym uruchomieniu aplikacji jego położenie będzie takie samo, jak przed jej zamknięciem. Należy zwrócić jednak uwagę, że funkcja ta będzie działać tylko, jeżeli komputer zostanie wyłączony prawidłowo. Jeżeli zostanie użyta kombinacja klawiszy „Ctrl + Alt + Del” lub zostanie on wyłączony zanim zostanie zamknięta aplikacja, ostatnie położenie potencjometru nie zostanie zapamiętane.

#### Stopień uprzywilejowania (Privilege level)

Ta funkcja jest używana do ochrony systemu kontroli przed nieuprzywilejowanymi użytkownikami. Może on przyjmować wartości od 0 do 255, z tym, że im większa liczba, tym wyższy stopień. Na przykład, jeśli

jest on ustawiony na 100, to stopień uprzywilejowania użytkownika musi być większy lub równy 100, aby mógł użyć tego potencjometru.

#### Pokazywanie znaczników (Show Tics)

Opcja pozwalająca na wyświetlanie lub ukrycie znaczników wokół potencjometru.

#### Początek skali (Start Ticks)

Określa dolną granicę zakresu regulacji za pomocą tego potencjometru.

#### Koniec skali (End Tics)

Określa górną granicę zakresu regulacji za pomocą tego potencjometru.

#### Odstępy między znacznikami (Ticks Rate)

Wartość w tym polu oznacza odstępy między nimi. Należy sprawdzić, czy nie są one zbyt małe, bo może to spowodować ich nieczytelność.

### **Wskaźnik zakresowy (Anameter (Analog Meter) Display Item)**

Wyświetla on dane z wybranego bloku strony zadań w formie miernika obrotowego z podzakresami o nastawnej rozpiętości zakresów. Można również ustawić ilość i kolor segmentów. Przedtem, jednak należy wybrać źródło sygnału dla miernika.

#### Źródło sygnału (Input from)

Pole to informuje, z którego bloku jest pobierany sygnał do wyświetlenia. Jeżeli żaden nie został wybrany, pole jest puste. Przyciskiem „Select” wywołuje się okno o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Następnie należy zatwierdzić wybór a nazwa bloku zostanie wyświetlona w oknie „Input from”.

#### Znaczniki (Show Tics)

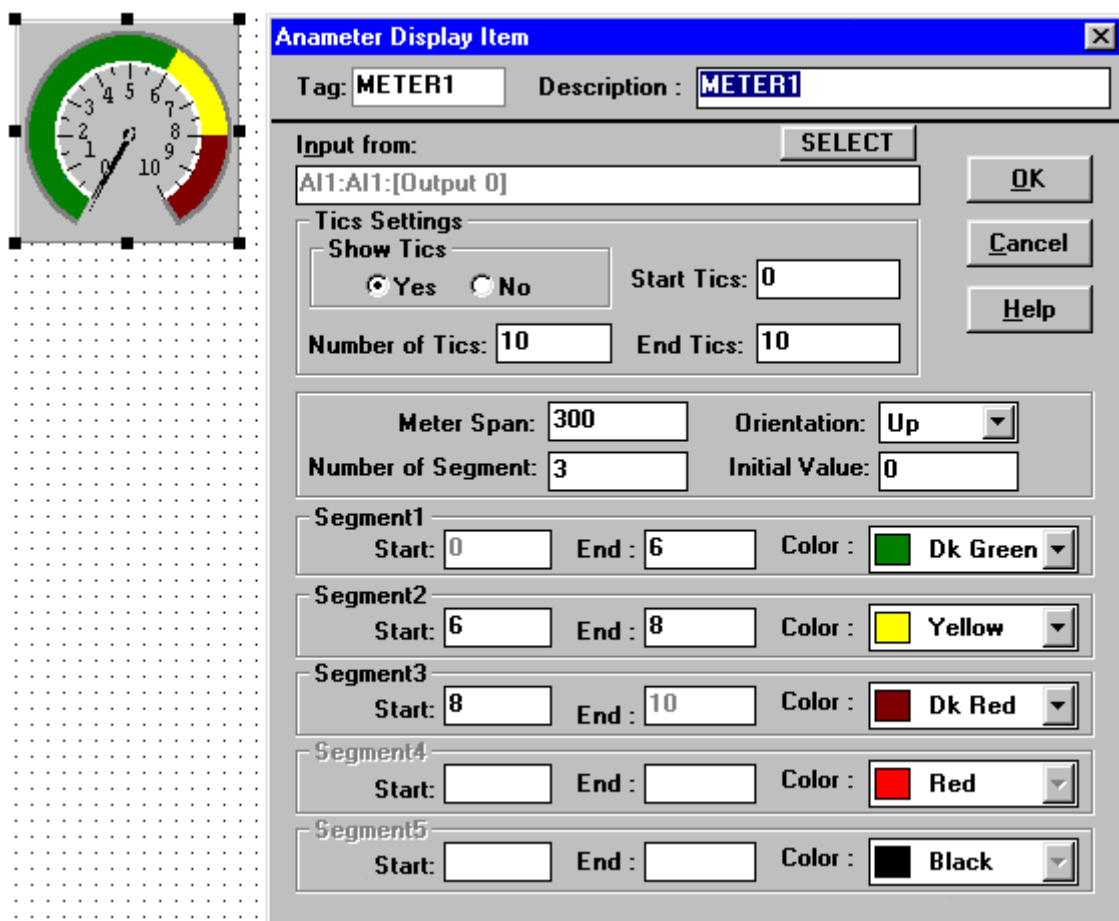
Należy wybrać, czy znaczniki mają być aktywne (widoczne), czy też nie.

#### Dolny zakres miernika (Start Tics)

W polu należy wpisać wartość dolnej granicy analizy miernika.

#### Górny zakres miernika (End Tics)

W polu należy wpisać wartość górnej granicy analizy miernika.



#### Ilość znaczników (Number of Tics)

W polu należy wpisać całkowitą ilość znaczników.

#### Rozpiętość miernika (Meter Span)

W polu należy wpisać rozpiętość całkowitą miernika w stopniach, tzn. w kącie o jakiej rozpiętości ma się zmieścić cały zakres miernika.

#### Orientacja (Orientation)

Pole wyboru. Można tu wybrać, w którą stronę, tzn. w górę (Up), w dół (Down), w prawo (Right), czy w lewo (Left) ma być skierowana część miernika odpowiadająca środkowi jego zakresu.

#### Ilość segmentów (Number Of Segments)

W pole należy wpisać ilość segmentów, jaka ma się znaleźć w zakresie pomiarowym miernika. Moduł pozwala na zastosowanie od 1 do 5 segmentów.

#### Wartość początkowa (Initial Value)

W pole należy wpisać wartość, na której ma się znaleźć igła miernika na początku pomiaru.

#### Zakresy segmentów

Można tu wybrać wartość początkową (Start) i końcową (End) każdego z zakresów, z tym, że wartość końcowa jednego zakresu jest przyjmowana automatycznie jako początkowa dla zakresu następnego, oraz kolor dla każdego z nich. Zalecane są kolory zwykle stosowane do sygnalizacji pracy procesu, np. kolor zielony dla normalnego trybu pracy, czy czerwony dla trybu alarmowego, itd.

### **Potencjometr przesuwany (Slider Control Display Item)**

Służy on do kontroli bloków strony zadań za pomocą generowanego w tym module sygnału. Posiada on suwak potencjometryczny z zaznaczonymi obok wartościami. Wielkość potencjometru może być dowolna, jak również po narysowaniu dowolnie zmieniana. Potencjometr ten jest używany jako moduł generujący

nastawny sygnał obsługiwany za pomocą myszy lub klawiatury. Sygnał ten jest dalej przekazywany do strony zadań do podłączonego bloku. Jest to sygnał zawsze w formacie liczby rzeczywistej (Floating point). W ten sposób można ręcznie kontrolować wejścia bloków i sterować procesem. Suwak może być również przemieszczany za pomocą strzałek na klawiaturze, o ile focus jest aktualnie na nim („focus” standardowe określenie Windows określa, który z modułów może być w danej chwili kontrolowany za pomocą klawiatury). Jeżeli nie jest, można to zmienić używając klawisza tabulatora „Tab”. Jego użycie powoduje zmianę focusa po kolei na każdy z umożliwiających kontrolę przez klawiaturę modułów.

#### Sposób działania potencjometru (Slider Action)

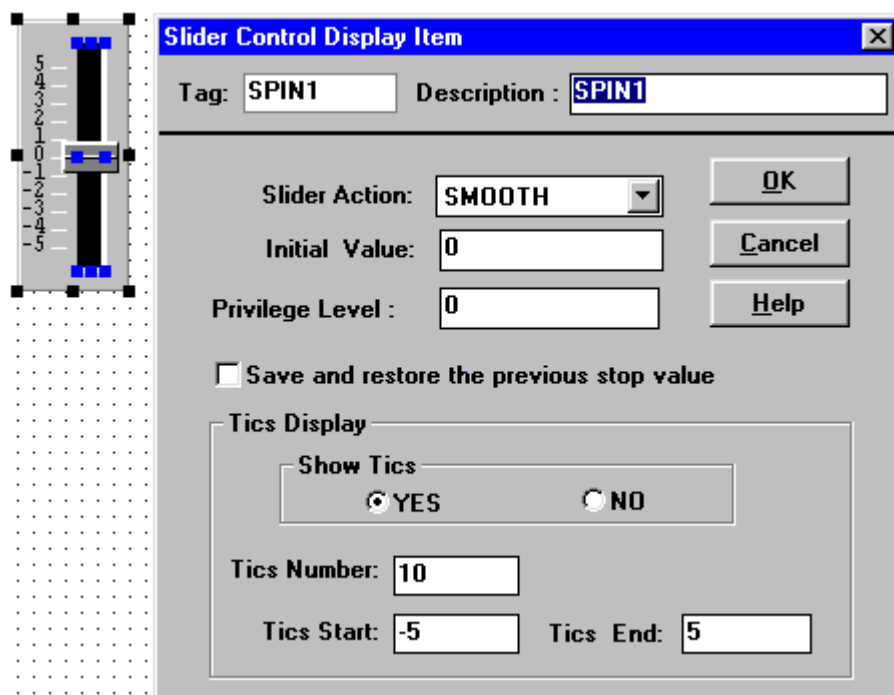
Do wyboru są dwa tryby jego działania: płynny (Smooth) – wartość zmienia się płynnie, oraz skokowy (Incremental) – wartość zmienia się skokowo w nastawnych odstępach.

#### Wartość początkowa (Initial Value)

W tym polu należy wpisać wartość położenia, jaką ma mieć potencjometr w chwili uruchomienia aplikacji. Po każdym jej uruchomieniu będzie on przyjmował tę wartość, o ile nie została uruchomiona funkcja zapisu ostatniego położenia potencjometru.

#### Zapis ostatniego położenia potencjometru (Save and restore the previous stop value)

Po zaznaczeniu tej funkcji ostatnie położenie potencjometru zostanie zapamiętane i po ponownym uruchomieniu aplikacji jego położenie będzie takie samo, jak przed jej zamknięciem. Należy zwrócić jednak uwagę, że funkcja ta będzie działać tylko, jeżeli komputer zostanie wyłączony prawidłowo. Jeżeli zostanie użyta kombinacja klawiszy „Ctrl + Alt + Del” lub zostanie on wyłączony zanim zostanie zamknięta aplikacja, ostatnie położenie potencjometru nie zostanie zapamiętane.



#### Stopień uprzywilejowania (Privilege level)

Ta funkcja jest używana do ochrony systemu kontroli przed nieuprzywilejowanymi użytkownikami. Może on przyjmować wartości od 0 do 255, z tym, że im większa liczba, tym wyższy stopień. Na przykład, jeśli jest on ustawiony na 100, to stopień uprzywilejowania użytkownika musi być większy lub równy 100, aby mógł użyć tego potencjometru.

#### Pokazywanie znaczników (Show Tics)

Opcja pozwalająca na wyświetlanie lub ukrycie znaczników obok potencjometru.

#### Początek skali (Start Ticks)

Określa dolną granicę zakresu regulacji za pomocą tego potencjometru.

#### Koniec skali (End Tics)



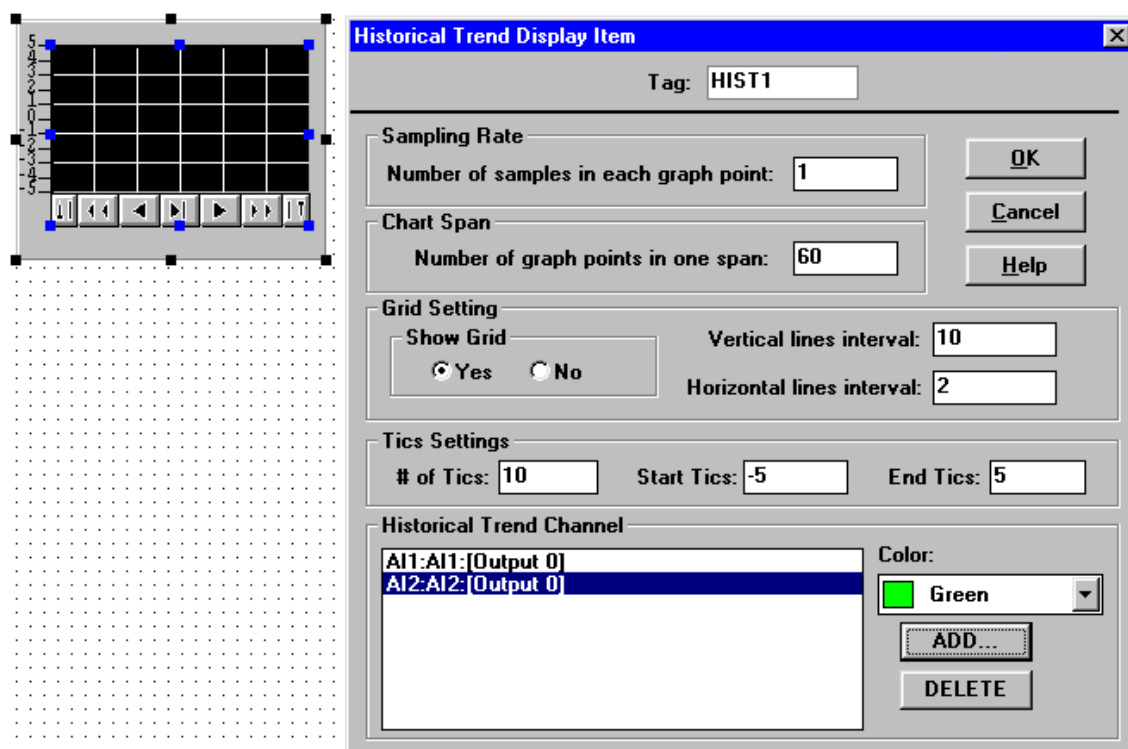
Określa górną granicę zakresu regulacji za pomocą tego potencjometru.

#### Ilość znaczników (Tics Number)

Wartość w tym polu oznacza całkowitą ilość znaczników równomiernie rozmieszczonych w zakresie potencjometru.

### **Wykres rejestrujący (Historical Trending Display Item)**

Wykres ten ilustruje przebieg wartości analizowanego sygnału w czasie, z tą jednak różnicą, że istnieje tu możliwość skokowego analizowania tego przebiegu na nieruchomych częściach wykresu. Rozmiar i wygląd wykresu mogą być dowolnie ustawiane, a potem również zmieniane. Wykres ten jest zdolny wyświetlać do ośmiu linii ilustrujących zmienność różnych sygnałów w czasie. Wartość sygnału jest odkładana na osi Y, zaś czas na osi X.



#### Kanały źródłowe (Historical Trending Channel)

W tym polu znajdują się dwa okna dialogowe. Jedno wyświetla wybrane źródła, natomiast drugie przyporządkowuje każdemu z nich kolor linii wykresu (Trace Color). Przyciskiem „Add” wywołuje się okno o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Wszystkie nazwy wybranych bloków zostaną wyświetlone w tym oknie. Sygnały te będą dynamicznie interpretowane i wyświetlane przy pomocy linii wykresu, których kolor należy wybrać dla każdej z nich. Aktualnie jest ustawiany kolor dla sygnału z bloku o podświetlonej w oknie listy połączeń nazwie.

#### Częstotliwość analizowania (Number of Samples for Each Graph Point)

Jest to dzielnik podobny do tego, który w innych blokach jest nazywany „Update Rate”. Liczba wpisana w tym polu informuje, na ile próbek pobranych przez program jedna ma zostać wyświetlona, np. jeśli zostanie wprowadzona liczba 5, to jedna zostanie wyświetlona, natomiast cztery próbki, mimo, że zostały zapisane wyświetlone nie będą.

#### Ilość punktów każdej linii wykresu (Number of Graph Points in One Span)

Liczba w tym polu oznacza ilość jednocześnie wyświetlanych punktów.

### Siatka (Show Grid)

Należy zaznaczyć, czy na wykresie ma znajdować się siatka, czy też nie.

### Odstępy między liniami siatki (Vertical/Horizontal lines interval)

Liczby w oknach oznaczają, ile znaczników ma się znaleźć pomiędzy sąsiadującymi liniami siatki dla osi pionowej (Vertical) i poziomej (Horizontal).

### Liczba znaczników (# of Tics)

Liczba w oknie oznacza całkowitą ilość znaczników na osi Y.

### Początek skali (Start Tics)

Liczba w oknie oznacza dolną granicę osi Y.

### Koniec skali (End Tics)

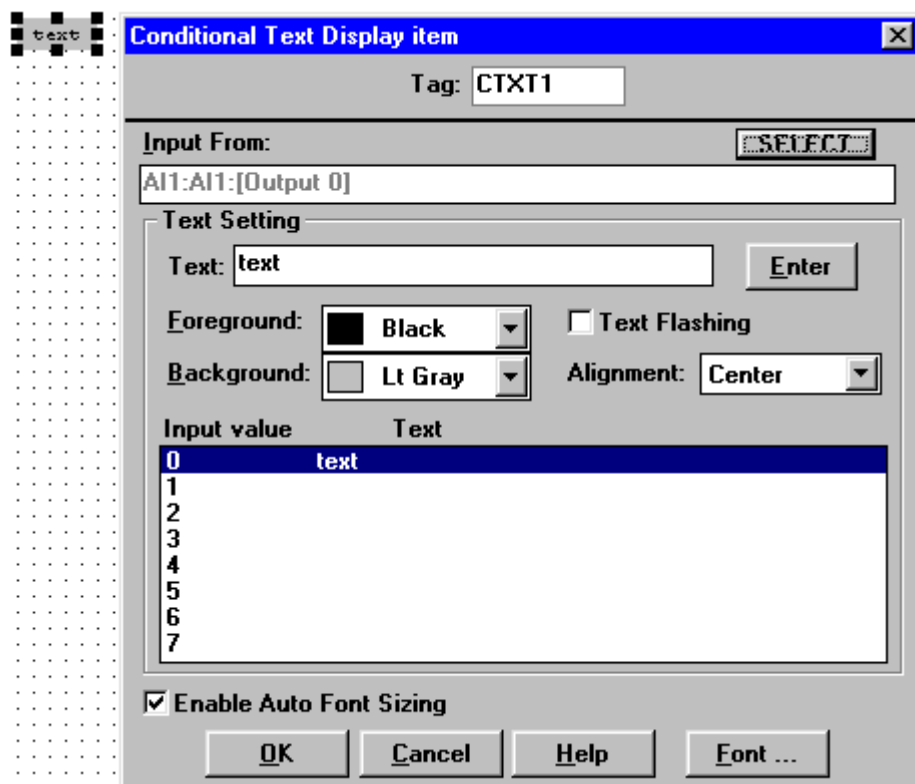
Liczba w oknie oznacza górną granicę osi Y.

### Kanały źródłowe (Historical Trend Channel)

Pole to jest używane do dodawania nazw bloków źródłowych w celu wyświetlania wartości pobieranych od nich sygnałów. Można tu dodać nowy (Add...), lub usunąć (Delete) istniejący już w spisie blok. Dodatkowo można wyszczególnić kolor linii wykresu dla każdego ze źródeł sygnału.

## **Dynamiczny wyświetlacz tekstu (Conditional Text Display Item)**

Ten moduł posiada wejścia i wyjścia. Przyjmuje on wartości z przedziału od 0 do 7, które to z kolei sterują tekstem, jaki ma być wyświetlany. Dodatkowo, dane z tego modułu w postaci ciągu znaków (String) mogą być podawane do innego modułu przyjmującego tego typu dane, tak, jak blok RS-232. Po dwukrotnym kliknięciu na wyświetlacz mamy do dyspozycji okno dialogowe modułu z przyporządkowanymi danej



wartości sygnału wejściowego tekstami.

### Źródło sygnału (Input From)

Przyciskiem „Select” wywołuje się okno o nazwie „Connection”, w którym należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego

będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Wartość sygnału wejściowego będzie traktowana jako polecenie wyboru wyświetlanego tekstu.

#### Wyświetlany tekst (Text)

W polu tym należy wpisać tekst, jaki ma się pojawić przy doprowadzonym sygnale o wartości podanej w oknie „Input Value, Text” (aktualnie podświetlonej). Gotowy tekst zatwierdza się przyciskiem „Enter” w oknie dialogowym.

#### Kolor tekstu/tła (Text/Background Color)

Do wyboru jest 16 kolorów, spośród których można wybrać kolor dla tekstu i tła dla każdego z wyświetleń.

#### Umiejscowienie wyświetlenia (Alignment)

Do wyboru są trzy ustawienia: tekst wyświetla się od lewej, od prawej strony, lub jest centrowany. Ustawień takich dokonuje się dla każdego z wyświetleń w ramach tego modułu.

#### Pulsacja tekstu (Text Flashing)

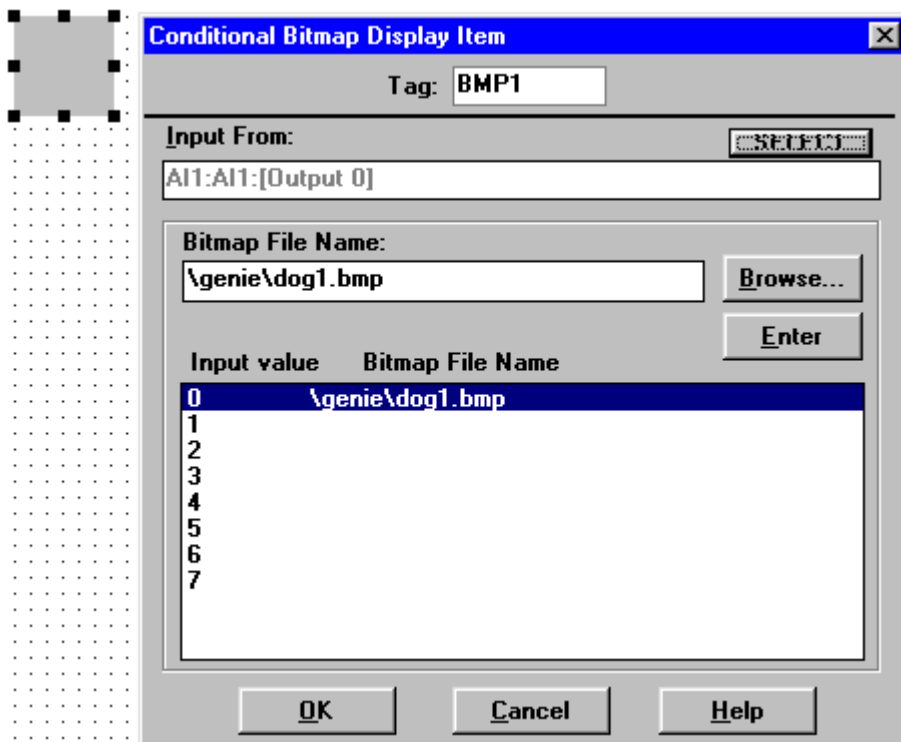
Opcja powodująca cykliczną zmianę koloru tła i tekstu wybranego wyświetlenia.

#### Czcionka (Font...)

Ustawień rodzaju czcionki, jej stylu i rozmiaru dokonuje się po użyciu przycisku „Font...”

### **Dynamiczny wyświetlacz bitmap (Conditional Bitmap Display Item)**

Wyświetlacz ten posiada jedynie zdolność wejściową. Akceptuje on tylko sygnały o wartościach całkowitych z zakresu od 0 do 7, przyporządkowując każdej z nich adres pliku z zawartością bitmapy, z którego ma być pozyskany obraz do wyświetlenia. Jest on zmieniany dynamicznie w zależności od zmieniającej się wartości wejściowej. Sygnał pobierany jest z bloku na stronie zadań, lub z modułu sygnałowego na stronie wyświetleń.



#### Źródło sygnału (Input From)

Za pomocą przycisku „Select” otworzyć należy okno dialogowe, w który należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego

będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Wartość sygnału wejściowego będzie traktowana jako polecenie wyboru wyświetlanej bitmapy.

#### Wybór pliku z bitmapą (Bitmap File Name)

Po użyciu przycisku „Browse...” wywołane zostaje okno dialogowe, ew którym wyświetlone są wszystkie aktualnie zainstalowane pliki o rozszerzeniu „\*.bmp”. Można stamtąd wybrać dla każdej z wartości sygnału wejściowego po jednym pliku, którego zawartość ma być wyświetlana. Po zatwierdzeniu jego ścieżka dostępu zostanie umieszczona w polu „Bitmap File Name”. Wtedy należy przyciskiem „Enter” przyporządkować go wybranej wcześniej wartości sygnału wejściowego.

### Przycisk z dynamicznym wyświetlaczem (Conditional Button Display Item)

Przycisk ten posiada zarówno zdolność wejściową, jak i wyjściową. Wyjście może zostać połączone z blokiem na stronie zadań przy wykorzystaniu jego nazwy systemowej (Tag). Jest on przyciskany za pomocą myszy lub klawisza „Enter” klawiatury (jeśli focus jest aktualnie na tym module strony; „focus” standardowe określenie Windows określa, który z modułów może być w danej chwili kontrolowany za pomocą klawiatury. Focus może być przesuwany z jednego modułu na drugi za pomocą klawisza tabulacji „Tab”). Przycisk wysyła wtedy, w zależności od stanu (wciśnięty/zwolniony) sygnał cyfrowy 1 lub 0 (lub odwrotnie – do wyboru) do podłączonego bloku. Wielkość przycisku może być dowolna oraz dowolnie po narysowaniu zmieniana.

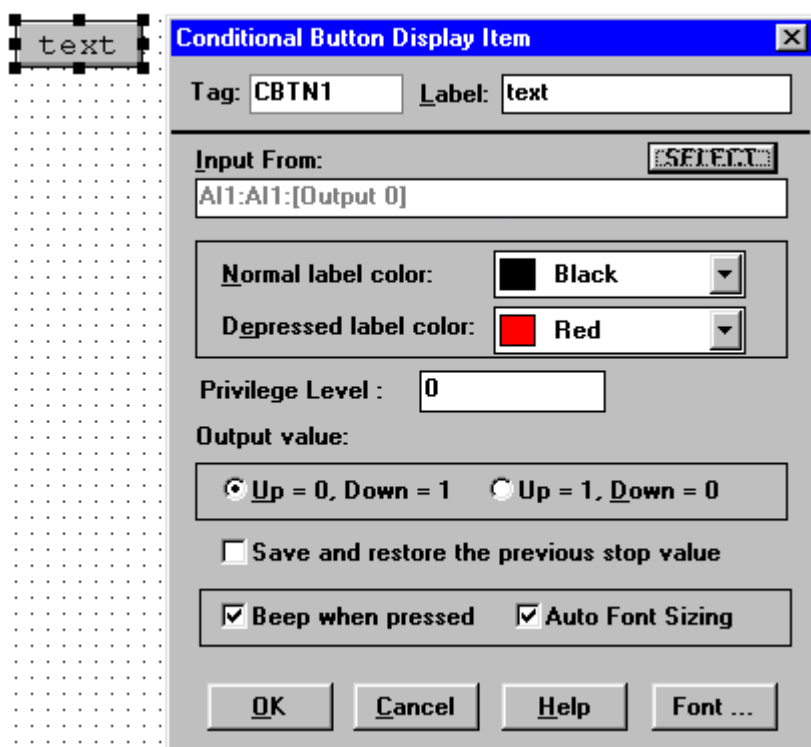
Przycisk ten różni się tym od przycisku dwupołożeniowego, że ten dodatkowo może być wciskany lub zwalniany za pomocą sygnału cyfrowego 0 lub 1 pobranego z bloku ze strony zadań.

#### Nazwa modułu (Tag)

W tym polu znajduje się nazwa systemowa (bez możliwości zmiany) modułu). Reprezentuje ona wartość wyjściową danego przycisku przy jego wyborze jako sygnał wejściowy do bloku na stronie zadań. W oknie dialogowym danego bloku należy wówczas znaleźć tę właśnie nazwę systemową, dodatkowo uzupełnioną dla ułatwienia w opis (etykietkę). Sygnał wyjściowy każdego przycisku jest dostępny dla dowolnej liczby bloków. Jest to przydatne, gdy zachodzi potrzeba kontrolowania funkcji kilku bloków jednym przyciskiem.

#### Etykietka (Label)

W tym polu należy wpisać tekst informujący o funkcji, jaką ma pełnić przycisk. Tekst ten zostanie wyświetlony na przycisku. Jego maksymalna długość to 30 znaków.



#### Źródło sygnału (Input From)

Za pomocą przycisku „Select” otworzyć należy okno dialogowe, w który należy wybrać (koniecznie po kolei) stronę (zadań, wyświetleń, wirtualną) (Task/Display/Virtual), nazwę bloku (Tag Name), z którego będzie pobierany sygnał oraz kanał tego bloku (jeśli posiada więcej, niż jeden). Wartość sygnału wejściowego będzie traktowana jako polecenie wciśnięcia lub zwolnienia przycisku.

#### Kolor etykiety (Label Color)

W tym polu można wybrać, w jakim kolorze ma pokazywać się na przycisku tekst etykiety. Można wybrać spośród 16 kolorów dla obu położań przycisku (zwolnionego „Normal label color” oraz wciśniętego „Depressed label color”).

#### Stopień uprzywilejowania (Privilege level)

Ta funkcja jest używana do ochrony systemu kontroli przed nieuprzywilejowanymi użytkownikami. Może on przyjmować wartości od 0 do 255, z tym, że im większa liczba, tym wyższy stopień. Na przykład, jeśli jest on ustawiony na 100, to stopień uprzywilejowania użytkownika musi być większy lub równy 100, aby mógł użyć tego przycisku.

#### Wartość wyjściowa (Output value)

W polu tym można wybrać, jaką wartość w zależności od położenia przycisku ma on wysyłać. Do wyboru są dwie opcje:

0 – przycisk zwolniony, 1 – wciśnięty „Up=0, Down=1” lub odwrotnie „Up=1, Down=0”

#### Zapis ostatniego położenia przycisku (Save and restore the previous stop value)

Po zaznaczeniu tej funkcji ostatnie położenie przycisku zostanie zapamiętane i po ponownym uruchomieniu aplikacji jego położenie będzie takie samo, jak przed jej zamknięciem. Należy zwrócić jednak uwagę, że funkcja ta będzie działać tylko, jeżeli komputer zostanie wyłączony prawidłowo. Jeżeli zostanie użyta kombinacja klawiszy „Ctrl + Alt + Del” lub zostanie on wyłączony zanim zostanie zamknięta aplikacja, ostatnie położenie przycisku nie zostanie zapamiętane.

#### Czcionka (Font...)

Czcionka tekstu na przycisku może być zmieniana po użyciu przycisku „Font...”. Otwiera się wtedy okno dialogowe, w którym można wybrać rodzaj, styl oraz rozmiar czcionki.

#### Funkcje opcjonalne.

Sygnal dźwiękowy (Beep when pressed). Naciśnięcie jest sygnalizowane krótkim dźwiękiem.

Dostosowanie formatu czcionki (Auto Font Sizing). Wielkość czcionki jest automatycznie dostosowywana do rozmiaru przycisku.

### **Przycisk menu (Menu Display Item)**

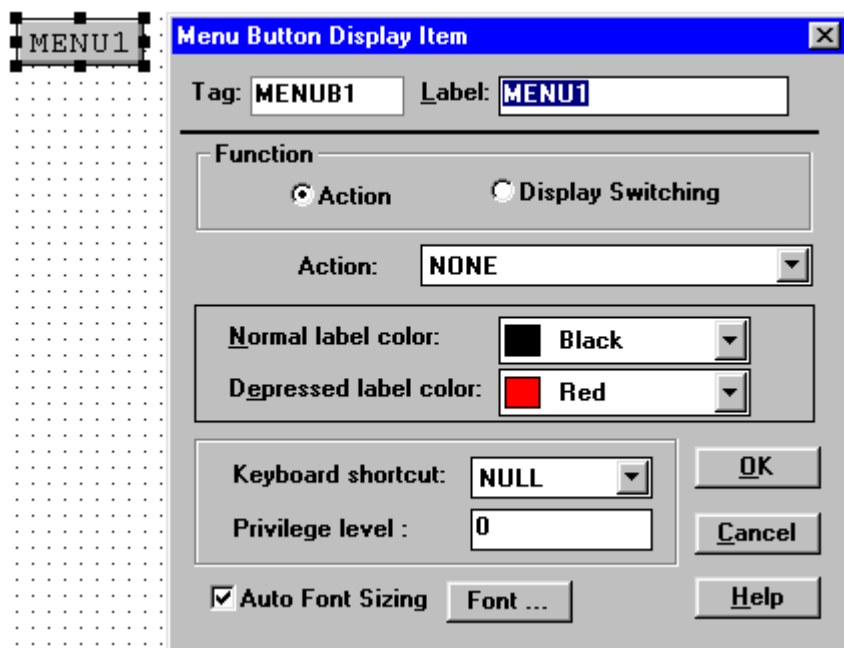
Przycisk ten posiada dwie funkcje. Jedną z nich jest bezpośredni wpływ na działanie aplikacji (Action), drugą natomiast jest możliwość przeskoku (Display Switching) z aktualnie wyświetlanej strony wyświetleń na inną (jeśli została jeszcze jakaś utworzona).

#### Sposób działania przycisku (Action)

Przy funkcji „Action” są do wyboru cztery opcje działania przycisku: uruchomienie (Start), całkowite zatrzymanie (Stop), chwilowe wstrzymanie (Pause) oraz wznowienie od momentu wstrzymania (Resume).

#### Kolor etykiety (Label Color)

W tym polu można wybrać, w jakim kolorze ma pokazywać się na przycisku tekst etykiety. Można wybrać spośród 16 kolorów dla obu położań przycisku (zwolnionego „Normal label color” oraz wciśniętego „Depressed label color”).



#### Przyporządkowanie klawiszy (Keyboard Shortcut)

Każdy przycisk może posiadać przyporządkowany mu klawisz, za pomocą którego będzie on uruchamiany. W tym polu są do wyboru klawisze funkcyjne od F2 do F8 oraz wszystkie litery od A do Z.

#### Stopień uprzywilejowania (Privilege level)

Ta funkcja jest używana do ochrony systemu kontroli przed nieuprzywilejowanymi użytkownikami. Może on przyjmować wartości od 0 do 255, z tym, że im większa liczba, tym wyższy stopień. Na przykład, jeśli jest on ustawiony na 100, to stopień uprzywilejowania użytkownika musi być większy lub równy 100, aby mógł użyć tego przycisku.

#### Czcionka (Font...)

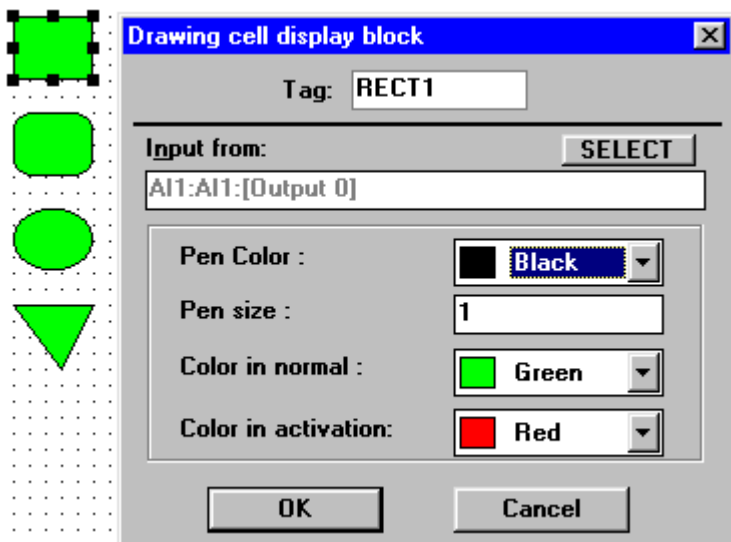
Czcionka tekstu na przycisku może być zmieniana po użyciu przycisku „Font...”. Otwiera się wtedy okno dialogowe, w którym można wybrać rodzaj, styl oraz rozmiar czcionki.

#### Dostosowanie rozmiaru czcionki (Auto Font Sizing)

Jeśli ta opcja jest zaznaczona, każde wyświetlenie zostanie, po zaakceptowaniu, automatycznie dostosowane do wielkości wyświetlacza.

### **Funkcje rysowania (Drawing cell display block)**

Ostatnie pięć ikon pozwala na łatwe rysowanie różnych kształtów geometrycznych zmieniających swój kolor w zależności od wartości sygnału wejściowego (Input from).



Kolor przy źródle aktywnym i nieaktywnym (Color in normal/in activation)

Narysowany kształt przyjmuje kolor nastawiony w oknie „Color in normal” dla sygnału wejściowego z przedziału od -1 do 1 oprócz tych wartości, natomiast kolor nastawiony w oknie „Color in activation” przy pozostałych wartościach (wliczając wartości -1 i 1).

Kolor/grubość linii krawędzi (Pen Color/size)

Do wyboru jest 16 kolorów oraz 3 grubości (od 1 do 3) linii krawędzi.