

EL-PIAST SP. Z O.O.

Macrocontrol

Instrukcja użytkownika

Ryszard Dworzecki

20.07.2016

Dokument zawiera podstawowe informacje dotyczące obsługi i działania środowiska Macrocontrol przeznaczonego do programowania, testowania i zarządzania sterownikami PLC firmy EL-Piast.

Spis treści

1.	Wprowadzenie.....	3
▪	Terminologia.....	3
2.	Instalacja oprogramowania.....	4
2.1	Minimalne wymagania sprzętowe.....	4
2.2	Wymagania oprogramowania.....	4
2.3	Instalacja.....	4
3.	Pierwsze kroki.....	5
3.1	Nowy projekt.....	5
3.2	Główne okno projektu.....	6
3.3	Tworzenie przykładowej prostej aplikacji PLC.....	7
4.	System sterowania ELP.....	10
4.1	Reprezentacje liczbowe.....	10
4.2	Zmienne.....	11
▪	Zmienne cyfrowe.....	11
▪	Zmienne analogowe.....	12
▪	Właściwości zmiennej.....	12
4.3	Wejścia / wyjścia.....	13
▪	Graficzna reprezentacja wejść / wyjść.....	14
▪	Właściwości wejść / wyjść.....	14
4.4	Połączenia.....	15
4.5	Biblioteki bloków funkcyjnych.....	16
▪	Utworzenie nowej biblioteki.....	16
▪	Dodawanie istniejącej biblioteki.....	16
▪	Usuwanie biblioteki.....	16
4.6	Bloki funkcyjne.....	16
▪	Graficzna reprezentacja.....	17
▪	Właściwości.....	18
▪	Operacje logiczne z dowolną ilością parametrów And i Or.....	18
▪	Alarmy.....	18
▪	Proste bloki funkcyjne.....	18
▪	Złożone bloki funkcyjne typu <i>Device</i>	19
▪	Złożone bloki funkcyjne typu <i>Macro</i>	19
5.	Menu główne.....	20
5.1	File.....	20
▪	New.....	20
▪	Open.....	20
▪	Save.....	20
▪	Save as.....	20
▪	Exit.....	20
5.2	Edit.....	21
▪	Undo.....	21
▪	Redo.....	21
▪	Cut.....	21
▪	Copy.....	21
▪	Paste.....	21
▪	Delete.....	21
▪	Select All.....	21

▪	Find.....	21
▪	Collision detection.....	21
▪	Zoom on scroll / Zoom on Ctrl+scroll.....	21
▪	Unlock FBD edit.....	21
5.3	View.....	22
▪	Show connection priority.....	22
▪	Settings.....	22
5.4	Commands.....	23
5.4.1	Create device.....	23
▪	Convert to device.....	23
▪	Break device.....	23
▪	Convert to macro.....	23
▪	Break macro.....	23
▪	Copy descriptions to IOs.....	23
▪	Optimize connection priority.....	23
5.5	Project.....	24
▪	Do it All.....	24
▪	Compile.....	24
▪	CMD compilation (Win8 troubleshooting).....	24
▪	Upload All.....	24
▪	Upload Variables and HMI menu.....	24
▪	Restore Variables After Upload.....	24
▪	Software Info.....	24
▪	Communication Settings.....	25
5.6	Tools.....	27
▪	HMI Editor.....	27
▪	Main Pages Editor.....	31
▪	BACnet Object Editor.....	40
▪	BACnet Integrator.....	40
▪	Virtual HMI.....	40
▪	In-Target Test.....	41
6.	Przywrócenie sterownika.....	42

1. Wprowadzenie

Macrocontrol jest zestawem narzędzi służących do programowania i testowania algorytmów sterowania dla sterowników PLC firmy **EL-Piast** z serii **ELP**. Środowisko wykorzystuje zewnętrzne darmowe narzędzia których wcześniejsza instalacja, w zależności od modelu sterownika, jest wymagana do poprawnej pracy. Wraz z powstawaniem nowych modeli sterowników PLC firmy **EL-Piast** środowisko *Macrocontrol* jest rozwijane o obsługę nowych urządzeń, a także poszerzane o nowe narzędzia ułatwiające pracę.

▪ Terminologia

- **Aplikacja PLC** – algorytm sterowania złożony z połączonych ze sobą bloków funkcyjnych zbudowany w środowisku *Macrocontrol*.
- **Blok funkcyjny / obiekt** - bloczek posiadający pewną funkcjonalność z których buduje się aplikację PLC. Bloki mogą być kilku typów np. *Device*, *Macro*, *Variable*.
- **Device** – złożony blok funkcyjny zawierający wewnątrz inne bloki funkcyjne. Bloki *Device* zwiększają przejrzystość aplikacji PLC grupując różne funkcjonalności algorytmu sterowania.
- **Macro** – skompilowany i zamknięty złożony blok funkcyjny
- **Variable** – zmienna w aplikacji PLC wykorzystywana do pamiętania wartości nastawy lub odczytu parametru pracy.
- **HMI** - interfejs użytkownika. Dotyczy interakcji z użytkownikiem poprzez menu wyświetlane na panelu HMI. Służy do podglądu oraz zmiany nastaw zmiennych aplikacji PLC.

2. Instalacja oprogramowania

2.1 Minimalne wymagania sprzętowe

- Procesor: Pentium III 700 MHz lub wyżej
- Pamięć: 256 MB
- Dysk: 50 MB wolnej przestrzeni na program
- System: Windows XP / Vista / 7 / 8 / 10

2.2 Wymagania oprogramowania

- System Windows powinien posiadać zainstalowany dodatek Microsoft .NET Framework w wersji nie niższej niż 4.0 (darmowy dodatek do systemu: <https://www.microsoft.com/pl-pl/download/details.aspx?id=17851>)

Sterowniki PLC z serii ELPxxRyy-... (ELP11R32-V, ELP12R10, ELP13R14-P, itp.)

- Sourcery G++ Lite (darmowy kompilator do pobrania ze strony <https://sourcery.mentor.com/GNUToolchain/release830>)
Mirror: <http://el-piast.com/files/soft/arm-2010q1-188-arm-none-eabi.exe>
- Sterownik USB STM32 Virtual COM Port Driver (producent STMicroelectronics)
http://el-piast.com/files/soft/VCP_V1.3.1_Setup.exe – systemy Windows 32-bit
http://el-piast.com/files/soft/VCP_V1.3.1_Setup_x64.exe – systemy Windows 64-bit

Starsze sterowniki PLC z serii ELPxxTyy-... (ELP09T30-V, ELP10T32-V, itp.)

- WinAVR (darmowy kompilator do pobrania ze strony autora <http://winavr.sourceforge.net>)
- Sterownik USB firmy FTDI dla układu FT232R (darmowy do pobrania ze strony producenta <http://www.ftdichip.com/Drivers/D2XX.htm>)

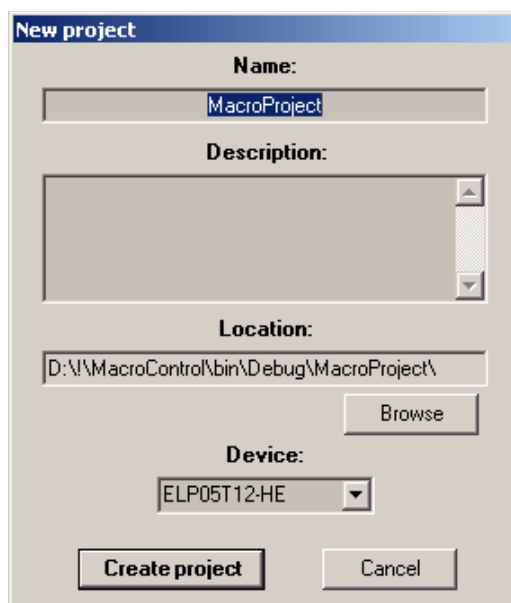
2.3 Instalacja

Przed instalacją *Macrocontrol* należy spełnić wymagania opisane w [2.2 Wymagania oprogramowania](#). Zalecana jest instalacja kompilatorów w folderach domyślnych proponowanych przez program instalacyjny. Po instalacji programów narzędziowych należy wykonać restart systemu. Oprogramowanie *Macrocontrol* instaluje się w systemie uruchamiając plik instalacyjny *MCsetup2.exe*. Ikona skrótu powinna znaleźć się na pulpicie. Wszystkie pliki o rozszerzeniach *mcf* zostaną skojarzone w systemie ze środowiskiem *Macrocontrol*. Oznacza to że pliki projektów *Macrocontrol* automatycznie będą otwierane w środowisku.

3. Pierwsze kroki

3.1 Nowy projekt

Po uruchomieniu środowiska *Macrocontrol* w celu zbudowania aplikacji sterującej PLC należy utworzyć nowy projekt. Kreator projektu uaktywnia się po wybraniu z menu *File* → *New* lub skrótem klawiszowym *Ctrl+N*.



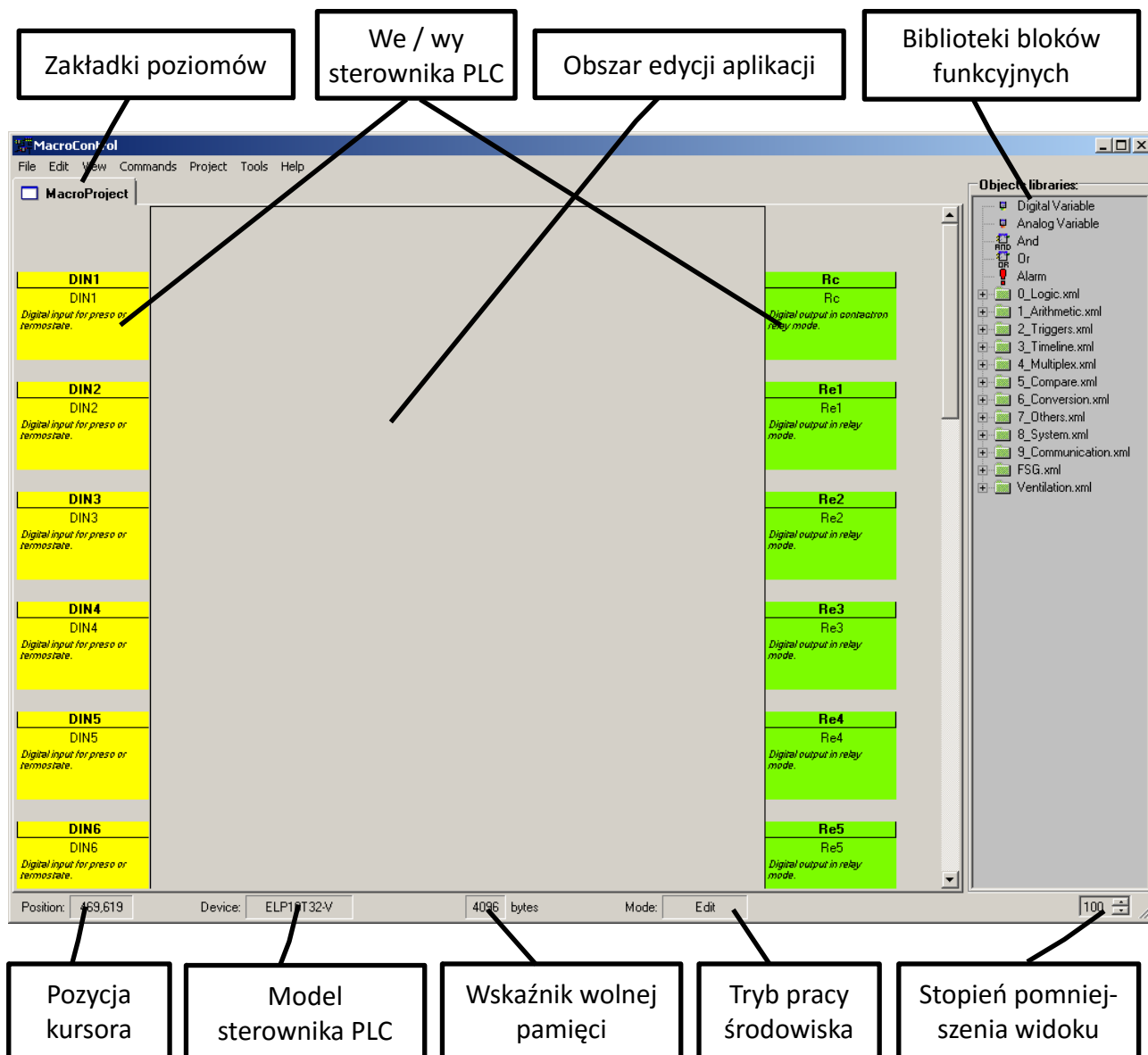
Należy wypełnić pola nazwy i opcjonalnie opisu, wybrać odpowiednią lokalizację dla projektu, oraz model sterownika PLC dla którego będzie przeznaczona aplikacja PLC. Aplikacja przeznaczona na inny sterownik nie będzie działała poprawnie i grozi zawieszeniem lub uszkodzeniem urządzenia. Folder projektu jest tworzony automatycznie i ma taką samą nazwę jak nazwa projektu. Po naciśnięciu przycisku *Create project* projekt zostanie utworzony.

W folderze aplikacji PLC powinien pojawić się plik wersji aplikacji (*version.hex*) oraz projektu (rozszerzenie *mcf*).

Uwaga: Ważne jest aby każda aplikacja była przechowywana w oddzielnym folderze. W przeciwnym wypadku kompilacja aplikacji może nie przebiegać prawidłowo.

3.2 Główne okno projektu

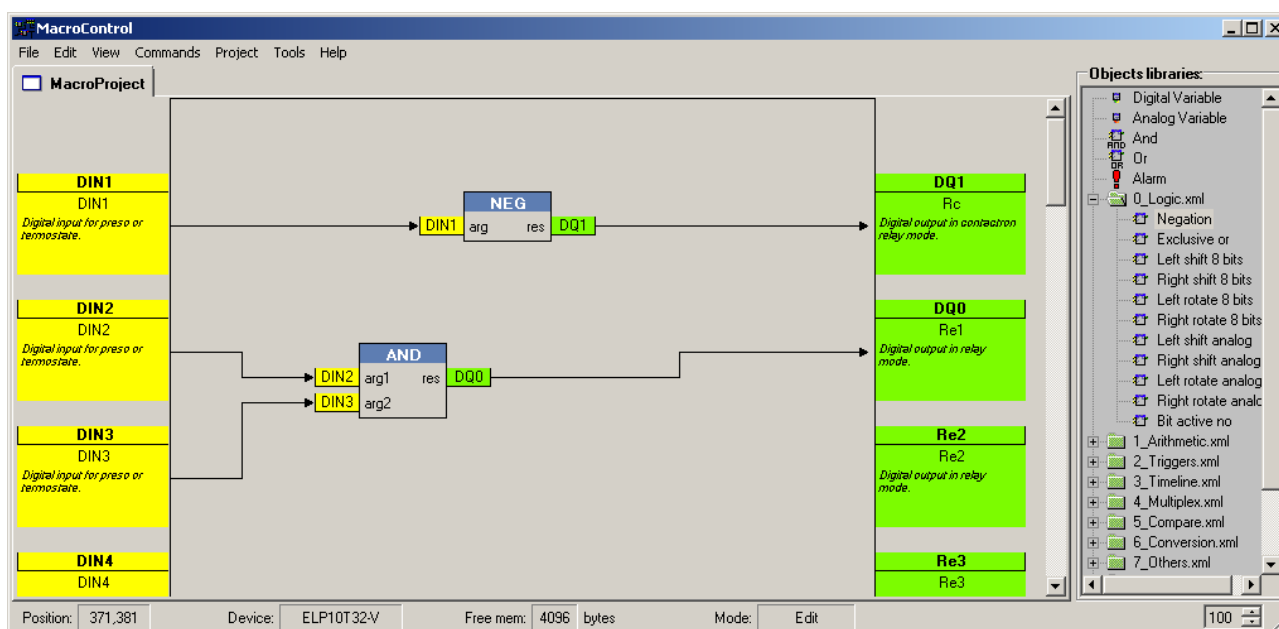
Gdy nowy projekt jest już utworzony na ekranie pojawia się główne okno aplikacji.



Zakładki poziomów przedstawiają aktualne zagłębienie w projekcie w przypadku wchodzenia w głąb bloków *Device* (podwójne kliknięcie na blok). W obszarze edycji wstawia się i łączy bloki funkcyjne budując aplikację PLC. Po lewej znajdują się wejścia sterownika, a po prawej wyjścia. Każdy typ we/wy jest w innym kolorze. W okienku po prawej znajdują się biblioteki bloków funkcyjnych oraz kliku bloków specjalnych (cyfrowa zmienna, analogowa zmienna, logiczny spójnik, logiczna suma, oraz blok alarmu). Biblioteki pogrupowane są w foldery według zastosowań. Opis działania bloków funkcyjnych znajduje się w odrębnym dokumencie *MacroControl - Func Descr.pdf*. Dolny pasek statusu zawiera informacje o aktualnej pozycji wskaźnika myszki, modelu sterownika PLC aplikacji, ilości wolnej pamięci sterownika, trybu pracy środowiska oraz stopnia pomniejszenia obszaru edycji. Rozmiar obszaru edycji można zmieniać rozciągając go za pomocą *spliterów* umieszczonych po prawej oraz dolnej części obszaru edycji aplikacji.

3.3 Tworzenie przykładowej prostej aplikacji PLC

W celu wprowadzenia bloków funkcyjnych do obszaru roboczego należy wybrać odpowiedni blok z biblioteki obiektów i przeciągnąć go do obszaru edycji lub po wybraniu go z biblioteki kliknąć w dowolnym miejscu obszaru edycji w którym ma pojawić się obiekt. Każde kolejne kliknięcie w obszarze edycji spowoduje pojawienie się kolejnego bloku tego samego typu. Sytuacja będzie się powtarzać do momentu wykonania innej akcji edycyjnej np. kliknięcie prawym przyciskiem myszki lub wybranie bloku lub we/wy w obszarze edycji. Łączenie bloków między sobą uzyskuje się poprzez dwukrotne kliknięcie na wybrane wejście lub wyjście. W takiej sytuacji powinna pojawić się linia łącząca wybrane wejście lub wyjście i kursor myszki. Następnie należy dwukrotnie kliknąć na we/wy z którym ma być nawiązane połączenie. Połączenie pokazuje przepływ informacji od wejścia do wyjścia. Istnieje kilka zasad łączenia we/wy. Nie można np. łączyć ze sobą dwóch wejść, jak również i dwóch wyjść. Jedynie wejścia sterownika PLC lub bloku *Device* (tylko wewnątrz bloku) mogą być łączone z wejściami innych bloków funkcyjnych i analogicznie wyjścia. Również łączenie we/wy różnych typów jest niedozwolone. Nie można połączyć we/wy analogowych z we/wy cyfrowymi. Każde wejście może mieć tylko jedno połączenie przychodzące, natomiast wyjście może mieć wiele wychodzących, wyjątek stanowią we/wy sterownika PLC i bloków *Device* (tylko wewnątrz bloku). Prosta aplikacja sterująca może wyglądać np. tak:

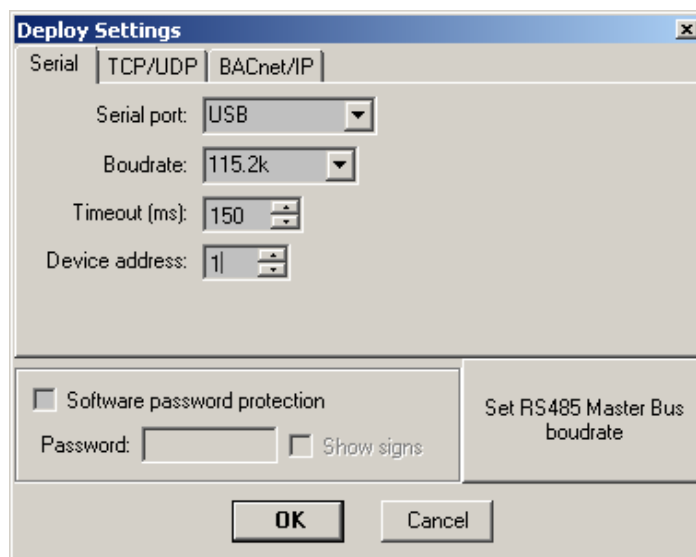


Wykorzystane zostały jedynie bloki negacji oraz spójnika logicznego. Analizując działanie aplikacji można stwierdzić że wyjście Rc sterownika PLC uzależnione jest od wejścia DIN1 w ten sposób że gdy $DIN1 = 0$ to $Rc = 1$ natomiast gdy $DIN1 = 1$ to $Rc = 0$. Natomiast drugi tor sterowania opisuje prostą zależność logiczną: gdy $DIN2 = 1$ i $DIN3 = 1$ to $Re1 = 1$, w przeciwnym wypadku $Re1 = 0$.

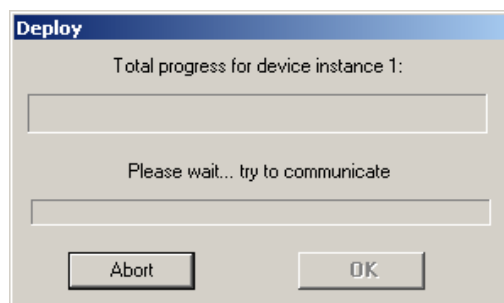
Gdy algorytm jest gotowy należy skompilować aplikację PLC. Można tego dokonać z menu *Project* → *Compile* lub skrótem klawiszowym *F6*. W zależności od modelu sterownika zostanie uruchomiony odpowiedni kompilator. Na dole w pasku w polu *Mode* pojawi się napis *Compile*. Jeżeli w trakcie kompilacji wystąpiły jakieś błędy zostanie wyświetlone dodatkowe okienko zawierające informację kompilatora. Jeżeli kompilacja przebiegła prawidłowo to w folderze aplikacji pojawią się wygenerowane pliki wynikowe do zapisu na sterownik PLC m.i. plik *project.hex* lub *project.bin*.

Po fazie kompilacji wynikową aplikację należy zapisać na sterowniku PLC. Sterownik należy

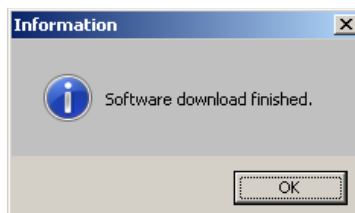
podłączyć do komputera poprzez łącze USB, konwerter USB/RS-485 lub Ethernet (RJ-45). Następnie należy skonfigurować tryb połączenia w *Macrocontrol*. W tym celu proszę wybrać z menu *Project* → *Communication settings*.



W przypadku typowego połączenia USB należy wybrać zakładkę *Serial*, port COMx, prędkość komunikacji 115.2k. Parametry *Timeout* = 300ms oraz *Device address* = 1 są domyślnymi nastawami, a opis ich działania można znaleźć w rozdziale [Communication Settings](#). Po zatwierdzeniu parametrów przyciskiem *OK*. Należy dokonać zapisu aplikacji na sterownik PLC wybierając z menu *Project* → *Upload All* lub skrótem klawiszowym *F7*.

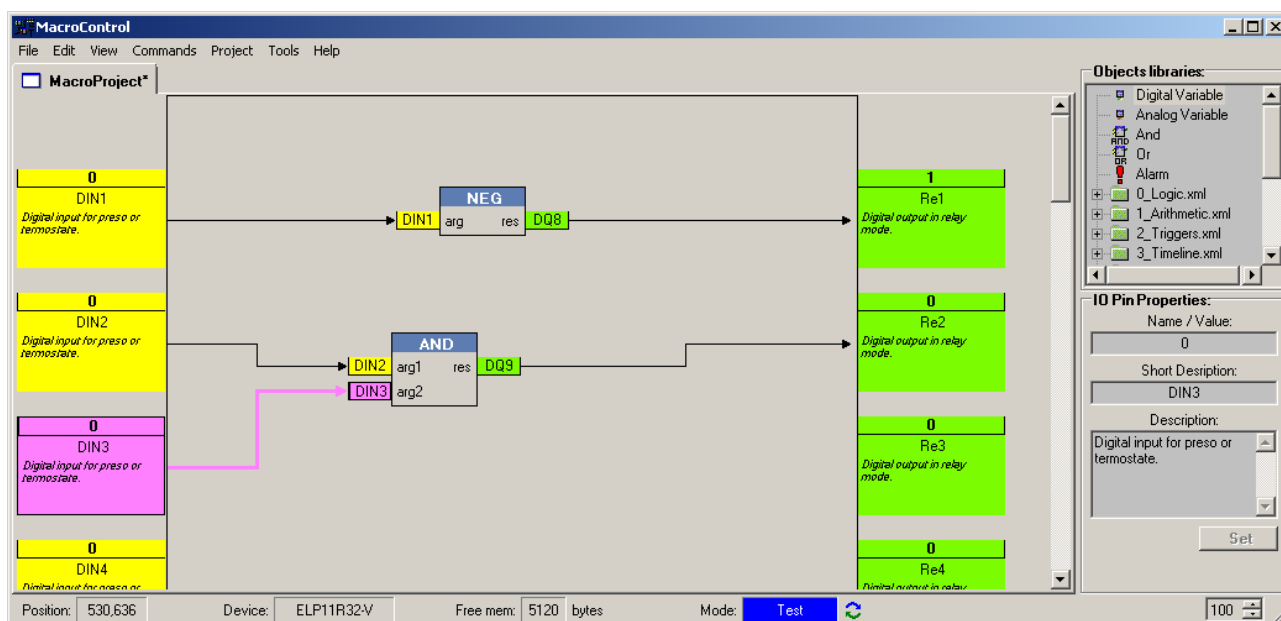


Następuje zapis na sterownik. Gdy pasek postępu dojdzie do końca pojawi się stosowny komunikat informujący o zakończeniu procesu zapisu.



Kolejnym etapem jest przetestowanie działania aplikacji PLC. Można tego dokonać bez potrzeby podłączania fizycznych sygnałów do we/wy sterownika. Aby uaktywnić emulację stanów na wejściach sterownika należy włączyć opcję w menu *Tools* → *Emulate IO in Test Mode*. Następnie należy uruchomić tryb testowania *Test Mode*. W tym celu proszę wybrać opcję menu *Tools* → *Start*

In-Target Test lub zastosować skrót klawiszowy *F12*.



Tryb pracy środowiska powinien zmienić się na *Test* a przy we/wy powinny pojawić się stany. Można teraz, wybierając odpowiednie wejście, zapisać na nim stan i obserwować reakcję wyjść. Stany na wejściach są emulowane bez konieczności podłączania fizycznych sygnałów. Opcję można wyłączyć w menu *Tools* → *Emulate IO in Test Mode* jedynie w trybie edycji. Stan wyjść jest zawsze widoczny w trybie testowania, a ich zmiany mają odzwierciedlenie na fizycznych wyjściach sterownika PLC. Tryb testowania wyłącza się w menu *Tools* → *Stop In-Target Test* lub skrótem klawiszowym *Shift+F12*.

W ten sposób można budować aplikację sterującą i testować jej funkcjonalność. Aby aplikacja była kompletna należy zbudować *Menu HMI* oraz skonfigurować obiekty *BACnet* dla systemów BMS (w przypadku sterowników *BACnet*).

4. System sterowania ELP

4.1 Reprezentacje liczbowe

W systemie sterowania *ELP* do obliczeń wykorzystuje się dwa typy wartości liczbowych: cyfrowy i analogowy. Wartości cyfrowe przechowywane są na 1 bicie informacji i mogą przyjmować stany 0 lub 1. Wartości analogowe mogą być różnie interpretowane w aplikacji PLC i posiadają kilka formatów reprezentacji w zależności od potrzeb. Dla wartości analogowej z przecinkiem liczba jest w postaci stałoprzecinkowej ze znakiem:

znak	23 bity części całkowitej	8 bitów części ułamkowej
------	---------------------------	--------------------------

Dokładność takiej liczby to $1/256 = 0,0039062$. W przypadku skomplikowanych operacji arytmetycznych gdy błąd obliczeń jest duży należy wszystkie wartości używane do obliczeń przemnożyć przez pewną stałą, a wynik podzielić przez tą samą stałą.

Wartość analogową można również przedstawić jako całkowitą ze znakiem w postaci dziesiętnej lub szesnastkowej:

znak	31 bitów części całkowitej
------	----------------------------

Dla przykładu wartość stałoprzecinkowa 26,3125 w postaci dziesiętnej to $26 \cdot 256 + 80 = 6736$. Natomiast wartość dziesiętna 987654 w postaci stałoprzecinkowej będzie równa 3858,0234.

Dla wartości analogowych maksymalną ujemną liczbę można wykorzystać jako reprezentację stanu nieokreślonego. Może to być przydatne np. przy pomiarach z czujników temperatury, gdy czujnik jest niepodłączony wejście temperaturowe przyjmuje nieokreśloną wartość analogową oznaczającą brak czujnika.

reprezentacja stałoprzecinkowa: -8388608

reprezentacja dziesiętna: -2147483648

reprezentacja szesnastkowa: 0x80000000

Taki sposób pamiętania wartości analogowych ułatwia operacje arytmetyczne procesorom nie wyposażonym w koprocesory arytmetyczne do obsługi operacji zmiennoprzecinkowych co znacznie przyczynia się do szybkości działania układu. W systemach wbudowanych najczęściej wykorzystuje się procesory RISC które nie wspierają operacji zmiennoprzecinkowych.

Istnieją jeszcze dwa formaty wartości analogowych określające czas i datę. Wartość analogowa może być interpretowana jako czas w następujący sposób:

8 bitów godzina	8 bitów minuta	8 bitów sekunda	8 bitów nieużywane
-----------------	----------------	-----------------	--------------------

lub jako datę:

8 bitów rok	8 bitów miesiąc	8 bitów dzień	8 bitów nieużywane
-------------	-----------------	---------------	--------------------

Podsumowując, 32-bitowa wartość analogowa może występować w pięciu formatach, które w zależności od bloku funkcyjnego różnie są interpretowane. Jeżeli ustalimy że wartość jest

w formacie daty i trafia ona na wejście bloku funkcyjnego na którym powinna być wartość stałoprzecinkowa to właśnie tak zostanie ona zinterpretowana. Poniżej przykład tej samej wartości analogowej w różnych formatach:

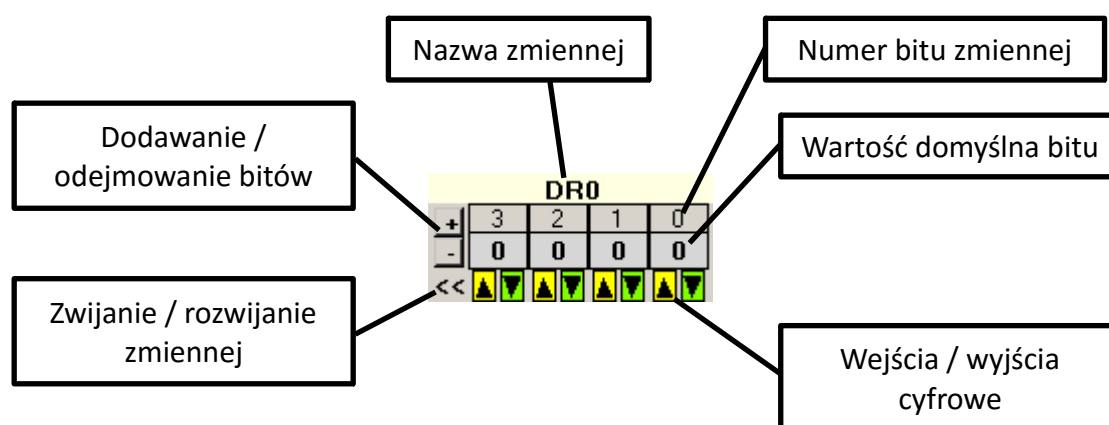
Format stałoprzecinkowy (Fixed)	986141,313
Format dziesiętny (Decimal)	252452176
Format heksadecymalny (Hex)	0F0C1D50
Format czasu (Time hh:mm:ss)	15:12.29
Format daty (Date DD-MM-YY)	29-12-15

4.2 Zmienne

Wszystkie zmienne w systemie są 32-bitowymi wartościami. Przy budowaniu aplikacji PLC można posługiwać się dwoma typami zmiennych: zmienne cyfrowe i zmienne analogowe. W rzeczywistości oba typy zmiennych są 32-bitowymi wartościami, jednak w przypadku zmiennych cyfrowych można w łatwy sposób dostać się do pojedynczego bitu informacji. Zmienne cyfrowe mogą wykorzystywać od 1 do 32 bitów jednak w sterowniku na każdą zmienną zawsze rezerwowana jest pamięć 32 bitów (4 bajty).

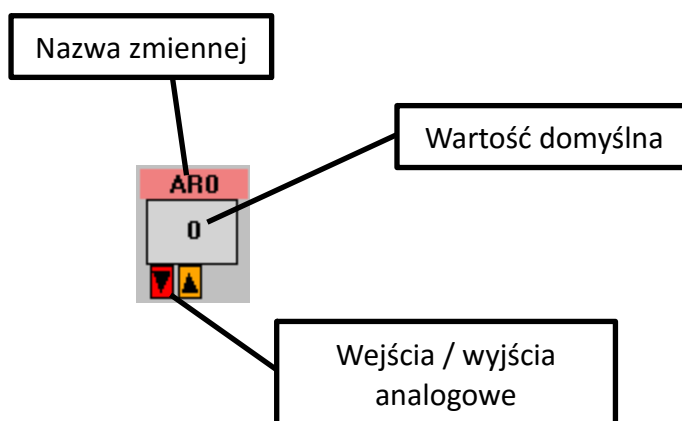
Zmienne mają za zadanie przechowywać parametry aplikacji PLC sterownika i mogą być odczytywane i/lub ustawiane z zewnątrz poprzez HMI lub inne nadrzędne urządzenie sterujące. Oznacza to że poprzez zmienne można komunikować się z systemami nadrzędnymi. Należy przy tym pamiętać że zmiana wartości każdej zmiennej nadpisywanej z zewnątrz powoduje operację zapisu do nieulotnej pamięci EEPROM. W związku z tym zapisywanie do zmiennych sterownika często zmieniających się wartości (np. temperatura), może spowodować szybkie uszkodzenie się sterownika. W takim wypadku należy zamiast zapisywać dane do sterownika, odczytywać je przez sterownik, lub zastosować specjalne komendy nie powodujące zapisu do EEPROM, dostępne dla komunikacji Modbus.

- Zmienne cyfrowe



Ilość bitów zmiennych zmienia się przyciskami +/- . Kolejność bitów zmienia się poprzez przeciągnięcie ich w lewo lub prawo. Zmienne cyfrowe zwijają się i rozwijają aby zmniejszyć ilość zajmowanej przestrzeni w obszarze edycji, przyciskiem oznaczonym strzałkami lub poprzez dwukrotne kliknięcie na zmiennej.

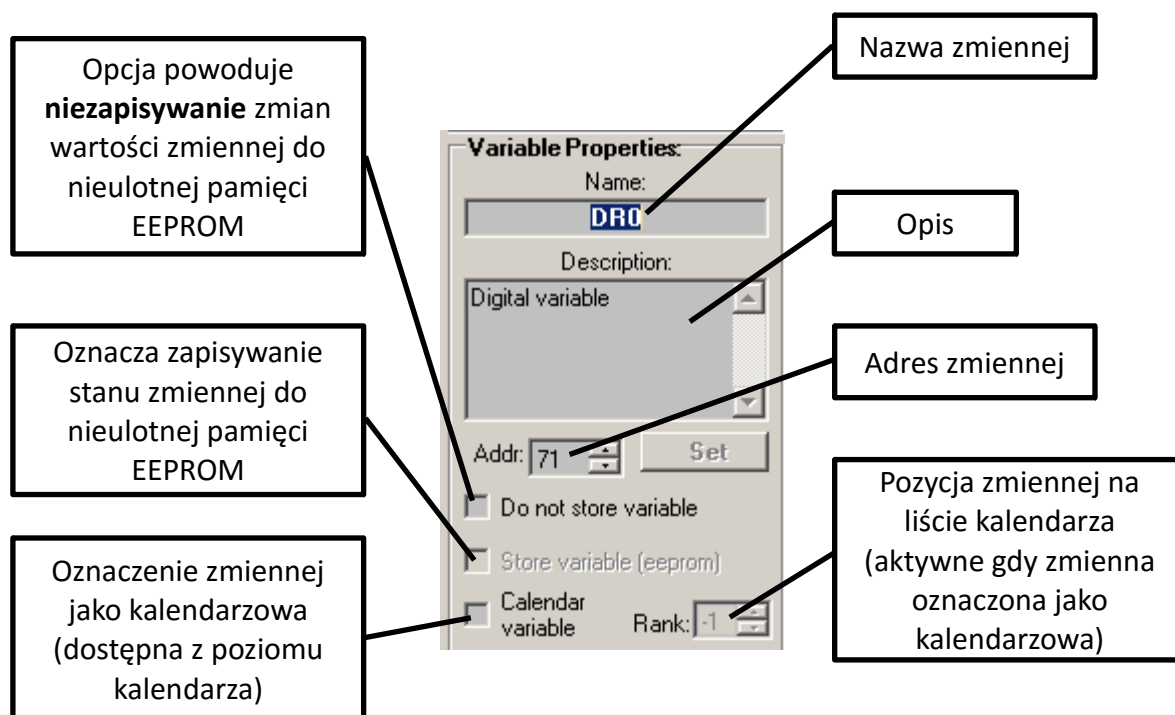
- Zmienne analogowe



Zmienne analogowe mogą wyświetlać wartość domyślną w wybranej reprezentacji. W zależności od reprezentacji zmienia się kolor czcionki wartości zmiennej.

- Właściwości zmiennej

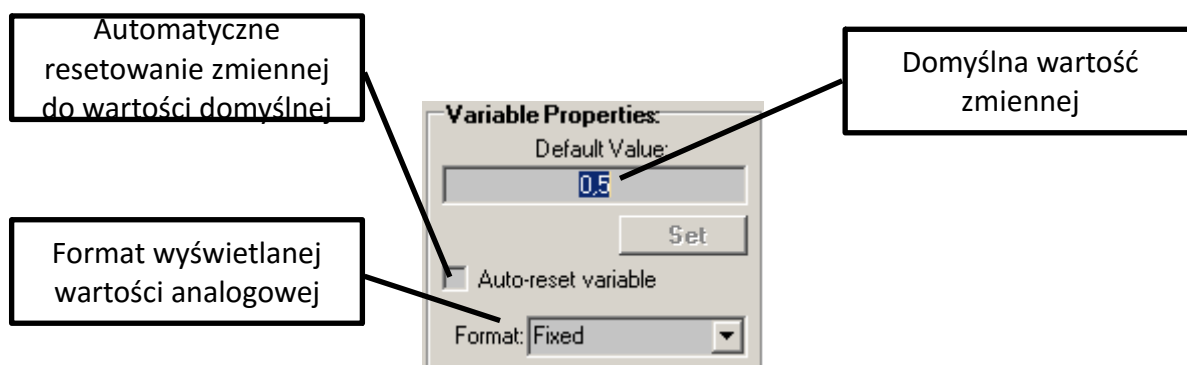
Po wybraniu dowolnej zmiennej poprzez kliknięcie lewym przyciskiem myszki na pasku z nazwą zmiennej, na dole po prawej stronie w edytorze *Macrocontrol* pojawi się karta właściwości zmiennej.



Na karcie właściwości zmiennej można ustalić jej nazwę i opis. Adres zmiennej jest wykorzystywany jako adres rejestru Modbus lub numer instancji obiektu BACnet. Jeżeli zmienna nie jest nadpisywana z poziomu aplikacji PLC (jej wejście jest wolne) to zmiana jej wartości jest możliwa jedynie z zewnątrz (z systemu nadrzędnego). Każda zmiana wartości zmiennej z zewnątrz jest zapamiętywana w nieulotnej pamięci EEPROM chyba że zostanie zaznaczona opcja *Do not*

store variable. Wtedy zmienna nie jest przechowywana w pamięci EEPROM, a po restarcie sterownika przyjmuje wartość domyślną. Gdy zmienna jest nadpisywana z poziomu aplikacji PLC (jej wejście jest podłączone) to domyślnie wartości zmiennej są zapisywane jedynie w pamięci RAM. Po restarcie sterownika zmienna przez moment ma wartość 0 jednak już po chwili zostaje nadpisana wartością obliczoną przez aplikację. Czasami konieczne jest aby zmienna dodatkowo była pamiętana w pamięci EEPROM tak aby po restarcie sterownika jej wartość była taka jak ostatnio. Zaznaczając tą opcję należy pamiętać że każda zmiana wartości wiąże się z zapisem do pamięci EEPROM dla której liczba zapisów jest ograniczona i w zależności od urządzenia może wynosić 100 tys. – 1 mln. zapisów. Tą opcję należy używać jedynie dla zmiennych których wartości zmieniają się co pewien czas. Dla przykładu zmiana wartości co 1 sekundę wiąże się z 3600 zapisami na godzinę co może spowodować uszkodzenie pamięci EEPROM już po niecałych 28 dniach. Oznaczenie zmiennej jako kalendarzowa powoduje że jej wartość może być nadpisywana z kalendarza pod warunkiem że kalendarz jest aktywny i istnieje program, który ma być wykonywany w danym momencie (zależne od daty i godziny zegara RTC) zawierający tą zmienną. W jednej aplikacji można oznaczyć maksymalnie 10 zmiennych kalendarzowych.

Zmianę wartości domyślnych zmiennych dokonuje się w na dodatkowej karcie właściwości zmiennej pojawiającej się po kliknięciu na wartość domyślną wybranej zmiennej.



Dopuszczalny zakres wartości domyślnych jest zależny od typu i formatu zmiennej wybranego w polu *Format*. Zmienna może automatycznie resetować swoją wartość do nastawy domyślnej po każdej zmianie. Jest to możliwe w przypadku zmiennych których wartości są nadpisywane z zewnątrz (wejście zmiennej jest wolne). Każda zmiana wartości zmiennej jest aktualna jedynie przez jeden przebieg algorytmu aplikacji PLC, a następnie z powrotem przyjmowana jest wartość domyślna. Funkcja jest bardzo przydatna do jednorazowego wyzwania pewnych akcji sterujących algorytmu PLC np. resetowania alarmów, liczników itp.

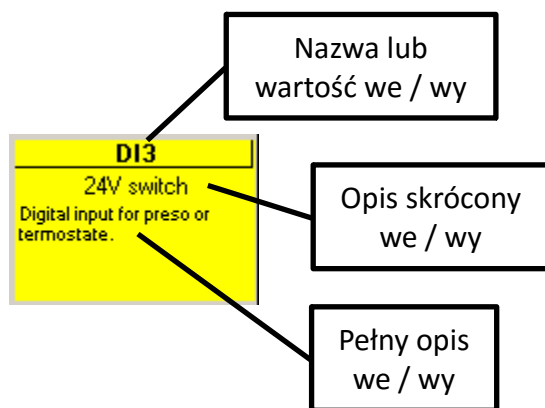
4.3 Wejścia / wyjścia

Wejścia i wyjścia służą do wymiany informacji między blokami funkcyjnymi i zmiennymi. Wejścia i wyjścia mogą być cyfrowe lub analogowe. Wyjście cyfrowe może być podłączone jedynie do wejścia cyfrowego, a analogowe do analogowego. Wejścia bloku sterownika lub złożonego bloku funkcyjnego typu *Device* są wyjściami dla bloków funkcji wewnątrz tego bloku i analogicznie dla wyjść. Łączenie we/wy odbywa się poprzez dwukrotne kliknięcie lewym przyciskiem myszki na we/wy które mają być połączone.

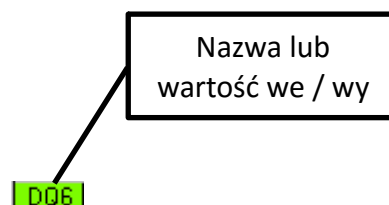
▪ Graficzna reprezentacja wejść / wyjść

Wejścia / wyjścia bloków funkcyjnych mogą być przedstawione w dużym formacie (tylko dla we/wy sterownika lub bloku funkcyjnego typu *Device*), lub małym formacie. Zmienne także posiadają wejścia i wyjścia zorientowane pionowo.

wejście cyfrowe (duży format)



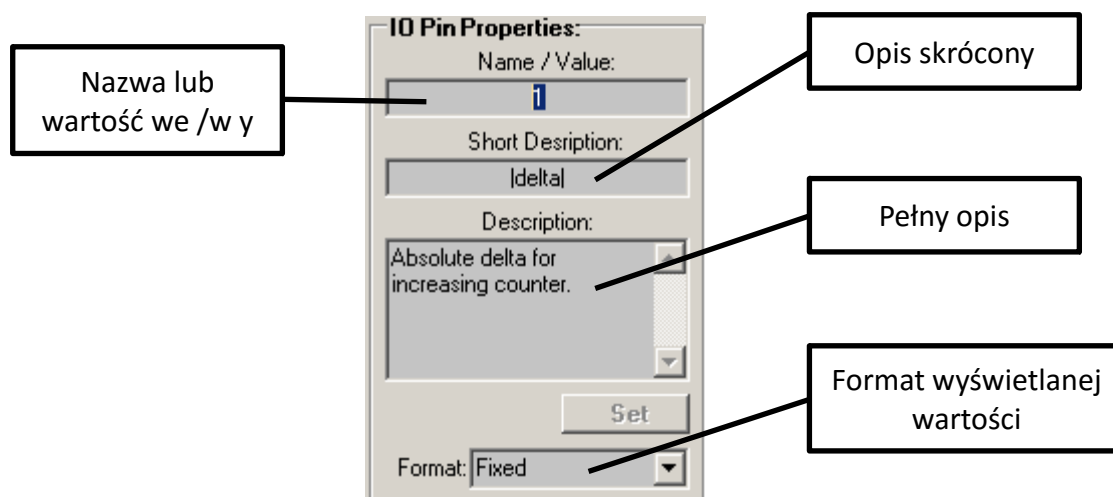
wejście cyfrowe (mały format)



Dla wejść bloków funkcyjnych, które nie posiadają połączeń przychodzących, zamiast nazwy widnieje stała wartość jaka ma być użyta dla wywołania funkcji bloku.

▪ Właściwości wejść / wyjść

Po wybraniu dowolnego we/wy poprzez kliknięcie na nim lewym przyciskiem myszki, na dole po prawej stronie wewnątrz okna *Macrocontrol* pojawi się karta właściwości we/wy.

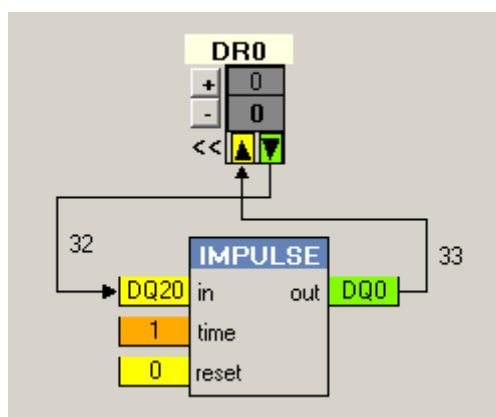


Na karcie właściwości we/wy można dokonać zmiany nazwy lub, gdy wejście jest wolne, stałej wartości dla wejścia, ustalić opis właściwości wejścia, a także, w przypadku we/wy analogowych, ustalić format liczbowy przyjmowanych wartości.

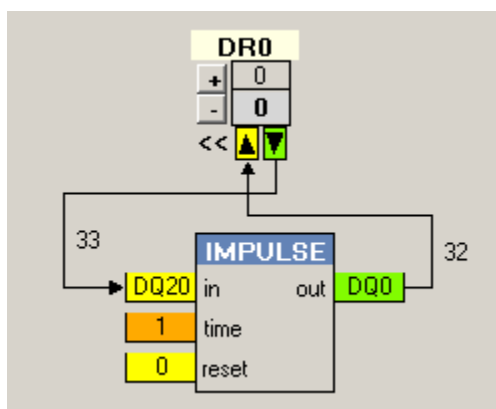
4.4 Połączenia

Wejścia i wyjścia można łączyć ze sobą. W celu dokonania połączenia należy dwukrotnie kliknąć na wybrane wejście / wyjście, powinna pojawić się linia podążająca za kursorem myszki. Następnie dwukrotnie kliknąć na drugie wejście / wyjście z którym ma być utworzone połączenie. Połączenie można dowolnie przesuwać w obszarze roboczym. Każde połączenie posiada swój priorytet wykonania. Włączenie wyświetlania priorytetów dla połączeń dokonuje się z menu *View* → *Show connections priority*. Priorytet oznacza kolejność przekazywania danych między we/wy. Niższa wartość oznacza wcześniejsze wykonanie działania. Kolejność ma największe znaczenie w przypadku tworzenia połączeń ze sprzężeniem zwrotnym.

Przykład wykorzystania priorytetów połączeń:



Powyższa operacja posiada sprzężenie zwrotne. Blok *IMPULSE* utrzymuje na wyjściu *out* stan wysoki przez okres *time* (w tym wypadku 1 sekundę) gdy na wejściu *in* pojawi się zbocze narastające (w praktyce jest to zmiana ze stanu 0 na stan 1). Kolejność wykonywania mówi że gdy zmienna *DR0* zostanie ustawiona na wartość 1 blok *IMPULSE* wykona swoją operację i w wyniku ustawi wyjście *out* na stan 1, a następnie ta wartość zostanie przepisana do zmiennej *DR0*. Po upływie 1 sekundy blok *IMPULSE* zmieni stan swojego wyjścia *out* z powrotem na wartość 0, a następnie zmienna *DR0* również przyjmie ten stan. Co się stanie gdy kolejność wykonywania działań będzie inna?



W tym wypadku najpierw stan z wyjścia *out* bloku *IMPULSE* zapisywany jest do zmiennej *DR0*, a następnie wartość ze zmiennej przekazywana jest na wejście *in*. W stanie spoczynkowym na wyjściu *out* bloku *IMPULSE* będzie wartość 0 i jakakolwiek zmiana wartości zmiennej *DR0* spowoduje natychmiastowe jej nadpisanie stanem 0. W praktyce żadna operacja nigdy się nie

wykona.

Gdy aplikacja sterująca jest duża i występuje w nich szereg zależności czasowych i możliwych sprzężeń zwrotnych należy zwrócić uwagę na priorytety wykonywania działań które mogą powodować niespodziewaną, błędną funkcjonalność. W celu ustalenia optymalnych wartości priorytetów wykonywania działań można skorzystać z funkcji optymalizacji połączeń *Commands* → *Optimize connections prior*. Funkcja jednak nie gwarantuje że aplikacja PLC będzie działała dokładnie tak, jak przewiduje użytkownik.

4.5 Biblioteki bloków funkcyjnych

Po prawej stronie okna aplikacji znajdują się biblioteki bloków funkcyjnych, z których można budować aplikacje PLC. Oprócz bibliotek w oknie znajduje się kilka bloków specjalnych. Każdy elementarny blok z okna bibliotek można przenieść do obszaru edycji poprzez „przeciągnij i upuść” lub wybierając go w oknie bibliotek, a następnie klikając myszą w miejscu gdzie ma się on pojawić w obszarze edycji. Można tworzyć własne nowe biblioteki (menu kontekstowe: *New*), dodawać istniejące (menu kontekstowe: *Add*), usuwać (menu kontekstowe: *Remove*). Biblioteki zaznaczone zieloną ikoną folderu nie można usuwać, są to domyślne biblioteki środowiska. Do bibliotek można dodawać makra. Z biblioteki nie będącej domyślną można usuwać bloki funkcyjne.

▪ Utworzenie nowej biblioteki

Aby utworzyć nową bibliotekę należy kliknąć prawym przyciskiem myszki w oknie bibliotek i z menu kontekstowego wybrać opcję *New* lub użyć skrótu klawiszowego *Ctrl+N* gdy okno bibliotek jest aktywne (jest na nim skupiony fokus). Gdy pojawi się okno plików *New library* należy określić lokalizację nowej biblioteki i wpisać jej nazwę. Biblioteka automatycznie otrzyma rozszerzenie *xml* i pojawi się w oknie bibliotek.

▪ Dodawanie istniejącej biblioteki

Należy z menu kontekstowego okna bibliotek wybrać opcję *Add* lub użyć skrótu klawiszowego *Ctrl+I*. W oknie plików *Add library* wybrać odpowiedni plik *xml*. Biblioteka powinna pojawić się w oknie bibliotek. Uwaga, gdy dodana biblioteka zmieni lokalizację na dysku należy ponownie ją dodać do projektu.

▪ Usuwanie biblioteki

W celu usunięcia biblioteki z projektu należy ją wybrać w oknie bibliotek a następnie z menu kontekstowego użyć opcji *Delete* lub skrótu klawiszowego *Ctrl+D*. Biblioteka jest usuwana z projektu ale nie z dysku. Nie można usunąć bibliotek domyślnych oznaczonych zielonym kolorem.

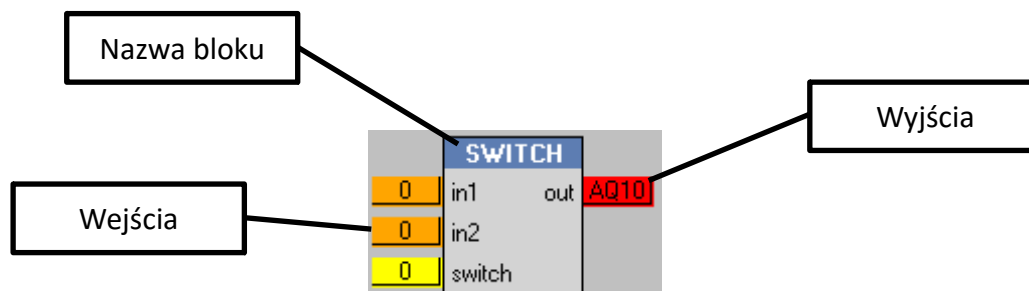
4.6 Bloki funkcyjne

Bloki funkcyjne dzielą się na 5 typów:

- Alarm
- Operacje logiczne z dowolną ilością parametrów And i Or

- Prosty blok funkcyjny
- Złożony blok funkcyjny (*Device*)
- Skompilowany złożony blok funkcyjny (*Macro*)

▪ Graficzna reprezentacja



Bloki funkcyjne *Alarm*, *And*, *Or*, *Prosty blok*, *Device*, *Macro* nie różnią się znacznie wyglądem graficznym. Można je odróżnić po kolorze wewnętrznego prostokąta bloku. Wszystkie kolory i rozmiary można dowolnie zmieniać według własnych preferencji w oknie ustawień widoku dostępnym z menu *View* → *Settings* opisanym w rozdziale [Settings](#). Graficzna reprezentacja bloku funkcyjnego zawiera jego nazwę oraz wejścia i wyjścia. Każde we/wy jest oznaczone skróconym opisem. Dwukrotne kliknięcie myszką na prostym bloku funkcyjnym lub bloku typu *Macro* pozwala wyświetlić okno w którym zawarty jest kod funkcji bloku napisany w języku C.

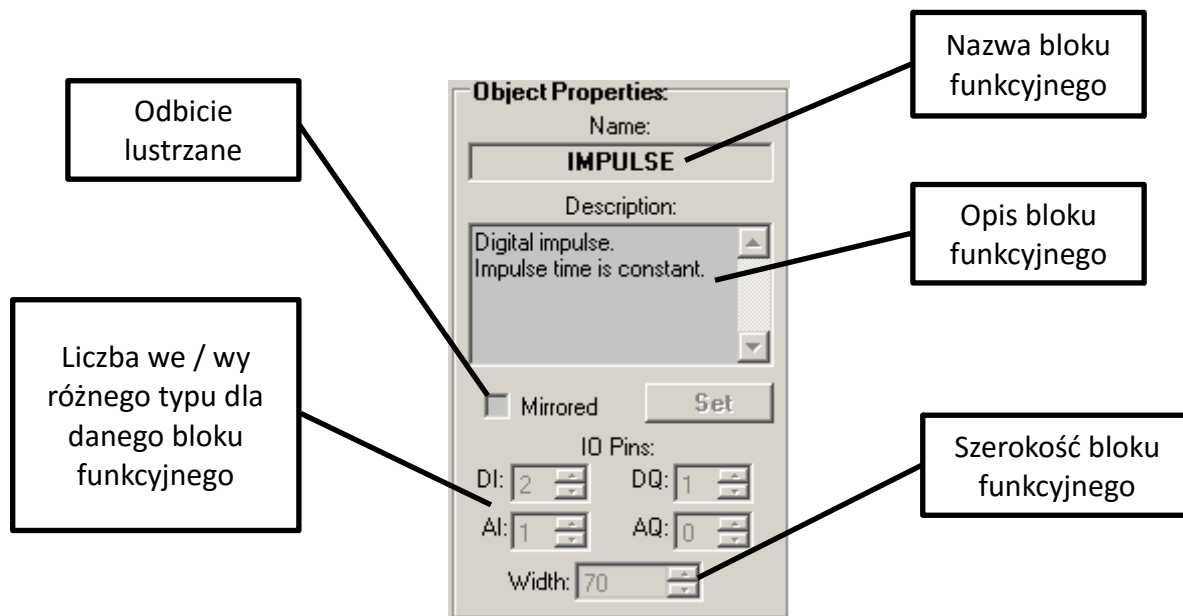
```
Source
void SIG(API_Analog a, API_Analog b, API_Analog c, API_Digital *d, )
{
    if (API_getGD(GD0))
    {
        if (a < b-c)
            API_setGD(GD0,0);
        }
    else
    {
        if (a > b)
            API_setGD(GD0,1);
        }
    *d = API_getGD(GD0);
}

Save Cancel
```

Kod wykorzystuje specjalne funkcje API sterownika. Aby edytować kod bloków funkcyjnych z bibliotek spoza domyślnych należy odblokować możliwość edycji w menu *Edit* → *Unlock FBD edit*.

▪ Właściwości

Dostęp do właściwości bloku funkcyjnego uzyskuje się po kliknięciu na nim. W prawym dolnym rogu pojawi się karta właściwości. W zależności od typu bloku funkcyjnego niektóre opcje nie mogą być poddane edycji.



Nazwa bloku funkcyjnego i opis może być edytowany tylko dla bloków typu *Alarm* oraz *Device*. Blok można wyświetlić w odbiciu lustrzanym. Dla bloków typu *And*, *Or* można ustalać dowolną ilość wejść cyfrowych natomiast dla bloków *Device* dowolną ilość wszystkich typów wejść i wyjść. Dla bloków *Device* można zmieniać szerokość tak aby wszystkie opisy mieściły się na ekranie. Po skonwertowaniu bloku *Device* do *Macro* możliwość zmiany szerokości, nazwy i ilości we /wy bloku zostaje zablokowana.

▪ Operacje logiczne z dowolną ilością parametrów And i Or

Bloki *And* i *Or* to dwie podstawowe operacje logiczne. Wejściami (parametrami wejściowymi) dla *And* czy *Or* są wartości cyfrowe. Wejść może być dowolnie dużo. Ilość wejść można zmieniać po kliknięciu na blok na karcie właściwości obiektu w polu *DI*. Wyjście jest zawsze tylko jedno. Graficzna reprezentacja bloku jest identyczna jak dla prostego bloku funkcyjnego.

▪ Alarmy

Bloki typu *Alarm* odpowiedzialne są za obsługę alarmów systemu sterowania. Nazwa bloku jest jednocześnie nazwą alarmu zgłaszanego przez sterownik PLC w momencie jego wystąpienia. Alarmy mogą być blokujące lub automatycznie resetujące. Dokładny opis bloku *Alarm* znajduje się w odrębnym dokumencie *MacroControl - Func Descr.pdf*

▪ Proste bloki funkcyjne

Proste bloki funkcyjne to funkcje spełniająca specyficzne, przypisane im, zadania. *Macrocontrol* zawiera kilkadziesiąt prostych bloków funkcyjnych w bibliotekach podzielonych według

funkcjonalności. Dzięki nim możliwe jest budowanie dowolnych aplikacji sterujących dla sterowników PLC. Użytkownik może sam definiować swoje własne biblioteki bloków rozszerzając możliwości oprogramowania o własne preferencje.

▪ Złożone bloki funkcyjne typu *Device*

Złożony blok funkcyjny posiada wewnątrz pewną liczbę bloków funkcyjnych (prostych lub złożonych) i w całości stanowi niezależny blok sterujący. Blok złożony *Device* posiada dodatkowe właściwości takie jak możliwość zmiany kolejności wejść / wyjść (poprzez przeciąganie myszką), dodawanie wejść / wyjść. Klikając dwukrotnie myszką na bloku można zajrzeć do jego wnętrza. Dowolną grupę bloków funkcyjnych można zamykać w złożone bloki funkcyjne, a następnie konwertować do bloków typu *Macro*. Aby tego dokonać należy zaznaczyć odpowiednie bloki w obszarze edycji a następnie z menu kontekstowego wybrać opcję *Convert to device* lub z menu głównego *Commands* → *Convert to device*. Bloki *Device* nie mogą wewnątrz zawierać zmiennych oraz bloków typu *Alarm*. W każdej chwili można dokonać de-konwersji wstecz czyli przenieść zawartość bloku *Device* na poziom wyżej. Przed wykonaniem tej operacji należy zapewnić sobie dostateczną ilość wolnej przestrzeni w obszarze roboczym wokół wybranego bloku *Device* zaznaczyć go, a następnie wybrać opcję z menu kontekstowego *Break device* lub z menu głównego *Commands* → *Break device*. Blok *Device* można w dalszym etapie zamienić na *Macro* opcją z menu kontekstowego *Convert to macro* lub z menu głównego *Commands* → *Convert to macro*.

▪ Złożone bloki funkcyjne typu *Macro*

Macro jest złożonym blokiem funkcyjnym zamkniętym i umieszczonym w bibliotece funkcji. Blok *Macro* zawiera wewnątrz inne bloki funkcyjne tak jak jest to w przypadku bloku *Device*. Zasadnicza różnica między blokiem *Device*, a *Macro* oprócz braku możliwości edycji jest taka że blok *Macro* jest funkcją której kod występuje jednorazowo, a każde jej użycie w aplikacji PLC jest jedynie wywołaniem funkcji z odpowiednimi parametrami. W przypadku bloku *Device* każde jego wystąpienie w aplikacji PLC traktowane jest jako inna odrębna funkcja. W praktyce wielokrotne użycie w aplikacji bloku *Macro* generuje mniej kodu wynikowego dla sterownika PLC. *Macro* jest zablokowane przed próbą edycji i jedynie konwersja wstecz (opcja *Break macro*) do bloku *Device* umożliwia dostanie się do wnętrza bloku. *Macro* ma zbliżone właściwości do prostego bloku funkcyjnego, lecz może zawierać w sobie wiele tego typu bloków.

5. Menu główne

5.1 File

Menu plików służy do zarządzania plikami projektów aplikacji PLC oprogramowania *Macrocontrol*.

- **New**

Uruchomienie kreatora nowego projektu aplikacji PLC. Opis tworzenia nowego projektu znajduje się w rozdziale [3.1 Nowy projekt](#).

- **Open**

Otwarcie pliku istniejącego projektu aplikacji PLC. Pliki *Macrocontrol* posiadają rozszerzenie *mcf* (*macrocontrol file*). Wszystkie pliki *mcf* są skojarzone w systemie z oprogramowaniem *Macrocontrol* w procesie instalacji. Uruchomienie pliku projektu spowoduje automatyczne otwarcie oprogramowania *Macrocontrol*.

- **Save**

Zapisanie pliku aktualnie edytowanej aplikacji PLC. Projekt jest zapisywany w domyślnym folderze projektu ustalonym w procesie tworzenia nowego projektu (opcja *New*). Operacja zapisu wywoływana jest automatycznie po uruchomieniu kompilatora. Każda zmiana edycyjna w aplikacji PLC spowoduje niezgodność z plikiem zapisanym na dysku. Informacja o zmianie w aplikacji oznaczana jest symbolem * przy zakładce z nazwą projektu.

- **Save as**

Zapisanie pliku aktualnie edytowanej aplikacji PLC pod inną nazwą. Jedynie plik aplikacji *mcf* zapisywany jest pod inną nazwą. Z reguły projekt zawiera również plik menu HMI (*CtrlConf.xml*, *MainConf.xml*) oraz plik zmiennych BACnet (*BACnetConf.xml*). Pliki te, jeżeli są potrzebne, należy ręcznie skopiować do folderu nowo nazwanej aplikacji PLC. Jeżeli aplikacja zostanie zapisana w tym samym folderze w którym istniała pod starą nazwą wywołanie kompilacji spowoduje usunięcie plików binarnych (wynikowych) starego projektu, jednak plik aplikacji *mcf* nie zostanie naruszony.

- **Exit**

Wyjście z aplikacji. Jeżeli ostatnie zmiany aktualnie edytowanej aplikacji PLC nie zostały zapisane *Macrocontrol* zapyta czy zrobić to przed zamknięciem.

5.2 Edit

Zestaw podstawowych operacji powiązanych bezpośrednio z edycją elementów w obszarze roboczym środowiska *Macrocontrol*.

- **Undo**

Cofa ostatnią operację edycyjną o jeden krok. *Undo* nie cofa wszystkich operacji edycyjnych, a tylko najważniejsze związane z usuwaniem i dodawaniem elementów.

- **Redo**

Przywraca ostatnio cofniętą przez *Undo* operację edycyjną.

- **Cut**

Wycina zaznaczone w obszarze edycyjnym obiekty i zapamiętuje je w schowku. Obiekty ze schowka można wkleić operacją *Paste*.

- **Copy**

Kopiuje zaznaczone obiekty do schowka. Obiekty ze schowka można wkleić operacją *Paste*.

- **Paste**

Wkleja wcześniej wycięte lub skopiowane obiekty znajdujące się w schowku. Miejsce wklejenia jest określone na podstawie ostatniego kliknięcia myszki w obszarze edycyjnym. Należy zapewnić wolne miejsce w obszarze edycyjnym dla wklejanych obiektów tak aby obiekty nie nachodziły na siebie lub nie poprzesuwały się.

- **Delete**

Usuwa zaznaczone obiekty z obszaru edycyjnego. Usunięcie można cofnąć operacją *Undo*.

- **Select All**

Zaznacza wszystkie obiekty zawarte w obszarze edycyjnym.

- **Find**

Umożliwia wyszukanie łańcucha znaków w nazwach i opisach obiektów i wejść / wyjść. Znaleziony obiekt jest zaznaczany. Ponowne wywołanie *Find* przeszukuje projekt od ostatnio znalezionej miejsca.

- **Collision detection**

Włącza lub wyłącza wykrywanie kolizji między obiektami. Gdy wykrywanie kolizji jest włączone aplikacja nie pozwala na ustawienie obiektów tak aby na siebie nachodziły. Jeżeli nie jest możliwe znalezienie wolnego miejsca obiekt ostatecznie zostanie umieszczony na innych obiektach.

- **Zoom on scroll / Zoom on Ctrl+scroll**

Tryb działania operacji pomniejszania. Pomniejszanie odbywa się scrollem myszki lub z dodatkowo przytrzymanym klawiszem *Ctrl*.

- **Unlock FBD edit**

Odblokowuje możliwość edycji kodu języka C bloków funkcyjnych z bibliotek użytkownika (dodanych ręcznie do projektu). Aby dokonać edycji należy dwukrotnie kliknąć myszką na wybranym bloku funkcyjnym, dokonać zmian i zapisać je opcją *Save*. Funkcjonalność jest

przeznaczona tylko dla użytkowników zaawansowanych.

5.3 View

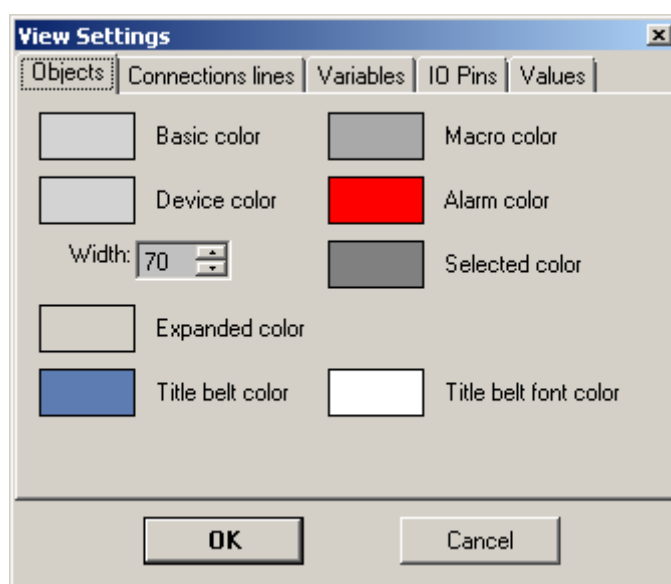
Ustawienia dotyczące sposobu prezentacji elementów graficznych znajdujących się w obszarze edycyjnym środowiska *Macrocontrol*.

- **Show connection priority**

Włącza lub wyłącza wyświetlanie priorytetów połączeń rzutujących na kolejność wykonywania operacji algorytmu sterowania. Opis działania priorytetów na przykładzie znajduje się w rozdziale [4.4 Połączenia](#).

- **Settings**

Ustawienia pozwalające dostosować wygląd edytora do własnych preferencji. Można ustawić kolory i rozmiary bloków funkcyjnych, zmiennych, połączeń, wejść, wyjść oraz wartości w zależności od typu i formatu liczbowego.



5.4 Commands

Zestaw zaawansowanych komend edycyjnych pozwalających wykonać ciąg pewnych operacji naraz które głównie dotyczą tworzenia złożonych bloków funkcyjnych.

5.4.1 Create device

Tworzy pusty złożony blok funkcyjny typu *Device*. Dalszym etapem jest dodanie wejść, wyjść oraz bloków funkcyjnych które będą wchodzić w skład bloku *Device*. Aby zbudować logikę bloku należy go otworzyć poprzez dwukrotne kliknięcie na nim.

- **Convert to device**

Konwertuje zaznaczone obiekty w złożony blok funkcyjny. Zmienne i bloki typu *Alarm* nie mogą być zamknięte w złożony blok funkcyjny. Jeżeli bloki poddawane konwersji zawierają połączenia zewnętrzne (z blokami spoza zaznaczonych) zostaną utworzone odpowiednie wejścia i wyjścia nowego obiektu. Opisy dla nowych wejść i wyjść mogą automatycznie być kopiowane z bloków połączonych z zewnątrz w tym celu należy zaznaczyć opcję *Commands* → *Copy descriptions to IOs*.

- **Break device**

Operacja odwrotna do opisywanej powyżej. Konwersja wstecz spowoduje że wszystkie bloki funkcyjne zamknięte w bloku *Device* przejdą na poziom wyżej. Przed operacją należy zapewnić wystarczające miejsce w obszarze edycyjnym tak aby bloki zmieściły się nie nachodząc na siebie.

- **Convert to macro**

Konwertuje zaznaczony złożony blok funkcyjny typu *Device* do bloku typu *Macro* i umieszcza go w wybranej bibliotece. Aby konwersja była możliwa należy umieścić w projekcie przynajmniej jedną bibliotekę użytkownika w której znajdzie się nowe *Macro*. Różnica między blokami *Device*, a *Macro* opisana jest w podrozdziale [Złożone bloki funkcyjne typu Macro](#).

- **Break macro**

Operacja odwrotna do opisywanej powyżej. Blok już nie będzie zablokowany i można poddawać go edycji. Konwersja wstecz przekształca blok *Macro* w blok *Device* którego nazwa musi różnić się od nazwy macierzystego bloku gdyż jest to już zupełnie inny, nowy obiekt.

- **Copy descriptions to IOs**

Zaznaczenie opcji spowoduje że operacja *Convert to device* będzie dodatkowo kopiowała opisy z wejść i wyjść zewnętrznych bloków funkcyjnych do połączonych z nimi nowo powstałych wejść i wyjść bloku *Device*.

- **Optimize connection priority**

Operacja próbuje dobrać optymalne wartości priorytetów dla wszystkich połączeń w obszarze edycyjnym. Zmiany priorytetów nie obejmują połączeń wewnątrz aktualnie nie edytowanych bloków *Macro* lub *Device*, czy w obszarach wyższych niż aktualnie edytowany. Oznacza to że operacja działa lokalnie na aktualnie edytowanym poziomie (zagłębieniu) projektu. Aby dokonać optymalizacji połączeń bloku *Device* należy otworzyć go i wtedy wywołać operację optymalizacji. Operacja nie zawsze daje pożądane skutki i może spowodować zmianę funkcjonalności aplikacji PLC, zwłaszcza w miejscach gdzie występują sprzężenia zwrotne. Więcej informacji o priorytetach połączeń znajduje się w rozdziale [4.4 Połączenia](#).

5.5 Project

Operacje dotyczące projektu aplikacji PLC: kompilacja i zapis aplikacji na sterownik PLC.

- Do it All

Wykonuje hurtowo operacje: *Compile* i zaraz po niej *Deploy All*.

- Compile

Kompilacja projekt do kodu binarnego. W zależności od modelu sterownika wykorzystywany jest odpowiedni zewnętrzny kompilator wymagający wcześniejszej instalacji według informacji z rozdziału [2.2 Wymagania oprogramowania](#). Każda zmiana w obszarze edycyjnym polegająca na wstawieniu / usunięciu bloku, zmiennej lub połączenia wymaga przeprowadzenia ponownej kompilacji. Opis procesu kompilacji dla przykładowej aplikacji znajduje się w rozdziale [3.3 Tworzenie przykładowej prostej aplikacji PLC](#).

- CMD compilation (Win8 troubleshooting)

Aktywacja tej opcji spowoduje że proces kompilacji nie będzie odbywał się w tle, a w oddzielnym oknie konsoli cmd. Może to pomóc rozwiązać spotykane problemy kompilacji w systemach nowszych niż Windows 7.

- Upload All

Zapisanie aplikacji PLC oraz wszystkich ustawień na sterownik. Wymagane jest aby sterownik był połączony z komputerem oraz aby zainstalowane było odpowiednie oprogramowanie. Wybór typu połączenia dokonuje się w oknie [Communication Settings](#).

- Upload Variables and HMI menu

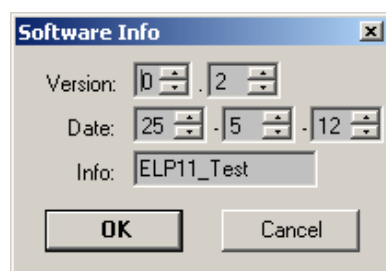
Zapisanie do sterownika tylko listy zmiennych oraz menu HMI. Zmieniając jedynie konfigurację menu HMI w edytorze HMI nie ma potrzeby jeszcze raz zapisywania aplikacji PLC do sterownika, wystarczy tylko zapisać ustawienia co znacznie skraca proces aktualizacji sterownika.

- Restore Variables After Upload

Zaznaczenie opcji spowoduje przywrócenie wartości wszystkich zmiennych z przed procesu aktualizacji aplikacji. Oznacza to że wszystkie nastawy, które zostały przeprowadzone w trakcie działania wcześniejszej wersji aplikacji nie zostaną utracone. Jeżeli kolejność zmiennych (adresy) zostały zmienione, lub na sterownik zapisuje się zupełnie inną aplikację, należy wyłączyć tą opcję gdyż może spowodować zapisanie niewłaściwych nastaw.

- Software Info

Otwiera okno ustawień informacji o aplikacji PLC.



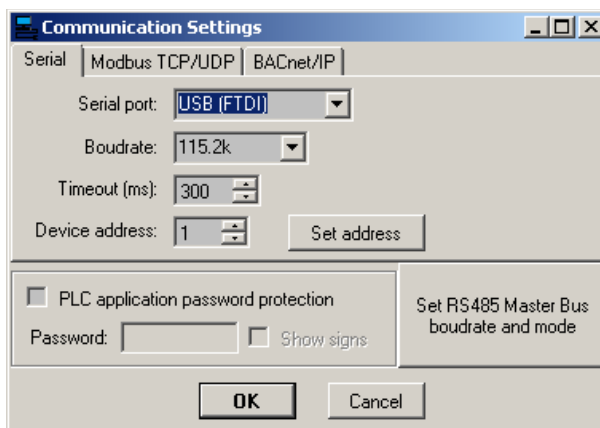
W oknie *Software Info* można ustawić wersję, datę i informację tekstową (11 znaków) dotyczącą wersji aplikacji PLC. Informacje te można później odczytać ze sterownika np. poprzez odpowiedni

element menu HMI *SoftInfo*.

- **Communication Settings**

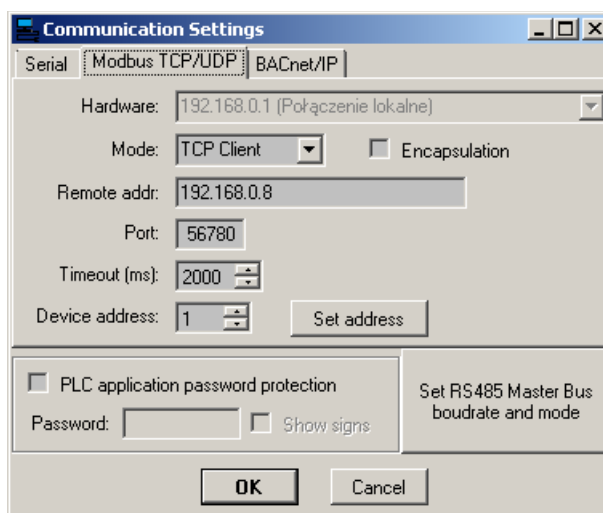
Otwiera okno ustawień komunikacji *Macrocontrol* ze sterownikiem PLC. W zależności od typu komunikacji należy wybrać odpowiednią zakładkę.

Komunikacja szeregową



Na zakładce można ustalić port komunikacyjny gdzie *USB (FTDI)* dotyczy komunikacji ze sterownikami z serii ELP10Txx-yy lub poprzez zewnętrzny konwerter oparty o układ FTDI. Prędkość transmisji jest różna w zależności od typu podłączenia i musi być taka sama jak ustawiona na sterowniku PLC. Opcja *Timeout* ustala czas po jakim sterownik musi wysłać odpowiedź. Im krótszy czas tym szybciej działa komunikacja jednak istnieje większe prawdopodobieństwo powstania błędu. Jeżeli występuje problem w trakcie komunikacji ze sterownikiem należy zwiększyć *Timeout* np. do 500 ms. Adres sterownika ustawiany zazwyczaj na przełącznikach w sterowniku. Do poprawnej komunikacji ze sterownikiem wartość ustawioną na zworkach sterownika musi być taka sama jak w polu *Device address*. Jeżeli sterownik nie posiada zworek adresu to adres można ustalić programowo korzystając z opcji *Set address*.

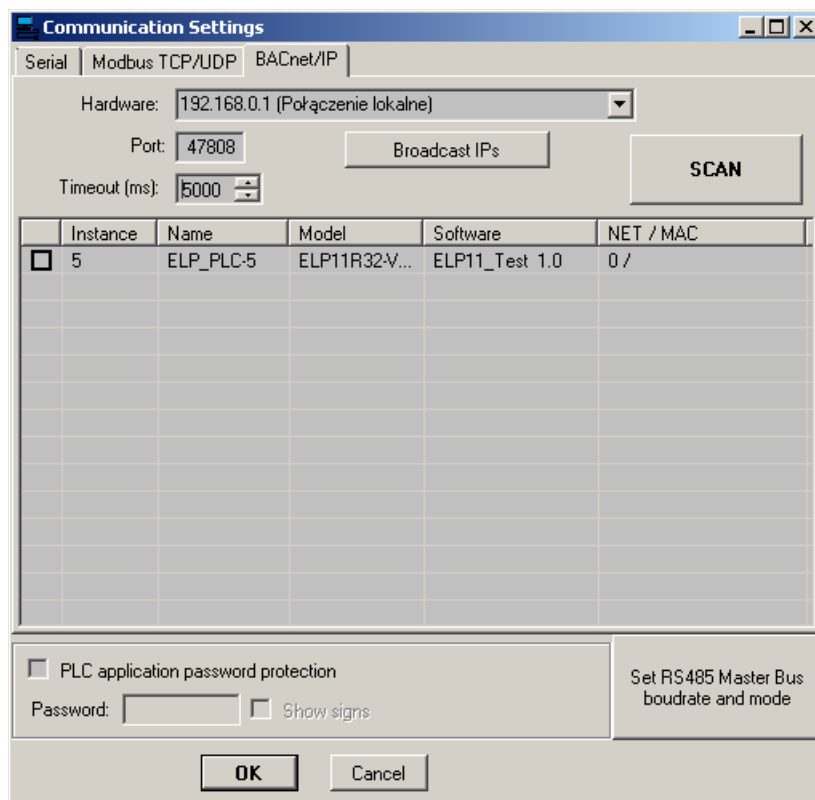
Komunikacja Modbus TCP/UDP



Modele sterowników wyposażone w złącze RJ45 lub zewnętrzny konwerter Ethernet mogą łączyć się bezpośrednio poprzez protokół TCP lub UDP. W zależności od protokołu należy ustalić która

karta sieciowa jest wykorzystywana do połączenia, adres zdalnego urządzenia (może być po domenie), port oraz adres sterownika PLC ustalany na przełącznikach w sterowniku tak samo jak w przypadku połączenia szeregowego. Zaznaczenie opcji *Encapsulation* powoduje że ramka Modbus RTU zostaje opakowana w TCP lub UDP, natomiast brak tej opcji wykorzystuje typowy Modbus TCP.

Komunikacja BACnet / IP



Tylko dla sterowników obsługujących protokołów BACnet. Należy wybrać odpowiednią kartę sieciową oraz port (domyślnie 47808). Przycisk skanowania umożliwia przeszukanie sieci odnajdując wszystkie sterowniki. Można zaznaczyć na które sterowniki ma być zapisywana aplikacja w procesie *Uploadu*.

Ochrona hasłem

Opcja *Software password protection* umożliwia ochronę przed możliwością ponownego zapisu aplikacji PLC do sterownika. Jeżeli hasło zostanie zagubione nie ma możliwości ponownego zaprogramowania sterownika, takie urządzenie musi być zwrócone do firmy EL-Piast w celu odblokowania.

Zmiana prędkości linii RS485 Master

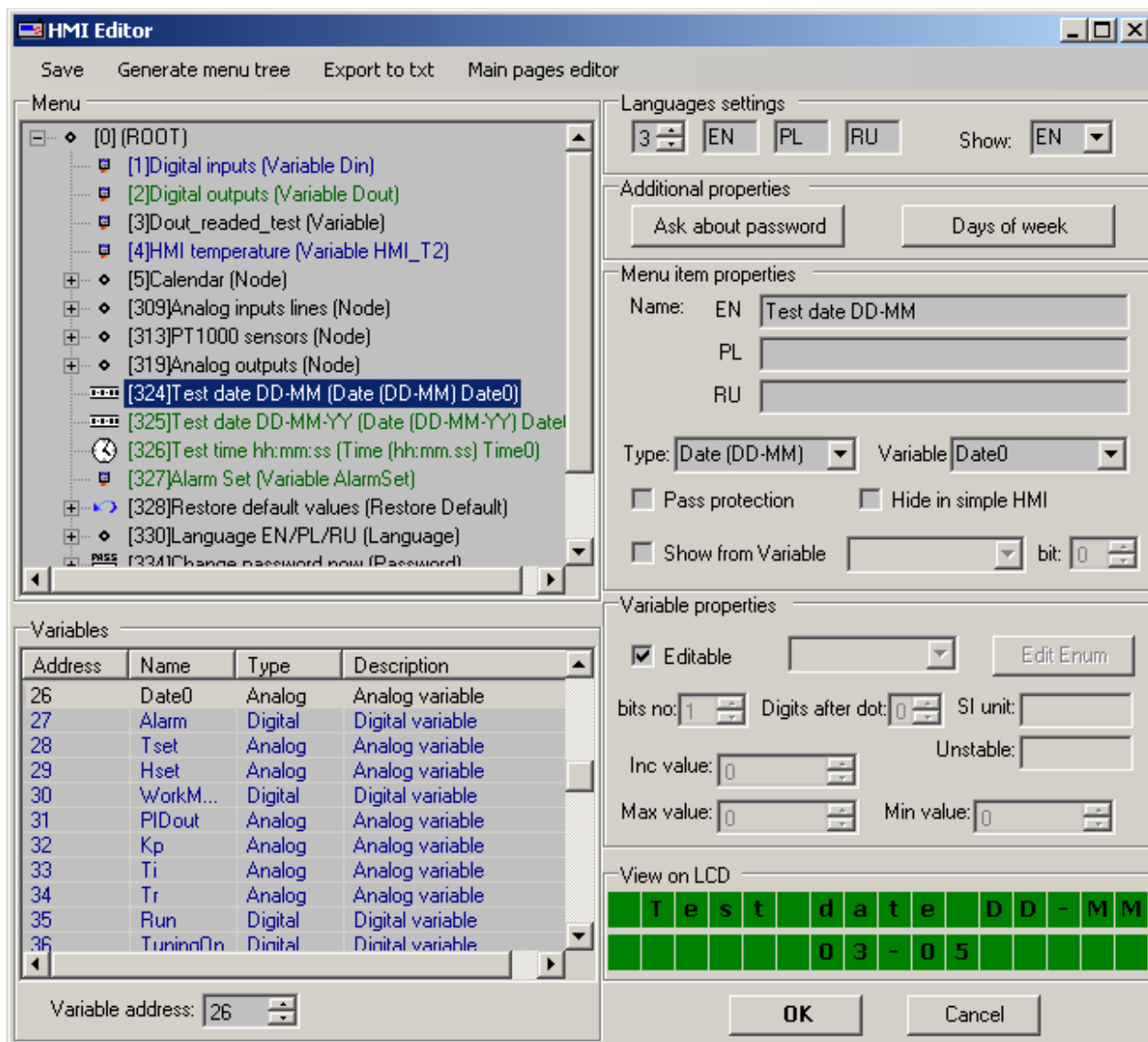
Opcja *Set RS485 Master Bus boudrate and mode* umożliwia zmianę prędkości komunikacji dla linii RS-485 Master zlokalizowanej na większości modeli sterowników ELP.

5.6 Tools

Menu zawierające dodatkowe narzędzia ułatwiające pracę z oprogramowaniem *Macrocontrol*.

- HMI Editor

Otwiera edytor menu HMI. Za pomocą tego narzędzia można zbudować menu HMI które będzie dostępne z poziomu wbudowanego wyświetlacza sterownika, dedykowanych paneli HMI lub webserwera.



Opcje paska Menu

Save – zapis menu HMI bez zamykania edytora.

Generate menu tree – umożliwia wygenerowanie pliku graficznego zawierającego całą strukturę menu HMI.

Export to txt – wygenerowanie pliku txt zawierającego menu, który można wkleić np. do arkusza kalkulacyjnego programu Excel.

Main pages editor – kolejny edytor umożliwiający projektowanie graficznych stron HMI. Opis działania edytora w podrozdziale [Main Pages Editor](#).

Okno Menu

W oknie znajduje się drzewo menu HMI. Można dodawać, odejmować i usuwać elementy menu za pomocą menu kontekstowego, które pojawia się po kliknięciu prawym przyciskiem myszki, lub za pomocą odpowiednich skrótów klawiszowych. Elementy menu można przesuwając poprzez przeciąganie ich w dowolne inne miejsce w strukturze drzewa. Elementu oznaczonego jako *ROOT* nie można usunąć. W oknie przy każdym elemencie wyświetlana jest **ikona typu elementu** (zmiana typu w ustawieniu *Type* w grupie *Menu item properties*), **nazwa elementu** (zmiana w ustawieniu *Name* w grupie *Menu item properties* i zmiana wyświetlanej wersji językowej w ustawieniu *Show* w grupie *Languages settings*) oraz **typ elementu**, a w przypadku zmiennej dodatkowo nazwa zmiennej. Po wybraniu konkretnego elementu menu po prawej stronie w grupie *Menu item properties* oraz, jeżeli element menu dotyczy zmiennej, w grupie *Variable properties* pojawiają się odpowiednie parametry elementu.

Okno Zmiennych

Zawiera listę wszystkich zmiennych zawartych w aplikacji PLC z informacją o ich adresie, nazwie, typie oraz opisie. Jeżeli element menu jest typu zmienna (*Variable*) to w polu *Variable* w grupie *Menu item properties* można do niego przypisać dowolny zmienną z listy zmiennych. Przypisanie zmiennej do elementu menu można wykonać na trzy sposoby:

- wybrać odpowiednią zmienną z listy w polu *Variable* w grupie *Menu item properties*
- przeciągnąć wybraną zmienną z okna zmiennych do pola *Variable* w grupie *Menu item properties*
- przeciągnąć wybraną zmienną z okna zmiennych do wybranego elementu menu w oknie *Menu*.

Grupa ustawień językowych

Umożliwia ustawienie liczby wersji językowych (1 – 3) oraz oznaczenie każdej z wersji językowej symbolem identyfikującym. Oznaczenia te są domyślnymi pod-elementami elementu menu typu *Language*, który służy do zmiany wersji językowej. W polu *Show* można wybrać która wersja językowa ma być bieżąco wyświetlana w oknie *Menu*.

Dodatkowe ustawienia

Przycisk *Ask about password* służy do ustawienia komunikatu będącego prośbą o podanie hasła z poziomu panelu HMI gdy użytkownik próbuje dostać się do elementów chronionych hasłem.

Przycisk *Days of week* służy do ustawienia nazw dla dni tygodnia wykorzystywanych w opcjach kalendarza i przy edycji daty.

Grupa właściwości elementu menu

W grupie *Menu item properties* wyświetlane są właściwości aktualnie wybranego elementu menu z okna *Menu*. W polu *Name* można ustawić nazwę elementu menu. W polu *Type* ustawia się typ elementu menu. Pole *Variable* pozwala wybrać zmienną przypisaną do elementu menu pod warunkiem że element menu jest zmienną. Dodatkowo można ustawić czy element menu ma być chroniony hasłem oraz czy element powinien być ukryty prostej wersji panelu HMI. Ochrona hasłem ma sens jedynie dla elementów menu typu *Node* które wewnątrz posiadają pod-menu. Wtedy wejście

do pod-menu wygeneruje komunikat z prośbą o podanie hasła który jest wcześniej ustalany w opcji *Ask about password*. Domyślne hasło: *1111* można zmienić z poziomu panelu HMI elementem menu typu *Password*. Podanie prawidłowego hasła w panelu HMI umożliwia wejście w głąb menu. Hasło jest aktywne przez okres 3 minut. Jeżeli użytkownik w panelu HMI porusza się po obszarze menu chronionym hasłem czas aktywności nie jest liczony. Dopiero po opuszczeniu obszaru chronionego zegar odlicza 3 minuty po upływie których ponowne wejście do menu chronionego hasłem spowoduje wygenerowanie komunikatu z prośbą o ponowne podanie hasła. Gdy hasło zostanie zapomniane istnieje procedura przywrócenia hasła domyślnego opisana w rozdziale [6 Przywrócenie sterownika](#). Zaznaczenie opcji *Hide in simple HMI* umożliwia ukrycie elementu menu przed użytkownikiem korzystającym z prostej wersji HMI. Panele HMI posiadają wewnątrz zworkę konfigurującą tryb pracy panelu. Opcja przydatna do ukrycia opcji serwisowych do których końcowy użytkownik nie powinien mieć wglądu.

Opcja *Show from Variable* pozwala wybrać bit zmiennej odpowiedzialny za pokazanie wybranego elementu menu. Jeżeli ta opcja zostanie zaznaczona to aby element był widoczny na panelu HMI konieczne jest zapisanie wybranego bitu zmiennej na wartość 1.

Typy elementu menu

<i>Node</i>	Węzeł. Element pośredni określa nazwę dla podmenu.
<i>Communicate</i>	Komunikat. Element informacyjny.
<i>Variable</i>	Zmienna. Prezentuje stan zmiennej według ustalonych parametrów w grupie <i>Variable properties</i> .
<i>Time (RTC)</i>	Czas wewnętrznego zegara RTC sterownika PLC. Umożliwia edycję lokalnego czasu sterownika.
<i>Date (RTC)</i>	Data wewnętrznego zegara RTC sterownika PLC.
<i>SoftInfo</i>	Informacja o oprogramowaniu (aplikacji PLC). Prezentacja danych ustalanych na karcie <i>Project</i> → <i>Software Info</i> opis w rozdziale Software Info .
<i>Language</i>	Zmiana wersji językowej menu. Element automatycznie generuje pod-elementy dla wszystkich wersji językowych.
<i>Password</i>	Zmiana hasła. Element automatycznie generuje pod-elementy zawierające prośby o podanie starego hasła, a następnie dwukrotnie nowego hasła. Domyślne hasło to <i>1111</i> i służy do ochrony elementów menu oznaczonych jako <i>Pass protection</i> .
<i>Restore Default</i>	Przywrócenie ustawień domyślnych. Element automatycznie generuje pod-element z prośbą o potwierdzenie. Stany wszystkich zmiennych zostaną przywrócone do nastaw domyślnych.
<i>Calendar</i>	Kalendarz. Automatycznie generuje strukturę kalendarza. W kolejnym kroku należy wybrać zmienną kalendarzową której dotyczy element menu.
<i>Date (DD-MM)</i>	Data w formacie DD-MM. Specjalny format prezentacji zmiennej.
<i>Date (DD-MM-YY)</i>	Data w formacie DD-MM-YY. Specjalny format prezentacji zmiennej.
<i>Time (hh:mm:ss)</i>	Czas w formacie hh:mm:ss. Specjalny format prezentacji zmiennej.

Grupa właściwości zmiennej

Grupa pojawia się gdy aktualnie zaznaczony element menu w oknie *Menu* jest zmienną. Można zaznaczyć zmienną jako edytowalną, wtedy możliwa jest zmiana jej wartości z poziomu panelu HMI. Można także wybrać w jaki sposób mają być prezentowane wartości przyjmowane przez zmienną.

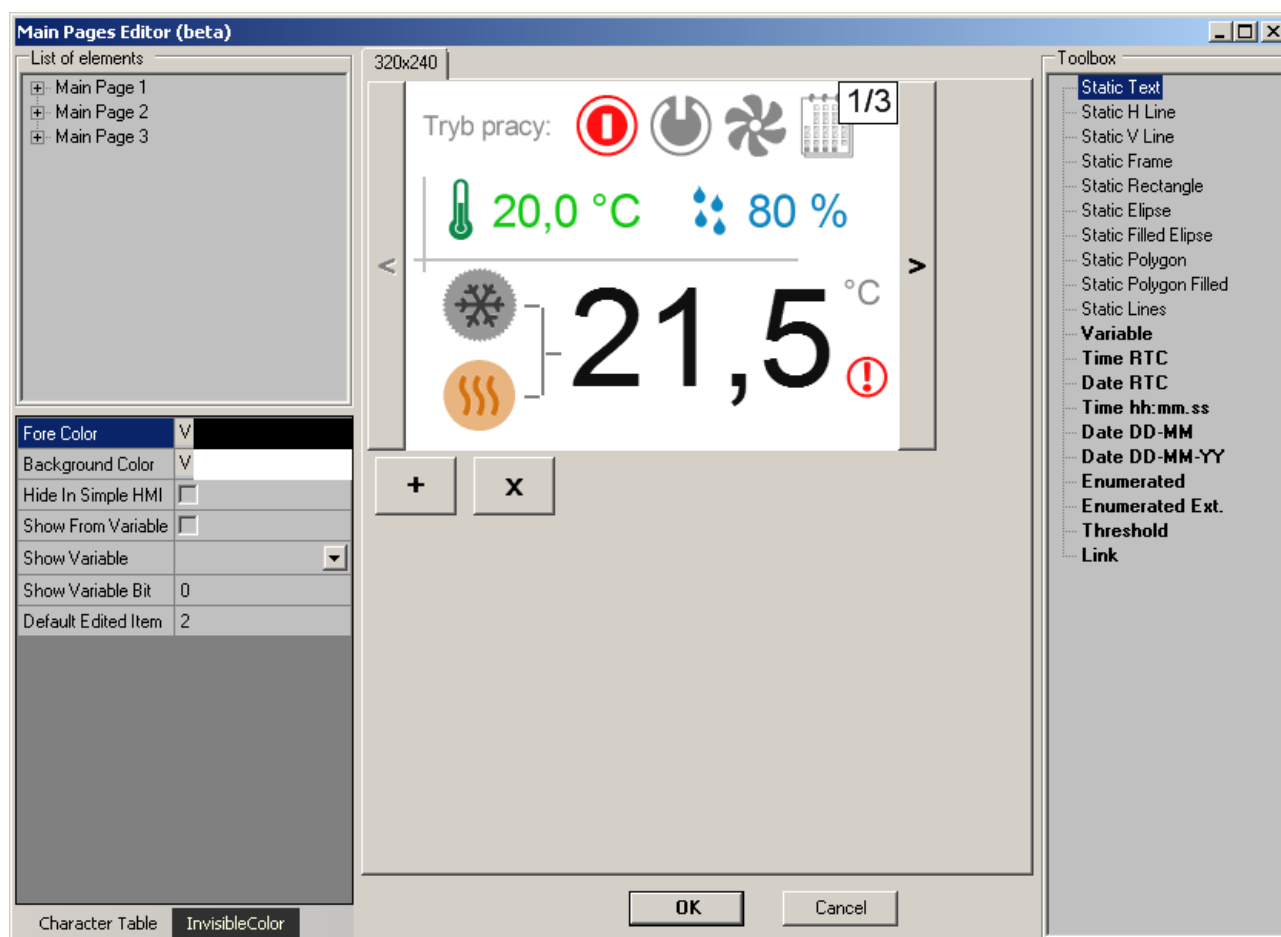
Można wybrać spośród 6 formatów prezentacji:

- Fixed* Zmienna przedstawiona jako stałoprzecinkowa zgodnie z formatem zmiennych opisanym w [4.1 Reprezentacje liczbowe](#). Ten format reprezentacji umożliwia ustalenie ilości wyświetlanych miejsc po przecinku, jednostki SI dla wartości, a także komunikat jaki ma być wyświetlony w przypadku wartości nieokreślonej np. brak podłączonego czujnika temperatury. Możliwe jest również ustalenie w jakim zakresie zawierają się wartości zmiennej oraz jaki jest krok zmiany wartości.
- Hex* Zmienna przedstawiona jako hexadecymalna.
- Decimal* Zmienna przedstawiona jako dziesiętna całkowita.
- Enum* Zmienna przedstawiona jako tekst odpowiadający konkretnej wartości liczbowej. Ustalenie nazw dla poszczególnych wartości odbywa się w dodatkowym oknie otwieranym po kliknięciu na przycisk *Edit Enum*.
- RadioBoxes* Radiobox to rząd okręgów reprezentujących pojedyncze bity zmiennej. Ilość bitów ustala się w polu *bits no*. Cechą charakterystyczną radioboxów jest możliwość zaznaczenia tylko jednego bitu.
- CheckBoxes* Prostokąty reprezentujące pojedyncze bity zmiennej. Ilość bitów ustala się w polu *bits no*. W odróżnieniu od Radioboxów można zaznaczać dowolną ilość bitów naraz.

▪ Main Pages Editor

Edytor głównych stron HMI. Można go uruchomić z poziomu HMI Edytora. W tej chwili wszystkie graficzne HMI firmy EL-Piast bazują na kolorowych wyświetlaczach o rozdzielczości 320x240 pikseli i tylko dla takiej rozdzielczości można edytować strony. Konfiguracja głównych stron zapisywana jest w pliku *MainConf.xml* w folderze aktualnego projektu. Plik ten można skopiować do innego projektu podobnie jak plik konfiguracji menu HMI *CtrlConf.xml*.

Przykład z folderu Macrocontrol / Examples / ELP11_Test zawiera przykładowe główne strony HMI. Każda zawiera inny sposób prezentowania elementów.



Okno edytora zawiera:

List of elements – listę aktualnie utworzonych elementów graficznych podzielonych na strony. Każda ze stron zawiera elementy ponumerowane od 0. Numer elementu jest istotny przy ustalaniu domyślnie edytowanego elementu strony (*Default Edited Item*) we właściwościach strony. Numer domyślnie edytowanego elementu musi wskazywać na element edytowalny. Jeżeli strona ma nie zawierać domyślnie edytowanego elementu wartość *Default Edited Item* powinna być równa 255.

Właściwości wybranego elementu graficznego – okno zawiera informacje określające zaznaczony element graficzny. Są one zależne od typu elementu graficznego, a ich opis znajduje się w dalszej części tego podrozdziału

Character Table – umożliwi utworzenie okna zawierającego tabelę znaków HMI. Jeżeli w oknie właściwości, w polu *Text* zostanie umieszczony kursor edycji można klikając na odpowiednie znaki tabeli wpisywać odpowiadające im kody.

Invisible Color – ustawienie koloru elementów graficznych które na ekranie w danej chwili nie są widoczne. Dzięki temu można wiedzieć gdzie jest ich pozycja i że w ogóle się znajdują w edytorze.

Okno wyświetlacza – zawiera widok aktualnie edytowalnej strony. Można zaznaczać i przesuwać dowolne elementy graficzne.

Toolbox – zawiera listę elementów graficznych które można umieścić na wyświetlaczu. Elementy wprowadza się na ekran poprzez przeciągnięcie wybranego elementu w okno wyświetlacza. Niektóre elementy nie można jeszcze używać, będą one dostępne w najbliższym czasie.

Nawigacja w edytorze

Poszczególne elementy graficzne można zaznaczyć klikając na nie lewym klawiszem myszki. Przytrzymując klawisz *Shift* można zaznaczyć lub odznaczyć więcej niż jeden element. Przytrzymując klawisz *Alt* zaznacza się całą grupę elementów takich jak *Static Group* lub *Enumerated*. Przesuwanie elementów graficznych można wykonać myszką przytrzymując lewy klawisz, lub precyzyjnie, kursorami na klawiaturze. Zaznaczanie elementów można również wykonać w oknie *List of elements*. Również w tym oknie można przeciągając zmienić miejsce elementu w hierarchii, lub przenieść element na inną stronę lub do innej grupy. Usuwanie zaznaczonych elementów uzyskuje się klawiszem *Del* lub z menu kontekstowego (prawy klawisz myszki).

Hierarchia i numeracja elementów graficznych

Kolejność elementów w drzewie w oknie *List of elements* nie jest przypadkowa. W takiej właśnie kolejności będą one generowane na ekranie HMI. Należy o tym pamiętać jeżeli któryś z elementów ma zasłaniać inny lub na niego nachodzić. Również numer każdego z elementów jest istotny dla ustalenia domyślnie edytowalnego elementu strony.

Typy elementów graficznych

- **Static Group**

Wszystkie elementy których nazwa rozpoczyna się od *Static* musi występować w grupie *Static Group* lub *Enumerated*. Jeżeli dowolny element *Static* zostanie przeciągnięty do okna wyświetlacza to automatycznie w oknie *List of elements* zostanie utworzona dla niego nowa grupa. Elementy *Static* można przeciągając grupować razem w jednej grupie. Jeżeli grupa nie zawiera już żadnego elementu statycznego to automatycznie zostanie usunięta.

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X, Y</i>	Pozycja na ekranie HMI całej grupy
<i>Hide In Simple HMI</i>	Umożliwia ukrycie grupy dla uproszczonego HMI (wybór typu odbywa się zworą Simple / Ext. w HMI)
<i>Show From Variable</i>	Zaznaczenie tej opcji spowoduje że grupa będzie widoczna tylko gdy bit odpowiedniej zmiennej będzie ustawiony
<i>Show Variable</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Wybór zmiennej sterownika która ma wpływać na wyświetlanie grupy

Show Variable Bit Tylko gdy opcja *Show From Variable* jest zaznaczona. Bit zmiennej *Show Variable*.

- **Static Text**

Dowolny tekst zawierający znaki z tabeli *Character Table*, który może występować w trzech wersjach językowych.

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	W zależności od <i>TextAlign</i> jest to pozycja horyzontalna lewej, środka, lub prawej krawędzi elementu.
<i>Y</i>	Pozycja wertykalna górnej krawędzi elementu.
<i>TextAlign</i>	Wyrównanie tekstu, istotne przy kliku wersjach językowych. Gdy następuje zmiana języka tekst może być innej długości i może nachodzić na inny element.
<i>Font Scale Factor</i>	Rozmiar czcionki. Wartość określa stopień zmniejszenia czcionki. Wartość 1 określa czcionkę w oryginalnym rozmiarze. Wartość 5 oznacza czcionkę 5-krotnie pomniejszoną.
<i>Font Color</i>	Kolor znaków
<i>Background Color</i>	Kolor tła tekstu
<i>Language</i>	Występowanie wersji językowych. Jeżeli opcja nie jest zaznaczona to zawsze będzie wyświetlany tekst z <i>Text1</i>
<i>Current Language</i>	Aktualnie wyświetlana wersja językowa. Tylko dla testu wyświetlania
<i>Text1, Text2, Text3</i>	Tekst w trzech wersjach językowych. Wstawianie znaków specjalnych można wykonać z pomocą tabeli <i>Character Table</i>

- **Static H Line**

Linia horyzontalna (pozioma)

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X, Y</i>	Pozycja lewego punktu początkowego linii
<i>Length</i>	Długość linii
<i>Thickness</i>	Grubość linii.

- **Static V Line**

Linia wertykalna (pionowa)

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X, Y</i>	Pozycja górnego punktu początkowego linii
<i>Length</i>	Długość linii
<i>Thickness</i>	Grubość linii.

- **Static Frame**

Prostokątna ramka

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X, Y</i>	Pozycja lewego górnego rogu ramki
<i>Width, Height</i>	Wymiary ramki
<i>Thickness</i>	Grubość ramki.
<i>Color</i>	Kolor ramki

- **Static Rectangle**

Prostokąt z wypełnieniem

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X, Y</i>	Pozycja lewego górnego rogu prostokąta
<i>Width, Height</i>	Wymiary prostokąta
<i>Frame Thickness</i>	Grubość ramki.
<i>Frame Color</i>	Kolor ramki
<i>Background Color</i>	Kolor wypełnienia prostokąta

- **Static Ellipse**

Elipsa (obrys)

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X, Y</i>	Pozycja punktu centralnego elipsy
<i>Horizontal Radius</i>	Promień elipsy
<i>Vertical Radius</i>	Promień elipsy
<i>Thickness</i>	Grubość obrysu elipsy
<i>Color</i>	Kolor obrysu elipsy

- **Static Filled Ellipse**

Wypełniona elipsa

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X, Y</i>	Pozycja punktu centralnego elipsy
<i>Horizontal Radius</i> <i>Vertical Radius</i>	Promień elipsy
<i>Contour Thickness</i>	Grubość obrysu elipsy
<i>Contour Color</i>	Kolor obrysu elipsy
<i>Fill Color</i>	Kolor wypełnienia elipsy

- **Static Polygon**

Wielokąt (obrys)

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	Minimalna pozycja X z wszystkich punktów wielokąta
<i>Y</i>	Minimalna pozycja Y z wszystkich punktów wielokąta
<i>Points</i>	Kolejne punkty wielokąta (maksymalnie 32 punkty)
<i>Thickness</i>	Grubość obrysu wielokąta
<i>Color</i>	Kolor obrysu wielokąta

- **Static Filled Polygon**

Wypełniony wielokąt

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	Minimalna pozycja X z wszystkich punktów wielokąta
<i>Y</i>	Minimalna pozycja Y z wszystkich punktów wielokąta
<i>Points</i>	Kolejne punkty wielokąta (maksymalnie 32 punkty)
<i>Contour Thickness</i>	Grubość obrysu wielokąta
<i>Contour Color</i>	Kolor obrysu wielokąta
<i>Fill Color</i>	Kolor wypełnienia wielokąta

- **Static Lines**

Ciąg połączonych linii

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	Minimalna pozycja X z wszystkich punktów
<i>Y</i>	Minimalna pozycja Y z wszystkich punktów
<i>Points</i>	Kolejne punkty linii (maksymalnie 32 punkty)
<i>Thickness</i>	Grubość linii
<i>Color</i>	Kolor linii

- **Variable**

Zmienna wyświetlana w postaci liczby

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	W zależności od <i>TextAlign</i> jest to pozycja horyzontalna lewej, środka, lub prawej krawędzi elementu.
<i>Y</i>	Pozycja wertykalna górnej krawędzi elementu.
<i>Hide In Simple HMI</i>	Umożliwia ukrycie zmiennej dla uproszczonego HMI (wybór typu odbywa się zworką Simple / Ext. w HMI)
<i>Show From Variable</i>	Zaznaczenie tej opcji spowoduje że zmienna będzie widoczna tylko gdy bit odpowiedniej zmiennej będzie ustawiony
<i>Show Variable</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Wybór zmiennej sterownika która ma wpływać na wyświetlanie tej zmiennej.
<i>Show Variable Bit</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Bit zmiennej <i>Show Variable</i> .
<i>TextAlign</i>	Wyrównanie wyświetlanej wartości zmiennej.
<i>Font Scale Factor</i>	Rozmiar czcionki. Wartość określa stopień zmniejszenia czcionki. Wartość 1 określa czcionkę w oryginalnym rozmiarze. Wartość 5 oznacza czcionkę 5-krotnie pomniejszoną.
<i>Font Color</i>	Kolor znaków
<i>Background Color</i>	Kolor tła zmiennej
<i>Variable</i>	Wybór zmiennej która ma być przypisana do elementu graficznego i której wartość ma być wyświetlana
<i>Visual</i>	Sposób wizualizacji wartości zmiennej
<i>Digits After Dot</i>	Ilość miejsc po przecinku (tylko dla <i>Visual = Fixed</i>)
<i>SI Unit</i>	Jednostka SI przypisana do zmiennej (można używać znaków z tabeli <i>Character Table</i>)

<i>Unstable</i>	Tekst wyświetlany dla wartości nieustalonej (np. gdy brak podłączonego czujnika temperatury)
<i>Editable</i>	Czy zmienna może być edytowalna
<i>Min. Value</i>	Minimalna wartość jaką można ustawić (tylko gdy zmienna edytowalna)
<i>Max. Value</i>	Maksymalna wartość jaką można ustawić (tylko gdy zmienna edytowalna)
<i>Inc. Value</i>	Krok zwiększania lub zmniejszania zmiennej (tylko gdy zmienna edytowalna)
<i>Test Value</i>	Testowa wartość, tylko w celu poglądowym

- **Time RTC / Date RTC**

Aktualny czas lub data z zegara RTC sterownika

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	W zależności od <i>TextAlign</i> jest to pozycja horyzontalna lewej, środka, lub prawej krawędzi elementu.
<i>Y</i>	Pozycja wertykalna górnej krawędzi elementu.
<i>Hide In Simple HMI</i>	Umożliwia ukrycie elementu dla uproszczonego HMI (wybór typu odbywa się zworką Simple / Ext. w HMI)
<i>Show From Variable</i>	Zaznaczenie tej opcji spowoduje że element będzie widoczny tylko gdy bit odpowiedniej zmiennej będzie ustawiony
<i>Show Variable</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Wybór zmiennej sterownika która ma wpływać na wyświetlanie tego elementu.
<i>Show Variable Bit</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Bit zmiennej <i>Show Variable</i> .
<i>TextAlign</i>	Wyrównanie wyświetlanego tekstu.
<i>Font Scale Factor</i>	Rozmiar czcionki. Wartość określa stopień zmniejszenia czcionki. Wartość 1 określa czcionkę w oryginalnym rozmiarze. Wartość 5 oznacza czcionkę 5-krotnie pomniejszoną.
<i>Font Color</i>	Kolor znaków
<i>Background Color</i>	Kolor tła zmiennej
<i>Editable</i>	Czy element może być edytowany

- **Time hh:mm:ss / Date DD-MM / Data DD-MM-YY**

Zmienna przedstawiona w formacie czasu, lub daty

Nazwa	Opis
-------	------

<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	W zależności od <i>TextAlign</i> jest to pozycja horyzontalna lewej, środka, lub prawej krawędzi elementu.
<i>Y</i>	Pozycja wertykalna górnej krawędzi elementu.
<i>Hide In Simple HMI</i>	Umożliwia ukrycie elementu dla uproszczonego HMI (wybór typu odbywa się zworą Simple / Ext. w HMI)
<i>Show From Variable</i>	Zaznaczenie tej opcji spowoduje że element będzie widoczny tylko gdy bit odpowiedniej zmiennej będzie ustawiony
<i>Show Variable</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Wybór zmiennej sterownika która ma wpływać na wyświetlanie tego elementu.
<i>Show Variable Bit</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Bit zmiennej <i>Show Variable</i> .
<i>TextAlign</i>	Wyrównanie wyświetlanego tekstu.
<i>Font Scale Factor</i>	Rozmiar czcionki. Wartość określa stopień zmniejszenia czcionki. Wartość 1 określa czcionkę w oryginalnym rozmiarze. Wartość 5 oznacza czcionkę 5-krotnie pomniejszoną.
<i>Font Color</i>	Kolor znaków
<i>Background Color</i>	Kolor tła zmiennej
<i>Editable</i>	Czy element może być edytowany
<i>Variable</i>	Wybór zmiennej która ma być przypisana do elementu graficznego i której wartość ma być wyświetlana
<i>Test Value</i>	Testowa wartość, tylko w celu poglądowym

- **Enumerated**

Zmienna przedstawiona jako zestaw elementów typu *Static* różny dla każdej wartości. Wybór wyświetlanego elementu odbywa się na podstawie wartości przypisanej zmiennej.

Nazwa	Opis
<i>Name</i>	Nazwa elementu. Tylko w celu informacyjnym dla edycji. Nieprzesyłana do HMI
<i>X</i>	W zależności od <i>TextAlign</i> jest to pozycja horyzontalna lewej, środka, lub prawej krawędzi elementu.
<i>Y</i>	Pozycja wertykalna górnej krawędzi elementu.
<i>Hide In Simple HMI</i>	Umożliwia ukrycie elementu dla uproszczonego HMI (wybór typu odbywa się zworą Simple / Ext. w HMI)
<i>Show From Variable</i>	Zaznaczenie tej opcji spowoduje że element będzie widoczny tylko gdy bit odpowiedniej zmiennej będzie ustawiony
<i>Show Variable</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Wybór

	zmiennej sterownika która ma wpływać na wyświetlanie tego elementu.
<i>Show Variable Bit</i>	Tylko gdy opcja <i>Show From Variable</i> jest zaznaczona. Bit zmiennej <i>Show Variable</i> .
<i>Variable</i>	Wybór zmiennej która ma być przypisana do elementu graficznego i której wartość ma być wyświetlana
<i>Editable</i>	Czy element może być edytowany
<i>Values</i>	Kolejne wartości jakie może przyjąć zmienna (typu <i>Decimal</i>)
<i>Selection</i>	Sposób zaznaczania wybranego elementu <i>Static</i> : <i>Only Selected</i> – tylko aktualnie wybrany element <i>Static</i> jest wyświetlany, <i>Selected & Prev</i> – wybrany element oraz wszystkie wcześniejsze są wyświetlane, <i>Sel. Color</i> – wybrany element ma inny ustalony poniżej kolor który będzie taki sam dla wybranego elementu, natomiast niewybrane elementy wyświetlane są według własnego indywidualnie ustalanego koloru, <i>Nonsel. Color</i> – niewybrane elementy mają inny jeden ustalony poniżej kolor, natomiast wybrany element wyświetlany jest według własnego indywidualnie ustalanego koloru, <i>Next Color</i> – wybrany element oraz wszystkie wcześniejsze są wyświetlane według swojego indywidualnie ustalonego koloru, natomiast niewybrane elementy mają inny, jeden ustalony poniżej kolor.
<i>Fore Color</i>	Kolor znaków (tylko dla <i>Selection</i> działające według koloru)
<i>Back Color</i>	Kolor tła
<i>Blinking</i>	Jak element ma się zachowywać w trakcie edycji: <i>All</i> – migają wszystkie elementy <i>Selected</i> – miga tylko aktualnie wybrany <i>Non-selected</i> – migają tylko niewybrane elementy
<i>Test Value</i>	Testowa wartość, tylko w celu poglądowym

Pozostałe elementy graficzne jeszcze nie są obsługiwane.

- BACnet Object Editor

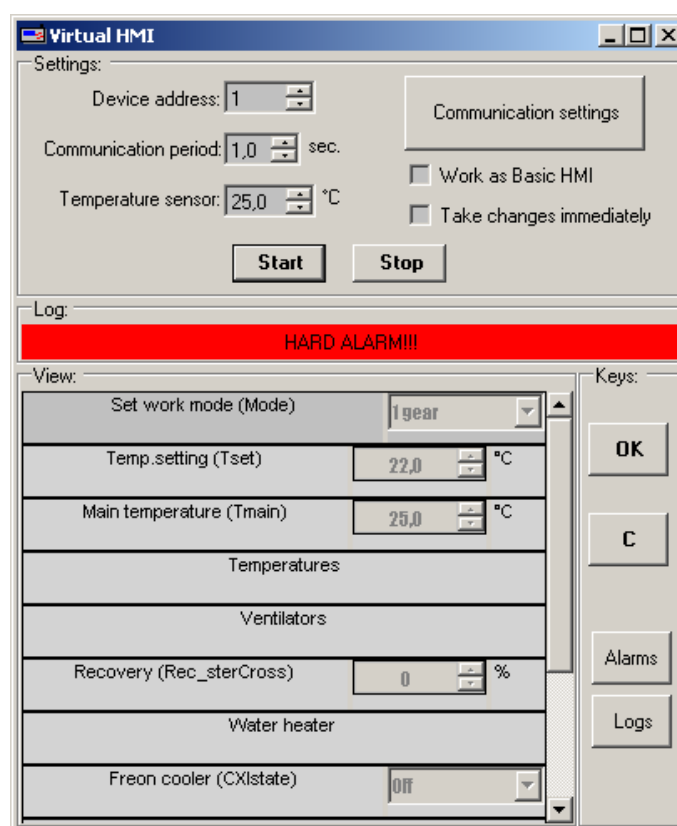
Edytor obiektów BACnet. W oknie *BACnet Objects Editor* można definiować zmienne jako obiekty protokołu BACnet. W zależności od typu obiektu dostępne są dodatkowe opcje konfiguracyjne. Obiekty zaznaczone jako *BACnet available* będą widoczne na liście zmiennych razem z obiektami reprezentującymi wszystkie wejścia i wyjścia sterownika PLC. Konfiguracja obiektów BACnet zapisywana jest na sterowniku wraz z aplikacją. Zmiana konfiguracji wiąże się z ponownym zaprogramowaniem sterownika (wystarczy zapisać zmienne i konfiguracje HMI).

- BACnet Integrator

Narzędzie pozwalające dokonać integracji sieci sterowników BACnet. Po otwarciu okna następuje przeszukanie sieci i wyświetlenie wszystkich dostępnych urządzeń wraz z ich obiektami. W sterownikach ELP można nadpisywać wartości dowolnych obiektów wartościami obiektów z innych urządzeń w sieci tworząc powiązania między nimi. Narzędzie *BACnet Integrator* jest w trakcie rozwoju i będzie podlegało znacznym modyfikacjom i usprawnieniom. Na tą chwilę jego opis jest pobieżny.

- Virtual HMI

Wirtualny panel HMI umożliwia podgląd i zmianę nastaw parametrów aplikacji PLC za pomocą menu utworzonego wcześniej w edytorze HMI.

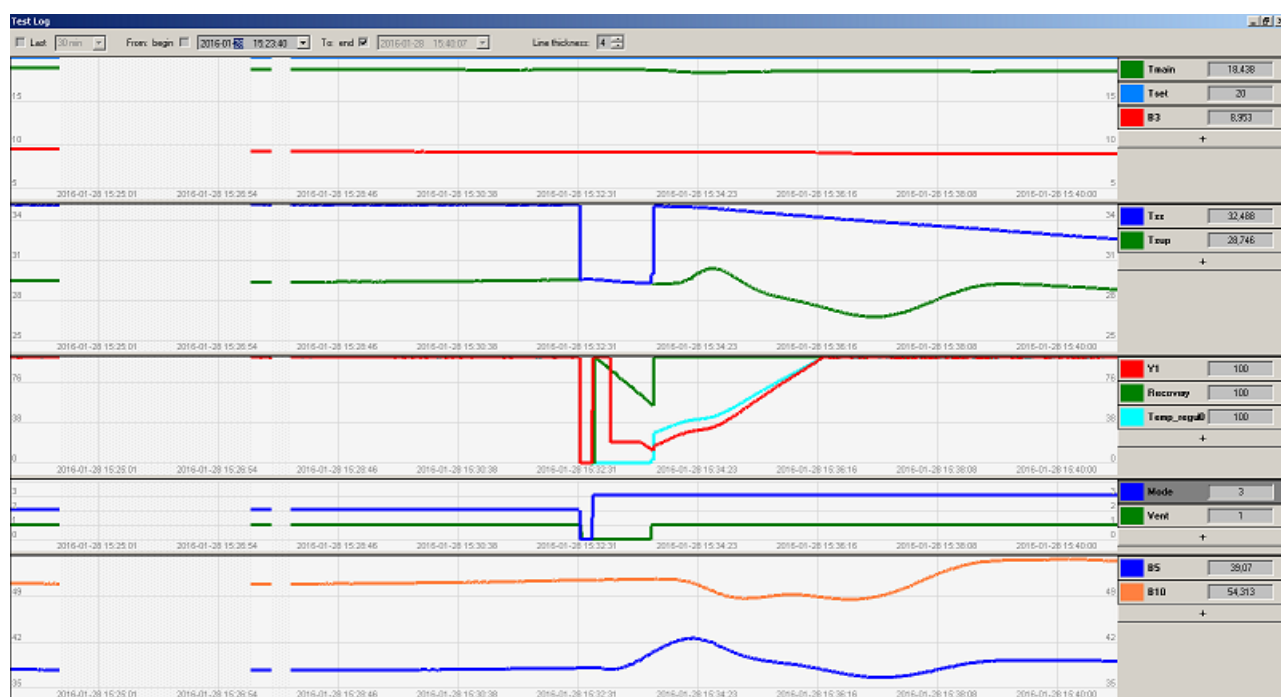


Wirtualny panel HMI demonstruje całe menu które potem można zobaczyć na zewnętrznych panelach HMI podłączonych do sterownika. Do poprawnego działania wymagane jest podłączenie

do komputera zaprogramowanego sterownika i ustawienie odpowiedniego typu połączenia w oknie *Communication settings*. Opcje programu umożliwiają nastawę adresu sterownika, częstotliwości komunikacji, symulowaną wartość temperatury na czujniku w panelu HMI. Można również ustalić tryb działania panelu jako wersja uproszczona (nie wszystkie opcje menu są dostępne w zależności od konfiguracji *Hide in simple HMI* w menu HMI) oraz czy efekt zmian wartości zmiennych powinien być zapisywany w sterowniku natychmiastowo czy dopiero po zatwierdzeniu klawiszem *OK*. Komunikację uruchamia się przyciskiem *Start*, a zatrzymuje przyciskiem *Stop*. Na pasku *Log* pojawiają się błędy komunikacji oraz stany alarmowe sterownika PLC. W okienku *View* wyświetlane są elementy menu. Aby dokonać edycji bądź wejść w kolejne zagłębienie menu należy kliknąć dwukrotnie na wybrany element. Po zmianie wartości należy ją zatwierdzić przyciskiem *OK* lub zrezygnować przyciskiem *C*, wtedy program przywróci wartość zmiennej sprzed edycji. Przycisk *Alarms* umożliwia wejście w menu alarmów. Przycisk *Logs* demonstruje ostatnie logi sterownika wykorzystywane do celów diagnostycznych.

▪ In-Target Test

Tryb testowania działania aplikacji sterownika. Dodatkowo narzędzie pozwala prezentować dane logowane w trakcie trwania testu. Jeżeli opcja *Open Test Log Window* jest zaznaczona to przy każdorazowym uruchomieniu trybu testowania okno *Test Log* zostanie również uruchomione.



W oknie *Test Log* można utworzyć wiele wykresów, zawierających dane z różnych zmiennych zawartych w aplikacji aktualnie otwartego projektu. Dodawanie kolejnych wykresów odbywa się poprzez kliknięcie podłużnego przycisku + w dolnej części okna. Wysokość każdego wykresu można dowolnie zmieniać. Dodawanie zmiennych wykresu odbywa się poprzez kliknięcie przycisku + po prawej stronie wykresu. Należy wybrać zmienną z listy dostępnych poprzez dwukrotne kliknięcie. Dla każdej zmiennej można ustalić kolor kreślonego wykresu, jak również zmieniać jej wartość w trakcie trwania testu (wpisanie nowej wartości potwierdzone klawiszem *Enter*). Usuwanie wybranej zmiennej z wykresu odbywa się poprzez wybranie opcji *Delete* z menu kontekstowego. W podobny sposób można usunąć cały wykres.

6. Przywrócenie sterownika

Istnieje możliwość zawieszenia się sterownika wskutek zapisu niepoprawnej / błędnie skompilowanej aplikacji PLC. W takim wypadku sterownik przestaje reagować i nie ma możliwości skomunikowania się z nim, a co za tym idzie, zapisu innej aplikacji sterującej. W celu przywrócenia sterownika i umożliwienia ponownego zapisu oprogramowania należy:

- wyłączyć sterownik z zasilania
- Przytrzymując przycisk *Reset* włączyć sterownik. Jeżeli sterownik nie posiada specjalnego przycisku *Reset* należy ustawić wszystkie zworki adresu (od spodu sterownika) na maksymalną wartość (szesnastkowo 0xFF) i wtedy włączyć sterownik
- Jeżeli zworki adresu zostały przestawione należy w środowisku *Macrocontrol* w ustawieniach *Project* → *Communication settings* wprowadzić nowy adres sterownika (255)
- jeżeli zapisano błędną aplikację PLC należy zapisać nową, poprawną wersję poprzez wybranie opcji *Project* → *Deploy All*
- wyłączyć sterownik z zasilania
- z powrotem ustalić poprzedni adres na zworkach jeżeli był zmieniany
- włączyć sterownik

Wykonanie powyższych operacji przywraca hasło menu HMI do wartości domyślnej *1111* oraz przełącza wersję językową na pierwszą z listy wersji językowych menu HMI, a także przywraca nastawy adresu *IP* wbudowanej karty *ETH*.