



**AKADEMIA GÓRNICZO-  
HUTNICZA  
IM. STANISŁAWA  
STASZICA  
W KRAKOWIE**

---

## **LABORATORIUM PRZEMYSŁOWYCH SYSTEMÓW STEROWANIA**

---



**Wydział Inżynierii Mechanicznej i Robotyki**



**Katedra Automatykacji Procesów**

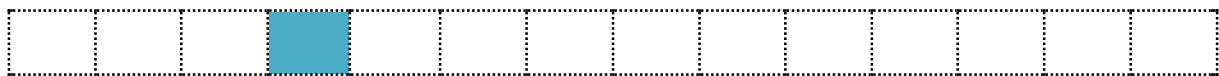
---

**Przedmiot:**

**Przemysłowe Systemy Sterowania (PSS)**

**Laboratorium 12:**

Programowanie paneli HMI firmy B&R w środowisku Automation Studio.



**Kraków**

## **Cel ćwiczeń laboratoryjnych**

1. Zapoznanie z budową i zasadą działania paneli HMI.
2. Poznanie metod programowania paneli HMI w środowisku Automation Studio.
3. Realizacja sterowania fizycznymi wejściami i wyjściami sterownika PLC z poziomu panelu HMI.

## **Po ukończeniu zajęć student powinien potrafić:**

1. Samodzielnie skonfigurować i skomunikować się z panelem HMI znajdującym się na stanowisku.
2. Potrafić obsłużyć i zaprogramować podstawowe elementy graficzne panelu HMI w środowisku Automation Studio.
3. Zrealizować interfejs dotykowy dla sterownika PLC do obsługi cyfrowych oraz analogowych wejść/wyjść.

## Wstęp

Panelem operatorskim nazywamy urządzenie współpracujące ze sterownikami przemysłowymi i umożliwiające komunikację człowiek-maszyna (HMI – *Human Machine Interface*). Obecnie panele znajdują zastosowanie nie tylko w dużych - bardzo rozbudowanych systemach automatyki, lecz coraz częściej używa się ich nawet w bardzo prostych aplikacjach.

Główne funkcje paneli operatorskich:

- wizualizacja procesu technologicznego (panele graficzne),
- wyświetlanie aktualnych wartości zmiennych procesowych,
- wyświetlanie trendów procesu,
- wyświetlanie pojawiających się alarmów,
- możliwość bezpośredniego wprowadzania nastaw i parametrów do sterownika.

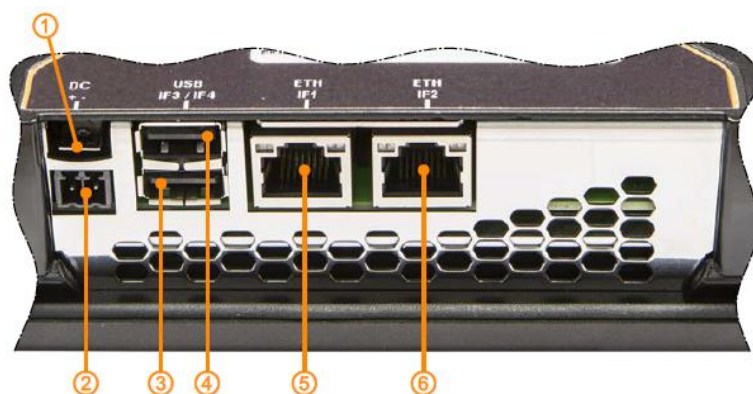
Obecny w Laboratorium Panel HMI to Power Panel T50. Panel ten ma pojemnościowy wyświetlacz o przekątnej 7 cali, rozdzielczości 800 na 480 pikseli z techniką *Multi-Touch*. W jego wnętrzu znajduje się procesor 800 MHz oraz 1GB pamięci RAM. Panel HMI posiada dwa tryby umożliwiające komunikację z nim:

- Tryb Klienta VNC (*Virtual Network Computing*) umożliwiający wyświetlanie wizualizacji stworzonej i zaprogramowanej w środowisku Automation Studio, wgranej do sterownika PLC.
- Wbudowaną przeglądarkę Web.



Rysunek 1. Widok trybu Runtime panelu PPT50.

## Konfiguracja Panelu HMI

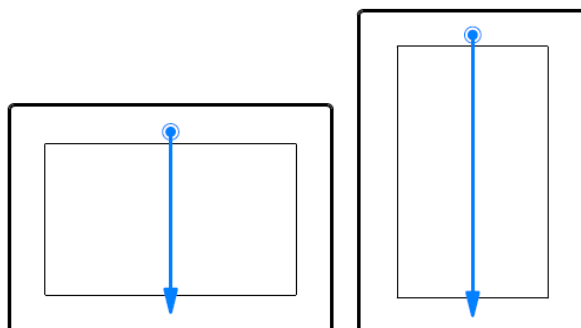


Rysunek 2. Przyłącza panelu HMI

- 1- Zacisk do podłączenia uziemienia
- 2- Zacisk do podłączenia zasilania panelu
- 3- Interfejs USB IF3
- 4- Interfejs USB IF4
- 5- Interfejs Ethernet IF1
- 6- Interfejs Ethernet IF2

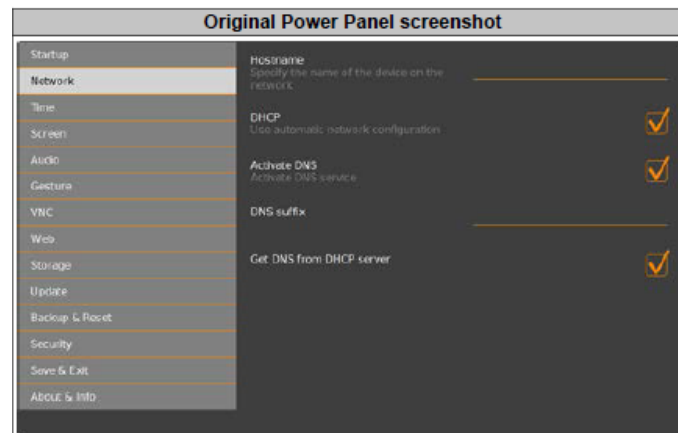
W celu prawidłowej pracy wyświetlacza konieczne jest podłączenie do niego zasilania oraz połączenie ze sterownikiem PLC. Na stanowiskach w laboratorium do jednego portu w panelu podłączony jest sterownik, drugi port połączony jest do sieci wewnątrz laboratorium. Przed rozpoczęciem pracy z panelem HMI należy poznać jego adres IP. Wymagane jest, aby panel HMI znajdował się w tej samej podsieci, co współpracujące z nim PLC oraz komputer inżynierski. Można ręcznie nadać adres IP zarówno dla panelu HMI, sterownika PLC jak i komputera PC, jednak w laboratorium te urządzenia łączą się z serwerem DHCP (*Dynamic Host Configuration Protocol*), który automatycznie przydziela odpowiednie IP podłączonym urządzeniom.

Aby sprawdzić adres IP panelu HMI należy wejść do trybu serwisowego przesuwając palec od górnej centralnej części panelu ku dołowi.



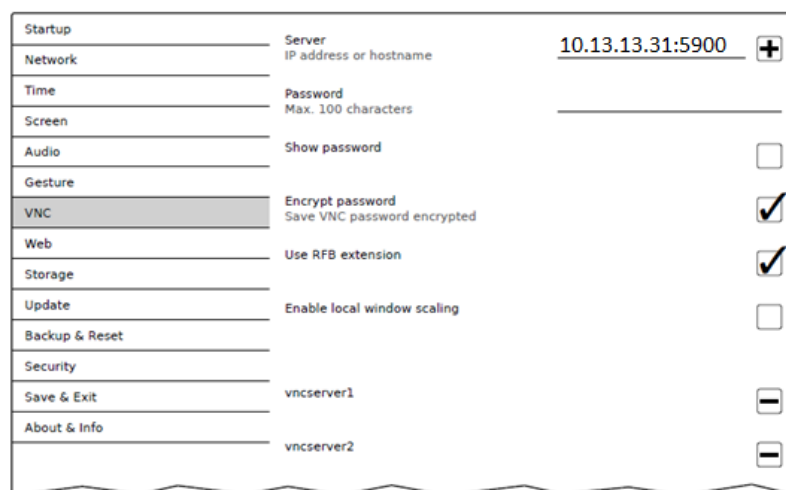
Rysunek 3. Przejście w tryb serwisowy

Ukazuje nam się strona serwisowa panelu. W menu po lewej stronie mamy dostęp do całej konfiguracji naszego HMI. W zakładce *Network* widzimy zaznaczoną opcję łączenia się panelu z serwerem DHCP. Oznaczając tę opcję możemy ręcznie wprowadzić parametry konfiguracji sieciowej panelu.



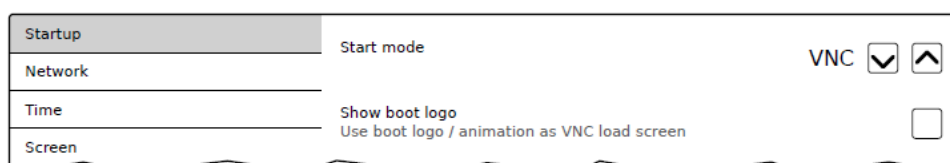
Rysunek 4. Zakładka Network

Jeżeli panel jest w trybie klienta VNC, konieczne jest w zakładce VNC dodanie adresu IP sterownika, z którym ten panel będzie się komunikował oraz numeru portu do komunikacji. Przykładowo dla stanowiska nr 3, zakładka VNC powinna być wypełniona w następujący sposób:



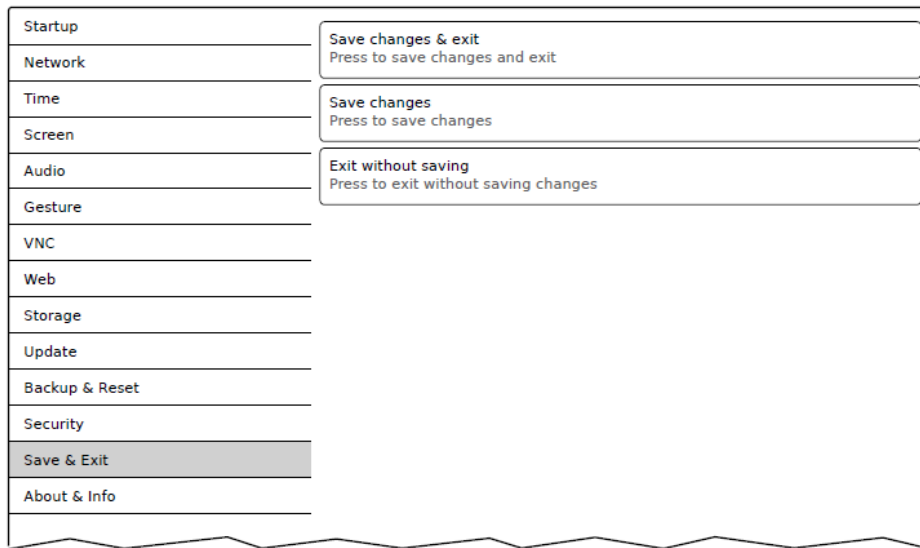
Rysunek 5. Zakładka VNC

Aby po wgraniu panel automatycznie wyświetlał wizualizację wgraną do sterownika PLC należy w zakładce *Startup* ustawić tryb startowy jako VNC.



Rysunek 6. Zakładka Startup

Po zakończeniu konfiguracji panelu HMI, należy zapisać wszystkie zmiany, w tym celu należy przejść do zakładki *Save&Exit* oraz zatwierdzić wprowadzone zmiany korzystając z 3 możliwych opcji.



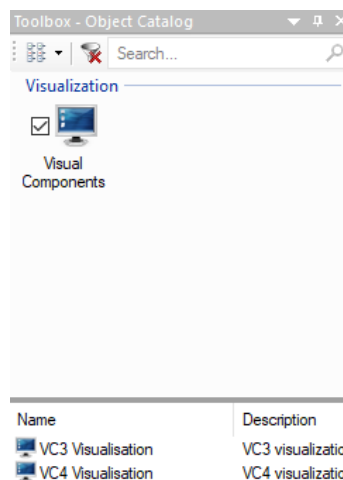
Rysunek 7. Zakładka Save&Exit

## Ćwiczenie 1

Zapoznaj się z trybem serwisowym panelu HMI dostępnym na stanowisku. Sprawdź, czy wszystkie ustawienia są zgodne z zaleceniami przedstawionymi powyżej, jest to niezbędne do jego prawidłowej współpracy ze sterownikiem PLC.

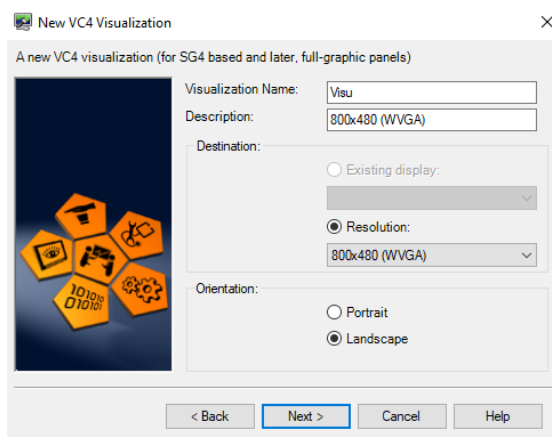
## Tworzenie wizualizacji w środowisku Automation Studio

W celu dodania wizualizacji do projektu, należy przeciągnąć ikonę wizualizacji z zakładki *Toolbox* do zakładki *Logical View*. Aby ikona wizualizacji była widoczna *toolboxie*, konieczne jest kliknięcie w pole *Logical View*.



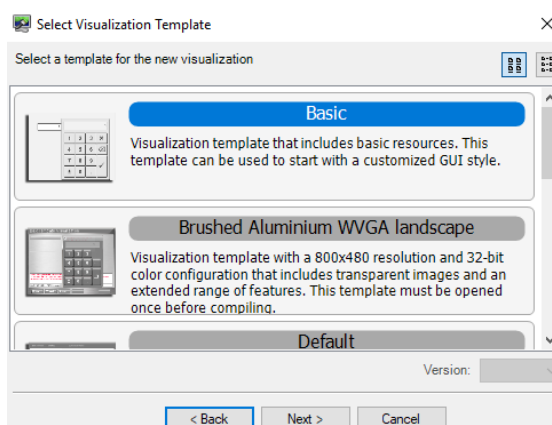
Rysunek 8. Dodanie wizualizacji do projektu

W naszym przypadku wybieramy *VC4 Visualisation*, po przeciągnięciu go do zakładki *Logical View* ukazuje się nam jego okno konfiguracji.



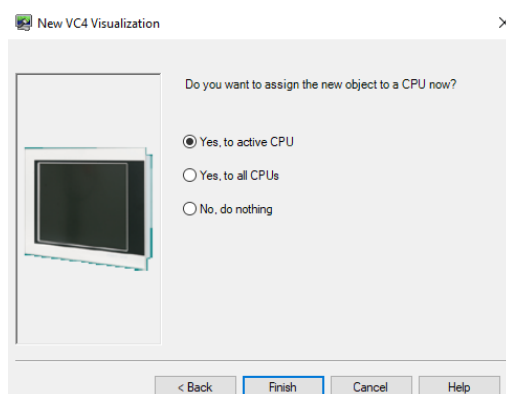
Rysunek 9. okno konfiguracyjne wizualizacji

Deklarujemy w nim nazwę naszej wizualizacji, opis oraz wybieramy odpowiednią rozdzielczość panelu HMI (800x480). Konieczne jest również wybór szablonu wizualizacji, wybierzemy opcję standardową Basic. Automation Studio daje również możliwość wyboru gotowych szablonów z częściowo dodanymi już elementami wizualizacji.



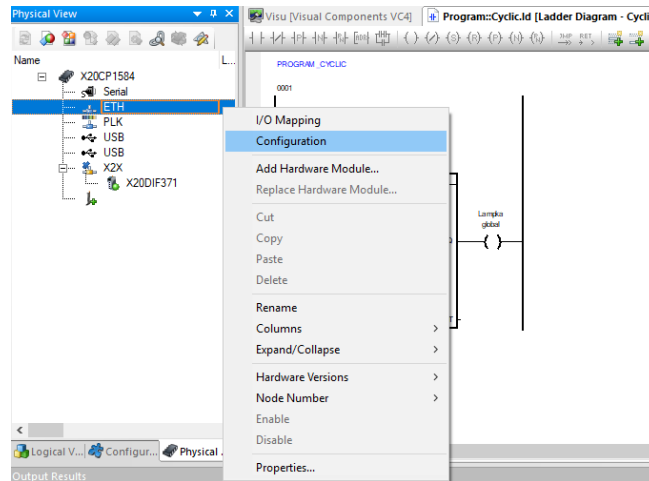
Rysunek 10. Wybór szablonu wizualizacji

Wgrywamy naszą wizualizację do aktywnej konfiguracji CPU sterownika PLC.



Rysunek 11. Dodanie wizualizacji do aktywnego CPU

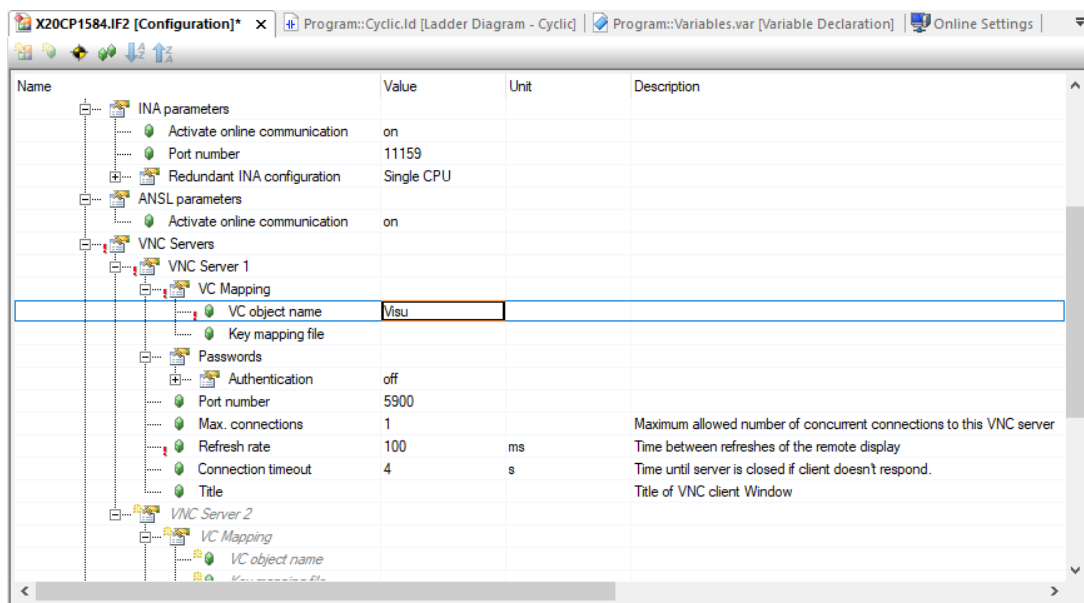
Po naciśnięciu przycisku *Finish* w zakładce *Logical View* ukaże się ikona wizualizacji. Przed przystąpieniem do pracy, należy ustawić połączenie wizualizacji z panelem. W tym celu w zakładce *Physical View* naciskamy prawym przyciskiem myszy na pole *ETH* (Ethernet) i wybieramy opcję *Configuration*.



Rysunek 12. Konfiguracja połączenia z panelem HMI

Po ukazaniu się okna konfiguracji, należy w kolumnie *Value* dla pola „*VC object name*” wybrać nazwę wizualizacji. Dodatkowo można również ustawić numer portu do komunikacji z panelem HMI oraz czas odświeżania programu przez panel HMI. Parametry konfiguracyjne wizualizacji powinny być takie jak na rysunku poniżej.

Domyślnie VNC korzysta z portów TCP 5900 – 5906, gdzie każdy z portów oznacza odrębną sesję (:0 do :6), lecz zarówno klient jak i serwer mogą zostać skonfigurowane do pracy na dowolnych innych portach.



Rysunek 13. Okno konfiguracji połączenia z panelem HMI

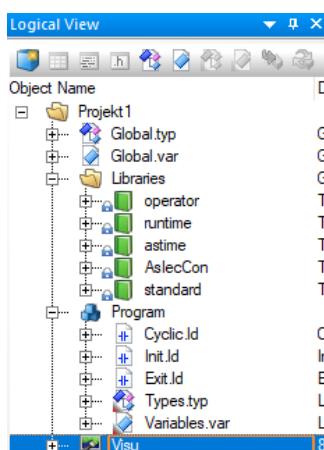


## Ćwiczenie 2

Dodaj do projektu komponent wizualizacji oraz skonfiguruj go odpowiednio do zaleceń podanych w instrukcji. Aby sprawdzić poprawność konfiguracji możesz wgrać pusty program do sterownika PLC, po zakończeniu procesu na panelu powinien wyświetlić się pusty ekran wizualizacji.

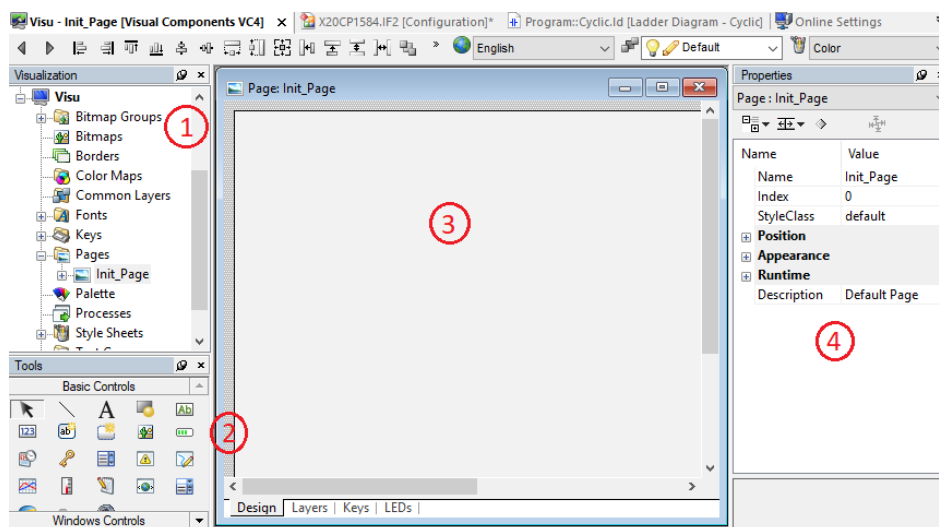
### Programowanie panelu HMI w Automation Studio

Po pomyślnym skonfigurowaniu panelu HMI i przetestowaniu poprawności jego komunikacji ze sterownikiem PLC możemy przejść do rozwinięcia jego wizualizacji. W zakładce *Logical View* należy dwukrotnie kliknąć na naszą ikonę wizualizacji.



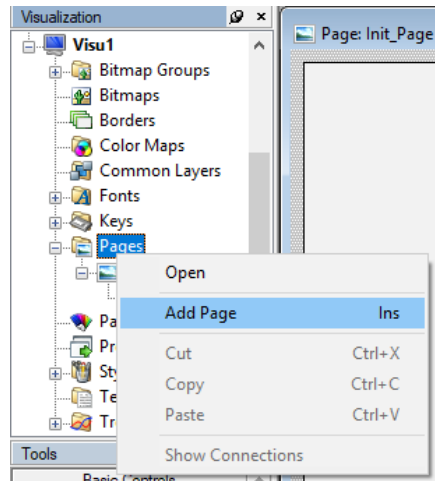
Rysunek 14. Widok logical View z komponentem wizualizacji

W drzewie po lewej stronie w folderze *Pages* znajdziemy pustą stronę domyślną naszej wizualizacji. Okno projektu wizualizacji składa się z 4 podstawowych elementów. Drzewa projektu z wszystkimi niezbędnymi komponentami dostępnymi dla użytkownika (1), podstawowych narzędzi do tworzenia wizualizacji (2), widoku poszczególnych ekranów panelu HMI (3) oraz ustawień aktualnie zaznaczonych elementów (4).



Rysunek 15. Okno edycji wizualizacji

Panele HMI programuje się za pomocą ekranów (ang. Screens). Domyślnie po utworzeniu wizualizacji utworzony zostaje ekran *Init\_Page*. Jest to ekran startowy, który wyświetla się jako pierwszy po przejściu panelu do trybu pracy *Runtime*. Można również dodać kolejne ekrany, klikając prawym przyciskiem na folder *Pages* i wybrać opcję *Add Page*.



Rysunek 16 Dodawanie kolejnego ekranu wizualizacji

W celu lepszego wytłumaczenia zasad programowania panelu HMI zaprezentowany zostanie przykład prostego programu.

## Przykład 1

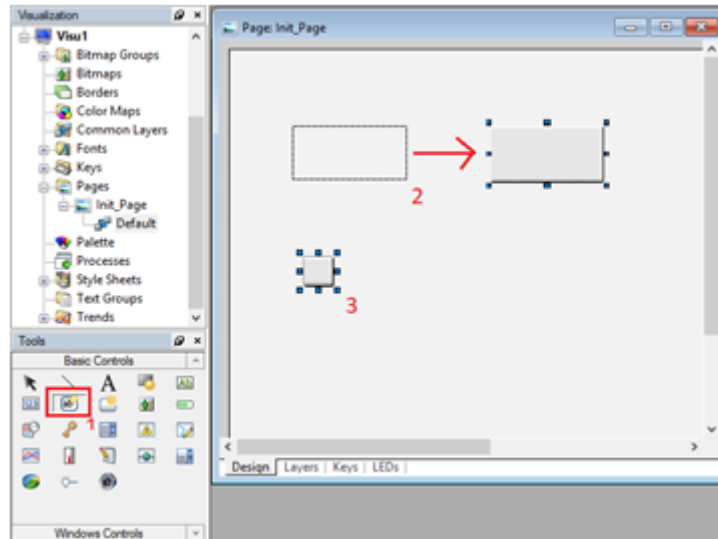
W przykładzie użyty zostanie wirtualny przycisk monostabilny i wirtualna dioda umieszczone na panelu HMI oraz fizyczny przycisk monostabilny z fizyczną diodą połączone elektrycznie do sterownika X20CP1584. Naciśnięcie wirtualnego przycisku, lub przycisku fizycznego będzie powodowało ustawienie stanu wysokiego dla wirtualnej diody i fizycznego wyjścia sterownika.

Należy zdefiniować trzy zmienne w sterowniku PLC:

Name	Type	Constant	Retain	Value	Description [1]
Przycisk	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
PrzyciskHMI	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Value_Lampka	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Value_LampkaHMI	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

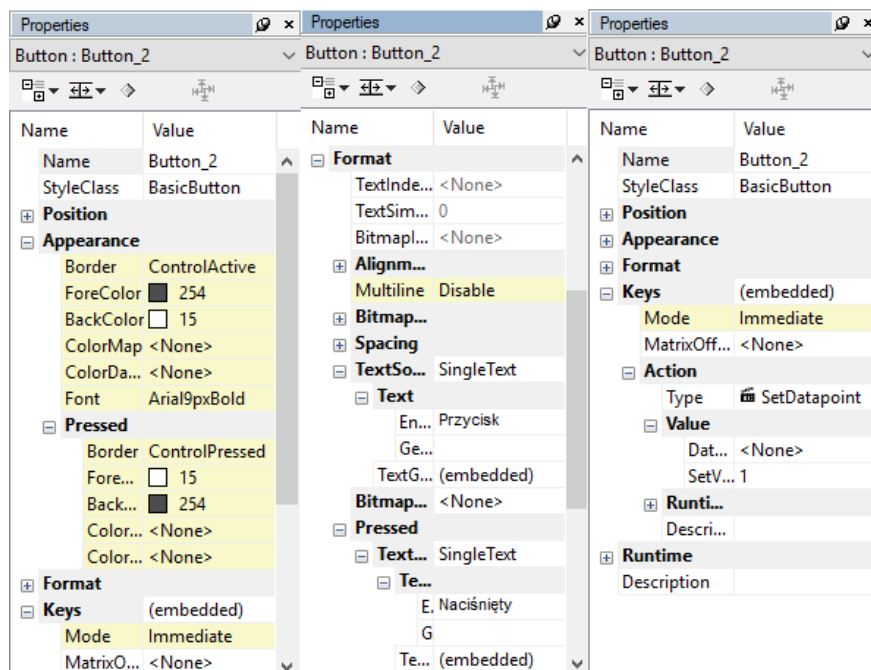
Rysunek 17 Deklaracja zmiennych.

Korzystając z okna narzędzi należy wybrać *Button* (1) i zaznaczyć miejsce na ekranie, gdzie ma zostać umieszczony przycisk, wyznaczając obszar jaki ma zajmować (2). Można również kliknąć na ekranie miejsce, gdzie chcemy dać przycisk, wtedy w tym miejscu pojawi się przycisk o domyślnych wymiarach w postaci kwadratu (3).



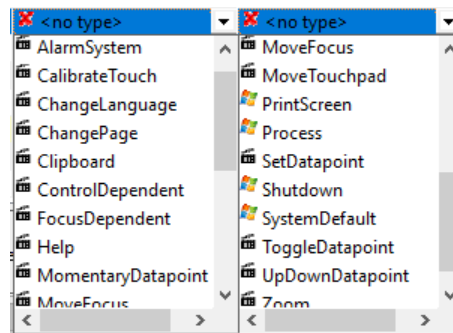
Rysunek 18 Dodawanie przycisku do ekranu wizualizacji

Naciskając na utworzony przycisk, w oknie po prawej stronie ukazują się jego ustawienia. W zakładce *Appearance* definiujemy wygląd przycisku. Zakładka *Format* w polach *Text Source* oraz *Pressed* pozwala m.in. wprowadzić tekst wyświetlany, gdy przycisk jest naciśnięty, lub nie. Zakładka *Keys* zawiera opcje *Action* oraz *Value*, które odpowiednio przypisują przyciskowi akcję po jego naciśnięciu oraz pozwalają na powiązanie przycisku ze zmienną, której dana akcja będzie dotyczyć.



Rysunek 19 Ustawienia przycisku

W środowisku Automation Studio mamy do wyboru wiele akcji dla elementu *button*. Lista z dostępnymi akcjami ukazuje się po rozwinięciu pola *Type* w zakładce *Keys->Action*.



Rysunek 20 Wybór akcji dla przycisku

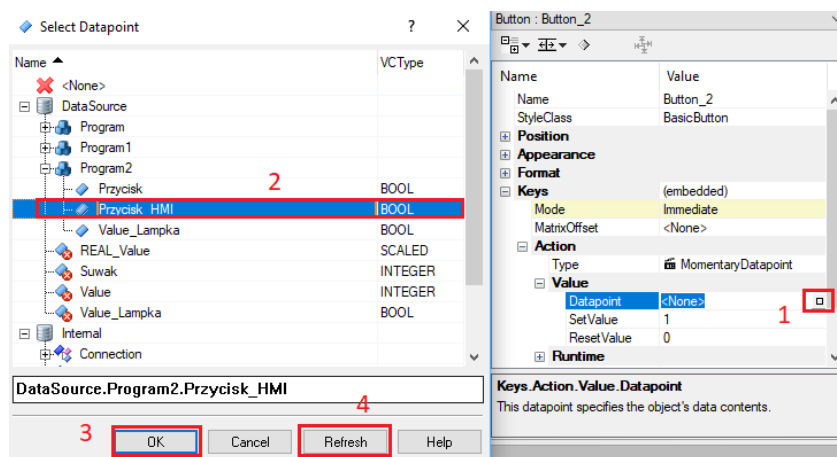
Najczęściej używane akcje:

- AlarmSystem- określa, czy wybrany alarm ma zostać potwierdzony, czy też ma zostać zablokowany.
- CalibrateTouch -Po naciśnięciu przycisku ekran dotykowy przechodzi do stanu kalibracji. Wyświetlone zostaną punkty kalibracji, po zakończeniu kalibracji ostatni wyświetlony obraz procesu zostanie ponownie wyświetlony.
- ChangeLanguage -Po naciśnięciu przycisku język w systemie docelowym jest przełączany między dostępnymi językami.
- ChangePage - Naciśnięcie przycisku powoduje zmianę strony.
- Clipboard ( schowek) - Operacja może być użyta do zaznaczania, kopiowania i wstawiania tekstu w kontrolce Edycja.
- Help - Służy do przełączenia wizualizacji w tryb Pomocy, wszystkie wprowadzane dane za pośrednictwem ekranu dotykowego zostaną zignorowane.
- MomentaryDatapoint- Zapisuje określoną wartość do zmiennej tak długo, jak długo naciśnięty jest przycisk. Po zwolnieniu przycisku do tego punktu danych zapisywana jest wartość resetu.
- PrintScreen- Za jego pomocą można wydrukować bieżącą zawartość ekranu na domyślnej drukarce skonfigurowanej dla terminala Windows. Ta czynność jest wykonywana tylko wtedy, gdy aplikacja wizualizacyjna została mapowana do terminala Windows. W systemie docelowym z systemem operacyjnym Automation Runtime ta czynność nie przynosi żadnego efektu.
- Process - Może być użyty do uruchomienia lub zatrzymania procesu (aplikacji Windows) w czasie działania. Ta czynność jest wykonywana tylko wtedy, gdy aplikacja wizualizacyjna została mapowana do terminala Windows. W systemie docelowym z systemem operacyjnym Automation Runtime ta czynność nie przynosi żadnego efektu.
- SetDatapoint - Po naciśnięciu przycisku SetDatapoint do danej zmiennej zapisywana jest zdefiniowana we właściwościach wartość.
- Shutdown - Czynność może być użyta do wyłączenia lub ponownego uruchomienia terminala Windows. Ta czynność jest wykonywana tylko wtedy, gdy aplikacja

wizualizacyjna została mapowana do terminala Windows. W systemie docelowym z systemem operacyjnym Automation Runtime ta czynność nie przynosi żadnego efektu.

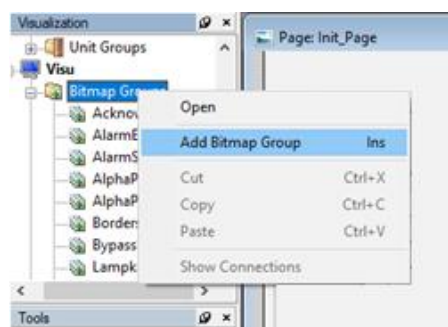
- ToggleDatapoint - Przełącza wartość zmiennej w przód i w tył pomiędzy dwoma wartościami za każdym naciśnięciem przycisku.
- UpDownDatapoint- Zmienia wartość zmiennej w górę lub w dół za każdym naciśnięciem przycisku. Zakres wartości może mieć górną i dolną granicę.
- Zoom- Za pomocą tego przycisku można zwiększać lub zmniejszać wielkość obszaru wyświetlanego w kontrolce Trend.

W naszym przypadku wykorzystamy akcje *MomentaryDatapoint*, tworząc monostabilny przycisk. Rozwijając opcję *Value* oraz naciskając w pole obok *Datapoint*, pojawia się ikona małego kwadratu(1), po naciśnięciu której ukazuje się okno w którym pojawiają się zmienne dostępne w programie . W naszym przypadku wybieramy zmienną *Przycisk\_HMI*(2), która będzie używana w programie sterownika. Operację kończymy przyciskiem OK(3), jednak istnieje możliwość, że pomimo zapisania zmiennych w projekcie, nie ukążą się one na liście, należy wtedy nacisnąć przycisk *Refresh*(4) aby zmienne te ukazały się na liście.



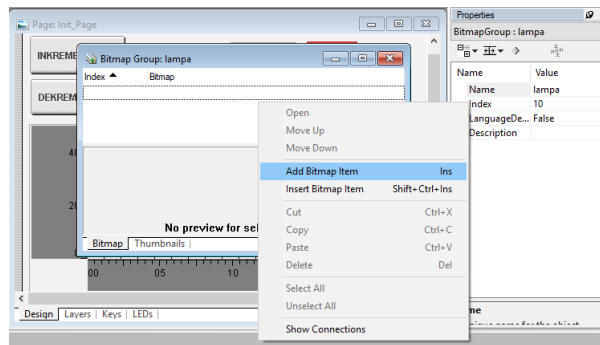
Rysunek 21 Powiązanie przycisku ze zmienną w projekcie

W Automation Studio w narzędziach nie ma elementów sygnalizacyjnych, lecz jest możliwość samodzielnego utworzenia animacji za pomocą grupy bitmap, co jest często wykorzystywane do tworzenia animacji działającego procesu. W tym celu do projektu w drzewie wizualizacji wybieramy folder bitmap group i dodajemy nową grupę.



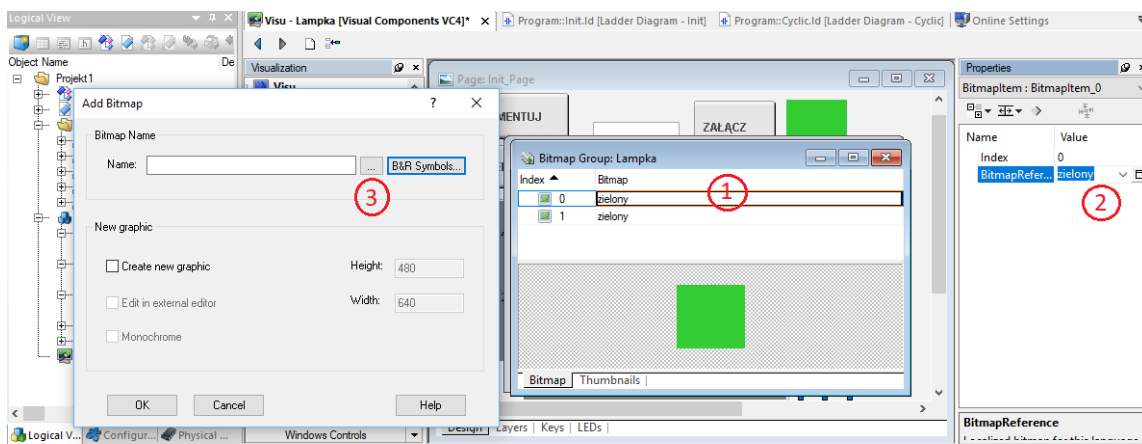
Rysunek 22 Dodawanie grupy bitmap

Po utworzeniu okna bitmapy, można w ustawieniach po prawej stronie zmienić jej nazwę, a naciskając prawym przyciskiem myszy w oknie dodać element bitmapy.



Rysunek 23 Dodawanie elementu bitmapy

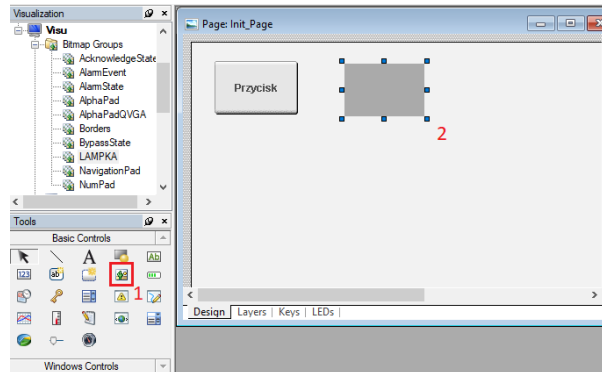
Po dodaniu dwóch elementów bitmapy, zaznaczając jeden z nich(1) i wybierając w ustawieniach opcję *Bitmap Reference*(2) otwiera nam się okno wyboru grafiki(3). Możemy skorzystać z gotowych symboli oferowanych przez B&R lub wykorzystać własne. Symbole BR znajdują się na dysku w katalogu z zainstalowanym *Automation Studio* w folderze *BR\_Symbols*. Do przedstawienia funkcjonalności lampki wystarczy własnoręcznie zrobiona grafika np. 50x50 pikseli w dowolnym programie graficznym (najlepiej jako mapa bitowa) w kolorze np. zielonym i czerwonym. Kolor zielony symbolizować może załączenie lampki, kolor czerwony natomiast jej wyłączenie.



Rysunek 24 Dodawania grafiki bitmapy

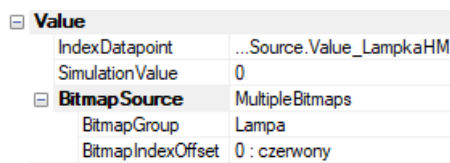
Index przy danej bitmapie informuje, przy jakim stanie zmiennej związanej z daną grupą bitmapy, zostanie wyświetlona dana grafika. W naszym przypadku konieczne jest zadeklarowanie zmiennej przechowującej tę wartość, np. *Value\_Lampka\_HMI* jest zmienną typu *BOOL*, której wartość decyduje o wyświetlanej grafice. Wartość ta będzie zmieniana w programie sterownika, po naciśnięciu fizycznego przycisku lub tego na panelu HMI.

Po utworzeniu grupy bitmap, konieczne jest jej dodanie do ekranu wizualizacji, w tym celu z zakładki narzędzi wybieramy element *bitmap(1)* i na ekranie myszką określamy kształt bitmapy(2), jest on obojętny, gdyż po wgraniu grafiki, automatycznie przyjmie ona wymiary zadeklarowane w grafikach danej bitmapy.



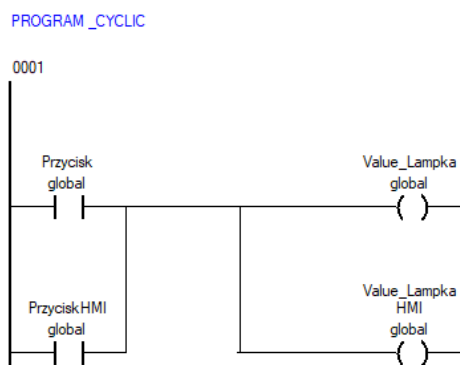
Rysunek 25 Dodanie bitmapy do ekranu wizualizacji

W ustawieniach bitmapy konieczne jest dodatkowo zadeklarowanie *BitmapSource* jako *MultipleBitmaps*. W polu *BitmapGroup* wybieramy utworzoną przez nas bitmapę Lampa, natomiast jako *IndexDatapoint* ustawiamy zmienną w programie powiązaną z naszą lampką.



Rysunek 26 Ustawienia bitmapy Lampa

Po dodaniu i skonfigurowaniu przycisku i lampki, konieczne jest napisanie kodu programu, który zrealizuje treść zadania w tym przykładzie. Przykładowy kod napisany w języku drabinkowym może wyglądać następująco:



Rysunek 27 Kod do przykładu w języku drabinkowym

### Ćwiczenie 3

Wykonaj ćwiczenie z zaproponowanego przykładu.

## Przykład 2

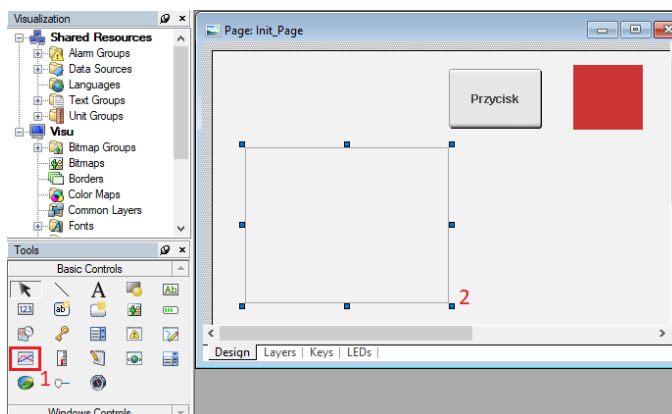
Aby lepiej zobrazować podstawowe funkcje dostępne w wizualizacji, do istniejącej wizualizacji dodamy kilka komponentów. Dodamy dwa przyciski, *bargraph*, *trend*, *numeric*, rozmieścimy na ekranie według uznania. Wykorzystując 2 przyciski będziemy inkrementowali i dekrementowali zmienną, której wartość będzie przedstawiona na wykresie *trendu*, *bargraphie* oraz w polu *numeric*.

Aby prezentować dane na wykresach dodamy zmienną typu INT, której wartość będzie na nich wyświetlana.

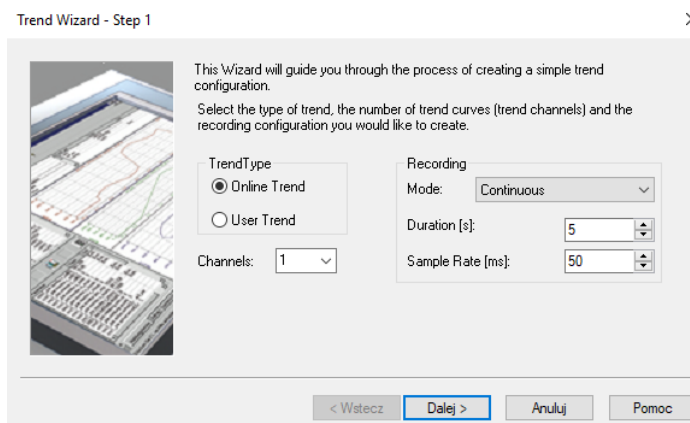
Name	Type	Constant	Retain	Value	Description [1]
Przycisk	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
PrzyciskHMI	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Value_Lampka	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Value_LampkaHMI	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Value	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Rysunek 28 Dodanie zmiennej typu INT

Przystępując do realizacji przykładu, w palety narzędzi wybieramy element *Trend*, na ekranie wizualizacji zaznaczamy obszar, jaki ma zajmować.



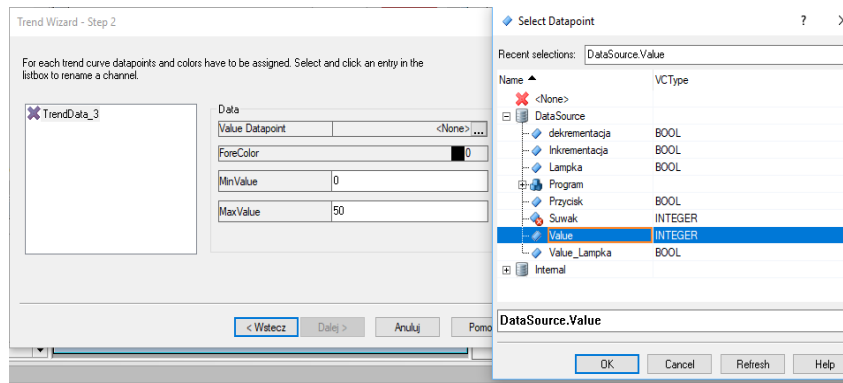
Dodając trend do ekranu, pojawia się okno konfiguracji trendu. Możemy ustawić odcinek czasu wyświetlany na trendzie, co jaki czas pobierane są próbki oraz typ wykresu i liczbę kanałów.



Rysunek 29. Okno konfiguracji trendu

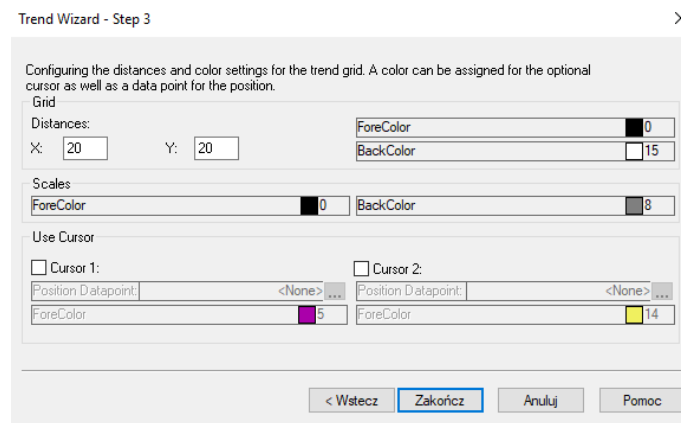


W kolejnym oknie wybieramy zmienną w polu „*Value Datapoint*”, która przedstawiana jest na wykresie *trendu* oraz zakres, jaki będzie wyświetlany.



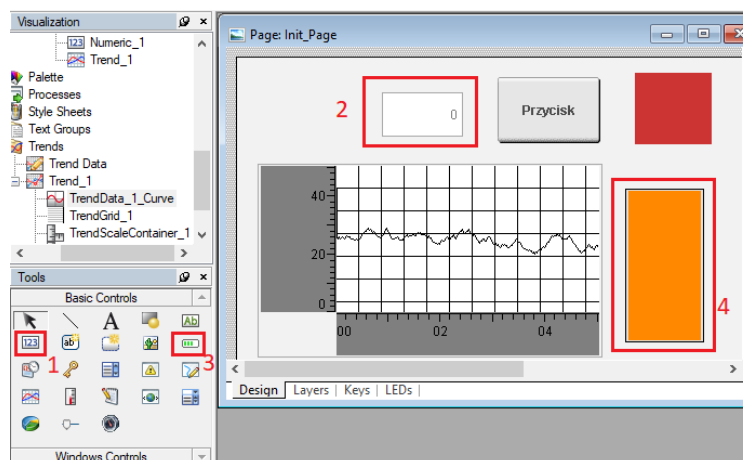
Rysunek 30. Konfiguracja trendu

Ostatnie okno umożliwia ustawienie dodatkowych opcji *trendu*:



Rysunek 31. Dodatkowe opcje trendu

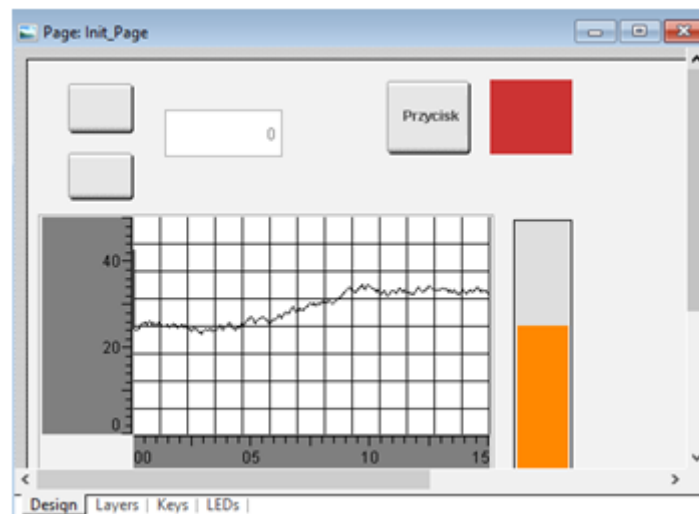
Identycznie postępujemy z elementem *bargraph(3)* oraz *numeric(1)*, wystarczy wybrać elementy z zakładki narzędzi i zaznaczyć na ekranie wizualizacji obszar, jaki mają zajmować(2,4).



Rysunek 32 Dodanie elementu bargraph oraz numeric

Ostatnim etapem jest dodanie dwóch przycisków do inkrementacji i dekrementacji zmiennej *Value*, instrukcja dodania elementów tego typu została zawarta w *Przykładzie 1*.

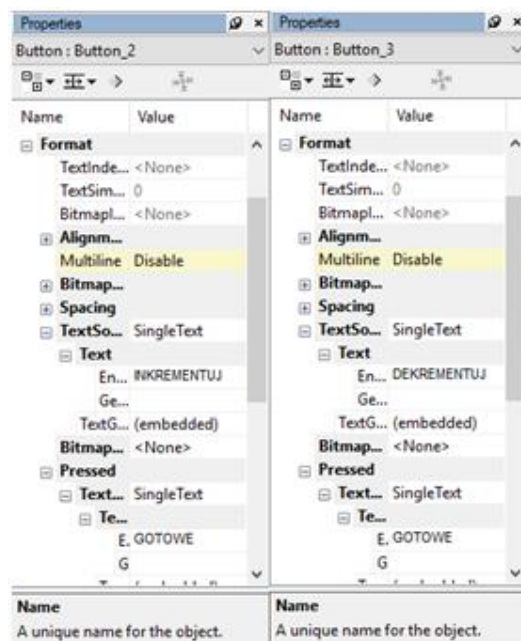
Przykładowe rozmieszczenie elementów na ekranie panelu HMI.



Rysunek 33. Rozmieszczenie elementów na ekranie wizualizacji

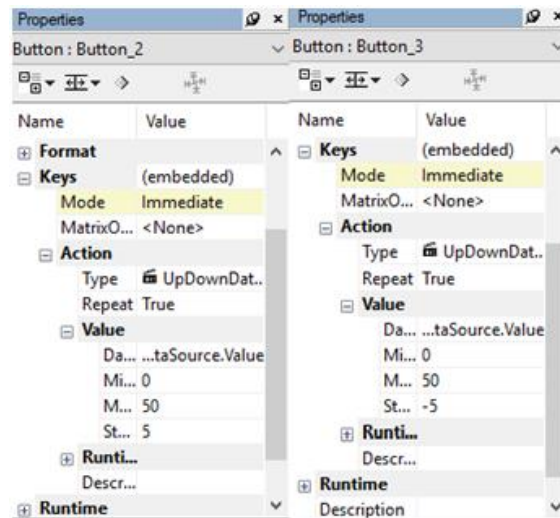
Następnym krokiem podczas tworzenia wizualizacji jest powiązanie jej komponentów z odpowiednimi zmiennymi oraz ich konfiguracja. Aby tego dokonać, zaznaczając odpowiedni element na ekranie ustawiamy opcje w oknie po prawej stronie

Rys. 34 przedstawia ustawienie napisów wyświetlanych na odpowiednich przyciskach. Aby je wyświetlać, trzeba ustawić opcję *Single Text* lub *Multiple Text*. Zakładka *Text Source* obrazuje napis na przycisku, gdy nie jest on naciśnięty, natomiast w gałęzi *Pressed* ustawiamy napis jaki pojawi się na przycisku po jego naciśnięciu.



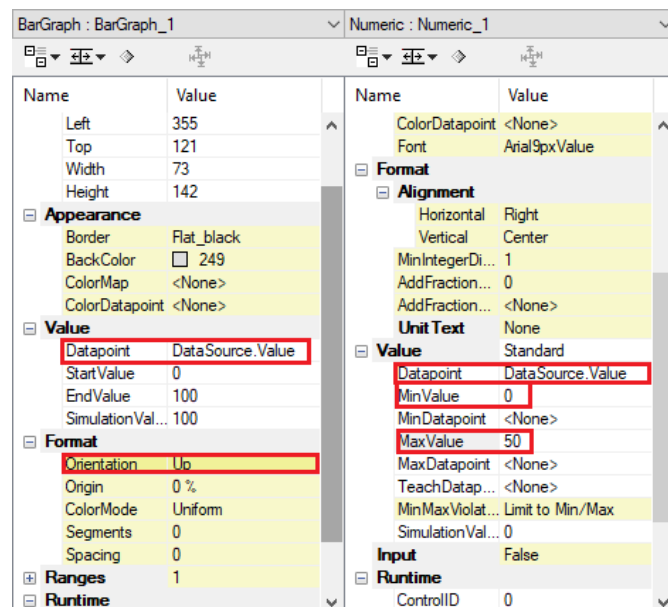
Rysunek 34. Ustawienia tekstu na przyciskach

Następnie ustawiamy funkcje realizowane przez przyciski. Przyciski inkrementacji i dekrementacji mają za zadanie odpowiednio modyfikować zmienną *Value*. Ustawiamy zakres zmiennej oraz w polu *Step Value* wpisujemy wartość, która będzie dodawana do zmiennej po każdorazowym naciśnięciu.



Rysunek 35. Ustawienia funkcji przycisków

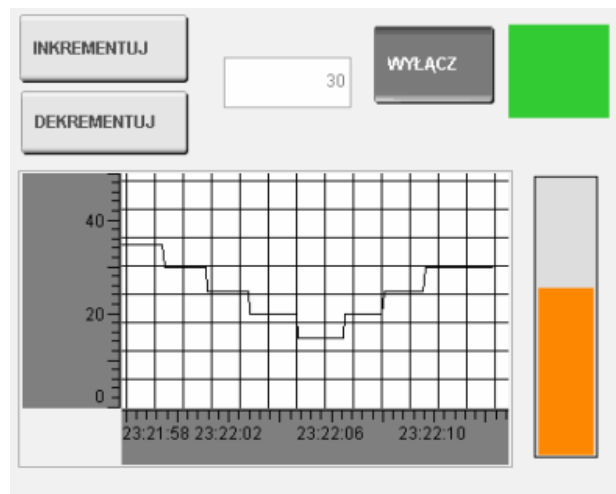
Ustawienia komponentów *bargraph* oraz *numeric* wymagają tylko przypisania zmiennej *Datapoint* oraz opcjonalnie podanie zakresów wyświetlania zmiennej przypisanej do danego elementu. Dodatkowo w elemencie *bargraph* możliwe jest określenie jego orientacji, w naszym przypadku wykres będzie rósł ku górze, zatem wybieramy *Orientation: Up*.



Rysunek 36. Ustawienia bargraph oraz numeric

Po ustawieniu wszystkich komponentów wizualizacji, jest ona gotowa do wgrania do sterownika. Wizualizacja jest przesyłana razem z programem podczas wgrzywania projektu, zatem nie są konieczne żadne dodatkowe kroki.

Po wgraniu projektu, na ekranie panelu HMI powinien pojawić się nasz ekran z wizualizacją. Naciskając przyciski inkrementacji lub dekrementacji, modyfikacje zmiennej *Value* powinny być widoczne zarówno na wykresie *trendu*, *bargraphie* jak i elemencie *numeric*.



Rysunek 37. Gotowy ekran wizualizacji na panelu HMI

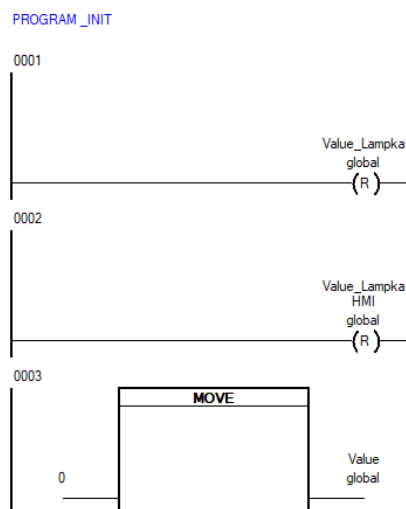
## Ćwiczenie 4

Wykonaj ćwiczenie z zaproponowanego przykładu oraz przetestuj jego działanie.

### Uwaga

Zmienne używane w wizualizacji muszą pojawić się w programie na sterowniku PLC, aby były prawidłowo przetwarzane. Jeżeli nie występuje potrzeba ich bezpośredniego użycia w programie cyklicznym, wystarczy wykorzystać program inicjalizujący, który po włączeniu sterownika ustawi odpowiednie wartości zmiennych.

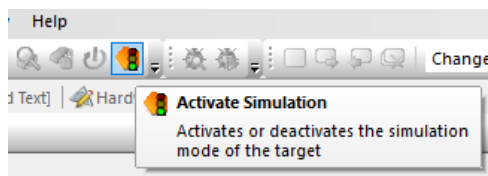
Poniżej pokazany jest przykładowy program inicjalizacji zmiennych używanych w wizualizacji. W tym przypadku jedynie zmienna *Value* nie występuje w programie cyklicznym, użycie pozostałych jest opcjonalne.



Rysunek 38. Program inicjalizacji zmiennych

## Symulacja sterownika w Automation studio

Środowisko Automation Studio posiada wbudowany symulator (AR000), który umożliwia przetestowanie programu bez konieczności posiadania fizycznego sprzętu. Aby przejść do symulacji na pasku narzędziowym odszukaj opcję *Activate Simulation* i kliknij na nią. Po chwili symulacja zostanie uruchomiona.



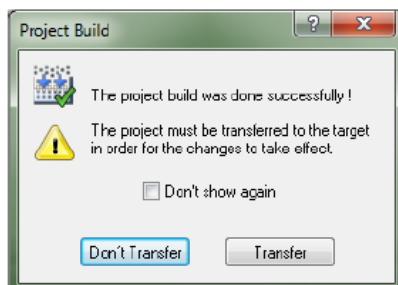
Rysunek 39 Aktywowanie symulacji

Przed rozpoczęciem wgrywania programu upewnij się czy ustanowione jest połączenie online z symulatorem/sterownikiem (rozpoznasz to po aktywowaniu się przycisków do obsługiwanego połączenia i informacji na pasku u dołu ekranu).

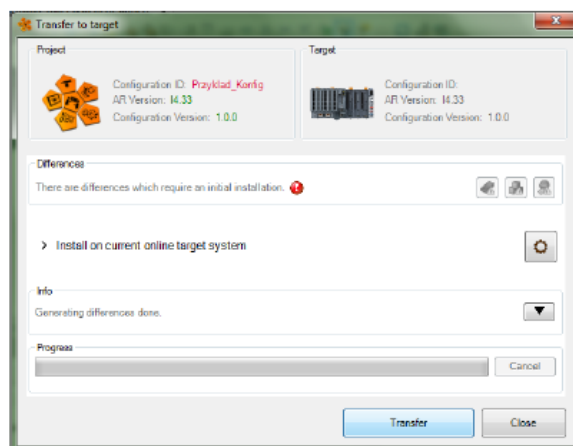


Rysunek 40 Informacje o stanie połączenia ze sterownikiem/symulatorem

Skompiluj program wybierając z menu *Project>Build Configuration*, wciskając klawisz *F7* lub klikając przycisk na pasku narzędziowym

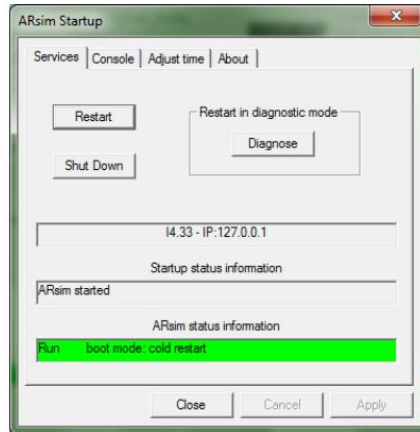


Rysunek 41 Przejście do transferu projektu po jego kompilacji



Rysunek 42 Okno Transferu projektu

Poprawne zakończenie procesu kompilacji zostanie zasygnalizowane pojawieniem się odpowiedniej informacji. W oknie, które się ukaże wybierz *Transfer*, by wgrać projekt do symulowanego sterownika. W kolejnym oknie również kliknij przycisk *Transfer*, a następnie w ostrzeżeniu które się pojawi wyraż zgodę na wgranie nowego programu. Jeżeli proces ten się powiedzie pojawi się okienko z parametrami symulacji ukazane poniżej.



Rysunek 43 Okno symulacji

Jeżeli dane okno nie pojawi się automatycznie na ekranie, można go odnaleźć w pasku powiadomień systemu Windows, który zawiera zasobnik systemowy (system tray). Odnajdziemy tam ikonę symulatora, po dwukrotnym kliknięciu ukaże się nam powyższe okno.



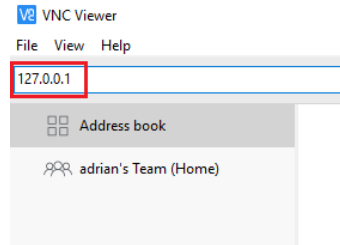
Rysunek 44 Ikona symulacji w pasku powiadomień

## Symulacja panelu HMI w VNC Viewer

Istnieje również możliwość symulacji panelu HMI na komputerze, bez konieczności posiadania fizycznego sprzętu. W tym celu po utworzeniu i skonfigurowaniu wizualizacji w środowisku Automation Studio należy uruchomić symulację sterownika według zaleceń przedstawionych powyżej.

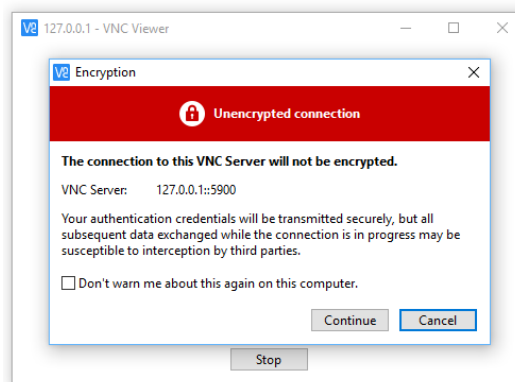
Ponieważ panel HMI korzysta z systemu VNC do komunikacji ze sterownikiem, konieczne jest zainstalowanie programu klienta np. *VNC Viewer* na komputerze, na którym chcemy skorzystać z symulacji naszej wizualizacji. Program ten jest zainstalowany na stanowisku laboratoryjnym.

Aby zasymulować panel, po wgraniu projektu do symulatora lub sterownika, wystarczy otworzyć program *VNC Viewer* i w pasku wpisać adres IP sterownika (obecnego na stanowisku, jeżeli pracujemy na fizycznym sprzęcie, lub adres z symulatora, widoczny na dolnym pasku informacyjnym w *Automaton Studio*).



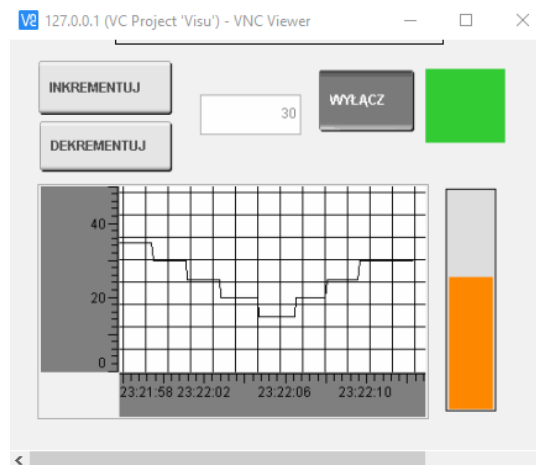
Rysunek 45 Wprowadzenie adresu IP w programie VNC Viewer

Po wprowadzeniu adresu IP pojawi się okno z ostrzeżeniem, że nasze połączenie nie jest szyfrowane, dodatkowo widzimy, że program odczytał automatycznie port używany do komunikacji, zadeklarowany w projekcie w środowisku *Automation Studio*. Należy zaakceptować komunikat naciskając przycisk *Continue*.



Rysunek 46 Informacja o nieszyfrowanym połączeniu VNC Viewer

W przypadku braku jakichkolwiek problemów po zatwierdzeniu komunikatu powinno ukazać się okno z widokiem wizualizacji naszego panelu HMI utworzonej wcześniej.

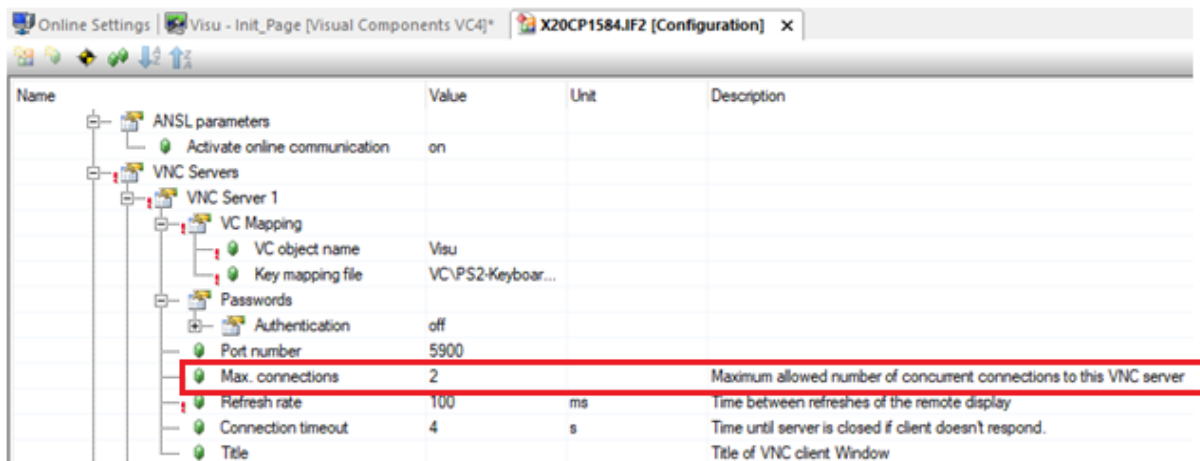


Rysunek 47 Ekran panelu HMI w programie VNC Viewer

## Praca równoległa fizycznego panelu HMI oraz wizualizacji w VNC Viewer.

Istnieje możliwość komunikacji ze sterownikiem z wieloma urządzeniami podłączonymi do tej samej sieci. Serwer VNC obecny w sterowniku PLC pozwala na synchroniczne przekazywanie obrazu do wszystkich urządzeń połączonych z nim, pełniących rolę klientów. Mogą one oddziaływać na serwer i wpływać na jego działanie, dzięki temu istnieje możliwość sterowania programem sterownika równoległe z poziomu panelu HMI, smartfona, czy komputera osobistego.

Aby umożliwić połączenie się większej liczby urządzeń, konieczne jest przejście do konfiguracji wizualizacji. Należy przejść do zakładki *Physical View*, nacisnąć prawym przyciskiem myszy na ikonę *ETH* oraz wybrać opcję *Configuration*, przejście do konfiguracji zostało pokazane na Rys.12. W zakładce *VNC Servers*, należy odnaleźć połączenie z naszą wizualizacją i w polu *Max. Connections* wprowadzić maksymalną liczbę urządzeń mogących połączyć się z serwerem (domyślnie ustawione jest połączenie z jednym urządzeniem).



Rysunek 48 Ustawienie maksymalnej liczby klientów VNC

### Ćwiczenie 5

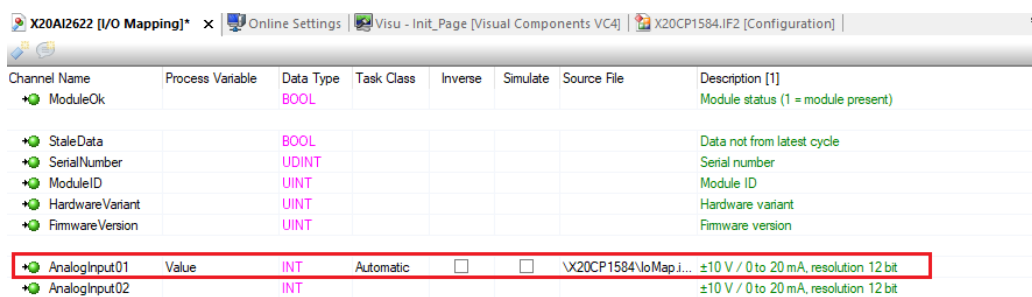
Zmień ustawienia konfiguracji serwera VNC zgodnie z powyższymi zaleceniami, połącz się z fizycznym sterownikiem znajdującym się na stanowisku laboratoryjnym. Wypróbuj działanie równoległe fizycznego panelu HMI i programu *VNC Viewer* na komputerze PC.



## Korzystanie z analogowych wejść/wyjść w sterownikach B&R

Większość układów sterowania oprócz standardowych wejść i wyjść cyfrowych wykorzystuje także wejścia i wyjścia analogowe m.in. do pomiarów temperatury, ciśnień, odległości, ale także np. do sterowania prędkościami silników z użyciem falowników.

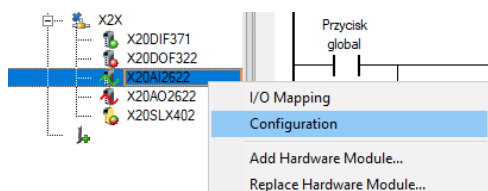
Na stanowisku laboratoryjnym do sterownika PLC dołączony jest moduł wejść analogowych X20AI2622 oraz moduł wyjść analogowych X20AO2622. W środowisku Automation Studio, aby wykorzystać dane wejścia lub wyjścia analogowe, wystarczy w konfiguracji sprzętowej danego modułu wybrać opcję mapowania wejść/wyjść oraz przypisać zmienną wykorzystaną w programie do danego wejścia/wyjścia. Istotny jest typ zmiennej przypisywanej, musi być ona typu INT.



Channel Name	Process Variable	Data Type	Task Class	Inverse	Simulate	Source File	Description [1]
ModuleOk		BOOL					Module status (1 = module present)
StaleData		BOOL					Data not from latest cycle
SerialNumber		UDINT					Serial number
ModuleID		UINT					Module ID
HardwareVariant		UINT					Hardware variant
FirmwareVersion		UINT					Firmware version
AnalogInput01	Value	INT	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\\X20CP1584\IoMap.i...	±10 V / 0 to 20 mA, resolution 12 bit
AnalogInput02		INT					±10 V / 0 to 20 mA, resolution 12 bit

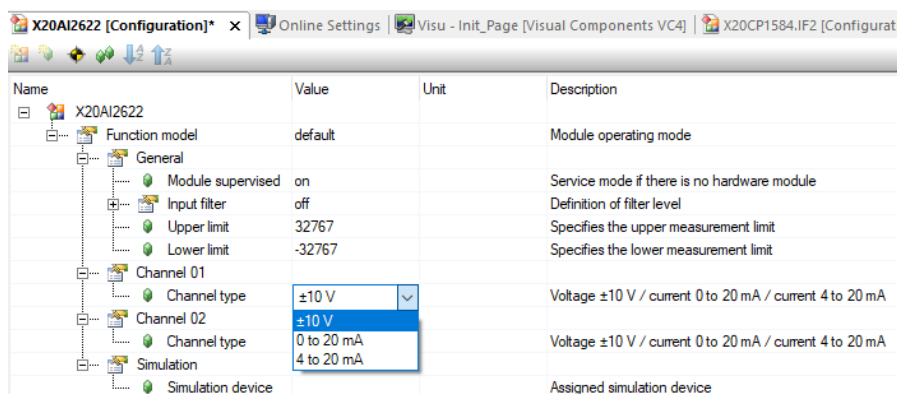
Rysunek 49 Przypisanie zmiennej do wejścia analogowego AnalogInput01

Naciskając prawym przyciskiem na moduł wejść/wyjść w zakładce *Physical View* możemy przejść do jego konfiguracji.



Rysunek 50 Przejście do konfiguracji wejść/wyjść analogowych

W konfiguracji można ustawić typ danego wejścia analogowego (napięciowy symetryczny lub dwa rodzaje wejść prądowych) oraz filtr wejściowy, niepozwalający na przekroczenie danej wartości zmiennej skojarzonej z danym wejściem analogowym..



Name	Value	Unit	Description
X20AI2622			
Function model	default		Module operating mode
General			
Module supervised	on		Service mode if there is no hardware module
Input filter	off		Definition of filter level
Upper limit	32767		Specifies the upper measurement limit
Lower limit	-32767		Specifies the lower measurement limit
Channel 01			
Channel type	±10 V		Voltage ±10 V / current 0 to 20 mA / current 4 to 20 mA
Channel 02			
Channel type	0 to 20 mA		Voltage ±10 V / current 0 to 20 mA / current 4 to 20 mA
Simulation			
Simulation device			Assigned simulation device

Rysunek 51 Konfiguracja wejścia analogowego

## Ćwiczenie 6

Zamiast przycisków inkrementacji oraz dekrementacji, wykorzystaj potencjometr znajdujący się na stanowisku laboratoryjnym i wyświetl wartość zmiennej powiązanej z wejściem analogowym na wykresie *trendu*, *bargraphie* oraz polu *numeric*.

### Skalowanie i normalizacja

Do przetwarzania zakresów zmiennych używane są przeważnie dwie operacje: Normalizacja oraz skalowanie, w praktyce występują one zawsze razem. Pierwsza z nich normalizuje parametr VALUE wewnątrz zakresu wartości wyspecyfikowanych przez parametry MIN i MAX. Druga operacja skaluje znormalizowany parametr VALUE ( $0.0 \leq \text{VALUE} \leq 1.0$ ) do typu danej i zakresu wartości wyspecyfikowanych przez parametry MIN i MAX.

Blok funkcyjny wykonujący operację skalowania i normalizacji można wykonać samemu w kilku liniach kodu. Należy utworzyć nowy blok funkcyjny i dodać go do programu. Potrzebne będą zmienne wejściowe zawierające wartości minimalne i maksymalne wartości wejściowej jak i wielkości skalowanej oraz sama zmienna skalowana. Ponieważ zmienna przechowująca wartość z wejścia analogowego jest typu „INT”, należy wejście zadeklarować w tym typie. Wyjściem bloku funkcyjnego będzie zmienna skalowana, która zależnie od potrzeb może być różnego typu. Można również dodać zmienną lokalną w programie przechowującą wartość wielkości znormalizowanej, zmienna ta może być np. typu „REAL”.

Przykładowa deklaracja zmiennych bloku funkcyjnego do skalowania wielkości analogowej.

Name	Type	& Reference	Scope	Constant	Retain	Replicable
Skalowanie		<input type="checkbox"/>				<input checked="" type="checkbox"/>
VALUE	INT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MIN_VALUE	INT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MAX_VALUE	INT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MIN_SCALE	INT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MAX_SCALE	INT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SCALE	INT	<input type="checkbox"/>	VAR_OUTPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NORM	REAL	<input type="checkbox"/>	VAR	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Rysunek 52 Deklaracja zmiennych lokalnych bloku funkcyjnego

```
FUNCTION_BLOCK Skalowanie
    NORM := (INT_TO_REAL(VALUE - MIN_VALUE) / (MAX_VALUE - MIN_VALUE));
    SCALE := REAL_TO_INT(NORM * (MAX_SCALE - MIN_SCALE) + MIN_SCALE);
END_FUNCTION_BLOCK
```

Rysunek 53 Przykładowy kod programu w języku ST do skalowania wielkości wejściowej.

## Ćwiczenie 7

Przeskaluj wartości zmiennej powiązanej z wejściem analogowym, tak aby na panelu HMI wyświetlane było napięcie jakie znajduje się na wyjściu potencjometru podłączonego do sterownika PLC.