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

# LABORATORIUM PRZEMYSŁOWYCH SYSTEMÓW STEROWANIA

**Wydział Inżynierii Mechanicznej i Robotyki**

**Katedra Automatyzacji Procesów**

**Przedmiot:**

**Przemysłowe Systemy Sterowania (PSS)**

**Laboratorium 11:**

Zapoznanie się ze sterownikami PLC firmy B&R

**Kraków**

## Cel ćwiczeń laboratoryjnych

1. Zapoznanie z architekturą sterowników PLC oraz sposobem mapowania jego wejść i wyjść.
2. Poznanie sposobu edytowania konfiguracji sprzętowej.
3. Poznanie mapy pamięci sterownika PLC.
4. Realizacja programowa funkcjonalności przerzutników z priorytetem na SET i RESET z zastosowaniem instrukcji **SET/RESET** oraz gotowych bloków. Zapoznanie z cyklem pracy sterownika.
5. Realizacja programu z użyciem podstawowych bloków liczników oraz zegarów.

## Po ukończeniu zajęć student powinien potrafić:

1. Samodzielnie edytować konfigurację sprzętową sterownika PLC i mapować wejścia/ wyjścia do zmiennych dla każdego stanowiska laboratoryjnego
2. Potrafić opisać najważniejsze obszary pamięci sterownika przemysłowego.
3. Zrealizować programowo funkcjonalność przerzutnika z priorytetem na SET i RESET.
4. Wykonać proste programy z użyciem zegarów oraz liczników.

# Wstęp

Sterowniki firmy Bernecker & Rainer (B&R) programowane są za pomocą programu Automation Studio. To zintegrowane narzędzie wykorzystywane jest ponadto do programowania paneli operatorskich oraz serwonapędów firmy B&R.

Oprogramowanie obejmuje również szeroką gamę języków programowania, narzędzi diagnostycznych i edytorów, ułatwiającym inżynierom rozwijanie projektów. Standardowe biblioteki dołączone przez B&R oraz integracja języków programowania w standardach IEC zapewniają wysoce sprawną realizację pracy nad projektami o dowolnej wielkości.

Sterowniki B&R mogą być programowane w językach programowania sterowników PLC określonych w normie IEC 61131-3 czyli:

- Ladder Diagram (LD)
- Instruction List (IL)
- Function Block Diagram (FBD)
- Sequential Function Chart (SFC)
- Structured Text (ST)

Ponadto użytkownik ma możliwość pisania programu wykorzystując języki:

- ANSI C/C++
- B&R Automation Basic (AB)
- Continuous Function Chart (CFC)

Przy programowaniu sterowników można używać dwóch podstawowych rodzajów zmiennych:

- Zmienne lokalne (*local variables*)- są to zmienne które obowiązują w obrębie danego programu.
- Zmienne globalne (*global variables*)- są to zmienne które obowiązują w obrębie całego projektu.

Deklarując zmienne możemy korzystać z następujących typów danych:

| Typ | Zakres wartości | Opis |
|---|---|---|
| BOOL | TRUE (1) lub FALSE(0) | Typ reprezentujący jeden bit |
| SINT | -128… +127 | Liczby całkowite w określonym zakresie |
| INT | -32768 … +32767 | Liczby całkowite w określonym zakresie |
| DINT | -2147483648 … +2147483647 | Liczby całkowite w określonym zakresie |
| USINT | 0 … 255 | Liczby całkowite dodatnie w określonym zakresie |
| UINT | 0 … 65535 | Liczby całkowite dodatnie w określonym zakresie |
| UDINT | 0 … 4294967295 | Liczby całkowite dodatnie w określonym zakresie |
| REAL | -3.4E38 ... +3.4E38 | Liczby zmiennoprzecinkowe w określonym zakresie |
| LREAL | -1.79769313486231E308 … +1.79769313486231E308 | Liczby zmiennoprzecinkowe w określonym zakresie |
| TIME | T#-24d_20h_31m_23s_648ms … T#24d_20h_31m_23s_647ms | Typ używany do przechowywania określonej wartości czasu |
| TIME_OF_DAY | TOD#00:00:00.000 … TOD#23:59:59.999 | Typ używany do przechowywania określonej godziny |
| DATE_AND_TIME | DT#1970-01-01-00:00:00 … DT#2106-02-07-06:28:15 | Typ używany do przechowywania daty i godziny |
| STRING | - | Typ używany do przechowywania znaków ASCII. Jest to łańcuch bajtów z których każdy reprezentuje jeden znak. |
| BYTE | Sekwencja 8 bitów | - |
| WORD | Sekwencja 16 bitów | - |
| DWORD | Sekwencja 32 bitów | - |
| WSTRING | - | Typ używany do przechowywania znaków UNICODE. Jest to łańcuch bajtów z których każde dwa reprezentują jeden znak. |

## Rodzaje pamięci sterownika PLC:

**UserRAM-** Miejsce w pamięci gdzie przechowywane są obiekty używane w trakcie wykonywania programu użytkownika. Nie jest kasowana przy tzw. *WARM RESTART*, ale zostaje wymazana przy tzw. *COLD RESTART*. Miejsce w tej pamięci można sprawdzić klikając na pasku menu *Online -> Info*.

**UserROM**- Miejsce w pamięci gdzie przechowywany jest program użytkownika. Pozostałe miejsce można sprawdzić klikając na pasku menu *Online -> Info.*

**FixRAM-** Miejsce w pamięci które może być skonfigurowane zarówno jako UserROM jak i UserRAM. Pamięć ta nie jest kasowana w trakcie tzw. COLD RESTART. Miejsce pozostałe w tej pamięci sprawdzić można klikając na pasku menu *Online-> Info*.

**MemCard-** Niektóre urządzenia wyposażone są w specjalną kartę pamięci na której można przechowywać dowolne dane.

**SystemROM-** W tym miejscu przechowywany jest system operacyjny urządzenia.

## Pamięć DRAM i SRAM

Pamięć RAM jest pamięcią szybkiego zapisu i odczytu, do której można ładować dane do wykonania Automation Runtime i aplikacji Automation Studio.

**DRAM (DynamicRAM)-** DRAM to pamięć RAM której stan po uruchomieniu jest nieokreślony. Podczas ładowania systemu, Automation Runtime wczytuje wszystkie niezbędne moduły BR do pamięci DRAM, aby umożliwić szybki do nich dostęp. Moduł BR w pamięci RAM wymaga skonfigurowanego w Automation Studio obszaru pamięci dla zmiennych lokalnych lub globalnych oraz jest odpowiedzialny za inicjalizację tego obszaru pamięci. Dane z tej pamięci są kasowane jedynie kiedy użytkownik użyje przycisku RESET wbudowanego w urządzenie, lub kiedy dojdzie do całkowitego braku zasilania (również tego z wbudowanej baterii).

**SRAM (StaticRAM)-** W przeciwieństwie do pamięci DRAM, SRAM (statyczna pamięć RAM) jest buforowana (podtrzymywana) przez baterię. Dzięki temu dane w niej przechowywane nie zostaną utracone w razie zaniku zasilania. Wymaga to nieustannej sprawności baterii.

**Tryby pracy sterownika PLC:**

- Tryb Serwisowy ***SERV.*** W tym trybie wyjścia sterownika są wyzerowane, a program nie jest wykonywany. Urządzenie automatycznie uruchamia się w tym trybie przy wgrywaniu nowego programu. Jest to też tryb awaryjny w który sterownik przejdzie po pojawieniu się błędu lub kiedy nastąpi zwarcie na linii zasilania. Można go uzyskać manualnie wciskając przycisk *STOP TARGET* w menu pracy *online*. Aby go opuścić należy przeprowadzić restart urządzenia
- Tryb pracy ***RUN*** jest wykorzystywany w trakcie normalnej pracy. Nie ma tu możliwości wprowadzania jakichkolwiek zmian.
- Tryb diagnostyczny ***DIAG.*** Tryb ten pozwala na zweryfikowanie różnice w konfiguracji i oprogramowaniu między projektem a sterownikiem. Można go wywołać z paska menu *Online-> Services-> Diagnostic*.

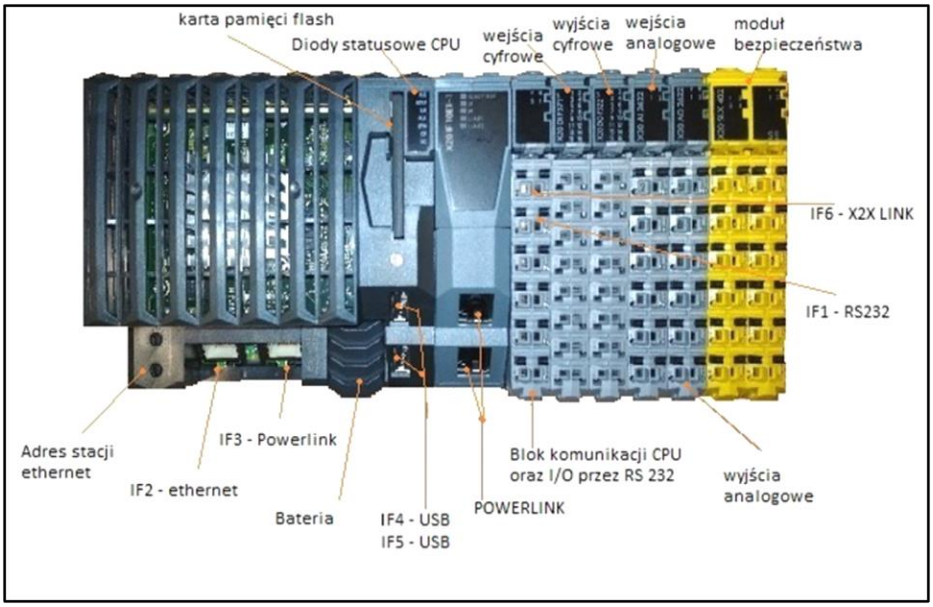
**Sterownik można zrestartować na dwa sposoby.**

Wykonując tzw. *COLD RESTART* wymazujemy pamięć RAM, a oprogramowanie sterownika zostaje przywrócone do pierwotnej wersji. Taki restart powoduje że sterownik uruchamia się w taki sposób jakby uruchamiał się po raz pierwszy.

*WARM RESTART* nie przywraca oprogramowania do wersji pierwotnej, a wszystkie dane zapisane w pamięci RAM zostają zachowane. Taki restart wykonuje się automatycznie po każdym transferze nowego programu

# Ćwiczenie 1

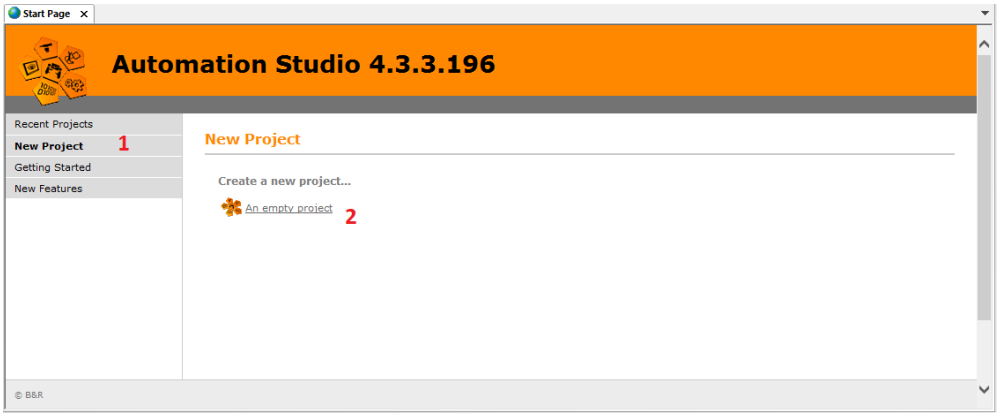
Zapoznaj się z architekturą sterowników przemysłowych PLC firmy B&R znajdujących się na stanowisku.




Rysunek 1. Sterownik firmy B&R X20CP1584

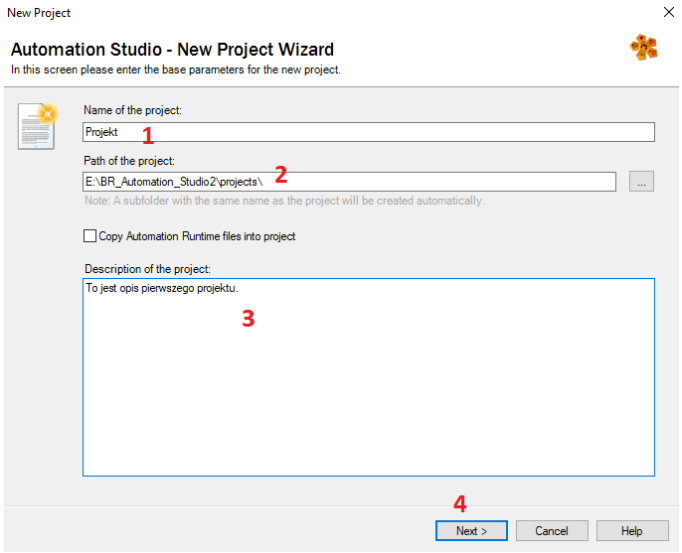
Powyższe zdjęcie prezentuje sterownik wraz z modułami obecny na stanowiskach w laboratorium. Aby lepiej zaznajomić się z architekturą sterownika utworzymy jego konfigurację w programie Automation Studio.

Aby przystąpić do pracy ze sterownikami B&R potrzebujemy otworzyć oprogramowanie Automation Studio. Po otworzeniu oprogramowania widzimy stronę startową środowiska. W celu utworzenia nowego projektu naciskamy „New Project" znajdujący się po lewej stronie strony startowej




Rysunek 2. Strona startowa środowiska Automation Studio

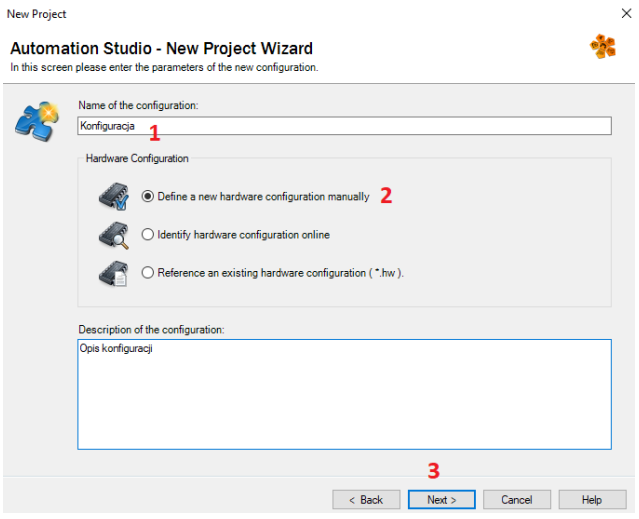
Następnie ukaże się okno kreatora nowego projektu, w którym trzeba wprowadzić nazwę projektu(1), jego ścieżkę na dysku(2) oraz opcjonalnie komentarz(3).




Rysunek 3. Kreator nowego projektu

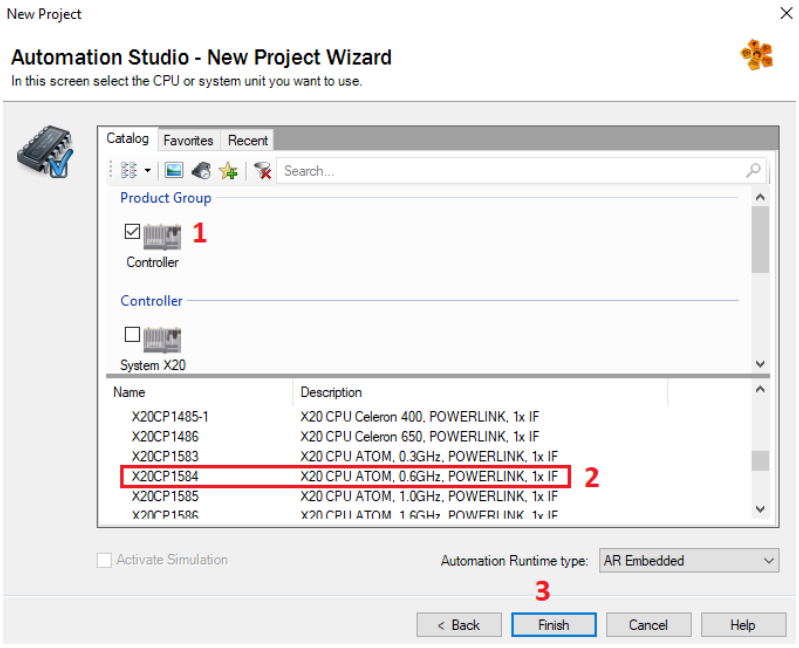
Środowisko Automation Studio umożliwia utworzenie własnej konfiguracji sprzętowej. Możliwe jest jednak również identyfikacja konfiguracji *online*, komputer będący w jednej sieci ze sterownikiem PLC automatycznie pobiera informację o podłączonym sterowniku oraz jego dodatkowych modułach. Na początek wprowadzimy ręcznie sterownik znajdujący się na stanowisku.

W oknie należy podać nazwę konfiguracji (1), zaznaczyć opcję utworzenia nowej konfiguracji(2) oraz zatwierdzić wybór przyciskiem *Next* (3).



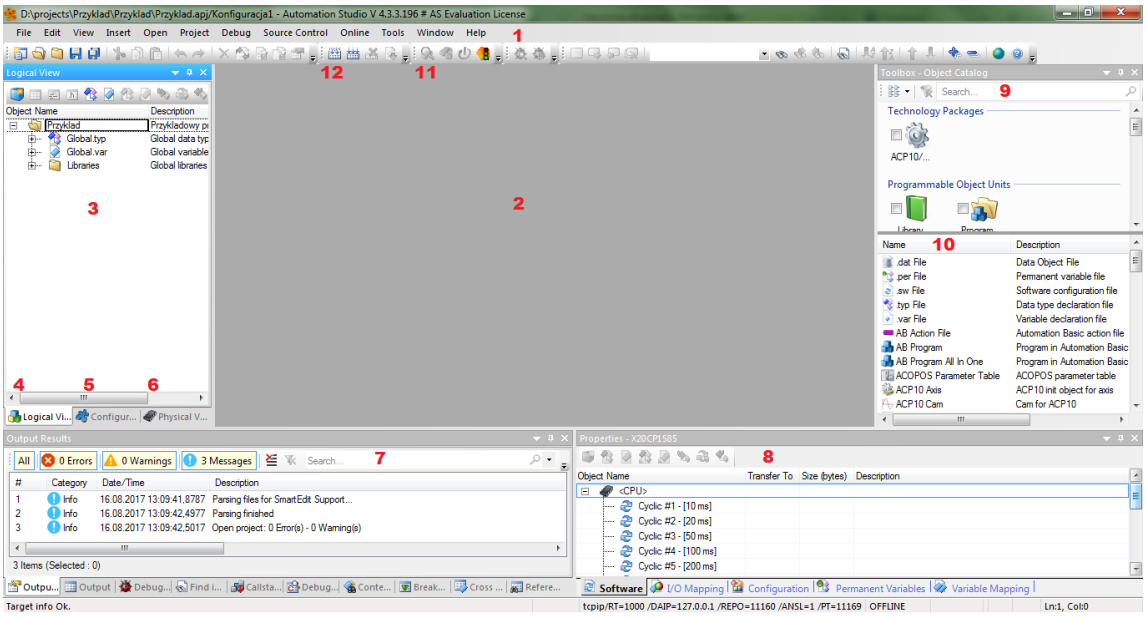

Rysunek 4. Okno konfiguracji sterownika

W tym celu wybieramy kategorię sterowników PLC *Controller*. Po wybraniu odpowiedniego sterownika (na stanowisku znajduje się X20CP1584), wybór zatwierdzamy zaznaczając sterownik oraz naciskając przycisk *Finish*. Sterownik można również dodać, wprowadzając jego model bezpośrednio w polu *Search*...




Rysunek 5. Wybór sterownika w nowej konfiguracji

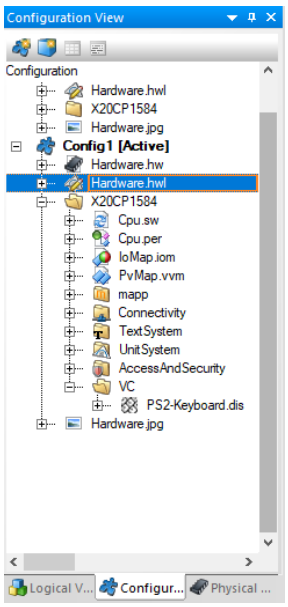
Po utworzeniu projektu w programie *Automation studio* wersji 4.0 lub wyższej, twoim oczom ukaże się następujące okno:




Rysunek 6. Okno projektu w "Automation studio" (wersja 4.0 lub wyższa)

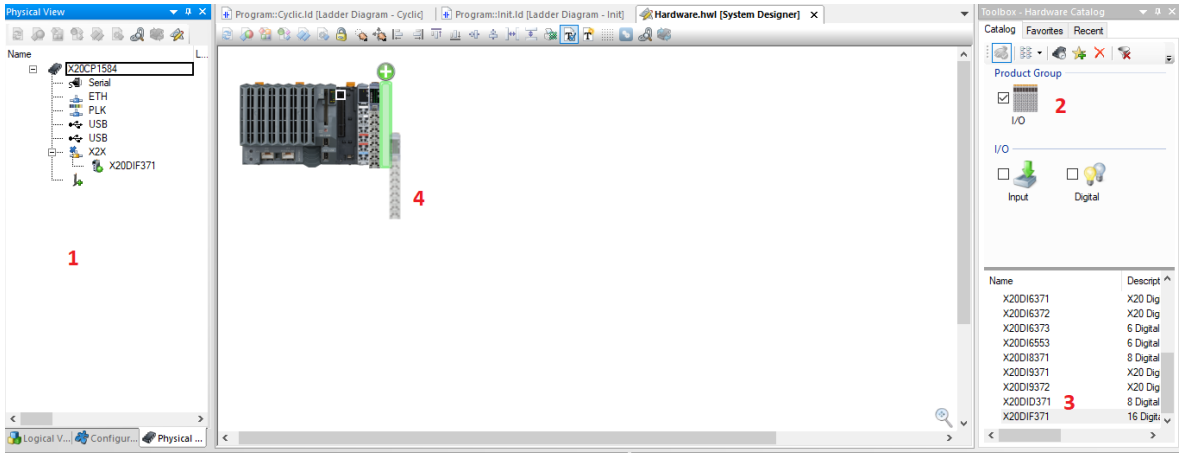
1. Pasek menu.
2. Obszar roboczy.
3. Menu robocze aktywnego projektu.
4. Zakładka *Logical View* zawierająca drzewko programu wraz z bibliotekami, zmiennymi etc.
5. Zakładka *Configuration View* zawierająca dane dotyczące aktualnej konfiguracji sprzętowej.
6. Zakładka *Physical View* pozwalająca na edycję konfiguracji sprzętowej, oraz na konfigurację parametrów sprzętu.
7. Pasek zawierający informacje na temat aktualnej kompilacji (Błędy, ostrzeżenia, etc.)
8. Pasek właściwości, jego zawartość zmienia się wraz ze zmianą aktywnego obszaru.
9. *Toolbox* do wyboru elementów. Zależnie od wybranej zakładki (4, 5 lub 6) jego zawartość pozwala na wybór i dodanie do projektu różnych elementów (moduły sprzętowe, wizualizacje, programy, bloki funkcyjne, etc.)
10. Przeglądarka elementów. Po wybraniu kategorii w oknie 9, przeglądarka wyświetli dostępne elementy.
11. Szybie menu do zarządzania aktualnie podpiętym sterownikiem.
12. Szybkie menu do sprawdzania i kompilowania projektu.

Program „Automation studio" jest dedykowanym środowiskiem producenta do programowania sterowników PLC firmy B&R. Po rozpoczęciu pracy nad nowym projektem mamy możliwość dodania modułów w które wyposażony jest sterownik. Widok sterownika wraz z dodanymi modułami uzyskamy przechodząc do zakładki *Configuration View* oraz naciskając dwukrotnie na ikonę *Hardware.hwl*



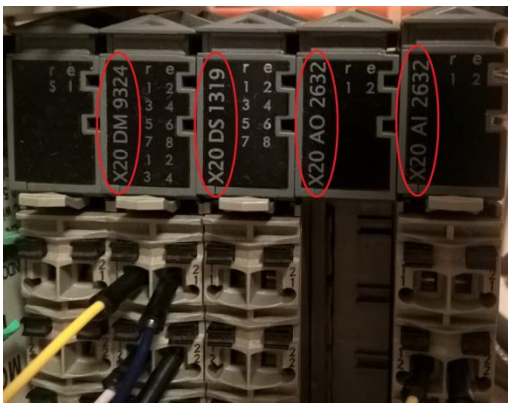

Rysunek 7 Przejście do widoku konfiguracji projektu

Wybierz zakładkę *PhysicalView(1), toolbox* wyboru elementów zmieni nazwę na *Hardware Catalog*. Wybierz katalog *I/O(2)* A następnie kolejno odszukaj na liście poniżej moduły przyłączone do twojego urządzenia(3). Dołącz je do niego przeciągając je lewym przyciskiem myszy do obszaru roboczego w okolice sterownika PLC(4). Puste miejsce obok urządzenia podświetli się na zielono co oznacza że moduł może być dołączony w tym miejscu.




Rysunek 8 Dodawanie modułów do konfiguracji

Informacje na temat modułów przyłączonych do sterownika PLC na twoim stanowisku znajdziesz badając urządzenie.

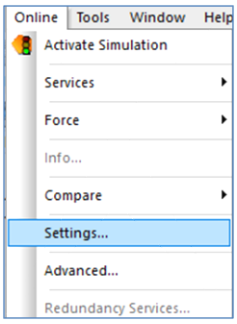



Rysunek 9 Oznaczenia modułów dołączonych do sterownika PLC

## Identyfikacja konfiguracji sterownika online

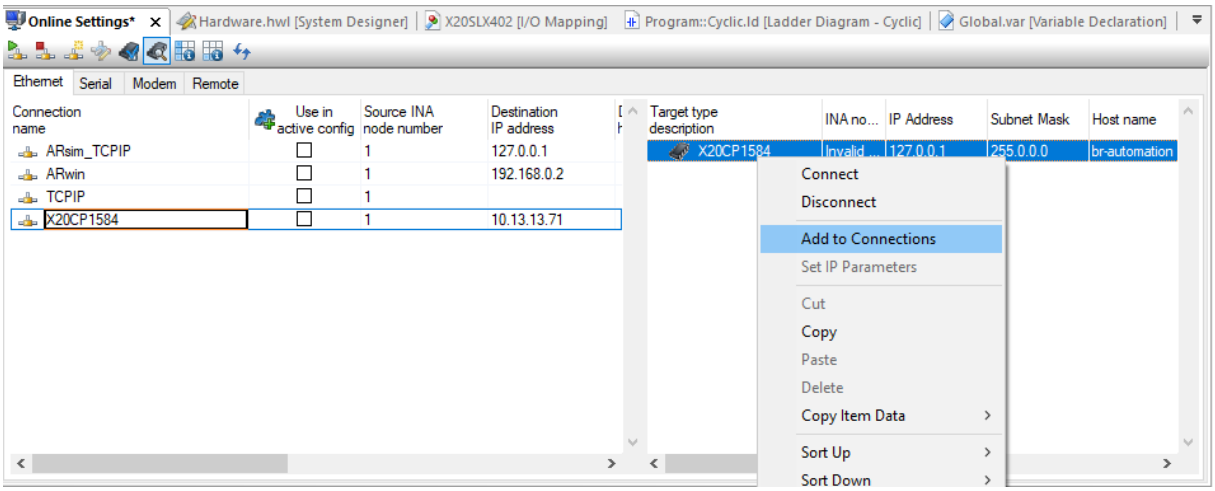
Aby ułatwić i przyśpieszyć pracę, nie trzeba dodawać ręcznie całej konfiguracji sterownika. W Automation Studio istnieje możliwość zidentyfikowania konfiguracji *online*.

Aby pobrać konfigurację sterownika, wejdźmy zakładkę *Online* w menu Automation Studio następnie wybierzmy *Settings*…




Rysunek 10. Ustawienia online projektu

Ukaże się nam okno z ustawieniami połączenia ze sterownikiem PLC. Klikając ikonę *Browse* jesteśmy w stanie przeszukać sieć w poszukiwaniu dostępnych sterowników PLC. Pokaże się nam lista sterowników wraz z ich adresem IP. Adres urządzenia posiada część dziesiętną związaną z numerem stanowiska oraz częścią jedności równą 1 dla sterownika PLC oraz 2 dla panelu HMI. W naszym oknie ukażą się tylko i wyłącznie sterowniki PLC.




Rysunek 11. Ustalenie połączenia ze sterownikiem online

Znajdując sterownik należący do danego stanowiska należy nacisnąć prawy przycisk myszy i kliknąć opcję *Add to Connection* a następnie *Connect*. W tym momencie na dole projektu powinien ukazać się pasek informujący o podłączeniu *online* ze sterownikiem:

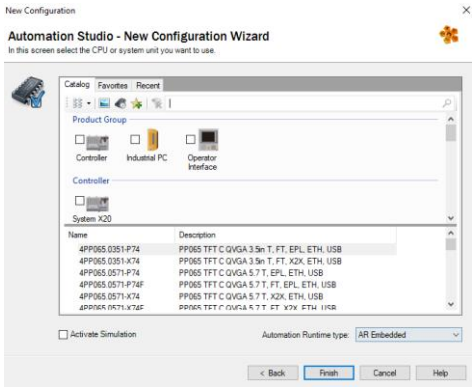

Informacja ta przedstawia nazwę połączenia w raz z IP sterownika oraz trybem, w którym aktualnie znajduje się sterownik.

## Uwaga

Nie należy łączyć się ze sterownikiem od razu po otworzeniu środowiska Automation Studio, aby proces przebiegł pomyślnie, należy postępować zgodnie w przedstawionymi wytycznymi.

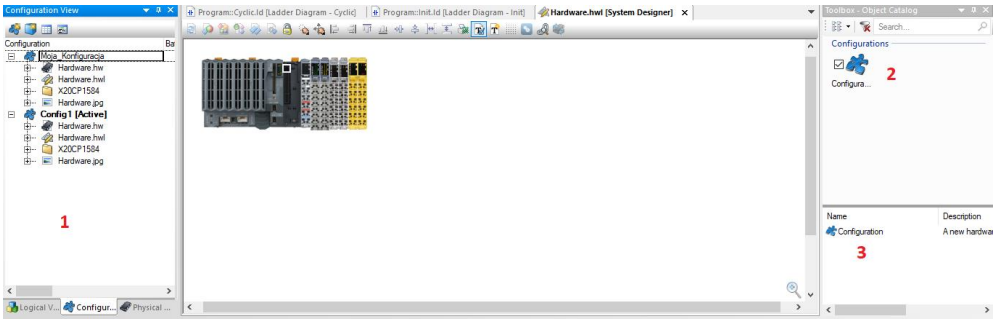
Połączenie się ze sterownikiem jest możliwe, jeżeli posiadamy otwarty projekt w programie Automation studio. W naszym przypadku posiadamy już utworzony projekt z własnoręcznie dobraną konfiguracją. Jednak aby przyśpieszyć proces wystarczy utworzyć nawet pusty projekt, nie jest konieczne deklarowanie sterownika PLC.

W celu stworzenia pustego obiektu nie potrzeba uzupełniać okna konfiguracji a jedynie nacisnąć przycisk *Finish*.



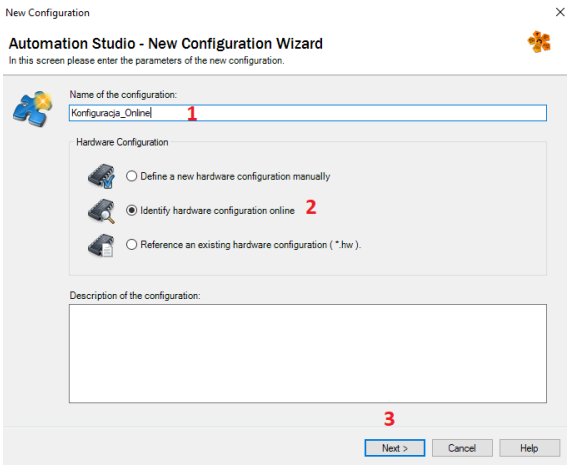

Rysunek 12 Tworzenie nowej konfiguracji

W oknie głównym pojawia się nam symbol panelu, nie jest to istotne, gdyż następnym naszym krokiem jest zidentyfikowanie konfiguracji *online* łącząc się ze sterownikiem PLC znajdującym się na stanowisku. Aby dodać automatycznie konfigurację, należy zaznaczyć zakładkę *Configuration View* oraz z *Toolboxa* przeciągnąć do niej ikonę *Configuration.*



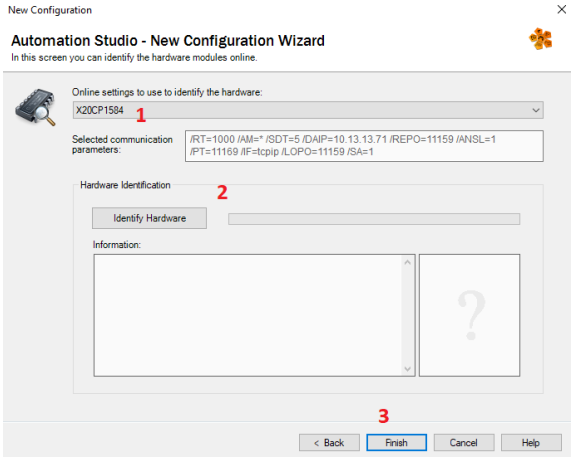

Rysunek 13. Dodanie nowej konfiguracji

Ukaże się nam znajome okno kreatora konfiguracji, tym razem zaznaczymy opcję *Identify hardware configuation online.*



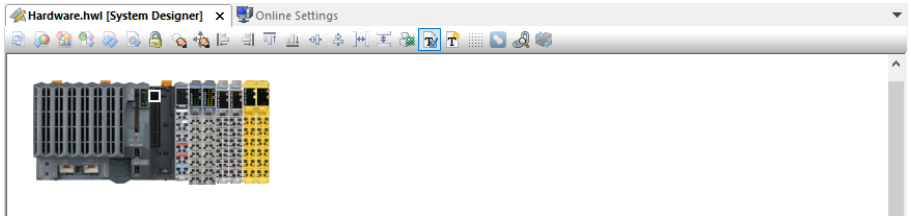

Rysunek 14. Okno konfiguracji PLC

Należy wybrać połączenie(1), które dodaliśmy w ustawieniach sieciowych, domyślnie jego nazwa będzie nazwą sterownika znajdującego się na stanowisku. Po naciśnięciu przycisku *Identify Hardware*(2), w oknie poniżej powinna ukazać się konfiguracja sterownika PLC znajdującego się na stanowisku laboratoryjnym. Naciśnięcie przycisku *Finish*(3) dodaje zidentyfikowaną konfigurację do projektu Automation Studio.




Rysunek 15. Identyfikacja konfiguracji sterownika

W oknie głównym projektu widoczna jest dodana konfiguracja sterownika obecnego na stanowisku.




Rysunek 16. Zidentyfikowana konfiguracja stanowiska laboratoryjnego

Takim sposobem w środowisku Automation Studio widzimy konfigurację naszego stanowiska laboratoryjnego. Możemy dokładnie zapoznać się z jego budową i funkcjami poszczególnych modułów. Klikając dwukrotnie na moduł mamy do dyspozycji wszystkie zmienne związane z tym modułem, otwierając właściwości dodatkowo możemy skonfigurować software'owo moduły w zależności od wymagań projektu.

## Ćwiczenie 2

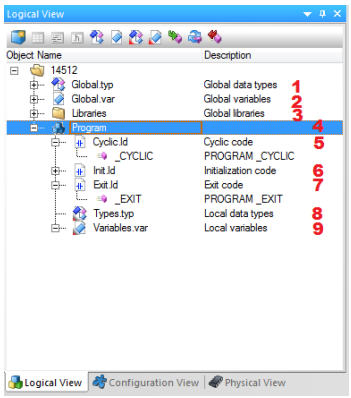
Utwórz nowy projekt w środowisku Automation Studio oraz dodaj do projektu konfigurację sterownika znajdującego się na Twoim stanowisku w oparciu o powyższe instrukcje. Zapoznaj się z właściwościami elementów obecnych na stanowisku.

## Tworzenie programu w środowisku Automation Studio

Aby dodać program w języku drabinkowym do naszego projektu należy zaznaczyć zakładkę *Logical View*, toolbox po prawej stronie ekranu zmieni nazwę na *Object Catalog*.


Rysunek 17. Wybór programu w Toolboxie

Zaznacz katalog *Program* i z listy dostępnych obiektów która wyświetli się poniżej wybierz *LD Program* a następnie przeciągnij go i upuść w zakładce *Logical View*. Twoje drzewo projektu powinno teraz wyglądać następująco:




Rysunek 18. Drzewo projektu

1. Globalne typy danych.
2. Zmienne globalne- zmienne zdefiniowane tutaj będą dostępne dla wszystkich programów w obrębie sterownika. Zmienne globalne można przypisać do wyjść/wejść modułów sterownika.
3. Biblioteki dołączone do projektu. Biblioteki dodajemy analogicznie jak konfigurację sprzętową i programy (przeciągając z odpowiedniego katalogu z toolboxa po prawej stronie ekranu). To jakie biblioteki będą wymagane w projekcie zależy jedynie od programisty.
4. Aktualnie edytowany program- znajdują się tutaj wszystkie segmenty tworzące program (Init, Cyclic, Exit) jak i dodatkowe funkcje i bloki funkcyjne utworzone przez użytkownika, oraz zmienne lokalne i lokalne typy danych.
5. Główna część programu, ten fragment programu wykonuje się cyklicznie w trakcie pracy sterownika PLC. Umieszczamy w nim kod według którego sterownik ma pracować.
6. Kod wykonujący się tylko raz, przed startem głównego programu.
7. Program wykonujący się tylko raz, po zakończeniu głównego programu.
8. Lokalne typy zmiennych obowiązujące w obrębie danego programu.
9. Zmienne lokalne obowiązujące w obrębie danego programu, można je również przypisać do wejść/wyjść sterownika.

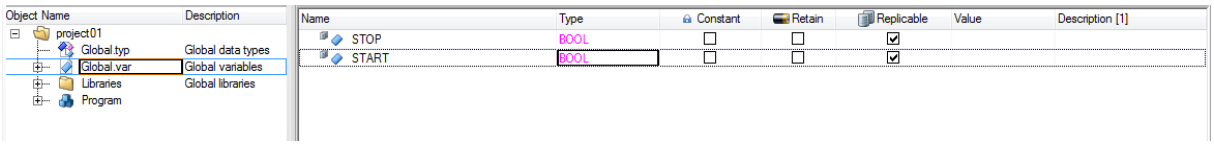
Po napisaniu programu należy go skompilować używając przycisków z paska menu:

Rysunek 19. Kompilacja projektu

W programie można dokonać kompilacji za pomocą dwóch ikon:
*Build(F7)*–kompiluje całą aktywną konfigurację, która uległa zmianie od ostatniej kompilacji
*Rebuild(Ctrl+F7)* –kompiluje całą aktywną konfigurację, bez względu na to, czy była zmieniana.

Aby móc dokonywać odczytu wejść i modyfikować stan wyjść sterownika PLC należy przypisać zmienne globalne lub lokalne do odpowiednich wyjść i wejść modułów w tym celu należy utworzyć zmienne klikając dwukrotnie na ikonę *Global.var* lub dla zmiennych lokalnych *Variables.var* . Po kliknięciu prawym klawiszem myszki należy wybrać opcję *Add variable* a następnie po podaniu nazwy, w drugiej kolumnie wpisać lub wybrać pożądany typ.

W naszym wypadku będą to zmienne logiczne (*BOOL*).

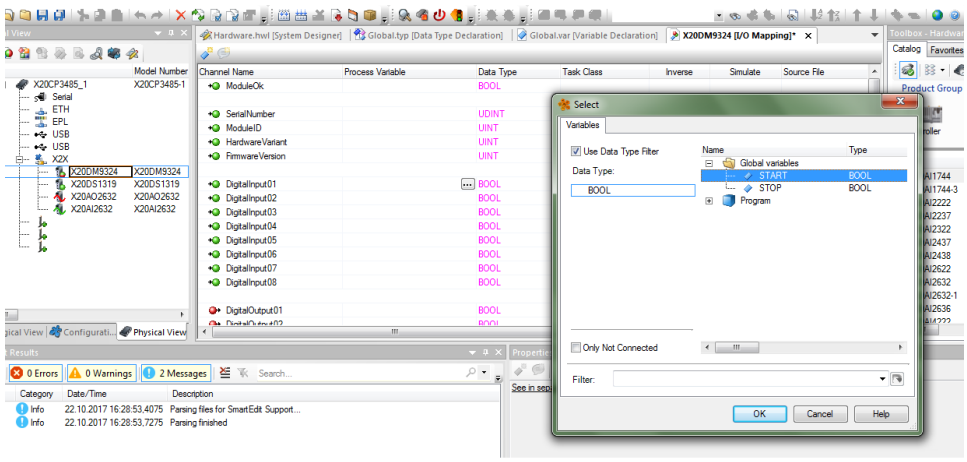



Rysunek 20. Zmienne globalne START i STOP

Następnie przejdź do zakładki *Physical View* i z listy połączonych ze sterownikiem modułów wybierz ten, do którego są podpięte przewody prowadzące od odpowiednich fizycznych przycisków. Następnie dwukrotnie kliknij pole *Proces variable* przy odpowiednim wejściu/ wyjściu i z listy wybierz pożądaną zmienną.

## Uwaga

Jeżeli nie możesz odnaleźć odnaleść na liście, zamknij okno które się pojawiło, a następnie zapisz projekt przyciskiem *Save All* z paska menu i spróbuj ponownie. Zmienne utworzone, a nie zapisane nie są dostępne w innych częściach programu.



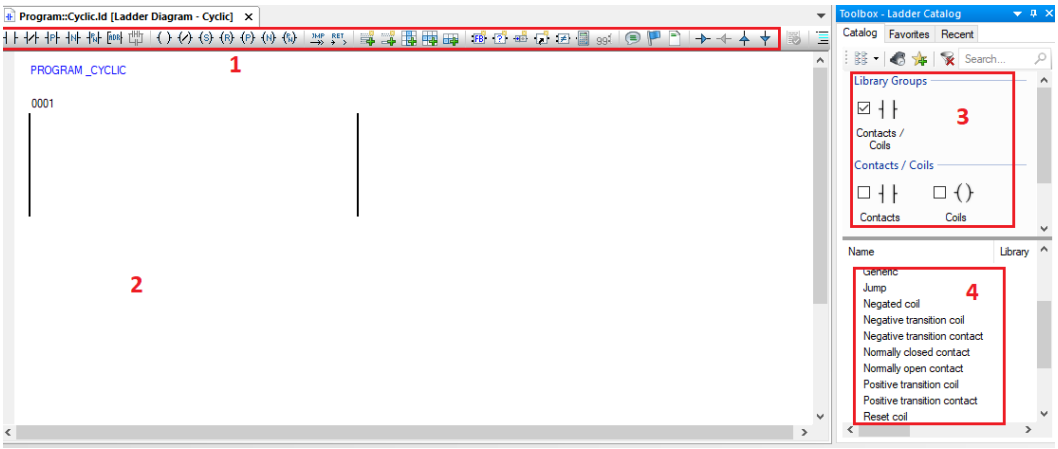

Rysunek 21. Mapowanie zmiennych do wejść sterownika

Po przypisaniu zmiennych do wejść lub wyjść sterownika PLC, zmiana wartości zmiennej wyjściowej będzie powodowała fizyczną zmianę wartości na wyjściu sterownika, a zmiana fizycznej wartości wejść będzie powodowała zmianę wartości zmiennych wejściowych. Dwukrotnie klikając w program *_CYCLIC* w głównym oknie projektu ukaże się kreator do tworzenia programu, zawiera on różne funkcje, zależnie od języka którym się posługujemy.

Struktura programu w  języku LD  wygląda następująco:


Rysunek 22 Struktura jednej linii kodu w języku drabinkowym

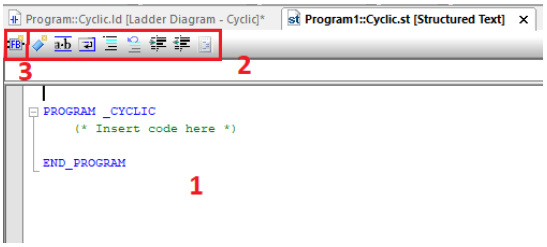
Aby dodać elementy do programu możemy posłużyć się menu w górnej części (1), klikając poszczególny element pojawi się on w miejscu w którym mamy aktywny znacznik w programie. Klikając w obszar roboczy programu (2) zawierającym podstawowe elementy języka, w oknie *toolboxa* pojawią się dostępne elementy języka ich dodanie jest możliwe po zaznaczeniu odpowiedniej grupy(3) i przeciągnięciu wybranego elementu(4) do żądanej linii programu w oknie głównym.




Rysunek 23 Dodawanie elementów języka drabinkowego

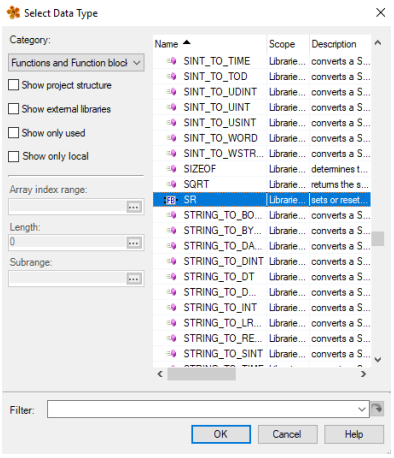
Do każdego nowego elementu programu musimy  przypisać określoną zmienną lokalną bądź globalną wpisując jej nazwę nad elementem. Po wstawieniu symbolu w umieszczone nad nim pole tekstowe wpisz odpowiednią nazwę symboliczną. Łączenie i rozgałęzianie elementów schematu można wykonywać za pomocą ikon  po umieszczeniu kursora i kliknięciu w odpowiednim miejscu schematu.

W przypadku, gdy do projektu dodajemy program w języku tekstowym, ukazuje się nam taki sam kreator, niezależnie z którego języka tekstowego korzystamy.



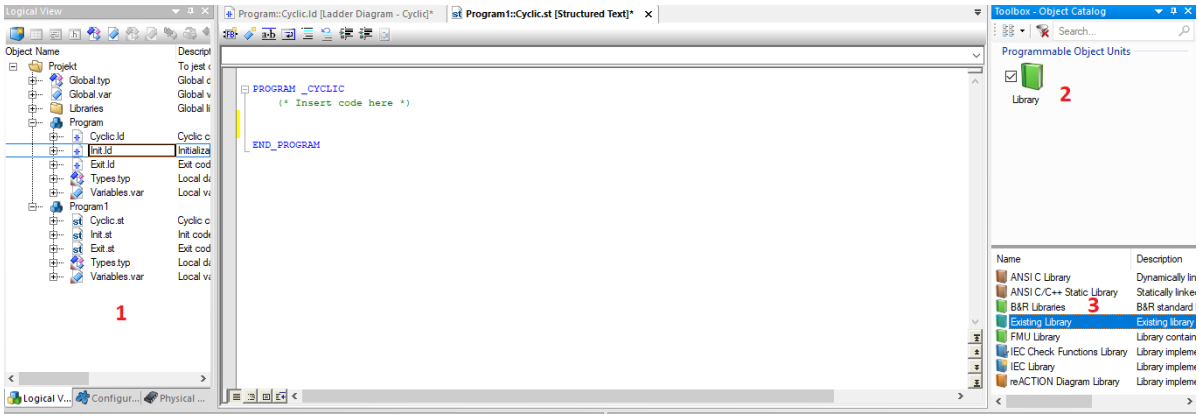

Rysunek 24. Widok okna programów tekstowych

W jego skład wchodzi główne okno programu(1), w którym umieszczamy nasz kod w języku tekstowym oraz górne menu(2), pozwalające formatować nasz tekst, dodawać komentarze oraz zmienne lub bloki funkcyjne. Klikając na ikonę *Insert Function/Function Block (3)* wyświetli się nam lista wszystkich funkcji i bloków funkcyjnych, których możemy użyć w projekcie.




Rysunek 25. Lista funkcji i bloków funkcyjnych dostępnych w projekcie

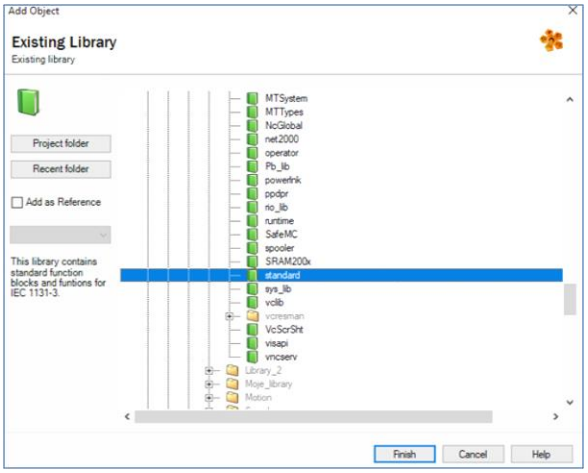
W większości bardziej złożonych programów będzie potrzeba dodatkowych funkcji. Środowisko Automation Studio umożliwia dodanie do projektu bibliotek rozszerzających funkcjonalność naszego programu. Aby dodać bibliotekę do projektu należy w *Toolboxie* wybrać ikonę *Library*.




Rysunek 26. Dodawanie bibliotek do projektu w Automation Studio

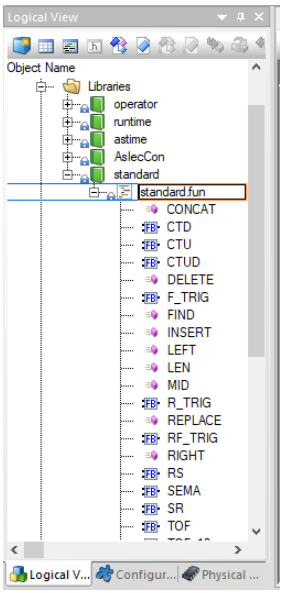
Automation Studio ma swoją bazę bibliotek, w tym celu wybieramy opcję Existing Library którą przeciągamy do folderu Libraries w zakładce Logical View.

Konieczne jest wskazanie folderu, w którym przechowywane są biblioteki Automation Studio (domyślnie C:\ – Program Files – Brautomation – AS – Library). Wchodząc w ten folder widzimy mnóstwo wbudowanych bibliotek dostępnych dla użytkownika, jedną z najczęściej używanych, zawierających podstawowe funkcje jest biblioteka standard.h, zaznaczając ją i naciskając przycisk Finish biblioteka zostaje dodana do naszego projektu i możemy korzystać w zadeklarowanych w niej funkcji.




Rysunek 27. Wybór biblioteki standard spośród ogólno dostępnych bibliotek

Po dodaniu standardowej biblioteki możemy podejrzeć w projekcie jej zawartość. Rozwijając ikonę dodanej biblioteki w zakładce *Logical View* widzimy funkcję i bloki funkcyjne znajdujące się w jej wnętrzu. Biblioteka ta zawiera podstawowe bloki zegarów i liczników,funkcje umożliwiające identyfikację zboczy oraz m.in. przerzutniki SR i RS.



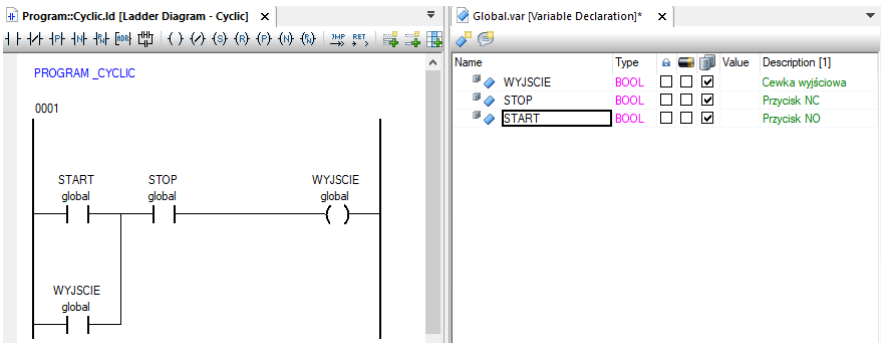

Rysunek 28 Zawartość biblioteki standard

# Ćwiczenie 3

Utwórz program w języku LD oraz ST i zapoznaj się z mapowaniem zmiennych do fizycznych wejść i wyjść modułów sterownika. Dodaj do projektu standardową bibliotekę umożliwiającą dodanie do projektu m.in. zegarów, liczników i przerzutników.

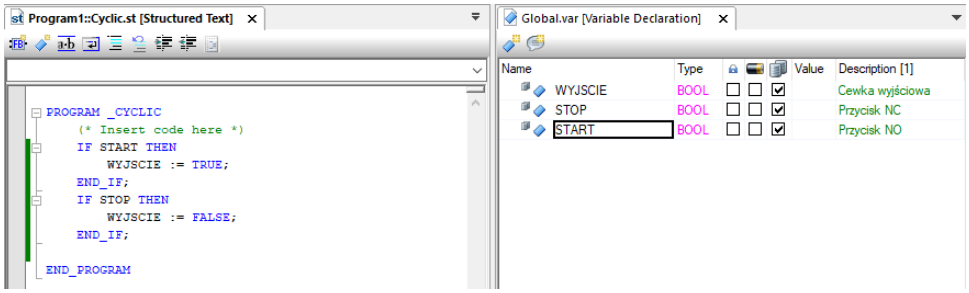
## Realizacja funkcjonalności przerzutnika z priorytetem na SET i RESET

Na postawie powyższych wiadomości możemy przystąpić do zrealizowania programu prostego przerzutnika z odpowiednim priorytetem. Do realizacji zadania potrzebne nam będą zmienne *START*, *STOP* oraz *WYJSCIE*. W przypadku języka LD po otworzeniu programu cyklicznego *_CYCLIC* umieszczamy w nim styki oraz cewki realizujące daną funkcjonalność. Poniżej przedstawiony został program przerzutnika z priorytetem na RESET.



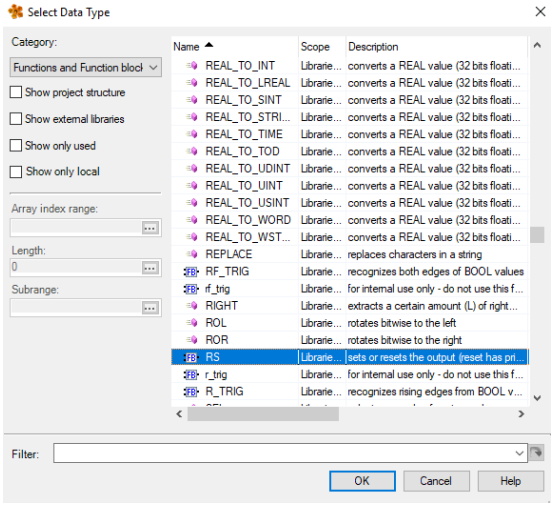

Rysunek 29. Przerzutnik z priorytetem na RESET (LD)

Program realizujący to samo zadanie, napisany w języku ST przyjmuje następującą postać.



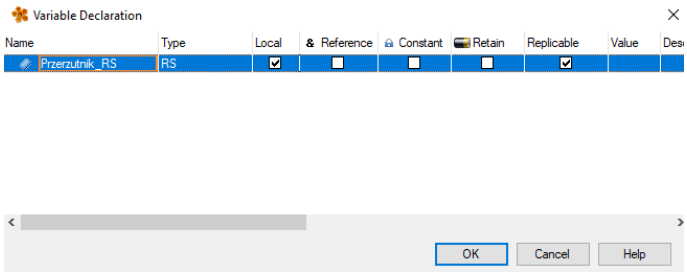

Rysunek 30. Przerzutnik z priorytetem na RESET (ST)

Realizacje przerzutnika możemy także zrealizować z pomocą gotowego bloku funkcyjnego FB dostępnego wraz z biblioteką *standard.* W tym celu w oknie programu naciskamy ikonę  i z listy dostępnych funkcji zaznaczamy np. blok funkcyjny RS.



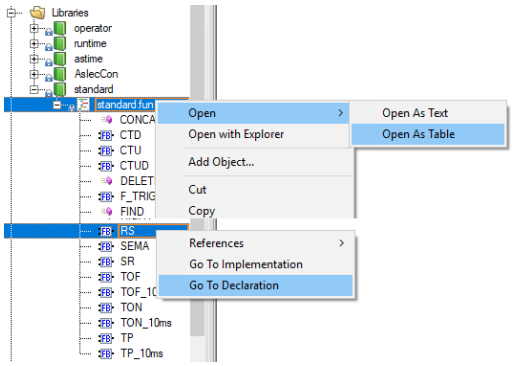

Rysunek 31. Okno wyboru bloku funkcyjnego przerzutnika.

W momencie dodania bloku funkcyjnego do projektu, konieczne jest zdefiniowanie zmiennej, tak zwanej instancji bloku funkcyjnego, przechowuje ona parametry potrzebne do jego pracy.



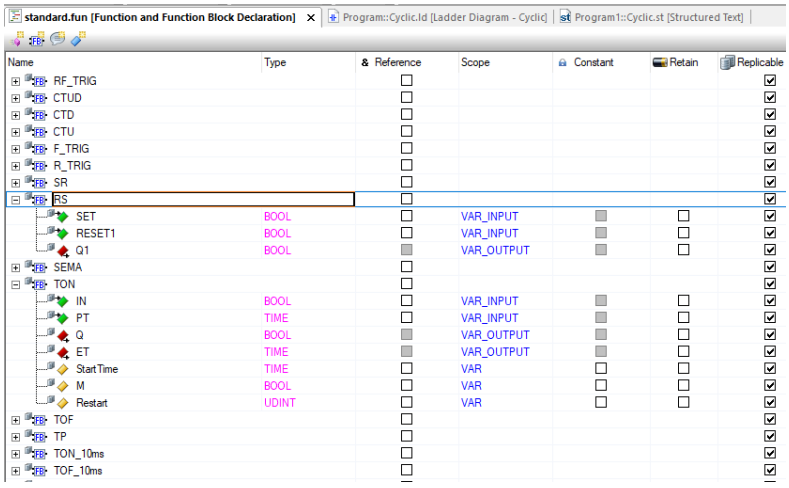

Rysunek 32 Definiowanie instancji bloku funkcyjnego

Możemy sprawdzić jakie zmienne posiada dany blok funkcyjny. Wystarczy w zakładce *Logical View* rozwinąć bibliotekę i otworzyć ikonę z rozszerzeniem *.fun* jako tabelę. Można również nacisnąć prawym przyciskiem na konkretną funkcję lub blok funkcyjny i prawym przyciskiem myszy wybrać opcję *Go to declaration*.




Rysunek 33 Przejście do deklaracji bloków funkcyjnych biblioteki standard

Pojawiają się wtedy deklarację wszystkich funkcji oraz bloków funkcyjnych, mamy dostęp do podglądu do zmiennych występujących w tych blokach.




Rysunek 34 Deklaracje bloków funkcyjnych biblioteki standard

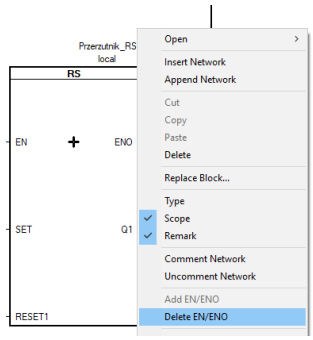
Spośród zmiennych zdefiniowanych w danych blokach możemy wyróżnić:

- VAR_INPUT – Zmienna wejściowa danego bloku funkcyjnego lub funkcji.
- VAR_OUTPUT– Zmienna wyjściowa danego bloku funkcyjnego lub funkcji.
- VAR – zmienna pomocnicza do wykonania działania danego bloku funkcyjnego lub funkcji, niedostępna z poziomu programu.

Możemy w kodzie programu wykorzystać daną zmienną wejściową lub wyjściową, odwołując się do niej za pomocą operatora kropki „.” występującego po nazwie danego bloku funkcyjnego, należy po nim wpisać nazwę zmiennej, do której się odwołujemy.

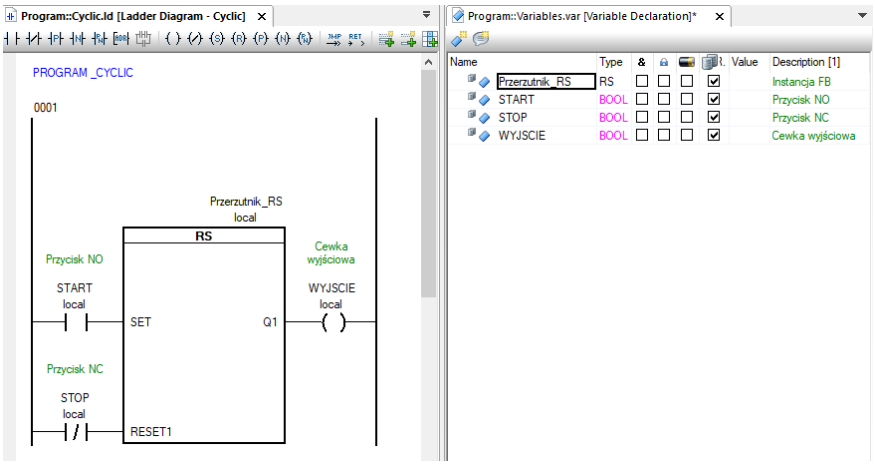
Dodając blok funkcyjny do programu, pojawiają się dwa dodatkowe parametry EN(Enable) i ENO(Enable Output). Pierwszy z nich załącza funkcjonalność bloku, w momencie gdy na jego wejściu pojawi się stan wysoki, w przeciwnym wypadku dany blok jest wyłączony. Parametr ENO wystawia na wyjście stan wysoki, w momencie gdy dany blok zostaje załączony, z jego pomocą możliwe jest kaskadowe załączanie bloków funkcyjnych jeden po drugim.

Jeżeli nie potrzebujemy określać dodatkowych warunków, kiedy dany blok ma być wykonywany, istnieje możliwość usunięcia parametrów EN i ENO bloku klikając prawym przyciskiem myszy na blok i naciskając *Delete EN/ENO*.

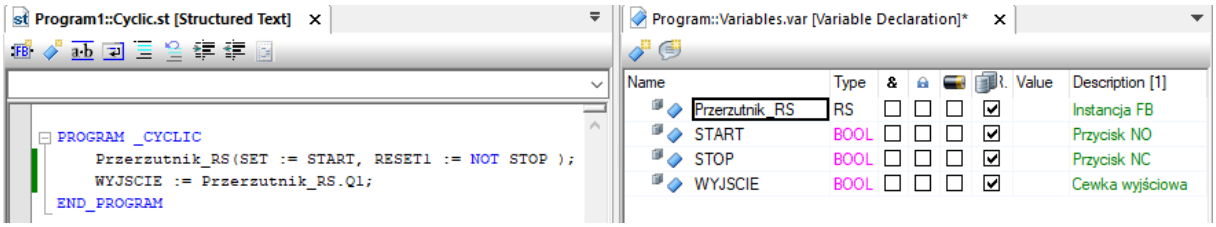



Rysunek 35 Usuwanie parametrów EN/ENO z bloku funkcyjnego

Program realizujący przerzutnik z priorytetem na RESET w języku LD oraz ST wygląda następująco;

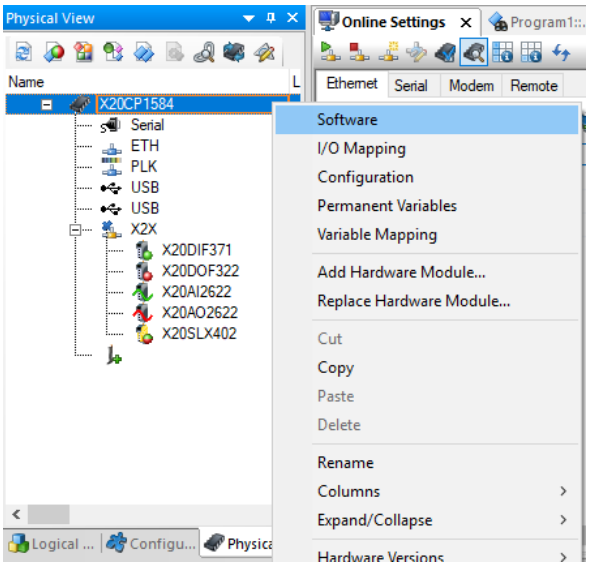



Rysunek 36. Przerzutnik RS (LD)



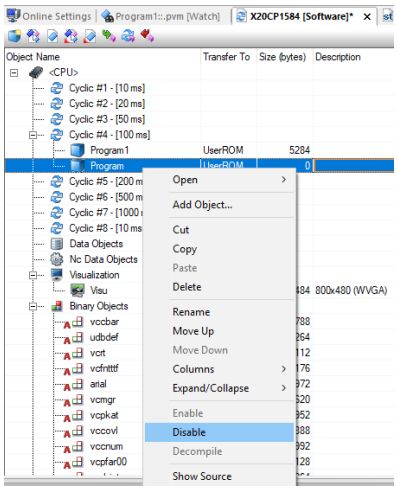

Rysunek 37. Przerzutnik RS (ST)

W przypadku, kiedy w projekcie znajdują się dwa programy, trzeba mieć na uwadze, że sterownik B&R jest systemem czasu rzeczywistego. Programy realizowane są cyklicznie w ustalonych odstępach czasu. Podgląd i zmiana tych ustawień możliwa jest po przejściu do zakładki *Physical View* i naciśnięciu prawym przyciskiem myszy na sterownik wybierając opcję *Software.*



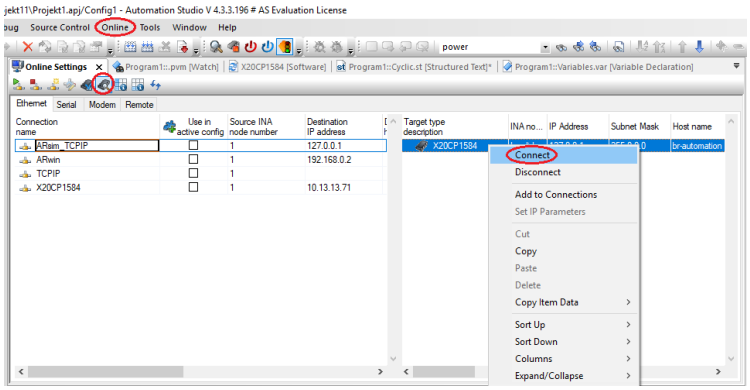

Rysunek 38. Przejście do okna oprogramowania sterownika

Zostanie otwarte okno przedstawiające strukturę wykonywanych programów w sterowniku B&R. Zmianę czasu pomiędzy kolejnymi cyklami wykonywanych programów można uzyskać przeciągając dany program i upuszczając w cyklu z żądanym czasem. W przypadku, gdy mamy dwa programy w projekcie mogą one oddziaływać na siebie powodując błędne zachowanie układu. W Automation Studio jest możliwość wyłączenia danego programu wybierając opcję *Disable*, po zaznaczeniu tej opcji program nie będzie wgrywany do sterownika.



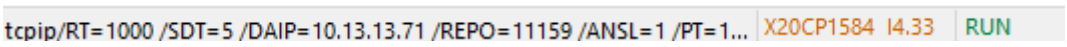

Rysunek 39. Wyłączenie programu w środowisku Automation Studio

Po napisaniu programu należy połączyć się ze sterownikiem. Z paska menu wybieramy *Online* a nastepnie *Settings*. Obszar roboczy zmieni się, odnajdziemy w nim przycisk *Browse* i naciskamy go. Program odszuka podłączony do komputera sterownik i wyświetli go w liście po prawej stronie przestrzeni roboczej. Klikamy odnaleziony sterownik prawym przyciskiem myszy i z menu kontekstowego wybieramy *Connect*.




Rysunek 40. Połączenie ze sterownikiem PLC

Jeśli połączenie zostało nawiązane poprawnie, przyciski dotyczące pracy *online* na pasku zadań zostaną odblokowane, a u dołu okna zobaczysz informacje na temat aktualnego stanu urządzenia.




Rysunek 41. Aktualny stan urządzenia

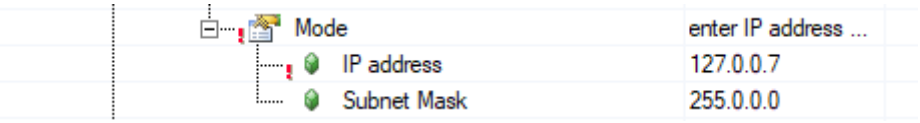

Rysunek 42 Przyciski pracy online

Powyższe przyciski pozwalają na:

1. Monitorowanie pracy urządzenia
2. Zatrzymanie urządzenia, następuje przejście do trybu serwisowego
3. Restart urządzenia do trybu roboczego (Warm Restart)
4. Pracę w symulatorze (kiedy nie ma możliwości transferu projektu do fizycznego sterownika)

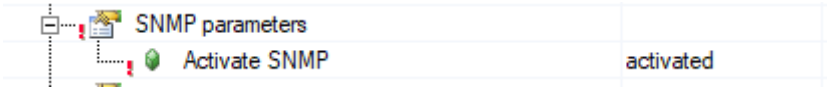
## Uwaga

Przed wgraniem programu do sterownika, trzeba zmienić tryb nadawania mu adresu IP. Aby to zrobić, należy wejść w konfigurację połączenia Ethernet w zakładce "*Physical view*" i zakładkę "*Mode*" ustawić na "*enter IP address manually*", oraz uzupełnić "*IP address*" oraz "*Subnet Mask*" wartościami takimi samymi jakie posiada sterownik z którym jesteśmy połączeni.




Rysunek 43 Ustawienie adresu IP i maski w programie

Oprócz zmiany trybu trzeba również uaktywnić parametr SNMP. Aby to zrobić, tak jak wyżej wchodzimy w konfiguracje połączenia Ethernet i przy „*SNMP parameters*" zaznaczamy „*activated*"



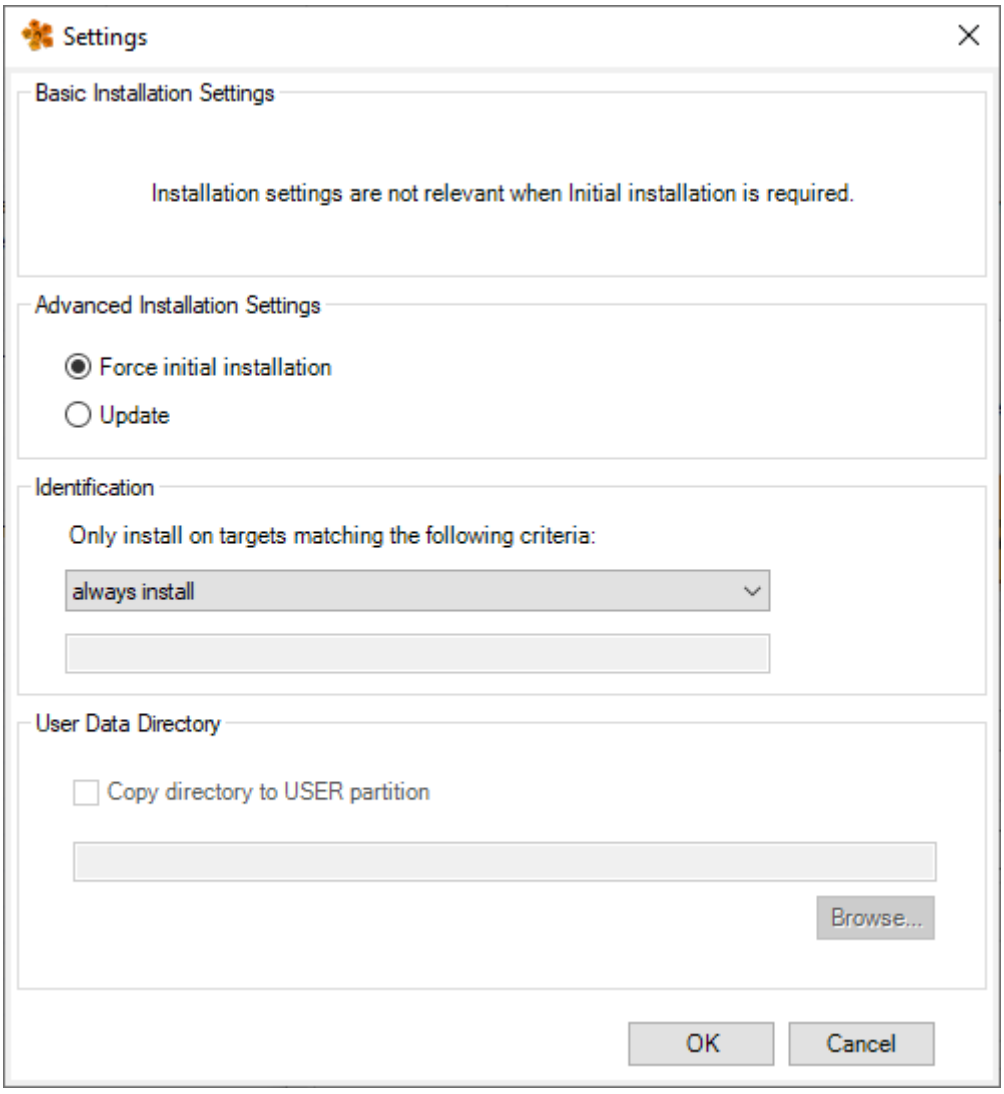

Rysunek 44 Aktywacja parametru SNMP

Jeżeli wgramy program, przed dokonaniem powyższych zmian połączenia Ethernet, nie będziemy w stanie połączyć się z sterownikiem w ten sposób.

## Wgranie programu do sterownika w trybie boot

Jeżeli nie można połączyć się z sterownikiem przez kabel Ethernet, ponieważ został wgrany na sterownik program bez odpowiednich ustawień, trzeba będzie go wyłączyć i zmienić tryb na *boot* oraz wgrać program za pomocą pendrive'a. Aby to zrobić należy napisać program w Automation studio z odpowiednimi ustawieniami połączenia Ethernet i wgrać go na

przenośnik pamięci. Robi się to wchodząc w zakładkę „*Project*” w *toolboxie* i wybierając opcję *Generate Project Installation Package*. Powinno otworzyć się nowe okno, w którym wybieramy opcję *Generate PIP incl. Suport of AR < 4.33 in a custom folder*, podajemy ścieżkę pendrive’a oraz w ustawieniach zaznaczamy opcję *Force initial installation.*

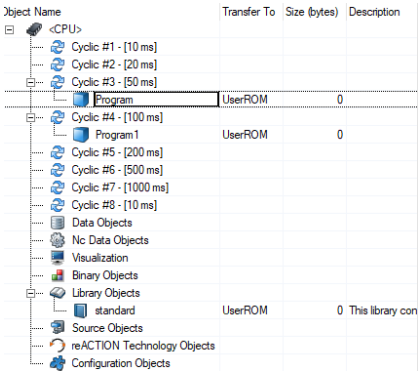



Rysunek 45 Ustawienie instalacji  programu używając pamięci przenośnej

Po nadaniu odpowiednich ustawień naciskamy *Generate Project installation package*. Kiedy program zostanie wgrany na pamięć, należy wyłączyć sterownik, odłączyć od niego kabel Ethernet i za pomocą śrubokrętu zmienić tryb włączania na *boot*. Następnie podłączyć do sterownika pendrive’a i uruchomić sterownik. Program powinien w ciągu chwili się wgrać. Gdy się tak stanie wyłączyć sterownik, ustawić ponownie tryb włączania na *run* i włączyć sterownik, który jest gotowy do połączenia się do komputera za pomocą kabla Ethernet.

## Wgranie programu do sterownika

Jeżeli konfiguracja sprzętu została dodana poprawnie to program podczas utworzenia trafia on automatycznie do bloku cyklicznego CPU, wykonującego się co 100 [ms]. Można również przenieść program i tym samym wpływać na to, co jaki czas będzie wykonywał się cykl danego programu. Aby tego dokonać należy z zakładki *Physical View* wybrać ponownie opcję *Software*.

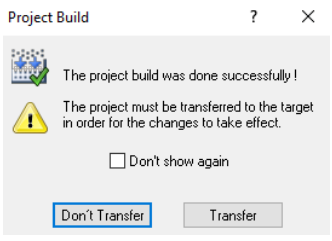



Rysunek 46 Struktura oprogramowania w Automation Studio

Aby przenieść napisany program do aktywnego sterownika użyj przycisku *Transfer* znajdującego się w menu głównym. Przed wgraniem projektu, jest on automatycznie kompilowany.

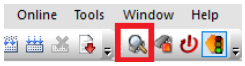

Rysunek 47 Transfer projektu do sterownika

Jeśli dokonujesz kompilacji przed wgraniem programu, w przypadku braku błędów zostaniesz zapytany czy projekt ma być przeniesiony do sterownika PLC.




Rysunek 48 Okno po poprawnie zakończonej kompilacji

Po zakończeniu transferu projektu sterownik wykona restart do trybu RUN. Aby obserwować jak zmieniają się stany wejść i wyjść oraz zmienne z nimi powiązane musisz wejść w tryb monitorowania klikając przycisk *Monitor*.



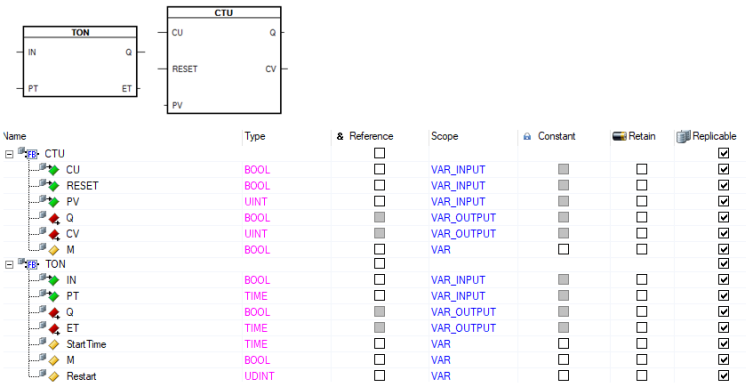

Rysunek 49 Załączenie trybu monitoringu

## Ćwiczenie 4

Napisz programy realizujące przerzutnik z priorytetem na SET w dwóch wersjach, klasycznej oraz z wykorzystaniem bloku funkcyjnego. Wgraj programy do sterownika i przetestuj ich działanie. Uważaj aby testować na sterowniku tylko 1 program, wyłączenie drugiego programu pokazane został w instrukcji powyżej.

# Użycie bloków zegarów i liczników w Automation Studio

Dodając do projektu bibliotekę *standard* zyskujemy dostęp do funkcji umożliwiających zliczanie określonych zdarzeń oraz odmierzanie czasu. Poniżej przedstawione zostały podstawowe bloki realizujące wyżej wymienione funkcje. Wywołanie każdego bloku funkcyjnego powoduje konieczność utworzenia zmiennej będącej jej instancją i przechowującą parametry dotyczące danego bloku.




Rysunek 50 Bloki funkcyjne zegara TON oraz licznika CTU wraz z instancjami

**TON (Timer ON Delay)**- po podaniu sygnału logicznej jedynki na wejście IN licznik zaczyna odmierzać czas. Po zrównaniu się odmierzonej wartości z wartością nastawioną na wejściu PT aktywowane jest wyjście Q. Wyjście to jest aktywne aż do czasu zmiany sygnału na wejściu IN z powrotem na logiczne zero . Na wyjściu ET odczytać można odczytać czas jaki upłynął od rozpoczęcia pracy licznika.

**CTU (Counter Upwards)-** licznik służący do zliczania impulsów. Każda zmiana sygnału na wejściu CU z zera na jedynkę logiczną zwiększa wartość licznika o 1. Po osiągnięciu wartości zadanej na wejściu PV aktywowane jest wyjście Q. Aktywne jest do momentu podania logicznej jedynki na wejście RESET. Powoduje to wyzerowanie licznika. Wyjście CV pozwala na odczytanie aktualnej wartości licznika

Ogólnie w bibliotece mamy do wyboru następujące bloki funkcyjne zegarów i liczników.

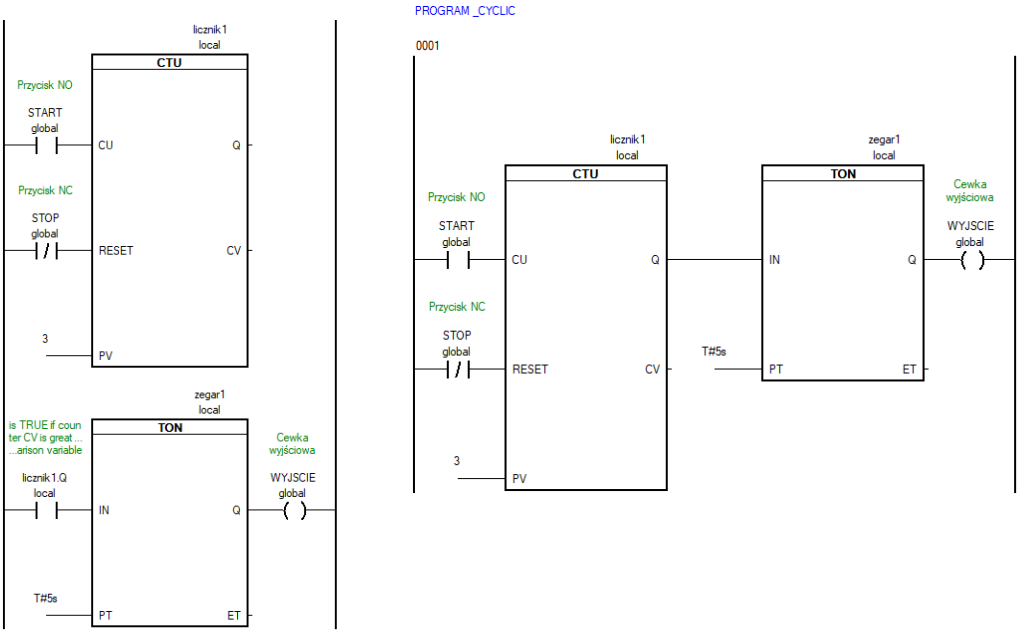
| Blok funkcyjny | Opis |
|---|---|
| CTD | Licznik zliczający w dół |
| CTU | Licznik zliczający w górę |
| CTUD | Licznik rewersyjny zliczający w górę lub w dół |
| TON | Zegar z opóźnionym załączaniem |
| TOF | Zegar z opóźnionym wyłączaniem |
| TP | Generator impulsu o zadanym czasie trwania |

# Ćwiczenie 5

Napisz program, który po trzykrotnym naciśnięciu przycisku START, załączy WYJSCIE z opóźnieniem 5 sekund, układ resetowany jest po naciśnięciu przycisku STOP. Programy należy napisać w językach LD oraz ST. Dodatkowo, układ można uzupełnić o wyłączanie dopiero po dwukrotnym naciśnięciu przycisku STOP.

W tym celu do projektu należy dodać blok funkcyjny *TON* oraz *CTU*. Aby układ realizował podaną funkcjonalność, przykładowy program w języku LD może wyglądać następująco.

Możliwe jest również wykorzystanie wyjścia bloków funkcyjnych w innym miejscu programu, za pomocą operatora kropki „.”. W ten sposób możemy odwoływać się do zmiennych bloku funkcyjnego z różnych miejsc programu.




Rysunek 48 Równoważne kody programu do ćwiczenia 5 w języku LD

Realizacja tego samego programu w języku ST będzie miała następującą postać.

```
PROGRAM _CYCLIC
    licznik1(CU := START , RESET := NOT STOP , PV := 3 );
    zegar1(IN := licznik1.Q , PT := T#5s );
    WYJSCIE := zegar1.Q;
END_PROGRAM
```

Rysunek 49 Kod programu do ćwiczenia 5 w języku ST