

# Instrukcja uruchomienia SV660N w sterowniku PLC serii Easy

Instrukcja przedstawia jak krok po kroku uruchomić w sterowniku PLC serii Easy sterowanie serwonapędem SV660N poprzez komunikację EtherCAT.

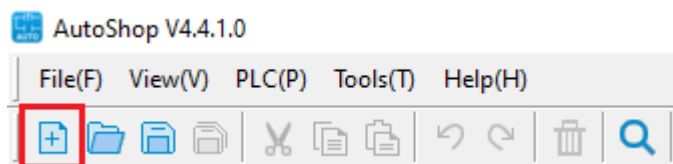
Potrzebne oprogramowanie: AutoShop

## Spis Treści

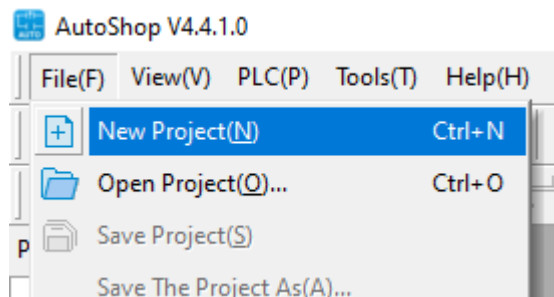
1. Utworzenie nowego projektu
2. Dodanie serwo napędu do komunikacji EtherCAT
3. Dodanie osi sterowania ruchem
4. Test komunikacji sterownika PLC Easy z serwonapędem SV660N.
5. Tworzenie programu do sterowania serwonapędem.
6. Konfiguracja osi.

### 1. Utworzenie nowego projektu

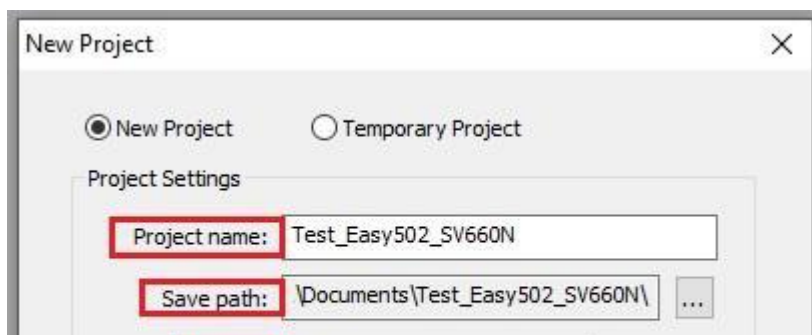
Po instalacji oraz uruchomieniu oprogramowania AutoShop, należy utworzyć nowy projekt. W tym celu należy kliknąć w ikonę tworzenia nowego projektu:



lub korzystając z zakładki wybrać „**File**” → „**New Project**”:



W oknie, które się pojawi należy uzupełnić i wybrać następujące pola:



Pola wymagane:

**Project name** – nazwa projektu

**Save path** – miejsce zapisu projektu na dysku

**Editor** – język programowania *Ladder Chart / Sequential Function Chart / Lite ST*

**Series and models** – seria i model sterownika

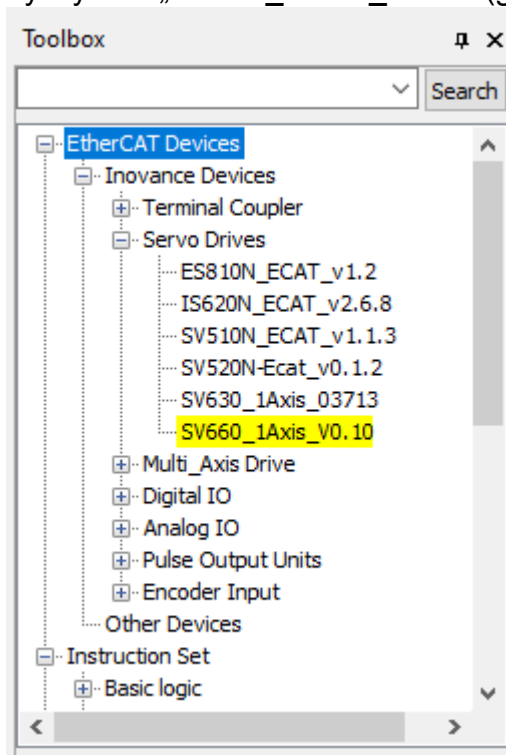
Pole opcjonalne:

**Project description** – krótki opis projektu

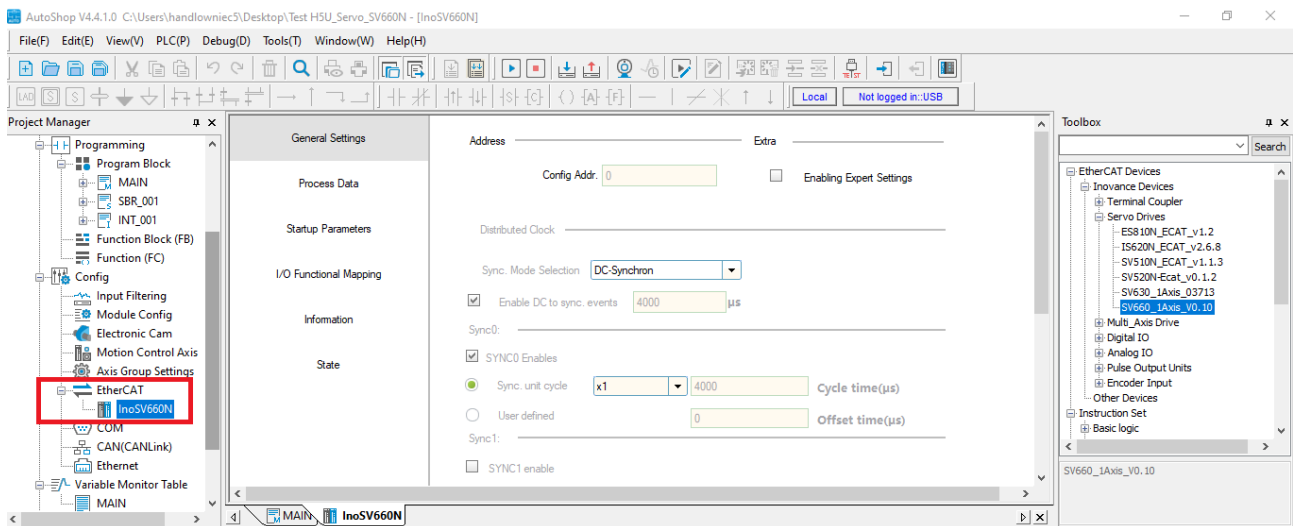
Po wypełnieniu obowiązkowych pól, należy zatwierdzić przyciskiem „**OK**” utworzenie nowego projektu.

## 2. Dodanie serwo napędu do komunikacji EtherCAT

W celu komunikacji sterownika Easy z serwonapędem SV660N należy w oprogramowaniu AutoShop dodać odpowiednie urządzenie do komunikacji EtherCAT. W tym celu z drzewka po prawej stronie należy wyszukać i wybrać odpowiednie urządzenie EtherCAT. Dla serwonapędu SV660N należy wybrać „**SV660\_1Axis\_Vx.xx**” (gdzie: x.xx to wersja pliku):



Po dwukrotnym wciśnięciu lewym przyciskiem myszy (LPM) na wybrane urządzenie, zostanie ono dodane do projektu.



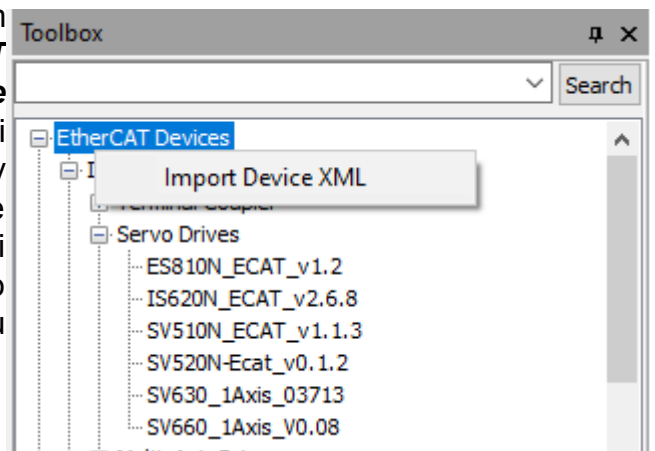
Automatycznie otworzy się w środkowej części ekranu widok konfiguracji urządzenia.

## UWAGA!

Oprogramowanie może mieć zaimplementowaną starszą wersję pliku urządzenia. Aktualne pliki konfiguracyjne dla serwo napędu znajdują się w zakładce „Do pobrania” produktu.

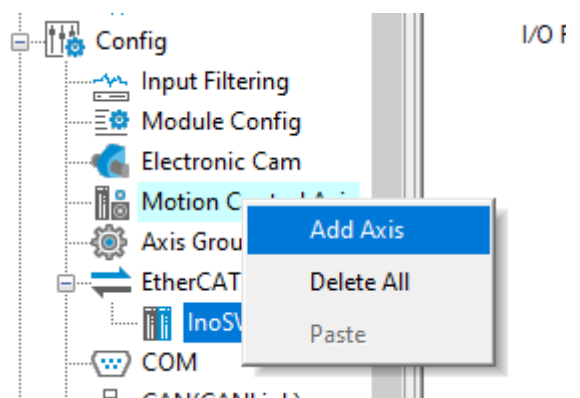
Jeśli dostępna do pobrania wersja pliku konfiguracyjnego jest nowsza (wartość po literze „V”), należy ją pobrać i zaktualizować urządzenie w AutoShop.

Aby zaktualizować urządzenie, należy prawym przyciskiem myszy (PPM) kliknąć w „**EtherCAT Devices**” a następnie wybrać „**Import Device XML**”. W kolejnym kroku należy znaleźć i wybrać wcześniej pobrany plik konfiguracyjny (rozszerzenie pliku to \*.xml). Oprogramowanie poinformuje o udanej aktualizacji pliku i poprosi o ponowne uruchomienie programu. Po ponownym uruchomieniu nowa wersja pliku konfiguracyjnego będzie już dostępna.



### 3. Dodanie osi sterowania ruchem

Gdy serwo napęd jest już dodany do projektu, kolejnym krokiem jest dodanie nowej osi, która będzie sterowana. W drzewku projektu należy prawym przyciskiem myszy kliknąć na „**Motion Control Axes**” i wybrać „**Add Axis**”.

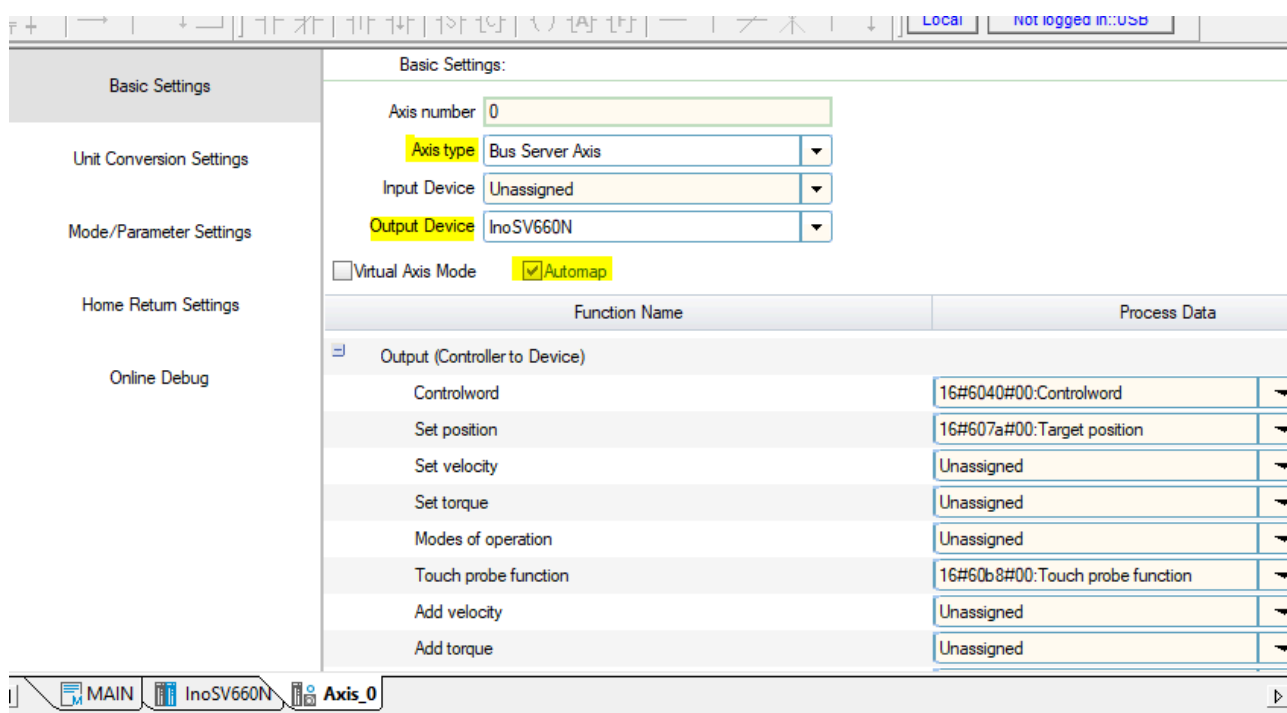


W drzewku pojawi się nowa oś:



Dwukrotne wciśnięcie lewego przycisku myszy na utworzonej osi, otworzy w polu głównym ustawienia. Należy ustawić następujące parametry w „**Basic Settings**”:

- „**Axis Type**” = „**Bus Server Axes**”
- „**Output Device**” wybrać „**InoSV660N**”
- Zaznaczyć opcję „**Auto Map**”



#### 4. Test komunikacji sterownika PLC Easy z serwonapędem SV660N.

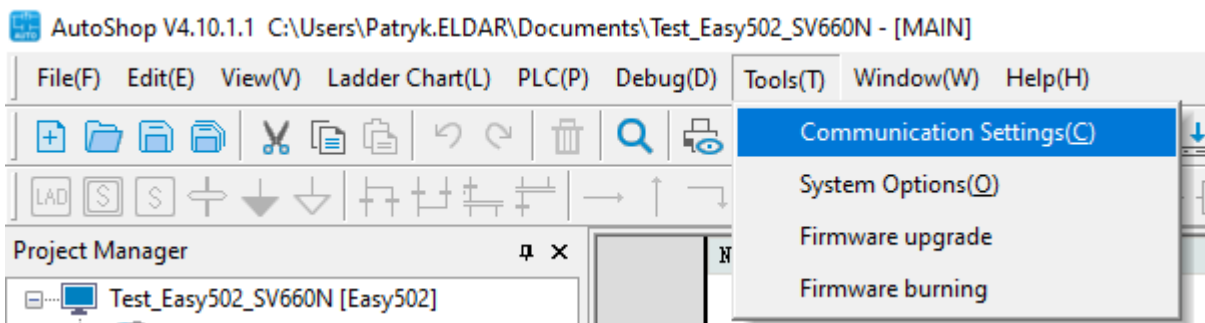
Po utworzeniu osi i przypisaniu do niej odpowiedniego urządzenia, można sprawdzić, czy sterownik PLC oraz serwo napęd komunikują się między sobą poprzez protokół EtherCAT.

Oba urządzenia należy zasilić i przewodem komunikacyjnym połączyć między sobą. W programie AutoShop należy skompilować stworzony projekt klikając w ikonkę „**Compile All**” lub wcisnąć klawisz **F7**.

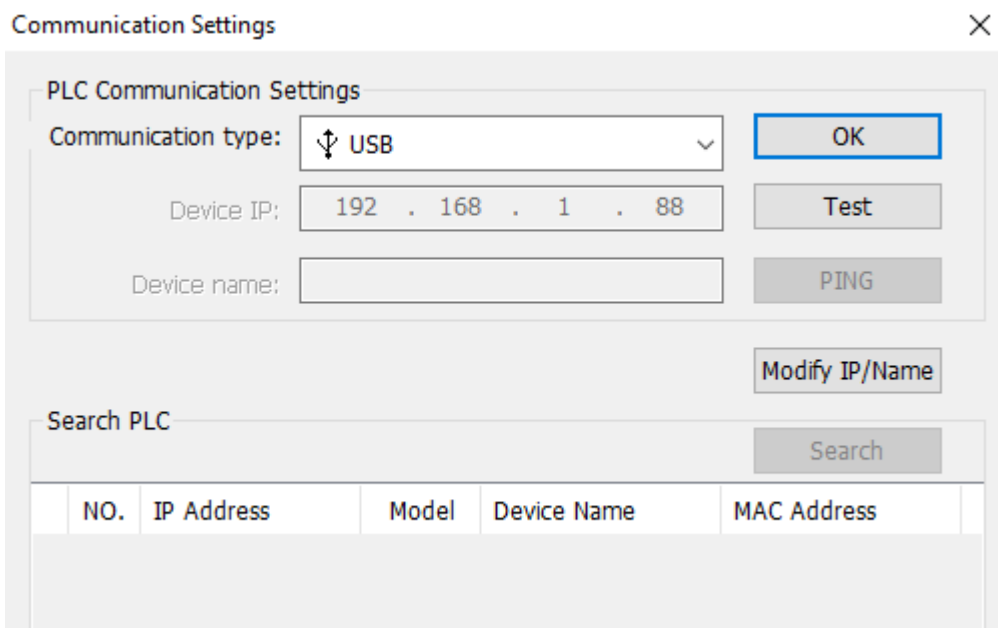


Następnie przy pomocy przewodu USB podłączyć sterownik PLC do PC.

W programie należy zmienić komunikację ze sterownikiem na USB. W tym celu z zakładki „**Tools**” wybrać „**Communication Settins**”



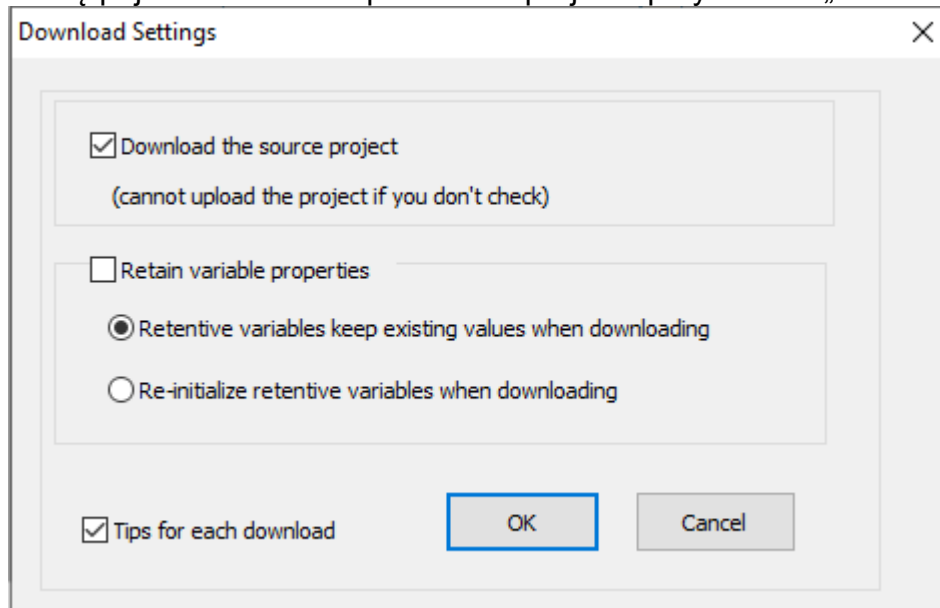
W oknie które się pojawiło w polu „**Communication type**” wybrać komunikację **USB**.



Po podłączeniu przewodu i ustawieniu komunikacji następnie należy kliknąć w ikonę „**Download**” lub wcisnąć klawisz **F8** w celu pobrania projektu i ustawień do sterownika PLC.



W oknie, które się pojawi zatwierdzić pobieranie projektu przyciskiem „**OK**”:



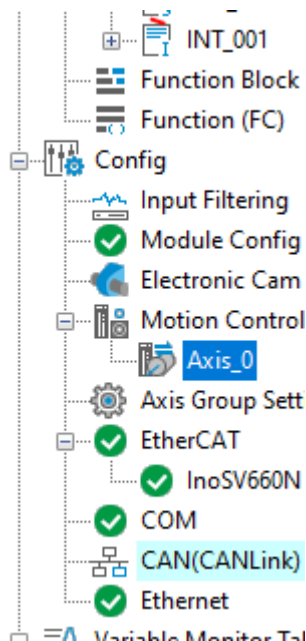
Po wgraniu projektu, należy uruchomić aplikację klikając w ikonę „**Run**”, wciskając klawisz **F5**, lub z poziomu sterownika PLC zmienić położenie przełącznika z „**STOP**” na „**RUN**”.



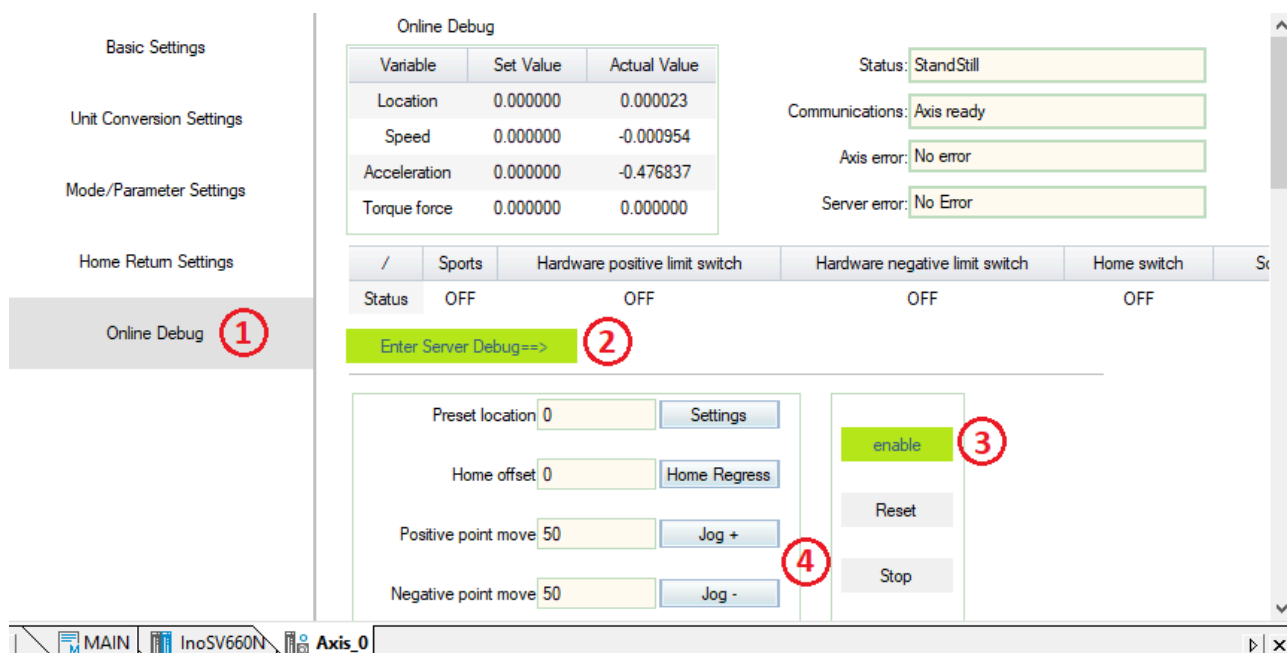
W tym momencie aplikacja wgrana do sterownika działa. Następnie należy uruchomić w programie monitorowanie klikając w ikonę „**Monitoring**” lub wciskając klawisz **F3**.



Jeśli nie wystąpiły, żadne błędy ani problemy to w drzewku projektu, przy niektórych pozycjach pojawią się zielone symbole oznaczające, że wszystko działa prawidłowo:



W ustawieniach osi *Axis\_0* należy przejść do zakładki „**Online Debug**”, następnie uruchomić debugowanie klikając w „**Enter Server Debug ==>**”. Po kliknięciu w przycisk „**enable**” serwo silnik zostanie załączony. Następnie przy pomocy przycisków „**JOG+**” oraz „**JOG-**” można wał serwo silnika wprowadzić w ruch. W czasie ruchu tabelka u góry, będzie wskazywała aktualny odczyt parametrów.

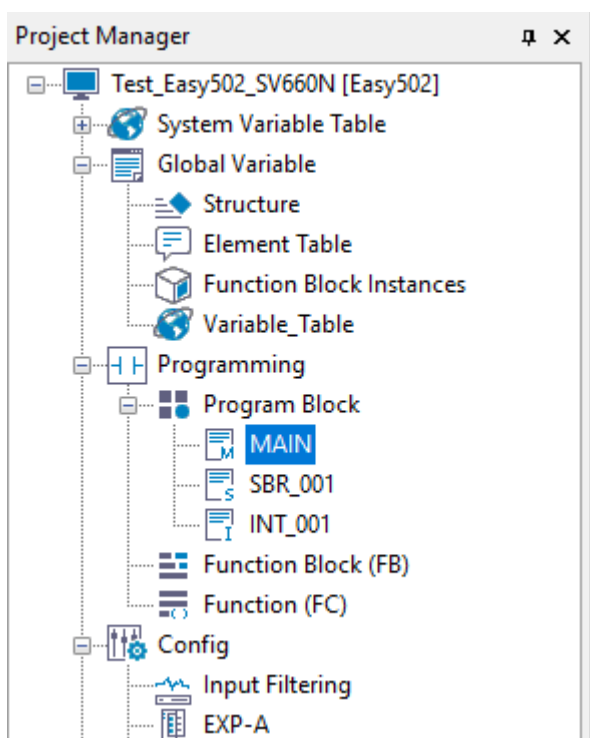


Jeśli tabelka na bieżąco aktualizuje wartości i wał serwo silnika obraca się zgodnie z poleceniami **JOG+** / **JOG-**, oznacza to, że sterownik Easy komunikuje się prawidłowo z serwonapędem SV660N. Można wyłączyć debugowanie i wprowadzić sterownik w tryb **STOP**. W chwili zatrzymania pracy sterownika PLC, na wyświetlaczu serwo wzmacniacza pojawi się błąd **EE08.2**. Jest to normalne i błąd ten informuje o utracie komunikacji ze sterownikiem. Zniknie po ponownym wprowadzeniu sterownika Easy w tryb pracy **RUN**.

## 5. Tworzenie programu do sterowania serwonapędem.

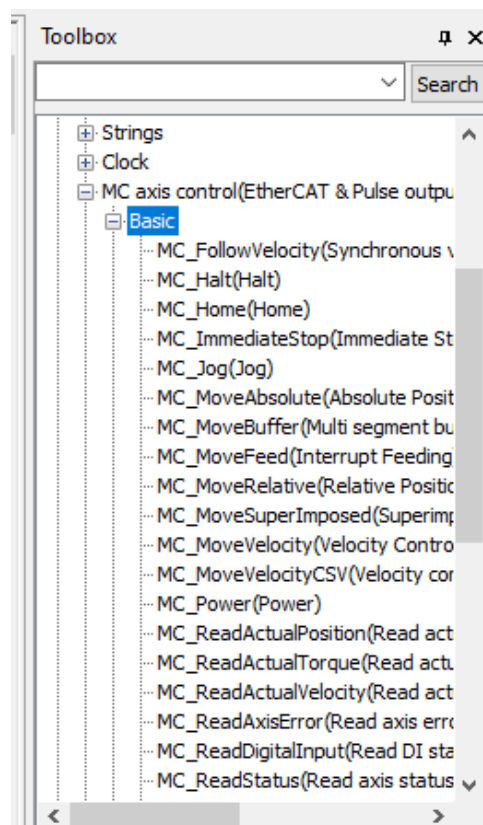
Po upewnieniu się, że sterownik Easy komunikuje się z serwonapędem SV660N po EtherCAT, można przejść do „napisania” prostego programu, który pozwoli na sterowanie serwo silnikiem.

W tym celu z drzewka projektu należy wybrać i otworzyć „**MAIN**”:



W oknie głównym otworzy się okno tworzenia programu w języku drabinkowym.

Do sterowania serwonapędem po komunikacji EtherCAT są przygotowane specjalne bloczki programowe, które to umożliwiają i znajdują się one w „**Toolbox**” w zakładce „**MC Axes control (EtherCAT & Pulse output)**”.



Bloczek podstawowej kontroli ruchu

MC\_Power

MC\_Reset

MC\_ReadStatus

MC\_ReadAxisError

MC\_ReadDigitalInput

MC\_ReadActualPosition

MC\_ReadActualVelocity

Opis funkcji

Włączenie kontroli osi

Reset błędu osi

Odczyt statusu osi

Odczyt błędu osi

Odczyt stanu wejść cyfrowych urządzenia

Odczyt aktualnej pozycji

Odczyt aktualnej prędkości

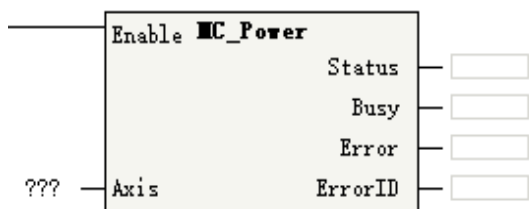


|                      |  |
|----------------------|--|
| MC_ReadActualTorque  | Odczyt aktualnego momentu  |
| MC_SetPosition       | Ustawienie pozycji osi   |
| MC_TouchProbe        | Zatrzaśnięcie pozycji  |
| MC_MoveRelative      | Pozycja względna osi   |
| MC_MoveAbsolute      | Pozycja absolutna osi  |
| MC_MoveVelocity      | Komenda prędkości osi  |
| MC_Jog               | Komenda JOG osi  |
| MC_TorqueControl     | Kontrola momentu osi   |
| MC_Home              | Bazowanie osi  |
| MC_Stop              | Stop osi   |
| MC_Halt              | Pauza osi  |
| MC_ImmediateStop     | Stop bezpieczeństwa osi  |
| MC_MoveFeed          | Pozycja przerwania osi   |
| MC_MoveBuffer        | Komenda multi pozycji osi  |
| MC_MoveSuperImposed  | Komenda nałożenia pozycji na oś  |
| MC_MoveVelocityCSV   | Kontrola prędkości CSV (z funkcją PWM)                                   |
| MC_SyncMoveVelocity  | Sterowanie prędkością CSV (z funkcją PWM, bez przyspieszania/zwalniania) |
| MC_SyncTorqueControl | Synchronizacja kontroli momentu obrotowego                               |

Objaśnienie najważniejszych blozków do sterowania serwonapędem:

### MC\_Power:

Komenda **MC\_Power** służy do przełączania stanu osi z wyłączonej na włączoną (aktywną), w stanie aktywnym oś jest gotowa do ruchu.



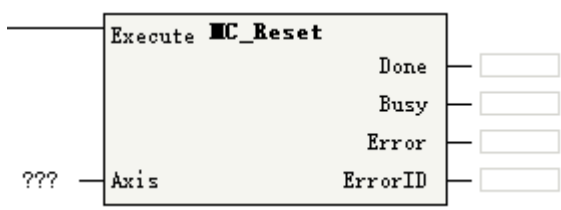
- *Enable* – sygnał załączenia
- *Axis* – numer osi (INT) lub nazwa osi(name(\_sMCAXIS\_INFO))
- *Status* – status, gdy =ON to oś jest załączona
- *Busy* – informuje o tym czy bloček jest zajęty wykonywaniem funkcji, czy nie
- *Error* – błąd

- *ErrorID* – ID błędu

### MC\_Reset:

Po wykonaniu **MC\_Reset.Done** oś przechodzi w stan wyłączenia lub zatrzymania.

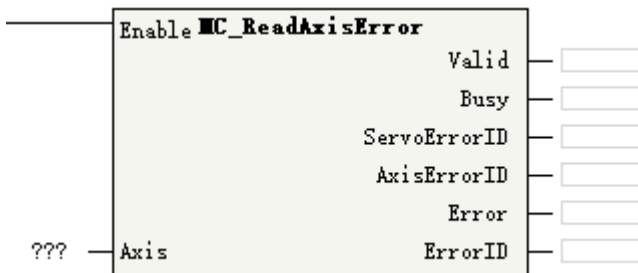
Należy zauważyć, że jeśli **MC\_Reset** jest używane, gdy nie ma błędu osi, ten blok funkcyjny będzie miał błąd.



- *Execute* – sygnał wykonania komendy
- *Done* – informuje o wykonaniu polecenia

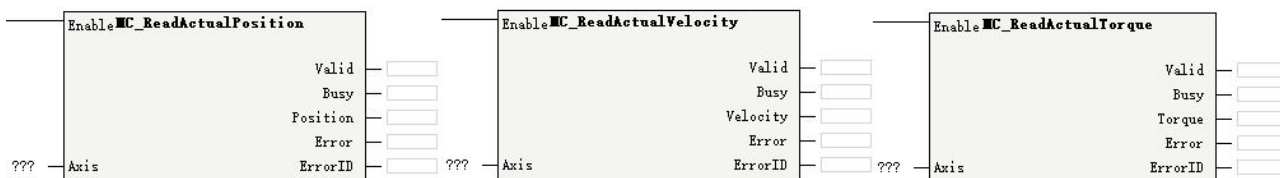
## MC\_ReadAxisError

Gdy *Enable*=ON, jeśli oś istniała i nie ma błędu konfiguracji, *Valid* będzie ON. *AxisErrorID* służy do wyświetlania kodu błędu, użytkownik może znaleźć informacje o błędzie zgodnie z kodem błędu w dokumentacji.



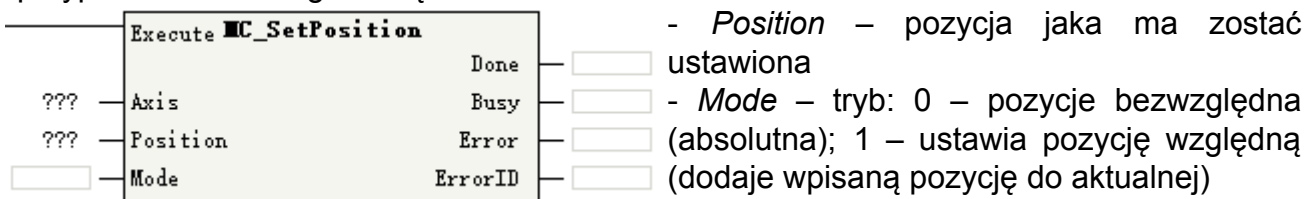
## MC\_ReadActualPosition, MC\_ReadActualVelocity, MC\_ReadActualTorque

Bloki te są używane do odczytu pozycji / prędkości / momentu obrotowego osi w czasie rzeczywistym. Gdy wejście *Enable* zostanie aktywowane, wówczas bloczki monitorują określoną wartość.



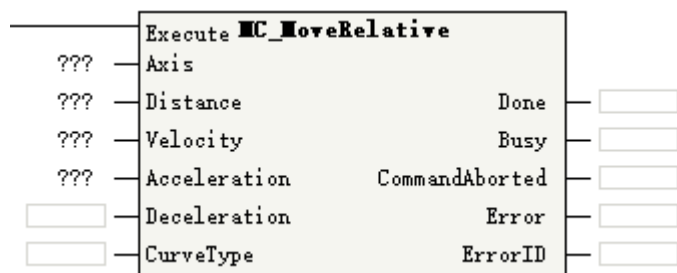
## MC\_SetPosition

Bloczek ten może zmodyfikować pozycję osi bez ruchu. Należy zauważyć, że tylko w stanie gotowości/wyłączenia/błędu zatrzymania oś może wykonać to polecenie, w innym przypadku bloczek zgłosi błąd.



## MC\_MoveRelative

Bloczek ten pozwala na wykonanie ruchu osi o zadany dystans z ustaloną prędkością oraz przyspieszaniem/hamowaniem.



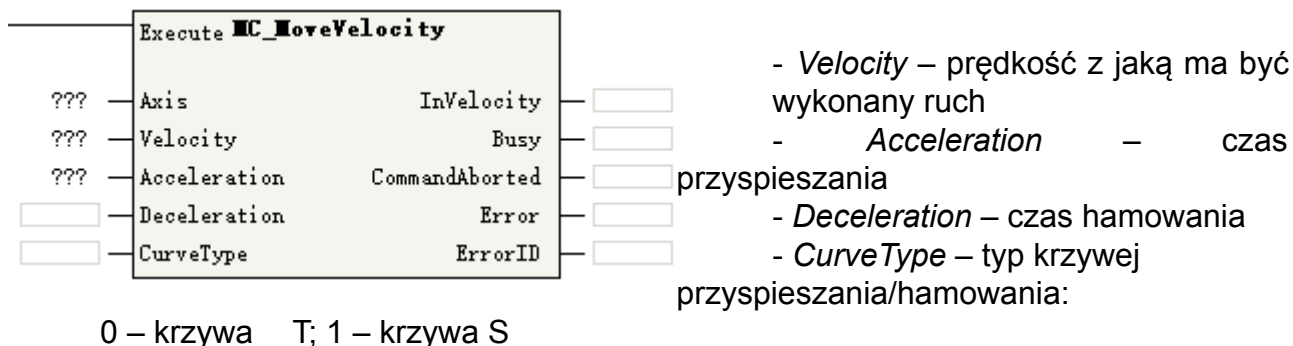
- *Distance* – dystans jaki ma być wykonany przez oś
- *Velocity* – prędkość z jaką ma być wykonany ruch
- *Acceleration* – czas przyspieszania
- *Deceleration* – czas hamowania
- *CurveType* – typ krzywej

przyspieszania/hamowania:

0 – krzywa T; 1 – krzywa S

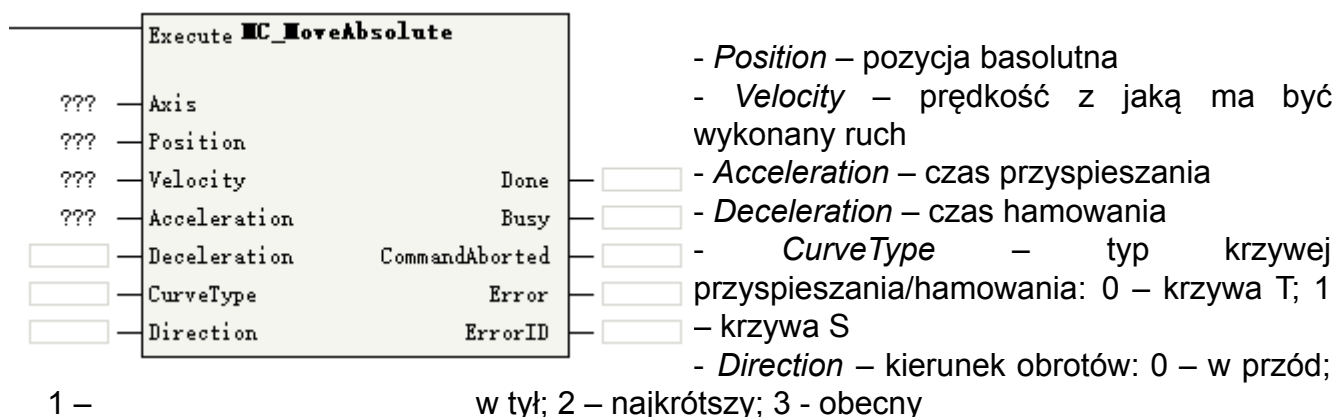
### MC\_MoveVelocity

Bloczek ten powoduje, że serwo silnik pracuje z zadaną prędkością, gdy *Execute* = ON. Ruch trwa do momentu kiedy zmieni się stan na *Execute* = OFF.



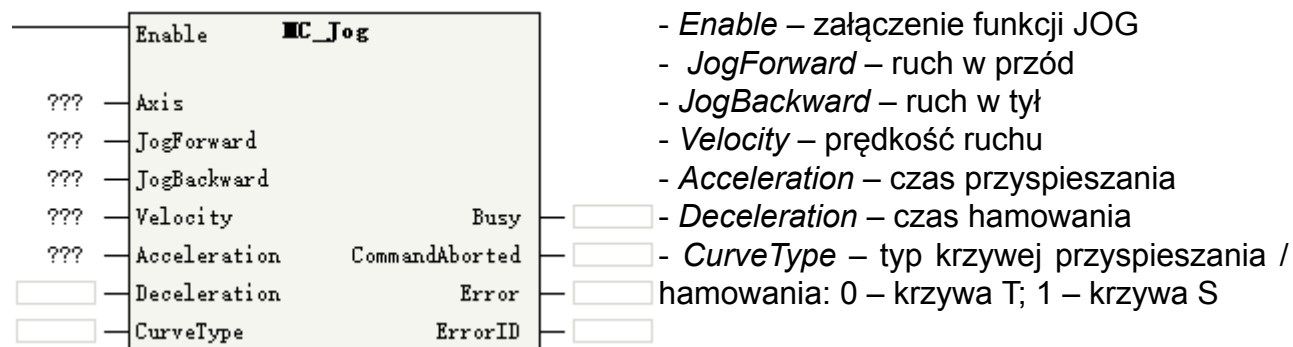
### MC\_MoveAbsolute

Bloczek ten pozwala na wykonanie ruchu osi do zadanej pozycji absolutnej z ustaloną prędkością oraz przyspieszeniem/hamowaniem.



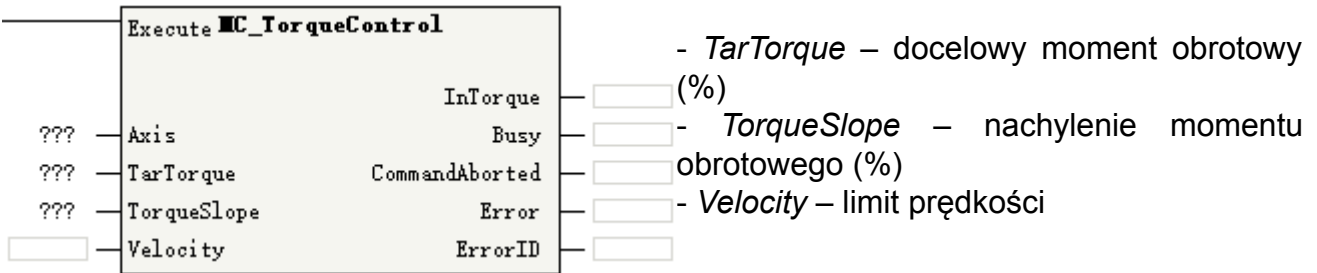
### MC\_Jog

Bloczek funkcyjny, który pozwala na sterowanie ręczne osią. W chwili wystereowania wejścia *JogForward* lub *JogBackward* (nie mogą być równocześnie wystereowane) oś wykona ruch z zadaną prędkością i przyspieszeniem/hamowaniem zgodnie z wybranym kierunkiem i będzie trwać do momentu zwolnienia sygnału wejściowego.



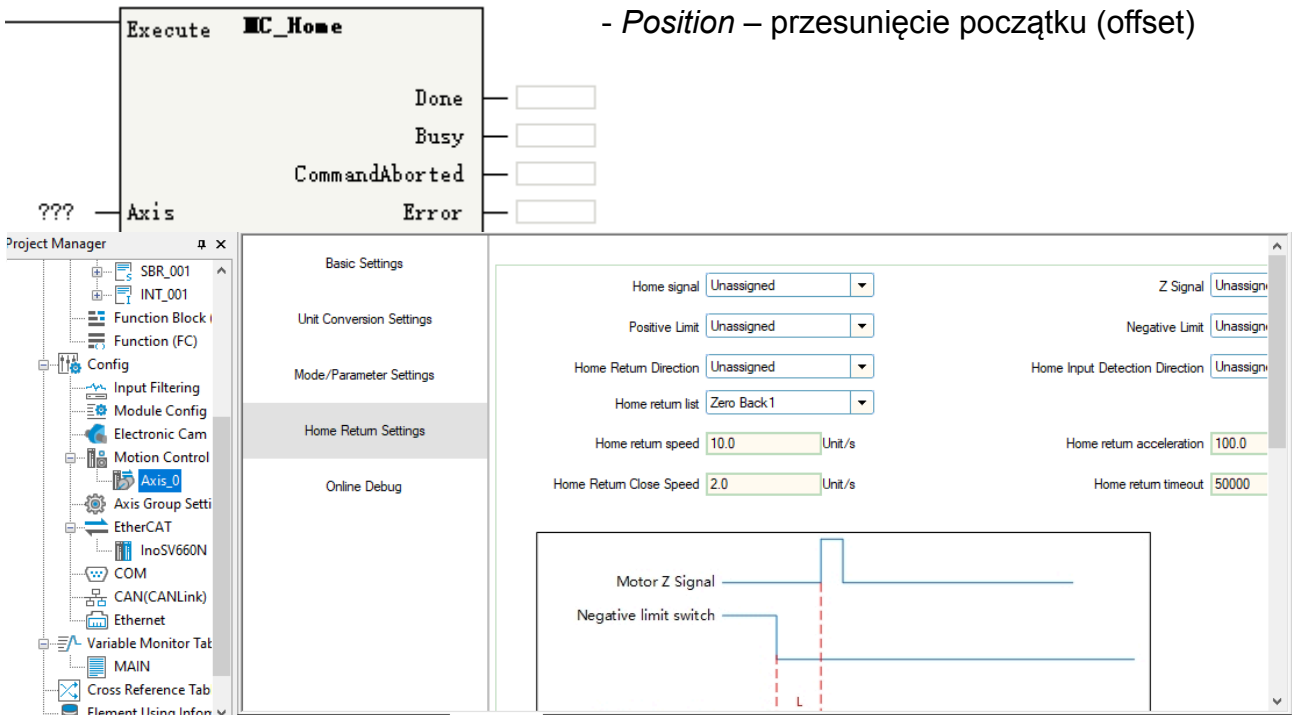
## MC\_TorqueControl

Bloczek pozwalający na ruch osi z ustalonym momentem z zadaną prędkością.



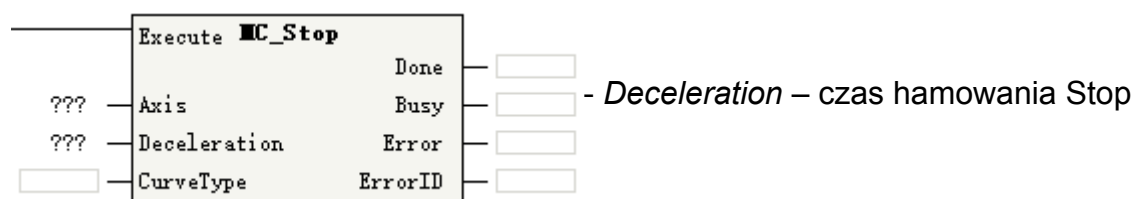
## MC\_Home

Bloczek funkcyjny uruchamiający procedurę bazowania (Homing). Parametry bazowania ustawiane są w ustawieniach konfiguracji osi (zakładka *Home Return Setting*)



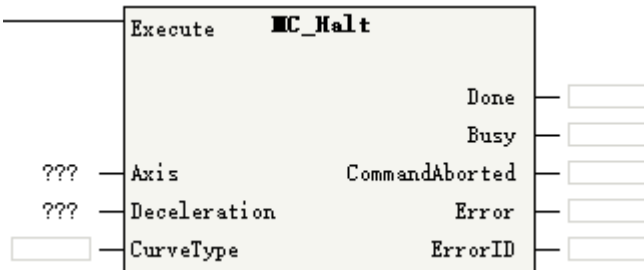
## MC\_Stop

Po wykonaniu funkcji *MC\_Stop* oś się zatrzymuje i przechodzi w tryb STOP. Gdy *Execute* = OFF, wówczas oś przechodzi w stan gotowości. Gdy STOP jest aktywny, wówczas ruch osi jest niemożliwy.



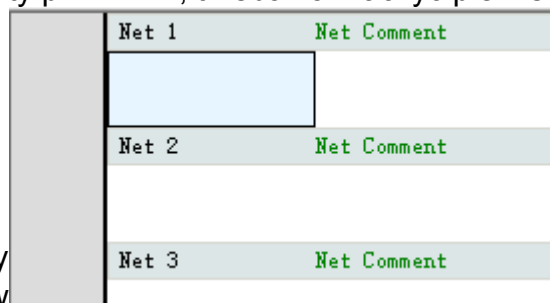
## MC\_Halt

Bloczek *MC\_Halt* wykorzystywany jest do wstrzymania ruchu osi. W chwili wystereowania *Execute* = ON oś się zatrzymuje w deklarowanym czasem hamowania. Zatrzymanie zadziała tylko raz, ciągłe wystereowanie bloczku *Halt* pozwala na ruch osi.

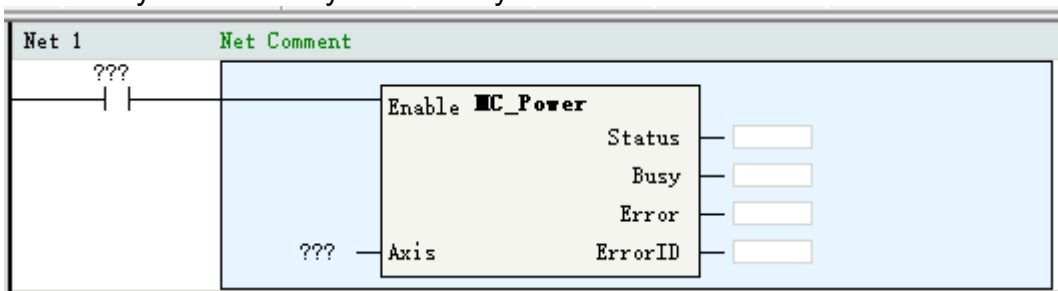


Znając podstawowe bloczki funkcyjne przeznaczone do sterowania osią po EtherCAT, kolejnym krokiem jest stworzenie prostego programu sterującego osią.

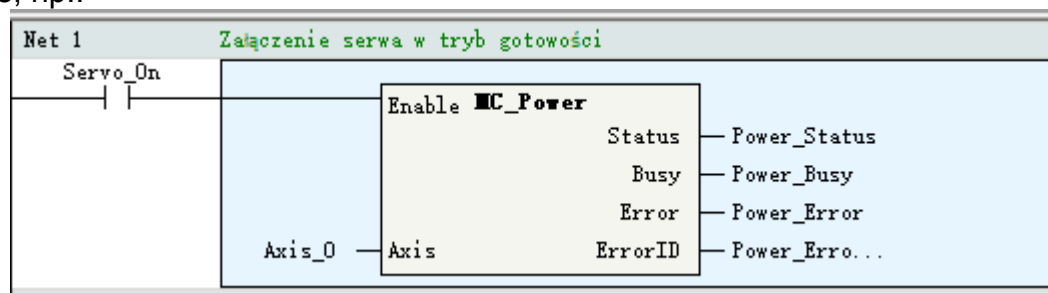
Pierwszym bloczkiem jaki zostanie wykorzystany jest *MC\_Power*, który aktywuje serwonapęd. Mając otwarty plik *MAIN*, trzeba zaznaczyć pierwsze pole jak na rysunku:



Następnie z listy dwukrotnie kliknąć w „*Toolbox*” wyszukać i „*MC\_Power*”. Bloczek zostanie automatycznie dodany wraz ze stykiem:

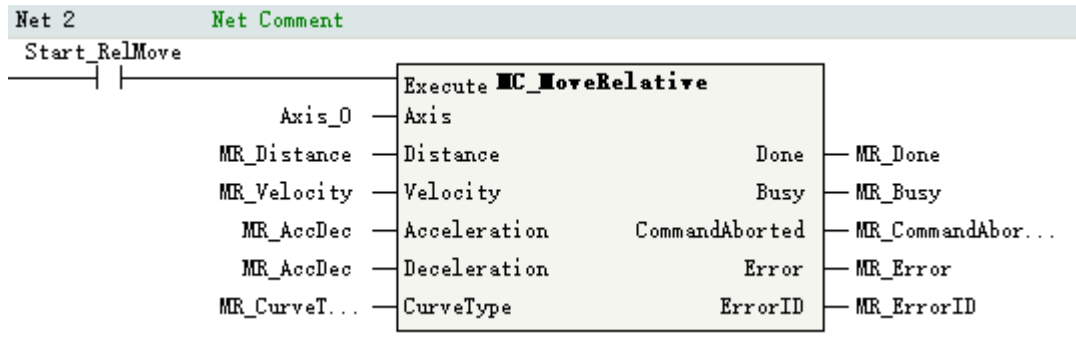


Aby łatwiej było się odnaleźć w projekcie, należy odpowiednio nazwać poszczególne zmienne, np.:



Uwaga! W polu „Axis” bloczku „MC\_Power” należy wprowadzić nazwę osi, do której bloczek ma się odwoływać.

W „Net 2” analogicznie dodać bloczek „MC\_MoveRelative”, który pozwoli na wykonanie ruchu osi do zadanej pozycji:

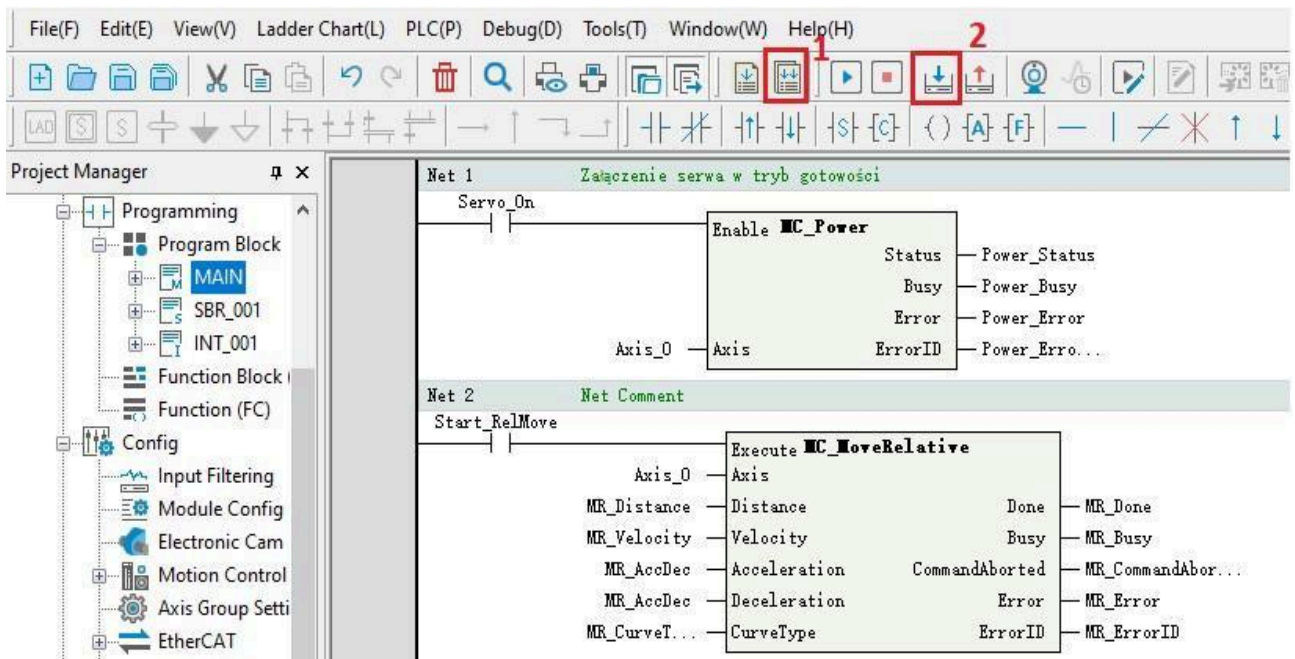


Dobłą praktyką jest przypisanie wartości początkowych dla poszczególnych zmiennych. Aby to zrobić, należy w drzewku projektu wyszukać i otworzyć „Variable\_Table” i ustawić wartości w tabelce „Initial Value”:

- dystans: 1000
- prędkość: 200
- czas przyspieszania / hamowania: 1000

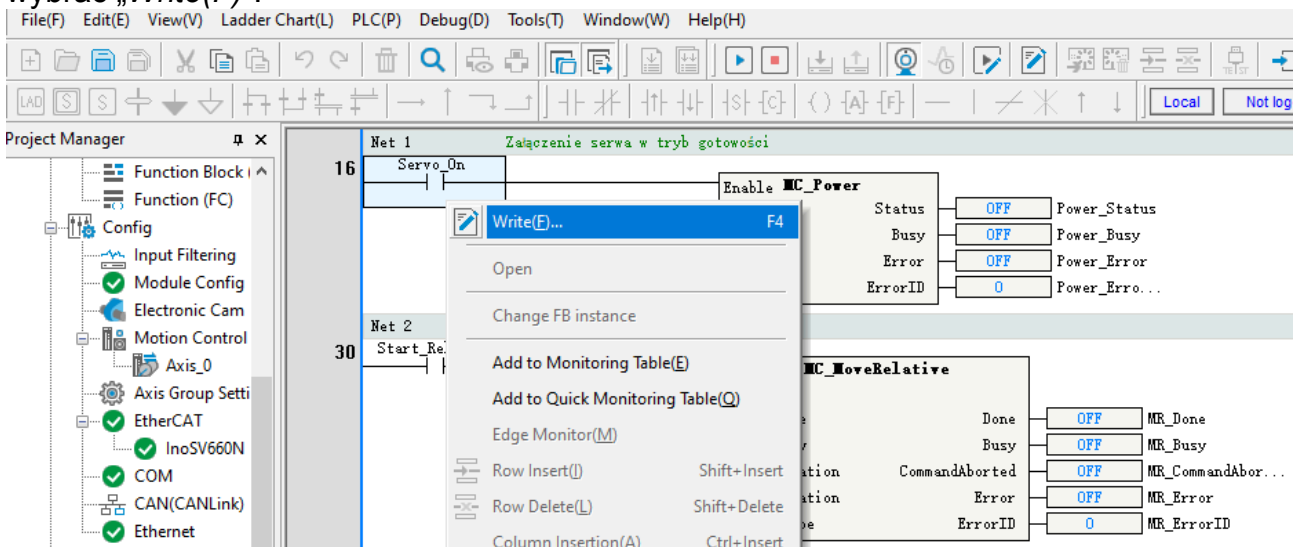
| NO. | Variable Name     | Data Type | Initial Value | Power Down Hold |
|-----|-------------------|-----------|---------------|-----------------|
| 1   | Servo_On          | BOOL      | OFF           | Non Retained    |
| 2   | Power_Status      | BOOL      | OFF           | Non Retained    |
| 3   | Power_Busy        | BOOL      | OFF           | Non Retained    |
| 4   | Power_Error       | BOOL      | OFF           | Non Retained    |
| 5   | Power_ErrorID     | INT       | 0             | Non Retained    |
| 6   | Start_RelMove     | BOOL      | OFF           | Non Retained    |
| 7   | MR_Distance       | REAL      | 1000.000      | Non Retained    |
| 8   | MR_Velocity       | REAL      | 200.0000      | Non Retained    |
| 9   | MR_AccDec         | REAL      | 1000.000      | Non Retained    |
| 10  | MR_CurveType      | INT       | 0             | Non Retained    |
| 11  | MR_Done           | BOOL      | OFF           | Non Retained    |
| 12  | MR_Busy           | BOOL      | OFF           | Non Retained    |
| 13  | MR_CommandAborted | BOOL      | OFF           | Non Retained    |
| 14  | MR_Error          | BOOL      | OFF           | Non Retained    |
| 15  | MR_ErrorID        | INT       | 0             | Non Retained    |

Mając tak przygotowany program, można go przetestować. Należy go skompilować, a następnie pobrać do sterownika PLC.

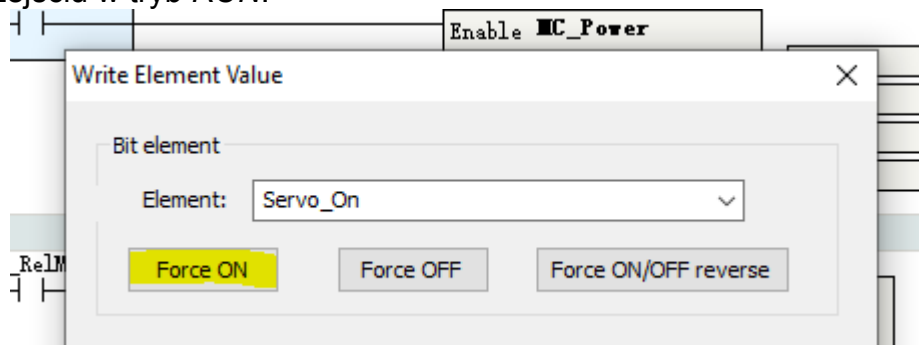


Po wgraniu programu, przełączyć tryb pracy sterownika w **RUN** i włączyć monitoring.

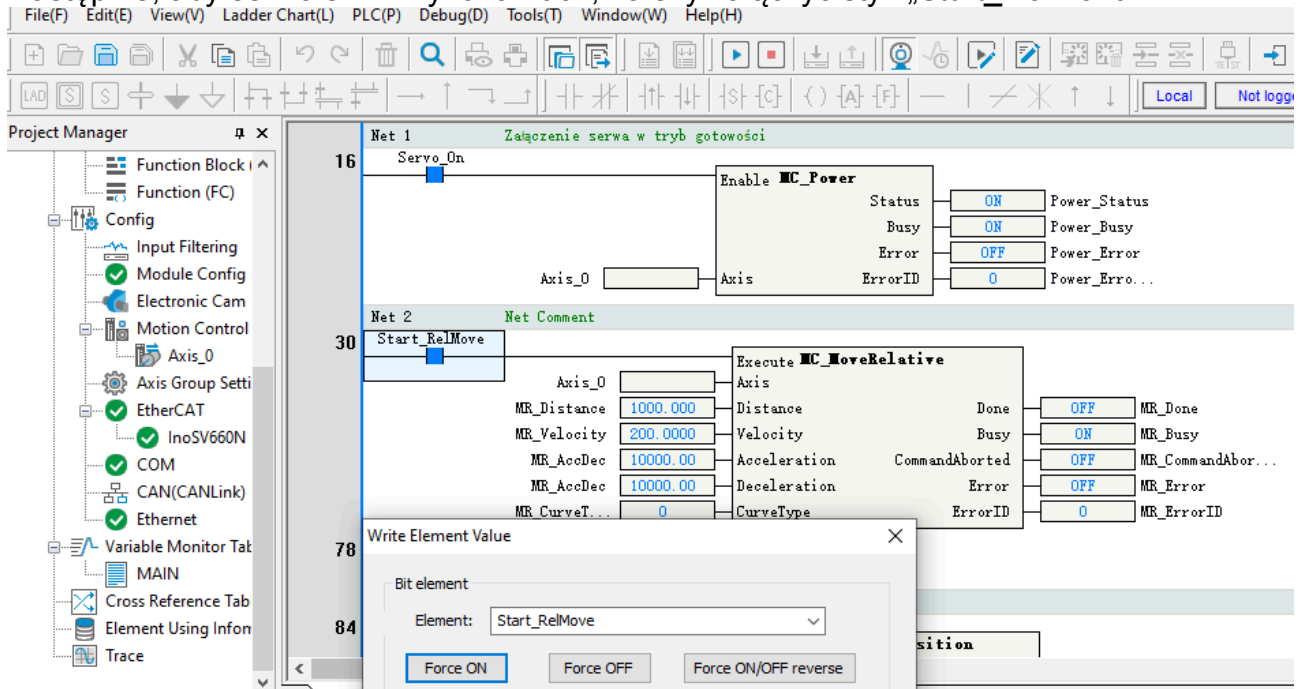
Na schemacie drabinkowym kliknąć prawym przyciskiem myszy w styk „Servo\_On” i wybrać „Write(F)”:



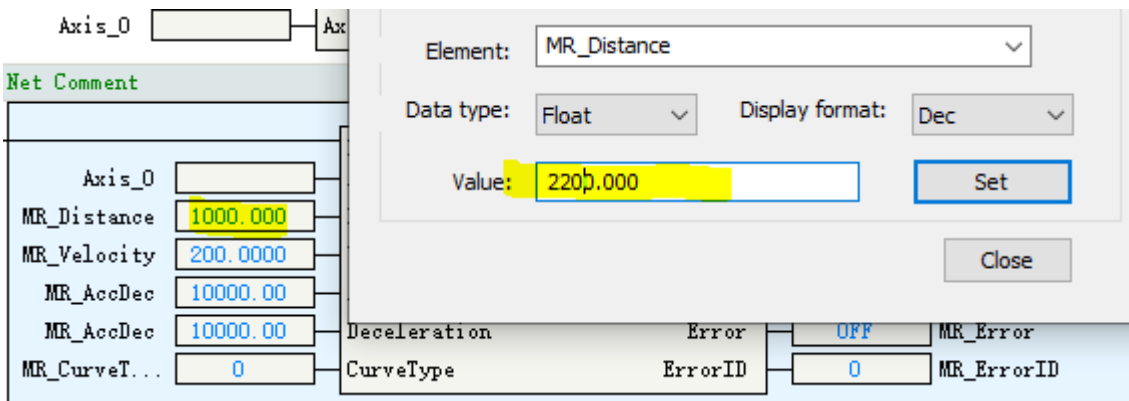
W okienku, które się pojawi wciśnięcie przycisku „Force ON”, spowoduje aktywację serwo napędu i przejściu w tryb **RUN**:



Następnie, aby serwo silnik wykonał ruch, należy załączyć styk „Start\_RelMove”:



Serwo silnik powinien wykonać ruch zgodnie z ustawieniami. Jak widać, załączone styki zmieniają kolor na niebieski. Przycisk „Force OFF” powoduje rozwarczenie styku. Ponowny ruch wymaga rozwarczenia i ponownego zwarcia styku „Start\_RelMove”. Można również zmienić wartości dystansu, prędkości, czy czasów przyspieszania i hamowania, wystarczy prawym przyciskiem myszy kliknąć w odpowiednie pole wybrać opcję „Write(F)” i edytować wartość, zatwierdzając przyciskiem „Set”:



W celu sprawdzenia, można wykonać ponownie ruch, zwiernając styk „Start\_RelMove”. Serwo silnik powinien w tym przypadku pokonać dłuższy dystans.

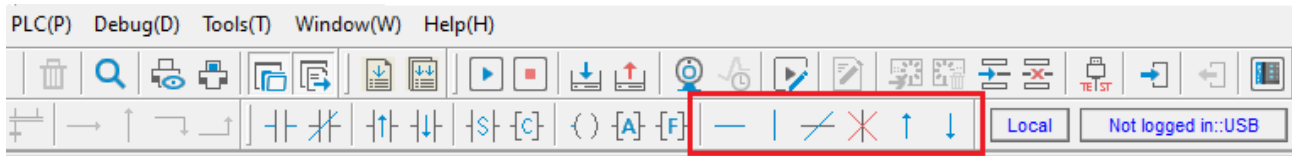
Aby mieć lepszy pogląd na aktualne parametry, można dodać odpowiednie bloczki, które będą odczytywać takie wartości jak pozycja, prędkość, czy moment obrotowy. Aby dodać nowe bloczki do programu, należy wyjść z opcji monitoringu (podglądu online) i można również sterownik PLC przełączyć w tryb STOP.



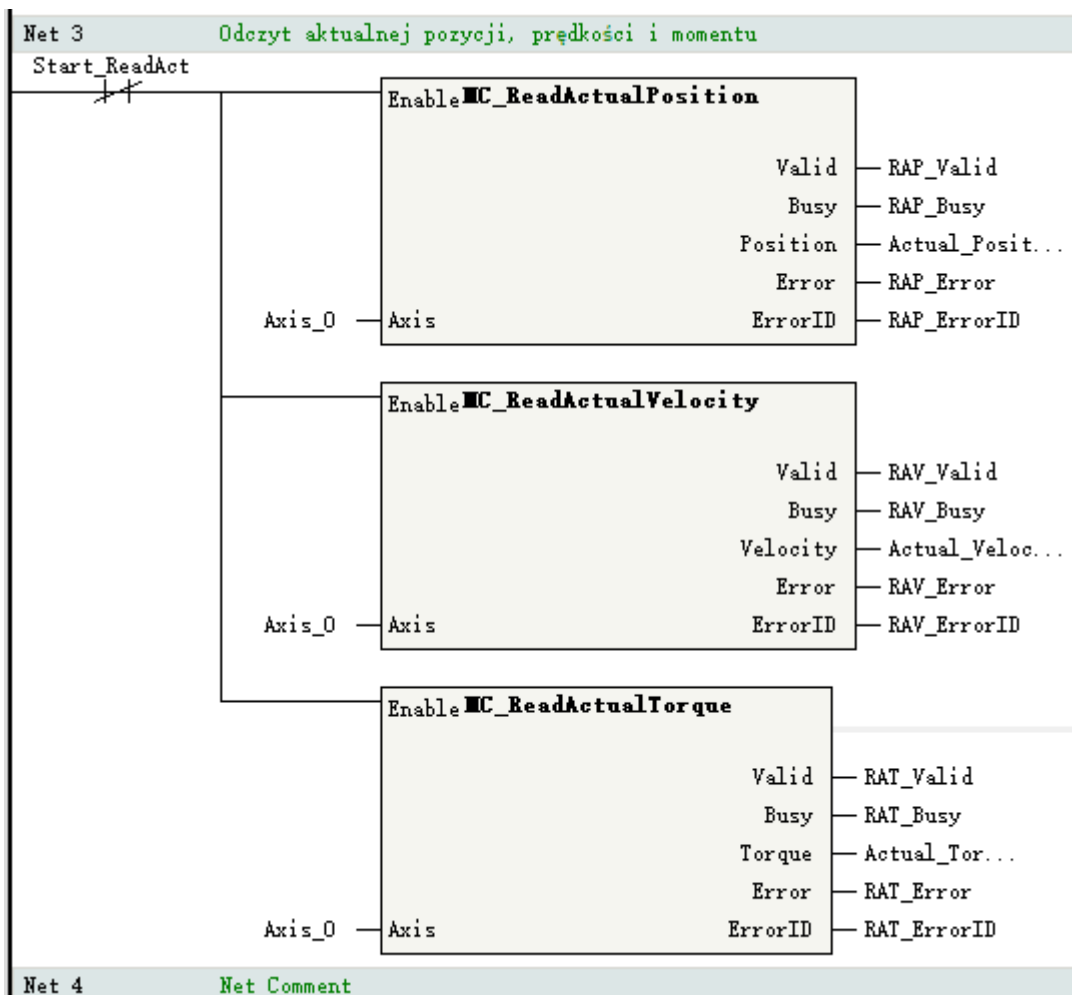
Do drabinki programu trzeba dodać następujące bloczki:

- „MC\_ReadActualPosition”
- „MC\_ReadActualVelocity”
- „MC\_ReadActualTorque”

Można umieścić wszystkie w jednej sieci „Net”, lub w 3 różnych. Aby odczyt parametrów działał zaraz po uruchomieniu sterownika (w trybie RUN) należy styk normalnie otwarty (NO) zastąpić stykiem normalnie zamkniętym (NC). Przy umieszczeniu wszystkich trzech bloczków w jednej sieci dobrze jest posłużyć się odpowiednimi narzędziami tworzenia/usuwania linii:

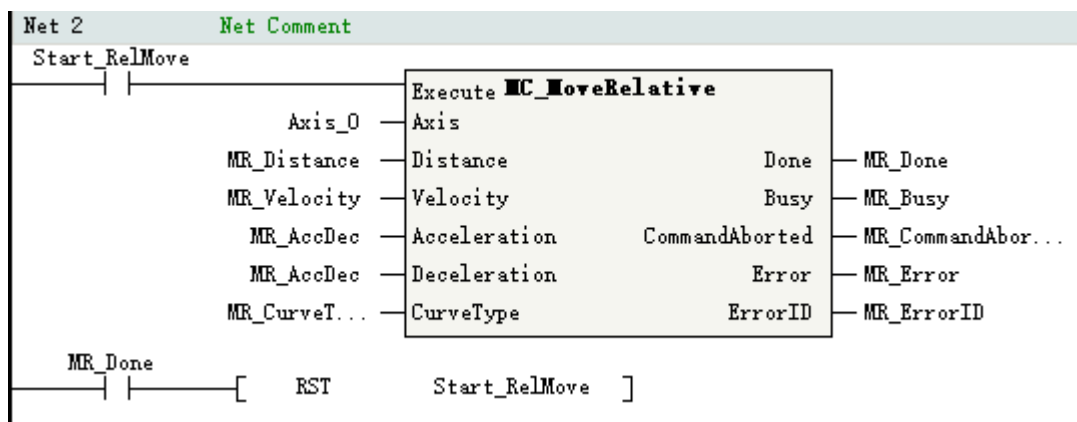
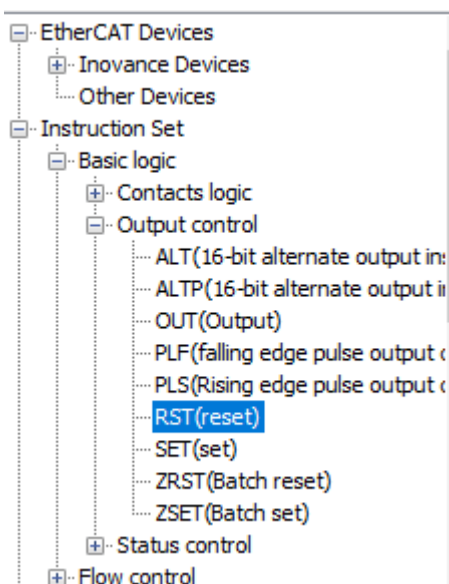


Przykład umieszczenie bloczków i nadania nazw dla odpowiednich wyjść:



Dodatkowo można usprawnić sterowanie ruchem. Aby ręcznie nie było trzeba rozłączać styku „*Start\_RelMove*” można dodać linijkę drabinki programu, która po ukończeniu ruchu (dojazdu na ustawioną pozycję) sama resetuje stan tego styku.

Pod bloczkiem „*MC\_MoveRelative*” dodać trzeba nową linię i wstawić cewkę resetującą:



Jak widać na powyższym schemacie, gdy bloček „*MC\_MoveRelative*” zakończy wykonywanie ruchu, wówczas wyjście „*MR\_Done*” zostanie aktywowane. Jednocześnie spowoduje to załączenie styku w linijce poniżej i tym samym reset styku „*Start\_RelMove*”.

Następnie program można skompilować i wgrać do sterownika PLC. Po tym, przełączyć sterownik w tryb RUN i włączyć monitoring pracy. Po tej czynności w dolnej części oprogramowania pojawi się pusta tabela monitoringu. Można do niej wprowadzić pozycje, które mają być sterowane, lub których wartości mają być obserwowane. Aby dodać wystarczy w tabeli w polu „*Element Name*” wprowadzić nazwy, które zostały użyte do oznaczeń poszczególnych zmiennych, np.:

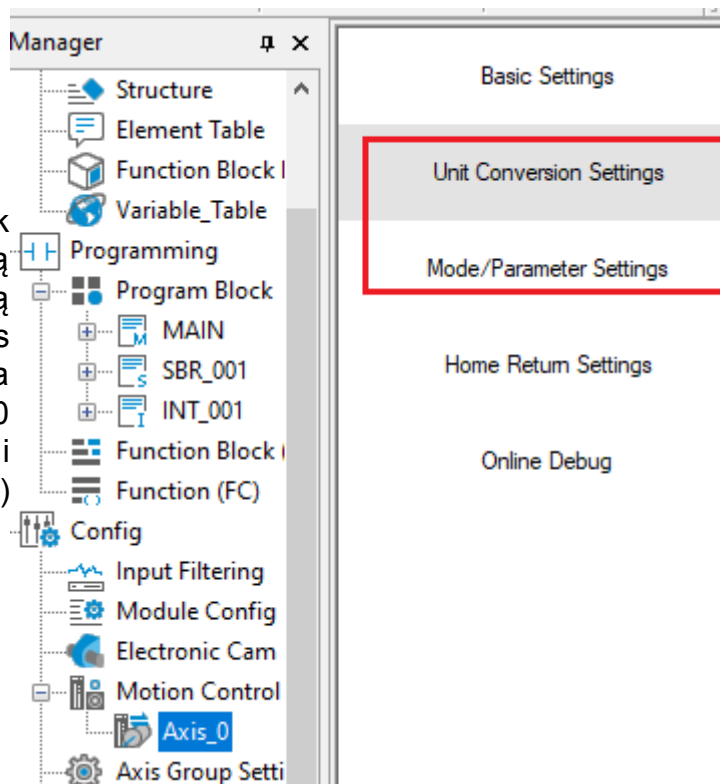
| Element Name | Data Type       | Display Format | Current Value | Comment  |
|--------------|-----------------|----------------|---------------|----------|
| 1            | Servo_On        | BOOL           | Bin           | OFF      |
| 2            | Start_RelMove   | BOOL           | Bin           | OFF      |
| 3            | Actual_Position | REAL           | Dec           | 2115.923 |
| 4            | Actual_Velocity | REAL           | Dec           | 0.000000 |
| 5            | Actual_Torque   | REAL           | Dec           | 0.000000 |
| 6            | MR_Distance     | REAL           | Dec           | 180.0000 |
| 7            | MR_Velocity     | REAL           | Dec           | 200.0000 |
| 8            | MR_AccDec       | REAL           | Dec           | 1000.000 |
| 9            |                 |                |               |          |
| 10           |                 |                |               |          |

W polu „Current Value” można odczytać i/lub zmienić aktualną wartość / stan.

## 6. Konfiguracja osi.

Podczas pracy sterownika PLC z serwonapędem, ważne jest odpowiednie skonfigurowanie osi. Do poprawnej pracy w maszynie, należy określić takie parametry jak liczbę impulsów na obrót, przełożenie, posuw liniowy/kołowy, czy tryb pomiaru inkrementalny/absolutny. Ustawień tych należy dokonać w zakładkach „Unit Conversion Settings” oraz „Mode/Parameter Settings”:

Przykładowo konwersji jednostek poniżej, spowodują ruch z prędkością oraz dystans serwo silnik wykona prędkością 200 przyspieszenia i odpowiednio krótki) pełnych obrotów.



parametry ustawione jak że przy wykonaniu ustawioną na 200 ustawiony na 600, ruch dokładnie z obr/min (o ile czas hamowania jest oraz wykona 10

Basic Settings

**Unit Conversion Settings**

Mode/Parameter Settings

Home Return Settings

Online Debug

Unit Conversion Settings:

Reverse

Number of pulses in one turn by motor/encoder:  Instruction Pulse  Decimal

Do not use gearbox

The amount of movement of the worktable in a circle:  Unit

Pulse number =  $\frac{\text{Number of pulses rotated by motor/encoder[DINT]}}{\text{Moving amount of worktable rotation[REAL]}}$  \*Moving distance(Unit)

Use gearbox

The amount of movement of the worktable in a circle:  Unit

Gear ratio molecule (number of teeth in (5) below):

Gear ratio denominator (number of teeth in (4) below):

Axis type is linear mode:

Pulse number =  $\frac{\text{Number of pulses rotated by motor/encoder[DINT]} * \text{Numerator of gear ratio}}{\text{Moving amount of worktable rotation[REAL]} * \text{Denominator of gear ratio}}$  \*Moving distance(Unit)

W zakładce „Mode/Parameter Settings” można ustawić odpowiednie limity prędkości, momentu obrotowego, czasu hamowania i przyspieszania oraz inne.

Mode Selection:

Encoder Mode  Incremental Mode  Absolute Mode

Mode Settings  Linear Mode  Rotation Mode

Software Limits  Enable

Negative limit value:  Unit Forward Limit:  Unit

Error Response Limit deceleration:  Unit/s<sup>2</sup> Axis Failure Deceleration:  Unit/s<sup>2</sup>

Threshold setting Follow error threshold:  Unit Speed reaches threshold:  Unit/s

Axis Speed Settings Max speed:  Unit/s Max acceleration:  Unit/s<sup>2</sup>  
Jog Max speed:  Unit/s

Torque setting Max positive torque:  0.1% Max negative torque:  0.1%

Odpowiedni dobór powyższych parametrów zapewnia prawidłową pracę jak i prawidłowe sterowanie serwo napędu.

Znając podstawy tworzenia programu zawarte w tej instrukcji, można stworzyć program do sterowania osią lub wieloma osiami w danej aplikacji.