

ROBOCHALLENGE

Robotyka zintegrowana w sterowniku PLC



Spis treści:

1. Opis sytuacji
2. Opis zadania
3. Programowanie
 - 3.1 Program bazowy
 - 3.2 Etap 1 – Dodanie kinematyki robota i jej parametryzacja
 - 3.3 Etap 2 – Dodanie bloku L_TT1P_PickAndPlaceState
 - 3.4 Etap 3 – Homing i obsługa ręczna z wizualizacji
 - 3.5 Etap 4 – Ustawienie trajektorii ruchu
4. Obsługa programu PLC Designer v3.xx
5. Pomocne linki
6. System oceniania

1. Opis sytuacji

Czeka cię poważny projekt, w którym pierwszy raz masz użyć robota sterowanego z PLC. Projekt będzie realizowany na wyjeździe a założenia nadal są mgliste. Aby przygotować się na najgorsze i podnieść swoje kwalifikacje, zabierasz się z zapałem do pisania pierwszego testowego programu w zaciszu swojego biura. Na szczęście stanowisko testowe wypożyczył Ci producent sprzętu.

2. Opis zadania

Należy napisać program testowy na sterownik PLC oraz ustawić trajektorię, która będzie zgodna z rysunkiem na stanowisku testowym. Zadanie zostało podzielone na 4 etapy:

- 2.1. Należy dodać do projektu odpowiednią kinematykę robota. Zlinkować do niej 2 pierwsze osie serwonapędów oraz ustawić parametry mechaniczne.
- 2.2. Teraz dodaj blok do obsługi robota L_TT1P_PickAndPlaceState z biblioteki L_TT1P_TechnologyModules_LM (nazwa instancji bloku = **PickPlace**). Do bloku należy podpiąć wcześniej skonfigurowaną kinematykę(AxesGroup) oraz strukturę wcześniej przygotowanych parametrów o nazwie **g_scPar**.
- 2.3. Wywołaj homing(bazowanie) robota. Dodaj i zlinkuj wizualizację do bloku L_TT1P_PickAndPlaceState. Sprawdź czy wizualizacja jest w pełni funkcjonalna wywołując ruchy ręczne w osiach X-Z (JogPos i JogNeg).
- 2.4. Dzięki wcześniej dodanej wizualizacji ustaw trajektorię ruch robota która pokryje się z wyrysowanym przebiegiem na stanowisku. Zwróć uwagę na przerywane linie, które sugerują, że robot powinien płynnie przejechać w pobliżu wierzchołków

PAMIĘTAJ!

- Nie należy przechodzić do kolejnego etapu przed ukończeniem poprzedniego.
- Wskazówek udziela reprezentant firmy LENZE, jednak za udzielane odpowiedzi punkty są potrącane.
- W zadaniu oceniana jest poprawność wykonania kolejnych etapów oraz czas wykonania całego zadania. Za ukończenie całego zadania przydzielane są dodatkowe punkty.
- W razie niewykonania całego etapu o punktacji decyduje jury.

W celu przyspieszenia zadania udostępniono program, który ma już zaimplementowane następujące funkcje:

- W projekcie są już dodane i sparametryzowane osie serwo i700 działające na sieci EtherCAT.
- W głównym programie PLC_PRG są już deklaracje potrzebnych bloków programowych oraz 2 akcje:
 - a. PAR_ACT – ustawienie bezpiecznych parametrów ruchu.
 - b. HOME_ACT – Procedura homingu (jest nieco złożona).
- W zakładce zmiennych globalnych (GVL) jest deklaracja parametrów bloku L_TT1P_PickAndPlaceState. Zmienna globalna o nazwie **g_scPar**.
- Język programu został już wybrany jako Function Block Diagram (FBD), ale jak się przekonasz nie wiele trzeba będzie programować – nie ma to więc większego znaczenia.

3. Programowanie

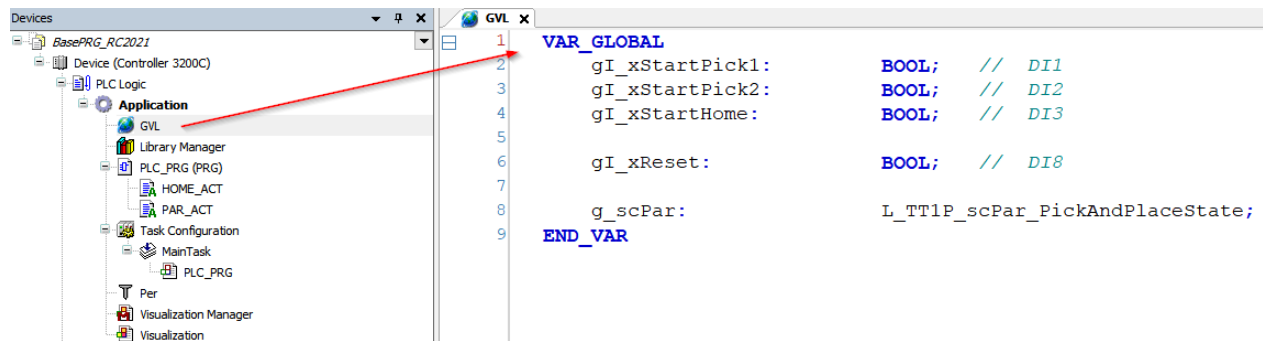
3.1. Program bazowy:

Tak wygląda drzewko programu bazowego. Nie przejmuj się kilkoma błędami kompilacji, po dodaniu właściwej kinematyki robota błędy znikną.

Powyżej deklaracja zmiennych:

```
1 | PROGRAM PLC_PRG
2 | VAR
3 |     // Główny blok do obsługi robota
4 |     PickPlace:           L_TT1P_PickAndPlaceState;
5 |
6 |     // Bloki i zmienne pomocnicze
7 |     TorqueLimit_A1:      L_MC1P_SetTorqueLimit;
8 |     TorqueLimit_A2:      L_MC1P_SetTorqueLimit;
9 |     CartesianTransform_PCS1: MC_SetCartesianTransform;
10 |     ActualPosition_PCS1: MC_GroupReadActualPosition;
11 |     iHomeStep:           INT;
12 |     iHomeIndex:          INT;
13 |     xHomeBusy:           BOOL;
14 |     xHomeDone:           BOOL;
15 |
16 |     // Miejsce na twoje zmienne
17 |
```

Dodatkowe zmienne globalne dostępne są na dodatkowych listach:

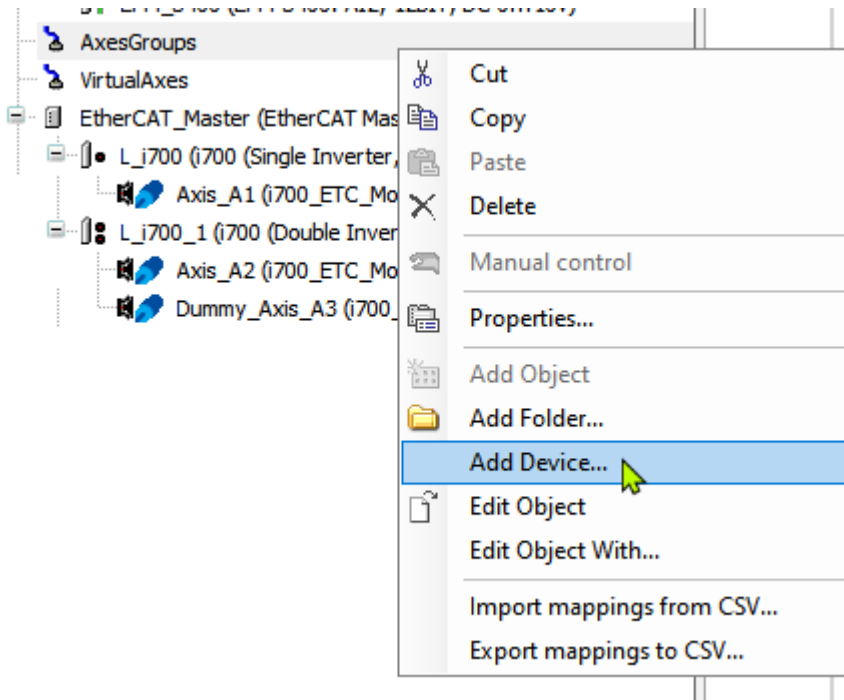


The screenshot shows a software interface with a project tree on the left and a variable declaration window on the right. The project tree is titled 'Devices' and shows a hierarchy: 'BasePRG_RC2021' -> 'Device (Controller 3200C)' -> 'PLC Logic' -> 'Application' -> 'GVL'. A red arrow points from the 'GVL' folder in the tree to the 'VAR_GLOBAL' window. The 'VAR_GLOBAL' window contains the following code:

```
1 | VAR_GLOBAL
2 |     gI_xStartPick1:      BOOL; // DI1
3 |     gI_xStartPick2:      BOOL; // DI2
4 |     gI_xStartHome:       BOOL; // DI3
5 |
6 |     gI_xReset:           BOOL; // DI8
7 |
8 |     g_scPar:             L_TT1P_scPar_PickAndPlaceState;
9 | END_VAR
```

3.2. Etap 1 – dodanie kinematyki robota i jej parametryzacja

Należy kliknąć prawym na AxesGropus i wybrać Add Device...:



Teraz wybierz właściwa kinematykę i zrób kompilację programu:

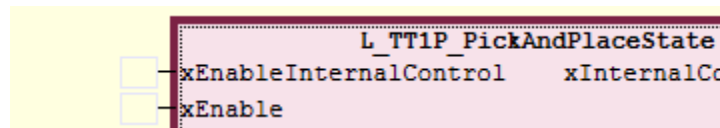


W komponencie kinematyki zlinkuj 2 osie rzeczywiste Axis_A1 i Axis_A2 oraz ustaw stałą posuwu (Feed Constant) dla kół zębatych. W zestawie testowym wynosi ona 120mm. Przełożenia przekładni planetarnych zostały już ustawione w osiach rzeczywistych.

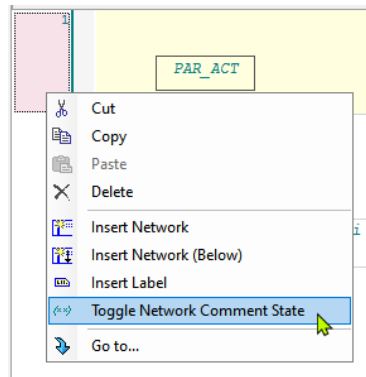
3.3. Etap 2 – Dodanie bloku L_TT1P_PickAndPlaceState

W trzecim networku głównego programu PLC_PRG dodaj blok do obsługi robota typu L_TT1P_PickAndPlaceState. Jego deklaracja jest już dodana w programie „PickPlace” należy jej użyć aby działały inne gotowe części programu. Do bloku podłącz konieczną kinematykę robota oraz zmienną globalną z parametrami „g_scPar”.

Nie zapomnij także dać 2 zezwolenia (sterowanie z wew. wizualizacji oraz odblokowanie bloku), wystarczy napisać „TRUE”:



Teraz odkomentuj dwa pierwsze networki:



Sprawdź czy program kompiluje się bez błędów.

UWAGA!

Nie zmieniaj nazw, kinematyki robota ani deklaracji bloku L_TT1P_PickAndPlaceState „PickPlace”, zmiennej z parametrami „g_scPar”. Są one podpięte w gotowych częściach programu (PAR_ACT i HOME_ACT).

HELP: Po prostu kliknij bloczek i wciśnij <F1>. Pojawi się okno z opisem.

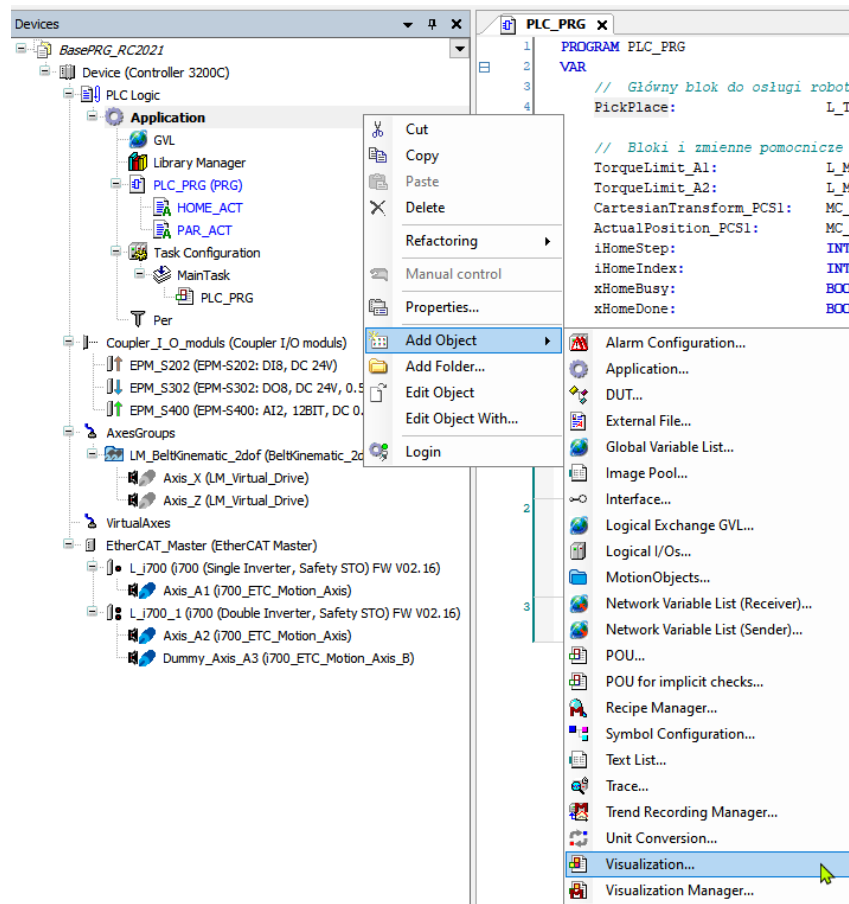
3.4. Etap 3 – Homing i obsługa ręczna z wizualizacji

Homing wywoływany jest zmienną globalną `gl_xStartHome` podpiętą do wejścia DI3. Jeżeli poprawnie wykonałeś poprzednie etapy, robot powinien wykonać homing po wciśnięciu DI3.

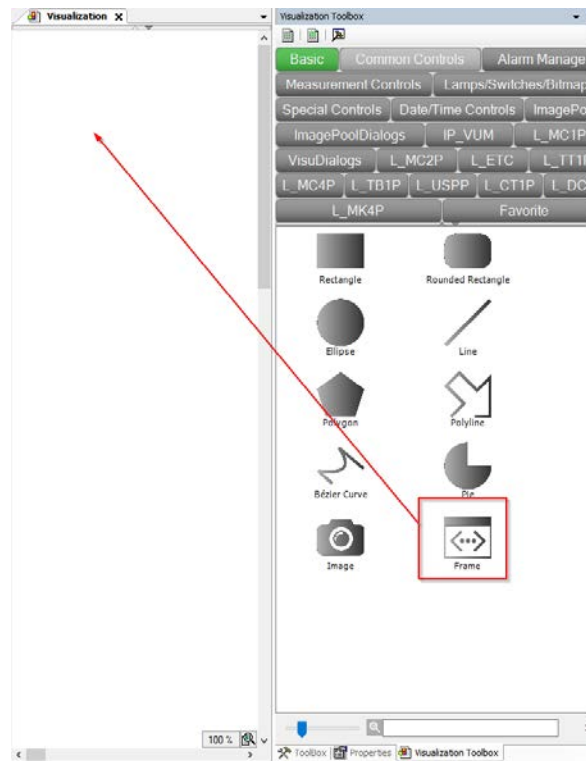
Procedura homingu składa się z 4 kroków:

- Szukanie bazy w dolnym limicie mechanicznym
- Szukanie bazy w lewym limicie mechanicznym
- Przesunięcie układu współrzędnych (robot nie jest w stanie dojechać do punktu (0,0))
- Podjazd do punktu początku trajektorii (30,30).

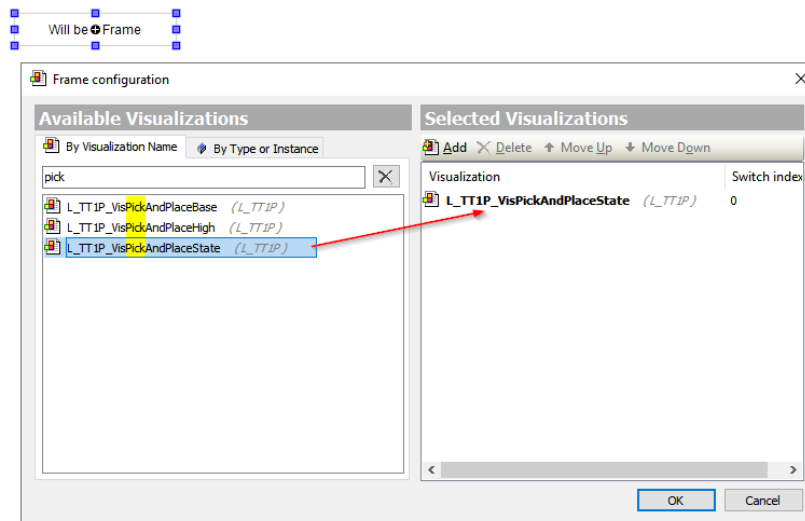
Teraz dodaj element wizualizacji:



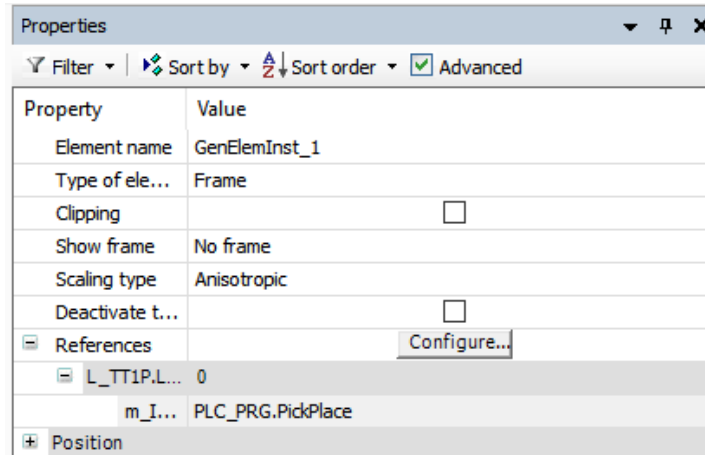
Przecignij i upuść element Frame:



Wybierz właściwy element wizualizacji:

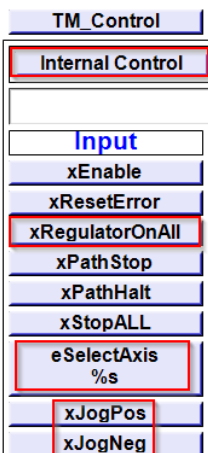


Oraz podepnij go z instancją bloku: PLC_PRG.PickPlace



Po podłączeniu online posteruj osiami X-Z w trybie ręcznym (pamiętaj że robot musi być wcześniej zbazowany).

- Załącz Internal Control (sterowanie z wew. wizualizacji)
- Odblokuj grupę osi robota RegulatorOnAll
- Wybierz oś X lub Z w polu SelectAxis
- Jedź do przodu / do tyłu komendami JogPos / JogNeg
Bezpieczne prędkości są już ustawione w programie



Pamiętaj że wciśnięcie przycisku <Internal Control> powoduje włączenie wew. sterowania z wizualizacji a tym samym serowanie z programu za pomocą przyłączy bloku jest nieaktywne. Czyli nie zadziała Homing z wejścia DI3!!! Oczywiście można włączać i wyłączać <Internal control> dowolną ilość razy.

3.5. Etap 4 – Ustawienie trajektorii ruchu

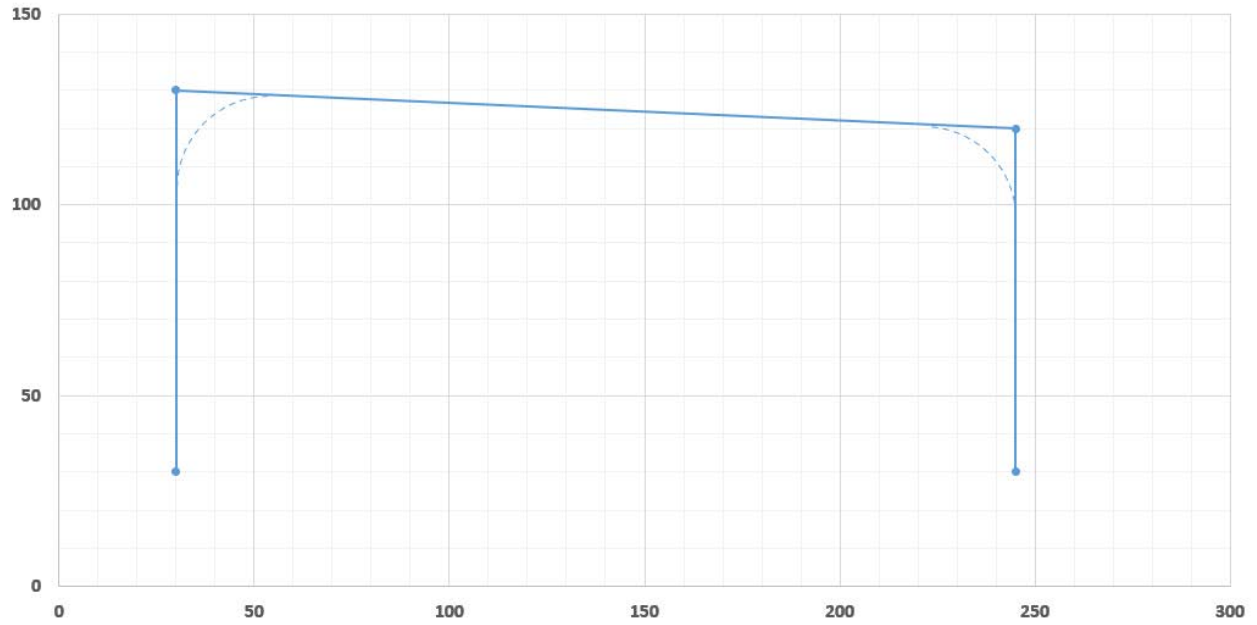
Planowanie trajektorii przeprowadza się w zakładce parametrów pod przyciskiem ascProfilePar

TM_Control	TM_Parameter	Limitations	AGBC	
L_TT1P_PickAndPlaceState				
PLC_PRG.PickPlace				
L_TT1P_scProfilePar X				
SelectPointNumber 1				
PointNumber	1	2	3	4
eCoordSystem	WCS	WCS	WCS	WCS
IrXPos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
IrYPos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
IrZPos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
IrAPos	180.00[u]	180.00[u]	180.00[u]	180.00[u]
IrBPos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
IrCPos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
IrAux1Pos	0.00[u]	0.00[u]	0.00[u]	0.00[u]
IrPathVel	0.00[u/s]	0.00[u/s]	0.00[u/s]	0.00[u/s]
IrPathAcc	0.00[u/s ²]	0.00[u/s ²]	0.00[u/s ²]	0.00[u/s ²]
IrPathDec	0.00[u/s ²]	0.00[u/s ²]	0.00[u/s ²]	0.00[u/s ²]
IrBlendingRadius	0.00[u]	0.00[u]	0.00[u]	0.00[u]
xHalt	xHalt	xHalt	xHalt	xHalt

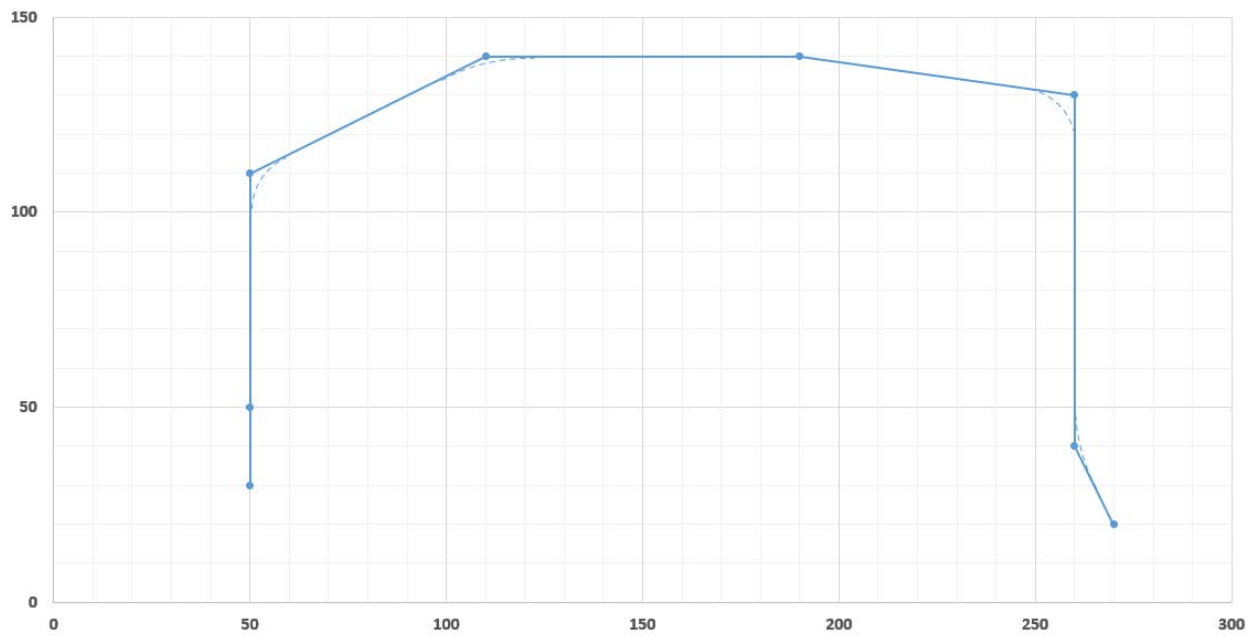
Pamiętaj aby używać systemu koordynacji PCS_1(wpisz „1”). Robot porusza się po płaszczyźnie X-Z. Osie dodatkowe A,B,C nie są używane. Aby wykonać wyoblania trajektorii użyj BlendingRadius. Zalecamy prędkość ścieżki **100mm/s**, przyspieszenia **250mm/s²**.

Do wyboru jest trajektoria 4 lub 8 punktowa (punktujemy odpowiednio 0,5p i 1p):

Trajektoria 1



Trajektoria 2



Jazdę startujemy komendą:

`xExecPickAndPlace`

UWAGA!!!

Jeżeli w trakcie parametryzacji trajektorii wyłączysz <Internal control> twoja tablica parametrów wróci do wartości domyślnych i stracisz część swojej pracy. Zrób bazowanie/homing i dopiero potem utwórz całą trajektorię!!!

4. Obsługa programu PLC Designer v3.xx

4.1. Kompilacja programu skrót <F11> lub



4.2. Ładowanie programu do sterownika PLC



Następnie w celu wgrania programu do sterownika należy wybrać jedną z dwóch opcji:

Login with online change,
Login with download.

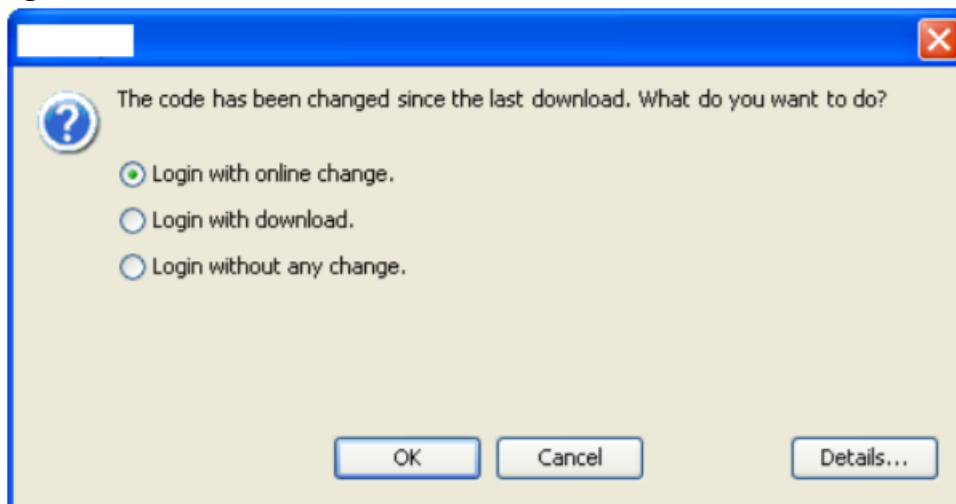


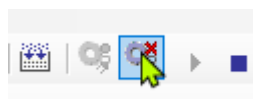
Abb. 33 Login dialog

Login with online change: This option is selected per default. So if you confirm the dialog with **OK**, the modifications will be loaded and immediately shown in the online view (monitoring) of the respective object(s).

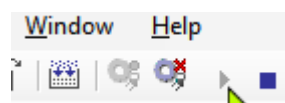
Login with download: Activate this option if the application project should be compiled and loaded completely.

Login without any change: Activate this option in order to keep the program running on the controller unchanged. Afterwards an explicit download (Download) might be done, thus loading the complete application project, or at the next re-login you will be asked again whether an online change should be done.

4.3. Rozłączenie ze sterownikiem



4.4. Start i Stop sterownika



4.5. Zmienne globalne

```
GVL x
1  VAR_GLOBAL
2      gI_xStartPick1:          BOOL;    //  DI1
3      gI_xStartPick2:          BOOL;    //  DI2
4      gI_xStartHome:           BOOL;    //  DI3
5
6      gI_xReset:                BOOL;    //  DI8
7
8      g_scPar:                  L_TT1P_scPar_PickAndPlaceState;
9  END_VAR
```

5. Pomocne linki

Ogólna dokumentacja programu narzędziowego PLC Designer v3.xx:

[https://download.lenze.com/TD/PLC%20Designer_PLC%20Designer%20\(R3-1\)_v4-1_EN.pdf](https://download.lenze.com/TD/PLC%20Designer_PLC%20Designer%20(R3-1)_v4-1_EN.pdf)

Dokumentacja bloku L_TT1P_PickAndPlaceState (Proszę zapoznać się z wersją „State”).

Należy zwrócić uwagę na zadawanie trajektorii:

https://download.lenze.com/TD/FAST_TM%20Pick%20and%20Place_v5-0_EN.pdf

Filmy instruktarzowe opisujące podstawy obsługi programu PLC Designer v3.xx:

https://www.youtube.com/watch?v=a-x4UY9xjaM&list=PLQCcCRGE4bNtLcBky_Uu5-zklgGHou3x

Wersja demo (30 dni) oprogramowania PLC Designer v3.xx:

<https://www.lenze.com/pl-pl/serwis/oprogramowanie-do-pobrania/narzedzia-inzynieryjne-easy/>

6. System oceniania

Za wykonanie etapów:

- Etap 1 – **1punkt**
Poprawnie dodana kinematyka robota, zlinkowane osie rzeczywiste, ustawiona mechanika (obwód koła pasowego – Feed constant)
- Etap 2 – **1 punkt**
Dodanie do programu głównego bloku kontrolującego ruch robota, połączenie go z kinematyką, połączenie ze strukturą parametrów.
- Etap 3 – **1 punkt**
Wykonanie Homingu, Integracja wizualizacji wewnętrznej, ręczne ruchy robotem X-Z z wizualizacji
- Etap 4 – **0,5 punktu** za trajektorie 4 punktową / **1 punkt** za trajektorie 8 punktową
Poprawne wpisanie trajektorii – robot podąża po śladzie

Czas wykonania zadania:

- Wykonanie zadania do 20 min – **1 punkt**
- Wykonanie zadania do 25 min – **2 punkty**
- Wykonanie zadania do 30 min – **3 punkty**

Realizacja całości – 1 punkt

