

ROBOCHALLENGE

Beckhoff PLC

**Zarządzanie pracą robota
z poziomu sterownika PLC**

1. Opis zadania

Zadanie polega na uruchomieniu komunikacji z Robotem FANUC i przesłaniu do niego danych parametryzujących proces paletyzacji.

W zadaniu zostanie wykorzystana komunikacja po protokole EtherCAT. Kompaktowy komputer przemysłowy **Beckhoff C6015** będzie w tej komunikacji pełnił rolę mastera EtherCAT'a, natomiast robot **FANUC LR Mate 200iD/7L** będzie miał zaimplementowaną funkcję EtherCAT Slave. Kolejnym krokiem, po nawiązaniu komunikacji i wykonaniu linkowania zmiennych zadeklarowanych w module TwinCAT PLC, będzie oprogramowanie sekwencji przesyłania parametrów paletyzacji do robota.

Po pomyślnym przesłaniu parametrów robot wykona paletyzację zgodną z przesłanymi parametrami.

Do dyspozycji zawodników będzie:

- **Beckhoff Embedded PC C6015**

<https://www.beckhoff.com/pl-pl/products/ipc/pcs/c60xx-ultra-compact-industrial-pcs/c6015-0010.html>

- **FANUC LR Mate 200iD/7L**

<https://www.fanuc.eu/pl/pl/roboty/robot-strona-filtrowania/lrmate-series/lrmate-200id-7l>

Zadanie zostało podzielone na cztery etapy, za każdy ukończony etap drużyna otrzyma 1 punkt.

Etapy zadania:

1. Uruchomienie komunikacji z robotem po protokole EtherCAT:
 - a. Dodanie pliku konfiguracyjnego robota do systemu.
 - b. Skonfigurowanie magistrali EtherCAT.
 - c. Zadeklarowanie i wykonanie linkowania zmiennych służących do realizacji komunikacji.
 - d. Odczytanie wartości wybranych zmiennych wysyłanych przez robota.
2. Przesłanie do robota sygnałów pozwolenia na pracę.
3. Zaprogramowanie procedury handshake'u pomiędzy PLC a robotem i przesłanie, co najmniej, jednego z parametrów paletyzacji.
4. Zautomatyzowanie procedury przesyłania parametrów, tak aby na żądanie programisty wszystkie parametry zostały kolejno wysłane do robota.

Realizacja zadania – etap 1

Na tym etapie należy dodać plik konfiguracyjny robota do systemu TwinCAT, skonfigurować magistralę EtherCAT oraz zadeklarować i zalinkować zmienne, które będą wykorzystywane na dalszych etapach tego zadania.

1. Dodanie pliku konfiguracyjnego do odpowiedniego katalogu.

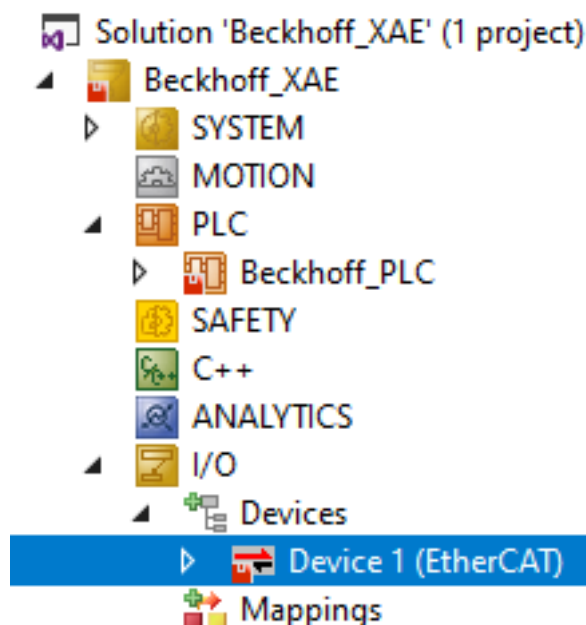
- a) Na pulpicie w katalogu o nazwie „Pliki_do_zadania” znajduje się plik konfiguracyjny robota – „fanuc_rs_esi0002.xml”
- b) Plik konfiguracyjny należy przekopiować do katalogu: „C:\TwinCAT\3.1\Config\lo\EtherCAT”.

c) Po umieszczeniu pliku we wskazanym katalogu należy załadować pliki konfiguracyjne do systemu TwinCAT 3 XAE.

Aby to wykonać wracamy do środowiska TwinCAT, następnie z górnego paska narzędzi, wybieramy: „TwinCAT” -> „EtherCAT Devices” -> „Reload Device Descriptions“.

2. Skanowanie magistrali w poszukiwaniu podłączonego do niej robota.

Rozwijamy drzewo projektu w następujący sposób:



a) Konfiguracja karty sieciowej do obsługi protokołu EtherCAT.

Klikamy dwukrotnie na „Device 1 (EtherCAT)” następnie, w prawej części, wybieramy zakładkę „Adapter”.

Klikamy na przycisk „Search...” i w nowo otwartym oknie „TwinCAT Ethernet Compatible Devices” wybieramy kartę sieciową o nazwie „EtherCAT” i klikamy „OK”.

b) Skanowanie magistrali.

Po zakończeniu konfiguracji karty sieciowej, klikamy prawym przyciskiem myszy na urządzenie „Device 1 (EtherCAT)” i z menu kontekstowego wybieramy opcję „Scan”.

Następuje wyszukanie wszystkich urządzeń podłączonych do magistrali EtherCAT.

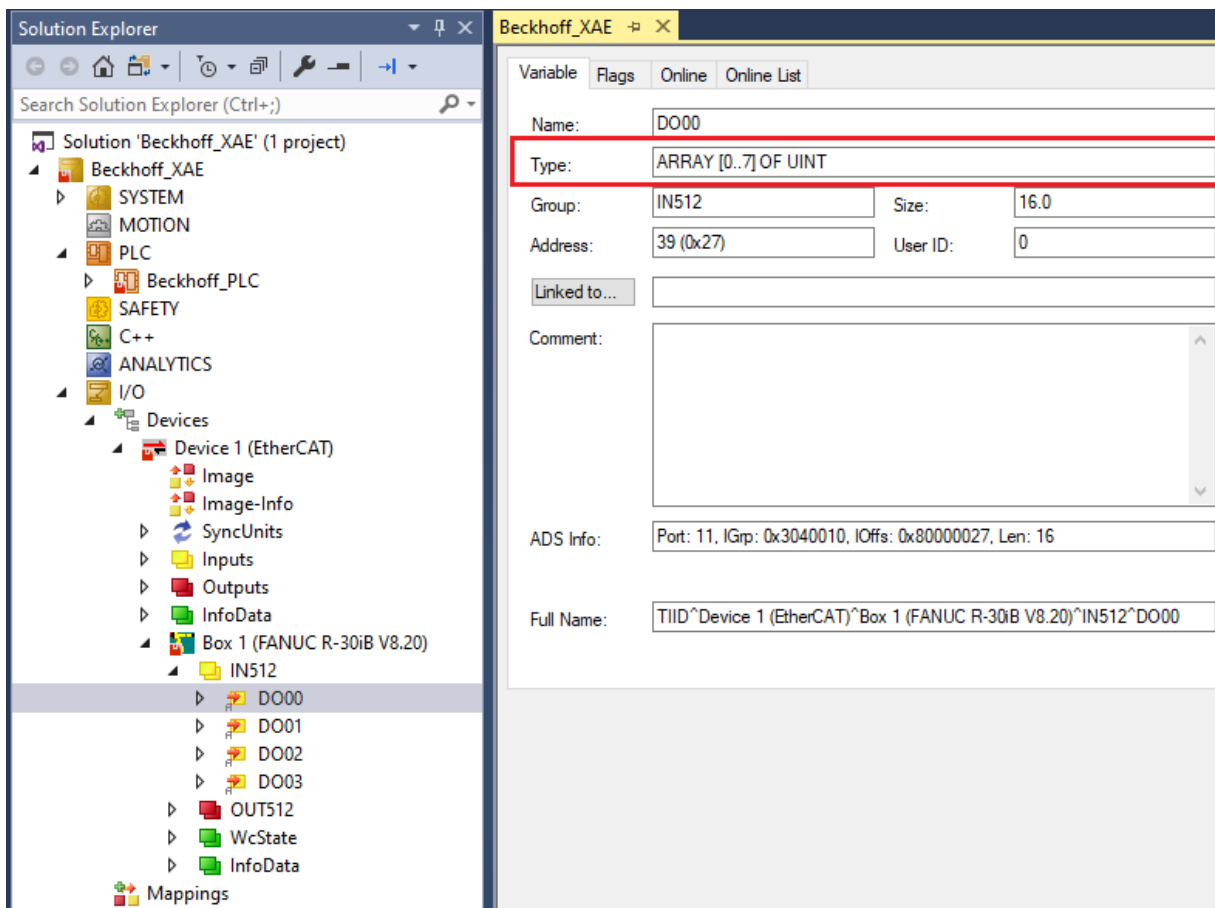
Po zakończeniu skanowania otwiera się okno pokazujące, jakie urządzenia zostały znalezione na magistrali komunikacyjnej.

Konfigurator magistrali powinien zaproponować konfigurację składającą się tylko z robota FANUC o nazwie „Box 1 (FANUC R-30iB Plus)”.

Sprawdzamy poprawność i zatwierdzamy zaproponowaną przez TwinCAT’a konfigurację.

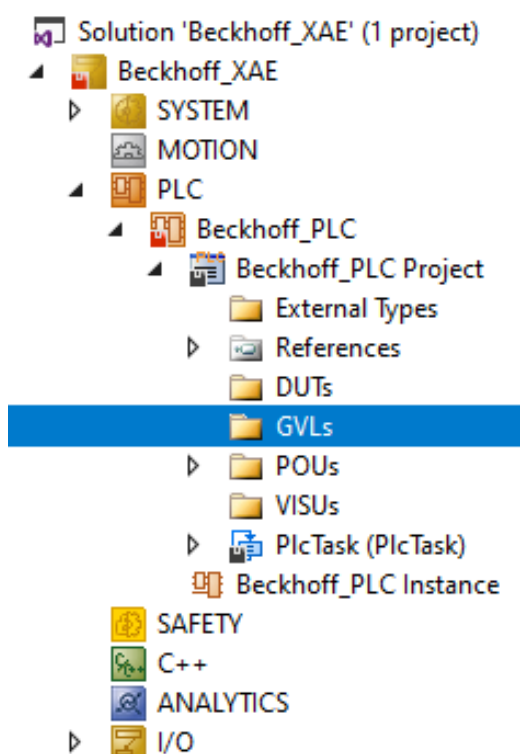
3. Deklaracja i linkowanie zmiennych.

W projekcie PLC należy zadeklarować odpowiednie zmienne, które zostaną zlinkowane z IO udostępnianymi przez robota. Aby poprawnie zadeklarować zmienne wejściowe i wyjściowe należy sprawdzić jakiego powinny być typu, można to zrobić w tym miejscu:



Gdy wiemy już jakiego typu powinny być zmienne, które będziemy linkować do zmiennych udostępnionych przez robota, należy odpowiednio zadeklarować je w projekcie PLC.

Wchodzimy w projekt PLC i w katalogu „GVLs” tworzymy plik (w instrukcji zastosowano nazwę „gvlVariablesToBeLinked”) do przechowania zmiennych globalnych (PPM -> „Add” -> „Global Variable List”):



W stworzonym pliku deklarujemy zmienne w taki sposób aby można było je zalinkować do zmiennych udostępnionych przez robota.

Przykładowe deklaracje zmiennych pozwalające na ich linkowanie z fizycznymi IO:

```
iTest_IN AT %I*: INT;  
bTest_IN AT %I*: BOOL;  
arrTest_IN AT %I*: ARRAY [0..10] OF INT;  
iTest_OUT AT %Q*: INT;  
bTest_OUT AT %Q*: BOOL;  
arrTest_OUT AT %Q*: ARRAY [0..10] OF INT;
```

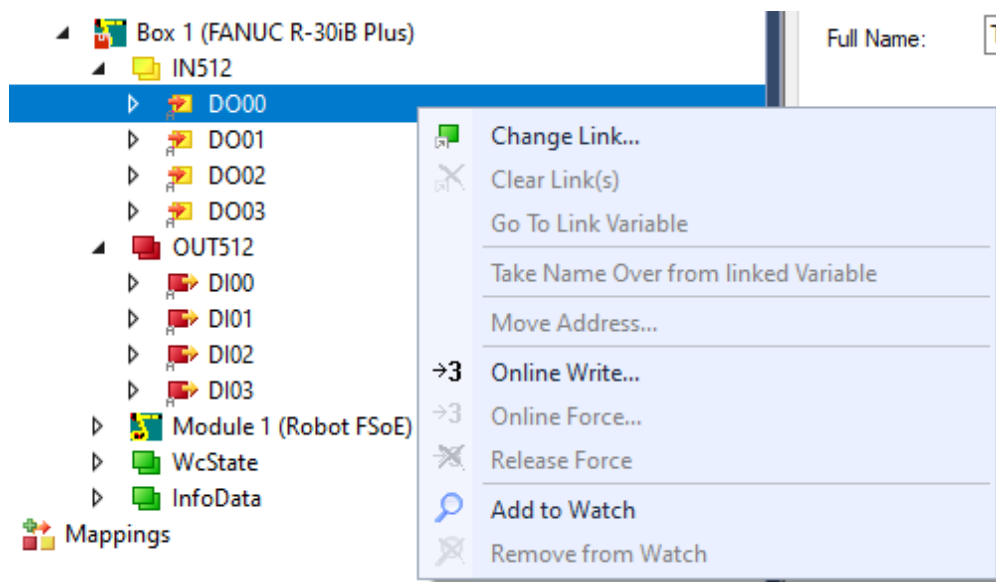
Po zadeklarowaniu zmiennych należy skompilować projekt.

Można to zrobić wybierając z menu następującą opcję:

„Build” -> „Build Solution” lub skorzystać z kombinacji klawiszy „Ctrl + Shift + B”.

Po pomyślnym zakończeniu kompilacji można przystąpić do linkowania zmiennych.

W tym celu klikamy PPM na zmienną, którą chcemy zalinkować i wybieramy „Change Link...”.



Otworzy się okno, w którym można wybrać zmienne (zadeklarowane wcześniej w projekcie PLC) do podlinkowania we wskazane miejsce.

Po zalinkowaniu wszystkich zmiennych (wejściowych i wyjściowych) należy aktywować konfigurację poprzez wybranie na pasku menu ikony „Activate Configuration” i uruchomić system w trybie „Run” z włączonym autostartem programu PLC.

Po uruchomieniu sterownika można sprawdzić czy zmienne są wymieniane pomiędzy PLC i robotem.

Realizacja zadania – etap 2

Na tym etapie należy przesłać do robota zestaw pozwoleń na pracę / ruch.

Przesłanie pozwoleń na pracę / ruch.

Na tym etapie należy przesłać do robota zestaw sygnałów, który zostanie zinterpretowany jako pozwolenie na pracę / ruch robota.

W tym celu należy wystawić stany wysokie na odpowiednie bity w zmiennych przesyłanych do robota, które odpowiadają sygnałom: „IMSTP”, „Hold”, „SFSPD”, „Enable”.

Sygnały są zlokalizowane na następujących pozycjach (zmienna „arrDataForRobot_00” jest zlinkowana z DI robota, nazwy zmiennych przykładowe):

```
arrDataForRobot_00[0].0 := TRUE; //IMSTP;  
arrDataForRobot_00[0].1 := TRUE; //Hold;  
arrDataForRobot_00[0].2 := TRUE; //SFSPD;  
arrDataForRobot_00[0].7 := TRUE; //Enable;
```

Po odpowiednim uzupełnieniu kodu w projekcie PLC należy zalogować się do sterownika w celu aktualizacji programu PLC. Robimy to wykorzystując zieloną ikonę o nazwie „Login”, która znajduje się na górnym pasku menu. Po kliknięciu tej ikony nastąpi kompilacja programu. Jeśli wykona się poprawnie program zostanie wgrany do sterownika PLC.

Na panelu robota sprawdzamy czy robot dostał wszystkie niezbędne pozwolenia na pracę, np. sygnał „Hold” nie powinien być aktywny.

Realizacja zadania – etap 3

Na tym etapie należy oprogramować procedurę przesyłania parametrów paletyzacji do robota, tzw. *handshake* pomiędzy PLC i robotem.

W celu uruchomienia paletyzacji należy przesać do robota wszystkie 8 parametrów w zadanej kolejności (w nawiasach podane wartości startowe):

*Number Of Cycles (1), Load Number (4), Load Type (1), Pallet Number (1)
Layer Number (1), Unit Number (0), Infeed Number (1), Alternate Infeed (0)*

Number Of Cycles (1), Load Number (4), Load Type (1), Pallet Number (1)

Layer Number (1), Unit Number (0), Infeed Number (1), Alternate Infeed (0)

Do podawania wartości kolejnych parametrów procesu paletyzacji należy wykorzystać **tą samą zmienną, czyli GI[1]**. Program, zaimplementowany w robocie, będzie z tej zmiennej pobierał wartość i zapisywał w odpowiednim rejestrze pamięci robota (dlatego parametry muszą być przesyłane w podanej wyżej kolejności).

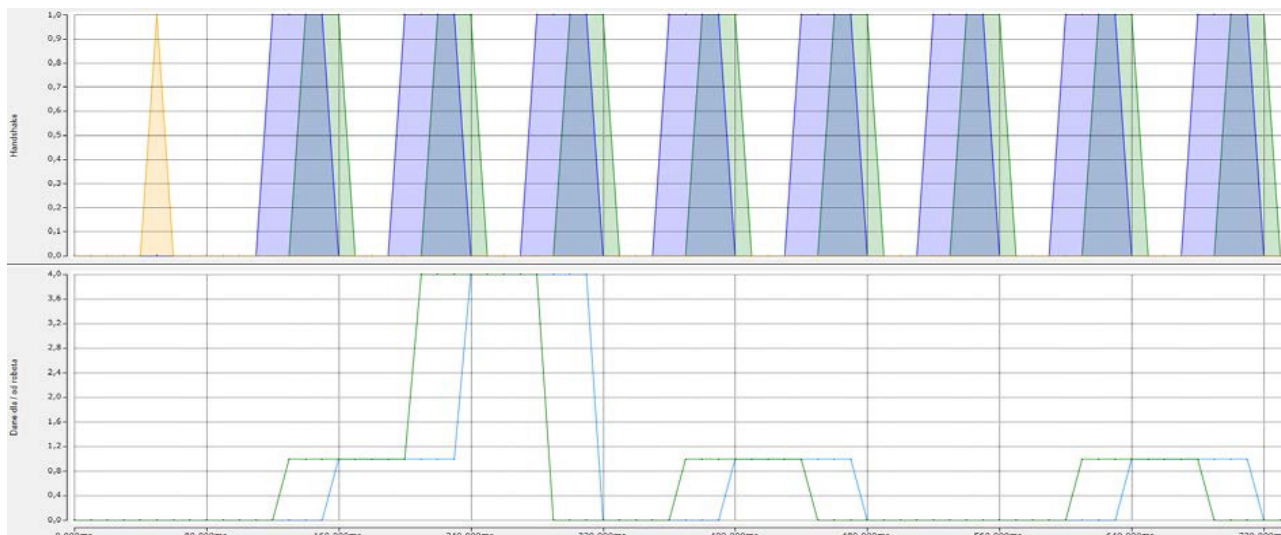
Procedura, przesłania pojedynczego parametru, wygląda następująco:

<ul style="list-style-type: none">• ...• Robot wystawia na DO[121] sygnał gotowości do pobrania danych.• Robot czeka na sygnał z PLC, że dane są przygotowane do pobrania.• ...• ...• ...• Robot zapisuje u siebie dane z GI[1] w odpowiednim rejestrze.• Robot wystawia pobrane dane (jako echo) na GO[1]• Robot kasuje sygnał na DO[121]• Robot czeka aż PLC skasuje sygnał na DI[121]• ...	<ul style="list-style-type: none">• PLC wystawia sygnał „ProdStart”• ...• ...• PLC przygotowuje dane dla robota na GI[1].• PLC wystawia sygnał, że dane są gotowe do pobrania na DI[121].• PLC czeka na sygnał, że dane zostały pobrane przez robota.• ...• ...• ...• ...• PLC porównuje dane na GO[1] z danymi, które wysłał -> powinny być jednakowe. PLC kasuje sygnał DI[121]
---	--

Dodatkowy słowny opis procedury przesłania pojedynczego parametru:

1. PLC ustawia sygnał „ProdStart” na „1” i czeka na sygnał „DO[121]” – robot gotowy do pobrania danych.
2. Robot wystawia sygnał „DO[121]” – gotowość do pobrania danych.
3. PLC przygotowuje dane dla robota na zmiennej „GI[1]”.
4. PLC wystawia sygnał, że dane są gotowe do pobrania ustawiając „DI[121]” na „1” i czeka na sygnał, że robot pobrał dane (skasowanie sygnału „DO[121]”).
5. Po odczytaniu danych robot wystawia pobrane dane na zmiennej „GO[1]” (jako echo) i kasuje sygnał gotowości do pobrania danych „DO[121]” co oznacza, że dane zostały pobrane przez robota.
6. PLC porównuje dane na GO[1] (echo danych wysłanych do robota) z danymi, które wysłał (powinny być jednakowe) i kasuje sygnał DI[121] - dane gotowe do pobrania.

Przebieg komunikacji na wykresie „Y(t)”:



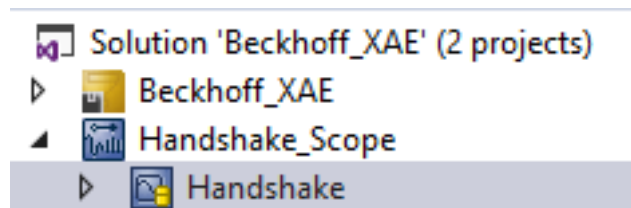
Wykres górny:

- a) Pomarańczowy – sygnał rozpoczęcia zautomatyzowanej procedury przesyłu danych do robota.
- b) Niebieski – DO[121], sygnał od robota dla PLC, robot gotowy na przyjęcie danych.
- c) Zielony – DI[121], sygnał od PLC dla robota, dane gotowe do pobrania przez robota.

Wykres dolny:

- a) Zielony – GI[1], wartość parametru przygotowanego w PLC do przesłania do robota.
- b) Niebieski – GO[1], echo od robota, ostatnia wartość jaką robot pobrał z PLC.

Przedstawiony wykres można otworzyć i dokładnie obejrzeć w środowisku TwinCAT:



Mapa pamięci

Mapa pamięci przydatna w realizacji zadania (operator „>” służy odwołaniu do konkretnego bitu, operator „[i]” służy odwołaniu do i-tego elementu tablicy):

arrDataForRobot_00[0].0 -> IMSTP
arrDataForRobot_00[0].1 -> Hold
arrDataForRobot_00[0].2 -> SFSPD
arrDataForRobot_00[0].7 -> Enable
arrDataForRobot_00[1].1 -> ProdStart
arrDataForRobot_00[0].4 -> FaultReset

arrDataFromRobot_00[0].2 -> PrgRunning
arrDataFromRobot_00[0].5 -> Falut

Zmienne potrzebne do sterowania przesyłaniem danych są umieszczone na następujących pozycjach:

//Dane od robota – nazwy przykładowe.
arrDataFromRobot_00[1].4 -> DO[121] -> Robot gotowy do pobrania danych.
arrDataFromRobot_00[2] -> GO[1] -> Wartość odebrana przez robota.

//Dane dla robota – nazwy przykładowe.
arrDataForRobot_00[1].3 -> DI[121] -> Dane gotowe do pobrania przez robota.
arrDataForRobot_00[2] -> GI[1] -> Wartość przygotowana przez PLC dla robota.

Realizacja zadania – etap 4

W tym etapie należy tak zmodyfikować procedurę przesyłania parametrów paletyzacji aby na żądanie programisty wszystkie parametry zostały automatycznie przesłane do robota.

Aby to zrealizować można posłużyć się jednowymiarową tablicą, w której na kolejnych pozycjach, w odpowiedniej kolejności, będą przechowywane wartości parametrów, które należy przesłać do robota.

W celu uzyskania dostępu do kolejnych elementów tablicy należy skorzystać ze zmiennej, która będzie służyć jako indeks tablicy.

Przykładowy kod realizujący dostęp do pojedynczych elementów tablicy:

```
//Zmienne do zadeklarowania:  
arrParametryPaletyzacji: ARRAY [0..7] OF UINT;  
iIndex: INT;
```

```
//Przykładowy kod realizujący przesuwanie się po elementach tablicy.  
FOR iIndex := 0 TO 7 DO  
    arrParametryPaletyzacji[iIndex];  
END_FOR
```

Do zautomatyzowania przesyłania danych, można wykorzystać instrukcję „CASE”.

Składnia tej instrukcji jest następująca ({} - wyrażenie opcjonalne, <krok> - zmienna typu całkowitego, np. INT) :

```
CASE <krok> OF  
    <krok_1>:  
        <Instrukcje do wykonania w kroku pierwszym>  
    <krok_2>:  
        <Instrukcje do wykonania w kroku drugim>  
    ...  
    <krok_N>:  
        <Instrukcje do wykonania w kroku N-tym>  
    {ELSE  
        <<Instrukcje do wykonania jeśli numer kroku nie występuje na liście>  
    }  
END_CASE;
```

Przykład użycia instrukcji „CASE”:

```
//Zmienne do zadeklarowania:  
iStep: INT;  
iVar1: INT;  
iVar2: INT;
```

```
//Przykładowy kod wykorzystujący instrukcję CASE.
```

```
CASE iStep OF  
  10:  
    iVar1 := 10;  
    iStep := iStep + 10;  
  
  20:  
    iVar2 := 20;  
    iStep := iStep + 10;  
  
  30:  
    IF iVar1 = 0 OR iVar2 = 0 THEN  
      iStep := 10;  
    ELSE  
      iStep := iStep + 10;  
    END_IF  
  
  40:  
    iStep := 0;  
  
ELSE  
  iStep = 10;  
  
END_CASE;
```

System oceniania

1. Osiągnięcie celu zadania – 1 pkt

Zadanie polega na uruchomieniu komunikacji z Robotem FANUC i przesłaniu do niego danych parametryzujących proces paletyzacji.

2. Zaliczenia etapów – łącznie 4 pkt

- a) Uruchomienie komunikacji z robotem po protokole EtherCAT (1 pkt.)
- b) Przesłanie do robota sygnałów pozwolenia na pracę. (1 pkt.)
- c) Zaprogramowanie procedury handshake'u pomiędzy PLC a robotem i przesłanie, co najmniej, jednego z parametrów paletyzacji. (1 pkt.)
- d) Zautomatyzowanie procedury przesyłania parametrów, tak aby na żądanie programisty wszystkie parametry zostały kolejno wysłane do robota. (1 pkt.)

3. Czas wykonania po zapoznaniu się z instrukcją – maksymalnie 3 pkt

- a) Ukończenie zadania poniżej 30 min (3 pkt)
- b) Ukończenie zadania poniżej 35 min (2 pkt)
- c) Ukończenie zadania poniżej 40 min (1 pkt)