

## Ćwiczenie G.

Ćwiczenie przeprowadzane jest w cyklu dwutygodniowym.

W pierwszym tygodniu studenci powinni zapoznać się z metodą programowania z wykorzystaniem panelu uczenia i ułożyć program przenoszenia detali do pojemników.

W drugim tygodniu studenci poznają metodę programowania z wykorzystaniem programu zewnętrznego. W tej części ćwiczenia należy stworzyć program, w którym akcje robota wywoływane są przez użytkownika na żądanie poprzez podanie numeru podzadania na panelu uczenia.

### 1. Opis programowania robota z wykorzystaniem panelu uczenia (tydzień 1)

#### Robot

Robot THL500 jest robotem typu SCARA, o stosunkowo prostej strukturze kinematycznej, przez co nie pozwala na zmianę orientacji chwyconego detalu w płaszczyźnie innej niż płaszczyzna XY.

#### Ogólny opis

W tekście poniżej listowane odwołania dotyczą pozycji Instruction Manual - **Operator's Manual** udostępnionego studentom w wersji elektronicznej dla ćwiczenia G.

Przed przystąpieniem do programowania należy zaznajomić się z panelem uczenia oraz płytą sterownika i jej przyciskami i określić ogólne znaczenie przycisków na podstawie instrukcji fabrycznej.

#### Wskazówki

Pierwsze uruchomienie układu powinno nastąpić przy załączonym przełączniku **Teaching** na płycie sterownika. W tym trybie przejść do trybu edycji programu (EDIT).

Stworzyć nazwę swojego pliku zaczynając się od przedrostka AIR, następnie wpisać skrót dnia zajęć ( np. pon) i uzupełnić cyframi – każda grupa ma do dyspozycji 19 liczb od  $20*(n-1)+1$  ...  $20*(n-1)+19$  (czyli np. od 1 do 19 dla  $n=1$ , gdzie  $n$  jest numerem grupy).

Edycję przeprowadzić z wykorzystaniem klawiatury znakowej. Na początku wpisać program pusty składający się z trzech linii tekstowych i jednej pustej

Np. **PROGRAM AIRPON 01**

**SPEED = 5** ‘ w programie proszę używać małych prędkości

#### END

Zapisać komendą SAVE. Program powinien się pojawić w spisie programów.

#### **Zabrania się edytowania innych niż własne pliki.**

Wybrać ponownie swój program i przystąpić do edycji właściwego programu.

Ogólnie fazy programowania i uruchomienia przedstawiają się następująco:

1. Edycja programu. ( EDIT)
2. Edycja danych pozycyjnych (nie musi wystąpić) (DEDIT)
3. Uruchomienie (RUN)

Struktura programu składa się z wykonywanych instrukcji ruchu (z ewentualnym wykorzystaniem danych pozycyjnych) oraz akcji efektora. Obecnie efekтором jest przysawka, której załączenie następuje po rozkazie **DOUT(1)** , wyłączenie **DOUT(-1)**.

Należy pamiętać o tym, że instrukcje *typu ruchu* i *dane pozycyjne* są od siebie niezależne, a więc istnieje możliwość zmiany jednego zbioru bez zmiany drugiego.

#### Wybrane wskazówki edycyjne ( tryb EDIT)

1. Rolę klawisza *Enter* pełni klawisz oznaczony na panelu uczenia jako EXE (1-8)
2. ESC pozwala na wyjście z danego trybu. (1-8)
3. Klawisze F1, F2, F3, F4, F5 skorelowane są z wyświetlanymi komendami w menu(1-8)
4. BS kasuje znaki. (1-8)
5. NEXT pozwala przejść do dalszej części menu (1-8)
6. Komenda z menu COPY pozwala wygodnie kopiować rozkazy o podobnej strukturze, które potem wymagają zamiany tylko pojedynczych parametrów (4-14)
7. Klawisz INS zmienia tryb wstawiania (4-6)

Po napisaniu programu zapisać go komendą **SAVE** lub wyjść bez zapisywania **NOSAVE**.

#### Wybrane wskazówki edycyjne dla trybu definiowania punktów (tryb DEDIT).

W trybie tym edytujemy poszczególne pozycje oznaczając je symbolicznie np. POZ1, POZ2, itd. dla danego pliku programu, który został wybrany.

Do wprowadzania nowych pozycji używamy klawisza **INS**. Znaczenie **Klawiszy** i **Komend** pokazane jest na stronie 5-5 i 5-6.

Można definiować pozycje wykorzystując komendę **TEACH** (5-27), która wprowadza dla aktualnie podświetlonej nazwy aktualną pozycję robota lub edytować pozycję ręcznie. Zaleca się zbadanie przestrzeni roboczej poprzez doprowadzenie robota do punktów w pobliżu jej końca i próbne wpisanie współrzędnych. Kasowanie i wstawianie opisane jest (5-31 – 5-34)

Po stworzeniu odpowiedniego zbioru punktów zapamiętać **SAVE**.

#### Obsługa błędów:

Jeśli przy zapisywaniu pojawi się błąd np. składniowy w programie (sygnalizacja dźwiękowa) – sprawdzić, jakiego rodzaju to błąd (F3-**ERROR**) i skasować (F3-**RESETEM**) (sygnalizacja dźwiękowa powinna się wyłączyć)

#### Uruchomienie programu:

1. Przełączyć przełącznik na płycie sterownika z trybu **Teaching** na **Internal** ( 6-2).
2. Wybrać napisany swój program korzystając z komendy **SEL**. ( EXE zatwierdza wybór)
3. Wybrać odpowiedni tryb pracy za pomocą komendy **MODE**. Mamy do wyboru następujące tryby **CONT**, **CYCLE**, **SEG**, **STEP** (6-6).
4. Załączyć **SERVO ON** (na płycie sterownika) , **RUN** uruchamia program **STOP** zatrzymuje (uwaga, producent **nie** zaleca używać do zatrzymywania przycisku **SERVO OFF** !)
5. W razie niebezpieczeństwa użyć **EMERGENCY stop**, ale należy mieć na uwadze, iż program powinien być tak napisany, aby nie było konieczności używania tego przycisku.
6. W razie pojawienia się błędu zobaczyć, jakiego jest rodzaju i ewentualnie zresetować.

W czasie wykonywania programu program można zatrzymać używając przycisku **FEED HOLD** z panelu uczenia - robot zatrzymuje się bezzwłocznie. Po powtórnym naciśnięciu **FEED HOLD** robot kontynuuje rozpoczęty ruch – tak więc przycisku tego używamy do *tymczasowego* zatrzymania robota.

Inną możliwością zatrzymania robota jest użycie z menu **STOP** – w takim wypadku robot jednak nie zatrzymuje się bezzwłocznie, ale dokończy rozpoczęty krok.

Bezzwłoczne przerwanie programu uzyskuje się natomiast poprzez **BREAK**.

W czasie ruchu podgląd wykonywanych linii programu uzyskujemy poprzez **LDISP** (z dalszej części menu).

Wykonanie pojedynczego rozkazu możliwe jest również za pomocą pozycji menu **DO**.

W tym trybie możliwe do wykorzystania są rozkazy wysyłania sygnału na wyjście **DOUT** (np. do sterowania przyssawką) lub **MOVEA** (ruch do absolutnej pozycji złączowej).

### Program zajęć:

1. Zbadać poruszając ręcznie w trybie **Teaching** zakresy ruchów robota obserwując czy nie zachodzi niebezpieczeństwo zderzenia efektora z przeszkodą zewnętrzną.
2. Stworzyć program demonstrujący ruchy o, w przybliżeniu, +/- 10 jednostek (stopnie lub mm) w każdym złączu.  
Wykorzystać rozkaz **MOVEI** (znaczenie rozkazu sprawdzić w opisie języka - **Robot Language Manual**)
3. Dla 3 płytek rozłożonych na taśmociągu ułożyć je w stos, a następnie rozłożyć w odwrotnej kolejności oraz przełożyć do drugiego pojemnika.  
Wykorzystać edytor punktów w trybie uczenia (tryb **DEDIT**) i rozkazy typu **MOVE**, ewentualnie **MOVEJ**.

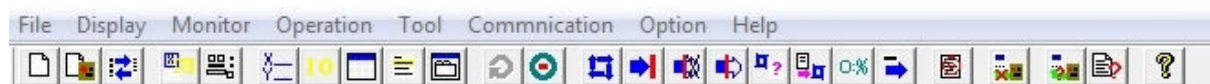
#### *Zadanie dodatkowe:*

4. Chwycić płytkę z narysowaną strzałką i trzymając ją z odpowiednią orientacją objechać (w granicach możliwości) zadany obiekt w taki sposób, aby wektor określony przez narysowaną strzałkę był zawsze prostopadły do krawędzi obiektu leżącego płasko. Ruch wykonywać na niewielkiej wysokości (do 5 mm) od krawędzi obiektu obrysowywanego.

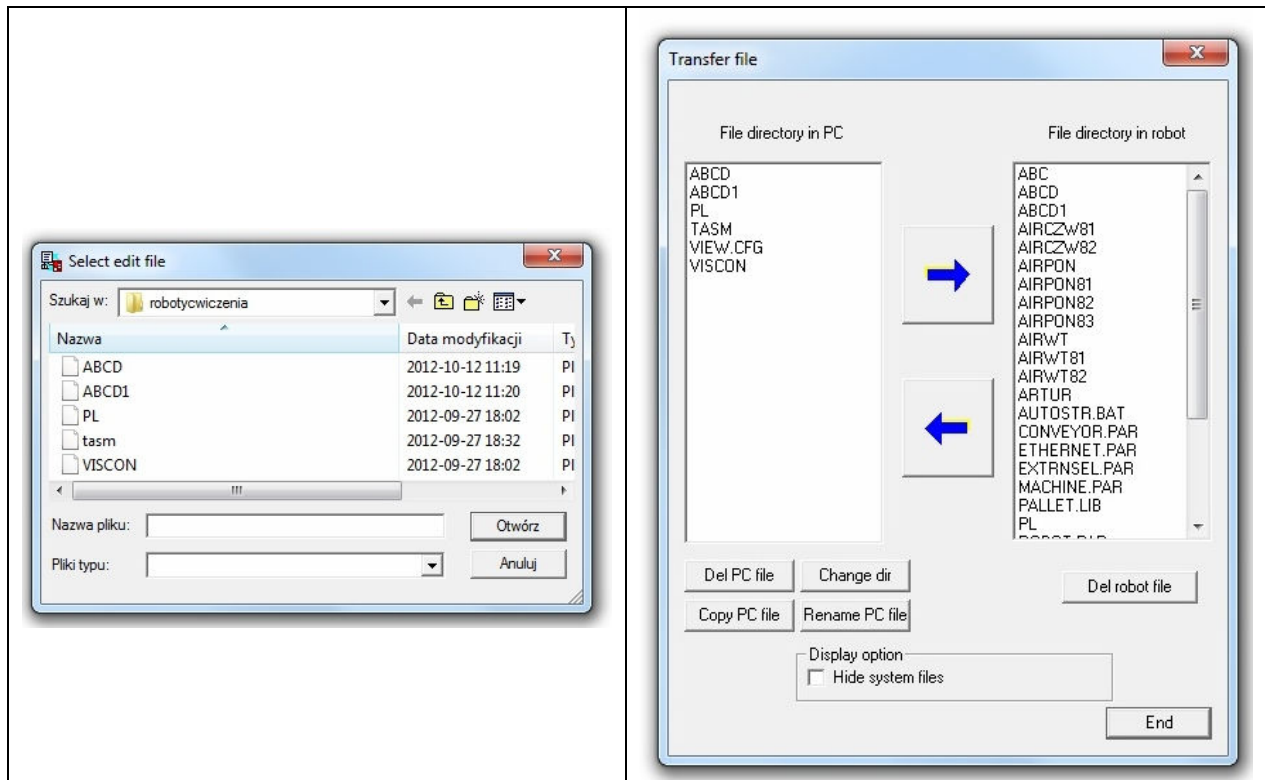
## 2. Obsługa programu TSPC ( tydzień 2)

Program TSPC jest kompleksowym środowiskiem do programowania robota odpowiadającym funkcjonalnie programowaniu z wykorzystaniem panelu uczenia. Aby możliwe było wykorzystanie programu przełącznik na płycie sterownika powinien być w pozycji **External**. Podstawowe menu programu zaprezentowane jest poniżej.

Pierwszą czynnością operatora winno być połączenie się ze sterownikiem z menu **Communication**.

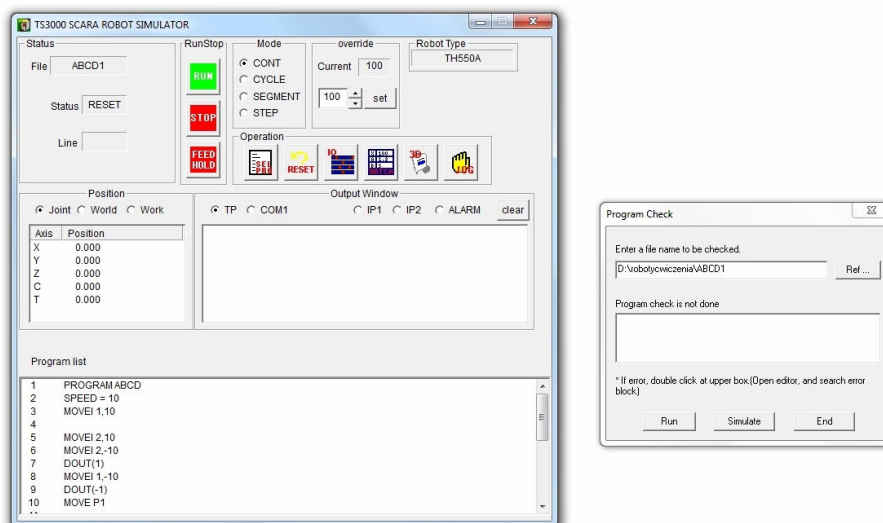


Menu **File** pozwala wybrać plik do edycji (**Edit**) jak również ładować pliki z/do pamięci sterownika (**File Mode**).



Menu **Operation** umożliwia wybór programu do wykonania (**Select**), komenda **Direct run** z tego menu umożliwia wykonanie pojedynczego rozkazu (co można wykorzystać np. do przemieszczenia robota w określone miejsce czy wyłączenia przysawki).

Po napisaniu programu można za pomocą symulatora sprawdzić działanie programu (**Tool\Program Check**) i przesłać go do sterownika.

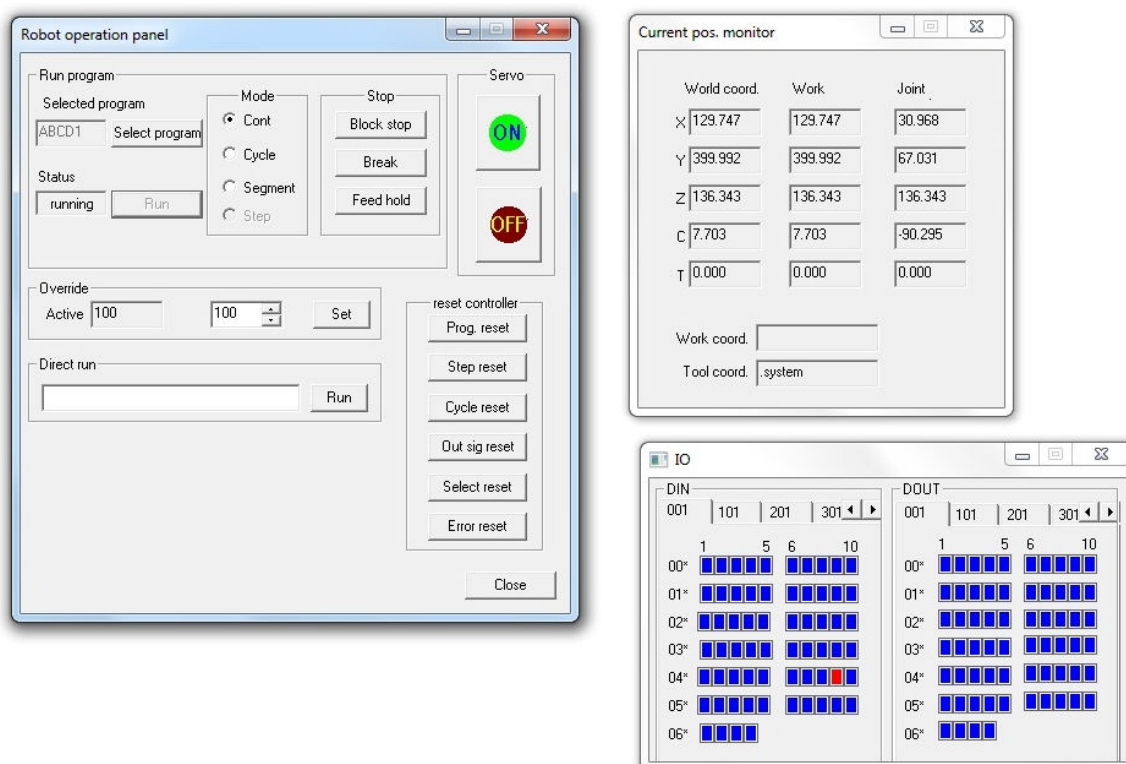


Jeżeli program jest poprawny poprzez menu **Display/Operation Panel** można uaktywnić Panel Operacyjny Robota. Pierwszym krokiem do uruchomienia jest załączenie **SERVO ON**. Jeżeli operacja została przeprowadzona pomyślnie przycisk **Run** umożliwia start programu w jednym z trybów pracy – ciągłym (w pętli zamkniętej), z pojedynczym wykonaniem programu, krokowym.

Zatrzymanie robota odbywa się za pomocą podobnych komend jak na panelu:

- **feed hold** - zatrzymanie tymczasowe
- **break** – przerwanie natychmiastowe
- **block stop**- przerwanie z dokończeniem kroku

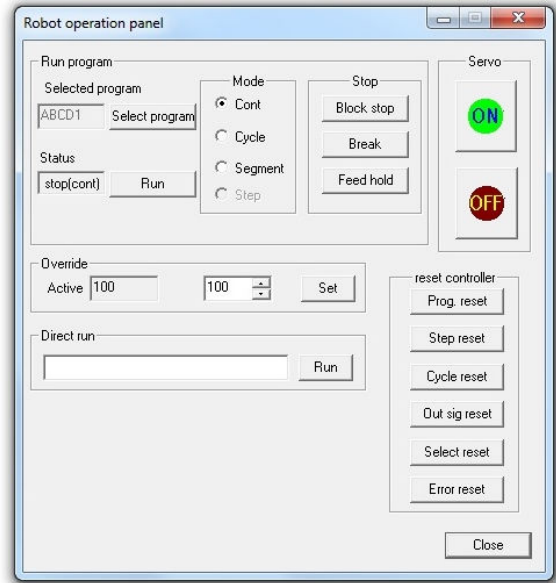
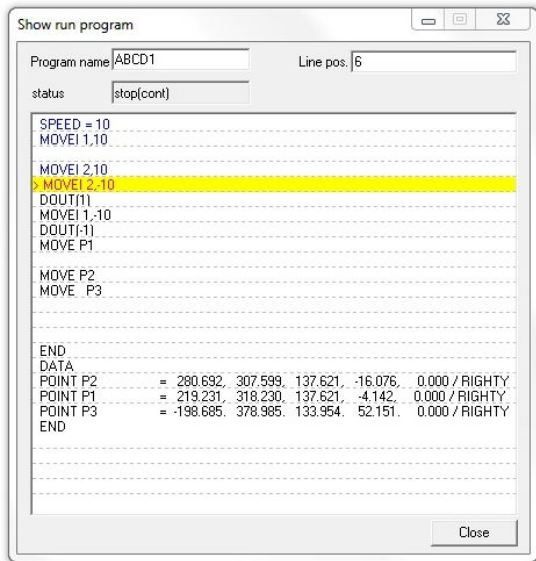
**Nie należy** używać wyłączenia Serva do wyłączenia działającego robota.



Dodatkowo w czasie pracy można wykorzystując **Monitor** podglądać aktualne wartości pozycji oraz stany wejść /wyjść (przy włączonym **Update data**)

Możliwe jest również monitorowanie bieżącej linii programu (wywoływane również przez Ctrl+A)

## Cwiczenie G



Po zakończeniu pracy z robotem należy wyłączyć **SERVO** oraz rozłączyć komunikację ze sterownikiem.

### Program zajęć:

Wykorzystując program stworzony na poprzednich zajęciach rozwinąć go do wersji, w której na żądanie operatora wykonywane są następujące akcje:

1. Przenoszenie elementów do pojemnika lewego
2. Przenoszenie elementów do pojemnika prawego
3. Przenoszenie elementów z pojemnika pełnego do pustego
4. Przenoszenie elementów z pojemnika pełnego na taśmociąg.

Do stworzenia programu wykorzystać następujące konstrukcje programowe:

```
Global
a=0 ‘ deklaracja zmiennej umieszczana przed programem
End
...
Input tp, a ‘ odczytywanie numeru akcji z panelu
Print tp, „komunikat” ‘ wyprowadzenie komunikatu o akcji na panel
...
If a==1 then goto etykieta1 ‘skok warunkowy do miejsca określonego przez etykietę
...
etykieta1:
move p1
delay 1
...
```