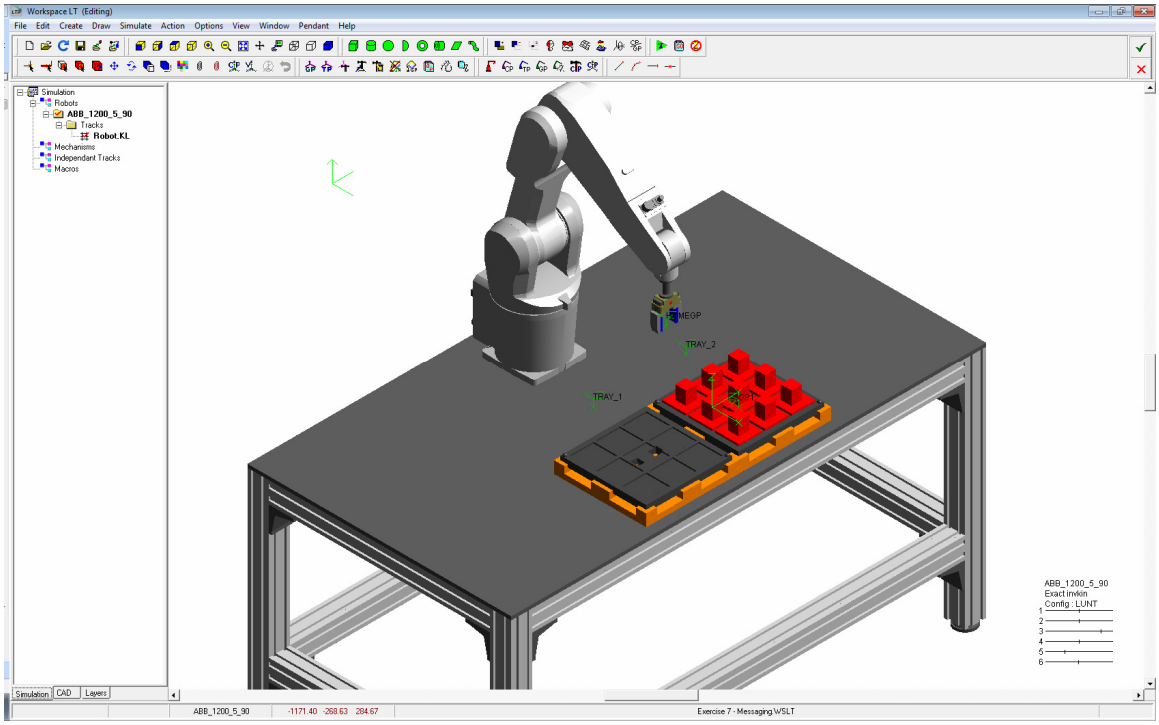


Exercise 7 - Messaging Via Console

Exercise 7 - Messaging Via Console

Objectives:

- Create basic robot programs in simulation.
- Using an end of arm gripper to manipulate parts.
- Using routines to assign various robot tasks.
- Integrate a robot, staging fixture and 2 pallets in simulation.
- Using the console to communicate messages and receiving instructions to control the simulation flow.

Materials;

- Workspace LT© simulation software.
- Workspace LT© project file "Exercise 7 - Messaging via Console.WSLT".
- Manual "Workspace LT© User Guide.pdf".

Helpful Hint;

Before starting this exercise, review previous exercises and the User Guide sections;

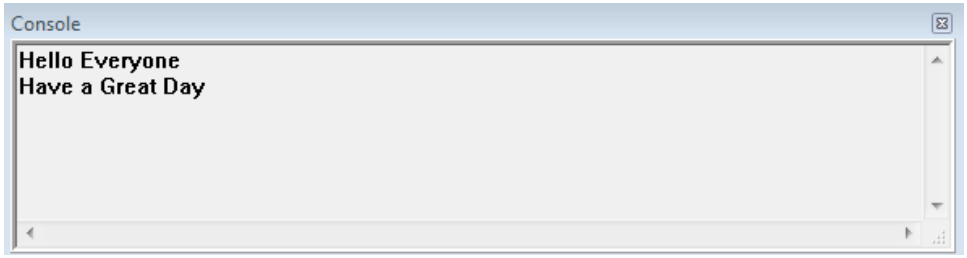
- 6.3 - Comments and Workspace commands
- 6.4.5 Variables
- 6.7.1 - INTEGER and REAL functions
- 8.1 - Communicating with the console
- 9.1.1 - IF ... ENDIF

1) **Procedure:** Create a Robot track to include Write and Read commands.
   a) Open Workspace LT simulation software.
   b) Open the project file "Exercise 7 - Messaging via Console.WSLT ".
   c) Add a track for robot ABB_1200_5_90. Enter the Track Name "Track01" and select the Language "KAREL 2".
   d) Edit the robot track by inserting the following syntax;

```
PROGRAM Track01
--  Workspace LT KAREL 2 Program for ABB_1200_5_90 Robot
BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  $MOTYPE=JOINT
  $TERMTYPE=FINE
  $UTOOL = POS(0,0,134,-90,-90,0,'')
  WRITE ('Hello Everyone',CR)
  WRITE ('Have a Great Day',CR)
```

END Track01

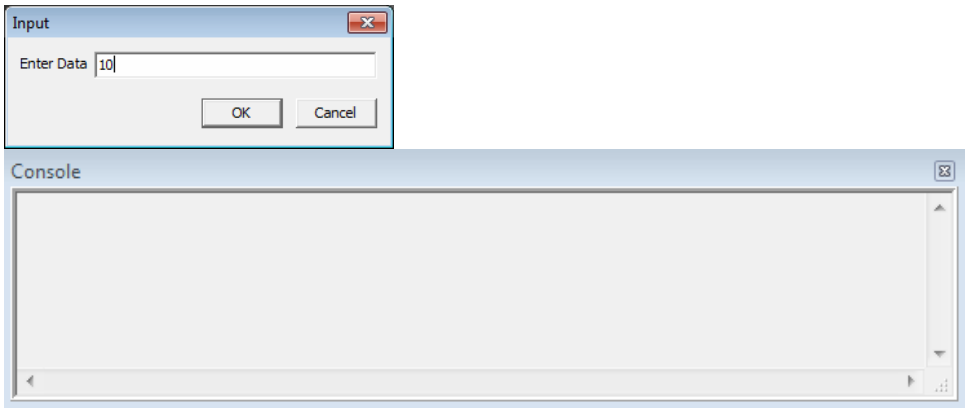
e) Open the console window. Using the mouse, select Window found on the tool bar. A drop down menu will appear, Using the mouse, select Console Window. the console window will appear, lower left of the modeling window.
f) Verify the tacks Track01.KL is active.
g) Using the mouse select (Play simulation) and observe the simulation.



h) Edit the robot track by inserting the additional syntax;

```
PROGRAM Track01
--  Workspace LT KAREL 2 Program for ABB_1200_5_90 Robot

VAR
  Rating : INTEGER

BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  $MOTYPE=JOINT
  $TERMTYPE=FINE
  $UTOOL = POS(0,0,134,-90,-90,0,'')
  DELAY 500
  WRITE ('Hello Everyone', CR)
  WRITE ('Have a Great Day', CR)
  WRITE ('How would you rate these exercises?', CR )
  WRITE ('Enter 1 for a low rating and up to 10 for a higher rating',)
  READ (Rating)
  WRITE (CR)
  WRITE ('You have given a rating of ' , Rating, ' Thank you!', CR)
END Robot
```

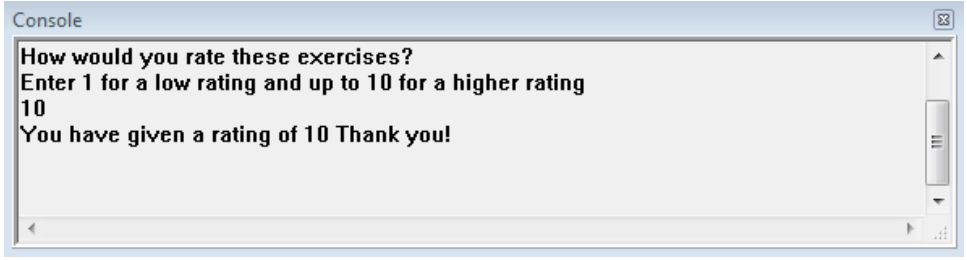
i) Play simulation. Interact and observe the simulation.



j) Add a delay to allow the scan off the consol to "catch up" to the scan of the track file. "Delay 500".

```
PROGRAM Track01
--  Workspace LT KAREL 2 Program for ABB_1200_5_90 Robot

VAR
  Rating : INTEGER

BEGIN
 $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  $MOTYPE=JOINT
  $TERMTYPE=FINE
  $UTOOL = POS(0,0,134,-90,-90,0,'')
  DELAY 500
  WRITE ('Hello Everyone', CR)
  WRITE ('Have a Great Day', CR)
  WRITE ('How would you rate these exercises?', CR )
  WRITE ('Enter 1 for a low rating and up to 10 for a higher rating',)
  READ (Rating)
  WRITE (CR)
  WRITE ('You have given a rating of ' , Rating, ' Thank you!', CR)
END Track01
```

k) Play simulation. Interact and observe the simulation.



2) **Procedure:** Create a Robot track to include Write and Read commands and add initial "error proofing".
   a) Add a track for robot ABB_1200_5_90. Enter the Track Name "Track02" and select the Language "KAREL 2".
   b) Edit the robot track by inserting the following syntax;

```
PROGRAM Track02
--  Workspace LT KAREL 2 Program for ABB_1200_5_90 Robot

VAR
  ROW : INTEGER
  COLUMN : INTEGER

Routine GetRowNo
BEGIN
  WRITE ('Enter Row Number 1, 2 OR 3')
  READ (ROW)
  WRITE (CR)
End GetRowNo

Routine GetColumnNo
BEGIN
  WRITE ('Enter Column Number 1, 2 OR 3')
  READ (COLUMN)
  WRITE (CR)
End GetColumnNo

Routine RetrievePart
BEGIN
  MOVE TO TRAY_2
  -- Robot move to pounce position above the column row nest,
  -- The spacing between each column and row is 100mm
  -- () must be used when calculating a position.
  $MOTYPE=LINEAR
  MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
  MOVE RELATIVE VEC(0,0,-150)
  MOVE AWAY 150.00
END RetrievePart

Routine PlacePart
BEGIN
  MOVE TO TRAY_1
  -- Robot move to pounce position above the column row nest,
  -- The spacing between each column and row is 100mm
  -- () must be used when calculating a position.
```

```
  $MOTYPE=LINEAR
  MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
  MOVE RELATIVE VEC(0,0,-150)
  MOVE AWAY 150.00
END PlacePart

-- ************  Main Program  ************

BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  $MOTYPE=JOINT
  $TERMTYPE=FINE
  $UTOOL = POS(0,0,134,-90,-90,0,'')
  MOVE TO HOMEGP
  GetColumnNo
  GetRowNo
  RetrievePart
  PlacePart
END Track02
```

c) Play simulation to test the track syntax. Interact and observe the simulation.
d) Edit the robot track to include gripping and moving the parts from TRAY_2 to TRAY_1 by inserting the following syntax;

```
PROGRAM Track02
--  Workspace LT KAREL 2 Program for ABB_1200_5_90 Robot

VAR
  ROW : INTEGER
  COLUMN : INTEGER
  TRAY : INTEGER

Routine GetRowNo
BEGIN
    WRITE ('Enter Row Number 1, 2 OR 3')
    READ (ROW)
    WRITE (CR)
End GetRowNo

Routine GetColumnNo
BEGIN
    WRITE ('Enter Column Number 1, 2 OR 3')
    READ (COLUMN)
    WRITE (CR)
End GetColumnNo
```

```
Routine RetrievePart
BEGIN
  MOVE TO TRAY_2
  -- Robot move to pounce position above the column row nest,
  -- The spacing between each column and row is 100mm
  -- () must be used when calculating a position.
  $MOTYPE=LINEAR
  MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
  MOVE RELATIVE VEC(0,0,-150)
  DELAY 1000
  IF ROW=1 THEN
    IF COLUMN=1 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C1R1'
    ENDIF
    IF COLUMN=2 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C2R1'
    ENDIF
    IF COLUMN=3 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C3R1'
    ENDIF
  ENDIF
  IF ROW=2 THEN
    IF COLUMN=1 THEN
      CLOSE HAND 1
      --! GRASPOBJ 'PART_C1R2'
    ENDIF
    IF COLUMN=2 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C2R2'
    ENDIF
    IF COLUMN=3 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C3R2'
    ENDIF
  ENDIF
  IF ROW=3 THEN
    IF COLUMN=1 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C1R3'
    ENDIF
    IF COLUMN=2 THEN
      CLOSE HAND 1
```

```
      -- ! GRASPOBJ 'PART_C2R3'
    ENDIF
    IF COLUMN=3 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C3R3'
    ENDIF
  ENDIF
  MOVE AWAY 150.00
END RetrievePart

Routine PlacePart
BEGIN
  MOVE TO TRAY_1
  -- Robot move to pounce position above the column row nest,
  -- The spacing between each column and row is 100mm
  -- () must be used when calculating a position.
  $MOTYPE=LINEAR
  MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
  MOVE RELATIVE VEC(0,0,-150)
  OPEN HAND 1
  MOVE AWAY 150.00
END PlacePart

-- ************  Main Program  ************

BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  $MOTYPE=JOINT
  $TERMTYPE=FINE
  $UTOOL = POS(0,0,134,-90,-90,0,'')

REPEAT
  -- ROBOT MOVE HOME POSITION
  MOVE TO HOMEGP
  DELAY 100
  WRITE ('ENTER OCCUPIED NEST IN TRAY 2',CR)
  GetColumnNo
  GetRowNo
  RetrievePart
  PlacePart
UNTIL FALSE
END Track02
```

e) Play simulation. Interact and observe the simulation. Transfer all parts from "TRAY_2" to "TRAY_1". Reload the simulation each time you play the simulation

f) Edit the robot track to include "error proofing" when inputting the column number and row number by inserting the following syntax;

```
PROGRAM Track02
--  Workspace LT KAREL 2 Program for ABB_1200_5_90 Robot

VAR
  ROW : INTEGER
  COLUMN : INTEGER
  TRAY : INTEGER
  CORRECTVALUE : BOOLEAN

Routine GetRowNo
BEGIN
  CORRECTVALUE=FALSE
  REPEAT
    WRITE ('Enter Row Number 1, 2 OR 3')
    READ (ROW)
    WRITE (CR)
    IF ROW=1 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF ROW=2 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF ROW=3 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF CORRECTVALUE=FALSE THEN
      WRITE ('Incorrect value Please Re-Enter',CR)
    ENDIF
UNTIL CORRECTVALUE=TRUE
End GetRowNo

Routine GetColumnNo
BEGIN
  CORRECTVALUE=FALSE
  REPEAT
    WRITE ('Enter Column Number 1, 2 OR 3')
    READ (COLUMN)
    WRITE (CR)
    IF COLUMN=1 THEN
      CORRECTVALUE=TRUE
```

```
    ENDIF
    IF COLUMN=2 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF COLUMN=3 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF CORRECTVALUE=FALSE THEN
      WRITE ('Incorrect value Please Re-Enter',CR)
    ENDIF
UNTIL CORRECTVALUE=TRUE
End GetColumnNo

Routine RetrievePart
BEGIN
  MOVE TO TRAY_2
  -- Robot move to pounce position above the column row nest,
  -- The spacing between each column and row is 100mm
  -- () must be used when calculating a position.
  $MOTYPE=LINEAR
  MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
  MOVE RELATIVE VEC(0,0,-150)
  DELAY 1000
  IF ROW=1 THEN
    IF COLUMN=1 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C1R1'
    ENDIF
    IF COLUMN=2 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C2R1'
    ENDIF
    IF COLUMN=3 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C3R1'
    ENDIF
  ENDIF
  IF ROW=2 THEN
    IF COLUMN=1 THEN
      CLOSE HAND 1
      --! GRASPOBJ 'PART_C1R2'
    ENDIF
    IF COLUMN=2 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C2R2'
    ENDIF
```

```
    IF COLUMN=3 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C3R2'
    ENDIF
  ENDIF
  IF ROW=3 THEN
    IF COLUMN=1 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C1R3'
    ENDIF
    IF COLUMN=2 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C2R3'
    ENDIF
    IF COLUMN=3 THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C3R3'
    ENDIF
  ENDIF
  MOVE AWAY 150.00
END RetrievePart

Routine PlacePart
BEGIN
  MOVE TO TRAY_1
  -- Robot move to pounce position above the column row nest,
  -- The spacing between each column and row is 100mm
  -- () must be used when calculating a position.
  $MOTYPE=LINEAR
  MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
  MOVE RELATIVE VEC(0,0,-150)
  OPEN HAND 1
  MOVE AWAY 150.00
END PlacePart

-- ************  Main Program  ************

BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  $MOTYPE=JOINT
  $TERMTYPE=FINE
  $UTOOL = POS(0,0,134,-90,-90,0,'')

REPEAT
  -- ROBOT MOVE HOME POSITION
```

```
    MOVE TO HOMEGP
    DELAY 100
    WRITE ('ENTER OCCUPIED NEST IN TRAY 2',CR)
    GetColumnNo
    GetRowNo
    RetrievePart
    PlacePart
  UNTIL FALSE
  END Track02
```

g) Play simulation. Interact and observe the simulation. Transfer all parts from "TRAY_2" to "TRAY_1". Test the "error proofing" when inputting the column number and row number. Reload the simulation each time you play the simulation

3) **Procedure:** Create a Robot track to include total error proofing and setup the track to loop a total of nine times.
   a) Add a track for robot ABB_1200_5_90. Enter the Track Name "Robot.kl" and select the Language "KAREL 2".
   b) Edit the robot track by inserting the following syntax;

```
PROGRAM Robot
--  Workspace LT KAREL 2 Program for ABB_1200_5_90 Robot

CONST
  PartsTotal=9

VAR
  ROW : INTEGER
  COLUMN : INTEGER
  TRAY : INTEGER
  PartsCount : INTEGER
  CORRECTVALUE : BOOLEAN

  NEST_C1R1 : BOOLEAN
  NEST_C2R1 : BOOLEAN
  NEST_C3R1 : BOOLEAN
  NEST_C1R2 : BOOLEAN
  NEST_C2R2 : BOOLEAN
  NEST_C3R2 : BOOLEAN
  NEST_C1R3 : BOOLEAN
  NEST_C2R3 : BOOLEAN
  NEST_C3R3 : BOOLEAN


Routine GetRowNo
```

```
BEGIN
  CORRECTVALUE=FALSE
  REPEAT
    WRITE ('Enter Row Number 1, 2 OR 3')
    READ (ROW)
    WRITE (CR)
    IF ROW=1 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF ROW=2 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF ROW=3 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF CORRECTVALUE=FALSE THEN
      WRITE ('Incorrect value Please Re-Enter',CR)
    ENDIF
UNTIL CORRECTVALUE=TRUE
End GetRowNo

Routine GetColumnNo
BEGIN
  CORRECTVALUE=FALSE
  REPEAT
    WRITE ('Enter Column Number 1, 2 OR 3')
    READ (COLUMN)
    WRITE (CR)
    IF COLUMN=1 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF COLUMN=2 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF COLUMN=3 THEN
      CORRECTVALUE=TRUE
    ENDIF
    IF CORRECTVALUE=FALSE THEN
      WRITE ('Incorrect value Please Re-Enter',CR)
    ENDIF
UNTIL CORRECTVALUE=TRUE
End GetColumnNo

Routine RetrievePart
BEGIN
  MOVE TO TRAY_2
```

```
-- Robot move to pounce position above the column row nest,
-- The spacing between each column and row is 100mm
-- () must be used when calculating a position.
$MOTYPE=LINEAR
MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
MOVE RELATIVE VEC(0,0,-150)
DELAY 1000
IF ROW=1 THEN
  IF COLUMN=1 THEN
    IF NEST_C1R1=TRUE THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C1R1'
      NEST_C1R1=FALSE
    ENDIF
  ENDIF
  IF COLUMN=2 THEN
    IF NEST_C2R1=TRUE THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C2R1'
      NEST_C2R1=FALSE
    ENDIF
  ENDIF
  IF COLUMN=3 THEN
    IF NEST_C3R1=TRUE THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C3R1'
      NEST_C3R1=FALSE
    ENDIF
  ENDIF
ENDIF
IF ROW=2 THEN
  IF COLUMN=1 THEN
    IF NEST_C1R2=TRUE THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C1R2'
      NEST_C1R2=FALSE
    ENDIF
  ENDIF
  IF COLUMN=2 THEN
    IF NEST_C2R2=TRUE THEN
      CLOSE HAND 1
      -- ! GRASPOBJ 'PART_C2R2'
      NEST_C2R2=FALSE
    ENDIF
  ENDIF
  IF COLUMN=3 THEN
```

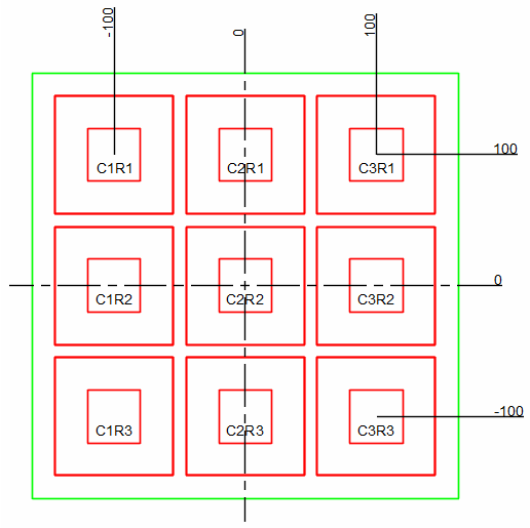
```
        IF NEST_C3R2=TRUE THEN
          CLOSE HAND 1
          -- ! GRASPOBJ 'PART_C3R2'
          NEST_C3R2=FALSE
        ENDIF
      ENDIF
    ENDIF
    IF ROW=3 THEN
      IF COLUMN=1 THEN
        IF NEST_C1R3=TRUE THEN
          CLOSE HAND 1
          -- ! GRASPOBJ 'PART_C1R3'
          NEST_C1R3=FALSE
        ENDIF
      ENDIF
      IF COLUMN=2 THEN
        IF NEST_C2R3=TRUE THEN
          CLOSE HAND 1
          -- ! GRASPOBJ 'PART_C2R3'
          NEST_C2R3=FALSE
        ENDIF
      ENDIF
      IF COLUMN=3 THEN
        IF NEST_C3R3=TRUE THEN
          CLOSE HAND 1
          -- ! GRASPOBJ 'PART_C3R3'
          NEST_C3R3=FALSE
        ENDIF
      ENDIF
    ENDIF
    MOVE AWAY 150.00
END RetrievePart

Routine PlacePart
BEGIN
    MOVE TO TRAY_1
    -- Robot move to pounce position above the column row nest,
    -- The spacing between each column and row is 100mm
    -- () must be used when calculating a position.
    $MOTYPE=LINEAR
    MOVE RELATIVE VEC((-100+ROW*100),(-100+COLUMN*100),0)
    MOVE RELATIVE VEC(0,0,-150)
    OPEN HAND 1
    DELAY 1000
    MOVE AWAY 150.00
END PlacePart
```

```
-- ************  Main Program  ************

BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  $MOTYPE=JOINT
  $TERMTYPE=FINE
  $UTOOL = POS(0,0,134,-90,-90,0,'')
  -- Set parts count to 0
  PARTSCOUNT=0

  NEST_C1R1=TRUE
  NEST_C2R1=TRUE
  NEST_C3R1=TRUE
  NEST_C1R2=TRUE
  NEST_C2R2=TRUE
  NEST_C3R2=TRUE
  NEST_C1R3=TRUE
  NEST_C2R3=TRUE
  NEST_C3R3=TRUE

REPEAT
  -- ROBOT MOVE HOME POSITION
  MOVE TO HOMEGP
  DELAY 100
  WRITE ('ENTER OCCUPIED NEST IN TRAY 2',CR)
  GetColumnNo
  GetRowNo
  RetrievePart
  PlacePart
  PARTSCOUNT=PARTSCOUNT+1
UNTIL PARTSCOUNT=PARTSTOTAL
END Robot
```

c) Play simulation. Interact and observe the simulation. Transfer all parts from "TRAY_2" to "TRAY_1". Test the "error proofing" when inputting the column number and row number. Reload  the simulation each time you play the simulation. Verify the track only loops a total of 9 times

d) Comment the track code explaining each function command and the process.

e) Save the project model  .



Fixture with Two Dimensional Matrix