

Exercise 4 - Logic Statements

Objectives:

- Create basic robot programs in simulation.
- Understanding Logic statements.
- Using an end of arm gripper to manipulate parts.
- Using integers to create a 2 dimensional matrix.
- Using routines to assign various robot tasks.
- Create an Animation file (an .avi).
- Introduction to Arrays.










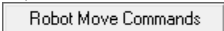
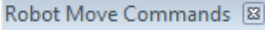

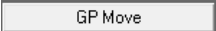
Materials;

- Workspace LT[®] simulation software.
- Workspace LT[®] project file "Exercise 4 - Logic Statements.WSLT"
- Manual "Workspace LT[®] User Guide.pdf".




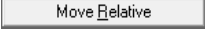









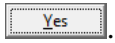

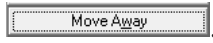

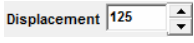




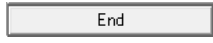


Helpful Hint; Before starting this exercise, review the User Guide sections;

- 6.3 - Comments and Workspace commands
- 6.4.5 Variables
- 6.7.1 - INTEGER and REAL functions
- 9.1.1 - IF ... ENDIF
- 9.2.3 - WHILE ... ENDWHILE
- 12.1.2 Recording an animation

1) Procedure: Create the Robot track

- a) Open Workspace LT simulation software. 
- b) Open  the project file "Exercise 4 - Logic Statements.WSLT".
- c) Add a track for robot  **ABB_1200_5_90**. Enter the Track Name "Robot" and select the Language "KAREL 2". Using the mouse  select  when completed.
- d) Visually determine where the tool frame is located.
- e) Begin recording a track using the newly created "Robot.KL".
- f) The  menu appears, select  to start recording.
- g) The  menu alters the command options.
- h) Record the tool frame at POS(0,0,125,0,0,0,").
- i) From the  Menu box, select . Once selected the  menu appears.
- j) Using the mouse , select the GP labeled "HOMEGP". Select the command . Although this Project's robot has a predefined home position

use the GP labeled "HOMEGP" as home. Confirmation will be required as is true for all GP moves during a track recording. This move has now been recorded in the Track.

- k) Using the mouse , select the GP labeled "PALLET1CLR". Select the command . This move has now been recorded in the Track. Confirmation will be required select .
- l) Select the command . The "Relative" dialogue box will appear, enter the values X = 0, Y = 0, Z = 0. select  to accept. The robot will not move from "PARTC1R1" which is directly over "PARTC1R1". Where "PART" is the object name and C1 represents Column 1 and R1 represents Row 1. This is considered a "pounce" position.
- m) Select the command . The "Relative" dialogue box will appear, enter the values X = 0, Y = 0, Z = -125. select  to accept. The robot will place the gripper directly over the object "PARTC1R1" where C1 represents column 1 and R1 represents row 1.
- n) Using the mouse , select the object labeled "PARTC1R1". From the  menu select , the  menu will appear. From this menu box select  and a  menu box will appear, select . The gripper will now close and the object "PARTC1R1" will now be attached to the gripper. In simulation, the gripper has grasped the part. The following lines have been recorded;
- i) CLOSE HAND 1
 - ii) -- ! GRASPOBJ 'PARTC1R1'
- o) From the  menu box, select . The  dialog box will appear, enter  in the Displacement field and select  to accept.
- p) Using the mouse , select the GP labeled "PALLET2CLR". Select the command . This move has now been recorded in the Track.
- q) From the  menu box select  to end recording.
- r) Save the project model .
- s) The basic programming has been recorded and going forward all remaining programming will be completed using the program editor.
- t) Open the text editor to view the program track Robot.KL. Using the mouse , select the "Robot.KL" twice in rapid successions to open the Robot.KL track in the program editor.
- u) Insert the following text to declare needed variables. The text needs to be inserted between the program description comment and the beginning of the main program.

```
CONST
  PARTSTOTAL=9

VAR
  ROW: INTEGER -- Row number
```

COLUMN: INTEGER -- Column number
PARTSCOUNT: INTEGER -- Number of parts processed

- The constant PARTSTOTAL will control the total number of times the track Robot.KL will loop. In this simulation the track will loop 9 times.
- The integer ROW will be indexed and used to calculate the relative move for the object's row position on the pallet.
- The integer COLUMN will be indexed and used to calculate the relative move for the object's column position on the pallet.
- The use of ROW and COLUMN is similar to a typical spread sheet.
- The integer PARTSCOUNT is a counter and is indexed per loop the program is executed.

v) Insert the following text. This is the first of two to routines required. The text needs to be inserted following the variable declarations and before the main program.

```
ROUTINE RETRIEVE_PART
-- Routine to retrieve part from Pallet 1
BEGIN
  $MOTYPE=LINEAR
  -- Robot move to Pallet nest pounce position
  MOVE RELATIVE VEC((-100+COLUMN*100),(100-ROW*100),0)
  -- Robot move gripper to pallet nest
  MOVE RELATIVE VEC(0,0,-125)
  CLOSE HAND 1
  -- Using a 2 dimensional array, Grasp the part
  -- Using variables COLUMN and ROW to determine the object to grasp
  IF ROW=1 THEN
    IF COLUMN=1 THEN
      --! GRASPOBJ 'PARTC1R1'
    ENDIF
    IF COLUMN=2 THEN
      --! GRASPOBJ 'PARTC2R1'
    ENDIF
    IF COLUMN=3 THEN
      --! GRASPOBJ 'PARTC3R1'
    ENDIF
  ENDIF
  IF ROW=2 THEN
    IF COLUMN=1 THEN
      --! GRASPOBJ 'PARTC1R2'
    ENDIF
    IF COLUMN=2 THEN
      --! GRASPOBJ 'PARTC2R2'
```

```

ENDIF
IF COLUMN=3 THEN
  --! GRASPOBJ 'PARTC3R2'
ENDIF
ENDIF
IF ROW=3 THEN
  IF COLUMN=1 THEN
    --! GRASPOBJ 'PARTC1R3'
  ENDIF
  IF COLUMN=2 THEN
    --! GRASPOBJ 'PARTC2R3'
  ENDIF
  IF COLUMN=3 THEN
    --! GRASPOBJ 'PARTC3R3'
  ENDIF
ENDIF
ENDIF
-- Robot move clear of pallet
MOVE RELATIVE VEC(0,0,+125)
END RETRIEVE_PART

```

- The move relative command contains the formula to the column as this value is for X direction. Formulas contained in this command must be in ()'s. The formula is $X=(-100+COLUMN*100)$ and the formula for Y=, $(100-ROW*100)$. This is an example of a 2 dimensional array.
- The "if" is a programming conditional statement. It follow a "Then". If the statement is true than the function performs the following action. In the track program the object "NAME" will be grasped or attached to the robot faceplate. all "if" conditions require an " ENDIF " statement.
- The "END" statement is needed to terminate the routine.

w) Insert the following text. This is the second of the two required routines.

```

ROUTINE LOAD_PALLET
BEGIN
  $MOTYPE=LINEAR
  -- Robot move to Pallet nest pounce position
  MOVE RELATIVE VEC((-100+COLUMN*100),(100-ROW*100),0)
  -- Robot move gripper to pallet nest
  MOVE RELATIVE VEC(0,0,-125)
  OPEN HAND 1
  MOVE RELATIVE VEC(0,0,+125)
END LOAD_PALLET

```

x) Insert the following text. This is the second of two required routines.

```
-- ***** MAIN PROGRAM *****

BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  -- Set speed of robot
  $SPEED=500
  -- Set initial values of integers
  ROW=1
  COLUMN=1
  PARTSCOUNT=0
  $MOTYPE=JOINT
  -- Set Tool Frame
  $UTOOL=POS(0,0,125,-90,-90,0,")
  MOVE TO HOMEGP
REPEAT
  $MOTYPE=JOINT
  MOVE TO PALLET1CLR
  RETRIEVE_PART
  $MOTYPE=JOINT
  MOVE TO PALLET2CLR
  LOAD_PALLET
  COLUMN=COLUMN+1 --Index column counter by 1
  IF COLUMN=4 THEN
    ROW=ROW+1 -- index row counter by 1
    COLUMN=1 -- reset column counter to 1
  ENDIF
  IF ROW=4 THEN
    ROW=1 -- reset row counter to 1
  ENDIF
  PARTSCOUNT=PARTSCOUNT+1 -- index part count by 1
  IF PARTSCOUNT=PARTSTOTAL THEN
    MOVE TO HOMEGP
  ENDIF
UNTIL PARTSCOUNT=PARTSTOTAL -- return to REPEAT until parts
count =3
END Robot
```

- The condition statement "COLUMN=COLUMN+1" is adding 1 to the integer " COLUMN " (indexing).
- The condition statement "IF COLUMN=4 THEN" if true will index the to the integer "ROW" by 1 and reset the integer "COLUMN" to 1 as there are only 3 columns.

- The condition statement " UNTIL PARTSCOUNT=PARTSTOTAL" is similar to a "DO...WHILE" condition statement. Unlike the "DO...WHILE" condition statement, the "UNTIL" condition statement will loop as long as the statement is false. If the statement is true, the program track will stop. The beginning of the loop is marked by the statement "REPEAT".
- y) The program track "Robot.KL" is listed in it's entirety. Comments are listed and lines of the "IF" statements are correctly indented 3 spaces. This technique is use as a visual aid when using "IF" statements.

```

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

CONST
  PARTSTOTAL=9

VAR
  ROW: INTEGER -- Row number
  COLUMN: INTEGER -- Column number
  PARTSCOUNT: INTEGER -- Number of parts processed

ROUTINE RETRIEVE_PART
-- Routine to retrieve part from Pallet 1
BEGIN
  $MOTYPE=LINEAR
  -- Robot move to Pallet nest pounce position
  MOVE RELATIVE VEC((-100+COLUMN*100),(100-ROW*100),0)
  -- Robot move gripper to pallet nest
  MOVE RELATIVE VEC(0,0,-125)
  CLOSE HAND 1
  -- Using a 2 dimensional array, Grasp the part
  -- Using variables COLUMN and ROW to determine the object to grasp
  IF ROW=1 THEN
    IF COLUMN=1 THEN
      --! GRASPOBJ 'PARTC1R1'
    ENDIF
    IF COLUMN=2 THEN
      --! GRASPOBJ 'PARTC2R1'
    ENDIF
    IF COLUMN=3 THEN
      --! GRASPOBJ 'PARTC3R1'
    ENDIF
  ENDIF
  IF ROW=2 THEN

```

```

IF COLUMN=1 THEN
  --! GRASPOBJ 'PARTC1R2'
ENDIF
IF COLUMN=2 THEN
  --! GRASPOBJ 'PARTC2R2'
ENDIF
IF COLUMN=3 THEN
  --! GRASPOBJ 'PARTC3R2'
ENDIF
ENDIF
IF ROW=3 THEN
  IF COLUMN=1 THEN
    --! GRASPOBJ 'PARTC1R3'
  ENDIF
  IF COLUMN=2 THEN
    --! GRASPOBJ 'PARTC2R3'
  ENDIF
  IF COLUMN=3 THEN
    --! GRASPOBJ 'PARTC3R3'
  ENDIF
ENDIF
-- Robot move clear of pallet
MOVE RELATIVE VEC(0,0,+125)
END RETRIEVE_PART

```

ROUTINE LOAD_PALLET

```

BEGIN
  $MOTYPE=LINEAR
  -- Robot move to Pallet nest pounce position
  MOVE RELATIVE VEC((-100+COLUMN*100),(100-ROW*100),0)
  -- Robot move gripper to pallet nest
  MOVE RELATIVE VEC(0,0,-125)
  OPEN HAND 1
  MOVE RELATIVE VEC(0,0,+125)
END LOAD_PALLET

```

-- ***** MAIN PROGRAM *****

```

BEGIN
  $USEMAXACCEL=TRUE
  %INCLUDE Robot#
  -- Set speed of robot
  $SPEED=500
  -- Set initial values of integers
  ROW=1
  COLUMN=1

```

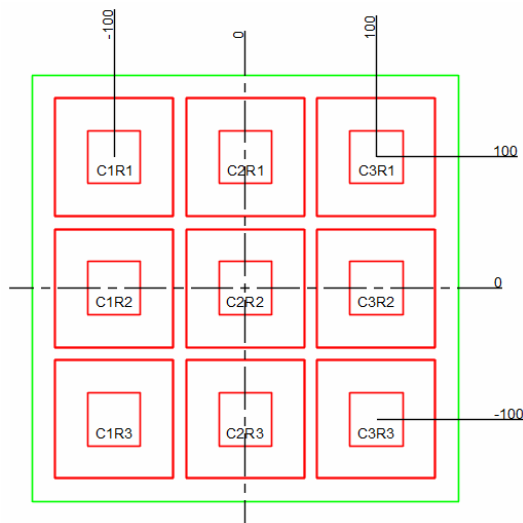


```






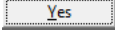
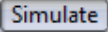
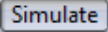
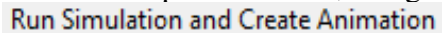
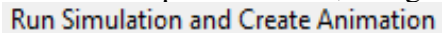
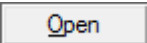
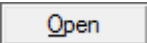
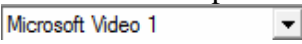
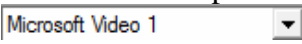




PARTSCOUNT=0
$MOTYPE=JOINT
-- Set Tool Frame
$UTOOL=POS(0,0,125,-90,-90,0,)
MOVE TO HOMEGP
REPEAT
  $MOTYPE=JOINT
  MOVE TO PALLET1CLR
  RETRIEVE_PART
  $MOTYPE=JOINT
  MOVE TO PALLET2CLR
  LOAD_PALLET
  COLUMN=COLUMN+1 --Index column counter by 1
  IF COLUMN=4 THEN
    ROW=ROW+1 -- index row counter by 1
    COLUMN=1 -- reset column counter to 1
  ENDIF
  IF ROW=4 THEN
    ROW=1 -- reset row counter to 1
  ENDIF
  PARTSCOUNT=PARTSCOUNT+1 -- index part count by 1
  IF PARTSCOUNT=PARTSTOTAL THEN
    MOVE TO HOMEGP
  ENDIF
UNTIL PARTSCOUNT=PARTSTOTAL -- return to REPEAT until parts
-- count equals 3

END Robot

```



Fixture 1 with Two Dimensional Matrix

- z) Save the project model .
- aa) Using the mouse  select  (Play simulation) and observe the simulation.
- bb) After the simulation is finished the project must be reloaded to reset before editing or playing the simulation again. Using the mouse  select  found on the left side of the tool bar. The reloading of the model may require a confirmation be confirmed .
- 2) Procedure: Create an AVI
- a) Using the mouse  select  found on the tool bar. This will open a drop down menu.
- b) from the drop down menu, using the mouse  select .
- c) The "Rename" dialogue box will appear, using the mouse  select  to accept the default values. The "Video Compression" menu box will open.
- d) Using the mouse  change the "Compressor" field to .
- e) Using the mouse  select  to begin the process of replaying the simulation and creating an AVI.
- f) After the simulation is finished and the AVI has been recorded, reloaded the project model. Using the mouse  select  found on the left side of the tool bar.
- g) The AVI can now be replayed in any Windows compatible media player program.