Drexel University
Goodwin School of Technology and Professional Studies
Engineering Technology
MET 205: Robotics and Mechatronics
Lab 5 Sensor and Programming - QRCX

Objective:

1. To learn interfacing sensors with the controller.

2. To pick and place objects.

3. Learning how to integrate the photoelectric switch sensor with robot controller.

Some More Commands:

1. DO (Digital Output)
   Syntax: DO(variable) = <expression>
   This command statement outputs the specified value to a DO (digital output) port.

2. Label Statement
   Syntax :< label>:
   This command defines labels that are located at the head of program lines. Labels must begin with a "* " and end with":".

3. GOTO Statement
   Syntax: GOTO <label>
   This statement will cause the robot to jump unconditionally to the <label> specified.

4. If Statement

   IF <expression> THEN | <label 1> / <command statement 1> | [ELSE | <label 2> / <command statement 2>]

   If the conditions of the <expression> are met, this command will execute block 1.
   If there is and ELSE condition specified, and the condition is not met, the robot will move to block 2.

Example:

```
*L0:
IF DI ( 1 ) = 1 THEN
MOVE P, P1
DO(30) = 1
DELAY 100
ELSE
MOVE P, P2
DO(31) = 1
DELAY 100
END IF
GOTO *L0
```

Exercise:

1. Write a program to pick an object at PT1 and place the object at PT2.
   Hint: DO(23) is the output to control the suction cup.

2. Write a program that picks and places an object to PT2 if the photoelectric switch sensor detects any object in front of it; else goes back to PT1 and keeps looping until the stop button is pressed. Use the If statement to write that program.
   Hint: DI(20) will equal to zero if there is an object in front of the sensor. Otherwise it will be one.

3. Try 3 different speeds in the program in step 1. In each test, record the minimum delay for the suction cup to pick the object successfully. Plot a graph of minimum delay vs. speed percent (or speed (ipm or mm/s)).

Types of Statements	Format	Example	Memory Usage [bytes]	Executable Condition (*1)
Shift Assignation Statement	[LET] Sm=<shift expression>	S1=S0 S[A]=-S1+S2	11 22	1
Jumping	GOTO | GO TO <label>	GOTO * LOOP	8	6
Decisions	IF<expression>THEN <label> | <expression> [ELSE <label> | <expression>]	IF A=1 THEN * PRGEND IF A=1 THEN * L1 ELSE * L2 IF A=1 THEN PRINT "OK" IF A=1 THEN MOVE P, P1 ELSE MOVE P, P2	21 26 25 36	6

IF	<expression> [THEN]
	:
	:
	[ELSE]
	:
	:
	ENDIF
		IF DI3(1)=1 THEN
		MOVE P, P1
		DO(30)=1
		ELSE
		MOVE P, P2
		DO(30)=0
		ENDIF
			19
			
			6
			
			4
				6

Multiple Decisions	SELECT[CASE]<expression> CASE<expression list 1> [Block 1] [CASE<expression list 2> [Block 2]] : : [CASE ELSE [[Block n]] END SELECT	SELECT CASE DI3() CASE 1 CALL *EXEC(1,10) CASE 2 CALL *EXEC(21,30) CASE ELSE END SELECT	10 10 10 6 4	6
Subroutines	GOSUB | GO SUB <label> : : RETURN	DECLARE * INITIALIZE GOSUB * SUBROUTIN GO SUB * INITIALIZE : RETURN	 8 8 4	6
Jumping on Condition	ON <expression> GOTO | GO TO <label> [, <label>...]	ON A GOTO * L10, * L20, * L30	22	6
Conditional Subroutine	ON <expression> GOSUB | GO SUB <label> [, <label>, ...]	ON A GOSUB * SB10, * SB20, * SB30	22	6
Loops	FOR <variable>=<expression 1> TO <expression 2> [STEP <expression 3>] : NEXT <variable>	FOR A=10 TO 4 STEP -2 : NEXT A FOR B(0)=0.1 TO 0.5 STEP 0.1 : NEXT B(0)	24 6 34 6	6 6
End Loops	EXIT FOR	EXIT FOR	4	6

Types of Statements	Format	Example	Memory Usage [bytes]	Executable Condition (*1)
Array Variable Declaration Statement	DIM <array definition> [, <array definition>,.....]	DIM A%(10) DIM B(2, 2, 2), C%(3,2) DIM C$ (5)	11 24 11	6
Function Definition Statements	DEF FN<name>[% | ! | $] [(<parameter>, <parameter>...)]=<function definition expression>	DEF FNAPAI=3.141592 DEF FNASIN(X)=ATN(X/SQR(-X^2+1))	13 29	6
Point Definition	Pm=x y z r a b	P123=100.00 200.00 0.00 0.00 0.00 0.00	31	1
Shift Coordinate Definition	Sm=x y z r	S0=-123.45 123.45 123.45 123.45	21	1
Definition	SUB[(<parameter>[, <parameter>...])] : END SUB	SUB *DISTANCE(X!, Y!, D!) : END SUB	19 4	6
Referring to Global Variables	SHARED <simple variable>[( )][, <simple variable>[( )]...]	SHARED X!, Y!, D!( )	12	6
Referring to External Symbols	DECLARE <label>[, <label>...] DECLARE SUB <label>[(<parameter>[, <parameter>....])]	DECLARE ÂñDISTANCE, *AREA DECLARE SUB *COMPARE DECLARE SUB *AREA(X!, Y!)	14 11 17	6
Main Robot Hand Definition	HAND Hn=<1st parameter><2nd parameter><3rd parameter>[R]	HAND H1= 0 150.0 0.0 HAND H2= 45.0 20.0 0.0 R	20 20	1
Sub Robot Hand Definition	HAND2 Hn=<1st parameter> <2nd parameter> <3rd parameter> [R]	HAND2 H5= 0 150.0 0.0 HAND2 H6= 45.0 20.0 0.0 R	21 21	1
Arithmetical Assignation Statement	[LET] <variable>=<expression>	A=10 B(0)=10.05 LOCX(P1)=A(1) LOCX(S1)=100.00	13 20 21 17	1
Character String Assignation Statement	[LET] <character string variable>=<character string expression>	A$="YAMAHA" B$ (1) =A$+" ROBOT"	24 29	1
Point Assignation Statement	[LET] Pm=<point expression>	P1=P100 P[A]=P200+P5 P[START_POINT]=P300/2-4*P3	13 23 34	1 6

Types of Statements	Format	Example	Memory Usage [bytes]	Executable Condition (*1)
Sub Robot Absolute Movement	MOVE2 P, <point definition>[, <, option>[, <option>...]]	MOVE2 P, P100 MOVE2 P, 100.00 100.00 200.00 0.00 0.00 0.00 MOVE2 P, P[A], P10+P20, S=70, Z=0 MOVE2 P, P1, STOPON DI3(0)=1	13 32 44 26	1
Sub Robot Relative Movement	MOVEI2 P, <point definition>[, <, option>[, <option>...]]	MOVEI2 P, P100 MOVEI2 P, 100.00 100.00 200.00 0.00 0.00 0.00 MOVEI2 P, P[A], P10+P20, S=70	13 32 36	1
Sub Group Absolute Axis Unit Movement	DRIVE2(<expression>,<expression> | <point expression>) [, (<expression>, <expression> | <point expression>)...] [,S | SPEED =<expression>]	DRIVE2(1,100.00) DRIVE2(2,100.00),(3,50.00) DRIVE2(A, P10), S=10	16 26 21	1
Sub Group Relative Axis Unit Movement	DRIVEI2(<expression>,<expression> | <point expression>) [, (<expression>, <expression> | <point expression>)...] [,S | SPEED =<expression>]	DRIVEI2(1,100.00) DRIVEI2(2,100.00),(3,50.00) DRIVEI2(A, P10), S=10	16 26 21	1
Sub Group SERVO Control	SERVO2 ON | OFF | FREE [(<expression>)]	SERVO2 ON SERVO2 OFF SERVO2 FREE(2)	7 7 11	1
DO Output	<DO variable>=<expression>	DO2( )=&B10111000 DO2(7, 5)=2 DO(27, 25, 23, 20)=&H0C	13 15 18	1
MO Output	<MO variable>=<expression>	MO( )=&B10111000 MO(7, 5)=2 MO(27, 25, 23, 20)=&H0C	13 15 18	1
TO Output	<TO variable>=<expression>	TO0( )=&B10111000 TO0(7, 5)=2 TO(07, 05, 03, 00)=&H0C	13 15 18	1
LO Output	<LO variable>=<expression>	LO0( )=&B10111000 LO0(7, 5)=2 LO(07, 05, 03, 00)=&H0C	13 15 18	1
SET Statements	SET <DO variable> | <MO variable> | <TO variable> | <LO variable> [, <expression>]	SET DO2(7, 5, 0) SET MO(27, 25, 23, 20), 1000 SET TO0() SET LO0(7, 5)	10 17 8 10	3
RESET Expressions	RESET <DO variable> | <MO variable> | <TO variable> | <LO variable>	RESET DO2( ) RESET MO(27, 25, 20) RESET TO0(7, 5, 6, 1) RESET LO(07, 05, 00)	8 13 10 13	1
PRINT Statements	PRINT [<expression>][, | ;] <expression>][, | ;]	PRINT "COUNT=";C, "TIME=";T PRINT A$ PRINT "P10=";P10 PRINT S[A]	33 8 17 14	1

Types of Statements	Format	Example	Memory Usage [bytes]	Executable Condition (*1)
Conditional	Loops WHILE <expression> : WEND	WHILE A>10 WHILE DI1( )=&B10111100 : WNED	14 14 6	6
Calling Procedure	CALL <label>[(parameter, [, parameter, ...])]	CALL * DISTANCE CALL * AREA(2.5, X!, REF Y!)	11 41	6
Ending Procedure	EXIT SUB	EXIT SUB	4	6
Error Processing	ON ERROR GOTO <label> | 0 : RESUME [NEXT | <label>]	ON ERROR GOTO * ER1 ON ERROR GOTO 0 : RESUME RESUME NEXT RESUME * L1	9 5 5 6 9	6
Temporary Suspension	HOLD [<expression> | <character string>]	HOLD HOLD "ERROR STOP"	5 19	6
Ending Movement	HALT [<expression> | <character string>]	HALT HALT "PROGRAM STOP"	5 21	6
Program Switching	SWI <program name>	SWI <ABC>	14	2
Main Robot Absolute Movement	MOVE P | L | C , <point definition>[, <, option> [, <option>...]]	MOVE P, P100 MOVE L, 100.00 100.00 200.00 0.00 0.00 0.00 MOVE P, P[A], P10+P20, S=70, Z=0 MOVE P, P1, STOPON DI3(0)=1 MOVE C, P1, P2, S=50 MOVE L, P1, STOPON DI3(0)=1	12 31 43 25 25 25	4
Main Robot Relative Movement	MOVEI P, <point definition>[, <, option>[, <option>...]]	MOVEI P, P100 MOVEI P, 100.00 100.00 200.00 0.00 0.00 0.00 MOVEI P, P[A], P10+P20, S=70	12 31 35	1
Main Group Absolute Axis Unit Movement	DRIVE(<expression>,<expression> | <point expression>) [, (<expression>, <expression> | <point expression>)...] [,S | SPEED =<expression>]	DRIVE(1,100.00) DRIVE(2,100.00),(3,50.00) DRIVE(A, P10), S=10	15 25 20	1
Main Group Relative Axis Unit Movement	DRIVEI(<expression>,<expression> | <point expression>) [, (<expression>, <expression>)...] | <point expression> [,S | SPEED =<expression>]	DRIVEI(1,100.00) DRIVEI(2,100.00),(3,50.00) DRIVEI(A, P10), S=10	15 25 20	1
Main Group SERVO Control	SERVO ON | OFF | FREE [(<expression>)]	SERVO ON SERVO OFF SERVO FREE(2)	6 6 10	1
Waiting for Condition	WAIT <condition expression>[, <expression>] WAIT ARM | ARM2 [(<expression>)]	WAIT DI2( )=&B10101011 WAIT DI2(3, 1)=&B10, 1000 WAIT DI(17, 15, 13, 10)=&H0C WAIT DO2(3, 1)=&B10 AND DI1(7)=1 WAIT ARM WAIT ARM2(1)	14 20 19 26 8 11	6