KOLLMORGEN®

Because Motion Matters™

This is a Discontinued Product

Contact Kollmorgen Customer Support at
1-540-633-3545 or email us at support.kollmorgen.com
if assistance is required.

Section 1 Programming Basics Exercise 1.1

Introduction

At the beginning of this course it is understood that you already have some basic programming experience. This assumption includes that you know about conditional and incremental looping structures, prompting the user for input and printing information to the terminal. Also, it is expected that you know the concept of variable definition and are familiar with I/O basics. Lastly, these exercises are based on the idea that you know **if…then** statements and their functionality. Hints will be given for solving the exercise at the bottom of the page, however these hints will only be given in the first exercise utilizing that concept. Some exercises will build on one another.

With this in mind, the purpose of this section, **Programming Basics,** is to introduce you to the specific commands used by **950 Basic**(SC950 language) and **MCPI** (Warpdrive language) to perform these fundamental programming concepts. In **Section 1** we will not control the motion of the motor, but only communicate to the drive.

Objective

Write a program that

- Calculates the distance traveled by a bottle on a conveyor belt in 5 seconds if the gear driving the belt has a 3 inch radius and is traveling at 60 RPM
- Outputs the calculated distance in the following format

" The bottle traveled [calculation] inches"

Given

The linear distance is equal to the circumferential length multiplied by the number of revolutions. The number of revolutions is equal to the velocity multiplied by time.

$$circumference = 2\pi r, \quad velocity = \frac{dist}{time}, \quad \text{min} = 60s$$

Hints	**950 Basic**	**MCPI**
1.	Dim [variable] as	[variable] =
2.	float, integer, string	
3.	Input	INPUT
4.	Print	PRINT#n

Exercise 1.2

Introduction

The purpose of this exercise is to introduce you to the **I/O** commands used by **950 Basic** and **MCPI**. In addition, you will use the conditional **While** loop to continuously loop a portion of the exercise.

Objective

Write a program that indefinitely looks at the state of input 1 (Warpdrive [or input 7{SC950}]) in order to set the state of output two. Write the program so that input 1 and output 2 begin at different states. As soon as input 1(Warp) or input 7(SC950) transfers state (by manually flipping the switch), command the drive to also reverse the state of output two. Maintain this functionality continuously until you stop the program from running

Hints	**950 Basic**	**MCPI**
1.	I/O active low	I/O active high
2.	While:Wend	DO {WHILE or UNTIL}:LOOP
3.	Inp[n]	IN[n]
4.	Out[n]	OUT[n]
5.	While 1:Wend	DO WHILE 1: LOOP

Exercise 1.3

Introduction

The purpose of this exercise is to have you become familiar with incremental looping using the Pacific Scientific SC950 and Superior Warpdrive programming environments

Objective

Write a program that solves the equation

$$y = x^3 - 3$$

for values of x in increments of 1 from 1 to 100. Print the values of x and y in the following format:

"x = [increment] , y[n] = [calculated value]"

Hints	950 Basic	MCPI
1.	For…Next	FOR…NEXT

Exercise 1.4

Introduction

The goal behind this exercise is to familiarize you with if…then statements used by the Warpdrive and the SC950.

Objective

Read through the following **950 Basic** code and fill in the blanks in the table at the bottom of the page. When this is done, convert this code to an **MCPI** program and check your answers by running it on the Warpdrive.

Code

```
X = 0
ACCELRATE = 1
DECELRATE = 1
INPUT "ENTER RUNSPEED"; RUNSPEED
IF RUNSPEED > 50 AND RUNSPEED < 100 THEN
        ACCELRATE = RUNSPEED * 0.5
        DECELRATE = ACCELRATE * 2
ELSEIF RUNSPEED > 100 AND RUNSPEED < 500 THEN
        ACCELRATE = RUNSPEED * .25
        DECELRATE = ACCELRATE * 4
        IF ACCELRATE < 75 THEN
                OUT1 = 0
                X = 1.5
        ELSE
                OUT1 = 1
                X = 1
        END IF
ELSEIF RUNSPEED > 500 AND RUNSPEED < 1500 THEN
        ACCELRATE = RUNSPEED * 1.5
        DECELRATE = ACCELRATE * 8
ELSE
        PRINT "VALUES OUT OF RANGE"
END IF
```

RUNSPEED	ACCELRATE	DECELRATE	OUT1	X
600				
300				
200				
50				
100				

Exercise 1.5

Introduction

This exercise will expose you to the way subroutines are handled by the programmable OC950 card and the SS2000D6i.

Objective

Write a program that:

- Prompts a user to enter the desired distance in inches the bottle from **Exercise 1.1** should travel on the conveyor belt
- Calls a subroutine to calculate the emulated encoder counts associated with the inputted distance of travel if the motor has a 4096 counts per rev
- Outputs this value to the terminal screen in the following format:

"Number of counts: [value]"

Hints	**950 Basic**	**MCPI**
1.	Sub…End Sub	GOSUB…RETURN

Exercise 1.6 (cont.)

D6i Code

```
REAL   X,VELOCITY

X = 0
ACCEL = 1
DECEL = 1

PRINT#1,"ENTER VELOCITY: ";
INPUT#1, VELOCITY
IF VELOCITY > 50 AND VELOCITY < 100 THEN
        ACCEL = VELOCITY * 0.5
        DECEL = ACCEL * 2
ELSE
        IF VELOCITY > 100 AND VELOCITY < 500 THEN
                ACCEL = VELOCITY * .25
                DECEL = ACCEL * 4
                IF ACCEL < 75 THEN
                        OUT(1)=0
                        X = 1.5
                ELSE
                        OUT(1)=1
                        X = 1
                END IF
        ELSE
                IF VELOCITY > 500 AND VELOCITY < 1500 THEN
                        ACCEL = VELOCITY * 1.5
                        DECEL = ACCEL * 8
                ELSE
                        PRINT#1, "VALUES OUT OF RANGE"
                END IF
        END IF
END IF
END
```

Section 2 – Creating Motion Profiles Exercise 2.1

Introduction

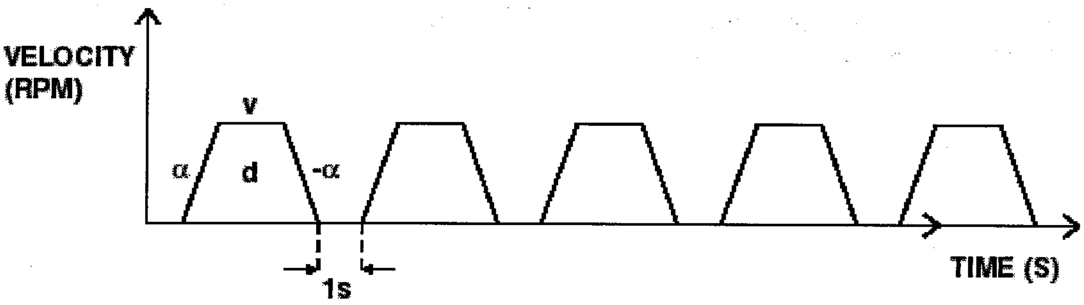
The purpose of this section is to familiarize the Sales Specialist with what is needed to create a motion profile. The information necessary to generate a motion profile is:

- Distance(d)
- Velocity(v)
- Accleration(α_1)
- Deceleration(α_2)

These parameters are commonly referred to as the move parameters and will be referred to as such throughout the remainder of the exercises. The two foundations of the motion profile we will look at are the index move and the absolute move. The difference between the two is whether you are generating a move in reference to a particular point (absolute move) or whether you command a move solely from the current point (incremental or index move). This exercise is concerned with incremental moves.

Objectives

Write a program that performs 5 identical index moves using the parameters listed below. Include between each move a 1 second pause. Print “move complete” after the last index has finished.



$\alpha_1 = -\alpha_2 = 3500$ RPM/s
v = 425 RPM
d = 15 Revs

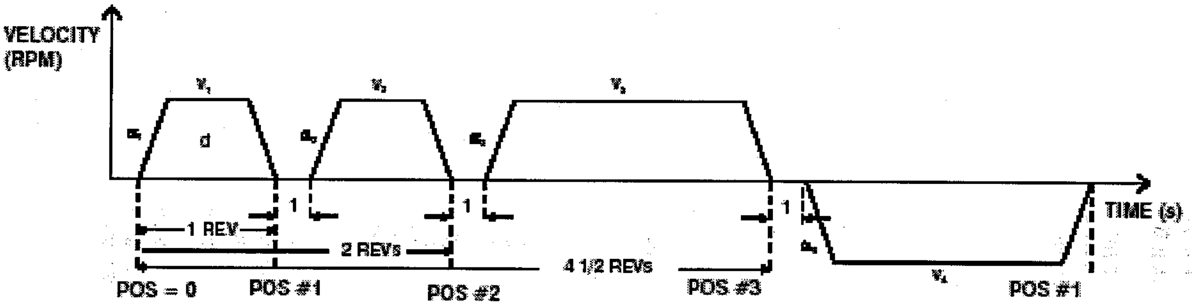
Hints	**950 Basic**	**MCPI**
1.	IndexDist	MOVEI
2.	GoIncr	
3.	RunSpeed	SPEED
4.	AccelRate	ACCEL
5.	DecelRate	DECEL
6.	Pause	WAIT

Exercise 2.2

Objectives

Write a program that performs four absolute moves all of which are referenced to the original position. Set the internal position counter equal to zero before the first move is begun to set the reference position.

The move should have final positions of 1 rev from 0, 2 revs from 0, 4.5 revs from 0, and 1 rev from 0.



$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 3600$ RPM/sec
$v_1 = v_2 = v_3 = v_4 = 200$ RPM

Hints	**950 Basic**	**MCPI**
1.	TargetPos	
2.	GoAbs	MOVEA
3.	PosCommand	ABSPOS

Exercise 2.3

Introduction

The purpose of this exercise is to have you make changes in a motion profile while the move is under way.

Objective

Write a program that commands a move of 100 revolutions from a zero reference point using the below listed parameters. At 50 revolutions change the speed so that the second part of the move takes half the time of the first.

α **= 2400 RPM/sec**
v_1 **= 300 RPM**
v_2 **= ?**
d = 100 revs

Hints	**950 Basic**	**MCPI**
1.	Position	ABSPOS
2.	When	SPEED
3.	UpdMove	

Section 3 General Purpose Commands Exercise 3.1

Introduction

The purpose of this exercise is to introduce you to how interrupts are handled in the **950 Basic** and **MCPI** software as well as expose you to jog moves.

Objective

Write a program that commands a constant velocity move of 50 RPM. Write an interrupt routine that advances the motor 5 rotations at a speed of 250 RPM and an acceleration of 2700 RPM/sec whenever input 1 is switched to its active state.

Hints	**950 Basic**	**MCPI**
1.	GoVel	JOG
2.	Intr	INTRONn
3.	Interrupt…End Interrupt	INTROFFn
4.		ON…INTRn

Exercise 3.2

Introduction

The purpose of this exercise is to familiarize you with a homing sequence using the PacSci SC950 Servo and Superior's WarpDrive stepper.

Objective

Write a program which will home a shuttle traveling on a foot long lead screw. This setup is such that the shuttle is on the far end of the screw and to return it to the home position the motor must be rotated CW. Emulate the trip of the prox sensor by the shuttle with input 1(Warp) or input 7(SC950) on the I/O board. Home this shuttle using the commands listed in the section entitled hints below.

In addition take note of the fact that with the warp drive you must go to the ***Configuration and Setup*** section of **MCPI** to configure the inputs for EVENT1.

Hints	**950 Basic**	**MCPI**
1.	When, INP7 =	EVENT1
2.	Continue	MOVEHOME
3.	WhenPosition	
4.	Position	