UNIT 15
INSTRUCTOR'S MANUAL

CASCADING TIMERS

1. No new keystrokes are introduced in this unit. However, the students are expected to use their existing knowledge in new ways. If they have any difficulties, you may wish to review earlier units before continuing.

2. This Experiment section is the first occasion for the students to develop a PLC program for an industrial application. If you have the opportunity, it would be helpful for the students to see a pneumatic robot first hand and to observe it operating under PLC control.

3. Because the robotics may be new to the students, you may wish to allow two class sessions for the completion of this unit. You should also consider having students work in pairs or groups of three to facilitate their successful completion of the Experiment.

UNIT 15

CASCADING TIMERS

Objectives

Upon completion of this unit the student will be able to:

1. Explain the operation of cascading timers.
2. Describe motion diagrams, motion times, and time listings.
3. Enter a program with cascaded timers.
4. Write a program using cascading timers to control a simple pick and place robot.

Background

In Unit 14 there were several timers in some of the programs. The timers always started running at the same time, which is very awkward. It is more difficult to monitor timers which are operating concurrently. It is also more difficult to "debug" (fix) or edit such a program.

For almost all PLC uses, timers are programmed so that only one timer is operating at a time. If outputs must operate at specific time intervals, one after the other, several timers are used. The first timer turns on the second timer, the second timer turns on the third timer and so forth. When timers control each other in this type of series they are called *cascading timers*. Figure 15-1 is a part of a PLC program. It shows only the cascading timers. Somewhere else in the program these timers each control one or more outputs which are activated at the preset time intervals. The reset elements for the cascaded timers are also located elsewhere in the program.

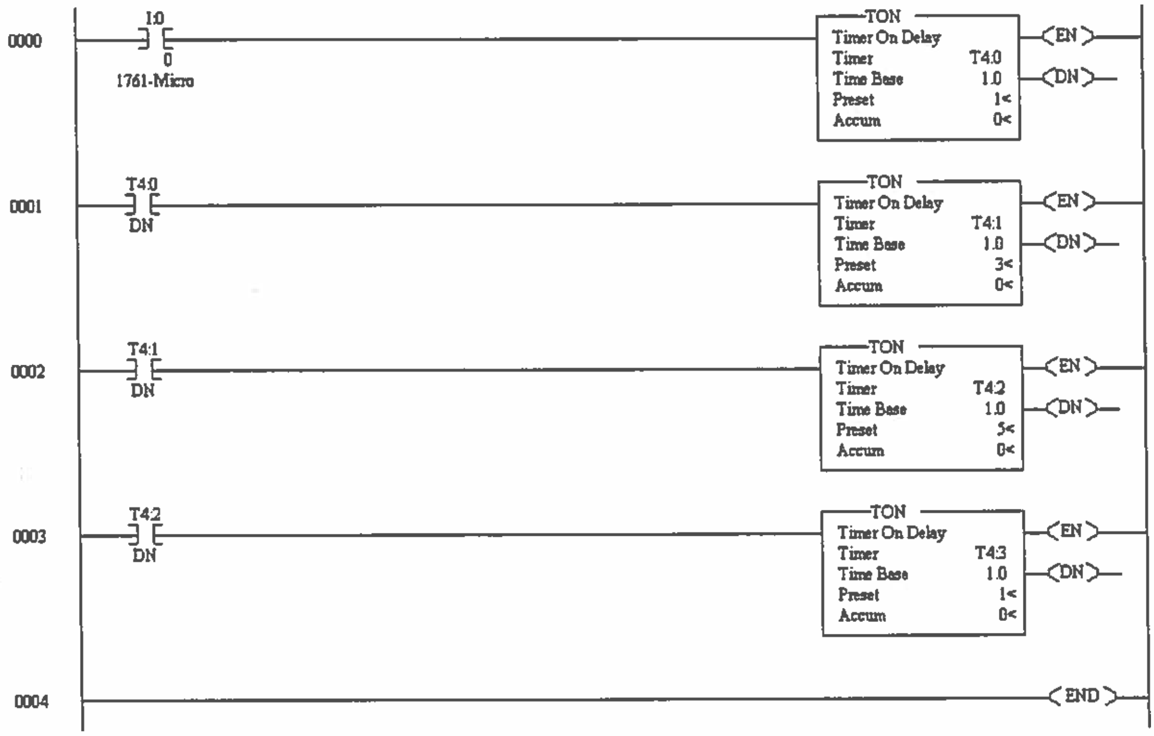


Figure 15-1
Cascading Timers to Control Several Outputs

Cascading timers are also useful when a large preset value is required. If a timer can only be programmed to 32,767 seconds, it is useless in a program requiring a preset value of 86,400 seconds. Three timers can be cascaded to solve the problem as shown in Figure 15-2.

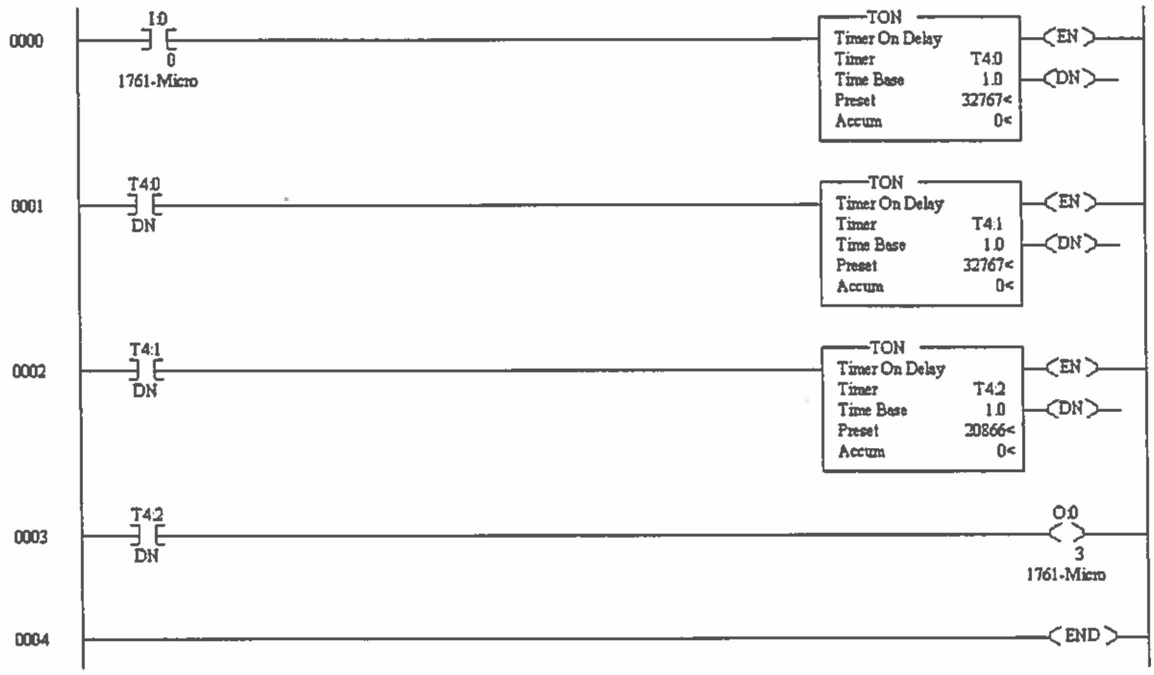


Figure 15-2
Cascading Timers to Extend Preset Time Interval

Cascading timers are commonly used to control pick and place pneumatic robot arms. In order to program the robot arm using cascading timers, it is important to understand how the robot arm operates. Arm motion must be listed and timed. This data is then entered into a ladder logic diagram.

Sample Pneumatic Robot Arm

Pneumatic robot arm are run by air pressure which is controlled by a set of electric (solenoid) air valves. Robot arms vary in the number of motions they have and the time it takes to execute each motion. The robot in this example has three motions of travel and a set of grippers (fingers) for picking up parts (see Figure 15-3). The arm moves left and right, up and down, and in and out. The grippers open and close.

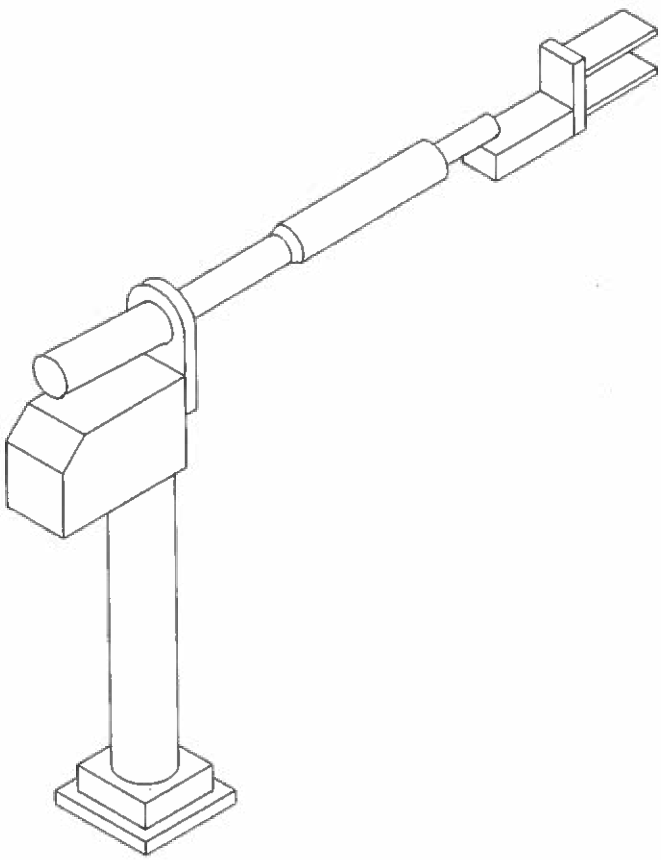


Figure 15-3
Pneumatic Robot Arm

This robot arm is controlled by a set of four solenoid valves. Table 15-1 lists each valve and gives the corresponding position of the arm when the valve is on and off. The table also identifies the program output which controls each valve.

Valve Number	On	Off	Output
1	Arm right	Arm left	0
2	Arm down	Arm up	1
3	Arm out	Arm in	2
4	Gripper close	Gripper open	3

Table 15-1
Sample Valve Operation

Every arm motion takes several seconds to complete. The motion times are listed in Table 15-2.

Motion	Time in seconds
Arm left	3.0
Arm right	3.0
Arm up	3.0
Arm down	2.0
Arm in	3.0
Arm out	3.0
Gripper open	1.0
Gripper close	1.0

Table 15-2
Sample Robot Arm Motion Times.

Figure 15-4 provides a diagram of the motions the robot must follow during its programmed operation. The numbers, which appear at the beginning of the motion arrow, represent individual steps in the robot's motion. They are not correlated to the rung numbers in the PLC program, but they do provide the sequence for the motions.

The motion legend to the right of the motion diagram defines what each straight arrow means. The curved arrows indicate the operation of the gripper. Every robot motion diagram begins with the robot in its home position. The home position for this robot is up and to the left, arm out with grippers open.

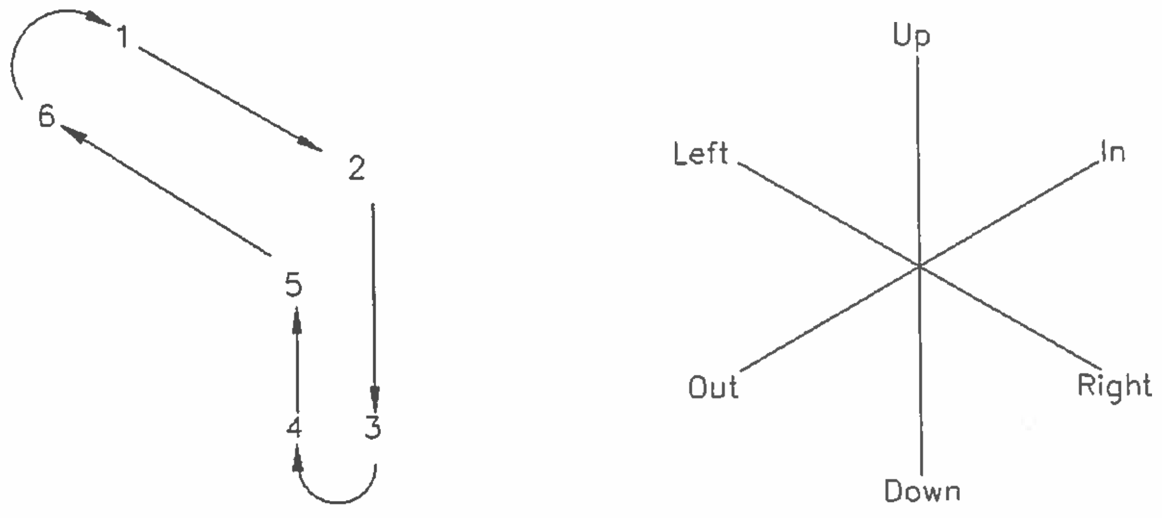


Figure 15-4
Simple Motion Diagram

The motion diagram is used to develop the time listing given in Table 15-3.

Motion step	Description	Time to Complete Move
1	Arm right	3.0 sec
2	Arm down	2.0 sec
3	Gripper close	1.0 sec
4	Arm up	3.0 sec
5	Arm left	3.0 sec
6	Gripper open	1.0 sec
	Total time	12.0 sec

Table 15-3
Simple Time Listing

A pneumatic robot uses a positive mechanical stop to end a motion. Upon completion of a motion, the robot arm stays in position at the stop until the opposite motion is initiated by the PLC. This characteristic is reflected in the design of the timing section of the PLC program controlling the robot arm.

A timer must be used to initiate all but the first motion in the series described by Table 15-3. The first motion is initiated by the input device. A latching relay can be used to start the program with a momentary input (switch). The timing section of the program, while latching relay, is given in Figure 15-5.

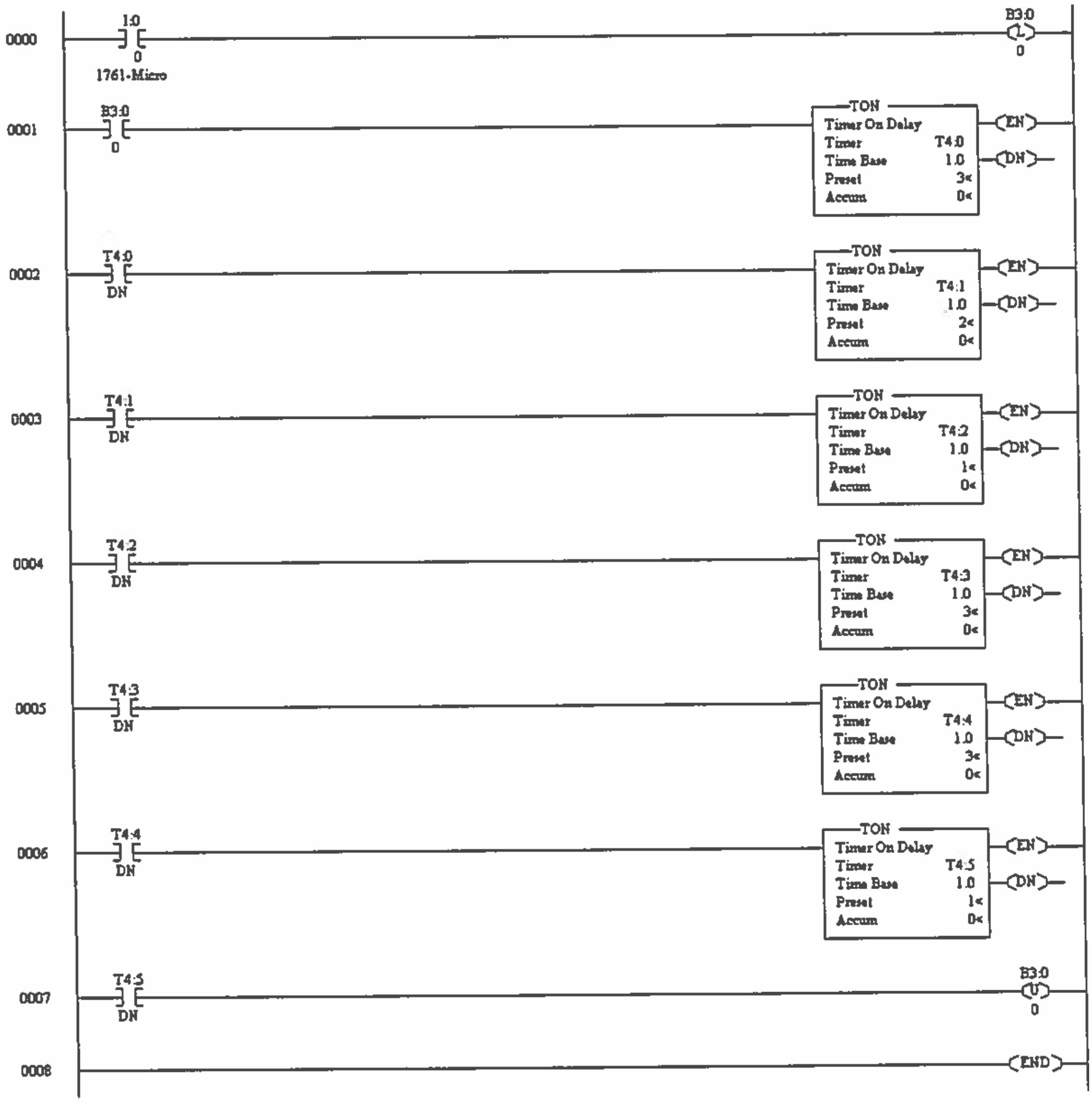


Figure 15-5
Sample Program Timing Section

If you examine the preset values for timers you will see that they follow a set pattern. The preset value of the first timer T4:0 is the time it takes to complete the motion initiated by the input device. The preset value of the second timer T4:1 is the time it takes to complete the motion initiated by the first timer T4:0. Likewise, the preset value of the timer T4:2 is the time it takes to complete the motion initiated by the second timer T4:1. This pattern continues through the entire timing sequence.

The latching relay and each of the timers T4:0 through T4:4 also control a valve powering the robot arm. Figure 15-6 illustrates the section of the ladder logic diagram in which timers control the program outputs.

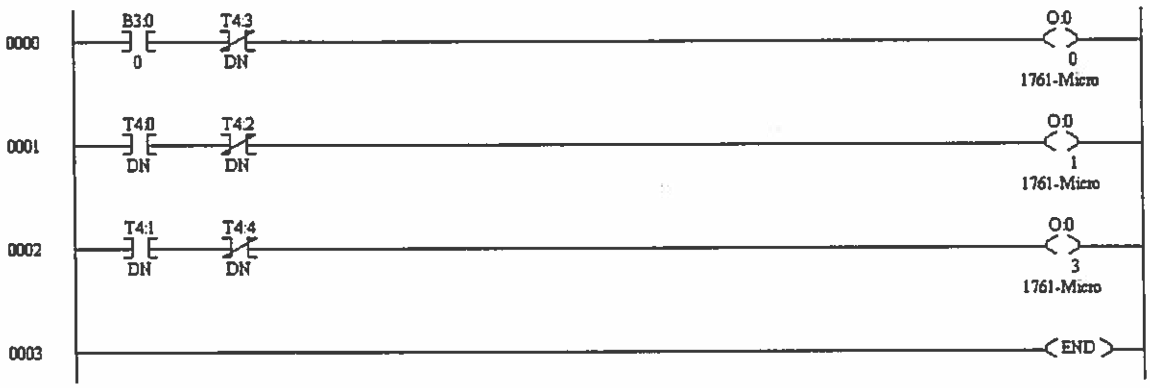


Figure 15-6
Sample Program Output Section

Figure 15-6 shows the standard configuration for sequencing outputs with cascading timers. Each output is controlled by two inputs, one normally open and one normally closed. Each pair of inputs consists of the timers controlling the on and off positions of a single valve. In the sample, B3:0/0 turns on the left/right valve and T4:3 turns off the left/right valve. The two inputs paired can then control both the on and off status of the valve. The same kind of pairing holds true for the other motion used in this program.

Many applications require a set of motions to be repeated under the control of additional timers. All pairs of timers controlling one motion must be ORed together. Suppose the sample program were expanded so that timer T4:11 closes the grippers and timer T4:16 re-opens them. The program output rung for the grippers would have to be modified as in Figure15-7. (You should be aware that the additional motions change the motion diagram, add time to the cycle, and require changes in timer reset operations, etc.)

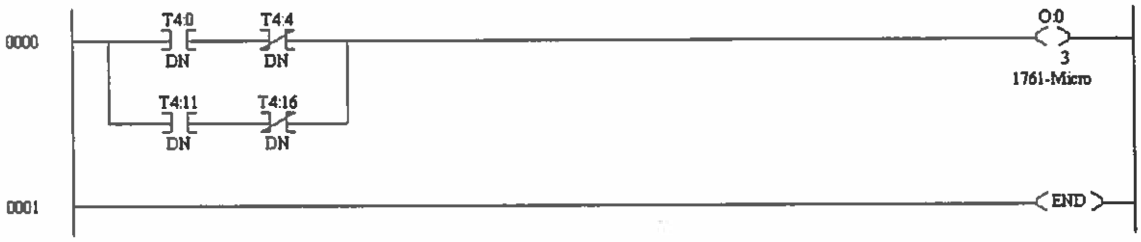


Figure 15-7
ORed Inputs for Repeated Motion

To create a repeating operation of the robot arm, a sixth timer T4:5 is used to reset the timers, including itself. The reset timer's preset value must be the time delay required to complete the last motion in the motion diagram. In this example it is 0.5 seconds. Figure 15-8 illustrates the retentive timers being reset by timer T4:5. It should be noted that it is necessary to place the activation of the reset timer after the output rungs. If it were placed elsewhere in the program, there is the risk that timers will be reset before they can activate their outputs.

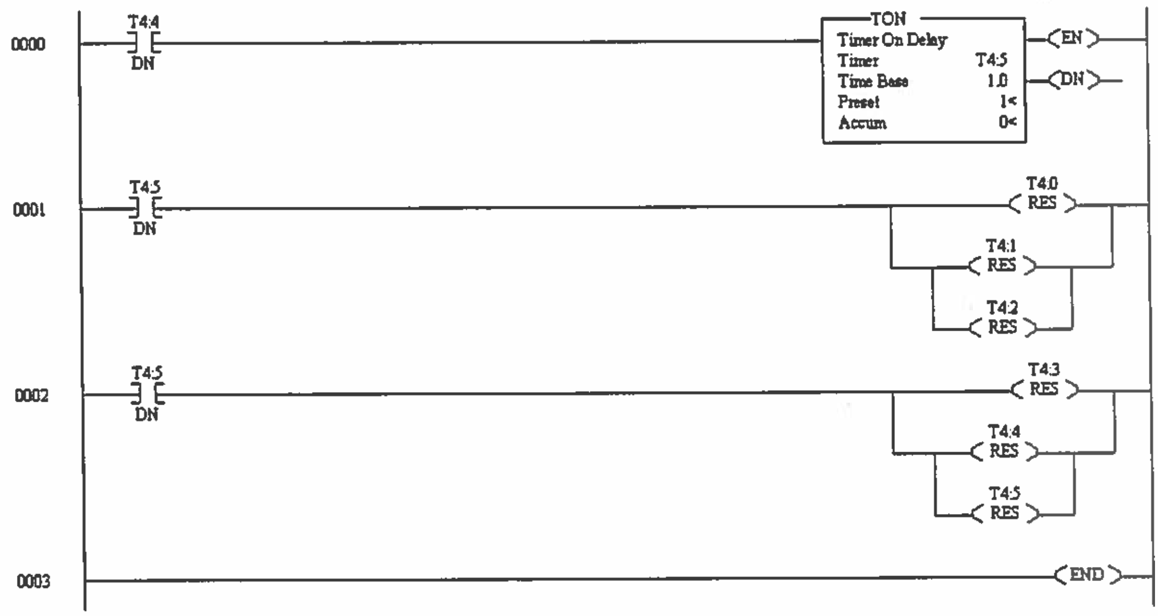


Figure 15-8
Timing Rungs with Resets for Retentive Timers

The program as it has developed will run continuously once it has started. To make it a single cycle program, a normally closed timer contact T4:5 could be used to unlatch relay B3:0/0 when it resets all timers. When a timer is used to unlatch the relay, the unlatching rung must be programmed and entered before the reset rung. This is necessary because the PLC scans and executes the rungs in the order in which they are entered. So, if the timer resets itself first, it will never unlatch the relay.

Programming Keystrokes

The keystrokes for entering cascaded timers are the same as they would be for any other timers. The difference lies in how the program uses the timer to control internal coils and external outputs. When timers are controlling a series of outputs, each timer controls both the timer which follows it and an output. When extending preset values, remember that only the last timer should control the output. Effective programming of cascading timers depends on developing accurate time listings and translating these into the timing section of the program.

EXPERIMENT

Purpose

To develop a control program using cascading timers.

Procedure

1. Assume the robot in this lab is similar to the one in the preceding sample. It begins its program in a preset position with the grippers open. Use the sample robot motion times in Table 15-2 along with the new motion diagram in Figure 15-9 to create a time listing table. Calculate the total cycle time for the robot arm and record all data in Table 15-4. As you work further on this programming task, please note that the in/out control by valve 3 has been reversed. (see Table 15-5).

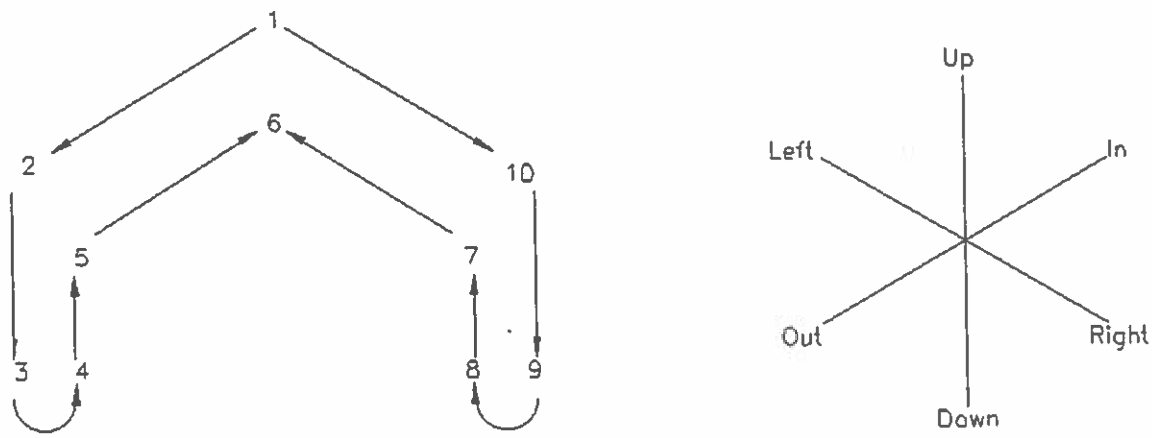
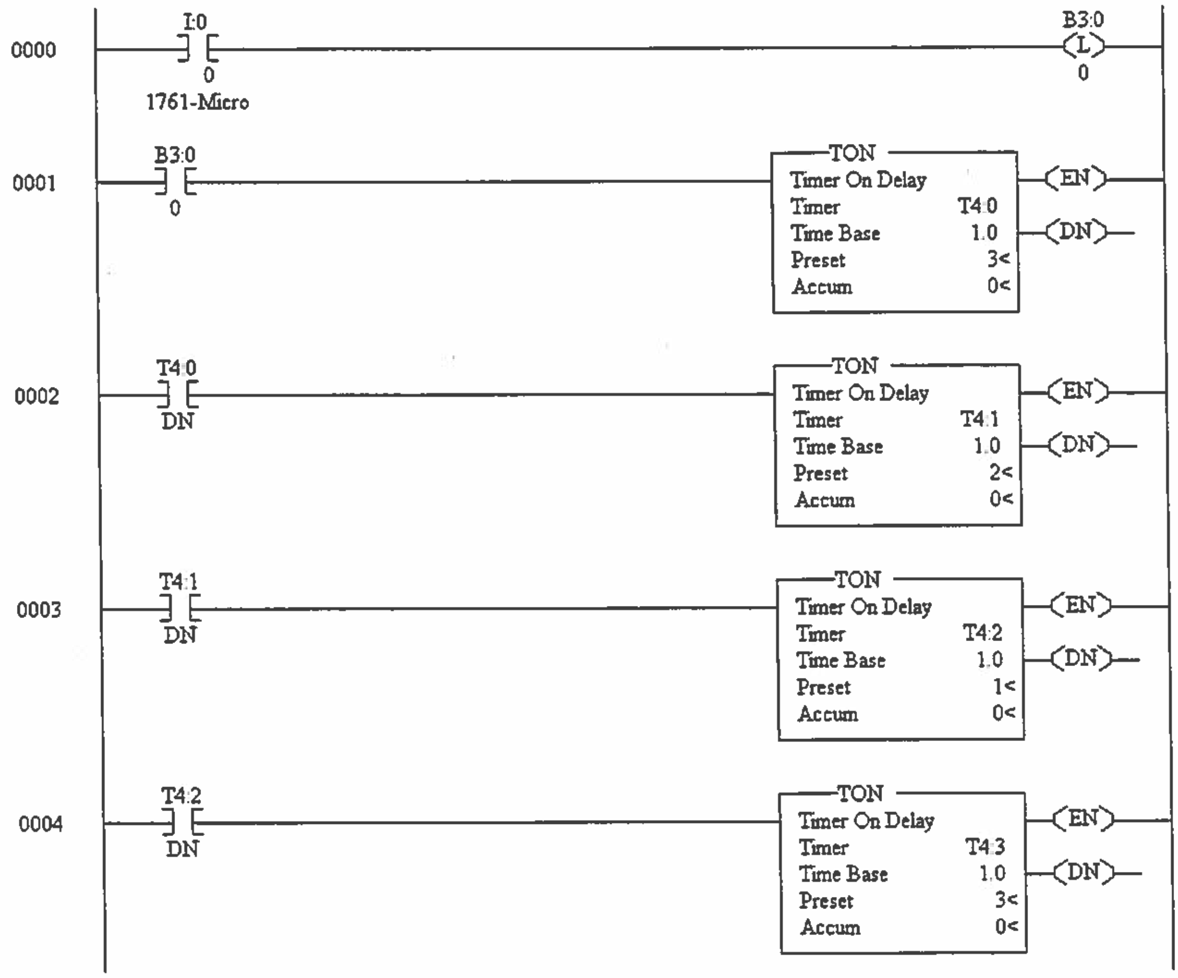


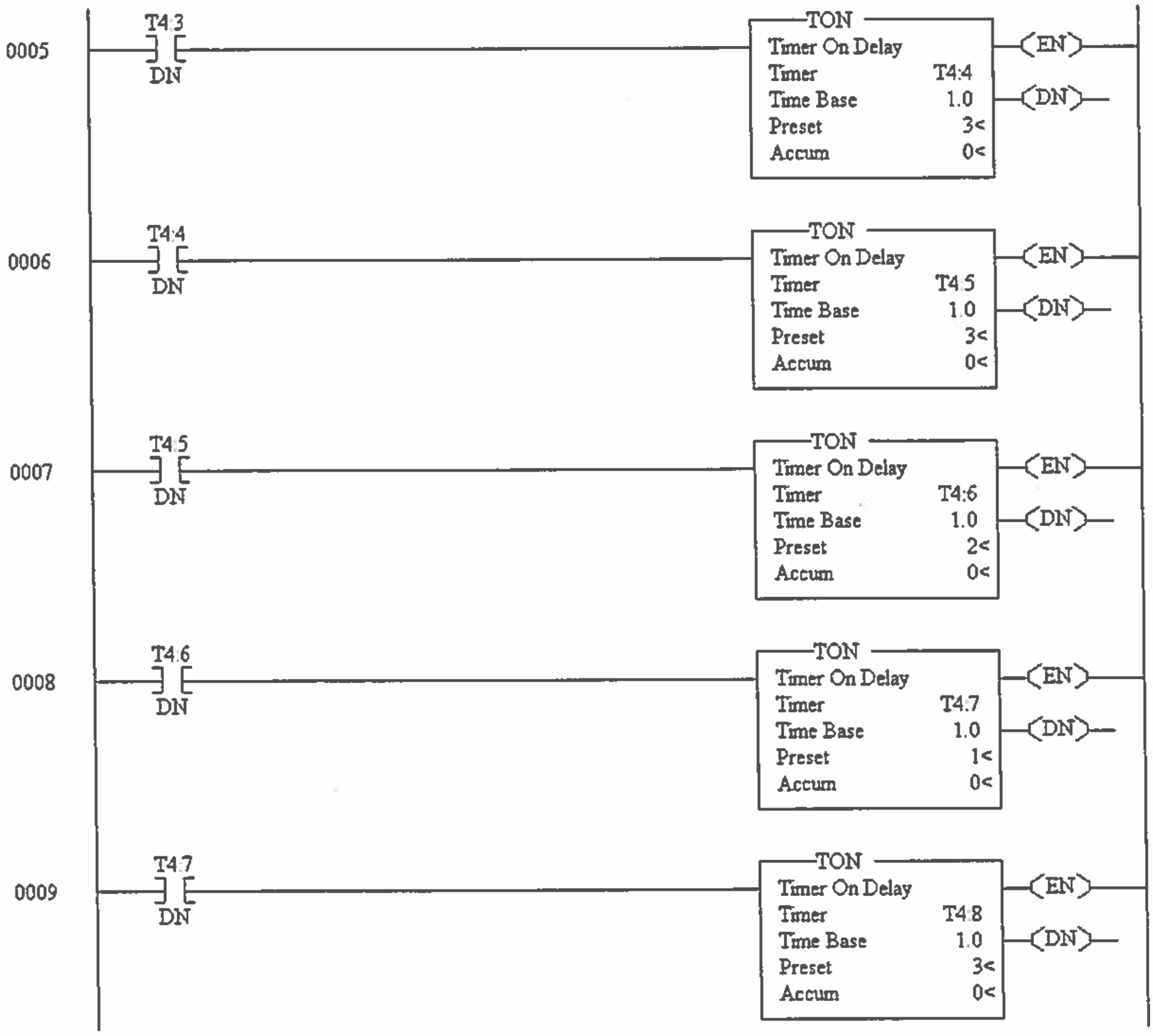
Figure 15-9
Motion Diagram for Step 1

Motion Step	Description	Time to Complete Move
1	Arm out	3.0 sec
2	Arm down	2.0 sec
3	Gripper close	1.0 sec
4	Arm up	3.0 sec
5	Arm in	3.0 sec
6	Arm right	3.0 sec
7	Arm down	2.0 sec
8	Gripper open	1.0 sec
9	Arm up	3.0 sec
10	Arm left	3.0 sec
	Total time	21.0 sec

Table 15-4
Time Listing

2. Using your timing listing, develop the timing section of the ladder logic diagram. Supply a momentary start switch to initiate the cycle and keep it running continuously. Use Figure 15-5 as a model.





(addresses may vary within their acceptable ranges)

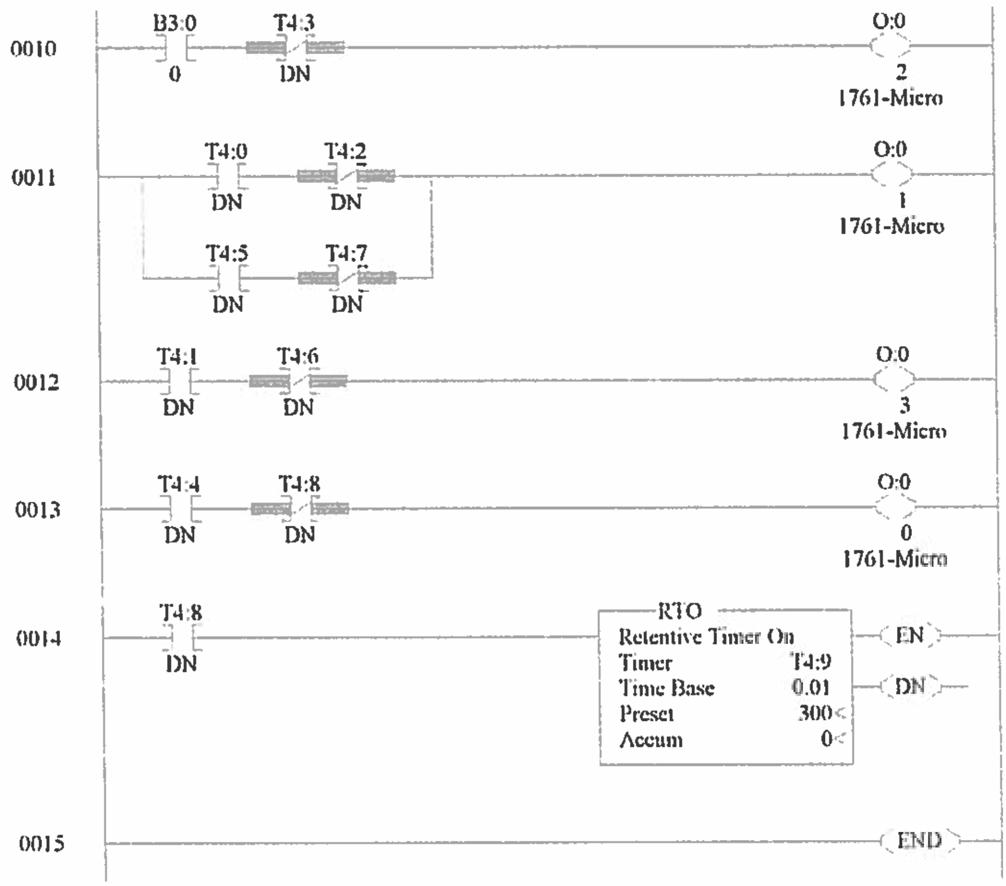
a. Which switch on the panel will you use? *0 or 1*

b. Momentary or maintain? *Momentary*

c. Explain your choice.

A normally open momentary switch is chosen to allow use of a latching relay with the program.

3. Use your timing and the value operation information in Table 15-5 to write the output section for running the valves. Remember the change in the in/out control. (Element addresses may vary within their acceptable ranges.)

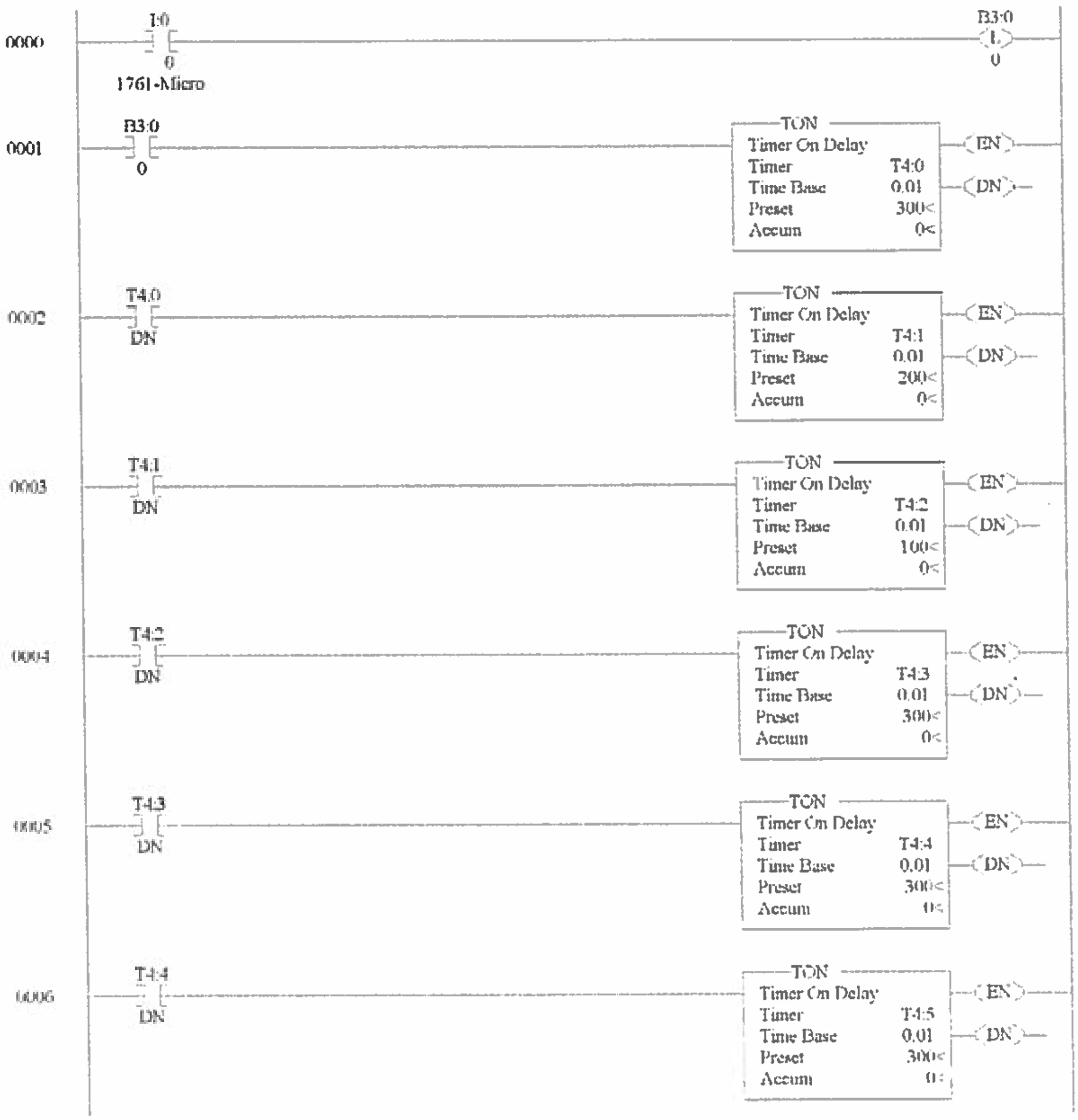


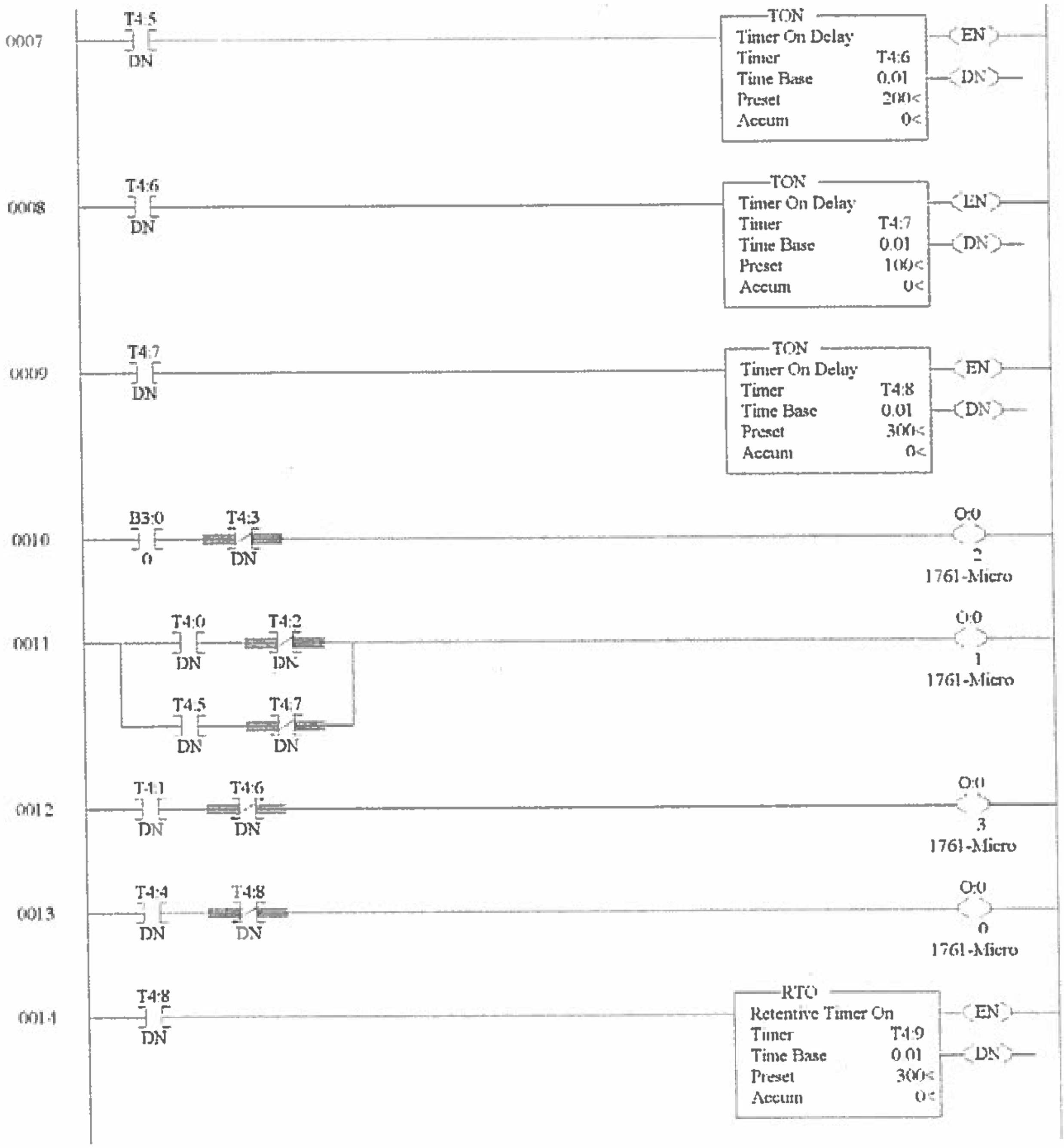
Valve number	On	Off	Output Number	Light Number
1	Arm right	Arm left	O:0/0	
2	Arm down	Arm up	O:0/1	
3	Arm out	Arm in	O:0/2	
4	Gripper close	Gripper open	O:0/3	

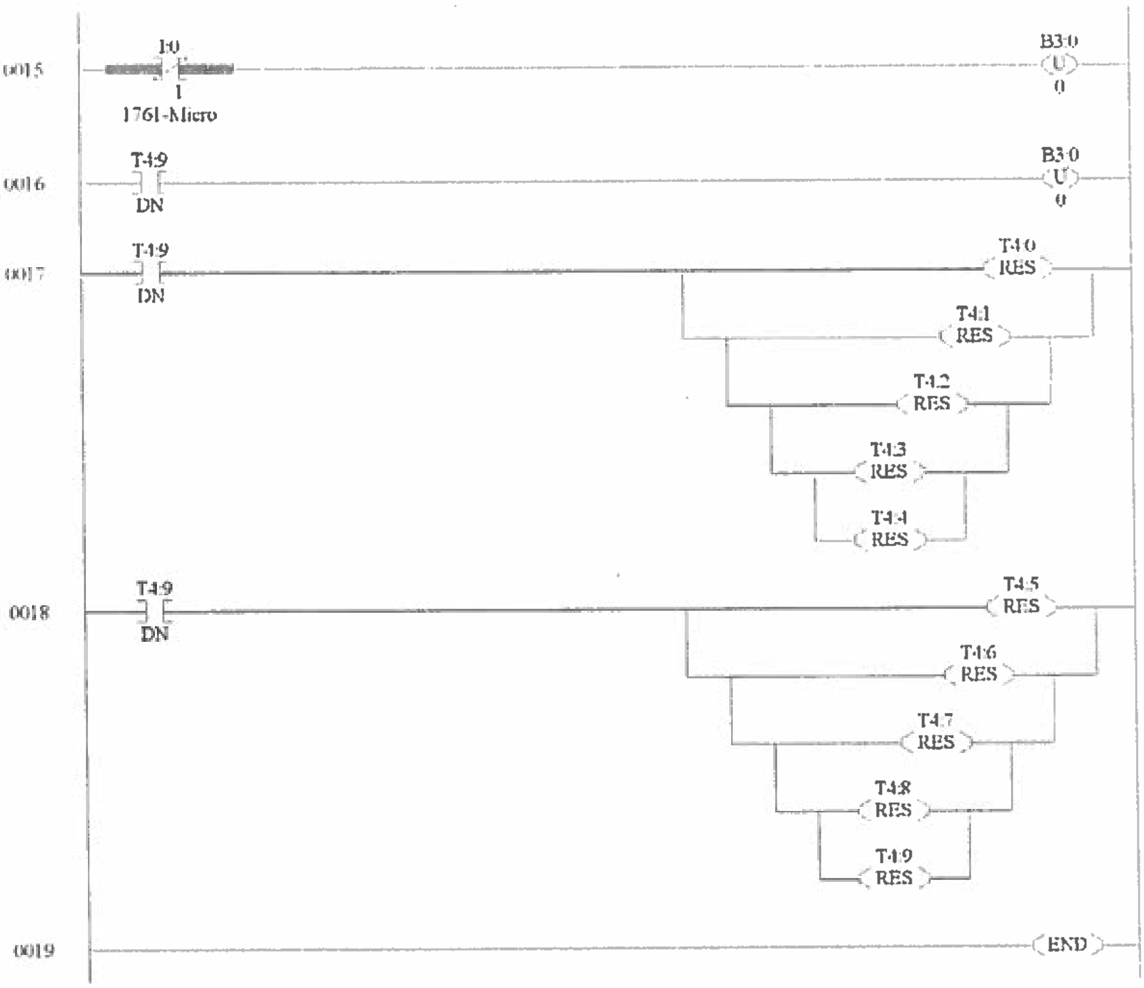
Table 15-5
Outputs for Robot Simulation

4. Select four (4) lights to simulate the valves. List the outputs and corresponding light numbers in Table 15-5. (using one light of each color should help distinguish motion equivalents.)

5. Make the timing cycle capable of continuous operation. Use a normally closed switch as a cycle stop button. This button when pressed should stop the motion of the arm, returning it to its home position. It must also reset all timers in the system. (Don't forget to modify all timing rungs to allow the reset of the timers.) Draw the entire program and list the ladder programming keystroke sequence. You may continue your answer onto the next pages.




0007
T4:5
DN
TON
Timer On Delay
Timer T4:6
Time Base 0.01
Preset 200<
Accum 0<
EN
DN
0008
T4:6
DN
TON
Timer On Delay
Timer T4:7
Time Base 0.01
Preset 100<
Accum 0<
EN
DN
0009
T4:7
DN
TON
Timer On Delay
Timer T4:8
Time Base 0.01
Preset 300<
Accum 0<
EN
DN
0010
B3:0
0
T4:3
DN
O:0
2
1761-Micro
0011
T4:0
DN
T4:2
DN
T4:5
DN
T4:7
DN
O:0
1
1761-Micro
0012
T4:1
DN
T4:6
DN
O:0
3
1761-Micro
0013
T4:4
DN
T4:8
DN
O:0
0
1761-Micro
0014
T4:8
DN
RTO
Retentive Timer On
Timer T4:9
Time Base 0.01
Preset 300<
Accum 0<
EN
DN


0015
I:0
1
1761-Micro
B3:0
U
0
0016
T4:9
DN
B3:0
U
0
0017
T4:9
DN
T4:0
RES
T4:1
RES
T4:2
RES
T4:3
RES
T4:4
RES
0018
T4:9
DN
T4:5
RES
T4:6
RES
T4:7
RES
T4:8
RES
T4:9
RES
0019
END

Which switch will you use for the stop button? Why?

Switch 2 or 3 because they are normally closed momentary switches. A momentary switch is sufficient to unlatch the latched relay.

6. Make all electrical connections needed for lights, switches and the controller.

7. Enter and test your program.

   a. What happened ?

If light 0 = left/right motion

light 2= up/down motion

light 4 = in/out motion

light 6 = gripper open/close

Then 0 -on 8.0 sec; off 14.0 sec

2- off 3.0 sec ; on 3.0 sec; off 8.0 sec; on 3.0 sec; off 6.0 sec

4- off 5.0 sec; on 11.0 sec; off 11.0 sec; off 6.0 sec

6- off 10.0 sec; on 8.0 sec; off 3.0 sec

(Substitute into the above listing the numbers of the lights which correspond to each motion. Then compare the student's times to the times given.)

   b. If the program did not operate as expected, why?

Answers will vary. The student should describe as fully as possible any errors he detects in his program.

c. How will you correct the errors? Use diagrams as needed.

Answers will vary. The student should describe some types of orderly process for identifying and correcting his programming errors.

8. Make all corrections in the program and test it again.

9. When you are satisfied that the program runs correctly, set aside the PLC and related hardware. Then complete the Questions which follow.

Questions

1. What is the fewest number of timers that could be cascaded to make a 24 hour timer? Explain.

The fewest number of timers would be three (3). 24 hours = 1,440 minutes and 86,400 seconds. Since the timer's preset value limit is 32,767 seconds. It would take a minimum of three cascaded timers to reach the required time.

2. What are the main reasons for cascading timers rather than starting them at all the same time?

It permits the timing cycle to be extended beyond the 323,767 second limit of a single timer's preset value. It is more difficult to monitor and debug PLC programs which contain a number of timers operating concurrently.

3. Describe the three sections of a cascaded non retentive timer program?

The first section contains all of the timers in sequence with the correct preset values. It also contains the initiating input and any latching or set relays which might be used for continuous operation. The second section contains all timer controlled outputs. The third section specifies the reset of all timers by a reset timer or second input. This section must also contain any latching relay or reset instruction for the set rung.

4. Explain what Figure 15-10 represents. Assume the robot arm is in a preset position with the gripper open.

Figure 15-11 is a robot arm motion diagram. The left hand portion is the motion diagram itself. The right hand portion is its legend. The motion diagram shows eight (8) sequenced steps in the following order: arm left, arm down, gripper close, arm up, arm right, arm in, gripper open, arm out.

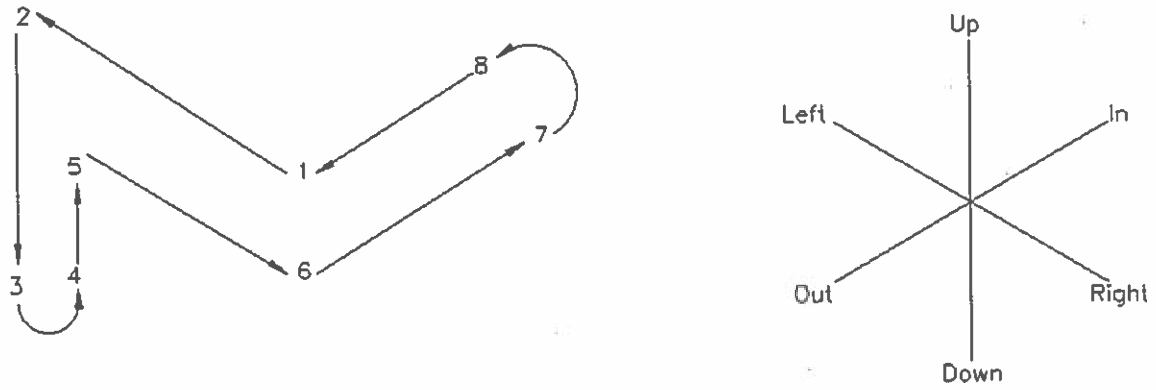


Figure 15-10

5. What are the differences between a motion diagram and a motion time listing?

The motion time listing table of all possible motions and the time it takes to execute each. The motion diagram is a graphic display of motions the robot must use for a specific purpose. The motion diagram specifies which motions will be used in what order they will occur.

UNIT 16
INSTRUCTOR'S MANUAL

TIMER/COUNTER PROGRAMS

1. Timer/counter programs are simply combinations of timers and counters. You should make certain that the students are comfortable with the operations of both counters and timers before they attempt this unit.

2. The students are asked to become more independent of specific supporting information as they work through the Experiment. They will need to use their editing and monitoring skills extensively. If programs malfunction, the students should be expected to troubleshoot their own program using the monitoring functions.

3. You may need to explain the 24-hour military clock to students who are unfamiliar with it. It is used in Step 7 of the Experiment.

4. One of the issues in timer/counter programming is designing a program which makes efficient use of the elements available. Question 2 of the Questions addresses this issue. The students may need assistance thinking about the alternatives and how to program them. There is more than one correct answer to the problem. The best answer is given in this manual, but you should accept others which attempt to solve the problem in a reasonable way.

UNIT 16

TIMER / COUNTER PROGRAMS

Objectives

Upon completions of this unit the student will be able to:

1. Describe the operation of a timer/counter program.
2. Explain the need for timer/counter programs.
3. Write a timer/counter program to meet given specifications.
4. Enter into the controller and monitor a timer/counter program.

Background

There are many events which occur on a timed basis in industrial controller applications. Typically, these timed events require more than 32,767 seconds (approximately 546.12 minutes), the limit of most PLC timers. These extended intervals can be obtained by cascading timers, but that uses up timers which may be needed to control outputs. Timer/counter combinations are used to reach the larger time intervals without using a lot of timers.

In a timer/counter program the counter counts specified units of time such as one unit for every 60 seconds timed. Each time the timer done relay is activated, the counter is incremented and the timer reset. When the counter reaches its preset value, its output is activated. A twenty-minute counter program activated by an external input is shown in Figure 16-1.

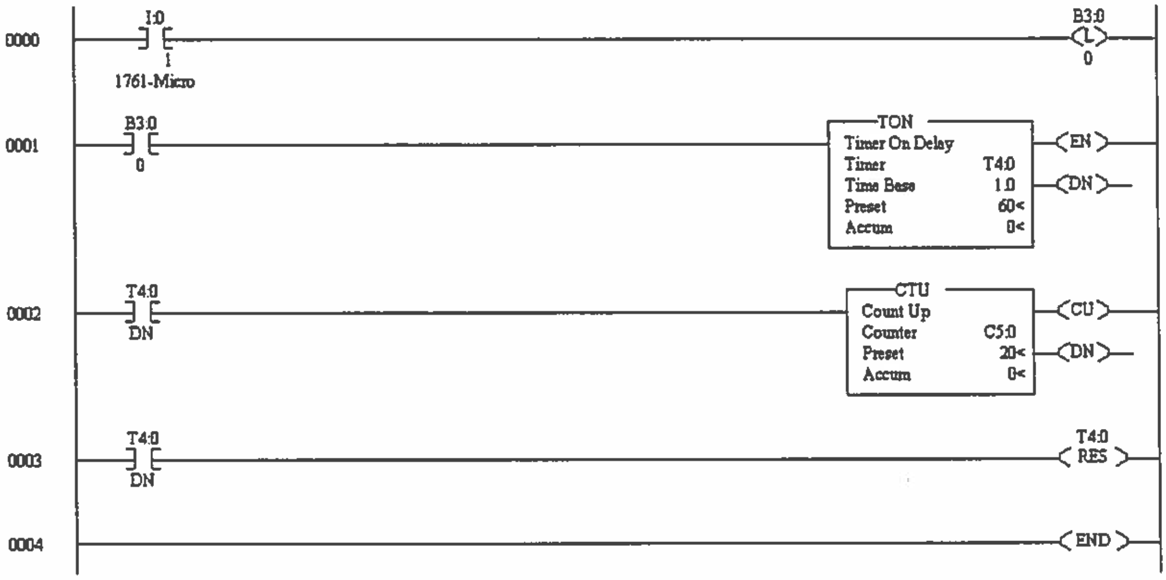


Figure 16-1
Twenty-Minute Timer/Counter Rungs

The rungs in Figure 16-1 do not make a complete program because the counter lacks a reset. There is no way to reset the counter and timer together without stopping the controller. A more complex 20-minute timer/counter program is given in Figure 16-2.

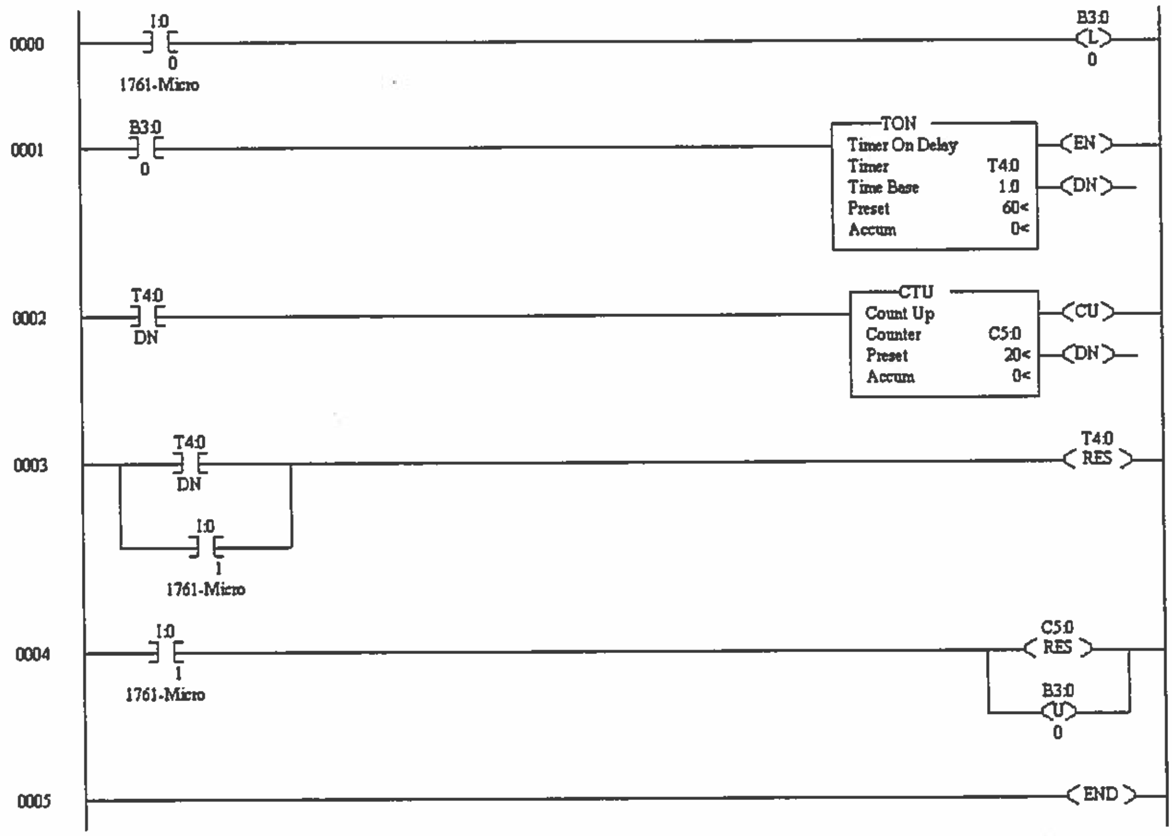


Figure 16-2
Twenty-Minute Timer/Counter Program

Input I:0/0, when energized, starts the program and keeps it running. Input I:0/1, when energized, resets both the timer and the counter and stops the program cycle. As long as I:0/0 remains energized, de-energizing I:0/1 restarts the program immediately. Input I:0/1 is able to effect the resets at any time during the program cycle.

Counters like timers, can cascade, creating the potential for very long timed cycles. Each counter acts as the input elements for the next counter in the cascade. And just as cascaded timers may reset themselves, cascaded counters may activate their own reset elements. The counters in the cascade can also activate different external outputs at different points in the program's cycle. Because the counters in this example are only activated for one program scan, the external outputs must be latched on if they are to remain on.

The timer/counter program in Figure 16-3 activates two external outputs. The first output, O:0/2 comes on after 20 minutes. Output O:0/2 latches itself on and stays on until the reset input I:0/0 is energized. The second output, O:0/4 is activated after 2 hours. Output O:0/4 cannot be turned off directly by an external input, but it does turn off when input I:0/0 is used to reset counter C5:5:1.

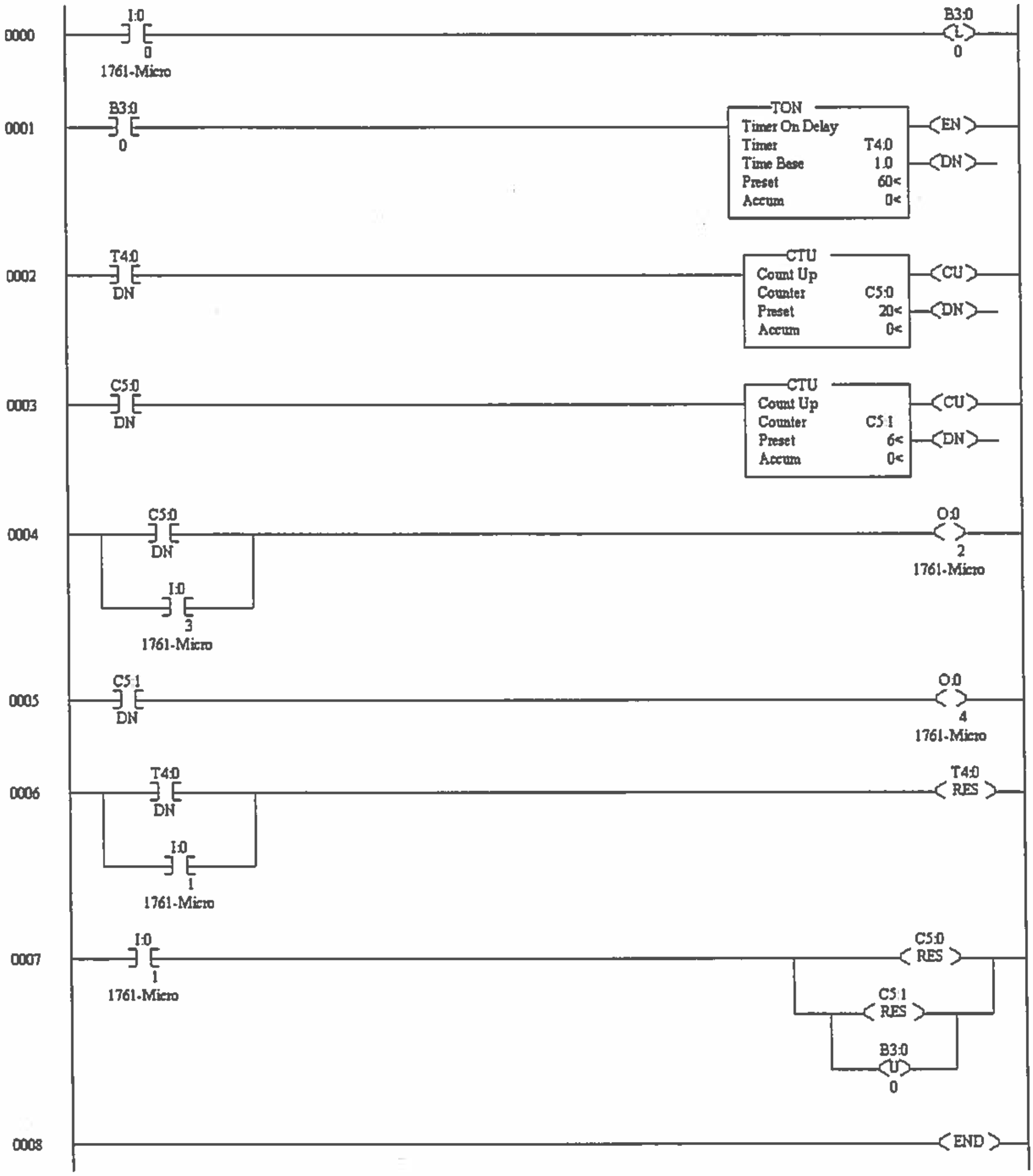


Figure 16-3
Timer/Counter Program with Cascaded Counters

Programming Keystrokes

The keystrokes used to enter timer/counter programs are the same as those used with individual timers and counters. The most important feature of timer/counter programs is how these timers and counters are programmed together. The timer must time an appropriate interval within its programming limits. The counter must be controlled by the timer and count the correct number of timed units. A separate input must be used to reset the counter once it has reached its preset value. The counter reset should also restart the timing cycle. Remember that the reset rungs must follow all rungs using the counters as inputs.

EXPERIMENT

Purpose

To develop, key in and monitor operation of a clock.

Procedure

1. Make all necessary electrical connections, including those given in Table 16-1. (No external outputs will be used during the experiment.)

Switch	Input	Function
0	1	Timer/start
1	2	Timer/counter Reset

Table 16-1
Connections for procedure

2. Examine the program in Figure 16-4. It is a simple timer/counter program which is activated by a normally open switch connected to input element I:0/0.

   a. What is the timing increment of the program? *1 second*

   b. What is the program's cycle time? *10 seconds*

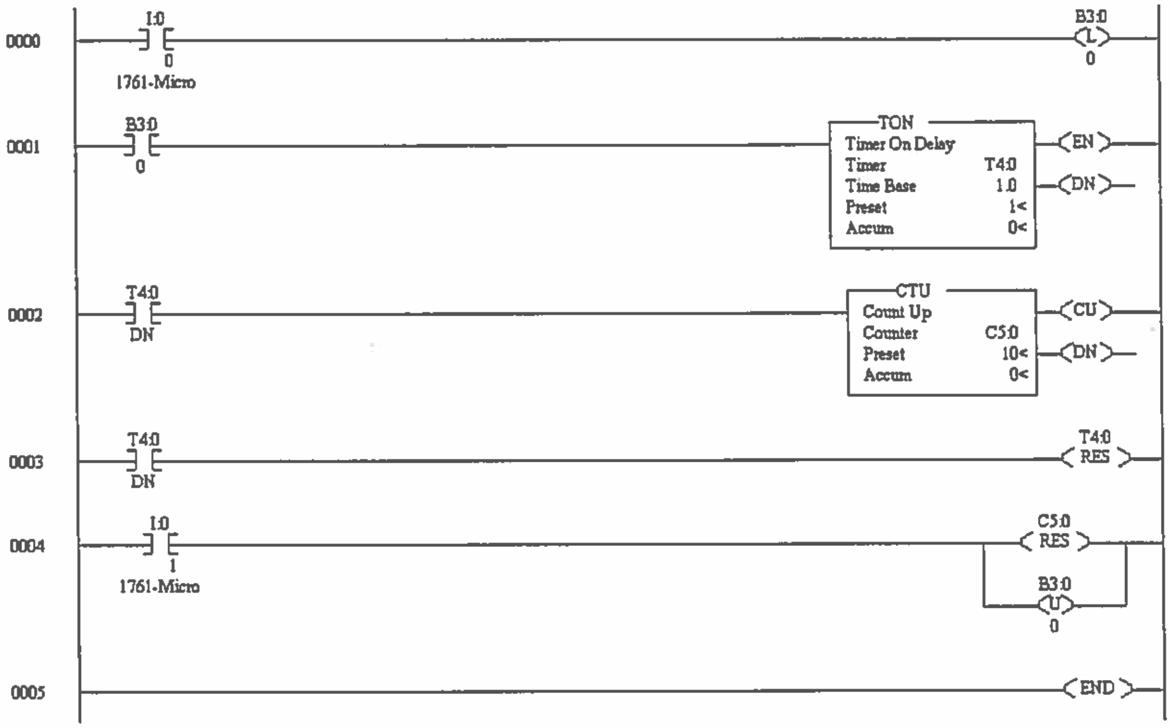
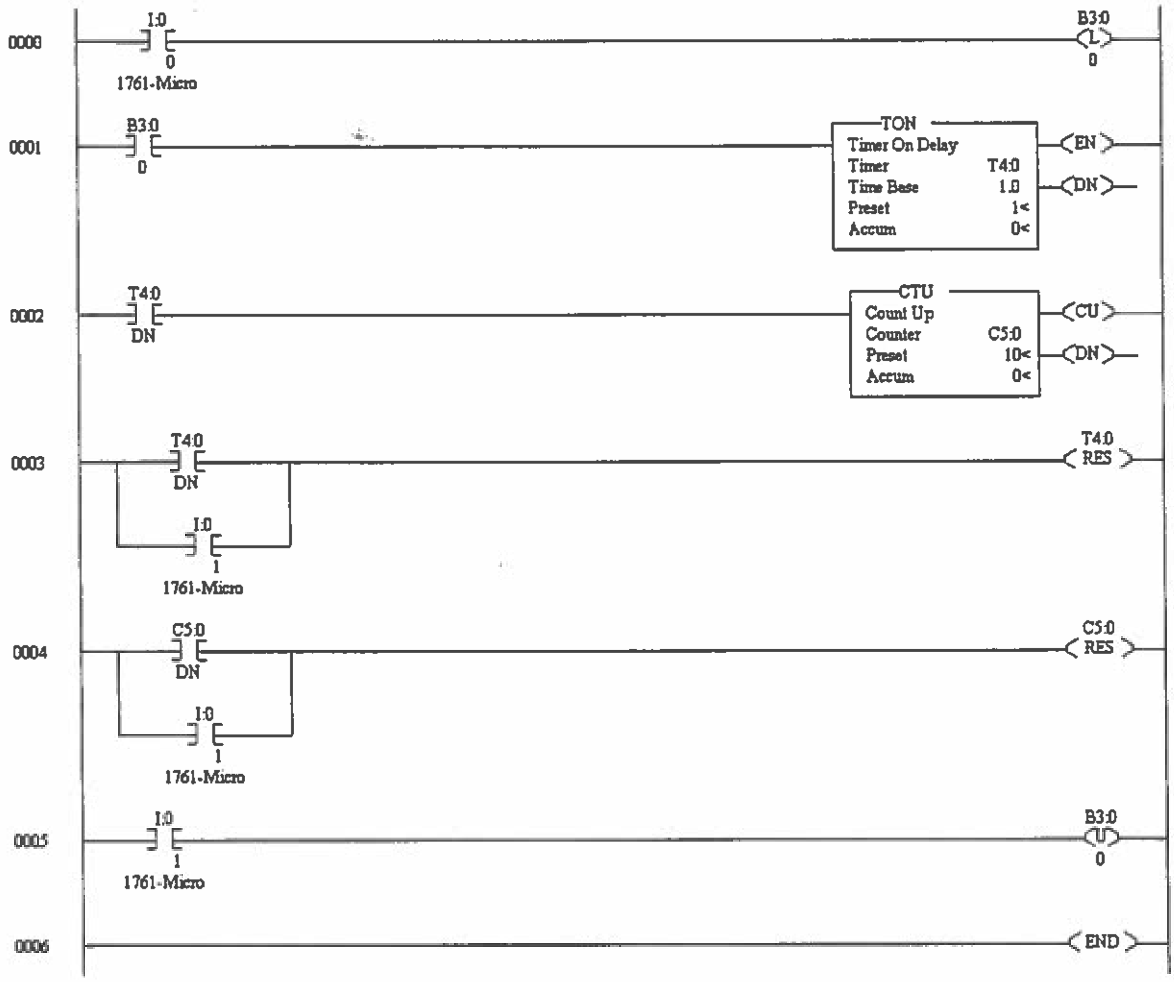


Figure 16-4
Program for Step 2

3. Enter the program and monitor the timer and counter. Run the program and note what you observe.

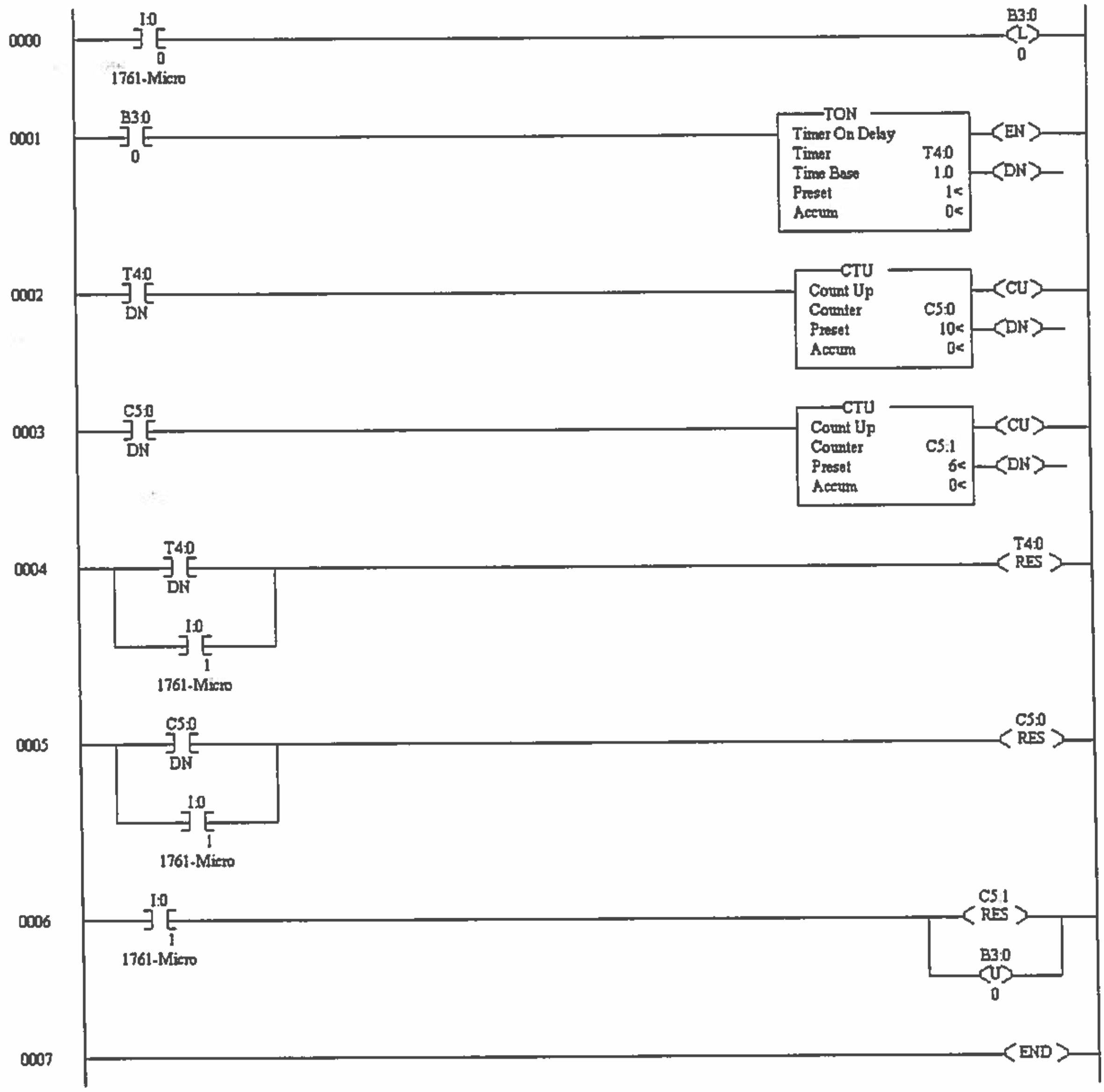
The timer operates only when switch 0 is being pressed. Because the timer is resetting every 1 second, it is not possible to see the accumulated value incrementing on the timer. The accumulated value of the counter increments one unit every 1 second. The counter will increment up to 10 units.

4. Modify the program so that a normally open switch connected to input I:0/1 can reset both the timer and the counter. Draw the complete ladder logic diagram in the space provided.

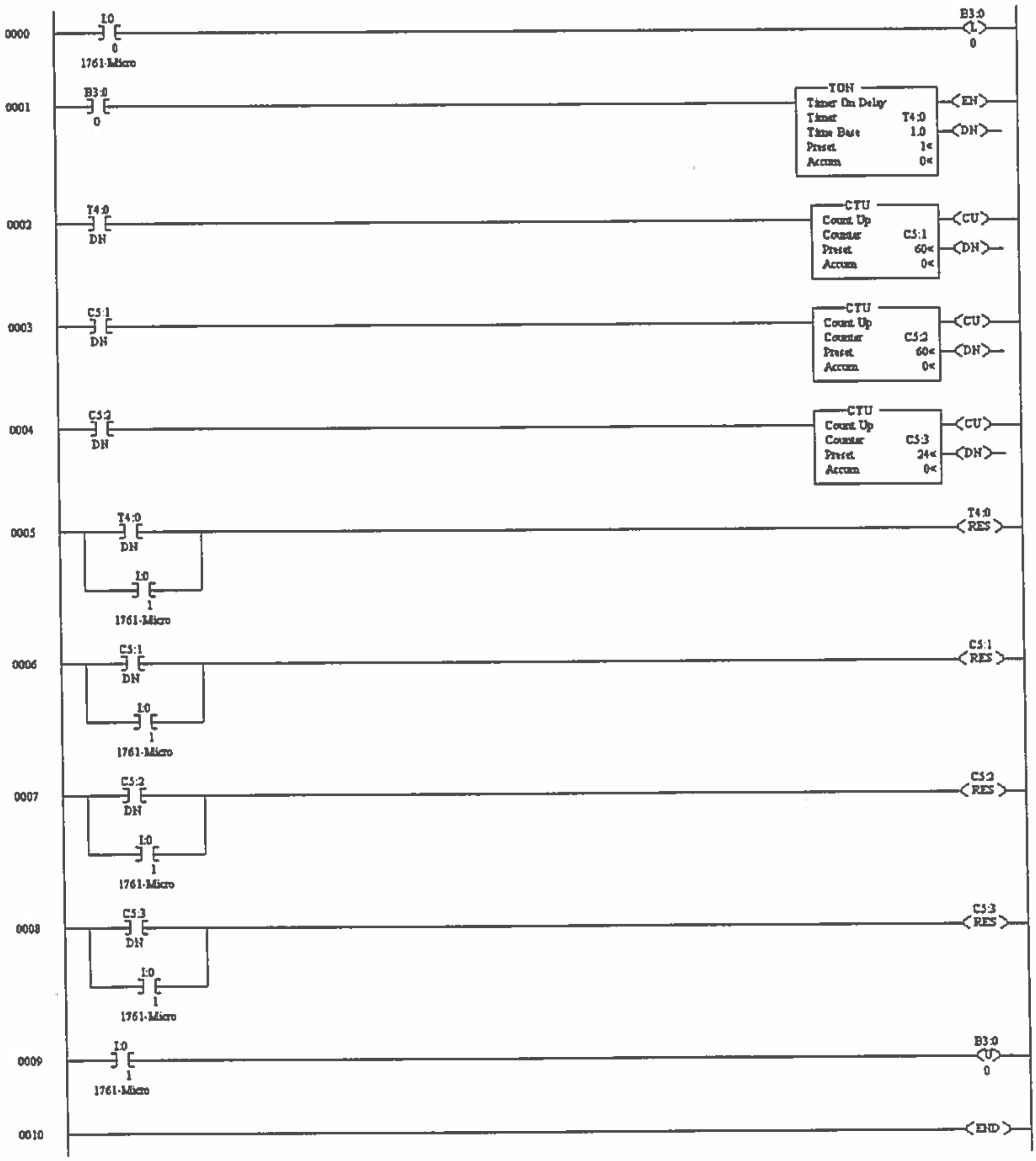


5. Enter, run and test your program design. Make any needed corrections in the program.

6. Further modify the program by cascading counters to make it capable of counting one minute and then stop. It must continue to reset with one external input. Refer to Figures 16-2 and 16-3 for guidance. Remember to consider the positions of the counter reset rungs. Draw the complete ladder logic diagram below. Then test your design and edit as needed.



7. Using the same programming principles, modify the program one more time to make it a 24 hour clock which counts hours military style (0-24). It must continue to reset completely by pressing a single external switch. Diagram the clock program below:



8. Enter the program as written and put the controller in RUN mode. Obviously, you cannot sit with the controller for 24 hours to test the program operation, so change the clock to time for only 2 hours. Use the edit keystrokes for rewriting a constant:

Use the arrow keys to the instruction parameter you want to monitor.
Counter address

9. Now run the program, monitoring its operation.

It is unnecessary to take two hours to test this program, so while the program is running, shorten its time still further. Change the minute counter to count 3 minutes by using the monitoring function. The keystrokes are:

Use the arrow keys to the instruction parameter you want to monitor.
Counter address

Now the program will run one complete cycle in six minutes. After the program has run a complete cycle further edit the program as needed.

10. Note your observations about the data available in monitor mode.

The student should describe his access to element status information and to the preset and accumulated values of counters and timers.

11. If you have difficulty making your 24 hour clock operational, consult with your instructor. Once you are satisfied you understand how to design a timer/counter program, answer the questions which follow.