UNIT IV

SONAR CIRCUITRY

CHAPTER 9

SONAR APPLICATIONS

OVERVIEW

In previous chapters, you learned the detailed operation of several systems with RB5X. You have also studied the expansion circuitry to convert analog signals to digital. The last area to be explored in this curriculum is RB5X's sonar circuitry.

Briefly, this circuitry loosely imitates the human sight sense. With its sonar activated, RB5X can move about the room and detect obstacles in its path.

In this chapter and the accompanying experiments, we study the simplified theory of the sonar-circuitry operation. This theory presentation is followed by several applications of the sonar for both RB5X and RobotLab.

OBJECTIVES

Upon completion of this material, you will be able to:

1. Understand the need for the sonar machine-code algorithm,

2. Understand the simplified principles behind sonar,

3. Understand the procedure to access the sonar machine-code algorithm with Tiny BASIC, and

4. Understand the procedure needed to use the sonar to activate the circuitry at the robot's I/O ports.

PREREQUISITES

1. An understanding of ASCII Pro.

2. An understanding of Tiny BASIC commands.

REFERENCES

None.

TECHNICAL INFORMATION

RB5X's sonar circuitry is similar in operation to radar circuitry. The sonar transmits a signal pulse of approximately 40k Hertz, pauses while it waits for the signal to reflect (or echo) off an object, measures the time interval between transmit and echo, and then transmits another pulse.

The interval between transmission and reception is interpreted as the distance the pulse has traveled, thus how far the object is from the robot's sonar "eye." (Keep in mind that this circuitry does not respond well to distances under one foot, because of the extremely short time needed for the sonar pulse to bounce off an object and return to the robot.)

Tiny BASIC programs run too slowly to accurately measure the time interval involved in sonar use, so the sonar routine must be implemented in machine code to be fast enough to measure the interval. The machine-code routine returns a value as variable D, which is proportional to the distance the pulse has traveled. Variable D can be accessed via Tiny BASIC programs, which must LINK to the machine-code routine and can then use the value stored in variable D to make RB5X respond to its environment.

Once the machine-code subroutine gains control, it initializes registers and I/O ports, commands the sonar to transmit, and begins to loop while waiting for the pulse to return. This loop includes a check

to see if the current loop count is still in the range of the sonar. If the subroutine finds that there is no object within sonar range, it reports back to the Tiny BASIC program with a -1 to variable D. If there is an object within range of the sonar, it returns a value to variable D that is proportional to the object's distance from the robot's "eye."

Appendix K, "Sonar-Ranging Subroutine Flow Chart," illustrates the algorithm used by the sonar subroutine. The Tiny BASIC program that follows accesses the sonar subroutine and turns on the flashing LEDs, if variable D contains a value between 140 and 150.

```
10 LET D=0:REM CLEARS VARIABLE D
20 LINK#1800:REM SONAR MACHINE-CODE ROUTINE
30 PRINT D:REM DISPLAYS PROPORTIONAL DISTANCE
40 IF D<100 PRINT"TOO CLOSE TO MEASURE"
50 REM CHECKS IF YOU ARE TOO CLOSE FOR
   ACCURATE MEASUREMENTS
60 IF D<140 GOTO 200:REM SETS MIN DISTANCE
70 IF D>150 GOTO 200:REM SETS MAX DISTANCE
80 @#7801=65:FOR I=1 TO 3:DELAY 1000:NEXT I:@#7801=0
90 REM FLASHING LEDS ON FOR 3 SECS IF D IS
   BETWEEN 141 AND 149
200 DELAY 1000
210 B=@#7800:REM CHECKS BUMPER SWITCHES
220 IF B<>215 GOTO 10:REM DETECTS BUMPER 1
230 STOP
```

Look over this program to see exactly how you would use it to cause the robot to respond to a specific distance.

SUMMARY

You should now have an understanding of the basic principles behind sonar, the need to implement the sonar routine in machine code, and how to use Tiny BASIC to access this subroutine. You should also understand how to use the returned value of the subroutine to activate I/O circuitry.

QUESTIONS

1. What two, separate sets of conditions return a value of -1 to variable D?

2. Why must you use machine code instead of Tiny BASIC to access the sonar circuitry?

3. If the sonar machine-code subroutine resides in addresses #2500 through #254E, what Tiny BASIC command should you used to access this subroutine?

4. The sonar circuitry is at address #7801. What bit switches this circuitry on and off?

5. Explain how a sonar signal measures distance.

Experiment 9.1

Displaying Sonar Measurements with the Computer

Overview

In this experiment, you practice using the sonar machine-code subroutine via Tiny BASIC -- specifically, you enter the machine-code subroutine as a series of Tiny BASIC POKE commands. This allows you to save the subroutine on your diskette and not have to re-enter it each time you switch on power to the robot after it has been switched off.

You develop, in this section, a Tiny BASIC program that allows you to access the sonar machine-code subroutine and to display on the computer CRT a value that is proportional to the distance of the obstacle from the sonar "eye."

Objectives

Upon completion of this experiment, you will be able to:

1. Use the POKE command to load the sonar machine-code subroutine into RAM,

2. Use Tiny BASIC programs to access the sonar machine-code subroutine, and

3. Use Tiny BASIC programs that return a decimal value proportional to the distance of the obstacle from the sonar "eye."

Prerequisites

1. Chapter 9, "Sonar Applications."

References

1. Section D, "Software," subsection on "Sonar Machine-Code Algorithm," in the RB5X Reference Manual.

Technical Information

RB5X's sonar control circuitry occupies one bit at address #7801 and two bits at address #7802. The following table illustrates the bit functions.

Address	Bit	Function	Active Level
7801	D-0	Sonar on/off	Active high
7802	D-5	Sonar transmit	Active low
7802	D-7	Sonar echo	Active high

Table 9.1 - Sonar Control Circuitry Bit Function

The machine-code subroutine uses these address bits to measure the time between transmission of the sonar pulse and reception of the echo. The value returned is always stored in Tiny BASIC variable D, which is accessible via Tiny BASIC in the same way as any other Tiny BASIC variable. If the sonar transducer does not receive an echo, the variable is set to a -1 value. This can happen under any or all of the following circumstances:

1. The nearest object is out of range. (The subroutine has transmitted another signal before the echo has had time to return to the transducer.)

2. The object is right next to the sonar transducer. (The echo returns before the circuitry is ready to receive it.)

Distances greater than 0 and less than approximately one foot return a constant value of approximately 95. The sonar circuitry was not designed to operate at distances less than one foot.

The sonar card connects to the robot via J13 on the right front of the mother board. The sonar transducer connects to the sonar card via two wires; these wires must not be reversed or the transducer does not operate.

NOTE: Avoid all physical contact with the sonar card and transducer while they are in operation, because there are short duration voltage pulses well in excess of 100 volts in the circuitry.

The programming needed to operate the sonar circuitry must be in machine code to meet its speed requirements. The machine-code program is loaded into the robot's RAM memory via Tiny BASIC POKE commands (for example, @#1800=#56). The machine-code program used in this experiment resides at addresses #1800 through #1840, although this program can be relocated in memory. (For information on relocating it, refer to Section D, "Software," of the RB5X Reference Manual.)

Once the machine-language program is loaded into memory, use the command NEW#2000<RET> followed by NEW<RET> instead of the standard NEW#1000 followed by NEW. The command NEW#2000 clears all memory above #2000. If you use NEW#1000, you clear the machine-code POKEs you just entered.

Procedure

1. The sonar board was connected at the factory, but make sure that all connections are secure.

2. Set up the robot (or RobotLab) for Tiny BASIC.

3. Position the transducer so that it is pointing toward an open space.

4. Enter the following series of Tiny BASIC POKE commands:

```
 4 @#7803=#98
 5 @#1800=#56
10 @#1801=#26
```

```
 15 @#1802=#00
 20 @#1803=#78
 25 @#1804=#27
 30 @#1805=#00
 35 @#1806=#18
 40 @#1807=#00
 45 @#1808=#00
 50 @#1809=#00
 55 @#180A=#00
 60 @#180B=#84
 65 @#180C=#00
 70 @#180D=#00
 75 @#180E=#09
 80 @#180F=#C4
 85 @#1810=#01
 90 @#1811=#CA
 95 @#1812=#01
100 @#1813=#C2
105 @#1814=#02
110 @#1815=#D4
115 @#1816=#20
120 @#1817=#6F
125 @#1818=#12
130 @#1819=#0B
135 @#181A=#B4
140 @#181B=#01
145 @#181C=#00
150 @#181D=#09
155 @#181E=#C2
160 @#181F=#02
165 @#1820=#D4
170 @#1821=#80
175 @#1822=#7F
180 @#1823=#37
185 @#1824=#0B
190 @#1825=#BC
195 @#1826=#30
200 @#1827=#03
205 @#1828=#06
210 @#1829=#D4
215 @#182A=#80
220 @#182B=#6F
225 @#182C=#18
230 @#182D=#C4
235 @#182E=#00
240 @#182F=#CA
245 @#18[illegible]=#01
```

```
250 @#1831=#84
255 @#1832=#FF
260 @#1833=#FF
265 @#1834=#5E
270 @#1835=#8A
275 @#1836=#06
280 @#1837=#5C
285 @#1838=#C4
290 @#1839=#00
295 @#183A=#CA
300 @#183B=#01
305 @#183C=#5E
310 @#183D=#0B
315 @#183E=#8A
320 @#183F=#06
325 @#1840=#5C
```

5. Check the program for errors and then save it on your diskette as SONAR DEMO.

6. Run the program. After the program executes, it returns a Tiny BASIC prompt >, indicating that it has loaded the sonar machine-code subroutine into RAM.

7. At the Tiny BASIC prompt >, type NEW#2000<RET> followed by NEW<RET>.

8. Enter the following Tiny BASIC program.

```
20 LINK#1800:REM ACCESSES SONAR SUBROUTINE
50 DELAY 1000
60 B=@#7800:REM READS BUMPERS
70 IF B<>251 GOTO 20:REM IF BUMPER 1 STOP
80 STOP
```

9. Run the program. You should hear coming from the sonar transducer a faint ticking sound, which is produced each time the sonar circuitry transmits a pulse.

10. Plug the RobotLab charger into a wall socket or push RB5X into its charger-nest. Notice that the sonar stops transmitting. This is because the charger sense bit changed and the sonar machine-code program is set to disable the sonar circuitry during charging. Unplug the RobotLab charger or move RB5X away from the charger-nest, and the program continues as usual.

11. Press bumper 1 to stop the ticking and the program execution.

12. To receive a decimal value proportional to the distance, add the following line to your program:

    30 PRINT D

13. Run the program.

14. The decimal numbers printed on the CRT are proportional to the distances traveled by the sonar wave.

15. Place your hand over the sonar transducer, being careful not to touch the connectors. The value on the CRT should be -1, which indicates that the object (your hand) is too close for the circuits to respond.

16. Position your hand six inches away from the transducer. The CRT should return a decimal value of something close to 95.

17. Position your hand one foot away from the transducer. It should still return a value close to 95.

18. Position your hand 18 inches from the transducer. The decimal value on the CRT should increase to approximately 106.

19. Position your hand at random distances from the transducer. Note that the value displayed on the CRT is proportional to the distance between your hand and the sonar transducer.

20. Press bumper 1 to terminate the routine.

21. Press the software reset button and allow the robot to initialize. Press bumpers 2 and 4 simultaneously to return to command mode.

22. Type in the following immediate execution command:

    B=@#1826:PRINT B

23. Switch the robot off.

24. Reinitialize the robot and return to command mode.

25. Type in the same immediate execution command as in Step 22.

26. The CRT should return a 0. (From the preceding series of steps, you should see that pressing the software reset button erases Tiny

BASIC programs, but has no effect on the machine-code routine. Switching off power erases all RAM memory.)

Summary

During this experiment, you gained experience in loading the sonar machine-code subroutine into RAM. You should now be familiar with the Tiny BASIC commands needed to activate the sonar and to return a decimal number to the CRT that is proportional to the distance traveled by the sonar pulse. You should also have realized that switching off the robot's power clears the machine-code routine from RAM, but that the pressing the software reset button only clears Tiny BASIC programs from memory.

Questions

1. If the SONAR DEMO program has been loaded into RAM, the following command returns what decimal value to the CRT?

   B=@#182D:PRINT D

   _____

2. What decimal value is returned when your hand is placed two feet from the sonar transducer?

   _____

3. What is the highest decimal value that the system returns? (Try positioning RB5X at one end of a long, clear hallway.)

   _____

4. What safety precaution should be observed when using the sonar?

   _____

5. Why does the sonar stop operation when the charger is plugged into a wall socket (RobotLab) or when the robot is pushed into the charger-nest (RB5X)?

_____

Experiment 9.2

Responding to Sonar Measurements

Overview

In Experiment 9.1, you learned how to use the POKE command to load the sonar machine-code subroutine into RAM and how to access the subroutine through software to return a decimal value proportional to the distance the sonar pulse travels.

In this experiment, you use this knowledge to activate the robot to respond to these returned values. For a robot to function in our human environment, it must be able to "see" and to respond to what it "sees." At present, RB5X's technology simply allows it to determine how far away an object is, but does not enable it to distinguish features of the object. Even with this limited capability, however, RB5X is capable of elaborate responses to what it "sees."

Objectives

Upon completion of this experiment, you will be able to:

1. Plot a graph of returned decimal values versus distance, and

2. Activate the system-defined ports in response to specific distances.

Prerequisites

1. Experiment 9.1, "Displaying Sonar Measurements with the Computer."

2. Understanding of the system-defined ports and their addresses.

References

1. Appendix G, "Functions."

Technical Information

From Appendix 11, "Functions," you are able to determine the addresses and the bits needed to activate the system-defined ports. This activation includes flashing LEDs, a fixed LED, the horn, and relays.

The sonar subroutine can be accessed via Tiny BASIC; variable D contains the distance of the object from the sonar transducer. This value can be used with Tiny BASIC to trigger any or all of the responses that RB5X is capable of demonstrating. As an example, if D=100, then @#7801=128 causes the horn to turn on when the sonar routine senses a distance that returns a decimal 100.

Procedure

1. Set up RB5X or RobotLab for Tiny BASIC.

2. Load the sonar machine-code subroutine into RAM, remembering to use NEW#2000 instead of NEW#1000.

3. Write a simple Tiny BASIC program to display the variable D from the sonar. (HINT: Check Experiment 9.1 for help.)

4. Run your program and plot a graph of distance from the sonar transducer versus the value of variable D. Use six-inch increments on the horizontal axis and variable D on the vertical axis. Start variable D at 90 with increments of 5 up to a reading of 250.

5. Note that after distances of approximately 18 inches, the response

becomes very linear. It may be necessary to take several readings at each distance and to average them. This can be done via the following Tiny BASIC commands.

```
FOR I=1 TO 5
LINK#1800
DELAY 1000
LET C=C+D
NEXT I
PRINT"THE AVERAGE IS ",C/5
```

6. Write a Tiny BASIC program to switch on the flashing LEDs when the sonar returns a value of 130. Remember that the subroutine clears all ports to 0 when it is linked. Therefore, a delay is needed after you turn on the LEDs to view the flashing action. Record your program below.

7. Enter the following Tiny BASIC program to activate the horn when the sonar subroutine returns a value between 3-1/2 and 4 feet.

```
10 CLEAR
20 LINK#1800:REM SONAR ROUTINE CALL
30 PRINT D:REM DISPLAYS PROPORTIONAL DISTANCE
40 IF D>150 GOSUB 200:REM CHECKS LOWER DISTANCE
   LIMIT
50 B=@#77800:IF B<>251 GOTO 10:REM IF BUMPER 1
   STOP
60 STOP
```

```
200 IF D>165 RETURN:REM CHECKS UPPER DISTANCE
    LIMIT
210 @#7801=128:DELAY 250:REM HORN ON FOR 1/4
    SECOND
220 RETURN
```

8. Run the program and observe that, when your hand is between approximately 3-1/2 and 4 feet from the sonar transducer, the horn sounds.

9. Write a program that switches on the two, front, green LEDs for three seconds when there is an obstacle between 3-1/2 and 4 feet from the sonar transducer and that switches on the two, rear, green LEDs for three seconds when the obstacle is between 2 and 2-1/2 feet from the sonar transducer. (HINT: @#7802=9 switches on the two, front, green LEDs.) Record your program below.

Summary

During this experiment, you plotted a graph to correlate the decimal number and the distance in inches of the sonar transducer from an obstacle. You also learned how to activate the system-defined ports when the obstacle is a specific distance from the transducer. RB5X now has rudimentary sight and the ability to respond to what it "sees."

Questions

1. What is the decimal number returned for a distance of four feet?

2. Write a Tiny BASIC program to switch on the flashing LEDs for one second when there is an obstacle six feet away from the sonar transducer.

3. Write a Tiny BASIC program to beep the horn twice and flash the LEDs for one second when the distance measured is between 4-1/2 and 5 feet.

4. What response would RB5X have to the following command?

   @#7802=5

5. Write a Tiny BASIC program to turn on the horn for one beep when there is an obstacle two feet away. Write the program so that it can be executed with the RS-232 cable disconnected. (HINT: Let one of the first statements in your program check the bumpers to continue.)

Experiment 9.3

RB5X's Response to Sonar

Overview

You have, to this point in Unit IV, learned how to:

1. Activate the sonar,
2. Return a decimal value proportional to the distance of an object from the robot's sonar transducer, and
3. Use the decimal value returned to trigger specific responses in RB5X.

Objectives

Upon completion of this experiment, you will be able to:

1. Cause RB5X to respond to bumper switch inputs, and
2. Cause RB5X to respond to the returned value of the sonar.

Prerequisites

1. Chapter 9, "Sonar Applications."
2. Experiment 9.1, "Displaying Sonar Measurements with the Computer."
3. Experiment 9.2, "Responding to Sonar Measurements."

References

1. Appendix G, "Functions."

Technical Information

There is no new technical information needed to understand this experiment. As you need to, please review any or all of the preceding chapters and experiments.

Procedure

1. Place RB5X on the floor so that it has room to maneuver.
2. Switch on power to the robot and allow it to initialize.
3. Check the battery-level indicator on the interface panel and be sure that the robot is fully charged.
4. Make sure the utility software module is in the socket, and press bumpers 2 and 5 simultaneously. This calls up a utility software subroutine that uses the sonar and bumpers to sense its location and to move about the floor turning to avoid obstacles.
5. When you are familiar with the robot's responses, press the software reset button to stop the routine and to reinitialize the robot.
6. Switch the robot off, connect the RS-232 cable to Port 1 on RB5X and set the robot up for Tiny BASIC.
7. Load the sonar machine-code subroutine, which you previously stored on diskette, into RAM. Remember to use the NEW#2000 (instead of NEW#1000) and NEW commands.

8. Enter the following Tiny BASIC program:

```
  10 B=@#7800:IF B<>190 GOTO 10:REM DELAYS START
     OF PROGRAM TIL BUMPERS 2 AND 5 ARE PRESSED
  20 CLEAR
  30 LINK#1800:REM ACTIVATES SONAR
  40 GOSUB 1000:REM CHECK BUMPERS TO STOP
  50 IF D>118 GOSUB 1050:REM CHECKS LOWER LIMIT
     VALUE
 999 GOTO 30
1000 B=@#7800:IF B<>251 RETURN:REM STOP IF
     BUMPER 1 IS PRESSED
1010 STOP
1050 IF D<123 GOTO 1070:REM CHECKS UPPER SONAR
LIMIT
1060 RETURN
1070 @#7802=9:FOR I=1 TO 4:DELAY 1000:NEXT I:REM
     FORWARD FOR 4 SECONDS
1080 @#7802=0:REM TURNS FORWARD MOTORS OFF
1090 RETURN
```

This program allows the robot to move forward if the sonar returns a value between 118 and 123. When you run the program, press bumpers 2 and 6; this gives you time to remove the RS-232 cable from the robot's interface panel, allowing RB5X more freedom to move.

9. Save the program on diskette, then type RUN<RET>.

10. Disconnect the RS-232 cable from the robot.

11. Press bumpers 2 and 6 to start the routine. The robot moves forward for four seconds when the sonar returns a value between 2 and 2-1/4 feet. It does not respond to other values.

12. Verify the program's operation.

13. Press bumper 1 to stop the routine.

14. Reconnect the RS-232 cable to the robot and LIST the program on your CRT.

15. Add the following lines to the program you entered in Step 8:

```
  60 IF D>140 GOSUB 2000:REM CHECKS LOWER LIMIT
2000 IF D<146 GOTO 2020:REM CHECKS UPPER LIMIT
2010 RETURN
```

```
2020 @#7802=5:FOR I=1 TO 3:DELAY 500:NEXT I:REM 180
     DEGREE TURN
2030 @#7802=0:REM TURNS MOTORS OFF
2040 RETURN
```

16. Save the expanded program on diskette and then type RUN<RET>.

17. Disconnect the RS-232 cable from the robot.

18. Press bumpers 2 and 6 simultaneously to start the routine. In addition to the previous responses, the robot turns 90 degrees for sonar values between 3 and 3-1/4 feet.

19. Verify the program's operation.

20. Press bumper 1 to stop the routine.

21. Reconnect the RS-232 cable to the robot and LIST the program.

22. Add the following lines to your program:

```
  70 IF D>163 GOSUB 2100:REM CHECKS LOWER LIMIT
2100 IF D<168 GOTO 2120:REM CHECKS UPPER LIMIT
2110 RETURN
2120 @#7802=6:FOR I=1 TO 4:DELAY 1000:NEXT I:REM
     BACK UP FOR 4 SECONDS
2130 @#7802=0:REM TURNS MOTORS OFF
2140 RETURN
```

23. Save the expanded program on diskette, then type RUN<RET>.

24. Disconnect the RS-232 cable from the robot.

25. Press bumpers 2 and 4 simultaneously to start the routine. In addition to the previous responses, the robot now backs up for four seconds for sonar values of 4 to 4-1/4 feet.

26. Verify the program's operation.

27. Press bumper 1 to stop the routine.

28. Reconnect the RS-232 cable to the robot and LIST the program.

29. Add the following lines to your program:

```
  80 IF D>184 GOSUB 2200:REM CHECKS LOWER LIMIT
2200 IF D<190 GOTO 2220:REM CHECKS UPPER LIMIT
```

```
2210 RETURN
2220 @#7802=10:FOR I=1 TO 3:DELAY 500:NEXT I:REM
     TURN 90 DEGREES
2230 2#7802=0:REM TURN OFF MOTORS
2240 RETURN
```

30. Save the expanded program on diskette, and then type RUN<RET>.

31. Disconnect the RS-232 cable from the robot.

32. Verify the program's operation.

33. Press bumpers 2 and 6 simultaneously to start the routine. In addition to the previous responses, what does the robot do and at what distance does the obstacle have to be to cause this response?

34. Press bumper 1 to stop the routine.

35. Reconnect the RS-232 cable and LIST this program.

36. At line 90, insert a program line that will GOSUB 3050 when a sonar value less than 18 inches is returned. Record this line below.

37. Beginning at line 3050, insert the line or lines needed to enable the robot to rotate 180 degrees to the left and to return to the sonar check portion of the program.

38. Save this expanded program on diskette, and then type RUN<RET>.

39. Disconnect the RS-232 cable from the robot.

40. Verify the program's operation.

41. Press bumper 1 to stop the program and then switch the robot off.

Summary

During this experiment, you observed RB5X in its interactive state. You were able to use the bumpers and the sonar to determine the routine executed by the robot. You are now able to use an assortment of physical phenomena as inputs to the robot to cause a wide variety of output responses. Your use of RB5X is now limited only by your own imagination!

Questions

1. Amend the final version of the program in this experiment so that at the end of each subroutine the horn sounds twice for 1/4-second each beep.

2. Write a Tiny BASIC program so that the robot responds with the following actions in the following order:

   A) Bumper 1 - Move toward rear
   B) Bumper 3 - Turn right
   C) Bumper 5 - Move forward
   D) Bumper 7 - Turn left

3. Add the line or lines to the program you wrote in Question 2 so that bumper 2 stops the robot.

_____

_____

_____

_____