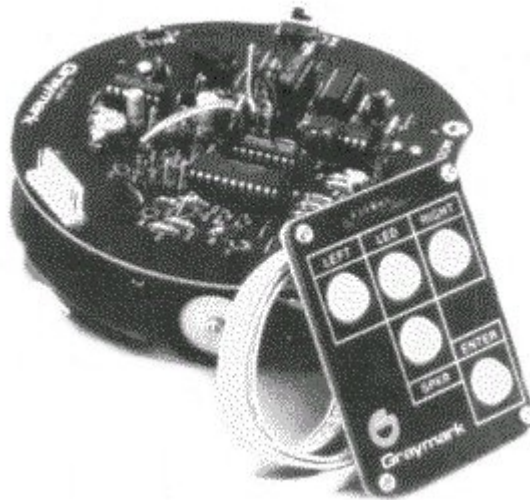


603A Digital Programmable Robot



By:
George A. Jackson

ITEC 497
Fall 1999

Supervised By:
Dr. G.H. Massiha

Table of Contents

Table of Contents	2
1.0 Abstract	3
2.0 Introduction	4
3.0 Operation	5
3.1 Program Mode	5
3.2 Run Mode	7
3.3 Clock Frequency Adjustment	8
4.0 Troubleshooting	8
5.0 Conclusion	10
6.0 Quiz	12
7.0 Program Instructions	14
7.1 Teach Pendant Program 1	14
7.2 Teach Pendant Program 2	14
7.3 Computer Parallel Interface Program 1	15
7.4 Computer Parallel Interface Program 2 - BASIC INPUT	16
7.5 Computer Parallel Interface Program 3 – Course Tracing Program	16
7.6 Computer Serial Interface Program 1 – Basic Input	18
7.7 Computer Serial Interface Program 2 – Course Tracing Program	19
Appendix	21
Power Source:	21
Serial Interface BAUD Rate:	21
Parity:	21
Number of STOP bits:	22
Number of Bits:	22

1.0 Abstract

The purpose of this project is to assemble the Graymark 603A Digital Programmable Robot. It is a kit that takes you through the steps of assembling the robot, programming the robot, and also troubleshooting the robot if there are any problems with the robot. Through this it teaches the fundamentals of circuit design and construction, along with showing how to construct and use logic circuits.

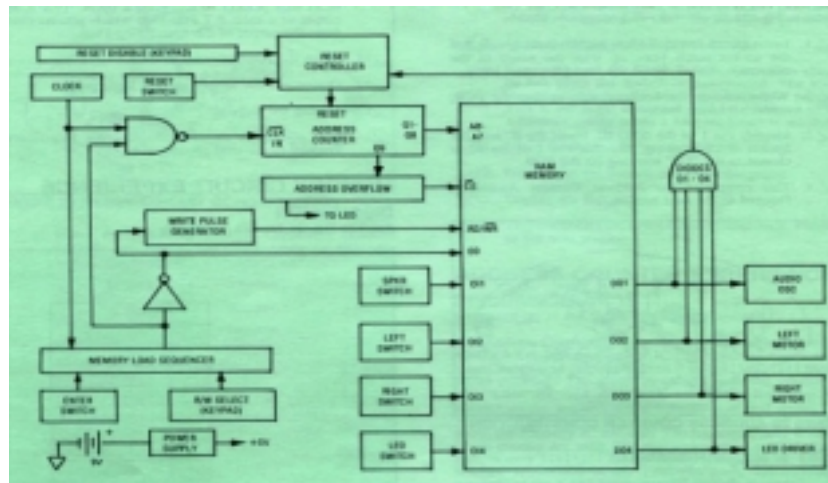
2.0 Introduction

The purpose of this project is to assemble the Graymark 603A Digital Programmable Robot. The robot is a kit designed to teach the user some basic fundamentals of digital electronics and some basic robotics. The 603A incorporates many of the basic elements of robot technology. It is a motorized vehicle with a memory connected to a logic circuit that can run a course that is programmed.

“The term robot was first used in a fictional play by the Czechoslovakian playwright Karl Cakap. It was derived from the Czech word “robota” meaning slave labor. Since then, the term robot has been applied to a wide variety of fictional and real machines that perform tasks normally requiring a human.”

By constructing the robot, I learned about robot technology, including principles of digital electronics, memory, and logic circuits. All of these elements are necessary for computer operation. By programming the robot, it demonstrates how computers store and execute instructions, as well as how robots can perform a sequence of events based on data stored in memory.

The heart of the 603A robot is the memory circuit. The memory circuit is where the keypad commands are memorized and stored for execution at a later time. “The memory device is a 256-word by 4-bit CMOS Static Random-Access Memory IC. This IC contains an array of 1024 memory cells that are arranged into 256 addressed locations that will handle 4 data bits of information per location.”



The kit is divided into sections for construction. Each section is a circuit for the robot. The instruction manual gives a schematic for each circuit, the instruction steps to assemble the circuit, and a test procedure to test the function of the circuit. A short explanation of the theory of the circuit operation is given at the beginning of each section.

3.0 Operation

3.1 Program Mode

In this section, I will describe how to program the robot.

1. The robot can be programmed using the keypad. This section will explain how to use the keypad supplied with the kit. See diagram 1-1.



Fig. 1-2 (keypad)

- a. Press and hold down the desired instruction button (LEFT, RIGHT, LED, SPKR).
 - b. Press and release the ENTER button.
 - c. Release the instruction button.
2. Be sure that Good batteries are installed.
3. Move power switch SW1 to off.
4. Plug the keypad into header P1.
5. Move power switch SW1 to ON, and press RESET.
6. LED instruction.

Press and hold the LED button. Press and release the ENTER button.

Release the LED button. The LED should light for about one second.

7. SPKR instruction.

Press and hold the SPKR button. Press and release the ENTER button.

Release the SPKR button. You should hear a high pitched tone.

8. TURN RIGHT Instruction.

Press and hold the RIGHT button. Press and release the ENTER button.

Release the RIGHT button. The right motor should run and the robot should turn right.

9. TURN LEFT Instruction.

Press and hold the LEFT button. Press and release the ENTER button.

Release the LEFT button. The left motor should run and the robot should turn left.

10. GO FORWARD instruction.

Press and hold both the LEFT and RIGHT buttons. Press and release the ENTER button. Release the LEFT and RIGHT buttons. Both the left and right motors should run and the robot should go forward.

11. PAUSE instruction.

Press and release the ENTER button. This instructs the robot to pause for one step before continuing.

12. RESET instruction.

Press and hold the LEFT, RIGHT, LED, and SPKR buttons. Press and release the ENTER button. This is the PROGRAM RESET or replay instruction. This instructs the robot to repeat all the instructions you have entered.

13. If more than 255 instructions are entered, the LED will light and stay lit. This is a signal that the robot has run out of memory. Programming can be started over by pressing the RESET button SW2, or the keypad can be unplugged and the robot can execute the programmed instructions.

3.2 Run Mode

In this section, I will show how to have the robot execute (run) the program.

1. While leaving the power switch, SW1, ON, disconnect the keypad from the robot.
2. Press the Reset Switch, SW2, to reset the memory to instruction #1. When the Reset Switch is released, the robot will start to execute the instructions that were programmed. The robot will run through the instructions until it is turned off or the keypad is plugged in.

3. The memory can be reset to instruction #1 at any time by pressing the Reset Switch, SW2.

3.3 Clock Frequency Adjustment

1. Potentiometer VR adjusts the Clock frequency. The Clock rate determines the execution time of each instruction.

Fully clockwise = slowest rate (longest time)

Fully counter-clockwise = fastest rate (shortest time)
2. When execution time is decreased, the motors run for a shorter time, which means that a turn instruction will not make the robot turn as far as when the execution time is greater. This allows more control because it allows the robot to turn different amounts. However, the total run time will also be affected because more instructions are executed each second. Experiment with different VR settings to determine which are best.
3. If the potentiometer is fully counter-clockwise, the Clock frequency will be at its maximum rate. The instructions will be executed at a faster rate, and for a shorter time.
4. If the potentiometer is fully clockwise, the Clock frequency will be at its minimum rate. The instructions will be executed at a slower rate, and for a longer time.

4.0 Troubleshooting

One of the most important things to be able to do when working on electrical circuits is knowing how to troubleshoot. The ability to troubleshoot will

get you a long way and save you many headaches. There are three basic steps that must be followed when servicing any electronic item. The first is to identify the defective section. Second, identify the defective component or connection. Third, repair or replace the defective component or connection.

I was given the task of troubleshooting a Graymark 603A Digital Programmable Robot that was constructed by another student. The robot was inoperable. The first step I tried, while attempting to troubleshoot, was to install new batteries in the robot and test it again. This is usually the best place to start when troubleshooting. This attempt failed. You then need to start troubleshooting the components of the unit.

When using the following troubleshooting procedures, perform each step before proceeding to the next step. If there is a section that is in need of repair, repair and test that section before proceeding on to the next step or section. When testing a section, remove any test equipment and resolder any unsoldered parts. If the robot is unable to operate properly, the test has failed. Go back to the step that was just tested and proceed to check all the components of the section thoroughly. Before starting the service procedures, make certain you understand clearly how the robot operates.

I removed the electronic circuit section of the robot from the motor drive section. Upon inspection of the circuit board I noticed some suspicious looking areas in the soldered connections and on some of the ribbon tracings of the printed circuit board. There were some solder connections that appeared to have been heated too much. There were some ribbon tracings that were lifting

from the printed circuit board. These are some good indications that there will be problems with the circuits on the board.

Next, check the placement of all the components. Be sure that the leads of all transistors are in the correct holes. Make sure the flat sides of all transistors are lined up with the PCB outlines. Be sure none of the transistor leads are crossed or touching another lead. Check the leads of all other components to be sure they are properly inserted into the holes, and are not broken or touching other components. Make sure that the ICs are installed facing the proper direction, and that none of the leads are bent, broken, or shorted.

If none of these steps gets the robot working, then you will have to test each section of the robot, and each component. It will be necessary to find the broken or bad component of the robot to get it working.

I went through steps troubleshooting the robot. I went through many components, but was unable to get the robot in operable condition. One of the primary components to start with was the voltage regulator circuit. With this part of the robot not working, it is impossible for the robot to work properly if at all.

5.0 Conclusion

This project took me through the steps of constructing the Graymark 603A Digital Programmable Robot. I learned the difference in using schematic and block diagrams. I learned some basic circuit design, and logic circuit construction. I also learned and put into practical use some of the basic

elements of robot technology. In working with the robot that was inoperable, I also learned some of the basic steps in troubleshooting.

6.0 Quiz

- | | | |
|--------------------------|--------------------------|---|
| T | <input type="checkbox"/> | 1. Robots became practical only after the introduction of the computer to robot technology. |
| <input type="checkbox"/> | F | 2. In an inverter circuit, if a signal at the input goes from low to high, the signal at the output will go from high to low. |
| T | <input type="checkbox"/> | 3. Because of a battery's internal resistance, the greater the current drain from the battery, the higher the output voltage. |
| T | <input type="checkbox"/> | 4. The term robot was invented by an American computer manufacturer. |
| <input type="checkbox"/> | F | 5. A digital circuit has two states, On and Off. |
| <input type="checkbox"/> | F | 6. Robots can often perform a job more economically than a person. |
| T | <input type="checkbox"/> | 7. Because robots are so expensive, they are not used to perform jobs that would be too dangerous for a person. |
| <input type="checkbox"/> | F | 8. Like humans, to perform a task, robots may make use of the five senses. |
| <input type="checkbox"/> | F | 9. A robot can make a decision based upon information it has received from its sensors. |
| T | <input type="checkbox"/> | 10. Robots never need humans to teach them how to perform their tasks. |
| T | <input type="checkbox"/> | 11. A Logic Block is what happens when a person has trouble making a logical decision. |
| <input type="checkbox"/> | F | 12. A computer stores information in its Memory. |

1. A series regulator keeps the voltage constant, even though the current draw may fluctuate.
2. A slide rule is an example of a manual computer.
3. An electronic switch used in digital circuits is called a logic gate.
4. If the input to a Buffer is HIGH, the buffer's output will be HIGH.
5. If one input of a two input OR gate is high and the other is low, the output will be HIGH.
6. If one input of a two input NAND gate is high and the other is low, the output will be HIGH.
7. If both inputs to an NAND gate are high, the output will be LOW.

8. A NOR gate is an OR gate with an inverter added to the output.
9. When resistors and capacitors are used to set the operating frequency, the frequency is determined from the RC time constant.
10. A counter made from flip-flops counts using binary arithmetic.
11. A block of bits in a memory circuit can be accessed by referring to its address.
12. The reset line on a counter sets the counter to zero.
13. A 3-stage binary counter can count to 8. A 4-stage binary counter can count to sixteen.
14. A table which shows possible input conditions and the associated output condition is called a truth table.
15. A circuit which provides system timing pulses is called a clock circuit.
16. Rather than being an Astable circuit, a flip-flop is a stable.

7.0 Program Instructions

These are some programming instructions using the Teach Pendant.

7.1 Teach Pendant Program 1

1. Forward
2. Forward
3. Forward
4. Forward, LED
5. Forward, LED
6. Forward, LED
7. Forward, Speaker
8. Forward, Speaker
9. Forward, Speaker
10. Right Turn
11. Forward
12. Forward
13. Forward
14. Forward
15. Forward
16. Forward
17. Right Turn
18. Right Turn
19. Forward, LED
20. Forward, LED
21. Forward, LED
22. Forward, LED
23. Forward, LED
24. Forward, LED
25. Forward, LED
26. Forward, LED
27. Forward, LED
28. Forward, LED
29. Forward, LED
30. Reset

7.2 Teach Pendant Program 2

1. Speaker, LED
2. Speaker, LED
3. Speaker, LED
4. Forward
5. Forward

6. Forward
7. Forward
8. Left Turn, LED
9. Left Turn, LED
10. Forward
11. Forward
12. Forward, LED
13. Forward
14. Forward, LED
15. Forward
16. Forward, LED
17. Forward
18. Left Turn, Speaker
19. Left Turn, Speaker
20. Left Turn, Speaker
21. Left Turn, Speaker
22. Forward
23. Forward
24. Forward, LED
25. Forward, Speaker
26. Forward, LED
27. Forward, Speaker
28. Forward, LED
29. Forward, Speaker
30. Forward, LED
31. Reset

The following are the programming instructions when using the Graymark Model 613 Parallel Interface or the 623 Serial Interface. These must be typed into a BASIC program editor, such as Qbasic, in order to be executed.

7.3 Computer Parallel Interface Program 1

```

1  REM **GEORGE JACKSON**
2  REM **ITEC 497 FALL 1999**
3  REM **DR. G.H MASSIHA**
10 INPUT A
20 B = A + 64
30 LPRINT CHR$(B);
40 GOTO 10

```

Press the power switch on the interface. The interface should be ON. Turn the robot power switch ON. Press the RESET TOUCH SWITCH on the robot. Start the program by typing "RUN" on your computer.

At the first "?" prompt, type the number "1" followed by a RETURN. After an instant the LED on the robot should light up.

Next enter "2". The beeper should sound.

Next enter "4". The robot should turn left.

Next enter "8". The robot should turn right.

Each of these commands enters an instruction into the robot's memory in the same manner as the teach pendant. To play the commands back, unplug the interface connector from the robot and press the RESET TOUCH SWITCH.

7.4 Computer Parallel Interface Program 2 - BASIC INPUT

This program accepts numbers directly from the keyboard which correspond directly to the bit patterns which activate the various commands in the robot. Entering the number 1 (binary 0001) makes the LED light. Entering number 2 (binary 0010) sounds the beeper. Number 4 (binary 0100) makes the robot turn right, and number 8 (binary 1000) makes it turn left. Entering numbers which correspond to more than one bit being set will send multiple commands at the same time. For example number 3 (binary 0011) will beep the beeper and light the LED. Number 12 (binary 1100) will operate both the left and right motors, making the robot go straight.

```
1  REM **GEORGE JACKSON**
2  REM **ITEC 497 FALL 1999**
3  REM **DR. G.H MASSIHA**
10 REM **NUMERICAL ENTRY ROBOT INTERFACE PROGRAM**
20 PRINT "ENTER INSTRUCTION NUMBER"
30 PRINT "1 = LIGHT LED"
40 PRINT "2 = SOUND BEEPER"
50 PRINT "4 = RIGHT MOTOR"
60 PRINT "8 = LEFT MOTOR"
65 PRINT "12 = STRAIGHT"
70 INPUT X
80 Y = X + 64
90 LPRINT CHR$(Y);
100 GOTO 1
```

7.5 Computer Parallel Interface Program 3 – Course Tracing Program

This program allows you to enter a letter corresponding to the command to be given to the robot. (EXAMPLE: The letter F = FORWARD). The computer will then trace the intended route for the robot on the computer's CRT screen, by

placing the instruction letter at each point as the command is entered.

To keep track of the robot's course, the program stores the robot's heading in the variable DIR (Direction). There are 8 possible headings, 1 through 8. As the robot makes a turn, DIR is either incremented (Right Turn) or decremented (Left Turn), thereby allowing the computer to keep track of the robot's intended navigation (lines 260-280). The computer then executes the proper cursor control statements based upon the heading stored in DIR. Lines 300 through 310 correct DIR if the calculated value is less than 1 or greater than 8.

```
1  REM **GEORGE JACKSON**
2  REM **ITEC 497 FALL 1999**
3  REM **DR. G.H MASSIHA**
10 REM ****ROBOT COURSE TRACING PROGRAM****
20 REM
90 CLS
100 PRINT "F = FORWARD"
110 PRINT "R = TURN RIGHT"
120 PRINT "L = TURN LEFT"
130 PRINT "B = SOUND BEEPER"
140 PRINT "X = LIGHT LED"
150 FOR S = 1 TO 16
160 PRINT
170 NEXT S
180 PRINT TAB(40); "X";
190 '
200 DIR = 1
210 '
220 '
230 A$ = INKEY$
240 IF A$ = "" THEN 230
250 '
260 IF A$ = "F" THEN DIR = DIR
270 IF A$ = "R" THEN DIR = DIR + 1
280 IF A$ = "L" THEN DIR = DIR - 1
290 '
300 IF DIR > 8 THEN DIR = DIR - 8
310 IF DIR < 1 THEN DIR = 8 - DIR
320 '
400 IF DIR = 1 THEN P$ = CHR$(29) + CHR$(30): REM CURSOR UP
410 IF DIR = 2 THEN P$ = CHR$(30) + CHR$(28): REM CURSOR UP,
    RIGHT
420 IF DIR = 3 THEN P$ = CHR$(28): REM CURSOR RIGHT
430 IF DIR = 4 THEN P$ = CHR$(31) + CHR$(28): REM CURSOR DOWN,
    RIGHT
440 IF DIR = 5 THEN P$ = CHR$(29) + CHR$(31): REM CURSOR DOWN
```

```

450 IF DIR = 6 THEN P$ = CHR$(29) + CHR$(29) + CHR$(29) + CHR$(31):
    REM CURSOR DOWN, LEFT
460 IF DIR = 7 THEN P$ = CHR$(29) + CHR$(29) + CHR$(29): REM
    CURSOR LEFT
470 IF DIR = 8 THEN P$ = CHR$(29) + CHR$(29) + CHR$(29) + CHR$(30):
    REM CURSOR UP, LEFT
480 IF A$ = "B" THEN P$ = CHR$(29)
490 IF A$ = "X" THEN P$ = CHR$(29)
500 '
510 PRINT P$; A$;
520 '
530 '
550 IF A$ = "F" THEN Z = 12: REM BINARY 1100 (BOTH MOTORS)
560 IF A$ = "R" THEN Z = 4: REM BINARY 1000 (LEFT MOTOR)
570 IF A$ = "L" THEN Z = 8: REM BINARY 0100 (RIGHT MOTOR)
580 IF A$ = "B" THEN Z = 2: REM BINARY 0010 (SOUND BEEPER)
590 IF A$ = "X" THEN Z = 1: REM BINARY 0001 (LIGHT LED)
600 '
610 W = Z + 64
620 LPRINT CHR$(W); : REM SEND TO ROBOT
630 '
640 GOTO 230

```

7.6 Computer Serial Interface Program 1 – Basic Input

Press the power switch on the interface. The interface should be ON. Turn the robot power switch ON. Press the RESET TOUCH SWITCH on the robot. Start the program by typing "RUN" on your computer.

At the first "?" prompt, type the number "1" followed by a RETURN. After an instant the LED on the robot should light up.

Next enter "2". The beeper should sound.

Next enter "4". The robot should turn left.

Next enter "8". The robot should turn right.

Each of these commands enters an instruction into the robot's memory in the same manner as the teach pendant. To play the commands back, unplug the interface connector from the robot and press the RESET TOUCH SWITCH.

```

10 REM ***SERIAL INTERFACE TEST PROGRAM***
15 REM ***IBM PC AND COMPATIBLES ONLY***
20 REM ***300 BAUD, NO PARITY, 8 BITS***
30 REM

```

```

40 OPEN "COM1:300,N,8" FOR RANDOM AS #1
50 INPUT A
60 PRINT #1, CHR$(A);
70 GOTO 50

```

7.7 Computer Serial Interface Program 2 – Course Tracing Program

This program allows you to enter a letter corresponding to the command to be given to the robot. (EXAMPLE: The letter F = FORWARD). The computer will then trace the intended route for the robot on the computer's CRT screen, by placing the instruction letter at each point as the command is entered.

To keep track of the robot's course, the program stores the robot's heading in the variable DIR (Direction). There are 8 possible headings, 1 through 8. As the robot makes a turn, DIR is either incremented (Right Turn) or decremented (Left Turn), thereby allowing the computer to keep track of the robot's intended navigation (lines 260-280). The computer then executes the proper cursor control statements based upon the heading stored in DIR. Lines 300 through 310 correct DIR if the calculated value is less than 1 or greater than 8.

```

10 REM ***ROBOT COURSE TRACING PROGRAM***
20 REM ***FOR IBM COMPATIBLE ONLY***
30 REM
40 OPEN "COM1:300,N,8" FOR RANDOM AS #1
90 CLS
100 PRINT "F = FORWARD"
110 PRINT "R = TURN RIGHT"
120 PRINT "L = TURN LEFT"
130 PRINT "B = SOUND BEEPER"
140 PRINT "X = LIGHT LED"
150 FOR S = 1 TO 16
160 PRINT
170 NEXT S
180 PRINT TAB(40); "X";
190 '
200 DIR = 1
210 '
220 '
230 A$ = INKEY$
240 IF A$ = "" THEN 230
250 '
260 IF A$ = "F" THEN DIR = DIR
270 IF A$ = "R" THEN DIR = DIR + 1
280 IF A$ = "L" THEN DIR = DIR - 1
290 '
300 IF DIR > 8 THEN DIR = DIR - 8
310 IF DIR < 1 THEN DIR = 8 - DIR

```

```

320 '
400 IF DIR = 1 THEN P$ = CHR$(29) + CHR$(30): REM CURSOR UP
410 IF DIR = 2 THEN P$ = CHR$(30) + CHR$(28): REM CURSOR UP,
    RIGHT
420 IF DIR = 3 THEN P$ = CHR$(28): REM CURSOR RIGHT
430 IF DIR = 4 THEN P$ = CHR$(31) + CHR$(28): REM CURSOR DOWN,
    RIGHT
440 IF DIR = 5 THEN P$ = CHR$(29) + CHR$(31): REM CURSOR DOWN
450 IF DIR = 6 THEN P$ = CHR$(29) + CHR$(29) + CHR$(29) + CHR$(31):
    REM CURSOR DOWN, LEFT
460 IF DIR = 7 THEN P$ = CHR$(29) + CHR$(29) + CHR$(29): REM
    CURSOR LEFT
470 IF DIR = 8 THEN P$ = CHR$(29) + CHR$(29) + CHR$(29) + CHR$(30):
    REM CURSOR UP, LEFT
480 IF A$ = "B" THEN P$ = CHR$(29)
490 IF A$ = "X" THEN P$ = CHR$(29)
500 '
510 PRINT P$; A$;
520 '
530 '
550 IF A$ = "F" THEN Z = 12: REM BINARY 1100 (BOTH MOTORS)
560 IF A$ = "R" THEN Z = 4: REM BINARY 1000 (LEFT MOTOR)
570 IF A$ = "L" THEN Z = 8: REM BINARY 0100 (RIGHT MOTOR)
580 IF A$ = "B" THEN Z = 2: REM BINARY 0010 (SOUND BEEPER)
590 IF A$ = "X" THEN Z = 1: REM BINARY 0001 (LIGHT LED)
600 '
610 Z = Z + 64
620 PRINT #1, CHR$(Z); : REM SEND TO ROBOT
630 '
640 GOTO 230

```

The Following are the binary combinations for the Basic Input Programs:

0	0000	Pause
1	0001	LED
2	0010	Beeper
3	0011	Beeper, LED
4	0100	Right Turn
5	0101	Right Turn, LED
6	0110	Right Turn, Beeper
7	0111	Right Turn, Beeper, LED
8	1000	Left Turn
9	1001	Left Turn, LED
10	1010	Left Turn, Beeper
11	1011	Left Turn, Beeper, LED
12	1100	Straight

Appendix

Power Source:

The robot has two different power supplies. A 9VDC battery is the source for the robot's logic circuits to utilize. It converts the 9VDC into 5VDC through its voltage regulator circuit.

The motor drive circuit has an independent power supply. It consists of two 1.5VDC AA batteries. With this configuration the motors for the drive circuit have their own power, and do not take any away from the operation of the logic functions of the robot.

The power for both the serial and parallel interfaces come from a 9VDC battery.

Serial Interface BAUD Rate:

The BAUD rate is the speed at which the interface communicates with the computer. The interface has been designed so that you can select one of the five most popular BAUD rates.

To set the BAUD rate the appropriate switch must be set to the ON position. The important thing is that the BAUD rate of the interface must match that of the computer.

<u>BAUD RATE</u>	<u>Switch Section ON</u>
300	1
600	2
1200	3
2400	4
4800	5

Parity:

Along with the data bits, an extra PARITY BIT is often sent to provide for ERROR checking.

Parity	Switch 6 ON
No Parity	Switch 6 OFF

Number of STOP bits:

After sending the data bits, a serial interface sends out either one or two STOP bits, which informs the receiving equipment that a complete set of bits have been sent.

One Stop Bit
Two Stop Bits

Switch 7 ON
Switch 7 OFF

Number of Bits:

Some computers send out 7 bits at a time, others send out 8. The computer and interface must match.

7 Bits
8 Bits

Switch 8 ON
Switch 8 OFF