

INTEL CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF ANY CIRCUITRY OTHER THAN CIRCUITRY EMBODIED IN AN INTEL PRODUCT. NO OTHER CIRCUIT PATENT LICENSES ARE IMPLIED.  
© INTEL CORPORATION, 1979 AFN-01364A-01

## USING THE 8049 AS AN 80 COLUMN PRINTER CONTROLLER

### I. INTRODUCTION

This Application Note details using INTEL's 8049 microcomputer as a dot matrix printer controller. Previous INTEL Application notes, (e.g. AP-27 and AP-54) described using intelligent processors and peripherals to control single printer mechanisms. This Application note expands upon the theme established in these prior notes and extends the concept to include a complete bi-directional 80 column printer using a single line buffer. For convenience this application note is divided into six sections:

1. INTRODUCTION
2. PRINT MECHANISM DESCRIPTION
3. INTERFACE CIRCUITRY
4. SOFTWARE
5. CONCLUSION
6. APPENDIX

Over the last few years 80 column output devices have become somewhat of a defacto output standard for business and some data processing applications. It should be mentioned that by no means is the 80 column format a "new" standard. 80 column computer cards have been around for more than 20 years and perhaps the existence of these cards in the early days of computers is why the 80 column format is a standard today.

Many CRT terminals use the 80 by N format and to complement this a number of printers use this same format. One reason, aside from those historic in nature, for the 80 column standard is that 80 columns of 12 pitch text on standard typewritten 8.5 inch by 11 inch paper completely fills up an entire line and allow ample room for margins. So, the 80 column format is an aesthetically convenient format.

Printers are usually divided into either impact or non-impact and a character or line oriented device. Impact printers actually use some type of "striker" to place ink on the paper. More often than not the ink is contained on a ribbon which is placed between the striker and the paper. Non-impact printers use some means other than direct pressure to place the characters on the paper. This type of printer is very fast because there is very little mechanical motion associated with placing the characters on the paper. However, because the paper is required to be treated with a special substance, it is not as convenient as an impact printer.

Character printers are capable of printing one character at a time. (Any standard home typewriter is in effect a character printer.) Line printers must print an

entire line at a time. Line printers are usually quite a bit faster than character printers, but they usually don't offer the print quality of character printers.

In recent years, the "computer boom" has caused the price of printers to tumble markedly. High volume production, competition, and the tremendous demand for reliable print mechanisms have all contributed to the decrease in price. Because of their simplicity, line printer mechanisms have decreased in price faster than other mechanisms. Therefore, when high quality print is not needed, a line printer is a very attractive choice.

This application note describes how to control an 80 column impact-line printer with an 8049/8039. The complete software listing is included in the appendix. The 8049 is the high-performance member of the MCS-48™ microcontroller family. The Processor has all of the features of the 8048 plus twice the amount of program and data memory and an 11MHz clock speed. For details about the 8049, please refer to the MCS-48 user's manual.

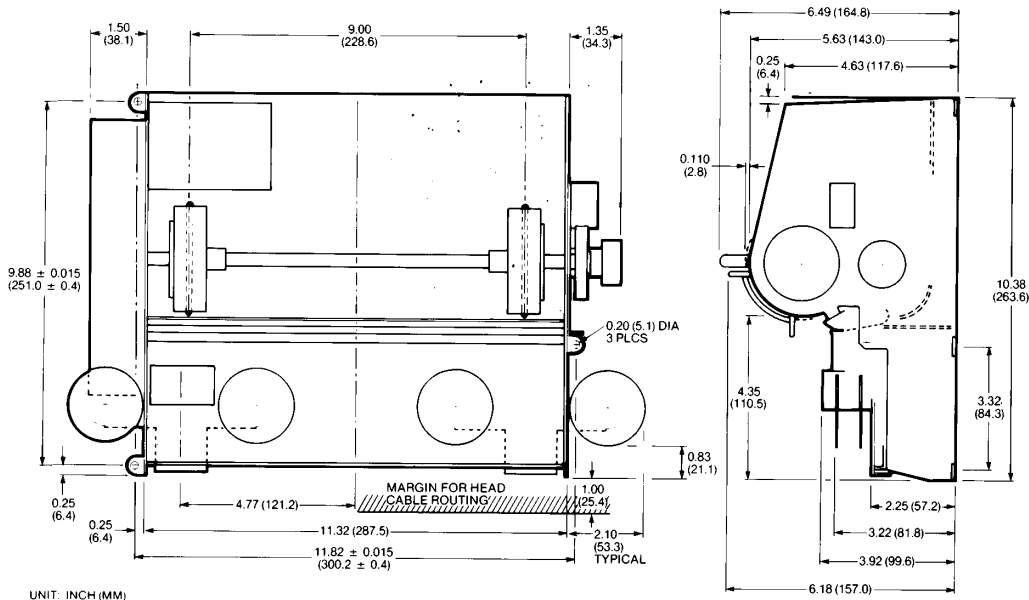
### II. PRINT MECHANISM DESCRIPTION

The model 820 printer is available from C. ITOH ELECTRONICS (5301 BEETHOVEN STREET, LOS ANGELES, CA 90066). This inexpensive and simple printer is ideal for applications requiring 80 columns of dot matrix alpha-numeric information.

The model 820 printer is comprised of three basic sub-assemblies; the chassis or frame, the paper feed mechanism, and the print head. The diagram in Figure 2.1 gives the physical dimensions of the basic print mechanism. The basic chassis for the printer is constructed out of four sheet metal stampings. These stampings are screwed together to form a sturdy base on which all other components of the printer are mounted.

The paper feed mechanism consists of a toothed wheel, a solenoid, a tension spring, and a "catcher." When the solenoid is activated, the arm of the solenoid pulls against the spring and drags over the toothed wheel. When the solenoid is released, its arm is pulled by the spring, but this time the arm grabs a tooth on the wheel and pulls the wheel forward which advances the paper. A "catcher," which is merely a piece of plastic held against the toothed wheel, is added to assure that the paper is advanced only one "tooth" position each time the solenoid is activated.

The print head is comprised of seven solenoids which are mounted in a common housing. The solenoids are physically mounted in a circle, but their hammers are positioned linearly along the vertical axis. These seven vertically positioned hammers are the strikers that actually do the printing.



UNIT: INCH (MM)  
DIMENSIONS IN INCH GOVERN

Figure 2.1 Physical Dimensions of C. ITOH Model 820 Printer

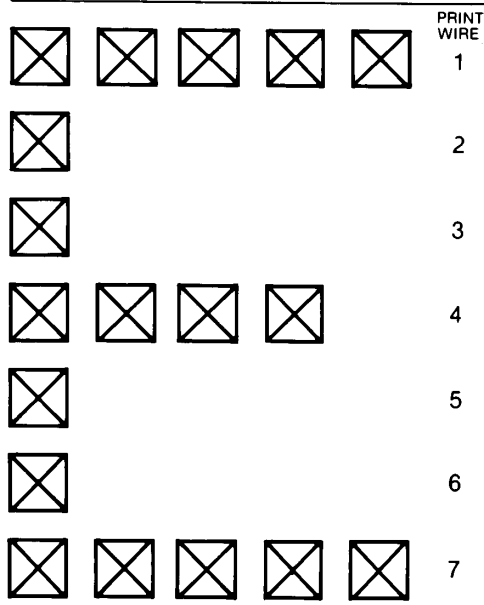


Figure 2.2 "Formation" of a Character by a Dot Matrix Printer

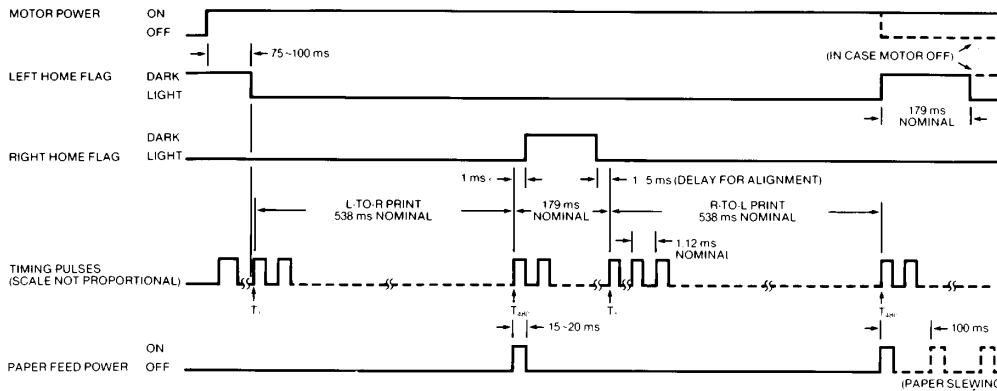
A motor, mounted toward the back of the print mechanism, drives a rubber toothed belt which turns a roller guide. A motor turns a guide that moves the print head from right to left and left to right. By properly timing the current flow through the solenoids while the print head is moving across the paper, characters can be formed. Figure 2.2 illustrates how the dot matrix printer "forms" its characters.

The timing pulses for the print head mechanism are generated by an opto-electronic sensor. This sensor, located on the left side plate of the printer, informs the print controller when to apply current to the print head mechanism. This "on-board timing wheel" assures that all characters will be properly spaced and that they will all be "in-line" in a vertical sense.

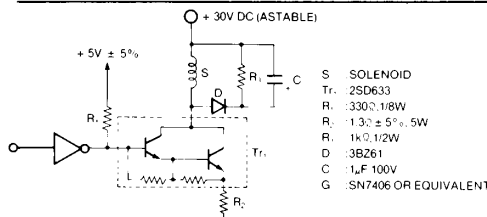
The print mechanism is also equipped with two additional sensors. These are the left home position sensor, located near the left front of the mechanism, and the right home position sensor, located near the right front of the print mechanism. These sensors simply tell the controller when the print head is in either the left or right home position. A complete timing chart for the printer is shown in Figure 2.3.

III. INTERFACE CIRCUITRY

The manual supplied with the printer recommends some specific interface circuitry. For the most part the circuitry used in this Application Note followed these suggestions. The circuitry needed to drive the print head solenoid is shown in Figure 3.1. This same



**Figure 2.3 Timing Diagram of C. ITOH Model 820 Printer**



**Figure 3.1 Solenoid Drive Circuit (Eliminate R2 for Line Feed Solenoid)**

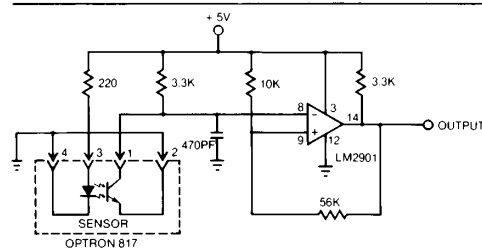
circuit is used to drive the line feed solenoid except that the current limiting resistor R2 is eliminated. This resistor is not needed because the line feed solenoid is physically much larger than the print head solenoids and can tolerate much higher levels of current.

The print head drivers are connected to an 8212 latch. The latch is interfaced to the BUS PORT on the 8049 and is enabled whenever the WR pin and the BIT 4 of PORT 1 are coincidentally low. The line feed driver is connected to PORT 1 BIT 1 of the 8049.

Note that the driver is simply a Darlington transistor that is driven by an open collector TTL gate. Resistor R2 is the current limiting resistor and diode D, capacitor C, and resistor R3 are used to "dampen" the inductive spike that occurs when driving solenoid S. This circuit is repeated for each of the seven solenoids in the print head. It should be mentioned that, although the type of Darlington transistor needed to drive the print head is not critical, a collector current rating of at least 5 amps and a breakdown voltage (V<sub>ceo</sub>) of at least 100 volts is needed. Transistors that do not meet these requirements will be damaged by the inductive kickback of the solenoids.

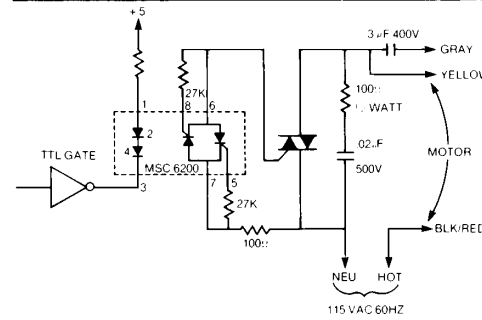
As mentioned in Section 2, the printer provides some sensor interface signals that are derived via three optoelectronic sensors. These signals must be amplified

and converted to TTL levels in order to interface to the controller. This conversion is accomplished with a simple voltage comparator. Figure 3.2 is a schematic of the sensor interface circuitry. Note that hysteresis is employed on the voltage comparators. This eliminates "false" sensing.



**Figure 3.2 Example of Sensor Circuit**

Motor control is accomplished by using a Monsanto MCS-6200 optically-coupled TRIAC. This part is ideal in this kind of application because it provides a simple means of controlling a line-operated motor without sacrificing the isolation needed for safe and reliable operation. Figure 3.3 is a schematic of the motor driving circuit.



**Figure 3.3 Motor Driving Circuit**

To interface 8049 to the outside world one 8212 latch was used. This latch was connected to the BUS PORT and is enabled by an INS or MOVX instruction coincident with BIT 4 of PORT 1 being in a logical zero state. In this configuration, the 8212 was used to hold the data until read by the 8049. The connection of the 8212 to the 8049 is shown in Figure 3.4 and the parallel port timing diagram is shown in Figure 3.5. The 8212 parallel port was connected to the LINE PRINTER OUTPUT of an INTELLEC MICROCOMPUTER DEVELOPMENT SYSTEM.

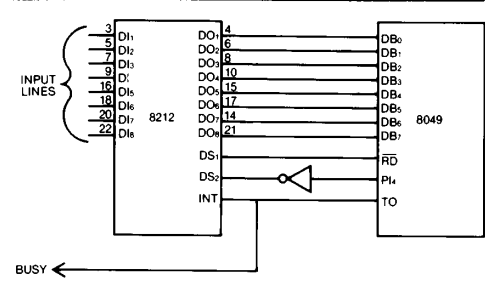


Figure 3.4 Connection of the 8212 Input Port to the 8049

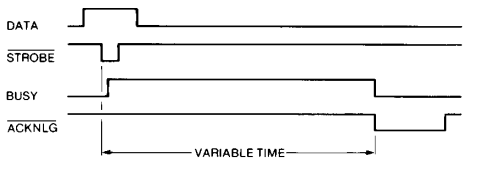


Figure 3.5 Parallel Port Timing

IV. SOFTWARE

As mentioned in Section 2, the bulk of the timing needed to control the printer is actually generated by the printer itself. Therefore, all the software must do is harness these timing signals and turn on and off the right solenoids at the right time.

To make things easy, the software needed to drive the printer is broken into four separate routines. These are:

1. INITIALIZATION ROUTINE
2. INPUT ROUTINE
3. OUTPUT ROUTINE
4. LOOKUP ROUTINE

The INITIALIZATION ROUTINE turns the motor on and checks the opto-electronic sensors. If a failure is found, the routine turns off the motor and loops on itself. This insures that the print mechanism is cycled properly before characters are accepted for printing.

This routine also initializes all of the variables used by the printer.

The INPUT ROUTINE reads the characters that are present in the 8212 input port and writes them into the 8049's buffer memory. The routine then checks the characters to see if a CARRIAGE RETURN (ASCII OCH) has been transmitted. If a CR is detected, the input routine automatically inserts a LINE FEED as the next character. When the input routine detects a LINE FEED, it stops reading characters and sets the direction bits and the print bit in the status register. This action evokes the OUTPUT ROUTINE. A detailed flowchart of the INPUT ROUTINE is shown in Figure 4.1.

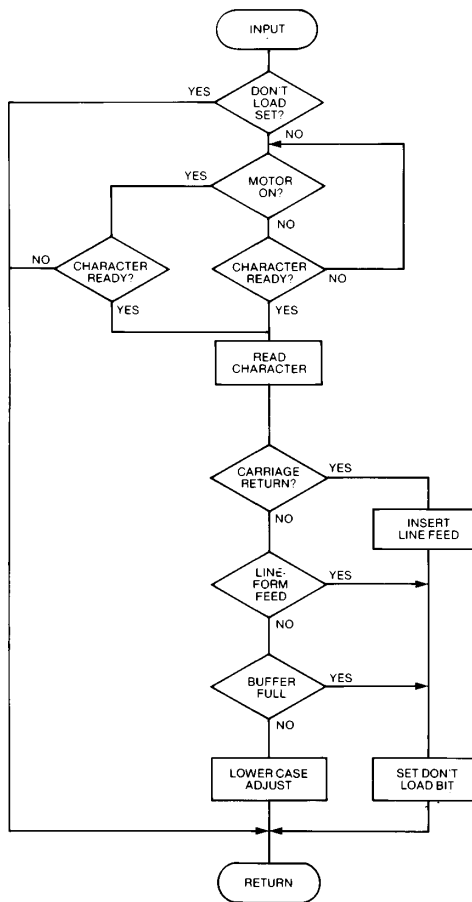


Figure 4.1 Input Routine Flowchart

The OUTPUT ROUTINE initializes both the input and output buffer pointers and then reads the characters from the 8049's buffer memory. After a character is read the OUTPUT ROUTINE calls the LOOKUP ROUTINE which reads the proper bit pattern to form that character. This bit pattern is then used to strobe the solenoids. After each character is printed, the OUTPUT ROUTINE calls the INPUT ROUTINE and another character is placed into the buffer memory. This type of operation guarantees that the input buffer cannot "overflow" the output buffer. A flowchart of the OUTPUT ROUTINE is shown in Figure 4.2.

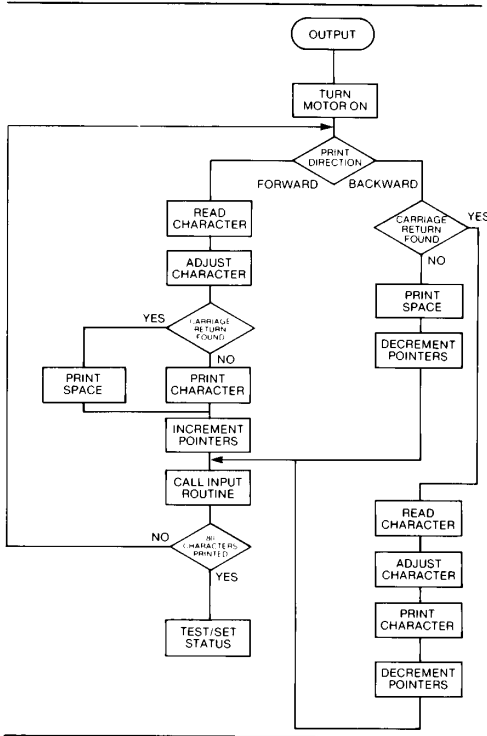


Figure 4.2 Output Routine Flowchart

**IV-I. HANDLING THE I/O BUFFER**

Since the C. ITOH Model 820 printer is capable of printing in both directions the 80 character buffer must be manipulated in a manner as to allow maximum input-output efficiency. This is accomplished by reversing the "direction" of the buffer memory each time the printer is printing from right to left. For simplicity, if it is assumed that the buffer is only five bytes long, Figure 4.3 can be used to help explain the buffer operation.

Initially the input buffer pointer is loaded with the address of the first location in the buffer memory. As characters are read, the input buffer pointer increments and fills the buffer memory as shown in Figure 4.3(b) through 4.3(f). When a CARRIAGE RETURN-LINE FEED (CRLF) is encountered the input buffer pointer and the output buffer pointer are reset back to the first location. The OUTPUT ROUTINE then reads the character from the first location in the buffer memory, increments the output buffer pointer and calls the INPUT ROUTINE, which reads another character from the parallel input port.

The OUTPUT ROUTINE reads the entire buffer, inserting space codes (20H) after a CR is detected, and the input buffer pointer follows the output buffer pointer as they "increment" up to the buffer memory. When the OUTPUT ROUTINE has printed the last character or space, the output buffer pointer and the input buffer pointer are set to point at the last location of the buffer memory. The OUTPUT ROUTINE then reads the character from the last location of the buffer memory and proceeds to "decrement" down the buffer memory. Space codes are inserted until a CR is found. Figure 4.3(1) to 4.3(0).

The input buffer pointer follows the output buffer pointer just as in the previous case. When the last, or in this case the first character is printed, the output buffer pointer and the input buffer pointer are set to point at the last location of the buffer memory. Now the pointers are "decrementing" down the buffer memory, but the printer is actually printing in a "normal" left to right fashion.

When the last character or space is printed, the output buffer and the input buffer pointer are set to the first location of the buffer memory and printing takes place in a reverse or right to left manner. After this line is printed, the print head and both buffer pointers are in the same position as they were initially. So, four lines must be printed before the buffer pointers and the print head complete a cycle. Each of these situations is handled separately by four different sub-routines: CASE0, CASE1, CASE2, and CASE3.

**IV-II. TIMING**

All critical timing for the printer controller came from two basic sources; the timing sensors on the printer and the internal eight-bit timer of the 8049.

The internal timer of the 8049 was used to control the length of time the solenoids were fired (600 microseconds) and was also used as a "one-shot" to align the printer. This alignment is needed to make the "backward" printing line up vertically with the normal or forward printing. The "one-shot" is used to measure the time from the last column of the last character position until the right sensor flag is covered.

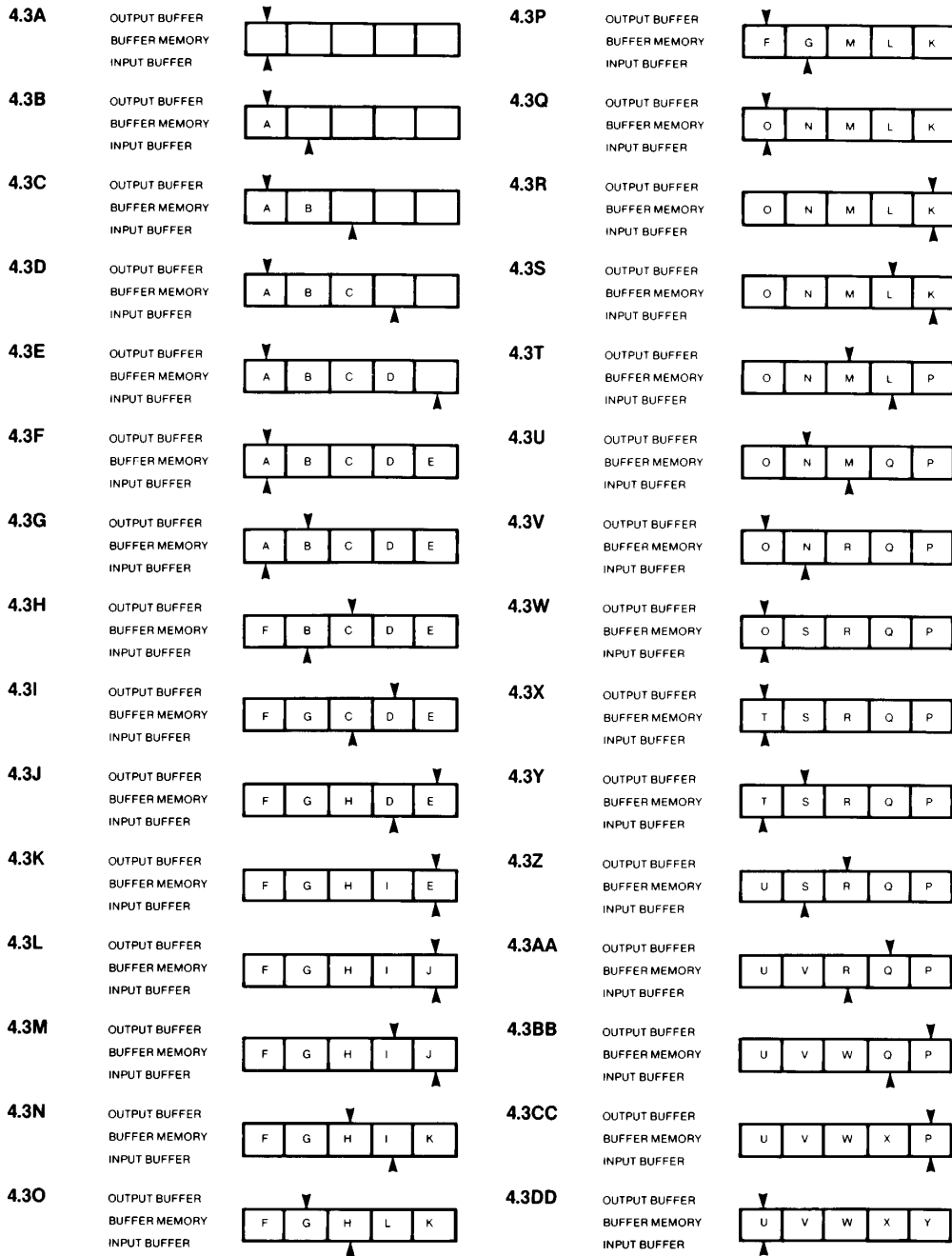


Figure 4.3 I/O Buffer Handler



When the print head reverses direction and the right sensor flag is uncovered, the timer is then used to determine where to start printing in the reverse direction.

The timer and the print wheel on the printer are used to determine when to place a character. The strobe from the print wheel informs the 8049 when to fire the solenoids and the timer allows the proper spacing between the characters.

#### V. CONCLUSION

Although the full speed of the 8049 was not used in this application, the high speed of the 8049 makes it possible to "fine-tune" any critical timing parameters. Additionally, the extra available CPU time could be used to add an interrupt driven keyboard and display, such as the ones discussed in AP-40, to the printer. This would allow the printer to function as a complete "terminal".

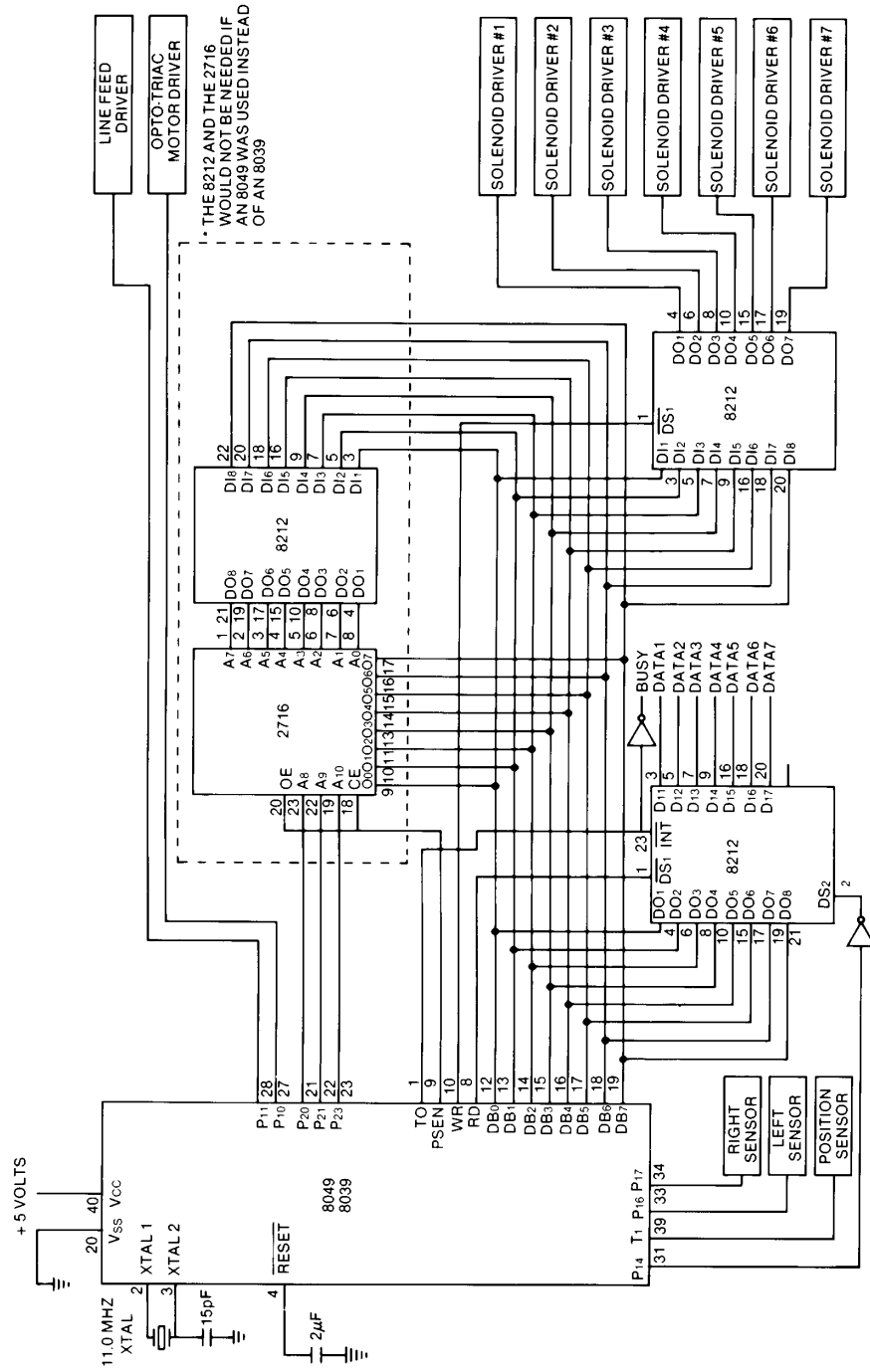
Very little attempt was made to optimize the software, but still the entire program fits easily in 1.25K of memory; 750 bytes for printer control and 500 bytes for character lookup. Adding lower case to the printer would require an additional 500 bytes of lookup table. The remaining 250 bytes should be used to add "user" features such as tabs, double width printing, etc.

The high speed of the 8049 combined with its hardware and software architecture make it an ideal choice for controlling an 80 column, bi-directional line printer. The I/O structure of the 8049 minimizes the amount of external hardware needed to control the printer and the large amount of on-board program and data memory allow quite a sophisticated control program to be implemented.





APPENDIX A. SCHEMATIC DIAGRAM





APPENDIX B. MONITOR LISTING

```

LOC  OBJ  SEQ  SOURCE STATEMENT
1      :
2      :*****
3      :
4      :THIS PROGRAM IMPLEMENTS CONTROL OF THE C. ITOM MODEL 828
5      :PRINTER. THE HARDWARE CONFIGURATION IS AS SUCH:
6      :8212 INPUT PORT ON BUS = DATA INPUT
7      :8212 OUTPUT PORT ON BUS = OUTPUT TO SOLENOID HAMMERS
8      :RT1 INPUT = CHARACTER POSITIONING SENSOR ON PRINTER
9      :LTB INPUT = INTERRUPT FROM 8212 INPUT PORT
10     :PORT 10 = MOTOR ON. LOW = ON
11     :PORT 11 = LINE FEED STROBE. LOW = ON
12     :PORT 16 = LEFT MARGIN SENSOR. LOW WHEN COVERED, HIGH WHEN OPEN
13     :PORT 17 = RIGHT MARGIN SENSOR. LOW WHEN COVERED, HIGH WHEN OPEN
14     :RT1 = PIN 2 OF LM339, PRINT WHEEL SENSOR
15     :PORT 16 = PIN 13 OF LM339
16     :PORT 17 = PIN 14 OF LM339
17     :
18     :*****
19     :
20     :SYSTEM EQUATES
21     :
0000   22 INBUF  EQU   R0      :POINTS AT INPUT LOCATION
0001   23 OUTBUF EQU   R1      :POINTS AT OUTPUT LOCATION
0002   24 SAVPHT EQU   R2      :STATUS FOR PRINTING
0003   25 STBCNT EQU   R3      :STROBE COUNTER
0004   26 TENPI  EQU   R4
0005   27 STATUS EQU   R5      :BIT 0 = LINE FEED SET
28     :BIT 1 = PRINT
29     :BIT 2 = CONTINUE
30     :BIT 3 = CR FOUND
31     :BIT 4 = LF FOUND
32     :BIT 5 = LF FOUND IN PRINTING
33     :BIT 6 = PRINT DIRECTION
34     :0 = RIGHT TO LEFT
35     :1 = LEFT TO RIGHT
36     :BIT 7 = BUFFER LOAD DIRECTION
37     :0 = FIRST TO MAX
38     :1 = MAX TO FIRST
0006   39 LINCNT EQU   R6      :THE LINE COUNTER
0007   40 JUNK1  EQU   R7
306F   41 MAX    EQU   6FH      :MAX BUFFER LOCATION
0020   42 FIRST  EQU   20H      :BOTTOM OF BUFFER
43 $EJECT

```



LOC	OBJ	SEQ	SOURCE STATEMENT
		44	:
		45	ORG 0000
		46	:
		47	:JUMP OVER THE INTERRUPT LOCATIONS
		48	:
0000	15	49	DIS I :DON'T USE INTERRUPTS
0001	0400	50	JMP RGIN :BEGIN THE PROGRAM
		51	:
000A		52	ORG 000A
		53	:
		54	:START THE PROGRAM
		55	:
		56	:LOOP UNTIL THE BUFFER FILLS UP
		57	:
000A	FD	58	PRNT: MOV A,STATUS :GET THE STATUS
000B	3211	59	JBI LPRNT :IF PRINTING, CONTINUE
000D	3400	60	CALL LOBUF :READ INTO THE BUFFER
000F	040A	61	JMP PRNT :LOOP
		62	:
		63	:THIS ROUTINE PRINTS A LINE
		64	:IT FIRST SAVES THE STATUS
		65	:AND THEN DETERMINES WHICH DIRECTION TO PRINT
		66	:AND HOW TO MANIPULATE THE BUFFER
		67	:
0011	0409	68	LPRNT: JMP STACHK :GO FIX UP THE STATUS
0013	F224	69	LPRNT: JBI CASE23 :JUMP TO CASE 2 AND 3
0015	0417	70	JMP CASE01 :JUMP TO CASE 0 AND 1
		71	:
		72	:CASE01, LOADING THE BUFFER FROM FIRST TO MAX
		73	:
0017	092B	74	CASE01: MOV OUTBUF,#FIRST :SET UP OUTBUF
0019	082B	75	MOV INBUF,#FIRST :SET UP INBUF
001B	FA	76	MOV A,SAVPNT :GET THE SAVED STATUS
001C	040C	77	CALL MOTON :TURN ON THE MOTOR
001E	0252	78	JBI CASE1 :PRINT FOWARD
0020	0403	79	CALL PRNTBK :GET READY TO PRINT BACKWARDS
0022	0431	80	JMP CASE0 :PRINT BACKWARDS
		81	:
		82	:CASE23, LOADING BUFFER FROM MAX TO FIRST
		83	:
0024	096F	84	CASE23: MOV OUTBUF,#MAX :SET UP OUTBUF
0026	086F	85	MOV INBUF,#MAX :SET UP INBUF
0028	FA	86	MOV A,SAVPNT :GET THE PRINT STATUS
0029	040C	87	CALL MOTON :TURN ON THE MOTOR
002B	02C2	88	JBI CASE3 :PRINT LEFT TO RIGHT
002D	0403	89	CALL PRNTBK :GET READY TO PRINT BACKWARDS
002F	040D	90	JMP CASE2 :PRINT RIGHT TO LEFT
		91	:
		92	#EJECT

LOC	OBJ	SEQ	SOURCE STATEMENT
0031	F1	93	CASEB: MOV A,@OUTBUF ;GET THE CHARACTER
0032	3491	94	CALL FXPRNT ;ADJUST FOR PRINTING
0034	B12B	95	MOV @OUTBUF,#2BH ;PUT A SPACE IN BUFFER RAM
0036	F242	96	FDC ;FOUND A CR
0038	945E	97	CALL INCTST ;UPDATE OUTBUF
003A	C6AE	98	JZ WATCHD ;WAIT FOR END
003C	BF2B	99	MOV JUNK1,#2BH ;GET A SPACE TO PRINT
003E	9463	100	CALL GTPRNT ;GO PRINT A SPACE
0040	B431	101	JMP CASEB ;LOOP
0042	BF2B	102	FDC: MOV JUNK1,#2BH ;GO PRINT THE LAST SPACE
0044	9463	103	FDC1: CALL GTPRNT ;GO PRINT A CHARACTER
0046	945E	104	CALL INCTST ;CHECK OUT BUFFER
0048	C6AE	105	JZ WATCHD ;WAIT FOR THE END
004A	F1	106	MOV A,@OUTBUF ;GET THE CHARACTER
004B	B12B	107	MOV @OUTBUF,#2BH ;PUT A SPACE THERE
004D	3491	108	CALL FXPRNT ;FIX THE CHARACTER UP
004F	AF	109	MOV JUNK1,A ;SAVE IT
005B	B444	110	JMP FDC1 ;LOOP
		111	;
		112	;
		113	;CASE 1, PRINTING LEFT TO RIGHT, LOADING BUFFER FROM
		114	;FIRST TO MAX
		115	;
0052	F1	116	CASE1: MOV A,@OUTBUF ;GET THE CHARACTER
0053	3491	117	CALL FXPRNT ;ADJUST FOR PRINTING
0055	AF	118	MOV JUNK1,A ;SAVE ACC
0056	B12B	119	MOV @OUTBUF,#2BH ;PUT A SPACE IN THE BUFFER
0058	F262	120	CRFOND ;FOUND A CR?
005A	9463	121	CALL GTPRNT ;GO PRINT THE CHARACTER
005C	945E	122	CALL INCTST ;CHECK THE BUFFER
005E	C675	123	JZ WATCH ;IS THE LAST CHARACTER BEING PRINTED?
0060	B452	124	JMP CASE1 ;LOOP
0062	B12B	125	CRFOND: MOV @OUTBUF,#2BH ;PUT A SPACE IN THE BUFFER MEMORY
0064	BF2B	126	MOV JUNK1,#2BH ;PUT A SPACE IN TEMP LOCATION
0066	9463	127	CALL GTPRNT ;GO PRINT THE SPACE
0068	945E	128	CALL INCTST ;CHECK THE BUFFER
006A	C675	129	JZ WATCH ;LAST CHARACTER PRINTED?
006C	F1	130	MOV A,@OUTBUF ;GET THE NEXT CHARACTER
006D	3491	131	CALL FXPRNT ;ADJUST IT
006F	B462	132	JMP CRFOND ;LOOP
		133	#EJECT

LDC	OBJ	SEQ	SOURCE STATEMENT
		134	;
		135	;THIS ROUTINE CALLS THE LINE FEED
		136	;
0071	9478	137	DOLF: CALL LINEFD ;STROBE LINE FEED SOLENOID
0073	040A	138	JMP PRNT ;GO BACK TO THE PRINT ROUTINE
		139	;
		140	;THIS ROUTINE COMPLETES A LINE WHEN THE PRINT
		141	;HEAD IS MOVING LEFT TO RIGHT
		142	;
0075	27	143	WATCH: CLR A ;ZERO ACC
0076	62	144	MOV T,A ;ZERO TIMER
0077	55	145	STRT T ;START THE TIMER
0078	3400	146	CALL LDBUF ;GO READ THE LAST CHARACTER
007A	09	147	LOOPM: IN A,P1 ;EXAMIN PORT ONE
007B	F27A	148	JB7 LOOPM ;CHECK RIGHT HAND SENSOR
007D	65	149	STOP TCNT ;STOP THE TIMER
007E	FD	150	MOV A,STATUS ;GET THE STATUS
007F	5205	151	JB2 OVR1 ;JUMP IF CONTINUE IS SET
0081	94DF	152	CALL MOTOFF ;TURN MOTOR OFF
0083	53FD	153	ANL A,#BFDH ;RESET BIT ONE
0085	53FB	154	OVR1: ANL A,#BFBH ;RESET CONTINUE BIT
0087	AD	155	MOV STATUS,A ;RESTORE STATUS
0088	FA	156	MOV A,SAVPNT ;GET THE SAVED STATUS
0089	0271	157	JB5 DOLF ;DO A LINE FEED IF BIT IS SET
008B	040A	158	JMP PRNT ;GO BACK TO PRINT ROUTINE
		159	;
		160	;
		161	;CASE 2, PRINTING RIGHT TO LEFT, LOADING BUFFER FROM
		162	;MAX TO FIRST
		163	;
		164	;
008D	F1	165	CASE2: MOV A,@OUTBUF ;GET THE CHARACTER
008E	3491	166	CALL FXPRNT ;ADJUST FOR PRINTING
0090	B120	167	MOV @OUTBUF,#2BH ;PUT A SPACE IN BUFFER RAM
0092	F29E	168	JB7 FDCR ;FIND A CR YET
0094	9472	169	CALL DECTST ;CHECK THE BUFFER
0096	C6AE	170	JZ WATCHD ;IF ZERO WAIT FOR SENSOR FLAG
0098	BF20	171	MOV JUNK1,#2BH ;PUT SPACE IN TEMP LOCATION
009A	9463	172	CALL GTPRNT ;GO PRINT SPACE
009C	0400	173	JMP CASE2 ;LOOP
009E	BF20	174	FDCR: MOV JUNK1,#2BH ;GET A SPACE
00A0	9463	175	FDCR1: CALL GTPRNT ;GO PRINT THE CHARACTER
00A2	9472	176	CALL DECTST ;CHECK THE BUFFER
00A4	C6AE	177	JZ WATCHD ;LEAVE IF DONE
00A6	F1	178	MOV A,@OUTBUF ;GET A CHARACTER
00A7	3491	179	CALL FXPRNT ;ADJUST THE CHARACTER FOR PRINTING
00A9	AF	180	MOV JUNK1,A ;SAVE IT
00AA	B120	181	MOV @OUTBUF,#2BH ;PUT A SPACE WHERE THE CHARACTER WAS
00AC	0400	182	JMP FDCR1 ;LOOP
		183	*EJECT

```

LOC  OBJ          SEQ          SOURCE STATEMENT
;
184          ;
185          ;THIS ROUTINE WAITS FOR THE SENSOR FLAGS TO BE COVERED
186          ;WHEN PRINTING RIGHT TO LEFT
187          ;
BBAE 3400      188 WATCHD: CALL   LDBUF          ;GO READ THE LAST CHARACTER
BBB8 B9        189          IN     A,P1             ;GET SENSOR INFORMATION
BBB1 D2AE      190          JB6   WATCHD         ;LOOP IF SENSOR IS NOT COVERED
BBB3 FD        191          MOV   A,STATUS        ;GET THE STATUS
BBB4 52BA      192          JB2   OVR           ;SEE IF CONTINUE IS SET
BBB6 94DF      193          CALL  MOTDF         ;TURN THE MOTOR OFF
BBB8 53FD      194          ANL   A,#BF0H        ;RESET BIT 1
BBBA 53FB      195 OVR:   ANL   A,#BF3H        ;RESET BIT 3
BBBC AD        196          MOV   STATUS,A          ;RESTORE STATUS
BBBD FA        197          MOV   A,SAVPNT         ;GET THE SAVED STATUS
BBBE B271      198          JB5   DDLF           ;DD A LINE FEED
BBCC B4BA      199          JMP   PRNT           ;EXIT
200          ;
201          ;CASE 3, PRINTING LEFT TO RIGHT, LOADING BUFFER FROM
202          ;MAX TO FIRST
203          ;
BBCC F1        204 CASE3: MOV   A,@OUTBUF         ;GET A CHARACTER
BBCC 3491      205          CALL  FXPRNT          ;FIX FOR PRINTING
BBCC AF        206          MOV   JUNK1,A          ;SAVE CHARACTER
BBCC B12B      207          MOV   @OUTBUF,#2BH        ;PUT A SPACE IN THE BUFFER
BBCC F2D2      208          JB7   CRFND         ;LEAVE IF A CR IS FOUND
BBCA 9463      209          CALL  GTPRNT          ;GO PRINT THE CHARACTER
BBCC 9472      210          CALL  DECTST         ;CHECK THE BUFFER
BBCE C675      211          JZ    WATCH          ;LEAVE IF DONE
BBDB B4C2      212          JMP   CASE3           ;LOOP
BBD2 B12B      213 CRFND: MOV   @OUTBUF,#2BH        ;PUT A SPACE IN THE BUFFER RAM
BBD4 BF2B      214          MOV   JUNK1,#2BH        ;GET A SPACE
BBD6 9463      215          CALL  GTPRNT          ;PRINT A SPACE
BBD8 9472      216          CALL  DECTST         ;CHECK THE BUFFER
BBDA C675      217          JZ    WATCH          ;LEAVE IF DONE
BBDC F1        218          MOV   A,@OUTBUF         ;GET NEXT CHARACTER
BBDD 3491      219          CALL  FXPRNT          ;ADJUST IT
BBDF B4D2      220          JMP   CRFND           ;LOOP
221          ;EJECT

```

LDC OBJ	SEQ	SOURCE STATEMENT	
	222	ORG 100H	
	223	;	
B100 09	224	LDBUF: IN A,P1	:READ PORT 1
B101 021C	225	JB5 LMODE	:BIT 5 = H = LINE MODE
B103 1207	226	JBB ARND	:JUMP AROUND IF MOTOR IS ON
B105 0901	227	ORL P1,#01H	:TURN THE MOTOR OFF
B107 920F	228	ARND: JB4 NOFF	:NO FORM FEED
B109 FE	229	MOV A,LINCHT	:GET THE LINE COUNTER
B10A 4300	230	ORL A,#00H	:SET MSB
B10C AE	231	MOV LINCNT,A	:RESTORE THE LINE COUNTER
B10D 23FF	232	MOV A,#0FFH	:SET ACC
B10F 721A	233	NOFF: JB3 NOLF	:JUMP IF NO LINE FEED
B111 9478	234	CALL LINEFD	:GO DO A LF OR FF
B113 09	235	BUTLOP: IN A,P1	:READ THE PORT
B114 721A	236	JB3 NOLF	:WAIT FOR SWITCH TO BE RELEASED
B116 921A	237	JB4 NOLF	:WAIT FOR SWITCH TO BE RELEASED
B118 2413	238	JMP BUTLOP	:LOOP
B11A 2400	239	NOLF: JMP LDBUF	:LOOP
	240	;	
	241	;	
	242	:FIRST SEE IF A CHARACTER IS PRESENT IN THE BUFFER	
	243	;	
B11C 261F	243	LHMODE: JMTB CHAR	:IF CHARACTER PRESENT, READ IT
B11E 03	244	RET	:IF NOT, EXIT ROUTINE
	245	;	
	246	;	
	247	:IF THERE IS A CHARACTER, READ IT	
	248	;	
B11F FD	248	CHAR: MOV A,STATUS	:GET THE STATUS
B120 5249	249	JB2 ARNDJP	:IF CONTINUE IS SET, DON'T LOAD
B122 9249	250	JB4 ARNDJP	:IF LF IS SET, DON'T LOAD
B124 724A	251	JB3 LFCRCK	:WAS CR SET, SEE IF NEXT CHAR IS LF
B126 9406	252	CALL GTCAR	:GO READ A CHARACTER
B128 3461	253	GOOD: CALL FXCHAR	:MAKE SURE IT IS OK
B12A AB	254	MOV #INBUF,A	:SAVE CHARACTER IN BUFFER MEMDRY
B12B FD	255	MOV A,STATUS	:GET THE STATUS
B12C F239	256	SUBI	:IF BIT 7 IS SET DECREMENT BUFFER
B12E 18	257	INC INBUF	:UPDATE INBUF
B12F 237B	258	MOV A,#MAX+1	:GET TOP
B131 D8	259	XRL A,INBUF	:ARE WE AT THE TOP?
B132 9649	260	ARNDJP	:IF NOT GET THE STATUS
B134 F8	261	MOV A,INBUF	:GET INBUF
B135 07	262	DEC A	:CHANGE BY ONE
B136 AB	263	MOV INBUF,A	:PUT IT BACK
B137 2449	264	JMP ARNDJP	:GET THE STATUS
B139 F8	265	SUBI: MOV A,INBUF	:GET INBUF
B13A 07	266	DEC A	:CHANGE BY ONE
B13B AB	267	MOV INBUF,A	:PUT INBUF BACK
B13C 231F	268	MOV A,#FIRST-1	:GET THE BOTTOM OF THE BUFFER
B13E D8	269	XRL A,INBUF	:TEST THE BUFFER
B13F 9649	270	JNZ ARNDJP	:IF NOT ZERO READ THE STATUS
B141 18	271	INC INBUF	:MOVE INBUF BACK
B142 2449	272	JMP ARNDJP	:GO GET STATUS
B144 FD	273	GETSTA: MOV A,STATUS	:GET THE STATUS
B145 1249	274	JBB ARNDJP	:IF BIT 0 SET, BYPASS
B147 9250	275	JB4 STBIT1	:IF LF IS FOUND, SET THE STATUS
B149 03	276	ARNDJP: RET	:EXIT
	277	;	
	278	:THIS ROUTINE "FORCES" A LF AFTER A CR	
	279	;	
B14A 9406	280	LFCRCK: CALL GTCAR	:READ A CHARACTER
B14C 230A	281	MOV A,#0AH	:GET A LINE FEED
B14E 242B	282	JMP GOOD	:JUMP BACK
	283	;	
	284	:THIS ROUTINE SETS THE STATUS BITS	
	285	;	
B150 FD	286	STBIT1: MOV A,STATUS	:LOAD THE STATUS
B151 3259	287	JB1 STPRNT	:IF STILL PRINTING, LEAVE
B153 4302	288	ORL A,#02H	:SET PRINT BIT
B155 0348	289	ADD A,#04H	:UPDATE POSITION COUNTER
B157 AD	290	MOV STATUS,A	:PUT STATUS BACK
B158 03	291	RET	:EXIT ROUTINE
B159 5260	292	STPRNT: JB2 BYEBYE	:CHECK CONTINUE BIT
B15B 4304	293	ORL A,#04H	:SET CONTINUE BIT
B15D 0348	294	ADD A,#04H	:UPDATE PRINT DIRECTION
B15F AD	295	MOV STATUS,A	:PUT THE STATUS BACK
B160 03	296	BYEBYE: RET	:EXIT
	297	;	

```

LDC OBJ      SEQ      SOURCE STATEMENT

298          ;THIS ROUTINE "CONVERTS" LOWER CASE LETTERS TO
299          ;UPPER CASE
300          ;
0161 97      301 FXCHAR: CLR      C          ;CLEAR THE CARRY
0162 537F    302 ANL      A,#7FH      ;STRIP MSB
0164 AF      303 MOV      JUNK1,A        ;SAVE ACC
0165 0308    304 ADD      A,#080BH        ;SEE IF NUMBER IS 6BH
0167 E670    305 JNC      FINE           ;IF CARRY ISN'T SET, JUMP
0169 FF      306 MOV      A,JUNK1      ;GET ACC BACK
016A 37      307 CPL      A              ;SUBTRACT 20H FROM THE ACC
0168 0320    308 ADD      A,#20H
016D 37      309 CPL      A
016E 2474    310 JMP      FIXDUN          ;JUMP TO TEST CR LF
0170 37      311 FINE:   CPL      A          ;NOW SUBTRACT 08H FROM ACC
0171 0308    312 ADD      A,#080BH
0173 37      313 CPL      A
0174 AF      314 FIXDUN: MOV     JUNK1,A        ;SAVE A
0175 030D    315 XRL      A,#0DH        ;IS CHARACTER A CR
0177 967F    316 JNZ      LFTEST        ;IF IT IS NOT TEST LF
0179 FD      317 MOV      A,STATUS      ;GET THE STATUS
017A 4308    318 ORL      A,#08H        ;SET BIT 3
017C AD      319 MOV      STATUS,A      ;RESTORE THE STATUS
017D 240F    320 JMP      FIXFIN        ;LEAVE
017F FF      321 LFTEST: MOV     A,JUNK1      ;GET CHARACTER BACK
0180 030A    322 XRL      A,#0AH        ;IS IT A LF
0182 C609    323 JZ       FIXUP          ;IF ITS NOT, WE ARE DONE
0184 FF      324 MOV      A,JUNK1      ;GET THE CHARACTER BACK
0185 030C    325 XRL      A,#0CH        ;IS IT A FORM FEED
0187 960F    326 JNZ      FIXFIN        ;IF NOT FORM FEED, JUMP
0189 FD      327 FIXUP:  MOV     A,STATUS      ;GET THE STATUS
018A 4310    328 ORL      A,#10H        ;SET BIT 4
018C AD      329 MOV      STATUS,A      ;RETURN THE STATUS
018D 3450    330 CALL     STBIT1        ;SET THE STATUS
018F FF      331 FIXFIN: MOV     A,JUNK1      ;GET THE CHARACTER
LDC OBJ      SEQ      SOURCE STATEMENT

0198 03      332 RET              ;EXIT FIXCHAR
333          ;
334          ;THIS ROUTINE RECOGNIZES A LF, FF, AND CR
335          ;DURING THE PRINT OPERATION
336          ;IT ALSO FORCES A SPACE IF A CHARACTER FOUND
337          ;IN THE BUFFER IS NOT IN THE LOOKUP TABLE
338          ;
0191 AF      339 FXPRNT: MOV     JUNK1,A        ;SAVE ACC
0192 030C    340 XRL      A,#0CH        ;FORM FEED
0194 C602    341 JZ       FFFIX        ;GO SET FORM FEED
0196 FF      342 MOV      A,JUNK1      ;RESTORE CHARACTER
0197 030D    343 XRL      A,#0DH        ;SEE IF IT IS A CR
0199 C608    344 JZ       CRFIX        ;LEAVE IF IT IS
0198 FF      345 MOV      A,JUNK1      ;GET ACC BACK
019C 030A    346 XRL      A,#0AH        ;SEE IF IT IS A LF
019E C609    347 JZ       LFFIX        ;LEAVE IF IT IS
01A0 FF      348 MOV      A,JUNK1      ;GET CHARACTER BACK
01A1 53E0    349 ANL      A,#0EBH        ;SEE IF IT IS A CHARACTER
01A3 960D    350 JNZ      JSCHAR        ;IF IT IS JUMP
01A5 2320    351 MOV      A,#20H        ;PUT A SPACE IN ACC
01A7 03      352 RET              ;EXIT
01A8 4308    353 CRFIX:  ORL     A,#08H        ;SET BIT 7
01AA 03      354 RET              ;EXIT
01AB FD      355 LFFIX:  MOV     A,STATUS      ;GET THE STATUS
01AC 4320    356 ORL      A,#20H        ;SET LF BIT IN STATUS
01AE AD      357 MOV      STATUS,A      ;PUT THE STATUS BACK
01AF 2320    358 MOV      A,#20H        ;GET A SPACE
01B1 03      359 RET              ;EXIT
01B2 FD      360 FFFIX:  MOV     A,STATUS      ;GET THE STATUS
01B3 4320    361 ORL      A,#20H        ;SET LINE FEED BIT
01B5 AD      362 MOV      STATUS,A      ;PUT THE STATUS BACK
01B6 FE      363 MOV      A,LINCNT      ;GET THE LINE COUNT
01B7 4308    364 ORL      A,#08H        ;SET BIT 7
01B9 AE      365 MOV      LINCNT,A      ;PUT LINE COUNT BACK
01BA 2320    366 MOV      A,#20H        ;GET A SPACE
01BC 03      367 RET              ;EXIT
01BD FF      368 ISCHAR: MOV     A,JUNK1      ;GET CHARACTER BACK
01BE 533F    369 ANL      A,#3FH        ;STRIP THE TWO MSB
01CB 03      370 RET              ;EXIT

```



```

LOC  OBJ      SEQ      SOURCE STATEMENT
      371      ;
      372      ;THIS ROUTINE PRINTS THE CHARACTER IN THE ACC
      373      ;
01C1 AC      374 PRNTIT:  MOV     TEMP1,A      ;SAVE CHARACTER
01C2 E7      375      RL     A           ;MULTIPLY BY TWO
01C3 E7      376      RL     A           ;MULTIPLY BY FOUR
01C4 6C      377      ADD    A,TEMP1      ;ADD ONCE TO MULTIPLY BY 5
      378      ;
      379      ;NOW SEE WHAT PART OF THE LOOKUP TABLE TO USE
      380      ;
01C5 2C      381      XCH   A,TEMP1      ;PUT CHARACTER IN A, TARGET IN TEMP1
01C6 B2CA    382      JBS   SHORT      ;JUMP TO HIGH ADDRESS IF BIT 5 SET
01C8 44AB    383      JMP   PAGE1      ;GO TO FIRST PART OF LOOKUP TABLE
01CA 64AB    384 SHORT:  JMP   PAGE2      ;GO TO SECOND PAGE OF LOOKUP TABLE
      385      ;
      386      ;THIS ROUTINE TRIGGERS THE SOLENOIDS FOR 600 MICROSECONDS
      387      ;AFTER WAITING FOR THE TRIGGER SIGNAL FROM THE PRINTER
      388      ;*
01CC AF      389 FIRE:  MOV     JUNK1,A      ;SAVE THE ACC
01CD FD      390      MOV     A,STATUS      ;GET THE STATUS
01CE D2D4    391      JBS   NT1          ;SEE IF FORWARD OR BACKWARDS
01D0 56D0    392 FIREX:  JTI   FIREX      ;WAIT FOR TI
01D2 24D6    393      JMP   FIREY      ;LEAVE
01D4 46D4    394 NT1:   JNT1   NT1          ;LOOP
01D6 FF      395 FIREY:  MOV     A,JUNK1      ;GET ACC BACK
01D7 90      396      MOVX   #RB,A      ;TRIGGER THE SOLENOID
      397      ;
      398      ;NOW KILL 600 MICROSECONDS
      399      ;
01D8 23F3    400      MOV     A,#0F3H      ;LOAD DELAY NUMBER
01DA 62      401      MOV     T,A          ;PUT IT IN TIMER
01DB 55      402      STRT   T           ;START THE TIMER
01DC 16EB    403 TSJTF:  JTF   KTDUH      ;LOOP ON TIMER FLAG
01DE 24DC    404      JMP   TSJTF      ;
01E0 27      405 KTDUH:  CLR    A           ;ZERO ACC
01E1 90      406      MOVX   #RB,A      ;TURN OFF SOLENOIDS
01E2 65      407      STOP   TCNT      ;STOP THE TIMER
01E3 03      408      RET     ;EXIT FIRE ROUTINE
01E9 $EJECT  409
    
```

```

LOC  OBJ      SEQ      SOURCE STATEMENT
      410      ;
      411      ;*****
      412      ;
      413      ;THIS IS THE LOOKUP TABLE. THE MSB IS NOT USED. THE MSB - 1
      414      ;IS THE DOT THAT IS THE TOP OF ANY GIVEN CHARACTER AND THE
      415      ;LSB IS THE DOT THAT IS THE BOTTOM OF ANY GIVEN CHARACTER
      416      ;
      417      ;*****
      418      ;
      0200     419      ORG      200H
      420      ;*
      0200 3E   421  TABLE1: DB      3EH      ; *****
      0201 41   422      DB      41H      ; * * * *
      0202 5D   423      DB      5DH      ; * * * *
      0203 59   424      DB      59H      ; * * * *
      0204 4E   425      DB      4EH      ; * * * *
      426
      0205 7C   427      DB      7CH      ; *****
      0206 12   428      DB      12H      ; * * * *
      0207 11   429      DB      11H      ; * * * *
      0208 12   430      DB      12H      ; * * * *
      0209 7C   431      DB      7CH      ; *****
      432
      020A 7F   433      DB      7FH      ; *****
      020B 49   434      DB      49H      ; * * * *
      020C 49   435      DB      49H      ; * * * *
      020D 49   436      DB      49H      ; * * * *
      020E 36   437      DB      36H      ; * * * *
      438
      020F 3E   439      DB      3EH      ; *****
      0210 41   440      DB      41H      ; * * * *
      0211 41   441      DB      41H      ; * * * *
      0212 41   442      DB      41H      ; * * * *
      0213 22   443      DB      22H      ; * * * *
      444
      0214 7F   445      DB      7FH      ; *****
      0215 41   446      DB      41H      ; * * * *
      0216 41   447      DB      41H      ; * * * *
      0217 41   448      DB      41H      ; * * * *
      0218 3E   449      DB      3EH      ; *****
      450
      0219 7F   451      DB      7FH      ; *****
      021A 49   452      DB      49H      ; * * * *
      021B 49   453      DB      49H      ; * * * *
      021C 49   454      DB      49H      ; * * * *
      021D 41   455      DB      41H      ; * * * *
      456  $EJECT
    
```



LOC	OBJ	SEQ	SOURCE STATEMENT
		457	
021E	7F	458	DB 7FH
021F	09	459	DB 09H
0220	09	460	DB 09H
0221	09	461	DB 09H
0222	01	462	DB 01H
		463	
0223	3E	464	DB 3EH
0224	41	465	DB 41H
0225	41	466	DB 41H
0226	51	467	DB 51H
0227	71	468	DB 71H
		469	
0228	7F	470	DB 7FH
0229	00	471	DB 00H
022A	00	472	DB 00H
022B	00	473	DB 00H
022C	7F	474	DB 7FH
		475	
022D	00	476	DB 00H
022E	41	477	DB 41H
022F	7F	478	DB 7FH
0230	41	479	DB 41H
0231	00	480	DB 00H
		481	
0232	20	482	DB 20H
0233	40	483	DB 40H
0234	40	484	DB 40H
0235	40	485	DB 40H
0236	3F	486	DB 3FH
		487	
0237	7F	488	DB 7FH
0238	00	489	DB 00H
0239	14	490	DB 14H
023A	22	491	DB 22H
023B	41	492	DB 41H
		493	
023C	7F	494	DB 7FH
023D	40	495	DB 40H
023E	40	496	DB 40H
023F	40	497	DB 40H
0240	40	498	DB 40H
		499	
0241	7F	500	DB 7FH
0242	02	501	DB 02H
0243	0C	502	DB 0CH
0244	02	503	DB 02H
0245	7F	504	DB 7FH
		505	
0246	7F	506	DB 7FH
0247	04	507	DB 04H
0248	00	508	DB 00H
0249	10	509	DB 10H
024A	7F	510	DB 7FH
		511	*EJECT

LOC	OBJ	SEQ	SOURCE STATEMENT
		512	
024B	3E	513	DB 3EH ; *****
024C	41	514	DB 41H ; * * *
024D	41	515	DB 41H ; * * *
024E	41	516	DB 41H ; * * *
024F	3E	517	DB 3EH ; *****
		518	
0250	7F	519	DB 7FH ; *****
0251	09	520	DB 09H ; * * *
0252	09	521	DB 09H ; * * *
0253	09	522	DB 09H ; * * *
0254	06	523	DB 06H ; **
		524	
0255	3E	525	DB 3EH ; *****
0256	41	526	DB 41H ; * * *
0257	51	527	DB 51H ; * * *
0258	21	528	DB 21H ; * * *
0259	5E	529	DB 5EH ; *****
		530	
025A	7F	531	DB 7FH ; *****
025B	09	532	DB 09H ; * * *
025C	19	533	DB 19H ; * * *
025D	29	534	DB 29H ; * * *
025E	46	535	DB 46H ; * **
		536	
025F	26	537	DB 26H ; * **
0260	49	538	DB 49H ; * * *
0261	49	539	DB 49H ; * * *
0262	49	540	DB 49H ; * * *
0263	32	541	DB 32H ; ** *
		542	
0264	01	543	DB 01H ; *
0265	01	544	DB 01H ; *
0266	7F	545	DB 7FH ; *****
0267	01	546	DB 01H ; *
0268	01	547	DB 01H ; *
		548	
0269	3F	549	DB 3FH ; *****
026A	40	550	DB 40H ; *
026B	40	551	DB 40H ; *
026C	40	552	DB 40H ; *
026D	3F	553	DB 3FH ; *****
		554	
026E	1F	555	DB 1FH ; *****
026F	20	556	DB 20H ; *
0270	40	557	DB 40H ; *
0271	20	558	DB 20H ; *
0272	1F	559	DB 1FH ; *****
		560	
0273	7F	561	DB 7FH ; *****
0274	20	562	DB 20H ; *
0275	18	563	DB 18H ; **
0276	20	564	DB 20H ; *
0277	7F	565	DB 7FH ; *****
		566	*EJECT

LOC	OBJ	SEQ	SOURCE STATEMENT
		567	
0278	63	568	DB 63H ; ** **
0279	14	569	DB 14H ; * *
027A	00	570	DB 00H ; *
027B	14	571	DB 14H ; * *
027C	63	572	DB 63H ; ** **
		573	
027D	03	574	DB 03H ; **
027E	04	575	DB 04H ; *
027F	78	576	DB 78H ; ****
0280	04	577	DB 04H ; *
0281	03	578	DB 03H ; **
		579	
0282	61	580	DB 61H ; ** *
0283	51	581	DB 51H ; * * *
0284	49	582	DB 49H ; * * *
0285	45	583	DB 45H ; * * *
0286	43	584	DB 43H ; * **
		585	
0287	7F	586	DB 7FH ; *****
0288	7F	587	DB 7FH ; *****
0289	41	588	DB 41H ; * * *
028A	41	589	DB 41H ; * * *
028B	41	590	DB 41H ; * * *
		591	
028C	02	592	DB 02H ; *
028D	04	593	DB 04H ; *
028E	00	594	DB 00H ; *
028F	10	595	DB 10H ; *
0290	20	596	DB 20H ; *
		597	
0291	41	598	DB 41H ; * * *
0292	41	599	DB 41H ; * * *
0293	41	600	DB 41H ; * * *
0294	7F	601	DB 7FH ; *****
0295	7F	602	DB 7FH ; *****
		603	
0296	10	604	DB 10H ; *
0297	00	605	DB 00H ; *
0298	04	606	DB 04H ; *
0299	00	607	DB 00H ; *
029A	10	608	DB 10H ; *
		609	
029B	40	610	DB 40H ; *
029C	40	611	DB 40H ; *
029D	40	612	DB 40H ; *
029E	40	613	DB 40H ; *
029F	40	614	DB 40H ; *
		615	*EJECT



LOC	OBJ	SEQ	SOURCE STATEMENT
		616	;
02A0	0000	617	PAGE1: MOV STBCNT, #00H ;ZERO STROBE COUNTER
02A2	FA	618	MOV A, SAVPNT ;GET DIRECTION
02A3	37	619	CPL A ;FLIP BITS
02A4	0203	620	JB6 BAKWRD ;IF BACKWARD JUMP OUT
02A6	FC	621	LKLO: MOV A, TEMP1 ;GET THE TARGET
02A7	A3	622	MOV A, 0A ;GET THE DATA
02A8	34CC	623	CALL FIRE ;STROBE THE SOLENDIDS
02AA	1C	624	INC TEMP1 ;INCREMENT THE POINTER
02AB	18	625	INC STBCNT ;INCREMENT THE STROBE COUNTER
02AC	F8	626	MOV A, STBCNT ;GET THE STROBE COUNTER
02AD	D305	627	XRL A, #05H ;IS IT FIVE
02AF	96A6	628	JNZ LKLO ;REPEAT IF NOT FIVE
02B1	04AE	629	JMP SETTIM ;GO BACK
02B3	FC	630	BAKWRD: MOV A, TEMP1 ;GET THE TARGET
02B4	0304	631	ADD A, #04H ;COMPENSATE FOR GOING BACKWARDS
02B6	AC	632	MOV TEMP1, A ;SAVE IT
02B7	FC	633	LKLO1: MOV A, TEMP1 ;GET THE TARGET
02B8	A3	634	MOV A, 0A ;GET THE DATA
02B9	34CC	635	CALL FIRE ;STROBE THE SOLENDIDS
02BB	FC	636	MOV A, TEMP1 ;GET TEMP1
02BC	07	637	DEC A ;DECREASE BY ONE
02BD	AC	638	MOV TEMP1, A ;PUT IT BACK
02BE	18	639	INC STBCNT ;INCREMENT THE STROBE COUNTER
02BF	F8	640	MOV A, STBCNT ;GET THE STROBE COUNTER
02CB	D305	641	XRL A, #05H ;IS IT FIVE
02C2	96B7	642	JNZ LKLO1 ;REPEAT IF NOT FIVE
02C4	04AE	643	JMP SETTIM ;GO BACK, CHARACTER IS DONE
		644	*EJECT

LDC	OBJ	SEQ	SOURCE STATEMENT
		645	;*
0300		646	ORG 300H
		647	;*
		648	
0300	00	649	DB 00H ;
0301	00	650	DB 00H ;
0302	00	651	DB 00H ;
0303	00	652	DB 00H ;
0304	00	653	DB 00H ;
		654	
0305	00	655	DB 00H ;
0306	00	656	DB 00H ;
0307	5F	657	DB 5FH ; * * * * *
0308	00	658	DB 00H ;
0309	00	659	DB 00H ;
		660	
030A	00	661	DB 00H ;
030B	07	662	DB 07H ; ***
030C	00	663	DB 00H ;
030D	07	664	DB 07H ; ***
030E	00	665	DB 00H ;
		666	
030F	14	667	DB 14H ; * *
0310	7F	668	DB 7FH ; * * * * *
0311	14	669	DB 14H ; * *
0312	7F	670	DB 7FH ; * * * * *
0313	14	671	DB 14H ; * *
		672	
0314	24	673	DB 24H ; * * *
0315	2A	674	DB 2AH ; * * *
0316	7F	675	DB 7FH ; * * * * *
0317	2A	676	DB 2AH ; * * *
0318	12	677	DB 12H ; * *
		678	
0319	23	679	DB 23H ; * * **
031A	13	680	DB 13H ; * * **
031B	00	681	DB 00H ; *
031C	64	682	DB 64H ; ** *
031D	62	683	DB 62H ; ** *
		684	
031E	36	685	DB 36H ; ** **
031F	49	686	DB 49H ; * * * *
0320	56	687	DB 56H ; * * **
0321	20	688	DB 20H ; *
0322	50	689	DB 50H ; * *
		690	*EJECT

LOC	OBJ	SEQ	SOURCE STATEMENT
		691	
0323	00	692	DB 00H :
0324	00	693	DB 00H :
0325	07	694	DB 07H : ***
0326	00	695	DB 00H :
0327	00	696	DB 00H :
		697	
0328	1C	698	DB 1CH : ****
0329	22	699	DB 22H : * * *
032A	41	700	DB 41H : * * *
032B	00	701	DB 00H :
032C	00	702	DB 00H :
		703	
032D	00	704	DB 00H :
032E	00	705	DB 00H :
032F	41	706	DB 41H : * * *
0330	22	707	DB 22H : * * *
0331	1C	708	DB 1CH : ****
		709	
0332	22	710	DB 22H : * * *
0333	14	711	DB 14H : * * *
0334	7F	712	DB 7FH : *****
0335	14	713	DB 14H : * * *
0336	22	714	DB 22H : * * *
		715	
0337	00	716	DB 00H : *
0338	00	717	DB 00H : *
0339	7F	718	DB 7FH : *****
033A	00	719	DB 00H : *
033B	00	720	DB 00H : *
		721	
033C	00	722	DB 00H :
033D	40	723	DB 40H : *
033E	30	724	DB 30H : **
033F	00	725	DB 00H :
0340	00	726	DB 00H :
		727	
0341	00	728	DB 00H : *
0342	00	729	DB 00H : *
0343	00	730	DB 00H : *
0344	00	731	DB 00H : *
0345	00	732	DB 00H : *
		733	
0346	00	734	DB 00H :
0347	00	735	DB 00H :
0348	40	736	DB 40H : *
0349	00	737	DB 00H :
034A	00	738	DB 00H :
		739	
034B	20	740	DB 20H : *
034C	10	741	DB 10H : *
034D	00	742	DB 00H : *
034E	04	743	DB 04H : *
034F	02	744	DB 02H : *
		745	
0350	3E	746	DB 3EH : *****
0351	51	747	DB 51H : * * * *
0352	49	748	DB 49H : * * * *
0353	45	749	DB 45H : * * * *
0354	3E	750	DB 3EH : *****
		751	
0355	00	752	DB 00H :
0356	42	753	DB 42H : * * *
0357	7F	754	DB 7FH : *****
0358	40	755	DB 40H : *
0359	00	756	DB 00H :
		757	
035A	62	758	DB 62H : * * *
035B	51	759	DB 51H : * * * *
035C	49	760	DB 49H : * * * *
035D	49	761	DB 49H : * * * *
035E	46	762	DB 46H : * * **
		763	
035F	21	764	DB 21H : * * *
0360	41	765	DB 41H : * * *





LDC	OBJ	SEQ	SOURCE STATEMENT	
B361	49	766	DB 49H	: * * *
B362	40	767	DB 40H	: * * * *
B363	33	768	DB 33H	: ** **
		769		
B364	18	770	DB 18H	: **
B365	14	771	DB 14H	: * * *
B366	12	772	DB 12H	: * * * *
B367	7F	773	DB 7FH	: * * * * *
B368	18	774	DB 18H	: *
		775		
B369	27	776	DB 27H	: * * * *
B36A	45	777	DB 45H	: * * * *
B36B	45	778	DB 45H	: * * * *
B36C	45	779	DB 45H	: * * * *
B36D	39	780	DB 39H	: * * * *
		781		
B36E	3C	782	DB 3CH	: * * * *
B36F	4A	783	DB 4AH	: * * * *
B370	49	784	DB 49H	: * * * *
B371	49	785	DB 49H	: * * * *
B372	31	786	DB 31H	: ** *
		787		
B373	81	788	DB 81H	: * *
B374	71	789	DB 71H	: * * * *
B375	89	790	DB 89H	: * * *
B376	85	791	DB 85H	: * * *
B377	83	792	DB 83H	: * *
		793		
B378	36	794	DB 36H	: * * * *
B379	49	795	DB 49H	: * * * *
B37A	49	796	DB 49H	: * * * *
B37B	49	797	DB 49H	: * * * *
B37C	36	798	DB 36H	: * * * *
		799	*EJECT	



LDC	OBJ	SEQ	SOURCE STATEMENT
		000	
037D	46	001	DB 46H ; * **
037E	49	002	DB 49H ; * * *
037F	49	003	DB 49H ; * * *
0380	29	004	DB 29H ; * * *
0381	1E	005	DB 1EH ; ****
		006	
0382	00	007	DB 00H ;
0383	00	008	DB 00H ;
0384	14	009	DB 14H ; * *
0385	00	010	DB 00H ;
0386	00	011	DB 00H ;
		012	
0387	00	013	DB 00H ;
0388	40	014	DB 40H ; *
0389	34	015	DB 34H ; ** *
038A	00	016	DB 00H ;
038B	00	017	DB 00H ;
		018	
038C	00	019	DB 00H ; *
038D	14	020	DB 14H ; * * *
038E	22	021	DB 22H ; * * *
038F	41	022	DB 41H ; * * *
0390	00	023	DB 00H ;
		024	
0391	14	025	DB 14H ; * *
0392	14	026	DB 14H ; * *
0393	14	027	DB 14H ; * *
0394	14	028	DB 14H ; * *
0395	14	029	DB 14H ; * *
		030	
0396	00	031	DB 00H ;
0397	41	032	DB 41H ; * * *
0398	22	033	DB 22H ; * * *
0399	14	034	DB 14H ; * * *
039A	00	035	DB 00H ; *
		036	
039B	02	037	DB 02H ; *
039C	01	038	DB 01H ; * * *
039D	59	039	DB 59H ; * * *
039E	05	040	DB 05H ; * * *
039F	02	041	DB 02H ; *
		042	\$EJECT



LOC	OBJ	SEQ	SOURCE STATEMENT	
03A0	0000	043	PAGE2: MOV STBCNT, #00H	:ZERO STROBE COUNTER
03A2	FA	044	MOV A, SAVPNT	:GET DIRECTION
03A3	37	045	CPL A	:FLIP BITS
03A4	0205	046	JBG BKWRD	:IF BACKWARD JUMP OUT
03A6	FC	047	LKHI: MOV A, TEMP1	:GET THE TARGET
03A7	0360	048	ADD A, #60H	:ADJUST THE TARGET
03A9	A3	049	MOV A, @A	:GET THE DATA
03AA	34CC	050	CALL FIRE	:STROBE THE SOLENOIDS
03AC	1C	051	INC TEMP1	:INCREMENT THE POINTER
03AD	10	052	INC STBCNT	:INCREMENT THE STROBE COUNTER
03AE	FB	053	MOV A, STBCNT	:GET THE STROBE COUNTER
03AF	0305	054	XRL A, #05H	:IS IT FIVE
03B1	96A6	055	JNZ LKHI	:REPEAT IF NOT FIVE
03B3	04AE	056	JMP SETTIM	:GO BACK
03B5	FC	057	BKWRD: MOV A, TEMP1	:GET THE TARGET
03B6	0364	058	ADD A, #64H	:COMPENSATE FOR GOING BACKWARDS
03B8	AC	059	MOV TEMP1, A	:SAVE IT
03B9	FC	060	LKHI1: MOV A, TEMP1	:GET THE TARGET
03BA	A3	061	MOV A, @A	:GET THE DATA
03BB	34CC	062	CALL FIRE	:STROBE THE SOLENOIDS
03BD	FC	063	MOV A, TEMP1	:GET TEMP1
03BE	07	064	DEC A	:DECREASE BY ONE
03BF	AC	065	MOV TEMP1, A	:PUT IT BACK
03C0	10	066	INC STBCNT	:INCREMENT THE STROBE COUNTER
03C1	FB	067	MOV A, STBCNT	:GET THE STROBE COUNTER
03C2	0305	068	XRL A, #05H	:IS IT FIVE
03C4	96B9	069	JNZ LKHI1	:REPEAT IF NOT FIVE
03C6	04AE	070	JMP SETTIM	:GO BACK. CHARACTER IS DONE
		071	#EJECT	

```

LOC  OBJ          SEQ          SOURCE STATEMENT
      872          ;
      873          ORG      400H
      874          ;
0400  27          875  BGIN:  CLR      A              ;ZERO ACC
0401  90          876          MOVX   @R0,A        ;TURN OFF THE SOLENOIDS
0402  9400        877          CALL  SETUP      ;SET UP THE PRINTER
0404  943F        878          CALL  VARSET     ;SET UP THE SOFTWARE
0406  040A        879          JMP    PRNT      ;GO START
      880          ;
0408  23FE        881  SETUP: MOV    A,#0FEH    ;LOAD ACC WITH VALUE TO TURN ON MOTOR
040A  39          882          OUTL  P1,A        ;TURN ON MOTOR
      883          ;
      884          ;NOW DELAY 3.2 SECONDS WHILE CHECKING RIGHT SENSOR
      885          ;
0408  BC05        886          MOV    TEMP1,#05H    ;LOAD DELAY VALUE ONE
040D  BFFF        887  SELFC: MOV    JUNK1,#0FFH    ;LOAD DELAY VALUE TWO
040F  BEFF        888  SELFB: MOV    LINCNT,#0FFH    ;LOAD DELAY VALUE THREE
0411  09          889  SELFA: IN    A,P1        ;READ PORT ONE
0412  37          890          CPL    A              ;MAKE THINGS RIGHT
0413  F21D        891          JB7   DONER      ;IS BIT 7 SET?
0415  EE11        892          DJNZ  LINCNT,SELFA ;SMALL LOOP
0417  EF0F        893          DJNZ  JUNK1,SELFB ;BIGGER LOOP
0419  EC0D        894          DJNZ  TEMP1,SELFC ;BIGGEST LOOP
041B  045A        895          JMP    ERROR      ;SOMETHING IS WRONG
      896          ;
      897          ;NOW MAKE SURE THE RIGHT SENSOR IS CLEARED
      898          ;
041D  BFFF        899  DONER: MOV    JUNK1,#0FFH    ;SET UP DELAY
041F  BEFF        900  SELF:  MOV    LINCNT,#0FFH    ;SOME MORE DELAY
0421  09          901  SELF1: IN    A,P1        ;GET THE FLAG INFORMATION
0422  F22A        902          JB7   DONEF      ;IS FLAG CLEARED?
0424  EE21        903          DJNZ  LINCNT,SELF1 ;IF NOT LOOP
0426  EF1F        904          DJNZ  JUNK1,SELF  ;LOOP SOME MORE
0428  045A        905          JMP    ERROR      ;LEAVE IF FLAG IS NOT UNCOVERED
      906          ;
      907          ;NOW CHECK THE LEFT SENSOR IN THE SAME MANNER AS THE
      908          ;RIGHT SENSOR, EXCEPT DELAY ONLY 2.5 SECONDS
      909          ;
042A  BC04        910  DONEF: MOV    TEMP1,#04H    ;LOAD DELAY 1
042C  BFFF        911  SELFC: MOV    JUNK1,#0FFH    ;LOAD DELAY 2
042E  BEFF        912  SELFB: MOV    LINCNT,#0FFH    ;LOAD DELAY 3
0430  09          913  SELFA: IN    A,P1        ;READ THE PORT
0431  37          914          CPL    A              ;CHANGE THINGS AROUND
0432  D23C        915          JB6   DONEL      ;OK IF BIT 6 IS A ZERO
0434  EE30        916          DJNZ  LINCNT,SELFA ;SMALL LOOP
0436  EF2E        917          DJNZ  JUNK1,SELFB ;BIGGER LOOP
0438  EC2C        918          DJNZ  TEMP1,SELFC ;BIGGEST LOOP
043A  045A        919          JMP    ERROR      ;SOMETHING IS WRONG
043C  0901        920  DONEL: ORL   P1,#01H    ;TURN MOTOR OFF
043E  03          921          RET                    ;GO BACK
      922          ;
      923          ;NOW SET UP THE VARIABLES
      924          ;
043F  23FE        925  VARSET: MOV    A,#0FEH    ;LOAD THE TIMER
0441  62          926          MOV    T,A
0442  55          927          START  T          ;START THE TIMER
0443  0020        928          MOV    INBUF,#FIRST ;LOAD INPUT BUFFER
0445  0000        929          MOV    LINCNT,#00H   ;SET LINE COUNT
0447  0000        930          MOV    STATUS,#00H   ;SET FORWARD BIT
      931          ;
      932          ;NOW CLEAR THE RAM AREA BY WRITING SPACE CODES
      933          ;
0449  0920        934          MOV    @OUTBUF,#FIRST ;LOAD OUTBUF
044B  2320        935  CLRMEM: MOV    A,#20H     ;PUT SPACE CODE IN ACC
044D  01          936          MOV    @OUTBUF,A     ;PUT SPACE CODE IN DATA MEMORY
044E  19          937          INC    @OUTBUF      ;UPDATE THE POINTER
044F  F9          938          MOV    A,@OUTBUF    ;MOVE THE POINTER INTO ACC
0450  D370        939          XRL   A,#0AH+1     ;SEE IF DONE
0452  0640        940          JNZ   CLRMEM       ;LOOP IF NOT CLEARED
      941          ;
      942          ;NOW CLEAR THE 0212
      943          ;
0454  99EF        944          ANL   P1,#0EFH     ;SET ENABLE BIT
0456  00          945          MOVX  A,@INBUF     ;CLEAR THE 0212 INPUT BUFFER
0457  0910        946          ORL   P1,#01H     ;RESET ENABLE BIT
      947          ;

```

```

LOC  OBJ      SEQ      SOURCE STATEMENT
      948      ;NOW EXIT VARSET
      949      ;
0459  83      950      RET              ;LEAVE INITIALIZATION
      951      ;
      952      ;THIS ROUTINE TURNS THE MOTOR OFF AND LOOPS
      953      ;
045A  89FF    954  ERROR: ORL      P1,0BFFH    ;TURN OFF MOTOR
045C  845C    955  DEAD:  JMP      DEAD        ;LOOP BECAUSE SOMETHING IS WRDNG
      956      ;
      957      ;THESE ARE ALL SUBROUTINES THAT ARE CALLED
      958      ;
045E  19      959  INCTST: INC      OUTBUF      ;UPDATE THE POINTER
045F  2378    960      MOV      A,#MAX+1    ;GET THE VALUE FOR THE LAST CHARACTER
0461  09      961      XRL      A,OUTBUF    ;DO THE TEST
0462  83      962      RET              ;EXIT
0463  09      963  GTPRNT: IN      A,P1      ;READ PORT ONE
0464  37      964      CPL      A              ;FLIP BITS
0465  0263    965      JNB      GTPRNT    ;LOOP UNTIL SENSOR IS UNCOVERED
0467  1668    966  TSTJTF: JTF      PIT          ;SEE IF TIMER FLAG IS SET
0469  8467    967      JMP      TSTJTF    ;TEST FLAG
0468  65      968  PIT:   STOP     TCNT        ;STOP THE TIMER
046C  FF      969      MOV      A,JUNK1     ;GET THE CHARACTER
046D  34C1    970      CALL     PRNTIT     ;PRINT THE CHARACTER
046F  341C    971      CALL     LNM00E     ;GET ANOTHER CHARACTER
0471  83      972      RET              ;EXIT
0472  F9      973  DECTST: MOV      A,OUTBUF    ;GET OUTBUF
0473  07      974      DEC      A              ;REDUCE BY ONE
0474  A9      975      MOV      OUTBUF,A    ;PUT BACK IN OUTBUF
0475  031F    976      XRL      A,#FIRST-1 ;SEE IF IT IS ALL THE WAY DOWN
0477  83      977      RET              ;EXIT
      978      ;
      979      ;THIS ROUTINE DOES A LINE FEED
      980      ;
0478  FE      981  LINEFD: MOV      A,LINCNT    ;GET THE LINE COUNT
0479  F298    982      JNB      DOFF        ;IF BIT 7 IS SET, DO A FORMFEED
047B  99FD    983  LFDO:  ANL      P1,0BFDH    ;TURN ON THE SOLENOID
047D  BC40    984      MOV      TEMP1,#40H    ;LOAD ONE DELAY
047F  BF93    985  LFLP1: MOV      JUNK1,#93H    ;LOAD ANOTHER DELAY
0481  EF81    986  LFLP2: DJNZ     JUNK1,LFLP2  ;LOOP
0483  EC7F    987      DJNZ     TEMP1,LFLP1  ;LOOP SOME MORE
0485  89B2    988      ORL      P1,0B2H      ;TURN OFF LF SOLENOID
0487  1E      989      INC      LINCNT      ;UPDATE THE LINE COUNTER
0488  FE      990      MOV      A,LINCNT    ;GET THE LINE COUNT
0489  0328    991      XRL      A,#28H      ;IS PAGE DONE
048B  968F    992      JNZ      NOTD00H     ;SKIP OVER
048D  BE88    993      MOV      LINCNT,#88H  ;ZERO LINE COUNTER
      994      ;
      995      ;NOW DELAY 90 MILLISECONDS
      996      ;
048F  BC88    997  NOTD0N: MOV      TEMP1,#88H    ;LOAD DELAY VALUES
0491  BFFF    998      LDP1:  MOV      JUNK1,0BFFH    ;
0493  EF93    999  LDP2:  DJNZ     JUNK1,LDP2    ;GENERATE DELAY
0495  EC91    1000     DJNZ     TEMP1,LDP1    ;
0497  83      1001     RET              ;LINE FEED IS DONE
      1002     ;
      1003     ;THIS ROUTINE DOES A FORM FEED
      1004     ;
0498  09      1005  DOFF:  IN      A,P1      ;GET THS STATUS
0499  37      1006     CPL      A              ;FLIP ACC
049A  53C8    1007     ANL      A,#0C8H      ;LEAVE ONLY TWO MSB'S
049C  C698    1008     JZ       DOFF        ;IF A FLAG ISN'T COVERED, LOOP
049E  8901    1009     ORL      P1,#01H      ;TURN THE MOTOR OFF
04A0  9478    1010     CALL     LFDO        ;GO DO ONE LINE FEED
04A2  FE      1011  FFCK:  MOV      A,LINCNT    ;GET THE LINE COUNT
04A3  537F    1012     ANL      A,#7FH      ;STRIP BIT SEVEN
04A5  0388    1013     XRL      A,#88H      ;IS IT DONE
04A7  C6A0    1014     JZ       FFDONE      ;LEAVE IF IT IS
04A9  9478    1015     CALL     LFDO        ;STRUBE THE SOLENOIDS
04AB  84A2    1016     JMP      FFCY        ;CHECK THE FORM FEED OUT
04AD  83      1017  FFDONE: RET              ;EXIT FORM FEED
      1018     ;
04AE  23E8    1019  SETTIM: MOV      A,#8E8H      ;GET DELAY VALUE
04B0  62      1020     MOV      T,A          ;PUT IN TIMER
04B1  55      1021     STRT   T              ;START THE TIMER
04B2  83      1022     RET              ;EXIT
      1023     ;
    
```



```

LOC  OBJ          SEQ          SOURCE STATEMENT
04B3  42          1024  PRNTBK:  MOV    A, T          ;GET THE TIMER
04B4  37          1025          CPL    A              ;TWO'S COMPLEMENT ACC
04B5  17          1026          INC    A
04B6  17          1027          INC    A
04B7  17          1028          INC    A
04B8  17          1029          INC    A
04B9  17          1030          INC    A          ;ADJUST TIMER
04BA  62          1031          MOV    T, A          ;PUT IT BACK IN THE TIMER
04BB  09          1032  INLOOP:  IN     A, P1         ;READ PORT 1
04BC  F2CB       1033          JB7    CONPBK        ;IF SENSOR IN HOT COVERED, LEAVE
04BE  04BB       1034          JMP    INLOOP        ;OTHERWISE LOOP
04CB  55          1035  CONPBK:  STRT   T              ;START THE TIMER
04C1  16C5       1036  CONPB:   JTF    RDTOPT      ;SEE IF READY TO PRINT
04C3  04C1       1037          JMP    CONPB         ;OTHERWISE LOOP
04C5  23FF       1038  RDTOPT:  MOV    A, #FFH      ;LOAD A
04C7  62          1039          MOV    T, A          ;PUT IT IN THE TIMER
04C8  03          1040          RET
;
1041          ;
1042          ;THIS ROUTINE ADJUSTS AND SAVES THE STATUS DURING PRINTING
1043          ;
04C9  FD          1044  STACHK:  MOV    A, STATUS     ;GET THE STATUS
04CA  32D2       1045          JB4    LFSET         ;SET LINE FEED BIT
04CC  AA          1046  B4RET:   MOV    SAVPNT, A  ;SAVE THE STATUS
04CD  53C2       1047          ANL    A, #BC2H      ;RESET EVERYTHING EXCEPT
1048          ;DIRECTION AND PRINT
04CF  AD          1049          MOV    STATUS, A     ;PUT THE STATUS BACK
04D0  0413       1050          JMP    LPRNT1        ;EXIT
04D2  4320       1051  LFSET:   ORL    A, #20H      ;SET BIT 5
04D4  04CC       1052          JMP    B4RET         ;JUMP BACK
1053          ;
1054          ;THIS ROUTINE READS A CHARACTER AND PUTS IT IN THE ACC
1055          ;
04D6  99EF       1056  GTCAR:   ANL    P1, #BEFH     ;SET ENABLE BIT
04D8  00          1057          MOVX   A, RINBUF     ;READ THE CHARACTER
04D9  091B       1058          ORL    P1, #1BH      ;RESET ENABLE BIT
04DB  03          1059          RET                  ;EXIT GTCAR
1060          ;
1061          ;THIS ROUTINE TURNS THE MOTOR ON
1062          ;
04DC  99FE       1063  MOTON:   ANL    P1, #BFEN     ;TURN MOTOR ON
04DE  03          1064          RET                  ;EXIT
1065          ;
1066          ;THIS ROUTINE TURNS THE MOTOR OFF
1067          ;
04DF  09B1       1068  MOTOF:   ORL    P1, #01H      ;TURN MOTOR OFF
04E1  03          1069          RET                  ;EXIT
1070          ;
1071          END                  ;DONE
    
```

USER SYMBOLS

ARND	0107	ARNDJP	0149	B4RET	04CC	BKWRD	0205	BGIN	0400	BKWRD	0305	BUTL0P	0113	BVEAVE	0160
CASE0	0031	CASE01	0017	CASE1	005C	CASE2	0000	CASE23	0024	CASE3	00C2	CHAP	011F	CLMEM	0446
CONPB	04C1	CONPBK	04CB	CPFIX	01AB	CPFND	00DC	CRFND	0062	DEAD	045C	DECTST	0472	DOFF	0490
COLF	0071	DONEF	0420	DONEL	043C	DONER	0410	EPRUP	0454	FDC	0042	FDC1	0044	FDCP	009E
FDCR1	0000	FFCK	044E	FFDNE	0400	FFFIX	0102	FINE	0170	FIRE	01CC	FIREX	0100	FIREY	0106
FIRST	0020	FIXDUN	0174	FIXFIN	010F	FIXUP	0109	FXCHAR	0161	FXPRNT	0191	GETSTA	0144	GOOD	0120
GTCAR	0406	GTPRNT	0463	INBUF	0000	INCSI	045E	INLOOP	0400	ISCHAR	0100	JUNK1	0007	KTDUN	01E0
LDBUF	0100	LFCKCK	0140	LPDD	0470	LPFIX	01AB	LPFL1	047F	LPFL2	0401	LFSET	0402	LFTEST	017F
LINCNT	0006	LINFED	0470	LKHI	03AK	LKHI1	0309	LKLO	0246	LKLO1	0207	LWODE	011C	LOOP4	0070
LOP1	0491	LOP2	0493	LPRNT	0011	LPRNT1	0015	MAX	006F	MOTD	040F	MOTON	040C	NOFF	010F
NOLF	0110	MOTD0N	049F	NT1	0104	OUTBUF	0001	OVR	000A	OVR1	0005	PAGE1	0240	PAGE2	0340
P1T	046B	PRNT	000A	PRNTBK	04B3	PRNTIT	01C1	RDTOPT	04C5	SAVPNT	0002	SEL	041F	SEL1	0421
SELFA	0411	SELFAA	0430	SELFB	040F	SELFB0	042E	SELFC	0400	SELFC0	042C	SETTIM	044E	SETUP	0400
SHOKT	01CA	STACHK	04C9	STATUS	0005	STCNT	0002	STBIT1	0150	STPRNT	0159	SUB1	0139	TABLE1	0200
TEMP1	0004	TSJTF	0101	TSJTF	0467	VARSET	043F	WATCH	0075	WATCH0	000E				

ASSEMBLY COMPLETE. NO ERRORS