Appendix F

Atari 400/800 Computers

The best way to connect your Atari to Delta is with the **Universal/Atari Parallel Interface** by Star Micronics. It comes complete with its own cable. Or Delta will connect to the Atari 850 interface, using a cable that is available from your dealer.

Setting the switches

We recommend that you set the DIP switches in Delta as shown below when connecting it to an Atari 400 or 800.

Recommended DIP Switch Settings for Atari

Switch	Setting	Function
1-1	ON	11 inch page size
1-2	ON	Normal print density
1-3	ON	10 CPI pitch
1-4	ON	Normal characters
1-5	ON	1/6 inch line feed
1-6	ON	U.S.A. Character set
1-7	ON	
1-8	ON	
2-1	ON	Paper-out detector active
2-2	OFF	Parallel interface
2-3	OFF	8-bit interface
2-4	ON	Auto line feed

Using Atari BASIC

While the Atari computers don't have any real problems in using the full capabilities of Delta, there are a couple of fairly unique things to keep in mind. Atari BASIC requires that *all* strings be dimensioned. The maximum string length is 99 characters, so Atari users must break up their dot graphics strings into 99 character sections.

Atari 850 Interface Module Parallel Cable

Delta			Atari 850	
Pin No.	**Function**		**Pin No.**	**Function**
1	STROBE	———	1	STROBE
2	DATA1	———	2	DATA1
3	DATA2	———	3	DATA2
4	DATA3	———	4	DATA3
5	DATA4	———	5	DATA4
6	DATA5	———	6	DATA5
7	DATA6	———	7	DATA6
8	DATA7	———	8	DATA7
16	SIG GND	———	11	SIG GND
11	BUSY	———	13	BUSY
9	DATA8	———	15	DATA8

To join two strings together, as when building a string of dot graphics data, the following format must be used:

```
A$(LEN(A$)+1)=B$
```

(This adds B$ to the end of A$.)

The Atari adds spaces to print lines when you use the LPRINT command. We recommend that you use the PRINT # command instead. You must open the printer as a device first. For example:

```
10 OPEN #4,8,0,"P"
20 PRINT #4;"TESTING"
```

It's a good idea to close the printer when your program is done using it, like this:

```
90 CLOSE #4
```

Atari BASIC also requires that you use semicolons between elements in a print statement where most BASICs will accept a

space. Your print commands must look like this:

```
40 PRINT CHR$(27); "B"; CHR$(3); "CONDENSED"
```

Listing programs

Listing BASIC programs to Delta from an Atari computer is relatively easy; just add "P:" to the normal LIST command so that it looks like this:

```
LIST "P:"
```

Program listings

The following programs were translated to work with the Atari 400 and 800 computers.

Chart program

```
10 REM ATARI 400 & 800 & 1200XL
20 DIM BL$(6),FD$(4)
30 TRUE=1:FALSE=0
100 OPEN #4,8,0,"P"
120 GOSUB 1000
130 PRINT #4;"*REGULAR*"
140 GOSUB 2000
150 PRINT #4;"*DOUBLE STRIKE*"
160 PRINT #4;CHR$(27);"G";
170 GOSUB 2000
180 PRINT #4;"*EMPHASIZED*":
190 EM=TRUE
200 GOSUB 2000
210 PRINT #4;"*DOUBLE STRIKE & EMPHASIZED*"
220 PRINT #4;CHR$(27);"G";CHR$(27);"E";
230 GOSUB 2000
240 CLOSE #4
250 END
1000 PRINT #4;CHR$(27);"@"
1010 PRINT #4;CHR$(14);"   NORMAL      ENLARGED   "
1020 PRINT #4;CHR$(27);"-";CHR$(1);
1030 PRINT #4;CHR$(27);"B";CHR$(3);"CONDENSED ";
1040 PRINT #4;CHR$(27);"B";CHR$(2);"  ELITE   ";
1050 PRINT #4;CHR$(27);"B";CHR$(1);"  PICA    ";
1060 PRINT #4;CHR$(27);"B";CHR$(3);"CONDENSED ";
1070 PRINT #4;CHR$(27);"B";CHR$(2);"  ELITE   ";
```

```
1Ø8Ø PRINT #4;CHR$(27);"B";CHR$(1);"  PICA    "
1Ø9Ø PRINT #4;CHR$(27);"@":RETURN
2ØØØ IT=FALSE:PRINT #4;CHR$(27);"5";
2Ø1Ø UN=FALSE:PRINT #4;CHR$(27);"-";CHR$(Ø);
2Ø2Ø EN=FALSE:PRINT #4;CHR$(27);"W";CHR$(Ø);
2Ø3Ø PI=FALSE
2Ø4Ø PRINT #4;CHR$(27);"B";CHR$(3);:GOSUB 213Ø
2Ø5Ø PRINT #4;CHR$(27);"B";CHR$(2);:GOSUB 213Ø
2Ø6Ø PRINT #4;CHR$(27);"B";CHR$(1);:PI=TRUE:GOSUB
   213Ø
2Ø7Ø IF EN=TRUE THEN PRINT #4:GOTO 2Ø9Ø
2Ø8Ø PRINT #4;CHR$(27);"W";CHR$(1);:EN=TRUE:GOTO
   2Ø3Ø
2Ø9Ø IF UN=TRUE THEN PRINT #4:GOTO 211Ø
21ØØ PRINT #4;CHR$(27);"-";CHR$(1);:UN=TRUE:GOTO
   2Ø2Ø
211Ø IF IT=TRUE THEN PRINT #4;CHR$(27);"@":RETURN
212Ø PRINT #4;CHR$(27);"4";:IT=TRUE:GOTO 2Ø1Ø
213Ø BL$="       ":FD$="...."
214Ø IF EM=FALSE THEN PRINT #4;"ABcd";:GOTO 219Ø
215Ø IF PI=FALSE THEN PRINT #4;FD$;:GOTO 217Ø
216Ø PRINT #4;"ABcd";
217Ø IF EN=TRUE THEN PRINT #4;" ";:RETURN
218Ø IF EN=FALSE THEN PRINT #4;BL$;:RETURN
219Ø IF EN=TRUE THEN PRINT #4;" ";:RETURN
22ØØ PRINT #4;CHR$(27);"S";CHR$(Ø);"Xx";
221Ø PRINT #4;CHR$(27);"S";CHR$(1);"Yy";"  ";
222Ø PRINT #4;CHR$(27);"T";
223Ø RETURN
```

Special character chart program

```
1Ø REM PRINT SPECIAL CHAR.SET
2Ø OPEN #4,8,Ø, "P"
3Ø FOR J=16Ø to 255 STEP 8
4Ø FOR I=J TO J+7
5Ø PRINT #4;I;"= ";CHR$(I);"   ";
6Ø NEXT I:PRINT #4:NEXT J
7Ø CLOSE #4
```

Macro program

```
5 REM DEFINE MACRO INSTRUCTION
1Ø OPEN #4,8,Ø,"P"
2Ø PRINT #4,CHR$(27);"+";
3Ø PRINT #4,CHR$(18);
```

```
40 PRINT #4,CHR$(27);"W";CHR$(0);
50 PRINT #4,CHR$(27);"F";
60 PRINT #4,CHR$(27);"H";
70 PRINT #4,CHR$(27);"-";CHR$(0);
80 PRINT #4,CHR$(27);"T";
90 PRINT #4,CHR$(27);"5";
95 PRINT #4,CHR$(30)
```

Bridge hand program

```
10 OPEN #4,8,0,"P"
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
60 CLOSE #4
70 END
1000 REM INITIALIZE VARIABLES
1010 DIM HA(4),DE(52),CA$(50),SU$(20)
1020 SU$="SHDC"
1030 CA$="   2  3  4  5  6  7  8  9  10 J  Q  K  A
   "
1035 FOR I=0 TO 4:HA(I)=0:NEXT I
1040 RETURN
2000 REM INITIALIZE PRINTER
2010 PRINT #4;CHR$(27);CHR$(68);CHR$(20);CHR$(40);
   CHR$(0)
2020 PRINT #4;CHR$(27);CHR$(43);CHR$(27);CHR$(36);
   CHR$(0);
2030 PRINT #4;CHR$(27);CHR$(69);CHR$(30)
2035 PRINT #4;CHR$(27);CHR$(42);CHR$(0)
2040 FOR I=1 TO 4
2050 PRINT #4;CHR$(27);CHR$(42);CHR$(1);
2060 FOR J=1 TO 13
2070 READ X:PRINT #4;CHR$(X);
2080 NEXT J
2090 NEXT I
2100 PRINT #4
2110 RETURN
2120 DATA 72,11,4,10,20,10,52,72,52,10,20,10,4
2130 DATA 83,11,16,8,20,8,86,41,86,8,20,8,16
2140 DATA 67,11,8,16,8,18,65,62,65,18,8,16,8
2150 DATA 68,11,8,0,28,0,62,65,62,0,28,0,8
3000 REM DEAL CARD
3010 FOR CA=1 TO 52
```

```
3020 X=INT(RND(0)*4+1)
3030 IF HA(X)=13 THEN 3020
3035 HA(X)=HA(X)+1
3040 DE(CA)=X
3050 NEXT CA
3060 RETURN
4000 REM PRINT FOUR HANDS
4010 PRINT #4;CHR$(27);"!";CHR$(9);"NORTH"
4020 PRINT #4;CHR$(27);"$";CHR$(1);CHR$(27);
   CHR$(70);
4030 HA=1
4040 FOR SU=0 TO 3
4050 PRINT #4;CHR$(9);
4060 GOSUB 4300
4070 PRINT #4
4080 NEXT SU
4090 PRINT #4;CHR$(27);"!";"WEST";CHR$(9);CHR$(9);
   "EAST"
4100 PRINT #4;CHR$(27);"$";CHR$(1);CHR$(27);
   CHR$(70);
4110 FOR SU=0 TO 3
4120 HA=2
4130 GOSUB 4300
4140 PRINT #4;CHR$(9);CHR$(9);
4150 HA=3
4160 GOSUB 4300
4170 PRINT #4
4180 NEXT SU
4190 PRINT #4;CHR$(27);"!";CHR$(9);"SOUTH"
4200 PRINT #4;CHR$(27);"$";CHR$(1);CHR$(27);
   CHR$(70);
4210 HA=4
4220 FOR SU=0 TO 3
4230 PRINT #4;CHR$(9);
4240 GOSUB 4300
4250 PRINT #4
4260 NEXT SU
4270 PRINT #4;CHR$(27);"$";CHR$(0);CHR$(27);CHR$(70)
4280 RETURN
4290 REM PRINT ONE LINE
4300 PRINT #4;SU$(SU+1,SU+1);
4310 FOR CA=13 TO 1 STEP -1
4320 IF DE(SU*13+CA)=HA THEN PRINT #4;
   CA$(CA*3,CA*3+2);
```

```
4330 NEXT CA
4340 RETURN
```

Numeral program

```
10 REM PROGRAM TO DEFINE AND PRINT NUMERALS
20 OPEN #4,8,0,"P"
30 FOR N1=160 TO 200
40 PRINT #4;CHR$(27);CHR$(42);CHR$(1);
50 PRINT #4;CHR$(N1);
60 READ N2
70 PRINT #4;CHR$(N2);
80 FOR S=1 TO 11
90 READ MS
100 PRINT#4;CHR$(MS);
110 NEXT S
120 NEXT N1
130 PRINT #4;CHR$(27);CHR$(88);CHR$(1)
135 PRINT #4;CHR$(27);"1"
140 FOR I=160 TO 200 STEP 4
150 PRINT #4;CHR$(I);CHR$(I+1);CHR$(200);
160 NEXT I
165 PRINT #4
170 FOR I=162 TO 200 STEP 4
180 PRINT #4;CHR$(I);CHR$(I+1);CHR$(200);
190 NEXT I
200 PRINT #4;CHR$(27);"@"
210 CLOSE #4
220 END
340 REM ZERO
350 DATA 11,0,96,16,104,16,44,30,14,0,2,1
360 DATA 11,2,1,2,1,6,8,38,88,32,8,32
370 DATA 11,3,12,9,12,51,0,96,0,96,0,96
380 DATA 11,0,32,0,48,0,28,3,12,3,4,3
390 REM ONE
400 DATA 11,0,0,0,0,0,4,0,4,0,4,126
410 DATA 9,12,114,12,114,12,2,0,0,0,0,0
420 DATA 11,64,0,64,0,64,0,64,32,80,47,80
430 DATA 9,47,80,47,64,0,64,0,64,0,0,0
440 REM TWO
450 DATA 11,0,0,0,0,0,0,12,16,14,0,6,0
460 DATA 11,3,0,3,0,70,56,70,56,4,24,0
470 DATA 11,64,0,64,32,64,32,80,32,80,40,64
480 DATA 11,44,64,38,65,34,65,32,80,32,88,0
490 REM THREE
```

```
500 DATA 11,0,0,0,0,0,0,4,2,4,2,4
510 DATA 11,34,84,34,92,34,76,34,68,2,64,0
520 DATA 11,16,0,48,0,56,64,48,64,32,64,32
530 DATA 11,64,32,64,48,9,54,22,9,6,1
540 REM FOUR
550 DATA 11,0,0,0,0,0,0,64,36,88,32,16
560 DATA 11,0,0,64,32,64,56,64,60,2,12,0
570 DATA 11,0,8,4,10,5,10,5,8,4,72,4
580 DATA 11,88,38,89,38,89,6,73,4,8,6,0
590 REM FIVE
600 DATA 11,0,0,0,0,64,32,84,50,76,34,68
610 DATA 10,34,68,34,68,34,68,2,68,2,0,0
620 DATA 10,0,32,24,101,24,97,0,64,0,64,0
630 DATA 11,64,0,96,1,48,15,48,15,16,15,0
640 REM SIX
650 DATA 11,0,96,0,112,0,120,0,92,0,102,0
660 DATA 11,98,0,98,0,98,0,70,0,14,0,6
670 DATA 11,7,8,23,8,55,8,99,0,65,0,64
680 DATA 11,0,96,0,112,1,62,1,30,1,14,0
690 REM SEVEN
700 DATA 11,0,16,8,6,8,6,8,6,8,6,8
710 DATA 9,70,8,102,8,54,8,6,0,2,0,0
720 DATA 11,0,64,0,96,0,120,0,124,0,30,1
730 DATA 9,6,1,0,0,0,0,0,0,0,0,0
740 REM EIGHT
750 DATA 11,0,0,0,0,24,36,24,102,24,102,0
760 DATA 11,67,0,67,0,99,28,34,28,34,28,0
770 DATA 11,12,18,44,19,108,19,96,1,64,0,64
780 DATA 11,0,96,1,112,15,48,15,16,14,0,0
790 REM NINE
800 DATA 11,0,0,120,4,120,6,120,6,0,3,0
810 DATA 11,3,0,3,0,67,4,123,4,122,4,120
820 DATA 11,48,0,56,0,113,0,99,0,99,0,99
830 DATA 11,0,115,0,57,0,31,0,15,0,7,0
840 REM SPACE
850 DATA 11,0,0,0,0,0,0,0,0,0,0,0
```

Download utility program

```
5 DIM CS$(1),SC$(1),Z(9,13),MM(11),MM$(11),
   SS$(1),BL$(40),SH$(35),PR$(35)
6 DIM PD$(35),AS$(35),AN$(1)
10 CS$="@":SC$="*":SS$="0"
15 BL$="  (40 characters)  "
16 AS$="ASCII (33-126)  (160-254) --> "
17 SH$="IF SHIFTED ENTER 1 ELSE 0 --> "
```

```
18 PR$="NORMAL OR PROPORTIONAL --> "
19 PD$="PROPORTIONAL DATA (4-11) --> "
20 GRAPHICS 0:GOSUB 660
30 GOSUB 2000
40 IF KEY=47 THEN GRAPHICS 0:END
50 IF KEY=10 THEN GOSUB 680:GOTO 30
60 IF KEY=42 THEN GOSUB 900:GOSUB 260:GOTO 30
70 GOTO 30
120 IF Z(G,H)=0 THEN POSITION Y+5,X+2:PRINT
   " ";:GOSUB 3000
130 IF Z(G,H)=1 THEN POSITION Y+5,X+2:PRINT SC$;:
   GOSUB 3000
140 RETURN
150 IF Z(G,H)=1 THEN POSITION Y+5,X+2:PRINT SS$;:
   GOSUB 3000
160 IF Z(G,H)=0 THEN POSITION Y+5,X+2:PRINT CS$;:
   GOSUB 3000
170 RETURN
260 REM EDIT LEVEL
265 X=1:Y=1:G=1:H=1
267 FOR I=1 TO 11:MM(I)=0:NEXT I
268 FOR I=1 TO 12:FOR J=1 TO 8:Z(J,I)=0:NEXT J:NEXT
   I
270 GOSUB 2000
280 IF KEY=0 THEN GOSUB 390:GOTO 370
290 IF KEY=40 THEN GOSUB 410:GOTO 370
300 IF KEY=58 THEN GOSUB 430:GOTO 370
310 IF KEY=11 THEN GOSUB 450:GOTO 370
320 IF KEY=13 THEN GOSUB 470:GOTO 370
330 IF KEY=18 THEN GOSUB 490:GOTO 370
340 IF KEY=47 THEN GOSUB 500:GOTO 380
370 GOTO 270
380 RETURN
390 GOSUB 120:Y=Y-3:H=H-1:IF Y<1 THEN Y=1:H=1
400 GOSUB 150:RETURN
410 GOSUB 120:Y=Y+3:H=H+1:IF Y>31 THEN Y=31:H=11
420 GOSUB 150:RETURN
430 GOSUB 120:X=X+2:G=G+1:IF X>13 THEN X=13:G=7
440 GOSUB 150:RETURN
450 GOSUB 120:X=X-2:G=G-1:IF X<1 THEN X=1:G=1
460 GOSUB 150:RETURN
470 IF Z(G,H-1)=1 OR Z(G,H+1)=1 THEN RETURN
480 Z(G,H)=1:POSITION Y+5,X+2:PRINT SS$;:GOSUB 3000:
   RETURN
```

```
490 Z(G,H)=0:POSITION Y+5,X+2:PRINT CS$;:GOSUB 3000:
   RETURN
500 REM GET OUT OF EDIT MODE
510 IF Z(G,H)=1 THEN POSITION Y+5,X+2:PRINT SC$;:
   GOSUB 3000
520 IF Z(G,H)=0 THEN POSITION Y+5,X+2:PRINT
   " ";:GOSUB 3000
525 GOSUB 4000
530 FOR I=1 TO 11:FOR J=1 TO 7:
   MM(I)=MM(I)+Z(J,I)*(INT (2^(J-1))+1)
535 NEXT J:NEXT I
536 GOSUB 5000
540 FOR I=1 TO 11 POSITION 2+I*3,18:PRINT MM(I);:
   NEXT I
550 GOSUB 660:RETURN
660 FOR I=1 TO 7:FOR J=1 TO 11:Z(I,J)=0:NEXT J:NEXT
   I
670 POSITION 1,23:PRINT
   "E)DIT P)RINTER Q)UIT                ";
675 RETURN
680 REM PRINT MODE
690 GOSUB 7000:PRINT PR$;:INPUT AN$
700 IF AN$="N" THEN PR=0:GOTO 750
710 IF AN$="P" THEN 730
720 GOTO 690
730 GOSUB 7000:PRINT PD$;:INPUT PR
740 IF PR<4 OR PR>11 THEN 730
750 GOSUB 7000:PRINT SH$;:INPUT SH
760 IF SH<0 OR SH>1 THEN GOTO 750
770 GOSUB 7000:PRINT AS$;:INPUT AS
780 IF (AS<33 OR AS>126) AND (AS<160 OR AS>254) THEN
   770
790 GOSUB 7000
800 IF SH=1 THEN SH=16
810 N1=AS:N2=PR+SH
820 FOR I=1 TO 11:MM$(LEN(MM$)+1)=CHR$(MM(I)):NEXT I
830 OPEN #4,8,0,"P"
835 PRINT #4,CHR$(27);"*";CHR$(1);CHR$(N1);CHR$(N2);
   MM$
840 IF AN$="N" THEN PRINT #4;CHR$(27);"$";CHR$(1):
   GOTO 860
850 PRINT #4;CHR$(27);"X";CHR$(1)
860 FOR I=1 TO 20:PRINT #4;CHR$(N1);" ";:NEXT I:
   PRINT #4
```

```
870 PRINT #4;CHR$(14);:FOR I=1 TO 10:PRINT #4;
   CHR$(N1);" ";:NEXT I:PRINT #4
880 PRINT #4;CHR$(15);:FOR I=1 TO 20:PRINT #4;
   CHR$(N1);" ";:NEXT I:PRINT #4
890 IF AN$="N" THEN PRINT #4;CHR$(27);"$";CHR$(0):
   GOTO 895
892 PRINT #4;CHR$(27);"X";CHR$(0)
895 PRINT #4;CHR$(27);"@":CLOSE #4:GOSUB 660:MM$="":
   RETURN
900 GRAPHICS 0
904 PRINT
905 PRINT "  M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11"
910 FOR I=0 TO 7:PRINT "  ";:FOR J=1 TO 11
915 PRINT "!--";:NEXT J:PRINT "!":IF I<7 THEN PRINT
   INT(2^I)+1:NEXT I
920 PRINT :PRINT :PRINT
930 PRINT "R)IGHT L)EFT D)OWN U)P "
940 PRINT "I)NSERT   C)LEAR   Q)UIT"
950 POSITION 6,3:PRINT CS$;
955 GOSUB 3000
960 RETURN
2000 REM SINGLE CHAR INPUT
2010 KEY=PEEK(764):IF KEY=255 THEN 2010
2020 POKE 764,255
2030 RETURN
3000 POSITION 35,21
3010 PRINT " ";
3020 RETURN
4000 POSITION 15,18
4010 PRINT "PLEASE STAND BY";
4020 RETURN
5000 POSITION 1,18
5010 FOR I=1 TO 5:PRINT BL$;:NEXT I
5020 RETURN
7000 POSITION 1,20
7010 PRINT BL$;
7020 POSITION 1,20
7030 RETURN
```

Delta Plot program

```
2 GRAPHICS 0
3 PRINT " ":PRINT " ":PRINT " "
4 PRINT " THIS PROGRAM TAKES ABOUT"
5 PRINT " 1 MINUTE  TO RUN, PLEASE"
6 PRINT " TURN ON YOUR PRINTER AND "
```

```
7 PRINT " STAND BY ..............."
8 PRINT " ":PRINT " ":PRINT " "
10 FOR I=1536 TO 1553
20 READ O
30 POKE I,O
40 NEXT I
50 DATA 104,104,133,205,104,133,204
60 DATA 104,5,205,133,213,104,5,204
70 DATA 133,212,96
100 REM MICRO-PLOT
110 DIM M(76,14),MASK(6)
120 DIM A$(100)
130 DIM B$(10)
132 FOR I=0 TO 14
133 FOR J=1 TO 76
134 M(J,I)=0
135 NEXT J
136 NEXT I
1000 REM SET PROGRAM CONSTANT
1010 MASK(1)=128:MASK(4)=16
1020 MASK(2)=64:MASK(5)=8
1030 MASK(3)=32:MASK(6)=4
1040 LX=20:LY=20
1050 XFAC=72/LX:YFAC=87/LY
2000 REM PLOT CURVE
2010 LET RAD=9
2020 X1=19:Y1=10
2030 FOR ANG=0 TO 360 STEP 10
2040 R1=ANG*6.28/360
2050 X2=RAD*COS(R1)+10
2055 Y2=RAD*SIN(R1)+10
2060 GOSUB 4000
2070 NEXT ANG
3000 REM SEND BIT IMAGE TO PRINTER
3005 OPEN #4,8,0,"P"
3010 PRINT #4;CHR$(27);"A";CHR$(6)
3020 FOR ROW=0 TO 14
3030 A$=""
3040 PRINT #4;CHR$(27);"K";CHR$(75);CHR$(0);
3050 FOR COL=1 TO 75
3054 RE=INT(M(COL,ROW))
3055 B$=CHR$(RE)
3060 A$(LEN(A$)+1)=B$
3070 NEXT COL
3080 PRINT #4;A$;" "
```

```
3090 NEXT ROW
3100 PRINT #4;CHR$(27);"A";CHR$(12)
3110 CLOSE #4
3150 END
4000 REM DRAW A LINE FROM X1,Y1 TO X2,Y2
4010 XL=X2-X1:YL=Y2-Y1
4020 NX=ABS(XL*XFAC):NY=ABS(YL*YFAC)
4030 IF NX<NY THEN NX=NY
4040 NS=INT(NX+1)
4050 DX=XL/NS
4055 DY=YL/NS
4060 FOR I=1 TO NS
4070 X1=X1+DX:Y1=Y1+DY
4080 GOSUB 5000
4090 NEXT I
4100 RETURN
5000 REM PLOT A POINT AT X1,Y1
5010 XX=X1*XFAC:YY=Y1*YFAC
5020 COL=INT(XX)+1
5030 ROW=INT(YY/6)
5040 XIT=INT(YY-ROW*6)+1
5050 A1=M(COL,ROW)
5060 A2=MASK(XIT)
5070 M(COL,ROW)=USR(1536,A1,A2)
5080 RETURN
```

Pie chart program

```
2 GRAPHICS 0
3 PRINT "PLEASE STAND BY"
10 FOR I=1536 TO 1553
20 READ O
30 POKE I,O
40 NEXT I
50 DATA 104,104,133,205,104,133,204
60 DATA 104,5,205,133,213,104,5,204
70 DATA 133,212,96
100 REM MICRO-PLOT
110 DIM M(76,11),MASK(6),BL$(80),N$(99)
120 DIM A$(100),T$(99),TT(20),RR(20),TI$(99)
130 DIM B$(10),PCT(10),T1$(99),T2$(99)
132 FOR I=0 TO 11
133 FOR J=1 TO 76
134 M(J,I)=0
135 NEXT J
136 NEXT I
```

```
140 GOSUB 7000
141 GRAPHICS 0
142 PRINT " ":PRINT " ":PRINT " "
143 PRINT " THIS PROGRAM TAKES ABOUT"
144 PRINT " 2 MINUTES TO RUN, PLEASE"
145 PRINT " TURN ON YOUR PRINTER AND "
146 " STAND BY ............."
147 PRINT " ":PRINT " ":PRINT " "
150 T1$="  (40 characters)  "
160 T1$(LEN(T1$)+1)=T1$
170 T2$=T1$
180 BL$="  (40 characters)  "
190 BL$(LEN(BL$)+1)=BL$
195 FOR I=1 TO 20:TT(I)=1:NEXT I
198 FOR I=1 TO 20:RR(I)=0:NEXT I
1000 REM SET PROGRAM CONSTANT
1010 MASK(1)=128:MASK (4)=16
1020 MASK(2)=64:MASK(5)=8
1030 MASK(3)=32:MASK(6)=4
1040 LX=20:LY=20
1050 XFAC=72/LX:YFAC=75/LY
2000 REM PLOT CURVE
2010 LET RAD=9
2020 X1=19:Y1=10
2030 FOR ANG=0 TO 360 STEP 5
2040 R1=ANG*6.28/360
2050 X2=RAD*COS(R1)+10
2055 Y2=RAD*SIN(R1)+10
2060 GOSUB 4000
2070 NEXT ANG
2080 FOR PI=1 TO NP
2090 X1=10:Y1=10
2100 TP=TP+PCT(PI)
2110 ANG=360*TP*0.01
2120 R1=ANG*6.28/360
2130 X2=RAD*COS(R1)+10:Y2=RAD*SIN(R1)+10
2140 GOSUB 4000
2160 GOSUB 6000
2170 NEXT PI
2180 IF LEN(T1$)<99 THEN T1$(LEN(T1$)+1)=" ":GOTO
   2180
2190 IF LEN(T2$)<99 THEN T2$(LEN(T2$)+1)=" ":GOTO
   2190
3000 REM SEND BIT IMAGE TO PRINTER
3005 OPEN #4,8,0,"P"
```

```
3010 PRINT #4;CHR$(27);"A";CHR$(3)
3012 WW=LEN(TI$)
3013 VV=INT((80-WW)/2)
3014 PRINT #4;BL$(1,VV);TI$:FOR I=1 TO 25:PRINT #4:
   NEXT I
3015 PRINT #4;BL$(1,TT(1));T1$(1,9);CHR$(10)
3016 PRINT #4;BL$(1,TT(2));T1$(10,19);CHR$(10)
3017 PRINT #4;BL$(1,TT(3));T1$(20,29);CHR$(10)
3020 FOR ROW=0 TO 11
3030 A$=""
3035 PRINT #4;BL$(1,35);
3040 PRINT #4;CHR$(27);"K";CHR$(75);CHR$(0);
3050 FOR COL=1 TO 75
3054 RE=INT(M(COL,ROW))
3055 B$=CHR$(RE)
3060 A$(LEN(A$)+1)=B$
3070 NEXT COL
3080 PRINT #4;A$
3085 IF ROW>5 THEN GOSUB 8000:GOTO 3090
3086 HH=(ROW+3)*10
3087 PRINT #4;BL$(1,TT(ROW+4));T1$(HH,HH+9)
3090 NEXT ROW
3095 PRINT #4;BL$(1,TT(16));T2$(60,69);CHR$(10);
3096 PRINT #4;BL$(1,TT(17));T2$(70,79);CHR$(10);
3097 PRINT #4;BL$(1,TT(18));T2$(80,89)
3100 PRINT #4;CHR$(27);"A";CHR$(12)
3110 CLOSE #4
3150 END
4000 REM DRAW A LINE FROM X1,Y1 TO Y2,Y2
4010 XL=X2-X1:YL=Y2-Y1
4020 NX=ABS(XL*XFAC):NY=ABS(YL*YFAC)
4030 IF NX<NY THEN NX=NY
4040 NS=INT(NX+1)
4050 DX=XL/NS
4055 DY=YL/NS
4060 FOR I=1 TO NS
4070 X1=X1+DX:Y1=Y1+DY
4080 GOSUB 5000
4090 NEXT I
4100 RETURN
5000 REM PLOT A POINT AT X1,Y1
5010 XX=X1*XFAC:YY=Y1*YFAC
5020 COL=INT(XX)+1
5030 ROW=INT(YY/6)
5040 XIT=INT(YY-ROW*6)+1
```

```
5Ø5Ø A1=M(COL,ROW)
5Ø6Ø A2=MASK(XIT)
5Ø7Ø M(COL,ROW)=USR(1536,A1,A2)
5Ø8Ø RETURN
6ØØØ REM
6Ø1Ø MA=(ANG+PA)/2
6Ø2Ø R1=MA*6.28/36Ø
6Ø3Ø X3=INT(8*SIN(R1))
6Ø35 Y3=INT(1Ø*COS(R1))
6Ø4Ø X4=1Ø+X3:Y4=4Ø+Y3
6Ø41 GOSUB 9ØØØ:RR(PI)=X4
6Ø45 IF MA>27Ø OR MA<9Ø THEN TT(X4)=Y4:GOTO 6Ø5Ø
6Ø47 TT(X4)=Y4-1Ø
6Ø5Ø IF X4>9 THEN GOSUB 65ØØ:GOTO 6Ø6Ø
6Ø52 DD=(X4-1)*1Ø+1
6Ø54 DF=(PI-1)*1Ø+1
6Ø56 T1$(DD,DD+9)=T$(DF,DF+9)
6Ø6Ø PA=ANG
6Ø7Ø RETURN
65ØØ X4=X4-9
652 DD=(X4-1)*1Ø+1
654 DF=(PI-1)*1Ø+1
656 T2$(DD,DD+9)=T$(DF,DF+9)
658 RETURN
7ØØØ GRAPHICS Ø
7ØØ1 PRINT "TITLE CAN BE UP TO 8Ø CHARACTERS LONG"
7ØØ2 PRINT "ENTER TITLE  ";:INPUT TI$
7ØØ4 IF LEN(TI$)>8Ø THEN TI$=TI$(1,8Ø)
7ØØ5 AS=Ø:AL=1ØØ:FL=9:OO=1
7Ø1Ø GRAPHICS Ø
7Ø2Ø PRINT "YOU CAN HAVE UP TO 9 FIELDS AND EACH
   FIELD CAN BE UP TO NINE CHARACTERS LONG"
7Ø25 IF LEN(T$)<99 THEN T$(LEN(T$)+1)=" ":GOTO 7Ø25
7Ø3Ø PRINT "AMOUNT SO FAR ";AS
7Ø4Ø PRINT "AMOUNT LEFT   ";AL
7Ø5Ø PRINT "FIELD LEFT    ";FL
7Ø6Ø PRINT :PRINT
7Ø7Ø PRINT "FIELD SIZE %  ";:INPUT FS
7Ø8Ø IF FS>AL OR FS=Ø THEN FS=AL
7Ø9Ø AL=AL-FS:AS=AS+FS
71ØØ PRINT "ENTER FIELD NAME ";:INPUT N$
711Ø IF LEN(N$)>9 THEN N$=N$(1,9)
712Ø IF LEN(N$)<9 THEN N$(LEN(N$)+1)=" ":GOTO 712Ø
713Ø PCT(OO)=FS
714Ø TR=(OO-1)*1Ø+1
```

```
7150 T$(TR,TR+9)=N$
7160 OO=OO+1
7170 IF OO>9 THEN PCT(OO-1)=PCT(OO-1)+AL:GOTO 7200
7180 if AL=0 THEN 7200
7185 FL=FL-1
7190 GOTO 7010
7200 NP=OO-1
7210 GRAPHICS 0
7220 RETURN
8000 HH=(ROW-6)*10+1
8010 PRINT #4;BL$(1,TT(ROW+4));T2$(HH,HH+9)
8020 RETURN
9000 FOR I=1 TO PI
9010 IF RR(I)=X4 THEN YY=1
9020 NEXT I
9025 IF YY=0 THEN 9080
9030 IF YY=1 THEN X4=X4-1
9040 IF X4<1 THEN X4=X4+2
9050 YY=0:GOTO 9000
9080 RETURN
```

Appendix G

Commodore VIC-20 and C-64 Computers

The best way to connect Delta to your Commodore computer is with a **Universal/Commodore Parallel Interface** by Star Micronics. Or you can use many of the other available parallel interface adapters for the Commodore computers.

Setting the switches

We recommend that you set the DIP switches in Delta as shown below when connecting it to a Commodore computer.

Recommended DIP Switch Settings for Commodore VIC-20 and C-64

Switch	Setting	Function
1-1	ON	11 inch page size
1-2	ON	Normal print density
1-3	ON	10 CPI pitch
1-4	ON	Normal characters
1-5	ON	1/6 inch line feed
1-6	ON	U.S.A. Character set
1-7	ON	
1-8	ON	
2-1	ON	Paper-out detector active
2-2	OFF	Parallel interface
2-3	OFF	8-bit interface
2-4	ON	Auto line feed

Using Commodore BASIC

Commodore computers can use the full capabilities of Delta. Commodore BASIC does, however, have a few differences from other BASICs.

Commodore BASIC has no LPRINT statement. You must

open the printer as a file and then direct your print statements to that file, like this:

```
1Ø OPEN 4,4
2Ø PRINT#4, "TESTING"
```

When the program is done printing, you should clear the buffer and close the file like this:

```
9Ø PRINT#4 : CLOSE 4
```

Listing programs

To list a program on the Commodore computers you must open the printer as a file and redirect screen output to the printer before issuing the LIST command. The correct sequence looks like this:

```
OPEN 4,4
CMD 4
LIST
```

When you are done listing your program you must close the printer channel to stop sending output to the printer. To do this, type:

```
PRINT#4 : CLOSE 4
```

Program listings

The following programs have been converted to run on Commodore computers.

Chart program

```
1Ø REM  COMMODORE 64 <> DELTA 10
1ØØ OPEN4,4:CMD4
11Ø GOSUB 1ØØØ
12Ø GOSUB 2ØØØ
13Ø PRINT "*REGULAR*"
14Ø GOSUB 3ØØØ
```

```
150 PRINT "*DOUBLE STRIKE*"
160 PRINT DS$;
170 GOSUB 3000
180 PRINT "*EMPHASIZED*"
190 EM=TRUE
200 GOSUB 3000
210 PRINT "*DOUBLE STRIKE & EMPHASIZED*"
220 PRINT DS$;EM$;
230 GOSUB 3000
240 PRINT#4:CLOSE4
250 END
1000 REM
1010 IT$=CHR$(27)+CHR$(52)
1020 RO$=CHR$(27)+CHR$(53)
1030 EN$=CHR$(27)+CHR$(87)+CHR$(1)
1040 NW$=CHR$(27)+CHR$(87)+CHR$(0)
1050 PI$=CHR$(27)+CHR$(66)+CHR$(1)
1060 EL$=CHR$(27)+CHR$(66)+CHR$(2)
1070 CO$=CHR$(27)+CHR$(66)+CHR$(3)
1080 EM$=CHR$(27)+CHR$(69)
1090 NE$=CHR$(27)+CHR$(70)
1100 DS$=CHR$(27)+CHR$(71)
1110 ND$=CHR$(27)+CHR$(72)
1120 UN$=CHR$(27)+CHR$(45)+CHR$(1)
1130 NU$=CHR$(27)+CHR$(45)+CHR$(0)
1140 SP$=CHR$(27)+CHR$(83)+CHR$(0)
1150 SB$=CHR$(27)+CHR$(83)+CHR$(1)
1160 NS$=CHR$(27)+CHR$(84)
1170 RA$=NE$+NU$+ND$+RO$+PI$+NW$
1180 TRUE=1:FALSE=0
1190RETURN
2000 PRINT RA$
2010 PRINT EN$;"   NORMAL      ENLARGED   "
2020 PRINT RA$;UN$;
2030 PRINT CO$;"CONDENSED ";
2040 PRINT EL$;"  ELITE    ";
2050 PRINT PI$;"  PICA     ";
2060 PRINT CO$;"CONDENSED ";
2070 PRINT EL$;"  ELITE    ";
2080 PRINT PI$;"  PICA     ";RA$
2090 RETURN
3000 IT=FALSE:PRINT RO$;
3010 UN=FALSE:PRINT NU$;
3020 EN=FALSE:PRINT NW$;
3030 PI=FALSE:
```

```
3040 PRINT CO$;:GOSUB 3130
3050 PRINT EL$;:GOSUB 3130
3060 PRINT PI$;:PI=TRUE:GOSUB 3130
3070 IF EN=TRUE THEN PRINT:GOTO 3090
3080 PRINT EN$;:EN=TRUE:GOTO 3030
3090 IF UN=TRUE THEN PRINT:GOTO 3110
3100 PRINT UN$;:UN=TRUE:GOTO 3020
3110 IF IT=TRUE THEN PRINT RA$:RETURN
3120 PRINT IT$;:IT=TRUE:GOTO 3010
3130 BL$="      ":FD$="...."
3140 IF EM=FALSE THEN PRINT "AB"+CHR$(99)+CHR$(100);
   :GOTO 3190
3150 IF PI=FALSE THEN PRINT FD$;:GOTO 3170
3160 PRINT "AB";CHR$(99);CHR$(100);
3170 IF EN=TRUE THEN PRINT " ";:RETURN
3180 IF EN=FALSE THEN PRINT BL$;:RETURN
3190 IF EN=TRUE THEN PRINT " ";:RETURN
3200 PRINT SP$;"X";CHR$(120);
3210 PRINT SB$;"Y";CHR$(121);"  ";
3220 PRINT NS$;
3230 RETURN
```

Special character chart program

```
5 REM COMMODORE 64 <> DELTASET
10 OPEN4,4:CMD4
20 FOR J=160 TO 255 STEP 8
30 FOR I=J TO J+7
40 PRINT I;" =";CHR$(I);CHR$(9);
50 NEXT I:PRINT:NEXT J
60 PRINT#4:CLOSE4
70 END
```

Macro program

```
10 REM COMMODORE 64 <> DELTAMACRO
20 OPEN4,4:CMD4
30 PRINT CHR$(27);"+";
40 PRINT CHR$(18);
50 PRINT CHR$(27);"W"CHR$(0);
60 PRINT CHR$(27);"F";
70 PRINT CHR$(27);"H";
80 PRINT CHR$(27);"-";CHR$(0);
90 PRINT CHR$(27);"T";
95 PRINT CHR$(27);"5";
```

```
98 PRINT CHR$(30)
99 PRINT#4:CLOSE4:END
```

Bridge hand program

```
10 REM COMMODORE 64 <> DELTA BRIDGE
15 OPEN4,4:CMD4
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
60 PRINT#4:CLOSE4
70 END
1000 REM  INITIALIZE VARIABLES
1010 DIM HA(4),DE(52),CA$(13),SU$(3)
1020 CA$(1)=" 2":CA$(2)=" 3":CA$(3)=" 4"
1030 CA$(4)=" 5":CA$(5)=" 6":CA$(6)=" 7"
1040 CA$(7)=" 8":CA$(8)=" 9":CA$(9)=" 10"
1050 CA$(10)=" J":CA$(11)=" Q":CA$(12)=" K":
   CA$(13)=" A"
1060 SU$(0)="S":SU$(1)="H":SU$(2)="D":SU$(3)="C"
1070 RETURN
2000 REM INITIALIZE PRINTER
2010 PRINT CHR$(27);CHR$(68);CHR$(20);CHR$(40);
   CHR$(0)
2020 PRINT CHR$(27)CHR$(43);CHR$(27)CHR$(36)CHR$(0);
   CHR$(27)CHR$(69)CHR$(30)
2030 PRINT CHR$(27);CHR$(42);CHR$(0)
2040 FOR I=1 TO 4
2050 PRINT CHR$(27);CHR$(42);CHR$(1);
2060 FOR J=1 TO 13
2070 READ X:PRINT CHR$(X);
2080 NEXT J
2090 NEXT I
2100 PRINT
2110 RETURN
2120 DATA 72,11,4,10,20,10,52,72,52,10,20,10,4
2130 DATA 83,11,16,8,20,8,86,41,86,8,20,8,16
2140 DATA 67,11,8,16,8,18,65,62,65,18,8,16,8
2150 DATA 68,11,8,0,28,0,62,65,62,0,28,0,8
3000 REM DEAL CARD
3010 FOR CA=1 TO 52
3020 X=INT(RND(1)*4+1)
3030 IF HA(X)=13 THEN 3020
3035 HA(X)=HA(X)+1
3040 DE(CA)=X
```

```
3Ø5Ø NEXT CA
3Ø6Ø RETURN
4ØØØ REM PRINT FOUR HANDS
4Ø1Ø PRINT CHR$(27);"!";CHR$(9);"NORTH"
4Ø2Ø PRINT CHR$(27);"$";CHR$(1);CHR$(27);CHR$(7Ø);
4Ø3Ø HA=1
4Ø4Ø FOR SU=Ø TO 3
4Ø5Ø PRINT CHR$(9);
4Ø6Ø GOSUB 43ØØ
4Ø7Ø PRINT
4Ø8Ø NEXT SU
4Ø9Ø PRINT CHR$(27);"!";"WEST";CHR$(9);CHR$(9);
   "EAST"
41ØØ PRINT CHR$(27);"$";CHR$(1);CHR$(27);CHR$(7Ø);
411Ø FOR SU=Ø TO 3
412Ø HA=2
413Ø GOSUB 43ØØ
414Ø PRINT CHR$(9)CHR$(9);
415Ø HA=3
416Ø GOSUB 43ØØ
417Ø PRINT
418Ø NEXT SU
419Ø PRINT CHR$(27);"!";CHR$(9);"SOUTH"
42ØØ PRINT CHR$(27);"$";CHR$(1);CHR$(27);CHR$(7Ø);
421Ø HA=4
422Ø FOR SU=Ø TO 3
423Ø PRINT CHR$(9);
424Ø GOSUB 43ØØ
425Ø PRINT
426Ø NEXT SU
427Ø PRINT CHR$(27);"$"CHR$(Ø);CHR$(27);CHR$(7Ø)
428Ø RETURN
429Ø REM PRINT ONE LINE
43ØØ PRINT SU$(SU);
431Ø FOR CA=13 TO 1 STEP -1
432Ø IF DE(SU*13+CA)=HA THEN PRINT CA$(CA);
433Ø NEXT CA
434Ø RETURN
```

Numeral program

```
5 REM COMMODORE 64 〈〉 DELTANUMERAL
1Ø REM PROGRAM TO DEFINE AND PRINT NUMERALS
2Ø REM EACH NUMERAL IS MADE UP OF 4 CHARACTERS (2
   WIDE * 2 HIGH )
25 OPEN4,4:CMD4
```

```
30 DD$=CHR$(27) + CHR$(42) + CHR$(1)
40 DP$ = CHR$(27) + CHR$(88) + CHR$(1)
50 NDP$ = CHR$(27) + CHR$(88) + CHR$(0)
60 L7$ = CHR$(27) + CHR$(49) : L12$= CHR$(27) +
   CHR$(50)
70 FOR N1= 160 TO 200
80 PRINT DD$;
90 PRINT CHR$(N1);
100 READ N2
110 PRINT CHR$(N2);
120 FOR S = 1 TO 11
130 READ MS
140 PRINT CHR$(MS);
150 NEXT S
160 NEXT N1
170 REM
180 AS = 160
190 FOR NUM = 0 TO 9
200 NT$(NUM)=CHR$(AS + 0) + CHR$(AS + 1)
210 NB$(NUM)=CHR$(AS + 2) + CHR$(AS + 3)
220 AS = AS + 4
230 NEXT NUM
240 BK$= CHR$(200)
250 PRINT DP$;L7$
260 FOR NUM = 0 TO 9
270 PRINT NT$(NUM);BK$;
280 NEXT NUM
290 PRINT
300 FOR NUM = 0 TO 9
310 PRINT NB$(NUM); BK$;
320 NEXT NUM
330 PRINT NP$;L12$
335 PRINT#4:CLOSE4
340 REM  ZERO
350 DATA 11,0,96,16,104,16,44,30,14,0,2,1
360 DATA 11,2,1,2,1,6,8,38,88,32,88,32
370 DATA 11,3,12,19,12,51,0,96,0,96,0,96
380 DATA 11,0,32,0,48,0,28,3,12,3,4,3
390 REM  ONE
400 DATA 11,0,0,0,0,0,4,0,4,0,4,126
410 DATA 9,12,114,12,114,12,2,0,0,0,0,0
420 DATA 11,64,0,64,0,64,0,64,32,80,47,80
430 DATA 9,47,80,47,64,0,64,0,64,0,0,0
440 REM TWO
450 DATA 11,0,0,0,0,0,0,12,16,14,0,6,0
```

```
460 DATA 11,3,0,3,0,70,56,70,56,4,24,0
470 DATA  11,64,0,64,32,64,32,80,32,80,40,64
480 DATA 11,44,64,38,65,34,65,32,80,32,88,0
490 REM THREE
500 DATA 11,0,0,0,0,0,0,4,2,4,2,4
510 DATA 11,34,84,34,92,34,76,34,68,2,64,0
520 DATA 11,16,0,48,0,56,64,48,64,32,64,32
530 DATA 11,64,32,64,48,9,54,9,22,9,6,1
540 REM  FOUR
550 DATA 11,0,0,0,0,0,0,64,36,88,32,16
560 DATA 11,0,0,64,32,64,56,64,60,2,12,0
570 DATA 11,0,8,4,10,5,10,5,8,4,72,4
580 DATA 11,88,38,89,38,89,6,73,4,8,6,0
590 REM  FIVE
600 DATA 11,0,0,0,0,64,32,84,50,76,34,68
610 DATA 10,34,68,34,68,34,68,2,68,2,0,0
620 DATA 10,0,32,24,101,24,97,0,64,0,64,0
630 DATA 11,64,0,96,1,48,15,48,15,16,15,0
640 REM SIX
650 DATA 11,0,96,0,112,0,120,0,92,0,102,0
660 DATA 11,98,0,98,0,98,0,70,0,14,0,6
670 DATA 11,7,8,23,8,55,8,99,0,65,0,64
680 DATA 11,0,96,0,112,1,62,1,30,1,14,0
690 REM  SEVEN
700 DATA 11,0,16,8,6,8,6,8,6,8,6,8
710 DATA 9,70,8,102,8,54,8,6,0,2,0,0
720 DATA 11,0,64,0,96,0,120,0,124,0,30,1
730 DATA 9,6,1,0,0,0,0,0,0,0,0,0
740 REM EIGHT
750 DATA 11,0,0,0,0,24,36,24,102,24,102,0
760 DATA 11,67,0,67,0,99,28,34,28,34,28,0
770 DATA 11,12,18,44,19,108,19,96,1,64,0,64
780 DATA 11,0,96,1,112,15,48,15,16,14,0,0
790 REM NINE
800 DATA 11,0,0,120,4,120,6,120,6,0,3,0
810 DATA 11,3,0,3,0,67,4,123,4,122,4,120
820 DATA 11,48,0,56,0,113,0,99,0,99,0,99
830 DATA 11,0,115,0,57,0,31,0,15,0,7,0
840 REM SPACE
850 DATA 11,0,0,0,0,0,0,0,0,0,0,0,0
```

Download utility program

```
4 ED$=" E)DIT P)RINTER Q)UIT                    "
5 POKE 53281,0:POKE 53280,0
6 PRINT CHR$(5)
```

```
7 DD=1150
8 Y=0:X=0
10 DIM Z(8,12),MM(11),ML$(11),KK$(11,5)
15 AD=1984
20 PRINT CHR$(147):GOSUB 660
30 CS$=CHR$(0):SC$=CHR$(42):SS$=CHR$(15)
40 GET A$:IF A$=""THEN 40
50 IF A$="Q" THEN PRINT CHR$(147):END
60 IF A$="P" THEN GOSUB 680:GOTO 40
70 IF A$="E" THEN GOSUB 900:GOSUB 260:GOTO 40
80 GOTO 40
120 IF Z(G,H)=0 THEN C$=" ":GOSUB 11000
130 IF Z(G,H)=1 THEN C$=SC$:GOSUB 11000
140 RETURN
150 IF Z(G,H)=1 THEN C$=SS$:GOSUB 11000
160 IF Z(G,H)=0 THEN C$=CS$:GOSUB 11000
170 RETURN
260 REM  EDIT LEVEL
265 H=1:G=1:X=0:Y=0
266 FOR I=1 TO 11:ML$(I)="":MM(I)=0:   NEXT I
267 FOR I=1 TO 11:FOR J=1 TO 5:KK$(I,J)="":NEXT J:
   NEXT I
270 GET A$:IF A$=""THEN 270
300 IF A$="L"THEN GOSUB 390:GOTO 370
310 IF A$="R" THEN GOSUB 410:GOTO 370
320 IF A$="D" THEN GOSUB 430:GOTO 370
330 IF A$="U" THEN GOSUB 450:GOTO 370
340 IF A$="I" THEN GOSUB 470:GOTO 370
350 IF A$="C" THEN GOSUB 490:GOTO 370
360 IF A$="Q" THEN GOSUB 500:GOTO 380
370 GOTO 270
380 RETURN
390 GOSUB 120:Y=Y-3:H=H-1:IF Y<0 then Y=0:H=1
400 GOSUB 150:RETURN
410 GOSUB 120:Y=Y+3:H=H+1:IF Y>30 THEN Y=30:H=11
420 GOSUB 150:RETURN
430 GOSUB 120:X=X+8:G=G+1:IF X>480  THEN X=480 :G=7
440 GOSUB 150:RETURN
450 GOSUB 120:x=x-8:G=G-1:IF X<0 THEN X=0:G=1
460 GOSUB 150:RETURN
470 IF Z(G,H-1)=1 OR Z(G,H+1)=1 THEN RETURN
480 Z(G,H)=1:C$=SS$:GOSUB 11000:RETURN
490 Z(G,H)=0:C$=CS$:GOSUB 11000:RETURN
500 REM  GET OF EDIT MODE
510 C$=ED:GOSUB 1000
```

```
52Ø IF Z(G,H)=1 THEN C$=SC$:GOSUB 11ØØØ:GOTO 54Ø
53Ø IF Z(G,H)=Ø THEN C$=" ":GOSUB 11ØØØ
54Ø REM PRINT COLUMN VALUES
55Ø FOR I=1 TO 11:FOR J=1 TO 7
56Ø MM(I)=MM(I)+Z(J,I)*2^(J-1):NEXT J:NEXT I
57Ø J=Ø:PRINT CHR$(19):FOR I=1 TO 8 :PRINT CHR$(17):
   NEXT I
574 PRINT "    ";
575 FOR I=1 TO 11:ML$(I)=STR$(MM(I)):NEXT I
58Ø FOR I=1 TO 11:FOR J=1 TO LEN(ML$(I)):
   KK$(I,J)=MID$(ML$(I),J,1):NEXT J
585 NEXT I
59Ø FOR I=1 TO 11:D1=17Ø7:FOR J=1 TO LEN(ML$(I))
592 POKE D1+I*3, ASC(KK$(I,J)):D1=D1+4Ø:NEXT J
594 NEXT I
6ØØ GOSUB 66Ø:RETURN
66Ø B$=ED$:GOSUB 1ØØØ
67Ø FOR I=1 TO 7:FOR J=1 TO 11:Z(I,J)=Ø:NEXT J:NEXT
   I:RETURN
68Ø REM PRINT MODE
69Ø PRINT CHR$(19):FOR I=1 TO 22:PRINT CHR$(17);:
   NEXT I
695 INPUT "NORMAL OR PROPORTIONAL (N/P) ";AN$
7ØØ IF AN$="N" THEN PR=Ø:GOTO 75Ø
71Ø IF AN$="P" THEN GOTO 73Ø
72Ø PRINT CHR$(145);:GOTO 695
73Ø GOSUB 12ØØØ:PRINT CHR$(145);
732 INPUT "PROPORTIONAL DATA (4-11) ";PR
74Ø IF PR<4 OR PR>11 THEN 73Ø
75Ø GOSUB 12ØØØ:PRINT CHR$(145);
755 INPUT "SHIFTED ENTER 1 ELSE ENTER Ø  ";SH
76Ø IF SH<Ø OR SH>1 THEN 75Ø
77Ø GOSUB 12ØØØ:PRINT CHR$(145);
775 INPUT "ASCII CODE  (33-126)  (16Ø-254) ";AS
777 IF (AS<33 OR  AS>126) AND (AS<16Ø OR AS>254)
   THEN 77Ø
779 GOSUB 12ØØØ
78Ø IF SH=1 THEN SH=16
79Ø FOR I=1 TO 11:MM$=MM$+CHR$(MM(I)):NEXT I
8ØØ N1=AS:N2=PR+SH
81Ø OPEN4,4:CMD4
82Ø PRINT CHR$(27);"*";CHR$(1);CHR$(N1);CHR$(N2);MM$
83Ø IF AN$="N" THEN PRINT CHR$(27);"$";CHR$(1):GOTO
   85Ø
84Ø PRINT CHR$(27);"X";CHR$(1)
```

```
85Ø FOR I=1 TO 2Ø:PRINT CHR$(N1);" ";:NEXT I:PRINT
86Ø PRINT CHR$(14);:FOR I=1 TO 1Ø:PRINT CHR$(N1);"
   ";:NEXT I:PRINT
87Ø PRINT CHR$(15);:FOR I=1 TO 2Ø:PRINT CHR$(N1);:
   NEXT I:PRINT
88Ø IF AN$="N" THEN PRINT CHR$(27);"$";CHR$(Ø);:GOTO
   89Ø
885 PRINT CHR$(27);"X";CHR$(Ø)
89Ø PRINT CHR$(27);"@":MM$="":PRINT#4:CLOSE4:RETURN
9ØØ PRINT CHR$(147)
92Ø PRINT "     M1 M2 M3 M4 M5 M6 M7 M8 M9 M1Ø M11"
93Ø FOR I=Ø TO 7 :PRINT "     ";:FOR J=1 TO 11
94Ø PRINT "!--";:NEXT J:PRINT "!":IF I<7 THEN
   PRINT  2^I
95Ø NEXT I
96Ø B$="R)GT L)FT U)P D)WN I)NSRT C)LEAR Q)UIT"
97Ø GOSUB 1ØØØ
975 X=Ø:Y=Ø:C$=CS$:GOSUB 11ØØØ
98Ø RETURN
1ØØØ FOR I=1 TO LEN(B$)
1Ø1Ø Z=ASC(MID$(B$,I,1))
1Ø2Ø IF Z>64 AND Z<91 THEN Z=Z-64
1Ø3Ø POKE AD+I,Z
1Ø4Ø NEXT I
1Ø5Ø RETURN
11ØØØ POKE DD+X+Y,ASC(C$):RETURN
12ØØØ FOR I=1 TO 4Ø:POKE I+1943,32:NEXT I:RETURN
```

Delta Plot program

```
1Ø REM COMMODORE 64 <> DELTAPLT
5Ø PRINT CHR$(147)
6Ø FOR I=1 TO 5:PRINT " ":NEXT I
7Ø PRINT "THIS  PROGRAM  TAKES ABOUT ONE MINUTE "
8Ø PRINT "TO RUN, SO PLEASE TURN ON YOUR PRINTER"
9Ø PRINT "AND STAND BY........................."
1ØØ REM DELTA-PLOT
11Ø DIM BIT%(76,14)
1ØØØ REM SET PROGRAM CONSTANTS
1Ø1Ø MASK%(1) = 64    : MASK%(4) = 8
1Ø2Ø MASK%(2) = 32    : MASK%(5) = 4
1Ø3Ø MASK%(3) = 16    : MASK%(6) = 2
1Ø4Ø LX = 2Ø          : LY = 2Ø
1Ø5Ø XFAC = 72/LX     : YFAC = 87/LY
2ØØØ REM PLOT CURVE
2Ø1Ø RAD = 9
```

```
2020 X1 = 19          : Y1 = 10
2030 FOR ANG = 0 TO 360 STEP 10
2040 R1 = ANG*6.28/360
2050 X2 = RAD*COS(R1)+10 : Y2 = RAD*SIN(R1)+10
2060 GOSUB 4000
2070 NEXT ANG
3000 REM SEND BIT IMAGE MAP TO PRINTER
3005 OPEN 4,4,5
3010 PRINT#4, CHR$(27) "A" CHR$(6)
3020 FOR ROW = 0 TO 14
3022 A$=""
3025 PRINT #4,CHR$(27);"K",CHR$(75 );CHR$(0);
3050 FOR COL = 1 TO 75
3060 A$=A$+ CHR$(BIT% (COL, ROW))
3070 NEXT COL
3080 PRINT#4,A$
3090 NEXT ROW
3095 PRINT#4,CHR$(27);"2"
3100 PRINT#4 : CLOSE4
3110 END
4000 REM DRAW A LINE FROM X1,Y1, TO X2,Y2
4010 XL = X2 - X1       : YL = Y2 = Y1
4020 NX = ABS(XL*XFAC) : NY = ABS(YL*YFAC)
4030 IF NX < NY THEN NX = NY
4040 NS% = INT(NX+1)
4050 DX = XL/NS%        : DY = YL/NS%
4060 FOR I=1 TO NS%
4070 X1 = X1 + DX       : Y1 = Y1 + DY
4080 GOSUB 5000
4090 NEXT I
4100 RETURN
5000 REM PLOT A POINT AT X1,Y1
5010 XX = X1 * XFAC     : YY = Y1 * YFAC
5020 COL% = INT(XX)+1
5030 ROW% = INT(YY/6)
5040 XIT% = INT(YY - ROW% * 6) + 1
5050 BIT%(COL%,ROW%) = BIT%(COL%,ROW%) OR
   MASK%(XIT%)
5060 RETURN
```

Pie chart program

```
50 PRINT CHR$(147);"PLEASE STAND BY"
100 REM PIECHART
```

```
11Ø DIM BIT%(19Ø,36), A$(36), PCT%(25), TXT$(42),
   PTXT$(25)
12Ø ES$=CHR$(27):LF$=CHR$(1Ø)
13Ø FF$=CHR$(12):VT$=CHR$(11)
14Ø EM$=ES$ + "E":CE$=ES$ + "F"
15Ø FOR I = 1 TO 168:SP$=SP$ + CHR$(Ø): NEXT I
1ØØØ REM SET PROGRAM CONSTANTS
1Ø1Ø MASK%(1)=64:MASK%(4)=8
1Ø2Ø MASK%(2)=32:MASK%(5)=4
1Ø3Ø MASK%(3)=16:MASK%(6)=2
1Ø4Ø LX = 2Ø: LY = 2Ø
1Ø5Ø XFAC=19Ø/LX:YFAC=216/LY
1Ø6Ø FOR I= 1 TO 42
1Ø7Ø FOR J= 1 TO 8Ø:TXT$(I)=TXT$(I) + " "
1Ø8Ø NEXT J:NEXT I
1Ø9Ø GOSUB 7ØØØ
1Ø92 PRINT CHR$(147): PRINT " ":PRINT " "
1Ø94 PRINT " ":PRINT " "
1Ø96 PRINT "THIS PROGRAM TAKES ABOUT"
1Ø97 PRINT "4 MINUTES TO RUN. PLEASE"
1Ø98 PRINT "TURN ON YOUR PRINTER AND"
1Ø99 PRINT "STAND BY....."
2ØØØ REM PLOT CURVE
2Ø1Ø RAD=9
2Ø2Ø X1=19:Y1=1Ø
2Ø25 PRINT " ";
2Ø3Ø FOR ANG=Ø TO 36Ø STEP 5
2Ø4Ø R1=ANG*6.28/36Ø
2Ø5Ø X2=RAD*COS(R1) + 1Ø:Y2=RAD*SIN(R1) + 1Ø
2Ø6Ø GOSUB 4ØØØ
2Ø7Ø NEXT ANG
2Ø8Ø FOR PI= 1 TO NP%
2Ø9Ø X1=1Ø:Y1=1Ø
21ØØ TP%=TP%+PCT%(PI)
211Ø ANG=36Ø*TP%*.Ø1
212Ø R1=ANG*6.28/36Ø
213Ø X2=RAD*COS(R1)+1Ø:Y2=RAD*SIN(R1) + 1Ø
214Ø GOSUB 4ØØØ
215Ø GOSUB 6ØØØ
216Ø NEXT PI
3ØØØ REM SEND BIT IMAGE MAP TO PRINTER
3Ø2Ø FOR ROW= Ø TO 35
3Ø22 A$(ROW)= ""
3Ø5Ø FOR COL= 1 TO 19Ø
3Ø6Ø A$(ROW)=A$(ROW)+CHR$(BIT%(COL,ROW))
```

```
3070 NEXT COL
3080 NEXT ROW
3100 OPEN4,4:CMD4
3110 X=(40-LEN(T$)/2)
3120 FOR I= 1 TO X: PRINT " ";:NEXT I
3130 PRINT EM$;T$ ;CE$;LF$
3140 PRINT VT$;VT$;VT$
3150 PRINT ES$;"A";CHR$(3)
3160 PRINT TXT$(1);LF$;TXT$(2);LF$;TXT$(3);LF$
3170 FOR ROW= 0 TO 35
3180 PRINT ES$;"K";CHR$(102);CHR$(1);SP$;A$(ROW)
3200 PRINT TXT$(ROW + 4)
3210 NEXT ROW
3220 PRINT TXT$(40);LF$
3230 PRINT TXT$(41);LF$
3240 PRINT TXT$(42);LF$
3250 PRINT ES$;"2";FF$
3254 PRINT#4:CLOSE4
3255 PRINT CHR$(147)
3260 END
4000 REM  DRAW A LINE FROM X1,Y1 TO X2,Y2
4010 XL=X2-X1:YL=Y2-Y1
4020 NX=ABS(XL*XFAC):NY=ABS(YL*YFAC)
4030 IF NX<NY THEN NX=NY
4040 NS%=INT(NX + 1)
4050 DX=XL/NS%:DY=YL/NS%
4060 FOR I= 1 TO NS%
4070 X1=X1 + DX:Y1=Y1+DY
4080 GOSUB 5000
4090 NEXT I
4100 RETURN
5000 REM  PLOT A POINT AT X1,Y1
5010 XX=X1*XFAC:YY=Y1*YFAC
5020 COL=INT(XX) + 1
5030 ROW=INT(YY/6)
5040 XIT%=INT(YY-(6*ROW)) + 1
5050 BIT%(COL,ROW)=BIT%(COL,ROW) OR MASK%(XIT%)
5060 RETURN
6000 REM
6010 MA%=(ANG + PA%)/2
6020 R1=MA%*6.28/360
6030 X3=INT(20*SIN(R1)):Y3=INT(22*COS(R1))
6040 X4=22 + X3: Y4= 40 + Y3
6050 IF MA%>270 OR MA%<90 THEN GOSUB 6100: GOTO 6070
6060 GOSUB 6200
```

```
6070 PA%=ANG
6080 RETURN
6100 MM$=TXT$(X4)
6102 LL$=LEFT$(MM$,Y4)
6104 PP=LEN(PTXT$(PI))
6106 RR$=RIGHT$(MM$,80-(Y4+PP))
6108 TXT$(X4)=LL$ + PTXT$(PI) +RR$
6110 RETURN
6200 MM$=TXT$(X4)
6202 PP=LEN (PTXT$(PI))
6204 LL$=LEFT$(MM$,(Y4 - PP))
6206 RR$=RIGHT$(MM$, (80 - Y4))
6208 TXT$(X4) = LL$ + PTXT$(PI) +RR$
6210 RETURN
7000 REM
7005 I=1
7010 PRINT CHR$(147): PRINT:PRINT:PRINT:PRINT
7020 INPUT "ENTER TITLE FOR CHART  ";T$
7030 AS%=0:AL%=100
7040 PRINT CHR$(147)
7050 PRINT "TOTAL SO FAR     :  ";AS%
7060 PRINT "TOTAL REMAINING  :  ";AL%
7070 INPUT "ENTER % FOR FIELD  ";PCT%(I)
7080 IF PCT%(I)>AL% OR PCT%(I)=0 THEN PCT%(I)=AL%
7090 AL%=AL%-PCT%(I)
7100 AS%=AS%+PCT%(I)
7110 INPUT "ENTER DESCRIPTION OF FIELD :  ";PTXT$(I)
7120 IF LEN(PTXT$(I))> 15 THEN PRINT "FIELD TOO LONG
   - 15 CHAR. MAX": GOTO 7110
7130 IF AL%=0 THEN GOTO 7200
7140 I=I+1
7150 GOTO 7040
7200 NP%=I
7210 IF NP%=1 THEN 7040
7220 PRINT CHR$(147)
7230 RETURN
```

Appendix H

DIP Switch Settings

The DIP (dual in-line package) switches control some of the functions of Delta. A DIP switch actually contains several individual switches. Delta has two DIP switches with 8 individual switches in them and one DIP switch with 4 individual switches. Figure H-1 is a drawing of a typical DIP switch.

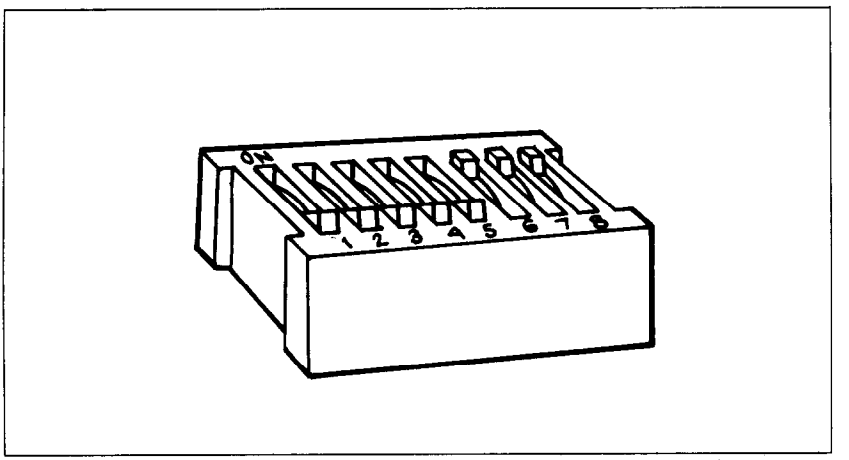


Figure H-1. *A DIP switch is actually a series of several small switches.*

DIP switch 2 is accessible from the rear of the printer, but to get to DIP switch 1 and DIP switch 3 you must remove the upper case. Chapter 10 tells you how to remove it.

Never change the settings of any of the DIP switches when the power is on. Turn off both the printer and your computer.

Table H-1 summarizes the functions of DIP switches 1 and 2. DIP switch 3 controls the serial interface and is covered in Appendix P. The individual switches on DIP switch 1 are called 1-1 through 1-8; those on switch 2 are 2-1 through 2-4.

Table H-1
DIP Switch Settings

Switch	ON	OFF
Switch 1		
1-1	11" page length	12" page length
1-2	Normal print	Emphasized print
1-3	10 CPI (pica pitch)	17 CPI (condensed pitch)
1-4	Normal	Italic
1-5	1/6" line feed	1/8" line feed
1-6	International character set selection	
1-7		
1-8		
Switch 2		
2-1	Paper-out detector on	Ignore paper-out
2-2	Serial interface	Parallel interface
2-3	7-bit interface	8-bit interface
2-4	Auto LF with CR	LF must be from host

DIP switch 1 controls the default settings for printing functions. It is located inside the case at the left rear. Figure H-2 shows the location of this switch. You must open the case to change the settings of this switch.

DIP switch 2 controls the interface. It can be reached from the back of the printer without opening the case. Figure H-3 shows the location of switch 2.

Switch Functions

Switch **Function**

1-1 Switch 1-1 sets the default page length for Delta. If switch 1-1 is ON, the page length is set to 11". When switch 1-1 is OFF the page length is set to 12". This switch is set ON at the factory.

1-2 This switch selects either normal or emphasized print for the default. If this switch is ON then Delta will print normal type when the power is turned on. If this switch is OFF then Delta will print emphasized type when the power is turned on. This switch is set ON at the factory.

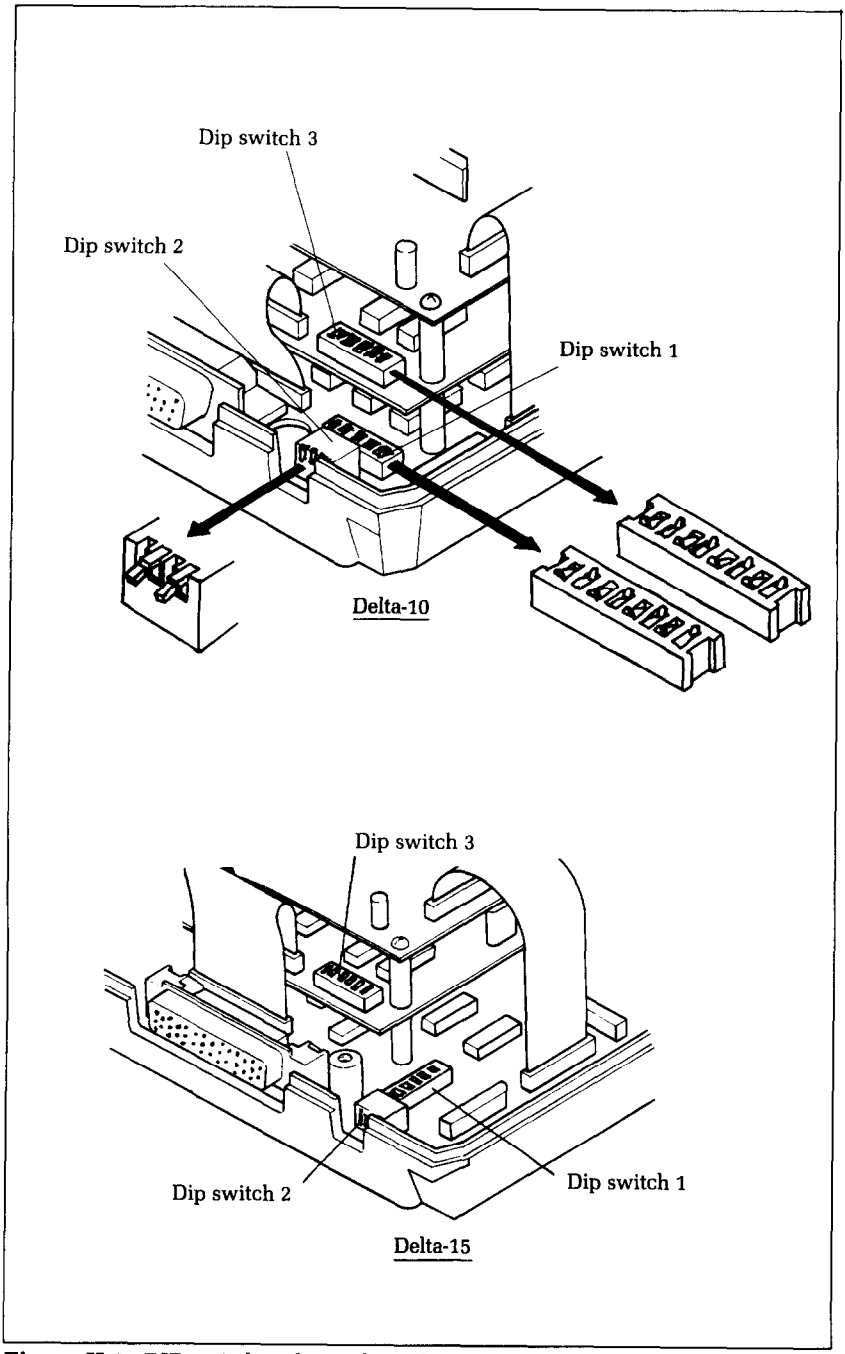


Figure H-2. *DIP switch 1, located inside Delta's case, controls default printing functions*

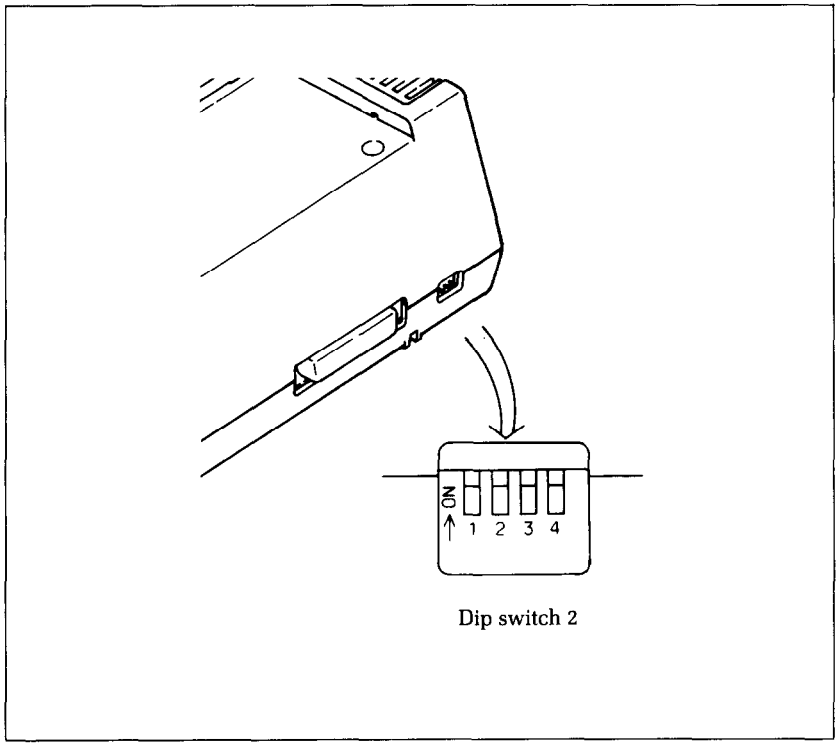


Figure H-3. *DIP switch 2, which controls the interface, is located on the back of the printer.*

1-3 This switch selects the default character pitch. If this switch is ON the default pitch is 10 CPI. If this switch is OFF the default pitch is 17 CPI. This switch is set ON at the factory.

1-4 Switch 1-4 selects the default character style. If this switch is ON then the default character style is normal characters. If this switch is OFF then the default character style is italic. This switch is set ON at the factory.

1-5 This switch sets the default line spacing. When this switch is ON the default line spacing is set to 1/6 inch. This means that Delta will advance the paper 1/6 inch each time it receives a line feed. When this switch is OFF the default line spacing is 1/8 inch. This switch is set ON at the factory.

1-6 - 1-8 These three switches determine the default international character set as shown in Table H-2. These switches are all set ON at the factory.

Table H-2
International character sets

Switch	USA	England	Germany	Denmark	France	Sweden	Italy	Spain
1-6	ON	OFF	ON	OFF	ON	OFF	ON	OFF
1-7	ON	ON	OFF	OFF	ON	ON	OFF	OFF
1-8	ON	ON	ON	ON	OFF	OFF	OFF	OFF

2-1 This switch disables the paper-out sensor. If this switch is ON the printer will signal the computer when it runs out of paper and will stop printing. If this switch is off the printer will ignore the paper-out sensor and will continue printing. This switch is set ON at the factory.

2-2 This switch selects the active interface. Turn this switch ON to use the serial interface. Turn this switch OFF to use the parallel interface. This switch is set OFF at the factory.

2-3 This switch controls the eighth bit of the parallel interface. If this switch is ON the printer will only read the first seven bits on the parallel interface and ignores the eighth bit. If this switch is OFF all eight bits will be read. This switch is set OFF at the factory.

2-4 When this switch is ON, Delta will automatically advance the paper one line every time it receives a carriage return. When this switch is OFF, the computer must send a line feed command every time the paper is to advance. (Most BASICs send a line feed with every carriage return, therefore, this switch should usually be off.) This switch is set OFF at the factory.

Appendix I

ASCII Codes

Standard and Italic Characters

Decimal	Character		Function	Decimal	Character		
0	NUL		End tab settings	47	/	/	
7	BEL		Bell	48	0	0	
8	BS		Backspace	49	1	1	
9	HT		Horizontal tab	50	2	2	
10	LF		Line feed	51	3	3	
11	VT		Vertical tab	52	4	4	
12	FF		Form feed	53	5	5	
13	CR		Carriage return	54	6	6	
14	SO		Expanded print on	55	7	7	
15	SI		Condensed print on	56	8	8	
17	DC1		On line	57	9	9	
18	DC2		Pica pitch	58	:	:	
19	DC3		Off line	59	;	;	
20	DC4		Expanded print off	60	<	<	
27	ESC		Escape	61	=	=	
30	RS		End macro	62	>	>	
32			Space	63	?	?	
33	!	!		64	@	@	*
34	"	"		65	A	A	
35	#	#	*	66	B	B	
36	$	$		67	C	C	
37	%	%		68	D	D	
38	&	&		69	E	E	
39	'	'	Apostrophe	70	F	F	
40	(	(		71	G	G	
41	)	)		72	H	H	
42	*	*		73	I	I	
43	+	+		74	J	J	
44	,	,	Comma	75	K	K	
45	-	-	Hyphen	76	L	L	
46	.	.	Period	77	M	M	

*These characters may be different if you are using an international character set other than the USA set. The characters for each set are shown on the next page.

Decimal	Character		
78	N	N	
79	O	O	
80	P	P	
81	Q	Q	
82	R	R	
83	S	S	
84	T	T	
85	U	U	
86	V	V	
87	W	W	
88	X	X	
89	Y	Y	
90	Z	Z	
91	[	[	*
92	\	\	*
93	]	]	*
94	^	^	*
95	_	_	
96	`	`	*
97	a	a	
98	b	b	
99	c	c	
100	d	d	
101	e	e	
102	f	f	

Decimal	Character		
103	g	g	
104	h	h	
105	i	i	
106	j	j	
107	k	k	
108	l	l	
109	m	m	
110	n	n	
111	o	o	
112	p	p	
113	q	q	
114	r	r	
115	s	s	
116	t	t	
117	u	u	
118	v	v	
119	w	w	
120	x	x	
121	y	y	
122	z	z	
123	{	{	*
124	|	|	*
125	}	}	*
126	~	~	*
127	DEL		Delete

*These characters may be different if you are using an international character set other than the USA set. The characters for each set are shown below.

International Character Sets

Decimal	USA	England	Germany	Denmark	France	Sweden	Italy	Spain
35	#	£	#	#	£	#	#	#
64	@	@	§	@	à	É	§	@
91	[	[	Ä	Æ	°	Ä	°	¡
92	\	\	Ö	Ø	ç	Ö	ç	Ñ
93	]	]	Ü	Å	§	Å	é	¿
94	^	^	^	^	^	Ü	^	^
96	`	`	`	`	`	é	ù	`
123	{	{	ä	æ	é	ä	à	´
124	|	|	ö	ø	ù	ö	ò	ñ
125	}	}	ü	å	è	å	è	}
126	~	~	ß	~	¨	Ü	ì	~

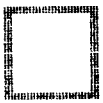
Special Characters

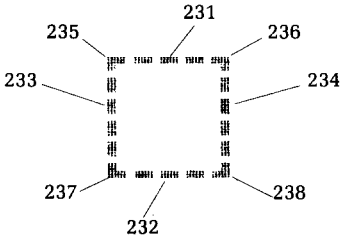
Decimal	Character	Function
135	BEL	Bell
136	BS	Backspace
137	HT	Horizontal tab
138	LF	Line feed
139	VT	Vertical tab
140	FF	Form feed
141	CR	Carriage return
142	SO	Expanded print on
143	SI	Condensed print on
145	DC1	On line
146	DC2	Pica pitch
147	DC3	Off line
148	DC4	Expanded print off
155	ESC	Escape
158	RS	End macro
160		
161		
162		
163		
164		
165		
166		
167		
168		
169		
170		
171		
172		
173		
174		
175		
176		
177		
178		
179		
180		
181		
182		
183		
184		

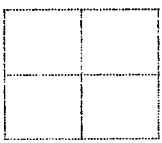
Decimal	Character
185	
186	
187	
188	
189	
190	
191	
192	
193	
194	
195	
196	
197	
198	
199	
200	
201	
202	
203	
204	
205	
206	
207	
208	
209	
210	
211	
212	
213	
214	
215	
216	
217	
218	
219	
220	
221	
222	
223	

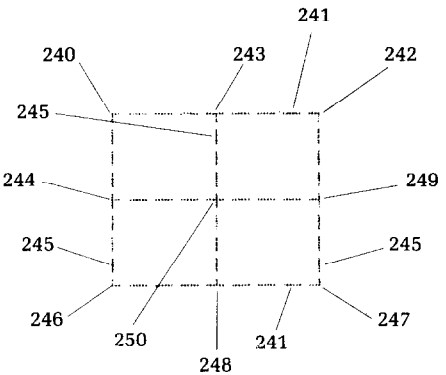
Block Graphics Characters

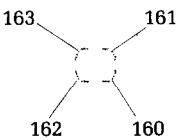
Decimal	Character	Decimal	Character
224	Space	240	
225		241	
226		242	
227		243	
228		244	
229		245	
230		246	
231		247	
232		248	
233		249	
234		250	
235		251	
236		252	
237		253	
238		254	
239			




231
235
236
233
234
237
238
232



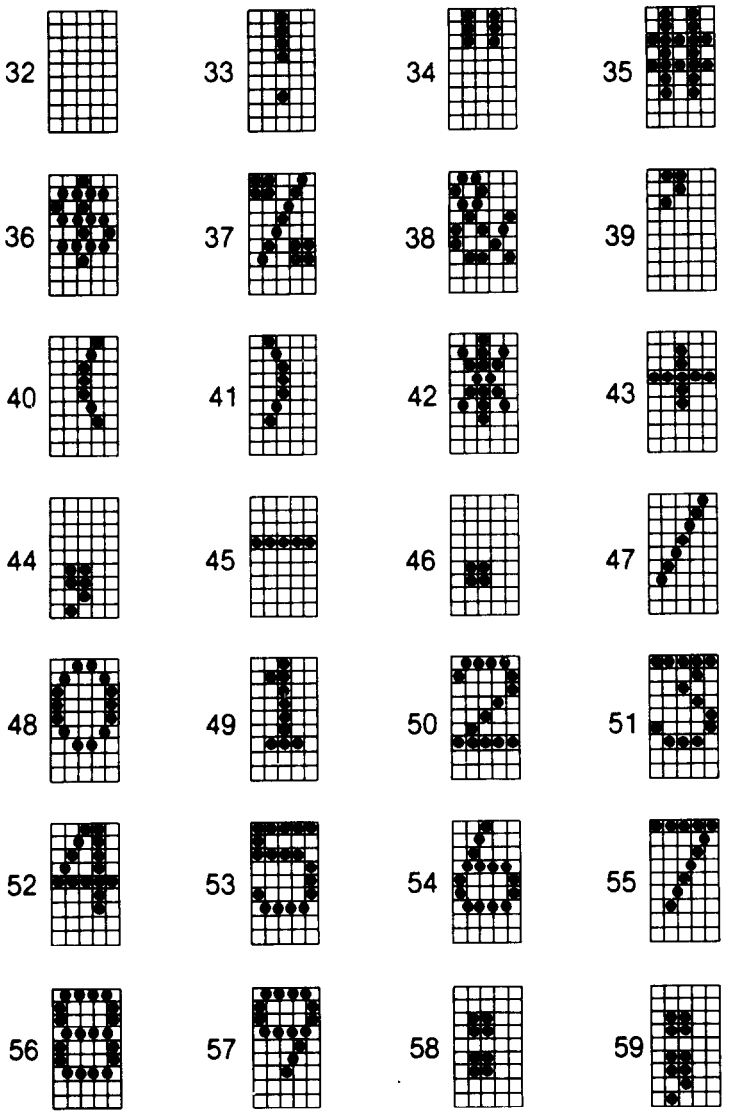

241
240
243
242
245
244
249
245
245
246
250
241
247
248

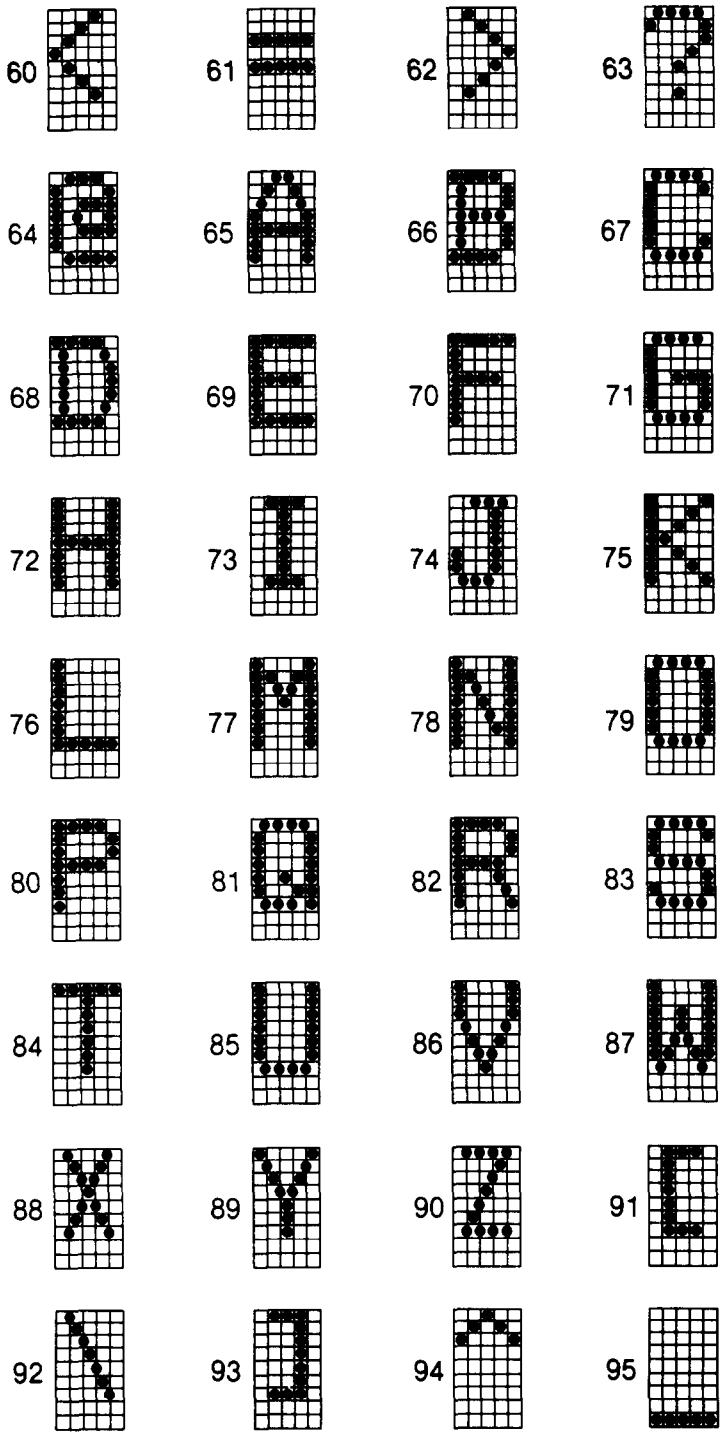

163
161
162
160

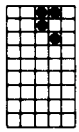
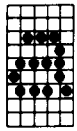
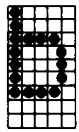
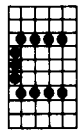
Appendix J

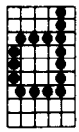
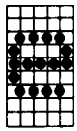
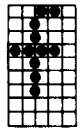
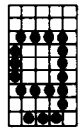
Character Style Charts

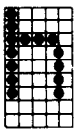
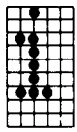
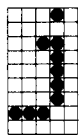
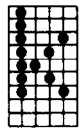
Standard Characters

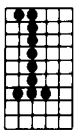
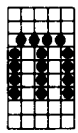
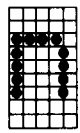
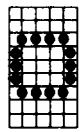


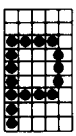
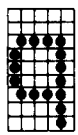
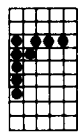
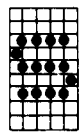

60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

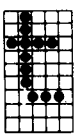
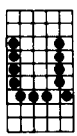
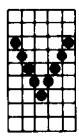
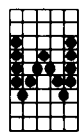
96 
97 
98 
99 

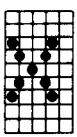
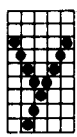
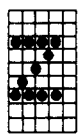
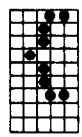
100 
101 
102 
103 

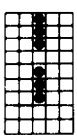
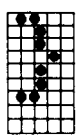
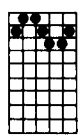
104 
105 
106 
107 

108 
109 
110 
111 

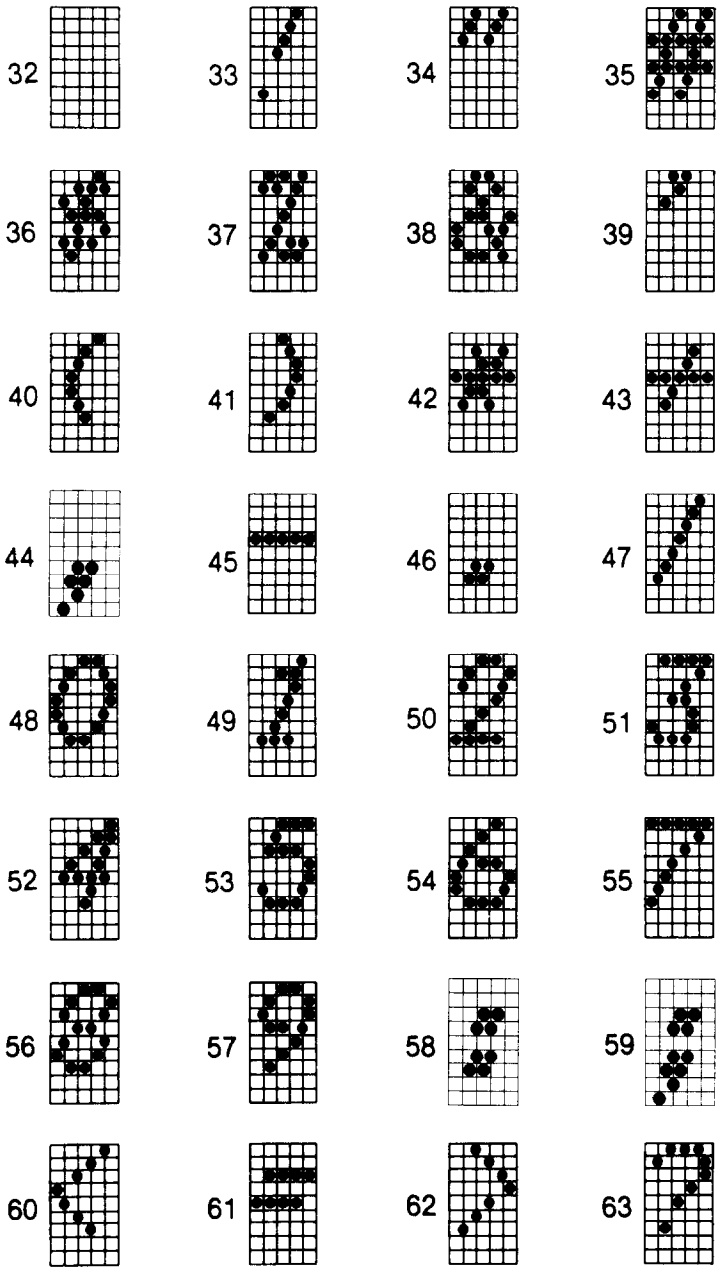
112 
113 
114 
115 

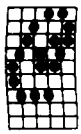
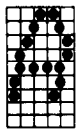
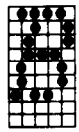
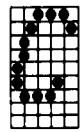
116 
117 
118 
119 

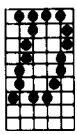
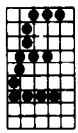
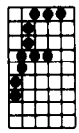
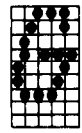
120 
121 
122 
123 

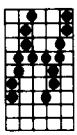
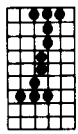
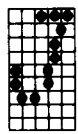
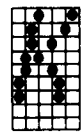
124 
125 
126 

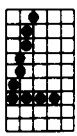
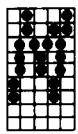
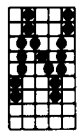
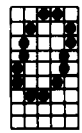
Italic Characters

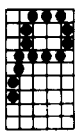
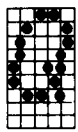
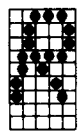
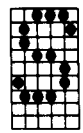


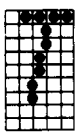
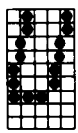
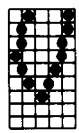
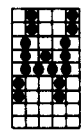
64 
65 
66 
67 

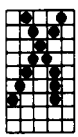
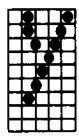
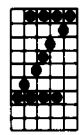
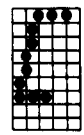
68 
69 
70 
71 

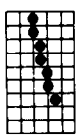
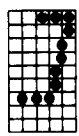
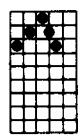
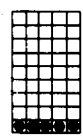
72 
73 
74 
75 

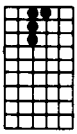
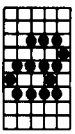
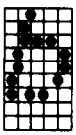
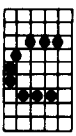
76 
77 
78 
79 

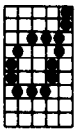
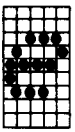
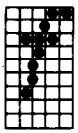
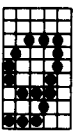
80 
81 
82 
83 

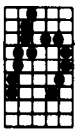
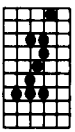
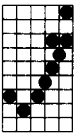
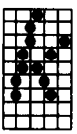
84 
85 
86 
87 

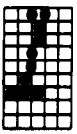
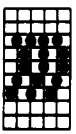
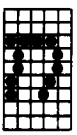
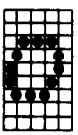
88 
89 
90 
91 

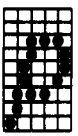
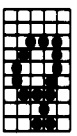
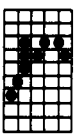
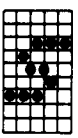
92 
93 
94 
95 

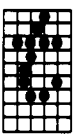
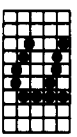
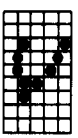
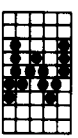
96 
97 
98 
99 

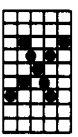
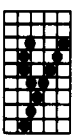
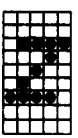
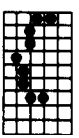
100 
101 
102 
103 

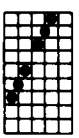
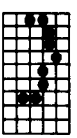
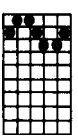
104 
105 
106 
107 

108 
109 
110 
111 

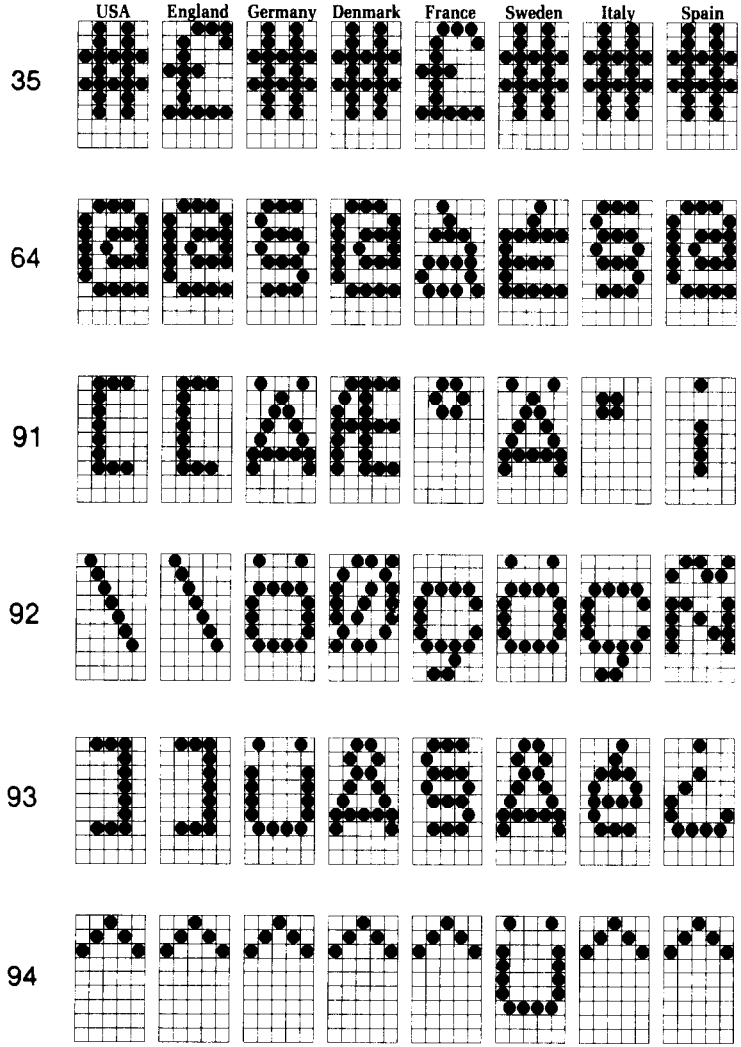
112 
113 
114 
115 

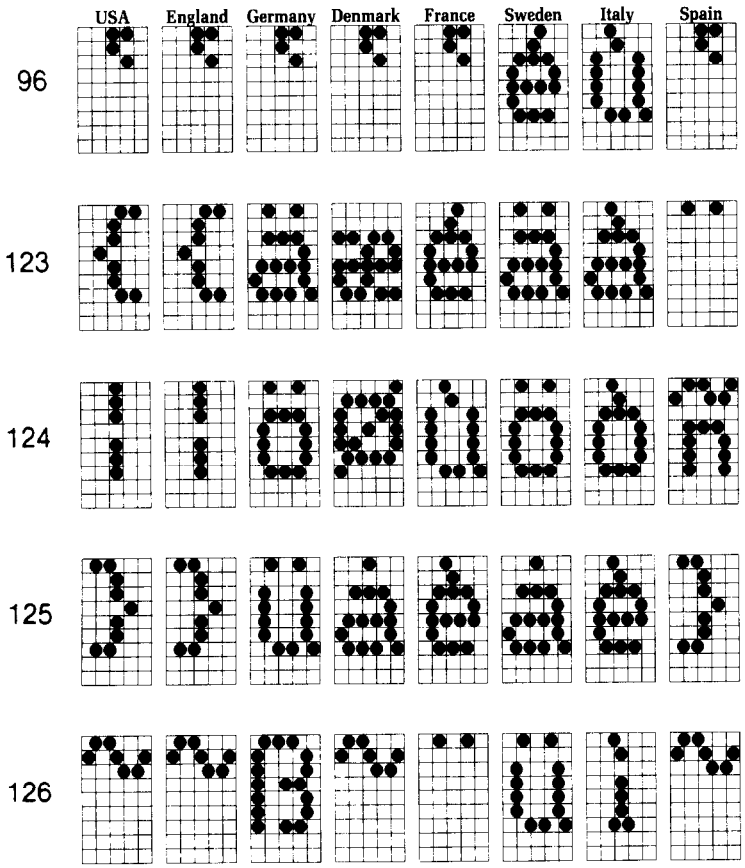
116 
117 
118 
119 

120 
121 
122 
123 

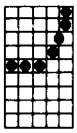
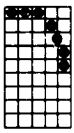
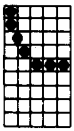
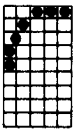
124 
125 
126 

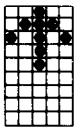
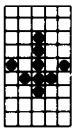
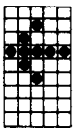
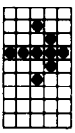
International Characters

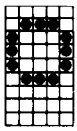
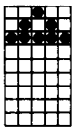
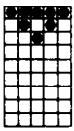
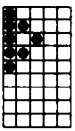


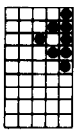
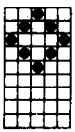
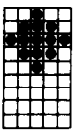
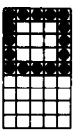

USA
England
Germany
Denmark
France
Sweden
Italy
Spain
96
123
124
125
126

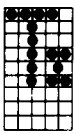
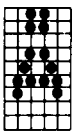
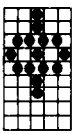
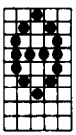
Special Characters

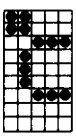
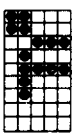
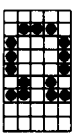
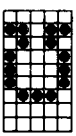
160 
161 
162 
163 

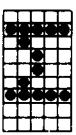
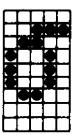
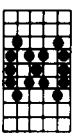
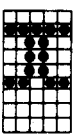
164 
165 
166 
167 

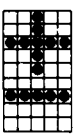
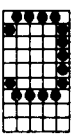
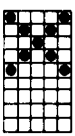
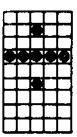
168 
169 
170 
171 

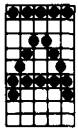
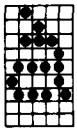
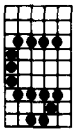
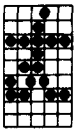
172 
173 
174 
175 

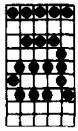
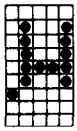
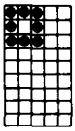
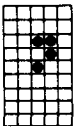
176 
177 
178 
179 

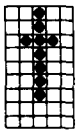
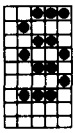
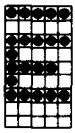
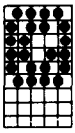
180 
181 
182 
183 

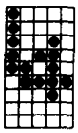
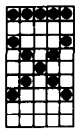
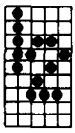
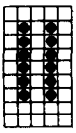
184 
185 
186 
187 

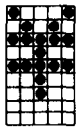
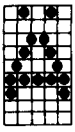
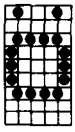
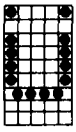
188 
189 
190 
191 

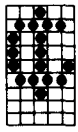
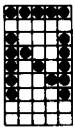
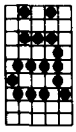
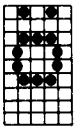
192 
193 
194 
195 

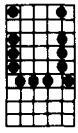
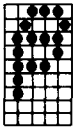
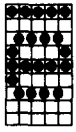
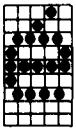
196 
197 
198 
199 

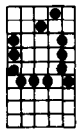
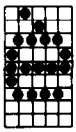
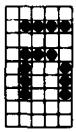
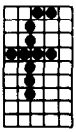
200 
201 
202 
203 

204 
205 
206 
207 

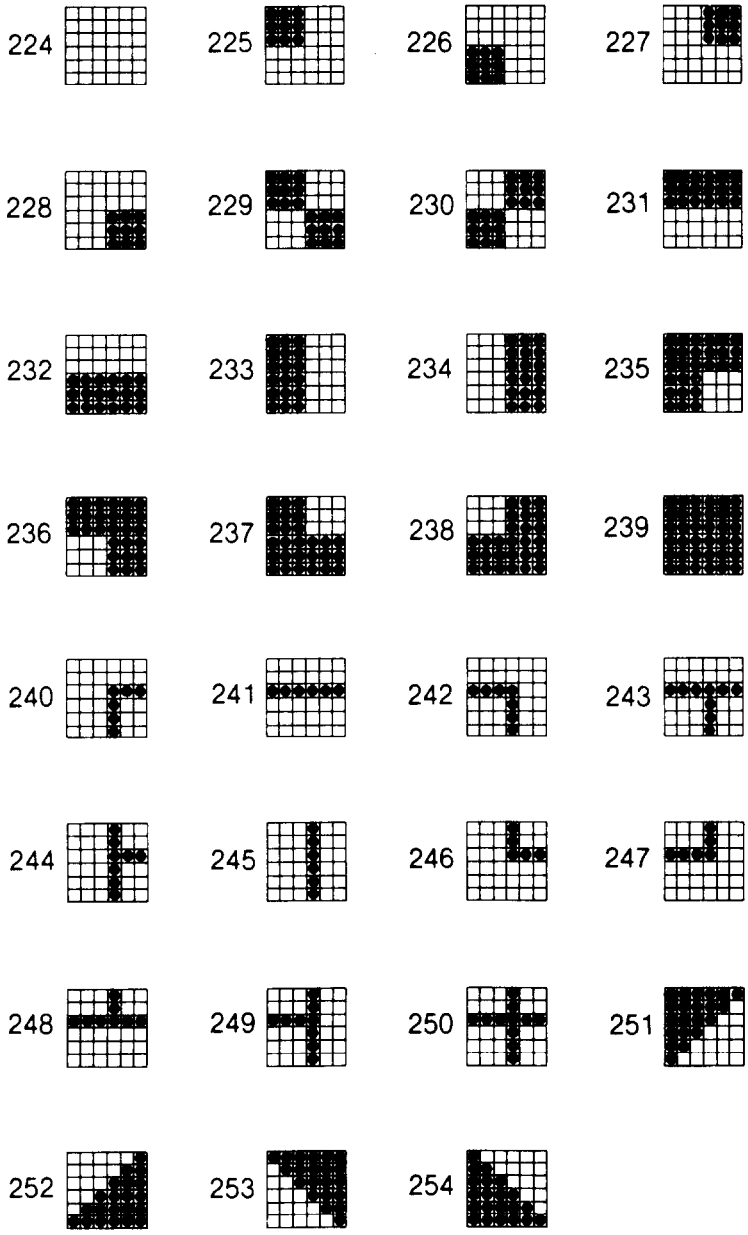
208 
209 
210 
211 

212 
213 
214 
215 

216 
217 
218 
219 

220 
221 
222 
223 

Block Graphics Characters



Appendix K

Function Code Reference

The purpose of this Appendix is to provide a quick reference for the various functions available on the Delta-10 and Delta-15. The descriptions of the codes appear in the following format:

PURPOSE:	**Tells what the function code does.**
CODE: (decimal ASCII) (hex ASCII)	Control code mnemonic ASCII decimal equivalent Hexadecimal equivalent
REMARKS:	Details how the command is used.
REFERENCE:	Tells which chapter of the manual describes the command in greater detail

There are several commands that require that you specify a value (or values) to Delta. In these cases, we have used an italic "n" or "m" to indicate a variable. You should insert the ASCII code for proper value here.

Commands to Control Print Style

These commands are used to control the font style, the print pitch, and special effects.

Font style controls

PURPOSE: **Select the standard character set.**

CODE:	⟨ESC⟩	"5"
(decimal ASCII)	27	53
(hex ASCII)	1B	35

REMARKS: This command causes the printer to cancel the italic character set and select instead the standard character set. You can select the standard character set as the power-on default by turning DIP switch 1-4 on.

REFERENCE: Chapter 3

PURPOSE: **Select the italic character set.**

CODE:	⟨ESC⟩	"4"
(decimal ASCII)	27	52
(hex ASCII)	1B	34

REMARKS: This command selects the italic character set. You can select the italic character set as the power-on default by turning DIP switch 1-4 off.

REFERENCE: Chapter 3

PURPOSE: **Select an international character set.**

CODE:	⟨ESC⟩	"7"	n
(decimal ASCII)	27	55	n
(hex ASCII)	1B	37	n

REMARKS: This command causes the printer to select an international character set determined by the value of n as shown in the table below:

n	Character set
0	U.S.A.
1	England
2	Germany
3	Denmark
4	France
5	Sweden
6	Italy
7	Spain

You can select a particular international character set as a power-on default, by adjusting the settings of DIP switches 1-6, 1-7, and 1-8.

REFERENCE: Chapter 6

Font pitch controls

PURPOSE: **Set the print pitch to pica (10 characters/ inch).**

CODE:	⟨ESC⟩	"B"	1
(decimal ASCII)	27	66	1
(hex ASCII)	1B	42	01

REMARKS: This command causes all subsequent printing to be done in pica type. This command also sets the maximum number of print columns to 80 on the Delta-10 and 136 on the Delta-15. You can select pica type as the power-on default by turning DIP switch 1-3 on.

REFERENCE: Chapter 3

PURPOSE: **Set the print pitch to elite (12 characters/ inch).**

CODE:	⟨ESC⟩	"B"	2
(decimal ASCII)	27	66	2
(hex ASCII)	1B	42	02

REMARKS: This command causes all subsequent printing to be done in elite type. This command also sets the maximum number of print columns to 96 on the Delta-10 and 163 on the Delta-15.

REFERENCE: Chapter 3

PURPOSE: **Set the print pitch to condensed (17 characters/inch).**

CODE:	⟨ESC⟩	"B"	3
(decimal ASCII)	27	66	3
(hex ASCII)	1B	42	03

REMARKS: This command causes all subsequent printing to be done in condensed type of 17 characters per inch. This command also sets the maximum number of print columns to 136 on the Delta-10 and 233 on the Delta-15. You can select condensed type as the power-on default by turning DIP switch 1-3 off.

REFERENCE: Chapter 3

PURPOSE: **Set the print pitch to pica (10 characters/ inch).**

CODE:	⟨DC2⟩
(decimal ASCII)	18
(hex ASCII)	12

REMARKS: This command is the same as ⟨ESC⟩ "B" 1, but can be used in applications where a single-character command is required.

REFERENCE: Chapter 3.

PURPOSE: **Set the print pitch to condensed (17 characters/inch).**

CODE:	⟨SI⟩
(decimal ASCII)	15
(hex ASCII)	0F

REMARKS: This command is the same as ⟨ESC⟩ "B" 3, but can be used in applications where a single-character command is required.

REFERENCE: Chapter 3

PURPOSE: **Set the print pitch to condensed (17 characters/inch).**

CODE:	⟨ESC⟩	⟨SI⟩
(decimal ASCII)	27	15
(hex ASCII)	1B	0F

REMARKS: Same as ⟨SI⟩, above.

PURPOSE: **Set the printer to expanded print.**

CODE:	⟨ESC⟩	"W"	1
(decimal ASCII)	27	87	1
(hex ASCII)	1B	57	01

REMARKS: This command causes all subsequent printing to be in expanded type. The size of the type is determined by the normal type size at the time the command is sent:

	Normal	Expanded
Pica	10 CPI	5 CPI
Elite	12 CPI	6 CPI
Condensed	17 CPI	8.5 CPI

REFERENCE: Chapter 3

PURPOSE: **Set the printer to expanded print for the remainder of the current line.**

CODE:	⟨SO⟩
(decimal ASCII)	14
(hex ASCII)	0E

REMARKS: This command causes the printer to print expanded characters until a carriage return is sent. The character widths are shown above in the description of the ⟨ESC⟩ "W" 1 command.

REFERENCE: Chapter 3

PURPOSE: **Set the printer to expanded print for the remainder of the current line.**

CODE:	⟨ESC⟩	⟨SO⟩
(decimal ASCII)	27	14
(hex ASCII)	1B	0E

REMARKS: Same as ⟨SO⟩, above.

REFERENCE: Chapter 3

PURPOSE: **Cancels expanded print.**

CODE:	⟨ESC⟩	"W"	0
(decimal ASCII)	27	87	0
(hex ASCII)	1B	57	00

REMARKS: This command resets the print size to whatever it was before being set to expanded print.

REFERENCE: Chapter 3

PURPOSE: **Cancels expanded print.**

CODE:	⟨DC4⟩
(decimal ASCII)	20
(hex ASCII)	14

REMARKS: This command is the same as ⟨ESC⟩ "W" 0, but can be used in applications where a single-character command is required.

REFERENCE: Chapter 3

Special print modes

PURPOSE: **Select double-strike printing.**

CODE:	⟨ESC⟩	"G"
(decimal ASCII)	27	71
(hex ASCII)	1B	47

REMARKS: This command causes all subsequent characters to be printed in double-strike mode. Double-strike mode causes all characters to be printed once, the paper moved up 1/144 inch, and the characters reprinted. Shifting in and out of double-strike mode on the same line can cause the line to slant slightly.

REFERENCE: Chapter 3

PURPOSE: **Cancel double-strike printing.**

CODE:	⟨ESC⟩	"H"
(decimal ASCII)	27	72
(hex ASCII)	1B	48

REMARKS: This command cancels double-strike printing and returns the printer to normal printing.

REFERENCE: Chapter 3

PURPOSE: **Select emphasized printing.**

CODE:	⟨ESC⟩	"E"
(decimal ASCII)	27	69
(hex ASCII)	1B	45

REMARKS: This command causes all subsequent characters to be printed in emphasized print. Emphasized print can only be used with pica-sized characters, or enlarged pica-sized characters (10 CPI and 5 CPI), and cannot be used with superscripts or subscripts. Emphasized print can, however, be used with double-strike mode to obtain "correspondence quality" printing. You can select emphasized printing as the power-on default by turning DIP switch 1-2 off.

REFERENCE: Chapter 3

PURPOSE: **Cancel emphasized printing.**

CODE:	⟨ESC⟩	"F"
(decimal ASCII)	27	70
(hex ASCII)	1B	46

REMARKS: This command cancels emphasized printing and returns the printer to normal printing. You can select normal printing as the power-on default by turning DIP switch 1-2 on.

REFERENCE: Chapter 3

PURPOSE: **Select underlining.**

CODE:	⟨ESC⟩	"-"	1
(decimal ASCII)	27	45	1
(hex ASCII)	1B	2D	01

REMARKS: This command causes all subsequent characters printed to be automatically underlined. Spaces are also underlined.

REFERENCE: Chapter 3

PURPOSE: **Cancel underlining.**

CODE:	⟨ESC⟩	"-"	0
(decimal ASCII)	27	45	0
(hex ASCII)	1B	2D	00

REMARKS: This command cancels underlining and returns the printer to normal printing.

REFERENCE: Chapter 3

PURPOSE: **Select superscripts.**

CODE:	⟨ESC⟩	"S"	0
(decimal ASCII)	27	83	0
(hex ASCII)	1B	53	00

REMARKS: This command causes all subsequent characters to be printed as superscripts. While in superscript mode, the normal bi-directional printing is cancelled and replaced with uni-directional printing. Printing is also set to double-strike mode. Superscripts may be used in conjunction with the italic font, and in pica, elite, and condensed pitches. It may not, however, be used in conjunction with emphasized or enlarged print.

REFERENCE: Chapter 3

PURPOSE: **Select subscripts.**

CODE:	⟨ESC⟩	"S"	1
(decimal ASCII)	27	83	1
(hex ASCII)	1B	53	01

REMARKS: This command causes all subsequent characters to be printed as subscripts. The same conditions and restrictions apply for subscripts as do for superscripts.

REFERENCE: Chapter 3

PURPOSE: **Cancel superscripts and subscripts.**

CODE:	⟨ESC⟩	"T"
(decimal ASCII)	27	84
(hex ASCII)	1B	54

REMARKS: This command cancels either superscript or subscript mode. It also cancels the uni-directional printing and double-strike which the mode had set.

REFERENCE: Chapter 3

Commands to Control Vertical Position of Print Head

These commands are used to move the paper relative to the location of the print head. By moving the paper up, the print head, in effect, moves down the page.

Line feed controls

PURPOSE: **Advance the paper one line (Line Feed).**

CODE:	⟨LF⟩
(decimal ASCII)	10
(hex ASCII)	0A

REMARKS: The actual distance advanced by the line feed is set either through the setting of DIP switch 1-5 or through various codes which can be sent (see below). When DIP switch 2-4 is "on" a line feed is automatically generated whenever the printer receives a carriage return.

REFERENCE: Chapter 4

PURPOSE: **Change the line spacing to 1/8 inch.**

CODE:	⟨ESC⟩	"0"
(decimal ASCII)	27	48
(hex ASCII)	1B	30

REMARKS: This command sets the distance the paper advances during all subsequent line feeds to 1/8 inch. You can select 1/8 inch line spacing as the power-on default by turning DIP switch 1-5 off.

REFERENCE: Chapter 4

PURPOSE: **Change the line spacing to 7/72 inch.**

CODE:	⟨ESC⟩	"1"
(decimal ASCII)	27	49
(hex ASCII)	1B	31

REMARKS: This command sets the actual distance the paper advances during all subsequent line feeds to 7/72 inch.

REFERENCE: Chapter 4

PURPOSE: **Change the line spacing to 1/6 inch.**

CODE:	⟨ESC⟩	"2"
(decimal ASCII)	27	50
(hex ASCII)	1B	32

REMARKS: This command sets the actual distance the paper advances during all subsequent line feeds to 1/6 inch. You can select 1/6 inch line spacing as the power-on default by turning DIP switch 1-5 on.

REFERENCE: Chapter 4

PURPOSE: **Change the line spacing to *n*/72 inch.**

CODE:	⟨ESC⟩	"A"	*n*
(decimal ASCII)	27	65	*n*
(hex ASCII)	1B	41	*n*

REMARKS: This command sets the distance the paper advances during all subsequent line feeds to *n*/72 inch. The value of *n* must be between 0 and 255.

REFERENCE: Chapter 4

PURPOSE: **Change the line spacing to n/144 inch.**

CODE:	⟨ESC⟩	"3"	n
(decimal ASCII)	27	51	n
(hex ASCII)	1B	33	n

REMARKS: This command sets the actual distance traveled during all subsequent line feeds to be n/144 inch. The value of n must be between 0 and 255.

REFERENCE: Chapter 4

PURPOSE: **Send a one-time line feed of n/144 inch.**

CODE:	⟨ESC⟩	"J"	n
(decimal ASCII)	27	74	n
(hex ASCII)	1B	4A	n

REMARKS: This command causes the printer to advance the paper n/144 inch. It does not change the current value of the line spacing and it does not cause a carriage return. The value of n must be between 0 and 255.

REFERENCE: Chapter 4

Form feed controls

PURPOSE: **Advance paper to top of next page (Form Feed).**

CODE:	⟨FF⟩
(decimal ASCII)	12
(hex ASCII)	0C

REMARKS: The actual length of a page ejected by a form feed is set either by the setting of DIP switch 1-1 or through various codes which can be sent (see below).

REFERENCE: Chapter 4

PURPOSE: **Set page length to *n* lines.**

CODE:	⟨ESC⟩	"C"	*n*
(decimal ASCII)	27	67	*n*
(hex ASCII)	1B	43	*n*

REMARKS: This command sets the length of all subsequent pages to *n* lines. The value of *n* must be between 1 and 127.

REFERENCE: Chapter 4

PURPOSE: **Set page length to *n* inches.**

CODE:	⟨ESC⟩	"C"	0	*n*
(decimal ASCII)	27	67	0	*n*
(hex ASCII)	1B	43	00	*n*

REMARKS: This command sets the length of all subsequent pages to *n* inches. The value of *n* must be between 1 and 32. You can select a power-on default form length of 11 inches or 12 inches by setting DIP switch 1-1.

REFERENCE: Chapter 4

PURPOSE: **Set the top margin.**

CODE:	⟨ESC⟩	"R"	*n*
(decimal ASCII)	27	82	*n*
(hex ASCII)	1B	52	*n*

REMARKS: This command sets the margin at the top of the page to n-1 lines. Printing will start on line n. The default value for *n* upon power on is 1. The value of *n* must be between 1 and 16.

REFERENCE: Chapter 4

PURPOSE: **Set the bottom margin.**

CODE:	⟨ESC⟩	"N"	n
(decimal ASCII)	27	78	n
(hex ASCII)	1B	4E	n

REMARKS: This command sets the margin at the bottom of the page to *n* lines. The printer will automatically execute a form feed when the number of lines left on a page is equal to *n*. The value of *n* must be between 1 and 127. This command is sometimes referred to as "skip-over-perforation."

REFERENCE: Chapter 4

PURPOSE: **Cancel top and bottom margins.**

CODE:	⟨ESC⟩	"O"
(decimal ASCII)	27	79
(hex ASCII)	1B	4F

REMARKS: This command cancels both the top margin set by ⟨ESC⟩ "R" *n* and the bottom margin set by ⟨ESC⟩ "N" *n*.

REFERENCE: Chapter 4

Vertical tabs

PURPOSE: **Advance paper to the next vertical tab position.**

CODE:	⟨VT⟩
(decimal ASCII)	11
(hex ASCII)	0B

REMARKS: This command causes the paper to be advanced to the next vertical tab position, or the top of the next page, whichever it finds first. The vertical tab positions are set upon power on at lines 6, 12, 18, 24, 30, 36, 42, 48, 54, and 60.

REFERENCE: Chapter 5

PURPOSE: **Set vertical tab positions.**

CODE:	⟨ESC⟩	"P"	*n1 n2 n3...*	0
(decimal ASCII)	27	80	*n1 n2 n3...*	0
(hex ASCII)	1B	50	*n1 n2 n3...*	00

REMARKS: This command cancels all current vertical tab positions and sets those defined at lines *n1*, *n2*, *n3*, etc. The maximum number of vertical tab positions allowed is 20. The ASCII 0 character is used as a command terminator. Each vertical tab position must be between 1 and 255, and they must be specified in ascending order.

REFERENCE: Chapter 5

PURPOSE: **Advance the paper *n* lines.**

CODE:	⟨ESC⟩	"a"	*n*
(decimal ASCII)	27	97	*n*
(hex ASCII)	1B	61	*n*

REMARKS: This command causes the printer to advance the paper *n* lines. It does not, however, change the current value of the vertical tab positions. The value of *n* must be between 1 and 255.

REFERENCE: Chapter 4

Commands to Control Horizontal Position of Print Head

PURPOSE: **Return print head to home position (Carriage Return).**

CODE:	⟨CR⟩
(decimal ASCII)	13
(hex ASCII)	0D

REMARKS: This command returns the print head to the home position (the left margin). If DIP switch 2-4 has been set on, then this command will also cause a line feed character to be generated after the carriage return, thereby advancing to the beginning of the next print line automatically.

REFERENCE: Chapter 4

PURPOSE: **Set the left print margin.**

CODE:	⟨ESC⟩	"M"	n
(decimal ASCII)	27	77	n
(hex ASCII)	1B	4D	n

REMARKS: This command sets the home position returned to during the execution of all subsequent carriage returns to be print position *n*. The power on default for *n* is 1. The value of *n* must be between 1 and 255. For Delta-10 the maximum print position for pica pitch is 80, for elite is 96, and for condensed pitch is 136. For Delta-15 the maximum print position for pica pitch is 136, for elite is 163, and for condensed pitch is 233.

REFERENCE: Chapter 5

PURPOSE: **Set the right print margin.**

CODE:	⟨ESC⟩	"Q"	*n*
(decimal ASCII)	27	81	*n*
(hex ASCII)	1B	51	*n*

REMARKS: This command sets the right hand print margin to print position *n*. After execution of this command, any attempt to print beyond print position *n* will cause the printer to automatically generate a carriage return and a line feed before printing the remainder of the line. The value for n must be between 1 and 255.

PURPOSE: **Move the print head to the next horizontal tab position.**

CODE:	⟨HT⟩
(decimal ASCII)	9
(hex ASCII)	09

REMARKS: This command causes the print head to advance to the next horizontal tab position. The horizontal tab positions are set at power-on to print positions 10, 20, 30, etc. (to the maximum print position).

REFERENCE: Chapter 5

PURPOSE: **Set horizontal tab positions.**

CODE:	⟨ESC⟩	"D"	*n1 n2 n3...*	0
(decimal ASCII)	27	68	*n1 n2 n3...*	0
(hex ASCII)	1B	44	*n1 n2 n3...*	00

REMARKS: This command cancels all current horizontal tab positions and sets those defined at print positions *n1*, *n2*, *n3*, etc. The maximum number of horizontal tab positions allowed is 255. The ASCII 0 character is used as a command terminator. Each horizontal tab position must be between 1 and 255, and they must be specified in ascending order.

REFERENCE: Chapter 5

PURPOSE: **Skip *n* print positions.**

CODE:	⟨ESC⟩	"b"	*n*
(decimal ASCII)	27	98	*n*
(hex ASCII)	1B	62	*n*

REMARKS: This command causes the print head to advance *n* print positions to the right. It does not, however, change the current value of the horizontal tab positions. The value of *n* must be between 1 and 255.

REFERENCE: Chapter 5

PURPOSE: **Move the print head back one print position (backspace).**

CODE:	⟨BS⟩
(decimal ASCII)	8
(hex ASCII)	08

REMARKS: This command shifts the print head one column to the left. If the print head is at the home position, the command is ignored. This command can be used to overstrike characters.

REFERENCE: Chapter 6

Download Character Commands

PURPOSE: **Define download characters into RAM.**

CODE:

⟨ESC⟩	"*"	1	*n1 n2 m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11*
27	42	1	*n1 n2 m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11*
1B	2A	01	*n1 n2 m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11*

REMARKS: This command is used to set up a user-defined character and store it into RAM for later use. RAM is cleared during power down. The value of *n1* is the position in RAM that this character is to occupy. It must be between 33 and 126 or between 160 and 254. That is, it must fall within the range of printable characters. The value of *n2* determines the attributes and width of the character. *m1* thru *m11* determine which dots form the character.

REFERENCE: Chapter 7

PURPOSE: **Copy standard character ROM fonts into RAM.**

CODE:

	⟨ESC⟩	"*"	0
(decimal ASCII)	27	42	0
(hex ASCII)	1B	2A	00

REMARKS: This command takes all of the characters in the standard ASCII character (others don't work) and copies them into RAM. This is helpful prior to defining characters in RAM because it allows standard ROM characters to be printed on the same line as download characters.

REFERENCE: Chapter 7

PURPOSE: **Select download character set with proportional spacing.**

CODE:	⟨ESC⟩	"X"	1
(decimal ASCII)	27	88	1
(hex ASCII)	1B	58	01

REMARKS: This command selects the download character set using the proportional spacing defined in the character attribute data.
NOTE: Download characters *cannot* be mixed with other characters on the same line.

REFERENCE: Chapter 7

PURPOSE: **Cancel download character set with proportional spacing.**

CODE:	⟨ESC⟩	"X"	0
(decimal ASCII)	27	88	0
(hex ASCII)	1B	58	00

REMARKS: This command cancels the download character set and selects the standard ASCII character set.

REFERENCE: Chapter 7

PURPOSE: **Select download character set with normal spacing.**

CODE:	⟨ESC⟩	"$"	1
(decimal ASCII)	27	36	1
(hex ASCII)	27	24	01

REMARKS: This command causes the printer to select the download character set using normal spacing and ignoring the proportional width data.
NOTE: Download characters *cannot* be mixed with other characters on the same line.

REFERENCE: Chapter 7

PURPOSE: **Cancel download character set with normal spacing.**

CODE:	⟨ESC⟩	"$"	0
(decimal ASCII)	27	36	0
(hex ASCII)	1B	24	00

REMARKS: This command cancels the download character set and selects the standard ASCII character set.

REFERENCE: Chapter 7

Commands to Control Graphics

PURPOSE: **Print normal-density graphics.**

CODE:	⟨ESC⟩	"K"	*n1 n2 m1 m2 m3...*
(decimal ASCII)	27	75	*n1 n2 m1 m2 m3...*
(hex ASCII)	1B	4B	*n1 n2 m1 m2 m3...*

REMARKS: This command selects 60 dots-per-inch, bit-image graphics mode. The values of *n1* and *n2* represent the number of graphics characters to be printed, where the total number of characters = *n2* times 256 + *n1*. There must be the correct number of graphic characters following *n2*. The ASCII value of these characters determine which pins are fired for each character.

REFERENCE: Chapter 8

PURPOSE: **Print double-density graphics**

CODE: ⟨ESC⟩ "L" *n1 n2 m1 m2 m3...*
(decimal ASCII) 27 76 *n1 n2 m1 m2 m3...*
(hex ASCII) 1B 4C *n1 n2 m1 m2 m3...*

REMARKS: This command selects 120 dots-per-inch, column-scan, bit-image graphics mode. The values of *n1* and *n2* are the same as in normal density graphics. There must be the correct number of graphic characters following *n2*. The ASCII value of these characters determine which pins are fired for each character.

REFERENCE: Chapter 8

PURPOSE: **Print double-density graphics with double-speed.**

CODE: ⟨ESC⟩ "y" *n1 n2 m1 m2 m3...*
(decimal ASCII) 27 121 *n1 n2 m1 m2 m3...*
(hex ASCII) 1B 79 *n1 n2 m1 m2 m3...*

REMARKS: This command selects 120 dots-per-inch, column-scan, bit-image graphics mode with double-speed. The values of *n1* and *n2* are the same as in normal density graphics. There must be the correct number of graphic characters following *n2*. The ASCII value of these characters determine which pins are fired for each character.

REFERENCE: Chapter 8

PURPOSE: **Print quadruple-density graphics.**

CODE: ⟨ESC⟩ "z" *n1 n2 m1 m2 m3...*
(decimal ASCII) 27 122 *n1 n2 m1 m2 m3...*
(hex ASCII) 1B 7A *n1 n2 m1 m2 m3...*

REMARKS: This command selects 240 dots-per-inch, column-scan, bit-image graphics mode. The values of *n1* and *n2* are the same as in normal density graphics. There must be the correct number of graphic characters following *n2*. The ASCII value of these characters determine which pins are fired for each character.

REFERENCE: Chapter 8

Macro Instruction Commands

PURPOSE: **Define macro instruction.**

CODE:	⟨ESC⟩	"+"	...	⟨RS⟩
(decimal ASCII)	27	43	...	30
(hex ASCII)	1B	2B	...	1E

REMARKS: This command cancels any existing macro instruction, and replaces it with the instruction defined. The maximum number of characters allowed in the macro instruction is 16. The ⟨RS⟩ character marks the end of the macro definition.

REFERENCE: Chapter 6

PURPOSE: **Execute macro instruction.**

CODE:	⟨ESC⟩	"!"
(decimal ASCII)	27	33
(hex ASCII)	27	21

REMARKS: This command executes a macro instruction that was previously defined.

REFERENCE: Chapter 6

Other Commands

PURPOSE: **Set the value of the eighth data bit to logical 1.**

CODE:	⟨ESC⟩	"⟩"
(decimal ASCII)	27	62
(hex ASCII)	1B	3E

REMARKS: This command forces the eighth data bit of each subsequent character sent to the printer to logical 1. This code allows users with a 7-bit interface to access those characters whose ASCII code is greater than 127. This code should not be used to transmit printer control codes.

REFERENCE: Chapter 6

PURPOSE: **Set the value of the eighth data bit to logical 0.**

CODE:	⟨ESC⟩	"="
(decimal ASCII)	27	61
(hex ASCII)	1B	3D

REMARKS: This command forces the eighth data bit of each subsequent character sent to the printer to logical 0. This code should not be used to transmit printer control codes.

REFERENCE: Chapter 6

PURPOSE: **Accept the value of the eighth data bit as is.**

CODE:	⟨ESC⟩	"#"
(decimal ASCII)	27	35
(hex ASCII)	1B	23

REMARKS: This command cancels either setting of the eighth data bit. The printer will use the value of the eighth data bit that is sent from the computer. This code allows users with only a 7-bit interface to resume normal functions after accessing those characters whose ASCII code is greater than 127.

REFERENCE: Chapter 6

PURPOSE: **Delete the last character sent.**

CODE:	⟨DEL⟩
(decimal ASCII)	127
(hex ASCII)	7F

REMARKS: This command deletes the last character received. This command is ignored if the last character received has already been printed, or if the last character received was all or part of a function code.

REFERENCE: Chapter 6

PURPOSE: **Set printer off line.**

CODE: ⟨DC3⟩
(decimal ASCII) 19
(hex ASCII) 13

REMARKS: This command causes the printer to set itself off line, disregarding all subsequent characters and function codes, with the exception of ⟨DC1⟩, which will return the printer to an on line state. This is not the same as pushing the ON-LINE button. When the ON-LINE light is out the printer will not respond to DC1.

REFERENCE: Chapter 6

PURPOSE: **Set printer on line.**

CODE: ⟨DC1⟩
(decimal ASCII) 17
(hex ASCII) 11

REMARKS: This code resets the printer to an on line state, thus allowing it receive and process all subsequent characters and function codes. This is not the same as pushing the ON-LINE button. When the ON-LINE light is out the printer will not respond to DC1.

REFERENCE: Chapter 6

PURPOSE: **Sound printer bell.**

CODE: ⟨BEL⟩
(decimal ASCII) 7
(hex ASCII) 07

REMARKS: This command causes the printer tone to sound for approximately one-fourth second.

REFERENCE: Chapter 6

PURPOSE: **Disable the printer bell.**

CODE:	⟨ESC⟩	"Y"	0
(decimal ASCII)	27	89	0
(hex ASCII)	1B	59	00

REMARKS: This command causes the printer to ignore the ⟨BEL⟩ character.

REFERENCE: Chapter 6

PURPOSE: **Enable the printer bell.**

CODE:	⟨ESC⟩	"Y"	1
(decimal ASCII)	27	89	1
(hex ASCII)	1B	59	01

REMARKS: This command causes the printer to respond to the ⟨BEL⟩ character normally by sounding the printer bell.

REFERENCE: Chapter 6

PURPOSE: **Disable paper-out detector.**

CODE:	⟨ESC⟩	"8"
(decimal ASCII)	27	56
(hex ASCII)	1B	38

REMARKS: This command causes the printer to disregard the signal sent by the paper-out detector. The paper-out signal normally sounds the printer bell and stops printing until paper is inserted and the printer is reset. This command is useful when printing on single sheets of paper because it allows printing to the bottom of the page.

REFERENCE: Chapter 6

PURPOSE: **Enable paper-out detector.**

CODE:	⟨ESC⟩	"9"
(decimal ASCII)	27	57
(hex ASCII)	1B	39

REMARKS: This command restores the function of the paper-out detector.

REFERENCE: Chapter 6

PURPOSE: **Select uni-directional printing.**

CODE:	⟨ESC⟩	"U"	1
(decimal ASCII)	27	85	1
(hex ASCII)	1B	55	01

REMARKS: This command causes all subsequent lines to be printed in uni-directional printing. Uni-directional printing is useful in printing tables or charts, since it ensures that vertical columns of characters will be in alignment.

REFERENCE: Chapter 6

PURPOSE: **Cancel uni-directional printing.**

CODE:	⟨ESC⟩	"U"	0
(decimal ASCII)	27	85	0
(hex ASCII)	1B	55	00

REMARKS: This command cancels uni-directional printing, and returns to the standard bi-directional printing, which is considerably faster.

REFERENCE: Chapter 6

PURPOSE: **Initialize printer.**

CODE:	⟨ESC⟩	"@"
(decimal ASCII)	27	64
(hex ASCII)	1B	40

REMARKS: This command reinitializes the printer. The print buffer is cleared, and the form length, character pitch, character set, line feed pitch, and international character set are all reset to the values defined by their respective DIP switches.
The main difference between the ⟨ESC⟩ "@" command and turning the printer off and back on is that download character RAM is preserved with this command.

REFERENCE: Chapter 6

Appendix L

Command Summary in Numeric Order

Control code	**Function**
CHR$(0)	End tab settings
CHR$(7)	Sound bell
CHR$(8)	Backspace
CHR$(9)	Horizontal tab
CHR$(10)	Line feed
CHR$(11)	Vertical tab
CHR$(12)	Form feed
CHR$(13)	Carriage return
CHR$(14)	Expanded print
CHR$(15)	Condensed print
CHR$(17)	On line
CHR$(18)	Pica type
CHR$(19)	Off line
CHR$(20)	Cancel enlarged print
CHR$(27)	Escape **(indicated as ⟨ESC⟩ below)**
CHR$(30)	End macro instruction definition
CHR$(127)	Delete last character
⟨ESC⟩ CHR$(14)	Expanded print
⟨ESC⟩ CHR$(15)	Condensed print
⟨ESC⟩ "!"	Use macro
⟨ESC⟩ "#"	Accept eighth bit as is
⟨ESC⟩ "$" CHR$(0)	Cancel normal download characters
⟨ESC⟩ "$" CHR$(1)	Use normal download characters
⟨ESC⟩ "*" CHR$(0)	Copy ROM characters to download RAM
⟨ESC⟩ "*" CHR$(1) *n1 n2 m1 m2 . . . m11*	Define download character
⟨ESC⟩ "+" . . . CHR$(30)	Define macro
⟨ESC⟩ "-" CHR$(0)	Stop underlining

Control code	Function
⟨ESC⟩ "-" CHR$(1)	Start underlining
⟨ESC⟩ "0"	1/8 inch line feed
⟨ESC⟩ "1"	7/72 inch line feed
⟨ESC⟩ "2"	1/6 inch line feed
⟨ESC⟩ "3" n	n/144 inch line feed
⟨ESC⟩ "4"	Italic print
⟨ESC⟩ "5"	Cancel italic print
⟨ESC⟩ "7" n	Select international character set
⟨ESC⟩ "8"	Ignore paper-out signal
⟨ESC⟩ "9"	Enable paper-out signal
⟨ESC⟩ "="	Set eighth bit to 0
⟨ESC⟩ ">"	Set eighth bit to 1
⟨ESC⟩ "@"	Reset the printer
⟨ESC⟩ "A" n	n/72 inch line feed
⟨ESC⟩ "B" CHR$(1)	Pica print
⟨ESC⟩ "B" CHR$(2)	Elite print
⟨ESC⟩ "B" CHR$(3)	Condensed print
⟨ESC⟩ "C" n	Set page length to n lines
⟨ESC⟩ "C" CHR$(0) n	Set page length to n inches
⟨ESC⟩ "D" . . . CHR$(0)	Set horizontal tabs
⟨ESC⟩ "E"	Emphasized print
⟨ESC⟩ "F"	Cancel emphasized print
⟨ESC⟩ "G"	Double-strike print
⟨ESC⟩ "H"	Cancel double-strike print
⟨ESC⟩ "J" n	Single line feed of n/144 inches
⟨ESC⟩ "K" n1 n2	Single density graphics
⟨ESC⟩ "L" n1 n2	Double density graphics
⟨ESC⟩ "M" n	Set left margin at column n
⟨ESC⟩ "N" n	Set bottom margin at n lines
⟨ESC⟩ "O"	Cancel top and bottom margins
⟨ESC⟩ "P" . . . CHR$(0)	Set vertical tabs
⟨ESC⟩ "Q" n	Set right margin at column n
⟨ESC⟩ "R" n	Set top margin at line n
⟨ESC⟩ "S" CHR$(0)	Superscript on
⟨ESC⟩ "S" CHR$(1)	Subscript on
⟨ESC⟩ "T"	Cancel super and subscripts
⟨ESC⟩ "U" CHR$(0)	Bidirectional print
⟨ESC⟩ "U" CHR$(1)	Unidirectional print
⟨ESC⟩ "W" CHR$(0)	Cancel enlarged print
⟨ESC⟩ "W" CHR$(1)	Enlarged print
⟨ESC⟩ "X" CHR$(0)	Cancel proportional download characters
⟨ESC⟩ "X" CHR$(1)	Use proportional download characters

Control code	**Function**
⟨ESC⟩ "Y" CHR$(0)	Disable bell
⟨ESC⟩ "Y" CHR$(1)	Enable bell
⟨ESC⟩ "a" *n*	Advance *n* line feeds
⟨ESC⟩ "b" *n*	Tab over *n* columns
⟨ESC⟩ "y" *n1 n2*	Double speed, double density graphics
⟨ESC⟩ "z" *n1 n2*	Quadruple density graphics

Appendix M

ASCII Code Conversion Chart

Standard ASCII Codes

Decimal	Hexadecimal	Binary	Control character	Character
0	00	0000 0000	Ctrl-@	NUL
1	01	0000 0001	Ctrl-A	
2	02	0000 0010	Ctrl-B	
3	03	0000 0011	Ctrl-C	
4	04	0000 0100	Ctrl-D	
5	05	0000 0101	Ctrl-E	
6	06	0000 0110	Ctrl-F	
7	07	0000 0111	Ctrl-G	BEL
8	08	0000 1000	Ctrl-H	BS
9	09	0000 1001	Ctrl-I	HT
10	0A	0000 1010	Ctrl-J	LF
11	0B	0000 1011	Ctrl-K	VT
12	0C	0000 1100	Ctrl-L	FF
13	0D	0000 1101	Ctrl-M	CR
14	0E	0000 1110	Ctrl-N	SO
15	0F	0000 1111	Ctrl-O	SI
16	10	0001 0000	Ctrl-P	
17	11	0001 0001	Ctrl-Q	DC1
18	12	0001 0010	Ctrl-R	DC2
19	13	0001 0011	Ctrl-S	DC3
20	14	0001 0100	Ctrl-T	DC4
21	15	0001 0101	Ctrl-U	
22	16	0001 0110	Ctrl-V	
23	17	0001 0111	Ctrl-W	
24	18	0001 1000	Ctrl-X	
25	19	0001 1001	Ctrl-Y	
26	1A	0001 1010	Ctrl-Z	
27	1B	0001 1011		ESC
28	1C	0001 1100		
29	1D	0001 1101		
30	1E	0001 1110		RS
31	1F	0001 1111		

Standard ASCII Codes

Decimal	Hexadecimal	Binary	Character
32	20	0010 0000	SP
33	21	0010 0001	!
34	22	0010 0010	"
35	23	0010 0011	#
36	24	0010 0100	$
37	25	0010 0101	%
38	26	0010 0110	&
39	27	0010 0111	'
40	28	0010 1000	(
41	29	0010 1001	)
42	2A	0010 1010	*
43	2B	0010 1011	+
44	2C	0010 1100	,
45	2D	0010 1101	-
46	2E	0010 1110	.
47	2F	0010 1111	/
48	30	0011 0000	0
49	31	0011 0001	1
50	32	0011 0010	2
51	33	0011 0011	3
52	34	0011 0100	4
53	35	0011 0101	5
54	36	0011 0110	6
55	37	0011 0111	7
56	38	0011 1000	8
57	39	0011 1001	9
58	3A	0011 1010	:
59	3B	0011 1011	;
60	3C	0011 1100	<
61	3D	0011 1101	=
62	3E	0011 1110	>
63	3F	0011 1111	?
64	40	0100 0000	@
65	41	0100 0001	A
66	42	0100 0010	B
67	43	0100 0011	C
68	44	0100 0100	D
69	45	0100 0101	E
70	46	0100 0110	F
71	47	0100 0111	G
72	48	0100 1000	H
73	49	0100 1001	I

Standard ASCII Codes

Decimal	Hexadecimal	Binary	Character
74	4A	0100 1010	J
75	4B	0100 1011	K
76	4C	0100 1100	L
77	4D	0100 1101	M
78	4E	0100 1110	N
79	4F	0100 1111	O
80	50	0101 0000	P
81	51	0101 0001	Q
82	52	0101 0010	R
83	53	0101 0011	S
84	54	0101 0100	T
85	55	0101 0101	U
86	56	0101 0110	V
87	57	0101 0111	W
88	58	0101 1000	X
89	59	0101 1001	Y
90	5A	0101 1010	Z
91	5B	0101 1011	[
92	5C	0101 1100	\
93	5D	0101 1101	]
94	5E	0101 1110	^
95	5F	0101 1111	_
96	60	0110 0000	`
97	61	0110 0001	a
98	62	0110 0010	b
99	63	0110 0011	c
100	64	0110 0100	d
101	65	0110 0101	e
102	66	0110 0110	f
103	67	0110 0111	g
104	68	0110 1000	h
105	69	0110 1001	i
106	6A	0110 1010	j
107	6B	0110 1011	k
108	6C	0110 1100	l
109	6D	0110 1101	m
110	6E	0110 1110	n
111	6F	0110 1111	o
112	70	0111 0000	p
113	71	0111 0001	q
114	72	0111 0010	r
115	73	0111 0011	s

Standard ASCII Codes

Decimal	Hexadecimal	Binary	Character
116	74	0111 0100	t
117	75	0111 0101	u
118	76	0111 0110	v
119	77	0111 0111	w
120	78	0111 1000	x
121	79	0111 1001	y
122	7A	0111 1010	z
123	7B	0111 1011	{
124	7C	0111 1100	|
125	7D	0111 1101	}
126	7E	0111 1110	~
127	7F	0111 1111	DEL
128	80	1000 0000	
129	81	1000 0001	
130	82	1000 0010	
131	83	1000 0011	
132	84	1000 0100	
133	85	1000 0101	
134	86	1000 0110	
135	87	1000 0111	BEL
136	88	1000 1000	BS
137	89	1000 1001	HT
138	8A	1000 1010	LF
139	8B	1000 1011	VT
140	8C	1000 1100	FF
141	8D	1000 1101	CR
142	8E	1000 1110	SO
143	8F	1000 1111	SI
144	90	1001 0000	
145	91	1001 0001	DC1
146	92	1001 0010	DC2
147	93	1001 0011	DC3
148	94	1001 0100	DC4
149	95	1001 0101	
150	96	1001 0110	
151	97	1001 0111	
152	98	1001 1000	
153	99	1001 1001	
154	9A	1001 1010	
155	9B	1001 1011	ESC
156	9C	1001 1100	
157	9D	1001 1101	

Standard ASCII Codes			
Decimal	**Hexadecimal**	**Binary**	**Character**
158	9E	1001 1110	RS
159	9F	1001 1111	
160	A0	1010 0000	
161	A1	1010 0001	
162	A2	1010 0010	
163	A3	1010 0011	
164	A4	1010 0100	
165	A5	1010 0101	
166	A6	1010 0110	
167	A7	1010 0111	
168	A8	1010 1000	
169	A9	1010 1001	
170	AA	1010 1010	
171	AB	1010 1011	
172	AC	1010 1100	
173	AD	1010 1101	
174	AE	1010 1110	
175	AF	1010 1111	
176	B0	1011 0000	
177	B1	1011 0001	
178	B2	1011 0010	
179	B3	1011 0011	
180	B4	1011 0100	
181	B5	1011 0101	
182	B6	1011 0110	
183	B7	1011 0111	
184	B8	1011 1000	
185	B9	1011 1001	
186	BA	1011 1010	
187	BB	1011 1011	
188	BC	1011 1100	
189	BD	1011 1101	
190	BE	1011 1110	
191	BF	1011 1111	
192	C0	1100 0000	
193	C1	1100 0001	
194	C2	1100 0010	
195	C3	1100 0011	
196	C4	1100 0100	
197	C5	1100 0101	
198	C6	1100 0110	
199	C7	1100 0111	

Standard ASCII Codes

Decimal	Hexadecimal	Binary	Character
200	C8	1100 1000	┼
201	C9	1100 1001	╔
202	CA	1100 1010	╩
203	CB	1100 1011	╦
204	CC	1100 1100	╠
205	CD	1100 1101	═
206	CE	1100 1110	╬
207	CF	1100 1111	╧
208	D0	1101 0000	╨
209	D1	1101 0001	╤
210	D2	1101 0010	╥
211	D3	1101 0011	╙
212	D4	1101 0100	╘
213	D5	1101 0101	╒
214	D6	1101 0110	╓
215	D7	1101 0111	╫
216	D8	1101 1000	╪
217	D9	1101 1001	┘
218	DA	1101 1010	┌
219	DB	1101 1011	█
220	DC	1101 1100	▄
221	DD	1101 1101	▌
222	DE	1101 1110	▐
223	DF	1101 1111	▀
224	E0	1110 0000	SP
225	E1	1110 0001	▪
226	E2	1110 0010	▫
227	E3	1110 0011	▬
228	E4	1110 0100	▭
229	E5	1110 0101	▮
230	E6	1110 0110	▯
231	E7	1110 0111	▰
232	E8	1110 1000	▱
233	E9	1110 1001	▲
234	EA	1110 1010	△
235	EB	1110 1011	▴
236	EC	1110 1100	▵
237	ED	1110 1101	▶
238	EE	1110 1110	▷
239	EF	1110 1111	▸
240	F0	1111 0000	▹
241	F1	1111 0001	►

Standard ASCII Codes

Decimal	Hexadecimal	Binary	Character
242	F2	1111 0010	‗
243	F3	1111 0011	¾
244	F4	1111 0100	¶
245	F5	1111 0101	§
246	F6	1111 0110	÷
247	F7	1111 0111	¸
248	F8	1111 1000	°
249	F9	1111 1001	¨
250	FA	1111 1010	·
251	FB	1111 1011	¹
252	FC	1111 1100	³
253	FD	1111 1101	²
254	FE	1111 1110	■
255	FF	1111 1111	

Appendix N

Technical Specifications

Printing

Printing method	Serial impact dot matrix
Printing speed	160 characters per second in 10 and 12 CPI 137 characters per second in 17 CPI
Paper feed	100 ms/line (at 1/6 inch/line) Sprocket or friction feed
Printing direction	Bidirectional, logic seeking Unidirectional in bit image modes
Character set	96 standard ASCII characters 88 international characters 64 special symbols 32 block graphics characters 189 user defined characters
Character size	2.4 mm x 2.0 mm standard 10 CPI characters
Character matrix	Standard characters: 9 dot x 9 dot Block graphics: 6 dot x 6 dot User defined: 7 dot x 4 to 11 dot Bit image modes: 7 or 8 dot x 60 dots/in. 7 or 8 dot x 120 dots/in. 7 or 8 dot x 240 dots/in.
Line spacing	1/6 or 1/8 inch standard *n*/72 inch or *n*/144 inch programmable

Column width

	Delta-10	Delta-15
10 CPI	80	136
12 CPI	96	163
17 CPI	136	233

Paper		Delta-10	Delta-15
Paper type	Single sheets	8-10 in. wide	8-15 in. wide
	Roll paper	8.5 in. wide	8.5 in. wide
	Sprocket-feed paper	3-10 in. wide	5-15.5 in. wide
Thickness	One-part forms	0.07-0.10 mm	0.07-0.10 mm
	Max. 3-part forms	0.28 mm max.	0.28 mm max.
Roll diameter		5 in. max.	5 in. max.

Printer		Delta-10	Delta-15
Dimensions	Height	148 mm (5.8 in.)	148 mm (5.8 in.)
	Width	392 mm (15.2 in.)	542 mm (21.3 in.)
	Depth	315 mm (12.4 in.)	315 mm (12.4 in.)
Weight		7.8 kg (17.2 lb.)	10.2 kg (22.5 lb.)
Power	120 VAC ± 10% 60Hz		
Ribbon	Star Micronics P/M SF-02B, or Underwood ½ in. x 2 in. spools		

Parallel interface

Interface	Centronics-compatible, 7 or 8 bit
Synchronization	By externally supplied strobe pulses
Handshaking	By ACK or BUSY signals
Logic level	TTL
Connector	57-30360 Amphenol

Serial interface

Interface	Asynchronous RS-232C
Bit rate	110, 300, 600, 1200, 2400, 4800, 9600 baud
Word length	1 start bit 7 or 8 data bits Odd, even or no parity 1 or 2 stop bits
Handshaking	Serial busy, 1 byte mode Serial busy, 1 block mode ACK mode XON/XOFF mode

Appendix O

The Parallel Interface

Delta has both a parallel interface and a serial interface to communicate with the computer that it is connected to. The operating specifications of the parallel interface are as follows:

Data transfer rate:	1,000 to 6,000 characters per second
Synchronization:	Via externally supplied $\overline{\text{STROBE}}$ pulses
Handshaking:	$\overline{\text{ACK}}$ and BUSY signals
Logic level:	Compatible with TTL level

Delta's parallel interface connects to the computer by a 36 pin connector on the back of the printer. This connector mates with an Amphenol 57-30360 connector. The functions of the various pins are summarized in Table O-1.

Functions of the Connector Signals

Communications between the computer and the Delta use many of the pins of the connector. To understand how the system of communications works we need to look at the functions of the various signals carried by the pins of the interface connector.

Pin 1 carries the $\overline{\text{STROBE}}$ pulse signal from the computer to the printer. This signal is normally held high by the computer. When the computer has data ready for the printer it sets this signal to a low value for at least 0.5 microseconds. When the printer sees this pulse on the strobe pin, it reads the data that the computer supplies on pins 2 through 9. Each of these lines carries one bit of information. A logical "1" is represented by a high signal level, and a logical "0" is represented by a low signal level. The computer must maintain these signals for a period beginning at least 0.5 microseconds before the strobe pulse starts and continuing for at least 0.5 microseconds after the strobe pulse ends.

When the Delta has successfully received the byte of data from the computer it sets pin 10 low for approximately 9 microse-

conds. This signal acknowledges the receipt of the data and so is called the $\overline{\text{ACK}}$ signal.

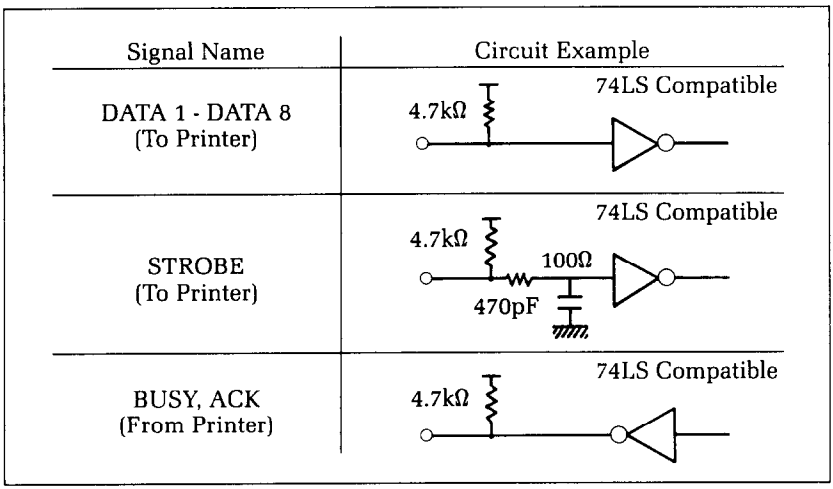


Figure O-1. *Delta interface timing diagram.*

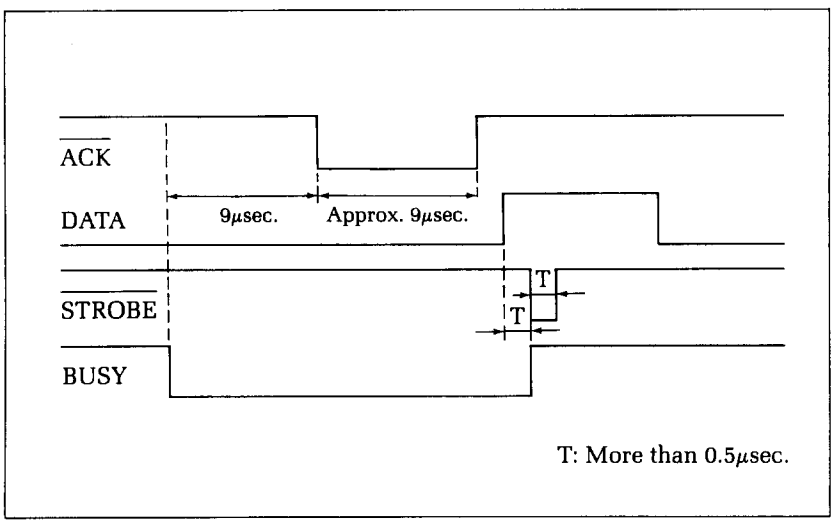


Figure O-2. *Typical interface circuit.*

Table O-1

Signal Pin No.	Signal Name	Direction	Function
1	$\overline{\text{STROBE}}$	IN	Signals when data is ready to be read. Signal goes from HIGH to LOW (for at least 0.5 microseconds) when data is available.
2	DATA1	IN	These signals provide the information of the first to eighth bits of parallel data. Each signal is at a HIGH level for a logical 1 and at a LOW level for a logical 0.
3	DATA2	IN	
4	DATA3	IN	
5	DATA4	IN	
6	DATA5	IN	
7	DATA6	IN	
8	DATA7	IN	
9	DATA8	IN	
10	$\overline{\text{ACK}}$	OUT	A 9 microsecond LOW pulse acknowledges receipt of data.
11	BUSY	OUT	When this signal goes LOW the printer is ready to accept data.
12	PAPER OUT	OUT	This signal is normally LOW. It will go HIGH if Delta runs out of paper. This signal can be held LOW permanently by turning DIP switch 2-1 off.
13	SELECTED	OUT	This signal is HIGH when the printer is on-line.
14-15	N/C		Unused.
16	SIGNAL GND		Signal ground.
17	CHASSIS GND		Printer's chassis ground, isolated from logic ground.
18	+5VDC	OUT	External supply of +5VDC.
19-30	GND		Twisted pair return signal ground level.
31	$\overline{\text{RESET}}$	IN	When this signal goes LOW the printer is reset to its power-on condition.
32	$\overline{\text{ERROR}}$	OUT	This signal is normally HIGH. This signal goes LOW to signal that the printer cannot print due to an error condition.
33	EXT GND		External ground.
34-36	N/C		Unused.

Pin 11 reports when the Delta is not able to receive data. The signal is called BUSY. When this signal is high, Delta cannot receive data. This signal will be high during data transfer, when the printer is off-line and when an error condition exists.

Appendix P

Serial Interface Specifications

Delta provides a very flexible RS232C serial interface. It can communicate at rates from 110 to 9600 baud and supports four different kinds of *handshaking*. The operating specifications of the interface are as follows:

Data transfer rate:	110-9600 baud
Word length:	1 start bit
	7 or 8 data bits
	Odd, even or no parity
	1 or 2 stop bits
Signal levels:	Mark or OFF, −3 to −15 volts
	Space or ON, +3 to +15 volts
Handshaking:	Serial busy, 1 byte mode
	Serial busy, 1 block mode
	ACK mode
	XON/XOFF mode

Delta has a DB-25 female connector on the back to connect to a computer. The functions of the pins are summarized in Table P-1

Configuring the Serial Interface

DIP switch 3 controls the configuration of the serial interface. Figure P-1 shows the location of DIP switch 3. You must remove Delta's upper case to reach this switch. See Chapter 10 for instructions on how to do this. Table P-2 describes the functions of the individual switches in DIP switch 3.

Delta's Serial Protocols

Delta has four serial protocols selected by DIP switches 3-3

and 3-4. Figure P-2 shows a typical byte of serial data and Figure P-3 shows timing charts for the 4 protocols.

Table P-1
Serial Interface Pin Functions

Signal Pin No.	Signal Name	Direction	Function
1	GND	—	Printer's chassis ground.
2	TXD	OUT	This pin carries data from the printer.
3	RXD	IN	This pin carries data to the printer.
4	RTS	OUT	This is ON when the printer is ready to receive data.
5	CTS	IN	This pin is ON when the computer is ready to send data.
6	DSR	IN	This pin is ON when the computer is ready to send data. Delta does not check this pin.
7	GND	—	Signal ground
8	DCD	IN	This pin is ON when the computer is ready to send data.
9-10	N/C		Unused
11	RCH	OUT	This is the signal line for the serial busy protocols. This pin goes OFF when Delta's buffer fills, and ON when Delta is ready to receive data. In the busy protocols this line carries the same signal as pin 20.
12	N/C		Unused
13	GND	—	Signal ground
14-19	N/C		Unused
20	DTR	OUT	Delta turns this pin ON when it is ready to receive data.
21-25	N/C		Unused.

Table P-2
DIP Switch 3

Switch	ON	OFF
3-1	7 data bits	8 data bits
3-2	Parity checked	No parity
3-3	Handshaking protocols—see below	
3-4		
3-5	Odd parity	Even parity
3-6	Data transfer rate—see below	
3-7		
3-8		

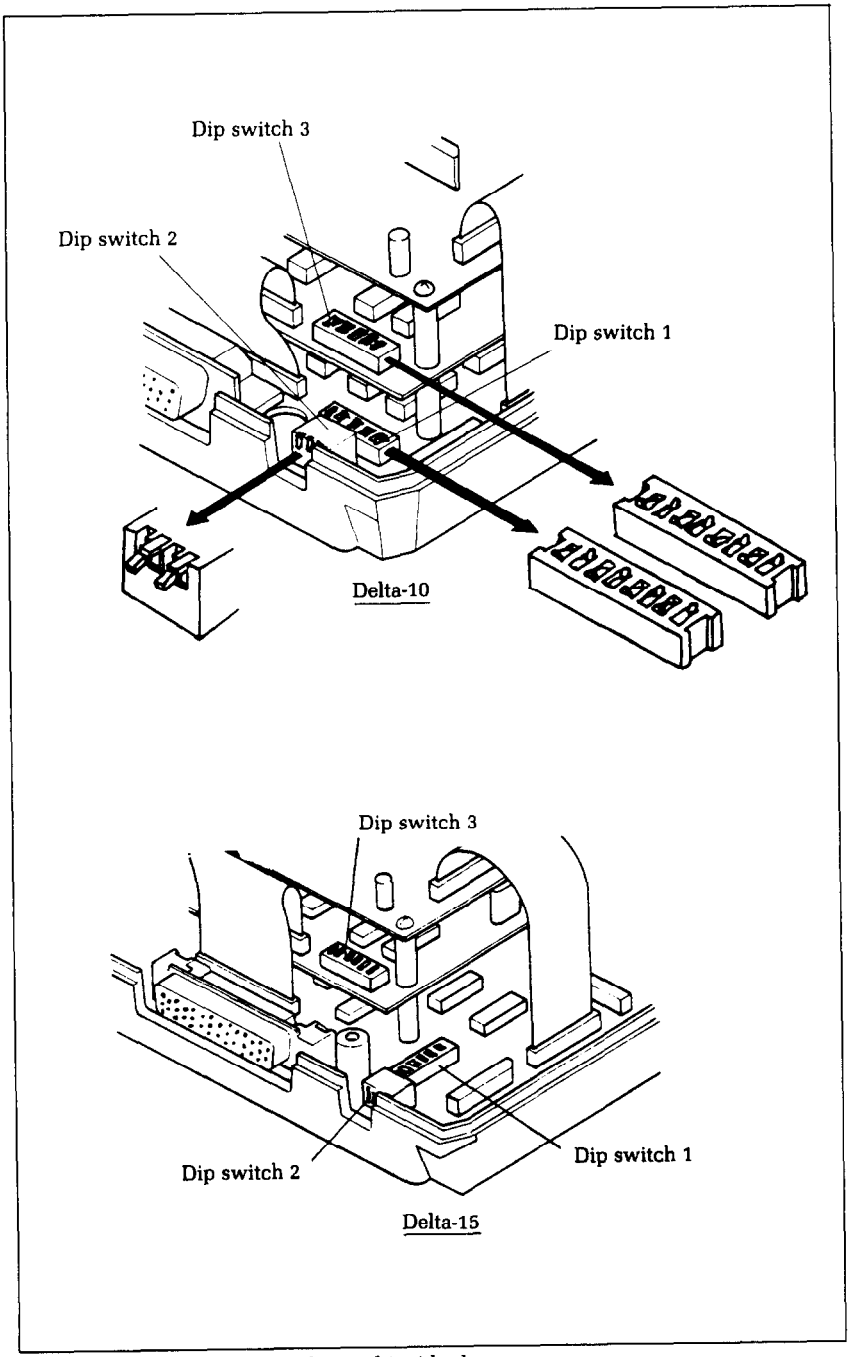


Figure P-1. *DIP switch 3 is located inside the case.*

Table P-3
Handshaking protocols

Protocol	Switch 3-3	Switch 3-4
Serial busy, 1 byte mode	OFF	OFF
Serial busy, 1 block mode	ON	OFF
ACK mode	OFF	ON
XON/XOFF	ON	ON

Table P-4
Data transfer rates

Baud rate	Switch 3-6	Switch 3-7	Switch 3-8
110	OFF	OFF	OFF
110	OFF	OFF	ON
300	OFF	ON	OFF
600	OFF	ON	ON
1200	ON	OFF	OFF
2400	ON	OFF	ON
4800	ON	ON	OFF
9600	ON	ON	ON

Serial busy protocols

In the serial busy protocols, Delta uses DTR (pin 20) and RCH (pin 11) to signal to the computer when it is able to accept data. These two pins go ON when Delta is ready to accept data. In the 1 byte mode they go OFF after each character is received. In the 1 block mode they only go OFF when Delta's buffer approaches capacity. In both cases they will stay OFF if the buffer is too full to accept more data.

XON/XOFF protocol

The XON/XOFF protocol uses the ASCII characters ⟨DC1⟩ and ⟨DC3⟩ (sometimes called XON and XOFF, respectively) to communicate with the computer. When Delta's buffer approaches capacity Delta will send a DC3 (ASCII 19) on TXD (pin 2) to tell the computer that it must stop sending data. When Delta is able to receive more data it sends a DC1 (ASCII 17) on TXD. The computer can then send more data until Delta sends another DC3.

ACK protocol

In the ACK protocol, Delta sends an ACK (ASCII 6) on TXD (pin 2) each time that it is prepared to receive a byte of data.

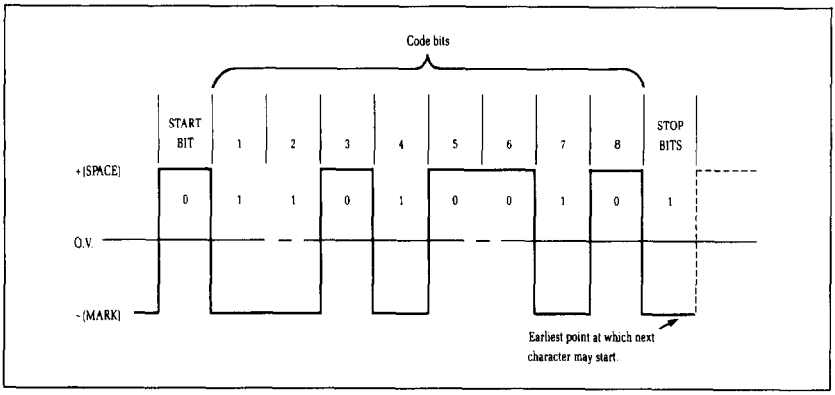


Figure P-2. *Typical data byte on the serial interface.*

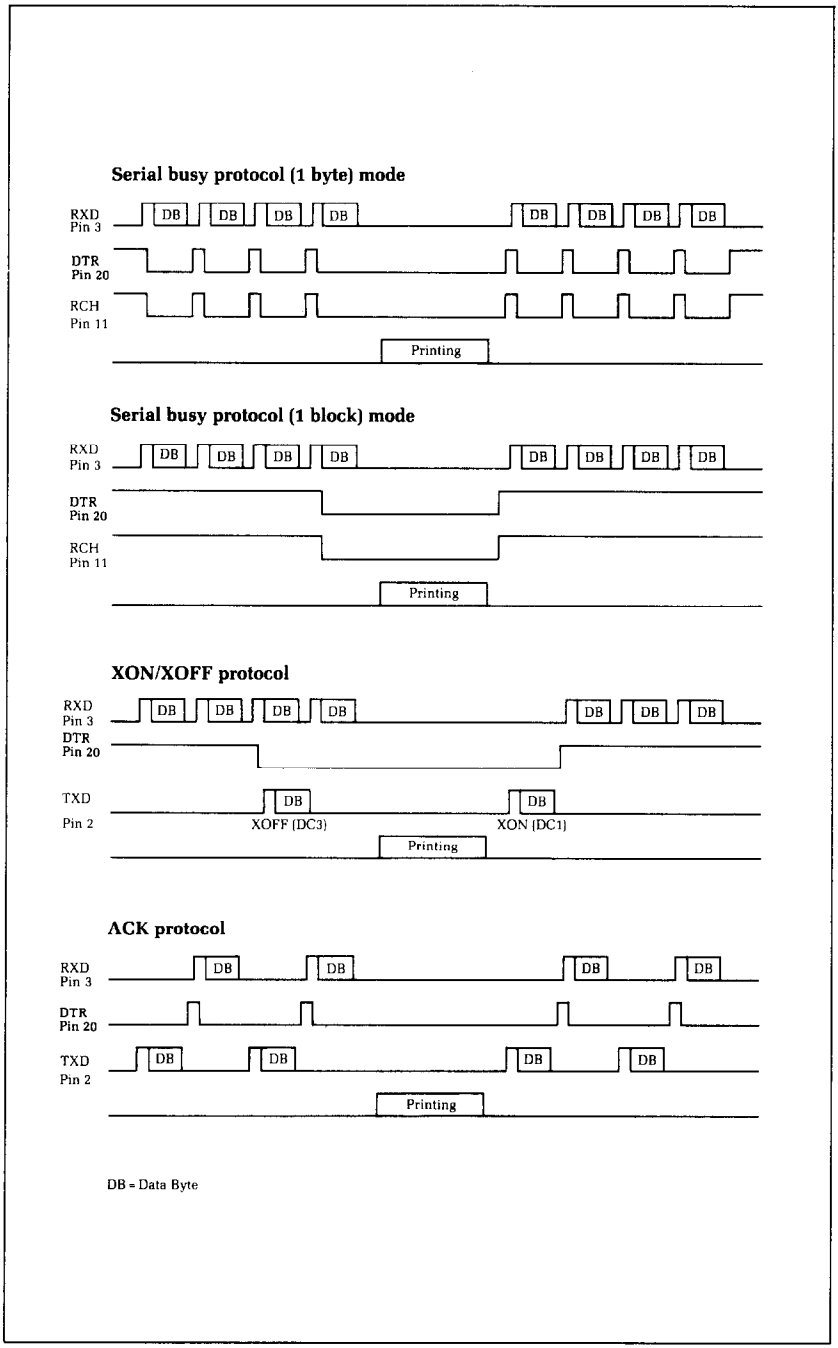


Figure P-3. *Serial protocol timing charts.*

Index

DIP Switch Settings

Switch	ON	OFF	SETTING
	DIP Switch 1		
1-1	11″ page length	12″ page length	
1-2	Normal print	Emphasized print	
1-3	10 CPI (pica pitch)	17 CPI (condensed pitch)	
1-4	Normal	Italic	
1-5	1/6″ line feed	1/8″ line feed	
1-6	International character set selection-see below		
1-7			
1-8			
	DIP Switch 2		
2-1	Paper-out detector on	Ignore paper-out	
2-2	Serial interface	Parallel interface	
2-3	7-bit interface	8-bit interface	
2-4	Auto LF with CR	LF must be from host	
	DIP Switch 3		
3-1	7 data bits	8 data bits	
3-2	Parity checked	No parity	
3-3	Handshaking protocols—see below		
3-4			
3-5	Odd parity	Even parity	
3-6	Data transfer rate—see below		
3-7			
3-8			

International character sets

Switch	USA	England	Germany	Denmark	France	Sweden	Italy	Spain
1-6	ON	OFF	ON	OFF	ON	OFF	ON	OFF
1-7	ON	ON	OFF	OFF	ON	ON	OFF	OFF
1-8	ON	ON	ON	ON	OFF	OFF	OFF	OFF

Handshaking protocols

Protocol	Switch 3-3	Switch 3-4
Serial busy, 1 byte mode	OFF	OFF
Serial busy, 1 block mode	ON	OFF
ACK mode	OFF	ON
XON/XOFF	ON	ON

Data transfer rates

Baud rate	Switch 3-6	Switch 3-7	Switch 3-8
110	OFF	OFF	OFF
110	OFF	OFF	ON
300	OFF	ON	OFF
600	OFF	ON	ON
1200	ON	OFF	OFF
2400	ON	OFF	ON
4800	ON	ON	OFF
9600	ON	ON	ON

Use the "setting" column to record the way the switches are set in your printer.

Command Quick Reference

Commands to control print style

⟨ESC⟩ "5"	Cancel italic print
⟨ESC⟩ "4"	Italic print
⟨ESC⟩ "7" n	Select international character set

Font pitch controls

⟨ESC⟩ "B" CHR$(1)	Pica pitch
⟨ESC⟩ "B" CHR$(2)	Elite pitch
⟨ESC⟩ "B" CHR$(3)	Condensed pitch
CHR$(18)	Pica pitch
CHR$(15)	Condensed pitch
⟨ESC⟩ CHR$(15)	Condensed pitch
⟨ESC⟩ "W" CHR$(1)	Expanded print
CHR$(14)	Expanded print
⟨ESC⟩ CHR$(14)	Expanded print
⟨ESC⟩ "W" CHR$(0)	Cancel expanded print
CHR$(20)	Cancel expanded print

Special print modes

⟨ESC⟩ "G"	Double-strike print
⟨ESC⟩ "H"	Cancel double-strike print
⟨ESC⟩ "E"	Emphasized print
⟨ESC⟩ "F"	Cancel emphasized print
⟨ESC⟩ "-" CHR$(1)	Start underlining
⟨ESC⟩ "-" CHR$(0)	Stop underlining
⟨ESC⟩ "S" CHR$(0)	Superscript on
⟨ESC⟩ "S" CHR$(1)	Subscript on
⟨ESC⟩ "T"	Cancel super and subscripts

Commands to control vertical position of the print head

CHR$(10)	Line feed
⟨ESC⟩ "0"	Set line feed to 1/8 inch
⟨ESC⟩ "1"	Set line feed to 7/72 inch
⟨ESC⟩ "2"	Set line feed to 1/6 inch
⟨ESC⟩ "A" n	Set line feed to n/72 inch
⟨ESC⟩ "3" n	Set line feed to n/144 inch
⟨ESC⟩ "J" n	Single line feed of n/144 inches

Form feed controls

CHR$(12)	Form feed
⟨ESC⟩ "C" n	Set page length to n lines
⟨ESC⟩ "C" CHR$(0) n	Set page length to n inches
⟨ESC⟩ "R" n	Set top margin at line n
⟨ESC⟩ "N" n	Set bottom margin at n lines
⟨ESC⟩ "O"	Cancel top and bottom margins

Vertical tabs

CHR$(11)	Vertical tab
⟨ESC⟩ "P" . . . CHR$(0)	Set vertical tabs
⟨ESC⟩ "a" *n*	Advance *n* line feeds

Commands to control horizontal position of the print head

CHR$(13)	Carriage return
⟨ESC⟩ "M" *n*	Set left margin at column *n*
⟨ESC⟩ "Q" *n*	Set right margin at column *n*
CHR$(9)	Horizontal tab
⟨ESC⟩ "D" . . . CHR$(0)	Set horizontal tabs
⟨ESC⟩ "b" *n*	Tab over *n* columns
CHR$(8)	Backspace

Download character commands

⟨ESC⟩ "*" CHR$(1) *n1 n2 m1 m2 . . . m11*	Define download character
⟨ESC⟩ "*" CHR$(0)	Copy standard ROM characters to download RAM
⟨ESC⟩ "X" CHR$(1)	Use proportional download characters
⟨ESC⟩ "X" CHR$(0)	Cancel proportional download characters
⟨ESC⟩ "$" CHR$(1)	Use normal download characters
⟨ESC⟩ "$" CHR$(0)	Cancel normal download characters

Commands to control graphics

⟨ESC⟩ "K" *n1 n2*	Normal density graphics
⟨ESC⟩ "L" *n1 n2*	Double density graphics
⟨ESC⟩ "y" *n1 n2*	Double speed, double density graphics
⟨ESC⟩ "z" *n1 n2*	Quadruple density graphics

Macro instruction commands

⟨ESC⟩ "+" . . . CHR$(30)	Define macro
⟨ESC⟩ "!"	Use macro

Other function codes

⟨ESC⟩ ">"	Set eighth bit to 1
⟨ESC⟩ "="	Set eighth bit to 0
⟨ESC⟩ "#"	Accept eighth bit as is
CHR$(127)	Delete last character
CHR$(19)	Off line
CHR$(17)	On line
CHR$(7)	Sounds bell
⟨ESC⟩ "Y" CHR$(0)	Disable bell
⟨ESC⟩ "Y" CHR$(1)	Enable bell
⟨ESC⟩ "8"	Ignore paper-out signal
⟨ESC⟩ "9"	Enable paper-out signal
⟨ESC⟩ "U" CHR$(1)	Unidirectional print
⟨ESC⟩ "U" CHR$(0)	Bidirectional print
⟨ESC⟩ "@"	Reset the printer