A BRaille PRINTER

Andrew W. Appel and Douglas W. Jones
Medical Computing Laboratory
School of Basic Medical Sciences
University of Illinois
Urbana, IL 61801

August 1980

1 Introduction

This braille printer design resulted from the observation that existing braille printers were either slow and expensive, or very expensive. These printers use relatively old designs, and contain many moving parts. Furthermore, they require a significant amount of software support from the host computer. On the basis of these observations, and a quick check of the unit prices of key components (micro-computer, stepping motors, and solenoids), it was determined that a fast braille printer with a built in micro-computer could be built for about $1000 worth of parts (1979 cost). The inclusion of the microcomputer would greatly simplify the mechanical design.

Two printers representative of the current state of the art are SAGEM's TEM/REM 8 BR and TRIFORMATION's LED-120. The SAGEM printers run at 15 characters per second, while the LED-120 runs at 120 characters per second. Both printers are implemented with "random" electronic logic, although the LED-120 uses some LSI; both use mechanical control systems. Both printers accept the ASCII character set and literally map 54 of the characters to the 64 braille signs (including space). The mapping provided is correct for the letters in Grade 1 English Braille, but departs from this for the remainder of the character set. Thus, typical computer output is difficult for even trained users to interpret, and special translation programs are required if standard Grade 1 or Grade 2 English Braille are to be produced.

The construction of this printer was financed by the School of Basic Medical Sciences and the School of Clinical Medicine of the University of Illinois at Urbana Champaign. The Physics Department supported the project with its machine shop facilities, and the microcomputer was purchased with Federal funds. Initial software for the microcomputer was written by students from Computer Science 350. Thanks for this support are due to: Frank Mabry, Bruce Sherwood, Peter Maggs, and A. B. Baskin.
The mechanical operation of the printer is quite simple. There are two stepping motors, one responsible for paper advance, and one for carriage motion. The carriage itself carries three solenoids, one for each of the three rows of bosses which make up braille text. The paper is embossed by stylus points ground on the ends of the solenoid plungers. Each stylus mates with a matching row of dies in the back bar (above the paper) which shape the bosses in the paper. While one solenoid is embossing the upper right dot of one character, another will be positioned under the center right dot of the character five print positions over, and the third will be positioned under the bottom right dot ten positions over.

The interface electronics performs the function of using the logic levels appearing on the outputs of the microcomputer to control the power delivered to the motors and solenoids. Additionally, the interface electronics includes a loudspeaker to be used for audible indication of various printer conditions. Stepping motors have multiple windings; motor rotation is accomplished by energizing the windings in the appropriate sequence. Thus, one bit is needed to control each of the windings. There are 4 windings on the large carriage motion motor, and three on the paper advance motor. Allocating one bit to the speaker and one to each of the three solenoids, a total of eleven bits are needed from the microcomputer.

The microcomputer itself is responsible for timing the transitions on each of the output bits to produce the desired mechanical motion, polling the input switches, and managing the asynchronous communications line. The input switches include an internal "carriage at left margin" switch and three internal mode switches, as well as the external page feed, line feed, and braille level select switches. Since each switch may be on or off, this is seven input bits. The microcomputer has the responsibility for translating the incoming ASCII characters to braille, and for converting the braille to motor and solenoid control sequences. Should the microcomputer discover anything wrong, it will signal this on the speaker. It may also transmit status reports on the asynchronous line.

The prototype achieves at present a speed of 13 characters per second. With patient optimization of the software delays to reduce resonance problems, a speed of 20 or 30 characters per second is likely.
2.1 The ASCII Printing Characters

ASCII (the American Standard Code for Information Interchange) is a 7 bit code for the entire alphabet in upper and lower case, the digits, 32 punctuation and accent marks, space, and 33 control characters. The 7 bits are numbered 0 to 6, with bit 0 being the least significant (rightmost) and bit 5 being the most significant (leftmost). In traditional communications terminology, the binary one and zero are called mark and space. On a serial communications line, the ASCII codes are sent least significant bit first. Asynchronous transmission involves a leading mark at the start of each character, and at least one trailing space at the end. These are called start and stop bits. On the RS-232 interface, marks are represented by negative voltages, spaces by positive voltages.

The ASCII character set is traditionally presented as a matrix, where the 3 most significant bits of the 7 bit representation label the columns and the 4 least significant bits label the rows. The printing characters occupy most of columns 2 through 7 (codes 20 to 7E). Note that space (code 20) is considered to be a printing character; the remaining codes are the control characters (codes 00 to 1F and code 7F, shaded in the diagram).

<table>
<thead>
<tr>
<th>COLUMNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>0 #== 0 @ P ' p</td>
</tr>
<tr>
<td>1 #== ! 1 A Q a q</td>
</tr>
<tr>
<td>2 #== &quot; 2 B R b r</td>
</tr>
<tr>
<td>3 #== $ 3 C S c s</td>
</tr>
<tr>
<td>4 #== $ 4 D T t</td>
</tr>
<tr>
<td>5 #== $ 5 E U u</td>
</tr>
<tr>
<td>R 6 #== &amp; 6 F V f v</td>
</tr>
<tr>
<td>O 7 #== ' 7 G W g w</td>
</tr>
<tr>
<td>W 8 #== ( 8 H X h x</td>
</tr>
<tr>
<td>S 9 #== ) 9 I Y i y</td>
</tr>
<tr>
<td>A #== * : J Z j z</td>
</tr>
<tr>
<td>B #== + ; K { k</td>
</tr>
<tr>
<td>C #== , &lt; L \ l</td>
</tr>
<tr>
<td>D #== - = M } m</td>
</tr>
<tr>
<td>E #== . &gt; N ^ n</td>
</tr>
<tr>
<td>F #== / ? O _ o #=</td>
</tr>
</tbody>
</table>

Note that on some printers and terminals, the ^ and _ characters are presented as and . Sometimes, the 7F code is used as a printing character. These changes are not part of the ASCII standard. It is standard to use the ' as either an accent mark (acute), as an apostrophe or as a single quote.

2.2 The ASCII Control Characters

The control characters are divided into broad groups which can be distinguished by the first two letters of their CCITT names as follows:

TC - transmission control.
FE - format effector, used to position data for display.
DC - device control, for auxiliary function control.
IS - information separator, used for machine processing.

The control characters which have meanings when recieved by
the printer are marked by an asterisk while those which it
transmits are marked with a plus in the following table.
Note that the CCITT name follows in parentheses when it
differs from the ASCII name. All unmarked characters in the
following table are ignored by the printer; most of these
are simply not applicable, but NUL and DEL must be ignored.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>NUL</td>
<td>10</td>
<td>DLE</td>
</tr>
<tr>
<td>01</td>
<td>SOH</td>
<td>11</td>
<td>*DC1</td>
</tr>
<tr>
<td>02</td>
<td>STX</td>
<td>12</td>
<td>DC2</td>
</tr>
<tr>
<td>03</td>
<td>ETX</td>
<td>13</td>
<td>DC3</td>
</tr>
<tr>
<td>04</td>
<td>EOT</td>
<td>14</td>
<td>DC4</td>
</tr>
<tr>
<td>05</td>
<td>*ENQ</td>
<td>15</td>
<td>+NAK</td>
</tr>
<tr>
<td>06</td>
<td>+ACK</td>
<td>16</td>
<td>SYN</td>
</tr>
<tr>
<td>07</td>
<td>*BEL</td>
<td>17</td>
<td>ETB</td>
</tr>
<tr>
<td>08</td>
<td>*BS</td>
<td>18</td>
<td>CAN</td>
</tr>
<tr>
<td>09</td>
<td>*HT</td>
<td>19</td>
<td>EM</td>
</tr>
<tr>
<td>0A</td>
<td>*LF</td>
<td>1A</td>
<td>SUB</td>
</tr>
<tr>
<td>0B</td>
<td>*VT</td>
<td>1B</td>
<td>ESC</td>
</tr>
<tr>
<td>0C</td>
<td>*FF</td>
<td>1C</td>
<td>FS</td>
</tr>
<tr>
<td>0D</td>
<td>*CR</td>
<td>1D</td>
<td>GS</td>
</tr>
<tr>
<td>0E</td>
<td>SO</td>
<td>1E</td>
<td>RS</td>
</tr>
<tr>
<td>0F</td>
<td>SI</td>
<td>1F</td>
<td>US</td>
</tr>
</tbody>
</table>

7F   DEL

ENQ - on receipt, causes an ACK to be transmitted when the
printer is able to accept more data, unless there is an
error condition, in which case it will send a NAK.

BEL - on receipt, sounds a brief (1/4 second) tone (2khz).

BS - on receipt, signals that the next character will
overprint the previous character. See note on braille
overprinting.

HT - on receipt, advances the print position to the next tab
stop.

LF - on receipt, terminates current line, advances to next
line. Note special interaction with overprinting.

VT - same as LF, but advances paper by only half as much.

FF - on receipt, terminates current line and advances to
top of next page (assumes 11 inches to the page).

CR - on receipt, terminates current line; any printing
characters arriving before the next line terminator will
overprint the previous line from the beginning. See
note on braille overprinting.

ESC - when followed by one of the following codes, switch
printing modes. Otherwise, both the ESC character
and the following character are ignored.

'A' - Use "transparent" mode, in which the
low-order 6 bits of any printing character
directly correspond to the 5 dots of the braille cell.

'5' - Use the Provisional Braille Code for Computer Notation.

'6' - Use English Braille, grade 1 or 2, depending upon the setting of the internal mode switch.

These remain in effect until another escape sequence is sent, or until the mode switch on the top of the printer is switched.
2.3 A Brief Introduction to Braille

The authoritative definition of braille is contained in the booklet "English Braille American Edition 1959" published by the American Printing House for the Blind, Louisville, Kentucky. Each braille character occupies a cell two dots wide by three dots high, with a nominal spacing of 0.1 inches between dots within a cell, and with cells repeated 4 per inch, and lines spaced roughly two per inch. Dots are embossed to protrude about 0.025 inches above the paper, and are about 0.07 inches in diameter. Conventionally, the dot positions are numbered from 1 to 6, starting at the upper left, down the left column and ending at the lower right, as follows:

```
 1 4
 2 5
 3 6
```

It is common practice to refer to a braille character in print by hyphenating together the dots that are raised, so "a" is 1, "b" is 1-2, "c" is 1-4, and "z" is 1-3-5-6.

The alphabet is conventionally presented as follows:

```
1 2 3 4 5 6 7 8 9 0
a b c d e f g h i j
k l m n o p q r s t
u v w x y z
```

Note that all alphabetic characters use a dot from the first column and a dot in the top row. This allows them to be distinguished from each other without ambiguity when they appear out of context. In a sense, braille is an asynchronous code, and the leftmost and topmost dots in a cell serve the function of start bits. There are a total of 44 braille characters which are unambiguous out of context in this way, the remaining 19 may be disambiguated by context, and are used for punctuation and other purposes, for example:

- **comma:** :`
- **period:** `
- **capital sign:** (ie: y e s = Yes)
- **numeral sign:** (ie: a b c = 123)
- **letter sign:** (ie: e e e = 5e, e e e = 5E)

Grade 1 English Braille is somewhat more complex than the above; English Braille, or Grade 2 Braille is much more complex. In Grade 2 Braille, most of the 53 codes have multiple meanings, allowing single symbols for 52 common words and diphthongs, such as:

```
ch gh sh th wh ed er ou ow ar
```
This braille printer has four modes in which it can interpret the ASCII printing characters. The primary two modes are Grade 1 and Grade 2 Braille; in these modes, the internal microprocessor does its best to produce readable output which corresponds to these two standards. The third mode of operation is transparent mode, in which a remote computer can directly control the printer operation for applications such as music or math braille, Grade 3 Braille, or more complete adherence to the rules of Grade 2 Braille. The fourth is the Provisional Braille Code for Computer Notation, which may be particularly appropriate for transcribing computer programs.

Selection between computer notation and English braille is accomplished both by remotely sent escape sequences, and by a switch switch on the top of the printer. This is because the nature of the material being printed governs the grade which is appropriate. For example, if a computer program is being printed, exact spelling and punctuation are important, so computer notation should be used; on the other hand, when printing text, grade 2 will save a considerable amount of paper, and is faster to read.

Transparent mode is entered whenever ESC 4 is received and remains in effect until the next escape sequence, or until the mode select switch on top of the printer is switched. In transparent mode, the control characters are interpreted normally, but the ASCII meanings of the printing characters are ignored. Instead, the least significant 6 bits of the ASCII character are taken to indicate the 6 positions in the cell to be printed. The bit - dot relation is:

ASCII bit number:  6 5 4 3 2 1 0
Braille dot number: x 1 2 3 4 5 6

A note on overprinting: In many computer applications, two or more characters are printed over each other. For example, "b" overprinted with "/" ("/\") is commonly used as a printing character for blank. Some programming languages, such as APL, make very extensive use of overprinting. Braille presents significant difficulties for overprinting because the superposition of any two braille characters simply produces another character. This printer solves this problem by printing overprints with a vertical displacement of 2 dots. As a result, any line containing overprints will be 5 dots high (instead of the normal 3), and the space to the next line will be appropriately increased. Multiple overstrikes will be simply superimposed, and overprinting
of spaces will not cause any displacement. Incidentally, this solution allows the 8 dot cell supported by the TRIFORMATION LED-120 to be reproduced.
English Braille, grade 1, is currently supported on this printer. Following is a list of rule references taken from "English Braille, American Edition", indicating which rules for grade 1 Braille are supported.

I. Punctuation Signs

All supported as given except:
- Opening double quotation mark: no ASCII equivalent
- Opening single quotation mark: no ASCII equivalent; apostrophe is used.
- Closing single quotation mark: same.
- Ellipsis: Three periods are used, as in ASCII.

1. Use and order of punctuation follows input text.
2. See above
3. Supported exactly
4. Supported exactly
5. Supported; follows input text.
6. Follows input text. (ASCII will use -- for dash)
7. See above
8. Following order will be used:
   Open parenthesis or bracket
   Open quotation sign
   (Italics not supported in ASCII)
   Apostrophe
   Capital or letter sign
   Accent sign

II. Special Braille Composition Signs

9. Supported almost exactly. The only case among the examples given on page 11-12 of the ink-print edition which is printed differently is C/O, in which the second capital sign would be omitted.
10. Italics are not supported in ASCII.
11. Not supported.
12. Supported:
   a. (1) When any uncapitalized letter, not just those from "a" to "j", follows a number or is separated by a hyphen following a number, the letter sign will be used.
   (2) Letters standing alone in the input text will be preceded by a letter sign.
   (3) Combinations of letters standing alone will NOT be preceded by a letter sign; individual letters will.
   (4) Letter sign will be used.
   b. (1) Supported.
   (2) Letter sign will be used for uncapitalized letters.
   (3) Letter sign will be used for uncapitalized k-z.
   (4) Letter sign will be used for uncapitalized k-z.
   (5) Letter sign will be used when apostrophe follows.
   (6) Letter sign will be used when period follows.
13. Follows input text.

VIII. Numbers

28. Cardinal numbers
(a) Supported
(b) Supported
(c) Supported
(d) Not supported.
(e) Mostly supported.
(f) Period is used instead of 4-5 for decimal point.
(g) Supported, except for decimal point.
(h) Follows input text.
(i) Follows input text.
(j) Supported, except that hyphen ( - ) is used for minus. Note that in standard ASCII practice, asterisk is used for times, and slash is used for divide. Dot 4 precedes the words for the symbols +, =, <, etc., as per section 31-a. Translations:

+ (dot 4) plus
\= (dot 4) eq
\< (dot 4) lt
\> (dot 4) gt

29. Follows input text.

30. Follows input text.
Text may be printed according to the Provisional Braille Code for Computer Notation (1972). Only sections 3 and 10 pertain to the printing of computer programs, and both are supported exactly (note that only those characters in section 3 which are present in the ascii character set can be supported).
2.5 Operation of the Braille Printer:

The printer is a compact unit, measuring 12" by 24" by 5". It was designed with the needs of the blind user in mind, so it is extremely simple to operate. Loading the paper is accomplished by feeding it under a bar, positioning the holes (which run down both sides of the paper) over the appropriate pins (which protrude through the top of the printer), and lowering two levers (which automatically snap into place). The whole operation takes only a few seconds even for someone unfamiliar with the machine.

The paper on which braille has just been printed travels across the top of the printer from front to back, so that the lines which have just been printed may be read.

The number of external controls has been minimized; there are, of course, no indicators could not be read by a blind person (there are no indicators at all!). There are four controls, all located on the top of the printer, and easily accessible:

1. On-Off switch
2. English Braille / Computer notation select switch.
3. Form Feed button
4. Line Feed button

There are a few internal control switches which need be set only at installation time or when maintenance is being done on the printer:

1. Wide/Narrow paper selection
2. Baud Rate selection
3. Run/Test mode selection

Variable Paper Widths:

The printer can accommodate either 9.5-inch-wide paper (that is, 8.5 inch paper with a half-inch track on either side for sprocket holes), or 12-inch wide paper. Switching between the two paper widths requires a screwdriver.
3 Construction
3.1 Bottom Assembly
3.1.1 Baseplate
3.1.2 Back Assembly
3.1.2.1 Back Plate
3.1.2.2 Fan Assembly
3.1.2.2.1 Fan
PAMOTOR Miniature Fan, Model 8500 C, 115v, 60Hz, 43cfm.
3.1.2.2.2 Finger Guard
PAMOTOR Finger guard, Model 5503.
3.1.2.3 Fuse Holder
LITELFUSE 3AG Panel Fuseholder, No. 341001;
with 3AG 3A fuse, L.F. No. 312003.
3.1.2.4 Line Cord
BELDEN 3-Conductor Line Cord, 16 awg, 9 ft, No. 17419s.
3.1.2.4.1 Strain Relief Grommet
HEYCO, for above 16-3 cord in a 5/8 in. hole.
3.1.2.5 Terminal Strip
CINCH Barrier Terminal Block 10-140 (3/8 in. spacing).
3.1.2.6 Delta Connector and Cable
3.1.2.6.1 Delta Connector
3M Scotchflex 25 pin Delta Socket No. 3483-1000.
3.1.2.6.2 Microcomputer Connector
3M Scotchflex 26 pin Socket, no. 3399-6026.
3.1.2.6.3 Cable
3M Scotchflex 26 wire flat cable, no. 3302/26, 10 inch;
connectors on one end spread in pairs to mate with wider
spacing needed by delta connector, conductor number 26
removed at spread end, conductor number 1 cut 4 inches
from spread end, separated back to one inch from spread
end.
3.1.3 Heat Sink Assembly
3.1.3.1 Heat Sinks (2 each)
INTERNATIONAL RECTIFIER Heat Exchanger No. HE50-3A2;
drilled for TO-3 socket and tie points.
3.1.3.1.1 Sockets (2 each)
CINCH Power Transistor Socket 2TS-2.
3.1.3.1.2 Terminal Strips (2 each)
CINCH Terminal Strip No. 2003 (3 terminal).
3.1.3.2 Heat Sink Side Plates (2 each)
3.1.4 Large Transformer Assembly
3.1.4.1 Separator Block
3.1.4.2 Terminal Strip
CINCH Barrier Terminal Block 3-140 (3/8 in. spacing).
3.1.4.3 18V Transformers (2 each)
TRIAD Dry Disk Rect. Transformer F-49U (17-18Vrms, 6A).
3.1.5 Small Transformers
3.1.5.1 6.3V
STANCO Filament Transformer P-6466 (6.3Vrms, 3A).
3.1.5.2 24V C.T.
STANCO Filament Transformer P-9189 (25.2Vrms, CT, 1A).
3.1.6 Capacitor Mount
3.1.7 Circuit Card Assembly
3.1.7.1 Heat sink angle
3.1.7.2 Card
3.1.7.3 Interface Connector
3.1.7.4 Interface Cable
3.1.7.4.1 Connectors (3 each)
3M Scotchflex 26 pin Socket, no. 3399-6026.

3.1.7.4.2 Cable
3M Scotchflex 26 wire flat cable, no. 3302/26, 18 inch; one end split 6 inches into 14 and 12 wire parts.

3.1.8 Fuse Block
LITTELFUSE 3AG Solder Terminal 2 Fuse Block, No. 357002.

3.1.9 Rectifier
INTERNATIONAL RECTIFIER Bridge Rectifier 100PB2P (10A).

3.1.10 Terminal Strips (3/8 in. barrier spacing).
3.1.10.1 CINCH Barrier Terminal Block 4-141
3.1.10.2 CINCH Barrier Terminal Block 5-141
3.1.10.3 CINCH Barrier Terminal Block 6-141
3.1.10.4 CINCH Barrier Terminal Block 7-141
3.1.10.5 CINCH Barrier Terminal Block 12-141

3.1.11 Tie Points (3 each)
CINCH Terminal Strip No. 2004 (4 terminal).

3.1.12 Microswitch Assembly
3.1.12.1 Adjusting Block
3.1.12.2 Microswitch
MICRO SWITCH Short Roller Lever Std Switch No. BZ-2RW22.

3.1.13 Speaker
OAKTRON 2.5 in. Speaker, 1 Watt, 8 ohm, No. 25ARC.

3.1.14 Angle
3.1.14.1 Back
3.1.14.2 Front
3.1.14.2.1 Ribbon Cable Clamp

3.1.15 Feet

3.2 Print Assembly
3.2.1 Paper Advance Mechanism
3.2.1.1 Paper Advance Plate
3.2.1.2 Motor Assembly
3.2.1.2.1 Motor Mount Plate
3.2.1.2.2 Angles
3.2.1.2.3 Gear
BOSTON GEAR 1 in., 32 pitch, brass, No. Y3232.
3.2.1.2.4 Motor
EASTERN AIR DEVICES Variable Reluctance Stepping Motor, 28V, 3 phase, 35 ounce-in. stall torque, No. LD20ACL-12.
3.2.1.3 Bearing Mounts
3.2.1.4 Shaft Assembly
3.2.1.4.1 Sprockets
LaVEZZI MACHINE WORKS Aluminum Sprocket, .500 pitch, 12 tooth, No. L12P25K.
3.2.1.4.2 Gear
3.2.1.4.3 Washers
3.2.1.4.4 Bearings
BOSTON GEAR Self Lubricating Bronze Bearing, .25 in. ID, .375 OD, .5 long.
3.2.1.4.5 Shaft
3.2.1.5 Terminal Strip
CINCH Barrier Terminal Block 4-140 (3/8 in. spacing).

3.2.2 Motor Mount
3.2.2.1 Motor Mount Plate
3.2.2.2 Motor
SUPERIOR ELECTRIC Slo-Syn Stepping Motor, M092-FD09.
3.2.2.3 Winding Drum
3.2.3 End Assemblies (2 each)
3.2.3.1 End Plates (2 each)
3.2.3.2 Pulleys (4 each)
3.2.3.2.1 Bearings (4 each)
   SCHATZ MANUFACTURING Ball Bearing, 5.2 in. ID, 1 in. OD,
   No. BR9903; seals and grease removed, re-lubricated with
   light grease.
3.2.3.2.2 Rims (4 each)
3.2.3.2.3 Shafts (4 each)
3.2.3.3 Bumpers (2 each)
   WALDOM Recessed Screw Rubber Bumper, 3/8 in. high, 3/4
   in. wide, 1/8 in. hole, 7/32 in. recess, No. B-324.
3.2.4 Front Bar
3.2.5 Print Bar Assembly
3.2.6 Shafts (2 each)
   THOMSON INDUSTRIES Solid 60 Case Shaft, 3/8 in. Dia,
   Tolerance Class L, 22.0 in. long.
3.2.7 Carriage
3.2.7.1 End Plates
3.2.7.1.1 Cable Clamp Bolts
3.2.7.1.2 Fine Adjustment Mechanisms
3.2.7.1.3 Linear Bearing Assemblies
3.2.7.1.3.1 Linear Bearings
   THOMSON INDUSTRIES Super-6 Ball Bushing.
3.2.7.1.3.2 Washers
3.2.7.1.3.3 External Retaining Rings
   THOMSON INDUSTRIES External Retaining Rings, Type W,
   No. W-375.
3.2.7.1.4 Ribbon Cable Clamp
3.2.7.2 Bottom Assembly
3.2.7.2.1 Bottom Plate
3.2.7.2.2 Terminal Strip
   KULKA single row terminal strip, 1/4 inch barrier
   spacing, 6 terminal, series 400 (CINCH type 171 will
   do in a pinch, though it is double row).
3.2.7.2.3 Rubber Mat
3.2.7.3 Top Assembly
3.2.7.3.1 Top Plate
3.2.7.3.2 Stylus Guide Bar
3.2.7.3.3 Solenoids
   GUARDIAN ELECTRIC Intermittent Push Solenoid, 24VDC,
   15 Ohm, 64 ounce at 1/16 inch, No. TP8 16-INT-24D.
3.2.7.3.3.1 Stylus
   Push Plunger removed from 3.2.7.3.3, straightened if
   needed, and ground to stylus point (Note: this grinding
   should be deferred until the print bar (3.2.5) has been
   completed). Groove filed 1/16" from bottom to hold
   spring.
3.2.7.3.3.2 Springs
   Tapered springs, 1/2" long; tight fit around large end
   of stylus at small end; loose fit at large end.
3.2.7.3.4 Alignment Pins
3.2.8 Cable
3.2.8.1 Cable
   SULLIVAN PRODUCTS Pylon Brand, 7 Strand Stainless Steel,
   .018 in. dia. Control Line, (50 lb. test), No. 018-60.
3.3 Cabinet
3.3.1 Front Plate
3.3.2 Side Plates
3.4 Lid
3.4.1 Computer Assembly
3.4.1.1 Single Board Computer
   CROMEMCO Single Card Computer
3.4.1.2 Card Guides
3.4.1.2.1 Metal Part
3.4.1.2.2 Plastic Part
3.4.1.2.3 Retainer
3.4.1.3 Connectors
3.4.1.3.1 S-100 Connector
3.4.2 Internal Switch Assembly
3.4.2.1 Angle
3.4.2.2 Switches
   ALCOSWITCH On-off-on SPDT Miniature Switch, No. MST-105E
   (ALCO MTA-105E and C&K 7103 are equivalent)
3.4.3 External Switches
3.4.3.1 Power
   ALCO MTA-106D Snap Action SPST Canopy Switch
   15A, 125VAC
3.4.3.2 Print Option
   ALCOSWITCH Miniature SPDT Toggle Switch, No. MTA-106D;
   this switch is small to make it easily distinguishable
   from the power switch (C&K 7101 is equivalent).
3.4.3.3 Line Feed and Page Eject Buttons
   GRAYHILL Push Button Switch, SPST NO, No. 4001 (GRAYHILL
   No. 2201, snap action, may be preferable); the important
   thing is that the switch button be relatively large
   without requiring much back panel space.
3.4.4 Terminal Strip
   CINCH Lug Type Terminal Strip No. 54 (4 lug).
3.4.5 Wire Hold Downs
   WALDOM Nylon Cable Clamp, 3/8 inch dia., CCN-35.
3.4.6 Cable to Switches
3.4.6.1 Connector
   3M Scotchflex 26 pin Socket, no. 3399-6026.
3.4.6.2 Cable
   3M Scotchflex 26 wire flat cable, no. 3302/25, 14 inch.
3.4.7 Paper Hold-downs (2 each)
3.4.7.1 Struts (2)
3.4.7.2 Mounting Blocks (2)
3.4.7.3 Ball Bearings (4)
   5/8" diameter, 1/4" bore, 0.2" wide
3.4.7.4 Pivot Pins (2)
3.4.7.5 Pivot Washers (2)
3.4.7.6 Bearing Washers (8)
3.4.7.7 Springs (2)
   1/8" dia. coil tension springs,
   1.3" long unextended
3.4.8 Back Hinges (2 each)
4 Electronics

4.1 Driver Circuits (11)

4.1.1 Transistors (11)
RCA SK 3180 NPN Si Darlington Transistor
\[ P_T=55W, I_c=10\,\text{mA}, V_{cb0}=80V, V_{ce0}=30V, \]
\[ V_{eb0}=5V, R_{fe}=20K, \text{Case TO-220} \]

4.1.2 Pull-up Resistors (11)
470 Ohm, 1/8 Watt.

4.1.3 Diode (10)
1N4002

4.1.4 TTL Buffer (2)
SN7407N hex noninverting buffer

4.1.5 Power dissipation resistors (4)
5 Ohm, 5 Watt

4.1.6 Resistor for speaker circuit
12 Ohm, 2 Watt

4.1.7 Capacitor for speaker circuit
15 uF, 10V

4.2 Current limiter circuit (2)

4.2.1 Transistor (2)
RCA SK3173 PNP Si Bipolar Transistor
\[ P_T=125W, I_e=-15\,\text{mA}, V_{cb0}=-90V, V_{ce0}=-80V, \]
\[ V_{eb0}=-5V, R_{fe}=70, \text{Case TO-3} \]

4.2.2 Diode (2)
1N4002

4.2.3 Zener Diode (2)
3.3V, 1W

4.2.4 Resistor (2)
280 Ohm, 2 Watt

4.2.5 Resistor (2)

4.2.6 Resistor (2)
0.5 Ohm, 12 Watt

4.2.7 Fuse (2)
1.6 Amp (?)

4.3 Switch Circuits

4.3.1 Capacitors (3)
0.1 uF disk capacitor

4.3.2 Pull-up Resistors (2)
1000 Ohm, 1/8 Watt

4.4 Terminal Strips

<table>
<thead>
<tr>
<th>Strip</th>
<th>#terminals</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>Rear Panel</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>Transformer Mount</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>Base Plate</td>
</tr>
<tr>
<td>D</td>
<td>12</td>
<td>&quot;</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>&quot;</td>
</tr>
<tr>
<td>F</td>
<td>4</td>
<td>&quot;</td>
</tr>
<tr>
<td>G</td>
<td>6</td>
<td>&quot;</td>
</tr>
<tr>
<td>H</td>
<td>5</td>
<td>&quot;</td>
</tr>
<tr>
<td>I</td>
<td>4</td>
<td>&quot;</td>
</tr>
<tr>
<td>J</td>
<td>4</td>
<td>Paper Advance Plate</td>
</tr>
<tr>
<td>K</td>
<td>4</td>
<td>Base Plate</td>
</tr>
<tr>
<td>L</td>
<td>6</td>
<td>Carriage</td>
</tr>
<tr>
<td>M</td>
<td>4</td>
<td>Computer Mounting Plate</td>
</tr>
</tbody>
</table>
5 Embedded Computer Programming
5.1 Computer Hardware
5.1.1 Input/Output
5.1.1.1 Serial Port
A UART chip is present on the computer card. This chip, with its support circuitry, drives an RS-232 port, which is the printer's means of connection to the host computer. Although both input and output are available, only input is used, as no data is sent to the host computer.

5.1.1.2 Parallel Ports
24 bits of parallel input and output are available. These are used to drive the electromechanical parts, and to read the switches on the printer.

5.1.2 Timers
Four programmable timer interrupts are available on the computer card. These are used whenever timing is needed.

5.1.3 Interrupts
Serial I/O and the timers cause interrupts of the processor.

5.1.4 Processor
Z-80A

5.1.5 Memory
1024 bytes of RAM
8192 bytes of ROM. The ROM chip is Erasable, programmable; 4 are required (2716 chips).
5.2 Software

The program burned into the read-only memory of the computer is responsible for all operation of the printer; there is no function not mediated by the computer. The software can conveniently be divided into two parts: "foreground" and "background."

The foreground subroutines are those which are driven by interrupts: they run independently of other processes, and tend to be those which demand precise timing. For example the programs which drive the solenoids or the speaker are driven by the timer interrupt.

The background subroutines are have a lower priority, running whenever all foreground routines have (for the time being) given up their claims to execution. The background process is responsible for the translation of ascii to Braille. It sends its output -- consisting of 40-cell buffers, each representing exactly one line to be printed -- to the foreground routine to be punched into the paper.

5.2.1 Background Process

Subroutine Function
transl read a character from the input queue, then call the appropriate translation subroutine to process it (transp, compnot, or gradel)

transp if char is control char, call control, else call pchar with low-order 6 bits

compnot if char is control, then call "control", else if char is space, add 1 to space count else print out the space buffer with either blanks or countable spaces, depending on how many there are.

Then look up the char in cntbl, and call pcharx with the resulting cell(s).

pcharx if at end of line, go to beginning of next line and print the overrun character. Then print the char passed to pcharx.

levell Store char in glc, then decide which kind of char it is: A-Z: b=0; a-z: b=1; 0-9: b=2; space or control char: b=4.

Then call statem with b and the char.

statem This is a finite state machine. Current state (0-5) is given by gls; for each state there are five possible action routines index from register b. These are labelled zl through zl8. These do such things as changing state, setting variable gll, and calling the subroutine "output"

output Looks up the character passed in outtbl or ltable, and sends the resulting cell(s) to pchar.

control call a subroutine depending on the value of the control character passed. Typical subroutines: plf for linefeed, pvt for vertical tab, pcr for carriage return, pdbl for bel, pnul does nothing.
plf  pass the address of the linefeed routine to swbuf

pcr  set column to -1 (which is the left margin)

pesc  read a char from the input queue. depending on its value, reset the variable which points to the current translation routine to either compnot, transp, or gradel.

pchar  increment column. If this makes it equal to the position of the right margin, call pcr and plf. Then pass the char to print (which was the parameter) to "write".

swbuf  mark the buffer just completed as "ready to print." then wait for the other buffer to become available, and set it as the one to write to. Then initialize the new write buffer.

write  If the current print position in the buffer marked for writing is empty, write the character there, and return. Else, if the corresponding print position (given in both cases by -column-) in the underscore line is empty, write it there. Else do nothing.
5.2.2 Foreground Processes

5.2.2.1 Reading from Serial Port

indata called by serial port interrupt; reads char from port, masks out the parity bit, and calls putql

putql puts character into the input queue

5.2.2.2 Polling the switches

intlize initializes state variables for switch polling and paper advance mechanism; starts up the interrupt to call readsw every 250 ticks (every 16 msec)

readsw debounces the switches by waiting 16 msec for the to settle. When a transition is detected and debounced, calls actnow

actnow checks linefeed button, formfeed button, mode switch. Calls makelf or makeff for linefeed or formfeed only if carriage is not in motion. If mode switch is switched, puts appropriate escape sequence into the input queue (ESC 5 or ESC 6)

5.2.2.3 Carriage and solenoid motion

fore Waits for buffer to be marked for punching. If first line is nonzero, calls punline to punch it. Then if underscore line is nonzero, generates a half line feed, and calls punline on the underscore line. Then calls whichever paper advance routine was specified in the buffer, and marks the buffer as ready to write to. Then waits for paper advance to finish, and starts over again.

punline if carriage is at right, calls "punl" to punch while going leftwards; if carriage is at left, calls "punr" to go rightwards.

punr initializes carcol to -1 (just before leftmost cell). Then call nextr to determine both the next 3-dot punch to make (one dot from cell N, one from cell N+5, and one from cell N+10), and the number of carriage steps needed to get to the right place to make it. Then:

REPEAT call movpun to move and punch; call nextr UNTIL nextr says there is nothing more to punch Then if paper=narrow, call carrmar to move to right margin; else (paper=wide) call carlmar to move to left margin.

punl similar to punr (calls nextr and movepun).

nextr Keep incrementing carcol and calling "read" until something is found; if nothing is found return a 0, else calculate number of steps to get there, and return both the bits to punch and the number of steps.
nextl  similar to nextr

movpun  call nsteps to move the distance passed as parameter; wait a short time; call pun to punch the solenoids with the appropriate values; wait some more.

nsteps  calculate destination, and store it for purposes of being able to recover if something misses. Then accelerate using the 100-step acceleration sequence (or as much as is needed of it), coast (if more than 200 steps total), and then reverse the acceleration sequence to decelerate.

pun  output the bits to the solenoids; wait a while; turn off the solenoids so they can retract.
TRANSFORMER MOUNT

(2 REQUIRED)

SHAFT SUPPORT

PAPER ADVANCE

RIGHT PULLEY MOUNTING PLATE

Note: The parts shown are on the other side.

The 8.05mm holes are for the other side.
Drill and tap for 8-32 screws, then mill a square-bottom 1/4" hole to depth of 0.10".

This square-bottom countersinking is to fit the head of an 8-32 cap screw.

Drill and tap for 8-32 screws.

Paper Guide Bar

Paper Advance Plate

Computer Support Plate

Material: 3/8" thick aluminum.
BASE PLATE LAYOUT
ASSEMBLY AND OPERATION

OF PAPER HOLD-DOWNS

PLATE MOUNT

PLATE BAR
Layout of Rear Cabinet Panel

- Ventilation slots
- Line cord connection
- Terminal strip on inside (A)
- Fuse
- Plug: RS-232 Input
- Fan (Blows out)
CARTRIDGE ASSEMBLY

When at bottom of travel:

1. Insert button to support submold plungers in cartridge bottom plate.

2. Insert button through guide plate.

3. Insert button through guide plate.

4. Insert button through guide plate.

5. Insert button through guide plate.
The bottom of the carriage.

Clamp the clamp bolt above the weigher. Cut off the wire, then.

Then twenty-pound weight to the other end of the wire. Then

Clamp on clamp bolt; then attack a

in the base plate. Clamp on clamp bolt, then down through

the carriage bottom plate, and down through the large holes,

red, down through the carriage clamp bolt, through the holes in the ends

of small hole in each carriage end-plate, down around the short base boss.

Then feed the wire around the pulleys on each end plate, and through the

wire

clamp screw

use a six-foot length of cable. Feed it through the hole in the

Installing the carriage cable: