UNIVERSAL FORMATTER OPTION

OVERVIEW

The Universal Formatter option allows you to create custom data output formats. Also, many of the 9100 Standard and SMART operating commands may be stored by the Universal Formatter as part of a set-up routine that will be activated at power up. Commands stored by the universal formatter override the switch settings.

Storing a setup routine is not difficult. Creating a data output format, conditional data output, and bit manipulation of the status characters assumes that the user is familiar with digital logic and boolean algebra. In either case, plan the format or setup routine on paper before you enter it from the host or menu.

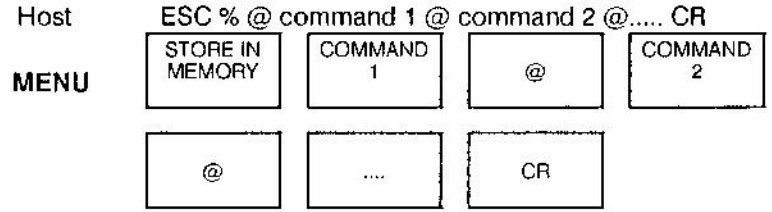
How to store commands is explained first, followed by the custom format commands.

STORING COMMANDS

All commands to be stored must be entered as part of a single string. A maximum of 100 characters of format and setup commands can be stored in the non-volatile memory, not counting the ESC % command prefix.

The ESC % @ command instructs the Universal formatter to store the next commands in memory. Several commands may be saved for automatic activation at power on. Within a string of commands. the @ character is used to concatenate the commands.

Enter the commands as a string, preceding each command with the @ character. The final command must be followed by a CR. The commands will be activated at the next power-up or after the reset command is sent to the tablet.



EXAMPLE

For example, assume you want the digitizer to power up in format 4, but with the resolution at 30 lines per millimeter, two significant digits to the right of the decimal point, increment line mode active, and increment size of 0.1 millimeter. Format 4 may be selected by the switch settings on the digitizing processor board. None of the other settings can be stored in the internal switches. They can be stored by the Universal Formatter as follows:

1. On paper, list the individual commands needed, without their command prefixes or the CR:

   a. Set resolution to 30 lpmm with two significant digits to the right of the decimal:

   Host J M 3Ø . 2

   Menu [SYSTEM FUNC 1] [M] [3] [Ø] [.] [2]

   b. Enable Increment Line mode:

   Host I U

   Menu [INCR] [U]

c. Set X Increment distance:
Host X 3 (at 3Ø lpmm, 3 lines is Ø.1 mm)
Menu [SET X INC] [3]

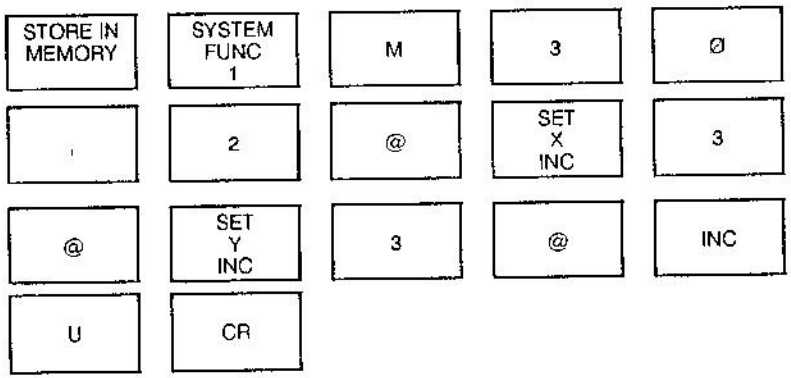
d. Set Y Increment distance:
Host Y 3
Menu [SET Y INC] [3]

2. From the host, send the command:

ESC % @ JM3Ø,2 @ X3 @ Y3 @ IU CR

The setup parameters will be activated at the next powerup, or as soon as the reset command, ESC % V R, is sent from the host.

3. From the menu, pick these blocks:



The setup parameters will be activated at the next powerup, or as soon as the [SYSTEM FUNC 2] [R] blocks are picked to reset the tablet.

CAUTION

DO NOT ENTER THE RESET COMMAND, @VR or [SYSTEM FUNC 2] [R], AS PART OF THE COMMAND STRING! IT WILL CREATE AN UNBREAKABLE RESET LOOP. THE DIGITIZER WILL NEED A SERVICE CALL TO BRING IT BACK INTO OPERATION.

Commands such as RELOCATE ORIGIN, which require that the operator digitize a point on the tablet surface, may not be stored. There is no way to complete the command within the setup routine. The tablet would probably lock up.

TESTING A COMMAND STRING

Entering the string WITHOUT THE FIRST @ SIGN will allow the string to immediately execute without being stored. If the string of commands does what you wanted it to do, re-enter it with the STORE IN MEMORY command.

The command string for the example could have been tested by entering:

ESC % JM3Ø,2 @ X3 @ Y3 @ IU CR

CORRECTING ERRORS

There is no provision for editing a command string once it has been entered in the Universal Formatter.

If you are entering the command string from a terminal, use the DEL or rubout key to erase errors before you store the command string with CR. If you discover the error after you have entered the command string, you must start over. Re-enter the entire string, including the ESC % command prefix. The old string will be overwritten by the new string.

From the menu, pick the [RUB OUT] block once for each wrong block you picked. If you have already ended the command string with [CR], you must re-enter the entire command string again.

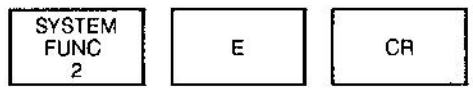
The tablet has no way to display the command string that has been stored.

CLEARING THE NON-VOLATILE MEMORY

The stored commands, and any format as well, may be cleared from memory with the following command

Host **ESC % V E CR**

MENU



CAUTION

This command cannot be stored. As soon as you reset the tablet, the memory will be erased.

No permanent damage to the memory will result.

CREATING A CUSTOM FORMAT

A format is entered by a host or menu command:

Host	**ESC % F {format command string} CR**
Menu	**[FORMAT][format command string] [CR]**

Where F is the ASCII character.

The format command string is made of one or more subordinate formatting commands. In the rest of this discussion the word "command" will refer to these subordinate commands. You must define the entire output format with these commands. The tablet will not transmit any data that is not defined.

Formatting commands are entered in the same order as the desired output. They may be separate by commas or spaces, but the command string, including any operating commands, is limited to 100 characters. Spaces and commas are included in the following examples for clarity; they are not required.

The examples with each formatting command are intended to clarify only that command. Examples of useful command strings, to produce the five 9100 output formats and others, begin on page 126.

Like a set-up routine, a format may be stored with the ESC % @ prefix, or the [STORE IN MEMORY] block. If you are storing a set-up routine too, the format command and its string of subordinate commands must be entered with the set-up commands.

NOTE

When you save a format in memory, the format will not become active until the tablet has been reset. **DO NOT** store the reset command.

The subordinate formatting commands are considered part of a single command for storage, and are not separated by @ signs:

enter this

ESC % @ F TA MA .... CR

not this

ESC % @ F @ TA @ MA.....CR

TYPES OF OUTPUT DATA

The Universal Formatter outputs the following data types:

1. Numeric Data
   a. X Data - The position of the transducer on the X axis, expressed as lines. The raw number is controlled by the resolution command, page 69.
   b. Y Data - The position of the transducer on the Y axis, expressed as lines. The raw number is under the control of the resolution command, page 69.
   c. Z Data - The height or volume factor. If the SMART option is installed, Z data (height) may be entered as a calculation constant and output with the X and Y data. The tablet does not measure height directly.
   d. K Count - The number of coordinate pairs transmitted since the last power-on or reset.The tablet counts up to 2^{24} then resets to zero and begins again with the next pair. You may also reset the counter with the ESC % V K command (see page 123).
2. ASCII Data

   ASCII characters inserted into the output data stream to transmit text.
3. Status Data

   The Status Data characters are transmitted when requested. They may be changed by the SET CHARACTER CONSTANTS command on page 52.
   a. Tablet status - A constant output.
   b. Mode - Indicates the operating mode.
   c. Cursor status - Indicates which button is pressed, or indicates that no buttons are pressed.
   d. Stylus status - Indicates whether the button is pressed or up.

DEFINING AN OUTPUT FORMAT

Formats are defined by listing the ASCII command for the data type, followed by the command for the output format or mathematical operation desired for that data type.

NUMERIC DATA

Numeric data may be output in integer, fixed point decimal, exponential or binary format. Any numeric data may be output in any format. A single data type may be output in more than one format. Different format types may be mixed in the command string.

NOTE

A tablet equipped with the SMART option uses the exponential format for numeric data output. It will not output binary format at any time. If the SMART commands are not being used, the tablet will output Integer or Fixed Point formats.

DATA CODE	DESCRIPTION
X	X axis position data
Y	Y axis position data
Z	Z data for output (SMART only)
K	Number of data pairs transmitted since last reset

FORMAT CODE	DESCRIPTION
I	Integer output
F	Fixed point decimal output
E	Exponential output
B or b	Binary output

The X, Y, Z and K data types follow the same formatting rules. To assign a format to a data type, enter the ASCII character for the data type followed by the ASCII code for the output format and the characters that define the exact output.

For example, to output the X-axis coordinates in binary, the command would be XB, or Xb, followed by the modifiers for binary data that specify number of bits, byte width and bits per byte. Specific examples of each output are given in the following pages.

INTEGER

Integer format outputs numeric data as a whole number. This command interacts with the resolution command, page 69, or the resolution settings in the internal switches.

The form of the command is Iw.d :

I	= ASCII I (uppercase i)
w	= maximum field width
.	= ASCII " . " to separate w from d
d	= displacement

The **field width** is the maximum number of characters, digits or polarity signs, to be output. Data with more than the maximum number of characters will overflow the field. The digitizer outputs asterisks to indicate an overflow.

The default display condition fills the field with leading spaces if the output has fewer than the maximum number of digits. Negative numbers are preceded by a minus sign immediately to the left of the number, which uses one of the field spaces. Positive numbers do not have a plus sign. The "Default Override" command, on page 106 , can change the default style, as shown in the last two examples.

The **displacement** interacts with the offset of the resolution command. The resolution command accepts the number of counts, as an integer, from the digitizer and moves the decimal point to the **left** according to the offset number entered as part of the resolution command. The integer format moves the decimal point back to the **right,** according to the value of the displacement.

If the resolution command's offset is greater than the integer command's displacement, the digits that remain at the left of the decimal point will be truncated.

If the resolution offset is smaller than the integer displacement, the output will have trailing zeros.

If no significant digits are to the left of the decimal point, the format outputs a zero.

EXAMPLES

The resolution is at 1000 LPI, with an offset of three. The X-axis position of the transducer is 10,583 lines away from the origin. The resolution command converts this to 10.583 inches. The examples below show the effect of various field widths and displacements on the final output.

COMMAND	OUTPUT	EXPLANATION
XI6.3	**SP1Ø583**	The displacement of three counteracted the offset of three. The five digit output leaves room for a leading space in the six-digit field.
XI6.Ø	**SPSPSPSP1Ø**	There was no displacement to counteract the offset. The output truncated all the digits to the right of the decimal.
XI4.1	**SP1Ø5**	The displacement of 1 allows the digitizer to output the first number to the right of the decimal. The other two are truncated.
XI4.3	*** * * ***	The displacement counteracts the offset, but the output of 10513 will not fit into a four-digit field width. The tablet outputs one * for each space of the field.
XI6.4	**1Ø583Ø**	The displacement is greater than the offset, resulting in a trailing zero.
S5 XI6.Ø	**+ØØØ1Ø**	The output is the same as the second example. The Default Override (see page 106) produces the leading zeros and a plus sign.

FIXED POINT

The fixed point format expresses the output as a decimal number. Like Integer, Fixed Point is affected by the Resolution command and internal switch settings.

The form of the command is Fw.d where:

F	= ASCII F
w	= maximum field width
.	= ASCII " . "
d	= fixed point displacement

The **field width** is the maximum number of characters in the output, just as in the integer command. The decimal point is always present and takes up one of the character spaces.

The number of characters to the right of the decimal point is controlled by the Resolution offset. The fixed point **displacement** controls how many of those characters will be output.

The value entered for the displacement may range from zero to the field width minus one. If the displacement equals the offset, the number will be output unchanged. If the displacement is smaller than the offset, the excess characters will be truncated. If the displacement is greater than the offset, the output will have trailing zeros.

The polarity signs and leading zeros or spaces follow the same default rules as the integer format. They may be altered by the Default Override command (page 106).

EXAMPLES

The resolution is at 1000 LPI, with an offset of three. The Y-axis position of the transducer is 15,725 lines from the origin. The resolution command converts this to 15,725 inches. The examples show the effect of field width and displacement.

COMMAND	OUTPUT	EXPLANATION
YF6.3	15.725	The displacement equals the offset. The output is not altered.
YF4.3	* * * *	The field width is too small to display the number to the right of the decimal point, so the field overflows.
YF7.4	15.7250	The displacement is greater than the offset. The output contains trailing zeros to fill up the alloted field width.
YF6.2	SP15.72	The displacement of two truncates one digit to the right of the decimal. The field width of six leaves room for a leading SP.
S4 YF6.2	+15.72	The output is the same as the previous example, but the Default Override was used to produce the leading plus sign.

DEFAULT OVERRIDE

The default output of the Integer and Fixed point data formats has leading spaces to fill unused field width, does not show a plus sign, and places the minus sign immediately to the left of any negative number. This can be changed by placing a modifying command into the command string before the data type to be modified.

Data type commands follow the latest applicable modifier in the string.

When modifier commands are used **anywhere** in the command string any numeric data to be output in the unmodified (default) manner must be preceded by SØ, the modifier calling out the default conditions.

If all numeric data is to be modified in the same manner, a single modifier may be placed at the beginning of the command string.

The default override command is Sn where:

S = ASCII "S"

n = an ASCII numeral from Ø to 5, modifying the output as explained below.

n	LEADING CHARACTER	MINUS SIGN	PLUS SIGN	EXAMPLE
Ø	spaces	after spaces	none	SP – XXXX or SPSPXXXX
1	zeros	before zeros	none	– ØXXXX or ØØXXXX
2	spaces	after spaces	after spaces	SP – XXXX or SP + XXXX
3	not used			
4	spaces	before spaces	before spaces	– SPXXXX or + SPXXXX
5	zeros	before zeros	before zeros	– ØXXXX or + ØXXXX

EXAMPLE

The resolution is at 1000 LPI, with an offset of three. The X-axis position of the transducer is 12,723 lines away from the origin. The resolution command converts this to 12.723 inches. The examples below show the effect of the default override on the output from an integer command. Fixed point would be modified in a similar fashion.

Because the presence and position of the polarity sign is also altered, each output is shown for positive and negative X values.

UNMODIFIED (DEFAULT) OUTPUT

XI7.3	SPSP12723 SP –12723	The five digit output leaves room for leading spaces in the seven-digit field.

MODIFIED OUTPUTS

COMMAND	OUTPUT	EXPLANATION
SØ XI7.3	SPSP12723 SP– 12723	The SØ is the command to produce the default output.
S1XI7.3	ØØ12723 Ø–12723	S1 produces leading zeros and places the minus sign behind any zeros.
S2XI7.3	SP+12723 SP–12723	Plus and minus signs are behind leading spaces.
S4XI7.3	+SP12723 –SP12723	Plus and minus signs precede the leading spaces.
S5XI7.3	–Ø12723 +Ø12723	Plus and minus signs precede the leading zeros.

EXPONENTIAL OUTPUT

The exponential format outputs the data as a signed decimal fraction and an exponent according to the ANSI FORTRAN method of notating exponents.

For example, +14.863 would be output as +.14863E+Ø2; -2.250 would be output as -.12250E+Ø1. The sign of the number always precedes the decimal point.

The form of the command is Ew.d where:

E	= ASCII "E"
w	= field width
.	= ASCII "."
d	= mantissa width

The **field width** is the maximum number of characters in the output. The field width must exceed the desired mantissa width by at least six. The output must always include the sign of the mantissa, the leading decimal point, the ASCII "E", the sign of the exponent and the two digit exponent. In addition, at least one digit must be present in the mantissa for a minimum field width of seven.

If there isn't enough room in the field for the specified mantissa width and the six mandatory characters, the field will fill with asterisks to indicate overflow.

If the mantissa width is smaller than the available space in the field width, the field will contain leading zeros.

The raw numbers to be formatted may contain any number of significant digits. If the mantissa allows for more digits than are significant, the mantissa will contain trailing zeros. If the number has more significant digits than the mantissa allows, the excess digits will be truncated.

The positive or negative output may range from .922E+19 to .542E-19. Numbers smaller than this are expressed as zero, output as .ØØØE-2Ø.

EXAMPLES

The resolution is at 1000 LPI, with an offset of three. The X-axis position of the transducer is 8,321 lines from the origin. The resolution command converts this to 8.321. The examples show the effect of field width and displacement.

COMMAND	OUTPUT	EXPLANATION
XE1Ø.4	+.8321E+Ø2	The output fills the field completely.
XE11.5	+.8321ØE+Ø2	There aren't enough significant digits to fill the specified mantissa width, resulting in a trailing zero.
XE1Ø.5	* * * * * * * * * *	The field width is too small for a 5-digit mantissa and the six mandatory characters. It overflows.
XE7.2	+.83E+Ø2	The output truncates the least significant digits to fit the output into the specified mantissa width of 2.
XE11.4	SP+.8321E+Ø2	The specified mantissa width plus the required characters do not fill the field width, resulting in a leading SP.

BINARY OUTPUT COMMANDS

The Binary output is only available on 9100 Standard systems with the Universal Formatter option. Tablets with the SMART option installed will not produce Binary output.

The form of the command is Bw.d or bw.d where:

B	= ASCII "B" or "b"
w	= bit count (1 to 24)
.	= ASCII "."
d	= data bit width (1-8)

The **bit count** is the total number of bits to be used in formatting the data. The data is transmitted in two's complement to preserve the polarity. If the bit count is too small to format the data, the least significant bits will be truncated.

The largest possible number from the largest tablet at the highest resolution needs 20 bits for transmission. (A 60" X-axis, at 2450 LPI, has 147000 lines on the X-axis.)

The **data bit width** is the number of data bits assigned to each 8-bit byte of the output. The data bits are loaded into the least significant side of each byte, filling the leftover bits with zeros. The formatter will distribute the data bits evenly across the bytes, if possible. A bit count of 16 and data byte width of four would produce four bytes with four data bits in each. A bit count of 16 and a data byte width of five would produce four bytes, three with five data bits and one with one data bit.

If the command was **Bw.d**, the byte containing the most significant data bit is transmitted first. The similar command **bw.d** produces binary output with the byte containing the least significant bit transmitted first.

CAUTION

If you specify a data byte width of eight, the DATA BITS setting of the switches on the Communication Interface board must also be set to eight or the board will truncate the most significant bit of each byte.

EXAMPLES

Command: XB18.6

			MSB					
Byte one	Ø	Ø	X17	X16	X15	X14	X13	X12
Byte two	Ø	Ø	X11	X1Ø	X9	X8	X7	X6
Byte three	Ø	Ø	X5	X4	X3	X2	X1	X0
								LSB

Command: b18.6

								LSB
Byte one	Ø	Ø	X5	X4	X3	X2	X1	X0
Byte two	Ø	Ø	X11	X1Ø	X9	X8	X7	X6
Byte three	Ø	Ø	X17	X16	X15	X14	X13	X12
			MSB					

The outputs are identical, except that the order of the bytes is inverted. Both outputs have 18 data bits divided into bytes with 6 data bits per byte.

Command YB12.5

							MSB	
Byte one	Ø	Ø	Ø	Ø	Ø	Ø	Y11	Y10
Byte two	Ø	Ø	Ø	Y9	Y8	Y7	Y6	Y5
Byte three	Ø	Ø	Ø	Y4	Y3	Y2	Y1	Y0
								LSB

Command: Yb12.5

								LSB
Byte one	Ø	Ø	Ø	Y4	Y3	Y2	Y1	Y0
Byte two	Ø	Ø	Ø	Y9	Y8	Y7	Y6	Y5
Byte three	Ø	Ø	Ø	Ø	Ø	Ø	Y11	Y10
							MSB	

Twelve data bits, and five data bits per byte. The number of bits wouldn't distribute evenly, so the formatter filled the remainder of the byte with zeros.

BINARY BIAS

The Binary Bias command adds an offset value to the binary output of X, Y, Z, or K data. The output byte becomes the sum of the original data byte and the offset value. The bias is added to each byte of the data, regardless of the number of bytes.

The bias is added to the whole byte, regardless of the number of data bits designated for data within the byte. If the sum of the bias plus the data byte exceeds 255, the resulting byte will contain the value of the sum minus 256. No carry or borrow is passed between bytes.

The form of the command is Bxx where:

B	= ASCII B
xx	= the value of the bias (offset) as a two-digit hexadecimal number

Like the default override, the Binary Bias command must precede the data type command it is to modify; if you are outputting a mix of biased and unbiased data, the unbiased data must be given a bias of ØØ Hex.

EXAMPLE

You want to output a 1 as the most significant bit in the Y-axis data bytes as a flag to a program. You want the X-axis data to be unbiased. You want both outputs to contain 18 data bits in three bytes.

The command would be:

BØØ XB18.6 B8Ø YB18.6

X-data:

			MSB					
Byte one	Ø	Ø	X17	X16	X15	X14	X13	X12
Byte two	Ø	Ø	X11	X10	X9	X8	X7	X6
Byte three	Ø	Ø	X5	X4	X3	X2	X1	X0 LSB

Y data before bias:

			MSB					
Byte one	Ø	Ø	Y17	Y16	Y15	Y14	Y13	Y12
Byte two	Ø	Ø	Y11	Y10	Y9	Y8	Y7	Y6
Byte three	Ø	Ø	Y5	Y4	Y3	Y2	Y1	Y0 LSB
Bias value of 80H added to each byte	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø

OUTPUT is biased Y data :

			MSB					
Byte one	1	Ø	Y17	Y16	Y15	Y14	Y13	Y12
Byte two	1	Ø	Y11	Y10	Y9	Y8	Y7	Y6
Byte three	1	Ø	Y5	Y4	Y3	Y2	Y1	Y0 LSB

OUTPUTTING CHARACTER STRINGS

ASCII characters may be inserted into the output data stream in three ways.

1. Most printing characters may be enclosed in single or double quotes. "CHARACTERS" or 'CHARACTERS' will output CHARACTERS in the data stream.
2. If one of the characters you want to output is a " ' " or ' " ', the digitizer will interpret the quote character as a string delimiter, not a character. To output a string that includes a quote, you may use the command:

   n H character string

   n is the number of characters in the string that follows the "H"

   H is an ASCII H
3. The ASCII control characters cannot be entered by the above methods. If you try to insert a Carriage Return, the digitizer will interpret it as the end of the command string.

   Insert control characters one at a time with the command:

   N x x

   N is an ASCII N.

   x x is the hexadecimal value of the control character to be output. For example, ØDH is the value of a carriage return. NØD will output a carriage return in the data string. N22 will produce a double quote ("). The ASCII chart is on page 167 of this manual.

   NOTE

   Character values between 8Ø and FF Hex (128 to 255 decimal) may repeat the standard ASCII character set, or may be used for special characters, depending on the host. Consult the host's technical manuals.

   Inserting control characters may have unexpected effects. Control Z (1A Hex) is used by many programs to indicate the end of a data file. Control C (Ø3 Hex) is used by others as an "exit this program" command.

   Remember that spaces must be counted as characters. Up to 5Ø characters may be output in the string.

EXAMPLE

Although you are using exponential format for the X and Y data, you want the digitizer to output an easy-to-read count of the number of data points that have been taken. Use the following format commands:

XE12.5 YE12.5 NØD
KI5.Ø " DATA POINTS HAVE BEEN TAKEN" NØD

The resulting output, after 3,573 data points would be:

3573 DATA POINTS HAVE BEEN TAKEN

The command KI5.Ø produces the integer output.

The message will be on a separate line because of the carriage returns (the command NØD) on either side of the K-count . The quotes will not be output.

To include the quotes in the above example, use the "nH" command:

3ØH" DATA POINTS HAVE BEEN TAKEN"

MIXING OUTPUT FORMATS

It is possible to mix output formats. Each data type will be modified by any modifiers which apply to its type and be unaffected by a modifier for another type.

For example, the output resulting from the command string

S5 BA0 XB18.6 YI6.Ø Kb1Ø.5

would be the X-coordinate, as a three-byte binary output, biased by A0H and transmitted most significant byte first. Then the tablet would transmit an integer output with leading zeros and polarity signs for the Y coordinate. Finally a two-byte binary output, biased by AØH and transmitted least significant byte first, would be transmitted for the K count. Each byte of the K-count would have 5 data bits.

STATUS DATA

The status data (Tablet, Mode, Cursor, or Pen) may be output as ASCII, binary, complementary binary or hexadecimal. The desired output format is indicated by a character placed after the character for the status data. For example, to output the cursor status character in ASCII, the command would be CA. To output the Mode Status in Hexadecimal, the command would be MH.

DATA CODE	DESCRIPTION
T	Tablet status, always an "A". Included for compatibility with older models.
M	Operating Mode - denotes current operating mode.
C	Cursor status - tells which button on the cursor is currently depressed.
P	Pen status - tells if the stylus tip is depressed or up.

OUTPUT FORMAT	COMMAND
ASCII	A
BINARY	B
COMPLEMENTARY	C (one's complemented binary)
HEXADECIMAL	H

The ASCII format outputs the default status data character or the character specified by the SET CHARACTER CONSTANTS command on page 52.

The Hexadecimal output produces the two-character hexadecimal output shown in the table.

Binary output of status data produces the binary equivalent of the Hexadecimal output.

The complemented binary output produces the one's complement of the binary output.

The ASCII default characters and the hexadecimal outputs are listed in the following table.

CHARACTER CONSTANTS

POSITION	ASCII	HEX	DESCRIPTION
1	A	ØØ	Tablet Status
2	A	ØØ	Answer/Display Flag (SMART units)
3	I	Ø1	Mode Status - Increment Modes
4	P	Ø2	Mode Status - Point Mode
5	U	Ø3	Mode Status - Line Mode
6	R	Ø4	Mode Status - Run Mode
7	T	Ø5	Mode Status - Track Mode
8	M	Ø6	Menu Active - Transducer on Menu
9	X	Ø7	Transducer Out of Proximity
1Ø	U	ØØ	Pen Status - Pen Up
11	D	FF	Pen Status - Pen Down
12	U	FF	Cursor Status - All Buttons Up
13	Ø	ØØ	Button Pressed - Button Ø
14	1	Ø1	Button Pressed - Button 1
15	2	Ø2	Button Pressed - Button 2
16	3	Ø3	Button Pressed - Button 3
17	4	Ø4	Button Pressed - Button 4
18	5	Ø5	Button Pressed - Button 5
19	6	Ø6	Button Pressed - Button 6
2Ø	7	Ø7	Button Pressed - Button 7
21	8	Ø8	Button Pressed - Button 8
22	9	Ø9	Button Pressed - Button 9
23	A	ØA	Button Pressed - Button A
24	B	ØB	Button Pressed - Button B
25	C	ØC	Button Pressed - Button C
26	D	ØD	Button Pressed - Button D
27	E	ØE	Button Pressed - Button E
28	F	ØF	Button Pressed - Button F
29-44	G-V		RESERVED

BIT MANIPULATION OF STATUS CHARACTERS

A status character may be ADDed, ORed, ANDed or XORed with a two-digit hexadecimal constant between ØØ and FF. The bits of the status byte may also be rotated to the left or right from one to seven places. ASCII status characters may be manipulated; the output will be the ASCII equivalent of the results of the bit manipulation.

Multiple manipulations may be performed on a status character. The formatter processes compound bit manipulation commands from left to right, performing each manipulation in the order it was entered.

The status characters may be changed, with the SET CHARACTER CONSTANTS command, page 52. The bit manipulation will be performed on the new status character.

MANIPULATION COMMANDS

ADD	+nn or −nn (subtracting by adding a negative number)
OR	^nn
XOR	~nn
AND	*nn

Where +, −, ^, ~, and * are the ASCII characters and nn is the hexadecimal constant between ØØ and FF.

If the sum of the ADD operation is greater than 255, the resulting output will be the sum minus 256.

Left rotation	<n
Right rotation	>n

Where < and > are the ASCII characters and n is an ASCII character, from one to seven, representing the number of places the bits are to shift. Bits pushed out of the end of the byte come around to rest in the other end of the byte.

EXAMPLES

CB +Ø1	adds Ø1H to the binary output of the cursor. ØØØ1Ø11Ø becomes ØØØ1Ø111
CA +Ø1	adds Ø1H to the ASCII output of the cursor ASCII "D" (44H) becomes "E"(45H)
MB ^33	Boolean ORs the mode status character with 33H Ø1Ø1ØØ1Ø (R, for run) becomes Ø111ØØ11
CH ~10	Boolean XORs the cursor status character with 10H FFH becomes EFH
CA *Ø1	Boolean ANDs the cursor output with Ø1H ASCII "D" (44H) becomes NUL (ØØH)
CB<3	Rotates the cursor status character output three places to the left. The bits move around into the least significant side of the byte.

Original output

Ø	Ø	1	1	Ø	1	1	Ø

After rotation

1	Ø	1	1	Ø	Ø	Ø	1

COMPOUND MANIPULATIONS

CB<2^8Ø	This will rotate the binary cursor output two places to the left, then OR it with 8ØH before outputting the cursor data.

Original cursor status byte:

Ø	Ø	Ø	C4	C3	C2	C1	CØ

Rotated cursor status byte, to be ORed with 8ØH:

	Ø	C4	C3	C2	C1	CØ	Ø	Ø
8ØH	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø

Final output:

1	C4	C3	C2	C1	CØ	Ø	Ø

3

INDIRECT MANIPULATION

The status data may be ORed with the character occupying the designated position of the data stream. The results of this OR operation are output in the designated position instead of the original character. This is used in conjunction with the rotate command to combine, for example, the cursor status character with one of the bytes of X or Y data to produce more compact output. The status data character is not output by this command, it is only used as part of the OR expression.

The Ln command may be invoked only after the character in the data stream to be ORed has been formatted.

The command is Ln where:

L	the ASCII character L
n	An ASCII numeral indicating which position in the data stream is occupied by the character to be ORed with the designated status character.

EXAMPLES

Command YB12.5 CB<2 L1

Original Y data bytes

							MSB	
Byte one	Ø	Ø	Ø	Ø	Ø	Ø	Y11	Y10
Byte two	Ø	Ø	Ø	Y9	Y8	Y7	Y6	Y5
Byte three	Ø	Ø	Ø	Y4	Y3	Y2	Y1	Y0 LSB

Original cursor status byte

	Ø	Ø	Ø	C4	C3	C2	C1	CØ

Rotated cursor status byte, to be ORed with the first position in the data stream, which is the first byte of the Y data.

	Ø	C4	C3	C2	C1	CØ	Ø	Ø

Output, with cursor information and Y data combined in one byte.

							MSB	
Byte one	Ø	C4	C3	C2	C1	CØ	Y11	Y10
Byte two	Ø	Ø	Ø	Y9	Y8	Y7	Y6	Y5
Byte three	Ø	Ø	Ø	Y4	Y3	Y2	Y1	Y0 LSB

CONDITIONAL OUTPUTS

The conditional commands allow the output to change depending on the condition of the data and status characters.

The conditional expression is always placed in brackets. It contains the formatting commands that will be executed if the conditions are met. If the conditions are not met, the formatter skips the bracketed commands.

The formatter may be instructed to skip the commands following the brackets if the conditions are met. Placing the ASCII characters QF (quit format) inside the brackets instructs the formatter to skip the rest of the format commands if the conditions are met. The QF command may be placed after commands inside the brackets if you want them to be carried out before the format is terminated.

The formatter will test for equality or inequality. The equality test is met if the value of the test expression is equal to the specified hex value. The inequality test is met if the value of the test expression equals any number other than the specified hex value.

The test is a single-character comparison. If a multiple- character ASCII value or character string is presented, the value of the **last character** is used. If multiple-byte binary data is presented, the **last byte** of the binary expression will be used. If a manipulated status character is presented, the manipulation will be performed **before** the comparison is made.

The conditional commands take the general format:

Equality: Test Expression = xx {conditional commands}
Inequality: Test Expression # xx {conditional commands}

Where Test Expression is the character string presented for comparison.

=	Is the ASCII character for equality
#	is the ASCII character for inequality
xx	is the two-digit hexadecimal value to compare to the test expression
{ and }	are ASCII characters enclosing the commands to be executed if the test conditions are met

EXAMPLE

You want the K count and a message to be output each time you press cursor button 9. Include the following command into the string of formatting commands.

CA=39 {KI5.Ø "DATA POINTS HAVE BEEN TAKEN"}

CA puts the cursor output into ASCII, = checks for the equality, and 39 is the hexadecimal equivalent of an ASCII 9. When the cursor status character equals 9, the last character of the Hexadecimal number, the tablet will output the information in the brackets.

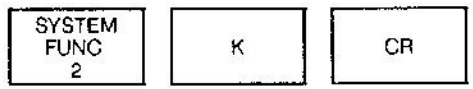
RESETTING THE K-COUNTER

The K-counter begins at one each time the tablet is powered up or receives the tablet reset command. The K-counter counts each data point transmitted, up to 2^{24}.

To reset the counter, use the following command:

Host **ESC % V K CR**

MENU



3

REPEATING EXPRESSIONS

Any piece of a format may be repeated. The repeating expression may be any legal data type, status characters, ASCII text or conditional expressions. Only one repeating expression is allowed in the format.

The ASCII characters QR, for quit repeat, may be inserted in a conditional expression within a repeating expression to allow the repeating expression to be cancelled.

The K counter is not advanced for each repetition; it only advances once for each formatting command. If X or Y data is within the repeating section of the format, the tablet will take a new position point for each repetition instead of outputting the same data over and over.

The command has the general format:

Rn (repeating expression)

Where:

R	Is the ASCII character
n	Is the number of times to repeat, from 1 to 255
(	ASCII character
repeating expression	The format commands to be repeated
)	ASCII character

NOTE

Disable the linefeed before you begin a repeating format. The results of repeating formats are not predictable if the linefeed is active.

EXAMPLE

Assume the resolution is at 1000 LPI, with the offset of three, giving output in whole and thousandths of inches.

Command:

KI3 R4 (S1"X" XI2.Ø "Y" YI2.Ø "|") "END" NØD

Explanation:

KI3	The K-count, as an integer
R4 (	Repeat the next part of the output four times.
S1	Places leading zeros into the output
"X"	Output the letter X
XI2.Ø	X output as integer, two places, offset of zero. This will give output in whole inches only.
"Y"	Output the letter Y
YI5.Ø	Y output as integer, two places, offset of zero, as in X.
"|"	Outputs | to mark the division between data points.
)	end of repeating section
"END"	Output the word END
NØD	Quoting a Carriage return into the output

Assume the first four data points you took were (4,5) (3,7) (6,12) and (8,2).

Output:

DATA POINTS	OUTPUT ON CRT OR TO HOST
4,5	ØØ1XØ4YØ5|
The output for the next data point is added to the string.	
3,7	ØØ1XØ4YØ5|XØ3YØ7|
6,12	ØØ1XØ4YØ5|XØ3YØ7|XØ6Y12|
8,2	ØØ1XØ4YØ5|XØ3YØ7|XØ6Y12|XØ8YØ2|END CR

Notice that the K count (the first three digits) doesn't increase with each data point. It will increase with the fifth point as the tablet starts over with the format.

EXAMPLES OF USEFUL OUTPUT FORMATS

To clarify the development of an entire output format, analyse the following examples. The spaces in the outputs and formatter commands are for clarity only. An actual space would be represented by SP.

Assuming that the resolution is set to 1000 LPI, with a default offset of 3, by the internal switches, the Universal Formatter commands would be as follows. If the resolution is different, adjust the X and Y output commands.

ASCII OUTPUT FORMATS

9100 FORMAT 1

Output	T M C XXXXX YYYYY CR
Command	TA MA CA XI5.3 YI5.3 NØD
TA	outputs the Tablet Status (always an A) in ASCII
MA	outputs the Mode Status Character in ASCII
CA	outputs the Cursor information in ASCII
XI5.3	outputs X data as a five-digit integer, with the ".3" counteracting the offset from the resolution setting of the switches.
YI5.3	Y data output
NØD	inserts a carriage return at the end of the data string

9100 FORMAT 2

Output	XXXXX, YYYYY, T M C CR
Command	XI5.3 "," YI5.3 "," TA MA CA NØD

This is simllar to Format 1, except for the inserted commas "," after the X and Y data.

9100 FORMAT 3

Output	C P XXXXX YYYYY CR
Command	CA PA XI5.3 YI5.3 NØD

Again, this is similar to Format 1, except for the Cursor and Pen Status characters.

9100 FORMAT 4

Output	SP XX.XXX, SP YY.YYY, T M C CR
Command	XF7.3 "," YF7.3 "," TA MA CA NØD
XF7.3	gives an output field width of 7, with three of the digits that are to the right of the decimal (as controlled by the Resolution command) to be output.
","	inserts the commas

NOTE

The SP does not have to be inserted. The default outputs for Fixed Point will insert a space if the data doesn't fill the field width. The largest number possible with this format and resolution has six digits. The field width is seven, which results in a leading space for the X and X-data. You could use N2Ø XF6.3 "," N2Ø YF6.3 to get the same results. N2Ø would insert a space by inserting the hexadecimal value of an ASCII space into the output.

BINARY OUTPUT

EXAMPLE ONE

Binary outputs frequently use the bit manipulation commands of the status characters to compress the output. In this example, instead of separate bytes for the cursor and proximity information, they are combined into one byte. The output is the same as a CalComp 2000 binary. The 2000's cursor output is always one greater than the number of the button pushed – pressing button "2" produces an output of binary "3" – while the normal 9100 cursor output equals the button pushed. Adding Ø1H is easier than redefining the cursor output.

CALCOMP 2000 BINARYOUTPUT:

BIT	7	6	5	4	3	2	1	Ø
BYTE 1	Ø	1	C3	C2	C1	CØ	Ø	Ø
BYTE 2	Ø	Ø	X5	X4	X3	X2	X1	XØ
BYTE 3	Ø	Ø	X11	X1Ø	X9	X8	X7	X6
BYTE 4	Ø	Ø	Y5	Y4	Y3	Y2	Y1	YØ
BYTE 5	Ø	Ø	Y11	Y1Ø	Y9	Y8	Y7	Y6

Command	CB + Ø1 ^1Ø<2 Xb12.6 Yb12.6
CB	cursor status character, in binary
+Ø1	ADD Ø1H to the cursor character
^1Ø	OR it with 1ØH (this produces the 1 in bit 6 of byte 1)
<2	shifts the bits of the first byte to the left 2 spaces
Xb12.6	X-data output in 12 data bits, six bits per byte, with the LEAST significant byte first
Yb12.6	Y-data output in 12 data bits, six bits per byte, with the LEAST significant byte first

EXAMPLE TWO

In this commonly used high-resolution binary output, the cursor bits are combined with part of the X data bits to make up the first byte. The cursor information is first manipulated, then ORed with the X data.

CALCOMP 9100 OUTPUT 5

BIT	7	6	5	4	3	2	1	Ø
BYTE 1	1	C4	C3	C2	C1	CØ	X15	X14
BYTE 2	Ø	X13	X12	X11	X1Ø	X9	X8	X7
BYTE 3	Ø	X6	X5	X4	X3	X2	X1	XØ
BYTE 4	Ø	Ø	Ø	Ø	Ø	Ø	Y15	Y14
BYTE 5	Ø	Y13	Y12	Y11	Y1Ø	Y9	Y8	Y7
BYTE 6	Ø	Y6	Y5	Y4	Y3	Y2	Y1	YØ

Command:

XB16.7 YB16.7 CB=FF{*ØØ ^2Ø <2 L1 QF} ^3Ø <2 L1

XB16.7	16 bits for X-data, with 7 data bits per byte
YB16.7	16 bits for Y-data, with 7 data bits per byte
CB=	Conditional statement, comparing cursor output with FF, which is produced when no buttons are pushed.
{	If no buttons are pressed, the tablet will output the following conditional format:
*ØØ	AND FFH with ØØH, producing ØØH
^2Ø	OR it with 2ØH, producing 2ØH
<2	Rotate it to the left two places, which places a 1 into bit 7 of byte one, for the leading zero.
L1	ORs it with the first character in the data stream, which has the most significant bits of the X data
QF	Quits the format
}	End of conditional output
^3Ø	IF any button has been pressed, the tablet skips to this, ORing the cursor output with 3Ø
<2	Rotates to the left 2 places, which places a 1 into bit seven as before, and also places a 1 into C4, the leading cursor bit (a leading 1 is in every cursor output of this format)
L1	ORs the results of ^3Ø <2 with the first character in the output, combining the cursor and X data.

THIS PAGE IS BLANK