Diagraph Series 2 Programmer's Manual

5700-329
Revision A

Diagraph•Series 2•Equipment•is

SERIES 2 PROGRAMMER'S MANUAL
TABLE OF CONTENTS

INTRODUCTION

The Diagraph *Series 2 Programmer's Manual* (5700-329) is a companion document to the Diagraph *Series 2 User's Manual* (5700-606). Together, they can guide an experienced programmer through the configuration of and successful printing with a Series 2 Ink Jet system. Specifically, this Programmer's Manual describes each host command that controls, enables reports, establishes communication and queries the Series 2 controller.

Intended Audience

This programmer's guide is intended for programmers who understand the basics of PC technology and who are familiar with industrial ink jet printing. The commands described in this book are unique to the Series 2 controller and the choice of communication protocols—RS-232, Allen-Bradley PLC or ARCnet—belongs to the user.

Document Structure

This *Introduction* gives an overview of this programmer's book and the conventions and provisions that cover its publication.

Chapter 1 identifies the Series 2 components required to program and print; the factory system defaults; and an introduction to the host commands with tips and examples.

Chapter 2 explains the variables of character formation, font characteristics and the parameters of message building.

Chapter 3 catalogues the complete command set with each command's full text explanation, format requirements, return data, parameters and examples.

Chapter 4 describes Series 2 autocodes, codes that allow you to add the date, time and other variable information to a label message.

Chapter 5 is a chart of the error codes (with explanations) that can appear when you program labels for the Series 2 controller.

Conventions

This book contains conventions that appear in all Diagraph publications.

Convention	*Description*
<CR>	ASCII ^M character
<ESC>	ASCII ^[character
<ESC>SLNS,200<CR>	Bolded text identifies an entry exactly as it is to be entered.
ENTER	Underlined words indicate keys on a computer keyboard.
Programming Tip	A bit of information that will facilitate successful Series 2 programming
SOH	ASCII ^A character
STX	ASCII ^B character

Disclaimer of Warranties

The author and the publisher have taken due care in preparing this manual. The author and the publisher make no warranties either expressed or implied in regard to the contents of this manual. The author and the publisher do not warrant that the software commands described herein will meet your specific requirements or that they will be error free. In no event shall the author or the publisher be liable for any damages, incidental or consequential, arising out of the use or performance of these software commands with your printing system.

CHAPTER 1
SYSTEM REQUIREMENTS

To communicate with the Diagraph Series 2 Printhead Controller you will need a computer (8088 or above) and Procomm® software or other software that can direct the computer as a dumb terminal. Cabling requirements vary with the kind of communication: serial, PLC block transfer or ARCnet.

RS-232

The controller can be accommodate either a modem or null modem cable.

1. Connect your PC to the controller (Serial Port 1) with a serial cable.
2. Select the appropriate serial port with your PC software.
3. Set the following parameters with your PC Software

Item	Setting	Item	Setting
Terminal Emulation	VT-100	Stop Bit	1
Baud Rate	9600	Parity	No Parity
Data Bits	8	Protocol	ASCII
Stop Bit	1	Duplex	Full or half

4. Turn the controller's power switch ON and wait 45 seconds for the controller to initialize.

   Press ENTER and the controller will respond with QERR,34,0 for "unknown command." This is your signal that communication has been established.

5. If no communication occurs, check your setup and swap a null modem for a modem cable or vice versa.

   Another option is to change the jumpers on the controller board. Switches are set for factory defaults when shipped.

   Controller RS-232 Defaults

   9600 baud — no parity
   8 data bits — handshaking is disabled
   1 stop bit

PLC with Remote I/O

An Allen-Bradley PLC can communicate with the Series 2 Controller by block transfer. You can utilize the entire PLC transmission length of 64 words by specifying a sequence of Host Commands. Do not exceed the maximum transmission size of 64 words. See Appendix I in the *Series 2 User's Manual* (5700-329) for complete setup instructions and a sample ladder logic program.

ARCnet

Installation instructions are in Chapter 2, Section 7.1 in the *Series 2 User's Manual* (5700-329)

ARCnet Address Switch Settings for Series 2

ARCnet Node Address	1	2	3	4	5	6	7	8
1	**OFF**	ON	ON	ON	ON	ON	ON	ON
2	ON	**OFF**	ON	ON	ON	ON	ON	ON
3	**OFF**	**OFF**	ON	ON	ON	ON	ON	ON
4	ON	ON	OFF	ON	ON	ON	ON	ON
5	**OFF**	ON	**OFF**	ON	ON	ON	ON	ON
6	ON	**OFF**	**OFF**	ON	ON	ON	ON	ON
7	**OFF**	**OFF**	**OFF**	ON	ON	ON	ON	ON
8	ON	ON	ON	**OFF**	ON	ON	ON	ON
9	**OFF**	ON	ON	**OFF**	ON	ON	ON	ON
10	ON	**OFF**	ON	**OFF**	ON	ON	ON	ON

Note that the node address switchblock is calculated in binary radix.

Series 2 Defaults

The table below shows the default settings for the Diagraph Series 2 Ink Jet System.

COMMAND	FACTORY DEFAULT
Sequence Count	value = 0 rollover = 999999999
Missing box count	0 (NOTE: You cannot check missing box count)
Shift count	0
User count 1 and 2	Step = 1 increment Rollover = 999999999 Value = 0
Date	Current Date
Time	Current Time
Speed Encoder Internal	Per site survey Resolution = 1000 Line speed = per site survey

INTRODUCTION TO THE HOST COMMANDS

The Series 2 Host Command Set is a group of system commands that define, configure and print labels. They consist of four letter mnemonics derived from the ASCII character set.

The following lists identifies all commands and messages.

Series 2 Host Commands by Group

Action Group

BOOT warm reBOOT of the controller	LDEL Label DELete
CLRC CLeaR Counts	RDEF Return to DEFault Settings
DPHD Delete PrintHeaD configuration	SDAT Set DATe for the Controller
FCLR Flash font CLeaR	SDRT Set Shift Date Rollover Time
FDEL Font DELete	SPAL Set PALlet count
FDIR Font DIRectory	STIM Set TIMe for controller
HDIR printHead DIRectory	

Asynchronous Data Return Group	Binary Commands
ALOG Asynchronous LOG	DDWN Device DoWNload

APRT Asynchronous PRinT	FDWN Font DoWNload
APSC Asynchronous Previous Shift Count	FRMD FiRMware Download
BCDT Bar Code scanner DaTa	

Configuration, Basic

BFLD Bar code FieLD	SBOX Set BOX Width
GLGL Get a single LoGical Line	SLNS Set LiNe Speed
LFLD Label Field Definition	SPHD Set PrintHeaD configuration

Configuration, Advanced

LCLS Label CLoSe	SPCI Set PhotoCell Inhibit
SBCD Set Bar CoDe	SPRD Set PRoDuct count
SDTP Set DoT Pitch	SRPX Set RPX page boundaries
SENC Set ENCoder Resolution	SSCL Set SCaLe message parameters
SFMT Set ForMaTting mode	SSEQ Set SEQuence count
SGST Set Global STring	SSFC Set ShiFt Count
SHMI Set Horizontal Motion Index	SSFT Set ShiFt Time
SHST Set HiSTory for identified port	SSPx Set Serial Port x configuration
SLGL Set LoGical Line	SUCx Set User Count for 1 or 2
SMBX Set Missing BoX count	SWEB Set WEB parameters
SMRG Set image MeRGe	TLNS Tune/tweak LiNe Speed

Configuration

BFLD Bar code FieLD	SBOX Set BOX Width
GLGL Get a single LoGical Line	SLNS Set LiNe Speed
LFLD Label Field Definition	SPHD Set PrintHeaD configuration

Label Management	Peer to Peer
LCPY Label CoPY	GLDR Get the ID of the group LeaDeaR
LDIR Label DIRectory	PING Information request from another node
LOPN Label OPeN	
LREN Label REName	PONG Response to a PING command

Printing

PRT1 PRinT identified label 1 time	PWEB Print in WEB mode
PRTC PRinT Continuously	XPRT cancel PRinTing
PURG PURGe indicated channels	

Query

GBCD Get Bar CoDe	GHMI Get Horizontal Motion Index
GBOX Get BOX width	GHST Get HiSTory for I/O ports
GDAT Get DATe setting of controller	GLAB Get LABel names
GDRT Get shift Date Rollover Time	GLBL Get the current LaBeL
GDTP Get DoT Pitch	GLIN Get a range of logical LINes
GENC Get the ENCoder resolution	GLNS Get the current LiNe Speed
GFMT Get the current FormaT mode	GMBX Get Missed BoXes
GGST Get Global String	GMRG Get the MeRGe setting
GHED Get printHEaD information	GPAL Get PALlet count

Query continued

GPCI Get PhotoCell Inhibit setting	QLAB Query LABels on the controller
GPHD Get Printhead Configuration	QLBL Query LaBeL Format

GPRD Get PRoDuct count
GPSC Get Previous Count
GREP Get REPeat distance
GRPX Get RPX settings
GSCL Get SCaLe
GSEQ Get SEQuence count
GSFC Get ShiFt Count
GSFT Get ShiFt Time
GSPx Get Serial Port x configuration
GTIM Get TIMe setting of controller
GUCx Get programmable User Count x
GWEB Get WEB mode settings
QAD1 Query Arcnet Data on port 1
QADR Query arcnet node ADdRess
QAS1 Query Arcnet Status on port 1
QBAT Query BATtery status
QCMD Query last ComManD
QERR Query ERRor status
QFON Query FONts stored on the controller
QFRM Query FiRMware revision
QHED Query printHEaDs on the controller
QLEX Query Label EXistence on controller
QLIN Query logical LINes
QLOG Query LOG
QMAX Query MAXimum line speed
QMEM Query MEMory
QNET Query NETwork map
QNXT Query NeXT station in the network
QPHD Query all PrintHeaDs
QPLN Query all Print LiNes
QPRT Query PRinTing
QPSM Query Print State Machine
QSDx Query Serial Data on port x
QSFT Query ShiFt setup
QSSx Query Serial Status on port x
QST1 Query STatus level 1
QST2 Query STatus level 2
QST3 Query STatus level 3
QST4 Query STatus level 4
QSTS Query STation Status

USE OF THE HOST COMMAND SET

A host command consists of data fields that when chained together form either an instruction or a message to the controller. An instruction has four parts: Preamble, Command, Arguments and an End of Command Indicator. A message consists of three parts: Preamble, Information and an End of Command Indicator.

Messages to the controller are included here in a command discussion because they function as commands when combined with autocodes inside a message **(**see Chapter 4 for explanations of Series 2 autocodes)**..**

The next section defines the fields that make up a command and explains how they function.

Command Fields

Preamble—The Preamble, the start of a communication to the controller, identifies the information that follows as either a command or a message:

ESC- Introduces a command;
STX- Introduces a message of bar code data;
SOH- Introduces a message of scale data.

Command—Command is a specific action for the controller to perform. It consists of four letters that indicate the kind of activity.

There are nine types of commands:

Action commands—for performing a specific function
Binary Commands—for downloading
Configuration, Basic—for basic system configuration
Configuration, Advanced—for advanced system configuration
Get commands—for retrieving stored information.
Label Management—for editing and managing labels
Peer to Peer—for communicating between controllers
Query Commands—for function status information from the controller
Set commands—for storing information

Messages act as commands when the autocode {W} or {Z}. (covered in Chapter 4) is included in the message of a label.

Argument(s)—An Argument is the parameter by which the action will be executed. For example, a single argument command to set the conveyor line speed at 200 feet per minute would be **SLNS, 200**. SLNS (Set LiNe Speed) is the command and 200 is the argument for 200 feet per minute. Commands that require multiple arguments use commas as delimiters.

End of Command Indicator—End of command indicator is the <CR> (carriage return) which indicates the end of a instruction. Press the ENTER key to send a End of Command Indicator.

Message Fields

Messages do not contain a Host Command. Instead, they contain data from a data aquisition device such as a bar code scanner or scale. Each message consists of Preamble, Information and End of Command Indicator.

Preamble—The message preamble is an STX or SOH:

STX indicates a bar code. STX = 02h
SOH indicates a scale. SOH = 01h

Since these preambles originate from devices other than a terminal or a PLC, these acquisition devices need to be programmed to transmit an STX or an SOH to the Series 2 controller.

Information—The information of message is the data acquired by the data acquisition device. For example, a bar code scanner would scan "00012345678905" from a bar code and send the following message to the controller:

<STX>"00012345678905"<CR>

This message would act like a command to the controller to ask the host PC for the label named **00012345678905** and to load it for printing.

End of Command Indicator—End of command indicator is the <CR> (carriage return) which indicates the end of a instruction. Program your data acquisition equipment to end all transmissions with this indicator.

ENTRY GUIDES

A) Series 2 host commands do not permit any editing with the arrow keys, erasing with the DEL key or overtyping. So, if you make a typing mistake, press ENTER, read the error message and retype the command.

B) There is no message of acceptance for successfully entered Host Commands, only error messages for commands with mistakes. In other words, absence of a message after a <CR> means that the controller has accepted your command.

C) Enclose all text entries inside of Host Commands with double quotes (“ “).

D) Do not include any extraneous space characters in a Host Command.

E) Series 2 firmware ALWAYS assigns logical lines (see Section) to printheads. The default is the logical line with the same number as the printhead.

F) All distances indications in Series 2 Host Commands are expressed in thousandths of an inch.

G) The text entries for messages must be in capital letters unless you have a special font. Call your Diagraph Sales Representative for more information about special fonts and logos.

H) The double quotation marks serve as delimiters of text. They are necessary because any message longer that one word will have a space character and that space character can cause problems. A space character, unless it is bracketed with quotes, will become a delimiter. Avoid this problem by always using quotes.

I) Quotes can not be part of a printed message so do not include single or double quotation marks inside the message text.

J) In commands that set print direction, 0 is left to right and 1 is right to left. You can determine the print direction by standing directly behind the printhead for programming and observing the direction the product approaches the printhead.

CONFIGURING PRINTHEADS AND PRINTING LABELS

Two Printhead Example

Configuration—The host commands in this example configure two 18-dot printheads with an offset of one inch. Printhead 1 is printing from left to right and printhead 2 is printing from right to left. To configure the printheads, enter the two commands exactly as shown below. If you make a typing mistake, press ENTER and retype the command. Series 2 host commands do not permit any editing with the arrow keys, erasing with the DEL key or overtyping.

<ESC>SPHD,18,1000,0,1<CR>

<ESC>SPHD,18,1000,1,2<CR>

<ESC> is the preamble for all Series 2 instructions.

The first SPHD (Set PrintHeaD configuration) sets the parameters for the first printhead: 18 dots with an offset of one inch (1000

thousandths) printing in the 0 direction in the first (1) position.

The second SPHD sets the parameters for the second printhead—18 dots with an offset of one inch (1000) printing in direction 1 in the second (2) position.

When these commands are entered, the firmware automatically assigns a logical line to each printhead. Each logical line contains the same number of dots that you assigned for each printhead. The command that governs this is SLGL, Get LoGical Line which occurs in the background.

Label and Message—To print a label named HELLO on both configured printheads, enter the commands exactly as shown below.

<ESC>LOPN,HELLO<CR>

<ESC>LFLD,16,1000,1,1,"PRINT TEST"<CR>

<ESC>LFLD,16,1000,1,2,"PRINT TEST"<CR>

<ESC>LCLS,NORMAL,12000,1<CR>

LOPN (Label OPeN) starts the process of sending a label to the controller and assigns the name "HELLO."

The first LFLD (Label FieLd Definition) sets the parameters for the first printhead: font number 16; label offset at one inch (1000 thousandths); restricts the number of logical lines to 1; fixes the logical line as number 1; and defines the message as "Print Test."

The second LFLD sets the parameters for the second printhead: font number 16; label offset at one inch (1000 thousandths); restricts the number of logical lines to 1; fixes the logical line as number 2; and defines the message as "Print Test."

LCLS (Label CLoSe) closes and saves the information received by the controller since the LOPN command: NORMAL sets the storage mode as static RAM, 12000 sets the box size at twelve inches and 1 identifies the repeat distance.

<CR> is the end of command indicator for all host commands.

Six Printhead Example

For this example, a six printhead configuration will consist of one 9-dot printhead and five 18-dot printheads. Three of the 18-dots will print left to right, direction 0, and the other two will print right to left, direction 1. Enter the following Host Commands exactly as written below.

<ESC>SPHD,18,500,0,1<CR>

<ESC>SPHD,18,500,0,2<CR>

<ESC>SPHD,18,1000,0,3<CR>

<ESC>SPHD,18,1500,1,4<CR>

<ESC>SPHD,18,2500,1,5<CR>

<ESC>SPHD,9,2000,0,6<CR>

Notice that all but the first two printheads are located at different distances from the photocell.

Printheads 1 and 2 are "stacked" on top of each other.

HOST COMMAND EXAMPLES

This section provides examples with actual data of commonly used host commands. For these examples, assume that the font is already resident on the controller board, the default encoder resolution and dot pitch are acceptable, and the time and date on the controller are correct.

SET LINE SPEED
<ESC>SLNS,Speed<CR>

This command example tells the controller to ignore the encoder (if connected) and to set the internal simulated conveyor line speed to 200 feet per minute.

<ESC>SLNS,200<CR>

ESC Preamble.

SLNS Set Line Speed Command

200 Speed Simulated line speed of 200 feet per minute.

<CR> End of command indicator.

LABEL OPEN
<ESC>LOPN,Name<CR>

This command example opens a label file called TEST.

<ESC>LOPN,TEST<CR>

ESC Preamble.

LOPN Command to open a label.

TEST Nam e Name of the label to open

<CR> End of command indicator.

LABEL FIELD DEFINITION
<ESC>LFLD,font,offset,num,L1...[Ln],data<CR>

This command example defines a label field and its necessary parameters. Note that all distances for the Series 2 Host Commands are expressed in thousandths of an inch.

<ESC>LFLD,10,20,2,1,3, "HELLO"<CR>

ESC Preamble.

LFLD Label field definition command used in configuring a label.

10 Font Number of the font.

2000 Offset Distance in thousandths of an inch from the edge of the box to where the message will start printing. In this example, it will be two inches from the edge.

2 Num The quantity of logical lines that will make up this message. The logical line identity numbers follow this quantity entry. In this example, the message will be split between two logical lines.

1	L1	Identity of the first logical line.
3	L2	Identity of the second logical line.
"HELLO"		Message text written all in capitals and enclosed in quotes. The text must be in capital letters unless you have a special font. Call your Diagraph Sales Representative for more information about special fonts and logos. The double quotation marks serve as delimiters of the text. The marks are recommended because any message longer that one word will have a space character and that space character can cause problems. A space character, unless it is bracketed with quotes, will become a delimiter. Avoid this problem by always using quotes. Quotes can not be part of a printed message so do not attempt to include single or double quotation marks inside the message text.
<CR>	End of command indicator.	

SET PRINTHEAD CONFIGURATION:

```
<ESC>SPHD,Resolution,Offset,Direction,Position<CR>
```

The SPHD command sets printhead parameters.

<ESC>SPHD,18,2000,0,1<CR>

ESC	preamble	
SPHD	command	
18	Resolution	Number of dots in the printhead.
2000	Offset	2000 = 2 inches from the photocell.
0	Direction	The print direction: 0 is left to right and 1 is right to left. Determine the print direction by standing directly behind the printhead for programming and observing the direction the product approaches the printhead.
1	Position	indicates that the command is for the first printhead in the chain.
<CR>	end of command indicator.	

COMMON FUNCTIONS OF THE HOST COMMAND SET

This section describes some of the more common Host Command Set functions such as creating, editing and printing a label and assigning logical lines.

CREATING A LABEL

OPENING A NEW LABEL
<ESC>LOPN,Name<CR>

This command will open a file called TEST in the temporary memory.

<ESC>LOPN,TEST<CR>

If you enter LOPN a second time with a different label name, then the first name will be ignored and the second will be the label name. The label name is case-sensitive.

If a label of the same name already exists in static memory, an error message will appear.

CREATING A LABEL FIELD
<ESC>LFLD,Font,Offset,Number of logical lines, Logical line identification, Message<CR>

This command will create a message in the label named TEST after sending the LOPN command.

<ESC>LFLD,10,5000,6,1,2,3,4,5,6,"TEST {SPD}"<CR>

The message will print with font 10 starting five (5000/1000) inches from the edge of the box with six logical lines assigned to six printheads numbered 1, 2,3,4,5 and 6. The message will be "Test " with the {SPD} autocode which prints the line speed.

You could also print the same message by writing to all the printheads separately:

<ESC>LFLD,10,5000,1,1,"TEST {SPD}"<CR>

<ESC>LFLD,10,5000,1,2,"TEST {SPD}"<CR>

<ESC>LFLD,10,5000,1,3,"TEST {SPD}"<CR>

<ESC>LFLD,10,5000,1,4,"TEST {SPD}"<CR>

<ESC>LFLD,10,5000,1,5,"TEST {SPD}"<CR>

<ESC>LFLD,10,5000,1,6,"TEST {SPD}"<CR>

Note that all LFLD commands that you use after the LOPN command are associated with a particular LOPN label name.

LABEL CLOSE
<ESC>LCLS,Mode<CR>

This command closes the label TEST with its associated label fields and stores them in the static memory.

<ESC>LCLS,NORMAL<CR>

This command has two mode choices—Permanent and Normal. "Permanent" will store the label in non-volatile memory and "Normal" will save it to static RAM.

PRINTING A LABEL

There are five different Printing commands:

1 PRT1, Print the identified label 1 time
2 PRTC, Print Continuously
3 PURG, Purge the indicated printhead channels
4 PWEB, Print in Web Mode

5 XPRT, Stop Printing

1. Print One Time, <ESC>PRT1,Label Name<CR>

   This command tells the controller to print the identified label only one time after a photocell trip.

   For example, **<ESC>PRT1,TEST<CR>** tells the controller to print the label **TEST** once only on a photocell signal. After printing the message, the controller will cancel printing and delete the label from static RAM.

2. Print Continously, <ESC>PRTC,Label Name<CR>

   This command sends a label to print continously until an XPRT command is sent. This is the command to use when the goal is to print the same label on many boxes. For example, **<ESC>PRTC,TEST<CR>** tells the controller to print the label **TEST** once for each trip of the photocell.

3. Purge, <ESC>PURG,Type,head-number,channel<CR>

   This command directs the indicated printhead channels to print continously for three seconds. An entry of zero for the channels to print will purge all channels in a printhead. For example, **<ESC>PURG,2,5,0<CR>** sets the kind of purge to 2 (1 is for an immediate purge and 2 is for a purge on a photocell trip) on printhead number 5 for all channels after the next photocell signal.

4. Print in Web Mode, <ESC>PWEB, Label Name, Repeat Distance<CR>

   This command prints the same label many times on the same product such as the repeated labels on a sheet of plywood.

   Repeat Distance is the distance between the end of the last message and the beginning of the next message expressed in thousandths of an inch. For example, **<ESC>PWEB, TEST,1000<CR>** tells the controller to print the label TEST repeatedly when the photocell is tripped and to allow one inch (1000 / 1000) after each label before the next print.

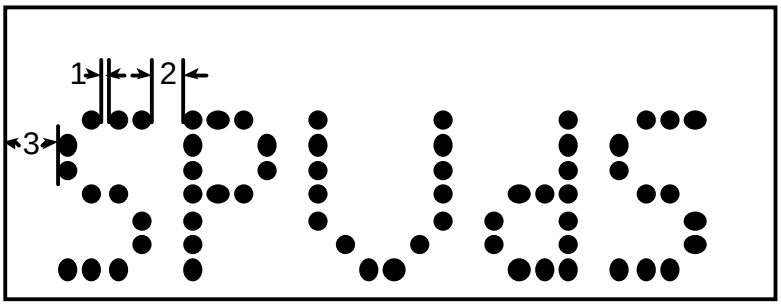
5. Cancel Printing, <ESC>XPRT<CR>

   The XPRT cancels printing at the selected station.

CHAPTER 2
CHARACTER FORMATION

The figure below shows a representation of a message printed with a Diagraph Series 2 system. The numbers reference the spaces associated with character formation which can be altered with host commands.

Figure 2-1
Definable Variables in Ink Jet Printing with Series 2



NUMBER	DESCRIPTION	DEFINING HOST COMMAND
1	Distance between dots in a character.	SDTP - Dot Pitch. See explanation below.
2	Distance between characters	SHMI - Horizontal Motion Index.
3	Distance from the edge of the box to the beginning of the printed message.	SBOX - for products traveling left to right. LFLD - Label Field for products right to left.

DOT-PITCH—Series 2 fonts define their own dot-pitches: For example, font 9BFD60N has a dot-pitch of sixty and 9BFD40N has a dot-pitch of 40. Changing it with SDTP can produce unreadable fonts.

SET HORIZONTAL MOTION INDEX
<ESC>SHMI,Index<CR>

The HMI command adds space to the end of a character.

<ESC>SHMI,3,<CR>

ESC Preamble.

SHMI Command to set horizontal motion index.

3 Index 3 is spacing between characters

<CR> End of command indicator.

Programming Tip Use 3 as the HMI value when programming for 9 or 18-dot printheads.

Related Command: GHMI, Get Horizontal Motion Index.

Fonts

Descriptions

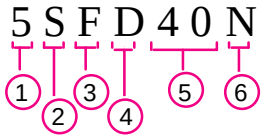
The fonts for the Series 2 system are fixed width fonts: each character uses the same amount of space. When printed, the letter "I" may seem to have more space on either side than would other characters. This is because the printing portion of the letter is small within the same size print matrix. The advantage of a fixed width font is the ease of placing text on the product in alignment with other printed text or graphics. A second advantage is the ease of centering or justifying the text of the product. In order to do this, the font descriptions must be deciphered.

The following is a list of the standard fonts supplied with the series 2 system:

SERIES 2 RESIDENT FONTS

Font Name	Description	Font Name	Description
5SFD40N	5 dot single	9SFD80N	9 dot single
5SFD60N	5 dot single	9BFD40N	9 dot bold
7SFD40N	7 dot single	0BFD60N	9 dot bold
7SFD60N	7 dot single	9BFD80N	9 dot bold
7SFD80N	7 dot single	18BFD40N	18 dot bold
7BFD40N	7 dot bold	18BFD60N	18 dot bold
7BFD60N	7 dot bold	18BFD80N	18 dot bold
7BFD80N	7 dot bold	18XFD60N	18 dot extra bold
9SFD60N	9 dot single	18XFD80N	18 dot extra bold

Series 2 Font Name



1 - 5, 7, 9 or 18 dots high
2 - Single, Bold or eXtra bold
3 - Fixed width
4 - Dual column
5 - 40, 60 or 80 thousandths
6 - Normal or Reverse

The first set of numbers in the font indicates the height of the font in dots and as you can see, there are only four heights: 5, 7, 9 and 18.

The letter immediately after the font height indicates the darkness of the font: S - single, B - bold, or X - extra bold. Following the bold factor is an "F" for fixed width, currently the only available font. The "D" following indicates Dual column printhead font. This is also the only available font of this type. The two numbers following the "D" indicate dot column width, not character width. The matrix size of a font changes with each of these letters to accommodate the extra dots.

Dot column width is the distance in thousandths of an inch from one dot column to the next within a font character. Available options are 40, 60, 80. For example, 40 means 40 thousandths of an inch from one column of print to another column of print. These distances help determine the character width of a character within a font. Custom sizes are available upon request.

Finally, the letter in the font name is either N for Normal, or R for Reverse. This can be misleading: the normal or reverse description refers to the column of the starting dot. There are two columns of dots in our printheads: dot 1 is in column 1 and dot 2 is in column 2. If the starting dot of the logical line is in column 1, then a Normal font is used, our standard. Reverse fonts are available upon request.

Matrix sizes and Character Widths

Font	Matrix	Width at 40	Width at 60	Width at 80
5 single font:	5x7	.28 in	.42 in.	.56 in.
7 single font:	7x8	.32 in.	.48 in.	.64 in.
7 bold font:	7x10	.4 in.	.6 in.	.8 in.
9 single font:	9x9	.36 in.	.54 in.	.72 in.
9 bold font:	9x12	.48 in.	.72 in.	.96 in.
18 bold font:	18x18	.72 in.	1.08 in.	1.44 in.
18 extra bold font:	18x24	.96 in.	1.44 in.	1.92 in.

Message Length Calculations

To calculate the character width, take the width from the matrix size and multiply it by the dot column to dot column distance. For example, the 5SFD40N font has a 5 x 7 matrix. Multiply the matrix width by the dot column width: 7 x .040 = .28 inches.

To calculate actual message lengths, you must also factor in the horizontal motion index (HMI). HMI refers to the number of print columns to space between printed characters. HMI is measured in hundredths of an inch with a default of 3 (.03 inches between characters). Therefore, to calculate a three character message with the font from the example above, multiply the character width; 0.28, by the number of characters; 3, then add the product of the HMI; 0.03, multiplied by the # of characters minus one; 2, thus .28 x 3 + .03 x (3-1) = .9 inches.

Font Character Qualifications

The ASCII characters defined in the fonts of the Series 2 system are hexadecimal 20 through 5F. Print tests have shown that the characters identified below are <u>not</u> defined a font or ASCII range:

- 00 1F - special control characters
- 60 the reverse apostrophe (`)
- 61 7A- the lowercase characters
- 7B left brace ({)
- 7C pipe (|)
- 7D right brace (})
- 7E tilde (~)
- 7F control character (del)

All characters 20 through 5F print normally, except for the following:

5C which should be the back slash (\) symbol actually prints the cent sign.

5E is the caret (^) symbol. If it is inserted into a string to be printed, the entire message will not print unless it is followed by a specific character (such as B, U, R, T). This symbol changes attributes of the characters that follow it such as bold (B), upside down (U), reverse (R) or test print (T).

22 is the initial quotation mark (“) symbol. This symbol should not be used because it causes errors which may result in the lost of print and/or labels. it is used as a delimiter on Series 2 and currently cannot be inserted in a string to be printed without an error occurring. This, however, is one condition which is not caught by the Series 2 software

3B is the semi-colon symbol (;) actually prints the registered trademark symbol ®.

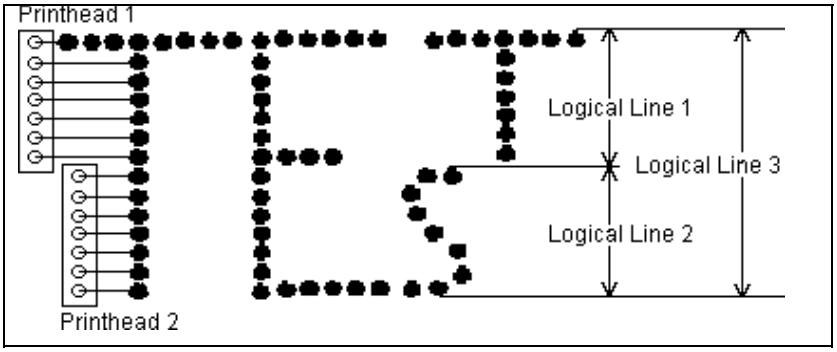
MESSAGE CHARACTERISTICS

This section covers some of the variables and parameters that control and define labels printed with a Series 2 System.

Logical Lines

A logical line is any number of dots in a vertical line configured to print a message or a logo. This line of dots is also called a “logical line.” The terms are interchangeable. See the *Series 2 User’s Manual*, Section 1.4 for more explanation.

Figure 2-2
Two Printheads Printing Three Logical Lines



The figure above shows three separate messages: Logical Lines 1 and 2 are each seven dots and Logical Line 3 is 14 dots high.

Message 1, the small **T**, was assigned to logical line #1.

Message 2, the small **S**, was assigned to logical line #2.

Message 3, the large **TE**, was assigned to logical line #3.

Whenever you configure a printhead with the SPHD command, the firmware automatically configures a logical line. Logical lines 1 and 2 were created with two SPHD commands:

<ESC>SPHD,7,50,0,1<CR>

<ESC>SPHD,7,150,0,2<CR>

<ESC>SLGL,3,1:14<CR>

The first command assigns seven dots, channels 1 through 7, with an offset of 50 to print left to right from printhead 1. The second SPHD assigns seven dots, channels 8 through 14, to the next available printhead which is number 2.

Since there are only two printheads in this example, a command other than SPHD is needed to configure the 14 dot logical line 3. The SLGL (Set LoGical Line) command allows you to create a logical line across several printheads. SLGL names the 14 dot line number (3), identifies its starting dot (1), interposes a colon and identifies the last dot (14).

MULTIPLE HEAD PRINTING

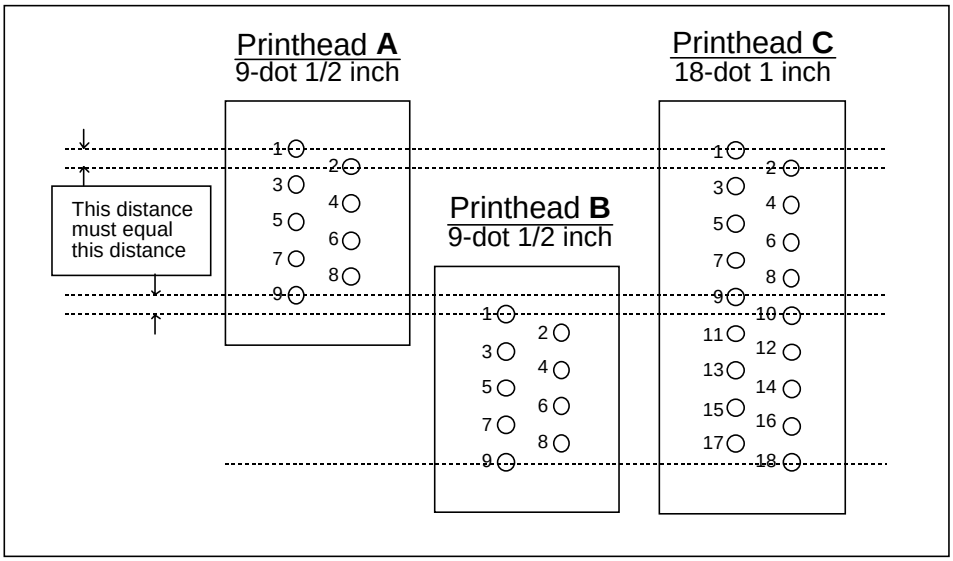
Guides and Cautions

The Series 2 system can print fonts and logos across multiple printheads—several printheads become one large "virtual" printhead. This section provides guides and cautions for successful printing with several printheads.

Printing Across Multiple Printheads

Pictured below is a front view of three Diagraph printhead face plates. On the left and in the center are two 9-dot 1/2" printheads (A and B). On the right is an 18-dot 1" printhead (C). Notice that the two 9-dot printheads are staggered so that together they cover the same print area as the 18-dot printhead. Orifice 1 on printhead A lines up with orifice 1 on printhead C while orifice 9 on printhead A lines up with orifice 9 on printhead C. In addition, orifice 1 on printhead B lines up with orifice 10 on printhead C and orifice 9 on printhead B lines up with orifice 18 on printhead C.

Figure 2-3
Two 9-dot Printheads vs. One 18-dot Printhead



In Series 2 applications, two staggered 9-dot printheads like A and B would not be able to print a readable 18-dot font: the top half of the font would look fine but the bottom half would be illegible. The reason for this lies in the design of the 18-dot fonts, which are designed for a printhead with 9 orifices on the left and 9 orifices on the right.

Notice that each of the two 9-dot printheads has five orifices on the left and four orifices on the right. When staggering two 9-dot printheads, you would have a total of ten orifices on the left and eight orifices on the right. On the other hand, the 18-dot printhead has nine orifices on the left and nine orifices on the right, exactly how the 18-dot fonts on Series two are designed. All Series 2 fonts factory-installed on the controller board are designed to have the top dot printed by an odd numbered orifice of a printhead (located on the left of the faceplate). Each subsequent dot alternates between an even and odd numbered orifice.

Standard fonts, such as 9BFD60N, contain an "N" at the end of the font name which stands for normal. "Normal" means that the top dot must be printed by an odd numbered orifice. Diagraph has also generated reverse fonts that allow the top dot to be printed by an even numbered orifice. These fonts are identified by an "R" at the end of the font name instead of "N," such as 9BFD60R. While a reverse font exists for every normal font, reverse fonts do not ship with a Series 2 system unless they are specifically requested.

Due to the design of Diagraph printhead faceplates, if all solenoids of a printhead were to fire at once, you would get a staggered print identical to the pattern of orifices on the faceplate. In order to get the dots fired from the orifices on the right to line up vertically with the dots fired from the orifices on the left, the Series Two system must compensate

by firing the solenoids on the left .019 of an inch sooner (or later depending on the direction the product is traveling) than the solenoids on the right. This is accomplished through "padding" in the fontÑadding space between individual dots.

Selecting the Odd Starting Dot

All Series 2 fonts are designed to have the top dot of a font print with an odd numbered orifice. Starting from an odd numbered orifice is critical to making the system print legible fonts. When configuring logical lines, you need to identify the dot (or orifice) at which a font will start printing. For example, with the three printheads shown on the previous page, you might want to print a 9-dot font with printhead A, a 7-dot font with printhead B and an 18-dot font with printhead C. For printhead B, even though you are only going to be using only seven of the available nine dots, the starting dot must either be orifice 1 or 3. It may not be orifice number 2.

Recommended Configuration

Because 18-dot 1" and 2" printheads have an even number of orifices, staggering them does not produce the same problem encountered when staggering 9-dot printheads. No matter how many 18-dot printheads you have in sequence, you will always have the same number of orifices on the left as you do on the right. This makes it possible to print fonts across multiple printheads when 18-dot printheads are used. However, as explained above, an odd numbered orifice of a printhead must always be designated the starting dot of any Series 2 font.

If during the installation of a Series 2 system, you get print from a printhead that is illegible, it may be caused by incorrect logical line configuration. Go back and ensure that the logical line configuration is correct as explained above.

CHAPTER 3
HOST COMMAND GUIDE

When command parameters have no default values, the "Default" field name has been eliminated.

"Related Cmd:" identifies commands related to the command under discussion.

ALOG Asynchronous LOG

Provides a summary that includes the last label printed, the sequence count, product count, number of pallets, pallet count, user defined count 1 and user defined count 2. It is sent to the controller after an XPRT, PRTC, PRT1 and PWEB.
*** THIS IS NOT A HOST COMMAND ***

Group: Asynchronous Data Return
Format: N/A
Return Data: ALOG, last label, sequence, product, pallets, palcnt, user 1, user 2<CR>
Parameter 1: Last Label: The last label printed by controller.
Parameter 2: Sequence: Current value of the sequence count.
Parameter 3: Product: The current value of the product count.
Parameter 4: Pallets: The number of pallets of product printed thus far.
Parameter 5: Palcnt: The number of items on the current pallet.
Parameter 6: User 1 and User 2: The values of the user definable counts.
Example: The following data would be sent after the label "DIAGRAPH" had printed and the print mode changed:
ALOG,DIAGRAPH,123456,34,2,4,123456,123456<CR>
Last label printed = DIAGRAPH
Sequence count = 123456
Product count = 34
Pallets = 2
Pallet count = 4
User 1 Count = 123456
User 2 Count = 123456
Related Cmd: APRT

APRT Asynchronous PRinT

This command is sent back to the host after completing a print operation so the host knows that the operation has completed.
*** THIS IS NOT A HOST COMMAND ***

Group: Asynchronous Data Return
Format: N/A
Return Data: <ESC> APRT, label name, MM/DD/YY, HH:MM:SS <CR>
Parameter 1: label name: Name of the label just completed.
Parameter 2: MM/DD/YY - date stamp
Parameter 3: HH:MM:SS - time stamp
Example: The controller will send back APRT,"DIAGRAPH",07/20/96,10:35:45 after completing the print operation of the label "DIAGRAPH" on 7/20/96 at 10:35:45.

APSC Asynchronous Previous Shift Count

This command is sent at the end of a shift and reports the shift count value for that shift. *** THIS IS NOT A HOST COMMAND ***

Group: Asynchronous Data Return

Format: N / A

Return Data: PSC, Prevcount

Parameter 1: Prev Count: Shift count for the shift just ending.

Example: At the end of a shift in which the printer printed 1,234 labels, the following would be sent to the controller: APSC,1234

BCDT Bar Code scanner DaTa

The controller sends BCDT automatically when a valid scan occurs when the scanner has been defined with the SBCD command.

Group: Asynchronous Data Return

Format: N / A

Return Data: BCDT,Data<CR>

Parameter 1: Data: ASCII data from scanning device. The data is enclosed in quotes (“”). Length and format are defined by the scanning device.

Example: The controller reports **BCDT,”00012345678905”<CR>** to the host after receiving 00012345678905 as a valid scan from the bar code scanner.

Related Cmd: SBCD

BFLD Bar code FieLD

Defines a bar code field. See LFLD parameters for definitions of offset, num, L1, L2, and LN.

Group: Basic Configuration

Format: <ESC>BFLD,offset,num,L1,[L2,]...[LN,]sym,chk,quiet, wb,nb,ws,ns,bbw,data<CR>

Return Data: None.

Parameter 1: sym: Bar code symbology. I25 for I 2 of 5 and 39 for Code 39.

Parameter 2: chk: Generate check character flag; 0 to disable check character generation and 1 to generate check character.

Parameter 3: quiet: Width of quiet zone in dots.

Parameter 4: wb: Width of wide bars in dots.

Parameter 5: nb: Width of narrow bars in dots.

Parameter 6: ws: Width of narrow spaces in dots.

Parameter 7: ns: Width of narrow bars in dots.

Parameter 8: bbw: Width of bearer bar in dots data: ASCII representation of data for output.. This parameter is not used in Code 39.

Parameter 9: data: ASCII representation of data for output.

Example: **<ESC>BFLD,12000,1,15,I25,1,25,4,1,5,3,8,0123<CR>**
will print the bar code 0123 as an I 2 of 5 bar code on logical line 15.
<ESC>BFLD,12000,1,15,C39,1,25,4,1,5,3,0123<CR>
will print the bar code 0123 as Code 39 on logical line 15

Related Cmd: LFLD

BOOT Warm re-BOOT of the Controller

Resets the controller, as if power were turned off then back on. Printing operations will be interrupted and halted until a new print command is issued. This command is useful when two controllers share a single power supply.

Group: Action

Format: <ESC>BOOT<CR>

Return Data: None

Parameter 1: None

Example: You need to reboot one controller when two controllers share a single enclosure. Issue this command: **<ESC>BOOT<CR>**

Related Cmd: RDEF

CLRC Clear Counts

Clears the sequence count, user counts, product count, shift count, and pallet counts to 0.

Group: Action

Format: <ESC>CLRC<CR>

Return Data: None.

Parameter 1: None

Example: **<ESC>CLRC<CR>** clears all counts in the controller:

Related Cmd: GSEQ, SSEQ, SPRD, GPRD.

DPHD Delete PrintHeaD configuration

Deletes a printhead from the daisy chain map of the heads

Group: Action

Format: <ESC>DPHD, Position<CR>

Return Data: None.

Parameter 1: Position: Nine ASCII digit number of the printhead position as wired into the printhead daisy chain.

Example: To delete the fourth printhead, send **<ESC>DPHD,4<CR>** to the controller:

Related Cmd: SPHD, QPHD

FCLR flash Font CLeaR

Erases the flash memory used to hold fonts 1 through 20. Not only are these fonts deleted, but the font control tables are cleared as well, restoring the flash memory to the completely erased factory-fresh state. Other fonts are unaffected.

Group: Action

Format: <ESC>FCLR<CR>

Return Data: None

Parameter 1: None

Example: **<ESC>FCLR<CR>** will delete all the fonts from flash memory

Related Cmd: FDEL

FDEL Font DELete

Deletes an identified font. The font numbered 0 is stored in EPROM and is non-volatile and cannot be deleted. Fonts numbered in the range 1-20 are stored in flash memory and are non-volatile. Fonts numbered in the range 21-22 are stored in RAM and are volatile.

Group: Action

Format: <ESC> FDEL, Font ID<CR>

Return Data: None.

Parameter 1: Font ID: Two ASCII digit number of the font to delete. Range is 1-22, inclusive.

Example: **<ESC>FDEL,14<CR>** will delete font #14

Related Cmd: FCLR

FDIR Font DIRectory

Returns a directory of all fonts currently resident on the controller. Used when creating labels from the Host Command Set. Font 0 is a font that is stored in EPROM.

Group: Action

Format: <ESC>FDIR<CR>

Return Data: FDIR, fnum1:fname1, ... , fnumN: fnameN

Parameter 1: fnumN: The font slot number for the Nth font

Parameter 2: fnameN: The name of the font in slot fnumN.

Example: **<ESC>FDIR<CR>** returns a directory of all fonts currently resident on the controller.

FDWN Font Download

Downloads a font. This is a binary command with multi-byte data items stored in a Little Endian format. Fonts numbered in the range 1-20 are stored in flash memory and are non-volatile. Fonts numbered in the range 21-22 are stored in RAM and are volatile.

Group: Binary Command

Format: FDWN,ReturnSize

Return Data: 16

Parameter 1:

Byte 0:	<si>
Bytes 1-4:	Mnemonic
Byte 5:	Font ID
Bytes 6-7:	Size
Bytes 8-(size-8):	Font Data

Parameter 2: Return Size: Size of the font that was downloaded

FRMD Firmware Download

Initiates a firmware download. Note that FIRMLOAD.EXE is required to reload the firmware.

Group: Binary Command

Format: <ESC>FRMD, 3563,07,094<CR>

Return Data:

Parameter 1: 3563,07,094 Requiring these parameters prevents the accidental erasure of the current firmware. When the system sees this command, it invalidates the current flash memory and reboots, necessitating downloading new firmware.

Example: **<ESC>FRMD, 3563,07,094<CR>** will initiate the firmware download sequence on the controller.

GBCD Get Bar CoDe

Gets the lookup bar code input setup.

Group: Query

Format: <ESC>GBCD<CR>

Return Data: GBCD, Start, mode, Readlen, Start, Len<CR>

Parameter 1: Start: This is the decimal equivalent of the ASCII code for the character to precede lookup data.

Parameter 2: mode: Indicates whether the bar codes are variable or constant length (V=variable, C=constant).

Parameter 3: Readlen: Length of the bar code reading string from the scanner in bytes.

Parameter 4: Start: Position of the first byte to be used. Any bytes before this position or after position Start+Len will be ignored.

Parameter 5: Len: Length of the portion of the string that is used.

Example: **<ESC>GBCD<CR>** will return the lookup bar code reader setup.

Related Cmd: SBCD

GBOX Get BOX width

Gets the current box width.

Group: Query

Format: <ESC>GBOX<CR>

Return Data: GBOX, XXX

Parameter 1: XXX: the box width in thousandths of an inch

Example: **<ESC>GBOX<CR>** will return the current box width.

Related Com: SBOX

GDAT — Get DATe setting of controller

Retrieve the date from the controller

Group: Query

Format: <ESC>GDAT<CR>

Return Data: GDAT, Date<CR>

Parameter 1: Date: Eight ASCII character string representing the current date in format, DD:MM:YY.

Example: **<ESC>GDAT<CR>** will return the date that would be printed by the {D} autocode.

Related Com: SDAT

GDRT — Get shift Date Rollover Time

Gets the time at which the date rolls over.

Group: Query

Format: <ESC>GDRT <CR>

Return Data: GDRT, HH:MM

Parameter 1: HH: The hour of the current time. The time is returned in 24 hour format.

Parameter 2: MM: The minutes of the current time.

Example: **<ESC>GDRT<CR>** will return the time at which the date rolls over.

Related Com: SDRT

GDTP — Get DoT Pitch

Retrieves the distance between dots.

Group: Query

Format: <ESC>GDTP<CR>

Return Data: GDTP, Dotpitch<CR>

Parameter 1: Dotpitch: Current dot pitch in thousandths of an inch.

Example: **<ESC>GDTP<CR>** will return the horizontal distance between dots.

Related Com: SDTP

GENC — Get ENCoder resolution

Get the number of encoder ticks in each inch of horizontal distance.

Group: Query

Format: <ESC>GENC<CR>

Return Data: GENC, Resolution<CR>

Parameter 1: Resolution: Nine ASCII digit representation of the actual encoder resolution in ticks per inch. This number does not reflect the hardware multiplier and is always in English units.

Example: **<ESC>GENC<CR>** will return the current encoder resolution in ticks per inch.

GFMT Get ForMaT mode

Returns the current formatting mode.

Group: Query

Format: <ESC>GFMT <CR>

Return Data: GFMT, mode

Parameter 1: Mode: The current format setting.

0 - Image Post-Format

1 - Image Pre-Format (PWEB & PRTC only).

Default 1: 0

Example: <ESC>GFMT<CR> will cause the controller to send .

GGST Get Global String

Retrieves global strings. This Global strings allow the user to set certain, constant data that may vary between locations. The SGST command sets global strings.

Group: Query

Format: <ESC>GGST<CR>

Return Data: GGST, ID, String<CR> For each of the 10 global strings

Parameter 1: ID: Two ASCII digit number representing the global string identifier. (Valid range 1 to 10.)

Parameter 2: String: Twenty-five ASCII character string.

Example: **<ESC>GGST<CR>** will return the values of the 10 global strings.

GHED Get HEaD

Reports the configurations of a quantity of printheads beginning with the printhead in position “start_offset.”

Group: Query

Format: <ESC> GHED, start_offset, count <CR>

Return Data: GHED, NUM1, DOTS1, OFFSET1, DIR1,...NUMcount, DOTScount, OFFSETcount, DIRcount

(This format is used for each head for which data is requested.)

Parameter 1: Start_offset - Number of printheads to skip before starting report

Parameter 2: Count - Number of printheads to report on.

Parameter 3: NUMn - Number of nth printhead in the list.

Parameter 4: DOTSn - Number of dots in the nth head in the list.

Parameter 5: OFFSEtn - Offset from photo-cell of print head n

Parameter 6: DIRn - Direction of print for printhead on.

0 - right

1 - left

Example: **<ESC>GHED,3,3<CR>** will return the head parameters for heads 3, 4 and 5.

Related Cmd: SPHD, QPHD

GHMI Get Horizontal Motion Index

Gets the current setting for the Horizontal Motion Index—the number of dots between adjacent characters.

Group: Query

Format: <ESC>GHMI<CR>

Return Data: GHMI, Index<CR>

Parameter 1: Index: Nine ASCII digit number indicating the number of dots between character columns when printing fixed pitch fonts. Spacing is affected by the index value for all printable ASCII characters.

Example: **<ESC>GHMI<CR>** returns the current value of the horizontal motion index.

Related Cmd: SHMI

GHST Get HiSTory

Retrieves parameter settings for gathering history data for each I/O port

Group: Query

For mat: <ESC>GHST<CR>

Return Data: GHST, Count, Size, ASP1 :Flag1, SP2 :Flag2, SP3 :Flag3, SP4 :Flag4, ARC1 :Flag5, PP1 :Flag6<CR>

Parameter 1: FlagN: 3 ASCII character mnemonic indicating status of data history for the specified port.
ON: Data history enabled
OFF: Data history disabled

Example: **<ESC>GHST<CR>** returns the current history gathering settings.

GLAB Get LABels

Returns label names starting at label start_offset. Labels are sorted alphabetically. The number of label names will be defined by the count parameters.

Group: Query

Format: <ESC> GLAB, start _ Offset, count <CR>

Return Data: GLAB, label _ 1 label _ 2,... label-count

Parameter 1: Start _ Offset number of labels to skip.

Parameter 2: Count: Number of labels names to return.

Parameter 3: Labeling: The nth label name returned.

Example: **<ESC>GLAB,3,5<CR>** will return labels 3, 4, 5, 6, and 7.

Related Com: GLBL

GLBL Get LaBeL

Returns the label stored on the controller.

Group: Query

Format: <ESC> GLBL, label-name <CR>

Return Data: GLBL, label-name_, line_cnt <CR> LOPN, label_name <CR>

List of LFLD and BFLD commands as appropriate LCLS, storage-type, box-width <CR>

Parameter 1: label_name: Name of the desired label.

Parameter 2: Line_cnt: Number of lines in the label.

Default 2: Storage Type: 10 ASCII character mnemonic indicating the storage made for the open label (possible values: PERMANENT, NORMAL).

Parameter 3: box-width: The width of the box in thousandths of an inch.

Example: **<ESC>GLBL,DIAGRAPH<CR>** returns the commands that are necessary to recreate that label in memory again.

GLDR Get LeaDR

Reports the ID of the leader in a group to which the current controller belongs or it reports the ID of the leader in which the current controller is in.

Group: Peer to Peer

Format: <ESC> GLDR <CR>

Return Data: GLDR, Leader <CR>

Parameter 1: leader - the leader in a group to which the current controller belongs or the ID of the leader in which the current controller is in.

Example: **<ESC>GLDR<CR>** reports the leader.

GLGL Get LoGical Line

Gets a logical line.

Group: Basic Configuration

Format: <ESC> GLGL, line <CR>

Return Data: <ESC>GLG, line, S1: E1: [,S2: E2]. . .[,SN :EN] <CR>

Parameter 1: Line: Two ASCII digit logical line number. Range is 1 to 36.

Parameter 2: SN: Nine ASCII digit representing the starting dot number of the specified logical line. This number must be followed by a colon.

Parameter 3: EN: Nine ASCII digit representing the ending dot number of the specified logical line.

Example: <ESC>GLGL, 5 <CR> will report the information on logical line 5.

GLIN Get LINes

Returns selected range of logical lines

Group: Query

Format: <ESC> GLIN start_offset, count <CR>

Return Data: GLIN, LINE_NUM1, FIRST_DOT1: LAST_DOT1,..., LINE.NUMcount, FIRST_DOTcount: LAST.DOT count <CR>

Parameter 1: start_offset: Number of logical lines to skip to get at the first to be repeated.

Parameter 2: Count: Number of logical line definitions to be reported.

Parameter 3: LINE-NUMn: Logical line number of the nth definition returned

Parameter 4: FIRST_DOTn: First dot of nth logical line definition returned.

Parameter 5: LAST_DOT n: Last dot of the nth logical line definition returned.

Example: **<ESC>GLIN,3,5<CR>** will return the information on logical lines 3, 4, 5, 6, and 7, in the above format.

GLNS Get Line Speed

Returns the current line speed.

Group: Query

Format: <ESC>GLNS<CR>

Return Data: GLNS, Speed, Mode<CR>

Parameter 1: Speed: Nine ASCII digit representation of the line speed in feet per minute or in tenths of meters per minute.

Parameter 2: Mode :Nine ASCII character string indicating the source of line speed. (Possible values: ACTUAL, or SIMULATED)

Example: **<ESC>GLNS<CR>** returns the present line speed and the source of the line speed information.

Related Cmd: SLNS

GMBX Get Missed BoXes

Returns the current value of the missing box count

Group: Query

Format: <ESC>GMBX<CR>

Return Data: GMBX,

Parameter 1: missedboxes: Nine ASCII character string giving the number of boxes that have not been printed on but have tripped the photocell.

Example: **<ESC>GMBX<CR>** returns the number of boxes missed.

GMRG Get MeRGe setting

Returns the current setting for the MRG control.

Group: Query

Format: <ESC>GMRG<CR>

Return Data: GMRG, setting

Parameter 1: setting - 0: merging is disabled
1: merging is enabled

Example: **<ESC>GMBX<CR>** returns the current missing box count value.

Related Cmd: SMRG

GPAL Get Pallet Count

Reports the current values for the pallet related counts

Group: Query

Format: <ESC>GPAL<CR>

Return Data: GPAL, Palcount, Count, Size<CR>

Parameter 1: Palcount: Number of item on the current pallet.

Parameter 2: Count: Nine ASCII digits representing the number of pallets that were filled with printed product.

Parameter 3: Size: Nine ASCII digits representing the number of items per pallet.

Example: **<ESC>GPAL<CR>** returns the current values of the pallet count and the items on the current pallet.

Related Cmd: SPAL

GPCI Get PhotoCell Inhibit

Gets the current setting for the Photocell Inhibit Setting

Group: Query

Format: <ESC> GPCI <CR>

Return Data: GPCI, mode,value

Parameter 1: mode -- Indicates whether photocell inhibit is on:

1 - enabled 0 - disabled

Parameter 2: Value - Distance, in .001 inch, that the photocell is ignored.

Example: **<ESC>GPCI<CR>** can return either GPCI, 0 or GPCI, 1

Related Cmd: SPCI

GPHD Get PrintHead Configuration

Returns the current printhead configuration.

Group: Query

Format: <ESC>GPHD, Head<CR>

Return Data: GPHD ,Resolution, Offset, Direction, Position<CR>

Parameter 1: Head: The number of the head that is to be examined.

Parameter 2: Resolution: Nine ASCII digit number of the number of vertical dots on the printhead.

Parameter 3: Offset: Nine ASCII digit number of the offset in the X direction given in thousandths of inches. Range 0 to 100,000 thousandths inches

Parameter 4: Direction: Nine ASCII digit number of printing direction. Zero indicates printing with the line direction and non-zero indicates printing opposite the line direction.

Parameter 5: Position: Nine ASCII digit number of printhead position as wired into the printhead daisy chain.

Example: **<ESC>GPHD,1<CR>** returns the current configuration of printhead #1.

Related Cmd: QPHD

GPRD Get Product Count

Retrieves the current product count.

Group: Query

Format: <ESC>GPRD<CR>

Return Data: GPRD, Count<CR>

Parameter 1: Count: Nine ASCII digits representing the number of items printed.

Example: **<ESC>GPRD<CR>** returns the current product count.

Related Cmd: SPRD, GSEQ, SSEQ

GPSC `Get Previous Shift Count`

Gets the value of the previous shift count

Group: Query

Format: <ESC>GPSC<CR>

Return Data: GPSC, Prevcount<CR>

Parameter 1: Prevcount: Shift count for the previous shift. Prevcount is maintained until the end of the current shift, when the current shift count (available with GSFC) becomes the previous shift count.

Example: **<ESC>GPSC<CR>** returns the count of items printed in the previous shift.

GREP `Get REPeat distance`

Returns the current repeat distance in thousandths of an inch.

Group: Query

Format: <ESC> GREP <CR>

Return Data: GREP, distance <CR>

Parameter 1: distance: Current repeat distance in thousandths.

Example: **<ESC> GREP<CR>** returns GREP, 10000 if the repeat distance is 10.000 inches.

Related Cmd: SREP

GRPX `Get RPX Settings`

Reports the RPX page boundaries. The RPX memory space is divided into two areas: a font page area and an image page area. This command allows you to determine which of the three possible page divisions is currently in use.

Group: Query

Format: <ESC>GRPX<CR>

Return Data: GRPX, mode

Parameter 1: Mode: A numeric value from 0 to 2

0: Font at 0xA00000, Image at 0xB00000

1: Font at 0xA00000, Image at 0xC00000

2: Font at 0xA00000, Image at 0xD00000

Default 1: Factory default: 0

GRPX, 0 would be returned after power-up of a controller using factory defaults (RDEF). Therefore, the font area would start at 0xA00000 and the image at 0xB00000

Example: **<ESC>GRPX<CR>** will return the current RPX mode.

Related Cmd: SRPX

GSCL `Get Scale`

Reports the setup for the scale input.

Group: Query

Format: <ESC>GSCL<CR>

Return Data: GSCL, StartChar, Mode, Readlen, Start, Len<CR>

Parameter 1: StartChar: Character that precedes the scale data. This is the decimal equivalent of the ASCII code.

Parameter 2: Mode: Flag that indicates whether the scale data is variable (V) or constant (C) length.

Parameter 3: Readlen: Length of the string received from the scale.

Parameter 4: Start: Position of the first byte to be used. Any bytes before this position, or after position Start+Len, will be ignored.

Parameter 5: Len: Length of the portion of the string that is used

Example: **<ESC>GSCL<CR>** will return the setup for the scale input.

Related Cmd: SSCL

GSEQ Get Sequence Count

Retrieves the sequence count.

Group: Query

Format: <ESC>GSEQ<CR>

Return Data: GSEQ, Count, Modulus<CR>

Parameter 1: Count: Nine ASCII digits representing the number of items printed.

Parameter 2: Modulus: Nine ASCII digits representing the wraparound value. When Count reaches Modulus, the sequence count wraps back to 1; unless Modulus is 0, in which case it wraps to 0 when it overflows from all 9's.

Example: **<ESC>GSEQ<CR>** returns the modulus and the current values of the sequence count. For example, a Modulus value of 000 would wrap to 000 when the sequence count reaches 999. Also, a modulus value of 3456 would wrap to 0001 when it reaches 3456.

Related Cmd: SSEQ

GSFC Get Shift Count

Reports the value of the shift count

Group: Query

Format: <ESC>GSFC<CR>

Return Data: GSFC, Count<CR>

Parameter 1: Count: Nine ASCII digits representing the number of items printed during the current shift.

Example: **<ESC>GSFC<CR>** retrieves the number of prints made during the current shift.

Related Cmd: SSFC

GSFT Get Shift Time

Retrieves the shift settings.

Group: Query

Format: <ESC>GSFT<CR>

Return Data: GSFT,shift1,....shiftN<CR>

Parameter 1: shiftN: Eight ASCII character string representing the starting shift time in a 24 hour format, for shift N. The shifts are returned in order from midnight (00:00).

Example: **<ESC>GSHFT<CR>** will return the shift settings.

Related Cmd: SSFT

GSPx Get Serial Port x Configuration

Returns the information on the indicated serial port 1 to 4.

Group: Query

Format: <ESC>GSPn<CR>

Return Data: GSPn, Use Code, Device Code, Baud Rate, Parity, Data Bits, Stop Bits, Handshake<CR>

Parameter 1: Use code: 4 ASCII character mnemonic indicating type of serial device connected.

Parameter 2: Device code: 4 ASCII character mnemonic that further describes the Use code.

Parameter 3: Baud Rate: Nine ASCII digit number indicating the baud rate of the serial port.

Parameter 4: Parity: 4 ASCII character mnemonic indicating type of parity implemented.

Parameter 5: Data Bits: Nine ASCII digit number indicating the number of data bits.

Parameter 6: Stop Bits: Nine ASCII digit number indicating the number of stop bits.

Parameter 7: Handshake: 4 ASCII character mnemonic indicating the type of hardware handshaking to be used.

Example: **<ESC>GSP2<CR>** will return the configuration of serial port #2.

Related Cmd: SSPX

GTIM Get Time Setting of Controller

Gets the current time, as set on the controller.

Group: Query

Format: <ESC>GTIM<CR>

Return Data: GTIM, Time<CR>

Param eter 1: Time: Eight ASCII character string representing the current time in 24 hour format, HH:MM:SS.

Example: **<ESC>GTIM<CR>** will return the current time as set on the controller.

Related Cmd: STIM

GUCx Get User Count 1 or Get User Count 2

Gets the value of the indicated user count, along with the settings for the programmable attributes.

Group: Query

Format: <ESC>GUCn<CR>

Return Data: GUCn, Val, Sign, Delta, Modulus<CR>

Parameter 1: Val: Nine ASCII digits representing the number of user definable items that were printed.

Parameter 2: Sign: + for increment by Delta, - for decrement.

Parameter 3: Delta: Nine ASCII digits representing the amount by which to increment or decrement after each item.

Parameter 4: Modulus: Nine ASCII digits representing the count at which to wrap around.

Example: **<ESC>GUC1<CR>** will return the current value and settings for User Definable count #1.

Related Cmd: SUCX

GWEB Get WEB mode settings

Returns the current web mode settings of immabort and counttrigs. "Immabort" tells when printing will stop and "counttrigs" indicates when label counts increase.

Group: Query

Format: <ESC>GWEB<CR>

Return Data: GWEB, immabort, counttrigs

Parameter 1: immabort: Single ASCII digit that indicates whether printing stops when the photocell no longer detects anything to print on after completing the last label: 1 = abort immediately and 0 = abort after completing label.

Parameter 2: counttrigs: Indicates whether the counts increase with label or only on each trip of the photocell: 1 = photocell and 0 = each label.

Example: **<ESC>GWEB<CR>** will return the current web mode settings.

Related Cmd: SWEB

HDIR printHead DIRectory

Reports the internal controller listing of defined printheads, setup by prior SPHD commands. HDIR, 1, 18, 2000,0 would be the output for a single 18-dot printhead that is two inches from the photocell, and is in the right-to-left print direction.

Group: Action

Format: <ESC>HDIR<CR>

Return Data: HDIR, head_list

Parameter 1: head_list: A list of the defined printheads. For each printhead, the head number, the number of dots, the printhead offset, and the head direction is listed, similar to the SPHD settings.

Default 1: No printheads are defined in the factory default

Example: **<ESC>HDIR<CR>** will return a directory of all heads currently defined.

Related Cmd: GPHD, SPHD, QPHD

LCLS Label Close

Closes and saves the information for the label information that has been received since the LOPN command (refer to the LOPN command).

Group: Advanced Configuration

Format: <ESC>LCLS, Mode, BoxSize, Repeat Distance<CR>

Return Data: None

Parameter 1: Mode: 10 ASCII character mnemonic indicating the storage mode for the open label. (Possible values: PERMANENT = store in nonvolatile memory, NORMAL = store in SRAM)

Example: **<ESC>LCLS,NORMAL,1200,1<CR>** will activate the LFLD and BFLD commands, since the last LOPN command, to be stored in battery-backed memory (SRAM).

Related Cmd: LOPN

LCPY Label CoPY

Copies an identified label.

Group: Label Management

Format: <ESC>LCPY,"label _ name","copy _ name"<CR>

Return Data: N / A

Parameter 1: Label _ name: Name of the existing label.

Parameter 2: copy _ name: Name to be given to the copy of the above label.

Example: **<ESC>LCPY,"DIAGRAPH","DIAGENT"<CR>** will copy the label "DIAGRAPH" to the label "DIAGENT." Both labels will then be available.

Related Cmd: LREN

LDEL Label Delete

Deletes an identified label from the controller.

Group: Action

Format: <ESC>LDEL,"Name"<CR>

Return Data: None

Parameter 1: Name: Twenty-five ASCII character name for label to be deleted. If Name is ?*?, all labels are deleted.

Example: **<ESC>LDEL,An Old Label<CR>** will delete the label "An Old Label" from the controller.

Related Cmd: LREN

LDIR Label DIRectory

Shows a directory of all labels currently stored on the controller

Group: Label Management

Format: <ESC> LDIR <CR>

Return Data: Returns a list of the labels stored on the controller.

Example: **<ESC>LDIR<CR>** will return a directory of all labels currently stored on the controller.

Related Cmd: GLAB

LFLD — Label FieLD definition

Defines a label field.

Group: Basic Configuration

Format: <ESC>LFLD,font,offset,num,L1,[L2,]...[LN,]data<CR>

Return Data: None

Parameter 1: font: Two ASCII digit font ID number.

Parameter 2: offset: Nine ASCII digit number indicating the distance in .001 from the left edge of the box. The distance can extend to 99,999.

Parameter 3: num: Two ASCII digit number indicating the number of print lines that will print the data. The range is 1 to 36.

Parameter 4: LnN: Comma delimited list of two ASCII digit print line number(s) that will print the data. The range is 1 to 36.

Parameter 5: Data: ASCII data to be printed. To allow spaces or punctuation, enclose in double quotes (" ").

Example: **<ESC>LFLD,1,5000,3,1,2,5,"Diagraph Inkjet"<CR>** will use font #1, positioning the text 5 inches from the edge of the box and will print the text "Diagraph Inkjet" on print lines 1, 2 and 5.

LOPN — Label OPen

Starts the process of sending a label to the controller.

Group: Label Management

Format: <ESC>LOPN,"Name"<CR>

Return Data: None

Parameter 1: Name: Twenty-five ASCII character name for label.

Example: **<ESC>LOPN,"A New Label"<CR>** starts sending the fields for the label "A New Label."

Related Cmd: LCLS

LREN — Label REName

Renames a label.

Group: Label Management

Format: <ESC>LREN,"old_label_name","new_label name"<CR>

Return Data: N / A

Parameter 1: Old_label_name: Original label name.

Parameter 2: New_label_ name: New label name.

Example: **<ESC>LREN,"OLDONE","DIAGRAPH"<CR>** will create the label "DIAGRAPH" and delete "OLDONE" from the controller.

PING — PING

Request for information from another node.

Group: Peer to Peer

Format: <ESC>PING,return-address<CR>

Return Data: PONG, node, group, leader

Parameter 1: Return address—node address of current controller

Parameter 2: node—station node number

Parameter 3: group—group id of group containing the current node.

Parameter 4: leader—Leader of the group containing the current node

Example: **<ESC>PING,3<CR>** requests that group information be sent to node 3.

Related Cmd: PONG

PONG PONG

Response to a PING command

Group: Peer to Peer

Format: <ESC> PONG, node, group, leader<CR>

Return Data: N / A

Parameter 1: node - node address of sending node

Parameter 2: group - group id of group to which sending node belongs.

Parameter 3: leader - leader of group specified above.

Example: **<ESC>PONG,3,4,1<CR>** indicates that station 3 is in group 4 with leader 1.

Related Cmd: PING

PRT1 PRinT once (on next box)

Prints the identified label one time

Group: Printing

Format: <ESC>PRT1,"Label_ID"<CR>

Return Data: None

Parameter 1: Label_ID: Twenty-five ASCII character label name to be printed only once.

Example: **<ESC>PRT1,"Apple Juice"<CR>** will print the label "Apple Juice" on the next photocell trip.

Related Cmd: PRTC

PRTC PRinT Continuously

Sets a label to print continuously until an XPRT command stops it. ALOG information is sent automatically with information from previous label.

Group: Printing

Format: <ESC>PRTC,"Label Name"<CR>

Return Data: ALOG, Label Name, Sequence Count, Product Count, User Count 1, User Count 2<CR>

Parameter 1: Label Name: Twenty-five ASCII character label name to be printed continuously.

Example: **<ESC>PRTC,"Apple Juice"<CR>** will print the label "Apple Juice" for all photocell trips until an XPRT is sent.

Related Cmd: PRT1, PWEB, ALOG

PURG PURGe

Directs the indicated channels to print continuously for 3 seconds.

Group: Printing

Format: <ESC>PURG,type, head-number, channel<CR>

Return Data: None

Parameter 1: Type: The type of purge (1-immediate, 2-on photocell)

Parameter 2: head-number: The position, in the daisy chain, of the head to be purged. (0 is all heads)

Parameter 3: channel: The channel number to be purged (0 is all channels).

Example: **<ESC>PURG,2,5,0<CR>** directs all channels on printhead #5 to purge on the next photocell trip.

PWEB Print in WEB mode

Print the named label in web mode.

Group: Printing

Format: <ESC>PWEB, name[,Repeat]<CR>

Return Data: ALOG, Label Name, Sequence Count, Product Count, User Count 1, User Count 2<CR>

Parameter 1: Distance: The distance in thousandths of an inch the box travels from the time the last printhead finished printing to when the first printhead is to start the next print.

Default 1: Distance: Normal box offset.

Parameter 2: Repeat: Optional second parameter indicates the distance from one print to the next.

Example: **<ESC>PWEB,"Distance Mark"<CR>** prints the label "Distance Mark" in web mode.

Related Cmd: PRTC, PRT1, SWEB

QAD1 Query Arcnet Data, Port 1

Returns the last 512 bytes of ARCnet data received by the controller if the ARCnet history has been enabled by a previous SHST host command.

Group: Query

Format: <ESC>QAD1<CR>

Return Data: QAD1, data_stream

Parameter 1: Data_stream: A sequence of characters stored in the port history buffer.

Default 1: HISTORY OFF, if history buffer not enabled and NO DATA if history is enabled but there is no data.

Parameter 2: QAD1, HISTORY OFF is typically returned until an SHST command has been given. From then on, the data_stream is output as received by the controller via the ARCnet port. No filtering is done, so escape character sequences may cause display problems when used.

Example: Enter **<ESC>SHST,ARC1,ON<CR>** to turn on history gathering for the ARCnet port and then enter **<ESC>QAD1<CR>** to send the most recent 512 bytes of data to a device on any other port.

QADR Query Arcnet node adDRess

Returns the ARCnet node address setting of the controller (DIP switch S2). The node address switch setting is read once at controller power-up and stored. Moving the node address switches with power on has no effect until after the next power-on.

Group: Query

Format: <ESC>QADR<CR>

Return Data: QADR, node_addr

Parameter 1: node_addr: a numeric value from 0 to 255. There are 254 possible controller node addresses given that node addresses 0 and 255 are reserved for special uses.

Default 1: Controllers may have any node address except zero and 255, which are reserved for special use. The default is the simply the S2 setting.

Example: **<ESC>QADR<CR>** returns the address of the station.

Related Cmd: QNET

QAS1 Query Arcnet Status (port 1)

Queries the current status of the controller ARCnet port 1. The returned status can be used to troubleshoot network problems.

Group: Query

Format: <ESC>QAS1<CR>

Return Data: QAS1, status_string

Parameter 1: Status_string: An ASCII message string describing the current error status, if any, for the ARCnet port.

Default 1: NO ERROR. Other possible status strings follow:

RX FULL—the ARCnet receive queue is currently full of data. The Host may be sending data faster than the controller can handle it.

TX FULL—the ARCnet transmit queue is currently full of data. The Host is not processing data fast enough.

TX FAIL—an ARCnet transmission was corrupted, possibly by noise.

ARC NAKS—the Host is not processing the ARCnet packets and is sending a NAK back to the controller.

RECON—the ARCnet node connections have been broken and not yet re-established.

Example: **<ESC>QAS1<CR>** will return the status of the ARCnet port.

QBAT Query BATtery Status

Reports the current state of the battery.

Group: Query

Format: <ESC>QBAT<CR>

Return Data: QERR, 37, 0 indicates a low battery. QBAT, OK indicates that the battery is good.

Parameter 1: No parameters.

Example: **<ESC> QBAT<CR>** returns the current state of the battery.

QCMD Query last CoMmanD

Reports the last host command used. Useful for debugging only

Group: Query

Format: <ESC>QCMD<CR>

Return Data: QCMD, comstring, device

Parameter 1: comstring: The last command mnemonic sent to the controller.

Parameter 2: device: The device from which the command originated. (0 = ARCnet, 1 = COM1, 2 = COM2, 3 = COM3 and 4 = COM4)

Example: **<ESC>QCMD<CR>** reports the last command sent to the controller and the port used.

QERR Query ERRor status

Returns one level of previous error status. The QERR codes are sent automatically to the last Host by the controller. In some cases, this asynchronous QERR might be lost. The Host may then issue a QERR and wait for this response.

Group: Query

Format: <ESC>QERR<CR>

Return Data: QERR, primary_code, secondary_code

Parameter 1: primary_code / secondary_code: The numeric error codes used by the controller to indicate error conditions. See error codes in Section 4 of this manual.

Default 1: QERR, 0, 0 is returned when there was no previous error, or the last command was a QERR.

Example: **<ESC>QERR<CR>** returns the previous error codes issued by the controller.

QFON Query FONTS

Queries the controller to ascertain the number of fonts on the controller

Group: Query

Format: <ESC>QFON<CR>

Return Data: QFON, count <CR>

Parameter 1: Count: Number of fonts currently stored on the controller (0-22).

Default 1: 19

Example: **<ESC>QFON<CR>** reports the number of fonts stored on the controller.

Related Cmd: FDIR

QFRM Query FiRMware revision

Requests the version number for the firmware.

Group: Query

Format: <ESC>QFRM<CR>

Return Data: QFRM, version, date <CR>

Parameter 1: version: The version number of the firmware being run on the controller.

Parameter 2: date: The release date of the firmware

Example: **<ESC>QFRM<CR>** reports the version number of the controller's firmware.

QHED Query HEaD

Reports the number of heads currently configured in the controller

Group: Query

Format: <ESC>QHED<CR> number of printheads currently defined

Return Data: QHED, number_of_printheads <CR>

Parameter 1: Number_of_ printheads: Number of configured printheads

Example: **<ESC>QHED<CR>** reports the number of heads currently configured in the controller

Related Cmd: GPHD, SPHD

QLAB Query, Labels

Reports the number of currently defined labels.

Group: Query

Format: <ESC> QLAB <CR>

Return Data: QLAB, count

Parameter 1: Count: Number of labels currently stored on the controller (0 - MAX _ LAB)

Default 1: 0

Example: **<ESC>QLAB<CR>** reports the number of labels currently defined.

QLBL Query LaBeL format

Queries the contents of the internal controller label format for a specified label name. The name of the label must be known from a previous LDIR command. The internal format the controller uses to store a label is subject to change. Note that the double quotes are necessary for label names using spaces or lowercase letters. The exact internal format for a label depends on the LFLD and BFLD commands used during the creation of the label format. Also note that each logical line is dumped individually. The logical lines are dumped last one first.

Group: Query

Format: <ESC>QLBL,"label_name"<CR>

Return Data: QLBL, label_internal_format

Parameter 1: label_internal_format: An dumpof the internal format used by the controller to store the LFLD and BFLD commands that make up a label definition.

Default 1: Dependent on previous LFLD and BFLDs.

Example: **<ESC> QLBL, "HANDLE WITH CARE"<CR>** will return the label in the format described above.

Related Cmd: LDIR, LFLD, BFLD

QLEX Query Label

Queries the controller for the existence of a label.

Group: Query

Format: <ESC> QLEX,"label"<CR>

Return Data: QLEX, exists

Parameter 1: label - name of label in question.

Parameter 2: exists - Logical value indicating whether the label exists (1 = TRUE, 0 = FALSE).

Example: **<ESC>QLEX,"DIAGRAPH"<CR>** will return QLEX, 1 if a label named "DIAGRAPH" is defined in the controller.

QLIN Query number of logical LINes

Queries the controller for the number of logical lines.

Group: Query

Format: <ESC>QLIN<CR>

Return Data: QLIN, number_of_lines <CR>

Parameter 1: Number_of_lines: number of defined logical lines.

Example: **<ESC>QLIN<CR>** will return the number of currently defined logical lines.

Related Cmd: GLGL

QLOG Query LOG

Provides a print summary that includes the last label printed, the sequence count, the product count, number of pallets, pallet count, user count 1 and user count 2.

Group: Query

Format: <ESC>QLOG<CR>

Return Data: QLOG, lastlabel, seq, prod, pallets, palcnt, user1, user2

Parameter 1: lastlabel: The last label printed by the controller.

Parameter 2: seq: The current value of the sequence count.

Parameter 3: prod: The current value of the product count.

Parameter 4: pallets: The number of pallets of product printed thus far.

Parameter 5: palcnt: The number of items on the current pallet.

Parameter 6: user1 and user2: The values of the user definable counts.

Example: **<ESC>QLOG<CR>** returns information for a log entry.

QMAX Query MAXimum Line Speed

Returns the maximum line speed at which the label currently selected by a print command (PRTC / PRT1 / PWEB) can be printed. In general, the more printheads used, the lower the maximum line speed. Note: If the line speed exceeds the maximum, the controller will issue a QERR,41,0.

Group: Query

Format: <ESC>QMAX<CR>

Return Data: QMAX, line_speed

Parameter 1: line_speed: the maximum line speed in FPM that the controller can handle for a given label and printhead configuration

Default 1: 1500 is returned until a valid print command has been issued.

Example: **<ESC>QMAX<CR>** returns the maximum line speed for the current label.

Related Cmd: GLNS, SLNS.

QMEM Query MEMory

Reports the free space in static and dynamic RAM

Group: Query

Format: <ESC>QMEM<CR>

Return Data: QMEM, totsram, largesram, totdram, largedram

Parameter 1: totsram: The total amount of SRAM available

Parameter 2: largesram: The size of the largest block of SRAM available.

Parameter 3: totdram: The total amount of DRAM available.

Parameter 4: largedram: The size of the largest block of DRAM available.

Example: **<ESC>QMEM<CR>** will return information on memory usage.

QNET Query NETwork map

Requests the current network map

Group: Query

Format: <ESC> QNET<CR>

Return Data: QNET, netid0,

Parameter 1: netid1,...netidn <CR>netid1 - netidn - notes addresses of rest of current network map

Example: **esc>QNET<CR>** will return the current network map

QNXT Query NeXT

Returns the next station in the network

Group: Peer to Peer

Format: <ESC> QNXT

Return Data: QNXT next _ id

Parameter 1: next_id = ARCnet address of next station in the network.

Example: **<ESC>QNXT<CR>** returns the id of the next station in the network.

QPHD Query all Print HeaDs

Retrieves information on each printhead currently defined in the controller

Group: Query

Format: <ESC>QPHD<CR>

Return Data: QPHD, number, dots, offset, direction <CR>

Parameter 1: number -- The electrical position of the head in the daisy chain.

Parameter 2: dots -- Number of channels in the head

Parameter 3: offset -- Offset of the head from the photocell in 0.001 inch.

Parameter 4: direction -- Direction of printing
0 - substrate coming from the right
1- substrate coming from the left

Example: **<ESC>QPHD<CR>** returns the information on all heads currently defined in the controller.

Related Cmd: GPHD.

QPLN Query all Print LiNes

The above is returned for each print line defined in the controller.

Group: Query

Format: <ESC>QPLN<CR>

Return Data: QPLN, number,start:end<CR>

Parameter 1: number—The identifying number for the print line

Parameter 2: start—Dot at which the current print line starts.

Parameter 3: end—Last dot printed by this print line.

Example: **<ESC>QPLN<CR>** will return the information on all print lines currently defined.

Related Cmd: GLGL, SLGL.

QPRT Query PRinTing

Enables the host command software to detect the print mode in use by the controller. This detection allows the host PC to be turned on or off without disrupting the printing underway with a network of controllers.

Group: Query

Format: <ESC>QPRT<CR>

Return Data: QPRT, MODE <CR>

Parameter 1: MODE: ASCII number indicating the current print mode. (Can be 0-not printing, 1-PRT1, 2-PRTC, 3-PWEB)

Example: **<ESC>QPRT<CR>** returns the current print mode.

Related Cmd: QST1, QST2.

QPSM Query Print State Machine

Returns current state of the internal print state machine.

Group: Query

Format: <ESC>QPSM<CR>

Return Data: QPSM, state<CR>

Parameter 1: State: Current state of print state machine.

0: Idle	3. Format Complete
1: PrintCmd. Rcvd.	4. Offset delay complete
2: Photocell tripped	5. Squirting ink

Example: **<ESC>QPSM<CR>** will return the current state for the print state machine.

QSDx Query Serial Data port x

Reports the history of an identified serial port. Command range is QSD1, QSD2, QSD3, QSD4, QAD1. To query a serial port for history, the history must first be activated with the SHST command which clears the data history for the specified port after the data has been presented. If the data history is disabled, then a message of "History Off" will be sent.

Group: Query

Format: <ESC>QSDx<CR>

Return Data: QSDn, Data1[,Data2] [,Data3] . . . [,DataN]<CR>

Parameter 1: DataN: ASCII character string or strings separated by commas that represent the history of data received by the specified port. Each string will represent one message or command. Any binary data will be filtered for display.

Example: **<ESC>QSD4<CR>** returns the history of serial port #4.

QSFT Query ShiFT Setup

Returns the internal list of shift start times kept by the controller. This list is sorted and reported in a fixed order starting with the first shift after midnight.

Group: Query

Format: <ESC>QSFT<CR>

Return Data: QSFT, list_of_shifts

Parameter 1: list_of_shifts: a listing of the shifts, with each shift consisting of two fields. First, the shift ordinal number (used with the {SN} autocode). Then the shift start time in HH:MM format. Note: the ordinal number is the original order of the shift start times given in the previous SSFT host command. Shift numbering is always sequential but ANY shift can be number 1. At some site locations, the first shift of the day may not be shift 1, but a carry-over from a previous day.

Default 1: The factory defaults:

03,00:00 (midnight)
01,08:00 (8 AM)
02,16:00 (4 PM)

Example: **<ESC>QSFT<CR>** will return the current shift configuration

---

QSSx Query Serial Port Status

Query the status of a serial port

Group: Query

Format: <ESC>QSSn<CR>

Return Data: QSSn, Status1, ... ,StatusN<CR>

Parameter 1: STatusN: Eight ASCII character mnemonic indicating the current status of the port.
RXFULL: Receive buffer full
TXFULL: Transmit buffer full

Parameter 2: TXFAIL: Too many transmit retries
OVERRUN: Receive data overrun
NOISE: Noise data errors
FRAMING: Framing data errors

Parameter 3: PARITY: Parity data errors
ARCNACKS: Excessive NAKs received
RECON: Arcnet reconfigure activated
RXHIMARK: Receive buffer almost full

Parameter 4: NO DATA: No data in receive buffer
NO ERROR: No errors

Example: **<ESC>QSS1<CR>** will return the current status of a serial port.

QST1 Query STatus level 1

Reports the top level status of a station. Mode of printing same as QPRT command: 0 = Not printing; 1 = PRT1; 2 = PRTC; 3 = PWEB.

Group: Query

Format: <ESC>QST1<CR>

Return Data: QST1, PrintMode, LabelName, ProdCount<CR>.

Parameter 1: PrintMode: Twenty-five character label format name if PrintMode is not zero. If PrintMode is zero, then LabelName is empty and nothing appears between the delimiting commas.

Parameter 2: LabelName: 25 ASCII character name of the label the station is currently printing.

Parameter 3: ProdCount: Current value of product count.

Example: **<ESC>QST1<CR>** will return the top level (system level) status of the station.

Related Cmd: QPRT, QST2, QST3.

QST2 Query STatus level 2

Mode of printing same as QPRT command: 0 = Not printing; 1 = PRT1; 2 = PRTC; 3 = PWEB.

Group: Query

Format: <ESC>GST2<CR>

Return Data: QST2, PrintMode,"LabelName", SeqCount, LineSpeed, PPM, SFC, PRD, PAL, UC1, UC2, PSC, MBX, Lines [,Line1[,Line2[...[,LineN]...]]]<CR>

Parameter 1: PrintMode: Twenty-five character label format name if PrintMode is not zero. If PrintMode is zero, then LabelName is empty and nothing appears between the delimiting commas.

Default 1: Parameters Continued
UC1: Current value of user count 1.

Parameter 2: SeqCount: Current value of sequence count.

Default 2: UC2: Current value of user count 2.

Parameter 3: LineSpeed: Line speed in feet per minute.

Default 3: PSC: Previous shift count.

Parameter 4: PPM: Prints per minute

Default 4: MBX: Current missing box count.

Parameter 5: SFC: Current shift count.

Default 5: Lines: Number of print lines in format.

Parameter 6: PRD: Current product count.

Default 6: Line1-N: Optional lines of actual printed data. Each line is to be enclosed in double quotes. Lines are delimited with a single comma. Lines with no data are returned as two double quotes ().

Parameter 7: PAL: Current count of pallets.

Example: **<ESC>QST2<CR>** returns detailed status.

Related Cmd: QPRT, QST1, QST3

QST3 Query STatus

Returns status information including verification data from a bar code decoder for up to two scans.

Group: Query

Format: <ESC> QST3 <CR>

Return Data: QST3, mode,"label name", prod, V1Good, V1Total, V1Data, V2Good, V2Total, V2Data

Parameter 1: mode: print mode
0 Not printing
4 Printing (PWFB)
1 Armed by PRT1, PRTC or PWEB
2 Printing Printing (PRT1)

Parameter 2: label name: Name of label printed.

Parameter 3: prod: Current value of the product count.

Parameter 4: V1Good: Number of good reads from bar code verifier scan head #1.

Parameter 5: V1Total: Number of reads or attempted reads made by bar code verifier scan head #1.

Parameter 6: V1Data: String containing last data from bar code verifier scan head 1.

Parameter 7: V2Good, V2Total V2Data on the same as their V1 counterparts, except they refer to verifier scan head #2.

Example: **<ESC>QST3<CR>** will return the current system level status information.

Related Cmd: QPRT, QST1, QST2.

QST4 Query Status 4

Queries for station status including verifier data.

Group: Query

Format: <ESC>QST4<CR>

Return Data: QST4, mode "label name", seq, VIGood, VQTotal, V1Data, V2Good, V2Total, V2Data, line speed, ppm, shift, prod, pallet, user1, user 2, pshift, mbox, N, Line1,..., Line N

Parameter 1: label name: most recent label printed.

Parameter 2: seq: sequence count value.

Parameter 3: V1Good: Number of good reads from bar code verifier scan head 1.
V1Total: Number of total reads from bar code verifier scan head 1.
V1 Data: Quoted string containing the last bar code scanned by verifier head 1.

Parameter 2: V2Good: Number of good reads from bar code verifier scan head 2.
V2Total: Number of total reads from bar code verifier scan head 2.
V2Data: Quoted string containing the last bar code scanned by verifier scan head 2.

Parameter 3: line speed: Current line speed in feet / min.

Parameter 4: ppm: prints per minute.
shift: shift count value.
prod: product count value.
pallet: pallet count value.

Parameter 5: user1: User definable count #1 value.
user2: User definable count #2 value.
Pshift: Previous shift count value.
mbox: Missing box count.

Parameter 6: N: number of label lines to follow.
Linex: Text for Line x of formatted label.

Example: **<ESC>QST4<CR>** will report verifier status information.

Related Cmd: QST1, QST2, QST3.

QSTS Query STation Status

Query for station status

Group: Query

Format: <ESC>QSTS, option<CR>

Return Data: QSTS, Label Name, Number of Lines, Line Speed, Prints / min, Line1<CR>...LineN<CR>

Parameter 1: Label Name: name of the label last printed. (In return value only)

Parameter 2: Number of Lines: number of lines in the label last printed.(In return value only)

Parameter 3: Line Speed: Speed of conveyor in feet per minute. (In return value only)

Parameter 4: Prints / min: Number labels printed in the last minute. (In return value only)

Parameter 5: LineN: Line number N of last label printed (returned only if option is 1)

Parameter 6: option: Indicates whether a long or short form of the returned values is returned. (0--short, 1--long)

Example: **<ESC>QSTS,1<CR>** will return the long version of the return format above.

Related Cmd: GLNS, QPRT, QST1.

RDEF Return to DEFault settings

Returns all settings to factory-default states. Because of the limited erase / program cycles allowed on the Flash memory, USE THIS COMMAND ONLY AS AN EXTREME LAST RESORT.

Group: Action

Format: <ESC>RDEF<CR>

Return Data: None

Parameter 1: None

Example: **<ESC>RDEF<CR>** will reset the controller to factory default.

CAUTION: Follow this command by powering the controller OFF then ON again or use the boot command.

Related Cmd: BOOT

SBCD Set Bar CoDe

Defines a bar code. If the Mode is set to Constant Length, then Readlens, Start and Length parameters are not necessary

Group: Advanced Configuration

Format: <ESC>SBCD, Start, Mode, Readlen, Start, Len<CR>

Return Data: Start:

Parameter 1: Start, Len<CR> Character that will precede bar code data. This is the decimal equivalent of the ASCII code. The range is 0-255

Default 1: Start: 2(STX)

Parameter 2: Mode: Set to V to variable length bar code. Set to C for a constant length bar code.

Default 2: Mode: C

Parameter 3: Readlen: Length of the bar code string from the bar code scanner in bytes.

Default 3: Readlen: 8

Parameter 4: Start: Position of the first byte to be used. Any bytes before this position or after position Start+Len will be ignored.

Default 4: Start: 1

Parameter 5: Len: Length of the portion of the string that is used.

Default 5: Len: 6

Example: **<ESC>SBCD,2,C,18,1,5<CR>** sets the bar code message to be 18 characters long, constant length and sends characters 1-5 to the host.

SBOX Set BOX width

Provides the controller with the box width information. It performs a calculation formerly done by the label designer in earlier versions.

Group: Basic Configuration

Format: <ESC>SBOX, width<CR>

Return Data: None

Parameter 1: width: Width of the boxes in thousandths of an inch

Example: **<ESC>SBOX,14125<CR>** will set the box length to 14.125 inches.

Related Cmd: GBOX.

SDAT Set DATe for controller

Sets the date on the controller.

Group: Action

Format: <ESC>SDAT, Date<CR>

Return Data: None

Parameter 1: Date: Eight ASCII character string representing the current date in format, DD:MM:YY.

Default 1: Date: Current date as stored on the controller.

Example: **<ESC>SDAT,19:05:96<CR>** will set the date to May 19, 1996

Related Cmd: GDAT.

SDRT Set Date Rollover Time

Sets the time when the date rolls over to another date.

Group: Action

Format: <ESC> SDRT, hh:mm<CR>

Return Data: N / A

Parameter 1: hh - hour of time that date rollover should occur.

Parameter 2: mm - minutes of time date rollover should occur.

Example: **<ESC>SDRT,13:00<CR>** sets the time at which the date advances to 1:00 PM.

Related Cmd: GDRT.

SDTP Set DoT Pitch

Sets the minimum distance from one dot to the next in thousandths of an inch.

Group: Advanced Configuration

Format: <ESC>SDTP, Dotpitch<CR>

Return Data: None

Parameter 1: Dotpitch: Dot pitch in thousandths of an inch.

Default 1: Dotpitch: 10

Example: **<ESC>SDTP,10<CR>** sets the distance between dots to 0.010

Related Cmd: GDTP

SENC Set ENCoder resolution

Sets the resolution of the encoder to a specific number of ticks per inch.

Group: Advanced Configuration

Format: <ESC>SENC, Resolution<CR>

Return Data: None

Parameter 1: Resolution: Nine ASCII digit representation of the actual encoder resolution in ticks per inch. This number MUST reflect the hardware multiplier and is always in English units.

Default 1: Resolution: 1000 ticks per inch.

Example: **<ESC>SENC,200<CR>** will set the controller for an encoder with 100 ticks per inch and a hardware multiplier of 2.

SFMT Set ForMaT

Sets the formatting mode

Group: Advanced Configuration

Format: <ESC>SFMT mode <CR>

Return Data: N / A

Parameter 1: Mode: 1 or 0 indicating whether the image of the message is created before or at photocell (1 - pre-format, 0 - post-format).

Default 1: 0

Example: **<ESC>SFMT,1<CR>** enables the pre-format feature.

Related Cmd: GFMT

SGST Set Global STring

Sets the indicated global string to the provided string.

Group: Advanced Configuration

Format: <ESC> SGST, ID, String<CR>

Return Data: None

Parameter 1: ID: Two ASCII digit number representing the global string identifier. Valid range 1 to 10.

Default 1: ID: 1

Parameter 2: String: Twenty-five ASCII character string. If the string length is greater than 25 characters, the string will be truncated. Strings shorter than 25 characters will NOT be padded to 25 characters.

Default 2: String: All ten global strings are ????.

Example: **<ESC>SGST,1,"Diagraph"<CR>** will set the first global string to "Diagraph".

Related Cmd: GGST

SHMI Set Horizontal Motion Index

Sets the number of dots between characters.

Group: Advanced Configuration

Format: <ESC> SHMI, Index<CR>

Return Data: None

Parameter 1: Index: Nine ASCII digit number indicating the number of dots between character columns when printing fixed pitch fonts. Spacing is affected by the index value for all printable ASCII characters. This index is ignored on proportional fonts except for their space characters.

Default 1: Index: 0.

Example: **<ESC>SHMI,3<CR>** ensures there are 3 dots between every character.

SHST Set HiSTory

Activates history-gathering for an identified port. Enabling a data history collects a list of data items. This list is reset after a query data command--QSDn or QADn--and is deleted when the data history is disabled.

Group: Advanced Configuration

Format: <ESC> SHST, Port ID, Flag<CR>

Return Data: None.

Parameter 1: Port ID: Four ASCII character mnemonic representing the port. SP1: Serial Port 1 SP4: Serial Port 4
SP2: Serial Port 2 ARC1: ARCnet Port 1
SP3: Serial Port 3 PP1: Parallel Port 1

Default 1: Flag: All data histories are disabled, OFF

Parameter 2: Flag: Three ASCII character mnemonic indicating status of data history for the specified port.
ON: Data history enabled
OFF: Data history disabled

Example: **<ESC>SHST,ARC1,ON<CR>** will turn the history-gathering on for ARCnet port #1.

Related Cmd: QSDn, QADn

SLGL Set LoGical Line

Maps logical lines to dot ranges.

Group: Advanced Configuration

Format: <ESC>SLGL, Line, S1 :E1[,S2 :E2] . . . [,SN :EN]<CR>

Return Data: None

Parameter 1: Line: Two ASCII digit logical line number. Range is 1 to 36.

Default 1: The default logical lines are a one-to-one mapping of printheads into logical lines. The default logical lines will only be created after a set printhead configuration command is properly issued.

Parameter 2: Sn: Nine ASCII digit number representing the starting dot number of the specified logical line. This number must be followed by a colon.

Parameter 3: En: Nine ASCII digit number representing the ending dot number of the specified logical line.

Example: **<ESC>SLGL,4,19:23<CR>** will associate print line 4 with dots 19-23

Related Cmd: GLGL, QLIN, QPLN.

SLNS Set Line Speed

Sets the line speed

Group: Basic Configuration

Format: <ESC>SLNS,Speed<CR>

Return Data: None

Parameter 1: Speed: Nine ASCII digit representation of the line speed in feet per minute or in tenths of meters per minute.

Default 1: Speed: 0

Example: **<ESC>SLNS,150<CR>** will set a simulation of 150 feet per minute in the controller. If the speed is zero, the system will determine speed from the shaft encoder inputs. Any non-zero value results in the controller simulating the shaft encoder output to achieve the specified speed. Setting the line speed to any non-zero value will reset the delta value used by the line speed command.

SMBX Set Missing BoX count

Sets the missing box count.

Group: Advanced Configuration

Format: <ESC> SMBX, newvalue <CR>

Return Data: None

Parameter 1: newvalue: The new value that the box count is to be set to

Example: **<ESC>SMBX,0<CR>** will adjust the missing box count to 0.

SMRG Set image MeRGe

Sets image merging. It is used in the PRTC mode only.

Group: Advanced Configuration

Format: <ESC>SMRG,Setting<CR>

Return Data: None

Parameter 1: 0 = merging is disabled
1 = merging enabled

Default 1: 1 for enabled merging.

Example: **<ESC>SMRG,1<CR>** will enable image merging.

Related Cmd: GMRG

SPAL Set PALlet count

Sets the pallet counter and changes the parameters and values for the counters related to the pallet counter.

Group: Action

Format: <ESC> SPAL, Itemcount, Palcount,Size<CR>

Return Data: None

Parameter 1: Itemcount: Nine ASCII digits representing the number of items that were printed on the current pallet

Default 1: Itemcount: 0.

Parameter 2: Palcount: Nine ASCII digits representing the number of completed pallets.

Default 2: Palcount: 0.

Parameter 3: Size: Nine ASCII digits representing the number of items per pallet.

Default 3: Size: 999999999.

Example: **<ESC>SPAL,0,0,25<CR>** sets the pallet count and item count to 0 and the size of a pallet to 25.

SPCI Set Photocell Inhibit

Sets a distance during which the controller will ignore photocell trips.

Group: Advanced Configuration

Format: <ESC> SPCI, mode, distance<CR>

Return Data: N / A

Parameter 1: mode - indicates whether photocell inhibit is to be enabled. 0 - disabled. 1 - enabled.

Parameter 2: distance - Distance in thousandths of an inch after a photocell trip that additional trips are ignored.

Example: **<ESC>SPCI1,14000<CR>** tells the controller to ignore all photocell trips for 14 inches (after the first).

SPHD Set PrintHead Configuration

Sets the parameters for a specific printhead.

Group: Basic Configuration

Format: <ESC> SPHD, Resolution, Offset, Direction,Position<CR>

Return Data: None

Parameter 1: Resolution: Nine ASCII digit number of the number of vertical dots on the printhead.

Default 1: Resolution: 24 dots.

Parameter 2: Offset: Nine ASCII digit number of the offset in the X direction given in thousandths of inches. Range 0 to 99,999 thousandths of inches.

Default 2: Offset: 0.0 inches.

Parameter 3: Direction: Nine ASCII digit number of printing direction. Printing with the line direction is indicated by zero, opposite the direction by non-zero.

Default 3: Direction: 0

Parameter 4: Position: Nine ASCII digit number of printhead position as wired into the printhead daisy chain. Up to 32 printheads are addressable.

Default 4: Position: 1

Example: **<ESC>SPHD,9,11250,1,5<CR>** will set the fifth printhead to be 9 dots high, 11.25 inches from the photocell and to print in the opposite direction from the line movement.

Related Cmd: GPHD, QHED, QPHD

SPRD Set PRoDuct count

Modifies the value of the product count.

Group: Advanced Configuration

Format: <ESC> SPRD, Count<CR>

Return Data: None

Parameter 1: Count: Nine ASCII digits representing the number of items printed.

Default 1: Count: 0

Example: **<ESC>SPRD,0<CR>** will set the value of the product count to 0.

SRPX Set RPX Settings

Sets the RPX page boundaries. The RPX memory space is divided into two areas: a font page area, and an image page area. This command allows the operator to select which of three possible page divisions is used depending on the S1 switch settings, the default size of the imgae will be either 1MB or 3MB, but it will start at 0xB00000 regardless. This command can be used to increase the amount of memory used for either fonts or image printing as the application needs dictate.

Group: Advanced Configuration

Format: <ESC>SRPX, mode<CR>

Return Data: None.

Parameter 1: Mode: A numeric value between 0 and 2
0: Font at 0xA00000, Image at 0xB00000
1: Font at 0xA00000, Image at 0xC00000
2: Font at 0xA00000, Image at 0xD00000

Default 1: Factory default: 0

Example: **<ESC>SRPX,1<CR>** will increase the amount of memory available for fonts.

SSCL Set SCaLe

Sets the parameters of the message from the scale

Group: Advanced Configuration

Format: <ESC>SSCL,StartChar,Mode,Readlen,Start,Len<CR>

Return Data: None

Parameter 1: StartChar: Character that precedes the scale data. This is the decimal equivalent of the ASCII code. The range is 0-255.

Default 1: StartChar: 1 (SOH)

Parameter 2: Mode: Flag that indicates whether the scale data is variable (V) or constant (C) length.

Default 2: Mode: C

Parameter 3: Readlen: The length of the string received from the scale in bytes.

Default 3: Readlen: 18

Parameter 4: Start: Position of the first byte to be used. Any bytes before this position or after position Start+Len will be ignored.

Defaul t 4: Start: 1

Parameter 5: Len: Length of the portion of the string that is used.

Default 5: Len: 6

Example: **<ESC>SSCL,18,1,5<CR>** will set up for a scale that returns 18 characters, the first 5 of which are significant.

SSEQ Set SEQuence count

Sets the value and modulus of the sequence count

Group: Advanced Configuration

Format: <ESC> SSEQ, Itemcount, Modulus<CR>

Return Data: None

Parameter 1: Itemcount: Nine ASCII digits representing the number of items printed.

Default 1: Count: 0.

Parameter 2: Modulus: Nine ASCII digits representing the wraparound value. When Count reaches Modulus, the sequence count wraps back to 1; unless Modulus is 0, in which case it wraps to 0 when it overflows from all 9's.

Default 2: Modulus: 999999999.

Example: **<ESC>SSEQ,1,999999<CR>** sets the count value to 1 and the rollover value to 999,999.

SSFC Set ShiFt Count

Set the value of the shift count

Group: Advanced Configuration

Format: <ESC>SSFC,Count<CR>

Return Data: None

Parameter 1: Count: Nine ASCII digits representing the number of items printed during the current shift

Default 1: Count: 0.

Example: **<ESC>SSHFT,0<CR>** sets the shift counter to 0.

SSFT Set ShiFt Time

Sets the shift time with a maximum of 24 shifts.

Group: Advanced Configuration

Format: <ESC>SSFT,Time1, ..., Timen<CR>

Return Data: None

Parameter 1: TimeN: Six ASCII character string representing the starting shift time in 24 hour format, HH:MM:SS.

Default 1: Time: 00:00,08:00,16:00

Example: **<ESC>SSFT,03:00,15:00<CR>** sets two shifts, one starting at 3:00 AM and a second starting at 3:00 PM.

SSPx Set Serial Port x Configuration

Configures the specified serial port from the range SSP1, SSP2, SSP3 and SSP4

Group: Advanced Configuration

Format: <ESC> SSPn, Use Code, Device Code, Baud Rate, Parity, Data Bits, Stop Bits, Handshake<CR>

Return Data: None

Parameter 1: Use code: Four character ASCII mnemonic for connected serial device:
INK for Ink supply
BARC for Bar code scanner
SCAL for Scale
HOST for Host computer
HAND for Hand-held computer
TERM for Standard terminal type
NONE for Port disabled.

Default 1: Use code: TERM = Standard terminal type.

Parameter 2: Device code: Four ASCII character mnemonic that further describes the Use code.

Default 2: Device code: DUMB - Dumb terminal.

Parameter 3: Baud Rate: Nine ASCII digit number indicating the baud rate of the serial port: 50, 110, 200, 300, 600, 1200, 2400, 4800, 7200, 9600, or 38400 baud.

Default 3: Baud Rate: 9600 baud

Parameter 4: Parity: Four ASCII character mnemonic indicating type of parity implemented:
NONE: No parity implemented;
EVEN: Even parity implemented;
ODD: Odd parity implemented

Default 4: Parity: NONE: No parity implemented

Parameter 5: Data Bits: Nine ASCII digit number indicating the number of data bits. 7 or 8 data bits are allowed.

Default 5: Data Bits: 8

Parameter 6: Stop Bits: Nine ASCII digit number indicating the number of stop bits. 1 or 2 stop bits are allowed.

Default 6: Stop Bits: 1

Parameter 7: Handshake: Four ASCII character mnemonic indicating the type of hardware handshaking to be used:
NONE: No DTR, RTS, or CTS handshaking;

DTR: DTR only handshaking;
FULL: DTR, RTS, and CTS handshaking

Default 7: Handshake: NONE: No DTR, RTS, or CTS handshaking.

Example: **<ESC>SSP2,TERM,DUMB,9600,NONE,8,1,NONE<CR>** will setup serial port 2 to accept input from a dumb terminal at 9600 baud with 8 data bits, 1 stop bit, no parity and no handshake.

STIM Set TIMe for controller

Sets the time on the controller.

Group: Action

Format: <ESC> STIM, Time<CR>

Return Data: None

Parameter 1: Time: Eight ASCII character string representing the current time in 24 hour format, HH:MM:SS.

Default 1: Time: Time as currently stored on the controller.

Example: **<ESC>STIM,13:45:00<CR>** will set the time to 1:45 PM.

SUCx Set User Count 1, 2

Sets the value, direction, delta and modulus

Group: Advanced Configuration

Format: <ESC> SUCn, Val, Sign, Delta, Modulus<CR>

Return Data: None

Parameter 1: Val: Nine ASCII digits representing the number of user definable items that were printed.

Default 1: Val: 0

Parameter 2: Sign: + for increment by Delta, - for decrement.

Default 2: Sign: +.

Parameter 3: Delta: Nine ASCII digits representing the amount by which to increment or decrement after each item.

Default 3: Delta: 1

Parameter 4: Modulus: Nine ASCII digits representing the count at which to wrap around

Default 4: Modulus: 999999999.

Example: **<ESC>SUC1,5000,-,5,5000<CR>** will set up user counter 1 with a value of 5000 as a down counter which will rollover to 5000 when it reaches 0.

SWEB Set Web Parameters

Sets the web mode attributes.

Group: Advanced Configuration

Format: <ESC>SWEB, , <CR>

Return Data: None

TLNS Tune/Tweak Line Speed

Fine-tunes the line speed, as seen by the controller, in increments of 1% of 1 foot / min.

Group: Advanced Configuration

Format: <ESC> TLNS, Delta<CR>

Return Data: None

Parameter 1: Delta: Signed, Two ASCII digit number representing the percentage of one line speed unit to add to the current line speed value. Units are feet per minute in English units and tenths of meters per minute in metric units. This command is only valid when

Default 1: Delta: 0

Parameter 2: Control Resolution:
English Units: 1 foot / min = .2 in / sec
1% of 1 foot / min => .002 in / sec
99% of 1 foot / min => .198 in / sec
Metric Units: .1 meter / min = .167 cm / sec
1% of .1 meter / min => .00167 cm / sec
99% of .1 meter / m

Example: **<ESC>TLNS,+2 <CR>** will increase the line speed by 2% of 1 ft. / min.

XPRT Cancel Printing

Cancels printing.

Group: Printing

Format: <ESC>XPRT<CR>

Return Data: ALOG, last label, sequence, product, pallets, palcnt, user 1, user 2<CR>.

Parameter 1: NONE

Example: **<ESC>XPRT<CR>** will cancel printing on the current station.

Related Cmd: PURG, PRTC, PWEB

CHAPTER 4
SERIES 2 AUTOCODES

Autocodes are codes enclosed by braces ({ }) that automatically add the date, time or other variable information to the label text. Some of the autocodes print their message in a conventional form, such as the date in "mm/dd/yy" format or the time in "hh:mm" format—the month as a single letter A-L or the hour as a single letter A-X. These codes can track production and inventory.

To combine text with an autocode on a label message line, enter your text before, between or after the braces. For example, **{D} #{N}** will print 04/12/96 #232. Leave a space where desired to keep the message from running together.

Many of the autocodes have an optional offset value that is added to the current day. This added value is counted in days and the result can specify an expiration date. It appears in an autocode as <offset>. If you enter no value, it will default to zero. Be sure to enter a space between the alpha character of the autocode and the offset numeric: **{J 5}**.

{@ <macronum> <maxlen>}	Invokes the macro known by <macronum>; <maxlen> specifies the maximum length of the resulting string. The returning string is then copied into the output string. Both <macronum> and <maxlen> must be specified.
{A <offset>}	Prints the day of the month as two digits (01 through 31). The offset value is optional.
{B <offset>}	Prints the day of the month as a single character (1 through 9 then A through W). The offset value is optional.
{C}	Prints the current time in minutes as two digits (00 through 59).
{D <offset>}	Prints the full date in the conventional format of MM/DD/YY. The offset value is optional.
{E}	Prints the week of the year as two digits (01-52).
{G}	Prints the hour of the day as a single letter. A is midnight and X is eleven PM.
{H}	Prints the hour of the day as two digits (00 through 23).
{I <offset>}	Prints the numerical day of the year, the Julian date as two letters--AA through OB. The offset value is optional.
{J <offset>}	Prints the numerical day of the year, the Julian date as three digits--001 through 366. The offset value is optional.
{L <offset>}	Prints the month of the year as a single letter (A through L). The offset value is optional.
{M <offset>}	Prints the month of the year as two digits (01 through 12). The offset value is optional.
{N}	Prints each item sequentially as it passes the print station. When the counter reaches the maximum number, 999999, it resets and starts again with 1.

{O <offset>}	Prints the month of the year as a three character abbreviation (JAN through DEC). The offset value is optional.
{PC}	Prints the pallet count as defined in the label definition screen. The maximum value is 999,999,999.
{PRD}	Prints the product count as nine digits
{Q}	Prints the time in fifteen minute intervals. The number of fifteen minute intervals since midnight is printed as a two digit number ranging from 00 to 95.
{R <offset>}	Prints the last single digit of the current year (0 through 9). The offset value is optional.
{SC}	Prints the number of products (labels) printed on the current shift.
{SEC}	Prints seconds as two digits (00 through 59).
{SN <numalpha>}	Prints the shift number. If <numalpha> is 1, the shift number will print as a single number starting with 1; if <numaplha> is 2, the shift number will print as a single letter starting with A. <numalpha> defaults to 1 if not specified.
{SPD}	Prints the line speed as three digits (000-999).
{STR <globstrnum>}	Prints the global string identified by <globstrum>. <globstrum> must be specified.
{T}	Prints the current time in a twenty-four hour format (00:00 though 23:59).
{TP}	Prints a test pattern of vertical dots the height of the logical line.
{USR <usrnum>}	Prints either user defined Count #1 or Count #2. <usernum> must be entered as either 1 or 2 to define the User Count to report.
{Vab}	Prints the hour of the day as a single letter, A through Z. a and b are user-defined and can be omitted leaving 24 letters, on for each hour.
{W}	Prints the current scale data and does not print if no data is available.
{X <option>}	Prints the day of the week as either a single digit (1-7) with option 1 or as a single letter (A-G) with option 2. With option 1, Monday would print as “1” and with option 2, Monday would print as "A".
{Y <offset>}	Prints the current year as two digits (00 through 99). The offset value is optional.
{Z}{ }	Prints either scale or scanner data—data received with an SOH or STX preamble.

SEQUENCE COUNT {N}

The {N} autocode enables you to print sequential numbers on products and have those numbers reset when they reach a wrap value. For example, {N} with a wrap value of 000 will return to 000 when the sequence count reaches 999 or will return to 0001 with a wrap value of 3456 when it reaches 3456.

To set an upper limit sequence number, include the upper limit immediately after the N inside the braces. For example, {N999} would use the numbers 001-999. With an upper limit assigned, each number contains the same number of digits as the upper limit. For example, 001 will be the three digit initial entry in a sequence with 999 as the upper limit. Without an upper limit, the {N} autocode does not include zeros.

To print the sequence numbers as 000 to 999, enter the autocode {N000}. Using the appropriate number of zeros in the upper limit forces a print with the same number of nines. For example, {N00000} creates the sequence 00000 to 99999.

The label "Fancy Green Beans" with a time, date and sequence number (up to 999999) could be entered as the 25-character line **"Fancy Green Beans {T D N}".** When printed, the label would contain the following 37 characters: "Fancy Green Beans 12:20 04/12/96 1289".

PALLET COUNT CODE {PC}

The pallet count increases by one every time the item count reaches the preset limit.

{PC} is the basic command to print a pallet count. **PC** must be followed by one to six digits that specify the number of items per pallet. If you enter {PC25}, the target item will be printed with the pallet count "2". To include an item count in the code, use the autocode {PI}. When in a message, it will print the item number within that pallet. When both pallet count and item count codes are used, they must both be followed by the same limit value—{PC25} {PI25}.

The pallet count and the item count can each have six digits. Note that the upper limit set in the pallet code is the number of items within the pallet, not the number of pallets to be marked. The number of items within the pallet sets the benchmark for the pallet count. If the pallet count is set at 25, the first pallet has 25 items and the pallet count will start at "2" for the next pallet of 25 items.

If the system is reset or an item is discarded from production which will offset the item number, the count must be reset to the proper value through the pallet count.

WEIGHT CODE {W}

Use the weight code {W} to insert any value into your message via the serial port from an external device such as a scanner or a scale.

VARIABLE DATA CODE {Z}{}

Code {Z} performs a function similar to {W}: it inserts data into a message from a collection device such as a scale or scanner. An example of the syntax of a {Z} code entry follows.

A message with fixed text, variable data and a date printed with autocodes would have five sets of braces:

"COSMIC {Z}{}{M}{A}{Y}"

COSMIC	constant text string
{Z}	reports the variable data
{ }	open and close braces that must follow a {Z} code
{M}	prints the month
{A}	prints the day
{Y}	prints the year.

A sample print would look like **COSMIC A102496**.

AUTOCODES SORTED BY FUNCTION

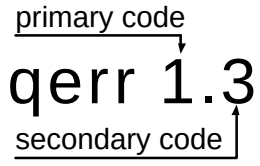
The table below contains all Series 2 autocodes sorted by function.

Function	Print Result	Autocode
Count - Pallet	up to 9 digits	{PC}
Count - Product	up to 9 digits	{PRD}
Count - Products by Shift		{SC}
Count - Sequential	up to 6 digits	{N}
Count - User Defined		{USR <usrnum>}
Date	mm/dd/yy	{D <offset>}
Day, month	1 character	{B <offset>}
Day, month -	2 digits	{A <offset>}
Day, week	1 digit or 1 letter	{X <option>}
Day, year	2 letters	{I <offset>}
Day, year	3 digits	{J <offset>}
Global String		{STR <globstrnum>}
Line Speed	3 digits	{SPD}
MACRO		{@ <macronum> <maxlen>}
Month,	1 letter	{L <offset>}
Month,	2 digits	{M <offset>}
Month,	3 letters	{O <offset>}
Scale Data		{W}
Shift Number	1 character	{SN <numalpha>}
Test Pattern	Vertical line	{TP}
Time, hours	1 letter	{G}
Time, hours	2 digits	{Vab}
Time, hours	2 digits	{H}
Time, hours	hh:mm	{T}
Time, minutes	2 digits	{C}
Time, minutes	2 digits, by 15 inc.	{Q}
Time, seconds	2 digits	{SEC}
Variable Data		{Z}{ }
Week	2 digits	{E}
Year	1 digit	{R <offset>}
Year	2 digits	{Y <offset>}

CHAPTER 5

Series 2 Error Codes and Definitions

Series 2 error codes have two parts to identify problems:



The primary code (1) identifies an "Invalid Autocode" and the secondary error code (3) narrows the problem down to a "Bad Autocode A."

The table of codes below will help to determine the meaning of the errors that appear onscreen.

Primary Code	Secondary Code	Definition
1	0	Invalid Autocode
	1	Unknown Autocode
	2	Bad Autocode @ (macro invocation)
	3	Bad Autocode A
	4	Bad Autocode B
	5	Bad Autocode C
	6	Bad Autocode D
	7	Bad Autocode E
	8	Bad Autocode G
	9	Bad Autocode H
	10	Bad Autocode I
	11	Bad Autocode J
	12	Bad Autocode L
	13	Bad Autocode M
	14	Bad Autocode N
	15	Bad Autocode O
	16	Bad Autocode PC
	17	Bad Autocode PI
	18	Bad Autocode PRD
	19	Bad Autocode Q
	20	Bad Autocode R
	21	Bad Autocode SC
	22	Bad Autocode SN
	23	Bad Autocode STR
	24	Bad Autocode T
	25	Bad Autocode USR
	26	Bad Autocode V
	27	Bad Autocode W
	28	Bad Autocode X
	29	Bad Autocode Y
	30	Bad Autocode Syntax
	31	Autocode output string too long

Primary Code	Secondary Code	Definition
2	0	Invalid font
3	0	Invalid macro
	1	Invalid character
	2	Invalid argument
	3	Divide by zero
	4	String overflow
	5	Numeric overflow
	6	Conversion error
	7	Constant too long
	8	Memory allocation error
	9	Free of invalid pointer
	10	Argument allocation error
	11	Function nesting too deep
	12	String allocation error
	13	Symbol table overflow
	14	Value stack overflow
	15	Macro allocation error
	16	Invalid subscript
	17	Label redefined
	18	Label undefined
	19	Array redefined
	20	Array undefined
	21	Invalid array dimension
4	0	ARCnet communication error
	1	Receive buffer Full
	2	Transmit buffer Full
	3	Too many transmit retries
	4	Excessive NAKs received
	5	ARCnet memory test failed
	6	Duplicate ARCnet id detected
5	0	Serial communication Error
	10	Parity error, serial port #1
	11	Rx buffer overflow error serial #1
	12	Overrun error, serial #1
	13	Framing error, serial #1
	20	Parity error, serial port 2
	21	Rx buffer overflow error serial #2
	22	Overrun error serial #2
	23	Framing error serial #2
	30	Parity error, serial port 3
	31	Rx buffer overflow error serial #3
	32	Overrun error, serial #3
	33	Framing error, serial #3
	40	Parity error, serial port #4
	41	Rx buffer overflow error serial
	42	Overrun error, serial #4
	43	Framing error, serial #4

Primary Code	Secondary Code	Definition
6	0	Backup battery DEAD / Critical
7	0	Macro run-time error
8	0	Insufficient time to parse label
9	0	Label line too long
10	0	Insufficient memory to store font
11	0	Insufficient memory to store macro
12	0	Insufficient memory to store .bmp
13	0	Label for printing is not resident
14	0	Font in label not resident
15	0	Macro not resident
16	0	Bitmap not resident
17	0	Host command syntax error
	1	Invalid GENC command
	2	Invalid SENC command
	3	Invalid GLNS command
	4	Invalid SLNS command
	5	Invalid GPLP command
	6	Invalid SPLP command
	7	Invalid GSP1 command
	8	Invalid GSP2 command
	9	Invalid GSP3 command
	10	Invalid GSP4 command
	11	Invalid SSP1 command
	12	Invalid SSP2 command
	13	Invalid SSP3 command
	14	Invalid SSP4 command
	15	Invalid GPHD command
	16	Invalid SPHD command
	17	Invalid GTIM command
	18	Invalid STIM command
	19	Invalid GDAT command
	20	Invalid SDAT command
	21	Invalid GSFT command
	22	Invalid SSFT command
	23	Invalid GPSD command
	24	Invalid GSFC command
	25	Invalid SSFC command
	26	Invalid GSEQ command
	27	Invalid SSEQ command
	28	Invalid GPRD command
	29	Invalid SPRD command
	30	Invalid GPAL command
	31	Invalid SPAL command
	32	Invalid GUC1 command
	33	Invalid GUC2 command
	34	Invalid SUC1 command
	35	Invalid SUC2 command

Primary Code	Secondary Code	Definition
17 cont.	36	Invalid GGST command
	37	Invalid SGST command
	38	Invalid GHST command
	39	Invalid SHST command
	40	Invalid GLGL command
	41	Invalid SLGL command
	42	Invalid GHMI command
	43	Invalid SHMI command
	44	Invalid GSCL command
	45	Invalid SSCL command
	46	Invalid GBCD command
	47	Invalid SBCD command
	48	Invalid GDTP command
	49	Invalid SDTP command
	50	Invalid QSTS command
	51	Invalid QAS1 command
	52	Invalid QSS1 command
	53	Invalid QSS2 command
	54	Invalid QSS3 command
	55	Invalid QSS4 command
	56	Invalid QAD1 command
	57	Invalid QSD1 command
	58	Invalid QSD2 command
	59	Invalid QSD3 command
	60	Invalid QSD4 command
	61	Invalid QPPS command
	62	Invalid QPPD command
	63	Invalid QMEN command
	64	Invalid RDEF command
	65	Invalid TLNS command
	66	Invalid FDWN command
	67	Invalid FDEL command
	68	Invalid LOPN command
	69	Invalid LDEL command
	70	Invalid LFLD command
	71	Invalid LCLS command
	72	Invalid LDIR command
	73	Invalid FDIR command
	74	Invalid BDWN command
	75	Invalid BDEL command
	76	Invalid MDWN command
	77	Invalid MDEL command
	78	Invalid PRT1 command
	79	Invalid PRTC command
	80	Invalid PWEB command
	81	Invalid XPRT command
	82	Invalid DPHD command

Primary Code	Secondary Code	Definition
17 cont.	83	Invalid FRMD command
	84	Invalid CLRC command
	85	Invalid QFRM command
	86	Invalid PURG command
	87	Invalid QLOG command
	88	Invalid QBAT command
	89	Invalid SWEB command
	90	Invalid GWEB command
	91	Invalid SMBX command
	92	Invalid GMBX command
	93	Invalid SDRT command
	94	Invalid GDRT command
	95	Invalid FPAD command
	96	Invalid BFLD command
	97	Invalid QPRT command
	98	Invalid QLBL command
	99	Invalid SRPX command
	100	Invalid GRPX command
	101	Invalid HDIR command
	102	Invalid QMAX command
	103	Invalid QST1 command
	104	Invalid QST2 command
	105	Invalid GBOX command
	106	Invalid SBOX command
	107	Invalid XPHD command
	108	Invalid IPHD command
	109	Invalid QCOM command
	110	Invalid QERR command
	111	Invalid QCMD command
	112	Invalid QADR command
	114	Invalid QRST command
	115	Invalid QSFT command
	116	Invalid TEST command
	117	Invalid QRUN command
	118	Invalid SATO command
	119	Invalid GATOcommand
	120	Invalid QBAR command
	121	Invalid BOOT command
	122	Invalid ABRT command
	123	Invalid FCLR command
	124	Invalid MTST command
	125	Invalid SGRP command
	126	Invalid GGRP command
	127	Invalid CGRP command
	128	Invalid QGRP command
	129	Invalid SDEPcommand

Primary Code	Secondary Code	Definition
17 cont.	130	Invalid GDEP command
	131	Invalid SLDR command
	132	Invalid GLDR command
	133	Invalid SYNC command
	135	Invalid QNET command
	136	Invalid PING command
	137	Invalid PONG command
	138	Invalid ANET command
	139	Invalid QPHD command
	140	Invalid QPLN command
	141	Invalid NEWScommand
	142	Invalid LGET command
	143	Invalid MTBF command
	144	Invalid GVDT command
	145	Invalid SVDT command
	146	Invalid SFMT command
	147	Invalid GFMT command
	148	Invalid SMRG command
	149	Invalid GMRG command
	150	Invalid QNXT command
	151	Invalid SPCI command
	152	Invalid GPCI command
	153	Invalid GLBL command
	154	Invalid QLAB command
	155	Invalid GLAB command
	157	Invalid LCPY command
	158	Invalid LREN command
	159	Invalid GARM command
	160	Invalid QFON command
	161	Invalid GFON command
	162	Invalid QHED command
	163	Invalid GHED command
	164	Invalid QLIN command
	165	Invalid GLIN command
	166	Invalid QLEX command
	167	Invalid QSET command
	168	Invalid QCNT command
	169	Invalid SREP command
	170	Invalid GREP command
	171	Invalid SBARcommand
	172	Invalid GBAR command
	173	Invalid SPRT command
	174	Invalid GPRT command
	175	Invalid QSTI command
	176	Invalid QST2 command
	177	Invalid QPSM command

Primary Code	Secondary Code	Definition
17 cont.	178	Invalid DDWN command
18	0	Macro language Syntax error
19	0	Attribute code syntax error
	1	Invalid font
	2	Nesting of ^U not allowed
	3	^U Translation length too large
	4	Invalid attribute code
20	0	Invalid label name
21	0	Invalid font name
22	0	Character not present in font
23	0	Real-time clock error
24	0	No label slots left
25	0	Out of space for labels
26	0	Label already exists
27	0	Memory allocation error
28	0	Can't delete a label while printing
29	0	No such logical line exists
30	0	No such printhead exists
31	0	Out of available logical line slots
32	0	Invalid dot range specified
33	0	Invalid SRAM detected
34	0	Unknown command encountered
35	0	Can't purge heads while printing
36	0	Printheads not defined-can't purge
37	0	Low battery detected
38	0	Invalid bar code syntax detected
39	0	Problem with flash memory
40	0	Problem with RPX chip
41	0	Moving too fast to print
42	0	ARCnet privilege violation
42	1	COM 1 privilege violation
42	2	COM 2 privilege violation
42	3	COM 3 privilege violation
42	4	COM 4 privilege violation
42	5	LC 100 privilege violation
42	6	PEER privilege violation
43	0	Duplicate group leader
44	0	Group without a leader
45	1	Sync command not issued before photocell
45	3	fifo underflow
45	4	fifo overflow
46	1	Invalid port number
46	2	CRC error

Primary Code	Secondary Code	Definition
46	3	Invalid algorithm
	4	Invalid device type
	5	Error programing device
47	0	Cannot do this while printing
48	1	Image pre-format not enabled
48	2	Photocell inhibit not enabled
48	3	Print window / zone too big for box
48	4	Photocell too far from first printhead
48	5	Virtual print window too large
48	6	Too many boxes in the print window
48	7	Photocell trip too soon for this box size.
48	8	Merge page not ready / preformat not done
48	9	Merge not allowed
48	10	Page change during page computations.
49		For future use
50		Zero printhead offset not allowed
51	0	Missing CIS, card information structure
51	1	Unknown tuple code
51	2	Unknown size of PCMCIA card
51	3	Unknown vendor code
51	4	Card is not R/W SRAM type
51	5	Card is not correct 150 nsec speed
51	6	Card is not 2.1 compliant
51	7	Card was removed after detected
51	8	Corrupt / missing DOS sector
51	9	Corrupt / missing BPB
51	10	Corrupt DOS FAT table
51	11	Corrupt DOS root directory
51	12	Card is write protected
51	13	PCMCIA write problem, check resistor mod

APPENDIX A

NEW HOST COMMANDS

When command parameters have no default values, the "Default" field name has been eliminated.

"Related Cmd:" identifies commands related to the command under discussion.

SARM Set ARM Mode

Sets the power-up print mode.

Group: Action

Format: <soc> SARM, mode

Return Data: N/A

Parameter 1: Mode: Power-up print command. 0=XPRT
1=Previous print command

Example: <esc>SARM,1<cr> sets the power up print mode to use the previous print mode.
