Alpha® Sign Communications Protocol

Revision E

Contents

1.0 Contents 3

2.0 Introduction 6

3.0 Document information 6

4.0 Protocol overview 7

5.0 Transmission packet formats 9

- 5.1 Standard transmission packet ("1-byte" or "^A") format 10
- 5.2 ASCII Printable formats 15

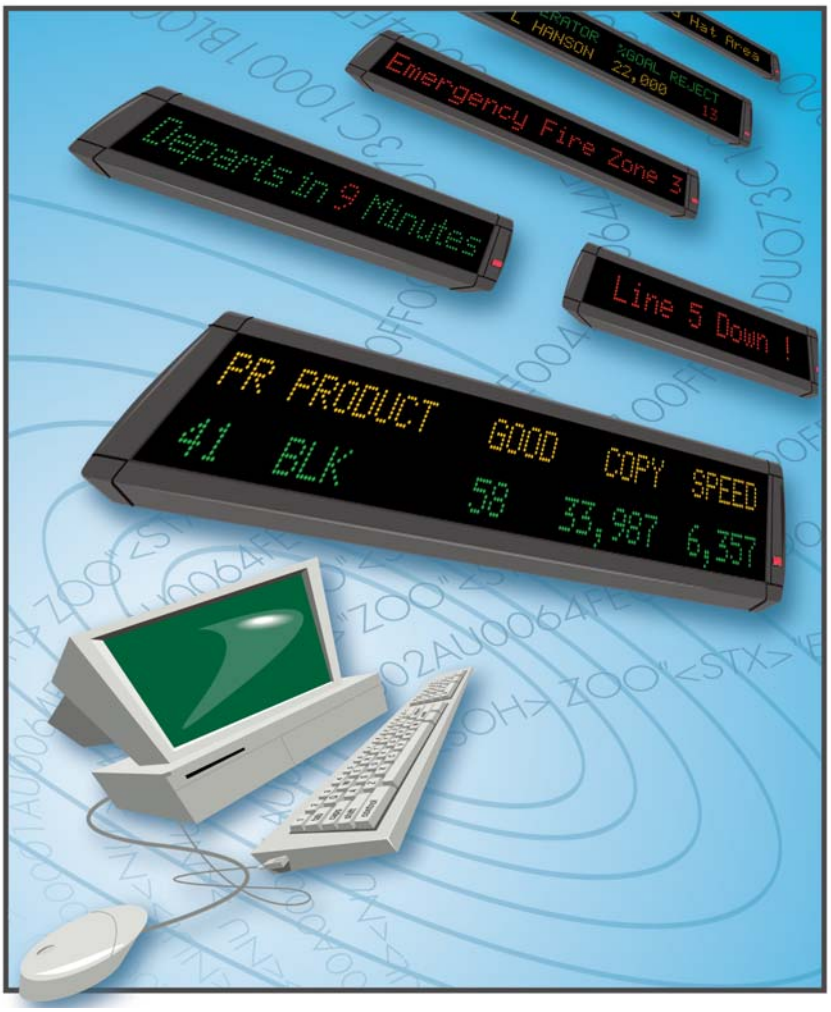
6.0 Command Codes 17

- 6.1 TEXT file commands 17
- 6.2 SPECIAL FUNCTION commands 21
- 6.3 STRING file commands 36
- 6.4 SMALL DOTS PICTURE file commands 38
- 6.5 LARGE DOTS PICTURE file commands 41
- 6.6 RGB DOTS PICTURE file commands 43
- 6.7 ALPHAVISION BULLETIN MESSAGE file commands 47

7.0 Appendixes 49

- 7.1 Appendix A: Valid File Labels 49
- 7.2 Appendix B: Valid Start and Stop times 50
- 7.3 Appendix C: Counter information 51
- 7.4 Appendix D: STRING file notes 52
- 7.5 Appendix E: Sample programs 55
- 7.6 Appendix F: Protocol examples 57
- 7.7 Appendix G: Alpha protocol ASCII table 80
- 7.8 Appendix H: ISO ASCII table 87
- 7.9 Appendix I: Modes, fonts, colors, and display options available on signs 88
- 7.10 Appendix J: Position rules for signs 93
- 7.11 Appendix K: Alpha 2.0 protocol additions 98
- 7.12 Appendix L: Alpha 3.0 protocol additions 122
- 7.13 Appendix M: Font character sets 125

This document explains how to use the Alpha sign communications protocol to send messages and graphics to Alpha signs.



For protocol examples, go to Adaptive's FTP site:
ftp://ftp.ams-i.com/alpha_protocol_examples/

ADAPTIVE®

Adaptive Micro Systems • 7840 North 86th Street • Milwaukee, WI 53224 USA • 414-357-2020 • 414-357-2029 (fax) • http://www.adaptivedisplays.com

August 1, 2003

9708-8061E

NOTE: Due to continuing product innovation, specifications in this document are subject to change without notice.

Trademarked names appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, the publisher states that it is using the names for editorial purposes and to the benefit of the trademark owner with no intention of improperly using the trademark.

Visit our Internet World Wide Web site: http://www.adaptivedisplays.com

1.0 Contents

2.0 Introduction . . . 6

3.0 Document information . . . 6

3.1 Revision history . . . 6
3.2 Document conventions . . . 6

4.0 Protocol overview . . . 7

4.1 Displaying text . . . 7
4.2 Displaying graphics . . . 7
4.3 Special functions . . . 7
4.4 Protocol version comparison . . . 8

5.0 Transmission packet formats . . . 9

5.1 Standard transmission packet (“1-byte” or “^A”) format . . . 10
5.1.1 Checksum format . . . 12
5.1.2 Nesting with Checksums format . . . 13
5.1.3 Nesting without Checksums format . . . 14
5.2 ASCII Printable formats . . . 15
5.2.1 ASCII Printable “2-byte” code . . . 15
5.2.2 ASCII Printable “3-byte” code . . . 16

6.0 Command Codes . . . 17

6.1 TEXT file commands . . . 17
6.1.1 Write TEXT file Command Code — “A” (41H) . . . 18
6.1.2 Read TEXT file Command Code — “B” (42H) . . . 19
6.1.3 Priority TEXT files . . . 20
6.2 SPECIAL FUNCTION commands . . . 21
6.2.1 Write SPECIAL FUNCTION Command Code — “E” (45H) . . . 21
6.2.2 Read SPECIAL FUNCTION Command Code — “F” (46H) . . . 28
6.3 STRING file commands . . . 36
6.3.1 Write STRING file Command Code — “G” (47H) . . . 36
6.3.2 Read STRING file Command Code — “H” (48H) . . . 37
6.4 SMALL DOTS PICTURE file commands . . . 38
6.4.1 Write SMALL DOTS PICTURE file Command Code — “I” (49H) . . . 38
6.4.2 Read SMALL DOTS PICTURE file Command Code — “J” (4AH) . . . 40
6.5 LARGE DOTS PICTURE file commands . . . 41
6.5.1 Write LARGE DOTS PICTURE file Command Code — “M” (4DH) . . . 41
6.5.2 Read LARGE DOTS PICTURE file Command Code — “N” (4EH) . . . 42
6.6 RGB DOTS PICTURE file commands . . . 43
6.6.1 Write RGB DOTS PICTURE file Command Code — “K” (4BH) . . . 43
6.6.2 Read RGB DOTS PICTURE file Command Code — “L” (4CH) . . . 45
6.6.3 RGB color chart . . . 46
6.7 ALPHAVISION BULLETIN MESSAGE file commands . . . 47
6.7.1 Write ALPHAVISION BULLETIN MESSAGE file Command Code — “O” (4FH) . . . 47
6.7.2 Stop ALPHAVISION BULLETIN MESSAGE file Command Code — “OT” (4F + 54H) . . . 48

7.0 Appendixes 49
7.1 Appendix A: Valid File Labels 49
7.2 Appendix B: Valid Start and Stop times 50
7.3 Appendix C: Counter information 51
7.3.1 Displaying Counter values 51
7.3.2 Setting up Counters 51
7.4 Appendix D: STRING file notes 52
7.4.1 Advantages of using STRING files 52
7.4.2 Using STRING files example 52
7.5 Appendix E: Sample programs 55
7.5.1 Sample C program 55
7.5.2 Sample BASIC program 56
7.6 Appendix F: Protocol examples 57
7.6.1 Standard transmission packet examples 57
7.6.2 Transmission packet with Checksum example 59
7.6.3 Nesting with checksums transmission packet example 60
7.6.4 Nesting without Checksum transmission packet example 61
7.6.5 Multiple Type Codes / Sign Addresses example 62
7.6.6 TEXT file examples 63
7.6.7 Priority TEXT file examples 67
7.6.8 SPECIAL FUNCTION examples 68
7.6.9 STRING file examples 73
7.6.10 DOTS PICTURE file examples 75
7.6.11 Displaying text at XY position examples 76
7.7 Appendix G: Alpha protocol ASCII table 80
7.7.1 Standard character set (00 –7FH) 80
7.7.2 Extended character set (80 - C1H) 83
7.8 Appendix H: ISO ASCII table 87
7.9 Appendix I: Modes, fonts, colors, and display options available on signs 88
7.9.1 Standard Modes 88
7.9.2 Special Modes 88
7.9.3 Special Graphics 89
7.9.4 Modes available on signs 90
7.9.5 Fonts and colors available on signs 91
7.9.6 Display options available on signs 92
7.10 Appendix J: Position rules for signs 93
7.10.1 Sign classes 93
7.10.2 Position classes 93
7.10.3 Position rule examples 93
7.11 Appendix K: Alpha 2.0 protocol additions 98
7.11.1 Speed control 98
7.11.2 Sound control (AlphaPremiere 9000 only) 99
7.11.3 Set Run File Time 100
7.11.4 Custom character sets 104
7.11.5 Automode table 107
7.11.6 Set Timeout Message 108
7.11.7 Dimming Control Register 108
7.11.8 Enable/Disable ACK/NAK response 111
7.11.9 Temperature Logging 112
7.11.10 Read External Temperature command 115
7.11.11 Read Internal Temperature command 116
7.11.12 Set Unit commands 118

7.11.13 Read Dim Times command . . . 120
7.12 Appendix L: Alpha 3.0 protocol additions . . . 122
7.13 Appendix M: Font character sets . . . 125
7.13.1 5-High Regular (SS5) . . . 125
7.13.2 7-High Regular (SS7) . . . 127
7.13.3 7-High Fancy (SF7) . . . 134
7.13.4 7-High True Descender Regular . . . 141
7.13.5 7-High True Descender Fancy . . . 141
7.13.6 7-High Fat Character . . . 142
7.13.7 8-High Regular (SS8) . . . 142
7.13.8 8-High Fancy (SF8) . . . 150
7.13.9 10-High Fancy (SF10) . . . 159
7.13.10 10-High True Descender Fancy . . . 165
7.13.11 15-High Regular (SS15) . . . 166
7.13.12 15-High Fancy (SF15) . . . 174
7.13.13 15-High True Descender Regular . . . 182
7.13.14 15-High True Descender Fancy . . . 183
7.13.15 15-High Fat Character . . . 183
7.13.16 16-High Regular (SS16) . . . 184
7.13.17 16-High Fancy (SF16) . . . 192
7.13.18 16-High Fat Character . . . 200

2.0 Introduction

This document is designed to allow a user to understand how to communicate with the Alpha line of electronic signs manufactured by Adaptive Micro Systems. The signs must have the Alpha firmware (EPROM) installed.

There are four versions of protocol with which you can communicate with a Alpha sign (see Table 3, "Protocol version comparison," on page 8):

- EZ KEY II
- Alpha 1.0 (EZ95)
- Alpha 2.0
- Alpha 3.0

These protocols were created to display text messages on electronic signs, but the protocols can also display graphics, temperature, counters, and more.

3.0 Document information

3.1 Revision history

Table 1: Revision history

Revision date	Document part number	Notes
May 17, 1995	9708-8061	First release.
August 4, 1995	9708-8061A	• PrintPak information added • Printable character terminations added • Identifier page with revision list added
May 1, 1998	9708-8061B	• Document reformatted
May 28, 1998	9708-8061B	• Corrections to 5/1/98 release.
July 1, 1999	9708-8061C	• Various corrections to 5/28/98 release. • "POCSAG" changed to "ASCII Printable" • PrintPak protocol information removed • Y2K date correction information added
August 15, 2002	9708-8061D	• added Alpha 2.0 protocol information • added Betabrite model 1036 character set and symbols • corrected the Extended Character Set in the Alpha protocol ASCII table • corrected the *Set Run Time Table* Special Function. • added new Special Function for AlphaVision character matrix signs (*Display Text at XY Location on Sign*) • added Position rules for signs in Appendix. • various minor corrections and additions • added the AlphaEclipse protocol addendum • added font character sets • added Set Automode Table information
August 1, 2003	9708-8061E	• added Alpha 3.0 protocol information (page 122) • expanded Alpha 2.0 protocol information (page 98) • added protocol version comparison table (page 8) • removed "Daylight Savings" command "=" (3DH) because it was never implemented • standardized terminology (for example, "frame" changed to "packet")

3.2 Document conventions

Table 2: Document conventions

Convention	Description
<SOH> or ^A	ASCII control character abbreviation (see page 87)
"A"	ASCII character (in this case the letter A)

Table 2: Document conventions

Convention	Description
11D	Decimal number (in this case, 11). Numbers that are not followed by any letter are also decimal.
0BH	Hexadecimal number (0B hex = 11 decimal)
01001100B	Binary number

4.0 Protocol overview

The Alpha line of products — which also includes AlphaVision, AlphaPremiere, and AlphaEclipse signs — supports several types of files and a number of special functions which are used for specific applications:

4.1 Displaying text

4.1.1 TEXT files

The ASCII message data and display mode information, along with various other control codes, are stored in TEXT files. DOTS PICTURE files and STRING files may be inserted into a TEXT file.

4.1.2 STRING files

The STRING files are used to store ASCII characters only. STRING files are used in applications where a string of frequently changing data must be transmitted to, and displayed by, a sign. Applications include the storage of a number which changes often, such as a temperature, a quantity, or a timer.

4.2 Displaying graphics

4.2.1 SMALL DOTS PICTURE files

SMALL DOTS PICTURE files contain data patterns that correspond to a display picture. These patterns can be used to create virtually any logo pattern on the display of the sign. These SMALL DOTS PICTURE files are accessed via TEXT files. SMALL DOTS PICTURE files have a maximum size of 31 x 255 pixels.

4.2.2 LARGE DOTS PICTURE (also called “ALPHAVISION DOTS PICTURE” or “FAR DOTS PICTURE”) files

LARGE DOTS PICTURE files are similar to the SMALL DOTS PICTURE file described above. However, a LARGE DOTS PICTURE file can be much larger. The LARGE DOTS PICTURE file supports data compression during serial transmission and has a maximum size of 65535 x 65535 pixels.

4.2.3 RGB DOTS PICTURE files

Based on LARGE DOTS PICTURE files, a RGB DOTS PICTURE can display over 16 million RGB (Red-Green-Blue) colors.

4.3 Special functions

The Alpha network supports a range of SPECIAL FUNCTION commands which give you access to internal registers, diagnostics, and other items.

4.4 Protocol version comparison

Table 3: Protocol version comparison

		EZKEY II	Alpha 1.0 (EZ95)		Alpha 2.0		Alpha 3.0	
First released:		1991	1995		2001		June 2003	
Data format	Baud rate:	1200, 2400, 4800	1200, 2400, 4800, 9600		1200, 2400, 4800, 9600, 19200, 38400			
	Start bits:	1						
	Data bits:	7	7	8	7	8	7	8
	Parity:	Even	Even	None	Even	None	Even	None
	Stop bits:	2	2	1	2	1	2	1
	Flow control:	None						
	Time-out period:	1 second[1]						
Sign compatibility[7]	200 Series[2]:	Yes	Yes	Yes	No	No	No	No
	220C:	Yes	Yes	Yes	No	No	No	No
	300 Series[3]:	Yes	Yes	Yes	No	No	No	No
	420C:	Yes	No	No	No	No	No	No
	430i:	Yes	No	No	No	No	No	No
	440i:	Yes	No	No	No	No	No	No
	460i:	Yes	No	No	No	No	No	No
	790i:	Yes	No	No	No	No	No	No
	4000 Series[4]:	Yes	Yes	Yes	No	No	No	No
	7000 Series[5]:	Yes	Yes	Yes	No	No	No	No
	AlphaEclipse 1500 Time & Temp[6]:	Yes	Yes	Yes	Yes	Yes	No	No
	AlphaEclipse 2500:	Yes	Yes	Yes	Yes	Yes	No	No
	AlphaEclipse 2600:	Yes	Yes	Yes	Yes	Yes	No	No
	AlphaEclipse 3500:	Yes	Yes	Yes	Yes[8]	Yes[8]	No	No
	AlphaEclipse 3600[9]:	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	AlphaPremiere:	Yes	Yes	Yes	Yes	Yes	No	No
	AlphaVision (full matrix):	Yes	Yes	Yes	No	No	No	No
	AlphaVision (character matrix):	Yes	Yes	Yes	No	No	No	No
	Betabrite:	Yes	Yes	Yes	No	No	No	No
	Big Dot:	Yes	Yes	Yes	No	No	No	No
	Director:	Yes	Yes	Yes	No	No	No	No
	PPD (Personal Priority Display):	Yes	Yes	Yes	No	No	No	No
	Serial LED clock[6]:	Yes	Yes	Yes	Yes	Yes	No	No
	Solar:	Yes	Yes	Yes	No	No	No	No

NOTES:

[1] This 1-second delay between each byte applies to the Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10). However, for ASCII Printable formats (see "ASCII Printable formats" on page 15) the delay can be as long as 30 seconds between each byte.

[2] This includes the 215R and 215C model signs ("C" = tricolor LEDs, "R" = red LEDs).

[3] This includes the 320C and 330C model signs ("C" = tricolor LEDs, "R" = red LEDs).

[4] This includes the 4080C, 4120C, 4120R, 4160C, 4160R, 4200C, 4200R, 4240C, and 4240R model signs ("C" = tricolor LEDs, "R" = red LEDs).

[5] This includes the 7080C, 7120C, 7160C, and 7200C model signs ("C" = tricolor LEDs, "R" = red LEDs).

[6] This sign can only display time updates from messaging software. This sign cannot display text messages or graphics.

[7] "Yes" means the protocol version specified above works with the specified sign.

[8] In order to use the Alpha 2.0 protocol Set Unit commands (see **Table 70** on page 98), an AlphaEclipse 3500 Series sign must either be (1) a Series A sign with revision "G" or greater main firmware, or a (2) Series B or greater sign. The Alpha 3.0 Set Unit commands "U7", "U8", and "U9" (see **Table 102** on page 122) are only usable with AlphaEclipse 3600 signs.

[9] This sign has RGB (red, green, and blue) LEDs that are capable of displaying over 16 million colors.

5.0 Transmission packet formats

Each of the protocols (EZ KEY II, Alpha 1.0, and so on) can be transmitted to a sign in either one of two, basic formats:

1. Standard (Figure 1) — also called the "1-byte" or "^A" format.

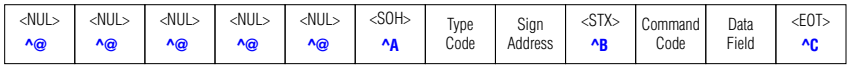


Figure 1: Standard transmission packet

The Standard format has several variations:

- Checksum
- Nesting with Checksums
- Nesting without Checksums

SPECIAL NOTE

When a sign receives an invalid Checksum, the data in the associated packet will not be processed.

To determine if a packet was received with a valid Checksum, you would have to read the Serial Error Status Register (page 30) immediately after a packet was written to the sign.

2. ASCII Printable — any one of the above Standard formats can be converted into an "ASCII Printable" format by simply making the non-printable control codes *printable* ASCII characters. There are two ways to do this:

- ASCII Printable "2-byte" format — non-printable characters (like <SOH>) are converted into *two*, printable ASCII characters (like "]!").
- ASCII Printable "3-byte" format — non-printable characters (like <SOH>) are converted into *three*, printable ASCII characters (like "_01")

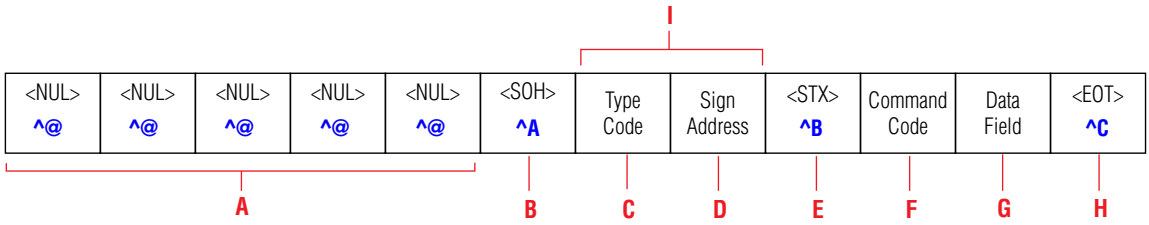
5.1 Standard transmission packet ("1-byte" or "^A") format

SHOW ME

An example of the Standard transmission packet is on page 57.

This is called the "1-byte" or "^A" format because single-byte, non-printable control characters like <SOH> are used in the packet:

Table 4: Standard transmission packet ("1-byte" or "^A") format



Item	Name	Description
A	<NUL>	A minimum of five <NUL>s (00H) must be transmitted as packet synchronization characters. Five <SOH>s (01H) may be substituted for the five <NUL>s. The sign uses these five characters to establish the baud rate.
B	<SOH>	The <SOH> (01H) is the "Start Of Header" ASCII character.
C	Type Code	A single ASCII character (to send multiple Type Codes, see item I):

Table 5: Type Codes

Sign Type Code		Sign
"!"	21H	All signs with Visual Verification. This code causes a sign to display the *Transmission OK* message when a transmission packet is received without an error. Otherwise, *Transmission Error* will appear.
"""	22H	Serial clock
"#"	23H	AlphaVision sign
"$"	24H	Full matrix AlphaVision sign
"%"	25H	Character matrix AlphaVision sign
"&"	26H	Line matrix AlphaVision
"0"	30H	Response code used only when a sign responds to a request.
"1"	31H	One-line signs
"2"	32H	Two-line signs
"?"	3FH	All signs
"C"	43H	430i sign
"D"	44H	440i sign
"E"	45H	460i sign
"F"	46H	AlphaEclipse 3600 display driver board
"G"	47H	AlphaEclipse 3600 Turbo Adapter board
"L"	4CH	Light sensor probe
"U"	55H	790i sign
"V"	56H	AlphaEclipse 3600 series
"W"	57H	AlphaEclipse Time/Temp
"X"	58H	AlphaPremiere 4000 and 9000 series
"Z"	5AH	All signs
"^"	5EH	Betabrite sign
"a"	61H	4120C sign
"b"	62H	4160C sign

Sign Type Code		Sign
"c"	63H	4200C sign
"d"	64H	4240C sign
"e"	65H	215R sign
"f"	66H	215C sign
"g"	67H	4120R sign
"h"	68H	4160R sign
"i"	69H	4200R sign
"j"	6AH	4240R sign
"k"	6BH	300 series sign
"l"	6CH	7000 series sign
"m"	6DH	96x16 matrix Solar sign
"n"	6EH	128x16 matrix Solar sign
"o"	6FH	160x16 matrix Solar sign
"p"	70H	192x16 matrix Solar sign
"q"	71H	PPD sign
"r"	72H	Director sign
"s"	73H	1005 digit controller
"t"	74H	4080C sign
"u"	75H	210C and 220C signs
"v"	76H	AlphaEclipse 3500 signs
"w"	77H	AlphaEclipse 1500 Time & Temp sign
"x"	78H	AlphaPremiere 9000 sign
"z"	7AH	All signs with memory configured for 26 files ("A" - "Z").

Table 4: Standard transmission packet ("1-byte" or "^A") format

D	Sign Address	The identifier or "address" of the sign represented by two ASCII digits as a number between "00" and "FF" (0 to 255). Address "00" is reserved as a broadcast address. The wildcard character "?" (3FH) can be used to send messages to a range of addresses. For example, a Sign Address of "0?" will access signs with address between 01H and 0FH (1 and 15). To send multiple Sign Addresses, see item I.
E	<STX>	"Start of TeXt" (02H) character. <STX> always precedes a Command Code. NOTE: When nesting packets, there must be at least a 100-millisecond delay after the <STX>.
F	Command Code	One ASCII character that defines the transmission and data types: **Table 6: Command Codes** **Command Code** / **Reference** "A" 41H — Write TEXT file (see page 18) "B" 42H — Read TEXT file (see page 19) "E" 45H — Write SPECIAL FUNCTION commands (see page 21) "F" 46H — Read SPECIAL FUNCTION commands (see page 28) "G" 47H — Write STRING file (see page 36) "H" 48H — Read STRING file (see page 37) "I" 49H — Write SMALL DOTS PICTURE file (see page 38) "J" 4AH — Read SMALL DOTS PICTURE file (see page 40) "K" 4BH — Write RGB DOTS PICTURE file (see page 43) (Alpha 3.0 protocol only) "L" 4CH — Read RGB DOTS PICTURE file (see page 45) (Alpha 3.0 protocol only) "M" 4DH — Write LARGE DOTS PICTURE file (see page 41) "N" 4EH — Read LARGE DOTS PICTURE file (see page 42) "O" 4FH — Write ALPHAVISION BULLETIN message (see page 47) "T" 54H — Set Timeout Message (see page 108) (Alpha 2.0 and 3.0 protocols only) NOTE: When nesting commands, only one "Read" Command Code may be used, and it must be the last Command Code before the <EOT>. NOTE: The "Write SPECIAL FUNCTION commands" to Speaker Tone Generation must be the last command in a nested string.
G	Data Field	Made up of ASCII characters. The Data Field format is dependent on the preceding Command Code.
H	<EOT>	"End Of Transmission" (04H) character
I	Multiple Type Codes and Sign Address	Instead of sending a single Type Code and Sign Address (like "g02"), multiple Type Codes and Sign Addresses can be transmitted using the following format: Aaa,Bbb,Ccc, . . . where: A, B, and C = ASCII Type Codes aa,bb,cc = ASCII Sign Addresses separated by commas (2CH), for example, g02,U01,21F,220

5.1.1 Checksum format

SHOW ME

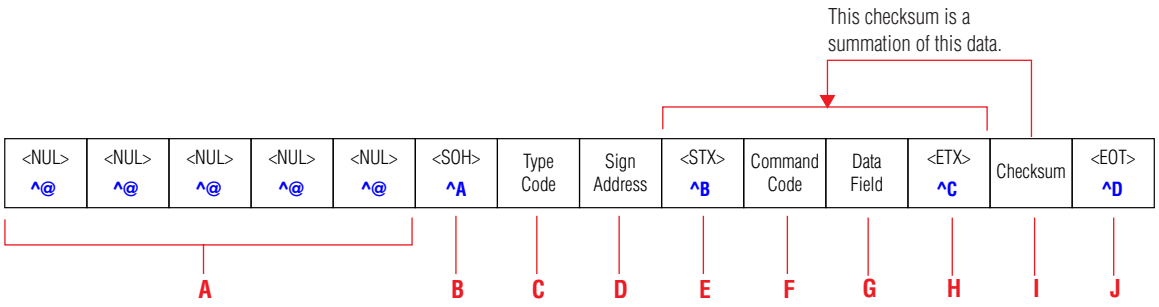
An example of the Transmission packet with Checksum is on page 59.

The standard transmission packet format has a few acceptable variations which have their own advantages, depending on the application.

If an <ETX> character is transmitted before the <EOT>, the sign will expect a Checksum.

When a sign receives an invalid Checksum, the associated data will not be processed.

Table 7: Standard transmission packet with Checksum format



Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.
B	<SOH>	
C	Type Code	
D	Sign Address	
E	<STX>	
F	Command Code	
G	Data Field	
H	<ETX>	"End of TeXt" (03H) character
I	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first. NOTE: When a sign receives an invalid Checksum, the associated data will not be processed. To see if a packet had a valid Checksum, use the Read SPECIAL FUNCTION to check the Serial Error Status Register (see page 28).
J	<EOT>	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.

5.1.2 Nesting with Checksums format

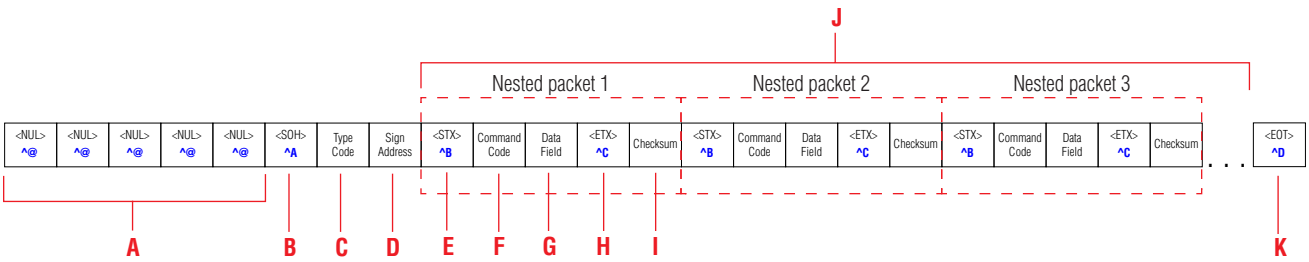
SHOW ME

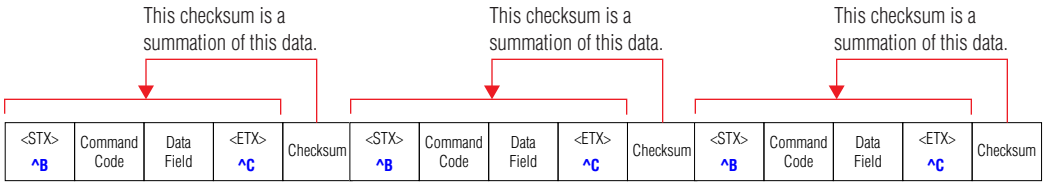
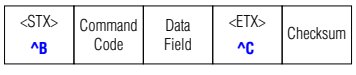
An example of the Nesting with Checksums is on page 60.

If more than one transmission packet is required consecutively, multiple Commands can be repeated or "nested" within a transmission packet.

A sign uses this format when a Memory Dump [see "Read SPECIAL FUNCTION Command Code — "F" (46H)" on page 28] is requested serially.

Table 8: Nesting with Checksums format



Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.
B	<SOH>	
C	Type Code	
D	Sign Address	
E	<STX>	
F	Command Code	
G	Data Field	
H	<ETX>	"End of TeXt" (03H) character
I	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first. For example, in the following three nested packets, this is how the checksums are calculated:  NOTE: When a sign receives an invalid Checksum, the associated data will not be processed. NOTE: When nesting packets, there must be at least a 100-millisecond delay after the <STX>. NOTE: When nesting commands, only one "Read" Command Code may be used, and it must be the last Command Code before the <EOT>. NOTE: The "Write SPECIAL FUNCTION commands" to Speaker Tone Generation must be the last command in a nested string.
J	Nested Commands with Checksums	Multiple Commands can be "nested" in a transmission packet. This is the format of the nested packet *with a Checksum*: 
K	<EOT>	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.

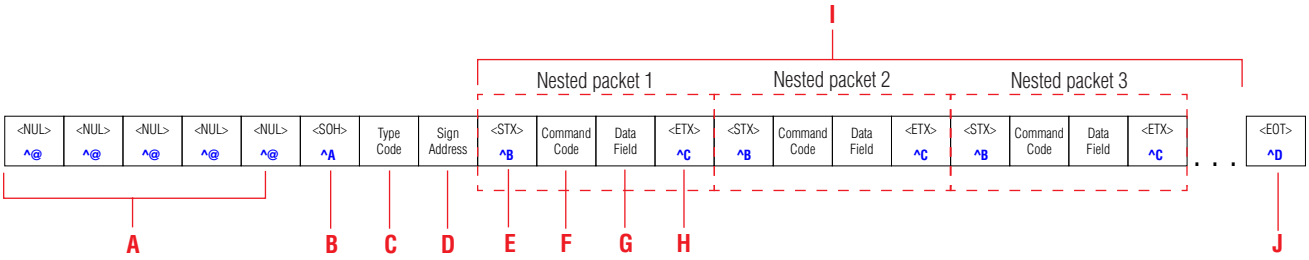
5.1.3 Nesting without Checksums format

SHOW ME

An example of the Nesting without Checksums is on page 61.

If an <STX> is transmitted immediately following an <ETX>, the sign will expect the next "nested" command.

Table 9: Nesting without Checksums transmission packet



Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.
B	<SOH>	
C	Type Code	
D	Sign Address	
E	<STX>	
F	Command Code	
G	Data Field	
H	<ETX>	"End of TeXt" (03H) character
I	Nested Commands without Checksums	Multiple Commands can be "nested" in a transmission packet. This is the format of the nested packet *without a Checksum*: <STX> ^B | Command Code | Data Field | <ETX> ^C NOTE: When nesting packets, there must be at least a 100-millisecond delay after the <STX>. NOTE: When nesting commands, only one "Read" Command Code may be used, and it must be the last Command Code before the <EOT>. NOTE: The "Write SPECIAL FUNCTION commands" to Speaker Tone Generation must be the last command in a nested string.
J	<EOT>	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.

5.2 ASCII Printable formats

SPECIAL NOTE

For ASCII Printable format baud rate, parity, etc., see Table 3, "Protocol version comparison," on page 8.

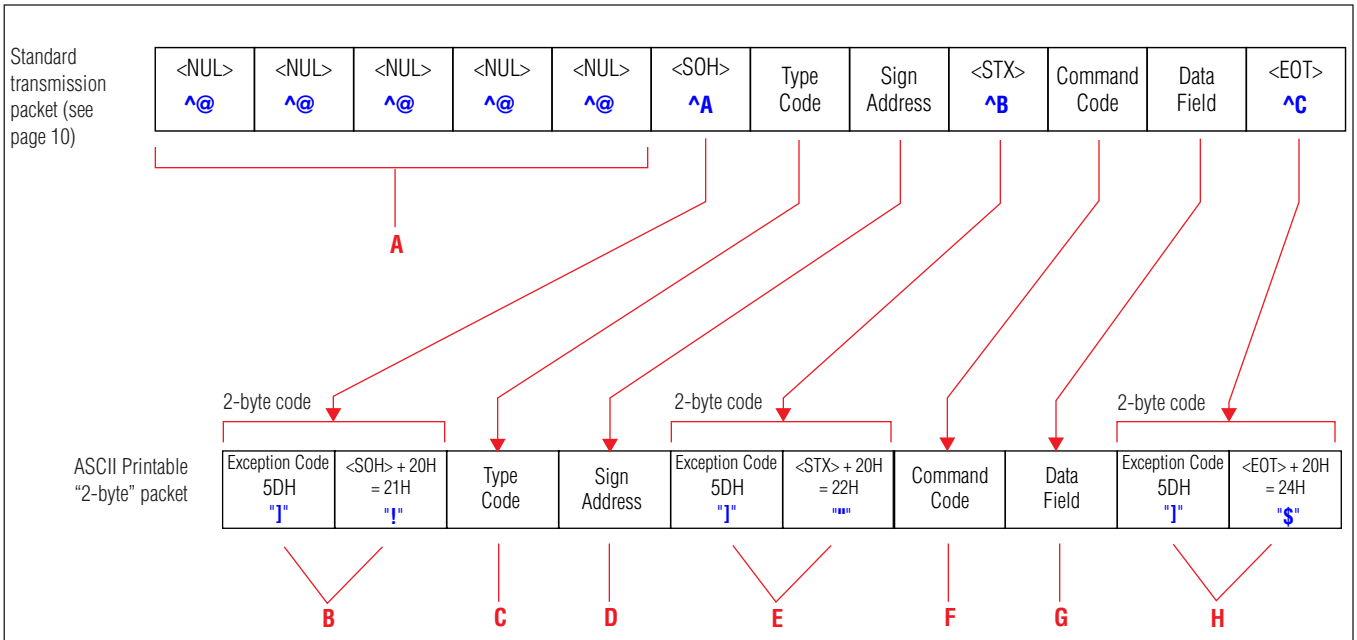
Many pagers and computer systems cannot receive or send ASCII control codes (characters lower than 20H). The ASCII Printable format is a variation of the transmission packet that allows the entire protocol to be transmitted *without* sending any ASCII control codes — thus allowing its use with pagers.

This can be implemented in two ways, as shown below. However, an Exception Code must precede all Control Codes that are used in a transmission.

5.2.1 ASCII Printable "2-byte" code

This format is often referred to as the "2-byte" protocol because of the use of the "]!" characters in the transmission packet.

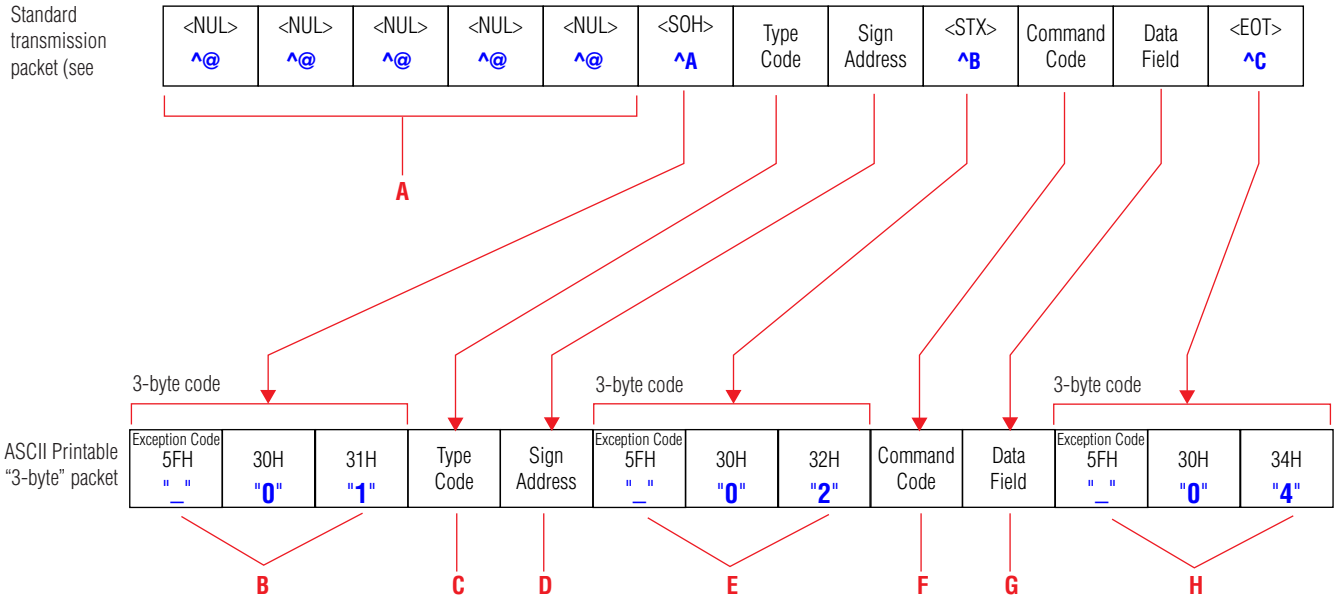
Table 10: Standard transmission packet compared with ASCII Printable "2-byte" code transmission packet



Item	Name	Description
A	<NUL>	The five <NUL>s (00H) used as packet synchronization are not used in the ASCII Printable transmission format. Since these <NUL>s were used to determine the baud rate, the ASCII Printable baud rate must be preset to 9600.
B	Exception Code	Exception Code = 5DH = "]"
	<SOH> + 20H	<SOH> + 20H = 21H = "!". The <SOH> (01H) ASCII control code is converted to a printable ASCII character by adding the 20H offset.
C	Type Code	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.
D	Sign Address	
E	Exception Code	Exception Code = 5DH = "]"
	<STX> + 20H	<STX> + 20H = 22H = """. The <STX> (02H) ASCII control code is converted to a printable ASCII character by adding the 20H offset.
F	Command Code	See Table 4, "Standard transmission packet ("1-byte" or "^A") format," on page 10.
G	Data Field	
H	Exception Code	Exception Code = 5DH = "]"
	<EOT> + 20H	<EOT> + 20H = 24H = "$". The <EOT> (04H) ASCII control code is converted to a printable ASCII character by adding the 20H offset.

5.2.2 ASCII Printable “3-byte” code

Table 11: Standard transmission packet compared with ASCII Printable “3-byte” code transmission packet



Item	Name		Description
A	<NUL>		The five <NUL>s (00H) used as packet synchronization are not used in the ASCII Printable transmission format. Since these <NUL>s were used to determine the baud rate, the ASCII Printable baud rate must be preset to 9600.
B	Exception Code		Exception Code = 5FH = “_”
	<SOH> =	“0” + “1”	The <SOH> (01H) is broken down into two ASCII bytes “0” and “1”.
C	Type Code		See Table 4, “Standard transmission packet (“1-byte” or “^A”) format,” on page 10.
D	Sign Address		
E	Exception Code		Exception Code = 5FH = “_”
	<STX> =	“0” + “2”	The <STX> (02H) is broken down into two ASCII bytes “0” and “2”.
F	Command Code		See Table 4, “Standard transmission packet (“1-byte” or “^A”) format,” on page 10.
G	Data Field		
H	Exception Code		Exception Code = 5FH = “_”
	<EOT> =	“0” + “4”	The <EOT> (04H) is broken down into two ASCII bytes “0” and “4”.

6.0 Command Codes

A Command Code (Table 6, "Command Codes," on page 11) is used to determine whether information is read from or written to signs.

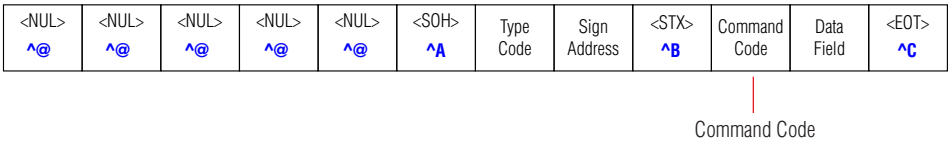


Figure 2: Command Code location in the Standard transmission packet

In addition to determining whether information is written or read, Command Codes determine the contents of the Data Field in the protocol transmission packet formats (see "Transmission packet formats" on page 9).

Command Codes fall into six, general categories:

- TEXT file commands
- SPECIAL FUNCTION commands (page 21)
- STRING file commands (page 36)
- SMALL DOTS PICTURE file commands (page 38)
- LARGE DOTS PICTURE file commands (page 41)
- RGB DOTS PICTURE file commands (page 43)
- ALPHAVISION BULLETIN MESSAGE file commands (page 47)

6.1 TEXT file commands

The ASCII message data and display mode information, along with various other control codes are stored in TEXT files. On initial power-up, the sign's memory is configured with one TEXT file (File Label = "A"). If multiple TEXT files are required, refer to the section in SPECIAL FUNCTION commands on Memory Configuration for further details.

When writing to a TEXT file, the display will blank. After the transmission is over, the unit will begin displaying the last received TEXT file.

When reading from a TEXT file, the display will pause when it is sending the transmission packet. Once the unit has completely transmitted the file, it will continue displaying the message from where it was interrupted.

As well as containing the actual message, "calls" to other types of files may be inserted into TEXT files. For example, if you wish to include a DOTS PICTURE as part of a TEXT file, you may simply include a call to a DOTS PICTURE file in the proper location in your TEXT file. Refer to the DOTS PICTURE files section or the STRING files section for further information.

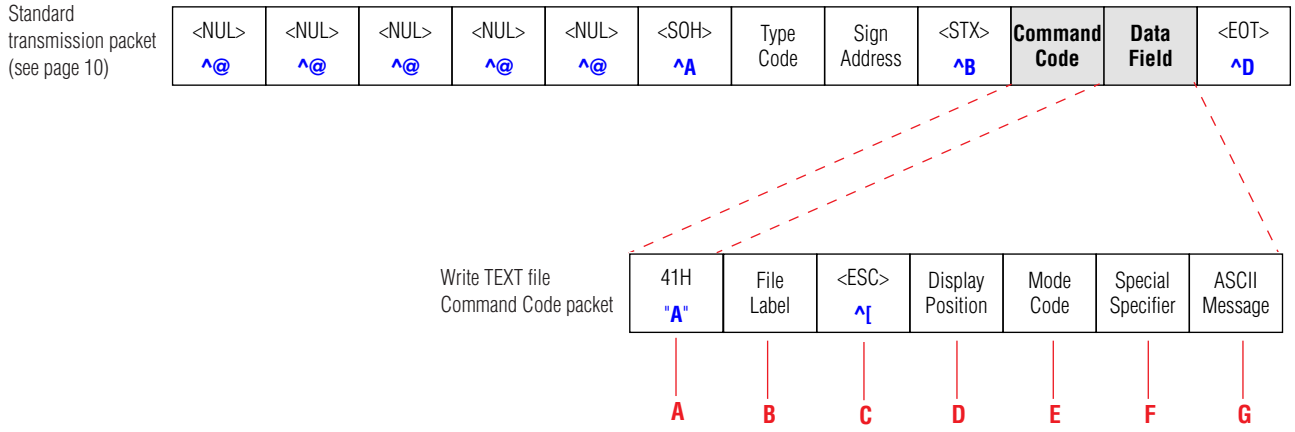
6.1.1 Write TEXT file Command Code — "A" (41H)

SHOW ME

An example of the Write TEXT Command Code is on page 62.

When writing to a TEXT file, the display will blank. After the transmission is over, the unit will begin displaying the last received TEXT file.

Table 12: Write TEXT file transmission packet format



Item	Name			Description
A	**Command Code**			"A" (41H) = Write TEXT file
B	**Data Field**	File Label		One ASCII character that indicates the TEXT file being accessed. See "Appendix A: Valid File Labels" on page 49. If the File Label = "0" (30H), then a Priority TEXT file will be written (see "Priority TEXT files" on page 20).
C		[1]TEXT file data format	Mode Field (optional)	<ESC> (1BH) always starts the Mode Field.
D				A single ASCII character that defines the line position on a multi-line sign: " " 20H Middle Line — Text centered vertically. "" 22H Top Line — Text begins on the top line of the sign and the sign will use all its lines minus 1 in order to display the text. For example, a 6-line sign will allow a maximum of 5 lines (6 minus 1) for the Top Position. The Top/Bottom Line break will remain fixed until the next Middle or Fill position is specified. "&" 26H Bottom Line — The starting position of the Bottom Line(s) immediately follows the last line of the Top Line. For example, a 6-line sign with 3 lines of text associated with the Top Line would start the Bottom Line text on the 4th line of the sign. "0" 30H Fill — The sign will fill all available lines, centering the lines vertically. "1" 31H Left — Text begins on the left side of the sign and the sign will use all its lines minus 1 in order to display the text (Alpha 3.0 protocol only). "2" 32H Right — Text begins on the left side of the sign and the sign will use all its lines minus 1 in order to display the text (Alpha 3.0 protocol only). NOTE: On one-line signs, the Display Position is irrelevant, but it still must be included.
E				A single ASCII character that represents a "mode" which is a way of displaying an ASCII message. See "Standard Modes" on page 88.
F				(Only required when preceding Mode Code is "n" for SPECIAL.) See "Special Modes" on page 88 and "Special Graphics" on page 89.
G			ASCII Message	The actual text to be displayed on a sign. This can also include special Control Characters (see "Appendix G: Alpha protocol ASCII table" on page 80). NOTE: An ASCII Message cannot be displayed if the previous field (Special Specifier) is a Special Graphic. To display text after a Special Graphic, another Mode Field must be used.

NOTE: [1]This can be repeated until the sign's internal memory limit is reached. This limit is dependent on the individual sign.

6.1.2 Read TEXT file Command Code — "B" (42H)

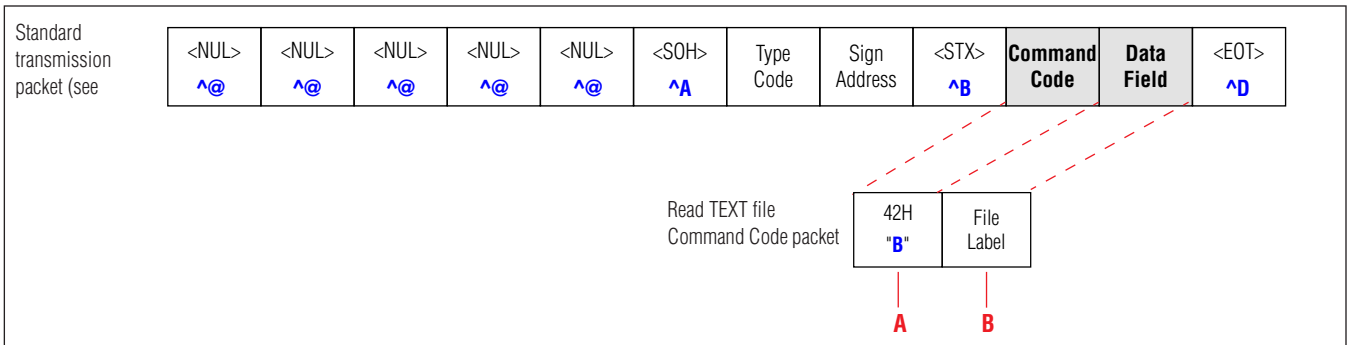
SHOW ME

An example of the Read TEXT file packet is on page 63.

This command asks a sign to send back a TEXT file.

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, *only one sign at a time should be written to or read from.*

Table 13: Read TEXT file transmission packet format



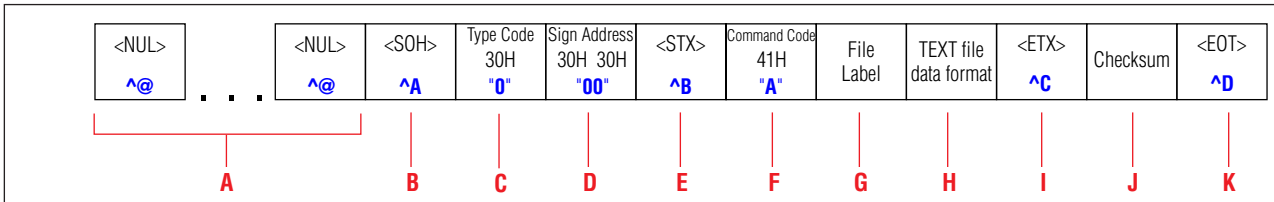
Item	Name		Description
A	**Command Code**		"B" (42H) = Read TEXT file
B	**Data Field**	File Label	One ASCII character that indicates the TEXT file being accessed. See "Appendix A: Valid File Labels" on page 49. If the File Label = "0" (30H), then the Priority TEXT file will be read (see "Priority TEXT files" on page 20).

SHOW ME

An example of the Read TEXT file sign response packet is on page 63.

Following the Read TEXT file Command Code, a sign will respond with the following:

Table 14: Read TEXT file sign response packet format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"A" (41H) is returned by the sign. (This is the Write TEXT Command Code.)
G	File Label	One ASCII character that indicates the TEXT file being accessed. See "Appendix A: Valid File Labels" on page 49.
H	TEXT file data format	See Table 12, "Write TEXT file transmission packet format," on page 18.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

6.1.3 Priority TEXT files

A Priority TEXT file is a special 125-byte message that does not need to be *configured* because it always exists on a sign. When data is written to a Priority TEXT file, all other TEXT files that are currently running will stop being displayed. A Priority TEXT file is created when a File Label = "0" (30H).

SHOW ME

Examples of Priority TEXT file packets are on page 67.

The Priority TEXT file will run all by itself until:

- a Write Priority TEXT file without any ASCII Message is sent
- a serial write to the Run Time table takes place
- a serial write to the Run Day table takes place
- an IR keyboard is pointed at the sign and the **PROG** key is pressed

Once a Priority TEXT file stops running, the sign will begin running the other TEXT files.

6.2 SPECIAL FUNCTION commands

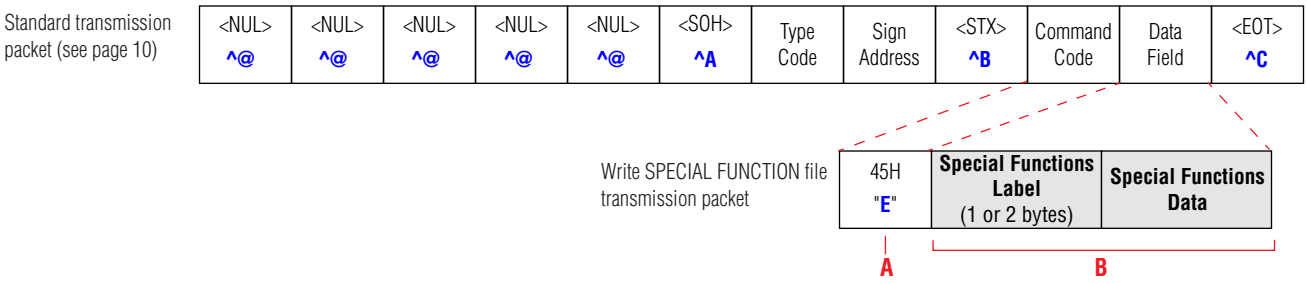
There are a number of special function commands which give the user additional information and control of the sign.

6.2.1 Write SPECIAL FUNCTION Command Code — "E" (45H)

SHOW ME

An example of the Write SPECIAL FUNCTIONs packet is on page 68.
Examples of Set Memory Configuration start on page 70.

Table 15: Write SPECIAL FUNCTION Command Code format — "E" (45H)



Item	Name	Description
A	Command Code	"E" (45H) = Write SPECIAL FUNCTION command
B	**Special Functions Label**	**Special Functions Data**
	" " 20H	**Set Time of Day** — four ASCII digits used to set the time of day (24-hour format) clock in a sign. The following format is used: HhMm where: H = ASCII digit representing hours (10's digit) h = ASCII digit representing hours (1's digit) M = ASCII digit representing minutes (10's digit) m = ASCII digit representing minutes (1's digit) To display the time, see "Control codes (00 – 1FH)" on page 80.
	"!" 21H	**Enable/Disable a Sign's Speaker** — two ASCII characters: "00" 30H + 30H = enable speaker "FF" 46H + 46H = disable speaker **(default)**

	"$" 24H	**Clear Memory/Set Memory Configuration** — To Clear Memory just use "E$". To Set Memory Configuration 11 (or multiples thereof) ASCII characters are used to set a sign's Memory Configuration table. Memory Configuration is a sign's internal battery-backed up RAM directory. A message file cannot be written until a Memory Configuration is written first — *unless the file is a Priority TEXT file or the default TEXT file "A"*. Also, whenever a Memory Configuration is written, the previous table is overwritten. Memory Configuration uses the following format: FTPSIZEQQQQ where:
		Repeat for *each* file to be configured.
		F = One ASCII character that represents the File Label. For valid File Labels, see "Appendix A: Valid File Labels" on page 49.
		T = One ASCII character that represents the file type. Valid file types are:
		"A" 41H = TEXT file
		"B" 42H = STRING file
		"D" 43H = DOTS PICTURE file
		P = One ASCII character that presents the keyboard protection status, either
		"U" 55H = Unlocked. Means that the file can be accessed via an IR keyboard.
		"L" 4CH = Locked. Means that the file can not be accessed via an IR keyboard.
		(For a STRING file, "L" *must* be selected.)
		[1]SIZE = Four ASCII characters that represent the hexadecimal file size in bytes of a TEXT or STRING file. For a DOTS PICTURE file, the first two bytes = # pixel rows and the last two bytes = the # of pixel columns in the picture.
		QQQQ = Four ASCII hexadecimal characters whose format depends on file type used:
		• For a TEXT file, the first two characters represent the file's Start Time and the last two characters represent the Stop Time. For valid entries, see "Appendix B: Valid Start and Stop times" on page 50.
		• For a STRING file, use "0000" as place holders because these four characters have no special meaning
		• For a DOTS PICTURE file, this represents the Color Status. Valid entries are "1000" = monochrome, "2000" = 3-color, "4000" = 8-color (The "E8" command is used for RGB signs. See page 26.)

Table 15: Write SPECIAL FUNCTION Command Code format — "E" (45H)

B (cont)	"$$$$"	24H (four)	**Clear Memory and Compact Flash** (Alpha 3.0 protocol only) — clears a sign's memory and its compact flash.
	"&"	26H	**Set Day of Week** — one ASCII digit that represents the day of the week. A sign will automatically update the day of the week at 12:00 am every day. Valid entries are "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday
	"'"	27H	**Set Time Format** — one ASCII character that represents how time is shown on a sign. Valid entries are "S" 53H = Standard am/pm format **(default)** "M" 4DH = 24-hour (military) time
	"("	28H	**Generate Speaker Tone** — [2]one to five ASCII characters which generate a tone from a sign's speaker. Valid entries are [3]"A" 41H = Turn sign speaker on. [3]"B" 42H = Turn sign speaker off. [4]"0" 30H = Generate a continuous tone for about 2 seconds [4]"1" 31H = Generate three, short beeps (total time about 2 seconds) [5]"2" 32H = Generate a programmable tone according to this format: FFDR where FF = Two ASCII hexadecimal characters that represent a speaker frequency. Valid entries are from "00" through "FE". D = One ASCII hexadecimal character that represents the duration of a tone in 0.1 second increments. Valid entries are from "1" through "F". R = One ASCII hexadecimal character that represents the number of times a tone is repeated. Valid entries are from "0" through "F". "3" 33H = (Alpha 2.0 and 3.0 protocols only) See "Store a programmable sound" on page 99. "4" 34H = (Alpha 2.0 and 3.0 protocols only) See "Trigger a programmable sound" on page 100.
	")"	29H	**Set Run Time Table** — [6]five ASCII characters used to set the start and stop times in the Run Time table in the following format: FQQQQ where F = One ASCII character that represents a TEXT File Label. QQQQ = Four ASCII hexadecimal characters. The first two characters represent a file's Start Time and the last two characters represent a file's Stop Time. For valid entries, see "Appendix B: Valid Start and Stop times" on page 50. These values overwrite the values currently stored in the Memory Configuration table.
	"+"	2BH	**Display Text at XY Position** — allows up to 250 characters to be displayed at a specified location on an ALPHAVISION character matrix sign using the following format: SFXYT where: XYT can repeat which permits *many* messages to be displayed in *many* different locations. Use DC2 (12H) as a delimiter after each XYT sequence except for the last sequence. S = Enable/Disable character where: "+" 2BH = Enable XY positioning. While in this mode, all other transmissions are ignored. For example, a write to a text file will be ignored. "-" 2DH = Disable XY positioning F = the File Label. Use "+" 2BH. X = Two ASCII decimal digit characters from "00" to "99" that represent the character position in a sign row to display the text. If X exceeds its limit, it wraps around to the next line or character. Y = Two ASCII decimal digit characters from "00" to "99" that represent the line to display the text. If Y exceeds its limit, it wraps around to the next line or character. T = Up to 250 ASCII characters that represent the message to be displayed. Control codes for color selection, font selection for 5- or 7-high characters, and flash characters are allowed. All other control codes will be ignored. NOTE: To enable XY positioning, first send "E+" or send the first message twice. NOTE: To be able to flash characters, an enable message (STX,"E+",EOT) must be sent at regular intervals. NOTE: See "Displaying text at XY position examples" on page 76 for examples of XY positioning.
	","	2CH	**Soft Reset** — causes a soft reset of the sign. There is no data in this field. A soft reset causes the sign to go through its power-up diagnostics. Memory will not be cleared (non-destructive).

Table 15: Write SPECIAL FUNCTION Command Code format — "E" (45H)

B (cont)	"." 2EH	**Set Run Sequence** — from 3 to 130 ASCII characters that specify the Run Sequence. From 1 to 128 TEXT files can be set using the following format: KPF where: *F* repeats for *each* file to be configured. K = One ASCII character that represents the type of Run Sequence order: "T" 54H = All subsequent TEXT File Labels in the Run Sequence will run according to their associated *times* **(default)**. "S" 53H = All subsequent TEXT File Labels in the Run Sequence will run *in order* regardless of each file's run time. "D" 44H = All subsequent TEXT file labels in the Run Sequence will run according to their associated times. Then when the file reaches an "off time", the file will be deleted. P = One ASCII character that represents the keyboard protection status: "U" 55H = Unlocked. This allows the Run Sequence to be changed from a hand-held IR keyboard **(default)**. "L" 4CH = Locked. This makes the Run Sequence inaccessible from a hand-held IR keyboard. F = One ASCII character that represents a valid TEXT File Label (See "Appendix A: Valid File Labels" on page 49). If a File Label is invalid or does not exist, the next File Label will be processed. Up to 128 File Labels can be in a Run Sequence.
	"/" 2FH	**Set Dimming Register** — four ASCII characters that are used to control sign dimming in the following format: WWww where WW = Two ASCII hexadecimal characters that represent *when* a sign should dim.: "00" = no dimming "01 to "15" is a range where "01" = dark outside and "15" = bright outside ww = Two ASCII hexadecimal characters that represent the *level of brightness*: "00" = 100% brightness "01" = 86% brightness "02" = 72% brightness "03" = 58% brightness "04" = 44% brightness NOTE: If dimming is not desired, set WWww = "0000" **(default)**. NOTE: Dimming is only available on some signs.
		Set Dimming Times — four ASCII characters that are used to control sign dimming in the following format: WWww where WW = Two ASCII hexadecimal characters that represent the Start Time of when a sign should dim. ww = Two ASCII hexadecimal characters that represent the Stop Time of when a sign should stop dimming. NOTE: If dimming is not desired, set WWww = "0000" **(default)**. NOTE: Dimming times is only available on some signs.

Table 15: Write SPECIAL FUNCTION Command Code format — "E" (45H)

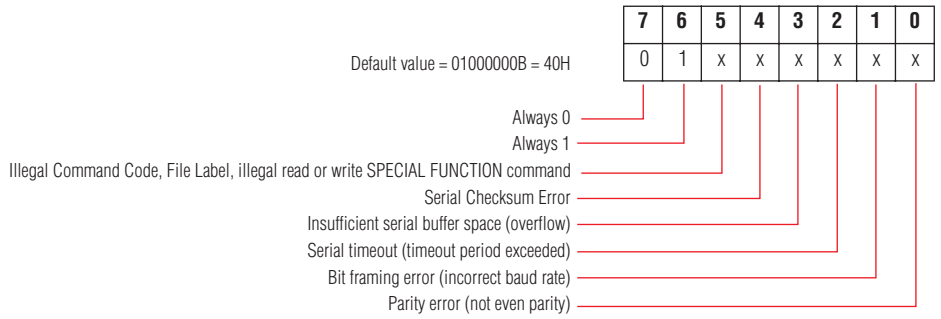
<table>
<tr><td rowspan="2">B
(cont)</td><td>"2" 32H</td><td>Set Run Day Table — three ASCII characters that are used for each TEXT File Label to set the start and stop days in the Run Day Table in the following format: FSs where
F = One ASCII character that represents the TEXT File Label. For valid File Labels, see "Appendix A: Valid File Labels" on page 49.

S = One ASCII hexadecimal character that represents run start day for the TEXT file specified by F. Valid start day characters are:
"0" 30H = Daily
"1" 31H = Sunday
"2" 32H = Monday
"3" 33H = Tuesday
"4" 34H = Wednesday
"5" 35H = Thursday
"6" 36H = Friday
"7" 37H = Saturday
"8" 38H = Monday-Friday
"9" 39H = Weekends
"A" 41H= Always
"B" 42H = Never

s = One hexadecimal character that represents the run stop day for the TEXT file specified by F. Valid stop day characters are:
"1" 31H = Sunday
"2" 32H = Monday
"3" 33H = Tuesday
"4" 34H = Wednesday
"5" 35H = Thursday
"6" 36H = Friday
"7" 37H = Saturday

NOTE: The stop day is required even though the start day may cover multiple days (e.g., Daily, Never, etc.) In this case, the stop day is ignored.</td></tr>
<tr><td>"4" 34H</td><td>Clear Serial Error Status Register — one ASCII character that is used to clear the Serial Error Status Register to its default value of 40H.

This register is set to its default value (40H or 01000000B) for the following Command Codes: (1) Read Serial Error Status Register, (2) Network Query, or (3) Clear Serial Error Status Register.

Serial Error Status Register
 

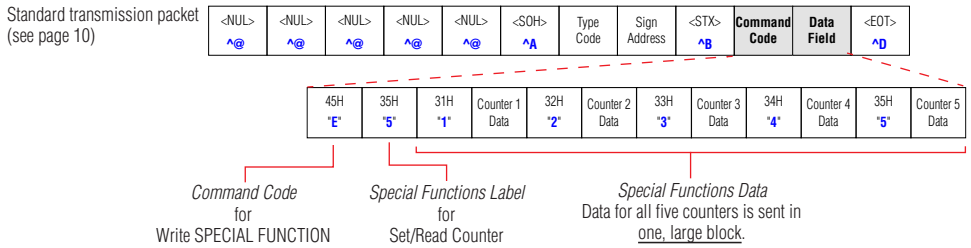
NOTE: This command should be used as the first command in a nested transmission frame to be sure that all subsequent serial errors or lack of serial errors recorded are applicable to the nested frame. Also, the last command in a nested transmission frame should be a Serial Error Status read (see the "*" command in Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28).

NOTE: Parity error (not even parity) is not used on most signs.</td></tr>
</table>

Table 15: Write SPECIAL FUNCTION Command Code format — "E" (45H)

<table>
<tr>
<td>B
(cont)</td>
<td>"5" 35H</td>
<td>
Set Counter — used to set one or more of the five internal timers available on counter-equipped signs. Data for all five counters must be sent as one, large block, in the following format:

NOTE: Even if you are only setting one counter, data must be sent to the other counters as well.



The format of Counter 1 Data, Counter 2 Data, etc from above is as follows:
BBTTttSSSSSSSSiiiiiiiiVVVVVVVVttttttttFFmmHH where:

BB = Two ASCII hexadecimal characters that set the 8 bits of the Counter Control Byte, whose default value is 01100100B (64H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Counter Control Byte. For example, to set the Counter Control Byte to its default value of 64H, an ASCII "6" (36H) and an ASCII "4" (34H) would be sent. Here's what the 8 bits of the Counter Control Byte mean:
bit 7 — 1 = counter on, 0 = counter off (default = 0)
bit 6 — 1 = increment, 0 = decrement (default = 1)
bit 5 — 1 = count minutes, 0 = don't count minutes (default = 1)
bit 4 — 1 = count hours, 0 = don't count hours (default = 0)
bit 3 — 1 = count days, 0 = don't count days (default = 0)
bit 2 — 1 = weekends on, 0 = weekends off (default = 1)
[7]bit 1 — 1 = Auto Reload ON, Auto Reload OFF (default = 0)
bit 0 — 0 (default = 0)

[8]TT = Two ASCII hexadecimal characters representing the Counter Start Time. See "Appendix B: Valid Start and Stop times" on page 50. (default = "FF" for Always)

[9]tt = Two ASCII hexadecimal characters representing the Counter Stop Time. See "Appendix B: Valid Start and Stop times" on page 50. The Counter Stop Time is ignored when the Counter Start Time = "FF" for Always. (default = "00")

[10]SSSSSSSS = Eight ASCII characters that represent an 8-digit BCD Counter Start Value. Valid values are from "00000000" to "99999999". (default = "00000000")

[10]iiiiiiii = Eight ASCII characters that represent an 8-digit BCD Counter Change Value. This is the number that is either incremented or decremented according to bit 6 of the Counter Control Byte. Valid values are from "00000000" to "99999999". (default = "00000001")

[10]VVVVVVVV = Eight ASCII characters that represent an 8-digit BCD Current Counter Value. Valid values are from "00000000" to "99999999". (default = "00000000")

[10]tttttttt = Eight ASCII characters that represent an 8-digit BCD Counter Target Value. When this value equals the Current Counter Value, from 0 to 5 Target file messages will be sent according to parameter FF (below). Valid values are from "00000000" to "99999999". (default = "00000000")

FF = Two ASCII hexadecimal characters that represent the Target File Byte whose default value is 00000000 (00H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Target File Byte. For example, to set a value of 1FH, an ASCII "1" (31H) and an ASCII "F" (46H) would be sent. Here's what the 8 bits of the Target File Byte mean:
bit 7 — 0 (default = 0)
bit 6 — 0 (default = 0)
bit 5 — 0 (default = 0)
bit 4 — Target File 1: 1 = enabled, 0 = disabled (default = 0)
bit 3 — Target File 2: 1 = enabled, 0 = disabled (default = 0)
bit 2 — Target File 3: 1 = enabled, 0 = disabled (default = 0)
bit 1 — Target File 4: 1 = enabled, 0 = disabled (default = 0)
bit 0 — Target File 5: 1 = enabled, 0 = disabled (default = 0)

[11]mm = Two ASCII hexadecimal characters that set the Counter Change Minutes Synchronization. Valid values are from "00" to "3B" (00 - 59). (default = "00")

[12]HH = Two ASCII hexadecimal characters that set the Counter Change Hours Synchronization. Valid values are from "00" to "17" (00 - 23) where "00" = 12 am, "01" = 1 am, and so on. (default = "00")
</td>
</tr>
</table>

Table 15: Write SPECIAL FUNCTION Command Code format — "E" (45H)

B (cont)	"7"	37H	**Set Serial Address** — Two ASCII hexadecimal characters used to set a sign's serial address. Valid values are from "00" through "FF". **(default = "00")** NOTE: If the serial address has been set using a hardware DIP switch to an address other than "00", the DIP switch address will override the address set here — once power to the sign has been cycled.
	"8"	38H	[13]**Set LARGE DOTS PICTURE Memory Configuration** — a data stream of 24 ASCII characters that repeats for each file configured in a sign. The format for this data stream is as follows: FFFFFFFFFPRRRRCCCCccrrrr where [14]FFFFFFFFF = A 9-character file name P = One ASCII character that represents the keyboard protection status. Valid values are: "U" 55H = Unlocked. This allows the DOTS PICTURE file to be changed from a hand-held IR keyboard **(default)**. "L" 4CH = Locked. This makes the DOTS PICTURE file inaccessible from a hand-held IR keyboard. RRRR = Four ASCII hexadecimal digits that represent the number of pixel rows. Leading zeroes are required (e.g., "0040" = 64 rows). CCCC = Four ASCII hexadecimal digits that represent the number of pixel columns. Leading zeroes are required (e.g., "0060" = 96 columns). cc = Two ASCII hexadecimal digits representing the number of colors in the LARGE DOTS PICTURE. Valid values are: "01" = monochrome DOTS PICTURE "02" = tricolor DOTS PICTURE "08" = RGB DOTS PICTURE (Alpha 3.0 protocol only) rrrr = reserved for future use. Four ASCII zeroes are required — "0000".
	"9"	39H	**Append to LARGE DOTS PICTURE file Memory Configuration** — allows appending to the LARGE DOTS PICTURE file Memory Configuration. The data format is the same as the LARGE DOTS PICTURE file Memory Configuration data format.
	":"	3AH	**Set Run File Times** (Alpha 2.0 and 3.0 protocols only) — see "Set Run File Time" on page 100.
	";"	3BH	**Set Date** — six ASCII characters that are used to set the date in the following format: mmddyy where mm = Two ASCII digits that represent the month dd = Two ASCII digits that represent the day [15]yy = Two ASCII digits that represent the year
	"<"	3CH	**Program Custom Character Set** (Alpha 2.0 and 3.0 protocols only) — see "Custom character sets" on page 104.
	">"	3EH	**Set Automode Table** (Alpha 2.0 and 3.0 protocols only) — see "Automode table" on page 107.
	"@"	3FH	**Set Dimming Control Register** (Alpha 2.0 and 3.0 protocols only) — see "Dimming Control Register" on page 108.
	"C"	43H	**Set Color Correction** (Alpha 3.0 protocol. AlphaEclipse 3600 sign only.) — sets color correction for an RGB sign where "0" 30H = color correction off. "1" 31H = RGB color correction **(default)**. "2" 32H = red gamma color correction for mono-color (red or amber) signs. EXAMPLE: <SOH>"Z00"<STX>"EC2"<EOT> *Turn on red gamma color correction.*
	"T"	54H	**Set Temperature Offset** — allows for improvement in temperature accuracy as displayed on message centers which support temperature display (790i, 460i, 440i, and 430i). The data format is as follows: SO where S = One ASCII character that stands for the sign of the temperature offset. Valid values are: "+" 2BH = a positive offset "-" 2DH = a negative offset O = One ASCII hexadecimal character that stands for the temperature offset. Valid values are from "0" through "9". *For a Solar sign*, an actual temperature is sent, not an offset. The Solar sign itself computes the offset. The data format for a Solar sign is as follows: SO where: S = One ASCII character that stands for the sign of the temperature. Valid values are: "+" 2BH = a positive temperature "-" 2DH = a negative temperature O = Three ASCII hexadecimal characters that stand for an actual temperature.
	"U1"	55H 31H	**Set Unit Columns and Rows** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U2"	55H 32H	**Set Unit Run Mode** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U3"	55H 33H	**Set Unit Serial Address** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U4"	55H 34H	**Set Unit Serial Data** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U5"	55H 35H	**Set Unit Configuration** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.

Table 15: Write SPECIAL FUNCTION Command Code format — "E" (45H)

B (cont)	"U7"	55H 37H	**Set Unit Internal Network** (Alpha 3.0 protocol only. AlphaEclipse 3600 sign only.) — allows access to the sign's internal network in the following format: HD where H = sign header packet D = data packet for sign's internal network This is the header packet format for the turbo adapter or RGB driver board: Type code — one ASCII byte "G" (turbo adapter) or "F" (RGB driver board) Serial address — two ASCII bytes that represent the hexadecimal address Turbo channel — two ASCII bytes that represent the turbo adapter channel number in hexadecimal NOTE: There is a 1-second wait for the peripheral device to respond back.
	"U8"	55H 38H	**Set Unit Slave Device** (Alpha 3.0 protocol only. AlphaEclipse 3600 sign only.) — displays the message specified in the File Label of this command on the slave sign. EXAMPLE: <SOH>"Z00"<STX>"EU8A"<EOT> *Displays the message in File Label= "A" on the slave sign.*
	"U9"	55H 39H	**Set Unit Internal Network** (Alpha 3.0 protocol only. AlphaEclipse 3600 sign only.) — Same as "U7" except there is no 1-second delay waiting for the peripheral device to respond.
	"UN"	55H 4EH	**Write Unit Register** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"s"	73H	**Enable/Disable ACK/NAK Response** (Alpha 2.0 and 3.0 protocols only) — see "Enable/Disable ACK/NAK response" on page 111.

NOTE: [1] The sum of all the file sizes (except for SMALL DOTS PICTURE and LARGE DOTS PICTURE files) plus 11 bytes of overhead for each file should not exceed the total amount of available memory in the pool. A value of "0000" is a valid SIZE for the last file in the Memory Configuration only if this last file is a TEXT file. This assigns all remaining memory to the file.

[2] When sending nested frames, the tone generation command must be the last transmission frame because the sign's serial port is disabled (and cannot receive any data) while a tone is generated. A tone generation command can never be part of any type of READ command, except on the AlphaPremiere sign, which can tone and receive at the same time.

[3] This command should not be used with the standard speaker/piezo alarm provided in the sign as it may damage the sign.

[4] Wait a minimum of 3 seconds before transmitting more data to the sign, except on the AlphaPremiere sign, which can tone and receive at the same time.

[5] Wait until the programmable tone has finished before transmitting more data to the sign, except on the AlphaPremiere sign, which can tone and receive at the same time.

[6] This 5-byte field repeats for each TEXT file configured in the sign. Not all TEXT files need to be updated, only those that require modification.

[7] When the Counter Target Value has been reached, Auto Reload ON will put into the Counter Start Value in Current Counter Value.

[8] Time codes "FD" and "FE" are not valid as Counter Start Times.

[9] Time codes "FD", "FE", and "FF" are not valid as Counter Stop Times.

[10] Leading 0's must be sent if the value is less than 8 digits long. For example, "256" would be sent as "00000256".

[11] This value is used when the Counter Control Byte is set to count hours or days. If minutes are being counted, this value is ignored. However, a value must still be supplied.

[12] This value is used when the Counter Control Byte is set to count days. If minutes or hours are being counted, this value is ignored. However, a value must still be supplied.

[13] See LARGE DOTS PICTURE Memory Configuration *only* applies to Full Matrix ALPHAVISION, Series 7000, AlphaEclipse, and AlphaPremiere signs.

[14] If a file name is less than 9 characters, it must be padded with leading spaces (20H) so that the total number of characters is always nine.

[15] For Alpha protocol version 2.0 and greater, the year (yy) is windowed as follows: "00 to "96" = 2000 to 2096. "97" to "99" = 1997 to 1999.

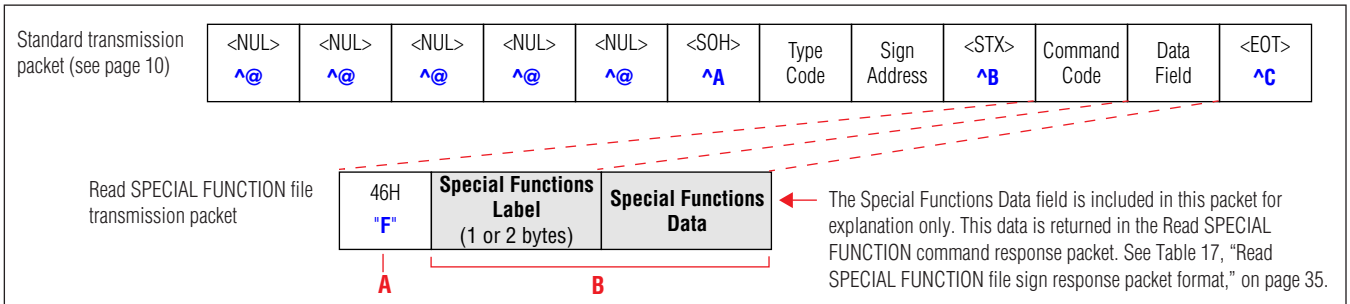
6.2.2 Read SPECIAL FUNCTION Command Code — "F" (46H)

SHOW ME

An example of the Read SPECIAL FUNCTION command is on page 68.

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a *unique* Serial Address. Also, only one sign at a time should be accessed or read from.

Table 16: Read SPECIAL FUNCTION Command Code format — "F" (46H)



Item	Name	Description
A	Command Code	"F" (46H) = Read SPECIAL FUNCTION file
B	**Special Functions Label**	**Special Functions Data** (This data is returned in a Read SPECIAL FUNCTION file sign response. See Table 17, "Read SPECIAL FUNCTION file sign response packet format," on page 35)
	" " 20H	**Read Time of Day** — returns four ASCII digits that represent the time of day (24-hour format) clock in a sign. The following format is used: HhMm where: H = ASCII digit representing hours (10's digit) h = ASCII digit representing hours (1's digit) M = ASCII digit representing minutes (10's digit) m = ASCII digit representing minutes (1's digit) To display the time on a sign, see the "Control characters" in "Appendix G: Alpha protocol ASCII table" on page 80.
	"!" 21H	**Read Speaker Status** — returns two ASCII characters: "00" 30H + 30H = speaker enabled "FF" 46H + 46H = speaker disabled **(default)**
	""" 22H	**Read General Information** — returns 28 to 59 ASCII characters in the following format (see also "Read Serial Error Log" on page 30): <NUL>FFFFFFFFfMmYyHhNnRSSPOOL,pool where [1]<NUL> = 00H FFFFFFFF = Eight ASCII characters that stand for the firmware installed in the sign f = One ASCII character that stands for the firmware revision letter MmYy = Four ASCII digits that stand for the release date of the firmware. For example, firmware released in January 1993 would be represented as "0193". HhNn = Four ASCII digits that represent the time of day (24-hour format) clock in a sign. The format is the same used for **Read Time of Day** above. R = One ASCII character that represents how time is displayed on a sign where: "S" 53H = standard am/pm format **(default)** "M" 4DH = 24-hour (or military) time SS = Speaker status where: "00" 30H + 30H = speaker enabled "FF" 46H + 46H = speaker disabled **(default)** POOL,pool = Memory Pool where: POOL = Four-digit ASCII hexadecimal number that represents the *total size* of the Memory Pool in bytes. The most significant digit is first. , = 2CH (a comma) pool = Four-digit ASCII hexadecimal number that represents the *unused* portion of the Memory Pool in bytes. The most significant digit is first. NOTE: General Information is most useful as a source of troubleshooting information.

Table 16: Read SPECIAL FUNCTION Command Code format — "F" (46H)

B (cont)	"#"	23H	**Read Memory Pool Size** — returns nine ASCII characters that indicate the total size and available amount of the Memory Pool. The Memory Pool is a sign's internal battery-backed up RAM that is available for file storage. Any unused memory is assigned to the first TEXT file listed in the Memory Configuration once the sign starts running. The Memory Pool is in the following format: POOL,pool. The format is the same used in **Read General Information** above.
	"$"	24H	**Read Memory Configuration** — returns eleven ASCII characters that represent a sign's Memory Configuration table. Memory Configuration is a sign's internal battery-backed up RAM directory. Memory Configuration uses the following format: FTPSIZEQQQQ where: F = One ASCII character that represents the File Label. For valid File Labels, see "Appendix A: Valid File Labels" on page 49. T = One ASCII character that represents the file type. Valid file types are: "A" 41H = TEXT file "B" 42H = STRING file "D" 43H = DOTS PICTURE file P = One ASCII character that presents the keyboard protection status, either "U" 55H = Unlocked. Means that the file can be accessed via an IR keyboard. "L" 4CH = Locked. Means that the file can not be accessed via an IR keyboard. [2]SIZE = Four ASCII characters that represent the hexadecimal file size in bytes of a TEXT or STRING file. QQQQ = Four ASCII hexadecimal characters whose format depends on file type used: • For a TEXT file, the first two characters represent the file's Start Time and the last two characters represent the Stop Time. For valid entries, see "Appendix B: Valid Start and Stop times" on page 50. • For a STRING file, "0000" is used as place holders because these four characters have no special meaning. • For a DOTS PICTURE file, this represents the Color Status. Valid entries are "1000" = monochrome DOTS PICTURE "2000" = 3-color DOTS PICTURE "4000" = 8-color DOTS PICTURE RGB signs use "F8" (see page 33)
	"%"	25H	**Memory Dump** — returns multiple nested transmission frames with checksums (see "Nesting with Checksums format" on page 13) in the following order: 1. Time-of-day setting (see **Read Time of Day** above) 2. Memory Configuration (see **Read Memory Configuration** above) 3. Transmission frame of each file (Write TEXT, STRING, or DOTS PICTURE file) in the order it appears in Memory Configuration 4. Run Sequence (see **Read Run Sequence** below) 5. Run Day Table (see **Read Run Day Table** below) 6. Day-of-Week setting (see **Read Day-of-Week** below) 7. Counter Functions (see **Read Counter Functions** below)
	"&"	26H	**Read Day of Week** — returns one ASCII digit that represents the day of the week. A sign will automatically update the day of the week at 12:00 am every day. Valid entries are "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday
	"'"	27H	**Read Time Format** — returns one ASCII character that represents how time is shown on a sign. Valid entries are "S" 53H = Standard am/pm format **(default)** "M" 4DH = 24-hour (military) time
	")"	29H	**Read Run Time Table** — returns the following ASCII characters: LqqqqFQQQQE where: L = "0" 30H which represents the PRIOTITY TEXT File Label. qqqq = Four ASCII hexadecimal characters which show the PRIORITY TEXT file status. There are only two possibilities for this: "FE00" = PRIORITY TEXT file is not running "FF00" = PRIORITY TEXT file is running. [3]F = One ASCII character that represents a TEXT File Label (see "Appendix A: Valid File Labels" on page 49) QQQQ = Four ASCII hexadecimal characters. The first two characters represent a file's Start Time and the last two characters represent a file's Stop Time. For valid entries, see "Appendix B: Valid Start and Stop times" on page 50. These values overwrite the values currently stored in the Memory Configuration table. E = One ASCII hexadecimal character which represents the file enable status. Valid codes are: "0" 30H = file is not currently being displayed "1" 31H = file is currently being displayed

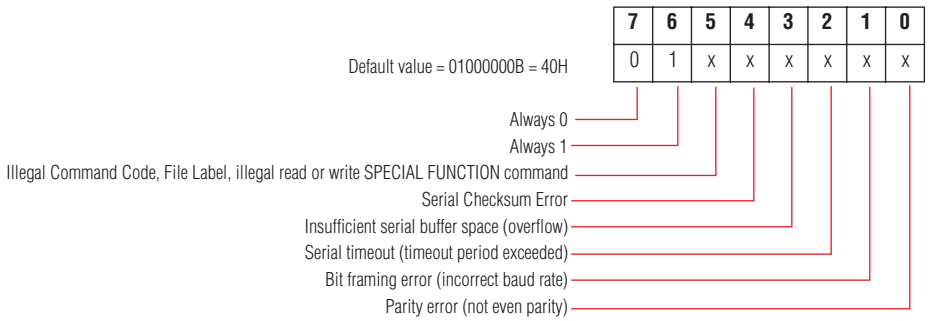
Table 16: Read SPECIAL FUNCTION Command Code format — "F" (46H)

<table>
<tr>
<td rowspan="2">B
(cont)</td>
<td>"*" 2AH</td>
<td>Read Serial Error Status Register — returns one bitmapped ASCII character read from a sign's Serial Error Status Register that represents serial errors recorded by a sign.

This register is set to its default value (40H or 01000000B) for the following Command Codes: (1) Read Serial Error Status Register, (2) Network Query, or (3) Clear Serial Error Status Register.

The sign begins error checking following a valid <SOH> (01H).

The Serial Error Status Register is bitmapped as follows:

Serial Error Status Register
 

NOTE: Errors are OR'd into the Serial Error Status Register. That is, more than one error at a time can be recorded in the register.

NOTE: Parity error (not even parity) is not used on most signs.</td>
</tr>
<tr>
<td>"*L" 2AH 4CH</td>
<td>Read Serial Error Log (Alpha 3.0 protocol only) — returns 256 sets of 4 ASCII characters which represent the last 256 packets sent to the sign, starting with the last packet sent, in the following format (see also "Read General Information" on page 28): ABCD where
A = one ASCII character that stands for the Command Code
B = one ASCII character that stands for the Data Field
C = one ASCII character that stands for the packet status:
"0" 30H — packet ok
"1" 31H — packet overflow error. Expected serial packet was too big.
"2" 32H — packet underflow error. Expected packet was too small.
"3" 33H — packet contains illegal data.
"4" 34H — serial buffer overflow.
"5" 35H — serial send timeout error.
"6" 36H — send resource error. Could not execute the command because of a resource error.
"7" 37H — memory access error. Could not allocate memory or access memory.
"8" 38H — message nesting error. Error in the nesting of packets.
"9" 39H — serial checksum error.
D = one ASCII character that stands for the Serial Error Status Register when the packet was received (see above).

NOTE: In there are less than 256 packets in the log, then four ASCII "0"s ("0000") are used as a placeholder for each missing packet.

EXAMPLE:
If these were the only three packets sent to a sign . . .

<SOH>"Z00"<STX>"AAHello World"<EOT>
Displays "Hello World". Command Code = "A". Data Field (File Label) = "A"

<SOH>"Z00"<STX>"ABMessage B"<ETX>"0000"<EOT>
Displays "Message B". Command Code = "A". Data Field (File Label) = "B"

<SOH>"Z00"<STX>"F*L<EOT>
Read Special Function. Command Code = "F". Data Field (Special Functions Label) = "*"

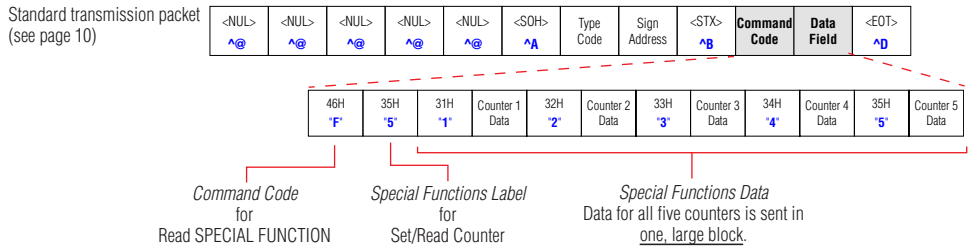
then this is the packet the sign would return (each packet is enclosed in brackets "[" and "]"):

<SOH>"000"<STX>"E*[F*0P][AB9P][AB0@][0000]. . .[0000]"<EOT></td>
</tr>
</table>

Table 16: Read SPECIAL FUNCTION Command Code format — "F" (46H)

B (cont)	"-" 2DH	**Network Query** — returns the unit type, Serial Address, and Serial Error Status Register for each sign on the network. The response from each sign is in the following format: UAAZ where: U = One ASCII character that stands for the unit type of a sign. For valid entries, see "Type Code" in "Standard transmission packet ("1-byte" or "^A") format" on page 10. AA = Two ASCII hexadecimal characters that represent a sign's serial address Z = One ASCII character that represents the Serial Error Status Register of a sign (see above) NOTE: Normally, a Network Query is broadcast to all signs using a "00" in the Sign Address field. When a Network Query is broadcast like this, all signs on the network respond in the following manner: Once the <EOT> is received by a sign, it will respond to the Network Query after a timed interval. This interval is a sum of 1 second plus the product of a sign's address and 0.5 seconds. For example, a sign with an address of 0FH (15), would reply after 1 + (15 x 0.5) = 8.5 seconds. NOTE: If there are two or more signs on a network with the *same* Serial Address, then a Network Query will produce unpredictable results. A response from one of these signs may be garbled because there is no collision detection.
	"." 2EH	**Read Run Sequence** — returns from 3 to 130 ASCII characters that specify the Run Sequence. From 1 to 128 TEXT files will be read in the following format: KPF where: F repeats for *each* file to be configured. K = One ASCII character that represents the type of Run Sequence order: "T" 54H = All subsequent TEXT File Labels in the Run Sequence will run according to their associated *times* **(default)**. "S" 53H = All subsequent TEXT File Labels in the Run Sequence will run *in order* regardless of each file's run time. P = One ASCII character that represents the keyboard protection status: "U" 55H = Unlocked. This allows the Run Sequence to be changed from a hand-held IR keyboard **(default)**. "L" 4CH = Locked. This makes the Run Sequence inaccessible from a hand-held IR keyboard. F = One ASCII character that represents a valid TEXT File Label (See "Appendix A: Valid File Labels" on page 49). If a File Label is invalid or does not exist, the next File Label will be processed. Up to 128 File Labels can be in a Run Sequence.
	"2" 32H	**Read Run Day Table** — returns three ASCII characters that are used for each TEXT File Label to read the start and stop days in the Run Day Table in the following format: FSs where F = One ASCII character that represents the TEXT File Label. For valid File Labels, see "Appendix A: Valid File Labels" on page 49. S = One ASCII hexadecimal character that represents run start day for the TEXT file specified by F. Valid start day characters are: "0" 30H = Daily "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday "8" 38H = Monday-Friday "9" 39H = Weekends "A" 41H= Always "B" 42H = Never s = One hexadecimal character that represents the run stop day for the TEXT file specified by F. Valid stop day characters are: "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday

Table 16: Read SPECIAL FUNCTION Command Code format — "F" (46H)

<table>
<tr>
<td>B
(cont)</td>
<td>"5" 35H</td>
<td>
Read Counter — returns data for all five counters is received as one, large block, in the following format:


The format of Counter 1 Data, Counter 2 Data, and so on from above is as follows:
BBTTttSSSSSSSSiiiiiiiiVVVVVVVVttttttttFFmmHH where:

BB = Two ASCII hexadecimal characters that stand for the 8 bits of the Counter Control Byte, whose default value is 01100100B (64H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Counter Control Byte. Here's what the 8 bits of the Counter Control Byte mean:
bit 7 — 1 = counter on, 0 = counter off (default = 0)
bit 6 — 1 = increment, 0 = decrement (default = 1)
bit 5 — 1 = count minutes, 0 = don't count minutes (default = 1)
bit 4 — 1 = count hours, 0 = don't count hours (default = 0)
bit 3 — 1 = count days, 0 = don't count days (default = 0)
bit 2 — 1 = weekends on, 0 = weekends off (default = 1)
[4]bit 1 — 1 = Auto Reload ON, Auto Reload OFF (default = 0)
bit 0 — 0 (default = 0)

[5]TT = Two ASCII hexadecimal characters representing the Counter Start Time. See "Appendix B: Valid Start and Stop times" on page 50. (default = "FF" for Always)

[6]tt = Two ASCII hexadecimal characters representing the Counter Stop Time. See "Appendix B: Valid Start and Stop times" on page 50. The Counter Stop Time is ignored when the Counter Start Time = "FF" for Always. (default = "00")

[7]SSSSSSSS = Eight ASCII characters that represent an 8-digit BCD Counter Start Value. Valid values are from "00000000" to "99999999". (default = "00000000")

[7]iiiiiiii = Eight ASCII characters that represent an 8-digit BCD Counter Change Value. This is the number that is either incremented or decremented according to bit 6 of the Counter Control Byte. Valid values are from "00000000" to "99999999". (default = "00000001")

[7]VVVVVVVV = Eight ASCII characters that represent an 8-digit BCD Current Counter Value. Valid values are from "00000000" to "99999999". (default = "00000000")

[7]tttttttt = Eight ASCII characters that represent an 8-digit BCD Counter Target Value. When this value equals the Current Counter Value, from 0 to 5 Target file messages will be sent according to parameter FF (below). Valid values are from "00000000" to "99999999". (default = "00000000")

FF = Two ASCII hexadecimal characters that represent the Target File Byte whose default value is 00000000 (00H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Target File Byte. For example, to set a value of 1FH, an ASCII "1" (31H) and an ASCII "F" (46H) would be sent. Here's what the 8 bits of the Target File Byte mean:
bit 7 — 0 (default = 0)
bit 6 — 0 (default = 0)
bit 5 — 0 (default = 0)
bit 4 — Target File 1: 1 = enabled, 0 = disabled (default = 0)
bit 3 — Target File 2: 1 = enabled, 0 = disabled (default = 0)
bit 2 — Target File 3: 1 = enabled, 0 = disabled (default = 0)
bit 1 — Target File 4: 1 = enabled, 0 = disabled (default = 0)
bit 0 — Target File 5: 1 = enabled, 0 = disabled (default = 0)

[8]mm = Two ASCII hexadecimal characters that set the Counter Change Minutes Synchronization. Valid values are from "00" to "3B" (00 - 59). (default = "00")

[9]HH = Two ASCII hexadecimal characters that set the Counter Change Hours Synchronization. Valid values are from "00" to "17" (00 - 23) where "00" = 12 am, "01" = 1 am, and so on. (default = "00")
</td>
</tr>
</table>

Table 16: Read SPECIAL FUNCTION Command Code format — "F" (46H)

B (cont)	"8"	38H	[10]**Read LARGE DOTS PICTURE Memory Configuration** — returns a data stream of 24 ASCII characters that repeats for each file configured in a sign. The format for this data stream is as follows: FFFFFFFFFPRRRRCCCCccrrrr where: [11]FFFFFFFFF = A 9-character file name P = One ASCII character that represents the keyboard protection status. Applies to the AlphaVision, AlphaEclipse, AlphaPremiere, and series 7000 signs. Valid values are: "U" 55H = Unlocked. This allows the DOTS PICTURE file to be changed from a hand-held IR keyboard **(default)**. "L" 4CH = Locked. This makes the DOTS PICTURE file inaccessible from a hand-held IR keyboard. RRRR = Four ASCII hexadecimal digits that represent the number of pixel rows. Leading zeroes are required (e.g., "0040" = 64 rows). CCCC = Four ASCII hexadecimal digits that represent the number of pixel columns. Leading zeroes are required (e.g., "0060" = 96 columns). cc = Two ASCII hexadecimal digits representing the number of colors in the LARGE DOTS PICTURE. Valid values are: "01" = a monochrome DOTS PICTURE "02" = a tricolor DOTS PICTURE "04" = 8-color DOTS PICTURE "08" = RGB DOTS PICTURE rrrr = reserved for future use. Four ASCII zeroes are required — "0000".
	":"	3AH	**Read Run File Times** (Alpha 2.0 and 3.0 protocols only) — see "Reading Run File Time" on page 101.
	";"	3BH	**Read Date** — returns six ASCII characters that are used to set the date in the following format: mmddyy where mm = Two ASCII digits that represent the month dd = Two ASCII digits that represent the day yy = Two ASCII digits that represent the year
	">"	3EH	**Read Automode Table** (Alpha 2.0 and 3.0 protocols only) — see "Automode table" on page 107.
	"C"	43H	**Read Color Correction** (Alpha 3.0 protocol. AlphaEclipse 3600 sign only.) — returns a single ASCII digit where "0" 30H = color correction off. "1" 31H = RGB color correction **(default)**. "2" 32H = red gamma color correction for mono-color (red or amber) signs. EXAMPLE: <SOH>"Z00"<STX>"FC"<EOT> *Reads current color correction.*
	"L"	4CH	**Read Temperature Log** (Alpha 2.0 and 3.0 protocols only) — see "Temperature Logging" on page 112.
	"T"	54H	**Read Temperature Offset** — returns two ASCII characters in the following format: SO where: S = One ASCII character that stands for the sign of the temperature offset. Valid values are: "+" 2BH = a positive offset "-" 2DH = a negative offset O = One ASCII hexadecimal character that stands for the temperature offset. Valid values are from "0" through "9". *For a Solar sign*, an actual temperature is read, not an offset. The Solar sign itself computes the offset. The data format for a Solar sign is as follows: SO where: S = One ASCII character that stands for the sign of the temperature. Valid values are: "+" 2BH = a positive temperature "-" 2DH = a negative temperature O = Three ASCII hexadecimal characters that stand for an actual temperature.
	"U1"	55H 31H	**Read Unit Columns and Rows** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U2"	55H 32H	**Read Unit Run Mode** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U3"	55H 33H	**Read Unit Serial Address** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U4"	55H 34H	**Read Unit Serial Data** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U5"	55H 35H	**Read Unit Configuration** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.
	"U6"	55H 36H	**Read Unit Register** (Alpha 2.0 and 3.0 protocols only) — see "Set Unit commands" on page 118.

Table 16: Read SPECIAL FUNCTION Command Code format — "F" (46H)

B (cont)	"v" 76H	**Read Firmware Revisions Command** (Alpha 3.0 protocol only)— reads comma-delimited firmware and FPGA part numbers in the following format: ABCDEFGHIJ where: A = firmware part number in ASCII followed by a comma B = FPGA part number in ASCII followed by a comma C = FPGA controller board part number in ASCII followed by a comma D = FPGA turbo board part number in ASCII followed by a comma E = backup FPGA part number in ASCII followed by a comma F = backup controller board part number in ASCII followed by a comma G = backup turbo board part number in ASCII H = boot code version I = Controller FPGA version AA = major revision ("00" - "FF") BB = minor revision ("00" - "FF") C = series letter ("A" - "Z") DD = build revision ("00" - "FF") J = Turbo FPGA version AA = major revision ("00" - "FF") BB = minor revision ("00" - "FF") C = series letter ("A" - "Z") DD = build revision ("00" - "FF") NOTE: This command only applies to AlphaPremiere and AlphaEclipse signs.

NOTE: [1] This byte is transmitted only on some signs.

[2] The sum of all the file sizes (except for SMALL DOTS PICTURE and LARGE DOTS PICTURE files) plus 11 bytes of overhead for each file should not exceed the total amount of available memory in the pool. A value of "0000" is a valid SIZE for the last file in the Memory Configuration only if this last file is a TEXT file. This assigns all remaining memory to the file.

[3] The last 6 bytes (FQQQQE) repeat for each TEXT file configured in the sign (with the exception of the PRIOTITY TEXT file which preceded this field.

[4] When the Counter Target Value has been reached, Auto Reload ON will put into the Counter Start Value in Current Counter Value.

[5] Time codes "FD" and "FE" are not valid as Counter Start Times.

[6] Time codes "FD", "FE", and "FF" are not valid as Counter Stop Times.

[7] Leading 0's must be sent if the value is less than 8 digits long. For example, "256" would be sent as "00000256".

[8] This value is used when the Counter Control Byte is set to count hours or days. If minutes are being counted, this value is ignored. However, a value must still be supplied.

[9] This value is used when the Counter Control Byte is set to count days. If minutes or hours are being counted, this value is ignored. However, a value must still be supplied.

[10] Read LARGE DOTS PICTURE Memory Configuration *only* applies to Full Matrix AlphaVision, AlphaEclipse, AlphaPremiere, and Series 7000 signs.

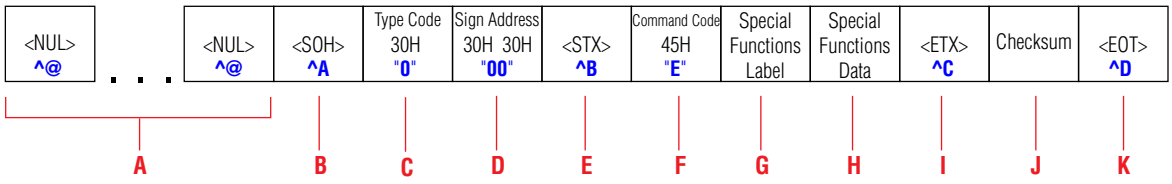
[1l] If a file name is less than 9 characters, it must be padded with leading spaces (20H) so that the total number of characters is always nine.

SHOW ME

An example of the Read SPECIAL FUNCTION file response packet is on page 69.

Following the Read SPECIAL FUNCTION file Command Code, a sign will respond with the following:

Table 17: Read SPECIAL FUNCTION file sign response packet format



Item	Name	Description
A	<NUL>	Twenty <NUL> (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"E" (45H) is returned by the sign. (The Write SPECIAL FUNCTIONs Command Code.)
G	Special Functions Label	One ASCII character that indicates the SPECIAL FUNCTION being accessed. See Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21 and Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.
H	Special Functions Data	See Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21. and Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

6.3 STRING file commands

SPECIAL NOTE

For more information on using STRING files, see "Appendix D: STRING file notes" on page 52.

STRING files are used to store short ASCII sets of characters which may be "called up" from a TEXT file. The main purpose of a STRING file is to display frequently changing information. When writing STRING files to a message center, the display will not blank as it does when writing TEXT files. This is because the STRING file data is buffered and TEXT file internal Checksum does not change. *Because the STRING file data is buffered, the size of a STRING file is limited to 125 bytes*.

Before writing to a STRING file, memory must be allocated for the STRING file in the sign. (For further information, see "Set Memory Configuration" in Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21.)

STRING files are called from a TEXT file using the TEXT file Control character designated for a "Call STRING file". (For further information, see "Control characters" in "Appendix G: Alpha protocol ASCII table" on page 80).

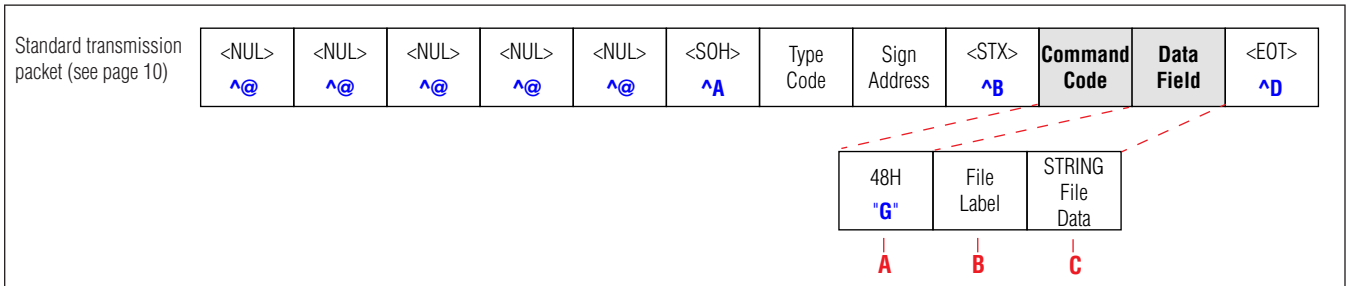
When reading from a STRING file, once the transmission packet has been sent, a sign will either pause or blank, depending on the sign type. Once a sign has transmitted the file, the sign will continue displaying the message from where it was interrupted.

6.3.1 Write STRING file Command Code — "G" (47H)

SHOW ME

An example of the Write STRING file packet is on page 73.

Table 18: Write STRING file transmission packet format



Item	Name		Description
A	Command Code		"G" (47H) = Write STRING file
B	Data Field	File Label	One ASCII character that indicates the STRING file being accessed. See "Appendix A: Valid File Labels" on page 49.
C	Data Field	STRING File Data	This data can be ASCII characters 20H through 7FH and the following Control characters (for more information, see "Appendix G: Alpha protocol ASCII table" on page 80) : 09H = No Hold speed 0DH = New line 11H = Disable wide characters **(default)** 12H = Enable wide characters 13H = Call Time (time of day will be called up) 15H = Speed 1 (slowest) 16H = Speed 2 17H = Speed 3 18H = Speed 4 **(default)** 19H = Speed 5 (fastest) 1AH = Select character set 1CH = Select character color (Rainbow 1 and 2 colors do not work in STRING files) 1EH = Select character spacing

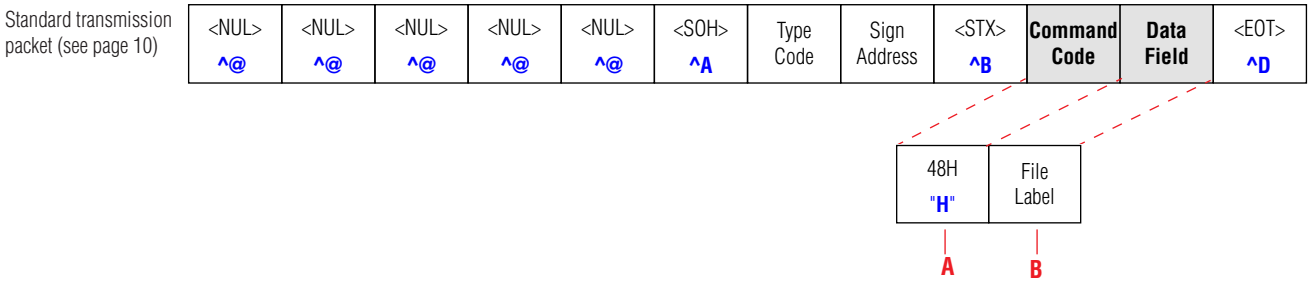
6.3.2 Read STRING file Command Code — "H" (48H)

SHOW ME

An example of the Read STRING file packet is on page 74.

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be read from.

Table 19: Read STRING file transmission packet format



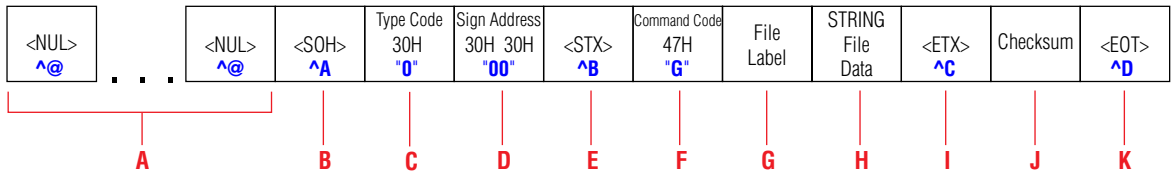
Item	Name		Description
A	Command Code		"H" (48H) = Read STRING file
B	Data Field	File Label	One ASCII character that indicates the STRING file being accessed. See "Appendix A: Valid File Labels" on page 49.

SHOW ME

An example of the Read STRING file sign response packet is on page 74.

Following the Read STRING file Command Code, a sign will respond with the following:

Table 20: Read STRING file sign response packet format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"G" (47H) is returned by the sign. (The Write STRING file Command Code.)
G	File Label	One ASCII character that indicates the STRING file being accessed. See "Appendix A: Valid File Labels" on page 49.
H	STRING File Data	See Table 18, "Write STRING file transmission packet format," on page 36.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

6.4 SMALL DOTS PICTURE file commands

SPECIAL NOTE

The size of a SMALL DOTS PICTURE file can be up to 31 x 255 pixels.

If a graphic needs to be larger than this, then use a LARGE DOTS PICTURE file (see "LARGE DOTS PICTURE file commands" on page 41).

SMALL DOTS PICTURE files are used to store dot patterns which are displayed by "calling" a picture file from a TEXT file. See "Call SMALL DOTS PICTURE" file in "Control codes (00 – 1FH)" on page 80.

The purpose of SMALL DOTS PICTURE files is to display small (up to 31 x 255 pixels) graphics, such as logos.

When a SMALL DOTS PICTURE file is written to a sign, the sign will go blank until the transmission is complete.

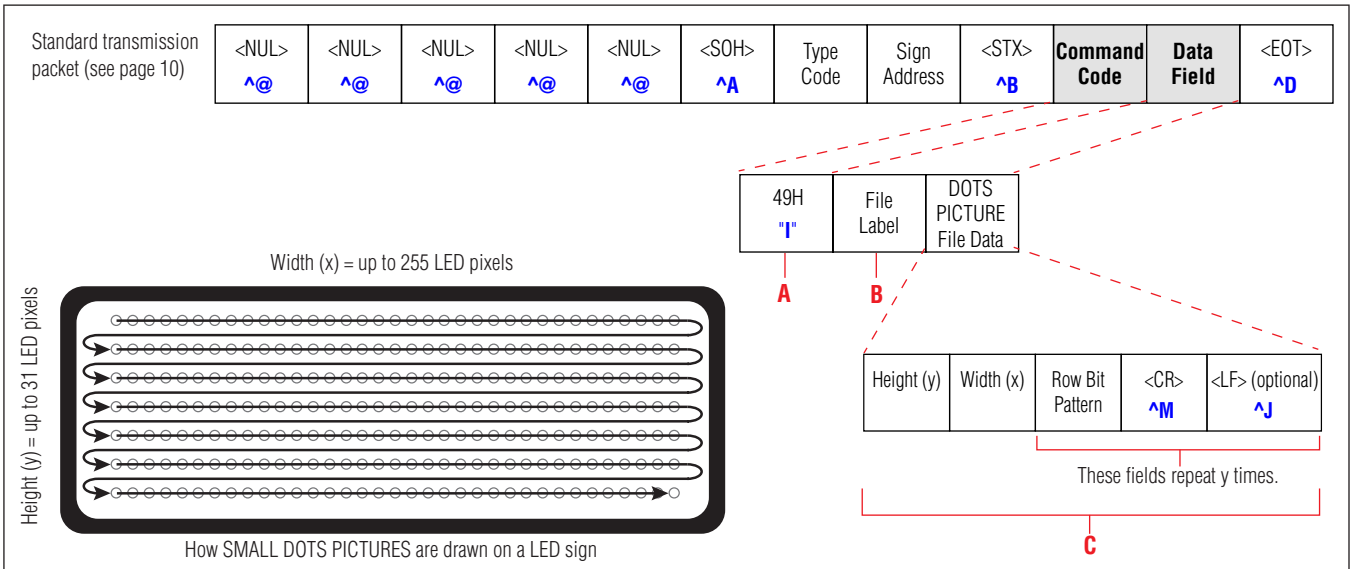
When reading from a SMALL DOTS PICTURE file, once the transmission packet has been sent, the sign will pause. Once a sign has completely transmitted the file, the sign will continue displaying the message from where it was interrupted.

6.4.1 Write SMALL DOTS PICTURE file Command Code — "I" (49H)

SHOW ME

An example of the Write SMALL DOTS PICTURE file packet is on page 75.

Table 21: Write SMALL DOTS PICTURE file transmission packet format



Item	Name			Description
A	Command Code			"I" (49H) = Write SMALL DOTS PICTURE file
B	Data Field	File Label		One ASCII character that indicates the DOTS PICTURE file accessed. See "Appendix A: Valid File Labels" on page 49.
C	Data Field	DOTS PICTURE File Data	Height (y)	Two ASCII hexadecimal bytes that represent the number of pixel rows in a SMALL DOTS PICTURE (a value from 0 to 31). This must match the pixel row bytes set up in "Set LARGE DOTS PICTURE Memory Configuration" in Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21. NOTE: In a LARGE DOTS PICTURE, four ASCII hexadecimal bytes are used to represent the number of pixel rows (a value from 0 to 65535).

Table 21: Write SMALL DOTS PICTURE file transmission packet format

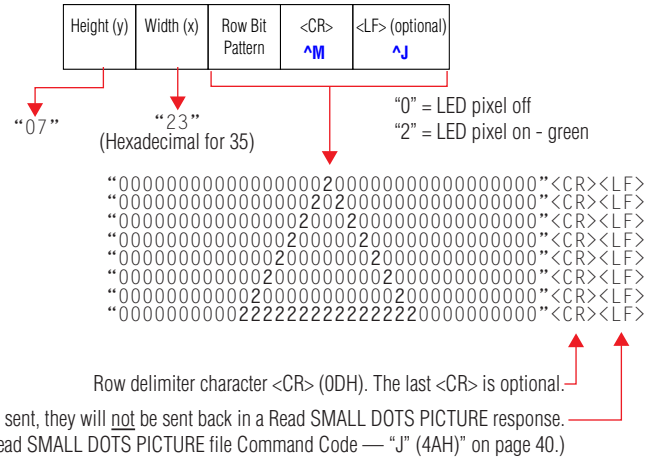
<table>
<tr><td rowspan="2">C
(cont)</td><td rowspan="2">Data
Field
(cont)</td><td rowspan="2">DOTS
PICTURE
File Data
(cont)</td><td>Width
(x)</td><td>Two ASCII hexadecimal bytes that represent the number of pixel columns in a SMALL DOTS PICTURE (a value from 0 to 255). This must match the pixel column bytes set up in “Set LARGE DOTS PICTURE Memory Configuration” in Table 15, “Write SPECIAL FUNCTION Command Code format — “E” (45H),” on page 21.

NOTE: In a LARGE DOTS PICTURE, four ASCII hexadecimal bytes are used to represent the number of pixel columns (a value from 0 to 65535).

NOTE: When sending a Write SMALL DOTS PICTURE file, the sign receiving the file will clear the current SMALL DOTS PICTURE file in memory immediately following the Width information.

NOTE: Following the Width bytes, there should be at least a 100 millisecond delay (not to exceed the timeout period) before sending the Row Bit Pattern.</td></tr>
<tr><td>[1]Row Bit
Pattern</td><td>The Width (x) number of ASCII characters which represent all the pixels in a row. The first ASCII character = the leftmost pixel in the row, the 2nd ASCII character = the next pixel in the row, etc. (see example below). Valid values are:
“0” 30H = pixel off
“1” 31H = pixel on - red
“2” 32H = pixel on - green
“3” 33H = pixel on - amber
“4” 34H = pixel on - dim red
“5” 35H = pixel on - dim green
“6” 36H = pixel on - brown
“7” 37H = pixel on - orange
“8” 38H = pixel on - yellow

NOTE: Some signs do not support the full range of colors.

To draw a green SMALL DOTS PICTURE like this (on the 7 x 35 pixel sign shown below) . . .
 
. . . the DOTS PICTURE File Data would look like this:
 

NOTE: If the number of row pixel characters is greater than the Width (x), then the extra row pixel characters will be ignored.
If the number of row pixel characters is less than the Width (x), then the remaining row pixel characters will be turned off (“0”).</td></tr>
</table>

NOTE: [1] DATA COMPRESSION — Row Bit Pattern can be data compressed as follows for LARGE DOTS PICTURE files. Data compression can be done anywhere within the Row Bit Pattern. The format for data compression is: <CTR-Q>XXB where:

<CTR-Q> = 11H

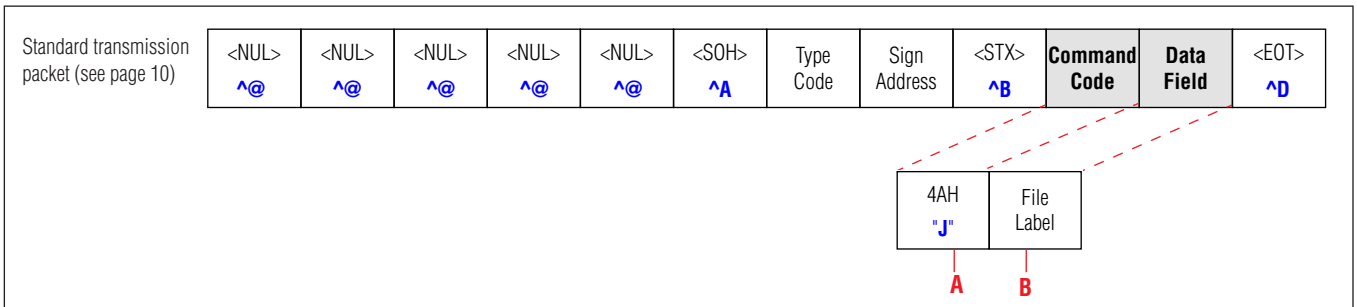
XX = Two ASCII hexadecimal characters from “00” to “FF” that stand for the number of times + 1 to repeat B (the pixel color). For example, a value of “0A” (10) means repeat 10 + 1 = 11 times.

B = Pixel color. Valid values are shown in Row Bit Pattern field above.

6.4.2 Read SMALL DOTS PICTURE file Command Code — "J" (4AH)

NOTE: Whenever doing a "read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be read from.

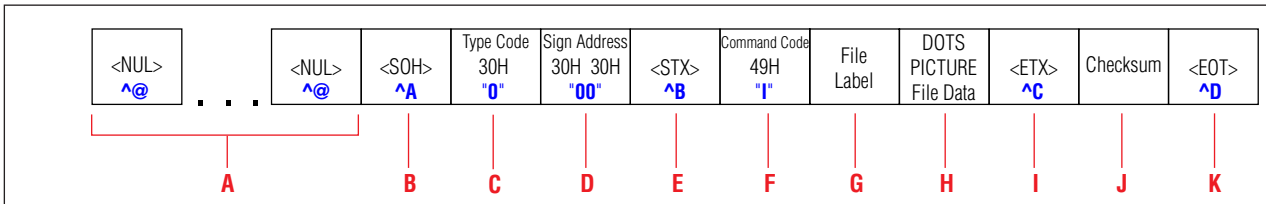
Table 22: Read SMALL DOTS PICTURE file transmission packet format



Item	Name		Description
A	Command Code		"J" (4AH) = Read SMALL DOTS PICTURE file
B	Data Field	File Label	One ASCII character that indicates the SMALL DOTS PICTURE file being accessed. See "Appendix A: Valid File Labels" on page 49.

Following the Read SMALL DOTS PICTURE file Command Code, a sign will respond with the following:

Table 23: Read SMALL DOTS PICTURE file sign response packet format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"I" (49H) is returned by the sign. (The Write SMALL DOTS PICTURE file Command Code.)
G	File Label	One ASCII character that indicates the SMALL DOTS PICTURE file being accessed. See "Appendix A: Valid File Labels" on page 49.
H	DOTS PICTURE File Data	See Table 21, "Write SMALL DOTS PICTURE file transmission packet format," on page 38.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

6.5 LARGE DOTS PICTURE file commands

SPECIAL NOTE

The size of an LARGE DOTS PICTURE file can be up to 65535 x 65535 pixels.

Only Alpha 7000, full matrix AlphaVision, AlphaPremiere, and AlphaEclipse signs support LARGE DOTS PICTURE files.

LARGE DOTS PICTURE files are used to store dot patterns which are displayed by "calling" a picture file from a TEXT file. See "Call LARGE DOTS PICTURE" file in "Control codes (00 – 1FH)" on page 80.

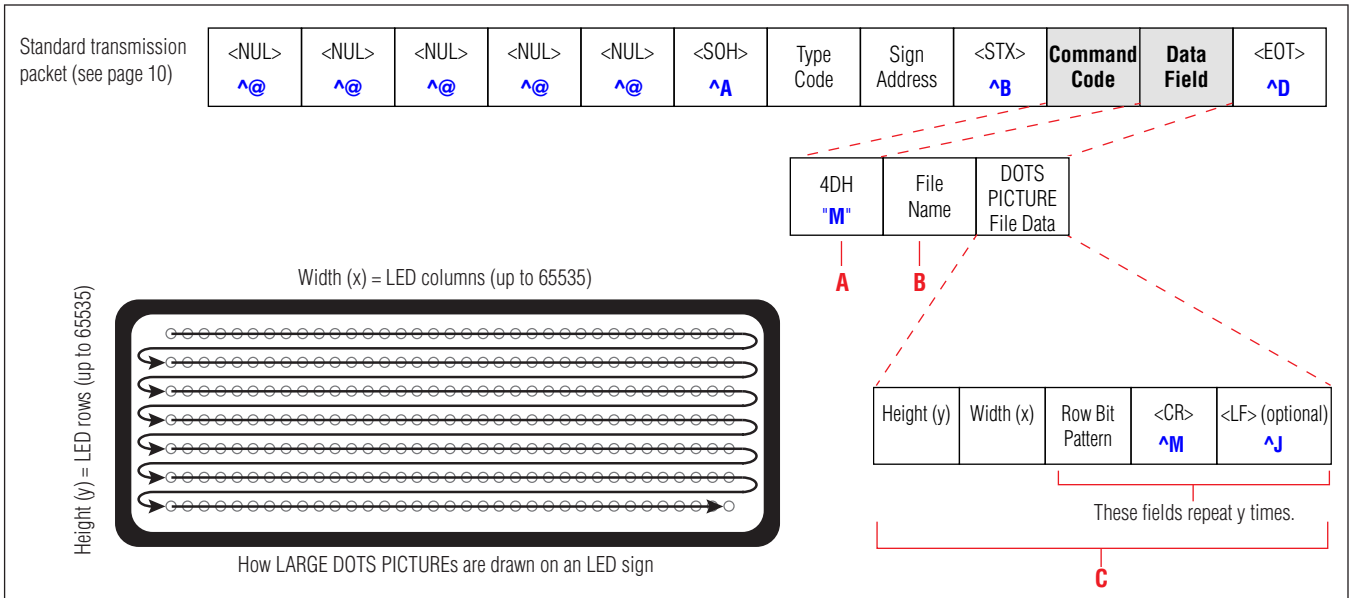
The main purpose of LARGE DOTS PICTURE files is to display large (up to 65535 x 65535 pixels) graphics.

When a LARGE DOTS PICTURE file is written to a sign, the sign will go blank until the transmission is complete.

When reading from a LARGE DOTS PICTURE file, once the transmission packet has been sent, a sign will either pause or blank, depending on the type of sign. Once a sign has completely transmitted the file, the sign will continue displaying the message from where it was interrupted.

6.5.1 Write LARGE DOTS PICTURE file Command Code — "M" (4DH)

Table 24: Write LARGE DOTS PICTURE file transmission packet format

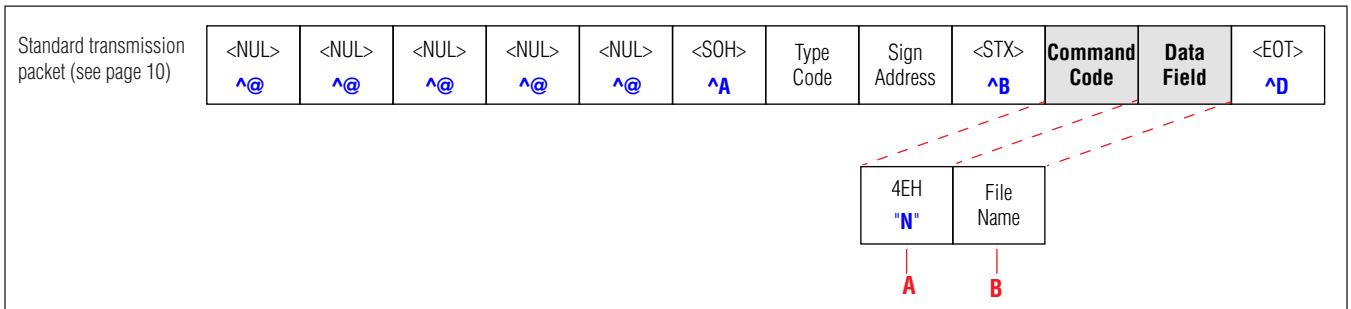


Item	Name			Description
A	Command Code			"M" (4DH) = Write LARGE DOTS PICTURE file
B	Data Field	File Name		Nine ASCII characters that indicate the LARGE DOTS PICTURE file being accessed.
C	Data Field	DOTS PICTURE File Data	Height (y)	See "DOTS PICTURE File Data" in Table 21, "Write SMALL DOTS PICTURE file transmission packet format," on page 38.
			Width (x)	
			Row Bit Pattern	
			<CR>	
			<LF>	

6.5.2 Read LARGE DOTS PICTURE file Command Code — "N" (4EH)

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be read from.

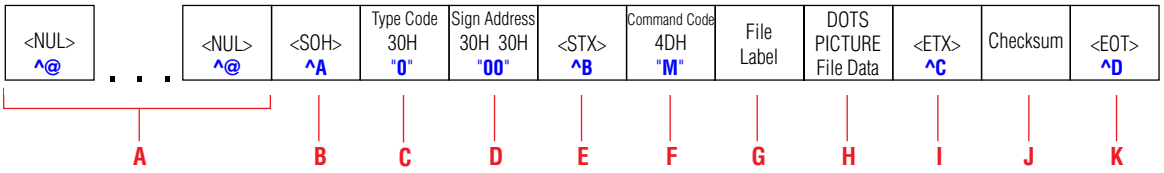
Table 25: Read LARGE DOTS PICTURE file transmission packet format



Item	Name		Description
A	Command Code		"N" (4EH) = Read LARGE DOTS PICTURE file
B	Data Field	File Name	Nine ASCII characters that indicate the LARGE DOTS PICTURE file being accessed.

Following the Read LARGE DOTS PICTURE file Command Code, a sign will respond with the following:

Table 26: Read LARGE DOTS PICTURE file sign response packet format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"M" (4DH) is returned by the sign. (This is the Write LARGE DOTS PICTURE file Command Code.)
G	File Name	One ASCII character that indicates the DOTS PICTURE file being accessed. See "Appendix A: Valid File Labels" on page 49.
H	DOTS PICTURE File Data	See Table 21, "Write SMALL DOTS PICTURE file transmission packet format," on page 38.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

6.6 RGB DOTS PICTURE file commands

SPECIAL NOTE

The size of an RGB DOTS PICTURE file can be up to 65535 x 65535 pixels.

Only AlphaEclipse 3600 signs support RGB DOTS PICTURE files.

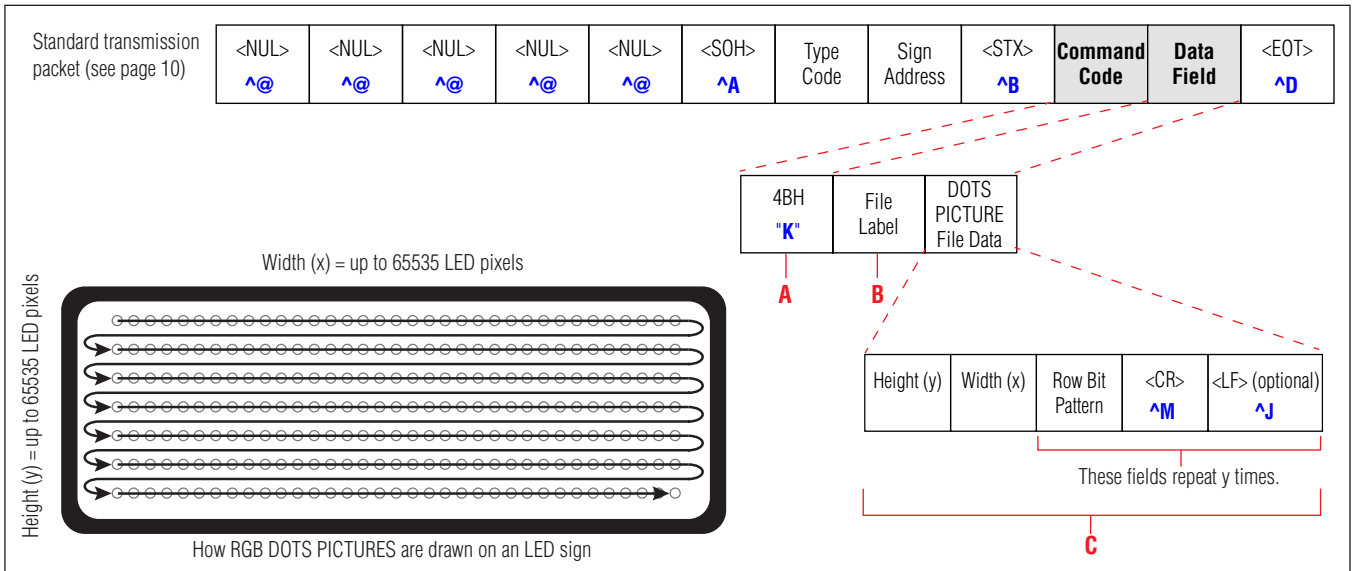
RGB DOTS PICTURE files are used to store RGB color dot patterns which are displayed by "calling" a picture file from a TEXT file. See "Call LARGE DOTS PICTURE" file in "Control codes (00 – 1FH)" on page 80.

The main purpose of RGB DOTS PICTURE files is to display RGB (Red-Green-Blue) graphics which could potentially have over 16 million colors.

When reading an RGB DOTS PICTURE file, the information on a sign will pause until the entire file has been received. Once a sign has completely transmitted the file, the sign will continue displaying the message from where it was interrupted.

6.6.1 Write RGB DOTS PICTURE file Command Code — "K" (4BH)

Table 27: Write RGB DOTS PICTURE file transmission packet format

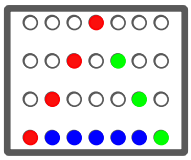
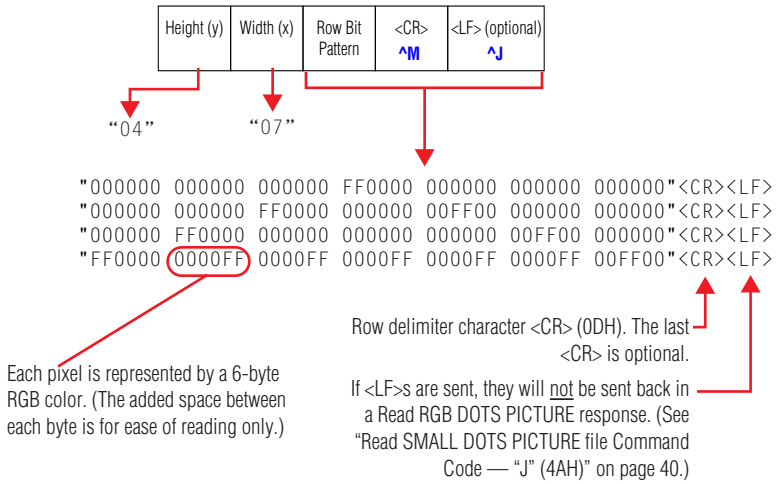


Item	Name			Description
A	Command Code			"K" (4BH) = Write RGB DOTS PICTURE file
B	Data Field	File Label		One ASCII character that indicates the RGB DOTS PICTURE file accessed. See "Appendix A: Valid File Labels" on page 49.
C	Data Field	DOTS PICTURE File Data	Height (y)	Four ASCII hexadecimal bytes that represent the number (0 to 65535) of pixel rows in a RGB DOTS PICTURE. This must match the pixel row bytes set up using "Set LARGE DOTS PICTURE Memory Configuration" (see page 26).
			Width (x)	Four ASCII hexadecimal bytes that represent the number (0 to 65535) of pixel columns in a RGB DOTS PICTURE. This must match the pixel column bytes set up using "Set LARGE DOTS PICTURE Memory Configuration" (see page 26). NOTE: Following the Width bytes, there should be at least a 100 millisecond delay (not to exceed the timeout period) before sending the Row Bit Pattern.

Table 27: Write RGB DOTS PICTURE file transmission packet format

<table>
<tr>
<td>C
(cont)</td>
<td>Data Field (cont)</td>
<td>DOTS PICTURE File Data (cont)</td>
<td>[1]Row Bit Pattern</td>
<td>
The Width (x) number of ASCII characters which represent all the pixels in a row. The first ASCII character = the leftmost pixel in the row, the 2nd ASCII character = the next pixel in the row, etc. (see example below).

Each RGB pixel is represented by six, ASCII hexadecimal characters in the format: RRGGBB where
• RR = a Red color value from “00” to “FF”
• GG = a Green color value from “00” to “FF”
• BB = a Blue color value from “00” to “FF”

To draw a small (4 pixels high x 7 pixels wide) RGB DOTS PICTURE like this . . .

. . . the RGB DOTS PICTURE File Data would look like this:


NOTE: If the number of row pixel characters is greater than the Width (x), then the extra row pixel characters will be ignored.
If the number of row pixel characters is less than the Width (x), then the remaining row pixel characters will be turned off (“0”).

NOTE: Since each LED pixel on a sign must be represented by a 6-byte RGB code, a large graphic could take a significant amount of time before it is displayed on a sign. For example, a 32 x 64 sign has 2048 pixels. An RGB graphic that size would equal 12,288 bytes (2048 x 6). If this RGB graphic was transmitted to a sign at a baud rate of 38,400 (or 4800 bytes/sec), then the sign would need about 2.5 seconds (12,288 / 4800) to display the graphic.
</td>
</tr>
</table>

NOTE: [1] DATA COMPRESSION — Row Bit Pattern can be data compressed as follows for RGB DOTS PICTURE files. Data compression can be done anywhere within the Row Bit Pattern. The format for data compression is: <CTR-Q>XXRRGGBB where:

<CTR-Q> = 11H

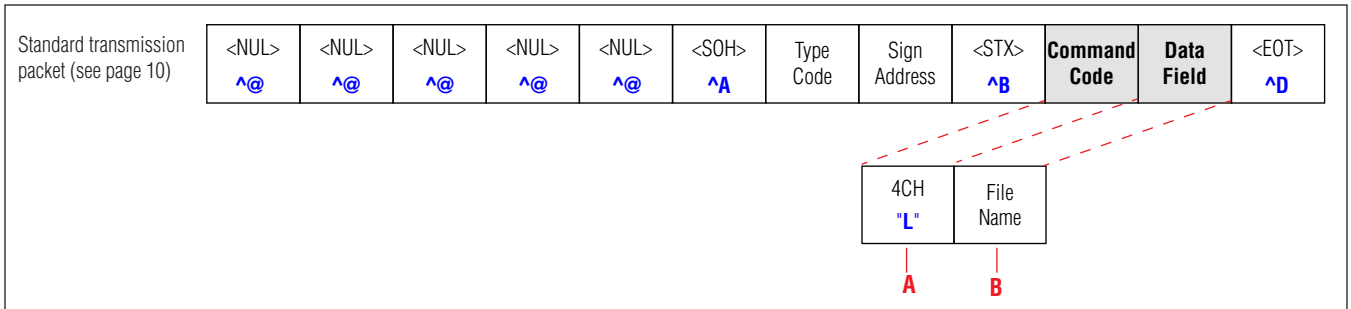
XX = Two ASCII hexadecimal characters from “00” to “FF” that stand for the number of times + 1 to repeat RRGGBB (the RGB pixel color). For example, a value of “0A” (10) means repeat 10 + 1 = 11 times.

RRGGBB = RGB pixel color. Valid values are shown in Row Bit Pattern field above.

6.6.2 Read RGB DOTS PICTURE file Command Code — "L" (4CH)

NOTE: Whenever doing a "Read" command on a network with multiple signs, it is important that each sign has a unique Serial Address. Also, only one sign at a time should be read from.

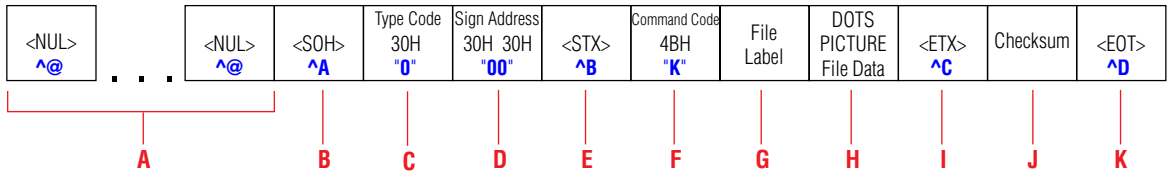
Table 28: Read RGB DOTS PICTURE file transmission packet format



Item	Name		Description
A	Command Code		"L" (4CH) = Read RGB DOTS PICTURE file
B	Data Field	File Name	Nine ASCII characters that indicate the RGB DOTS PICTURE file being read.

Following the Read LARGE DOTS PICTURE file Command Code, a sign will respond with the following:

Table 29: Read RGB DOTS PICTURE file sign response packet format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"K" (4BH) is returned by the sign. (This is the Write RGB DOTS PICTURE file Command Code.)
G	File Name	One ASCII character that indicates the DOTS PICTURE file being accessed. See "Appendix A: Valid File Labels" on page 49.
H	DOTS PICTURE File Data	See Table 27, "Write RGB DOTS PICTURE file transmission packet format," on page 43.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

6.6.3 RGB color chart

This chart of 216 RGB colors will render color accurately on almost any computer monitor that can display at least 256 colors.

In the chart, each color is defined by a hexadecimal and a decimal number. For example, the color in the uppermost left corner has a RGB hexadecimal value of “990033” and decimal values of “153”, “000”, and “051”:

- Red value = 99H, 153D
- Green value = 00H, 0D
- Blue value = 33H, 51D

NOTE: This chart represents a small percentage of the possible 16,777,216 (256 x 256 x 256) RBG color combinations.

990033 R: 153 G: 000 B: 051	FF3366 R: 255 G: 051 B: 102	CC0033 R: 204 G: 000 B: 051	FF0033 R: 255 G: 000 B: 051	FF9999 R: 255 G: 153 B: 153	CC3366 R: 204 G: 051 B: 102	FFCCFF R: 255 G: 204 B: 255	CC6699 R: 204 G: 051 B: 153	993366 R: 153 G: 051 B: 102	660033 R: 102 G: 000 B: 051	CC3399 R: 204 G: 051 B: 153	FF99CC R: 255 G: 153 B: 204	FF66CC R: 255 G: 102 B: 204	FF99FF R: 255 G: 153 B: 255	FF6699 R: 255 G: 102 B: 153	CC0066 R: 204 G: 000 B: 102
FF0066 R: 255 G: 000 B: 102	FF3399 R: 255 G: 051 B: 153	FF0099 R: 255 G: 000 B: 153	FF33CC R: 255 G: 051 B: 204	FF00CC R: 255 G: 000 B: 204	FF66FF R: 255 G: 102 B: 255	FF33FF R: 255 G: 051 B: 255	FF00FF R: 255 G: 000 B: 255	CC0099 R: 204 G: 000 B: 153	990066 R: 153 G: 000 B: 102	CC66CC R: 204 G: 102 B: 204	CC33CC R: 204 G: 051 B: 204	CC99FF R: 204 G: 153 B: 255	CC66FF R: 204 G: 102 B: 255	CC33FF R: 204 G: 051 B: 255	993399 R: 153 G: 051 B: 153
CC00CC R: 204 G: 000 B: 204	CC00FF R: 204 G: 000 B: 255	9900CC R: 153 G: 000 B: 204	990099 R: 153 G: 000 B: 153	CC99CC R: 204 G: 153 B: 204	996699 R: 153 G: 102 B: 153	663366 R: 102 G: 051 B: 102	660099 R: 102 G: 000 B: 153	9933CC R: 153 G: 051 B: 204	660066 R: 102 G: 000 B: 102	9900FF R: 153 G: 000 B: 255	9933FF R: 153 G: 051 B: 255	9966CC R: 153 G: 102 B: 204	330033 R: 051 G: 000 B: 051	663399 R: 102 G: 051 B: 153	6633CC R: 102 G: 051 B: 204
6600CC R: 102 G: 000 B: 204	9966FF R: 153 G: 102 B: 255	330066 R: 051 G: 000 B: 102	6600FF R: 102 G: 000 B: 255	6633FF R: 102 G: 051 B: 255	CCCCFF R: 204 G: 204 B: 255	9999FF R: 153 G: 153 B: 255	9999CC R: 153 G: 153 B: 204	6666CC R: 102 G: 102 B: 204	6666FF R: 102 G: 102 B: 255	666699 R: 102 G: 102 B: 153	333366 R: 051 G: 051 B: 102	333399 R: 051 G: 051 B: 153	330099 R: 051 G: 000 B: 153	3300CC R: 051 G: 000 B: 204	3300FF R: 051 G: 000 B: 255
3333FF R: 051 G: 051 B: 255	3333CC R: 051 G: 051 B: 204	0066FF R: 000 G: 102 B: 255	0033FF R: 000 G: 051 B: 255	3366FF R: 051 G: 102 B: 255	3366CC R: 051 G: 102 B: 204	000066 R: 000 G: 000 B: 102	000033 R: 000 G: 000 B: 051	0000FF R: 000 G: 000 B: 255	000099 R: 000 G: 000 B: 153	0033CC R: 000 G: 051 B: 204	0000CC R: 000 G: 000 B: 204	336699 R: 051 G: 102 B: 153	0066CC R: 000 G: 102 B: 204	99CCFF R: 153 G: 204 B: 255	6699FF R: 102 G: 153 B: 255
003366 R: 000 G: 051 B: 102	6699CC R: 102 G: 153 B: 204	006699 R: 000 G: 102 B: 153	3399CC R: 051 G: 153 B: 204	0099CC R: 000 G: 153 B: 204	66CCFF R: 102 G: 204 B: 255	3399FF R: 051 G: 153 B: 255	003399 R: 000 G: 051 B: 153	0099FF R: 000 G: 153 B: 255	33CCFF R: 051 G: 204 B: 255	00CCFF R: 000 G: 204 B: 255	99FFFF R: 153 G: 255 B: 255	66FFFF R: 102 G: 255 B: 255	33FFFF R: 051 G: 255 B: 255	00FFFF R: 000 G: 255 B: 255	00CCCC R: 000 G: 204 B: 204
009999 R: 000 G: 153 B: 153	669999 R: 102 G: 153 B: 153	99CCCC R: 153 G: 204 B: 204	CCFFFF R: 204 G: 255 B: 255	33CCCC R: 051 G: 204 B: 204	66CCCC R: 102 G: 204 B: 204	339999 R: 051 G: 153 B: 153	336666 R: 051 G: 102 B: 102	006666 R: 000 G: 102 B: 102	003333 R: 000 G: 051 B: 051	00FFCC R: 000 G: 255 B: 204	33FFCC R: 051 G: 255 B: 204	33CC99 R: 051 G: 204 B: 153	00CC99 R: 000 G: 204 B: 153	66FFCC R: 102 G: 255 B: 204	99FFCC R: 153 G: 255 B: 204
00FF99 R: 000 G: 255 B: 153	339966 R: 051 G: 153 B: 102	006633 R: 000 G: 102 B: 051	336633 R: 051 G: 102 B: 051	669966 R: 102 G: 153 B: 102	66CC66 R: 102 G: 204 B: 102	99FF99 R: 153 G: 255 B: 153	66FF66 R: 102 G: 255 B: 102	339933 R: 051 G: 153 B: 051	99CC99 R: 153 G: 204 B: 153	66FF99 R: 102 G: 255 B: 153	33FF99 R: 051 G: 255 B: 153	33CC66 R: 051 G: 204 B: 102	00CC66 R: 000 G: 204 B: 102	66CC99 R: 102 G: 204 B: 153	009966 R: 000 G: 153 B: 102
009933 R: 000 G: 153 B: 051	33FF66 R: 051 G: 255 B: 102	00FF66 R: 000 G: 255 B: 102	CCFFCC R: 204 G: 255 B: 204	CCFF99 R: 204 G: 255 B: 153	99FF66 R: 153 G: 255 B: 102	99FF33 R: 153 G: 255 B: 051	00FF33 R: 000 G: 255 B: 051	33FF33 R: 051 G: 255 B: 051	00CC33 R: 000 G: 204 B: 051	33CC33 R: 051 G: 204 B: 051	66FF33 R: 102 G: 255 B: 051	00FF00 R: 000 G: 255 B: 000	66CC33 R: 102 G: 204 B: 051	006600 R: 000 G: 102 B: 000	003300 R: 000 G: 051 B: 000
009900 R: 000 G: 153 B: 000	33FF00 R: 051 G: 255 B: 000	66FF00 R: 102 G: 255 B: 000	99FF00 R: 153 G: 255 B: 000	66CC00 R: 102 G: 204 B: 000	00CC00 R: 000 G: 204 B: 000	33CC00 R: 051 G: 204 B: 000	339900 R: 051 G: 153 B: 000	99CC66 R: 153 G: 204 B: 102	669933 R: 102 G: 153 B: 051	99CC33 R: 153 G: 204 B: 051	336600 R: 051 G: 102 B: 000	669900 R: 102 G: 153 B: 000	99CC00 R: 153 G: 204 B: 000	CCFF66 R: 204 G: 255 B: 102	CCFF33 R: 204 G: 255 B: 051
CCFF00 R: 204 G: 255 B: 000	999900 R: 153 G: 153 B: 000	CCCC00 R: 204 G: 204 B: 000	CCCC33 R: 204 G: 204 B: 051	333300 R: 051 G: 051 B: 000	666600 R: 102 G: 102 B: 000	999933 R: 153 G: 153 B: 051	CCCC66 R: 204 G: 204 B: 102	666633 R: 102 G: 102 B: 051	999966 R: 153 G: 153 B: 102	CCCC99 R: 204 G: 204 B: 153	FFFFCC R: 255 G: 255 B: 204	FFFF99 R: 255 G: 255 B: 153	FFFF66 R: 255 G: 255 B: 102	FFFF33 R: 255 G: 255 B: 051	FFFF00 R: 255 G: 255 B: 000
FFCC00 R: 255 G: 204 B: 000	FFCC66 R: 255 G: 204 B: 102	FFCC33 R: 255 G: 204 B: 051	CC9933 R: 204 G: 153 B: 051	996600 R: 153 G: 102 B: 000	CC9900 R: 204 G: 153 B: 000	FF9900 R: 255 G: 153 B: 000	CC6600 R: 204 G: 102 B: 000	993300 R: 153 G: 051 B: 000	CC6633 R: 204 G: 102 B: 051	663300 R: 102 G: 051 B: 000	FF9966 R: 255 G: 153 B: 102	FF6633 R: 255 G: 102 B: 051	FF9933 R: 255 G: 153 B: 051	FF6600 R: 255 G: 102 B: 000	CC3300 R: 204 G: 051 B: 000
996633 R: 153 G: 102 B: 051	330000 R: 051 G: 000 B: 000	663333 R: 102 G: 051 B: 051	996666 R: 153 G: 102 B: 102	CC9999 R: 204 G: 153 B: 153	993333 R: 153 G: 051 B: 051	CC6666 R: 204 G: 102 B: 102	FFCCCC R: 255 G: 204 B: 204	FF3333 R: 255 G: 051 B: 051	CC3333 R: 204 G: 051 B: 051	FF6666 R: 255 G: 102 B: 102	660000 R: 102 G: 000 B: 000	990000 R: 153 G: 000 B: 000	CC0000 R: 204 G: 000 B: 000	FF0000 R: 255 G: 000 B: 000	FF3300 R: 255 G: 051 B: 000
CC9966 R: 204 G: 153 B: 102	FFCC99 R: 255 G: 204 B: 153	FFFFFF R: 255 G: 255 B: 255	CCCCCC R: 204 G: 204 B: 204	999999 R: 153 G: 153 B: 153	666666 R: 102 G: 102 B: 102	333333 R: 051 G: 051 B: 051	000000 R: 000 G: 000 B: 000								

6.7 ALPHAVISION BULLETIN MESSAGE file commands

An ALPHAVISION BULLETIN MESSAGE allows a text message of up to 225 characters to be rotated on a sign's display without interrupting the current operation.

6.7.1 Write ALPHAVISION BULLETIN MESSAGE file Command Code — "O" (4FH)

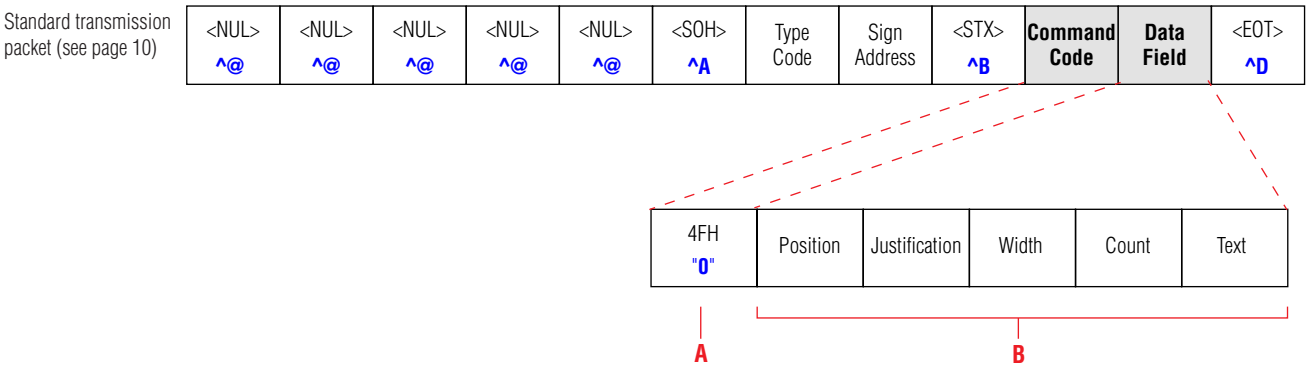
Only AlphaVision and Series 7000 signs support this command.

NOTE: Only the size of the ALPHAVISION BULLETIN MESSAGE window is cleared, not the entire line.

NOTE: Only seven high characters are supported.

NOTE: Only AlphaVision signs support the ability to vary window Position and Justification. An Alpha Series 7000 sign displays an ALPHAVISION BULLETIN MESSAGE across the entire width of the sign.

Table 30: Write ALPHAVISION BULLETIN MESSAGE file transmission packet format

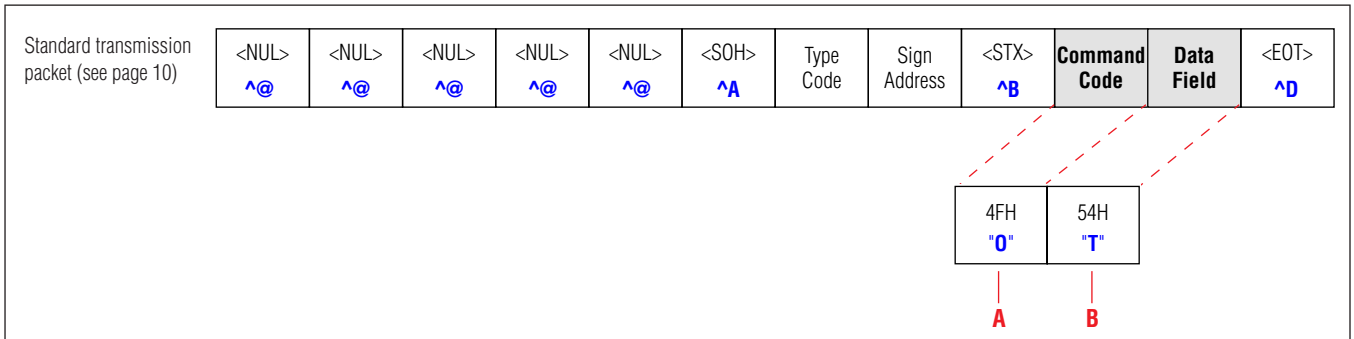


Item	Name		Description
A	Command Code		"O" (4FH) = Write ALPHAVISION BULLETIN MESSAGE file
B	Data Field	Position	One ASCII character that stands for the position of the bulletin message on a sign. Valid values are: "T" 54H = Top of the display "B" 42H = Bottom of the display
		Justification	One ASCII character that stands for the alignment of the bulletin message on a sign. Valid values are: "L" 4CH = Left side of the display "C" 43H = Center of the display "R" 52H = Right side of the display
		Width	Two ASCII hexadecimal digits that specify the total number of characters in the Text field below. This number will be rounded up to the nearest 32-column width. For example, if the total number of characters = 78, this number would be rounded up to 32 x 3 = 96. The maximum Width is 255 ("FF").
		Count	Two ASCII hexadecimal digits that stand for the number of times the bulletin message should be displayed.
		Text	Up to 225 ASCII characters that comprise the actual bulletin message itself. Messages longer than 225 characters will be truncated. NOTE: The only ASCII Control characters allowed in a bulletin message are color codes. (For more information, see "Appendix G: Alpha protocol ASCII table" on page 80.)

6.7.2 Stop ALPHAVISION BULLETIN MESSAGE file Command Code — "OT" (4F + 54H)

To stop an ALPHAVISION BULLETIN MESSAGE before the Count field (above) has been reached, use this Command Code:

Table 31: Terminate ALPHAVISION BULLETIN MESSAGE file transmission packet format



Item	Name	Description
A	Command Code	"O" (4FH)
B	Data Field	"T" (54H) is the termination character.

7.0 Appendixes

7.1 Appendix A: Valid File Labels

A File Label is a single ASCII character. Messages are stored in or retrieved from the memory file that is defined by this label in the Memory Configuration.

File Labels can be anywhere in the range 20H through 7EH inclusive.

The only special case occurs when File Label "0" (30H) is used for a Priority TEXT file (see "Priority TEXT files" on page 20) which is pre-configured as a set portion of memory outside of the Memory Pool.

Table 32: Valid File Labels

20H - sp	30H - "0"	40H - "@"	50H - "P"	60H - "`"	70H - "p"
21H - "!"	31H - "1"	41H - "A"	51H - "Q"	61H - "a"	71H - "q"
22H - """	32H - "2"	42H - "B"	52H - "R"	62H - "b"	72H - "r"
23H - "#"	33H - "3"	43H - "C"	53H - "S"	63H - "c"	73H - "s"
24H - "$"	34H - "4"	44H - "D"	54H - "T"	64H - "d"	74H - "t"
25H - "%"	35H - "5"	45H - "E"	55H - "U"	65H - "e"	75H - "u"
26H - "&"	36H - "6"	46H - "F"	56H - "V"	66H - "f"	76H - "v"
27H - "'"	37H - "7"	47H - "G"	57H - "W"	67H - "g"	77H - "w"
28H - "("	38H - "8"	48H - "H"	58H - "X"	68H - "h"	78H - "x"
29H - ")"	39H - "9"	49H - "I"	59H - "Y"	69H - "i"	79H - "y"
2AH - "*"	3AH - ":"	4AH - "J"	5AH - "Z"	6AH - "j"	7AH - "z"
2BH - "+"	3BH - ";"	4BH - "K"	5BH - "["	6BH - "k"	7BH - "{"
2CH - ","	3CH - "<"	4CH - "L"	5CH - "\"	6CH - "l"	7CH - "|"
2DH - "-"	3DH - "="	4DH - "M"	5DH - "]"	6DH - "m"	7DH - "}"
2EH - "."	3EH - ">"	4EH - "N"	5EH - "¢"	6EH - "n"	7EH - 1/2 sp
2FH - "/"	3FH - "?"	4FH - "O"	5FH - "_"	6FH - "o"	7FH - reserved

NOTE: File Label "0" (30H) is used for a Priority TEXT file (see "Priority TEXT files" on page 20).

NOTE: File Label "0" (30H) and "?" (3FH) can not be used as STRING file labels.

NOTE: If the Counter feature ("Appendix C: Counter information" on page 51) of a sign is used, then File Labels "1" (31H) through "5" (35H) are reserved for Target files.

NOTE: sp = space
1/2 sp = 1/2 space

7.2 Appendix B: Valid Start and Stop times

The Start and Stop times are represented in ASCII. For example, a 8:50 am time = 35H = "35" (the ASCII characters 33H and 35H). Stop Time is ignored when Start Time is set to *Always* (FF):

Table 33: Valid TEXT file Start and Stop times

12:00 a.m. - 00H	8:00 a.m. - 30H	4:00 p.m. - 60H
12:10 a.m. - 01H	8:10 a.m. - 31H	4:10 p.m. - 61H
12:20 a.m. - 02H	8:20 a.m. - 32H	4:20 p.m. - 62H
12:30 a.m. - 03H	8:30 a.m. - 33H	4:30 p.m. - 63H
12:40 a.m. - 04H	8:40 a.m. - 34H	4:40 p.m. - 64H
12:50 a.m. - 05H	8:50 a.m. - 35H	4:50 p.m. - 65H
1:00 a.m. - 06H	9:00 a.m. - 36H	5:00 p.m. - 66H
1:10 a.m. - 07H	9:10 a.m. - 37H	5:10 p.m. - 67H
1:20 a.m. - 08H	9:20 a.m. - 38H	5:20 p.m. - 68H
1:30 a.m. - 09H	9:30 a.m. - 39H	5:30 p.m. - 69H
1:40 a.m. - 0AH	9:40 a.m. - 3AH	5:40 p.m. - 6AH
1:50 a.m. - 0BH	9:50 a.m. - 3BH	5:50 p.m. - 6BH
2:00 a.m. - 0CH	10:00 a.m. - 3CH	6:00 p.m. - 6CH
2:10 a.m. - 0DH	10:10 a.m. - 3DH	6:10 p.m. - 6DH
2:20 a.m. - 0EH	10:20 a.m. - 3EH	6:20 p.m. - 6EH
2:30 a.m. - 0FH	10:30 a.m. - 3FH	6:30 p.m. - 6FH
2:40 a.m. - 10H	10:40 a.m. - 40H	6:40 p.m. - 70H
2:50 a.m. - 11H	10:50 a.m. - 41H	6:50 p.m. - 71H
3:00 a.m. - 12H	11:00 a.m. - 42H	7:00 p.m. - 72H
3:10 a.m. - 13H	11:10 a.m. - 43H	7:10 p.m. - 73H
3:20 a.m. - 14H	11:20 a.m. - 44H	7:20 p.m. - 74H
3:30 a.m. - 15H	11:30 a.m. - 45H	7:30 p.m. - 75H
3:40 a.m. - 16H	11:40 a.m. - 46H	7:40 p.m. - 76H
3:50 a.m. - 17H	11:50 a.m. - 47H	7:50 p.m. - 77H
4:00 a.m. - 18H	12:00 p.m. - 48H	8:00 p.m. - 78H
4:10 a.m. - 19H	12:10 p.m. - 49H	8:10 p.m. - 79H
4:20 a.m. - 1AH	12:20 p.m. - 4AH	8:20 p.m. - 7AH
4:30 a.m. - 1BH	12:30 p.m. - 4BH	8:30 p.m. - 7BH
4:40 a.m. - 1CH	12:40 p.m. - 4CH	8:40 p.m. - 7CH
4:50 a.m. - 1DH	12:50 p.m. - 4DH	8:50 p.m. - 7DH
5:00 a.m. - 1EH	1:00 p.m. - 4EH	9:00 p.m. - 7EH
5:10 a.m. - 1FH	1:10 p.m. - 4FH	9:10 p.m. - 7FH
5:20 a.m. - 20H	1:20 p.m. - 50H	9:20 p.m. - 80H
5:30 a.m. - 21H	1:30 p.m. - 51H	9:30 p.m. - 81H
5:40 a.m. - 22H	1:40 p.m. - 52H	9:40 p.m. - 82H
5:50 a.m. - 23H	1:50 p.m. - 53H	9:50 p.m. - 83H
6:00 a.m. - 24H	2:00 p.m. - 54H	10:00 p.m. - 84H
6:10 a.m. - 25H	2:10 p.m. - 55H	10:10 p.m. - 85H
6:20 a.m. - 26H	2:20 p.m. - 56H	10:20 p.m. - 86H
6:30 a.m. - 27H	2:30 p.m. - 57H	10:30 p.m. - 87H
6:40 a.m. - 28H	2:40 p.m. - 58H	10:40 p.m. - 88H
6:50 a.m. - 29H	2:50 p.m. - 59H	10:50 p.m. - 89H
7:00 a.m. - 2AH	3:00 p.m. - 5AH	11:00 p.m. - 8AH
7:10 a.m. - 2BH	3:10 p.m. - 5BH	11:10 p.m. - 8BH
7:20 a.m. - 2CH	3:20 p.m. - 5CH	11:20 p.m. - 8CH
7:30 a.m. - 2DH	3:30 p.m. - 5DH	11:30 p.m. - 8DH
7:40 a.m. - 2EH	3:40 p.m. - 5EH	11:40 p.m. - 8EH
7:50 a.m. - 2FH	3:50 p.m. - 5FH	11:50 p.m. - 8FH
ALL DAY - FDH	NEVER - FEH	ALWAYS - FFH

7.3 Appendix C: Counter information

NOTE: In order to use counters, a sign must have a counter firmware upgrade.

7.3.1 Displaying Counter values

SHOW ME

An example of displaying a Counter value is on page 66.

TEXT files can use Control codes to display counter values. (See "Counters" in the "Extended character set" in "Appendix G: Alpha protocol ASCII table" on page 80).

7.3.2 Setting up Counters

7.3.2.1 Memory Configuration

The default Memory Configuration on EZ95 signs and all EZII signs *equipped with the counter upgrade* (in addition to the default TEXT file "A" and DOTS PICTURE file "A") contains five TARGET TEXT files with labels "1" through "5". Each file is set up with a keyboard status of "unlocked" and is 100 bytes in length (64H). The default Run Start Time for each is "Never" (FEH). It is important to keep in mind that when writing a new Memory Configuration that TEXT files "1" through "5" need to be included, as these are the TARGET files. (See "Set Memory Configuration" in "Write SPECIAL FUNCTION Command Code — "E" (45H)" on page 21.)

7.3.2.2 Memory Dump

A Memory Dump response from a sign equipped with the counter upgrade also contains the counter information. (See "Memory Dump" in "Read SPECIAL FUNCTION Command Code — "F" (46H)" on page 28.)

7.3.2.3 Run Sequence

It is important to set up a Run Sequence which runs according to the file run times. Also, all five Target File Labels ("1" through "5") should always be included in the Run Sequence, along with other desired TEXT files. (See "Set Run Sequence" in "Write SPECIAL FUNCTION Command Code — "E" (45H)" on page 21.)

7.3.2.4 Run Day Table

It is important to set up a Run Day Table which accounts for, in addition to all user TEXT files, the Target files. The default Start Day value for all Target TEXT files is "0" (Daily), and the default Stop Day value is "2" (ignored). (See "Set Run Day Table" in "Write SPECIAL FUNCTION Command Code — "E" (45H)" on page 21.)

7.4 Appendix D: STRING file notes

A STRING file is a short stream of data that is "called" from a TEXT file. A typical use of a STRING file would be to update a count (e.g., a count-down timer) that is continuously displayed on a sign.

7.4.1 Advantages of using STRING files

- When STRING files are used to update data on a sign, the sign won't "blink" or flash during the update. (However, a sign will blink when TEXT files are updated.)
- Using STRING files saves sign memory. For example, if some important data is displayed multiple times within a TEXT file, this data only needs to be stored once in a STRING file, then "called" from the appropriate location within the TEXT file.

7.4.2 Using STRING files example

To use STRING files, there are three basic steps:

STEP 1 — Allocate memory in a sign for the STRING file (and the TEXT file that calls it).

STEP 2 — Write the TEXT file which calls the STRING file.

STEP 3 — Update the STRING file.

SPECIAL NOTE

STEP 1 and STEP 2 are used to initialize a STRING file.

STEP 3 is used to change the information in a STRING file once it has been initialized.

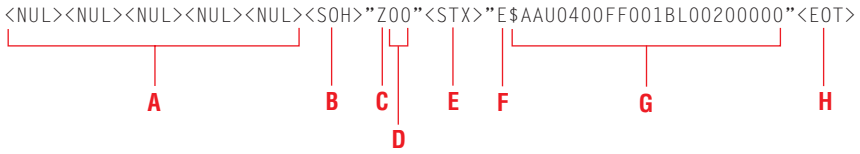
NOTE: The default character spacing is proportional, rather than fixed width. Because of this, a sign's auto-centering will move the displayed data around with the changing character widths in order to keep the data centered.
To avoid this distracting data movement on a sign:
(a) always send the same number of characters in the STRING file data, and
(b) always use fixed width characters by embedding the following 2-byte sequence in your TEXT file *before* the STRING file call: 1EH (Control "^") + 31H ("1").

7.4.2.1 STEP 1 — Allocate memory for a STRING file (and the TEXT file that calls it)

To allocate memory for one STRING file and the TEXT file which calls the STRING file, the following transmission packet could be sent to a network of signs:

Table 34: Using STRING files example: STEP 1

This following is a Standard Transmission packet (see page 10):



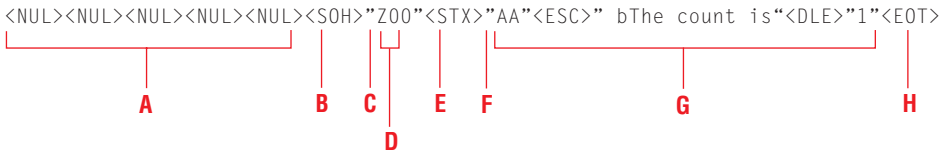
Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	This is the "Write SPECIAL FUNCTIONs" Command Code. (See "SPECIAL FUNCTION commands" on page 21.)
G	Data Field	"$AAU0400FF001 BL00200000"	"$" (24H) is the Write SPECIAL FUNCTIONs Command Code for **Set Memory Configuration** (see Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21). The remaining characters have the following meaning: "A" = File Label of the TEXT file which will "call" the STRING file "A" = TEXT File Type "U" = this TEXT file is Unlocked "0400" = the TEXT file size in hexadecimal ("0400" = 1024D) "FF" = the TEXT file's Start Time ("FF" = Always) "00" = the TEXT file's Stop Time (even though the TEXT message will always run, "00" must be included as padding) "1" = File Label of the STRING file "B" = STRING File Type "L" = this STRING file is Locked "0020" = the STRING file size in hexadecimal ("0020" = 32D). "0000" = padding
H	<EOT>	04H	End Of Transmission character

7.4.2.2 STEP 2 — Write the TEXT file which calls the STRING file

After allocating memory for the TEXT and the STRING files, write the TEXT file which will call the STRING file:

Table 35: Using STRING files example: STEP 2

This following is a Standard Transmission packet (see page 10):



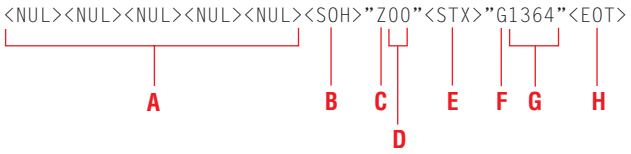
Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	"A"<ESC>" bThe count is "<DLE>"1"	The characters have the following meaning: "A" = File Label of the TEXT file which will include the STRING file "A" = TEXT File Type <ESC> (1BH) = signals the start of a Mode field " " (20H) = middle line position "b" = Hold Mode "The count is " = the text of this TEXT file <DLE> (10H) = Call STRING file "1" = the STRING File Label to call
H	<EOT>	04H	End Of Transmission character

7.4.2.3 STEP 3 — Update the STRING file

To update the STRING file data (e.g., "The count is 364"), this would be sent:

Table 36: Using STRING files example: STEP 3

This following is a Standard Transmission packet (see page 10):



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"G"	"G" (47H) is the "Write STRING file" Command Code. (See Table 18, "Write STRING file transmission packet format," on page 36.)
G	Data Field	"1364"	The characters have the following meaning: "1" = the STRING File Label to write to "364" = the STRING file data
H	<EOT>	04H	End Of Transmission character

7.5 Appendix E: Sample programs

Other sample programs will be included at Adaptive's FTP site:

ftp://ftp.ams-i.com/alpha_protocol_examples/.

7.5.1 Sample C program

```
/*****************************************************************
 *
 * Program Name..........SIMPLE C NETWORK PROGRAM NO LIBRARIES
 * Filename .............SIMPLEC.C
 * Version ..............1.0
 * Version Date .........February 27, 1991
 * Comments .............none
 *
 * COPYRIGHT (C) 1991 - 1998.  All Rights Reserved.
 * Adaptive Micro Systems, Inc.  Milwaukee, WI  USA.
 *
 *****************************************************************/

#define PORT_SETUP   0xde  /* = 4800 baud */
/*
#define PORT_SETUP   0x9e  /* = 1200 baud */
#define PORT_SETUP   0xbe  /* = 2400 baud */
#define PORT_SETUP   0xde  /* = 4800 baud */
#define PORT_SETUP   0xfe  /* = 9600 baud */
*/

#define COM_PORT     0  /* = com port 1 */

/*
#define COM_PORT     0  /* = com port 1 */
#define COM_PORT     1  /* = com port 2 */
*/
struct WORDREGS {
unsigned int ax, bx, cx, dx, si, di, cflag, flags;
};

struct BYTEREGS {
unsigned char   al, ah, bl, bh, cl, ch, dl, dh;
};

unionREGS  {
struct   WORDREGS x;
struct   BYTEREGS h;
};

main()
{
int x;
/* open the com port */
serinit();
/* send 20 nulls */
for (x = 0; x < 20; x++)
outc(0,COM_PORT);
outc(0x01,COM_PORT);   /* send a SOH */
outc("Z",COM_PORT);    /* send the sign type (Z = all signs, F = 480 etc) */
outc("0",COM_PORT);    /* send the address (00 = all signs) */
outc("0",COM_PORT);
outc(0x02,COM_PORT);   /* send a STX */
outc("A",COM_PORT);    /* send the command "WRITE TEXT file" */
outc("A",COM_PORT);    /* send TEXT File Label to write to (A = default) */
outc(0x1b,COM_PORT);   /* send an escape (precedes all mode commands) */
outc(0x20,COM_PORT);   /* send a position code (0x20 = middle full height) */
outc("b",COM_PORT);    /* send a mode (b = hold) */
outs("HELLO",COM_PORT);/* send out the string of characters */
outc(0x04,COM_PORT);   /* send out the EOT to end the transmission */
return(0);
}

/* function that outputs a string to the com port */
outs (unsigned char *s,int port)
{
while (*s)
outc(*s++,port);
```

```
return(0);
}

/* function that outputs a char to the com port */
outc (unsigned char c,int port)
{
union REGS regs;
regs.h.ah = 01;
regs.h.al = c;
regs.x.dx = port;
int86(0x14,&regs,&regs);/* Turbo C function which triggers the serial interrupt.
Check compiler for similar function */
return(0);
}

/* function which opens the com port */
serinit()
{
union REGS regs;
regs.h.ah = 0;
regs.h.al = PORT_SETUP;
regs.x.dx = COM_PORT;
int86(0x14,&regs,&regs);
return(0);
}
```

7.5.2 Sample BASIC program

```
 10 CLS:PRINT"ALPHA NETWORK INSTALL PROGRAM":PRINT:PRINT:INPUT "COMMUNICATION PORT
(1 OR 2) :";A$
 20 IF A$ = "1" THEN OPEN "COM1:4800,E,7,,CS,DS,CD" AS #1
 30 IF A$ = "2" THEN OPEN "COM2:4800,E,7,,CS,DS,CD" AS #1
 35 IF A$ <> "1" AND A$ <> "2" THEN CLS:PRINT "ERROR IN COM PORT SELECTION":END
 40 REM
 50 REM OPEN THE COMMUNICATIONS PORT FOR 1200 BAUD 7 BITS EVEN PARITY
 60 REM ( NOTE: 4800 OR 9600 ETC CAN BE USED)
 70 REM
130 CLS
140 FOR X = 1 TO 20: PRINT #1, CHR$(0);:NEXT
150 REM
160 REM SEND 20 NULLS
170 REM
180 A$ = CHR$(1)+"Z00"+CHR$(2)+"AA"+CHR$(27)+" b"+STR$(Y)+CHR$(4)
190 REM
200 REM
210 REM CHR$(1)= START OF HEADER MARKER
220 REM "Z"= ALL SIGNS RESPOND ("E" = 460 ONLY)
230 REM "00"= ALL ADDRESSES RESPOND("01","02" ETC. CAN BE SUBSTITUTED)
240 REM CHR$(2)= START OF TEXT MARKER
250 REM "A"= WRITE TO TEXT file COMMAND
260 REM "A"= TEXT file LABEL ("A" FILE IS THE DEFAULT)
270 REM CHR$(27) = ESCAPE CODE TELLS SIGN THAT A MODE IS COMING
280 REM " " = BIG CHARS(OTHER CODES CAN BE SUB'D FOR TOP OR BOTTOM)
290 REM "b" = HOLD MODE (OTHER MODES CAN BE SUB'D)
300 REM STR$(Y) = TEXT TO BE DISPLAYED (IN THIS CASE ITS A NUMBER)
310 REM CHR$(4) = END OF TRANSMISSION MARKER
320 REM
330 PRINT #1, A$
340 REM
350 REM SEND THE MESSAGE TO THE SIGN
360 PRINT:PRINT "      ";Y
370 REM
380 FOR X = 1 TO 10000:NEXT
390 REM
400 REM DELAY A LITTLE
410 REM
420 Y = Y + 1: IF Y = 10000 THEN Y = 1
430 REM
440 REM INC THE COUNTER, RESET IF 10000
450 REM
460 REM DELAY A LITTLE
470 REM
480 GOTO 140
490 REM GO BACK AND LOOP AGAIN
```

7.6 Appendix F: Protocol examples

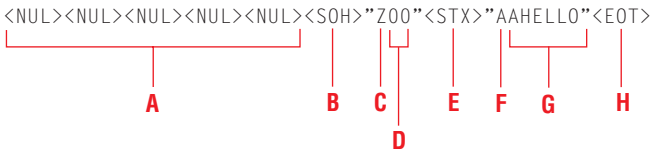
NOTE: In the following examples, it is assumed that the Memory Configuration table (**Table 15** on page 21) in each sign has already been set up properly.

7.6.1 Standard transmission packet examples

7.6.1.1 Send a message to all signs on a network example

The following example will display "HELLO" on all signs attached to a network:

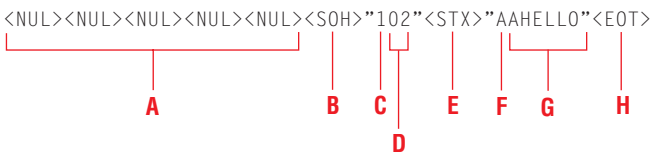
Table 37: Send a message to all signs example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	File Label	"A"	File Label of the TEXT file
		ASCII Message	"HELLO"	The actual text to be displayed on a sign
H	<EOT>		04H	End Of Transmission character

7.6.1.2 Send a message to all 1-line signs on a network with a Sign Address of 02H example

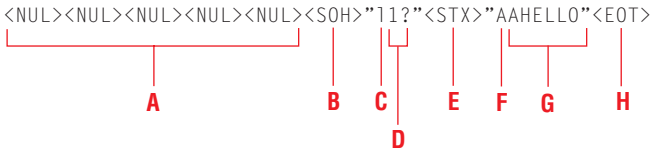
Table 38: Send a message to all 1-line signs on a network with a Sign Address of 02H example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"1"	This means that this transmission is directed to all 1-line signs.
D	Sign Address		"02"	This means only 1-line signs with a Sign Address of 02H on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	File Label	"A"	File Label of the TEXT file
		ASCII Message	"HELLO"	The actual text to be displayed on a sign
H	<EOT>		04H	End Of Transmission character

7.6.1.3 Send a message to all Series 7000 signs on a network with Sign Addresses 10H through 1FH example

Table 39: Send a message to all Series 7000 signs on a network with Sign Addresses 10H through 1FH example

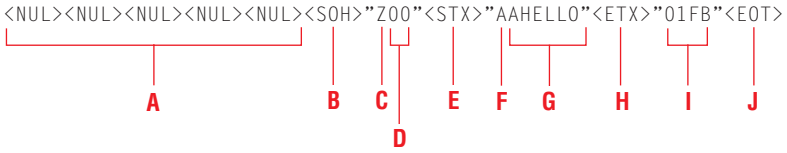


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"I"	"I" (6CH) means that this transmission is directed to all Series 7000 signs.
D	Sign Address		"1?"	"1?" (31H 3FH) means only Series 7000 signs with Sign Addresses between 10H and 1FH inclusive on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	File Label	"A"	File Label of the TEXT file
		ASCII Message	"HELLO"	The actual text to be displayed on a sign
H	<EOT>		04H	End Of Transmission character

7.6.2 Transmission packet with Checksum example

This example is identical to the previous example in Table 7.6.1.1, "Send a message to all signs on a network example," on page 57 except that a Checksum is used in the following example:

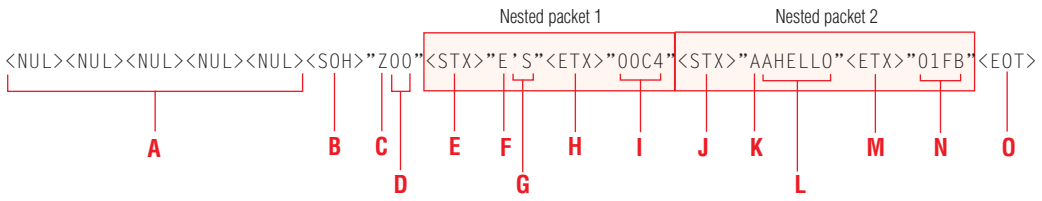
Table 40: Transmission packet with Checksum example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	File Label	"A"	File Label of the TEXT file
		ASCII Message	"HELLO"	The actual text to be displayed on a sign
H	<ETX>		03H	End of TeXt (03H) character
I	Checksum		"01FB"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item H) inclusive. The most significant digit is first.
J	<EOT>		04H	End Of Transmission character

7.6.3 Nesting with checksums transmission packet example

Table 41: Nesting with checksums transmission packet example

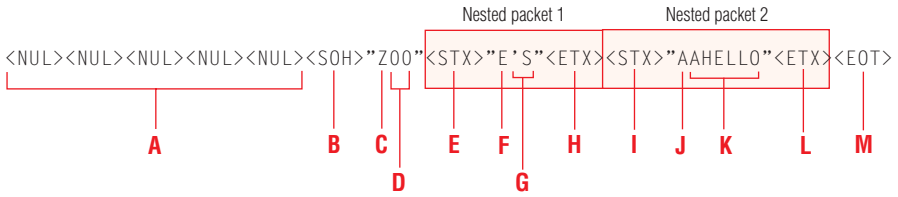


Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>			01H	Start Of Header character
C	Type Code			"Z"	"Z" (5AH) means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address			"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	Nested packet 1	<STX>		02H	Start of Nested packet 1
F	Nested packet 1	Command Code		"E"	"E" (45H) is the "Write SPECIAL FUNCTIONs" Command Code. (See Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21.)
G	Nested packet 1	Data Field	Special Functions Label	""	"" (27H) means Set Time Format
			Special Functions Data	"S"	"S" (53H) sets the sign's time to the standard am/pm format.
H	Nested packet 1	<ETX>		03H	End of Nested packet 1
I	Nested packet 1	Checksum		"00C4"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item I) inclusive. The most significant digit is first.
J	Nested packet 2	<STX>		02H	Start of Nested packet 2
K	Nested packet 2	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
L	Nested packet 2	Data Field	File Label	"A"	File Label of the TEXT file
			ASCII Message	"HELLO"	The actual text to be displayed on a sign
M	Nested packet 2	<ETX>		03H	End of Nested packet 2
N	Nested packet 2	Checksum		"01FB"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item H) inclusive. The most significant digit is first.
O	<EOT>			04H	End Of Transmission character

7.6.4 Nesting without Checksum transmission packet example

This packet is identical to the previous packet in **Table 41** on page 60 except that the Checksums are omitted after each nested packet's <ETX>:

Table 42: Nesting without Checksums transmission packet example



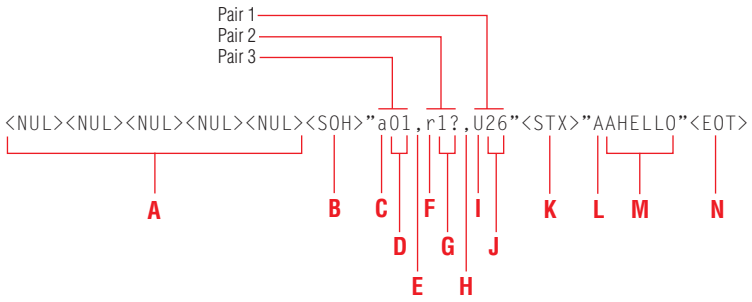
Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>			01H	Start Of Header character
C	Type Code			"Z"	"Z" (5AH) means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address			"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	Nested packet 1	<STX>		02H	Start of Nested packet 1
F	Nested packet 1	Command Code		"E"	"E" (45H) is the "Write SPECIAL FUNCTIONs" Command Code. (See Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21.)
G	Nested packet 1	Data Field	Special Functions Label	""	"" (27H) means Set Time Format
	Nested packet 1	Data Field	Special Functions Data	"S"	"S" (53H) sets the sign's time to the standard am/pm format.
H	Nested packet 1	<ETX>		03H	End of Nested packet 1
I	Nested packet 2	<STX>		02H	Start of Nested packet 2
J	Nested packet 2	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
K	Nested packet 2	Data Field	File Label	"A"	File Label of the TEXT file
	Nested packet 2	Data Field	ASCII Message	"HELLO"	The actual text to be displayed on a sign
L	Nested packet 2	<ETX>		03H	End of Nested packet 2 (Optional when <EOT> is the next character.)
M	<EOT>			04H	End Of Transmission character

7.6.5 Multiple Type Codes / Sign Addresses example

In this example three Type Code/Sign Address pairs are shown:

NOTE: The effects of Type Codes are cumulative. For instance, in this example the message would be sent to all 4120C signs and Director signs and 790i signs on the network.

Table 43: Multiple Type Codes / Sign Addresses example



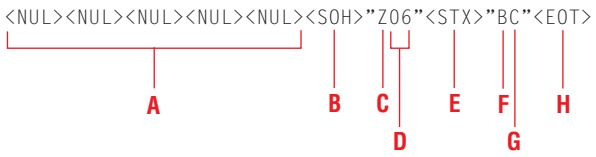
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Pair 1	Type Code	"a"	"a" (61H) means that this transmission is directed to all 4120C signs.
D	Pair 1	Sign Address	"01"	"01" (30H 31H) means only 4120C signs with a Sign Address of 01H on the network should "listen" to this transmission.
E	Delimiter		","	"," (2C) separates each Type Code/Sign Address pair.
F	Pair 2	Type Code	"r"	"r" (72H) means that this transmission is directed to all Director signs.
G	Pair 2	Sign Address	"1?"	"1?" (31H 3FH) means that all signs with a Sign Address between 10H and 1FH inclusive on the network should "listen" to this transmission.
H	Delimiter		","	"," (2CH) separates each Type Code/Sign Address pair.
I	Pair 3	Type Code	"U"	"U" (55H) means that this transmission is directed to all 790i signs.
J	Pair 3	Sign Address	"26"	"26" (32H 36H) means only 790i signs with a Sign Address of 26H on the network should "listen" to this transmission.
K	<STX>		02H	Start of TeXt character
L	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
M	Data Field	File Label	"A"	File Label of the TEXT file
M	Data Field	ASCII Message	"HELLO"	The actual text to be displayed on a sign
N	<EOT>		04H	End Of Transmission character

7.6.6 TEXT file examples

7.6.6.1 Read TEXT file example

The response to this read file request is shown in **Table 45** on page 63.

Table 44: Read TEXT file example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address		"06"	"06" (30H 36H) means only signs with a Sign Address of 06H on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"B"	"B" (42H) is the "Read TEXT file" Command Code. (See Table 13, "Read TEXT file transmission packet format," on page 19.)
G	Data Field	File Label	"C"	File Label of the TEXT file to read
H	<EOT>		04H	End Of Transmission character

7.6.6.2 Response to Read TEXT file example

This is the response to the read file request shown in the **Table 44** on page 63.

NOTE: For the sake of this example, we'll assume that the TEXT file with the File Label "C" just contains the text "FILE C".

Table 45: Response to Read TEXT file example

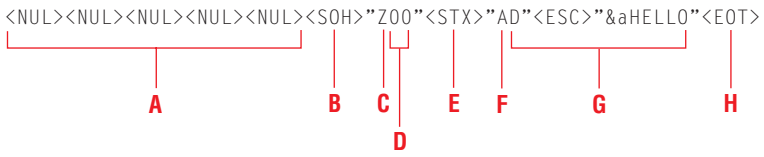
Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> characters
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	The Response Type Code
D	Sign Address	"00"	"00" (30H 30H) is always sent.
E	<STX>	02H	Start of TeXt character
F	Command Code	"A"	"A" (41H) is sent in response to the "Read TEXT file" Command Code.
G	File Label	"C"	File Label of the TEXT file that is being read
H	TEXT file data format	"FILE C"	The actual text stored in TEXT file "C"
I	<ETX>	03H	End of TeXt character
J	Checksum	"020C"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item H) inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

7.6.6.3 TEXT file data format examples

7.6.6.3.1 Rotate “Hello” example

This example uses the Rotate Mode to move the text “HELLO” on the bottom line of a sign:

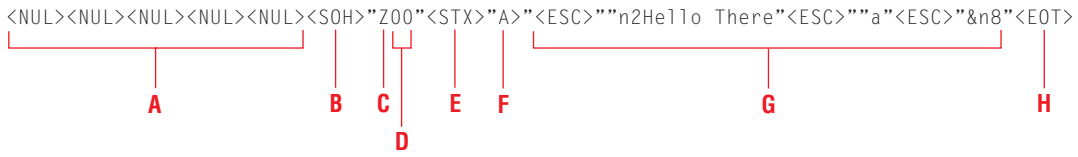
Table 46: Rotate “Hello” example



Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>			01H	Start Of Header character
C	Type Code			“Z”	“Z” (5AH) means that this transmission is directed to all signs.
D	Sign Address			“00”	“00” (30H 30H) means all signs on the network should “listen” to this transmission.
E	<STX>			02H	Start of TeXt character
F	Command Code			“A”	“A” (41H) is the “Write TEXT file” Command Code. (See Table 12, “Write TEXT file transmission packet format,” on page 18.)
G	Data Field	File Label		“D”	File Label of the TEXT file that will be written
		Mode Field	<ESC>	1BH	Escape character
			Display Position	“&”	“&” (26H) means that the ASCII Message should be displayed on the bottom line of a sign.
			Mode Code	“a”	“a” (61H) Rotate code.
		ASCII Message		“HELLO”	The actual text to be displayed
H	<EOT>			04H	End Of Transmission character

7.6.6.3.2 Combining text and graphics example

Table 47: Combining text and graphics example



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>	01H	Start Of Header character
C	Type Code	“Z”	“Z” (5AH) means that this transmission is directed to all signs.
D	Sign Address	“00”	“00” (30H 30H) means all signs on the network should “listen” to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	“A”	“A” (41H) is the “Write TEXT file” Command Code. (See Table 12, “Write TEXT file transmission packet format,” on page 18.)

Table 47: Combining text and graphics example

G	Data Field	File Label			">"	File Label of the TEXT file that will be written
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	""	"" (22H) means that the ASCII Message will begin on the Top Line of the sign
				Mode Code	"n"	"n" (6EH) is used in conjunction with the Special Specifier to use the Special Modes (see "The following would write a DOTS PICTURE file labeled "A", 15 pixel rows high x 9 pixel columns wide to a 4160C sign:" on page 75).
				Special Specifier	"2"	"2" (32H) means that the Special Mode called SNOW will be used.
			ASCII Message		"Hello There"	The actual text to be displayed
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	""	"" (22H) means the Top Line of the sign.
				Mode Code	"a"	"a" (61H) is the ROTATE Mode Code. This means that the previous ASCII Message ("Hello There") will be ROTATEd off the Top Line of the sign. This is often referred to as a "Trailing Mode".
			ASCII Message			In this case, there is no ASCII Message because of the "trailing" ROTATE Mode.
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	"&"	"&" (22H) means that the ASCII Message will begin on the Bottom Line of the sign
				Mode Code	"n"	"n" (6EH) is used in conjunction with the Special Specifier to use the Special Modes (see "The following would write a DOTS PICTURE file labeled "A", 15 pixel rows high x 9 pixel columns wide to a 4160C sign:" on page 75).
				Special Specifier	"8"	"8" (38H) means that the Special Mode called WELCOME will be used.
			ASCII Message			In this case, there is no ASCII Message because of the WELCOME animation.
H	<EOT>				04H	End Of Transmission character

7.6.6.3.3 Displaying a Counter value example

Table 48: Displaying a Counter value example

<NUL><NUL><NUL><NUL><NUL><SOH>"h00"<STX>"A1"<ESC>""bCongratulations!"<CR><BS>"z days without an accident!"<EOT>

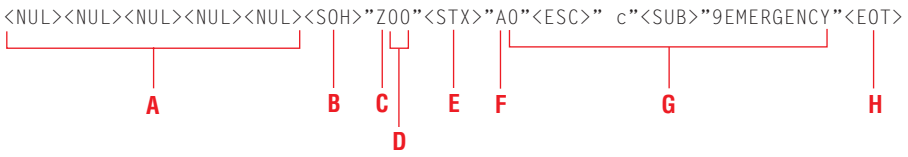
A B C D E F G H

Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>			01H	Start Of Header character
C	Type Code			"h"	"h" (68H) means that this transmission is directed to all 4160R signs.
D	Sign Address			"00"	"00" (30H 30H) means all 4160R signs on the network should "listen" to this transmission.
E	<STX>			02H	Start of TeXt character
F	Command Code			"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	File Label		"1"	File Label of the TEXT file
		TEXT file data format	Mode Field <ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
			Mode Field Display Position	"""	""" (22H) means that the ASCII Message will begin on the Top Line of the sign
			Mode Field Mode Code	"b"	"b" (62H) is the HOLD Mode Code (see page 75)
			ASCII Message	"Congratulations"<CR><BS>"z days without an accident!"	The actual text (with Control Codes) to be displayed on a sign. These Control Codes are used: <CR> (0DH) = means that text after the <CR> will be an the next line of the sign <BS> (08H) + "z" = a 2-byte code used to display a counter, in this case Counter 1 (see "Appendix C: Counter information" on page 51).
H	<EOT>			04H	End Of Transmission character

7.6.7 Priority TEXT file examples

7.6.7.1 Write a Priority TEXT file example

Table 49: Write a Priority TEXT file example

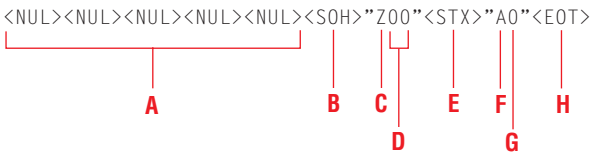


Item	Name				Value	Description
A	<NUL>				00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>				01H	Start Of Header character
C	Type Code				"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address				"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>				02H	Start of TeXt character
F	Command Code				"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	File Label			"0"	"0" (30H) Priority TEXT File Label
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	" "	" " (20H) means that the ASCII Message will be on the Middle Line of the sign
				Mode Code	"c"	"c" (62H) is the FLASH Mode Code (see page 75)
			ASCII Message		<SUB>"9EMERGENCY"	The actual text (with Control Codes) to be displayed on a sign. These Control Codes are used: <SUB> (1AH) + "9" = a 2-byte code used to select a character set, in this case Full Height Standard (see "Appendix G: Alpha protocol ASCII table" on page 80).
H	<EOT>				04H	End Of Transmission character

7.6.7.2 Disable a Priority TEXT file example

The following transmission will disable the Priority TEXT file. Whatever was running on a sign *before* the Priority TEXT file was sent will resume running.

Table 50: Disable a Priority TEXT file example



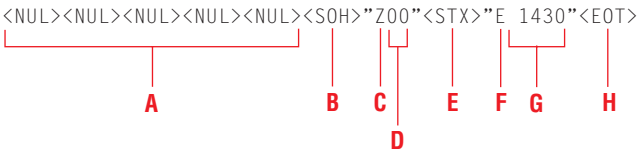
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address		"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"A"	"A" (41H) is the "Write TEXT file" Command Code. (See Table 12, "Write TEXT file transmission packet format," on page 18.)
G	Data Field	File Label	"0"	"0" (30H) Priority TEXT File Label
H	<EOT>		04H	End Of Transmission character

7.6.8 SPECIAL FUNCTION examples

7.6.8.1 Write SPECIAL FUNCTION example

The following sets the time on all networked signs to 2:30 pm (1430 in 24-hour format):

Table 51: Write SPECIAL FUNCTION example

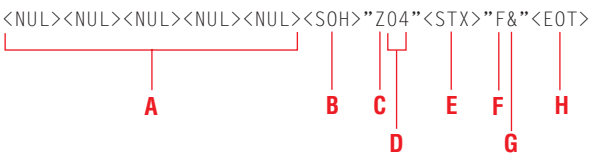


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address		"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"E"	"E" (45H) is the "Write SPECIAL FUNCTIONs file" Command Code. (See Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21.)
G	Data Field	Special Functions Label	" "	" " (20H) = Set Time of Day
		Special Functions Data	"1430"	The time to set (in 24-hour format)
H	<EOT>		04H	End Of Transmission character

7.6.8.2 Read SPECIAL FUNCTION example

The following reads the day of week from a sign with a Sign Address of 4:

Table 52: Read SPECIAL FUNCTION example

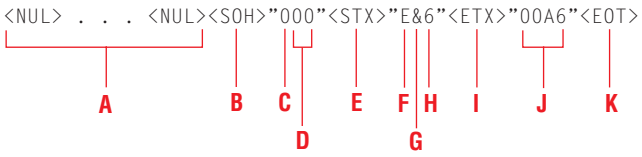


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address		"04"	"04" (30H 34H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"F"	"F" (46H) is the "Read SPECIAL FUNCTIONs file" Command Code. (See Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.)
G	Data Field	Special Functions Label	"&"	"&" (26H) Read Day of Week
H	<EOT>		04H	End Of Transmission character

7.6.8.3 Response to Read SPECIAL FUNCTION example

The following is the response to the Read SPECIAL FUNCTION example in **Table 52** above:

Table 53: Response to Read SPECIAL FUNCTION example



Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> (00H) characters
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	"0" (30H) is the Response code
D	Sign Address	"00"	"00" (30H 30H) is sent regardless of the sign's actual address.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	"E" (45H) is the "Read SPECIAL FUNCTIONs file" Command Code. (See Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.)
G	Special Functions Label	"&"	"&" (26H) Read Day of Week
H	Special Functions Data	"6"	"6" (36H) stands for Friday
I	<ETX>	03H	End of TeXt character
J	Checksum	"00A6"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

7.6.8.4 SPECIAL FUNCTION data formats example

7.6.8.4.1 Set Memory Configuration example #1 — Counter data not included

This example writes the following file information to all signs:

- a TEXT file “A”, unlocked, 265 (100H) bytes in length, to run always
- a DOTS PICTURE file “m”, unlocked, 7 x 60 (rows x columns), one color
- a STRING file “l”, locked, 10 bytes in length

Table 54: Set Memory Configuration example #1 — Counter data not included

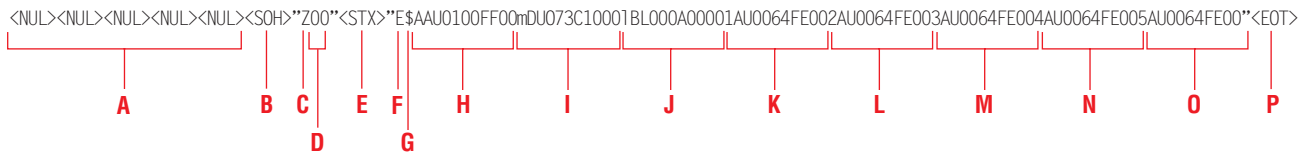
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	“Z” (5AH) means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		“00”	“00” (30H 30H) means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“E”	“E” (45H) is the “Write SPECIAL FUNCTIONs file” Command Code. (See Table 15, “Write SPECIAL FUNCTION Command Code format — “E” (45H),” on page 21.)
G	Special Functions Label		“$”	“$” (24H) means Set Memory Configuration
H	Special Functions Data	TEXT file	“AAU0100FF00”	These bytes mean the following: “A” = File Label “A” = file type (in this case, a STRING file) “U” = an unlocked file “0100” = the size of this file in bytes (256D) “FF” = the TEXT file’s Start Time (in this case Always) “00” = the TEXT file’s Stop Time (ignored when the Start Time is Always)
I		DOTS PICTURE file	“mDU073C1000”	These bytes mean the following: “m” = File Label “D” = file type (in this case, a DOTS PICTURE file) “U” = an unlocked file “07” = number of pixel rows in the DOTS PICTURE file (7D) “3C” = number of pixel columns in the DOTS PICTURE file (60D) “1000” = a monochrome DOTS PICTURE file
J		STRING file	“lBL000A0000”	These bytes mean the following: “l” = File Label “B” = file type (in this case, a TEXT file) “L” = a locked file “000A” = the size of this file in bytes (10D) “0000” = these are just placeholders for a STRING file
K	<EOT>		04H	End Of Transmission character

7.6.8.4.2 Set Memory Configuration example #2 — Counter data included

The Memory Configuration from the previous example (**Table 54**) is used. However, in this example, in order to use a sign's Counters, the five Target files must be set up. (See also "Appendix C: Counter information" on page 51.)

NOTE: Once a Current Counter Value reaches its Counter Target Value, all Target files are triggered (as set up in the Target File Byte). This means that the Start Times for the appropriate Target files will be automatically set to Always.

Table 55: Set Memory Configuration example #2 — Counter data included



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"E"	"E" (45H) is the "Write SPECIAL FUNCTIONs file" Command Code. (See Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21.)
G	Special Functions Label		"$"	"$" (24H) means Set Memory Configuration
H	Special Functions Data	TEXT file	"AAU0100FF00"	These bytes mean the following: "A" = File Label "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0100" = the size of this file in bytes (256D) "FF" = the TEXT file's Start Time (in this case Always) "00" = the TEXT file's Stop Time (ignored when the Start Time is Always)
I		DOTS PICTURE file	"mDU073C1000"	These bytes mean the following: "m" = File Label "D" = file type (in this case, a DOTS PICTURE file) "U" = an unlocked file "07" = number of pixel rows in the DOTS PICTURE file (7D) "3C" = number of pixel columns in the DOTS PICTURE file (60D) "1000" = a monochrome DOTS PICTURE file
J		STRING file	"lBL000A0000"	These bytes mean the following: "l" = File Label "B" = file type (in this case, a STRING file) "L" = a locked file "000A" = the size of this file in bytes (10D) "0000" = these are just placeholders for a STRING file
K		TEXT file (this is the Target File for Counter 1)	"1AU0064FE00"	These bytes mean the following: "1" = File Label for Counter 1 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)

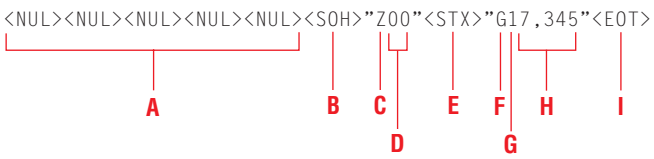
Table 55: Set Memory Configuration example #2 — Counter data included

L	Special Functions Data (continued)	TEXT file (this is the Target File for Counter 2)	"2AU0064FE00"	These bytes mean the following: "2" = File Label for Counter 2 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
M		TEXT file (this is the Target File for Counter 3)	"3AU0064FE00"	These bytes mean the following: "3" = File Label for Counter 3 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
N		TEXT file (this is the Target File for Counter 4)	"4AU0064FE00"	These bytes mean the following: "4" = File Label for Counter 4 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
O		TEXT file (this is the Target File for Counter 5)	"5AU0064FE00"	These bytes mean the following: "5" = File Label for Counter 5 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
P	<EOT>		04H	End Of Transmission character

7.6.9 STRING file examples

7.6.9.1 Write STRING file example

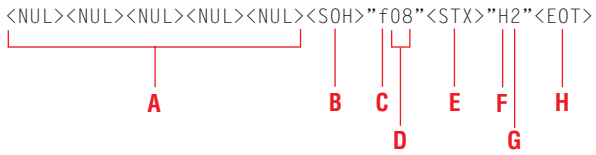
Table 56: Write STRING file example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address		"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"G"	"G" (47H) is the "Write STRING file" Command Code. (See Table 18, "Write STRING file transmission packet format," on page 36.)
G	Data Field	File Label	"1"	"1" (31H) File Label of the STRING file
H		STRING File Data	"7,345"	This is the actual STRING file data.
I	<EOT>		04H	End Of Transmission character

7.6.9.2 Read STRING file example

Table 57: Read STRING file example

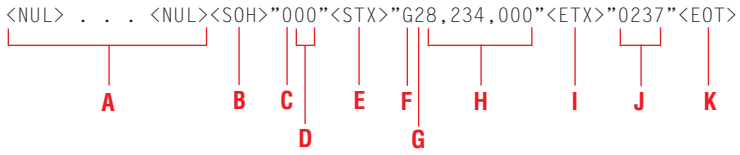


Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"f"	"f" (66H) means that this transmission is directed to all 215C signs.
D	Sign Address	"08"	"08" (30H 38H) means all 215C signs with an address of 08H on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"H"	"H" (48H) is the "Read STRING file" Command Code. (See Table 19, "Read STRING file transmission packet format," on page 37.)
G	File Label	"2"	File Label of the STRING file to read
H	<EOT>	04H	End Of Transmission character

7.6.9.3 Response to Read STRING file example

The following would be the response from the previous (**Table 57**) example:

Table 58: Response to Read STRING file example



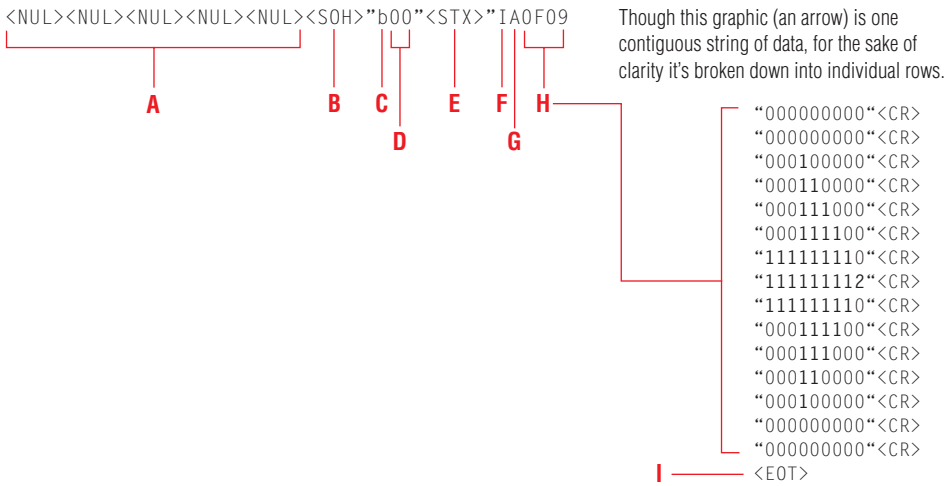
Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> (00H) characters
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	"0" (30H) is the Response code
D	Sign Address	"00"	"00" (30H 30H) is sent regardless of the sign's actual address.
E	<STX>	02H	Start of TeXt character
F	Command Code	"G"	"G" (47H) is returned by the sign
G	File Label	"2"	"2" (32H) is the File Label of the STRING file accessed
H	STRING File Data	"8,234,000"	The actual data in the STRING file
I	<ETX>	03H	End of TeXt character
J	Checksum	"0237"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

7.6.10 DOTS PICTURE file examples

7.6.10.1 Write DOTS PICTURE file example

The following would write a DOTS PICTURE file labeled "A", 15 pixel rows high x 9 pixel columns wide to a 4160C sign:

Table 59: Write DOTS PICTURE file example



Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>			01H	Start Of Header character
C	Type Code			"b"	"b" (62H) means that this transmission is directed to all 4160C signs.
D	Sign Address			"00"	"00" (30H 30H) means all 4160C signs on the network should "listen" to this transmission.
E	<STX>			02H	Start of TeXt character
F	Command Code			"I"	"I" (49H) is the "Write SMALL DOTS PICTURE file" Command Code. (See Table 21, "Write SMALL DOTS PICTURE file transmission packet format," on page 38.)
G	File Label			"A"	File Label of the DOTS file
H	Data Field	DOTS PICTURE File Data	Height (y)	"0F"	"0F" (30H 46H) = pixel height of graphic = 15D
			Width (x)	"09"	"09" (30H 39H) = pixel width of graphic = 9D
			Row Bit Pattern	"000000000"<CR> "000000000"<CR> "000**1**00000"<CR> "000**11**0000"<CR> "000**111**000"<CR> "000**1111**00"<CR> "**111111110**"<CR> "**111111112**"<CR> "**111111110**"<CR> "000**1111**00"<CR> "000**111**000"<CR> "000**11**0000"<CR> "000**1**00000"<CR> "000000000"<CR> "000000000"<CR>	Each row of the graphic is followed by a <CR> (0DH). "0" (30H) = sign pixel off "1" (31H) = sign pixel on - red "2" (32H) = sign pixel on - green
I	<EOT>			04H	End Of Transmission character

7.6.11 Displaying text at XY position examples

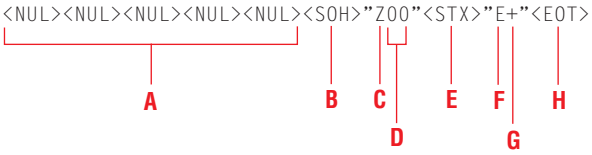
Text messages up to 250 characters can be displayed in a particular location on AlphaVision character matrix sign. This can be done by specifying a character position in a sign line (X) and a line position (Y) using the SPECIAL FUNCTION "+" command (see page 22).

The following examples will show how to:

- enable XY positioning
- display text at an XY location
- display multiple text at XY locations
- disable XY positioning

7.6.11.1 Enable SPECIAL FUNCTION XY positioning example

Table 60: Enable SPECIAL FUNCTION XY positioning example



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address	"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	"E" (45H) is the "Write SPECIAL FUNCTION example" on page 68.
G	Data Field	"+"	"+" (2BH) Enable XY positioning
H	<EOT>	04H	End Of Transmission character

7.6.11.2 Display text at an XY location example

The following example shows how to display text in a specified location on an imaginary 4-line x 20-character AlphaVision character matrix sign.

The text "Hello world!" will be displayed starting at character position 14 (X) on line 2 (Y) as shown in the illustration below.

NOTE: Counting starts from 0, not 1, for both the X and the Y location.

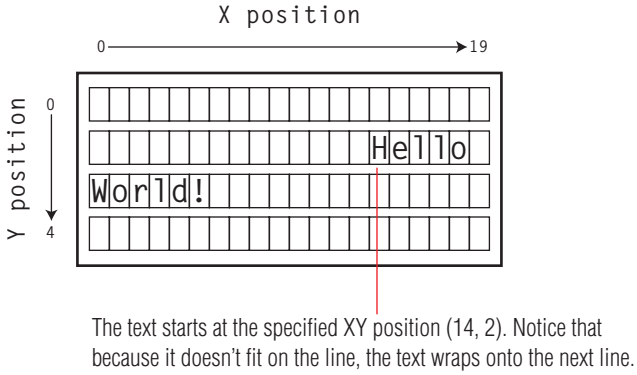
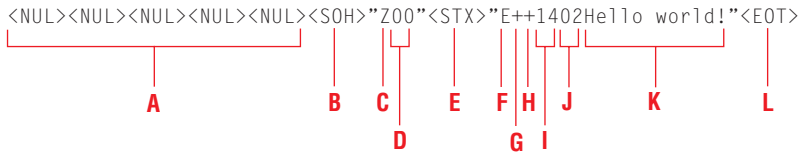


Table 61: Display text at an XY location example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address		"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"E"	"E" (45H) is the "Write SPECIAL FUNCTION example" on page 68.
G	Data Field	Special Functions Label	"+"	"+" (2BH) Enable XY positioning
H		File Label	"+"	File Label
I		X position	"14"	"14" (31H 34H) Two ASCII decimal digits that represent the character position
J		Y position	"02"	"02" (30H 32H) Two ASCII decimal digits that represent the line position
K		Message Text	"Hello world!"	ASCII message text (up to 250 characters)
L	<EOT>		04H	End Of Transmission character

7.6.11.3 Display multiple text at XY locations example

The following example shows how to display three text messages at 3 different locations:

"Hello world!" starts at $X_1 = 4$, $Y_1 = 0$.
"This is a test" starts at $X_2 = 0$, $Y_2 = 3$.
"The end" starts at $X_3 = 13$, $Y_3 = 4$.

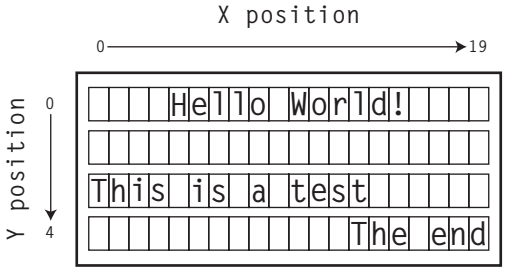


Table 62: Display multiple text at XY locations example

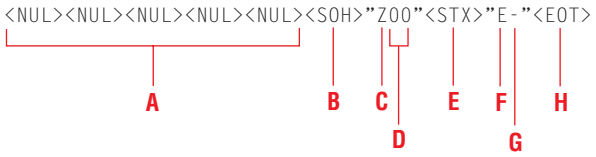
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"E++0400Hello world!"<DC2>"0003This is a test"<DC2>"1304The end"<EOT>

A B C D E F G H I J K L M N O P Q R S T

Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address		"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"E"	"E" (45H) is the "Write SPECIAL FUNCTION example" on page 68.
G	Data Field	Special Functions Label	"+"	"+" (2BH) Enable XY positioning
H		File Label	"+"	File Label
I		X_1 position	"04"	"04" (30H 34H) Two ASCII decimal digits that represent the character position of the first text message
J		Y_1 position	"00"	"00" (30H 30H) Two ASCII decimal digits that represent the line position of the first text message
K		Message Text 1	"Hello world!"	First ASCII message text (up to 250 characters)
L		<DC2>	12H	Device Control 2 character which signals another XY position
M		X_2 position	"00"	"00" (30H 30H) Two ASCII decimal digits that represent the character position of the second text message
N		Y_2 position	"03"	"03" (30H 33H) Two ASCII decimal digits that represent the line position of the second text message
O		Message Text 2	"This is a test"	Second ASCII message text (up to 250 characters)
P		<DC2>	12H	Device Control 2 character which signals another XY position
Q		X_3 position	"13"	"13" (31H 33H) Two ASCII decimal digits that represent the character position of the third text message
R		Y_3 position	"04"	"04" (30H 34H) Two ASCII decimal digits that represent the line position of the third text message
S		Message Text 3	"The end"	Third ASCII message text (up to 250 characters)
T	<EOT>		04H	End Of Transmission character

7.6.11.4 Disable SPECIAL FUNCTION XY positioning example

Table 63: Disable SPECIAL FUNCTION XY positioning example



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address	"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	"E" (45H) is the "Write SPECIAL FUNCTION example" on page 68.
G	Data Field	"-"	"-" (2DH) Disable XY positioning
H	<EOT>	04H	End Of Transmission character

7.7 Appendix G: Alpha protocol ASCII table

7.7.1 Standard character set (00 –7FH)

7.7.1.1 Control codes (00 – 1FH)

	Dec	Hex	Character	Meaning
Control codes	0	00	^@	NUL
	1	01	^A	SOH
	2	02	^B	STX
	3	03	^C	ETX
	4	04	^D	EOT
	5	05	^E	Double high characters (2-byte format) • 05H + "0" (30H) = Double height off (default) • 05H + "1" (31H) = Double height on
	6	06	^F	True descenders (2-byte format) • 06H + "0" (30H) = True descenders off (default) • 06H + "1" (31H) = True descenders on
	7	07	^G	Character flash (2-byte format) • 07H + "0" (30H) = Character flash off (default) • 07H + "1" (31H) = Character flash on
	8	08	^H	Extended character sets (2-byte format) • 08H + Offset (20H through 61H) (see the following "Extended character set") Display temperature (2-byte format): • 08H + "^\" (1CH) = display temperature in Celsius (only on Solar, 790i, 460i, 440i, and 430i) • 08H + "^]" (1DH) = display temperature in Fahrenheit (only on Solar, 790i, 460i, 440i, and 430i)
	9	09	^I	No Hold speed — when used, there will be virtually no pause following the mode presentation. This is not applicable for the Rotate or Compressed Rotate modes.
	10	0A	^J	
	11	0B	^K	Call date (2-byte format) — the date will be displayed, where DD = date, MM = month, YY = year, MMM = month abbreviation, and YYYY = year: • 0BH + "0" (30H) = MM/DD/YY • 0BH + "1" (31H) = DD/MM/YY • 0BH + "2" (32H) = MM-DD-YY • 0BH + "3" (33H) = DD-MM-YY • 0BH + "4" (34H) = MM.DD.YY • 0BH + "5" (35H) = DD.MM.YY • 0BH + "6" (36H) = MM DD YY • 0BH + "7" (37H) = DD MM YY • 0BH + "8" (38H) = MMM.DD, YYYY • 0BH + "9" (39H) = Day of week
	12	0C	^L	New page — start of next display page
	13	0D	^M	New line — start of new line
	14	0E	^N	
	15	0F	^O	Speed control — see "Speed control" on page 98. (Alpha 2.0 protocol only)
	16	10	^P	Call STRING file (2-byte format) — must be followed by a STRING File Label.
	17	11	^Q	Disable wide characters
	18	12	^R	Enable wide characters
	19	13	^S	Call Time — time of day will be called up.
	20	14	^T	Call SMALL DOTS PICTURE file (2-byte format) — must be followed by a DOTS PICTURE File Label.
	21	15	^U	Speed 1 (slowest)
	22	16	^V	Speed 2
	23	17	^W	Speed 3
	24	18	^X	Speed 4
	25	19	^Y	Speed 5 (fastest)

	Dec	Hex	Character	Meaning
Control codes (continued)	26	1A	^Z	Select character set (2-byte format): • 1AH + "1" (31H) = Five high standard (or Five slim[1]) • 1AH + "2" (32H) = Five stroke[1] • 1AH + "3" (33H) = Seven high standard (or Seven slim[1]) • 1AH + "4" (34H) = Seven stroke[1] • 1AH + "5" (35H) = Seven high fancy (or Seven slim fancy[1]) • 1AH + "6" (36H) = Ten high standard (or Seven stroke fancy[1]) • 1AH + "7" (37H) = Seven shadow[1] • 1AH + "8" (38H) = Full height fancy (or Wide stroke seven fancy[1]) • 1AH + "9" (39H) = Full height standard (or Wide stroke seven[1]) • 1AH + ":" (3AH) = Seven shadow fancy[1] • 1AH + ";" (3BH) = Five wide[1] • 1AH + "<" (3CH) = Seven wide[1] • 1AH + "=" (3DH) = Seven fancy wide[1] • 1AH + ">" (3EH) = Wide stroke five[1] • 1AH + "W" (57H) = (Alpha 2.0 and 3.0 protocols only)[2] Five high custom character set • 1AH + "X" (58H) = (Alpha 2.0 and 3.0 protocols only)[2] Seven high custom character set • 1AH + "Y" (59H) = (Alpha 2.0 and 3.0 protocols only)[2] Ten high custom character set • 1AH + "Z" (5AH) = (Alpha 2.0 and 3.0 protocols only)[2] Fifteen high custom character set [1] only applies to Betabrite model 1036 signs. [2] see "Custom character sets" on page 104.
	27	1B	^[	Start of Mode field
	28	1C	^\	Select character color (some signs do not support all the following colors): • 1CH + "1" (31H) = Red • 1CH + "2" (32H) = Green • 1CH + "3" (33H) = Amber • 1CH + "4" (34H) = Dim red • 1CH + "5" (35H) = Dim green • 1CH + "6" (36H) = Brown • 1CH + "7" (37H) = Orange • 1CH + "8" (38H) = Yellow • 1CH + "9" (39H) = Rainbow 1 • 1CH + "A" (41H) = Rainbow 2 • 1CH + "B" (42H) = Color mix • 1CH + "C" (43H) = Autocolor • 1CH + "ZRRGGBB" = (Alpha 3.0 protocol only. AlphaEclipse 3500 sign only.) Change the font color to this RGB value ("RRGGBB" = Red, Green, and Blue color intensities in ASCII hexadecimal from "00" to "FF".) • 1CH + "YRRGGBB" = (Alpha 3.0 protocol only. AlphaEclipse 3500 sign only.) Change the color of the shaded portion of the font to this RGB value ("RRGGBB" = Red, Green, and Blue color intensities in ASCII hexadecimal from "00" to "FF".)
	29	1D	^]	Select character attribute (3-byte format) — 1st byte is control code; 2nd byte is the attribute; and 3rd byte specifies either ON ["1" (31H)] or OFF ["0" (30H)]. OFF is the default setting for all of the following: • 1DH + "0" (30H) + "1" or "0" = Wide ON or OFF • 1DH + "1" (31H) + "1" or "0" = Double wide ON or OFF • 1DH + "2" (32H) + "1" or "0" = Double high ON or OFF • 1DH + "3" (33H) + "1" or "0" = True descenders ON or OFF • 1DH + "4" (34H) + "1" or "0" = Fixed width ON or OFF • 1DH + "5" (35H) + "1" or "0" = Fancy ON or OFF • 1DH + "6" (36H) + "1" or "0" = Auxiliary Port ON or OFF (Series 4000 & 7000 signs only.) • 1DH + "7" (37H) + "1" or "0" = Shadow characters ON or OFF (Betabrite model 1036 and AlphaPremiere 9000 signs only)
	30	1E	^^	Select character spacing (2-byte format) • 1EH + "0" (30H) = Proportional characters (default) • 1EH + "1" (31H) = Fixed width left justified characters
	31	1F	^_	Call picture or animation file (15-byte format): The display is cleared before each picture or animation is shown. 1FH + SFFFFFFFFFtttt where • S = "C" (43H) = Quick Flick animation. S = "G" (47H) = Faster Flicks animation (Alpha 3.0 protocol only). Hold times are in hundreths of seconds (0.01). S = "L" (4CH) = DOTS PICTURE file. If text from a TEXT file is displayed with the DOTS PICTURE file, the display hold time is ignored and the TEXT file display speed is used instead. • FFFFFFFFF (9 bytes) = file name. If the file name is less than 9 characters, spaces (20H) should precede the file name, so that the total number of characters is always fixed at 9. • tttt (4 bytes) — display hold time. A 4-digit ASCII hex number indicating tenths of seconds (0.1) for Quick Flick animations and DOTS PICTURE files and hundreths of seconds (0.01) for Faster Flicks animations. Leading 0's are ignored. For example, for a Quick Flick animation, "0020" = 32 tenths of seconds (32 x 0.1) = 3.2 seconds.

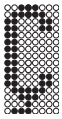
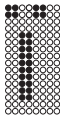
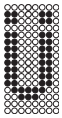
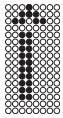
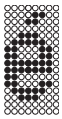
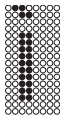
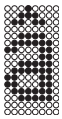
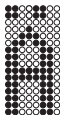
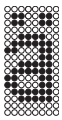
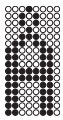
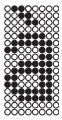
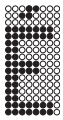
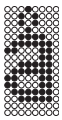
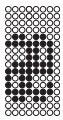
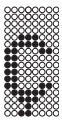
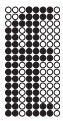
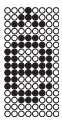
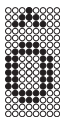
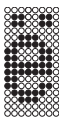
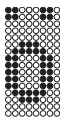
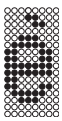
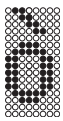
7.7.1.2 Standard ASCII characters (20 – 7FH)

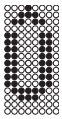
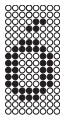
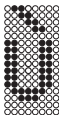
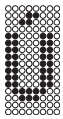
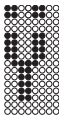
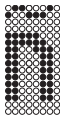
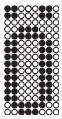
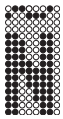
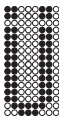
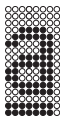
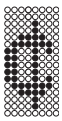
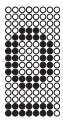
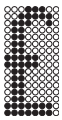
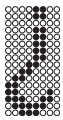
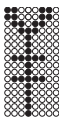
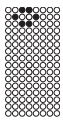
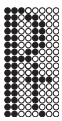
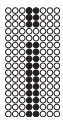
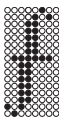
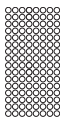
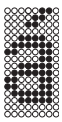
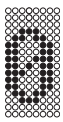
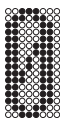
	Dec	Hex	Character	Dec	Hex	Character
Standard ASCII characters	32	20	space	80	50	P
	33	21	!	81	51	Q
	34	22	“	82	52	R
	35	23	#	83	53	S
	36	24	$	84	54	T
	37	25	%	85	55	U
	38	26	&	86	56	V
	39	27	‘	87	57	W
	40	28	(	88	58	X
	41	29	)	89	59	Y
	42	2A	*	90	5A	Z
	43	2B	+	91	5B	[
	44	2C	,	92	5C	\
	45	2D	-	93	5D	]
	46	2E	.	94	5E	¢
	47	2F	/	95	5F	_
	48	30	0	96	60	‘
	49	31	1	97	61	a
	50	32	2	98	62	b
	51	33	3	99	63	c
	52	34	4	100	64	d
	53	35	5	101	65	e
	54	36	6	102	66	f
	55	37	7	103	67	g
	56	38	8	104	68	h
	57	39	9	105	69	i
	58	3A	:	106	6A	j
	69	3B	;	107	6B	k
	60	3C	<	107	6C	l
	61	3D	=	109	6D	m
	62	3E	>	110	6E	n
	63	3F	?	111	6F	o
	64	40	@	112	70	p
	65	41	A	113	71	q
	66	42	B	114	72	r
	67	43	C	115	73	s
	68	44	D	116	74	t
	69	45	E	117	75	u
	70	46	F	118	76	v
	71	47	G	119	77	w
	72	48	H	120	78	x
	73	49	I	121	79	y
	74	4A	J	122	7A	z
	75	4B	K	123	7B	{
	76	4C	L	124	7C	l
	77	4D	M	125	7D	}
	78	4E	N	126	7E	1/2 space
	79	4F	O	127	7F	block

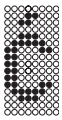
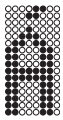
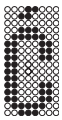
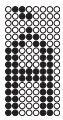
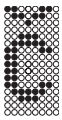
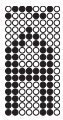
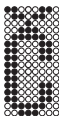
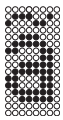
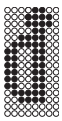
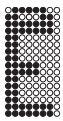
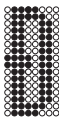
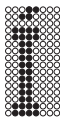
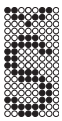
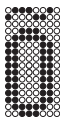
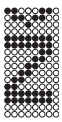
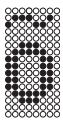
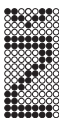
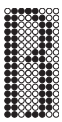
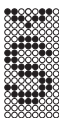
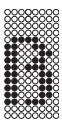
7.7.2 Extended character set (80 - C1H)

The following characters can be displayed by combining a control code (^H) with an offset (as shown below).

NOTE: This character set is not available with the 5-high character set.

	Dec	Hex	Character	Control code combination	Dec	Hex	Character	Control code combination
Extended character set	128	80		08H + 20H	139	8B		08H + 2BH
	129	81		08H + 21H	140	8C		08H + 2CH
	130	82		08H + 22H	141	8D		08H + 2DH
	131	83		08H + 23H	142	8E		08H + 2EH
	132	84		08H + 24H	143	8F		08H + 2FH
	133	85		08H + 25H	144	90		08H + 30H
	134	86		08H + 26H	145	91		08H + 31H
	135	87		08H + 27H	146	92		08H + 32H
	136	88		08H + 28H	147	93		08H + 33H
	137	89		08H + 29H	148	94		08H + 34H
	138	8A		08H + 2AH	149	95		08H + 35H

	Dec	Hex	Character	Control code combination	Dec	Hex	Character	Control code combination
Extended character set (cont)	150	96		08H + 36H	162	A2		08H + 42H
	151	97		08H + 37H	163	A3		08H + 43H
	152	98		08H + 38H	164	A4		08H + 44H
	153	99		08H + 39H	165	A5		08H + 45H
	154	9A		08H + 3AH	166	A6		08H + 46H
	155	9B		08H + 3BH	167	A7		08H + 47H
	156	9C		y08H + 3CH	168	A8		08H + 48H
	157	9D		08H + 3DH	169	A9		08H + 49H
	158	9E		08H + 3EH	170	AA		08H + 4AH
	159	9F		08H + 3FH	171	AB		08H + 4BH
	160	A0		08H + 40H	172	AC		08H + 4CH
	161	A1		08H + 41H	173	AD		08H + 4DH

	Dec	Hex	Character	Control code combination	Dec	Hex	Character	Control code combination
Extended character set (cont)	174	AE		08H + 4EH	186	BA		08H + 5AH
	175	AF		08H + 4FH	187	BB		08H + 5BH
	176	BO		08H + 50H	188	BC		08H + 5CH
	177	B1		08H + 51H	189	BD		08H + 5DH
	178	B2		08H + 52H	190	BE		08H + 5EH
	179	B3		08H + 53H	191	BF		08H + 5FH
	180	B4		08H + 54H	192	C0		08H + 60H
	181	B5		08H + 55H	193	C1		08H + 61H
	182	B6		08H + 56H	194	C2	EURO symbol	08H + 62H
	183	B7		08H + 57H	195	C3	Y punctuation key	08H + 63H[1]
	184	B8		08H + 58H	196	C4	Up arrow	08H + 64H[1]
	185	B9		08H + 59H	197	C5	Down arrow	08H + 65H[1]

	Dec	Hex	Character	Control code combination
Extended character set (cont)	198	C6	Left arrow	08H + 66H[1]
	199	C7	Right arrow	08H + 67H[1]
	200	C8	Packman	08H + 68H[1]
	201	C9	Sail boat	08H + 69H[1]
	202	CA	Ball	08H + 6AH[1]
	203	CB	Telephone	08H + 6BH[1]
	204	CC	Heart	08H + 6CH[1]
	205	CD	Car	08H + 6DH[1]
	206	CE	Handicap	08H + 6EH[1]
	207	CF	Rhino	08H + 6FH[1]
Special commands	208	D0	Mug	08H + 70H[1]
	209	D1	Satellite dish	08H + 71H[1]
	210	D2	Copyright symbol	08H + 72H[1]
	211	D3	Male symbol	08H + 73H[1]
	212	D4	Female symbol	08H + 74H[1]
	213	D5	Bottle	08H + 75H[1]
	214	D6	Diskette	08H + 76H[1]
	215	D7	Printer	08H + 77H[1]
	216	D8	Musical note	08H + 78H[1]
	217	D9	Infinity symbol	08H + 79H[1]
	Temperature			08H + "^\" (1CH)[2]
				08H + "^]" (1DH)[2]
	Counters			08H + "z" (7AH) Displays the current value in Counter 1.
				08H + "{" (7BH) Displays the current value in Counter 2.
				08H + "|" (7CH) Displays the current value in Counter 3.
				08H + "}" (7DH) Displays the current value in Counter 4.
				08H + "~" (7EH) Displays the current value in Counter 5.

Dec	Hex	Character	Control code combination

NOTES:

[1] Only applies to Betabrite 1036, AlphaPremiere 9000, and AlphaEclipse signs.

[2] Displays temperature in Celsius (only on Solar, 790i, 460i, 440i, 430i, and AlphaEclipse signs).

7.8 Appendix H: ISO ASCII table

This is the standard ASCII character set:

	Character			Hex	Dec			Character		Hex	Dec
Control characters	NUL	^@	null	00	0		Uppercase letters	@		40	64
	SOH	^A	start of heading	01	1			A		41	65
	STX	^B	start of text	02	2			B		42	66
	ETX	^C	end of text	03	3			C		43	67
	EOT	^D	end of transmission	04	4			D		44	68
	ENQ	^E	enquiry	05	5			E		45	69
	ACK	^F	acknowledge	06	6			F		46	70
	BEL	^G	bell	07	7			G		47	71
	BS	^H	backspace	08	8			H		48	72
	HT	^I	horizontal tab	09	9			I		49	73
	LF, NL	^J	line feed, new line	0A	10			J		4A	74
	VT	^K	vertical tab	0B	11			K		4B	75
	FF, NP	^L	form feed, new page	0C	12			L		4C	76
	CR	^M	carriage return	0D	13			M		4D	77
	SO	^N	shift out	0E	14			N		4E	78
	SI	^O	shift in	0F	15			O		4F	79
	DLE	^P	data link escape	10	16			P		50	80
	DC1	^Q	device control 1	11	17			Q		51	81
	DC2	^R	device control 2	12	18			R		52	82
	DC3	^S	device control 3	13	19			S		53	83
	DC4	^T	device control 4	14	20			T		54	84
	NAK	^U	negative acknowledge	15	21			U		55	85
	SYN	^V	synchronous idle	16	22			V		56	86
	ETB	^W	end of transmission block	17	23			W		57	87
	CAN	^X	cancel	18	24			X		58	88
	EM	^Y	end of medium	19	25			Y		59	89
	SUB	^Z	substitute	1A	26			Z		5A	90
	ESC	^[	escape	1B	27			[		5B	91
	FS	^\	file separator	1C	28			\		5C	92
	GS	^]	group separator	1D	29			]		5D	93
	RS	^^	record separator	1E	30			^		5E	94
	US	^_	unit separator	1F	31			_		5F	95
Special characters and numbers			space	20	32		Lowercase letters	`		60	96
		!		21	33			a		61	97
		"		22	34			b		62	98
		#		23	35			c		63	99
		$		24	36			d		64	100
		%		25	37			e		65	101
		&		26	38			f		66	102
		'		27	39			g		67	103
		(		28	40			h		68	104
		)		29	41			i		69	105
		*		2A	42			j		6A	106
		+		2B	43			k		6B	107
		,		2C	44			l		6C	108
		-		2D	45			m		6D	109
		.		2E	46			n		6E	110
		/		2F	47			o		6F	111
		0		30	48			p		70	112
		1		31	49			q		71	113
		2		32	50			r		72	114
		3		33	51			s		73	115
		4		34	52			t		74	116
		5		35	53			u		75	117
		6		36	54			v		76	118
		7		37	55			w		77	119
		8		38	56			x		78	120
		9		39	57			y		79	121
		:		3A	58			z		7A	122
		;		3B	69			{		7B	123
		<		3C	60			|		7C	124
		=		3D	61			}		7D	125
		>		3E	62			~		7E	126
		?		3F	63			DEL		7F	127

7.9 Appendix I: Modes, fonts, colors, and display options available on signs

Modes are ways of displaying information on a sign. For example, the ROTATE Mode makes text or graphics travel from right to left on a sign.

7.9.1 Standard Modes

When a Standard Mode Code of "n" (6EH) is given (see **Table 64**), the following Special Modes (**Table 65**) or Special Graphics (**Table 66**) can be designated in the Special Specifier field (see "TEXT file commands" on page 17).

Table 64: Standard Modes

Mode name	ASCII code	Hex code	Description
ROTATE	"a"	61H	Message travels right to left.
HOLD	"b"	62H	Message remains stationary.
FLASH	"c"	63H	Message remains stationary and flashes.
reserved	"d"	64H	
ROLL UP	"e"	65H	Previous message is pushed up by a new message.
ROLL DOWN	"f"	66H	Previous message is pushed down by a new message.
ROLL LEFT	"g"	67H	Previous message is pushed left by a new message.
ROLL RIGHT	"h"	68H	Previous message is pushed right by a new message.
WIPE UP	"i"	69H	New message is wiped over the previous message from bottom to top.
WIPE DOWN	"j"	6AH	New message is wiped over the previous message from top to bottom.
WIPE LEFT	"k"	6BH	New message is wiped over the previous message from right to left.
WIPE RIGHT	"l"	6CH	New message is wiped over the previous message from left to right.
SCROLL	"m"	6DH	New message line pushes the bottom line to the top line if 2-line sign.
AUTOMODE	"o"	6FH	Various Modes are called upon to display the message automatically.
ROLL IN	"p"	70H	Previous message is pushed toward the center of the display by the new message.
ROLL OUT	"q"	71H	Previous message is pushed outward from the center by the new message.
WIPE IN	"r"	72H	New message is wiped over the previous message in an inward motion.
WIPE OUT	"s"	73H	New message is wiped over the previous message in an outward motion.
COMPRESSED ROTATE	"t"	74H	Message travels right to left. Characters are approximately one half their normal width. (Only available on certain sign models.)
EXPLODE	"u"	75H	Message flies apart from the center (Alpha 3.0 protocol).
CLOCK	"v"	76H	Wipe in a clockwise direction (Alpha 3.0 protocol).
SPECIAL	"n"	6EH	This is followed by a Special Specifier ASCII character which defines one of the Special Modes. See "Special Modes" on page 88.

7.9.2 Special Modes

Table 65: Special Modes

Mode name	ASCII code	Hex code	Description (animations do NOT work on AlphaEclipse 3600 signs)	Will Mode appear on this length AlphaEclipse?	
				64 column	> 80 columns
TWINKLE	"0"	30H	Message will twinkle on the sign.	Yes	Yes
SPARKLE	"1"	31H	New message will sparkle over the current message.	Yes	Yes
SNOW	"2"	32H	Message will "snow" onto the display.	Yes	Yes
INTERLOCK	"3"	33H	New message will interlock over the current message in alternating rows of dots from each end.	Yes	Yes
SWITCH	"4"	34H	Alternating characters "switch" off the sign up and down. New message "switches" on in a similar manner.	Yes	Yes

Table 65: Special Modes

Mode name	ASCII code	Hex code	Description (animations do NOT work on AlphaEclipse 3600 signs)	Will Mode appear on this length AlphaEclipse?	
				64 column	> 80 columns
SLIDE or CYCLE COLORS[1]	"5"	35H	New message slides onto the sign one character at a time from right to left.	Yes[2]	Yes[2]
SPRAY	"6"	36H	New message sprays across and onto the sign from right to left.	Yes	Yes
STARBURST	"7"	37H	"Starbursts" explode the new message onto the sign (animation).	Yes	Yes
WELCOME	"8"	38H	The word "Welcome" is written in script across the sign (animation).	No	Yes
SLOT MACHINE	"9"	39H	Slot machine symbols appear randomly across the sign (animation).	No	Yes
NEWS FLASH[1]	"A"	3AH	News flash animation	—	—
TRUMPET ANIMATION[1]	"B"	3BH	Trumpet animation	—	—
CYCLE COLORS	"C"	43H	Color changes from one color to another.	Yes[3]	Yes[3]

[1] only available on Betabrite model 1036 signs
[2] SLIDE will appear, but COLOR CYCLE will only work on AlphaEclipse 3600 signs
[3] COLOR CYCLE will only work on AlphaEclipse 3600 signs

7.9.3 Special Graphics

Table 66: Special Graphics

Mode name	ASCII code	Hex code	Description (animations do NOT work on AlphaEclipse 3600 signs)	Will Mode appear on this length AlphaEclipse?	
				64 columns	> 80 columns
THANK YOU	"S"	53H	The words "Thank You" are written in script across the sign (animation).	No	Yes
NO SMOKING	"U"	55H	A cigarette image appears, is then extinguished and replaced with a no smoking symbol (animation).	No	Yes
DON'T DRINK & DRIVE	"V"	56H	A car runs into a cocktail glass and is replaced with the text "Please don't drink and drive" (animation)	No	Yes
RUNNING ANIMAL or FISH ANIMATION[1]	"W"	57H	An animal runs across the sign (animation).	Yes[2]	Yes[2]
FIREWORKS	"X"	58H	Fireworks explode randomly across the sign (animation).	Yes	Yes
TURBO CAR or BALOON ANIMATION[1]	"Y"	59H	A car drives across the sign (animation).	Yes	Yes
CHERRY BOMB	"Z"	5AH	A bomb fuse burns down followed by an explosion (animation).	Yes	Yes

[1] only available on Betabrite model 1036 signs
[2] FISH ANIMATION is only available on Betabrite model 1036 signs

7.9.4 Modes available on signs

Table 67: Modes available on signs

Signs	Modes																										
					Roll			Rotate										Switch				Wipe					
	Automode	Flash	Hold	Interlock	Up/Down/Left/Right	In/Out (horizontal)	In/Out (vertical)	Standard	Condensed	Scroll	Slide	Cycle Color	Snow	Sparkle	Spray	Starburst	Switch	Switch half the display	Twinkle	Up/Down/Left/Right	In /Out (horizontal)	In/Out (vertical)	Explode	Clock	Left/Right Display Position
200 Series[2]:	●	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●				
220C:	●	●	●	●	●		●	●	●	●	●[1]		●	●	●	●		●	●	●		●			
300 Series[3]:	●	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●				
420C:	●	●	●	●	●		●	●	●	●	●[1]		●	●	●[1]	●		●	●	●		●			
430i:	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				
440i:	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				
460i:	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				
790i:	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				
4000 Series[4]:	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				
7000 Series[5]:	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				
AlphaEclipse 1500 Time & Temp[6]:																									
AlphaEclipse 2500:	●	●	●	●	●		●	●	●[8]	●	●		●	●	●	●	●	●	●	●	●	●			
AlphaEclipse 2600:	●	●	●	●	●		●	●	●[8]	●	●		●	●	●	●	●	●	●	●	●	●			
AlphaEclipse 3500:	●	●	●	●	●		●	●	●[8]	●	●		●	●	●	●	●	●	●	●	●	●			
AlphaEclipse 3600[7]:	●	●	●	●	●		●	●	●[8]	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaPremiere:	●	●	●	●	●		●	●		●	●	●	●	●	●	●	●	●	●	●	●	●			
AlphaVision (full matrix):	●	●	●	●	●	●		●		●			●	●					●	●	●				
AlphaVision (character matrix):	●	●	●			●														●	●				
BetaBrite:	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●		●			
Big Dot:	●	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●				
Director:	●	●	●																	●	●				
PPD:	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				
Serial LED clock[6]:																									
Solar	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●				

NOTES:

[1] If the *Slide* mode is selected for either the 220C or 420C sign, the *Cycle Color* mode will be used instead. The same applies to the *Spray* mode for the 420C sign only ("C" = tricolor LEDs).

[2] This includes the 215R and 215C model signs ("C" = tricolor LEDs, "R" = red LEDs).

[3] This includes the 320C and 330C model signs ("C" = tricolor LEDs, "R" = red LEDs).

[4] This includes the 4080C, 4120C, 4120R, 4160C, 4160R, 4200C, 4200R, 4240C, and 4240R model signs ("C" = tricolor LEDs, "R" = red LEDs).

[5] This includes the 7080C, 7120C, 7160C, and 7200C model signs ("C" = tricolor LEDs, "R" = red LEDs).

[6] This sign can only display time updates from messaging software. This sign cannot display text messages or graphics.

[7] This sign has RGB (red, green, and blue) LEDs that are capable of displaying over 16 million colors.

[8] 7-high character set only.

7.9.5 Fonts and colors available on signs

Table 68: Fonts and colors available on signs

Signs	Characters													
	15/16 Row Normal	15/16 Row Fancy	Ten Row	Seven Row Normal	Seven Row Fancy	Five Row	Color [1]	Normal	Wide	Double Wide	Flashing	Double Height	True Descenders	Fixed Width
200 Series[2]:				●	●	●	●	●	●	●				●
220C:				●	●	●	●	●	●	●	●			●
300 Series[3]:				●	●	●	●	●	●	●	●			●
420C:				●	●	●	●	●	●	●	●			●
430i:				●		●		●	●	●				●
440i:				●		●		●	●	●				●
460i:				●		●		●	●	●				●
790i:				●		●		●	●	●				●
4000 Series[4]:	●	●		●	●	●	●	●	●	●	●			●
7000 Series[5]:	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaEclipse 1500 Time & Temp[6]:														
AlphaEclipse 2500:	●	●	●	●	●	●		●	●	●	●	●	●	●
AlphaEclipse 2600:	●	●	●	●	●	●		●	●	●	●	●	●	●
AlphaEclipse 3500:	●	●	●	●	●	●		●	●	●	●	●	●	●
AlphaEclipse 3600[7]:	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaPremiere:	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaVision (full matrix):	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaVision (character matrix):				●		●	●	●			●			
BetaBrite:				●	●	●	●	●	●	●	●			●
Big Dot:				●	●	●	●	●	●	●	●			●
Director:				●		●	●	●			●			
PPD:				●	●	●		●	●	●	●			●
Serial LED clock[6]:														
Solar:	●	●		●	●	●	●	●	●	●	●			●

[1] Sign models ending in “C”, such as 4120C, have color capabilities. Sign names ending in “R”, such as 4120R, can display in red only.
[2] This includes the 215R and 215C model signs (“C” = tricolor LEDs, “R” = red LEDs).
[3] This includes the 320C and 330C model signs (“C” = tricolor LEDs, “R” = red LEDs).
[4] This includes the 4080C, 4120C, 4120R, 4160C, 4160R, 4200C, 4200R, 4240C, and 4240R model signs (“C” = tricolor LEDs, “R” = red LEDs).
[5] This includes the 7080C, 7120C, 7160C, and 7200C model signs (“C” = tricolor LEDs, “R” = red LEDs).
[6] This sign can only display time updates from messaging software. This sign cannot display text messages or graphics.
[7] This sign has RGB (red, green, and blue) LEDs that are capable of displaying over 16 million colors.

7.9.6 Display options available on signs

Table 69: Display options available on signs

Signs	Options															
	Time	Date	Temperature		Speed	New Line	New Page	Animation	String	Ticker Symbol	Variable	Counter	Graphic [1]	Gif [1]	Flick [1]	Message
			Fahrenheit	Celsius												
200 Series[2]:	●	●			●	●		●	●	●	●	●	●			●
220C:	●	●			●	●		●	●	●	●	●	●			●
300 Series[3]:	●	●			●	●		●	●	●	●	●	●			●
420C:	●	●			●	●		●	●	●	●	●	●			●
430i:	●		●	●	●	●		●	●	●	●	●	●			●
440i:	●		●	●	●	●		●	●	●	●	●	●			●
460i:	●		●	●	●	●		●	●	●	●	●	●			●
790i:	●		●	●	●	●		●	●	●	●	●	●			●
4000 Series[4]:	●	●			●	●		●	●	●	●	●	●			●
7000 Series[5]:	●	●			●	●	●		●	●	●	●	●	●	●	●
AlphaEclipse 1500 Time & Temp[6]:																
AlphaEclipse 2500:	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaEclipse 2600:	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaEclipse 3500:	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaEclipse 3600[7]:	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●
AlphaPremiere:	●	●			●	●	●	●	●	●	●	●	●	●	●	●
AlphaVision (full matrix):	●	●			●	●	●		●	●	●	●	●	●	●	●
AlphaVision (character matrix):	●	●			●	●	●		●	●	●	●				●
Big Dot:	●	●			●	●		●	●	●	●	●	●			●
BetaBrite:	●	●			●	●		●	●	●	●		●			●
Director:	●	●			●	●			●	●	●	●				●
PPD:	●	●			●	●		●	●	●	●		●			●
Serial LED clock[6]:																
Solar:	●	●	●	●	●	●		●	●	●	●	●	●			●

[1] A graphic, gif, or flick must be designed for the resolution of the sign. For example, a 4120C sign has a resolution of 120 columns by 16 rows. Therefore, in order to fit on a 4120C, an image can be no greater than 120 x 16 pixels in size.
[2] This includes the 215R and 215C model signs ("C" = tricolor LEDs, "R" = red LEDs).
[3] This includes the 320C and 330C model signs ("C" = tricolor LEDs, "R" = red LEDs).
[4] This includes the 4080C, 4120C, 4120R, 4160C, 4160R, 4200C, 4200R, 4240C, and 4240R model signs ("C" = tricolor LEDs, "R" = red LEDs).
[5] This includes the 7080C, 7120C, 7160C, and 7200C model signs ("C" = tricolor LEDs, "R" = red LEDs).
[6] This sign can only display time updates from messaging software. This sign cannot display text messages or graphics.
[7] This sign has RGB (red, green, and blue) LEDs that are capable of displaying over 16 million colors.

7.10 Appendix J: Position rules for signs

Position rules deal with *where* text will appear on a sign.

7.10.1 Sign classes

- One-line signs — like the Betabrite, 220C, and 300 series are of varying lengths, but are always 7 dots (or pixels) high.
- Two-line signs — like the 4000 series are of varying lengths, but are always 16 dots high.
- Three-line signs (like the 7000 series) and Multiple-line full matrix signs (like the Director) are of varying lengths and heights.
- Multiple-line character matrix signs — like certain AlphaVision models are of varying lengths and widths.

7.10.2 Position classes

- Top
- Bottom
- Middle
- Fill

7.10.3 Position rule examples

7.10.3.1 One-line sign example

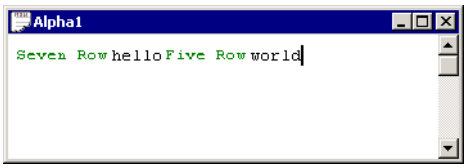
RULE:

All characters line up at the bottom of the sign and work their way up for as many dots as the font supports:

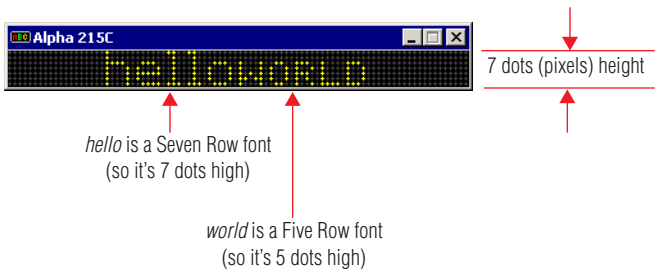
NOTE

The screen shots used in these examples are taken from the Emulator program that is part of the AlphaNET sign messaging software.

This is how a message is created in AlphaNET software:



This is how the message would appear on a one-line sign:



EXCEPTION CONDITIONS:

- If a sign receives a font that is larger than the sign can display, then the sign will "size down" or reduce the font size. For example, on a one-line sign, 15 high fancy characters would be replaced by 7 high fancy characters.
- If a graphic is received that is taller than what a one-line sign can display, then only the top 7 rows will be displayed.
- If a graphic is received that is longer than what a one-line sign can display, then only the leftmost columns will be displayed.

- If a graphic is received that is smaller than 7 dots high, then the graphic will be displayed from the bottom of the sign working up.
- If a character font is not specified, then 7-high normal will be used.
- If Top, Bottom, or Fill positions are received Middle is used.

7.10.3.2 Two-line sign example

7.10.3.2.1 Top position

RULE:

Defined as the top 7 dots of the sign. The Top position functions in the same manner as a one-line sign (see exception conditions for a one-line sign).

7.10.3.2.2 Bottom position

RULE:

Defined as the bottom 7 dots of the sign. The Bottom position functions in the same manner as a one-line sign (see exception conditions for a one-line sign).

7.10.3.2.3 Middle position

RULE:

The Middle position is treated as though it was 1 line sign 16 dots high. Each line of text presented on this line is prescanned to determine the largest piece of text (or graphic) to be displayed. For example, is a line of 5-high text has just a single 10-high character, the line is viewed as a 10-high line. This means that 10-high characters will be displayed with 3 dots above and below the characters (3+10+3 = 16).

EXCEPTION CONDITIONS:

- If the sign receives a font that is larger than the sign can display, then the sign will "size down" or reduce the font size. On a two-line sign, the only characters that are too large would be characters using the "double high" control code. In this case, the control code would be ignored.
- If a graphic is received that is taller than what a two-line sign can display, then only the top 16 rows will be displayed.
- If a graphic is received that is longer than what a two-line sign can display, then only the leftmost columns will be displayed.
- If a character font is not specified, then 16-high normal will be used.

7.10.3.2.4 Fill position

RULE:

On a two-line sign, the Fill position indicates that you wish to use no more than 7-high characters and that you wish to fit as much text on the screen as you can. When using the Fill position, the sign sees itself as having two lines of 7-high characters, and no means of displaying characters larger than 7-high. If a graphic is selected, then at most 7 rows of that graphic will be displayed. Also, if the last piece of a message is just one line, then the sign will center this line on the screen.

If the sign is operating on the *top* row, then the bottom of that row is assumed to be the 7th row of dots. All text is started from there and

worked up: 5-high characters will use rows 3 to 7 and 7-high characters will use rows 1 to 7.

If the sign is operating on the *bottom* row, then the sign works its way up from row 16: 5-high characters will use rows 12 to 16 and 7-high characters will use rows 10 to 16.

EXCEPTION CONDITIONS:

- If, when using the Top, Bottom, or Fill position, a sign receives a font that is larger than 7-high, then the sign will "size down" or reduce the font size. For example, 15 high fancy characters would be replaced by 7 high fancy characters.
- If a graphic is received that is taller than 7 rows high (15 high for Middle position), then only the top 7 (top 15 for Middle position) rows will be displayed.
- If a graphic is received that is longer than what a one-line sign can display, then only the leftmost columns will be displayed.
- If a character font is not specified, then 7-high normal will be used.

7.10.3.3 Three-line sign example

7.10.3.3.1 Top/Bottom positions

RULE:

The Top and Bottom positions work in tandem with each other. There is an imaginary line between the top and bottom half of the sign. This is called the "centerline". The centerline divides what is used for the Top from what is used for the Bottom positions (see example below).

The location of the centerline is usually established by the first Top command the sign receives, and the rest of the space is used for the Bottom position. If a Bottom position command comes first, then the centerline is placed at its highest position — row 8, allowing for a single line of 7-high characters on the Top position.

Once a centerline has been established, it remains fixed until a Fill or Middle position command is received. The centerline can not be changed with another Top or Bottom position command.

However, if the first command specifies a Top, and not a Bottom, position, then the centerline's position is determined by the amount of text following the position command. For example,

- If one 7-high line of text is received (following a Top position command), then the centerline will be fixed at row 8.
- If one line of 10-high characters is received (following a Top position command), then the centerline will be fixed at row 11.
- If two lines of 5-high characters are received (following a Top position command), then the centerline is placed at row 12 (5 rows for each line of text plus a blank row between the lines).

EXCEPTION CONDITIONS:

- The centerline is never placed higher than 8 rows from the top of the sign.
- The centerline is never placed lower than 8 rows from the bottom of the sign.

7.10.3.3.2 Middle position

RULE:

The Middle position is treated as though it were a one-line sign with as many rows as the sign is tall. Each line of text on the sign is prescanned to determine the largest piece of text (or graphic) to be displayed. The line of text is then vertically centered based on that largest piece of text or graphic. For example, if you have a line of text which has mostly 5-high characters, but has one 10-high character, then this line is considered a 10-high line. Assuming that this is a 24-row sign, this would leave 14 extra rows so there would be 7 blank rows on top and 7 on the bottom (7+10+7=24). All text and graphics are then lined up on this new virtual bottom (the 21st line) and treated the same as in a one-line sign.

EXCEPTION CONDITIONS:

- If a graphic is received that is taller than what the sign can display, then only the top most rows will be displayed.
- If a graphic is received that is longer than what a sign can display, then only the leftmost columns will be displayed.
- If a character font is not specified, then 7-high normal will be used.

7.10.3.3.3 Fill position

RULE:

On a 7000 series or an AlphaVision sign, the Fill position indicates that you wish to fit as much text on the screen as you can. Unlike the 4000 series signs, in the Fill position you can select characters larger than 7-high.

The sign will start from top of the screen working down. If you select a 15-high character set, then the sign will fit as many 15 row lines of text on the screen as possible. As soon as the sign detects that the next

line will not fit, the sign will stop creating the current page and display it. The next page will begin with the line the did not fit. If the text does not use up the entire display, then the sign will center the text vertically, splitting the blank space between the top and the bottom.

EXCEPTION CONDITIONS:

- If a graphic is received that is taller than 7 rows high, then only the top 7 rows will be displayed.
- If a graphic is received that is longer than what the sign can display, then only the leftmost columns will be displayed.
- If a graphic is received that is smaller than 7 dots high, the graphic will be displayed from the bottom of the sign working up.
- If a character font is not specified, then 7-high normal will be used.

7.10.3.4 Multiple-line character matrix sign example

The sign will work exactly like the three-line full matrix signs (described in the previous section) with the following exceptions:

- If a mode other than Wipe is received, it is replace with the Hold mode.
- The sign will ignore all the following:
  - graphics
  - all character set commands, except 5- and 7-high normal
  - wide
  - double wide
  - double high
  - true descenders
  - proportional spacing
  - animations
- If a character font is not specified, then 7-high normal will be used.

7.11 Appendix K: Alpha 2.0 protocol additions

NOTE: As of the writing of this protocol manual, the Alpha 2.0 protocol is only available for the AlphaPremiere and AlphaEclipse signs.

The Alpha 2.0 protocol adds the following functions to the existing Alpha 1.0 protocol:

Table 70: Alpha 2.0 protocol additions

Function	Type	Description	Reference
Speed control	Control code ^0 0FH	Sets the amount of time to hold the current page and all subsequent pages.	"Speed control" on page 98
Sound control	Option "3" 33H for Write SPECIAL FUNCTION "(" 28H	Allows the creation and playing of multi-note sounds	"Sound control (AlphaPremiere 9000 only)" on page 99
Set Run File Times	Write/Read SPECIAL FUNCTION ":" 3AH	Allows setting/reading a start and end run time for a file configured with a standard time of NEVER.	"Set Run File Time" on page 100
Custom character sets	Write SPECIAL FUNCTION "<" 3CH	Programs up to four custom character sets.	"Custom character sets" on page 104
	Control codes: • 1AH "W" (Five-high custom character set) • 1AH + "X" (Seven/Eight-high custom character set) • 1AH + "Y" (Ten-high custom character set) • 1AH + "Z" (Fifteen/Sixteen-high custom character set)	Select s a custom character set.	"Control codes (00 – 1FH)" on page 80
Custom Automode table	Write/Read SPECIAL FUNCTION ">" 3EH	Creates a custom Automode table with up to 15 modes.	"Automode table" on page 107
Set timeout message	Command Code "T" 54H	Allows setting a timeout period after which a custom message will appear.	Table 6, "Command Codes," on page 11
Read/Set Dimming Control Register	Read SPECIAL FUNCTION "@" 3FH Write SPECIAL FUNCTION "@" 3FH	Allows enabling/disabling a sign's light sensor and setting the brightness level a sign dims to.	"Dimming Control Register" on page 108
ACK/NAK response	Write SPECIAL FUNCTION "s" 73H	Allows enabling/disabling of an ACK/NAK response after every <EOT>.	"Enable/Disable ACK/NAK response" on page 111
Read temperature log	Read SPECIAL FUNCTION "L" 4CH	Reads a sign's temperature log	"Temperature Logging" on page 112
Read external temperature	Read SPECIAL FUNCTION "T" 54H	Reads the external temperature of a sign equipped with a functioning temperature probe.	"Read External Temperature command" on page 115
Read internal temperature	Read SPECIAL FUNCTION "TI" 54H 49H	Reads the internal temperature of a sign.	"Read Internal Temperature command" on page 116
Set Unit commands	Write/Read SPECIAL FUNCTION "U1", "U2", "U3", "U4", "U5", "U6", and "UN"	A series of commands that allows setting and reading sign parameters such as serial address.	"Set Unit commands" on page 118

7.11.1 Speed control

This control code (**Table 6** on page 11) sets the amount of time to hold the current page and all subsequent pages. For compatibility with some older AlphaVision signs, Speed control has two modes:

- Seconds mode
- Tenths-of-seconds mode

7.11.1.1 Seconds mode

Table 71: Speed control seconds mode syntax

Syntax:	CXX where: C = ^O (0FH) XX = two ASCII hexadecimal numbers that represent the numbers of seconds to hold, ranging from "00" to "FF" (255) seconds
Example:	^O"1A" means: hold text for 26 (1AH) seconds

7.11.1.2 Tenths-of-seconds mode

Table 72: Speed control tenths-of-seconds mode syntax

Syntax:	CIXXX where: C = ^O (0FH) I = "T" (54H) an indicator to switch to tenths-of-second mode XXX = three ASCII hexadecimal numbers that represent the number of tenths-of-seconds to hold
Example:	^O"T258" means: hold text for 1 minute (258H = 600 x 0.1 sec = 60 seconds)

7.11.2 Sound control (AlphaPremiere 9000 only)

There are two new options for the Write SPECIAL FUNCTION Command Code Generate Speaker Tone ("(" 28H), see page 22:

- Store a programmable sound
- Trigger a programmable sound

NOTE: A Clear Memory Write SPECIAL FUNCTION command ("$" 24H) will delete all sound files.

7.11.2.1 Store a programmable sound

Table 73: Store a programmable sound syntax

Syntax:	CLONAVRDP where: this section repeats for each note C = "3" (33H) follows the Generate Speaker Tone SPECIAL FUNCTION label: "(" 28H (see page 22). L = one ASCII hexadecimal character that represents the sound file label. Valid characters are 20H through 2FH which allows up to 16 sounds files. O = one ASCII hexadecimal character that represents the octave. Valid entries are "0" through "7". N = one ASCII hexadecimal character that represents the musical note. Valid entries are "A" through "G". Each sound file can have up to 32 notes. A = one ASCII hexadecimal character that represents the accidental. Valid entries are: "N" for Natural, "S" for sharp, and "F" for flat. (*Currently only Naturals are implemented.*) V = one ASCII hexadecimal character that represents the sound volume. Valid entries are "0" through "F". R = one ASCII hexadecimal character that represents the number of times to repeat the musical note. Valid entries are from "0" through "F". D = one ASCII hexadecimal character that represents the musical note's on duration in 0.1 second increments. Valid entries are from "0" through "F" where "0" = turn off the sound file and "F" = musical note will stay on until another trigger. P = one ASCII hexadecimal character that represents the pause or off time duration in 0.1 second increments. Valid entries are from "0" through "F".

Table 73: Store a programmable sound syntax

Example:	"3$4CNF1524ENF1524GNF152" means: sound file label = "$" octave = "4" note = "C" accidental = "N" ("N" = Natural) volume = "F" (15 = maximum) repeat note = "1" (once) duration of the note = "5" (0.5 sec = 5 x 0.1) pause time before next note = "2" (0.2 sec = 2 x 0.1) octave = "4" note = "E" accidental = "N" ("N" = Natural) volume = "F" (15 = maximum) repeat note = "1" (once) duration of the note = "5" (0.5 sec = 5 x 0.1) pause time before next note = "2" (0.2 sec = 2 x 0.1) octave = "4" note = "G" accidental = "N" ("N" = Natural) volume = "F" (15 = maximum) repeat note = "1" (once) duration of the note = "5" (0.5 sec = 5 x 0.1) pause time before next note = "2" (0.2 sec = 2 x 0.1)

7.11.2.2 Trigger a programmable sound

If a sound file is currently running and a new sound file trigger occurs, then the new sound file trigger will immediately replace an old sound file.

Table 74: Trigger a programmable sound syntax

Syntax:	CL where: C = "4" (34H) follows the Generate Speaker Tone SPECIAL FUNCTION label: "(" 28H (see page 22). L = one ASCII hexadecimal character that represents the sound file label to be triggered. Valid characters are 20H through 2FH.
Example:	"49" means: play sound file "9"

7.11.3 Set Run File Time

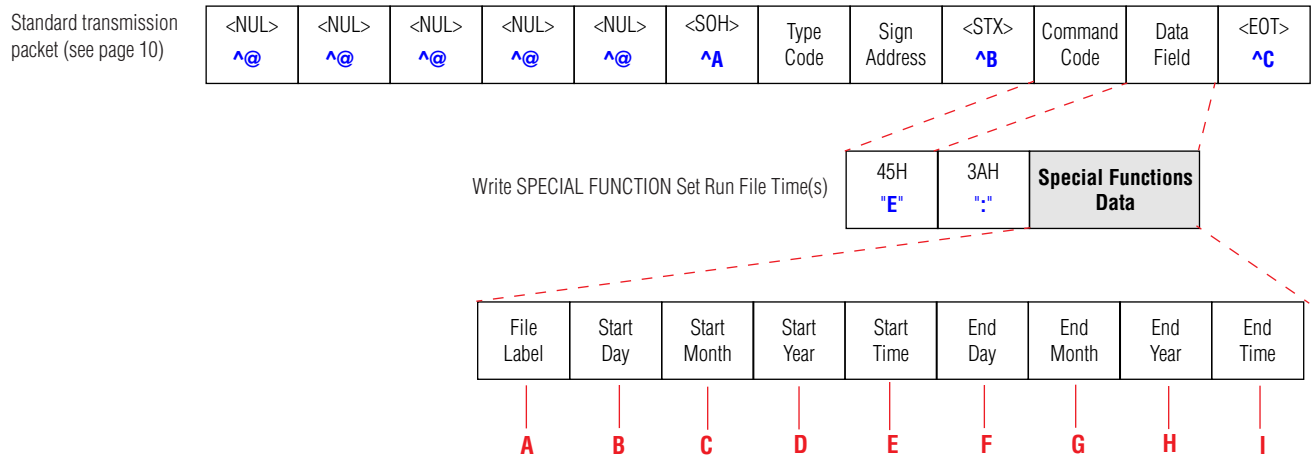
The Set Run File Time SPECIAL FUNCTION allows setting a start and end run time for a file configured with a standard run time of NEVER. That is, if the file can not run for another reason, the sign will check to see if there is a valid Run File Time for the file. If a valid file exists and the sign's current time is within the specified start and stop period, the file will run.

In determining the start and end time window criteria, a run time period begins when the minute reaches the start time. A run time period ends when it reaches the end time. (If start time = end time, then the file will not run.)

Multiple start and end times per file are acceptable. The total number (combined for all files) of start and end times that can be stored is 100.

All start and end times are erased with the Clear Memory (E$) Set Memory Configuration Write SPECIAL FUNCTION command (page 21).

Table 75: Set Run File Time(s) packet format



Item	Name	Description
A	File Label	One ASCII character that represents a valid file name.
B	Start Day	Start day represented by two ASCII decimal digits. Valid entries range from “01” (30H)(31H) through “31” (33H)(31H), depending on the month.
C	Start Month	Start month represented by two ASCII decimal digits. Valid entries range from “01” (30H)(31H) through “12” (31H)(32H).
D	Start Year	Start year represented by four ASCII decimal digits. Valid entries range from “0000” (30H)(30H)(30H)(30H) through “9999” (39H)(39H)(39H)(39H).
E	Start Time	Start time in 24-hour format represented by four ASCII decimal digits. Valid entries range from “0000” (30H)(30H)(30H)(30H) through “2359” (32H)(33H)(35H)(39H).
F	End Day	End day represented by two ASCII decimal digits. Valid entries range from “01” (30H)(31H) through “31” (33H)(31H), depending on the month.
G	End Month	End month represented by two ASCII decimal digits. Valid entries range from “01” (30H)(31H) through “12” (31H)(32H).
H	End Year	End year represented by four ASCII decimal digits. Valid entries range from “0000” (30H)(30H)(30H)(30H) through “9999” (39H)(39H)(39H)(39H).
I	End Time	End time in 24-hour format represented by four ASCII decimal digits. Valid entries range from “0000” (30H)(30H)(30H)(30H) through “2359” (32H)(33H)(35H)(39H).

7.11.3.1 Removing Run File Times

All Run File entries must be removed for a given file at once. To remove all Run File entries, specify the File Label as a Priority TEXT file (“0” 30H).

In the instance where it is *not* preferable to remove all run entries for a given file, use the following procedure:

- Read all the Run Time entries for the file
- Remove these times (as far as the sign is concerned)
- Rewrite the desired ones to the sign

To delete all start and end times for a file, use the Set Run Time syntax (**Table 75** on page 101), except set all parameters to “9”. For example, to delete all Run Time entries for file “D” use: “D99999999999999999999999”.

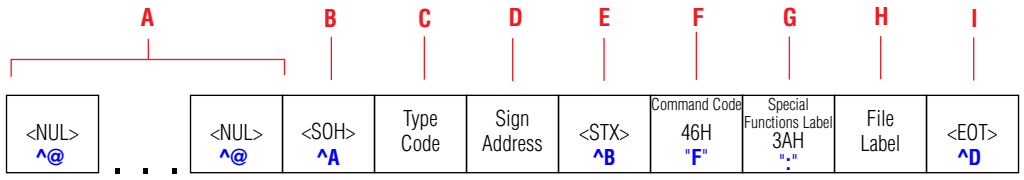
7.11.3.2 Reading Run File Time

The start and end time data can be read back from a sign. Additional information is returned as well, such as the total number of start and end entries

for all files as well as statuses.

This is the message format for retrieving start and end entries:

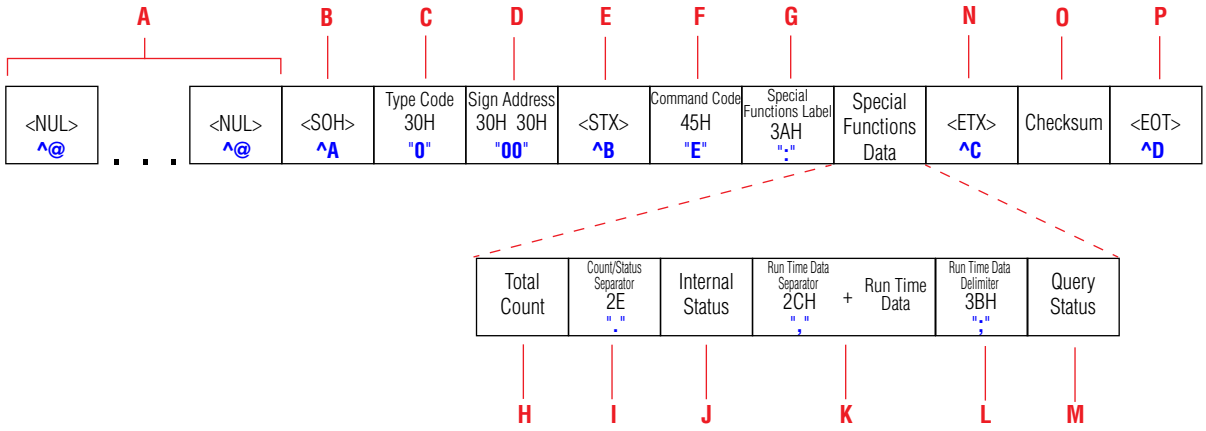
Table 76: Read Run File Time(s) file transmission packet format



Item	Name	Description
A	<NUL>	Twenty <NUL> (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	See Table 4, “Standard transmission packet (“1-byte” or “^A”) format,” on page 10.
D	Sign Address	See Table 4, “Standard transmission packet (“1-byte” or “^A”) format,” on page 10.
E	<STX>	<STX> (02H) character
F	Command Code	“F” (46H) Read SPECIAL FUNCTION Command Code.
G	Special Functions Label	“:” (3AH) Read Run File Times code
H	File Label	The Run File to read. Use “0” (30H) to read all files.
I	<EOT>	<EOT> (04H) character

The data from the sign is returned in the following format:

Table 77: Read Run File Time file response packet format



Item	Name	Description
A	<NUL>	Twenty <NUL> (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	“0” (30H) is the Response code
D	Sign Address	“00” (30H + 30H) is sent regardless of the sign’s actual address.
E	<STX>	<STX> (02H) character
F	Command Code	“E” (45H) is returned by the sign. (The Read SPECIAL FUNCTIONs Command Code.)
G	Special Functions Label	“:” (3AH) is returned by the sign. (The Run File Time code)

Table 77: Read Run File Time file response packet format

H	Special Function Data	Total Count	Two ASCII hexadecimal digits that represent the *total* number of run times entries for *all* files.
I		Count/Status Separator	"." (2EH) is used to separate Total Count from Internal Status.
J		Internal Status	Two ASCII hexadecimal digits that represent the current internal entry table status. Status values are: • "00" = OKAY — no problem • "01" = NOROOM — out of storage • "02" = BADFILE — file not in configuration, no such file • "03" = BADDATA — data (time/date) invalid • "04" = INCOMPLETE — error during transfer of new data • "05" = LOCKED — attempted to access a locked file • "09" = NOTFOUND — attempted to delete/retrieve entries for a file that isn't in the table
K		Run Time Data Separator + Run Time Data	More than one Run Time Data entry can be returned. Each Run Time Data entry will be returned in this format: SFDDMMYYYYTTTTEENNZZZZUUUU where: S = "." (2EH) Run Time Data separator F = File Label DD = Start day represented by two ASCII decimal digits. Valid entries range from "01" (30H)(31H) through "31" (33H)(31H), depending on the month. MM = Start month represented by two ASCII decimal digits. Valid entries range from "01" (30H)(31H) through "12" (31H)(32H). YYYY = Start year represented by four ASCII decimal digits. Valid entries range from "0000" (30H)(30H)(30H)(30H) through "9999" (39H)(39H)(39H)(39H). TTTT = Start time in 24-hour format represented by four ASCII decimal digits. Valid entries range from "0000" (30H)(30H)(30H)(30H) through "2359" (32H)(33H)(35H)(39H). EE = End day represented by two ASCII decimal digits. Valid entries range from "01" (30H)(31H) through "31" (33H)(31H), depending on the month. NN = End month represented by two ASCII decimal digits. Valid entries range from "01" (30H)(31H) through "12" (31H)(32H). ZZZZ = End year represented by four ASCII decimal digits. Valid entries range from "0000" (30H)(30H)(30H)(30H) through "9999" (39H)(39H)(39H)(39H). UUUU = End time in 24-hour format represented by four ASCII decimal digits. Valid entries range from "0000" (30H)(30H)(30H)(30H) through "2359" (32H)(33H)(35H)(39H).
L		Run Time Data Delimiter	";" (3BH) is used to indicate the end of Run Time Data.
M		Query Status	Two ASCII hexadecimal digits that represent the status of this entry table status. Status values are: • "00" = OKAY — no problem • "01" = NOROOM — out of storage • "02" = BADFILE — file not in configuration, no such file • "03" = BADDATA — data (time/date) invalid • "04" = INCOMPLETE — error during transfer of new data • "05" = LOCKED — attempted to access a locked file • "09" = NOTFOUND — attempted to delete/retrieve entries for a file that isn't in the table
N	<ETX>		<ETX> (03H) character
O	Checksum		Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
P	<EOT>		<EOT> (04H) character

7.11.4 Custom character sets

7.11.4.1 Custom character set memory requirements

Four custom character sets can be programmed. These sets will work just like the standard character sets. Character sets should allow for characters 20H to C1H. This is the full ASCII set minus the control codes.

Custom character sets take up memory (RAM) in a sign:

Table 78: Custom character set memory requirements

Font	Characters	Memory requirements (bytes)
5 high	20H - 60H (lowercase not used)	320
7 high	20H - C1H	1127
8 high	20H - C1H	1288 (AlphaEclipse™ 3500 1-line sign)
10 high	20H - C1H	1610
15 high	20H - C1H	2415 (AlphaPremiere 9000 signs)
16 high	20H - C1H	2576 (AlphaEclipse outdoor signs)
If all sets are used, then 9336 bytes are required.		

7.11.4.2 Custom character set identifiers

Custom character set identifiers (see the 1AH control code in "Appendix G: Alpha protocol ASCII table" on page 80):

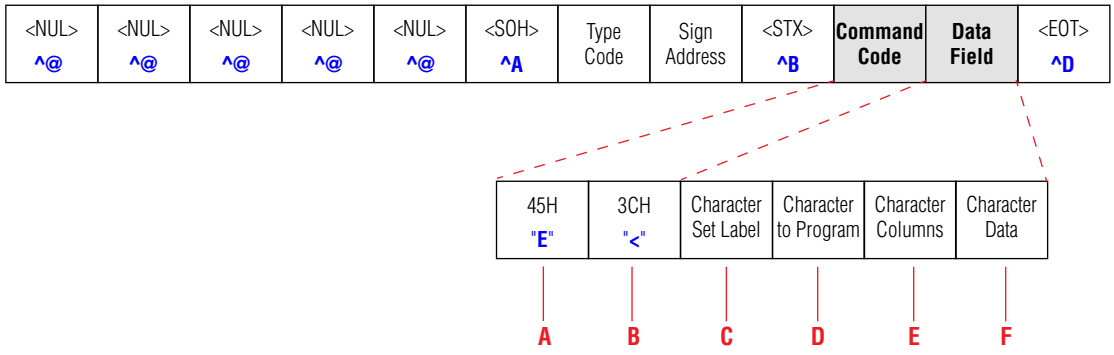
- 1AH + "W" = Five high custom character set
- 1AH + "X" = Seven/Eight high custom character set
- 1AH + "Y" = Ten high custom character set
- 1AH + "Z" = Fifteen/Sixteen high custom character set

7.11.4.3 Program Custom Character Sets

To create a custom character set, a new Write SPECIAL FUNCTION code ("<") is used:

Table 79: Program Custom Character Sets packet format

Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10):



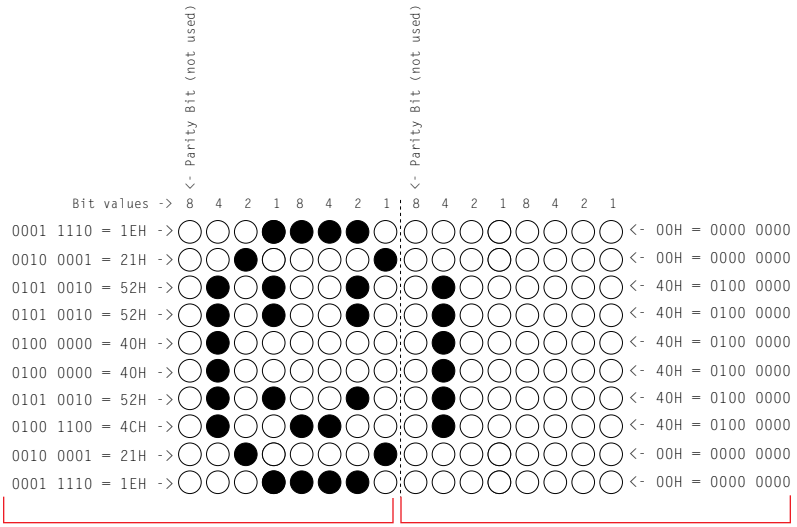
Item	Name	Description
A	Command Code	"E" (45H) = Write SPECIAL FUNCTION file
B	Special Functions Label	"<" (3CH) = Program Custom Character Set

Table 79: Program Custom Character Sets packet format

C	Character Set Label	One ASCII character. Valid entries are: • "W" (57H) = Five high custom character set • "X" (58H) = Seven/Eight high custom character set • "Y" (59H) = Ten high custom character set • "Z" (5AH) = Fifteen/Sixteen high custom character set
D	Character to Program	Two ASCII characters. Valid entries are: • "20" through "60" for Five high set • "20" through "C1" for all other sets NOTE: To clear a character set, send "00". For example, to clear the 10 high character set, send: ^AZ00^BE<Y00^D.
E	Character Columns	Two ASCII characters. Valid entries are: • Maximum of 6 for Five high and Seven/Eight high sets • Maximum of 8 for Ten high set • Maximum of 11 for Fifteen/Sixteen high set
F	Character Data	Two hexadecimal bytes for each character row, starting with the top of a character. Both bytes combine to form a bitmapped representation of a character row. Number of rows is dependent on the character set.

7.11.4.4 Program custom character example

This example shows how to create a single 10 high custom character — a Smily Face:



A custom character is transmitted by sending one of its rows at a time, starting from the top of the character. Each character row is defined by two bitmapped bytes. For example, 1EH 00H defines the first character row above. The 8th bit in both bytes is not used and is always 0.

Table 80: Program custom character (Smiley Face) example

<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"E<Y2008"1E00 2100 5240 5240 4040 4040 5240 4C40 2100 1E00<EOT>

A B C D E F G H I J K L

This is *hexadecimal*, not ASCII, data.
A space is used between each pairs of bytes for ease of reading.

Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	"Z" (5AH) means that this transmission is directed to all signs.
D	Sign Address	"00"	"00" (30H 30H) means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character

Table 80: Program custom character (Smiley Face) example

F	Command Code		"E"	"E" (45H) is the "Write SPECIAL FUNCTION example" on page 68.
G	Data Field	Special Functions Label	"<"	"<" (3CH) Program Custom Character Set command
H		Character Set Label	"Y"	"Y" (59H) 10-high custom character set
I		Character to Program	"20"	This is normally the ASCII space character.
J		Character Columns	"08"	The maximum number of columns for the 10-high character set = 8.
K		Character Data	1EH 00H	= (00011110 00000000) bitmapped representation of character row 1 (top)
			21H 00H	= (00100001 00000000) bitmapped representation of character row 2
			52H 40H	= (01010010 01000000) bitmapped representation of character row 3
			52H 40H	= (01010010 01000000) bitmapped representation of character row 4
			40H 40H	= (01000000 01000000) bitmapped representation of character row 5
			40H 40H	= (01000000 01000000) bitmapped representation of character row 6
			52H 40H	= (01010010 01000000) bitmapped representation of character row 7
			4CH 40H	= (01001100 01000000) bitmapped representation of character row 8
			21H 00H	= (00100001 00000000) bitmapped representation of character row 9
			1EH 00H	= (00011110 00000000) hexadecimal bitmapped representation of character row 10 (bottom)
L	<EOT>		04H	End Of Transmission character

7.11.5 Automode table

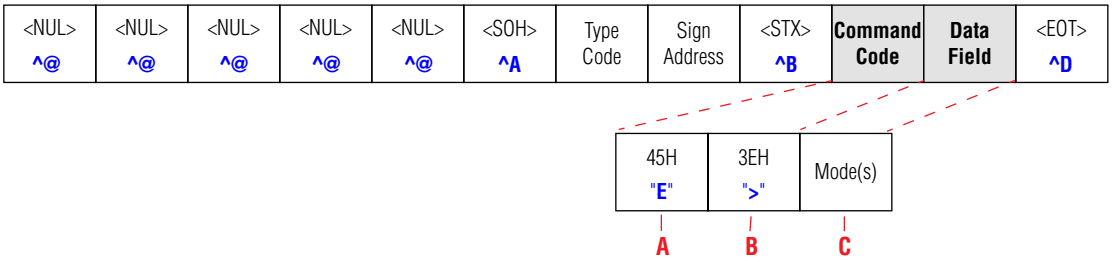
This SPECIAL FUNCTION command (“>” 3EH) is used to create (or read) a custom Automode table.

When a message has no modes specified, then the modes in the Automode table will be used to display the message. If the Automode table is cleared or not programmed, then the default Automode table modes are used.

7.11.5.1 Set Automode table command packet format

Table 81: Set Automode table command packet format

Standard transmission packet (see “Standard transmission packet (“1-byte” or “^A”) format” on page 10):

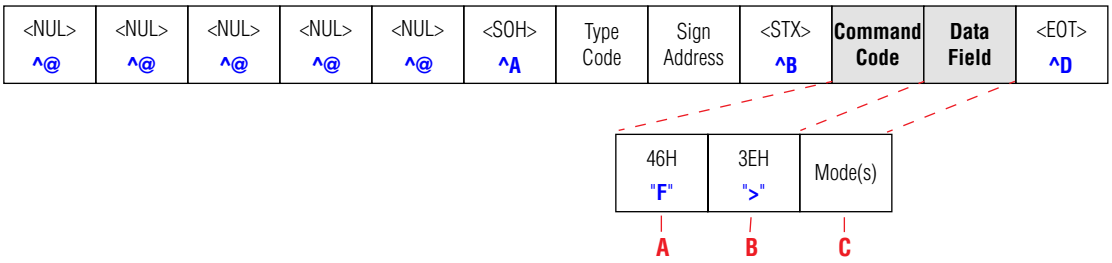


Item	Name	Description
A	Command Code	“E” (45H) = Write SPECIAL FUNCTION file
B	Special Functions Label	“>“ (3EH) = Program Custom Character Set
C	Mode(s)	From 2 – 30 ASCII characters in 2-byte pairs where each ASCII pair specifies a mode. This allows up to 15 modes to be programmed into the Automode table. If more than 15 modes are programmed, the command is ignored. If no modes are specified, then the table is cleared. Most Standard Modes (see Table 64, “Standard Modes,” on page 88) only require a single ASCII character — for example “g” (67H) for ROLL LEFT Mode. For these 1-byte Modes, the ASCII character “0” (30H) is added. Therefore, ROLL LEFT would be represented by the ASCII pair “g0”. On the other hand, the SPECIAL Standard Mode (see Table 64, “Standard Modes,” on page 88), requires two ASCII characters: “n” (6EH) plus another character which specifies a Special Mode (see **Table 65** on page 88) or a Special Graphic (see **Table 66** on page 89). For example, to program Rotate, Hold, Flash, and Slide modes in the Automode table, send: <NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”E>a0b0c0n5”<EOT>. To clear the Automode table, send: <NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”E>”<EOT>

7.11.5.2 Read Automode table command packet format

Table 82: Read Automode table command packet format

Standard transmission packet (see “Standard transmission packet (“1-byte” or “^A”) format” on page 10):



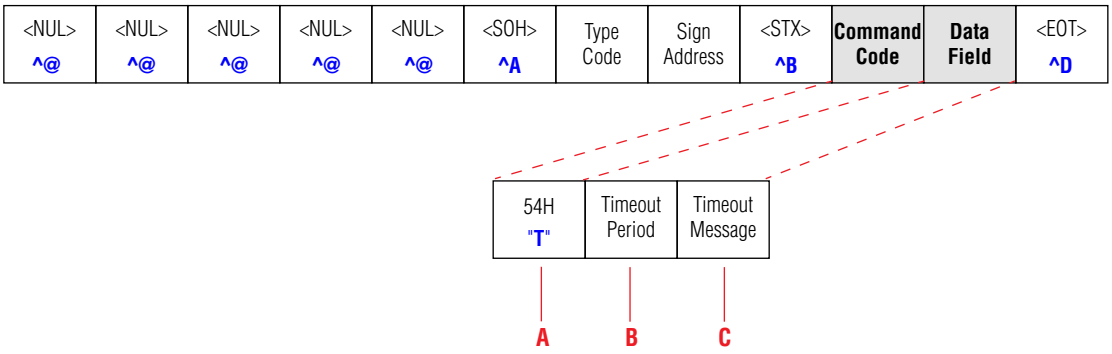
Item	Name	Description
A	Command Code	“F” (46H) = Read SPECIAL FUNCTION file
B	Special Functions Label	“>“ (3EH) = Program Custom Character Set
C	Mode(s)	From 2 – 30 ASCII characters in 2-byte pairs where each ASCII pair specifies a mode.

7.11.6 Set Timeout Message

This Command Code allows you to specify a timeout period after which a custom message will appear on the sign.

Table 83: Set Timeout Message syntax

Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10):



Item	Name	Description
A	Command Code	"T" (54H) = Set Timeout Message
B	Timeout Period	Three ASCII hexadecimal digits used to set the number of 1/10s of seconds in which if no serial transmission is received, then the Timeout Message will be displayed. Valid values range from: "000" to "FFF".
C	Timeout Message	ASCII character message

NOTE: The Clear Memory command will not delete a Timeout Message. To clear a Timeout Message, either (1) set the clear memory DIP switch and cycle power or (2) send a NULL message as follows: <SOH>"Z00"<STX>"T000"<EOT>.

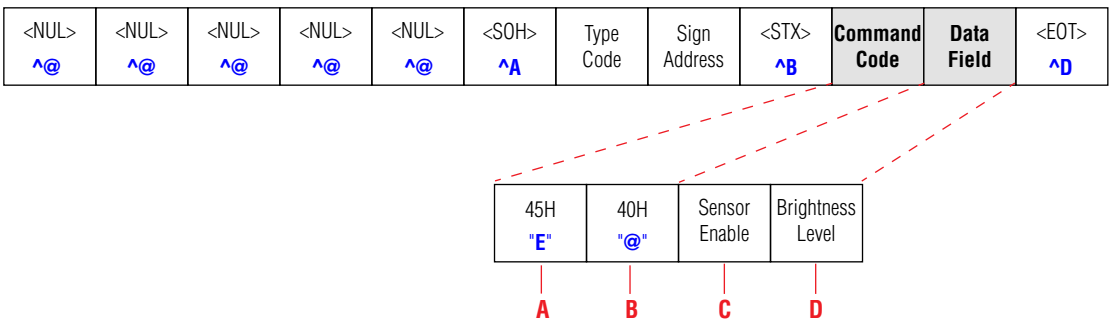
7.11.7 Dimming Control Register

The Dimming Control Register controls the brightness percentage when an AlphaEclipse sign is in dim mode. The register also enables or disables a sign's light sensor. Changing the brightness level in this register also alters the brightness level that the Set Dimming Register Write SPECIAL FUNCTION (page 23) dims to.

7.11.7.1 Set Dimming Control Register command packet format

Table 84: Set Dimming Control Register syntax

Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10):



Item	Name	Description
A	Command Code	"E" (45H) = Write SPECIAL FUNCTION file
B	Special Functions Label	"@" (40H) = Set Dimming Control Register

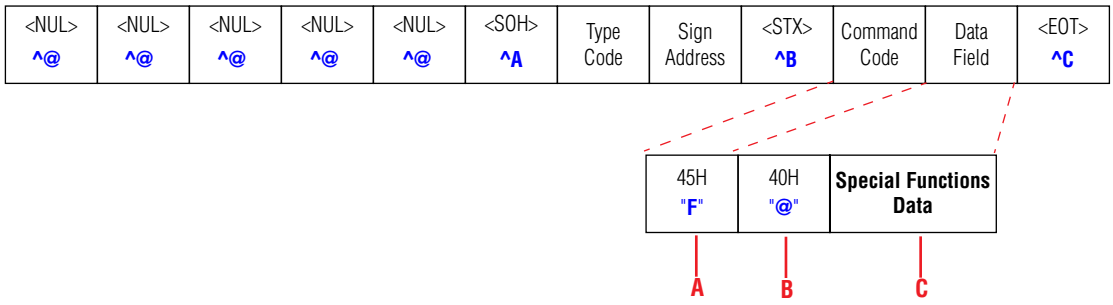
Table 84: Set Dimming Control Register syntax

C	Sensor Enable	One ASCII character. Valid entries are: "0" 30H = sign sensor OFF "1" 31H = sign sensor ON
D	Brightness Level	Two ASCII characters. Valid entries are: "00" through "12" = 12.5% of full brightness "13" through "25" = 25% of full brightness "26" through "37" = 37.5% of full brightness "38" through "50" = 50% of full brightness "51" through "62" = 62.5% of full brightness "63" through "75" = 75% of full brightness "76" through "87" = 87.5% of full brightness "88" through "99" = 100% of full brightness

7.11.7.2 Read Dimming Control Register command packet format

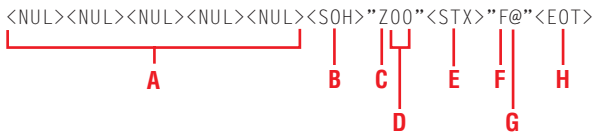
"@" — Sending "F@" will read the dimming percentage currently in this register, current brightness level, whether the photocell is enabled or disabled, and what is currently causing the display to dim. s

Table 85: Read Dimming Control Register command packet format



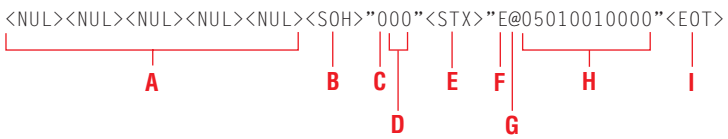
Special Functions Label		Special Functions Data Command Code "F" (46H) reads sign parameters.
A	Command Code	"F" (46H) = Read SPECIAL FUNCTION command
B	Special Functions Label	"@" (40H) = Read Dimming Control Register
C	Special Functions Data	This data is returned in the response packet (see below).

Table 86: Read Dimming Control Register command example 1



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that all signs should respond to this command.
D	Sign Address		"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>		02H	Start of TeXt character
F	Command Code		"F"	"F" (46H) is the "Read SPECIAL FUNCTION" Command Code. (See Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.)
G	Data Field	Special Functions Label	"@"	Read Dimming Control Register command.
H	<EOT>		04H	End Of Transmission character

Table 87: Read Dimming Control Register response example 2



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	"0" (30H) is the Response code.
D	Sign Address	"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	"E" (45H) is the response to a Read SPECIAL FUNCTION command.
G	Special Functions Label	"@"	Read Dimming Control Register command.

H	Data Field	"05010010000"	Eleven ASCII decimal characters in the following format: AAABBBCDEFG where:
			• AAA = a percentage that represents the current dimming control register setting. This setting is the percentage that the display will dim if the sign's photocell causes dimming. Valid values are from "000" to "100".
			• BBB = a percentage that represents the sign's current brightness level. If the sign is dimmed because of E, F, G, or the Set Dimming Time command (page 23) when this command is sent, the dimming level will be returned to the current brightness level. Valid values are from "000" to "100".
			• C = sign photocell enabled/disabled flag.
			  ◦ "0" (30H) = photocell disabled
			  ◦ "1" (31H) = photocell enabled
			• D = sign photocell dimming
			  ◦ "0" (30H) = sign photocell is not causing sign dimming
			  ◦ "1" (31H) = sign photocell is causing sign dimming
			• E = display load dimming
			  ◦ "0" (30H) = display load is not\ causing sign dimming
			  ◦ "1" (31H) = display load is causing sign dimming
			• F = internal or external display temperature dimming
			  ◦ "0" (30H) = display temperature is not causing sign dimming
			  ◦ "1" (31H) = display temperature is causing sign dimming
			• G = Set Dimming Time ("E/" page 23) dimming
			  ◦ "0" (30H) = Set Dimming Time command is not causing sign dimming
			  ◦ "1" (31H) = Set Dimming Time command is causing sign dimming

I	<EOT>	04H	End Of Transmission character

7.11.8 Enable/Disable ACK/NAK response

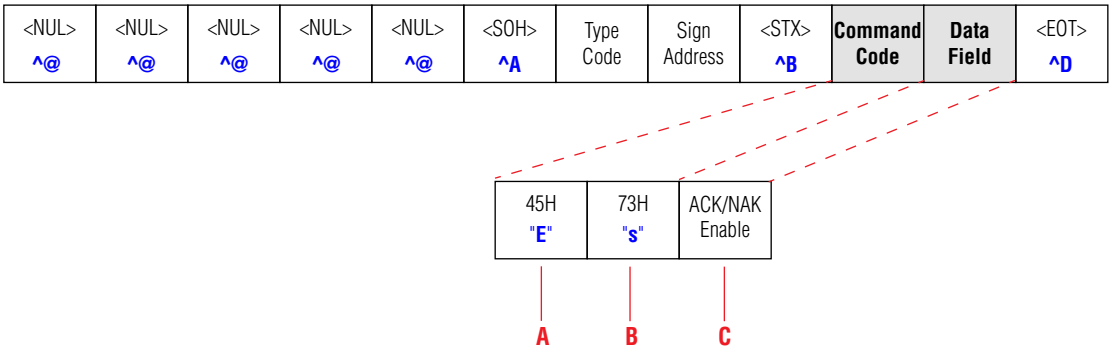
When the ACK/NAK response is enabled, a sign will respond with one of the following transmissions whenever an <EOT> occurs:

- <ACK>[Serial Error Status Register value] — response to a good serial transmission
- <NAK>[Serial Error Status Register value] — response to an incorrect serial transmission

NOTE: The Serial Error Status Register value is one ASCII character that represents the bitmapped value of the Serial Error Status Register (page 30).

Table 88: Enable/Disable ACK/NAK packet format

Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10):



Item	Name	Description
A	Command Code	"E" (45H) = Write SPECIAL FUNCTION file
B	Special Functions Label	"s" (73H) = Enable/Disable ACK/NAK response
C	ACK/NAK Enable	One ASCII character: "0" 30H = disable ACK/NAK sign response **(default)** "1" 31H = enable ACK/NAK sign response

7.11.9 Temperature Logging

After the temperature is read, it is compared to the previous read and the maximum and minimum temperatures are stored. The board and external temperatures (minimum and maximum) are recorded every 30 minutes over the past 24 hours.

Board temperature is in Celsius and external temperature is in Fahrenheit.

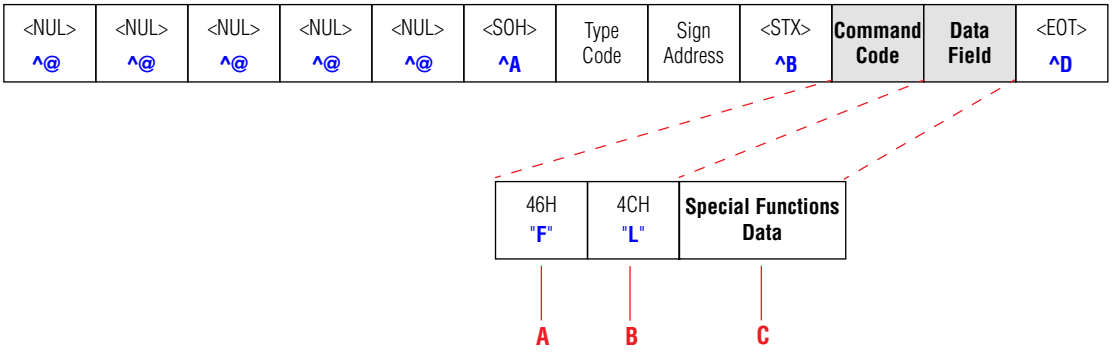
By ignoring the Alpha packet codes, you should be able to store the log as a text file.

NOTE: "–127" is returned by the firmware if there is no probe connected to the display, or when the probe connected is malfunctioning. Also, only simulating a virgin power up clears this log.

7.11.9.1 Read Temperature Log command packet format

Table 89: Read Temperature Log packet format

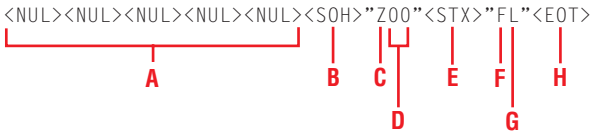
Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10):



Item	Name	Description
A	Command Code	"F" (46H) = Write SPECIAL FUNCTION file
B	Special Functions Label	"L" (4CH) = Read Temperature Log
C	Special Functions Data	This data is returned in the response packet (see "Read Temperature Log command example" on page 113).

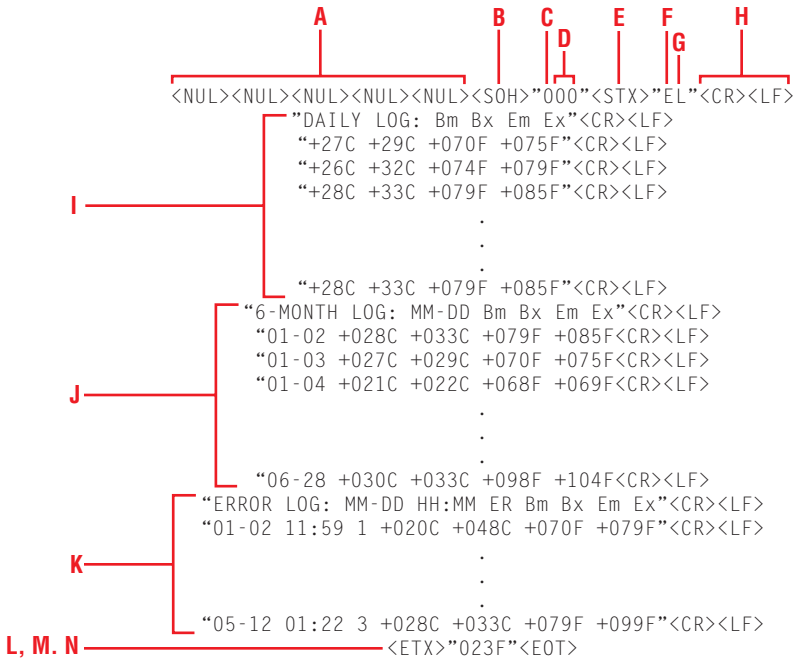
7.11.9.2 Read Temperature Log command example

Table 90: Temperature Log command example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that all signs should respond to this command.
D	Sign Address		"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>		02H	Start of TeXt character
F	Command Code		"F"	"F" (46H) is the "Read SPECIAL FUNCTION" Command Code. (See Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.)
G	Data Field	Special Functions Label	"L"	Read Temperature Log register.
H	<EOT>		04H	End Of Transmission character

Table 91: Temperature Log command response example



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	"0" (30H) is the Response code.
D	Sign Address	"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	"E" (45H) is the response to the Read Temperature Log request.
G	Special Functions Label	"L"	Read Temperature Log register.

Table 91: Temperature Log command response example

H	Special Functions Data	<CR><LF>	0DH 0AH	Used to format the log for readability.
I		Daily Log	"DAILY LOG: ..."	48 entries recorded every half-hour from the previous half-hour in the following format: TITLEAAAAABBBBBCCCCCDDDDD where: • TITLE = the ASCII string "DAILY LOG: Bn Bx Em Ex" which only appears once at the top of the entries. • AAAAA = five ASCII characters that represent the Controller board *minimum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "C" for Centigrade. • BBBBB = five ASCII characters that represent the Controller board *maximum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "C" for Centigrade. • CCCCC = five ASCII characters that represent the sign's external *minimum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "F" for Fahrenheit. • DDDDD = five ASCII characters that represent the sign's external *maximum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "F" for Fahrenheit.
J	Special Functions Data	6-Month Log	"6-MONTH LOG: . . ."	178 entries recorded for the previous 178 days in the following format: TITLEAAAAABBBBBCCCCCDDDDDEEEEE where: • TITLE = the ASCII string "6-MONTH LOG: MM-DD Bn Bx Em Ex" which only appears once at the top of the entries. • AAAAA = five ASCII characters representing the 2-digit month, a dash ("-" 2DH), and the 2-digit day. • BBBBB = five ASCII characters that represent the Controller board *minimum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "C" for Centigrade. • CCCCC= five ASCII characters that represent the Controller board *maximum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "C" for Centigrade. • DDDDD = five ASCII characters that represent the sign's external *minimum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "F" for Fahrenheit. • EEEEE = five ASCII characters that represent the sign's external *maximum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "F" for Fahrenheit.

K		Error Log	"ERROR LOG: . . ."	An event-driven log that records the last 48 errors which were caused by either dimming or shutdown. The error log is in the following format:
				TITLEAAAAABBBBBCDDDDDEEEEEFFFFFGGGGG
				• TITLE = the ASCII string "ERROR LOG: MM-DD ER Bn Bx Em Ex" which only appears once at the top of the entries.
				• AAAAA = five ASCII characters representing the 2-digit month, a dash ("-" 2DH), and the 2-digit day.
				• BBBBB = five ASCII characters representing the 2-digit hour, a colon (":" 3AH), and the 2-digit minute.
				• C = one ASCII number representing the type of error, where:
				  • "2" = Controller temperature caused overheat mode
				  • "5" = Controller caused dimming mode
				  • "6" = external temperature caused dimming mode
				• DDDDD = five ASCII characters that represent the Controller board *minimum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "C" for Centigrade.
				• EEEEE= five ASCII characters that represent the Controller board *maximum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "C" for Centigrade.
				• FFFFF = five ASCII characters that represent the sign's external *minimum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "F" for Fahrenheit.
				• GGGGG = five ASCII characters that represent the sign's external *maximum* temperature: a "+" or "-", followed by a 3-digit temperature, followed by "F" for Fahrenheit.

L	<ETX>		03H	End of TeXt character
M	Checksum		"023F"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
N	<EOT>		04H	End Of Transmission character

7.11.10 Read External Temperature command

NOTE: The packet format of this command is similar to "Read Temperature Log command example" on page 113.

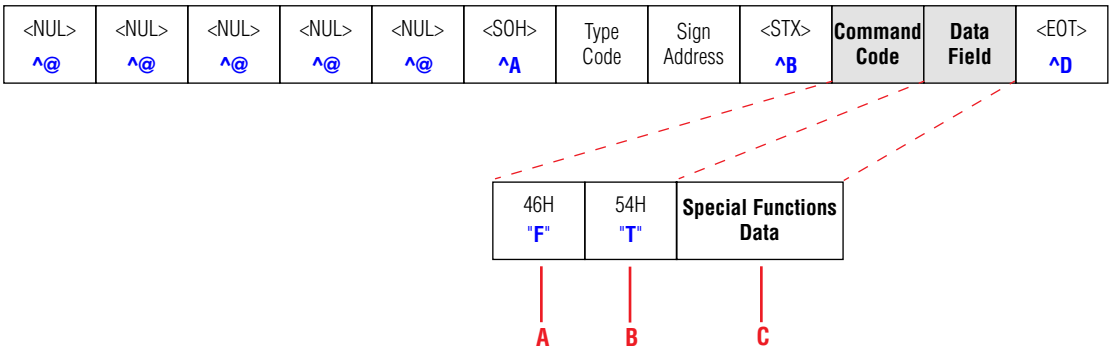
"T" — Sending "FT" will read the external temperature provided there is a functioning external temperature probe connected to the controller being queried.

If there is no probe connected or if it is not functioning properly, the sign will return "-127" for the temperature value (in Fahrenheit). In addition, "ERR" will appear on the sign in place of the temperature.

7.11.10.1 Read External Temperature command packet format

Table 92: Read External Temperature command packet format

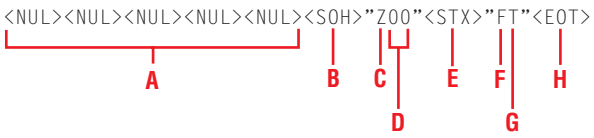
Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10):



Item	Name	Description
A	Command Code	"F" (46H) = Write SPECIAL FUNCTION file
B	Special Functions Label	"T" (54H) = Read External Temperature.
C	Special Functions Data	This data is returned in the response packet (see "Read Temperature Log command example" on page 113).

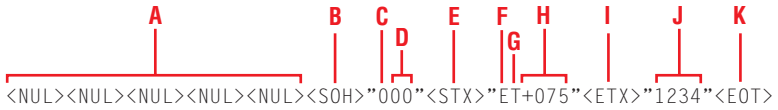
7.11.10.2 Read External Temperature command example

Table 93: Read External Temperature command example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that all signs should respond to this command.
D	Sign Address		"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>		02H	Start of TeXt character
F	Command Code		"F"	"F" (46H) is the "Read SPECIAL FUNCTION" Command Code. (See Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.)
G	Data Field	Special Functions Label	"T"	Read External Temperature command.
H	<EOT>		04H	End Of Transmission character

Table 94: Read External Temperature command response example



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	"0" (30H) is the Response code.
D	Sign Address	"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	"E" (45H) is the response to the Read External Temperature command.
G	Special Functions Label	"T"	Read External Temperature command.
H	Data Field	"+075"	The external temperature in degrees Fahrenheit.
I	<ETX>	03H	End of TeXt character
J	Checksum	"023F"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

7.11.11 Read Internal Temperature command

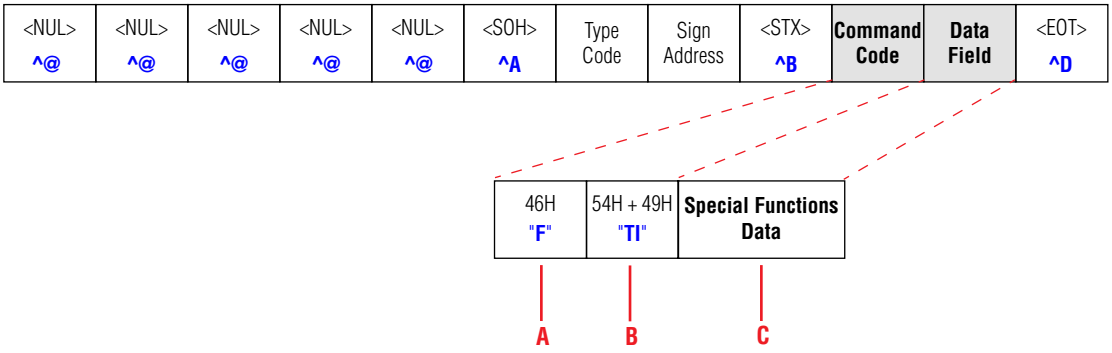
"TI" — Sending "FTI" will read the internal temperature.

NOTE: The format of this command is similar to "Read Temperature Log command example" on page 113.

7.11.11.1 Read Internal Temperature command packet format

Table 95: Read Internal Temperature command packet format

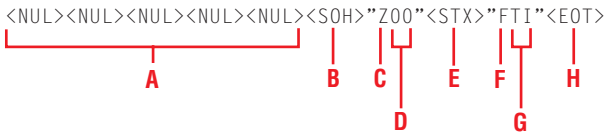
Standard transmission packet (see "Standard transmission packet ("1-byte" or "^A") format" on page 10):



Item	Name	Description
A	Command Code	"F" (46H) = Write SPECIAL FUNCTION file
B	Special Functions Label	"TI" (54H + 49H) = Read Internal Temperature.
C	Special Functions Data	This data is returned in the response packet (see "Read Temperature Log command example" on page 113).

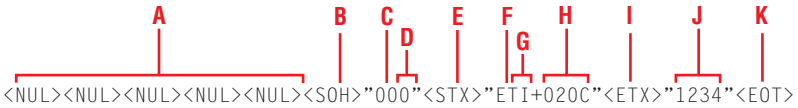
7.11.11.2 Read Internal Temperature command example

Table 96: Read Internal Temperature command example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	“Z” (5AH) means that all signs should respond to this command.
D	Sign Address		“00”	“00” (30H 30H) means that this command is sent to all signs.
E	<STX>		02H	Start of TeXt character
F	Command Code		“F”	“F” (46H) is the “Read SPECIAL FUNCTION” Command Code. (See Table 16, “Read SPECIAL FUNCTION Command Code format — “F” (46H),” on page 28.)
G	Data Field	Special Functions Label	“TI”	Read Internal Temperature command.
H	<EOT>		04H	End Of Transmission character

Table 97: Read Internal Temperature command response example



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>	01H	Start Of Header character
C	Type Code	“0”	“0” (30H) is the response code.
D	Sign Address	“00”	“00” (30H 30H) means that this command is sent to all signs.
E	<STX>	02H	Start of TeXt character
F	Command Code	“E”	“E” (45H) is the response to the Read Internal Temperature command.
G	Special Functions Label	“TI”	Read Internal Temperature command.
H	Data Field	“+020C”	The internal temperature in degrees Centigrade.
I	<ETX>	03H	End of TeXt character
J	Checksum	“023F”	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

7.11.12 Set Unit commands

SPECIAL NOTE

Set Unit commands are only available in AlphaEclipse 2500, 2600, 3500 Series B. and 3600 signs.

These commands are used to set sign parameters, such as the serial address. Once a sign receives a Set Unit command, the sign will reset and go through its power-up messages. In addition, the message "DIP DISABLED" will appear. The sign's DIP switch settings will now be ignored.

Further changes to sign parameters can then only be made through a Set Unit command — unless the "UN" command is sent to the sign.

When the "UN" command is sent to a sign, the sign will use its DIP switch settings.

Sending a clear memory command ("E$"), a soft reset command ("E,"), or updating the firmware will have no affect on a sign's parameters.

Multiple write Set Unit commands can be combined in a packet, for example:

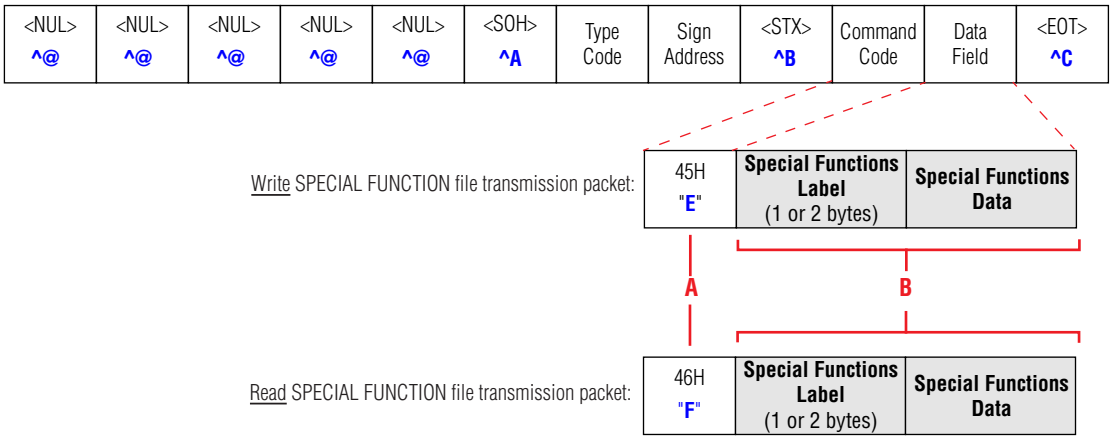
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"EU10080008OU201U501000110000000000"<EOT>

U1 command

U2 command

U5 command

Table 98: Set Unit commands packet format



Special Functions Label (2 bytes)	Special Functions Data Command Code "E" (45H) sets sign parameters. Command Code "F" (46H) reads sign parameters.
"U1" 55H 31H	**Set/Read Unit Size** (write or read)— sets or reads the sign's columns and rows. Eight ASCII hexadecimal digits in the format AAAABBBB where: • AAAA = Column size of the sign (valid values = 0010 – 00F0 columns) • BBBB = Row size of the sign (valid values = 0008 – 0080 rows)
"U2" 55H 32H	**Set/Read Unit Run Mode** (write or read) — sets or reads the mode that the sign is running. Two ASCII hexadecimal digits where: • "00" 30H + 30H = Run normally **(default)** • "01" 30H + 31H = Test 1 (production test mode) • "02" 30H + 32H = Test 2 (test pattern) • "03" 30H + 33H = Test 3 (test match mode) • "04" 30H + 34H = Test 4 (temperature test mode) • "05" – "FF" = Future use
"U3" 55H 33H	**Set/Read Unit Serial Address** (write or read) — sets or reads the sign's serial address. Two ASCII hexadecimal digits from 0 to 255 in the following format: • "00" 30H + 30H to "FF" 46H + 46H

Table 98: Set Unit commands packet format

"U4" 55H 34H	**Set/Read Unit Serial Data** (write or read) — sets or reads the sign's baud rate and data format. Two ASCII hexadecimal digits from 0 to 12 in the following format: NOTE: Note that this command will reset the baud rate. Your next packet must be at that baud rate. You cannot use this command packet in a nested transmission. • "00" 30H + 30H = Autobaud from 38400 baud (8N1/7E2 data format) • "01" 30H + 31H = 1200 baud (8N1 data format) • "02" 30H + 32H = 1200 baud (7E2 data format) • "03" 30H + 33H = 2400 baud (8N1 data format) • "04" 30H + 34H = 2400 baud (7E2 data format) • "05" 30H + 35H = 4800 baud (8N1 data format) • "06" 30H + 36H = 4800 baud (7E2 data format) • "07" 30H + 37H = 9600 baud (8N1 data format) • "08" 30H + 38H = 9600 baud (7E2 data format) • "09" 30H + 39H = 19200 baud (8N1 data format) • "0A" 30H + 3AH = 19200 baud (7E2 data format) • "0B" 30H + 34B = 38400 baud (8N1 data format) • "0C" 30H + 3CH = 38400 baud (7E2 data format) When a sign is configured for autobaud, every packet sent to the display must be preceded by at least five <NUL> or <SOH> characters in order for the firmware to be able to calculate the baud rate of the transmission.

"U5" 55H 35H	**Set/Read Unit Configuration** (write or read) — sets or reads various sign parameters. Seventeen ASCII characters in the format FGHIJKLZZZZZZZZZZ where:
	
	• F = Clear memory flag
	  • "0" 30H — Do not clear memory on power-up
	  • "1" 31H — Clear memory on power-up (simulates a virgin power-up, the first time power is applied to a sign)
	• G = Master/Slave flag
	  • "0" 30H — Master sign
	  • "1" 31H — Slave sign
	• H = Demo message flag (not applicable for AlphaEclipse™ signs, but a value must be used as a place holder)
	  • "0" 30H — Off
	  • "1" 31H — On
	• I = Color flag (not applicable for AlphaEclipse™ signs, but a value must be used as a place holder)
	  • "0" 30H — Mono
	  • "1" 31H — Color unit
	• J = IR flag (not applicable for AlphaEclipse™ signs, but a value must be used as a place holder)
	  • "0" 30H — IR off
	  • "1" 31H — IR on
	• K = RS485 echo flag (not applicable for AlphaEclipse™ signs, but a value must be used as a place holder)
	  • "0" 30H — Off
	  • "1" 31H — On
	• L = Driver height
	  • "0" 30H — 8 High
	  • "1" 31H —16 High
	• ZZZZZZZZZZ — Ten ASCII characters. For future use. Send ten "0" 30H if not used. (not applicable for AlphaEclipse™ signs, but a value must be used as a place holder)

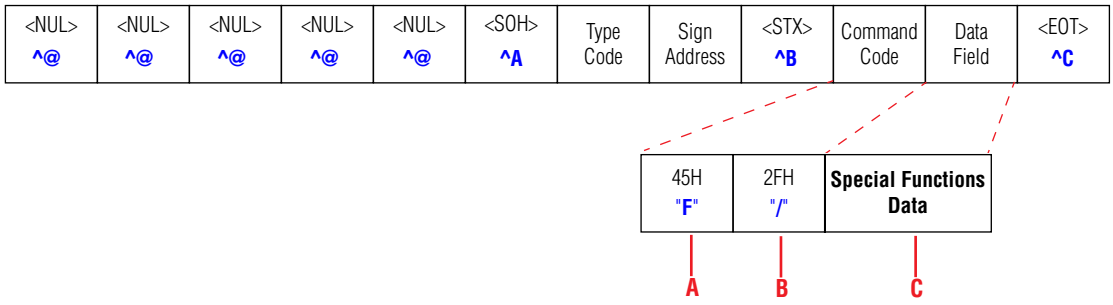
"U6" 55H 36H	**Read Unit Register** (read only) — reads the sign's DIP switches and memory (RAM). Twelve ASCII hexadecimal digits in the format AABBCCDDXXXX where: • AA = DIP switch bank 1 value • BB = DIP switch bank 2 value • CC = DIP switch bank 3 value • DD = DIP switch bank 4 value • XXXX = total amount of RAM in kilobytes (for example, "03E8" = 1000 decimal = 1000 kilobytes = 1 megabyte)
"UN" 55H 4EH	**Reset command** (write only) — for an AlphaEclipse 2500, 2600, and 3500, this command resets all parameters to the values set on the sign's DIP switches. After receiving this command, a sign will use its DIP switch settings for parameter values. For an AlphaEclipse 3600 sign, the sign is reset to its default factory settings and custom user configurations are erased.

7.11.13 Read Dim Times command

This Read SPECIAL FUNCTION command returns the sign's dim on and off times encoded in a four-byte, ASCII hexadecimal code. For the meaning of these codes, see "Appendix B: Valid Start and Stop times" on page 50.

7.11.13.1 Read Dim Times command packet format

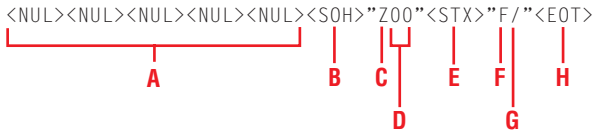
Table 99: Read Dim Times command packet format



Special Functions Label		Special Functions Data Command Code "F" (46H) reads sign parameters.
A	Command Code	"F" (46H) = Read SPECIAL FUNCTION command
B	Special Functions Label	"/" (2FH) = Read Dim Times
C	Special Functions Data	This data is returned in the response packet (see Table 101, "Read Dim Times command response example," on page 120).

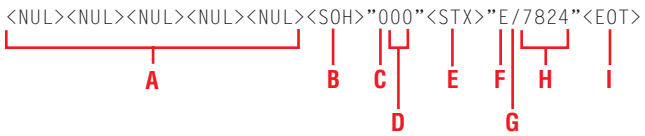
7.11.13.2 Read Dim Times command example

Table 100: Read Dim Times command example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	"Z" (5AH) means that all signs should respond to this command.
D	Sign Address		"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>		02H	Start of TeXt character
F	Command Code		"F"	"F" (46H) is the "Read SPECIAL FUNCTION" Command Code. (See Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.)
G	Data Field	Special Functions Label	"/"	Read Dim Times command.
H	<EOT>		04H	End Of Transmission character

Table 101: Read Dim Times command response example



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)

Table 101: Read Dim Times command response example

B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	"0" (30H) is the Response code.
D	Sign Address	"00"	"00" (30H 30H) means that this command is sent to all signs.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	"E" (45H) is the response to a Read SPECIAL FUNCTION command.
G	Special Functions Label	"/"	Read Dim Times
H	Data Field	"7824"	Four, encoded ASCII hexadecimal characters that represent the dim on and dim off times. In this case, • "78" = a dim on time of 8:00 pm • "24" = a dim off time of 6:00 am For a list of these encoded times, see "Appendix B: Valid Start and Stop times" on page 50.
I	<EOT>	04H	End Of Transmission character

7.12 Appendix L: Alpha 3.0 protocol additions

NOTE: As of the writing of this protocol manual, the Alpha 3.0 protocol is only available for the AlphaEclipse 3600 sign.

The Alpha 3.0 protocol adds the following functions to the existing Alpha 1.0 and Alpha 2.0 protocols:

Table 102: Alpha 3.0 protocol additions

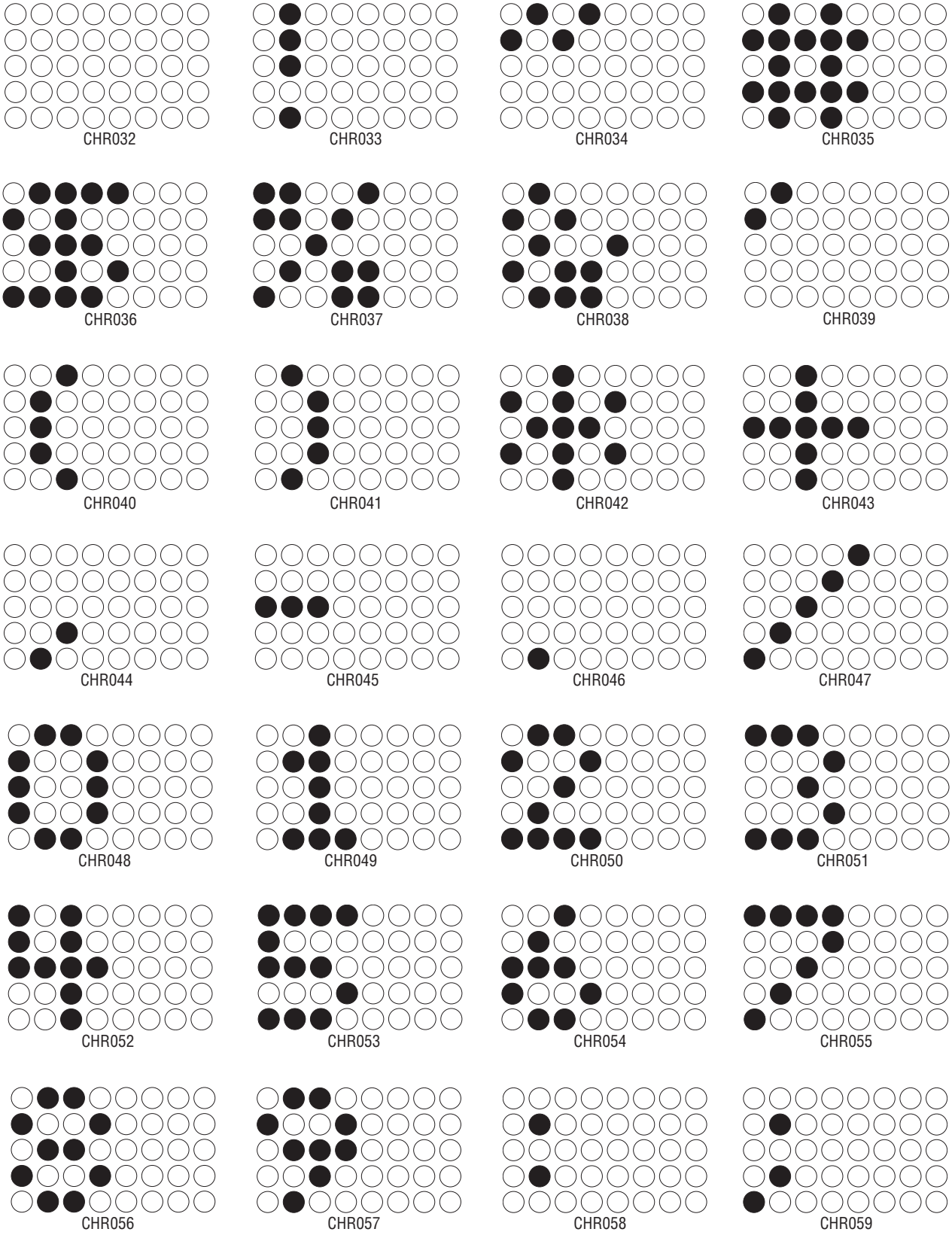
Function	Type	Description	Reference
Explode Mode	Standard Mode “u” 75H	Text “explodes” into four pieces and directions.	Table 64, “Standard Modes,” on page 88
Clock Mode	Standard Mode “v” 76H	A clockwise text wipe	Table 64, “Standard Modes,” on page 88
Left/Right Display Position	Text file Left Display Position “1” 31H Text file Right Display Position “2” 32H	These two new positions work like the Top and Bottom positions, but for the left and right parts of the display.	Table 12, “Write TEXT file transmission packet format,” on page 18
Faster Flicks	Control Code for Call picture or animation file	Faster Flicks can be displayed in 0.01 second increments instead of 0.1.	“Control codes (00 – 1FH)” on page 80.
Clear Compact Flash	Write SPECIAL FUNCTION Special Functions Label and data “$$$$”	Used to clear memory and compact flash.	See the Special Functions Label “$” in Table 15, “Write SPECIAL FUNCTION Command Code format — “E” (45H),” on page 21.
Color functions	• Character color — Control Code for Character Color <1CH>”Z”	• RGB (Red-Green-Blue) character color coding added which permits over 1.6 million (256 x 256 x 256) color combinations.	• “Control codes (00 – 1FH)” on page 80.
	• Shadow color — Control Code for Character Color <1CH>”Y”	• RGB (Red-Green-Blue) character shadow color coding added which permits over 1.6 million (256 x 256 x 256) color combinations.	• “Control codes (00 – 1FH)” on page 80.
	• Write SPECIAL FUNCTION Special Functions Label “8” 38H Memory Configuration for an RGB DOTS PICTURE	• Used to set up sign memory for an RGB LARGE DOTS PICTURE.	• See the Special Functions Label “8” in Table 15, “Write SPECIAL FUNCTION Command Code format — “E” (45H),” on page 21.
	• Write RGB DOTS PICTURE Command Code “K” 4BH	• Used to create an RGB DOTS PICTURE file in a sign.	• See “Write RGB DOTS PICTURE file Command Code — “K” (4BH)” on page 43.
	• Read RGB DOTS PICTURE Command Code “L” 4CH	• Use to read an RGB DOTS PICTURE file from a sign	• See “Read RGB DOTS PICTURE file Command Code — “L” (4CH)” on page 45.
	• Call RGB DOTS PICTURE Control Code <1FH>	• Used to display an RGB DOTS PICTURE on a sign.	• See “Control codes (00 – 1FH)” on page 80.
	• Write/Read SPECIAL FUNCTION Special Functions Label “C” 43H Color Correction command for an RGB or mono-color AlphaEclipse 3600 sign.	• Use Write to turn RGB or red gamma color correction on or off. Red gamma correction is used for mono-color (red or amber) AlphaEclipse 3600 signs. Use Read to find out if color correction is on or off.	• See the Special Functions Label “C” in Table 15, “Write SPECIAL FUNCTION Command Code format — “E” (45H),” on page 21 and in Table 16, “Read SPECIAL FUNCTION Command Code format — “F” (46H),” on page 28.
Read Firmware Revisions	Read SPECIAL FUNCTION Special Functions Label “v” 76H	Used to read the firmware and FPGA versions.	See “v” in Table 16, “Read SPECIAL FUNCTION Command Code format — “F” (46H),” on page 28.

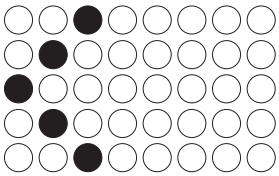
Table 102: Alpha 3.0 protocol additions

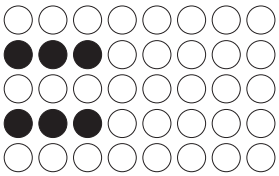
Function	Type	Description	Reference
Read Serial Error Log	Read SPECIAL FUNCTION Special Functions Label "*L" 2AH 4CH	Used to read the serial error log of the last 256 received packets.	See "*L" in Table 16, "Read SPECIAL FUNCTION Command Code format — "F" (46H)," on page 28.
New Set Unit commands	Write/Read SPECIAL FUNCTION "U7", "U8", and "U9".	Used to communicate with a sign's internal network ("U7" and "U9") and trigger a message on a slave sign ("U8").	See "U7", "U8", and "U9" in Table 15, "Write SPECIAL FUNCTION Command Code format — "E" (45H)," on page 21.

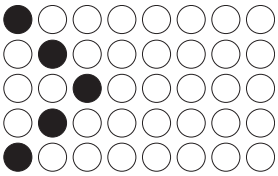
7.13 Appendix M: Font character sets

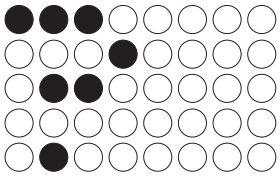
7.13.1 5-High Regular (SS5)

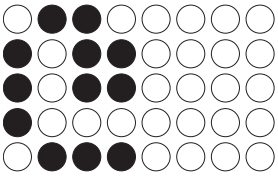


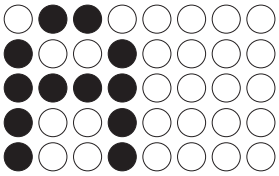

CHR060

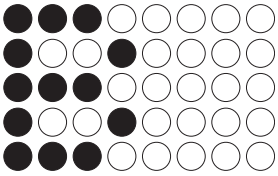

CHR061

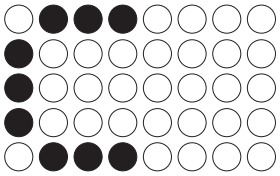

CHR062

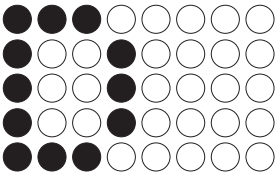

CHR063

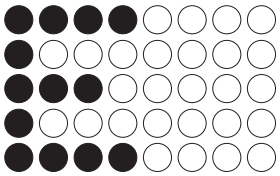

CHR064

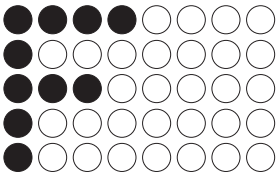

CHR065

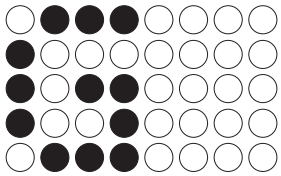

CHR066

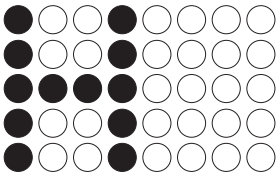

CHR067

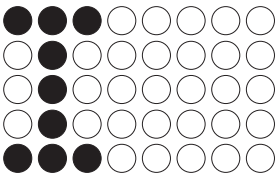

CHR068

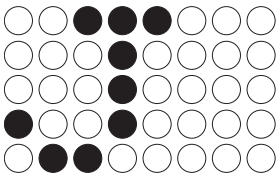

CHR069

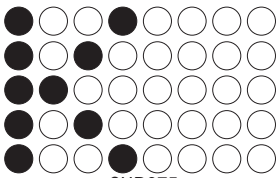

CHR070

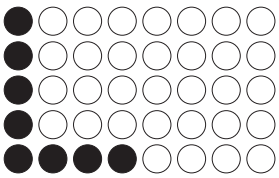

CHR071

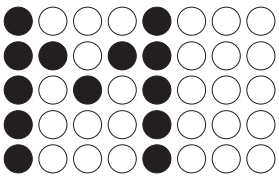

CHR072

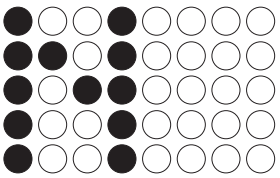

CHR073

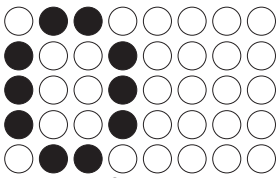

CHR074

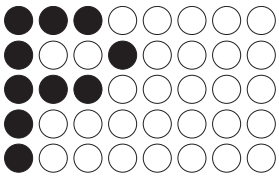

CHR075

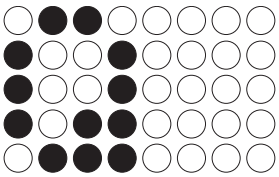

CHR076

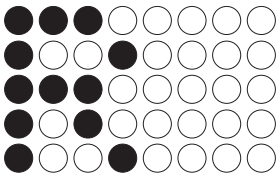

CHR077

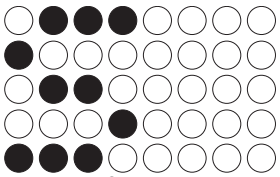

CHR078

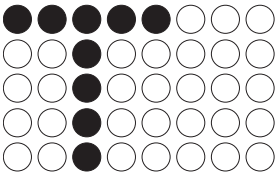

CHR079

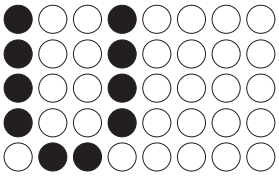

CHR080

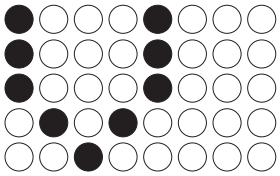

CHR081

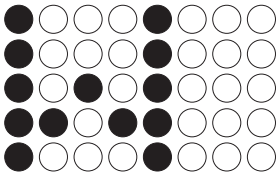

CHR082

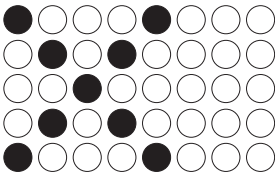

CHR083

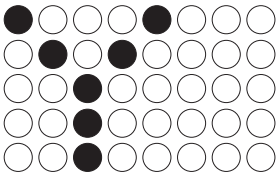

CHR084

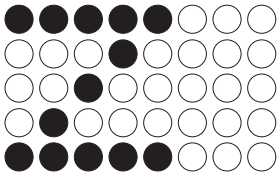

CHR085

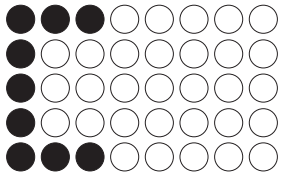

CHR086

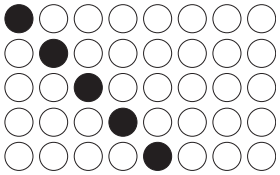

CHR087

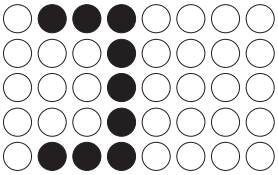

CHR088

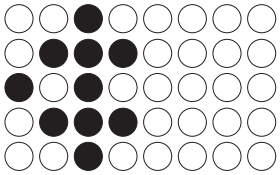

CHR089

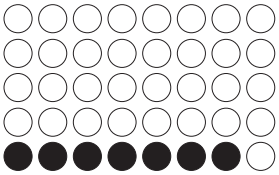

CHR090

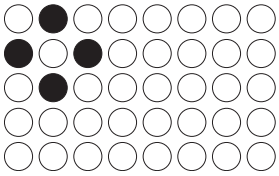

CHR091


CHR092

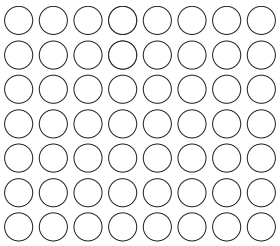

CHR093

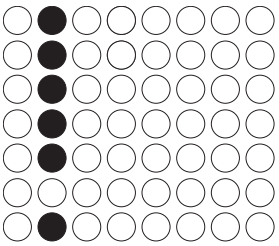

CHR094

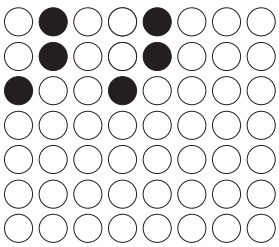

CHR095

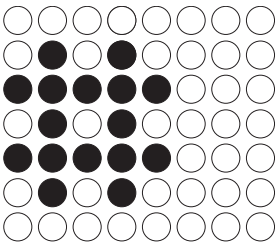

CHR096

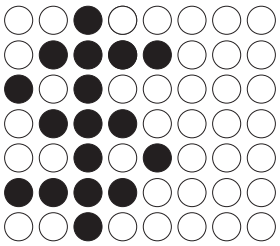
7.13.2 7-High Regular (SS7)

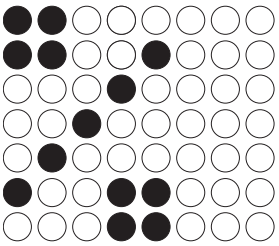

CHR032

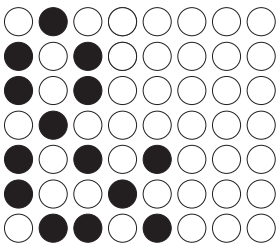

CHR033

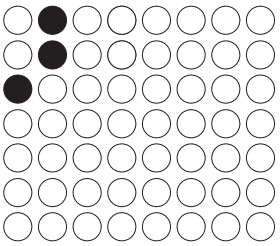

CHR034

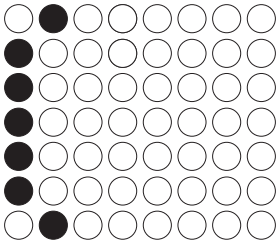

CHR035

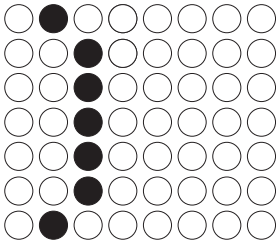

CHR036

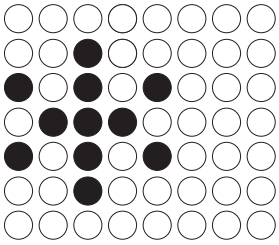

CHR037

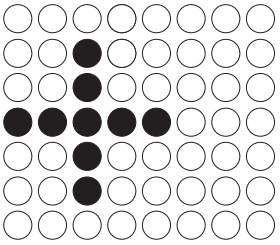

CHR038

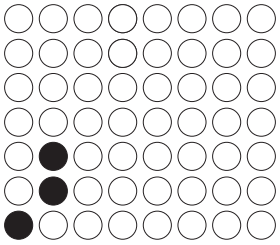

CHR039

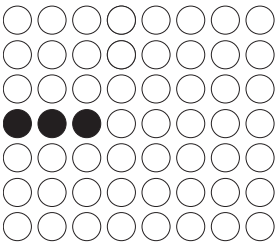

CHR040

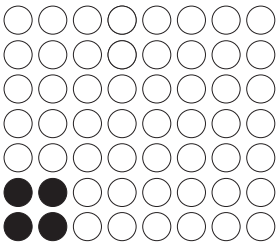

CHR041

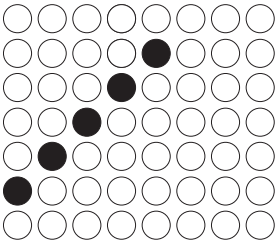

CHR042

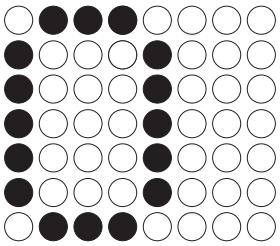

CHR043

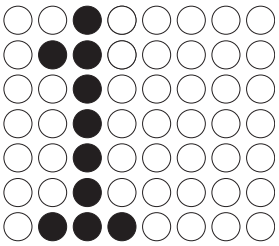

CHR044

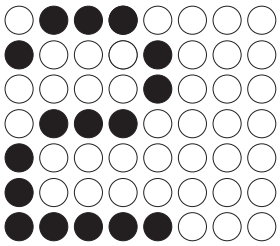

CHR045

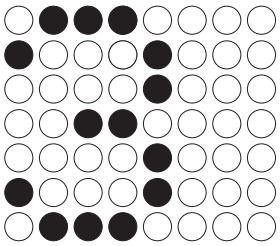

CHR046

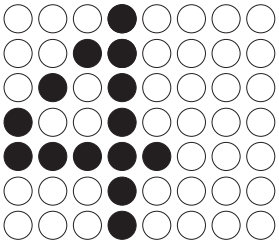

CHR047

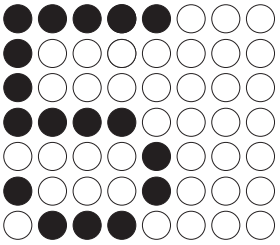

CHR048

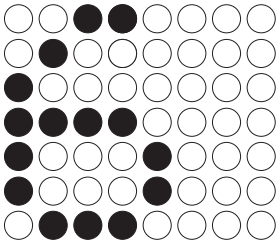

CHR049

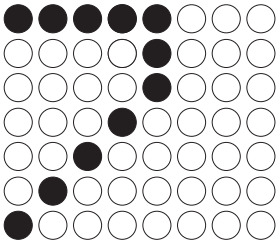

CHR050

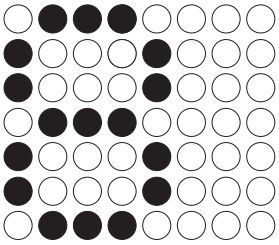

CHR051

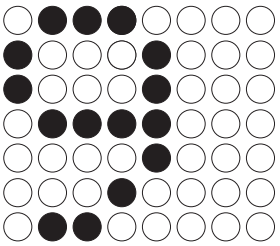

CHR052

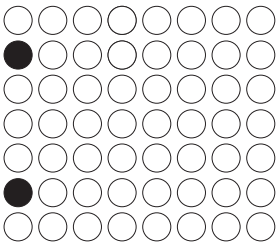

CHR053

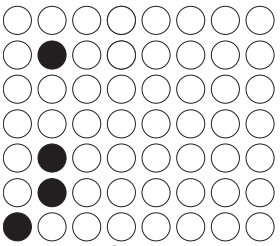

CHR054

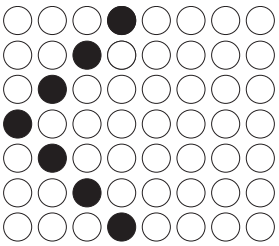

CHR055

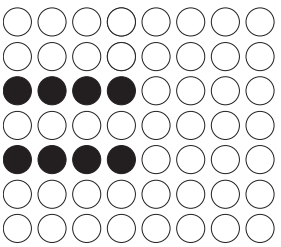

CHR056

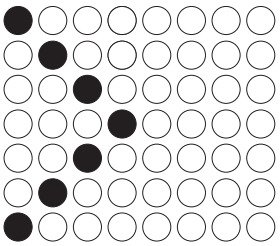

CHR057

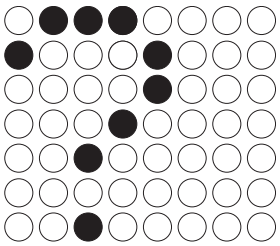

CHR058

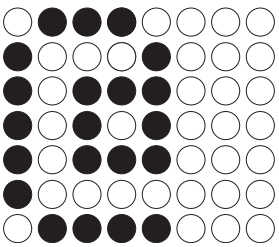

CHR059

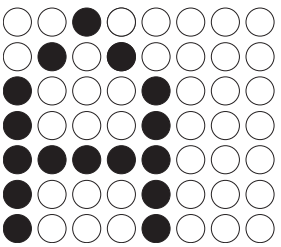

CHR060

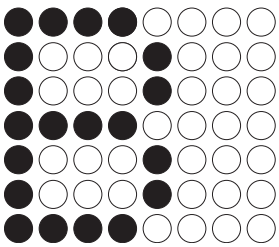

CHR061

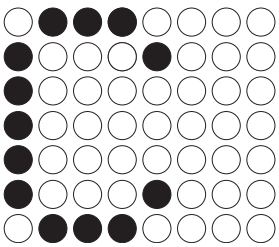

CHR062

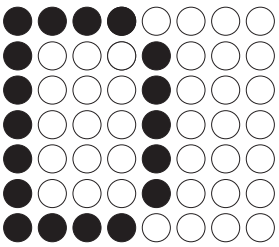

CHR063

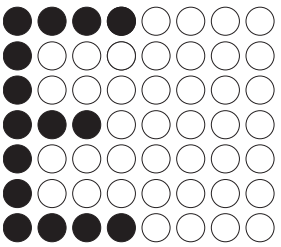

CHR064

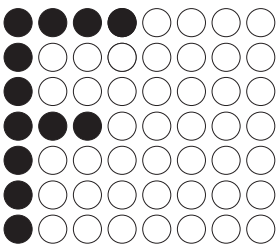

CHR065

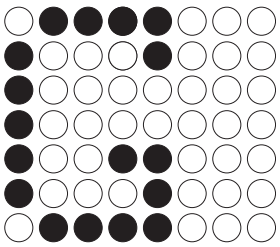

CHR066

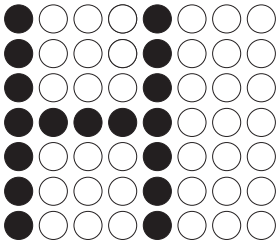

CHR067

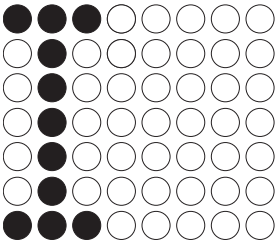

CHR068

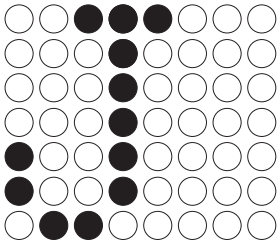

CHR069

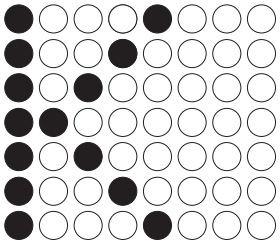

CHR070

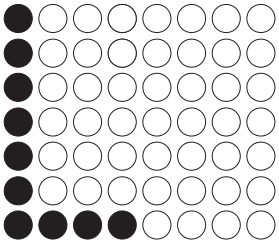

CHR071

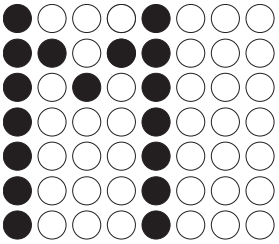

CHR072

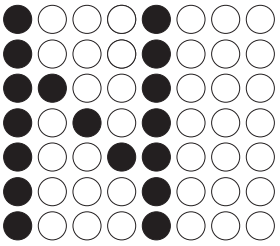

CHR073

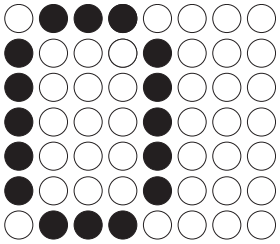

CHR074

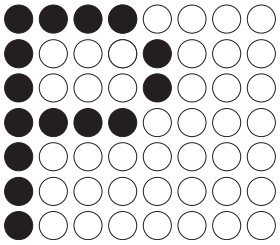

CHR075

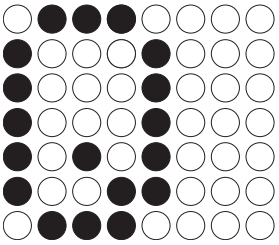

CHR076

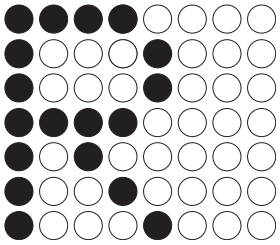

CHR077

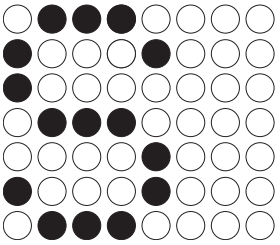

CHR078

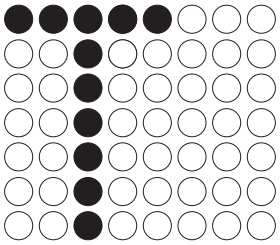

CHR079

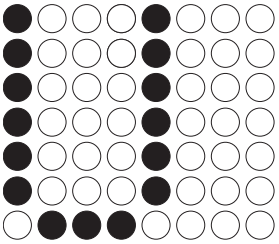

CHR080

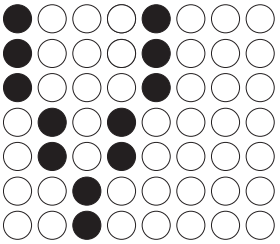

CHR081

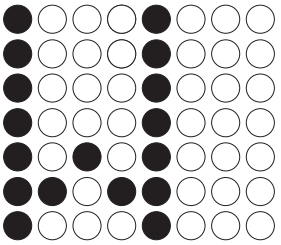

CHR082

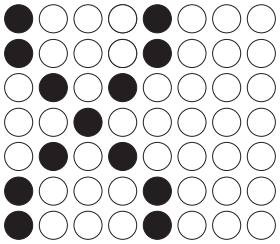

CHR083

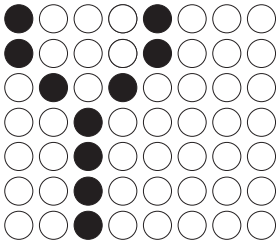

CHR084

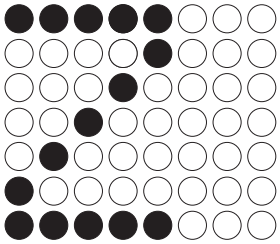

CHR085

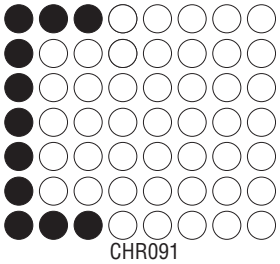

CHR086

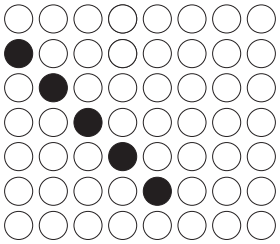

CHR087

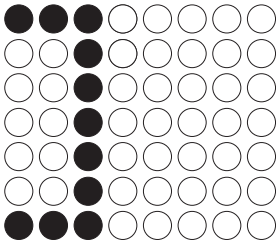

CHR088

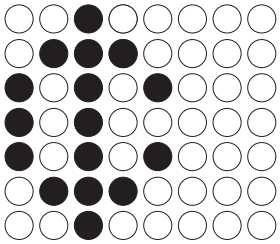

CHR089

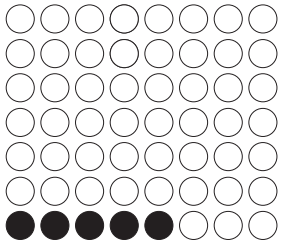

CHR090

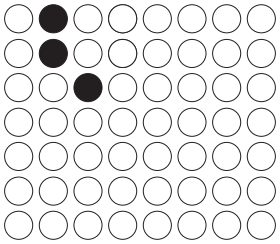

CHR091

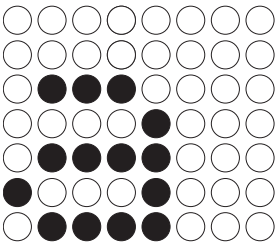

CHR092

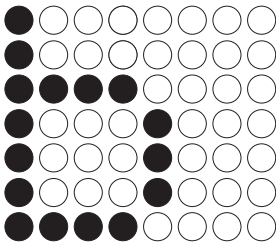

CHR093

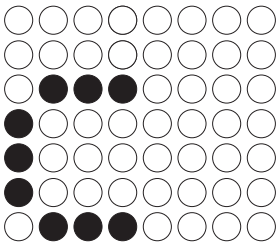

CHR094

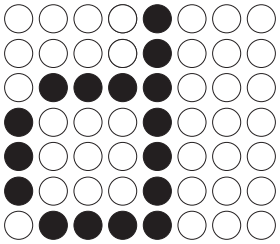

CHR095

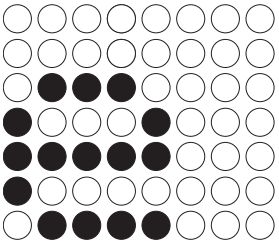

CHR096

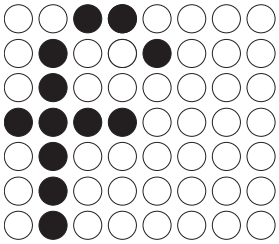

CHR097

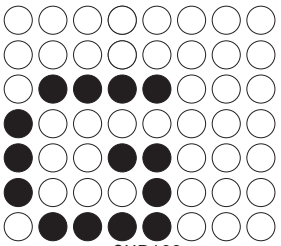

CHR098

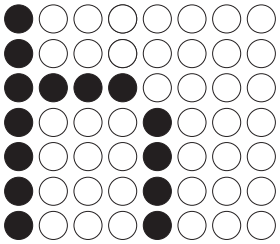

CHR099

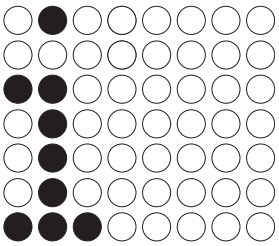

CHR100

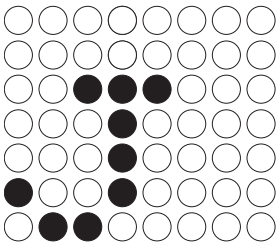

CHR101

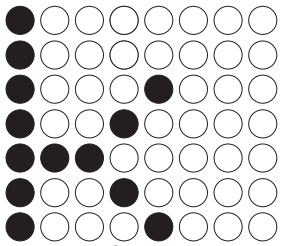

CHR102

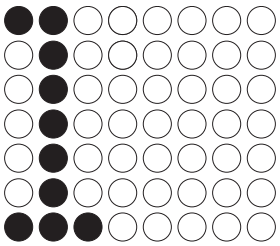

CHR103

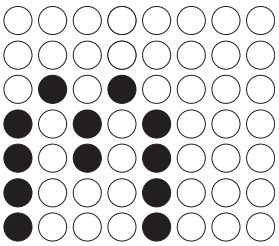

CHR104

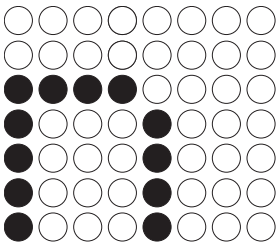

CHR105

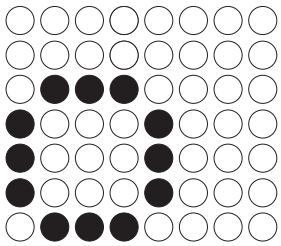

CHR106

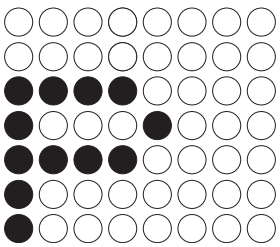

CHR107

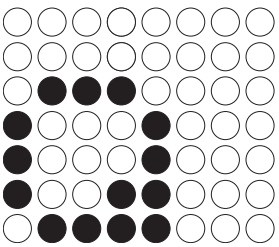

CHR108

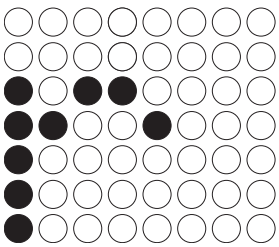

CHR109

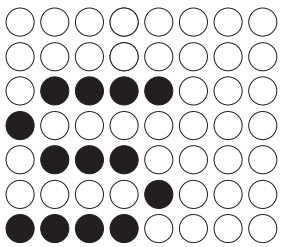

CHR110

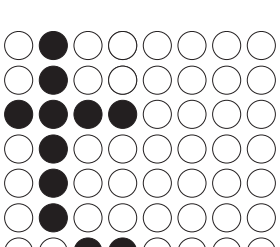

CHR111

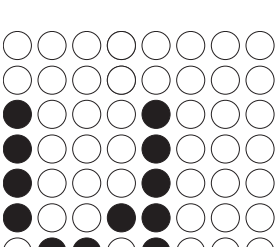

CHR112

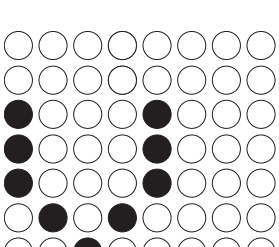

CHR113

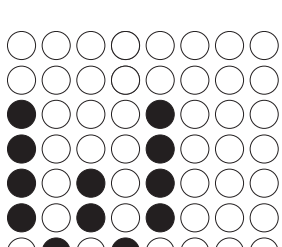

CHR114

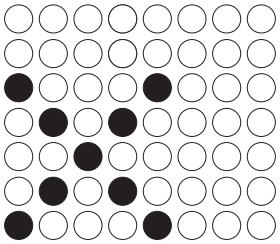

CHR115

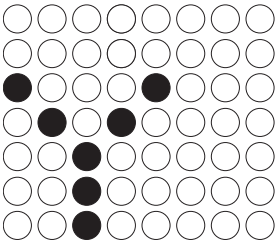

CHR116

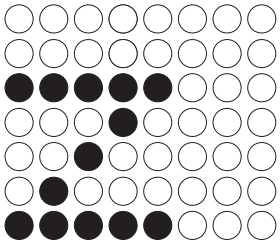

CHR117

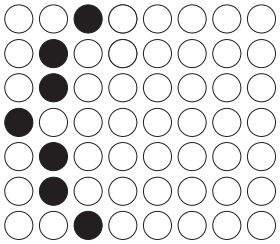

CHR118

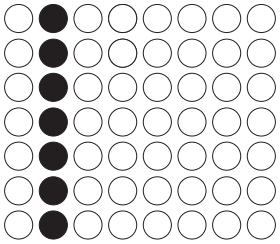

CHR119

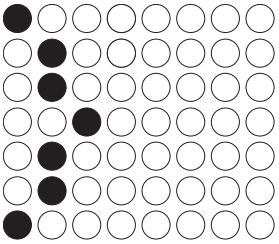

CHR120

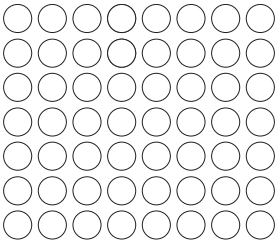

CHR121

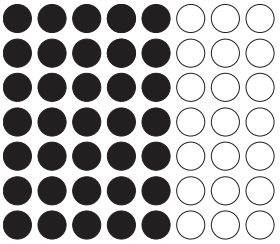

CHR122

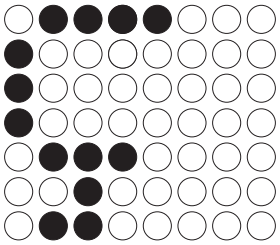

CHR123

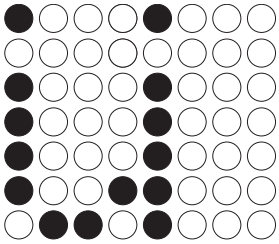

CHR124

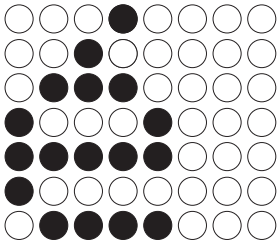

CHR125

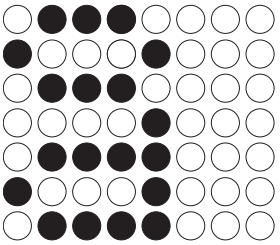

CHR126

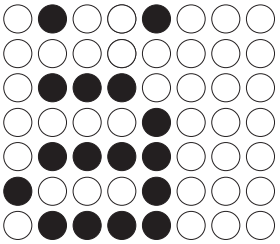

CHR127

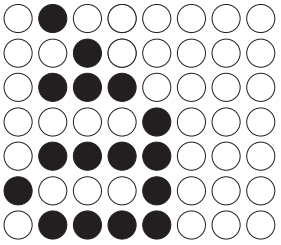

CHR128

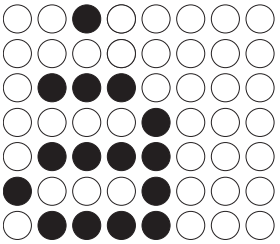

CHR129

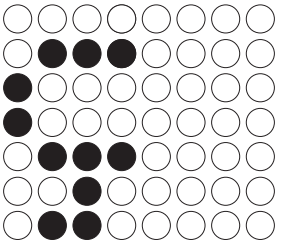

CHR130

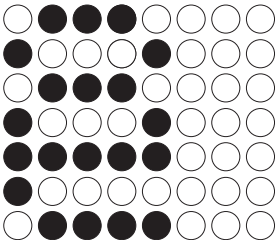

CHR131

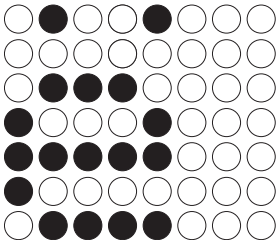

CHR132

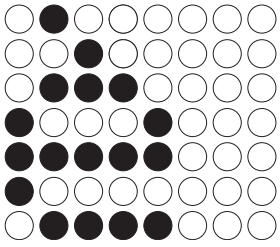

CHR133

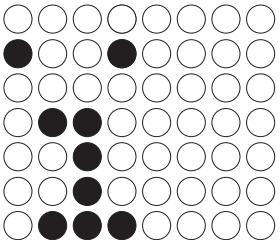

CHR134

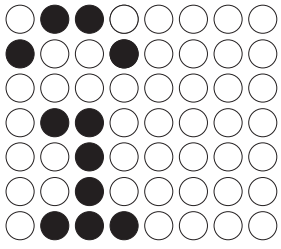

CHR135

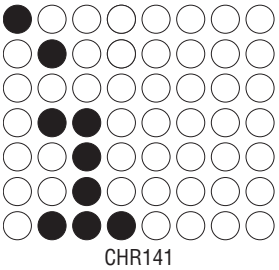

CHR136

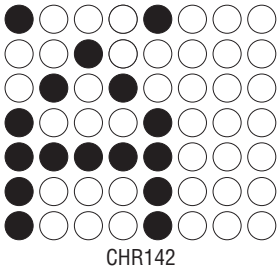

CHR137

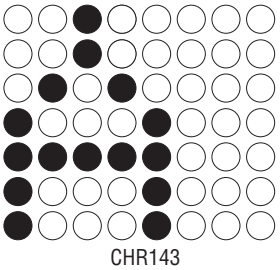

CHR138

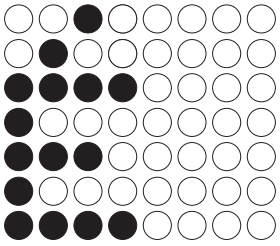

CHR139

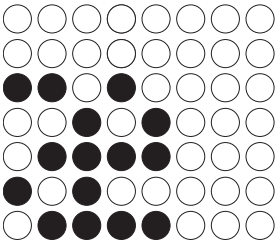

CHR140

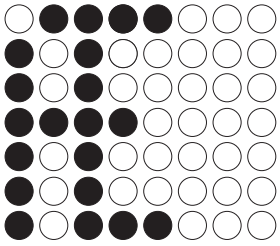

CHR141

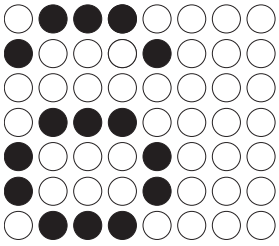

CHR142

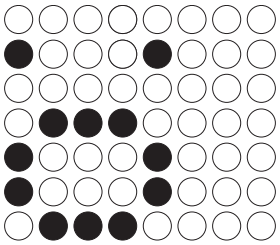

CHR143

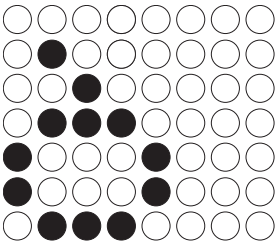

CHR144

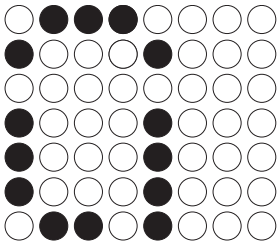

CHR145

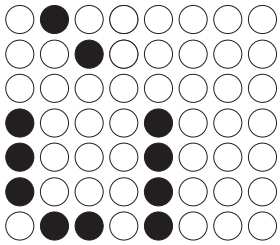

CHR146

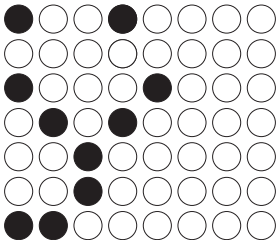

CHR147

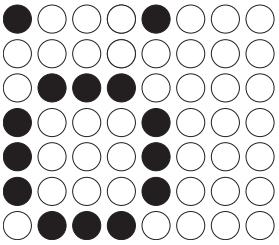

CHR148

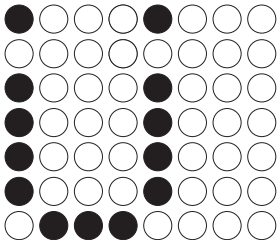

CHR149

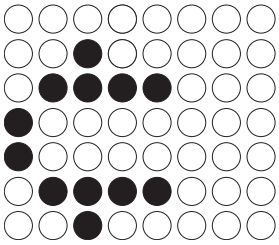

CHR150

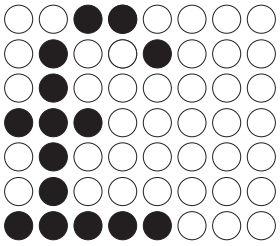

CHR151

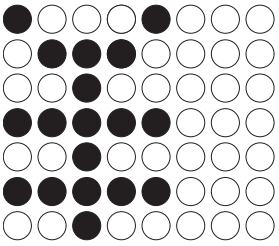

CHR152

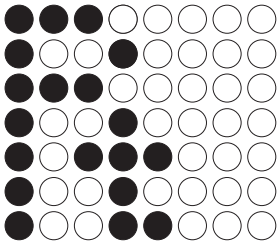

CHR153

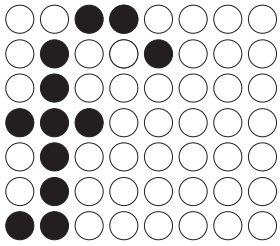

CHR154

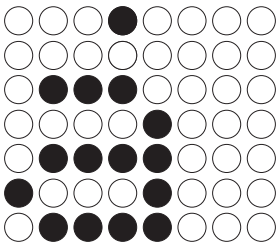

CHR155

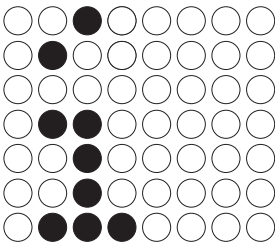

CHR156

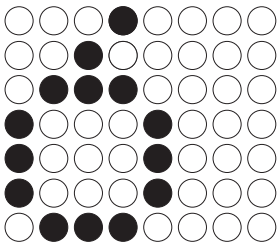

CHR157

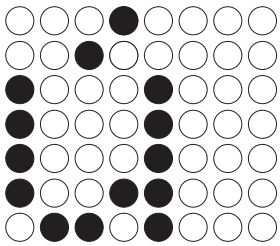

CHR158

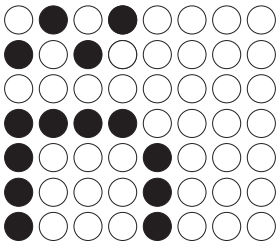

CHR159

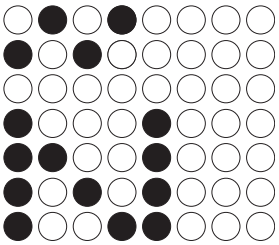

CHR160

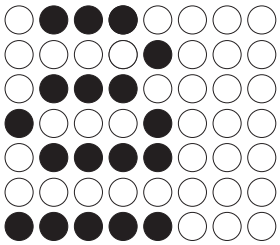

CHR161

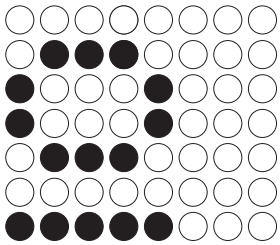

CHR162

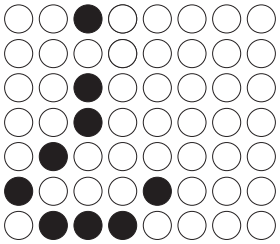

CHR163

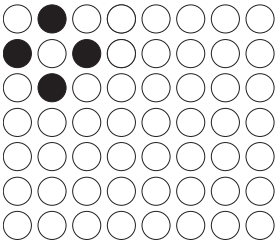

CHR164

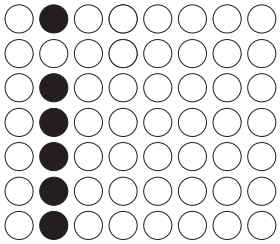

CHR165

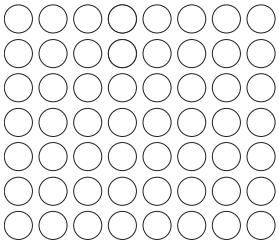

CHR166

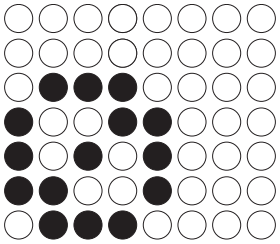

CHR167

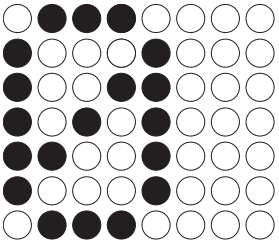

CHR168

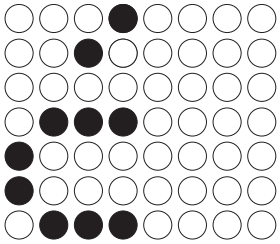

CHR169

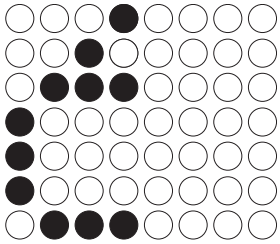

CHR170

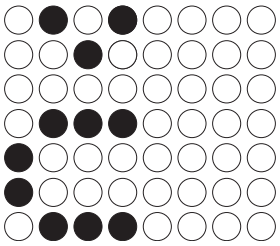

CHR171

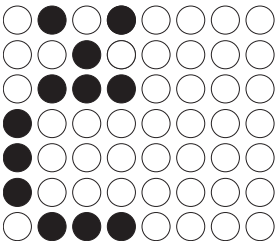

CHR172

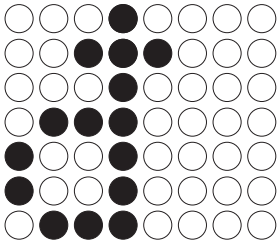

CHR173

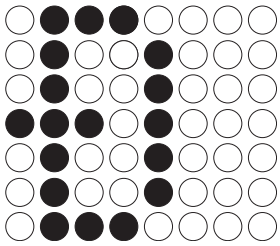

CHR174

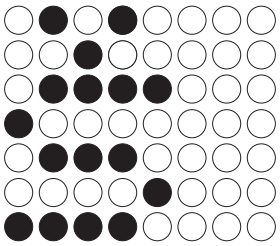

CHR175

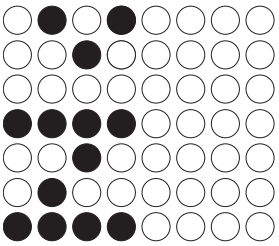

CHR176

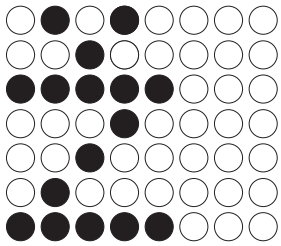

CHR177

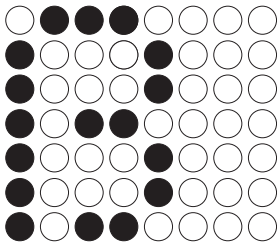

CHR178

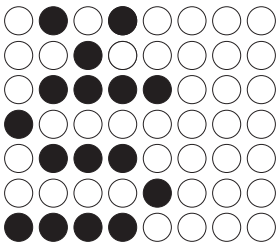

CHR179

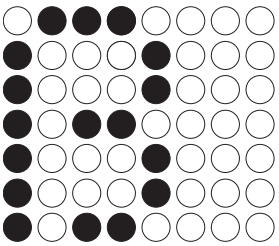

CHR180

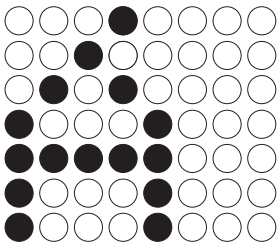

CHR181

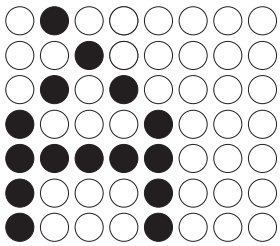

CHR182

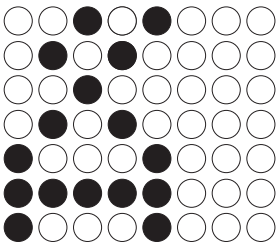

CHR183

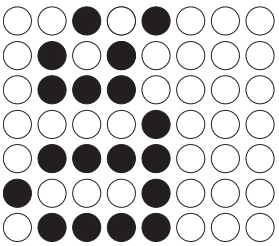

CHR184

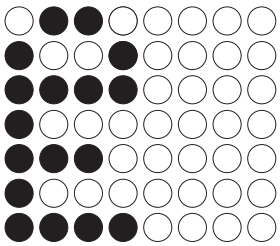

CHR185

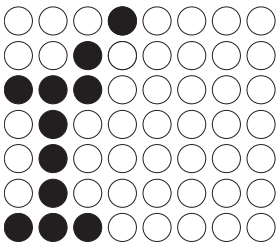

CHR186

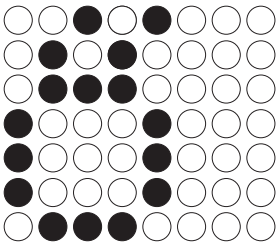

CHR187

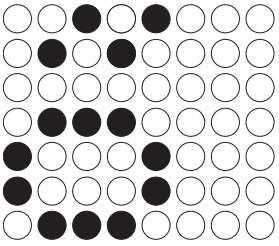

CHR188


CHR189

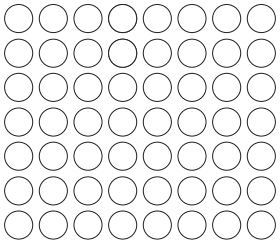

CHR190

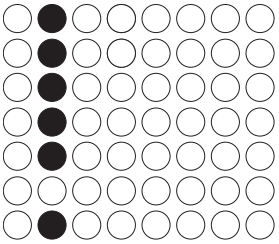

CHR191

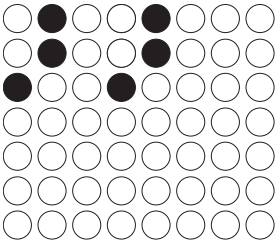

CHR192

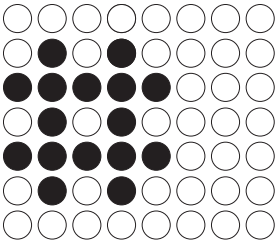

CHR193

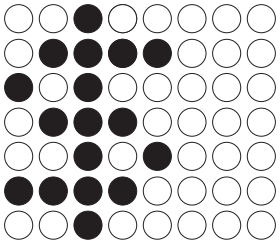
7.13.3 7-High Fancy (SF7)

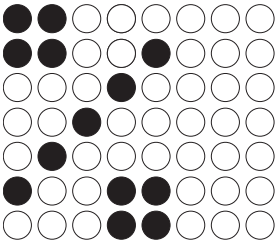

CHR032

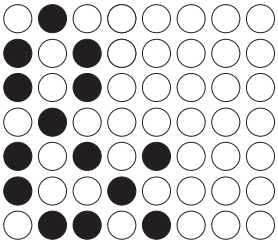

CHR033

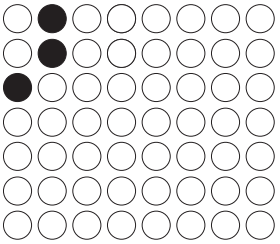

CHR034

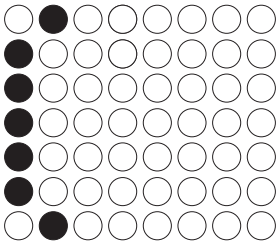

CHR035

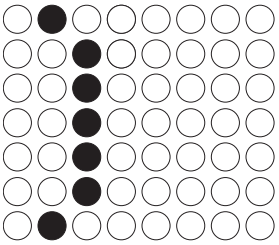

CHR036

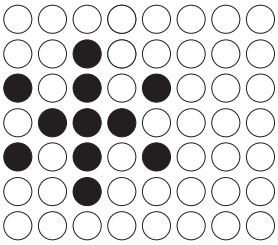

CHR037

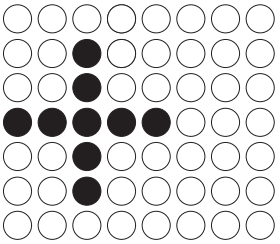

CHR038

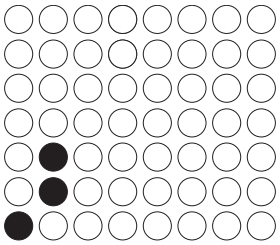

CHR039

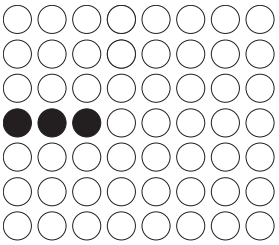

CHR040

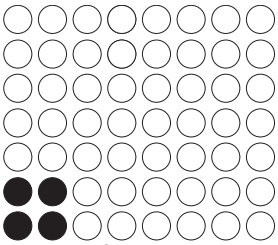

CHR041

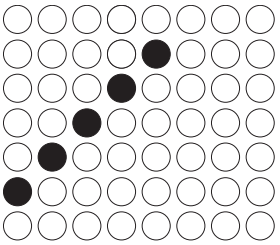

CHR042

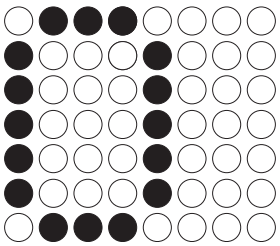

CHR043

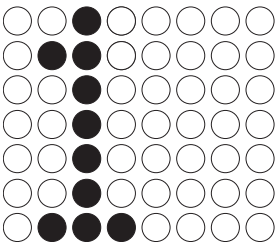

CHR044

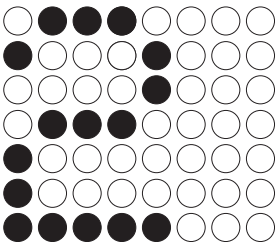

CHR045

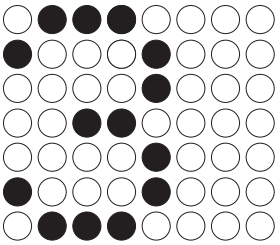

CHR046

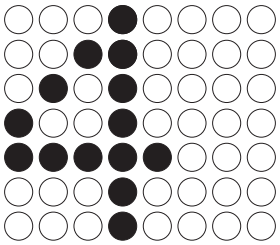

CHR047

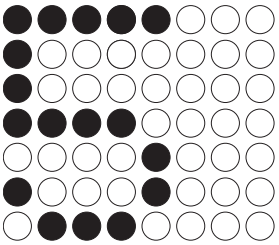

CHR048

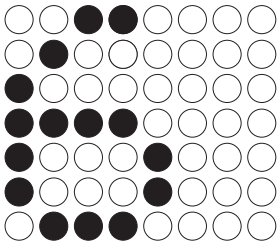

CHR049

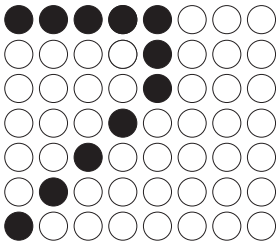

CHR050

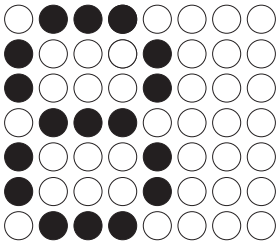

CHR051

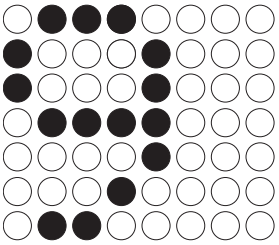

CHR052

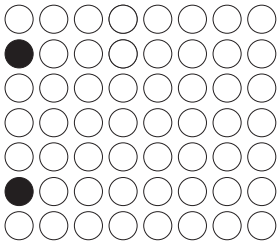

CHR053

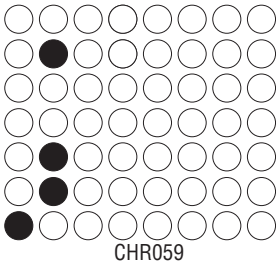

CHR054

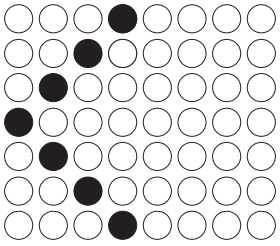

CHR055

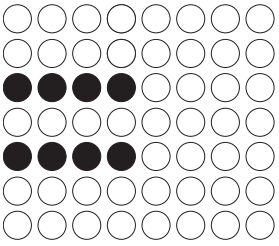

CHR056

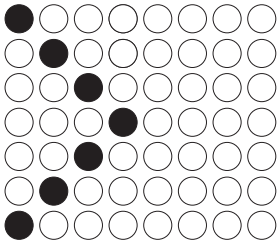

CHR057

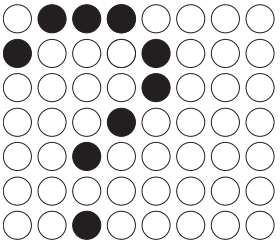

CHR058

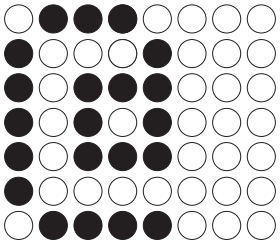

CHR059

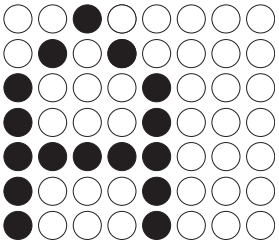

CHR060

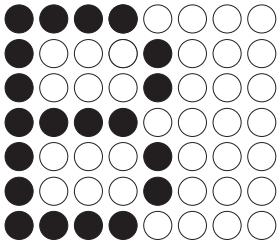

CHR061

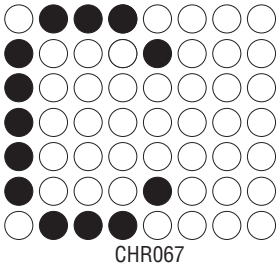

CHR062

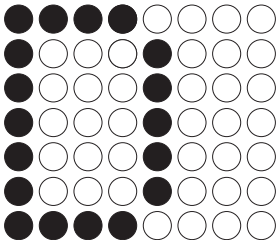

CHR063

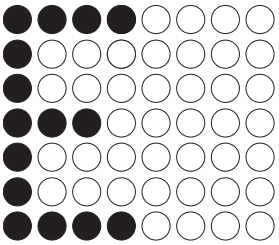

CHR064

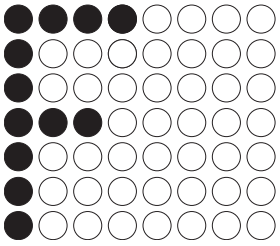

CHR065

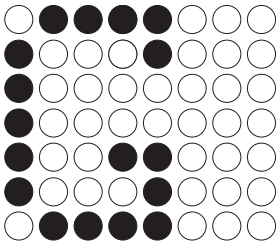

CHR066

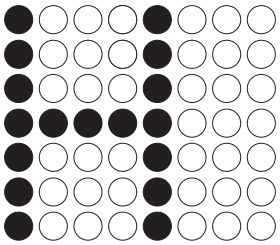

CHR067

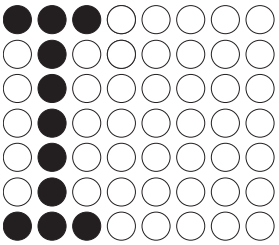

CHR068

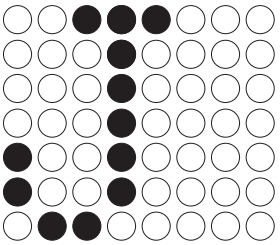

CHR069

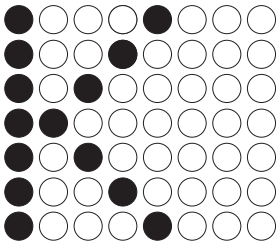

CHR070

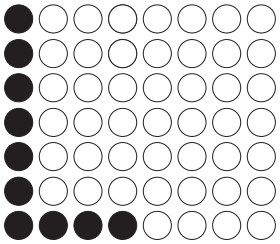

CHR071

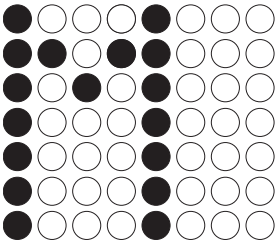

CHR072

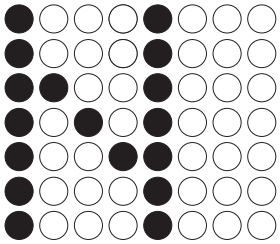

CHR073

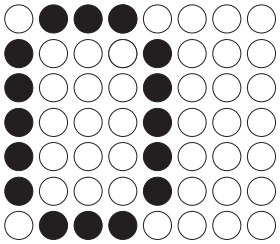

CHR074

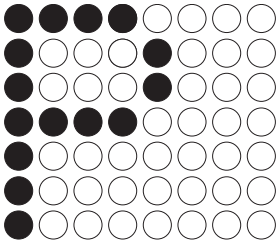

CHR075

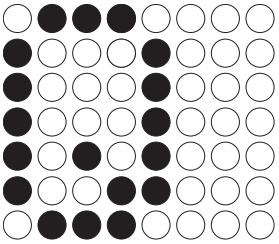

CHR076

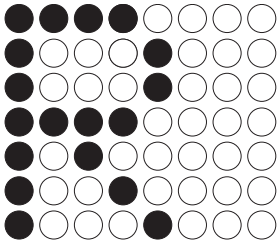

CHR077

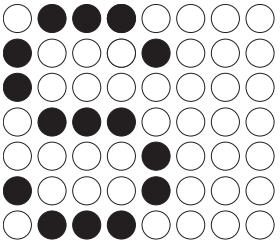

CHR078

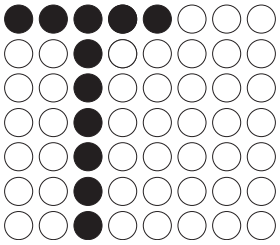

CHR079

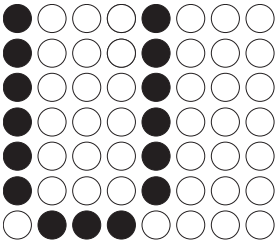

CHR080

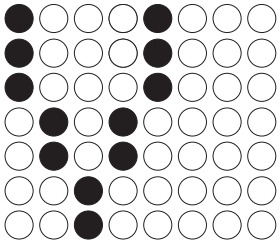

CHR081

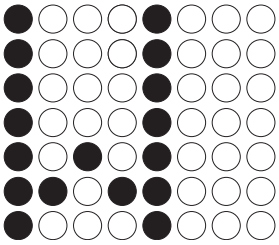

CHR082

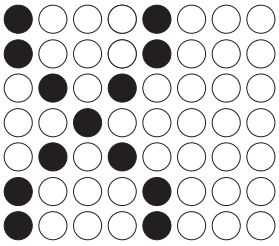

CHR083

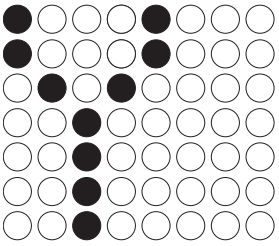

CHR084

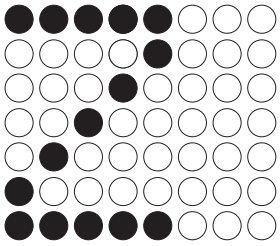

CHR085

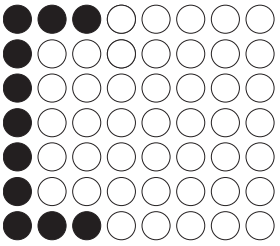

CHR086

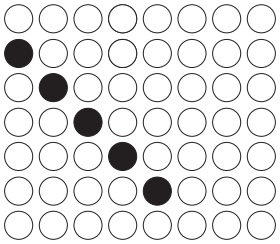

CHR087

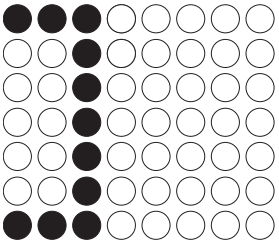

CHR088

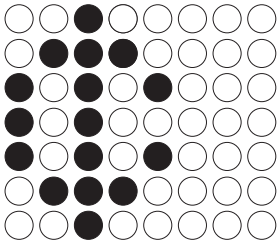

CHR089

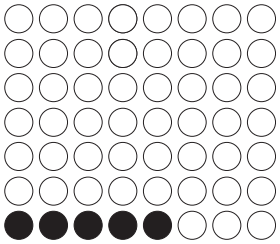

CHR090

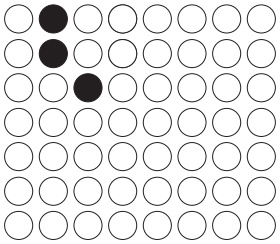

CHR091

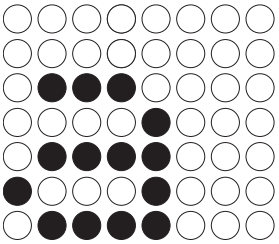

CHR092

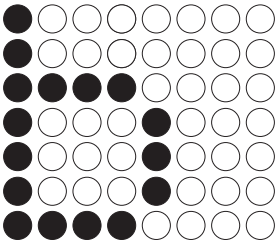

CHR093

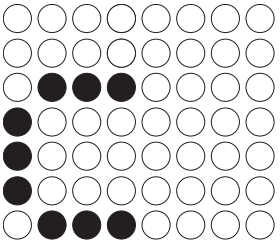

CHR094

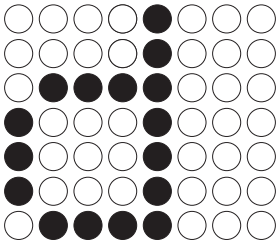

CHR095

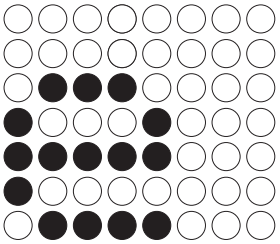

CHR096

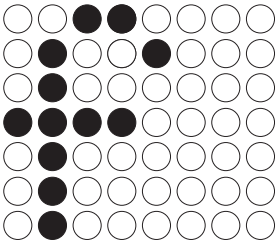

CHR097

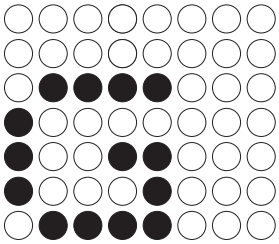

CHR098

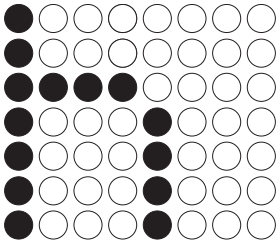

CHR099

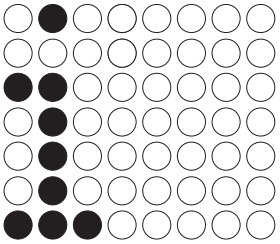

CHR100

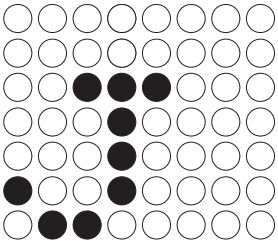

CHR101

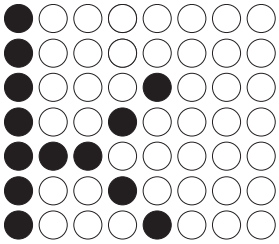

CHR102

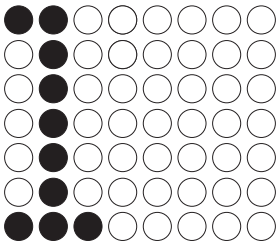

CHR103

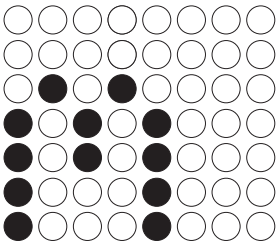

CHR104

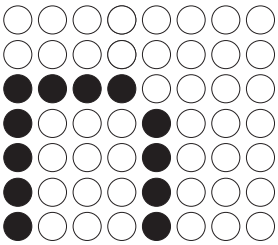

CHR105

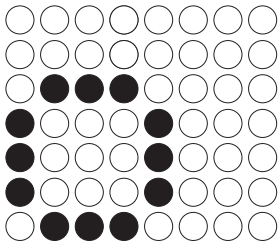

CHR106

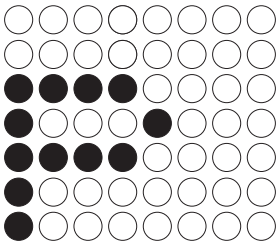

CHR107

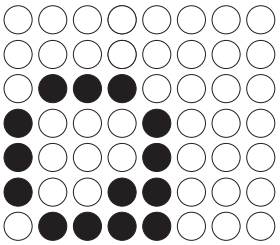

CHR108

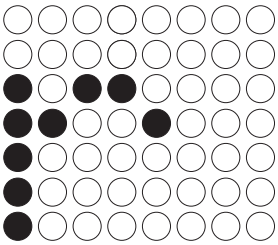

CHR109

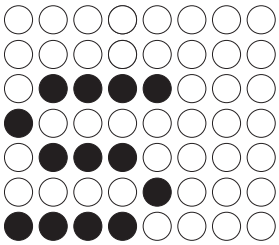

CHR110

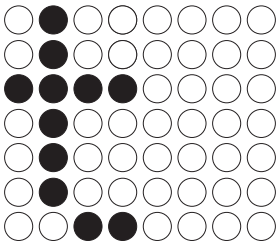

CHR111

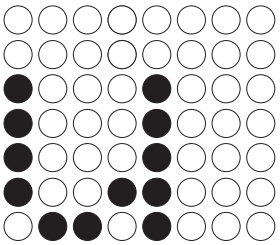

CHR112

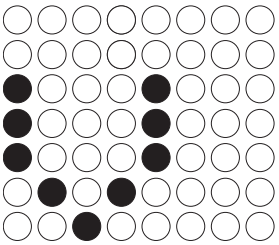

CHR113

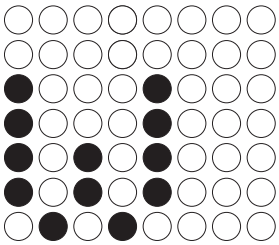

CHR114

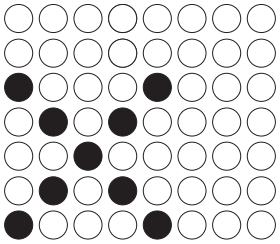

CHR115

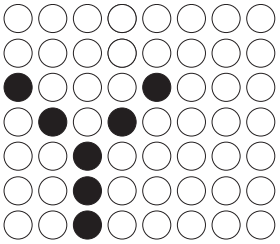

CHR116

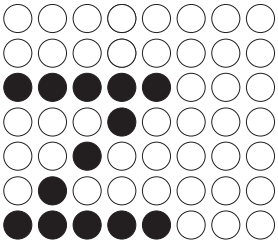

CHR117

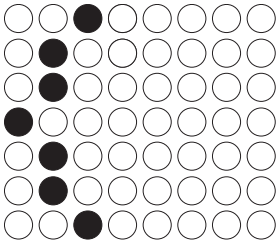

CHR118

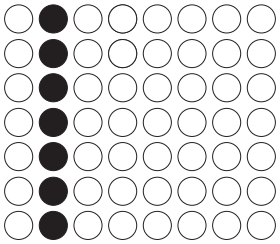

CHR119

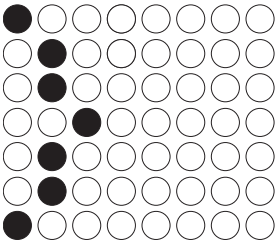

CHR120

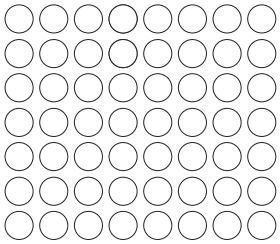

CHR121

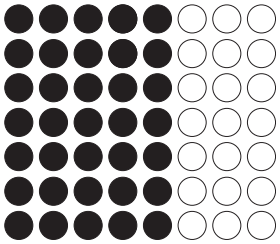

CHR122

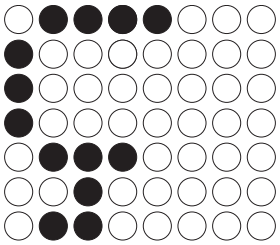

CHR123

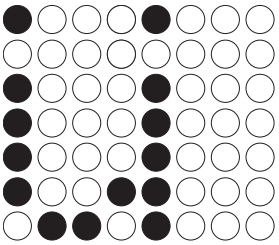

CHR124

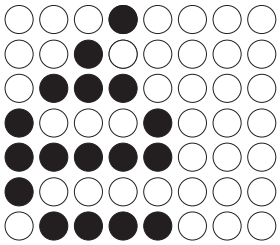

CHR125

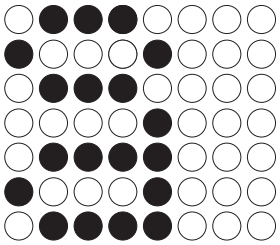

CHR126

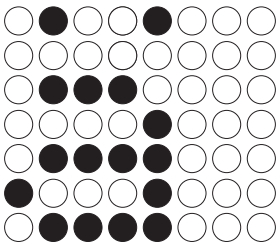

CHR127

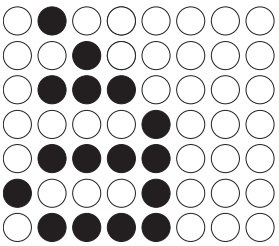

CHR128

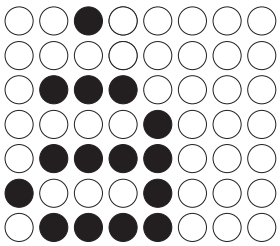

CHR129

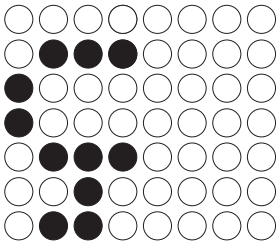

CHR130

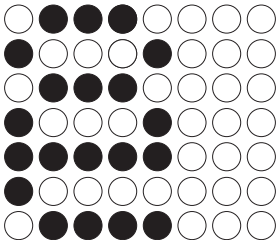

CHR131

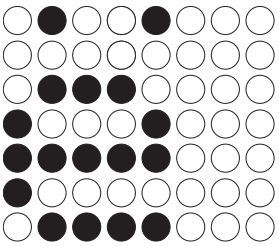

CHR132

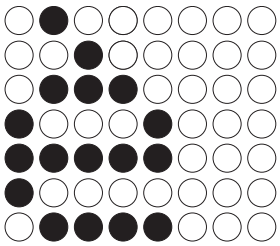

CHR133

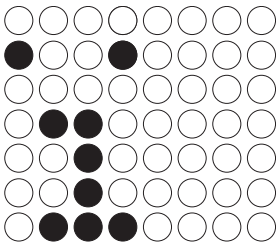

CHR134

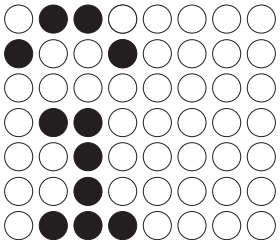

CHR135

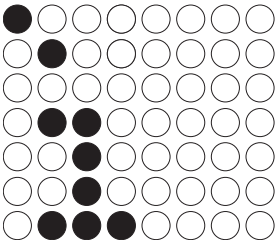

CHR136

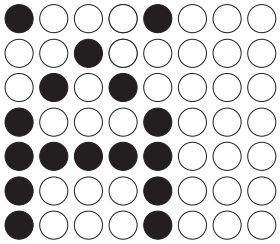

CHR137

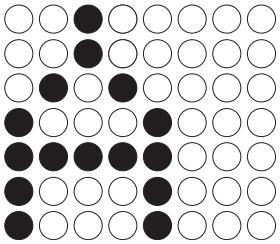

CHR138

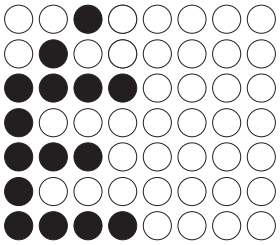

CHR139

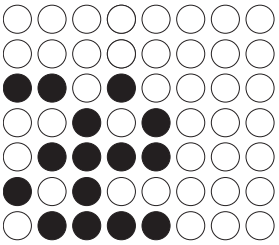

CHR140

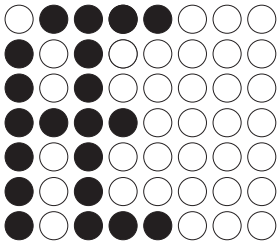

CHR141

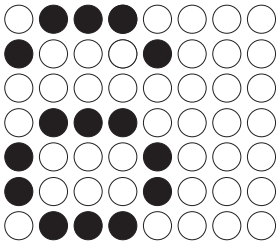

CHR142

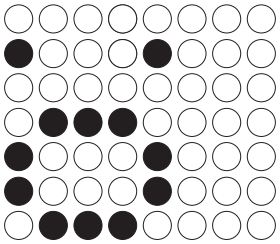

CHR143

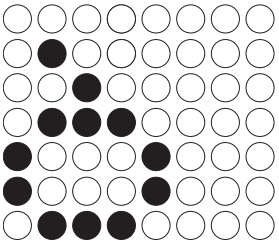

CHR144

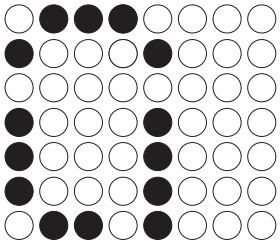

CHR145

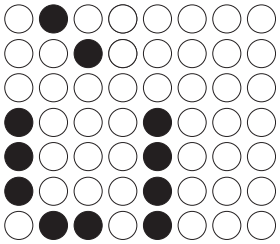

CHR146

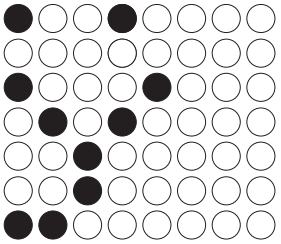

CHR147

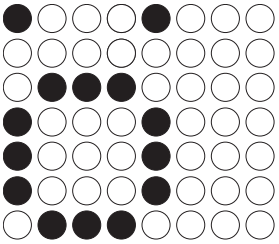

CHR148

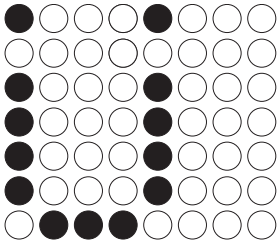

CHR149

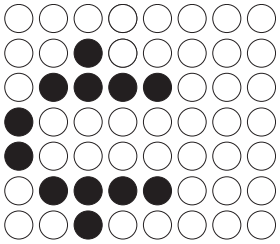

CHR150

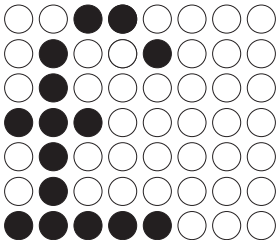

CHR151

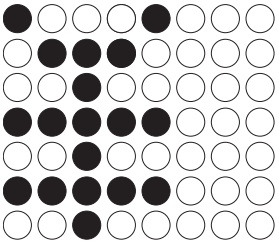

CHR152

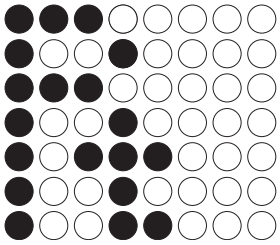

CHR153

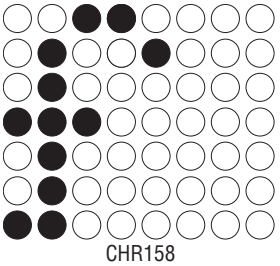

CHR154

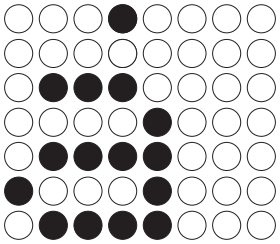

CHR155

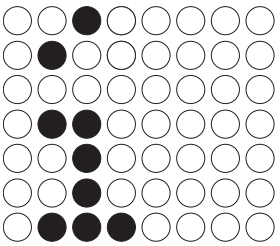

CHR156

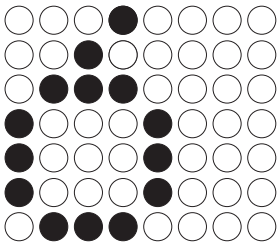

CHR157

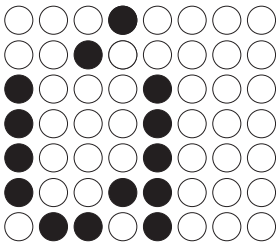

CHR158

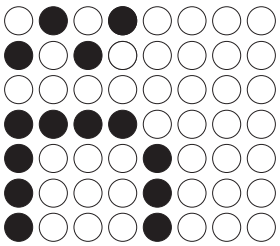

CHR158

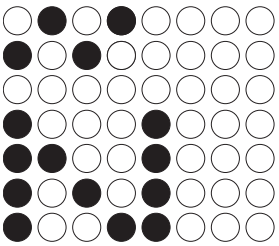

CHR160

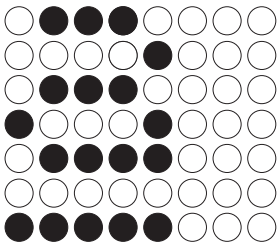

CHR161

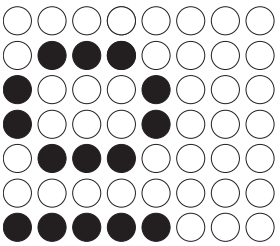

CHR162

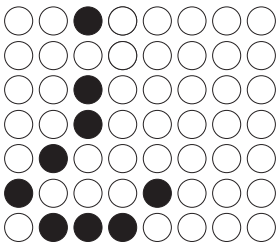

CHR163

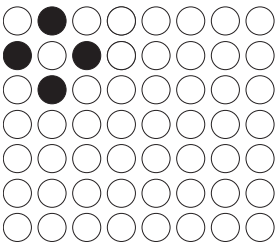

CHR164

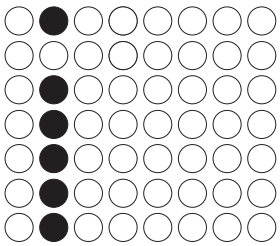

CHR165

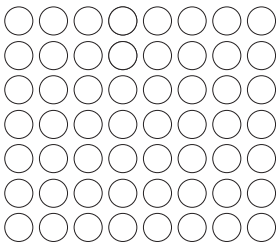

CHR166

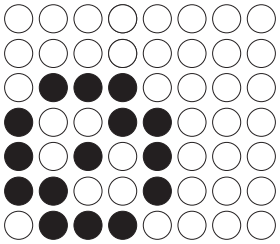

CHR167

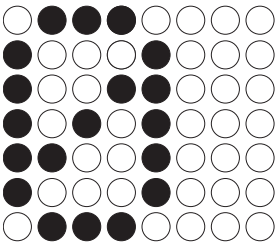

CHR168

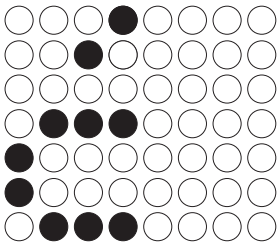

CHR169

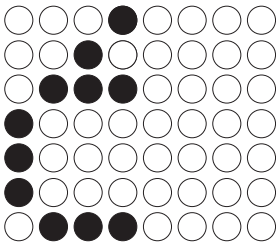

CHR170

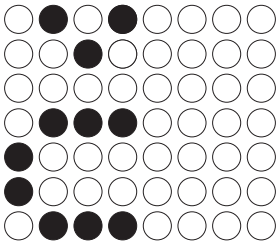

CHR171

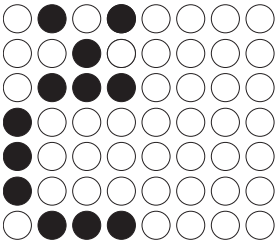

CHR172

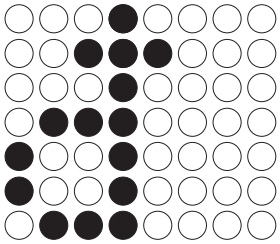

CHR173

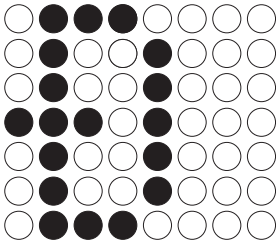

CHR174

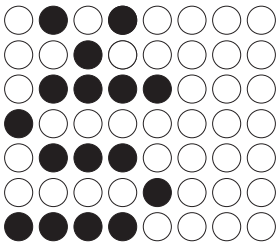

CHR175

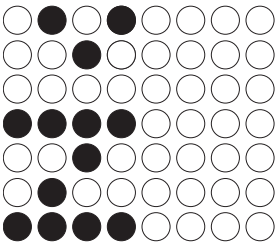

CHR176

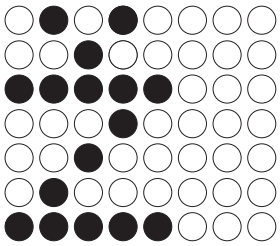

CHR177

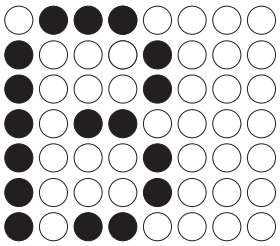

CHR178

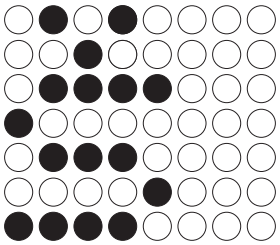

CHR179

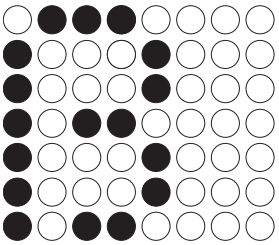

CHR180

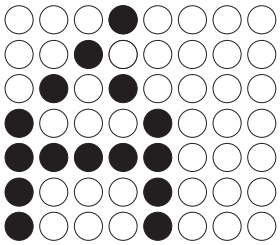

CHR181

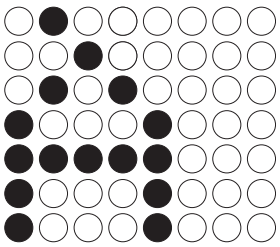

CHR182

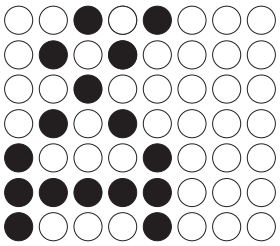

CHR183

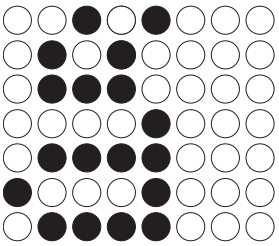

CHR184

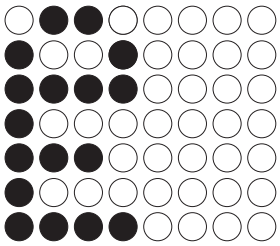

CHR185

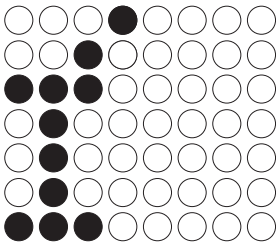

CHR186

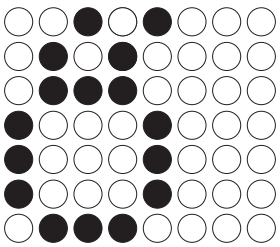

CHR187

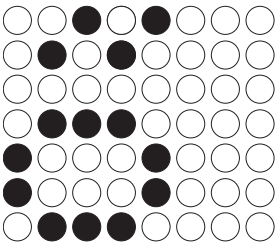

CHR188


CHR189

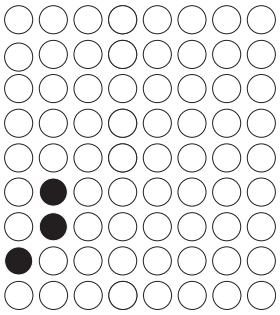

CHR190

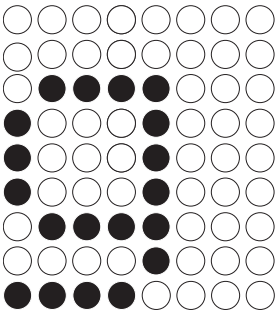

CHR191

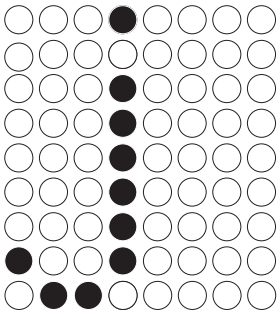

CHR192

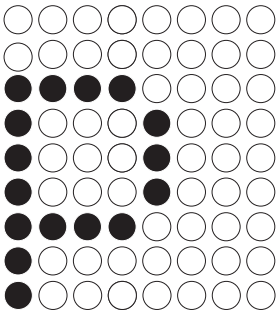

CHR193

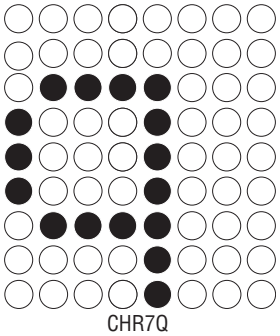
7.13.4 7-High True Descender Regular

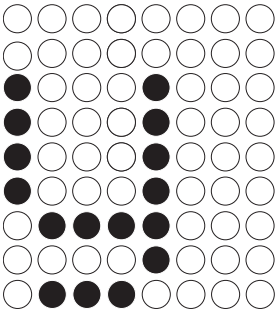

CHR7


CHR7G

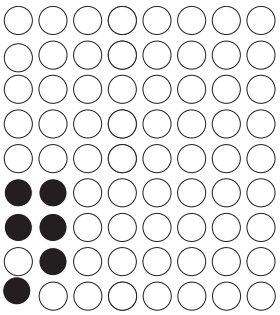

CHR7J

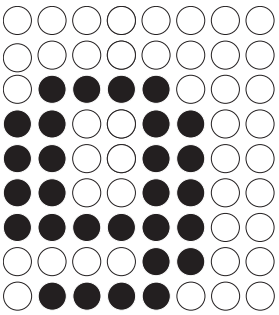

CHR7P

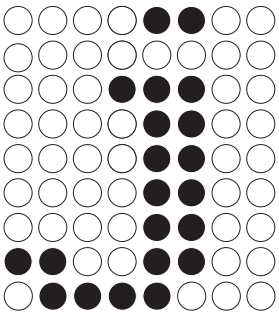

CHR7Q

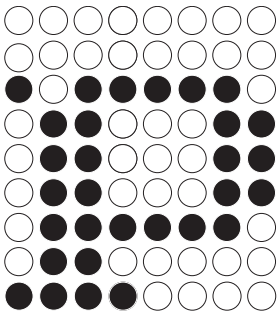

CHR7Y

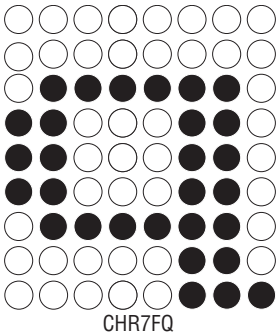
7.13.5 7-High True Descender Fancy

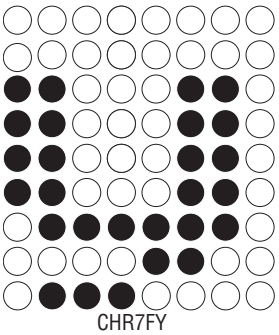

CHR7FC


CHR7FG

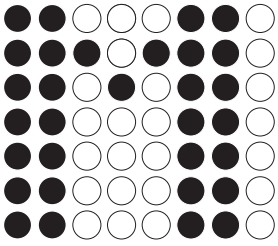

CHR7FJ

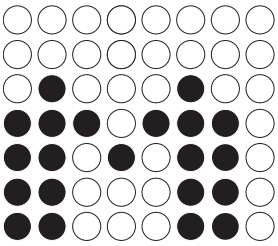

CHR7FP

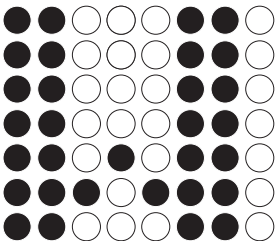

CHR7FQ

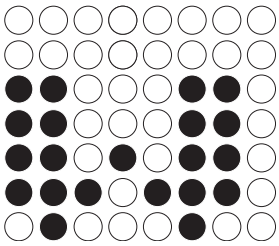

CHR7FY

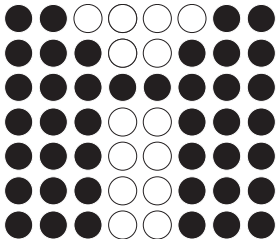
7.13.6 7-High Fat Character

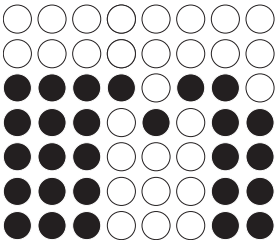

FTSM

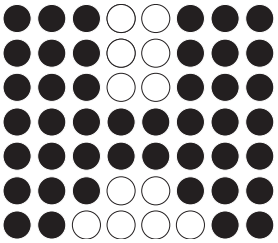

FTSM1

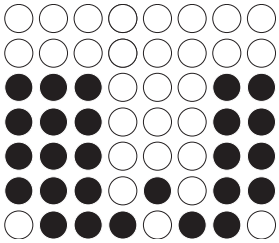

FTSW


FTSW1

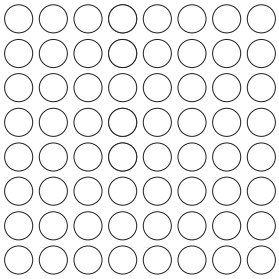

SFTM

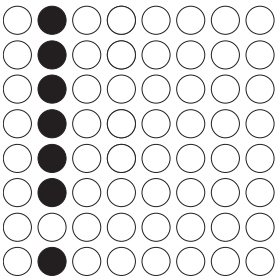

SFTM1

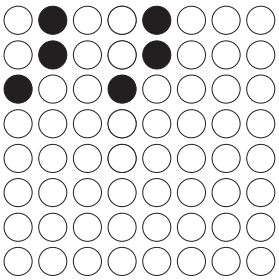

SFTW

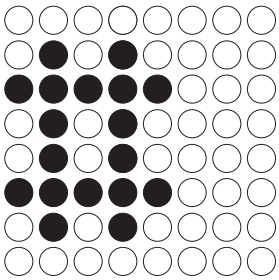

SFTW1

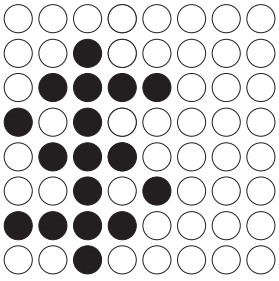
7.13.7 8-High Regular (SS8)

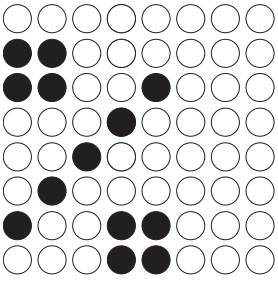

CHR032

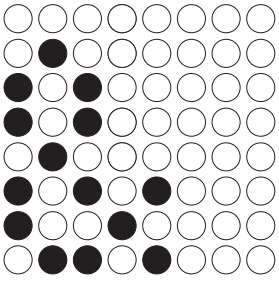

CHR033

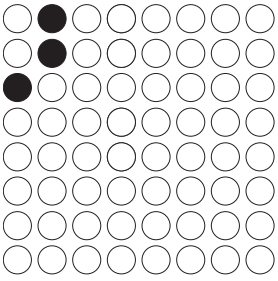

CHR034

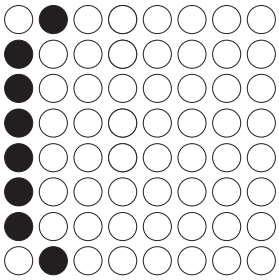

CHR035

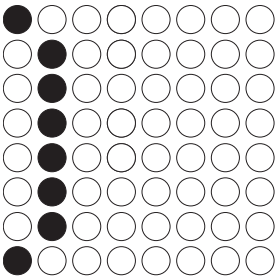

CHR036

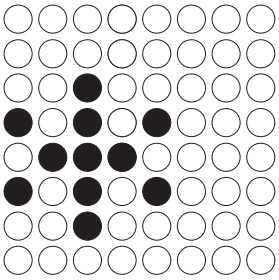

CHR037

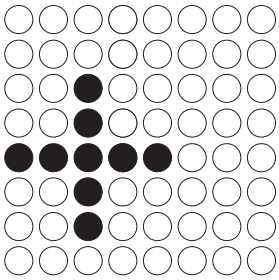

CHR038

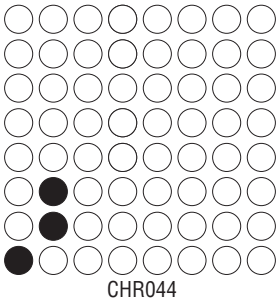

CHR039

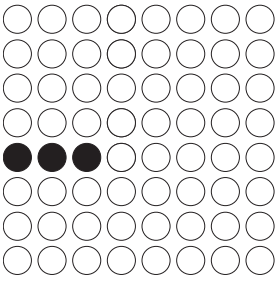

CHR040

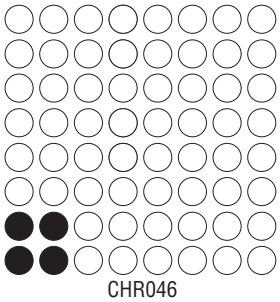

CHR041

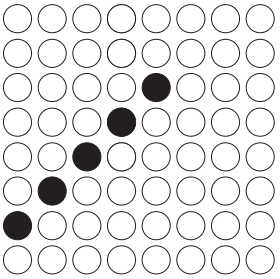

CHR042

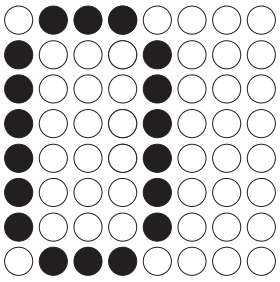

CHR043

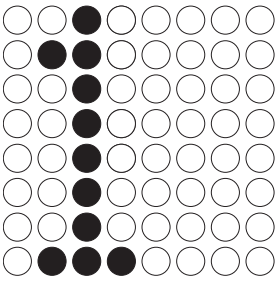

CHR044

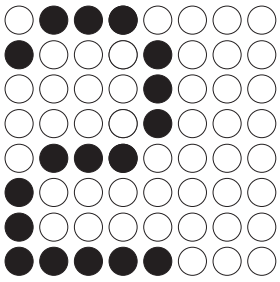

CHR045

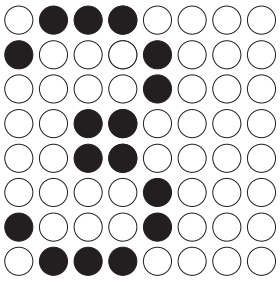

CHR046

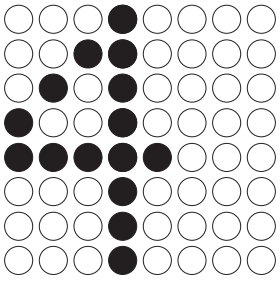

CHR047

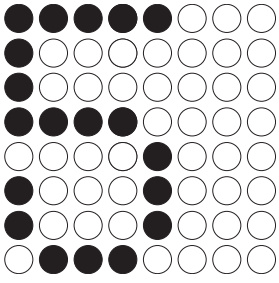

CHR048

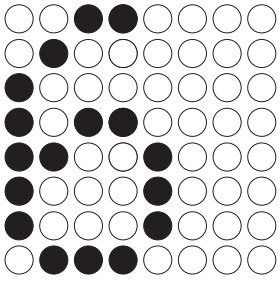

CHR049

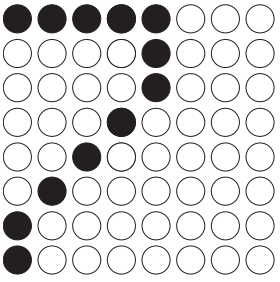

CHR050

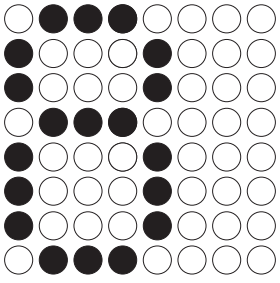

CHR051

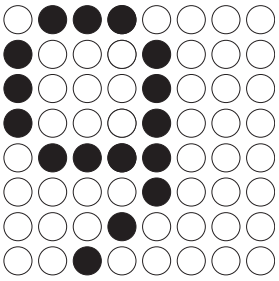

CHR052

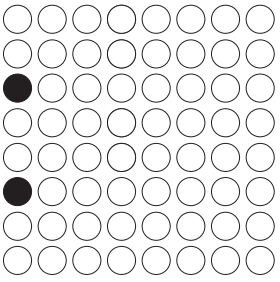

CHR053

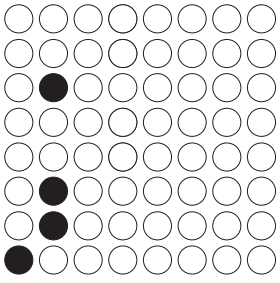

CHR054

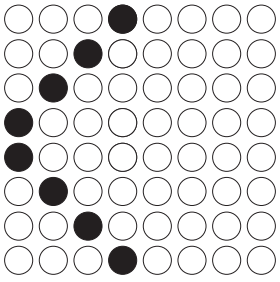

CHR055

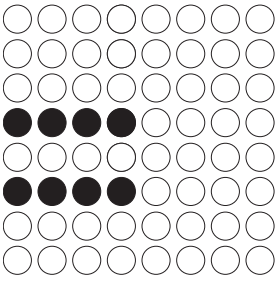

CHR056

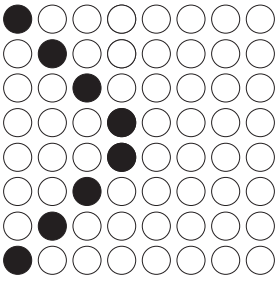

CHR057

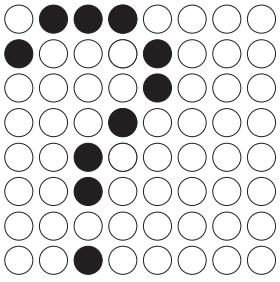

CHR058

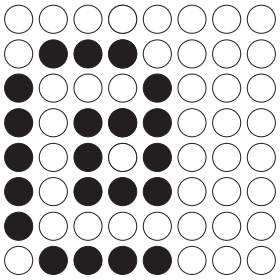

CHR059

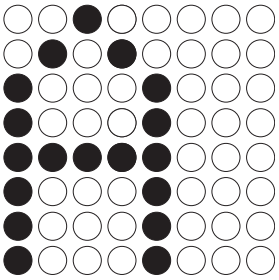

CHR060

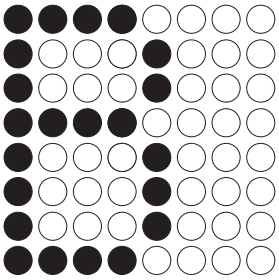

CHR061

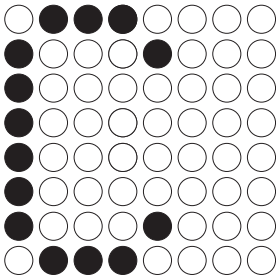

CHR062

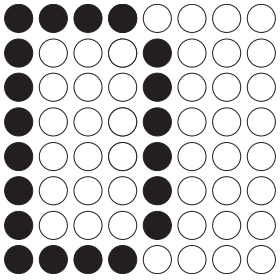

CHR063

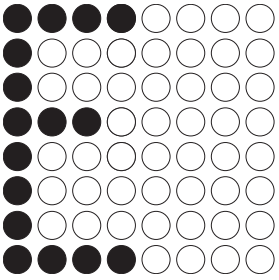

CHR064

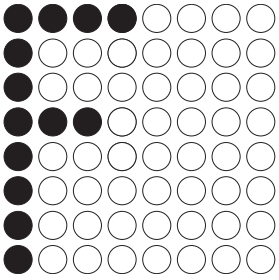

CHR065

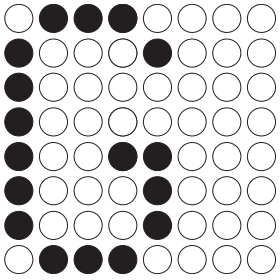

CHR066

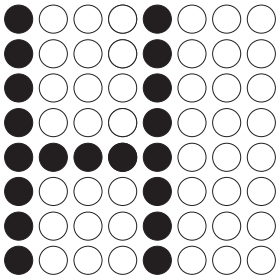

CHR067

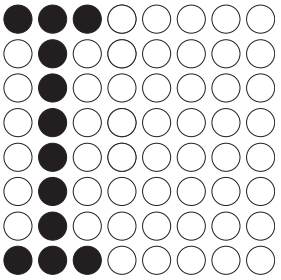

CHR068

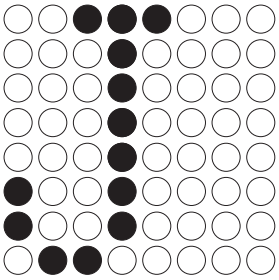

CHR069

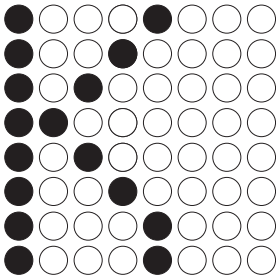

CHR070

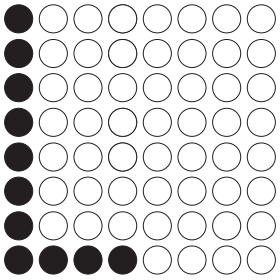

CHR071

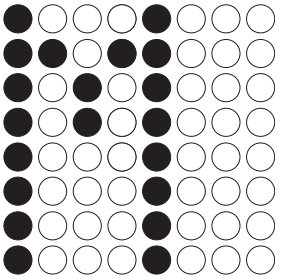

CHR072

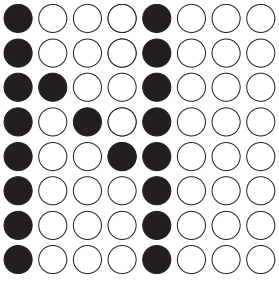

CHR073

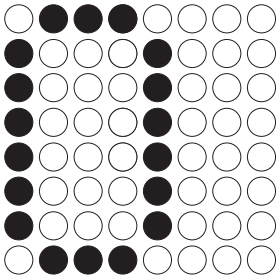

CHR074

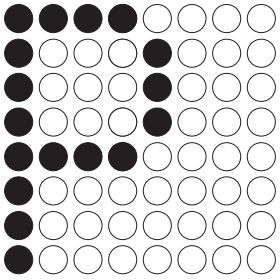

CHR075

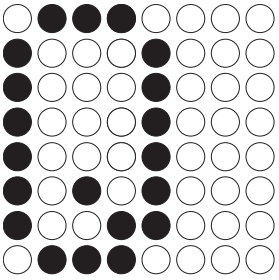

CHR076

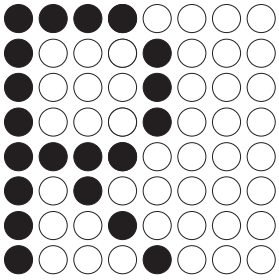

CHR077

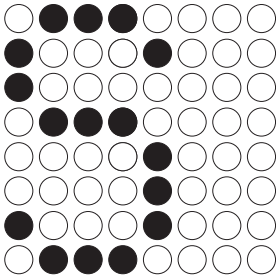

CHR078

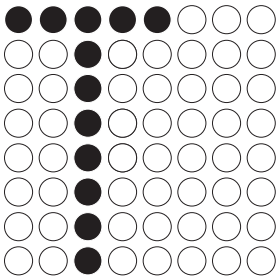

CHR079

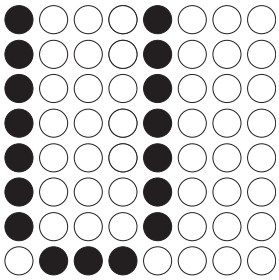

CHR080

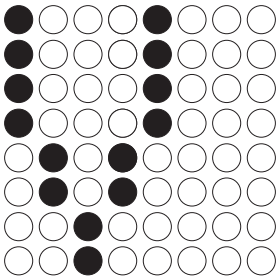

CHR081

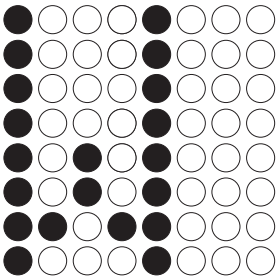

CHR082

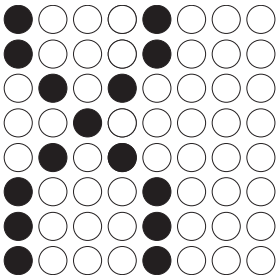

CHR083

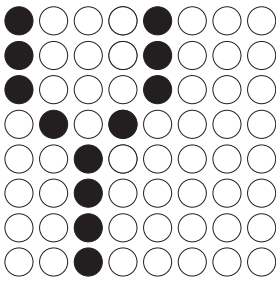

CHR084

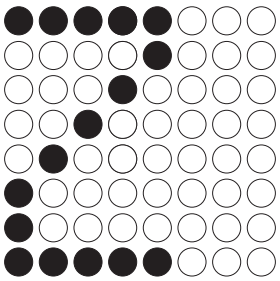

CHR085

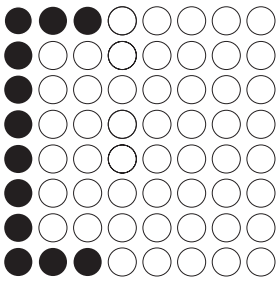

CHR086

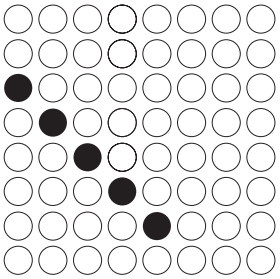

CHR087

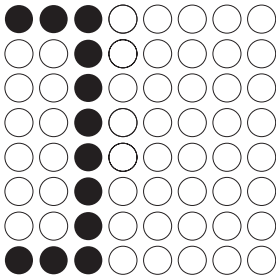

CHR088

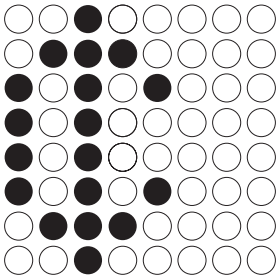

CHR089

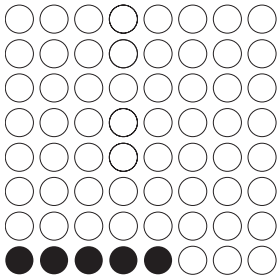

CHR090

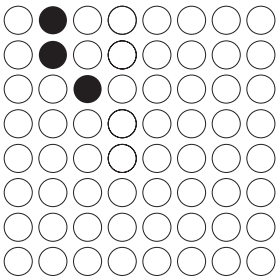

CHR091

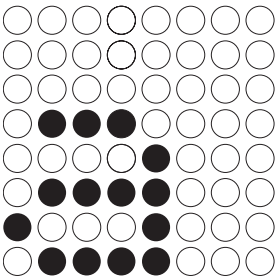

CHR092

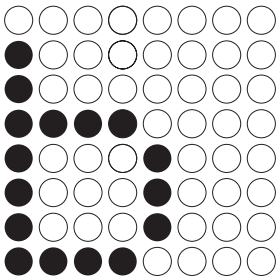

CHR093

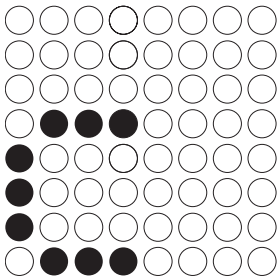

CHR094

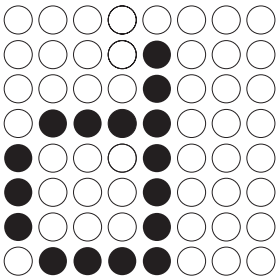

CHR095

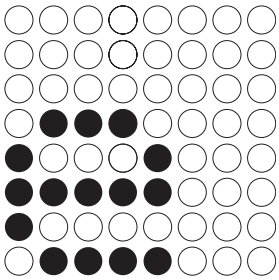

CHR096

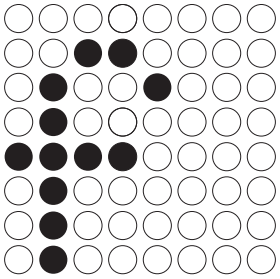

CHR097

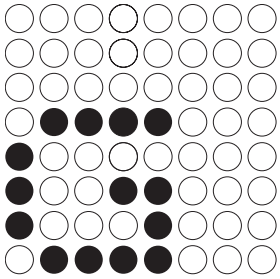

CHR098

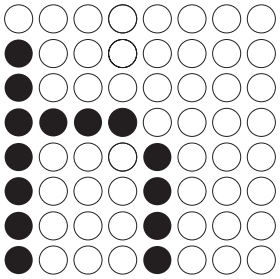

CHR099

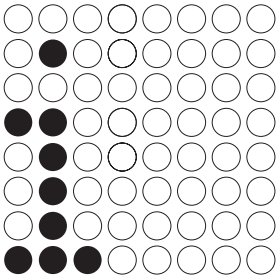

CHR100

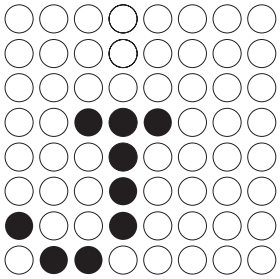

CHR101

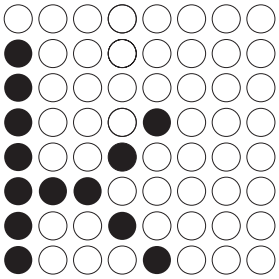

CHR102

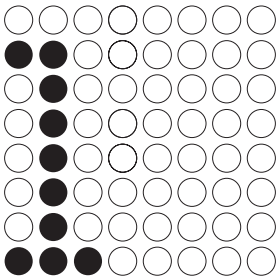

CHR103

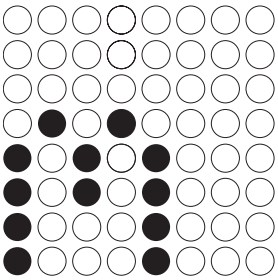

CHR104

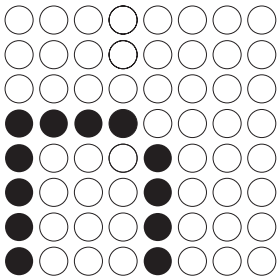

CHR105

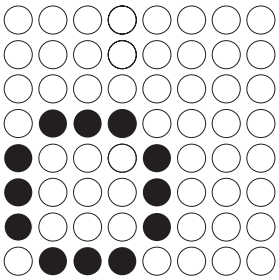

CHR106

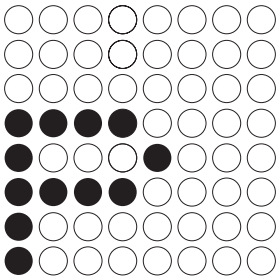

CHR107

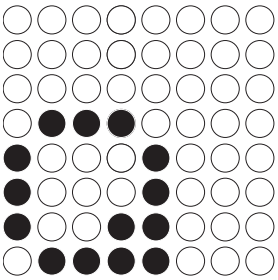

CHR108

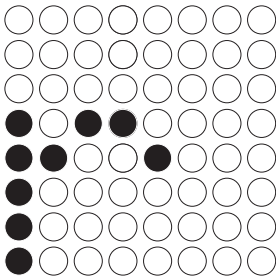

CHR109

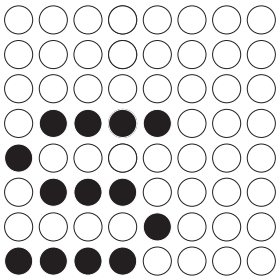

CHR110

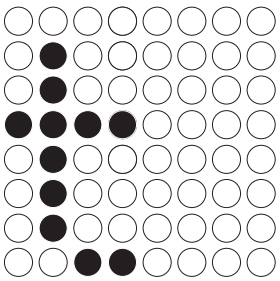

CHR111

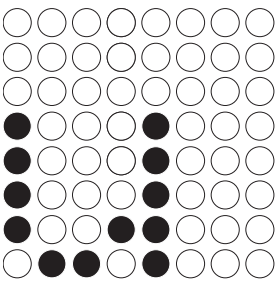

CHR112

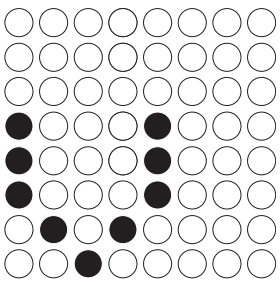

CHR113

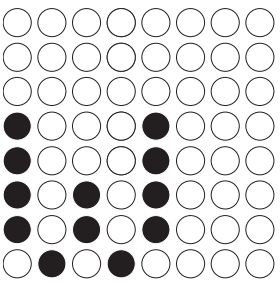

CHR114

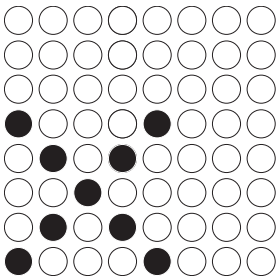

CHR115

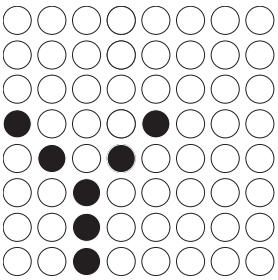

CHR116

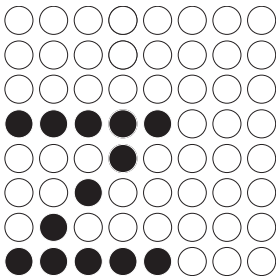

CHR117

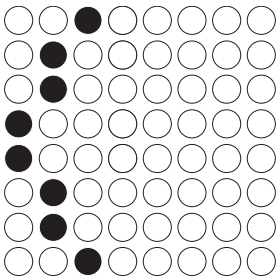

CHR118

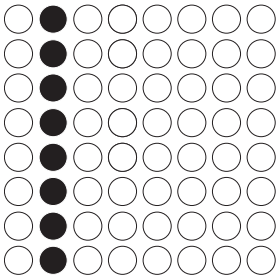

CHR119

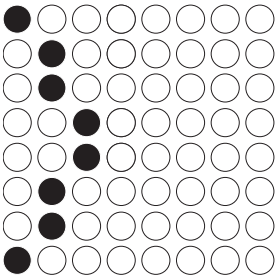

CHR120

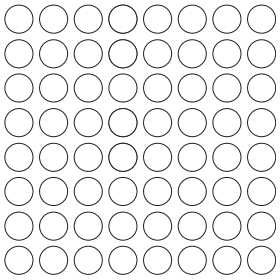

CHR121

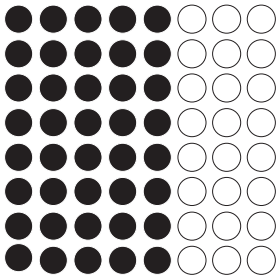

CHR122

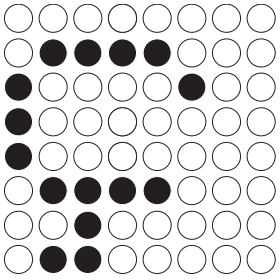

CHR123

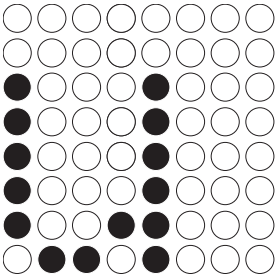

CHR124

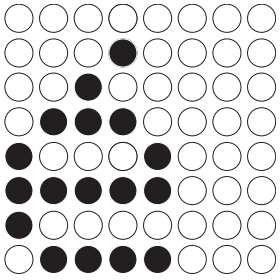

CHR125

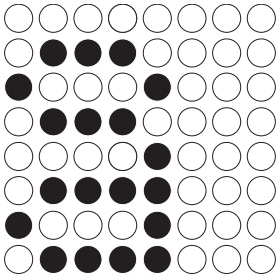

CHR126

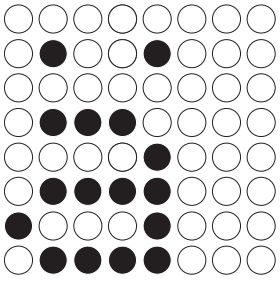

CHR127

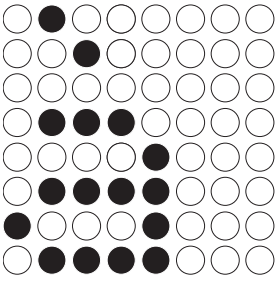

CHR128

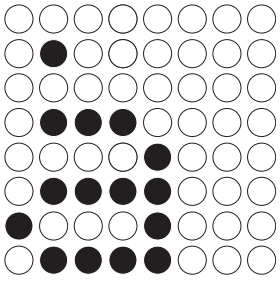

CHR129

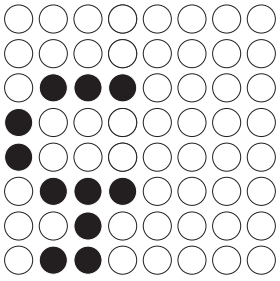

CHR130

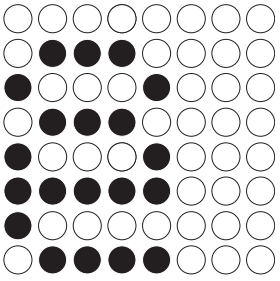

CHR131

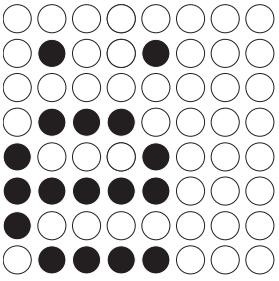

CHR132

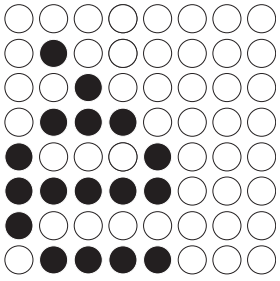

CHR133

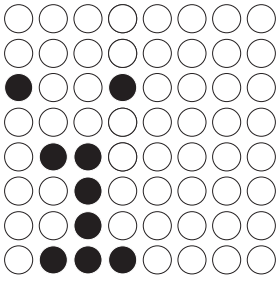

CHR134

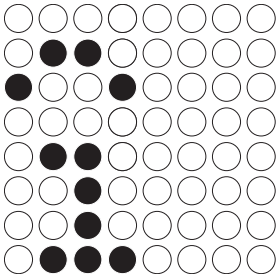

CHR135

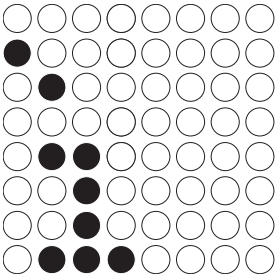

CHR136

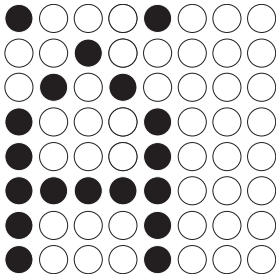

CHR137

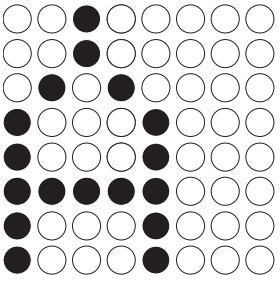

CHR138

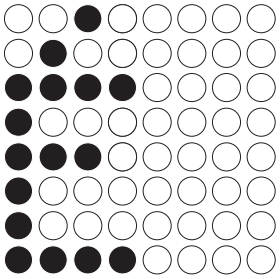

CHR139

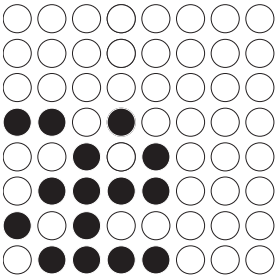

CHR140

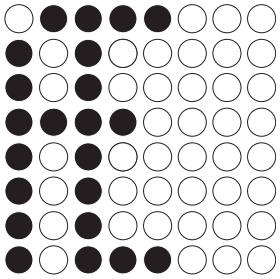

CHR141

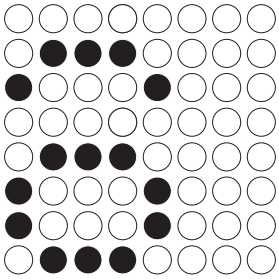

CHR142

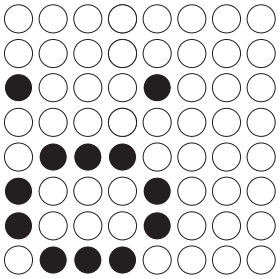

CHR143

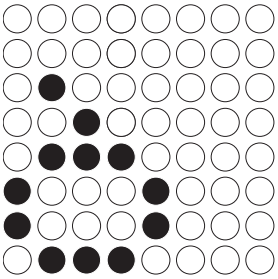

CHR144

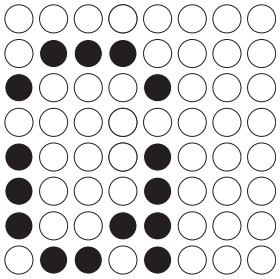

CHR145

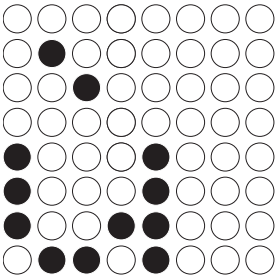

CHR146

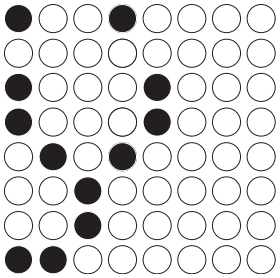

CHR147

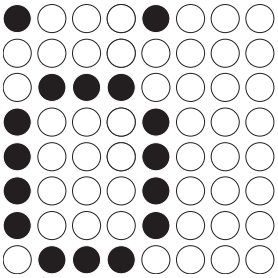

CHR148

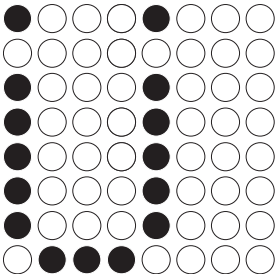

CHR149

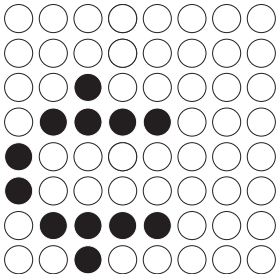

CHR150

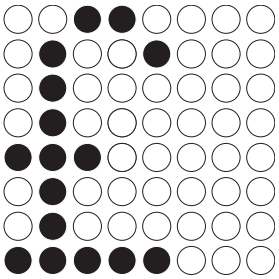

CHR151

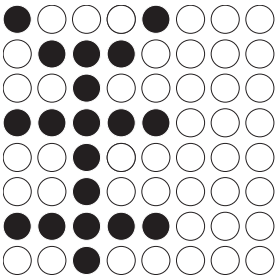

CHR152

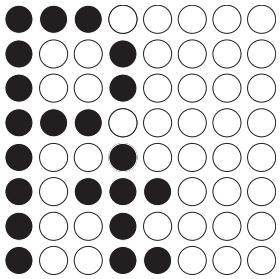

CHR153

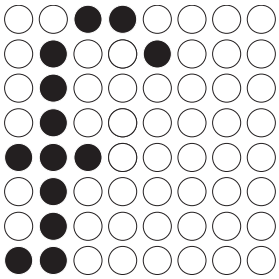

CHR154

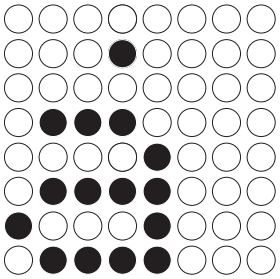

CHR155

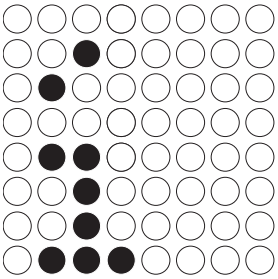

CHR156

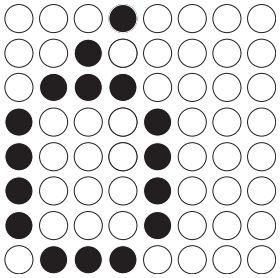

CHR157

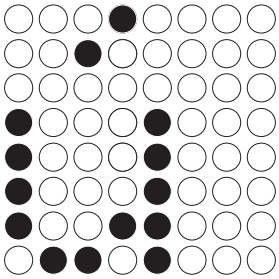

CHR158

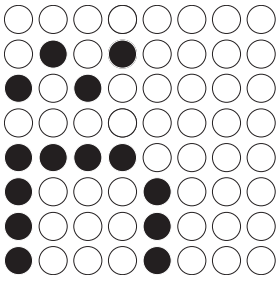

CHR159

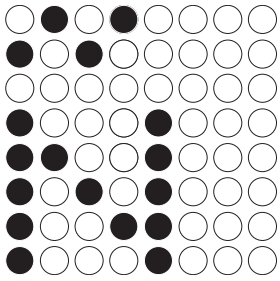

CHR160

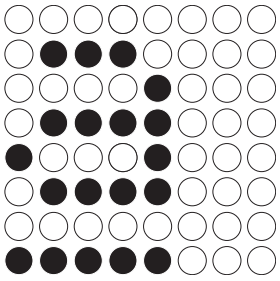

CHR161

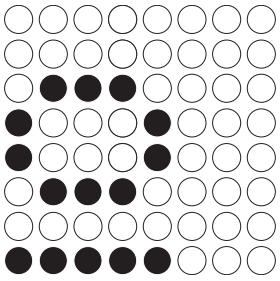

CHR162

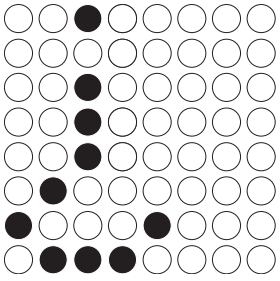

CHR163

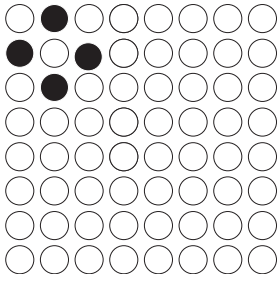

CHR164

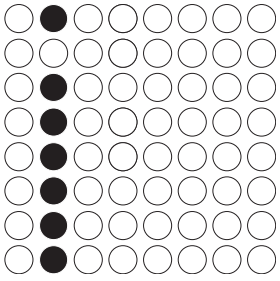

CHR165

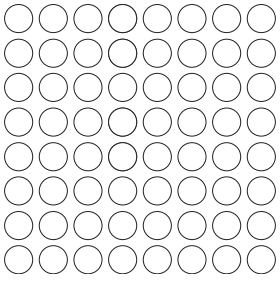

CHR166

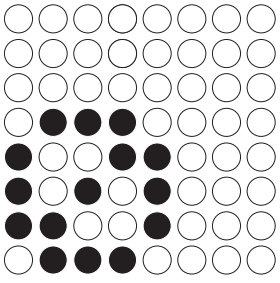

CHR167

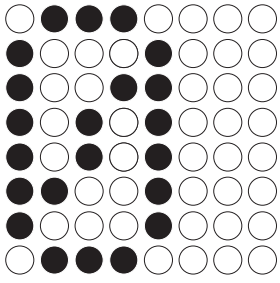

CHR168

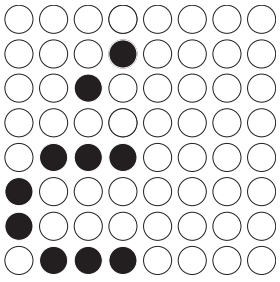

CHR169

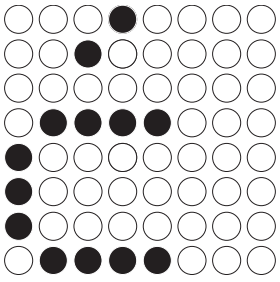

CHR170

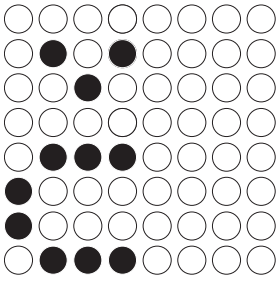

CHR171

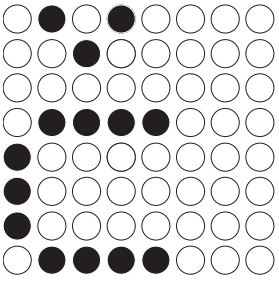

CHR172

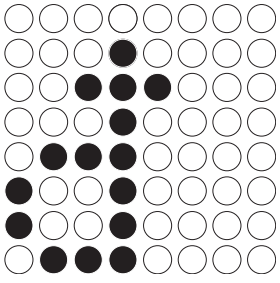

CHR173

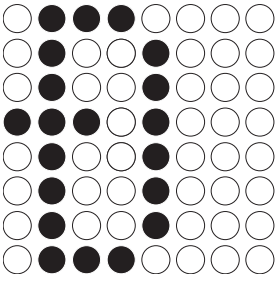

CHR174

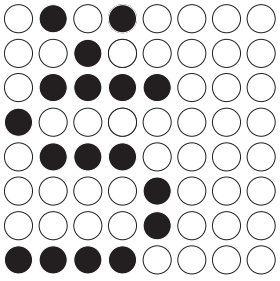

CHR175

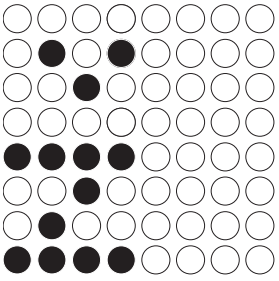

CHR176

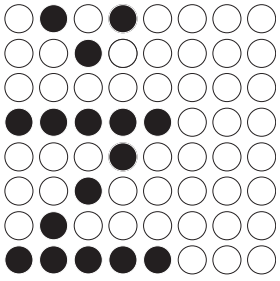

CHR177

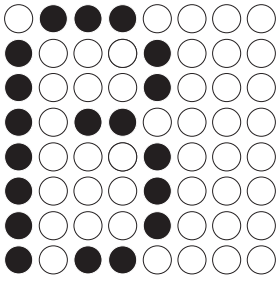

CHR178

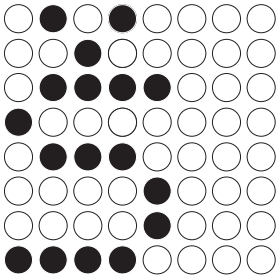

CHR179

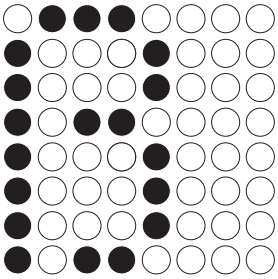

CHR180

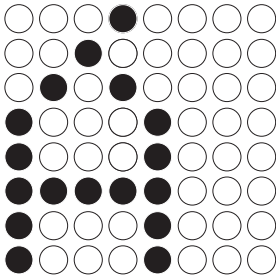

CHR181

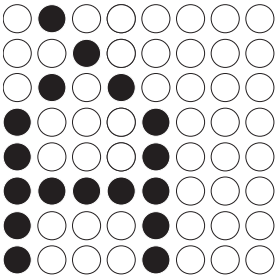

CHR182

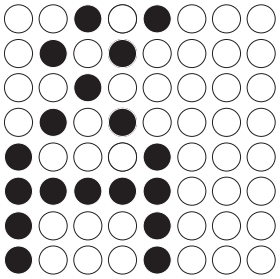

CHR183

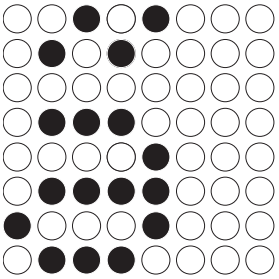

CHR184

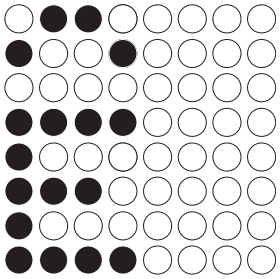

CHR185

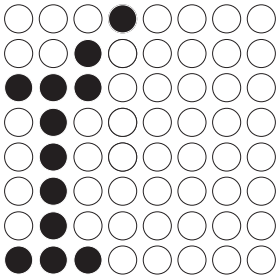

CHR186

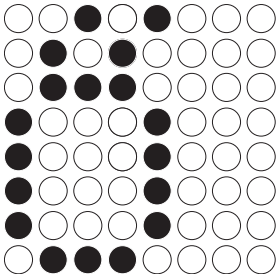

CHR187

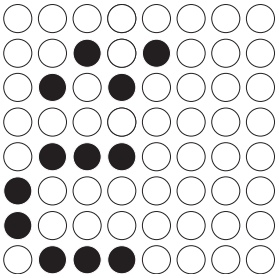

CHR188


CHR189

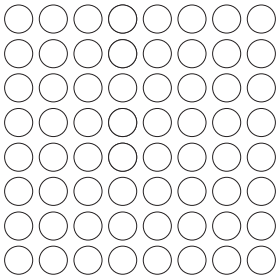

CHR190

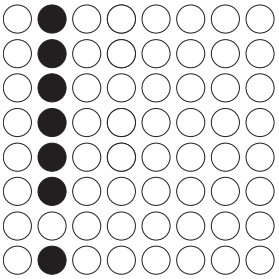

CHR191

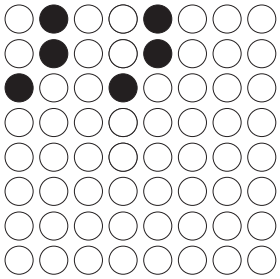

CHR192

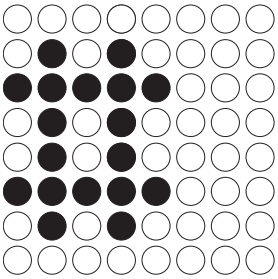

CHR193

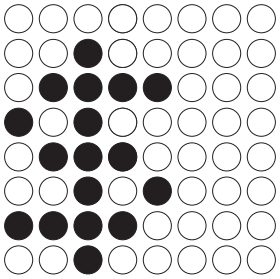
7.13.8 8-High Fancy (SF8)

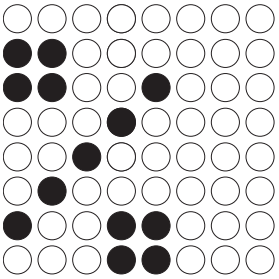

CHR032

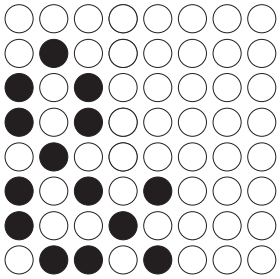

CHR033

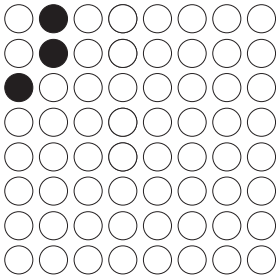

CHR034

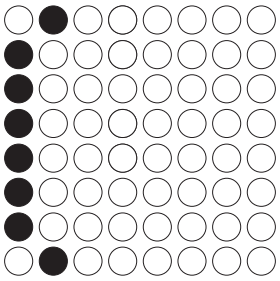

CHR035

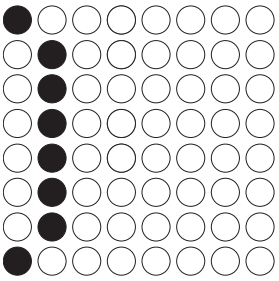

CHR036

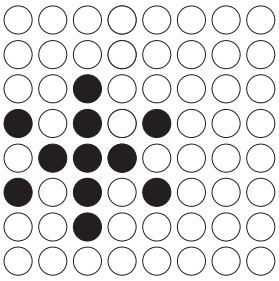

CHR037

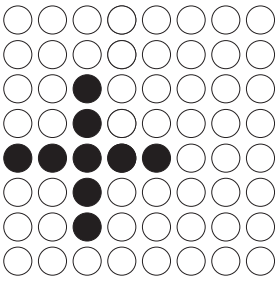

CHR038

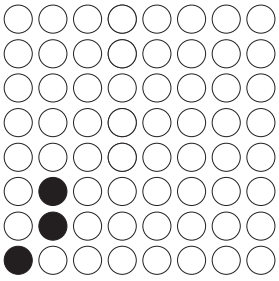

CHR039

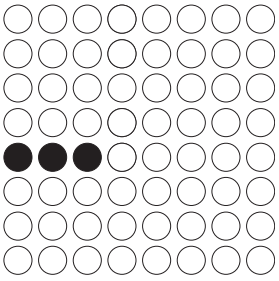

CHR040

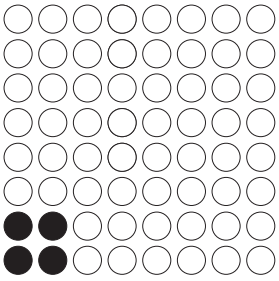

CHR041

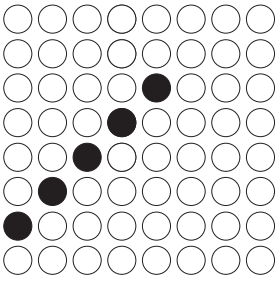

CHR042

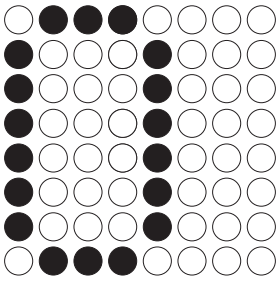

CHR043

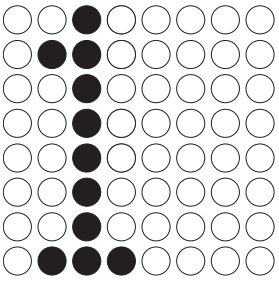

CHR044

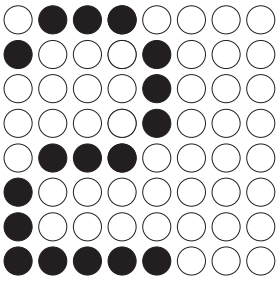

CHR045

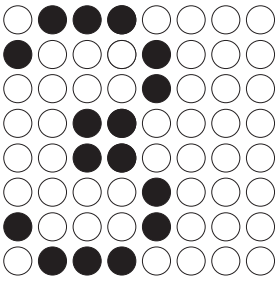

CHR046

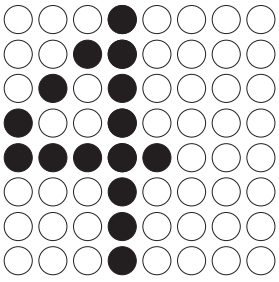

CHR047

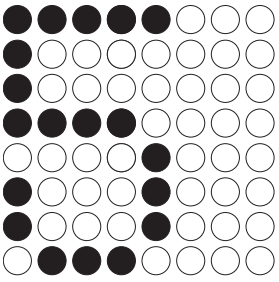

CHR048

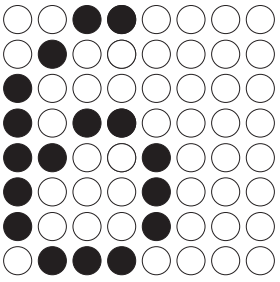

CHR049

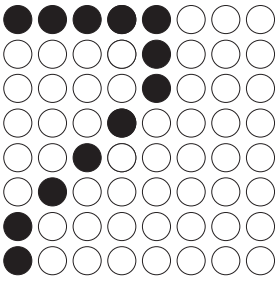

CHR050

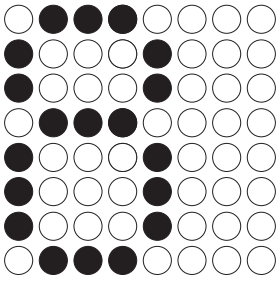

CHR051

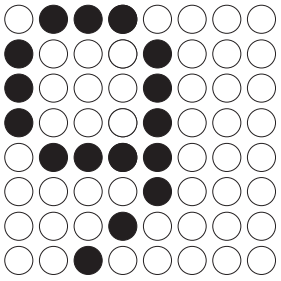

CHR052

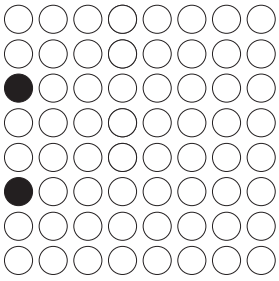

CHR053

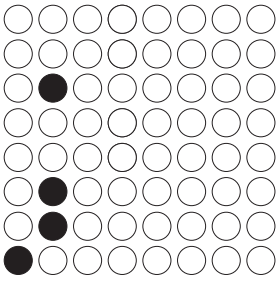

CHR054

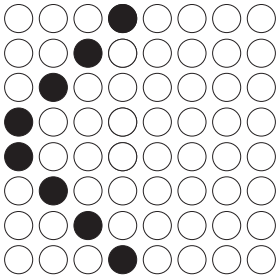

CHR055

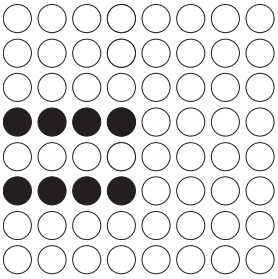

CHR056

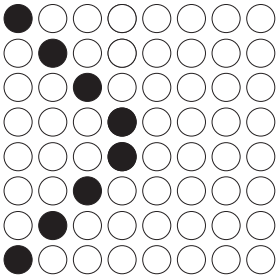

CHR057

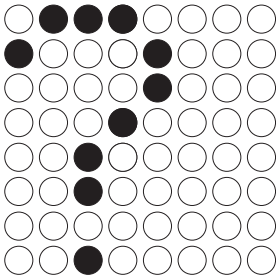

CHR058

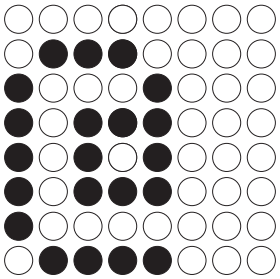

CHR059

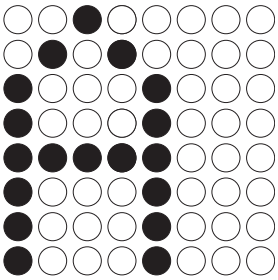

CHR060

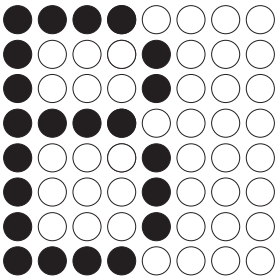

CHR061

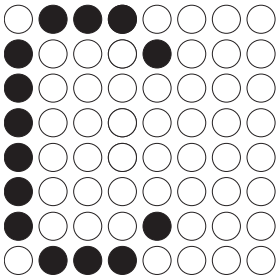

CHR062

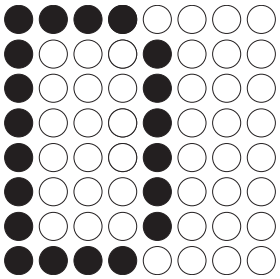

CHR063

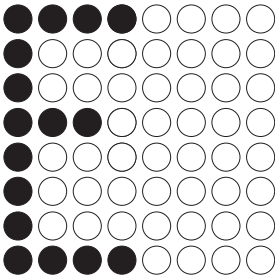

CHR064

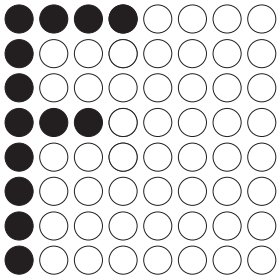

CHR065

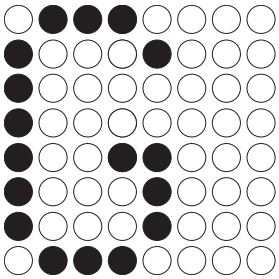

CHR066

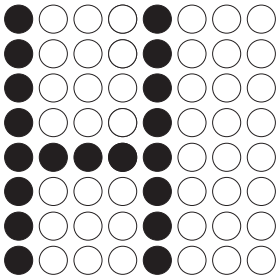

CHR067

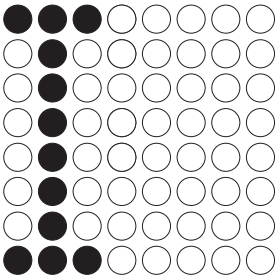

CHR068

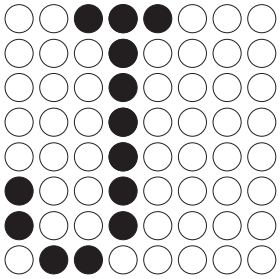

CHR069

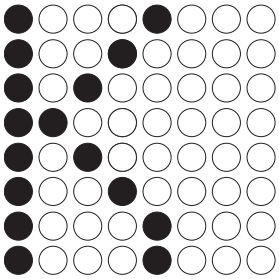

CHR070

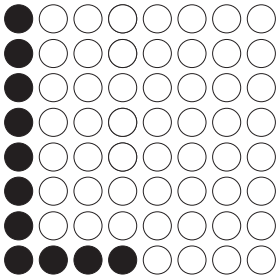

CHR071

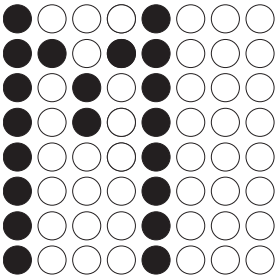

CHR072

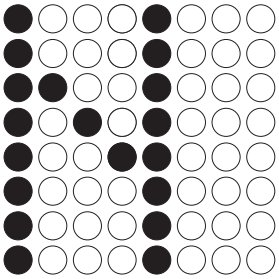

CHR073

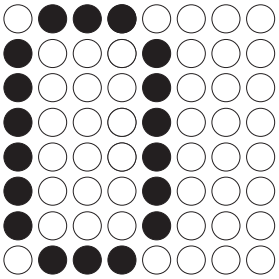

CHR074

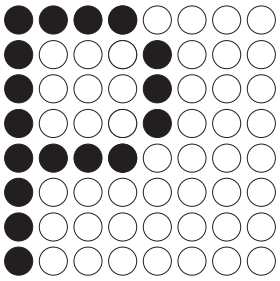

CHR075

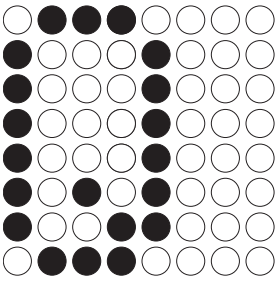

CHR076

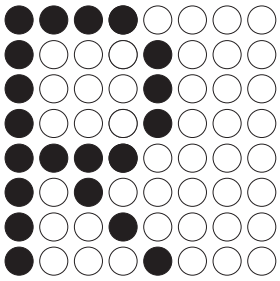

CHR077

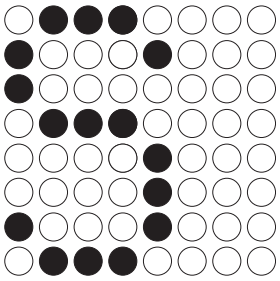

CHR078

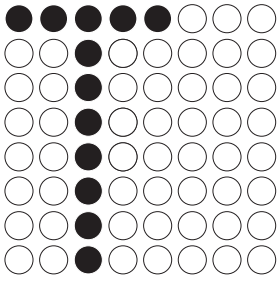

CHR079

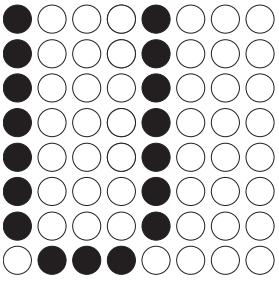

CHR080

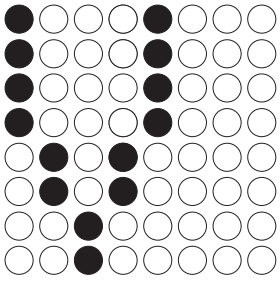

CHR081

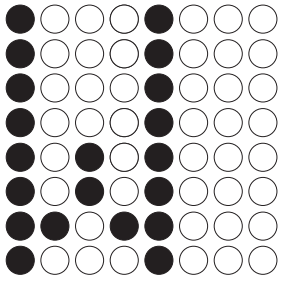

CHR082

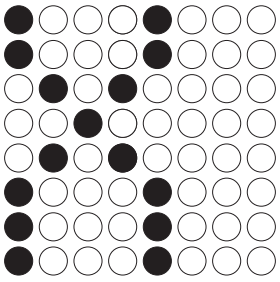

CHR083

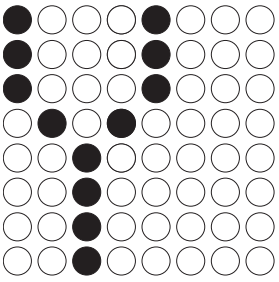

CHR084

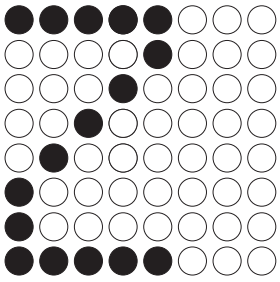

CHR085

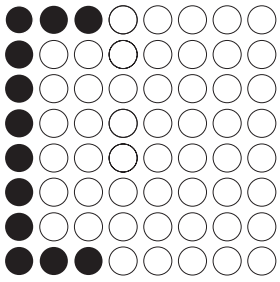

CHR086

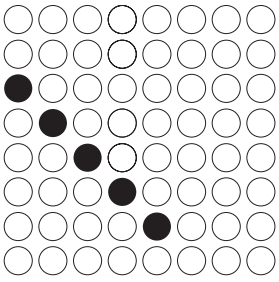

CHR087

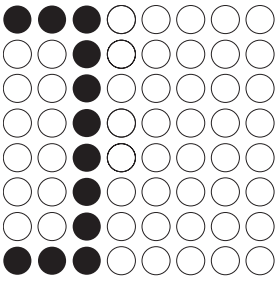

CHR088

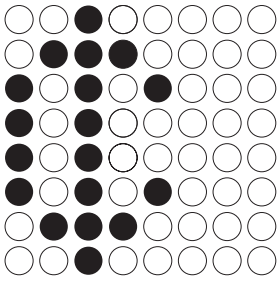

CHR089

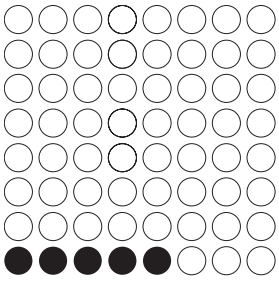

CHR090

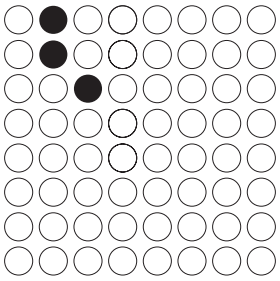

CHR091

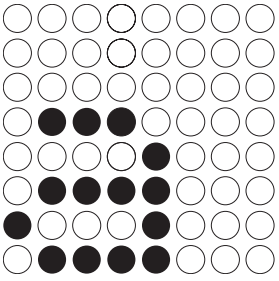

CHR092

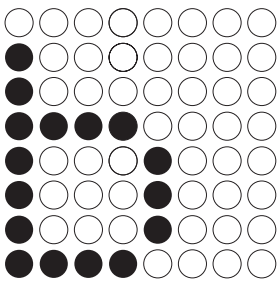

CHR093

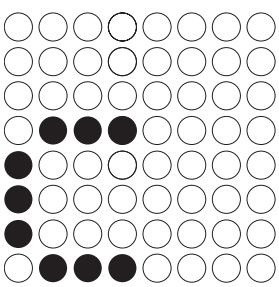

CHR094

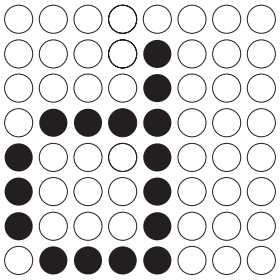

CHR095

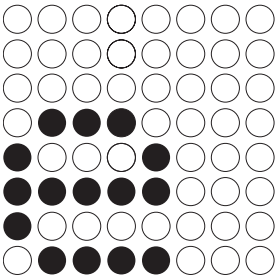

CHR096

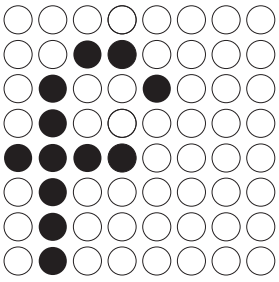

CHR097

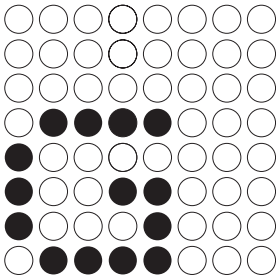

CHR098

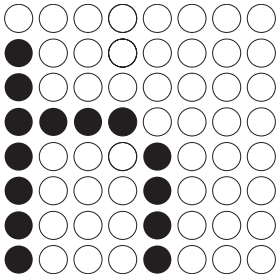

CHR099

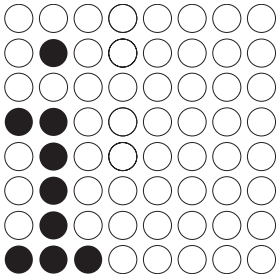

CHR100

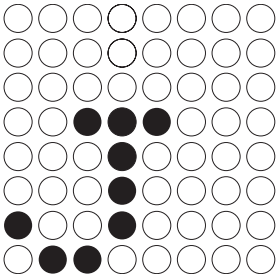

CHR101

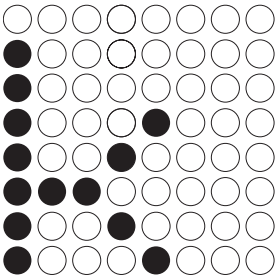

CHR102

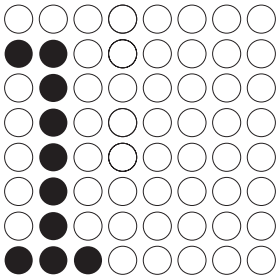

CHR103

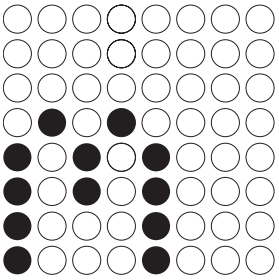

CHR104

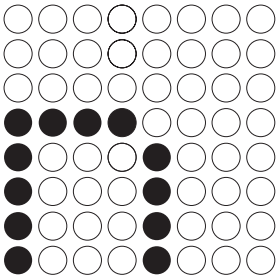

CHR105

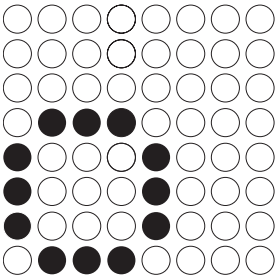

CHR106

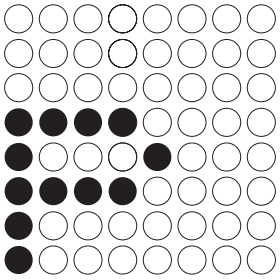

CHR107

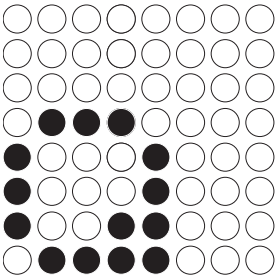

CHR108

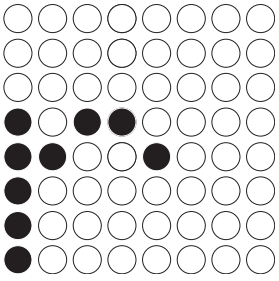

CHR109

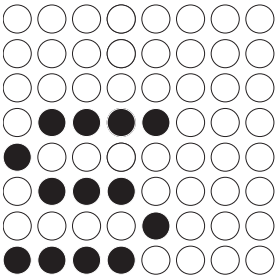

CHR110

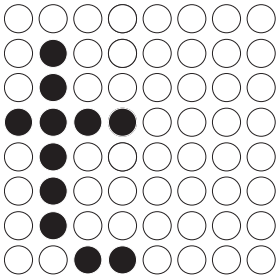

CHR111

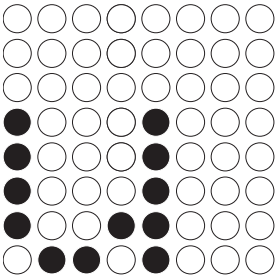

CHR112

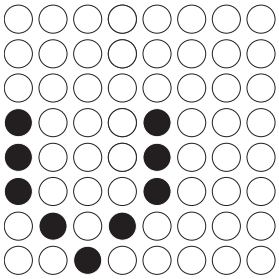

CHR113

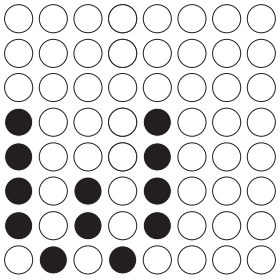

CHR114

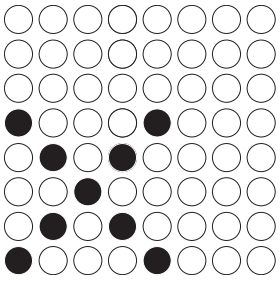

CHR115

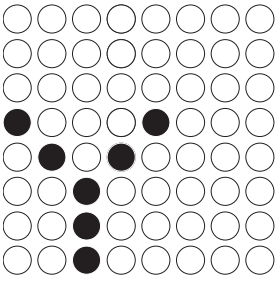

CHR116

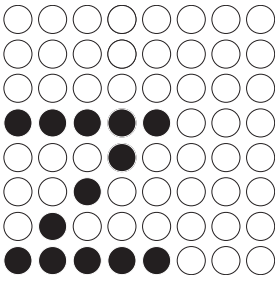

CHR117

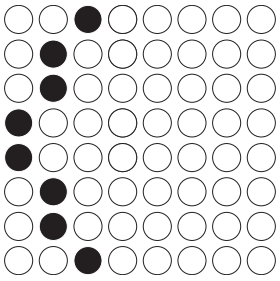

CHR118

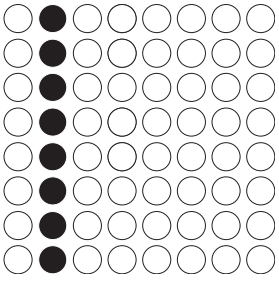

CHR119

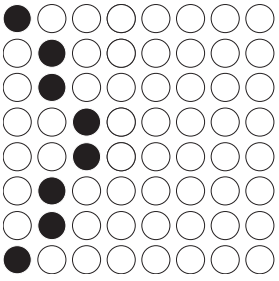

CHR120

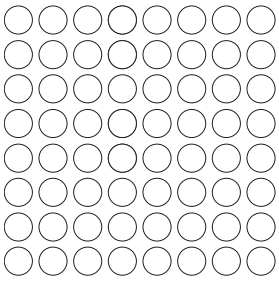

CHR121

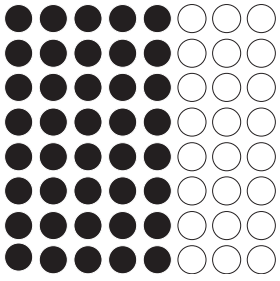

CHR122

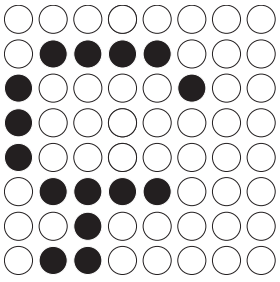

CHR123

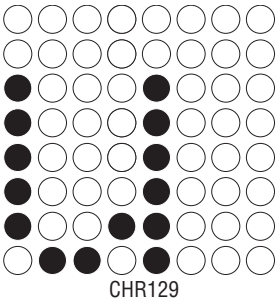

CHR124

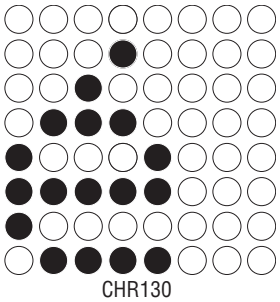

CHR125

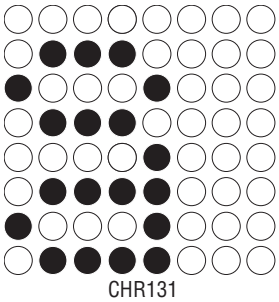

CHR126

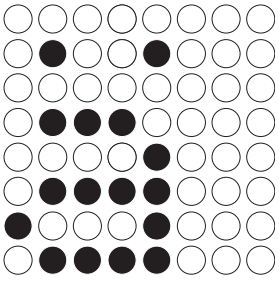

CHR127

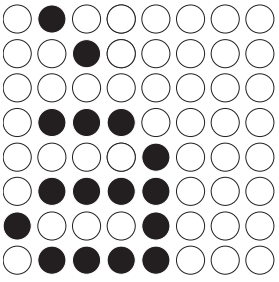

CHR128

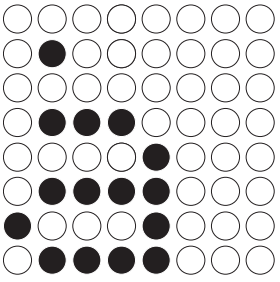

CHR129

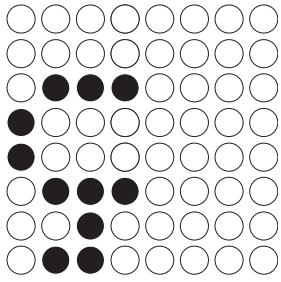

CHR130

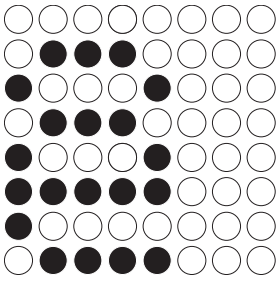

CHR131

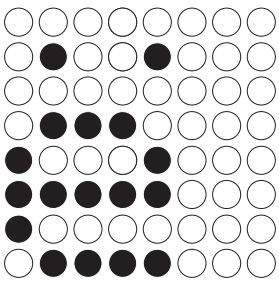

CHR132

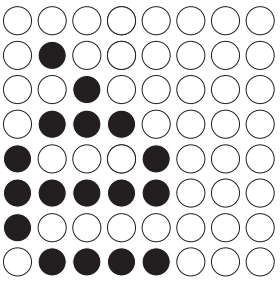

CHR133

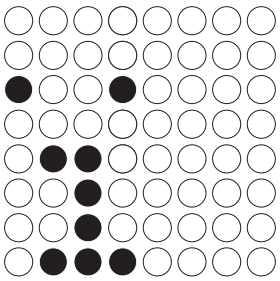

CHR134

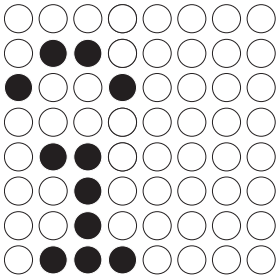

CHR135

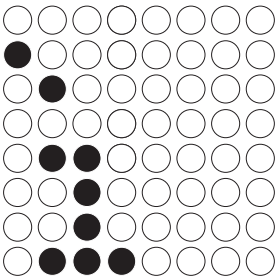

CHR136

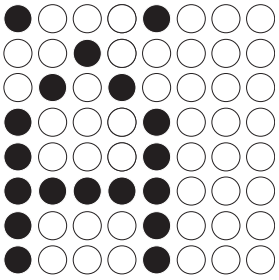

CHR137

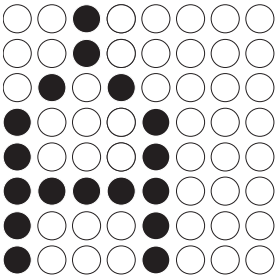

CHR138

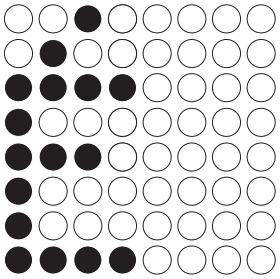

CHR139

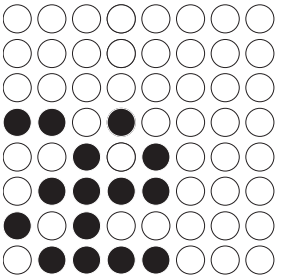

CHR140

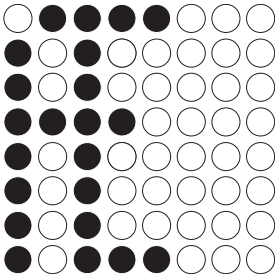

CHR141

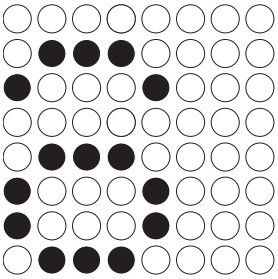

CHR142

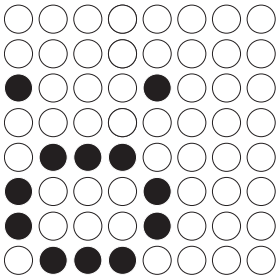

CHR143

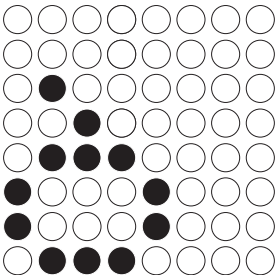

CHR144

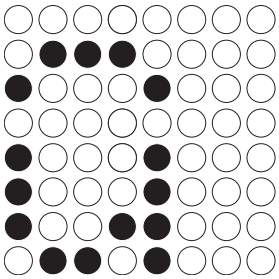

CHR145

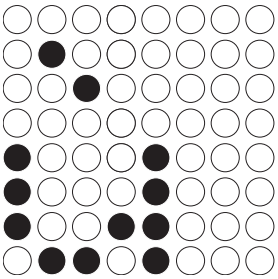

CHR146

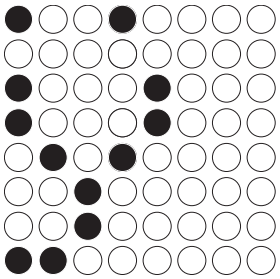

CHR147

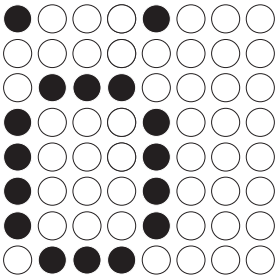

CHR148

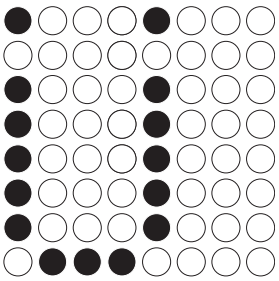

CHR149

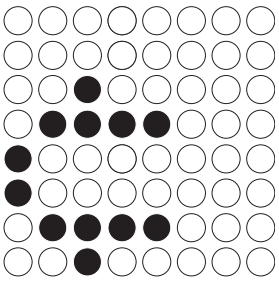

CHR150

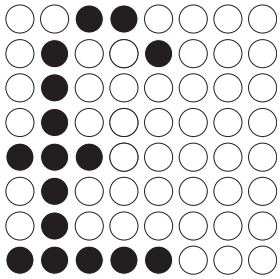

CHR151

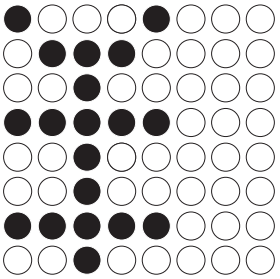

CHR152

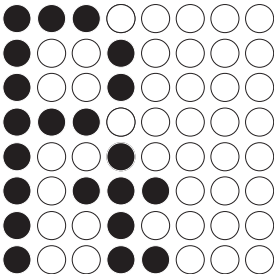

CHR153

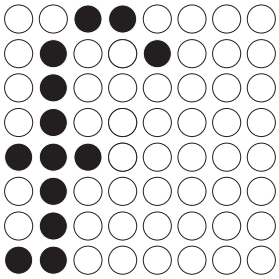

CHR154

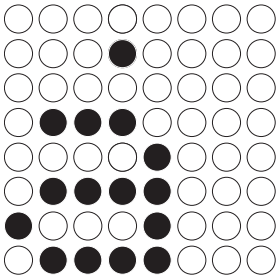

CHR155

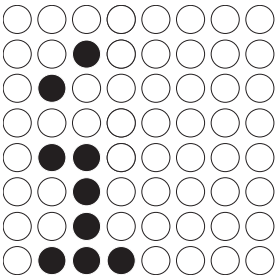

CHR156

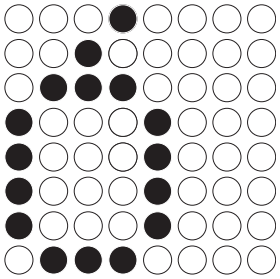

CHR157

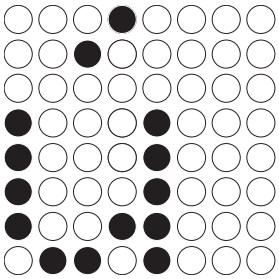

CHR158

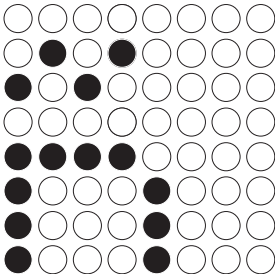

CHR159

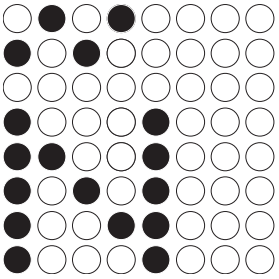

CHR160

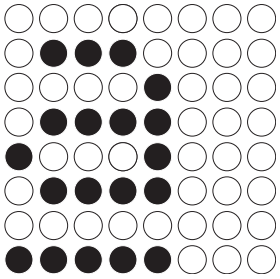

CHR161

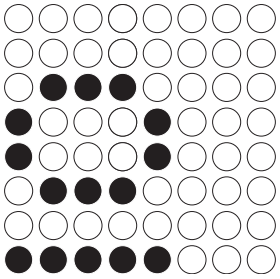

CHR162

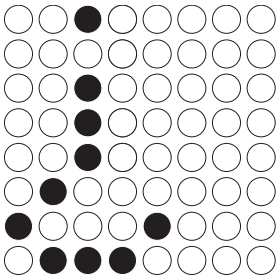

CHR163

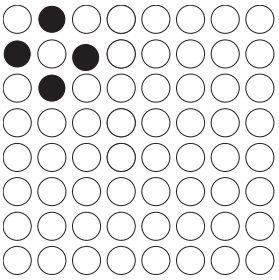

CHR164

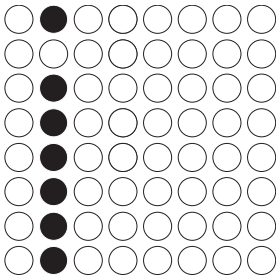

CHR165

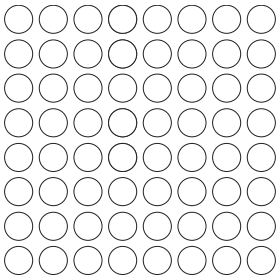

CHR166

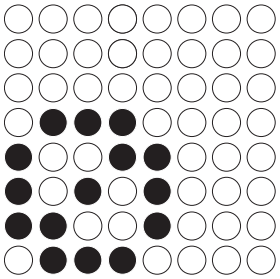

CHR167

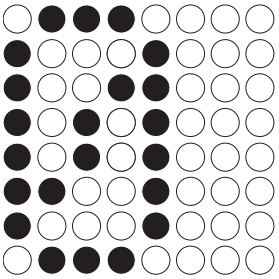

CHR168

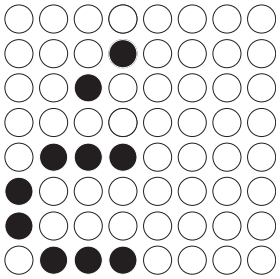

CHR169

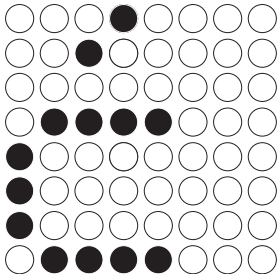

CHR170

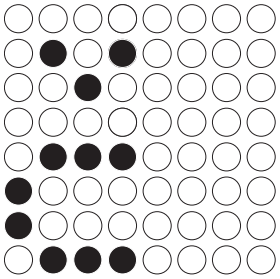

CHR171

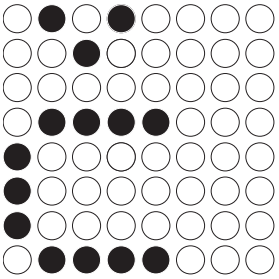

CHR172

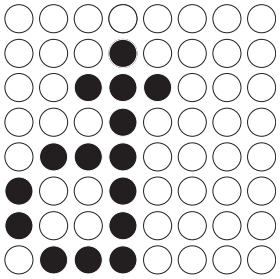

CHR173

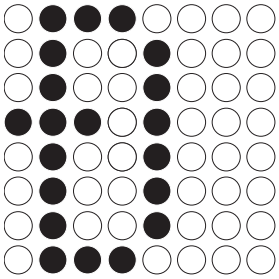

CHR174

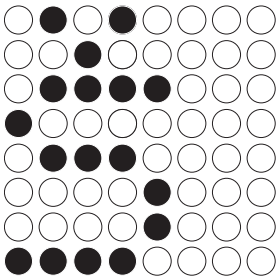

CHR175

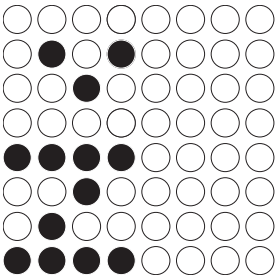

CHR176

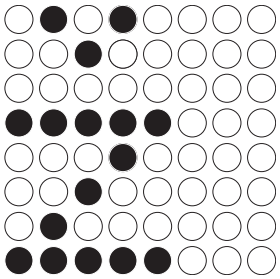

CHR177

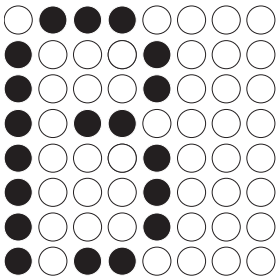

CHR178

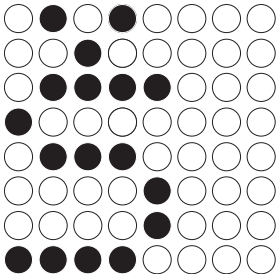

CHR179

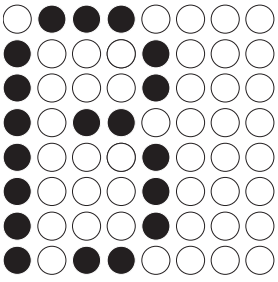

CHR180

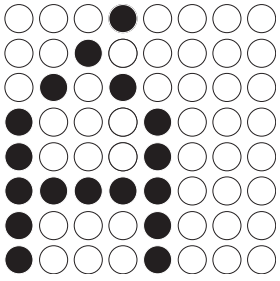

CHR181

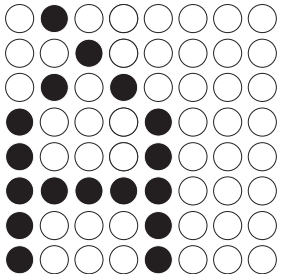

CHR1826

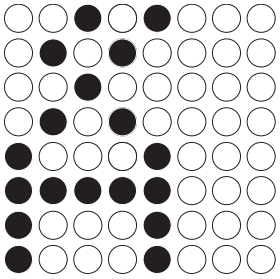

CHR183

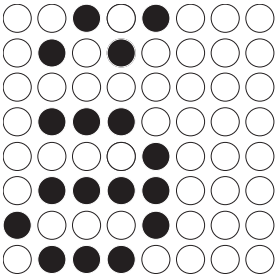

CHR184

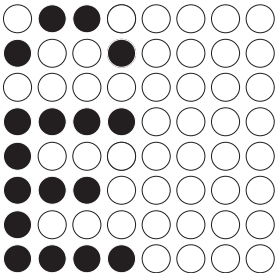

CHR185

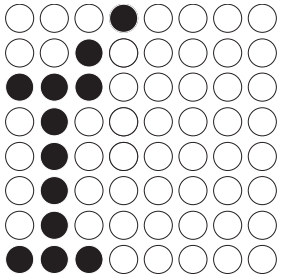

CHR186

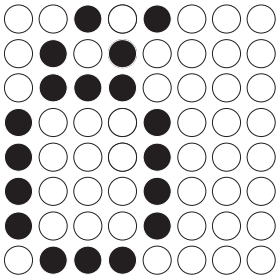

CHR187

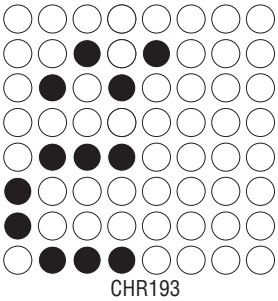

CHR188


CHR189

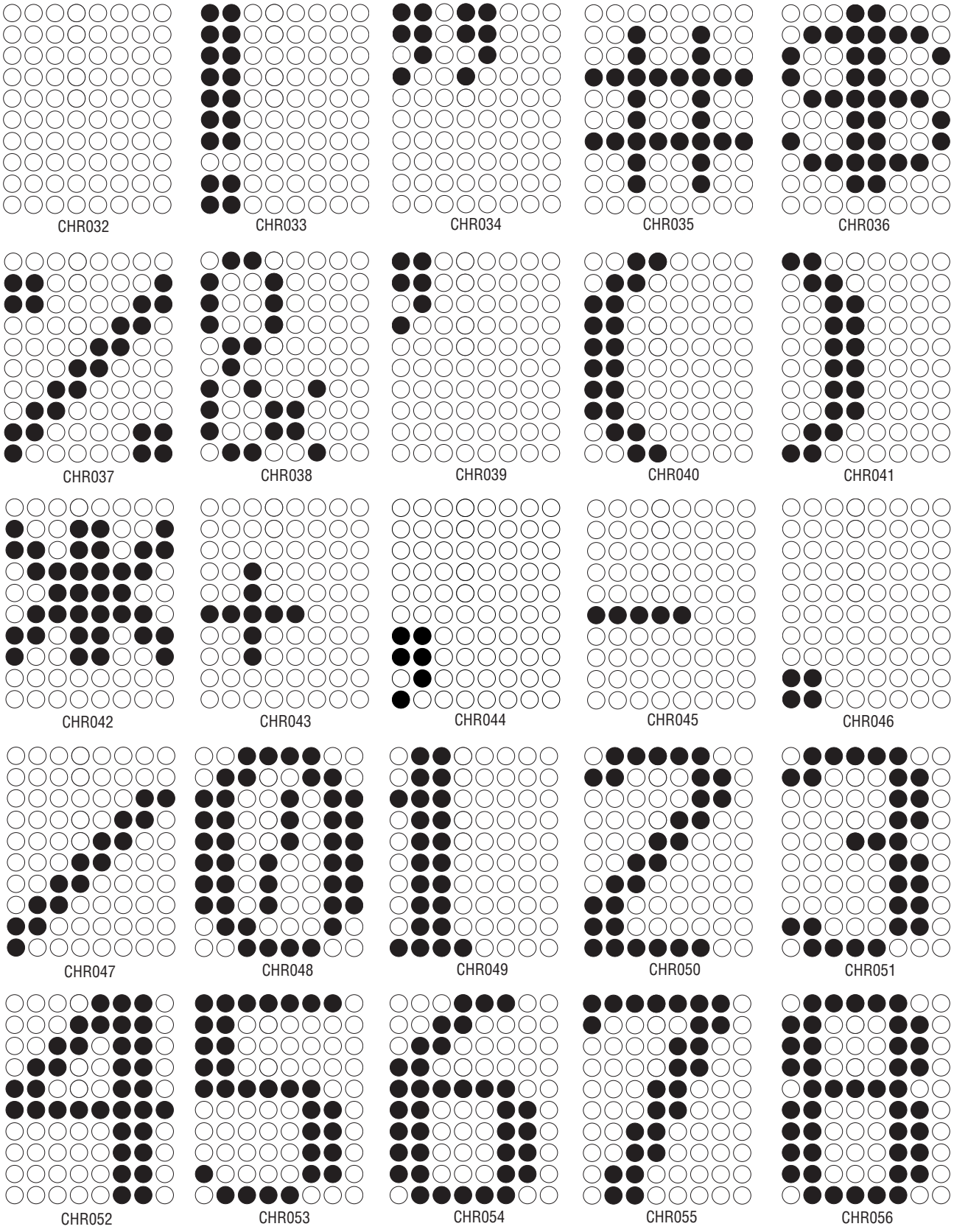

CHR190

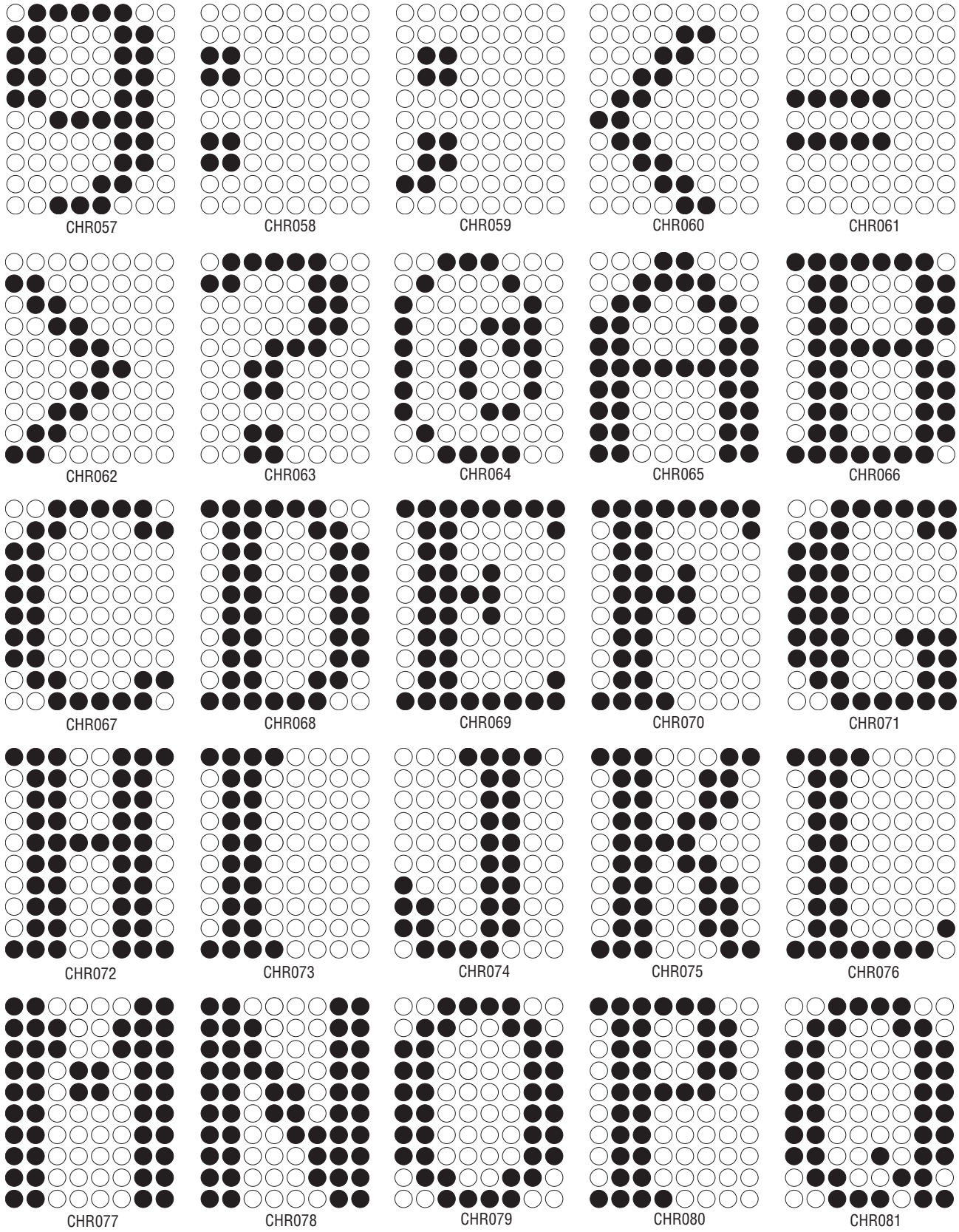

CHR191

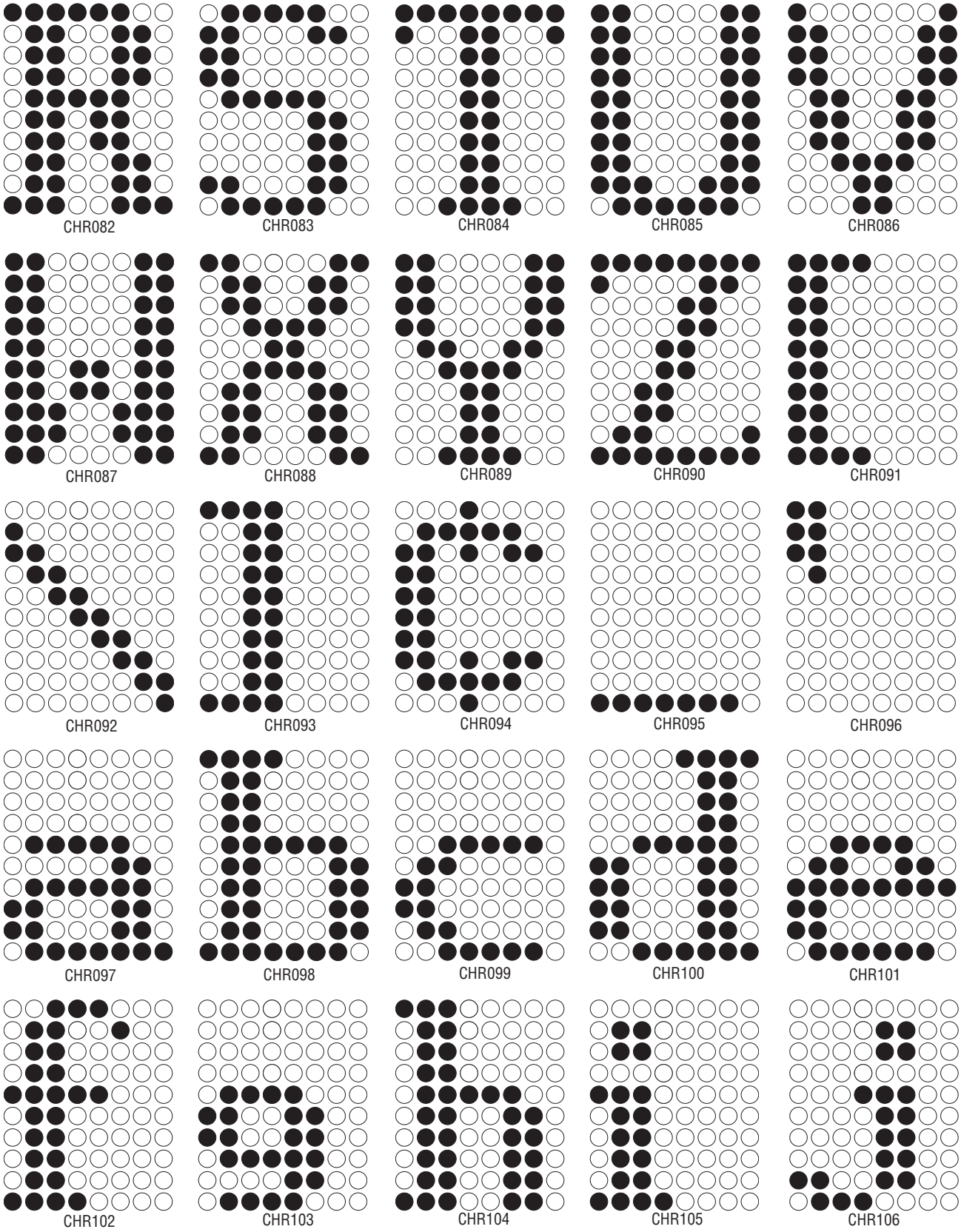

CHR192

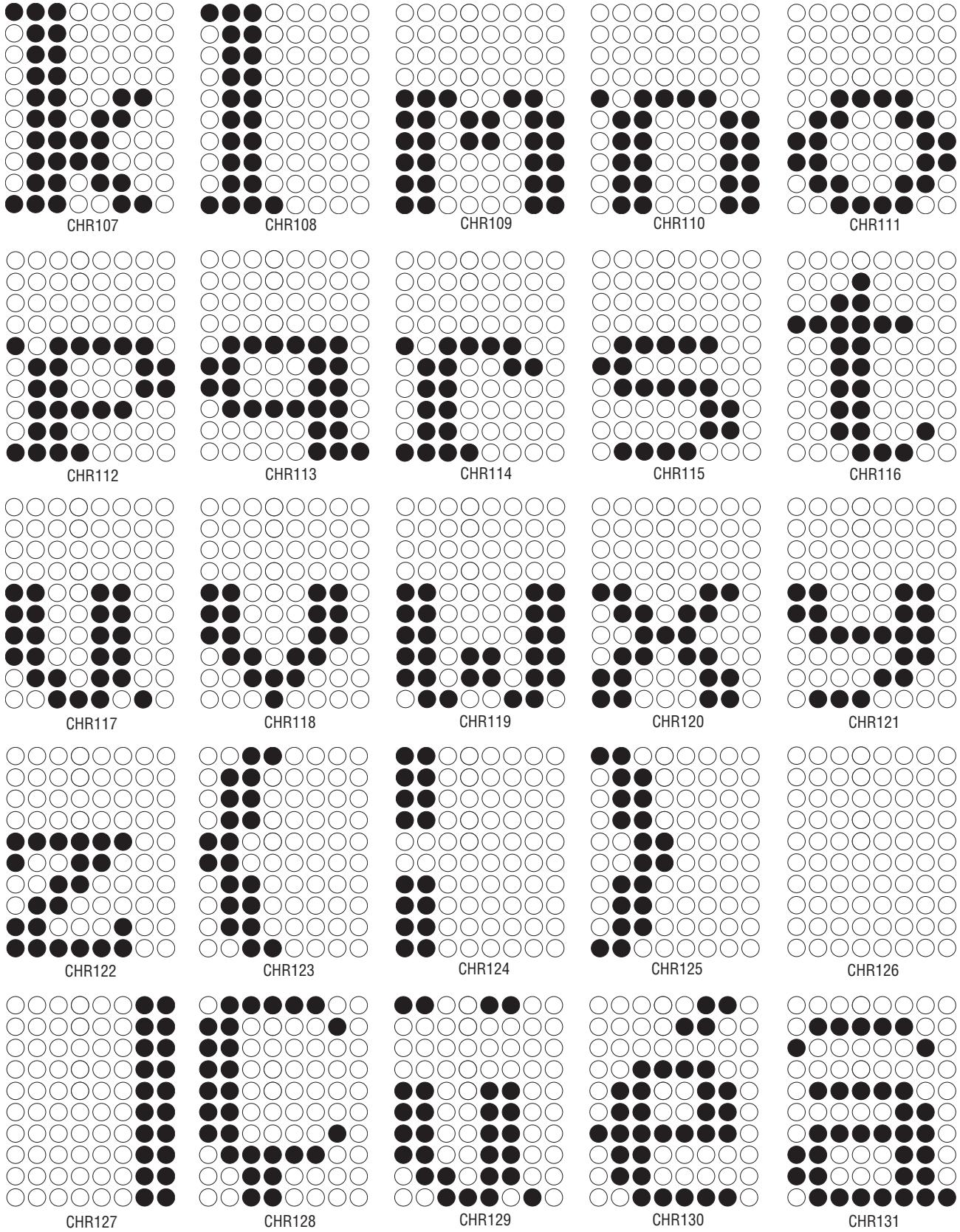

CHR193

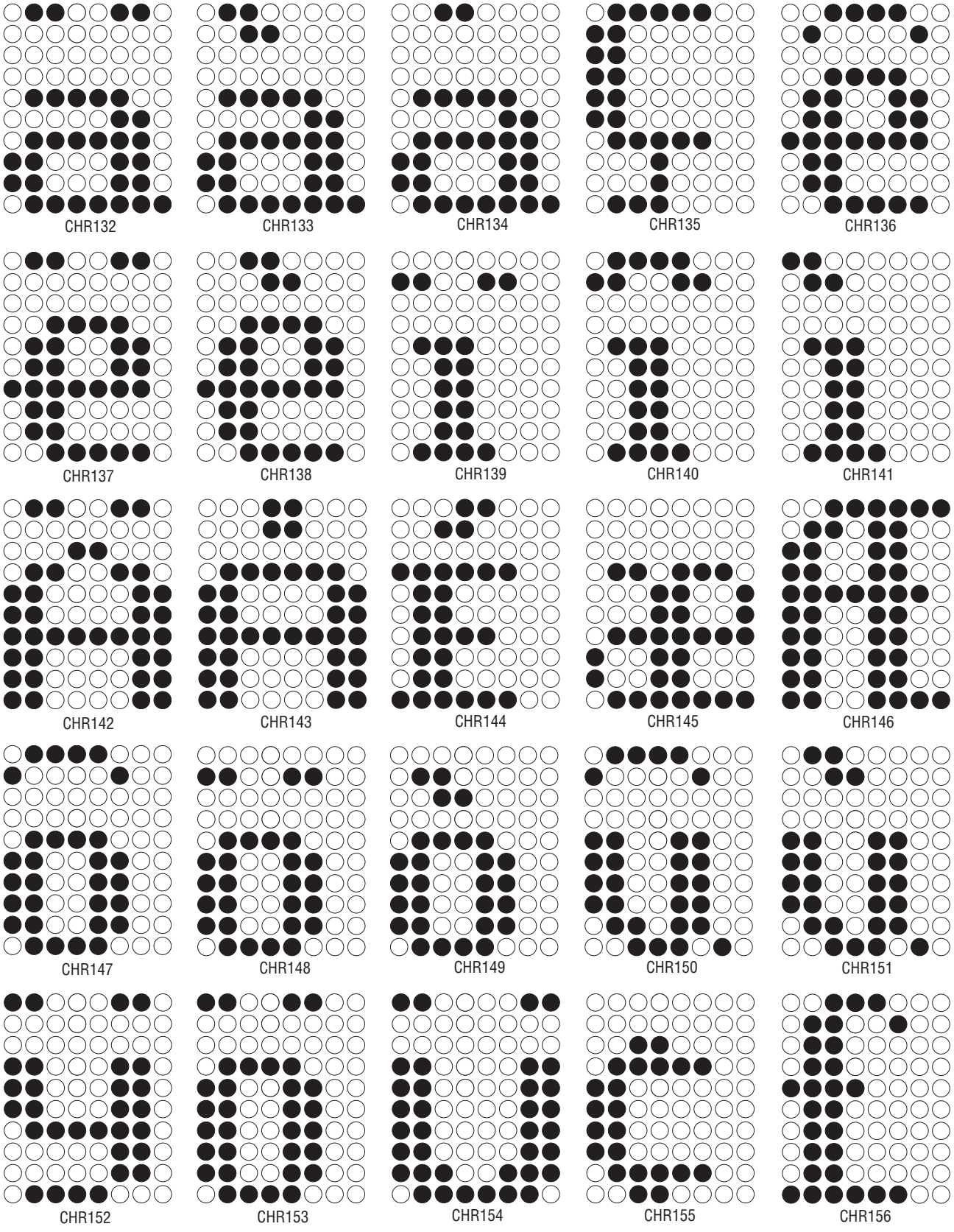
7.13.9 10-High Fancy (SF10)

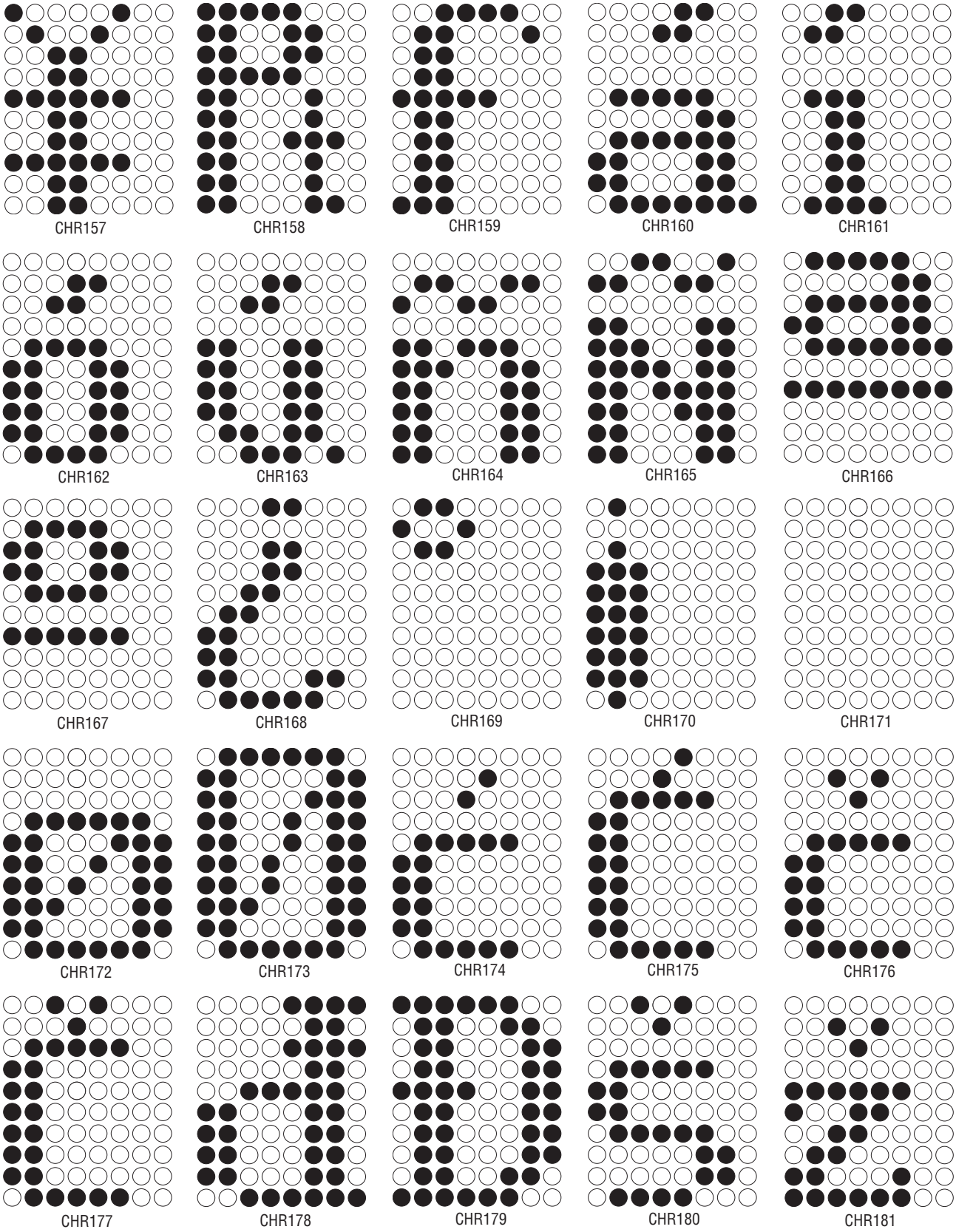


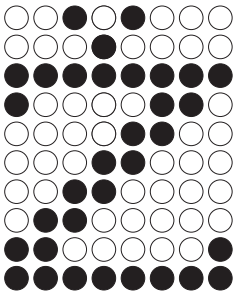

CHR057
CHR058
CHR059
CHR060
CHR061
CHR062
CHR063
CHR064
CHR065
CHR066
CHR067
CHR068
CHR069
CHR070
CHR071
CHR072
CHR073
CHR074
CHR075
CHR076
CHR077
CHR078
CHR079
CHR080
CHR081

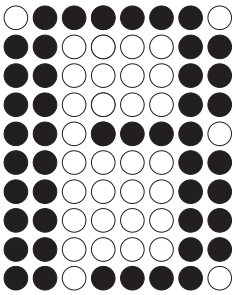

CHR082
CHR083
CHR084
CHR085
CHR086
CHR087
CHR088
CHR089
CHR090
CHR091
CHR092
CHR093
CHR094
CHR095
CHR096
CHR097
CHR098
CHR099
CHR100
CHR101
CHR102
CHR103
CHR104
CHR105
CHR106

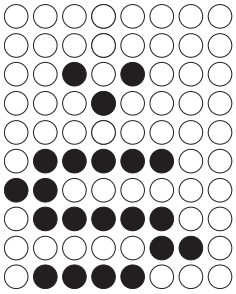

CHR107
CHR108
CHR109
CHR110
CHR111
CHR112
CHR113
CHR114
CHR115
CHR116
CHR117
CHR118
CHR119
CHR120
CHR121
CHR122
CHR123
CHR124
CHR125
CHR126
CHR127
CHR128
CHR129
CHR130
CHR131

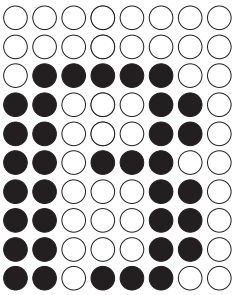

CHR132
CHR133
CHR134
CHR135
CHR136
CHR137
CHR138
CHR139
CHR140
CHR141
CHR142
CHR143
CHR144
CHR145
CHR146
CHR147
CHR148
CHR149
CHR150
CHR151
CHR152
CHR153
CHR154
CHR155
CHR156

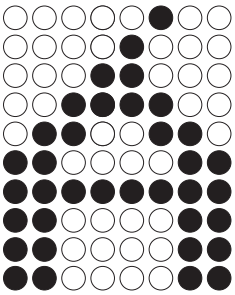

CHR157
CHR158
CHR159
CHR160
CHR161
CHR162
CHR163
CHR164
CHR165
CHR166
CHR167
CHR168
CHR169
CHR170
CHR171
CHR172
CHR173
CHR174
CHR175
CHR176
CHR177
CHR178
CHR179
CHR180
CHR181

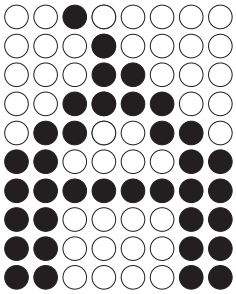

CHR182

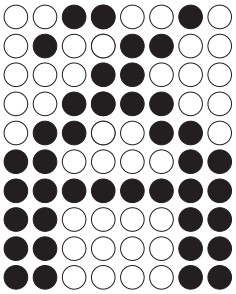

CHR183

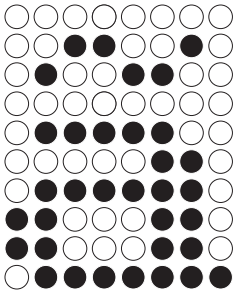

CHR184

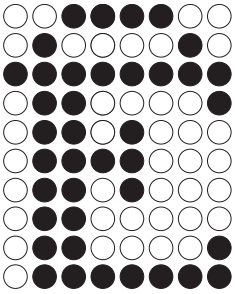

CHR185

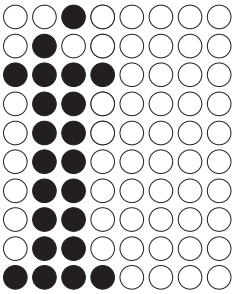

CHR186

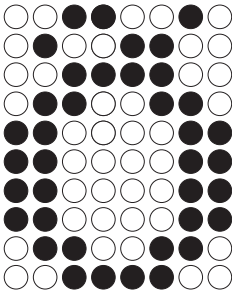

CHR187

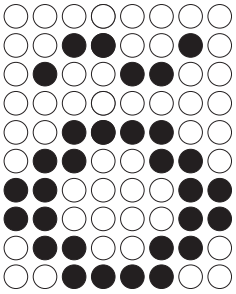

CHR188


CHR189

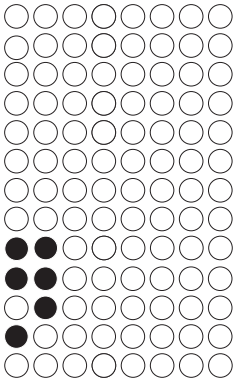

CHR190

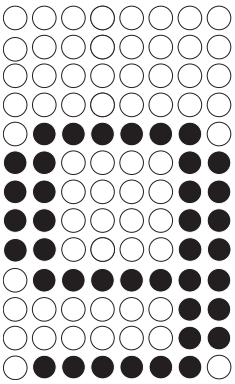

CHR191

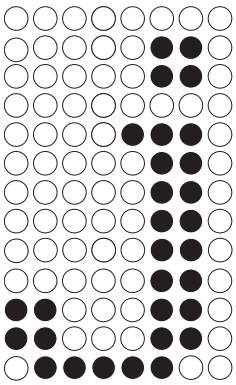

CHR192

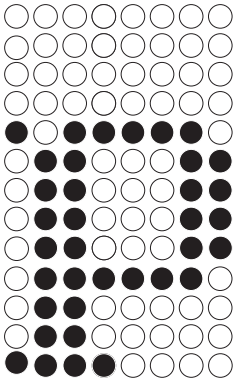

CHR193

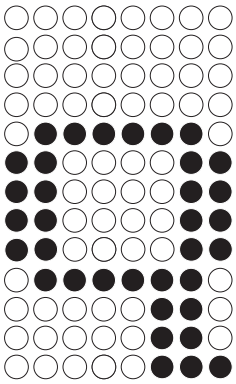
7.13.10 10-High True Descender Fancy

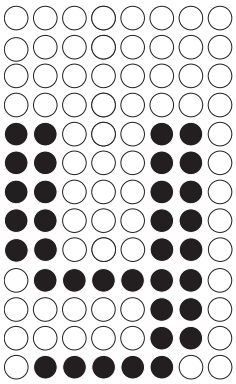

CHR10C


CHR10G

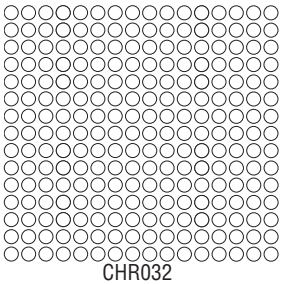

CHR10J

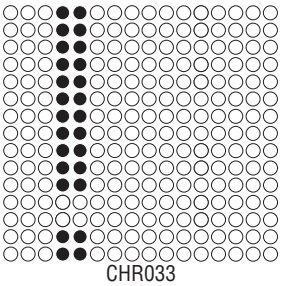

CHR10P

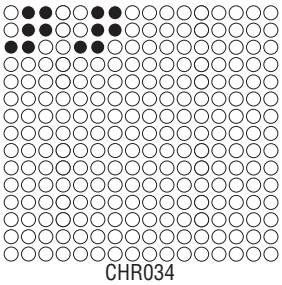

CHR10Q

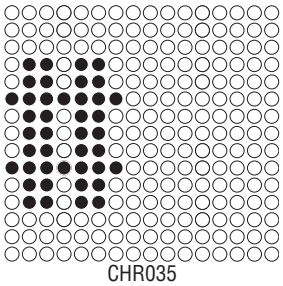

CHR10Y

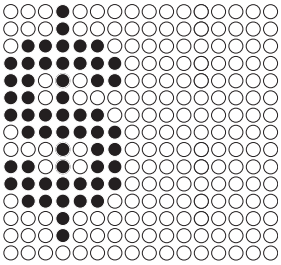
7.13.11 15-High Regular (SS15)

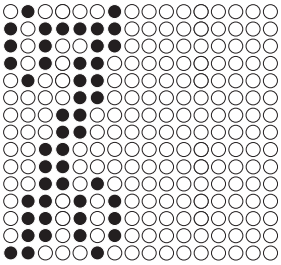

CHR032

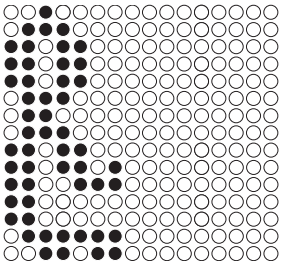

CHR033

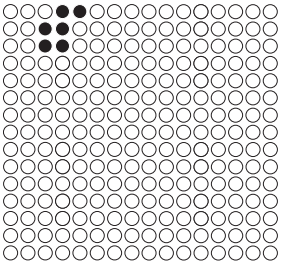

CHR034

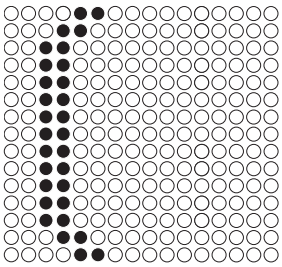

CHR035

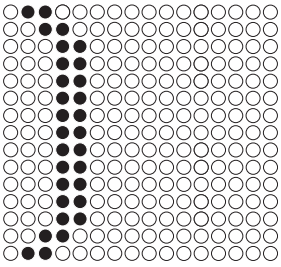

CHR036

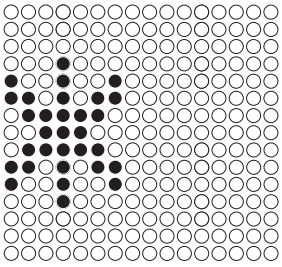

CHR037

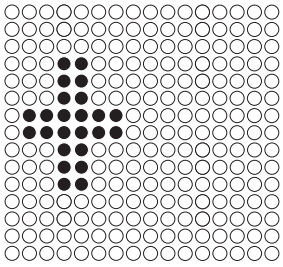

CHR038

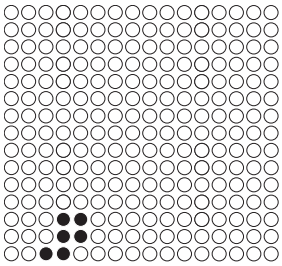

CHR039

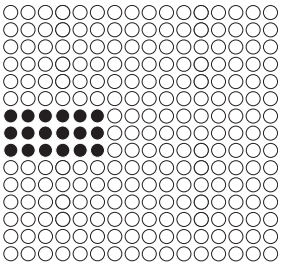

CHR040

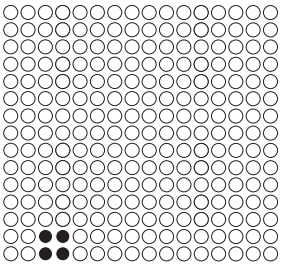

CHR041

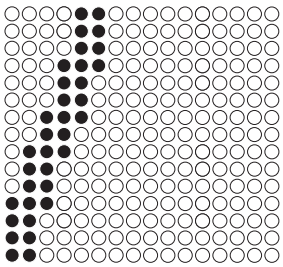

CHR042

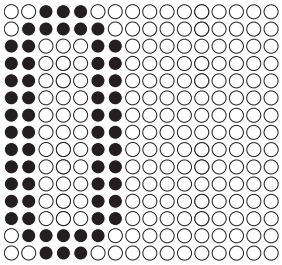

CHR043

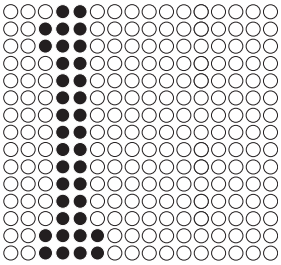

CHR044

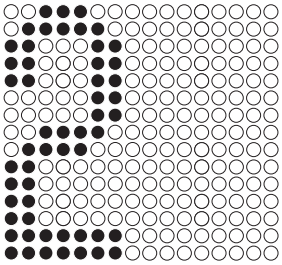

CHR045

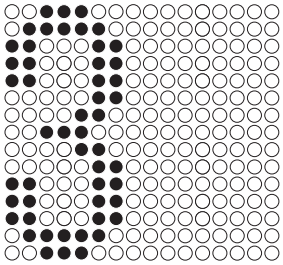

CHR046

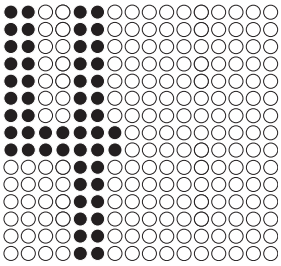

CHR047

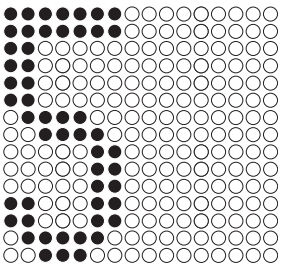

CHR048

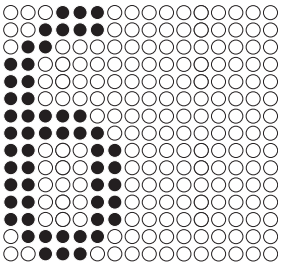

CHR049

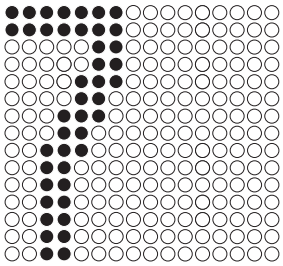

CHR050

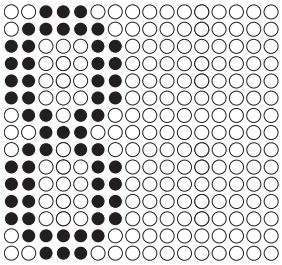

CHR051

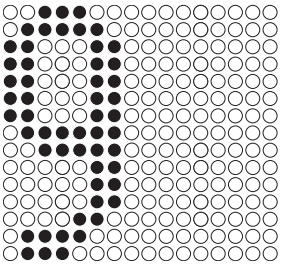

CHR052

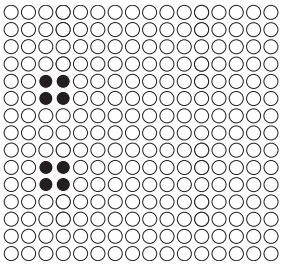

CHR053

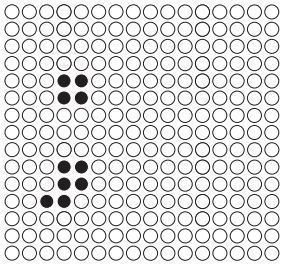

CHR054

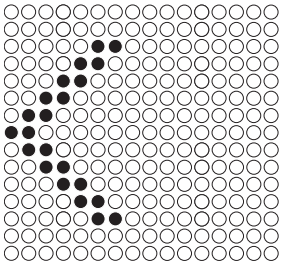

CHR055

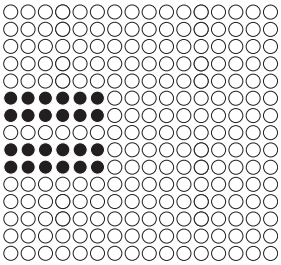

CHR056

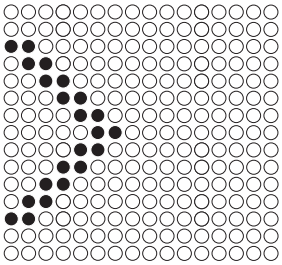

CHR057

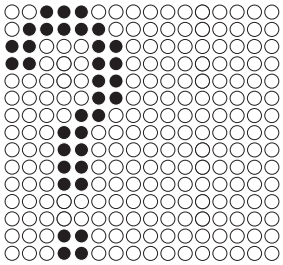

CHR058

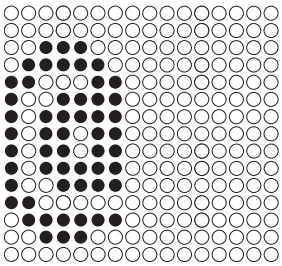

CHR059

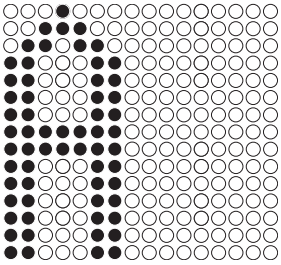

CHR060

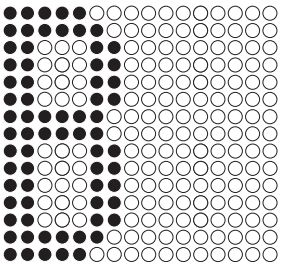

CHR061

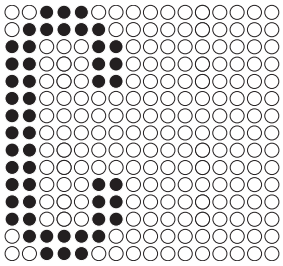

CHR062

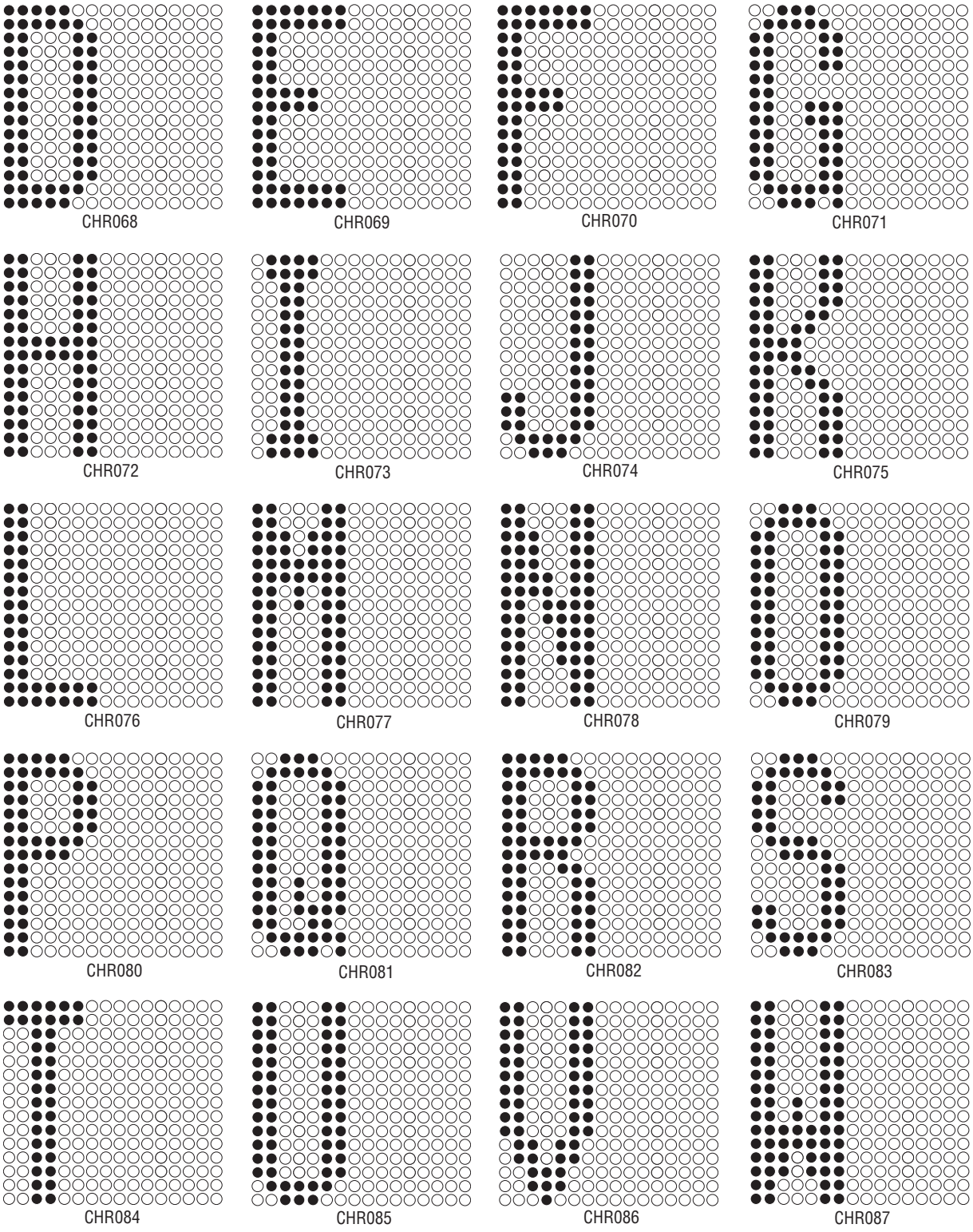

CHR063

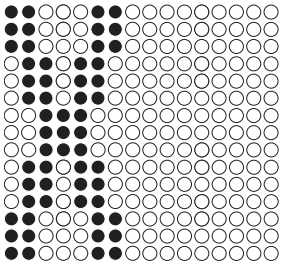

CHR064

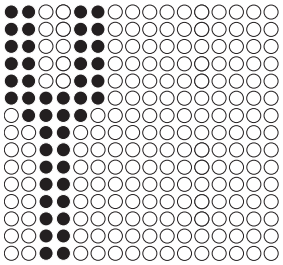

CHR065

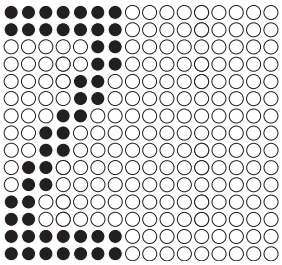

CHR066

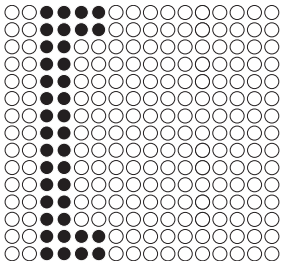

CHR067

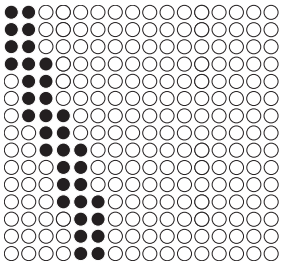

CHR068
CHR069
CHR070
CHR071
CHR072
CHR073
CHR074
CHR075
CHR076
CHR077
CHR078
CHR079
CHR080
CHR081
CHR082
CHR083
CHR084
CHR085
CHR086
CHR087

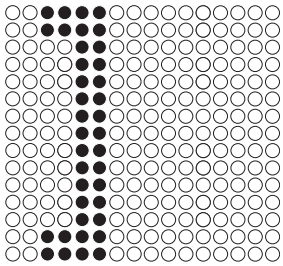

CHR088

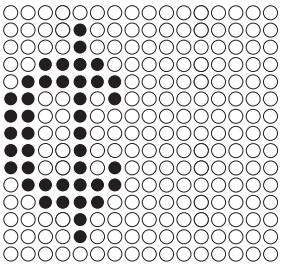

CHR089

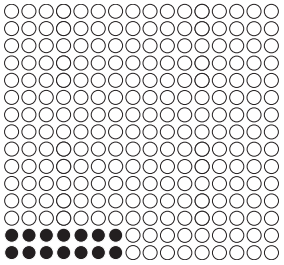

CHR090

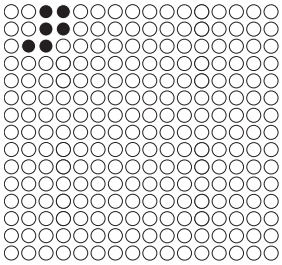

CHR091

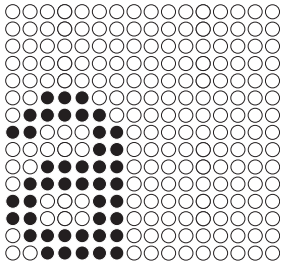

CHR092

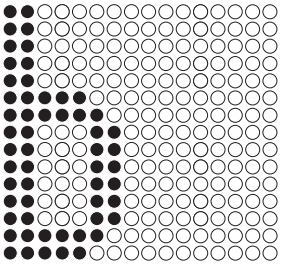

CHR093

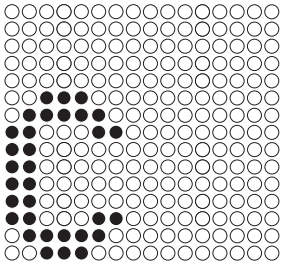

CHR094

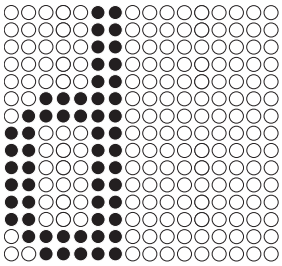

CHR095

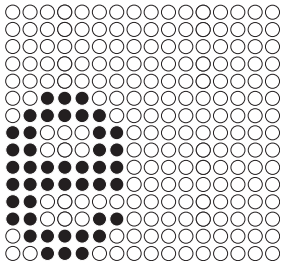

CHR096

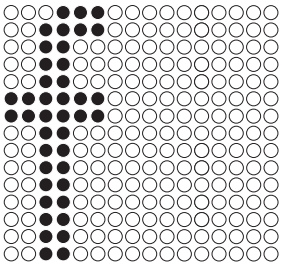

CHR097

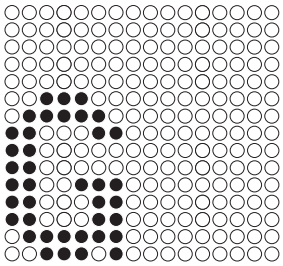

CHR098

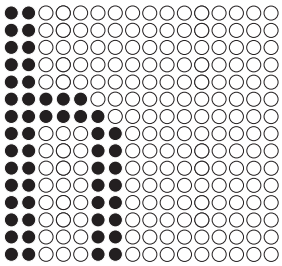

CHR099

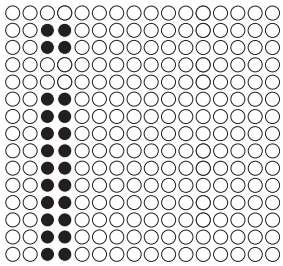

CHR100

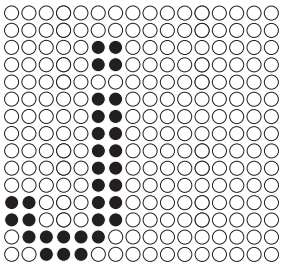

CHR101

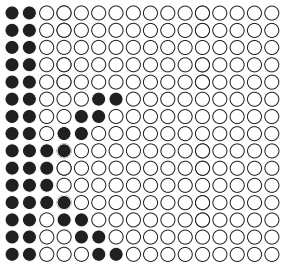

CHR102

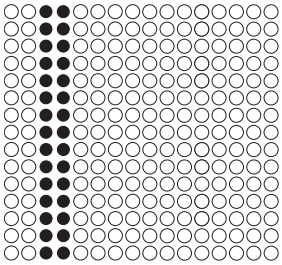

CHR103

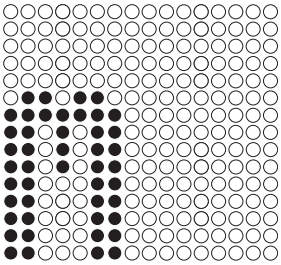

CHR104

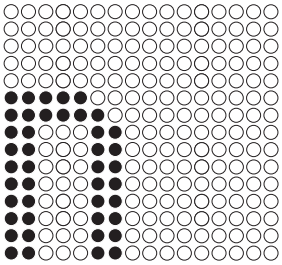

CHR105

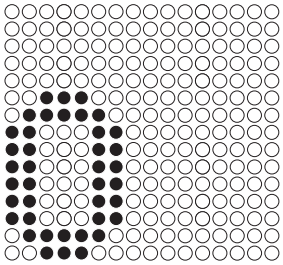

CHR106

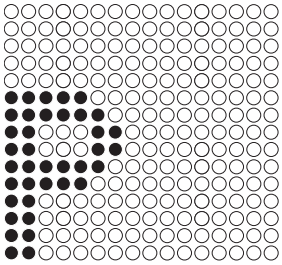

CHR107

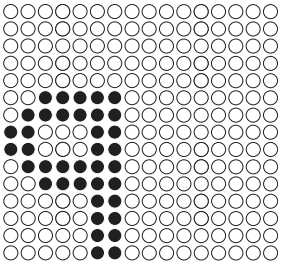

CHR108

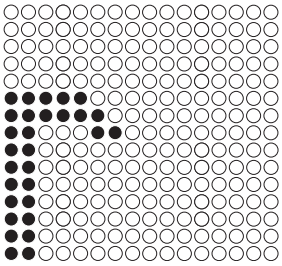

CHR109

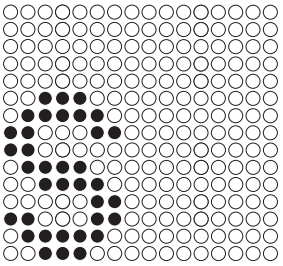

CHR110

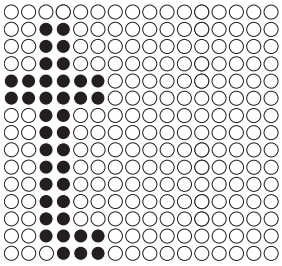

CHR111

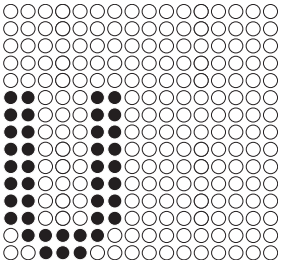

CHR112

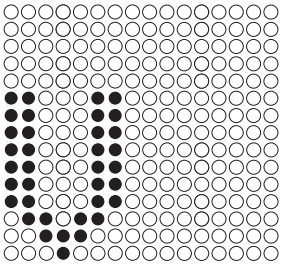

CHR113

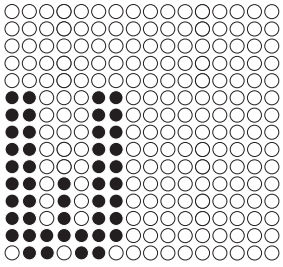

CHR114

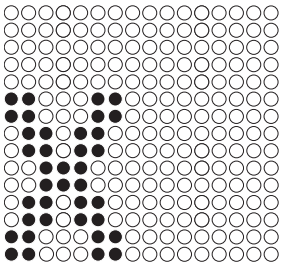

CHR115

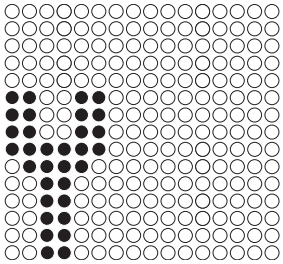

CHR116

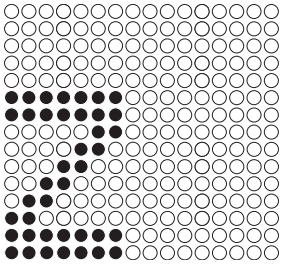

CHR117

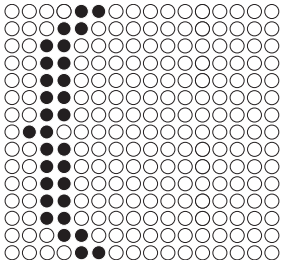

CHR118

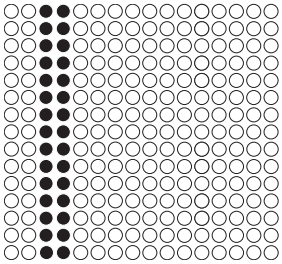

CHR119

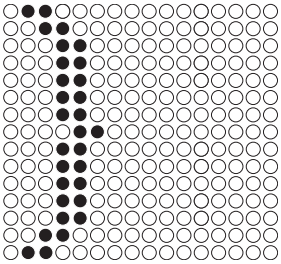

CHR120

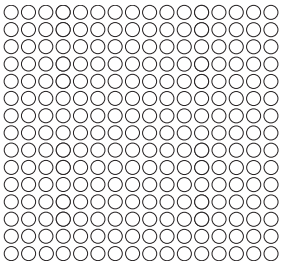

CHR121

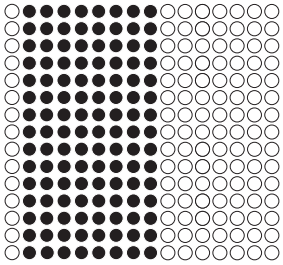

CHR122

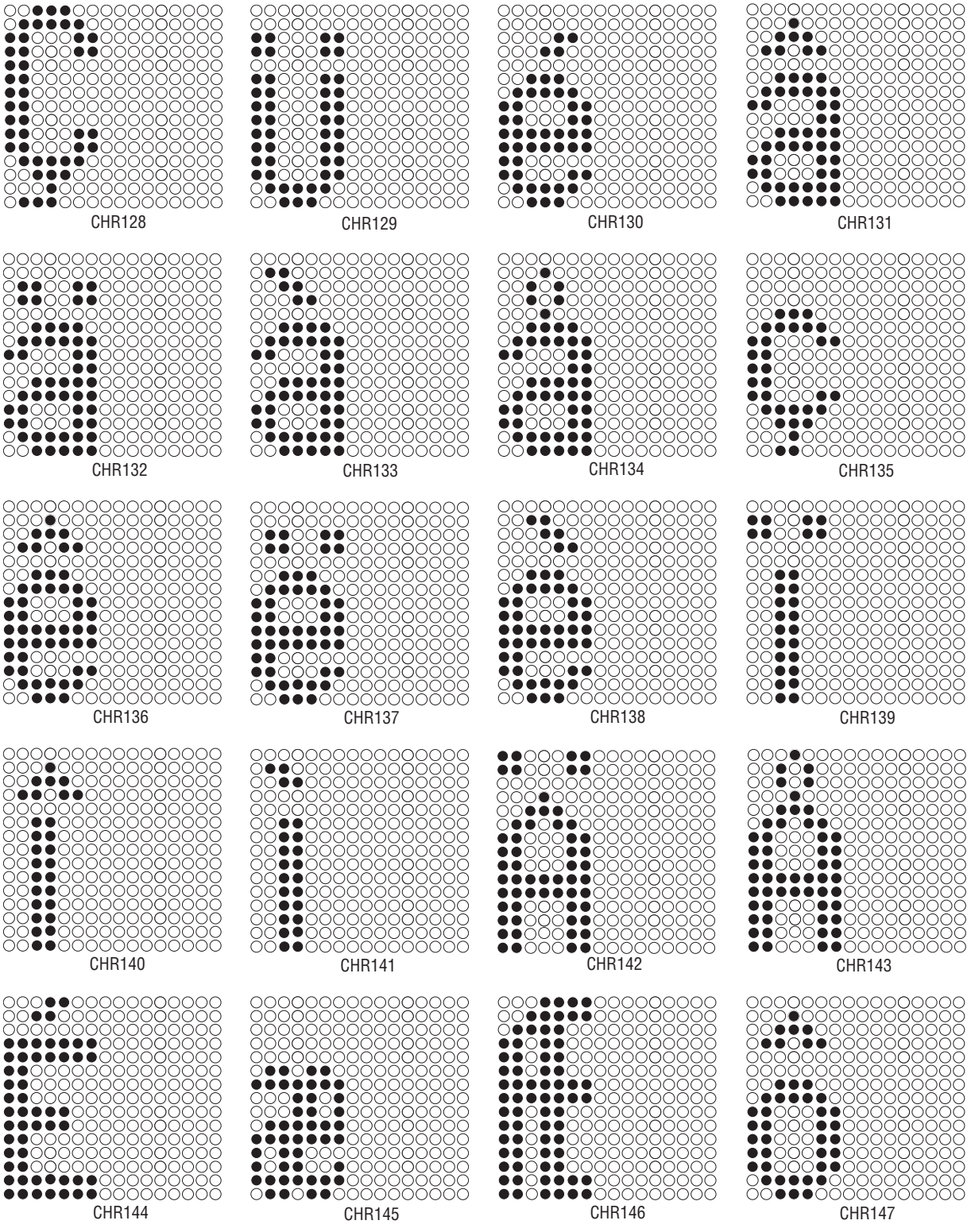

CHR123

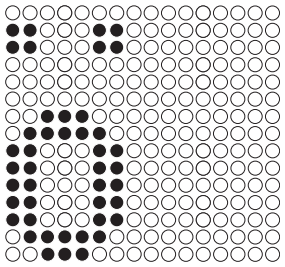

CHR124

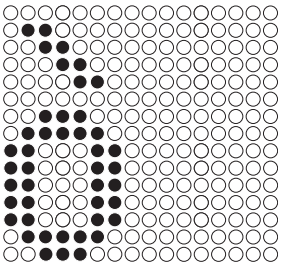

CHR125

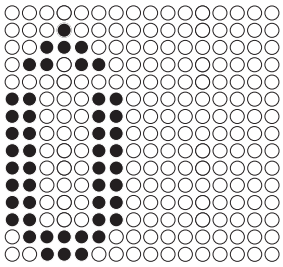

CHR126

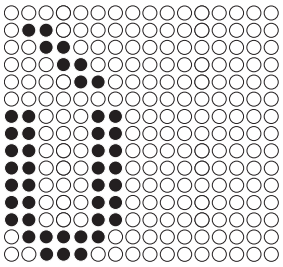

CHR127

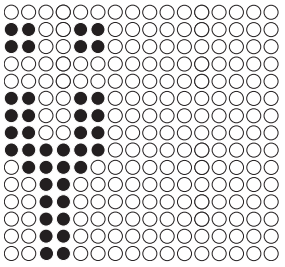

CHR128
CHR129
CHR130
CHR131
CHR132
CHR133
CHR134
CHR135
CHR136
CHR137
CHR138
CHR139
CHR140
CHR141
CHR142
CHR143
CHR144
CHR145
CHR146
CHR147

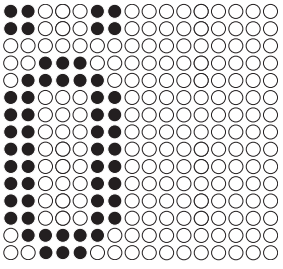

CHR148

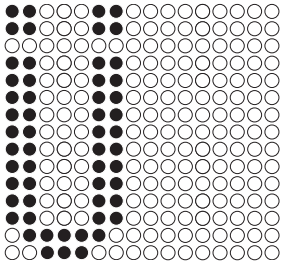

CHR149

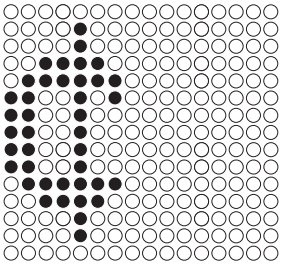

CHR150

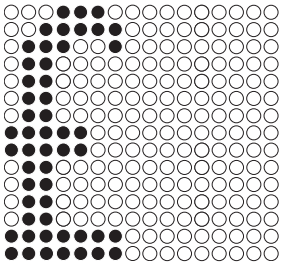

CHR151

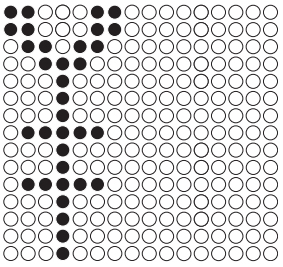

CHR152

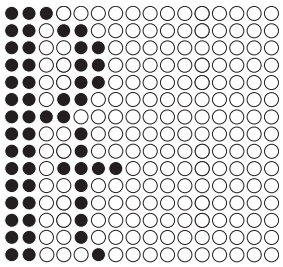

CHR153

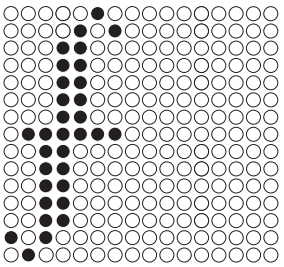

CHR154

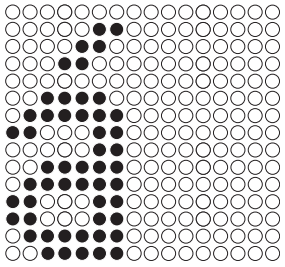

CHR155

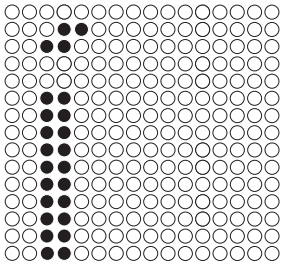

CHR156

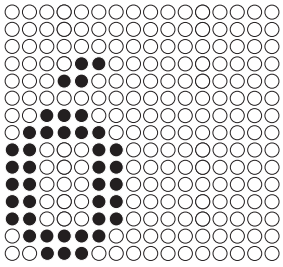

CHR157

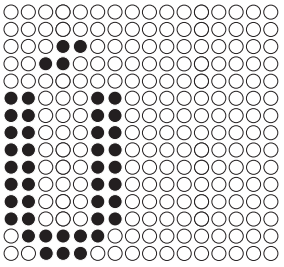

CHR158

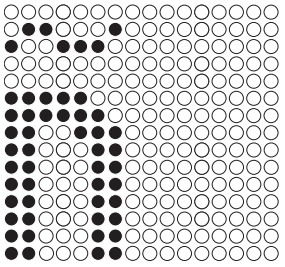

CHR159

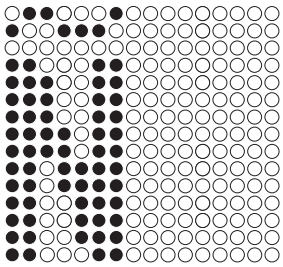

CHR160

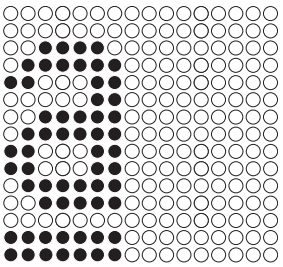

CHR161

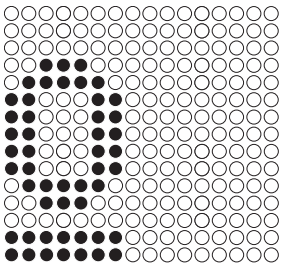

CHR162

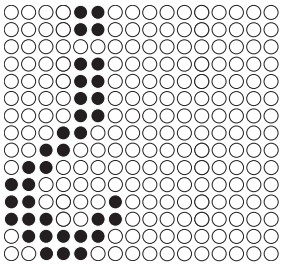

CHR163

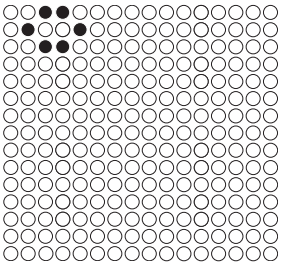

CHR164

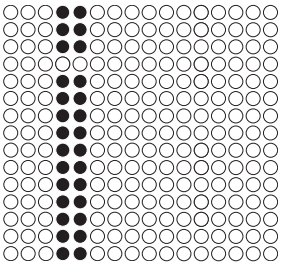

CHR165

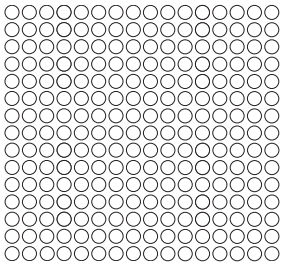

CHR166

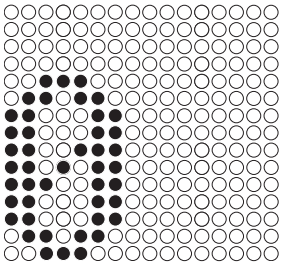

CHR167

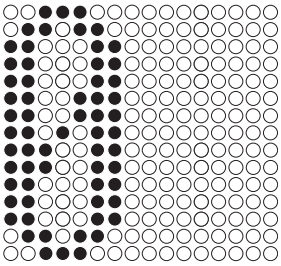

CHR168

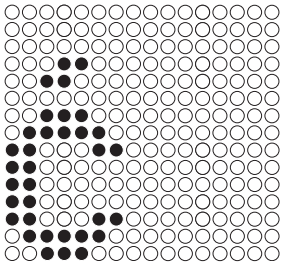

CHR169

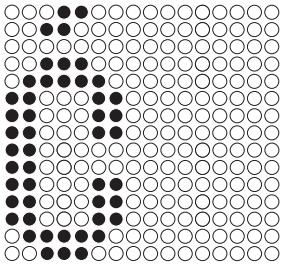

CHR170

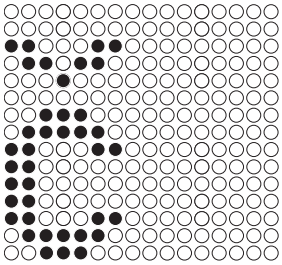

CHR171

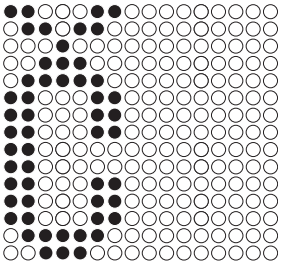

CHR172

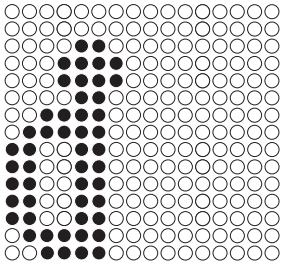

CHR173

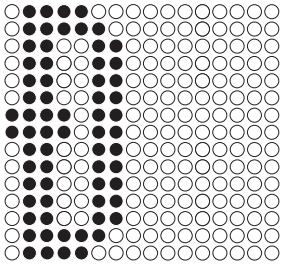

CHR174

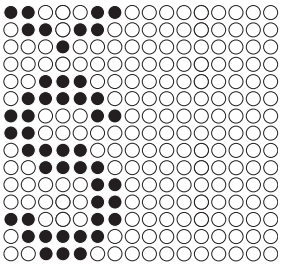

CHR175

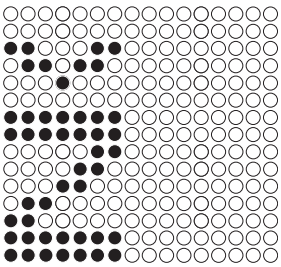

CHR176

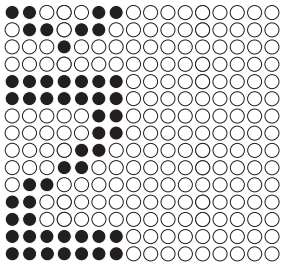

CHR177

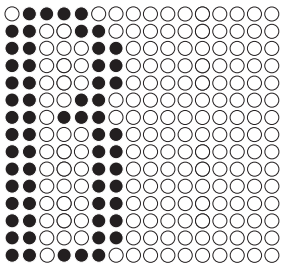

CHR178

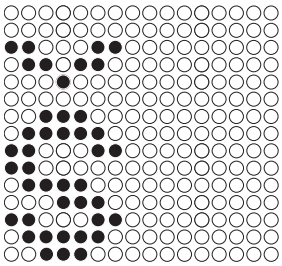

CHR179

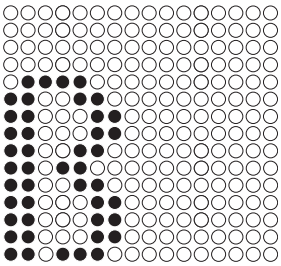

CHR180

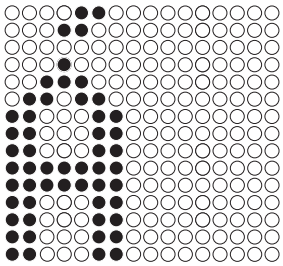

CHR181

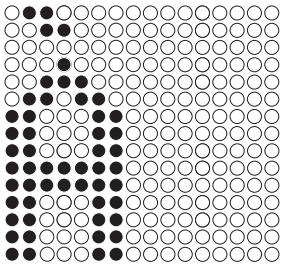

CHR182

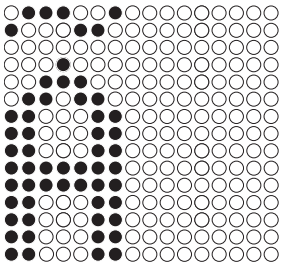

CHR183

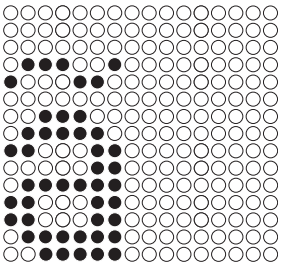

CHR184

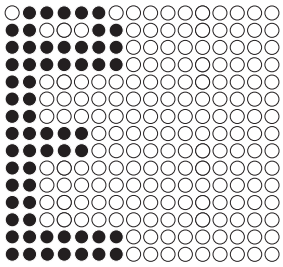

CHR185

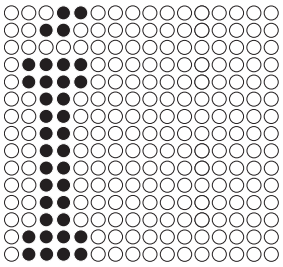

CHR186

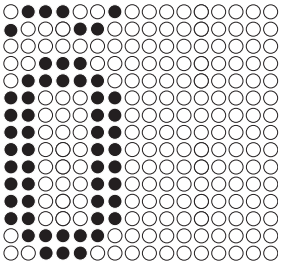

CHR187

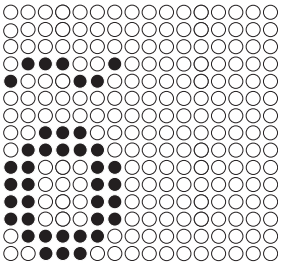

CHR188


CHR189

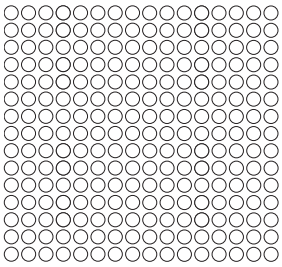

CHR190

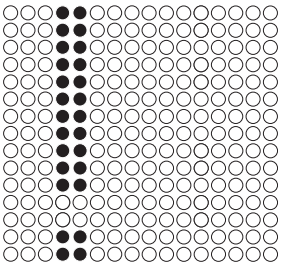

CHR191

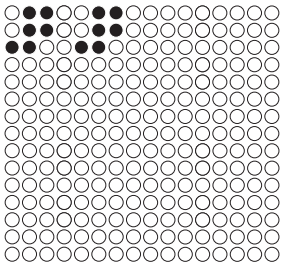

CHR192

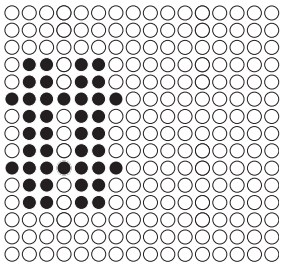

CHR193

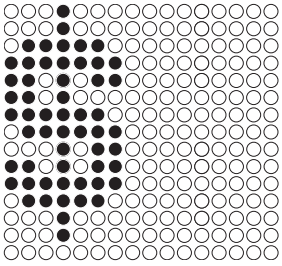
7.13.12 15-High Fancy (SF15)

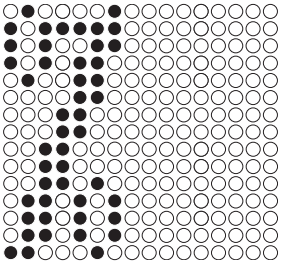

CHR032

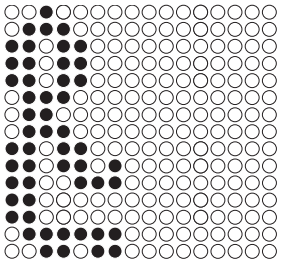

CHR033

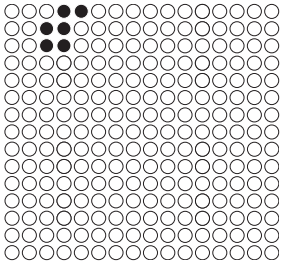

CHR034

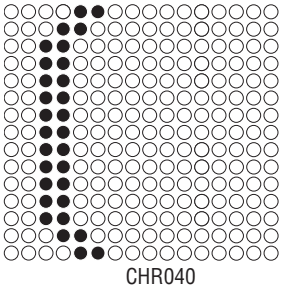

CHR035

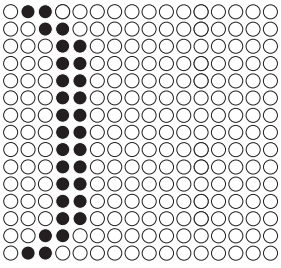

CHR036

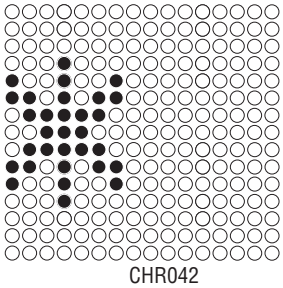

CHR037

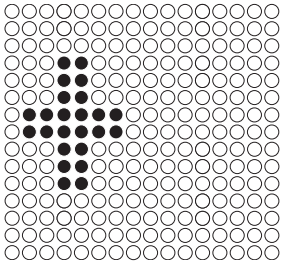

CHR038

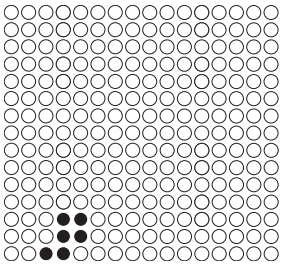

CHR039

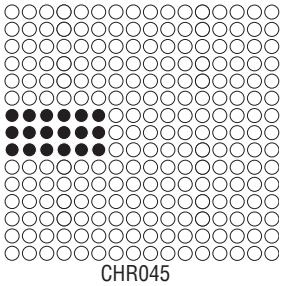

CHR040

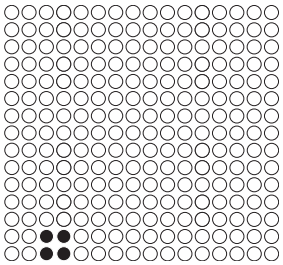

CHR041

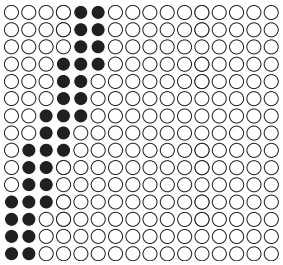

CHR042

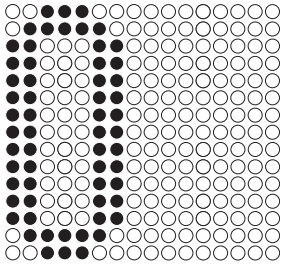

CHR043

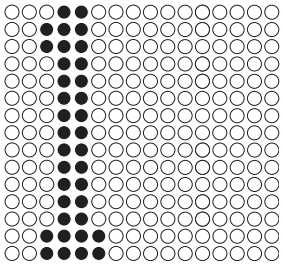

CHR044

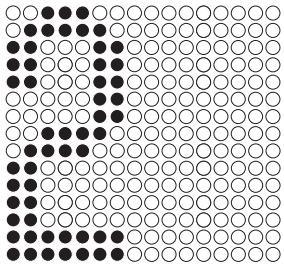

CHR045

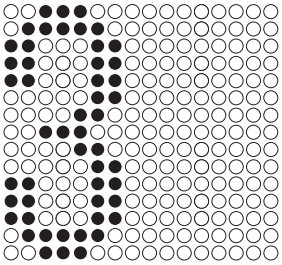

CHR046

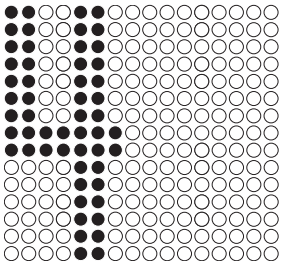

CHR047

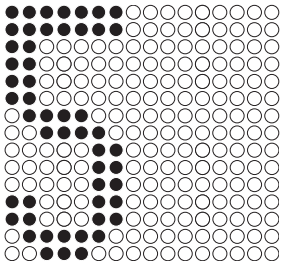

CHR048

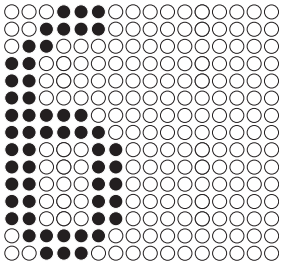

CHR049

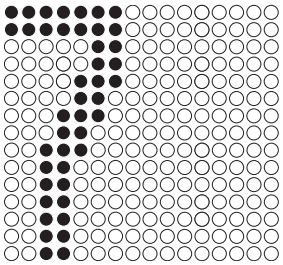

CHR050

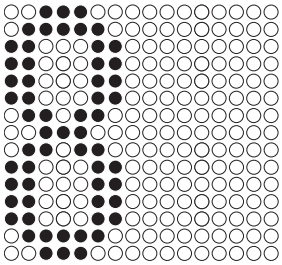

CHR051

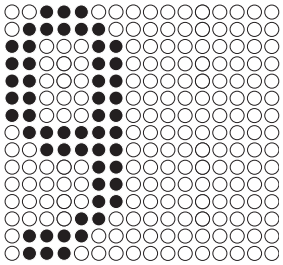

CHR052

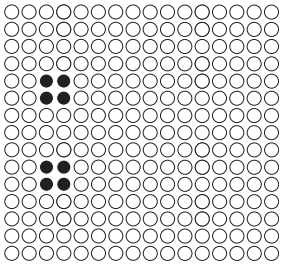

CHR053

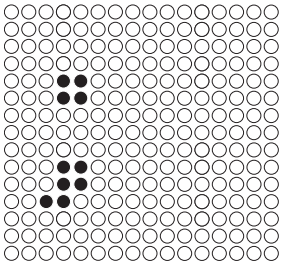

CHR054

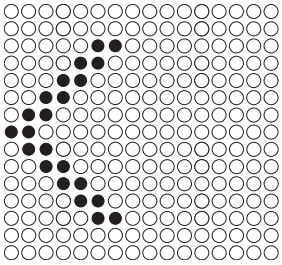

CHR055

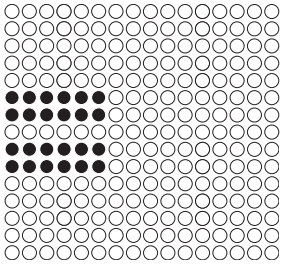

CHR056

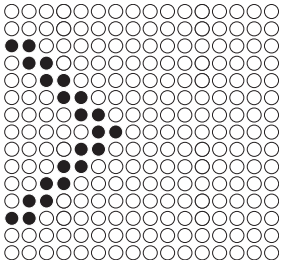

CHR057

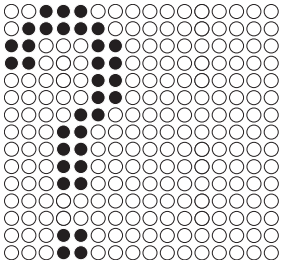

CHR058

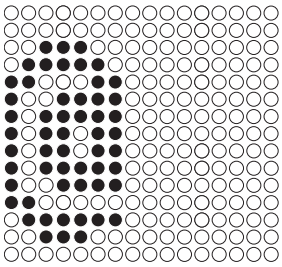

CHR059

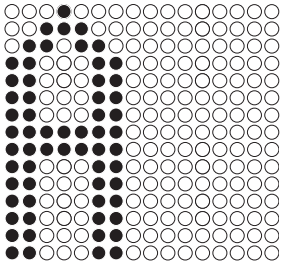

CHR060

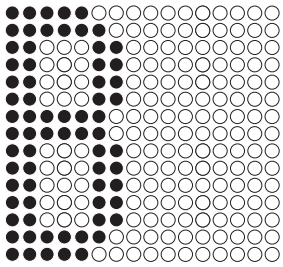

CHR061

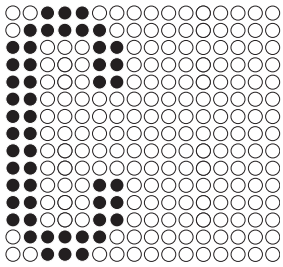

CHR062

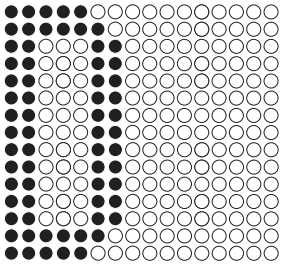

CHR063

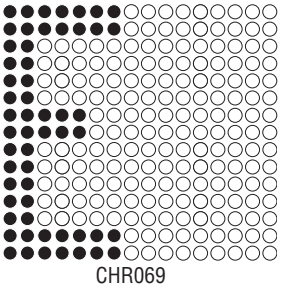

CHR064

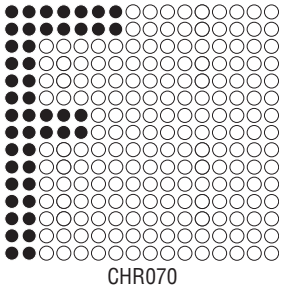

CHR065

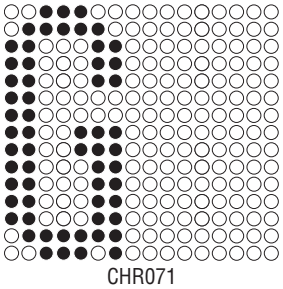

CHR066

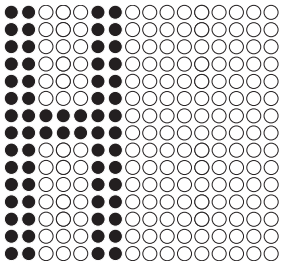

CHR067

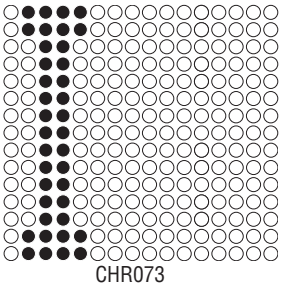

CHR068

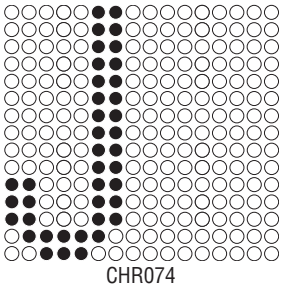

CHR069

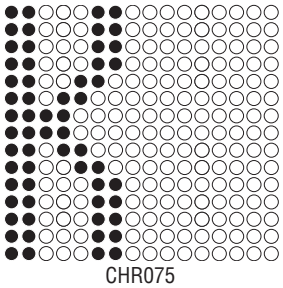

CHR070

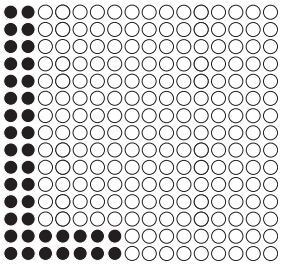

CHR071

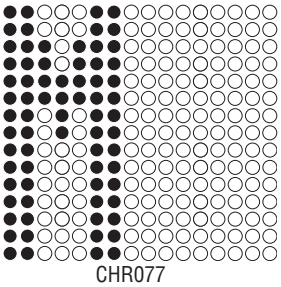

CHR072

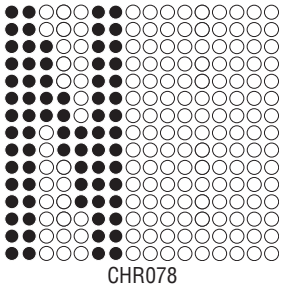

CHR073

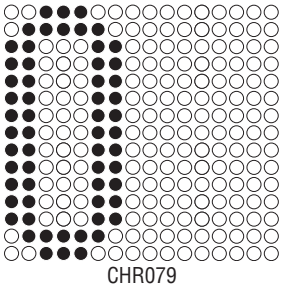

CHR074

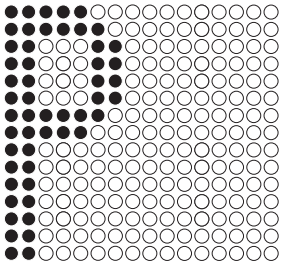

CHR075

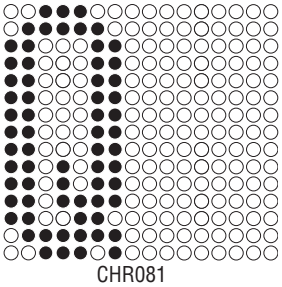

CHR076

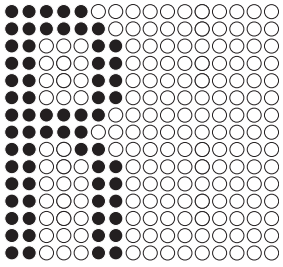

CHR077

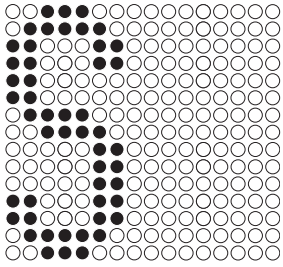

CHR078

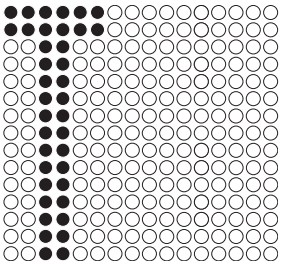

CHR079

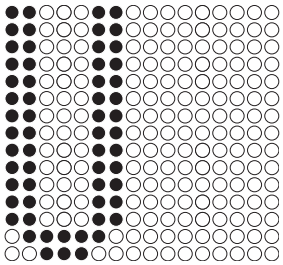

CHR080

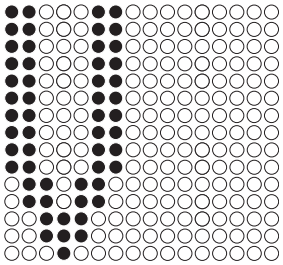

CHR081

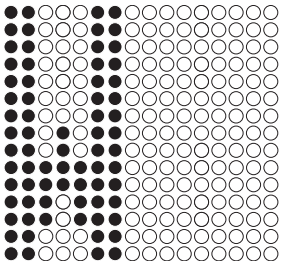

CHR082

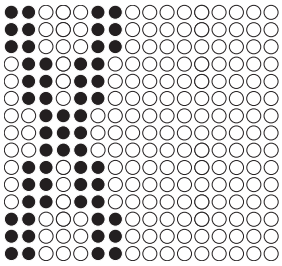

CHR083

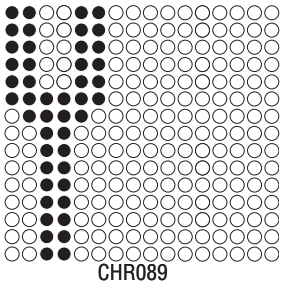

CHR084

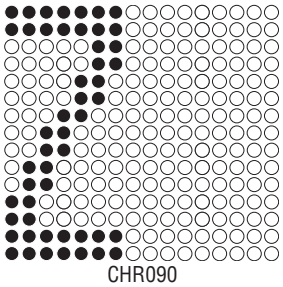

CHR085

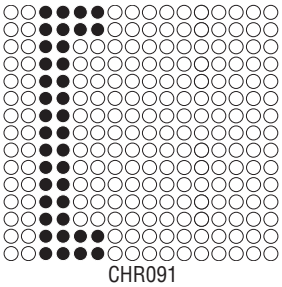

CHR086

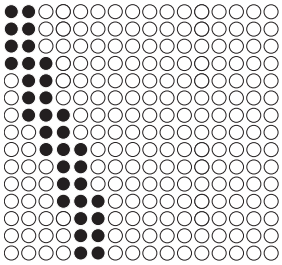

CHR087

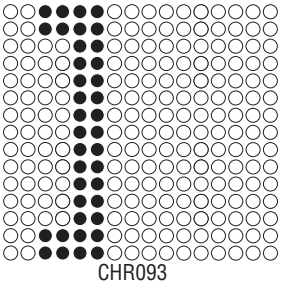

CHR088

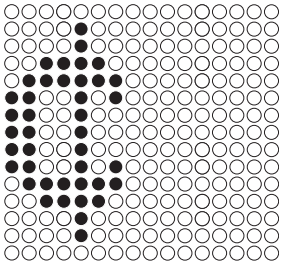

CHR089

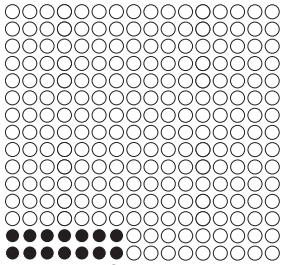

CHR090

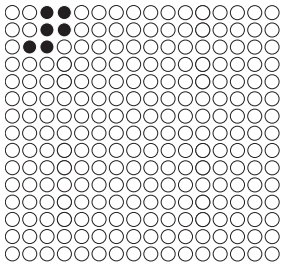

CHR091

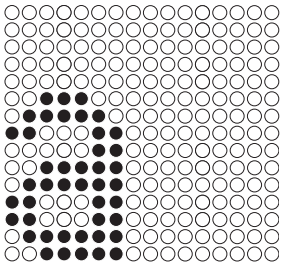

CHR092

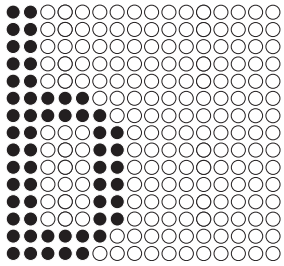

CHR093

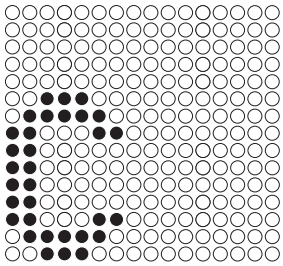

CHR094

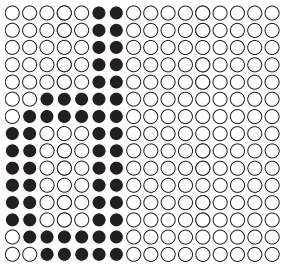

CHR095

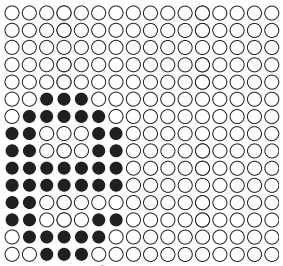

CHR096

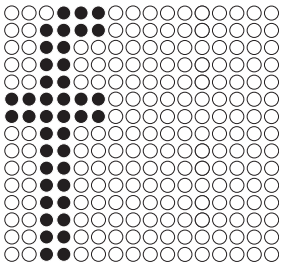

CHR097

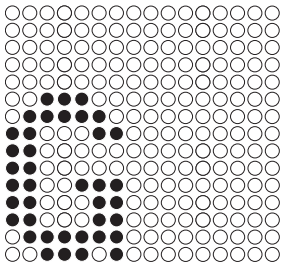

CHR098

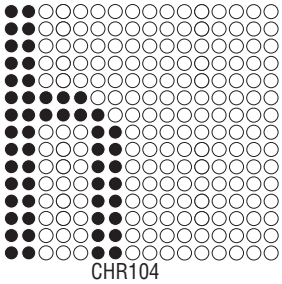

CHR099

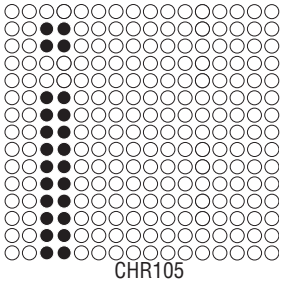

CHR100

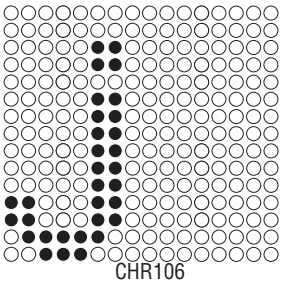

CHR101

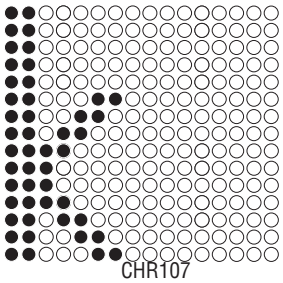

CHR102

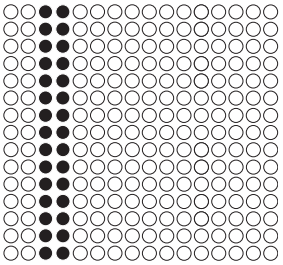

CHR103

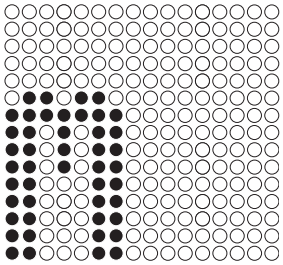

CHR104

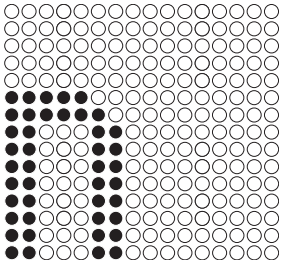

CHR105

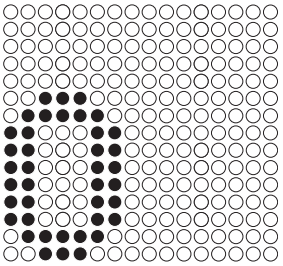

CHR106

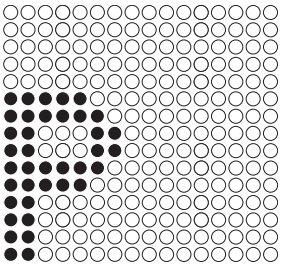

CHR107

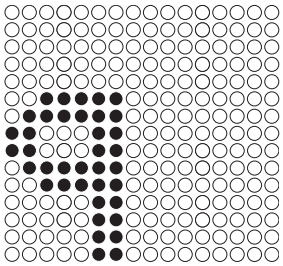

CHR108

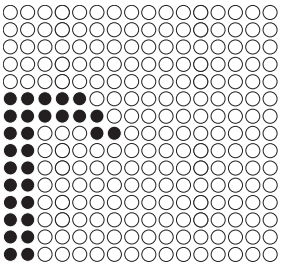

CHR109

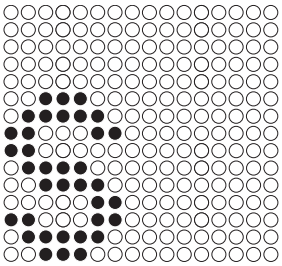

CHR110

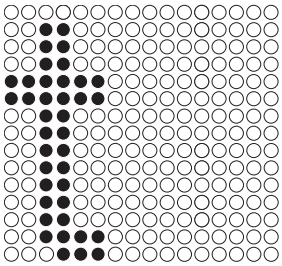

CHR111

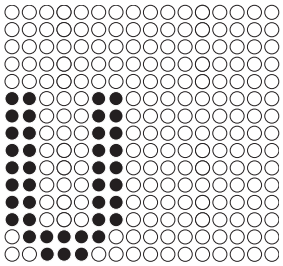

CHR112

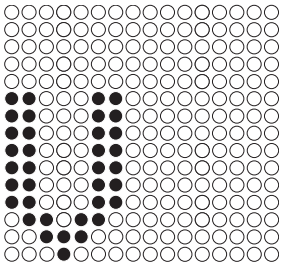

CHR113

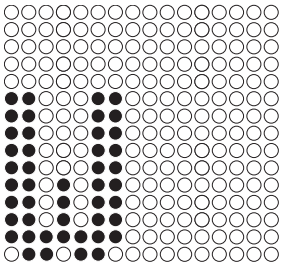

CHR114

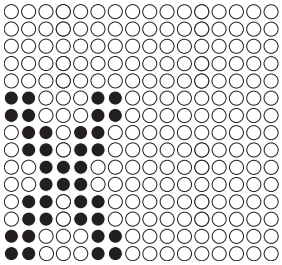

CHR115

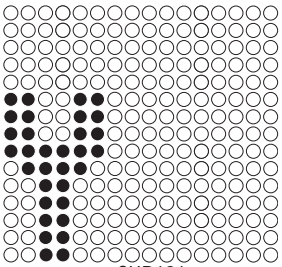

CHR116

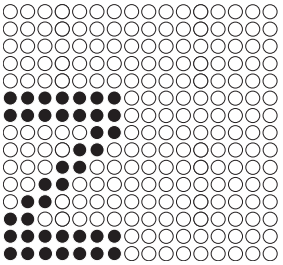

CHR117

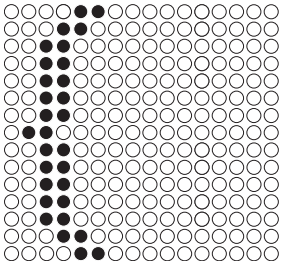

CHR118

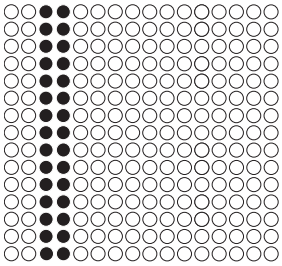

CHR119

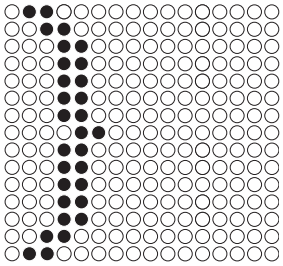

CHR120

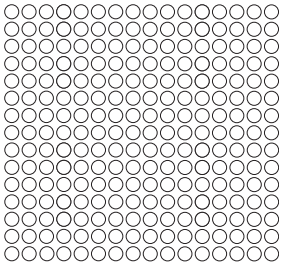

CHR121

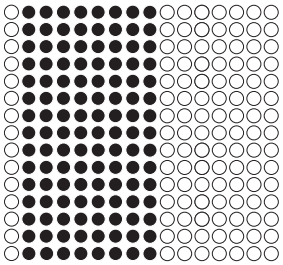

CHR122

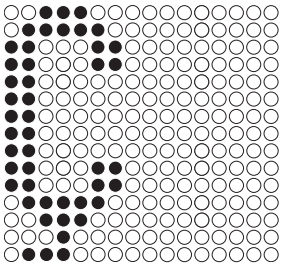

CHR123

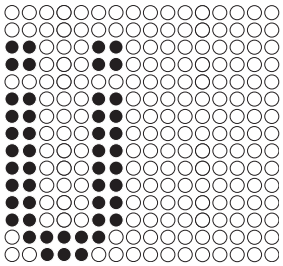

CHR124

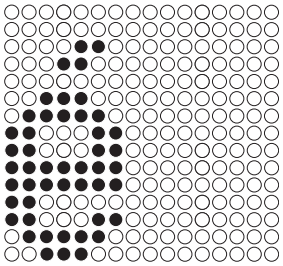

CHR125

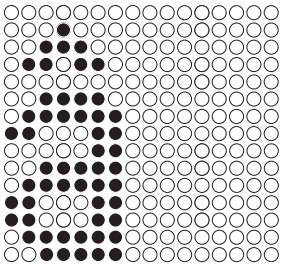

CHR126

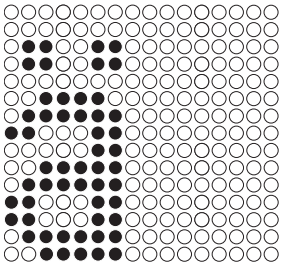

CHR127

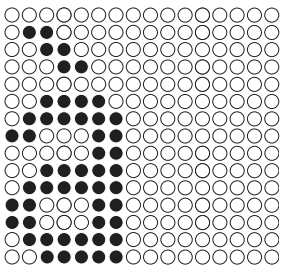

CHR128

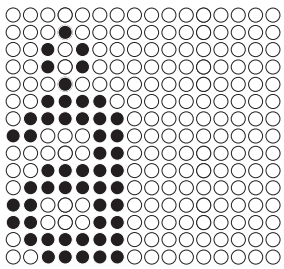

CHR129

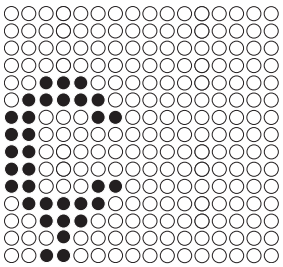

CHR130

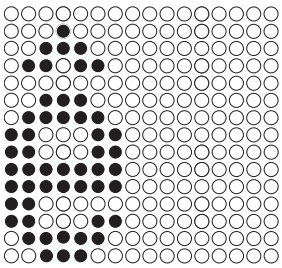

CHR131

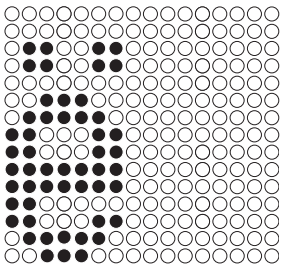

CHR132

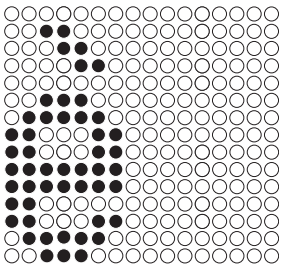

CHR133

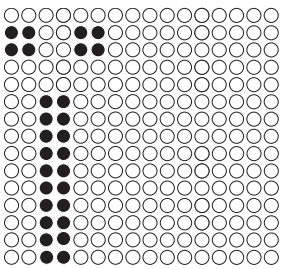

CHR134

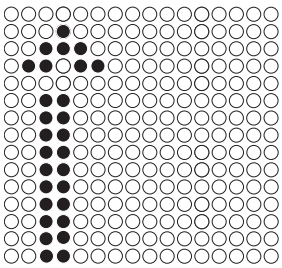

CHR135

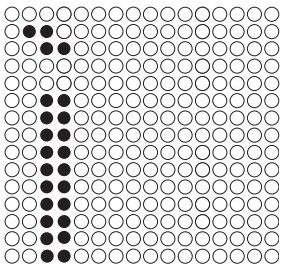

CHR136

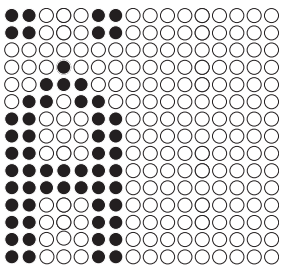

CHR137

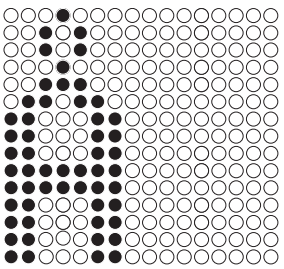

CHR138

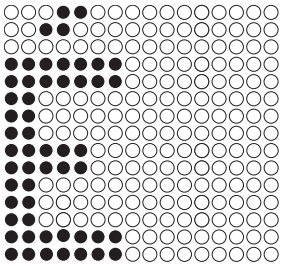

CHR139

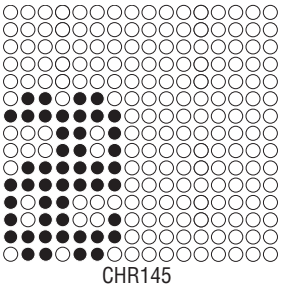

CHR140

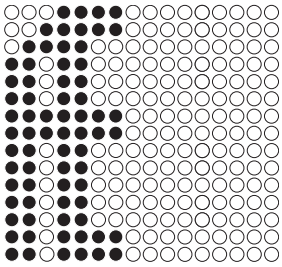

CHR141

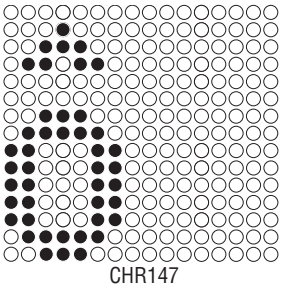

CHR142

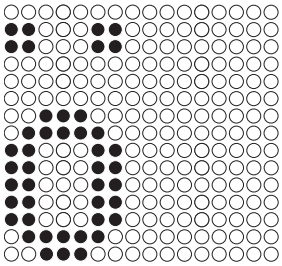

CHR143

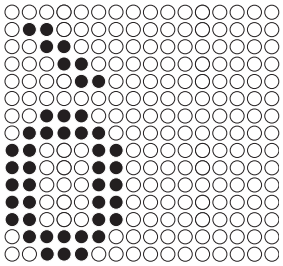

CHR144

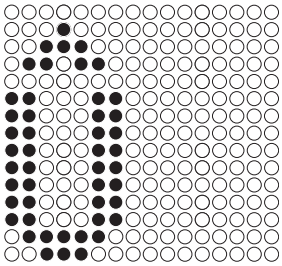

CHR145

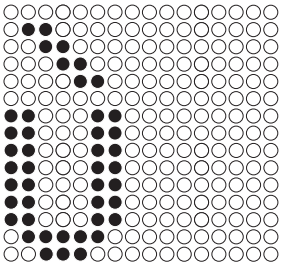

CHR146

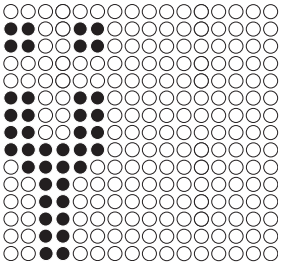

CHR147

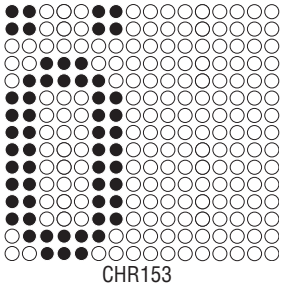

CHR148

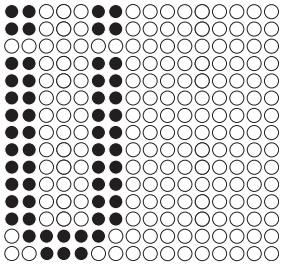

CHR149

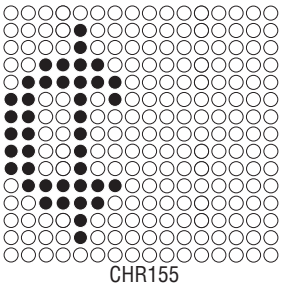

CHR150

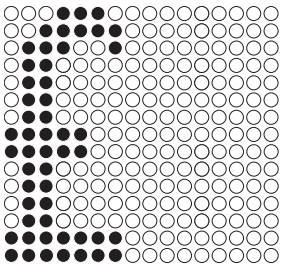

CHR151

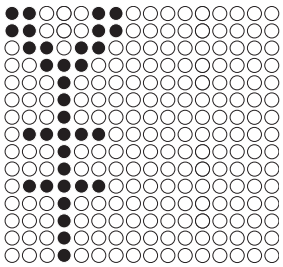

CHR152

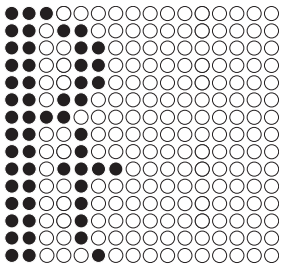

CHR153

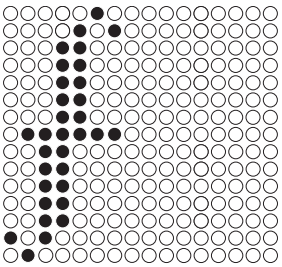

CHR154

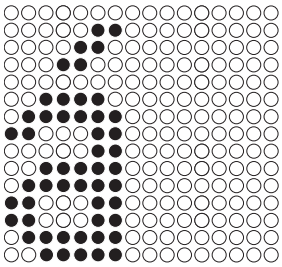

CHR155

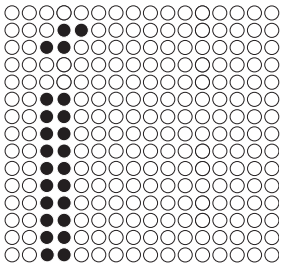

CHR156

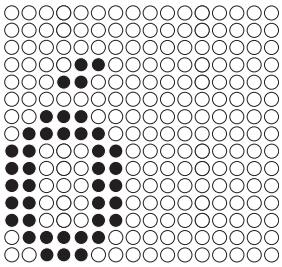

CHR157

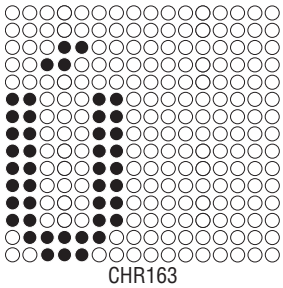

CHR158

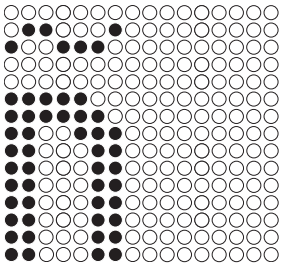

CHR159

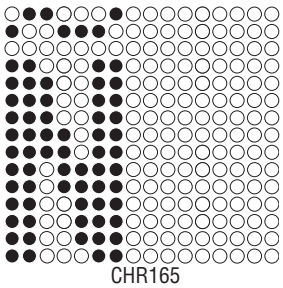

CHR160

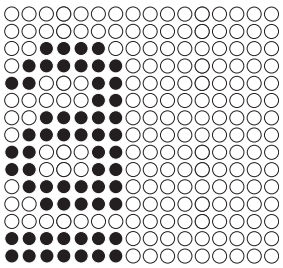

CHR161

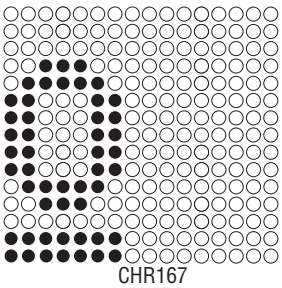

CHR162

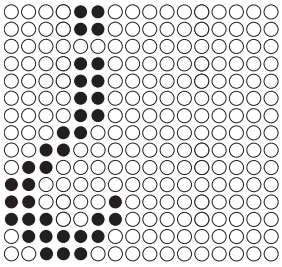

CHR163

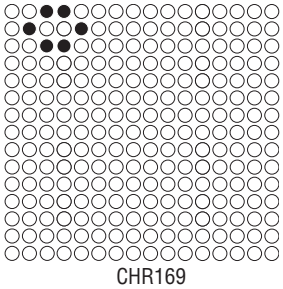

CHR164

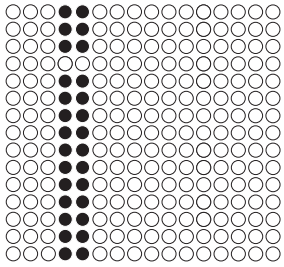

CHR165

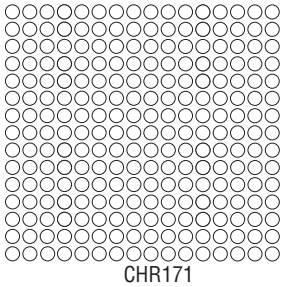

CHR166

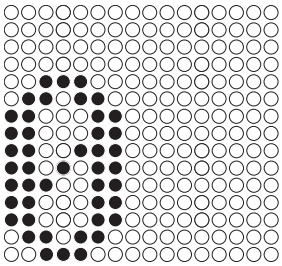

CHR167

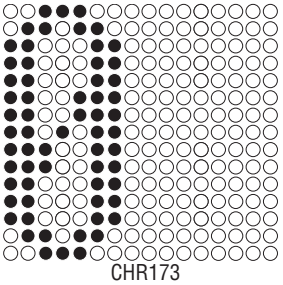

CHR168

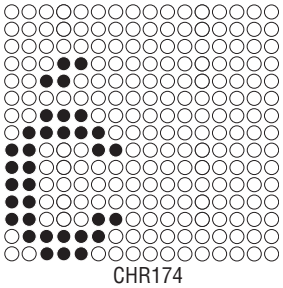

CHR169

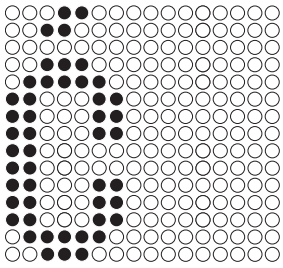

CHR170

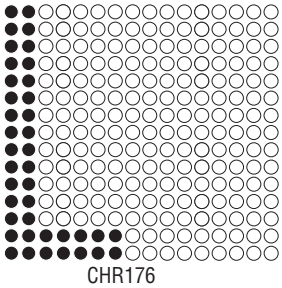

CHR171

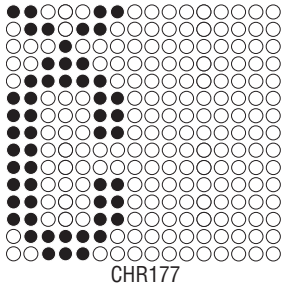

CHR172

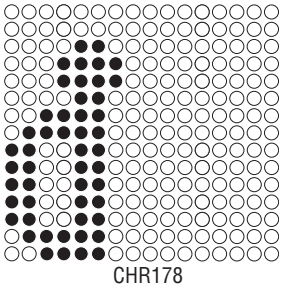

CHR173

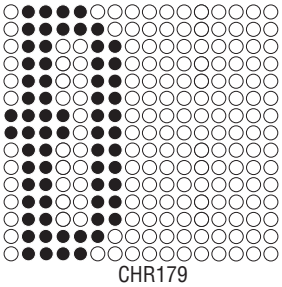

CHR174

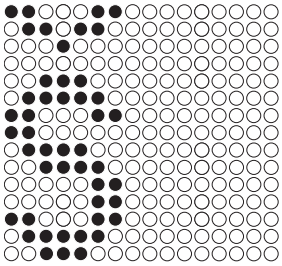

CHR175

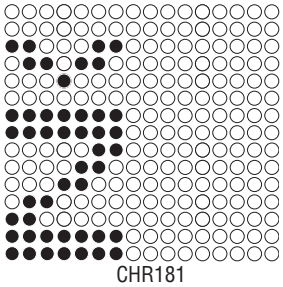

CHR176

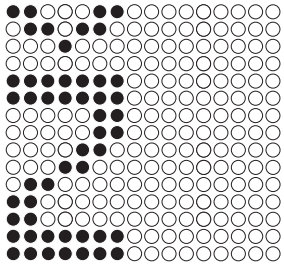

CHR177

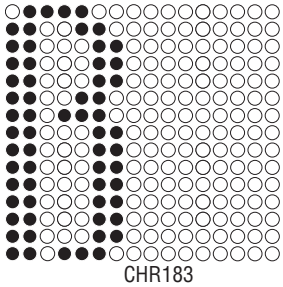

CHR178

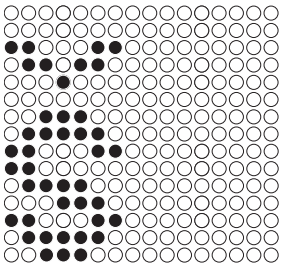

CHR179

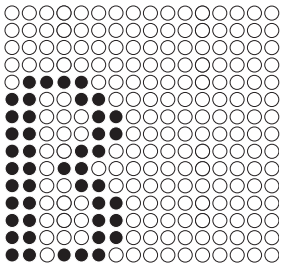

CHR180

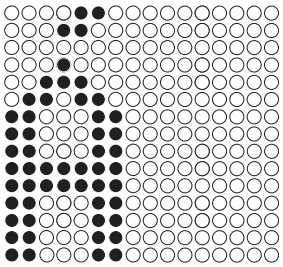

CHR181

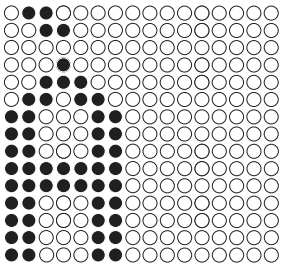

CHR182

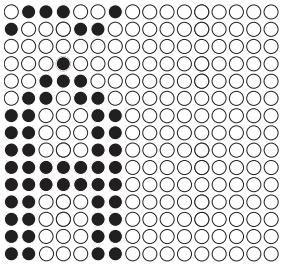

CHR183

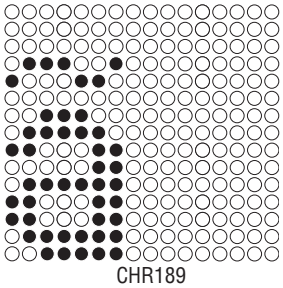

CHR184

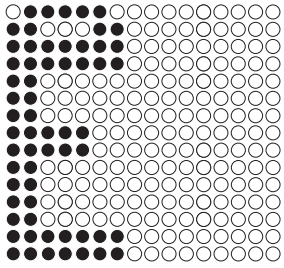

CHR185

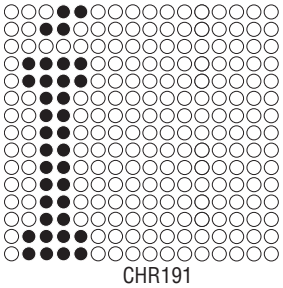

CHR186

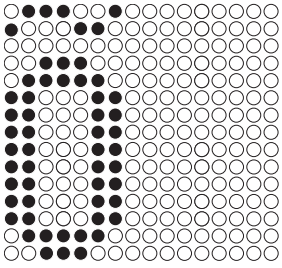

CHR187

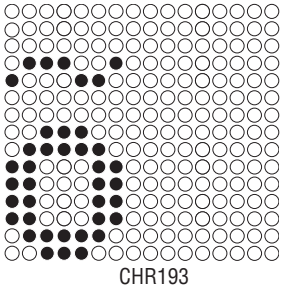

CHR188


CHR189

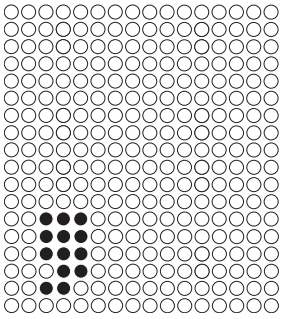

CHR190

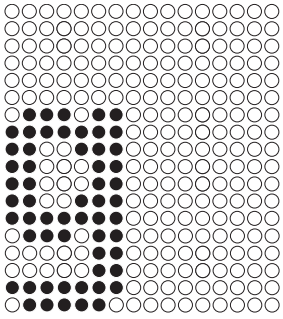

CHR191

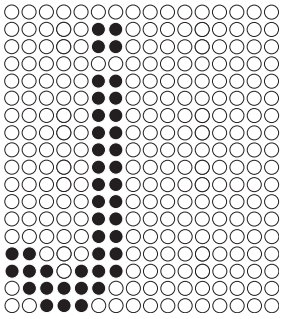

CHR192

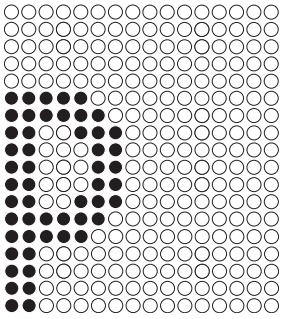

CHR193

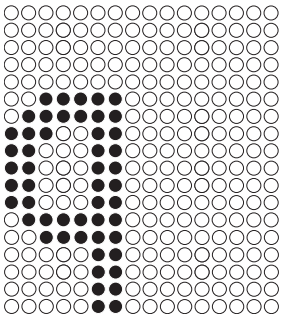
7.13.13 15-High True Descender Regular

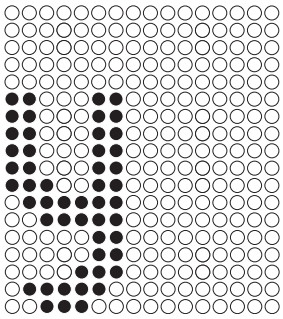

CHR15C


CHR15G

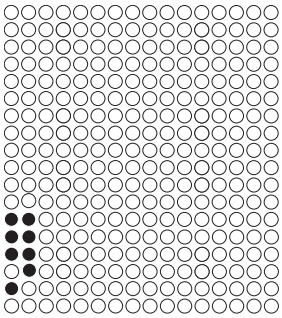

CHR15J

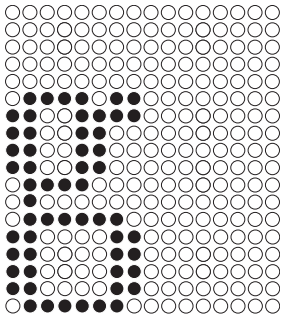

CHR15P

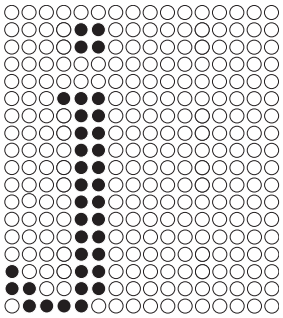

CHR15Q

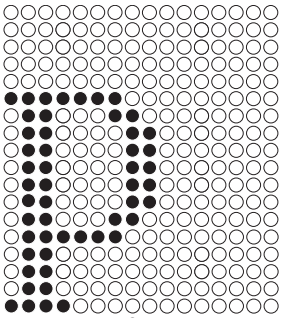

CHR15Y

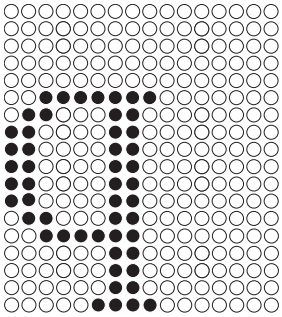
7.13.14 15-High True Descender Fancy

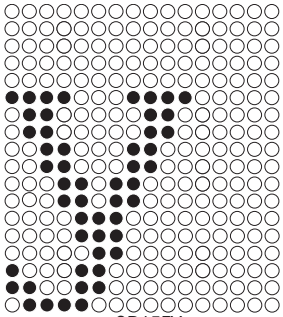

CR15FC


CR15FG

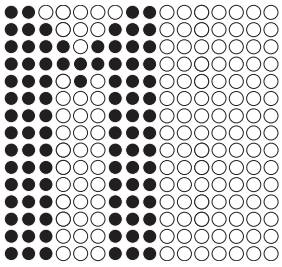

CR15FJ

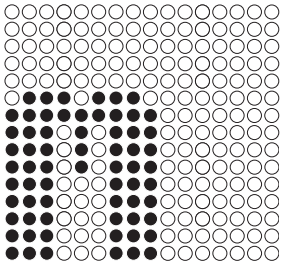

CR15FP

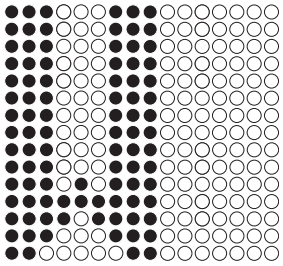

CR15FQ

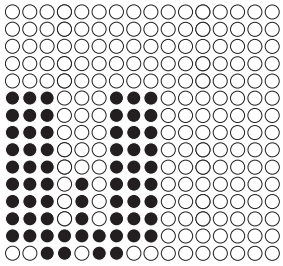

CR15FY

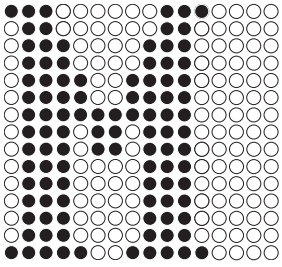
7.13.15 15-High Fat Character

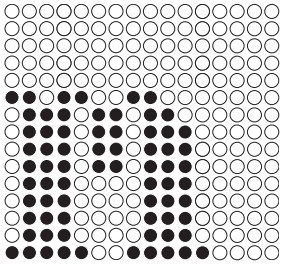

FATM

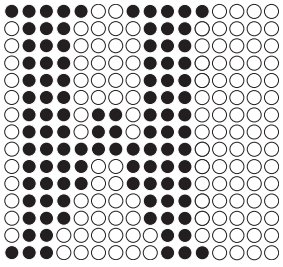

FATM1

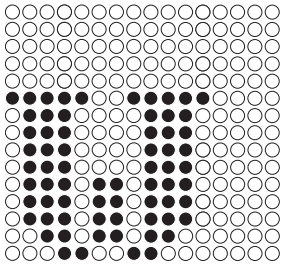

FATW


FATW1

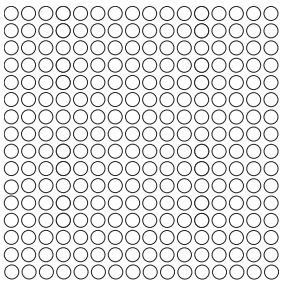

FFATM

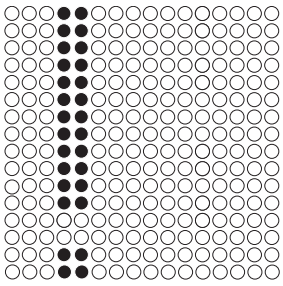

FFATM1

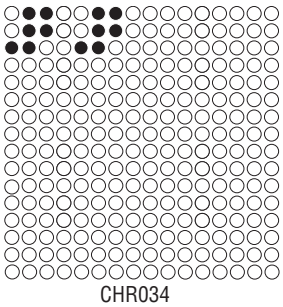

FFATW

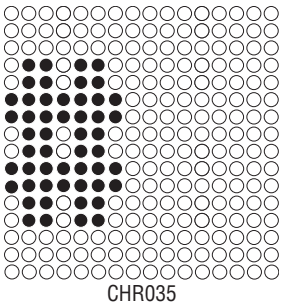

FFATW!

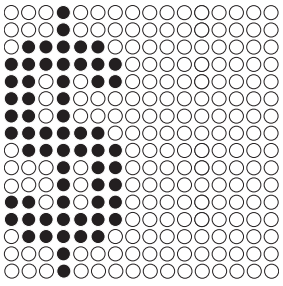
7.13.16 16-High Regular (SS16)

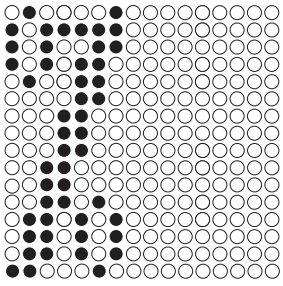

CHR032

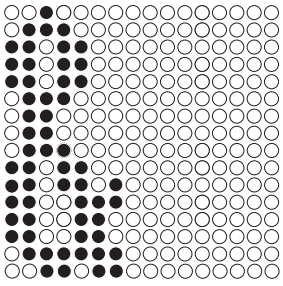

CHR033

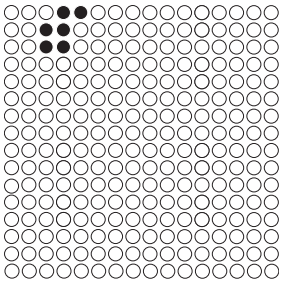

CHR034

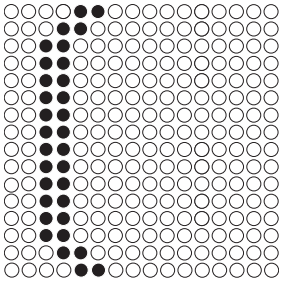

CHR035

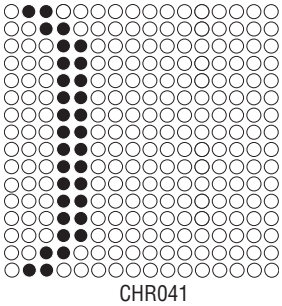

CHR036

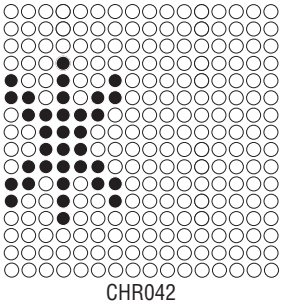

CHR037

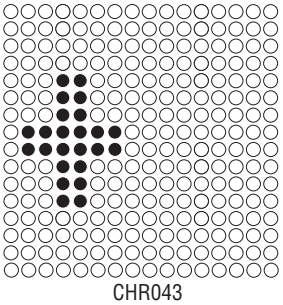

CHR038

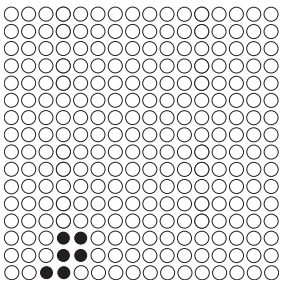

CHR039

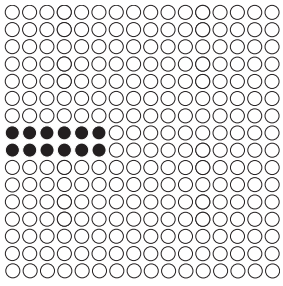

CHR040

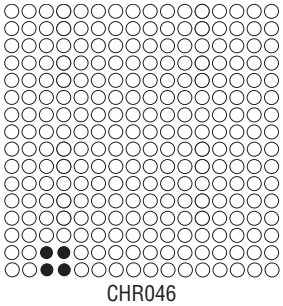

CHR041

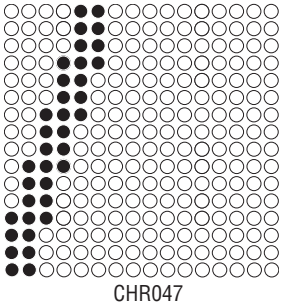

CHR042

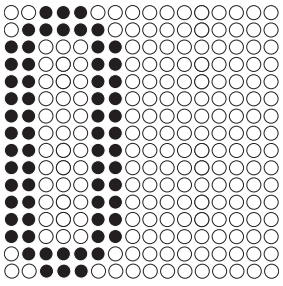

CHR043

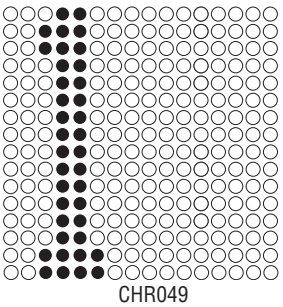

CHR044

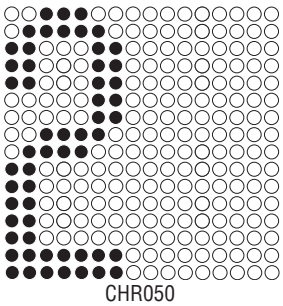

CHR045

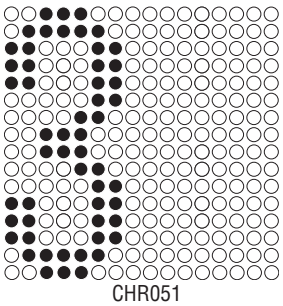

CHR046

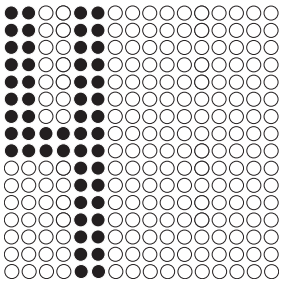

CHR047

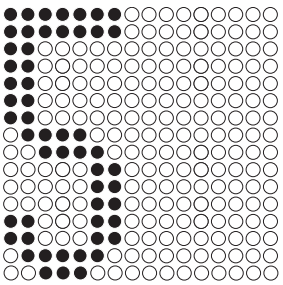

CHR048

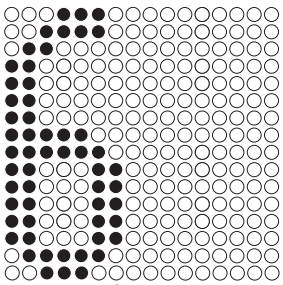

CHR049

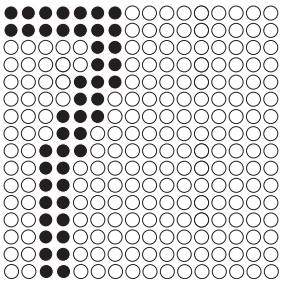

CHR050

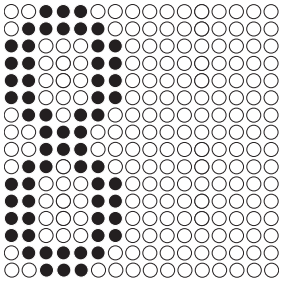

CHR051

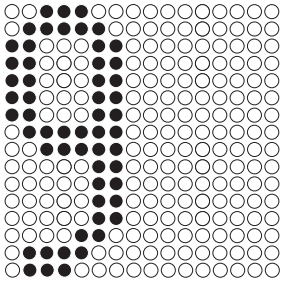

CHR052

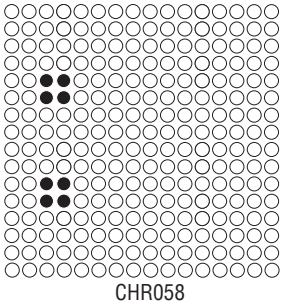

CHR053

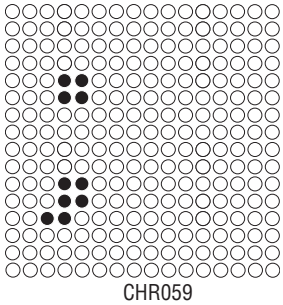

CHR054

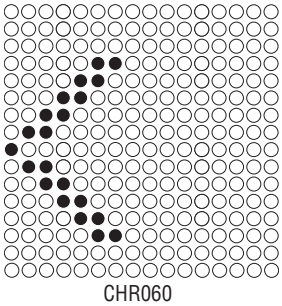

CHR055

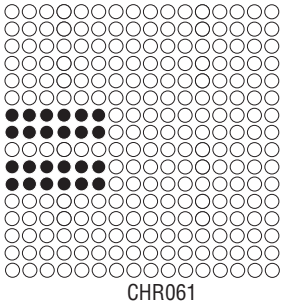

CHR056

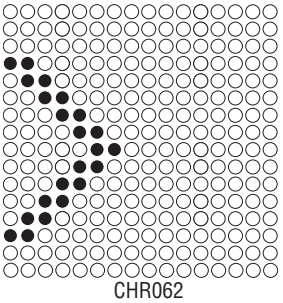

CHR057

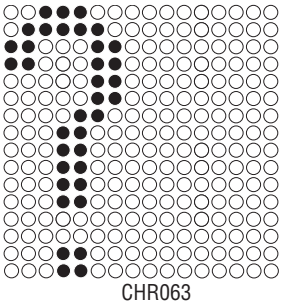

CHR058

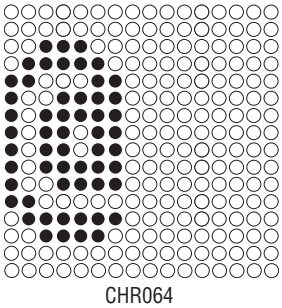

CHR059

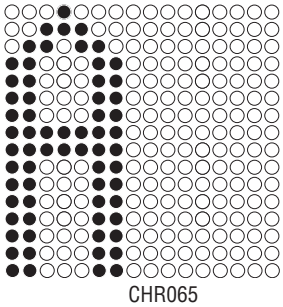

CHR060

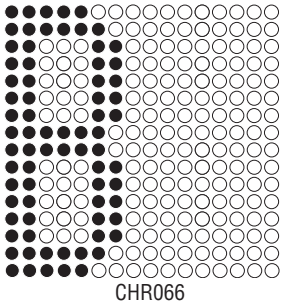

CHR061

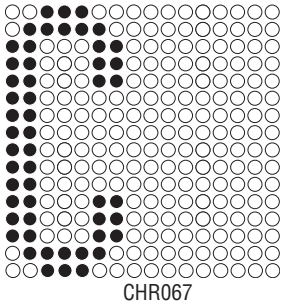

CHR062

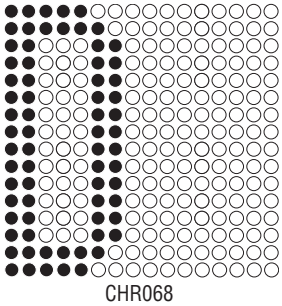

CHR063

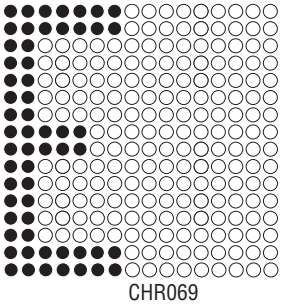

CHR064

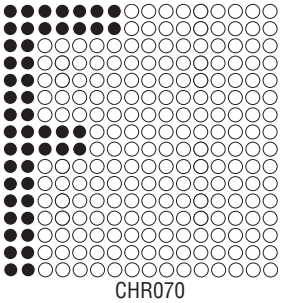

CHR065

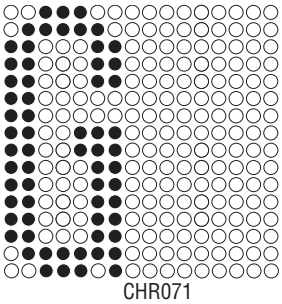

CHR066

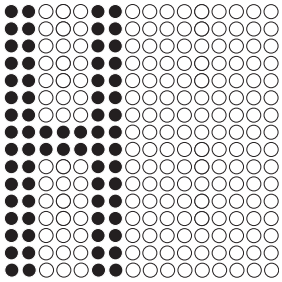

CHR067

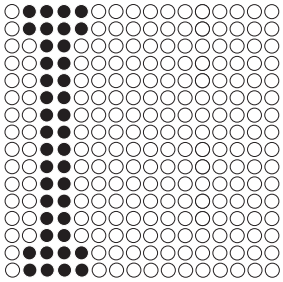

CHR068

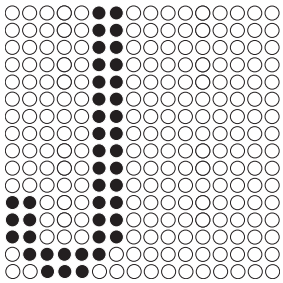

CHR069

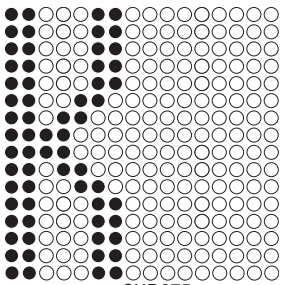

CHR070

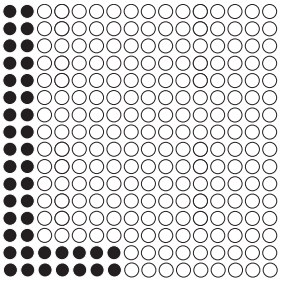

CHR071

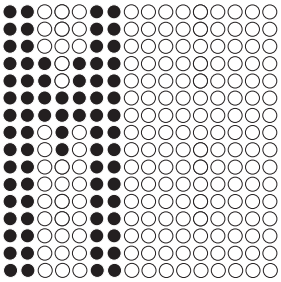

CHR072

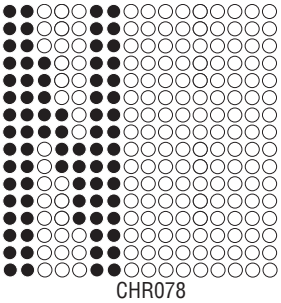

CHR073

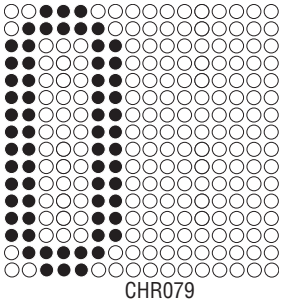

CHR074

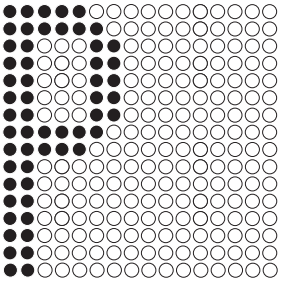

CHR075

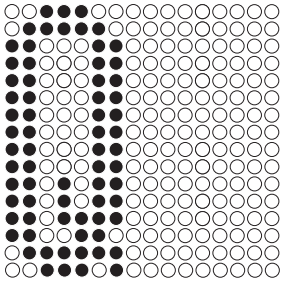

CHR076

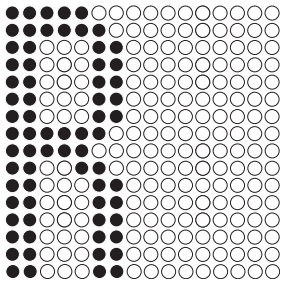

CHR077

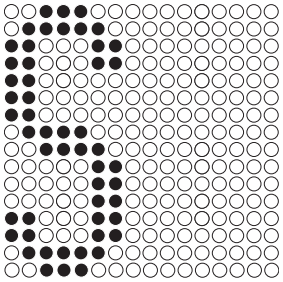

CHR078

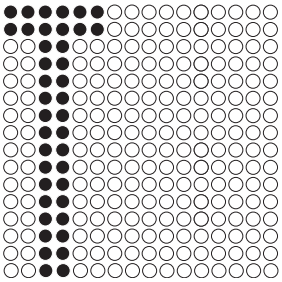

CHR079

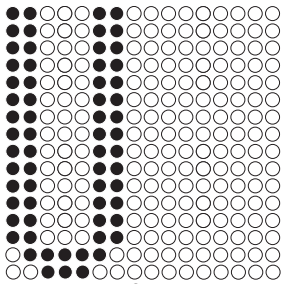

CHR080

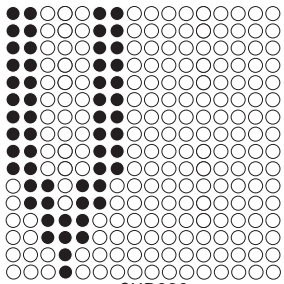

CHR081

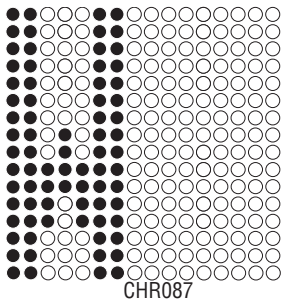

CHR082

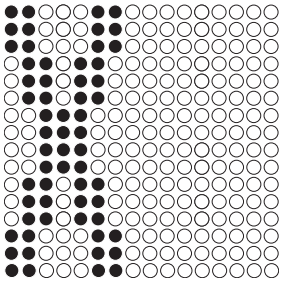

CHR083

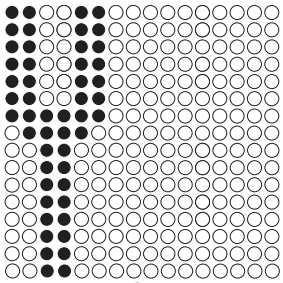

CHR084

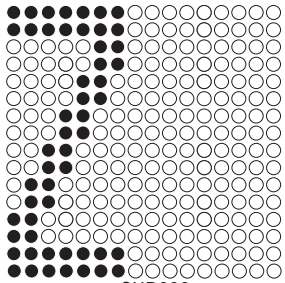

CHR085

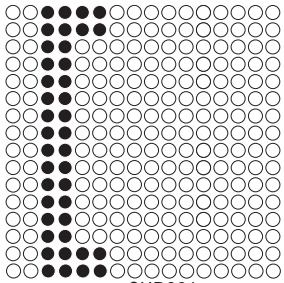

CHR086

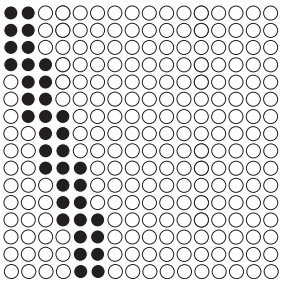

CHR087

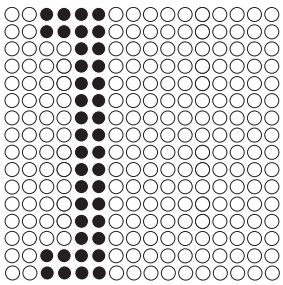

CHR088

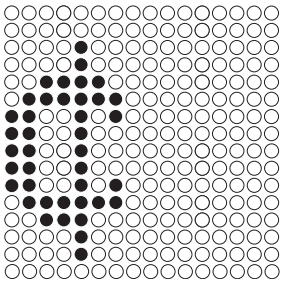

CHR089

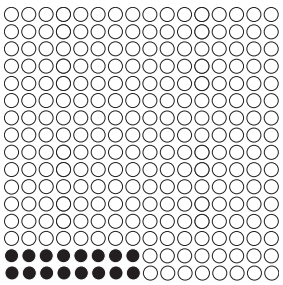

CHR090

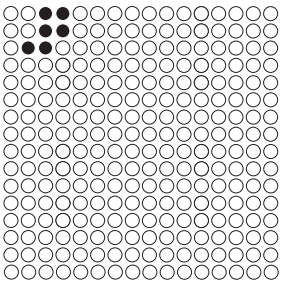

CHR091

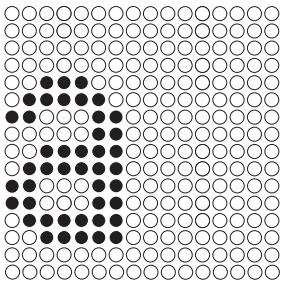

CHR092

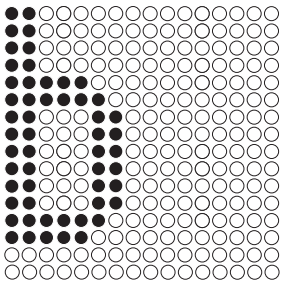

CHR093

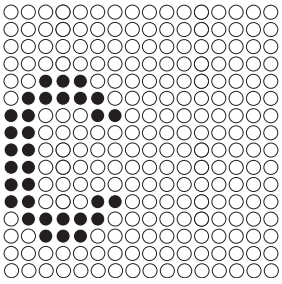

CHR094

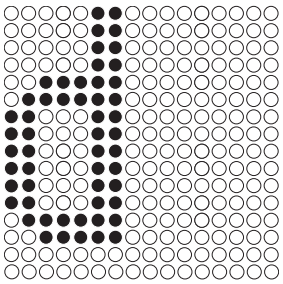

CHR095

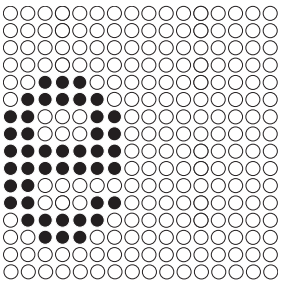

CHR096

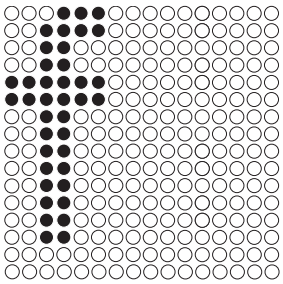

CHR097

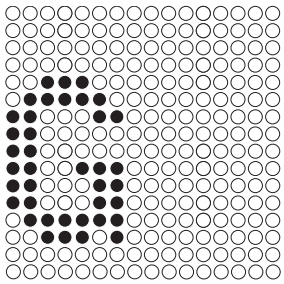

CHR098

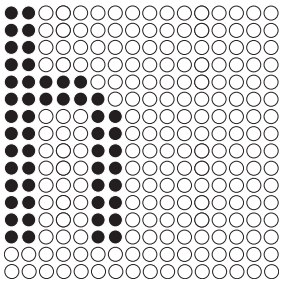

CHR099

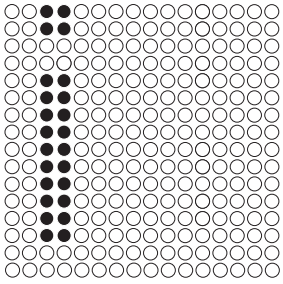

CHR100

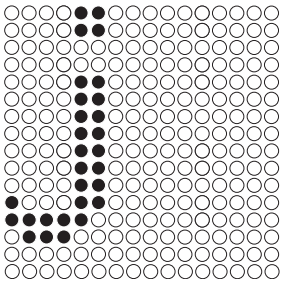

CHR101

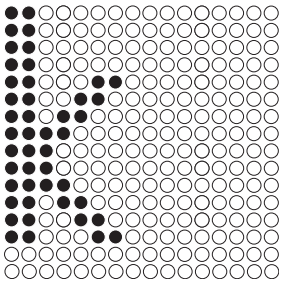

CHR102

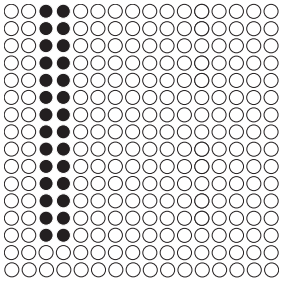

CHR103

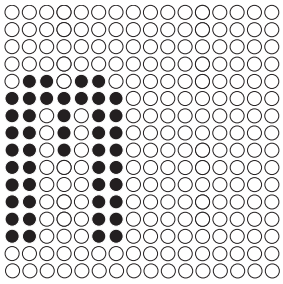

CHR104

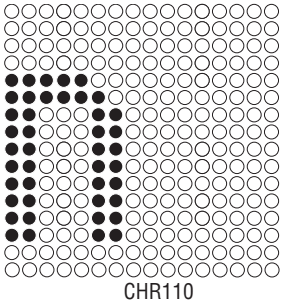

CHR105

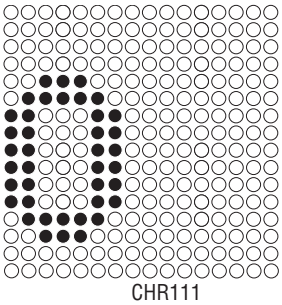

CHR106

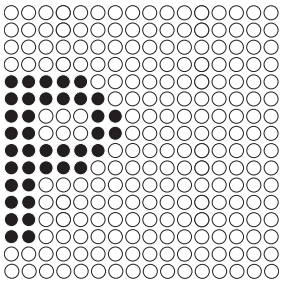

CHR107


CHR108

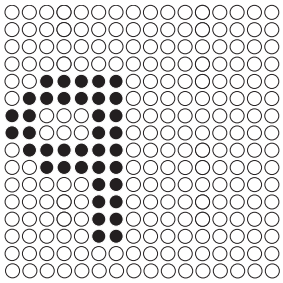

CHR109


CHR110

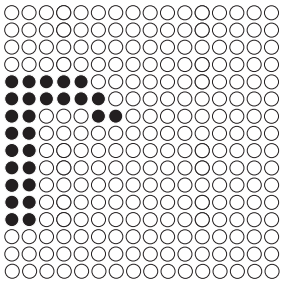

CHR111



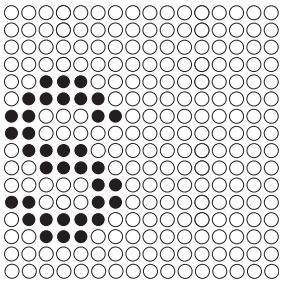
CHR112



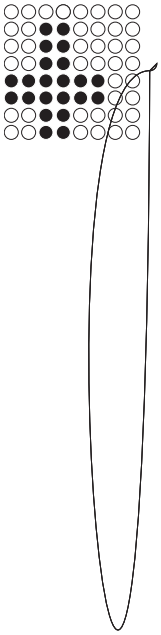
CHR113

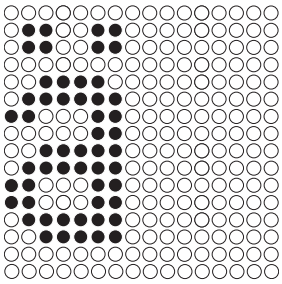


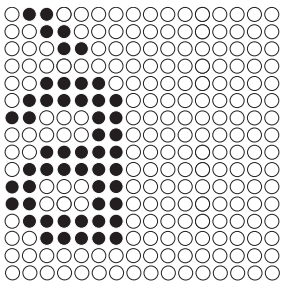
CHR114

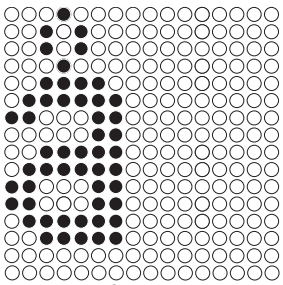


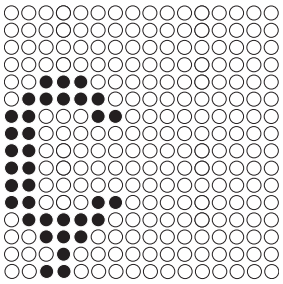
CHR115

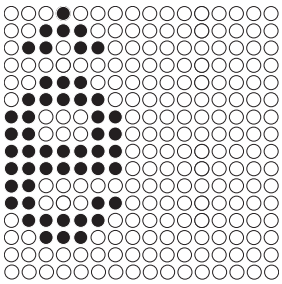


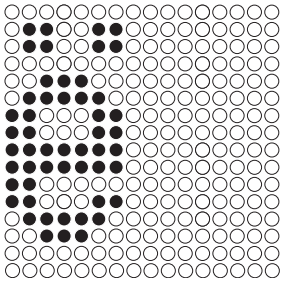

CHR132

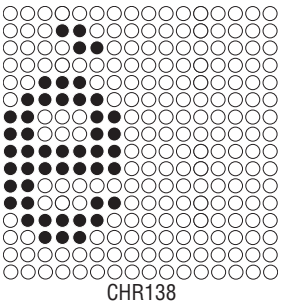

CHR133

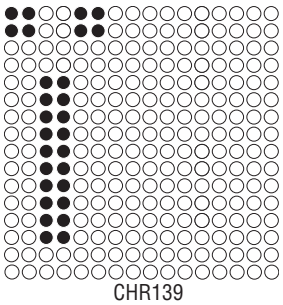

CHR134

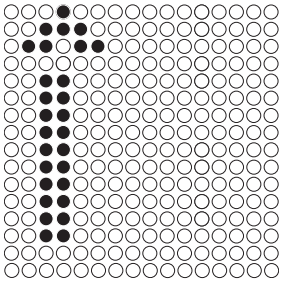

CHR135

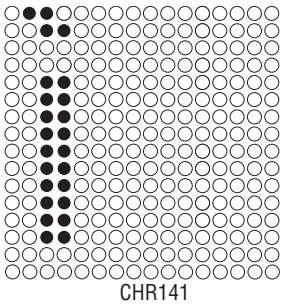

CHR136

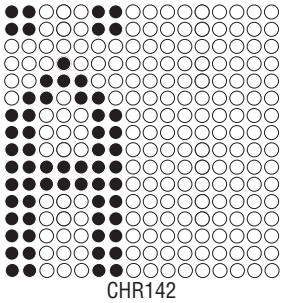

CHR137

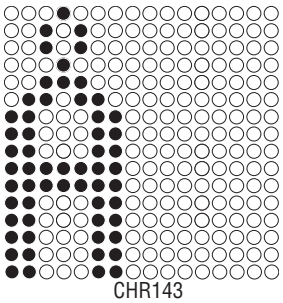

CHR138

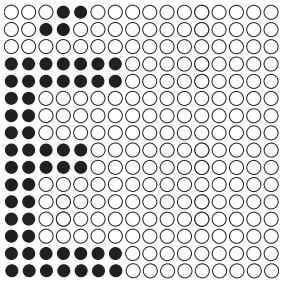

CHR139

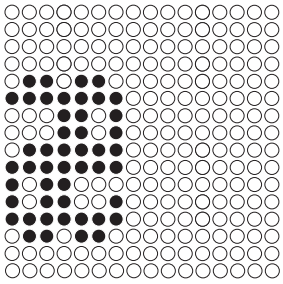

CHR140

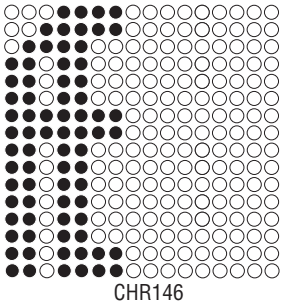

CHR141

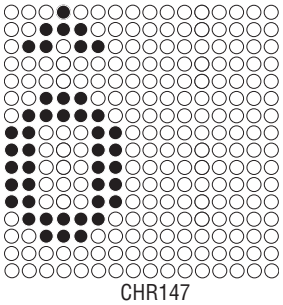

CHR142

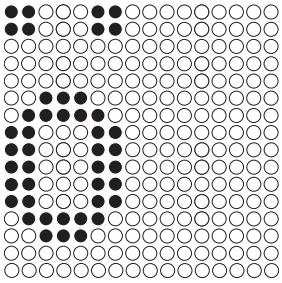

CHR143

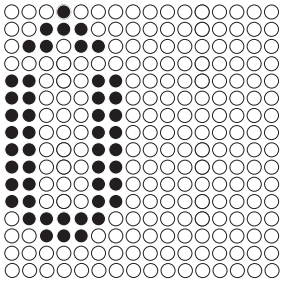

CHR144

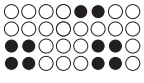

CHR145

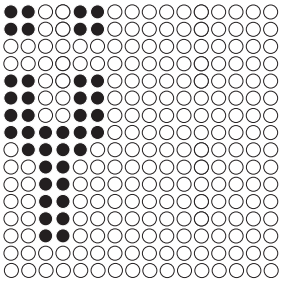

CHR146

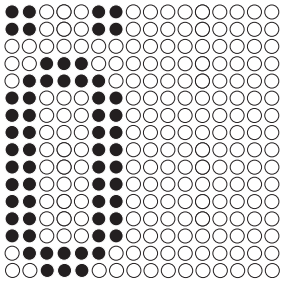

CHR147

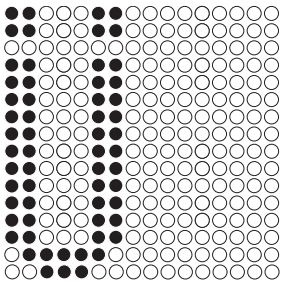


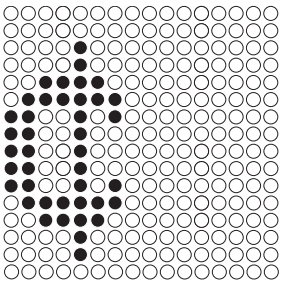


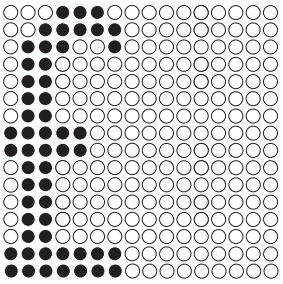


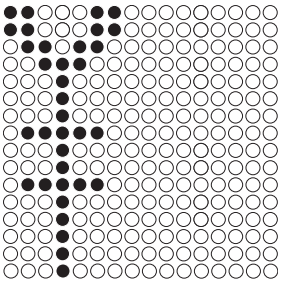

CHR152

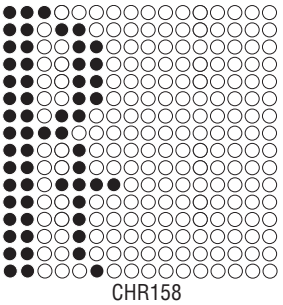

CHR153

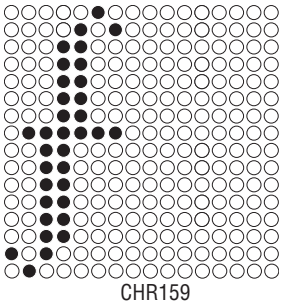

CHR154

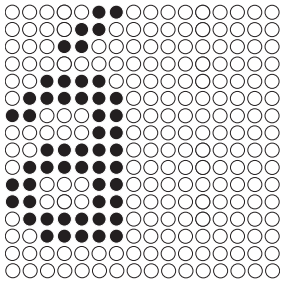

CHR155

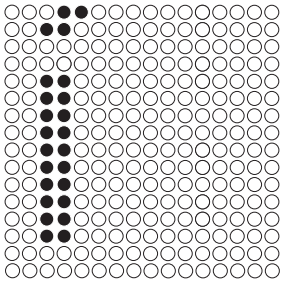

CHR156

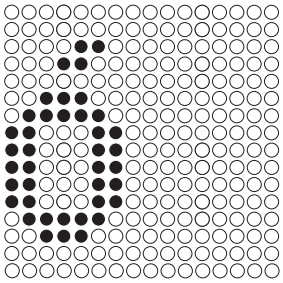

CHR157

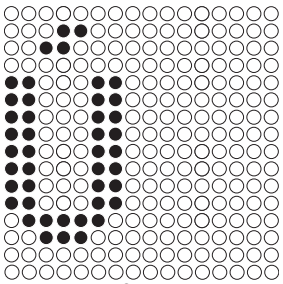

CHR158

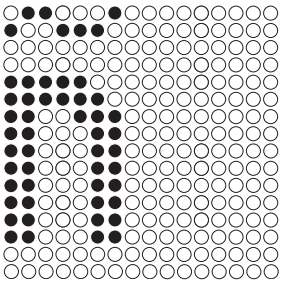

CHR159

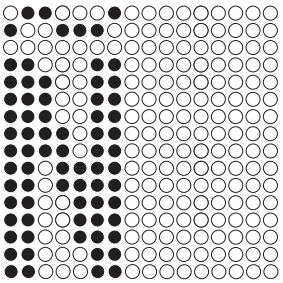

CHR160

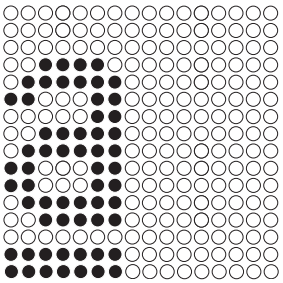

CHR161

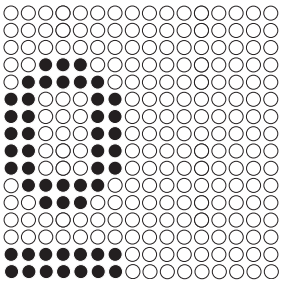

CHR162

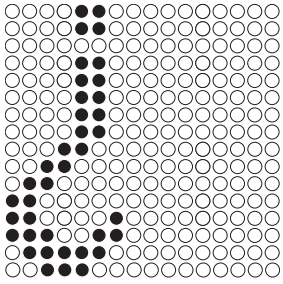

CHR163

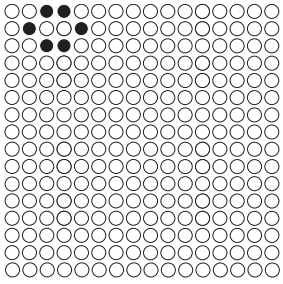

CHR164

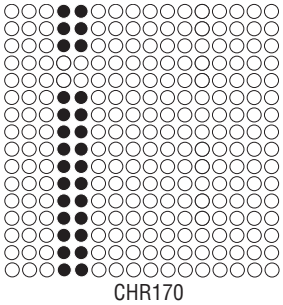

CHR165

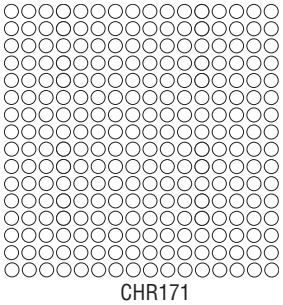

CHR166

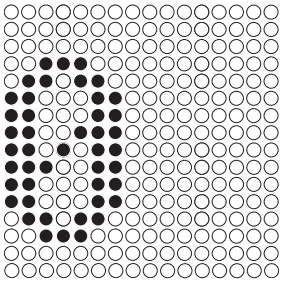

CHR167

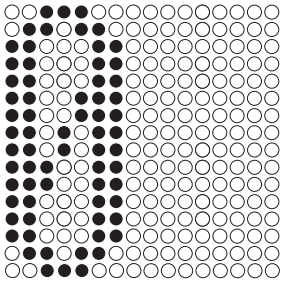

CHR168

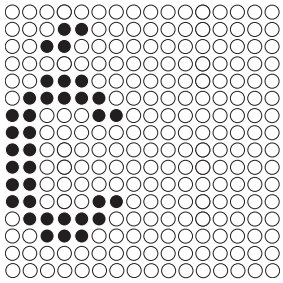

CHR169

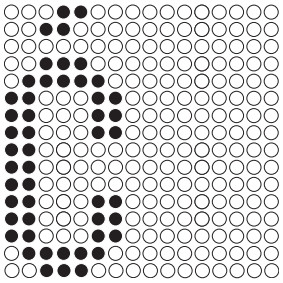

CHR170

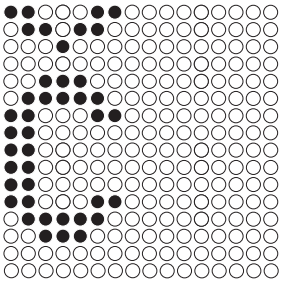

CHR171

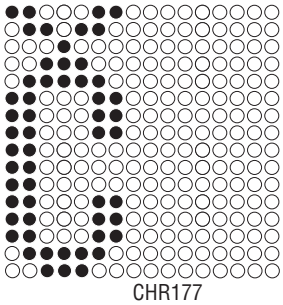

CHR172

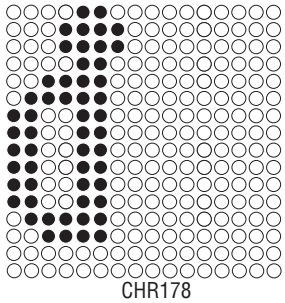

CHR173

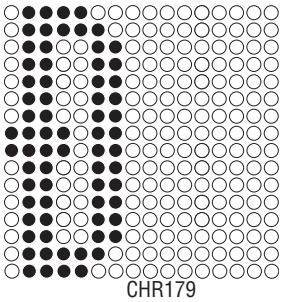

CHR174

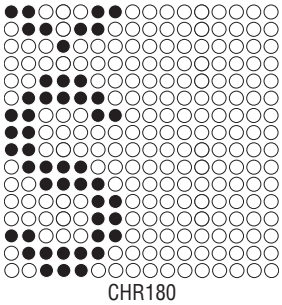

CHR175

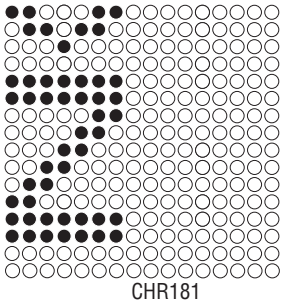

CHR176

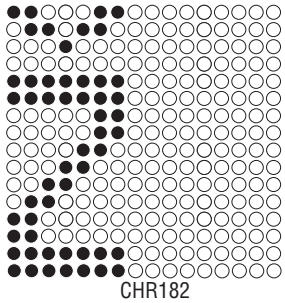

CHR177

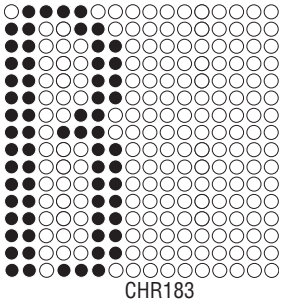

CHR178

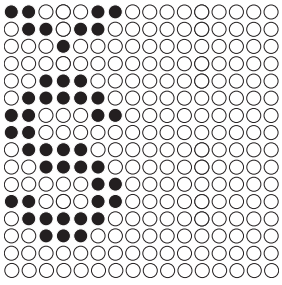

CHR179

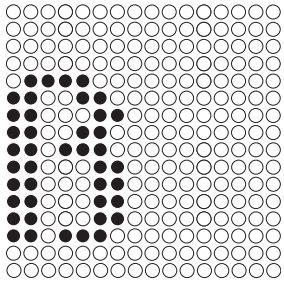

CHR180

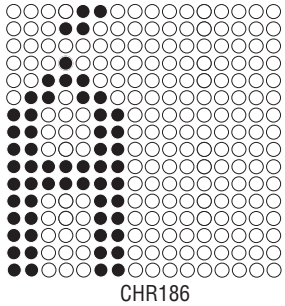

CHR181

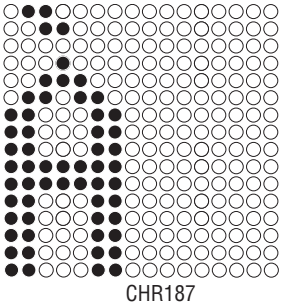

CHR182

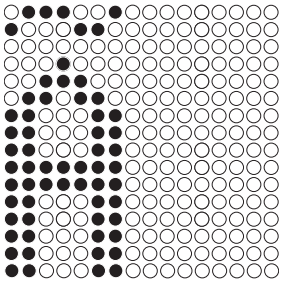

CHR183

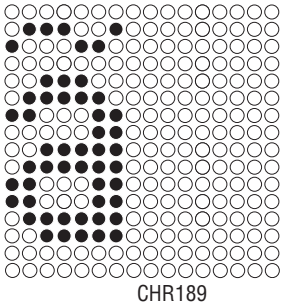

CHR184

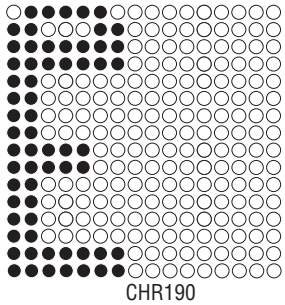

CHR185

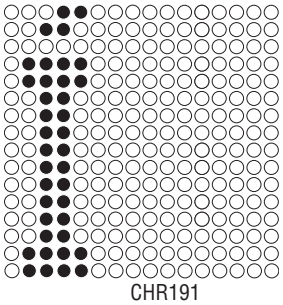

CHR186

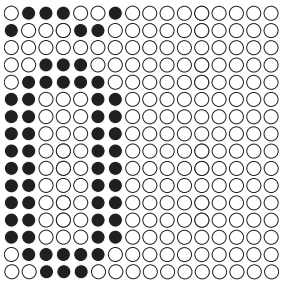

CHR187

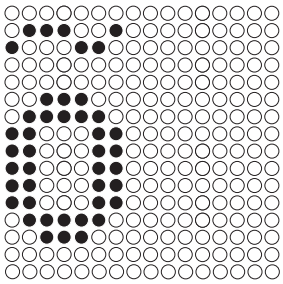

CHR188


CHR189

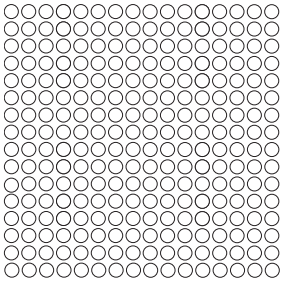

CHR190

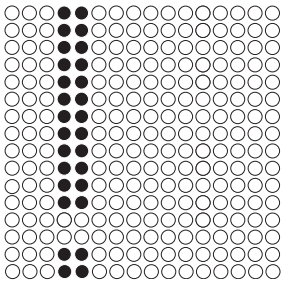

CHR191

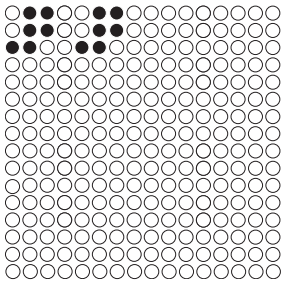

CHR192

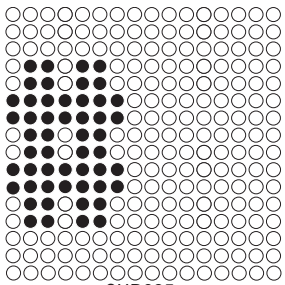

CHR193

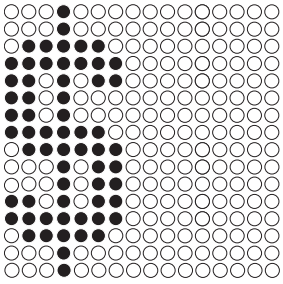
7.13.17 16-High Fancy (SF16)

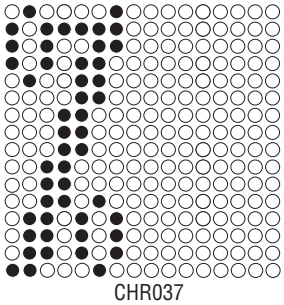

CHR032

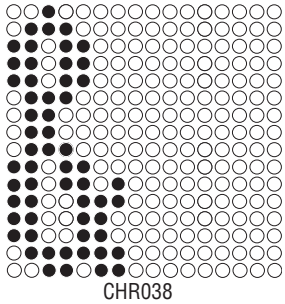

CHR033

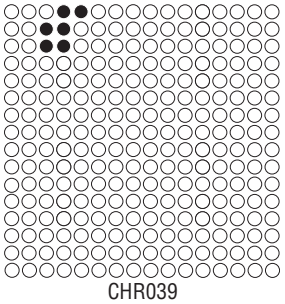

CHR034

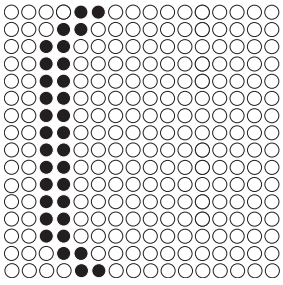

CHR035

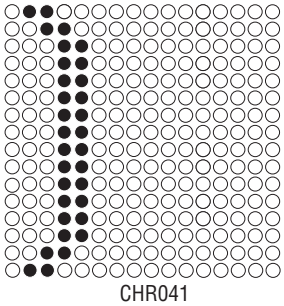

CHR036

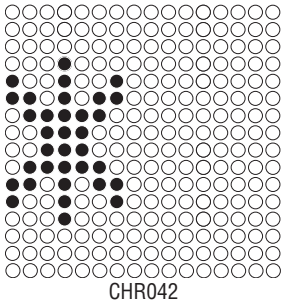

CHR037

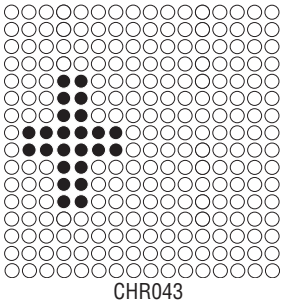

CHR038

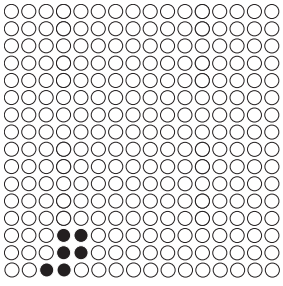

CHR039

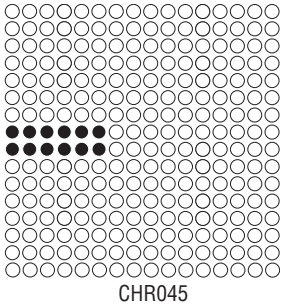

CHR040

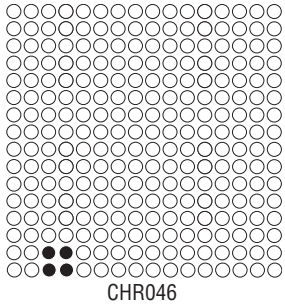

CHR041

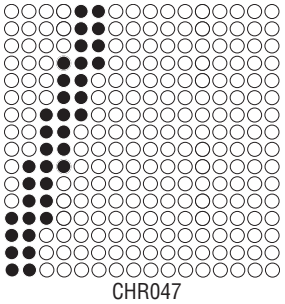

CHR042

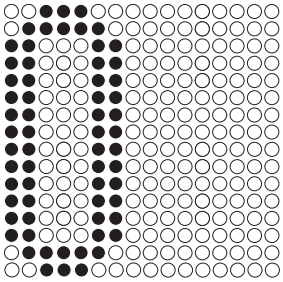

CHR043

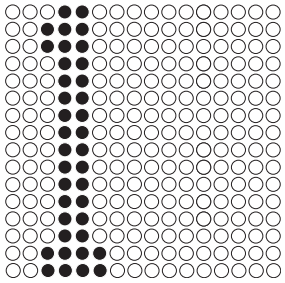

CHR044

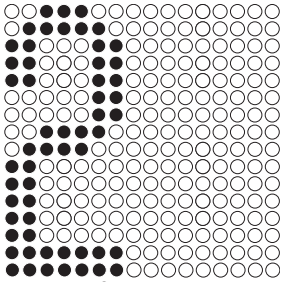

CHR045

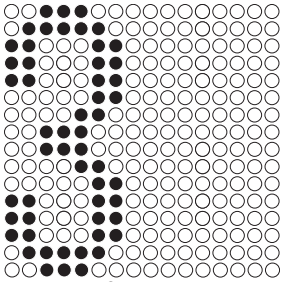

CHR046

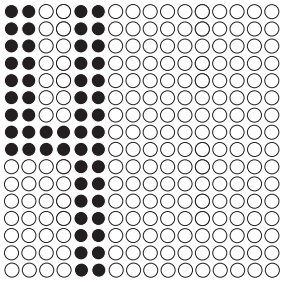

CHR047

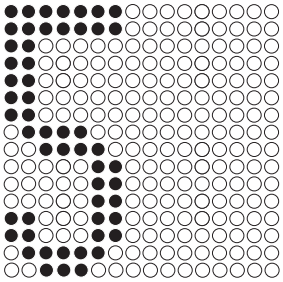

CHR048

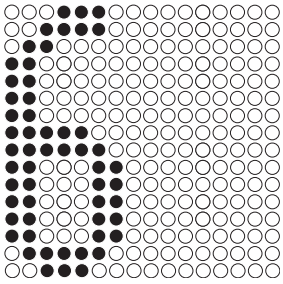

CHR049

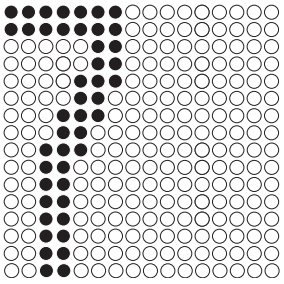

CHR050

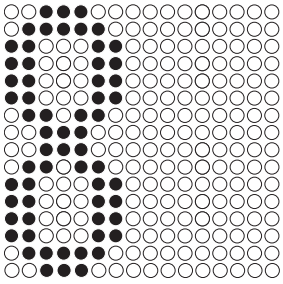

CHR051

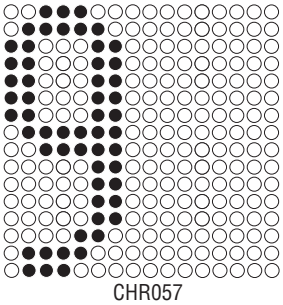

CHR052

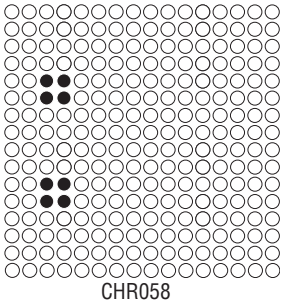

CHR053

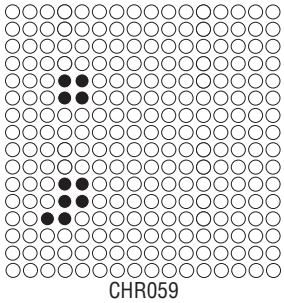

CHR054

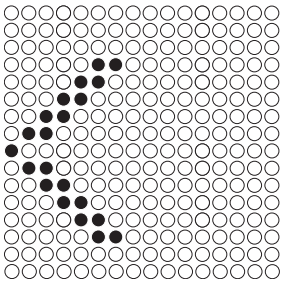

CHR055

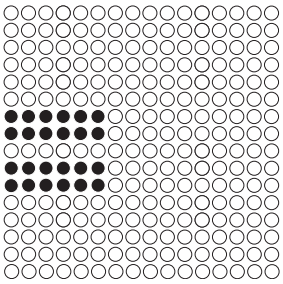

CHR056

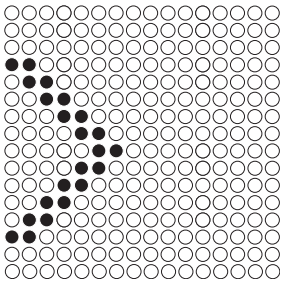

CHR057

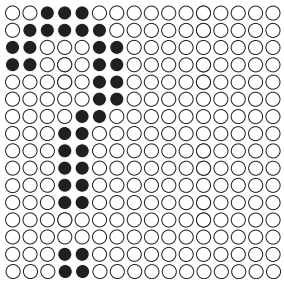

CHR058

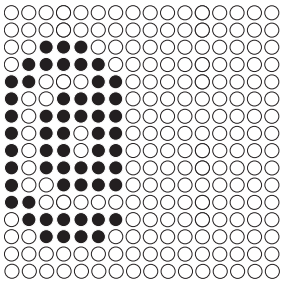

CHR059

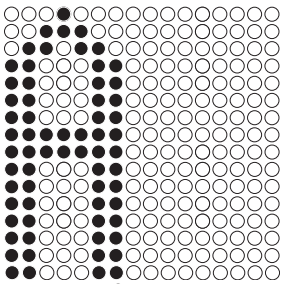

CHR060

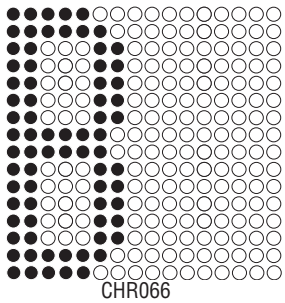

CHR061

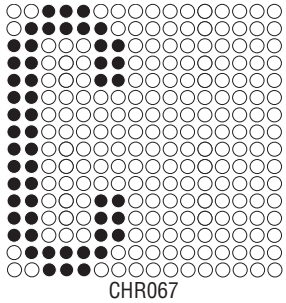

CHR062

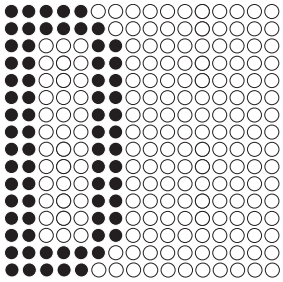

CHR063

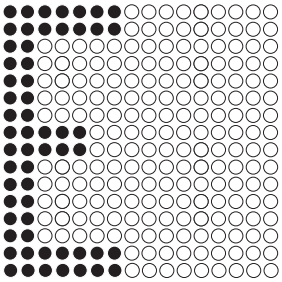

CHR064

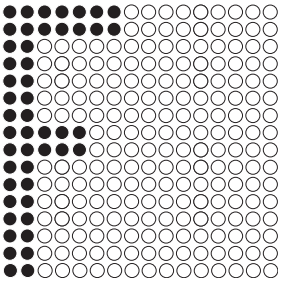

CHR065

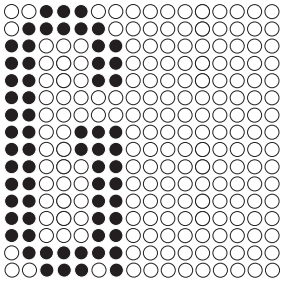

CHR066

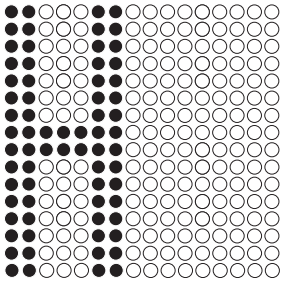

CHR067

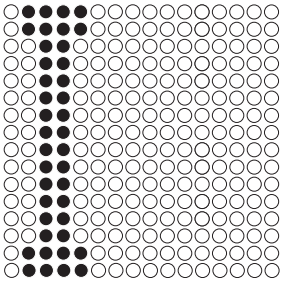

CHR068

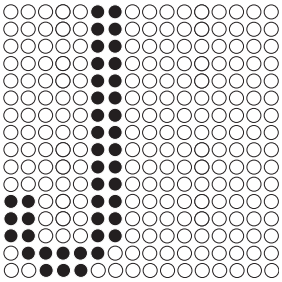

CHR069

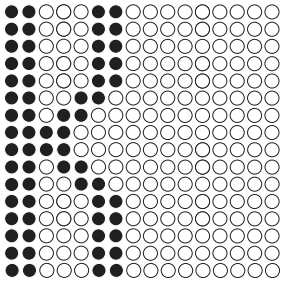

CHR070

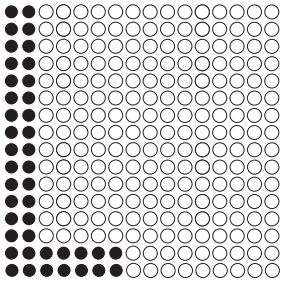

CHR071

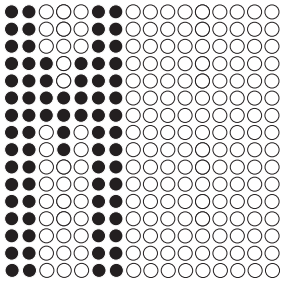

CHR072

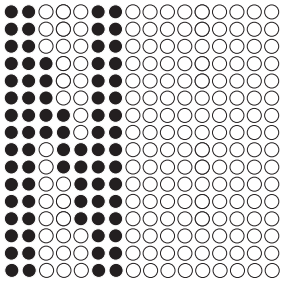

CHR073

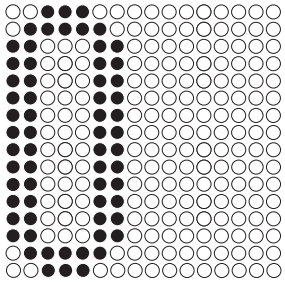

CHR074

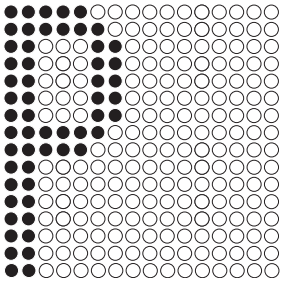

CHR075

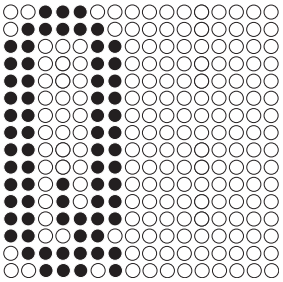

CHR076

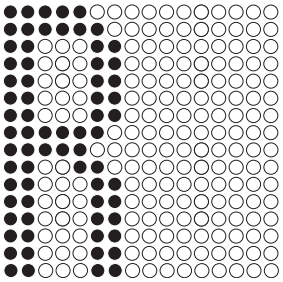

CHR077

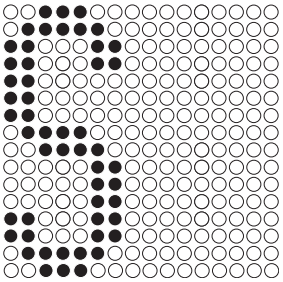

CHR078

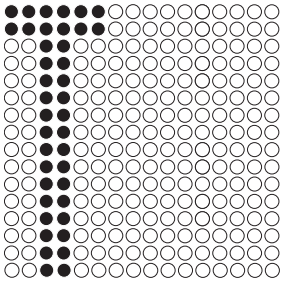

CHR079

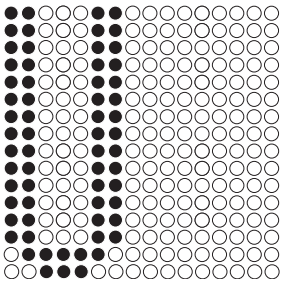

CHR080

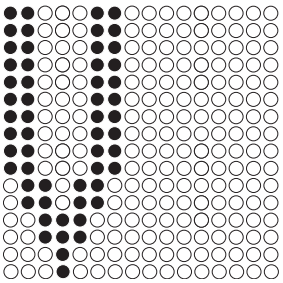

CHR081

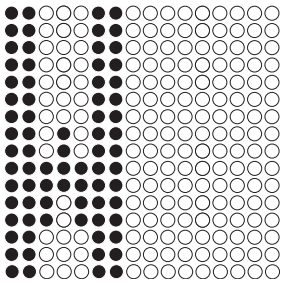

CHR082

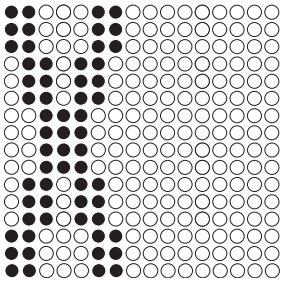

CHR083

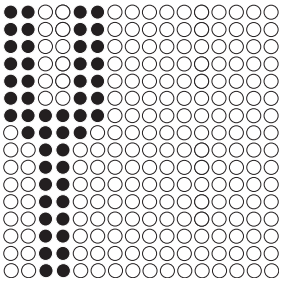

CHR084

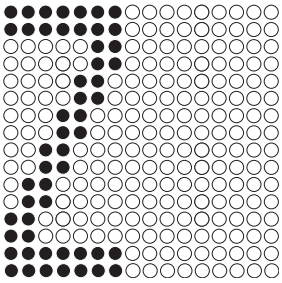

CHR085

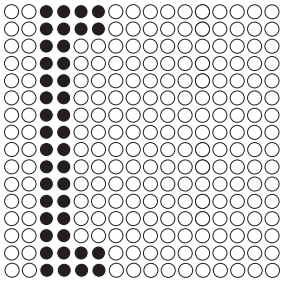

CHR086

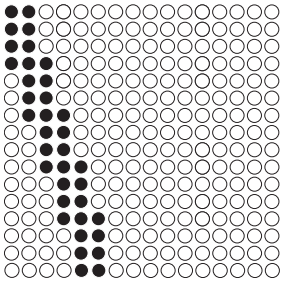

CHR087

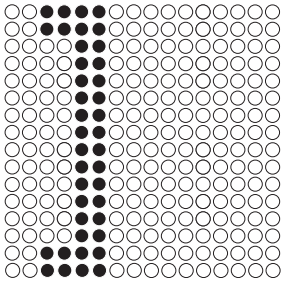

CHR088

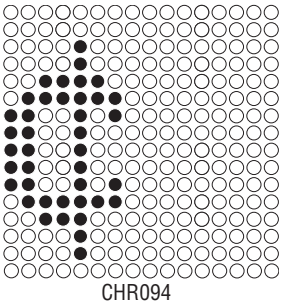

CHR089

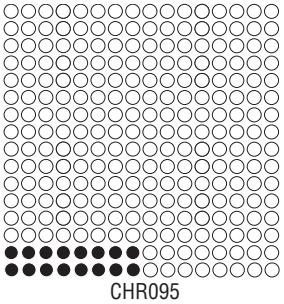

CHR090

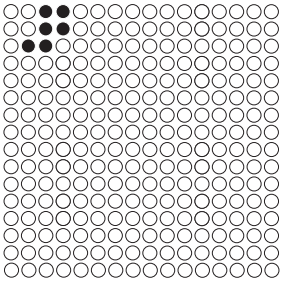

CHR091

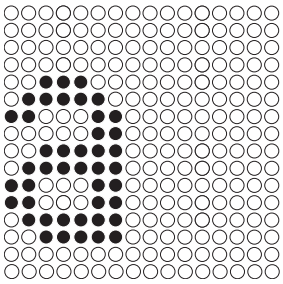

CHR092

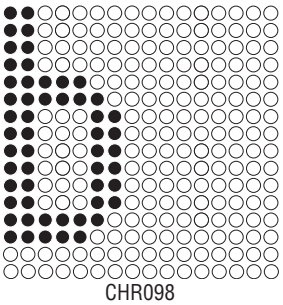

CHR093

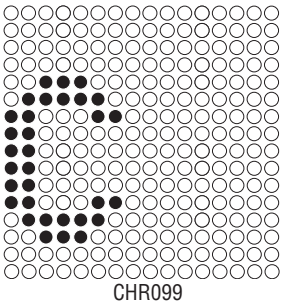

CHR094

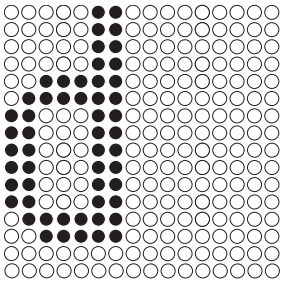

CHR095

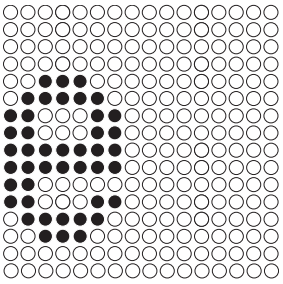

CHR096

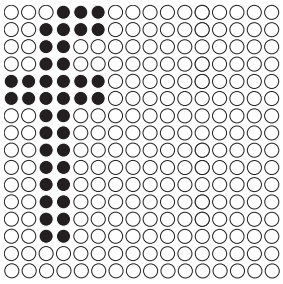

CHR097

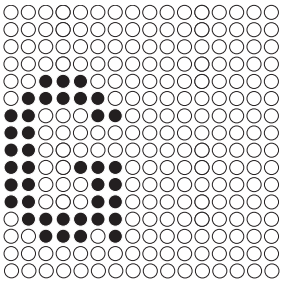

CHR098

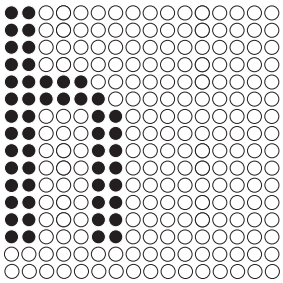

CHR099

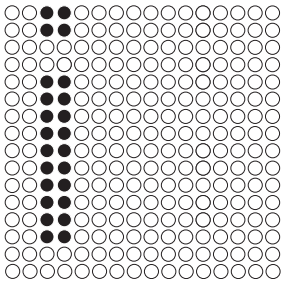

CHR100

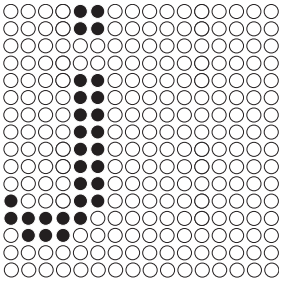

CHR101

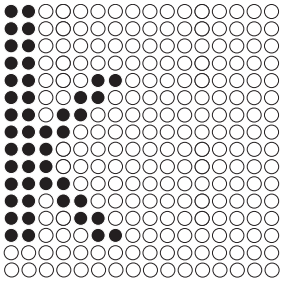

CHR102

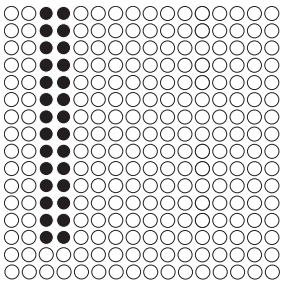

CHR103

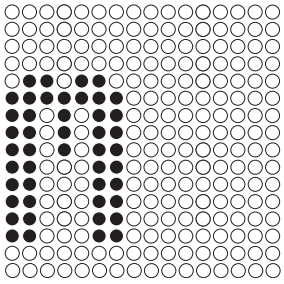

CHR104

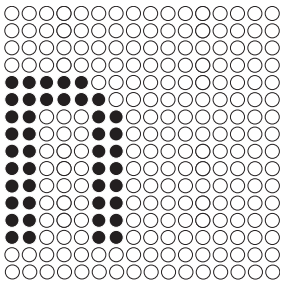

CHR105

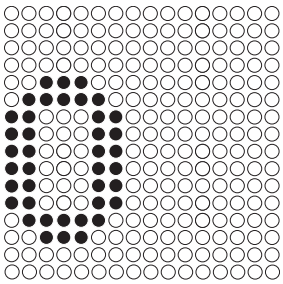

CHR106

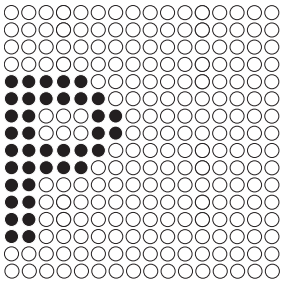

CHR107

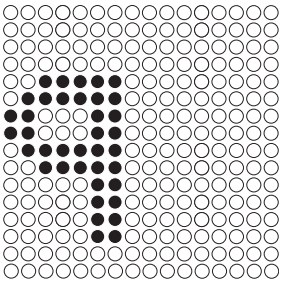

CHR108

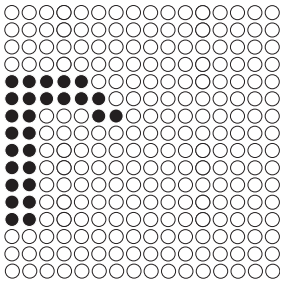

CHR109

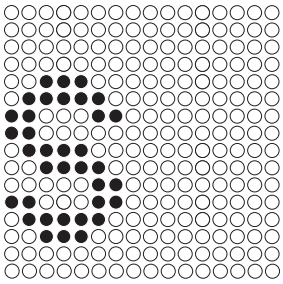

CHR110

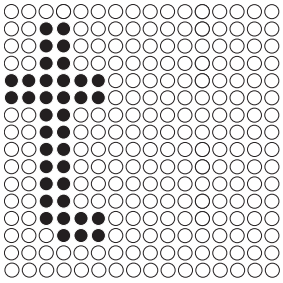

CHR111

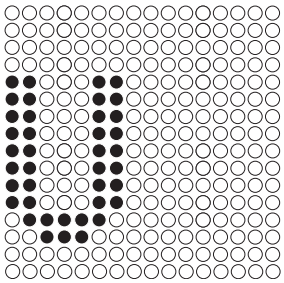

CHR112

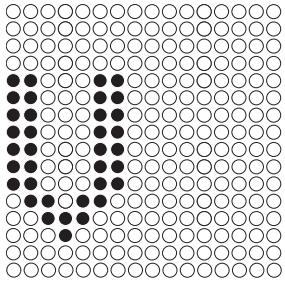

CHR113

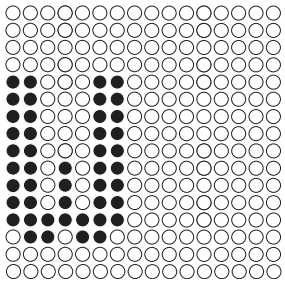

CHR114

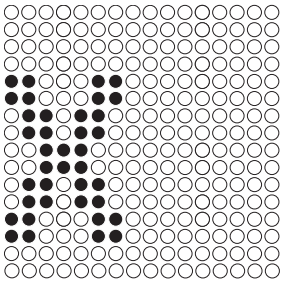

CHR115

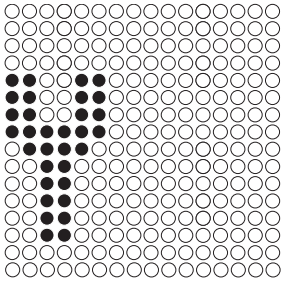

CHR116

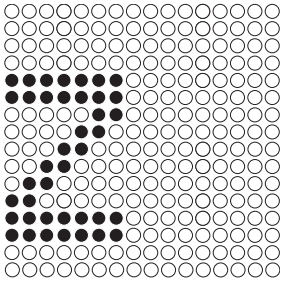

CHR117

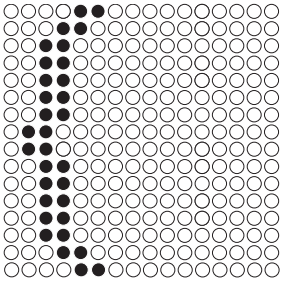

CHR118

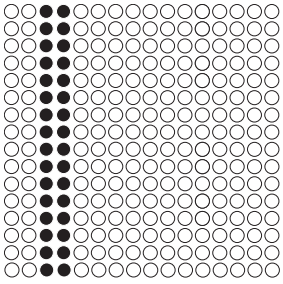

CHR119

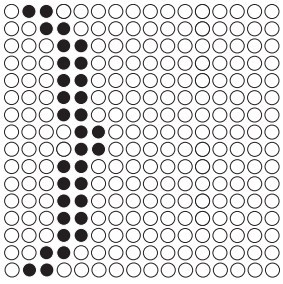

CHR120

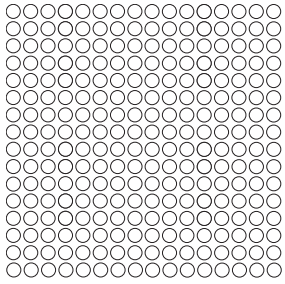

CHR121

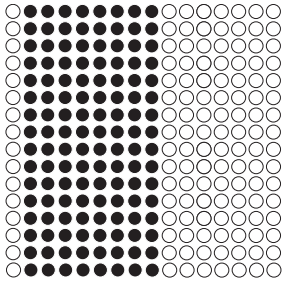

CHR122

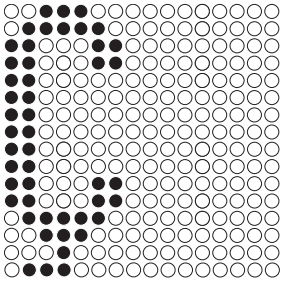

CHR123

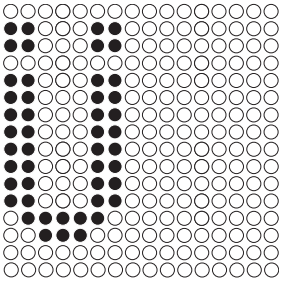

CHR124

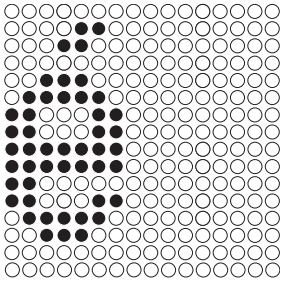

CHR125

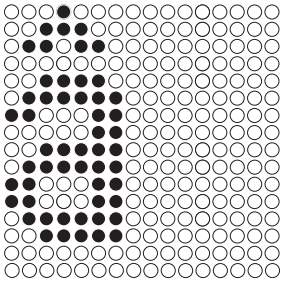

CHR126

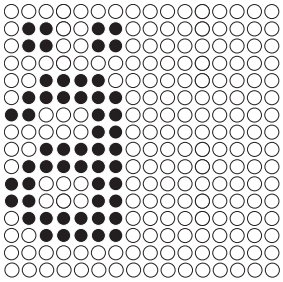

CHR127

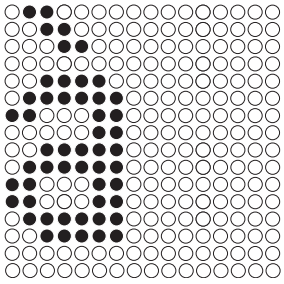

CHR128

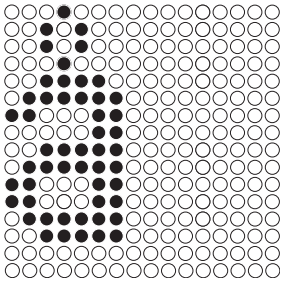

CHR129

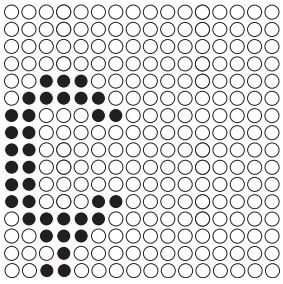

CHR130

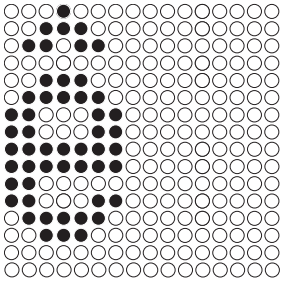

CHR131

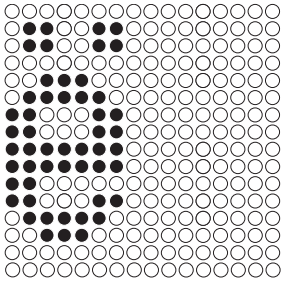

CHR132

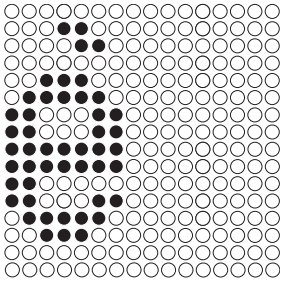

CHR133

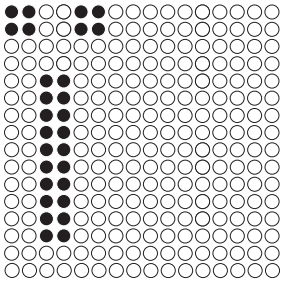

CHR134

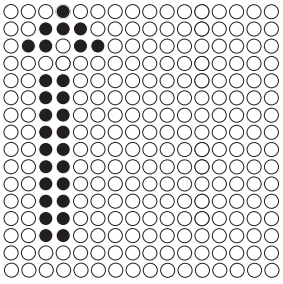

CHR135

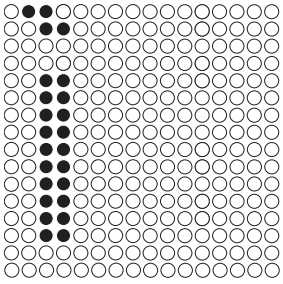

CHR136

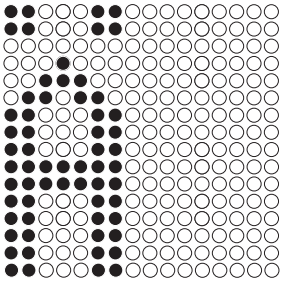

CHR137

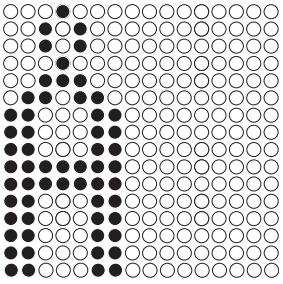

CHR138

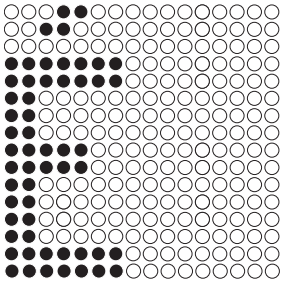

CHR139

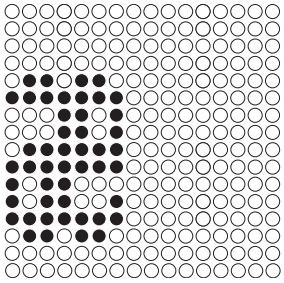

CHR140

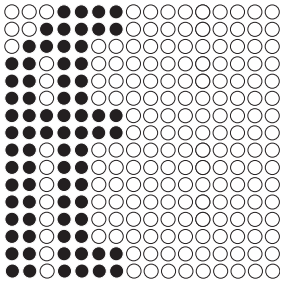

CHR141

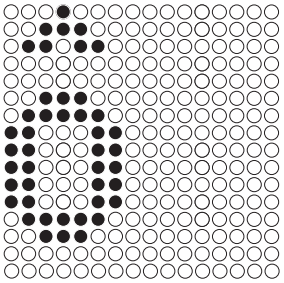

CHR142

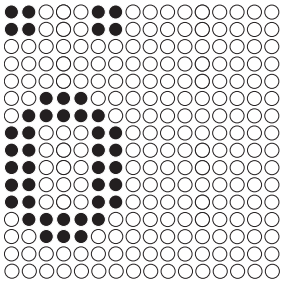

CHR143

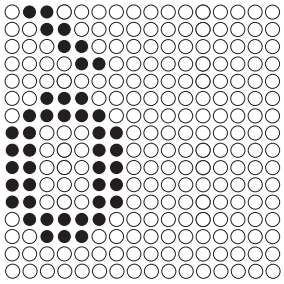

CHR144

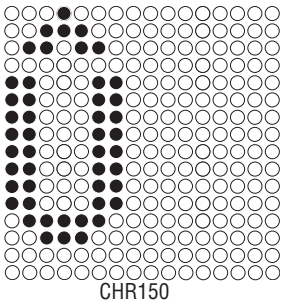

CHR145

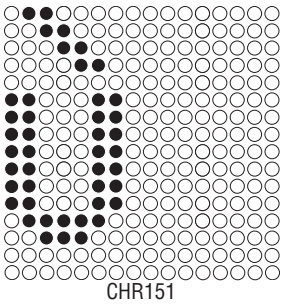

CHR146

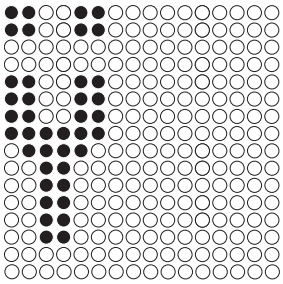

CHR147

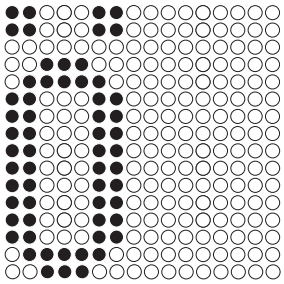

CHR148

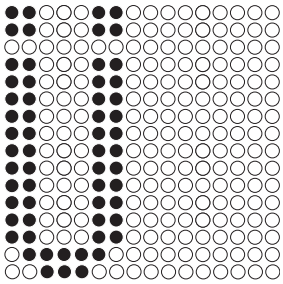

CHR149

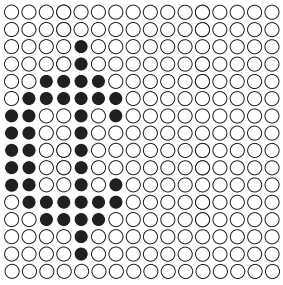

CHR150

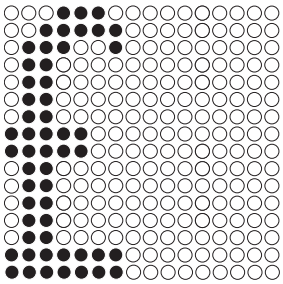

CHR151

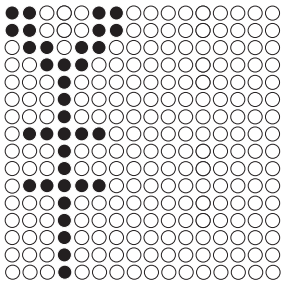

CHR152

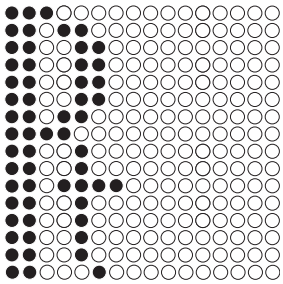

CHR153

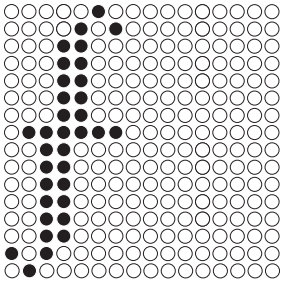

CHR154

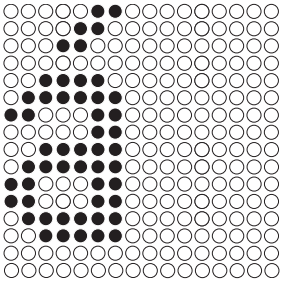

CHR155

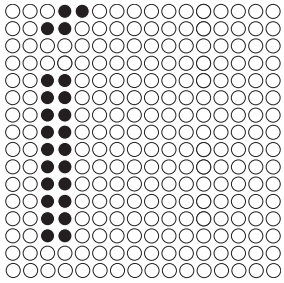

CHR156

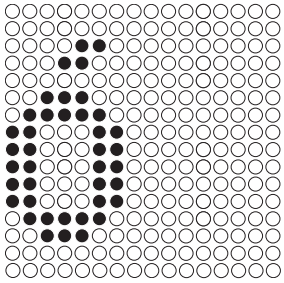

CHR157

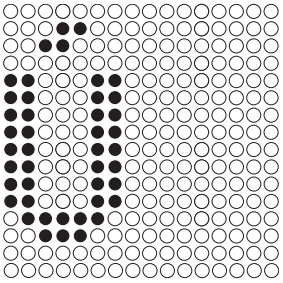

CHR158

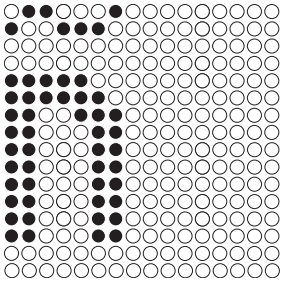

CHR159

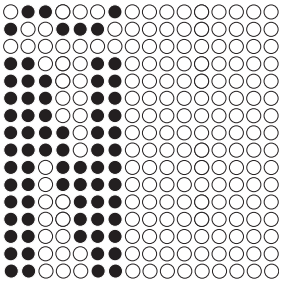

CHR160

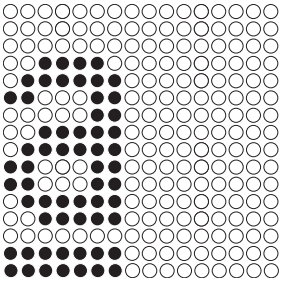

CHR161

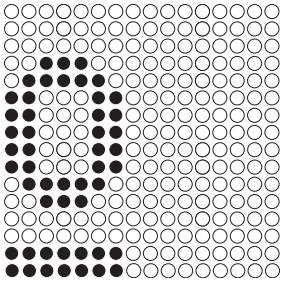

CHR162

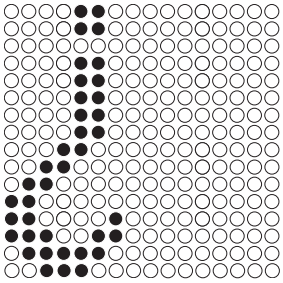

CHR163

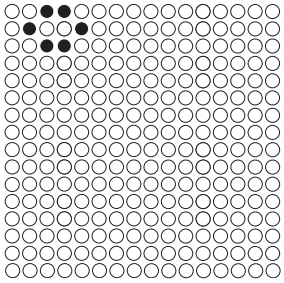

CHR164

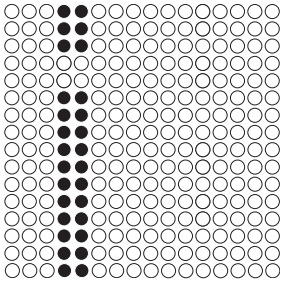

CHR165

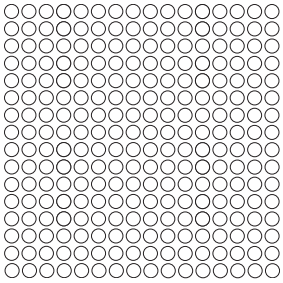

CHR166

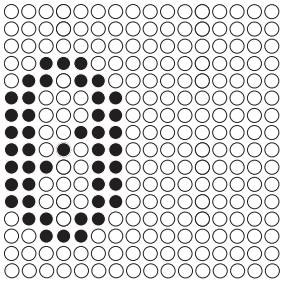

CHR167

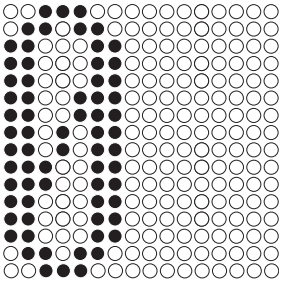

CHR168

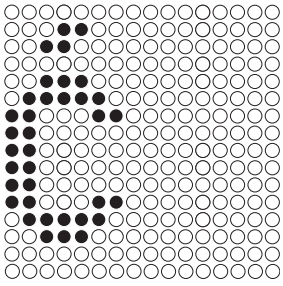

CHR169

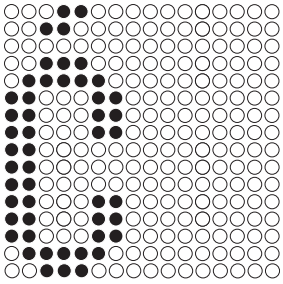

CHR170

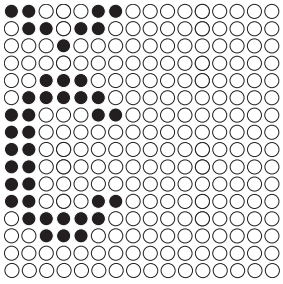

CHR171

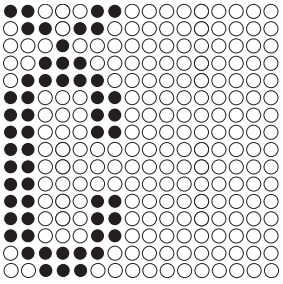

CHR172

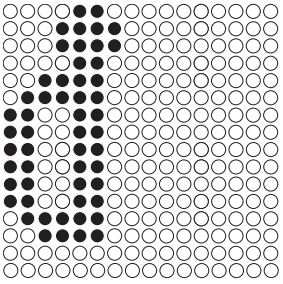

CHR173

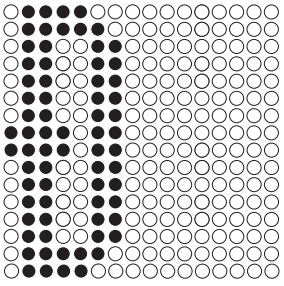

CHR174

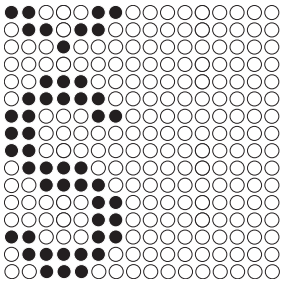

CHR175

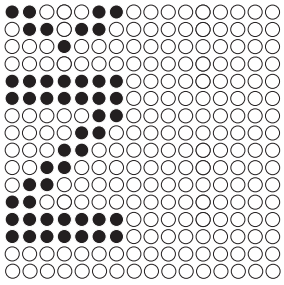

CHR176

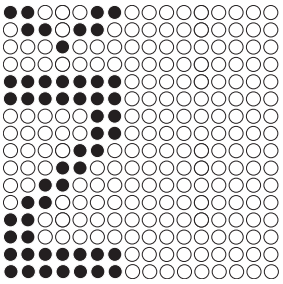

CHR177

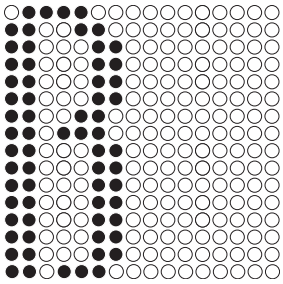

CHR178

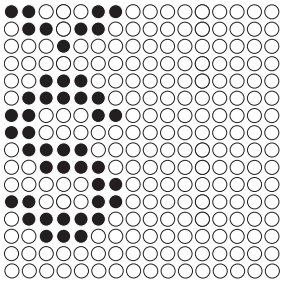

CHR179

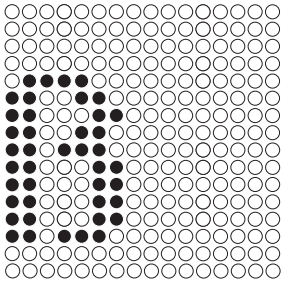

CHR180

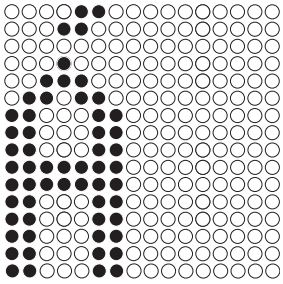

CHR181

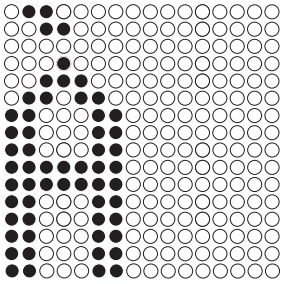

CHR182

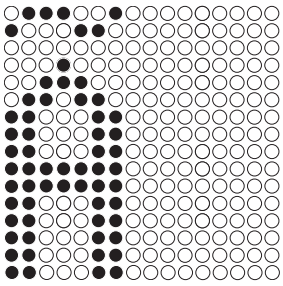

CHR183

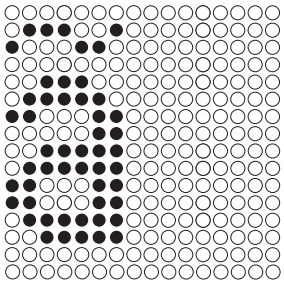

CHR184

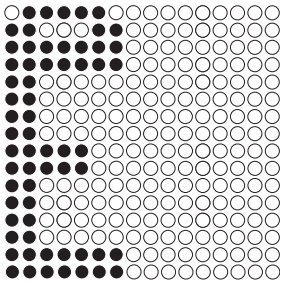

CHR185

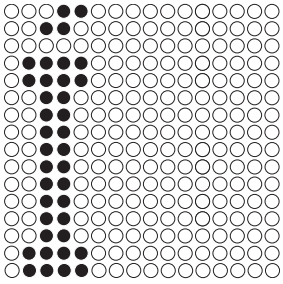

CHR186

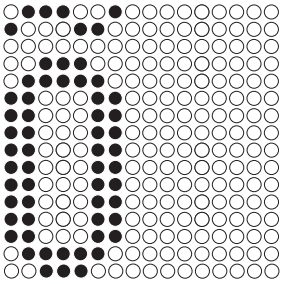

CHR187

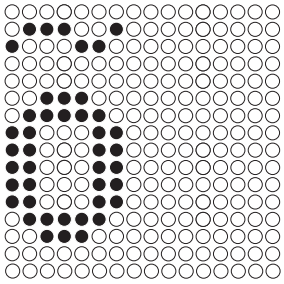

CHR188


CHR189

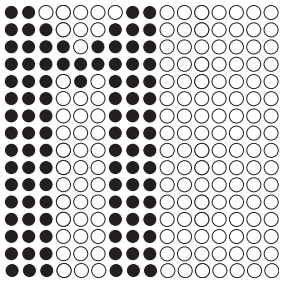

CHR190

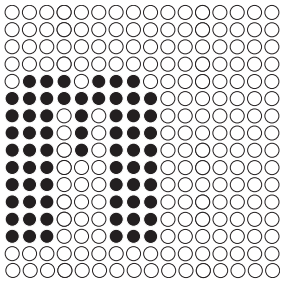

CHR191

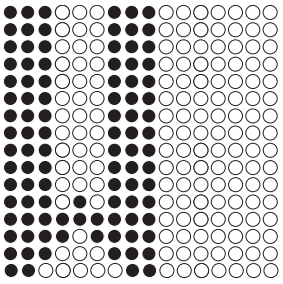

CHR192

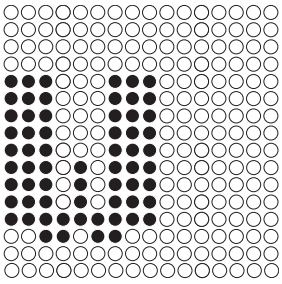

CHR193

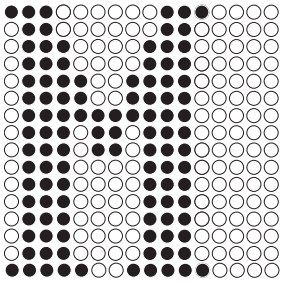
7.13.18 16-High Fat Character

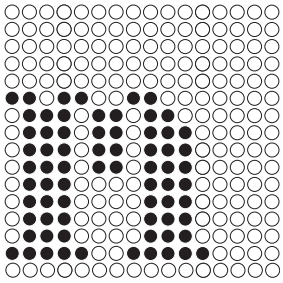

FATM

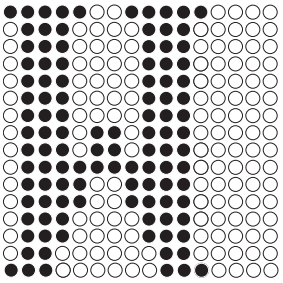

FATM1

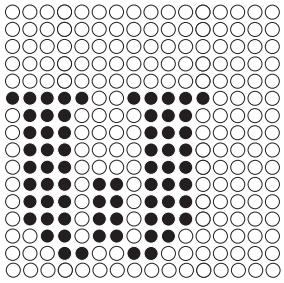

FATW


FATW1


FFATM


FFATM1


FFATW


FFATW1