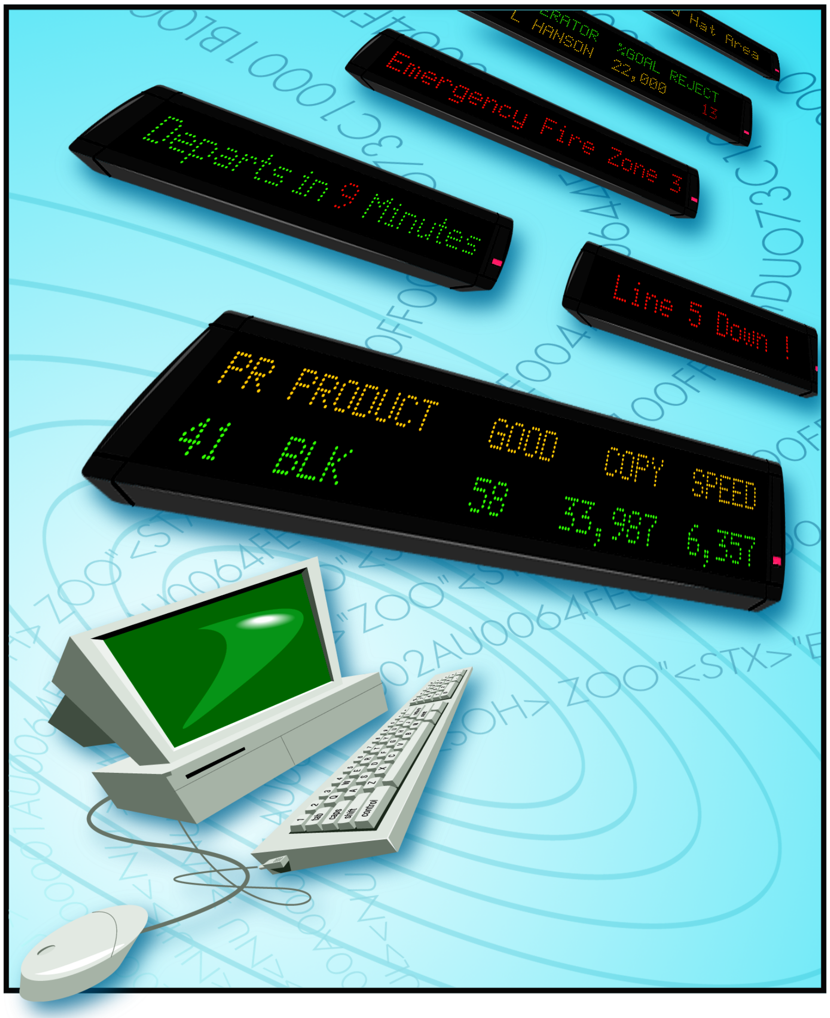
Alpha® Sign Communications Protocol Revision D

Contents

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



ADAPTIVE®

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

August 15, 2002 9708-8061D

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

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. The standard Alpha® EPROM contains three versions of protocol with which you can communicate with a Alpha® sign:

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

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

3.2 Document conventions

Table 2: Document conventions

Convention	Description
<SOH> or ^A	ASCII control character abbreviation (see page 86)
“A”	ASCII character (in this case the letter A)
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 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 files

LARGE DOTS PICTURE files are supported only on the AlphaVision, AlphaEclipse, AlphaPremiere, and Alpha series 7000 products. It is similar to the standard SMALL DOTS PICTURE file as described above. The LARGE DOTS PICTURE file can be much larger than the standard DOTS picture. The LARGE DOTS PICTURE file supports data compression during serial transmission and has a maximum size of 65535 x 65535 pixels.

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 data specifications

Table 3: Protocol data specifications

	EZ KEY II	Alpha 1.0 (EZ95)		Alpha 2.0	
Baud rate:	1200, 2400, 4800	1200, 2400, 4800, 9600		1200, 2400, 4800, 9600, 19200, 38400	
Start bits:	1	1		1	
Data bits:	7	8	7	8	7
Parity:	Even	None	Even	None	Even
Stop bits:	2	1	2	1	2
Flow control:	None	None		None	
Time-out period:	1 second (delays between bytes cannot exceed this)				

5.0 Transmission frame formats

Each of the three protocols (EZ KEY II, Alpha 1.0, and Alpha 2.0) can be transmitted to a sign in any one of two, basic transmission frame formats:

<NUL>	<NUL>	<NUL>	<NUL>	<NUL>	<SOH> ^A	Type Code	Sign Address	<STX> ^B	Command Code	Data Field	<EOT> ^C

Figure 1: Standard transmission frame

SPECIAL NOTE

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

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

1. Standard — also called the “1-byte” or “^A” format. This format has several variations:
   - Checksum
   - Nesting with Checksums
   - Nesting without Checksums
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”)

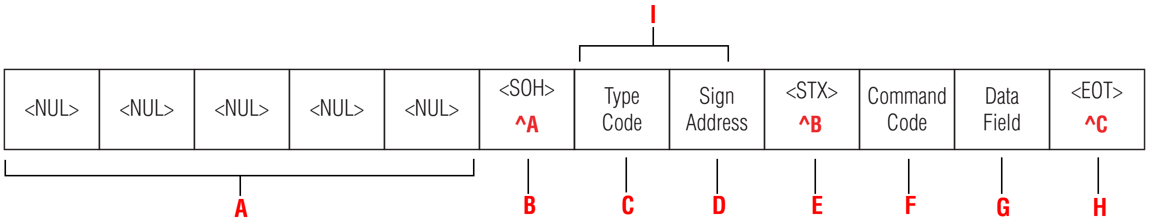
5.1 Standard transmission frame (“1-byte” or “^A”) format

SHOW ME

An example of the Standard transmission frame is on page 52.

This is called the “1-byte” or “^A” format because the <SOH> frame start character is a single, non-ASCII printable byte:

Table 4: Standard transmission frame (“1-byte” or “^A”) format



Item	Name	Description
A	<NUL>	A minimum of five <NUL>s (00H) must be transmitted as frame 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): “!” 21H All signs with Visual Verification. This code causes a sign to display the *Transmission OK* message when a transmission frame 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 “U” 55H 790i sign “V” 76H Eclipse series “W” 57H AlphaEclipse Time/Temp “X” 58H Premiere 4000 and 9000 series “Z” 5AH All signs “^” 5EH Betabrite sign “a” 61H 4120C sign “b” 62H 4160C 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 “t” 74H 4080C sign “u” 75H 210C and 220C signs “v” 76H AlphaEclipse signs (except Time/Temp sign) “x” 78H Premiere 9000 sign “z” 7AH All signs first configure memory for 26 files (“A” - “Z”) of 150 characters each and then execute the specified command.
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 frames, there must be at least a 100 millisecond delay after the <STX>.

Table 4: Standard transmission frame (“1-byte” or “^A”) format

F	Command Code	One ASCII character that defines the transmission and data types: “A” 41H = Write TEXT file (see page 17) “B” 42H = Read TEXT file (see page 18) “E” 45H = Write SPECIAL FUNCTION commands (see page 20) “F” 46H = Read SPECIAL FUNCTION commands (see page 27) “G” 47H = Write STRING file (see page 34) “H” 48H = Read STRING file (see page 35) “I” 49H = Write SMALL DOTS PICTURE file (see page 36) “J” 4AH = Read SMALL DOTS PICTURE file (see page 38) “M” 4DH = Write LARGE DOTS PICTURE file (page 39) “N” 4EH = Read LARGE DOTS PICTURE file (page 40) “O” 4FH = Write ALPHAVISION BULLETIN message (page 41) “T” 54H = Set Timeout Message (see 107) (Alpha 2.0 protocol 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

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

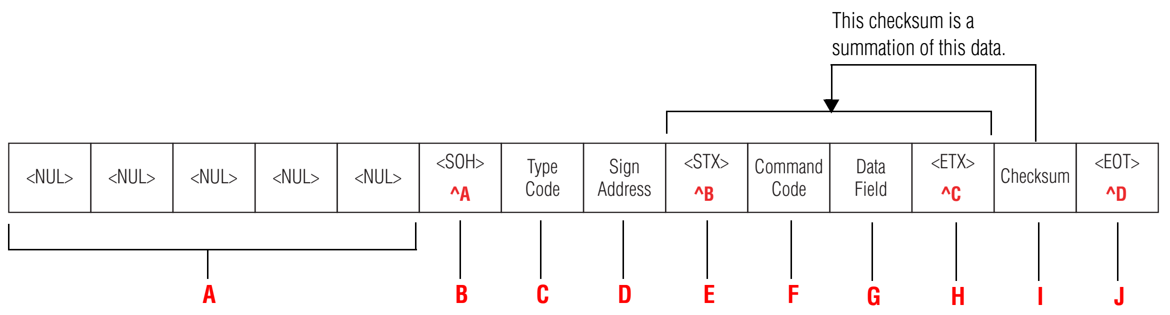
If an <ETX> character is transmitted before the <EOT>, the sign will expect a Checksum.

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

SHOW ME

An example of the Transmission frame with Checksum is on page 54.

Table 5: Standard transmission frame with Checksum format



<NUL>	<NUL>	<NUL>	<NUL>	<NUL>	<SOH> ^A	Type Code	Sign Address	<STX> ^B	Command Code	Data Field	<ETX> ^C	Checksum	<EOT> ^D
A					B	C	D	E	F	G	H	I	J

Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission frame ("1-byte" or "^A") format," on page 9.
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 frame had a valid Checksum, use the Read SPECIAL FUNCTION to check the Serial Error Status Register (see page 27).
J	<EOT>	See Table 4, "Standard transmission frame ("1-byte" or "^A") format," on page 9.

5.1.2 Nesting with Checksums format

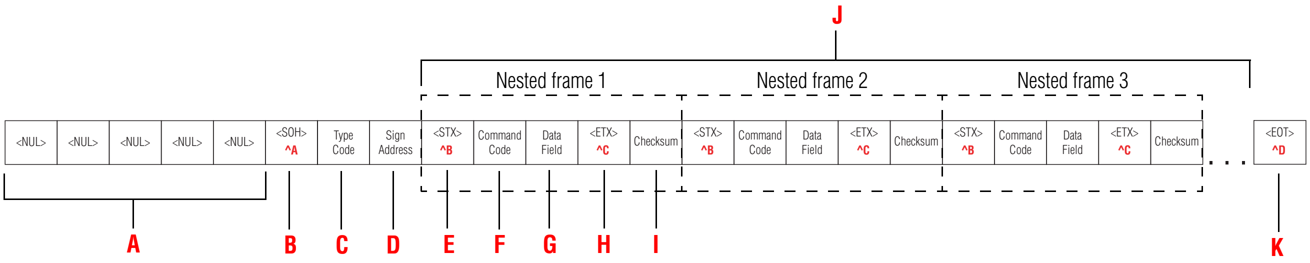
SHOW ME

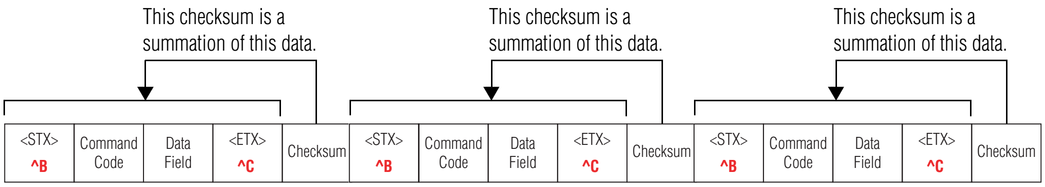
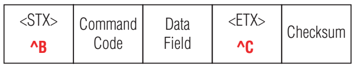
An example of the Nesting with Checksums is on page 55.

If more than one transmission frame is required consecutively, multiple Commands can be repeated or “nested” within a transmission frame.

A sign uses this format when a Memory Dump [see “Read SPECIAL FUNCTIONs Command Code — “F” (46H)” on page 27] is requested serially.

Table 6: Nesting with Checksums format



Item	Name	Description
A	<NUL>	See Table 4, “Standard transmission frame (“1-byte” or “^A”) format,” on page 9.
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 frames, this is how the checksums are calculated:  This checksum is a summation of this data. This checksum is a summation of this data. This checksum is a summation of this data. <STX> ^B Command Code Data Field <ETX> ^C Checksum <STX> ^B Command Code Data Field <ETX> ^C Checksum <STX> ^B Command Code Data Field <ETX> ^C Checksum NOTE: When a sign receives an invalid Checksum, the associated data will not be processed. NOTE: When nesting frames, 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 frame. This is the format of the nested frame *with a Checksum*:  <STX> ^B Command Code Data Field <ETX> ^C Checksum
K	<EOT>	See Table 4, “Standard transmission frame (“1-byte” or “^A”) format,” on page 9.

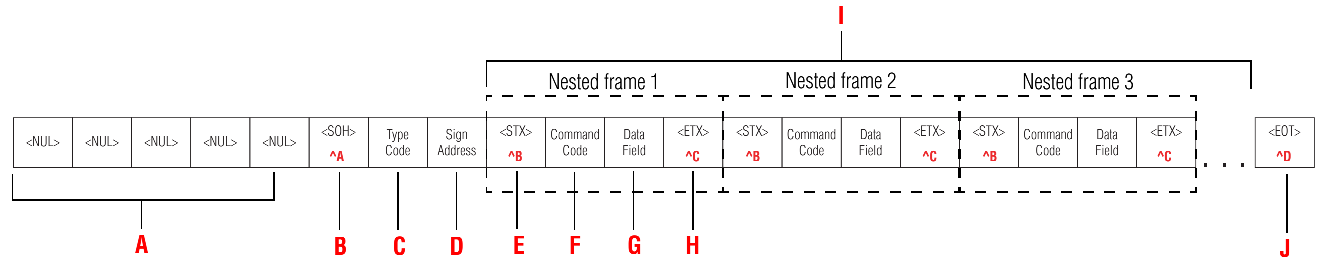
5.1.3 Nesting without Checksums format

SHOW ME

An example of the Nesting without Checksums is on page 56.

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

Table 7: Nesting without Checksums transmission frame



Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission frame ("1-byte" or "^A") format," on page 9.
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 frame. This is the format of the nested frame *without a Checksum*: <STX> ^B | Command Code | Data Field | <ETX> ^C NOTE: When nesting frames, 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 frame ("1-byte" or "^A") format," on page 9.

5.2 ASCII Printable formats

SPECIAL NOTE

For ASCII Printable format baud rate, parity, etc., see Table 3, “Protocol data specifications,” on page 7.

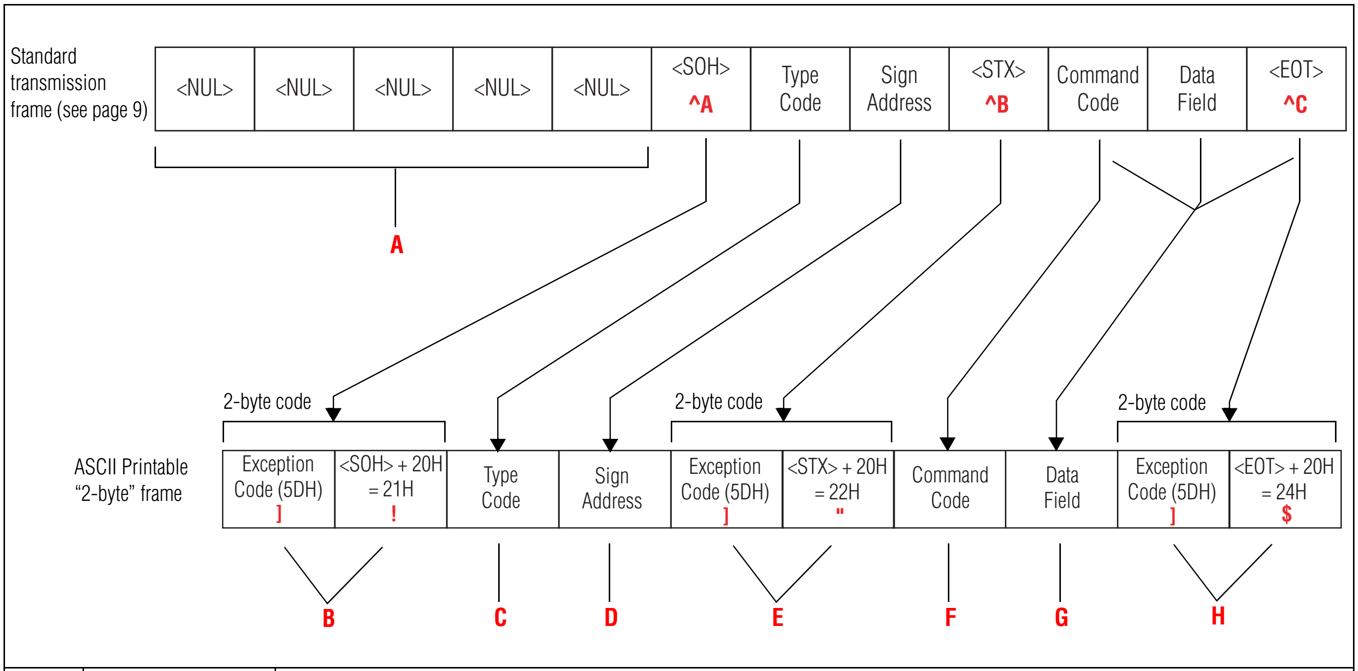
Many pagers and computer systems cannot receive or send ASCII control codes (characters lower than 20H). The ASCII Printable format is a variation of the transmission frame that allows the entire protocol to be transmitted *without* sending any ASCII control codes — thus allowing its use with pagers.

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 frame.

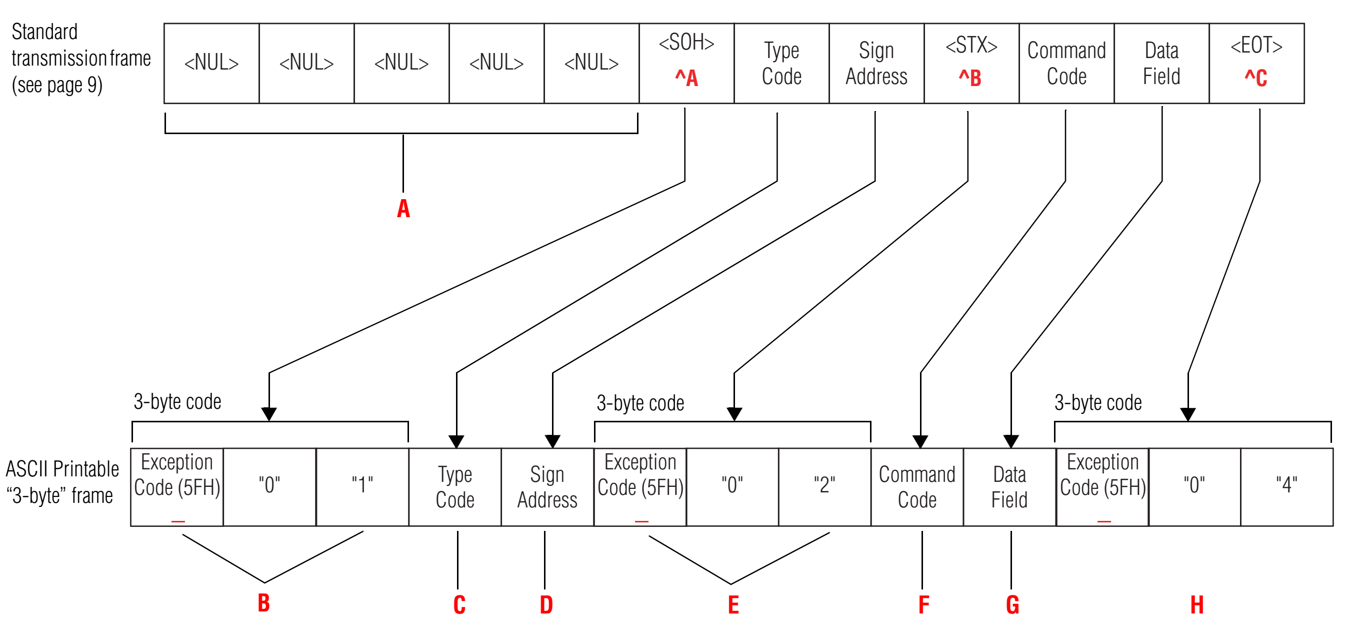
Table 8: Standard transmission frame compared with ASCII Printable “2-byte” code transmission frame



Item	Name	Description
A	<NUL>	The five <NUL>s (00H) used as frame 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 frame (“1-byte” or “^A”) format,” on page 9.
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 frame (“1-byte” or “^A”) format,” on page 9.
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

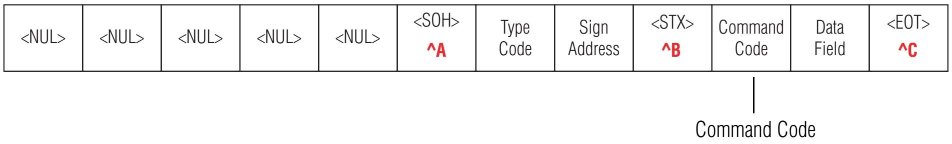
Table 9: Standard transmission frame compared with ASCII Printable “3-byte” code transmission frame



Item	Name		Description
A	<NUL>		The five <NUL>s (00H) used as frame 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 frame (“1-byte” or “^A”) format,” on page 9.
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 frame (“1-byte” or “^A”) format,” on page 9.
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 is a single-byte field in a protocol transmission frame. It is used to determine whether information is read from or written to signs.



<NUL>	<NUL>	<NUL>	<NUL>	<NUL>	<SOH> ^A	Type Code	Sign Address	<STX> ^B	Command Code	Data Field	<EOT> ^C

Figure 2: Command Code location in the Standard transmission frame

Command Codes not only determine the contents of the Command Code field, but also the Data Field in the protocol transmission frame formats (see "Transmission frame formats" on page 8).

Command Codes fall into six, general categories:

- TEXT file commands
- SPECIAL FUNCTION commands (page 20)
- STRING file commands (page 34)
- SMALL DOTS PICTURE file commands (page 36)
- LARGE DOTS PICTURE file commands (page 39)
- ALPHAVISION BULLETIN MESSAGE file commands (page 41)

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 frame. 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.

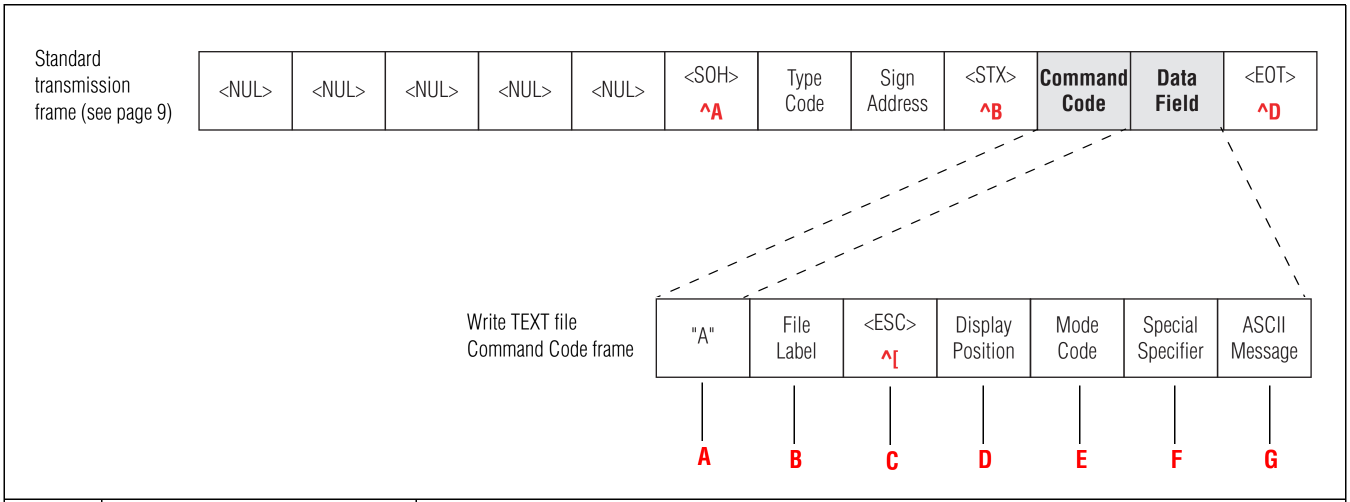
6.1.1 Write TEXT file Command Code — “A” (41H)

SHOW ME

An example of the Write TEXT Command Code is on page 57.

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 10: Write TEXT file transmission frame format



Standard transmission frame (see page 9)

<NUL>	<NUL>	<NUL>	<NUL>	<NUL>	<SOH> ^A	Type Code	Sign Address	<STX> ^B	**Command Code**	**Data Field**	<EOT> ^D

Write TEXT file Command Code frame

"A"	File Label	<ESC> ^[	Display Position	Mode Code	Special Specifier	ASCII Message
A	B	C	D	E	F	G

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 43. If the File Label = “0” (30H), then a Priority TEXT file will be written (see “Priority TEXT files” on page 19).
C		[1]TEXT file data format	Mode Field (optional)	<ESC>	<ESC> (1BH) always starts the Mode Field.
D				Display Position	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. NOTE: On one-line signs, the Display Position is irrelevant, but it still must be included.
E				Mode Code	A single ASCII character that represents a “mode” which is a way of displaying an ASCII message. See “Standard Modes” on page 87.
F				Special Specifier	(Only required when preceding Mode Code is “n” for SPECIAL.) See “Special Modes” on page 87 and “Special Graphics” on page 88.
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 75). 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)

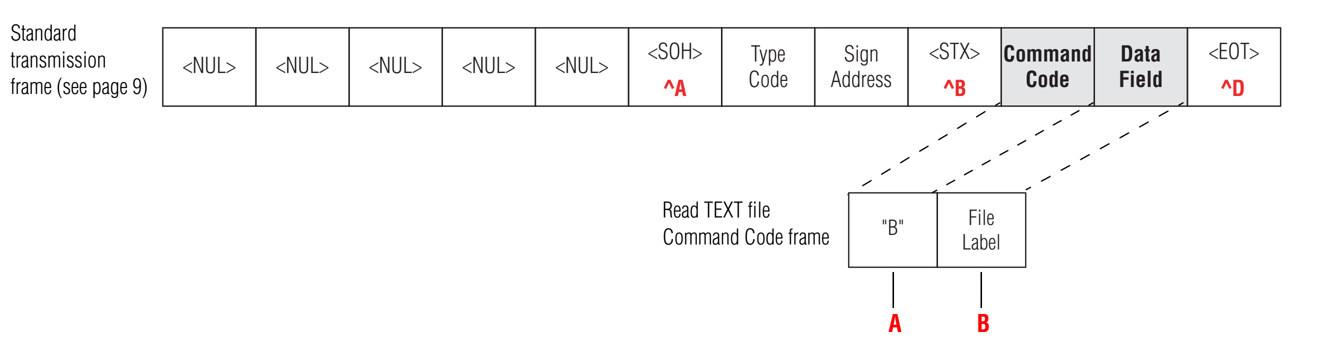
SHOW ME

An example of the Read TEXT file frame is on page 58.

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 11: Read TEXT file transmission frame format



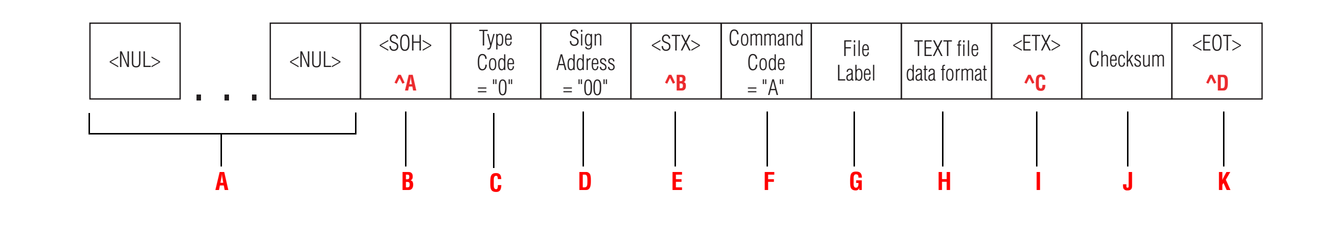
Item	Name		Description
A	**Command Code**		“B” (42H) = Read TEXT file
B	**Data Field**	File Label	One ASCII character that indicates the TEXT file being accessed. See “Appendix A: Valid File Labels” on page 43. If the File Label = “0” (30H), then the Priority TEXT file will be read (see “Priority TEXT files” on page 19).

SHOW ME

An example of the Read TEXT file sign response frame is on page 58.

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

Table 12: Read TEXT file sign response frame 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” 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 43.
H	TEXT file data format	See Table 10, “Write TEXT file transmission frame format,” on page 17.
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 frames are on page 62.

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

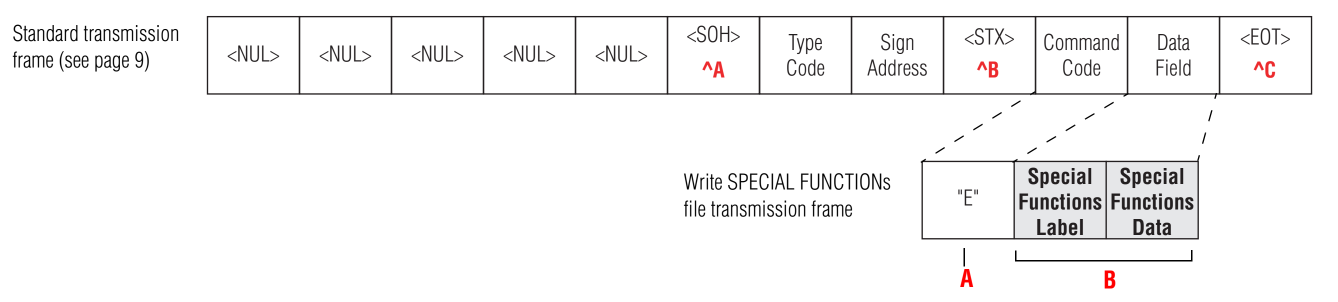
There are a number of special function commands which give the user additional information and control of the sign.

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

SHOW ME

An example of the Write SPECIAL FUNCTIONs frame is on page 63.
Examples of Set Memory Configuration start on page 65.

Table 13: Write SPECIAL FUNCTION Command Code format



Item	Name	Description
A	Command Code	“E” (45H) = Write SPECIAL FUNCTION command
B	**Special Functions Label** (one ASCII character)	**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 the “Control characters” in “Appendix G: Alpha® protocol ASCII table” on page 75.

	“!” 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: $\underbrace{\text{FTPSIZEQQQQ}}_{\text{Repeat for each file to be configured.}}$ where:
		F = One ASCII character that represents the File Label. For valid File Labels, see “Appendix A: Valid File Labels” on page 43.
		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 44.
		• 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

Table 13: Write SPECIAL FUNCTION Command Code format

B (cont)	“&” 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 = See “Store a programmable sound (33H)” on page 98. (Alpha 2.0 protocol only) “4” 34H = See “Trigger a programmable sound (34H)” on page 99 (Alpha 2.0 protocol only)
	“)” 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 44. 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 71 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 13: Write SPECIAL FUNCTION Command Code format

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 43). 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 Alpha Solar 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 Big Dot signs.
	"2" 32H	**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 43. 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.

Table 13: Write SPECIAL FUNCTION Command Code format

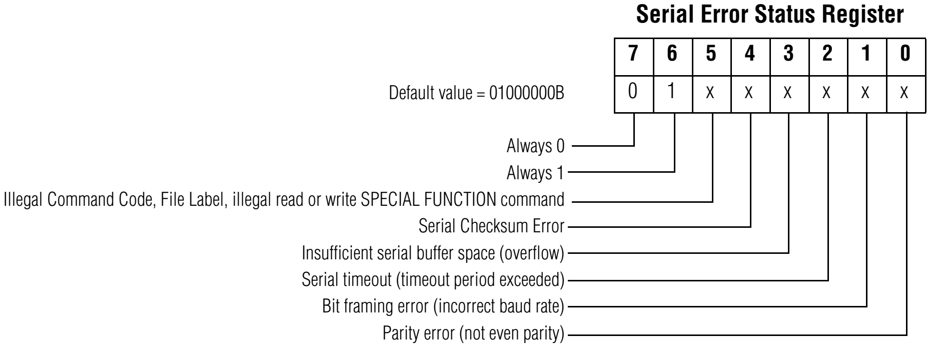
	“4” 34H	**Clear Serial Error Status Register** — one ASCII character that is used to clear the Serial Error Status Register to its default value of 40H. This register is set to its default value (40H or 01000000B) for the following Command Codes: (1) Read Serial Error Status Register, (2) Network Query, or (3) Clear Serial Error Status Register.  Serial Error Status Register 7 6 5 4 3 2 1 0 Default value = 01000000B 0 1 x x x x x x Always 0 Always 1 Illegal Command Code, File Label, illegal read or write SPECIAL FUNCTION command Serial Checksum Error Insufficient serial buffer space (overflow) Serial timeout (timeout period exceeded) Bit framing error (incorrect baud rate) Parity error (not even parity) **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 14, “Read SPECIAL FUNCTIONs file transmission frame format,” on page 27). **NOTE:** Parity error (not even parity) is not used on most signs.

Table 13: Write SPECIAL FUNCTION Command Code format

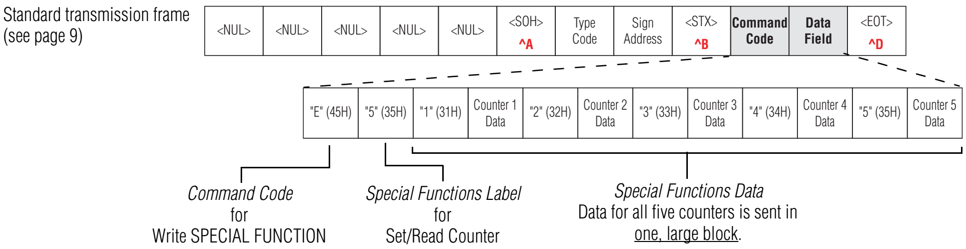
B (cont)	“5”	35H	**Set Counter** — used to set one or more of the five internal timers available on *counter-equipped* signs. Data for all five counters must be sent as *one, large block*, in the following format: NOTE: Even if you are only setting one counter, data must be sent to the other counters as well.  Standard transmission frame (see page 9) <NUL> <NUL> <NUL> <NUL> <NUL> <SOH> ^A Type Code Sign Address <STX> ^B Command Code Data Field <EOT> ^D "E" (45H) "5" (35H) "1" (31H) Counter 1 Data "2" (32H) Counter 2 Data "3" (33H) Counter 3 Data "4" (34H) Counter 4 Data "5" (35H) Counter 5 Data Command Code for Write SPECIAL FUNCTION Special Functions Label for Set/Read Counter Special Functions Data Data for all five counters is sent in one, large block. 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 44. **(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 44. 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”)**

Table 13: Write SPECIAL FUNCTION Command Code format

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 FAR DOTS PICTURE. Valid values are: “01” = a monochrome DOTS PICTURE “02” = a tricolor DOTS PICTURE 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. Applies to the AlphaVision, AlphaEclipse, AlphaPremiere, and Alpha 7000 series signs.
	“:”	3AH	**Set Run File Times** — see “Set Run File Time (3AH)” on page 99. (Alpha 2.0 protocol only, currently for the AlphaEclipse and AlphaPremiere signs).
	“;”	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** — see “Custom character sets” on page 103. (Alpha 2.0 protocol only)
	“=”	3DH	**Enable/Disable Daylight Savings Time** — see “Enable/Disable Daylight Saving Time (3DH)” on page 106. (Alpha 2.0 protocol only)
	“>”	3EH	**Set AutoMode Table** — see “Set AutoMode Table (3EF)” on page 106. (Alpha 2.0 protccol only)
	“@”	3FH	**Set Dimming Control Register** — see “Set Dimming Control Register (“@”)” on page 108. (Alpha 2.0 protocol only)
	“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.
	“s”	73H	**Enable/Disable ACK/NAK Response** — see “Enable/Disable ACK/NAK Response (“s”)” on page 109. (Alpha 2.0 protocol only)

Table 13: Write SPECIAL FUNCTION Command Code format

NOTE: [1]The sum of all the file sizes (except for DOTS PICTURE and ALPHAVISION 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]Set 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 firmware version 2.0 and greater, the year (yy) is windowed as follows: "00 to "96" = 2000 to 2096. "97" to "99" = 1997 to 1999.

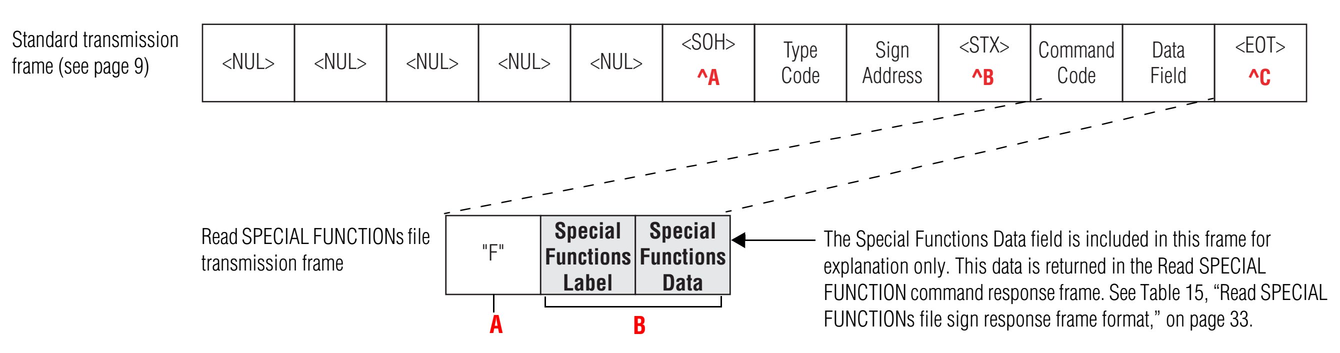
6.2.2 Read SPECIAL FUNCTIONs Command Code — “F” (46H)

SHOW ME

An example of the Read SPECIAL FUNCTION command is on page 63.

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 14: Read SPECIAL FUNCTIONs file transmission frame format



The Special Functions Data field is included in this frame for explanation only. This data is returned in the Read SPECIAL FUNCTION command response frame. See Table 15, “Read SPECIAL FUNCTIONs file sign response frame format,” on page 33.

Item	Name	Description
A	Command Code	“F” (46H) = Read SPECIAL FUNCTIONS file
B	**Special Functions Label** (one ASCII character)	**Special Functions Data** (This data is returned in a Read SPECIAL FUNCTIONS file sign response. See Table 15, “Read SPECIAL FUNCTIONs file sign response frame format,” on page 33)

	“ “ 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 75.

	“!” 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 or 29 ASCII characters in the following format: <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” 45H + 45H = 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 14: Read SPECIAL FUNCTIONs file transmission frame format

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 43. 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 44. • 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 entires are “1000” = monochrome DOTS PICTURE “2000” = 3-color DOTS PICTURE “4000” = 8-color DOTS PICTURE
	“%” 25H	**Memory Dump** — returns multiple nested transmission frames with checksums (see “Nesting with Checksums format” on page 12) 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 43) 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 44. 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 14: Read SPECIAL FUNCTIONs file transmission frame format

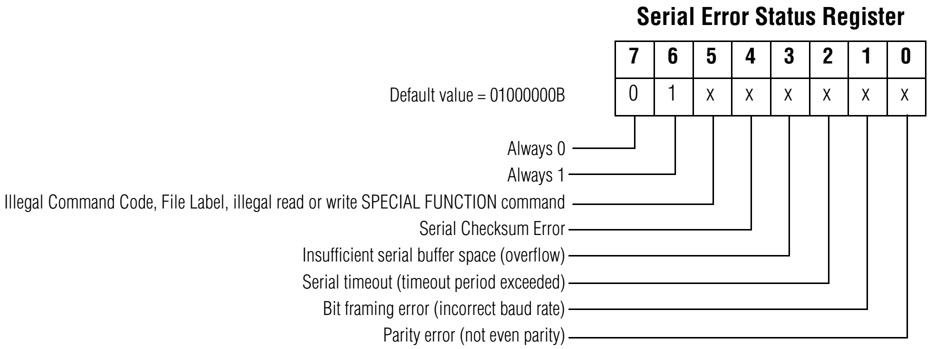
B (cont)	“*”	2AH	**Read Serial Error Status Register** — returns one bitmapped ASCII character read from a sign’s Serial Error Status Register that represents serial errors recorded by a sign. This register is set to its default value (40H or 01000000B) for the following Command Codes: (1) Read Serial 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 7 6 5 4 3 2 1 0 Default value = 01000000B 0 1 x x x x x x Always 0 Always 1 Illegal Command Code, File Label, illegal read or write SPECIAL FUNCTION command Serial Checksum Error Insufficient serial buffer space (overflow) Serial timeout (timeout period exceeded) Bit framing error (incorrect baud rate) Parity error (not even parity) **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.
	“-”	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 frame (“1-byte” or “^A”) format” on page 9. 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 (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 43). 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.

Table 14: Read SPECIAL FUNCTIONs file transmission frame format

B (cont)	“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 43. 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 14: Read SPECIAL FUNCTIONs file transmission frame format

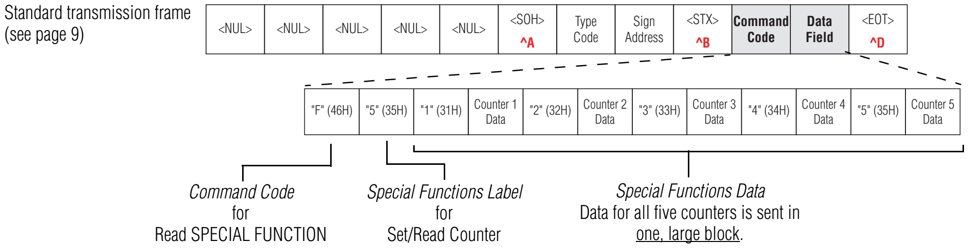
B (cont)	“5”	35H	**Read Counter** — returns data for all five counters is received as *one, large block*, in the following format:  Standard transmission frame (see page 9) <NUL> <NUL> <NUL> <NUL> <NUL> <SOH> ^A Type Code Sign Address <STX> ^B Command Code Data Field <EOT> ^D "F" (46H) "5" (35H) "1" (31H) Counter 1 Data "2" (32H) Counter 2 Data "3" (33H) Counter 3 Data "4" (34H) Counter 4 Data "5" (35H) Counter 5 Data Command Code for Read SPECIAL FUNCTION Special Functions Label for Set/Read Counter Special Functions Data Data for all five counters is sent in one, large block. The format of *Counter 1 Data*, *Counter 2 Data*, etc 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 44. **(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 44. 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”)**

Table 14: Read SPECIAL FUNCTIONs file transmission frame format

B (cont)	"8"	38H	[10]**Read ALPHAVISION 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 Far Dots Picture. Valid values are: "01" = a monochrome DOTS PICTURE "02" = a tricolor DOTS PICTURE rrrr = reserved for future use. Four ASCII zeroes are required — "0000".
	":"	3AH	**Read Run File Times** — see "Reading Run File Time" on page 100. (Alpha 2.0 protocol only)
	";"	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
	"="	3DH	**Read Daylight Savings Time** — see "Enable/Disable Daylight Saving Time (3DH)" on page 106. (Alpha 2.0 protocol only)
	">"	3EH	**Read AutoMode Table** — see "Set AutoMode Table (3EF)" on page 106. (Alpha 2,0 protocol only)
	"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.

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

[2]The sum of all the file sizes (except for DOTS PICTURE and FAR 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 excpetion 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 ALPHAVISION DOTS PICTURE Memory Configuration *only* applies to Full Matrix AlphaVision, AlphaEclipse, AlphaPremiere, and series 7000 signs.

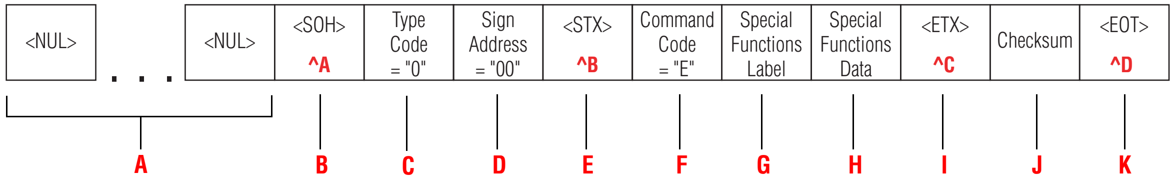
[11]If a file name is less than 9 characters, it must be padded with leading spaces (20H) so that the total number of characters is always nine.

SHOW ME

An example of the Read SPECIAL FUNCTIONs file response frame is on page 64.

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

Table 15: Read SPECIAL FUNCTIONs file sign response frame 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” 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 13, “Write SPECIAL FUNCTION Command Code format,” on page 20 and Table 14, “Read SPECIAL FUNCTIONs file transmission frame format,” on page 27.
H	Special Functions Data	See Table 13, “Write SPECIAL FUNCTION Command Code format,” on page 20. and Table 14, “Read SPECIAL FUNCTIONs file transmission frame format,” on page 27.
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 46.

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 13, "Write SPECIAL FUNCTION Command Code format," on page 20.)

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 75).

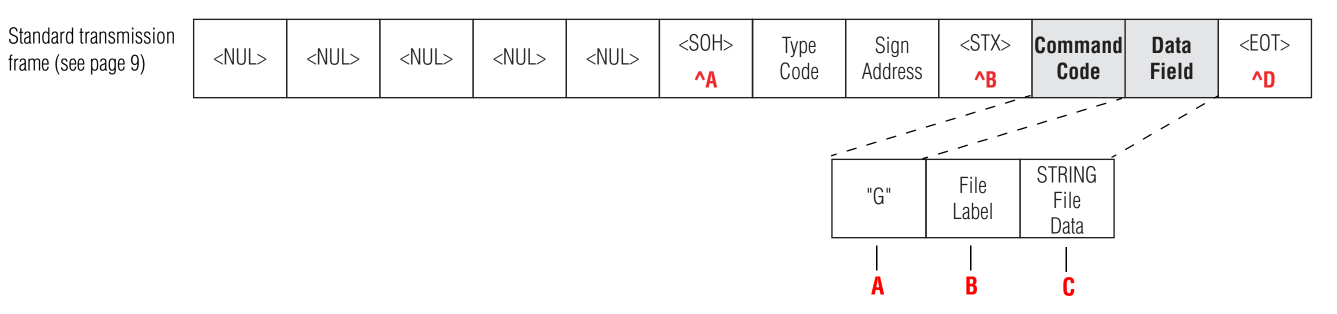
When reading from a STRING file, once the transmission frame has been sent, a sign will either pause or blank, depending on the sign type. Once a sign has transmitted the file, the sign will continue displaying the message from where it was interrupted.

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

SHOW ME

An example of the Write STRING file frame is on page 68.

Table 16: Write STRING file transmission frame 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 43.
C		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 75) : 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

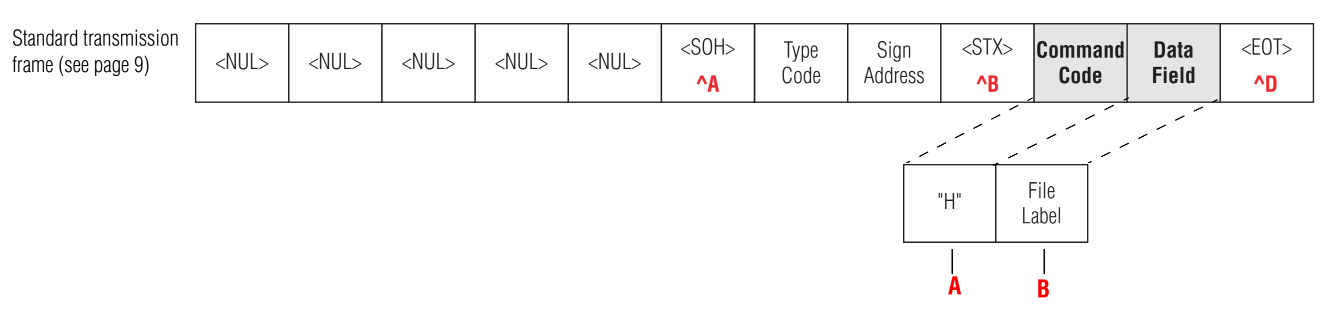
6.3.2 Read STRING file Command Code — “H” (48H)

SHOW ME

An example of the Read STRING file frame is on page 69.

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 17: Read STRING file transmission frame format



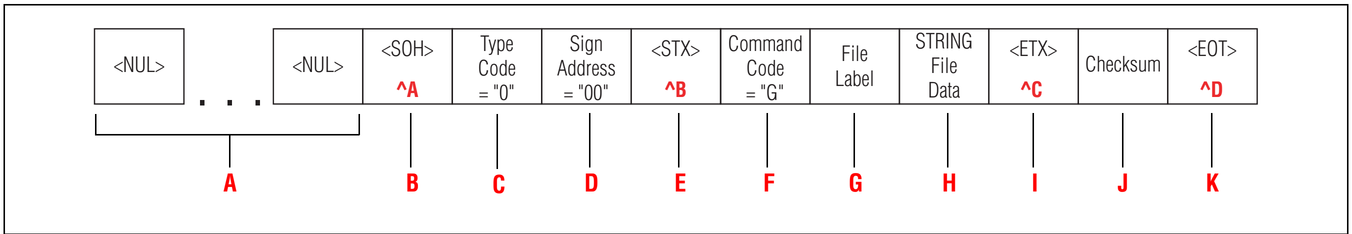
Item	Name		Description
A	Command Code		“H” (48H) = Read STRING file
B	Data Field	File Label	One ASCII character that indicates the STRING file being accessed. See “Appendix A: Valid File Labels” on page 43.

SHOW ME

An example of the Read STRING file sign response frame is on page 69.

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

Table 18: Read STRING file sign response frame 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” 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 43.
H	STRING File Data	See Table 16, “Write STRING file transmission frame format,” on page 34.
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

SMALL DOTS PICTURE files are used to store dot patterns which may be "called" from a TEXT file. The main purpose of SMALL DOTS PICTURE files are to allow users to display custom graphics, such as logos.

When a SMALL DOTS PICTURE exceeds a pixel height of 16 rows or a pixel width of 255 columns, the LARGE DOTS PICTURE file must be used.

AlphaVision, AlphaEclipse, AlpaPremiere, and 7000 series signs support both SMALL DOTS PICTURE and LARGE DOTS PICTURE files.

SMALL DOTS PICTURE files are "called" from TEXT files using the TEXT file Control character for a "Call SMALL DOTS PICTURE" file. For further information, see "Appendix G: Alpha® protocol ASCII table" on page 75.

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

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

SPECIAL NOTE

If a graphic's height and width are *greater* than 31 x 255 pixels, then a LARGE DOTS PICTURE file format must be used.

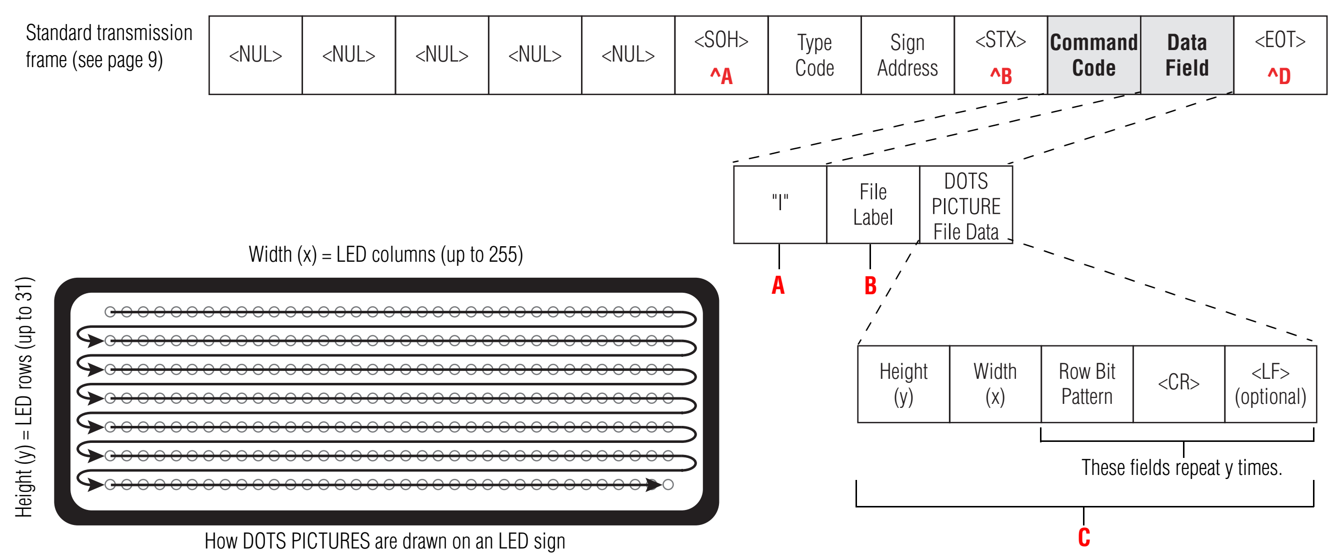
LARGE DOTS PICTURE files can only be used on Full Matrix AlphaVision signs and Series 7000 signs.

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

SHOW ME

An example of the Write SMALL DOTS PICTURE file frame is on page 70.

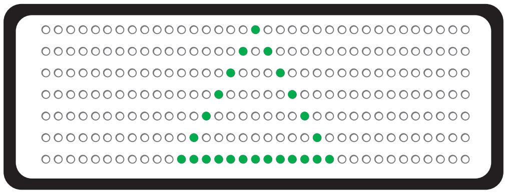
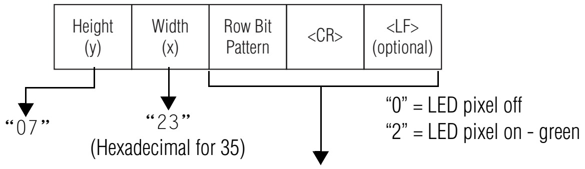
Table 19: Write SMALL DOTS PICTURE file transmission frame format



How DOTS PICTURES are drawn on an LED sign

Item	Name			Description
A	Command Code			"I" (49H) = Write DOTS PICTURE file
B	Data Field	File Label		One ASCII character that indicates the DOTS PICTURE file being accessed. See "Appendix A: Valid File Labels" on page 43.
C		DOTS PICTURE File Data	Height (y)	Two ASCII hexadecimal bytes that represent the number of pixel rows in a 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 13, "Write SPECIAL FUNCTION Command Code format," on page 20. 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).

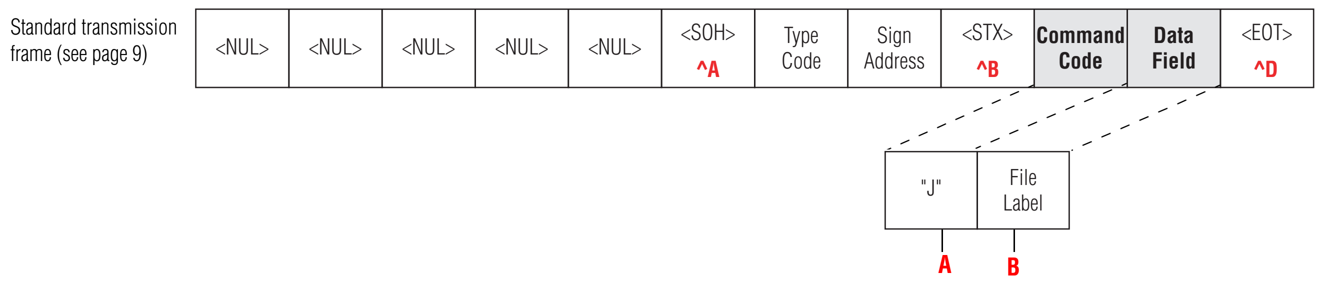
Table 19: Write SMALL DOTS PICTURE file transmission frame format

C (cont)	Data Field (cont)	SMALL DOTS PICTURE File Data (cont)	Width (x)	Two ASCII hexadecimal bytes that represent the number of pixel columns in a DOTS PICTURE (a value from 0 to 255). This must match the pixel column bytes set up in "Set ALPHAVISION DOTS PICTURE Memory Configuration" in Table 13, "Write SPECIAL FUNCTION Command Code format," on page 20. 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 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.
			[1]Row Bit Pattern	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 "4" 34H = pixel on - dim red "1" 31H = pixel on - red "5" 35H = pixel on - dim green "2" 32H = pixel on - green "6" 36H = pixel on - brown "3" 33H = pixel on - amber "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:  Height (y) Width (x) Row Bit Pattern <CR> <LF> (optional) "07" "23" (Hexadecimal for 35) "0" = LED pixel off "2" = LED pixel on - green "00000000000000000200000000000000000"<CR><LF> "00000000000000002020000000000000000"<CR><LF> "00000000000000020002000000000000000"<CR><LF> "00000000000000200000200000000000000"<CR><LF> "00000000000002000000020000000000000"<CR><LF> "00000000000020000000002000000000000"<CR><LF> "00000000000200000000000200000000000"<CR><LF> "00000000002222222222222220000000000"<CR><LF> Row delimiter character <CR> (0DH). The last <CR> is optional. If <LF>s are sent, they will not be sent back in a Read DOTS PICTURE response. (See "Read DOTS PICTURE file Command Code — "J" (4AH)" on page 38.) 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: [1]DATA COMPRESSION — Row Bit Pattern can be data compressed as follows for ALPHAVISION 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 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.

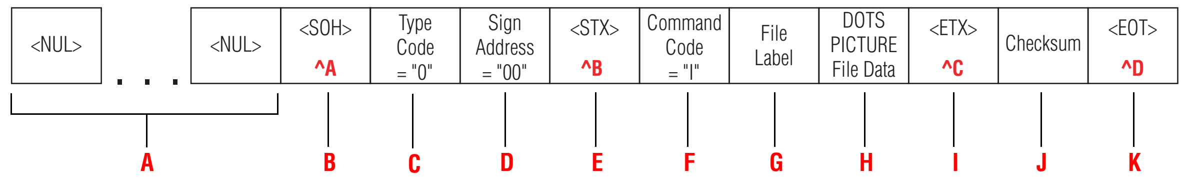
Table 20: Read DOTS PICTURE file transmission frame format



Item	Name		Description
A	Command Code		“J” (4AH) = Read DOTS PICTURE file
B	Data Field	File Label	One ASCII character that indicates the DOTS PICTURE file being accessed. See “Appendix A: Valid File Labels” on page 43.

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

Table 21: Read DOTS PICTURE file sign response frame 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” is returned by the sign. (The Write DOTS PICTURE file Command Code.)
G	File Label	One ASCII character that indicates the DOTS PICTURE file being accessed. See “Appendix A: Valid File Labels” on page 43.
H	DOTS PICTURE File Data	See Table 19, “Write SMALL DOTS PICTURE file transmission frame 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

6.5 ALPHAVISION DOTS PICTURE file commands

ALPHAVISION DOTS PICTURE files are used to store dot patterns which may be "called" from a TEXT file. The main purpose of ALPHAVISION DOTS PICTURE files are to allow users to display custom graphics, such as logos.

Full Matrix ALPHAVISION signs and Series 7000 signs support both DOTS PICTURE and ALPHAVISION DOTS PICTURE files.

ALPHAVISION DOTS PICTURE files are "called" from TEXT files using the TEXT file Control character for a "Call ALPHAVISION DOTS PICTURE" file. For further information, see "Appendix G: Alpha® protocol ASCII table" on page 75.

When an ALPHAVISION DOTS PICTURE file is sent to a sign, the sign will go blank until the transmission is complete.

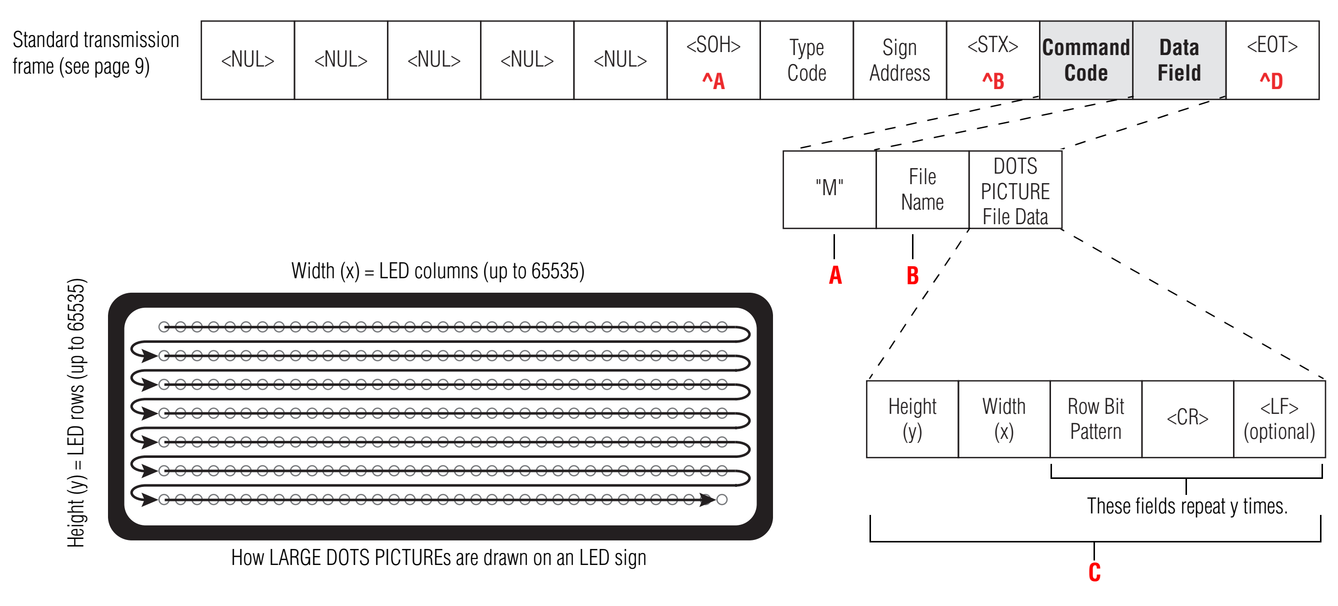
When reading from an ALPHAVISION DOTS PICTURE file, once the transmission frame has been sent, a sign will either pause or blank, depending on the type of sign. Once a sign has completely transmitted the file, the sign will continue displaying the message from where it was interrupted.

SPECIAL NOTE

A LARGE DOTS PICTURE file could, in theory, be 65535 pixels high by 65535 pixels wide. On the other hand, a SMALL DOTS PICTURE cannot exceed a pixel height of 31 rows and a pixel width of 255 columns.

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

Table 22: Write ALPHAVISION DOTS PICTURE file transmission frame format

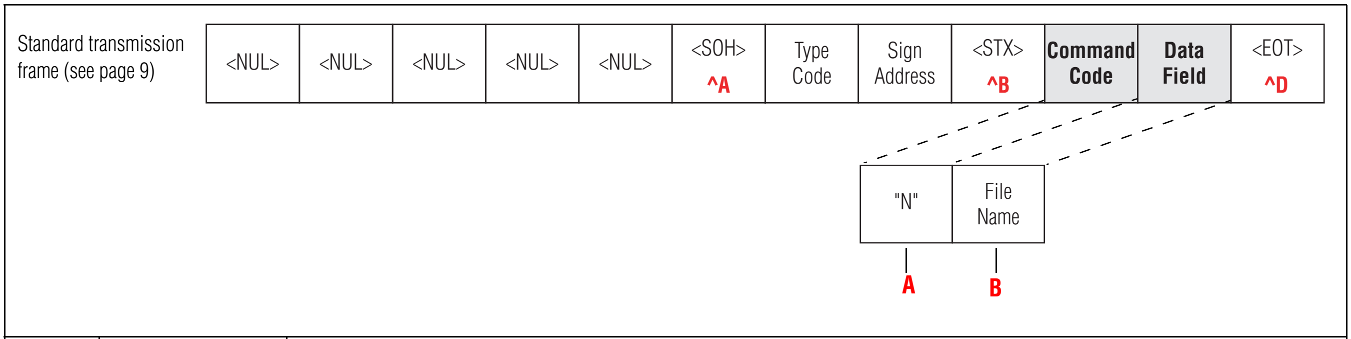

Standard transmission frame (see page 9)
<NUL>
<NUL>
<NUL>
<NUL>
<NUL>
<SOH> ^A
Type Code
Sign Address
<STX> ^B
Command Code
Data Field
<EOT> ^D
"M"
File Name
DOTS PICTURE File Data
A
B
Width (x) = LED columns (up to 65535)
Height (y) = LED rows (up to 65535)
How LARGE DOTS PICTUREs are drawn on an LED sign
Height (y)
Width (x)
Row Bit Pattern
<CR>
<LF> (optional)
These fields repeat y times.
C

Item	Name			Description
A	Command Code			"M" (4DH) = Write ALPHAVISION DOTS PICTURE file
B	Data Field	File Name		Nine ASCII characters that indicate the ALPHAVISION DOTS PICTURE file being accessed.
C		DOTS PICTURE File Data	Height (y)	See "DOTS PICTURE File Data" in Table 19, "Write SMALL DOTS PICTURE file transmission frame format," on page 36.
			Width (x)	
			Row Bit Pattern	
			<CR>	
			<LF>	

6.5.2 Read ALPHAVISION 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.

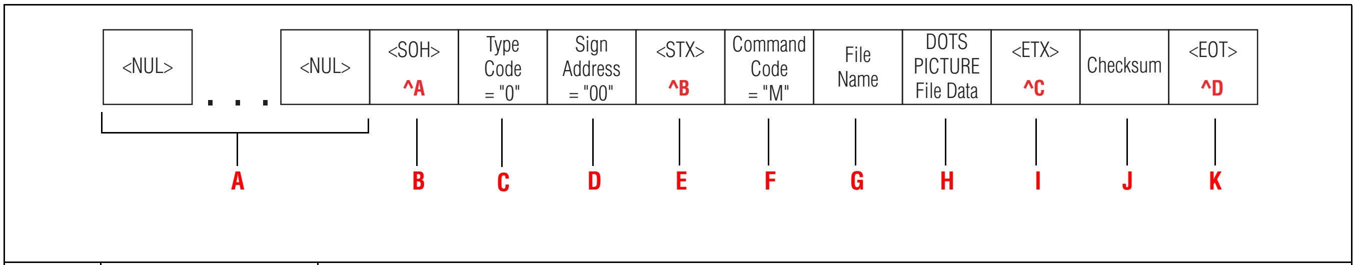
Table 23: Read ALPHAVISION DOTS PICTURE file transmission frame format



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

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

Table 24: Read ALPHAVISION DOTS PICTURE file sign response frame 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” is returned by the sign. (This is the Write ALPHAVISION 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 43.
H	DOTS PICTURE File Data	See Table 19, “Write SMALL DOTS PICTURE file transmission frame 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

6.6 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.

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

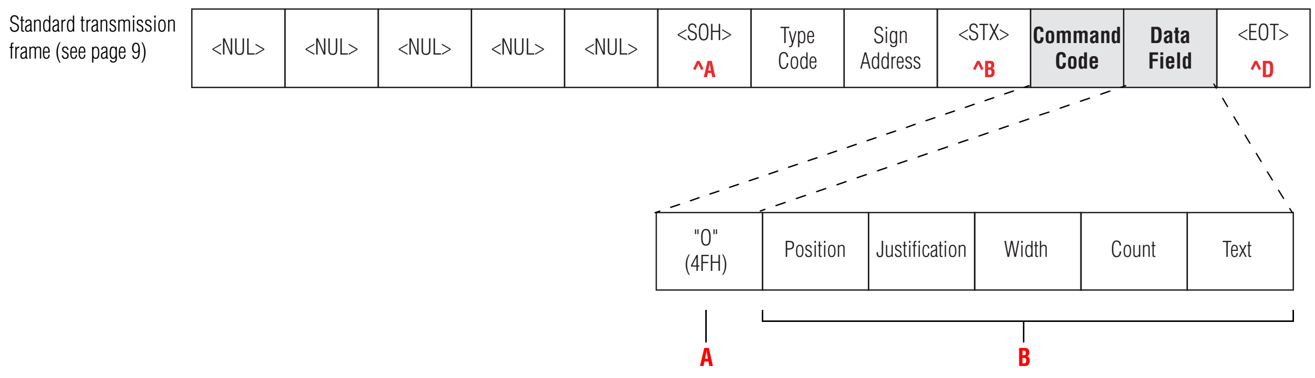
Only AlphaVision and 7000 series signs support this command.

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 25: Write ALPHAVISION BULLETIN MESSAGE file transmission frame format

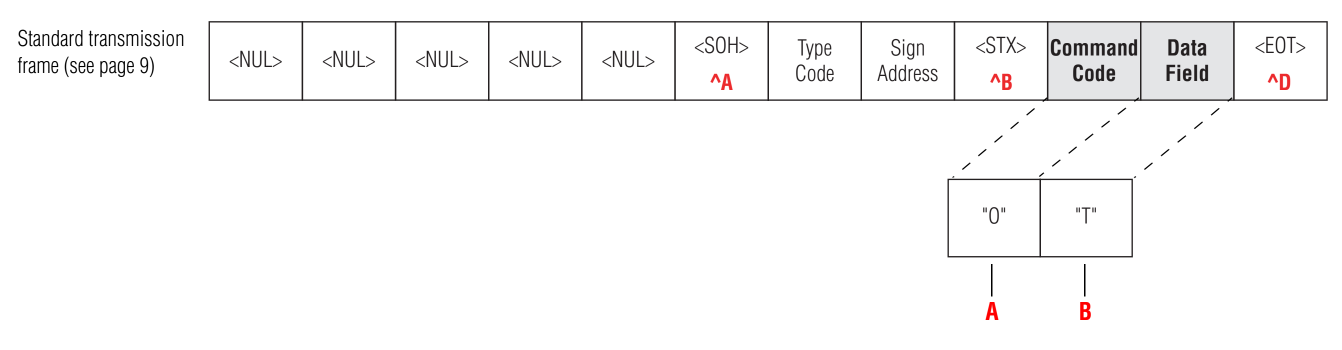


Item	Name		Description
A	Command Code		"O" (4FH) = Write ALPHAVISION BULLETIN MESSAGE file
B	Data Field	Position	One ASCII character that stands for the postion of the bulletin message on a sign. Valid values are: "T" 54H = Top of the display "B" 42H = Bottom of the display
		Justification	One ASCII 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 75.)

6.6.2 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 26: Terminate ALPHAVISION BULLETIN MESSAGE file transmission frame format



Standard transmission frame (see page 9)

<NUL>	<NUL>	<NUL>	<NUL>	<NUL>	<SOH> ^A	Type Code	Sign Address	<STX> ^B	**Command Code**	**Data Field**	<EOT> ^D

Item	Name	Description
A	Command Code	“O” (4FH)
B	Data Field	“T” (54H) is the terminator 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 19) which is pre-configured as a set portion of memory outside of the Memory Pool.

Table 27: 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 19).

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 45) 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 28: 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 frimware upgrade.

7.3.1 Displaying Counter values

SHOW ME

An example of displaying a Counter value is on page 60.

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 75).

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 20.)

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 FUNCTIONs Command Code — “F” (46H)” on page 27.)

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” thru “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 20.)

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 20.)

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:

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.

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.

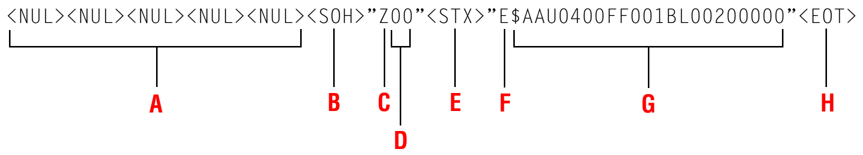
NOTE: The default character spacing is proportional, rather than fixed width. Because of this, a sign’s auto-centering will move the displayed data around with the changing character widths in order to keep the data centered.
To avoid this distracting data movement on a sign:
(a) always send the same 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 frame could be sent to a network of signs:

Table 29: Using STRING files example: STEP 1

This following is a Standard Transmission Frame (see page 9):



```
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"E$AAU0400FF001BL00200000"<EOT>
```

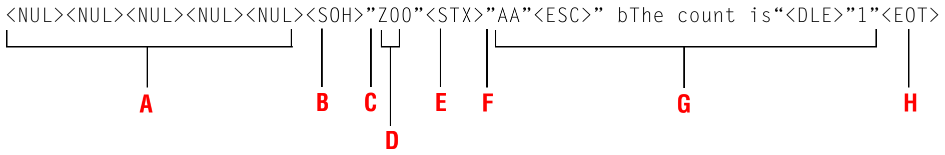
Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>	01H	Start Of Header character
C	Type Code	“Z”	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	“00”	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 20.)
G	Data Field	“$AAU0400FF001BL00200000”	“$” is the Write SPECIAL FUNCTIONs Command Code for **Set Memory Configuration** (see Table 13, “Write SPECIAL FUNCTION Command Code format,” on page 20). 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 30: Using STRING files example: STEP 2

This following is a Standard Transmission Frame (see page 9):



```
<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”AA”<ESC>” bThe count is“<DLE>”1”<EOT>
```

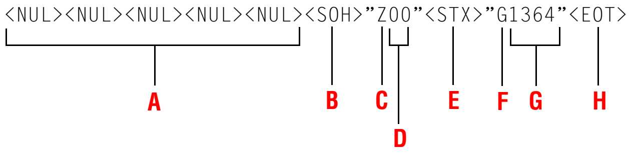
Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>	01H	Start Of Header character
C	Type Code	“Z”	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
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 31: Using STRING files example: STEP 3

This following is a Standard Transmission Frame (see page 9):



```
<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”G1364”<EOT>
```

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”	This is the “Write STRING file” Command Code. (See Table 16, “Write STRING file transmission frame format,” on page 34.)
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 web site: www.ams-i.com.

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

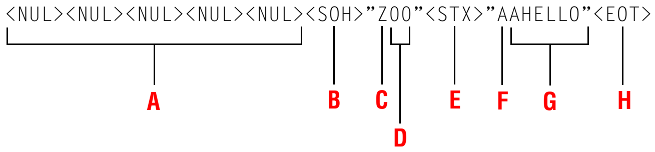
NOTE: In the following examples, it’s assumed that the Memory Configuration table (**Table 13** on page 20) in each sign has already been set up properly.

7.6.1 Standard transmission frame 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:

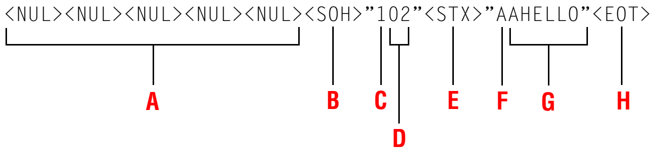
Table 32: Send a message to all signs example



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”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
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 33: Send a message to all 1-line signs on a network with a Sign Address of 02H example

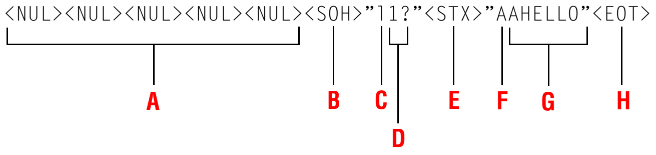


```
<NUL><NUL><NUL><NUL><NUL><SOH>”102”<STX>”AAHELLO”<EOT>
```

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”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
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 34: Send a message to all Series 7000 signs on a network with Sign Addresses 10H through 1FH example



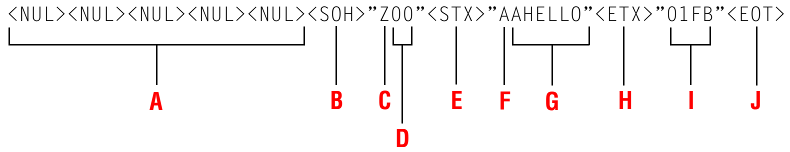
```
<NUL><NUL><NUL><NUL><NUL><SOH>”l1?”<STX>”AAHELLO”<EOT>
```

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		“l”	This means that this transmission is directed to all Series 7000 signs.
D	Sign Address		“1?”	This 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”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
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 frame 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 52 except that a Checksum is used in the following example:

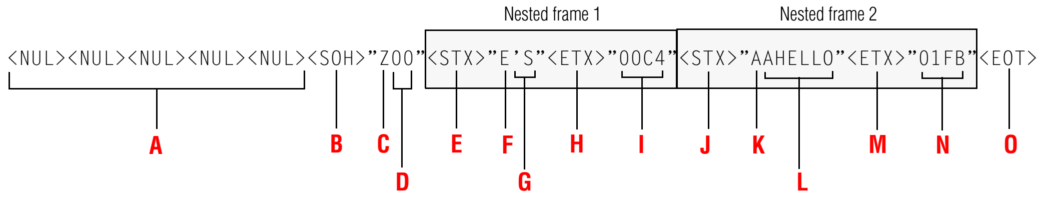
Table 35: Transmission frame with Checksum example


<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”AAHELLO”<ETX>”01FB”<EOT>
A
B
C
D
E
F
G
H
I
J

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”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
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 frame example

Table 36: Nesting with checksums transmission frame example

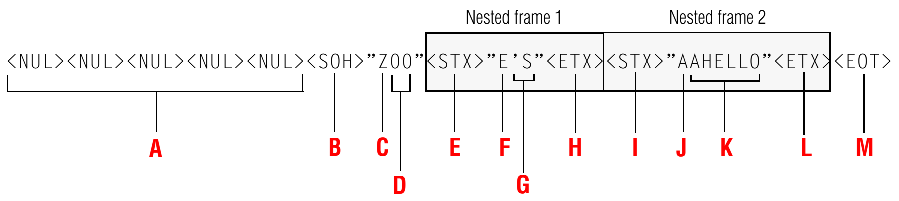

Nested frame 1
Nested frame 2
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"E'S"<ETX>"00C4"<STX>"AAHELLO"<ETX>"01FB"<EOT>
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O

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	Nested frame 1	<STX>		02H	Start of Nested frame 1
F		Command Code		“E”	This is the “Write SPECIAL FUNCTIONs” Command Code. (See Table 13, “Write SPECIAL FUNCTION Command Code format,” on page 20.)
G		Data Field	Special Functions Label	“‘”	“‘” (27H) means Set Time Format
			Special Functions Data	“S”	This sets the sign’s time to the standard am/pm format.
H		<ETX>		03H	End of Nested frame 1
I		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 frame 2	<STX>		02H	Start of Nested frame 2
K		Command Code		“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
L		Data Field	File Label	“A”	File Label of the TEXT file
			ASCII Message	“HELLO”	The actual text to be displayed on a sign
M		<ETX>		03H	End of Nested frame 2
N		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 frame example

This frame is identical to the previous frame in **Table 36** on page 55 except that the Checksums are omitted after each nested frame’s <ETX>:

Table 37: Nesting without Checksums transmission frame example


Nested frame 1
Nested frame 2
<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”E’S”<ETX><STX>”AAHELLO”<ETX><EOT>
A
B
C
E
F
H
I
J
K
L
M
D
G

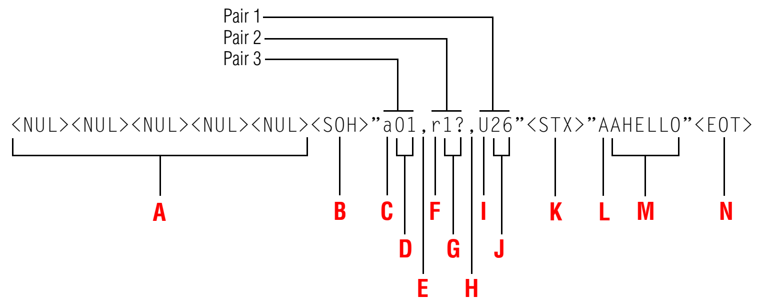
Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>			01H	Start Of Header character
C	Type Code			“Z”	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address			“00”	This means all signs on the network should “listen” to this transmission.
E	Nested frame 1	<STX>		02H	Start of Nested frame 1
F		Command Code		“E”	This is the “Write SPECIAL FUNCTIONs” Command Code. (See Table 13, “Write SPECIAL FUNCTION Command Code format,” on page 20.)
G		Data Field	Special Functions Label	“’”	“’” (27H) means Set Time Format
			Special Functions Data	“S”	This sets the sign’s time to the standard am/pm format.
H		<ETX>		03H	End of Nested frame 1
I	Nested frame 2	<STX>		02H	Start of Nested frame 2
J		Command Code		“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
K		Data Field	File Label	“A”	File Label of the TEXT file
			ASCII Message	“HELLO”	The actual text to be displayed on a sign
L		<ETX>		03H	End of Nested frame 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 <u>and</u> Director signs <u>and</u> 790i signs on the network.

Table 38: Multiple Type Codes / Sign Addresses example


Pair 1
Pair 2
Pair 3
<NUL><NUL><NUL><NUL><NUL><SOH>"a01,r1?,U26"<STX>"AAHELLO"<EOT>
A
B
C
F
I
K
L
M
N
D
G
J
E
H

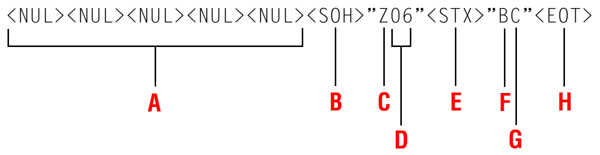
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Pair 1	Type Code	“a”	This means that this transmission is directed to all 4120C signs.
D		Sign Address	“01”	This means only 4120C signs with a Sign Address of 01H on the network should “listen” to this transmission.
E	Delimiter		“,”	This separates each Type Code/Sign Address pair.
F	Pair 2	Type Code	“r”	This means that this transmission is directed to all Director signs.
G		Sign Address	“1?”	This means that all signs with a Sign Address between 10H and 1FH inclusive on the network should “listen” to this transmission.
H	Delimiter		“,”	This separates each Type Code/Sign Address pair.
I	Pair 3	Type Code	“U”	This means that this transmission is directed to all 790i signs.
J		Sign Address	“26”	This 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”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
M	Data Field	File Label	“A”	File Label of the TEXT file
		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 40** on page 58.

Table 39: Read TEXT file example



```
<NUL><NUL><NUL><NUL><NUL><SOH>"Z06"<STX>"BC"<EOT>
```

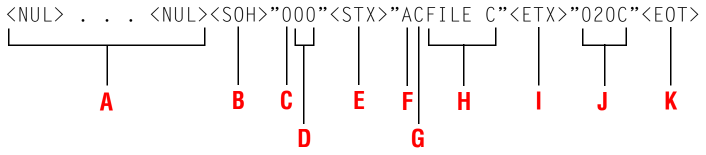
Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	This means that this transmission is directed to all signs.
D	Sign Address	"06"	This means only signs with a Sign Address of 06H on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"B"	This is the "Read TEXT file" Command Code. (See Table 11, "Read TEXT file transmission frame format," on page 18.)
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 39** on page 58.

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 40: Response to Read TEXT file example



```
<NUL> . . . <NUL><SOH>"000"<STX>"ACFILE C"<ETX>"020C"<EOT>
```

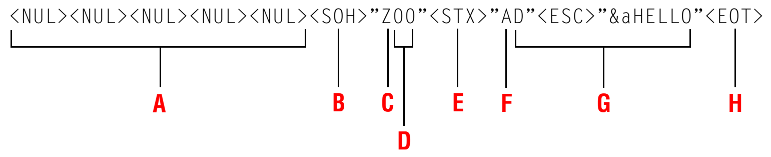
Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> characters
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	The Response Type Code
D	Sign Address	"00"	"00" is always sent.
E	<STX>	02H	Start of TeXt character
F	Command Code	"A"	This is sent in response to the "Read TEXT file" Command Code.
G	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 41: Rotate “Hello” example

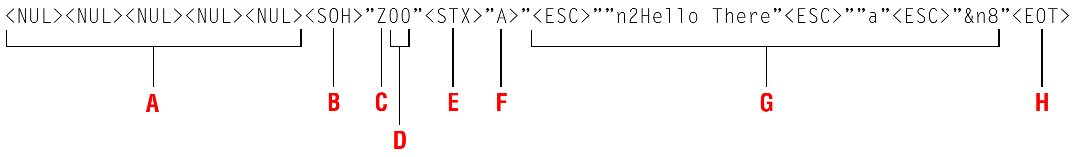


```
<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”AD”<ESC>”&aHELLO”<EOT>
```

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 signs.
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”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
G	Data Field	File Label		“D”	File Label of the TEXT file that will be written
		Mode Field	<ESC>	1BH	Escape character
			Display Position	“&”	This means that the ASCII Message should be displayed on the bottom line of a sign.
			Mode Code	“a”	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 42: Combining text and graphics example



```
<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”A>”<ESC>””n2Hello There”<ESC>””a”<ESC>”&n8”<EOT>
```

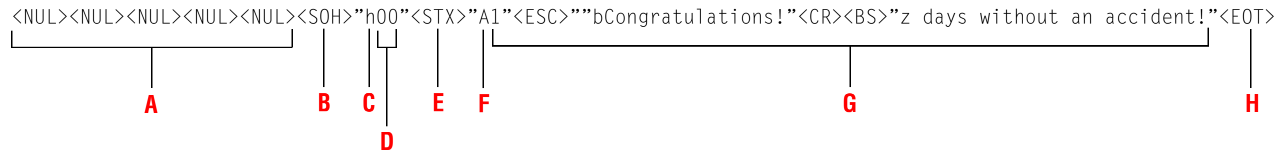
Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>	01H	Start Of Header character
C	Type Code	“Z”	This means that this transmission is directed to all signs.
D	Sign Address	“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)

Table 42: 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 70).
				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 70).
				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 43: Displaying a Counter value example



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

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”	This means that this transmission is directed to all 4160R signs.
D	Sign Address	“00”	This means all 4160R signs on the network should “listen” to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)

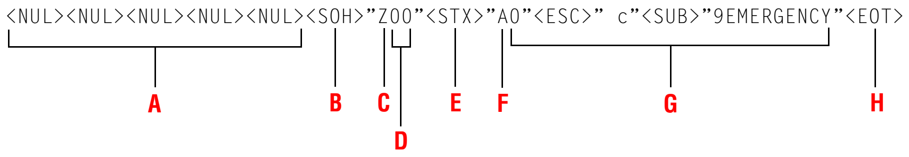
Table 43: Displaying a Counter value example

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
				Display Position	“““	“““ (22H) means that the ASCII Message will begin on the Top Line of the sign
				Mode Code	“b”	“b” (62H) is the HOLD Mode Code (see page 70)
			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 45).
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 44: Write a Priority TEXT file example

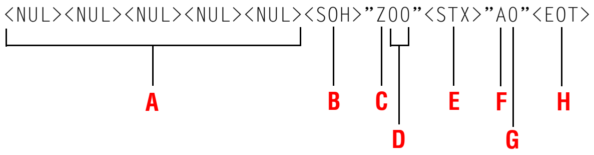

```
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"A0"<ESC>" c"<SUB>"9EMERGENCY"<EOT>
```

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 signs.
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”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
G	Data Field	File Label			“0”	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 70)
			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 75).
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 45: Disable a Priority TEXT file example


```
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"A0"<EOT>
```

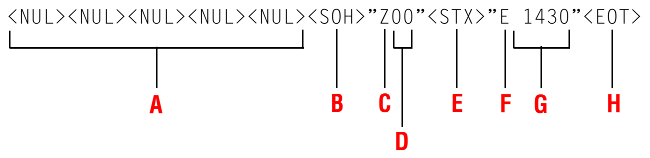
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 17.)
G	Data Field	File Label	“0”	Priority TEXT File Label
H	<EOT>		04H	End Of Transmission character

7.6.8 SPECIAL FUNCTIONs examples

7.6.8.1 Write SPECIAL FUNCTIONs example

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

Table 46: Write SPECIAL FUNCTIONs example

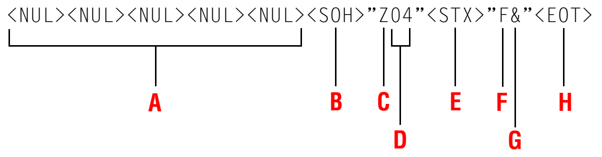


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“E”	This is the “Write SPECIAL FUNCTIONs file” Command Code. (See Table 13, “Write SPECIAL FUNCTION Command Code format,” on page 20.)
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 FUNCTIONs example

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

Table 47: Read SPECIAL FUNCTIONs example

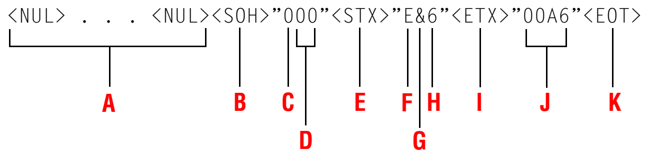


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“04”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“F”	This is the “Read SPECIAL FUNCTIONs file” Command Code. (See Table 14, “Read SPECIAL FUNCTIONs file transmission frame format,” on page 27.)
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 FUNCTIONs example

The following is the response to the Read SPECIAL FUNCTIONs example in **Table 47** above:

Table 48: Response to Read SPECIAL FUNCTIONs 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”	This is the “Read SPECIAL FUNCTIONs file” Command Code. (See Table 14, “Read SPECIAL FUNCTIONs file transmission frame format,” on page 27.)
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

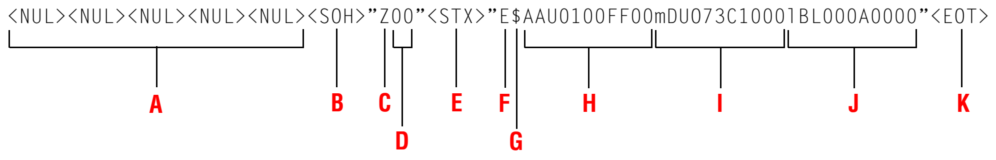
7.6.8.4 SPECIAL FUNCTIONs data formats example

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



<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”E$AAU0100FF00mDU073C1000lBL000A0000”<EOT>
A
B
C
D
E
F
G
H
I
J
K

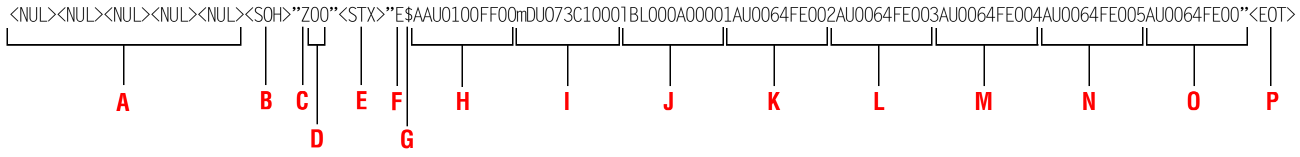
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		“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 file” Command Code. (See Table 13, “Write SPECIAL FUNCTION Command Code format,” on page 20.)
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 49**) 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 45.)

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 50: Set Memory Configuration example #2 — Counter data included



<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”E$AAU0100FF00mDU073C1000IBL000A00001AU0064FE002AU0064FE003AU0064FE004AU0064FE005AU0064FE00”<EOT>

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 file” Command Code. (See Table 13, “Write SPECIAL FUNCTION Command Code format,” on page 20.)
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	“IBL000A0000”	These bytes mean the following: “I” = 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)

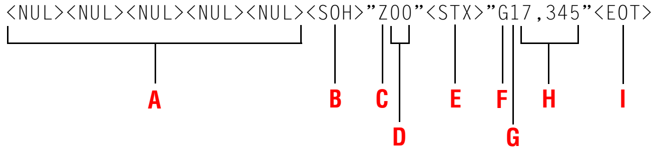
Table 50: Set Memory Configuration example #2 — Counter data included

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

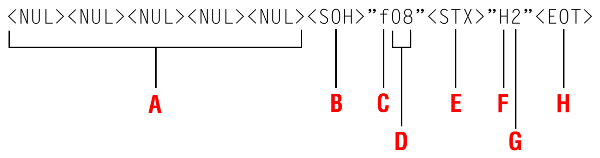
Table 51: Write STRING file example



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”	This means that this transmission is directed to all signs.
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”	This is the “Write STRING file” Command Code. (See Table 16, “Write STRING file transmission frame format,” on page 34.)
G	Data Field	File Label	“1”	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 52: Read STRING file example



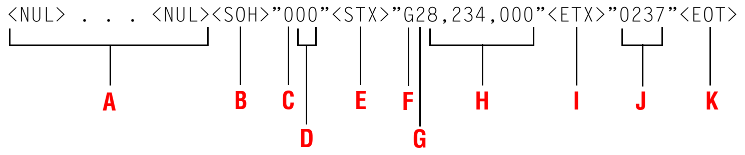
```
<NUL><NUL><NUL><NUL><NUL><SOH>"f08"<STX>"H2"<EOT>
```

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"	This means that this transmission is directed to all 215C signs.
D	Sign Address	"08"	This 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"	This is the "Read STRING file" Command Code. (See Table 17, "Read STRING file transmission frame format," on page 35.)
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 52**) example:

Table 53: Response to Read STRING file example



```
<NUL> . . . <NUL><SOH>"000"<STX>"G28,234,000"<ETX>"0237"<EOT>
```

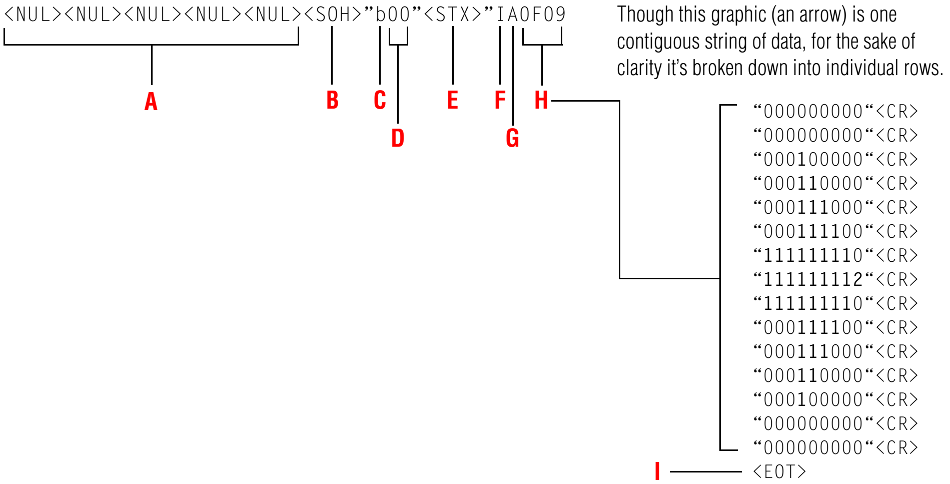
Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> (00H) characters
B	<SOH>	01H	Start Of Header character
C	Type Code	"0"	"0" (30H) is the Response code
D	Sign Address	"00"	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	02H	Start of TeXt character
F	Command Code	"G"	"G" is 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 54: Write DOTS PICTURE file example



```
<NUL><NUL><NUL><NUL><NUL><SOH>”b00”<STX>”IA0F09
“000000000“<CR>
“000000000“<CR>
“000100000“<CR>
“000110000“<CR>
“000111000“<CR>
“000111100“<CR>
“111111110“<CR>
“111111112“<CR>
“111111110“<CR>
“000111100“<CR>
“000111000“<CR>
“000110000“<CR>
“000100000“<CR>
“000000000“<CR>
“000000000“<CR>
<EOT>
```

Though this graphic (an arrow) is one contiguous string of data, for the sake of clarity it’s broken down into individual rows.

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”	This means that this transmission is directed to all 4160C signs.
D	Sign Address			“00”	This means all 4160C signs on the network should “listen” to this transmission.
E	<STX>			02H	Start of TeXt character
F	Command Code			“I”	This is the “Write DOTS PICTURE file” Command Code. (See Table 19, “Write SMALL DOTS PICTURE file transmission frame format,” on page 36.)
G	File Label			“A”	File Label of the DOTS file
H	Data Field	DOTS PICTURE File Data	Height (y)	“0F”	“0F” (15D) = pixel height of graphic
			Width (x)	“09”	“09” (9D) = pixel width of graphic
			Row Bit Pattern	“000000000“<CR> “000000000“<CR> “000**1**00000“<CR> “000**11**0000“<CR> “000**111**000“<CR> “000**1111**00“<CR> “**11111111**0“<CR> “**111111112**“<CR> “**11111111**0“<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” = sign pixel off “1” = sign pixel on - red “2” = sign pixel on - green
I	<EOT>			04H	End Of Transmission character

7.6.11 Displaying text at XY position examples

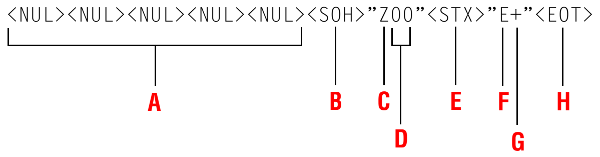
Text messages up to 250 characters can be displayed in a particular location on AlphaVision character matrix sign. This can be done by specifying a character position in a sign line (X) and a line position (Y) using the SPECIAL FUNCTION "+" command (see page 21).

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 55: Enable SPECIAL FUNCTION XY positioning example



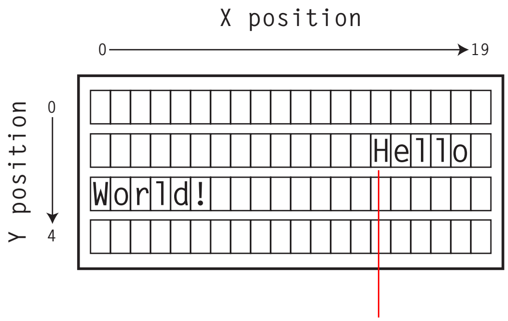
Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	This means that this transmission is directed to all signs.
D	Sign Address	"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	This is the "Write SPECIAL FUNCTIONs example" on page 63.
G	Data Field	"+"	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.

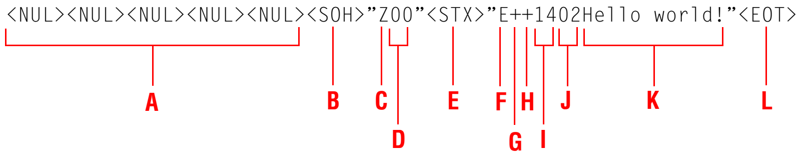
The text “Hello world!” will be displayed starting at character position 14 (X) on line 2 (Y) as shown in the illustration below.

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



The text starts at the specified XY position (14, 2). Notice that because it doesn’t fit on the line, the text wraps onto the next line.

Table 56: Display text at an XY location example



```
<NUL><NUL><NUL><NUL><NUL><SOH>”Z00”<STX>”E++1402Hello world!”<EOT>
```

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 signs.
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 example” on page 63.
G	Data Field	Special Functions Label	“+”	Enable XY positioning
H		File Label	“+”	File Label
I		X position	“14”	Two ASCII decimal digits that represent the character position
J		Y position	“02”	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 $X_3 = 13$, $Y_3 = 4$.

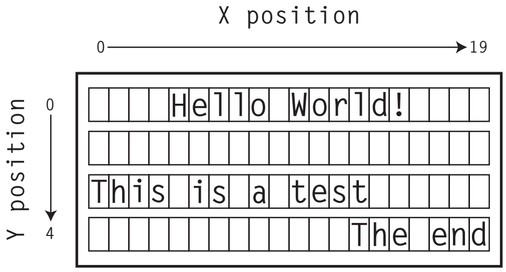
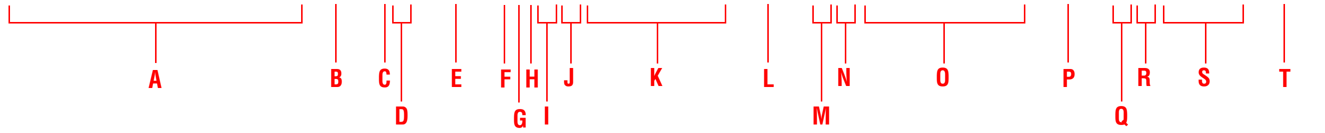


Table 57: Display multiple text at XY locations example

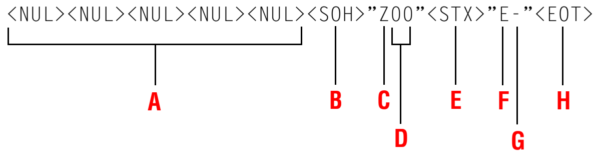
```
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"E++0400Hello world!"<DC2>"0003This is a test"<DC2>"1304The end"<EOT>
```



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 signs.
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 example” on page 63.
G	Data Field	Special Functions Label	“+”	Enable XY positioning
H		File Label	“+”	File Label
I		X_1 position	“04”	Two ASCII decimal digits that represent the character position of the first text message
J		Y_1 position	“00”	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”	Two ASCII decimal digits that represent the character position of the second text message
N		Y_2 position	“03”	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”	Two ASCII decimal digits that represent the character position of the third text message
R		Y_3 position	“04”	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 58: Disable SPECIAL FUNCTION XY positioning example



```
<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"E-"<EOT>
```

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 signs.
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 example” on page 63.
G	Data Field	“-”	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)

	Dec	Hex	Char	Meaning
Control characters	0	00	^@	NULL
	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 tempterature in Celsius (only on Solar, 790i, 460i, 440i, and 430i) • 08H + “^]” (1DH) = display tempterature 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 • BH + “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 (0FH)” on page 97. (Alpha2.0 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 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)

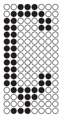
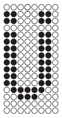
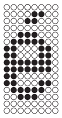
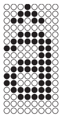
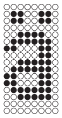
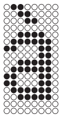
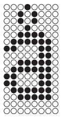
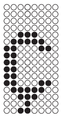
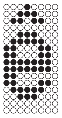
	Dec	Hex	Char	Meaning
Control characters (continued)	26	1A	^Z	Select character set (2-byte format): • 1AH + “1” (31H) = Five high standard (or Five slim[1]) • 1AH + “2” (32H) = Five stroke[1] • 1AH + “3” (33H) = Seven high standard (or Seven slim[1]) • 1AH + “4” (34H) = Seven stroke[1] • 1AH + “5” (35H) = 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) = Five high custom character set (Alpha 2.0 protocol only)[2] • 1AH + “X” (58H) = Seven high custom character set (Alpha 2.0 protocol only)[2] • 1AH + “Y” (59H) = Ten high custom character set (Alpha 2.0 protocol only)[2] • 1AH + “Z” (5AH) = Fifteen high custom character set (Alpha 2.0 protocol only)[2] [1] only applies to Betabrite model 1036 signs. [2] see “Custom character sets” on page 103.
	27	1B	^[	Start of Mode field
	28	1C	^\	Select character color (2-byte format) — 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
	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 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 ALPHAVISION DOTS PICTURE file (15-byte format): 1FH + SFFFFFFFFFtttt where • S = “C” (43H) if the file is running as part of a Quick Flick animation. The display is cleared before each ALPHAVISION DOTS PICTURE is shown. S = “L” (4CH) if the file running is a 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. Leading 0’s are ignored. For example, “0020” = 32 tenths of seconds = 3.2 seconds.

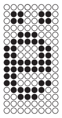
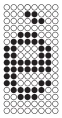
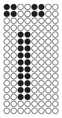
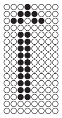
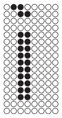
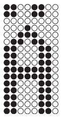
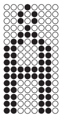
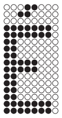
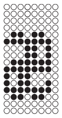
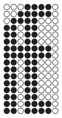
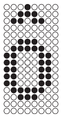
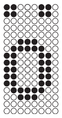
	Dec	Hex	Char	Meaning
Standard ASCII characters	32	20	space	
	33	21	!	
	34	22	“	
	35	23	#	
	36	24	$	
	37	25	%	
	38	26	&	
	39	27	‘	
	40	28	(	
	41	29	)	
	42	2A	*	
	43	2B	+	
	44	2C	,	
	45	2D	-	
	46	2E	.	
	47	2F	/	
	48	30	0	
	49	31	1	
	50	32	2	
	51	33	3	
	52	34	4	
	53	35	5	
	54	36	6	
	55	37	7	
	56	38	8	
	57	39	9	
	58	3A	:	
	69	3B	;	
	60	3C	<	
	61	3D	=	
	62	3E	>	
	63	3F	?	
	64	40	@	
	65	41	A	
	66	42	B	
	67	43	C	
	68	44	D	
	69	45	E	
	70	46	F	
	71	47	G	
	72	48	H	
	73	49	I	
	74	4A	J	
	75	4B	K	
	76	4C	L	
	77	4D	M	
	78	4E	N	
	79	4F	O	
	80	50	P	
	81	51	Q	
	82	52	R	
	83	53	S	
	84	54	T	
	85	55	U	
	86	56	V	

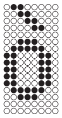
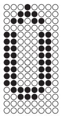
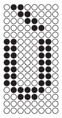
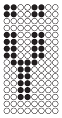
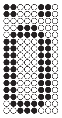
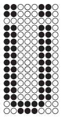
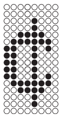
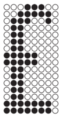
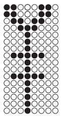
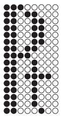
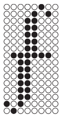
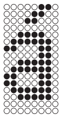
	Dec	Hex	Char	Meaning
Standard ASCII characters	87	57	W	
	88	58	X	
	89	59	Y	
	90	5A	Z	
	91	5B	[	
	92	5C	\	
	93	5D	]	
	94	5E	¢	
	95	5F	_	
	96	60	‘	
	97	61	a	
	98	62	b	
	99	63	c	
	100	64	d	
	101	65	e	
	102	66	f	
	103	67	g	
	104	68	h	
	105	69	i	
	106	6A	j	
	107	6B	k	
	107	6C	l	
	109	6D	m	
	110	6E	n	
	111	6F	o	
	112	70	p	
	113	71	q	
	114	72	r	
	115	73	s	
	116	74	t	
	117	75	u	
	118	76	v	
	119	77	w	
	120	78	x	
	121	79	y	
	122	7A	z	
	123	7B	{	
	124	7C	|	
	125	7D	}	
	126	7E	1/2 space	
	127	7F	block	

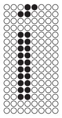
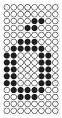
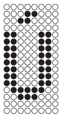
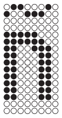
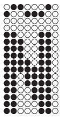
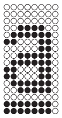
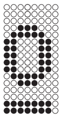
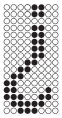
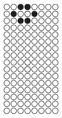
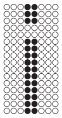
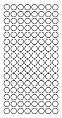
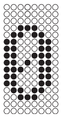
7.7.2 Extended character set (80 - C1H)

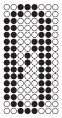
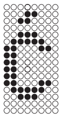
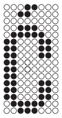
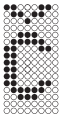
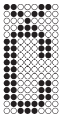
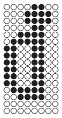
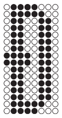
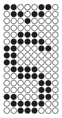
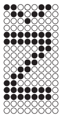
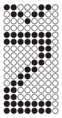
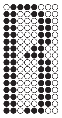
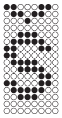
The following characters can be displayed by combining a control code (^H) with an offset (as shown below).

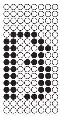
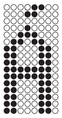
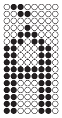
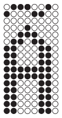
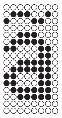
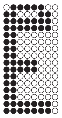
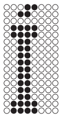
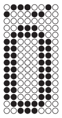
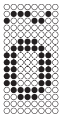
	Dec	Hex	Char	Control code combination
Extended character set	128	80		08H + 20H
	129	81		08H + 21H
	130	82		08H + 22H
	131	83		08H + 23H
	132	84		08H + 24H
	133	85		08H + 25H
	134	86		08H + 26H
	135	87		08H + 27H
	136	88		08H + 28H

	Dec	Hex	Char	Control code combination
Extended character set	137	89		08H + 29H
	138	8A		08H + 2AH
	139	8B		08H + 2BH
	140	8C		08H + 2CH
	141	8D		08H + 2DH
	142	8E		08H + 2EH
	143	8F		08H + 2FH
	144	90		08H + 30H
	145	91		08H + 31H
	146	92		08H + 32H
	147	93		08H + 33H
	148	94		08H + 34H

	Dec	Hex	Char	Control code combination
Extended character set	149	95		08H + 35H
	150	96		08H + 36H
	151	97		08H + 37H
	152	98		08H + 38H
	153	99		08H + 39H
	154	9A		08H + 3AH
	155	9B		08H + 3BH
	156	9C		y08H + 3CH
	157	9D		08H + 3DH
	158	9E		08H + 3EH
	159	9F		08H + 3FH
	160	A0		08H + 40H

	Dec	Hex	Char	Control code combination
Extended character set	161	A1		08H + 41H
	162	A2		08H + 42H
	163	A3		08H + 43H
	164	A4		08H + 44H
	165	A5		08H + 45H
	166	A6		08H + 46H
	167	A7		08H + 47H
	168	A8		08H + 48H
	169	A9		08H + 49H
	170	AA		08H + 4AH
	171	AB		08H + 4BH
	172	AC		08H + 4CH

	Dec	Hex	Char	Control code combination
Extended character set	173	AD		08H + 4DH
	174	AE		08H + 4EH
	175	AF		08H + 4FH
	176	B0		08H + 50H
	177	B1		08H + 51H
	178	B2		08H + 52H
	179	B3		08H + 53H
	180	B4		08H + 54H
	181	B5		08H + 55H
	182	B6		08H + 56H
	183	B7		08H + 57H
	184	B8		08H + 58H

	Dec	Hex	Char	Control code combination
Extended character set	185	B9		08H + 59H
	186	BA		08H + 5AH
	187	BB		08H + 5BH
	188	BC		08H + 5CH
	189	BD		08H + 5DH
	190	BE		08H + 5EH
	191	BF		08H + 5FH
	192	C0		08H + 60H
	193	C1		08H + 61H
	194	C2	Carriage return symbol	08H + 62H (only applies to Betabrite 1036 signs)
	195	C3	Y punctuation key	08H + 63H (only applies to Betabrite 1036 signs)
	196	C4	Up arrow	08H + 64H (only applies to Betabrite 1036 signs)
	197	C5	Down arrow	08H + 65H (only applies to Betabrite 1036 signs)
	198	C6	Left arrow	08H + 66H (only applies to Betabrite 1036 signs)
	199	C7	Right arrow	08H + 67H (only applies to Betabrite 1036 signs)
	200	C8	Packman	08H + 68H (only applies to Betabrite 1036 signs)
	201	C9	Sail boat	08H + 69H (only applies to Betabrite 1036 signs)
	202	CA	Ball	08H + 6AH (only applies to Betabrite 1036 signs)
	203	CB	Telephone	08H + 6BH (only applies to Betabrite 1036 signs)
	204	CC	Heart	08H + 6CH (only applies to Betabrite 1036 signs)
	205	CD	Car	08H + 6DH (only applies to Betabrite 1036 signs)
	206	CE	Handicap	08H + 6EH (only applies to Betabrite 1036 signs)
	207	CF	Rhino	08H + 6FH (only applies to Betabrite 1036 signs)

	Dec	Hex	Char	Control code combination
Special commands	208	D0	Mug	08H + 70H (only applies to Betabrite 1036 signs)
	209	D1	Satellite dish	08H + 71H (only applies to Betabrite 1036 signs)
	210	D2	Copyright symbol	08H + 72H (only applies to Betabrite 1036 signs)
	211	D3	Male symbol	08H + 73H (only applies to Betabrite 1036 signs)
	212	D4	Female symbol	08H + 74H (only applies to Betabrite 1036 signs)
	213	D5	Bottle	08H + 75H (only applies to Betabrite 1036 signs)
	214	D6	Diskette	08H + 76H (only applies to Betabrite 1036 signs)
	215	D7	Printer	08H + 77H (only applies to Betabrite 1036 signs)
	216	D8	Musical note	08H + 78H (only applies to Betabrite 1036 signs)
	217	D9	Infinity symbol	08H + 79H (only applies to Betabrite 1036 signs)
	Temperature			08H + “^\” (1CH) Displays temperature in Celsius (only on Solar, 790i, 460i, 440i, and 430i).
				08H + “^]” (1DH) Displays temperature in Fahrenheit (only on Solar, 790i, 460i, 440i, and 430i).
	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.

7.8 Appendix H: ISO ASCII table

This is the standard ASCII character set:

Character				Hex	Dec		Character		Hex	Dec
Control characters	NULL	^@	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 seperator	1C	28			\	5C	92
	GS	^]	group seperator	1D	29			]	5D	93
	RS	^^	record seperator	1E	30			^	5E	94
	US	^_	unit seperator	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 59**), the following Special Modes (**Table 60**) or Special Graphics (**Table 61**) can be designated in the Special Specifier field (see "TEXT file commands" on page 16).

Table 59: 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 appoximately one half their normal width. (Only available on certain sign models.)
SPECIAL	"n"	6EH	This is followed by a Special Specifier ASCII character which defines one of the Special Modes. See "Special Modes" on page 87.

7.9.2 Special Modes

Table 60: Special Modes

Mode name	ASCII code	Hex code	Description
TWINKLE	"0"	30H	Message will twinkle on the sign.
SPARKLE	"1"	31H	New message will sparkle over the current message.
SNOW	"2"	32H	Message will "snow" onto the display.
INTERLOCK	"3"	33H	New message will interlock over the current message in alternating rows of dots from each end.
SWITCH	"4"	34H	Alternating characters "switch" off the sign up and down. New message "switches" on in a similar manner.
SLIDE or CYCLE COLORS*	"5"	35H	New message slides onto the sign one character at a time from right to left.
SPRAY	"6"	36H	New message sprays across and onto the sign from right to left.

Table 60: Special Modes

Mode name	ASCII code	Hex code	Description
STARBURST	“7”	37H	“Starbursts” explode the new message onto the sign.
WELCOME	“8”	38H	The word “Welcome” is written in script across the sign.
SLOT MACHINE	“9”	39H	Slot machine symbols appear randomly across the sign.
NEWS FLASH*	“A”	3AH	News flash animation
TRUMPET ANIMATION*	“B”	3BH	Trumpet animation
* only available on Betabrite model 1036 signs			

7.9.3 Special Graphics

Table 61: Special Graphics

Mode name	ASCII code	Hex code	Description
THANK YOU	“S”	53H	The words “Thank You” are written in script across the sign.
NO SMOKING	“U”	55H	A cigarette image appears, is then extinguished and replaced with a no smoking symbol.
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”
RUNNING ANIMAL or FISH ANIMATION*	“W”	57H	An animal runs across the sign.
FIREWORKS	“X”	58H	Fireworks explode randomly across the sign.
TURBO CAR or BALOON ANIMATION*	“Y”	59H	A car drives across the sign.
CHERRY BOMB	“Z”	5AH	A bomb fuse burns down followed by an explosion.
* only available on Betabrite model 1036 signs			

7.9.4 Modes available on signs

Table 62: Modes available on signs

Sign	Modes																				
	Automode	Flash	Hold	Interlock	Roll			Rotate		Scroll	Slide		Snow	Sparkle	Spray	Starburst	Switch		Twinkle	Wipe		
					Up/Down/Left/Right	In/Out (horizontal)	In/Out (vertical)	Standard	Condensed		Slide	Slide -> Cycle Color					Switch	Switch half the display		Up/Down/Left/Right	In /Out (horizontal)	In/Out (vertical)
200 Series	●	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●	
220C	●	●	●	●	●		●	●	●	●	1	1	●	●	●	●		●	●	●		●
300 Series	●	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●	
420C	●	●	●	●	●		●	●	●	●	1	1	●	●	1	●		●	●	●		●
790i (also the 430i, 440i, and 460i)	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●	
4000 Series	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●	
7000 Series	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●	
AlphaEclipse™	●	●	●	●	●		●	●		●	●		●	●	●	●	●	●	●	●	●	●
AlphaPremiere™	●	●	●	●	●		●	●		●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaVision™ (Full Matrix)	●	●	●	●	●	●		●		●			●	●					●	●	●	
AlphaVision™ (Character Matrix)	●	●	●			●														●	●	
Big Dot®	●	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●	
BetaBrite®	●	●	●	●	●		●	●	●	●		●	●	●	●	●		●	●	●		●
Director™	●	●	●																	●	●	
PPD™	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●	
Solar™	●	●	●	●	●	●		●		●	●		●	●	●	●	●		●	●	●	

[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.

7.9.5 Fonts and colors available on signs

Table 63: Fonts and colrs available on signs

Sign	Characters													
	15/16 Row Normal	15/16 Row Fancy	Ten Row	Seven Row Normal	Seven Row Fancy	Five Row	Color (see NOTE)	Normal	Wide	Double Wide	Flashing	Double Height	True Descenders	Fixed Width
200 Series				●	●	●	●	●	●	●				●
220C				●	●	●	●	●	●	●	●			●
300 Series				●	●	●	●	●	●	●	●			●
420C				●	●	●	●	●	●	●	●			●
790i (also the 430i, 440i, and 460i)				●		●		●	●	●				●
4000 Series	●	●		●	●	●	●	●	●	●	●			●
7000 Series	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaEclipse™	●	●		●	●	●		●	●	●	●			●
AlphaPremiere™	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaVision™ (Full Matrix)	●	●	●	●	●	●	●	●	●	●	●	●	●	●
AlphaVision™ (Character Matrix)				●		●	●	●			●			
BetaBrite®				●	●	●	●	●	●	●	●			●
Big Dot®				●	●	●	●	●	●	●	●			●
Director™				●		●	●	●			●			
PPD™				●	●	●		●	●	●	●			●
Solar™	●	●		●	●	●	●	●	●	●	●			●

NOTE: Sign names ending in “C”, such as 4120C, have color capabilities. Sign names ending in “R”, such as 4120R, can display in red only.

7.9.6 Display options available on signs

Table 64: Display options available on signs

Sign	Options															
	Time	Date	Temperature		Speed	New Line	New Page	Animation	String	Ticker Symbol	Variable	Counter	Graphic (see NOTE)	Gif (see NOTE)	Flick (see NOTE)	Message
			Fahrenheit	Celsius												
200 Series	●	●			●	●		●	●	●	●	●	●			●
220C	●	●			●	●		●	●	●	●	●	●			●
300 Series	●	●			●	●		●	●	●	●	●	●			●
420C	●	●			●	●		●	●	●	●	●	●			●
790i (also the 430i, 440i, and 460i)	●		●	●	●	●		●	●	●	●	●	●			●
4000 Series	●	●			●	●		●	●	●	●	●	●			●
7000 Series	●	●			●	●	●		●	●	●	●	●	●	●	●
AlphaEclipse™	●	●	●	●	●	●		●	●	●	●	●	●	●		●
AlphaPremiere™	●	●			●	●	●	●	●	●	●	●	●			●
AlphaVision™ (Full Matrix)	●	●			●	●	●		●	●	●	●	●	●	●	●
AlphaVision™ (Character Matrix)	●	●			●	●	●		●	●	●	●				●
Big Dot®	●	●			●	●		●	●	●	●	●	●			●
BetaBrite®	●	●			●	●		●	●	●	●		●			●
Director™	●	●			●	●			●	●	●	●				●
PPD™	●	●			●	●		●	●	●	●		●			●
Solar™	●	●	●	●	●	●		●	●	●	●	●	●			●

NOTE: 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.

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

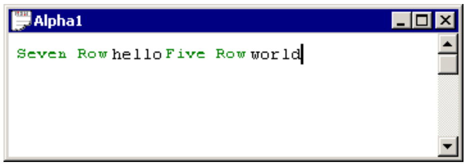
RULE:

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.

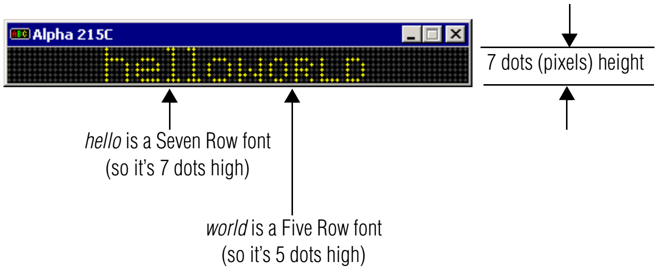
This is how a message is created in AlphaNET software:



Alpha1

Seven Row hello Five Row world

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



Alpha 215C

helloWORLD

hello is a Seven Row font (so it's 7 dots high)

world is a Five Row font (so it's 5 dots high)

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 recieved that is taller than what a one-line sign can display, then only the top 7 rows will be displayed.
- If a graphic is recieved that is longer than whata 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 treaded 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 thst 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 recieved that is taller than what a two-line sign can display, then only the top16 rows will be displayed.
- If a graphic is recieved 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 frm row 16: 5-high characters will use rows 12 to 16 and 7-high characters will use rows 10 to 16.

EXCEPTION CONDITIONS:

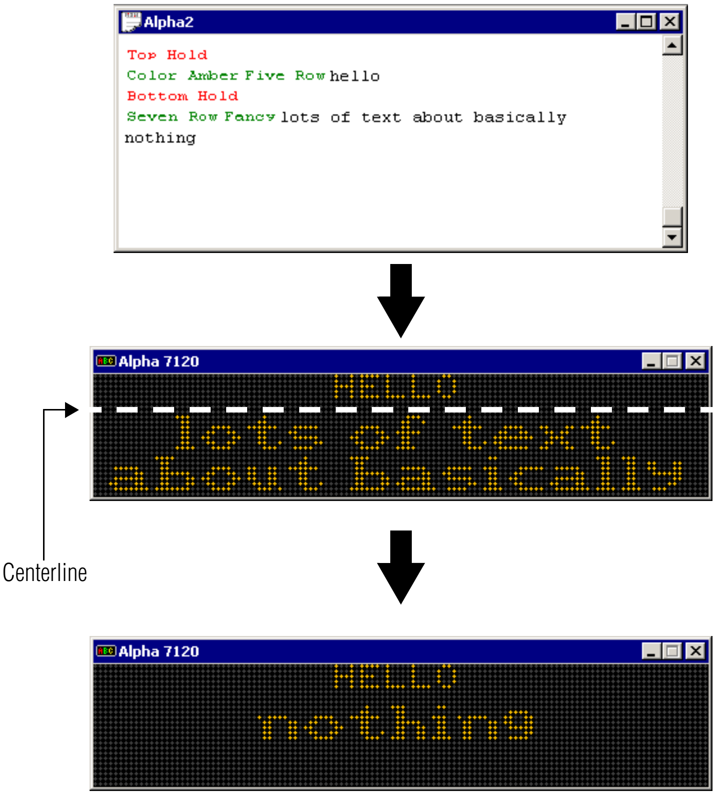
- If, when using the Top, Bottom, or Fill position, a sign receives a font that is larger than 7-high, then the sign will "size down" or reduce the font size. For example, 15 high fancy characters would be replaced by 7 high fancy characters.
- If a graphic is recieved that is taller than 7 rows high (15 high for Middle position), then only the top 7 (top 15 for Middle position) rows will be displayed.
- If a graphic is recieved 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).



```
Top Hold
Color Amber Five Row hello
Bottom Hold
Seven Row Fancy lots of text about basically
nothing
```

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 comand 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 thta 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 recieved that is taller than what the sign can display, then only the top most rows will be displayed.
- If a graphic is recieved 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 charcters ets 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 doesn 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 recieved that is taller than 7 rows high, then only the top 7 rows will be displayed.
- If a graphic is recieved that is longer than what the sign can display, then only the leftmost columns will be displayed.
- If a graphic is recieved that is smaller than 7 dots high, the graphic will be dipslayed 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

The Alpha® 2.0 protocol adds a number of features to the existing Alpha® 1.0 protocol:

- support for true time and date message scheduling
- programmable Hold mode times, ranging from 0.1 seconds to almost 7 minutes
- programmable character set table
- additional standard character sets
- time of day daylight savings mode control
- supports message delivery confirmation
- way of enabling a timeout message (no communication after a certain time puts up a message)

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

7.11.1 Speed control (0FH)

This control code 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 65: 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 66: 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

The following are two new options for the Write SPECIAL FUNCTION command code Generate Speaker Tone (page 21):

NOTE: A clear memory command will delete all sound files.

7.11.2.1 Store a programmable sound (33H)

Table 67: 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 21). 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 represnts the musical note. Valid entries are “A” through “G”. Each sound file can have up to 32 notes. A = one ASCII hexadecimal character that represnts the accidental. Valid entries are: “N” for Natural, “S” for sharp, and “F” fro flat. V = one ASCII hexadecimal character that represnts the sound volume. Valid entries are “0” through “F”. R = one ASCII hexadecimal character that represnts the number of times to repeat the musical note. Valid entries are from “0” through “F”. D = one ASCII hexadecimal character that represnts 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 represnts the pause or off time duration in 0.1 second increments. Valid entries are from “0” through “F”.

Example:	“3$4CF152EF152GF152” means:
	sound file label = “$”
	octave = “4”
	note = “C”
	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)
	note = “E”
	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)
	note = “G”
	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 (34H)

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 68: Trigger a programmable sound syntax

Syntax:	CL where: C = "4" (34H) follows the Generate Speaker Tone SPECIAL FUNCTION label: "(" 28H (see page 21). 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 (3AH)

The Set Run File Time SPECIAL FUNCTION allows setting a sart 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 withing the specified start and stop peroid, 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 the 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 thath 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 20).

Table 69: Set Run Time syntax

Syntax:	FDDMMYYYYTTTTEENNZZZZUUUU where: 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).
Example:	"A050120021300050120021830" means: Start running File Label "A" on January 5, 2002 ("05012002") at 1:00 pm ("1300"). Stop running the file on Janury 1, 2002 ("05012002") at 6:30 pm ("1830").

7.11.3.1 Removing Run File Time(s)

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 Priorty 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

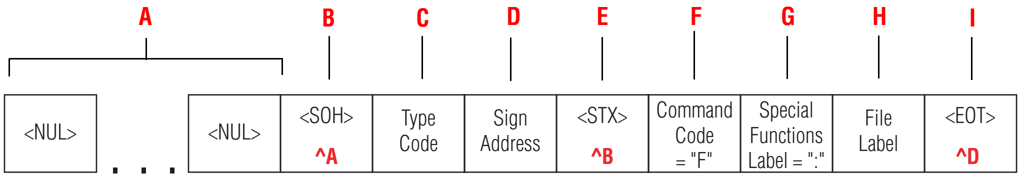
To delete all start and end times for a file, use the Set Run Time syntax (**Table 69** on page 99), except set all parameters to “9”. For example, to delete all Run Time entries for file “D” use: “D999999999999999999999999”.

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 retreiving start and end entries:

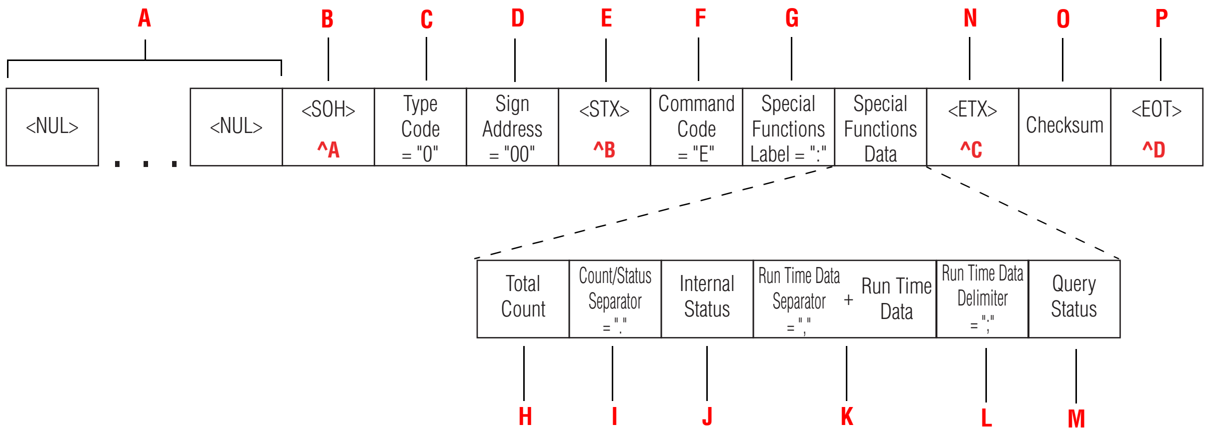
Table 70: Read Run File Time(s) file transmission frame format



Item	Name	Description
A	<NUL>	Twenty <NUL> (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	See Table 4, “Standard transmission frame (“1-byte” or “^A”) format,” on page 9.
D	Sign Address	See Table 4, “Standard transmission frame (“1-byte” or “^A”) format,” on page 9.
E	<STX>	<STX> (02H) character
F	Command Code	“F” = Read SPECIAL FUNCTIONs Command Code.
G	Special Functions Label	“:” = Read Run File Times code
H	File Label	The Run File to read.
I	<EOT>	<EOT> (04H) character

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

Table 71: Read Run File Time file response frame 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	“F” is returned by the sign. (The Read SPECIAL FUNCTIONs Command Code.)
G	Special Functions Label	“:” is returned by the sign. (The Run File Time code)

Table 71: Read Run File Time file response frame format

H	Special Functions 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 RAM memory in a sign:

Table 72: 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 (Series 7000 and 9000 signs)
16 high	20H - C1H	2576 (Series 4000 and 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 75):

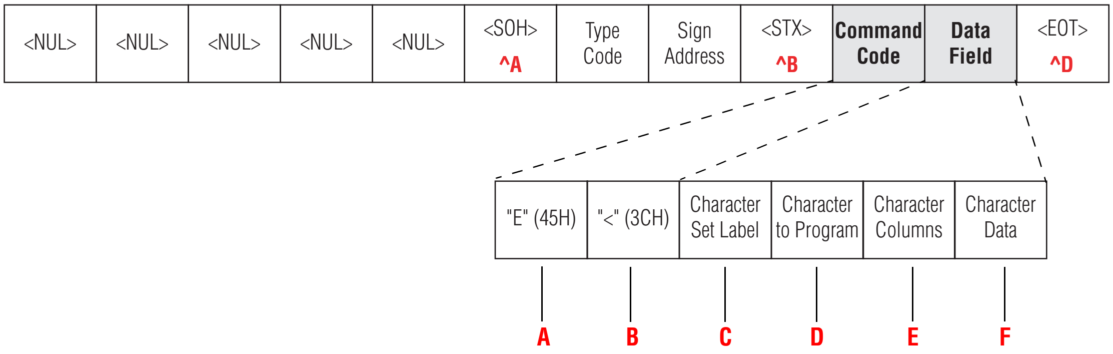
- 1AH + “W” = Five high custom character set
- 1AH + “X” = Seven/Eight high custom character set
- 1AH + “Y” = Ten high custom character set
- 1AH + “Z” = Fifteen/Sixteen high custom character set

7.11.4.3 Program Custom Character Sets (3CH)

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

Table 73: Program Custom Character Sets syntax

Standard transmission frame (see "Standard transmission frame ("1-byte" or "^A") format" on page 9):



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

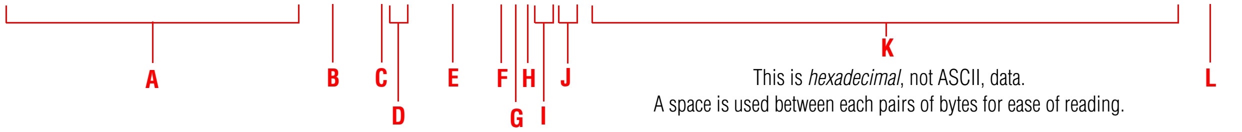
This example shows how to create a single 10 high custom character — a Smily Face:

	<- Parity Bit (not used)								<- Parity Bit (not used)								
Bit values ->	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	
0001 1110 = 1EH ->	○	○	○	●	●	●	●	○	○	○	○	○	○	○	○	○	<- 00H = 0000 0000
0010 0001 = 21H ->	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	○	<- 00H = 0000 0000
0101 0010 = 52H ->	○	●	○	●	○	○	●	○	○	●	○	○	○	○	○	○	<- 40H = 0100 0000
0101 0010 = 52H ->	○	●	○	●	○	○	●	○	○	●	○	○	○	○	○	○	<- 40H = 0100 0000
0100 0000 = 40H ->	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	<- 40H = 0100 0000
0100 0000 = 40H ->	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	<- 40H = 0100 0000
0101 0010 = 52H ->	○	●	○	●	○	○	●	○	○	●	○	○	○	○	○	○	<- 40H = 0100 0000
0100 1100 = 4CH ->	○	●	○	○	●	●	○	○	○	●	○	○	○	○	○	○	<- 40H = 0100 0000
0010 0001 = 21H ->	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	○	<- 00H = 0000 0000
0001 1110 = 1EH ->	○	○	○	●	●	●	●	○	○	○	○	○	○	○	○	○	<- 00H = 0000 0000

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 define the first character row above. The 8th bit in both bytes is not used and is always 0.

Table 74: 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>
```



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”	This means that this transmission is directed to all signs.
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 example” on page 63.
G	Data Field	Special Functions Label	“<”	Program Custom Character Set command
H		Character Set Label	“Y”	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 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) hexidecimal bitmapped representation of character row 10 (bottom)
L	<EOT>		04H	End Of Transmission character

7.11.5 Enable/Disable Daylight Saving Time (3DH)

This SPECIAL FUNCTION command ("=") enables or disables auto daylight savings time.

NOTE: This command is not implemented in the AlphaEclipse™ and AlphaPremiere™ sign firmware.

Table 75: Enable/Disable Daylight Savings Time syntax

Syntax:	N where: "0" = no auto Daylight Savings Time "1" = auto Daylight Savings Time enabled

7.11.6 Set AutoMode Table (3EF)

This SPECIAL FUNCTION command (">") is used to create a custom AutoMode table.

The modes programmed in the Automode table are used to run messages with Automode as their programmed mode. If the Automode table is cleared or not programmed, the default Automode table modes are used.

Table 76: Set Automode Table syntax

Start of packet				**Address specifier**			STX	**Command**		**Mode**		EOT
Syntax	NULL	...	SOH	Type Code	Address 1	Address 2		E	>	b1	b2	
Where "E " is the write special function and ">" is the special function for Set Automode Table.												

Two characters, b1 and b2, must be used, otherwise the command will be ignored. These characters specify the modes. The b1 is the mode code or the special mode specifier, and b2 is the special mode or a place holder "0" 30H. The data supplied can be from 1 to 15 modes, where each of the 15 entries is a standard or special mode (see Table 59, "Standard Modes," on page 87). If more than 15 modes are used, the command is ignored. If no modes are specified, the table is cleared.

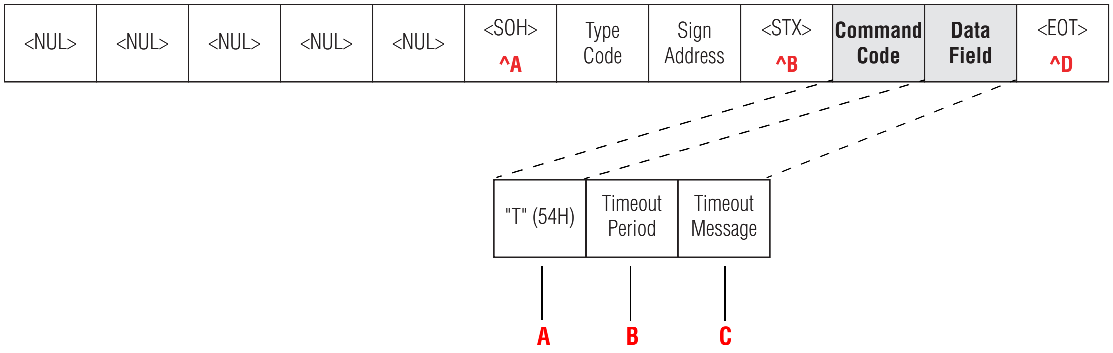
For example, to program Rotate, Hold, Flash, and Slide modes in the Automode table, send [NULL]...[SOH]Z00[STX]E>a0b0c0n5[EOT]. To clear the Automode table, send [NULL]...[SOH]Z00[STX]E>[EOT].

7.11.7 Set Timeout Message (“T”)

This Command Code [see “Command Code” in Table 4, “Standard transmission frame (“1-byte” or “^A”) format,” on page 9] allows you to specify a timeout period after which a custom message will appear on the sign.

Table 77: Set Timeout Message syntax

Standard transmission frame (see “Standard transmission frame (“1-byte” or “^A”) format” on page 9):



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

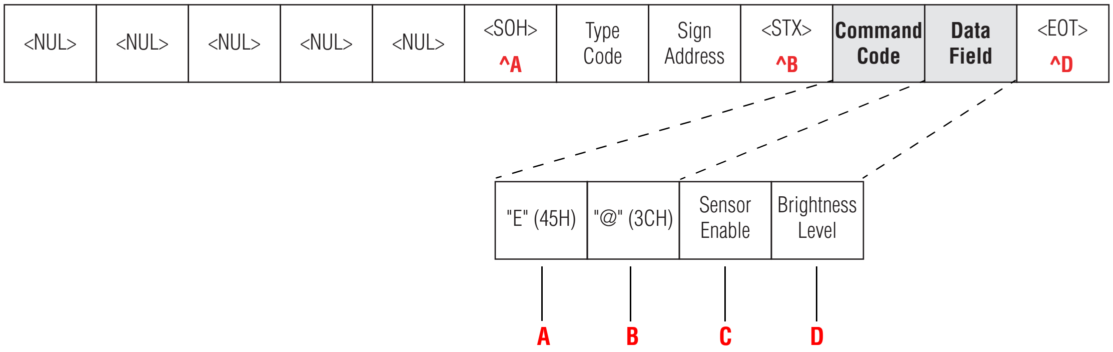
NOTE: The Clear Memory command will not delete a Timeout Message. To clear a Timeout Message, either (1) set the clear memory DIP switch and cycle power or (2) send a NULL message as follows: ^AZ00^BT000^D.

7.11.8 Set Dimming Control Register (“@”)

The Dimming Control Register controls the brightness percentage when a 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 22) dims to.

Table 78: Set Dimming Control Register syntax

Standard transmission frame (see “Standard transmission frame (“1-byte” or “^A”) format” on page 9):



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

7.11.9 Enable/Disable ACK/NAK Response (“s”)

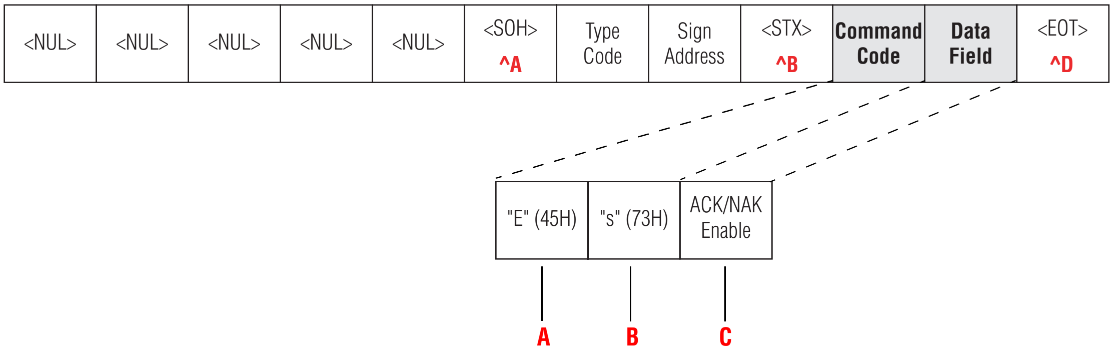
When the ACK/NAK response is enabled by using the “s’ Write SPECIAL FUNCTION command, then a sign will respond with one of the following transmissions:

- [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 29).

Table 79: Enable/Disable ACK/NAK syntax

Standard transmission frame (see “Standard transmission frame (“1-byte” or “^A”) format” on page 9):



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. Valid entries are: “0” 30H = disable ACK/NAK sign response **(default)** “1” 31H = enable ACK/NAK sign response

7.12 Appendix L: AlphaEclipse™ Protocol Addendum v1.05

The default serial address for the AlphaEclipse™ is "01" if the Set Unit command or the Set Serial Address command (E7) is not used to set it otherwise.

Where it appears in this section, "[NULL](5x)" should be read as "[NULL][NULL][NULL][NULL][NULL]". For an example, see "Read External Temperature Command (Read Register "T")" on page 111.

7.12.1 Temperature Logging

7.12.1.1 Reading Temperature

After the temperature is read, it is compared to the previous read and the maximum and minimum temperatures are stored. 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.

7.12.1.2 Daily Log

The board and external temperatures (minimum and maximum) are recorded every thirty minutes over the past 24 hours.

Table 80: Temperature Log

Item	Description
FL	Read Daily Log
EL	Response to Read Daily Log command

[CR][LF]DAILY LOG: Bm Bx Em Ex[CR][LF]	Daily log—48 entries recorded every half-hour from the previous half-hour.
	Bm = Board minimum temperature
	Bx = Board maximum temperature
	Em = External minimum temperature
	Ex = External maximum temperature

AAAAC AAAAC AAAAF AAAAF[CR][LF]	AAAA - (+/-)000 (plus or minus sign and a three digit temperature)
6 MONTH LOG: MM-DD Bm Bx Em Ex[CR][LF]	Yearly log—records the board and external temperatures (minimum and maximum) for the previous 178 days. MM—Month DD—Day
MM-DD AAAAC AAAAC AAAAF AAAAF[CR][LF]	MM—Month DD—Day
ERROR LOG: MM-DD HH:MM ER Bm Bx Em Ex[CR][LF]	Error Log—an event-driven log that records the last 48 errors caused only by dimming and shutdown. It records date, time, temperature board (minimum and maximum), external temperature (minimum and maximum), and error. HH—Hour MM—Minute ER—Errors flag • 1—Display caused overheat mode • 2—Controller temperature caused overheat mode • 3—External temperature caused overheat mode • 4—Display caused dimming mode • 5—Controller caused dimming mode • 6—External temperature caused dimming mode
MM-DD HH:MM ER AAAAC AAAAC AAAAF AAAAF[CR][LF]	HH—Hour MM—Minute

7.12.1.3 Read temperature Log Command "L"

The read temperature log register is "L" [4C]. [SOH]Z00[STX]FL[EOT]

An example of the sign’s response:

- **[SOH]Z00[STX]EL[CR][LF] DAILY LOG: Bm Bx Em Ex[CR][LF]+027 C +029C +070F +075F[CR][LF]** (repeats for next 47 logs)
- **6 MONTH LOG: MM-DD Bm Bx Em Ex[CR][LF]01-02 +027 C +029C +070F +075F[CR][LF]** (repeats for next 177 logs)
- **ERROR LOG: MM-DD HH:MM ER Bm Bx Em Ex[CR][LF]01-02 11:59 1 +020 C +04AC +070F +079F[CR][LF]** (repeats for 48 logs)
- **[ETX]XXXX[EOT]** (where XXXX is the checksum of the Alpha packet)

Note that “–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. See the Set Unit command for further details.

7.12.2 Read External Temperature Command (Read Register “T”)

“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.

For example, [NULL](5x) [SOH]Z00[STX]FT[EOT] will return [NULL](5x) [SOH]000[STX]+075[ETX]xxxx[EOT], where ‘xxxx’ equals the packet checksum for a fahrenheit temperature of 75 degrees.

7.12.3 Read Internal Temperature Command (Read Register "F")

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

For example, [NULL](5x) [SOH]Z00[STX]FTI[EOT]. The unit sends back for 20 C, which reads [NULL](5x) [SOH]000[STX]ET+020C[ETX]xxxx[EOT], where ‘xxxx’ equals the packet checksum.

7.12.4 Unit Command Register Write

This command is used to serially configure the sign. Once the display receives this command and the packet is formatted properly, the sign will reset and cycle through the power-up messages. The dipswitch settings will be ignored. Changes to the configuration of the sign can then only be made through this command, unless the “UN” command is sent. The “UN” command is used to return the display to the dipswitch settings. Also, when any one of the set unit commands is sent, the sign resets and the first message to appear in the power up messages is “Dip Disabled,” indicating that a set unit command has been sent and the dipswitches are disabled.

Sending an Alpha protocol clear memory command (“E$”), soft reset command (“E,”), or updating the firmware will have no affect on the configuration settings. The only way to alter the serial configuration settings is to change them by sending this command or the “UN” command.

Table 81: Write/Read Unit Command Register

Command	Description
Set Unit Size	Sets the unit column and rows. • U1—Write register set unit size • AAAA—(HEX) Column size of the sign (0010-00F0 columns) • BBBB—(HEX) Row size of the sign (0008-0080 rows)
Set Run Mode	Sets the unit run mode. • U2—Write register set unit run mode. Up to 255 tests. • CC—(HEX) • 00—Run normally (default) • 01—Test 1 (production test mode) • 02—Test 2 (test pattern) • 03—Test 3 (test match mode) • 04—Test 4 (temperature test mode) • 05–FF—Future use
Set Serial Address	Sets the unit serial address. • U3—Write register set unit serial address • DD—Serial address HEX (00–FF)
Set Serial Data	Sets the unit serial baud rate and data format. • U4—Write register set unit serial data. *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.* • EE—Baud rate (HEX) • 00—Autobaud from 38400 (8N1/7E2) • 01—1200 (8N1) • 02—1200 (7E2) • 03—2400 (8N1) • 04—2400 (7E2) • 05—4800 (8N1) • 06—4800 (7E2) • 07—9600 (8N1) • 08—9600 (7E2) • 09—19200 (8N1) • 0A—19200 (7E2) • 0B—38400 (8N1) • 0C—38400 (7E2) When a sign is configured for autobaud, every packet sent to the display must be preceded by at least five nulls [NULL] or five [SOH] characters in order for the firmware to be able to decipher the baud rate of the transmission.
Set Unit Configuration	Sets the unit configurations. • U5—Write register set unit configuration. • F—Clear memory flag • 0—Do not clear memory on power-up • 1—Clear memory on power-up (simulates a virgin power-up) • G—Master/Slave flag • 0—Master • 1—Slave • H—Demo message flag (not applicable for the AlphaEclipse™, but must be used as a place holder) • 0—Off • 1—On • I—Color flag (not applicable for the AlphaEclipse™, but must be used as a place holder) • 0—Mono • 1—Color unit • J—IR flag (not applicable for the AlphaEclipse™, but must be used as a place holder) • 0—IR off • 1—IR on • K—RS485 echo flag (not applicable for the AlphaEclipse™, but must be used as a place holder) • 0—Off • 1—On • L—Driver height • 0–8 High • 1–16 High • ZZZZZZZZZZ—For future use. Send “0000000000” 30H if not used. (not applicable for the AlphaEclipse™, but must be used as a place holder)

All write set unit commands can be combined into a packet in any combination.

For example, [SOH]Z00[STX]EU100800080U201U500000001111111111[EOT].

When the clear memory flag is set, it simulates a virgin power-up. This is the same as when the clear memory dipswitch is set, meaning all programmed information in the following list will be cleared (this information is not cleared by the clear memory protocol command, “E$”):

- Automode table—“<” command
- Programmed custom characters—“<” command
- Serial Timeout message—“T” command
- Dimming control register—“@” command (this register is set to its default setting of 50% during a virgin power-up)
- Temperature log

7.12.5 Unit Command Register Read Only

Table 82: Read Unit Command Register

Command	Description
Read Unit Register	Reads DIP and memory size. • U6—Read unit register • Aabbccdd—Dip switch reading(HEX) • aa—Dip reading 1 • bb—Dip reading 2 • cc—Dip reading 3 • dd—DIP reading 4 • XXXX—Total RAM Kilobytes (HEX) • Example, 03E8—1 megabyte of RAM Note that the DIP readings above do not correspond to the actual DIP switches.

7.12.6 Unit Command Reset Unit to DIP switch

Sending this command will erase the configuration of the sign that was programmed with the commands above and allow for the dipswitches to be used to configure the sign.

Table 83: Write Unit Command Register

Command	Description
Write Unit Register	Reset unit to the DIP configuration. • UN—Resets the unit to the DIP default configuration

7.12.7 Read Dim Time Command (Read Register "F")

"/"—Sending "F/" will read the dim on and off times.

For example, [NULL](5x) [SOH]Z00[STX]F/[EOT]. The unit sends back “7824” for a dim on time (“78”) of 8:00 PM and a dim off time (“24”) of 6:00 AM, which reads [NULL](5x) [SOH]000[STX]E/7824[EOT].

For more information, see “Appendix B: Valid Start and Stop times” on page 44.

7.12.8 Read Dimming Control Register

"@"—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.

Table 84: Read Dimming Commands

Item	Description
F@	Read dimming control register command.
AAA	Current dimming control register setting. This setting is the percentage the display will dim to if the photocell causes dimming.
BBB	Current brightness level. If the display is dimming at the time this command is sent, the dimming level will be returned in this position. This dimming could be caused by options E, F, G, or the set dim time command ("/").
C	Photocell enabled/disabled flag. Enabling/disabling the photocell is controlled by the "E@" set dimming control register command. • 0—Disabled • 1—Enabled
D	Photocell causing dimming flag. • 0—Photocell is not currently causing dimming • 1—Photocell is currently causing dimming
E	Display load causing dimming flag. • 0—Display load is not currently causing dimming • 1—Display load is currently causing dimming
F	Display temperature, internal or external, causing dimming flag. • 0—Display temperature is not currently causing dimming • 1—Display temperature is currently causing dimming
G	Dimming time command, "E/", causing dimming flag. • 0—Dimming time command is not currently causing dimming • 1—Dimming time command is currently causing dimming

7.12.8.1 Example 1

[NULL](5x) [SOH]Z00[STX]F@[EOT]

The unit returns "E@05010010000" for a dimming level register setting of 50%, a brightness level of 100%, enabled photocell, and nothing causing dimming.

For example, [NULL](5x) [SOH]000[STX]E@05010010000[EOT]

7.12.8.2 Example 2

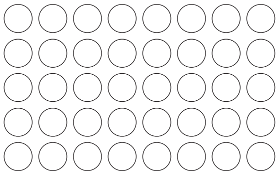
[NULL](5x) [SOH]Z00[STX]F@[EOT]

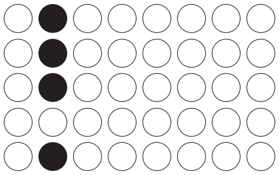
The unit returns "E@05005011000" for a brightness level register setting of 50%, a brightness level of 50%, enabled photocell, and photocell-caused dimming.

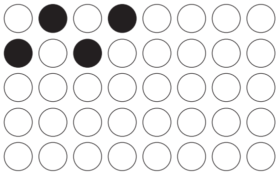
For example, [NULL](5x) [SOH]000[STX]E@05005011000[EOT]

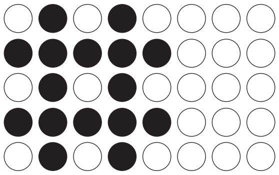
7.13 Appendix M: Font character sets

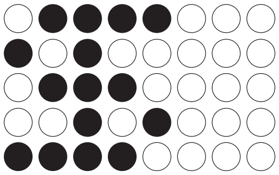
7.13.1 5-High Regular (SS5)

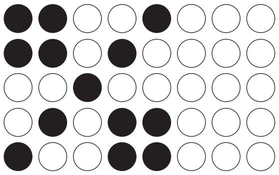

CHR032

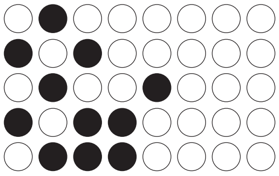

CHR033

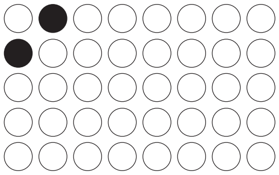

CHR034

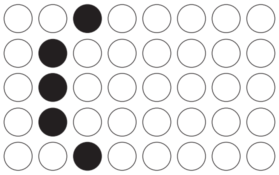

CHR035

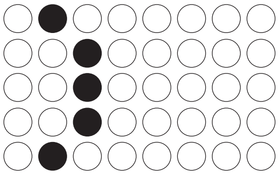

CHR036

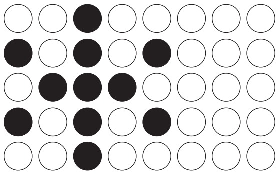

CHR037

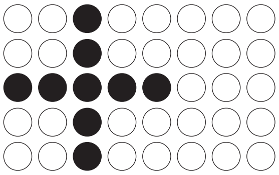

CHR038

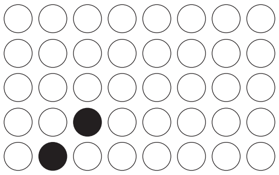

CHR039

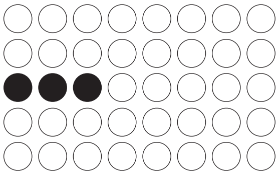

CHR040

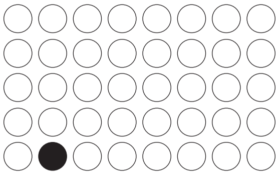

CHR041

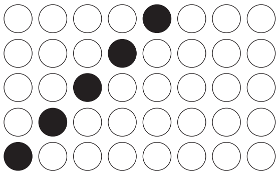

CHR042

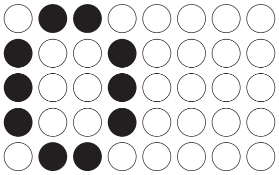

CHR043

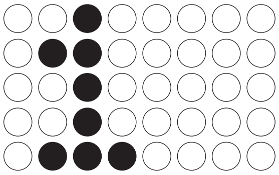

CHR044

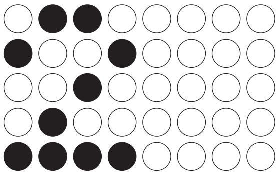

CHR045

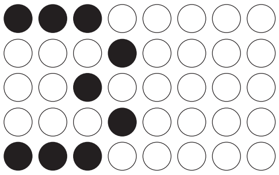

CHR046

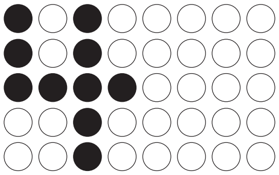

CHR047

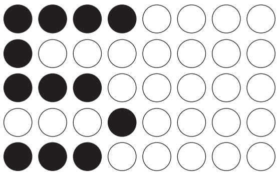

CHR048

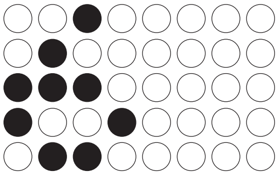

CHR049

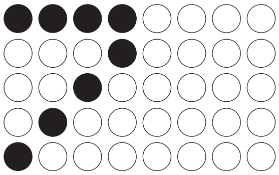

CHR050

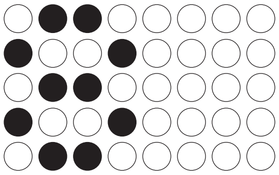

CHR051

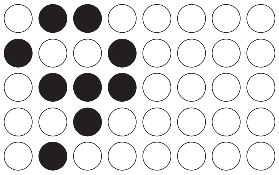

CHR052

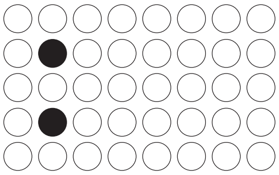

CHR053

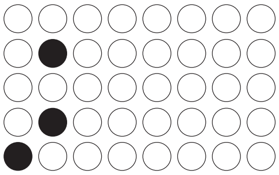

CHR054

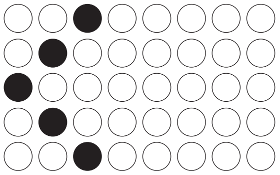

CHR055

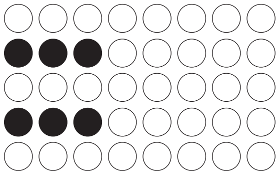

CHR056

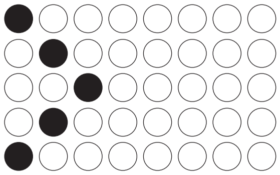

CHR057

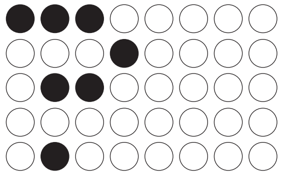

CHR058

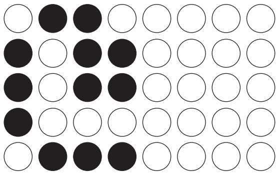

CHR059

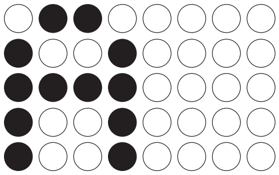

CHR060

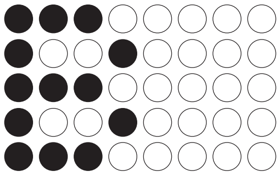

CHR061

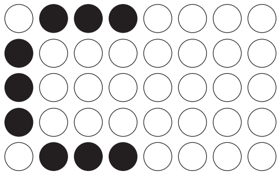

CHR062

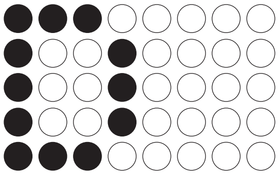

CHR063

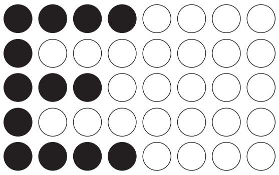

CHR064

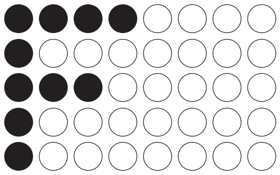

CHR065

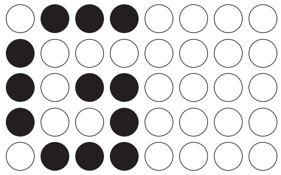

CHR066

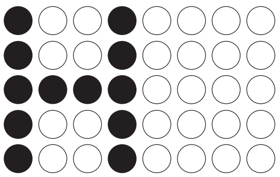

CHR067

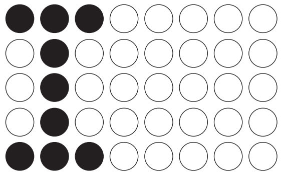

CHR068

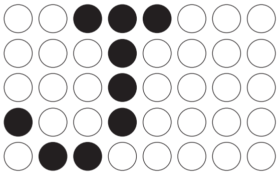

CHR069

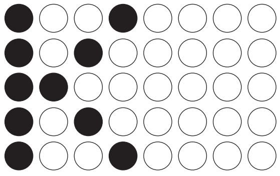

CHR070

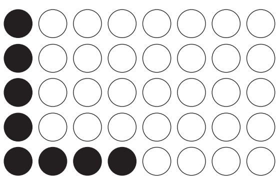

CHR071

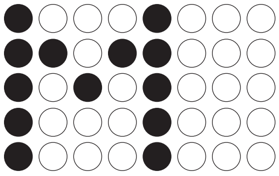

CHR072

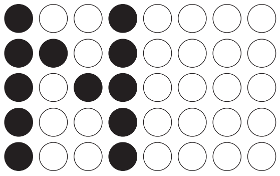

CHR073

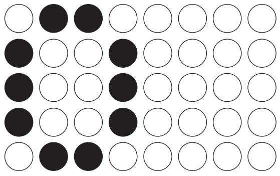

CHR074

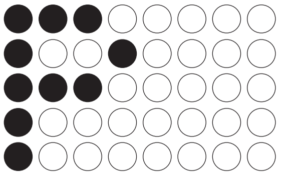

CHR075

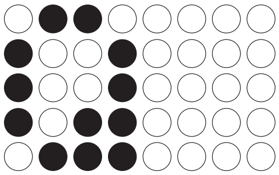

CHR076

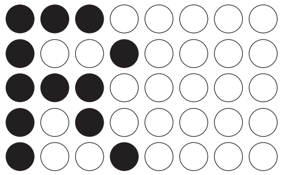

CHR077

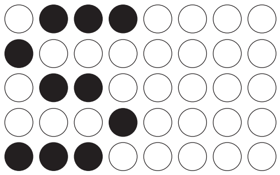

CHR078

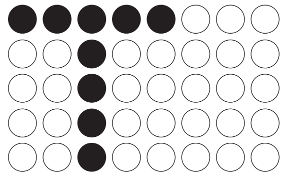

CHR079

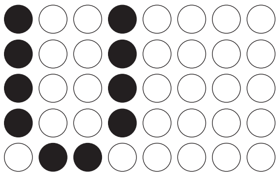

CHR080

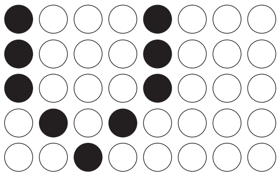

CHR081

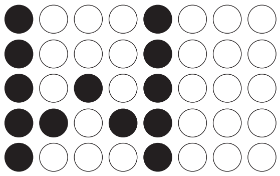

CHR082

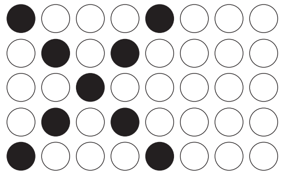

CHR083

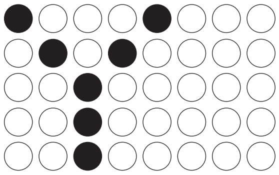

CHR084

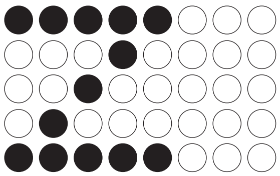

CHR085

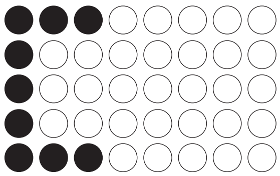

CHR086

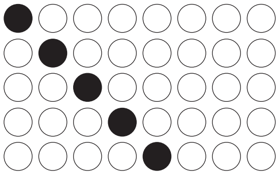

CHR087


CHR088

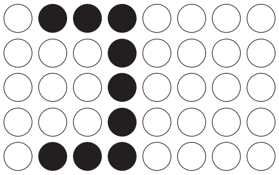

CHR089


CHR090

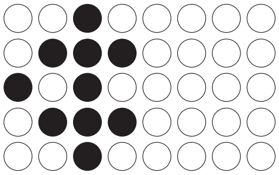

CHR091



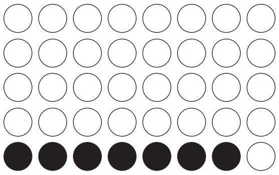
CHR092



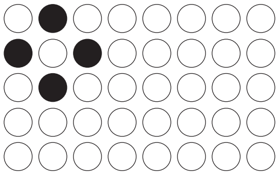
CHR093



CHR094

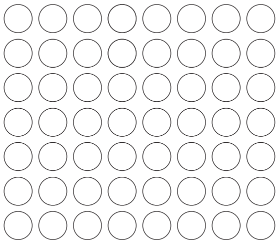


CHR095

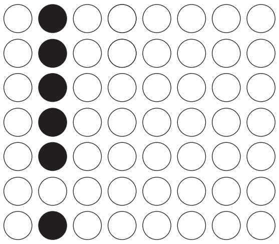


CHR096

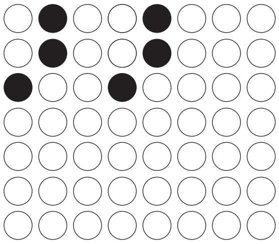
7.13.2 7-High Regular (SS7)



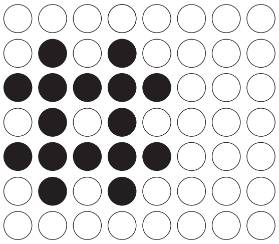
CHR032



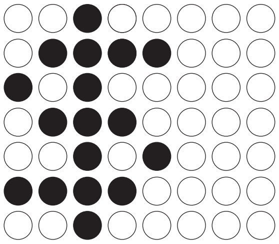
CHR033



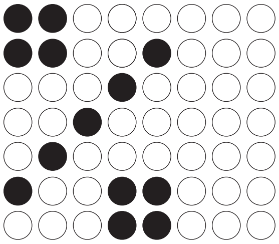
CHR034



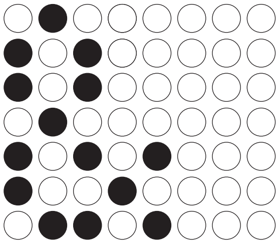
CHR035



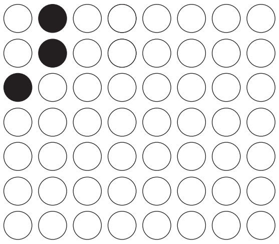
CHR036



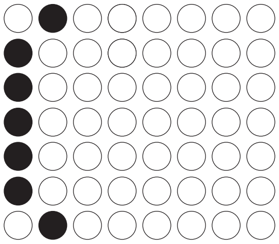
CHR037



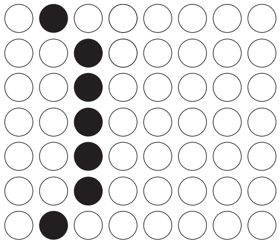
CHR038



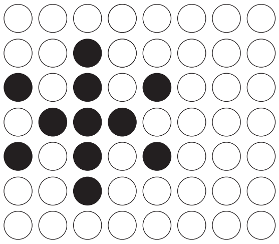
CHR039



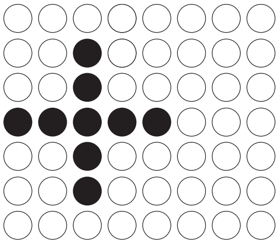
CHR040



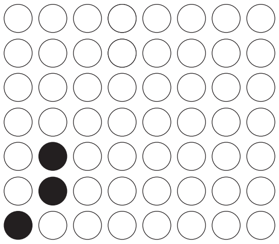
CHR041



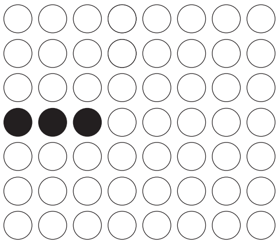
CHR042



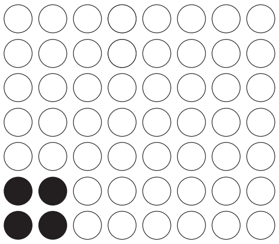
CHR043



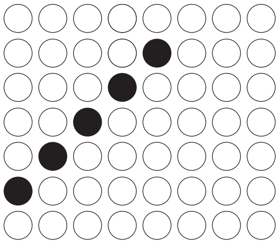
CHR044



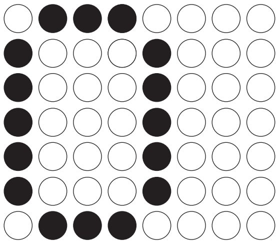
CHR045

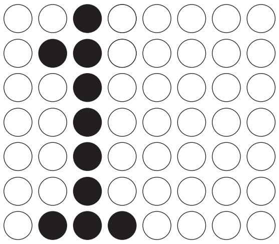


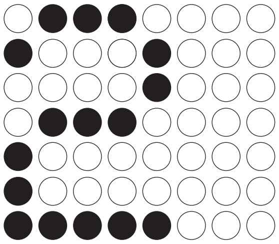
CHR046

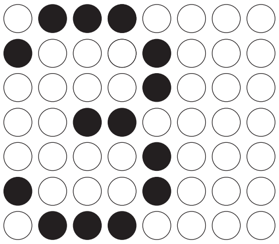


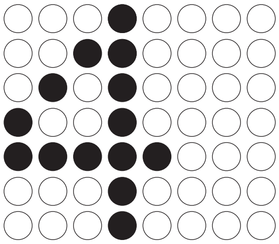
CHR047

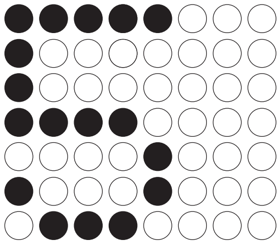

CHR048

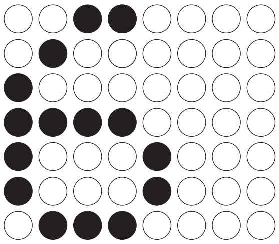

CHR049

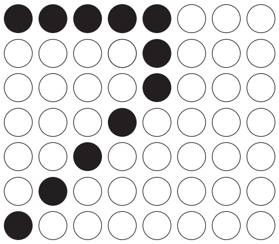

CHR050

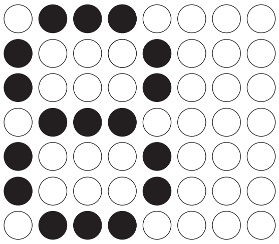

CHR051

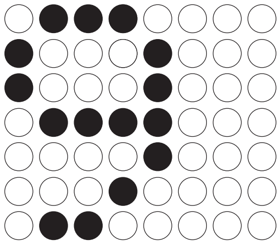

CHR052

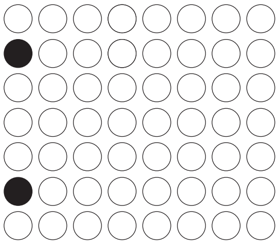

CHR053

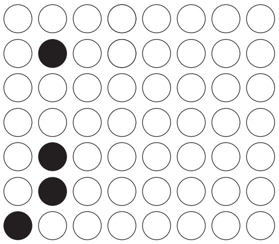

CHR054

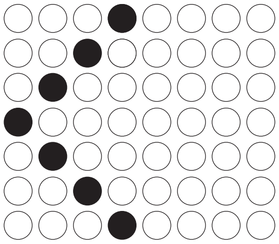

CHR055

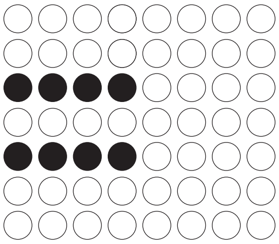

CHR056

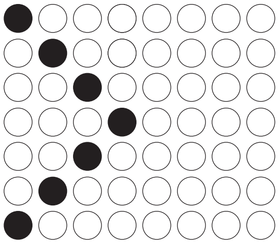

CHR057

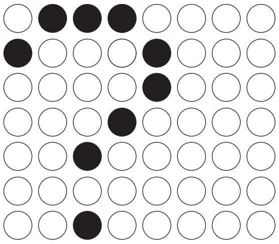

CHR058

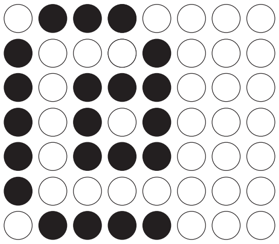

CHR059

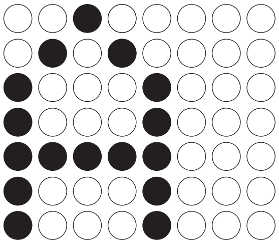

CHR060

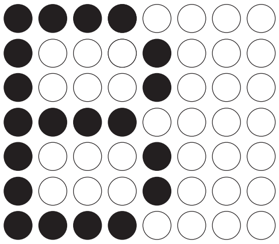

CHR061

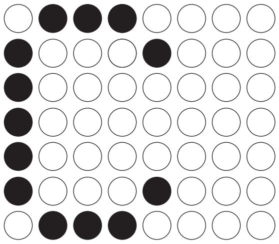

CHR062

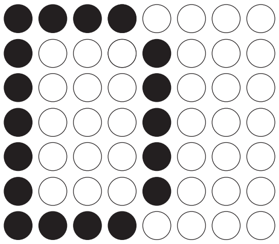

CHR063

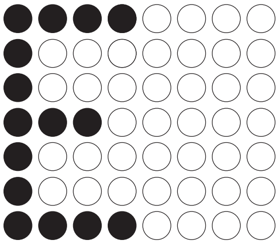

CHR064

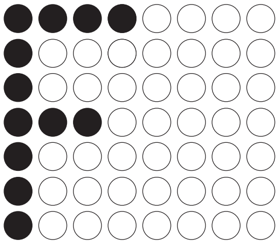

CHR065

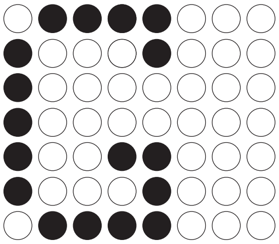

CHR066

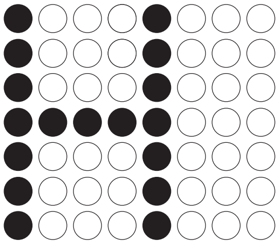

CHR067

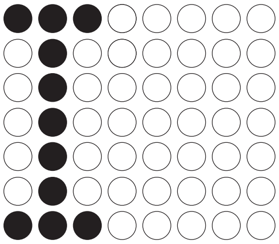

CHR068

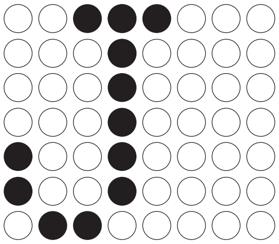

CHR069

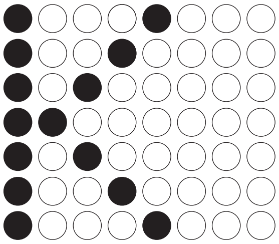

CHR070

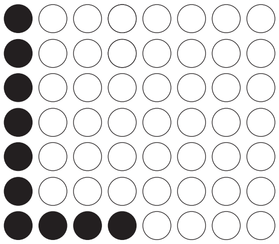

CHR071

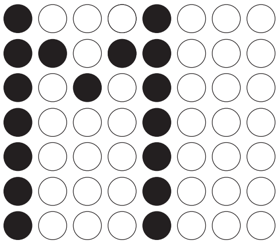

CHR072

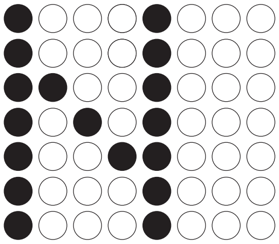

CHR073

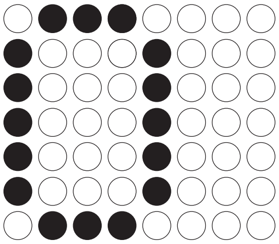

CHR074

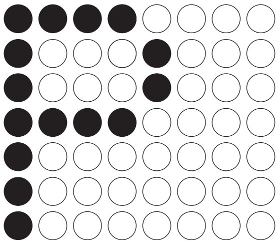

CHR075

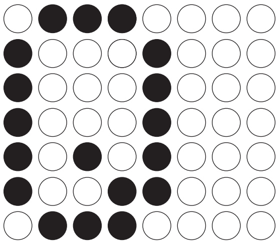

CHR076

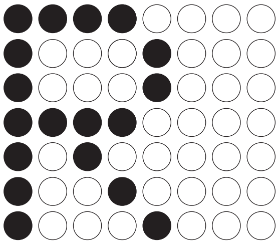

CHR077

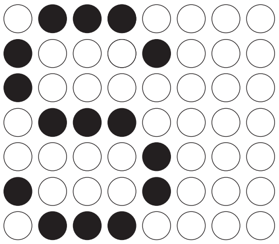

CHR078

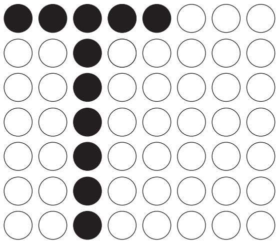

CHR079

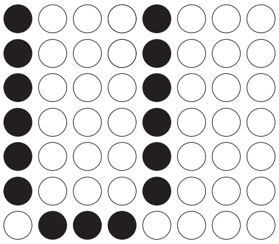

CHR080

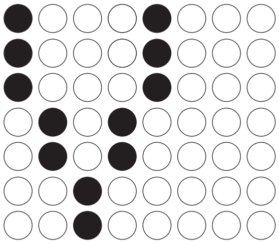

CHR081

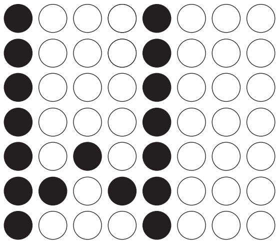

CHR082

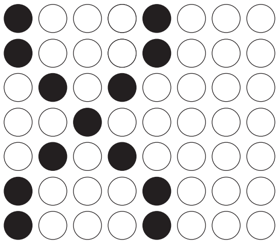

CHR083

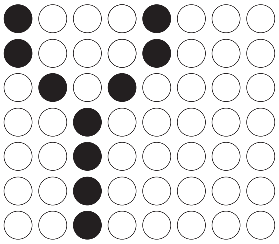

CHR084

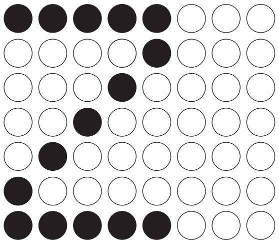

CHR085

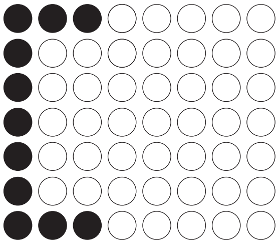

CHR086

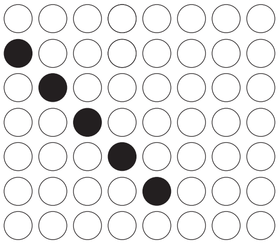

CHR087

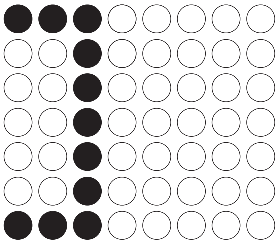

CHR088

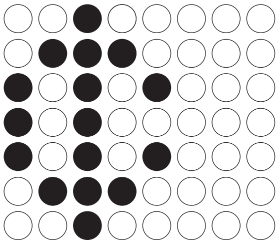

CHR089

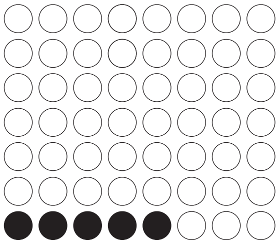

CHR090

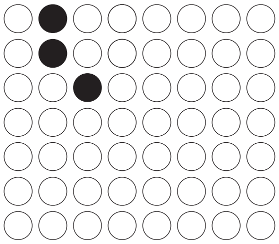

CHR091


CHR092

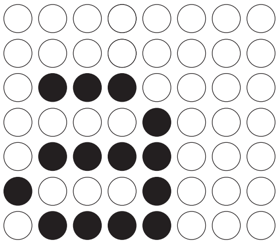

CHR093


CHR094

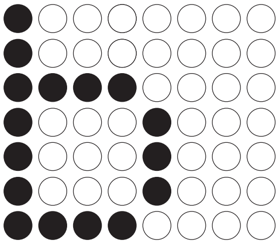

CHR095



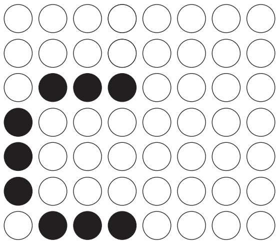
CHR096



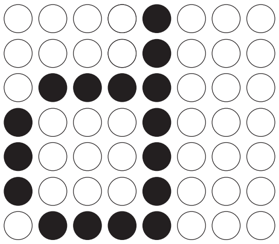
CHR097



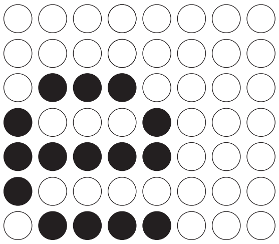
CHR098



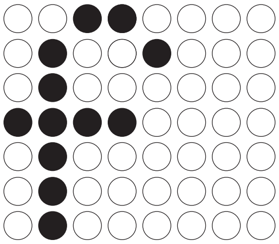
CHR099



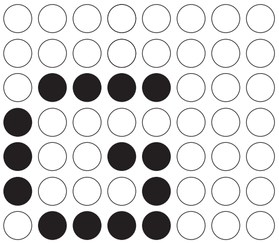
CHR100



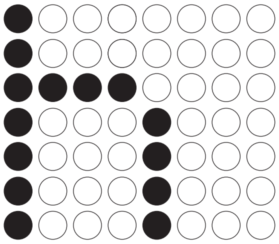
CHR101



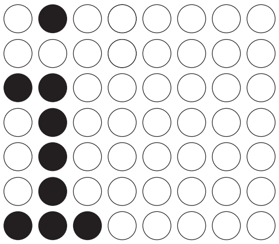
CHR102



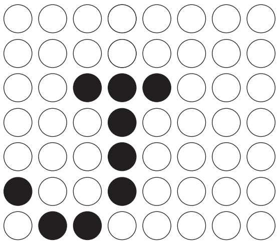
CHR103



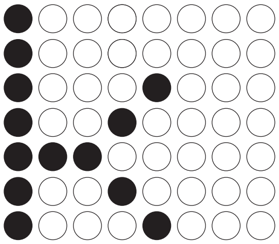
CHR104



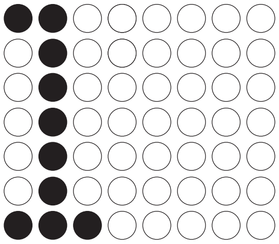
CHR105



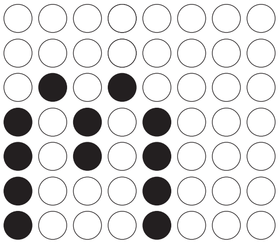
CHR106



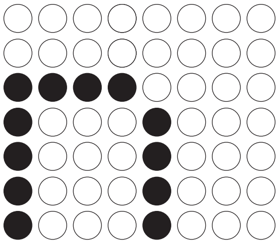
CHR107



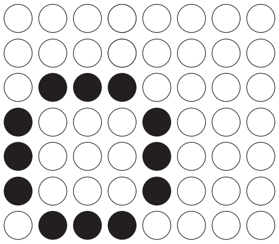
CHR108



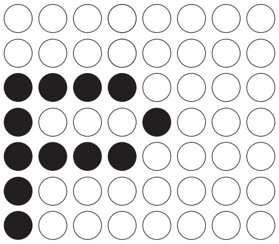
CHR109



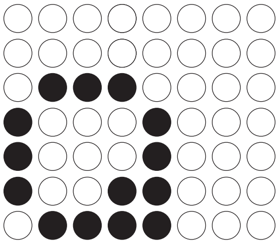
CHR110



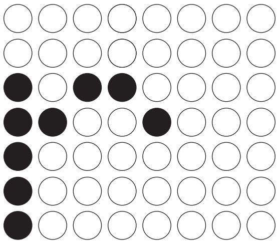
CHR111



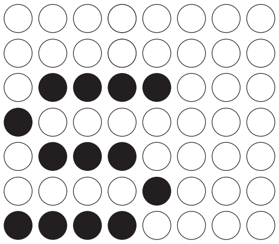
CHR112



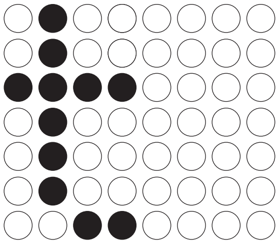
CHR113



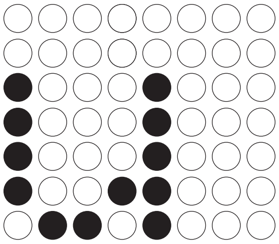
CHR114



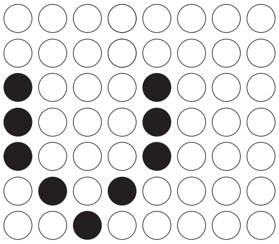
CHR115



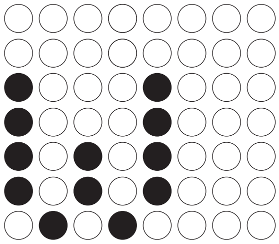
CHR116



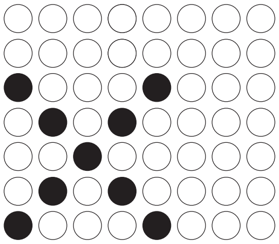
CHR117

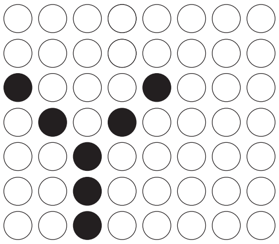


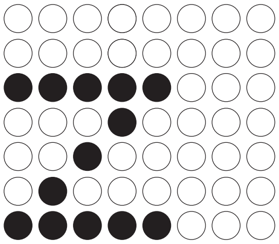
CHR118

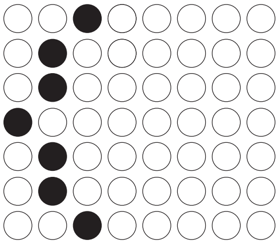


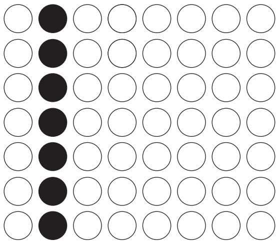
CHR119

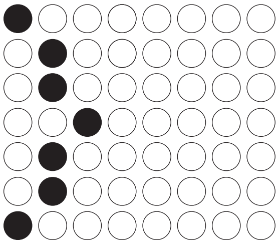

CHR120

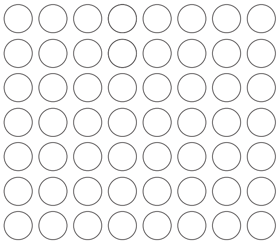

CHR121

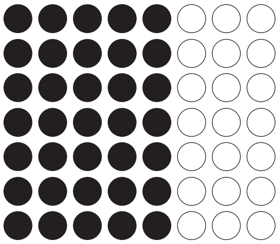

CHR122

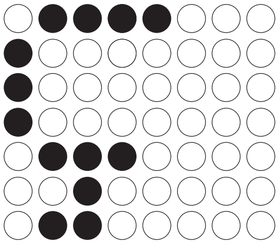

CHR123

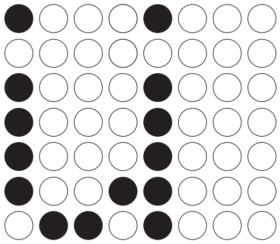

CHR124

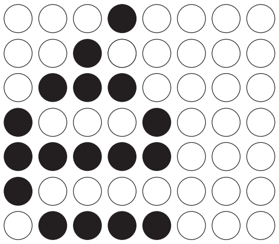

CHR125

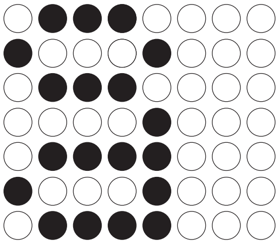

CHR126

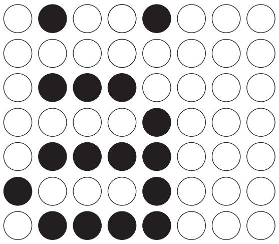

CHR127

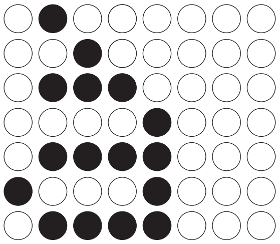

CHR128

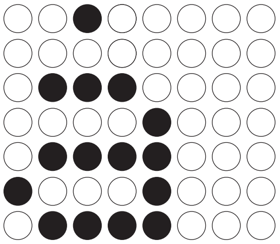

CHR129

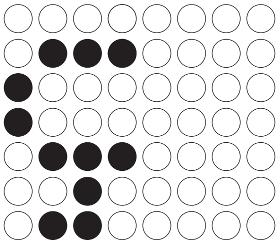

CHR130

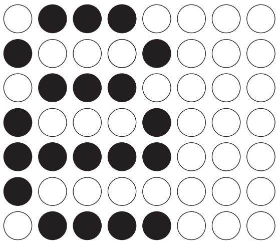

CHR131

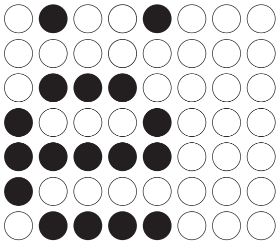

CHR132

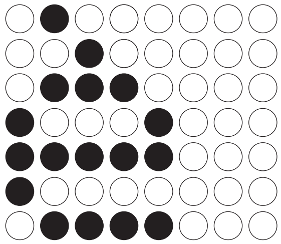

CHR133

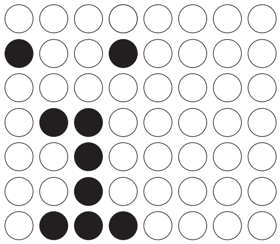

CHR134

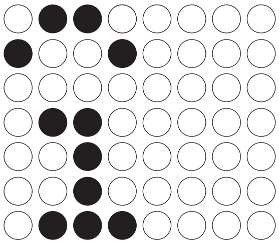

CHR135

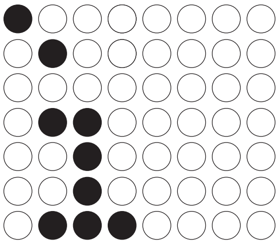

CHR136

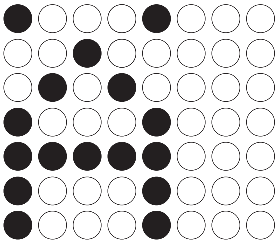

CHR137

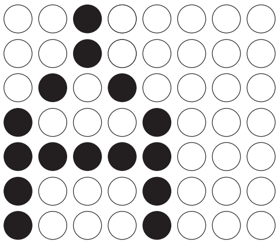

CHR138

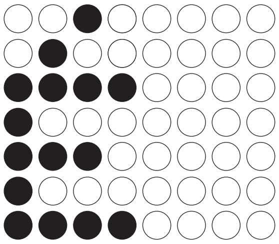

CHR139


CHR140

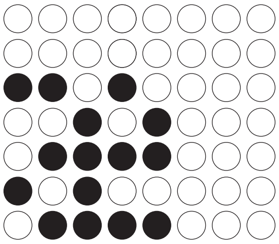

CHR141


CHR142

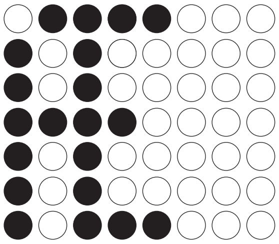

CHR143



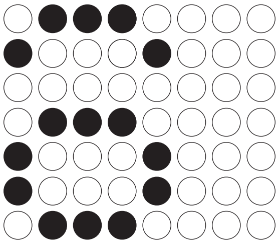
CHR144



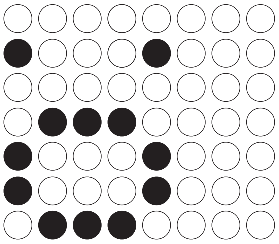
CHR145



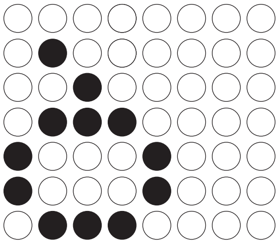
CHR146



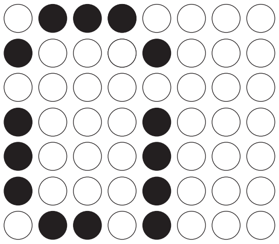
CHR147



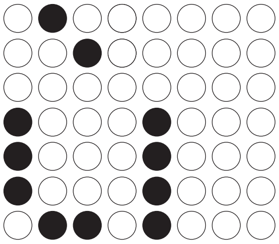
CHR148



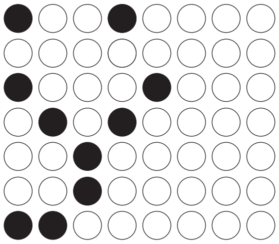
CHR149



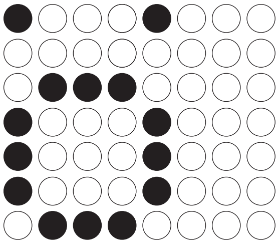
CHR150



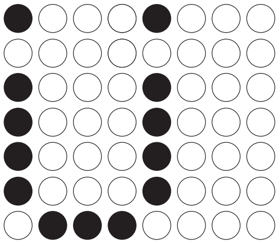
CHR151



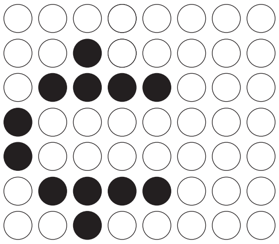
CHR152



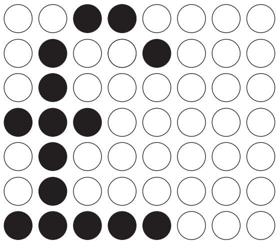
CHR153



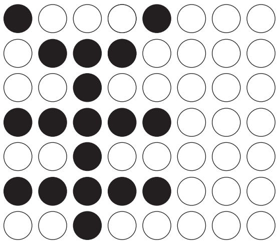
CHR154



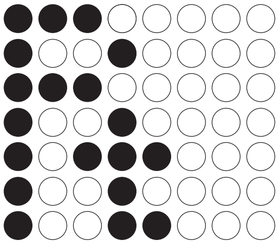
CHR155



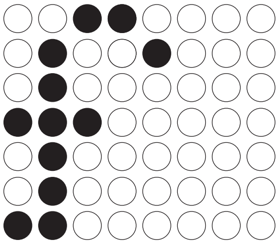
CHR156



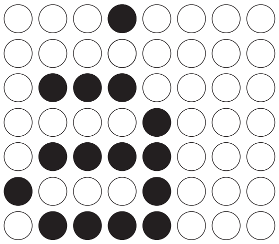
CHR157



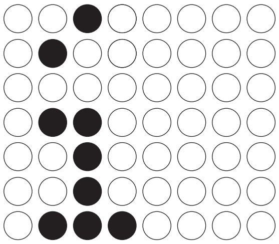
CHR158



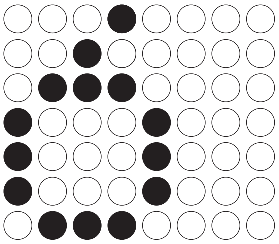
CHR159



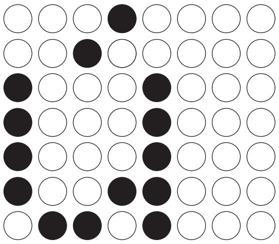
CHR160



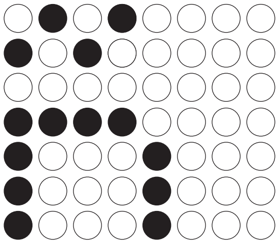
CHR161



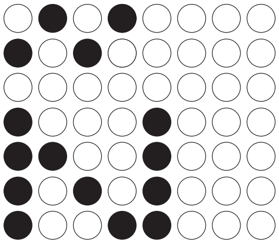
CHR162



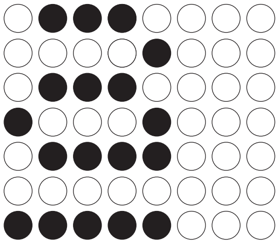
CHR163



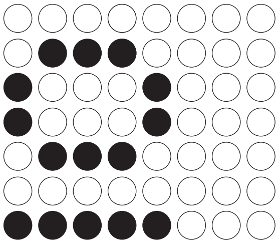
CHR164



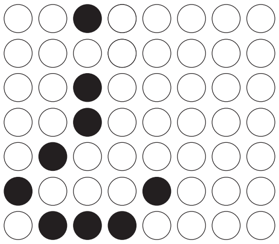
CHR165



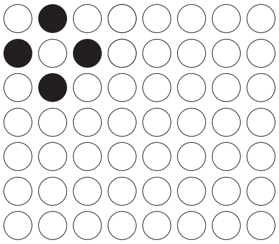
CHR166



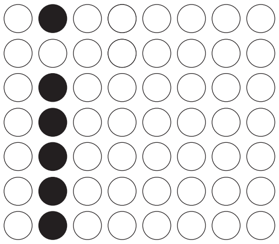
CHR167



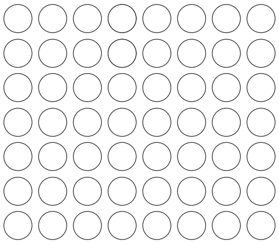
CHR168



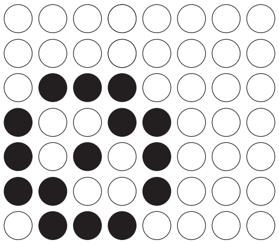
CHR169



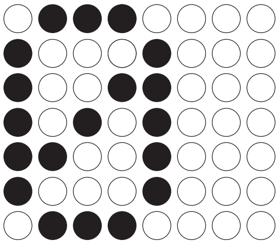
CHR170



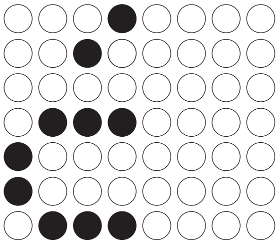
CHR171



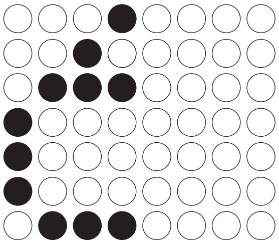
CHR172



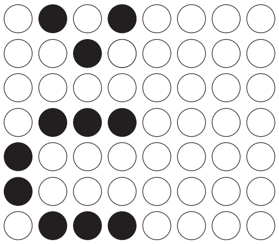
CHR173



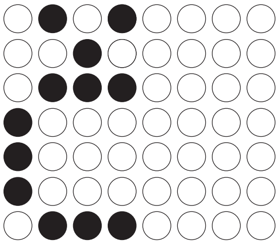
CHR174



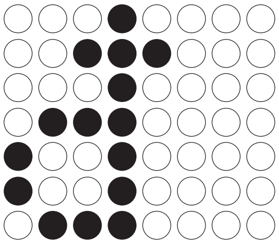
CHR175



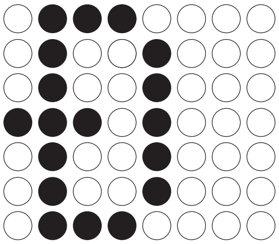
CHR176



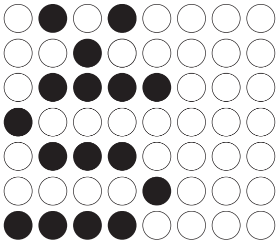
CHR177



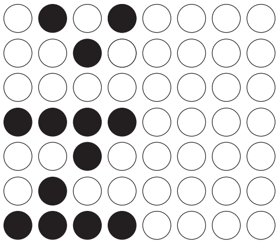
CHR178



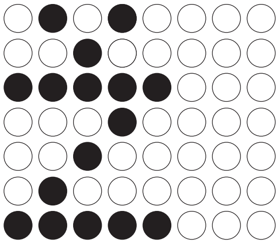
CHR179



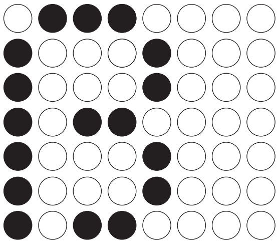
CHR180



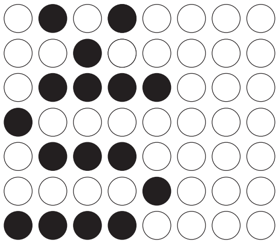
CHR181



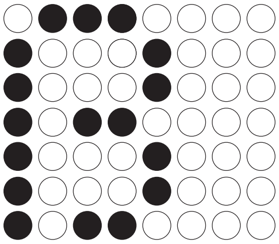
CHR182



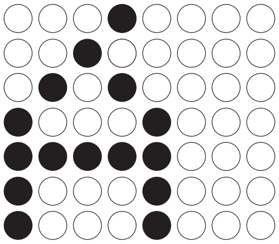
CHR183



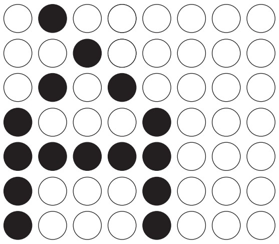
CHR184



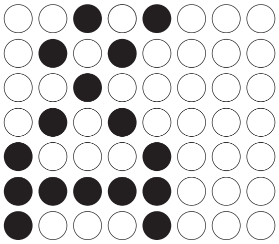
CHR185



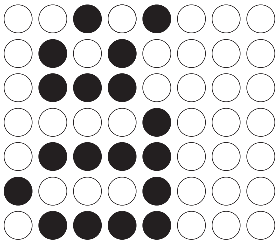
CHR186



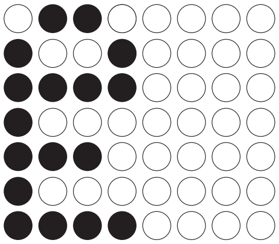
CHR187



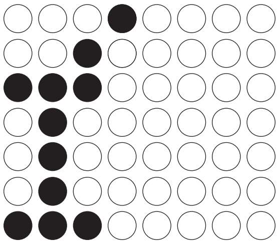
CHR188



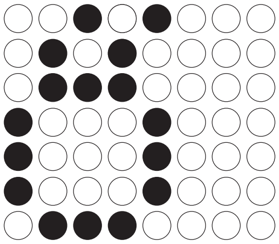
CHR189

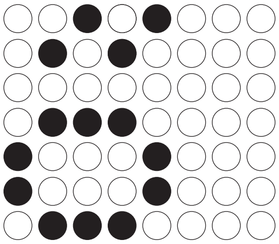


CHR190

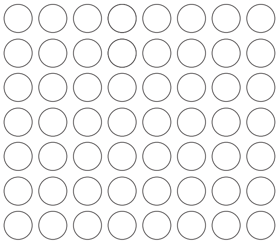


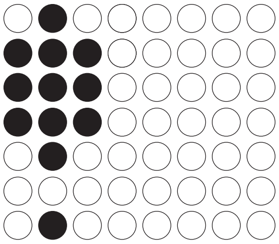
CHR191

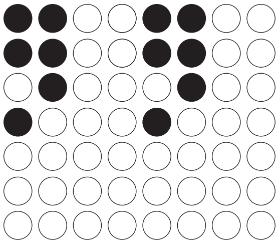

CHR192

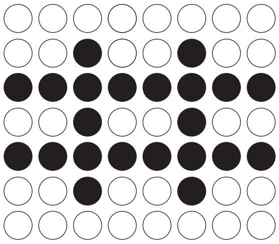

CHR193

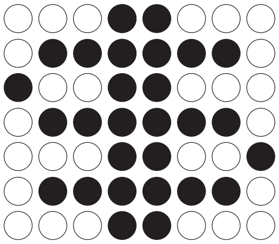
7.13.3 7-High Fancy (SF7)

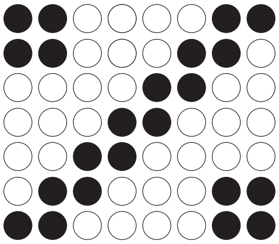

CHR032

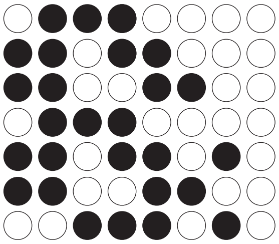

CHR033

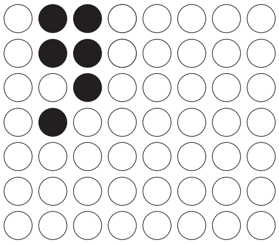

CHR034

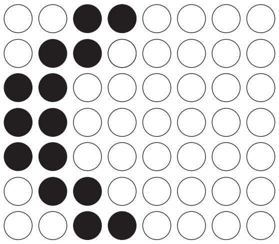

CHR035

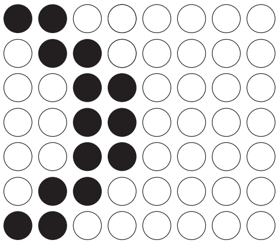

CHR036

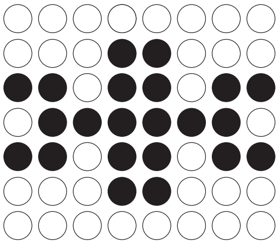

CHR037

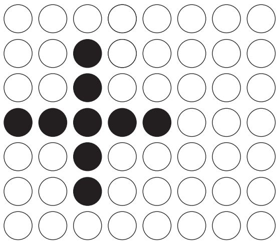

CHR038

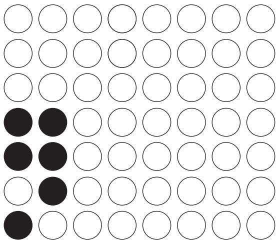

CHR039

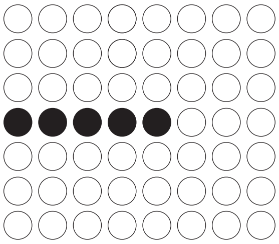

CHR040

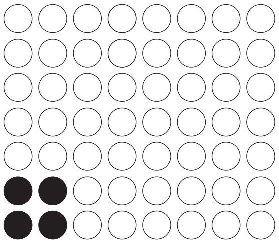

CHR041

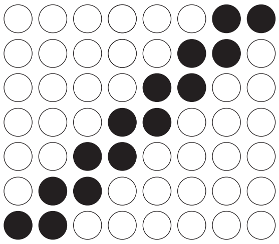

CHR042

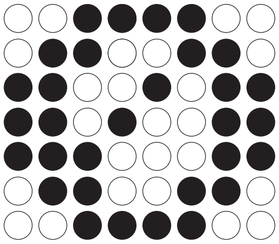

CHR043

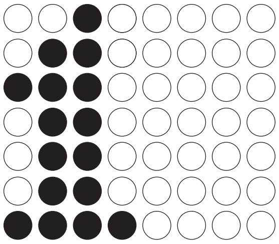

CHR044

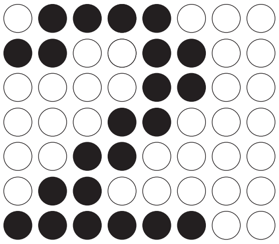

CHR045

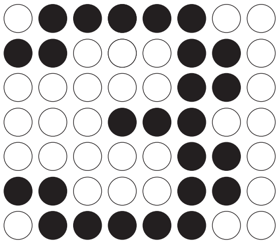

CHR046

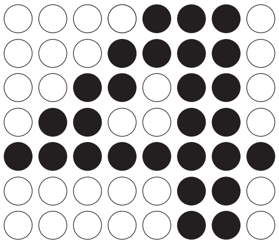

CHR047


CHR048

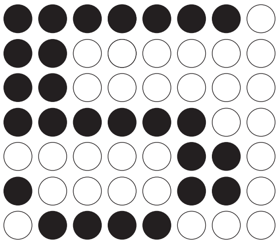

CHR049


CHR050

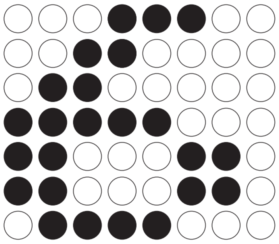

CHR051



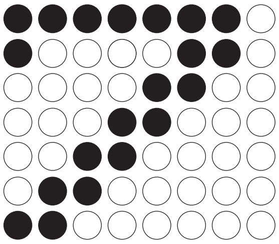
CHR052



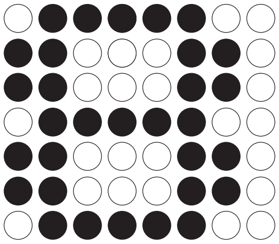
CHR053



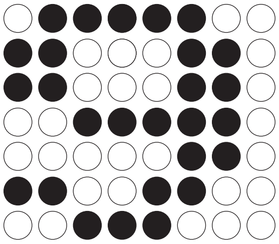
CHR054



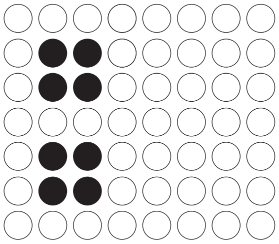
CHR055



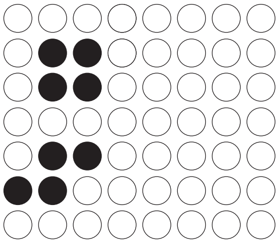
CHR056



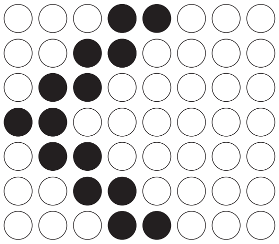
CHR057



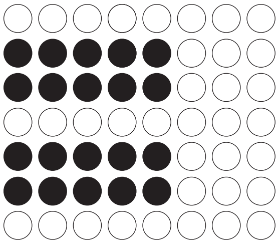
CHR058



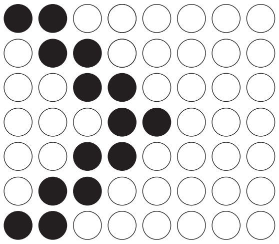
CHR059



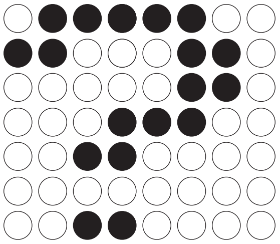
CHR060



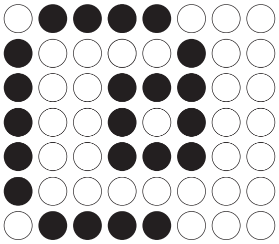
CHR061



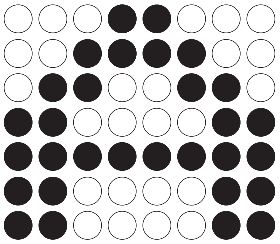
CHR062



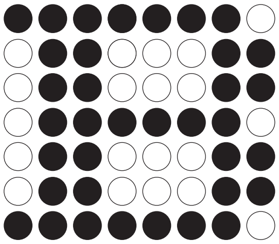
CHR063



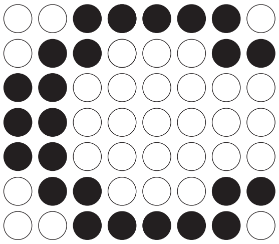
CHR064



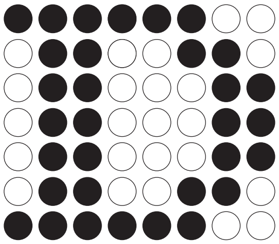
CHR065



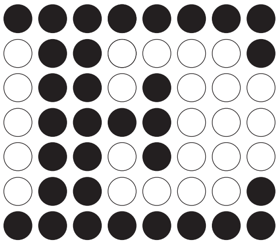
CHR066



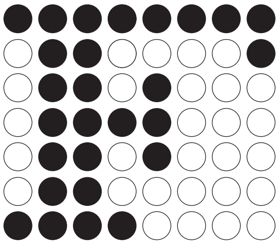
CHR067



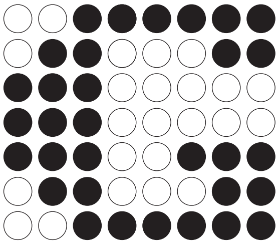
CHR068



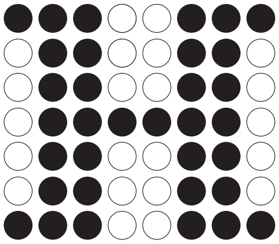
CHR069



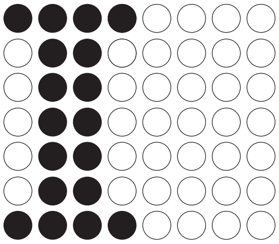
CHR070



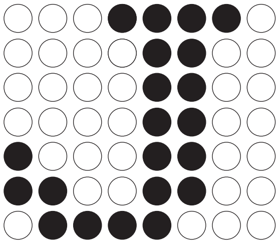
CHR071



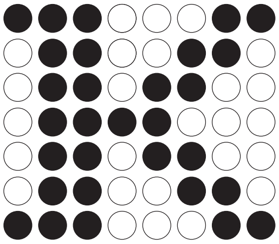
CHR072



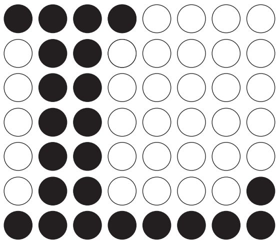
CHR073



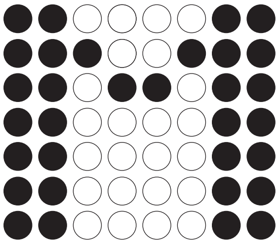
CHR074



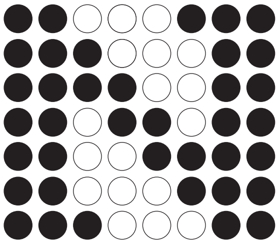
CHR075



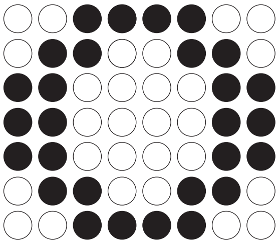
CHR076



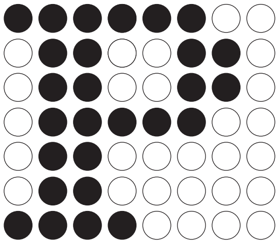
CHR077



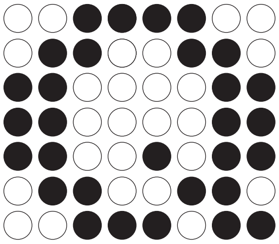
CHR078



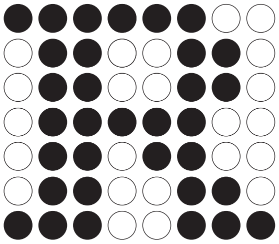
CHR079



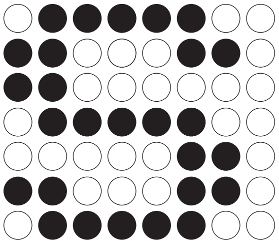
CHR080



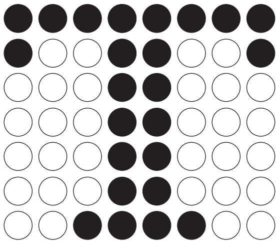
CHR081



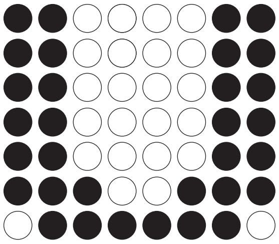
CHR082



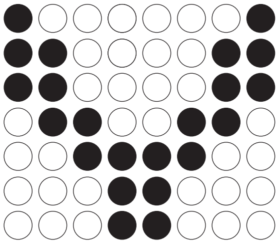
CHR083



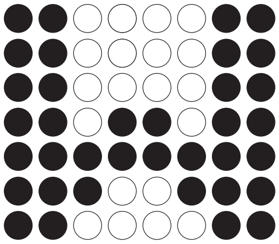
CHR084



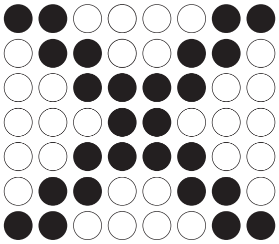
CHR085



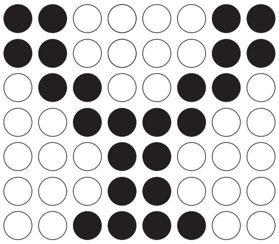
CHR086



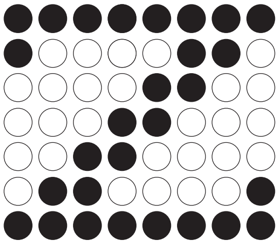
CHR087



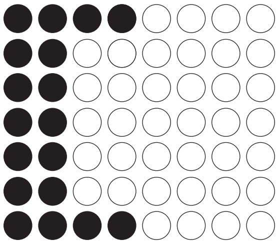
CHR088



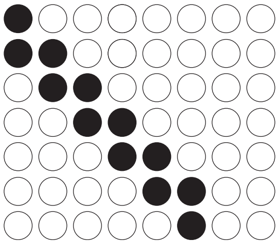
CHR089



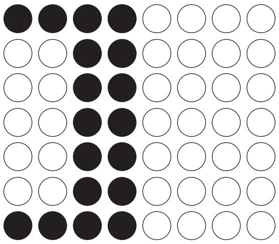
CHR090



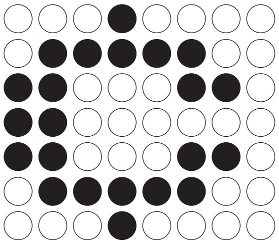
CHR091



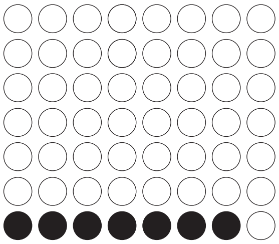
CHR092



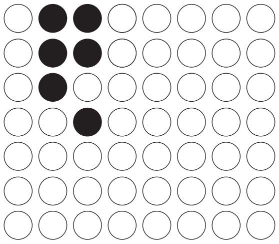
CHR093



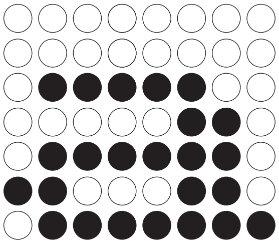
CHR094



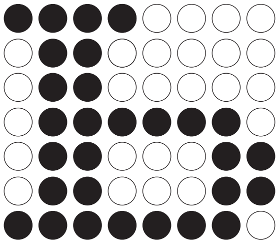
CHR095



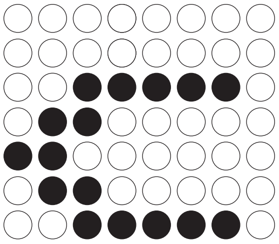
CHR096



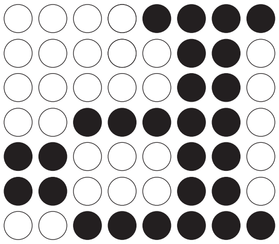
CHR097

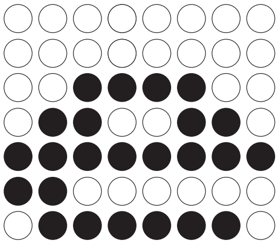


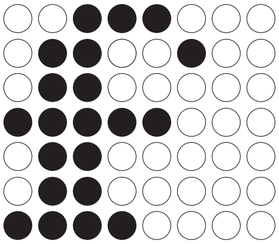
CHR098

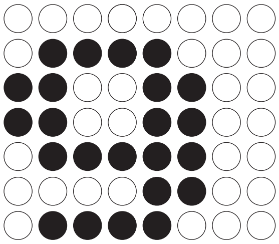


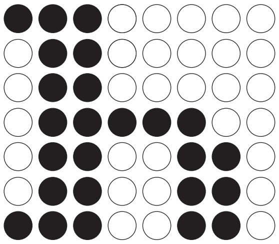
CHR099

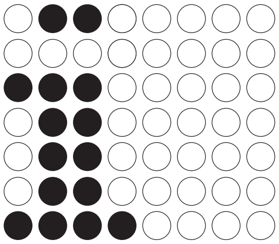

CHR100

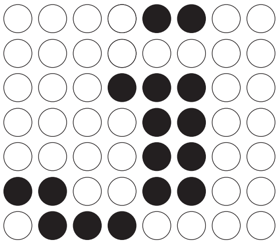

CHR101

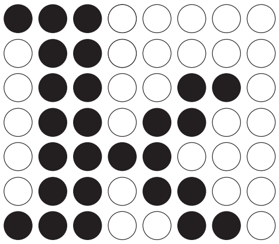

CHR102

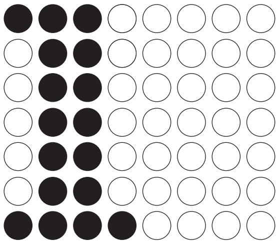

CHR103

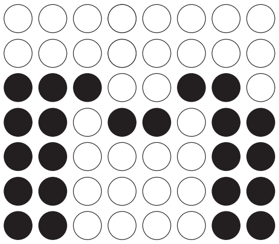

CHR104

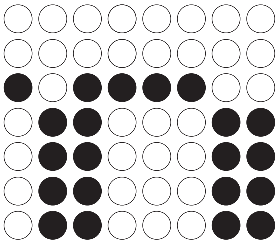

CHR105

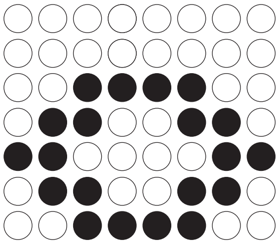

CHR106

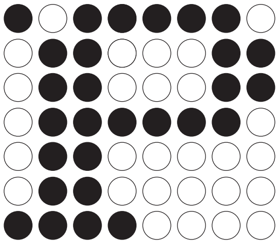

CHR107

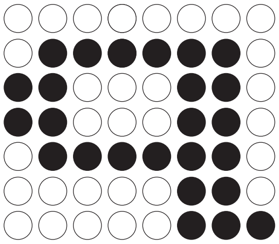

CHR108

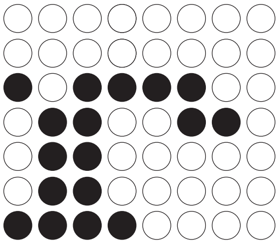

CHR109

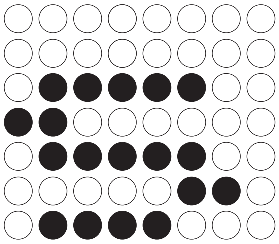

CHR110

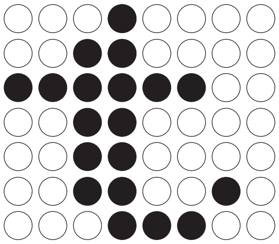

CHR111

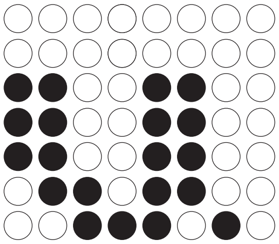

CHR112

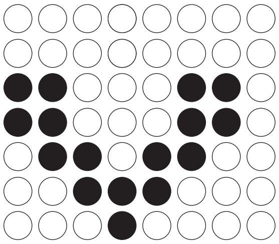

CHR113

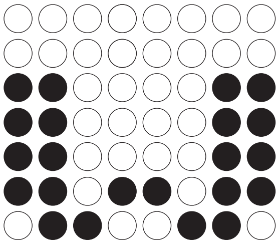

CHR114

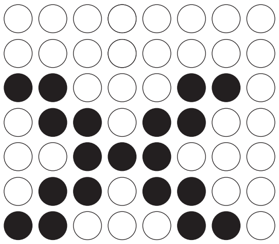

CHR115

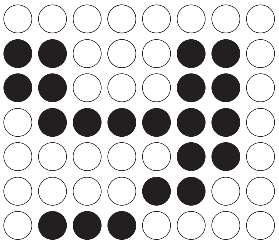

CHR116

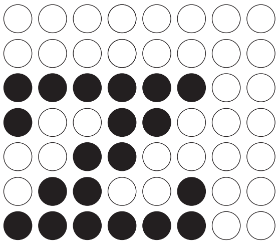

CHR117

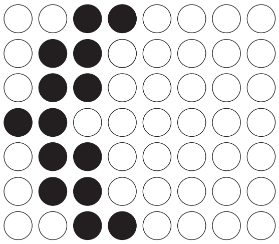

CHR118

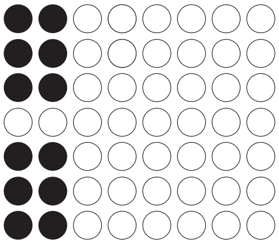

CHR119


CHR120

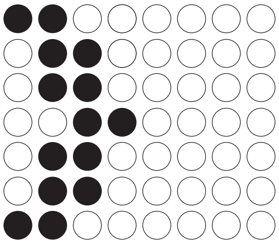

CHR121


CHR122

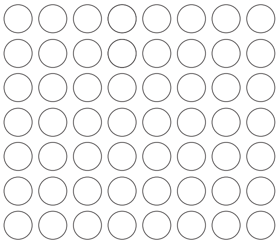

CHR123



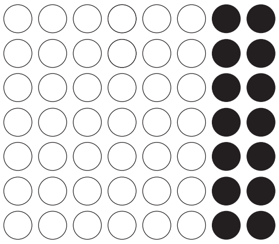
CHR124



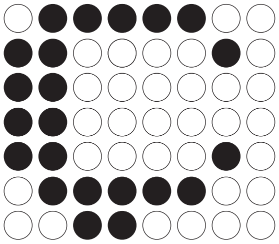
CHR125



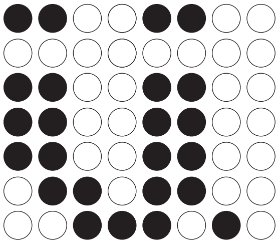
CHR126



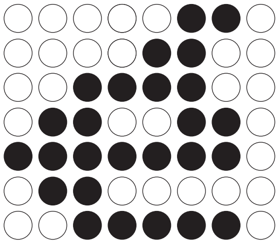
CHR127



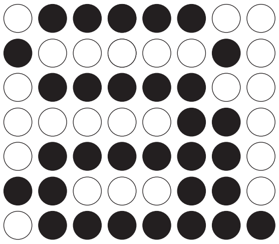
CHR128



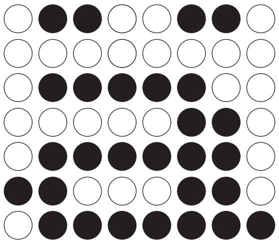
CHR129



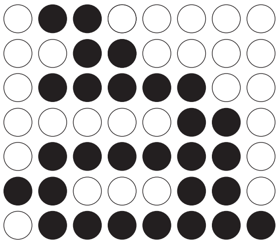
CHR130



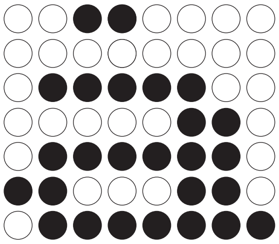
CHR131



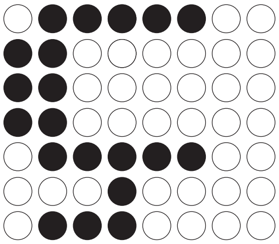
CHR132



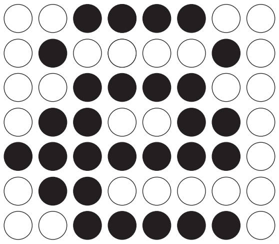
CHR133



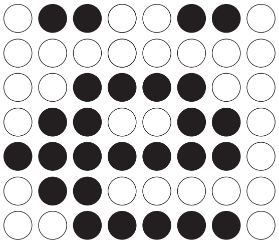
CHR134



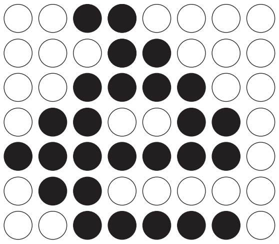
CHR135



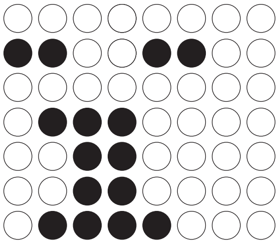
CHR136



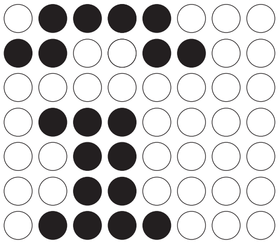
CHR137



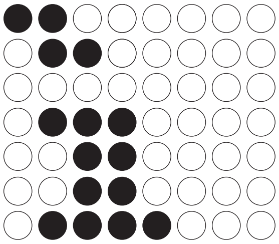
CHR138



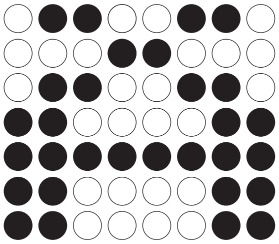
CHR139



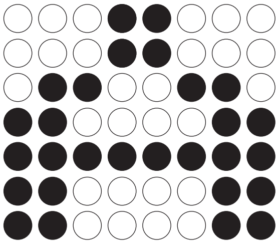
CHR140



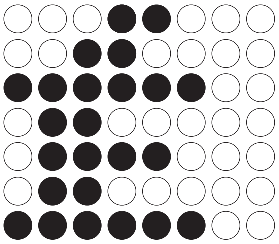
CHR141



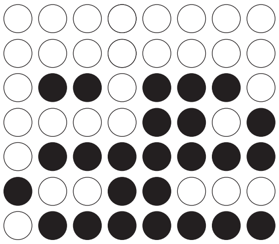
CHR142



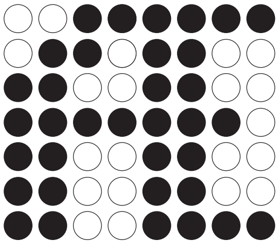
CHR143



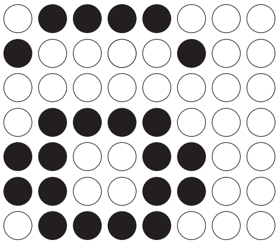
CHR144



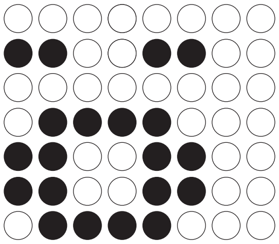
CHR145



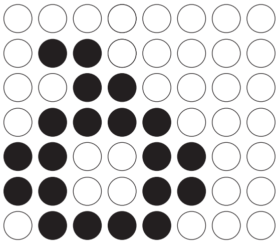
CHR146



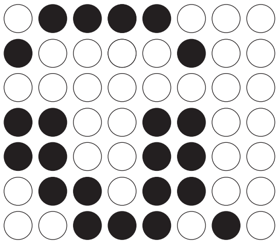
CHR147



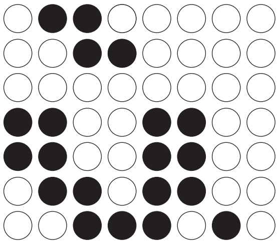
CHR148



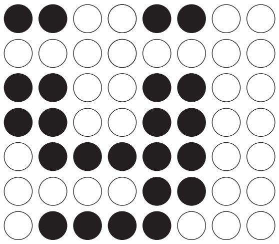
CHR149



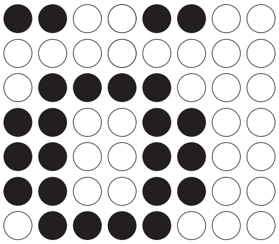
CHR150



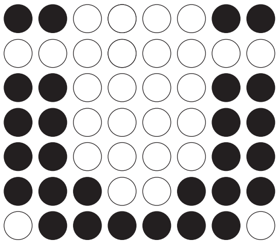
CHR151



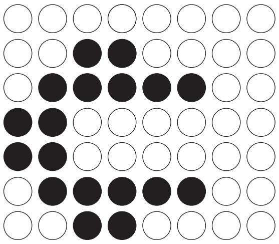
CHR152



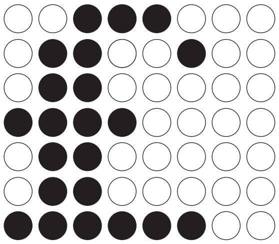
CHR153



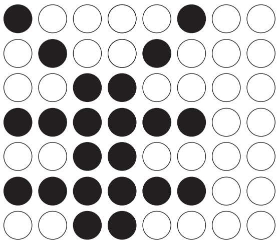
CHR154



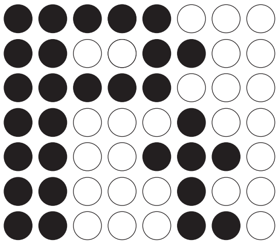
CHR155



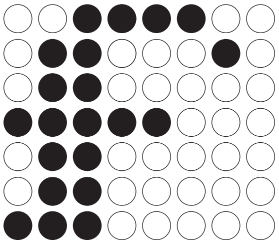
CHR156



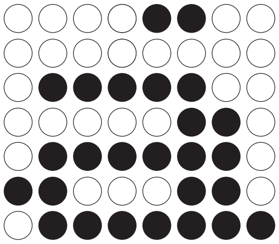
CHR157



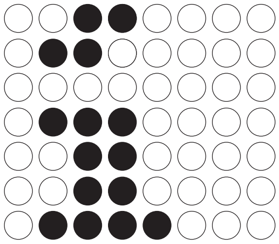
CHR158



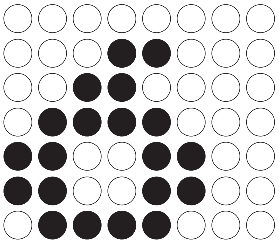
CHR158



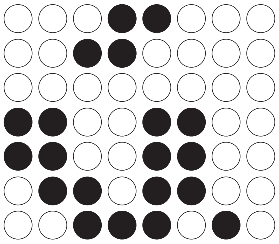
CHR160



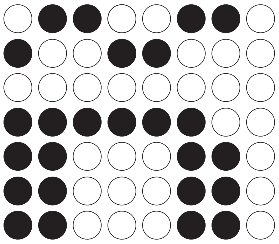
CHR161



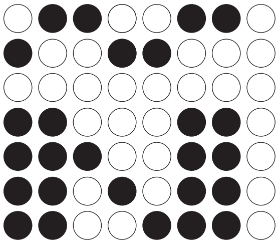
CHR162



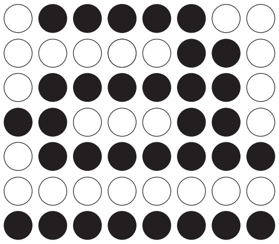
CHR163



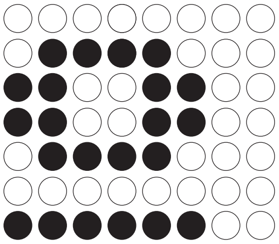
CHR164



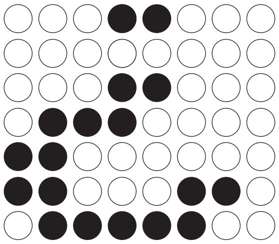
CHR165



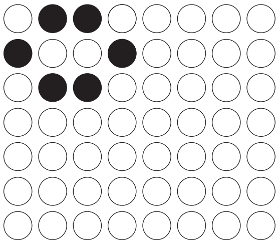
CHR166



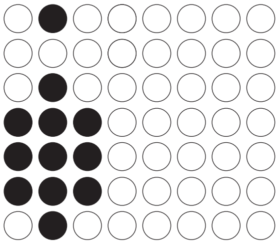
CHR167



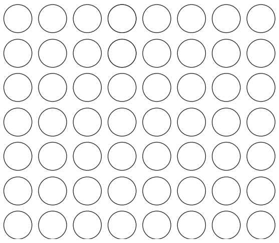
CHR168



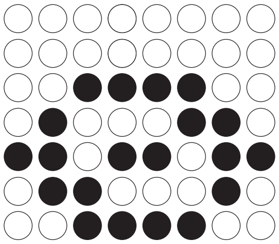
CHR169



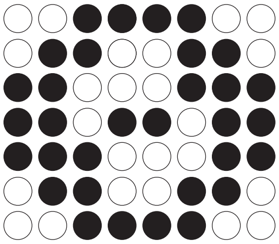
CHR170



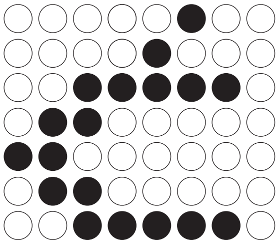
CHR171



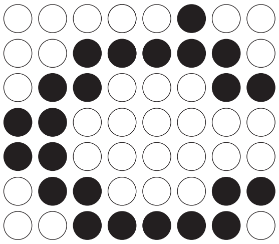
CHR172



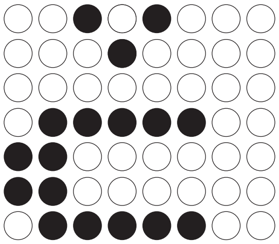
CHR173



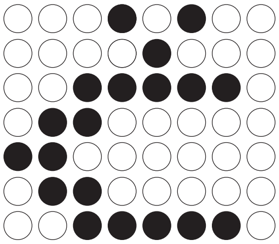
CHR174



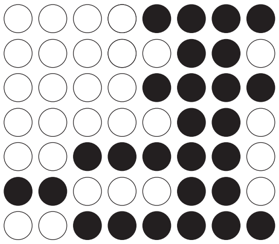
CHR175



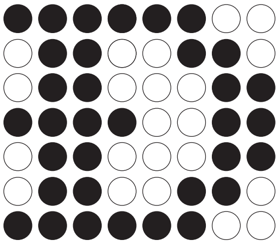
CHR176



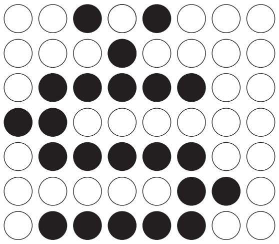
CHR177



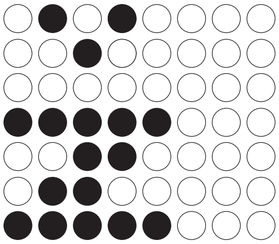
CHR178



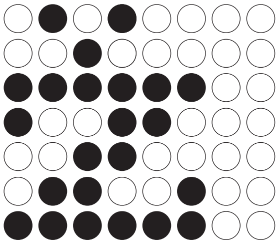
CHR179



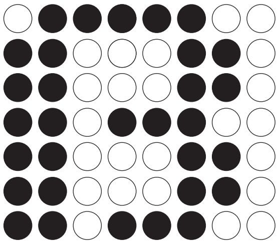
CHR180



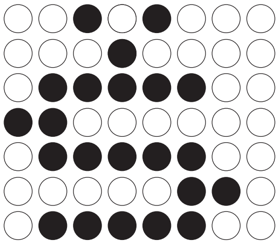
CHR181



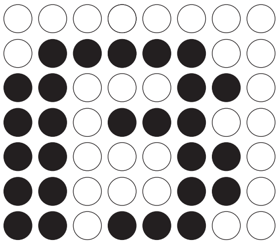
CHR182



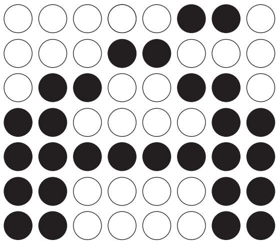
CHR183



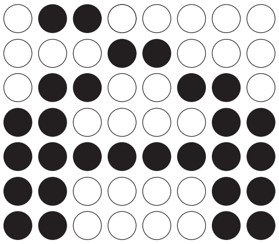
CHR184



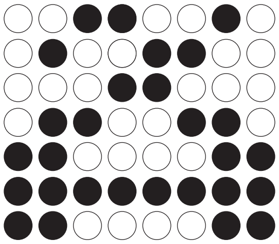
CHR185



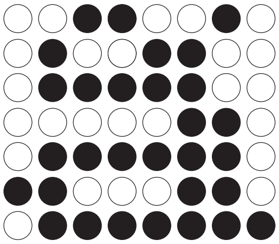
CHR186



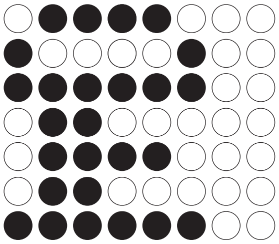
CHR187



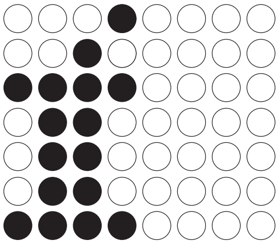
CHR188



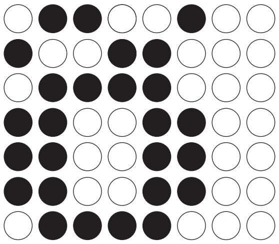
CHR189



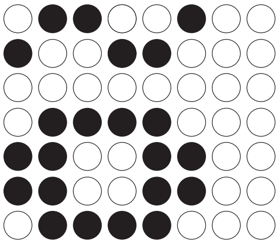
CHR190



CHR191

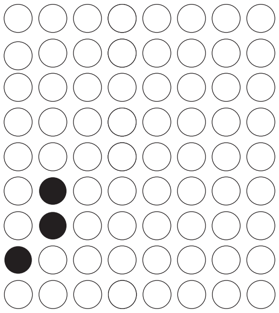


CHR192

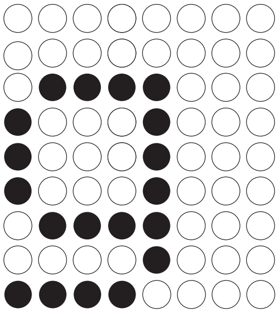


CHR193

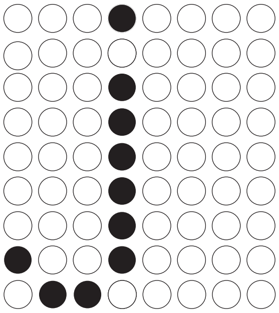
7.13.4 7-High True Descender Regular



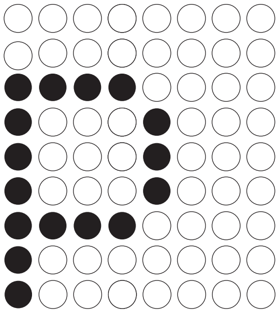
CHR7



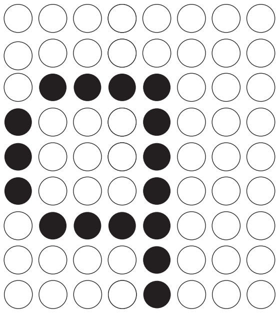
CHR7G



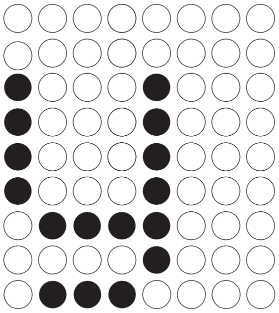
CHR7J



CHR7P

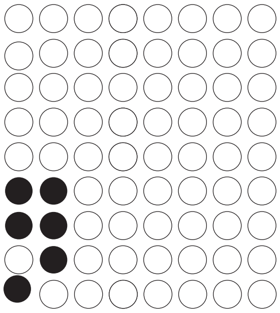


CHR7Q

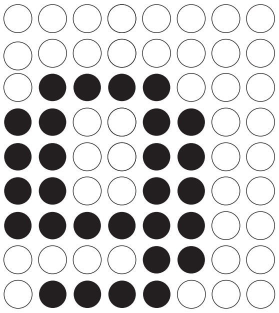


CHR7Y

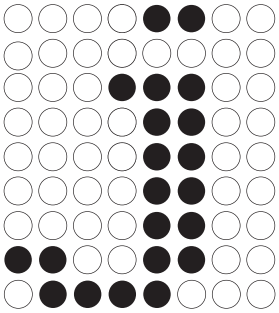
7.13.5 7-High True Descender Fancy



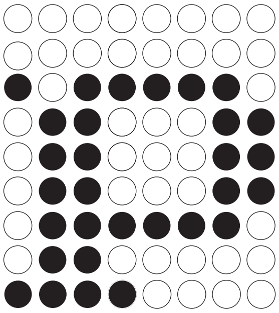
CHR7FC



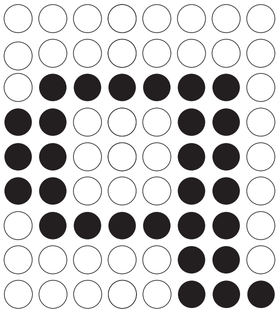
CHR7FG



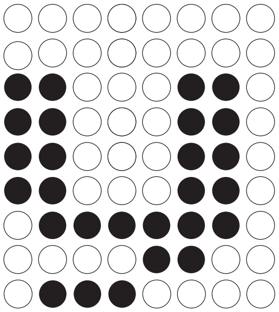
CHR7FJ



CHR7FP

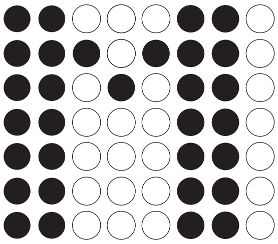


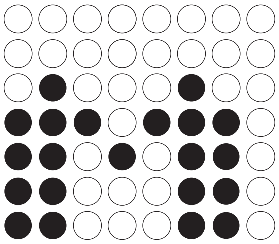
CHR7FQ

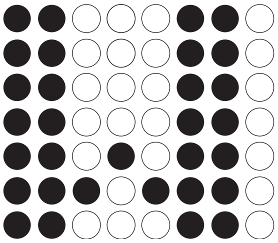


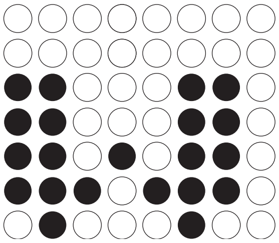
CHR7FY

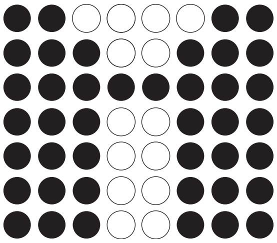
7.13.6 7-High Fat Character

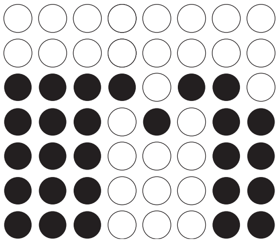

FTSM

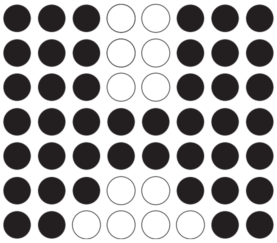

FTSM1

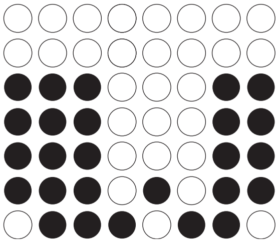

FTSW


FTSW1

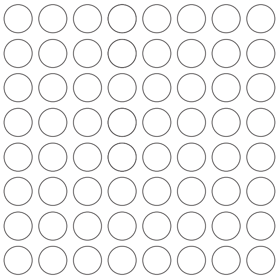

SFTM

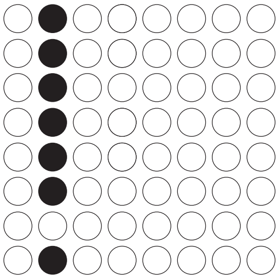

SFTM1

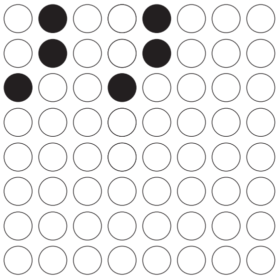

SFTW

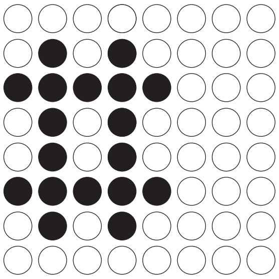

SFTW1

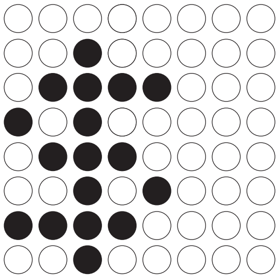
7.13.7 8-High Regular (SS8)

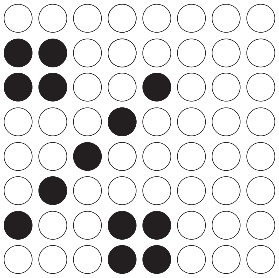

CHR032

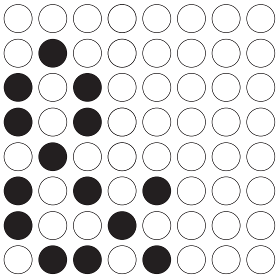

CHR033

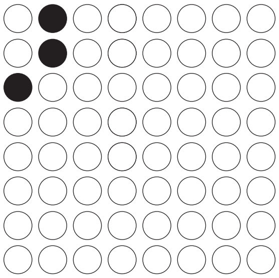

CHR034

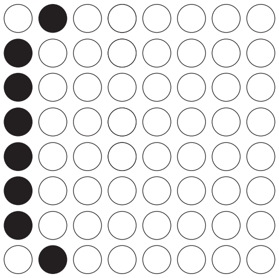

CHR035

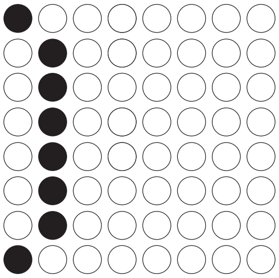

CHR036

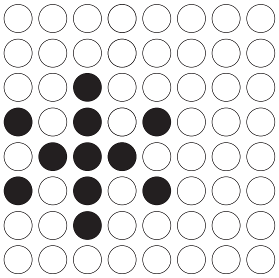

CHR037

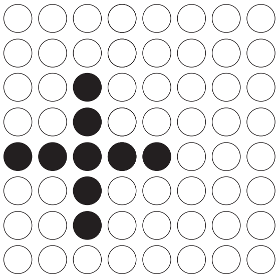

CHR038

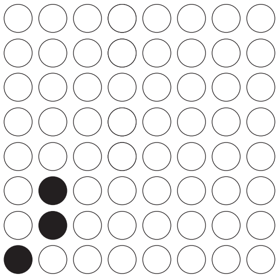

CHR039

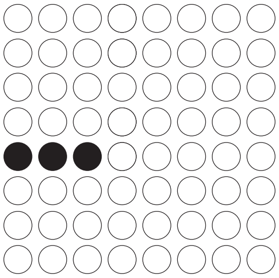

CHR040

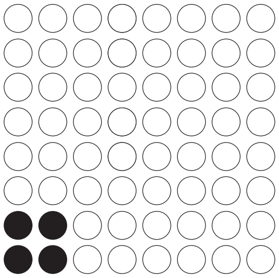

CHR041

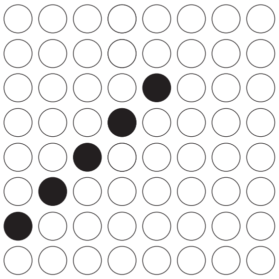

CHR042

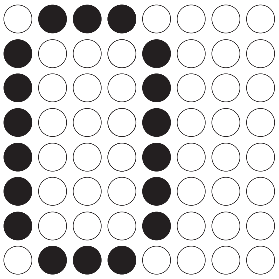

CHR043

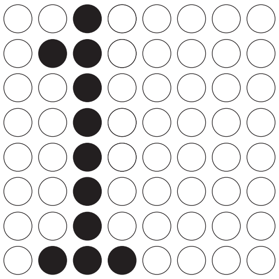

CHR044

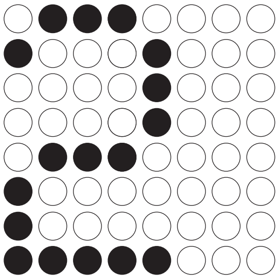

CHR045

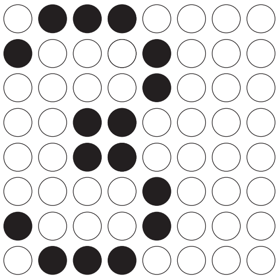

CHR046

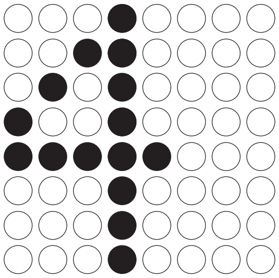

CHR047

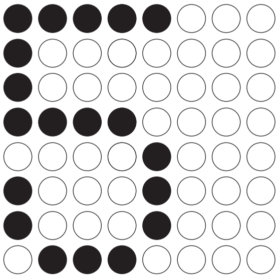

CHR048

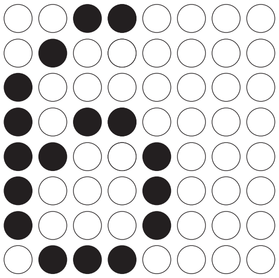

CHR049

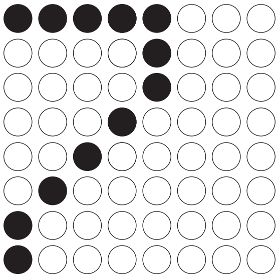

CHR050

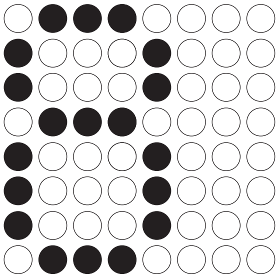

CHR051

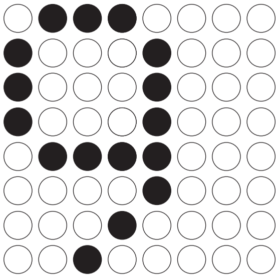

CHR052

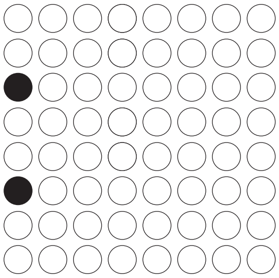

CHR053

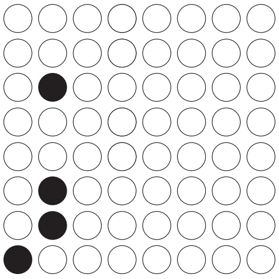

CHR054

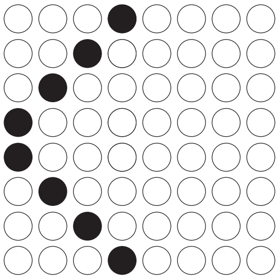

CHR055

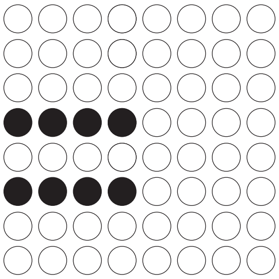

CHR056

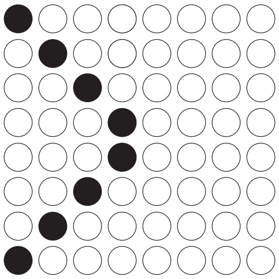

CHR057

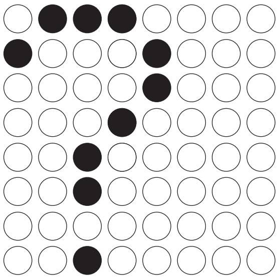

CHR058

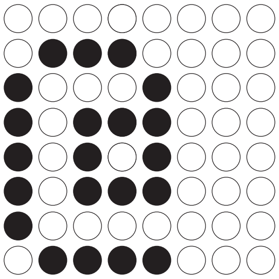

CHR059


CHR060

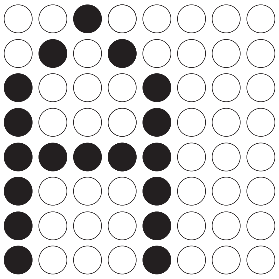

CHR061


CHR062

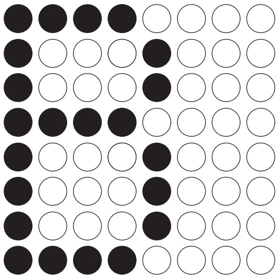

CHR063



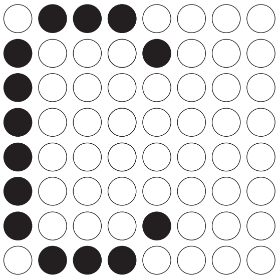
CHR064



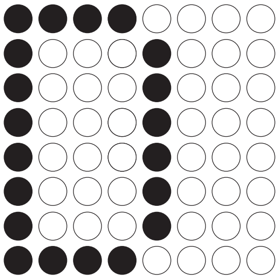
CHR065



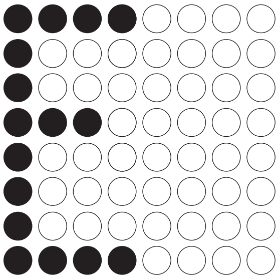
CHR066



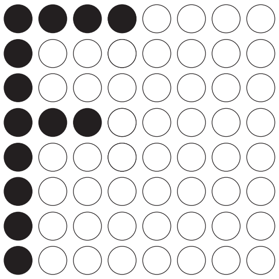
CHR067



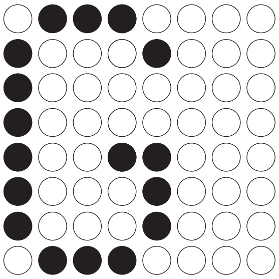
CHR068



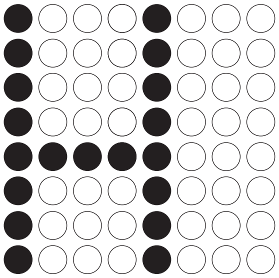
CHR069



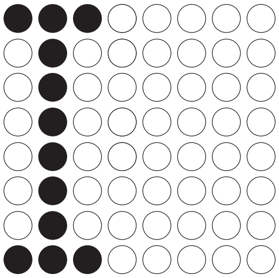
CHR070



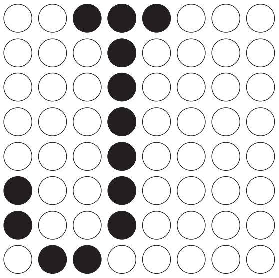
CHR071



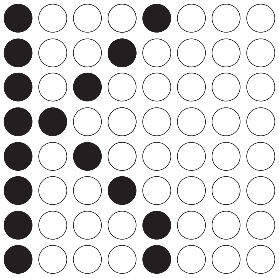
CHR072



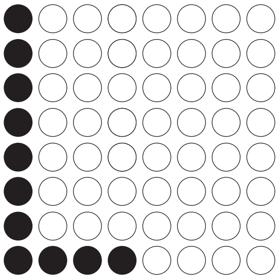
CHR073



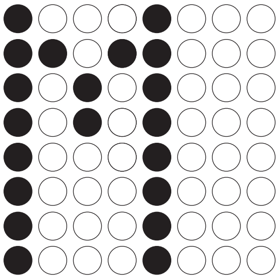
CHR074



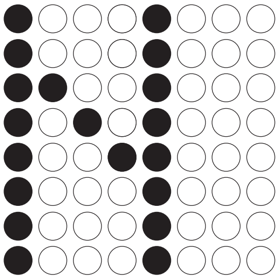
CHR075



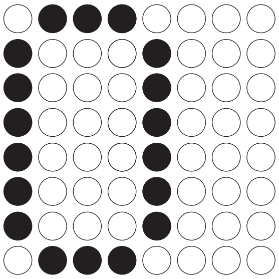
CHR076



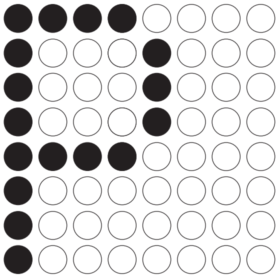
CHR077



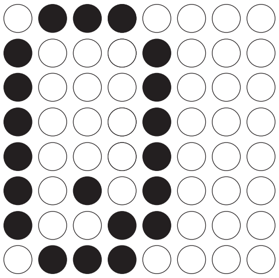
CHR078



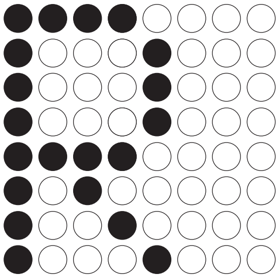
CHR079



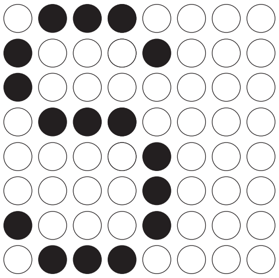
CHR080



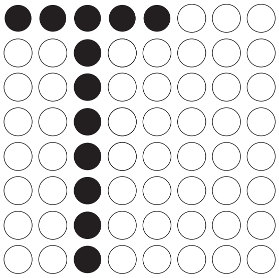
CHR081

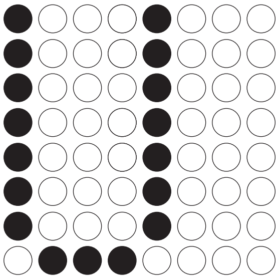


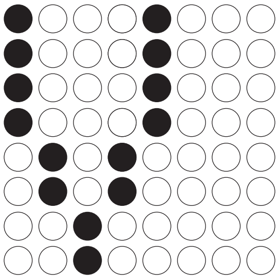
CHR082

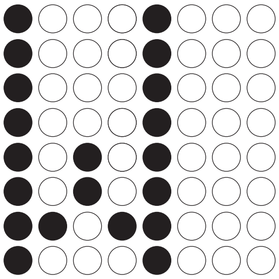


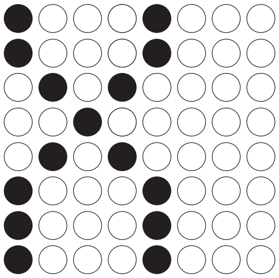
CHR083

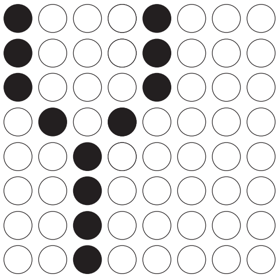

CHR084

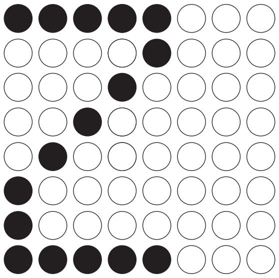

CHR085

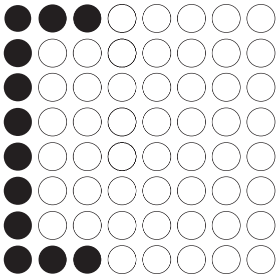

CHR086

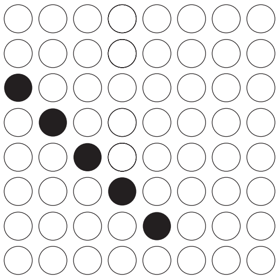

CHR087

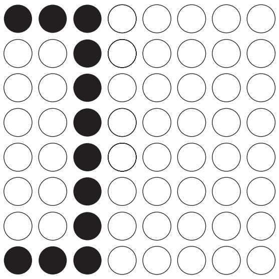

CHR088

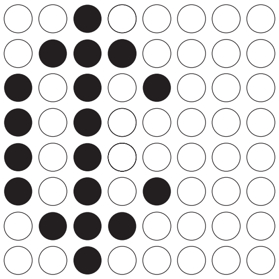

CHR089

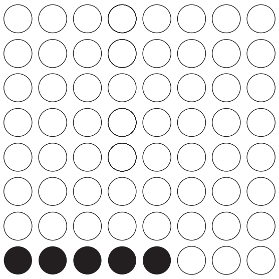

CHR090

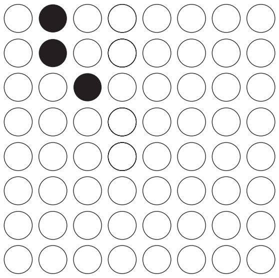

CHR091

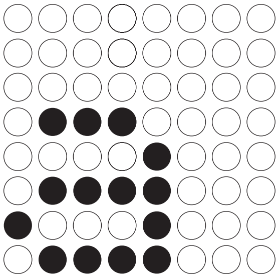

CHR092

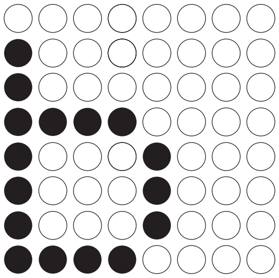

CHR093

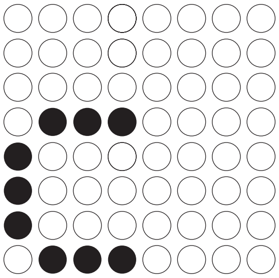

CHR094

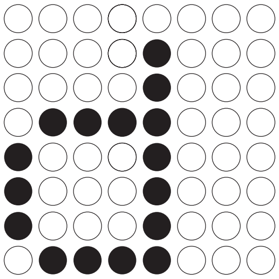

CHR095

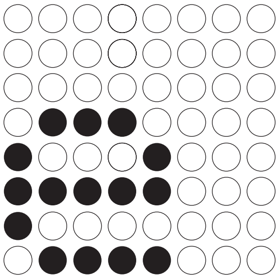

CHR096

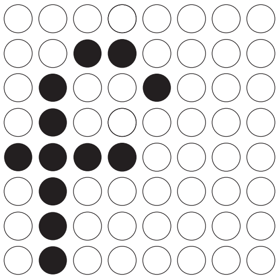

CHR097

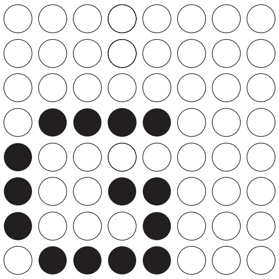

CHR098

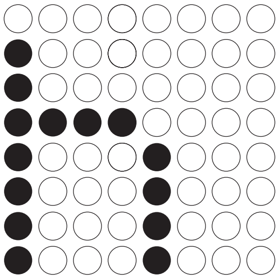

CHR099


CHR100

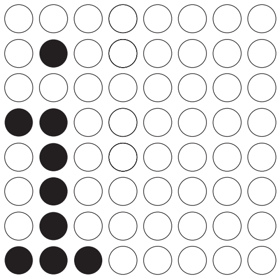

CHR101


CHR102

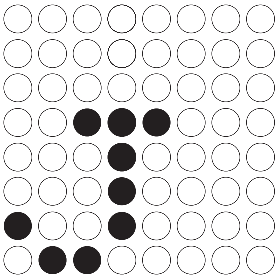

CHR103



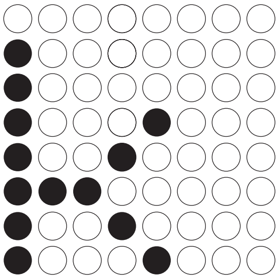
CHR104



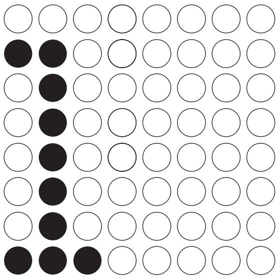
CHR105



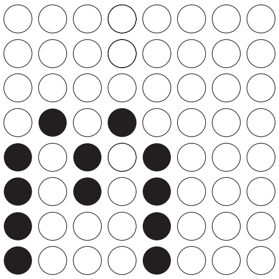
CHR106



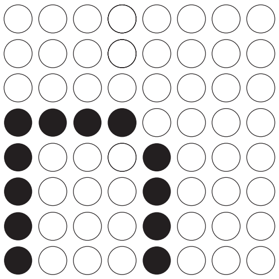
CHR107



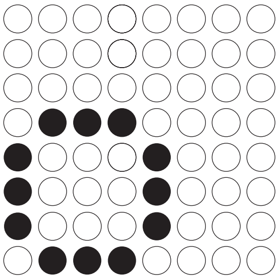
CHR108



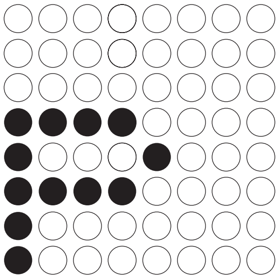
CHR109



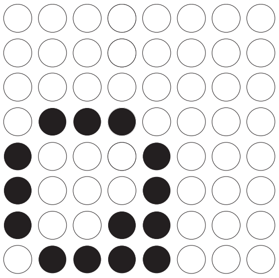
CHR110



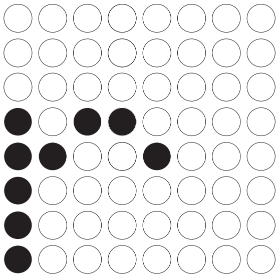
CHR111



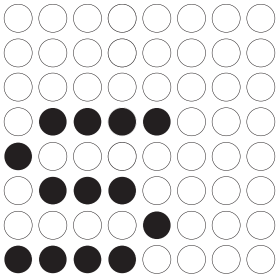
CHR112



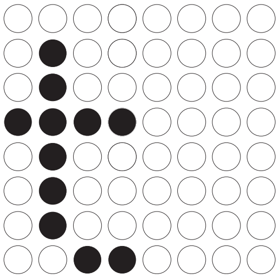
CHR113



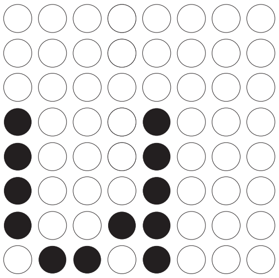
CHR114



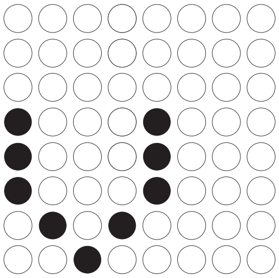
CHR115



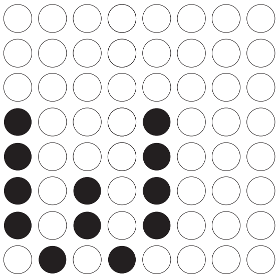
CHR116



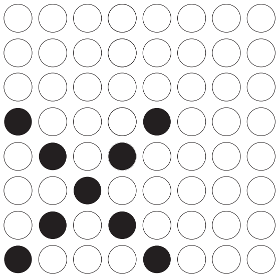
CHR117



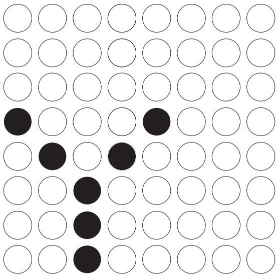
CHR118



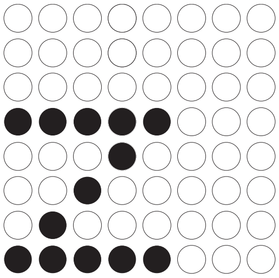
CHR119



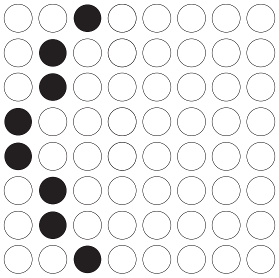
CHR120



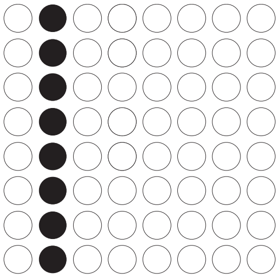
CHR121

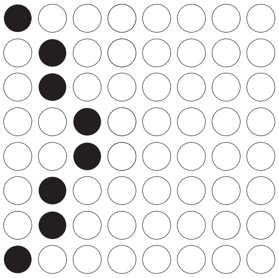


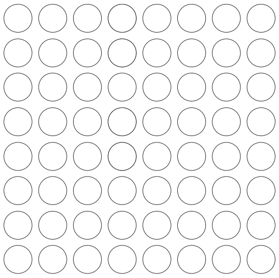
CHR122

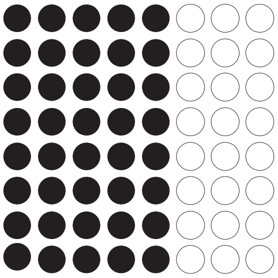


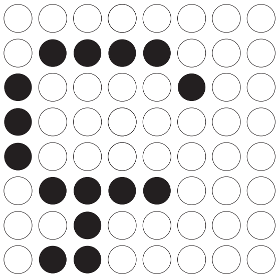
CHR123

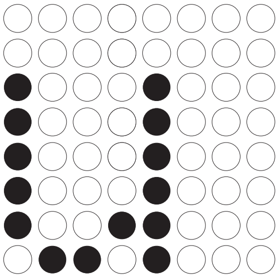

CHR124

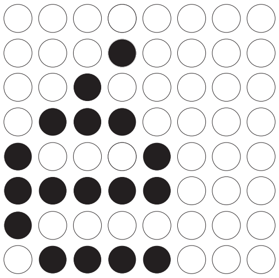

CHR125

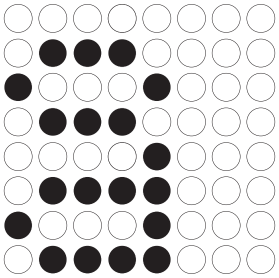

CHR126

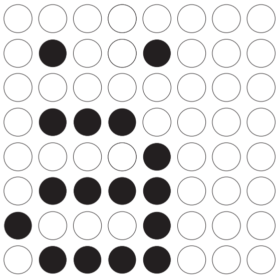

CHR127

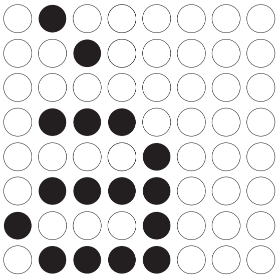

CHR128

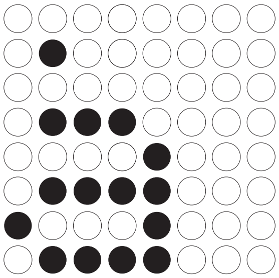

CHR129

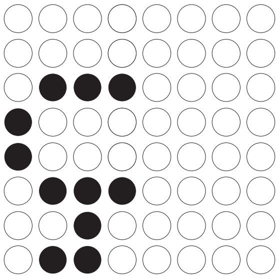

CHR130

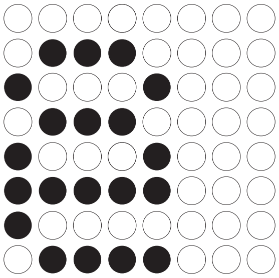

CHR131

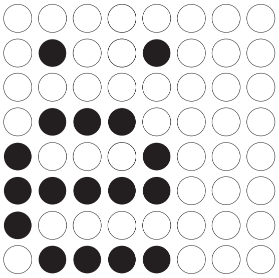

CHR132

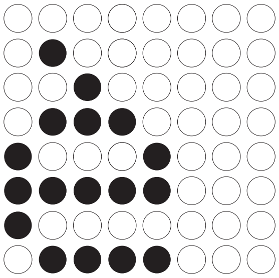

CHR133

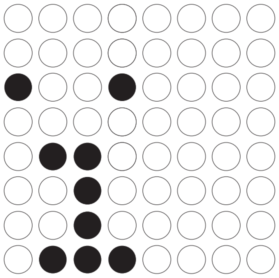

CHR134

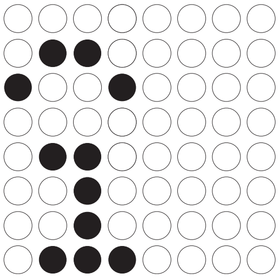

CHR135

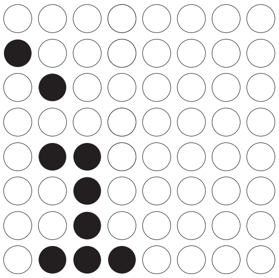

CHR136

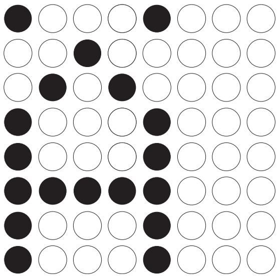

CHR137

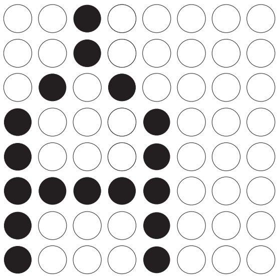

CHR138

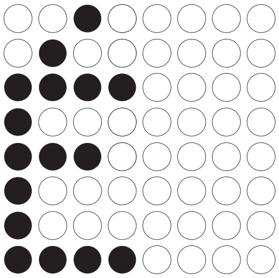

CHR139

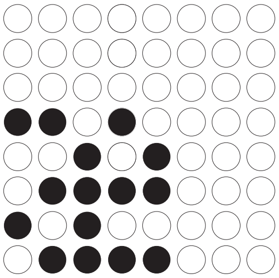

CHR140

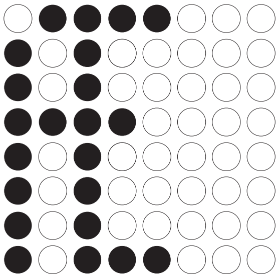

CHR141

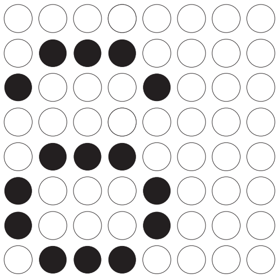

CHR142

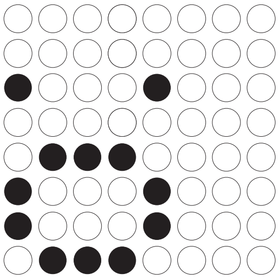

CHR143

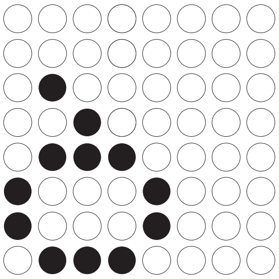

CHR144

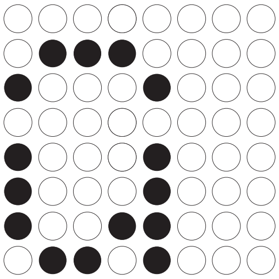

CHR145

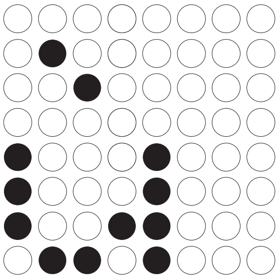

CHR146

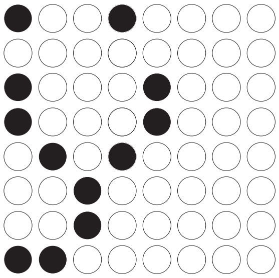

CHR147

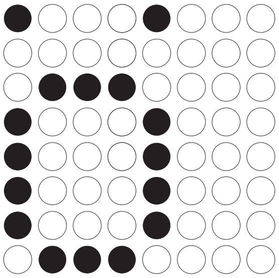

CHR148

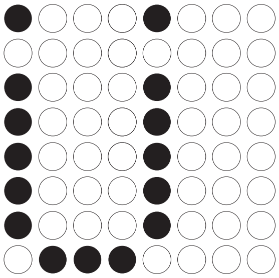

CHR149

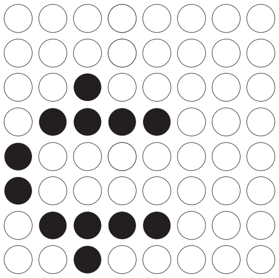

CHR150

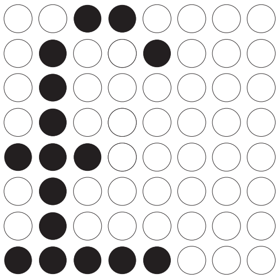

CHR151

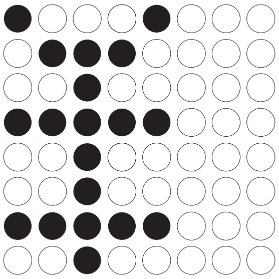

CHR152

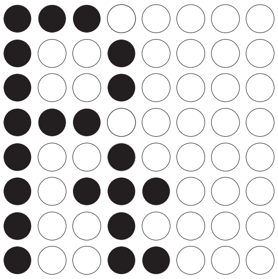

CHR153

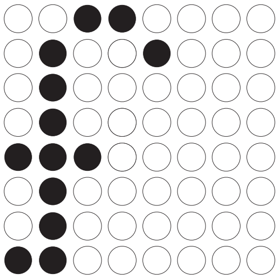

CHR154

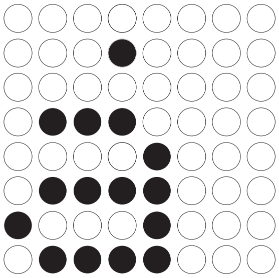

CHR155

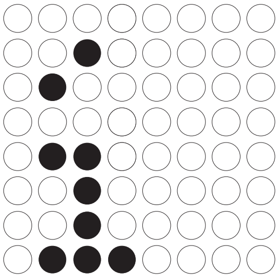

CHR156

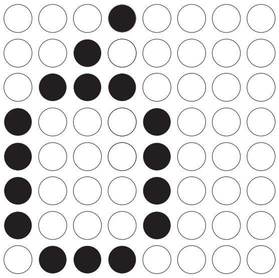

CHR157

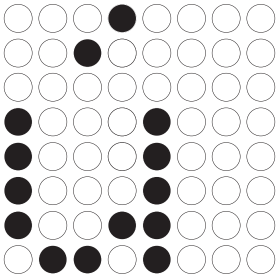

CHR158

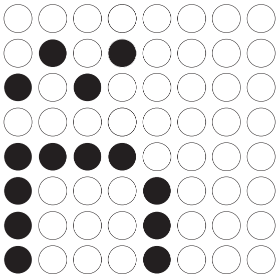

CHR159

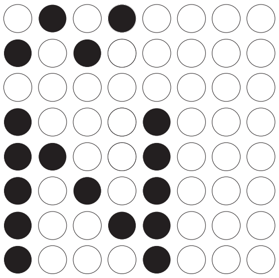

CHR160

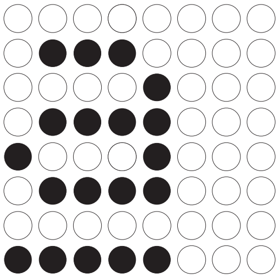

CHR161

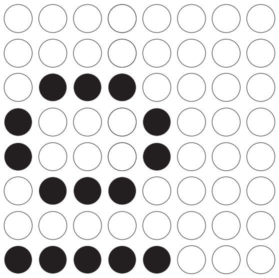

CHR162

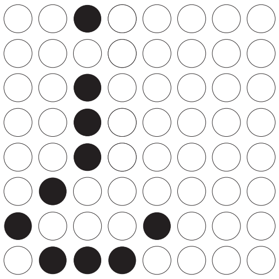

CHR163

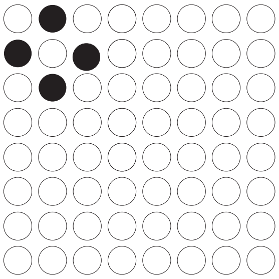

CHR164

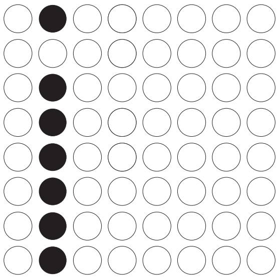

CHR165

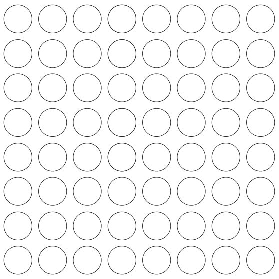

CHR166

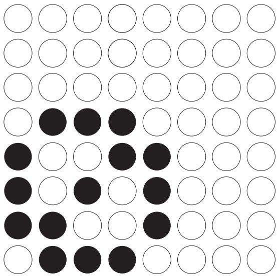

CHR167

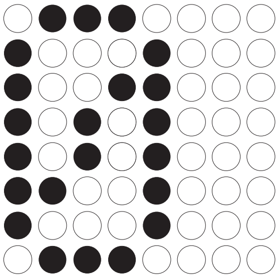

CHR168

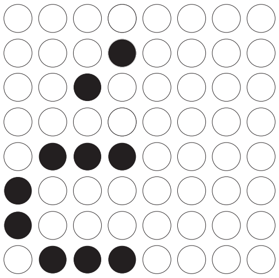

CHR169

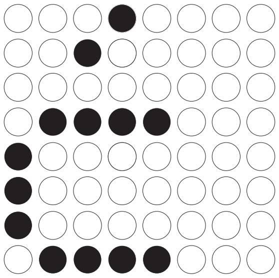

CHR170

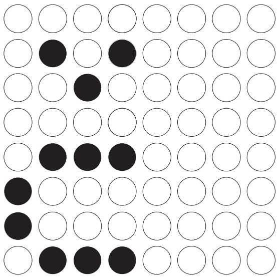

CHR171

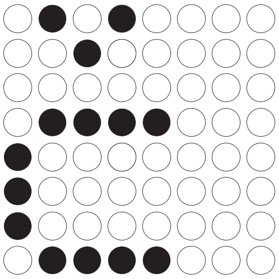

CHR172

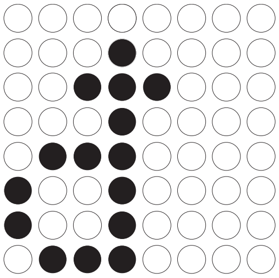

CHR173

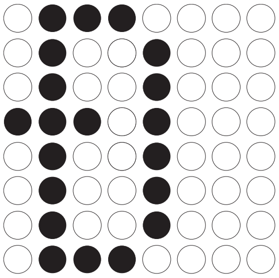

CHR174

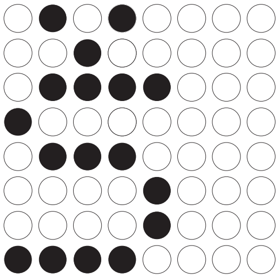

CHR175

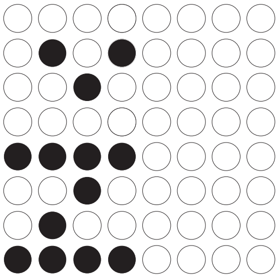

CHR176

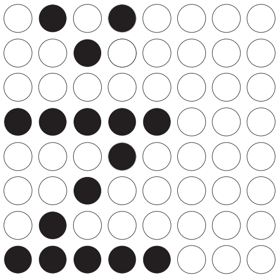

CHR177

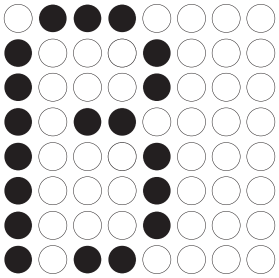

CHR178

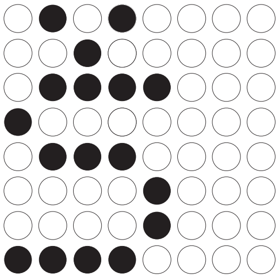

CHR179


CHR180

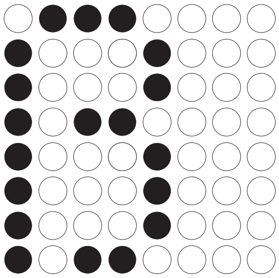

CHR181


CHR182

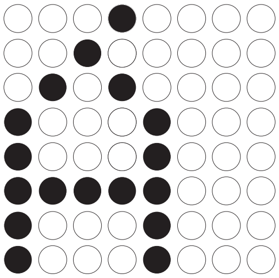

CHR183



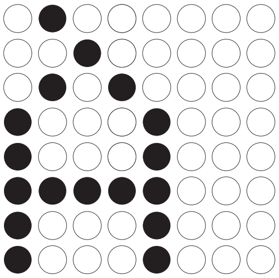
CHR184



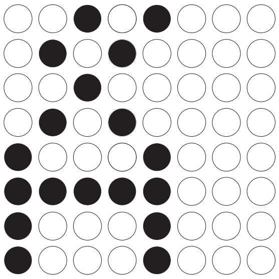
CHR185



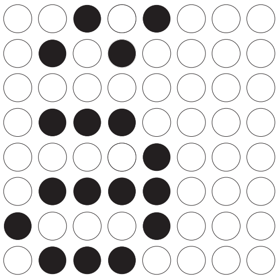
CHR186



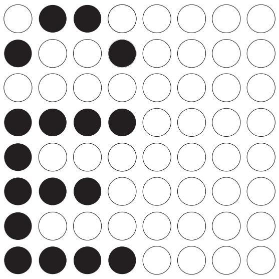
CHR187



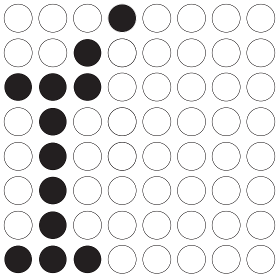
CHR188



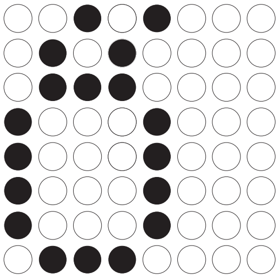
CHR189



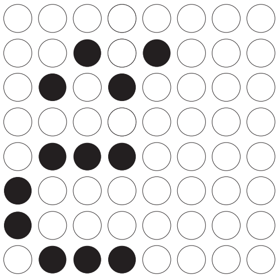
CHR190



CHR191

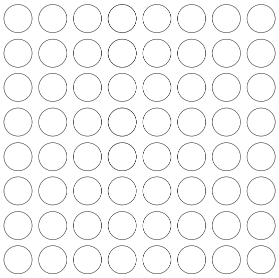


CHR192

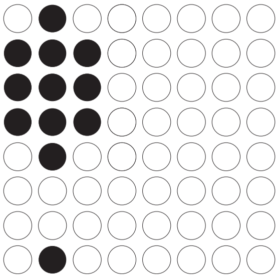


CHR193

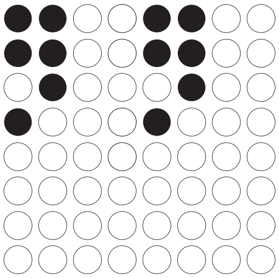
7.13.8 8-High Fancy (SF8)



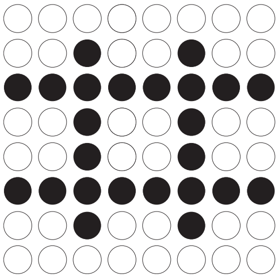
CHR032



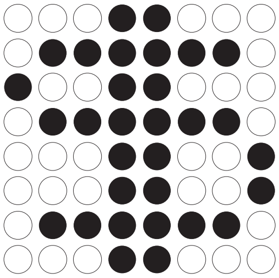
CHR033



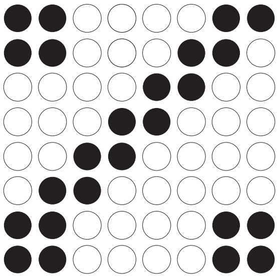
CHR034



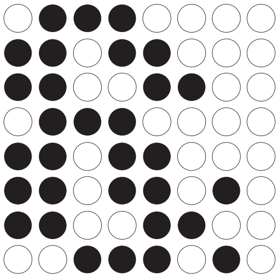
CHR035



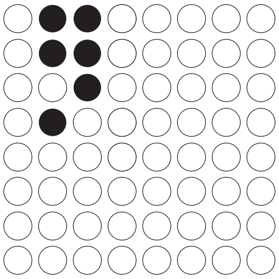
CHR036



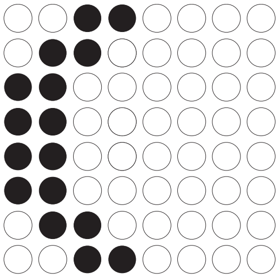
CHR037

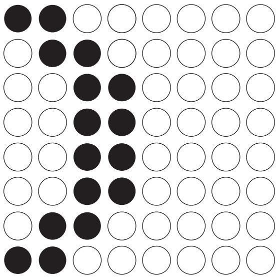


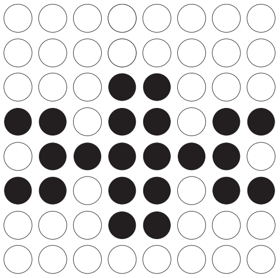
CHR038

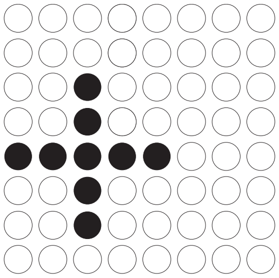


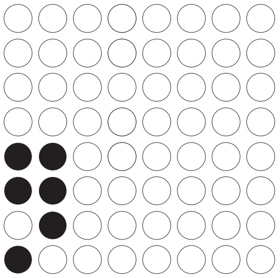
CHR039

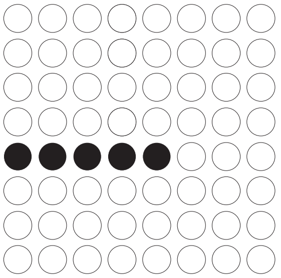

CHR040

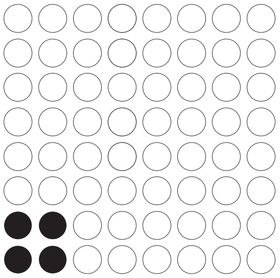

CHR041

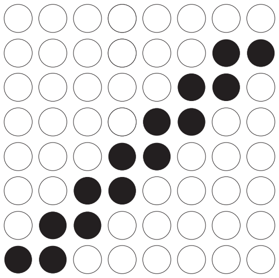

CHR042

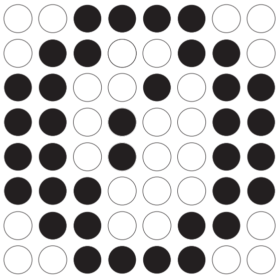

CHR043

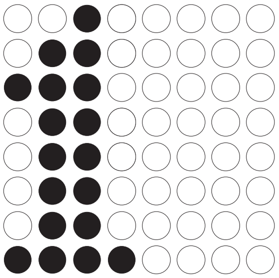

CHR044

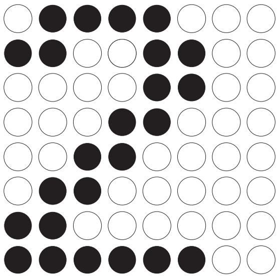

CHR045

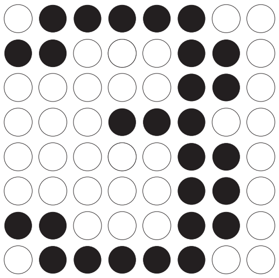

CHR046

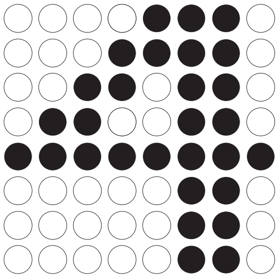

CHR047

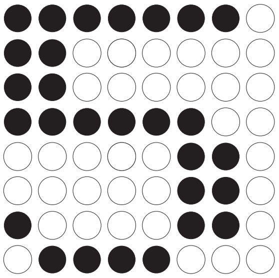

CHR048

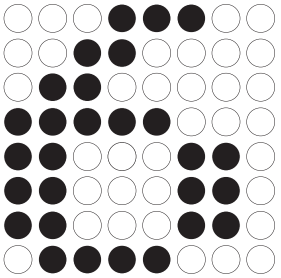

CHR049

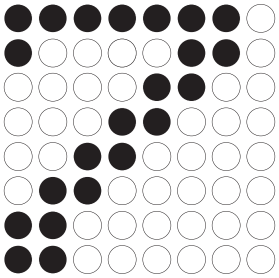

CHR050

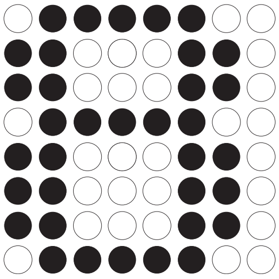

CHR051

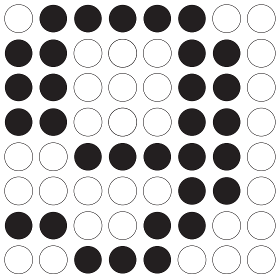

CHR052

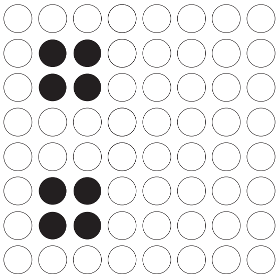

CHR053

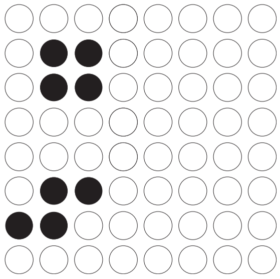

CHR054

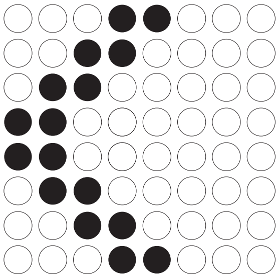

CHR055

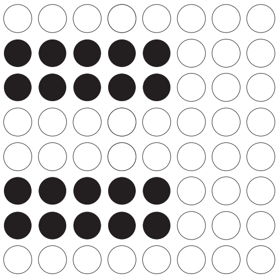

CHR056

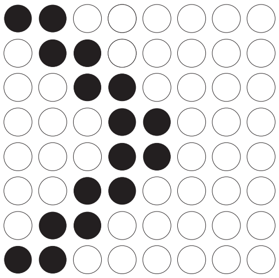

CHR057

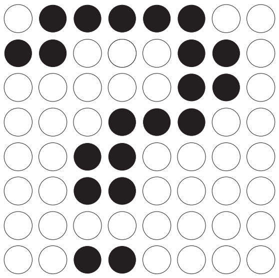

CHR058

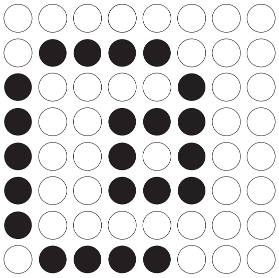

CHR059

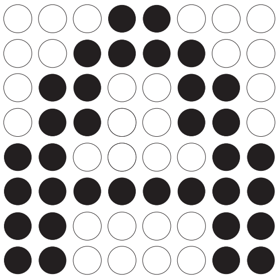

CHR060

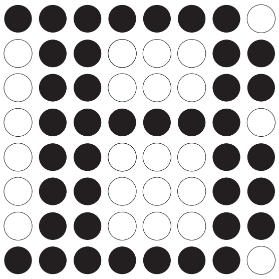

CHR061

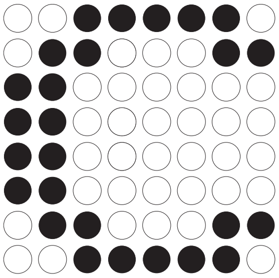

CHR062

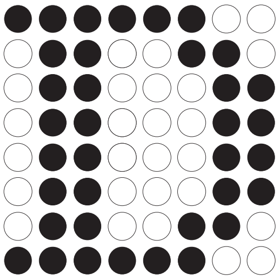

CHR063

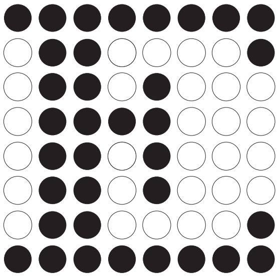

CHR064

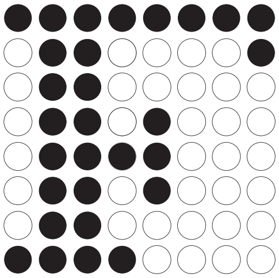

CHR065

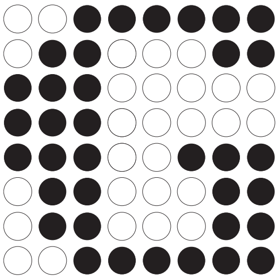

CHR066

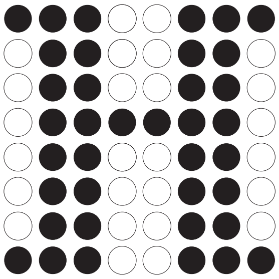

CHR067

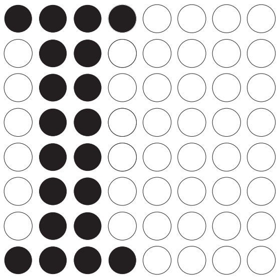

CHR068

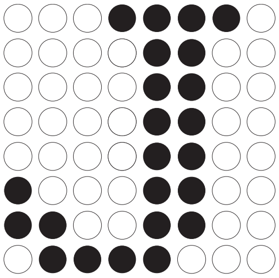

CHR069

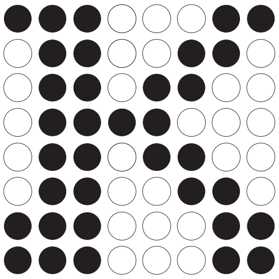

CHR070

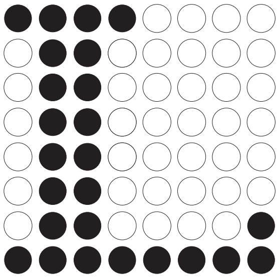

CHR071

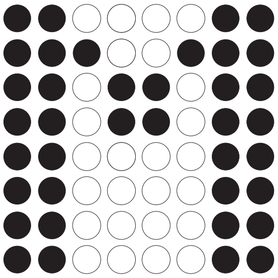

CHR072

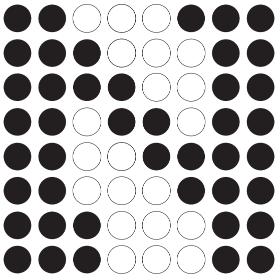

CHR073

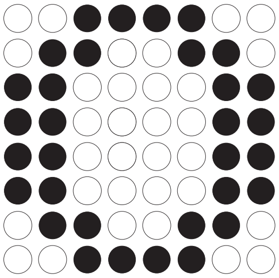

CHR074

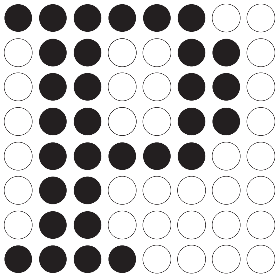

CHR075

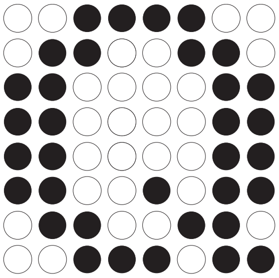

CHR076

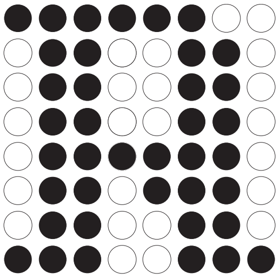

CHR077

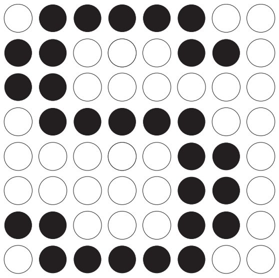

CHR078

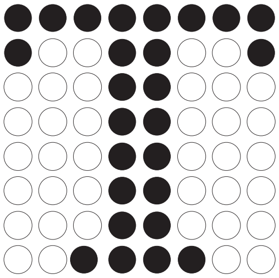

CHR079

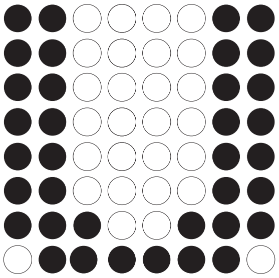

CHR080

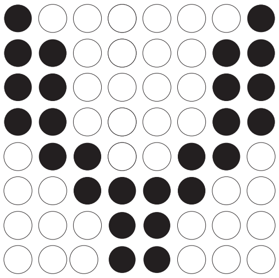

CHR081

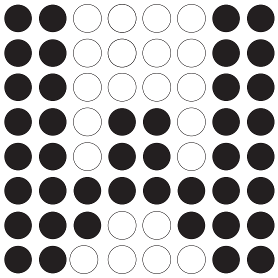

CHR082

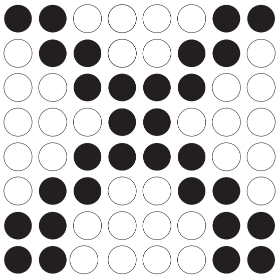

CHR083

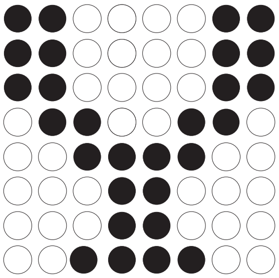

CHR084

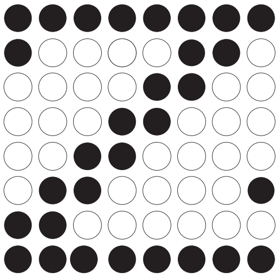

CHR085

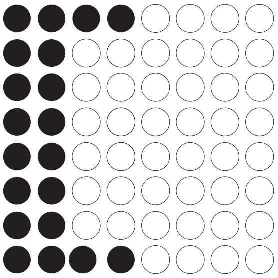

CHR086

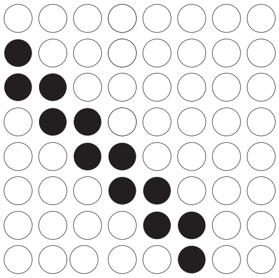

CHR087

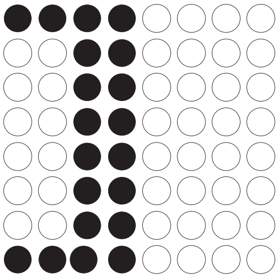

CHR088

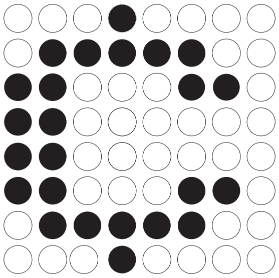

CHR089

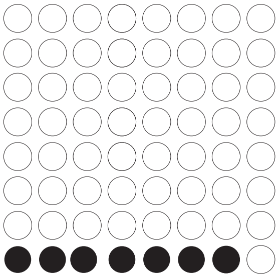

CHR090

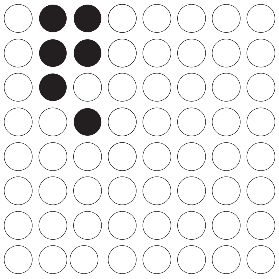

CHR091

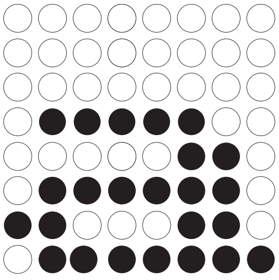

CHR092

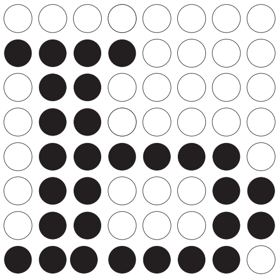

CHR093

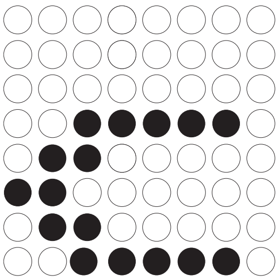

CHR094

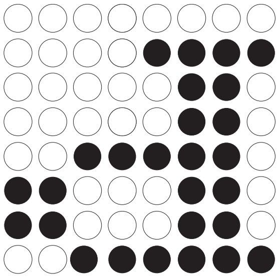

CHR095

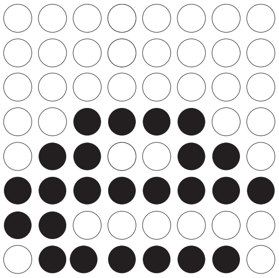

CHR096

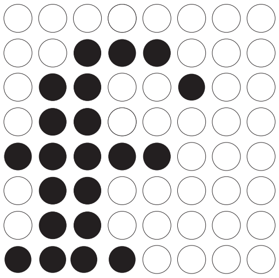

CHR097

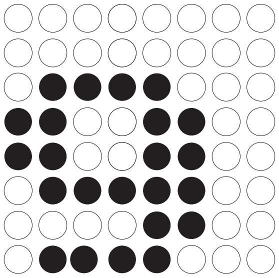

CHR098

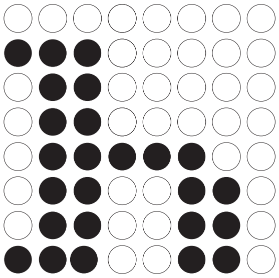

CHR099

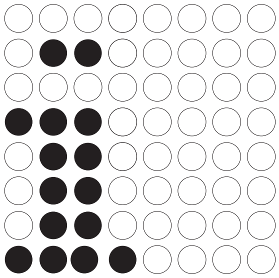

CHR100

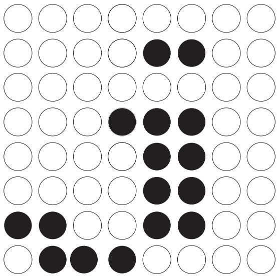

CHR101

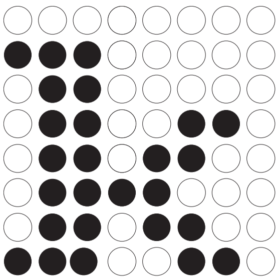

CHR102

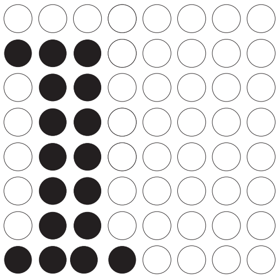

CHR103

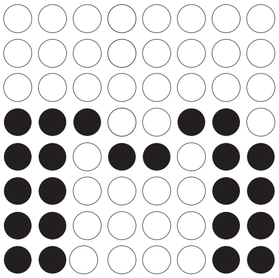

CHR104

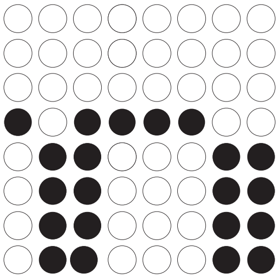

CHR105

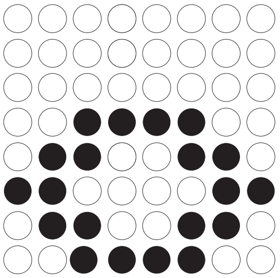

CHR106

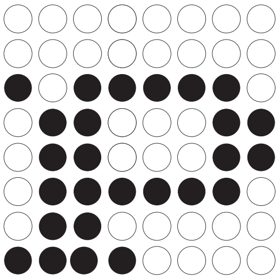

CHR107

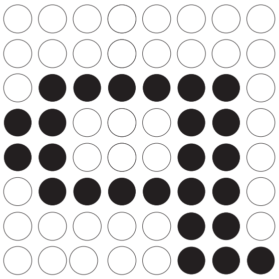

CHR108

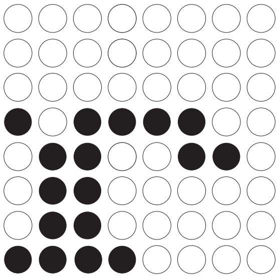

CHR109

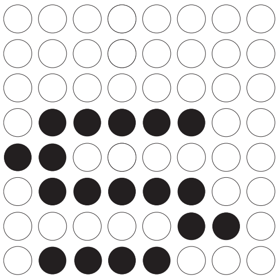

CHR110

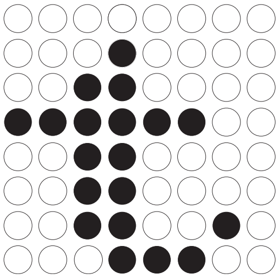

CHR111

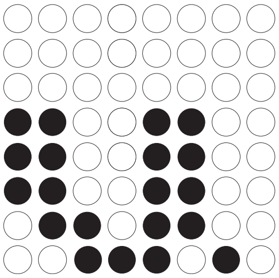

CHR112

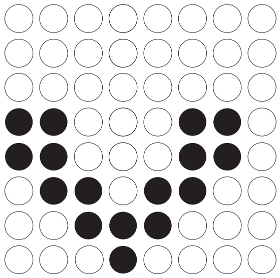

CHR113

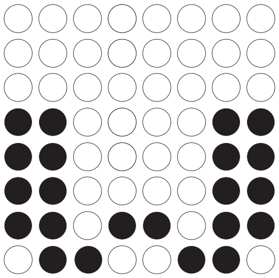

CHR114

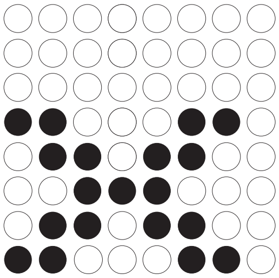

CHR115


CHR116

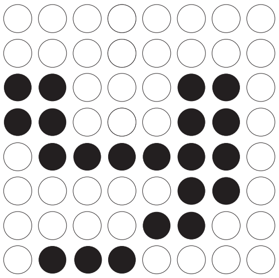

CHR117


CHR118

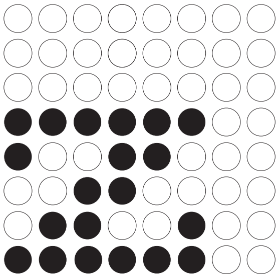

CHR119



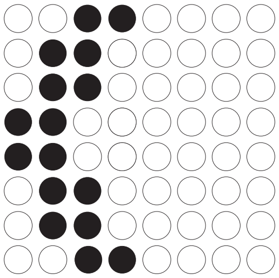
CHR120



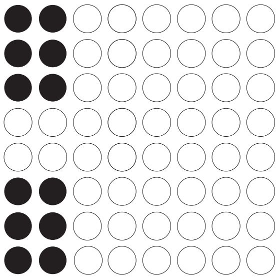
CHR121



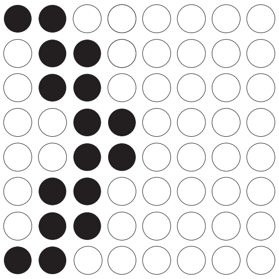
CHR122



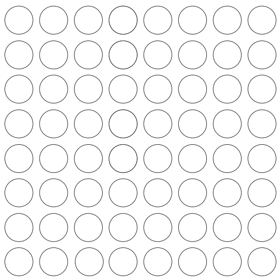
CHR123



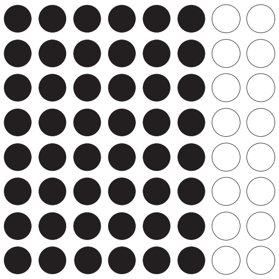
CHR124



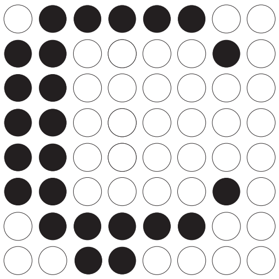
CHR125



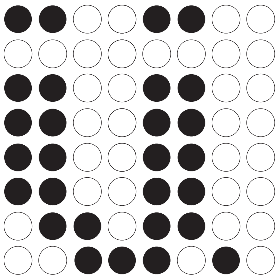
CHR126



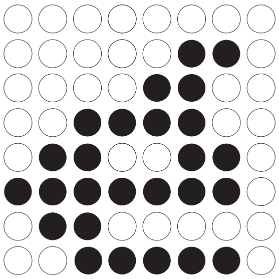
CHR127



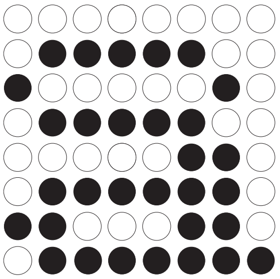
CHR128



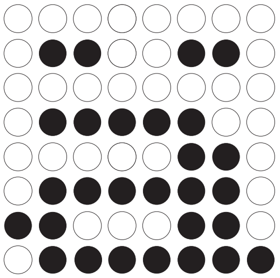
CHR129



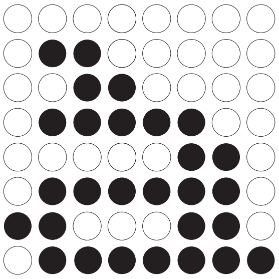
CHR130



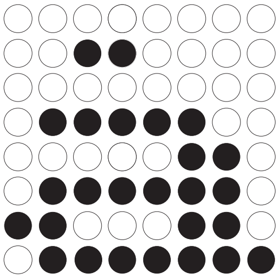
CHR131



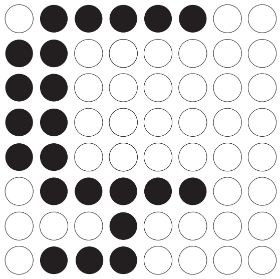
CHR132



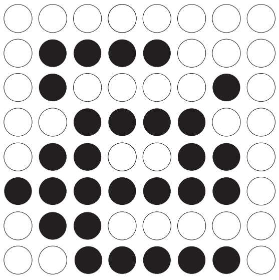
CHR133



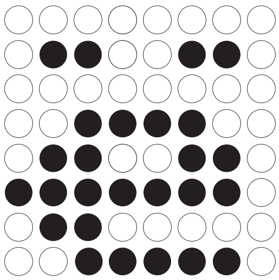
CHR134



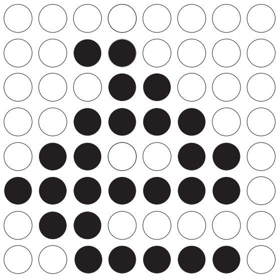
CHR135



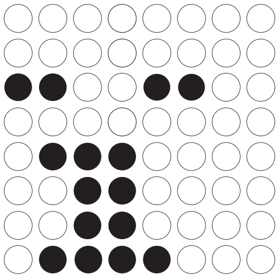
CHR136



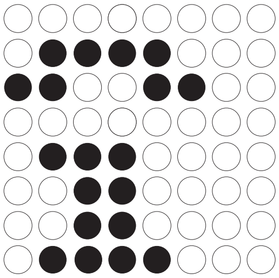
CHR137



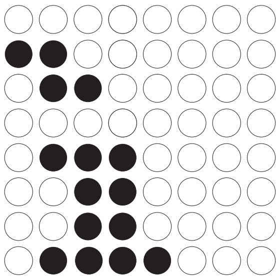
CHR138



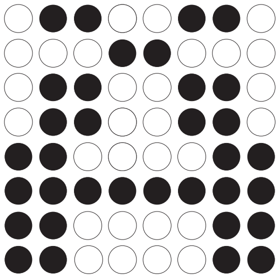
CHR139



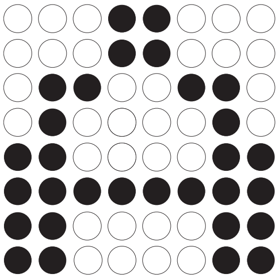
CHR140



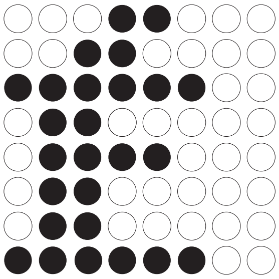
CHR141



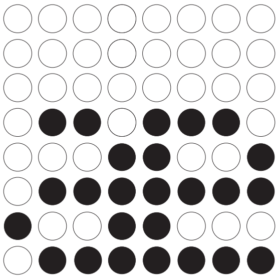
CHR142



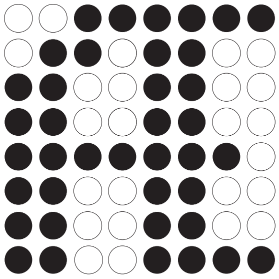
CHR143



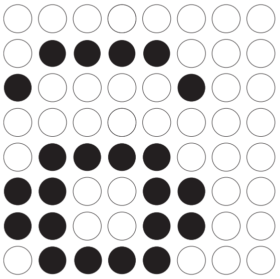
CHR144



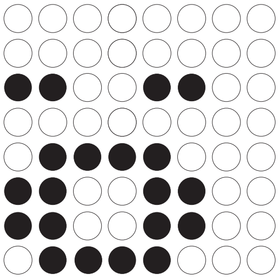
CHR145



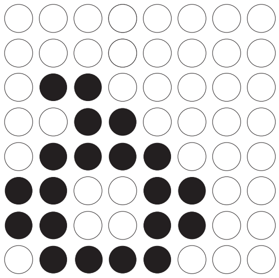
CHR146



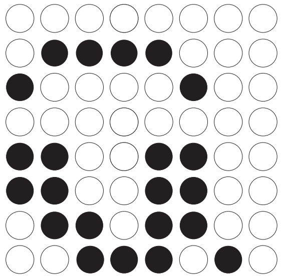
CHR147



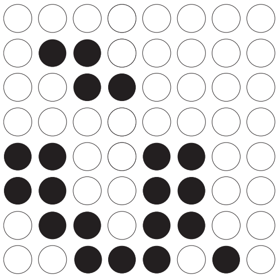
CHR148



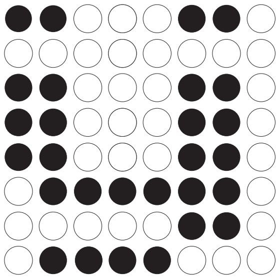
CHR149



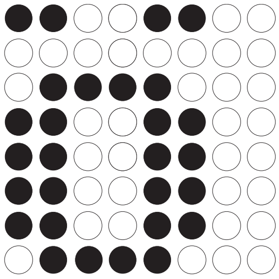
CHR150



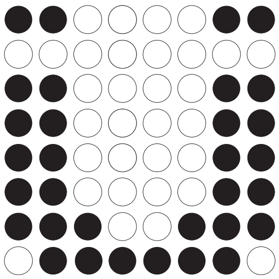
CHR151



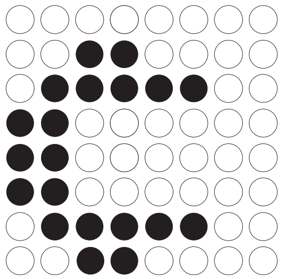
CHR152



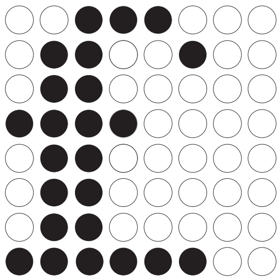
CHR153



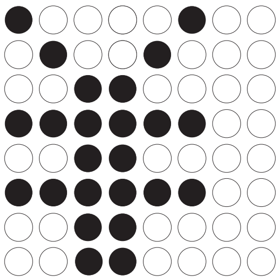
CHR154



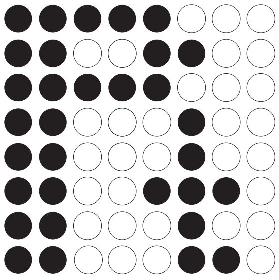
CHR155



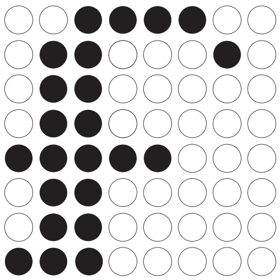
CHR156



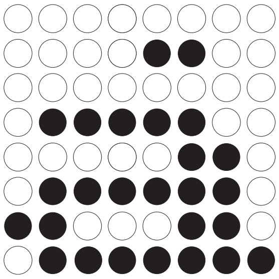
CHR157

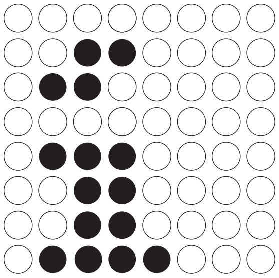


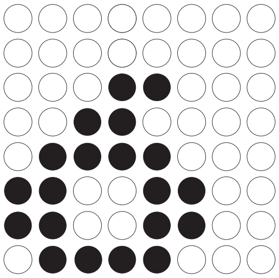
CHR158

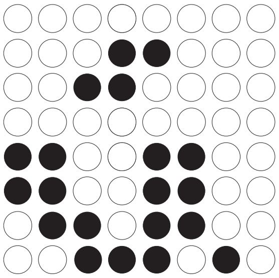


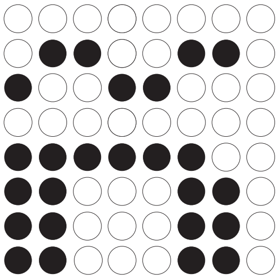
CHR159

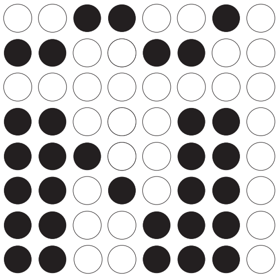

CHR160

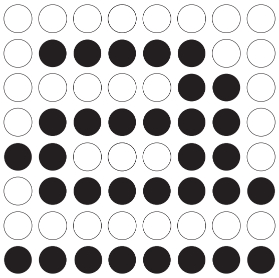

CHR161

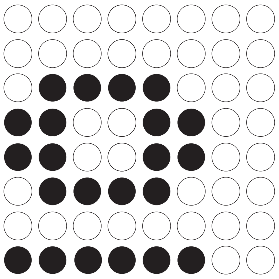

CHR162

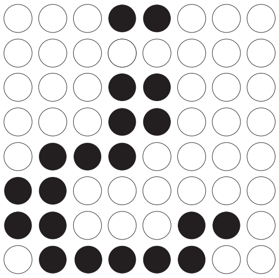

CHR163

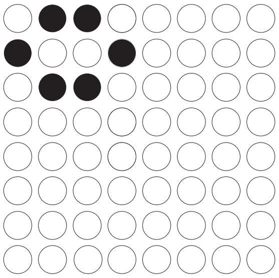

CHR164

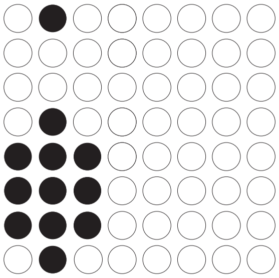

CHR165

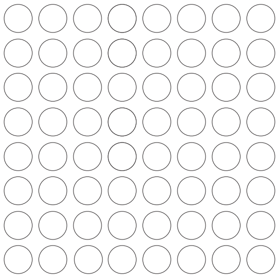

CHR166

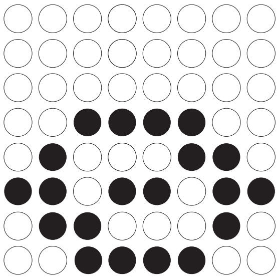

CHR167

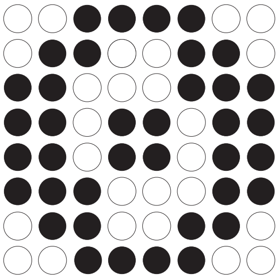

CHR168

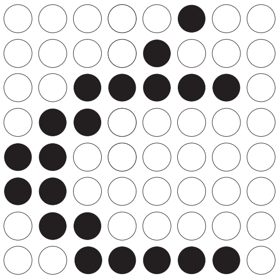

CHR169

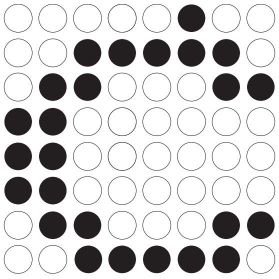

CHR170

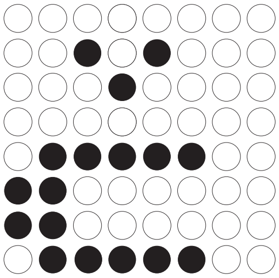

CHR171

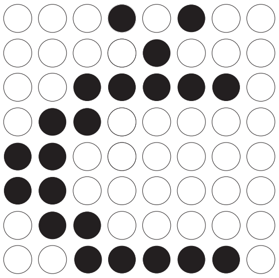

CHR172

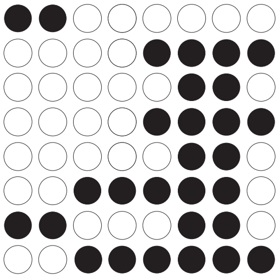

CHR173

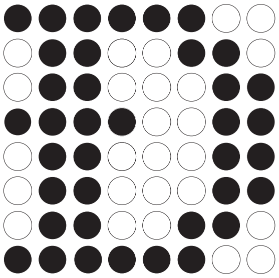

CHR174

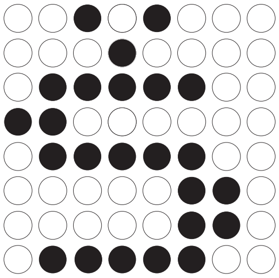

CHR175

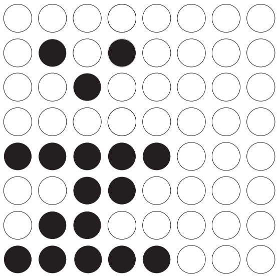

CHR176

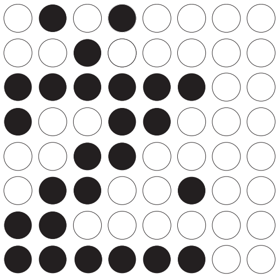

CHR177

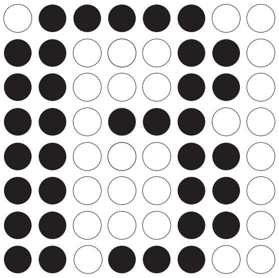

CHR178

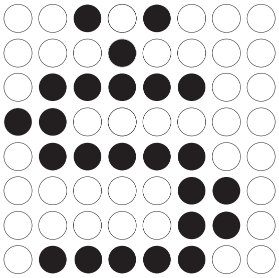

CHR179

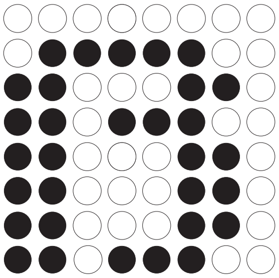

CHR180

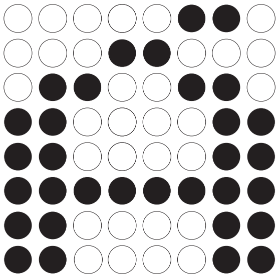

CHR181

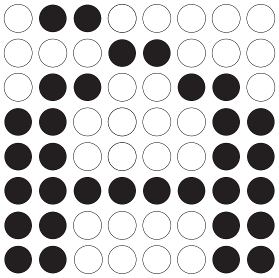

CHR1826

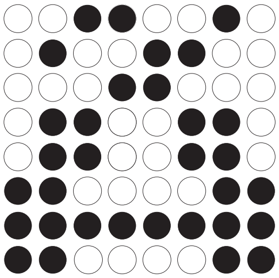

CHR183

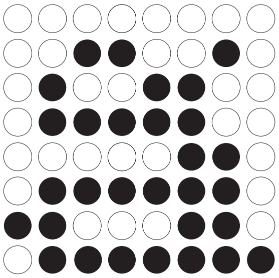

CHR184

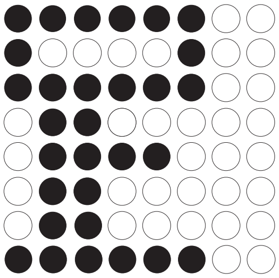

CHR185

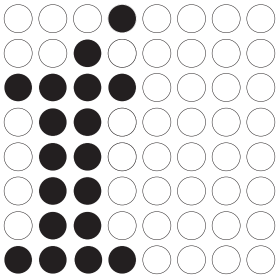

CHR186

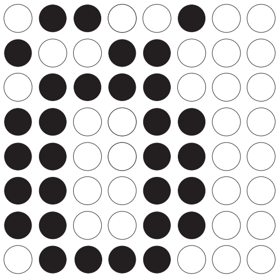

CHR187

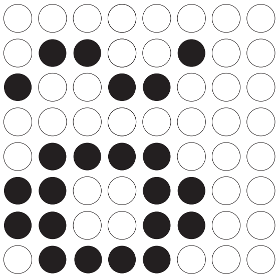

CHR188


CHR189

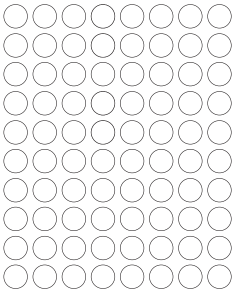

CHR190

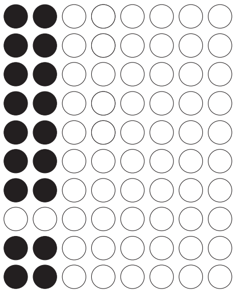

CHR191

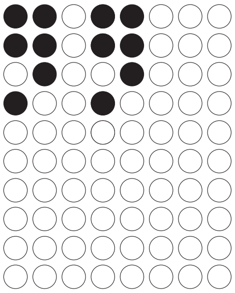

CHR192

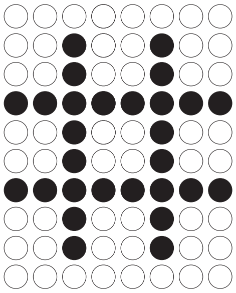

CHR193

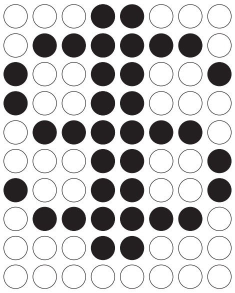
7.13.9 10-High Fancy (SF10)

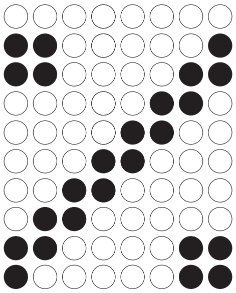

CHR032

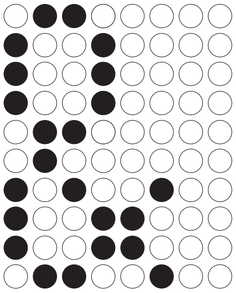

CHR033

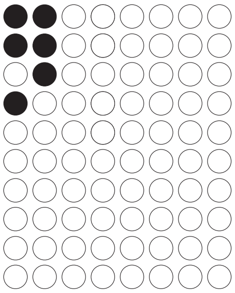

CHR034

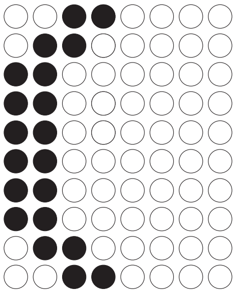

CHR035

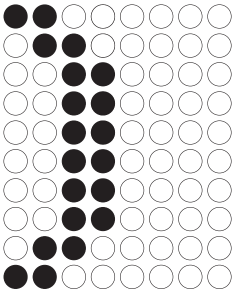

CHR036

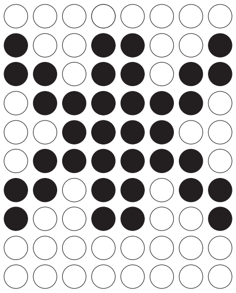

CHR037

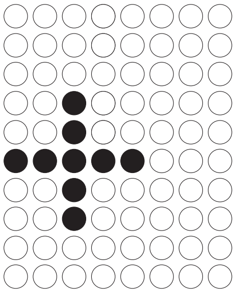

CHR038

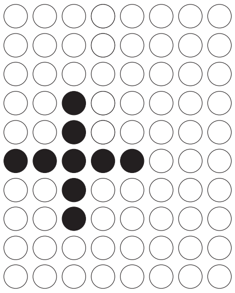

CHR039

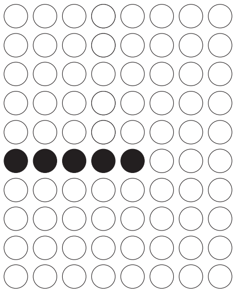

CHR040

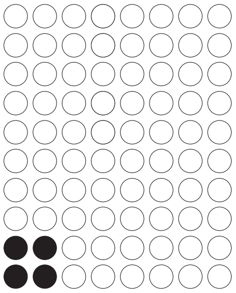

CHR041

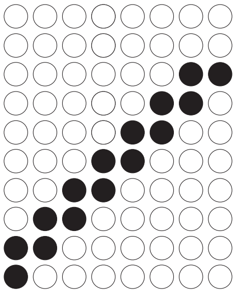

CHR042

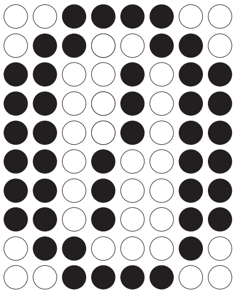

CHR043

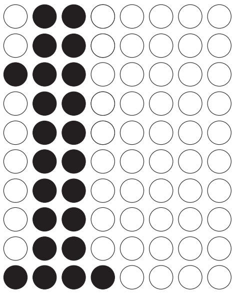

CHR044

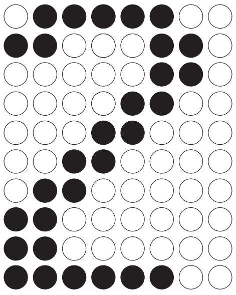

CHR045

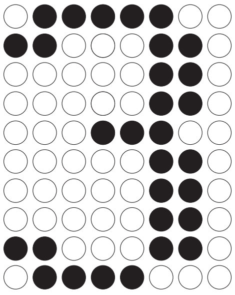

CHR046

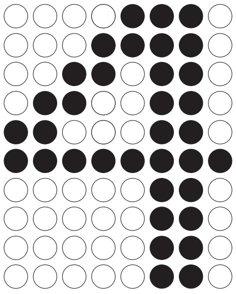

CHR047

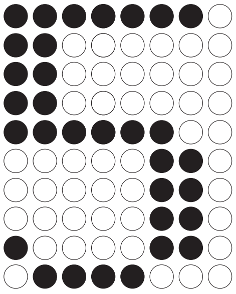

CHR048

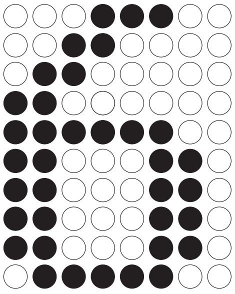

CHR049

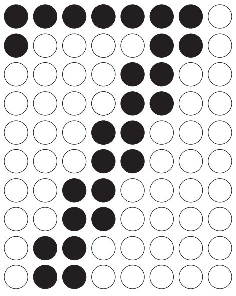

CHR050

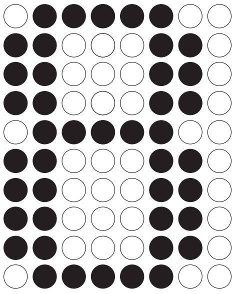

CHR051

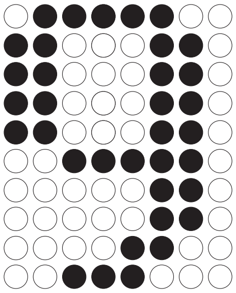

CHR052

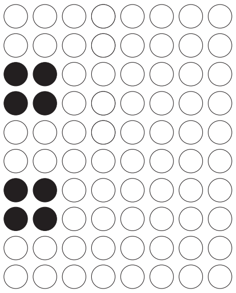

CHR053

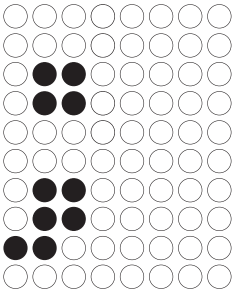

CHR054

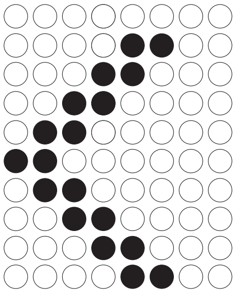

CHR055

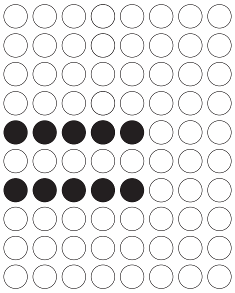

CHR056

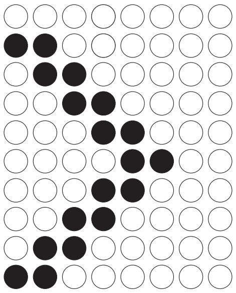

CHR057

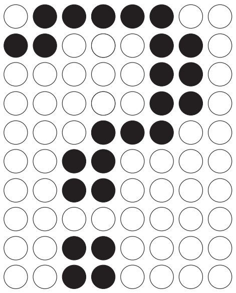

CHR058

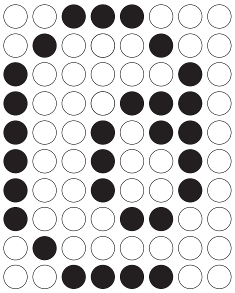

CHR059

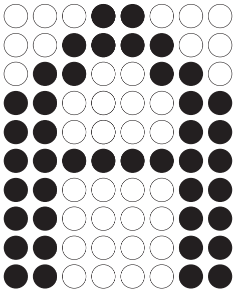

CHR060

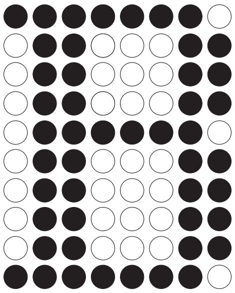

CHR061

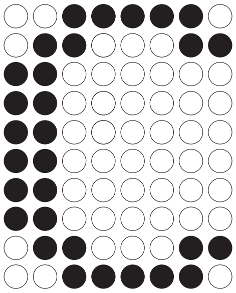

CHR062

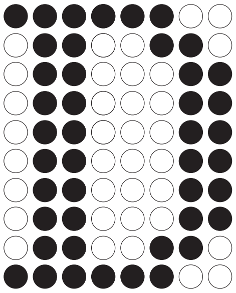

CHR063

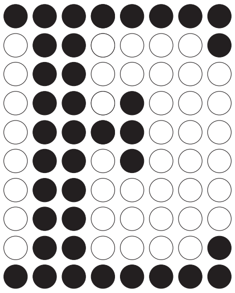

CHR064

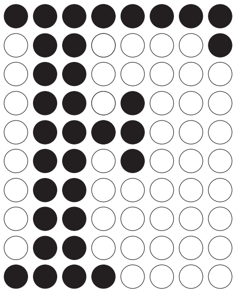

CHR065

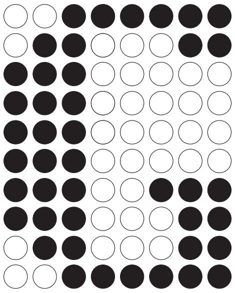

CHR066

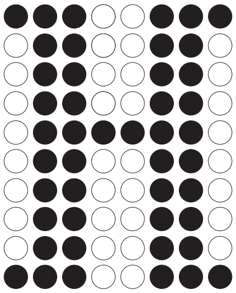

CHR067

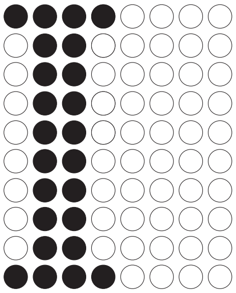

CHR068

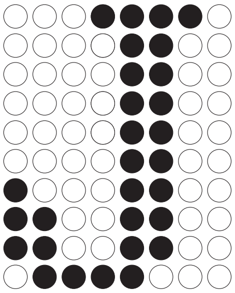

CHR069

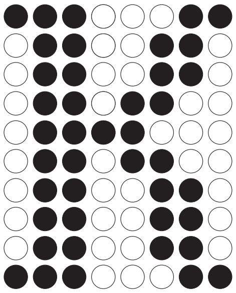

CHR070

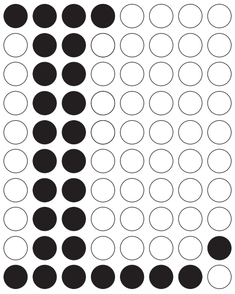

CHR071

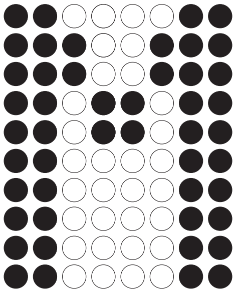

CHR072

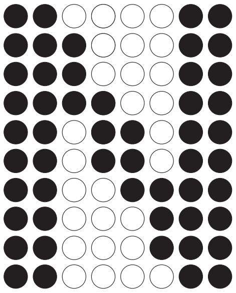

CHR073

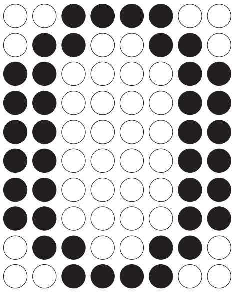

CHR074

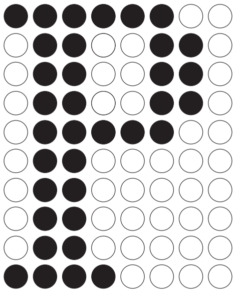

CHR075

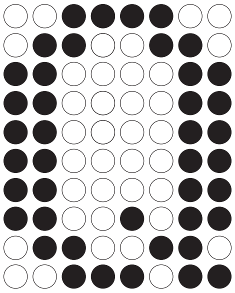

CHR076

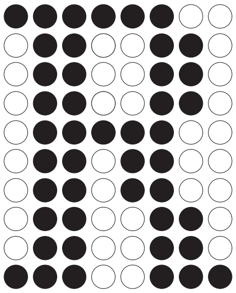

CHR077

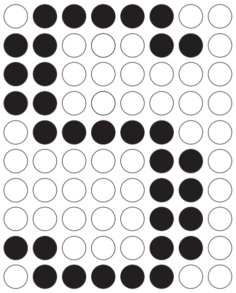

CHR078

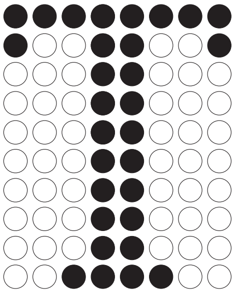

CHR079

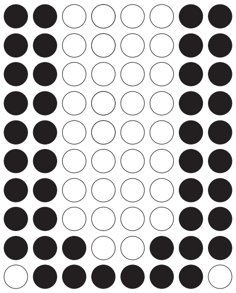

CHR080

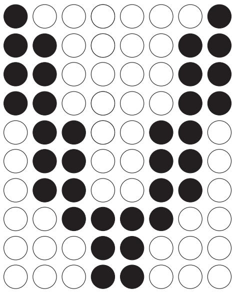

CHR081

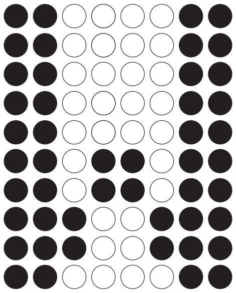

CHR082

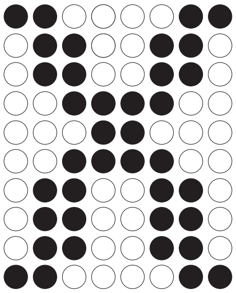

CHR083

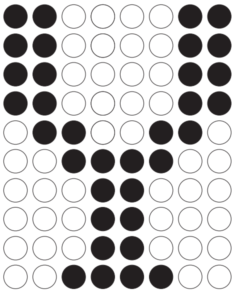

CHR084

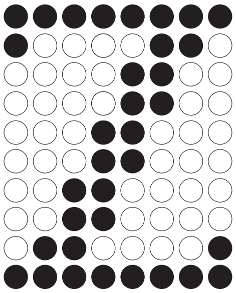

CHR085

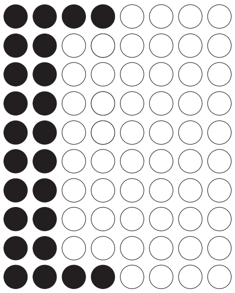

CHR086

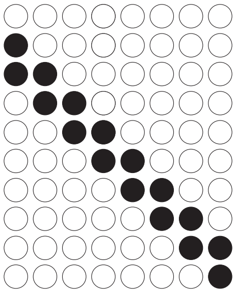

CHR087

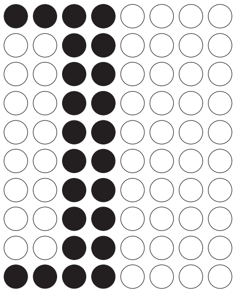

CHR088

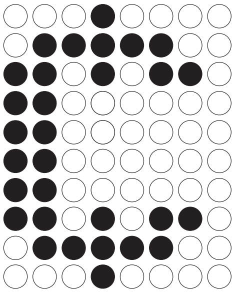

CHR089

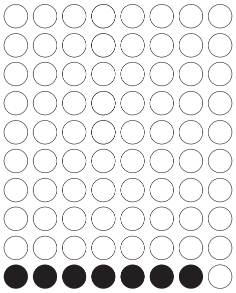

CHR090

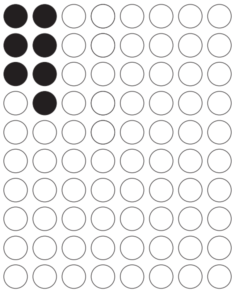

CHR091

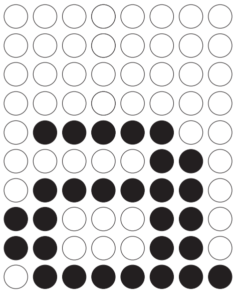

CHR092

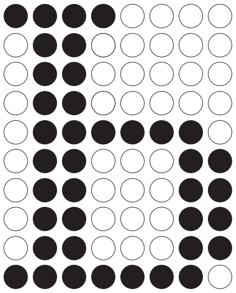

CHR093

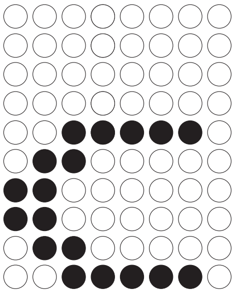

CHR094

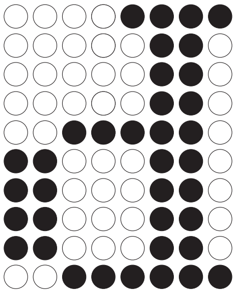

CHR095

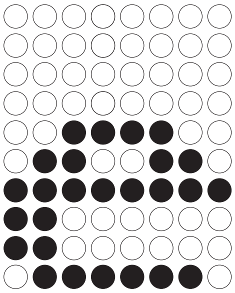

CHR096

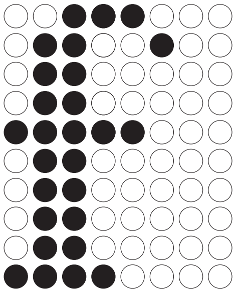

CHR097

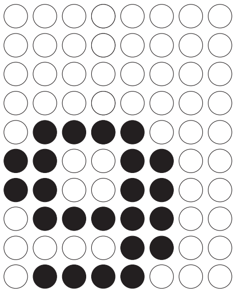

CHR098

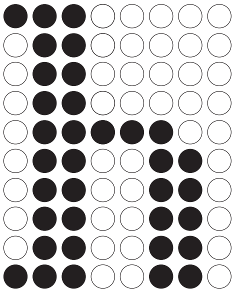

CHR099

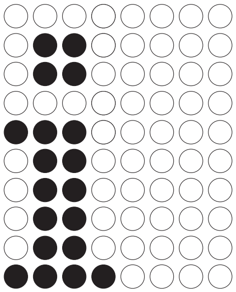

CHR100

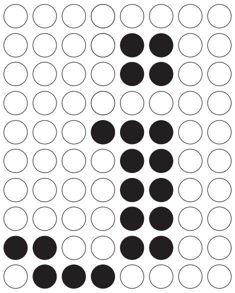

CHR101

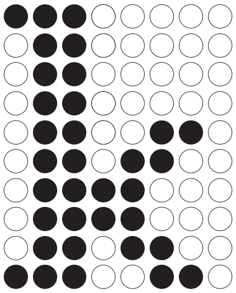

CHR102

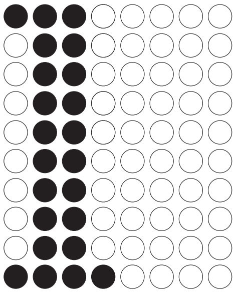

CHR103

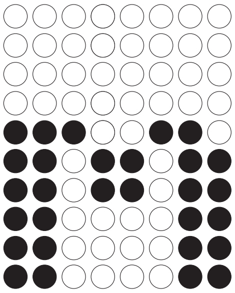

CHR104

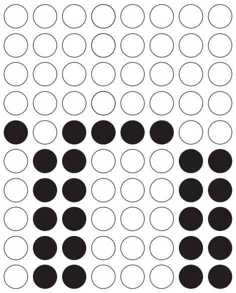

CHR105

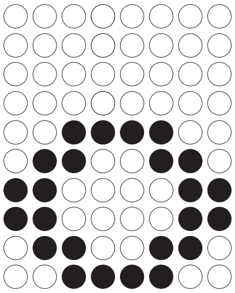

CHR106

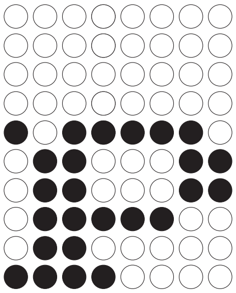

CHR107

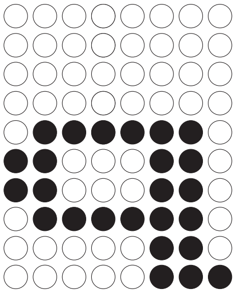

CHR108

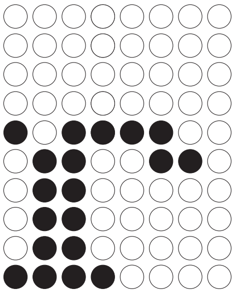

CHR109

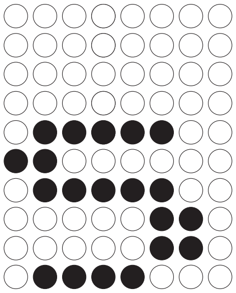

CHR110

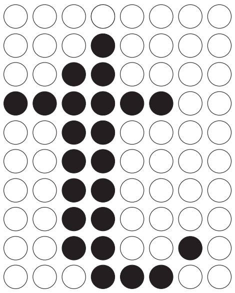

CHR111

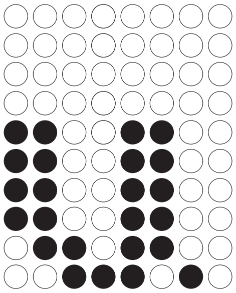

CHR112

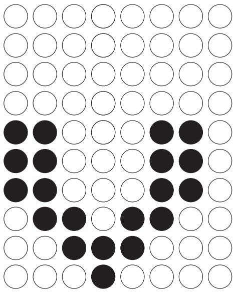

CHR113

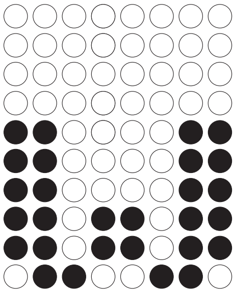

CHR114

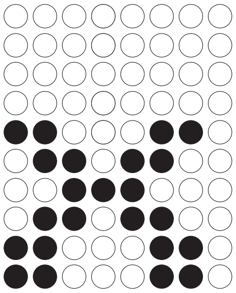

CHR115

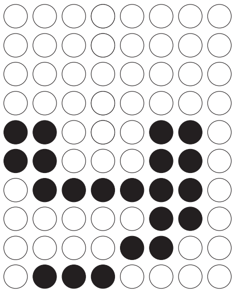

CHR116

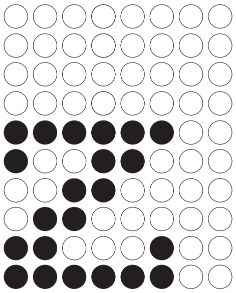

CHR117

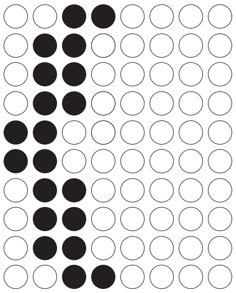

CHR118

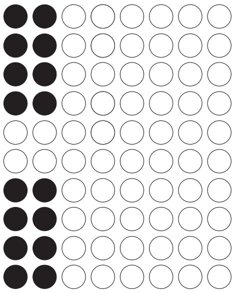

CHR119

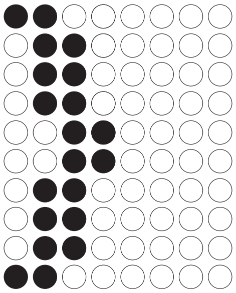

CHR120

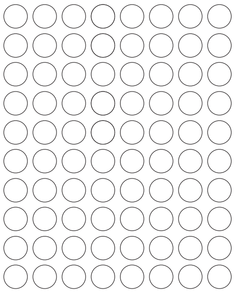

CHR121

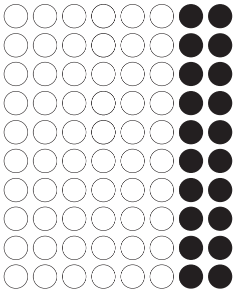

CHR122

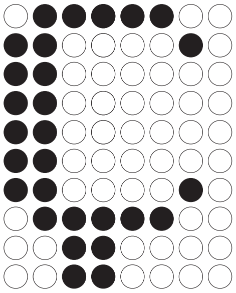

CHR123

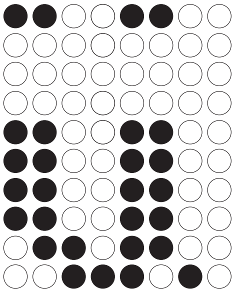

CHR124

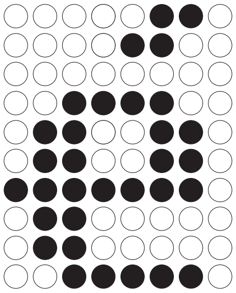

CHR125

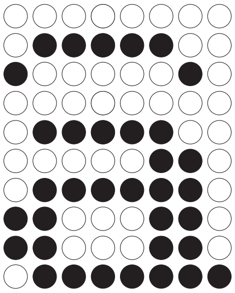

CHR126

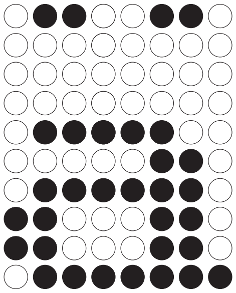

CHR127

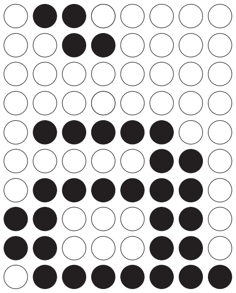

CHR128

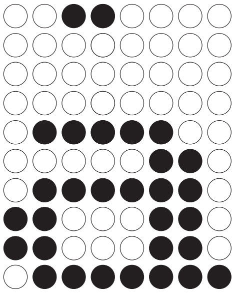

CHR129

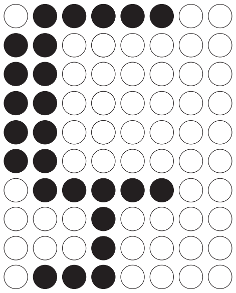

CHR130

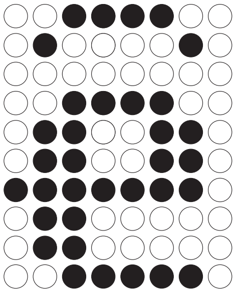

CHR131

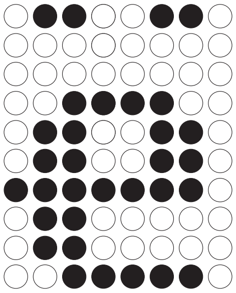

CHR132

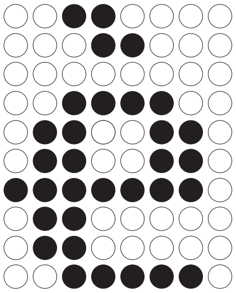

CHR133

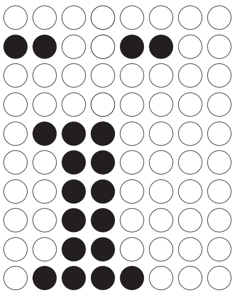

CHR134

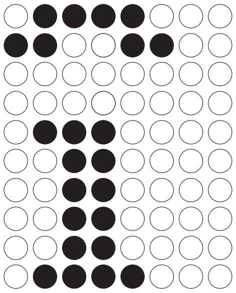

CHR135

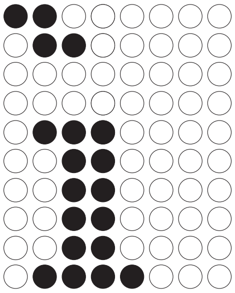

CHR136

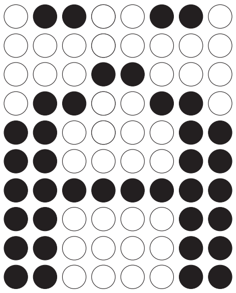

CHR137

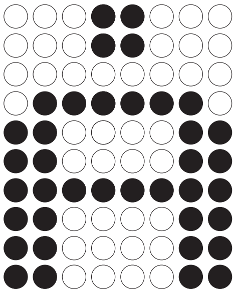

CHR138

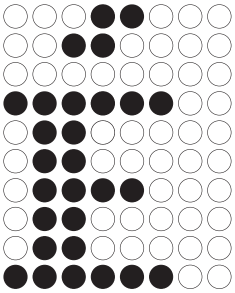

CHR139

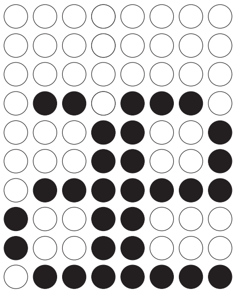

CHR140

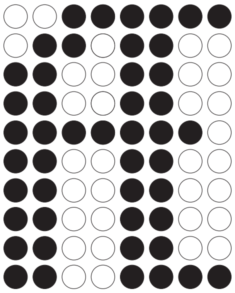

CHR141

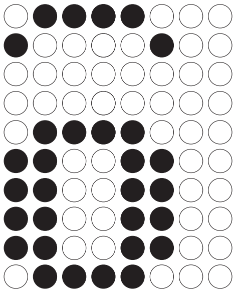

CHR142

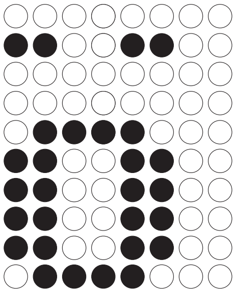

CHR143

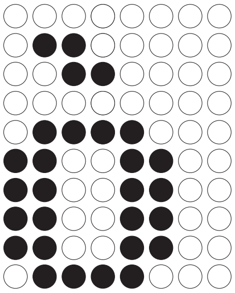

CHR144

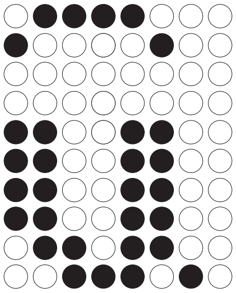

CHR145

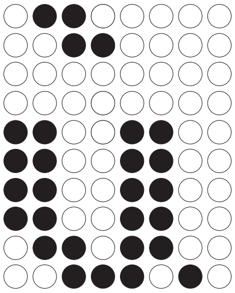

CHR146

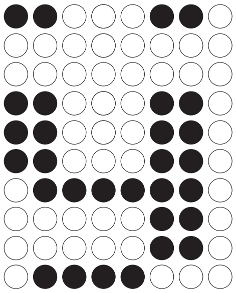

CHR147

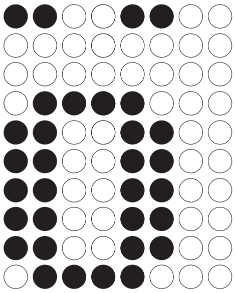

CHR148

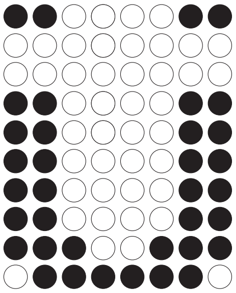

CHR149

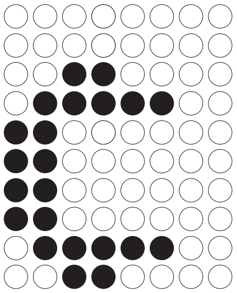

CHR150

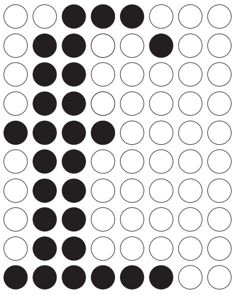

CHR151

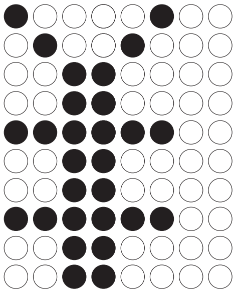

CHR152

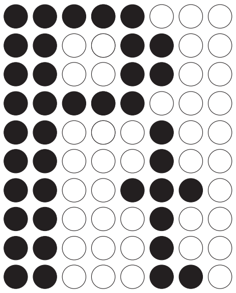

CHR153

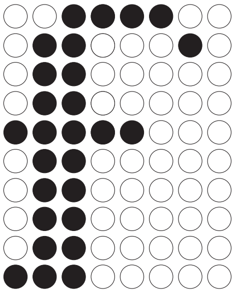

CHR154

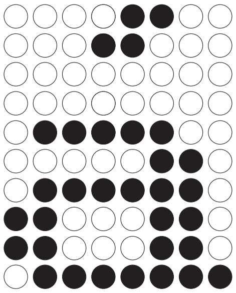

CHR155

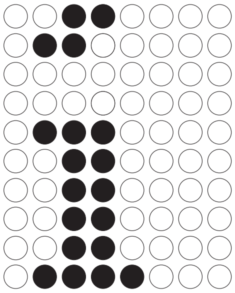

CHR156

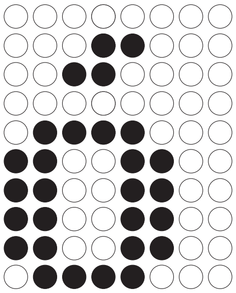

CHR157

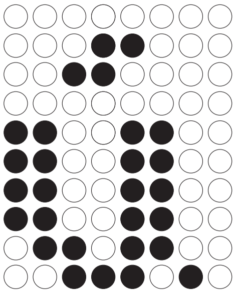

CHR158

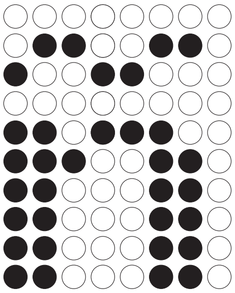

CHR159

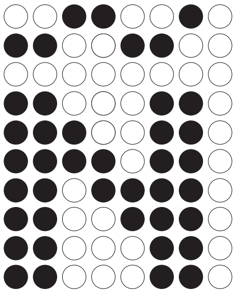

CHR160

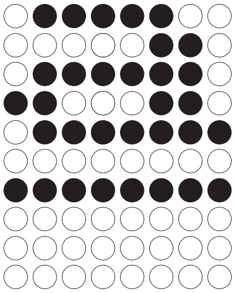

CHR161

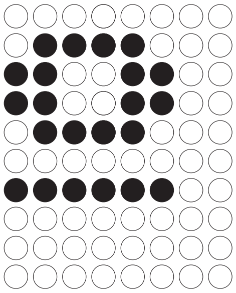

CHR162

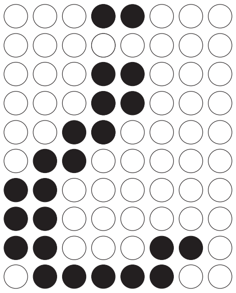

CHR163

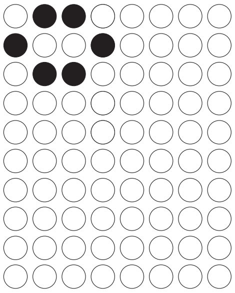

CHR164

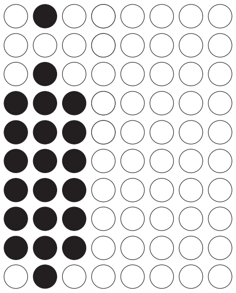

CHR165

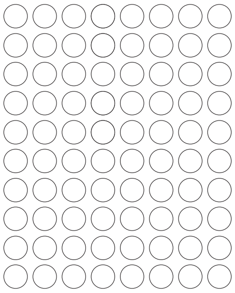

CHR166

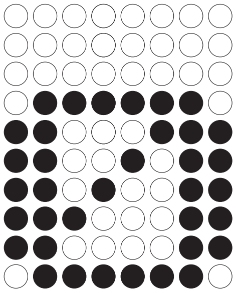

CHR167

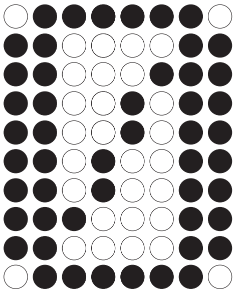

CHR168

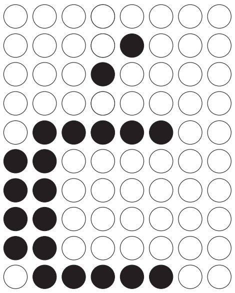

CHR169

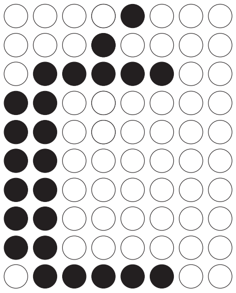

CHR170

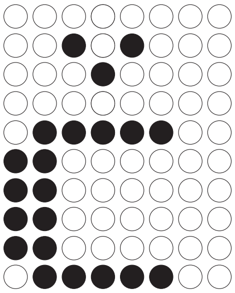

CHR171

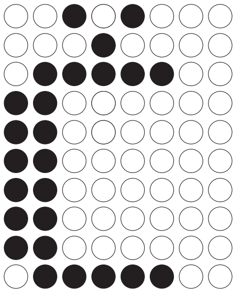

CHR172

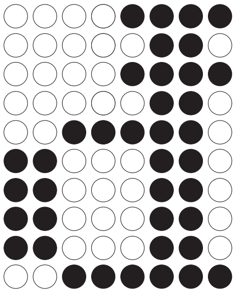

CHR173

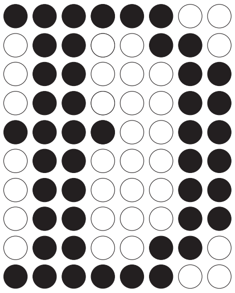

CHR174

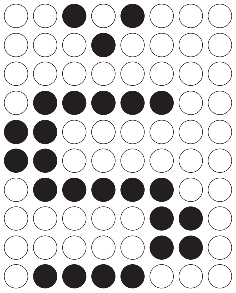

CHR175

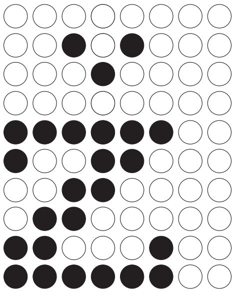

CHR176

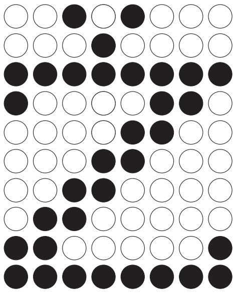

CHR177

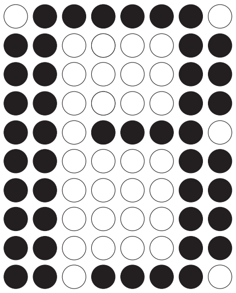

CHR178

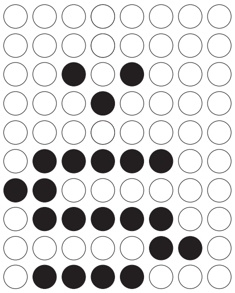

CHR179

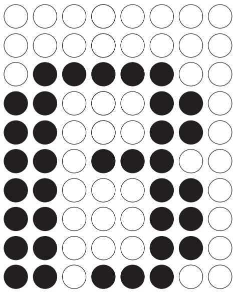

CHR180

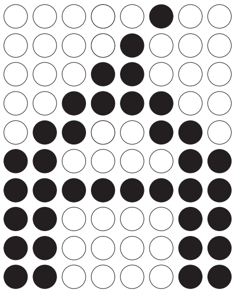

CHR181

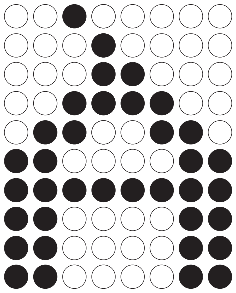

CHR182

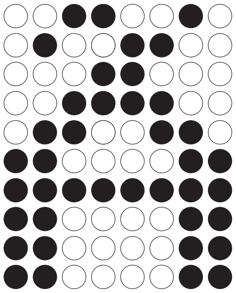

CHR183

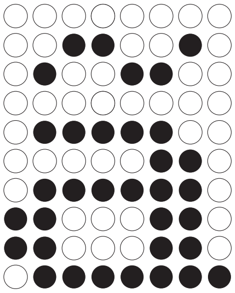

CHR184

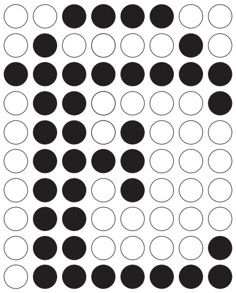

CHR185

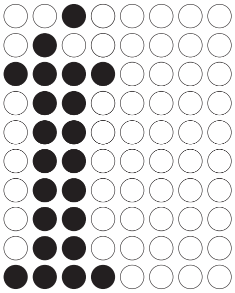

CHR186

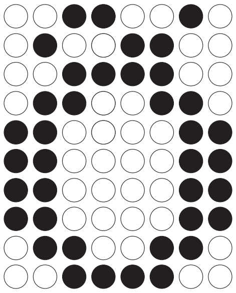

CHR187

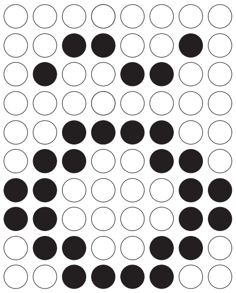

CHR188


CHR189

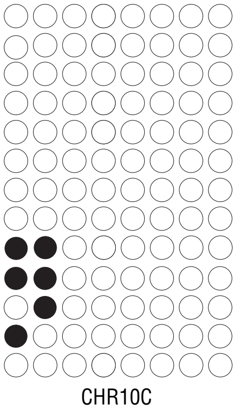

CHR190

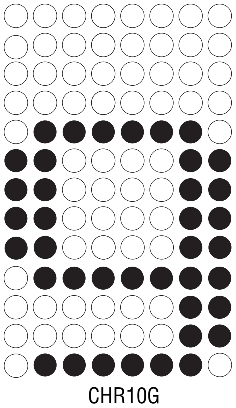

CHR191

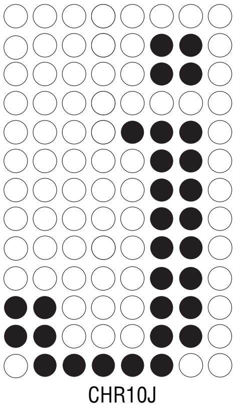

CHR192

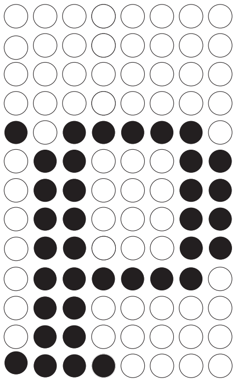

CHR193

7.13.10 10-High True Descender Fancy

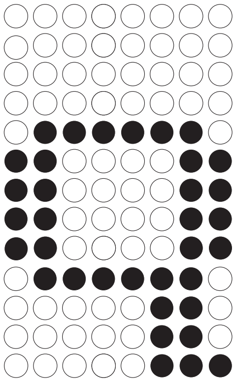

CHR10C


CHR10G

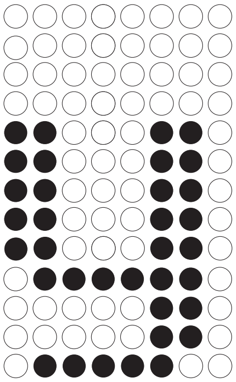

CHR10J



CHR10P

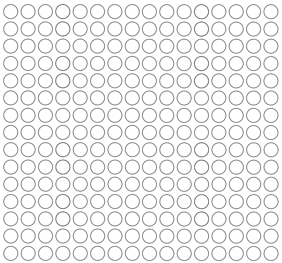


CHR10Q

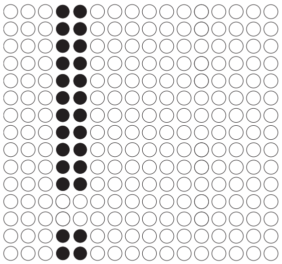


CHR10Y

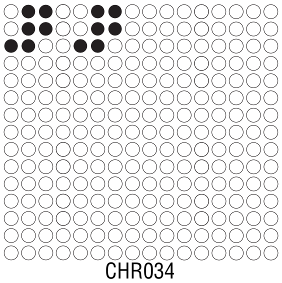
7.13.11 15-High Regular (SS15)



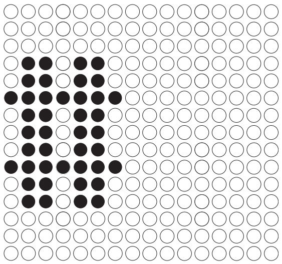
CHR032



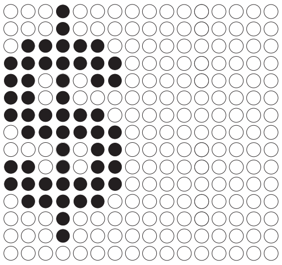
CHR033



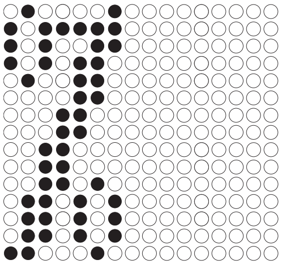
CHR034



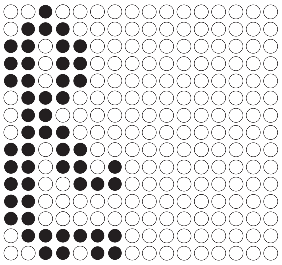
CHR035



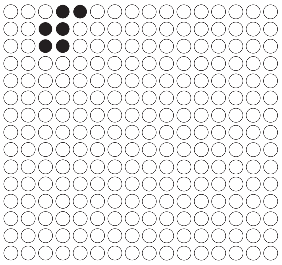
CHR036



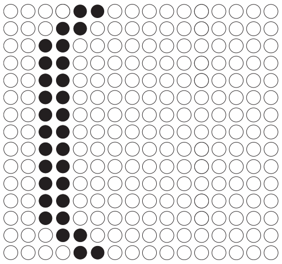
CHR037



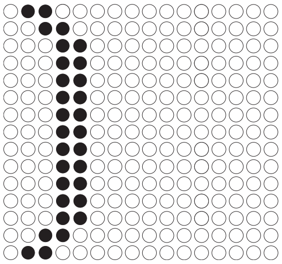
CHR038



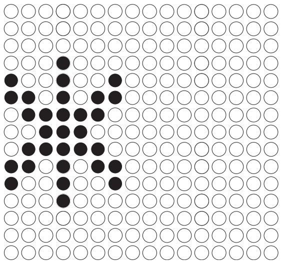
CHR039



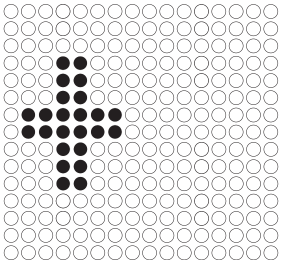
CHR040



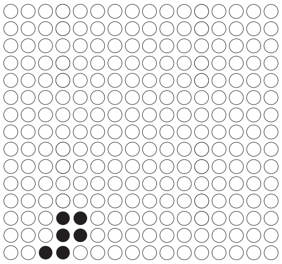
CHR041



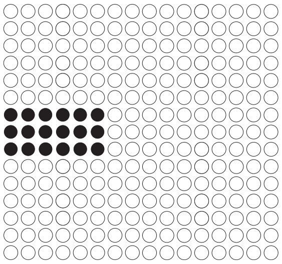
CHR042



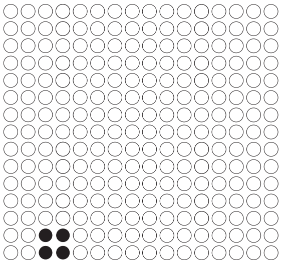
CHR043



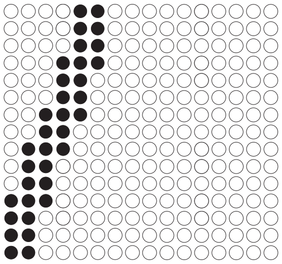
CHR044



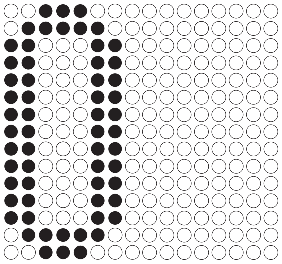
CHR045



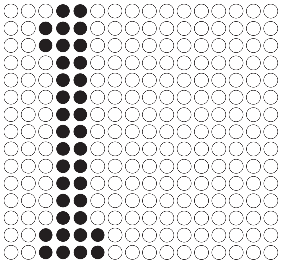
CHR046



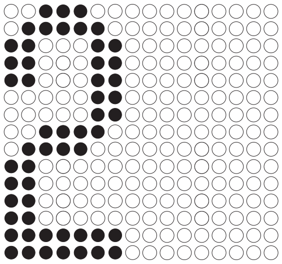
CHR047



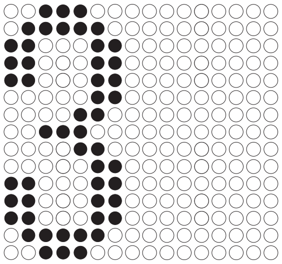
CHR048



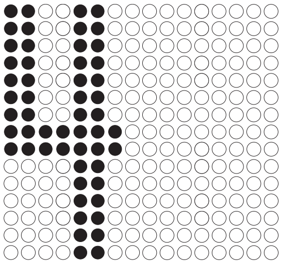
CHR049



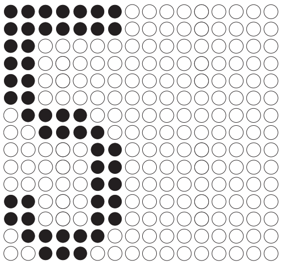
CHR050



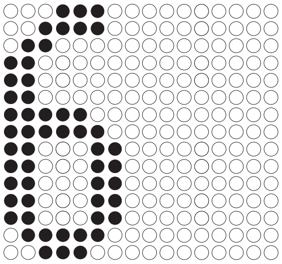
CHR051



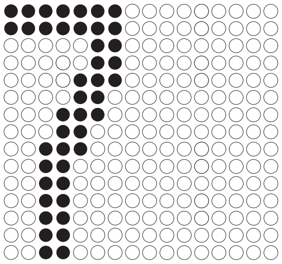
CHR052



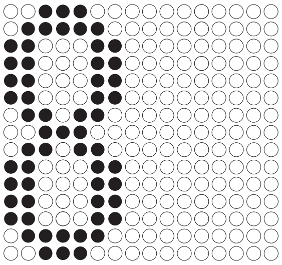
CHR053



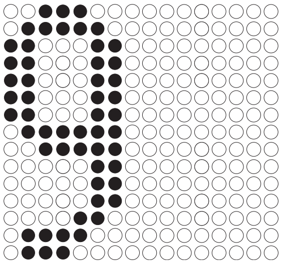
CHR054



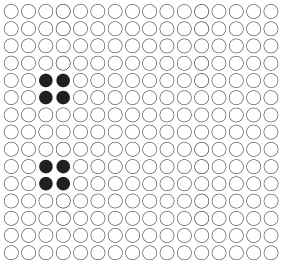
CHR055



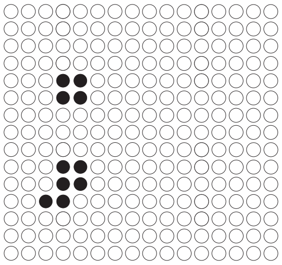
CHR056



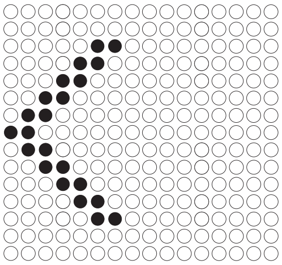
CHR057



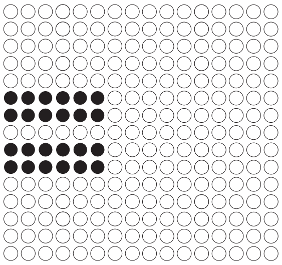
CHR058



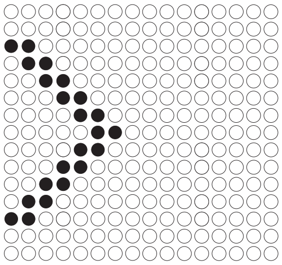
CHR059



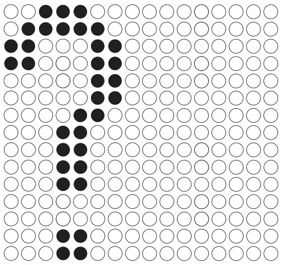
CHR060



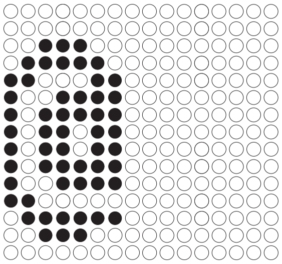
CHR061



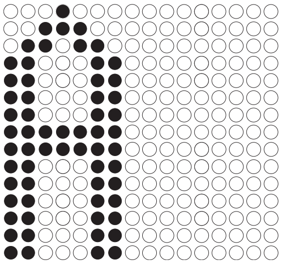
CHR062



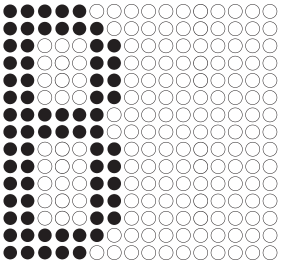
CHR063



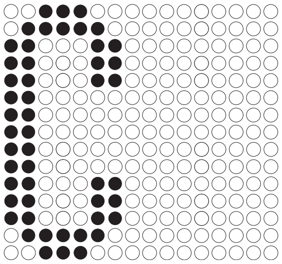
CHR064



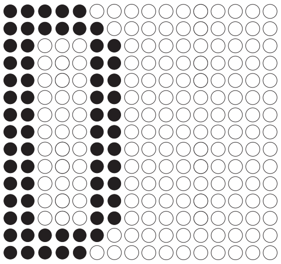
CHR065

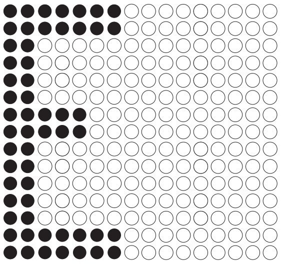


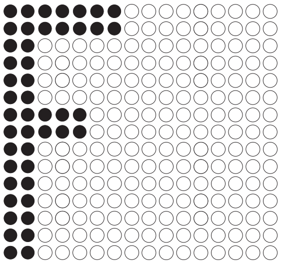
CHR066

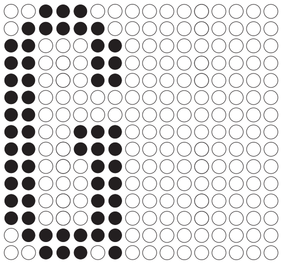


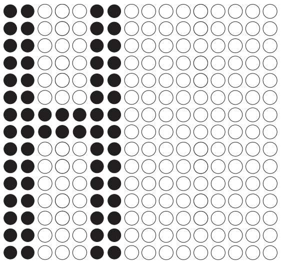
CHR067

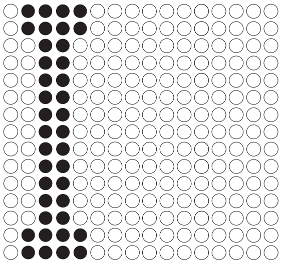

CHR068

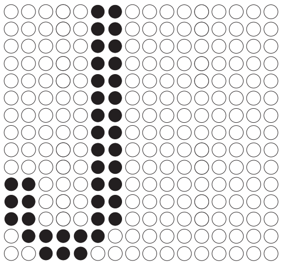

CHR069

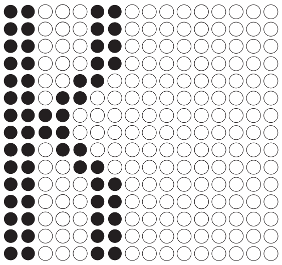

CHR070

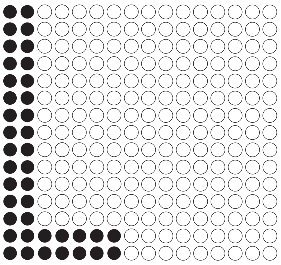

CHR071

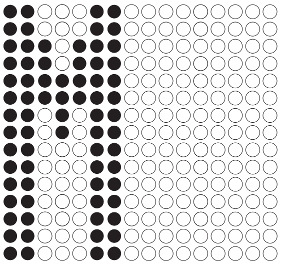

CHR072

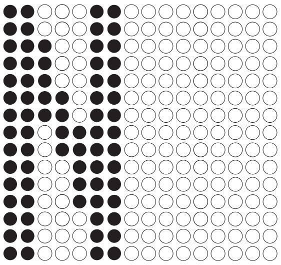

CHR073

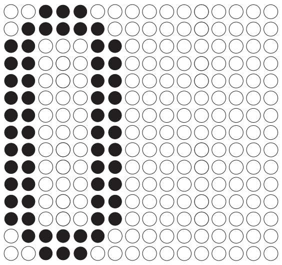

CHR074

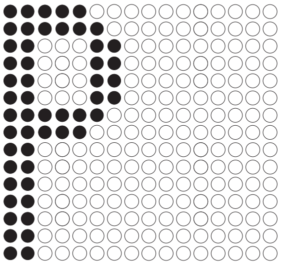

CHR075

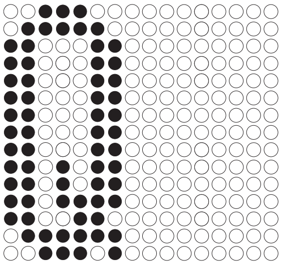

CHR076

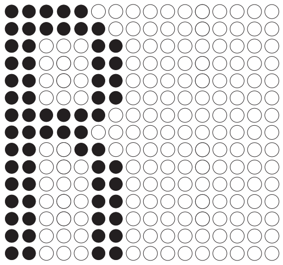

CHR077

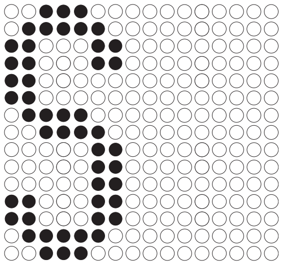

CHR078

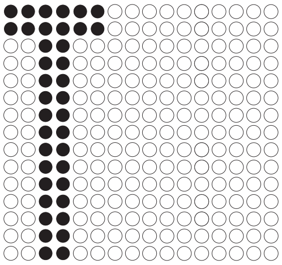

CHR079

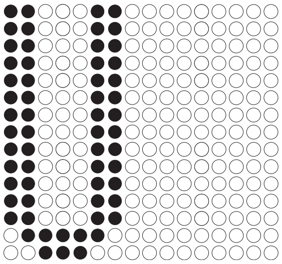

CHR080

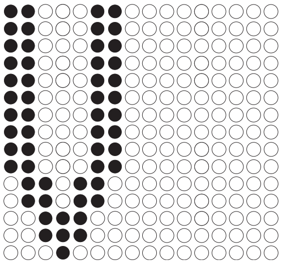

CHR081

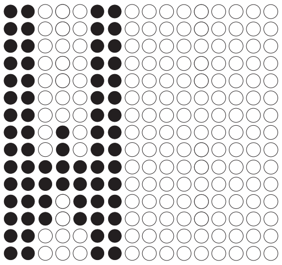

CHR082

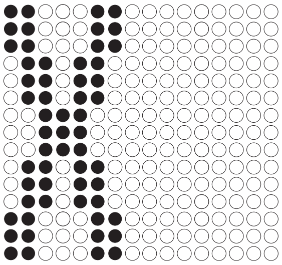

CHR083


CHR084

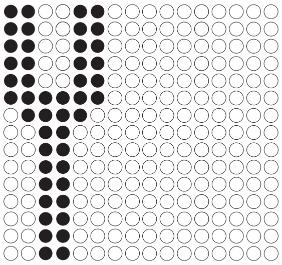

CHR085


CHR086

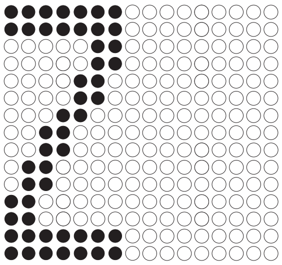

CHR087



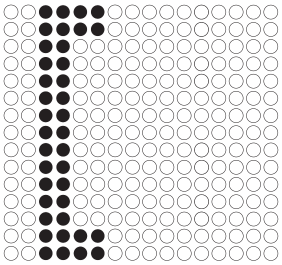
CHR088



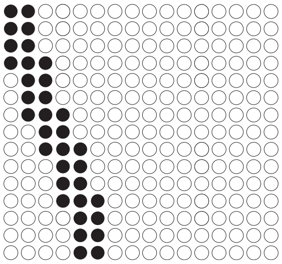
CHR089



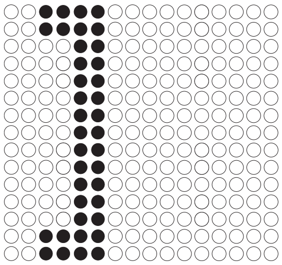
CHR090



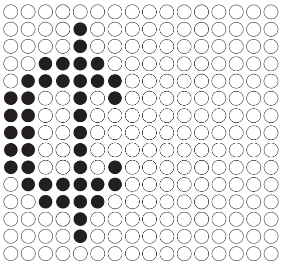
CHR091



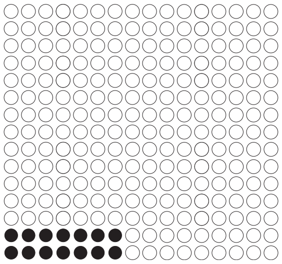
CHR092



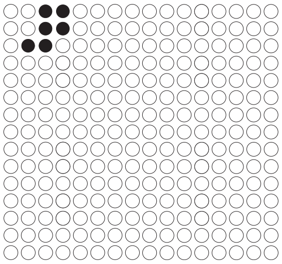
CHR093



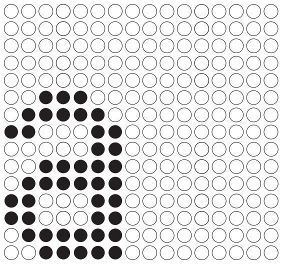
CHR094



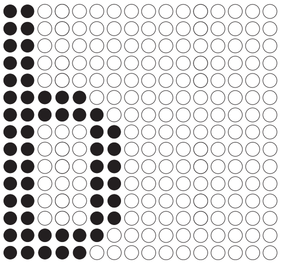
CHR095



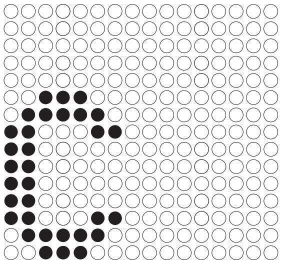
CHR096



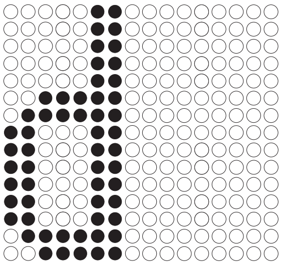
CHR097



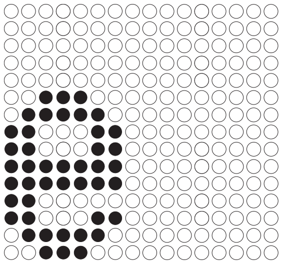
CHR098



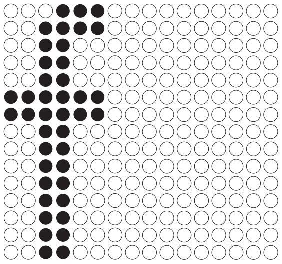
CHR099



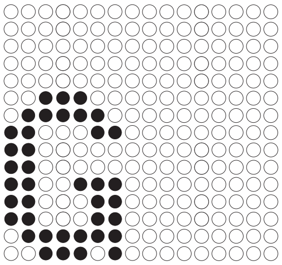
CHR100



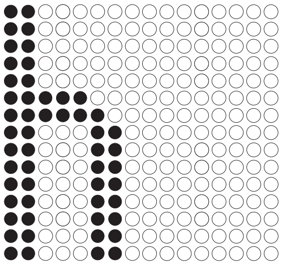
CHR101



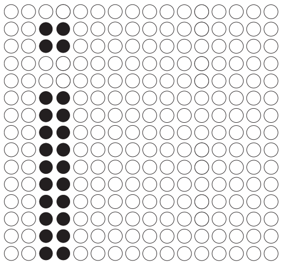
CHR102



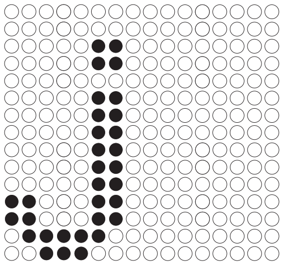
CHR103



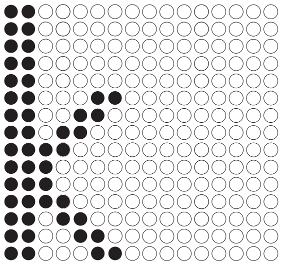
CHR104



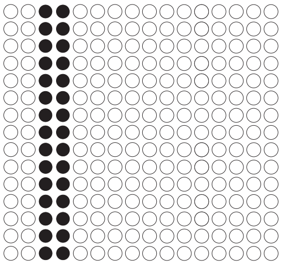
CHR105



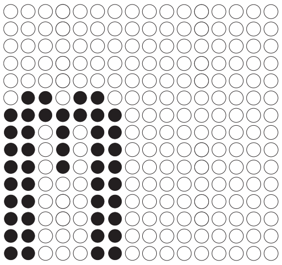
CHR106



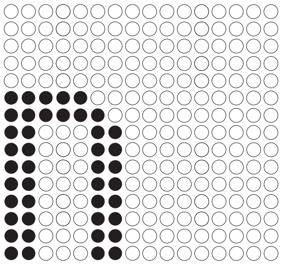
CHR107



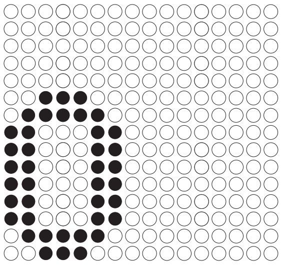
CHR108



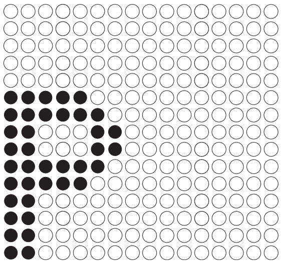
CHR109



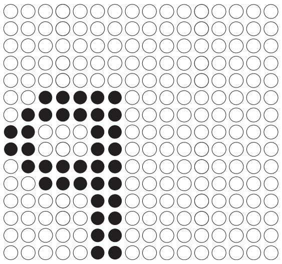
CHR110



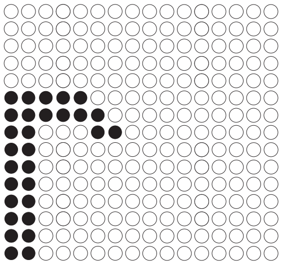
CHR111



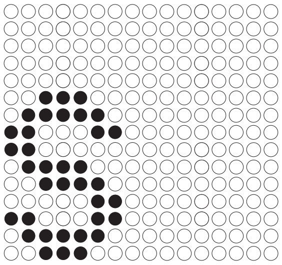
CHR112



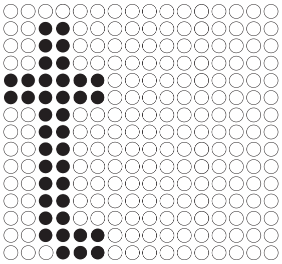
CHR113



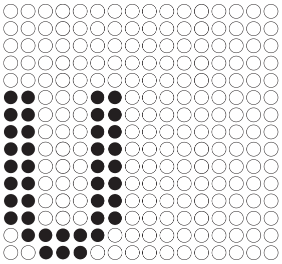
CHR114



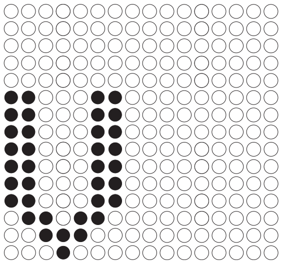
CHR115



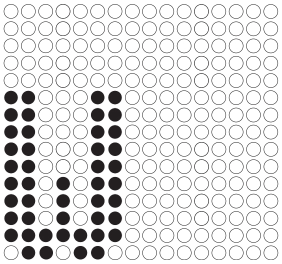
CHR116



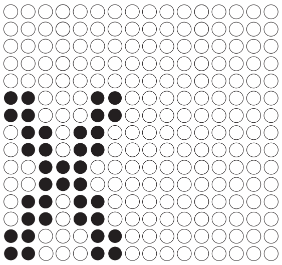
CHR117



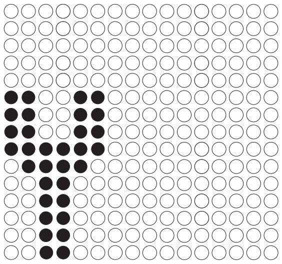
CHR118



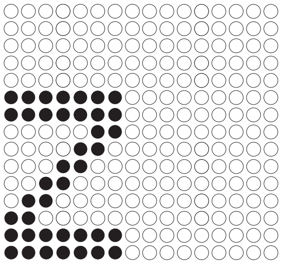
CHR119



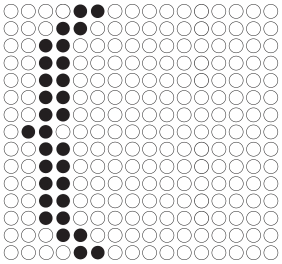
CHR120



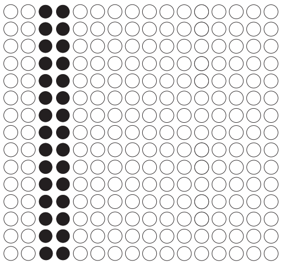
CHR121



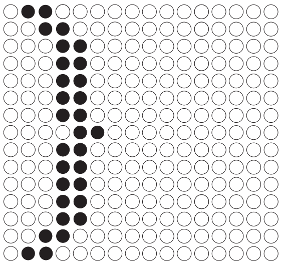
CHR122



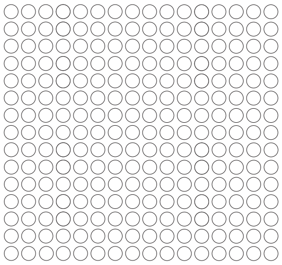
CHR123



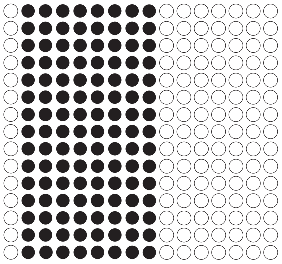
CHR124



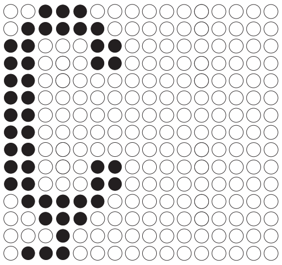
CHR125



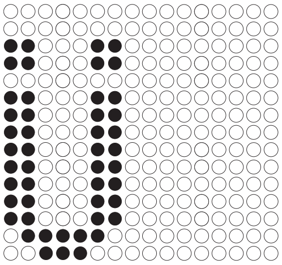
CHR126



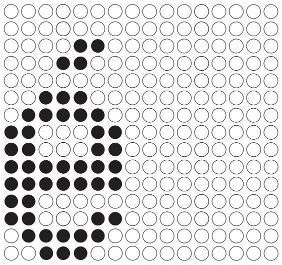
CHR127



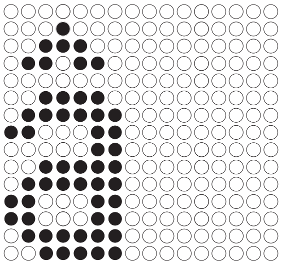
CHR128



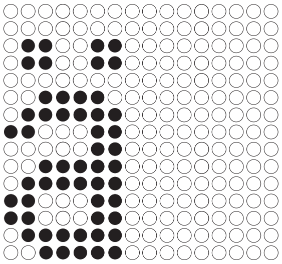
CHR129



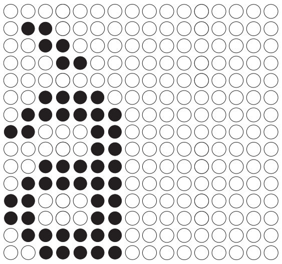
CHR130



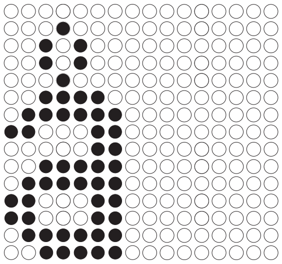
CHR131



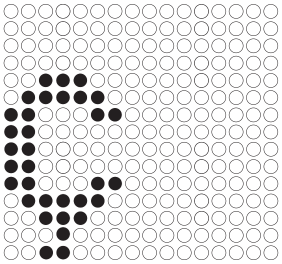
CHR132



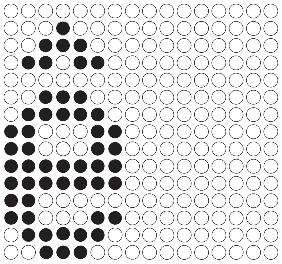
CHR133



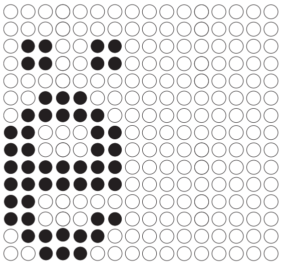
CHR134



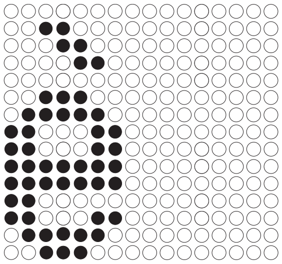
CHR135



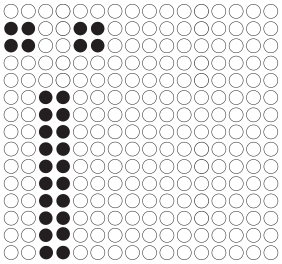
CHR136



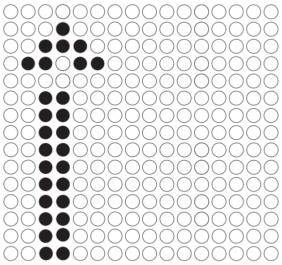
CHR137



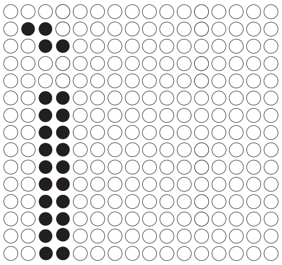
CHR138



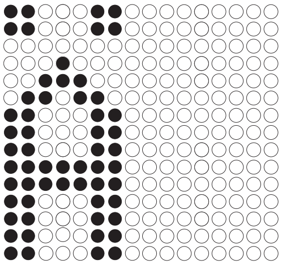
CHR139



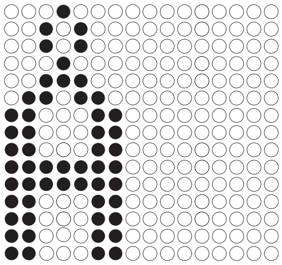
CHR140



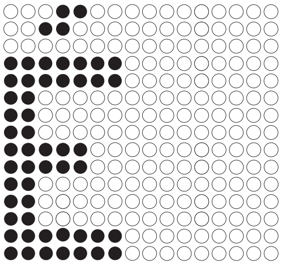
CHR141



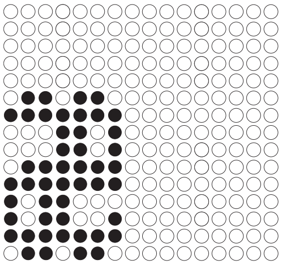
CHR142



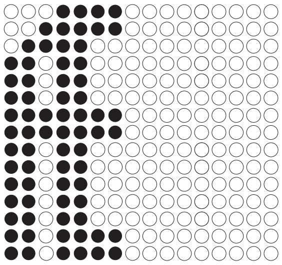
CHR143



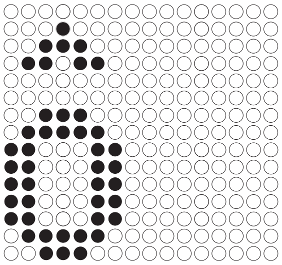
CHR144



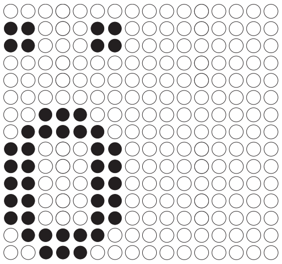
CHR145



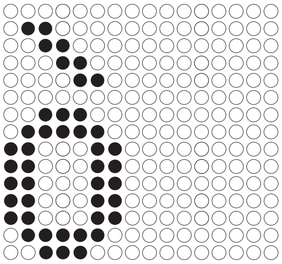
CHR146



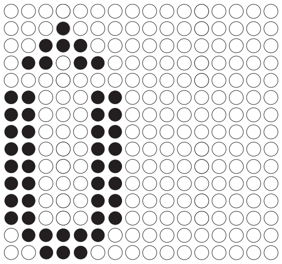
CHR147



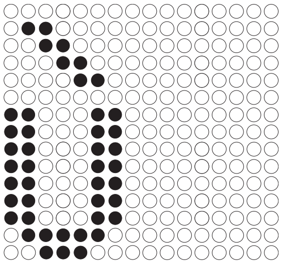
CHR148



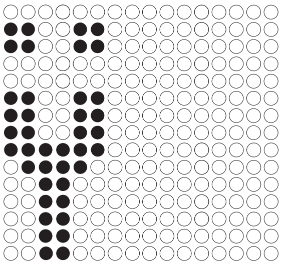
CHR149



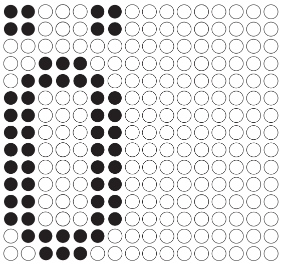
CHR150



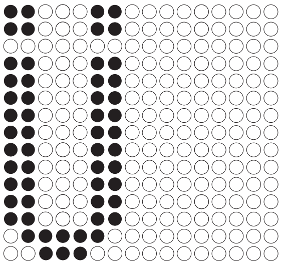
CHR151



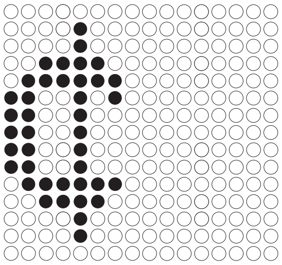
CHR152



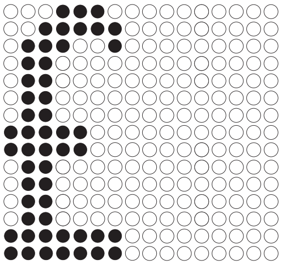
CHR153



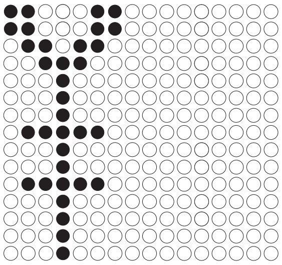
CHR154



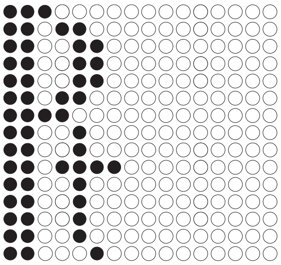
CHR155



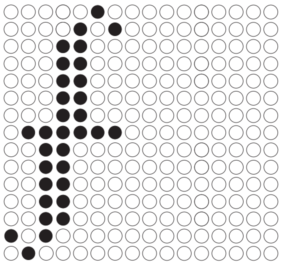
CHR156



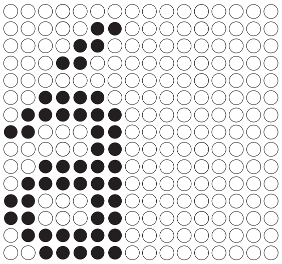
CHR157



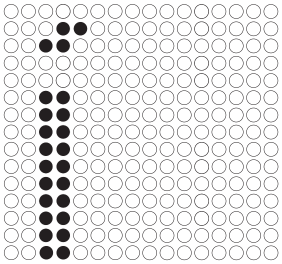
CHR158



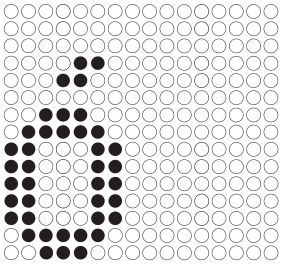
CHR159



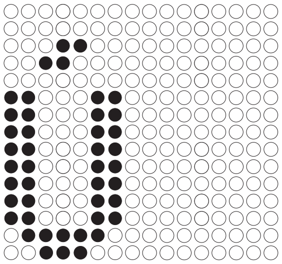
CHR160



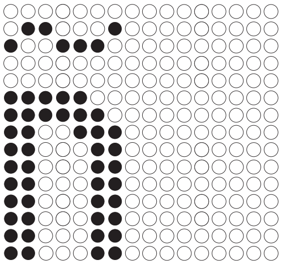
CHR161



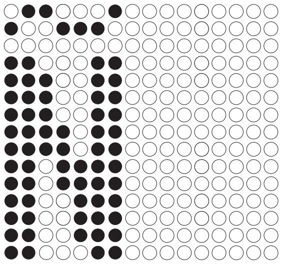
CHR162



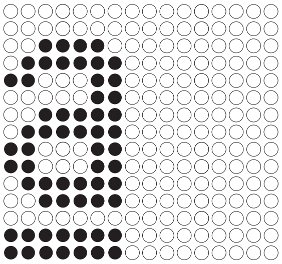
CHR163



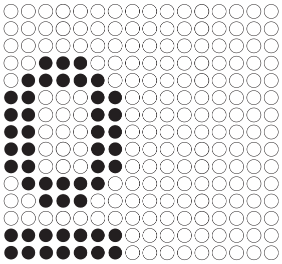
CHR164



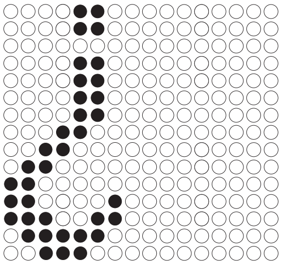
CHR165

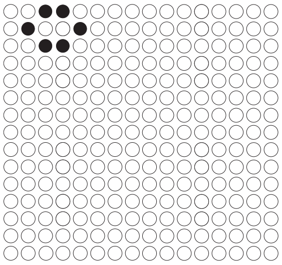


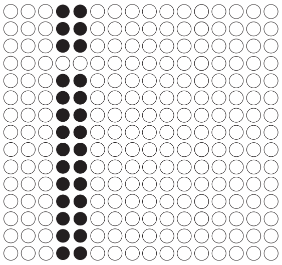
CHR166

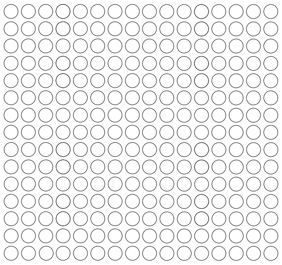


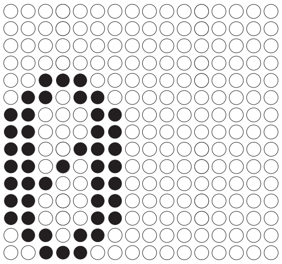
CHR167

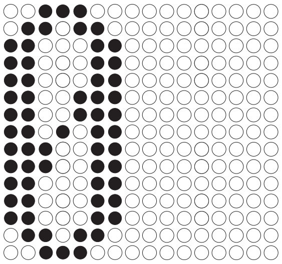

CHR168

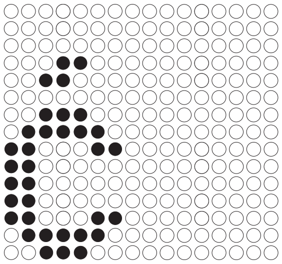

CHR169

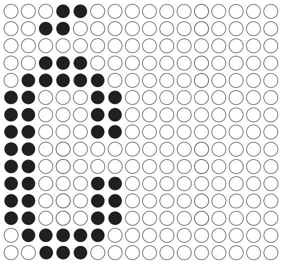

CHR170

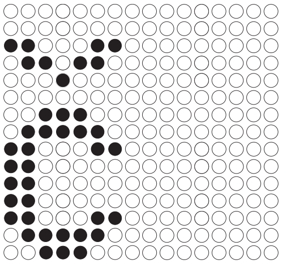

CHR171

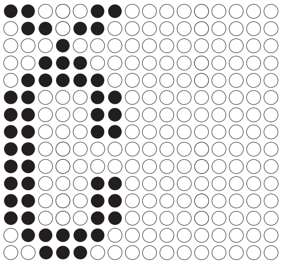

CHR172

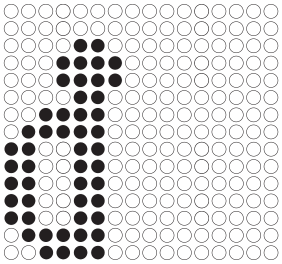

CHR173

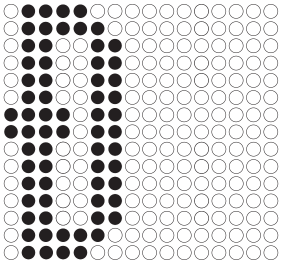

CHR174

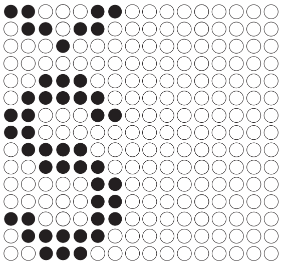

CHR175

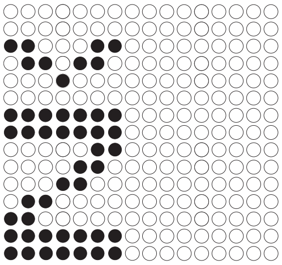

CHR176

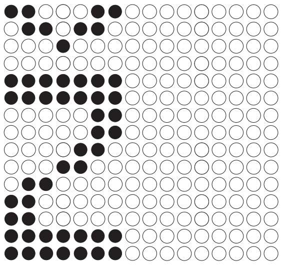

CHR177

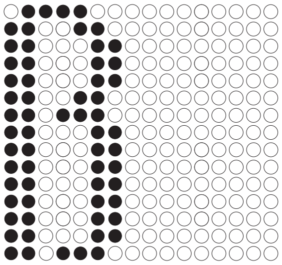

CHR178

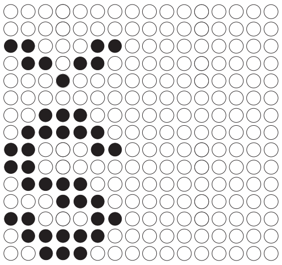

CHR179

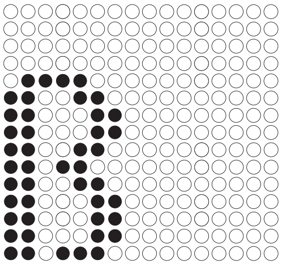

CHR180

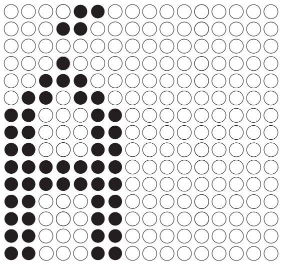

CHR181

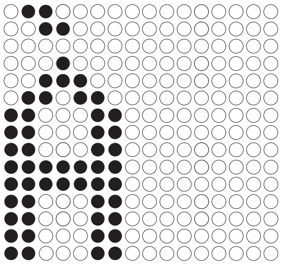

CHR182

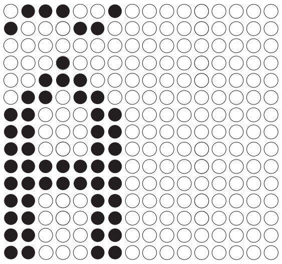

CHR183


CHR184

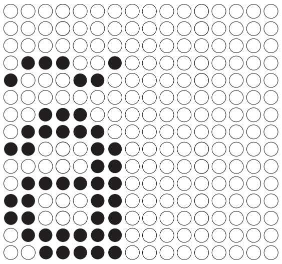

CHR185


CHR186

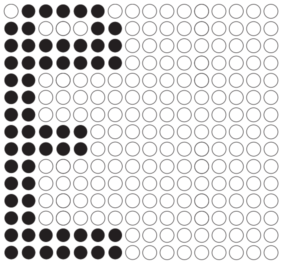

CHR187



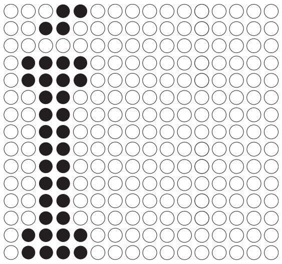
CHR188



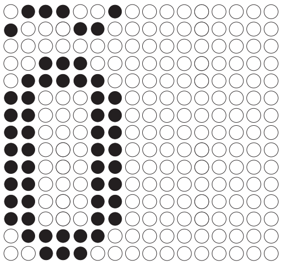
CHR189



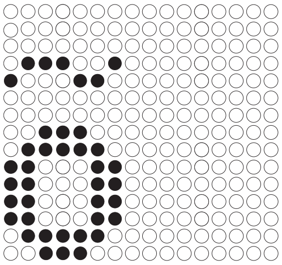
CHR190



CHR191

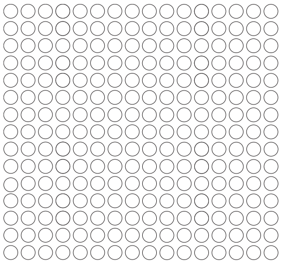


CHR192

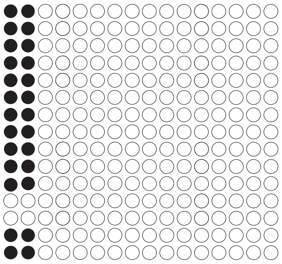


CHR193

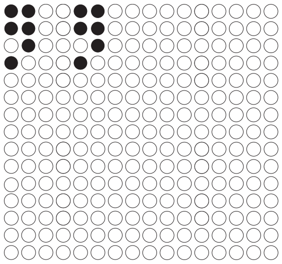
7.13.12 15-High Fancy (SF15)



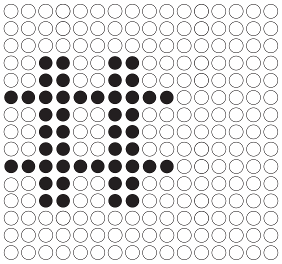
CHR032



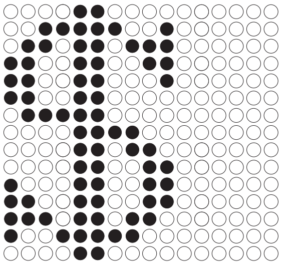
CHR033



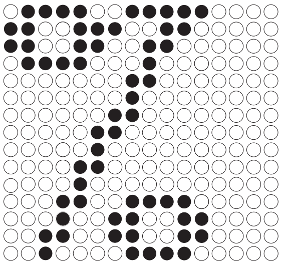
CHR034



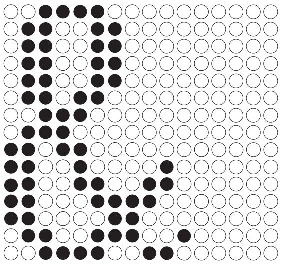
CHR035



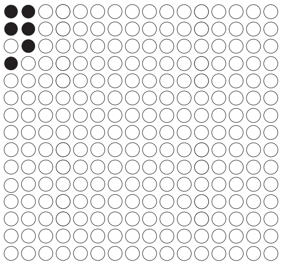
CHR036



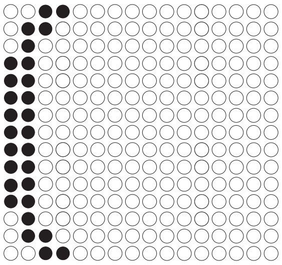
CHR037



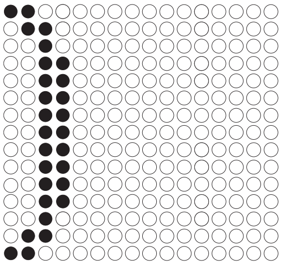
CHR038



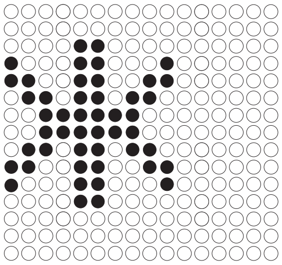
CHR039



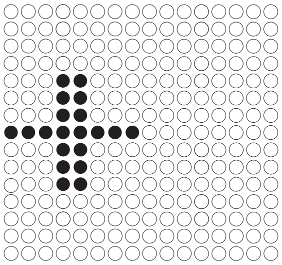
CHR040



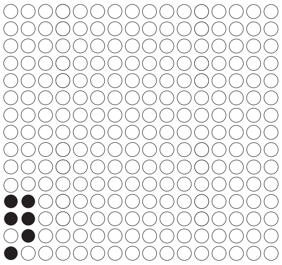
CHR041



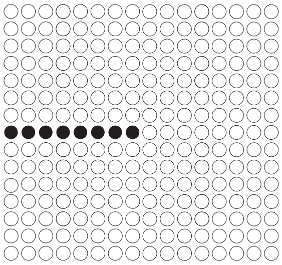
CHR042



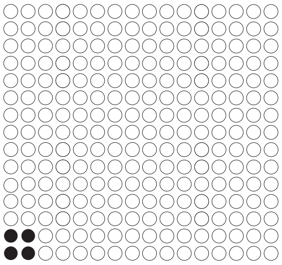
CHR043



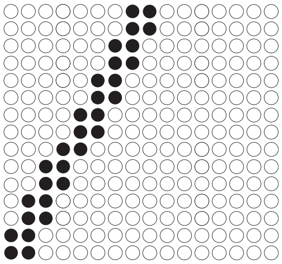
CHR044



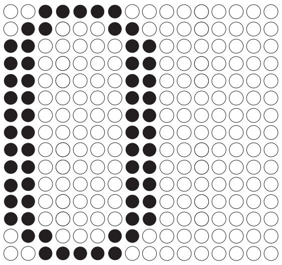
CHR045



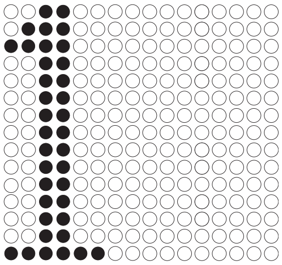
CHR046



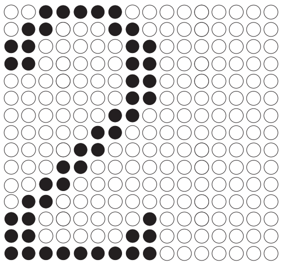
CHR047



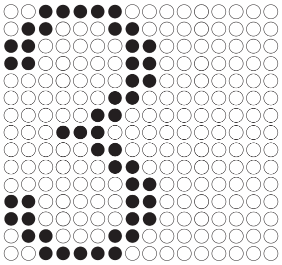
CHR048



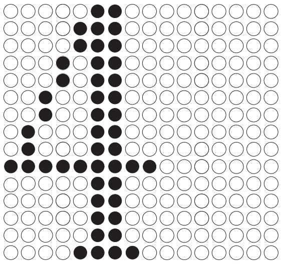
CHR049



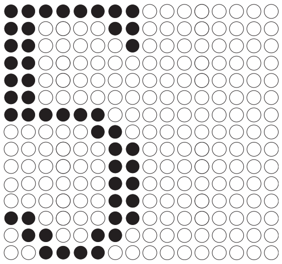
CHR050



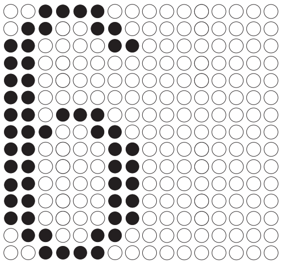
CHR051



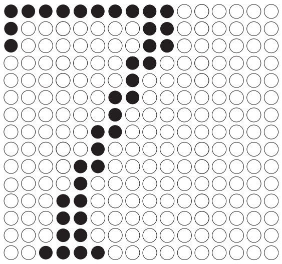
CHR052



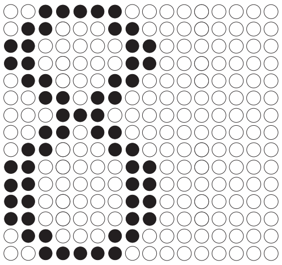
CHR053



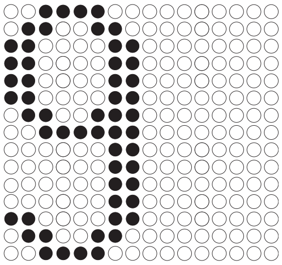
CHR054



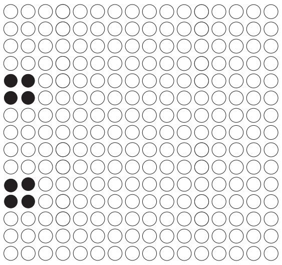
CHR055



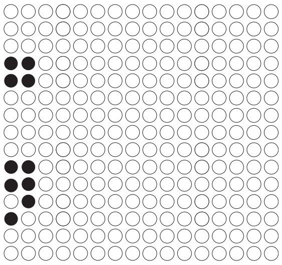
CHR056



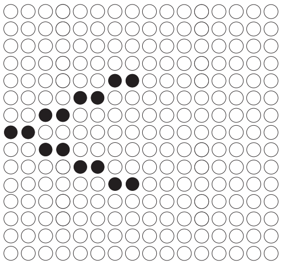
CHR057



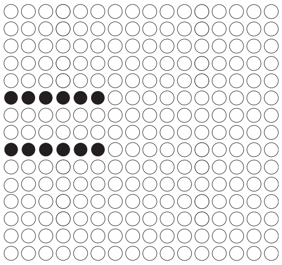
CHR058



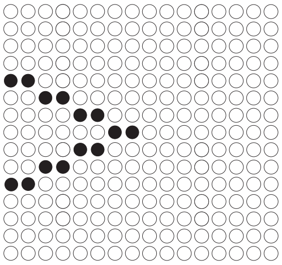
CHR059



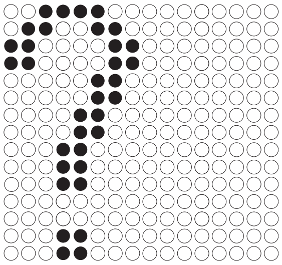
CHR060



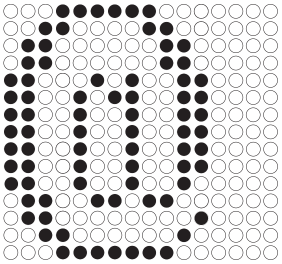
CHR061

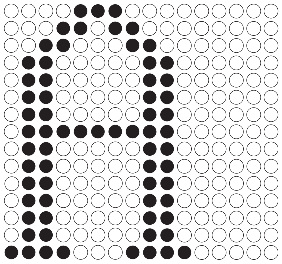


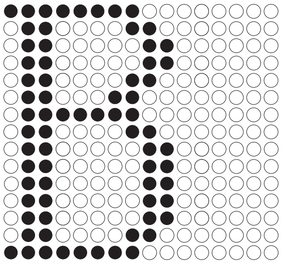
CHR062

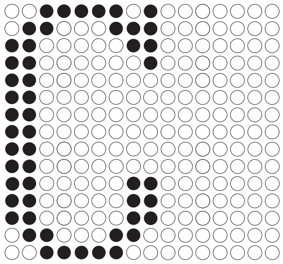


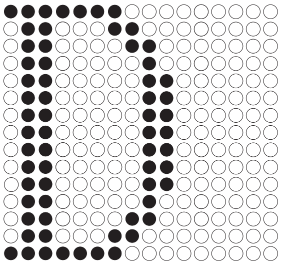
CHR063

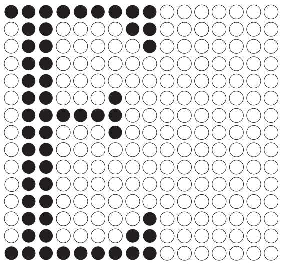

CHR064

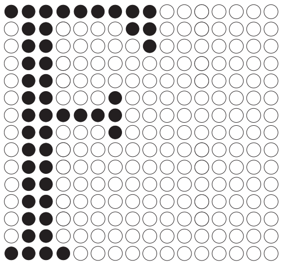

CHR065

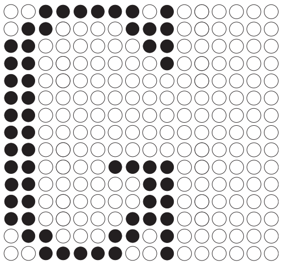

CHR066

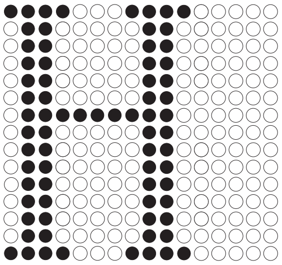

CHR067

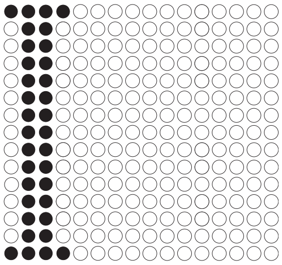

CHR068

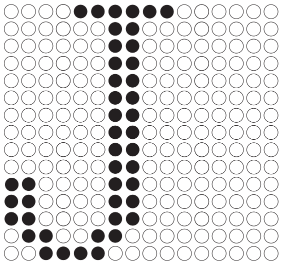

CHR069

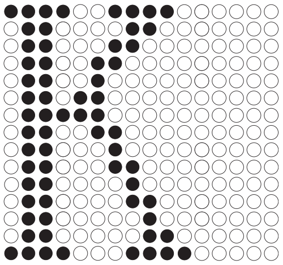

CHR070

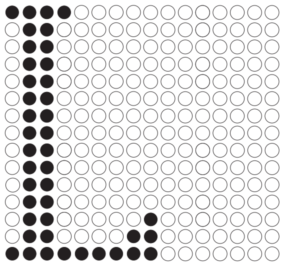

CHR071

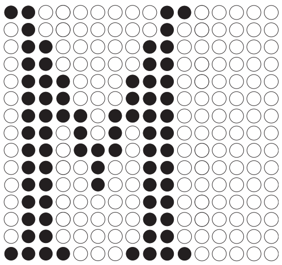

CHR072

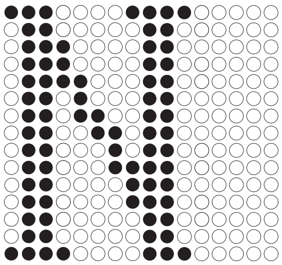

CHR073

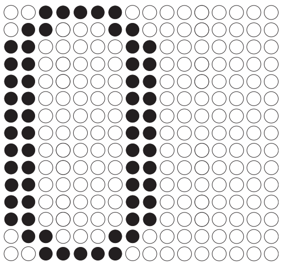

CHR074

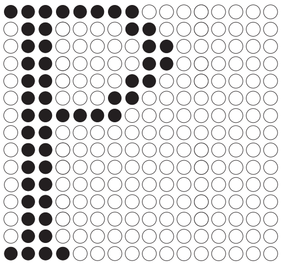

CHR075

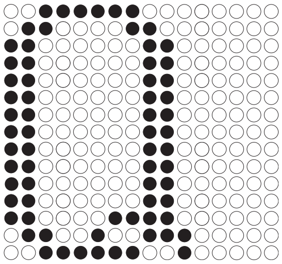

CHR076

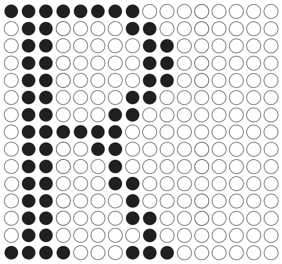

CHR077

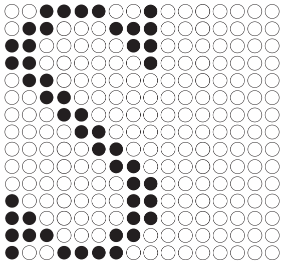

CHR078

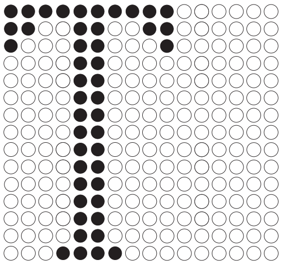

CHR079

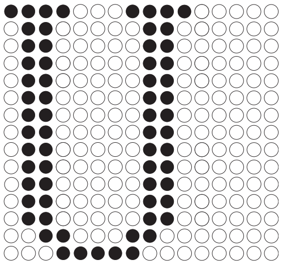

CHR080

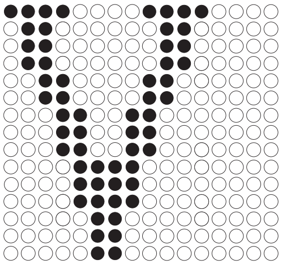

CHR081

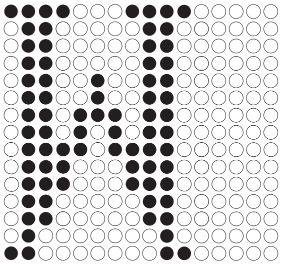

CHR082

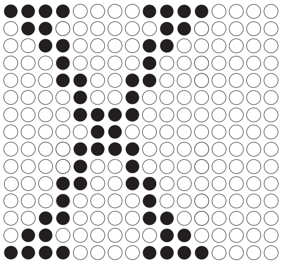

CHR083

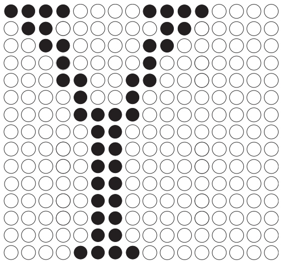

CHR084

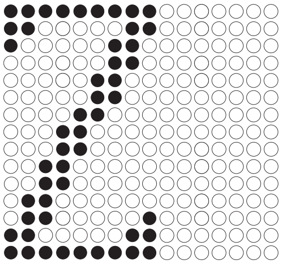

CHR085

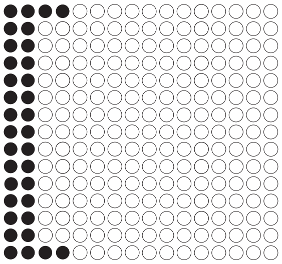

CHR086

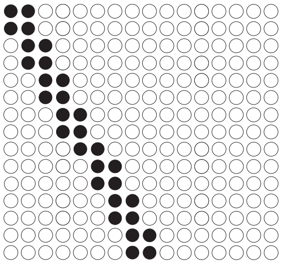

CHR087

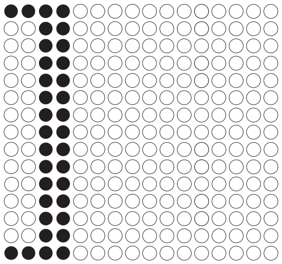

CHR088

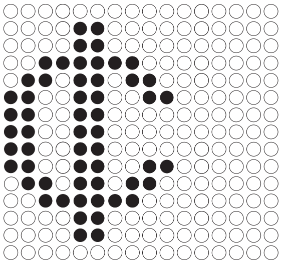

CHR089

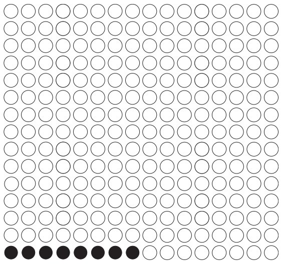

CHR090

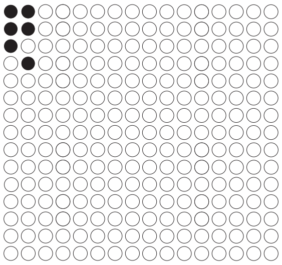

CHR091

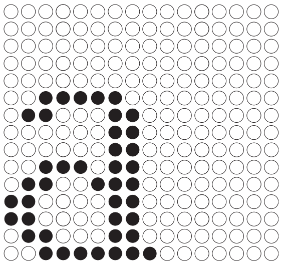

CHR092

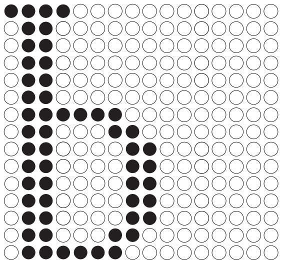

CHR093

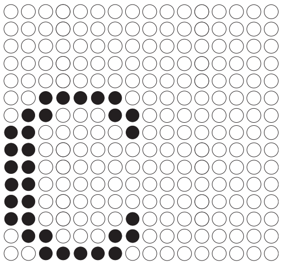

CHR094

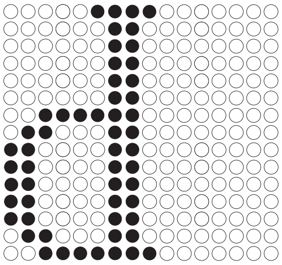

CHR095

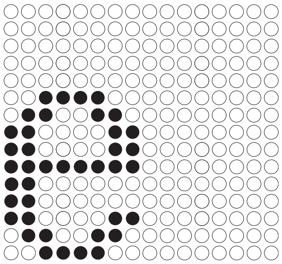

CHR096

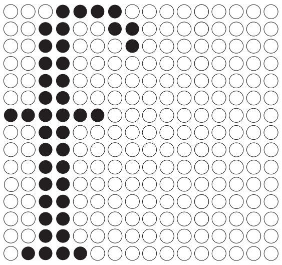

CHR097

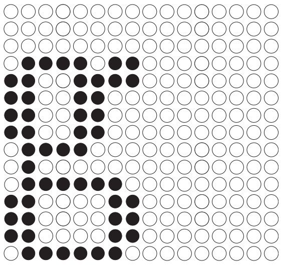

CHR098

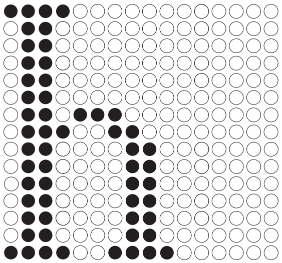

CHR099


CHR100

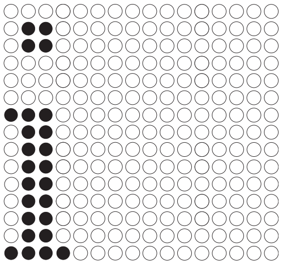

CHR101


CHR102

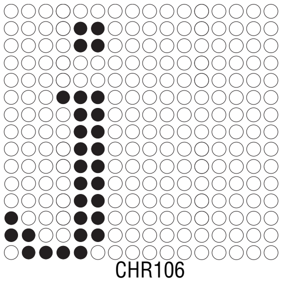

CHR103



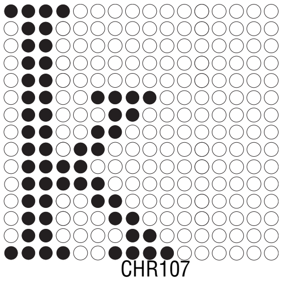
CHR104



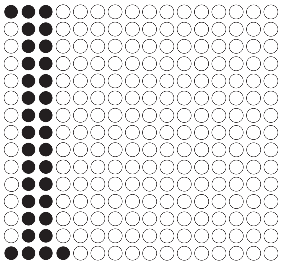
CHR105



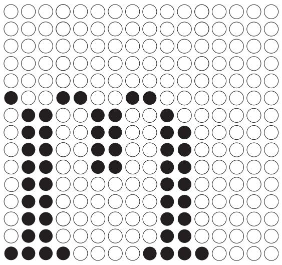
CHR106



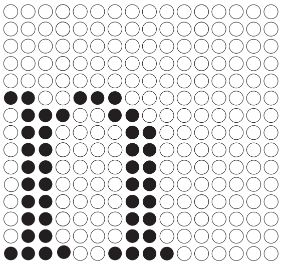
CHR107



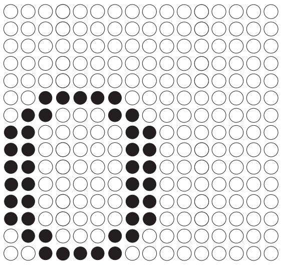
CHR108



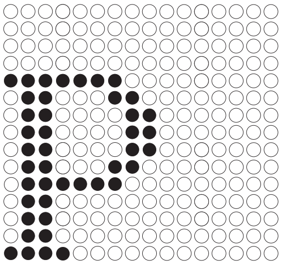
CHR109



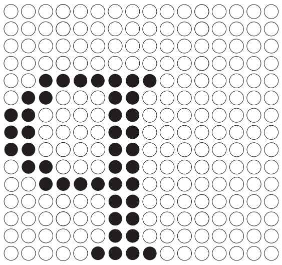
CHR110



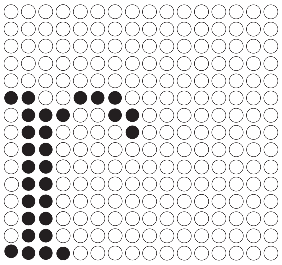
CHR111



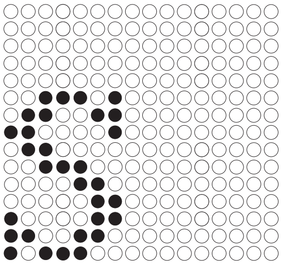
CHR112



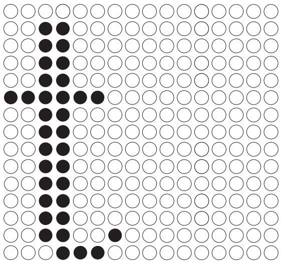
CHR113



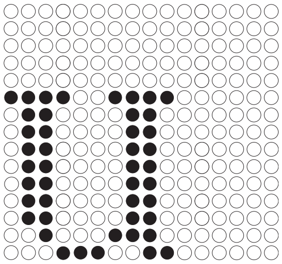
CHR114



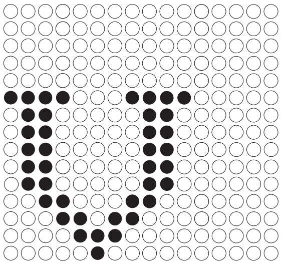
CHR115



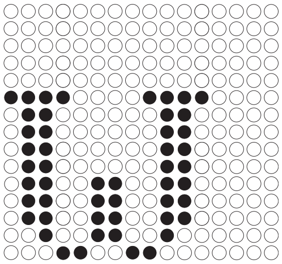
CHR116



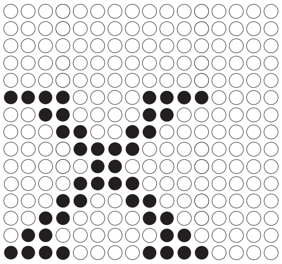
CHR117



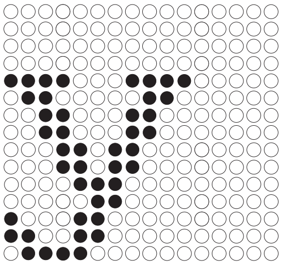
CHR118



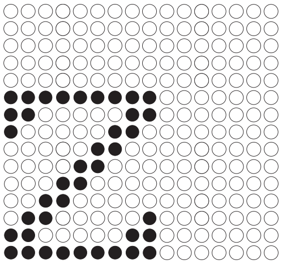
CHR119



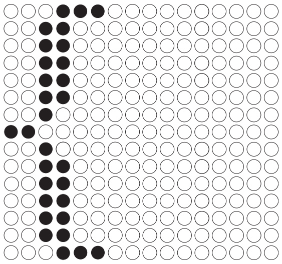
CHR120



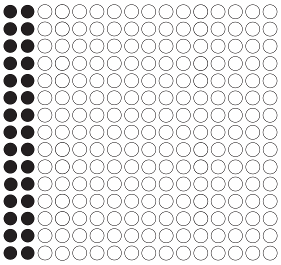
CHR121

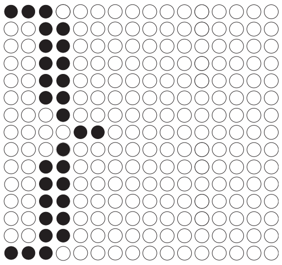


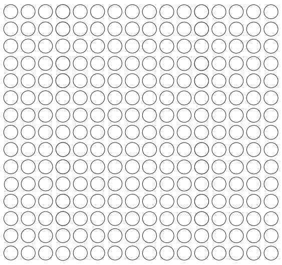
CHR122

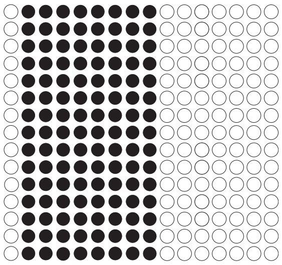


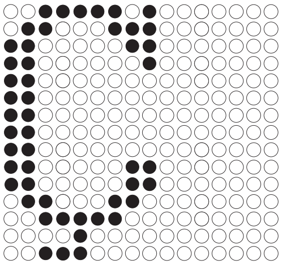
CHR123

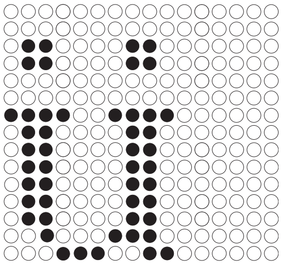

CHR124

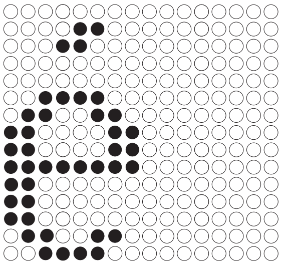

CHR125

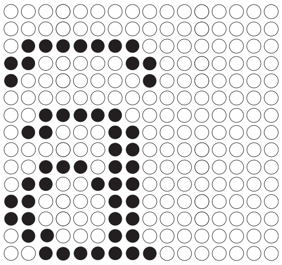

CHR126

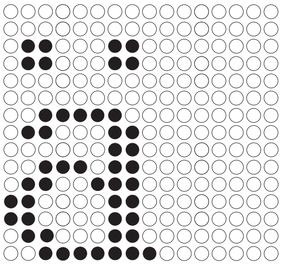

CHR127

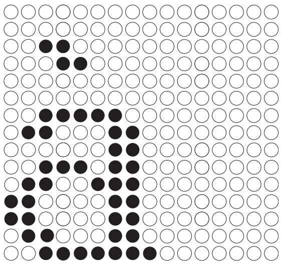

CHR128

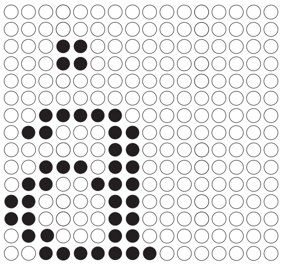

CHR129

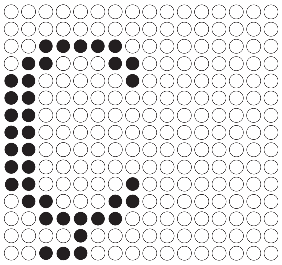

CHR130

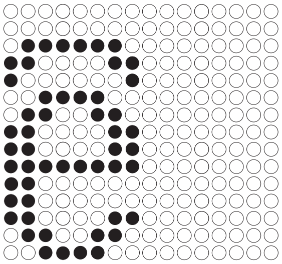

CHR131

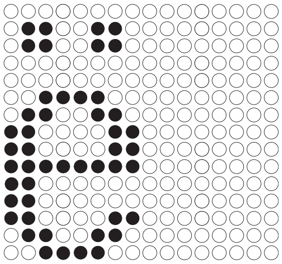

CHR132

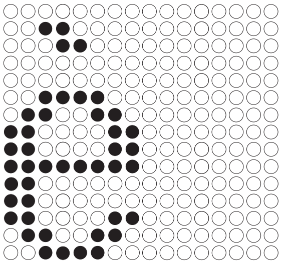

CHR133

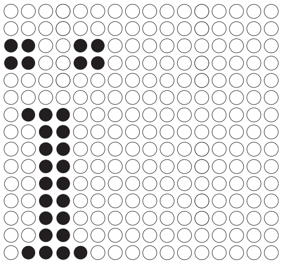

CHR134

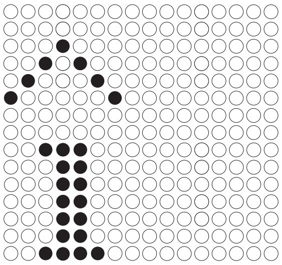

CHR135

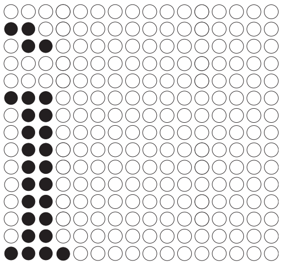

CHR136

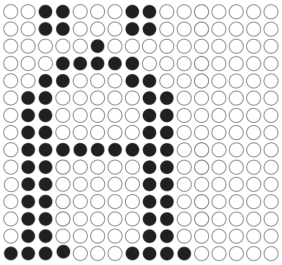

CHR137

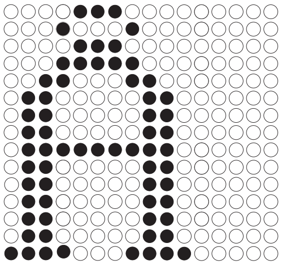

CHR138

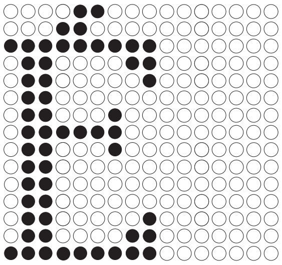

CHR139


CHR140

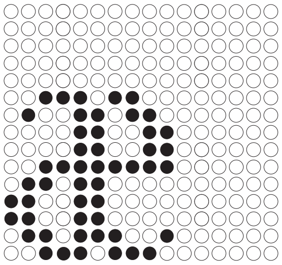

CHR141


CHR142

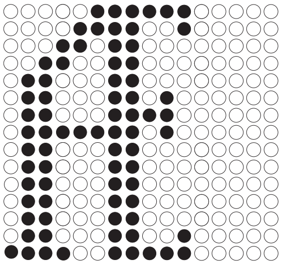

CHR143



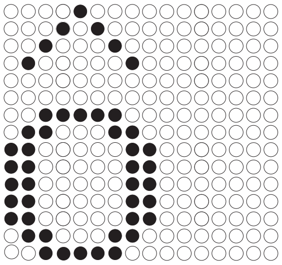
CHR144



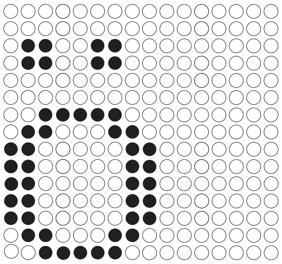
CHR145



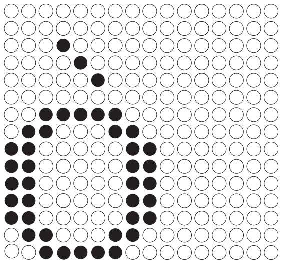
CHR146



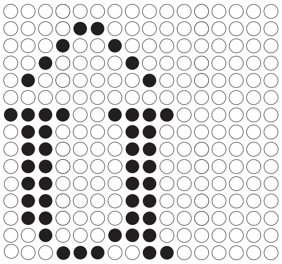
CHR147



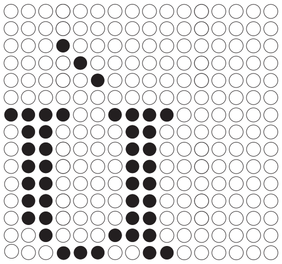
CHR148



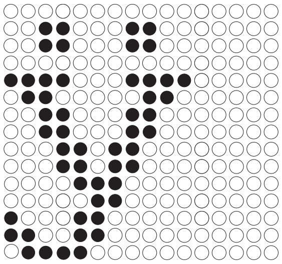
CHR149



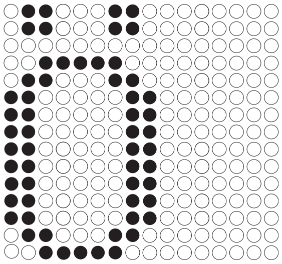
CHR150



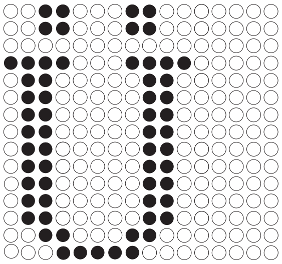
CHR151



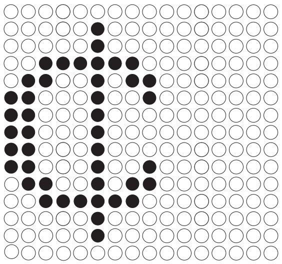
CHR152



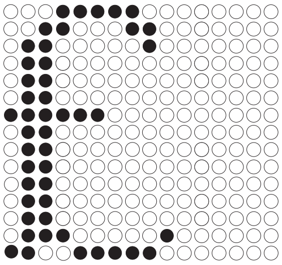
CHR153



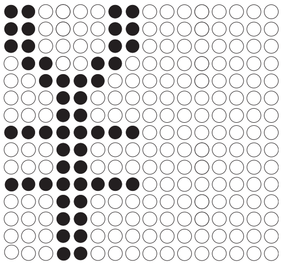
CHR154



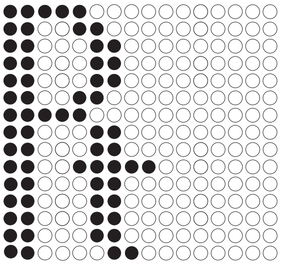
CHR155



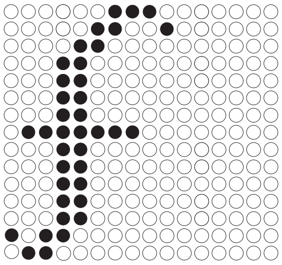
CHR156



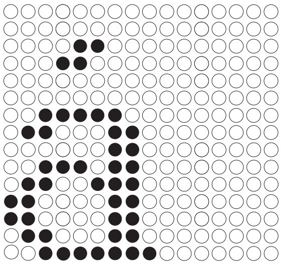
CHR157



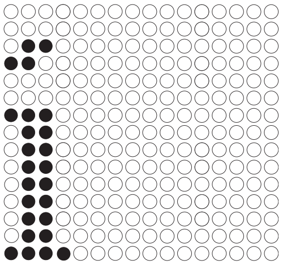
CHR158



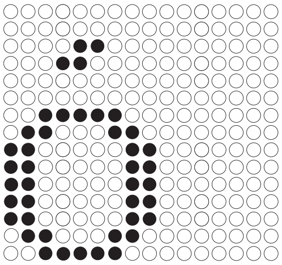
CHR159



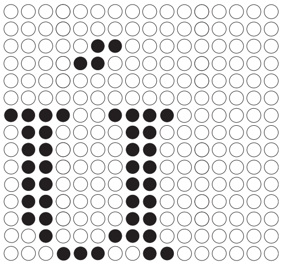
CHR160



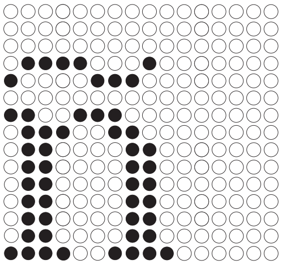
CHR161

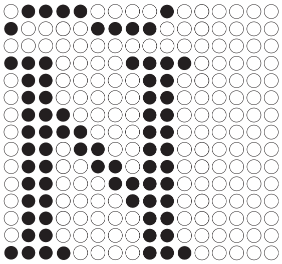


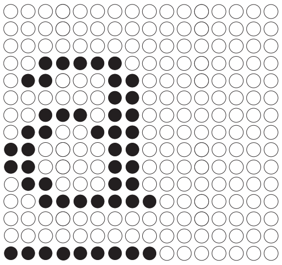
CHR162

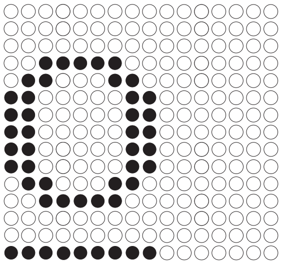


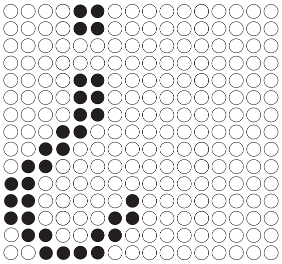
CHR163

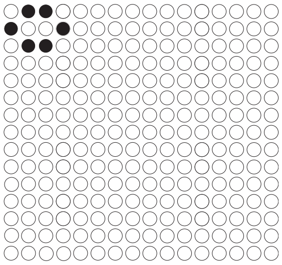

CHR164

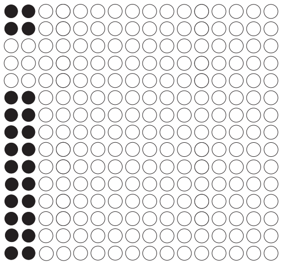

CHR165

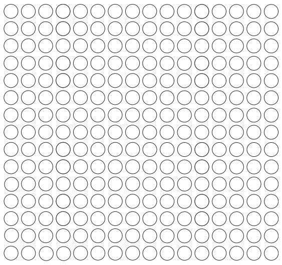

CHR166

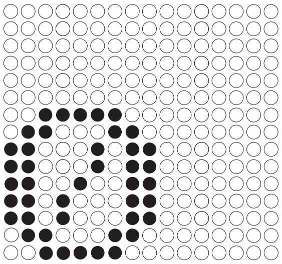

CHR167

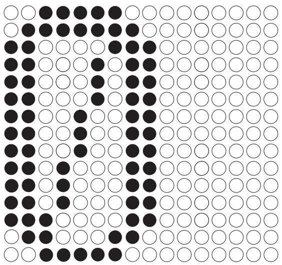

CHR168

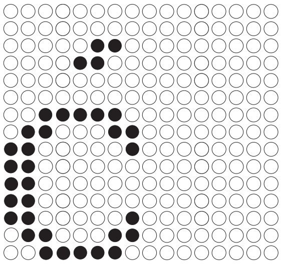

CHR169

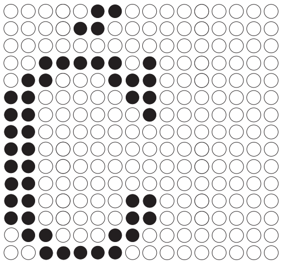

CHR170

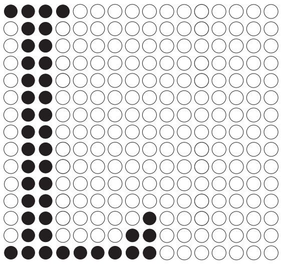

CHR171

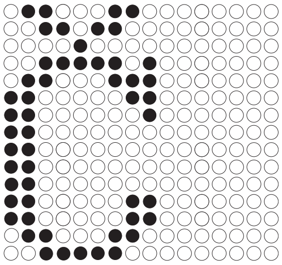

CHR172

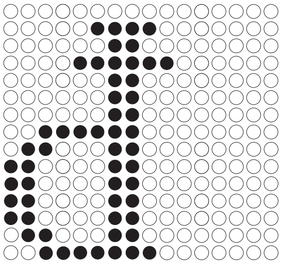

CHR173

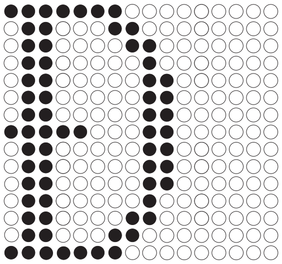

CHR174

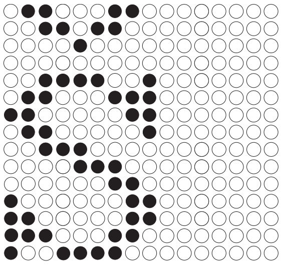

CHR175

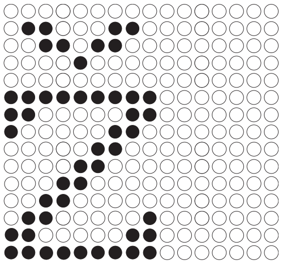

CHR176

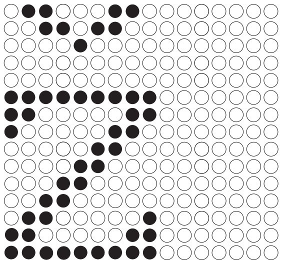

CHR177

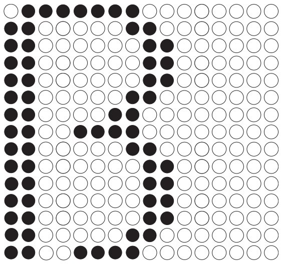

CHR178

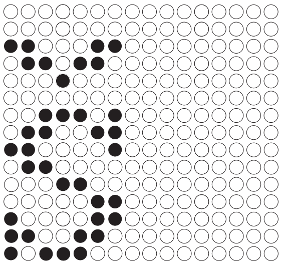

CHR179

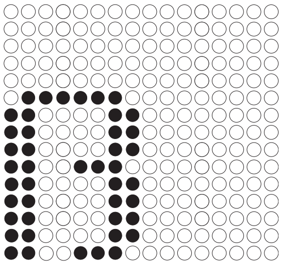

CHR180

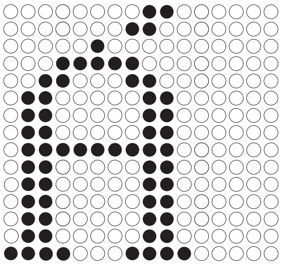

CHR181

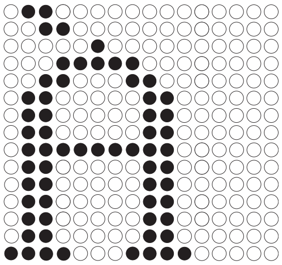

CHR182

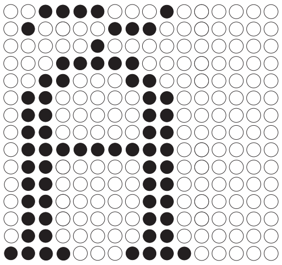

CHR183

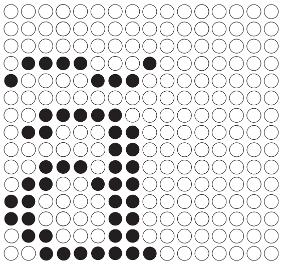

CHR184

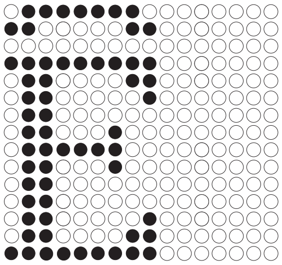

CHR185

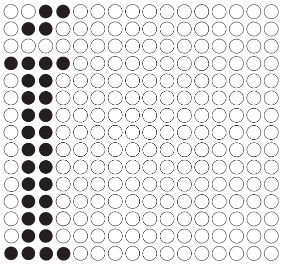

CHR186

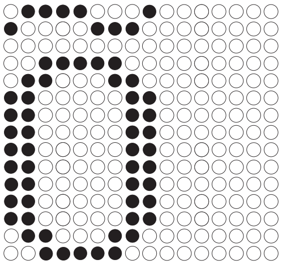

CHR187

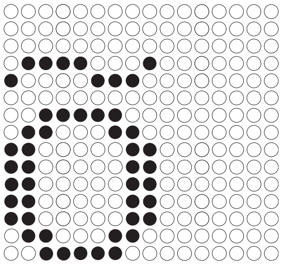

CHR188


CHR189

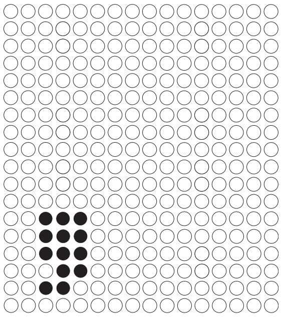

CHR190

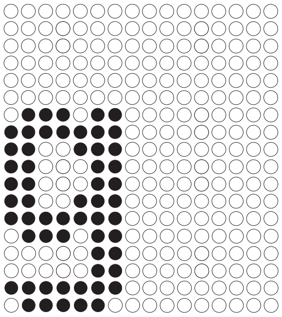

CHR191

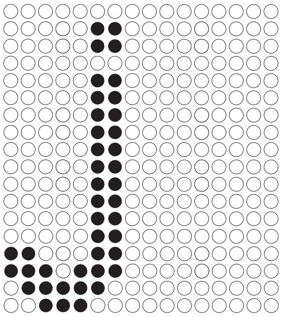

CHR192

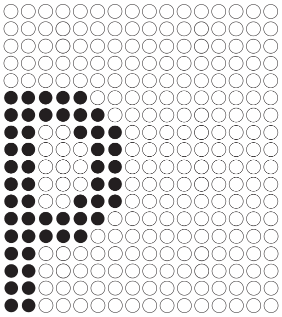

CHR193

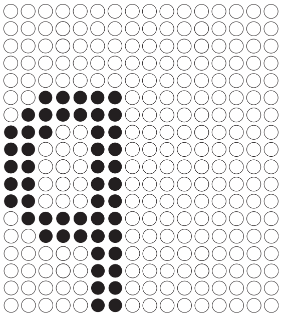
7.13.13 15-High True Descender Regular

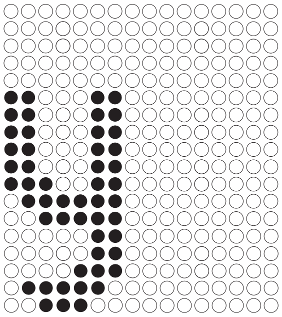

CHR15C


CHR15G

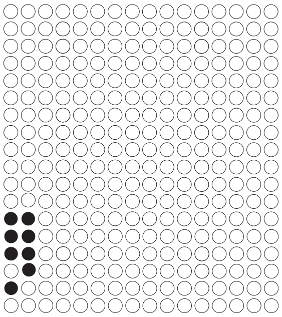

CHR15J


CHR15P

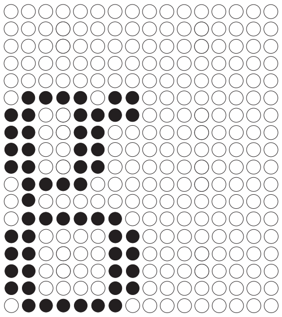

CHR15Q


CHR15Y

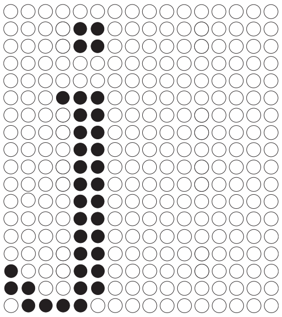
7.13.14 15-High True Descender Fancy



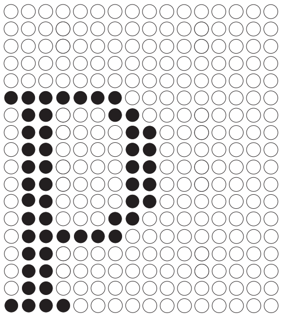
CR15FC



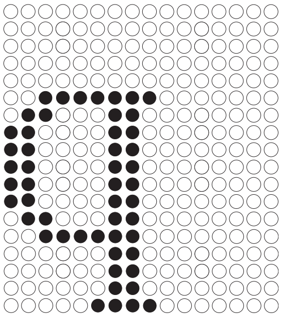
CR15FG



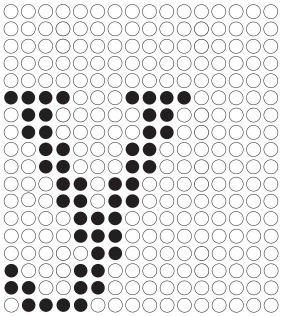
CR15FJ



CR15FP

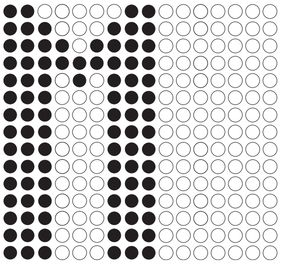


CR15FQ

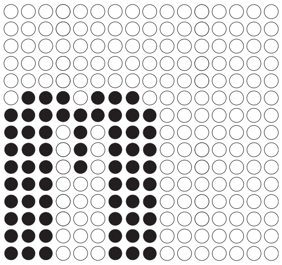


CR15FY

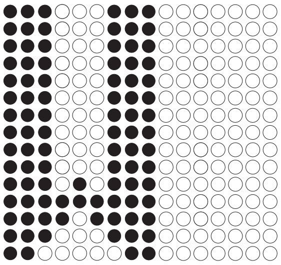
7.13.15 15-High Fat Character



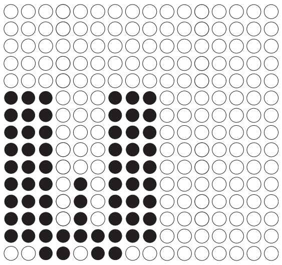
FATM



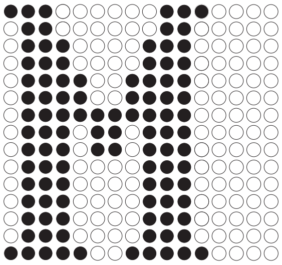
FATM1



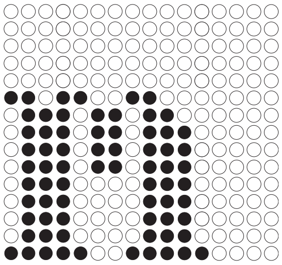
FATW



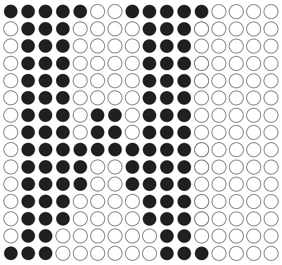
FATW1



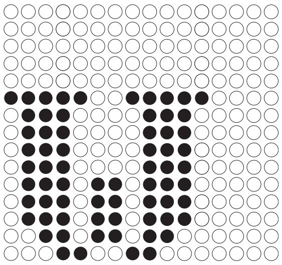
FFATM



FFATM1

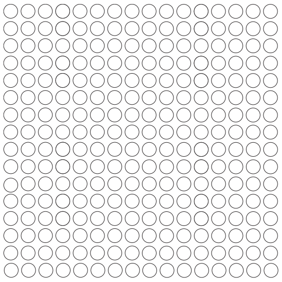


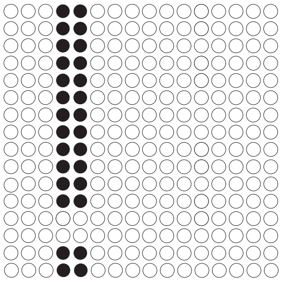
FFATW

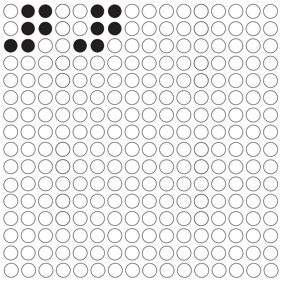


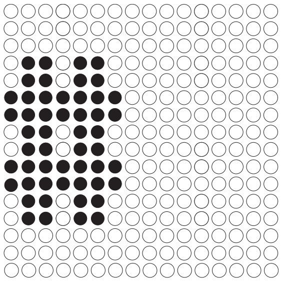
FFATW!

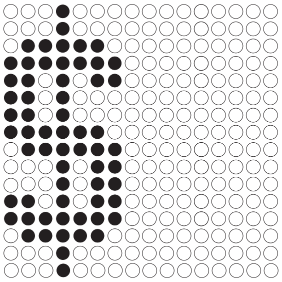
7.13.16 16-High Regular (SS16)

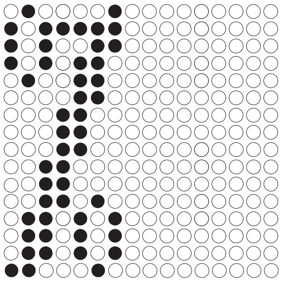

CHR032

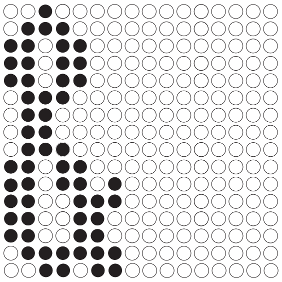

CHR033

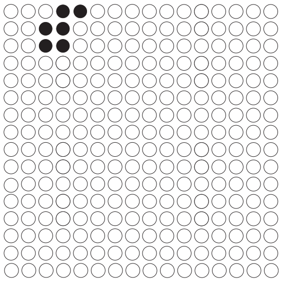

CHR034

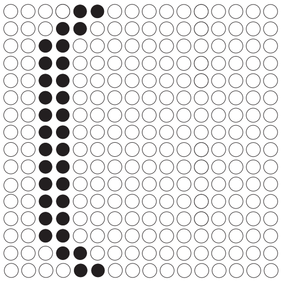

CHR035

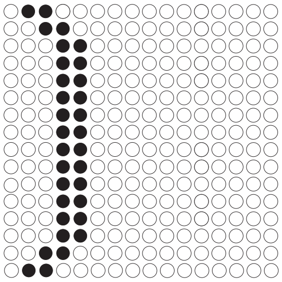

CHR036

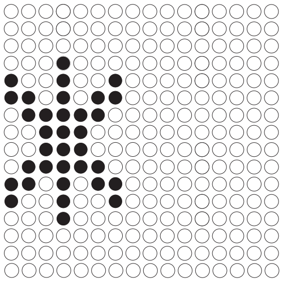

CHR037

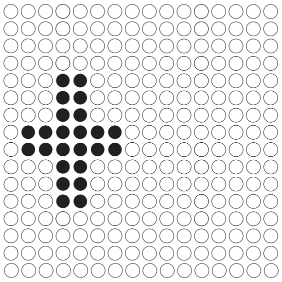

CHR038

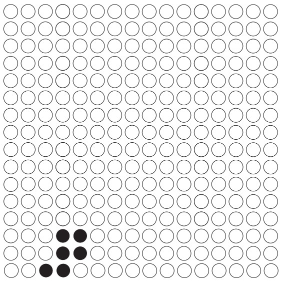

CHR039

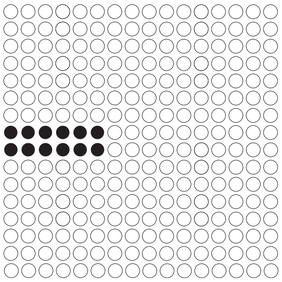

CHR040

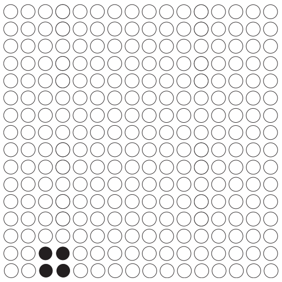

CHR041

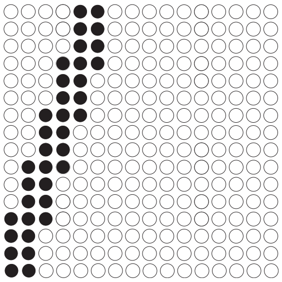

CHR042

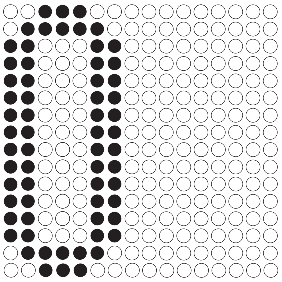

CHR043

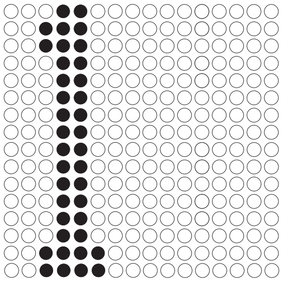

CHR044

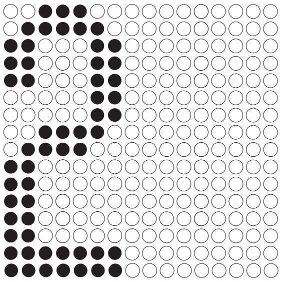

CHR045

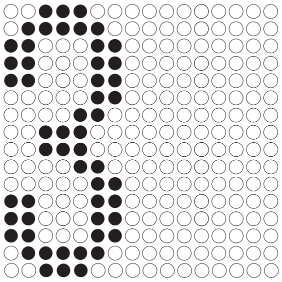

CHR046

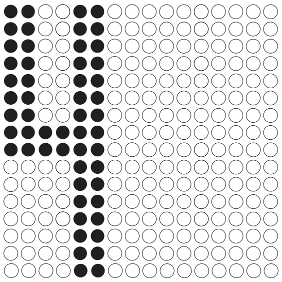

CHR047

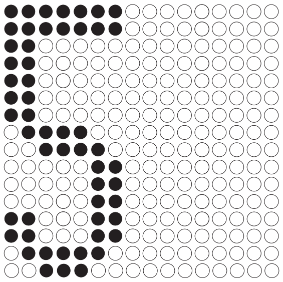

CHR048

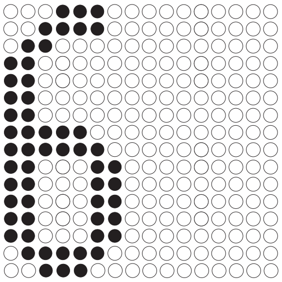

CHR049

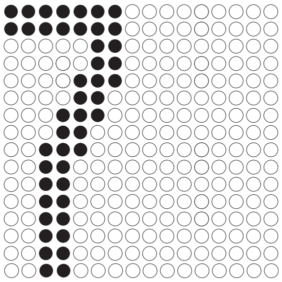

CHR050

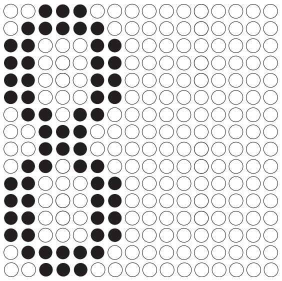

CHR051

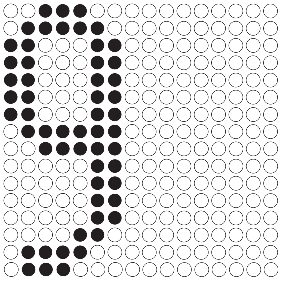

CHR052

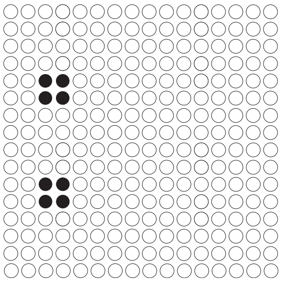

CHR053

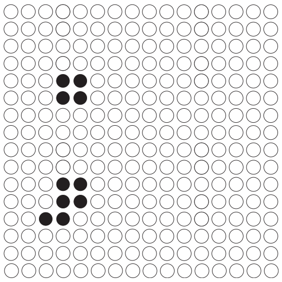

CHR054

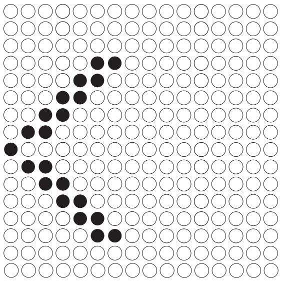

CHR055

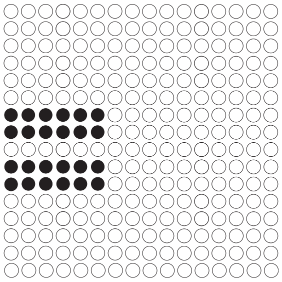

CHR056

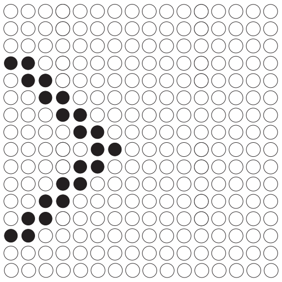

CHR057

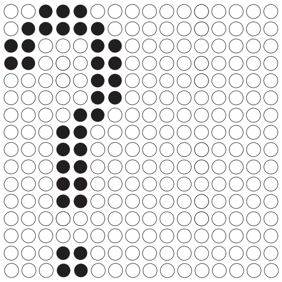

CHR058

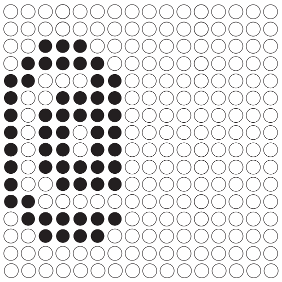

CHR059

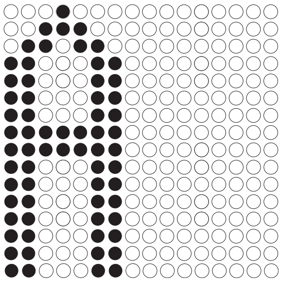

CHR060

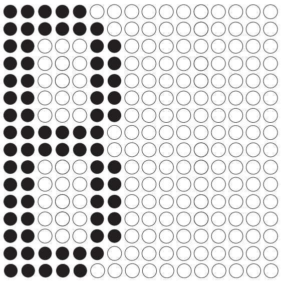

CHR061

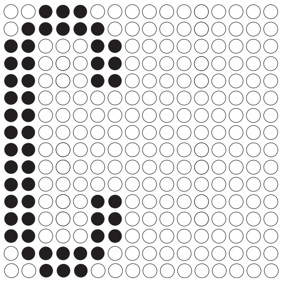

CHR062

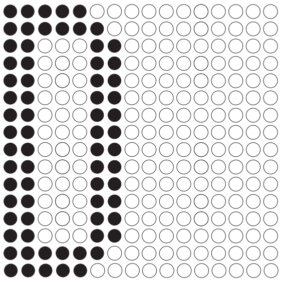

CHR063

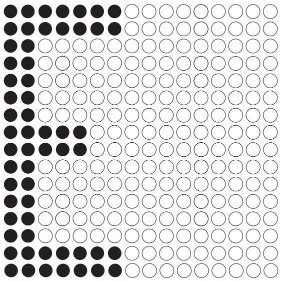

CHR064

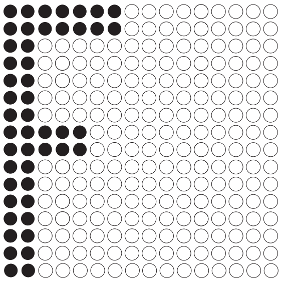

CHR065

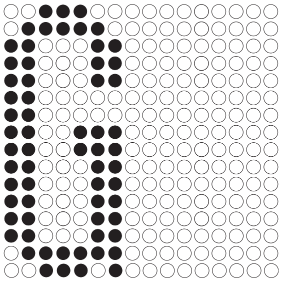

CHR066

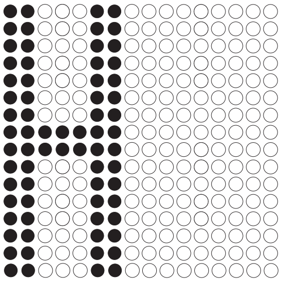

CHR067

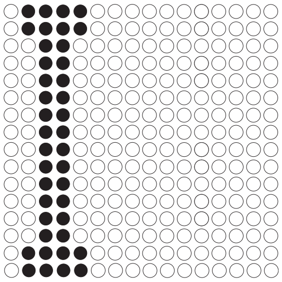

CHR068

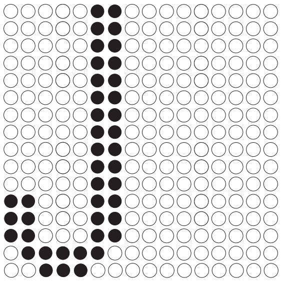

CHR069

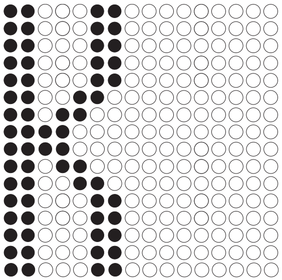

CHR070

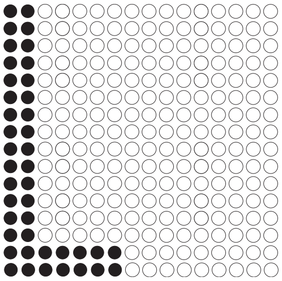

CHR071

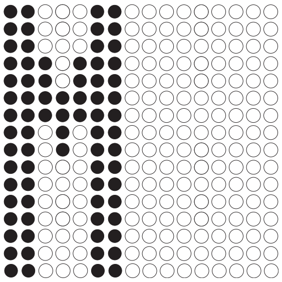

CHR072

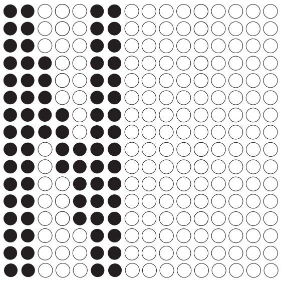

CHR073

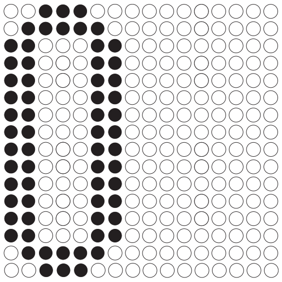

CHR074

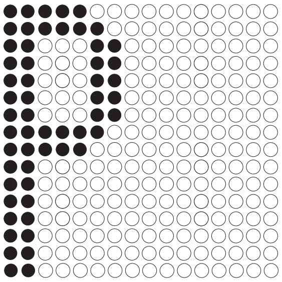

CHR075

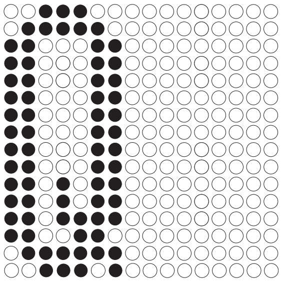

CHR076

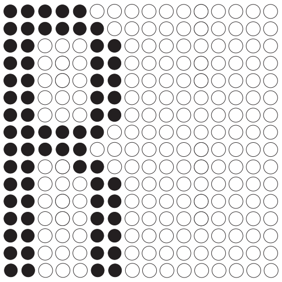

CHR077

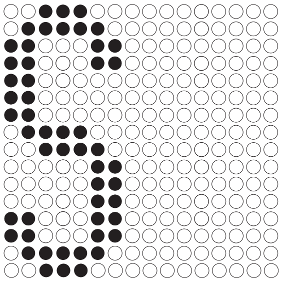

CHR078

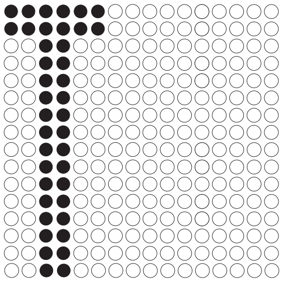

CHR079

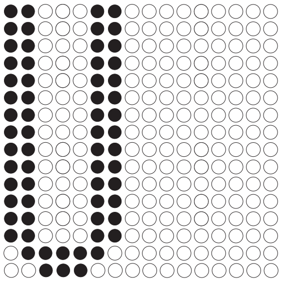

CHR080

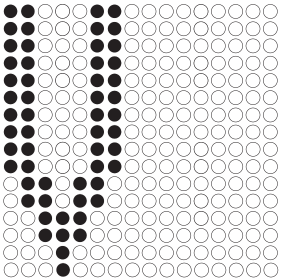

CHR081

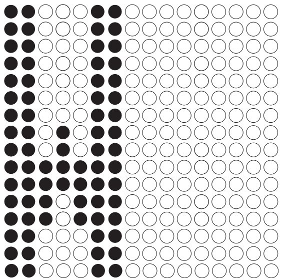

CHR082

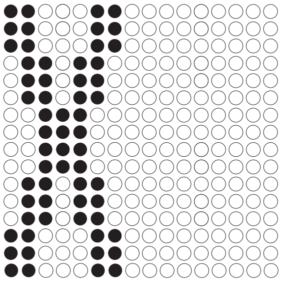

CHR083

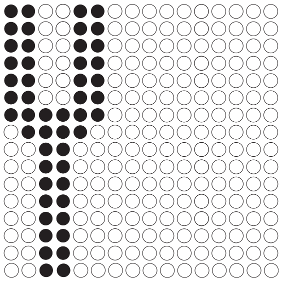

CHR084

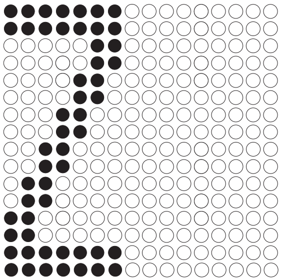

CHR085

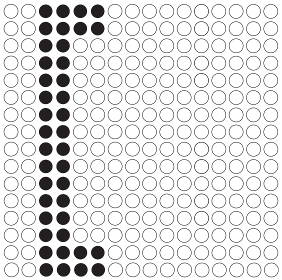

CHR086

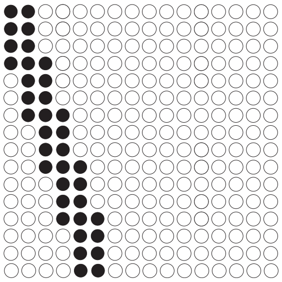

CHR087


CHR088

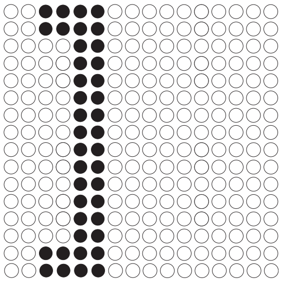

CHR089


CHR090

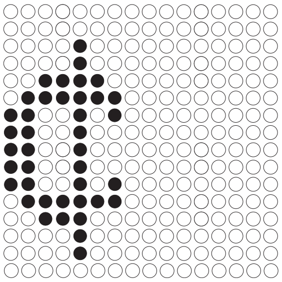

CHR091



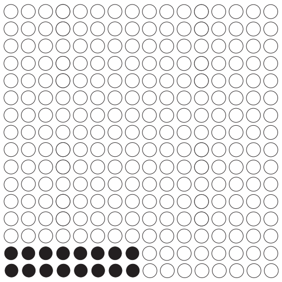
CHR092



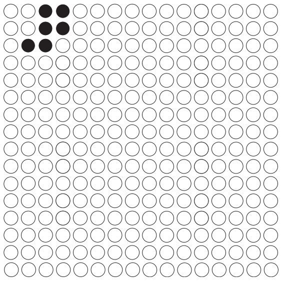
CHR093



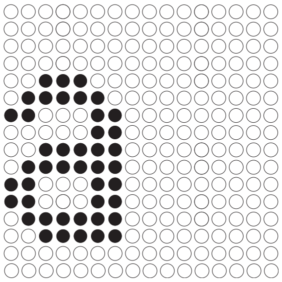
CHR094



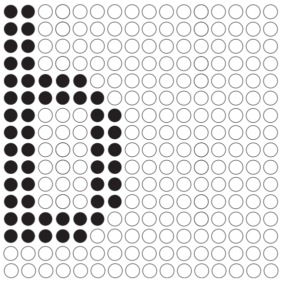
CHR095



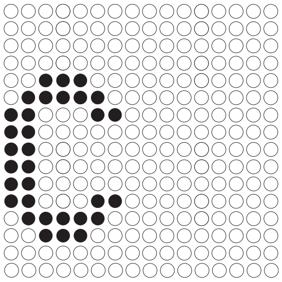
CHR096



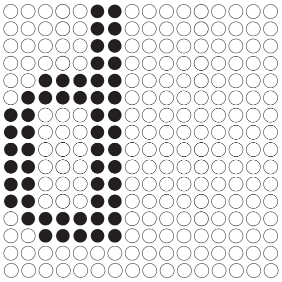
CHR097



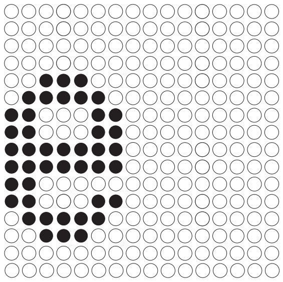
CHR098



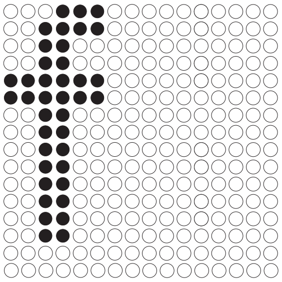
CHR099



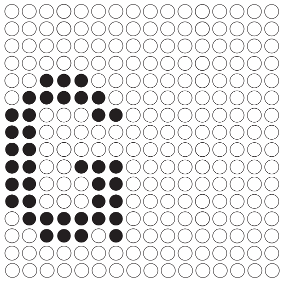
CHR100



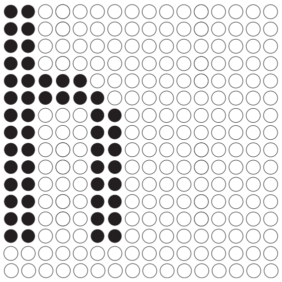
CHR101



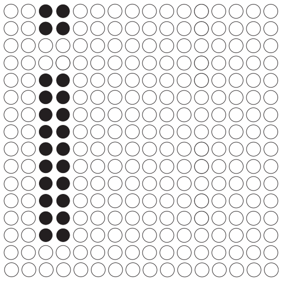
CHR102



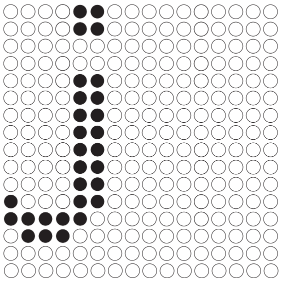
CHR103



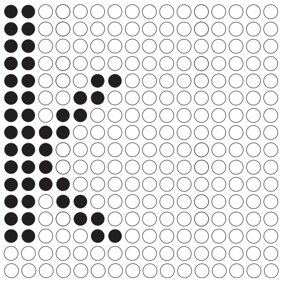
CHR104



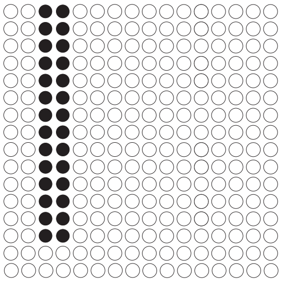
CHR105



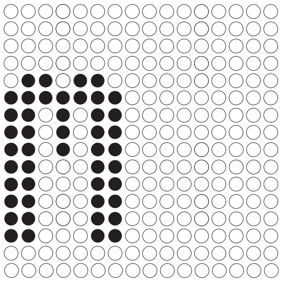
CHR106



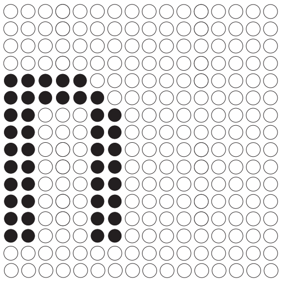
CHR107



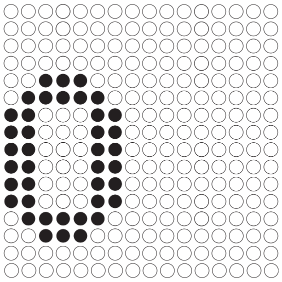
CHR108



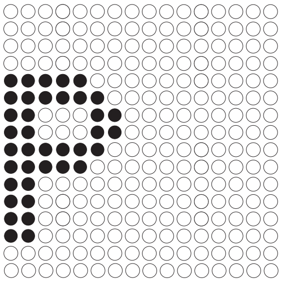
CHR109

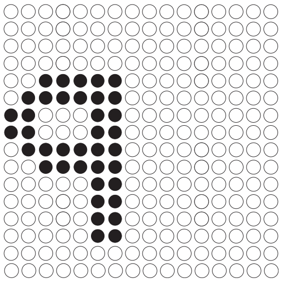


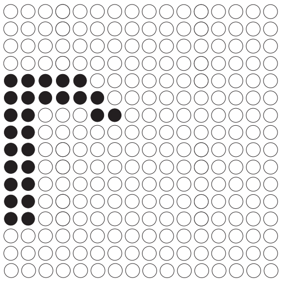
CHR110

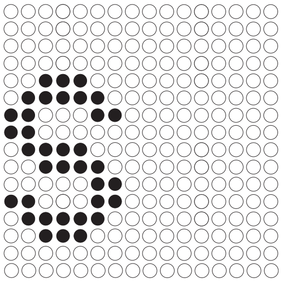


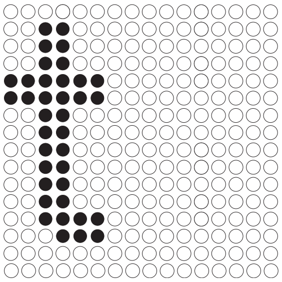
CHR111

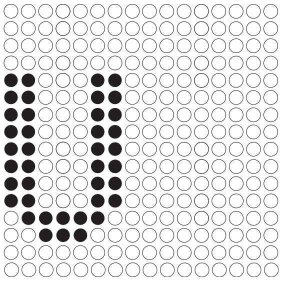

CHR112

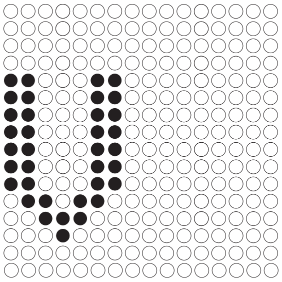

CHR113

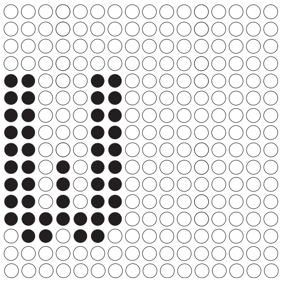

CHR114

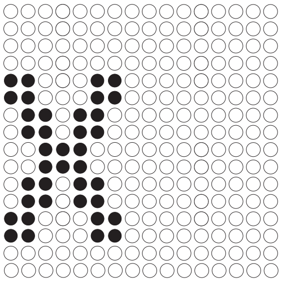

CHR115

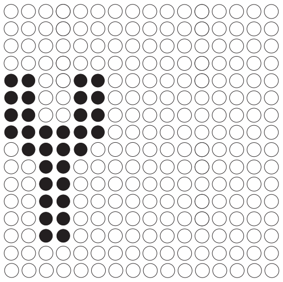

CHR116

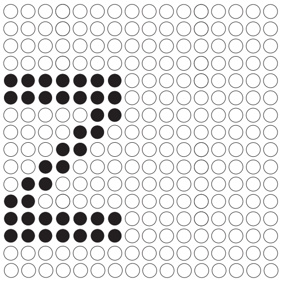

CHR117

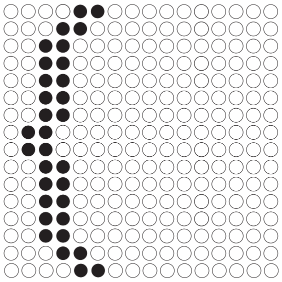

CHR118

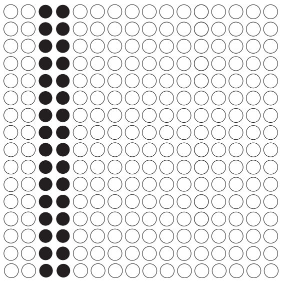

CHR119

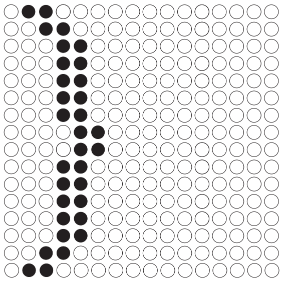

CHR120

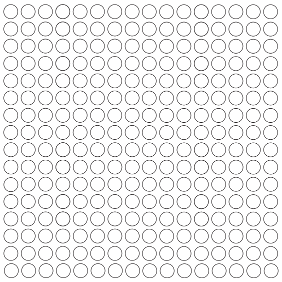

CHR121

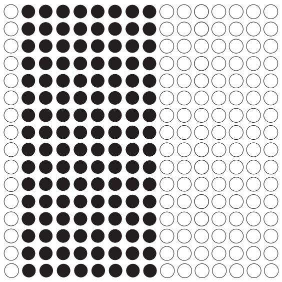

CHR122

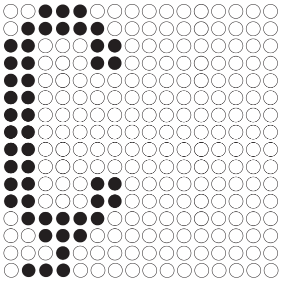

CHR123

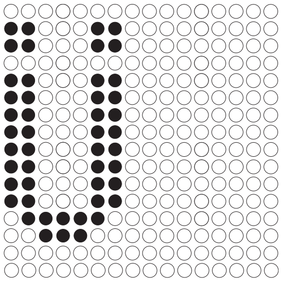

CHR124

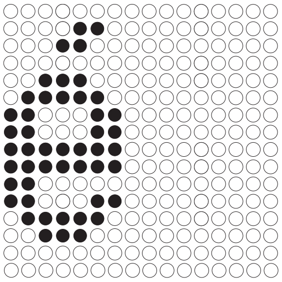

CHR125

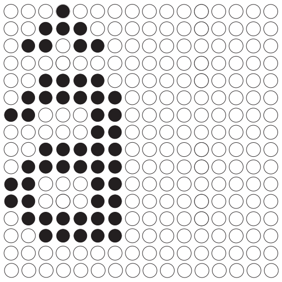

CHR126

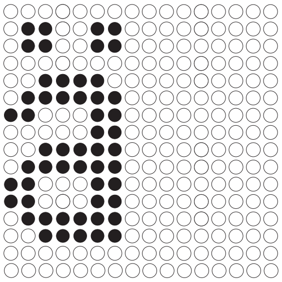

CHR127

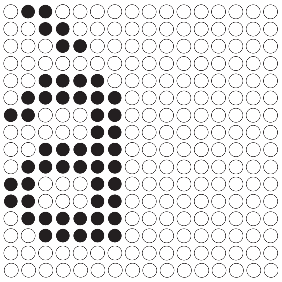

CHR128

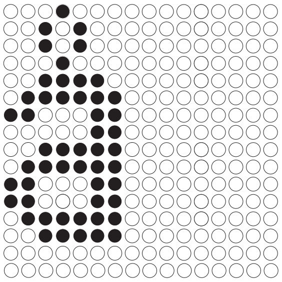

CHR129

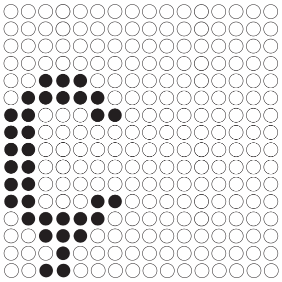

CHR130

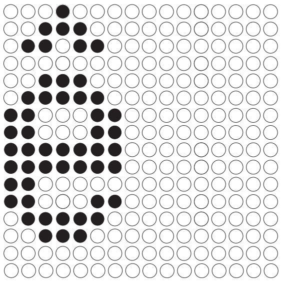

CHR131

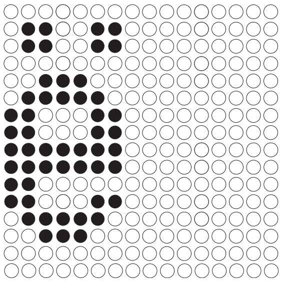

CHR132

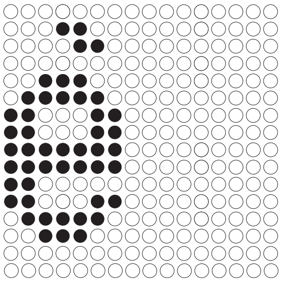

CHR133

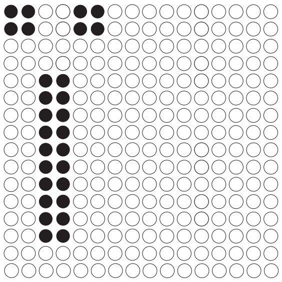

CHR134

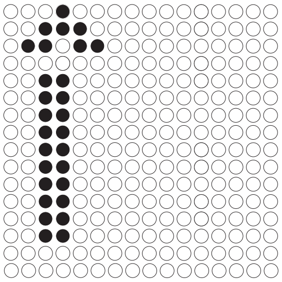

CHR135

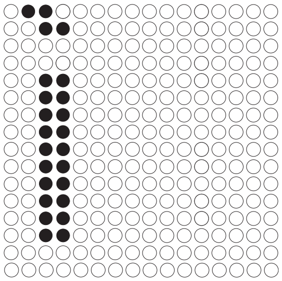

CHR136

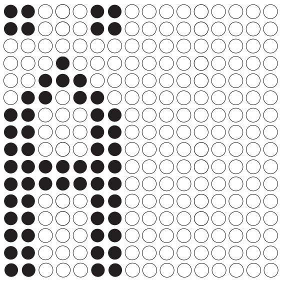

CHR137

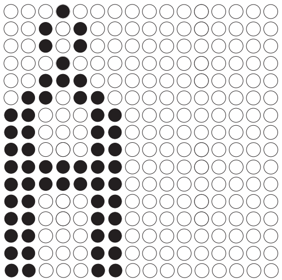

CHR138

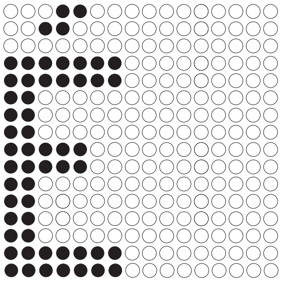

CHR139

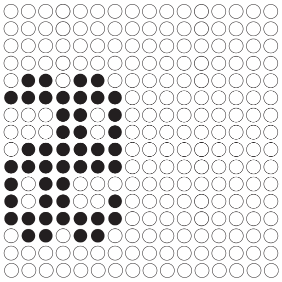

CHR140

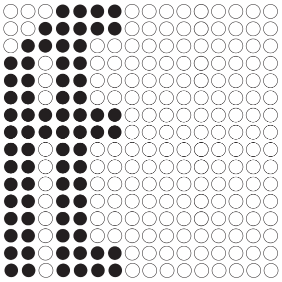

CHR141

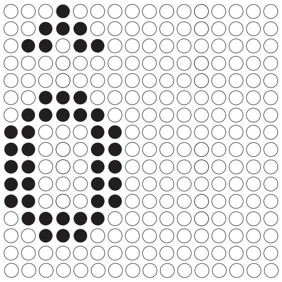

CHR142

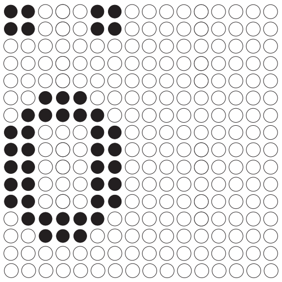

CHR143

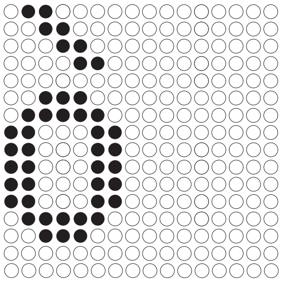

CHR144

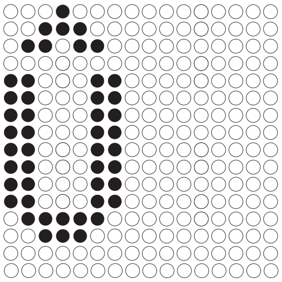

CHR145

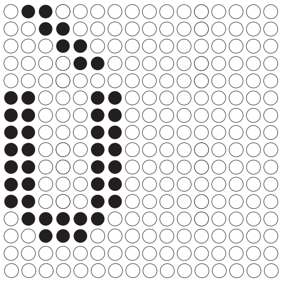

CHR146

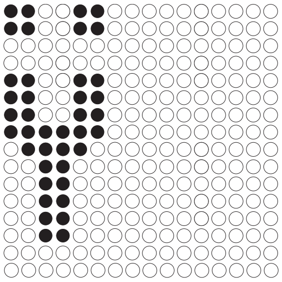

CHR147

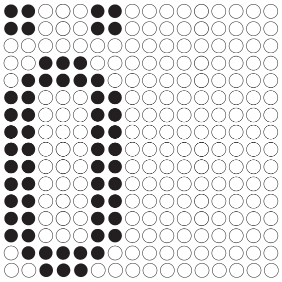

CHR148

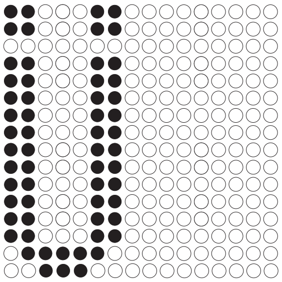

CHR149

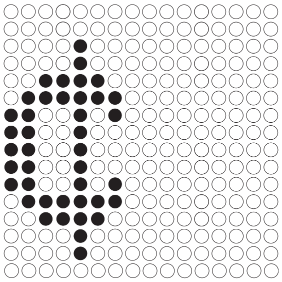

CHR150

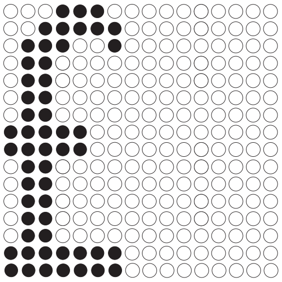

CHR151

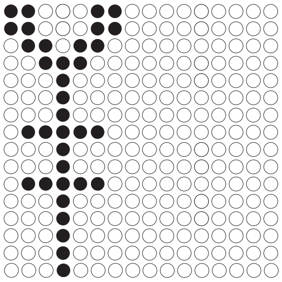

CHR152

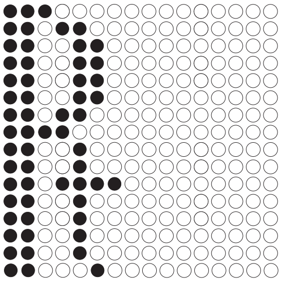

CHR153

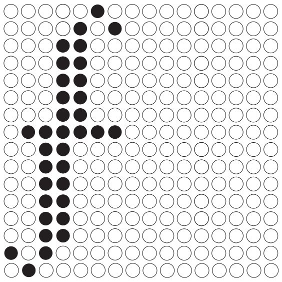

CHR154

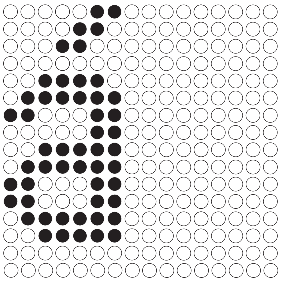

CHR155

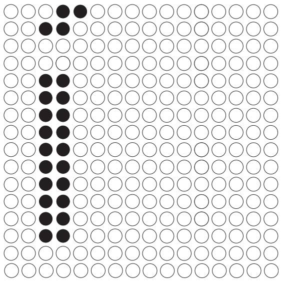

CHR156

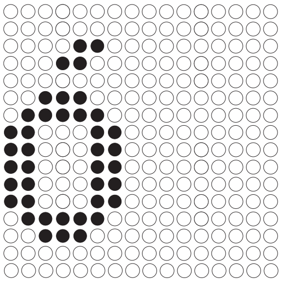

CHR157

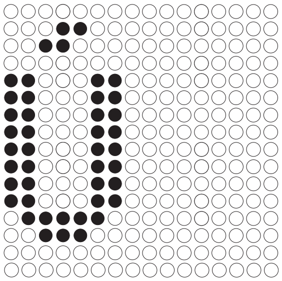

CHR158

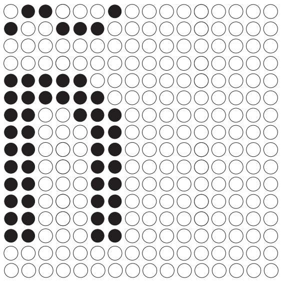

CHR159

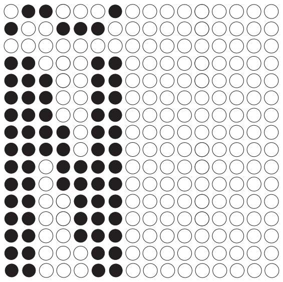

CHR160

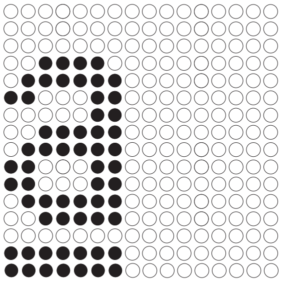

CHR161

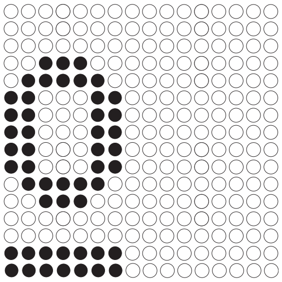

CHR162

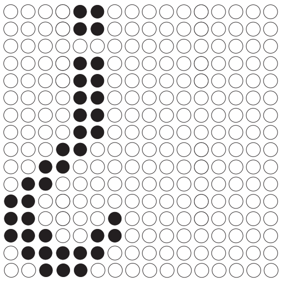

CHR163

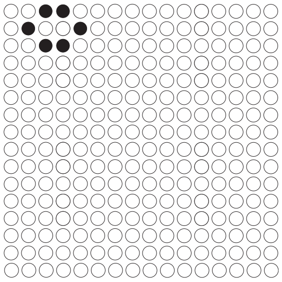

CHR164

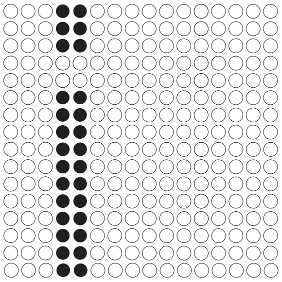

CHR165

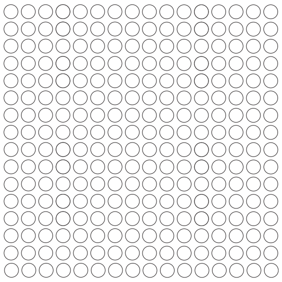

CHR166

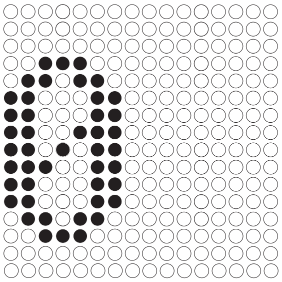

CHR167

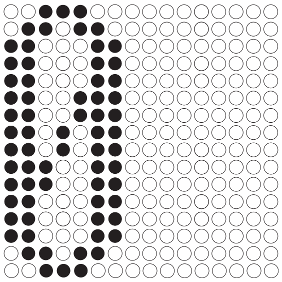

CHR168

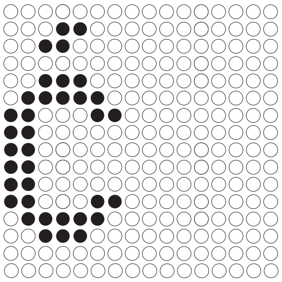

CHR169

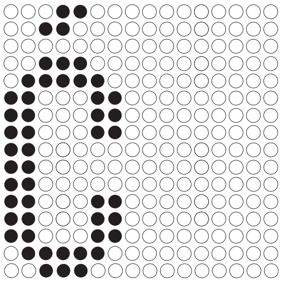

CHR170

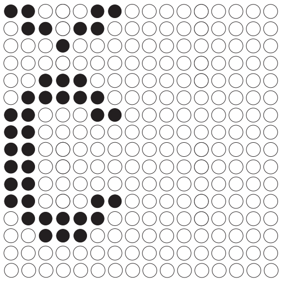

CHR171

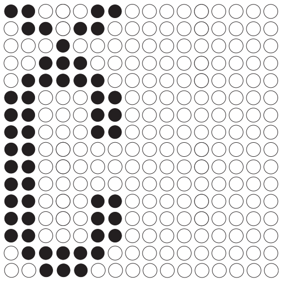

CHR172

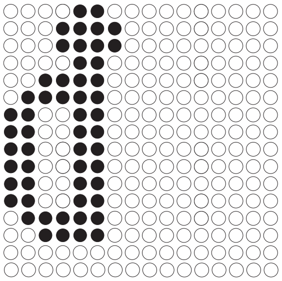

CHR173

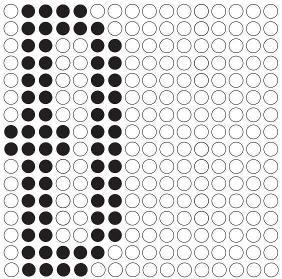

CHR174

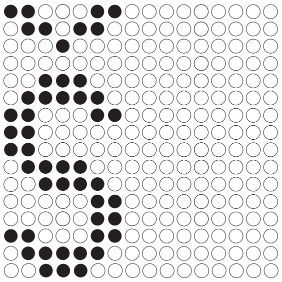

CHR175

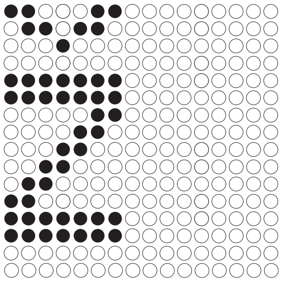

CHR176

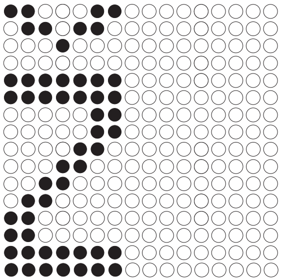

CHR177

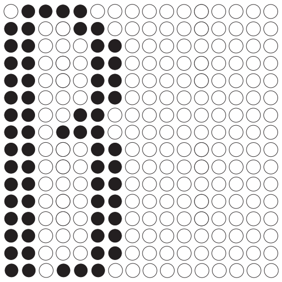

CHR178

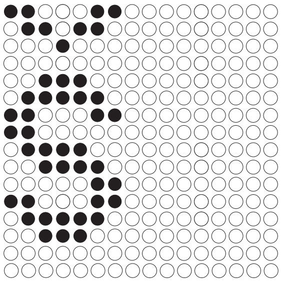

CHR179

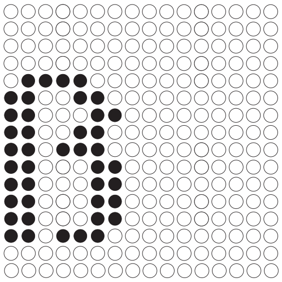

CHR180

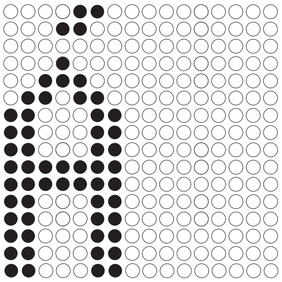

CHR181

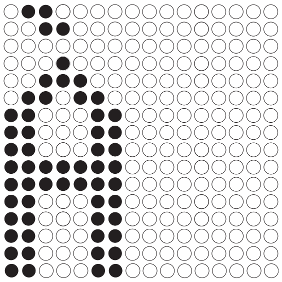

CHR182

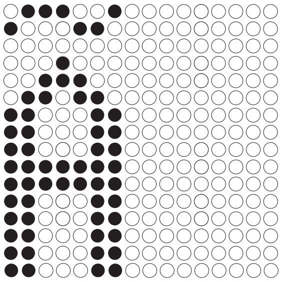

CHR183

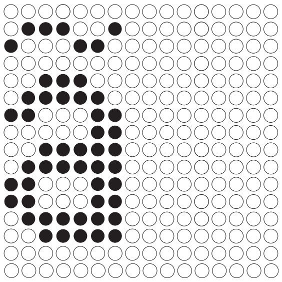

CHR184

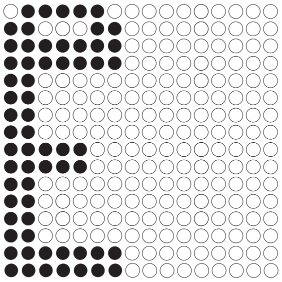

CHR185

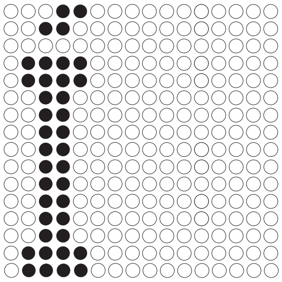

CHR186

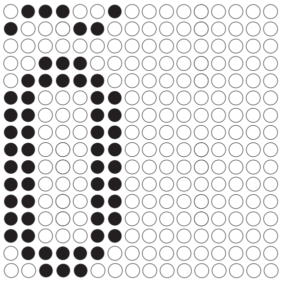

CHR187


CHR188

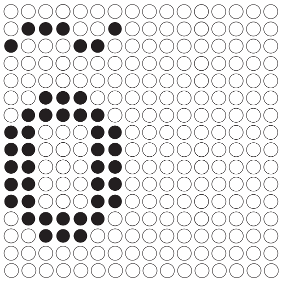

CHR189


CHR190


CHR191

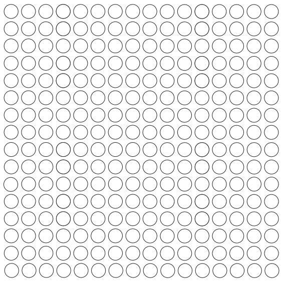


CHR192

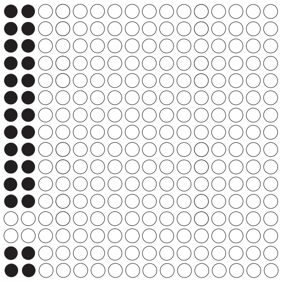


CHR193

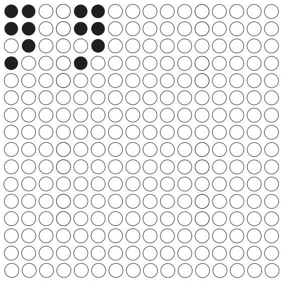
7.13.17 16-High Fancy (SF16)



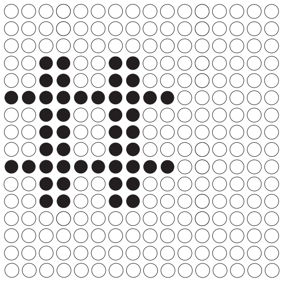
CHR032



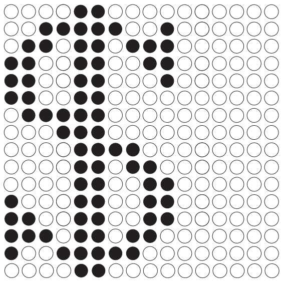
CHR033



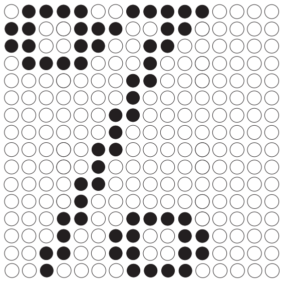
CHR034



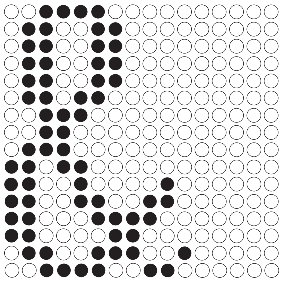
CHR035



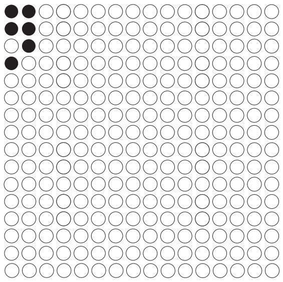
CHR036



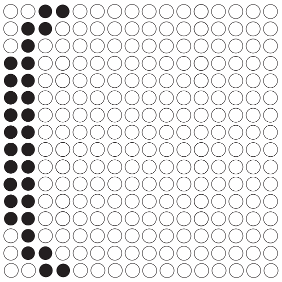
CHR037



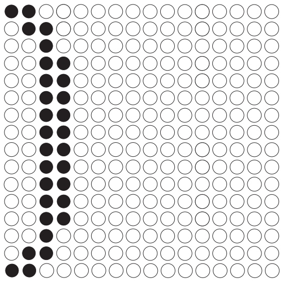
CHR038



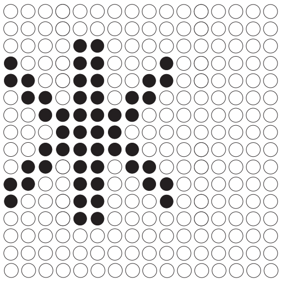
CHR039



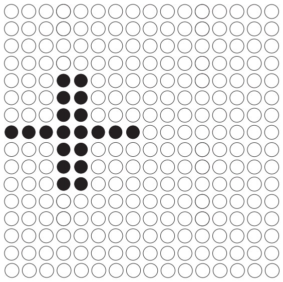
CHR040



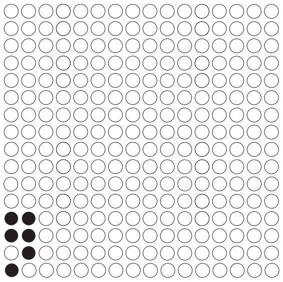
CHR041



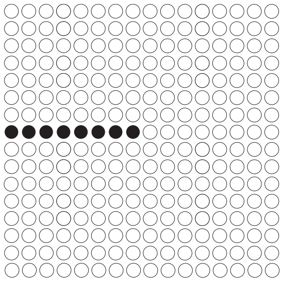
CHR042



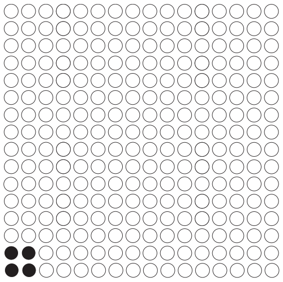
CHR043



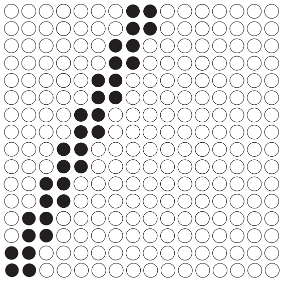
CHR044



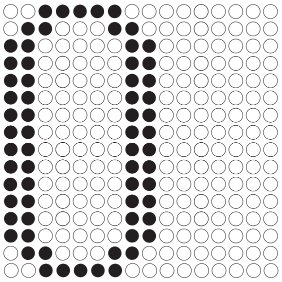
CHR045



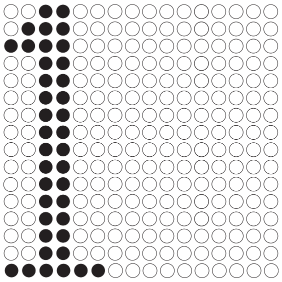
CHR046



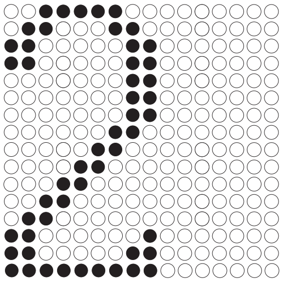
CHR047



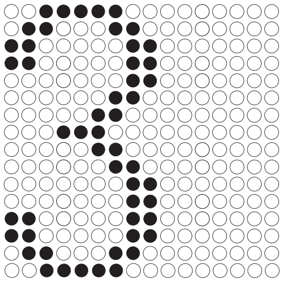
CHR048



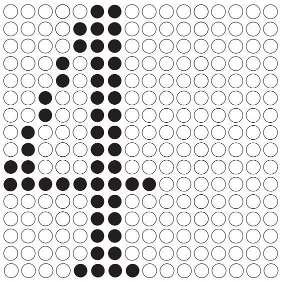
CHR049



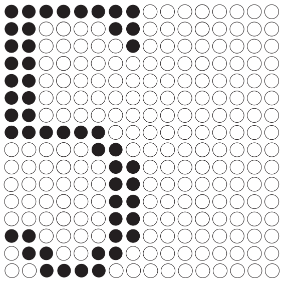
CHR050



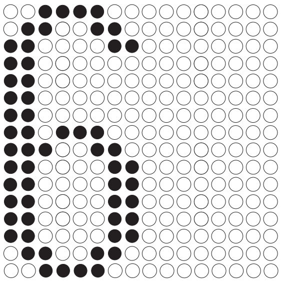
CHR051



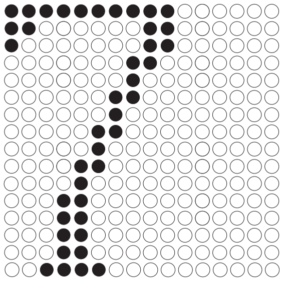
CHR052



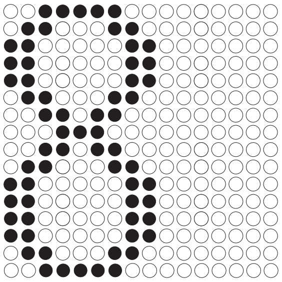
CHR053



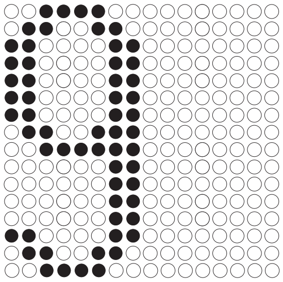
CHR054



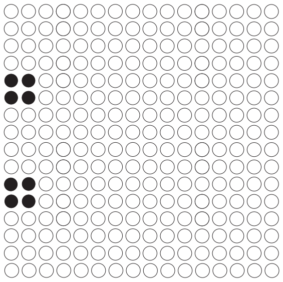
CHR055



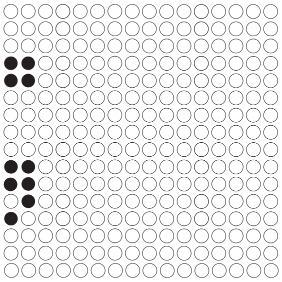
CHR056



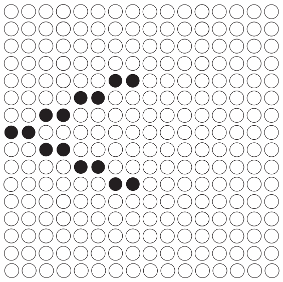
CHR057



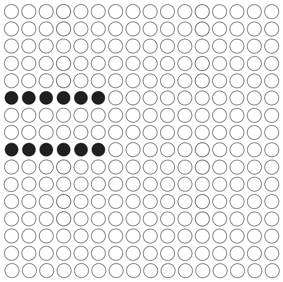
CHR058



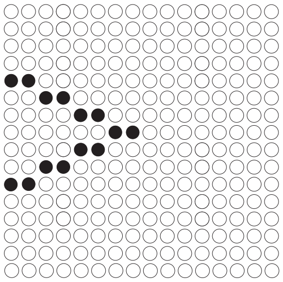
CHR059



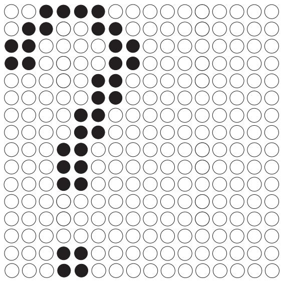
CHR060



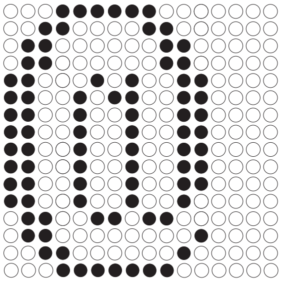
CHR061



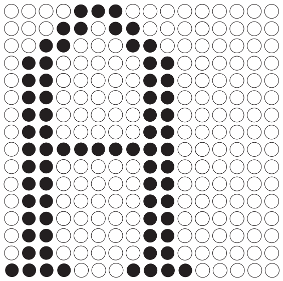
CHR062



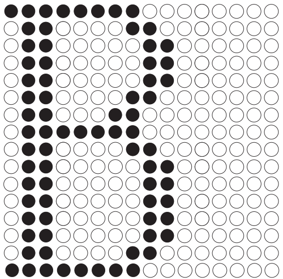
CHR063



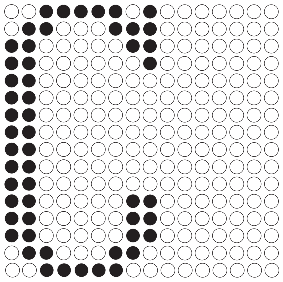
CHR064



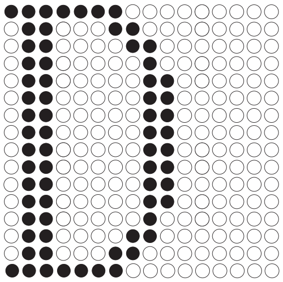
CHR065



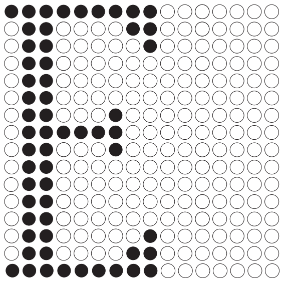
CHR066



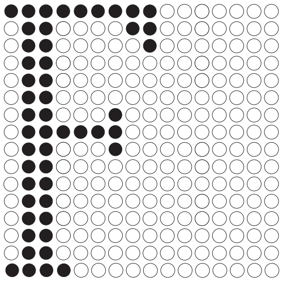
CHR067



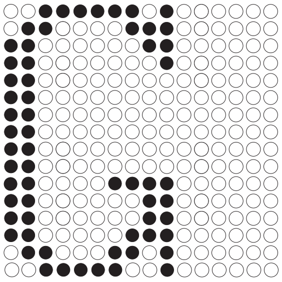
CHR068



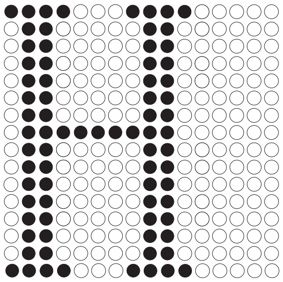
CHR069



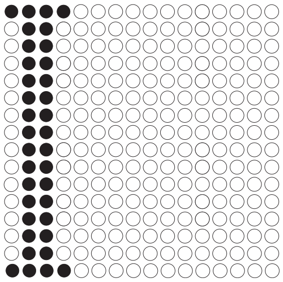
CHR070



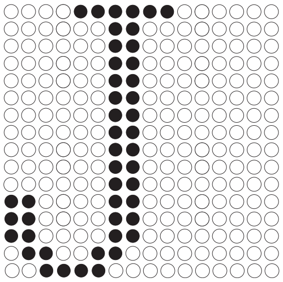
CHR071



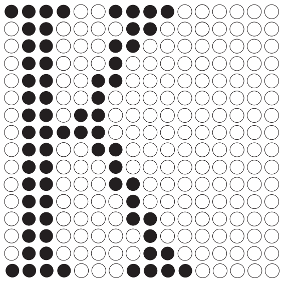
CHR072



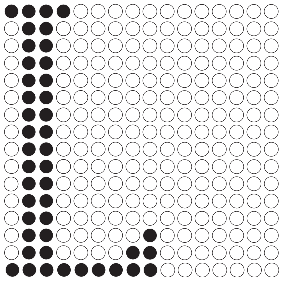
CHR073



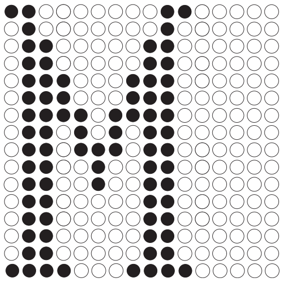
CHR074



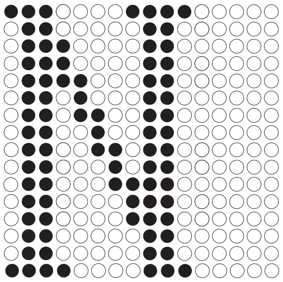
CHR075



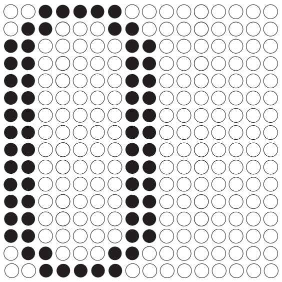
CHR076



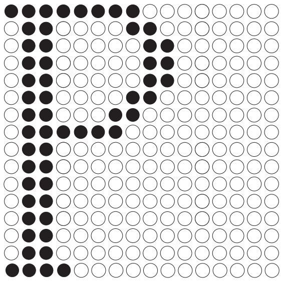
CHR077



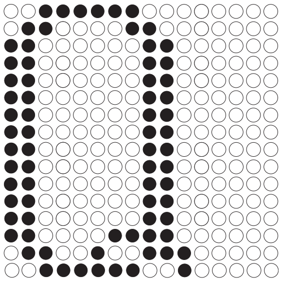
CHR078



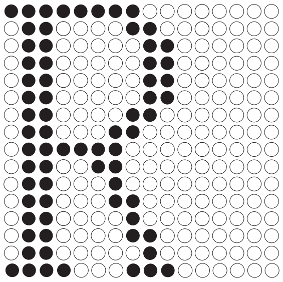
CHR079



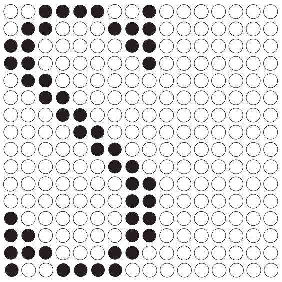
CHR080



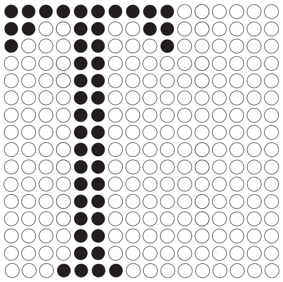
CHR081



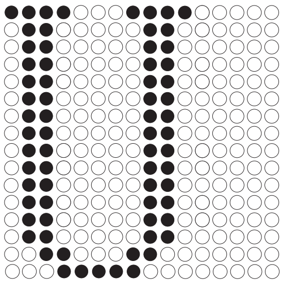
CHR082



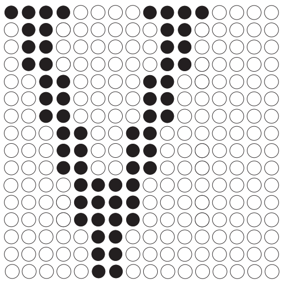
CHR083



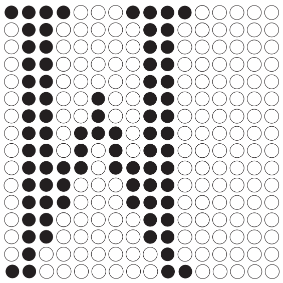
CHR084



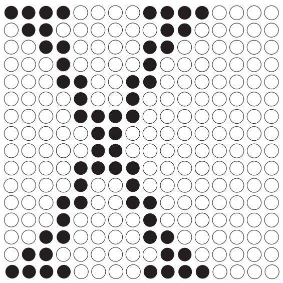
CHR085



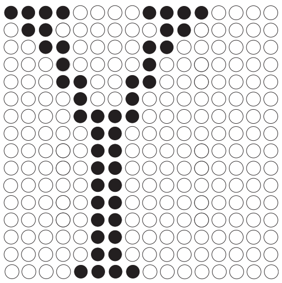
CHR086



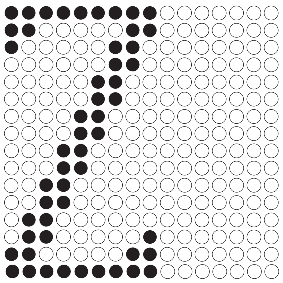
CHR087



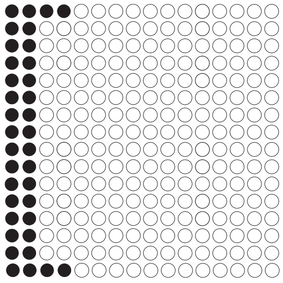
CHR088



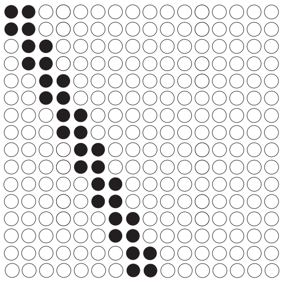
CHR089



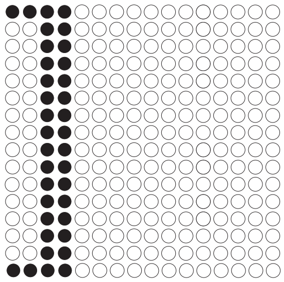
CHR090



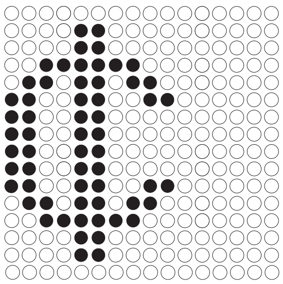
CHR091



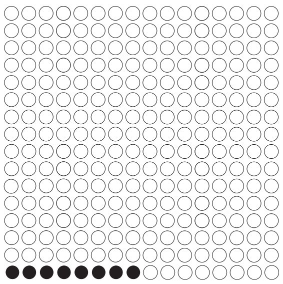
CHR092



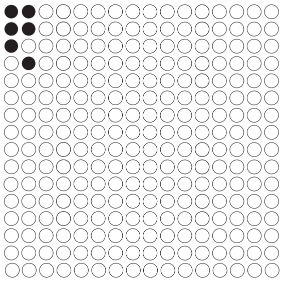
CHR093



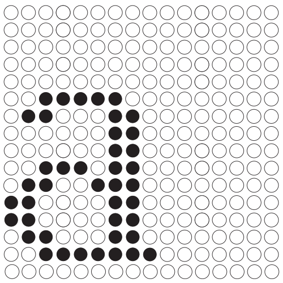
CHR094



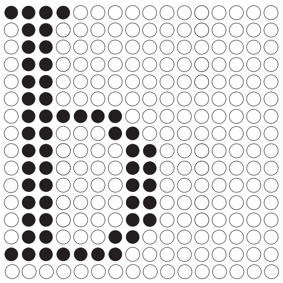
CHR095



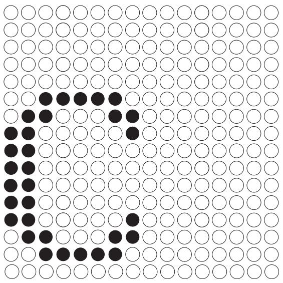
CHR096



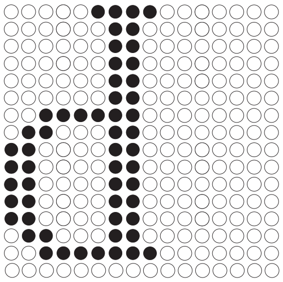
CHR097



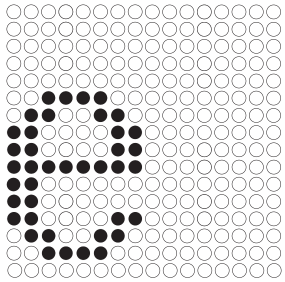
CHR098



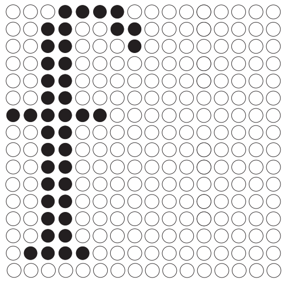
CHR099



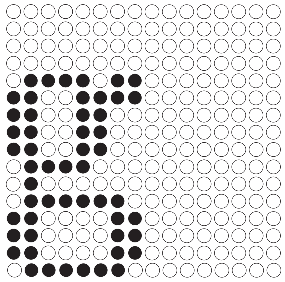
CHR100



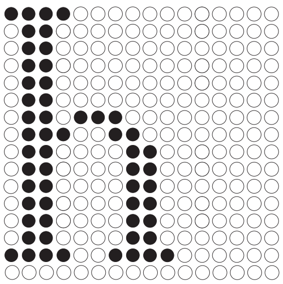
CHR101



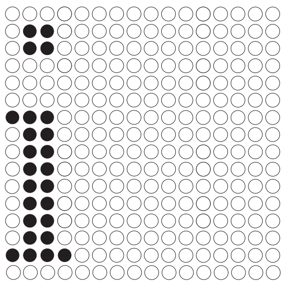
CHR102



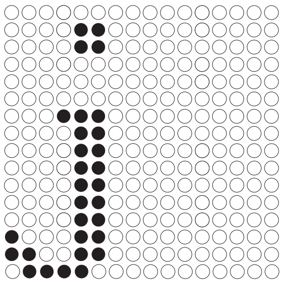
CHR103



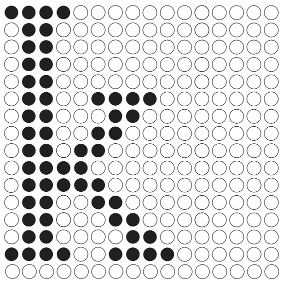
CHR104



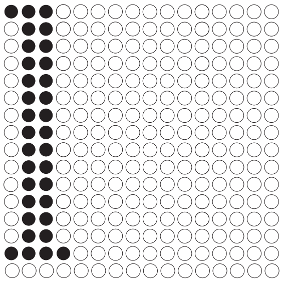
CHR105



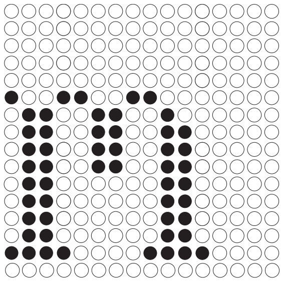
CHR106



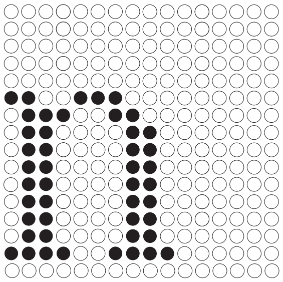
CHR107



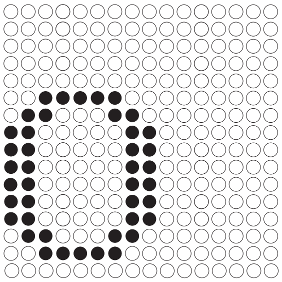
CHR108



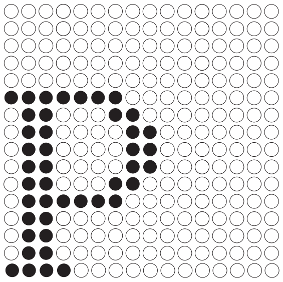
CHR109



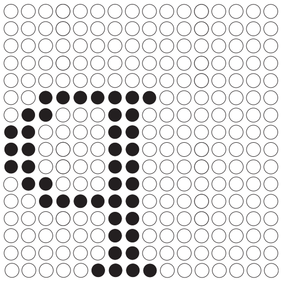
CHR110



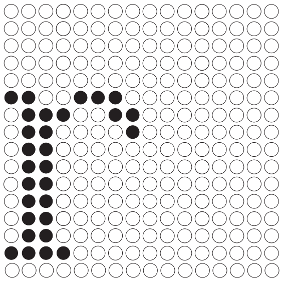
CHR111



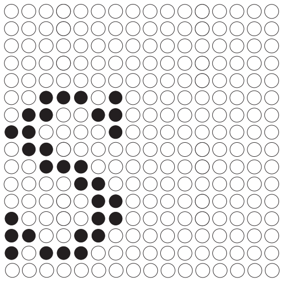
CHR112



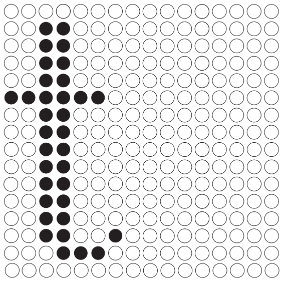
CHR113



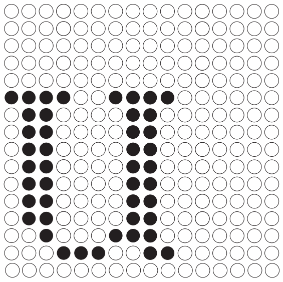
CHR114



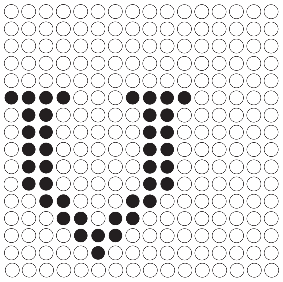
CHR115



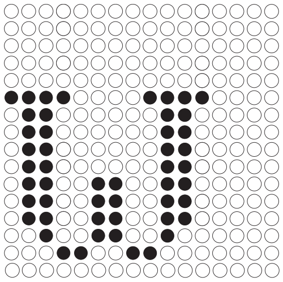
CHR116



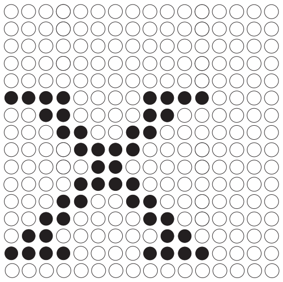
CHR117



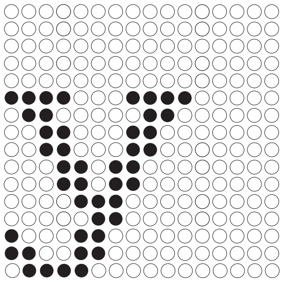
CHR118



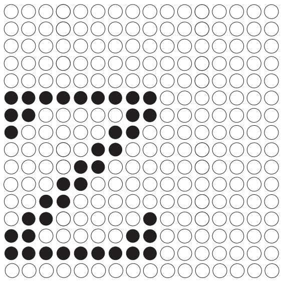
CHR119



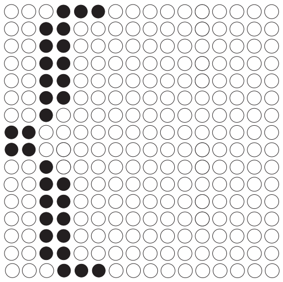
CHR120



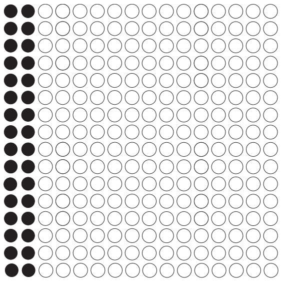
CHR121



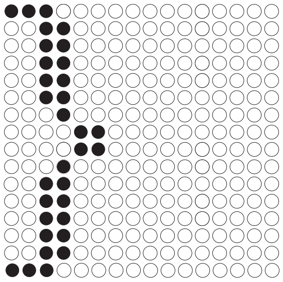
CHR122



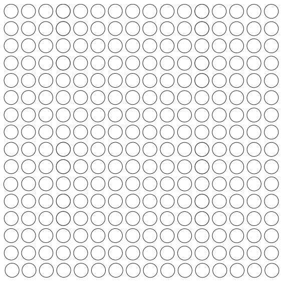
CHR123



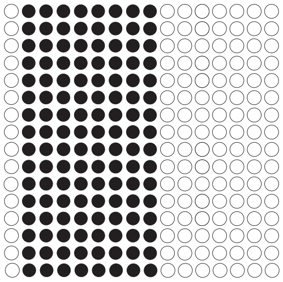
CHR124



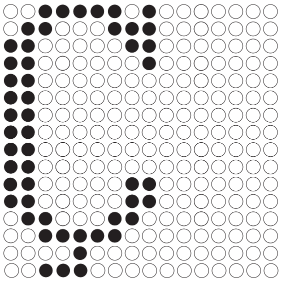
CHR125

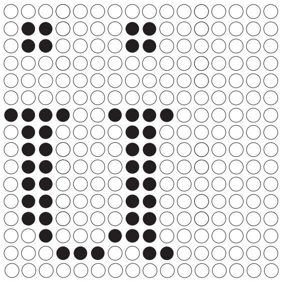


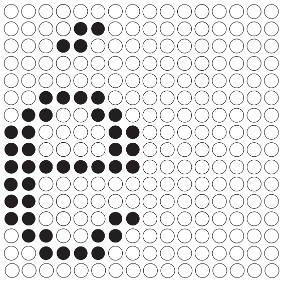
CHR126

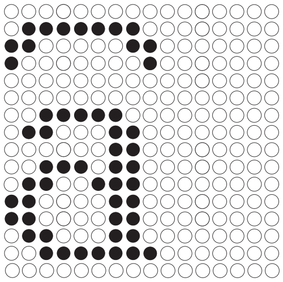


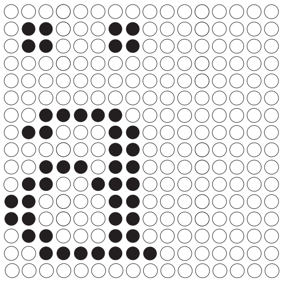
CHR127

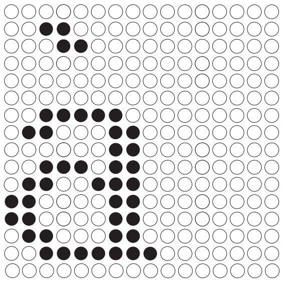

CHR128

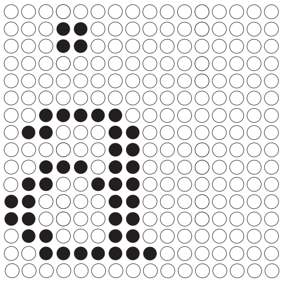

CHR129

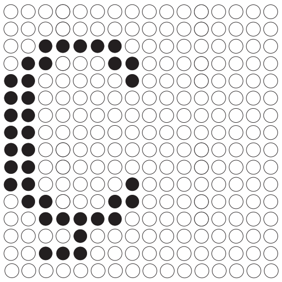

CHR130

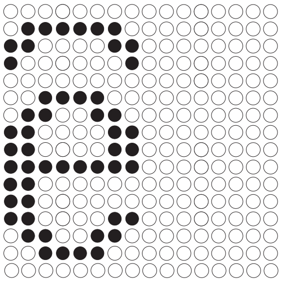

CHR131

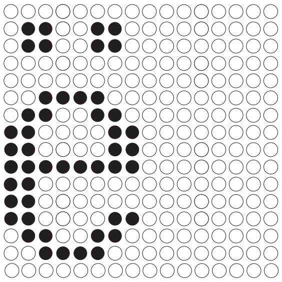

CHR132

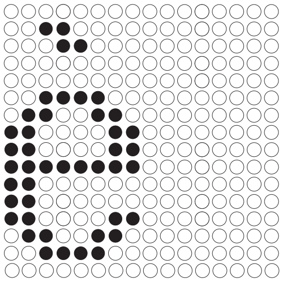

CHR133

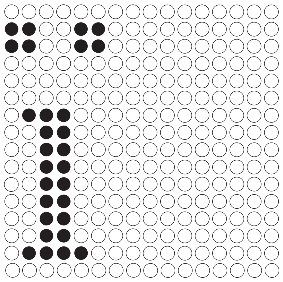

CHR134

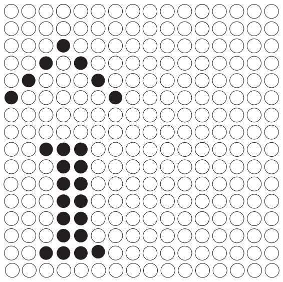

CHR135

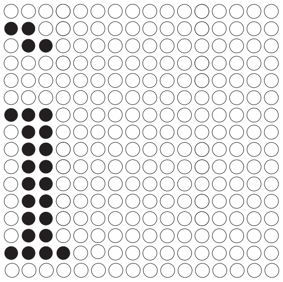

CHR136

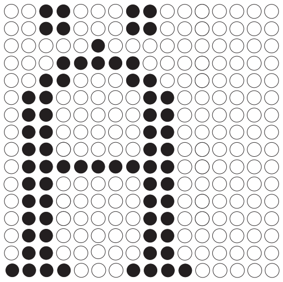

CHR137

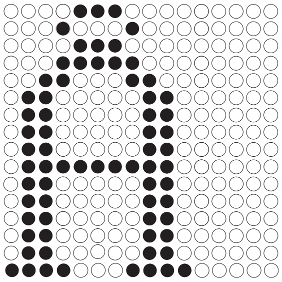

CHR138

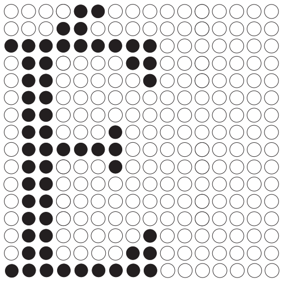

CHR139

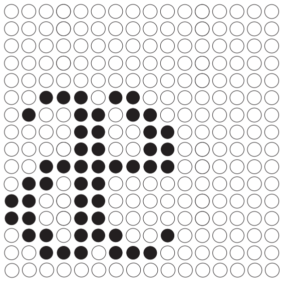

CHR140

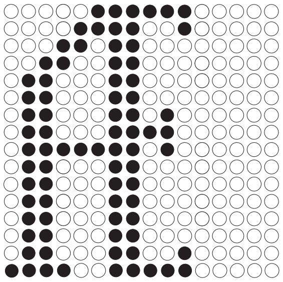

CHR141

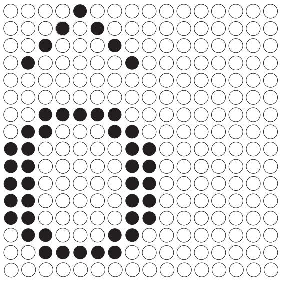

CHR142

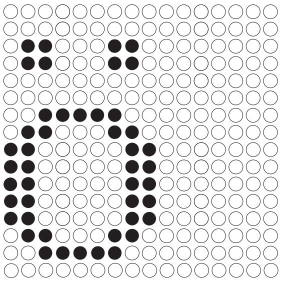

CHR143


CHR144

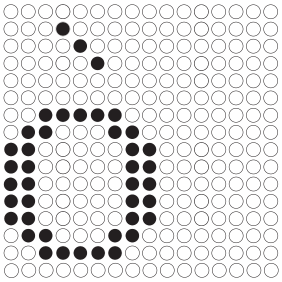

CHR145


CHR146

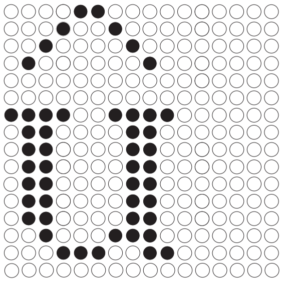

CHR147



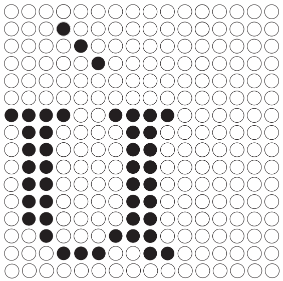
CHR148



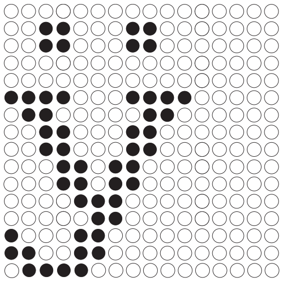
CHR149



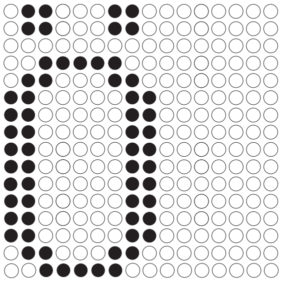
CHR150



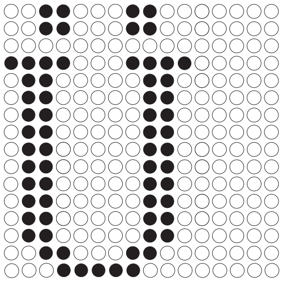
CHR151



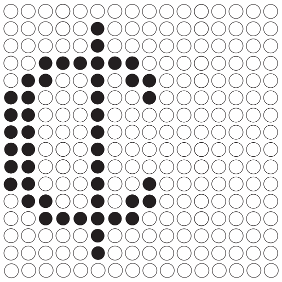
CHR152



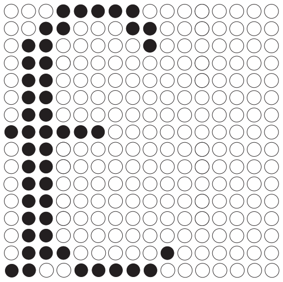
CHR153



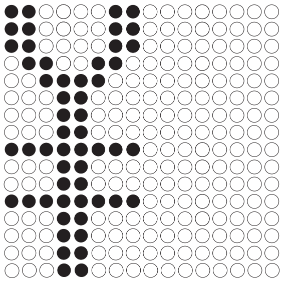
CHR154



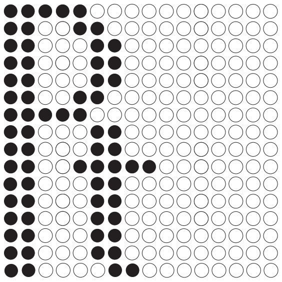
CHR155



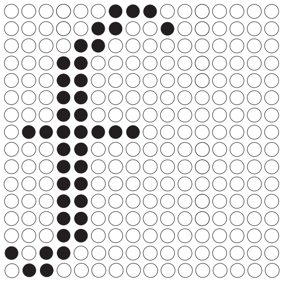
CHR156



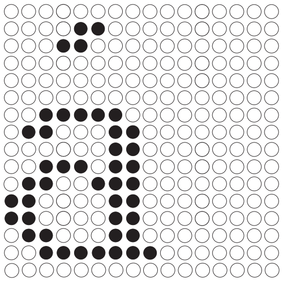
CHR157



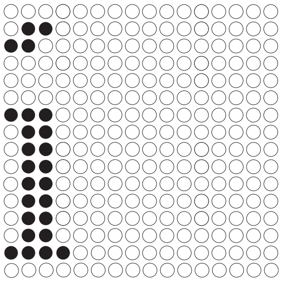
CHR158



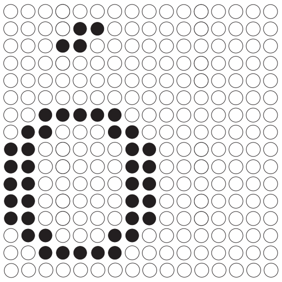
CHR159



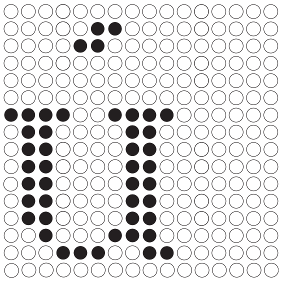
CHR160



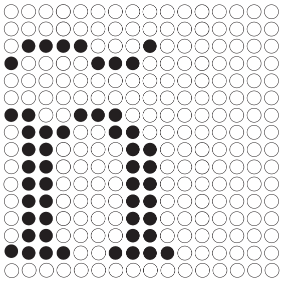
CHR161



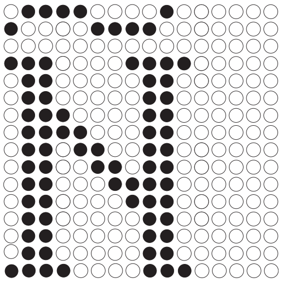
CHR162



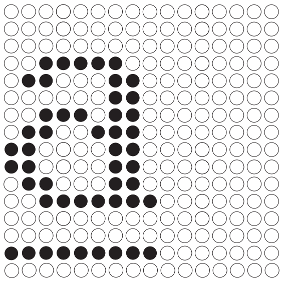
CHR163



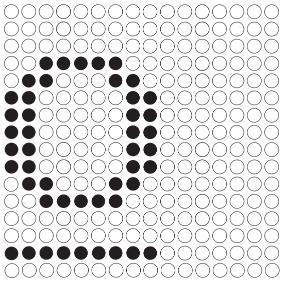
CHR164



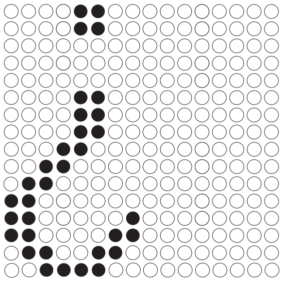
CHR165

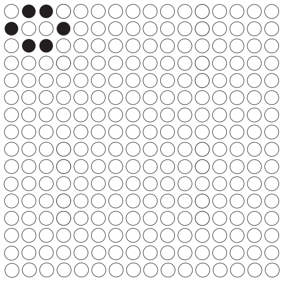


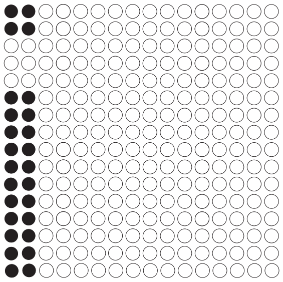
CHR166

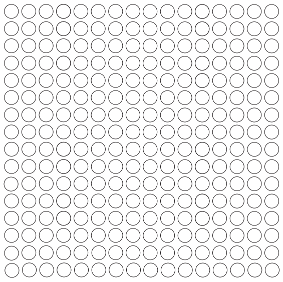


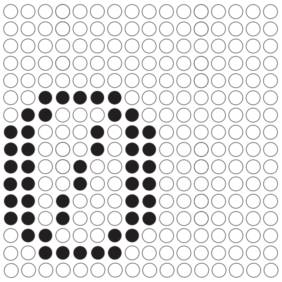
CHR167

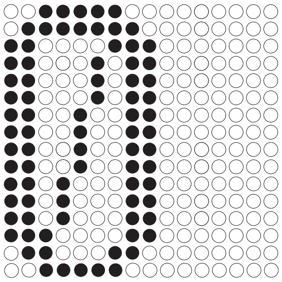

CHR168

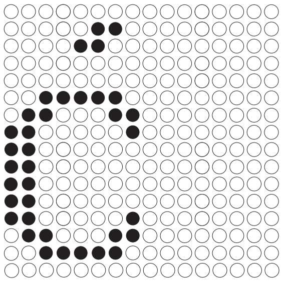

CHR169

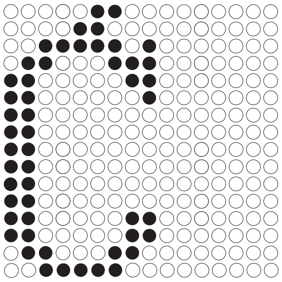

CHR170

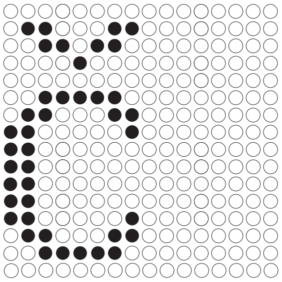

CHR171

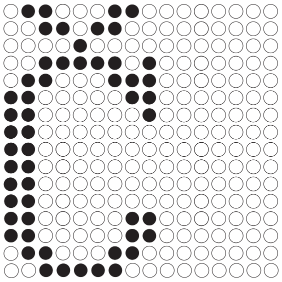

CHR172

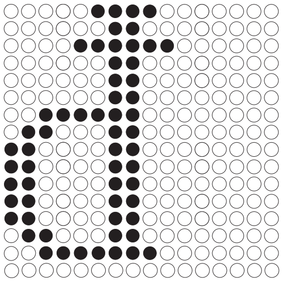

CHR173

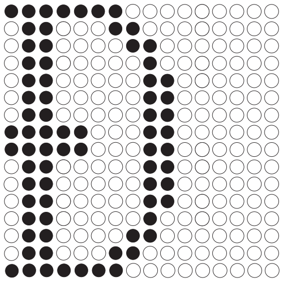

CHR174

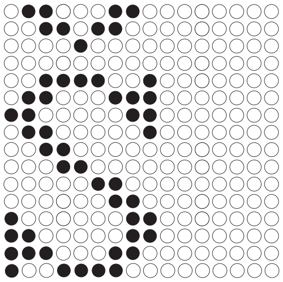

CHR175

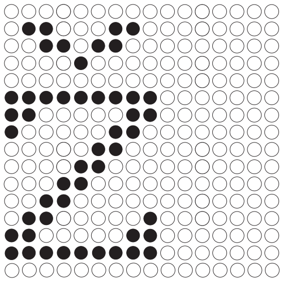

CHR176

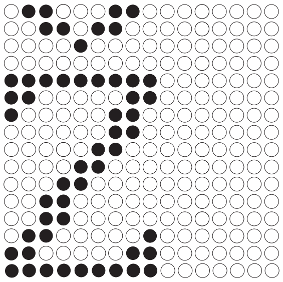

CHR177

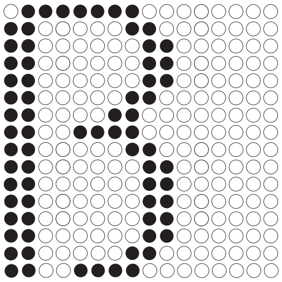

CHR178

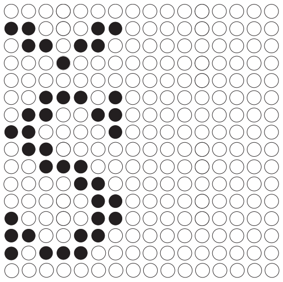

CHR179

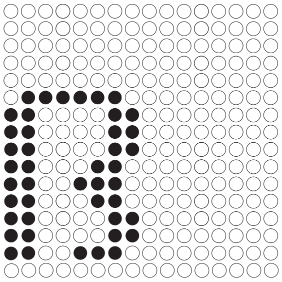

CHR180

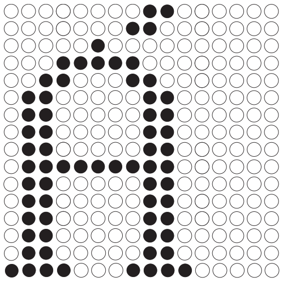

CHR181

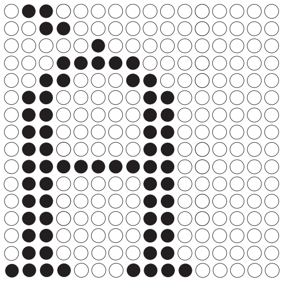

CHR182

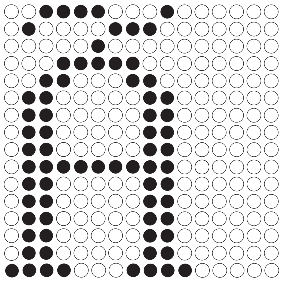

CHR183


CHR184

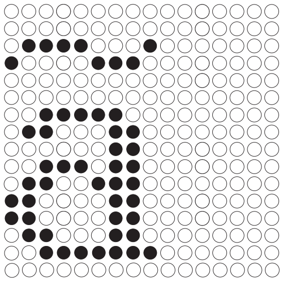

CHR185


CHR186

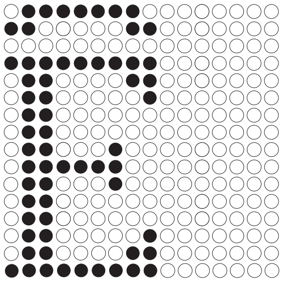

CHR187



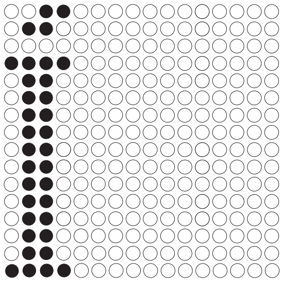
CHR188



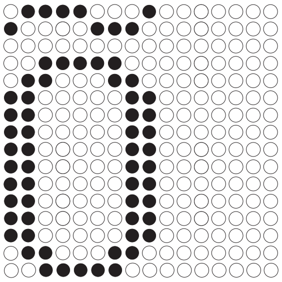
CHR189



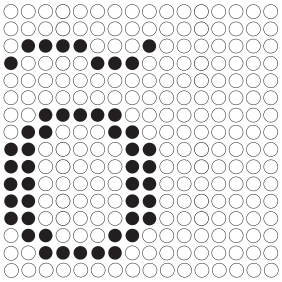
CHR190



CHR191

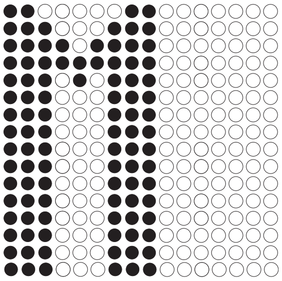


CHR192

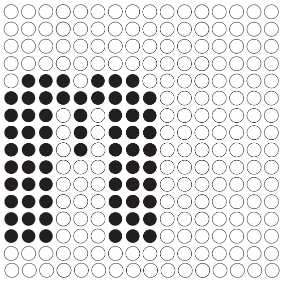


CHR193

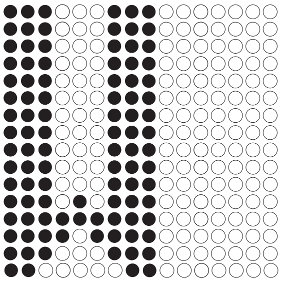
7.13.18 16-High Fat Character



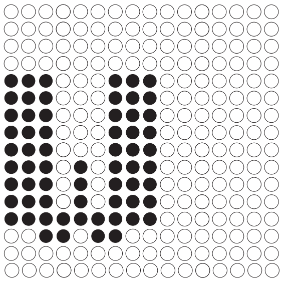
FATM



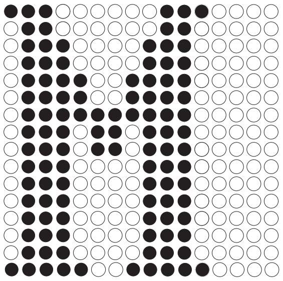
FATM1



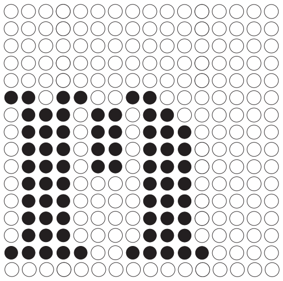
FATW



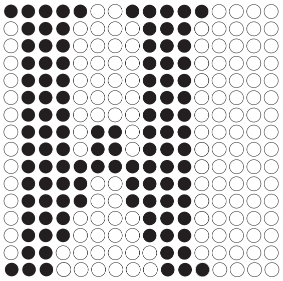
FATW1



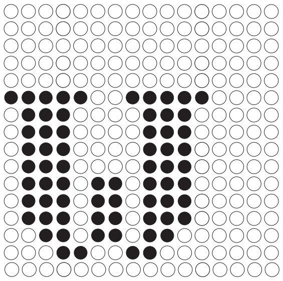
FFATM



FFATM1



FFATW



FFATW1