nascom

Nascom Microcomputer DOCUMENTATION

CP/M

ADVANCED VIDEO CARD

TERMINAL MANUAL

ISSUE 1.1.

Nascom Microcomputers
Division of Lucas Logic Limited
Welton Road Wedgnock Industrial Estate
Warwick CV34 5PZ
Tel: 0926 497733 Telex: 312333

Lucas Logic

Contents

Section 1
Introduction

The Advanced Video Controller allows the addition of a standard 80/40 column, 25 line display to the Nascom CP/M system giving access to a wide range of standard software.
Many advanced display features are supported along with enhanced keyboard operations.

Features

40 or 80 columns per line	Screen windowing
Inverse video	Programmable character set
Direct cursor addressing	Underlining
Screen editing	Function keys

Instructions for installing the AVC text software are given in APPENDIX A. Refer to the AVC hardware manual for instructions on installing the AVC card into a NASCOM microcomputer system.

CP/M devices

When the AVC text support software is loaded it configures itself as the User Console Device **UC1:** and becomes the active console. The Nascom 16 * 48 screen remains as the Console device **CRT:**.

Switching between the two devices is accomplished using the CP/M **STAT** command.

STAT CON:=CRT: selects the 16 * 48 screen

STAT CON:=UC1: selects the AVC screen

Section 2
Special keyboard functions

Local mode

A local mode is provide to allow characters entered at the keyboard to be sent directly to the display.
Local mode is selected by pressing **GRAPH and L** and is terminated in the same way. Whilst in local mode all screen control codes may be entered including escape sequences.

Screen editing

A screen edit mode is provided to allow for speedy editing and re-entry of command lines. This mode is invoked by pressing **GRAPH and S.** The terminal will enter local edit mode as described in the previous section except that editing will be terminated by the **ENTER** key.
When **ENTER** is pressed the complete line containing the cursor is returned except for the prompts issued by CP/M (>A, >B * - #) which are deleted.

Echo to LST:

The key combination **GRAPH and E** will cause all following characters to be echoed to the LST: device with no checking for control functions. This allows direct access to any special functions that may be supported by the printer such as bold or variable character pitch. This mode is terminated when **ENTER** is pressed.

Screen dump

The key combination **GRAPH and** D will cause the contents of the screen to be sent to the LST: device which is normally a printer.

Shift toggle

The key combination **CTRL and ENTER** toggles the action of the shift key.

Function keys

The keys **'0' to '9'** when used in conjunction with the **GRAPH** key will act as ten programmable function keys. A character string of 1 to 10 characters is associated with each of these keys. The keys are defined as having the following functions by default but may be changed under software or keyboard control. (see section 3)

1 - DIR A:	**2 - DIR B:**	**3 - DIR C:**	**4 - DIR D:**
5 - STAT	**6 - DDT**	**7 - PIP**	**8 - MBASIC**
9 - FILES	**0 - SYSTEM**		

Section 3
Terminal functions

The Advanced Video Controller will accept both single character control codes and multiple character escape sequences in order to provide a wide range of terminal functions.
Single character control codes have ASCII values from 0 to 19 hex (0 to 31 decimal). They may be generated from the keyboard by pressing the CTRL key along with one of the alphanumeric keys as shown below.
Multiple character escape sequences are initiated by sending the ESC character (1B hex, 27 decimal) followed by an alphanumeric character. Some escape sequences may require other characters to follow which are indicated where required.
If a control character (0 to 19 hex) is preceded by an escape character (1B hex) then character will be displayed on the screen and its control function ignored.
NOTE: nested escape sequences are not permitted.

Control codes

08H ^H Backspace

The cursor is moved left one position. The character now under the cursor is erased. If the cursor is in the home position then the code is ignored.

0AH ^J Linefeed

The cursor is moved down one line If the cursor is already on the bottom line then the screen is scrolled up one line and the bottom line erased.

0CH ^L Form feed/Clear screen

The cursor is moved to the home position and the screen is cleared.

0DH ^M Carriage return

The cursor is moved to the first position in the current line.

12H ^R Erase to end of line

The character under the cursor and all the characters to the right of it in the current line are erased.

13H ^S Erase to end of screen

The character under the cursor and all the characters to the right of it and below it are erased.

14H ^T Delete line

The cursor line is erased and all the lines below it scrolled up. A blank is inserted at the bottom of the screen.

15H ^U Insert line

The cursor line and all the lines below it are scrolled down. A blank line is inserted and the bottom line is lost.

16H ^V Delete character

The character under the cursor is deleted and all the characters to the right of it are shifted left one place a space being inserted in the last character position.

17H ^W Insert character

A space is inserted in the current cursor position. The cursor character and all the characters to the right of it are shifted right one place. The last character on the line is lost.

18H ^X Erase line

The line containing the cursor is erased and the cursor moved to the first position in the line.

19H ^Y Cursor home

The cursor is moved to the first row and column in the screen.

1BH ^[Escape

First character of an escape sequence.

1CH ^\ Cursor left

The cursor is moved one position to the left. If it is at the start of a line then it is moved to the end of the line above. If it is in the home position then no action is taken.

1DH ^] Cursor right

The cursor is moved one position to the right. If it is at the end of a line then it is moved to the start of the next line. The screen is scrolled up if required.

1EH ^^ Cursor up

The cursor is moved up one line. If it is on the top line then no action is taken.

1FH ^_ Cursor down

The cursor is moved down one line. If it is on the bottom line then no action is taken.

Escape codes

ESC = Set cursor position

The following two characters define the row and column to which the cursor is subsequently moved. The row number is expected first followed by the column number as ASCII values offset by 20 hex (See table 1).

ESC ? Report cursor position

The next three characters read from the console will be the row and column numbers of the current cursor position followed by a carriage return. The row number is returned first followed by the column number as ASCII values offset by 20 hex (See table 1).

ESC A 80 column mode

The line length is set to 80 characters and the screen is cleared. (see ESC B)

ESC B 40 column mode

The line length is set to 40 characters and the screen is cleared. (see ESC A)

ESC C Alternate character set

The characters displayed for codes 0 - 127 will be from the alternate character set. Character codes 128 - 255 will display characters from the standard character set. If the alternate character set is the inverse of the standard set (ESC P) then this code will cause the characters to appear in inverse video. (see ESC D)

ESC D Standard character set

The characters displayed for codes 0 - 127 will be from the standard character set. Character codes 128 - 255 will display characters form the alternate character set. (See ESC C)

ESC E Read cursor line

Characters subsequently requested from the console will be read from the current cursor line on the screen. The line will be terminated by a carriage return. Trailing spaces and non-ASCII characters will be deleted.

ESC F Read keyboard line

This is similar to the previous command except that screen editing is invoked and the cursor line returned when the ENTER key is pressed. (see ESC E)

ESC G Reverse line feed

The cursor is moved up the screen one line. If it was on the top line then the screen is scrolled down and the top line erased.

ESC J Enable wraparound

When the cursor is moved beyond the last character position in a line then it will move to the first character position in the next line. The screen will be scrolled if required. (see ESC K)

ESC K Disable wraparound

The cursor will stay in the last character position in a line until a cursor control code moves it from there. The character will be overwritten if any more displayable characters are received. This mode is used primarily to prevent the screen scrolling when the last character on the last line is displayed. (see ESC J)

ESC L Single line cursor

The cursor is displayed on the screen as a single line underneath the character. (see ESC M)

ESC M Block cursor

The cursor is displayed on the screen as a block the full height of a character. (see ESC L)

ESC N Overwrite cursor

The cursor will overwrite the character currently on the screen in inverse video. (see ESC O)

ESC O Invert cursor

The cursor will invert the character currently on the screen. (see ESC N)

ESC P Invert character set

The alternate character set is made to be the inverse of the standard character set. Selecting the alternate character set will now select inverse video. (see ESC C)

ESC Q Inverse video

The screen cleared and set to inverse video mode. (see ESC R)

ESC R Normal video

The screen is cleared and set to normal video mode. (see ESC Q)

ESC S Enable cursor

The cursor appears on the screen at the current cursor position. (see ESC T)

ESC T Disable cursor

The cursor is not displayed on the screen. (see ESC S)

ESC U Enable text display

The text is displayed on the screen. (see ESC V)

ESC V Disable text display

The text is not displayed on the screen but is received and processed as normal. (see ESC U)

ESC W Enable underlining

All characters subsequently displayed on the screen will be underlined. (see ESC X)

ESC X Disable underlining

Turns off the underlining. (see ESC W)

ESC Y Define window

A scrolling window is defined on the screen. The first line of the window is the current cursor line. The number of scrolled lines is sent next offset by 20h. (see table 1)

ESC Z Define character

The dot matrix displayed for a particular ASCII code may be defined using this escape sequence. The next character code received will be the ASCII code of the character to be defined. The next ten codes will be the bit patterns for successive scan lines of the display starting with the top line. (see Appendix E)

ESC [Reset

The AVC parameters are set to the default values and the entire display memory is erased. The current character set is left unchanged.

ESC \ Define key

The key character string returned by any of the function keys is defined using this facility. The next character is the number of characters to be assigned to the key offset by 20h (see table 1). The maximum number of characters allowed is ten. The following characters define the string to be used.

ESC] Page mode

This code disables the normal scrolling of the screen. When entry continues beyond the last character on the bottom line the display is halted. When a key is pressed and then the screen is cleared and display continues from the home position. (see ESC ^) This mode is useful for quickly scanning through a text file.

ESC ^ Scroll mode

Normal scrolling is resumed. (see ESC])

Char	ASCII hex	Number dec	Char	ASCII hex	Number dec	Char	ASCII hex	Number dec
SP	20	0	;	3B	27	V	56	54
!	21	1	<	3C	28	W	57	55
"	22	2	=	3D	29	X	58	56
#	23	3	>	3E	30	Y	59	57
$	24	4	?	3F	31	Z	5A	58
%	25	5	@	40	32	[	5B	59
&	26	6	A	41	33	\	5C	60
'	27	7	B	42	34	]	5D	61
(	28	8	C	43	35	^	5E	62
)	29	9	D	44	36	_	5F	63
*	2A	10	E	45	37	`	60	64
+	2B	11	F	46	38	a	61	65
,	2C	12	G	47	39	b	62	66
-	2D	13	H	48	40	c	63	67
.	2E	14	I	49	41	d	64	68
/	2F	15	J	4A	42	e	65	69
0	30	16	K	4B	43	f	66	70
1	31	17	L	4C	44	g	67	71
2	32	18	M	4D	45	h	68	72
3	33	19	N	4E	46	i	69	73
4	34	20	O	4F	47	j	6A	74
5	35	21	P	50	48	k	6B	75
6	36	22	Q	51	49	l	6C	76
7	37	23	R	52	50	m	6D	77
8	38	24	S	53	51	n	6E	78
9	39	25	T	54	52	o	6F	79
:	3A	26	U	55	53	p	70	80

TABLE 1. ASCII code/Decimal number conversion

The table shows the characters used to represent decimal numbers when offset by the ASCII value of a space (20 hex). For the ASCII code in decimal add 32 to the decimal number.

Appendix A
Installing the AVC control software

The AVC text handling routines and character set are contained in the file **AVCTXT.COM.** The AUTO.SUB file supplied may be used to auto load the software if required.

IMPORTANT The AVC text routines require **CP/M 2.2 Nascom Rev 2.1** or later. They are incompatible with CP/M 2.2 Nascom Rev 1.1 and other CP/M implementations on the Nascom.

The AVC control software occupies 4.5 kilobytes of memory. It is located in reserved memory above the CP/M operating system. Memory must be reserved for this by using MOVCPM to create a CP/M system smaller than the memory available. The size of system to be created can be calculated as follows:-
Subtract 4.5 from the memory available in kilobytes and round this down to the next kilobyte. Create a system of this size using MOVCPM as described in the CP/M documentation. (see table 2)

The control software is self relocating ie it configures itself for the memory available. The base address is determine by the free memory pointer maintained by the CP/M system. After relocation the pointer is updated to reflect the new top of memory.

It is usually most convenient for the AVC software to Auto load upon cold boot. This is accomplished by using CONFIG to select the auto load option and using the supplied AUTO.SUB file as described in the CP/M documentation.

The AVC is normally set up for a base address of 8000 hex. The AVC text software will however configure itself for AVC's based at 0000 hex, 4000 hex or 8000 hex as may be required. The base address must however be selected to avoid any overlap between the AVC memory and the text handling routines in main memory. (see table 2)

The software assumes that the Nascom 2 video has been connected to the external video input of the AVC card. This allows switching between the two video system on one monitor. Instructions for accomplishing this are contained in the **AVC hardware manual.**

Table 2. Usual Memory allocation.

Memory size	CP/M size	AVC base address
32k	27k	8000h or 0000h
48k	43k	4000h or 0000h
60k	55k	8000h or 4000h or 0000h

Four column directory display

The DIR command only uses two columns to display the information. This may be changed to give a four column display by using DDT or

ZSID to modify the MOVCPM.COM file as follows.

Change locations 0EB2h and 0EE7h from 01h to 03h.

Appendix B
Error messages

The following error messages may be encountered whilst the AVC software is installing itself.

NO AVC PRESENT

This message would result if no AVC card is present or if it has been incorrectly installed in the system. For installation details see the **AVC HARDWARE MANUAL.**

INSUFFICIENT MEMORY RESERVED FOR AVC

This message would result if insufficient memory has been reserved above the CP/M operating system for the AVC control software.

MEMORY CONFLICT

This message would result if the AVC text routines are overlapped by the AVC on board memory.

Appendix C
Customising the AVC control software

Many of the AVC parameters may be changed to allow tailoring of the software to suit individual requirements.
These changes are effected by using DDT or ZSID on the AVC.COM file and then saving the modified version.

NOTE It is preferable to change the application software if at all possible rather than change the terminal functions.

Patchable locations

CRT Controller initialization values

These values should not require changing unless difficulty is experienced with the particular display device in use. Further details may be found in the AVC hardware manual.

name	location	value	description
HTOT	0A0C	129	Horizontal total - 1
HDISP	0A0D	128	Horizontal displayed
HSINC	0A0E	110	Horizontal sync position
SYNWID	0A0F	9	Sync width
VTOT	0A10	76	Vertical total
VADJ	0A11	3	Vertical adjust
VDISP	0A12	64	Vertical displayed
VSYNC	0A13	68	Vertical sync position
INT	0A14	0	Interlace mode
LPC	0A15	3	Scan lines per character - 1
CST	0A16	0	Cursor start
CEN	0A17	1	Cursor end
MSTART	0A18/9	0	Memory start address

Control and Escape code table

This table may be changed if it is required to change the control codes used to accomplish a particular terminal function.
Each byte value in the code table has a corresponding two byte address in the jump vector table pointing to the handling routine for the function.
Entry values from 0 to 1F hex are used where single byte control codes are used. Values from 1F to 7F hex correspond to the second byte of a two byte escape code. The first byte normally being ESC 1B hex but this may be changed by patching the appropriate location.

name	location	value	description
CTLTAB	0A1A	47	Number of control codes in table
CTLTB	0A1B	0Dh ^M	Carriage return
	0A1C	0Ah ^J	Line feed
	0A1D	08h ^H	Back space
	0A1E	0Ch ^L	Form feed
	0A1F	16h ^V	Delete character
	0A20	17h ^W	Insert character

name	location	value	description
	0A21	19h ^Y	Home
	0A22	1Bh ^[	Escape
	0A23	1Ch ^\	Cursor left
	0A24	1Dh ^]	Cursor right
	0A25	1Eh ^^	Cursor up
	0A26	1Fh ^_	Cursor down
	0A27	18h ^X	Erase line
	0A28	41h A	80 column mode
	0A29	42h B	40 column mode
	0A2A	43h C	Alternate character set
	0A2B	44h D	Standard character set
	0A2C	12h ^R	Erase to end of line
	0A2D	13h ^S	Erase to end of screen
	0A2E	3Dh =	Set cursor position
	0A2F	47h G	Reverse line feed
	0A30	15h ^U	Insert line
	0A31	14h ^T	Delete line
	0A32	4Ah J	Enable wrap around
	0A33	4Bh K	Disable wrap around
	0A34	4Ch L	Line cursor
	0A35	4Dh M	Block cursor
	0A36	4Eh N	Overwrite cursor
	0A37	4Fh O	Invert cursor
	0A38	50h P	Invert character set
	0A39	51h Q	Invert video
	0A3A	52h R	Normal video
	0A3B	53h S	Enable cursor
	0A3C	54h T	Disable cursor
	0A3D	55h U	Enable text display
	0A3E	56h V	Disable text display
	0A3F	57h W	Underline on
	0A40	58h X	Underline off
	0A41	59h Y	Set window
	0A42	5Ah Z	Define character
	0A43	5Bh [	Initialise AVC
	0A44	5Ch \	Define function key
	0A45	5Dh]	Page mode
	0A46	5Eh ^	Scroll mode
	0A47	3Fh ?	Cursor position report
	0A48	45h E	Get keyboard line
	0A49	46h F	Get screen line
VECTTB	0A4A-0AA7		Jump vectors for control codes

Special key function table

The key combinations used to activate the special key functions may be changed by patching the following table. Any of the functions may be dissabled by inserting a value of 0 where required.

name	location	value	description
SKEYTB	0AA8	4	Number of special keys
SKEYT	0AA9	F3h gra S	Screen edit
	0AAA	E5h gra E	Echo to LST:
	0AAB	F6h gra D	Screen dump
	0AAC	ECh gra L	Line/Local
SVECTB	0AAD-0AB4		Jump vectors for special keys

Function key table

The character strings returned by each of the function keys may be changed by patching this table. The last character (which may or may not be a carriage return) must have its most significant bit set.

name	location	value	description
KEYTBL	0AB5-0ABE	"SYSTEM"	Function 0
	0ABF-0AC8	"DIR A:"	Function 1
	0AC9-0AD2	"DIR B:"	Function 2
	0AD3-0ADC	"DIR C:"	Function 3
	0ADD-0AE6	"DIR D:"	Function 4
	0AE7-0AF0	"STAT"	Function 5
	0AF1-0AFA	"DDT"	Function 6
	0AFB-0B04	"PIP"	Function 7
	0B05-0B0E	"MBASIC"	Function 8
	0B0F-0B18	"SYSTEM"	Function 9

Other parameters and flags

name	location	value	description
FLADDR	0B19/A	4144H	First line address
WORKTB	0B1B	28h	Initial value for AVC control port
	0B1C	79	Last column number
	0B1D	0	Initial column position
	0B1E	0	Initial row position
	0B1F	0	First line
	0B20	24	Last line
	0B21	1	1 - Line cursor
			10 - Block cursor
	0B22	00h	00h - Normal video
			FFh - Inverse video
	0B23	0	00h - Standard character set
			80h - Alternate character set
	0B24	FFh	00h - Wraparound disabled
			FFh - Wraparound enabled
	0B25	00h	00h - Overwrite cursor
			FFh - Invert cursor
	0B26	FFh	00h - Cursor disabled
			FFh - Cursor enabled

name	location	value	description
	0B27	00h	00h - Underline disabled FFh - Underline enabled
	0B28	00h	00h - Scroll mode FFh - Page mode
	0B29-30		Miscellaneous

Character set tables

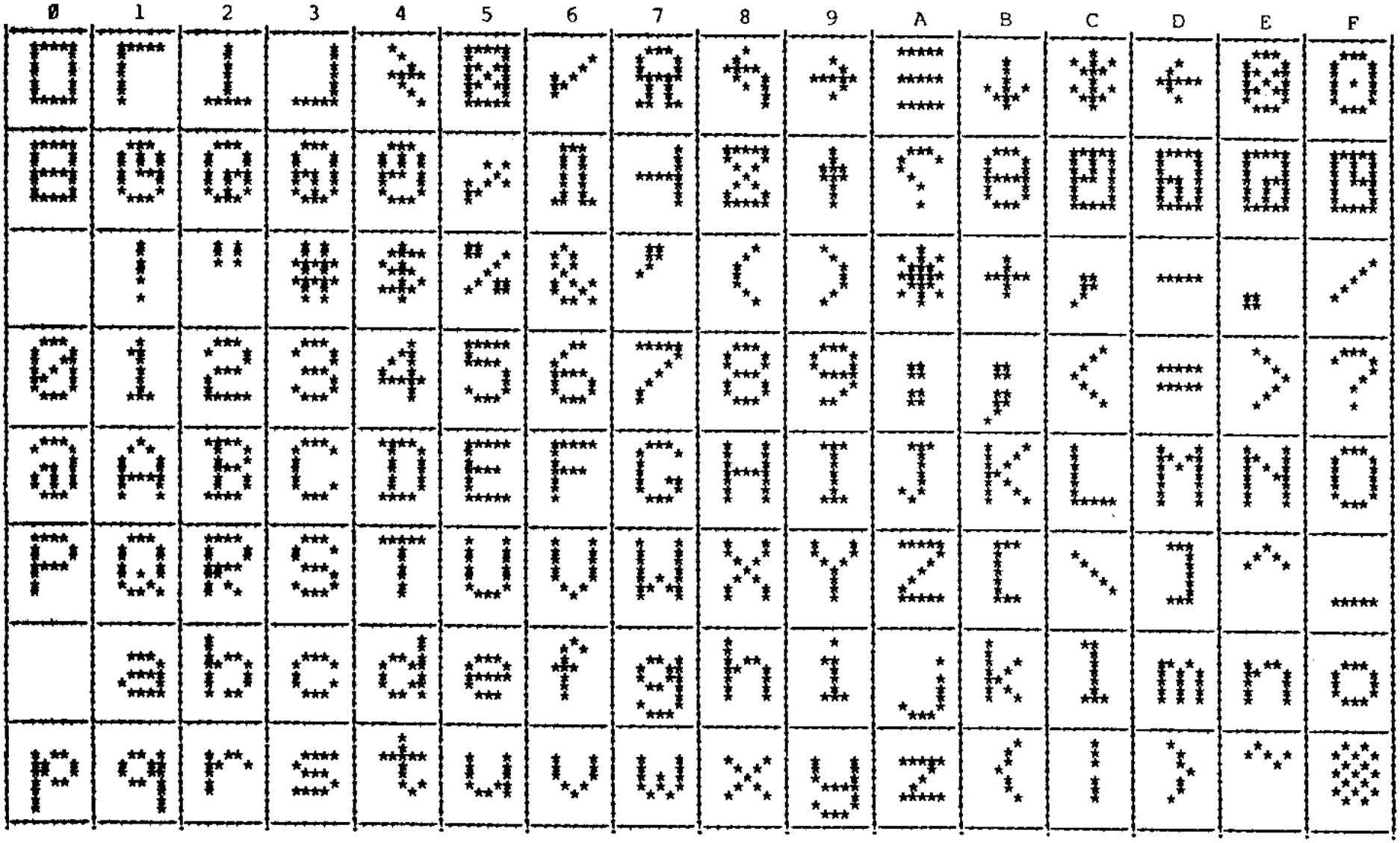
The characters are stored as 10 bytes per character the first byte being the top row. Character code 0 is first in the table.

name	location	value	description
CHRFLG	11FF	0	Character table invert flag
SCSET	1200-17FF		Standard character set
ACSET	1800-1DFF		Alternate character set

Appendix D
Example program

This short example shows how some of the features of the AVC text handling may be put to use. The program was written in MBASIC but should be easy to implement in most BASIC dialects.

```
1 REM ************************************************
2 REM * Demonstration program to set up the function
3 REM * keys and print the key functions on the bottom
4 REM * line of the screen leaving 24 scrolled lines.
6 REM ************************************************
10 REM set width to 255 to avoid unwanted new lines
20 WIDTH 255
30 REM set E$ to ESCAPE character
40 E$=CHR$(27)
50 REM select 80 column mode
60 PRINT E$;"A";
70 REM set up function keys
80 PRINT E$;"\1";CHR$(32+7);"DIR A:";CHR$(13);
90 PRINT E$;"\2";CHR$(32+7);"DIR B:";CHR$(13);
100 PRINT E$;"\3";CHR$(32+7);"DIR C:";CHR$(13);
110 PRINT E$;"\4";CHR$(32+7);"DIR D:";CHR$(13);
120 PRINT E$;"\5";CHR$(32+5);"STAT";CHR$(13);
130 PRINT E$;"\6";CHR$(32+4);"DDT";CHR$(13);
140 PRINT E$;"\7";CHR$(32+4);"PIP";CHR$(13);
150 PRINT E$;"\8";CHR$(32+7);"MBASIC";CHR$(13);
160 PRINT E$;"\9";CHR$(32+6);"FILES";CHR$(13);
170 PRINT E$;"\0";CHR$(32+7);"SYSTEM";CHR$(13);
180 REM alternate character set is inverse video
190 PRINT E$;"P";
200 REM print on bottom line
210 PRINT E$;"=";CHR$(32+24);CHR$(32+0);
220 REM disable wraparound
230 PRINT E$;"K";
240 REM print bottom line
250 PRINT E$;"D";"DIR A:   ";
260 PRINT E$;"C";"DIR B:   ";
270 PRINT E$;"D";"DIR C:   ";
280 PRINT E$;"C";"DIR D:   ";
290 PRINT E$;"D";"STAT     ";
300 PRINT E$;"C";"DDT      ";
310 PRINT E$;"D";"PIP      ";
320 PRINT E$;"C";"MBASIC   ";
330 PRINT E$;"D";"FILES    ";
340 PRINT E$;"C";"SYSTEM   ";
350 REM set screen window to 24 lines
360 PRINT E$;"=";"  ";CHR$(27);"Y";CHR$(32+24);
370 REM clear screen
380 PRINT CHR$(12);
390 REM enable wraparound
400 PRINT E$;"J";
410 REM set normal video
420 PRINT E$;"D";
430 REM set width back to 80 columns
440 WIDTH 80
450 SYSTEM
```



Appendix E

Standard Character Set Font (codes 00 to 7F hex)

The first hex digit is down the side the second is along the top.

Appendix E

Alternate Character Set Font (codes 80 to FF hex)

The first digit is down the side the second along the top.