Propeller 80 Column Card for MTX

Firmware Version 2

W Brendling
11 Feb 2017

I developed a low chip count 80 column VGA display card for the MTX intended to compliment the CFX storage & CP/M system developed by Martin. The original firmware was designed primarily to demonstrate the capabilities of the new card. Dave suggested that, in order to maximise software support, the new card should be as far as possible compatible with the original MTX 80 column card.

Therefore, revised firmware has been developed for the Propeller, which gives the new 80 column card three operating modes:

❿ Compatible with the MTX 80 column card using colour output.

❿ Compatible with the MTX 80 column card using monochrome output.

❿ Enhanced mode, providing the full capabilities of the Propeller card.

Compatible Modes

The Propeller card starts in colour compatible mode. The following character sequences (escape codes) are used to switch between modes:

Character Sequence	Mode Selected
0x1B, 0x9C	Colour compatible mode
0x1B, 0x9D	Monochrome compatible mode
0x1B, 0x9E	Enhanced mode

These escape sequences do nothing on the original MTX, and have been chosen as unlikely to be generated accidentally. Note that the second character has to be exactly as given, none of the other characters with the same 5 lsb work (unlike most escape codes on the MTX).

When in compatible mode, apart from the above, all other control and escape codes function as near as possible to that documented for the MTX 80 column card (see my note "MTX 80 Column Driver").

The following differences are known:

❿ The display buffer in the Propeller is only 2,000 characters, rather than 2,048. This will only be noticeable if a non-default write mask is set, and characters or attributes are scrolled off the top of the screen and back on the bottom.

❿ A different line drawing algorithm is used for ESC,"B". This might give slightly different results for sloped lines.

❿ The Propeller card has no bell for <Ctrl+G>.

Enhanced Mode

Printable characters are identical in all modes. The following sections document the control and escape sequences where they differ from the colour compatible mode. If a sequence is not documented, then its behaviour is unchanged from compatible mode.

The control and escape sequences for enhanced modes have been kept largely consistent with the compatible modes. The main differences are that the colours are 6-bit (64 colours) rather than 3-bit (8 colours), and the attributes are separate from the colours rather than combined.

Colours are specified as bit pattern 00bbggrr, where bb are to bits defining the blue intensity, gg are two bits defining green intensity and rr are two bits defining red intensity. Therefore some example colour codes are:

Binary	Hex	Decimal	Colour
00 00 00 00	0x00	0	Black
00 00 00 01	0x01	1	Dark Red
00 00 00 11	0x03	3	Bright Red
00 00 10 00	0x08	8	Medium Green
00 00 11 11	0x0F	15	Bright Yellow
00 10 00 10	0x22	34	Medium Magenta
00 11 00 11	0x30	48	Bright Blue
00 11 11 11	0x3F	63	Bright White

64 bit foreground and background colours are selected using <Ctrl+D> and <Ctrl+F> respectively. Control codes <Ctrl+P> to <Ctrl+W> select the eight high intensity colours.

Attributes are specified by a byte containing the following bits:

Bit	Effect
0 lsb	Underline
1	No effect
2	No effect
3	No effect
4	No effect
5	Inverse video
6	Blink
7 msb	Graphics mode

These bits have been chosen to be consistent with the MTX 80 column card in monochrome mode. (Note: Internally different bits are used to store the attributes).

Escape sequences “B”, “N”, “P”, “T”, “U” and “V” are used to set the attribute bits.

Control Codes (0x00 – 0x1F)

Control Code	Data Bytes	Action
^D (0x04)	m	Set background colour: 0x00 – 0x3F: Sets both printing and non-printing background colour to 6 lsb of m. 0x40 – 0x7F: Sets printing background only. 0x80 – 0xBF: Sets non-printing background only.
^F (0x06)	m	Set foreground colour: 0x00 – 0x3F: Set both the printing and non-printing foreground colour to 6 lsb of m. 0x40 – 0x7F: Sets printing foreground only. 0x80 – 0xBF: Sets non-printing foreground only.

Escape Sequences

The following additional escape sequences are defined, in addition to the mode selection ones.

Command Character	Data Bytes	Action
“H” (0x48)		Deletes the character under the cursor, shifting the remainder of the line one space to the left and inserting a space character with non-printing colour and attributes on the end of the line.
“K” (0x4B)		Duplicates the line containing the cursor, moving all the remaining lines down one.
“L” (0x4C)	ch, r0 .. r19	Redefines the glyph for alpha character ch. Bytes r0 .. r19 define the pixels for each row, from top to bottom. For each row the lsb is output first (to left) and msb last (to right).
“M” (0x4D)	ch, r0 .. r19	Redefines the glyph for graphics character ch. Bytes r0 .. r19 define the pixels for each row, from top to bottom. For each row the lsb is output first (to left) and msb last (to right).
“O” (x4F)	n	Select virtual screen specified by the 3 lsb of n. By default virtual screen 0 is selected.
“Q” (0x51)	m, c1 .. cm	Outputs m 8-bit characters with no processing of control or escape codes and no font mapping, using printing colours and attributes. Probably most useful with the graphics mode attribute set.
“R” (0x52)	m, b1 ...	Output m characters in raw (internal) format. Requires four bytes to define the contents of each character cell.
“Y” (0x59)	n, w, h, x0, y0	Defines the limits of the window for the virtual screen specified by the 3 lsb of n. w = width (1 to 80), h = height (1 to 24), x0 = first column (0 to 80-w), y0 = first row (0 to 24-h). Note: Defining the currently selected virtual screen (initially screen 0) has no effect.

Command Character	Data Bytes	Action
“Z” (0x5A)		Reboot the propeller chip. Amongst other things, resets all characters glyphs to their default.

Hardware Interface

Port 0x60

Writing bytes to port 0x60 sends characters to display, and control and escape codes to process to the propeller chip.

Reading from port 0x60 returns an indication of the number of characters in the propeller input queue waiting processing. A value of zero indicates that the queue is empty and the display is up to date. A value of 0xFF (255) indicates that the queue is full and any further input will cause an overflow and characters will be lost. If the number of bytes in the queue is N, then the status value returned is (N+7)/8.

Port 0x61

This port provides the ability to read back the contents of the propeller display buffer, and some of the display parameters.

To select data to read back, write bytes to the port as follows:

Value	Interpretation
0x00 – 0x4F	Select character data read, starting from this column in the currently selected row.
0x80 - 0x98	Select character data read, starting from the currently selected column of the row specified by the 5 lsb of the byte.
0xA0-0xB3	Select display parameter read, starting from the location specified by the 5 lsb of the byte.
0xFF	Reboot the propeller chip

Note: Position selection and character data read-back is for the whole screen, irrespective of any virtual screen selected: Each character cell will return four bytes:

❿ Background colour in 6 msb, with 2 lsb clear.
❿ Foreground colour in 6 msb, with 2 lsb clear.
❿ Character code
❿ Attributes in internal bit ordering:
⇲ Bit 0 – Graphics mode (9-th bit of character code)
⇲ Bit 1 – Underscore
⇲ Bit 2 – Inverse video
⇲ Bit 3 – Blink
⇲ Bit 4 – Character contains cursor

To write this data back to the display, first position the cursor using <Ctrl+C>, then use raw write (Esc “R”) to write the data.

Display parameter read-back returns the following values:

Bytes	Interpretation
0-3	Printing character format (background, foreground, null, attributes)
4-7	Non-printing character format
8	Width of currently selected virtual screen
9	Height of currently selected virtual screen
10	Left column of currently selected virtual screen
11	Top row of currently selected virtual screen
12	Scroll method (0=whole screen (pointer update), 1=character copy)
13	Column position of cursor (relative to left edge of virtual screen)
14	Row position of cursor (relative to top of virtual screen)
15	Cursor state (0=hide, 1=show)
16	Page mode (0=scroll, 1=page)
17	Font selection (0=normal, 1=alternate, 2=special graphic)
18	Write mask (0=write both, 1=write character, 2=write format)
19	Number of currently selected virtual screen
20	Compatibility mode (0=colour, 1=mono, 2=enhanced)