16 The Firmware Indirections.

This section gives the detailed entry and exit conditions and effects of the routines in the indirections jumpblock. See section 14.2 for a list of these routines.

The firmware indirections are taken by the firmware at key points. They allow the user to intercept and alter a number of firmware actions without having to provide a complete new firmware package.

The descriptions given are for the default settings of the indirections. Replacement routines need not perform all the actions that the default routine performs although they are advised to do so.

IND : TXT DRAW CURSOR #BDCD

Place the cursor blob on the screen (if enabled).

Default action:

If the cursor is enabled and turned on then the cursor blob is drawn on the screen. If not then no action is taken. The current text position is forced into the window (see TXT VALIDATE) and the cursor blob is written at the resulting position. The cursor blob is an inverse patch. This routine will only be called twice if TXT UNDRAW CURSOR is called in between.

Entry conditions:

No conditions.

Exit conditions:

AF corrupt.
All other registers preserved.

Notes:

This indirection is provided to allow the user to change the form of the cursor blob. See TXT PLACE CURSOR for a description of how the cursor blob is normally written.

The Text VDU routines call this indirection whenever the cursor is placed on the screen. All the Text VDU routines that read from the screen, write to the screen or change the current position remove the cursor (using TXT UNDRAW CURSOR) before performing their action and place it back on the screen afterwards (using TXT DRAW CURSOR). An example of such a routine is TXT WR CHAR that writes a character on the screen.

This indirection is set up when TXT INITIALISE or TXT RESET is called.

Related entries:

TXT PLACE CURSOR
TXT UNDRAW CURSOR

IND : TXT UNDRAW CURSOR #BDD0

Remove the cursor blob from the screen (if enabled).

Default action:

If the cursor is enabled and turned on then the cursor blob is removed from the screen. If not then no action is taken. The cursor blob is an inverse patch. This routine will only be called after TXT DRAW CURSOR has been used to place the cursor on the screen.

Entry conditions:

No conditions.

Exit conditions:

AF corrupt.
All other registers preserved.

Notes:

This indirection is provided to allow the user to change the form of the cursor blob. See TXT REMOVE CURSOR for a description of how the cursor blob is normally removed.

The Text VDU routines call this indirection to remove the cursor from the screen. All the Text VDU routines that read from the screen, write to the screen or change the current position remove the cursor (using TXT UNDRAW CURSOR) before performing their action and place it back on the screen afterwards (using TXT DRAW CURSOR). An example of such a routine is TXT WR CHAR that writes a character on the screen.

This indirection is set up when TXT INITIALISE or TXT RESET is called.

Related entries:

TXT DRAW CURSOR
TXT REMOVE CURSOR

IND : TXT WRITE CHAR #BDD3

Write a character onto the screen.

Default action:

Put a character on the screen at a character position.

Entry conditions:

A contains the character to write.
H contains the physical column to write at.
L contains the physical row to write at.

Exit conditions:

AF, BC, DE and HL corrupt.
All other registers preserved.

Notes:

The character position to write at is given in physical coordinates. i.e. Row 0, column 0 is the top left corner of the screen. The position is not checked for legality.

TXT WRITE CHAR is called by TXT WR CHAR to print a character on the screen. The removing of the cursor blob and the calculation of the new current position are performed by TXT WR CHAR and not by TXT WRITE CHAR.

This indirection is set up when TXT INITIALISE or TXT RESET is called.

Related entries:

TXT OUTPUT
TXT UNWRITE
TXT WR CHAR

IND : TXT UNWRITE #BDD6

Read a character from the screen.

Default action:

Try to read a character from the screen at a character position.

Entry conditions:

H contains the physical column to read from.
L contains the physical row to read from.

Exit conditions:

If a readable character was found:

Carry true.
A contains the character read.

If no recognisable character was found:

Carry false.
A contains zero.

Always:

BC, DE, HL and other flags corrupt.
All other registers preserved.

Notes:

The character position to read from is given in physical coordinates. i.e. Row 0, column 0 is the top left corner of the screen. The position is not checked for legality.

This indirection is called by TXT RD CHAR to read a character from the screen. TXT RD CHAR removes the cursor from the screen before calling this indirection.

The read is performed by comparing the matrix found on the screen with the matrices used to generate characters. As a result changing a character matrix, changing the pen or paper inks or changing the screen (e.g. drawing a line through a character) may make the character unreadable. In particular the cursor blob will cause confusion and so it should not be on the screen

Special precautions are taken against generating inverse space (character #8F). Initially the character is read assuming that the background to the character was written in the current paper ink. If this fails to generate a recognisable character or it generates inverse space then another try is made by assuming that the character was written in the current pen ink.

The characters are scanned starting with #00 and finishing with #FF. Thus, if there are two possible character matrices that match the screen, the smaller of the two character numbers will be returned.

This indirection is set up when TXT INITIALISE or TXT RESET is called.

Related entries:

TXT RD CHAR
TXT WRITE CHAR

IND : TXT UNWRITE #BDD9

Output a character or control code.

Default action:

Print a character on the screen or obey a control code (characters #00..#1F). Works on the currently selected stream (except as noted below).

Entry conditions:

A contains the character or code.

Exit conditions:

AF, BC, DE and HL corrupt.
All other registers preserved.

Notes:

This indirection is called by TXT OUTPUT to do the work of printing characters or obeying the control codes. It is provided to allow the user to change the method of dealing with characters and control codes or to allow the user to redirect output (to the printer for example). TXT OUTPUT merely preserves the registers around the call of TXT OUT ACTION.

Control codes may take up to 9 parameters. These are the characters sent following the initial control code. The characters sent are stored in a buffer until sufficient have been received to make up all the required parameters. The control code buffer is only long enough to accept 9 parameter characters.

There is only one control code buffer which is shared between all the streams. It is, therefore, possible to get unpredictable results if the output stream is changed part of the way through sending a control code sequence.

If the VDU is disabled then no characters will be printed on the screen. In V1.1 firmware control codes that are specially marked in the control code table will not be obeyed if the VDU is disabled. Other control codes and all control codes in V1.0 firmware will be obeyed.

If the graphics character write mode is enabled then all characters and control codes are printed using the Graphics VDU (see GRA WR CHAR) and are not obeyed. Normally characters are written by the Text VDU (see TXT WR CHAR).

This indirection is set up when TXT INITIALISE or TXT RESET is called.

Related entries:

TXT OUTPUT
TXT WR CHAR

IND : GRA PLOT #BDDC

Plot a point.

Default action:

Check if the point lies inside the current window and if so write it in the current graphics pen ink and using the current graphics write mode. The current graphics position is always moved to the specified point.

Entry conditions:

DE contains the user X coordinate of the point to plot.
HL contains the user Y coordinate of the point to plot.

Exit conditions:

AF, BC, DE and HL corrupt.
All other registers preserved.

Notes:

The position of the point to plot is given in user coordinates, i.e. relative to the user origin.

This indirection is called by GRA PLOT RELATIVE and GRA PLOT ABSOLUTE to plot the point requested. It is provided to allow the user to change the method for plotting (to output to an X-Y plotter for example). GRA PLOT RELATIVE converts from relative to user coordinates and then calls this indirection; GRA PLOT ABSOLUTE calls this indirection immediately.

To write the point on the screen the SCR WRITE indirection is used. Thus the point is plotted using the current graphics write mode.

This indirection is set up when GRA INITIALISE or GRA RESET is called.

Related entries:

GRA PLOT ABSOLUTE
GRA PLOT RELATIVE
GRA TEST
SCR WRITE

IND : GRA TEST #BDDF

Test a point.

Default action:

Cheek if the point is inside the graphics window and return the ink it is currently set to if so. Otherwise, return the current graphic paper ink. The current graphics position is always moved to the specified point.

Entry conditions:

DE contains the user X coordinate of the point to test.
HL contains the user Y coordinate of the point to test.

Exit conditions:

A contains the decoded ink of the specified point.

BC, DE, HL and flags corrupt.
All other registers preserved.

Notes:

The position of the point to test is given in user coordinates, i.e. relative to the user origin.

This indirection is used by GRA TEST RELATIVE and GRA TEST ABSOLUTE to test the point requested. It is provided to allow the user to change the method for testing (comparing with the current pen ink for example). GRA TEST RELATIVE converts from relative to user coordinates and then calls this indirection; GRA TEST ABSOLUTE calls this indirection immediately.

To test the ink of a point inside the window the SCR READ indirection is used.

This indirection is set up when GRA INITIALISE or GRA RESET is called.

Related entries:

GRA PLOT
GRA TEST ABSOLUTE
GRA TEST RELATIVE
SCR READ

IND : GRA LINE #BDE2

Draw a line.

Default action:

Draw a line between the current graphics position and the given endpoint using the current graphics write mode. Points on the line that lie outside the current graphics window will not be plotted. The current graphics position is always moved to the specified endpoint.

Entry conditions:

DE contains the user X coordinate of the endpoint.
HL contains the user Y coordinate of the endpoint.

Exit conditions:

AF, BC, DE and HL corrupt.
All other registers preserved.

Notes:

The position of the endpoint is given in user coordinates, i.e. relative to the user origin.

This indirection is used by GRA LINE RELATIVE and GRA LINE ABSOLUTE to draw the line requested. It is provided to allow the user to change the method for line drawing (to output to an X-Y plotter for example). GRA LINE RELATIVE converts from relative to user coordinates and then calls the indirection; GRA LINE ABSOLUTE calls the indirection immediately.

The line is split up into horizontal or vertical sections that are drawn separately (see SCR HORIZONTAL and SCR VERTICAL). The SCR WRITE indirection is called to write the points in these sections. This means that the line is plotted using the current graphics write mode.

In V1.0 firmware the line is plotted in the current pen ink. But in V1.l firmware the setting of the line mask determines how pixels on the line will be plotted. The line mask is bit significant and is used repeatedly in the order bit 7, bit 6 down to bit 0 for each pixel in the line. If the bit is one then the pixel is plotted in the graphics pen ink. If the bit is zero then the action taken depends on the graphics background write mode. If the background mode is opaque then the pixel is plotted in the graphics paper ink. If the background mode is transparent then the pixel is not plotted.

In V 1.1 firmware the first pixel of the line (that at the current graphics position) is not plotted if the first point plotting mode is set false.

This indirection is set up when GRA INITIALISE or GRA RESET is called.

Related entries:

GRA LINE ABSOLUTE
GRA LINE RELATIVE
GRA SET BACK
GRA SET FIRST
GRA SET LINE MASK
SCR HORIZONTAL
SCR VERTICAL

IND : SCR READ #BDE5

Read a pixel from the screen.

Default action:

Read a pixel from the screen and decode its ink.

Entry conditions:

HL contains the screen address of the pixel.
C contains the mask for the pixel.

Exit conditions:

A contains the decoded ink that the pixel was set to.

Flags corrupt.
All other registers preserved.

Notes:

The mask supplied must be a mask for a single pixel otherwise the decoding of the ink read from the screen will not work correctly.

This indirection is set up when SCR INITIALISE or SCR RESET is called. It is called by GRA TEST.

Related entries:

GRA TEST
SCR WRITE

IND : SCR WRITE #BDE8

Write pixel(s) using the current graphics write mode.

Default action:

Plot a pixel or pixels on the screen using the current graphics write mode.

Entry conditions:

HL contains the screen address of the pixel(s).
C contains the mask for the pixel(s).
B contains the encoded ink to plot with.

Exit conditions:

AF corrupt.
All other registers preserved.

Notes:

The pixel mask supplied can be for a single pixel or more than one pixel (or even no pixels). The ink supplied should be encoded to cover the whole of a byte (see SCR INK ENCODE).

The pixel is plotted using the current Graphics VDU write mode. These modes are:

FORCE	Pixel is set to the new ink irrespective of the old ink.
XOR	Pixel is set to the ink formed by exclusive-orIng the new ink for the pixel and its current setting.
AND	Pixel is set to the ink formed by anding the new ink for the pixel and its current setting.
OR	pixel is set to the ink formed by oring the new ink for the pixel and its current setting.

The write mode can be set by calling SCR ACCESS appropriately.

This indirection is called by all Graphics VDU write routines, in particular GRA PLOT RELATIVE, GRA PLOT ABSOLUTE, GRA LINE RELATIVE, GRA LINE ABSOLUTE and GRA WR CHAR, to plot pixels on the screen. It is provided to allow the user to intercept the lowest level of point plotting (perhaps to add yet another plotting mode).

This indirection is set up when SCR INITIALISE or SCR RESET is called.

Related entries:

GRA PLOT
SCR ACCESS
SCR PIXELS
SCR READ

IND : SCR MODE CLEAR #BDEB

Clear the screen to ink 0.

Default action:

Clear the screen memory to zeros. This indirection is provided to allow the user to prevent the screen being cleared after the mode is changed.

Entry conditions:

No conditions.

Exit conditions:

AF, BC, DE and HL corrupt.
All other registers preserved.

Notes:

Normally this indirection performs the actions described in SCR CLEAR. In V1.0 firmware it is necessary for the user to set up the inks if this indirection is intercepted (see Appendix XIII). In V1.1 firmware the screen pack sets up the inks for the user after SCR MODE CLEAR has been called. This indirection is set up when SCR INITIALISE or SCR RESET is called. N.B. When this indirection is called the text and graphics VDUs are in non-standard states.

Related entries:

SCR CLEAR
SCR SET MODE

IND : KM TEST BREAK #BDEE

Test for break (or reset).

Default action:

Test if the escape key is pressed, if not then no action is taken. If escape, shift and control are all pressed and no other keys then the system is reset. Otherwise, a break event is reported (see KM BREAK EVENT).

Entry conditions:

Interrupts disabled.
C contains shift and control key states.

Exit conditions:

AF and HL corrupt.
All other registers preserved.

Notes:

This indirection is called by the firmware from the interrupt path. Thus interrupts are disabled and they must remain disabled.

If bit 7 of C is set then the control key is pressed. If bit 5 of C is set then one of the shift keys is pressed.

This indirection is called after the keys have been scanned and the escape key was found to have been pressed. It is provided to allow the user to alter the action of a break (particularly to prevent the system reset, see RESET ENTRY).

This indirection is set up when KM INITIALISE or KM RESET is called.

Related entries:

KM BREAK EVENT

IND : MC WAIT PRINTER #BDF1

Print a character or time out.

Default action:

Wait for the Centralises port to become not busy and then send a character to it. If the port remains busy for a long time the routine times out and the character is not sent.

Entry conditions:

A contains the character to send.

Exit conditions:

If the character was sent OK:

Carry true.

If the Centronics port timed out:

Carry false.

Always:

A and BC corrupt.
All other registers preserved.

Notes:

This indirection is provided to allow the user to drive the printer in a different way. For example, 'escape sequences' could be handled or the time out could be changed.

This indirection is called by the routine MC PRINT CHAR. It tests whether the printer is busy in the same way as MC BUSY PRINTER and sends the character in the same way as MC SEND PRINTER.

This indirection is set up when MC RESET PRINTER is called.

Related entries:

MC BUSY PRINTER
MC PRINT CHAR
MC SEND PRINTER

IND : KM SCAN KEYS #BDF4

Scan the keyboard.

Default action:

Scans the keyboard and updates the key state map. Newly pressed keys are detected and appropriate markers are inserted into the key buffer.

Entry conditions:

No conditions except that interrupts must be disabled.

Exit conditions:

AF, BC, DE and HL corrupt.
All other registers preserved and interrupts remain disabled.

Notes:

This indirection is called every fiftieth of a second during a ticker interrupt. The repeat speeds and start-up delays of the keys are measured in scans of the keyboard and hence fiftieths of a second.

If the escape key is pressed then the indirection KM TEST BREAK is called to process the break.

Related entries:

KL SCAN NEEDED
KM READ KEY
KM TEST BREAK
KM TEST KEY