

Extended EhBasic 2.25 command set for the Junior Computer][

Miscellaneous Commands

BEEP

Chime a system beep.

DELAY <ms>

Pauses for a delay time of *ms* milliseconds, where *ms* lies in the range from 1 to 32767.

PLIST

Send program listing to printer port.

PR# <dev>

Sets current character output to device *dev*.

IN# <dev>

Sets current character input to device *dev*.

Text Screen Commands

INVERSE

Switch to inverse text - Terminal only.

FLASH

Switch to blinking text - Terminal only.

NORMAL

Switch back to normal text - Terminal only.

HOME

Moves the cursor to the upper left screen position (1,1) without clearing the screen.

CLS

Clears the screen and moves the cursor to the upper left position (1,1).
Works in Terminal, Text and Graphics mode.

LOCATE <x,y>

Sets the cursor to horizontal position x and vertical position y. Value x must be in the range from 1 to <Max Screen Width>, where y must be in the range from 1 to <Max Screen Height>.

Input/Output Commands for the GPIO pins and the I2C Interface

PORTIO <dev,iomask>

Sets the VIA port input and output lines. A one in *iomask* makes the corresponding line an output, while a null sets the line as an input. A *dev* value of 0 set the IO mask for VIA port A, while a *dev* value of 1 corresponds to port B.

Examples:

```
PORTIO 0,%11000000  Port A pins 6 and 7 are outputs, pins 0 to 5 are inputs.  
PORTIO 1,$FF        Set all port B pins as an output.
```

PORT <adr,data>

Sends the 8-bit word *data* to port *adr*. Port address 0 corresponds to VIA port A, while port address 1 corresponds to port B. Every given *adr* value in the range from 2 to 255 addresses a hardware device connected to the I2C port. For example, the device addresses \$20 to \$27 are reserved for PCF8574P I2C 8-bit I/O Port expander devices.

Note: Every line on the VIA ports A or B you wish to write to, must be formerly set as an output pin (see PORTIO).

Note: Normally I2C address 0 is used for broadcasting to all devices. This is not possible with the PORT command, because the device 0 address is used to identify VIA port A.

Note: Sending data to a I2C port needs to be terminated by a I2CStop command.

Examples:

```
PORT 1,255           Sends the 8-bit word 255 to VIA port B. Every port 1  
                     line formerly set as an output will go to a high value  
                     (see PORTIO).  
  
PORT $20,%10101010  Sends the given bit value to the I2C device with the  
                     address $20 (a PCF8574P).
```

PORT(*adr*)

Receives a 8-bit word from the port *adr*. A *adr* value of 0 corresponds to VIA port A and 1 to port B, while a *adr* value in the range from 2 to 255 addresses a hardware device connected to the I2C port (see PORT *adr,data*).

Note: Every line on the VIA ports A or B you wish to read from, must be formerly set as an input pin (see PORTIO).

Note: Every formerly set output line is reading back its last written output value. To suppress every output value to a 0 value you should mask out all outputs by ANDing it with a 0 bit. For example use the command `PORT(1) AND %11100000` to mask out the port B lines 0 to 4.

Note: Receiving data from a I2C port needs to be terminated by a I2CStop command.

Examples:

<code>PRINT PORT(0)</code>	Prints the value of all input lines on VIA port A. Every output line is reading back its last written value.
----------------------------	--

<code>A = PORT(\$20)</code>	Save the input value of I2C device \$20 (a PCF8574) to variable A.
-----------------------------	--

I2CStop

Send a Stop signal to the I2C interface.

If you start sending to or receiving data from the I2C interface using the PORT command you have to close the communication after the last data byte.

This allows sending or receiving multiple bytes of data in a row.

Examples:

```
FOR i = 0 TO 9 : PORT10,i : NEXT i : I2CStop
```

Send the values 0 to 9 to I2C device at address 10.

```
A = PORT($20) : B = PORT($20) : I2CStop
```

Save the input value of I2C device \$20 (a PCF8574) to variable A and B.

Graphics Screen Commands

SCREEN <mode>

Switches to one of the following Text and Graphics modes

- | | |
|---|---|
| 0 | 40 columns black and white text mode |
| 1 | 80 columns black and white text mode |
| 2 | 40 columns color text mode |
| 3 | 80 columns color text mode |
| 4 | 256 x 212 pixel 16 color graphics mode |
| 7 | 256 x 212 pixel 256 color graphics mode |

COLOR <colour>

Set the current drawing color for PIXEL, LINE, RECT and BAR.

Example:

`Color 3` Set the drawing color to purple.

The following 16 colors number are defined for the 16 color modes.

- | | |
|----|------------|
| 0 | Black |
| 1 | Dark Red |
| 2 | Dark Blue |
| 3 | Purple |
| 4 | Dark Green |
| 5 | Dark Gray |
| 6 | Blue |
| 7 | Light Blue |
| 8 | Brown |
| 9 | Orange |
| 10 | Light Gray |
| 11 | Pink |
| 12 | Green |
| 13 | Yellow |
| 14 | Aquamarine |
| 15 | White |

For the 256 color mode values from 0 to 255 are allowed and the color is composed with the following byte bit coding *RRBBBGGG* (red, green, blue).

PIXEL <x,y>

Draw a pixel with the current color at screen position x,y.

LINE <x1,y1,x2,y2>

Draws a line with the current color. The start point of the line is given by x1,y1 and the end point by x2,y2.

RECT <x1,y1,x2,y2>

Draws a framed and unfilled rectangle with the current color. The top/left corner of the rectangle is given by x1,y1 while the bottom/right corner is x2,y2.

BAR <x1,y1,x2,y2>

Draws a filled rectangle with the current color. The top/left corner of the rectangle is given by x1,y1 while the bottom/right corner is x2,y2.

Note: This command is currently not working.

Saving and Loading Basic Files

SAVE {dev {,name}}

Saves a file with filename *name* to storage device *dev*. The device ID must be in the range from 0 to 255. Currently defined device IDs:

0 - XModem data transfer protokoll.

1 - Datasette.

If no device ID is specified, then device 0 is addressed. Because the XModem protokoll doesn't need a filename, no *name* parameter is allowed for device 0. A filename must be surrounded by "". If an empty or "*" filename is given, then the file is saved as untitled.

Examples:

SAVE	Sends a file via XModem
SAVE 0	Same as above
SAVE 1, ""	Save an untitled file to the tape device
SAVE 1, "Test File"	Save the file "Test File" to the tape device

Load {dev {,name}}

Loades a file with filename *name* from storage device *dev*. The device ID must be in the range from 0 to 255. Currently defined device IDs:

0 - XModem data transfer protokoll.

1 - Datasette.

If no device ID is specified, then device 0 is addressed. Because the XModem protokoll doesn't need a filename, no *name* parameter is allowed for device 0. A filename must be surrounded by "". If a blank or "*" filename is specified, the next file found on tape will be loaded. If a filename ends up to *, then the next file beginning with the prepended filename is loaded from tape.

Examples:

LOAD	Loads a file via XModem
LOAD 0	Same as above
LOAD 1, ""	Loads the next file that is found on the tape device
LOAD 1, "Test File"	Loads the file „Test File“ from the tape device
LOAD 1, "Tes*"	Loads the next file, whose filename starts with „Tes“