

Library CPMPlus for ccz80

System

startup

<i>Arguments</i>	startup()
<i>Description</i>	Prepare the system to use all the features of this library
<i>Result</i>	Not significant.
<i>Example</i>	startup();
<i>Notes</i>	It is not necessary to explicitly use this function, because it runs automatically to include the library file.

System

beep

<i>Arguments</i>	beep()
<i>Description</i>	Beeeps from the speaker.
<i>Result</i>	Not significant.
<i>Example</i>	beep();
<i>Notes</i>	

System

computertype

<i>Arguments</i>	computertype()
<i>Description</i>	Gets the type of computer that is running the program.
<i>Result</i>	0 = Amstrad CPC 6128 1 = Amstrad PCW 8000/9000/10 series 3 = Spectrum +3 65 = Amstrad PCW16
<i>Example</i>	if (computertype() != #31) return;
<i>Notes</i>	

System

cpmversion

<i>Arguments</i>	cpmversion()
<i>Description</i>	Gets the version number of CP/M on which is running the program.
<i>Result</i>	Returns version number of CP/M, as follow:

#00 = Version 1
 #20 = Version 2.0
 #21 = Version 2.1
 #22 = Version 2.2
 #25 = DOS +
 #28 = Personal CP/M-80
 #30 = MP/M II, MP/M-86
 #31 = CP/M Plus
 #33 = Apricot PCP/M-86
 #41 = DOSPLUS 1
 #50 = DOSPLUS 2

Example `if (cpmversion != #31) return;`

Notes

Keyboard

speedkey

Arguments `speedkey(<initial delay>, <subsequent delay>)`

Description Sets the speed of key repeat to the <initial delay> and <subsequent delay>.

Result Not significant.

Example `speedkey(10, 1);`

Notes

Keyboard

keydef

Arguments `keydef(<key number>, <shift states>, <code>)`

Description Assigns the character <code> to <key number> with the status specified by <shift states>. Value of <shift states> is:

0 = unshifted key
 1 = Shift pressed for Amstrad PCW/PCW16, Caps Shift for Spectrum +3
 2 = Alt pressed for Amstrad PCW, Symbol Shift for Spectrum +3, Ctrl for Amstrad PCW16
 3 = Shift+Alt pressed for Amstrad PCW, Extend for Spectrum +3, Shift+Control for Amstrad PCW16
 4 = Extra pressed for Amstrad PCW

Result Not significant.

Example `keydef(46, 1, #80);`

Notes

Keyboard

key

Arguments `key(<key>, <string>)`

Description Assigns the <string> to specified <key>.

Result Not significant.

Example `key(#80, "hello");`

Notes

Keyboard

clearinput

Arguments `clearinput()`

Description Removes all key pressed from keyboard buffer.

Result Not significant.

Example `clearinput();`

Notes

Keyboard

inkey

Arguments `inkey()`

Description Gets the ASCII value of the next key in keyboard buffer.

Result Returns the ASCII value of key pressed. If nothing is pressed returns 0.

Example `if (inkey() = 'a') value = 1;`

Notes

Keyboard

input

Arguments `input(<string>)`

Description Gets a string from keyboard and store it in <string>. During the introduction of string, backspace key can be used to delete last character and return key to complete the entry.

Result Not significant.

Example `string name[50];
input(name);`

Notes

Screen

resetScreen

Arguments `resetScreen()`

Description Restores all screen settings to default values.

Result Not significant.

Example `resetScreen();`

Notes

Screen

charset

<i>Arguments</i>	charset(<country>)
<i>Description</i>	Sets the international character set for the <country> specified, as follows: 0 = USA 1 = France 2 = Germany 3 = United Kingdom 4 = Denmark 5 = Sweden 6 = Italy 7 = Spain
<i>Result</i>	Not significant.
<i>Example</i>	charset(6);
<i>Notes</i>	

Screen

mode

<i>Arguments</i>	mode(<mode>)
<i>Description</i>	Set the screen mode as follows:
<i>Result</i>	Not significant.
<i>Example</i>	mode(2);
<i>Notes</i>	Only valid for Amstrad CPC 6128.

Screen

mode80x24

<i>Arguments</i>	mode80x24(<on/off>)
<i>Description</i>	Sets the display to 80 x 25 characters for programs that require it to run. If <on/off> is 0 mode 80 x 25 is off, if other than 0 is active.
<i>Result</i>	Not significant.
<i>Example</i>	mode80x24(1);
<i>Notes</i>	

Screen

window

<i>Arguments</i>	window(<x>, <y>, <width>, <height>)
<i>Description</i>	Sets the window screen work in the physical row and column <x> and <y>, with the <width> and <height> specified. The first column and row is number 1.

Result Not significant.

Example `window(10, 5, 60, 20);`

Notes

Screen

windowx

Arguments `windowx()`

Description Gets the physical column where starts the window screen work.

Result Returns the physical column where it starts the defined window. The first column is number 1.

Example `byte a;`
 `a = windowx();`

Notes

Screen

windowy

Arguments `windowy()`

Description Gets the physical row where starts the window screen work.

Result Returns the physical row where it starts the defined window. The first row is number 1.

Example `byte a;`
 `a = windowy();`

Notes

Screen

windowwidth

Arguments `windowwidth()`

Description Gets the width in characters of the window screen work.

Result Returns the defined window width in characters.

Example `byte a;`
 `a = windowwidth();`

Notes

Screen

windowheight

Arguments `windowheight()`

Description Gets the height in characters of the window screen work.

Result Returns the defined window height in characters.

Example `byte a;
a = windowheight();`

Notes

Screen

cursor

Arguments `cursor(<on/off>)`

Description Sets the cursor display. If <on/off> is 0 cursor is hide, if other than 0 is visible.

Result Not significant.

Example `cursor(0);`

Notes

Screen

status

Arguments `status(<on/off>)`

Description Sets the status line mode. If <on/off> is 0 status line is off and error messages are show in main window, if other than 0 status line is active and last line of screen is reserved to it.

Result Not significant.

Example `status(1);`

Notes

Screen

border

Arguments `border(<colour>)`

Description Sets the border colour of screen. The <colour> value is the combination of the intensity of green (bits 5 and 4), red (bits 3 and 2) and blue (bits 1 and 0), where the combination 00 indicates the color off, 01 or 10 indicates medium intensity and 11 indicates high intensity.

Result Not significant.

Example `border(63);`

Notes Only valid for Amstrad CPC. On the Amstrad PCW bit 7 of <colour> is set for reversed video and bit 6 is set for screen blank.

Screen

ink

Arguments `ink(<ink>, <colour 1>, <colour 2>)`

Description Assign to <ink> the <colour 1> and <colour 2>. If colours are differents flashing effect is produced. Value 0 for <ink> specifies the background and border, value 1 for <ink> specifies the foreground color. The <colour 1> and <colour 2> values are the combination of the intensity of green (bits 5 and 4), red (bits 3 and 2) and blue (bits 1 and 0), where the

combination 00 indicates the color off, 01 or 10 indicates medium intensity and 11 indicates high intensity.

Result Not significant.

Example `ink(0, 12, 15);`

Notes This function has not effect becuae exists a bug in it. This bug can be fixed replacing some bytes in CP/M code as described in <http://www.seasip.demon.co.uk/Cpm/bdos.html>.

Screen

paper

Arguments `paper(<colour>)`

Description Sets the background colour of screen. The <colour> value is the combination of the intensity of green (bits 5 and 4), red (bits 3 and 2) and blue (bits 1 and 0), where the combination 00 indicates the color off, 01 or 10 indicates medium intensity and 11 indicates high intensity.

Result Not significant.

Example `paper(63);`

Notes For Amstrad PCW only are valids values for <colour> 0 and 63 to specify dark and light color repectively.

Screen

pen

Arguments `pen(<colour>)`

Description Sets the foreground colour of screen. The <colour> value is the combination of the intensity of green (bits 5 and 4), red (bits 3 and 2) and blue (bits 1 and 0), where the combination 00 indicates the color off, 01 or 10 indicates medium intensity and 11 indicates high intensity.

Result Not significant.

Example `pen(0);`

Notes For Amstrad PCW only are valids values for <colour> 0 and 63 to specify dark and light color repectively.

Screen

inverse

Arguments `inverse(<on/off>)`

Description Sets the inverse print mode. If <on/off> is 0 colour background and colour foreground are used for background and text colour, if other than 0 the colours are exchanged.

Result Not significant.

Example `inverse(1);`

Notes

Screen

underline

Arguments `underline(<on/off>)`

Description Sets the underline print mode. If <on/off> is 0 underline mode is off, if other than 0 underline mode is on.

Result Not significant.

Example `underline(1);`

Notes

Screen

cls

Arguments `cls()`

Description Erases screen and move the cursor to position 0, 0.

Result Not significant.

Example `cls();`

Notes

Screen

locate

Arguments `locate(<x>, <y>)`

Description Moves the cursor to column <x> and row <y>. First column and row are number 1.

Result Not significant.

Example `locate(10, 5);`

Notes

Screen

posx

Arguments `posx()`

Description Gets the column number where the cursor.

Result Returns the column number where the cursor. The first column is number 1.

Example `if (posx() > 50) newline();`

Notes

Screen

posy

Arguments `posy()`

Description Gets the row number where the cursor.

Result Returns the row number where the cursor. The first row is number 1.

Example `if (posy() > 20) cls();`

Notes

Screen

putc

Arguments `putc(<character>)`

Description Display the <character> at cursor position. Control characters are interpreted.

Result Not significant.

Example `print('A');`

Notes

Screen

prints

Arguments `prints(<string>)`

Description Display the <string> at cursor position. Control characters are interpreted.

Result Not significant.

Example `prints("hello");`

Notes

Screen

printf

Arguments `printf(<value>)`

Description Display the decimal representation of byte <value> at cursor position.

Result Not significant.

Example `byte a = #100;
printf(a + 1);`

Notes

Screen

printw

Arguments `printw(<value>)`

Description Display the decimal representation of word <value> at cursor position.

Result Not significant.

Example word a = #1000;
 printw(a);

Notes

Screen

newline

Arguments newline()

Description Moves the cursor to start of next line.

Result Not significant.

Example newline();

Notes

Disc

getuser

Arguments getuser()

Description Get the current user number for disc operations.

Result Returns the current user number, between 0 and 15.

Example if (getuser() != 0) return;

Notes

Disc

setuser

Arguments setuser(<user number>)

Description Set the current user number for disc operations, between 0 and 15.

Result Not significant.

Example setuser(5);

Notes

Disc

resetdiscs

Arguments resetdiscs()

Description Restores all settings of discs to default values and ends any operation started.

Result Not significant.

Example resetdiscs();

getdisc

Arguments getdisc()
Description Gets the number of current disc drive.
Result Returns the current disc number. Disc A is number 0.
Example byte a;
 a = getdisc();
Notes

setdisc

Arguments setdisc(<drive number>
Description Sets the number of current disc drive, where 0 is for disc A, 1 is for disc B, ...
Result Not significant.
Example setdisc(1);
Notes

new

Arguments new(<filename>, <handle>)
Description Creates the file <filename> with the <handle> specified. The <filename> can have the disc letter, if not, file is created in current disc.
Result Returns 0 if the operation is successful, another value if error occurred.
Example array byte f[36];
 new("b:file.dat", f);
Notes

open

Arguments open(<filename>, <handle>)
Description Opens the existing file <filename> with the <handle> specified. The <filename> can have the disc letter, if not, file is open in current disc.
Result Returns 0 if the operation is successful, another value if error occurred.
Example array byte f[36];
 open("b:file.dat", f);
Notes

close

Arguments `close(<handle>)`

Description Closes the previously created or opened file specified by <handle>.

Result Returns 0 if the operation is successful, another value if error occurred.

Example `array byte f[34];
new("file.dat", f);
close(f);`

Notes

readnext

Arguments `readnext(<handle>, <data>)`

Description Reads the next record from previously created or opened file specified by <handle> and store it in <data>. Records are always 128 bytes.

Result Returns 0 if the operation is successful, another value if error occurred.

Example `array byte f[36], data[128];
open("file.dat", f);
readnext(f, data);`

Notes

writenext

Arguments `writenext(<handle>, <data>)`

Description Writes the next record in previously created or opened file specified by <handle>. Record data written is in <data>. Records are always 128 bytes.

Result Returns 0 if the operation is successful, another value if error occurred.

Example `array byte f[36], data[128];
open("file.dat", f);
writenext(f, data);`

Notes

read

Arguments `read(<handle>, <record number>, <data>)`

Description Reads the record <record number> from previously created or opened file specified by <handle> and store it in <data>. Records are always 128 bytes. First record is number 0.

Result Returns 0 if the operation is successful, another value if error occurred.

Example `array byte f[36], data[128];
open("file.dat", f);`

```
read(f, 10, data);
```

Notes

Disc

write

Arguments write(<handle>, <record number>, <data>)

Description Writes the record <record number> in previously created or opened file specified by <handle>. Record data writed is in <data>. Records are alway 128 bytes. First record is number 0.

Result Returns 0 if the operation is successful, another value if error occurred.

Example

```
array byte f[36], data[128];
open("file.dat", f);
write(f, 10, data);
```

Notes

Disc

delete

Arguments delete(<filename>

Description Deletes from disc the <filename>.

Result Returns 0 if the operation is successful, another value if error occurred.

Example

```
delete("file.bak");
```

Notes

Disc

rename

Arguments rename(<filename>, <new name>)

Description Rename in disc the <filename> with <new name>.

Result Returns 0 if the operation is successful, another value if error occurred.

Example

```
rename("file.dat", "name.old");
```

Notes

Printer

lprintc

Arguments lprintc(<character>)

Description Writes to printer the <character>. Control characters are interpreted by printer.

Result Not significant.

Example

```
lprintc(15);
```

lprints

Arguments lprints(<string>)

Description Writes to printer the <string>. Control characters are interpreted by printer.

Result Not significant.

Example lprints("\x0FCompressed\x12Not compressed");

Notes

lprintb

Arguments lprintb(<value>)

Description Writes to printer the decimal representation of byte <value>.

Result Not significant.

Example byte a = 15;
lprintb(a * 2);

Notes

lprintw

Arguments lprintw(<value>)

Description Writes to printer the decimal representation of word <value>.

Result Not significant.

Example word a = 15;
lprintb(a * 5);

Notes

lnewline

Arguments lnewline()

Description Moves the printhead the beginning of the next line.

Result Not significant.

Example lnewline();

Notes

lnewpage

<i>Arguments</i>	lnewpage()
<i>Description</i>	Ejects the current page in printer.
<i>Result</i>	Not significant.
<i>Example</i>	lnewpage();
<i>Notes</i>	