Libraries CPC464 and CPC6128 for ccz80

after

Arguments after(<time>, <label>, <buffer>)

Description Equivalent to AFTER <time> GOSUB <label> command in BASIC, where <buffer> is a 13 bytes space for identifier the timer.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte buffer[13];
byte a, switch = 0;

after(250, routine, buffer);
prints("Guess a letter in 5 seconds\n\r");
do
{
  if (switch) return; // End program
  a = inkey();
}
while (a != rand() % 26 + 97);

printc(a);
prints(" is correct. You win.\n\r");
sound(1, 478, 20, 12, 0, 0, 0);
sound(1, 358, 20, 12, 0, 0, 0);
return; // End program

routine:
prints("Too later. I win.\n\r");
sound(1, 2000, 20, 12, 0 ,0 ,0);
switch = 1;
return;
```

Notes -

every

Arguments every(<time>, <label>, <buffer>)

Description Equivalent to EVERY <time> GOSUB <label> command in BASIC, where <buffer> is a 13 bytes space for identifier the timer.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte buffer[13];

every(50, routine, buffer);
while (1); // Infinite loop

routine:
sound(1, 20, 20, 12, 0, 0, 0);
return;
```

Notes -

remain

Arguments remain(<buffer>)

Description Equivalent to REMAIN function in BASIC where <buffer> is the timer identifier specified in AFTER or EVERY.

Result Same as REMAIN function in BASIC.

Example

```
include "cpc6128.ccz80";

array byte after_1[13], after_2[13];

after(500, routine1, after_1);
after(100, routine2, after_2);
while (1) prints("Program running\r\n");

routine1:
// This routine will not invoked because remain funcion impede it
return;

routine2:
prints("\r\r\nTimer 1 will stoped by remain funcion.\r\n");
prints("Left time units: ");
printw(remain(after_1));
prints("\r\n");
return;
```

Notes -

Timer

ei

Arguments ei()

Description Equivalent to EI command in BASIC.

Result Return value not significant.

Example -

Notes Only works with synchronous events.

Timer

di

Arguments di()

Description Equivalent to DI command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte buffer[13];
word x1, x2, x, y;
byte c;

cls();
tag();
every(20, routine, buffer);
while (1)
{
  x1 = rand() % 320;
  x2 = rand() % 320;
  y = 200 + rand() % 200;
  c = rand() % 256;
```

```
  for (x = 320 - x1; x <= 320 + x2; x += 4)
  {
    di();
    graphicspen(1);
    move(320, 0);
    move(x - 2, y);
    move(x, y);
    printc(' ');
    printc(c);
    frame();
    ei();
  }
}

routine:
graphicsmode(1);
move(320, 0);
graphicspen(3);
draw(x + 8, y - 16); // Paint line
move(320, 0);
graphicspen(3);
draw(x + 8, y - 16); // Erase line
graphicsmode(0);
return;
```

Notes Only works with synchronous events.

time

Arguments time(<long>)

Description Equivalent to TIME function in BASIC leaving the result in <long>.

Result The <long> address.

Example

```
include "cpc6128.ccz80";

byte hour, minute, second;
array byte text[256];
array byte reference[5], time_value[5], n[5];
array byte t[4];

cls(); // Clock
prints("Hour: ");
input(text);
hour = atob(text);
prints("\r\nMinute: ");
input(text);
minute = atob(text);
prints("\r\nSecond: ");
input(text);
second = atob(text);

start1:
cls();
div(ltof(reference, time(t)), wtof(n, 300));

start2:
btof(time_value, 0);
while (hour < 13)
{
  while (minute < 60)
  {
    while (cmp(time_value, btof(n, 60)) == -1)
    {
      add(sub(int(div(ltof(time_value, time(t)), wtof(n, 300))), reference),
btof(n, second));
      locate(1, 1);
      prints(btoa(text, hour) + 1);
      printc(' ');
      prints(btoa(text, minute) + 1);
      printc(' ');
      prints(btoa(text, ftob(time_value)) + 1);
    }
    btof(time_value, 0);
    second = 0;
```

```
    ++minute;
    goto start1;
  }
  minute = 0;
  ++hour;
}
hour = 1;
goto start2;
```

Notes -

Keyboard

testkey

Arguments testkey(<n>)

Description Equivalent to INKEY(<n>) function in BASIC.

Result Same as INKEY function in BASIC.

Example

```
include "cpc6128.ccz80";

while (testkey(55) != 32);
prints("You press [SHIFT] and V");
clearinput();
return;
```

Notes -

Keyboard

joy

Arguments joy(<n>)

Description Equivalent to JOY(<n>) function in BASIC.

Result Same as JOY function in BASIC.

Example

```
include "cpc6128.ccz80";

loop:
prints("For stop the programa move the joystick\r\n");
if (joy(0)) return;
goto loop;
```

Notes -

Keyboard

inkey

Arguments inkey()

Description Equivalent to INKEY$() function in BASIC.

Result The ASCII code of the key readed or 0 if nothing readed.

Example

```
include "cpc6128.ccz80";

byte a;

cls();
prints("Choose Yes or No (Y/N)\r\n");
start:
while (!(a=inkey()));
if (a == 'y' || a == 'Y') goto label_yes;
```

```
if (a == 'n' || a == 'N') goto label_no;
goto start;

label_yes:
prints("You answer YES");
return;

label_no:
prints("You answer NO");
return;
```

Notes -

Keyboard

clearinput

Arguments clearinput()

Description Equivalent to CLEAR INPUT command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte t;

cls();
prints("Type now!");
for (t = 1; t <= 30; ++t) frame(); // Delay
clearinput();
return;
```

Notes -

Keyboard

keydef

Arguments keydef(<n>, <repeat>, <normal>, <shift>, <control>)

Description Equivalent to KEY DEF <n>, <repeat>, <normal>, <shift>, <control> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

key(159, "this is the TAB key");
keydef(68, 1, 159, 9, 9);
prints("Press TAB key");
return;
```

Notes -

Keyboard

key

Arguments key(<n>, <string>)

Description Equivalent to KEY <n>, <string> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

key(11, "border 13:paper 0:pen 1:ink 0,13:ink 1,0:mode 2:list\x0D");
prints("Press ENTER key");
```

```
return;
```

Notes -

Keyboard

speedkey

Arguments speedkey(<delay>, <repeat>)

Description Equivalent to SPEED KEY <delay>, <repeat> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte k;
array byte a[256];

cls();
// Don't use k >= 1
for (k = 7; k != -1; k -= 2)
{
  prints("Type your name and press [RETURN]\r\n");
  speedkey(k, k);
  input(a);
  prints("\n\r");
}
prints("It's a name very interesting");
return;
```

Notes -

Text

cls

Arguments cls()

Description Equivalent to CLS #s command in BASIC, where #s is the current stream specified with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

stream(2);
paper(3);
cls();
return;
```

Notes -

Text

paper

Arguments paper(<n>)

Description Equivalent to PAPER #s, <n> command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte p, t;
```

```
mode(0);
pen(0);
ink(0, 13, 13);
for (p = 1; p <= 15; ++p)
{
  paper(p);
  cls();
  locate(7, 12);
  prints("paper ");
  printb(p);
  for (t = 1; t <= 20; ++t) frame(); // Delay
}
return;
```

Notes -

Text

pen

Arguments pen(<n>)

Description Equivalent to PEN #s, <n> command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte p, t;
array byte spaces[48];

mode(0);
pen(0);
ink(0, 13, 13);
while (1)
  for (p = 1; p <= 15; ++p)
  {
    pen(p);
    prints(strset(spaces, ' ', 47));
    prints("pen ");
    printb(p);
    prints("\r\n");
    for (t = 1; t <= 20; ++t) frame(); // Delay
  }
```

Notes -

Text

printc

Arguments printc(<character>)

Description Equivalent to PRINT #s, CHR$(<character>); command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example -

Notes -

Text

prints

Arguments prints(<string>)

Description Equivalent to PRINT #s, <string>; command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte a[256], b[256];

strcpy(a, "short");
strcpy(b, "this string is long");
prints(a);
prints(a);
prints("\r\n");
prints(a);
printc(' ');
prints(a);
prints("\r\n");
prints(b);
prints(b);
prints("\r\n");
prints(b);
printc(' ');
prints(b);
prints("\r\n");
return;
```

Notes -

Text

printb

Arguments printb(<n>)

Description Equivalent to PRINT #s, <n>; command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example -

Notes -

Text

printw

Arguments printw(<n>)

Description Equivalent to PRINT #s, <n>; command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example -

Notes -

Text

input

Arguments input(<string>)

Description Equivalent to LINE INPUT #s, <string> command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte text[16], a[5], b[5], c[5];

mode(2);
prints("Write the first value: ");
input(text);
atof(a, text);
prints("\r\nWrite the second value: ");
input(text);
atof(b, text);
fcpy(c, a);
mul(c, b);
prints("\r\n");
printf(a);
prints(" multiply by ");
printf(b);
prints(" is ");
printf(c);
return;
```

Notes -

Text

locate

Arguments locate(<x>, <y>)

Description Equivalent to LOCATE #s, <x>, <y> command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte n;

mode(1);
for (n = 1; n <= 20; ++n)
{
  locate(n, n);
  printc(143);
  prints("position ");
  printb(n);
  printc(',');
  printb(n);
}
return;
```

Notes -

Text

pos

Arguments pos()

Description Equivalent to POS(#s) function in BASIC, where #s is the current stream specified with stream function.

Result Same as POS function in BASIC.

Example

```
include "cpc6128.ccz80";

byte n, p;

mode(1);
border(0, 0);
locate (6,2);
```

```
prints("Use left and right cursor keys");
window(1, 40, 12, 12);
cursor(1, 1);
for (n = 1; n <= 19; ++n) printc(9);
while (1)
{
  if (testkey(1) != -1) printc(9);
  if (testkey(8) != -1) printc(8);
  p = pos();
  stream(1);
  locate(2, 24);
  prints("Cursor text horizontal position = ");
  printb(p);
  stream(0);
}
```

Notes -

Text

vpos

Arguments vpos()

Description Equivalent to VPOS(#s) function in BASIC, where #s is the current stream specified with stream function.

Result Same as VPOS function in BASIC.

Example

```
include "cpc6128.ccz80";

byte p;

mode(1);
border(0, 0);
locate(8, 2);
prints("Use up/down cursor keys");
window(39, 39, 1, 25);
cursor(1, 1);
locate(1, 13);
while (1)
{
if (testkey(0) != -1) printc(11);
if (testkey(2) != -1) printc(10);
  p = vpos();
stream(1);
locate(1, 23);
prints("Vertical position of\r\ntext cursor = ");
printb(p);
stream(0);
}
```

Notes -

Text

stream

Arguments stream(<n>)

Description Defines <n> as current stream for all functions working with stream.

Result The current stream before the change.

Example -

Notes -

Text

window

Arguments window(<x1>, <x2>, <y1>, <y2>)

Description Equivalent to WINDOW #s, <x1>, <x2>, <y1>, <y2> command in BASIC, where #s is the active stream defined with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

mode(0);
border(0, 0); // Test screen
ink(0, 0, 0);
ink(1, 25, 25);
ink(2, 23, 23);
ink(3, 21, 21);
ink(4, 17, 17);
ink(5, 6, 6);
ink(6, 2, 2);
ink(7, 26, 26);
paper(0);
cls();
paper(1);
window(2, 4, 1, 18);
cls();
paper(2);
window(5, 7, 1, 18);
cls();
paper(3);
window(8, 10, 1, 18);
cls();
paper(4);
window(11, 13, 1, 18);
cls();
paper(5);
window(14, 16, 1, 18);
cls();
paper(6);
window(17, 19, 1, 18);
cls();
paper(7);
window(2, 19, 19, 25);
cls();
while (1); // Infinite loop
```

Notes -

Text

windowswap

Arguments windowswap(<n1>, <n2>)

Description Equivalent to WINDOW SWAP <n1>, <n2> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte t;

mode(1);
ink(1, 24, 24);
ink(2, 9, 9);
ink(3, 6, 6);

window(21, 40, 13, 25);
paper(3);
cls();
prints(" Window number 0");

stream(1);
window(1, 20, 1, 12);
paper(2);
```

```
cls();
prints(" Window number 1");

while (1)
{
stream(0);
locate(1, 6);
prints(" Red window (0)  ");

stream(1);
locate(1, 6);
prints(" Green window (1)");

for (t = 1; t <= 20; ++t) frame(); // Delay
windowswap(0, 1);
}
```

Notes -

Text

symbolafter

Arguments symbolafter(<n>, <buffer>)

Description Equivalent to SYMBOL AFTER <n> command in BASIC, where <buffer> is a space of 8 * (256 - <n>) bytes for the character table.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte symbol_buffer[1128]; // (256 - 115) * 8 = 1128
array byte symbol_data = { 0, 56, 64, 64, 48, 8, 8, 112 };

cls();
symbolafter(115, symbol_buffer);
symbol(115, symbol_data);
prints("a s\r\n");
prints("This definition of s is canceled with\r\nthe next line.\n\r");
symbolafter(240, symbol_buffer);
prints("a s");
return;
```

Notes -

Text

symbol

Arguments symbol(<n>, <array>)

Description Equivalent to SYMBOL <n>, <n1>, <n2>, <n3>, <n4>, <n5>, <n6>, <n7>, <n8> command in BASIC where <n1> to <n8> are the elements in <array>.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte symbol_buffer[1208]; // (256 - 105) * 8 = 1208
array byte symbol_data = { 255, 129, 189, 153, 153, 189, 129, 255 };

mode(1);
symbolafter(105, symbol_buffer);
prints("Next line redefine the letter i (105).\r\n");
prints("Type this letter some times.");
symbol(105, symbol_data);
return;
```

Notes -

tag

Arguments tag()

Description Equivalent to TAG #s command in BASIC, where #s is the current stream specified with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte a[256];
word x, y, f, b;

prints("Type your name: ");
input(a);
cls();
prints("You move too ");
prints(a);
tag();
while (1)
{
  x = strlen(a) * 17;
  y = 50 + rand() % 300;
  move(-x, y);
  // Don't use f <= 640
  for (f = 0; f <= 640; f += 4)
  {
    move(f, y);
    printc(' ');
    prints(a);
    frame();
  }
  for (b = 640; b; b -= 4)
  {
    move(b, y);
    prints(a);
    printc(' ');
    frame();
  }
}
```

Notes -

tagoff

Arguments tagoff()

Description Equivalent to TAGOFF #s command in BASIC, where #s is the current stream specified with stream function.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word x, period;
array byte text[6];

mode(2);
tag(); // Text linked to graphic cursor
period = 1984;
for (x = 1; x <= 640; x += 70)
{
  move(x, 400);
  drawr(0, -350);
  *wtoa(text, ++period) = ' ';
  prints(text);
}
tagoff(); // Text unlinked to graphic cursor
locate(35, 25);
prints("Yearly data");
```

```
while (1); // Infinite loop
```

Notes -

Text

copychr

Arguments copychr()

Description Equivalent to COPYCHR$(#s) function in BASIC, where #s is the current stream specified with stream function.

Result The readed character code or 0 if it's not recognized.

Example

```
include "cpc6128.ccz80";

byte a;

cls();
prints("Left top corner");
locate(1, 1);
a = copychr();
locate(1, 20);
printc(a);
return;
```

Notes -

Text

textmode

Arguments textmode(<n>)

Description If <n> is 0 disables the transparent text mode, if <n> isn't activates it.

Result Return value not significant.

Example -

Notes -

Text

cursor

Arguments cursor(<system>, <user>)

Description Equivalent to CURSOR <system>, <user> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte a;

cursor(1, 1);
prints("question?");
while (!(a = inkey()));
printc(a);
cursor(0, 0);
return;
```

Notes -

clg

Arguments clg()

Description Equivalent to CLG command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

locate(1, 20);
graphicspaper(3);
clg();
return;
```

Notes -

origin

Arguments origin(<x>, <y>)

Description Equivalent to ORIGIN <x>, <y> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word x;

mode(1);
border(13, 13);
tag();
origin(0, 0);
graphicswindow(100, 540, 300, 100);
graphicspaper(3);
clg();
while (1)
  // Don't use <= -340 !
  for (x = 550; x != -350; x -= 10)
  {
    move(x, 206);
    prints("This is a graphics window ");
    frame();
  }
```

Notes -

graphicswindow

Arguments graphicswindow(<x1>, <x2>, <y1>, <y2>)

Description Equivalent to specify <n1>, <n2>, <n3>, <n4> as 3th to 6th parameters in ORIGIN commando in BASIC.

Result Return value not significant.

Example -

Notes -

graphicspen

Arguments graphicspen(<n>)

Description Equivalent to GRAPHICS PEN <n> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

mode(0);
graphicspen(15);
move(200, 0);
draw(200, 400);
move(639, 0);
fill(15);
return;
```

Notes -

Graphics

graphicspaper

Arguments graphicspaper(<n>)

Description Equivalent to GRAPHICS PAPER <n> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

mode(0);
mask(15, 1);
graphicspaper(3);
draw(640, 0);
return;
```

Notes -

Graphics

xpos

Arguments xpos()

Description Equivalent to XPOS function in BASIC.

Result Same as XPOS function in BASIC.

Example

```
include "cpc6128.ccz80";

mode(2);
draw(320, 200);
prints("Horizontal position of graphic cursor = ");
printw(xpos());
return;
```

Notes -

Graphics

ypos

Arguments ypos()

Description Equivalent to YPOS function in BASIC.

Result Same as YPOS function in BASIC.

Example

```
include "cpc6128.ccz80";

mode(2);
draw(320, 200);
prints("Vertical position of graphic cursor = ");
printw(ypos());
return;
```

Notes -

Graphics

move

Arguments move(<x>, <y>)

Description Equivalent to MOVE <x>, <y> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word x, y;

mode(1);
tag();
while (1)
{
  x = rand() % 800 - 100;
  y = rand() % 430;
  move(x, y);
  prints("I am here");
}
```

Notes -

Graphics

mover

Arguments mover(<x>, <y>)

Description Equivalent to MOVER <x>, <y> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte n;

mode(1);
tag();
move(0, 16);
prints("first up");
for (n = 1; n <= 8; ++n)
{
  mover(-28, 16);
  prints("up");
}
prints(" and then down");
for (n = 1; n <= 8; ++n)
{
  mover(-56, -16);
  prints("down");
}
return;
```

Notes -

Graphics

plot

Arguments plot(<x>, <y>)

Description Equivalent to PLOT <x>, <y> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word i;
array byte n[5], t[4];
array byte x[5], y[5];
array byte time1[5], time2[5];

mode(1);
border(0, 0);
paper(0);
pen(1);
ink(0, 0, 0);
ink(1, 26, 26);
ink(2, 13, 26);
deg();
for (i = 1; i <= 360; ++i)
{
  origin(320, 200);
  graphicspen(1);
  btof(n, 50);
  draw(ftow(mul(cos(wtof(x, i)), n)), ftow(mul(sin(wtof(y, i)), n)));
  plot(ftow(mul(cos(wtof(x, i)), btof(n, 100))), ftow(mul(sin(wtof(y, i)),
btof(n, 25))));
}
origin(0, 0);
add(ltof(time1, time(t)), wtof(n, 300));
while (cmp(ltof(time2, time(t)), time1) == -1) plot(rand() % 640, rand() %
400);
graphicspen(2);
plot(rand() % 640, rand() % 400);
while (1); // Infinite loop
```

Notes -

Graphics

plotr

Arguments plotr(<x>, <y>)

Description Equivalent to PLOTR <x>, <y> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

// Paint lines with cursor keys
border(0, 0);
graphicspen(1);
mode(1);
start:
plot(320, 200);
while (1)
{
if (!testkey(0)) plotr(0, 1);
if (!testkey(1)) plotr(1, 0);
if (!testkey(2)) plotr(0, -1);
if (!testkey(8)) plotr(-1, 0);
if (!testkey(9)) goto start;
}
```

Notes -

Graphics

draw

Arguments draw(<x>, <y>)

Description Equivalent to DRAW <x>, <y> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word x, y;
byte z;

mode(0);
border(0, 0);
paper(0);
ink(0, 0, 0);
while (1)
{
  x = rand() % 640;
  y = rand() % 400;
  z = rand() % 15;
  graphicspen(z);
  draw(x, y);
}
```

Notes -

Graphics

drawr

Arguments drawr(<x>, <y>)

Description Equivalent to DRAWR <x>, <y> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte n;

cls();
prints("stairs");
move(0, 350);
for (n = 1; n <= 8; ++n)
{
  drawr(50, 0);
  drawr(0, -50);
}
move(340, 0);
fill(3);

while (1); // Infinite loop
```

Notes -

Graphics

fill

Arguments fill(<n>)

Description Equivalent to FILL <n> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word n;
byte pencolor;

mode(0);
for (n = 1; n <= 500; ++n) printc('O');
while (1)
{
  pencolor = 2 + rand() % 14;
  fill(pencolor);
}
```

Notes Not exists in cpc464.ccz80 library.

Graphics

test

Arguments test(<x>, <y>)

Description Equivalent to TEST <x>, <y> command in BASIC.

Result Same as TEST function in BASIC.

Example

```
include "cpc6128.ccz80";

cls();
prints("Use the pen number ");
printb(test(10, 386));
prints("\r\nChange the pen and the mode ... and run another time the
program.");
return;
```

Notes -

Graphics

testr

Arguments testr(<x>, <y>)

Description Equivalent to TESTR <x>, <y> command in BASIC.

Result Same as TESTR function in BASIC.

Example

```
include "cpc6128.ccz80";

byte x, n;
array byte string[11];

mode(0);
for (x = 1; x <= 15; ++x)
{
  locate(1, x);
  pen(x);
  prints(strset(string, 143, 10));
}
move(200, 400);
pen(1);
for (n = 1; n <= 23; ++n)
{
  locate(12, n);
  prints("pen ");
  printb(testr(0, -16));
}
return;
```

Notes -

graphicsmode

Arguments graphicsmode(<n>)

Description Equivalent to specify <n> value as 4th parameter in PLOT, PLOTR, DRAW, DRAWR, MOVE, MOVER commands in BASIC.

Result Return value not significant.

Example -

Notes -

Graphics

mask

Arguments mask(<pattern>, <first>)

Description Equivalent to MASK <pattern>, <first> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte x;

mode(0);
ink(5, 21, 21);
ink(8, 16, 16);
while (1)
{
move(-(rand() % 100), rand() % 400);
while (xpos() < 640)
{
  for (x = 1; x <= 8; ++x)
  {
    mask(1 << (8 - x), 1);
    graphicspen(x);
    graphicsmode(1);
    drawr(32, 0);
    mover(-32, 0);
  }
  mover(34, 0);
}
}
```

Notes Not exists in cpc464.ccz80 library.

Screen

mode

Arguments mode(<n>)

Description Equivalent to MODE <n> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte m = 0;

while (1)
{
if (++m > 2) m = 0;
mode(m);
```

```
prints("This is mode ");
printb(m);
prints("\r\nPress any key.");
while (!inkey());
}
```

Notes -

Screen

border

Arguments border(<n1>, <n2>)

Description Equivalent to BORDER <n1>, <n2> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

// 729 combinations for border
byte a,b;
word t;

speedink(5, 5);
for (a = 0; a <= 26; ++a)
  for (b = 0; b <= 26; ++b)
  {
    border(a, b);
    cls();
    locate(14, 13);
    prints("border ");
    printb(a);
    printc(',');
    printb(b);
    for (t = 1; t <= 50; ++t) frame(); // Delay
  }
return;
```

Notes -

Screen

ink

Arguments ink(<n>, <n1>, <n2>)

Description Equivalent to INK <n>, <n1>, <n2> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte p, i, t;

mode(1);
paper(0);
pen(1);
for (p = 0; p <= 1; ++p)
  for (i = 0; i <= 26; ++i)
  {
    ink(p, i, i);
    locate(16, 12);
    prints("ink ");
    printb(p);
    printc(',');
    printb(i);
    for (t = 1; t <= 50; ++t) frame(); // Delay
  }
ink(0, 1, 1);
ink(1, 24, 24);
cls();
return;
```

Notes -

Screen

speedink

Arguments speedink(<n1>, <n2>)

Description Equivalent to SPEED INK <n1>, <n2> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

byte i, t;

border(7, 18);
for (i = 30; i >= 1; --i)
{
  speedink(i, i);
  for (t = 1; t <= 20; ++t) frame(); // Delay
}
return;
```

Notes -

Screen

frame

Arguments frame()

Description Equivalent to FRAME command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word x;
byte f;

start:
f = 0;
mode(0);
prints("without FRAME");
start2:
tag();
move(0, 200);
for (x = 0; x <= 500; x += 4)
{
  if (f == 1) frame();
  move(x, 200);
  printc(' ');
  printc(143);
}
if (f == 1) goto start;
cls();
tagoff();
prints("with FRAME");
f = 1;
goto start2;
```

Notes -

Sound

sound

Arguments sound(<channel>, <frecuency>, <duration>, <volume>, <volume envelope>, <tone

envelope>, <noise>)

Description Equivalent to SOUND <channel>, <frecuency>, <duration>, <volumen envelope>, <tone envelope>, <noise>

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

word z;

for (z = 0; z <= 4095; ++z) sound(1, z, 1, 12, 0, 0, 0);
return;
```

Notes -

Sound

release

Arguments release(<channel>)

Description Equivalent to RELEASE <channel> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

sound(65, 1000, 100, 12, 0, 0, 0);
prints("Press R to release the sound");
while (testkey(50) == -1);
release(1);
return;
```

Notes -

Sound

ent

Arguments ent(<number>, <sections>)

Description Defines the tone envelope <number> with sections number specified at first byte in array <sections>, and each next three bytes is a section equivalent to a section specified in ENT command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte sections = { 2, 10, 206, 10, 10, 50, 10 };

ent(1, sections); // ENT 1,10,-50,10,10,50,10
sound(1, 500, 200, 10, 0, 1, 0);
return;
```

Notes -

Sound

env

Arguments env(<number>, <sections>)

Description Defines the volume envelope <number> with sections number specified at first byte in array <sections>, and each next three bytes is a section equivalent to a section specified in ENV command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte sections = { 2, 15, 255, 10, 15, 1, 10 };

env(1, sections); // ENV 1,15,-1,10,15,1,10
sound(1, 200, 300, 15, 1, 0, 0);
return;
```

Notes -

sq

Arguments sq(<channel>)

Description Equivalent to SQ(<channel>) function in BASIC.

Result Same as SQ function in BASIC.

Example

```
include "cpc6128.ccz80";

sound(65, 100, 100, 12, 0, 0, 0);
printb(sq(1));
return;
```

Notes -

onsq

Arguments onsq(<channel>, <label>)

Description Equivalent to ON SQ <channel> GOSUB <label> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte env_data = { 1, 15, 255, 255 };
array word sound_list = { 50, 60, 90, 100, 35, 200, 24, 500, 0 };
word sound_pointer;
byte x;
word s;

sound_pointer = sound_list;
env(1, env_data);
onsq(1, sound_routine);
mode(0);
origin(0, 0);
graphicswindow(200, 440, 100, 300);
while (1)
for (x = 1; x <=13 ; ++x)
{
  frame();
  graphicspen(x);
  move(320, 200);
  fill(x);
}

sound_routine:
s = **sound_pointer;
if (!s)
{
  sound_pointer = sound_list;
  goto sound_routine;
}
sound(1, s, 25, 15, 1, 0, 0);
onsq(1, sound_routine);
```

```
return;
```

Notes Call to SOUND ARM EVENT not works.

File

openin

Arguments openin(<file>)

Description Equivalent to OPENIN <file> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte text[256];
array byte n[5], a[256];

// Open a disc file for input
openin("data");
finputs(text);
atof(n, text);
finputs(text);
strcpy(a, text);
closein();
prints("The two values are:\r\n");
printf(n);
printc(' ');
prints(a);
return;
```

Notes -

File

closein

Arguments closein()

Description Equivalent to CLOSEIN command in BASIC.

Result Return value not significant.

Example

```
See openin function.
```

Notes -

File

finputc

Arguments finputc()

Description Reads the next character from file opened for input.

Result The next character readed from file opened with openin function or 0 if error.

Example -

Notes -

File

finputs

Arguments finputs(<string>)

Description Equivalent to LINE INPUT #9, <string> command in BASIC.

Result Returns <string> address.

Example -

Notes -

File

eof

Arguments eof()

Description Equivalent to EOF function in BASIC.

Result Returns 1 if the EOF has been reached.

Example

```
include "cpc6128.ccz80";

array byte a[256];

openin("ex1.bas");
while (!eof())
{
  finputs(a);
  prints(a);
  prints("\r\n");
}
closein();
return;
```

Notes -

File

load

Arguments load(<address>)

Description Equivalent to LOAD <file>, <address> command in BASIC, where <file> is a file before opened with openin function.

Result Return value not significant.

Example -

Notes -

File

openout

Arguments openout(<file>)

Description Equivalent to OPENOUT <file> command in BASIC.

Result Return value not significant.

Example

```
include "cpc6128.ccz80";

array byte text[256];
array byte n[16], a[256];

// Open a file disc for output
prints("Type a number: ");
input(text);
atof(n, text);
prints("\r\nType a word: ");
input(a);
openout("data");
fprintf(n);
fprints("\r\n");
fprints(a);
fprints("\r\n");
closeout();
prints("\r\nData saved in disc");
return;
```

Notes -

File

closeout

Arguments closeout()

Description Equivalent to CLOSEOUT command in BASIC.

Result Return value not significant.

Example `See openout function.`

Notes -

File

fprintc

Arguments fprintc(<character>)

Description Equivalent to PRINT #9, CHR$(<character>); command in BASIC.

Result Return value not significant.

Example -

Notes -

File

fprints

Arguments fprints(<string>)

Description Equivalent to PRINT #9, <string>; command in BASIC.

Result Return value not significant.

Example -

Notes -

File

fprintb

Arguments fprintb(<n>)

Description Equivalent to PRINT #9, <n>; command in BASIC.

Result Return value not significant.

Example -

Notes -

File

fprintw

Arguments fprintw(<n>)

Description Equivalent to PRINT #9, <n>; command in BASIC.

Result Return value not significant.

Example -

Notes -

File

save

Arguments save(<address>, <length>, <start>)

Description Equivalent to SAVE <file>, B, <address>, <lenght>, <start> command in BASIC, where <file> is the file before opened with openout function.

Result Return value not significant.

Example -

Notes -

File

speedwrite

Arguments speedwrite(<n>)

Description Equivalent to SPEED WRITE <n> command in BASIC.

Result Return value not significant.

Example -

Notes -

Printer

lprintc

Arguments lprintc(<character>)

Description Equivalent to PRINT #8, CHR$(<character>); command in BASIC.

Result Return value not significant.

Example -

Notes -

Printer

lprints

Arguments lprints(<string>)

Description Equivalent to PRINT #8, <string>; command in BASIC.

Result Return value not significant.

Example -

Notes -

Printer

lprintb

Arguments lprintb(<n>)

Description Equivalent to PRINT #8, <n>; command in BASIC.

Result Return value not significant.

Example -

Notes -

Printer

lprintw

Arguments lprintw(<n>)

Description Equivalent to PRINT #8, <n>; command in BASIC.

Result Return value not significant.

Example -

Notes -

Math

fcpy

Arguments fcpy(<real1>, <real2>)

Description Copy <real2> in <real1>.

Result Return <real1> address.

Example -

Notes -

Math

btof

Arguments btof(<real>, <n>)

Description Converts the byte value <n> to corresponding real value in <real>.

Result The <real> address.

Example -

Notes -

Math

wtof

Arguments wtof(<real>, <n>)

Description Converts the word value <n> to corresponding real value in <real>.

Result The <real> address.

Example -

Notes -

Math

ftob

Arguments ftob(<real>)

Description Converts the <real> value to corresponding byte value.

Result The converted value.

Example -

Notes -

Math

ftow

Arguments ftow(<real>)

Description Converts the <real> value to corresponding word value.

Result The converted value.

Example -

Notes -

Math

ltof

Arguments ltof(<real>, <long>)

Description Converts the long value in <long> to corresponding real value in <real>.

Result The <real> address.

Example -

Notes -

Math

ftol

Arguments ftol(<long>, <real>)

Description Converts the real value in <real> to corresponding long value in <long>.

Result The <real> address.

Example -

Notes -

Math

ftoa

Arguments ftoa(<string>, <real>)

Description Converts the real value in <real> to corresponding ASCII representation in <string>.

Result The <string> address.

Example -

Notes -

Math

atof

Arguments atof(<real>, <string>)

Description Converts the ASCII representation in <string> to corresponding real value in <real>.

Result The <real> address.

Example -

Notes -

Math

add

Arguments add(<real1>, <real2>)

Description Equivalent to <real1> = <real1> + <real2> expression in BASIC.

Result The <real1> address.

Example -

Notes -

Math

sub

Arguments sub(<real1>, <real2>)

Description Equivalent to <real1> = <real1> - <real2> expression in BASIC.

Result The <real1> address.

Example -

Notes -

Math

mul

Arguments mul(<real1>, <real2>)

Description Equivalent to <real1> = <real1> * <real2> expression in BASIC.

Result The <real1> address.

Example -

Notes -

Math

div

Arguments div(<real1>, <real2>)

Description Equivalent to <real1> = <real1> / <real2> expression in BASIC.

Result The <real1> address.

Example -

Notes -

Math

cmp

Arguments cmp(<real1>, <real2>)

Description Compares <real1> with <real2>.

Result Returns 0 if <real1> equal to <real2>, returns 1 if <real1> greather than <real2>, returns -1 if <real1> less than <real2>.

Example -

Notes -

Math

abs

Arguments abs(<real>)

Description Equivalent to <real> = ABS(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(abs(atof(number, "-67.98")));
return;
```

Notes -

Math

atn

Arguments atn(<real>)

Description Equivalent to <real> = ATN(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(atn(btof(number, 1)));
return;
```

Notes -

Math

cos

Arguments cos(<real>)

Description Equivalent to <real> = COS(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

deg();
printf(cos(btof(number, 45)));
return;
```

Notes -

exp

Arguments exp(<real>)

Description Equivalent to <real> = EXP(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(exp(atof(number, "6.876")));
return;
```

Notes -

fix

Arguments fix(<real>)

Description Equivalent to <real> = FIX(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(fix(atof(number, "9.99999")));
return;
```

Notes -

int

Arguments int(<real>)

Description Equivalent to <real> = INT(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(int(atof(number, "-1.995")));
return;
```

Notes -

log

Arguments log(<real>)

Description Equivalent to <real> = LOG(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(log(wtof(number, 9999)));
return;
```

Notes -

Math

log10

Arguments log10(<real>)

Description Equivalent to <real> = LOG10(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(log10(wtof(number, 9999)));
return;
```

Notes -

Math

neg

Arguments neg(<real>)

Description Equivalent to <real> = -<real> expression in BASIC.

Result Returns <real> address.

Example -

Notes -

Math

pi

Arguments pi(<real>)

Description Equivalent to <real> = PI expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(pi(number));
return;
```

Notes -

Math

pow

Arguments pow(<real1>, <real2>)

Description Equivalent to <real1> = <real1> ^ <real2> expression in BASIC.

Result Returns <real1> address.

Example -

Notes -

Math

round

Arguments round(<real>, <n>)

Description Equivalent to <real> = ROUND(<real>, <n>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

byte n;
array byte f[5];

// Don't use n >= -3
for (n = 3; n != -4; --n)
{
  printf(round(atof(f, "1234.5678"), n));
  prints(" with <decimals> = ");
  if (n & #80)
  {
    // n is negative
    printc('-');
    printb(-n);
  }
  else
  {
    printb(n);
  }
  prints("\r\n");
}
return;
```

Notes -

Math

sgn

Arguments sgn(<real>)

Description Evaluates the <real> sign.

Result Returns 1 if value is positive, 0 if it's zero and -1 if value is negative.

Example

```
include "cpc6128.ccz80";

byte i, s;
array byte n[5], f[5];

for (i = 1, btof(n, 200); i <= 21; ++i, sub(n, btof(f, 20)))
{
  prints("sgn is ");
  s = sgn(n);
  if (s & 80)
  {
```

```
    printc('-');
    printb(-s);
  }
  else
  {
    printb(s);
  }
  prints(" when number is ");
  printf(n);
  prints("\r\n");
}
return;
```

Notes -

Math

sin

Arguments sin(<real>)

Description Equivalent to <real> = SIN(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

word i;
array byte y[5], n[5], f[5];

cls();
deg();
origin(0, 200);
for (i = 0; i <= 720; ++i)
{
  sin(wtof(y, i));
  plot(ftow(div(mul(wtof(n, i), wtof(f, 640)), wtof(f, 720))), ftow(mul(y,
wtof(f, 198))));
}
while (1); // Infinite loop
```

Notes -

Math

sqr

Arguments sqr(<real>)

Description Equivalent to <real> = SQR(<real>) expression in BASIC.

Result Returns <real> address.

Example

```
include "cpc6128.ccz80";

array byte number[5];

printf(sqr(btof(number, 9)));
return;
```

Notes -

Math

tan

Arguments tan(<real>)

Description Equivalent to <real> = TAN(<real>) expression in BASIC.

Result Returns <real> address.

Example
```
include "cpc6128.ccz80";

array byte number[5];

printf(tan(btof(number, 45)));
return;
```

Notes -

Math

deg

Arguments deg()

Description Equivalent to DEG command in BASIC.

Result Return value not significant.

Example -

Notes -

Math

rad

Arguments rad()

Description Equivalent to RAD command in BASIC.

Result Return value not significant.

Example -

Notes -

Math

printf

Arguments printf(<real>)

Description Equivalent to PRINT <real> ; command in BASIC.

Result Return value not significant.

Example -

Notes -

Math

lprintf

Arguments lprintf(<real>)

Description Equivalent to PRINT #8, <real> ; command in BASIC.

Result Return value not significant.

Example -

Notes -

fprintf

Arguments fprintf(<real>)

Description Equivalent to PRINT #9, <real> ; command in BASIC.

Result Return value not significant.

Example -

Notes -