JOYCE for PCW programmers

JOYCE makes the following special opcodes available to Z80 programs. Some functions may only be present in certain versions; in this case, a version number in brackets, such as (v1.28+) will show the minimum version that supports this call. If so, programs must first check that the JOYCE version number is greater than or equal to the number shown.

ED FD: Replace BIOS function

If this opcode is executed at 0FC4Eh, reads the PC's real-time clock and sets the time fields of the SCB. Thus the BIOS "TIME" jump can be replaced by 0EDh 0FDh 0C9h. This is done by RTC.COM.

(v1.28+) Similarly, this can be inserted at 0FC0Fh (LIST) and 0FC2Dh (LISTST) to capture printing - CENPORT.COM does this.

In ANNE, the same instruction sequence is inserted automatically into the Rosanne BIOS jumpblock by the 'Rosanne Acceleration' menu.

In the future, other functions may be replaced in this way.

ED FE: JOYCE API.

This is the main entry point for JOYCE functions. What function is executed is determined by the A register.

A=0: Test for JOYCE.

Returns A=0FFh, HL=JOYCE version number (packed BCD). If A is not 0 and not 0FFh, then JOYCE has been changed in a way that would break existing programs.

A=1: Boot.

Enter with B = boot disc number (boot*b*.dsk). Reads the first sector and jumps to it. Only returns if the disc was not bootable.

A=2: Set screen colours.

Enter with BCD = RGB for "white", EHL = RGB for "black".

A=3: FID_D_LOGIN for JOYCEDRV.FID

Parameters and results as for the corresponding Z80 routine.

A=4: FID_D_READ for JOYCEDRV.FID

Parameters and results as for the corresponding Z80 routine.

A=5: FID_D_WRITE for JOYCEDRV.FID

Parameters and results as for the corresponding Z80 routine.

A=6: FID_D_FLUSH for JOYCEDRV.FID

Parameters and results as for the corresponding Z80 routine.

A=7: FID_EMS helper for JOYCEDRV.FID

Enter with:

BC= FID flags. If C=0FFh, the FID only supports MYZ80-format hard drives.

D= reason for call:

- **0:** Drive successfully added (drive number in E).
- **1:** Drive not added - no memory available
- **2:** Drive not added - all drive letters in use
- **3:** Incorrect FID environment - JOYCEDRV cannot run.
- **4:** Initialise - this is the first call made to this function by JOYCEDRV.FID.

HL= address of message buffer; messages can be copied to here by JOYCE.

Returns:

A=0 Return with carry clear and message at HL.

A=1 Return with carry set and message at HL.

A=2 Attempt to add a drive. B=drive number; HL IX IY hold parameters for the SVC_D_HOOK routine.

A=8: Save JOYCE settings.

A=9: Set/Get current LPT port.

(DOS only) This call does not have any effect in JOYCE 1.9.0+. In earlier versions, it switches to the LPT port indicated by DE, and returns the previous port number in HL.. DE=0,1,2 for LPT1, LPT2, LPT3; or 0FFFFh to return current assignment in HL.

A=10: Operations on host filesystem files.

The operations are selected using the C register:

C=0Fh: Open existing file for read/write. DE->pathname (0 terminated). Returns HL=handle or 0 if failed.

C=10h: Close file. DE=handle. Returns HL=0 if failed, 1 if OK.

C=14h: Read byte from open file. DE=handle. Returns byte in B; HL=0 if end of file, 1 if OK.

C=15h: Write a byte to open file. DE=handle, B=byte. Returns HL=0 if failed, 1 if OK.

C=16h: Create new file for read/write, DE->name (0 terminated). Returns HL=0 if failed, file handle if succeeded.

C=69h: Get the local time from the PC clock. DE->time buffer (in CP/M datestamp format). Returns A=BCD seconds, and buffer filled.

C=91h: Expand filename with wildcards.

DE-> ambiguous filename, 0-terminated.

HL-> result buffer

B= which file (0=first, 1=second...)

Returns HL=0 if not found, 1 if found.

First 11 bytes of buffer hold filename in FCB form

Remainder of buffer holds ASCIIZ filename with path

A=11: (v1.10+) Screen operations

The subfunction to perform is given in the C register:

0 Select standard emulated PCW screen.

1 Select 800x600x256 screen.

2 Write character in E to the 800x600 screen.

3 Write status line character in E to the 800x600 screen.

4 (v1.20+) Execute a GSX call. DE->GSX PB

5 FID_EMS helper for VGA.FID. Expands an error number in D to a message at HL. Error numbers are:

0 OK.

1 Device table full

2 Attempt to load VGA.FID twice

3 CP/M environment is not compatible.

6 (v1.20+) Get capabilities bitmap for screen into HL.

Bit 0 set if 800x600 mode is possible.

Bit 1 set if GSX is possible (versions 1.20 to 1.36, and 2.1.7 onward).

Bit 2 set if LIOS is possible.

Other bits of HL, DE BC IX IY all zero (reserved)

For versions < 1.10, none of these is possible.

For versions < 1.20, 800x600 mode (only) is possible.

A=12: (v1.12+) Keyboard operations

The subfunction is given by C:

0: Get mapping for key DE, into HL.

1: Set mapping for key DE, in HL.

2: Set mapping for <DE> keys, at (IX). Each mapping is 4 bytes:

```
DW de_value
DW hl_value
```

In each case, the "de_value" is the SDL key number (see SDL_keysym.h). The low byte of the "hl_value" is the offset of the byte corresponding to the key, minus 0FFF0h (so it will be 0-16; 0-15 for PCW keys and 16 for JOYCE special functions). The high byte of the "hl_value" is the mask of bits to set/clear in that byte.

A=13: (v1.21-v1.36) Serial port operations

This function is a stub in v1.9.0+; serial comms should instead go via the emulated CPS8256. All the subfunctions may still be called, but they will either be no-ops or return dummy values.

C = subfunction:

0: Reset buffers

1: Set handshaking lines. E=lines:

bit 0 => DTR
bit 1 => RTS
bit 2 => USER
bit 4 => LOOPBACK

2: Set Baud rate & stop bits

HL= encoded baud rate.

```
 1:50     2:75     3:110    4:134    5:150
 6:300    7:600    8:1200   9:1800  10:2400
11:3600  12:4800  13:7200  14:9600
15:19200 16:14400 17:28800 18:33600
19:38400 20:57600 21:115200
```

E= settings:

bit 0 set for 8-bits, reset for 7
bit 1 = 1
bit 2 set for 2 stop bits, reset for 1
bit 3 set if parity, else 0
bit 4 set if parity even, clear if odd

3: Character input status. Returns A=0 if no character available, 1 if there is.

4: Poll for character, return it in A (A=0 if none)

5: Character output status. Returns A=0 if no, 1 if yes

6: Output character in E

7: FID_EMS helper for COM port FID

8: Switch to using COM port no. E

9: Get COM port no. into A

A=14: (v1.22+) FID_D_MESSAGE for JOYCEDRV.FID

All parameters and results the same as for the corresponding Z80 routine.

A=15: (v1.22+) Call the Logo I/O system.

- E = LIOS function number; SP->parameters. Results returned in HL
- E=0FFh to initialise, SP->DW address of info area, HL=address of message area. Returns HL=0 if error, 1 if OK

A=16: (v1.22+) Parse the boot list (joycebt.xml).

B=number of boot disc (0-9, or 0FFh). HL->16-byte buffer for name.
Returns A=1 if entry exists, else 0.
(v1.9.0+) If B = 0FFh, special function in C. Returns A=1 if OK, A=0 if function not supported. Special function is:

0 Choose boot disc using JOYCE menus, not Z80 code.

1 Display helpscreen.

This function is present in ANNE (v2.1.2+) but if B=0FFh it always returns A=0.

A=0FCh: (v1.30+) BDOS filter for REDIR.COM and its programs.

Entered with:

- Carry set.
- C = BDOS function
- DE = BDOS parameter
- HL = address of REDIR.RSX in memory.

Returns Carry set to continue BDOS call, Carry clear to return to caller.

A=0FDh: (v1.27+) Poll the mouse

If an automatic mouse patch has been applied, this will update the correct memory locations with the mouse coordinates and button status. If no such patch has been applied, nothing happens.

A=0FEh: Set / Get timing parameters.

This allows the timing of JOYCE functions to be adjusted.

A:=Screen refresh rate (900 / Hz value).

HL=Number of Z80 cycles that should be executed every $\frac{1}{900}$of a second.

C = if nonzero, new refresh rate (900 / Hz).

DE = if nonzero, new number of Z80 cycles to execute every $\frac{1}{900}$of a second. This changes the ratio between the speed of the processor and the speed of the PCW, rather than the overall speed.

Since JOYCE v2.x has automatic speed regulation, it should not be necessary to alter these figures from program code.

ED FF: (v2.1.2+) Stop emulation.

This opcode causes the emulated Z80 to stop. In normal use, it will cause the emulator to shut down instantly without asking for confirmation.

It is also used by the emulators if a native code function needs to call a Z80 function (JOYCE does not do this at the moment, but ANNE does). At the end of the Z80 function, the ED FF opcode will be executed to return to native code.